EMC²

Data Science and Big Data Analytics

Discovering, Analyzing, Visualizing and Presenting Data

EMC Education Services

Data Science & Big Data Analytics

Data Science & Big Data Analytics

Discovering, Analyzing, Visualizing and Presenting Data

EMC Education Services

WILEY

Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data

Published by
John Wiley & Sons, Inc.
10475 Crosspoint Boulevard
Indianapolis, IN 46256
www.wiley.com

Published simultaneously in Canada

ISBN: 978-1-118-87613-8
ISBN: 978-1-118-87622-0 (ebk)
ISBN: 978-1-118-87605-3 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Library of Congress Control Number: 2014946681

Credits

Executive Editor
Carol Long
Project Editor
Kelly Talbot
Production Manager
Kathleen Wisor
Copy Editor
Karen Gill
Manager of Content Development and Assembly
Mary Beth Wakefield
Marketing Director
David Mayhew
Marketing Manager
Carrie Sherrill
Professional Technology and Strategy Director
Barry Pruett
Business Manager
Amy Knies
Associate Publisher
Jim Minatel
Project Coordinator, Cover
Patrick Redmond
Proofreader
Nancy Carrasco
Indexer
Johnna VanHoose Dinse
Cover Designer
Mallesh Gurram

About the Key Contributors

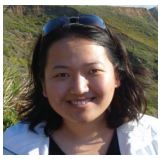
David Dietrich heads the data science education team within EMC Education Services, where he leads the curriculum, strategy and course development related to Big Data Analytics and Data Science. He co-authored the first course in EMC's Data Science curriculum, two additional EMC courses focused on teaching leaders and executives about Big Data and data science, and is a contributing author and editor of this book. He has filed 14 patents in the areas of data science, data privacy, and cloud computing.

David has been an advisor to several universities looking to develop academic programs related to data analytics, and has been a frequent speaker at conferences and industry events. He also has been a a guest lecturer at universities in the Boston area. His work has been featured in major publications including Forbes, Harvard Business Review, and the 2014 Massachusetts Big Data Report, commissioned by Governor Deval Patrick.

Involved with analytics and technology for nearly 20 years, David has worked with many Fortune 500 companies over his career, holding multiple roles involving analytics, including managing analytics and operations teams, delivering analytic consulting engagements, managing a line of analytical software products for regulating the US banking industry, and developing Software-as-a-Service and BI-as-a-Service offerings. Additionally, David collaborated with the U.S. Federal Reserve in developing predictive models for monitoring mortgage portfolios.

Barry Heller is an advisory technical education consultant at EMC Education Services. Barry is a course developer and curriculum advisor in the emerging technology areas of Big Data and data science. Prior to his current role, Barry was a consultant research scientist leading numerous analytical initiatives within EMC's Total Customer Experience organization. Early in his EMC career, he managed the statistical engineering group as well as led the data warehousing efforts in an Enterprise Resource Planning (ERP) implementation. Prior to joining EMC, Barry held managerial and analytical roles in reliability engineering functions at medical diagnostic and technology companies. During his career, he has applied his quantitative skill set to a myriad of business applications in the Customer Service, Engineering, Manufacturing, Sales/Marketing, Finance, and Legal arenas. Underscoring the importance of strong executive stakeholder engagement, many of his successes have resulted from not only focusing on the technical details of an analysis, but on the decisions that will be resulting from the analysis. Barry earned a B.S. in Computational Mathematics from the Rochester Institute of Technology and an M.A. in Mathematics from the State University of New York (SUNY) New Paltz.

Beibei Yang is a Technical Education Consultant of EMC Education Services, responsible for developing several open courses at EMC related to Data Science and Big Data Analytics. Beibei has seven years of experience in the IT industry. Prior to EMC she worked as a software engineer, systems manager, and network manager for a Fortune 500 company where she introduced new technologies to improve efficiency and encourage collaboration. Beibei has published papers to prestigious conferences and has filed multiple patents. She received her Ph.D. in computer science from the University of Massachusetts Lowell. She has a passion toward natural language processing and data mining, especially using various tools and techniques to find hidden patterns and tell stories with data. Data Science and Big Data Analytics is an exciting domain where the potential of digital information is maximized for making intelligent business decisions. We believe that this is an area that will attract a lot of talented students and professionals in the short, mid, and long term.



Acknowledgments

EMC Education Services embarked on learning this subject with the intent to develop an "open" curriculum and certification. It was a challenging journey at the time as not many understood what it would take to be a true data scientist. After initial research (and struggle), we were able to define what was needed and attract very talented professionals to work on the project. The course, "Data Science and Big Data Analytics," has become well accepted across academia and the industry.

Led by EMC Education Services, this book is the result of efforts and contributions from a number of key EMC organizations and supported by the office of the CTO, IT, Global Services, and Engineering. Many sincere thanks to many key contributors and subject matter experts **David Dietrich, Barry Heller,** and **Beibei Yang** for their work developing content and graphics for the chapters. A special thanks to subject matter experts **John Cardente** and **Ganesh Rajaratnam** for their active involvement reviewing multiple book chapters and providing valuable feedback throughout the project.

We are also grateful to the following experts from EMC and Pivotal for their support in reviewing and improving the content in this book:

Aidan O'Brien	Joe Kambourakis
Alexander Nunes	Joe Milardo
Bryan Miletich	John Sopka
Dan Baskette	Kathryn Stiles
Daniel Mepham	Ken Taylor
Dave Reiner	Lanette Wells
Deborah Stokes	Michael Hancock
Ellis Kriesberg	Michael Vander Donk
Frank Coleman	Narayanan Krishnakumar
Hisham Arafat	Richard Moore
Ira Schild	Ron Glick
Jack Harwood	Stephen Maloney
Jim McGroddy	Steve Todd

Jody Goncalves	Suresh Thankappan
Joe Dery	Tom McGowan

We also thank Ira Schild and Shane Goodrich for coordinating this project, Mallesh Gurram for the cover design, Chris Conroy and Rob Bradley for graphics, and the publisher, John Wiley and Sons, for timely support in bringing this book to the industry.

Nancy Gessler
Director, Education Services, EMC Corporation
Alok Shrivastava
Sr. Director, Education Services, EMC Corporation

Contents

Foreword

Technological advances and the associated changes in practical daily life have produced a rapidly expanding "parallel universe" of new content, new data, and new information sources all around us. Regardless of how one defines it, the phenomenon of Big Data is ever more present, ever more pervasive, and ever more important. There is enormous value potential in Big Data: innovative insights, improved understanding of problems, and countless opportunities to predict—and even to shape—the future. Data Science is the principal means to discover and tap that potential. Data Science provides ways to deal with and benefit from Big Data: to see patterns, to discover relationships, and to make sense of stunningly varied images and information.

Not everyone has studied statistical analysis at a deep level. People with advanced degrees in applied mathematics are not a commodity. Relatively few organizations have committed resources to large collections of data gathered primarily for the purpose of exploratory analysis. And yet, while applying the practices of Data Science to Big Data is a valuable differentiating strategy at present, it will be a standard core competency in the not so distant future.

How does an organization operationalize quickly to take advantage of this trend? We've created this book for that exact purpose.

EMC Education Services has been listening to the industry and organizations, observing the multi-faceted transformation of the technology landscape, and doing direct research in order to create curriculum and content to help individuals and organizations transform themselves. For the domain of Data Science and Big Data Analytics, our educational strategy balances three things: *people*—especially in the context of data science teams, *processes*—such as the analytic lifecycle approach presented in this book, and *tools and technologies*—in this case with the emphasis on proven analytic tools.

So let us help you capitalize on this new "parallel universe" that surrounds us. We invite you to learn about Data Science and Big Data Analytics through this book and hope it significantly accelerates your efforts in the transformational process.

Introduction

Big Data is creating significant new opportunities for organizations to derive new value and create competitive advantage from their most valuable asset: information. For businesses, Big Data helps drive efficiency, quality, and personalized products and services, producing improved levels of customer satisfaction and profit. For scientific efforts, Big Data analytics enable new avenues of investigation with potentially richer results and deeper insights than previously available. In many cases, Big Data analytics integrate structured and unstructured data with real-time feeds and queries, opening new paths to innovation and insight.

This book provides a practitioner's approach to some of the key techniques and tools used in Big Data analytics. Knowledge of these methods will help people become active contributors to Big Data analytics projects. The book's content is designed to assist multiple stakeholders: business and data analysts looking to add Big Data analytics skills to their portfolio; database professionals and managers of business intelligence, analytics, or Big Data groups looking to enrich their analytic skills; and college graduates investigating data science as a career field.

The content is structured in twelve chapters. The first chapter introduces the reader to the domain of Big Data, the drivers for advanced analytics, and the role of the data scientist. The second chapter presents an analytic project lifecycle designed for the particular characteristics and challenges of hypothesis-driven analysis with Big Data.

Chapter 3 examines fundamental statistical techniques in the context of the open source R analytic software environment. This chapter also highlights the importance of exploratory data analysis via visualizations and reviews the key notions of hypothesis development and testing.

Chapters 4 through 9 discuss a range of advanced analytical methods, including clustering, classification, regression analysis, time series and text analysis.

Chapters 10 and 11 focus on specific technologies and tools that support advanced analytics with Big Data. In particular, the MapReduce paradigm and its instantiation in the Hadoop ecosystem, as well as advanced topics in SQL and in-database text analytics form the focus of these chapters.

Chapter 12 provides guidance on operationalizing Big Data analytics projects. This chapter focuses on creating the final deliverables, converting an analytics project to an ongoing asset of an organization's operation, and creating clear, useful visual outputs based on the data.

EMC Academic Alliance

University and college faculties are invited to join the Academic Alliance program to access unique "open" curriculum-based education on the following topics:

- Data Science and Big Data Analytics
- Information Storage and Management
- Cloud Infrastructure and Services
- Backup Recovery Systems and Architecture

The program provides faculty with course resources to prepare students for opportunities that exist in today's evolving IT industry at no cost. For more information, visit `http://education.EMC.com/academicalliance`.

EMC Proven Professional Certification

EMC Proven Professional is a leading education and certification program in the IT industry, providing comprehensive coverage of information storage technologies, virtualization, cloud computing, data science/Big Data analytics, and more.

Being proven means investing in yourself and formally validating your expertise.

This book prepares you for Data Science Associate (EMCDSA) certification. Visit `http://education.EMC.com` for details.

1

Introduction to Big Data Analytics

Key Concepts

Big Data overview
State of the practice in analytics
Business Intelligence versus Data Science
Key roles for the new Big Data ecosystem
The Data Scientist
Examples of Big Data analytics

Much has been written about Big Data and the need for advanced analytics within industry, academia, and government. Availability of new data sources and the rise of more complex analytical opportunities have created a need to rethink existing data architectures to enable analytics that take advantage of Big Data. In addition, significant debate exists about what Big Data is and what kinds of skills are required to make best use of it. This chapter explains several key concepts to clarify what is meant by Big Data, why advanced analytics are needed, how Data Science differs from Business Intelligence (BI), and what new roles are needed for the new Big Data ecosystem.

1.1 Big Data Overview

Data is created constantly, and at an ever-increasing rate. Mobile phones, social media, imaging technologies to determine a medical diagnosis—all these and more create new data, and that must be stored somewhere for some purpose. Devices and sensors automatically generate diagnostic information that needs to be stored and processed in real time. Merely keeping up with this huge influx of data is difficult, but substantially more challenging is analyzing vast amounts of it, especially when it does not conform to traditional notions of data structure, to identify meaningful patterns and extract useful information. These challenges of the data deluge present the opportunity to transform business, government, science, and everyday life.

Several industries have led the way in developing their ability to gather and exploit data:

- Credit card companies monitor every purchase their customers make and can identify fraudulent purchases with a high degree of accuracy using rules derived by processing billions of transactions.
- Mobile phone companies analyze subscribers' calling patterns to determine, for example, whether a caller's frequent contacts are on a rival network. If that rival network is offering an attractive promotion that might cause the subscriber to defect, the mobile phone company can proactively offer the subscriber an incentive to remain in her contract.
- For companies such as LinkedIn and Facebook, data itself is their primary product. The valuations of these companies are heavily derived from the data they gather and host, which contains more and more intrinsic value as the data grows.

Three attributes stand out as defining Big Data characteristics:

- **Huge volume of data:** Rather than thousands or millions of rows, Big Data can be billions of rows and millions of columns.
- **Complexity of data types and structures:** Big Data reflects the variety of new data sources, formats, and structures, including digital traces being left on the web and other digital repositories for subsequent analysis.
- **Speed of new data creation and growth:** Big Data can describe high velocity data, with rapid data ingestion and near real time analysis.

Although the volume of Big Data tends to attract the most attention, generally the variety and velocity of the data provide a more apt definition of Big Data. (Big Data is sometimes described as having 3 Vs: volume, variety, and velocity.) Due to its size or structure, Big Data cannot be efficiently analyzed using only traditional databases or methods. Big Data problems require new tools and technologies to store, manage, and realize the business benefit. These new tools and technologies enable creation, manipulation, and

management of large datasets and the storage environments that house them. Another definition of Big Data comes from the McKinsey Global report from 2011:

> ***Big Data is data whose scale, distribution, diversity, and/or timeliness require the use of new technical architectures and analytics to enable insights that unlock new sources of business value.***
>
> McKinsey & Co.; Big Data: The Next Frontier for Innovation, Competition, and Productivity [1]

McKinsey's definition of Big Data implies that organizations will need new data architectures and analytic sandboxes, new tools, new analytical methods, and an integration of multiple skills into the new role of the data scientist, which will be discussed in Section 1.3. Figure 1-1 highlights several sources of the Big Data deluge.

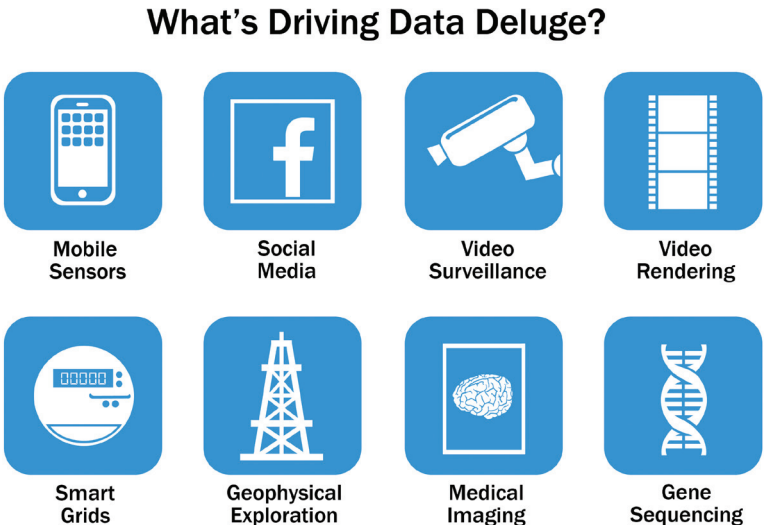


FIGURE 1-1 *What's driving the data deluge*

The rate of data creation is accelerating, driven by many of the items in Figure 1-1.

Social media and genetic sequencing are among the fastest-growing sources of Big Data and examples of untraditional sources of data being used for analysis.

For example, in 2012 Facebook users posted 700 status updates per second worldwide, which can be leveraged to deduce latent interests or political views of users and show relevant ads. For instance, an update in which a woman changes her relationship status from "single" to "engaged" would trigger ads on bridal dresses, wedding planning, or name-changing services.

Facebook can also construct social graphs to analyze which users are connected to each other as an interconnected network. In March 2013, Facebook released a new feature called "Graph Search," enabling users and developers to search social graphs for people with similar interests, hobbies, and shared locations.

Another example comes from genomics. Genetic sequencing and human genome mapping provide a detailed understanding of genetic makeup and lineage. The health care industry is looking toward these advances to help predict which illnesses a person is likely to get in his lifetime and take steps to avoid these maladies or reduce their impact through the use of personalized medicine and treatment. Such tests also highlight typical responses to different medications and pharmaceutical drugs, heightening risk awareness of specific drug treatments.

While data has grown, the cost to perform this work has fallen dramatically. The cost to sequence one human genome has fallen from $100 million in 2001 to $10,000 in 2011, and the cost continues to drop. Now, websites such as 23andme (Figure 1-2) offer genotyping for less than $100. Although genotyping analyzes only a fraction of a genome and does not provide as much granularity as genetic sequencing, it does point to the fact that data and complex analysis is becoming more prevalent and less expensive to deploy.

FIGURE 1-2 *Examples of what can be learned through genotyping, from 23andme.com*

As illustrated by the examples of social media and genetic sequencing, individuals and organizations both derive benefits from analysis of ever-larger and more complex datasets that require increasingly powerful analytical capabilities.

1.1.1 Data Structures

Big data can come in multiple forms, including structured and non-structured data such as financial data, text files, multimedia files, and genetic mappings. Contrary to much of the traditional data analysis performed by organizations, most of the Big Data is unstructured or semi-structured in nature, which requires different techniques and tools to process and analyze. [2] Distributed computing environments and massively parallel processing (MPP) architectures that enable parallelized data ingest and analysis are the preferred approach to process such complex data.

With this in mind, this section takes a closer look at data structures.

Figure 1-3 shows four types of data structures, with 80–90% of future data growth coming from non-structured data types. [2] Though different, the four are commonly mixed. For example, a classic Relational Database Management System (RDBMS) may store call logs for a software support call center. The RDBMS may store characteristics of the support calls as typical structured data, with attributes such as time stamps, machine type, problem type, and operating system. In addition, the system will likely have unstructured, quasi- or semi-structured data, such as free-form call log information taken from an e-mail ticket of the problem, customer chat history, or transcript of a phone call describing the technical problem and the solution or audio file of the phone call conversation. Many insights could be extracted from the unstructured, quasi- or semi-structured data in the call center data.

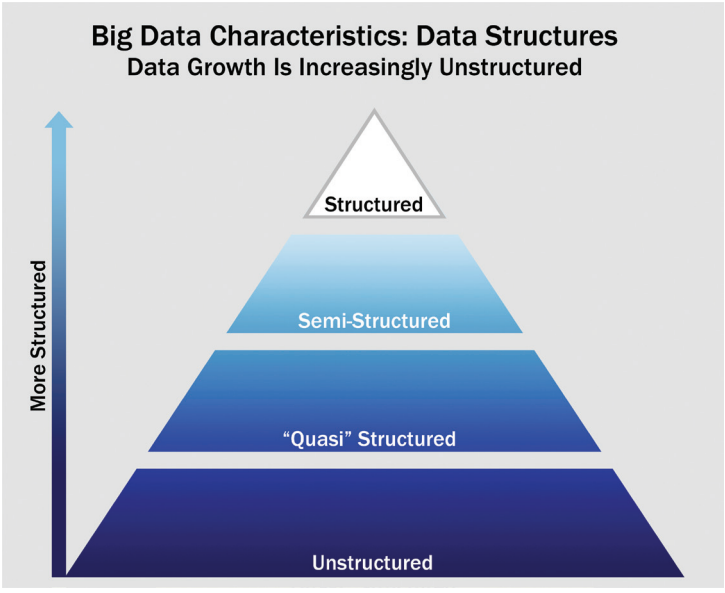


Figure 1-3 *Big Data Growth is increasingly unstructured*

Although analyzing structured data tends to be the most familiar technique, a different technique is required to meet the challenges to analyze semi-structured data (shown as XML), quasi-structured (shown as a clickstream), and unstructured data.

Here are examples of how each of the four main types of data structures may look.

- **Structured data:** Data containing a defined data type, format, and structure (that is, transaction data, online analytical processing [OLAP] data cubes, traditional RDBMS, CSV files, and even simple spreadsheets). See Figure 1-4.

SUMMER FOOD SERVICE PROGRAM 1]				
(Data as of August 01, 2011)				
Fiscal Year	Number of Sites	Peak (July) Participation	Meals Served	Total Federal Expenditures 2]
	----------Thousands----------		--Mil.--	---Million $---
1969	1.2	99	2.2	0.3
1970	1.9	227	8.2	1.8
1971	3.2	569	29.0	8.2
1972	6.5	1,080	73.5	21.9
1973	11.2	1,437	65.4	26.6
1974	10.6	1,403	63.6	33.6
1975	12.0	1,785	84.3	50.3
1976	16.0	2,453	104.8	73.4
TQ 3]	22.4	3,455	198.0	88.9
1977	23.7	2,791	170.4	114.4
1978	22.4	2,333	120.3	100.3
1979	23.0	2,126	121.8	108.6
1980	21.6	1,922	108.2	110.1
1981	20.6	1,726	90.3	105.9
1982	14.4	1,397	68.2	87.1
1983	14.9	1,401	71.3	93.4
1984	15.1	1,422	73.8	96.2
1985	16.0	1,462	77.2	111.5
1986	16.1	1,509	77.1	114.7
1987	16.9	1,560	79.9	129.3
1988	17.2	1,577	80.3	133.3
1989	18.5	1,652	86.0	143.8
1990	19.2	1,692	91.2	163.3

Figure 1-4 *Example of structured data*

- **Semi-structured data:** Textual data files with a discernible pattern that enables parsing (such as Extensible Markup Language [XML] data files that are self-describing and defined by an XML schema). See Figure 1-5.
- **Quasi-structured data:** Textual data with erratic data formats that can be formatted with effort, tools, and time (for instance, web clickstream data that may contain inconsistencies in data values and formats). See Figure 1-6.
- **Unstructured data:** Data that has no inherent structure, which may include text documents, PDFs, images, and video. See Figure 1-7.

Quasi-structured data is a common phenomenon that bears closer scrutiny. Consider the following example. A user attends the EMC World conference and subsequently runs a Google search online to find information related to EMC and Data Science. This would produce a URL such as `https://www.google.com/#q=EMC+ data+science` and a list of results, such as in the first graphic of Figure 1-5.

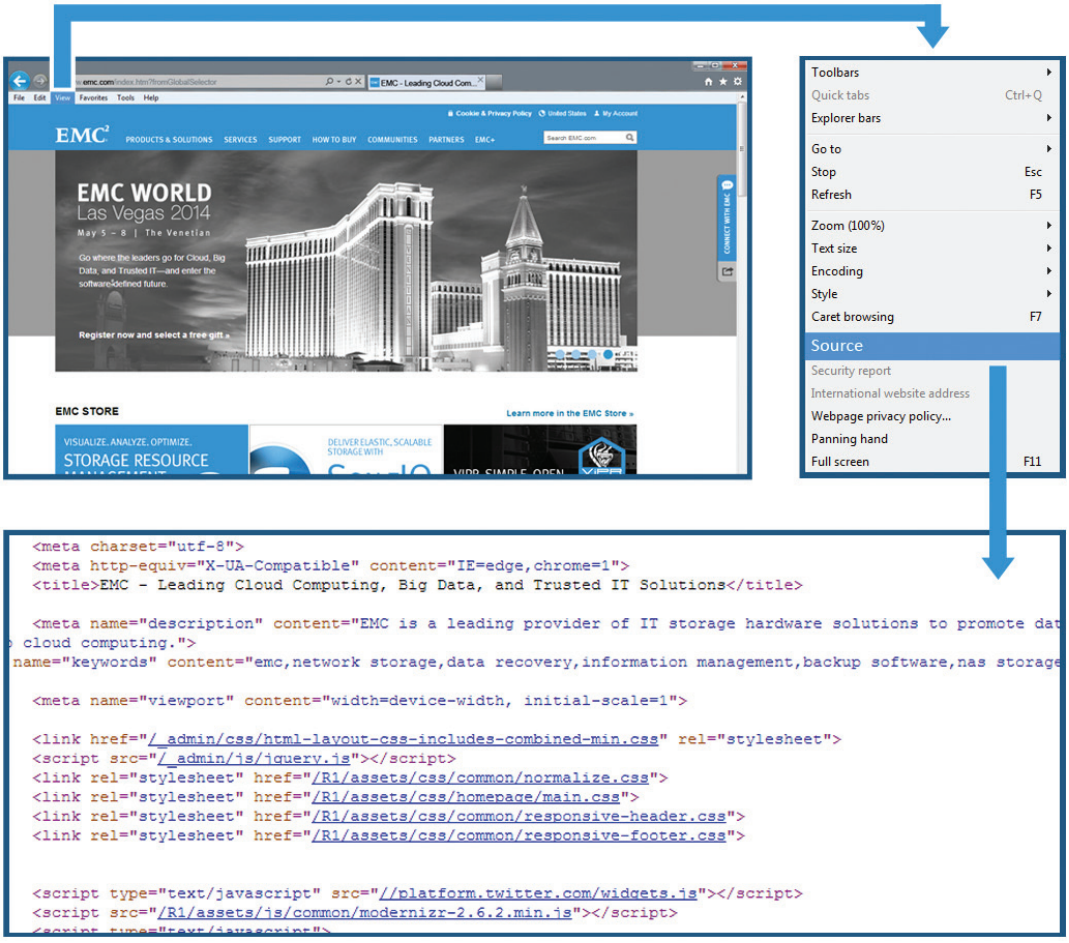


FIGURE 1-5 *Example of semi-structured data*

After doing this search, the user may choose the second link, to read more about the headline "Data Scientist—EMC Education, Training, and Certification." This brings the user to an `emc.com` site focused on this topic and a new URL, `https://education.emc.com/guest/campaign/data_science`

.aspx, that displays the page shown as (2) in Figure 1-6. Arriving at this site, the user may decide to click to learn more about the process of becoming certified in data science. The user chooses a link toward the top of the page on Certifications, bringing the user to a new URL: `https://education.emc.com/guest/certification/framework/stf/data_science.aspx`, which is (3) in Figure 1-6.

