

DATA MINING AND PREDICTIVE ANALYTICS FOR BUSINESS DECISIONS

A Case Study Approach



ANDRES FORTINO

**DATA MINING AND
PREDICTIVE ANALYTICS
FOR BUSINESS DECISIONS**

LICENSE, DISCLAIMER OF LIABILITY, AND LIMITED WARRANTY

By purchasing or using this book and its companion files (the “Work”), you agree that this license grants permission to use the contents contained herein, but does not give you the right of ownership to any of the textual content in the book or ownership to any of the information, files, or products contained in it. *This license does not permit uploading of the Work onto the Internet or on a network (of any kind) without the written consent of the Publisher.* Duplication or dissemination of any text, code, simulations, images, etc. contained herein is limited to and subject to licensing terms for the respective products, and permission must be obtained from the Publisher or the owner of the content, etc., in order to reproduce or network any portion of the textual material (in any media) that is contained in the Work.

MERCURY LEARNING AND INFORMATION (“MLI” or “the Publisher”) and anyone involved in the creation, writing, production, accompanying algorithms, code, or computer programs (“the software”), and any accompanying Web site or software of the Work, cannot and do not warrant the performance or results that might be obtained by using the contents of the Work. The author, developers, and the Publisher have used their best efforts to insure the accuracy and functionality of the textual material and/or programs contained in this package; we, however, make no warranty of any kind, express or implied, regarding the performance of these contents or programs. The Work is sold “as is” without warranty (except for defective materials used in manufacturing the book or due to faulty workmanship).

The author, developers, and the publisher of any accompanying content, and anyone involved in the composition, production, and manufacturing of this work will not be liable for damages of any kind arising out of the use of (or the inability to use) the algorithms, source code, computer programs, or textual material contained in this publication. This includes, but is not limited to, loss of revenue or profit, or other incidental, physical, or consequential damages arising out of the use of this Work.

The sole remedy in the event of a claim of any kind is expressly limited to replacement of the book and only at the discretion of the Publisher. The use of “implied warranty” and certain “exclusions” vary from state to state, and might not apply to the purchaser of this product.

Companion files also available for downloading from the publisher by writing to info@merclearning.com.

DATA MINING AND PREDICTIVE ANALYTICS FOR BUSINESS DECISIONS

A Case Study Approach

Andres Fortino, PhD
NYU School of Professional Studies



MERCURY LEARNING AND INFORMATION
Dulles, Virginia
Boston, Massachusetts
New Delhi

Copyright ©2023 by MERCURY LEARNING AND INFORMATION LLC. All rights reserved.

This publication, portions of it, or any accompanying software may not be reproduced in any way, stored in a retrieval system of any type, or transmitted by any means, media, electronic display or mechanical display, including, but not limited to, photocopy, recording, Internet postings, or scanning, without prior permission in writing from the publisher.

Publisher: David Pallai

MERCURY LEARNING AND INFORMATION
22841 Quicksilver Drive
Dulles, VA 20166
info@merclearning.com
www.merclearning.com
1-800-232-0223

A. Fortino. *Data Mining and Predictive Analytics for Business Decisions*.
ISBN: 978-1-68392675-7

The publisher recognizes and respects all marks used by companies, manufacturers, and developers as a means to distinguish their products. All brand names and product names mentioned in this book are trademarks or service marks of their respective companies. Any omission or misuse (of any kind) of service marks or trademarks, etc. is not an attempt to infringe on the property of others.

Library of Congress Control Number: 2022950710
232425321 Printed on acid-free paper in the United States of America.

Our titles are available for adoption, license, or bulk purchase by institutions, corporations, etc. For additional information, please contact the Customer Service Dept. at 800-232-0223(toll free).

All of our titles are available for sale in digital format at numerous digital vendors. Companion files for this title can also be downloaded by writing to info@merclearning.com. The sole obligation of MERCURY LEARNING AND INFORMATION to the purchaser is to replace the book, based on defective materials or faulty workmanship, but not based on the operation or functionality of the product.

To my wife, Kathleen

CONTENTS

<i>Preface</i>	<i>xv</i>
<i>Acknowledgments</i>	<i>xix</i>
Chapter 1: Data Mining and Business	1
Data Mining Algorithms and Activities	2
Data is the New Oil	2
Data-Driven Decision-Making	3
Business Analytics and Business Intelligence	4
Algorithmic Technologies Associated with Data Mining	5
Data Mining and Data Warehousing	6
Case Study 1.1: Business Applications of Data Mining	7
Case A – Classification	7
Case B – Regression	7
Case C – Anomaly Detection	7
Case D – Time Series	8
Case E – Clustering	8
Reference	8
Chapter 2: The Data Mining Process	9
Data Mining as a Process	10
Exploration	10
Analysis	10
Interpretation	10
Exploitation	11
Selecting a Data Mining Process	11
The CRISP-DM Process Model	12
Business Understanding	12
Data Understanding	12
Data Preparation	13
Modeling	13
Evaluation	13

Deployment	13
Selecting Data Analytics Languages	14
The Choices for Languages	15
References	16
Chapter 3: Framing Analytical Questions	17
How Does CRISP-DM Define the Business and Data Understanding Step?	18
The World of the Business Data Analyst	19
How Does Data Analysis Relate to Business Decision-Making?	20
How Do We Frame Analytical Questions?	21
What Are the Characteristics of Well-framed Analytical Questions?	22
Exercise 3.1 – Framed Questions About the Titanic Disaster	23
Case Study 3.1 – The San Francisco Airport Survey	25
Case Study 3.2 – Small Business Administration Loans	26
References	29
Chapter 4: Data Preparation	31
How Does CRISP-DM Define Data Preparation?	32
Steps in Preparing the Data Set for Analysis	33
Data Sources and Formats	34
What is Data Shaping?	35
The Flat-File Format	35
Application of Tools for Data Acquisition and Preparation	37
Exercise 4.1 – Shaping the Data File	37
Exercise 4.2 – Cleaning the Data File	40
Ensuring the Right Variables are Included	42
Using SQL to Extract the Right Data Set from Data Warehouses	44
Case Study 4.1: Cleaning and Shaping the SFO Survey Data Set	45
Case Study 4.2: Shaping the SBA Loans Data Set	46
Case Study 4.3: Additional SQL Queries	48
Reference	49
Chapter 5: Descriptive Analysis	51
Getting a Sense of the Data Set	52
Describe the Data Set	53
Explore the Data Set	53
Verify the Quality of the Data Set	54
Analysis Techniques to Describe the Variables	54
Exercise 5.1 – Descriptive Statistics	54
Distributions of Numeric Variables	54
Correlation	55
Exercise 5.2 – Descriptive Analysis of the Titanic Disaster Data	57
Case Study 5.1: Describing the SFO Survey Data Set	59
Solution Using R	59

Solution Using Python	62
Case Study 5.2: Describing the SBA Loans Data Set	66
Solution Using R	67
Solution Using Python	72
Reference	76
Chapter 6: Modeling	77
What is a Model?	77
How Does CRISP-DM Define Modeling?	78
Selecting the Modeling Technique	79
Modeling Assumptions	79
Generate Test Design	79
Design of Model Testing	80
Build the Model	80
Parameter Setting	80
Models	80
Model Assessment	80
Where Do Models Reside in a Computer?	81
The Data Mining Engine	81
The Model	82
Data Sources and Outputs	82
Traditional Data Sources	83
Static Data Sources	83
Real-Time Data Sources	84
Analytic Outputs	84
Model Building	84
Step 1: Framing Questions	85
Step 2: Selecting the Machine	86
Step 3: Selecting Known Data	86
Step 4: Training the Machine	87
Step 5: Testing the Model	87
Step 6: Deploying the Model	88
Step 7: Collecting New Data	88
Step 8: Updating the Model	88
Step 9: Learning – Repeat Steps 7 and 8	88
Step 10: Recommending Answers to the User	89
Reference	89
Chapter 7: Predictive Analytics with Regression Models	91
What is Supervised Learning?	92
Regression to the Mean	92
Linear Regression	93
Simple Linear Regression	93
The R-squared Coefficient	95
The Use of the p-value of the Coefficients	96

Strength of the Correlation Between Two Variables	97
Exercise 7.1 – Using SLR Analysis to Understand Franchise Advertising	98
Multivariate Linear Regression	99
Preparing to Build the Multivariate Model	100
Exercise 7.2 – Using Multivariate Linear Regression to Model Franchise Sales	100
Logistic Regression	102
What is Logistic Regression?	103
Exercise 7.3 – PassClass Case Study	103
Multivariate Logistic Regression	105
Exercise 7.4 – MLR Used to Analyze the Results of a Database Marketing Initiative	105
Where is Logistic Regression Used?	107
Comparing Linear and Logistic Regressions for Binary Outcomes	108
Case Study 7.1: Linear Regression Using the SFO Survey Data Set	108
Solution in R	109
Solution in Python	112
Case Study 7.2: Linear Regression Using the SBA Loans Data Set	114
Solution in R	114
Solution in Python	115
Case Study 7.3: Logistic Regression Using the SFO Survey Data Set	116
Solution in R	117
Solution in Python	118
Case Study 7.4: Logistic Regression Using the SBA Loans Data Set	118
Solution in R	118
Solution in Python	119
Chapter 8: Classification	121
Classification with Decision Trees	122
Building a Decision Tree	122
Exercise 8.1 – The Iris Data Set	125
The Problem with Decision Trees	128
Classification with Random Forest	128
Using a Random Forest Model	129
Exercise 8.2 – The Iris Data Set	129
Classification with Naïve Bayes	131
Exercise 8.3 – The HIKING Data Set	131
Computing the Conditional Probabilities	132
Case Study 8.1: Classification with the SFO Survey Data Set	133
Solution in R	134
Solution in Python	135
Case Study 8.2: Classification with the SBA Loans Data Set	136
Solution in R	136
Solution in Python	137
Case Study 8.3: Classification with the Florence Nightingale Data Set	137
Solution in Python	138
Reference	139

Chapter 9: Clustering	141
What is Unsupervised Machine Learning?	141
What is Clustering Analysis?	142
Applying Clustering to Old Faithful Eruptions	142
Examples of Applications of Clustering Analysis	143
A Simple Clustering Example Using Regression	143
Hierarchical Clustering	145
Applying Hierarchical Clustering to Old Faithful Eruptions	147
Exercise 9.1 – Hierarchical Clustering and the Iris Data Set	148
K-Means Clustering	150
How Does the K-Means Algorithm Compute Cluster Centroids?	150
Applying K-Means Clustering to Old Faithful Eruptions	152
Exercise 9.2 – K-Means Clustering and the Iris Data Set	152
Hierarchical vs. K-Means Clustering	153
Case Study 9.1: Clustering with the SFO Survey Data Set	154
Solution in R	154
Solution in Python	159
Case Study 9.2: Clustering with the SBA Loans Data Set	167
Solution in R	167
Solution in Python	170
Chapter 10: Time Series Forecasting	173
What is a Time Series?	174
Time Series Analysis	174
Types of Time Series Analysis	175
What is Forecasting?	176
Exercise 10.1 – Analysis of the US and China GDP Data Set	176
Case Studies	178
Case Study 10.1: Time Series Analysis of the SFO Survey Data Set	178
Solution in Excel	179
Case Study 10.2: Time Series Analysis of the SBA Loans Data set	180
Solution in R	181
Solution in Python	183
Case Study 10.3: Time Series Analysis of a Nest Data Set	185
Solution in Python	186
Reference	188
Chapter 11: Feature Selection	189
Using the Covariance Matrix	190
Factor Analysis	191
When to Use Factor Analysis	192
First Step in FA – Correlation	192
FA for Exploratory Analysis	192
Selecting the Number of Factors – The Scree Plot	193

Example 11.1: Restaurant Feedback	194
Factor Interpretation	195
Summary Activities to Perform a Factor Analysis	196
Case Study 11.1: Variable Reduction with the SFO Survey Data Set	196
Solution in R	196
Solution in Python	198
Case Study 11.2: Hunting Diamonds	201
Solution in R	201
Solution in Python	203
Chapter 12: Anomaly Detection	205
What is an Anomaly?	205
What is an Outlier?	206
The Case Studies for the Exercises in Anomaly Detection	207
Anomaly Detection by Standardization – A Single Numerical Variable	207
Exercise 12.1 – Outliers in the Airline Delays Data Set – Z-Score	208
Anomaly Detection by Quartiles – Tukey Fences – With a Single Variable	208
Comparing Z-scores and Tukey Fences	209
Exercise 12.2 – Outliers in the Airline Delays Data Set – Tukey Fences	210
Anomaly Detection by Category – A Single Variable	211
Exercise 12.3 – Outliers in the Airline Delays Data Set – Categorical	212
Anomaly Detection by Clustering – Multiple Variables	213
Exercise 12.4 – Outliers in the Airline Delays Data Set – Clustering	214
Anomaly Detection Using Linear Regression by Residuals – Multiple Variables	214
Exercise 12.5 – Outliers in the Airline Delays Data Set – Residuals	215
Case Study 12.1: Outliers in the SFO Survey Data Set	217
Solution in R	217
Solution in Python	220
Case Study 12.2: Outliers in the SBA Loans Data Set	222
Solution in R	223
Solution in Python	224
References	225
Chapter 13: Text Data Mining	227
What is Text Data Mining?	228
What are Some Examples of Text-Based Analytical Questions?	229
Tools for Text Data Mining	230
Sources and Formats of Text Data	230
Term Frequency Analysis	231
How Does It Apply to Text Business Data Analysis?	232
Exercise 13.1 – Case Study Using a Training Survey Data Set	232
Word Frequency Analysis Using R	233
Keyword Analysis	234
Exercise 13.2 – Case Study Using Data Set D: Résumé and Job Description	234

Keyword Word Analysis in Voyant	235
Term Frequency Analysis in R	236
Visualizing Text Data	239
Exercise 13.3 – Case Study Using the Training Survey Data Set	240
Visualizing the Text Using Excel	240
Visualizing the Text Using Voyant	241
Visualizing the Text Using R	241
Text Similarity Scoring	243
What is Text Similarity Scoring?	243
Exercise 13.4 – Case Study Using the Occupation Description Data Set	244
Analysis Using an Online Text Similarity Scoring Tool	244
Similarity Scoring Analysis Using R	246
Exercise 13.5 – Rfsumf and Job Descriptions Similarly Scoring Using R	246
Case Study 13.1 – Term Frequency Analysis of Product Reviews	248
Term Frequency Analysis Using Voyant	249
Term Frequency Analysis Using R	250
References	251
Chapter 14: Working with Large Data Sets	253
Using Sampling to Work with Large Data Files	254
Exercise 14.1 – Big Data Analysis	254
Case Study 14.1 Using the BankComplaints Big Data File	258
Chapter 15: Visual Programming	259
Comparing Visual Programming to Command-line Coding	261
Leading Visual Programming Environments	261
Visual Programming with the SAS Enterprise Guide	262
Visual Programming with IBM SPSS	263
Visual Programming with RapidMiner	264
Visual Programming with Orange 3	265
Installing Orange 3	266
References	266
<i>Index</i>	267

PREFACE

Data mining is a recent development in the area of data analysis within the last 20 years. With many recent advances in data science, we now have many more tools and techniques available for data analysts to extract information from data sets. This book aims to assist data analysts to move up from simple tools such as Excel for descriptive analytics to answer more sophisticated questions using machine learning. Data mining is a very sophisticated and organized activity with a well-defined process encoded in the CRISP-DM standard. In this book we develop an understanding of the tools and techniques to assist the individual data analyst, but not necessarily a data science team. This book intends to assist individual data analysts in helping them improve their understanding and skills to answer more sophisticated questions.

Most of the exercises use R and Python, today's most common analysis tools. But rather than focus on coding algorithms with these tools, as is most often the case, we employ interactive interfaces to these tools to perform the analysis. That way, we can focus on the technique and its interpretation rather than developing coding skills. We rely on the Jamovi and the JASP interfaces to the R program and the Orange3 data mining interface to Python. Where appropriate, we introduce additional easy-to-acquire and use tools, such as Voyant, for text analytics, that are available as open source. The techniques covered in this book range from basic descriptive statistics, such as summarization and tabulation, to more sophisticated predictive techniques, such as linear and logistic regression, clustering, classification, and text analytics.

We follow the CRISP-DM process throughout, but only as a simple guide to the various steps without necessarily implementing all its procedures. We intend to focus on data analytics, not necessarily the more sophisticated data science approaches. This book is for you if you wish to improve your analytical skills and get practical knowledge of some machine learning approaches. Suppose you're looking for a more profound treatment of many of the techniques presented here, such as their mathematical foundations or more detailed considerations in the use of the algorithms. In that case, you are best served by consulting more advanced texts. This book is not meant to explain the origins or characteristics of each method thoroughly. Instead, at the heart of the book is a series of exercises and real-life case studies, putting each technique or tool to work in different business

situations. We leave it for other authors and other texts to present the theoretical and explanatory understanding of the tools. A significant contribution of this book is a curated database of business data files that should provide plenty of practice to acquire skills in each of the techniques presented. The exercises and cases in each chapter are presented with step-by-step explanations to help you acquire skills in their use.

Chapter 1 introduces data mining and data analytics and their use in business decision-making. It covers how and why data mining and analytics are used in a business context, and the information needs to drive the use of advanced data analysis tools, and the formalism of the data mining process. It puts in context data mining, machine learning, data analysis, data analytics, decision science, and data science. We introduce in this first chapter the roles of data analysts and differentiates them from data scientists. With the rise in data science development, we now have many remarkable techniques and tools to extend data analysis from a simple summary of numeric data or tabulation categorical data to answer sophisticated questions with numerical models. We discuss when data mining is appropriate and when it is not.

In Chapter 2, we cover the data mining process and introduce and develop an understanding of the CRISP-DM process. We introduce the role of machine learning in the data mining process and discuss the difference between data mining and machine learning. Knowledge Discovery in Databases (KDD), a parallel construct, and its relationship to DM are introduced. Case studies on the application of CRISP-DM in business. We discuss the context and the information need that drives data mining, and we set up the next chapter on framing questions as the critical end point to this activity.

Chapter 3 introduces the concept of the business need and the context that drove those needs. Chapter 3 is the first heavy lift for the data analyst: converting the nebulous information requests stemming from the information needs of management and co-workers into crisp, well-formed analytical questions that yield to analytical techniques. That is the heart of data analysis: producing well-formed questions that yield the analysis tools. The rest of the work is finding the data and applying the tools. Framing questions requires insight and creativity.

Once questions have been framed, we look for data to answer the questions Chapter 4 deals with acquisition and preparation. It must be pointed out that often we have the data, which is being collected in on-line transaction processing systems (OLTP) and in summarized data repositories, such as data warehouses (OLAP), from which we extract the required data using SQL queries. We often already possess the data set, but we need to be judicious about which part of the data to prepare for our analysis to answer framed questions. We often need to manipulate the data further to make it ready for the application of our algorithms. Chapter 5 shows you how to explore the data and what is termed descriptive statistics. We use summarization and tabulation tools to get a sense of the data and each variable and answer simple descriptive questions: how many, how much, when, what are the averages, and obtain simple cross-comparisons.

Chapter 6 introduces the concept of a model and shows how data models are built. It reviews the many steps in using algorithms to make mathematical expressions into valuable tools for prediction. We lay the foundation here for the benefit of various computational

algorithms presented in the subsequent seven chapters. We differentiate between machines, machine learning, data model making, supervised and unsupervised learning, and train and test data sets as major model-making concepts. A well-formed model based on answering well-framed questions is the product of the data mining process and the basis for ongoing business decision-making.

Using the CRISP-DM data mining standard, we use the early chapters present in conducting the preparatory steps in data mining: translating business information needs into framed analytical questions and data preparation. Chapter 1 gives plenty of practice of framing analytical questions applied to text data. Chapter 2 briefly covers the most common tools for data preparation and data mining. Chapter 3 explores where text data may be found in business databases and situations and the forms it might take. Chapter 4 covers data preparation and shaping the data set for analysis.

The following seven chapters cover the principal analytical approaches to data mining; the tools. In Chapter 7, we cover prediction algorithms using regression, linear regression and logistic regression. We cover classification in Chapter 8, using decision trees, Random Forests, and Naïve Bays algorithms. Chapter 9 covers the unsupervised techniques of clustering, such as hierarchical clustering and k-means clustering. Chapter 10 extends Chapter 7 on regression to produce trends and forecasts for time series data. Chapter 11 introduces the advanced technique of feature selection using variable reduction algorithms. We apply it with great success to various business situations. We also present tools to discover anomalies in our data in Chapter 12. We define an outlier in two ways: the three-sigma approach and Tukey fences using quartiles. Lastly, we tackle text data analysis in Chapter 13 with various techniques: term frequency analysis, keyword analysis, visualizing text using word clouds, and text similarity scoring.

Chapter 14 is the complement of Chapter 6. We defined modeling and reviewed the steps in model making in Chapter 6. Now, in Chapter 14, we finish modeling by presenting model validation techniques, the use of train and test data sets to check for overfitting, and how to correct for variable confusion by checking for collinearity. Chapter 15 helps with big data files by sampling to extract a representative smaller set of data for preliminary analysis or using tools (like Excel) that are limited to the data set's size.

ON THE COMPANION FILES

The exercises require access to the data sets used in analyzing the cases. They may be accessed on the companion disc or the companion files by writing to the publisher at info@merclearning.com. A file folder, *Case Data*, has all the files referenced in the case studies at the end of each chapter. The data files for the exercises for each chapter are in chapter folders. These can be downloaded and made available on a local drive. The solution folders within the chapter folder contain some illustrative charts and tables as well as solution files.

Dr. Andres Fortino
January 2023

ACKNOWLEDGMENTS

This book was a personal journey of discovery reinforcing early life lessons as a numerical methods scientist working on engineering problems and later applying the emerging field of machine learning to business. I worked as a translator for my students so they could also master the field of data mining. I had a great deal of help from my students, for which I am grateful, some directly in writing many of the exercises.

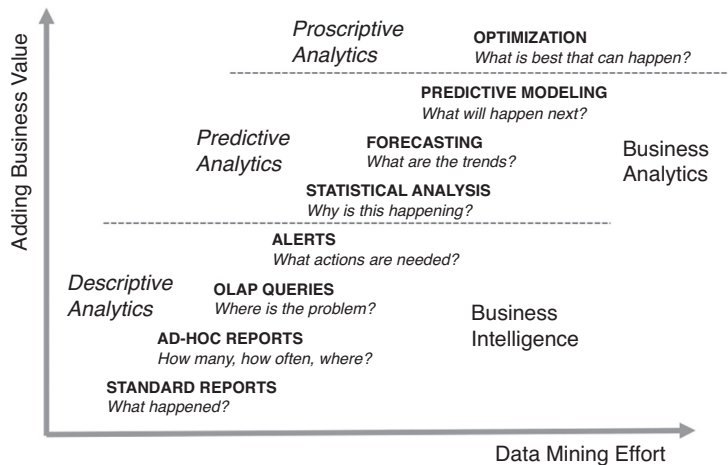
I am most grateful to Mr. Andy Sheng, for his staunch support and assistance. He worked right alongside me in developing the exercises for the book and became an integral part of the finished product you see before you.

I am also grateful to many other students who collaborated with me in exploring data mining and co-authored many papers in the area, some award-winning, stemming from their capstone research projects. I want to thank my graduate students at the NYU School of Professional Studies, and the many American Management Association professionals who attended my AMA seminars, and with whom I shared these techniques.

The entire team of editors and artists at Mercury Learning was terrific. They have my gratitude. A special thanks to Jim Walsh, my editor, who kept asking for more and helped shape an excellent book, and to Martha Zaytoun for her thorough editing of the manuscript.

Finally, I wish to acknowledge my loving and patient wife, Kathleen. This book was partly written during the worldwide COVID pandemic tragedy. It was stressful to have all that time indoors during lockdown, but it assisted tremendously in finishing the book. Having Kathleen by my side with her infinite patience and constant encouragement helped me survive the pandemic and complete this book in peace.

DATA MINING AND BUSINESS



Data mining, in simple terms, is finding useful patterns in a data set. Knowledge discovery in databases (KDD) is an earlier and perhaps more instructive expression for this activity. KDD means a sophisticated, non-trivial process of identifying valid, novel, potentially useful, and understandable patterns or relationships in data to make crucial business decisions. Those decisions need to be made in support of many human activities. In our case, we will apply this process and technique to business decision-making. Data mining differs from many other data analysis situations. It is more sophisticated and involves very complex algorithms. It goes beyond the basic forms of data analysis that provide data summaries and basic data understanding. More straightforward data analysis activities might include descriptive statistics, exploratory data visualization, dimensional slicing, hypothesis testing, and even queries from a database.

This chapter uses data mining techniques to explore answers to analytical questions beyond tabulating how many, what is the average, and what are the totals, the frequency of categories,

or even the crosstabulation of categories. Data mining helps us to discover why something happened, what are the outliers, how to cluster customers into meaningful groups, and how to predict possible future values of action.

DATA MINING ALGORITHMS AND ACTIVITIES

We may also use our tools to categorize business decision problems that need to be tackled with data mining techniques. For simple data analysis, we might use spreadsheet tools, such as Microsoft Excel. As powerful as spreadsheets are, they fall short of easily managing data mining problems. For these, we need to invoke algorithms such as regression or clustering, which require analytical programming tools such as Python and R, or commercial tools such as RapidMiner, SAS, or SPSS. Figure 1.1 shows some examples of data mining approaches, algorithms, and applications.

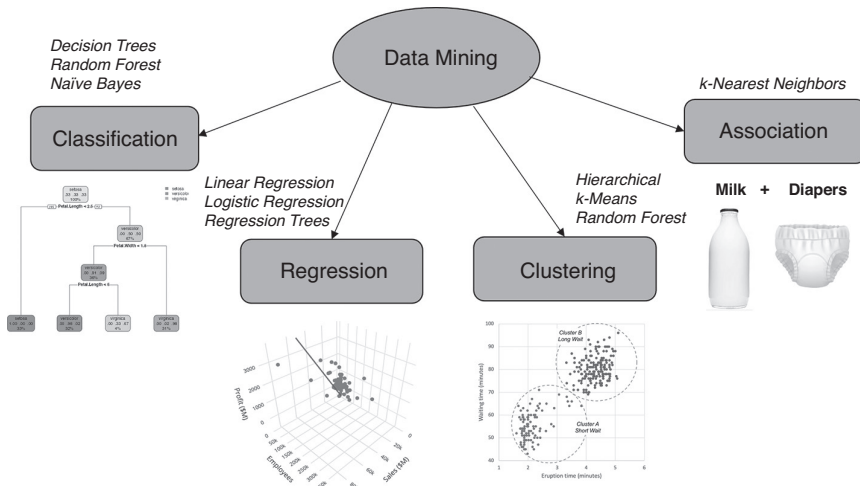


FIGURE 1.1 Some principal examples of data mining approaches, algorithms, and applications.

At the end of this chapter, we discuss cases where these activities are put to good use in answering business questions.

DATA IS THE NEW OIL

In today’s business environment, we often hear “Data is the new oil,” a phrase coined by Clive Humby in 2006 (Humby 2006). It is a useful metaphor that underscores the need for management to embrace data-driven decision-making. For us, it is an appropriate metaphor for the process of distilling data into knowing what to do, or what business actions to take. Let’s see what that means for the analyst. The elements of the metaphor and their equivalencies are summarized in Figure 1.2.

The raw material is crude oil; in business, the raw material is data. Just like oil, data does not provide any significant benefit in and of itself. It must be processed to be used. The oil must be extracted from the surrounding environment (rocks and soil) and collected, transported, and




	Oil Industry	Business	Knowledge Management
	Raw Material: Crude Oil	Data	Raw data is collected in databases and flat files. ERPs, CRM, RDBMS data warehouses, Excel spreadsheets.
	Distillates: Gasoline	Information	Through analysis data is converted to information information, a collection of facts. We draw conclusions. At the end of the process we are informed.
	Conversion to energy: Gasoline Engine	Knowledge	The facts become the basis for decision making: "Data driven decision making". Convert facts to decisions. Now we know what to do, what action to take.

FIGURE 1.2 “Data is the new oil.”

stored. It is the same with data. It must be cleaned, shaped, and adequately stored before applying analytical tools to extract useful information.

The raw material is most useful when distilled into byproducts that can be readily consumed and easily converted to energy. Thus, we refine various products from natural oil: crude oil, gasoline, kerosene, and other valuable distillates, like benzene. Data must also be distilled to yield useful information products. The data distillation process is called *data analysis*. Some analysis processes are straightforward descriptive statistical summaries using pivot tables and histograms. Others are more elaborate and refined analyses, such as predictive analytic products, which require sophisticated techniques such as decision trees or clustering. In the end, applying analysis to data yields information that we distill into facts and summarize into conclusions.

Oil distillates by themselves do not generally produce useful work. They can be burned to produce heat (a home-heating furnace) and light (a kerosene lamp). However, the most useful conversion process is a gasoline-burning engine, which generates mechanical power. Either way, we need a mechanism to transform oil distillates into work. It is the same with information distilled from data. It is nice to know the facts, but when they are converted into action, they become very powerful. In the case of business, it is the organization’s decision-making engine doing the conversion. Whether utilized by a single executive, a manager, or a committee, there is a business decision-making process that consumes information produced by analysts and generates decisions useful to the business. Information processed by the analyst-informed decision-making organizational engine translates into *knowing what to do*. Analysts are the transformers or distillers of data through their analysis process and the generation of facts and conclusions. They feed the decision-making engine of the organization, managers, and executives responsible for taking action.

DATA-DRIVEN DECISION-MAKING

Data, upon analysis, becomes information (*we become informed*), which then becomes the basis of knowledge (*knowing what to do*). As data analysts, it is our task to convert data into information and offer the resulting facts to our business colleagues for decision-making. Figure 1.3 describes this process in detail.

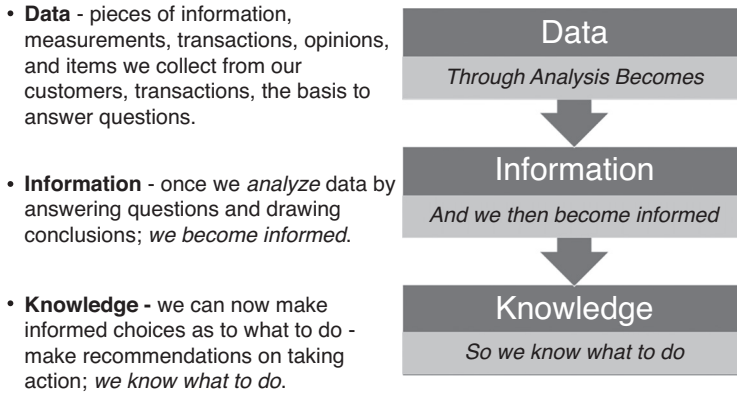


FIGURE 1.3 The data-driven decision-making process.

BUSINESS ANALYTICS AND BUSINESS INTELLIGENCE

In this field, as in many others, numerous technical terms are used. When the same term is used for different—and sometimes overlapping—meanings, it can lead to confusion. We will try to clarify the differences between a few of these widely used expressions.

We will refer to business intelligence as an analysis that would require calculating descriptive results. It explains what has occurred, gives the level of quantities, and reports on the flows of items and people, as well as money in and out of categories (accounts). It helps us report on the past. It may even be set up to trigger alerts when certain situations occur. For example, it could alert us when fund levels in bank accounts are low, inventory levels need replenishment, personnel turnover rates rise above a pre-set level, or expense levels rise above pre-set budget amounts.

Usually, sophisticated Enterprise Resource Management (ERP) information systems, such as Oracle Financials, or Customer Relationship Management systems (CRMs), such as Salesforce, make available reporting tools to produce these data-based reports. They are referred to as BI tools. We do not consider this to be data mining, but they are excellent tools to extract data from corporate transactional databases for data mining. We use simple summarization tools such as tabulation and aggregation. In general, we call this analysis and reporting effort *data analysis*.

These tools are not to be confused with more sophisticated approaches that use the processes and algorithms of data mining, which we may call *business analytics*. In business analytics, we go beyond business intelligence to answer more insightful questions, such as “Why did that happen?,” “What is the trend?,” or “What will happen next (prediction)?” The use of algorithms exemplifies more sophistication and greater care and requires considerable knowledge of additional tools and approaches. Some of the tools required include IBM SPSS, SAS Enterprise Guide, and Rapid Miner. To carry out these processes, data must first be extracted from the corporate databases (often using the Structured Query Language (SQL) language) to which the algorithms discussed above can then be applied. We term this effort *data analytics* to differentiate it from the simpler summarization efforts of data analysis.

The ability to perform data analysis is becoming the norm for most business staff. They are expected to be able to work with analysis tools such as Excel and provide cogent summaries and data-driven evidence to back up their regular reporting. However, data analytics requires greater effort and expertise, so the ability to execute it is at a premium among corporate office workers. Data analytics, or business analytics, yields greater value than the less sophisticated data analysis, or business intelligence, as we show in Figure 1.4.

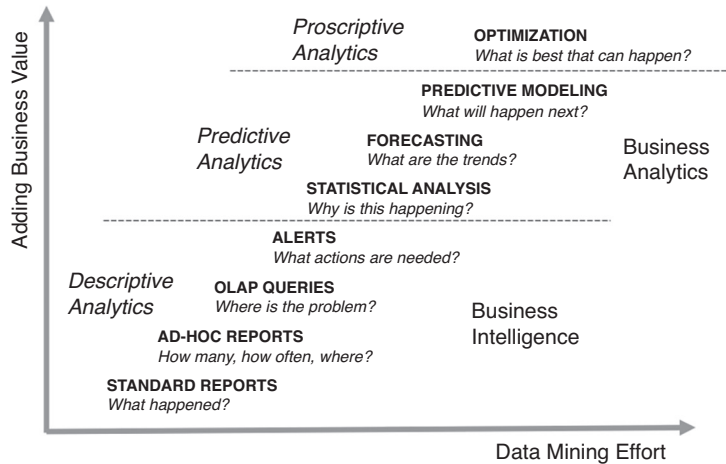


FIGURE 1.4 The difference between business analytics and business intelligence.

It is fair to say that the BI tools, CRM, and ERP systems, which currently offer only the simpler data analysis capability, will begin sporting some data mining tools and evolve in their data analytic capabilities. Thus, business intelligence will eventually also include business analytics.

ALGORITHMIC TECHNOLOGIES ASSOCIATED WITH DATA MINING

Data analytics, or data mining, encompasses the use of many sophisticated algorithms. They frequently have been developed as academic exercises or for other fields. Figure 1.5 shows eight major classifications of such algorithms, with some of the most prominent examples of each. We will be taking up each of the algorithms in turn. These algorithms are computer-based and have mostly been developed by computer scientists for a variety of fields. Often scientists, such as astronomers (who need to analyze large amounts of data), develop advances in data analysis in collaboration with computer scientists. Likewise, statisticians who need to create a sophisticated analysis of epidemiological data from clinical trials or accurate human lifespan models under various conditions to price insurance products have contributed to developing data analysis processes. We call the latter type of data analyst an *actuary*. We derive many of the sophisticated computer tools we use today to do business analytics from these early workers. The whole field has evolved into a thriving industry that we call *data science*: the development of ever-more sophisticated data mining algorithms.

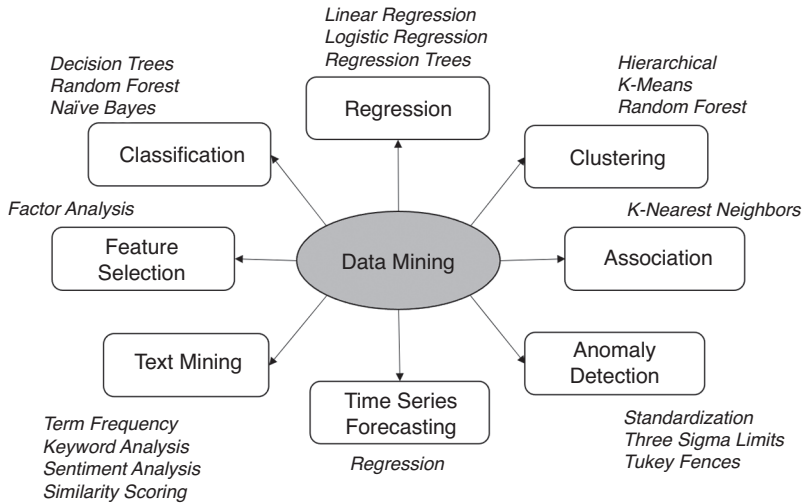


FIGURE 1.5 Algorithmic technologies associated with data mining activities.

DATA MINING AND DATA WAREHOUSING

Another technology closely associated with data mining is *data warehousing*. As discussed previously, corporate data is stored and processed using commercial systems such as ERP and CRM. These tools allow an organization to capture transaction data (sales, invoices, payments, customer orders, inventory transfers, financial transactions, and personnel changes) in an organized and secure manner. The database format of choice to organize such data is a Relational Database Management System or RDBMS. Interactions with such systems daily occur with applications set up for such purposes (Accounts Payable, Payroll, and Inventory Control.) These systems also have a Structured Query Language (SQL) that is used to set up, manage, and extract information for such systems. We will need to use SQL to run queries to extract the data we will need for data mining.

These systems are optimized for transaction processing. The data analysis reporting tools embedded in such systems are optimized for straightforward business intelligence analysis. Routine daily operation questions are easily answered with such systems. Obtaining answers to more strategic questions with these BI tools is often very difficult and requires a great deal of effort. Very often, these require data mining approaches. We have a technique to make answering strategic questions from RDBMS databases easier: *data warehousing*.

Once we know the types of strategic questions we wish to pose and to answer regularly, we set up an analysis system where we pose queries to the operational database. Then, the extracted data is uploaded to this new data structure for easy querying to support answering the strategic questions. The database structure of transactional systems is often complex and sophisticated, using many interrelated tables to support complicated application systems for transaction processing. The data warehouse structure is much simpler and easier to query and to extract tables for data mining. The data warehouse data is loaded from the transactional systems regularly to keep it up to date. Queries for extracting and loading are written once and subsequently run many times to create the foundation for data mining activities.

To summarize, data mining can be run from data extracted from operational and transactional systems, but these are often ad hoc (one-time only) queries. Repeated queries that require data analytics are run from a data warehouse set up for such data mining activities. Thus, data warehouses are set up for the express purpose of performing business analytic activities using data mining.

CASE STUDY 1.1: BUSINESS APPLICATIONS OF DATA MINING

Below are some examples of data mining tasks, the algorithms that may be used to implement them, and the types of questions data mining would answer in contrast to the simpler descriptive questions answered by business intelligence activities.

Case A - Classification

Description of Data Mining Activity

Predict if a data point belongs to one of the pre-defined classes. The prediction will be based on learning from a known data set.

Data Mining Algorithms

Decision trees, neural networks, Random Forest algorithm, Bayesian net models, induction rules, and K-nearest neighbors.

Data Mining Analysis Framed Questions

What voters belong to which buckets by a political party from known voting demographic data?

Descriptive Analysis Framed Questions

What was the vote count by party and geographic location?

Case B - Regression

Description of Data Mining Activity

Predict the numeric target label of a data point. The prediction will be based on learning from an unknown data set.

Data Mining Algorithms

Linear regression and logistic regression.

Data Mining Analysis Framed Questions

What is the unemployment rate for next year? What should be the insurance premiums for a particular insurance instrument?

Descriptive Analysis Framed Questions

How many people were unemployed last year by demographic characteristics? How much money did we receive in revenue from the sale of particular insurance instruments?

Case C - Anomaly Detection

Description of Data Mining Activity

Predict if a data point is an outlier compared to other data points in a data set.

Data Mining Algorithms

Distance-based, density-based, local outlier factors.

Data Mining Analysis Framed Question

Which credit card transactions were fraudulent in the past month?

Descriptive Analysis Framed Question

What was the total amount of credit card charges processed last month?

Case D – Time Series

Description of Data Mining Activity

Predict the value of the target variable for a future timeframe based on historical matters.

Data Mining Algorithms

Exponential smoothing, autoregression integrated moving average (ARIMA), and regression.

Data Mining Framed Questions

What will be next month, next quarter, and next year's sales? What will be a production forecast for the next quarter?

Descriptive Analysis Framed Questions

What were sales for the past month, the past quarter, and the past year? How many items did we produce last quarter, last month, and for all of last year?

Case E – Clustering

Description of Data Mining Activity

Identify natural clusters with a data set based on inherent properties within the data set.

Data Mining Algorithms

k-means clustering, density-based clustering, and random forest clustering.

Data Mining Analysis

Using customer information based on the transactions, web activity, and customer call data segment; the customer database for database marketing purposes.

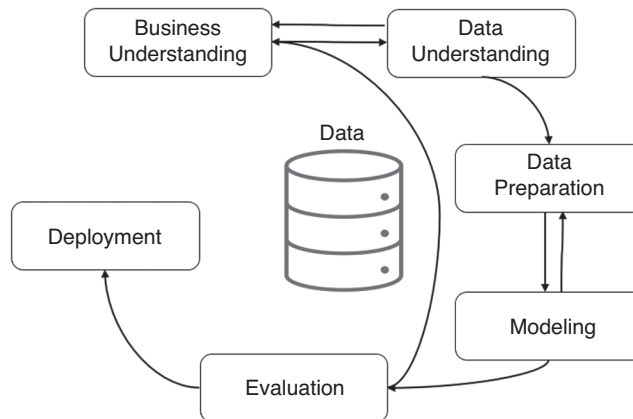
Descriptive Analysis

How many calls were made by call type, geographic location, and lifetime value of the customer?

REFERENCE

Humby, Clive. 2006. "Data is the new oil." Proc. ANA Sr. Marketer's Summit. Evanston, IL, USA (2006).

THE DATA MINING PROCESS



As with any practical intellectual activity that yields business results, using a well-thought-out process delivers the best results. In the case of data mining, there are two activities, one embedded within the other. The first is the primary process — or overarching activity — of analyzing a business analytics problem or need for information and organizing the step-by-step actions needed to solve the problem and satisfy that need. That activity we call the *data mining process*. The second activity of performing data analysis within the process produces a data model to yield business benefits. We call that *modeling*. Generally, the data mining process adapts general project management techniques that are typically applied to analyzing data and answering business information questions.

This chapter presents the various choices in adopting a process model for data mining and discusses some options for analytics languages, comparing one to the other. We also analyze current trends to assist with analytics technology adoption decisions. Ultimately, it will become clear that the only viable data mining process today is the CRISP-DM process model. In general, for

business data modeling, the choice of analytics languages will come down to using either the R language or Python. It must also include both Excel for simple analysis and Structured Query Language (SQL) for querying.

DATA MINING AS A PROCESS

An effective data mining process should support all the steps, from project initiation to model deployment. The process looks very similar to the software development process, as most of the activities deal either with managing data sets or deploying software analysis programs. The hardware technology is essential but secondary, since most practical analysis tools run on most platforms under most operating systems. Extensive data sets (Big Data) need specialized support systems, such as distributed computing (computing clusters). In this book, we primarily focus on the analysis of data sets that are considered smaller than Big Data.

In general, as with any technical problem, there are four phases in the data mining process: exploration, analysis, interpretation, and exploitation.

Exploration

Exploration involves the understanding of the business environment and the business need for information (in other words, the formulation of the business case for the analysis). It also requires the analyst to convert the business questions into well-formulated or well-framed analytical and directly computable questions. The analyst also explores, acquires, and prepares the necessary data sets to be analyzed or mined for answers during this stage.

Analysis

There are two stages to the analysis step. The first step involves a thorough understanding of the data elements. Each variable is analyzed with basic statistical methods to understand its characteristics. Sometimes this leads to a realization that the data set is not complete, necessitating further data gathering to ensure there is a complete set to begin the data mining analysis. The second activity is data mining: the use of sophisticated analytic methods to extract answers from the data set. This activity involves the pursuit of answers to well-framed questions through analytical methods. This goes beyond fundamental statistical analysis and often requires the use of advanced techniques, such as machine learning. Here we are looking for deeper meanings and patterns from our data rather than a surface-level understanding. Often, we are seeking to understand why things happened, rather than merely tabulating what occurred. This step can be characterized as a knowledge discovery journey. The endpoint of this phase is either a set of deeply insightful answers to the ad hoc questions posed by the business executives driving the project or the construction of a sophisticated data model to be used in the future as the data is updated and the questions need to be answered repeatedly.

Interpretation

This phase is more of a logical rather than a technical step. It requires business judgment as well as an understanding of the limitations of the models constructed. It seeks meaningful conclusions based on the facts obtained from analysis as impartial input to the business decision-making process. It is the basis for the “data-driven decision-making” results which is much prized in today’s business environment. The analysis step provides answers which are encapsulated into

what we might term *facts*. Then, the interpretation step completes the information gathering activity by deducing *conclusions* derived from those facts.

Exploitation

This last phase involves the institutionalizing of the data-mining efforts, meaning any answers obtained and conclusions derived are then turned over to the business managers and executives needing to make data-driven *decisions*. The data-mining efforts often result in helpful data models that can be used repeatedly to support an ongoing decision-making cycle. To preserve the investment in data mining efforts, more work is needed beyond the initial analysis. Models often require suitable user interfaces to be used by novice users, not just experienced analysts. Models and data sets also require recalibration for suitability to the questions being asked and changing economic, political, legal, and business conditions. In other words, the final phase ensures that we produce and maintain a business tool usable by all who need it in this final phase.

In this book, we concentrate on the first two phases, exploration and analysis, which are the more technical aspects of the process. We leave the more advanced topics of interpretation and exploitation to other texts.

SELECTING A DATA MINING PROCESS

There are some viable alternatives of adoption for a data mining process. Figure 2.1 shows the more popular processes (Chapman 2000). Before many of these models were defined and in use, organizations created their own, making ad hoc models popular. Notice the rise in popularity of the ad hoc models before the diffusion and final dominance of the CRISP-DM model. With the advent of standardization, several process models emerged, namely CRISP-DM, KDD, and SEMMA. CRISP-DM took an early lead, which it held from 2007–2020. The other two leading contenders, KDD and SEMMA, have all but disappeared from use. More recently, Microsoft has proposed its own model, TSDP, which may challenge the leader in the future.

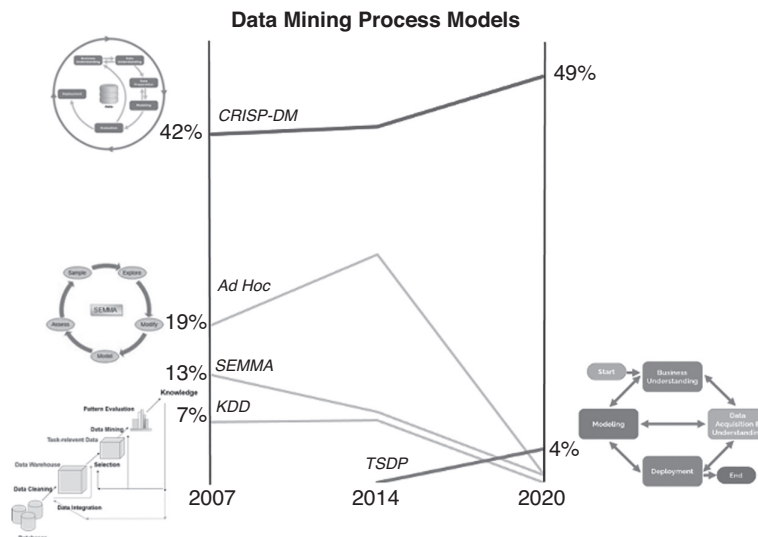


FIGURE 2.1 Trends in data mining process models. CRISP-DM continues to hold the lead. Data from a Data Science Process Alliance poll (Saltz 2022).

The most popular process models are described by Saltz (2022):

- **CRISP-DM:** The Cross Industry Standard Process for Data Mining (CRISP-DM) is a process model with six phases (business understanding, data understanding, data preparation, modeling, evaluation, and deployment) that naturally describes the data science life cycle. Published in 1999 to standardize data mining processes across industries, it has become the most common methodology for data mining, analytics, and data science projects.
- **KDD:** Knowledge Discovery in Database (KDD) is the general process of discovering knowledge in data through data mining or extracting patterns and information from large data sets using machine learning, statistics, and database systems. It dates back to 1989; it represents the overall process of collecting data and methodically refining it. KDD presents a complete data lifecycle process, with data mining as one step in that process.
- **SEMMA:** Developed by SAS, SEMMA defines five phases of a project (Sample, Explore, Modify, Model, and Assess). Although designed to help guide users through tools in SAS Enterprise Miner for data mining, SEMMA is often considered a general data mining methodology. SEMMA is promoted as an organized, functional toolset, as claimed by SAS, its developer, and is associated with its SAS Enterprise Miner platform.
- **TDSP:** (Team Data Science Process): Launched by Microsoft in 2016, TDSP defines five stages of the data science life cycle (Business Understanding, Data Acquisition & Understanding, Modeling, Deployment, and Customer Acceptance). It combines aspects of Scrum and CRISP-DM into a data science life cycle that incorporates the team aspect of executing data projects.

CRISP-DM is a serviceable, popular model and worthy of adoption. Let's now explore it in more detail.

THE CRISP-DM PROCESS MODEL

The CRISP-DM (Cross Industry Standard Process for Data Mining) reference model is a useful and practical process for any data mining project. The model was developed by the CRISP-DM consortium (Chapman 2000). The first step in the process is to ascertain and document a business understanding of the problem to be analyzed. Wirth and Hipp (2000), two of the project originators, summarized the method as follows: "This initial phase focuses on understanding the project objectives and requirements from a business perspective and then converting this knowledge into a data mining problem definition, and a preliminary project plan designed to achieve the objectives." Figure 2.2 shows the six steps and their relationship to each other.

Business Understanding

This initial phase focuses on understanding the project objectives and requirements from a business perspective, then converting this knowledge into a data mining problem definition and a preliminary plan designed to achieve the objectives.

Data Understanding

The data understanding phase starts with initial data collection and proceeds with activities that enable the analyst to become familiar with the data, identify data quality problems, discover first insights into the data, and/or detect interesting subsets to form hypotheses regarding hidden information.

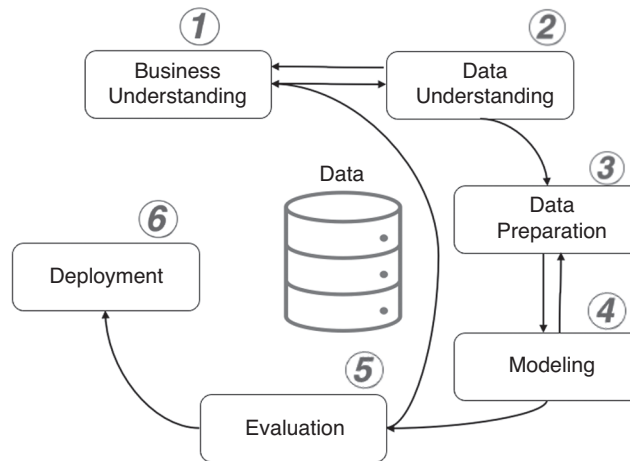


FIGURE 2.2 The six-step CRISP-DM process.

Data Preparation

The data preparation phase covers all activities needed to construct the final data set (data that will be fed into the modeling tool(s)) from the initial raw data. Data preparation tasks are likely performed multiple times and not in any prescribed order. Tasks include the selection of tables, records, and attributes, as well as the transformation and cleaning of data for modeling tools.

Modeling

Various modeling techniques are selected and applied in this phase, and their parameters are calibrated to optimal values. Typically, there are several techniques for the same data mining problem type. Some methods have specific requirements regarding the form of the data. Therefore, it is often necessary to return to the data preparation phase.

Evaluation

At this stage in the project, the analyst has built a model (or models) of high quality from a data analysis perspective. Before proceeding to the final deployment of the model, it is essential to thoroughly evaluate it and review the steps used to create it to ensure that the model properly achieves the business objectives. An important aim is to determine whether there is some critical business issue that has not been sufficiently considered. At the end of this phase, a decision on the use of the data mining results should be reached.

Deployment

The creation of the model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that the customer can use it. This often involves applying “live” models within

an organization’s decision-making processes, for example, real-time personalization of web pages or repeated scoring of marketing databases. Depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process across the enterprise. In many cases, it is the customer, not the data analyst, who carries out the deployment steps. Even if the analyst will carry out the deployment effort, it is important for the customer to understand upfront what actions need to be carried out to actually make use of the created models.

Figure 2.3 presents an outline of phases accompanied by generic tasks (in bold) and outputs (in italics) associated with each step. This summary follows the CRISP-DM standard. In the following chapters, we describe each generic task and its outputs in more detail. We focus our attention on task overviews and summaries of outputs.

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives <i>Background</i> <i>Business Objectives</i> <i>Business Success</i> <i>Criteria</i>	Collect Initial Data <i>Initial Data Collection Report</i>	Select Data <i>Rationale for Inclusion/Exclusion</i>	Select Modeling Techniques <i>Modeling Technique</i> <i>Modeling</i> <i>Assumptions</i>	Evaluate Results <i>Assessment of Data</i> <i>Mining Results w.r.t. Business Success</i> <i>Criteria</i> <i>Approved Models</i>	Plan Deployment <i>Deployment Plan</i>
Assess Situation <i>Inventory of Resources</i> <i>Requirements,</i> <i>Assumptions, and Constraints</i> <i>Risks and Contingencies</i> <i>Terminology</i> <i>Costs and Benefits</i>	Describe Data <i>Data Description Report</i>	Clean Data <i>Data Cleaning Report</i>	Generate Test Design <i>Test Design</i>	Review Process <i>Review of Process</i>	Plan Monitoring and Maintenance <i>Monitoring and Maintenance Plan</i>
Determine Data Mining Goals <i>Data Mining Goals</i> <i>Data Mining Success</i> <i>Criteria</i>	Explore Data <i>Data Exploration Report</i>	Construct Data <i>Derived Attributes</i> <i>Generated Records</i>	Build Model <i>Parameter Settings</i> <i>Models</i> <i>Model Descriptions</i>	Determine Next Steps <i>List of Possible Actions</i> <i>Decision</i>	Produce Final Report <i>Final Report</i> <i>Final Presentation</i>
Produce Project Plan <i>Project Plan</i> <i>Initial Assessment of Tools and Techniques</i>	Verify Data Quality <i>Data Quality Report</i>	Integrate Data <i>Merged Data</i>	Assess Model <i>Model Assessment</i> <i>Revised Parameter Settings</i>	Review Project <i>Experience</i> <i>Documentation</i>	
		Format Data <i>Reformatted Data</i> <i>Dataset</i> <i>Dataset Description</i>			

FIGURE 2.3 Generic tasks (bold) and outputs (italic) of the CRISP-DM reference model (Chapman 2000).

SELECTING DATA ANALYTICS LANGUAGES

In step 4 of the the CRISP-DM model, the modeling step, we are involved in concrete data analysis. In some cases, we are deploying sophisticated machine learning models. It is essential to understand what decisions an analyst must make in choosing analytics platforms or computer programming languages. Again, as with process models, there are many alternatives. The only viable options for adoptions are few. The most basic choice, and one the analyst should not overlook, is Excel. However, for sophisticated model building, as must be the case here, we need to go beyond Excel. For small data sets (up to maybe 100 MB or one million rows of data), Excel is a handy tool, especially for the quick preparation of data sets for analysis. Excel has a few very helpful and quick analysis tools, such as pivot tables. However, it soon runs out of computing power. Excel should not be overlooked as a data visualization tool to create charts and graphs to communicate the analysis results. While some workers prefer a more sophisticated data visualization tool for that purpose, like Tableau, Excel is easy to work with and has many chart forms that can convey information compellingly.

The analyst also must contend with data sets that need to be extracted from data warehouses and commercial application programs based on an RDBMS (Relational Database Management System), such as Oracle. For that purpose, familiarity with SQL is required. We recommend that SQL be used just for extraction. Do not attempt to do any processing or analysis of the data in SQL. There are more powerful tools for that purpose, as we will show.

After extracting the data set with SQL and preparing it with Excel, most of the analysis and model building is left to one or both popular analytics languages: R and Python. Figure 2.4 shows the popularity of these languages with analysts over time (Piatetsky 2019). With R and Python, it is not so much that the analyst must choose between them, but that both tools should be in the toolset. Each tool has its place. R is quick and easy to program and deploy for simple machine learning models. For more sophisticated models and production systems, the industry tends to rely on Python. Some analysts report Python is harder to master and work with, so unless you need the sophistication of the Python library of routines, you may want to start your project with R.

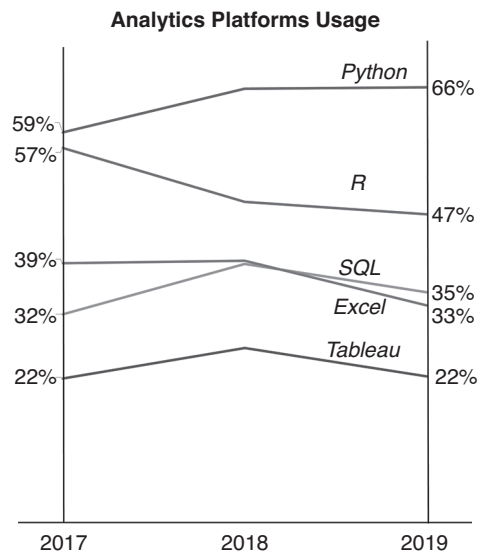


FIGURE 2.4 Trends in the more common analytics programming languages and data visualization tools, popular with analysts and data scientists (Piatetsky 2019).

THE CHOICES FOR LANGUAGES

- **Python** is currently the most popular data science programming language. It is easy to use and relatively easy to learn. Python provides all the necessary libraries for the four significant steps of dealing with data: data collection and cleaning, data exploration, data modeling, and data visualization. Python also has many advanced deep-learning libraries, which makes it the default language for artificial intelligence. Python's versatility makes it the most popular language for data science.
- **The R language** is another popular programming language for data science. R is highly extensible and easy to learn. It is excellent for statistical computing and graphics. R has a powerful scripting language, which means that it can handle large and complex data sets. It is compelling when performing statistical operations. R has many positive attributes, including being open-

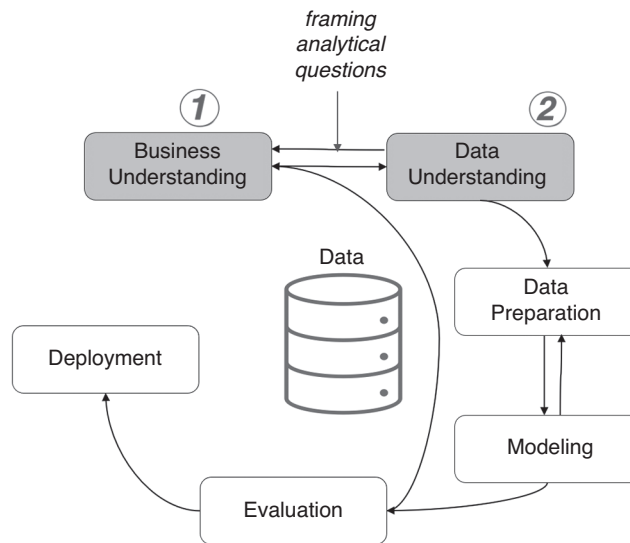
source (it garners a large amount of support from the programming community) and has multiple packages, quality plotting and graphing, and various machine learning operations. However, it has inadequate security, making it difficult to secure models that have been written in it.

- **SQL** is an essential language for analysts to master because it is often necessary to extract data from structured databases. SQL is the standard and most widely used programming language for relational databases. SQL is a non-procedural language, meaning that it does not require the use of traditional programming logic, which makes using SQL easier and does not require the analyst to be an expert coder.
- **Microsoft Excel** is a spreadsheet-based tool developed by Microsoft for Windows, macOS, Android, and iOS. It features calculation, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications (VBA). Excel forms part of the Microsoft Office suite of software. Excel is useful largely because of its ubiquity; most business computers have Excel installed on them, so it is readily available.
- **Tableau** is a robust analysis and data visualization tool used in the business intelligence industry. Tableau helps present the data that professionals can understand at any level in an organization. It also allows non-technical users to create customized dashboards. It is best known for data visualization, but it also has some powerful menu-driven analysis features.

REFERENCES

- Chapman, Pete. 2000. *CRISP-DM 1.0: Step-by-Step Data Mining Guide*. SPSS.
- Hotz, Nick. 2022. "The CRISP-DM Process Model." Data Process Alliance. August 8, 2022. <https://www.datascience-pm.com/crisp-dm-2/>.
- Piatetsky, Gregory. 2019. "Python Leads the 11 Top Data Science, Machine Learning Platforms: Trends and Analysis." KDnuggets. May 30, 2019. <https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html>.
- Saltz, Jeff. 2022. "CRISP-DM Is Still the Most Popular Framework for Executing Data Science Projects." Data Science Process Alliance. May 2, 2022. <https://www.datascience-pm.com/crisp-dm-still-most-popular/>.
- Wirth, Rudiger, and Jochen Hipp. 2000. "CRISP-DM: Towards a Standard Process Model for Data Mining." Essay. In *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, 29–40. 11-13 April 2000. Manchester.

FRAMING ANALYTICAL QUESTIONS



Any analytical efforts in support of a business must begin with the business’s purpose in mind. We use the word “business” here to mean all the operational and strategic activities of any organization used to manage itself, be it for-profit, non-profit, or governmental. This chapter presents the practical aspects of the preparatory processes needed to apply analytical tools to answer business questions. We start with the stated business’s informational needs, which drive the framing of the analytical problems. For the analysis to be effective, it is essential to do some homework first. An analysis of the context of the informational needs must first be conducted. Discovering the Key Performance Indicators (KPIs) driving the needs and current gaps in the performance of those indicators must motivate the analytical work. That way, we ensure we fulfill the immediate information requests of the organization while shedding light on the underlying KPI gaps.

The CRISP-DM (Cross Industry Standard Process for Data Mining) reference model is a useful and practical process for any data mining project, as we saw in Chapter 2. The model was developed by the CRISP-DM consortium (Chapman 2000). The first step in the process is to ascertain and document a business understanding of the problem to be analyzed. Wirth and Hipp (2000), two of the project originators, summarized the method as follows: “This initial phase focuses on understanding the project objectives and requirements from a business perspective and then converting this knowledge into a data mining problem definition, and a preliminary project plan designed to achieve the objectives.”

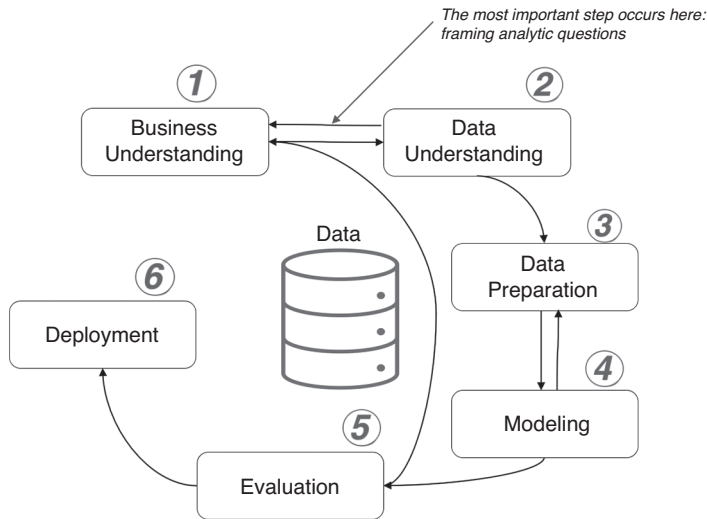


FIGURE 3.1 The CRISP-DM Model and framing analytical questions.

This chapter outlines a practical approach for the discovery of the business needs driving the analytics project. We understand that the critical step in this process is formulating well-framed analytical questions. The exercises provided in this chapter help the reader acquire the necessary skills to ensure that business needs drive their analytics projects.

HOW DOES CRISP-DM DEFINE THE BUSINESS AND DATA UNDERSTANDING STEP?

The CRISP-DM model definition defines these two steps as follows:

Business Understanding

This initial phase focuses on understanding the project objectives and requirements from a business perspective, then converting this knowledge into a data mining problem definition and a preliminary plan designed to achieve the objectives.

Data Understanding

“The data understanding phase starts with initial data collection and proceeds with activities that enable you to become familiar with the data, identify data quality problems, discover first insights into the data, and detect interesting subsets to form hypotheses regarding hidden information.

The early model definition is silent on the framing of analytical questions. However, a recent evolution of the model rightly acknowledges that framing analytical questions is an important step. Figure 3.2 shows the CRISP-DM defined tasks and activities at these two initial steps.

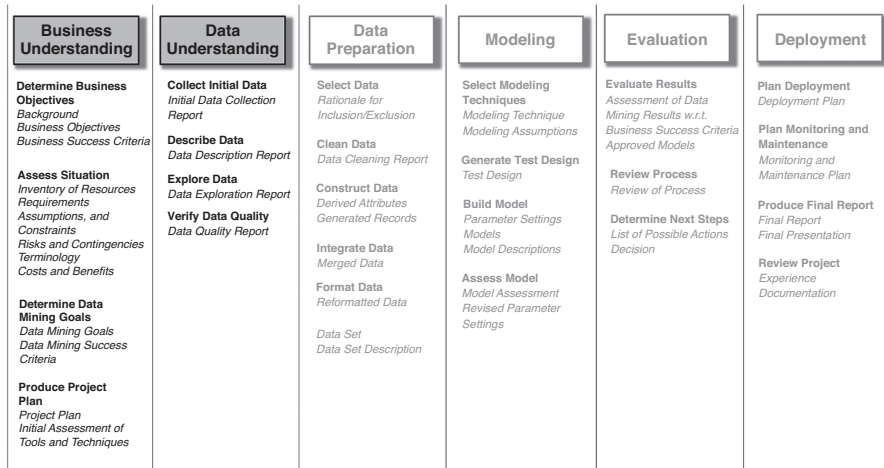


FIGURE 3.2 The CRISP-DM process model, with the business and data understanding deliverables and activities highlighted.

THE WORLD OF THE BUSINESS DATA ANALYST

Data analysis in a business context supports business decision-making. To be useful, data analysis must be driven by well-framed analytical questions. There is a well-developed process for creating well-framed questions. With their expertise in analysis, knowing what can be done, and knowing what the results might look like after analysis, the data analyst is the best-placed person to create computable analytical tasks based on information needs. Information needs are those questions formulated by business managers and staff who require facts to make decisions. Framed questions are produced by an analyst as they translate the information needs into computable queries. Figure 3.3 shows some of the steps followed by the business data analyst to present the results of their investigations.

Although the diagram shows the *business information need* following the *context* step, ascertaining the information need is usually the first step. An executive, a manager, or a fellow staff member approaches the business analyst with an information request to discover the answer to some pressing business issues. That request, called the *business information need*, is often expressed in nebulous terms: “Are we profitable this month?,” “Why do you think shipments have been late in the past six months?,” or “Are we over budget?”

The analyst cannot give an immediate answer because the questions are not posed in a manner in which they can be immediately computed. Thus, the analyst must translate the need into questions that can be used in the computation. These we term *framed analytical questions*.

In addition, it is the analyst’s responsibility to investigate the *business context* driving the information need. When answering the framed analytical questions, analysts must go beyond

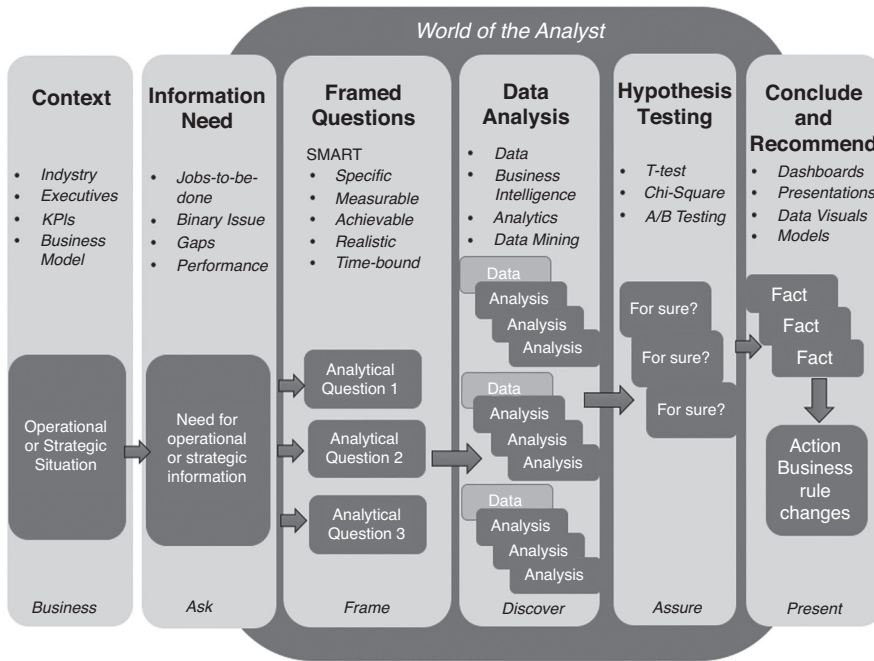


FIGURE 3.3 The world of the analyst: the process of business information needs analysis.

the immediate need and provide support for the underlying context driving the need. The context has to do with the industry the business is in, the business model the company is using, and the current status of the KPIs driving the management of the organization. The process of thinking through all the elements to arrive at the point of framing questions is presented rather well by Max Shron in his *Thinking with Data* book (Shron 2014). He presents the CoNVO model: (Co) context, (N) information need, and (V) vision are related to the solution (including the framed questions), and the (O) outcome. We use some of his observations and process model here.

HOW DOES DATA ANALYSIS RELATE TO BUSINESS DECISION-MAKING?

We answer framed analytical questions by applying analytical techniques to the data sets that we collect and shape. Applying the analysis to the data yields information: we, as analysts, become informed. At the end of our analysis process, we become subject matter experts (SMEs) on that business issue, the most informed person on the topic at the moment. We communicate our findings as facts and conclusions and perhaps venture some recommendations to our colleagues and managers. Using our findings, they are then in the best position to take action (make decisions): they know what is to be done. The data, upon analysis, becomes information (*we become informed*), which then becomes the basis of knowledge (*we know what to do*). As data analysts, it is our task to convert data into information and offer the resulting facts to our business colleagues for decision-making. Figure 3.4 describes this process in detail.

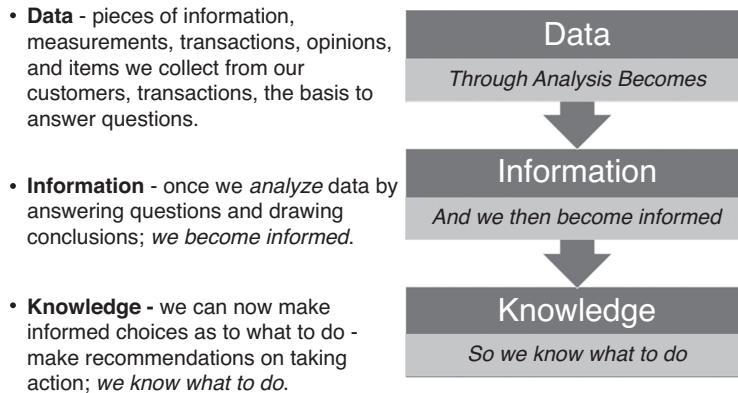


FIGURE 3.4 The data-driven decision-making process.

HOW DO WE FRAME ANALYTICAL QUESTIONS?

Translating nebulous, probably not well-formed, information needs into a computable, well-framed set of questions is a critical step for the analyst. One of the “raw materials” of the analysis process is the statement of the information need. It must be parsed and taken apart word-for-word to derive its actual meaning. From that parsing process comes a thorough understanding of what must be computed to bring back a good answer. In the parsing process, the analyst asks each element of the information request: “What does this mean?” The answers yield definition, clarity, and an understanding of what must be computed and the elements of the data that will need to be collected.

The parsing process brings an understanding of the major elements of the analysis:

- (a) What is the population (rows of our data table) that needs to be studied?
- (b) What variables or features of the population (columns) must populate the database to be collected?
- (c) What computations will be needed to use these variables (the framed questions)?

As the analyst begins to understand the meaning of the elements of the request, questions form in their mind that need to be answered in the analysis process. The quantitative questions (what, who, how much, and when) will yield to the analysis tools at the command of the analyst. These are questions that can be answered by tabulating categorical variables or applying mathematical tools to the numerical variables. At this point, depending on how versed the analyst is on machine learning and other data science tools, they can formulate an approach to answering the question with more sophisticated tools. These become the framed analytical questions.

At this stage, generating as many computable questions as possible yields the best results. In brainstorming, we perform a divergent exercise of collecting all possible questions that we wish to know. Then, before starting the analysis, we perform the convergent exercise of prioritizing the brainstormed questions, looking for the most important ones to be tackled first. It often happens that as the analysis work progresses, new vital framed questions are discovered

and may be added to the work. Therefore, the initial set of framed questions does not need to be complete. Even so, care must be taken to start with a reasonably good set of framed questions.

WHAT ARE THE CHARACTERISTICS OF WELL-FRAMED ANALYTICAL QUESTIONS?

Well-framed analytical questions exhibit the same characteristics we have come to associate with well-framed goals and objectives: they must be SMART. Generally, SMART goals and objectives are

- *Specific* Target a specific area for improvement or goal to be achieved.
- *Measurable* Quantify or at least suggest an indicator of progress toward that goal.
- *Assignable* Specify who will do it and who is involved.
- *Realistic* State what results can realistically be achieved, given the available resources.
- *Time-related* Specify when the result(s) can be achieved.

When applied to framing analytical questions, the concepts translate to the following (see Figure 3.5):

- *Specific* The framed question must be focused and detailed.
- *Measurable* The framed question must be computable.
- *Attainable* The framed question must be able to be answered by the techniques known to the analyst who will do the analysis.
- *Relevant* The answers to the framed question must apply to the business.
- *Time-related* Some elements of time should be considered in the analysis.

Information needs are often expressed in nebulous non-specific terms. Thus, information needs by their nature do not fit the SMART rules. Some information needs are specific and may be computed without further analysis. In general, additional specific framing is needed.

The best way to learn is to practice. In the following exercise, we go through the major steps outlined above using a classic situation: We want to know if the Law of the Sea was followed by the seamen on the Titanic in response to the sinking disaster. We take up the case of analyzing the Titanic disaster below.



FIGURE 3.5 SMART, well-framed analytical questions.

EXERCISE 3.1 – FRAMED QUESTIONS ABOUT THE TITANIC DISASTER

The Case

Imagine that you work for a famous newspaper. Your boss is the news editor of the newspaper. It is almost the 100th anniversary of the Titanic disaster. The editor assigned a reporter to cover the story. The reporter submitted an article that states, “the crew of the Titanic followed the Law of the Sea in responding to the disaster.” The editor is concerned that this may not be true and assigned you to fact-check this item. You decide to approach it from an analytic point of view. Your analysis of the assignment should yield the following:

The Information Need

Did the crew of the Titanic follow the Law of the Sea in responding to the disaster?

The Context

You work for a newspaper; it prints articles of interest to the general public as the end result of its business processes; its revenue sources are subscription fees, but mostly advertising revenue.

The KPI and Performance Gaps

The editor is concerned that the articles in the newspaper be as truthful as possible, which is why there is such emphasis on fact-checking. There is a concern that the public trusts the paper to publish truthful information, or there will be a loss of readership, resulting in a reduction in subscriptions, but more importantly, a potential loss in advertising revenue.

Parsing the Information Need

To translate the information need above into framed computable questions, we need to ascertain the following by parsing the statement of the information need:

- (a) What do we mean by “the crew?” Who are these people? What was their mindset at the time they had to make decisions on whom to allow to board the lifeboats?
- (b) What does it mean for the crew to “follow the Law of the Sea?”
- (c) What is “the Law of the Sea?”
- (d) What do we mean by *responding*? When we say responding to the disaster, what does the response look like? What were the actions taken by the crew in response to the disaster?
- (e) What is “the disaster?” Was it when the iceberg struck? Was it when the crew realized the ship was going to sink? Was it when the boat sank and lifeboats were away?

Figure 3.6 describes the parsing process.

By researching the story, we determined that the crew assigned to each lifeboat were the gatekeepers on who boarded the lifeboats. We found they were ordinary seamen assigned by their officers to serve as gatekeepers to the lifeboats. Since there were not enough boats for everybody on the Titanic, this decision-making needed to be performed by the crew. The decision probably followed the well-known “Law of the Sea,” meaning “women and children first.” The seamen were charged with filling the boats with women and children before any men got aboard. Did that happen? If you find a positive answer, then we can tell the editor the reporter is truthful in the story. The facts will either support the decision to run the story as is or to change it before it prints.

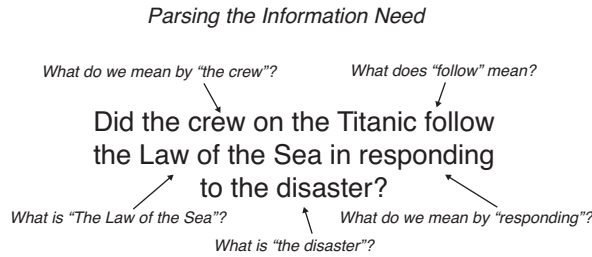


FIGURE 3.6 Parsing the request.

The Data Set

The critical data set for our purposes is the Titanic’s passenger manifest (<https://public.opendatasoft.com/explore/embed/dataset/titanic-passengers/table/>).

This publicly available data set shows the 1,309 passengers, their names, ages (for some), passenger class, and survival status. These features or variables in our data set should help us form and answer some well-framed questions. A copy of the data set can also be found in the *Case Data* data repository for this book under the *Chapter 3* folder.

Knowing all the facts about the information needs and the context that drives those needs, we are now prepared to frame some analytical questions and begin our analysis. Keep in mind these questions must be SMART: specific, measurable (computable), attainable, relevant, and have some time element in them.

The Framed Analytical Questions

We determined that there are some computations we could undertake that would support a positive or negative answer to the information need.

*What is the survival rate of women, and how does it compare to the survival rate of men?*²

*What is the survival rate of children, and how does it compare to the survival rate of adults?*²

They certainly would give us a powerful indication of whether the crew was following the Law of the Sea. That would be the question our editor was looking to answer. However, we could find a more valuable answer if we included additional information. We could analyze the survival rates for men, women, and children, and break those rates down by passenger class.

What are the survival rates of men, women, and children broken down by class?

The answer to this question might result in useful insights for the editor and the reporter, who could add to the story and make it more interesting. For example, it could give a competitive edge to the story versus the stories published in competing magazines. This is how an analyst adds value to their work: bringing back in-depth answers beyond the original information needs and supporting the KPIs driving those information needs.

We provide two additional case studies on framing analytics questions. They support the two major data mining case studies in this book: the San Francisco Airport Customer Survey case study and the Small Business Administration Loans Program case study.

CASE STUDY 3.1 – THE SAN FRANCISCO AIRPORT SURVEY

The Case

The San Francisco Airport (SFO) is in the airport operations industry. Its revenues come from two categories, which include (1) aviation-related revenues such as airline fees and passenger charges from supporting services, like baggage and cargo handling, as well as (2) rental space and non-aviation related revenues, such as concession grants, advertising, parking, and retail by providing different brand locations and exposure to their potential customers. The pressures that such an organization faces mainly come from three different aspects. The first challenge is the competitive environment, in which all competitors aim to become the leading airline hub in the western region of the US. The second is the challenge to reduce costs by improving operational efficiency and increasing revenues. Finally, SFO will face the effects of environmental impact because the company must meet government regulations and the public's expectations. SFO's management wants to perform better to survive in the airport industry and to provide the company with sustainable competitive advantages.

The Information Need

As expressed by the lead executive (the Chief Marketing Officer, in this case), the need is to find out what is keeping them from reaching their number one objective to be a highly ranked airport, and he also wants a recommendation of what can be done to improve performance. The organization has collected a lot of data with their survey over many years, but they do not have a smart way to recognize the key elements that impact their performance according to the data. With an intelligent analysis, the CMO expects their ranking to reach #1 and for premium passengers to choose them for their flights. The CMO needs to know the reasons why SFO is not the number one ranked airport in its class and wants to find out what can make SFO number one. The CMO also wants to know what motivates SFO's premium customers, what these premium customers care about, and any source of dissatisfaction with the airport.

The Context

SFO is not only in the transportation industry, but also in the public service industry. In their sector, although airports have some regional monopoly, they still need to compete with other nearby airports in many aspects, such as more optional airlines, more convenient facilities, and better ambiance to attract more passengers and airline companies to bring them more revenue, which is also the reason that they need to perform better. The analyst's results will be reported to the CMO in the marketing department of SFO. They are generally trying to achieve the #1 ranking by passengers as an International Gateway airport in their class and to be the airport of choice for premium passengers. After identifying the context, the project will do further business analysis to create business answers furthering SFO's efforts to achieve their objectives.

The KPI and Performance Gaps

The KPIs in this case are SFO's ranking in the ASQ survey (SFO 2022), sales revenue per customer, customer retention, customer satisfaction, and customer loyalty. The goals of the organization are (1) to be ranked first by passengers as an International Gateway airport in their

class; and (2) to be the airport of choice for premium passengers. The gaps are (1) SFO is not ranked first in the ASQ survey in the most recent year and what to do to improve their ranking to be first; and (2) to find out what shows improvement year over year. Therefore, the SFO CMO has a considerable need for improvement to raise the ranking among airports and satisfy their premium customers.

The Initial Set of Framed Analytical Questions

Among all the rating questions, which have the strongest impact on passenger satisfaction of SFO airport as a whole in the most recent survey?

Which elements have the most significant impact on customer satisfaction, as in the question “How would you rate the airport as a whole” in the most recent survey?

Which factors tells us of the most complaints from customers and measures the most significant performance gap we need to improve year over year?

What are the characteristics (both demographic and psychographic) of our premium customers (defined by people who fly 100,000 miles or more per year) year over year?

CASE STUDY 3.2 – SMALL BUSINESS ADMINISTRATION LOANS

The Context

Increasingly policymakers are looking to the small business sector as a potential engine of economic growth. Policies to promote the growth of small businesses include tax relief and direct and indirect subsidies through government lending programs. Encouraging lending to small businesses is the primary policy objective of the Small Business Administration (SBA) loan-guarantee program.

In the past 20 years, small businesses have created more than two-thirds of the new jobs in the United States. Twenty-eight million small businesses have employed 60 million Americans, accounting for 50 percent of the workforce in the private sector. It is worth noting that the SBA has played a positive role in the innovation and entrepreneurship of small businesses in the United States, providing a large number of loans, financing guarantees, government procurement services, business consulting services, and many other services. Among the various forms of direct services provided by the SBA, its SME credit financing service is the most successful service model, with rich practical experience. Analysis of the practices of the SBA in promoting credit financing for small enterprises provides the government with experience and reference in promoting the development of SMEs' innovation and entrepreneurship.

Why is This Analysis Critical?

Modern market economy countries, such as the US, are constantly re-examining their strategic positioning of small businesses and their impact on the national economy and social development. Modern market-economy countries generally pay more attention to the social value of small businesses and realize their irreplaceable decisive significance in promoting economic development, creating employment opportunities, maintaining social stability, and enhancing national competitiveness. For example, as a significant economic competitor, China also is constantly evaluating the social value, and the strategic significance of small businesses needs to be

further promoted to advance their economic development. The US must also be continually re-evaluating the effectiveness of its government programs, as well.

The Information Need

The purpose of this case is to study whether the current SBA policies and plans for its loan program plan are effective in macroeconomic and microeconomic terms. The SBA guaranteed lending program is one program aimed at improving small firms' access to credit. SBA loan guarantees are well established, and their volume has grown significantly over the past two decades.

Nearly 20 million small businesses have received direct or indirect help from one or another of the SBA's programs since 1953. As of 2020, the SBA's current business loan portfolio of roughly 219,000 loans is worth more than \$45 billion, making it the largest single financial backer of small businesses in the United States. Over the period 1991 to 2000, the SBA assisted almost 435,000 small businesses in obtaining more than \$94.6 billion in loans, more than in the entire history of the agency before 1991. No other lender in this country has been responsible for as much small business financing as the SBA has during that time (SBA 2022). These lending numbers are remarkable when one considers that SBA loan guarantees are aimed at that segment of small business borrowers that presumably would not otherwise have access to credit. *Is there a market failure that justifies the intervention of this magnitude?*⁹ Many economists believe that credit markets — whose efficient operation depends heavily on the ease with which lenders can gather information on borrowers — are indeed prone to failure when the nature of the borrowers makes it hard to obtain this information, as is the case with small businesses. *SBA management wants to know how well the SBA lending program is addressing that failure (and doing so with no adverse side effects). We also have to ensure that we know what the mechanics of the failure are and ask whether the program is designed to target only those areas where the breakdown in natural market forces arises.*

The KPIs and the Gap

*Has the SBA small business loan program been successful?*⁹ A particular area of concern for policymakers is whether small businesses have access to adequate credit. Growing businesses have an acute need for credit, but many small firms may have a hard time obtaining it because they are young and have little or no credit history. Lenders may also be reluctant to fund small firms with new and innovative products because of the difficulty associated with evaluating the risk of such products. If small businesses lack a sufficient supply of credit, policymakers should be concerned, for the next Google, Microsoft, or Starbucks might wither on the vine for want of funding. To the extent that some market failure significantly impairs the access small businesses have to credit, a rationale exists for supporting these businesses through government programs aimed at improving their access. We need to use historical data to study whether the implementation of this program is the same as its original intention.

The Initial Set of Framed Analytical Questions

The study will attempt to answer the following business information question: *Has the SBA small business loan program been successful?*⁹ To judge whether this program is successful, we choose a series of quantifying measurable indicators and associated framed questions.

1. Aiming at the goal of (in the words of the SBA): “strengthening cooperation with the federal government to ensure that the target share of the federal government procurement

contract is reached and exceeded, so as to increase the opportunities of small businesses, and to strengthen the fairness of the federal procurement contract certification process and data” (SBA 2022), we can pose the following framed question:

Was this a specific, measurable indicator of whether the contract share of small businesses in government procurement reached the statutory goal set by the federal government?

2. Another important measurable goal of this program is “to increase the investment in education, consultation and training resources of the SBA, and to help new small businesses and support existing small businesses. Focusing on core program resources to meet the needs of ordinary small enterprises and high growth small businesses” (SBA 2022), we can ask the following question:

Were the specific, measurable goals of supporting the need to establish a required minimum number of small businesses met?

3. Given the goal set by the SBA to “ensure that disaster assistance provided by the small business administration to enterprises, non-profit organizations, property owners, and lessees quickly, effectively, and efficiently, in order to maintain employment opportunities and help small businesses resume production and operation” (SBA 2022), the specific, measurable indicators are the proportion of household loans and corporate loans meeting the statutory standards of the loan operation.

Did the SBA meet the goal of the statutory requirements set by the government for the agency?

4. Aiming at the goal of “strengthening the connection between the small business administration and high growth small businesses and entrepreneurs through existing projects and job innovation, to drive innovation and create job opportunities more effectively”(SBA 2022), the specific, measurable indicators are the investment of long-term capital.

Did the SBA meet its goal of issuing a sizable portion of its loans to high growth technology-intensive businesses?

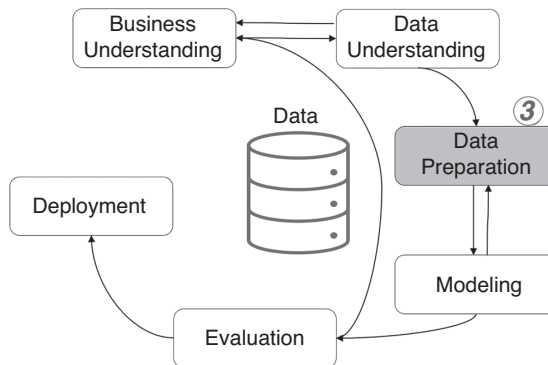
5. Because of the goal set by the SBA of “guiding federal agencies to understand the adverse effects of various unreasonable management regulations on small enterprises, reduce the burden of small enterprises, enhance the research on small enterprises, and create a good enterprise environment” (SBA 2022), the measurable indicators are the cost of various management systems for small businesses:

Were the costs of management systems within reach of small businesses served by the SBA?

REFERENCES

- Chapman, Pete. 2000. *CRISP-DM 1.0: Step-by-Step Data Mining Guide*. SPSS.
- SBA Office of Advocacy. 2022. “Fiscal Year 2022 Congressional Budget Justification, and Fiscal Year 2020 Annual Performance Report.” Accessed November 2, 2022. <https://cdn.advocacy.sba.gov/wp-content/uploads/2022/03/31085616/FY-2022-Congressional-Budget-Justification-and-FY-2020-Performance-Report.pdf>.
- SFO. 2022. “Customer Survey Data- 2010 Annual Customer Satisfaction Survey.” San Francisco International Airport. March 16, 2022. <https://www.flysfo.com/media/customer-survey-data>.
- Shron, Max. 2014. *Thinking with Data*. Sebastopol, CA: O’Reilly Media.
- Wirth, Rudiger, and Jochen Hipp. 2000. “CRISP-DM: Towards a Standard Process Model for Data Mining.” Essay. In *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, 29–40. 11-13 April 2000. Manchester.

DATA PREPARATION



This chapter covers the essential and time-consuming process of shaping data into a file form that can be analyzed. The result of our scraping, cleaning, and shaping will result in the production of simple flat files ready for application of the modeling tools. The flat-file format is popular for preparing data for analysis. Here, we describe the format and present examples of files that are in the flat-file format. Flat files are also popular for data input when programming in R. They are called *data frames*. We will also show how to shape categorical variables into other usable forms, such as a binary unit variable, and how to bin a numeric variable to create categories for analysis. We make extensive use of these techniques throughout the book.

The chapter includes several exercises on data shaping and data cleaning to further develop skills in this area. We continue to use and develop further two important case studies based on the two case data sets used throughout the rest of the book (the SFO Survey data set and SBA Loan data sets). These are important case studies in shaping and cleaning the data sets and preparing them for further data mining in later chapters. This chapter concludes with a case study in SQL queries to further develop skills with that important data extraction tool.

HOW DOES CRISP-DM DEFINE DATA PREPARATION?

Once we have analyzed the business situation and information needs, we need to prepare the data. The CRISP-DM process model defines this phase as follows.

Data Preparation

The data preparation phase covers all activities needed to construct the final data set [data that will be fed into the modeling tool(s)] from the initial raw data. Data preparation tasks are likely to be performed multiple times and not in any prescribed order. Tasks include table, record, and attribute selection, as well as transformation and cleaning of data for modeling tools.

This phase occurs after we have a good understanding of the business and of the data available to answer the framed questions. Remember that in the architecture of modeling, data preparation (extraction from databases, cleaning, and shaping) comes before model building, as seen in Figure 4.1.

CRISP-DM informs us of the activities and deliverables that are needed in this phase (see Data Preparation in Figure 4.2).

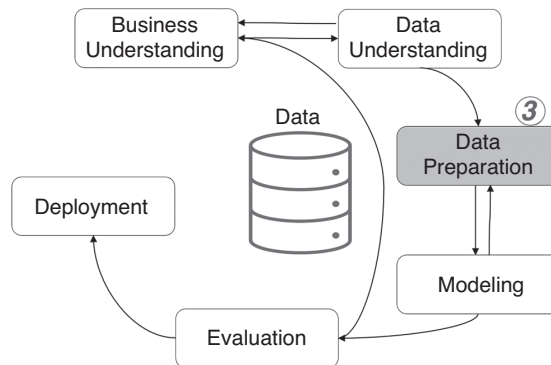


FIGURE 4.1 The CRISP-DM process model highlighting the data preparation phase and its relationship to the other phases.

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives Background Business Objectives Business Success Criteria Assess Situation Inventory of Resources Requirements Assumptions and Constraints Risks and Contingencies Terminology Costs and Benefits Determine Data Mining Goals Data Mining Goals Data Mining Success Criteria Produce Project Plan Project Plan Initial Assessment of Tools and Techniques	Collect Initial Data Initial Data Collection Report Describe Data Data Description Report Explore Data Data Exploration Report Verify Data Quality Data Quality Report	Select Data Rationale for Inclusion/Exclusion Clean Data Data Cleaning Report Construct Data Derived Attributes Generated Records Integrate Data Merged data Format Data Reformatted Data Data Set Data Set Description	Select Modeling Techniques Modeling Technique Modeling Assumptions Generate Test Design Test Design Build Model Parameter Settings Models Model Descriptions Assess Model Model Assessment Revised Parameter Settings	Evaluate Results Assessment of Data Mining Results w.r.t. Business Success Criteria Approved Models Review Process Review of Process Determine Next Steps List of Possible Actions Decision	Plan Deployment Deployment Plan Plan Monitoring and Maintenance Monitoring and Maintenance Plan Produce Final Report Final Report Final Presentation Review Project Experience Documentation

FIGURE 4.2 The CRISP-DM process model, with the data preparation step deliverables and activities highlighted.

In this phase, we decide what data is best suited for building our model. We extract data from various sources. We combine them and create a flat file of rows (population) and columns (variables) best suited to answer the framed questions. In the process, we may need to shape and clean the data set until we are satisfied. The resulting data set is ready for the application of algorithms to build models. In the language of machine learning, we now have a data set for training our model.

STEPS IN PREPARING THE DATA SET FOR ANALYSIS

In the first set of exercises, we will look at the importance of shaping and cleaning data files. Figure 4.3 shows the data cleansing cycle, with the many activities needed to prepare data for analysis, starting with importing the data, merging the data sets, standardizing and normalizing data, rebuilding missing data, de-duplicating, and verifying and enriching the data set. The object is to produce a data set in a form that is called a *flat-file format*. When expressed in that format, the first row of the table must contain all the variable names; every row is of the same nature, and there are no empty rows or columns. All other rows and columns outside of the table area should be clear of data. Once in that format, the table is ready for analysis, and we can safely apply analytic tools.

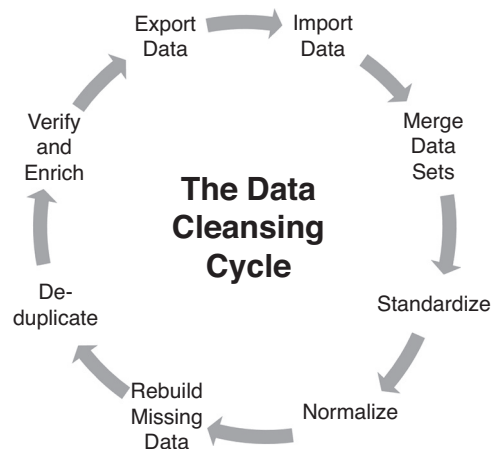


FIGURE 4.3 The data cleansing cycle.

The data source varies; sometimes, we extract data from a DBMS using SQL queries. Other times, we may obtain a comma-separated values file (with a .CSV extension) or we may obtain a formatted text file (with a .TXT extension). Due to the ubiquity of the Internet, many data sets are scraped from HTML-formatted web pages.

Once we practice loading data in various formats, we explore below cleaning it in Analysis Case 4.1. We practice using a small data file that contains several errors that need to be corrected. You are directed to the original data to find the original values. The exercise allows you to use tools in Excel that make the data cleaning process efficient.

The whole process of scraping, uploading, cleaning, annotating, and shaping the data file is often called *data wrangling*. Many studies have shown that this process is tedious and can take up to 80 percent of the overall time needed to perform the analysis. However, it is critical for

success in the analysis. The more skilled you are in the use of cleaning and shaping tools, and the wiser you are in their use, the sooner you will start the analysis and the less time you will need to find answers.

DATA SOURCES AND FORMATS

Analysts must deal with many data sources and formats. The various data and variable types are shown in Figure 4.4. The most common types of data we gather from business transactions are numeric and categorical. Computerized data is first collected and then stored to analyze financial transactions, for example. The emphasis in analysis is often on summarizing numeric data, which can easily be done with mathematical tools such as the averages, sum, maximum, and minimum. Summarizing categorical data used to be complicated. Initially, the only thing we could do with categories was to tabulate them, counting the occurrence of each category. It was not until the advent of Excel pivot table analysis that evaluating categorical data in detail became as easy and commonplace as analyzing numerical data.

Text data is much harder to evaluate. It requires us to count words, but much of the work also requires tabulation and quantization by hand. We developed some very laborious measures to do so. We show you how to work with text data in Chapter 13. Not until the advent of social media and electronic commerce — when we began to be flooded with textual data — did we need to go further and automate quantizing it to make sense of text data.

Numerical and categorical data are the most common data types, as shown in Figure 4.4. We use standard techniques to work with these data types, such as pivot tables and numerical summarization functions. With the advent of social networks and sophisticated data tools, text data analysis is now more commonplace.

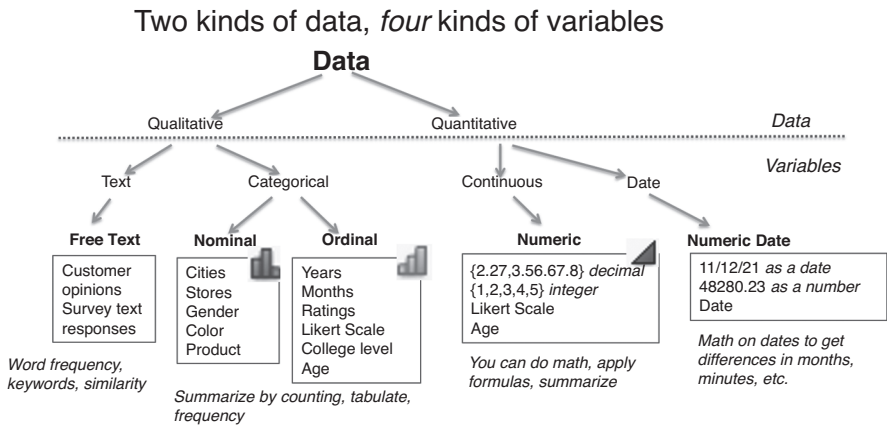


FIGURE 4.4 Categorizing the two types of data formats and four variable types.

It is essential to know where data is found and in what form in order to optimize the scraping and shaping process and ultimately produce it in the right format for analysis. In this chapter, we discuss the various forms in which we receive data. We also study how to extract data from RDBMS databases using the SQL query language.

WHAT IS DATA SHAPING?

Data comes to us in many shapes and sizes, as we saw in the previous chapter. For most of our analytical tools, the data should be in a tabular format. Shaping the data set entails transforming it from whatever shape the data is acquired in (such as a report, an SQL query output, a CSV file, or an Excel file) into the proper format ready to be analyzed with our tools.

Data analysts spend the great majority of their time cleaning and shaping the data set. A survey conducted by the data science company Crowdflower in 2016 shows the breakdown in tasks and the time needed to complete each (Figure 4.5).

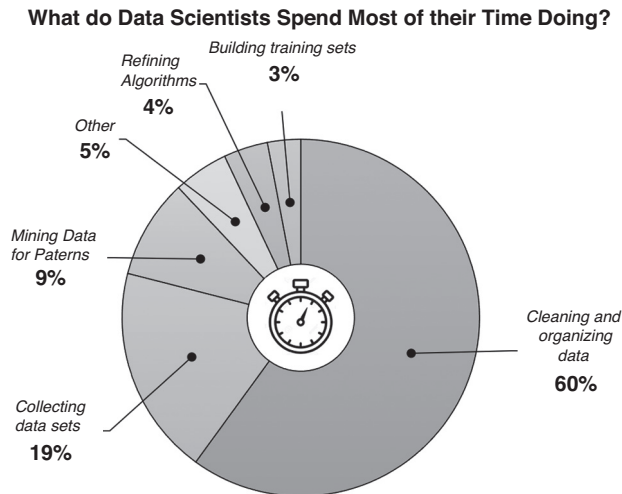


FIGURE 4.5 Typical proportions for cleaning, shaping, and analysis (CrowdFlower 2016).

THE FLAT-FILE FORMAT

Storing data in a simple structure of rows and columns is common today, often create using an Excel spreadsheet. This format has many limitations that may be overcome with sophisticated data structures of interconnected tables, such as an RDBMS, including indexing, economical storage, easier retrieval, and additional massive data sets. Corporate data may sometimes be found stored using these more complex systems, but to use the data to answer questions, we extract the data from these complex data sets and present it to the analyst in the form of a flat file of rows and columns.

A flat file can also be considered a database, albeit a simple one, with data stored uniformly. Records (the rows) follow a consistent format, and there are no structures for indexing or recognizing relationships between records. Columns are the named variables. The file is simple. A flat file can be a plain text file or a binary file. Relationships may be inferred from the data in the file, but the table format itself does not make those relationships explicit. Typically, all the rows are about the same population, such as orders, customers, patients, companies, or payments.

We often use spreadsheets as a form of database or as a container for data. We usually load these spreadsheets with many non-data elements not useful for data analysis. For example, a particular spreadsheet may be a report with titles, page numbers, and the coloring of specific cells to make it easier for humans to read and interpret the information. Some of this is *metadata* (data about the data set), such as the *data dictionary*. To make the analysis more straightforward,

we need to remove all of these human interface elements from the spreadsheet and format the raw data into a row and column format. Some of the summarization tools in spreadsheets, such as pivot tables, require us to format the file in this format.

Moreover, the programming language for statistical analysis, R, readily ingests data in this form. In R, we refer to this format of data as *data frames*. In most cases, the flat-file format is a convenient structure for analysis. Figure 4.6 shows a spreadsheet in the flat-file format.

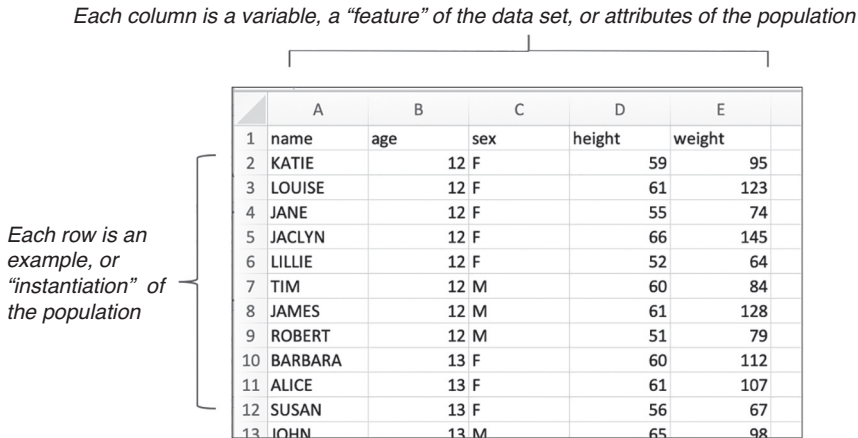


FIGURE 4.6 The flat-file format showing the elements of the rows and columns.

Before applying an analysis tool to a file, remember to ask “*Is this data set in a flat-file format?*” Shaping the data set into a flat file results in a tabular format, with each variable in a column and the top row of the table containing the variable names. Each row of the table is an instantiation for the population being documented. Figure 4.7 is an excellent example of a table in such a format.

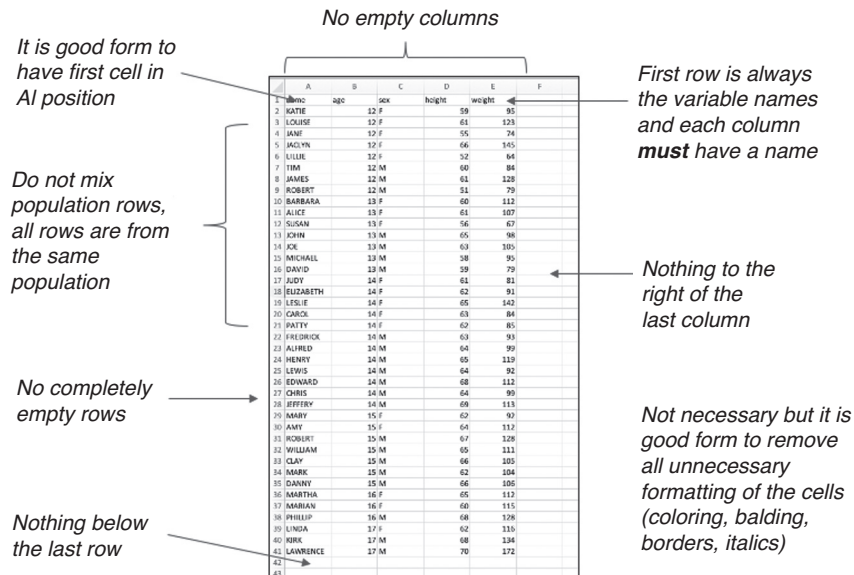


FIGURE 4.7 The characteristics of a flat file.

Let's consider the table given in Figure 4.8.

	A	B	C	D	E	F
1	PIZZA	BAKERY	SHOES	GIFTS	PETS	
2		80	150	48	100	25
3		125	40	35	96	80
4		35	120	95	35	30
5		58	75	45	99	35
6		110	160	75	75	30
7		140	60	115	150	28
8		97	45	42	45	20
9		50	100	78	100	75
10		65	86	65	120	48
11		79	87	125	50	20
12		35	90			50
13		85				75
14		120				55
15						60
16						85
17						110
18						
19	Business Startup Costs					
20						
21	The following data represent business startup costs (thousands of dollars) for shops.					
22	PIZZA = startup costs for pizza					
23	BAKERY = startup costs for baker/donuts					
24	SHOES = startup costs for shoe stores					
25	GIFTS = startup costs for gift shops					
26	PETS = startup costs for pet stores					
27	Reference: <i>Business Opportunities Handbook</i>					
28						

FIGURE 4.8 A data file not in the flat-file format.

Apply the flat-file format question to the table: *Is this table in a flat-file format?* The obvious answer is *no*. Excel can handle this data format, and a great deal of analysis can be performed in Excel on the data set as is, but most analysis programs will not accept the data in that form. In the table in Figure 4.8, we see two variables in the data set. One variable is the type of store that was being started, and the other variable is the starting capital needed to open that type of store. One variable is categorical; the other variable is numerical. In reality, there should be a two-column, or two-variable, table, with each row of the table being a particular store type needing individual capital cost. We might even add a third variable with a unique identifier to each row or observation, but it is not necessary to do so. It would require significant data shaping to put the data from this table into a flat-file format so it can be uploaded to an analysis platform other than Excel. If you have many tables that need to be reshaped into flat files, this process should be automated with an Excel macro or a VBA script.

APPLICATION OF TOOLS FOR DATA ACQUISITION AND PREPARATION

In the next two exercises, we show approaches to shaping the data file into a flat-file format and how to remove errors from a data file.

EXERCISE 4.1 – SHAPING THE DATA FILE

Often, we either use a program to produce a report or receive a report file electronically from coworkers or other colleagues. If the data is already in tabular form, shaping the data set into a flat file is relatively straightforward. If the data has been shaped into a reporting format that is more complex, we are in for extensive re-work.

Take, for example, a proposed yearly departmental budget broken down by month, as shown in Figure 4.9. The data file may be found in the *Case Data* store of files that accompanies this book. It will be in the *Chapter 4* folder. Open the *Data Cleaning and Shaping Exercise.xlsx* file.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Budget Proposal													
2														
3	Planned Expenses	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	YEAR
4	Employee Costs													
5	Employee wages	\$85,000	\$85,000	\$85,000	\$87,500	\$87,500	\$87,500	\$87,500	\$92,400	\$92,400	\$92,400	\$92,400	\$92,400	\$1,067,000
6	Employee benefits	22,950	22,950	22,950	23,625	23,625	23,625	23,625	24,948	24,948	24,948	24,948	24,948	288,090
7	Employee training	4,000	4,000	4,000	4,000	4,000	4,000	4,000	4,000	4,000	4,000	4,000	4,000	48,000
8	Subtotal	\$107,950	\$107,950	\$107,950	\$111,125	\$111,125	\$111,125	\$111,125	\$117,348	\$117,348	\$117,348	\$117,348	\$117,348	\$1,355,090
9														
10	Office Costs													
11	Office lease	\$12,000	\$12,000	\$12,000	\$12,000	\$12,000	\$12,000	\$12,000	\$12,000	\$12,000	\$12,000	\$12,000	\$12,000	\$144,000
12	Office utilities	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	24,000
13	Office supplies	250	250	250	250	250	250	250	250	250	250	250	250	3,000
14	Subtotal	\$14,250	\$14,250	\$14,250	\$14,250	\$14,250	\$14,250	\$14,250	\$14,250	\$14,250	\$14,250	\$14,250	\$14,250	\$171,000
15														
16	Marketing Costs													
17	Web site costs	\$2,000	\$2,000	\$2,000	\$2,000	\$2,000	\$2,000	\$2,000	\$2,000	\$2,000	\$2,000	\$2,000	\$2,000	\$24,000
18	Collateral costs	500	500	500	500	500	500	500	500	500	500	500	500	6,000
19	Marketing events	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	60,000
20	Subtotal	\$7,500	\$7,500	\$7,500	\$7,500	\$7,500	\$7,500	\$7,500	\$7,500	\$7,500	\$7,500	\$7,500	\$7,500	\$90,000
21														
22														
23	TOTALS													
24	Monthly Planned Expenses	\$129,700	\$129,700	\$129,700	\$132,875	\$132,875	\$132,875	\$132,875	\$139,098	\$139,098	\$139,098	\$139,098	\$139,098	\$1,616,090
25	TOTAL Planned Expenses	\$129,700	\$259,400	\$389,100	\$521,975	\$654,850	\$787,725	\$920,600	\$1,059,698	\$1,198,796	\$1,337,894	\$1,476,992	\$1,616,090	

FIGURE 4.9 A typical departmental budget proposal in report format.

Your boss received this budget proposal as an Excel file (with no data table to support it). The boss is wondering if you can generate a report of expense types broken down by costs types, similar to what is shown in Figure 4.10. That can easily be generated by a pivot table in Excel but not from the existing data format given in the budget spreadsheet.

Sum of Expense	Column Labels			
Row Labels	Employee Costs	Marketing Costs	Office Costs	Grand Total
Employee wages	\$ 1,067,000			\$ 1,067,000
Employee benefits	\$ 288,090			\$ 288,090
Office lease			\$ 144,000	\$ 144,000
Marketing events		\$ 60,000		\$ 60,000
Employee training	\$ 48,000			\$ 48,000
Web site costs		\$ 24,000		\$ 24,000
Office utilities			\$ 24,000	\$ 24,000
Collateral costs		\$ 6,000		\$ 6,000
Office supplies			\$ 3,000	\$ 3,000
Grand Total	\$ 1,403,090	\$ 90,000	\$ 171,000	\$ 1,664,090

FIGURE 4.10 Required report of expenses by cost types.

The original table that created this budget report may be extracted from the formatted budget report by lifting each number from the budget report and creating the flat file by hand. That is probably too much work. You could, by clever manipulation, extract the main data points from the original table and create the flat file from which many questions can be answered by further analysis, including the results produced in Figure 4.10. We need to shape the data.

If you look carefully and analyze the table, you notice that there are four variables of interest in that table. Everything else is summaries or titles. We also do not need the coloring of the cells, which is not required for a machine to process the data. The four variables are *Month*, *Expense Type*, *Cost Type*, and *Amount*.

Starting with the given budget table (Figure 4.9), we copy and paste (values only) the entire table into another worksheet. Delete all summary rows and columns and the cells with titles to leave behind the data, as shown in Figure 4.11.

The table is laid out with the months across the top of the table. The months need to be in a column in our flat file since it is one of the four variables. The best way to quickly further

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Planned Expenses	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2	Employee Costs												
3	Employee wages	85000	85000	85000	87500	87500	87500	87500	92400	92400	92400	92400	92400
4	Employee benefits	22950	22950	22950	23625	23625	23625	23625	24948	24948	24948	24948	24948
5	Employee training	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000	4000
6	Office Costs												
7	Office lease	12000	12000	12000	12000	12000	12000	12000	12000	12000	12000	12000	12000
8	Office utilities	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000
9	Office supplies	250	250	250	250	250	250	250	250	250	250	250	250
10	Marketing Costs												
11	Web site costs	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000
12	Collateral costs	500	500	500	500	500	500	500	500	500	500	500	500
13	Marketing events	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000

FIGURE 4.11 Budget table stripped down to just the data elements without subtotals and labeling components.

transform the table is to copy and paste the transformation of the table into another worksheet. Figure 4.12 shows the transposed table pasted into a new tab of the Excel file.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Planned Expenses	Employee Costs	Employee wages	Employee benefits	Employee training	Office Costs	Office lease	Office utilities	Office supplies	Marketing Costs	Web site costs	Collateral costs	Marketing events
2	Jan		85000	22950	4000		12000	2000	250		2000	500	5000
3	Feb		85000	22950	4000		12000	2000	250		2000	500	5000
4	Mar		85000	22950	4000		12000	2000	250		2000	500	5000
5	Apr		87500	23625	4000		12000	2000	250		2000	500	5000
6	May		87500	23625	4000		12000	2000	250		2000	500	5000
7	Jun		87500	23625	4000		12000	2000	250		2000	500	5000
8	Jul		87500	23625	4000		12000	2000	250		2000	500	5000
9	Aug		92400	24948	4000		12000	2000	250		2000	500	5000
10	Sep		92400	24948	4000		12000	2000	250		2000	500	5000
11	Oct		92400	24948	4000		12000	2000	250		2000	500	5000
12	Nov		92400	24948	4000		12000	2000	250		2000	500	5000
13	Dec		92400	24948	4000		12000	2000	250		2000	500	5000

FIGURE 4.12 The transposed table.

Now, the hard work begins. Each number in the table has associated with it three categorical variables: a month, an expense type, and a cost type. There are monthly columns across the table, and they need to be lifted and copied to be in four vertical columns. The result is one table, shown in Figure 4.13, with four variables, the month, the expense type, the cost type, and the amount.

As always, when moving a lot of data, we want to make sure that we do not introduce any errors. Thus, we should reproduce the budget report using a pivot table and compare it to ensure that we have the same numbers. Once we have the data in a flat-file format, creating a summary using a pivot table is a simple matter. We can easily recreate the budget report (Figure 4.9) with a pivot table shown in Figure 4.14 by judicious use of a pivot definition table with subcategories, defined in Figure 4.15.

When comparing the pivot table (shown in Figure 4.14) that recreated the budget report to the original report we were given (shown in Figure 4.9), we notice something very curious. The numbers in row eight of the original report do not match the correspondingly computed numbers in our pivot table report. How can that be? Which is wrong, the pivot table or the original report?

You double-check the individual data points. They all seem right; they appear to have transferred over properly. Why is there a discrepancy? Then you check the formulas behind the subtotals in row eight and notice that whoever created the report did not pick up all three rows of cost data above the subtotal to aggregate them, but only two of the rows for row eight.

	A	B	C	D
1	Month	Category	Expense Type	Expense
2	Jan	Employee Costs	Employee wages	85000
3	Feb	Employee Costs	Employee wages	85000
4	Mar	Employee Costs	Employee wages	85000
5	Apr	Employee Costs	Employee wages	87500
6	May	Employee Costs	Employee wages	87500
7	Jun	Employee Costs	Employee wages	87500
8	Jul	Employee Costs	Employee wages	87500
9	Aug	Employee Costs	Employee wages	92400
10	Sep	Employee Costs	Employee wages	92400
11	Oct	Employee Costs	Employee wages	92400
12	Nov	Employee Costs	Employee wages	92400
13	Dec	Employee Costs	Employee wages	92400
14	Jan	Employee Costs	Employee benefits	22950
15	Feb	Employee Costs	Employee benefits	22950
16	Mar	Employee Costs	Employee benefits	22950
17	Apr	Employee Costs	Employee benefits	23625
18	May	Employee Costs	Employee benefits	23625
19	Jun	Employee Costs	Employee benefits	23625
20	Jul	Employee Costs	Employee benefits	23625
21	Aug	Employee Costs	Employee benefits	24948
22	Sep	Employee Costs	Employee benefits	24948

FIGURE 4.13 Final flat file showing the four variables: Month, Expense Type, Cost Type, and Amount.

Sum of Expense	Column L												Grand Total
Row Labels	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Grand Total
Employee Costs	111,950	111,950	111,950	115,125	115,125	115,125	115,125	121,348	121,348	121,348	121,348	121,348	1,403,090
Employee benefits	22,950	22,950	22,950	23,625	23,625	23,625	23,625	24,948	24,948	24,948	24,948	24,948	288,090
Employee training	4,000	4,000	4,000	4,000	4,000	4,000	4,000	4,000	4,000	4,000	4,000	4,000	48,000
Employee wages	85,000	85,000	85,000	87,500	87,500	87,500	87,500	92,400	92,400	92,400	92,400	92,400	1,067,000
Marketing Costs	7,500	7,500	7,500	7,500	7,500	7,500	7,500	7,500	7,500	7,500	7,500	7,500	90,000
Collateral costs	500	500	500	500	500	500	500	500	500	500	500	500	6,000
Marketing events	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	5,000	60,000
Web site costs	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	24,000
Office Costs	14,250	14,250	14,250	14,250	14,250	14,250	14,250	14,250	14,250	14,250	14,250	14,250	171,000
Office lease	12,000	12,000	12,000	12,000	12,000	12,000	12,000	12,000	12,000	12,000	12,000	12,000	144,000
Office supplies	250	250	250	250	250	250	250	250	250	250	250	250	3,000
Office utilities	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	24,000
Grand Total	133,700	133,700	133,700	136,875	136,875	136,875	136,875	143,098	143,098	143,098	143,098	143,098	1,664,090

FIGURE 4.14 Recreation of the budget report using a pivot table applied to the final flat file.

The formulas are correct for all the other subtotals, but not for row eight. The original report was wrong! Once you change the formula, the two tables match. You have been able to clean up the original report, which, if circulated as is, would have given the wrong budget numbers in the proposed budget. The step-by-step data shaping we just went through is given in the various worksheets in the spreadsheet containing the original data.

EXERCISE 4.2 - CLEANING THE DATA FILE

Let's perform additional data cleaning and data shaping work with some data files. A drug manufacturer has collected drug test data on 178 patients. We suspect that the data has transcription problems. (There were errors when entering data into the computer from the experiment

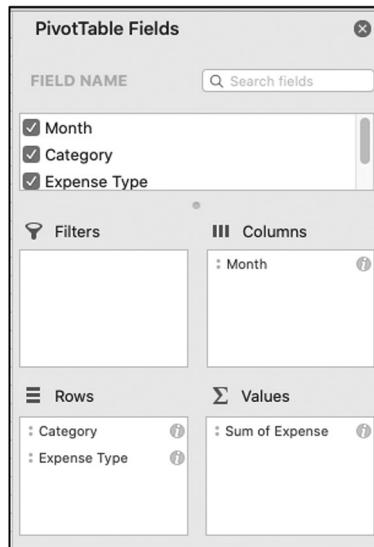


FIGURE 4.15 The pivot table definition for the budget report recreation.

notes.) Before analyzing the data set, we want to make sure it is free of errors. Follow the instructions below to prepare the data set for analysis.

You may also obtain the data by this other method:

1. Using the *Case Data* set and in the *Chapter 4* folder, find the file *calciumdata.txt*. The data dictionary for this file is *calcium.txt*.
2. Open *calciumdata.txt* using Excel.
3. Employ the Excel function to import a data file from the Excel Data ribbon set of functions.
4. The data is in columns, but there are no column titles. Use the data dictionary to add column titles, so all the variables are labeled. Note that the file is now in a flat-file format with columns as variables and rows as records.

Save the file as *calcium.xlsx*.

The file contains many errors. Clean it by looking at it and correcting these errors. For example, some of the numbers in the “sex” column are coded as “22” instead of “2.” That may be easily corrected. Fixing data coded as “12” is more challenging. If you need to refer to the original data that was collected, use this link to access the original observations: <http://academic.csuohio.edu/holcomb/learn/bigtable.htm>.

When you feel reasonably sure you have a clean data file, use Excel to answer the following questions:

How many men and how many women were in the study? Sort by gender and compute a sum for each group.

Were the tests evenly distributed over the labs? Sort by lab type and compute subtotals by lab type.

Are the calcium levels for the males above or below the average for the females in the test? Sort by gender and use the AVG function to average the CAMMOL columns for each sex.

Check your results against the solutions in Figure 4.16. Keep cleaning the data until you have found all the errors.

Solution	
Question 1	
Male	91
Female	87
Question 2	
Lab 1	88
Lab 2	42
Lab 3	16
Lab 4	14
Lab 5	11
Lab 6	6
Question 3	
Average for CAMMOL for males	2.32
Average for CAMMOL for females	2.39

FIGURE 4.16 Analysis results computed after cleaning the data file.

ENSURING THE RIGHT VARIABLES ARE INCLUDED

We are used to accepting data sets as given to us. However, when we begin investigating a data set and describing the various variables, we may discover we do not have all the variables needed to answer the questions. We might have looked at the framed questions and made sure that we acquired data with the variables we needed to answer them. In some cases, we might need to compute a variable to answer the framed questions.

Take, for example, the question in Chapter 3 on the Titanic disaster case. In the parsing of the case, we framed two analytical (computable questions):

What is the survival rate of women, and how does it compare to the survival rate of men?

What is the survival rate of children, and how does it compare to the survival rate of adults?

We determined that the available data set (the passenger manifest with survival statistics per passenger) would probably suffice (see Figure 4.17).

	A	B	C	D	E	F	G
1	pclass	survived	name	sex	age	embarked	home.dest
2	1st	1	Allen, Miss. Elisabeth Walton	female	29	Southampton	St Louis, MO
3	1st	1	Allison, Master. Hudson Trevor	male	0.9	Southampton	Montreal, PQ / Chesterville, ON
4	1st	0	Allison, Miss. Helen Loraine	female	2	Southampton	Montreal, PQ / Chesterville, ON
5	1st	0	Allison, Mr. Hudson Joshua Crei	male	30	Southampton	Montreal, PQ / Chesterville, ON
6	1st	0	Allison, Mrs. Hudson J C (Bessi	female	25	Southampton	Montreal, PQ / Chesterville, ON
7	1st	1	Anderson, Mr. Harry	male	48	Southampton	New York, NY
8	1st	1	Andrews, Miss. Kornelia Theodos	female	63	Southampton	Hudson, NY
9	1st	0	Andrews, Mr. Thomas Jr	male	39	Southampton	Belfast, NI
10	1st	1	Appleton, Mrs. Edward Dale (Cha	female	53	Southampton	Bayside, Queens, NY
11	1st	0	Artagaveytia, Mr. Ramon	male	71	Cherbourg	Montevideo, Uruguay
12	1st	0	Astor, Col. John Jacob	male	47	Cherbourg	New York, NY
13	1st	1	Astor, Mrs. John Jacob (Madelei	female	18	Cherbourg	New York, NY
14	1st	1	Aubart, Mme. Leontine Pauline	female	24	Cherbourg	Paris, France
15	1st	1	Barber, Miss. Ellen ("Nellie"	female	26	Southampton	
16	1st	1	Barkworth, Mr. Algernon Henry W	male	80	Southampton	Hessle, Yorks
17	1st	0	Baumann, Mr. John D	male	NA	Southampton	New York, NY
18	1st	0	Baxter, Mr. Quigg Edmond	male	24	Cherbourg	Montreal, PQ
19	1st	1	Baxter, Mrs. James (Helene DeLa	female	50	Cherbourg	Montreal, PQ
20	1st	1	Bazzani, Miss. Albin	female	32	Cherbourg	

FIGURE 4.17 Titanic passenger manifest data set showing some of the variables needed (such as sex, survival, and class) and missing variables required, such as a child/adult variable.

We can answer the men vs. women survival ratio question with the *sex* variable, but what do we do with the children vs. adult question? Yes, there is a variable called *age*, but it is not easy to use as it stands to separate children from adults as categories to use in a pivot table. One way to do it is to create a new variable. We recognize that we can use the numeric variable *age* by converting it to a categorical variable *child/adult* with two states, those below 12 years old labeled as a child, and those above 12 years old labeled as an adult. This step is called *binning*, as in categorizing date ranges or any other variable ranges, into categories.

Further, since this is a binary choice (only two choices), it is a binary binning step. Since there were passengers whose age we were not given (labeled as NA), we must be careful not to assign one of the two categories to that variable. All of this can be done with the proper sorting and the use of a logical formula (shown in Figure 4.18).

	A	B	C	D	E	F	G	H
1	pclass	survived	name	sex	age	child/adult	embarked	home.dest
2	1st	1	Allen, Miss. Elisabeth Walton	female	29	adult	Southampton	St Louis, MO
3	1st	1	Allison, Master. Hudson Trevor	male	0.9	child	Southampton	Montreal, PQ / Chesterville, ON
4	1st	0	Allison, Miss. Helen Loraine	female	2	child	Southampton	Montreal, PQ / Chesterville, ON
5	1st	0	Allison, Mr. Hudson Joshua Crei	male	30	adult	Southampton	Montreal, PQ / Chesterville, ON
6	1st	0	Allison, Mrs. Hudson J C (Bessi	female	25	adult	Southampton	Montreal, PQ / Chesterville, ON
7	1st	1	Anderson, Mr. Harry	male	48	adult	Southampton	New York, NY
8	1st	1	Andrews, Miss. Kornelia Theodos	female	63	adult	Southampton	Hudson, NY
9	1st	0	Andrews, Mr. Thomas Jr	male	39	adult	Southampton	Belfast, NI
10	1st	1	Appleton, Mrs. Edward Dale (Cha	female	53	adult	Southampton	Bayside, Queens, NY
11	1st	0	Artagaveytia, Mr. Ramon	male	71	adult	Cherbourg	Montevideo, Uruguay
12	1st	0	Astor, Col. John Jacob	male	47	adult	Cherbourg	New York, NY
13	1st	1	Astor, Mrs. John Jacob (Madelei	female	18	adult	Cherbourg	New York, NY
14	1st	1	Aubart, Mme. Leontine Pauline	female	24	adult	Cherbourg	Paris, France
15	1st	1	Barber, Miss. Ellen \"Nellie\"	female	26	adult	Southampton	
16	1st	1	Barkworth, Mr. Algernon Henry W	male	80	adult	Southampton	Hessle, Yorks
17	1st	0	Baumann, Mr. John D	male	NA		Southampton	New York, NY
18	1st	0	Baxter, Mr. Quigg Edmond	male	24	adult	Cherbourg	Montreal, PQ
19	1st	1	Baxter, Mrs. James (Helene DeLa	female	50	adult	Cherbourg	Montreal, PQ
20	1st	1	Bazzani, Miss. Albina	female	32	adult	Cherbourg	

FIGURE 4.18 Titanic passenger manifest data with binned variable *child/adult* and the formula needed to assign a category to each passenger.

The choice of the boundary between adult and child must be made carefully. It is often a judgment call by the analyst given circumstances and the desired bin sizes. In this case, we need to consider who qualifies as a child. It is easy to use today's measure of majority and say anyone under 18 is a child, but careful historical and circumstantial indicators would dictate otherwise. Consider that it was 100 years ago. Who was considered a child then, especially when we did not have child work laws? One could quickly say those much younger than 18. Also, consider the circumstances. A seaman was standing by each lifeboat and making the decision. Who looked like a child to *him*? For female children, it was an easy decision: she was a female and females go in the boat! For males, those who were considered children were probably much younger – maybe 10 or 11 or even 12, probably not more. Just ensure that your bin boundaries are defensible when people challenge you with the following questions: “Who do you put in each bin?” and “How did you decide on the bin boundaries?”

One way to check the sensitivity to the bin boundary selection (10 or 11 or 12 years old as the child-to-adult transition age) is to analyze with various age levels (10, 11, and 12) and see if there is much difference when using each one. Usually, there is very little difference, and then you are ready with a defensible answer. The case studies at the end of this chapter provide more opportunities to practice creating needed additional variables using binning.

USING SQL TO EXTRACT THE RIGHT DATA SET FROM DATA WAREHOUSES

We sometimes get to create the table from scratch by extracting it from a complex, rich, and detailed data set contained in a corporate database. These very often come in the form of an RDBMS, such as Oracle financials, Salesforce CRM, or some other corporate data repository. Today, they may also be found as part of a data warehouse, which most certainly is an RDBMS. In this case, we get a chance to determine precisely which variables we need in the table and can constrain them to fulfill complex logical filtering criteria.

The tool to extract this flat file with our data is the language used in RDBMS environments to create, manage, and read data sets: scripts written in SQL. Although SQL has functions that allow it to perform complex mathematical and summarization tasks, it is best we leave computations to our data mining tool. We shall limit the use of SQL to extract what we need to build our table. Thus, we will only need to be familiar with a few SQL read commands.

To get ready to use SQL for data extraction, we must have available and consult a very important document: the data dictionary. Please consult with the database administrator in your organization to get access to this tool. It details all variables, what their ranges are, and in which tables they may be found. The schema, the organization of the database, is also detailed there. The data dictionary helps you to know which tables you need to access.

Let's say we have a two-table database, as shown in Figure 4.19.

customerid	order_date	item	quantity	price
10330	30-Jun-1999	Pogo stick	1	28.00
10101	30-Jun-1999	Raft	1	58.00
10298	01-Jul-1999	Skateboard	1	33.00
10101	01-Jul-1999	Life		
10299	06-Jul-1999	Para		
10339	27-Jul-1999	Umb		
10449	13-Aug-1999	Unic		
10439	14-Aug-1999	Ski f		
10101	18-Aug-1999	Rain		

customerid	firstname	lastname	city	state
10101	John	Gray	Lynden	Washington
10298	Leroy	Brown	Pinetop	Arizona
10299	Elroy	Keller	Snoqualmie	Washington
10315	Lisa	Jones	Oshkosh	Wisconsin
10325	Ginger	Schultz	Pocatello	Idaho
10329	Kelly	Mendoza	Kailua	Hawaii
10330	Shawn	Dalton	Cannon Beach	Oregon
10338	Michael	Howell	Tillamook	Oregon
10339	Anthony	Sanchez	Winslow	Arizona
10408	Elroy	Cleaver	Globe	Arizona

FIGURE 4.19 A simple two-table database for the SQL query example.

A simple SQL command to extract a table would look like the following:

```
SELECT firstname, lastname, city
FROM customers
WHERE state = 'Washington';
```

The extract command is `SELECT`, followed by which variables you want to be extracted (they become the columns of your flat file output). The `FROM` command lets the interpreter know which table to extract it from, followed by the table name. Using the `WHERE` clause restricts the output to the rows where some variable condition is specified, as shown.

Of course, SQL is much more complex and powerful than this simple query shows, but you get the idea. It pays for a data miner to be fluent in SQL, since so much of the data we need is embedded in corporate RDBMS systems and has to be extracted from them. We provide additional SQL exercises in Case Study 4.3 at the end of this chapter.

CASE STUDY 4.1: CLEANING AND SHAPING THE SFO SURVEY DATA SET

To prepare the data set for further analysis, we need to clean the data set. Using Excel, open the *SFO 2018 Survey Data.xlsx* file found in the *Case Data* depository in the *SFO Survey Data* folder. The data dictionary is found under the *codes* tab in the spreadsheet. For opinion questions (codes as Q7 questions) we have a mix of codes. Some are the customer's opinion (1-5) but there also other codes, (0 = left blank, 6 = no opinion.) We should remove all rows that have no impact on the result we are looking for. For example, in column Q7ALL, the values 0, 6, or blank are meaningless in our analysis, and we should remove those rows. In Excel, we can easily filter them out using the Filter function and delete them. The results are shown in Figure 4.20.

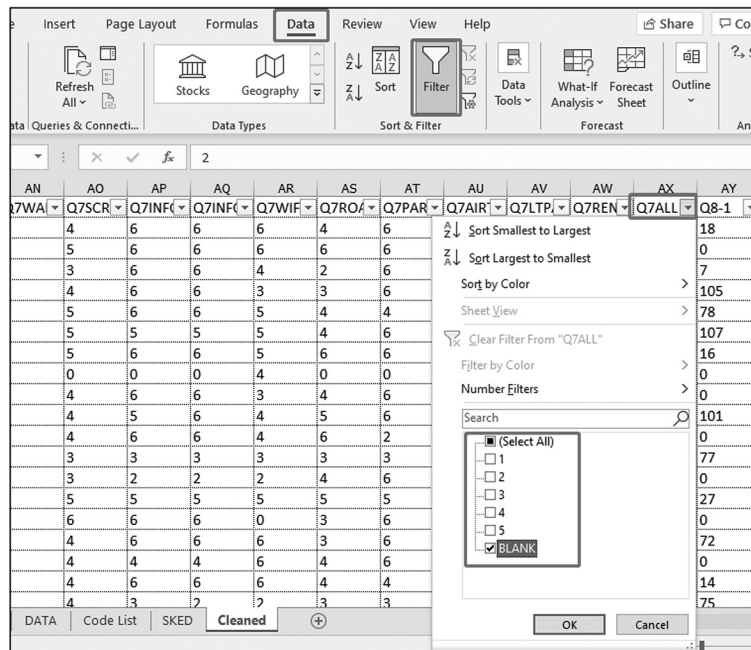


FIGURE 4.20 Use the Filter function to filter out meaningless rows.

Once you see only the rows satisfying the specific condition you set, delete those cells to obtain a result like that shown in Figure 4.21.

Turn off the Filter function, and the remaining rows are the data you want to analyze. Similar to Q7ALL, when you analyze Q9ALL and Q10SAFE, you should remove cells with 0, 6, or blank to get a more meaningful result.

After removing all meaningless cells, a new sheet will look like that shown in Figure 4.22.

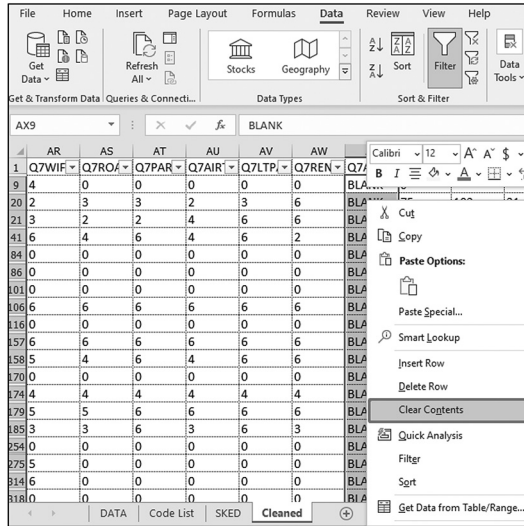


FIGURE 4.21 Steps to delete specific cells.

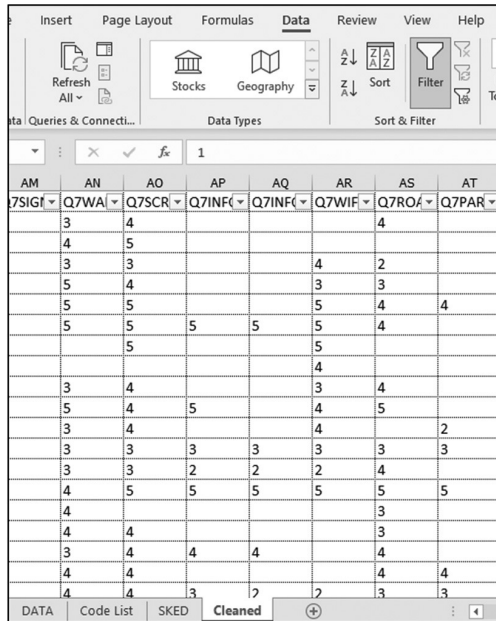


FIGURE 4.22 Output after cleaning.

CASE STUDY 4.2: SHAPING THE SBA LOANS DATA SET

Excel is useful for cleaning and shaping small data sets; however, when the data set is large, Excel may not be a good tool. In this case study, we will use R to prepare the *FOIA Loans Data* data set due to the size of the data set. The data set may be found in the *Case Data* depository under *SBA Loans Data* folder.

Import the data set into RStudio, as shown in Figure 4.23(a) and (b).

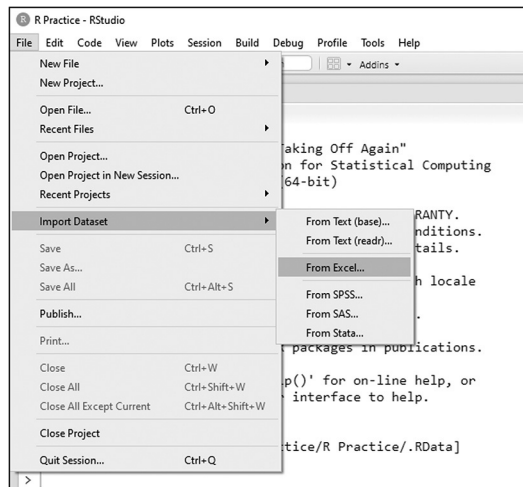


FIGURE 4.23(a) Steps to import data set into *RStudio*.

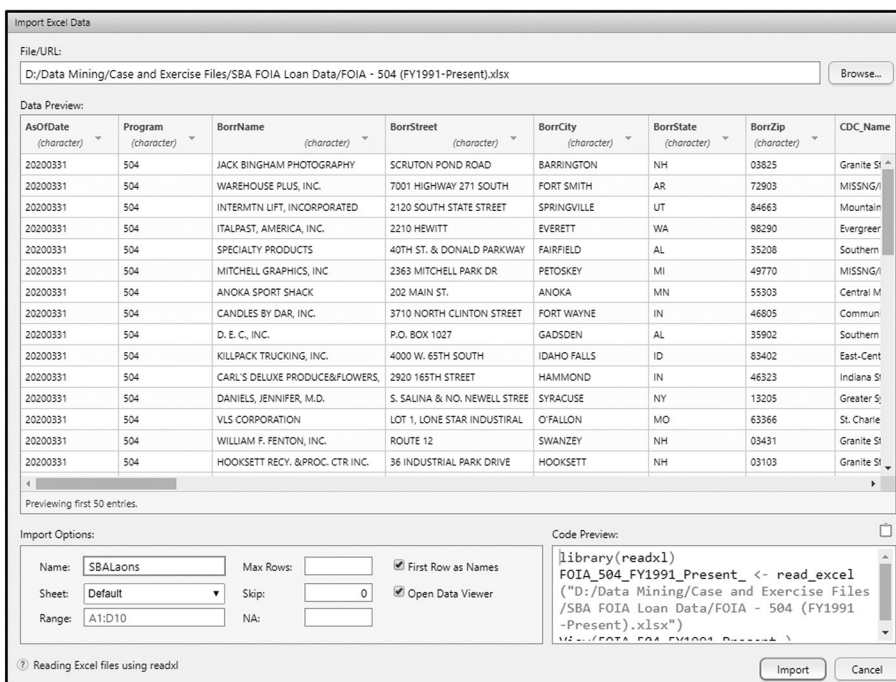


FIGURE 4.23(b) Steps to import data set in *RStudio*.

We want to know what year each loan was approved. To do so, we can use the following code to create a new variable `ApprovalYear`.

```
SBALaons$ApprovalYear <- as.numeric(format(SBALaons$ApprovalDate, "%Y"))
```

We may want to know how many months each loan is used to pay all back. To do so, we can use the interval function under the package `lubridate`.

```
install.packages("lubridate")
library(lubridate)
SBALaons$PaidInFullPeriod <-interval(SBALaons$ApprovalDate,
SBALaons$PaidInFullDate) %/% months(1)
```

We can also create a new variable, FUNDED, to substitute variable LoanStatus for future analysis by assigning “NOT FUNDED” to 0, “PIF” or “CHGOFF” to 1, and all others to blank by using the following code.

```
library(dplyr)
SBALaons <- SBALaons %>% mutate(FUNDED = case_when(
  SBALaons$LoanStatus == "NOT FUNDED" ~ "0",
  SBALaons$LoanStatus == "PIF" | LoanStatus == "CHGOFF" ~ "1",
  TRUE ~"Blank"
))
```

Like the previous situation, we can *bin* (make into a binary variable with only two values, 0 or 1) the variable LoanStatus to create a new variable PAIDOFF.

```
SBALaons <- SBALaons %>% mutate(PAIDOFF = case_when(
  SBALaons$LoanStatus == "CHGOFF" ~ "0",
  SBALaons$LoanStatus == "PIF" ~ "1",
  TRUE ~"Blank"
))
```

The shaped data set will look like Figure 4.24.

Date	ChargeOffDate	GrossChargeOffAmount	JobsSupported	ApprovalYear	PaidInFullPeriod	FUNDED	PAIDOFF
1	NA	0	4	1990	175	1	1
1	NA	0	8	1990	175	1	1
1	NA	0	35	1990	255	1	1
1	NA	0	11	1990	175	1	1
	NA	0	15	1990	NA	Blank	Blank
	NA	0	74	1990	NA	Blank	Blank
1	NA	0	8	1990	175	1	1
1	NA	0	38	1990	175	1	1
1	NA	0	12	1990	175	1	1
1	NA	0	12	1990	176	1	1
1	NA	0	32	1990	199	1	1
1	NA	0	4	1990	175	1	1
1	NA	0	21	1990	175	1	1
1	NA	0	36	1990	175	1	1
	NA	0	19	1990	NA	Blank	Blank
1	NA	0	41	1990	175	1	1

FIGURE 4.24 Newly created columns by running the given R code scripts.

CASE STUDY 4.3: ADDITIONAL SQL QUERIES

Sometimes our data sets are stored in the data warehouse, and SQL will be an excellent choice to extract well-shaped data from the warehouse. In this study, we will use SQL to extract data.

Use the SQL interpreter, found at the following site:

<https://www.sqltutorial.org/seeit/>

Use the seven tables already loaded, found at the following site:

<https://www.sqltutorial.org/sql-sample-database/>

We need to address the following requirements:

Extract a table with employee names, IDs, emails, hire date, salary, department name, job title, region, country name, city, state, and country ID for all employees who work in Europe.

To perform this exercise, use the query shown in Figure 4.25.

```
SELECT first_name, last_name, employee_id, email,
hire_date, Salary, department_name, job_title, city,
state_province AS state, country_name, region_name
FROM employees AS E
JOIN departments AS D ON E.department_id =
D.department_id
JOIN jobs AS J ON E.job_id = J.job_id
JOIN locations AS L ON D.location_id = L.location_id
JOIN countries AS C ON L.country_id = C.country_id
JOIN regions AS R ON C.region_id = R.region_id
WHERE region_name = 'Europe';
```

FIGURE 4.25 Input SQL query into the interpreter.

Figure 4.26 shows the query input to the interpreter and a portion for the resulting table displayed by the query tool.

The screenshot shows an SQL query interpreter interface. The top section is labeled "SQL QUERY" and contains the following SQL code:

```
1 SELECT first_name, last_name, employee_id, email, hire_date, Salary, department_name, job_title, city, state_province AS state, cour
2 FROM employees AS E
3 JOIN departments AS D ON E.department_id = D.department_id
4 JOIN jobs AS J ON E.job_id = J.job_id
5 JOIN locations AS L ON D.location_id = L.location_id
6 JOIN countries AS C ON L.country_id = C.country_id
7 JOIN regions AS R ON C.region_id = R.region_id
8 WHERE region_name = 'Europe';
9
```

Below the query input, there are buttons for "Execute", "Clear", "Beautify", "Minify", and "Reload". The bottom section is labeled "RESULT" and displays a table with the following data:

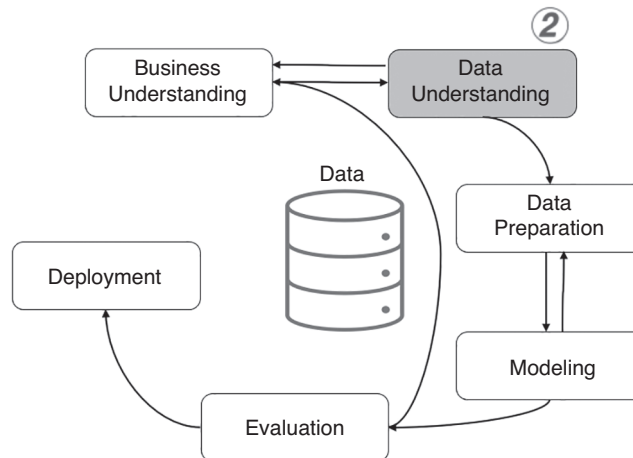
first_name	last_name	employee_id	email	hire_date	salary	department_name	job_title
John	Russell	145	john.russell@sqltutorial.org	1996-10-01	14000	Sales	Sales Manager
Karen	Partners	146	karen.partners@sqltutorial.org	1997-01-05	13500	Sales	Sales Manager
Jonathon	Taylor	176	jonathon.taylor@sqltutorial.org	1998-03-24	8600	Sales	Sales Representative

FIGURE 4.26 Output after running the SQL query.

REFERENCE

CrowdFlower 2016 Data Science Report.” 2016. CrowdFlower. http://visit.crowdflower.com/rs/416-ZBE-142/images/CrowdFlower_DataScienceReport_2016.pdf.

DESCRIPTIVE ANALYSIS



Descriptive analysis helps us understand data. We summarize the data using descriptive statistical techniques. In the process, we also obtain answers to business questions dealing with the past and the present (What happened?). In this way, it differs from predictive analytics, which deals with the future (What will happen next? What might happen?). These latter questions are more appropriate for data mining. We may ask additional questions that deal with the assurance of results, referred to as *inferential statistics* (Are we sure, or is this the result of some random event?). We will employ descriptive statistics here in this chapter to help us understand our data as a summarization of numerical (averages, sums, and extremes) or categorical variables (tabulation). We will tackle predictive analytics throughout the rest of the book. We will leave inferential analysis to be explored in other texts. Descriptive analysis also differs from summarizing text data (What are people saying?). We examine text data analysis in Chapter 13.

Descriptive analysis techniques are also useful to answer business questions such as “How many are there?,” “How much?,” and “How do they compare?”

There are a number of tools for descriptive analysis: (1) five-point summaries (available in the Excel ToolPak); (2) R tools for quick data discovery, such as Jamovi or JASP; and (3) Python menu-driven tools for data exploration, such as Orange3. We will explore data using summarization tools, such as quartiles and averages, and medians, maximum, minimum, and measures of the spread of the data, such as variances and interquartile ranges. This chapter also introduces two other useful tools: (1) the creation of boxplots as a summarization and visualization tool for numeric variables; and (2) the use of tabulation tools (such as pivot tables in Excel) and categorical variables, and the creation of contingency tables (crosstabs). We will also use histograms as a tool to get a sense of the distribution of numeric variables.

In this chapter, we demonstrate basic techniques using exercises. We challenge the reader to apply techniques presented using the more advanced situations posed in the case studies. In the cases, we continue to apply the principles of data understanding using the case studies of the SFO Survey and the SBA Loan data sets.

GETTING A SENSE OF THE DATA SET

The CRISP-DM process model (shown in Figure 5.1) tells us that before we dive into using algorithms to produce models for prediction and insightful answers, we need to get a good sense of our data. We investigate each of the variables individually and in combination to understand them. We can use descriptive statistics to investigate the nature of the variables. As we discovered in Chapter 1, if all our business questions can be answered with descriptive statistics, we would not need data mining (or business analysis, for that matter). We are pursuing the use of business analytic techniques to give us additional insights from data, including predictive analysis, for example, which leads to data mining.

As the CRISP-DM process model informs us (Wirth 2000);

“The data understanding phase starts with initial data collection and proceeds with activities that enable you to become familiar with the data, identify data quality

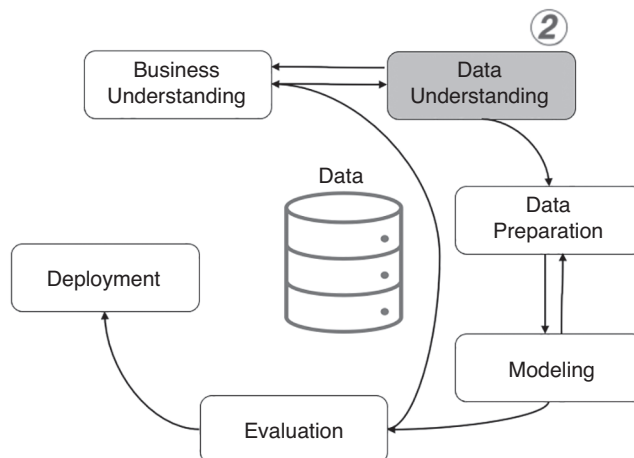


FIGURE 5.1 The CRISP-DM process model highlighted with the data understanding phase and its relationship to the other phases.

problems, discover first insights into the data, and detect interesting subsets to form hypotheses regarding hidden information.”

There are several activities and reports involved in completing the data understanding step of the process model as shown in Figure 5.2.

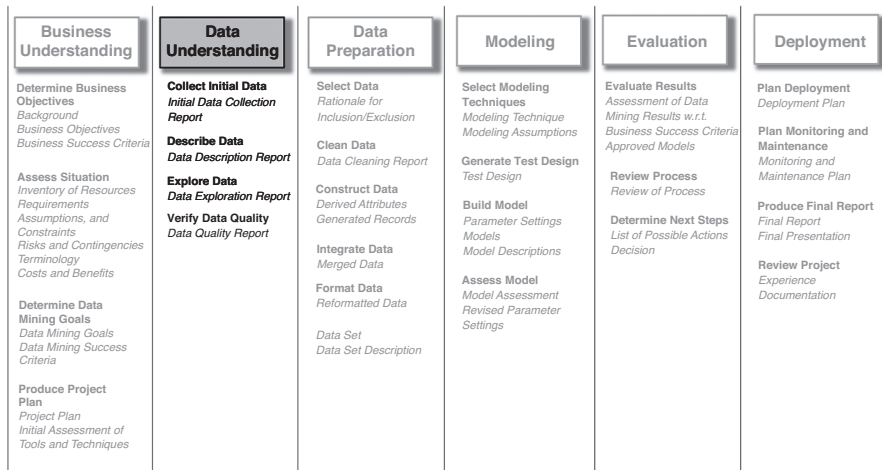


FIGURE 5.2 The CRISP-DM process model: the data understanding activities and reports are highlighted.

Describe the Data Set

In the words of the model, we are to examine the “gross” or “surface” properties of the acquired data and report on the results. This Data Description Report would include a description of the acquired information, including the format, the quantity (for example, the number of records and fields in each table), the identities of the fields, and any other “surface” features of the data set. This is also a good place to evaluate whether the data acquired satisfies the relevant data mining requirements. In other words, we determine how suitable this data set is to answer the information needs and the framed analytical questions found in the Business Understanding step (Chapter 3).

Explore the Data Set

This task addresses data mining questions using querying, visualization, and reporting techniques. These include the distribution of key attributes (for example, the target attribute of a prediction task), relationships between pairs or small numbers of attributes, results of simple aggregations, properties of significant sub-populations, and simple statistical analyses. These analyses may directly address the data mining goals; they may also contribute to or refine the data description and quality reports and feed into the transformation and other data preparation steps needed for further analysis.

A result from this task might contain the first findings or initial hypothesis and their impact on the remainder of the project. If appropriate, include graphs and plots to indicate data characteristics that may suggest further examination of interesting data subsets.

Verify the Quality of the Data Set

In this Data Understanding step, we examine the quality of the data. We might address such questions as: Is the data complete (does it cover all the cases required)? Is it correct, or does it contain errors and, if there are errors, how common are they? Are there missing values in the data? If so, how are they represented, where do they occur, and how common are they? Our Data Quality Report might include the results of the data quality verification, if quality problems exist, with a list of possible solutions. Solutions to data quality problems generally depend heavily on both data and business knowledge.

ANALYSIS TECHNIQUES TO DESCRIBE THE VARIABLES

Exploratory data analysis is one expression used to describe the analysis of data sets to summarize their main characteristics. The exploration is often best done visually, so we create many graphs of the data. These are analysis graphs and are not meant for communications, so they are done quickly without regard to visual detail. Once they are created, we move on to more analysis. Later, if we wish to use this graph to communicate some findings to colleagues, it pays to spend some time to make the graph more compelling.

What are we interested in discovering? First, we are interested in the types of data (variables) we are dealing with (Is it numerical, categorical, time, or text?). The techniques needed to deal with each type of data are quite different, and it is good to be aware of our computation expectations. We need to see how the data is distributed, potentially identify outliers, and identify apparent patterns, if any exist. We look for any relationships between the variables. We will use previously enumerated techniques to do so: boxplots, tabulation, trend analysis of time series, correlation analysis, tabulations, and contingency analysis. Along the way, we may be able to answer simple framed questions about the business: summarizing the *what*, the *who*, the *how much*, and the *when* questions. The following exercises will exemplify how this is best done.

EXERCISE 5.1 - DESCRIPTIVE STATISTICS

Let's say we collected a few basic facts about a group of students at a local secondary school. We want to explore the collected data.

Using the *Case Data* set provided, open the *Chapter 5* folder, and find the file *BigClass.xlsx*. Open it using Excel.

We are going to answer these questions:

What is the correlation between age and weight and age and height for these students?

What is the distribution of height and weight by age?

We will create a CSV version of the data set, upload it to Jamovi, and perform some exploratory computations.

Distributions of Numeric Variables

For a first pass, let's tabulate a summary of the numeric variables *Age*, *Height*, and *Weight*, and create histograms for them, as shown in Figure 5.3.



FIGURE 5.3 Summaries and histograms of the three numeric variables *Age*, *Height*, and *Weight* in the *Bigclass* data set produced using Jamovi.

Use Jamovi to create a boxplot of all three variables, *Age*, *Height*, and *Weight*. This gives a quick picture of the distribution of the variables (shown in Figure 5.4). Note that the boxplot form used by Jamovi displays a few outliers at the low end of the spectrum for height and weight.

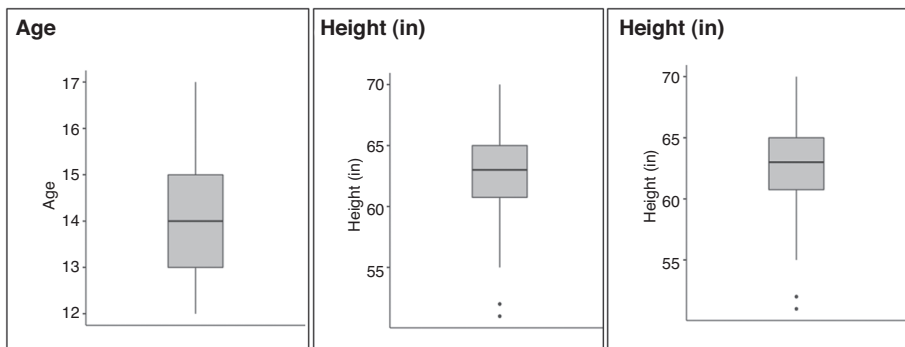


FIGURE 5.4 Boxplots of the three numeric variables *Age*, *Height*, and *Weight* in the *Bigclass* data set produced using Jamovi.

Correlation

We can see if there is a relationship between these three variables by computing the correlation matrix. We see that there is a strong positive correlation between these three variables. Figure 5.5 shows the result of the analysis.

If you are wondering how we interpret correlations, Figure 5.6 shows a chart that might help. This is how we speak about various levels of correlation. Be mindful these are not definite ranges, but approximate definitions.

We should also summarize the categorical variable *Gender* by tabulating it (Figure 5.7).

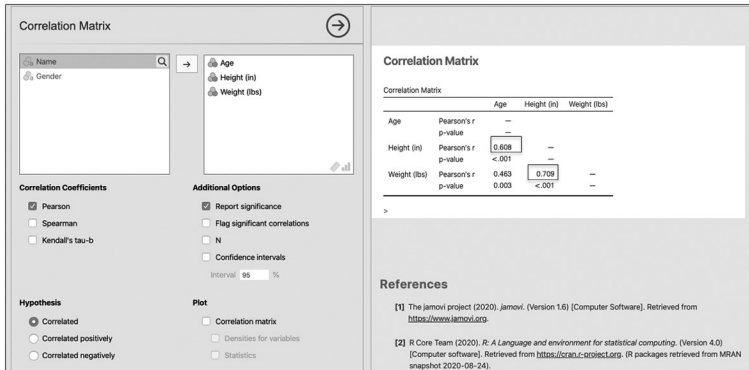


FIGURE 5.5 Correlation matrix of the three numeric variables *Age*, *Height*, and *Weight* in the *Bigclass* data set produced using Jamovi.

Correlation	Strength
1	Strongly positively correlated
0.7	
0.7	Positively correlated
0.4	
0.4	Positive weakly correlated
0.2	
0.2	Not correlated
0	
0	Not correlated
-0.2	
-0.2	Negatively weakly correlated
-0.4	
-0.4	Negatively correlated
-0.7	
-0.7	Strongly negatively correlated
-1	

FIGURE 5.6 Typical interpretations of correlation value ranges.

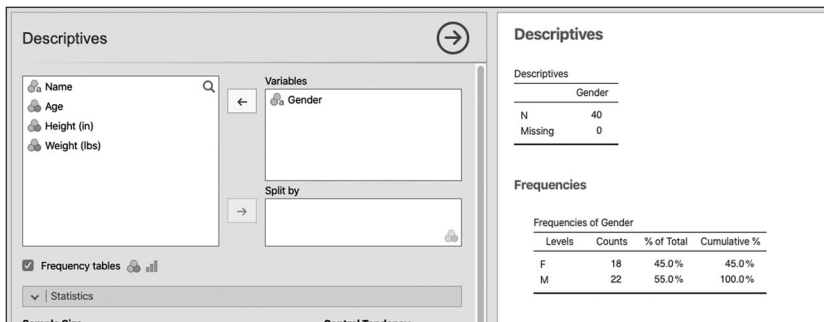


FIGURE 5.7 Tabulation of the *Gender* categorical variable.

We can now use some of the categorical variables, such as *Gender*, to split a numerical variable *Age* to see what the age distribution by gender might be (shown in Figure 5.8). Is there a difference in the average age of boys and girls in our group of students? There does not seem to be much difference.

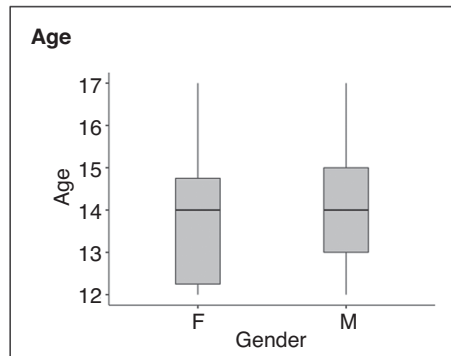


FIGURE 5.8 Boxplots of the *Age* split by the categorical variable *Gender*.

Age is a wonderful example of a variable that can be treated as a number and a category. We can ask the average age of our group and how many children can be in each age category, which leads to some insightful answers. Figure 5.9 shows the progression of height and weight of male and female children as they age. We added a linear trend line to the boy's data and a power trend line to the girl's data. For this sample of students, it seems that female height seems to start leveling off as they age into the teenage years. For the boys, it seems to climb into adulthood. Figure 5.9 shows the resulting final graphs.

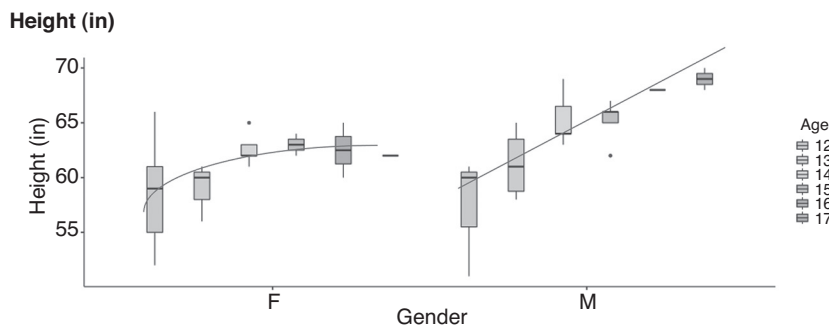


FIGURE 5.9 Boxplots of the *Height* split by the categorical variable *Age* and *Gender*, demonstrating that girls' heights seem to level off as they grow past their teenage years, but boys' heights seem to keep increasing into adulthood.

EXERCISE 5.2 – DESCRIPTIVE ANALYSIS OF THE TITANIC DISASTER DATA

Using the *Case Data* set and in the *Data Preparation Files* folder, find the file *Titanic.xlsx*. Open *Titanic.csv* using Jamovi.

We are going to answer these questions:

Were male passengers older or younger, on average, than female passengers?

What are the descriptive statistics for each gender? Compare them.

Repeat for each class of passengers and compare.

Which gender stood a greater chance of survival? Was there a difference by class?

Create a summary by age. Perform a tabulation by age and enter the maximum age for males and females. Do a sub-summary under each gender of the passenger's name. Now you have two lists, one above the other, of passenger ages sorted by gender, as in Figure 5.10.

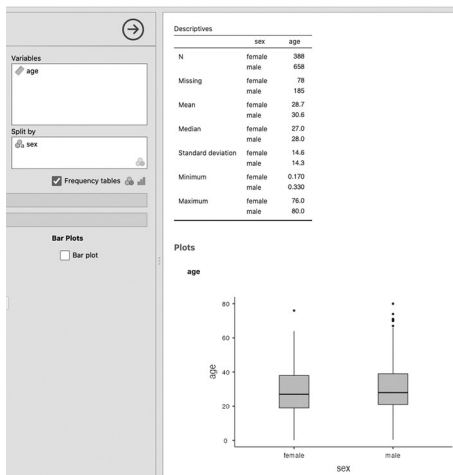


FIGURE 5.10 Summary of the Age of passengers by Gender.

We obtain similar results, but they are now further split by passenger class. Obtain the descriptive statistics of age by gender (labeled sex in the file) and create a box plot diagram (shown in Figure 5.11). We see that, in general, the males were, on average, older than females for the class types.

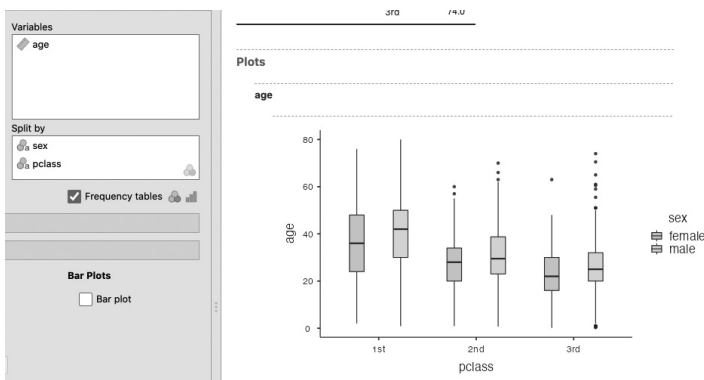


FIGURE 5.11 Final boxplots summarizing passengers' ages, compared by gender and by class.

Finally, we are close to the most important question: survival. By creating a contingency table of survival by gender and further split by class (shown in Figure 5.12), we see that women did survive at a higher rate than men, and first- and second-class women passengers were surviving at a much higher rate than third-class women passengers.

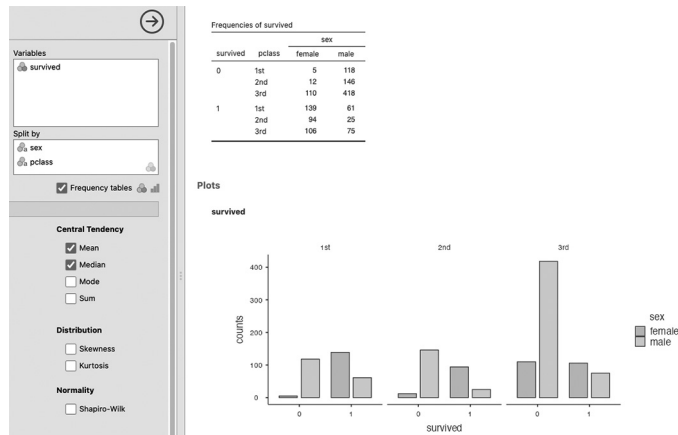


FIGURE 5.12 Contingency table and plot summarizing passenger survival by gender and class.

CASE STUDY 5.1: DESCRIBING THE SFO SURVEY DATA SET

We shall explore some of the main variables associated with the SFO Airport Survey. We intend to analyze the variables to see their distributions and tabulations and perhaps do some cross-tabulation to explore relationships between them. Use the data dictionary file to read the description for the variables, noting their units, possible occurrences, and their definitions and limits. For this case study, we will limit ourselves to the 2018 survey data. Using Excel, open the *SFO 2018 Survey Data.xlsx* file found in the *Case Data* depository in the *SFO Survey Data* folder.

Since the population being surveyed here are passengers that use the airport, it stands to reason we want to get the demographics of the passengers.

We will attempt to address the following:

What is the breakdown of the survey respondents by gender? By age ranges? By income? (Tabulation)

What is the percentage of frequent flier passengers to all passengers found in this sample? What about business fliers versus economy class fliers? (Tabulation)

Perform crosstabulations of income level by gender and by age and flying class by gender and by age.

What are the top destination cities for those originating their flights in San Francisco?

SOLUTION USING R

To answer the first question, we will use Jamovi. Once the data has been successfully cleaned and prepared, import the data set into Jamovi for further analysis (convert it to a CSV file first).

To describe the SFO Survey data set, we can use the descriptive function under exploration. The steps and results are shown in Figures 5.13 and 5.14.

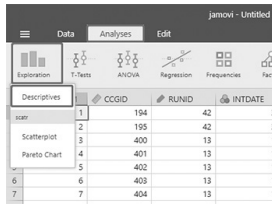


FIGURE 5.13 Use the *Descriptive* function to create descriptive analyses in Jamovi.

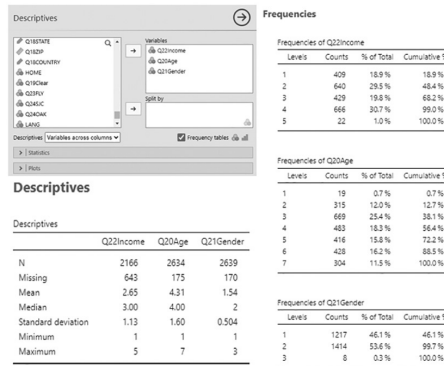


FIGURE 5.14 Descriptive output for the SFO data set.

To answer the second question, we will use the variable *Q23FLY* and define passengers who fly 100,000 miles or more as frequent fliers, as shown in Figure 5.15. Be sure to look up the codes for *Q23FLY* in the data dictionary *code list* tab in the spreadsheet.

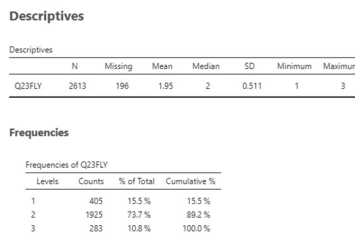


FIGURE 5.15 Descriptive analyses on variable *Q23FLY*.

We also will define *business/work/job interview* and *conference/convention* coded rows as business fliers and the rest as non-business fliers. We should transform the existing variable *Q2PURPI* into a new variable with the condition we are asked to solve. The steps and results are shown in Figure 5.16.

We will use the same technique and add a *split* function for this question, as shown in Figure 5.17.

Similar to the first question, before we can use the *descriptive* function under exploration, we need to change the data type of the variable *DESTINATION* into Nominal or Ordinal, as shown in Figure 5.18.