Visiting these three websites adds three URLs to the log files monitoring the user's computer or network use. These three URLs are:

```
https://www.google.com/#q=EMC+data+science
https://education.emc.com/guest/campaign/data_science.aspx
https://education.emc.com/guest/certification/framework/stf/data_
science.aspx
```

FIGURE 1-6 *Example of EMC Data Science search results*

FIGURE 1-7 *Example of unstructured data: video about Antarctica expedition [3]*

This set of three URLs reflects the websites and actions taken to find Data Science information related to EMC. Together, this comprises a *clickstream* that can be parsed and mined by data scientists to discover usage patterns and uncover relationships among clicks and areas of interest on a website or group of sites.

The four data types described in this chapter are sometimes generalized into two groups: structured and unstructured data. Big Data describes new kinds of data with which most organizations may not be used to working. With this in mind, the next section discusses common technology architectures from the standpoint of someone wanting to analyze Big Data.

1.1.2 Analyst Perspective on Data Repositories

The introduction of spreadsheets enabled business users to create simple logic on data structured in rows and columns and create their own analyses of business problems. Database administrator training is not required to create spreadsheets: They can be set up to do many things quickly and independently of information technology (IT) groups. Spreadsheets are easy to share, and end users have control over the logic involved. However, their proliferation can result in "many versions of the truth." In other words, it can be challenging to determine if a particular user has the most relevant version of a spreadsheet, with the most current data and logic in it. Moreover, if a laptop is lost or a file becomes corrupted, the data and logic within the spreadsheet could be lost. This is an ongoing challenge because spreadsheet programs such as Microsoft Excel still run on many computers worldwide. With the proliferation of data islands (or spreadmarts), the need to centralize the data is more pressing than ever.

As data needs grew, so did more scalable data warehousing solutions. These technologies enabled data to be managed centrally, providing benefits of security, failover, and a single repository where users

could rely on getting an "official" source of data for financial reporting or other mission-critical tasks. This structure also enabled the creation of OLAP cubes and BI analytical tools, which provided quick access to a set of dimensions within an RDBMS. More advanced features enabled performance of in-depth analytical techniques such as regressions and neural networks. Enterprise Data Warehouses (EDWs) are critical for reporting and BI tasks and solve many of the problems that proliferating spreadsheets introduce, such as which of multiple versions of a spreadsheet is correct. EDWs—and a good BI strategy—provide direct data feeds from sources that are centrally managed, backed up, and secured.

Despite the benefits of EDWs and BI, these systems tend to restrict the flexibility needed to perform robust or exploratory data analysis. With the EDW model, data is managed and controlled by IT groups and database administrators (DBAs), and data analysts must depend on IT for access and changes to the data schemas. This imposes longer lead times for analysts to get data; most of the time is spent waiting for approvals rather than starting meaningful work. Additionally, many times the EDW rules restrict analysts from building datasets. Consequently, it is common for additional systems to emerge containing critical data for constructing analytic datasets, managed locally by power users. IT groups generally dislike existence of data sources outside of their control because, unlike an EDW, these datasets are not managed, secured, or backed up. From an analyst perspective, EDW and BI solve problems related to data accuracy and availability. However, EDW and BI introduce new problems related to flexibility and agility, which were less pronounced when dealing with spreadsheets.

A solution to this problem is the analytic sandbox, which attempts to resolve the conflict for analysts and data scientists with EDW and more formally managed corporate data. In this model, the IT group may still manage the analytic sandboxes, but they will be purposefully designed to enable robust analytics, while being centrally managed and secured. These sandboxes, often referred to as *workspaces,* are designed to enable teams to explore many datasets in a controlled fashion and are not typically used for enterprise-level financial reporting and sales dashboards.

Many times, analytic sandboxes enable high-performance computing using in-database processing—the analytics occur within the database itself. The idea is that performance of the analysis will be better if the analytics are run in the database itself, rather than bringing the data to an analytical tool that resides somewhere else. In-database analytics, discussed further in Chapter 11, "Advanced Analytics—Technology and Tools: In-Database Analytics," creates relationships to multiple data sources within an organization and saves time spent creating these data feeds on an individual basis. In-database processing for deep analytics enables faster turnaround time for developing and executing new analytic models, while reducing, though not eliminating, the cost associated with data stored in local, "shadow" file systems. In addition, rather than the typical structured data in the EDW, analytic sandboxes can house a greater variety of data, such as raw data, textual data, and other kinds of unstructured data, without interfering with critical production databases. Table 1-1 summarizes the characteristics of the data repositories mentioned in this section.

TABLE 1-1 *Types of Data Repositories, from an Analyst Perspective*

Data Repository	Characteristics
Spreadsheets and data marts ("spreadmarts")	Spreadsheets and low-volume databases for recordkeeping Analyst depends on data extracts.

Data Warehouses	Centralized data containers in a purpose-built space Supports BI and reporting, but restricts robust analyses Analyst dependent on IT and DBAs for data access and schema changes Analysts must spend significant time to get aggregated and disaggregated data extracts from multiple sources.
Analytic Sandbox (workspaces)	Data assets gathered from multiple sources and technologies for analysis Enables flexible, high-performance analysis in a nonproduction environment; can leverage in-database processing Reduces costs and risks associated with data replication into "shadow" file systems "Analyst owned" rather than "DBA owned"

There are several things to consider with Big Data Analytics projects to ensure the approach fits with the desired goals. Due to the characteristics of Big Data, these projects lend themselves to decision support for high-value, strategic decision making with high processing complexity. The analytic techniques used in this context need to be iterative and flexible, due to the high volume of data and its complexity. Performing rapid and complex analysis requires high throughput network connections and a consideration for the acceptable amount of latency. For instance, developing a real-time product recommender for a website imposes greater system demands than developing a near-real-time recommender, which may still provide acceptable performance, have slightly greater latency, and may be cheaper to deploy. These considerations require a different approach to thinking about analytics challenges, which will be explored further in the next section.

1.2 State of the Practice in Analytics

Current business problems provide many opportunities for organizations to become more analytical and data driven, as shown in Table 1-2.

Table 1-2 *Business Drivers for Advanced Analytics*

Business Driver	Examples
Optimize business operations	Sales, pricing, profitability, efficiency
Identify business risk	Customer churn, fraud, default
Predict new business opportunities	Upsell, cross-sell, best new customer prospects
Comply with laws or regulatory requirements	Anti-Money Laundering, Fair Lending, Basel II-III, Sarbanes-Oxley (SOX)

Table 1-2 outlines four categories of common business problems that organizations contend with where they have an opportunity to leverage advanced analytics to create competitive advantage. Rather than only performing standard reporting on these areas, organizations can apply advanced analytical techniques to optimize processes and derive more value from these common tasks. The first three examples do not represent new problems. Organizations have been trying to reduce customer churn, increase sales, and cross-sell customers for many years. What is new is the opportunity to fuse advanced analytical techniques with Big Data to produce more impactful analyses for these traditional problems. The last example portrays emerging regulatory requirements. Many compliance and regulatory laws have been in existence for decades, but additional requirements are added every year, which represent additional complexity and data requirements for organizations. Laws related to anti-money laundering (AML) and fraud prevention require advanced analytical techniques to comply with and manage properly.

1.2.1 BI Versus Data Science

The four business drivers shown in Table 1-2 require a variety of analytical techniques to address them properly. Although much is written generally about analytics, it is important to distinguish between BI and Data Science. As shown in Figure 1-8, there are several ways to compare these groups of analytical techniques.

One way to evaluate the type of analysis being performed is to examine the time horizon and the kind of analytical approaches being used. BI tends to provide reports, dashboards, and queries on business questions for the current period or in the past. BI systems make it easy to answer questions related to quarter-to-date revenue, progress toward quarterly targets, and understand how much of a given product was sold in a prior quarter or year. These questions tend to be closed-ended and explain current or past behavior, typically by aggregating historical data and grouping it in some way. BI provides hindsight and some insight and generally answers questions related to "when" and "where" events occurred.

By comparison, Data Science tends to use disaggregated data in a more forward-looking, exploratory way, focusing on analyzing the present and enabling informed decisions about the future. Rather than aggregating historical data to look at how many of a given product sold in the previous quarter, a team may employ Data Science techniques such as time series analysis, further discussed in Chapter 8, "Advanced Analytical Theory and Methods: Time Series Analysis," to forecast future product sales and revenue more accurately than extending a simple trend line. In addition, Data Science tends to be more exploratory in nature and may use scenario optimization to deal with more open-ended questions. This approach provides insight into current activity and foresight into future events, while generally focusing on questions related to "how" and "why" events occur.

Where BI problems tend to require highly structured data organized in rows and columns for accurate reporting, Data Science projects tend to use many types of data sources, including large or unconventional datasets. Depending on an organization's goals, it may choose to embark on a BI project if it is doing reporting, creating dashboards, or performing simple visualizations, or it may choose Data Science projects if it needs to do a more sophisticated analysis with disaggregated or varied datasets.

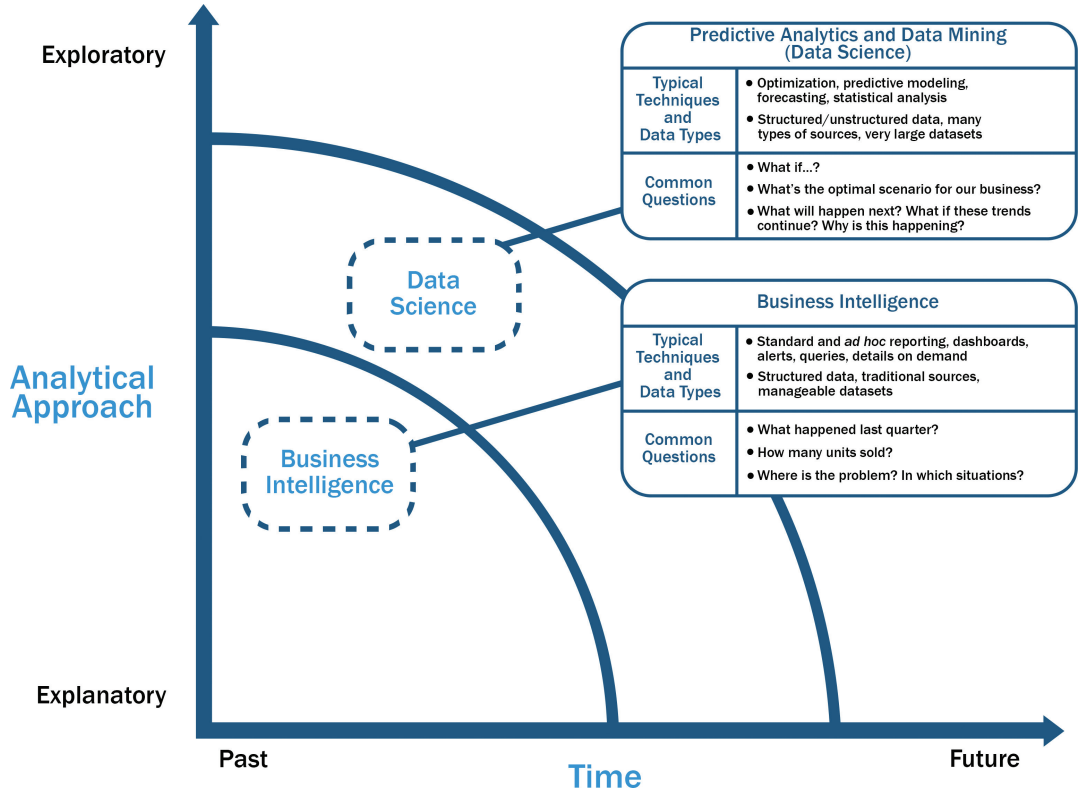


FIGURE 1-8 *Comparing BI with Data Science*

1.2.2 Current Analytical Architecture

As described earlier, Data Science projects need workspaces that are purpose-built for experimenting with data, with flexible and agile data architectures. Most organizations still have data warehouses that provide excellent support for traditional reporting and simple data analysis activities but unfortunately have a more difficult time supporting more robust analyses. This section examines a typical analytical data architecture that may exist within an organization.

Figure 1-9 shows a typical data architecture and several of the challenges it presents to data scientists and others trying to do advanced analytics. This section examines the data flow to the Data Scientist and how this individual fits into the process of getting data to analyze on projects.

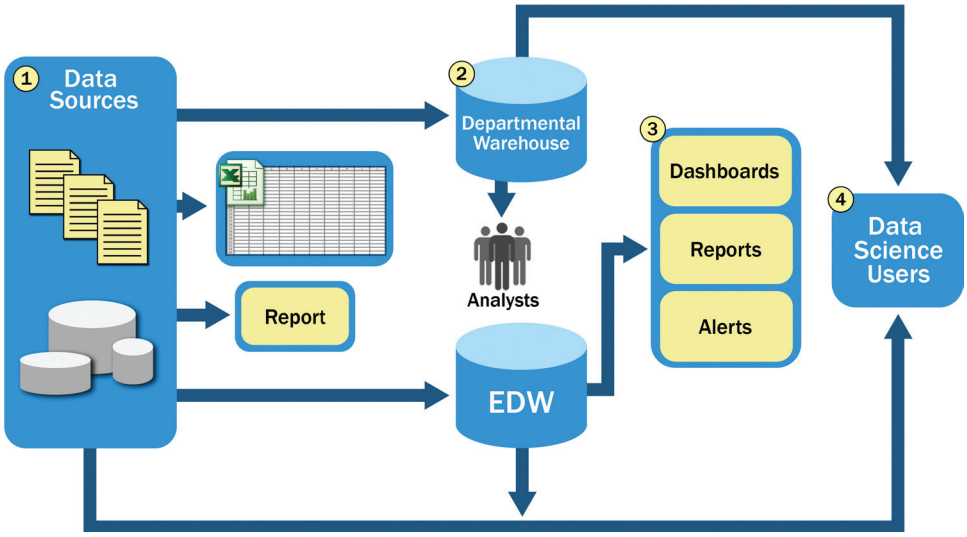


FIGURE 1-9 *Typical analytic architecture*

1. For data sources to be loaded into the data warehouse, data needs to be well understood, structured, and normalized with the appropriate data type definitions. Although this kind of centralization enables security, backup, and failover of highly critical data, it also means that data typically must go through significant preprocessing and checkpoints before it can enter this sort of controlled environment, which does not lend itself to data exploration and iterative analytics.
2. As a result of this level of control on the EDW, additional local systems may emerge in the form of departmental warehouses and local data marts that business users create to accommodate their need for flexible analysis. These local data marts may not have the same constraints for security and structure as the main EDW and allow users to do some level of more in-depth analysis. However, these one-off systems reside in isolation, often are not synchronized or integrated with other data stores, and may not be backed up.
3. Once in the data warehouse, data is read by additional applications across the enterprise for BI and reporting purposes. These are high-priority operational processes getting critical data feeds from the data warehouses and repositories.
4. At the end of this workflow, analysts get data provisioned for their downstream analytics. Because users generally are not allowed to run custom or intensive analytics on production databases, analysts create data extracts from the EDW to analyze data offline in R or other local analytical tools. Many times these tools are limited to in-memory analytics on desktops analyzing samples of data, rather than the entire population of a dataset. Because these analyses are based on data extracts, they reside in a separate location, and the results of the analysis—and any insights on the quality of the data or anomalies—rarely are fed back into the main data repository.

Because new data sources slowly accumulate in the EDW due to the rigorous validation and data structuring process, data is slow to move into the EDW, and the data schema is slow to change.

Departmental data warehouses may have been originally designed for a specific purpose and set of business needs, but over time evolved to house more and more data, some of which may be forced into existing schemas to enable BI and the creation of OLAP cubes for analysis and reporting. Although the EDW achieves the objective of reporting and sometimes the creation of dashboards, EDWs generally limit the ability of analysts to iterate on the data in a separate nonproduction environment where they can conduct in-depth analytics or perform analysis on unstructured data.

The typical data architectures just described are designed for storing and processing mission-critical data, supporting enterprise applications, and enabling corporate reporting activities. Although reports and dashboards are still important for organizations, most traditional data architectures inhibit data exploration and more sophisticated analysis. Moreover, traditional data architectures have several additional implications for data scientists.

- High-value data is hard to reach and leverage, and predictive analytics and data mining activities are last in line for data. Because the EDWs are designed for central data management and reporting, those wanting data for analysis are generally prioritized after operational processes.
- Data moves in batches from EDW to local analytical tools. This workflow means that data scientists are limited to performing in-memory analytics (such as with R, SAS, SPSS, or Excel), which will restrict the size of the datasets they can use. As such, analysis may be subject to constraints of sampling, which can skew model accuracy.
- Data Science projects will remain isolated and ad hoc, rather than centrally managed. The implication of this isolation is that the organization can never harness the power of advanced analytics in a scalable way, and Data Science projects will exist as nonstandard initiatives, which are frequently not aligned with corporate business goals or strategy.

All these symptoms of the traditional data architecture result in a slow "time-to-insight" and lower business impact than could be achieved if the data were more readily accessible and supported by an environment that promoted advanced analytics. As stated earlier, one solution to this problem is to introduce analytic sandboxes to enable data scientists to perform advanced analytics in a controlled and sanctioned way. Meanwhile, the current Data Warehousing solutions continue offering reporting and BI services to support management and mission-critical operations.

1.2.3 Drivers of Big Data

To better understand the market drivers related to Big Data, it is helpful to first understand some past history of data stores and the kinds of repositories and tools to manage these data stores.

As shown in Figure 1-10, in the 1990s the volume of information was often measured in terabytes. Most organizations analyzed structured data in rows and columns and used relational databases and data warehouses to manage large stores of enterprise information. The following decade saw a proliferation of different kinds of data sources—mainly productivity and publishing tools such as content management repositories and networked attached storage systems—to manage this kind of information, and the data began to increase in size and started to be measured at petabyte scales. In the 2010s, the information that organizations try to manage has broadened to include many other kinds of data. In this era, everyone and everything is leaving a digital footprint. Figure 1-10 shows a summary perspective on sources of Big Data generated by new applications and the scale and growth rate of the data. These applications, which generate data volumes that can be measured in exabyte scale, provide opportunities for new analytics and driving new value for organizations. The data now comes from multiple sources, such as these:

- Medical information, such as genomic sequencing and diagnostic imaging
- Photos and video footage uploaded to the World Wide Web
- Video surveillance, such as the thousands of video cameras spread across a city
- Mobile devices, which provide geospatial location data of the users, as well as metadata about text messages, phone calls, and application usage on smart phones
- Smart devices, which provide sensor-based collection of information from smart electric grids, smart buildings, and many other public and industry infrastructures
- Nontraditional IT devices, including the use of radio-frequency identification (RFID) readers, GPS navigation systems, and seismic processing

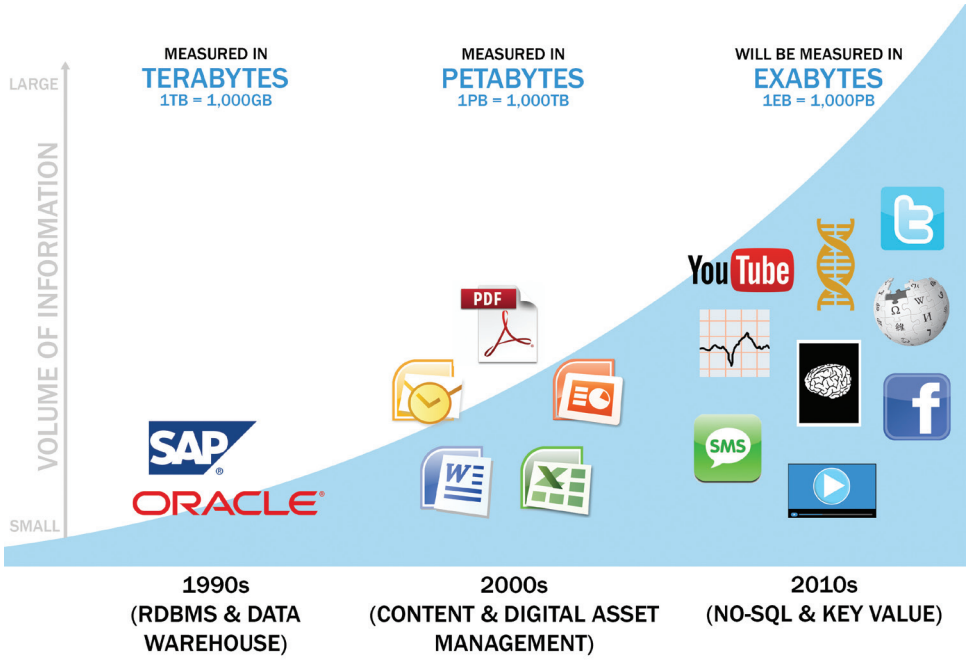


FIGURE 1-10 *Data evolution and the rise of Big Data sources*

The Big Data trend is generating an enormous amount of information from many new sources. This data deluge requires advanced analytics and new market players to take advantage of these opportunities and new market dynamics, which will be discussed in the following section.

1.2.4 Emerging Big Data Ecosystem and a New Approach to Analytics

Organizations and data collectors are realizing that the data they can gather from individuals contains intrinsic value and, as a result, a new economy is emerging. As this new digital economy continues to

evolve, the market sees the introduction of data vendors and data cleaners that use crowdsourcing (such as Mechanical Turk and GalaxyZoo) to test the outcomes of machine learning techniques. Other vendors offer added value by repackaging open source tools in a simpler way and bringing the tools to market. Vendors such as Cloudera, Hortonworks, and Pivotal have provided this value-add for the open source framework Hadoop.

As the new ecosystem takes shape, there are four main groups of players within this interconnected web. These are shown in Figure 1-11.