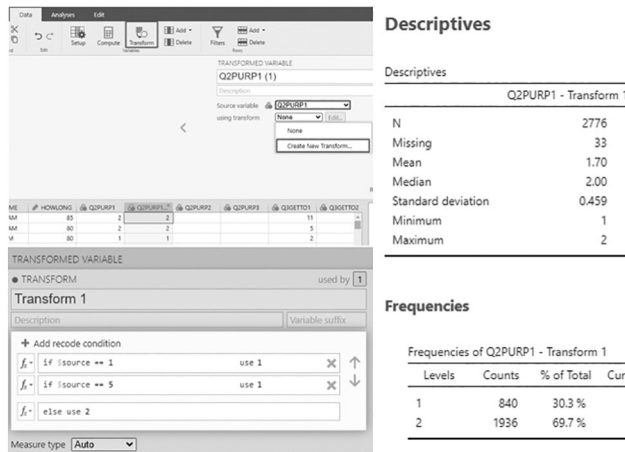


FIGURE 5.16 Steps to create a new transform column and descriptive analyses.

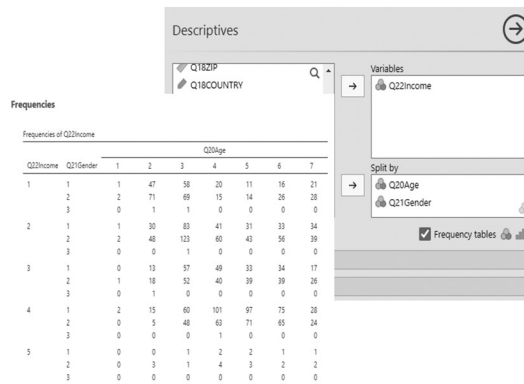


FIGURE 5.17 Create frequency analyses on Income and split by Age and Gender.

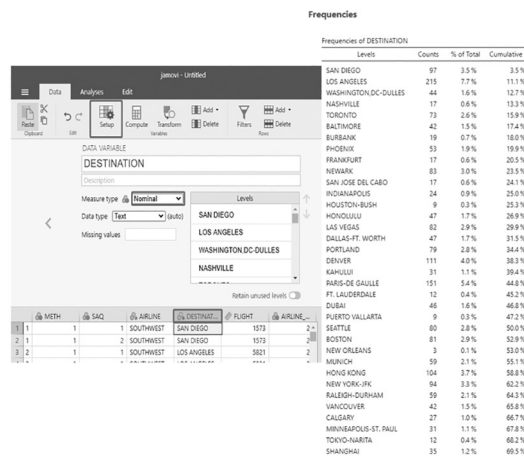


FIGURE 5.18 Create frequency analyses on DESTINATION.

Note that since there is no option to sort the frequency table in Jamovi, we can install the module called Rj, and run the R command directly to get a more readable result. The installation is shown in Figure 5.19.

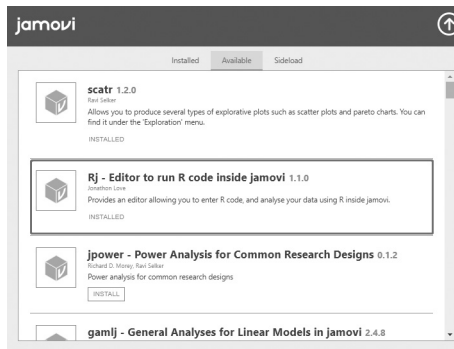


FIGURE 5.19 R package in the Jamovi module.

Once you successfully install the module in Jamovi, run the following commands, and you should get the same frequency table (but with a decreasing order by a count of each destination). The result is shown in Figure 5.20.

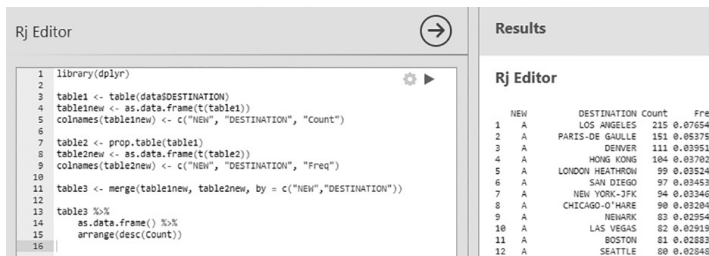


FIGURE 5.20 Frequency table with decreasing order from R.

```
library(dplyr)
table1 <- table(data$DESTINATION)
table1new <- as.data.frame(t(table1))
colnames(table1new) <- c("NEW", "DESTINATION", "Count")
table2 <- prop.table(table1)
table2new <- as.data.frame(t(table2))
colnames(table2new) <- c("NEW", "DESTINATION", "Freq")
table3 <- merge(table1new, table2new, by = c("NEW", "DESTINATION"))
table3 %>%
  as.data.frame() %>%
  arrange(desc(Count))
```

SOLUTION USING PYTHON

Create a new file in Orange3, and import the SFO data set into the file. Make sure to change the type of variables that we want to analyze later, as shown in Figure 5.21.

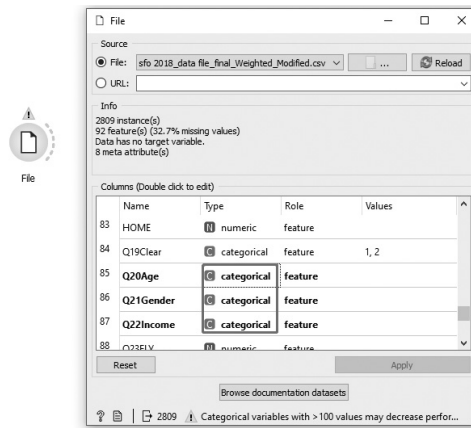


FIGURE 5.21 Editing data type in Orange3.

Then create a new widget to select columns for future analysis, and connect this widget with the file, as shown in Figure 5.22.

Descriptives

Descriptives			
	JobsSupportedCompanySize	GrossApproval - money was disbursed	
N	1		141668
	2		8381
	3		427
Missing	1		31080
	2		2041
	3		117

FIGURE 5.22 Select feature variables.

Create a new widget to build a pivot table, and connect this widget with the selected columns' widget. The steps and results are shown in Figure 5.23(a), (b), and (c).

		Q20Age								
		Count	1	2	3	4	5	6	7	Total
Q20Age	1	19.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	19.0
	2	0.0	315.0	0.0	0.0	0.0	0.0	0.0	0.0	315.0
	3	0.0	0.0	669.0	0.0	0.0	0.0	0.0	0.0	669.0
	4	0.0	0.0	0.0	483.0	0.0	0.0	0.0	0.0	483.0
	5	0.0	0.0	0.0	0.0	416.0	0.0	0.0	0.0	416.0
	6	0.0	0.0	0.0	0.0	0.0	428.0	0.0	0.0	428.0
	7	0.0	0.0	0.0	0.0	0.0	0.0	304.0	0.0	304.0
Total		19.0	315.0	669.0	483.0	416.0	428.0	304.0		2634.0

FIGURE 5.23(a) Create pivot tables of the feature columns.

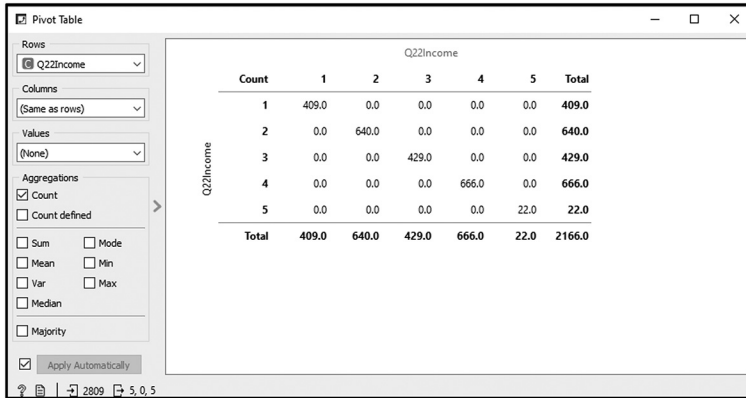


FIGURE 5.23(b) Create pivot tables of the feature columns.

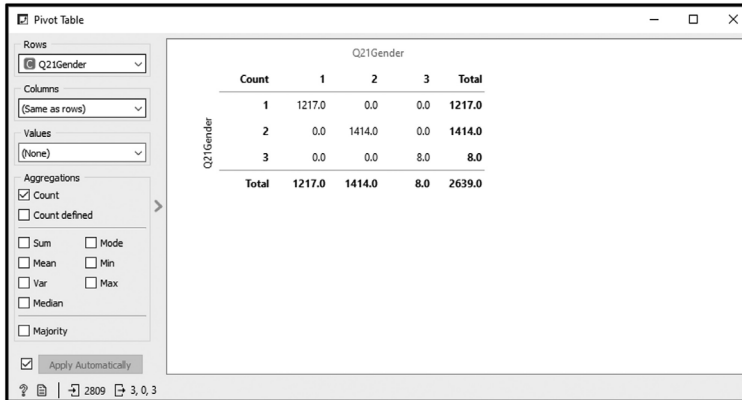


FIGURE 5.23(c) Create pivot tables of the feature columns.

We select the column *Q23FLY* and define that passengers who fly 100,000 miles or more as frequent fliers, as shown in Figure 5.24.

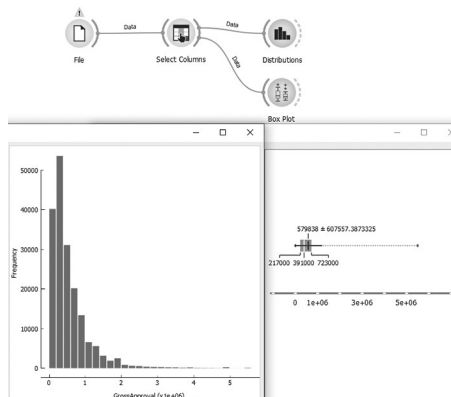


FIGURE 5.24 Frequency table of frequent customers.

We also define those traveling for *business/work/job interview* and *conference/convention* as business fliers and the rest as non-business fliers. We should transform the existing column *Q2PURP1* into a new variable with the condition we are asked to solve. To do so, we need to go back to Excel or R and reshape the data set. Once you have the modified data ready, follow similar steps; you should have the frequency table shown in Figure 5.25.

		business		
Count		1	2	Total
business	1	840.0	0.0	840.0
	2	0.0	1969.0	1969.0
	Total	840.0	1969.0	2809.0

FIGURE 5.25 Frequency table of business customers.

Since there is no option to create cross-tabulation by more than two variables, we will create tables one by one. Follow similar steps, and you should be able to create tables like those in Figure 5.26(a)-(d).

Redo all the above steps and select *DESTINATION* as the column we need to analyze, and you should be able to create a table like that in Figure 5.27.

		Q21Gender			
Count		1	2	3	Total
Q22Income	1	174.0	229.0	2.0	405.0
	2	255.0	376.0	1.0	632.0
	3	205.0	215.0	1.0	421.0
	4	381.0	278.0	2.0	661.0
	5	7.0	15.0	0.0	22.0
Total	1022.0	1113.0	6.0	2141.0	

FIGURE 5.26(a) Frequency tables of Gender and Income.

		Q21Gender			
Count		1	2	3	Total
business	1	474.0	316.0	3.0	793.0
	2	743.0	1098.0	5.0	1846.0
	Total	1217.0	1414.0	8.0	2639.0

FIGURE 5.26(b) Frequency tables of Gender and business.

		Q20Age							
Count		1	2	3	4	5	6	7	Total
Q22Income	1	3.0	120.0	128.0	35.0	26.0	42.0	51.0	405.0
	2	4.0	78.0	208.0	102.0	75.0	90.0	74.0	631.0
	3	1.0	33.0	109.0	90.0	74.0	74.0	43.0	424.0
	4	2.0	21.0	109.0	165.0	168.0	141.0	54.0	660.0
	5	0.0	3.0	2.0	6.0	5.0	3.0	3.0	22.0
Total	10.0	255.0	556.0	398.0	348.0	350.0	225.0	2142.0	

FIGURE 5.26(c) Frequency tables of Gender and Age.

		Q20Age							
business	Count	1	2	3	4	5	6	7	Total
	1	2.0	39.0	205.0	242.0	162.0	100.0	39.0	789.0
	2	17.0	276.0	464.0	241.0	254.0	328.0	265.0	1845.0
Total	19.0	315.0	669.0	483.0	416.0	428.0	304.0	2634.0	

FIGURE 5.26(d) Frequency tables of *business* and *Age*.

Count	AMSTE...	ATLANTA	AUCKL...	AUSTIN	BAKER...	BALTI...	BEIJING	BO
AMSTE...	8.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ATLANTA	0.0	23.0	0.0	0.0	0.0	0.0	0.0	0.0
AUCKL...	0.0	0.0	23.0	0.0	0.0	0.0	0.0	0.0
AUSTIN	0.0	0.0	0.0	34.0	0.0	0.0	0.0	0.0
BAKER...	0.0	0.0	0.0	0.0	6.0	0.0	0.0	0.0
BALTI...	0.0	0.0	0.0	0.0	0.0	42.0	0.0	0.0
BEIJING	0.0	0.0	0.0	0.0	0.0	0.0	23.0	0.0

FIGURE 5.27 Frequency tables of *DESTINATION*.

CASE STUDY 5.2: DESCRIBING THE SBA LOANS DATA SET

We shall explore some of the main variables associated with the SBA Loan data set. We intend to analyze the variables to see their distributions and tabulations and perhaps do some cross-tabulation to explore the relationships between them. Use the data dictionary file to read the description for the variables, noting their units, possible occurrences, and their definitions and limits.

Since the population being surveyed here are companies that received loans from the SBA, we want to get the demographics of the loan recipients. We will address the following:

What is the breakdown of loan recipients by state in the USA? (tabulation)

For loans that were processed and money was disbursed, what is the distribution by loan amount (histogram and boxplot) and by the size of the firm (number of employees) (histogram)?

Tabulate the total amount lent out by year (summarize and tabulate).

For those approved and money disbursed loans, tabulate the percentage of loans that were not repaid. Tabulate the exact percentage breakdown by year.

Create boxplots of the loan amount by paid-back and not paid-back status. What differences do you see?

Create a categorical variable, call it *Company Size*, and categorize all firms by the number of employees as *SMALL*, *MEDIUM*, and *LARGE*. Use reasonable ranges for all three categories. What is the total number of loans approved for each category?

SOLUTION USING R

Use the *FOIA Loans Data* data set found in the *Case Data* depository under *SBA Loans Data* folder. Open *FOIA Loans Data.csv* using Jamovi. Use the Setup tool under Data, and change the data type of the variable *BorrState* to Nominal for the summary statistics in the next step.

Apply the *Descriptive* function under Exploration in Analyses. Select the column *BorrState* as the variable and use the *Frequency tables* function to get a summary of this column, like that shown in Figure 5.28.

NOTE Run the following command, and you should get the same frequency table, but with decreasing order by the count of each State.

```
library(dplyr)
table1 <- table(data$BorrState)
table1new <- as.data.frame(t(table1))
```

Frequencies

Frequencies of BorrState			
Levels	Counts	% of Total	Cumulative %
NH	3307	1.8%	1.8%
AR	681	0.4%	2.2%
UT	6279	3.4%	5.6%
WA	3952	2.2%	7.7%
AL	2719	1.5%	9.2%
MI	4398	2.4%	11.6%
MN	7440	4.0%	15.7%
IN	4229	2.3%	18.0%
ID	2622	1.4%	19.4%
NY	7056	3.8%	23.2%
MO	3922	2.1%	25.4%
VA	3806	2.1%	27.4%
WI	4365	2.4%	29.8%
SD	1175	0.6%	30.5%
CA	36260	19.7%	50.2%
GA	4925	2.7%	52.9%

FIGURE 5.28 Frequency analyses of state.


```

colnames(table1new) <- c("NEW", "DESTINATION", "Count")
table2 <- prop.table(table1)
table2new <- as.data.frame(t(table2))
colnames(table2new) <- c("NEW", "DESTINATION", "Freq")
table3 <- merge(table1new, table2new, by = c("NEW", "DESTINATION"))
table3 %>%
  as.data.frame() %>%
  arrange(desc(Count))

```

To get a histogram and a box plot for the loan amount, we need to use the *Plots* function under *Descriptive*. Before that, we need to filter those loans that had been processed and money was disbursed. To do so, we need to create another column after *GrossApproval* and use the following transformation. The steps and results are shown in Figure 5.29(a)-(e).

For loans that were processed and money was disbursed, the distribution by size of the firm can be plotted by following the steps and creating a transformed variable *JobsSupportedCompanySize*. The steps and results are shown in Figure 5.30(a)-(d).



FIGURE 5.29(a) First step to create the histogram and box plot.

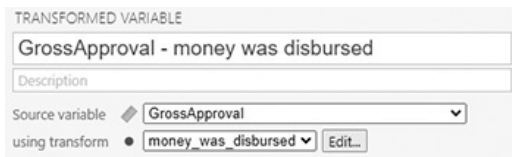


FIGURE 5.29(b) Second step to create the histogram and box plot.

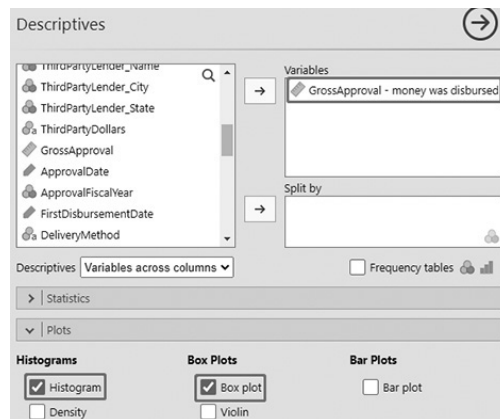


FIGURE 5.29(c) Third step to create the histogram and box plot.

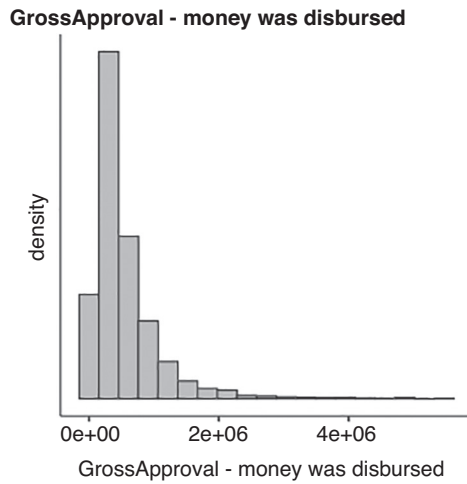


FIGURE 5.29(d) Fourth step to create the histogram and box plot.

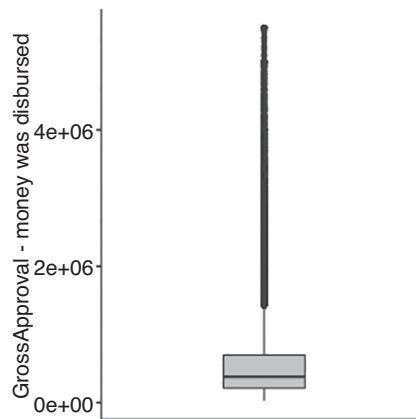


FIGURE 5.29(e) Fifth step to create a histogram and boxplot.

● TRANSFORM used by 1

CompanySize

CompanySize Variable suffix

+ Add recode condition

f_1	if :source > 200	use 3	×	↑
f_2	if :source < 50	use 1	×	↓
f_3	else use 2			

FIGURE 5.30(a) First step to create the histogram and box plot.

TRANSFORMED VARIABLE

JobsSupportedCompanySize

Description

Source variable \blacklozenge JobsSupported

using transform \bullet CompanySize Edit...

FIGURE 5.30(b) Second step to create the histogram and boxplots

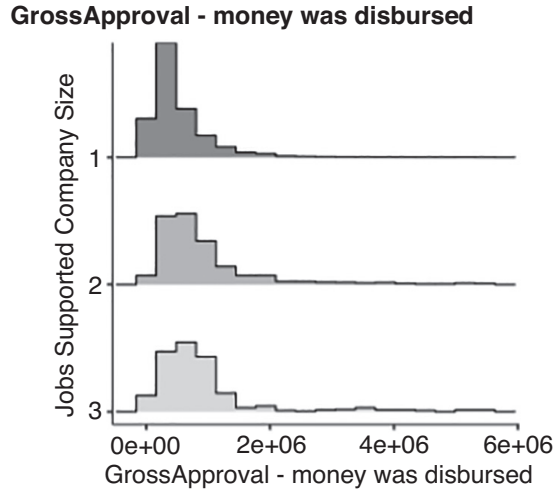


FIGURE 5.30(c) Histograms of the distributions by size of the firm.

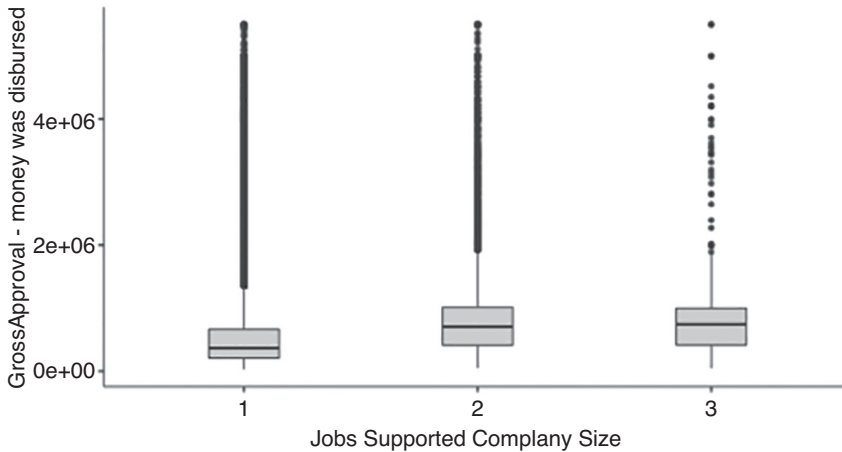


FIGURE 5.30(d) Boxplot distributions by size of the firm.

To tabulate the total amount lent out by year, we can describe the sum of the gross approval and split it by approval year, as shown in Figure 5.31.

To find the percentage of loans that were not repaid, we can use the variable *PAIDOFF* to get the result in Figure 5.32(a) and (b).

To answer the next question, we should plot the gross approval amount and split it by whether it was paid off or not, as shown in Figure 5.33.

In this case, as we have transformed before, we use 1 to represent a small company with less than or equal to 20 persons and use 3 to represent a large company that has greater than or equal to 500 persons. Finally, we use 2 to represent the rest of the companies. The result is shown in Figure 5.34.

Descriptives

Descriptives		
	ApprovalYear	GrossApproval - money was disbursed
Sum	1990	8.25e+7
	1991	4.38e+8
	1992	5.96e+8
	1993	8.20e+8
	1994	1.16e+9
	1995	1.41e+9
	1996	1.93e+9
	1997	1.32e+9
	1998	1.55e+9
	1999	1.61e+9
	2000	1.48e+9
	2001	2.04e+9
	2002	2.25e+9
	2003	2.74e+9
	2004	3.41e+9
	2005	4.68e+9
	2006	4.83e+9
	2007	5.42e+9
	2008	3.80e+9
	2009	3.47e+9
2010	3.93e+9	
2011	4.07e+9	
2012	5.93e+9	
2013	4.14e+9	
2014	3.54e+9	
2015	3.71e+9	
2016	3.94e+9	
2017	3.85e+9	
2018	3.11e+9	
2019	2.17e+9	
2020	1.35e+8	

FIGURE 5.31 Descriptive table of the gross approval money.

Frequencies

Frequencies of PAIDOFF			
Levels	Counts	% of Total	Cumulative %
0	11523	6.3 %	6.3 %
1	82821	45.1 %	51.4 %
Blank	89370	48.6 %	100.0 %

FIGURE 5.32(a) Frequency table of PAIDOFF and distribution by year.

Descriptives

Frequencies

Frequencies of PAIDOFF						
PAIDOFF	1990	1991	1992	1993	1994	1995
0	14	53	65	59	130	152
1	263	1381	1799	2334	3224	3812
Blank	44	211	246	389	593	712

FIGURE 5.32(b) Frequency table of PAIDOFF and distribution by year.

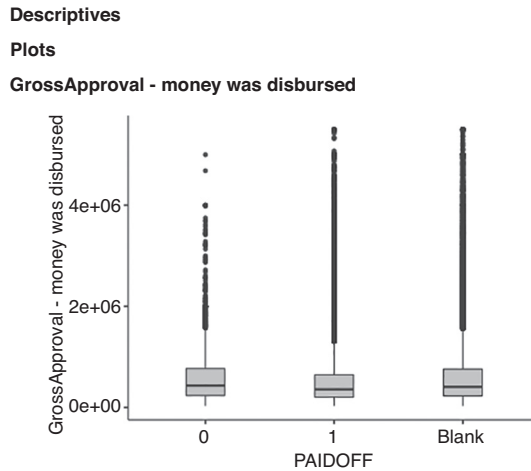


FIGURE 5.33 Boxplots of gross approval money by PAIDOFF.

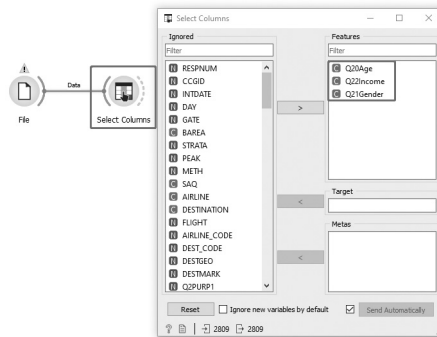


FIGURE 5.34 Descriptive table of gross approval money by firm size.

SOLUTION USING PYTHON

Use the *FOIA Loans Data* data set found in the *Case Data* depository under *SBA Loans Data* folder. Open *FOIA Loans Data.csv* using Orange3.

Select the variable *BorrState* only and show the variable in the pivot table as shown in Figure 5.35.

To find the distribution by loan amount, we need to go back to Excel or R and reshape the data set to filter out those approved, but never disbursed. Once you have the data ready, follow the aforementioned steps. You should be able to create the graphs shown in Figure 5.36.

For loans that were processed and money was disbursed, the distribution by the size of the firm can be plotted by following similar steps. You must go back to Excel or R, and create a new categorical variable *JobsSupportedCompanySize*. Follow the same steps, and you will get the result shown in Figure 5.37.

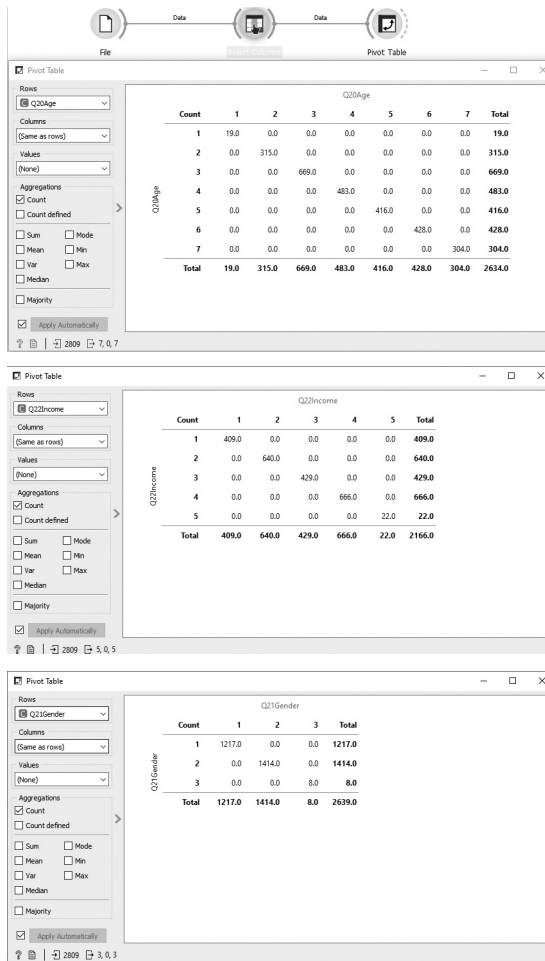


FIGURE 5.35 Steps to create a pivot table for states.

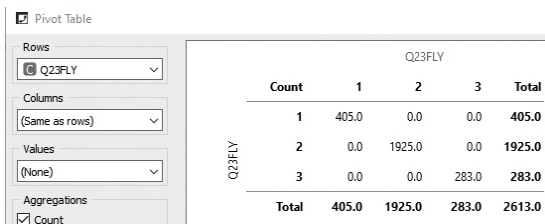


FIGURE 5.36 Steps to create the histogram and box plot.

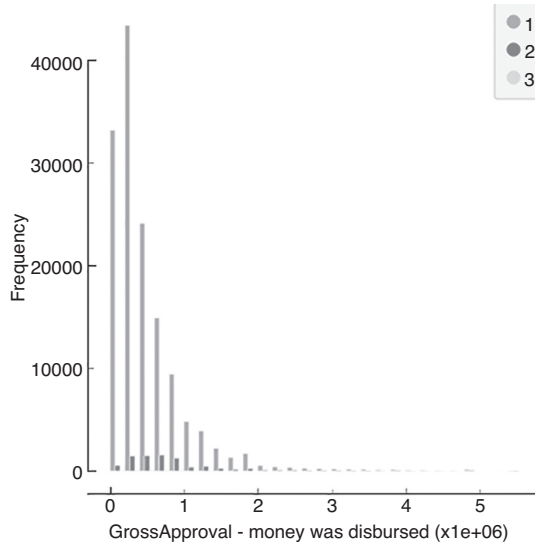


FIGURE 5.37 Frequency analyses on gross approval money by firm size.

To tabulate the total amount lent out by year, we can describe the sum of gross approval and split it by approval year by using a pivot table, as shown in Figure 5.38.

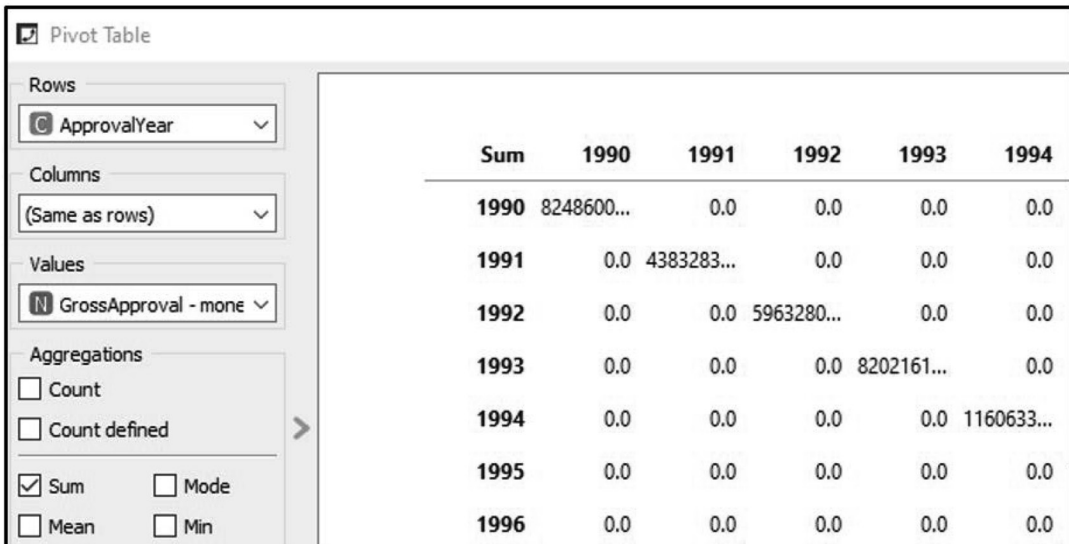


FIGURE 5.38 Pivot table of the year and approval money.

To find the percentage of loans that were not repaid, we can use the variable *PAIDOFF* to get the results shown in Figure 5.39(a) and (b).

To answer the next question, we should plot the gross approval amount first, and then split it by whether it was paid off or not. Steps and results are shown in Figure 5.40.

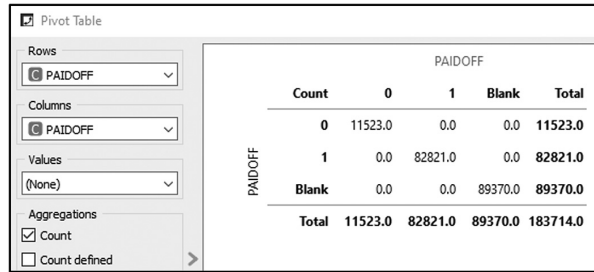


FIGURE 5.39(a) Frequency analysis on *PAIDOFF* and distribution by year

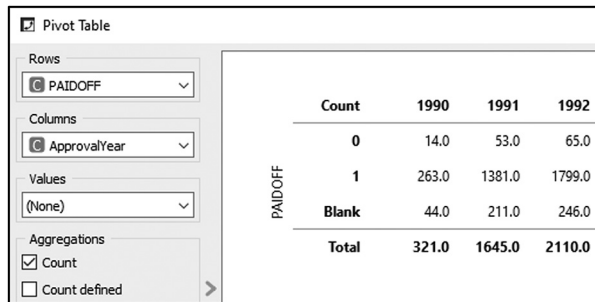


FIGURE 5.39(b) Frequency analysis on *PAIDOFF* and distribution by year

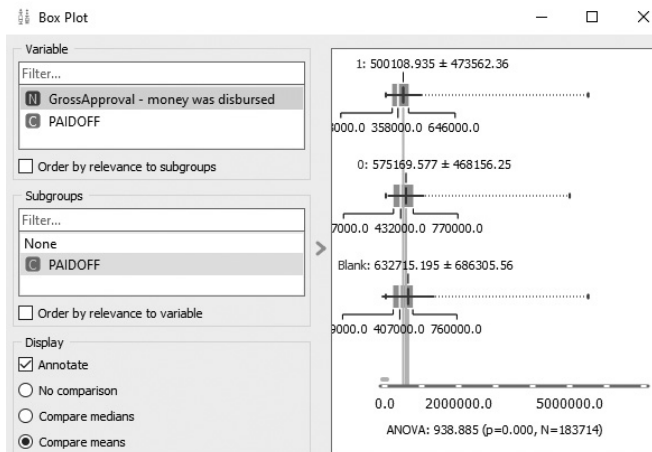


FIGURE 5.40 Box plot for approval money and distribution by *PAIDOFF*.

Here, we use 1 to represent a small company with less than or equal to 20 persons and use 3 to represent a large company with greater than or equal to 500 persons. Finally, we use 2 to represent the rest of the companies, as shown in Figure 5.41.

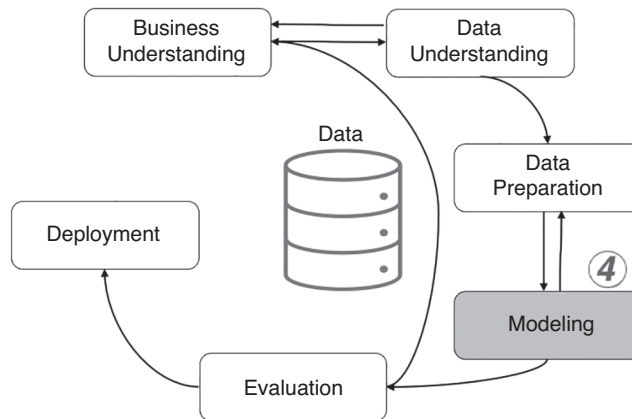
		JobsSupportedCompanySize				
		Count	1	2	3	Total
JobsSupportedCompanySize	1	172748.0	0.0	0.0		172748.0
	2	0.0	10422.0	0.0		10422.0
	3	0.0	0.0	544.0		544.0
	Total	172748.0	10422.0	544.0		183714.0

FIGURE 5.41 Frequency analysis on firm size

REFERENCE

Wirth, R., & Hipp, J. (2000, April). CRISP-DM: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* (Vol. 1, pp. 29-39).

MODELING



The fourth step in the CRISP-DM process is to create models using data. In this chapter, we tackle important modeling questions before plunging into detailed modeling techniques and algorithms.

What is a model? In particular, how do we create a numerical model of a business reality and use it to answer current and ongoing business questions with data? We are also interested in exploring how data mining and data modeling integrate. How does the creation of data models fit in the data mining process? We will explore the activities involved in modeling and how modeling relates to the ubiquitous concept of machine learning.

We will discuss how to check that the model is valid and produces accurate results. This chapter lays the foundation for subsequent work on the various algorithms we will use in our model building.

WHAT IS A MODEL?

In this context, we shall define it as an *analytics model* — the selection of a mathematical algorithm and computing the algorithm's coefficients to conform to a known data set. The process is termed *training* or *machine* (or algorithm) *learning*. The resulting computational tool is

no longer generic, but specific to the real-world environment generating that data type. Say, for example, you are tracking the behavior of customers in your business. You collect observations of their transactions, such as which product they might buy. You may identify which customers abandoned your service (such as is done with the customers of telephone service providers or credit card companies, for example). You also collect customer demographic data (age, income, education, and addresses). Using this collected data, you want to know if you can predict whether a customer will keep or use your service, buy your product, or repay a loan. Using past data to train a generic predictive algorithm, we produce a model (the algorithm with specific coefficients appropriate to your data) which the business can use for that purpose. We have produced a model of customer behavior that approximates the reality of their real behavior. It is flawed in that it does not always make accurate predictions. However, if we select the right algorithm and are careful in building the model, it can be a good tool for business decision-making, even though it only approximates the reality of customer behavior in all circumstances. This chapter discusses the steps for creating such a model from algorithmic choices and the available data set.

HOW DOES CRISP-DM DEFINE MODELING?

Once we have analyzed the business situation and information needs driving our project, we build a data-driven model to answer the framed analytical questions. The CRISP-DM (Chapman 2000, page 14) process defines the modeling step as follows:

“Modeling - In this phase, various modeling techniques are selected and applied, and their parameters are calibrated to optimal values. Typically, there are several techniques for the same data mining problem type. Some techniques have specific requirements in the form of data. Therefore, going back to the data preparation phase is often necessary.”

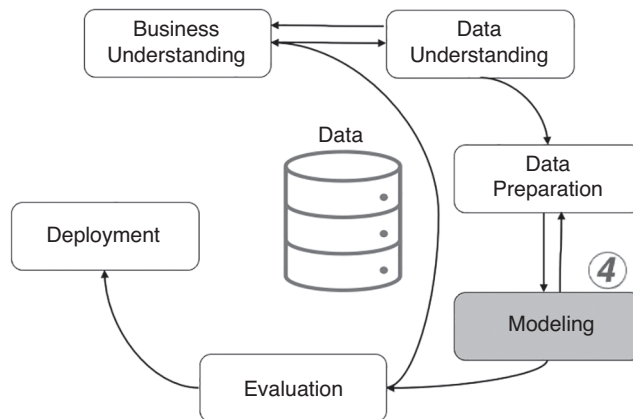


FIGURE 6.1 The CRISP-DM process model with the modeling step highlighted and its relationship to the other steps in the process (Chapman 2000).

The modeling step is the heart of the data mining process. This is where the real work of data mining occurs. Everything that occurs before is simply planning and preparation. What comes after is the application to the business. The real work by the algorithms to produce an answer to our business question happens in Step 4, the modeling step.

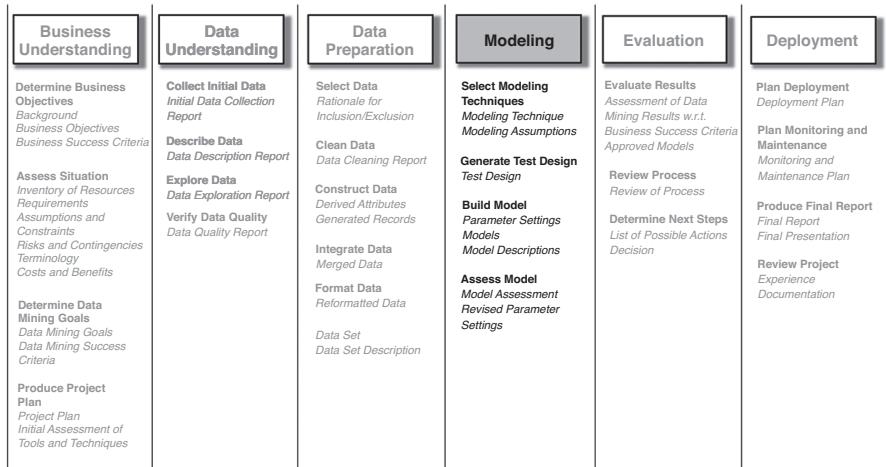


FIGURE 6.2 The CRISP-DM process model with the modeling step deliverables and activities highlighted.

The CRISP-DM process definition informs us of the activities and deliverables that occur in this step (see the Modeling step in Figure 6.2).

In this step, we decide on the best models to develop to answer the framed questions. We also decide which machine learning algorithms to use. We may even select several algorithms, construct models, and evaluate the results to see which one yields the most accurate answers. From the data set we use to train the machine, we set aside some of the data for testing to ensure our model is not overfitting (*overfitting* means that we get accurate results with training data, but a poor fit when we try to predict for the testing data). We may set up the model for it to continue to be updated as new data becomes available in the future (the machine continues to learn).

Selecting the Modeling Technique

The first step in modeling is to select the actual modeling technique to be used. Although you may have already selected a specific tool during the Business Understanding phase, this task refers to the specific modeling technique, e.g., building a decision-tree or building a neural network generation model with backpropagation. If multiple techniques are applied, we perform this task separately for each technique. We then document the actual modeling technique that is to be used.

Modeling Assumptions

Many modeling techniques make specific assumptions about the data, for example, all attributes have uniform distributions, no missing values allowed, and the class attribute must be symbolic. Record any such assumptions made.

Generate Test Design

Before we build a model, we need to generate a procedure or mechanism to test the model's quality and validity. For example, in supervised data mining tasks such as classification, it is common to use error rates as quality measures for data mining models. Therefore, we typically separate the data set into train and test sets, build the model on the train set, and estimate its quality on the separate test set. We need to design and document a plan for how this will be implemented.

Design of Model Testing

Describe the intended plan for training, testing, and evaluating the models. A primary component of the plan is determining how to divide the available data set into training, test, and validation data sets.

Build the Model

Typically, in model making, there are two algorithms in play. The first algorithm is the one to be used for computing answers to business questions and it is the one to be trained. The other algorithm, or procedure, is the method used to train the first algorithm. For example, in linear regression, the model is the equation of a line, but the regression process is the one used to instantiate the coefficients of the line based on the training dataset set. The model-making algorithms then use the modeling tool on the prepared data set to create one or more models. This transforms the generic algorithms, which are the basis of our model, into the model. The algorithm transforms from a mathematical expression with generic coefficients to an instantiation with actual coefficients that match our data set.

Parameter Setting

With any modeling tool, there are often a large number of parameters that can be adjusted. List the parameters and their chosen values, along with the rationale for the choice of parameter settings. We term these *hyperparameters*, those parameters of the model assigned ad hoc by the analysis, such as the number of repetitions before we consider the model done or the number of tree branches for a decision-tree algorithm. These hyperparameters should be vetted by the analysis team and documented along with other model elements.

Models

These are the actual models produced by the modeling tool. They should be described in some detail. We should also report on the interpretation of the models and document any difficulties encountered with their meanings.

Model Assessment

The data mining analyst interprets the models according to his domain knowledge, the data mining success criteria, and the desired test design. The data mining analyst judges the success of the application of modeling and discovery techniques technically; the analyst contacts business analysts and domain experts after model building to discuss the data mining results in the business context. Please note that this task only considers models, whereas the evaluation phase also considers all other results that were produced in the course of the project.

The data mining analyst should rank the models built of different algorithmic approaches. He assesses the models according to the evaluation criteria. As much as possible, the analyst also takes into account business objectives and business success criteria. In most data mining projects, the data mining analyst applies a single technique more than once or generates data mining results with several different techniques. In this task, all results are compared according to the evaluation criteria.

Once the models are compared, a summary of the results is prepared listing qualities of generated models (i.e., in terms of accuracy) and rank their quality in relation to each

other. At this point, parameter settings are appropriately revised and tuned for the next run in the Build Model task. Iterate model building and assessment until you strongly believe that you have found the best model(s). All such revisions and assessments need documentation.

WHERE DO MODELS RESIDE IN A COMPUTER?

Typically, an organization will use a set of software programs to support its operations, financial reporting, and strategic planning. These may collectively be known as Enterprise Resources Planning (ERP) tools. ERP tools are a type of software that organizations use to manage day-to-day business activities, such as accounting, procurement, project management, risk management and compliance, and supply chain operations. There are financial tracking and planning tools (the accounting system) and Customer Relationship Management (CRM) tools. The data mining environment stands parallel to this set of software. It uses the data from transactions recorded by such systems, along with data accumulated in data warehouse repositories (also extracted from such operational systems). The data mining models are constructed in sophisticated data mining tools environments. Some of the models are constructed directly using programming environments, such as Python and R (as discussed in Chapter 1). Very sophisticated commercial settings, such as SAS Enterprise Guide, Rapid Miner, IBM Intelligent Miner, and several others, provide their unique platform to construct models.

The Data Mining Engine

Figure 6.3 shows a schematic of the arrangement. The central cloud construct represents this data mining environment, whether as a basis for programming or as a commercial tool. It could be “in the cloud,” i.e., accessible over public networks, or on some internal organizational server accessible over the corporate intranet. The data mining programming environment provides links to data sources, whether residing statically as databases on some network disk drive, via Structured Query Language (SQL) queries from a traditional data repository, or ingested in real-time form some social network feed, such as Facebook or Twitter. These are accessed by programming an Application Programming Interface (API) between the tool and the data source.

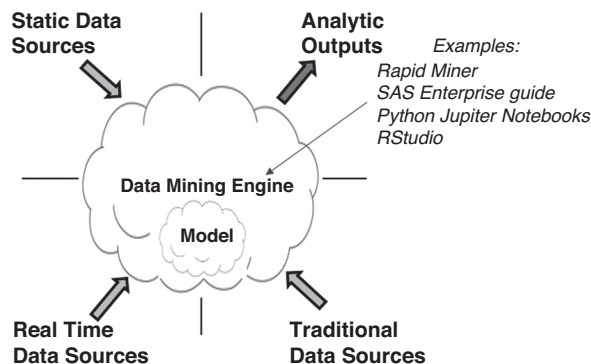


FIGURE 6.3 The Data Mining Engine, Data Models, and Data Sources and their relationships.

The Model

The model is then built by programming using the language (such as Python or R) or the unique programming tools that are part of the tool (such as SAS EG, RapidMiner, and IBM IM). It can be constructed ad hoc using these tools. Once perfected, the model can then be deployed within the tool with input and output User Interfaces (UI) properly designed to provide access and answers to non-technical users. Figure 6.3 shows the model as some component of the data mining engine running within it to provide the computing power of the model. The user control of the model must be provided via some input UI interface properly constructed to give non-technical users control of the model. The resulting analytical outputs are piped through some UI display, which is typically arranged as a dashboard of some kind.

DATA SOURCES AND OUTPUTS

Some data mining architects classify data courses by structure. For example, we can categorize data as residing as (1) Flat Files; (2) Relational Databases; (3) Data Warehouses; (4) Transactional Databases; (5) Multimedia Databases; (6) Spatial Databases; (7) Time Series Databases; and (8) Internet Databases. In some cases, this is a valuable construct, but it is more beneficial to have a more straightforward and more inclusive classification by source and timing.

Figure 6.3 also shows the significant classes of data streams that could feed into the model. Traditional data sources comprise all the transactions processed with an Online Application Processing system (OLAP), such as operations, finance, sales, and customer relations and then entered into a Relational Database Management System (RDBMS). Static data accumulates in a database, which is either SQL-based or other non-SQL-based, such as a JSON database, or it could be Excel spreadsheets. The third major category of data input streams is from volatile data sources that enter our system in real time, such as a stream of Tweets, for example. These could be accumulated into a static database for later analysis, but what makes them interesting and unique is that we often need to analyze them on the fly, as the data is being received, i.e., in real time. Figure 6.4 schematically shows some of these sources and examples of each. We further explain their characteristics later in the chapter.

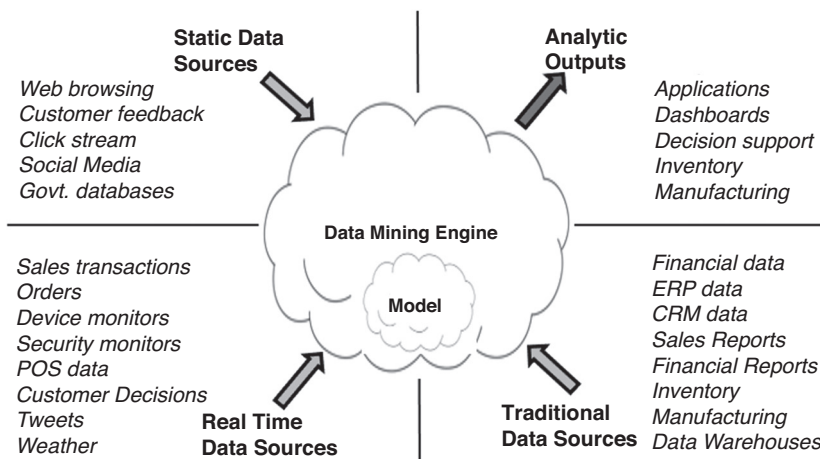


FIGURE 6.4 The traditional, static, and real-time sources of data input to the model and analytic outputs of the model, with illustrative examples.

Traditional Data Sources

Traditional data sources are depositories of transactional data. Examples of these data stores are ERP data, organizational financial systems, logistics, inventory, human resources, and customer relationship management transactions. Many results in the recording of financial transactions and have a great deal of data security. Others record ephemeral yet significant transactions, such as the results of customer interactions. Figure 6.3 shows a few more examples of what we call traditional data sources.

There are several essential characteristics of this data: (1) more often than not, they are recorded in RDBMS systems, so the data is available via the use of SQL queries; (2) the volume of the data records per unit time (a day, a month, a year) is relatively low compared to the other two data sources; (3) it is very high-quality data, it has the few errors and needs minor data preparation to be used for mining; and (4) it has the greatest level of security, so may be difficult to access.

Static Data Sources

Static data sources are those data stores that are not used for operations. In a sense, data warehouses are such static data sources. They are extracted from transactional systems; they are frozen-in-time slices of data that we call historical. Once extracted, they are no longer current, but they can be instrumental in building predictive models. They are also the repository of performance measures of systems or people or actions that are tracked on an ongoing basis but are not operational, financial, or customer transactions. For example, website performance is measured as Web analytics, such as web page visits (one-time or recurring), how long a visitor stayed on a page, or what they linked to next from that page.

Staff in organizations often collect data on recurring transactions that are not captured in operational systems. They may use Excel or some other home-grown database system to track their daily work. Over the years and much use, these may grow voluminously and become intrinsic to the work of a business. They can be sources of data for mining. One example is information security data that needs to be analyzed for security breaches off-line. Another example may be vehicle maintenance data, which accumulates from logs that are kept on vehicles. The historical data can be a valuable source of insight, if adequately modeled. Often, this data is voluminous and is suitable, containing content that can be mined. Imagine accumulating data like this if you operated an extensive vehicle fleet, such as with a package delivery service or an airline or cruise company.

Another static, non-traditional source is data in formats other than traditional numeric or categorical form. One such data format is text data. Open-ended survey responses, customer product feedback on websites, emails, or social media postings are captured at one point in time and saved in a frozen data file for off-line analysis of what the textual data contains. IBM expresses this non-traditional data as a form of data *variety*. Another form of variety in data is images and videos (and is different from numbers and categories). The metadata classifying the images and videos may be captured in a traditional data format, but the content of the image or the video, if treated as data, requires specialized machine learning algorithms. Both the metadata and the contents are collected in static data sources.

Some of the characteristics of static data sources are (1) they are frozen-in-time extractions of data from data capture systems; (2) they come in many forms (spreadsheets, JASON NoSQL formats, text files); (3) the data has been collected off-line; (4) it often requires a lot of data

preparation to get it ready for data mining; and (5) there is often a wide variety of data types (video, images, and social media postings).

Real-Time Data Sources

With the advent social media, organizations have the opportunity to capture and analyze comments from customers, voters, supporters, employees, and many other classes of people in real time. With the advent of sophisticated machine learning algorithms that can process this data in real time, we now have a powerful means for organizations to act on what they see transpiring in real time.

The way to capture this data is to connect the streaming data source to the data processing platform via an Application Programming Interface (API). This is a special piece of code written in a format made available by the company providing the stream of data (Twitter or Facebook, for example) to connect to the data computation platform. Sometimes, the data processing software program company (such as IBM, SAS, Google, and Amazon) provide the API to connect their programs to well-known streams. Instructions are freely available in open-source format for using a programming language like Python or R to connect the streams of data to the appropriate models.

An exciting source of real-time data that could be used to build powerful models is the Internet of Things (IoT). These embedded devices collect and transmit real-time data of processes to a server in the cloud. The resulting data stream can be collected into a static data set to be analyzed periodically or, more importantly, it can be analyzed as it streams in. Take, for example, IoT devices monitoring the functioning of an airplane, even as it is flying, such as what Boeing did with the 777 aircraft. That data is ingested and analyzed in real-time with powerful predictive models to monitor and inform decisions on maintenance and safety. The static collection of all the real-time data can be used for engineers to make better future aircrafts.

Some of the characteristics of real-time data streams are (1) the use of APIs to connect the stream to the models; (2) the amount of data or, as IBM has termed it, data *volume*; (3) the possible need for sophisticated parallel computer systems to process the data in parallel, such as computer clusters; and (4) the potential necessity for processing speeds to work on data that is ingested in short periods of time, or as IBM terms it, planning for high data *velocity* (the necessity of connecting to servers collecting new types of data, such as streaming video content or IoT data from thousands of courses at once).

Analytic Outputs

The outcomes of our models occasionally feed into other models. More often, however, the output of a model is a human being, a user, or a customer of the data mining system. The most common user interface for such model outputs is a dashboard. These are sophisticated data visualization constructs set up by arrangements in consultation with users of the system. Powerful data visualization products can come into play here, such as the commercial product Tableau. However, the simple bar, pie, and scatter plot charts of our programming language or the data mining software are sufficient to convey a graphical image of the analysis.

MODEL BUILDING

Let's now turn our attention to how we will build the model that processes the data. What are the steps? Figure 6.5 shows one such detailed decomposition of the step-by-step process

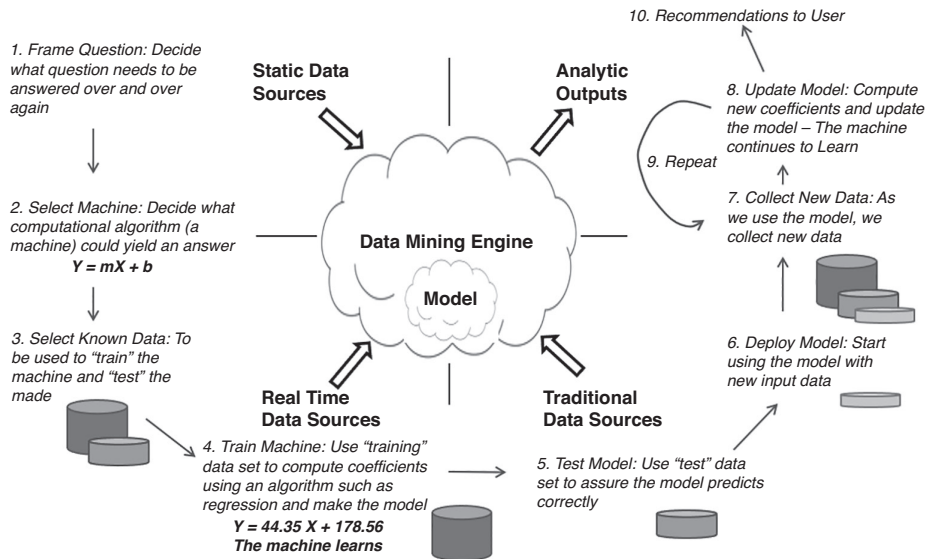


FIGURE 6.5 The ten-step process of creating, validating, and updating a data model.

for model building. We may decompose this most important step into ten activities. Except for the last few steps that call for repetition as the system ingests new data points and continually updates its parameters, it is a fairly linear process. One such decomposition consists of the following:

1. Framing Questions
2. Selecting the Machine
3. Selecting Known Data
4. Training the Machine – Making the Model
5. Testing the Model
6. Deploying the Model
7. Collecting New Data
8. Updating the Model
9. Machine Learning – Repeating Steps 7 and 8
10. Making Recommendations to the User

We will examine the details of each of these steps in the next section.

STEP 1: FRAMING QUESTIONS

We have studied this step in some detail in earlier chapters, but it is essential to include it here, since framing questions drive the creation of the model. We saw in Chapter 2 how to derive framed analytical questions from the business context and the information needs of the users. These questions give us our dependent and independent variables, as well as a hint of what the

computational needs are. The framing of the question naturally indicates the types of algorithms to be used in seeking answers. If we are attempting to compute the ratios of the frequency of categorical variables, that suggests we should use pivot tables, for example.

STEP 2: SELECTING THE MACHINE

This is where the fun begins. It requires all the imagination, skill, and knowledge of the analyst to successfully select the basis of the model. What do we mean by the “machine?” Here, we refer to the mathematical formula that will be invoked to perform computations. If it is a prediction, we have many choices: various formulas relating the predicted variable or outcome to the various input or independent variables. Here we may have options of various techniques, such as the straight-line equation or linear relation. You could also choose a second-order or parabolic relationship or an exponential or logarithmic relationship.

The machine could be a logical decision-making tool that classifies a certain set of input conditions into which class of output that instantiation belongs. It could be expressed as a mathematical construct, such as the k-Means computation for cluster assignment, or a set of logical decisions based on criteria applied to the levels or categories of input variables, such as in a decision-tree.

We select the computational engine that fits the nature of the desired outcome. We look at the framed question and then speculate as to what type of algorithm may be needed. We will categorize the question as to what type of problem it is. Is it a predictive problem? Is it a classification problem? Is it a time series problem? Is it an association problem? Are we looking for anomalies?

The first step is to classify the question or the problem by the nature of its output, or lack thereof. Data scientists teach us that algorithms used for model building can be classified as either a *supervised* or *unsupervised* problem. *Supervised* problems are those in which we seek to predict an output variable (also called a *labeled* or *dependent variable*) based on several inputs (or *independent variables*, also called *features*, of the data set). *Unsupervised* problems are those in which there is no outcome variable; rather, we are perhaps seeking to create classes based on some of the features or variables in the data set with the intent to predict which category a new data point belongs in.

Analysts should be cognizant and know to work with many algorithms to determine the ones that, in their judgment, best fit the question. It is often a matter of degree. An analyst does not need to know all available algorithms to be effective as a data miner. A data analyst just starting to work in the discipline may know a few basic algorithms and uses those competently for most problems. By continually exposing themselves to new techniques and gradually adding to one’s tool box, data analysts can acquire more sophisticated techniques. The analyst then grows into a more effective data miner. (In later chapters, we will cover some of the more basic algorithms and techniques to apply to each type of question.)

STEP 3: SELECTING KNOWN DATA

We reviewed, in detail, data selection and preparation in Chapter 3. Suffice it to say here that the framed question strongly indicates what data (population, variables, and conditions) needs to be collected or extracted from known data sources. This data will be used to train the model algorithm and convert it from a generic algorithm to a model instated on the data set.

STEP 4: TRAINING THE MACHINE

The mathematical basis for computing the answer to the question is what we term the *machine*. To turn it into a model, we need to instantiate it to a data set. The data are real-life observations of our population being tracked and observed and measured. It is codified in a data set (in our case, a table) of rows for each observation and columns for the instantiations of the measurements or observation for that instantiation of the population in that row. The *machine* has generic coefficients. Using a separate computational process — such as another algorithm and the data set — we compute actual values for the generic machine’s coefficients to best fit our data. Once the machine has been instantiated to a data set, we say it has been *trained* or it has *learned*. Thus, we use the term *machine learning*.

Let’s use as an example the most basic of machines, the equation of a line $Y = M * X + B$. First, this algorithm may be classified as a supervised learning situation. It can be used to predict the outcome variable Y using an input variable X . Both X and Y , in this case, are variables in a data table that must be numeric. We may also term Y as the label (the variable in the table we label as the output that we are trying to predict) and X as a feature (one of the variables to be used as a predictor.) We selected this algorithm because somewhere in the framed question, there is a phrase like, “... *given a value of X we can predict the value of Y...?*”

We turn next to the data set we wish to train the machine with, and using a separate process or algorithm, compute the coefficients M and B of the line. In this case, a very popular algorithm to compute the slope M and the intercept B of the line is linear regression. (We will learn to activate this algorithm in Chapter 7). Linear regression returns a value of M as 3.75, and B as 101.6, for example. The model we can use to predict future values of Y given any X is $Y = 3.75 * X + 101.6$.

We have now created a model.

STEP 5: TESTING THE MODEL

Although limited and perhaps imperfect, a model is the best tool we have to predict or classify future unknown situations or conditions. We want our model to give us some control over future events by helping us know the outcome under known and controlled conditions. Thus, we want our model to represent reality as closely as possible. We know there will be limitations of how well any model may do that, but nevertheless, we want to know how close the model can come to reality and its limitations. We want it to be accurate and valid.

To assure ourselves that we have a decent model that reflects real life as closely as possible, there are several tests we can perform. One such test is performed to make sure the algorithm to compute the coefficients did a good job of fitting the model to the data so as to avoid poor predicting using new data points. Such a situation of poor prediction is called *overfitting*. We also want to see how close the model outcomes come to the actual data, especially when predicting. We call that *accuracy*.

A common technique to test for overfitting is to split the known population (rows) of the data set into rows used to create the algorithm (*training data set*) and some rows for testing the algorithm that it does not overfit (*testing data set*). The machine learns with the training data set, and, using the trained model, we try to predict the outcome using the testing data set. We then compute the accuracy (an error rate of predicted versus actual) with both data sets. If they

have comparable error rates (they are not significantly different), we are satisfied there is little overfitting.

These are some of the many tests we may apply to our models. We may even train several models on the same data using competing algorithms to see which one produces the least overfitting and the most accurate results.

STEP 6: DEPLOYING THE MODEL

Once the model has been developed and tested, we begin processing new data. We make sure the model is well aligned to live data streams and that it has useful and meaningful output streams. Deployment may also require careful design and implementation of scaling up. The model may have been developed with a limited data set or in what we may call laboratory conditions. We may see difficulties or problems once the model is deployed on natural live data streams, which may have a higher volume or velocity than the data streams used for model building and testing. These issues should have been accounted for in the original system design. Otherwise, these problems need to be solved as the model is put into production. The model should be monitored over time as users employ it to measure its effectiveness.

STEP 7: COLLECTING NEW DATA

As the model runs in production mode, it processes new data to generate ongoing answers for users. This new data may also be collected as a future input to update the model. Provisions are made to update the model building table with those elements from the input data stream needed for model making.

STEP 8: UPDATING THE MODEL

We could, of course, continue to use the built model as-is for all future time. If business conditions that drive the model do not change radically, there probably will not be a need to update the model. Periodic checks on model performance (overfitting and accuracy) can be done on a periodic basis. Using preset levels of performance, we can see when the model may drift from usability. At such a time, we may, once again, decide to go through the model-building process outlined above to take into account the changing nature of the underlying data.

STEP 9: LEARNING – REPEAT STEPS 7 AND 8

Collecting new data (Step 7) and updating the model (Step 8) may be occasional or scheduled events. The machine continues to learn and the model is updated. This can be occasional, or it can be made into a continuous, ongoing cycle. It may even be continuous, where the model coefficients get updated on a continuous basis. Some sophisticated algorithms, such as recommender systems, depend on the constant update as a significant feature of the system. Recommender system model output recommends to a user what product they may want to consider as an alternative or complement the product they are considering buying at that moment. The algorithm depends not only on the purchase history of the customer, but the current viewing choices, plus all the other purchasing choices made by other customers, current and past. The system requires

close to real-time updating of the model coefficients to be accurate and useful. Imagine that in selecting a movie for Friday night viewing, the system makes a recommendation based on data from a month ago or even one week ago. This data is not very fresh. The system needs to recommend based on what you and others watched less than a day ago, or even in the last hour, or even as people are making viewing choices in real-time. Not all models need that much recency in their updating, but such an extreme example gives a good glimpse of the need to design the right frequency of model updating.

STEP 10: RECOMMENDING ANSWERS TO THE USER

Last, the model must be used. Its purpose is to provide business answers to business questions users have posed (ad hoc) and will continue to pose periodically moving forward. Figure 6.6 summarizes the detailed steps we have covered earlier to provide recommendations to the user. In reality, what we term here as recommendations are answers to the posed framed analytical questions. These answers become pieces of evidence, facts in the data-driven decision-making process that inform recommended actions that managers and executives may take. We covered this process in detail in Chapters 1 and 2.

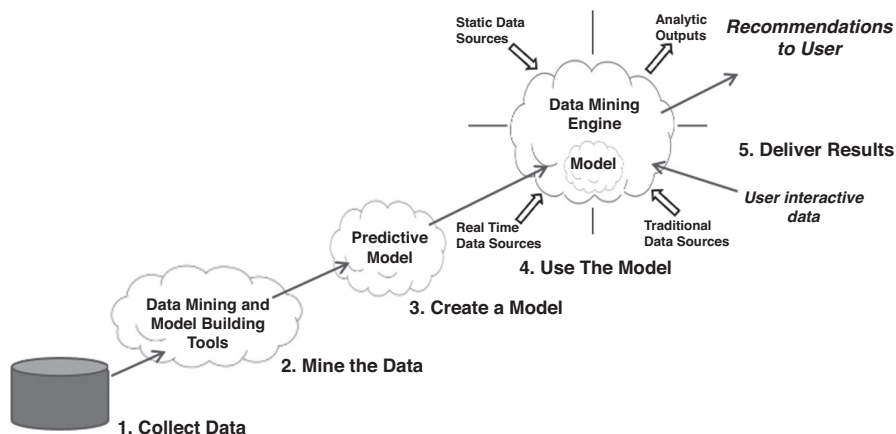
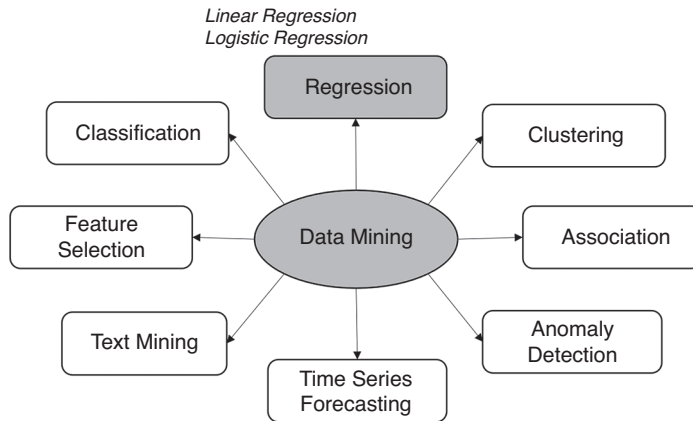


FIGURE 6.6 The process of using a model and deploying it to provide recommended answers to business user questions.

REFERENCE

Chapman, Pete. 2000. *CRISP-DM 1.0: Step-by-Step Data Mining Guide*. SPSS.

PREDICTIVE ANALYTICS WITH REGRESSION MODELS



A regression algorithm is a common technique for predicting an outcome variable from several input variables. This technique is sometimes called the *regression to the mean*. It is the most popular form of machine learning and the first one we think of when we think of prediction. In reality, this technique consists of the application of two algorithms. First, we select the type of predictive machine we will use to create a model. It could be the equation of a line, a first-order polynomial, relating an output way to input X with a simple straight-line equation. An equally crucial predictive algorithm is that of an output variable Y related to an input variable X by logarithm, which is related to the natural logarithm e raised to some power of X. These two equations, linear and logarithmic, are by necessity generic with generic positions relating X and Y. We instantiate the coefficients by using a data set, and a different technique is used for determining the results in that instance.

The second algorithm is regression, the most common algorithm used to determine the coefficients in the linear and logarithmic models. We assume either a linear or logistic relationship between X and Y and find acquisitions for that relationship that minimize the overall distance between the resulting linear logistic models and all the data points. That distance is computed as the root mean square of the distance between the line and data points. The coefficient is

discovered iteratively by assuming some values and then changing them until the error between the regression line and the data is relatively small. At this point, the algorithm stops iterating.

The resulting instantiated linear or logistic model is an average, or mean, of all the points (this process is sometimes referred to as the regression to the mean). We treat the resulting model as an average value for any one value of X for all the possible values of Y.

This chapter will show how we can develop a linear or a logistic regression model using a data set and terminating one variable as the outcome variable and one or several other variables as input. As in machine learning, the machine has either a linear or logistical relationship between X and Y. In both these cases, the input variables, by necessity, have to be numeric. In the case of linear regression, so does the output variable; it also has to be numeric. In the case of logistic regression, the output variable is a binary, often expressed as zero or one.

There are other forms of regression, which may include categorical variables, that is, the use of decision trees to estimate an output variable. In the case of the output, the variable is numeric, and the algorithm is used as a decision tree (We call it a *regression tree*). Decision trees are discussed in a later chapter, but we do not explore regression trees in any meaningful detail. We will only use the two most common types of predictive algorithms here: linear regression and logistic regression.

WHAT IS SUPERVISED LEARNING?

Linear and logistic regression are also forms of supervised learning, by which we mean techniques that predict the value of the output variable based on a set of input variables. *Supervised*, or *directed*, *learning* tries to infer a function or relationship based on labeled training data and uses this function to map new unlabeled data. What do we mean by this precise data science-based terminology? First, we are training (using data from the past) a model that predicts one of the variables in the data set using some of the other variables as input.

The variable we predict using the resulting trained machine is known by various names. Data scientists call this variable the *label*; in other words, we have placed a label on this variable as the outcome or predicted variable in our data set. It is the *target* of our prediction. When such a variable is designated, we say we are dealing with *labeled data* (one variable has been so designated as the target of our prediction). This is the main difference between supervised (there is a variable we hope to predict) and unsupervised learning (there is no prediction of a particular variable). The second form is where we hope to group rows of data into like segments, or clusters, by some of the features (some of the other variables) in the data set. The labeled variable is also called the *predicted variable* or the *outcome variable*. The input variables in our model are known in data science as the data set's *features* and are also referred to as the *input variables*. Regression is an algorithm to produce a model by applying a supervised machine learning technique.

REGRESSION TO THE MEAN

The most common machines or production algorithms are (1) a linear model, where the equation of a straight line relates the variables; (2) a logistic model, where the input model is related to the logarithm of the output model; and (3) a random forest relationship based on regression trees.

The technique to compute the coefficients of the model equations, the act of training the machine, is a different algorithm from the actual model competing algorithm. In the case of linear and logistic models, the computing of the coefficients is done with a process called *regression*. In that process, successive coefficient values are tried to reduce the overall error of fitting the model equation to the data points. We sometimes speak of a *linear regression* model where what we mean is that we trained a linear relationship algorithm (the first-order linear equation of a line) using a regression algorithm with training data to produce a linear model.

Linear regression models are effective in many circumstances where there is a numeric output variable and one or more numeric input variables. Logistic regression is used frequently where a binary (or two-factor categorical) variable is the output, and the input variable is one of the more numeric variables. Regression tree-based models are beneficial in predicting a numeric variable when there is a mix of numeric and categorical input variables.

Not all relationships may be linear or logistic. In certain circumstances, a power law relationship may exist, or an exponential growth relationship is apparent. Sometimes the equation of a parabola (quadratic) or a cubic relationship is more appropriate. Then, using a regression algorithm to obtain the parameters, we may find a model other than linear or logistic more accurate. We will discuss this using a model-fitting measure of goodness called R-squared later in the chapter.

LINEAR REGRESSION

A simple linear regression (SLR) machine is the linear mathematical equation that defines the best linear relationship between two variables, X and Y, as shown in Figure 7.1.

$$Y = f(X) = mX + b$$

Here, Y is the response (outcome, output, label, or dependent) variable, what we are predicting (e.g., catalog expenditures or lifetime value of a customer). X is the predictor (input, feature, or independent) variable (e.g., age or income). *b* is a constant numerical value and denotes the Y-intercept when X takes on a value of zero. *m* is the slope of the regression line.

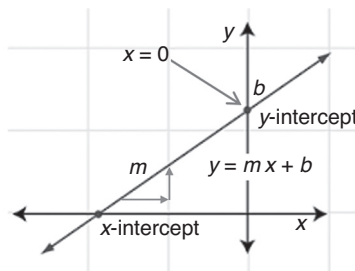


FIGURE 7.1 The equation of a straight line showing the coefficients and their relationships.

SIMPLE LINEAR REGRESSION

Let us assume we have a data set of customers' purchasing behavior versus their income level. We extract a sample of 10 customers to build a predictive model of 1-year Lifetime Value (LTV), which is their cumulative purchases for the previous year, as the outcome, or predicted

variable. We use their income level as the predictor. Figure 7.2 shows the data table of the descriptive statistics of both variables and the correlation between them computed using Jamovi. It looks like they are highly positively correlated. A scatter plot with a trend line shows that this data set is a candidate for linear regression modeling.

Income in thousands (x)	58	42	24	76	33	69	31	46	51	38
1-Yr. Lifetime Value (y)	76	45	26	102	42	97	33	49	52	40

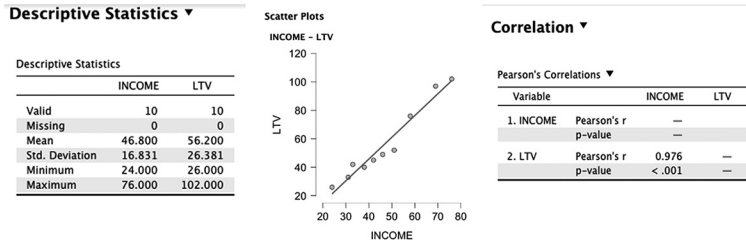


FIGURE 7.2 The data table, the descriptive statistics of both variables, and their correlation computed using Jamovi.

The purpose of simple linear regression modeling is to define the relationship between two variables by fitting a straight line through the data points. Using an iterative process to reduce the overall error between the line and the data, the regression algorithm chooses the line that best fits the data, as shown in Figure 7.3.

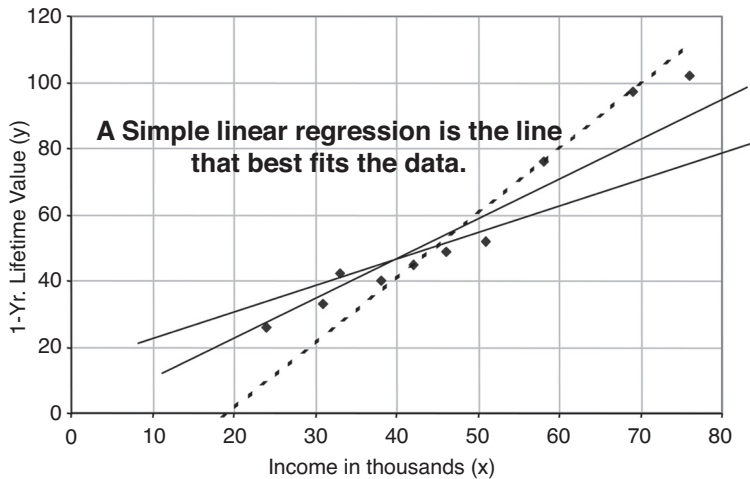


FIGURE 7.3 The iterative process to reduce the overall error between the line and the data shows how the regression algorithm chooses the line that best fits the data.

We used Jamovi to produce a linear regression model of the data. The resulting model from training the line with the customer data is shown in Figure 7.4. We see the two coefficients of the linear model, the slope $m = 1.53$ and the intercept $= -15.395$.

The resulting model is

$$\text{1-year LTV} = 1.53 \times \text{INCOME} - 15.395$$

We can now use the computed coefficients in a model equation to predict the 1-year LTV of a new customer whose income we know, as shown in Figure 7.5. We also show the error between

Coefficients ▼

Model	Unstandardized	Standard Error	Standardized	t	p
H ₀	(Intercept) -15.395	5.964		-2.581	0.033
	INCOME 1.530	0.121	0.976	12.683	< .001

FIGURE 7.4 The resulting model from training the line with the customer data.

C15				
	A	B	C	D
1	INCOME	LTV	LTV Predicted	Error
2	58	76	77.0	1.3%
3	42	45	51.5	14.5%
4	24	26	22.8	-12.2%
5	76	102	105.7	3.6%
6	33	42	37.2	-11.5%
7	69	97	94.5	-2.6%
8	31	33	34.0	3.0%
9	46	49	57.9	18.1%
10	51	52	65.8	26.6%
11	38	40	45.1	12.8%
12			Average error	5.4%
13				
14	New Customer			
15	50		64.3	

FIGURE 7.5 The application of the model to a new customer by applying the formula for the line from the discovered coefficients and showing the average error rate of our regression.

the line (predicted) and actual (data points). Our average error is 5.4%, which is not an alarming error rate. For a new customer whose income is \$50,000 per year, we can predict that on average, they will produce for our business a \$64,300 of Life Time Value.

THE R-SQUARED COEFFICIENT

R-squared is a statistical measure of how close the data are to the fitted regression line and the goodness of the fit of a regression model. It is also known as the *coefficient of determination* or the *coefficient of multiple determination* for multiple regression. R-squared is the percentage of the outcome variable's variation explained by a linear model.

$$\text{R-squared} = \frac{\text{Explained variation of the data}}{\text{Total variation of the data}}$$

R-squared is always between 0 and 1, or maybe as a percentage: (1) 0% indicates that the model explains none of the variability of the response data around its mean, and at the other extreme, (2) 100% indicates that the model explains all the variability of the response data around its mean. Generally, the higher the R-squared, the better the model fits your data. Figure 7.6 shows four conditions from an exact fit (R-squared = 100% or .1) where the model explains 100% of the variability of the data to an 80% fit where the model explains 80% of the variability

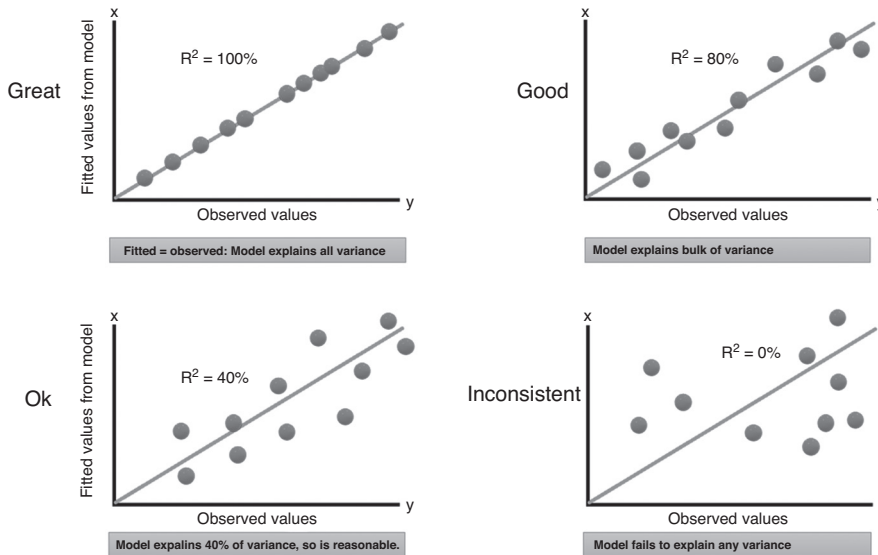


FIGURE 7.6 The R-squared coefficient indicates how well the model fits the data.

of the data, and we say we have an 80% fit. The same holds for a 40% fit when it does not fit at all (0%) and when the model cannot explain any of the variability of the data.

When we construct a linear regression model, such as the one shown in Figure 7.7, the R-squared coefficient of the model fit is often given as a model parameter.

Linear Regression ▼

Model Summary - LTV				
Model	R	R ²	Adjusted R ²	RMSE
H ₀	0.976	0.953	0.947	6.090

Note. Null model includes INCOME

Coefficients ▼						
Model		Unstandardized	Standard Error	Standardized	t	p
H ₀	(Intercept)	-15.395	5.964		-2.581	0.033
	INCOME	1.530	0.121	0.976	12.683	<.001

FIGURE 7.7 The R-squared coefficient and the p-values for the simple linear regression model showing how well the model fits the data.

THE USE OF THE P-VALUE OF THE COEFFICIENTS

Another measure of the goodness of the fit of the computed regression is the assurance that the model coefficients are adequate, such a measure of goodness for the coefficients in the p-value. We encounter p-values in inferential statistics, where we compare the means of two normal distributions and whether they are significantly different. We also encounter the p-value in dealing with the independence of categorical values. We ask whether the dependence we see is purely random or if there is some underlying bias relating to the two variables. Here, we use the p-value to ensure the goodness of fit of the model coefficients.

The p-value for each independent variable tests the null hypothesis that the variable does not correlate with the dependent variable. If there is no correlation, there is no association between

the changes in the independent variable and the shifts in the dependent variable. In other words, there is insufficient evidence to conclude that they are related. Suppose the p-value for a variable is less than a significance level (an alpha level is usually set to .05 for business analysis). In that case, we can conclude that the data provide enough evidence to reject the conclusion that the two variables are not correlated. The data favors the hypothesis that there is a non-zero correlation. The variable we have computed a model coefficient for is statistically significant and probably worthwhile keeping in the regression model.

The lower the p-value, the greater the statistical significance of the observed difference. A p-value of 0.05 or lower is generally considered statistically significant. The p-value associated with a coefficient can serve as a confidence level for that variable as a significant addition to the model. High p-values indicate that the evidence is not strong enough to suggest that this variable sufficiently affects the outcome variable.

A p-value more significant than the significance level indicates insufficient evidence in the data sample to conclude that a non-zero correlation exists. An effect might exist, but the effect size may be too small, the sample size is too small, or there is too much variability to detect. Typically, you use the coefficient p-values to determine which terms to keep in the regression model. Keeping such a variable could introduce noise in the model, causing a higher-than-normal error rate. We should consider removing this input variable from the model and creating the model without it.

In the example in Figure 7.7, we see that the p-value for the *INCOME* variable has a value of less than 0.001, well within the 0.05 acceptable limit. Thus, we have high confidence that the outcome variable and the *INCOME* input variable would keep this variable in the model. The predictor variable, *INCOME*, has a statistically significant relationship with the outcome variable.

STRENGTH OF THE CORRELATION BETWEEN TWO VARIABLES

We have other ways of assessing the strength of the relationship between two numeric variables. There are many different ways of doing this, but the most common is a Pearson correlation, where we are asking the question: How well does a straight line fit the scatterplot of the X and Y points? The R-squared coefficient is a good measure of the fit of a linear regression to X and Y data points. The Person correlation coefficient, in this case, is the square root of R-squared. Figure 7.7 shows the computed R-squared coefficient as the result of the regression, with the correlation coefficient given right alongside R (R-squared = 0.953 and R = 0.976, its square root). Or,

$$\text{Correlation Coefficient} = R = \sqrt{\text{R-squared}}$$

We see in Figure 7.8 the progression from the perfect alignment of the data points to a linear regression, which we call a *perfect correlation* in a positive direction (the figure on the extreme

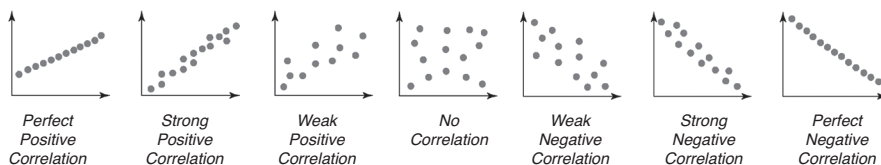


FIGURE 7.8 The correlation coefficient measures the strength of the relationship between two numeric variables.

left). We progress in scattering levels to no correlation in the center to where a straight line does not fit, a perfectly negative correlation on the extreme right.