- **Data devices** [shown in the (1) section of Figure 1-11] and the "Sensornet" gather data from multiple locations and continuously generate new data about this data. For each gigabyte of new data created, an additional petabyte of data is created about that data. [2]
    - For example, consider someone playing an online video game through a PC, game console, or smartphone. In this case, the video game provider captures data about the skill and levels attained by the player. Intelligent systems monitor and log how and when the user plays the game. As a consequence, the game provider can fine-tune the difficulty of the game, suggest other related games that would most likely interest the user, and offer additional equipment and enhancements for the character based on the user's age, gender, and interests. This information may get stored locally or uploaded to the game provider's cloud to analyze the gaming habits and opportunities for upsell and cross-sell, and identify archetypical profiles of specific kinds of users.
    - Smartphones provide another rich source of data. In addition to messaging and basic phone usage, they store and transmit data about Internet usage, SMS usage, and real-time location. This metadata can be used for analyzing traffic patterns by scanning the density of smartphones in locations to track the speed of cars or the relative traffic congestion on busy roads. In this way, GPS devices in cars can give drivers real-time updates and offer alternative routes to avoid traffic delays.
    - Retail shopping loyalty cards record not just the amount an individual spends, but the locations of stores that person visits, the kinds of products purchased, the stores where goods are purchased most often, and the combinations of products purchased together. Collecting this data provides insights into shopping and travel habits and the likelihood of successful advertisement targeting for certain types of retail promotions.
- **Data collectors** [the blue ovals, identified as (2) within Figure 1-11] include sample entities that collect data from the device and users.
    - Data results from a cable TV provider tracking the shows a person watches, which TV channels someone will and will not pay for to watch on demand, and the prices someone is willing to pay for premium TV content
    - Retail stores tracking the path a customer takes through their store while pushing a shopping cart with an RFID chip so they can gauge which products get the most foot traffic using geospatial data collected from the RFID chips
- **Data aggregators** (the dark gray ovals in Figure 1-11, marked as (3)) make sense of the data collected from the various entities from the "SensorNet" or the "Internet of Things." These organizations compile data from the devices and usage patterns collected by government agencies, retail stores,

and websites. In turn, they can choose to transform and package the data as products to sell to list brokers, who may want to generate marketing lists of people who may be good targets for specific ad campaigns.

- **Data users and buyers** are denoted by (4) in Figure 1-11. These groups directly benefit from the data collected and aggregated by others within the data value chain.
    - Retail banks, acting as a data buyer, may want to know which customers have the highest likelihood to apply for a second mortgage or a home equity line of credit. To provide input for this analysis, retail banks may purchase data from a data aggregator. This kind of data may include demographic information about people living in specific locations; people who appear to have a specific level of debt, yet still have solid credit scores (or other characteristics such as paying bills on time and having savings accounts) that can be used to infer credit worthiness; and those who are searching the web for information about paying off debts or doing home remodeling projects. Obtaining data from these various sources and aggregators will enable a more targeted marketing campaign, which would have been more challenging before Big Data due to the lack of information or high-performing technologies.
    - Using technologies such as Hadoop to perform natural language processing on unstructured, textual data from social media websites, users can gauge the reaction to events such as presidential campaigns. People may, for example, want to determine public sentiments toward a candidate by analyzing related blogs and online comments. Similarly, data users may want to track and prepare for natural disasters by identifying which areas a hurricane affects first and how it moves, based on which geographic areas are tweeting about it or discussing it via social media.

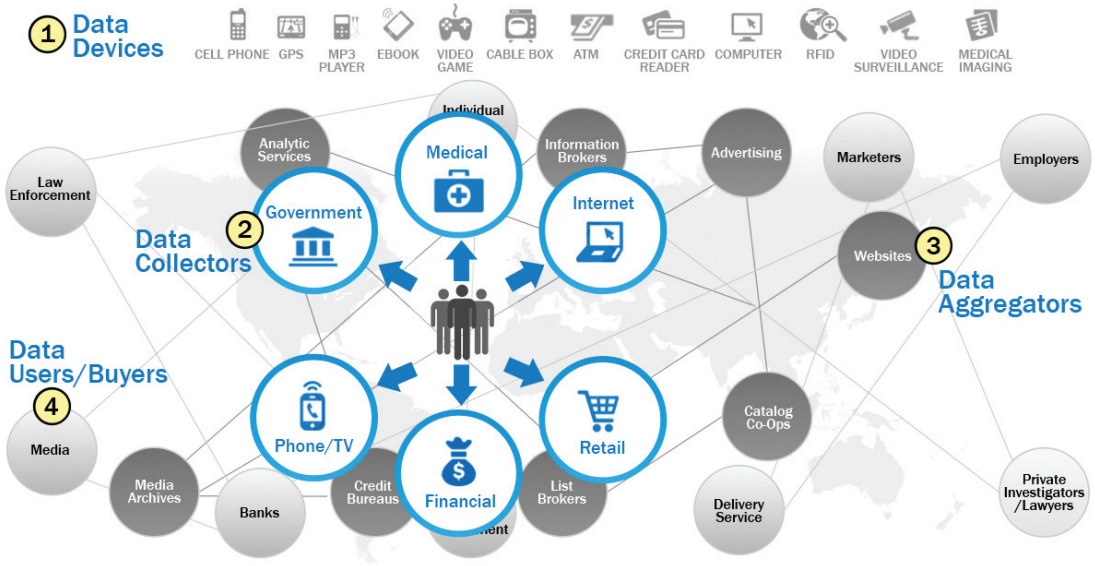


FIGURE 1-11 *Emerging Big Data ecosystem*

As illustrated by this emerging Big Data ecosystem, the kinds of data and the related market dynamics vary greatly. These datasets can include sensor data, text, structured datasets, and social media. With this in mind, it is worth recalling that these datasets will not work well within traditional EDWs, which were architected to streamline reporting and dashboards and be centrally managed. Instead, Big Data problems and projects require different approaches to succeed.

Analysts need to partner with IT and DBAs to get the data they need within an analytic sandbox. A typical analytical sandbox contains raw data, aggregated data, and data with multiple kinds of structure. The sandbox enables robust exploration of data and requires a savvy user to leverage and take advantage of data in the sandbox environment.

1.3 Key Roles for the New Big Data Ecosystem

As explained in the context of the Big Data ecosystem in Section 1.2.4, new players have emerged to curate, store, produce, clean, and transact data. In addition, the need for applying more advanced analytical techniques to increasingly complex business problems has driven the emergence of new roles, new technology platforms, and new analytical methods. This section explores the new roles that address these needs, and subsequent chapters explore some of the analytical methods and technology platforms.

The Big Data ecosystem demands three categories of roles, as shown in Figure 1-12. These roles were described in the McKinsey Global study on Big Data, from May 2011 [1].

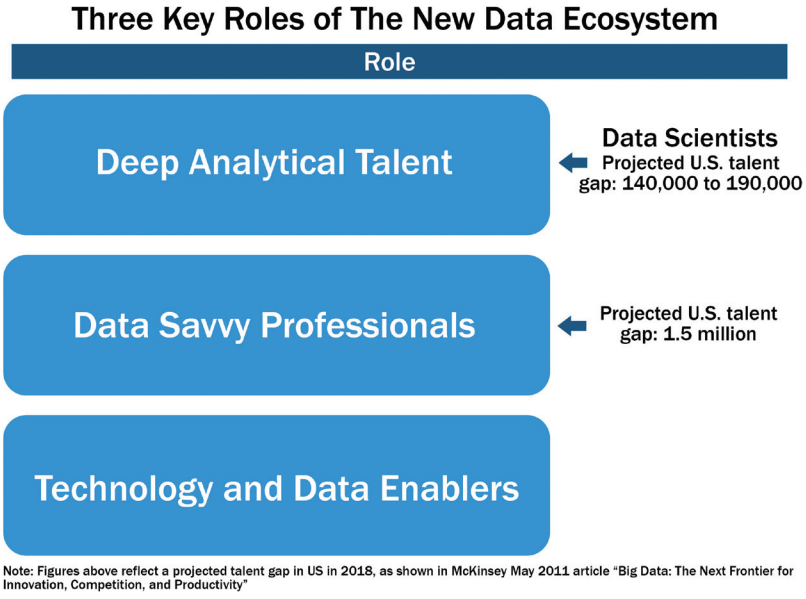


FIGURE 1-12 *Key roles of the new Big Data ecosystem*

The first group—Deep Analytical Talent— is technically savvy, with strong analytical skills. Members possess a combination of skills to handle raw, unstructured data and to apply complex analytical techniques at

massive scales. This group has advanced training in quantitative disciplines, such as mathematics, statistics, and machine learning. To do their jobs, members need access to a robust analytic sandbox or workspace where they can perform large-scale analytical data experiments. Examples of current professions fitting into this group include statisticians, economists, mathematicians, and the new role of the Data Scientist.

The McKinsey study forecasts that by the year 2018, the United States will have a talent gap of 140,000–190,000 people with deep analytical talent. This does not represent the number of people needed with deep analytical talent; rather, this range represents the difference between what will be available in the workforce compared with what will be needed. In addition, these estimates only reflect forecasted talent shortages in the United States; the number would be much larger on a global basis.

The second group—Data Savvy Professionals—has less technical depth but has a basic knowledge of statistics or machine learning and can define key questions that can be answered using advanced analytics. These people tend to have a base knowledge of working with data, or an appreciation for some of the work being performed by data scientists and others with deep analytical talent. Examples of data savvy professionals include financial analysts, market research analysts, life scientists, operations managers, and business and functional managers.

The McKinsey study forecasts the projected U.S. talent gap for this group to be 1.5 million people by the year 2018. At a high level, this means for every Data Scientist profile needed, the gap will be ten times as large for Data Savvy Professionals. Moving toward becoming a data savvy professional is a critical step in broadening the perspective of managers, directors, and leaders, as this provides an idea of the kinds of questions that can be solved with data.

The third category of people mentioned in the study is Technology and Data Enablers. This group represents people providing technical expertise to support analytical projects, such as provisioning and administrating analytical sandboxes, and managing large-scale data architectures that enable widespread analytics within companies and other organizations. This role requires skills related to computer engineering, programming, and database administration.

These three groups must work together closely to solve complex Big Data challenges. Most organizations are familiar with people in the latter two groups mentioned, but the first group, Deep Analytical Talent, tends to be the newest role for most and the least understood. For simplicity, this discussion focuses on the emerging role of the Data Scientist. It describes the kinds of activities that role performs and provides a more detailed view of the skills needed to fulfill that role.

There are three recurring sets of activities that data scientists perform:

- **Reframe business challenges as analytics challenges.** Specifically, this is a skill to diagnose business problems, consider the core of a given problem, and determine which kinds of candidate analytical methods can be applied to solve it. This concept is explored further in Chapter 2, "Data Analytics Lifecycle."
- **Design, implement, and deploy statistical models and data mining techniques on Big Data.** This set of activities is mainly what people think about when they consider the role of the Data Scientist:

namely, applying complex or advanced analytical methods to a variety of business problems using data. Chapter 3 through Chapter 11 of this book introduces the reader to many of the most popular analytical techniques and tools in this area.

- **Develop insights that lead to actionable recommendations.** It is critical to note that applying advanced methods to data problems does not necessarily drive new business value. Instead, it is important to learn how to draw insights out of the data and communicate them effectively. Chapter 12, "The Endgame, or Putting It All Together," has a brief overview of techniques for doing this.

Data scientists are generally thought of as having five main sets of skills and behavioral characteristics, as shown in Figure 1-13:

- **Quantitative skill:** such as mathematics or statistics
- **Technical aptitude:** namely, software engineering, machine learning, and programming skills
- **Skeptical mind-set and critical thinking:** It is important that data scientists can examine their work critically rather than in a one-sided way.
- **Curious and creative**: Data scientists are passionate about data and finding creative ways to solve problems and portray information.
- **Communicative and collaborative:** Data scientists must be able to articulate the business value in a clear way and collaboratively work with other groups, including project sponsors and key stakeholders.

FIGURE 1-13 *Profile of a Data Scientist*

Data scientists are generally comfortable using this blend of skills to acquire, manage, analyze, and visualize data and tell compelling stories about it. The next section includes examples of what Data Science teams have created to drive new value or innovation with Big Data.

1.4 Examples of Big Data Analytics

After describing the emerging Big Data ecosystem and new roles needed to support its growth, this section provides three examples of Big Data Analytics in different areas: retail, IT infrastructure, and social media.

As mentioned earlier, Big Data presents many opportunities to improve sales and marketing analytics. An example of this is the U.S. retailer Target. Charles Duhigg's book *The Power of Habit* [4] discusses how Target used Big Data and advanced analytical methods to drive new revenue. After analyzing consumer-purchasing behavior, Target's statisticians determined that the retailer made a great deal of money from three main life-event situations.

- Marriage, when people tend to buy many new products
- Divorce, when people buy new products and change their spending habits
- Pregnancy, when people have many new things to buy and have an urgency to buy them

Target determined that the most lucrative of these life-events is the third situation: pregnancy. Using data collected from shoppers, Target was able to identify this fact and predict which of its shoppers were pregnant. In one case, Target knew a female shopper was pregnant even before her family knew [5]. This kind of knowledge allowed Target to offer specific coupons and incentives to their pregnant shoppers. In fact, Target could not only determine if a shopper was pregnant, but in which month of pregnancy a shopper may be. This enabled Target to manage its inventory, knowing that there would be demand for specific products and it would likely vary by month over the coming nine- to ten-month cycles.

Hadoop [6] represents another example of Big Data innovation on the IT infrastructure. Apache Hadoop is an open source framework that allows companies to process vast amounts of information in a highly parallelized way. Hadoop represents a specific implementation of the MapReduce paradigm and was designed by Doug Cutting and Mike Cafarella in 2005 to use data with varying structures. It is an ideal technical framework for many Big Data projects, which rely on large or unwieldy datasets with unconventional data structures. One of the main benefits of Hadoop is that it employs a distributed file system, meaning it can use a distributed cluster of servers and commodity hardware to process large amounts of data. Some of the most common examples of Hadoop implementations are in the social media space, where Hadoop can manage transactions, give textual updates, and develop social graphs among millions of users. Twitter and Facebook generate massive amounts of unstructured data and use Hadoop and its ecosystem of tools to manage this high volume. Hadoop and its ecosystem are covered in Chapter 10, "Advanced Analytics—Technology and Tools: MapReduce and Hadoop."

Finally, social media represents a tremendous opportunity to leverage social and professional interactions to derive new insights. LinkedIn exemplifies a company in which data itself is the product. Early on, LinkedIn founder Reid Hoffman saw the opportunity to create a social network for working professionals.

As of 2014, LinkedIn has more than 250 million user accounts and has added many additional features and data-related products, such as recruiting, job seeker tools, advertising, and InMaps, which show a social graph of a user's professional network. Figure 1-14 is an example of an InMap visualization that enables a LinkedIn user to get a broader view of the interconnectedness of his contacts and understand how he knows most of them.

Figure 1-14 *Data visualization of a user's social network using InMaps*

Summary

Big Data comes from myriad sources, including social media, sensors, the Internet of Things, video surveillance, and many sources of data that may not have been considered data even a few years ago. As businesses struggle to keep up with changing market requirements, some companies are finding creative ways to apply Big Data to their growing business needs and increasingly complex problems. As organizations evolve their processes and see the opportunities that Big Data can provide, they try to move beyond traditional BI activities, such as using data to populate reports and dashboards, and move toward Data Science- driven projects that attempt to answer more open-ended and complex questions.

However, exploiting the opportunities that Big Data presents requires new data architectures, including analytic sandboxes, new ways of working, and people with new skill sets. These drivers are causing organizations to set up analytic sandboxes and build Data Science teams. Although some organizations are fortunate to have data scientists, most are not, because there is a growing talent gap that makes finding and hiring data scientists in a timely manner difficult. Still, organizations such as those in web retail, health care, genomics, new IT infrastructures, and social media are beginning to take advantage of Big Data and apply it in creative and novel ways.

Exercises

1. What are the three characteristics of Big Data, and what are the main considerations in processing Big Data?
2. What is an analytic sandbox, and why is it important?
3. Explain the differences between BI and Data Science.
4. Describe the challenges of the current analytical architecture for data scientists.
5. What are the key skill sets and behavioral characteristics of a data scientist?

Bibliography

[1] C. B. B. D. Manyika, "Big Data: The Next Frontier for Innovation, Competition, and Productivity," McKinsey Global Institute, 2011.

[2] D. R. John Gantz, "The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East," IDC, 2013.

[3] `http://www.willisresilience.com/emc-datalab` [Online].

[4] C. Duhigg, *The Power of Habit: Why We Do What We Do in Life and Business*, New York: Random House, 2012.

[5] K. Hill, "How Target Figured Out a Teen Girl Was Pregnant Before Her Father Did," Forbes, February 2012.

[6] `http://hadoop.apache.org` [Online].

2

Data Analytics Lifecycle

Key Concepts

Discovery
Data preparation
Model planning
Model execution
Communicate results
Operationalize

Data science projects differ from most traditional Business Intelligence projects and many data analysis projects in that data science projects are more exploratory in nature. For this reason, it is critical to have a process to govern them and ensure that the participants are thorough and rigorous in their approach, yet not so rigid that the process impedes exploration.

Many problems that appear huge and daunting at first can be broken down into smaller pieces or actionable phases that can be more easily addressed. Having a good process ensures a comprehensive and repeatable method for conducting analysis. In addition, it helps focus time and energy early in the process to get a clear grasp of the business problem to be solved.

A common mistake made in data science projects is rushing into data collection and analysis, which precludes spending sufficient time to plan and scope the amount of work involved, understanding requirements, or even framing the business problem properly. Consequently, participants may discover mid-stream that the project sponsors are actually trying to achieve an objective that may not match the available data, or they are attempting to address an interest that differs from what has been explicitly communicated. When this happens, the project may need to revert to the initial phases of the process for a proper discovery phase, or the project may be canceled.

Creating and documenting a process helps demonstrate rigor, which provides additional credibility to the project when the data science team shares its findings. A well-defined process also offers a common framework for others to adopt, so the methods and analysis can be repeated in the future or as new members join a team.

2.1 Data Analytics Lifecycle Overview

The Data Analytics Lifecycle is designed specifically for Big Data problems and data science projects. The lifecycle has six phases, and project work can occur in several phases at once. For most phases in the lifecycle, the movement can be either forward or backward. This iterative depiction of the lifecycle is intended to more closely portray a real project, in which aspects of the project move forward and may return to earlier stages as new information is uncovered and team members learn more about various stages of the project. This enables participants to move iteratively through the process and drive toward operationalizing the project work.

2.1.1 Key Roles for a Successful Analytics Project

In recent years, substantial attention has been placed on the emerging role of the data scientist. In October 2012, Harvard Business Review featured an article titled "Data Scientist: The Sexiest Job of the 21st Century" [1], in which experts DJ Patil and Tom Davenport described the new role and how to find and hire data scientists. More and more conferences are held annually focusing on innovation in the areas of Data Science and topics dealing with Big Data. Despite this strong focus on the emerging role of the data scientist specifically, there are actually seven key roles that need to be fulfilled for a high-functioning data science team to execute analytic projects successfully.

Figure 2-1 depicts the various roles and key stakeholders of an analytics project. Each plays a critical part in a successful analytics project. Although seven roles are listed, fewer or more people can accomplish the work depending on the scope of the project, the organizational structure, and the skills of the participants. For example, on a small, versatile team, these seven roles may be fulfilled by only 3 people, but a very large project may require 20 or more people. The seven roles follow.

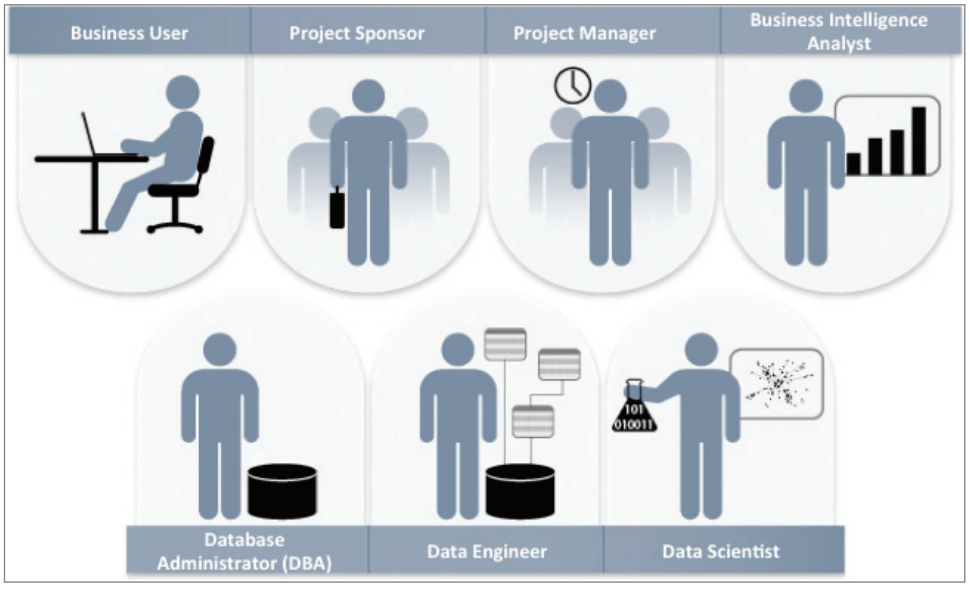


FIGURE 2-1 *Key roles for a successful analytics project*

- **Business User:** Someone who understands the domain area and usually benefits from the results. This person can consult and advise the project team on the context of the project, the value of the results, and how the outputs will be operationalized. Usually a business analyst, line manager, or deep subject matter expert in the project domain fulfills this role.
- **Project Sponsor:** Responsible for the genesis of the project. Provides the impetus and requirements for the project and defines the core business problem. Generally provides the funding and gauges the degree of value from the final outputs of the working team. This person sets the priorities for the project and clarifies the desired outputs.
- **Project Manager:** Ensures that key milestones and objectives are met on time and at the expected quality.
- **Business Intelligence Analyst:** Provides business domain expertise based on a deep understanding of the data, key performance indicators (KPIs), key metrics, and business intelligence from a reporting perspective. Business Intelligence Analysts generally create dashboards and reports and have knowledge of the data feeds and sources.
- **Database Administrator (DBA):** Provisions and configures the database environment to support the analytics needs of the working team. These responsibilities may include providing access to key databases or tables and ensuring the appropriate security levels are in place related to the data repositories.
- **Data Engineer:** Leverages deep technical skills to assist with tuning SQL queries for data management and data extraction, and provides support for data ingestion into the analytic sandbox, which

was discussed in Chapter 1, "Introduction to Big Data Analytics." Whereas the DBA sets up and configures the databases to be used, the data engineer executes the actual data extractions and performs substantial data manipulation to facilitate the analytics. The data engineer works closely with the data scientist to help shape data in the right ways for analyses.

- **Data Scientist:** Provides subject matter expertise for analytical techniques, data modeling, and applying valid analytical techniques to given business problems. Ensures overall analytics objectives are met. Designs and executes analytical methods and approaches with the data available to the project.

Although most of these roles are not new, the last two roles—data engineer and data scientist—have become popular and in high demand [2] as interest in Big Data has grown.

2.1.2 Background and Overview of Data Analytics Lifecycle

The Data Analytics Lifecycle defines analytics process best practices spanning discovery to project completion. The lifecycle draws from established methods in the realm of data analytics and decision science. This synthesis was developed after gathering input from data scientists and consulting established approaches that provided input on pieces of the process. Several of the processes that were consulted include these:

- **Scientific method** [3], in use for centuries, still provides a solid framework for thinking about and deconstructing problems into their principal parts. One of the most valuable ideas of the scientific method relates to forming hypotheses and finding ways to test ideas.
- **CRISP-DM** [4] provides useful input on ways to frame analytics problems and is a popular approach for data mining.
- Tom Davenport's **DELTA** framework [5]: The DELTA framework offers an approach for data analytics projects, including the context of the organization's skills, datasets, and leadership engagement.
- Doug Hubbard's **Applied Information Economics (AIE)** approach [6]: AIE provides a framework for measuring intangibles and provides guidance on developing decision models, calibrating expert estimates, and deriving the expected value of information.
- **"MAD Skills"** by Cohen et al. [7] offers input for several of the techniques mentioned in Phases 2–4 that focus on model planning, execution, and key findings.

Figure 2-2 presents an overview of the Data Analytics Lifecycle that includes six phases. Teams commonly learn new things in a phase that cause them to go back and refine the work done in prior phases based on new insights and information that have been uncovered. For this reason, Figure 2-2 is shown as a cycle. The circular arrows convey iterative movement between phases until the team members have sufficient information to move to the next phase. The callouts include sample questions to ask to help guide whether each of the team members has enough information and has made enough progress to move to the next phase of the process. Note that these phases do not represent formal stage gates; rather, they serve as criteria to help test whether it makes sense to stay in the current phase or move to the next.