Figure 7.9 shows a translation of the terms we use to describe correlation to the actual numeric ranges and how we speak of that relationship. For example, a correlation coefficient of $+0.75$ would be considered a strongly positively correlated relationship. A value of -0.25 would be considered a weakly negatively correlated coefficient. A correlation coefficient of $.05$ would indicate no correlation between the variables.

Correlation	Strength
1	Strongly positively correlated
0.7	
0.7	Positively correlated
0.4	
0.4	Positive weakly correlated
0.2	
0.2	Not correlated
0	
0	Not correlated
-0.2	
-0.2	Negatively weakly correlated
-0.4	
-0.4	Negatively correlated
-0.7	
-0.7	Strongly negatively correlated
-1	

FIGURE 7.9 The correlation coefficient as an indicator of the relationship between two variables and the language we use to describe the strength and direction of the relationship.

We often use the correlation coefficient as an additional indicator of whether a variable is sufficiently related to the outcome variable to be included in our linear regression model.

EXERCISE 7.1 - USING SLR ANALYSIS TO UNDERSTAND FRANCHISE ADVERTISING

Your boss wants your help to understand what drives the performance of stores in the franchise. He (or she) asks you to be able to predict how much advertising should be allocated, given a sales goal for a particular franchise. Your boss helps you acquire data from last year's sales and the demographic factors of each store (*FRANCHISES.csv*).

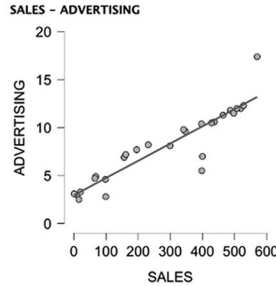
You decide to build a linear regression model of advertising versus sales. This is your well-framed question:

If there is a strong relationship between sales and advertising budget, what is the linear model that can predict how much advertising should be in the budget given an annual sales goal, and how good is that model?

Perform the analysis and then give your answer to the above question, with your reason for the answers. The Excel data file has a data dictionary describing the variables' meaning. Notice in Figure 7.10 that the correlation between *SALES* and *ADVERTISING* is very high, 0.914.

Correlation

Pearson's Correlations		
Variable	SALES	ADVERTISING
1. SALES	Pearson's r p-value	— —
2. ADVERTISING	Pearson's r p-value	0.914 < .001

Scatter Plots

Linear Regression
Model Summary - SALES

Model	R	R ²	Adjusted R ²	RMSE
H ₀	0.000	0.000	0.000	192.062
H ₁	0.914	0.835	0.829	79.455

ANOVA

Model		Sum of Squares	df	Mean Square	F	p
H ₁	Regression	801254.086	1	801254.086	126.920	< .001
	Residual	157826.266	25	6313.051		
	Total	959080.352	26			

Note. The intercept model is omitted, as no meaningful information can be shown.

Coefficients

Model		Unstandardized	Standard Error	Standardized	t	p
H ₀	(Intercept)	286.574	36.962		7.753	< .001
H ₁	(Intercept)	-90.150	36.770		-2.452	0.022
	ADVERTISING	46.509	4.128	0.914	11.266	< .001

FIGURE 7.10 An SLR model for *SALES* predicted by *ADVERTISING* for the *FRANCHISES* data set.

Notice that when the regression model computes, the R-value is given as 0.914, which is the correlation coefficient. Note that the R-squared goodness of fit of the model is 0.835, or the model explains 85% of the variability of the data and is a good fit. Note that the p-value for the *ADVERTISING* variable coefficient is less than 0.001, which speaks of its strong relationship to *SALES* and the adequacy of the model.

MULTIVARIATE LINEAR REGRESSION

So far, we have considered the simple case of one predictor and one predicted variable, and there are many situations where this is sufficient. The more interesting case is where we have many input variables to predict an outcome variable. Just as when there is one variable, or univariate when we have multiple input variables, we call it multivariate; thus, multivariate linear regression (MLR) is our next topic of study. Again, we are going only to consider cases where the input variables and the output variables are numeric since this is the case where linear regression applies. We can then write the relationship between a Y variable as a target with many X variables as predictors as a model specified by a linear equation:

$$Y = b + m_1X_1 + m_2X_2 + m_3X_3 + \dots$$

Here, *Y* is the response variable, what you are predicting (e.g., an order or a cancellation of service, etc.). *X*₁, *X*₂, *X*₃, ... are the multiple predictor variables (e.g., age, income, or RFM data elements). *b* is a constant numerical value, the y-intercept. *m*₁, *m*₂, *m*₃, ... are the numerical coefficients (weights) associated with each of the predictor variables, the univariate slopes.

As before, the response variable is also called the *dependent* variable, and the predictor variables are the *independent* variables. Figure 7.11 shows the conceptual transition from univariate linear regression to multivariate linear regression, from two dimensions to three or more dimensions.

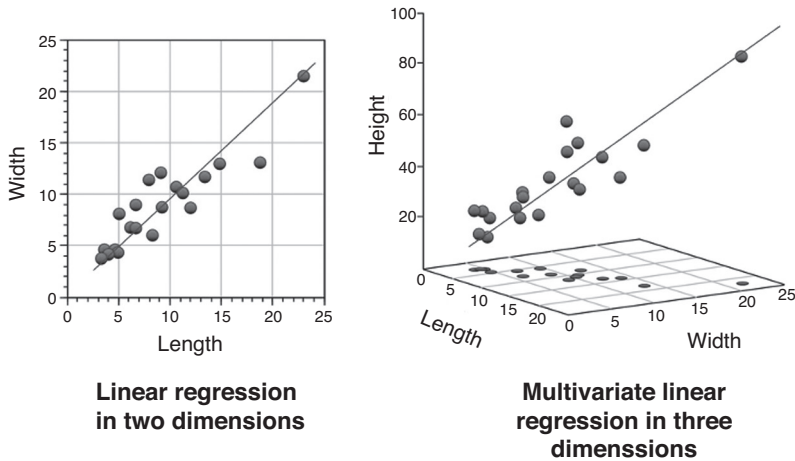


FIGURE 7.11 Univariate and multivariate linear regression examples.

PREPARING TO BUILD THE MULTIVARIATE MODEL

As we did with SLR, we created a sample data file and split it into one set for training and another for testing. Using the training portion, we first run a correlation analysis to help reduce the set of potential predictor variables to a more manageable number. We discard those predictors with low correlation coefficients to the output, assuming that their inclusion will not make the model more accurate but will probably introduce noise into our model. If we had left them in and created a regression model, we would see that the p-value for those uncorrelated coefficients would be above the alpha criterion and discarded them in any event.

EXERCISE 7.2 - USING MULTIVARIATE LINEAR REGRESSION TO MODEL FRANCHISE SALES

Let us apply MLR and create a model of franchise sales predicted from franchise store attributes. We want to be able to tell what the sales of a store are given specific store attributes: inventory, advertising budget, store size, number of customers, and the strength of the competition. We shall use a multivariate (many variables) linear regression model.

Locate the *Franchises.xls* data set in the *Chapter 7* folder of the *Case Data* repository and open it with JASP. Build a multivariate regression model to predict sales given a set of other variables. We are going to answer the following business questions:

What factors affect sales in our franchises, and how?

Can we create a model to design a store size, advertising spending, and specific demographics to achieve a certain sales level?

Your boss wants your help to understand what drives the performance of stores in the franchise. You acquire data from last year's sales and the demographic factors of each store and decide to build a linear regression model of all store attributes versus sales. There are well-framed questions:

What is the relationship between SALES and all the other variables?

If we had five possible new locations, which one is more likely to yield a minimum desired sales level?

Let us use linear regression to build our model and answer these questions. First, we perform a correlation analysis between all the variables, as shown in Figure 7.12. An analysis of the correlation coefficients between SALES as the target and all the other variables as predictors shows that they are strongly correlated, all positively correlated (above 0.9) except for STORES, which is negatively correlated (-0.912) as expected. As the number of competitor stores increases, we expect sales to decrease.

Correlation ▾

All variables are highly correlated to SALES

Pearson's Correlations ▾

Variable	SALES	SQFT	INVENTORY	ADVERTISING	FAMILIES	STORES
1. SALES	Pearson's r p-value	— —				
2. SQFT	Pearson's r p-value	0.894 < .001	— —			
3. INVENTORY	Pearson's r p-value	0.946 < .001	0.844 < .001	— —		
4. ADVERTISING	Pearson's r p-value	0.914 < .001	0.749 < .001	0.906 < .001	— —	
5. FAMILIES	Pearson's r p-value	0.954 < .001	0.838 < .001	0.864 < .001	0.795 < .001	— —
6. STORES	Pearson's r p-value	-0.912 < .001	-0.766 < .001	-0.807 < .001	-0.841 < .001	-0.870 < .001

FIGURE 7.12 Correlation coefficient matrix for SALES as the output variable versus all the other factors as input variables.

The results of the MLR model are shown in Figure 7.13. Notice that the overall correlation between SALES and the other variables is very high ($R=0.997$), as expected. The overall fit of the linear model is also very high ($R\text{-squared} = 0.993$). The p-values of all the model coefficients are above our alpha criterion of 0.05, which says that all the predictors used are highly related to the target and should be kept, as expected.

Linear Regression

Model Summary - SALES

Model	R	R ²	Adjusted R ²	RMSE
H ₀	0.997	0.993	0.992	17.649

Note. Null model includes SQFT, INVENTORY, ADVERTISING, FAMILIES, STORES

Coefficients

Model		Unstandardized	Standard Error	Standardized	t	p
H ₀	(Intercept)	-18.859	30.150		-0.626	0.538
	SQFT	16.202	3.544	0.170	4.571	< .001
	INVENTORY	0.175	0.058	0.174	3.032	0.006
	ADVERTISING	11.526	2.532	0.227	4.552	< .001
	FAMILIES	13.580	1.770	0.363	7.671	< .001
	STORES	-5.311	1.705	-0.135	-3.114	0.005

FIGURE 7.13 The Multiple Linear Regression model for SALES versus all the other factors.

The resulting model equation is

$$SALES = SQFT*(16.202) + INVENTORY*(1.75) + ADVERTISING*(11.526) + FAMILIES*(13.58) + STORES*(-5.311) + (-18.859)$$

Let us use the model to perform some analysis. We are going to answer the following question:

We just opened a store in a neighborhood with 5,000 families. The store is 5,000 sq. ft.; we are planning to spend \$5,000 a month on advertising, carry \$250,000 in inventory, and have five competing stores in the neighborhood. What are the projected sales?

Make sure to use the information in the data dictionary to normalize the variables properly to use the model equation properly. The answer is as follows:

$$SALES = 5 * 16.202 + 250 * 1.75 + 5 * 11.526 + 5 * 13.58 + 5 * (-5.311) + (-18.859)$$

Which computes to be

$$SALES = \$598 \text{ (in thousands)}$$

LOGISTIC REGRESSION

In the prior section, we saw where we could predict a continuous numeric variable from several continuous numeric variables as predictors using linear regression. What if the target variable was a simple binary outcome? Would we still use linear regression, knowing that a straight line would have significant errors? Figure 7.14 shows just such a situation. A straight-line regression prediction would be mostly in error everywhere (Figure 7.14(b)). Is there a better approach to modeling this binary set of events? Could we use a different mathematical formula, a different machine, to fit the data better?

Fortunately, the answer is yes, there is such a machine. It is a natural logarithmic relationship between the predictor and the predicted, which produces a much more accurate model when employed in a regression model. We see such a function applied to data in Figure 7.14(c). Such an algorithm is called a *logistic regression*, which we explore next. We will also compare it to linear regression to discover when to apply which algorithm.

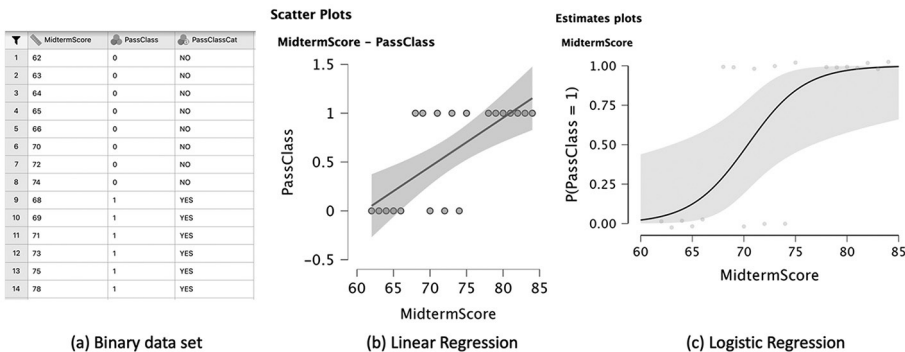


FIGURE 7.14 A comparison between linear regression and logistic regression-based models for binary data.

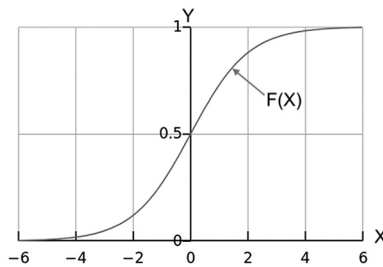
WHAT IS LOGISTIC REGRESSION?

Logistic regression measures the relationship between a categorical dependent variable and one or more independent variables, usually (but not necessarily) continuous, by estimating probabilities. Logistic regression addresses the issue of predicting a target variable that may be binary or binomial (such as 0 or 1, yes or no) using predictors or attributes, which are numeric.

The function relating outcome to input is

$$Y = F(X) = 1 / (1 + \exp - (B_0 + B_1 * X_1))$$

When plotted, it looks like the curve in Figure 7.15(a), and it is computed in Excel as shown in Figure 7.15(b).



(a) Plot of the logistic function

	A	B	C	D
1	x	b0	b1	y
2	2	0.58	-0.876	0.236494

$$=1/(1+EXP(-1*(B2+C2*A2)))$$

$$y = 1 / (1 + \exp - (b_0 + b_1 * x))$$

(b) Computed in Excel

FIGURE 7.15 The plot of the logistic function and how it is computed in Excel with an example set of coefficient and variable values.

EXERCISE 7.3 – PASSCLASS CASE STUDY

Let us create a logistic regression model for a target variable and a single predictor. Can you predict if a student will pass a class knowing his midterm score? We will use the file *PassClass.csv* data file of the performance in this class of previous students. The file may be found in the *Chapter 7* folder in the *Case Data* depository. We will use *MidtermScore* as the predictor and *PassClass* as the target. We will use last semester's grades to build the model and use it in the current semester right after the midterm exam to coach students with a high likelihood of not passing the pass. *MidtermScore* is necessarily a continuous numeric variable. *PassClass* is a categorical YES/NO binary variable. Correlation analysis of the two variables reveals a Pearson's correlation coefficient of $R=0.705$ (we expect their midterm score to strongly influence the likelihood of passing the class.) Note that the scatter plot diagram with a linear regression line shows that a linear regression model will not be as accurate as a logistic regression one. Figure 7.16 shows the results of the descriptive analysis of the two variables.

Let us use the JASP logistic function to build a logistic regression model. The result of the analysis is shown in Figure 7.17.

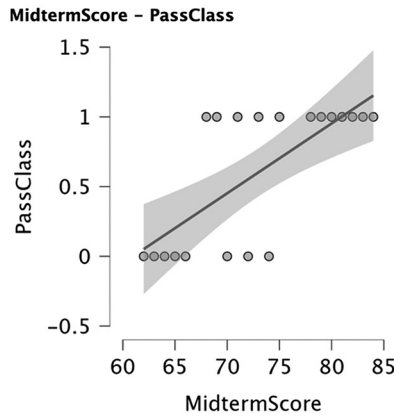
The resulting logistic regression model is

$$Y = F(X) = 1 / (1 + \exp - (-25.6 + 0.364 X))$$

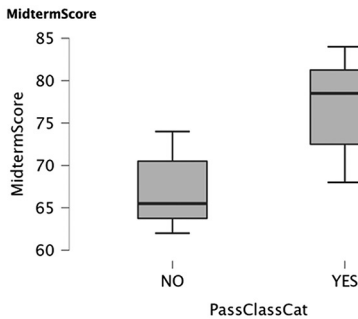
Descriptive Statistics ▾

Descriptive Statistics ▾		
	MidtermScore	PassClass
Valid	20	20
Missing	0	0
Mean	72.950	0.600
Std. Deviation	7.075	0.503
Minimum	62.000	0.000
Maximum	84.000	1.000

Scatter Plots ▾



Boxplots ▾



Correlation Matrix

		MidtermScore	PassClass
MidtermScore	Pearson's r	—	
	p-value	—	
PassClass	Pearson's r	0.705	—
	p-value	<.001	—

FIGURE 7.16 The descriptive statistics of the variables in our *PassClass* case study.

Logistic Regression

Coefficients

	Estimate	Standard Error	z	Wald Test		
				Wald Statistic	df	p
(Intercept)	-25.602	11.184	-2.289	5.240	1	0.022
MidtermScore	0.364	0.158	2.300	5.289	1	0.021

Note. PassClass level '1' coded as class 1.

Performance Diagnostics ▾

Confusion matrix ▾

Observed	Predicted		% Correct
	0	1	
0	6	2	75.000
1	2	10	83.333
Overall % Correct			80.000

Note. The cut-off value is set to 0.5

FIGURE 7.17 The logistics regression model was built to predict passing the class knowing the midterm score and the confusion matrix showing a measure of the model's accuracy.

Note that the p-value of the coefficient *MidtermScore* is less than the cutoff 0.05 (p-value = .022), as we expected from the correlation analysis. We also computed the confusion matrix, a measure of the model's accuracy. If the model is used to predict the value of the training data and we compare if the predictions match the data, we find an overall accuracy of 80%—a good result.

Now use the model to compute the “odds that a midterm grade of X will result in passing the class.” If a student gets a 65 in this exam, in this particular class, and if all other exams and assignments are similar to the previous semester with the same teacher, you expect that his chance of passing the class is 12.5%. He needs help.

MULTIVARIATE LOGISTIC REGRESSION

As we did with linear regression, we have considered the simple case of one predictor and one predicted variable. Again, there are many situations where this is sufficient. But we saw that the more interesting case is where we have many input variables to predict an outcome variable. Just as when there is one variable univariate when we have multiple input variables, we call it multivariate; thus, multivariate logistic regression (MLR) in this case. Again, we are going only to consider cases where the input variables are numeric, and the output variable is binary since this is the case where logistic regression applies. We can then write the relationship between a Y variable as a target with many X variables as predictors as a model specified by a natural logarithmic relation as

$$Y = F(X) = 1 / (1 + \exp - (B_0 + B_1 * X_1 + B_2 * X_2 + B_3 * X_3 + \dots))$$

All input variables need to be numeric, and the outcome variable is binary.

Let us use it to develop a model to predict if a customer will purchase a product based on their past purchasing history.

EXERCISE 7.4 – MLR USED TO ANALYZE THE RESULTS OF A DATABASE MARKETING INITIATIVE

Entertainment Today is a club marketer of music videos. They are testing a new music club concept. A 25,000 sample of names from the *Entertainment Today* customer database was test mailed for this brand-new music club concept. Those who joined the new club received ten free CDs and agreed to purchase two more over the next 12 months.

The marketplace test had a response rate of 40% for the initial offer of 10 free CDs. All customer data was saved as a point-in-time response data set of the promotion for future analysis purposes. *Entertainment Today* did decide to roll out a new music club concept. They wish to promote the new club to only some names in their customer database. As such, they have requested the construction of a response model to help them select the names most likely to join the club. Since the response or target variable is binary (BUY/NOT BUY) and the predictors are various numeric customer attributes (such as lifetime customer value, number of recent orders, and the total dollar volume of all orders), a logistic regression model seems to be the best choice.

Using the frozen response file, *SmallSample.xlsx*, you will build a multivariate logistic regression response model predicting who will most likely join the new music club. The file may be found in the *Chapter 7* folder in the *Case Data* depository. We will use Excel for this exercise and base the analysis on a sub-sample of 150 names randomly drawn from the 25,000 sample database. We will then use the model to score all customers on the database and only promote the club to those highly likely to buy. This is the essence of database marketing.

All variables in this sample are numeric. The definition for each variable is as follows:

- *Cust_ID* = unique customer id number
- *TSLO* = Time since last order, the elapsed time, in months, since the last order
- *NM_ORD* = Total number of orders since becoming a customer
- *DOLL_CR* = Total dollars credited (spent) since becoming a customer
- *ORDER* = 1 means the customer joined the new music club, and 0 means they did not respond

Let us run a multivariate logistic regression model using all three variables (*TSLO*, *DOLL_CR*, and *NM_ORD*) as the predictors and the order indicator (*ORDER*) as the dependent variable.

The resulting model predicting *ORDER* using JASP is shown in Figure 7.18.

Note that the resulting model is

$$Y = F(X) = 1 / (1 + \exp(-(-0.444 - 0.021 \text{ DOLL_CR} + 0.085 * \text{NM_ORD} - 0.195 \text{ TSLO})))$$

Logistic Regression

Coefficients

	Estimate	Standard Error	z	Wald Test		
				Wald Statistic	df	p
(Intercept)	-0.444	0.508	-0.875	0.766	1	0.382
TSLO	-0.195	0.075	-2.594	6.729	1	0.009
NM_ORD	0.850	0.827	1.028	1.058	1	0.304
DOLL_CR	-0.022	0.035	-0.622	0.387	1	0.534

Note. ORDER level '1' coded as class 1.

Performance Diagnostics

Confusion matrix

Observed	Predicted		% Correct
	0	1	
0	76	14	84.444
1	32	28	46.667
Overall % Correct			69.333

Note. The cut-off value is set to 0.5

Pearson's Correlations ▼

Variable		ORDER
1. ORDER	Pearson's r	—
	p-value	—
2. TSLO	Pearson's r	-0.286
	p-value	< .001
3. NM_ORD	Pearson's r	0.362
	p-value	< .001
4. DOLL_CR	Pearson's r	0.373
	p-value	< .001

FIGURE 7.18 The logistics regression model predicts who would buy the product.

The model shown in Figure 7.17 exhibits some exciting features. First, we added a correlation coefficient matrix analysis to the figure to analyze the relationship between the order variable and the intended predictors. We see that the correlation coefficients vary from -0.2682 to 0.373 , which are moderate correlations. We also notice the model accuracy at around 70%, as given by the confusion matrix, which is acceptable. A more significant concern is the p-values for the model coefficients. The *TSLO* coefficient p-value is an acceptable 0.009, but the p-value for the other two predictors is above our .05 criterion. What is going on here? In reality, both the number of orders variable and the total dollar amount for all order variables are measuring the same thing, and so they are confusing the model. We call variables of this nature that measure very similar things *co-linear*. Used together they could, as in this case, introduce noise into the model. One way to resolve the problem is to drop the least correlated of the two co-linear variables from the model and rerun it. Figure 7.19 shows the model with only the *TSLO* and the

Logistic Regression

Coefficients						
	Estimate	Standard Error	z	Wald Test		
				Wald Statistic	df	p
(Intercept)	-0.580	0.488	-1.188	1.411	1	0.235
TSLO	-0.149	0.058	-2.567	6.588	1	0.010
DOLL_CR	0.014	0.004	3.586	12.858	1	< .001

Note. ORDER level '1' coded as class 1.

Performance Diagnostics ▼

Confusion matrix ▼			
Observed	Predicted		% Correct
	0	1	
0	74	16	82.222
1	31	29	48.333
Overall % Correct			68.667

Note. The cut-off value is set to 0.5

FIGURE 7.19 The logistic regression model to predict who would buy the product was resolved with the co-linearity of two predictors.

DOLL_CR variable. Notice that the p-value for the *DOLL_CR* variable is now within acceptable limits. Even though the model accuracy did not improve (it dropped a little), this is a more accurate model.

Where is Logistic Regression Used?

This predictive technique is beneficial and popular for situations where the outcome variable is binary. There are many such situations where it can be used:

1. Gaming – Predicting Win vs. Loss.
2. Sales – Predicting Buying vs. Not buying is an excellent tool to model sales funnels, tracking customers from one part of the sales cycle to the next.
3. Marketing – Predicting Response vs. No Response to a marketing campaign. Very popular with database marketing professionals
4. Credit Cards & Loans – Predicting Default vs. Non-Default, popular with financial institutions to predict customer behavior
5. Churn – Predicting Remain or Leave, predicting if a customer will abandon a service, very popular with telecom companies and credit card companies.
6. Operations – Predicting Attrition vs. Retention, a handy tool to predict retention of students in college from one year to the next.
7. Websites – Predicting Click vs. No click. It is helpful to optimize website effectiveness.
8. Fraud identification – Fraud vs. Non-Fraud
9. Healthcare – Predicting Cure vs. No Cure. The modeling of clinical trial results for pharmaceuticals to cure disease

There are probably many more situations that are naturally binary. We may even take a numerical variable and make it binary by dividing the range of the variable into two bins and

trying to predict whether a customer's behavior will fall into one or the other of the bins. For example, consider the ubiquitous 5-star ratings so often used for products and services. We might bundle a 4- and 5-star rating into a "high" rating (1) and 1-, 2- and 3- star ratings into a "low" rating (0) and employ logistic regression to predict what causes a customer to rate us "high" or "low."

COMPARING LINEAR AND LOGISTIC REGRESSIONS FOR BINARY OUTCOMES

Linear models are less accurate in predicting binary outcomes. The residuals from the linear model are much greater than the residuals from the logistic model, as the plots in Figure 7.3 show.

Logistic models are a better fit for binary data. For example, when we process the results of the linear and logistic models for the *PassClass* data set, we see that the linear model has a 27% error rate, whereas the logistic model has a 23% error rate. If we look at the confusion matrix for both models, we see they have the same error rate in this case. The sample size may be too small to show a pronounced difference. More likely, this sample size has too many students scoring in the middle range of scores (not too low, not too high), where both machines need help with accurate predictions.

In this case, the probability of a student flunking the class, given the midterm score, is a valuable parameter. Such scoring allows us to focus our coaching on those most likely to flunk. We use the score to separate the students into three categories: those needing immediate help, those who may flunk but will probably recover on their own without intervention, and those who are doing well and need no attention. This demonstrates how the models may effectively be used (to score students or customers): rating from most likely to least likely. Then, we draw arbitrary demarcations to categorize our study subjects. Who is most likely to buy? We promote our product to them and ignore the rest. Which students are most likely to fail? We extend coaching efforts to those students for the most part.

Thus, for binary outcome situations, we are interested in scoring the likelihood and then sorting the subjects for the proper binary intervention. Knowing the accuracy of the model is necessary, but not that critical. A linear regression may score customers more accurately than a logistic regression model. Still, to separate customers into whom we will promote and whom we will not, the slight error difference in the models does not prompt us to switch from linear to logistic regression for modeling. For these likelihood scoring purposes, linear is often good enough. Although we see that logistic modeling is more accurate for binary outcomes with a lower predicted error, in practice, we use linear regression in most cases, which is good enough for scoring.

CASE STUDY 7.1: LINEAR REGRESSION USING THE SFO SURVEY DATA SET

If we are the SFO airport marketing director, we want to know what may cause people to be unhappy with the airport, so we will know what to fix to improve our standing among other airports. We want to explore the causes of passengers giving us low scores. This is where a linear regression model can come in very handy. For example, we will use the overall satisfaction score Q7ALL as our target or outcome variable. We will use the rest of the Q7 survey questions for input variables or features. The coefficients in the linear regression for the input variables

will give us the strength and direction of the influence of each of these on the outcome—a convenient guide as to where our attention should be placed to improve our overall score. We will also perform the analysis for another indicator of satisfaction, the net promoter score, variable *NETPRO*.

Ensure that both variables, *Q7ALL* and *NETPRO*, have been adequately prepared by removing any values that are not a rating (1–5 for *Q7ALL* and 1–10 for *NETPRO*). You may have done this already in another exercise.

The questions we will answer using linear regression for this case are as follows:

What is the relationship between Q7ALL and the rest of the Q7 question answers?

What does this relationship tell us about the factors influencing the overall score in Q7ALL?

What is the relationship between NETPRO and all the Q7 question answers?

What does this relationship tell us about the factors influencing the overall NETPRO score?

SOLUTION IN R

To ensure that variables *Q7* and *NETPRO* are adequately prepared, you should follow the data cleaning steps in Chapter 4 and remove all meaningless cells (values with a 6 or a 0).

Use the *SFO Q7 Data 2018 Q7 and NETPRO.xlsx* data file found in the *SFO Survey Data* folder of the *Case Data* depository. Once all variables are ready to analyze, import the well-prepared data set into Jamovi. Select *Correlation Matrix* under *Regression*, and select all variables we want to analyze to get the correlation matrix, as shown in Figure 7.20(a), (b), and (c).

Select *Linear Regression* under *Regression* and select all variables we want to analyze to get the linear regression, as shown in Figure 7.21.

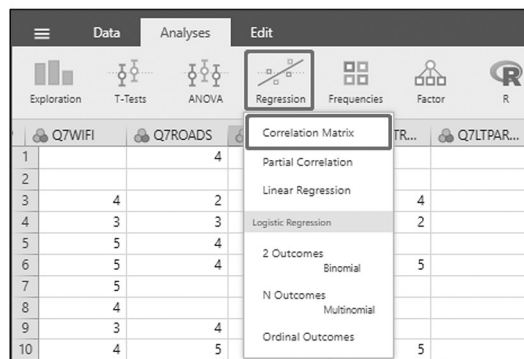


FIGURE 7.20(a) The first step to create the correlation matrix and result.

The results of the MLR model are shown in Figure 7.21. Notice that the overall correlation between *Q7ALL* and the other variables is high ($R=0.855$). This is somewhat expected from the Pearson correlations in Figure 7.21, since it appears that some of the predictors are not highly correlated to the target. The overall fit of the linear model is also medium (R -squared

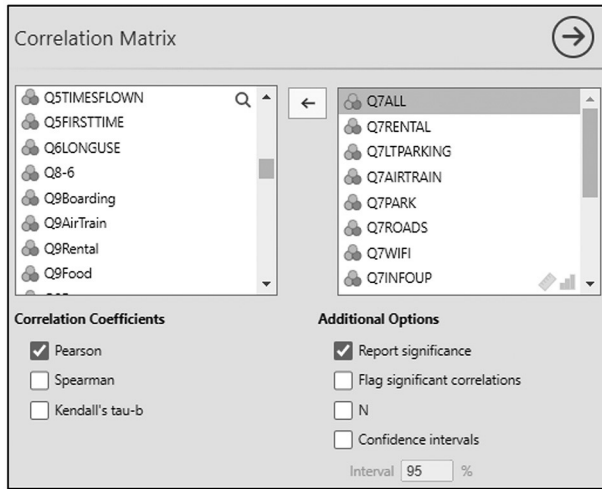


FIGURE 7.20(b) The second step to create the correlation matrix and result.

Correlation Matrix		Q7ALL	Q7RENTAL	Q7LTPARKING	Q7AIRTRAIN	Q7PARK	Q7ROADS	Q7WIFI	Q7INFOUP	Q7INFODOWN	Q7SCREENS	Q7WALKWAY	Q7SIGN	Q7STORE	Q7FOOD	Q7ART
Q7ALL	Pearson's r	—														
	p-value	—														
Q7RENTAL	Pearson's r	0.552	—													
	p-value	<.001	—													
Q7LTPARKING	Pearson's r	0.593	0.794	—												
	p-value	<.001	<.001	—												
Q7AIRTRAIN	Pearson's r	0.479	0.606	0.761	—											
	p-value	<.001	<.001	<.001	—											
Q7PARK	Pearson's r	0.585	0.678	0.703	0.674	—										
	p-value	<.001	<.001	<.001	<.001	—										
Q7ROADS	Pearson's r	0.502	0.547	0.602	0.485	0.670	—									
	p-value	<.001	<.001	<.001	<.001	<.001	—									
Q7WIFI	Pearson's r	0.414	0.370	0.448	0.349	0.347	0.367	—								
	p-value	<.001	<.001	<.001	<.001	<.001	<.001	—								
Q7INFOUP	Pearson's r	0.579	0.533	0.617	0.466	0.565	0.499	0.422	—							
	p-value	<.001	<.001	<.001	<.001	<.001	<.001	<.001	—							
Q7INFODOWN	Pearson's r	0.553	0.555	0.612	0.462	0.565	0.469	0.406	0.932	—						
	p-value	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	—						
Q7SCREENS	Pearson's r	0.555	0.498	0.501	0.431	0.516	0.476	0.357	0.687	0.669	—					
	p-value	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	—					
Q7WALKWAY	Pearson's r	0.526	0.520	0.494	0.455	0.522	0.470	0.320	0.580	0.574	0.643	—				
	p-value	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	—				
Q7SIGN	Pearson's r	0.566	0.456	0.469	0.425	0.508	0.486	0.319	0.604	0.572	0.623	0.610	—			
	p-value	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	—		
Q7STORE	Pearson's r	0.562	0.359	0.457	0.303	0.409	0.364	0.316	0.485	0.473	0.405	0.415	0.432	—		
	p-value	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	—	
Q7FOOD	Pearson's r	0.570	0.294	0.416	0.333	0.385	0.288	0.296	0.469	0.455	0.377	0.368	0.366	0.740	—	
	p-value	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	—
Q7ART	Pearson's r	0.521	0.329	0.393	0.314	0.383	0.304	0.226	0.425	0.409	0.378	0.400	0.404	0.490	0.516	—
	p-value	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001

FIGURE 7.20(c) The third step to create the correlation matrix and result.

= 0.730), which says that the model explains 73% of the variability in *Q7ALL*. The p-values for most of the model coefficients are above our alpha criterion of 0.05. Only five out of 13, *Q7FOOS*, *Q7SIGN*, *Q7WIFI*, *Q7ROADS*, and *Q7PARKING*, have p-values below 0.05. It would be helpful to create this model using only the predictors with low p-values and see if we develop a better model.

Let us now analyze *NETPRO* as the target against all the *Q7* predictors. Follow similar steps and get the correlation matrix and linear regression between *NETPRO* and all *Q7* questions. The result is shown in Figure 7.22.

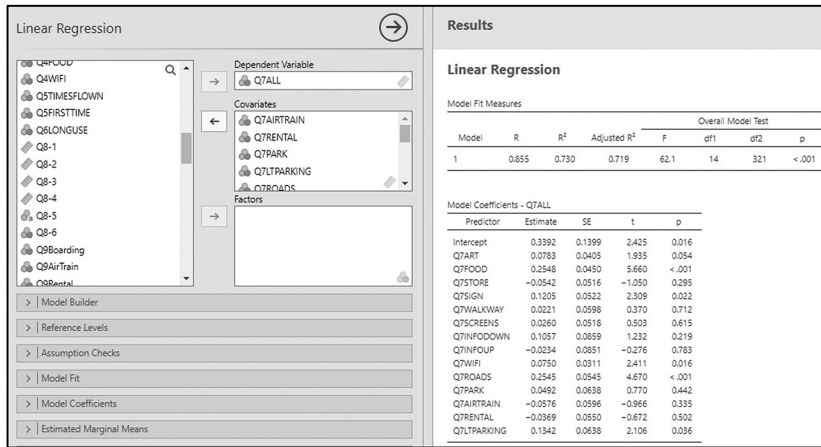


FIGURE 7.21 Creating the linear regression and result.

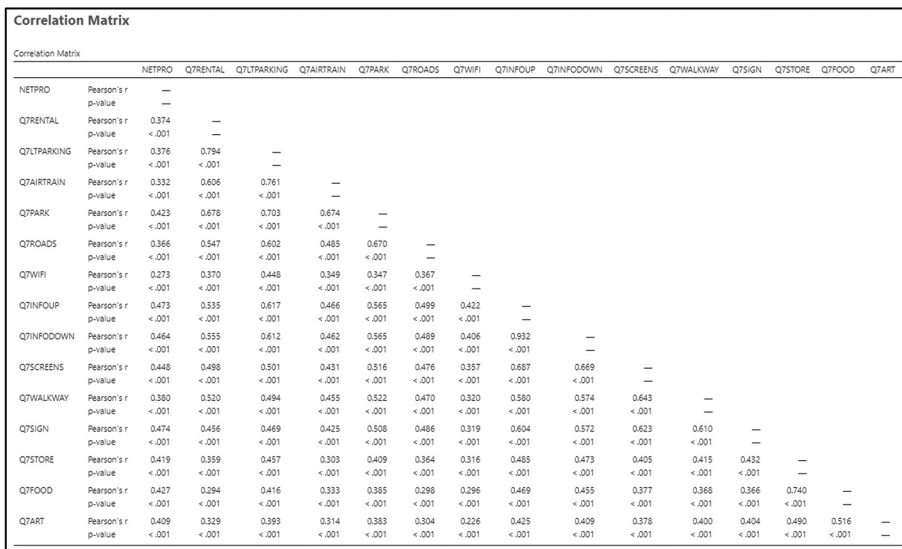


FIGURE 7.22 Correlation matrix with NETPRO included.

Similar to the previous question, having *NETPRO* as the dependent variable will provide the coefficients table to use; see Figure 7.23.

The results of the MLR model are shown in Figure 7.23. Notice that the overall correlation between *NETPRO* and the other variables is moderate ($R=0.665$), as expected, since some predictors are not highly correlated to the target. The overall fit of the linear model is also low ($R\text{-squared} = 0.429$), which says that the model explains less than 50% of the variability in *NETPRO*. The p-values for some of the model coefficients are above our alpha criterion of 0.05, and some are below, as expected. It would be helpful to create this model using only the highly correlated predictors and see if we develop a better model.

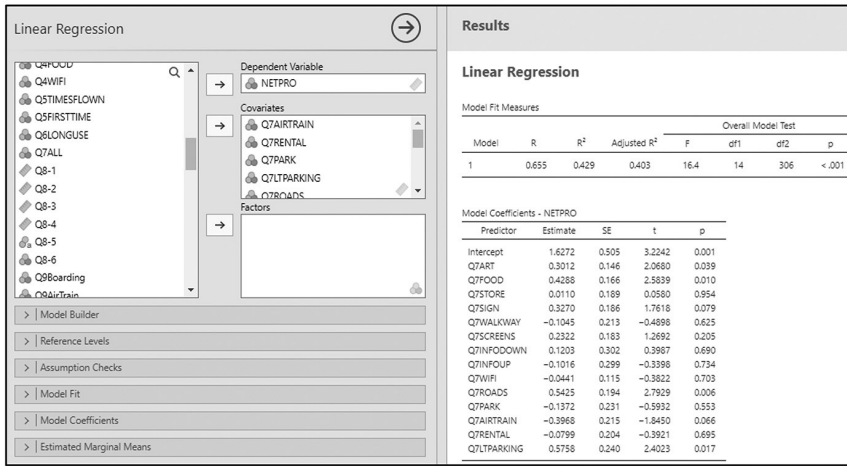


FIGURE 7.23 Steps to create a linear regression with NETPRO included.

SOLUTION IN PYTHON

Import the same well-prepared data set into Orange3. Use the *Select Column* widget to select all Q7 question answers and connect it to a new widget, *Correlations* (see Figure 7.24).

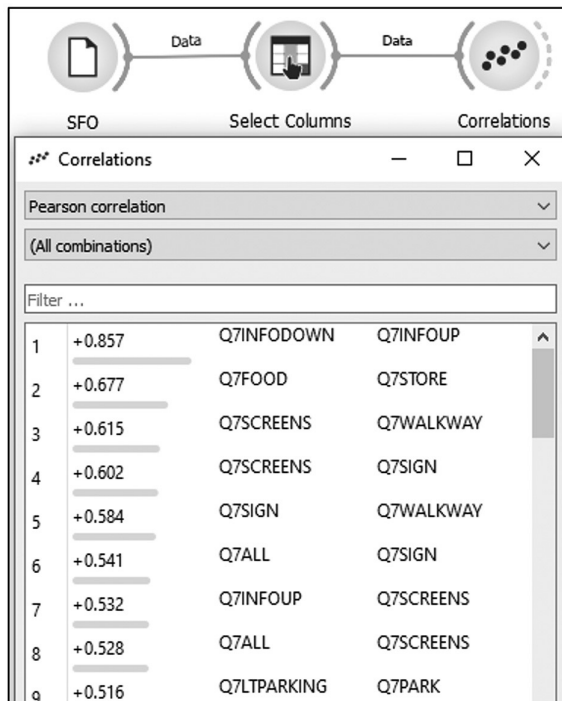


FIGURE 7.24 Steps to create a correlation matrix in Orange3.

To obtain the linear regression, we should set *Q7ALL* as a target in the step of the select column. Then, connect to a *Linear Regression* widget and get the coefficients table. The steps and results are shown in Figure 7.25.

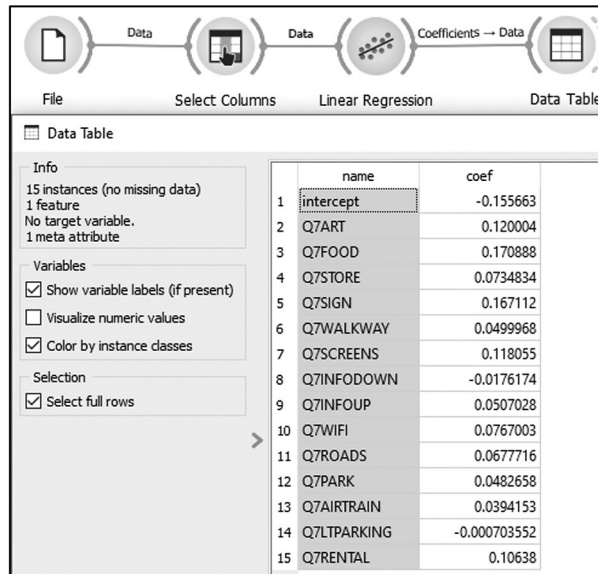


FIGURE 7.25 Steps to create linear regression in Orange3.

Following similar steps and setting *NETPRO* as the target variable, you can get the correlation matrix and coefficients table linear regression as shown in Figure 7.26(a) and (b).

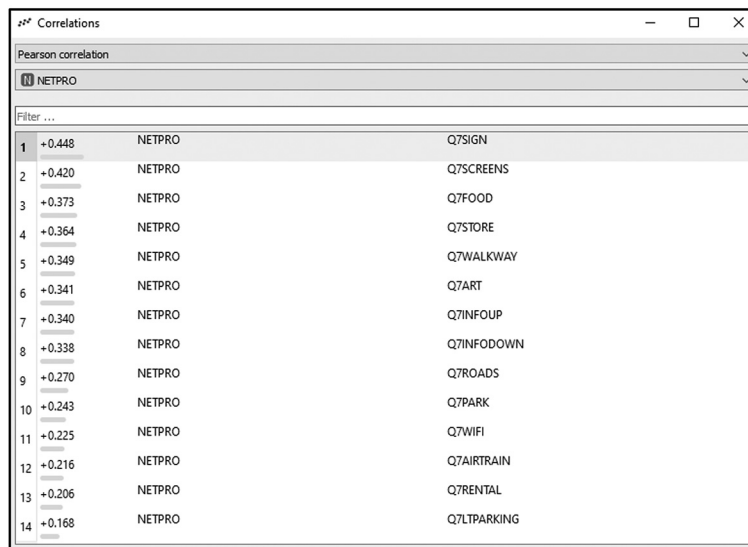


FIGURE 7.26(a) Correlation matrix and linear regression with *NETPRO* included.

	name	coef
1	intercept	0.493362
2	Q7ART	0.249175
3	Q7FOOD	0.322838
4	Q7STORE	0.15305
5	Q7SIGN	0.52674
6	Q7WALKWAY	-0.0605419
7	Q7SCREENS	0.359117
8	Q7INFODOWN	0.151935
9	Q7INFOUP	0.00401948
10	Q7WIFI	0.0638802
11	Q7ROADS	0.0917632
12	Q7PARK	0.10139
13	Q7AIRTRAIN	0.0417515
14	Q7LTPARKING	-0.153957
15	Q7RENTAL	0.0925352

FIGURE 7.26(b) Correlation matrix and linear regression with *NETPRO* included.

CASE STUDY 7.2: LINEAR REGRESSION USING THE SBA LOANS DATA SET

Imagine yourself in the job of Small Business Administration Commissioner. You must answer the US Congress on how well your loan program is performing. After all, they appropriated money for it and they wanted to see it well spent. You ask your analyst to tell you how well the loan program is doing. For example, it would be good to know if the size of the company is a predictor of the loan size. In other words, are we lending more money to companies with more employees (an index of company size and the number of workers supported by the SBA program)? Use some of the framed analytical questions we discovered in Chapter 3:

What is the relationship between loan size and the number of employees in the company?

Is there a relationship between the size of the loan and the term of the loan? In other words, does the SBA give you more time to repay a loan if you ask for more money?

For the input variable or feature, we will use *GrossAmount*. We will use *Jobs Supported* variable for the first question and *TermInMonths* for the second question. The coefficients of the linear regression will give us the strength and direction of the influence of each of these on the outcome variable.

SOLUTION IN R

Use the *FOIA Loans Data.xlsx* data file found in the *SBA Loans Data* folder of the *Case Data* depository. Make sure to only use the data in the *PIF Loans* tab. To ensure the needed variables are adequately prepared, you should follow the data cleaning steps in Chapter 4 and remove all meaningless cells. Once all variables are ready for analysis, import the well-prepared data set into Jamovi.

Select *Correlation Matrix* under *Regression*, as well as *GrossApproval* and *JobsSupported*, which we want to analyze to get the correlation matrix. The steps and results are shown in Figure 7.27.

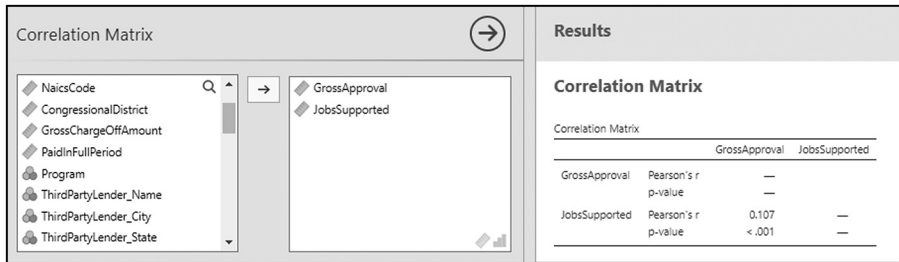


FIGURE 7.27 Correlation matrix between firm size and gross approval amount.

To answer the first question, we see that there is a very small correlation between firm size (indicated by *JobsSupported*) and loan size. Thus, linear regression model would not be a productive way of predicting whether there is a relationship between loan amount and firm size.

To answer the second question, we run a linear regression on *GrossApproval* as the target and *PaidInFullPeriod* as the predictor to get the coefficients table, as shown in Figure 7.28.

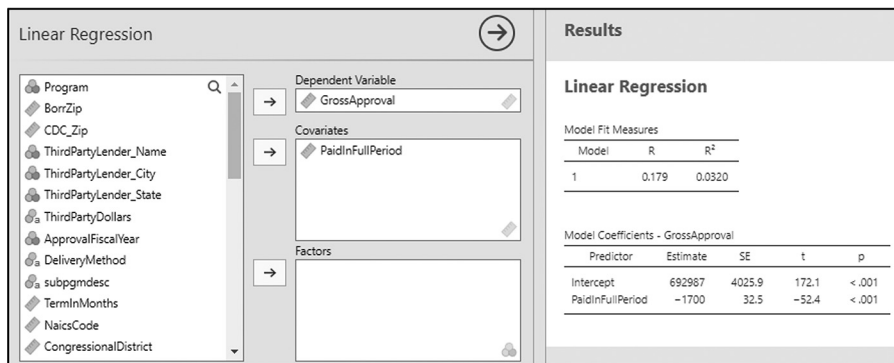


FIGURE 7.28 Linear regression between *PaidInFullPeriod* and *GrossApproval* amount

The results of the SLR model are shown in Figure 7.28. Notice that the overall correlation between *PaidInFullPeriod* and *GrossApproval* is very low ($R = 0.179$), as expected. The overall fit of the linear model is also inferior ($R\text{-squared} = 0.0320$). The p-values of the model coefficient are above our alpha criterion of 0.05, which says that all the predictors used are highly unrelated to the target and should be discarded, as expected. This is not a very good model and should not be used.

SOLUTION IN PYTHON

Import the well-prepared data set into Orange3. Use the *Select Column* widget to select the variables *GrossApproval* and *JobsSupported*. Then, connect to a new widget, *Correlations*, as shown in Figure 7.29.

Go back to the *Select Column* widget to change *JobsSupported* to *PaidInFullPeriod*. Be sure to set *GrossApproval* as the target variable.

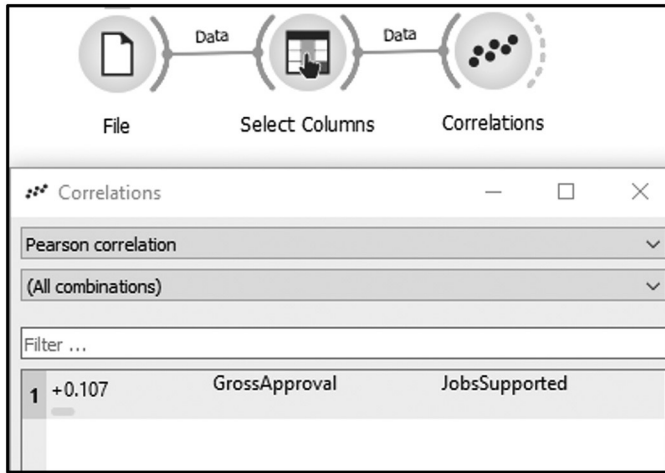


FIGURE 7.29 Correlation between gross approval amount (signified by variable *GrossApproval*) and firm size (signified by variable *JobsSupported*).

Run the *Linear Regression* function, and create a table for coefficients to get the result shown in Figure 7.30.

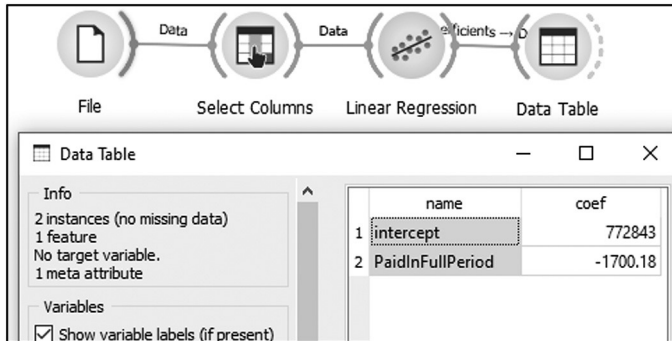


FIGURE 7.30 Linear regression between between gross approval amount (signified by variable *GrossApproval*) and firm size (signified by variable *JobsSupported*).

CASE STUDY 7.3: LOGISTIC REGRESSION USING THE SFO SURVEY DATA SET

Continuing with our study of the overall ratings the SFO passengers give the airport, let us approach it from a binary point of view, which can be analyzed using logistic regression. We can bin the *Q7ALL* variable into two categories. This first is “Do you like this airport?,” where “Yes” = 5, 4, and “No” = 3, 2, 1; we replace the 6 and 0 scores with blanks. Likewise, we can bin the Net Promoter Score, *NETPRO*, into two categories: “Would you recommend this airport to your friends?,” where for “Yes,” *NETPRO* = 10, 9, 8, and for “No,” *NETPRO* = all other numbers; and we leave a blank for any other response. We will predict their response based on their other responses to other airport features (such as food and parking) and the other *Q7* answers as features or input variables.

The question we will answer using logistic regression for this case is

Can we predict what factors are most important for customers to give us a good score?

SOLUTION IN R

Use the *SFO Q7 Data 2018 Q7 and NETPRO.xlsx* data file found in the *SFO Survey Data* folder of the *Case Data* depository and modify it to create variables to bin *Q7ALL* and *NETPRO* variables as explained above. To ensure the variables *Q7* and *NETPRO* are adequately prepared, you should follow the data cleaning steps in Chapter 4, remove all meaningless cells, and change all others into two categorical values, “Yes” and “No.”

Once you have all the data well-prepared, open the data set in Jamovi, click on the *Regression* tab, and then on two outcomes below the *Logistic Regression* minor header. The results are shown in Figure 7.31(a) and (b).

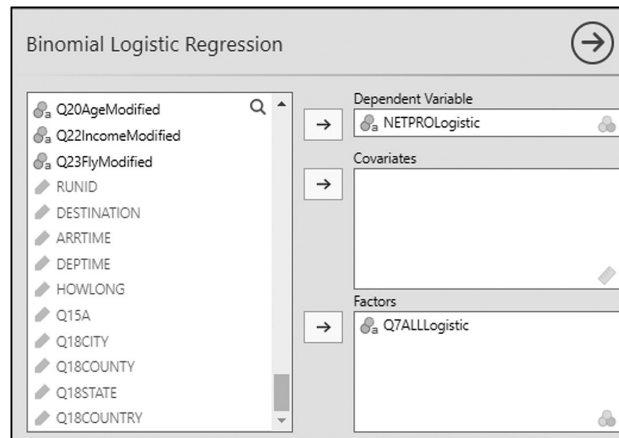


FIGURE 7.31(a) The first step to create the logistic regression in Jamovi.

Binomial Logistic Regression				
Model Fit Measures				
Model	Deviance	AIC	R^2_{McF}	
1	2430	2434	0.193	
Model Coefficients - NETPROLogistic				
Predictor	Estimate	SE	Z	p
Intercept	-0.982	0.0974	-10.1	< .001
Q7ALLLogistic:				
Yes - No	2.548	0.1142	22.3	< .001
Note. Estimates represent the log odds of "NETPROLogistic = Yes" vs. "NETPROLogistic = No"				

FIGURE 7.31(b) The second step to create the logistic regression in Jamovi.

SOLUTION IN PYTHON

Use the *SFO Q7 Data 2018 Q7 and NETPRO.xlsx* data file found in the *SFO Survey Data* folder of the *Case Data* depository and modify it to create variables to bin *Q7ALL* and *NETPRO* variables as explained above. To ensure the variables *Q7* and *NETPRO* are adequately prepared, you should follow the data cleaning steps in Chapter 4, remove all meaningless cells, and change all others into two categorical values, “Yes” and “No.”

Once you have all the data well-prepared, open the data set in Orange3 and follow the steps to select columns and then add a new widget, *Logistic Regression*, to run the regression. Connect to a *Data Table* widget and view the result (see Figure 7.32).

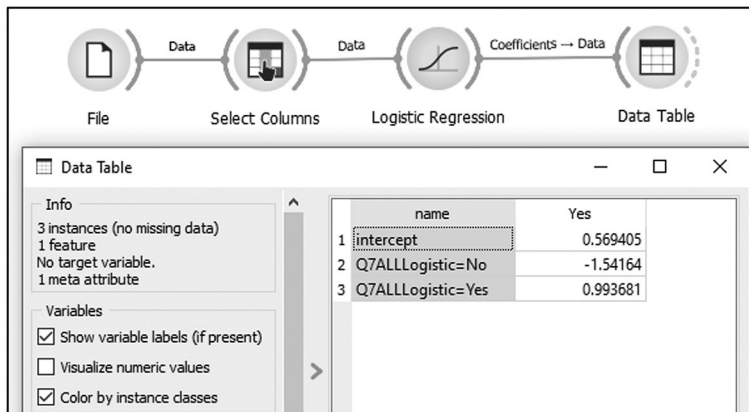


FIGURE 7.32 Steps to create logistic regression in Orange3.

CASE STUDY 7.4: LOGISTIC REGRESSION USING THE SBA LOANS DATA SET

Continuing with our study of the SBA loans’ overall performance, let us now approach it from a binary point of view, which can be analyzed using logistic regression. We can bin the *LoanStatus* variable into two categories. The first is “Was the loan paid off, or did they default?,” where “Yes” = PIF, “No” = CHGOFF, and we leave empty all other status categories. We will use *GrossAmount*, the loan size, *TermInMonths*, and *Jobs Supported* as the input variables.

The question we will answer using logistic regression for this case is

Can we predict what factors are most important for success in paying back a loan?

SOLUTION IN R

To make sure the variables are correctly prepared, you should follow the data cleaning steps in Chapter 4 and bin the *LoanStatus* variable into two categories. These answer the question “Was the loan paid off or did they default?” The categories are YES = PIF, NO = CHGOFF, and leave empty all other status categories.

Once you have all the data well-prepared, open the data set in Jamovi, click on the *Regression* tab, and then on two outcomes below the *Logistic Regression* minor header. The results are shown in Figure 7.33.

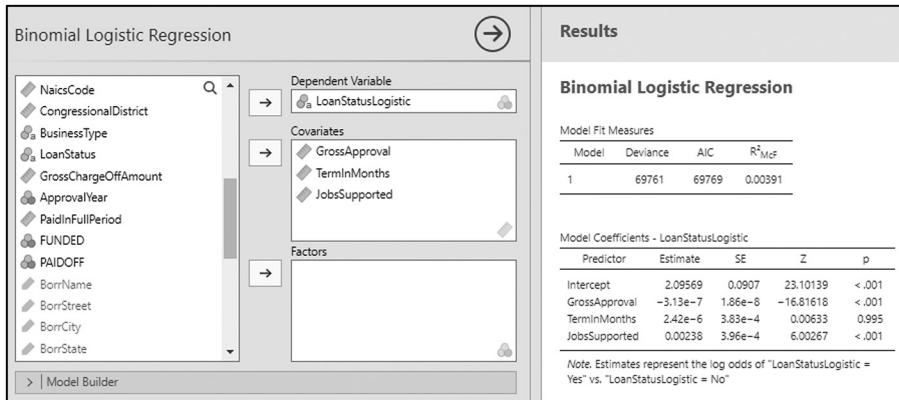


FIGURE 7.33 Logistic regression on loan status, gross approval amount, term in a month, and firm size Jamovi.

SOLUTION IN PYTHON

To make sure the variables are correctly prepared, you should follow the data cleaning steps in Chapter 4 and bin the *LoanStatus* variable into two categories. The first is “Was the loan paid off or did they default?” The categories are YES = PIF, NO= CHGOFF, and leave empty all other status categories.

Once you have all the data well-prepared, open the data set in Orange3 and follow the steps to select columns and then add a new widget, *Logistic Regression*, to run the regression. Connect to a *Data Table* widget and view the results (see Figure 7.34).

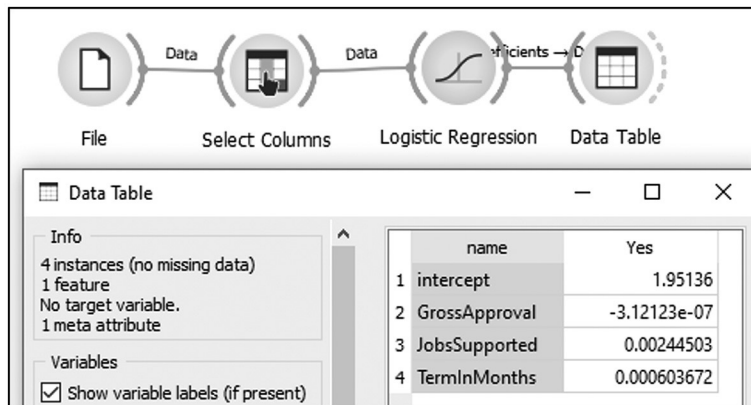
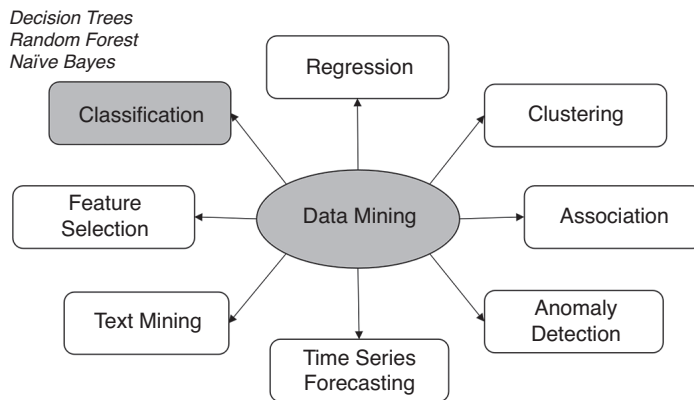


FIGURE 7.34 Logistic regression on loan status, gross approval amount, term in the month, and firm size in Orange3.

CLASSIFICATION



Classification is the process of organizing data by relevant categories to make it easy to find, store, and analyze. Automated data classification uses machine learning algorithms to classify unseen data using predefined tags. Once a categorization scheme is found, the logic to arrive at any one category becomes the model to categorize future candidates to be added to the data set. In other words, it can be used to classify or predict where a future member of the data set belongs.

In a way, any categorical variable or feature of a data set can be used to categorize the rows of the data set into categories. If that categorical variable becomes the label or predicted variable and the rest of the variables in the data set are the predictors or features, ascertaining where a new data row belongs according to the predicted class using the rest of the features is a classification problem. This problem is amenable to machine learning solutions, which are considered classification algorithms. Two such classification algorithms we explore in this chapter are decision trees (and their extension as random forest) and Naïve Bayes, as shown in Figure 8.1.

Let's look at a simple example. Consider that you are running a restaurant, and you decide that there are many dishes you want to offer your customers. The dishes have many characteristics, which we enter into a database. One such feature is the kind of dish. Is it an appetizer, a

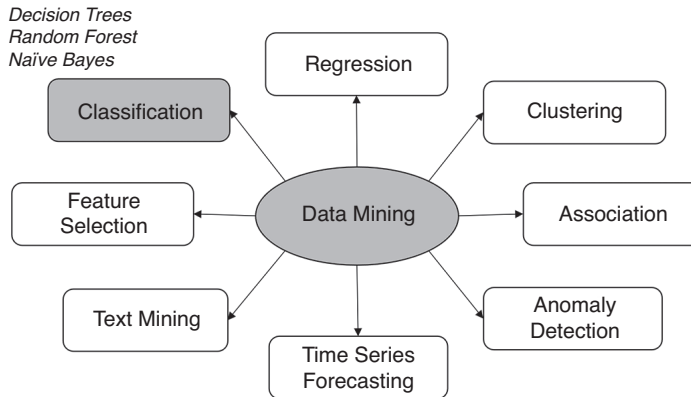


FIGURE 8.1 Classification is an important data mining machine learning tool. Three prominent examples of classifiers are decision trees, random forest, and Naïve Bayes algorithms.

dessert, or a main dish? Some of the other features might tabulate the number of calories, level of spiciness, temperature at which it is served, and some of the ingredients. We then create a menu using the type of dish as a primary classifier, as is typically done with menus. Now we want to introduce a new dish. Its characteristics are entered into the database, but not the type of dish. If you had a model that categorizes and predicts what kind of menu item this could be classified as, you would be in good shape. Note that given that we have an outcome variable we are trying to predict, this classifier can be classed under supervised machine learning.

CLASSIFICATION WITH DECISION TREES

A *decision tree* is a decision support tool that uses a tree-like model of decisions and their possible consequences, such as chance event outcomes. It is one way to display the logic of an algorithm that only contains conditional control statements. A decision tree is often drawn as a flowchart-like structure in which each internal node represents a “test” on an attribute (e.g., whether a coin flip comes up with heads or tails), and each branch represents the outcome of the test. Each leaf node represents a class label (decision taken after computing all attributes). The paths from the root to the leaf represent classification rules. Figure 8.2 shows the decision tree to classify who survived and who did not survive in the Titanic disaster. Note the labeling of the tree elements and where the decision points are. If we had the characteristics of an unknown passenger (gender, age, and class they were traveling under), we could predict their survival chances. Later in this chapter, we show how the tool would be used to compute such probability.

BUILDING A DECISION TREE

Using examples is an excellent way to learn how decision trees are built. Suppose you own a computer store, so you collect information about customers who walk through the door and you note who buys a computer and who does not. Figure 8.3 shows the collection of a dozen or more customers, their information, and the classifier outcome variable of whether they made a purchase or not.

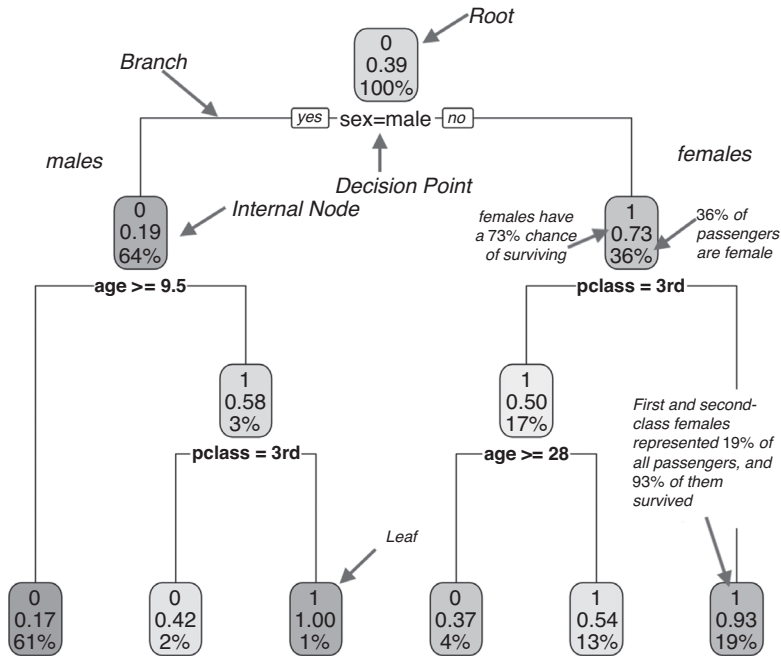


FIGURE 8.2 Decision tree for passengers of the Titanic, showing how the classification process works and the labels of the tree characteristics.

Student	Income	Credit	Age	Buys
No	High	Fair	<30	No
No	High	Excellent	<30	Yes
No	High	Fair	30...40	Yes
No	Medium	Fair	>40	Yes
Yes	Low	Fair	>40	Yes
Yes	Low	Excellent	>40	No
Yes	Low	Excellent	30...40	Yes
No	Medium	Fair	<30	No
Yes	Low	Fair	<30	No
Yes	Medium	Fair	>40	Yes
Yes	Medium	Excellent	<30	Yes
No	Medium	Excellent	30...40	Yes
Yes	High	Fair	30...40	Yes
No	Medium	Excellent	>40	No

FIGURE 8.3 Computer store customer database. Note that the right-most column with the variable *Buys* is the predicted or labeled variable.

We can compute the frequencies for each categorical variable using frequency analysis in Jamovi, which will allow us to build the percentages for each tree’s branches and leaves. We have used Jamovi’s descriptive statistics analysis to get the frequency of appearance of each category in each variable. Figure 8.4 shows such a computed table of frequencies.

Frequency Tables ▼

Frequencies for Student

Student	Frequency	Percent	Valid Percent	Cumulative Percent
No	7	50.000	50.000	50.000
Yes	7	50.000	50.000	100.000
Missing	0	0.000		
Total	14	100.000		

Frequencies for Age

Age	Frequency	Percent	Valid Percent	Cumulative Percent
30...40	4	28.571	28.571	28.571
<30	5	35.714	35.714	64.286
>40	5	35.714	35.714	100.000
Missing	0	0.000		
Total	14	100.000		

Frequencies for Income

Income	Frequency	Percent	Valid Percent	Cumulative Percent
High	4	28.571	28.571	28.571
Low	4	28.571	28.571	57.143
Medium	6	42.857	42.857	100.000
Missing	0	0.000		
Total	14	100.000		

Frequencies for Buys

Buys	Frequency	Percent	Valid Percent	Cumulative Percent
No	5	35.714	35.714	35.714
Yes	9	64.286	64.286	100.000
Missing	0	0.000		
Total	14	100.000		

Frequencies for Credit ▼

Credit	Frequency	Percent	Valid Percent	Cumulative Percent
Excellent	6	42.857	42.857	42.857
Fair	8	57.143	57.143	100.000
Missing	0	0.000		
Total	14	100.000		

FIGURE 8.4 Using Jamovi to compute the percentages for all of the possible outcomes as a way to build our decision tree.

One possible resulting decision tree is shown in Figure 8.5. Note the translation of the percentages from the tables in Figure 8.4 into the values in the tree branches in the diagram in Figure 8.5.

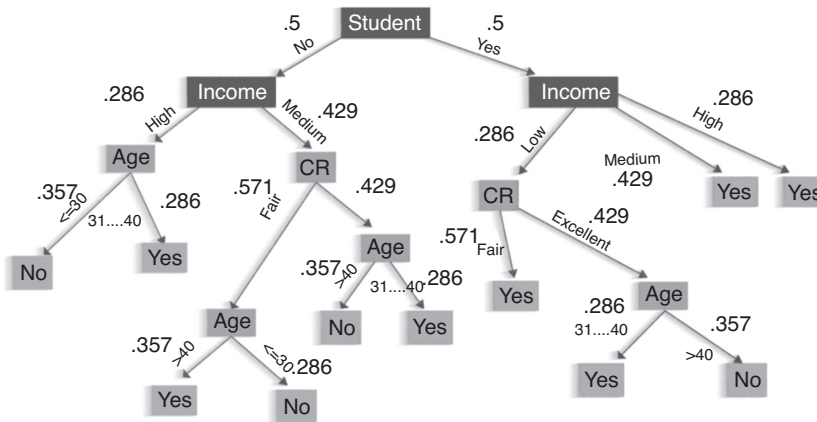


FIGURE 8.5 The annotated decision tree.

We now have a decision tree classifier-based model we can use to answer questions about customers as they come into our store. First, let's see the likelihood that a new customer walking through the door would buy a computer. Go down the tree for all possible paths, multiplying the percentages to get all the possible "yes" and "no" answers. Figure 8.6 shows the results. We see there is a 64% chance the new customer would buy, given the history of those who walk through our store door.

Let's use our classifier to compute the likelihood that a customer with specific characteristics would buy a computer. In other words, would we classify them as a buyer or not? You can see how this can be very useful to classify customers for marketing purposes, for example. Figure 8.7 shows the walk down the tree and how to compute the resulting purchase probability for a low-income student between the ages of 30 and 40 with an excellent credit rating.

It is less than a 2% chance.

Student		Income		Credit rating		Age		Likelihood	
Yes	0.5	High	0.286					Yes	0.143
Yes	0.5	Medium	0.429					Yes	0.215
Yes	0.5	Low	0.286	Fair	0.571			Yes	0.082
Yes	0.5	Low	0.286	Excellent	0.571	31-40	0.286	Yes	0.023
Yes	0.5	Low	0.286	Excellent	0.571	>40	0.357	No	0.029
No	0.5	High	0.286			<30	0.357	No	0.051
No	0.5	High	0.286			31-40	0.286	Yes	0.041
No	0.5	Medium	0.429	Fair	0.571	>40	0.357	Yes	0.077
No	0.5	Medium	0.429	Fair	0.571	<30	0.286	No	0.061
No	0.5	Medium	0.429	Excellent	0.429	>40	0.357	No	0.077
No	0.5	Medium	0.429	Excellent	0.429	31-40	0.286	Yes	0.061
								Yes	0.641
								No	0.218

FIGURE 8.6 Using the decision tree to compute the likelihood of a new customer buying a computer.

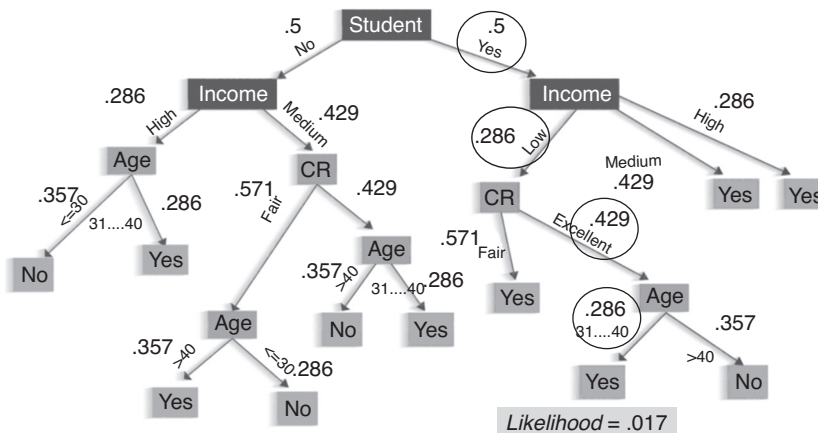


FIGURE 8.7 The computed probability of a purchase probability by a low-income student between the ages of 30 and 40 with an excellent credit rating.

Notice one more interesting thing. We can predict an outcome using only categorical variables, something we could not do with the regression models of the last chapter unless we did some fancy binning. Decision trees can handle numeric and categorical variables for features and labels. If the variable we try to predict is categorical, we call the classifier a decision tree. If the outcome variable is numeric, we call it a regression tree. When using JASP for building decision trees, you will find each type of classifier under different menus. Now, let's see how decision tree classifiers are used for other data sets.

EXERCISE 8.1 – THE IRIS DATA SET

We will play botanist for this exercise. Say you are an experienced botanist specializing in irises. You are such an expert that you can recognize the genus and species of any iris in the world. You want to train your students to achieve that level of mastery. You decide to train a

machine to predict the type of iris by measuring the characteristics of the flower. That way, when a student attempts to categorize an iris in the wild, they can ascertain whether they get it right or not by comparing it to the results using the machine. We are building a robot tutor. Figure 8.8 shows the elements of an iris we will measure to characterize each flower.

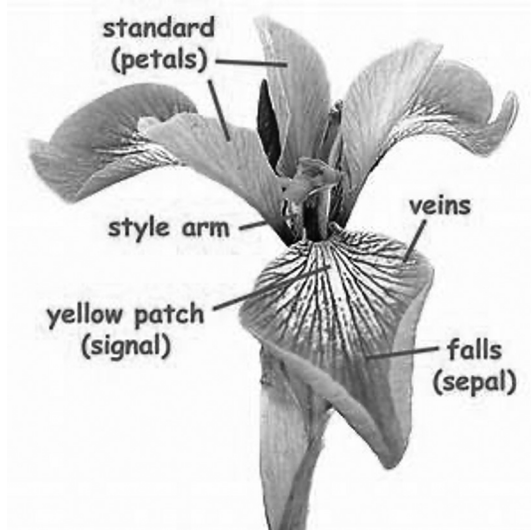


FIGURE 8.8 Characteristic parts of an iris.

As the expert botanist, you compile a training set from flower measurements and your expert categorization of each species. The iris data set (Dua 2019) (use the *iris.csv* file, found in the *Chapter 8* folder in the *Case Data* depository) contains four features (length and width of the sepals and petals) of 50 samples each of three species of iris (Iris Setosa, Iris Virginica, and Iris Versicolor). The three types of irises being classified have measured petal widths and lengths and sepal lengths and width. The descriptive statistics of each type of iris (Figure 8.9) show that

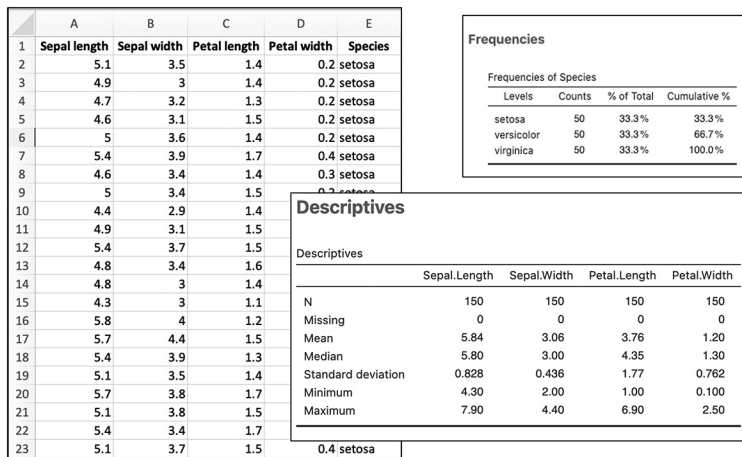


FIGURE 8.9 The Iris database's descriptive statistics.

average petal length and width are distinctive for each species (there is little overlap of the distributions). However, there is considerable overlap of the sepal length and width distributions. We will see that decision trees based on all four characteristics are not as accurate as decision trees that use only the petal variables. Figure 8.10 shows this peculiar effect.

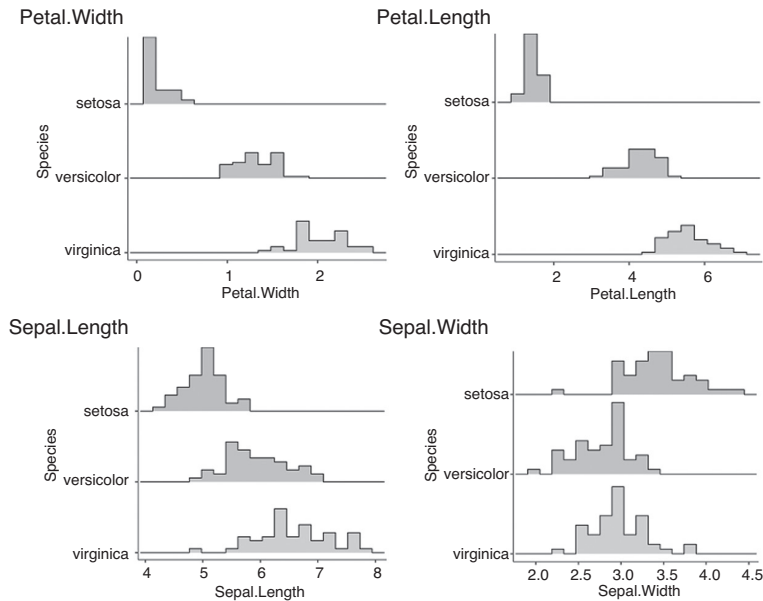


FIGURE 8.10 The Iris database's characteristics of petal length and width and sepal length and width by specie type, showing overlaps in certain distributions.

We use Jamovi to create a decision tree (Figure 8.11). Note that the two most important features for classification are petal length and petal width, as noted from the distribution separations above. The algorithm that recursively creates the tree searches for each variable that tends

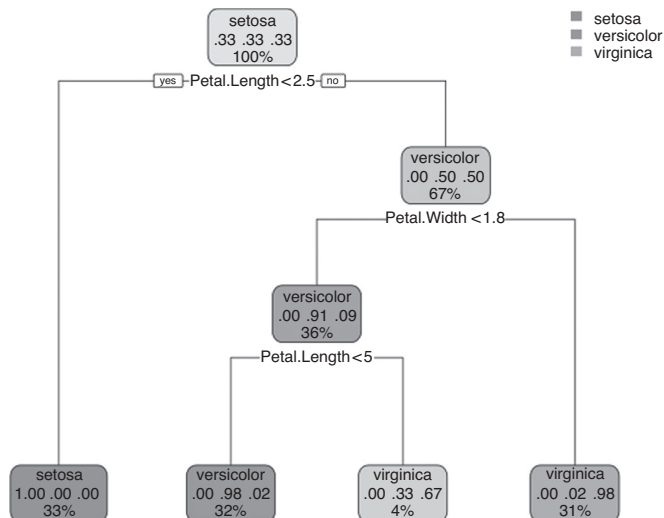


FIGURE 8.11 The decision tree model of the Iris database characteristics classifying the three species.

to separate the species from each other the best, noting the variable's value for the split and the resulting frequencies on each side of the split.

Let's apply the decision tree to recognize the mystery iris shown in Figure 8.12. The sepal width = 3 cm, the sepal length = 7 cm, the petal width = 1 cm, and the petal length = 4 cm. Using just the petal length and width measurements, we can categorize this as Iris Versicolor. This is a Northern Blue Flag iris categorized by botanists as belonging to the Iris Versicolor species.



FIGURE 8.12 Mystery iris, with sepal width=3 cm, sepal length=7 cm, petal width=1 cm, and petal length=4 cm.

THE PROBLEM WITH DECISION TREES

Decision trees are sensitive to the specific data on which they are trained. If the training data is changed, the resulting decision tree can be quite different, and in turn, the predictions can be quite different. Decision trees are also computationally expensive to train, carry a significant risk of overfitting, and tend to find local optima because they cannot go back after they have split.

These weaknesses are addressed using a random forest approach, which combines many decision trees into one model. Decision trees are sensitive to the data and the variables selected, and so they easily overfit. The Random Forest algorithm overcomes these limitations of simple decision trees.

CLASSIFICATION WITH RANDOM FOREST

A *random forest* model is built by constructing a multitude of decision trees at training time and outputting, during computation, the class that is the average of the classes (classification) or mean prediction (regression) of the individual trees. The ensemble of trees (the forest) computes by averaging the outcomes of each tree, producing a classification and issuing their vote on what the answer should be. The random forest machine polls all the trees and averages their vote as the final answer. The result is more stable (adding more rows to the training data does not change the results appreciably, for example), and the resulting model is less prone to overfitting.

How is such a forest of trees built? Let's say there is a data set with N rows of data. We start by selecting R ($R < N$) of the training data rows (maybe 60%) randomly, keeping aside the others (40%) to check for accuracy (validating, sometimes called *out-on-bag* data). That builds the

first tree. We repeat the process by randomly selecting another R number of rows and building a second tree. We check for overfitting on the validating 40% of rows for every tree and create as many trees as necessary until the overall accuracy rates for the training and validation data are comparable. We stop when the overfitting does not appreciably decrease. We now have a set of trees (a forest of trees from randomly selected rows for every tree = random forest). Then we use the model by averaging the results of using each tree to process an unknown. Figure 8.13 shows both a single decision tree solution and a set of four decision trees created from randomly selected subsets of the data used for the whole tree. The A and B dots are the resulting predicted answers by the branches of each tree. They represent the decision rules of each tree.

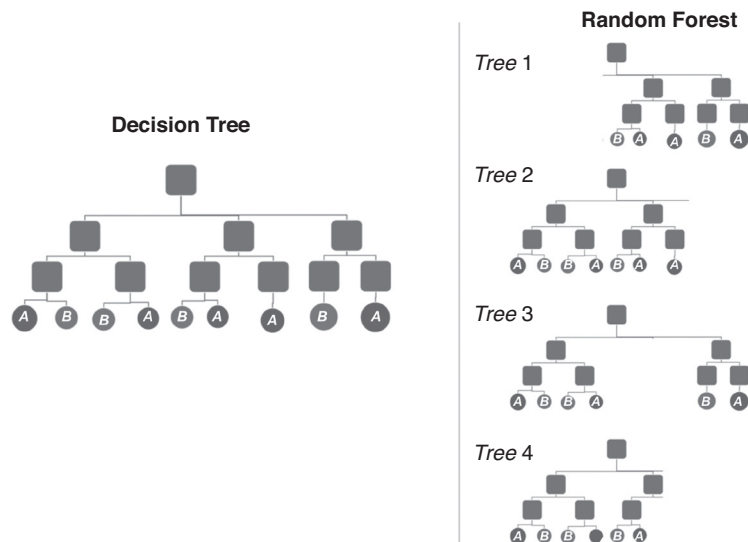


FIGURE 8.13 The difference between decision trees and random forest algorithms. Note that random forest comprises many trees from randomly selected data rows.

USING A RANDOM FOREST MODEL

We saw earlier how, once trained, a decision tree model produces a classification or prediction based on some unknown characteristics. The left portion of Figure 8.14 shows how a single tree model would solve the parameters of some unknown population member. On the other hand, a random forest model would process the same parameters to document as many tree answers as we have trees and then average (for regression) or vote (for classification) to arrive at a more robust solution. The right side of Figure 8.14 shows how the random forest model would arrive at (probably) a much better answer.

EXERCISE 8.2 – THE IRIS DATA SET

Now let's apply the random forest technique to the Iris data set (use the *iris.csv* file, found in the *Chapter 8* folder in the *Case Data* depository). We use JASP since it will allow us to perform a random forest analysis. We set the train/test ratio to be 80/20%. We also set the train/validation



FIGURE 8.14 How a decision tree arrives at one answer, compared to random forest trees, which arrives at many answers aggregated in a voting function to arrive at a better answer when applying the decision rules to an unknown.

ratio of the training rows to be 80/20%. That leaves 96 rows out of 120 for training and 24 rows for validation (out-of-bag). Figure 8.15 shows the setup parameters of the model and some of the results.

Random Forest Classification ▼

Random Forest Classification								
Trees	Predictors per split	n(Train)	n(Validation)	n(Test)	Validation Accuracy	Test Accuracy	OOB Accuracy	
14	2	96	24	30	1.000	0.967	0.857	

Note. The model is optimized with respect to the *out-of-bag accuracy*.

Data Split

Train: 96	Validation: 24	Test: 30	Total: 150
-----------	----------------	----------	------------

Confusion Matrix

		Predicted		
		setosa	versicolor	virginica
Observed	setosa	13	0	0
	versicolor	0	10	1
	virginica	0	0	6

FIGURE 8.15 The random forest model of the Iris database characteristics built with the JASP tool classifying the three species.

Although we set up the maximum number of trees for iteration to 100, it came to a reasonable solution with 14 trees. Figure 8.16 shows the reduction in overfitting as the number of trees in the model increases. Note that once it reached 14 trees, the model’s accuracy using the validation data is higher than the accuracy using the training data, and we stop adding trees. Note carefully in Figure 8.15 the lower table called the “Confusion Matrix.” That compares the actual

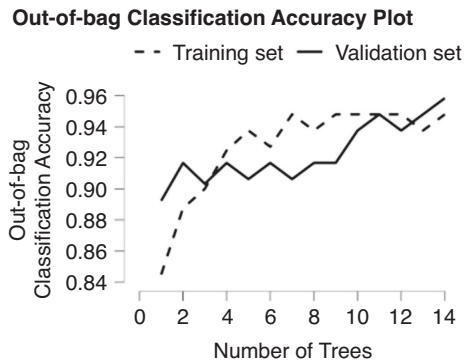


FIGURE 8.16 Out-of-bag accuracy plot showing that once the number of trees reaches 14, the model’s accuracy using validation data exceeds that of the training set.

to the predicted outcome for the 30 test rows of data. It is a measure of overfitting. We see that we have an excellent result, with only one error out of 30 predictions.

CLASSIFICATION WITH NAÏVE BAYES

Naïve Bayes is another popular classification algorithm. It is used when the output variable is discrete. It can seem daunting because it requires prior mathematical knowledge of conditional probability and the Bayes Theorem, but it is a straightforward and “naïve” concept.

It is naïve and straightforward to compute because we assume that the input variables are independent of each other (in real life, that is not always the case, but this is an excellent first approximation). We can check assumption that by using a chi-squared test, but we rarely take that step when using this algorithm.

Again, as in previous situations, learning about Naïve Bayes is best done with an example. We want to go for a hike and want to make a decision based on weather conditions—a perfect case for this type of machine.

EXERCISE 8.3 – THE HIKING DATA SET

Suppose we collected data on the characteristics of the weather (outlook, temperature, humidity, and wind) and whether you took a hike or not under those conditions from previous hikes.

Can we create a decision model to help us decide whether we will hike in the future, given some weather conditions?

Here is some data (Figure 8.17) you collected on previous occasions when you had to make such a decision to go on a hike. Let’s use it to build a Naïve Bayes decision classifier to go hiking.

The first thing we do is compute the conditional frequencies of hiking and the outcome variable (or not) for all the weather conditions for each input variable (use the *HIKING.xlsx* file, found in the *Chapter 8* folder in the *Case Data* depository). Figure 8.18 shows the result of contingency tables for all variables (*Outlook*, *Temperature*, *Humidity*, *Windy*) in the data set against the outcome variable *hike* using JASP.

Outlook	Temperature	Humidity	Windy	Hike
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

FIGURE 8.17 The HIKING data set.

Contingency Tables				Contingency Tables ▼					
		Hike				Hike			
Outlook		no	yes	Temperature		no	yes		
		Total				Total			
overcast	Count	0.000	4.000	4.000	cool	Count	1.000	3.000	4.000
	% within column	0.000%	44.444%	28.571%		% within column	20.000%	33.333%	28.571%
rainy	Count	2.000	3.000	5.000	hot	Count	2.000	2.000	4.000
	% within column	40.000%	33.333%	35.714%		% within column	40.000%	22.222%	28.571%
sunny	Count	3.000	2.000	5.000	mild	Count	2.000	4.000	6.000
	% within column	60.000%	22.222%	35.714%		% within column	40.000%	44.444%	42.857%
Total	Count	5.000	9.000	14.000	Total	Count	5.000	9.000	14.000
	% within column	100.000%	100.000%	100.000%		% within column	100.000%	100.000%	100.000%

Contingency Tables				Contingency Tables					
		Hike				Hike			
Humidity		no	yes	Windy		no	yes		
		Total				Total			
high	Count	4.000	3.000	7.000	FALSE	Count	2.000	6.000	8.000
	% within column	80.000%	33.333%	50.000%		% within column	40.000%	66.667%	57.143%
normal	Count	1.000	6.000	7.000	TRUE	Count	3.000	3.000	6.000
	% within column	20.000%	66.667%	50.000%		% within column	60.000%	33.333%	42.857%
Total	Count	5.000	9.000	14.000	Total	Count	5.000	9.000	14.000
	% within column	100.000%	100.000%	100.000%		% within column	100.000%	100.000%	100.000%

FIGURE 8.18 The HIKING data set contingency tables for all variables using JASP.

COMPUTING THE CONDITIONAL PROBABILITIES

Take the *Temperature* variable and the condition “hot.” When crossed with the variable *Hike* (yes), we get a number, a probability estimated from the contingency table of how often we hiked when it was hot. In mathematical terms, we just computed the probability of it being hot *given* that we hiked. The mathematical notation is $P(\text{hot}|\text{yes})$, and its value is a 22% chance. This is known as *conditional probability* and is essential to understand it. Compute this for all the possible conditions: $P(\text{hot}|\text{yes})$, $P(\text{hot}|\text{no})$, $P(\text{sunny}|\text{yes})$, and $P(\text{sunny}|\text{no})$, etc.. They are obtained from the contingency tables. Once you have these ratios, then you can predict whether you will hike or not for any combination of weather characteristics.

Let’s compute the probability of taking a hike given a set of weather conditions. Imagine that we have a day with the following characteristics: *Outlook* = sunny, *Temperature* = mild, *Humidity* = normal, and *Windy* = FALSE.

Are we likely to go for a hike or not?

First, we'll calculate the probability that you will hike given X (for all the conditions), $P(\text{yes}|X)$. Then compute the probability that you will not hike given X , $P(\text{no}|X)$, and compare. The contingent probabilities are tabulated in Figure 8.19. In this case, the total probability of hiking is the product of all the probabilities.

$$P(\text{yes}|\text{sunny,mild,normal,FALSE}) = P(\text{sunny}|\text{yes}) * P(\text{mild}|\text{yes}) * P(\text{normal}|\text{yes}) * P(\text{FALSE}|\text{yes})$$

Parameter	Condition	$P(\text{no} X)$	No	$P(\text{yes} X)$	Yes
Outlook	sunny	$P1(\text{nohike} \text{sunny}) =$	60.00%	$P1(\text{hike} \text{sunny}) =$	22.22%
Temperature	mild	$P2(\text{nohike} \text{mild}) =$	40.00%	$P2(\text{hike} \text{mild}) =$	44.44%
Humidity	normal	$P3(\text{nohike} \text{normal}) =$	20.00%	$P3(\text{hike} \text{normal}) =$	66.67%
Windy	FALSE	$P4(\text{nohike} \text{FALSE}) =$	40.00%	$P4(\text{hike} \text{FALSE}) =$	66.67%
Overall	Hike	$P5(\text{nohike}) =$	36%	$P5(\text{hike}) =$	64%
	Likelihood	$P1 * P2 * P3 * P4 * P5$	0.69%	$P1 * P2 * P3 * P4 * P5$	2.8%

FIGURE 8.19 Computing the probability we would go for a hike given certain conditions.

Because $P(\text{yes}|X)$, which is 2.8%, is more significant than $P(\text{no}|X)$, which is 0.69%, then we can predict you would go hiking given that the outlook is sunny, the temperature is mild, the humidity is average, and it is not windy.

This is the essence of Naïve Bayes!

Unfortunately, some of our tools, such as JASP and Jamovi, are not yet able to build a Naïve Bayes model, but Orange3 does have a widget that builds Naïve Bayes models. Figure 8.14 shows the hiking Naïve Bayes model for the *HIKING.CSV* database based on using all the given data for training. Use the *HIKING.xlsx* file, found in the *Chapter 8* folder in the *Case Data* depository. Figure 8.20 shows the processing of a proposed hike when it is sunny, with mild temperatures, normal humidity, and no wind. The Python model predicts that we would most likely take a hike! Naïve Bayes is a conditional probability model.

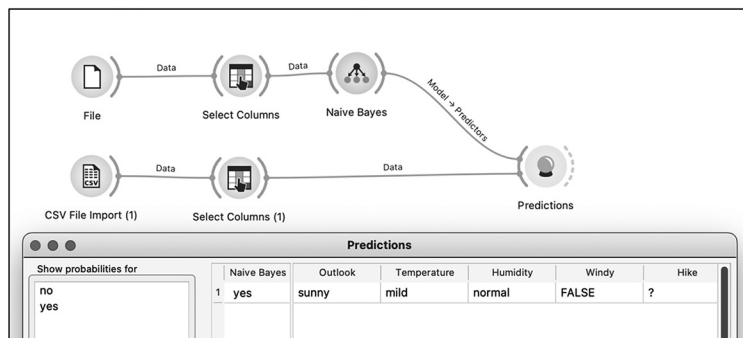


FIGURE 8.20 The Naïve Bayes model of the HIKING data set and prediction for the proposed hike.

CASE STUDY 8.1: CLASSIFICATION WITH THE SFO SURVEY DATA SET

As the SFO marketing director, we may want to know what type of customer may be satisfied with your airport services and what type is not. You want a report on the characteristics of airport customers and their declared like or dislikes of your airport. This is where decision tree analysis could yield

some good results. We want to explore the profiles of passengers giving us low scores. For example, we will use the overall satisfaction score *Q7ALL* as our target or outcome variable. Rather than using the full range of scores from 1 to 5, we will convert it to a binary (0,1) score where 0 = not happy with the airport, using scores of 1-3; and 1 = happy with the airport, using scores 4 and 5. That will be our outcome or target variable. For input variables or features, we will use some customer demographic elements: *Q2PURP1* (trip purpose), *Q17LIVE* (local), *Q20Age* (age), *Q22Gender* (gender), *Q24Income* (income), and *Q23FLY* (frequent flier). You can also bin and perform the analysis for another satisfaction indicator, the net promoter score, the variable *NETPRO*.

Use the *SFO 2018 Survey Data.xlsx.csv* data file found in the *SFO Survey Data* folder of the *Case Data* depository. Ensure that both variables, *Q7ALL* and *NETPRO*, have been adequately prepared by binning them (0=1-3, 1=4,5 for *Q7ALL* and 0=1-7, 1= 8,9,10 for *NETPRO*). Also, be sure to remove 0's and 6's from all the input variable (replace with blanks) so it does not produce an error (remember, a 0, in that case, is not a score or category, but it signifies the respondent left it blank and it should not be counted).

The question we will answer using linear regression for this case is as follows:

What are the most important factors that influence a positive satisfaction score?

SOLUTION IN R

Ensure that both variables, *Q7ALL* and *NETPRO*, have been adequately prepared by binning them (0=1-3, 1=4,5 for *Q7ALL* and 0=1-7, 1= 8,9,10 for *NETPRO*). You can follow the steps in Chapter 4 and prepare the data set.

Once your data set is ready to analyze, import it into JASP instead of Jamovi since Jamovi does not support Random Forest classification analysis.

After importing the data set into JASP, select *Random Forest Classification* under *Machine Learning* at the top. Select *Q7ALL* as the target value and all other variables or features into predictors. The steps and results are shown in Figure 8.21(a) and (b).

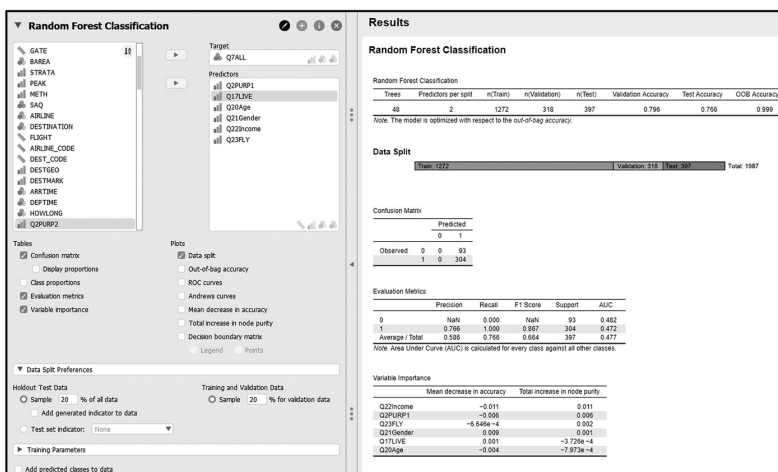


FIGURE 8.21(a) Steps and result of the random forest classification.

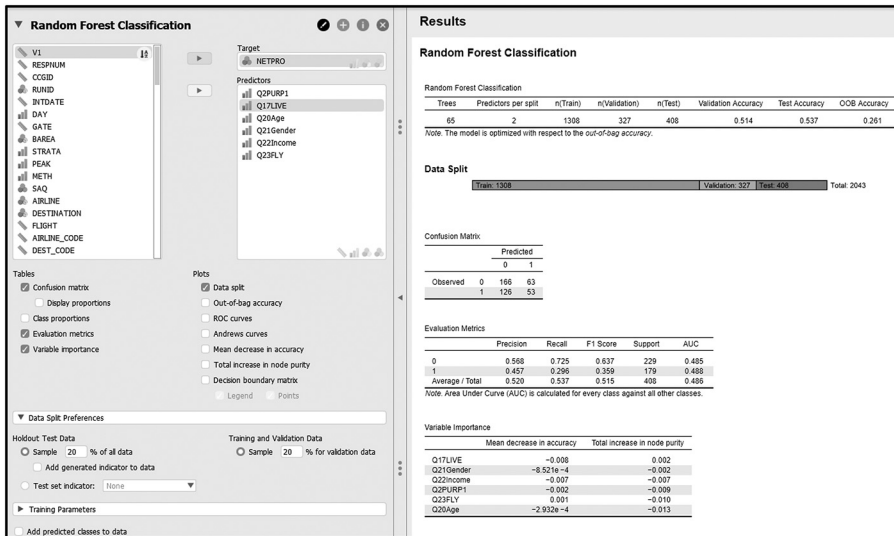


FIGURE 8.21(b) Steps and results of the random forest classification.

SOLUTION IN PYTHON

Ensure that both variables, *Q7ALL* and *NETPRO*, have been adequately prepared by binning them (0=1–3, 1=4–5 for *Q7ALL* and 0=1–7, 1= 8,9,10 for *NETPRO*). You can follow the steps in Chapter 4 and prepare the data set.

Once your data set is ready to analyze, import it into Orange3.

In this case, you need to install all necessary add-ons in Orange to analyze data using different models. Select *Options* and then go to add-ons. On the installer page, you can install all add-ons, just in case you will need one of them in the future. Please ensure you successfully install add-ons; otherwise, you cannot use those models in Orange.

Follow the steps in Figure 8.22 and perform a random forest analysis.

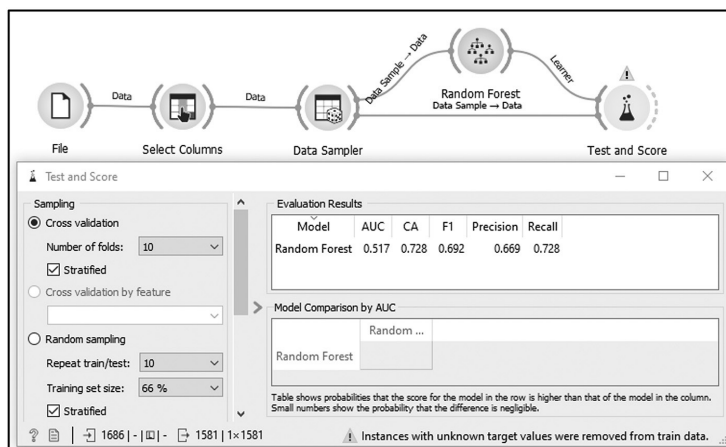


FIGURE 8.22 Steps and result of the random forest analysis.

CASE STUDY 8.2: CLASSIFICATION WITH THE SBA LOANS DATA SET

Again, imagine yourself as the head of the Small Business Administration. You will be required to answer the US Congress on how well your loan program performs. You want to ensure that loans are made to customers who will tend to repay and not default. You ask your analyst to give you criteria on what type of applicant is likely to repay the loan and which applicant is likely to default. We want to know what factors influence repayment success. Use the following framed analytical question:

What factors are strong indicators of loan repayment success?

The output variable, or target, is *LoanStatus*, which has been binned as a binary variable (0 = CHGOFF, default, and 1 = PIF, paid off loan). We use *GrossAmount*, *JobsSupported*, *TermInMonths*, Business Type, NAICS code, and CDC_State for input variables or features.

SOLUTION IN R

Use the *FOIA Loans Data.xlsx* data file found in the *SBA Loans Data* folder of the *Case Data* depository. Make sure the variables under *LoanStatus* have been adequately prepared by binning it (0 = CHGOFF, default, and 1 = PIF, paid off loan) and entering them into a new variable. You can follow the steps in Chapter 4 and prepare the data set. Once your data set is ready to analyze by converting it into a CSV file, import it into JASP. After importing the data set into JASP, select *Random Forest Classification* under *Machine Learning* at the top. Select *LoanStatus* as the target value and all other variables or features into predictors. The steps and results are shown in Figure 8.23.

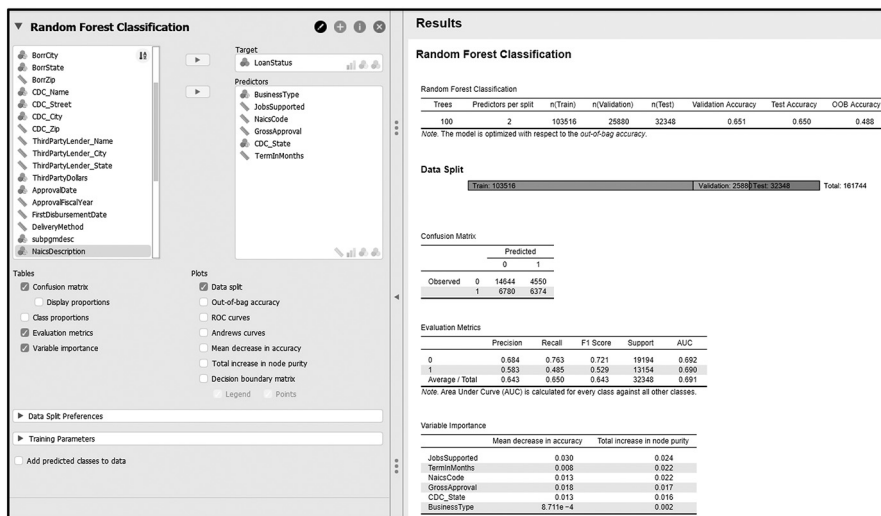


FIGURE 8.23 Steps and result of the random forest classification in JASP.

SOLUTION IN PYTHON

Make sure the variables under *LoanStatus* have been adequately prepared by binning it (0 = CHGOFF, default, and 1 = PIF, paid off loan). You can follow the steps in Chapter 4 and prepare the data set.

Once your data set is ready to analyze, import it into Orange.

Follow the steps in Figure 8.24 and perform a random forest analysis.

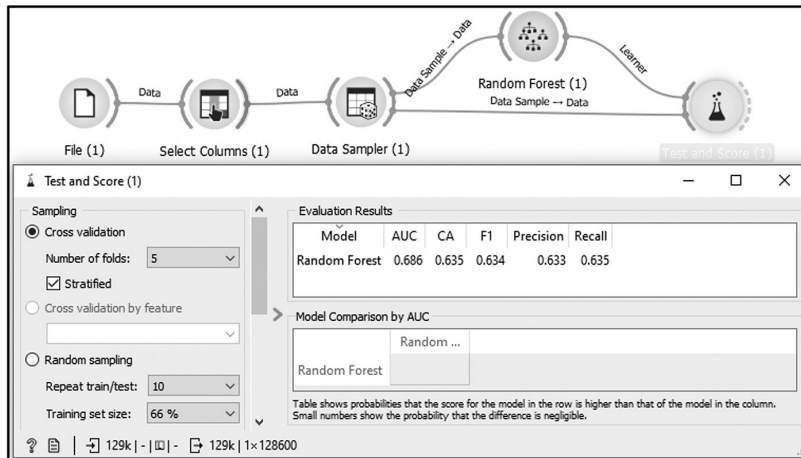


FIGURE 8.24 Steps and result of the random forest classification in Orange3.

CASE STUDY 8.3: CLASSIFICATION WITH THE FLORENCE NIGHTINGALE DATA SET

Domino Data Labs provides data science products for machine learning and data mining to over 20% of Fortune 100 companies. They are promoting their services by educating the public on data science. They built a classification game for children and are using it to educate customers on data science techniques. Domino Data Labs has hired you as a data analyst to help them with running contest. They send parents and teachers a children's book based on a story about a data scientist librarian named Florence Nightingale and the children who borrow books from her (see a copy of the book found in the data repository under this case study data set). Domino makes the fable book available to children, parents, and teachers, hoping to inspire children to become data scientists. In addition to publishing the book, they are running a contest among children (involving parents and teachers) to see if they can detect a pattern in the books Florence the Librarian (and data scientist) gives to the children in the story. They need to solve a classification problem by creating a model from the books Florence has given out based on the previous children's choices and predicting the missing decisions. They need these answers to properly score the children's responses to the contest to award prizes. They want you to model the data using machine learning and document the approach.

Using the *Florence Data Set.xlsx* data set (found in the *Florence Nightingale Data Set* folder in the Case Data depository), build a Naïve Bayes classification model to predict the classification

of those books that her friends did not classify. The output variable, or label, is *Animal*. The input variables or factors are (a) *Dog, Cat* or *Cow*, (b) *Real* or *Make Believe*, and (c) *Strong* or *Fly*. Use the training and unknown data sets in the Case Data depository.

What animal would you predict for the unknown rows (where Animal is blank)?

SOLUTION IN PYTHON

Prepare the data set by splitting the original set into two sets, namely, training and prediction. Import both data sets into Orange, and the two data sets should look like Figure 8.25.

The screenshot shows the Orange3 interface with two data tables. The top table, 'Data Table (1)', is connected to a 'Prediction' widget. The bottom table, 'Data Table', is connected to a 'Data' widget. Both tables have the same four columns: 'Dog, Cat, Cow', 'Real/make-believe', 'stong/fly', and 'animal'.

Data Table (1)

	Dog, Cat, Cow	Real/make-believe	stong/fly	animal
1	cat	real	strong	?
2	dog	real	strong	?
3	dog	real	fly	?
4	cow	make	strong	?
5	cow	real	fly	?

Data Table

	Dog, Cat, Cow	Real/make-believe	stong/fly	animal
1	dog	make	strong	Dragon
2	cow	make	fly	eagle
3	cow	real	strong	eagle
4	cat	make	strong	unicorn
5	cat	make	fly	unicorn
6	cat	real	fly	unicorn
7	dog	make	fly	unicorn

FIGURE 8.25 Import two data sets into Orange3.

Now, let's add a new Naïve Bayes widget and connect the training set. Then connect a prediction model with *Naïve Bayes* and *Prediction* widgets to predict the result shown in Figure 8.26.

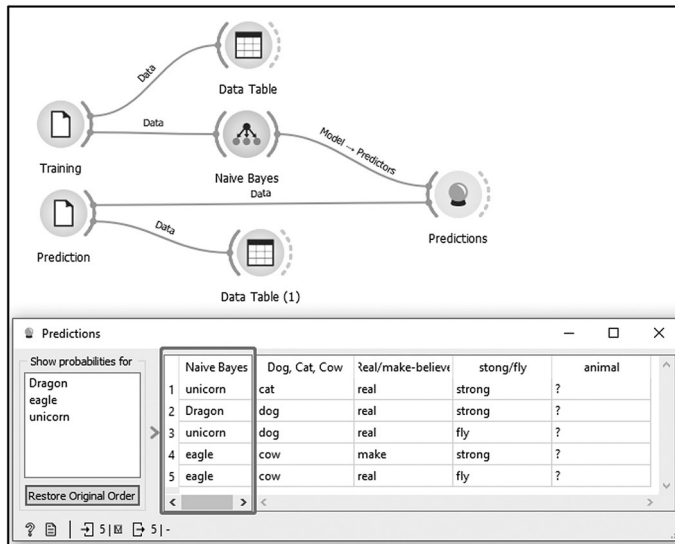
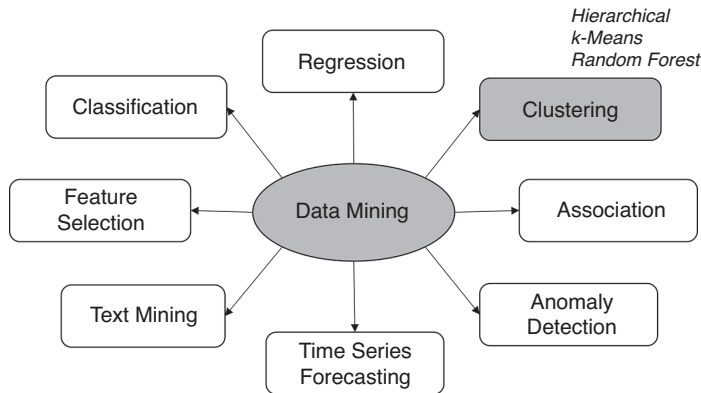


FIGURE 8.26 Steps and result of the Naïve Bayes prediction in Orange3.

REFERENCE

Dua, D, and C. Graff. 2019. "UCI Machine Learning Repository." The University of California, School of Information and Computer Science. <http://archive.ics.uci.edu/ml>.

CLUSTERING



Clustering is a statistical technique used to identify significant clusters among a population based on some factors. It is a way to uncover the natural groupings of the rows in a data set. The object of cluster analysis is to divide the data set into groups, where the observations within each group are relatively homogeneous, yet the groups are unlike each other. Clustering is considered an unsupervised machine learning technique in that no identified outcome variable needs to be predicted. Instead, the object is to identify natural groupings of rows of data.

WHAT IS UNSUPERVISED MACHINE LEARNING?

Data mining and machine learning problems can be broadly categorized into supervised or unsupervised learning approaches. Unsupervised or undirected machine learning uncovers hidden patterns in unlabeled data. It results in associating rows of data with shared characteristics into groups. In unsupervised machine learning, there are no output variable (or labeled variable) to predict; we are interested in grouping the population into clusters of rows. In the previous chapters, we saw regression and decision tree techniques where the object was to predict an

identified outcome variable given a set of factors or input variables. We classified those as supervised learning approaches. Here we tackle popular unsupervised learning techniques.

WHAT IS CLUSTERING ANALYSIS?

Clustering analysis is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). The groups of objects we are working with in our data sets are rows of data that contain the characteristics of members of our population being studied. The chart in Figure 9.1 is an example of finding clusters of a credit card company's customer population based on income and debt variables.

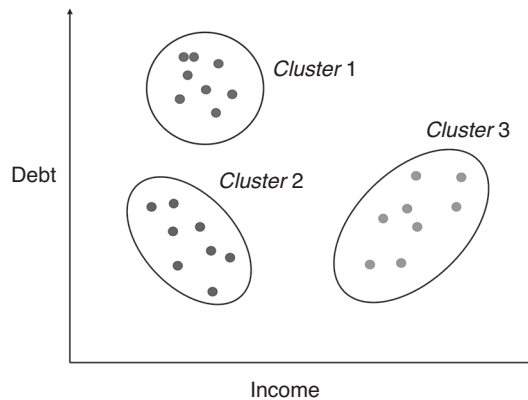


FIGURE 9.1 An example of clustering using two factors, *income* and *debt*.

Clustering is a technique used to profile the portfolio or customer database initially. After understanding the portfolio or customer base, an objective modeling technique is used to build a specific analysis strategy. In this case, the debt/income coordinates for each customer in the database make them susceptible to being grouped, as shown in Figure 9.1. Of course, it is not always as clean or evident as in this figure; there is often a great deal of overlap in the clusters. However, using several algorithms we discuss later in the chapter, we can use it to identify prominent groups.

APPLYING CLUSTERING TO OLD FAITHFUL ERUPTIONS

Another example of clustering is the separation of the eruptions of the Old Faithful geyser at Yellowstone National Park in the United States (<https://www.stat.cmu.edu/~larry/all-of-statistics/=data/faithful.dat>) into long and short waiting times between eruptions. Figure 9.2 shows the results of a clustering analysis. This data set shows the bimodal nature of the physical eruption characteristics of this natural feature. The cluster, in this case, has been selected by inspection. No clustering model was used or developed. We are just identifying when Old Faithful seems to erupt in a bimodal fashion. You try it. Download the data from the URL given above and, using Excel, create a scatter plot of eruption duration versus waiting time for the next eruption for the 272 observations to see the clustering we demonstrate in Figure 9.2.

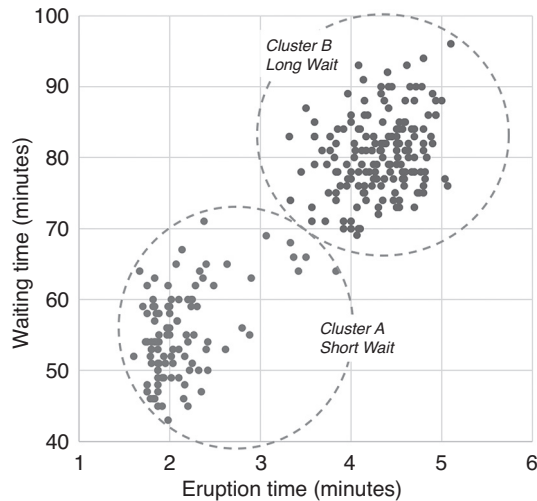


FIGURE 9.2 Grouping of Old Faithful eruptions into two clusters, short and long, by inspection of the scatter plot of 272 observations.

In subsequent chapter sections, we will create clustering-based models using hierarchical and k-means clustering.

EXAMPLES OF APPLICATIONS OF CLUSTERING ANALYSIS

There are many applications of clustering analysis, many having to do with the segmentation of customer databases. For example, television marketing companies use cluster analysis to group television shows based on viewer/group characteristics. This helps them attract similar audiences within each group. Credit card companies segment customers into groups based on the number of purchases made, whether balances are paid off every month, and where the purchases are made.

Grocery stores with loyalty-card programs cluster their customers based on the number, frequency, and types of purchases. After customers are segmented, advertising can be appropriately targeted. We can cluster college applicants by their characteristics to make better educational admissions decisions. We can better match the characteristics of applicants to the school offerings and provide better assurance of scholastic success to admitted students. We can cluster enrolled students as they continue their studies to identify students in academic trouble and provide appropriate assistance to each needy group to help them succeed.

SIMPLE CLUSTERING EXAMPLE USING REGRESSION

In Chapter 7, we saw the development and application of a powerful computing machine, linear regression. In many cases, we can use it as a clustering tool. Take, for example, a univariate linear regression model. Using a two-dimensional scatterplot of the data with the predicted linear regression will illustrate how it can be used as a clustering tool. Only some population members will lie on the linear regression line; many will deviate from it. We can use this deviation from the predicted as an index of cluster membership: above and below the regression line.

The linear regression becomes a boundary line between two zones that divides solution space into two regions, above and below the trend line.

Let's consider the model of the *Franchises.csv* data set, where the sales of each franchise store (*SALES*) had associated with it several characteristics: the square footage of the store (*SQFT*), the inventory the store carried (*INVENTORY*), the amount of monthly advertising spent by each store (*ADVERTISING*), the size of the neighboring market in measures by the number of families (*FAMILIES*) within a certain distance to the store, and the number of competitor's stores in the area (*STORES*). We can create a linear regression model of sales as predicted by the size of the market scatter plot with linear regression as a clustering example. The linear regression is the predicted average sales expected for a given market size. We then can imagine those stores for which actual sales are higher than the predicted (average) sales for that size market could be considered a high performer. Those stores with actual sales below the predicted average sales are poor performers. That gives us two clusters!

We can go further and institute a performance band; say, those within \$10,000 above and below average (predicted) performance are considered average performers. Then, those with actual sales above \$10,000 predicted will be considered good performers, and conversely, those with sales below \$10,000 of the predicted amount would be considered poor performers. That gives us three clusters. That is the case that is exemplified in Figure 9.3. The linear regression algorithm has an associated table beside the prediction of the outcome variable, which is its variation from the actual, which we call a residual. So, the residual plot or table can be used as a clustering mechanism, as described.

Suppose we were to use all factors in the data set (SqFt, Inventory, Advertising, Families, and Stores) in a linear regression to predict sales performance. In that case, we could use the overall residuals to create clusters. We would use the residuals, again, to find the top performers (more than 10K above the expected sales), average performers (between 10K and -10K around

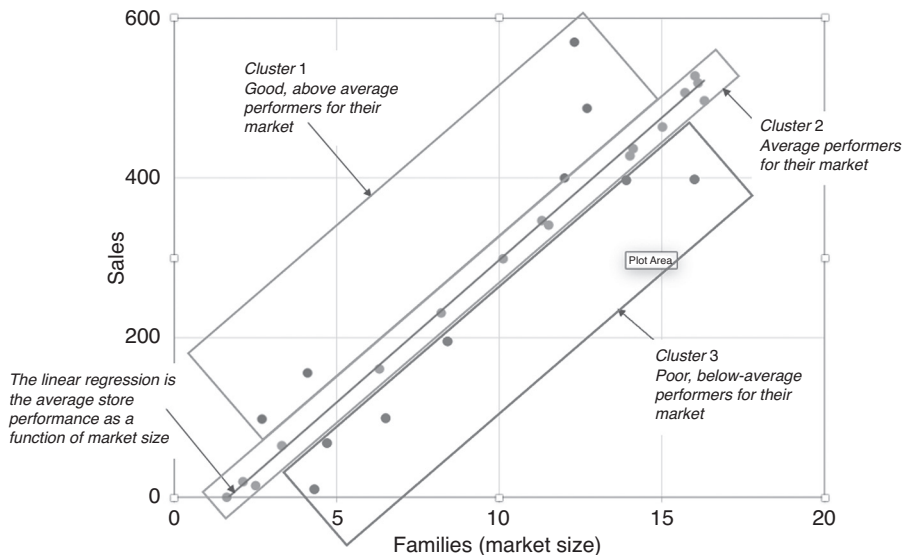


FIGURE 9.3 Using linear regression as a clustering tool using market size (*Families* variable) to differentiate sales performance between stores and to group them as top performers, average performers, and poor performers clusters.

the expedited sales); and poor performers (those with less than 10K in sales below the expected sales) as the three clusters. Figure 9.4 shows the residuals for such a multivariate linear regression applied to the *Franchises.xls* data set.

RESIDUAL OUTPUT			
<i>Observation</i>	<i>Predicted SALES</i>	<i>Residuals</i>	<i>Performance</i>
6	446	41	Above Average
17	370	28	Above Average
16	70	28	Above Average
2	129	27	Above Average
22	89	10	Above Average
14	8	7	Average
12	424	4	Average
18	157	4	Average
19	393	4	Average
1	229	2	Average
11	569	1	Average
7	298	1	Average
5	439	-2	Average
9	24	-4	Average
13	469	-5	Average
24	352	-5	Average
15	70	-5	Average
25	347	-6	Average
4	527	-8	Average
20	506	-9	Average
21	538	-10	Below Average
26	518	-11	Below Average
27	412	-12	Below Average
23	17	-17	Below Average
10	86	-18	Below Average
3	29	-19	Below Average
8	221	-26	Below Average

FIGURE 9.4 Using multivariate linear regression as a clustering tool using all five data set variables as factors (*SqFt*, *Inventory*, *Advertising*, *Families*, and *Stores*) to differentiate sales performance between stores and to group them as top performers, average performers, and poor performers clusters. The analyst has arbitrarily set the boundary between the tree clusters.

HIERARCHICAL CLUSTERING

We now consider two machine learning clustering algorithms: hierarchical and k-means. Let's see how the hierarchical algorithm works. Say we have six observations or members of our population, and we are tracking two factors to create our clusters. Figure 9.5 shows these six observations (labeled a, b, c, d, e, and f) plotted in two dimensions. At the start of the algorithm, each observation is considered its own cluster. The distance between each cluster and all other clusters is computed, and the nearest clusters are merged. If there are n observations, this process

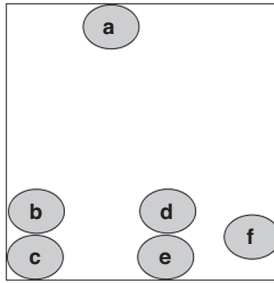


FIGURE 9.5 Six members of a population of customers with two of their characteristics plotted in a scatter plot showing the distribution of the data points in the x-y factor space.

is repeated ($n-1$) until there is only one large cluster. Visually, this process is represented by a tree-like figure called a dendrogram, as we will see in some practical examples later. Inspecting the dendrogram allows the user to make a judicious choice about the number of clusters to use.

This technique operates on the principle that a data point closer to the base point will behave more similarly to a data point farther away from the base point. For example, a, b, c, d, e, and f are a group of six customers, and we wish to group them into clusters. They have two attributes (x and y), and we plot their attribute coordinates with a scatter plot.

Hierarchical clustering sequentially groups these customers, and we can stop the process at any number of clusters we want. Figure 9.6 shows an illustrative chain of clustering.

The first and starting pass assumes each customer is in its own cluster. We successively create clusters based on the distance between a cluster center and each cluster member, creating clusters as we go along until the last pass has everyone in one cluster.

If we want three clusters, [a], [b c], and [d e f] are the required clusters. The hierarchical clustering technique uses basic clustering analysis based on cluster center distances and is very stable. The problem with hierarchical clustering is that it can only handle a small number of data points and the computations are very time-consuming since we have to compute and compare the distances from every point to every other point in every pass. This is because it tries to calculate the distance between all possible combinations and then takes one decision to combine two groups/individual data points. We use it when we have few members in our population (such

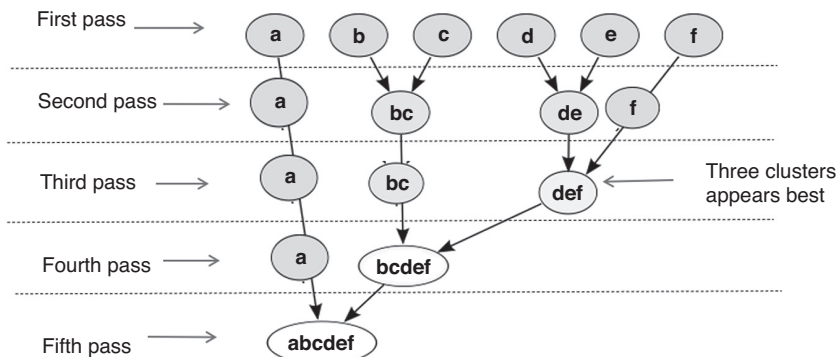


FIGURE 9.6 The six members of a population of customers using their plotted characteristics and observing the interpoint distances to successively group customers by distances from each data point to all other data points.

as 150 or fewer, for example). Or, to get an idea of what the clusters might look like, we could randomly sample a larger population and perform hierarchical clustering on the sampled rows.

APPLYING HIERARCHICAL CLUSTERING TO OLD FAITHFUL ERUPTIONS

Let's see how we can apply this method to the Old Faithful eruption data we analyzed by inspection earlier in the chapter. Figure 9.7 shows a hierarchical cluster analysis done in JASP of the 272 observations and analysis for two clusters. If the variables are normalized (converted into a Z score), we observe two very clean and minimally overlapping clusters, as shown by the

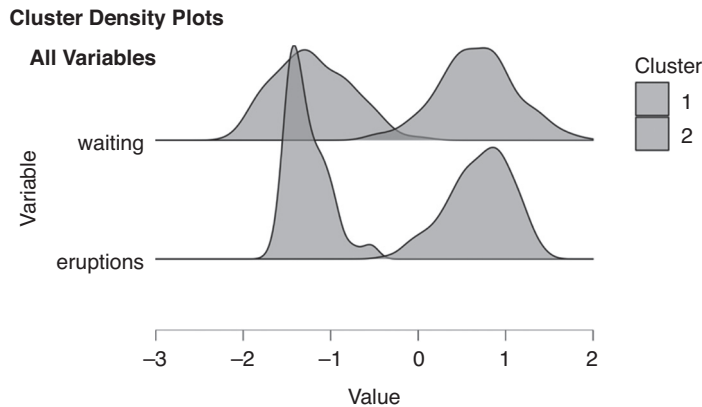


FIGURE 9.7 Hierarchical clustering model of the Old Faithful eruption data analyzed using JASP: the cluster density plots.

Hierarchical Clustering					
Clusters	N	R ²	AIC	BIC	Silhouette
2	272	0.824	8909.770	8924.190	0.720

Note. The variables in the model are **unstandardized**.

Cluster Means		
	eruptions	waiting
Cluster 1	4.298	80.285
Cluster 2	2.094	54.750

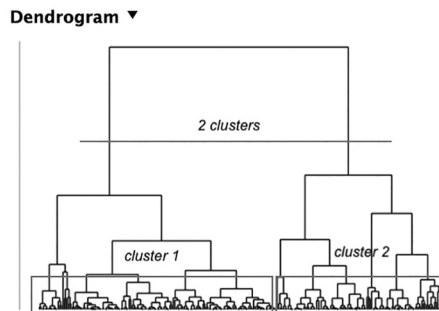


FIGURE 9.8 Hierarchical clustering model of the Old Faithful eruption data analyzed using JASP: the cluster centers and R², a measure of model quality and the dendrogram.

distribution of waiting times and irruption times. Notice that the peaks of the two clusters are widely separated, with just a small amount of overlap between the distributions.

Figure 9.8 shows the dendrogram displaying the hierarchical clustering model by giving the cluster centers for both clusters. Notice that the model explains over 82% of the variability in the data ($R^2 = .824$), which is quite good.

EXERCISE 9.1 – HIERARCHICAL CLUSTERING AND THE IRIS DATA SET

Let's apply this technique to the same data set we used when studying machine learning in Chapter 8: the Iris data set. Here, we will use the petal length and width and sepal length and width to group the 150 irises into three groups or clusters. Hopefully, this will correspond to our three types of irises. Notice that we are not trying to predict the Iris genus, but will see if the irises in our set clustered into what will turn out to be the proper classification by genus.

We use JASP to create a hierarchical clustering model of 150 irises. The first plot in Figure 9.9 is a classic dendrogram that shows, at the very top, all the irises in one cluster, and at the bottom, every iris in one of 150 clusters. As you move up from the bottom, the irises begin clustering, and we get to a certain point where we see that we could have three clusters very close to the top, with the associated irises flowing from each of the three clusters to their membership in the corresponding clusters. That is our model. Each cluster has a cluster center denoted by the average sepal length and width and petal length and width.

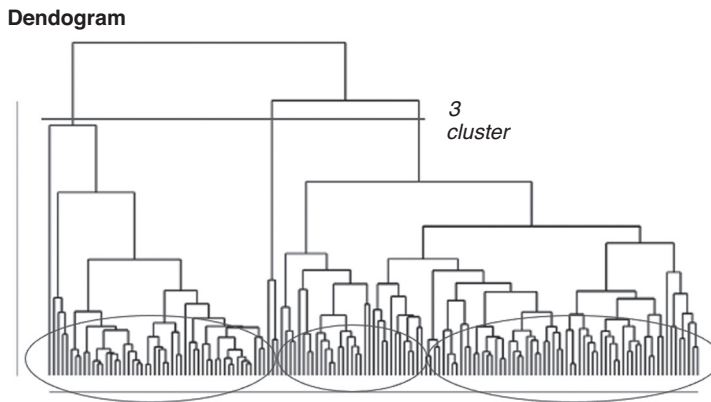


FIGURE 9.9 Dendrogram of hierarchical clustering of the 150-iris data set showing three possible clusters corresponding to the three iris species identified by a trained botanist.

We see the distribution of the characteristic factors in Figure 9.10. We see that the petal characteristics have peaks that are widely separated with little overlap for each cluster. However, the sepal characteristics have quite a bit of overlap, and the separations are not as pronounced; therefore, we can conclude that petal characteristics are a better clustering indicator than the sepal characteristics, where the latter might produce some confusion in clustering. We may recompute this model with only the petal characteristics as features, leaving out the sepal characteristics to produce a much more accurate machine.

We see from the table in Figure 9.11 that the model using all four characteristics explains over 88% of the variability of the data ($R^2 = .883$), which is quite good. We also obtain from this

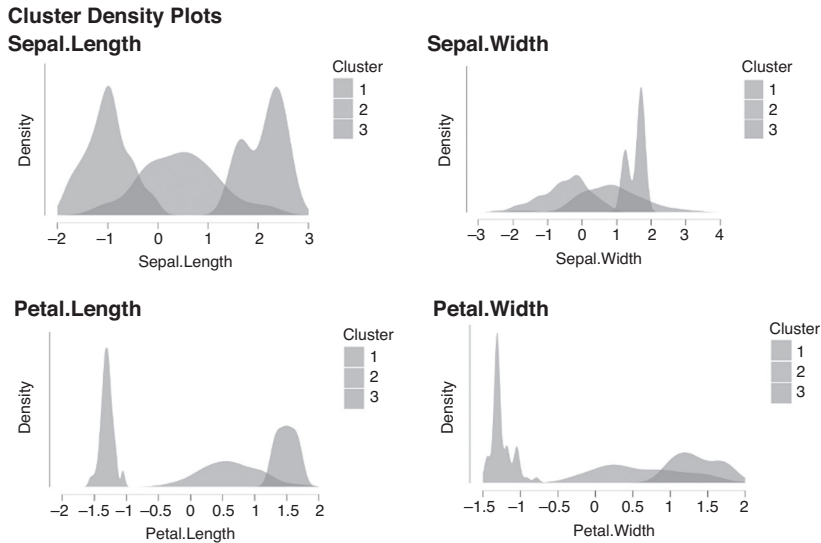


FIGURE 9.10 The cluster density plots of the four variables used to create hierarchical clusters. Notice the overlaps in the sepal length and width, which are not present in the petal length and width distributions.

Hierarchical Clustering ▼

Hierarchical Clustering

Clusters	N	R ²	AIC	BIC	Silhouette
3	150	0.883	103.450	139.570	0.550

Note. The variables in the model are **unstandardized**.

Cluster Information

	Cluster	1	2	3
Size		50	64	36
Explained proportion within-cluster heterogeneity		0.191	0.532	0.277
Within sum of squares		15.151	42.264	22.030

Cluster Means ▼

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Cluster 1	5.006	3.428	1.462	0.246
Cluster 2	5.930	2.758	4.411	1.439
Cluster 3	6.853	3.075	5.786	2.097

*Cluster centroids
(averages for each
variable)*

FIGURE 9.11 The solution for three clusters shows the distribution of the 150 specimens assigned to the three clusters and the table of cluster centroids.

JASP model the cluster centers, which become the model to predict cluster membership for unknown irises.

To use the model, we measure the characteristics of an unclassified iris and use the measurements to compare to the cluster centroids. Figure 9.12 shows how this model may be applied to identify which cluster an unknown iris, *Iris 1*, belongs in. We measure the Cartesian distance between the unknown iris and each cluster center to identify which of the three distances is shorter and derive cluster membership thereby.

Cluster Means						
	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width		
Cluster 1	5.006	3.428	1.462	0.246		
Cluster 2	5.93	2.758	4.411	1.439		
Cluster 3	6.853	3.075	5.786	2.097		
Iris 1	3	7	1	4		
Iris 1						
Dist 1	2.006	3.572	0.462	3.754	5.5757672	Cluster 1
Dist 2	2.93	4.242	3.411	2.561	6.6912709	Cluster 2
Dist 3	3.853	3.925	4.786	1.903	7.5351469	Cluster 3
Applying the model						
Cluster Distance = SQRT(ABS(Iris Sepal L-Cluster Sepal L)^2+ ABS(Iris Sepal W – Cluster Sepal L)^2+ ABS(Iris Petal L – Cluster Petal L)^2+ABS(Iris Petal W – Cluster Petal W)^2)						

FIGURE 9.12 Applying the hierarchical cluster model represented by the centroids of the clusters to identify an unknown iris, *Iris 1*.

K-MEANS CLUSTERING

A more robust clustering technique is k-means clustering. This technique is more frequently used in the analytics industry as it can handle many data points. Its primary advantage is that it can handle massive data sets. Where hierarchical clustering may handle hundreds of data rows, k-means is perfectly able to handle millions, together with tens of input factors, not just one or two. One major disadvantage is that in distinction with hierarchical, which can handle categorical and numerical variables for clustering, k-means can only handle numerical variables as input factors. This is seldom a problem, as we have many clustering situations with multiple numerical features. Another way to deal with categorical variables and still include them in a k-means clustering model is to convert them to binary variables.

HOW DOES THE K-MEANS ALGORITHM COMPUTE CLUSTER CENTROIDS?

How does the k-means clustering algorithm work? Let's go through a simple example, as shown in Figure 9.13. The algorithm aims to discover the cluster centers and assign cluster memberships to our database's data rows. Before we start, we need to ascertain how many clusters we wish to create. This is not much of a problem if we have a database of customers and we wish to create a certain number of microsegments of our customer database. Sometimes we wish to find out if there is a minimum set of natural clusters. The k-means algorithm cannot quickly help us in that situation because you need to declare how many clusters you will separate the database in at the start. One approach to finding a natural set of clusters is randomly sampling

the database and creating a hierarchical cluster model of the sampled rows. A scree plot will be associated with the hierarchical model, indicating the minimum set of clusters that will yield an optimum separation between the database rows.

Let's assume, for our example, that we determined that two clusters are what we wish to discover. The algorithm starts by randomly picking the coordinates of two cluster centers, called *seeds*. The seeds in the following figure are the two dots; all the data points (we use only two factors in our database) are plotted as diamonds. The next step is to compute the Cartesian distance between every data point and each of the two cluster centers (the square root of the sum of the squares of the differences between the centroid and the data point for each feature). We now have every data point distance to each of the cluster centers. The critical step is comparing these two distances; we use the measure of whichever of the two distances is shorter to assign that data point to the cluster to which it is closest. That will separate all the data points into two clusters very quickly (the dark versus light colored diamonds). This is a very efficient algorithm for that reason, and the algorithm can go through millions of data points in short order.

Once we identify the membership cluster for every data point, we re-compute each cluster center of the two newly formed clusters. The centroids of the clusters will move from the original location, which was randomly selected to the new location, which is now newly computed. We see of movement of the centroids (the dots) as we compute our way from Step 1 through Step 2 to Step 3 in Figure 9.13. We then re-compute all the distances between each data point and the new centroids and compare the two distances for every data point to see which is shorter. Each data point may stay within the original computed cluster or move to the other, depending on the new distances to the new centroids, as we see in Step 4 in Figure 9.13.

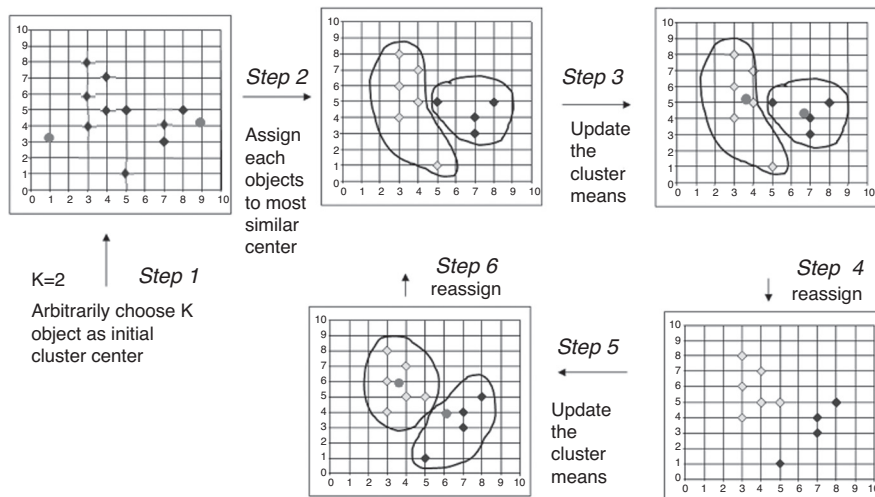


FIGURE 9.13 The recursive method of locating cluster centroids, starting with an initial arbitrary set.

We then re-compute cluster centers and see them moving to new locations. We repeat Steps 3 and 4 until the cluster center locations hardly move. We say that the algorithm re-iterates till the general penalty term is minimized (cluster centers moving a minimal amount between the iterations), which causes us to stop and conclude that we have a stable solution. At that point, the cluster centers become our model.

APPLYING K-MEANS CLUSTERING TO OLD FAITHFUL ERUPTIONS

Let's see how we can apply k-means clustering to the Old Faithful eruption data we analyzed by inspection earlier in the chapter. Figure 9.14 shows a k-means cluster analysis done in Jamovi of the 272 observations and analysis for two clusters. We used the un-normalized variables in this case. We also observed two very clean and minimally overlapping clusters, as shown by the cluster plot of the 272 data points. The plot also includes a table of the cluster centers. Notice a slight overlap at the edges of the two distributions.

Centroids of clusters Table

	Cluster No	eruptions	waiting
1	1.00	4.298	80.285
2	2.00	2.094	54.750

[4]

Cluster plot

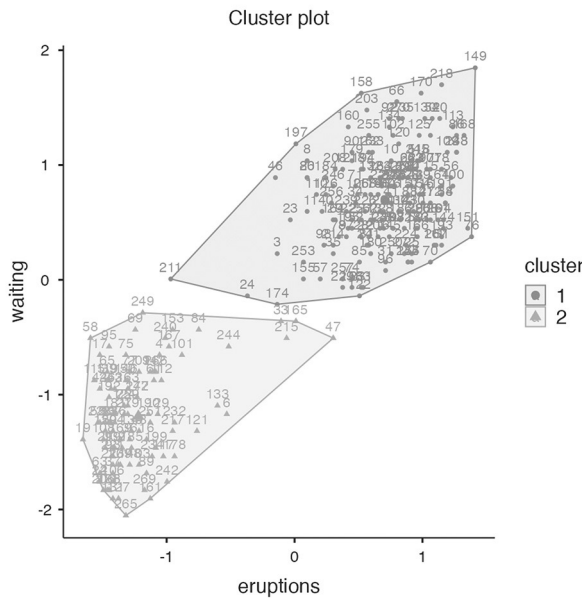


FIGURE 9.14 K-means clustering model of the Old Faithful data set, showing the cluster centroids and the plot of the data clustering into two clusters.

EXERCISE 9.2 – K-MEANS CLUSTERING AND THE IRIS DATA SET

Returning to our data set of 150 irises, let's see if we can apply k-means clustering to create clusters of irises. If we load the *iris.csv* data set into JASP (found in the *Chapter 9* folder of the *Case Data* depository), select three clusters, and do not standardize the factors, we can create a model as shown in Figure 9.15. The model explains over 88% of the variability in the data ($R^2 = .883$), which is quite good and compares well with the hierarchical model. We also obtain the cluster centers.

K-Means Clustering

K-Means Clustering

Clusters	N	R ²	AIC	BIC	Silhouette
3	150	0.884	102.850	138.980	0.550

Note. The variables in the model are **unstandardized**.

Cluster Information

	Cluster	1	2	3
Size		50	62	38
Explained proportion within-cluster heterogeneity		0.192	0.505	0.303
Within sum of squares		15.151	39.821	23.879

Cluster Means

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Cluster 1	5.006	3.428	1.462	0.246
Cluster 2	5.902	2.748	4.394	1.434
Cluster 3	6.850	3.074	5.742	2.071

FIGURE 9.15 The k-means solution for three clusters, showing the distribution of the 150 specimens assigned to the three clusters and the table of cluster centroids.

If we were to repeat the clustering exercise using k-means with three clusters but only the petal measurements, we would see a marked performance improvement. The model under those conditions would explain over 94% of the variability in the data ($R^2 = .943$), which is much better. The cluster centers will be comparable between the two models so that we will see little difference in them, but we can achieve a better performance by only using petal measurements.

HIERARCHICAL VS. K-MEANS CLUSTERING

Let's compare the two techniques. Hierarchical clustering starts with individual data points and sequentially clusters them to find the final cluster. K-means clustering starts from some initial set of clusters and then tries to reassign data points to k clusters to minimize the total error term. Hence, k-means uses far fewer iterations for many data points than hierarchical clustering. The k-means algorithm is sensitive to the units in which the variables are measured. Suppose we have three variables (length, weight, and value). In that case, we will get one set of clusters if the units of measurement are inches, pounds, and dollars, and (probably) a radically different set of clusters if the units of measurement are feet, ounces, and cents. To avoid this problem, ensure the box for *Columns Scaled Individually* is checked.

Clustering is not a purely data-driven exercise. An expert business analyst must perform careful statistical analysis and interpretation to produce good clusters. Second, many iterations may be needed to achieve the goal of producing good clusters, and some of these iterations may require field testing. Clustering is not a purely statistical exercise, and good use of the method requires knowledge of statistics and the characteristics of the business problem and the industry.

CASE STUDY 9.1: CLUSTERING WITH THE SFO SURVEY DATA SET

As the SFO marketing director, we can categorize groups of customers by their demographic characteristics as well as their opinions. We shall first seek clusters of customers by their demographics (such as age, income, and business travel habits). That will give us an idea of our customer base, who they are, and their characteristics. Then we will add the opinion variables to see which customers feel good about the airport and which do not. To explore the customer base by demographics, we will work with clustering techniques that require numeric variables only (k-means and hierarchical). We need to use only numeric variables, or at least create numeric variables for those that have ranges. Remember, clustering is an unsupervised machine learning technique. As much as we want to build models from training data (existing surveys) for database marketing, we are also interested in discovering the significant clusters of our customer base and the corresponding characteristics.

For k-means and hierarchical clustering, we first use *Q5TIMESFLOWN*, *Q5FIRSTTIME*, *Q20Age* (create a variable using the center of the age range as the age, i.e., $18 - 24 = 20.5$), *Q21Gender* (only use 1, 2), *Q22Income* (again, create a variable using the center of the age range as the income, i.e., $\$50,000 - 1000,000 = \$75,000$), and *Q23Fly* (only use 1, 2). As a second pass, create clusters that include the overall satisfaction score *Q7ALL* (make sure to use the full range of scores from 1 to 5 (exclude 6 and 0); and also use the other indicator of satisfaction, the net promoter score, the variable *NETPRO*. Be sure to remove 0s from all the variables (replace with blanks) so they do not produce an error (remember, a 0, in that case, is not a score or category, but it signifies the respondent left it blank, and it should not be counted). Experiment with creating four and seven microsegments of the population. See if any segmentation strategy produces a particularly insightful result.

The question we will answer using clustering for this case is as follows:

What are the characteristics of the most essential San Jose Airport customer segments?

As an additional exercise, answer the following framed question:

Can we identify segments that are particularly dissatisfied with the airport?

SOLUTION IN R

Before importing the data set into JASP, please follow the steps in Chapter 4 and get the data set well-prepared. Import the well-prepared data set into JASP. To meet the requirements of clustering analysis, we need to change the data type of those variables. In JASP, you can click the type of icon of each variable and choose the right type you need. Use the *SFO Survey Data.csv* data set found in the *SFO Survey Data* folder in the *Case Data* depository.

Select *K-Mean Clustering* under *Machine Learning* on the top, then move all variables you want to analyze into the box of variables for computation. First, we create the Elbow plot to indicate the number of clusters we need to generate, as shown in Figure 9.16.

From the Elbow method plot, we find that four clusters should work for this data, and we can create the table for four and seven clusters by following the steps in Figure 9.17(a) and (b).

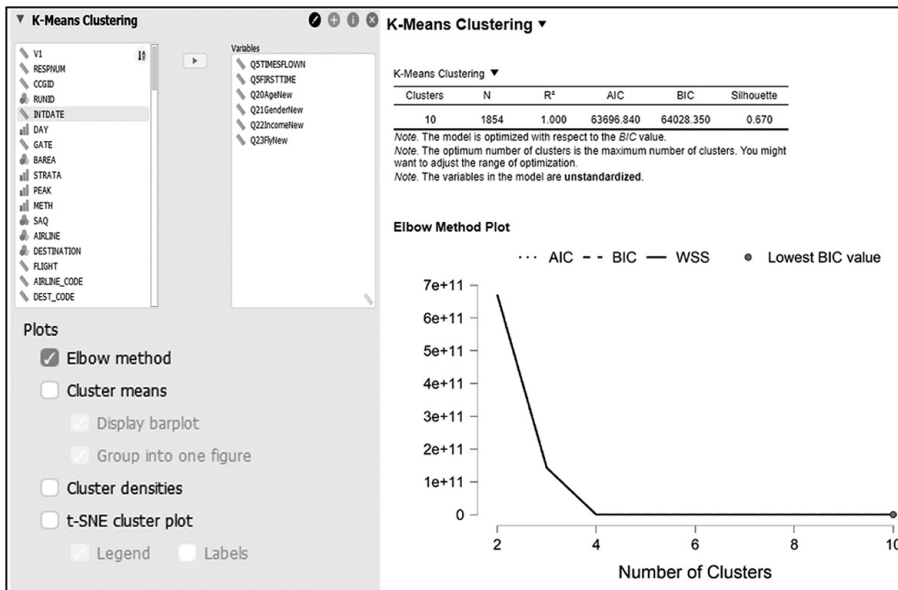


FIGURE 9.16 Steps to create the k-means Elbow method plot and the result using JASP.

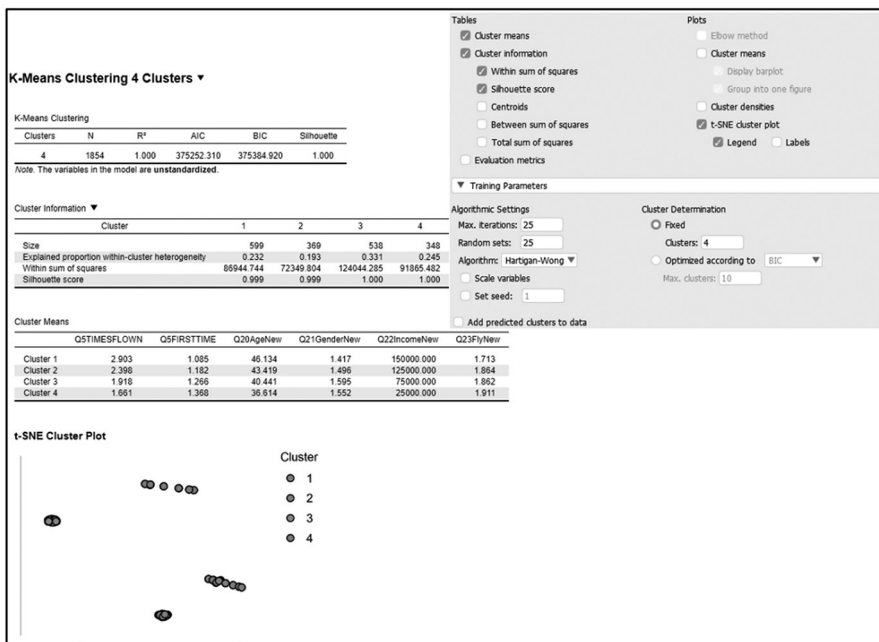


FIGURE 9.17(a) The steps to create four clusters in the k-means clustering method using JASP.

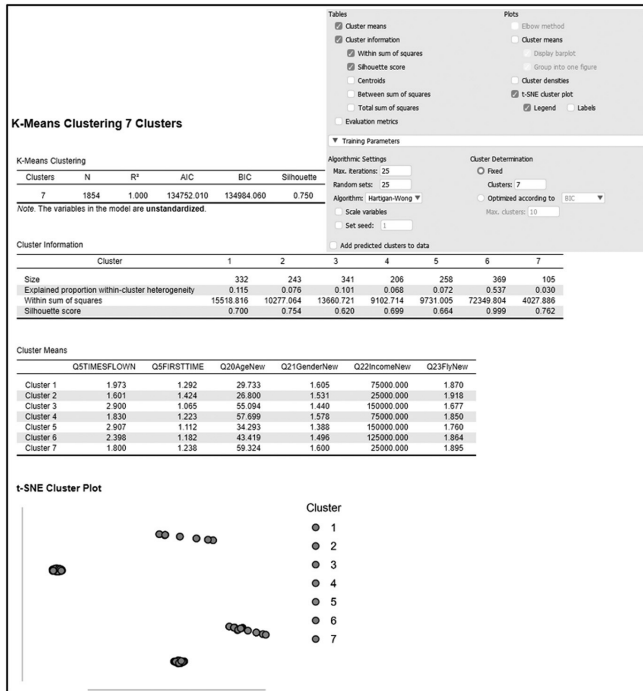


FIGURE 9.17(b) Steps to create seven clusters in the k-means clustering method using JASP.

Redo all steps, add *Q7ALL* into a variable list, and observe the k-means clustering result, as shown in Figure 9.18(a) and (b).

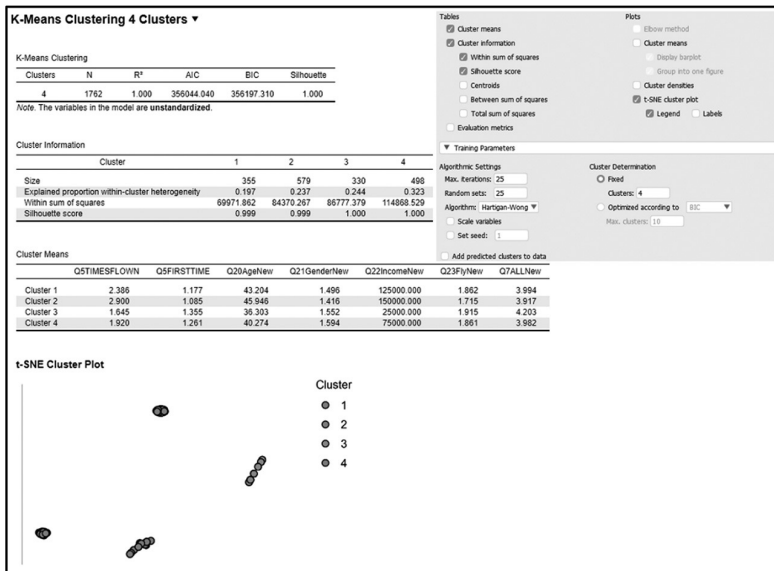


FIGURE 9.18(a) Steps to create four clusters in the k-means clustering method with *Q7ALL* using JASP.

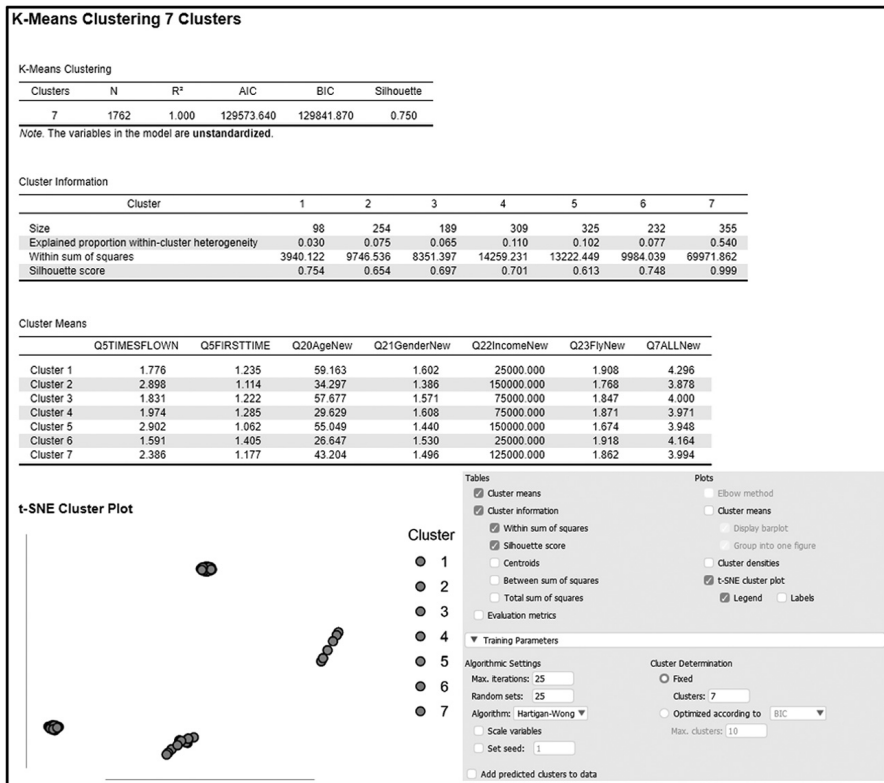


FIGURE 9.18(b) Steps to create seven clusters in the k-means clustering method with *Q7ALL* using JASP.

Redo all steps, add *NETPRO* to the variable list, and observe the k-means clustering result, as shown in Figure 9.19(a), (b), and (c).

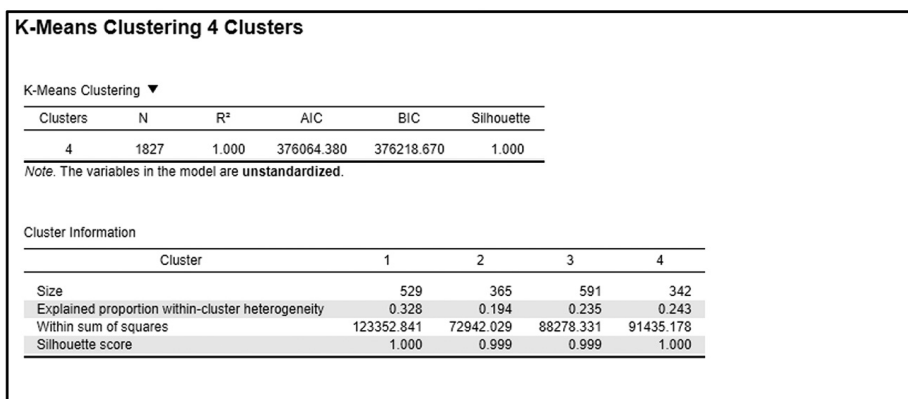


FIGURE 9.19(a) Four clusters in the k-means clustering method with *NETPRO* using JASP.

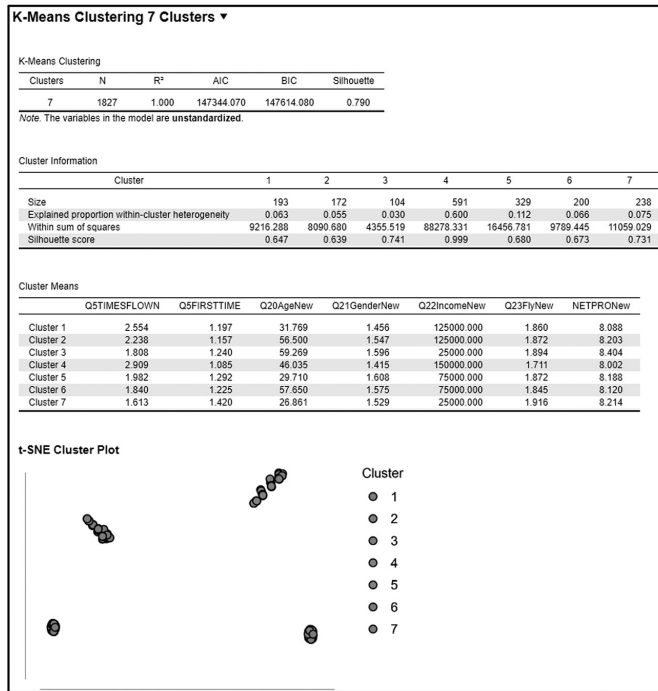


FIGURE 9.19(b) Seven clusters in the k-means clustering method with NETPRO using JASP.

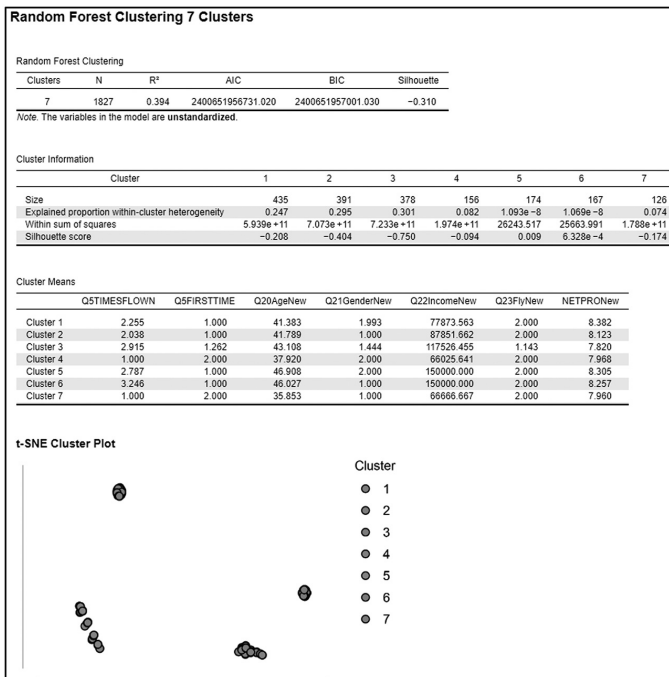


FIGURE 9.19(c) Seven clusters in the random forest clustering method with NETPRO Using JASP.

SOLUTION IN PYTHON

Import the well-prepared data set into Orange3. Select variables using the *Select Columns* widget and connect to a *k-Means* widget, as shown in Figure 9.20(a), (b), and (c).

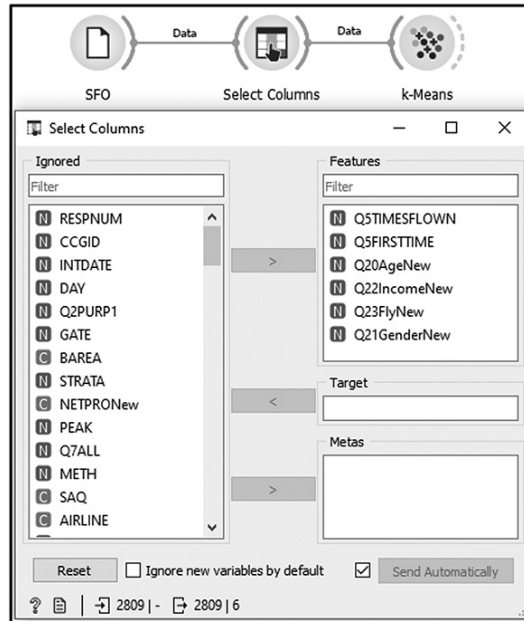


FIGURE 9.20(a) Select corresponding variables into features.

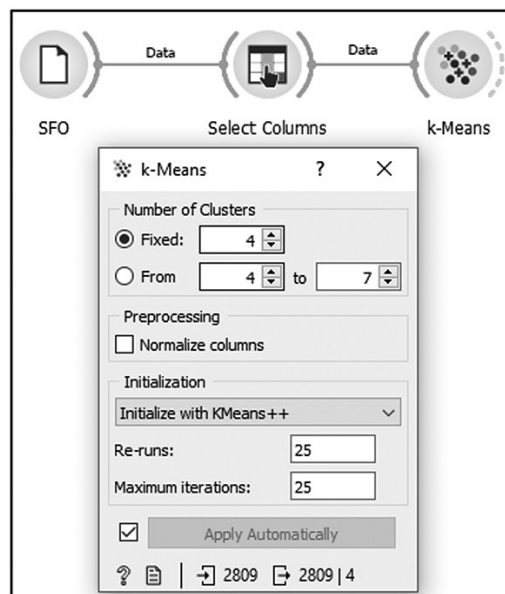


FIGURE 9.20(b) Steps to create four clusters in the k-means clustering method in Orange3.

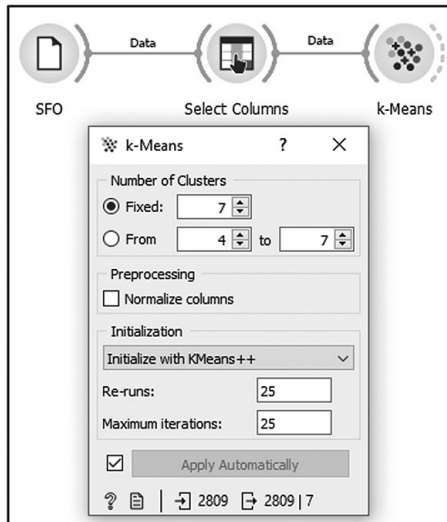


FIGURE 9.20(c) Steps to create seven clusters in the k-means clustering method in Orange3.

You can view the output data table and centroids data table by connecting to the *Data Table* with the *k-Means* widget, as shown in Figure 9.21(a)–(d).

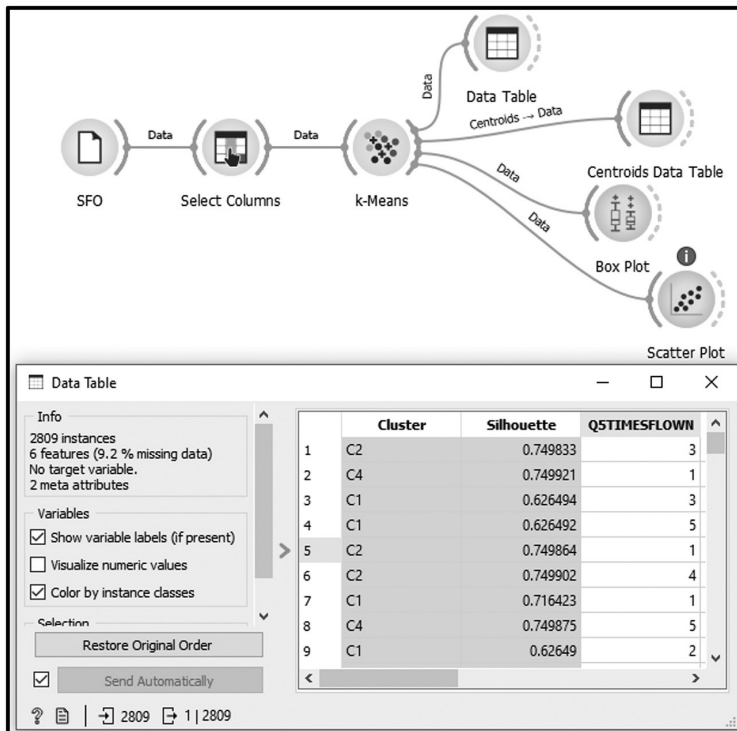


FIGURE 9.21(a) Steps to read the silhouette output of four clusters using the k-means clustering method in Orange3.

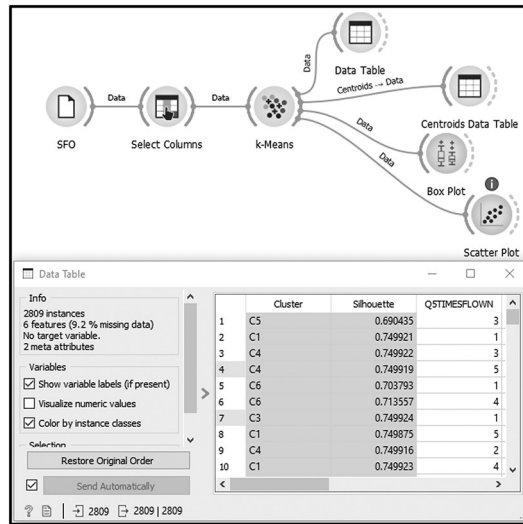


FIGURE 9.21(b) Steps to read the silhouette output of seven clusters using the k-means clustering method in Orange3

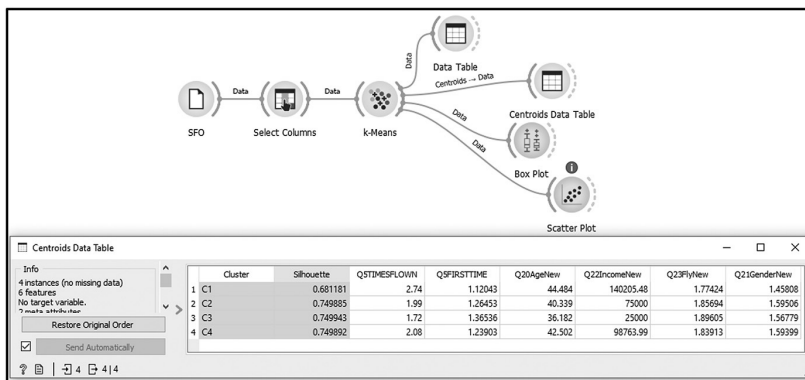


FIGURE 9.21(c) Steps to read the centroids output of four clusters using the k-means clustering method in Orange3.

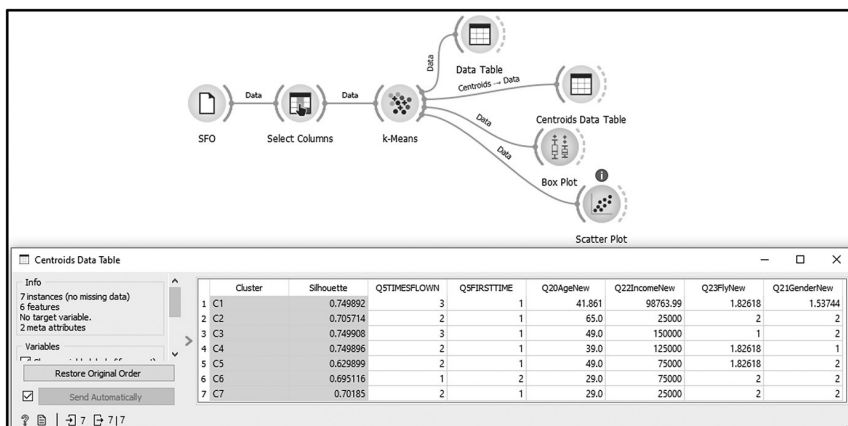


FIGURE 9.21(d) Steps to read the centroids output of seven clusters using the k-means clustering method in Orange3.

If we place the data into a scatter plot, we can also observe the clusters and silhouettes. The steps and results are shown in Figure 9.22(a) and (b).

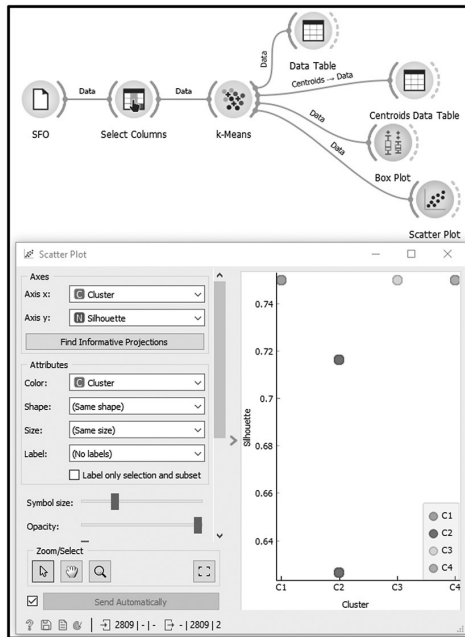


FIGURE 9.22(a) Steps to read four clusters using the k-means clustering method in a scatter plot.