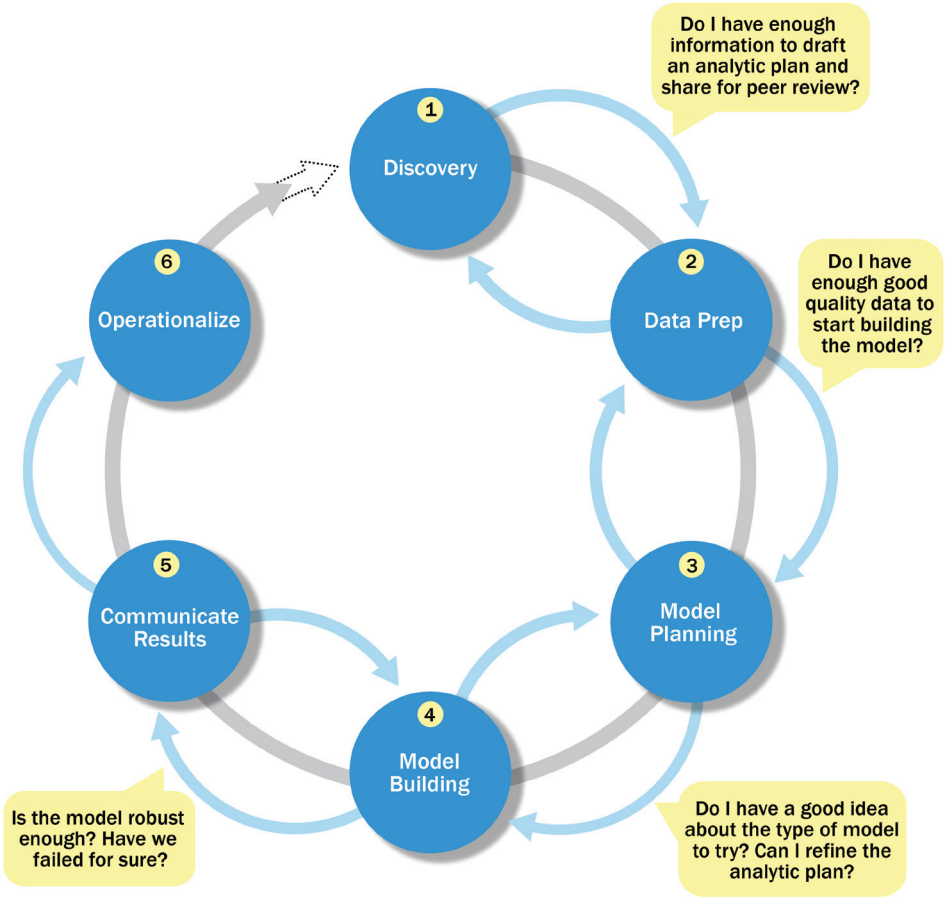


FIGURE 2-2 *Overview of Data Analytics Lifecycle*

Here is a brief overview of the main phases of the Data Analytics Lifecycle:

- **Phase 1—Discovery:** In Phase 1, the team learns the business domain, including relevant history such as whether the organization or business unit has attempted similar projects in the past from which they can learn. The team assesses the resources available to support the project in terms of people, technology, time, and data. Important activities in this phase include framing the business problem as an analytics challenge that can be addressed in subsequent phases and formulating initial hypotheses (IHs) to test and begin learning the data.
- **Phase 2—Data preparation:** Phase 2 requires the presence of an analytic sandbox, in which the team can work with data and perform analytics for the duration of the project. The team needs to execute extract, load, and transform (ELT) or extract, transform and load (ETL) to get data into the sandbox. The ELT and ETL are sometimes abbreviated as ETLT. Data should be transformed in the ETLT process so the team can work with it and analyze it. In this phase, the team also needs to familiarize itself with the data thoroughly and take steps to condition the data (Section 2.3.4).

- **Phase 3—Model planning:** Phase 3 is model planning, where the team determines the methods, techniques, and workflow it intends to follow for the subsequent model building phase. The team explores the data to learn about the relationships between variables and subsequently selects key variables and the most suitable models.
- **Phase 4—Model building:** In Phase 4, the team develops datasets for testing, training, and production purposes. In addition, in this phase the team builds and executes models based on the work done in the model planning phase. The team also considers whether its existing tools will suffice for running the models, or if it will need a more robust environment for executing models and workflows (for example, fast hardware and parallel processing, if applicable).
- **Phase 5—Communicate results:** In Phase 5, the team, in collaboration with major stakeholders, determines if the results of the project are a success or a failure based on the criteria developed in Phase 1. The team should identify key findings, quantify the business value, and develop a narrative to summarize and convey findings to stakeholders.
- **Phase 6—Operationalize:** In Phase 6, the team delivers final reports, briefings, code, and technical documents. In addition, the team may run a pilot project to implement the models in a production environment.

Once team members have run models and produced findings, it is critical to frame these results in a way that is tailored to the audience that engaged the team. Moreover, it is critical to frame the results of the work in a manner that demonstrates clear value. If the team performs a technically accurate analysis but fails to translate the results into a language that resonates with the audience, people will not see the value, and much of the time and effort on the project will have been wasted.

The rest of the chapter is organized as follows. Sections 2.2–2.7 discuss in detail how each of the six phases works, and Section 2.8 shows a case study of incorporating the Data Analytics Lifecycle in a real-world data science project.

2.2 Phase 1: Discovery

The first phase of the Data Analytics Lifecycle involves discovery (Figure 2-3). In this phase, the data science team must learn and investigate the problem, develop context and understanding, and learn about the data sources needed and available for the project. In addition, the team formulates initial hypotheses that can later be tested with data.

2.2.1 Learning the Business Domain

Understanding the domain area of the problem is essential. In many cases, data scientists will have deep computational and quantitative knowledge that can be broadly applied across many disciplines. An example of this role would be someone with an advanced degree in applied mathematics or statistics.

These data scientists have deep knowledge of the methods, techniques, and ways for applying heuristics to a variety of business and conceptual problems. Others in this area may have deep knowledge of a domain area, coupled with quantitative expertise. An example of this would be someone with a Ph.D. in life sciences. This person would have deep knowledge of a field of study, such as oceanography, biology, or genetics, with some depth of quantitative knowledge.

At this early stage in the process, the team needs to determine how much business or domain knowledge the data scientist needs to develop models in Phases 3 and 4. The earlier the team can make this assessment

the better, because the decision helps dictate the resources needed for the project team and ensures the team has the right balance of domain knowledge and technical expertise.

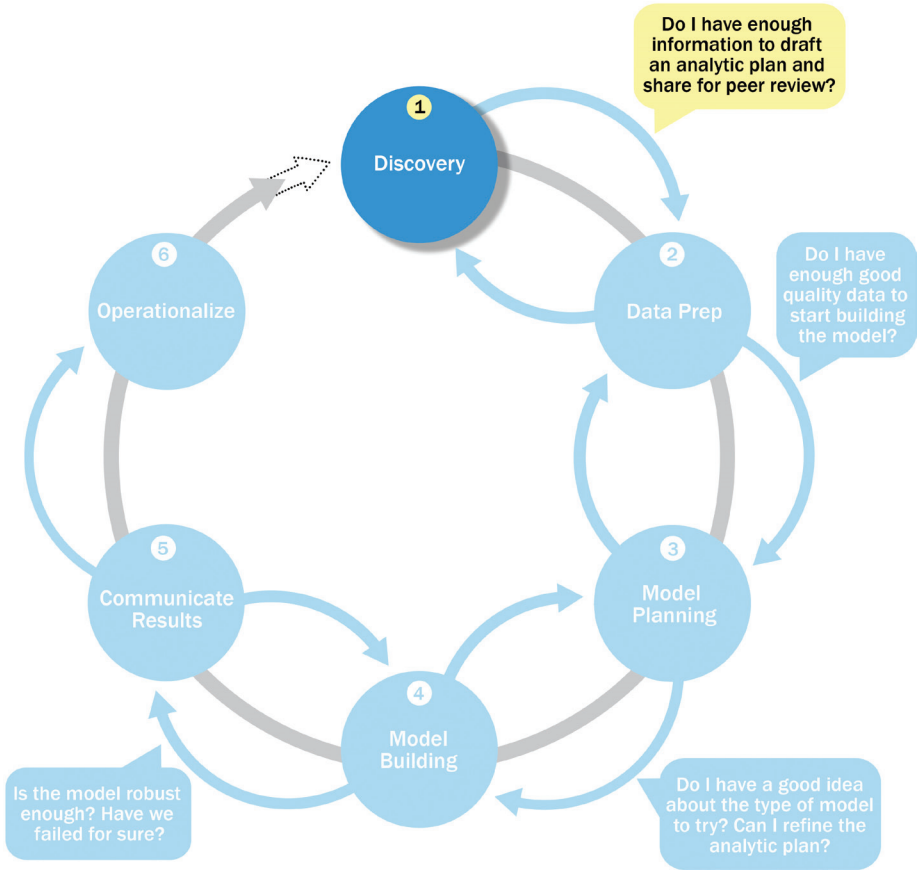


FIGURE 2-3 *Discovery phase*

2.2.2 Resources

As part of the discovery phase, the team needs to assess the resources available to support the project. In this context, resources include technology, tools, systems, data, and people.

During this scoping, consider the available tools and technology the team will be using and the types of systems needed for later phases to operationalize the models. In addition, try to evaluate the level of analytical sophistication within the organization and gaps that may exist related to tools, technology, and skills. For instance, for the model being developed to have longevity in an organization, consider what types of skills and roles will be required that may not exist today. For the project to have long-term success,

what types of skills and roles will be needed for the recipients of the model being developed? Does the requisite level of expertise exist within the organization today, or will it need to be cultivated? Answering these questions will influence the techniques the team selects and the kind of implementation the team chooses to pursue in subsequent phases of the Data Analytics Lifecycle.

In addition to the skills and computing resources, it is advisable to take inventory of the types of data available to the team for the project. Consider if the data available is sufficient to support the project's goals. The team will need to determine whether it must collect additional data, purchase it from outside sources, or transform existing data. Often, projects are started looking only at the data available. When the data is less than hoped for, the size and scope of the project is reduced to work within the constraints of the existing data.

An alternative approach is to consider the long-term goals of this kind of project, without being constrained by the current data. The team can then consider what data is needed to reach the long-term goals and which pieces of this multistep journey can be achieved today with the existing data. Considering longer-term goals along with short-term goals enables teams to pursue more ambitious projects and treat a project as the first step of a more strategic initiative, rather than as a standalone initiative. It is critical to view projects as part of a longer-term journey, especially if executing projects in an organization that is new to Data Science and may not have embarked on the optimum datasets to support robust analyses up to this point.

Ensure the project team has the right mix of domain experts, customers, analytic talent, and project management to be effective. In addition, evaluate how much time is needed and if the team has the right breadth and depth of skills.

After taking inventory of the tools, technology, data, and people, consider if the team has sufficient resources to succeed on this project, or if additional resources are needed. Negotiating for resources at the outset of the project, while scoping the goals, objectives, and feasibility, is generally more useful than later in the process and ensures sufficient time to execute it properly. Project managers and key stakeholders have better success negotiating for the right resources at this stage rather than later once the project is underway.

2.2.3 Framing the Problem

Framing the problem well is critical to the success of the project. ***Framing*** is the process of stating the analytics problem to be solved. At this point, it is a best practice to write down the problem statement and share it with the key stakeholders. Each team member may hear slightly different things related to the needs and the problem and have somewhat different ideas of possible solutions. For these reasons, it is crucial to state the analytics problem, as well as why and to whom it is important. Essentially, the team needs to clearly articulate the current situation and its main challenges.

As part of this activity, it is important to identify the main objectives of the project, identify what needs to be achieved in business terms, and identify what needs to be done to meet the needs. Additionally, consider the objectives and the success criteria for the project. What is the team attempting to achieve by doing the project, and what will be considered "good enough" as an outcome of the project? This is critical to document and share with the project team and key stakeholders. It is best practice to share the statement of goals and success criteria with the team and confirm alignment with the project sponsor's expectations.

Perhaps equally important is to establish failure criteria. Most people doing projects prefer only to think of the success criteria and what the conditions will look like when the participants are successful. However, this is almost taking a best-case scenario approach, assuming that everything will proceed as planned

and the project team will reach its goals. However, no matter how well planned, it is almost impossible to plan for everything that will emerge in a project. The failure criteria will guide the team in understanding when it is best to stop trying or settle for the results that have been gleaned from the data. Many times people will continue to perform analyses past the point when any meaningful insights can be drawn from the data. Establishing criteria for both success and failure helps the participants avoid unproductive effort and remain aligned with the project sponsors

2.2.4 Identifying Key Stakeholders

Another important step is to identify the key stakeholders and their interests in the project. During these discussions, the team can identify the success criteria, key risks, and stakeholders, which should include anyone who will benefit from the project or will be significantly impacted by the project. When interviewing stakeholders, learn about the domain area and any relevant history from similar analytics projects. For example, the team may identify the results each stakeholder wants from the project and the criteria it will use to judge the success of the project.

Keep in mind that the analytics project is being initiated for a reason. It is critical to articulate the pain points as clearly as possible to address them and be aware of areas to pursue or avoid as the team gets further into the analytical process. Depending on the number of stakeholders and participants, the team may consider outlining the type of activity and participation expected from each stakeholder and participant. This will set clear expectations with the participants and avoid delays later when, for example, the team may feel it needs to wait for approval from someone who views himself as an adviser rather than an approver of the work product.

2.2.5 Interviewing the Analytics Sponsor

The team should plan to collaborate with the stakeholders to clarify and frame the analytics problem. At the outset, project sponsors may have a predetermined solution that may not necessarily realize the desired outcome. In these cases, the team must use its knowledge and expertise to identify the true underlying problem and appropriate solution.

For instance, suppose in the early phase of a project, the team is told to create a recommender system for the business and that the way to do this is by speaking with three people and integrating the product recommender into a legacy corporate system. Although this may be a valid approach, it is important to test the assumptions and develop a clear understanding of the problem. The data science team typically may have a more objective understanding of the problem set than the stakeholders, who may be suggesting solutions to a given problem. Therefore, the team can probe deeper into the context and domain to clearly define the problem and propose possible paths from the problem to a desired outcome. In essence, the data science team can take a more objective approach, as the stakeholders may have developed biases over time, based on their experience. Also, what may have been true in the past may no longer be a valid working assumption. One possible way to circumvent this issue is for the project sponsor to focus on clearly defining the requirements, while the other members of the data science team focus on the methods needed to achieve the goals.

When interviewing the main stakeholders, the team needs to take time to thoroughly interview the project sponsor, who tends to be the one funding the project or providing the high-level requirements. This person understands the problem and usually has an idea of a potential working solution. It is critical

to thoroughly understand the sponsor's perspective to guide the team in getting started on the project. Here are some tips for interviewing project sponsors:

- Prepare for the interview; draft questions, and review with colleagues.
- Use open-ended questions; avoid asking leading questions.
- Probe for details and pose follow-up questions.
- Avoid filling every silence in the conversation; give the other person time to think.
- Let the sponsors express their ideas and ask clarifying questions, such as "Why? Is that correct? Is this idea on target? Is there anything else?"
- Use active listening techniques; repeat back what was heard to make sure the team heard it correctly, or reframe what was said.
- Try to avoid expressing the team's opinions, which can introduce bias; instead, focus on listening.
- Be mindful of the body language of the interviewers and stakeholders; use eye contact where appropriate, and be attentive.
- Minimize distractions.
- Document what the team heard, and review it with the sponsors.

Following is a brief list of common questions that are helpful to ask during the discovery phase when interviewing the project sponsor. The responses will begin to shape the scope of the project and give the team an idea of the goals and objectives of the project.

- What business problem is the team trying to solve?
- What is the desired outcome of the project?
- What data sources are available?
- What industry issues may impact the analysis?
- What timelines need to be considered?
- Who could provide insight into the project?
- Who has final decision-making authority on the project?
- How will the focus and scope of the problem change if the following dimensions change:
  - **Time:** Analyzing 1 year or 10 years' worth of data?
  - **People:** Assess impact of changes in resources on project timeline.
  - **Risk:** Conservative to aggressive
  - **Resources:** None to unlimited (tools, technology, systems)
  - **Size and attributes of data:** Including internal and external data sources

2.2.6 Developing Initial Hypotheses

Developing a set of IHs is a key facet of the discovery phase. This step involves forming ideas that the team can test with data. Generally, it is best to come up with a few primary hypotheses to test and then be creative about developing several more. These IHs form the basis of the analytical tests the team will use in later phases and serve as the foundation for the findings in Phase 5. Hypothesis testing from a statistical perspective is covered in greater detail in Chapter 3, "Review of Basic Data Analytic Methods Using R."

In this way, the team can compare its answers with the outcome of an experiment or test to generate additional possible solutions to problems. As a result, the team will have a much richer set of observations to choose from and more choices for agreeing upon the most impactful conclusions from a project.

Another part of this process involves gathering and assessing hypotheses from stakeholders and domain experts who may have their own perspective on what the problem is, what the solution should be, and how to arrive at a solution. These stakeholders would know the domain area well and can offer suggestions on ideas to test as the team formulates hypotheses during this phase. The team will likely collect many ideas that may illuminate the operating assumptions of the stakeholders. These ideas will also give the team opportunities to expand the project scope into adjacent spaces where it makes sense or design experiments in a meaningful way to address the most important interests of the stakeholders. As part of this exercise, it can be useful to obtain and explore some initial data to inform discussions with stakeholders during the hypothesis-forming stage.

2.2.7 Identifying Potential Data Sources

As part of the discovery phase, identify the kinds of data the team will need to solve the problem. Consider the volume, type, and time span of the data needed to test the hypotheses. Ensure that the team can access more than simply aggregated data. In most cases, the team will need the raw data to avoid introducing bias for the downstream analysis. Recalling the characteristics of Big Data from Chapter 1, assess the main characteristics of the data, with regard to its volume, variety, and velocity of change. A thorough diagnosis of the data situation will influence the kinds of tools and techniques to use in Phases 2-4 of the Data Analytics Lifecycle. In addition, performing data exploration in this phase will help the team determine the amount of data needed, such as the amount of historical data to pull from existing systems and the data structure. Develop an idea of the scope of the data needed, and validate that idea with the domain experts on the project.

The team should perform five main activities during this step of the discovery phase:

- **Identify data sources:** Make a list of candidate data sources the team may need to test the initial hypotheses outlined in this phase. Make an inventory of the datasets currently available and those that can be purchased or otherwise acquired for the tests the team wants to perform.
- **Capture aggregate data sources:** This is for previewing the data and providing high-level understanding. It enables the team to gain a quick overview of the data and perform further exploration on specific areas. It also points the team to possible areas of interest within the data.
- **Review the raw data:** Obtain preliminary data from initial data feeds. Begin understanding the interdependencies among the data attributes, and become familiar with the content of the data, its quality, and its limitations.

- **Evaluate the data structures and tools needed:** The data type and structure dictate which tools the team can use to analyze the data. This evaluation gets the team thinking about which technologies may be good candidates for the project and how to start getting access to these tools.
- **Scope the sort of data infrastructure needed for this type of problem:** In addition to the tools needed, the data influences the kind of infrastructure that's required, such as disk storage and network capacity.

Unlike many traditional stage-gate processes, in which the team can advance only when specific criteria are met, the Data Analytics Lifecycle is intended to accommodate more ambiguity. This more closely reflects how data science projects work in real-life situations. For each phase of the process, it is recommended to pass certain checkpoints as a way of gauging whether the team is ready to move to the next phase of the Data Analytics Lifecycle.

The team can move to the next phase when it has enough information to draft an analytics plan and share it for peer review. Although a peer review of the plan may not actually be required by the project, creating the plan is a good test of the team's grasp of the business problem and the team's approach to addressing it. Creating the analytic plan also requires a clear understanding of the domain area, the problem to be solved, and scoping of the data sources to be used. Developing success criteria early in the project clarifies the problem definition and helps the team when it comes time to make choices about the analytical methods being used in later phases.

2.3 Phase 2: Data Preparation

The second phase of the Data Analytics Lifecycle involves data preparation, which includes the steps to explore, preprocess, and condition data prior to modeling and analysis. In this phase, the team needs to create a robust environment in which it can explore the data that is separate from a production environment. Usually, this is done by preparing an analytics sandbox. To get the data into the sandbox, the team needs to perform ETLT, by a combination of extracting, transforming, and loading data into the sandbox. Once the data is in the sandbox, the team needs to learn about the data and become familiar with it. Understanding the data in detail is critical to the success of the project. The team also must decide how to condition and transform data to get it into a format to facilitate subsequent analysis. The team may perform data visualizations to help team members understand the data, including its trends, outliers, and relationships among data variables. Each of these steps of the data preparation phase is discussed throughout this section.

Data preparation tends to be the most labor-intensive step in the analytics lifecycle. In fact, it is common for teams to spend at least 50% of a data science project's time in this critical phase. If the team cannot obtain enough data of sufficient quality, it may be unable to perform the subsequent steps in the lifecycle process.

Figure 2-4 shows an overview of the Data Analytics Lifecycle for Phase 2. The data preparation phase is generally the most iterative and the one that teams tend to underestimate most often. This is because most teams and leaders are anxious to begin analyzing the data, testing hypotheses, and getting answers to some of the questions posed in Phase 1. Many tend to jump into Phase 3 or Phase 4 to begin rapidly developing models and algorithms without spending the time to prepare the data for modeling. Consequently, teams come to realize the data they are working with does not allow them to execute the models they want, and they end up back in Phase 2 anyway.

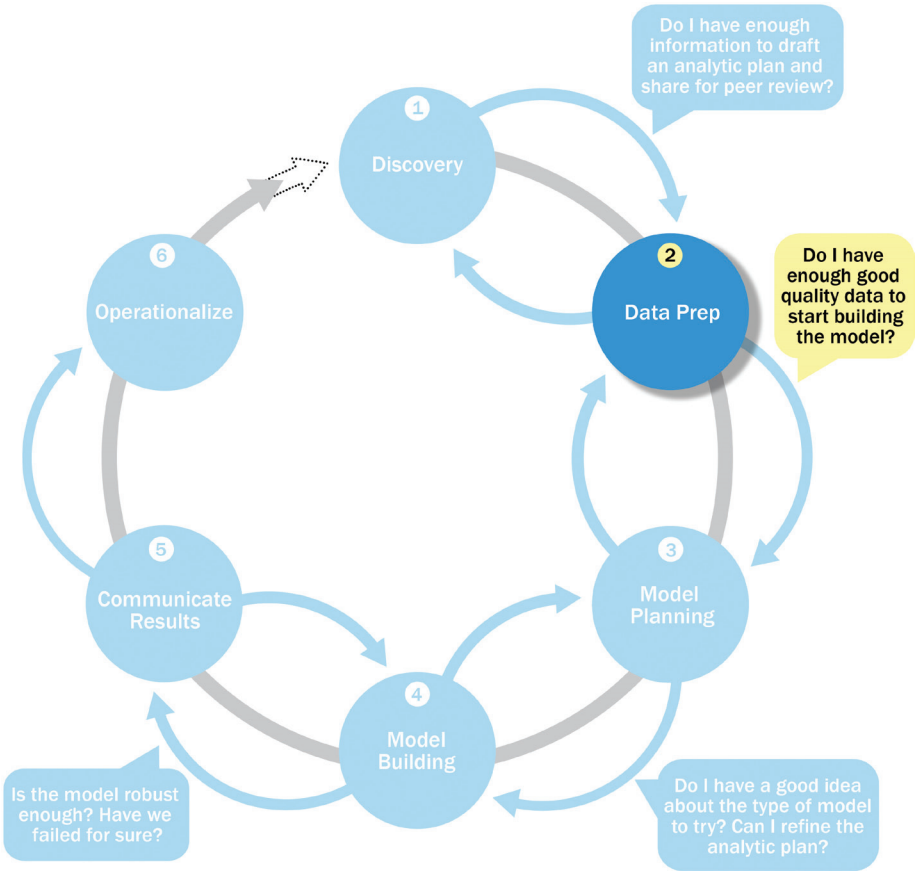


FIGURE 2-4 *Data preparation phase*

2.3.1 Preparing the Analytic Sandbox

The first subphase of data preparation requires the team to obtain an analytic sandbox (also commonly referred to as a ***workspace***), in which the team can explore the data without interfering with live production databases. Consider an example in which the team needs to work with a company's financial data. The team should access a copy of the financial data from the analytic sandbox rather than interacting with the production version of the organization's main database, because that will be tightly controlled and needed for financial reporting.