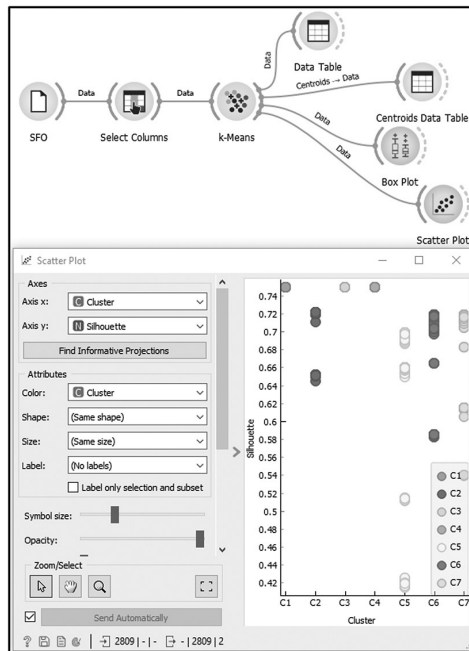


FIGURE 9.22(b) Steps to read seven clusters using the k-means clustering method in a scatter plot.

You can also observe the differences between clusters in the box plots, as shown in Figure 9.23(a) and (b).

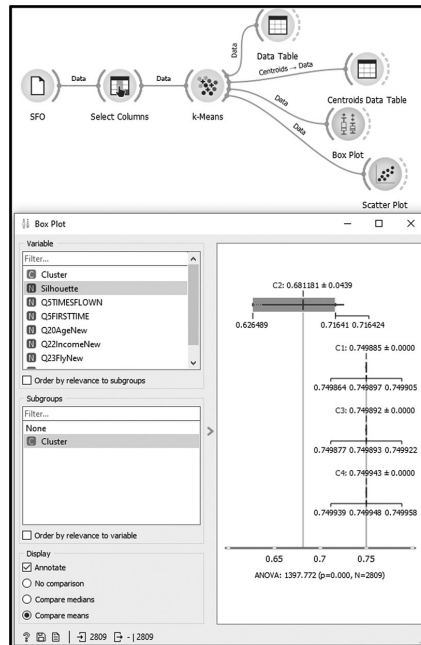


FIGURE 9.23(a) Steps to read four clusters using the k-means clustering method in the box plot.

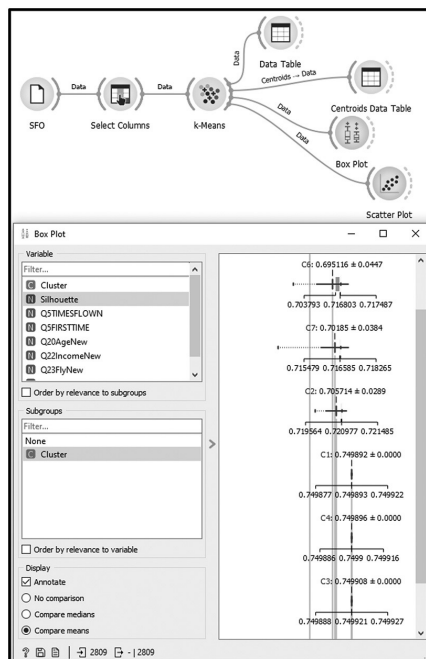


FIGURE 9.23(b) Steps to read seven clusters using the k-means clustering method in the box plot.

We can now create four and seven clusters using *Q7ALL*, as shown in Figure 9.24(a)–(d).

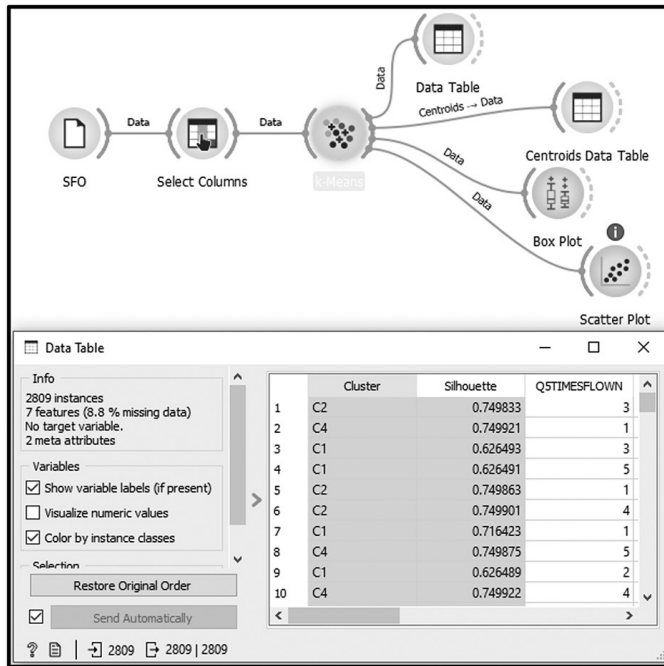


FIGURE 9.24(a) Steps to create four clusters in the k-means clustering method with *Q7ALL*.

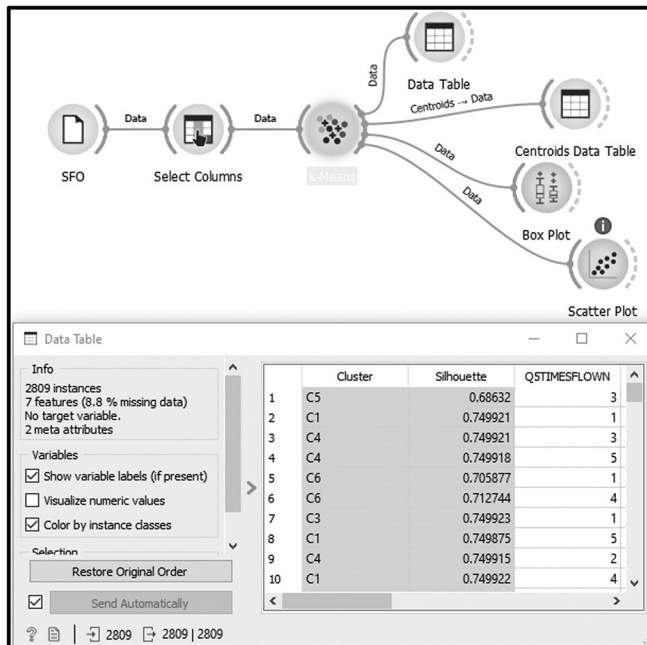


FIGURE 9.24(b) Steps to create seven clusters using the k-means clustering method with *Q7ALL*.

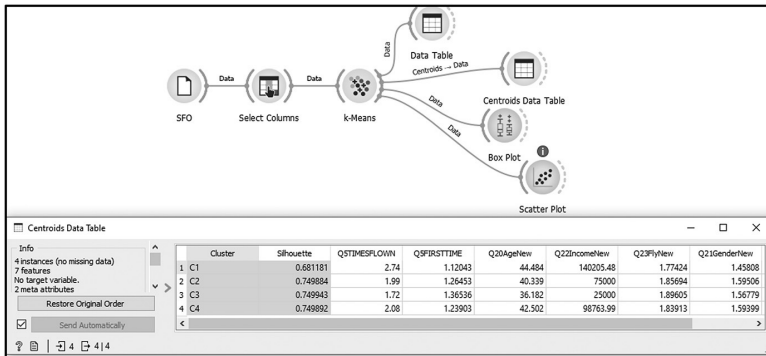


FIGURE 9.24(c) Steps to read the centroids output of the four clusters of the k-means clustering method.

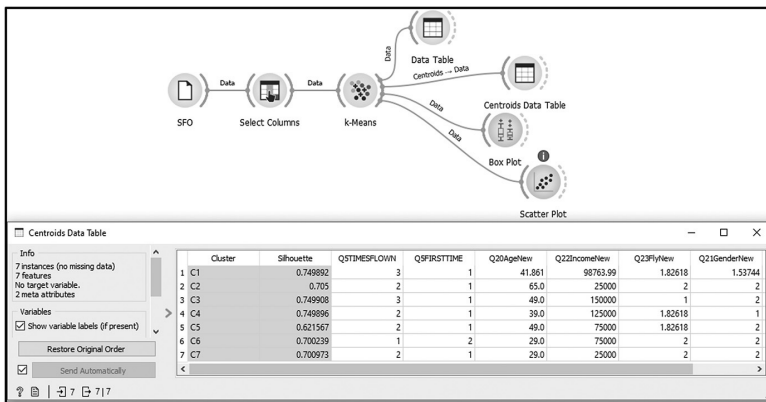


FIGURE 9.24(d) Steps to read the centroids output of seven clusters using the k-means clustering method.

We can also create four and seven clusters using *NETPRO*, as shown in Figure 9.25(a)–(d).

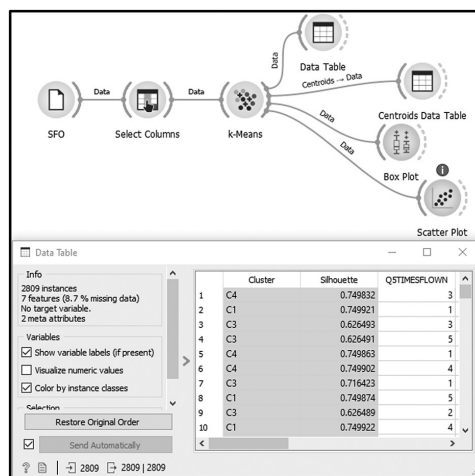


FIGURE 9.25(a) Steps to create four clusters in the k-means clustering method with *NETPRO*.

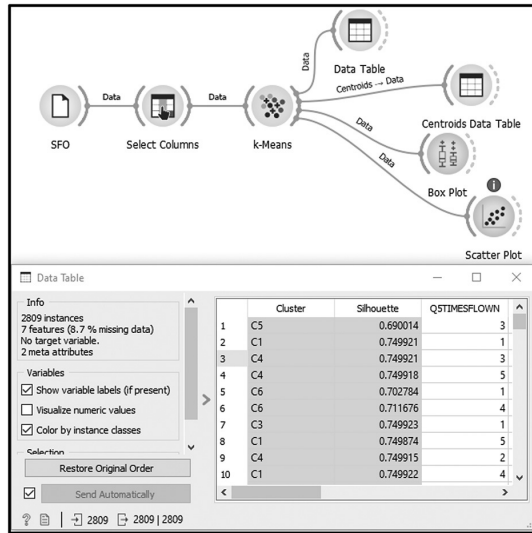


FIGURE 9.25(b) Steps to create seven clusters in the k-means clustering method with NETPRO.

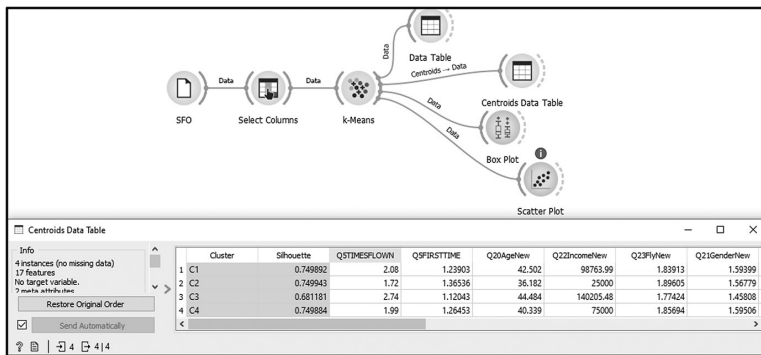


FIGURE 9.25(c) Steps to read the centroids output of four clusters in the k-means clustering method with NETPRO.

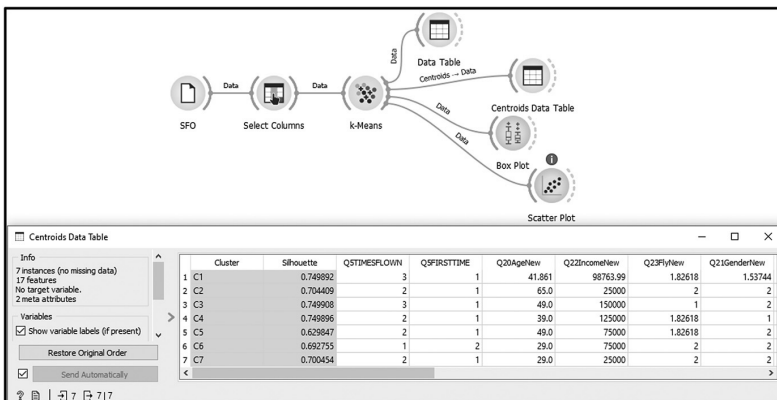


FIGURE 9.25(d) Steps to read centroids output of four clusters in the k-means clustering method with NETPRO.

CASE STUDY 9.2: CLUSTERING WITH THE SBA LOANS DATA SET

As the head of the Small Business Administration, you want to know the characteristics of the SBA loan program customers. Whom are you serving well, and who is having trouble paying off their loans? We want to know customer characteristics that influence repayment success. Use the following framed analytical question:

What type of customer uses the SBA loan program?

As an additional exercise, answer the following framed question:

What customer characteristics are strong indicators of loan repayment success?

Use the *FOIA Loans Data.csv* data set found in the *SBA Loans Data* folder on the *Case Data* depository. The *LoanStatus* variable has been binned as a binary variable (0 = CHGOFF, default, and 1 = PIF, paid-off loan). We use *GrossAmount*, *JobsSupported*, *TermInMonths*, *Business Type*, *NAICS code*, and *CDC_State* variables for additional features.

SOLUTION IN R

Because the SBA Loans database is too big to analyze directly in JASP, we should generate sample data from the original data set using a sample size calculator, as shown in Figure 9.26. You can use the following link or search online and find any calculators you like: <https://www.surveymonkey.com/mp/sample-size-calculator/>

FIGURE 9.26 Sample size calculator.

Once we figure out how many samples we need from this data set, go back to Excel or R and collect the sample data. Here are the steps you want to follow in R to take a sample.

```
index <- sample(1:nrow(SBA), 384)
SBAsample <- SBA[index,]
write.csv(SBAsample, file = "SBAsample.csv")
```

Now, import your sample data set into JASP and create the Elbow method plot, as shown in Figure 9.27(a) and (b).

Following the same steps from the last case, try to create four and seven clusters, and see if any one set of clusters gives more insight. The steps and results are shown in Figure 9.28(a) and (b).

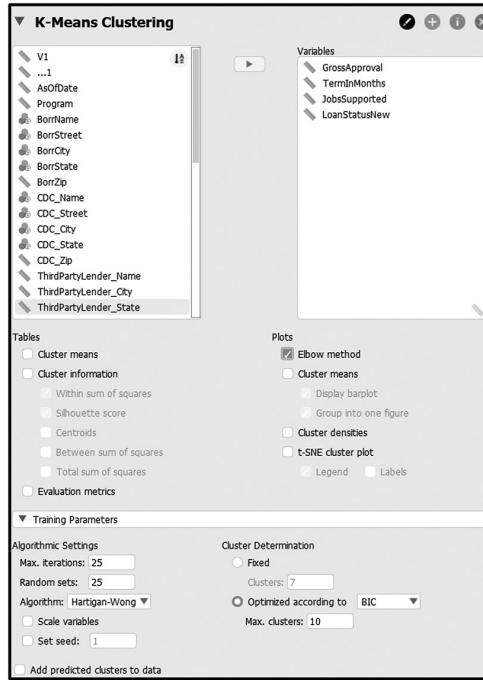


FIGURE 9.27(a) Steps to create the k-means Elbow method plot and the result.

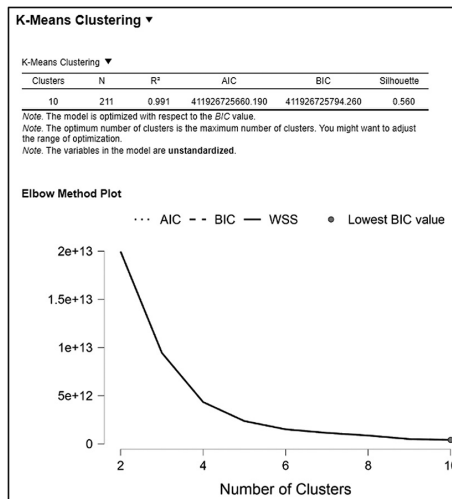


FIGURE 9.27(b) Steps to create the k-means Elbow method plot and the result.

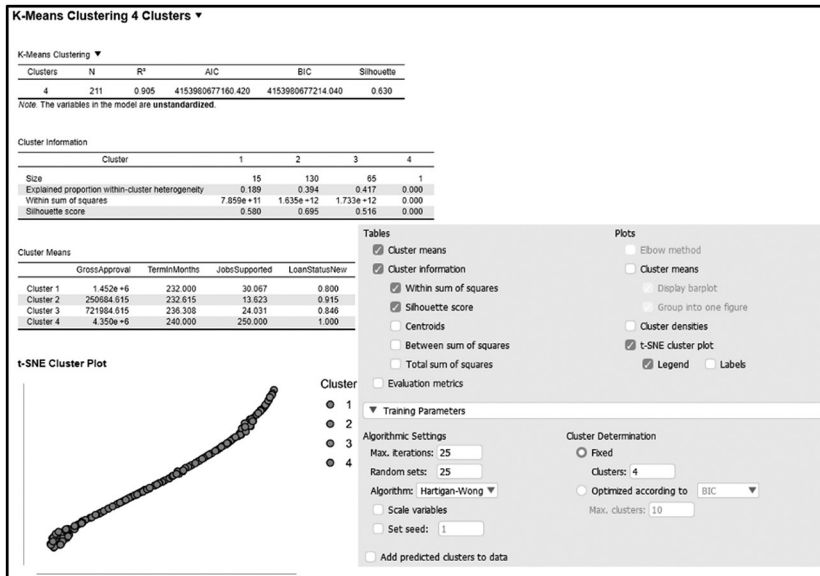


FIGURE 9.28(a) Steps to create four clusters in the k-means clustering method

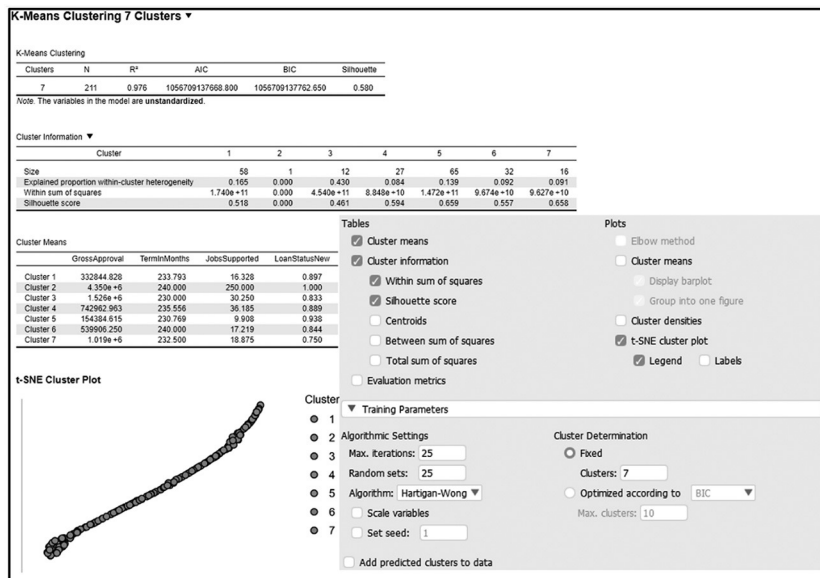


FIGURE 9.28(b) Steps to create seven clusters in the k-means clustering method

SOLUTION IN PYTHON

Import the well-prepared data set into Orange3. Select variables using the *Select Columns* widget and connect to a *k-Means* widget. Since the silhouette scores are not computed for >5000 samples, we need to create sample data from the original data set using a sample size calculator, as shown in Figure 9.29.

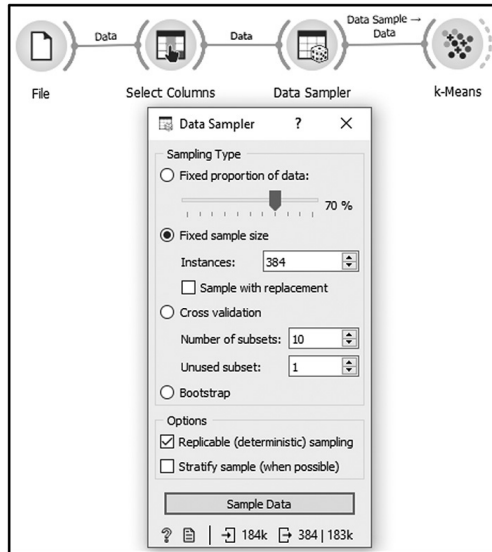


FIGURE 9.29 Steps to select sample size in Orange3.

Once you have the sample data, follow the steps in Figure 9.30(a) and (b) and create k-means clusters.

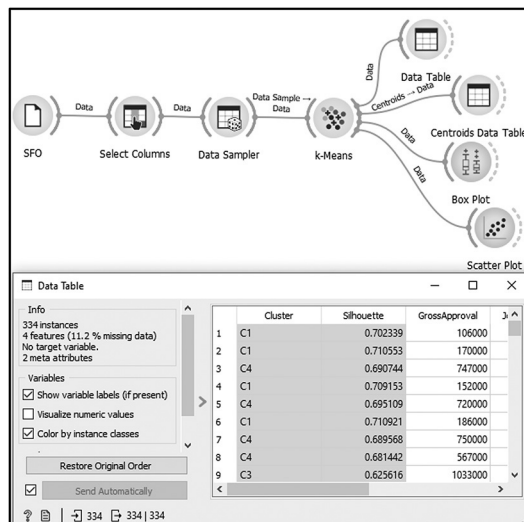


FIGURE 9.30(a) Steps to create four clusters in the k-means clustering method.

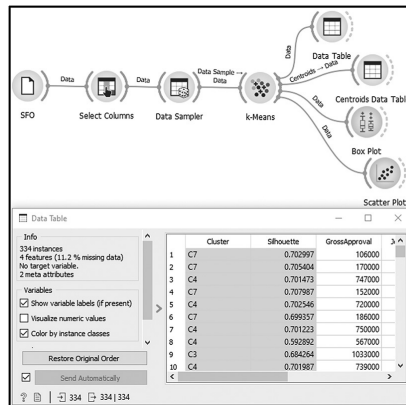


FIGURE 9.30(b) Steps to create seven clusters in the k-means clustering method.

You can view the output data table and centroids data table by connecting to the *Data Table* with the *k-Means* widget, as shown in Figure 9.31(a) and (b).

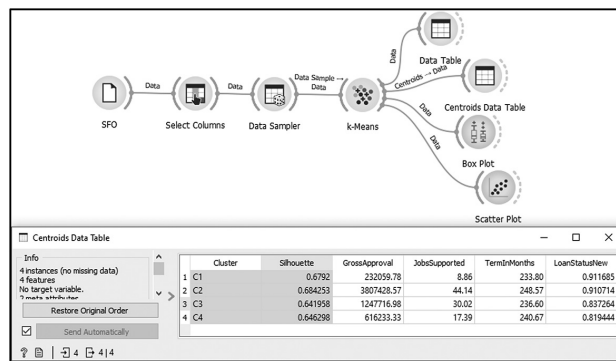


FIGURE 9.31(a) Steps to read the centroids output of four k-means clustering method.

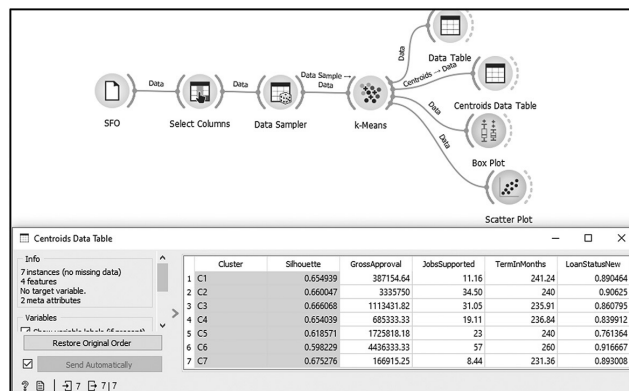


FIGURE 9.31(b) Steps to read the centroids output of seven k-means clustering method.

You can also observe the differences between clusters in the box plot in Figure 9.32 (a) and (b).

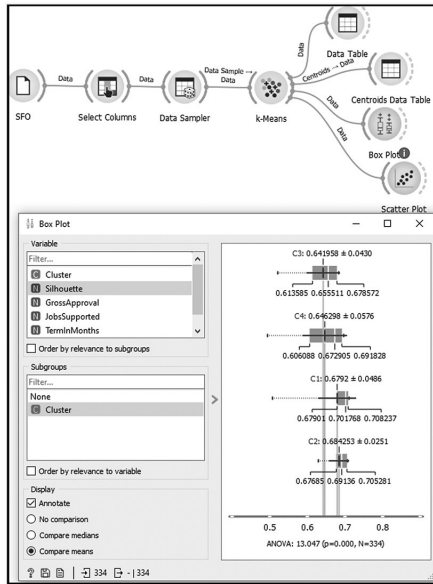


FIGURE 9.32(a) Steps to read the output of k-means in the box plots.

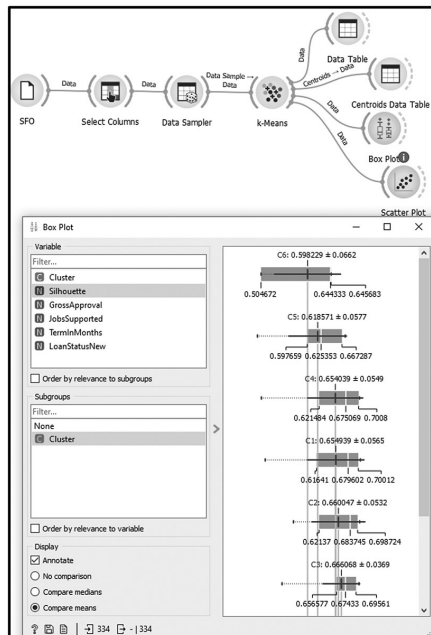
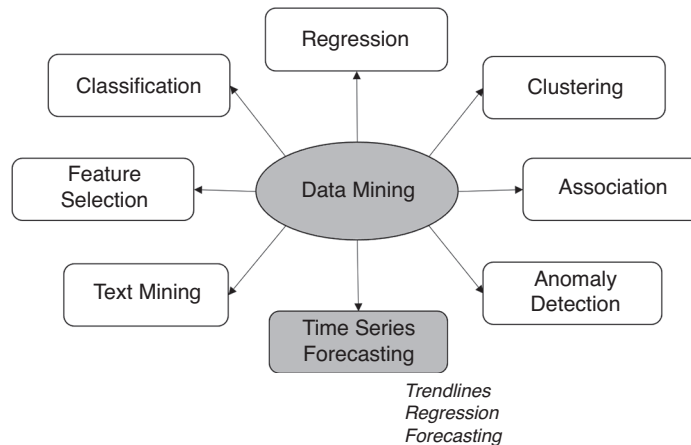


FIGURE 9.32(b) Steps to read the output of k-means in the box plots.

TIME SERIES FORECASTING



In this chapter, we extend the work of Chapter 7 on regression and apply it to time series analysis. Consider the situation where one of the data columns in our table is a date variable. We may use the date variable on the x-axis in a scatter plot with the y-axis (the value of some variable then) being one or more numeric columns of data, and where each column will show up as a series on the plot. The resulting plot is an actual scatter plot, but since the x-axis deals with time, we now call it a *time series*. We can then model each time series plot with a regression model. The simplest is a linear model; the resulting model is termed a *trendline*. This is most commonly and easily done in Excel. For the case studies, we develop the models in R (Jamovi of JASP, actually) and Python (Orange3), as has been the practice in previous chapters. The trendline equation can be used to interpolate or extrapolate (we also call it a *forecast*). Excel allows us to easily extend the trendline graphically into future periods, which is very useful. Sometimes, when we observe the resulting trendline, it does not fit very well. The tools provide other regression models, such as exponential and moving averages, which might give a better fit. We exercise all these options in the exercises later in the chapter.

This specialized data mining technique answers business questions such as “What is the trendline for this time series?,” “Can we forecast into future periods?,” and “What is the nature of the trend: linear, exponential, or some other form?”

As in previous chapters, we demonstrate the technique in the first exercise and allow for more challenging work in subsequent exercises and cases.

WHAT IS A TIME SERIES?

Time series analysis is a specific way of analyzing a sequence of data points collected over time. In time series analysis, we see that data points are recorded at consistent intervals over a set *period* rather than just recording data points intermittently or randomly.

We refer to a time series as a sequence of measurements or observations taken at successive equally spaced points in time. It is a sequence of discrete-time data. Examples of typical time series are hourly stock values in the stock exchange, monthly sales of inventory items, yearly corporate asset values for fixed assets by asset category, and the daily closing value of the Dow Jones Industrial Average.

Time series may have four components:

- A *time-based trend*, which describes the movement over time, and an average trend curve, for example. We would perform a regression analysis, giving us the average trendline.
- *Seasonal variations*, which represent seasonal changes from the average trendline. We are looking for variations to match the yearly seasons (winter, spring, summer, and fall).
- *Cyclical fluctuations* correspond to periodical but not seasonal variations. These could be due to natural (sunspots, El Niño) or artificial (economic cycles or college enrollment) variations.
- *Irregular variations* are other nonrandom sources of variations, including natural (hurricanes) or artificial (holidays or traffic accidents).

TIME SERIES ANALYSIS

Time series analysis is a form of data analysis for a sequence of data points collected over time. For time series analysis, we record data points at consistent intervals over a set period rather than just recording the data points intermittently or randomly. Time series analysis goes beyond collecting data over time to extracting information on trends, making forecasts, and predicting future states. What makes time series data different is that its analysis can show how variables change over time. In other words, time is crucial here because it shows how the data changes over the data points and in the final results. The time data provides an additional source of information and may set an order to the dependencies between other variables. Time series analysis typically requires many data points to ensure consistency and reliability. An extensive data set ensures we can cut through the noise. It also ensures that trends or patterns are not outliers and can account for seasonal variations. In addition, as noted before, time series data can be used for forecasting—predicting future values based on historical or past data.

Time series analysis helps organizations understand the underlying causes of trends or systemic patterns over time. Business users can see seasonal trends and dig deeper into why these trends occur. With modern analytics platforms, these visualizations can go far beyond trend graphs. When organizations analyze data over consistent intervals, they can also use time series forecasting to predict the likelihood of future events. Time series forecasting is part of predictive analytics. It can show likely changes in the data, like seasonality or cyclic behavior, which provides a better understanding of data variables and helps forecast better. Today's technology allows us to collect massive amounts of data daily, and it is easier than ever to gather enough consistent data for comprehensive analysis.

Examples of time series analysis cases include

- quarterly sales
- inflation rates over time
- unemployment rates over time
- bank balances
- weather data
- temperature readings
- heart rate monitoring (EKG)
- stock prices
- industry forecasts
- interest rates

Because time series analysis includes many categories or data variations, analysts sometimes must make complex models. However, analysts cannot always account for all variances and cannot generalize a specific model for every sample. Models that are too complex or try to do too many things can lead to a lack of fit. Lack of fit or overfitting models leads to models that do not distinguish between random error and genuine relationships, leaving analysis skewed and resulting in incorrect forecasts.

TYPES OF TIME SERIES ANALYSIS

We use different types and models of time series analysis to achieve different results. The following are types of time series analysis with example applications:

- *Descriptive analysis*: Identifies patterns in time series data, like trends, cycles, or seasonal variation
- *Forecasting*: Predicts future data. This type is based on historical trends. It uses historical data as a model for future data, predicting scenarios that could happen along future plot points.
- *Exploratory analysis*: Highlights the main characteristics of the time series data, usually in a visual format
- *Segmentation*: Splits the data into segments to show the underlying properties of the source information
- *Curve fitting*: Plots the data along a curve to study the relationships of variables within the data

WHAT IS FORECASTING?

Time series forecasting analyzes time series data using modeling to make predictions and inform strategic decision-making. Predictions are not always exact, and forecast likelihoods can vary wildly, especially when dealing with fluctuating variables in time series data and factors outside our control. However, forecasting insight about which outcomes are more likely—or less likely—to occur than other potential outcomes is useful in a business context.

It follows that the more numerous and comprehensive the data is, the more accurate the forecast. Forecasting and prediction are generally understood to mean the same thing, but there is a subtle difference. In some industries, forecasting might refer to data values at a specific future point, while prediction refers to future data in general. Time series analysis involves developing models to understand the data and the underlying causes of the forecasted values. Deep and thoughtful analysis can provide the “why” behind the visible outcome. Once informed by forecasting, the following steps of what to do with that knowledge and the predictable extrapolations of what might happen in the future are supported.

Let’s see how all these concepts play out in a simple time series and forecasting exercise. Consider that it is the year 2012, and US economists are concerned that the Chinese economy, which at this point is a small fraction of the US economy, is steeply rising and that China may someday compete economically with the US. When will that day be? When will the US and Chinese GDPs be equal? Ten, twenty, or thirty years in the future? The answer surprised the economists. Let’s find out. We will use a well-regarded economic data set from the World Bank (World Bank 2022).

EXERCISE 10.1 – ANALYSIS OF THE US AND CHINA GDP DATA SET

We can use linear regression and other forms of curve fitting to create forecasting models.

Find the file *WDISelectedData.xlsx* in the *Chapter 10* file folder in the *Case Data* depository. It was derived from the World Bank World Economic Indicators data bank (World Bank 2022). Open *WDISelectedData.xlsx* using Excel. We are going to answer these questions:

What are the Chinese GDP and US GDP growth rates?

When is the Chinese GDP projected to catch up with the US GDP?

Following our practice of not changing the raw data, we will create a shaped file for our analysis in a new spreadsheet. Select the title row for the file and enter it into a new spreadsheet. Repeat the GDP data rows for the US and China. It would help if you had a simple table (Figure 10.1).

Let’s open the spreadsheet with Excel. To create a properly shaped file, delete the first two columns and the column with country abbreviations. Replace the dates on the top row

	A	B	C	D	E	F	G	H	I	J	K	
1	Series Name	Series Code	Country Nam	Country Code	2000 [YR2000	2001 [YR2001	2002 [YR2002	2003 [YR2003	2004 [YR2004	2005 [YR2005	2006 [YR2006	2007 [YR2007
2	GDP (current	NY.GDP.MKTF	China	CHN	1.1985E+12	1.3248E+12	1.4538E+12	1.641E+12	1.9316E+12	2.2569E+12	2.71E+12	3
3	GDP (current	NY.GDP.MKTF	United States	USA	1.0285E+13	1.0622E+13	1.0978E+13	1.1511E+13	1.2275E+13	1.3094E+13	1.3856E+13	1

FIGURE 10.1 Scraped and cleaned data for the US vs. China GDP in US dollars ready for analysis.

with simple dates (2000, 2001, 2002, etc.). Create another two rows with the country data normalized by dividing by 1 trillion (1,000,000,000,000). This gives us numbers we can easily recognize. Swap China and the US so the US is in the first row of data (so the colors of the resulting bars are semantically correct, and the colors associated with each country are what the viewer expects). Select the data rows and insert a bar chart into the spreadsheet. Enter the titles of each series and put the dates into the x-axis. Move the legend on the chart to the bottom.

Now let's create a predictive model. Right-click on the US data points on the chart (make sure all data points for the US series are selected). Click on "Add a Trendline." Make sure to click to have the R-squared factor added. You will get a linear model (Figure 10.2).

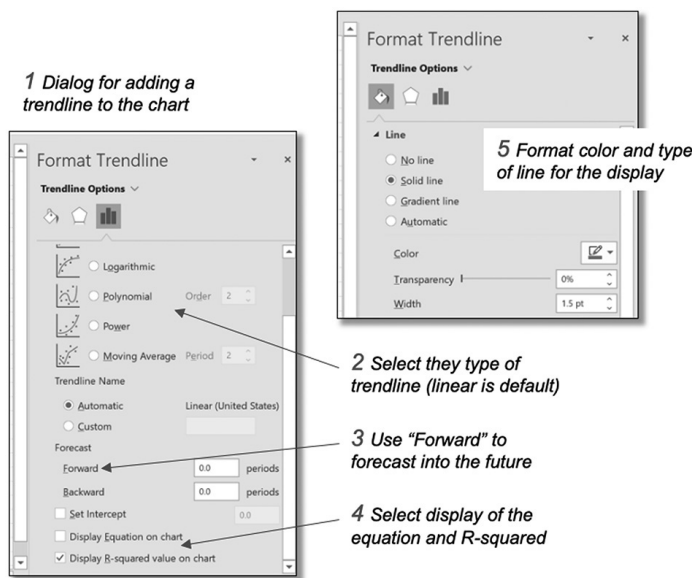


FIGURE 10.2 Using dialogs to add and format the trendline

Repeat these steps for China. Make it a linear model as well. What is the R-squared factor for each linear model? Acceptable? Does the linear model look suitable for China's GDP trend?

Click on the data for China once again and add another trendline. Select an exponential model for this new trendline. Make sure to ask for R-squared and change the color of this trendline to red. Does it fit the data better?

Now, let's use these as forecasts. Click on the US linear model and select "Format Trendline." You can add a forecast in the middle of the dialog box. Note that the data is in years. That indicates that the "periods" in a forecast will be "years." In the trendline dialog box, find the "periods" section and enter five periods (years in this case). Repeat for the Chinese exponential model. Where do the lines cross? What is the answer to our question?

Repeat the Chinese GDP linear model and see what five-year forecast it creates (Figure 10.3).

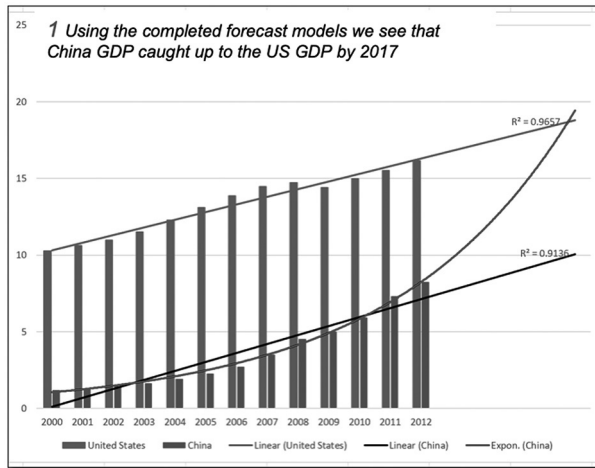


FIGURE 10.3 Steps in creating forecasts to compare US and Chinese GDP and analysis of the crossover year.

CASE STUDIES

The following case studies look at time series data from several industries. The San Francisco Airport survey provides marketing data, from which we investigate the changes in customer perceptions of the airport over time. We will also look at a financial data set for the Small Business Administration loans to small businesses, where we investigate the changes in loans over time. Last, we will look at seasonal variations and data examples and how to analyze for long-term trends versus seasonal changes.

CASE STUDY 10.1: TIME SERIES ANALYSIS OF THE SFO SURVEY DATA SET

You could explore many variables in marketing airport services to your customers. These could all be analyzed for trends over time, such as passenger demographics. However, for the project we are pursuing in this case study, we will focus on customer satisfaction indices over time and try to glean some insights from the trends.

We will focus on the opinion variables to see how customers' opinions have changed. We will include the overall satisfaction score *Q7ALL* (make sure to use the full range of scores from 1 to 5; exclude 6 and 0) and all the other *Q7* satisfaction scores for particular issues (such as food, signage, and parking). We will also use the other indicator of satisfaction, the net promoter score, the variable *NETPRO*. Be sure to remove 0s from all the variables (replace with blanks) so they do not produce an error (remember a 0, in this case, is not a score or category, but it signifies the respondent left it blank, and it should not be counted).

Since we are looking at trends, we will need to create a table of *Q7* scores and *NETPRO* over time by extracting the corresponding columns with these variable scores from all the survey response tables for each year from 2013 to 2019. Use the *SFO Survey Data.xlsx* data set found in the *SFO Survey Data* folder in the *Case Data* depository.

The questions we will answer using time series analysis for this case is as follows:

What are the trends over time?

SOLUTION IN EXCEL

Using a data set of the combined 2018 and 2017 customer surveys for the San Francisco Airport, we can do an Excel trend analysis from one year to the next. Figure 10.4 is a slope chart (a form of time series) showing there was little change in the *NETPRO* score for the airport between 2017 and 2018. The slight downward slope shows a minor decay but is not significant. We only have two years' *NETPRO* scores, so all we could plot for the 2017–2018-time frame was a slope chart. We have another customer opinion variable over time, *Q7ALL* — the overall opinion score. If we plot this score over six years (2013–2018), we see a general upward trend but slight yearly variations (see Figure 10.5).

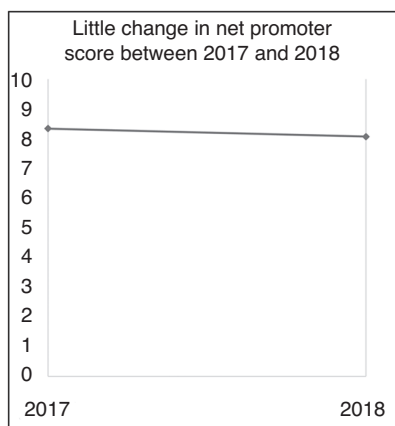


FIGURE 10.4 The net promoter (*NETPRO*) score between 2017 and 2018 shows a slight decrease in the two years (the net promoter scale is from 1 to 10).

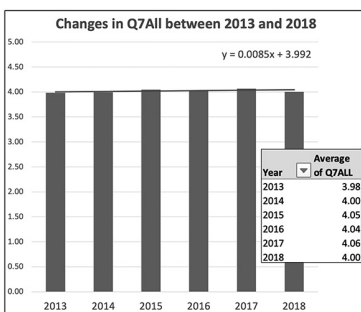


FIGURE 10.5 Changes in *Q7ALL*, the overall score for the airport over six years (2013–2018), on a scale from 1–5. The trendline equation shows a minimal, positive slope.

We could plot the time series for customer opinions of all the airport areas. Some areas perform at high levels (almost 4.2) and others are inferior performers (in the 3.6 level). We plot all opinions for all 13 areas over time as line charts (see Figure 10.6, the gray plots) and average all the opinions (the bar chart in Figure 10.6). We see that the average opinion over time peaks in 2016, and by 2018 it was declining, but overall, the trendline is upward (the line across the top of the bars). We could ask the following question:

When will the average score per area reach an average score of 4.0?

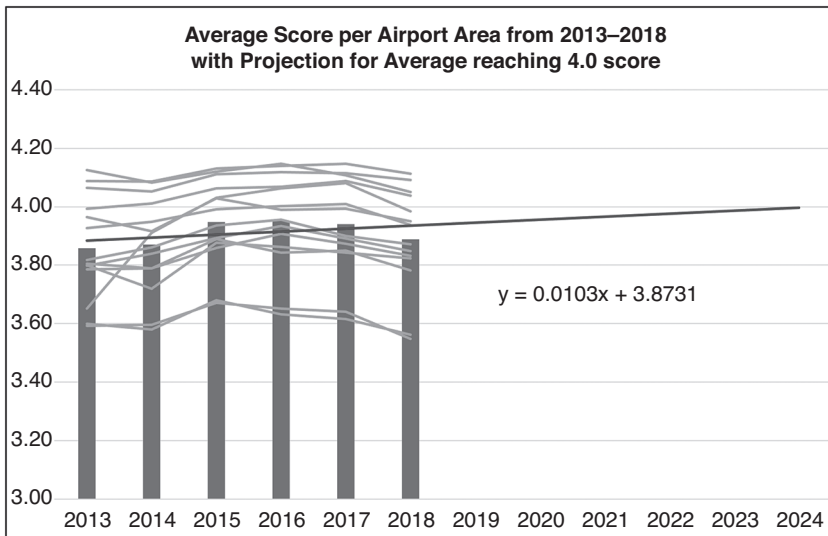


FIGURE 10.6 Change in the average score of all Q7 scores (the bars and the trendline is the average, and the gray lines are each of the 14 individual area scores) between 2013 and 2018, and a forecast to 2024.

We performed a forecast of the trendline of the average score and saw that it takes over six years before the average score reaches 4.0 if the current rate of increase persists, as we can see in Figure 10.6. There is a change in the average score of all Q7 scores (the bars and the trendline is the average, and the gray lines are each of the 14 individual area scores) between 2013 and 2018, showing slight increases over the years. At the rate of increase, shown by the trendline slope, it would take another six years for the average score to reach a 4.0 score.

CASE STUDY 10.2: TIME SERIES ANALYSIS OF THE SBA LOANS DATA SET

As the head of the Small Business Administration, you have been asked to report to the Secretary of Commerce and Congress on the loan program's impact over the years. You need to report on how many small businesses versus big businesses have been supported. Are you fulfilling the mission of loaning to small businesses for the most part? How many small businesses have benefited over the years? How much money has been lent out over the years, and how have those loans performed? How has the default rate changed from year to year? What is the trend in the type of industry receiving business loans? Do you see a trend in the questions? The words “over the years” or “from year to year” are clues that we need to do some time series analysis. Use the following framed analytical questions:

What is the trend in the ratio of loans to small businesses versus relatively large businesses over the past ten years?

What is the trend in the default rate of loans over the past 20 years?

What is the trend in the overall number of loans and total money lent by year over the last 20 years?

As an additional exercise, answer the following framed question:

What are the trends for the total amount of loans by industry and by year for the top five industries?

Use the *FOIA Loans Data.csv* data set found in the *SBA Loans Data* folder in the Case Data depository. We use the variable *LoanStatus*, which should be binned as a binary variable (0 = CHGOFF, default, and 1 = PIF, paid off loan) as the variable for the default rate questions. For additional features, we will use *GrossAmount*, *Business Type*, *NAICS* code, and *JobsSupported* (be sure to bin it as SMALL = 50 jobs or more minor, LARGE = more than 50 jobs supported.)

As the industry variable, we should use the NAICS industry code and extract and enter into a new variable called *Industry* the first two digits of the NAICS code. Create another variable name *IndustrySector* and enter the 2-number code with the corresponding name of the sector from the *NAICS Industry Sector* table provided in the same folder as the data set.

Aggregate by year using a pivot table; export the tables to the appropriate CSV files for further analysis using R and Python.

SOLUTION IN R

Use the *PivotTable* function and aggregate all needed information in Excel. Once the pivot table is ready for analysis, import it into JASP. For the first question, we need to change the variable, and the trend will be indicated easily, as shown in Figure 10.7(a) and (b).

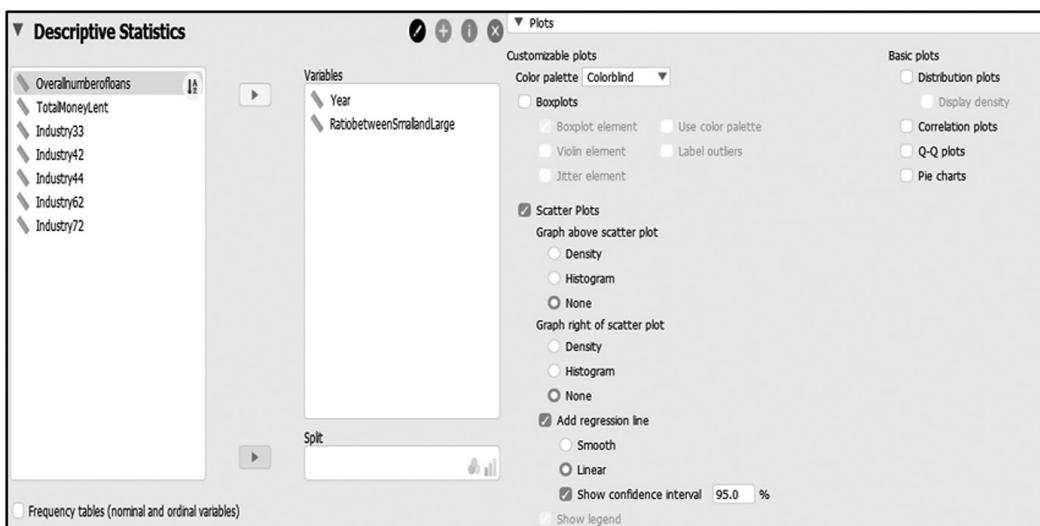


FIGURE 10.7(a) Steps and result of the scatter plot with a trendline.

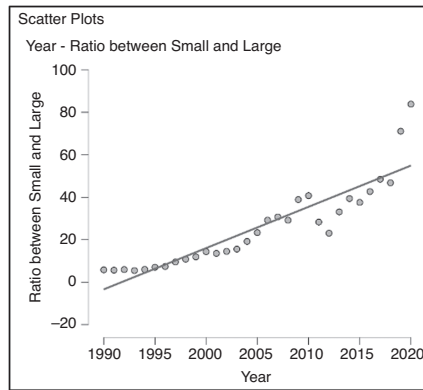


FIGURE 10.7(b) Steps and result of scatter plot with the trendline.

To answer the third question, follow the same steps and add the new variable *overallnumber-loans*. The result is shown in Figure 10.8.

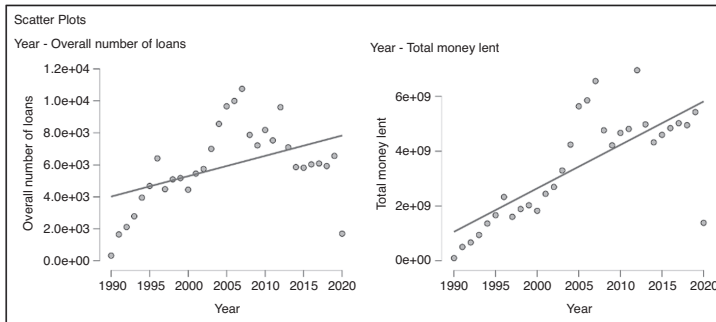


FIGURE 10.8 Steps and results of the scatter plots with trendlines.

Follow the same steps and find trends for the total amount of loans by industry and by year for the top five industries, as shown in Figure 10.9.

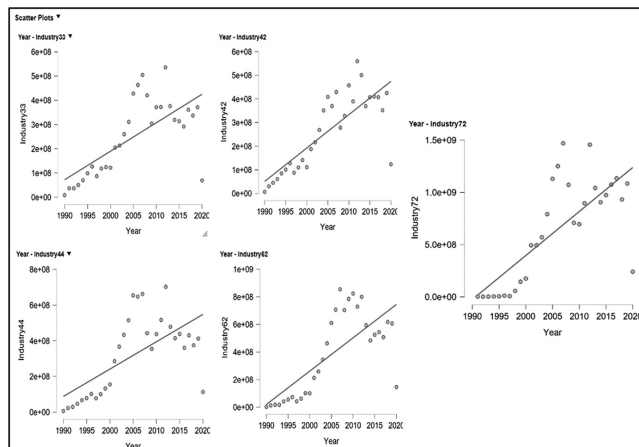


FIGURE 10.9 Trends for the total amount of loans by industry and year.

SOLUTION IN PYTHON

Use the *FOIA Loans Data.xlsx* data set found in the *SBA Loans Data* folder in the Case Data depository. Use the *PivotTable* function and aggregate all needed information in Excel. Once the pivot table is ready for analysis, import it into Orange3. To answer each question, we can generate a scatter plot and add a regression line by using a scatter plot after selecting columns. You should be able to get the result like that shown in Figure 10.10.

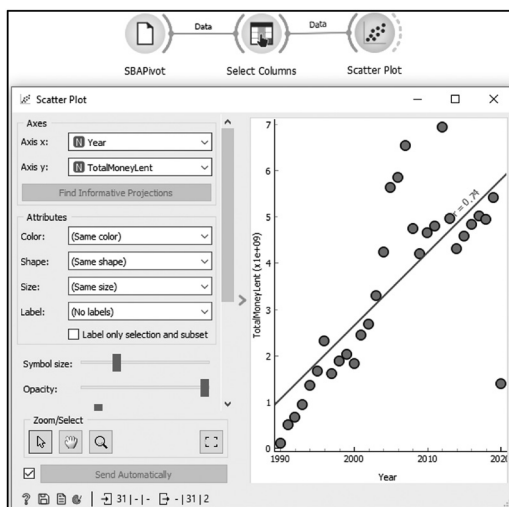


FIGURE 10.10 Steps and result of the scatter plot with the trendline.

For the next question, you can change the variable in the select variable widget, and the plot will be changed automatically, as shown in Figure 10.11.

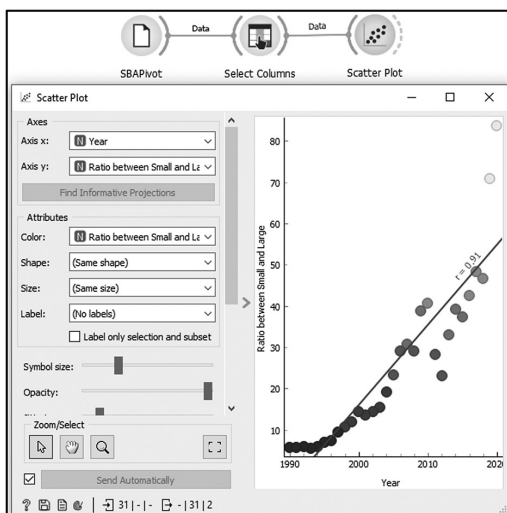


FIGURE 10.11 Steps and result of the scatter plot with the trendline.

To answer the third question, follow the same steps and add the new variables *overallnumberloans* and *totalmoneyleft*, as shown in Figure 10.12(a) and (b).

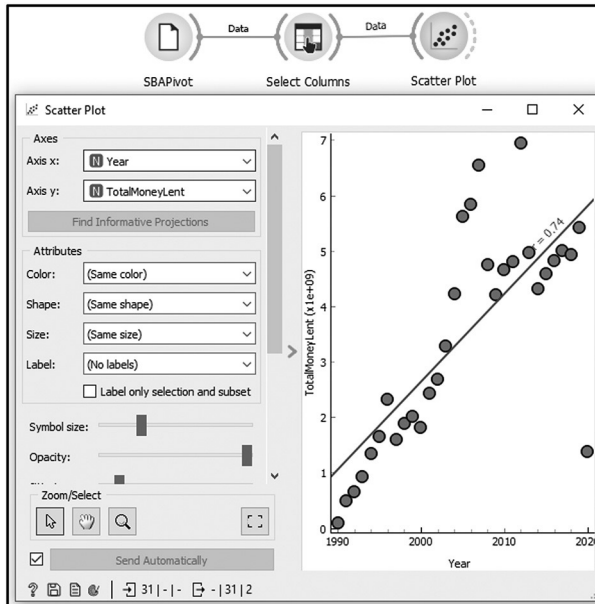


FIGURE 10.12(a) Steps and result of the scatter plot with a trendline on the total amount.

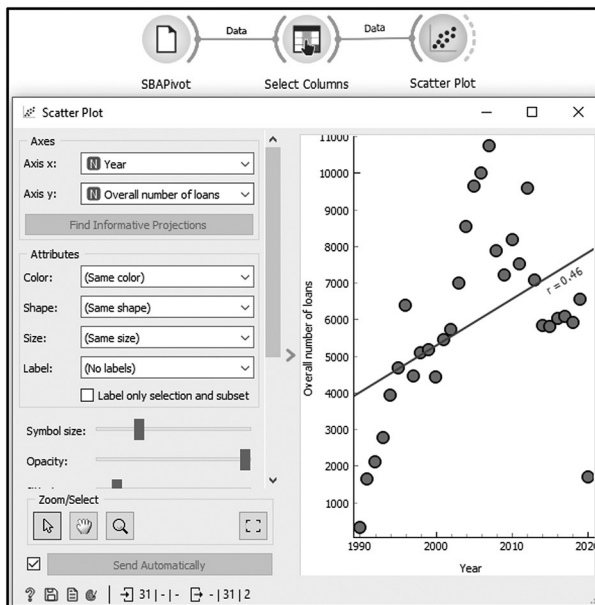


FIGURE 10.12(b) Steps and result of the scatter plot with a trendline on several loans.

Follow the same steps and find trends for the total amount of loans by industry and by year for the top five industries, as shown in Figure 10.13.

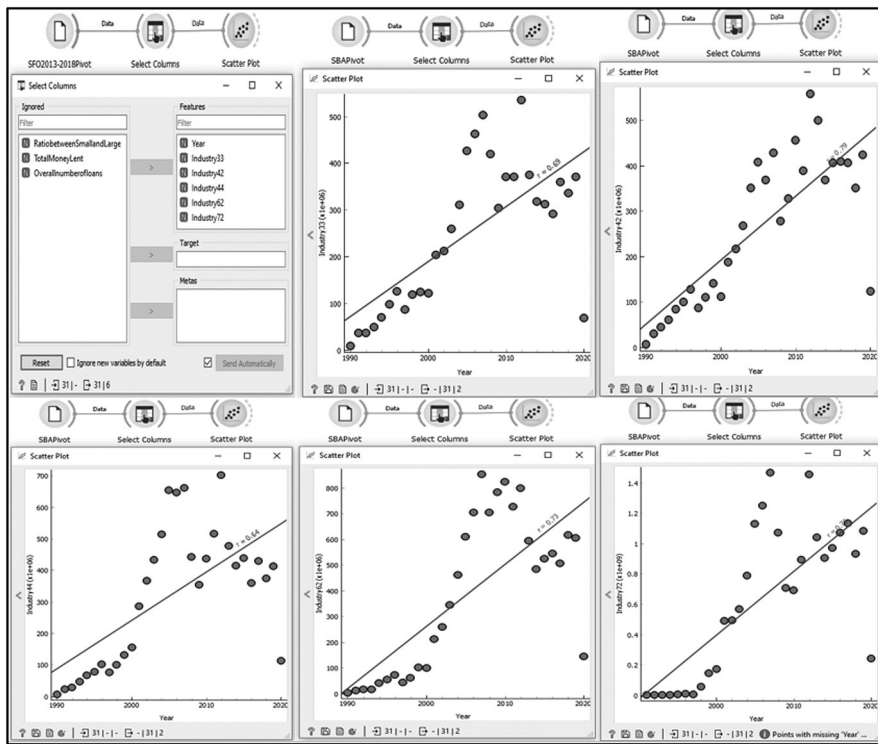


FIGURE 10.13 Steps and result of the scatter plot with a trendline for various industries.

CASE STUDY 10.3: TIME SERIES ANALYSIS OF A NEST DATA SET

A homeowner is very proud that she installed Nest thermostats throughout her house. For the past 48 months, she has been tracking household heating and cooling using the Nest report sent by the Nest company and comparing it against the local weather station heating and cooling data. The Nest reports on heating and cooling hours per month. The weather station reports the cooling and heating data in degree days. However, we can make a good comparison of these two different units of the trends. As a homeowner, we have several questions we want to answer.

Does the Nest data track well with respect to the weather station data?⁹ In other words, do they both show the same pattern?⁹

What is the overall trend for heating?⁹ For cooling?⁹ Can we make some statements concerning the pattern we see?⁹ Do we generally have weather patterns of cooling or heating?⁹

We will plot the cooling and heating data as a time series for the Nest and weather station data. If they are both on the same graph but on a different axis, we can easily superimpose the two and make a good comparison. Alternatively, we can normalize all data columns to the maximum and plot the normalized Nest and weather station sets for cooling on one graph and heating

on another. By performing a linear regression, we can see the general heating and cooling patterns. Once normalized, we can subtract the Nest data from the weather station data and see if the Nest thermostat data tracks the weather station data.

Degree days are based on the assumption that when the outside temperature is 65°F, we do not need heating or cooling to be comfortable. Degree days differ between the daily temperature mean (high temperature plus low temperature divided by two) and 65°F. If the daily temperature average is above 65°F, we subtract 65 from the mean temperature for the day, and the result is a *Cooling Degree Day* designation for that day. If the temperature average is below 65°F, we subtract the mean from 65, and the result is a *Heating Degree Day* designation. There is no easy way to convert degree days to hours of heating or cooling as measured by the Nest thermostat since there are many factors relating the two: the amount of cooling or heating needed to manage the interior departure of the hours and the thermostat settings at any one time and the outside temperature at that moment in time. However, we can still calibrate our Nest thermostat to the number of monthly heating or cooling degree days.

We can create a model to predict the hours of heating or cooling needed in her house from the degree days prediction. We will create two models, one for heating hours and another for cooling hours, based on degree days projections from the weather bureau using weather data and Nest data as the training data set.

What model can we use to predict hours of cooling and hours of heating for the house, given the projected degrees of cooling or heating for any month? A time series model, of course!

SOLUTION IN PYTHON

Use the *NEST Data.xlsx* data set found in the *Chapter 10* folder in the Case Data depository. Prepare the Nest data in Excel or R and create normalized variables. Once the data is ready for analysis, import it into Orange3. Select all variables you want to analyze by connecting a select columns widget after importing the data set into Orange3. Then connect to a line chart widget to view the pattern. See Figure 10.14.

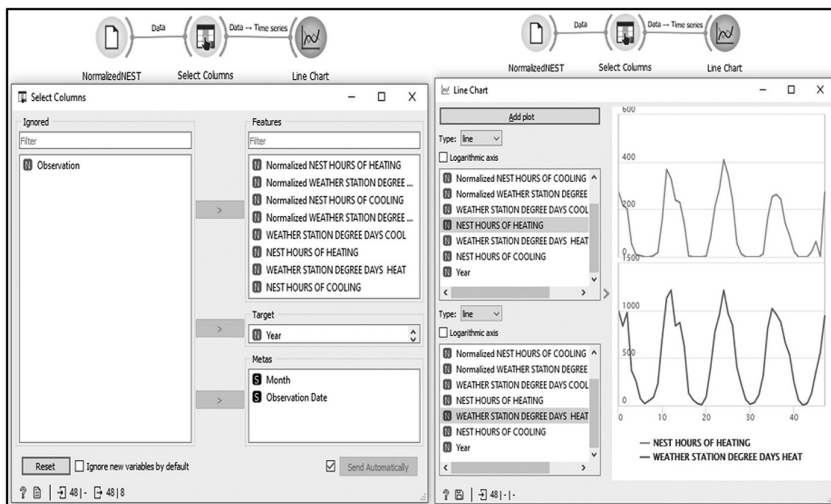


FIGURE 10.14 Steps and results for the time series analysis in Orange3.

To view the overall trend for heating and cooling, we need to generate all line plots, as shown in Figure 10.15.

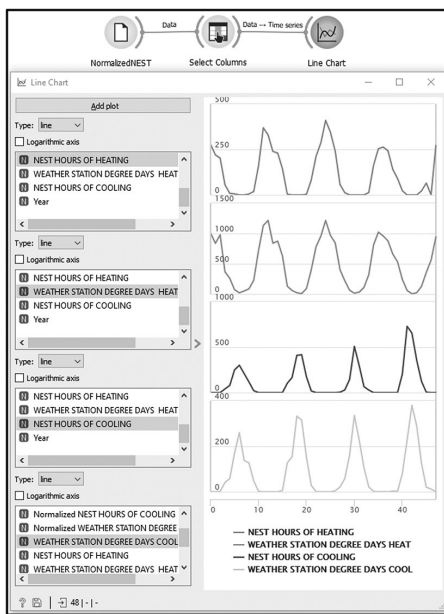


FIGURE 10.15 Steps and result of the overall trend.

To build models to predict the hours of heating or cooling, we need to connect linear regression and prediction, as shown below. Then we can view the result in the data table and scatter plot, as shown in Figures 10.16(a) and (b).

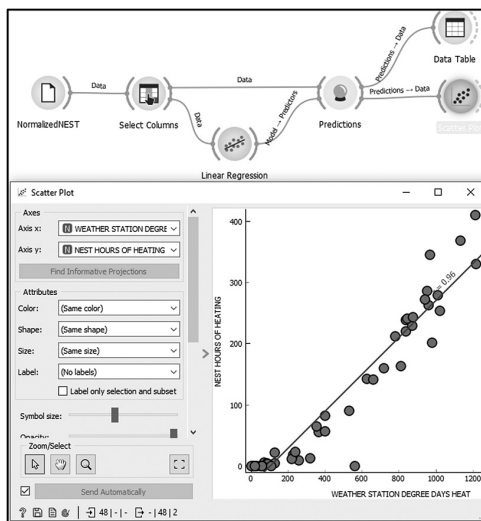


FIGURE 10.16(a) Steps to create a scatter plot.

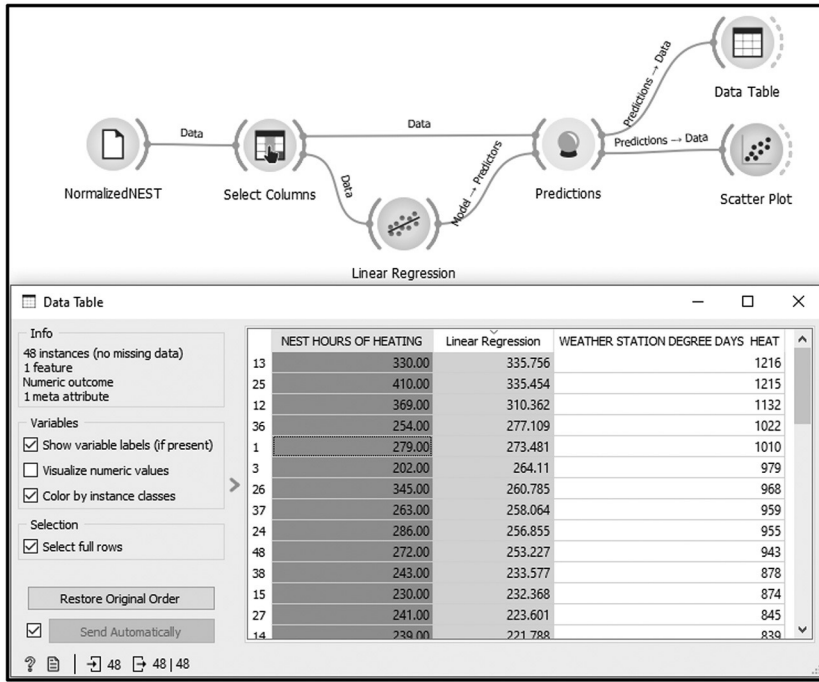
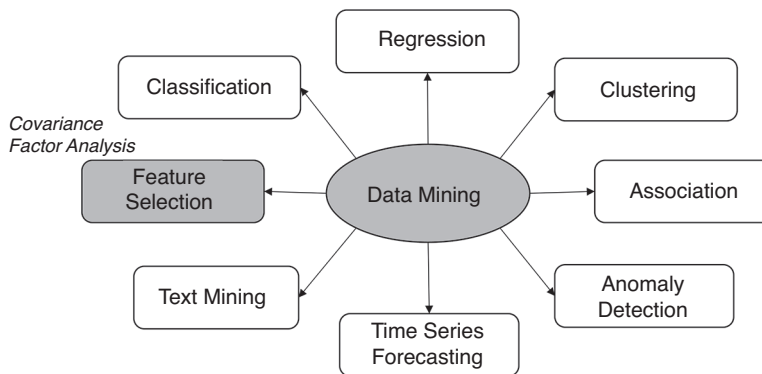


FIGURE 10.16(b) Steps to view the data table.

REFERENCE

World Bank Open Data: World Development Indicators. World Bank, 2022, <https://datacatalog.worldbank.org/search/dataset/0037712/World-Development-Indicators>.

FEATURE SELECTION



In this chapter, we consider two techniques to reduce the complexity of our models by reducing the number of input variables to a model. The first technique, considered an accurate feature selection algorithm, uses the covariance matrix. The second involves combining highly correlated features into a few factors before using the factors in regression or some other model. Figure 11.1 shows the place of feature selection in the data mining toolbox and two of the most common approaches.

Consider a data set with many tens of columns (or potential input variables), also known as *features*. We want to build a predictive model or a decision tree or maybe perform cluster analysis. Having all of these features in the model may not be desirable. For one, not all the input variables are strong indicators or predictors of the variable we are trying to predict, so they do not need to be considered. Considering these weakly related input variables does not necessarily make our model more accurate, just noisier. When developing a predictive model feature selection reduces the number of input variables to the few that matter most. Reducing the number of input variables is also desirable to reduce the computational cost of modeling and, in some cases, improve the model's performance.

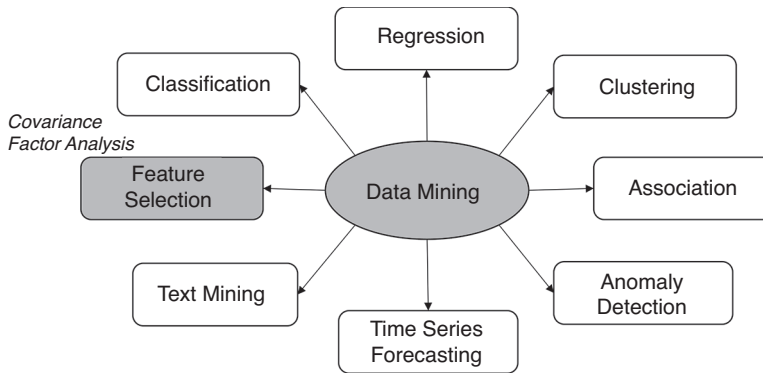


FIGURE 11.1 Data mining algorithms highlight feature selection activities, including Covariance and Factor Analysis.

USING THE COVARIANCE MATRIX

The most common method of reducing the number of variables in a regression model is only to consider those features that are highly correlated to the output variable. Let's use the *Franchises.xlsx* data file found in the *Chapter 11* folder in the *Case Data* depository. In JASP, open the *Franchises.csv* data that relates sales of our franchise stores to contributing features. We see all the possible input variables or features (*SQFT*, *INVENTORY*, *ADVERTISING*, *FAMILIES*, and *STORES*) are highly correlated to *SALES*. We would use all of them in a regression model, with the results of the covariance analysis show in Figure 11.2.

Correlation ▼		Highly correlated					
Pearson's Correlations ▼		SALES	SQFT	INVENTORY	ADVERTISING	FAMILIES	STORES
Variable							
1. SALES	Pearson's r	—					
	p-value	—					
2. SQFT	Pearson's r	0.894	—				
	p-value	< .001	—				
3. INVENTORY	Pearson's r	0.946	0.844	—			
	p-value	< .001	< .001	—			
4. ADVERTISING	Pearson's r	0.914	0.749	0.906	—		
	p-value	< .001	< .001	< .001	—		
5. FAMILIES	Pearson's r	0.954	0.838	0.864	0.795	—	
	p-value	< .001	< .001	< .001	< .001	—	
6. STORES	Pearson's r	-0.912	-0.766	-0.807	-0.841	-0.870	—
	p-value	< .001	< .001	< .001	< .001	< .001	—

FIGURE 11.2 Covariance matrix showing the highly correlated features in our franchise to the outcome variable *SALES*.

Let's consider another data file, *Diamonds Data.csv* found in the same folder. There we see that if we had a database of diamonds where we collected their attributes (*CARAT*, *DEPTH*, *TABLE*, *COLOR*, and *CLARITY*) and we tried to predict *PRICE*, we find that only two of the features are strongly correlated (*CARAT* and *COLOR*) and the rest are weakly correlated, as Figure 11.3 shows. We would tend to build a model with only two and not all five input variables. We reduce the six diamond scoring variables to two significant factors in Case Study 11.2 at the end of the chapter.

Correlation ▾

Pearson's Correlations

Variable		PRICE	CARAT	DEPTH	TABLE	COLOR	CLARITY
1. PRICE	Pearson's r	—					
	p-value	—					
2. CARAT	Pearson's r	0.767	—				
	p-value	< .001	—				
3. DEPTH	Pearson's r	-0.131	-0.197	—			
	p-value	0.492	0.298	—			
4. TABLE	Pearson's r	-0.247	-0.272	0.035	—		
	p-value	0.187	0.146	0.853	—		
5. COLOR	Pearson's r	-0.441	0.013	-0.028	0.084	—	
	p-value	0.015	0.944	0.882	0.659	—	
6. CLARITY	Pearson's r	-0.200	0.046	0.150	0.083	0.179	—
	p-value	0.289	0.808	0.430	0.664	0.343	—

FIGURE 11.3 Covariance matrix showing the highly correlated features to the outcome variable.

The first step in most regression models is finding features strongly related to the outcome variable. The covariance analysis is the most popular tool for that purpose.

FACTOR ANALYSIS

Factor Analysis is another popular, albeit more complex, dimensionality reduction technique. It can successfully reduce the number of features by combining those highly related to each other into a few factors before we use them in a regression model, for example. We use Factor Analysis (FA) to reduce a large number of variables into a few factors, not to select a few out of many, which is actual dimensionality reduction. Most FA techniques extract the maximum common variance from all variables and put them into a standard score. We can use this score for further analysis as an index of all variables. FA needs to make several assumptions before being employed: that there is a linear relationship between the features, that there is little multicollinearity, that it includes only relevant variables in the analysis, and that there is an actual correlation between variables and factors. Several methods are available for FA, but Principal Component Analysis (PCA) is the most commonly used.

FA itself is considered an unsupervised machine learning algorithm. Then, once the factors are discovered, using them in a regression model is considered dimensionality reduction. This algorithm creates factors from the observed variables to represent the common variance, i.e., variance due to correlation among the observed variables.

The overall objective of FA can be broken down into four objectives:

- to understand how many factors are needed to explain common themes among a given set of variables
- to determine the extent to which each variable in the data set is associated with a common theme or factor
- to provide an interpretation of the common factors in the data set
- to determine the degree to which each observed data point represents each theme or factor.

WHEN TO USE FACTOR ANALYSIS

Consider several issues when determining when to use particular statistical methods to get the most insight from your data. There are three main principal uses of FA. If your goal aligns with any of these uses, then you should choose FA as your analysis method of choice:

- **Exploratory FA** should be used when there is a need to develop a hypothesis about a relationship between variables. Given a set of variables, what underlying dimensions (factors) account for the patterns of collinearity among the variables? Example: Given multiple items of information gathered on applicants applying for admission to a university, how many independent factors are being measured by these items?
- **Confirmatory FA** should be used to test a hypothesis about the relationship between variables. Given a theory with four concepts that purport to explain some behavior, do multiple measures of the behavior reduce to these four factors? Example: Given a theory that attributes delinquency to four independent factors, can multiple measures on delinquents be explained by these four factors?
- **Construct Validity** should be used to test the degree to which a survey measures what it is intended to measure.

FIRST STEP IN FA – CORRELATION

To apply FA, we must ensure that our data is suitable. The most straightforward approach would be to look at the correlation matrix of the features and identify groups of intercorrelated variables. If there are some correlated features with a correlation degree of more than 0.3 between them, perhaps it would be interesting to use FA. Groups of features highly intercorrelated will be merged into one latent variable, called a *factor*. One limitation of this approach is that as the number of variables in the data set increases, it becomes practically impossible to keep track of the relationships among variables.

The sample size should be large enough to yield reliable estimates of correlations among the variables. Let's say you are conducting a survey. In a survey, k survey items or questions are asked, and N responses are collected. Ideally, there should be a large ratio of N/k (responses/survey items), possibly 20:1 or more. If there are 20 question items in the survey, there would be at least 400 responses for FA to yield a good response. FA can still be reasonably done with a minimum of a 7:1 ratio of survey items being tracked to survey responses being collected.

FA FOR EXPLORATORY ANALYSIS

There are two basic approaches to FA: PCA and Common FA. Overall, FA involves techniques to help produce a smaller number of linear combinations of variables so that the reduced variables account for and explain most of the variance in the correlation matrix pattern. PCA is the most popular, and we explore it in this chapter. PCA is an approach to FA that considers the total variance in the data. Unlike Common FA, PCA transforms the original variables into a smaller set of linear combinations. The diagonal of the correlation matrix consists of unities, and the total variance is brought into the Factor Matrix, which is the matrix that contains

the computed factor loadings of all the variables on all the extracted factors. The term “factor loadings” are the simple correlation between the factors and the variables. PCA is recommended when the analyst’s primary concern is to determine the minimum number of factors that will account for the maximum variance in the data. While conducting PCA, the analyst should know significant terms such as standard deviations and eigenvalues, which refer to the total variance explained by each factor. The standard deviation measures the variability of the data.

SELECTING THE NUMBER OF FACTORS - THE SCREE PLOT

A scree plot analysis is the most common method for selecting the number of factors. It is a plot of the eigenvalues of the resulting factors. It is a visual tool to select the optimum number of factors to reduce the data set. Use the plot of the eigenvalues against the number of factors and determine how many factors will produce eigenvalues greater than but closest to one. The example below shows an example of a scree plot (Figure 11.5). Most PCA tools will produce the scree plot to determine the optimum number of factors as part of the PCA analysis.

Correlation							
Pearson's Correlations							
Variable		Waiting Time	Personnel	Food temp	Freshness	Cleanliness	Food Taste
1. Waiting Time	Pearson's r	—					
	p-value	—					
2. Personnel	Pearson's r	0.667	—				
	p-value	< .001	—				
3. Food temp	Pearson's r	0.408	0.612	—			
	p-value	< .001	< .001	—			
4. Freshness	Pearson's r	0.612	0.408	0.250	—		
	p-value	< .001	< .001	0.025	—		
5. Cleanliness	Pearson's r	-0.218	-0.327	-0.535	0.535	—	
	p-value	0.052	0.003	< .001	< .001	—	
6. Food Taste	Pearson's r	0.000	-0.456	-0.559	0.559	0.896	—
	p-value	1.000	< .001	< .001	< .001	< .001	—

FIGURE 11.4 Correlation of the six variables showing possible highly correlated variables which might combine into two factors.

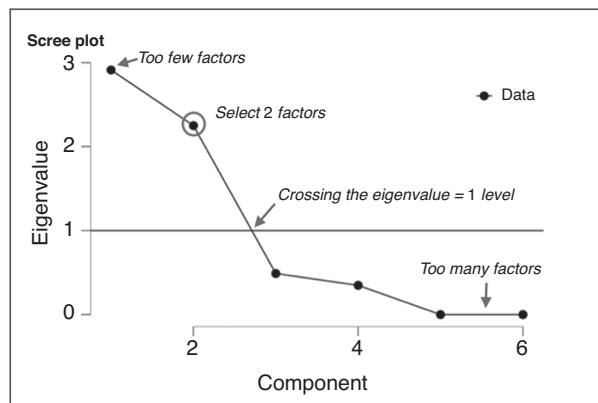


FIGURE 11.5 The scree plot of a PCA FA of the six restaurant feedback variables. The annotations are not part of the original JASP plot.

EXAMPLE 11.1: RESTAURANT FEEDBACK

The best way to understand FA is to tackle a real-life problem. Consider that you are the owner of a restaurant. You are very interested in knowing what your customers think about your restaurant's food and service. You survey your customers at the end of each meal, asking for their feedback on the food and the service. Was the wait too long? Did your server treat you well? Was the food at the right temperature? Did it taste good? (You could ask other questions of that nature.) You have six survey questions, with six resulting scores captured in six variables. You collect over 80 responses. (The ratio of capital N/k is over 13 to 1, so we are in good shape to apply PCA.) The data set for this exercise, *Restaurant Survey.csv*, may be found in the *Chapter 11* folder of the *Case Data* depository that accompanies this book.

You are concerned that keeping track of six dimensions of feedback to improve your service is too difficult, and you wonder if there are ways of combining the six dimensions into just a few scores that would allow you and your staff to keep track of the practical improvements. The framed analytical question then becomes

What factors can be derived from the six-question survey responses to reduce the tracking to a few meaningful scores?

If we perform a correlation analysis of the six variables, we see that two groups of variables are highly correlated. One group comprises *WAITING TIME*, *PERSONNEL*, *FOOD TEMP*, and *FRESHNESS*. The other comprises *PERSONNEL*, *FOOD TEMP*, *FRESHNESS*, *CLEANLINESS*, and *FOOD TASTE*. As you will see, they will become the elements in some combination of the two factors representing customer sentiment. Figure 11.4 shows the correlation matrix with these very distinct high correlations. Note that not all variables are correlated. *FOOD TASTE* is not related at all to *WAITING TIME*, for example. The same is true of *CLEANLINESS* and *WAITING TIME*.

The JASP program has a PCA analysis tool. If you were to load the *Restaurant Survey.csv* data file into JASP and perform the PCA analysis, you could display the scree plot, as shown in Figure 11.5. We show where the plot crosses the eigenvalue of one. The number of factors with eigenvalues above 1, but closest to the crossing point gives us the optimum set of factors we should use. In this case, it is 2 factors, which we suspected, given the correlation pattern observed from the correlation matrix.

The solution to the PCA analysis, essentially the resulting model, is the factor loading matrix. It is how the factor variables must be combined to create the two factors. For any data role in the data table, we must multiply the value in each of the six variables by the loading factors to obtain the value of each factor. Figure 11.6 shows the factor loadings in this case.

Factor Loadings			
	Factor 1	Factor 2	Uniqueness
Food Taste	0.996		-0.004
Cleanliness	0.910		0.150
Freshness	0.651	0.776	-0.026
Food temp	-0.460	0.610	0.417
Personnel		0.807	0.256
Waiting Time		0.781	0.390

Note. Applied rotation method is varimax.

FIGURE 11.6 The resulting two factors from a PCA analysis for the restaurant data set show the six variables' factor loadings.

We can characterize and give each factor a name by observing the most prominent variables in each factor (highest loadings). In our case, Factor 1 appears to be about food quality and Factor 2 is about service.

The JASP program also provides various forms of creating the factors. Please select the most popular technique where the final factors are computed by rotating the factors until they achieve the lowest correlation between them, hopefully completely uncorrelated or orthogonal. The Varimax algorithm is most typically used for this purpose. The orthogonal rotation with Varimax yields completely uncorrelated factors, as seen in the correlation matrix shown in Figure 11.7.

Factor Correlations		
	Factor 1	Factor 2
Factor 1	1.000	0.000
Factor 2	0.000	1.000

FIGURE 11.7 The correlation between the resulting two-factor model shows no correlation between the final two factors.

Suppose we asked for three factors; what would they look like? Figure 11.8 shows the factor loading matrix for three factors. You might name these three factors food quality, temperature, and service. It is your call as the restaurant owner to decide whether these three factors are better than having only the other two. It is a business decision.

Monitoring two or three performance indicators is probably better than monitoring all six.

Component Loadings ▼				
	RC1	RC2	RC3	Uniqueness
Cleanliness	0.952			0.044
Food Taste	0.847			0.026
Freshness	0.811			0.001
Waiting Time		1.135		0.034
Personnel		0.574		0.185
Food temp			1.052	0.059

FIGURE 11.8 The resulting factor loading matrix from a PCA analysis with three factors for the restaurant feedback survey data set.

FACTOR INTERPRETATION

Once we have the FA model, we must interpret the factors. That is the point of dimensionality reduction with FA: having an interpretable data set with fewer features. Factor loading is the weight given to each feature being subsumed into the new and indicates the contribution of each feature to that factor. Most FA tools usually provide such a matrix. It tells you which variables are aggregated into which factors and the percent contribution of each. By looking at the aggregation of the variables, we can get a sense of what each of the factors represents. We can, at that moment, give the factors representative names corresponding to the aggregated variables. In the restaurant survey, the factor loadings range from -1 to 1 and can be interpreted as the correlation of the variable with the factor.

SUMMARY ACTIVITIES TO PERFORM A FACTOR ANALYSIS

1. Prepare the data set.
 - a. Encode binary and categorical variables into continuous numeric variables.
 - b. Remove outliers.
 - c. Split data into testing and training data sets.
2. Decide if the data set is suitable for FA.
 - a. Is the ratio of rows/features $> 7:1$ at least?
 - b. Are some of the variables correlated ($>.3$)?
3. Train the FA model.
 - a. Use a scree plot of eigenvalues to select an appropriate number of factors.
 - b. Perform rotational analysis with the Varimax rotation.
4. Interpret the factors (name the resulting factors).
5. Employ the factors in classification or regression (use the factor loading matrix).
6. Test the model for overfitting.

CASE STUDY 11.1: VARIABLE REDUCTION WITH THE SFO SURVEY DATA SET

The SFO marketing executive would like to create a dashboard to track progress on initiatives to improve the airport customer experience. Our current survey is comprehensive and granular, and it helps track not only the overall satisfaction score (*Q7ALL* and *NETPRO*), but also 14 other essential parameters (*Q1ART* through *Q7RENTAL*). He would like to reduce these 14 parameters to just a few. He has determined that these 14 are probably too many to use as KPIs to focus his team on improvement initiatives. Many of these variables measure very similar things (*Q7PARKING*, *Q7RENTAL*, and *Q7AIRTRAIN*, for example.) Perhaps there is a way to aggregate these variables into a meaningful few.

The issues we will address using FA are as follows:

What good set of aggregated variables to combine the 14 customer scores on airport satisfaction makes sense?

Create a linear regression model to predict Q7ALL and NETPRO scores from the newly created factors.

We will generate a FA model to create the reduced set of variables. Use all the Q7 variables (minus *Q7ALL*) from the 2018 data set. Make sure to remove the 0 and 6 scores. Once created, name the factors by using what variables are being aggregated.

As an additional exercise, perform the following:

Once reduced, apply the model to previous years and create trendlines for each factor.

SOLUTION IN R

Use the *SFO 2018 Survey Data.xlsx* data set found in the *SFO Survey Data* folder in the *Case Data* depository. To ensure that the variables are correctly prepared, you should follow the data cleaning steps in Chapter 4. Once the data set is well-prepared, import it into Jamovi.

To create reasonable aggregated variables to combine the 14 customer scores on airport satisfaction, we can run the Exploratory FA on all *Q7* variables except *Q7ALL*, as shown in Figure 11.9.

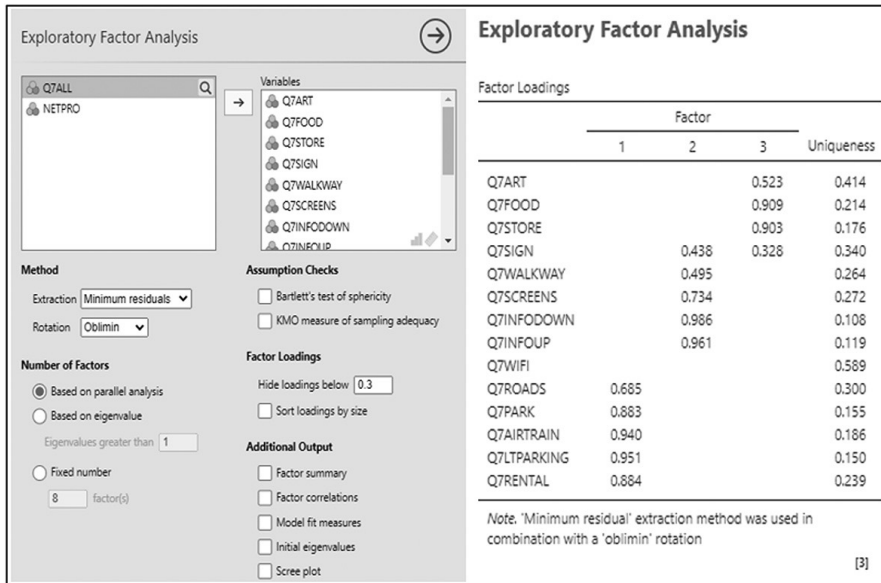


FIGURE 11.9 Steps and output of FA in Jamovi.

Once we have the result from FA, create new aggregated variables using the Edit function within Jamovi, as shown in Figure 11.10.

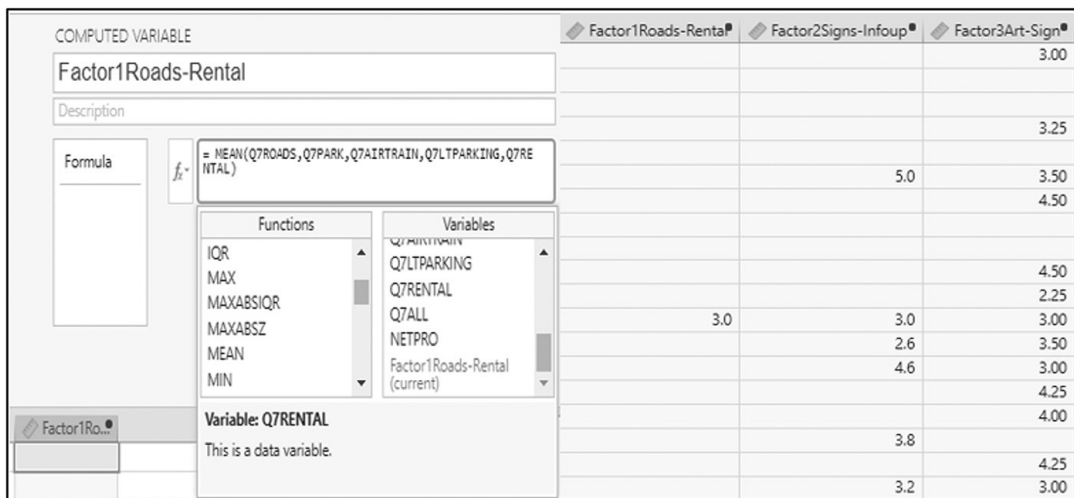


FIGURE 11.10 Creating the mean on the new aggregated variables.

Create a linear regression model to predict *Q7ALL* and *NETPRO* scores from the newly created factors. See Figures 11.11(a) and (b).

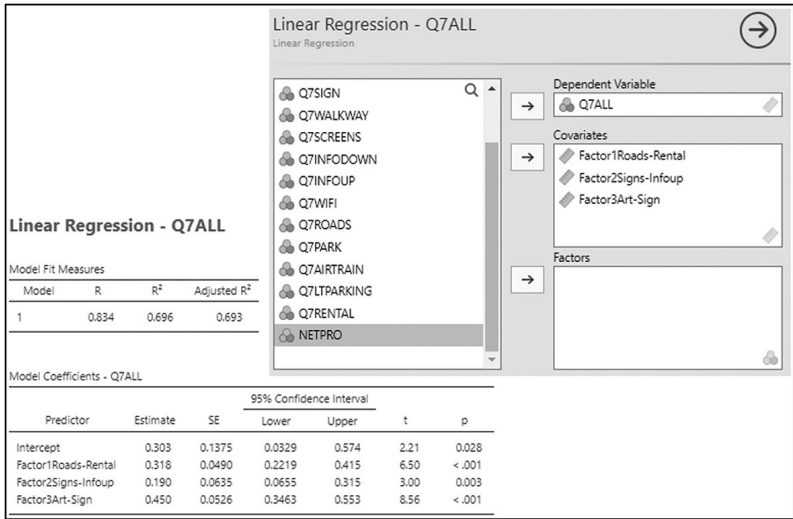


FIGURE 11.11(a) Linear regression model on Q7ALL scores from the newly created factors.

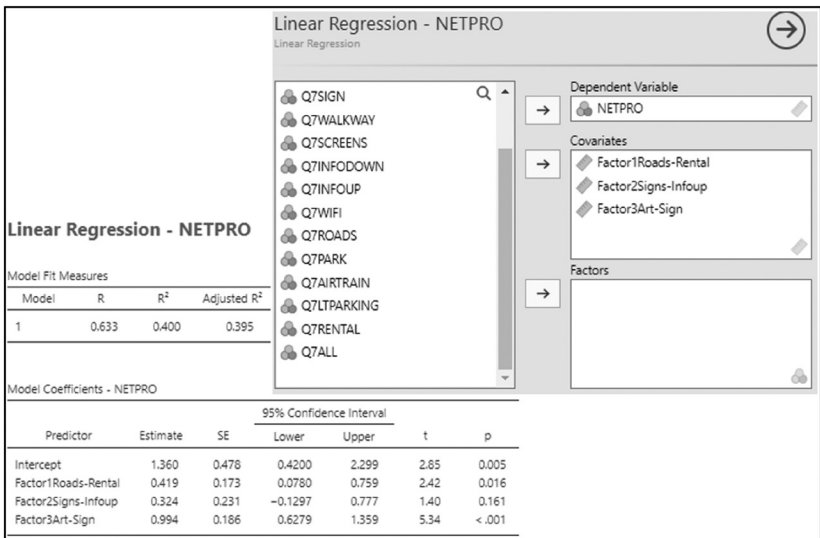


FIGURE 11.11(b) Linear regression model on NETPRO scores from the newly created factors.

SOLUTION IN PYTHON

To ensure the variables are correctly prepared, you should follow the data cleaning steps in Chapter 4. Once the data set is well-prepared, import it into Jamovi. To create reasonable aggregated variables to combine the 14 customer scores on airport satisfaction, we can run the PCA on all Q7 variables except Q7ALL, as shown in Figure 11.12.

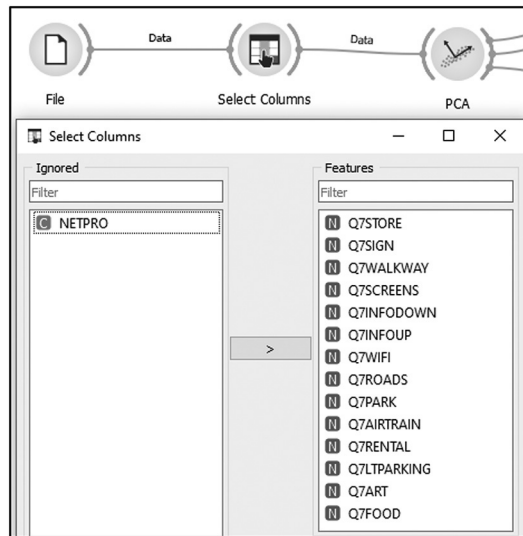


FIGURE 11.12 Steps to create PCA in Orange3.

We can decide how many principal components to go with by selecting the first few components that explain 80% of variability, as shown in Figure 11.13. In the *PCA* widget, you can observe the proportion of explained variance in the diagram.

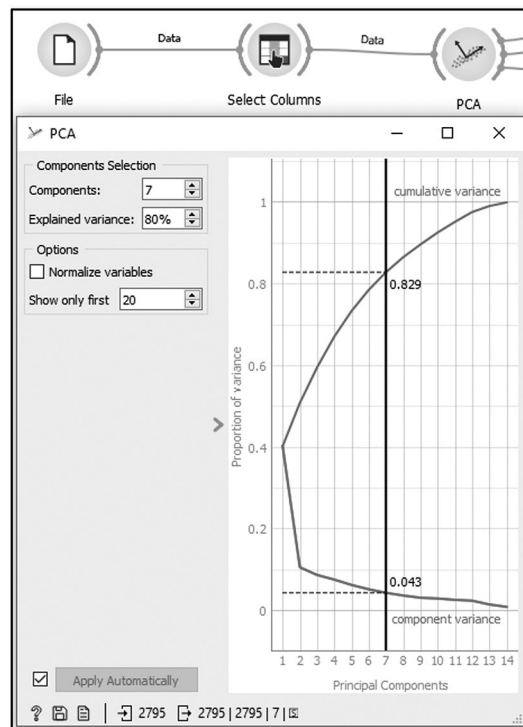


FIGURE 11.13 Adjust explained variable of PCA.

Transform the data into a data table to obtain the result like that shown in Figure 11.14.

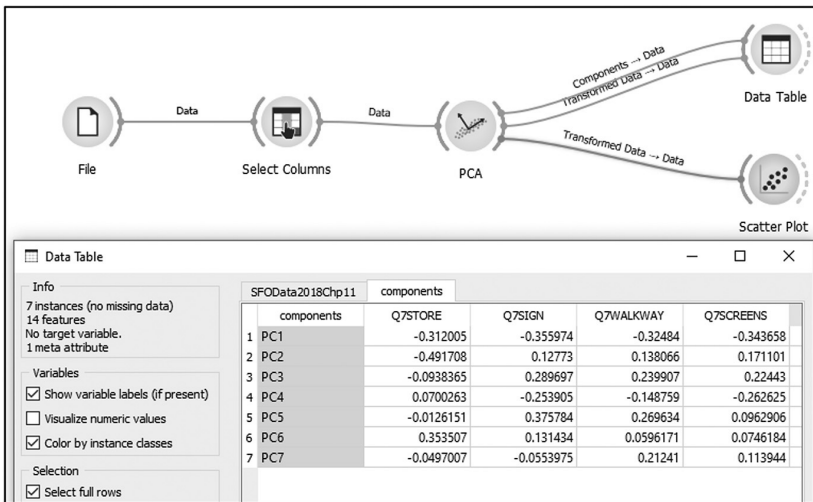


FIGURE 11.14 Create the output of PCA.

You can plot the transformed data in a scatter plot, as shown in Figure 11.15.

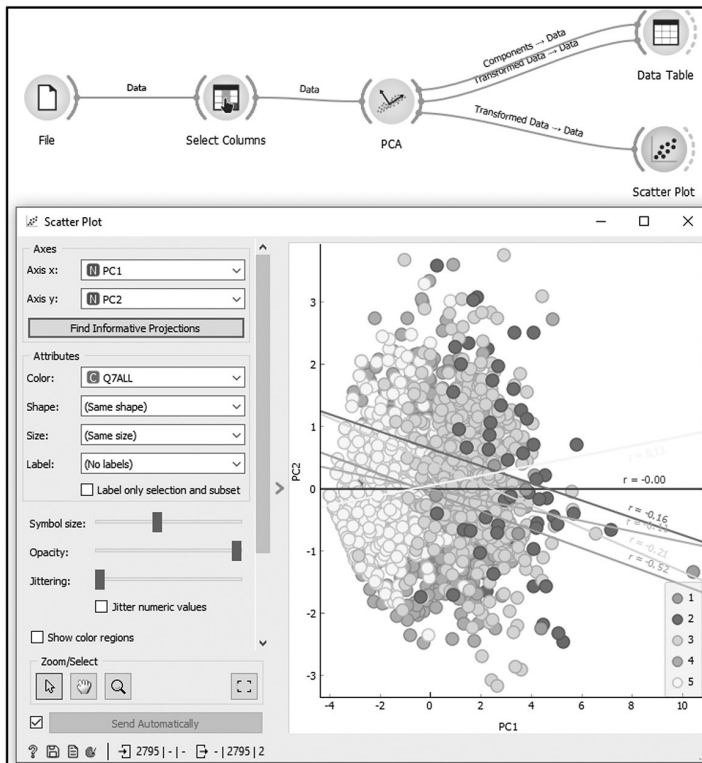


FIGURE 11.15 Observe the output of the PCA by using a scatter plot.

CASE STUDY 11.2: HUNTING DIAMONDS

A young man is keen on getting a diamond ring for the person he intends to propose to. He wants to get a good deal. Since he is also a data analyst, he decides to track the information he gets from jewelers during his diamond hunting trips. He collects an extensive data set (*Diamonds Data.xlsx*). He uses the gemological industry set of five variables in rating diamonds (carat, clarity, color, depth, and table.) Check out the definitions of these variables in the data dictionary tab in the Excel file.

To this data set, he adds perhaps the most critical variable of interest to him: price. After many trips to jewelry stores, he compiles quite a database and is ready to generate a little formula he can use to compare choices and get the best deal. However, he realizes having six variables is probably too many for quick decisions (this one, not that one...) as he continues his hunt for the best ring. Given his experience in data mining, he decides to use FA to reduce the six variables to maybe two or three.

What is an optimum set of factors that will reduce the six diamond features into two or three aggregate factors?

What factors can be used to describe a diamond purchase from the features of a diamond to make a purchasing decision?

Note that the data set *Diamonds Data.xlsx* contains data on 30 diamonds. Given that there are six features we want to reduce to a few factors, does this database pass the rule that there should be more rows than seven times the number of variables ($N/k > 7$) to be aggregated? It would help if you computed a scree plot on the data set and the variables to determine an optimum number of factors to create. Once the model is created, give each factor an adequate name according to its aggregated variables.

SOLUTION IN R

Once you have the data set ready, import it into JASP. Select *Correlation* under *Regression* and choose all variables into the variable selection box, as shown in Figure 11.16.

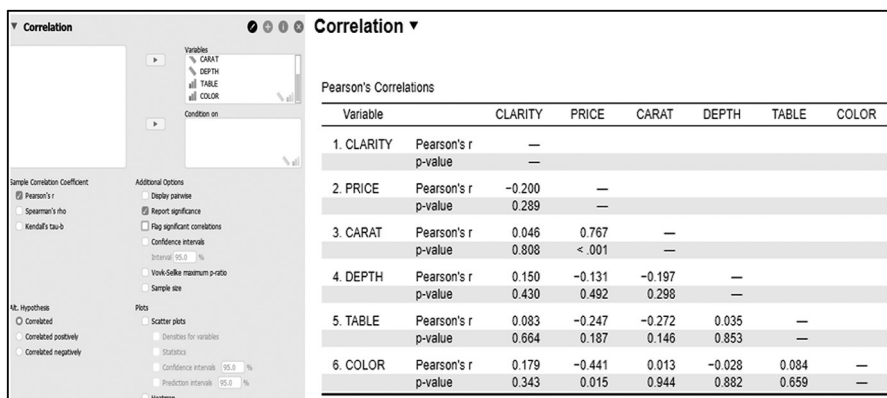


FIGURE 11.16 Steps to create the correlation matrix.

To create a screen plot, go to *Factor* and select *PCA*. See Figures 11.17(a) and (b).

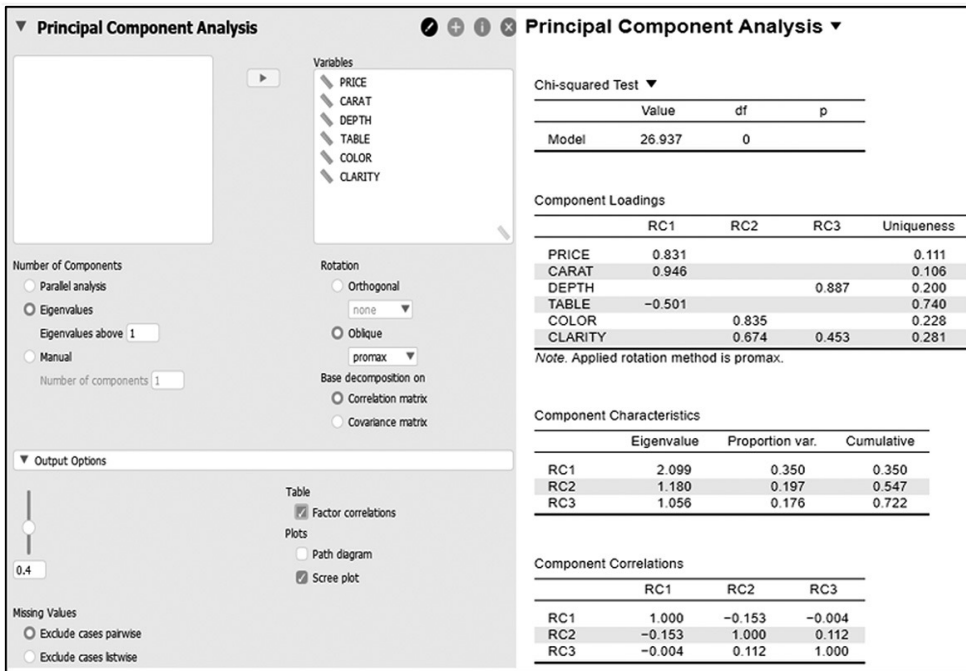


FIGURE 11.17(a) Steps and output of PCA in JASP.

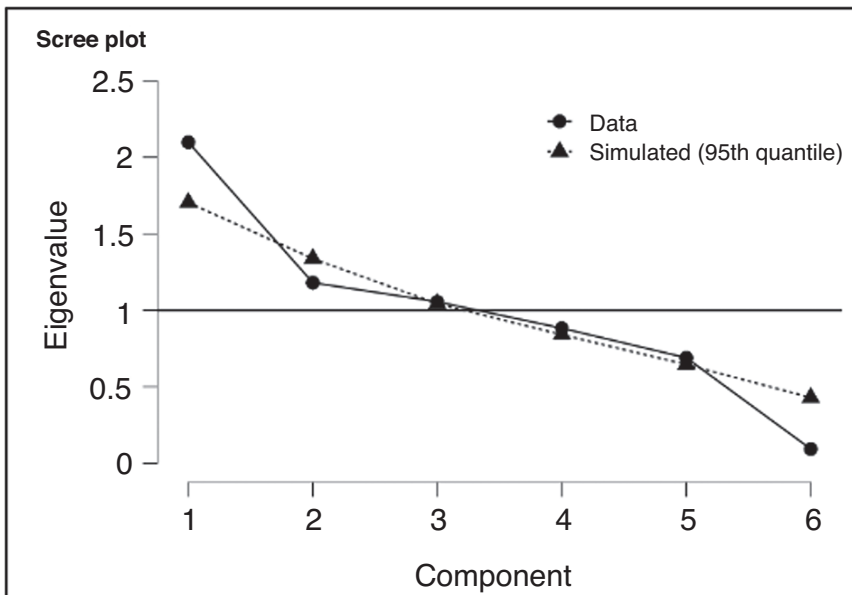


FIGURE 11.17(b) Steps and output of PCA in JASP.

SOLUTION IN PYTHON

Once you have the data set ready, import it into Orange3. To create a correlation table, you can connect to a correlation widget, as shown in Figure 11.18.

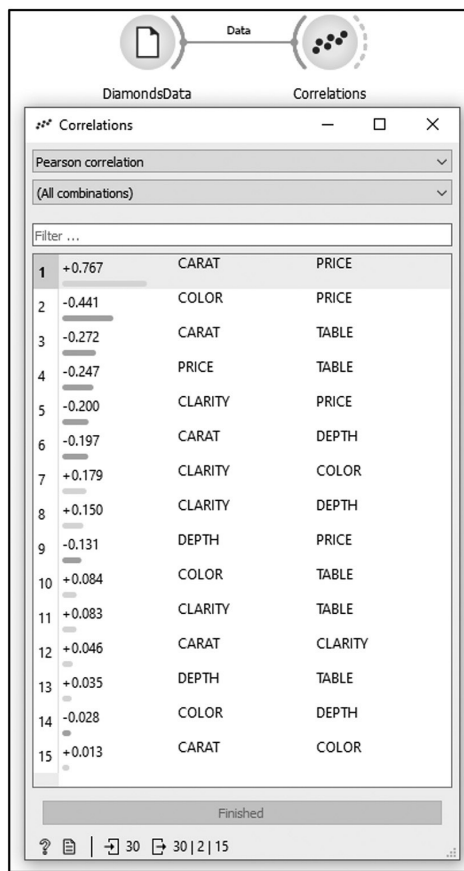


FIGURE 11.18 Steps and output of correlation matrix in Orange3.

We can run the PCA on all variables. Select three components in PCA and observe components by connecting them to a data table like Figure 11.19.

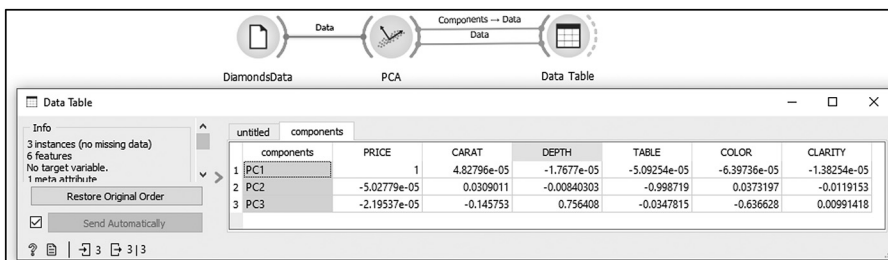
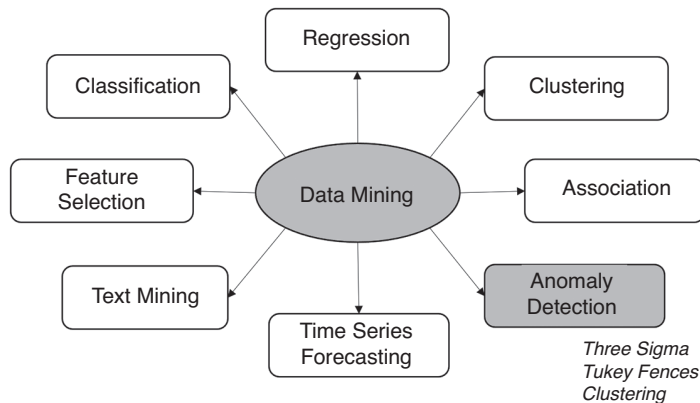


FIGURE 11.19 Steps and output of PCA.

ANOMALY DETECTION



A *nomaly detection* in data analysis, also referred to as *outlier detection*, generally involves the identification of items, events, or observations in the data set which deviate significantly from most of the data and do not conform to a well-defined notion of normal behavior. Such examples may arouse suspicions of being generated by a different mechanism or appear inconsistent with the remainder of that data set.

WHAT IS AN ANOMALY?

There is a difference in contextual meaning between anomalies and outliers. Consider a population of items (such as things, people, or organizations). An *anomaly* is an item in the population whose attributes set it apart from the bulk of the population (something outside the norm). It is a qualitative measure imposed by the analyst, or a judgment call. Similar to the process we use in creating bins of a numerical variable to create categories for categorical analysis, we likewise separate the population into “normal” members and “anomalous” members with criteria to match what is considered anomalous.

Typically, the anomalous members of the population are in the minority, with the regular members being the majority, the many. Say you have a group of tourists visiting a city and their average age is 25, but there are a few tourists in the group who are over 65. Those who are over 65 would be considered anomalous. It may be in reverse: a tour group is primarily adults, but one couple brought their teenage granddaughter, who would be the anomaly.

It is a matter of scarcity: a few are the anomalies; the many are the normal. For example, a physical feature is the eruption of the Old Faithful geyser at Yellowstone National Park in the United States. If we measure the length of eruptions and waiting times between eruptions for some eruptions (our population), we find that the geyser has two modes of erupting. Sometimes it blows much steam, and we must wait a long time until it erupts again (we might call these “long eruptions”), and sometimes it erupts for a short period (“short eruptions”). It is a bimodal feature, as seen in Figure 12.1.

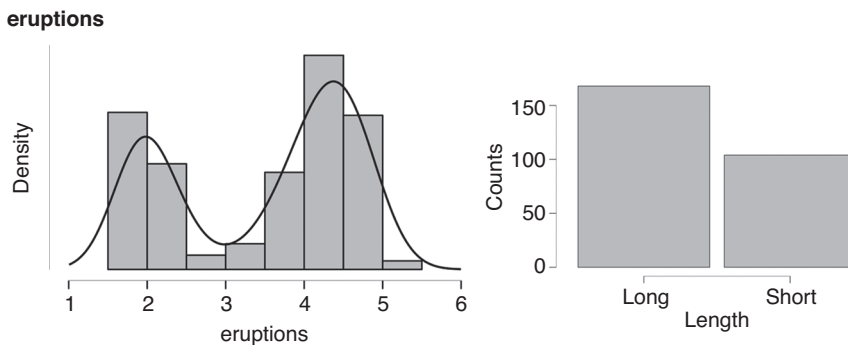


FIGURE 12.1 Old Faithful eruption lengths show bimodal operation.

We see that the average eruption time is 3.5 minutes. If we use that as the criteria for binning the eruption times into two bins, short and long, we find that for this set of 272 consecutive observations, the ratio is 60% long to 40% short eruptions. Could you say short eruptions are an anomaly? Probably not; they are too numerous. A judgment call is necessary to determine whether there are any anomalies of the system operating in two standard modes, rather than a normal mode and an anomalous mode.

WHAT IS AN OUTLIER?

We see then that identifying anomalies requires a judgment call on the boundary between normal and anomalous and how many not-normal members we find to identify an anomaly.

We will explore in this chapter several of the most common definitions or computations of outliers. Those judgment calls are based on specific attributes for identification, which require a metric. It can be based on any type of attribute. We can count the frequency of appearance of a categorical variable as the criteria. Only having five red cars in a car lot of 100 cars probably makes red cars anomalous. In the previous example, having one teenager in a group of 25 adult tourists makes the teenager anomalous. Still, the decision was based on binning the age attribute of the tourist population, a numerical variable.

We translate the fuzzy question of “what is anomalous?” into a framed question of what members of the population are anomalous based on a metric imposed on one of the attributes

(variables) in our data set. We are looking for population members whose values are outside the norm based on a designated variable. We call these “outside the normal” variable values *outliers*.

We define an anomaly as the qualitative criteria for what is expected and what is not. We analyze for outliers as a way to convert the definition of an anomaly into a framed analytical question, something we can measure. The next section shows how outliers may be computed for various variables and situations.

THE CASE STUDIES FOR THE EXERCISES IN ANOMALY DETECTION

We use the case of data mining airline delays to demonstrate the various ways to compute outliers and search for anomalies. Let’s say we work for a major US airline like Delta Airlines. They have been receiving many complaints of long flight delays. Delta wants to know how frequent these long flight delays are, what causes them, how lengthy are “long” flight delays (defined as *Very Late*), and whether they are more frequent for AM or PM flights on certain days of the week or certain months of the year.

The US government Bureau of Transportation compiles abundant airline travel data. Flight statistics can be found at <https://www.transtats.bts.gov/ontime/Arrivals.aspx>. As a case study, we will track flight data for Delta Airlines arriving at Chicago O’Hare International airport throughout 2021. The data file *Detailed_Statistics_Arrivals Delta Chicago 2021.xls* may be found in the *Chapter 12* file folder in the *Case Data* depository. We will look for outliers in this data file in the exercises below.

ANOMALY DETECTION BY STANDARDIZATION – A SINGLE NUMERICAL VARIABLE

An easy way to define huge numbers (or tiny numbers) in a numerical attribute is to see how far away from the mean each value is. We will assume that the distribution is reasonably normal (see the curve shown in Figure 12.2). We will define any number that lies on the number line more than three sigmas away from the mean in either direction as an outlier. Since the six-sigma limits (± 3 sigma away from the mean) represents 99.95% of all values, we are talking about very few values.

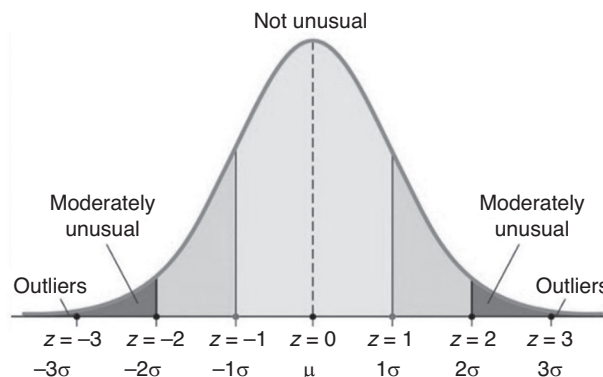


FIGURE 12.2 Outliers defined by normalization; data outside the three-sigma limits.

To label any value as an outlier, we standardize it. That means we compute its distance from the mean of the attribute in sigmas. That is called the *Z-score* of the number.

$$\text{Z-Score} = \frac{\text{Number} - \text{Mean}}{\text{Sigma}}$$

Any number whose Z-score is more than +3 or less than -3 will be considered an outlier. The higher the Z-score, the stronger our justification for labeling it an outlier.

EXERCISE 12.1 – OUTLIERS IN THE AIRLINE DELAYS DATA SET – Z-SCORE

Let's see how many of Delta's flights flying into Chicago in 2021 experienced unusually long delays (arrived *Very Late*). Let's use the data set *Detailed_Statistics_Arrivals Delta Chicago 2021.xls*. The raw data downloaded for the Bureau of Transportation (Bureau 1997) website has been transformed into a flat file and several columns have been added (such as day of the week, month, and AM/PM). The variable of interest is *Arrival Delay*. We compute the Z-score for the delay variable in a separate column by using the STANDARDIZE Excel function. We can bin the Z-score into five bins: *Very Late* (>3), *Late* (less than 3, but above 0), *On-time* (0 delays), *Early* (less than 0 but more than -3), and *Very Early* (<-3).

By summarizing using a pivot table on the *Delay* categorical variable, we see very few *Very Late* flights (139 flights or 2% of all flights that year). By our binning criteria, though, *Late* flights range from 1 minute late to the 3-sigma limit of 147 minutes (over 2 hours) and account for over 33% of all flights. Delta is doing very well because on-time and early arrivals account for almost 90% of all flights in 2021. Interestingly, the airline industry classifies any flight over 45 minutes delayed as being late. We might want to make the 0-3 sigma range into several lateness ranges. However, to define *Very Late* flights, our 3-sigma, 147-minute delay boundary is reasonable. Notice that there are no *Very Early* flights. Figure 12.3 shows the results of the pivot table analysis.

Level of Delay	Count of Flights	Percent of all Flights
Early	4209	63%
Late	2201	33%
On Time	172	3%
Very Late	139	2%
Grand Total	6721	100%

FIGURE 12.3 Analysis of outliers (*Very Late* flights) for Delta flights into Chicago in 2021.

ANOMALY DETECTION BY QUARTILES – TUKEY FENCES – WITH A SINGLE VARIABLE

A different way to graphically display the spread of a numeric variable is to use a box plot diagram. Quartiles are used to show the majority of the data points (range of Q1 to Q3) as a box with the median (Q2) as a line in the box. From Q1, the lower quartile (25% of the data points) to the minimum data point, we show a whisker line to a vertical stop called a *fence* (sometimes, these fences are not shown). The same holds for the data points above Q3 to the maximum value. With a box plot, we get a sense of the range (max to min) as we see the extent of the variable from fence to fence. We also see where the bulk of the data points lie (50%), as shown by the Q3-Q1

box, with a sense of the central tendency being the median or Q2. Figure 12.4 shows the classic form of the box plot. We can easily compare the distributions of comparable variables in a data set by showing their box plots side by side.

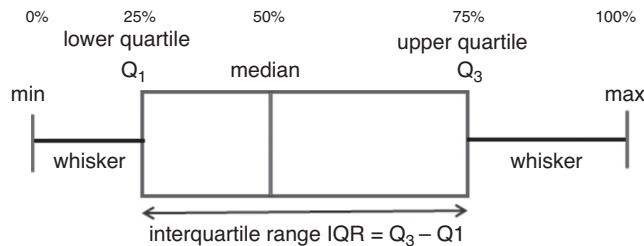


FIGURE 12.4 Classic depiction of a box plot and its components. Notice the ends of the whiskers (the fences are set at the maximum and minimum values of the range of values of the numeric variable).

John Tukey, a renowned statistician, defined a limit for outliers using quartiles as any value below Q_1 by 1.5 times the Interquartile Range ($IQR = Q_3 - Q_1$). The same is true for significant outliers, defined as values 1.5 times the IQR above Q_3 . Tukey's box plots have the fences at these two levels rather than at the maximum and minimum values and he also displays any data points outside these limits as separate points. We now call these two new maximum and minimum levels of normality *Tukey fences*. Figure 12.5 shows a box plot with Tukey fences displayed. There may or may not be any "outliers" as defined by Tukey, and we would see this in the redefined box plot. You can read about it in his original paper (Tukey 1977) or a more accessible explanation presented elsewhere (Hoaglin 2003).

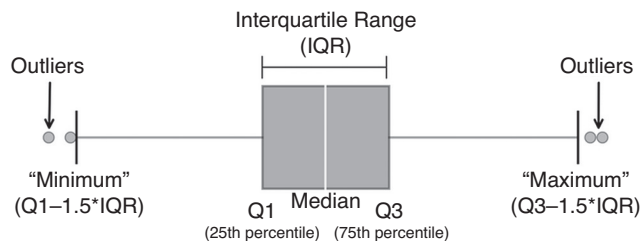


FIGURE 12.5 Definition of a box plot with the fences set at the Tukey-defined limits of $(Q_3 + 1.5 * IQR)$ and $(Q_1 - 1.5 * IQR)$ to separate and highlight the outliers.

To determine whether a particular data point is an outlier using the Tukey definition, we must compute the quartiles (Q_1 and Q_2) and the interquartile range ($IQR = Q_3 - Q_1$). Then we compute the upper Tukey fence value ($Q_3 + 1.5 * IQR$) and lower Tukey fence value ($Q_1 - 1.5 * IQR$) and use these to determine whether the data points in our variable are outliers.

COMPARING Z-SCORES AND TUKEY FENCES

When we compare the popular computation by Z-score and the Tukey definition, we see they are not that far apart. Figure 12.6 shows the comparison of Tukey fences and three-sigma limits. The Tukey fences encompass 99.3% of the data points, and the three-sigma limit encompasses 99.95% of the distribution. That could be a big difference depending on the spread of the data.

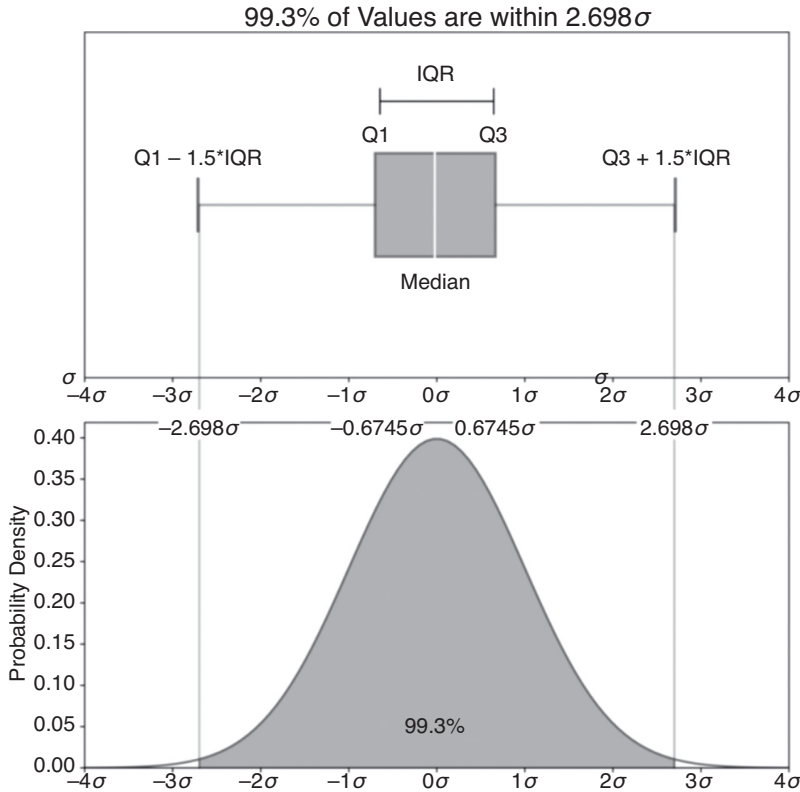


FIGURE 12.6 Comparing the definitions of outliers by box plots and standard curve definitions.

In the following exercise, we compare computing outliers both ways and show that they could produce significantly different results.

EXERCISE 12.2 – OUTLIERS IN THE AIRLINE DELAYS DATA SET – TUKEY FENCES

Using Tukey’s outlier definition, let’s see how many of Delta’s flights into Chicago in 2021 experienced unusually long flight delays (arrived *Very Late*). Once again, let’s use the data set *Detailed_Statistics_Arrivals Delta Chicago 2021.xls*. The raw data downloaded for the Bureau of Transportation website has been transformed into a flat file and several columns have been added (such as day of the week, month, and AM/PM). Again, the variable of interest is *Arrival Delay*.

First, let’s compare all the quartiles, the IQR and the Tukey outlier limits. Figure 12.7 shows the results of the computation.

In a separate column and using the definition of the upper and lower Tukey limits, bin the data points as it is an outlier or not. We can bin the data into five bins: *Very Late* (above the upper Tukey limit), *Late* (below the upper limit but above 0), *On-time* (0 delays), *Early* (less than 0 but above the lower Tukey limit), and *Very Early* (below the lower limit).

Tukey Fence Computation	Delay in Minutes
Max	914
Min	-47
Q1	-15
Q2 (Median)	-6
Q3	7
IQR	22
1.5*IQR	33
Upper Tukey Fence	40
Lower Tukey Fence	-48

FIGURE 12.7 Results of computing the quartiles, the IQR, and the Tukey outlier limits for the Delta flight delay data.

By summarizing using a pivot table on the *Delay* categorical variable, see very few *Very Late* flights (614 flights or 9% of all flights that year), as seen in Figure 12.8. By our binning criteria, though, *Late* flights range from 1 minute late to the upper Tukey limit of 40 minutes and account for over 26 % of all flights. Again, by this definition, Delta is still doing very well in that *On-Time* and *Early* arrivals account for almost 90% of all flights in 2021. Interestingly, the upper Tukey limit of 40 minutes more closely aligns with the airline industry classification of any delay over 45 minutes being late.

Arrival	Number of Flights	Percent of Flights	Average of Arrival Delay (Minutes)
Early	4209	63%	-13
Late	1726	26%	13
On-time	172	3%	0
Very Late	614	9%	113
Grand Total	6721	100%	5

FIGURE 12.8 Analysis of outliers (*Very Late* flights) for Delta flights into Chicago in 2021 using the Tukey limit definition of outliers.

When we compare the outlier computation using both techniques (the tables in Figures 12.3 and 12.7), we see that the Z-score technique classifies fewer data points as outliers (2% versus 9%, 139 versus 614 data points). Still, the Tukey limits are more closely aligned with the industry definition of lateness.

ANOMALY DETECTION BY CATEGORY – A SINGLE VARIABLE

What if you want to identify population members by a categorical variable? What if we do not have a numeric variable to which to apply a Z-score or Tukey limits computation? One approach is to do what is frequently done for categorical variables: tabulate! Look for the frequency of appearance of the various categories and seek those that are unusually infrequent. The issue here is defining what it means to be “infrequent.” Unfortunately, as

we did with the numeric variables shown earlier, we do not have a standard definition. It comes down to a judgment call. Is less than 1% frequency an outlier, or should we cut it off at 2% or 5%? The analyst must use context and the business question to be answered as a guide for what to label as an outlier. The case examined in the exercise below exemplifies this point.

EXERCISE 12.3 – OUTLIERS IN THE AIRLINE DELAYS DATA SET – CATEGORICAL

Once again, let’s use the data set *Detailed_Statistics_Arrivals Delta Chicago 2021.xls*. Let’s see which originating airports are outliers. In this case, looking at all the flights Delta operated into Chicago in 2021, we have to determine which were the most and least frequent points of origination of those flights. First, we analyze the overall set of flights (shown in Figure 12.9). We see that most flights originated in Atlanta, and very few, less than 2%, originated in Boston. We could say that the Boston flights are an outlier in Delta’s Chicago schedule. The variable of interest is *Origin Airport*.

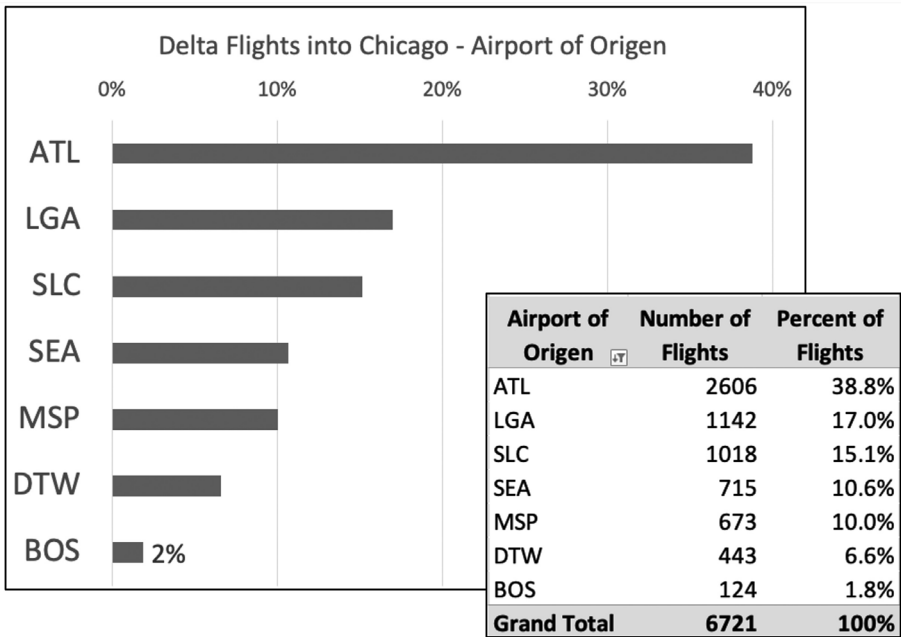


FIGURE 12.9 Clearly, Boston is an outlier in terms of originating Delta flights into Chicago.

When we cross-tabulate the airport of origin with the outlier score using Z-scores, we see that Boston flights only account for less than 1% of all *Very Late* flights. Figure 12.10 shows the tabulation of the *Very Late* flights cross-tabbed with the airport of origin computation, making Boston originating flights outliers in accounting for *Very Late* flights. Out of 124 Boston flights, only one was considered *Very Late* in all of 2021.

	Number of Flights	Percent of Flights
Airport of Origin		
ATL	39	28%
SEA	27	19%
SLC	27	19%
LGA	25	18%
DTW	11	8%
MSP	9	6%
BOS	1	1%
Grand Total	139	100%

FIGURE 12.10 Boston is also an outlier in originating *Very Late* Delta flights into Chicago. It originates very few of them, one out of 6721.

ANOMALY DETECTION BY CLUSTERING – MULTIPLE VARIABLES

If we desire to look for outliers in a situation where there are multiple causes of an effect, we may wish to use the popular clustering machine learning algorithm. We perform a cluster analysis to look for small clusters that are radically different in attribute values than the densest clusters. This might be very useful in a marketing situation where we segment our customer base to look for uniquely different customers from the typical customer. When used in marketing, this often signals a customer group just starting to use our product which we may have overlooked. As we study their attributes, we may discover a new emerging market segment that could be profitable to pursue.

Figure 12.11 shows a clustering analysis where there are outliers. If we identify the clusters, we may gain insight into what makes these unusual members of the population outliers. A popular clustering approach, especially if there is a large population (more than 150, for example), uses a k-means clustering machine learning algorithm. It is popular because it can handle many thousands of rows of data (millions, even), but it requires that the input variables all be numeric.

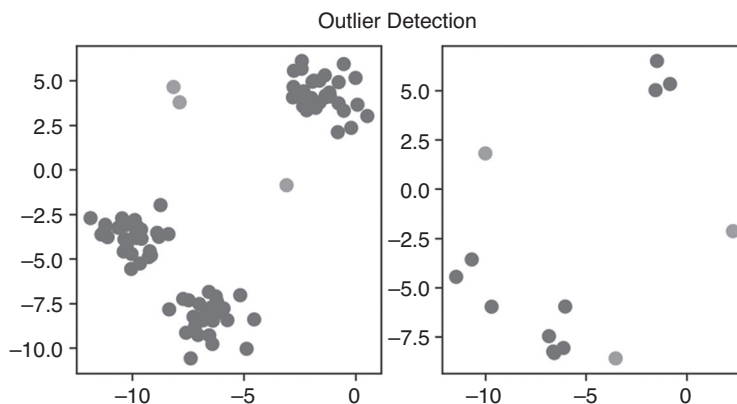


FIGURE 12.11 Detecting outliers with multiple variables using clustering.

We suggest the analyst experiment with just a few clusters rather than let the analysis program select the optimum clusters. Find the number of clusters that makes the most sense to explain the variabilities in the data. It may take some trial and error.

This is a valuable technique when there is no outcome variable to predict but many numeric attributes that may be used to identify outliers. As we discovered earlier, we term this approach unsupervised machine learning.

EXERCISE 12.4 – OUTLIERS IN THE AIRLINE DELAYS DATA SET – CLUSTERING

Let's continue to analyze the Delta Airlines data set, looking for clusters. In this situation, let's see if we can identify the causes of the observed delays. We will relate *Arrival Delay* with *Delay by Carrier*, *Delay Weather*, *Delay Late Aircraft Arrival*, and *Delay National Aviation System*. The definitions of these variables are found in the data dictionary for the data set. Using these five numeric variables and a k-means analysis, we find that four clusters seem to explain much and give us some outlier clusters. As with all other R-based solutions, we used the machine learning functions in the JASP program.

Most delays (6,027, cluster 3) seem to have a minimum average delay so we may call these on-time flights, as seen in Figure 12.12. Cluster 1, the next most significant cluster of 570 flights, has an average delay of 71 minutes, and the causes are relatively evenly distributed across all the other inputs. This is not of much interest. We could call these “Late Flights.” There are two interesting outlier clusters. Cluster 2, which has 52 flights, has an average long delay of almost 290 minutes. The delays there seem to be caused entirely by carrier-caused delays (average delay contribution of 266 minutes). The last cluster, Cluster 4, also seems to be composed of very much delayed flights (an average of 237 minutes of delay). Unlike Cluster 2, these delays seemed to be caused by the National Aviation System (not Delta's fault), with an average delay contribution of 137 minutes for this cluster.

By performing this analysis, we see Delta would be well served by investigating the attributes of the 52 flights in Cluster 2 to determine what the company can do to correct any glaring problems with these flights.

ANOMALY DETECTION USING LINEAR REGRESSION BY RESIDUALS – MULTIPLE VARIABLES

Another approach to identifying outliers when multiple variables are involved is using linear regression. It is beneficial when there is an outcome variable as the predicted variable. In other

Cluster Information

Cluster	1	2	3	4
Size	570	52	6027	72
Explained proportion within-cluster heterogeneity	0.268	0.180	0.138	0.415
Within sum of squares	2.656e+6	1.784e+6	1.364e+6	4.113e+6

Cluster Means

	Arrival Delay	Delay Carrier	Delay Weather	Delay Late Aircraft Arrival	Delay National Aviation System
Cluster 1	71.126	35.477	2.730	12.674	19.800
Cluster 2	286.750	266.750	0.962	3.423	14.481
Cluster 3	-6.023	0.696	0.035	0.356	0.901
Cluster 4	236.833	5.708	47.694	46.264	137.167

FIGURE 12.12 Outliers in the Airline Delays data set using k-means clustering.

words, this is the application of supervised machine learning. We may only apply this technique when all the variables, predicted and predictors, are numeric. We use linear regression as the mean of the distribution (the root mean square mean), and we look for those data points that are very far away from the predicted linear regression, the mean. Figure 12.13 shows a typical linear regression analysis, the linear regression, and a possible outlier.

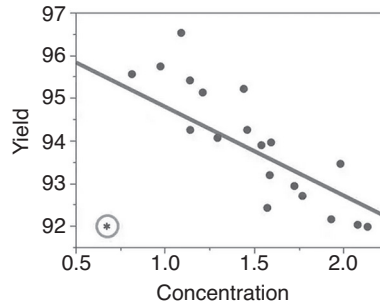


FIGURE 12.13 Outliers are defined by the distance from the linear regression, or mean, of the distribution.

A residual computation measures the distance of the data from the linear prediction for that data point, shown in Figure 12.14. To decide which members of the population are outliers based on the multiple input criteria, we compute the Z-score of the residual and only the three-sigma limit as we did with a single variable. Of course, we may also use the Tukey limit approach, as well.

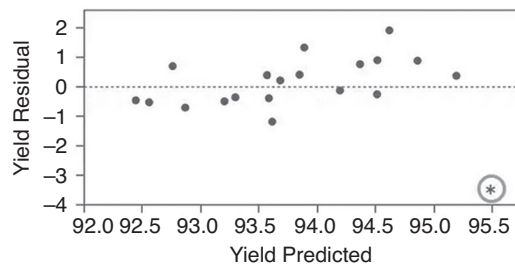


FIGURE 12.14 The size of the residual defines outliers compared to the rest of the residuals.

EXERCISE 12.5 – OUTLIERS IN THE AIRLINE DELAYS DATA SET – RESIDUALS

Let's now apply the residual coefficient techniques to identify outliers in the Delta flight delay project. We will relate *Arrival Delay* as the predicted variable with *Delay by Carrier*, *Delay Weather*, *Delay Late Aircraft Arrival*, and *Delay National Aviation System* as the predictors. First, let's make sure there is a significant correlation between the outcome overall delay variable and the four potential delay causes. Figure 12.15 shows the correlation matrix for all five variables. We see a substantial positive correlation between the possible causes of the overall delay and the other delay variables, so we will use them all in the linear regression computation.

Pearson's Correlations ▼		
Variable	Arrival Delay (Minutes)	
1. Arrival Delay (Minutes)	Pearson's r	—
	p-value	—
2. Delay Carrier (Minutes)	Pearson's r	0.691
	p-value	< .001
3. Delay Weather (Minutes)	Pearson's r	0.321
	p-value	< .001
4. Delay National Aviation System (Minutes)	Pearson's r	0.549
	p-value	< .001
5. Delay Late Aircraft Arrival (Minutes)	Pearson's r	0.354
	p-value	< .001

FIGURE 12.15 Partial view of the correlation coefficient matrix of the variable proposed for the linear regression study of outliers.

Using the JASP linear regression algorithm, we compute the model and request the residuals. Be sure to request *All* under *Residuals – Casewise Diagnostics*. Figure 12.16 shows the regression analysis results, and Figure 12.17 shows a partial view of the residual table. Copy the table in the display and paste it into Excel. Perform a Z-score analysis looking for outliers in the residuals.

Coefficients						
Model		Unstandardized	Standard Error	Standardized	t	p
H ₀	(Intercept)	5.387	0.575		9.368	< .001
H ₁	(Intercept)	-7.760	0.130		-59.642	< .001
	Delay Late Aircraft Arrival	1.068	0.009	0.313	118.026	< .001
	Delay Weather	1.025	0.009	0.304	114.573	< .001
	Delay Carrier	1.046	0.004	0.666	251.142	< .001
	Delay National Aviation System	1.051	0.005	0.518	195.172	< .001

FIGURE 12.16 Partial view of the residual from the linear regression analysis.

Casewise Diagnostics ▼					
Case Number	Std. Residual	Arrival Delay	Predicted Value	Residual	Cook's Distance
1	-4.339	914.000	953.021	-39.021	1.120
2	-2.714	739.000	765.664	-26.664	0.128
3	-2.129	623.000	644.101	-21.101	0.061
4	-2.104	618.000	638.874	-20.874	0.059
5	-0.781	570.000	576.971	-6.971	0.039
6	-1.973	531.000	550.418	-19.418	0.065
7	-1.749	520.000	537.655	-17.655	0.018
8	-1.869	512.000	530.445	-18.445	0.054
9	-1.551	503.000	518.566	-15.566	0.020
10	-1.519	496.000	511.252	-15.252	0.019
11	-0.630	477.000	483.014	-6.014	0.012
12	-1.121	412.000	423.326	-11.326	0.007

FIGURE 12.17 Partial view of the residual from the linear regression analysis.

When we label any residuals with values greater than three sigmas away from the residual mean, we find 140 flights, or 2%, meet the criteria of being outliers. This is similar to the results found earlier when we considered only the average delay, but now we have used multiple variables to identify the outliers.

CASE STUDY 12.1: OUTLIERS IN THE SFO SURVEY DATA SET

As the SFO marketing director, you are looking for unusual customer groups that do not fit the typical customer profile of what you might consider your regular customers (in other words, outliers). Rather than apply the outlier analysis to any one variable, let's see if we can identify if there are any unusual groups by clustering over several behavioral and demographic variables. We will use several demographic variables (gender, age, and income) and behavioral variables (frequency of travel and level of travel) and satisfaction with the airport (net promoter score). We will use clustering and investigate if there are small groups of customers with unusual characteristics. If we select numeric variables (and all the ones mentioned above have numeric scores), we use k-means clustering to identify an unusual cluster, our "outliers."

We will use the overall satisfaction score *NETPRO* as one of the variables. For demographic variables, we use *Q20Age* (age), *Q22Gender* (gender), and *Q24Income* (income). For behavioral characteristics, we use *Q23FLY* (frequent flier), *Q5TIMESFLOWN* (experienced flier), and *Q6LONGUSE* (how long they have flown through SFO).

The questions we will answer using k-means clustering analysis for this case are as follows:

Is there an unusual cluster of passengers that do not fit the typical customer profile? How large is the cluster as a percentage of the airport passengers, and what is the group's profile?

An additional exercise is as follows:

Analyze prior years (2017 and 2016, at least) and see if the identified group in 2018 is relatively new or if it was present in previous years.

SOLUTION IN R

Import the *SFO 2018 Survey Data.csv* data set into JASP (found in the SFO Survey Data folder of the Case Data depository). Choose *K-Means Clustering* under *Machine Learning* and select all demographic variables and behavioral characteristics as variables (change the data type when necessary), as shown in Figures 12.18(a) and (b).



FIGURE 12.18(a) Setting up the k-means clustering analysis for the SFO2018 data set showing all the parameters set up.

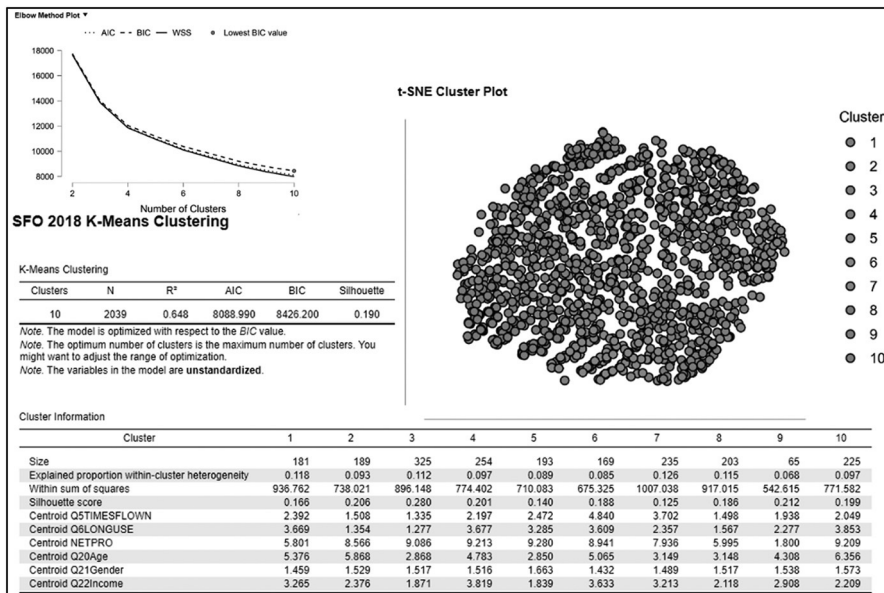


FIGURE 12.18(b) Steps and output of k-means clustering with the SFO2018 data set.

Repeat the previous steps for the 2017 data set, and you will be able to get the results in Figures 12.19(a) and (b).

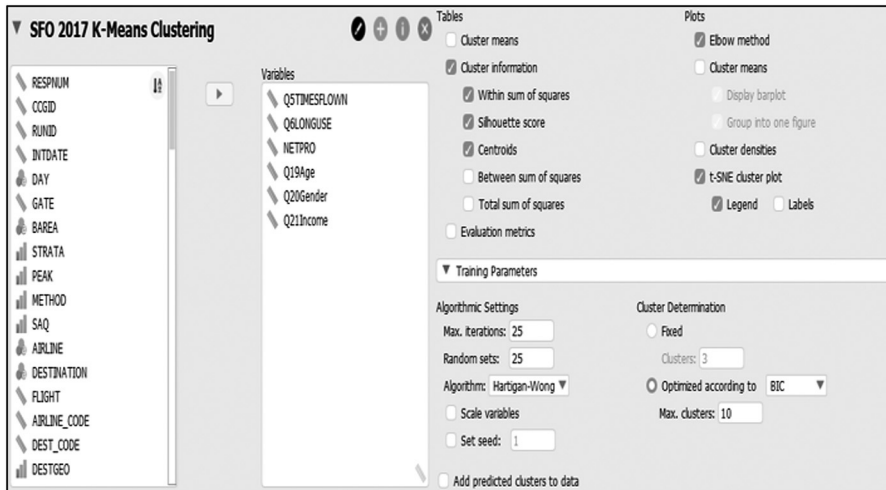


FIGURE 12.19(a) Steps and output of k-means clustering with the SFO2017 data set.

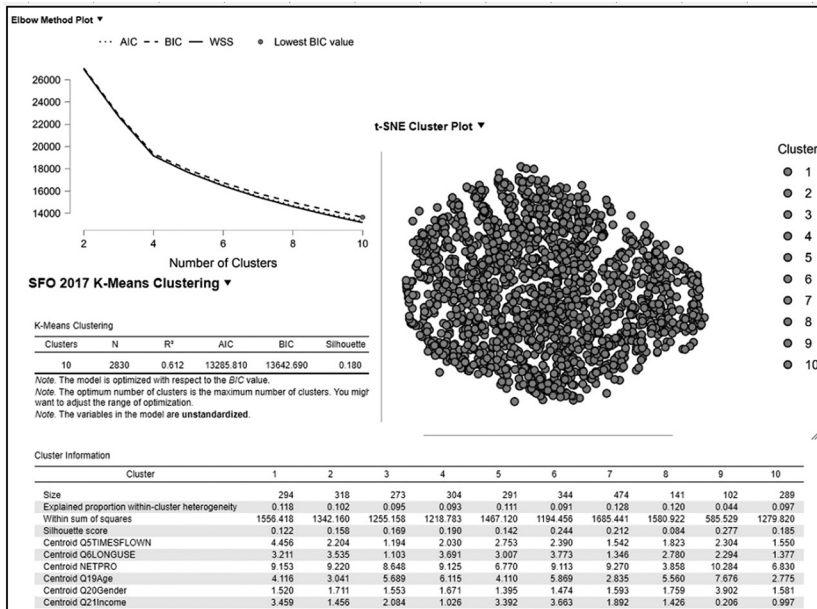


FIGURE 12.19(b) Steps and output of k-means clustering with the SFO2017 data set.

SOLUTION IN PYTHON

Import the *SFO 2018 Survey Data.csv* data set into Orange3. Select all demographic variables and behavioral characteristics by connecting to the *Select Columns* widget. Ask for 10 clusters in the *k-Means* widget and observe the data in the data table and the differences between clusters in the box plot, as shown in Figures 12.20(a), (b), and (c).

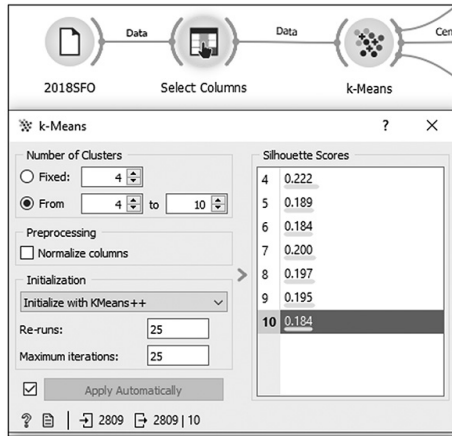


FIGURE 12.20(a) Steps and output of k-means clustering with the SFO2018 data set in Orange3.

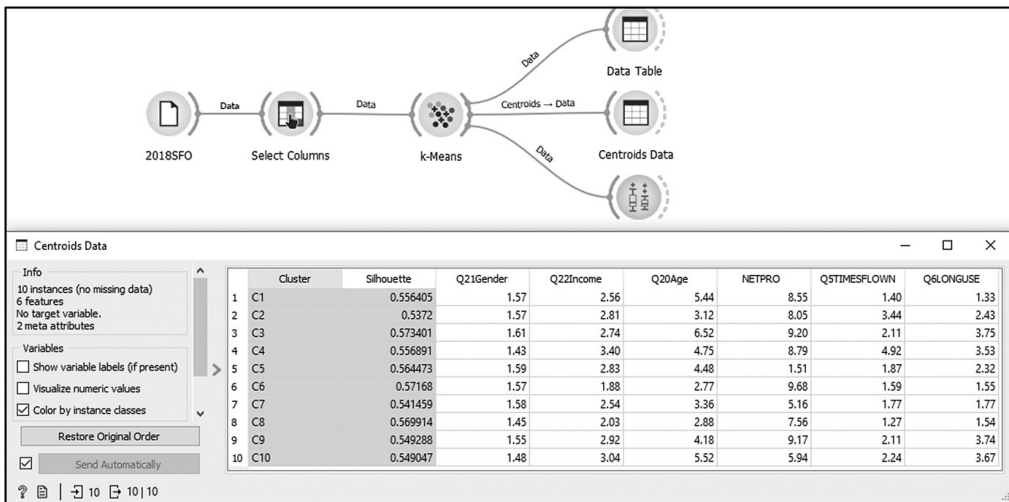


FIGURE 12.20(b) Steps and output of k-means clustering with the SFO2018 data set in Orange3.

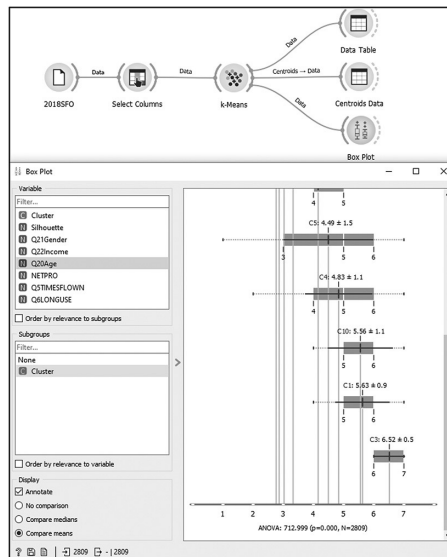


FIGURE 12.20(c) Steps and output of k-means clustering with the SFO2018 data set in Orange3.

Repeat the previous steps for the 2017 data set, as shown in Figures 12.21(a), (b), and (c).



FIGURE 12.21(a) Steps and output of k-means clustering with the SFO2017 data set.

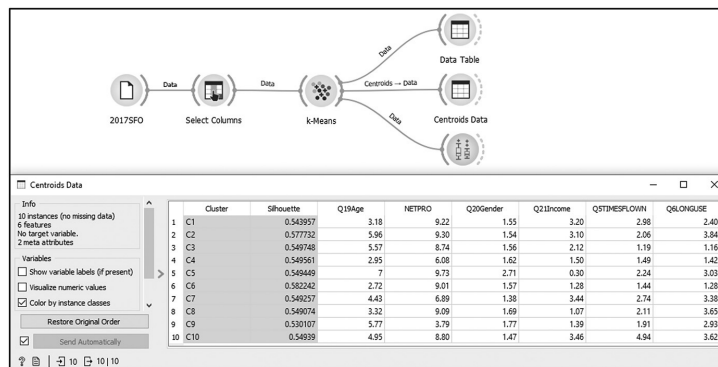


FIGURE 12.21(b) Steps and output of k-means clustering with the SFO2017 data set.

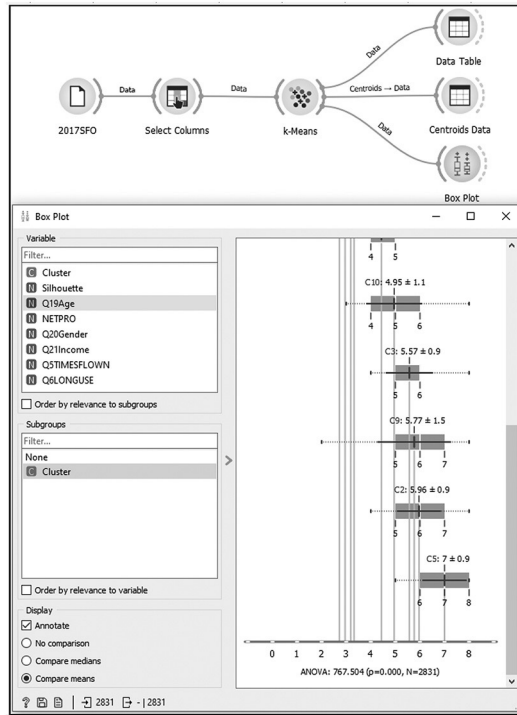


FIGURE 12.21(c) Steps and output of k-means clustering with the SFO2017 data set.

CASE STUDY 12.2: OUTLIERS IN THE SBA LOANS DATA SET

Imagine yourself as the head of the Small Business Administration (SBA). You must report to the US Congress how well your loan program performs. You are concerned that you are making too many loans to what may be considered “large” businesses. After all, you are the agency catering to small businesses. You want to show that you make few loans to large businesses. How many large businesses get loans under your program? Are they frequent, or are they outliers you can easily explain away? You are also concerned that there may be too many “large” loans. Are there loan amount outliers?

You ask your analyst to give you an analysis of the distribution of loans by the size of the business and by the size of the loan. Once we have identified the outliers, we can analyze the characteristics of these businesses to find out their nature and glean any details that will help us explain their uniqueness.

Use the following framed analytical questions:

How many loans may be considered outliers in terms of loan size and business size?

What are some identifiable characteristics of these outliers, if they exist?

Use the *FOIA Loans Data.xlsx* data set found in the *SBA Loans Data* folder in the Case Data depository. The variables for anomaly analysis will be *GrossAmount* and *Jobs Supported*. In

Excel, we will standardize both variables and enter their standardized scores into two additional columns in our table. We will keep only jobs that were issued, so we will keep the binning of *LoanStatus* (as a binary variable, 0 = CHGOFF, default, and 1 = PIF, paid off loan). Once we have identified the outliers, let's do a demographic study by characterizing them from their features: *LoanStatus*, *TermInMonths*, *BusinessType*, *NaicsCode*, and *CDC_State*.

SOLUTION IN R

Once you have all three data tables ready, import them into Jamovi. Run a *Descriptive Analysis* under *Exploration* for different tables, as shown in Figures 12.22(a)–(d).

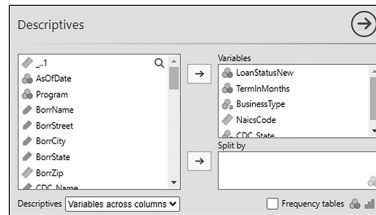


FIGURE 12.22(a) Steps of descriptive analysis in Jamovi.

LoanSD > 3 Descriptives					
Descriptives					
	LoanStatusNew	TermInMonths	BusinessType	NaicsCode	CDC_State
N	3720	3720	3720	3712	3712
Missing	0	0	0	8	8
Mean	0.199	247		535714	
Median	0.00	240		531210	
Standard deviation	0.399	31.2		167206	
Minimum	0	120		111219	
Maximum	1	300		812990	

FIGURE 12.22(b) Output of descriptive analysis.

JobSD > 3 Descriptives					
Descriptives					
	LoanStatusNew	TermInMonths	BusinessType	NaicsCode	CDC_State
N	813	813	812	544	808
Missing	0	0	1	269	5
Mean	0.603	229		449608	
Median	1	240		424480	
Standard deviation	0.490	35.7		155936	
Minimum	0	64		111219	
Maximum	1	300		812990	

FIGURE 12.22(c) Output of descriptive analysis.

SBALoans Descriptives					
Descriptives					
	LoanStatusNew	TermInMonths	BusinessType	NaicsCode	CDC_State
N	183714	183714	183670	162162	183106
Missing	0	0	44	21552	608
Mean	0.451	237		532983	
Median	0.00	240		541211	
Standard deviation	0.498	28.5		169962	
Minimum	0	0		111150	
Maximum	1	389		999990	

FIGURE 12.22(d) Output of descriptive analysis.

SOLUTION IN PYTHON

Use the *FOIA Loans Data.xlsx* data set found in the *SBA Loans Data* folder in the Case Data depository. Once you have all three data tables ready, import them into Orange3. Select all variables in the *Select Columns* widget and follow this with the *Feature Statistics* widget as shown in Figures 12.23(a), (b), and (c).



FIGURE 12.23(a) Steps and output of the descriptive analysis.



FIGURE 12.23(b) Steps and output of the descriptive analysis.

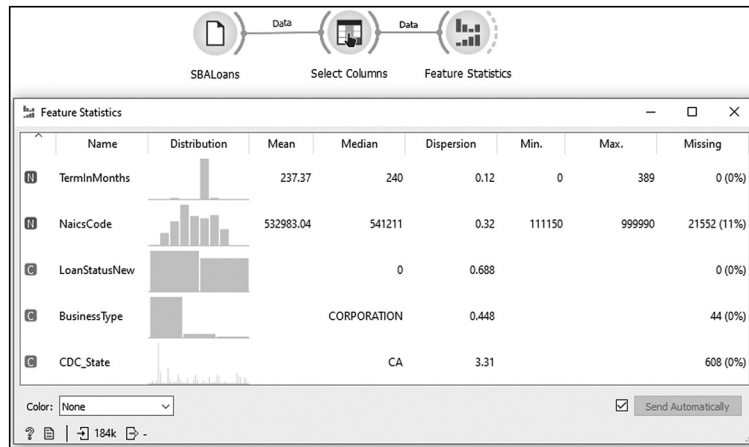
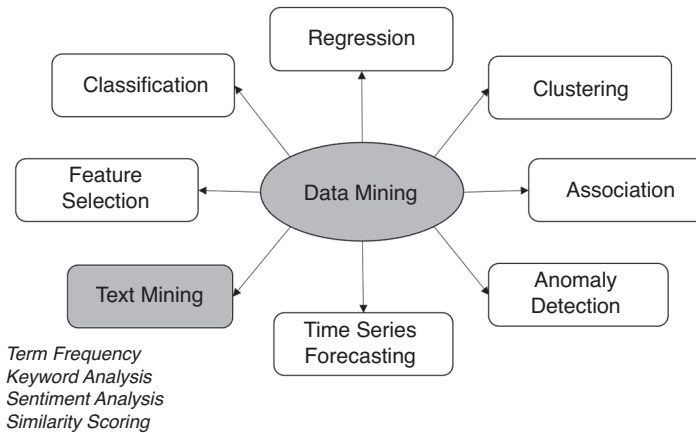


FIGURE 12.23(c) Steps and output of the descriptive analysis.

REFERENCES

- Bureau of Transportation Statistics. 1997. Library of Congress. 1997. <https://www.loc.gov/item/lcwaN0031204/>.
- Hoaglin, D. C. 2003. "John W. Tukey and Data Analysis." *Statistical Science* 18 (3): 311–18.
- Tukey, John W. 1977. *Exploratory Data Analysis*, 131–60. Reading, MA: Addison-Wesley.

TEXT DATA MINING

Data science has developed considerably, so we now have many remarkable techniques and tools to extend data analysis from numeric and categorical data to textual data. Sifting through the open-ended responses from a survey, for example, was an arduous process when performed by hand. With the advent of text data analytic tools to comb through social media, the data set for analysis grew from just a few hundred survey responses to tens of thousands of social media postings, which would be impossible to analyze by hand unless the process is automated. The result is the rise in the need and the solutions for text data mining. It is an essential approach in the business world, where we want to quickly extract customer sentiment, for example, or categorize social media postings. Accelerating advances in natural language processing techniques was the response. These techniques have come out of the lab and become mainstream in their use. It is now widespread and even imperative to analyze text variables in a data set alongside techniques to mine information from numeric and categorical variables. This chapter aims to make the emerging text analytical techniques accessible to business data analysts. This subject is explored in great detail in another book in this series (Fortino 2021), and we invite the reader to use the resources there.

WHAT IS TEXT DATA MINING?

Text data mining, in simple terms, is finding valuable patterns in text data. We will explore some of these useful and practical text data mining techniques. Figure 13.1 shows many examples of text data mining techniques.

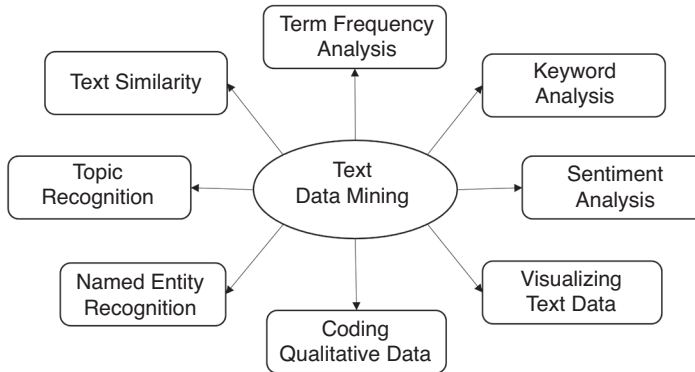


FIGURE 13.1 The many techniques of text data mining.

A *keyword analysis* is analyzing the keywords or search phrases that bring visitors to your website through organic and paid searches, for example. As such, keyword analysis is the starting point and cornerstone of search marketing campaigns. By understanding what queries are typed into search engines by qualified visitors to your website, search marketers can better customize their content and landing pages to drive more traffic and increase conversion rates. For this reason, keyword analysis is an essential skill for SEO (search engine optimization) and PPC (per-click marketing) experts.

Sentiment analysis (also known as *opinion mining* or *emotion AI*) uses natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to the Voice Of the Customer (VOC) materials such as reviews and survey responses, online and social media postings, and healthcare materials for applications that range from marketing to customer service to clinical medicine.

Text visualization uses graphs, charts, or word clouds to showcase written data visually. This provides quick insight into the most relevant keywords in a text, summarizes content, and reveals trends and patterns across documents.

Coding Qualitative Data is the process of labeling and organizing your qualitative data to identify different themes and the relationships between them. When coding customer feedback, you assign labels to words or phrases that represent important (and recurring) themes in each response. These labels can be words, phrases, or numbers; we recommend using words or short phrases since they are easier to remember, skim, and organize. Coding qualitative research to find common themes and concepts is part of thematic analysis, which is part of qualitative data analysis. Thematic analysis extracts themes from the text by analyzing the word and sentence structure.

Named-entity recognition (NER) (also known as *(named) entity identification*, *entity chunking*, and *entity extraction*) is a subtask of information extraction that seeks to locate and classify

named entities mentioned in unstructured text into pre-defined categories such as personal names, organizations, locations, medical codes, time expressions, quantities, monetary values, and percentages.

Topic analysis is a Natural Language Processing (NLP) technique that allows us to automatically extract meaning from text by identifying recurrent themes or topics. A topic model is a type of statistical model for discovering the abstract “topics” that occur in a collection of documents. Topic models are a suite of algorithms that uncover the hidden thematic structure in document collections. These algorithms help us develop new ways to search, browse, and summarize extensive archives of texts. Topic models provide a simple way to analyze large volumes of unlabeled text. A *topic* consists of a cluster of words frequently occurring together.

Text similarity helps us determine how “close” two pieces of text are both in surface closeness (lexical similarity) and meaning (semantic similarity).

WHAT ARE SOME EXAMPLES OF TEXT-BASED ANALYTICAL QUESTIONS?

Let’s consider the case study on the Titanic disaster introduced in Chapter 3 on framing analytical questions. Pretend you work for a famous newspaper. Your boss is the news editor of the newspaper. It is almost the one-hundredth anniversary of the Titanic disaster. The editor assigned a reporter to cover the story. The reporter submitted an article stating, “The crew of the Titanic followed the Law of the Sea in responding to the disaster.” The editor is concerned that this may not be true and assigned you to fact-check this item. You decide to approach it from an analytic point of view. Your analysis of the assignment yielded answers to the following question:

Did the crew of the Titanic follow the Law of the Sea in responding to the disaster?

Suppose that in pursuing the story, the reporter conducted interviews to ask them questions about the disaster. Some questions yielded categorical or numerical data, as many surveys do. However, as is often the case, an open-ended question was posed at the end of the survey: “How do you feel about the operators of the ocean liner not supplying enough lifeboats for everyone to be saved?” This is an example of a possible question the reporter may have asked at that time. Suppose the disaster had occurred recently. Typically, reporters will collect answers to that question from a few individuals as a sample of how “the general public feels.” In this case, the survey would be conducted electronically through social media and collected hundreds of responses. The reporter in this recent case was overwhelmed and sought your help as an analyst to extract additional meaning from this larger data set.

Parsing the information need: This is about the feelings of each respondent, and their conception of the operators of ocean-going cruises, exemplified by the Titanic, which is supported by their experience or knowledge of cruises.

Framed text analytical questions: We determine that there is some text analysis we could undertake to extract meaning from the collected textual responses.

Do the people posting comments about the disaster feel positively or negatively toward the operators of the Titanic?

What keywords are mainly used to express their opinion?

Is there a visual that can easily represent these keywords and their sentiment?

TOOLS FOR TEXT DATA MINING

What tools are used in text data mining? There are a few accessible toolsets available for the practical data analyst who has to accomplish straightforward text analysis tasks. Here, we describe some of the more common tool sets, either free or as part of a standard set of software in use in most businesses and universities.

Excel

Excel is an excellent tool for cleaning and shaping data files. Because we deal with so much text, Word, another Microsoft product, is a helpful companion tool to Excel to shape text data. The combined use of these two tools, Excel and Word, should suffice for any text data-wrangling needs of the average data analyst. Other spreadsheet software is equally helpful if you have the skills to work with it. Google Sheets may be used in place of Excel, but it does not offer any particular advantage. Use whichever spreadsheet program is most familiar to you to create the necessary data tables.

Microsoft Word

Word is the workhorse text manipulation platform for most text data mining purposes. First, it is ubiquitous and readily available. Second, because of its ubiquity, most professionals are skilled in its use. These skills can be put to work for our text data manipulation needs. Creative uses of the Edit -> Find -> Replace function can go a long way to shaping data that has been scraped from a document or a website and converting it into a text form usable for analysis.

R and JASP and Jamovi

In this book, we use the R advanced analytics environment. R is a programming language often used for statistical computing and, more recently, for more advanced analysis such as machine learning. R comes with a programming interface that is command-line driven. It needs to be programmed to perform any analysis. Graphical user interfaces offer pull-down menus (a GUI) to make R easier to use, such as JASP and Jamovi. However, there is no simple graphic interface for the text analytics capabilities in R. For text data mining, we must invoke the R functionality via the command line to use R's powerful text analytics capabilities (the *tidytext* package) (Silge 2016).

Voyant

Voyant Tools is an open-source, web-based application for performing text analysis (Sinclair 2016). It supports scholarly reading and interpretation of texts or a corpus, particularly by scholars in the digital humanities, but it can also be used by students and the general public. It can be used to analyze online texts or any text uploaded by users. We use Voyant throughout this chapter as an effective analysis platform for textual data. It can be used via a web interface or downloaded for those who are security-minded and do not want their text data uploaded to an unknown web server. Voyant is a web-based and downloadable program available at <https://voyant-tools.org/>. The code is under a GPL3 license and the content of the web application is under a Creative Commons by Attribution License 4.0, International License.

SOURCES AND FORMATS OF TEXT DATA

Numerical and categorical data are the most common data types. We use standard techniques to work with these data types, such as pivot tables and numerical summarization functions. With

the advent of social networks and the development of sophisticated data tools, text data analysis is now more commonplace.

Business managers often want to know about certain aspects of the business, such as “What is the meaning of what people are saying about our company on Twitter or Facebook?” or “Does our use of keywords on our website match or surpass that of our competitors?” In other words, “Do we have the right keywords or enough of them for search engines to classify the company website higher in search returns than our competitors (i.e., search engine optimization and SEO analysis)?” These types of questions require that analysts do a thorough job analyzing the web page content text.

In customer conversational interactions, it is essential to look at the text that a person wrote. Why? We already know that a combination of keywords and phrases is essential to a post. Before analyzing, we need to know what words and phrases people use. This analysis is accomplished by looking at the texts with an eye to word frequency, sentiment, and keywords.

It is essential to know where text data is found and in what form to optimize the scraping and shaping process and ultimately produce it in the correct format for analysis. We discuss the various forms in which text data comes across our desk. We also investigate how to extract text data from its native format and shape it into a form that can be easily analyzed with our tools. We also cover some techniques you may need to employ to acquire the data.

Social Media Data: Examples of social media data sources are Facebook, Twitter, and LinkedIn. They can be excellent sources of customer text data.

Customer opinion data from commercial sites: There are essential customer feedback data from online shopping. This data is available as text and can be evaluated using the techniques in this book. Customer reviews and opinions are other excellent product and service feedback sources in text form.

Email: Emails are another interesting source of text data. The stream of emails can be analyzed in real-time as we would with social media data, but that would require sophisticated commercial software.

Documents: Documents are another source of text data and may be in the form of contracts, wills, and corporate financial reports.