When developing the analytic sandbox, it is a best practice to collect all kinds of data there, as team members need access to high volumes and varieties of data for a Big Data analytics project. This can include

everything from summary-level aggregated data, structured data, raw data feeds, and unstructured text data from call logs or web logs, depending on the kind of analysis the team plans to undertake.

This expansive approach for attracting data of all kind differs considerably from the approach advocated by many information technology (IT) organizations. Many IT groups provide access to only a particular subsegment of the data for a specific purpose. Often, the mindset of the IT group is to provide the minimum amount of data required to allow the team to achieve its objectives. Conversely, the data science team wants access to everything. From its perspective, more data is better, as oftentimes data science projects are a mixture of purpose-driven analyses and experimental approaches to test a variety of ideas. In this context, it can be challenging for a data science team if it has to request access to each and every dataset and attribute one at a time. Because of these differing views on data access and use, it is critical for the data science team to collaborate with IT, make clear what it is trying to accomplish, and align goals.

During these discussions, the data science team needs to give IT a justification to develop an analytics sandbox, which is separate from the traditional IT-governed data warehouses within an organization. Successfully and amicably balancing the needs of both the data science team and IT requires a positive working relationship between multiple groups and data owners. The payoff is great. The analytic sandbox enables organizations to undertake more ambitious data science projects and move beyond doing traditional data analysis and Business Intelligence to perform more robust and advanced predictive analytics.

Expect the sandbox to be large. It may contain raw data, aggregated data, and other data types that are less commonly used in organizations. Sandbox size can vary greatly depending on the project. A good rule is to plan for the sandbox to be at least 5–10 times the size of the original datasets, partly because copies of the data may be created that serve as specific tables or data stores for specific kinds of analysis in the project.

Although the concept of an analytics sandbox is relatively new, companies are making progress in this area and are finding ways to offer sandboxes and workspaces where teams can access datasets and work in a way that is acceptable to both the data science teams and the IT groups.

2.3.2 Performing ETLT

As the team looks to begin data transformations, make sure the analytics sandbox has ample bandwidth and reliable network connections to the underlying data sources to enable uninterrupted read and write. In ETL, users perform extract, transform, load processes to extract data from a datastore, perform data transformations, and load the data back into the datastore. However, the analytic sandbox approach differs slightly; it advocates extract, load, and then transform. In this case, the data is extracted in its raw form and loaded into the datastore, where analysts can choose to transform the data into a new state or leave it in its original, raw condition. The reason for this approach is that there is significant value in preserving the raw data and including it in the sandbox before any transformations take place.

For instance, consider an analysis for fraud detection on credit card usage. Many times, outliers in this data population can represent higher-risk transactions that may be indicative of fraudulent credit card activity. Using ETL, these outliers may be inadvertently filtered out or transformed and cleaned before being loaded into the datastore. In this case, the very data that would be needed to evaluate instances of fraudulent activity would be inadvertently cleansed, preventing the kind of analysis that a team would want to do.

Following the ELT approach gives the team access to clean data to analyze after the data has been loaded into the database and gives access to the data in its original form for finding hidden nuances in the data. This approach is part of the reason that the analytic sandbox can quickly grow large. The team may want clean data and aggregated data and may need to keep a copy of the original data to compare against or

look for hidden patterns that may have existed in the data before the cleaning stage. This process can be summarized as ETLT to reflect the fact that a team may choose to perform ETL in one case and ELT in another.

Depending on the size and number of the data sources, the team may need to consider how to parallelize the movement of the datasets into the sandbox. For this purpose, moving large amounts of data is sometimes referred to as Big ETL. The data movement can be parallelized by technologies such as Hadoop or MapReduce, which will be explained in greater detail in Chapter 10, "Advanced Analytics—Technology and Tools: MapReduce and Hadoop." At this point, keep in mind that these technologies can be used to perform parallel data ingest and introduce a huge number of files or datasets in parallel in a very short period of time. Hadoop can be useful for data loading as well as for data analysis in subsequent phases.

Prior to moving the data into the analytic sandbox, determine the transformations that need to be performed on the data. Part of this phase involves assessing data quality and structuring the datasets properly so they can be used for robust analysis in subsequent phases. In addition, it is important to consider which data the team will have access to and which new data attributes will need to be derived in the data to enable analysis.

As part of the ETLT step, it is advisable to make an inventory of the data and compare the data currently available with datasets the team needs. Performing this sort of gap analysis provides a framework for understanding which datasets the team can take advantage of today and where the team needs to initiate projects for data collection or access to new datasets currently unavailable. A component of this subphase involves extracting data from the available sources and determining data connections for raw data, online transaction processing (OLTP) databases, online analytical processing (OLAP) cubes, or other data feeds.

Application programming interface (API) is an increasingly popular way to access a data source [8]. Many websites and social network applications now provide APIs that offer access to data to support a project or supplement the datasets with which a team is working. For example, connecting to the Twitter API can enable a team to download millions of tweets to perform a project for sentiment analysis on a product, a company, or an idea. Much of the Twitter data is publicly available and can augment other datasets used on the project.

2.3.3 Learning About the Data

A critical aspect of a data science project is to become familiar with the data itself. Spending time to learn the nuances of the datasets provides context to understand what constitutes a reasonable value and expected output versus what is a surprising finding. In addition, it is important to catalog the data sources that the team has access to and identify additional data sources that the team can leverage but perhaps does not have access to today. Some of the activities in this step may overlap with the initial investigation of the datasets that occur in the discovery phase. Doing this activity accomplishes several goals.

- Clarifies the data that the data science team has access to at the start of the project
- Highlights gaps by identifying datasets within an organization that the team may find useful but may not be accessible to the team today. As a consequence, this activity can trigger a project to begin building relationships with the data owners and finding ways to share data in appropriate ways. In addition, this activity may provide an impetus to begin collecting new data that benefits the organization or a specific long-term project.
- Identifies datasets outside the organization that may be useful to obtain, through open APIs, data sharing, or purchasing data to supplement already existing datasets

Table 2-1 demonstrates one way to organize this type of data inventory.

TABLE 2-1 *Sample Dataset Inventory*

Dataset	Data Available and Accessible	Data Available, but not Accessible	Data to Collect	Data to Obtain from Third Party Sources
Products shipped	●			
Product Financials		●		
Product Call Center Data		●		
Live Product Feedback Surveys			●	
Product Sentiment from Social Media				●

2.3.4 Data Conditioning

Data conditioning refers to the process of cleaning data, normalizing datasets, and performing transformations on the data. A critical step within the Data Analytics Lifecycle, data conditioning can involve many complex steps to join or merge datasets or otherwise get datasets into a state that enables analysis in further phases. Data conditioning is often viewed as a preprocessing step for the data analysis because it involves many operations on the dataset before developing models to process or analyze the data. This implies that the data-conditioning step is performed only by IT, the data owners, a DBA, or a data engineer. However, it is also important to involve the data scientist in this step because many decisions are made in the data conditioning phase that affect subsequent analysis. Part of this phase involves deciding which aspects of particular datasets will be useful to analyze in later steps. Because teams begin forming ideas in this phase about which data to keep and which data to transform or discard, it is important to involve multiple team members in these decisions. Leaving such decisions to a single person may cause teams to return to this phase to retrieve data that may have been discarded.

As with the previous example of deciding which data to keep as it relates to fraud detection on credit card usage, it is critical to be thoughtful about which data the team chooses to keep and which data will be discarded. This can have far-reaching consequences that will cause the team to retrace previous steps if the team discards too much of the data at too early a point in this process. Typically, data science teams would rather keep more data than too little data for the analysis. Additional questions and considerations for the data conditioning step include these.

- What are the data sources? What are the target fields (for example, columns of the tables)?
- How clean is the data?

- How consistent are the contents and files? Determine to what degree the data contains missing or inconsistent values and if the data contains values deviating from normal.
- Assess the consistency of the data types. For instance, if the team expects certain data to be numeric, confirm it is numeric or if it is a mixture of alphanumeric strings and text.
- Review the content of data columns or other inputs, and check to ensure they make sense. For instance, if the project involves analyzing income levels, preview the data to confirm that the income values are positive or if it is acceptable to have zeros or negative values.
- Look for any evidence of systematic error. Examples include data feeds from sensors or other data sources breaking without anyone noticing, which causes invalid, incorrect, or missing data values. In addition, review the data to gauge if the definition of the data is the same over all measurements. In some cases, a data column is repurposed, or the column stops being populated, without this change being annotated or without others being notified.

2.3.5 Survey and Visualize

After the team has collected and obtained at least some of the datasets needed for the subsequent analysis, a useful step is to leverage data visualization tools to gain an overview of the data. Seeing high-level patterns in the data enables one to understand characteristics about the data very quickly. One example is using data visualization to examine data quality, such as whether the data contains many unexpected values or other indicators of dirty data. (Dirty data will be discussed further in Chapter 3.) Another example is skewness, such as if the majority of the data is heavily shifted toward one value or end of a continuum.

Shneiderman [9] is well known for his mantra for visual data analysis of "overview first, zoom and filter, then details-on-demand." This is a pragmatic approach to visual data analysis. It enables the user to find areas of interest, zoom and filter to find more detailed information about a particular area of the data, and then find the detailed data behind a particular area. This approach provides a high-level view of the data and a great deal of information about a given dataset in a relatively short period of time.

When pursuing this approach with a data visualization tool or statistical package, the following guidelines and considerations are recommended.

- Review data to ensure that calculations remained consistent within columns or across tables for a given data field. For instance, did customer lifetime value change at some point in the middle of data collection? Or if working with financials, did the interest calculation change from simple to compound at the end of the year?
- Does the data distribution stay consistent over all the data? If not, what kinds of actions should be taken to address this problem?
- Assess the granularity of the data, the range of values, and the level of aggregation of the data.
- Does the data represent the population of interest? For marketing data, if the project is focused on targeting customers of child-rearing age, does the data represent that, or is it full of senior citizens and teenagers?
- For time-related variables, are the measurements daily, weekly, monthly? Is that good enough? Is time measured in seconds everywhere? Or is it in milliseconds in some places? Determine the level of granularity of the data needed for the analysis, and assess whether the current level of timestamps on the data meets that need.

- Is the data standardized/normalized? Are the scales consistent? If not, how consistent or irregular is the data?
- For geospatial datasets, are state or country abbreviations consistent across the data? Are personal names normalized? English units? Metric units?

These are typical considerations that should be part of the thought process as the team evaluates the datasets that are obtained for the project. Becoming deeply knowledgeable about the data will be critical when it comes time to construct and run models later in the process.

2.3.6 Common Tools for the Data Preparation Phase

Several tools are commonly used for this phase:

- **Hadoop** [10] can perform massively parallel ingest and custom analysis for web traffic parsing, GPS location analytics, genomic analysis, and combining of massive unstructured data feeds from multiple sources.
- **Alpine Miner** [11] provides a graphical user interface (GUI) for creating analytic workflows, including data manipulations and a series of analytic events such as staged data-mining techniques (for example, first select the top 100 customers, and then run descriptive statistics and clustering) on Postgres SQL and other Big Data sources.
- **OpenRefine** (formerly called Google Refine) [12] is "a free, open source, powerful tool for working with messy data." It is a popular GUI-based tool for performing data transformations, and it's one of the most robust free tools currently available.
- Similar to OpenRefine, **Data Wrangler** [13] is an interactive tool for data cleaning and transformation. Wrangler was developed at Stanford University and can be used to perform many transformations on a given dataset. In addition, data transformation outputs can be put into Java or Python. The advantage of this feature is that a subset of the data can be manipulated in Wrangler via its GUI, and then the same operations can be written out as Java or Python code to be executed against the full, larger dataset offline in a local analytic sandbox.

For Phase 2, the team needs assistance from IT, DBAs, or whoever controls the Enterprise Data Warehouse (EDW) for data sources the data science team would like to use.

2.4 Phase 3: Model Planning

In Phase 3, the data science team identifies candidate models to apply to the data for clustering, classifying, or finding relationships in the data depending on the goal of the project, as shown in Figure 2-5. It is during this phase that the team refers to the hypotheses developed in Phase 1, when they first became acquainted with the data and understanding the business problems or domain area. These hypotheses help the team frame the analytics to execute in Phase 4 and select the right methods to achieve its objectives.

Some of the activities to consider in this phase include the following:

- Assess the structure of the datasets. The structure of the datasets is one factor that dictates the tools and analytical techniques for the next phase. Depending on whether the team plans to analyze textual data or transactional data, for example, different tools and approaches are required.
- Ensure that the analytical techniques enable the team to meet the business objectives and accept or reject the working hypotheses.

- Determine if the situation warrants a single model or a series of techniques as part of a larger analytic workflow. A few example models include association rules (Chapter 5, "Advanced Analytical Theory and Methods: Association Rules") and logistic regression (Chapter 6, "Advanced Analytical Theory and Methods: Regression"). Other tools, such as Alpine Miner, enable users to set up a series of steps and analyses and can serve as a front-end user interface (UI) for manipulating Big Data sources in PostgreSQL.

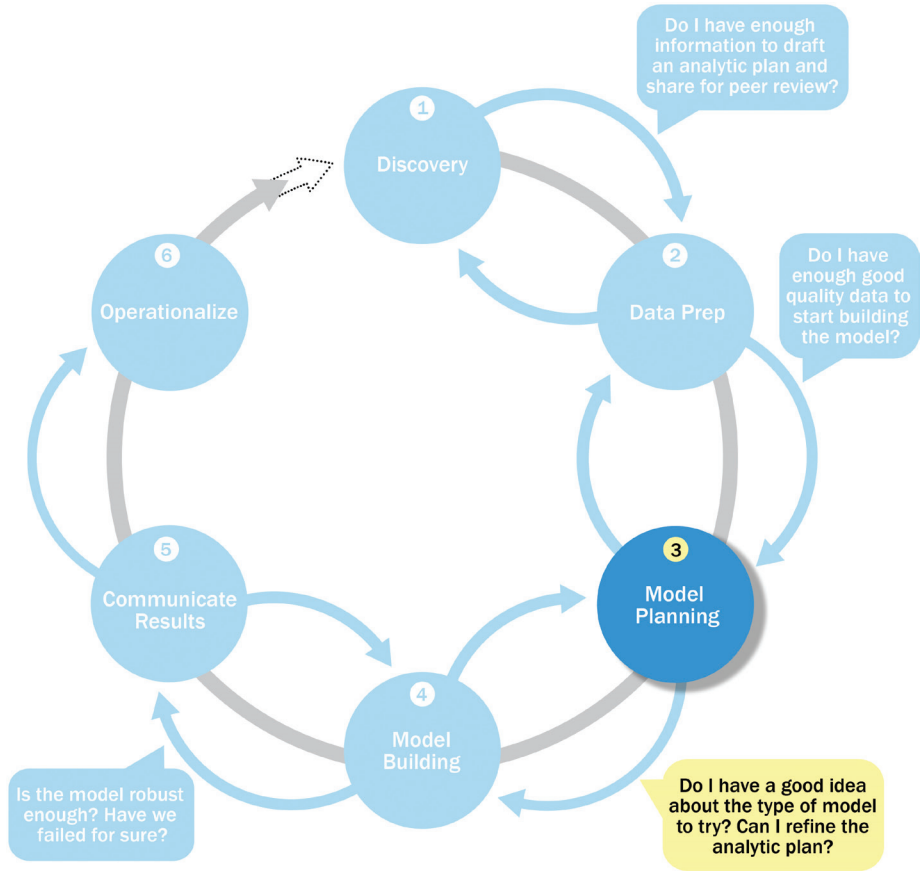


FIGURE 2-5 *Model planning phase*

In addition to the considerations just listed, it is useful to research and understand how other analysts generally approach a specific kind of problem. Given the kind of data and resources that are available, evaluate whether similar, existing approaches will work or if the team will need to create something new. Many times teams can get ideas from analogous problems that other people have solved in different industry verticals or domain areas. Table 2-2 summarizes the results of an exercise of this type, involving several domain areas and the types of models previously used in a classification type of problem after conducting research on churn models in multiple industry verticals. Performing this sort of diligence gives the team

ideas of how others have solved similar problems and presents the team with a list of candidate models to try as part of the model planning phase.

TABLE 2-2 *Research on Model Planning in Industry Verticals*

Market Sector	Analytic Techniques/Methods Used
Consumer Packaged Goods	Multiple linear regression, automatic relevance determination (ARD), and decision tree
Retail Banking	Multiple regression
Retail Business	Logistic regression, ARD, decision tree
Wireless Telecom	Neural network, decision tree, hierarchical neurofuzzy systems, rule evolver, logistic regression

2.4.1 Data Exploration and Variable Selection

Although some data exploration takes place in the data preparation phase, those activities focus mainly on data hygiene and on assessing the quality of the data itself. In Phase 3, the objective of the data exploration is to understand the relationships among the variables to inform selection of the variables and methods and to understand the problem domain. As with earlier phases of the Data Analytics Lifecycle, it is important to spend time and focus attention on this preparatory work to make the subsequent phases of model selection and execution easier and more efficient. A common way to conduct this step involves using tools to perform data visualizations. Approaching the data exploration in this way aids the team in previewing the data and assessing relationships between variables at a high level.

In many cases, stakeholders and subject matter experts have instincts and hunches about what the data science team should be considering and analyzing. Likely, this group had some hypothesis that led to the genesis of the project. Often, stakeholders have a good grasp of the problem and domain, although they may not be aware of the subtleties within the data or the model needed to accept or reject a hypothesis. Other times, stakeholders may be correct, but for the wrong reasons (for instance, they may be correct about a correlation that exists but infer an incorrect reason for the correlation). Meanwhile, data scientists have to approach problems with an unbiased mind-set and be ready to question all assumptions.

As the team begins to question the incoming assumptions and test initial ideas of the project sponsors and stakeholders, it needs to consider the inputs and data that will be needed, and then it must examine whether these inputs are actually correlated with the outcomes that the team plans to predict or analyze. Some methods and types of models will handle correlated variables better than others. Depending on what the team is attempting to solve, it may need to consider an alternate method, reduce the number of data inputs, or transform the inputs to allow the team to use the best method for a given business problem. Some of these techniques will be explored further in Chapter 3 and Chapter 6.

The key to this approach is to aim for capturing the most essential predictors and variables rather than considering every possible variable that people think may influence the outcome. Approaching the problem in this manner requires iterations and testing to identify the most essential variables for the intended analyses. The team should plan to test a range of variables to include in the model and then focus on the most important and influential variables.

If the team plans to run regression analyses, identify the candidate predictors and outcome variables of the model. Plan to create variables that determine outcomes but demonstrate a strong relationship to the outcome rather than to the other input variables. This includes remaining vigilant for problems such as serial correlation, multicollinearity, and other typical data modeling challenges that interfere with the validity of these models. Sometimes these issues can be avoided simply by looking at ways to reframe a given problem. In addition, sometimes determining correlation is all that is needed ("black box prediction"), and in other cases, the objective of the project is to understand the causal relationship better. In the latter case, the team wants the model to have explanatory power and needs to forecast or stress test the model under a variety of situations and with different datasets.

2.4.2 Model Selection

In the model selection subphase, the team's main goal is to choose an analytical technique, or a short list of candidate techniques, based on the end goal of the project. For the context of this book, a *model* is discussed in general terms. In this case, a model simply refers to an abstraction from reality. One observes events happening in a real-world situation or with live data and attempts to construct models that emulate this behavior with a set of rules and conditions. In the case of machine learning and data mining, these rules and conditions are grouped into several general sets of techniques, such as classification, association rules, and clustering. When reviewing this list of types of potential models, the team can winnow down the list to several viable models to try to address a given problem. More details on matching the right models to common types of business problems are provided in Chapter 3 and Chapter 4, "Advanced Analytical Theory and Methods: Clustering."

An additional consideration in this area for dealing with Big Data involves determining if the team will be using techniques that are best suited for structured data, unstructured data, or a hybrid approach. For instance, the team can leverage MapReduce to analyze unstructured data, as highlighted in Chapter 10. Lastly, the team should take care to identify and document the modeling assumptions it is making as it chooses and constructs preliminary models.

Typically, teams create the initial models using a statistical software package such as R, SAS, or Matlab. Although these tools are designed for data mining and machine learning algorithms, they may have limitations when applying the models to very large datasets, as is common with Big Data. As such, the team may consider redesigning these algorithms to run in the database itself during the pilot phase mentioned in Phase 6.

The team can move to the model building phase once it has a good idea about the type of model to try and the team has gained enough knowledge to refine the analytics plan. Advancing from this phase requires a general methodology for the analytical model, a solid understanding of the variables and techniques to use, and a description or diagram of the analytic workflow.

2.4.3 Common Tools for the Model Planning Phase

Many tools are available to assist in this phase. Here are several of the more common ones:

- **R** [14] has a complete set of modeling capabilities and provides a good environment for building interpretive models with high-quality code. In addition, it has the ability to interface with databases via an ODBC connection and execute statistical tests and analyses against Big Data via an open source connection. These two factors make R well suited to performing statistical tests and analytics on Big Data. As of this writing, R contains nearly 5,000 packages for data analysis and graphical representation. New packages are posted frequently, and many companies are providing value-add

services for R (such as training, instruction, and best practices), as well as packaging it in ways to make it easier to use and more robust. This phenomenon is similar to what happened with Linux in the late 1980s and early 1990s, when companies appeared to package and make Linux easier for companies to consume and deploy. Use R with file extracts for offline analysis and optimal performance, and use RODBC connections for dynamic queries and faster development.

- **SQL Analysis services** [15] can perform in-database analytics of common data mining functions, involved aggregations, and basic predictive models.
- **SAS/ACCESS** [16] provides integration between SAS and the analytics sandbox via multiple data connectors such as OBDC, JDBC, and OLE DB. SAS itself is generally used on file extracts, but with SAS/ACCESS, users can connect to relational databases (such as Oracle or Teradata) and data warehouse appliances (such as Greenplum or Aster), files, and enterprise applications (such as SAP and Salesforce.com).

2.5 Phase 4: Model Building

In Phase 4, the data science team needs to develop datasets for training, testing, and production purposes. These datasets enable the data scientist to develop the analytical model and train it ("training data"), while holding aside some of the data ("hold-out data" or "test data") for testing the model. (These topics are addressed in more detail in Chapter 3.) During this process, it is critical to ensure that the training and test datasets are sufficiently robust for the model and analytical techniques. A simple way to think of these datasets is to view the training dataset for conducting the initial experiments and the test sets for validating an approach once the initial experiments and models have been run.

In the model building phase, shown in Figure 2-6, an analytical model is developed and fit on the training data and evaluated (scored) against the test data. The phases of model planning and model building can overlap quite a bit, and in practice one can iterate back and forth between the two phases for a while before settling on a final model.

Although the modeling techniques and logic required to develop models can be highly complex, the actual duration of this phase can be short compared to the time spent preparing the data and defining the approaches. In general, plan to spend more time preparing and learning the data (Phases 1–2) and crafting a presentation of the findings (Phase 5). Phases 3 and 4 tend to move more quickly, although they are more complex from a conceptual standpoint.

As part of this phase, the data science team needs to execute the models defined in Phase 3.

During this phase, users run models from analytical software packages, such as R or SAS, on file extracts and small datasets for testing purposes. On a small scale, assess the validity of the model and its results. For instance, determine if the model accounts for most of the data and has robust predictive power. At this point, refine the models to optimize the results, such as by modifying variable inputs or reducing correlated variables where appropriate. In Phase 3, the team may have had some knowledge of correlated variables or problematic data attributes, which will be confirmed or denied once the models are actually executed. When immersed in the details of constructing models and transforming data, many small decisions are often made about the data and the approach for the modeling. These details can be easily forgotten once the project is completed. Therefore, it is vital to record the results and logic of the model during this phase. In addition, one must take care to record any operating assumptions that were made in the modeling process regarding the data or the context.

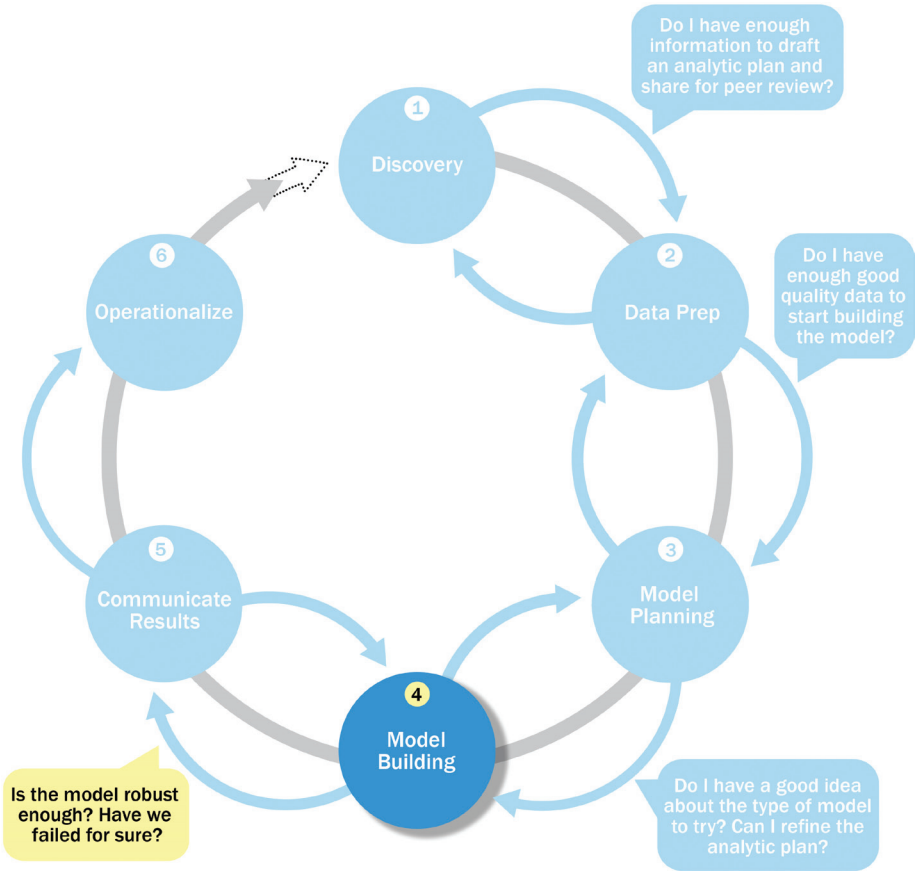


FIGURE 2-6 *Model building phase*

Creating robust models that are suitable to a specific situation requires thoughtful consideration to ensure the models being developed ultimately meet the objectives outlined in Phase 1. Questions to consider include these:

- Does the model appear valid and accurate on the test data?
- Does the model output/behavior make sense to the domain experts? That is, does it appear as if the model is giving answers that make sense in this context?
- Do the parameter values of the fitted model make sense in the context of the domain?
- Is the model sufficiently accurate to meet the goal?
- Does the model avoid intolerable mistakes? Depending on context, false positives may be more serious or less serious than false negatives, for instance. (False positives and false negatives are discussed further in Chapter 3 and Chapter 7, "Advanced Analytical Theory and Methods: Classification.")

- Are more data or more inputs needed? Do any of the inputs need to be transformed or eliminated?
- Will the kind of model chosen support the runtime requirements?
- Is a different form of the model required to address the business problem? If so, go back to the model planning phase and revise the modeling approach.

Once the data science team can evaluate either if the model is sufficiently robust to solve the problem or if the team has failed, it can move to the next phase in the Data Analytics Lifecycle.

2.5.1 Common Tools for the Model Building Phase

There are many tools available to assist in this phase, focused primarily on statistical analysis or data mining software. Common tools in this space include, but are not limited to, the following:

- Commercial Tools:
  - **SAS Enterprise Miner** [17] allows users to run predictive and descriptive models based on large volumes of data from across the enterprise. It interoperates with other large data stores, has many partnerships, and is built for enterprise-level computing and analytics.
  - **SPSS Modeler** [18] (provided by IBM and now called IBM SPSS Modeler) offers methods to explore and analyze data through a GUI.
  - **Matlab** [19] provides a high-level language for performing a variety of data analytics, algorithms, and data exploration.
  - **Alpine Miner** [11] provides a GUI front end for users to develop analytic workflows and interact with Big Data tools and platforms on the back end.
  - **STATISTICA** [20] and **Mathematica** [21] are also popular and well-regarded data mining and analytics tools.
- Free or Open Source tools:
  - **R and PL/R** [14] R was described earlier in the model planning phase, and PL/R is a procedural language for PostgreSQL with R. Using this approach means that R commands can be executed in database. This technique provides higher performance and is more scalable than running R in memory.
  - **Octave** [22], a free software programming language for computational modeling, has some of the functionality of Matlab. Because it is freely available, Octave is used in major universities when teaching machine learning.
  - **WEKA** [23] is a free data mining software package with an analytic workbench. The functions created in WEKA can be executed within Java code.
  - **Python** is a programming language that provides toolkits for machine learning and analysis, such as scikit-learn, numpy, scipy, pandas, and related data visualization using matplotlib.
  - **SQL** in-database implementations, such as **MADlib** [24], provide an alterative to in-memory desktop analytical tools. MADlib provides an open-source machine learning library of algorithms that can be executed in-database, for PostgreSQL or Greenplum.

2.6 Phase 5: Communicate Results

After executing the model, the team needs to compare the outcomes of the modeling to the criteria established for success and failure. In Phase 5, shown in Figure 2-7, the team considers how best to articulate the findings and outcomes to the various team members and stakeholders, taking into account caveats, assumptions, and any limitations of the results. Because the presentation is often circulated within an organization, it is critical to articulate the results properly and position the findings in a way that is appropriate for the audience.

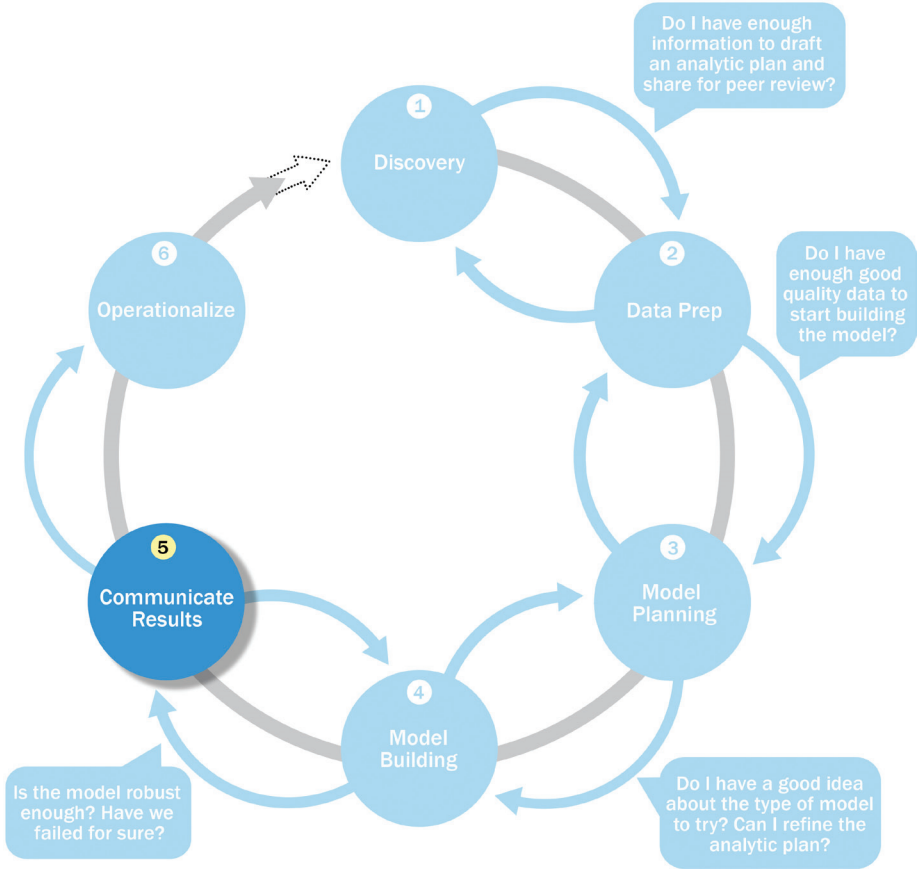


FIGURE 2-7 *Communicate results phase*

As part of Phase 5, the team needs to determine if it succeeded or failed in its objectives. Many times people do not want to admit to failing, but in this instance failure should not be considered as a true failure, but rather as a failure of the data to accept or reject a given hypothesis adequately. This concept can be counterintuitive for those who have been told their whole careers not to fail. However, the key is

to remember that the team must be rigorous enough with the data to determine whether it will prove or disprove the hypotheses outlined in Phase 1 (discovery). Sometimes teams have only done a superficial analysis, which is not robust enough to accept or reject a hypothesis. Other times, teams perform very robust analysis and are searching for ways to show results, even when results may not be there. It is important to strike a balance between these two extremes when it comes to analyzing data and being pragmatic in terms of showing real-world results.

When conducting this assessment, determine if the results are statistically significant and valid. If they are, identify the aspects of the results that stand out and may provide salient findings when it comes time to communicate them. If the results are not valid, think about adjustments that can be made to refine and iterate on the model to make it valid. During this step, assess the results and identify which data points may have been surprising and which were in line with the hypotheses that were developed in Phase 1. Comparing the actual results to the ideas formulated early on produces additional ideas and insights that would have been missed if the team had not taken time to formulate initial hypotheses early in the process.

By this time, the team should have determined which model or models address the analytical challenge in the most appropriate way. In addition, the team should have ideas of some of the findings as a result of the project. The best practice in this phase is to record all the findings and then select the three most significant ones that can be shared with the stakeholders. In addition, the team needs to reflect on the implications of these findings and measure the business value. Depending on what emerged as a result of the model, the team may need to spend time quantifying the business impact of the results to help prepare for the presentation and demonstrate the value of the findings. Doug Hubbard's work [6] offers insights on how to assess intangibles in business and quantify the value of seemingly unmeasurable things.

Now that the team has run the model, completed a thorough discovery phase, and learned a great deal about the datasets, reflect on the project and consider what obstacles were in the project and what can be improved in the future. Make recommendations for future work or improvements to existing processes, and consider what each of the team members and stakeholders needs to fulfill her responsibilities. For instance, sponsors must champion the project. Stakeholders must understand how the model affects their processes. (For example, if the team has created a model to predict customer churn, the Marketing team must understand how to use the churn model predictions in planning their interventions.) Production engineers need to operationalize the work that has been done. In addition, this is the phase to underscore the business benefits of the work and begin making the case to implement the logic into a live production environment.

As a result of this phase, the team will have documented the key findings and major insights derived from the analysis. The deliverable of this phase will be the most visible portion of the process to the outside stakeholders and sponsors, so take care to clearly articulate the results, methodology, and business value of the findings. More details will be provided about data visualization tools and references in Chapter 12, "The Endgame, or Putting It All Together."

2.7 Phase 6: Operationalize

In the final phase, the team communicates the benefits of the project more broadly and sets up a pilot project to deploy the work in a controlled way before broadening the work to a full enterprise or ecosystem of users. In Phase 4, the team scored the model in the analytics sandbox. Phase 6, shown in Figure 2-8, represents the first time that most analytics teams approach deploying the new analytical methods or models in a production environment. Rather than deploying these models immediately on a wide-scale

basis, the risk can be managed more effectively and the team can learn by undertaking a small scope, pilot deployment before a wide-scale rollout. This approach enables the team to learn about the performance and related constraints of the model in a production environment on a small scale and make adjustments before a full deployment. During the pilot project, the team may need to consider executing the algorithm in the database rather than with in-memory tools such as R because the run time is significantly faster and more efficient than running in-memory, especially on larger datasets.

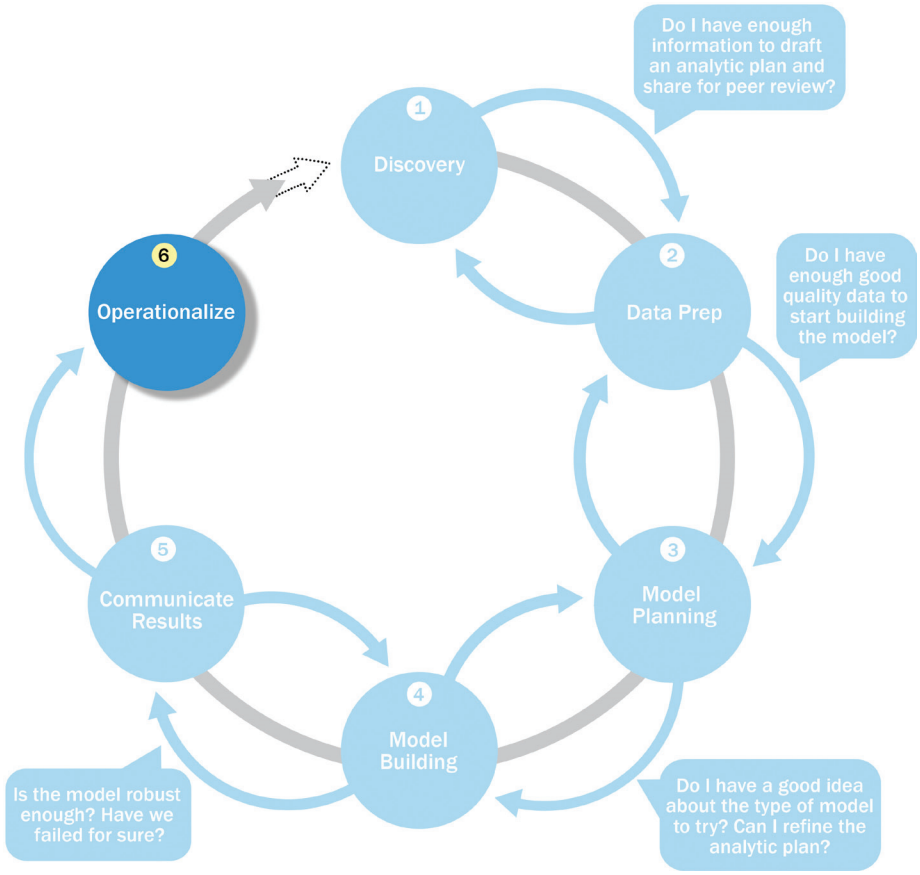


FIGURE 2-8 *Model operationalize phase*

While scoping the effort involved in conducting a pilot project, consider running the model in a production environment for a discrete set of products or a single line of business, which tests the model in a live setting. This allows the team to learn from the deployment and make any needed adjustments before launching the model across the enterprise. Be aware that this phase can bring in a new set of team members—usually the engineers responsible for the production environment who have a new set of issues and concerns beyond those of the core project team. This technical group needs to ensure that

running the model fits smoothly into the production environment and that the model can be integrated into related business processes.

Part of the operationalizing phase includes creating a mechanism for performing ongoing monitoring of model accuracy and, if accuracy degrades, finding ways to retrain the model. If feasible, design alerts for when the model is operating "out-of-bounds." This includes situations when the inputs are beyond the range that the model was trained on, which may cause the outputs of the model to be inaccurate or invalid. If this begins to happen regularly, the model needs to be retrained on new data.

Often, analytical projects yield new insights about a business, a problem, or an idea that people may have taken at face value or thought was impossible to explore. Four main deliverables can be created to meet the needs of most stakeholders. This approach for developing the four deliverables is discussed in greater detail in Chapter 12.

Figure 2-9 portrays the key outputs for each of the main stakeholders of an analytics project and what they usually expect at the conclusion of a project.

- **Business User** typically tries to determine the benefits and implications of the findings to the business.
- **Project Sponsor** typically asks questions related to the business impact of the project, the risks and return on investment (ROI), and the way the project can be evangelized within the organization (and beyond).
- **Project Manager** needs to determine if the project was completed on time and within budget and how well the goals were met.
- **Business Intelligence Analyst** needs to know if the reports and dashboards he manages will be impacted and need to change.
- **Data Engineer** and **Database Administrator** (DBA) typically need to share their code from the analytics project and create a technical document on how to implement it.
- **Data Scientist** needs to share the code and explain the model to her peers, managers, and other stakeholders.

Although these seven roles represent many interests within a project, these interests usually overlap, and most of them can be met with four main deliverables.

- Presentation for project sponsors: This contains high-level takeaways for executive level stakeholders, with a few key messages to aid their decision-making process. Focus on clean, easy visuals for the presenter to explain and for the viewer to grasp.
- Presentation for analysts, which describes business process changes and reporting changes. Fellow data scientists will want the details and are comfortable with technical graphs (such as Receiver Operating Characteristic [ROC] curves, density plots, and histograms shown in Chapter 3 and Chapter 7).
- Code for technical people.
- Technical specifications of implementing the code.

As a general rule, the more executive the audience, the more succinct the presentation needs to be. Most executive sponsors attend many briefings in the course of a day or a week. Ensure that the presentation gets to the point quickly and frames the results in terms of value to the sponsor's organization. For instance, if the team is working with a bank to analyze cases of credit card fraud, highlight the frequency of fraud, the number of cases in the past month or year, and the cost or revenue impact to the bank

(or focus on the reverse—how much more revenue the bank could gain if it addresses the fraud problem). This demonstrates the business impact better than deep dives on the methodology. The presentation needs to include supporting information about analytical methodology and data sources, but generally only as supporting detail or to ensure the audience has confidence in the approach that was taken to analyze the data.

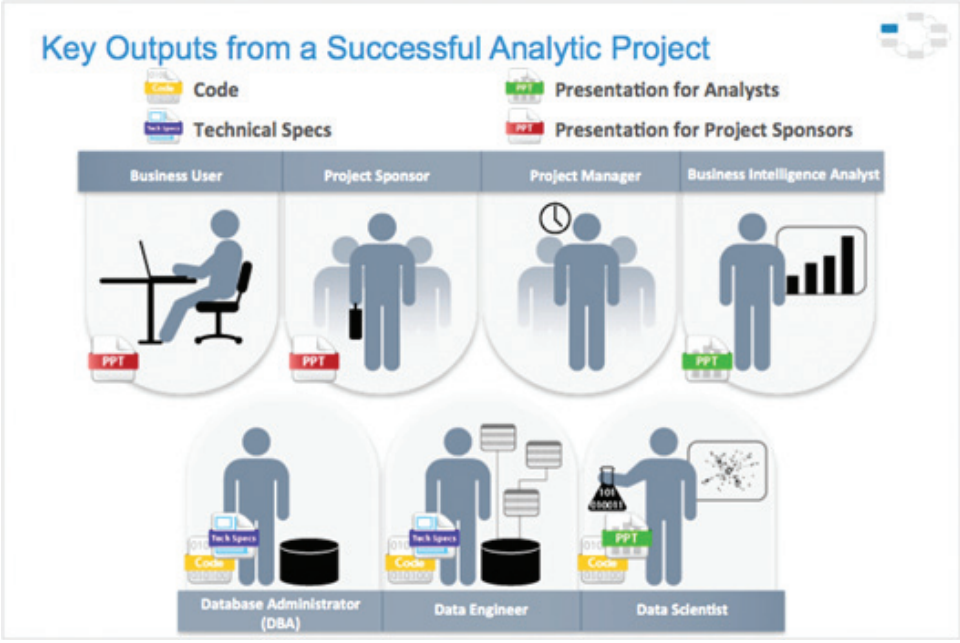


FIGURE 2-9 *Key outputs from a successful analytics project*

When presenting to other audiences with more quantitative backgrounds, focus more time on the methodology and findings. In these instances, the team can be more expansive in describing the outcomes, methodology, and analytical experiment with a peer group. This audience will be more interested in the techniques, especially if the team developed a new way of processing or analyzing data that can be reused in the future or applied to similar problems. In addition, use imagery or data visualization when possible. Although it may take more time to develop imagery, people tend to remember mental pictures to demonstrate a point more than long lists of bullets [25]. Data visualization and presentations are discussed further in Chapter 12.

2.8 Case Study: Global Innovation Network and Analysis (GINA)

EMC's Global Innovation Network and Analytics (GINA) team is a group of senior technologists located in centers of excellence (COEs) around the world. This team's charter is to engage employees across global COEs to drive innovation, research, and university partnerships. In 2012, a newly hired director wanted to

improve these activities and provide a mechanism to track and analyze the related information. In addition, this team wanted to create more robust mechanisms for capturing the results of its informal conversations with other thought leaders within EMC, in academia, or in other organizations, which could later be mined for insights.

The GINA team thought its approach would provide a means to share ideas globally and increase knowledge sharing among GINA members who may be separated geographically. It planned to create a data repository containing both structured and unstructured data to accomplish three main goals.

- Store formal and informal data.
- Track research from global technologists.
- Mine the data for patterns and insights to improve the team's operations and strategy.

The GINA case study provides an example of how a team applied the Data Analytics Lifecycle to analyze innovation data at EMC. Innovation is typically a difficult concept to measure, and this team wanted to look for ways to use advanced analytical methods to identify key innovators within the company.

2.8.1 Phase 1: Discovery

In the GINA project's discovery phase, the team began identifying data sources. Although GINA was a group of technologists skilled in many different aspects of engineering, it had some data and ideas about what it wanted to explore but lacked a formal team that could perform these analytics. After consulting with various experts including Tom Davenport, a noted expert in analytics at Babson College, and Peter Gloor, an expert in collective intelligence and creator of CoIN (Collaborative Innovation Networks) at MIT, the team decided to crowdsource the work by seeking volunteers within EMC.

Here is a list of how the various roles on the working team were fulfilled.

- **Business User, Project Sponsor, Project Manager:** Vice President from Office of the CTO
- **Business Intelligence Analyst:** Representatives from IT
- **Data Engineer** and **Database Administrator (DBA):** Representatives from IT
- **Data Scientist:** Distinguished Engineer, who also developed the social graphs shown in the GINA case study

The project sponsor's approach was to leverage social media and blogging [26] to accelerate the collection of innovation and research data worldwide and to motivate teams of "volunteer" data scientists at worldwide locations. Given that he lacked a formal team, he needed to be resourceful about finding people who were both capable and willing to volunteer their time to work on interesting problems. Data scientists tend to be passionate about data, and the project sponsor was able to tap into this passion of highly talented people to accomplish challenging work in a creative way.

The data for the project fell into two main categories. The first category represented five years of idea submissions from EMC's internal innovation contests, known as the Innovation Roadmap (formerly called the Innovation Showcase). The Innovation Roadmap is a formal, organic innovation process whereby employees from around the globe submit ideas that are then vetted and judged. The best ideas are selected for further incubation. As a result, the data is a mix of structured data, such as idea counts, submission dates, inventor names, and unstructured content, such as the textual descriptions of the ideas themselves.

The second category of data encompassed minutes and notes representing innovation and research activity from around the world. This also represented a mix of structured and unstructured data. The structured data included attributes such as dates, names, and geographic locations. The unstructured documents contained the "who, what, when, and where" information that represents rich data about knowledge growth and transfer within the company. This type of information is often stored in business silos that have little to no visibility across disparate research teams.

The 10 main IHs that the GINA team developed were as follows:

- **IH1:** Innovation activity in different geographic regions can be mapped to corporate strategic directions.
- **IH2:** The length of time it takes to deliver ideas decreases when global knowledge transfer occurs as part of the idea delivery process.
- **IH3:** Innovators who participate in global knowledge transfer deliver ideas more quickly than those who do not.
- **IH4:** An idea submission can be analyzed and evaluated for the likelihood of receiving funding.
- **IH5:** Knowledge discovery and growth for a particular topic can be measured and compared across geographic regions.
- **IH6:** Knowledge transfer activity can identify research-specific boundary spanners in disparate regions.
- **IH7:** Strategic corporate themes can be mapped to geographic regions.
- **IH8:** Frequent knowledge expansion and transfer events reduce the time it takes to generate a corporate asset from an idea.
- **IH9:** Lineage maps can reveal when knowledge expansion and transfer did not (or has not) resulted in a corporate asset.
- **IH10:** Emerging research topics can be classified and mapped to specific ideators, innovators, boundary spanners, and assets.