Surveys: When we conduct surveys, we have specific questions in mind, and we are cautious about how we ask those questions. Typically, the answer to those questions yields either categorical or numerical data, which can be analyzed using standard techniques. Surveys often ask, “Do you have anything else to tell us?” We expect a sentence or two of free-form text with the respondent’s opinion.

Websites: Websites are a good source of text data. We may want to do a similarity scoring of a page on our company’s website against our competitors or perform a keyword analysis of our page to improve our standing concerning search engines (SEO).

TERM FREQUENCY ANALYSIS

Word frequency analysis is the most fundamental technique in text analysis. It is essentially the counting of words and is the starting point for most investigations. We presume that the most frequently appearing words hold some meaning; they are more important than other words. We tabulate their frequency because they likely have greater significance in our text’s context. We ignore nuances, such as grammatical structure (for example, is the word a noun or a verb?)

and how a term is used (Sarcasm? Irony? Fact?). We also know that not all words carry significant meaning, such as prepositions or conjunctions (for example, “and,” “we,” “at,” or “in.”) We assume we can safely ignore these words (called stopwords) and remove them from the text. We strive to create a Bag-of-Words text data file (document) or text data field (data point in a spreadsheet cell). Then we count the most meaningful words in the text to determine which are more and less important.

This technique is also called Term Frequency (TF) analysis. It helps us quickly extract meaning from the text being analyzed. We count the occurrence of each word in the document. For business purposes, we are not interested in an in-depth linguistic analysis of the text but to quickly getting a sense of what is contained in it and what it means. We want to find the most salient topics and themes.

HOW DOES IT APPLY TO TEXT BUSINESS DATA ANALYSIS?

We can use this technique in several ways. Say we surveyed what customers wanted in a product. Besides asking quantitative questions (such as “on a scale of 1 to 5...”), we also ask open-ended questions (such as “Is there anything else you want us to know?”). Doing a word frequency analysis of the text responses to the open-ended question and comparing it to the word frequency analysis of the product description tells us if we are meeting expectations. The more the word frequency tables match their numeric responses, the more we match the customer’s expectations.

EXERCISE 13.1 – CASE STUDY USING A TRAINING SURVEY DATA SET

Suppose you were just hired to teach a course on data analysis. You would want to know what your audience desires to know. You send each participant a simple three-question survey: (1) What do you want to get out of the class?, (2) How would you rate your Microsoft Excel skill level?, and (3) What is your official title? The last two questions can be analyzed using standard methods because the variables are categorical. The first question is about free-form text and is more difficult to analyze. We use word frequency analysis to discover what they want to learn.

In this chapter, we use word frequency analysis to answer the following question:

What are the most frequent words that signify the essence of what the trainees want to learn in this seminar?

Access the *Case Data* depository and find in the *Chapter 13* folder the file *Attendee PreSurvey Results in Data.csv*. Open it with Excel. Save it as *Attendee PreSurvey Results Data.xlsx*. You will see the textual responses from 17 attendees in the second column. We will use the Voyant cloud-based server accessible tool over the internet at: <https://voyant-tools.org/>.

Copy the contents of the attendee’s comments on what they want to see in the course (cells B2-B18) to your computer buffer.

Open the Voyant tool using the web-based version (<https://voyant-tools.org/>) or run the Voyant server, which can be downloaded and installed on your computer. Paste the attendees’ comments into the data entry box in Voyant and press *Reveal*. In the upper left-hand corner panel, click on *Terms* to switch from the word cloud mode to the table mode. The resulting word frequency list sorted by most frequent words should look something like that in Figure 13.2.

		Term	Count	Trend
+	<input type="checkbox"/>	1 data	27	
+	<input type="checkbox"/>	2 analysis	9	
+	<input type="checkbox"/>	3 excel	6	
+	<input type="checkbox"/>	4 learn	5	
+	<input type="checkbox"/>	5 looking	5	
+	<input type="checkbox"/>	6 tools	5	
+	<input type="checkbox"/>	7 better	4	
+	<input type="checkbox"/>	8 interpreting	4	
+	<input type="checkbox"/>	9 like	4	
+	<input type="checkbox"/>	10 quantitative	4	
+	<input type="checkbox"/>	11 use	4	

FIGURE 13.2 Word frequency result in Voyant for attendees' survey responses.

WORD FREQUENCY ANALYSIS USING R

Access the case files' repository and find and open the file *Attendee PreSurvey Results in Data.csv* with Excel. Copy and rename the *Attendee PreSurvey Result data.csv* as *casea.csv*.

Open R in *RStudio*. Install the packages we need using *Repository* (CRAN):

```
dplyr, tidytext
```

Import the library and read the data:

```
> library(dplyr)
> library(tidytext)
> casea <- read.csv(file.path("casea.csv"), stringsAsFactors = F)
```

Tokenize the contents of the data set and remove the stop words:

```
> tidy_a <- casea %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words)
```

Get the results of the word frequency analysis (shown in Figure 13.3):

word	n
<chr>	<int>
data	27
analysis	9
excel	6
learn	5
tools	5
interpreting	4
quantitative	4
analyze	3
decisions	3
understanding	3

1-10 of 109 rows

FIGURE 13.3 Word frequency data frame of the training survey text field as computed in R.

```
> tidy_a %>%
  count(word, sort = TRUE)
```

Notice that the results using the Voyant program and as computed by R are very similar.

KEYWORD ANALYSIS

A keyword analysis is also known as *keyword extraction* or *keyword detection*. A *keyword analysis* is a technique that extracts the most frequent and most important words and expressions from text data. It helps summarize the content of texts and recognize the main topics discussed. It is most powerfully used to optimize how search engines find and index web pages from your company's website. It is an integral part of SEO.

With keyword analysis, you can find keywords from all types of text data: documents, business reports, social media postings, online reviews, and news reports. Suppose you want to analyze your product's many online reviews on your favorite e-commerce site, like Amazon. Keyword extraction helps you sift through all the text data comprising the reviews and quickly obtain the most important and frequent words that best describe the reviews. Then you can see what your customers are mentioning most often, saving you the work of examining all the reviews manually.

We can use the techniques we studied in above to compute the word frequencies in a document to find keywords. It is just a matter of determining how many of the most frequent words are enough to fully characterize a document by its keywords. The analyst makes that decision. The top 5? The top 10? The top 25? You determine that by inspection.

You can start with keywords and look for their occurrence in a target document. Let's say you want to know what percentage of your customers are price sensitive. Looking at social media postings or product feedback data, you can do a keyword search across all postings and determine how frequent "price" is as a keyword. You can compare two documents once you compute the keywords (the topmost frequently used words). Look for the frequency of occurrence in a target document of the keywords extracted from a source document. Looks also for price-associated words like "expensive" or "inexpensive," "pricey" or "cheap," for example. We can make it the basis for document comparison, as well. This technique has many uses in the business world. We explore a few such uses through the exercises in this chapter.

The Excel techniques we present here are based on single-word searches. Other tools we use — Voyant, and even R — can also analyze two- or three-word keyword phrases, which can be very useful. If Excel is your tool of choice and you do not venture further, there are inexpensive add-ons to Excel that do word frequency analysis, including two- and three-word sequences.

EXERCISE 13.2 – CASE STUDY USING DATA SET D: RÉSUMÉ AND JOB DESCRIPTION

In this exercise, we try to match a résumé to job descriptions. Suppose you are looking for work and select a small number of jobs that seem interesting. Can you narrow the search and prioritize the assignments using keyword analysis? Essentially, you want to answer the following question:

Which jobs am I interested in applying for that match my résumé most closely?

We extract the most frequent words from a résumé and use the most appropriate ones for a keyword search through the list of jobs. We use a generic engineer's résumé and select the definitions of a few software development occupations from the O*NET database of occupations in place of real jobs (O*Net 2021). We perform a word frequency analysis of the résumé to get the list of the most frequent words (using techniques in the section above). We also do a word frequency analysis of the selected occupations and call these the keywords to be used against the résumé. Then we use `COUNTIF` to see how many of these keywords appear in the résumé for each occupation to rank the frequency of appearance of the keywords in the résumé.

KEYWORD WORD ANALYSIS IN VOYANT

Access the *Case Data* depository of case files, and in the folder *Chapter 13*, open file *O*NET JOBS Plus Resume.csv* in Excel. Open the Voyant tool using the web-based version (<https://voyant-tools.org/>). From the *O*NET JOBS Plus Resume.csv* file, scrape the résumé from the first row in the *description* cell.

Paste the résumé's text into the Voyant data entry screen and press *Reveal*. You will see the familiar analysis screen. Select *Table* instead of *Cirrus* (word cloud) in the upper left-hand panel. This gives the word frequencies as a table rather than as a picture, as shown in Figure 13.4.

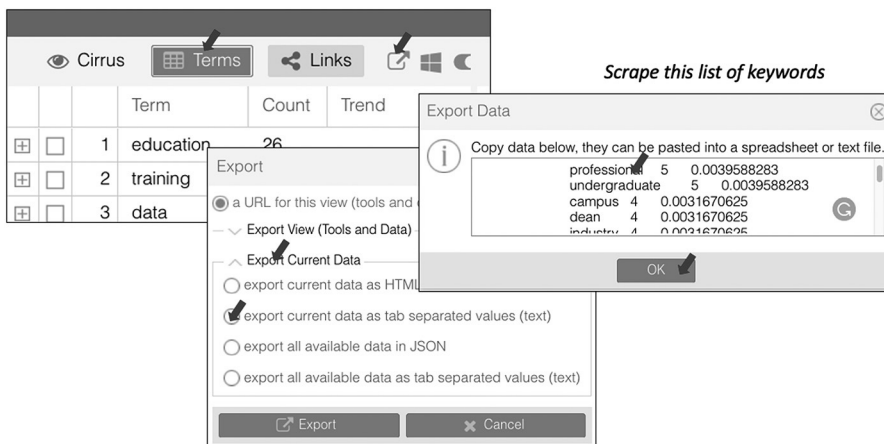


FIGURE 13.4 Finding the term frequency for the résumé using Voyant.

Scrape the keywords and paste the data into a new Excel spreadsheet. Label the column *Resume*. Repeat steps 3, 4, and 5 for *Software Developers, Application; Software Developers, Systems; Database Administrators; Database Architects; and Data Warehousing Specialists*. After saving the resulting frequent term tables, scrape each term frequency and paste them into the opened spreadsheet in separate columns; label each column with the occupation name. Once the Excel spreadsheet is populated with the résumé terms and keywords from each occupation, use the Excel `COUNTIF` function to find all the occupation keyword occurrences in the résumé.

Add all the occurrences by occupation to find the occupation with the highest number of hits against the résumé. Figure 13.5 shows the resulting computations. The résumé appears to be most similar to the two *System Development* jobs than the others, but not by much.

Resume	Database Administrators	Database Architects	Data Warehousing Specialists	Software Developers, Applications	Software Developers, Systems Software
education		5	4	5	7
training	computer	1 database	0 computer	1 software	0 software
data	databases	0 design	0 databases	0 analyze	0 systems
new	coordinate	0 systems	1 coordinate	0 applications	0 design
york	database	0 architects	0 database	0 computer	1 aerospace
managemen	implement	0 construct	0 implement	0 design	0 analysis
academic	administer	0 databases	0 administer	0 develop	1 analyze
development	administrators	0 enterprise	0 administrators	0 aim	0 applications
ny	applying	0 existing	0 applying	0 application	1 apply
technology	changes	0 functionality	0 changes	0 area	1 business
fortino	knowledge	1 integrate	0 knowledge	1 client	0 communications
program	management	1 large	0 management	1 coordinating	0 compilers
business	measures	0 new	1 measures	0 create	0 computer
university	plan	0 operations	0 plan	0 customize	0 computing
analytics	safeguard	0 performance	1 safeguard	0 database	0 develop
courses	security	1 programming	0 security	1 databases	0 developers
faculty	systems	1 refine	0 systems	1 developers	0 distribution
graduate	test	0 relational	0 test	0 development	1 embedded

FIGURE 13.5 Completed comparison analysis to determine which job had the highest keyword hits against the résumé (partial view).

TERM FREQUENCY ANALYSIS IN R

In the *Case Data* file folder under *Chapter 13*, copy *O*NET JOBS.csv* and name the copy *cased.csv*. Install the packages we need using Repository (CRAN):

```
dplyr, tidytext, textstem, readr
```

Import the library and read the case data (Figure 13.6 shows the example résumé data):

```
> library(dplyr)
> library(tidytext)
> library(textstem)
> library(readr)

> cased <- read.csv(file.path("cased.csv"), stringsAsFactors = F)
> resume <- read_file("fortino_resume.txt")

# make resume content a dataframe
> resume_df <- tibble(text= resume)
> resume_df
```

	text
1	ANDRÉS GUILLERMO FORTINO, PE, PHD 75 Grist Mill L...

FIGURE 13.6 Data frame object of *resume_df*.

Select the jobs (occupations) we are interested in (results shown in Figure 13.7):

```
> target_d <- cased %>% filter_at(vars(job), any_vars(. %in% c('Software
Developers, Applications',
'Software Developers, Systems Software',
'Database Administrators',
'Database Architects',
'Data Warehousing Specialists'))))

# concatenate the job title and job description
```

```
> target_d <- target_d %>%
  unite(txt, job, description, sep = " ", remove = FALSE)
```

^	txt	job	description
1	Software Developers, Applications Software Developer...	Software Developers, Applications	Software Developers, Applications Develop, create, an
2	Software Developers, Systems Software Software Deve...	Software Developers, Systems Software	Software Developers, Systems Software Research, desi
3	Database Administrators Database Administrators Ad...	Database Administrators	Database Administrators Administer, test, and imple..
4	Database Architects Database Architects Design strat...	Database Architects	Database Architects Design strategies for enterprise d
5	Data Warehousing Specialists Data Warehousing Speci...	Data Warehousing Specialists	Data Warehousing Specialists Design, model, or imple

FIGURE 13.7 Data frame object of *target_df*.

Add additional stop words for the résumé content, if needed (optional):

```
> my_stop_words <- tibble(
  word = c("ny", "york", "fortino"), lexicon = "resume")

> all_stop_words <- stop_words %>%
  bind_rows(my_stop_words)
```

Tokenize the contents of the data set, lemmatize the words, and remove the stop words:

```
# for 'Software Developers, Applications'
> t1 <- target_d[1,] %>%
  unnest_tokens(word, txt) %>%
  mutate(word = lemmatize_words(word)) %>%
  anti_join(stop_words)

# for 'Software Developers, Systems Software'
> t2 <- target_d[2,] %>%
  unnest_tokens(word, txt) %>%
  mutate(word = lemmatize_words(word)) %>%
  anti_join(stop_words)

# for 'Database Administrators'
> t3 <- target_d[3,] %>%
  unnest_tokens(word, txt) %>%
  mutate(word = lemmatize_words(word)) %>%
  anti_join(stop_words)

# for 'Database Architects'
> t4 <- target_d[4,] %>%
  unnest_tokens(word, txt) %>%
  mutate(word = lemmatize_words(word)) %>%
  anti_join(stop_words)

# for 'Data Warehousing Specialists'
> t5 <- target_d[5,] %>%
  unnest_tokens(word, txt) %>%
  mutate(word = lemmatize_words(word)) %>%
  anti_join(stop_words)

> tidy_resume <- resume_df %>%
  unnest_tokens(word, text) %>%
```

```
mutate(word = lemmatize_words(word)) %>%
anti_join(all_stop_words)
```

Find the top ten most frequently used keywords in the *résumé* (Figure 13.8):

```
> kwtop10 <- tidy_resume %>%
  count(word, sort = TRUE) %>%
  filter(n>3) %>%
  slice(1:10)

> kwtop10["word"]
```

	word	n
1	education	26
2	train	20
3	program	18
4	datum	17
5	management	15
6	academic	13
7	development	13
8	develop	11
9	technology	11
10	business	9

FIGURE 13.8 Top ten most frequently used words in the *résumé* and number counts.

We now use these frequently used *résumé* words as keywords for our analysis. Using these top ten words as keywords, do a frequency search for them in the job file to match the *résumé* to the jobs. Find which jobs have the most keywords corresponding to the keywords of the *résumé* (Figure 13.9 gives an example of this matching):

```
> kwt1 <- t1 %>%
  count(word, sort = TRUE) #>% filter(n>1)

> kwt2 <- t2 %>%
  count(word, sort = TRUE) #>% filter(n>1)

> kwt3 <- t3 %>%
  count(word, sort = TRUE) #>% filter(n>1)

> kwt4 <- t4 %>%
  count(word, sort = TRUE) #>% filter(n>1)

> kwt5 <- t5 %>%
  count(word, sort = TRUE) #>% filter(n>1)

# find out which keywords appear both in the job (Software Developers,
# Applications) and resume
> intersect(kwt1["word"], kwtop10["word"])
```

word
<chr>
develop
development
program

FIGURE 13.9 Words that appear both in the résumé and the job description for Software Developers (Applications).

```
# get the total number of keywords
> length(intersect(kwt1["word"], kwttop10["word"])$word)
# find out which keywords appear both in the job (Software Developers,
Systems Software) and resume
> intersect(kwt2["word"], kwttop10["word"])

# get the total number of keywords
> length(intersect(kwt2["word"], kwttop10["word"])$word)

# find out which keywords appear both in the job (Database
Administrators) and resume
> intersect(kwt3["word"], kwttop10["word"])

# get the total number of keywords
> length(intersect(kwt3["word"], kwttop10["word"])$word)

# find out which keywords appear both in the job (Database Architects)
and resume
> intersect(kwt4["word"], kwttop10["word"])

# get the total number of keywords
> length(intersect(kwt4["word"], kwttop10["word"])$word)

# find out which keywords appear both in the job (Data Warehousing
Specialists) and resume
> intersect(kwt5["word"], kwttop10["word"])

# get the total number of keywords
> length(intersect(kwt5["word"], kwttop10["word"])$word)
```

VISUALIZING TEXT DATA

An analyst typically creates visuals of the analysis results as the analysis progresses. These are graphs of data for analysis; they are rough graphs with no thought given to making them compelling at this point in the analysis. It is likely that no one other than the analyst will ever see most of those rough analysis charts. These graphs may even accumulate in an electronic research notebook (typically a PowerPoint document) with slides as containers for the analysis charts. At the end of the analysis, these graphs and numerical summaries of results accumulated in such a notebook are used to draw conclusions and answer questions. We call this charting process *data visualization for analysis*. The tools and techniques shown in this chapter help create preliminary charts to make sense of the textual data and start the detailed analysis process. The last step

is to create compelling visuals that tell the story. This last step in creating a story with data is *data visualization for communication*.

The process of creating a few well-crafted visuals from the many used for analysis is described in the book *Data Visualization for Business Decisions* (Fortino 2020). Often, analysts are not given much time to present their findings. If you look at the work of neurobiologist John Medina (Medina 2008), he encourages us to use no more than ten minutes to make our case, lest we bore our audience. In any event, we must present our findings with as few slides as possible. The analyst looks over the rough graphs produced in the analysis, looks at the conclusions, and then asks: “Which of these are the most powerful visuals to make the point and underscore conclusions most compellingly?” Probably no more than three or four such visuals must be recreated or enhanced to make them more readable to new eyes.

In this section, we will concentrate on producing visuals of our text to understand the import of all those words, whether for analysis or to communicate.

EXERCISE 13.3 – CASE STUDY USING THE TRAINING SURVEY DATA SET

You work for Human Resources in the training and development department. You polled employees in your company who are about to undergo training in data analysis. You want to inform the instructor what the students wish to learn in the class. With this analysis, we want to quickly understand what the employees are telling us they want from the class, so we also create a word cloud of their input. The question we want to answer is as follows:

Can we create a picture of the most frequent words of their open-ended requests from the survey?

VISUALIZING THE TEXT USING EXCEL

The first place to start is to do a word frequency analysis, as shown earlier in the chapter. From the *Case Data* depository, in the *Chapter 13* folder, open the *Attendee PreSurvey Results Data.csv* in Excel, and copy the open-ended comments column into the computer buffer. Return to the earlier section and perform a term frequency analysis, using Voyant and paste the results into Excel.

Paste the Voyant results into a worksheet and select all the words with occurrences greater than and equal to 3, as shown in Figure 13.10.

Row Labels	Count of COUNT
Data	39
business	14
Analysis	11
questions	10
tools	8
value	6
decisions	5
using	5
goes	4
analyzing	4
interpreting	4
Answer	4
organization	4
basic	3
Identify	3
Create	3
Issues	3
information	3

FIGURE 13.10 Word frequency table from the training survey file.

From the main Excel ribbon, select *Insert*, then select *Treemap*. The resulting visual of the word frequencies appear in Figure 13.11. It does not yield an actual word cloud but a very reasonable facsimile. It is a suitable pictorial representation of the most important words. This is an excellent approach to visualizing term frequencies in Excel.

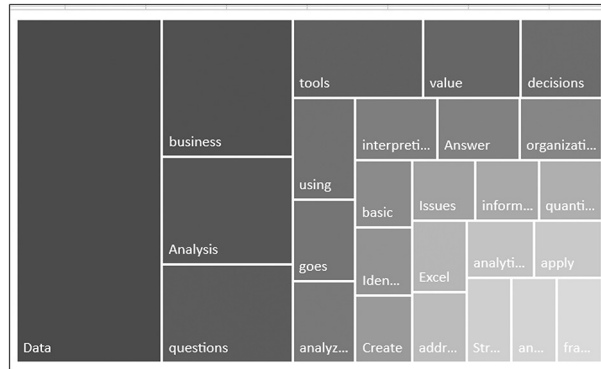


FIGURE 13.11 Training survey word cloud in Excel using a tree map.

VISUALIZING THE TEXT USING VOYANT

From the *Case Data* depository, in the *Chapter 13* folder, open the *Attendee PreSurvey Results Data.csv* in Excel, and copy the open-ended comments column into the computer buffer. Use a web browser with access to the internet. Load the Voyant text analysis program found at <https://voyant-tools.org>. You should see a screen similar to that in Figure 13.12.



FIGURE 13.12 Training survey word cloud of attendee comments using Voyant.

VISUALIZING THE TEXT USING R

From the *Case Data* depository, in the *Chapter 13* folder, open the *Attendee PreSurvey Results Data.csv* and save it as *casea.csv*. Use R and RStudio and install the packages we need using `Repository(CRAN)`:

```
dplyr, tidytext, word cloud
```

Import the library and read the data:

```
> library(dplyr)
> library(tidytext)
> casea <- read.csv(file.path("casea.csv"), stringsAsFactors = F)
```

Tokenize the contents of the data set and remove the stop words:

```
> tidy_a <- casea %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words)
```

Get the results of the word frequency analysis (shown in Figure 13.13):

```
> tidy_a %>%
  count(word, sort = TRUE)
```

word <chr>	n <int>
data	27
analysis	9
excel	6
learn	5
tools	5
interpreting	4
quantitative	4
analyze	3
decisions	3
understanding	3

1-10 of 109 rows

FIGURE 13.13 Word frequency data frame of the training survey.

Visualize the word frequency by word cloud (similar to that in Figure 13.14):

```
> library(wordcloud)
> pal = brewer.pal(8, "Dark2") # set up color parameter
> tidy_a %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 20, random.order = F,
  random.color = T, color = pal))
```



FIGURE 13.14 Word cloud of the training survey results.

TEXT SIMILARITY SCORING

In text data mining, we often encounter the situation where we wish to compare a *source document* (for example, a contract we are drawing up) to several previously existing contracts, called the *target documents*, to see which of the targets is most similar to the source. This could apply to other similar situations, for example: (1) a résumé as a source and many job descriptions as targets to see which jobs are most like the source résumé; (2) a job description as a source with many candidates' résumés as targets to see which candidate résumé is most similar to the job description; or (3) a source patent against the text of many similar patent targets to see which of the target patents is most similar to the source.

A popular way to compare two documents is by seeing how similar they are. The most popular and robust method is TF-IDF (Term Frequency–Inverse Document Frequency). It is easy to implement and yields very beneficial results in many circumstances. It is based on a Bag-of-Words approach and uses counts of the frequency of words for comparison, which is similar to what we have done earlier in the chapter, but here we carry it a bit further. It matches the frequency of words that are precisely the same or have the same root (such as “clean,” “cleaning,” “cleans,” and “cleaned”) in the two documents. It does not perform well when we want to associate two words in the two documents with similar meanings (such as “cleans” and “scrub”). We refer to this more advanced type of word association as semantic similarity. Nonetheless TF-IDF does an excellent job for most business applications. The TF-IDF method is an excellent way to score a set of candidate résumés against a job description, for example. The algorithm scores the résumés and returns an ordered list sorted by the résumés most similar to the job description. You can also score a résumé against a group of possible jobs to see which ones are more similar to the résumé.

WHAT IS TEXT SIMILARITY SCORING?

This is an elementary explanation of the TF-IDF algorithm with a cosine similarity score. Take, for example, these three texts:

- Most mornings, I like to go out for a run.
- Running is an excellent exercise for the brain.
- The lead runner broke away from the pack early in the race.

We want to compare these statements against this one-sentence document:

- The sergeant led the platoon in their daily run early in the day.

Which of the three texts above is most similar to the fourth text? Can we produce a ranked order list? The three sentences are the target, and the fourth is our source. In the first step, the algorithm extracts all the terms and produces a Bag-of-Words for each (as we did earlier in the chapter), as shown in Figure 13.15.

Text A	Text B	Text C		Source
Most	Running	The		The
mornings	is	lead		sargeant
I	great	runner		led
like	exercsie	broke		the
to	for	away		platoon
go	the	from		fin
out	brain	the		their
for		pack		daily
a		early		run
run		in		early
		the		in
		race		the
				day

FIGURE 13.15 The three target texts and the source document are sorted into a Bag-of-Words.

In the next step, the algorithm removes all the stop words (such as “I,” “to,” and “a”). Then it tokenizes and lemmatizes all terms (*run* and *runner* get converted to *run*). The TF is computed next (essentially, it performs a word frequency analysis). If some words are too frequent, they may not be too interesting (like the word “lawyer” in contracts: we all know they will be there, so they are commonplace and should be downplayed). The algorithm downplays them by using the inverse of the frequency (the IDF part). We are left with lists of words and their inverse frequencies. Now we compare the list of words and their score to see if they have words in common and compute a standard score normalized to 1 (the *cosine similarity score*). For this set of documents, the score is shown in Figure 13.16.

TEXT	description	similarity_score
Text A	Most mornings I like to go out for a run.	0.099
Text C	The lead runner broke away from the pack early in the race.	0.091
Text B	Running is great exercsie for the brain.	0.083

FIGURE 13.16 Similarity scoring of the three target texts against the source text scored and sorted by the cosine similarity.

We used a web-based tool called Simi Bot for this demonstration in the exercises that follow. Simi Bot may be found at <https://wukunchen.shinyapps.io/SimiBot/>. The scores it provides are pretty low, but even so, the ordering of the texts is uncannily accurate. A sergeant taking the platoon for a morning run is most similar to me going out for my morning run when compared to other choices.

EXERCISE 13.4 – CASE STUDY USING THE OCCUPATION DESCRIPTION DATA SET

Analysis Using an Online Text Similarity Scoring Tool

Simi Bot, the online scoring tool, requires two data files. The first is a simple (UFT-8) text data version of the source file. It could be a résumé, a job description, a contract, or a source text file. It must be a text version of the document. It can be called *Source*, but the name is not critical; its text format is essential.

We converted a job applicant’s résumé (Dr. Andres Fortino) into a text data file as an exemplar. It is called “resume” and we saved it as a UTF-8 text file. Any text-based credential would do as well (e.g., a LinkedIn profile or a curriculum vitae converted to UTF-8 text).

The target file is a simple Excel flat file exported into the CSV file format with job *titles* in the first column and the text of job *descriptions* in the second. The first row should have column

titles. Additional information can be added in separate columns (such as the company and location). Still, the tool will only use the text's column labeled “description” to compare to the exemplar. That column title must be the variable name for the rows of data to be used for comparison. It preserves the additional information columns in the output document. It is essential to follow the instructions in the tool for creating the target file.

As our exemplar target file, we used the text file *O*NET.csv* with 1,100 jobs downloaded from the Bureau of Labor Statistics O*NET database (ONET 2021). A copy of *O*NET.csv* may be found in the *Chapter 13* folder in the *Case Data* depository. You can use this file against your résumé to see the career jobs your résumé is similar to. You can build a similar target CSV file from job descriptions downloaded from any job search engine (such as Monster.com, Indeed.com, or Glass Door).

We will use similarity scoring to answer the following questions:

*Which occupations from the O*NET database is this person most suited for based on their résumé as the source?*

Use a browser and find the online similarity scoring tool at <https://wukunchen.shinyapps.io/SimiBot/>. It may take a minute to set it up. You will see a data entry screen like that shown in Figure 13.17.

The screenshot shows the Simi Bot web application interface. At the top, there is a navigation bar with 'Home', 'Help', and 'Reset' options. The main content area is divided into three sections: 'Comparison Source', 'Comparison Target', and 'Configuration'. Each section has 'Upload' and 'Paste' buttons. The 'Comparison Source' section is for uploading a .txt file, and the 'Comparison Target' section is for uploading a .csv file. The 'Configuration' section has a dropdown menu for 'Number of Word Combinations (n-gram)' set to '1-gram'.

FIGURE 13.17 Similarity scoring tool data entry screen found at <https://wukunchen.shinyapps.io/SimiBot/>.

Use the *resume.txt* file found in the *Case Data* repository, *Chapter 13* folder for the source text file. The Target CSV table uses the *O*NET JOBS.csv* file in the *Case Data* repository, *Chapter 13* folder. Press the *Analyze Data* button. A table such as that shown in Figure 13.18 should appear.

The occupations are sorted by similarity score to the résumé. Note that the top ten returned occupations fit with the information on the résumé. Scroll down to the bottom of the table and note that the least similar jobs do not apply to this résumé (see Figure 13.19.) Use the CSV button at the top of the display to obtain a copy of the table in CSV format.

Home Your source belongs to Cluster Group 1 Clustering Result

Scoring Result Clustering Result Search: _____

Show Top 100 Show All Rows Copy CSV Excel PDF

Help

Title	Description	Similarity Score	Cluster Group
Education, Training, and Library Workers, All Other	All education, training, and library workers not listed separately.	21.50%	2
Vocational Education Teachers, Postsecondary	Teach or instruct vocational or occupational subjects at the postsecondary level (but at less than the baccalaureate) to students who have graduated or left high school. Includes correspondence school, industrial, and commercial instructors; and adult education teachers and instructors who prepare persons to operate industrial machinery and equipment and transportation and communications equipment. Teaching may take place in public or private schools whose primary business is education or in a school associated with an organization whose primary business is other than education.	21.40%	1
Training and Development Specialists	Design and conduct training and development programs to improve individual and organizational performance. May analyze training needs.	21.26%	1
Education Teachers, Postsecondary	Teach courses pertaining to education, such as counseling, curriculum, guidance, instruction, teacher education, and teaching English as a second language. Includes both teachers primarily engaged in teaching and those who do a combination	17.16%	1

FIGURE 13.18 Similarity scoring of a résumé versus O*NET occupation data.

Ship and Boat Captains	Command vessels in oceans, bays, lakes, rivers, or coastal waters.	0.00%	1
Mates- Ship, Boat, and Barge	Supervise or coordinate activities of crew aboard ships, boats, barges, or dredges.	0.00%	1
Parking Lot Attendants	Park vehicles or issue tickets for customers in a parking lot or garage. May collect fee.	0.00%	1
Freight and Cargo Inspectors	Inspect the handling, storage, and stowing of freight and cargoes.	0.00%	1
Crane and Tower Operators	Operate mechanical boom and cable or tower and cable equipment to lift and move materials, machines, or products in many directions.	0.00%	1
Excavating and Loading Machine and Dragline Operators	Operate or tend machinery equipped with scoops, shovels, or buckets, to excavate and load loose materials.	0.00%	1
Machine Feeders and Offbearers	Feed materials into or remove materials from machines or equipment that is automatic or tended by other workers.	0.00%	1
Wellhead Pumps	Operate power pumps and auxiliary equipment to produce flow of oil or gas from wells in oil field.	0.00%	1

Showing 1,101 to 1,110 of 1,110 entries Previous 1 ... 52 53 54 55 56 Next

FIGURE 13.19 The bottom occupations returned by the similarity scoring tool.

SIMILARITY SCORING ANALYSIS USING R

EXERCISE 13.5 - RÉSUMÉ AND JOB DESCRIPTIONS SIMILARLY SCORING USING R

In the *Case Data* depository *Chapter 13* file folder copy *O*NET JOBS.csv* and name the copy *cased.csv*. The résumé's text referenced in the exercise may also be found in that data repository. Use R and RStudio to install the packages we need using Repository (CRAN):

```
dplyr, tidytext, textstem, readr, text2vec, stringr
```

Import the library and read the case data:

```
> library(dplyr)
> library(tidytext)
```

```

> library(text2vec)
> library(readr)
> library(stringr)

> cased <- read.csv(file.path("cased.csv"), stringsAsFactors = F)
> resume_f <- read_file("resume.txt")

# make resume content a dataframe
> resume_fdf <- tibble(job = "Fortino", description= resume_f)

# combine resume and job description
> case_d_resume <- rbind(resume_fdf,cased)

# data cleaning function
  > prep_fun = function(x) {
    # make text lower case
    x = str_to_lower(x)
    # remove non-alphanumeric symbols
    x = str_replace_all(x, "[^[:alnum:]]", " ")
    # collapse multiple spaces
    str_replace_all(x, "\\s+", " ")
  }

```

The cleaned résumé document is shown in Figure 13.20.

description_clean		
<chr>		
andrés guillermo fortino pe phd 75 grist mill lane pleasant valley ny 12569 agfortino gmail com 845 242 7614 educ...		
chief executives determine and formulate policies and provide overall direction of companies or private and public s...		
chief sustainability officers communicate and coordinate with management shareholders customers and employees t...		
general and operations managers plan direct or coordinate the operations of public or private sector organizations d...		
legislators develop introduce or enact laws and statutes at the local tribal state or federal level includes only workers...		
advertising and promotions managers plan direct or coordinate advertising policies and programs or produce collat...		
green marketers create and implement methods to market green products and services		
marketing managers plan direct or coordinate marketing policies and programs such as determining the demand for...		
sales managers plan direct or coordinate the actual distribution or movement of a product or service to the custome...		
public relations and fundraising managers plan direct or coordinate activities designed to create or maintain a favor...		

1-10 of 1,111 rows | 3-3 of 3 columns

Previous 2 3 4 5 6 ... 100 Next

FIGURE 13.20 Job description column after data cleaning.

```

# clean the job description data and create a new column
  > case_d_resume$description_clean = prep_fun(case_d_
    resume$description)

# use vocabulary_based vectorization
  > it_resume = itoken(case_d_resume$description_clean,
    progressbar = FALSE)
  > v_resume = create_vocabulary(it_resume)
  > v_resume = prune_vocabulary(v_resume, doc_proportion_max =
    0.1, term_count_min = 5)
  > vectorizer_resume = vocab_vectorizer(v_resume)

# apply TF-IDF transformation

```

```
➤ dtm_resume = create_dtm(it_resume, vectorizer_resume)
➤ tfidf = TfIdf$new()
➤ dtm_tfidf_resume = fit_transform(dtm_resume, tfidf)
```

The results of the computed cosine similarity are shown in Figure 13.21.

```
# compute similarity score against each row
➤ resume_tfidf_cos_sim = sim2(x = dtm_tfidf_resume, method = "cosine", norm = "l2")
➤ resume_tfidf_cos_sim[1:5,1:5]

# create a new column for similarity_score of data frame
➤ case_d_resume["similarity_score"] = resume_tfidf_cos_sim[1:1111]

# sort the dataframe by similarity score
➤ case_d_resume[order(-case_d_resume$similarity_score),]
```

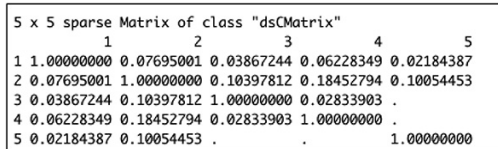


FIGURE 13.21 Cosine similarity score against each row (job).

The results of the cosine similarity of the jobs against the résumé ordered by similarity score are shown in Figure 13.22.

job	similarity_score
Fortino	1.0000000000
Training and Development Specialists	0.2601687893
Education, Training, and Library Workers, All Other	0.2449726919
Training and Development Managers	0.2433782818
Education Teachers, Postsecondary	0.1786205110
Vocational Education Teachers, Postsecondary	0.1690794638
Adult Basic and Secondary Education and Literacy Teachers and Instructors	0.1594161953
Industrial-Organizational Psychologists	0.1546035291
Special Education Teachers, All Other	0.1543412842
Health Educators	0.1496801049

FIGURE 13.22 Jobs against résumé data ordered by the similarity score.

CASE STUDY 13.1 – TERM FREQUENCY ANALYSIS OF PRODUCT REVIEWS

You work for a product manufacturer and have received a file with customer feedback on your product. You want to learn what your customers are saying about the product and decide to do a term frequency analysis on the text of their comments.

Let’s use the term frequency analysis to answer the following question:

What are the most frequent words that best represent what customers say about the product?

TERM FREQUENCY ANALYSIS USING VOYANT

Access the Case Data repository *Chapter 13* file folder and open the *Product Reviews.csv* with Excel.

Copy the contents of the column *reviews.text* to your computer buffer. Open the Voyant tool using the web-based version (<https://voyant-tools.org/>). Paste the customer comments in the data entry box in Voyant and press *Reveal*. In the upper left-hand corner panel, click on *Terms* to switch from the word cloud mode to the table mode. The resulting term frequency list sorted by most frequent words should look something like that in Figure 13.23. Figure 13.24 shows the word cloud for the exact text.

			Term	Count
<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	mop	2670
<input checked="" type="checkbox"/>	<input type="checkbox"/>	2	use	1210
<input checked="" type="checkbox"/>	<input type="checkbox"/>	3	spray	1042
<input checked="" type="checkbox"/>	<input type="checkbox"/>	4	bottle	852
<input checked="" type="checkbox"/>	<input type="checkbox"/>	5	product	802
<input checked="" type="checkbox"/>	<input type="checkbox"/>	6	love	739
<input checked="" type="checkbox"/>	<input type="checkbox"/>	7	bought	665
<input checked="" type="checkbox"/>	<input type="checkbox"/>	8	great	652
<input checked="" type="checkbox"/>	<input type="checkbox"/>	9	just	607
<input checked="" type="checkbox"/>	<input type="checkbox"/>	10	cleaning	605
<input checked="" type="checkbox"/>	<input type="checkbox"/>	11	handle	532
<input checked="" type="checkbox"/>	<input type="checkbox"/>	12	used	532
<input checked="" type="checkbox"/>	<input type="checkbox"/>	13	like	525
<input checked="" type="checkbox"/>	<input type="checkbox"/>	14	trigger	493
<input checked="" type="checkbox"/>	<input type="checkbox"/>	15	reveal	487
<input checked="" type="checkbox"/>	<input type="checkbox"/>	16	floor	474
<input checked="" type="checkbox"/>	<input type="checkbox"/>	17	clean	466
<input checked="" type="checkbox"/>	<input type="checkbox"/>	18	broke	459

FIGURE 13.23 Term frequency results using Voyant for the Rubbermaid mop customer reviews.



FIGURE 13.24 The Voyant word cloud for the Rubbermaid mop customer comments.

TERM FREQUENCY ANALYSIS USING R

In the *Case Data* depository *Chapter 13* file folder, copy and rename the *Product Reviews.csv* as *casec.csv*.

Install the packages we need using `Repository(CRAN)` Into and R session using RStudio:

```
dplyr, tidytext
```

Import the library and read the data:

```
> library(dplyr)
> library(tidytext)

> casec <- read.csv(file.path("casec.csv"), stringsAsFactors = F)

# concatenate reviews text and reviews title
> casec <- casec %>%
  unite(review_combined, reviews.text, review.title, sep = " ", remove
= FALSE)
```

Tokenize the contents of the data set and remove the stop word:

```
> tidy_c <- casec %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words)
```

Get the results of the word frequency analysis (shown in Figure 13.25):

```
> tidy_c %>%
  count(word, sort = TRUE)
```

word <chr>	n <int>
mop	3103
spray	1201
product	990
love	935
bottle	880
bought	671
cleaning	646
reveal	640
trigger	582
handle	565

1-10 of 11 rows

FIGURE 13.25 The term frequency data frame of product reviews computed using R.

These are the most frequent significant words employed by customers to complain about the product. Correctly interpreted, they should give the analyst some indication of the themes or throughlines in the stories customers use to describe their negative experiences (or complaints) with the product.

REFERENCES

- Fortino, Andres G. 2020. *Data Visualization for Business Decisions: A Case Study Approach*. Dulles, VA: Mercury Learning and Information.
- Fortino, Andres G. 2021. *Text Analytics for Business Decisions: A Case Study Approach*. Dulles, VA: Mercury Learning and Information.
- O*NET OnLine Help: Find Occupations.” *O*NET OnLine*, National Center for O*NET Development, www.onetonline.org/help/onlinefind_occ.
- Silge, J., & Robinson, D. (2016). tidytext: Text mining and analysis using tidy data principles in R. *Journal of Open Source Software*, 1(3), 37.
- Sinclair, Stéfan, and Geoffrey Rockwell. 2016. Voyant Tools. 2016. <http://voyant-tools.org/>

WORKING WITH LARGE DATA SETS



This chapter presents techniques applicable when dealing with data sets too large to load into working computer memory. This problem is particularly acute when using Excel for analysis, where there is a single worksheet limitation for loading a data table. Even if the data file fits into an Excel worksheet if it has a large number of rows (>750,000) and a large number of variables (>50), any computation with the table will often take a long time in Excel. Both R and Python also experience this problem when the data set is so voluminous that it exceeds the working memory capacity of a computer. One helpful way to overcome this problem is to randomly sample the “too-big-to-fit-into-memory” table and analyze the sampled table composed of the sampled rows. Excel has a randomization function, which we could use to extract the sample rows. The problem with that approach is that we cannot get the entire table into Excel. So, we must use a different tool to perform the sampling. We will do this with the R program. There is an exercise in this chapter where you are guided through how to set up and use R to extract a meaningful sample of rows for a large data set. We also show how to compute how many rows your sample needs to obtain statistically significant results when using the table of samples. Once the sample rows are extracted, any previously demonstrated tools may be used to get answers using the skills taught in earlier chapters. This technique answers the following business question:

How do we work with data sets too large to load into Excel?

As in previous chapters, we demonstrate the technique in the first exercise and allow for more challenging work in subsequent exercises.

USING SAMPLING TO WORK WITH LARGE DATA FILES

This exercise's premise is that we wish to use Excel as our analysis tool, but are aware of its limitations concerning large files. Typically, the problem is not that there are too many variables, but too many rows. Let's say we have a vast data file of hundreds of megabytes consisting of hundreds of thousands (or perhaps millions) of rows. How do we use Excel when we cannot load the entire file in a spreadsheet? The answer is to make a tradeoff. We are willing to accept a slight decrease in accuracy in our statistical results for the convenience of using Excel for the analysis.

EXERCISE 14.1 - BIG DATA ANALYSIS

The technique is to randomly sample the large (or big data) file and obtain a random sample of manageable rows of data. We first use one tool to compute an adequate sample size, and then we use another tool to sample the original file. We use a free web-based tool to compute sample size, and then we use RStudio to extract a random sample.

The data files for this exercise (as listed in Figure 14.1) may be found in the *Case Data* repository, under the *Chapter 14* folder. First, let's compute an adequate sample size. The entire file is our population. For example, we wish to have 95% confidence in our statistical analysis using our sample and no more than a 1% margin of error in our results (these are very typical parameters in business). Take the 306 MB *BankComplaints.csv* big data file with 753,324 rows (see Figure 14.1). Using the online sample size calculator found at <https://www.surveymonkey.com/mp/sample-size-calculator/>, we need a random sample of 9,484 rows to achieve our desired level of accuracy and margin of error (Figure 14.2).

Name	Size (MB)	Rows	Columns	Source	Description
ORDERS.csv	1.8	8,400	22	Company	Office supplies orders
Community.csv	70	376,000	551	US Census	2013 ACS census file
Courses.csv	73	631,139	21	MIT	edX 2013 MOOC Courses
BankComplaints.csv	306	753,324	18	US FTC	Bank complaints to the FTC

FIGURE 14.1 Characteristics of the data files used to demonstrate the sampling of large data sets.

FIGURE 14.2 Using an online sample size calculator to reduce the 306 MB *Bank Complaints* file to a manageable set of rows will yield significant results.

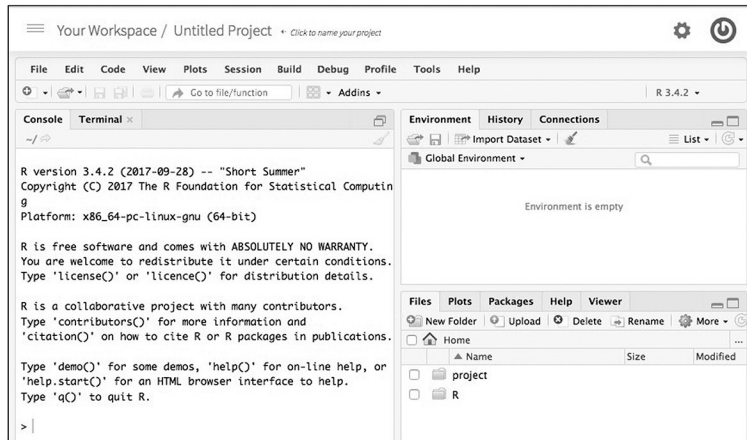
As an additional exercise, use the online calculator to compute the necessary number of random rows in the other sample files for various accuracy levels in Table 14.1. Note that the rightmost column has the answer.

We now use a popular, free version of the R program called RStudio. Load RStudio. Download and install it, if you haven't yet. In RStudio, create a new project. The typical RStudio interface appears. Note the ">_" prompt in the lower-left-hand corner of the left screen. It should be

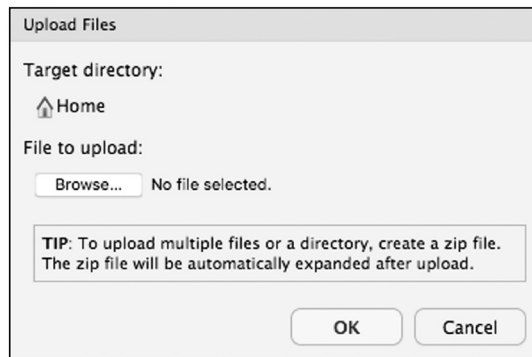
Table 14.1 Computed elements of the sampling of the data sets

Name	Size(MB)	Population Rows	Confidence Level %	Margin of Error %	Random Sample Rows
ORDERS.csv	1.8	8,400	95	1	4,482
Community.csv	70	376,000	95	1	9,365
Community.csv	70	376,000	99	1	15,936
Courses.csv	73	631,139	95	1	9,461
Courses.csv	73	631,139	95	2	2,394

blinking and waiting for your R commands. The resulting screen in your browser should look like Figure 14.3.

**FIGURE 14.3** Interface screen of RStudio Cloud.

First, we upload all the files we are sampling. Using the *Case Data* repository; under the *Chapter 14* folder, find the files *ORDERS.csv*, *Courses.csv*, and *Community.csv* files. Click on the *Files* tab in your browser's lower-right-hand pane of the RStudio desktop. Then, click *Upload* in the new row. You will get the interface shown in Figure 14.4.

**FIGURE 14.4** The RStudio Cloud tool used to upload files to the web for analysis.

Click the *Browse* button and upload each of the three files. Be patient, as some larger files take some time to upload. When done, the *File* area in the upper-right-hand screen should be like that shown in Figure 14.5.

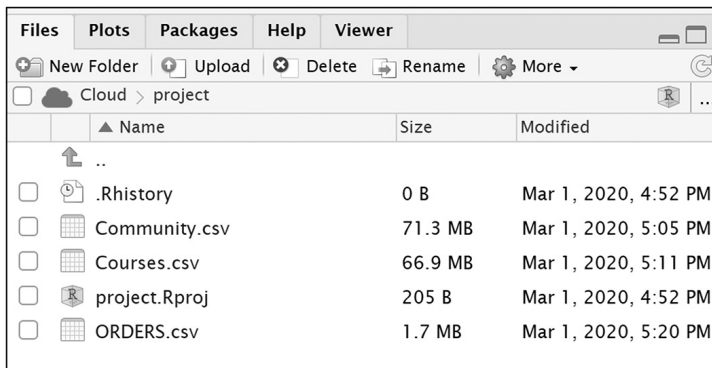


FIGURE 14.5 Screen of the uploaded data files ready to be processed with an R script.

We start by sampling the smaller file (*ORDERS*) and then move on to the larger files. In the upper-left-hand panel, pull down the *File > Open* function and select the *ORDERS.csv* file from the list. That loads the file into the workspace (note that the *Source* panel now appears and has information about the file). Drop down to the lower-left-hand panel and click in front of the “>_” cursor. It should start blinking, ready for your command.

Enter the following sets of commands:

```
> set.seed(123)
> Y <- read.csv("ORDERS.csvv")
> View(Y)
> index <- sample(1:nrow(Y), 4482)
> Z <- Y[index, ]
> View(Z)
> write.csv(Z, 'Z.csv')
```

Enter the random number of rows required (4482), but without a comma, or the command will be interpreted with 4482 as a part of the command and not as part of the number. We are using *Y* and *Z* as temporary containers for our data. Note that the *Source* upper-left-hand panel shows the original data in table form (the result of the *View* command).

Also, note that the upper-right-hand panel shows two files in the workspace, *Y* and *Z*, and their characteristics. Note that *Y* has the original set of rows, 8,399, and *Z* has the sample rows, 4,482. Random sampling was done with the *sample* command.

We outputted the sample rows to the *Z* file, and the program wrote it out to the disk as *Z.csv*. The lower-right-hand panel has that file in the directory (Figure 14.6).

Now, we need to download the file from the cloud directory to our computer. You should check the box next to the *Z.csv* file. In the lower-right-hand panel, click on the *More* icon (it looks like a blue gear). Select *Export*, and follow the directions to download the file to your

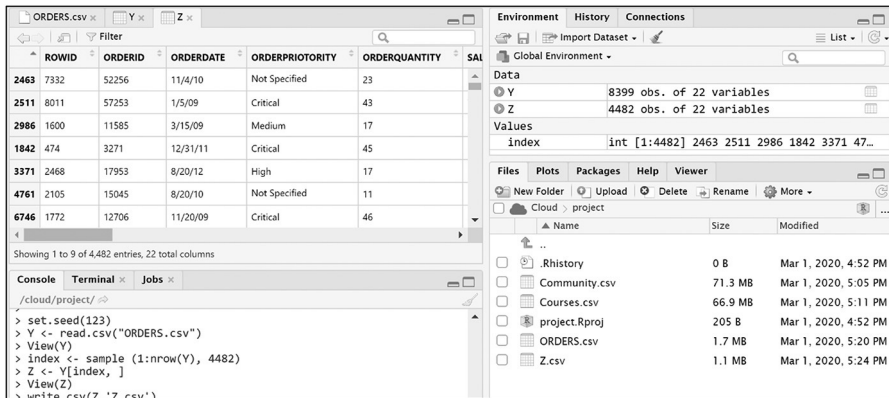


FIGURE 14.6 RStudio Cloud interface screen showing the data file (upper left), R script (lower left), details of the input and output data files (upper right), and files in a directory (lower right).

desktop for now. Rename the file to *ORDERSSample.csv* as you save it. (It is important to note that we only used *Y* and *Z* as temporary, easy-to-use containers.)

To check our work, we compute some results using the original population and the sample rows and compare them.

Open *ORDERS.csv* and *ORDERSSample.csv* in Excel. Notice that the sample data set contains a new column (at the extreme left) that identifies each sample row uniquely (a random number). It would help if you labeled that column (for example, *SAMPLEID*).

Using pivot tables, tabulate the total sales by region for both files. Compare the results from both tables (Figure 14.7). Compute the difference between the total population and the sample. You will find it well within the 5% margin of error.

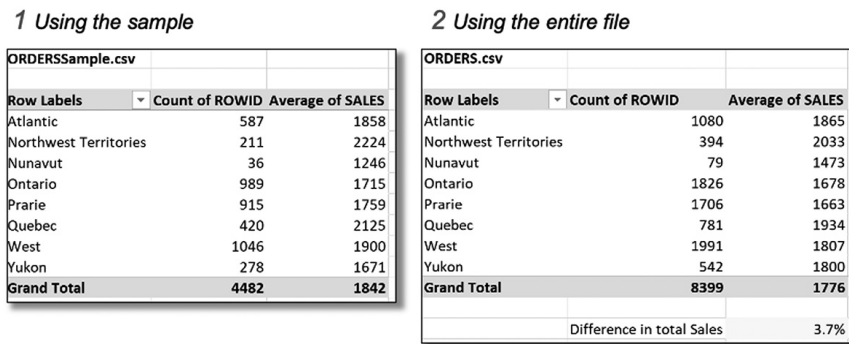


FIGURE 14.7 Comparison of the same analysis using the entire file and the sample, showing less than a 5% error difference.

Note that the computed total from the sample file is accurate compared to the original file's computed total. However, there is a much broader error in the individual regional results, especially for regions with fewer rows. If you repeat for the *PROFIT* variable rather than *SALES*, you will see a much wider variation. Repeat these steps using the two other data files as additional exercises.

Repeat the process for the *Community.csv* and *Courses.csv* files for a 95% confidence level and a 2% margin of error. Compute the summary of one of the variables for both the total population and the sampled files and compare the results.

CASE STUDY 14.1 USING THE BANKCOMPLAINTS BIG DATA FILE

Load the *BankComplaints.csv* 306 MB file in RStudio.

Using an online sample size calculator mentioned in Exercise 14.1, compute the number of random rows to select an adequate sample with a 95% confidence level and a 1% margin of error (Table 14.2).

Table 14.2 Computed elements of the sampling of the data sets

Name	Size (MB)	Population Rows	Confidence Level %	Margin of Error %	Random Sample Rows
BankComplaints.csv	306	753,324	95	1	4,484

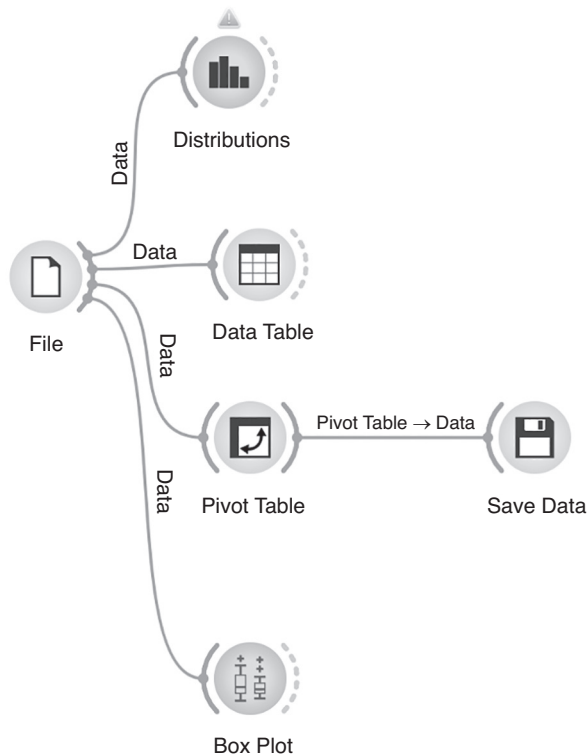
Use the R commands given earlier to sample the file and save it as *BankComplaintsSample.csv*. (Make sure to use the correct file name in the commands.)

Use the file of samples to tabulate the percentage of complaints by the state to discover the states with the most and least complaints.

Use the *US Pop by State.xlsx* file in the *Chapter 14* folder in the *Case Data* depository. Add the size of the population of each state and normalize the complaints per million residents of each state to the *Complaints* file. Determine the states with the least and the most complaints per capita. Compute other descriptive statistics of this variable. Use Excel and get a summary of the descriptive statistics (Figure 14.8). Determine from the resulting table which states have the most and the least complaints per capita.

COMPLAINTS/MPERSON	
Mean	24.46
Standard Error	1.45
Median	24.25
Mode	#N/A
Standard Deviation	10.24
Sample Variance	104.91
Kurtosis	-0.53
Skewness	0.43
Range	41.82
Minimum	6.76
Maximum	48.58
Sum	1222.99
Count	50.00

FIGURE 14.8 Descriptive statistics of the sample extracted from the *BankComplaints.csv* data file.

VISUAL PROGRAMMING

This chapter is a brief introduction to visual programming. *Visual programming* is a technique that has been around for some time and is used here to create computer models without programming. Each of the significant commercial environments presented produces programs in their proprietary code. We will see that with the open-source product we present below, Orange 3, the code it produces is in Python. We use Orange 3

throughout this book for our case studies as a way to create models in Python without any command-line coding. It is a complementary approach to command-line coding and quickly generates sophisticated analytical models which can be further developed with additional coding. Visual programming makes model-making in Python accessible to analysts with modest Python coding skills.

Visual programming is a language (more like an environment) that does not need coding or remembering functions, function names, or libraries, and the rules for invoking them within a script. The analyst moves modules around, embeds them visually on a design canvas, and connects them to each other to perform data analysis. It is akin to constructing a puzzle with interlocking pieces until the picture emerges.

Today, analysts are confronted with visual programming everywhere because software development shops use many visual programming tools. Visual programming does not obviate the need to know how to program in Python. Knowing how to program at the command line by coding scripts in Python is essential because it is fundamental and is where analysts have the most significant control. In many cases, command-line programming is how analysts first start their analysis. However, in large organizations, applications are often built by programmers working with visual programming techniques that ultimately become the code for the final product.

We have seen visual coding in Chapter 13, where we used Voyant for text data mining. That tool was a visual programming tool of sorts. Voyant accepted data input using a prompted interface. Once data is ingested and adequately configured by selecting the correct module from a list, the tool executes some programming functions and produces a result. The Voyant tool has limited configuration functionality for each module, and the analyst does not have much control over how the tool performs the analysis. In past chapters, when we used JASP or Jamovi as the R front-end with pulldown menus, the functions called upon were adjusted visually via menu choices. We can think of JASP or Jamovi as visual programming tools with a primitive functional interface. We might consider them as front ends to a programming language. Often, these front ends will generate the code in R from the visually configured analysis process. The R code produced by the front-end tool is a starting point for building a more sophisticated product. Again, as with Voyant, the analyst has little control of the embedded functions in these front-end tools, only what the front-end designers have programmed into the pulldown functions of the tool.

Actual visual programming is a more robust environment that goes beyond the functionality of these front-end tools. With truly visual programming tools, the analyst has much more control in building a series of connected modules, called *widgets*, to perform an analysis task without having to write code at the command line. The analyst concatenates the widgets, connecting them to perform some complex analysis tasks. The analyst has great control, since each widget can be individually configured. All “programming” is done visually, rather than by entering code at the command line or creating a script. Many of these visual programming environments are sophisticated and expensive high-end platforms offered by well-known commercial software developers, such as IBM and SAS.

There are open-source versions of visual programming tools, such as Orange 3, which, like JASP and Jamovi, are free and very robust. Orange 3 is also a perfect front end to Python and works well for the beginner analyst to learn visual programming. Orange 3 also generates good enough basic Python code as a starting point in building a Python-based model. When an analyst’s

supervisor wants them to do an analysis project using Python, creating the model visually in Orange 3 can be a good starting point. A prototype can be quickly built. Then the program's resulting Python code becomes a starting point for building a more robust enterprise-wide analysis product that can be optimized, modified, and maintained.

COMPARING VISUAL PROGRAMMING TO COMMAND-LINE CODING

Figure 15.1 shows the difference in data analysis with the results of visual programming on the left versus Python script coding at the command line on the right. On the left, we see the visual programming environment using widgets. Each widget is a Python script that performs some pre-determined Python function. The widgets are connected, so data flows from one widget to another. The analysis results are transferred from widget to widget as data flows through and is processed by the Python code. The connection between widgets is made by creating a visual link between widgets in the design canvas and feeding the data through these components. The analysis is typically arranged to progress from left to right. Data ingestion, or input, occurs by the widgets on the left of the diagram. The data is then processed through various analysis widgets in the center. Finally, the results are piped to output visualization, or reporting, widgets on the right. The widgets on the right of the canvas write output files with the analysis results or produce graphs that can be displayed.

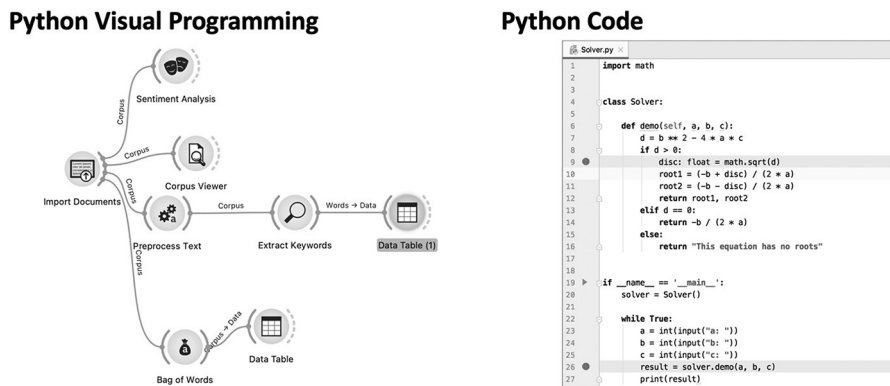


FIGURE 15.1 Visual programming versus command-line coding in Python.

As you see, the design of the analysis process is visual, which is much more functional and accessible than writing the code shown on the right side of Figure 15.1. Visual programming leaves much out because it differs from direct coding, where the analyst has much more control. However, visual programs are more accessible for others to follow, helping others to see the analysis flow in a better way than a coded program. Visual programming has many compelling characteristics: it is easier to program, faster to generate, more accessible to document, and the models get completed much sooner than with straight coding.

LEADING VISUAL PROGRAMMING ENVIRONMENTS

Let's see what some of the leading visual programming environments are. They are the offerings of major software vendors. The Forrester organization is an industry analysis and

reporting company that researches trends in the IT industry and publishes reports on their findings. They produce comprehensive reports on the state of the IT industry. They report on all forms of information technology, like computers and communications, and programming languages. They produce deep research and detailed industry reports for technology adoption companies to track the top commercial products they may consider acquiring and using. The Forrester Organization is a leading industry analyst in the IT industry. Among many IT areas of analysis, they watch for developments in the area of AI and machine learning. A critical analysis tool unique to Forrester is the Forrester Wave™ plot. It displays the significant offerings of the technology sector. It scores the technology products along two dimensions: the strength of any company's offering and the strength of the company strategy in that sector. They recently produced an analysis of the leading visual programming environments (Gualtieri 2020). Their analysis shows that SAS, IBM, and RapidMiner offer the three leading commercial products in visual programming. The open-source product Orange 3, which we have been using, is not included by Forrester in this wave chart, but we also consider it a fourth important product.

VISUAL PROGRAMMING WITH THE SAS ENTERPRISE GUIDE

According to Forrester, the leading contender in visual programming is the development environment offered by SAS. SAS is a leading IT software company. SAS offers two powerful analytic products, the ad hoc analytics product JMP and the visual programming environment Enterprise Guide (EG for short) (Parr-Rudd 2014). The analytic product JMP has pull-down menu functions similar to Jamovi and JASP. It has a proprietary script language for generating reusable code. JMP is a visual tool because it has pull-down menus to invoke analysis functions. However, SAS EG is their more robust visual programming environment for building sophisticated enterprise-wide data analytic products. EG is much more visual than JMP and can be programmed in a different proprietary language.

Figure 15.2 shows a Windows computer's SAS EG Data Miner visual programming environment. The SAS EG software allows the analyst to call for a widget dragged onto the design canvas in the upper right-hand panel, as shown. Access to the data sources is available through the upper left-hand panel. The lower left-hand panel has access to all the available EG widgets. The lower right-hand panel shows the SAS EG code created from the visually displayed analysis diagram shown by the interconnect widgets. Program logic flows from left to right (on the code panel, the logic is in the traditional top-to-bottom flow). Data ingestion modules are on the left of the diagram; then, they are piped into data shaping modules. The data is then piped via connection links to analysis widgets in the center that perform the various types of analysis. The results are piped to output modules that either write them to a file or document or display them as a chart or visual on the right. The analyst does not have to write any code; she connects all these pieces of data through the input widget to the data processing widget and then to the output function widgets. A sophisticated analytics model may be created using only visual programming. The SAS EG code generated is displayed in the lower panel below the canvas. This generated SAS EG proprietary code becomes the basis for deploying the final product into an executable program (.exe) and conducting further development and maintenance of the program.

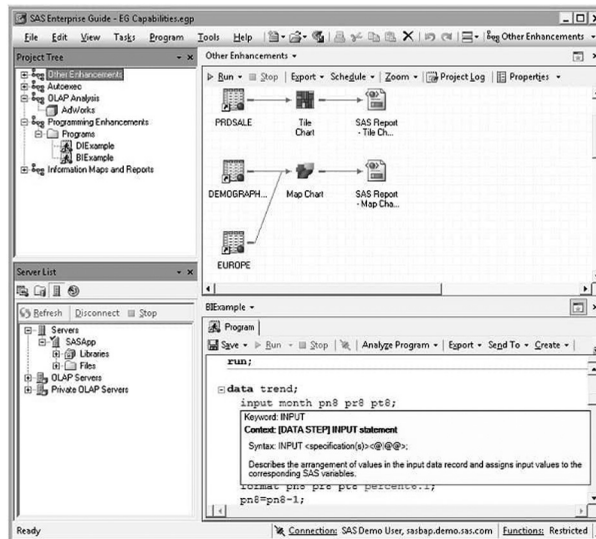


FIGURE 15.2 Visual Programming with SAS Enterprise Miner.

Let's consider a case study of how SAS EG Data Miner might be used. Let's say you work for an insurance company. They want a program developed that their claims adjusters can use to process insurance claims from their laptops when connected to the corporate servers. Typically, this software cannot be bought off-the-shelf. It is proprietary and based on company-specific policies and data assets. They hire programmers to develop this company-specific program. The company CTO selects the programming environment. Perhaps it is already in use (over 50% of the companies use SAS EG), or they decide to start using SAS Enterprise Guide just for this project. The programmers and analysts convert the functional specifications of the program into code. They would do it visually and then generate a program based on the SAS EG code generated by EG, which can be deployed as an executable on the company's servers and clients. In the future, the EG code would be maintained as code rather than going back to the visual programming version. Because of the extensive SAS EG installed base, there is a high demand for SAS EG programmers.

VISUAL PROGRAMMING WITH IBM SPSS

Another prominent software developer, IBM, also offers one of the top visual programming environments. It is based on their script-driven statistical analysis program SPSS (George and Mallery 2019). SPSS has been the primary statistical analysis software, offered by IBM, for many years. IBM recently developed a visual programming front end to SPSS that you see in Figure 15.3. It is similar to the SAS EG product, but has some structural differences. The same principles apply — access to data sources, a list of widgets, a design canvas, and a display of the SPSS script being created as the widgets are interconnected during design. Interestingly, the IBM environment also provides a way to document the elements of the analysis project in the architecture of CRISP-DM (notice the lower right panel).

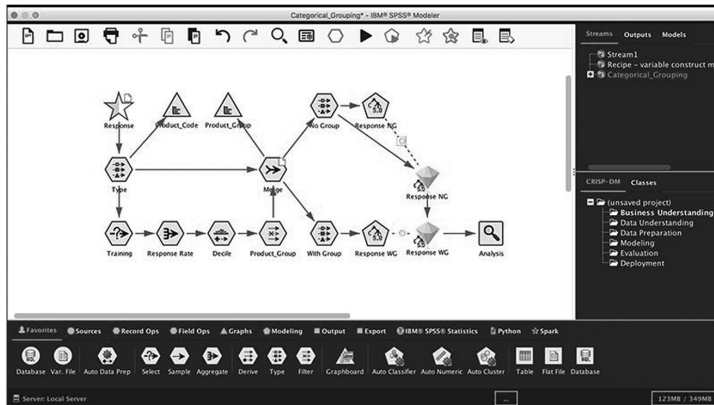


FIGURE 15.3 The visual programming environment of IBM SPSS.

VISUAL PROGRAMMING WITH RAPIDMINER

The Forrester analysis presents a third alternative for visual programming and one that is gaining popularity quickly — RapidMiner. These three (SAS EG, IBM SPSS, and RapidMiner) comprise the top data mining tools that can be programmed visually. RapidMiner only makes one product, a visual programming analytics platform (Kotu and Deshpande 2014). It is powerful, robust, and has many features. It does not generate Python or R code. Still, it has its proprietary programming language, similar to SAS EG or IBM SPSS, that may be used to build sophisticated analytic systems for data mining. As with any visual programming environment, the analyst would double-click the widget to configure it. A panel opens on the right to allow the configuration of the widget parameters. Figure 15.4 shows the widgets on a canvas and the configuration panel on the right for one of them. Figure 15.4 shows the RapidMiner design canvas with sources on the left panel, and listing of widgets in the lower left, the connection of all the widgets in the middle and then the configuration of the widgets on panels on the right. It is similar to configuring a pivot table in Excel, where the pivot table is in the middle, and the pivot table parameters appear on a panel on the right.

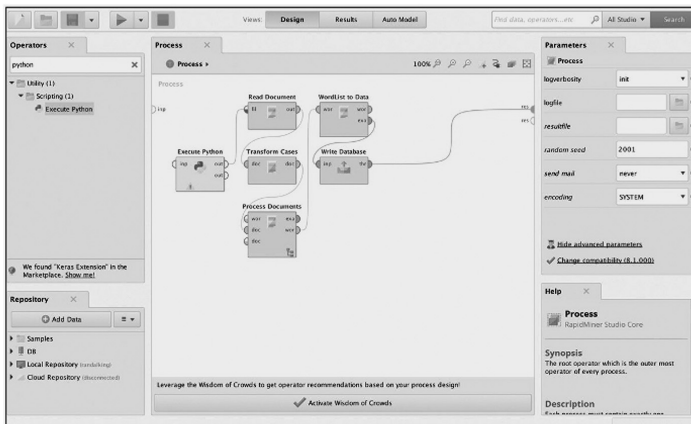


FIGURE 15.4 Visual programming with RapidMiner.

VISUAL PROGRAMMING WITH ORANGE 3

Orange 3 is a prevalent open-source visual programming environment (Demsar et al. 2013). It is not in the Forrester Wave because (a) there is not much of an installed base yet, and (b) it is not a commercial product. Most organizations frequently purchase commercially available products and avoid adopting open-source products when developing their analytic products. The commercial products have much better tech support, so they are a safer choice. On the other hand, the open-source platforms are excellent for research and teaching analytics in an academic environment, but businesses do not favor them. As seen in previous chapters where machine learning techniques were studied, we used Orange 3 as the visual programming front-end to Python. The code the Orange 3 produces is a Python script that can be used with any Python environment, such as a Jupiter notebook. When using Orange 3, we can genuinely say we are programming in Python. Orange 3 has many features and many widgets. It does a lot right out of the box and has many add-ons to provide excellent data mining functionality.

Figure 15.5 shows the Orange 3 design canvas. The interface is much more straightforward than the other commercial products discussed above. Orange 3 has two panels, the panel on the left is a listing of all widgets, and the center panel is the design canvas. To start designing any analysis process, drag an appropriate widget to the canvas and configure it. To configure a widget, it needs to be double-clicked. Then a superimposed configuration wizard appears, allowing adjustment for the widget parameters. Once a widget is placed on the canvas, it is connected to other widgets and configured. Data is ingested from the left, analyzed in the center of the diagram, and outputted to the right using the proper output widgets, just as we saw with the commercial products.

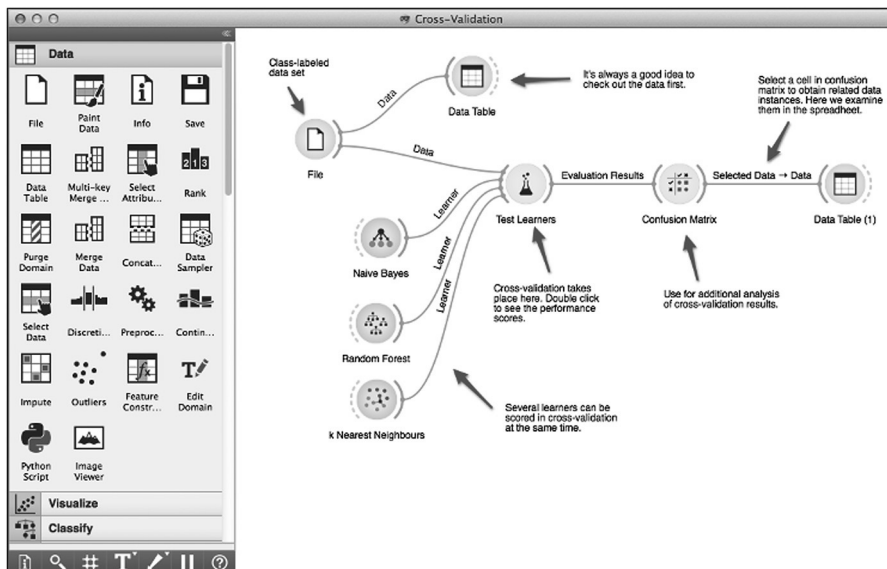


FIGURE 15.5 Orange 3 visual programming for Python.

Orange 3 can also generate Python code, which is easy to debug in its visual mode on the canvas. Once satisfied that the visual model works as intended, the code can be generated to run in a Python environment like a Jupyter notebook. The transferred code may be tweaked, added to, modified, and designed into a more permanent analytics product. In most cases, though, analytics requires straightforward ad hoc work: set up the problem in Orange 3, analyze it, obtain an answer, and complete the project.

We use Orange 3 extensively in this book to obtain answers, ostensibly in Python, but in reality, in a Python visual programming environment. This is much more convenient, especially for analysts with minimal or nonexistent Python skills.

INSTALLING ORANGE 3

To install Orange 3, go to the web page (<https://orangedatamining.com/download>) to download and install the program. Be careful to select the version for the correct operating system. There are Windows, Mac, and Linux versions. The Mac version has two versions since Apple offers operating systems for two CPUs, the Intel-based CPU and Apple's M1 chip. Be careful to select the correct one for your machine.

You may want to take advantage of the excellent YouTube tutorial videos Orange 3 makes available. You can quickly start using Orange 3 for many different types of analysis in a few minutes by following these tutorials. The videos are 5 or 10 minutes each, and within a few minutes, you will be up to speed using the widgets (Orange 2022).

REFERENCES

- Demšar, Janez, Tomaž Curk, Aleš Erjavec, Črt Gorup, Tomaž Hočevar, Mitar Milutinovič, Martin Možina et al. "Orange: data mining toolbox in Python." *the Journal of Machine Learning Research* 14, no. 1 (2013): 2349-2353.
- Gualtieri, Mike. 2020. "The Forrester Wave™: Multimodal Predictive Analytics and Machine Learning, Q3 2020." Forrester. 2020. https://www.forrester.com/report/the-forrester-wave-multimodal-predictive-analytics-and-machine-learning-q3-2020/RES157465?ref_search=0_1669902206550.
- George, Darren, and Paul Mallery. 2020. *IBM SPSS Statistics 26 Step by Step: A Simple Guide and Reference*. New York: Routledge.
- Kotu, Vijay, and Bala Deshpande. 2015. *Predictive Analytics and Data Mining: Concepts and Practice with Rapidminer*. Amsterdam: Elsevier/Morgan Kaufmann.
- "Orange Data Mining." n.d. YouTube. Orange 3. Accessed November 18, 2022. <https://www.youtube.com/channel/UCIKKWBe2SCAEyv7ZNGhIe4g>.
- Parr-Rud, Olivia. 2014. *Business Analytics Using SAS Enterprise Guide and SAS Enterprise Miner: A Beginner's Guide*. SAS Institute.

INDEX

A

Anomaly detection
 case study, 207
 by categorical variable, 211–212
 by clustering, 213–214
 definition, 205
 linear regression, 214–217
 old Faithful eruption, 206
 outliers, 206
 in airline delays data set, 208
 by normalization, 207
 in the SBA loans data set, 222–225
 in the SFO survey data set, 217–222
 by standardization, 207–208
 Tukey fences
 with single variables, 208–209
 vs. Z-scores, 209–210

B

Binning step, 43
Business analytics, 4–5
Business information need, 19

C

Classification
 decision tree
 annotated, 124
 build, 122–123
 problems, 128
 with Florence Nightingale data set, 137–139
 iris data set, 129–131

 characteristic parts, 126
 decision tree model, 127
 descriptive statistics, 126
 with Naïve Bayes, 131
 random forest model, 128–129
 with SBA loan data set, 136–137
 with SFO survey data set, 133–135
Clustering, 141
 applications of, 143
 hierarchical, 145–147
 Iris data set, 148–150
 to Old Faithful eruptions, 147–148
 income and debt variables, 142
 k-means clustering, 150
 cluster centroids, 151
 vs. hierarchical clustering, 153
 Iris data set, 152–153
 to Old Faithful eruptions, 152
 working principle, 150
 linear regression, 143–145
 Old Faithful eruptions, 142–143
 with SBO loans data set, 167–172
 with SFO survey data set, 154–166
Coefficient of determination, 95
Command-line coding, 261
Conditional probability, 132–133
Correlation between two variables, 97–98
Cross Industry Standard Process for Data Mining
 (CRISP-DM)
 business understanding phase, 12, 18
 data preparation phase, 13, 32

- data understanding phase, 12, 18
- definition, 18
- deployment phase, 13–14
- evaluation stage, 13
- framing analytical questions, 18
- modeling techniques, 13

D

- Data analysis, 3
 - business information need, 19
 - data-driven decision-making process, 20–21
- Data dictionary, 35
- Data frames, 31
- Data mining
 - algorithms and applications, 2, 5–6
 - analysis step, 10
 - analytics languages, 14–15
 - choice of, 15–16
 - business analytics and intelligence, 4–5
 - case study, 7–8
 - CRISP-DM, 12
 - data-driven decision-making process, 3–4
 - Data is the new oil, 2–3
 - data warehousing, 6–7
 - exploitation, 11
 - exploration, 10
 - interpretation step, 10–11
 - KDD, 12
 - modeling, 9
 - overarching activity, 9
 - SEMMA, 12
 - TDSP, 12
- Data shaping, 35
 - and data cleaning, 40–42
 - into a flat file, 37–40
 - framed questions, 42–43
- Data sources and formats
 - numeric and categorical, 34
 - text data, 34
- Data warehousing, 6–7
- Descriptive analysis techniques
 - data set
 - description, 53
 - explore, 53
 - quality examination, 54
 - data understanding step, 52–53
 - for group of students at a local secondary school, 54–57
 - SBA Loan data set, 66–76
 - SFO Airport Survey, 59–66
 - of the titanic disaster data, 57–59

- tools for, 52
- variables, 54

Directed learning. *See* Supervised learning

E

- Enterprise Resources Planning (ERP) tools, 81

F

- Factor Analysis (FA)
 - correlations, 192
 - diamond hunting case study, 201–203
 - factor interpretation, 195
 - objectives, 191
 - PCA and Common FA, 192–193
 - restaurant survey, 194–195
 - scree plot analysis, 193
 - of SFO survey data set, 196–200
 - summary activities, 196
 - uses of, 192
- Feature selection, covariance analysis, 190–191
- Flat-file format, 32
 - characteristics, 36
 - limitations, 35
 - in spreadsheet, 36
 - variables in data set, 37
- Forrester organization, 261–262
- Forrester Wave™ plot, 262
- Framing analytical questions
 - CRISP-DM model, 18–19
 - parsing process, 21, 24
 - San Francisco airport survey, 25–26
 - SBA loan-guarantee program, 26–28
 - Titanic disaster, 23–24

I

- IBM SPSS, 263, 264

K

- Key Performance Indicators (KPIs), 17
- Keyword analysis, 234
- Knowledge discovery in databases (KDD), 1

L

- Large data sets, 253–258
- Linear regression models, 93
 - using SBA loans data set, 114–116
 - using SFO survey data set, 108–114
- Logistic regression, 102–103
 - using SBA loans data set, 118–119
 - using SFO survey data set, 116–118

M

- Microsoft Excel, 16
- Modeling techniques
 - analytic outputs, 84
 - CRISP-DM process model
 - access model, 80–81
 - build model, 80
 - generate test design, 79–80
 - select the actual modeling technique, 79
 - definition of model, 77–78
 - ERP tools
 - Data Mining Engine, 81
 - model, 82
 - real-time data sources, 84
 - static data sources, 83–84
 - ten-step process, 84–89
 - traditional data sources, 83
- Multivariate linear regression (MLR), 99–100
 - model of franchise sales, 100–102
- Multivariate logistic regression (MLR), 105
 - database marketing, 105–107
 - vs. logistic models
 - binary outcome situations, 108

N

- Named-entity recognition (NER), 228–229

O

- Orange 3, 259–260, 265–266
- Outlier detection. *See* Anomaly detection

P

- PassClass case study, 103–105
- P-value of the coefficients, 96–97
- Python, 15

R

- RapidMiner, 264
- Regression, 93
- Regression to the mean, 91–93
- R language, 15–16
- R-squared coefficient, 95–96

S

- San Francisco Airport (SFO) survey, 25–26
 - anomaly detection, 217–222
 - classification, 133–135
 - cleaning and shaping, 45–46
 - clustering, 154–166
 - descriptive analysis techniques, 59–66
 - factor Analysis, 196–200

- linear regression models, 108–114
- logistic regression, 116–118
- time series forecasting, 178–180
- SAS Enterprise Guide (EG), 262–263
- Simple linear regression (SLR), 93–95
 - franchise advertising, 98–99
- Small Business Administration (SBA) loan
 - data set, 26–28
 - anomaly detection, 222–225
 - classification, 136–137
 - cleaning and shaping, 46–48
 - descriptive analysis techniques, 66–76
 - framing analytical questions, 26–28
 - linear regression models, 114–116
 - logistic regression, 118–119
 - time series forecasting, 180–185
- SMART
 - goals and objectives, 22
 - well-framed analytical questions, 22
- SQL, 16
 - for data extraction, 44–45
 - queries, 48–49
- Supervised learning, 92

T

- Tableau, 16
- Term Frequency (TF) analysis, 232
- Text data mining
 - case study, 232–239
 - coding qualitative data, 228
 - data visualization for analysis, 239–240
 - description, 228
 - keyword analysis, 228, 234
 - named-entity recognition, 228–229
 - sentiment analysis, 228
 - sources and formats, 230–231
 - term frequency analysis, 232
 - text similarity, 229
 - text similarity scoring, 243
 - text visualization, 228
 - Titanic disaster case study, 229
 - tools, 230
 - topic analysis, 229
 - word frequency analysis, 231
- Text similarity scoring, 243
- Time series forecasting
 - components, 174
 - description, 176
 - of Nest data set, 185–188
 - of SBA loans data set, 180–185
 - of SFO survey data set, 178–180

types, 175
of US and Chinese GDP data set, 176–178

U

Unsupervised or undirected machine learning, 141–142

V

Visual Basic for Applications (VBA), 16

Visual programming

vs. command-line coding, 261

vs. command-line coding, 261

environments, 261–262

Forrester organization, 261–262

IBM SPSS, 263

with IBM SPSS, 263, 264

open-source tools, 260

Orange 3, 265–266

with Orange 3, 265–266

RapidMiner, 264

with RapidMiner, 263–265

SAS EG, 262–263

widgets, 260

Voyant tool, 260

keyword word analysis in, 235

term frequency analysis, 249

text visualization, 241

W

Well-framed analytical questions, 22

Widgets, 260

Word frequency analysis, 231