The GINA (IHs) can be grouped into two categories:

- Descriptive analytics of what is currently happening to spark further creativity, collaboration, and asset generation
- Predictive analytics to advise executive management of where it should be investing in the future

2.8.2 Phase 2: Data Preparation

The team partnered with its IT department to set up a new analytics sandbox to store and experiment on the data. During the data exploration exercise, the data scientists and data engineers began to notice that certain data needed conditioning and normalization. In addition, the team realized that several missing datasets were critical to testing some of the analytic hypotheses.

As the team explored the data, it quickly realized that if it did not have data of sufficient quality or could not get good quality data, it would not be able to perform the subsequent steps in the lifecycle process. As a result, it was important to determine what level of data quality and cleanliness was sufficient for the

project being undertaken. In the case of the GINA, the team discovered that many of the names of the researchers and people interacting with the universities were misspelled or had leading and trailing spaces in the datastore. Seemingly small problems such as these in the data had to be addressed in this phase to enable better analysis and data aggregation in subsequent phases.

2.8.3 Phase 3: Model Planning

In the GINA project, for much of the dataset, it seemed feasible to use social network analysis techniques to look at the networks of innovators within EMC. In other cases, it was difficult to come up with appropriate ways to test hypotheses due to the lack of data. In one case (IH9), the team made a decision to initiate a longitudinal study to begin tracking data points over time regarding people developing new intellectual property. This data collection would enable the team to test the following two ideas in the future:

- **IH8:** Frequent knowledge expansion and transfer events reduce the amount of time it takes to generate a corporate asset from an idea.
- **IH9:** Lineage maps can reveal when knowledge expansion and transfer did not (or has not) result(ed) in a corporate asset.

For the longitudinal study being proposed, the team needed to establish goal criteria for the study. Specifically, it needed to determine the end goal of a successful idea that had traversed the entire journey. The parameters related to the scope of the study included the following considerations:

- Identify the right milestones to achieve this goal.
- Trace how people move ideas from each milestone toward the goal.
- Once this is done, trace ideas that die, and trace others that reach the goal. Compare the journeys of ideas that make it and those that do not.
- Compare the times and the outcomes using a few different methods (depending on how the data is collected and assembled). These could be as simple as t-tests or perhaps involve different types of classification algorithms.

2.8.4 Phase 4: Model Building

In Phase 4, the GINA team employed several analytical methods. This included work by the data scientist using Natural Language Processing (NLP) techniques on the textual descriptions of the Innovation Roadmap ideas. In addition, he conducted social network analysis using R and RStudio, and then he developed social graphs and visualizations of the network of communications related to innovation using R's `ggplot2` package. Examples of this work are shown in Figures 2-10 and 2-11.

FIGURE 2-10 *Social graph [27] visualization of idea submitters and finalists*

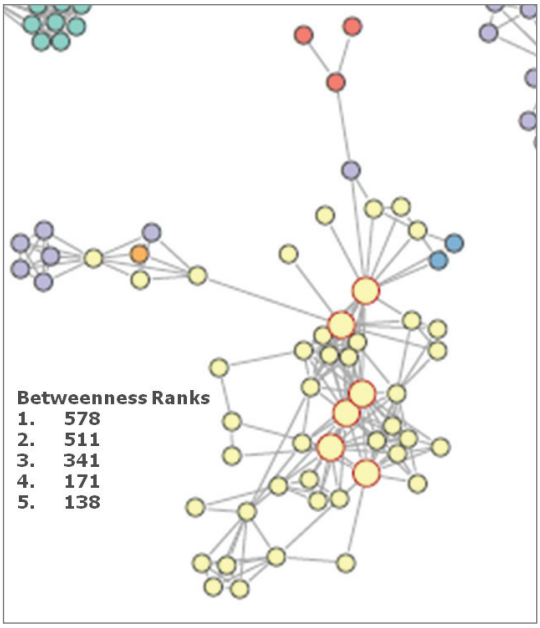


FIGURE 2-11 *Social graph visualization of top innovation influencers*

Figure 2-10 shows social graphs that portray the relationships between idea submitters within GINA. Each color represents an innovator from a different country. The large dots with red circles around them represent hubs. A ***hub*** represents a person with high connectivity and a high "betweenness" score. The cluster in Figure 2-11 contains geographic variety, which is critical to prove the hypothesis about geographic boundary spanners. One person in this graph has an unusually high score when compared to the rest of the nodes in the graph. The data scientist identified this person and ran a query against his name within the analytic sandbox. These actions yielded the following information about this research scientist (from the social graph), which illustrated how influential he was within his business unit and across many other areas of the company worldwide:

- In 2011, he attended the ACM SIGMOD conference, which is a top-tier conference on large-scale data management problems and databases.
- He visited employees in France who are part of the business unit for EMC's content management teams within Documentum (now part of the Information Intelligence Group, or IIG).
- He presented his thoughts on the SIGMOD conference at a virtual brownbag session attended by three employees in Russia, one employee in Cairo, one employee in Ireland, one employee in India, three employees in the United States, and one employee in Israel.
- In 2012, he attended the SDM 2012 conference in California.
- On the same trip he visited innovators and researchers at EMC federated companies, Pivotal and VMware.
- Later on that trip he stood before an internal council of technology leaders and introduced two of his researchers to dozens of corporate innovators and researchers.

This finding suggests that at least part of the initial hypothesis is correct; the data can identify innovators who span different geographies and business units. The team used Tableau software for data visualization and exploration and used the Pivotal Greenplum database as the main data repository and analytics engine.

2.8.5 Phase 5: Communicate Results

In Phase 5, the team found several ways to cull results of the analysis and identify the most impactful and relevant findings. This project was considered successful in identifying boundary spanners and hidden innovators. As a result, the CTO office launched longitudinal studies to begin data collection efforts and track innovation results over longer periods of time. The GINA project promoted knowledge sharing related to innovation and researchers spanning multiple areas within the company and outside of it. GINA also enabled EMC to cultivate additional intellectual property that led to additional research topics and provided opportunities to forge relationships with universities for joint academic research in the fields of Data Science and Big Data. In addition, the project was accomplished with a limited budget, leveraging a volunteer force of highly skilled and distinguished engineers and data scientists.

One of the key findings from the project is that there was a disproportionately high density of innovators in Cork, Ireland. Each year, EMC hosts an innovation contest, open to employees to submit innovation ideas that would drive new value for the company. When looking at the data in 2011, 15% of the finalists and 15% of the winners were from Ireland. These are unusually high numbers, given the relative size of the Cork COE compared to other larger centers in other parts of the world. After further research, it was learned that the COE in Cork, Ireland had received focused training in innovation from an external consultant, which

was proving effective. The Cork COE came up with more innovation ideas, and better ones, than it had in the past, and it was making larger contributions to innovation at EMC. It would have been difficult, if not impossible, to identify this cluster of innovators through traditional methods or even anecdotal, word-of-mouth feedback. Applying social network analysis enabled the team to find a pocket of people within EMC who were making disproportionately strong contributions. These findings were shared internally through presentations and conferences and promoted through social media and blogs.

2.8.6 Phase 6: Operationalize

Running analytics against a sandbox filled with notes, minutes, and presentations from innovation activities yielded great insights into EMC's innovation culture. Key findings from the project include these:

- The CTO office and GINA need more data in the future, including a marketing initiative to convince people to inform the global community on their innovation/research activities.
- Some of the data is sensitive, and the team needs to consider security and privacy related to the data, such as who can run the models and see the results.
- In addition to running models, a parallel initiative needs to be created to improve basic Business Intelligence activities, such as dashboards, reporting, and queries on research activities worldwide.
- A mechanism is needed to continually reevaluate the model after deployment. Assessing the benefits is one of the main goals of this stage, as is defining a process to retrain the model as needed.

In addition to the actions and findings listed, the team demonstrated how analytics can drive new insights in projects that are traditionally difficult to measure and quantify. This project informed investment decisions in university research projects by the CTO office and identified hidden, high-value innovators. In addition, the CTO office developed tools to help submitters improve ideas using topic modeling as part of new recommender systems to help idea submitters find similar ideas and refine their proposals for new intellectual property.

Table 2-3 outlines an analytics plan for the GINA case study example. Although this project shows only three findings, there were many more. For instance, perhaps the biggest overarching result from this project is that it demonstrated, in a concrete way, that analytics can drive new insights in projects that deal with topics that may seem difficult to measure, such as innovation.

Table 2-3 *Analytic Plan from the EMC GINA Project*

Components of Analytic Plan	GINA Case Study
Discovery Business Problem Framed	Tracking global knowledge growth, ensuring effective knowledge transfer, and quickly converting it into corporate assets. Executing on these three elements should accelerate innovation.
Initial Hypotheses	An increase in geographic knowledge transfer improves the speed of idea delivery.
Data	Five years of innovation idea submissions and history; six months of textual notes from global innovation and research activities

(*continues*)

Table 2-3 *Analytic Plan from the EMC GINA Project* (*Continued*)

Components of Analytic Plan	GINA Case Study
Model Planning Analytic Technique	Social network analysis, social graphs, clustering, and regression analysis
Result and Key Findings	1. Identified hidden, high-value innovators and found ways to share their knowledge 2. Informed investment decisions in university research projects 3. Created tools to help submitters improve ideas with idea recommender systems

Innovation is an idea that every company wants to promote, but it can be difficult to measure innovation or identify ways to increase innovation. This project explored this issue from the standpoint of evaluating informal social networks to identify boundary spanners and influential people within innovation sub-networks. In essence, this project took a seemingly nebulous problem and applied advanced analytical methods to tease out answers using an objective, fact-based approach.

Another outcome from the project included the need to supplement analytics with a separate data-store for Business Intelligence reporting, accessible to search innovation/research initiatives. Aside from supporting decision making, this will provide a mechanism to be informed on discussions and research happening worldwide among team members in disparate locations. Finally, it highlighted the value that can be gleaned through data and subsequent analysis. Therefore, the need was identified to start formal marketing programs to convince people to submit (or inform) the global community on their innovation/research activities. The knowledge sharing was critical. Without it, GINA would not have been able to perform the analysis and identify the hidden innovators within the company.

Summary

This chapter described the Data Analytics Lifecycle, which is an approach to managing and executing analytical projects. This approach describes the process in six phases.

1. Discovery
2. Data preparation
3. Model planning
4. Model building
5. Communicate results
6. Operationalize

Through these steps, data science teams can identify problems and perform rigorous investigation of the datasets needed for in-depth analysis. As stated in the chapter, although much is written about the analytical methods, the bulk of the time spent on these kinds of projects is spent in preparation—namely,

in Phases 1 and 2 (discovery and data preparation). In addition, this chapter discussed the seven roles needed for a data science team. It is critical that organizations recognize that Data Science is a team effort, and a balance of skills is needed to be successful in tackling Big Data projects and other complex projects involving data analytics.

Exercises

1. In which phase would the team expect to invest most of the project time? Why? Where would the team expect to spend the least time?
2. What are the benefits of doing a pilot program before a full-scale rollout of a new analytical methodology? Discuss this in the context of the mini case study.
3. What kinds of tools would be used in the following phases, and for which kinds of use scenarios?
   a. Phase 2: Data preparation
   b. Phase 4: Model building

Bibliography

[1] T. H. Davenport and D. J. Patil, "Data Scientist: The Sexiest Job of the 21st Century," *Harvard Business Review,* October 2012.
[2] J. Manyika, M. Chiu, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big Data: The Next Frontier for Innovation, Competition, and Productivity," McKinsey Global Institute, 2011.
[3] "Scientific Method" [Online]. Available: `http://en.wikipedia.org/wiki/Scientific_method`.
[4] "CRISP-DM" [Online]. Available: `http://en.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining`.
[5] T. H. Davenport, J. G. Harris, and R. Morison, *Analytics at Work: Smarter Decisions, Better Results*, 2010, Harvard Business Review Press.
[6] D. W. Hubbard, *How to Measure Anything: Finding the Value of Intangibles in Business*, 2010, Hoboken, NJ: John Wiley & Sons.
[7] J. Cohen, B. Dolan, M. Dunlap, J. M. Hellerstein and C. Welton, *MAD Skills: New Analysis Practices for Big Data*, Watertown, MA 2009.
[8] "List of APIs" [Online]. Available: `http://www.programmableweb.com/apis`.
[9] B. Shneiderman [Online]. Available: `http://www.ifp.illinois.edu/nabhcs/abstracts/shneiderman.html`.
[10] "Hadoop" [Online]. Available: `http://hadoop.apache.org`.
[11] "Alpine Miner" [Online]. Available: `http://alpinenow.com`.
[12] "OpenRefine" [Online]. Available: `http://openrefine.org`.
[13] "Data Wrangler" [Online]. Available: `http://vis.stanford.edu/wrangler/`.
[14] "CRAN" [Online]. Available: `http://cran.us.r-project.org`.
[15] "SQL" [Online]. Available: `http://en.wikipedia.org/wiki/SQL`.
[16] "SAS/ACCESS" [Online]. Available: `http://www.sas.com/en_us/software/data-management/access.htm`.

[17] "SAS Enterprise Miner" [Online]. Available: http://www.sas.com/en_us/software/analytics/enterprise-miner.html.

[18] "SPSS Modeler" [Online]. Available: http://www-03.ibm.com/software/products/en/category/business-analytics.

[19] "Matlab" [Online]. Available: http://www.mathworks.com/products/matlab/.

[20] "Statistica" [Online]. Available: https://www.statsoft.com.

[21] "Mathematica" [Online]. Available: http://www.wolfram.com/mathematica/.

[22] "Octave" [Online]. Available: https://www.gnu.org/software/octave/.

[23] "WEKA" [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/.

[24] "MADlib" [Online]. Available: http://madlib.net.

[25] K. L. Higbee, *Your Memory—How It Works and How to Improve It*, New York: Marlowe & Company, 1996.

[26] S. Todd, "Data Science and Big Data Curriculum" [Online]. Available: http://stevetodd.typepad.com/my_weblog/data-science-and-big-data-curriculum/.

[27] T. H Davenport and D. J. Patil, "Data Scientist: The Sexiest Job of the 21st Century," *Harvard Business Review*, October 2012.

3

Review of Basic Data Analytic Methods Using R

Key Concepts

Basic features of R
Data exploration and analysis with R
Statistical methods for evaluation

The previous chapter presented the six phases of the Data Analytics Lifecycle.

- Phase 1: Discovery
- Phase 2: Data Preparation
- Phase 3: Model Planning
- Phase 4: Model Building
- Phase 5: Communicate Results
- Phase 6: Operationalize

The first three phases involve various aspects of data exploration. In general, the success of a data analysis project requires a deep understanding of the data. It also requires a toolbox for mining and presenting the data. These activities include the study of the data in terms of basic statistical measures and creation of graphs and plots to visualize and identify relationships and patterns. Several free or commercial tools are available for exploring, conditioning, modeling, and presenting data. Because of its popularity and versatility, the open-source programming language R is used to illustrate many of the presented analytical tasks and models in this book.

This chapter introduces the basic functionality of the R programming language and environment. The first section gives an overview of how to use R to acquire, parse, and filter the data as well as how to obtain some basic descriptive statistics on a dataset. The second section examines using R to perform exploratory data analysis tasks using visualization. The final section focuses on statistical inference, such as hypothesis testing and analysis of variance in R.

3.1 Introduction to R

R is a programming language and software framework for statistical analysis and graphics. Available for use under the GNU General Public License [1], R software and installation instructions can be obtained via the Comprehensive R Archive and Network [2]. This section provides an overview of the basic functionality of R. In later chapters, this foundation in R is utilized to demonstrate many of the presented analytical techniques.

Before delving into specific operations and functions of R later in this chapter, it is important to understand the flow of a basic R script to address an analytical problem. The following R code illustrates a typical analytical situation in which a dataset is imported, the contents of the dataset are examined, and some modeling building tasks are executed. Although the reader may not yet be familiar with the R syntax, the code can be followed by reading the embedded comments, denoted by #. In the following scenario, the annual sales in U.S. dollars for 10,000 retail customers have been provided in the form of a comma-separated-value (CSV) file. The `read.csv()` function is used to import the CSV file. This dataset is stored to the R variable *`sales`* using the assignment operator `<-`.

```
# import a CSV file of the total annual sales for each customer
sales <- read.csv("c:/data/yearly_sales.csv")

# examine the imported dataset
head(sales)
```

```
summary(sales)

# plot num_of_orders vs. sales
plot(sales$num_of_orders,sales$sales_total,
        main="Number of Orders vs. Sales")

# perform a statistical analysis (fit a linear regression model)
results <- lm(sales$sales_total ~ sales$num_of_orders)
summary(results)

# perform some diagnostics on the fitted model
# plot histogram of the residuals
hist(results$residuals, breaks = 800)
```

In this example, the data file is imported using the `read.csv()` function. Once the file has been imported, it is useful to examine the contents to ensure that the data was loaded properly as well as to become familiar with the data. In the example, the `head()` function, by default, displays the first six records of *`sales`*.

```
# examine the imported dataset
head(sales)
  cust_id sales_total num_of_orders gender
1  100001      800.64             3      F
2  100002      217.53             3      F
3  100003       74.58             2      M
4  100004      498.60             3      M
5  100005      723.11             4      F
6  100006       69.43             2      F
```

The `summary()` function provides some descriptive statistics, such as the mean and median, for each data column. Additionally, the minimum and maximum values as well as the 1st and 3rd quartiles are provided. Because the *`gender`* column contains two possible characters, an "F" (female) or "M" (male), the `summary()` function provides the count of each character's occurrence.

```
summary(sales)

    cust_id        sales_total      num_of_orders    gender
 Min.   :100001   Min.   :  30.02   Min.   : 1.000   F:5035
 1st Qu.:102501   1st Qu.:  80.29   1st Qu.: 2.000   M:4965
 Median :105001   Median : 151.65   Median : 2.000
 Mean   :105001   Mean   : 249.46   Mean   : 2.428
 3rd Qu.:107500   3rd Qu.: 295.50   3rd Qu.: 3.000
 Max.   :110000   Max.   :7606.09   Max.   :22.000
```

Plotting a dataset's contents can provide information about the relationships between the various columns. In this example, the `plot()` function generates a scatterplot of the number of orders (`sales$num_of_orders`) against the annual sales (`sales$sales_total`). The `$` is used to reference a specific column in the dataset `sales`. The resulting plot is shown in Figure 3-1.

```
# plot num_of_orders vs. sales
plot(sales$num_of_orders,sales$sales_total,
        main="Number of Orders vs. Sales")
```

FIGURE 3-1 *Graphically examining the data*

Each point corresponds to the number of orders and the total sales for each customer. The plot indicates that the annual sales are proportional to the number of orders placed. Although the observed relationship between these two variables is not purely linear, the analyst decided to apply linear regression using the `lm()` function as a first step in the modeling process.

```
results <- lm(sales$sales_total ~ sales$num_of_orders)
results

Call:
lm(formula = sales$sales_total ~ sales$num_of_orders)

Coefficients:
        (Intercept)  sales$num_of_orders
             -154.1                166.2
```

The resulting intercept and slope values are –154.1 and 166.2, respectively, for the fitted linear equation. However, *`results`* stores considerably more information that can be examined with the `summary()` function. Details on the contents of *`results`* are examined by applying the `attributes()` function. Because regression analysis is presented in more detail later in the book, the reader should not overly focus on interpreting the following output.

```
summary(results)

Call:
lm(formula = sales$sales_total ~ sales$num_of_orders)

Residuals:
   Min     1Q Median     3Q    Max
-666.5 -125.5  -26.7   86.6 4103.4

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         -154.128      4.129  -37.33   <2e-16 ***
sales$num_of_orders  166.221      1.462  113.66   <2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 210.8 on 9998 degrees of freedom
Multiple R-squared:  0.5637,      Adjusted R-squared:  0.5637
F-statistic: 1.292e+04 on 1 and 9998 DF,  p-value: < 2.2e-16
```

The `summary()` function is an example of a generic function. A ***generic function*** is a group of functions sharing the same name but behaving differently depending on the number and the type of arguments they receive. Utilized previously, `plot()` is another example of a generic function; the plot is determined by the passed variables. Generic functions are used throughout this chapter and the book. In the final portion of the example, the following R code uses the generic function `hist()` to generate a histogram (Figure 3-2) of the residuals stored in *`results`*. The function call illustrates that optional parameter values can be passed. In this case, the number of *`breaks`* is specified to observe the large residuals.

```
# perform some diagnostics on the fitted model
# plot histogram of the residuals
hist(results$residuals, breaks = 800)
```

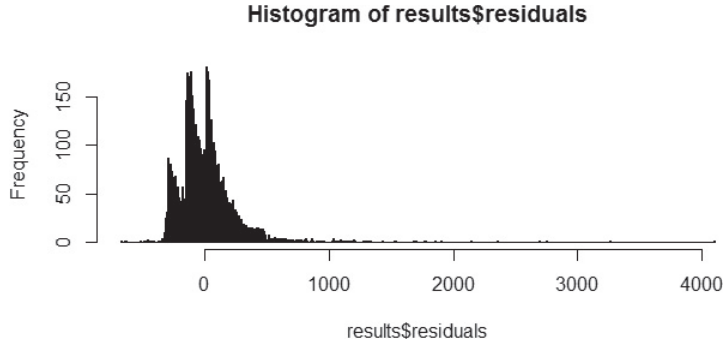


FIGURE 3-2 *Evidence of large residuals*

This simple example illustrates a few of the basic model planning and building tasks that may occur in Phases 3 and 4 of the Data Analytics Lifecycle. Throughout this chapter, it is useful to envision how the presented R functionality will be used in a more comprehensive analysis.

3.1.1 R Graphical User Interfaces

R software uses a command-line interface (CLI) that is similar to the BASH shell in Linux or the interactive versions of scripting languages such as Python. UNIX and Linux users can enter command `R` at the terminal prompt to use the CLI. For Windows installations, R comes with RGui.exe, which provides a basic graphical user interface (GUI). However, to improve the ease of writing, executing, and debugging R code, several additional GUIs have been written for R. Popular GUIs include the R commander [3], Rattle [4], and RStudio [5]. This section presents a brief overview of RStudio, which was used to build the R examples in this book. Figure 3-3 provides a screenshot of the previous R code example executed in RStudio.

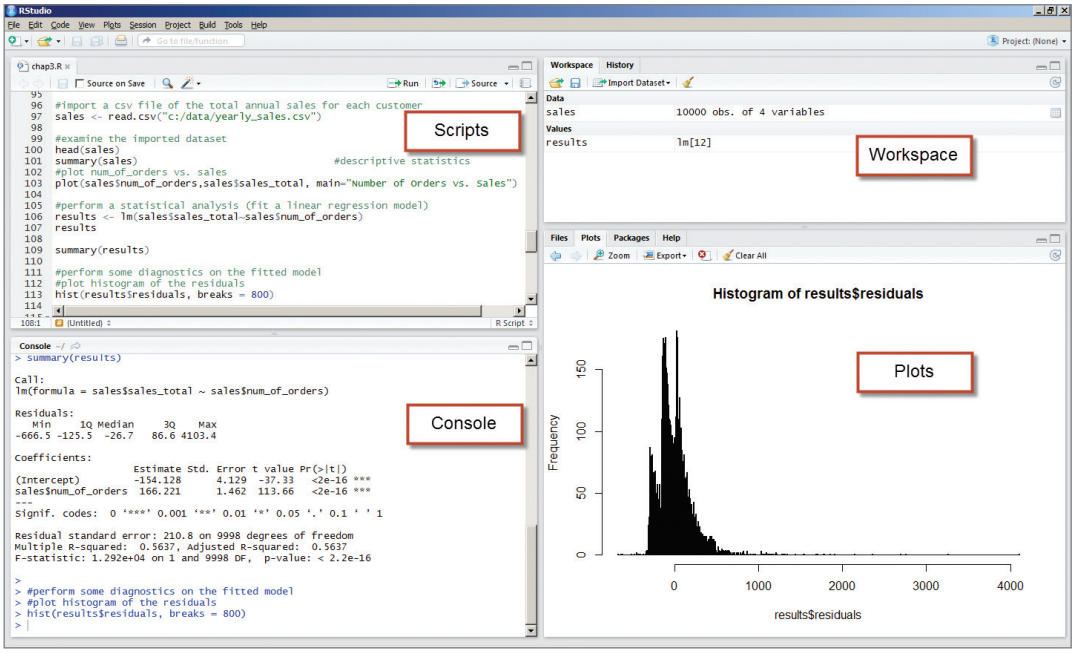


FIGURE 3-3 *RStudio GUI*

The four highlighted window panes follow.

- **Scripts:** Serves as an area to write and save R code
- **Workspace:** Lists the datasets and variables in the R environment
- **Plots:** Displays the plots generated by the R code and provides a straightforward mechanism to export the plots
- **Console:** Provides a history of the executed R code and the output

Additionally, the console pane can be used to obtain help information on R. Figure 3-4 illustrates that by entering `?lm` at the console prompt, the help details of the `lm()` function are provided on the right. Alternatively, `help(lm)` could have been entered at the console prompt.

Functions such as `edit()` and `fix()` allow the user to update the contents of an R variable. Alternatively, such changes can be implemented with RStudio by selecting the appropriate variable from the workspace pane.

R allows one to save the workspace environment, including variables and loaded libraries, into an `.Rdata` file using the `save.image()` function. An existing `.Rdata` file can be loaded using the `load.image()` function. Tools such as RStudio prompt the user for whether the developer wants to save the workspace connects prior to exiting the GUI.

The reader is encouraged to install R and a preferred GUI to try out the R examples provided in the book and utilize the help functionality to access more details about the discussed topics.

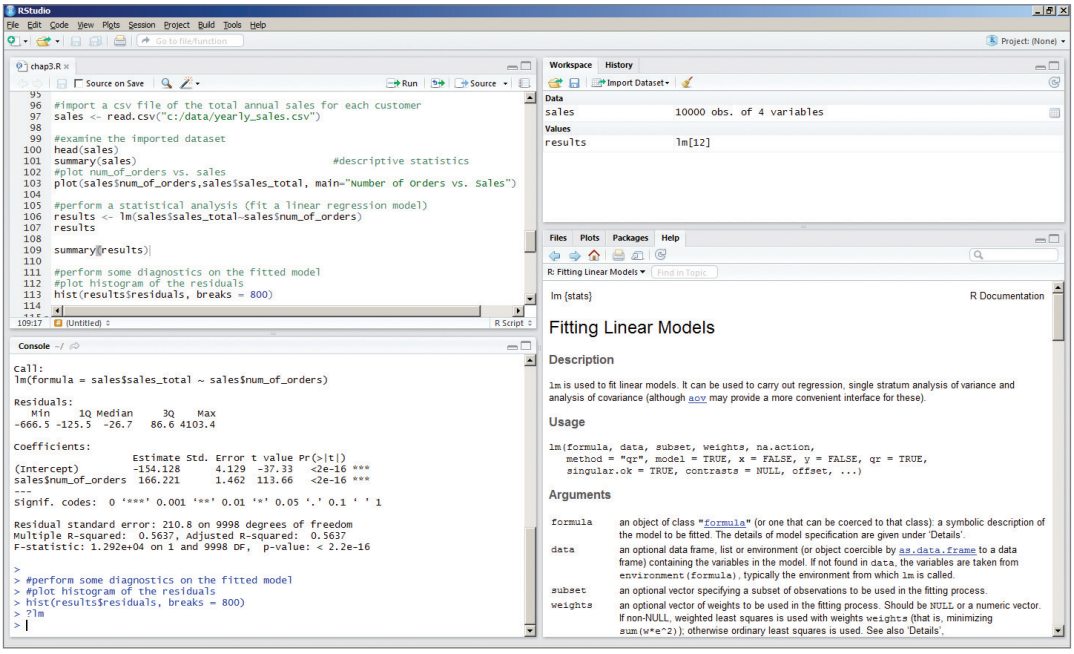


FIGURE 3-4 *Accessing help in Rstudio*

3.1.2 Data Import and Export

In the annual retail sales example, the dataset was imported into R using the `read.csv()` function as in the following code.

```
sales <- read.csv("c:/data/yearly_sales.csv")
```

R uses a forward slash (/) as the separator character in the directory and file paths. This convention makes script files somewhat more portable at the expense of some initial confusion on the part of Windows users, who may be accustomed to using a backslash (\) as a separator. To simplify the import of multiple files with long path names, the `setwd()` function can be used to set the working directory for the subsequent import and export operations, as shown in the following R code.

```
setwd("c:/data/")
sales <- read.csv("yearly_sales.csv")
```

Other import functions include `read.table()` and `read.delim()`, which are intended to import other common file types such as TXT. These functions can also be used to import the `yearly_sales.csv` file, as the following code illustrates.

```
sales_table <- read.table("yearly_sales.csv", header=TRUE, sep=",")
sales_delim <- read.delim("yearly_sales.csv", sep=",")
```

The main difference between these import functions is the default values. For example, the `read.delim()` function expects the column separator to be a tab ("`\t`"). In the event that the numerical data

in a data file uses a comma for the decimal, R also provides two additional functions—`read.csv2()` and `read.delim2()`—to import such data. Table 3-1 includes the expected defaults for headers, column separators, and decimal point notations.

TABLE 3-1 *Import Function Defaults*

Function	Headers	Separator	Decimal Point
`read.table()`	FALSE	" "	"."
`read.csv()`	TRUE	","	"."
`read.csv2()`	TRUE	";"	","
`read.delim()`	TRUE	"\t"	"."
`read.delim2()`	TRUE	"\t"	","

The analogous R functions such as `write.table()`, `write.csv()`, and `write.csv2()` enable exporting of R datasets to an external file. For example, the following R code adds an additional column to the sales dataset and exports the modified dataset to an external file.

```
# add a column for the average sales per order
sales$per_order <- sales$sales_total/sales$num_of_orders

# export data as tab delimited without the row names
write.table(sales,"sales_modified.txt", sep="\t", row.names=FALSE
```

Sometimes it is necessary to read data from a database management system (DBMS). R packages such as `DBI` [6] and `RODBC` [7] are available for this purpose. These packages provide database interfaces for communication between R and DBMSs such as MySQL, Oracle, SQL Server, PostgreSQL, and Pivotal Greenplum. The following R code demonstrates how to install the `RODBC` package with the `install.packages()` function. The `library()` function loads the package into the R workspace. Finally, a connector (*conn*) is initialized for connecting to a Pivotal Greenplum database *training2* via open database connectivity (ODBC) with user `user`. The *training2* database must be defined either in the `/etc/ODBC.ini` configuration file or using the Administrative Tools under the Windows Control Panel.

```
install.packages("RODBC")
library(RODBC)
conn <- odbcConnect("training2", uid="user", pwd="password")
```

The connector needs to be present to submit a SQL query to an ODBC database by using the `sqlQuery()` function from the `RODBC` package. The following R code retrieves specific columns from the *housing* table in which household income (`hinc`) is greater than $1,000,000.

```
housing_data <- sqlQuery(conn, "select serialno, state, persons, rooms
                                from housing
                                where hinc > 1000000")
head(housing_data)
  serialno state persons rooms
1  3417867     6       2     7
2  3417867     6       2     7
```

```
3  4552088     6        5     9
4  4552088     6        5     9
5  8699293     6        5     5
6  8699293     6        5     5
```

Although plots can be saved using the RStudio GUI, plots can also be saved using R code by specifying the appropriate graphic devices. Using the `jpeg()` function, the following R code creates a new JPEG file, adds a histogram plot to the file, and then closes the file. Such techniques are useful when automating standard reports. Other functions, such as `png()`, `bmp()`, `pdf()`, and `postscript()`, are available in R to save plots in the desired format.

```
jpeg(file="c:/data/sales_hist.jpeg") # create a new jpeg file
hist(sales$num_of_orders)            # export histogram to jpeg
dev.off()                            # shut off the graphic device
```

More information on data imports and exports can be found at `http://cran.r-project.org/doc/manuals/r-release/R-data.html`, such as how to import datasets from statistical software packages including Minitab, SAS, and SPSS.

3.1.3 Attribute and Data Types

In the earlier example, the *`sales`* variable contained a record for each customer. Several characteristics, such as total annual sales, number of orders, and gender, were provided for each customer. In general, these characteristics or attributes provide the qualitative and quantitative measures for each item or subject of interest. Attributes can be categorized into four types: nominal, ordinal, interval, and ratio (NOIR) [8]. Table 3-2 distinguishes these four attribute types and shows the operations they support. Nominal and ordinal attributes are considered categorical attributes, whereas interval and ratio attributes are considered numeric attributes.

TABLE 3-2 *NOIR Attribute Types*

	Categorical (Qualitative)		Numeric (Quantitative)	
	Nominal	**Ordinal**	**Interval**	**Ratio**
Definition	The values represent labels that distinguish one from another.	Attributes imply a sequence.	The difference between two values is meaningful.	Both the difference and the ratio of two values are meaningful.
Examples	ZIP codes, nationality, street names, gender, employee ID numbers, `TRUE` or `FALSE`	Quality of diamonds, academic grades, magnitude of earthquakes	Temperature in Celsius or Fahrenheit, calendar dates, latitudes	Age, temperature in Kelvin, counts, length, weight
Operations	$=, \neq$	$=, \neq,$ $<, \leq, >, \geq$	$=, \neq,$ $<, \leq, >, \geq,$ $+, -$	$=, \neq,$ $<, \leq, >, \geq,$ $+, -,$ $\times, \div$

Data of one attribute type may be converted to another. For example, the *quality* of diamonds {Fair, Good, Very Good, Premium, Ideal} is considered ordinal but can be converted to nominal {Good, Excellent} with a defined mapping. Similarly, a ratio attribute like *Age* can be converted into an ordinal attribute such as {Infant, Adolescent, Adult, Senior}. Understanding the attribute types in a given dataset is important to ensure that the appropriate descriptive statistics and analytic methods are applied and properly interpreted. For example, the mean and standard deviation of U.S. postal ZIP codes are not very meaningful or appropriate. Proper handling of categorical variables will be addressed in subsequent chapters. Also, it is useful to consider these attribute types during the following discussion on R data types.

Numeric, Character, and Logical Data Types

Like other programming languages, R supports the use of numeric, character, and logical (Boolean) values. Examples of such variables are given in the following R code.

```
i <- 1                          # create a numeric variable
sport <- "football"             # create a character variable
flag <- TRUE                    # create a logical variable
```

R provides several functions, such as `class()` and `typeof()`, to examine the characteristics of a given variable. The `class()` function represents the abstract class of an object. The `typeof()` function determines the way an object is stored in memory. Although *i* appears to be an integer, *i* is internally stored using double precision. To improve the readability of the code segments in this section, the inline R comments are used to explain the code or to provide the returned values.

```
class(i)                        # returns "numeric"
typeof(i)                       # returns "double"

class(sport)                    # returns "character"
typeof(sport)                   # returns "character"

class(flag)                     # returns "logical"
typeof(flag)                    # returns "logical"
```

Additional R functions exist that can test the variables and coerce a variable into a specific type. The following R code illustrates how to test if *i* is an integer using the `is.integer()` function and to coerce *i* into a new integer variable, *j*, using the `as.integer()` function. Similar functions can be applied for double, character, and logical types.

```
is.integer(i)                   # returns FALSE
j <- as.integer(i)              # coerces contents of i into an integer
is.integer(j)                   # returns TRUE
```

The application of the `length()` function reveals that the created variables each have a length of 1. One might have expected the returned length of *sport* to have been 8 for each of the characters in the string `"football"`. However, these three variables are actually one element, ***vectors***.

```
length(i)                       # returns 1
length(flag)                    # returns 1
length(sport)                   # returns 1 (not 8 for "football")
```

Vectors

Vectors are a basic building block for data in R. As seen previously, simple R variables are actually vectors. A vector can only consist of values in the same class. The tests for vectors can be conducted using the `is.vector()` function.

```
is.vector(i)                    # returns TRUE
is.vector(flag)                 # returns TRUE
is.vector(sport)                # returns TRUE
```

R provides functionality that enables the easy creation and manipulation of vectors. The following R code illustrates how a vector can be created using the combine function, `c()` or the colon operator, `:`, to build a vector from the sequence of integers from 1 to 5. Furthermore, the code shows how the values of an existing vector can be easily modified or accessed. The code, related to the *z* vector, indicates how logical comparisons can be built to extract certain elements of a given vector.

```
u <- c("red", "yellow", "blue") # create a vector "red" "yellow" "blue"
u                               # returns "red" "yellow" "blue"
u[1]                            # returns "red" (1st element in u)
v <- 1:5                        # create a vector 1 2 3 4 5
v                               # returns 1 2 3 4 5
sum(v)                          # returns 15
w <- v * 2                      # create a vector 2 4 6 8 10
w                               # returns 2 4 6 8 10
w[3]                            # returns 6 (the 3rd element of w)
z <- v + w                      # sums two vectors element by element
z                               # returns 3  6  9 12 15
z > 8                           # returns FALSE FALSE  TRUE  TRUE  TRUE
z[z > 8]                        # returns 9 12 15
z[z > 8 | z < 5]                # returns 3  9 12 15 ("|" denotes "or")
```

Sometimes it is necessary to initialize a vector of a specific length and then populate the content of the vector later. The `vector()` function, by default, creates a logical vector. A vector of a different type can be specified by using the `mode` parameter. The vector *c*, an integer vector of length 0, may be useful when the number of elements is not initially known and the new elements will later be added to the end of the vector as the values become available.

```
a <- vector(length=3)           # create a logical vector of length 3
a                               # returns FALSE FALSE FALSE
b <- vector(mode="numeric", 3)  # create a numeric vector of length 3
typeof(b)                       # returns "double"
b[2] <- 3.1                     # assign 3.1 to the 2nd element
b                               # returns 0.0 3.1 0.0
c <- vector(mode="integer", 0)  # create an integer vector of length 0
c                               # returns integer(0)
length(c)                       # returns 0
```

Although vectors may appear to be analogous to arrays of one dimension, they are technically dimensionless, as seen in the following R code. The concept of arrays and matrices is addressed in the following discussion.

```
length(b)                          # returns 3
dim(b)                             # returns NULL (an undefined value)
```

Arrays and Matrices

The `array()` function can be used to restructure a vector as an array. For example, the following R code builds a three-dimensional array to hold the quarterly sales for three regions over a two-year period and then assign the sales amount of $158,000 to the second region for the first quarter of the first year.

```
# the dimensions are 3 regions, 4 quarters, and 2 years
quarterly_sales <- array(0, dim=c(3,4,2))
quarterly_sales[2,1,1] <-  158000
quarterly_sales

, , 1

       [,1] [,2] [,3] [,4]
[1,]      0    0    0    0
[2,] 158000    0    0    0
[3,]      0    0    0    0

, , 2

     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0    0    0    0
[3,]    0    0    0    0
```

A two-dimensional array is known as a ***matrix***. The following code initializes a matrix to hold the quarterly sales for the three regions. The parameters `nrow` and `ncol` define the number of rows and columns, respectively, for the *`sales_matrix`*.

```
sales_matrix <- matrix(0, nrow = 3, ncol = 4)
sales_matrix

     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0    0    0    0
[3,]    0    0    0    0
```

R provides the standard matrix operations such as addition, subtraction, and multiplication, as well as the transpose function `t()` and the inverse matrix function `matrix.inverse()` included in the `matrixcalc` package. The following R code builds a 3×3 matrix, M, and multiplies it by its inverse to obtain the identity matrix.

```
library(matrixcalc)
M <- matrix(c(1,3,3,5,0,4,3,3,3),nrow = 3,ncol = 3) # build a 3x3 matrix
```

```
M %*% matrix.inverse(M)                          # multiply M by inverse(M)

     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

Data Frames

Similar to the concept of matrices, data frames provide a structure for storing and accessing several variables of possibly different data types. In fact, as the `is.data.frame()` function indicates, a data frame was created by the `read.csv()` function at the beginning of the chapter.

```
#import a CSV file of the total annual sales for each customer
sales <- read.csv("c:/data/yearly_sales.csv")
is.data.frame(sales)             # returns TRUE
```

As seen earlier, the variables stored in the data frame can be easily accessed using the `$` notation. The following R code illustrates that in this example, each variable is a vector with the exception of `gender`, which was, by a `read.csv()` default, imported as a ***factor***. Discussed in detail later in this section, a factor denotes a categorical variable, typically with a few finite levels such as "F" and "M" in the case of *`gender`*.

```
length(sales$num_of_orders)      # returns 10000 (number of customers)

is.vector(sales$cust_id)         # returns TRUE
is.vector(sales$sales_total)     # returns TRUE
is.vector(sales$num_of_orders)   # returns TRUE
is.vector(sales$gender)          # returns FALSE

is.factor(sales$gender)          # returns TRUE
```

Because of their flexibility to handle many data types, data frames are the preferred input format for many of the modeling functions available in R. The following use of the `str()` function provides the structure of the *`sales`* data frame. This function identifies the integer and numeric (double) data types, the factor variables and levels, as well as the first few values for each variable.

```
str(sales)               # display structure of the data frame object

'data.frame': 10000 obs. of 4 variables:
 $ cust_id      : int  100001 100002 100003 100004 100005 100006 ...
 $ sales_total  : num  800.6 217.5 74.6 498.6 723.1 ...
 $ num_of_orders: int  3 3 2 3 4 2 2 2 2 2 ...
 $ gender       : Factor w/ 2 levels "F","M": 1 1 2 2 1 1 2 2 1 2 ...
```

In the simplest sense, data frames are lists of variables of the same length. A subset of the data frame can be retrieved through ***subsetting operators***. R's subsetting operators are powerful in that they allow one to express complex operations in a succinct fashion and easily retrieve a subset of the dataset.

```
# extract the fourth column of the sales data frame
sales[,4]
# extract the gender column of the sales data frame
```

```
sales$gender
# retrieve the first two rows of the data frame
sales[1:2,]
# retrieve the first, third, and fourth columns
sales[,c(1,3,4)]
# retrieve both the cust_id and the sales_total columns
sales[,c("cust_id", "sales_total")]
# retrieve all the records whose gender is female
sales[sales$gender=="F",]
```

The following R code shows that the class of the `sales` variable is a data frame. However, the type of the `sales` variable is a list. A ***list*** is a collection of objects that can be of various types, including other lists.

```
class(sales)
"data.frame"
typeof(sales)
"list"
```

Lists

Lists can contain any type of objects, including other lists. Using the vector v and the matrix M created in earlier examples, the following R code creates *`assortment`*, a list of different object types.

```
# build an assorted list of a string, a numeric, a list, a vector,
# and a matrix
housing <- list("own", "rent")
assortment <- list("football", 7.5, housing, v, M)
assortment

[[1]]
[1] "football"

[[2]]
[1] 7.5

[[3]]
[[3]][[1]]
[1] "own"

[[3]][[2]]
[1] "rent"


[[4]]
[1] 1 2 3 4 5

[[5]]
```

```
     [,1] [,2] [,3]
[1,]    1    5    3
[2,]    3    0    3
[3,]    3    4    3
```

In displaying the contents of *assortment*, the use of the double brackets, `[[]]`, is of particular importance. As the following R code illustrates, the use of the single set of brackets only accesses an item in the list, not its content.

```
# examine the fifth object, M, in the list
class(assortment[5])            # returns "list"
length(assortment[5])           # returns 1

class(assortment[[5]])          # returns "matrix"
length(assortment[[5]])         # returns 9 (for the 3x3 matrix)
```

As presented earlier in the data frame discussion, the `str()` function offers details about the structure of a list.

```
str(assortment)
List of 5
 $ : chr "football"
 $ : num 7.5
 $ :List of 2
  ..$ : chr "own"
  ..$ : chr "rent"
 $ : int [1:5] 1 2 3 4 5
 $ : num [1:3, 1:3] 1 3 3 5 0 4 3 3 3
```

Factors

Factors were briefly introduced during the discussion of the *gender* variable in the data frame *sales*. In this case, *gender* could assume one of two levels: `F` or `M`. Factors can be ordered or not ordered. In the case of *gender*, the levels are not ordered.

```
class(sales$gender)             # returns "factor"
is.ordered(sales$gender)        # returns FALSE
```

Included with the `ggplot2` package, the *diamonds* data frame contains three ordered factors. Examining the *cut* factor, there are five levels in order of improving cut: Fair, Good, Very Good, Premium, and Ideal. Thus, `sales$gender` contains nominal data, and `diamonds$cut` contains ordinal data.

```
head(sales$gender)    # display first six values and the levels

F F M M F F
Levels: F M

library(ggplot2)
data(diamonds)         # load the data frame into the R workspace
```

```
str(diamonds)
'data.frame': 53940 obs. of 10 variables:
 $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 ...
 $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 ...
 $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 ...
 $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2  ...
 $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
 $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
 $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
 $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
 $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
 $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...

head(diamonds$cut)       # display first six values and the levels
Ideal     Premium   Good      Premium   Good      Very Good
Levels: Fair < Good < Very Good < Premium < Ideal
```

Suppose it is decided to categorize *sales$sales_totals* into three groups—small, medium, and big—according to the amount of the sales with the following code. These groupings are the basis for the new ordinal factor, spender, with levels {small, medium, big}.

```
# build an empty character vector of the same length as sales
sales_group <- vector(mode="character",
                      length=length(sales$sales_total))

# group the customers according to the sales amount
sales_group[sales$sales_total<100] <- "small"
sales_group[sales$sales_total>=100 & sales$sales_total<500] <- "medium"
sales_group[sales$sales_total>=500] <- "big"

# create and add the ordered factor to the sales data frame
spender <- factor(sales_group,levels=c("small", "medium", "big"),
                              ordered = TRUE)
sales <- cbind(sales,spender)

str(sales$spender)
Ord.factor w/ 3 levels "small"<"medium"<..: 3 2 1 2 3 1 1 1 2 1 ...

head(sales$spender)
big    medium small  medium big    small
Levels: small < medium < big
```

The `cbind()` function is used to combine variables column-wise. The `rbind()` function is used to combine datasets row-wise. The use of factors is important in several R statistical modeling functions, such as analysis of variance, `aov()`, presented later in this chapter, and the use of contingency tables, discussed next.

Contingency Tables

In R, ***table*** refers to a class of objects used to store the observed counts across the factors for a given dataset. Such a table is commonly referred to as a contingency table and is the basis for performing a statistical test on the independence of the factors used to build the table. The following R code builds a contingency table based on the `sales$gender` and `sales$spender` factors.

```
# build a contingency table based on the gender and spender factors
sales_table <- table(sales$gender,sales$spender)
sales_table
  small medium  big
F  1726   2746  563
M  1656   2723  586

class(sales_table)              # returns "table"
typeof(sales_table)             # returns "integer"
dim(sales_table)                # returns 2 3

# performs a chi-squared test
summary(sales_table)
Number of cases in table: 10000
Number of factors: 2
Test for independence of all factors:
  Chisq = 1.516, df = 2, p-value = 0.4686
```

Based on the observed counts in the table, the `summary()` function performs a chi-squared test on the independence of the two factors. Because the reported *p*-value is greater than 0.05, the assumed independence of the two factors is not rejected. Hypothesis testing and *p*-values are covered in more detail later in this chapter. Next, applying descriptive statistics in R is examined.

3.1.4 Descriptive Statistics

It has already been shown that the `summary()` function provides several descriptive statistics, such as the mean and median, about a variable such as the *`sales`* data frame. The results now include the counts for the three levels of the *`spender`* variable based on the earlier examples involving factors.

```
summary(sales)
    cust_id        sales_total      num_of_orders    gender     spender
 Min.   :100001   Min.   :  30.02   Min.   : 1.000  F:5035  small :3382
 1st Qu.:102501   1st Qu.:  80.29   1st Qu.: 2.000  M:4965  medium:5469
 Median :105001   Median : 151.65   Median : 2.000          big   :1149
 Mean   :105001   Mean   : 249.46   Mean   : 2.428
 3rd Qu.:107500   3rd Qu.: 295.50   3rd Qu.: 3.000
 Max.   :110000   Max.   :7606.09   Max.   :22.000
```

The following code provides some common R functions that include descriptive statistics. In parentheses, the comments describe the functions.

```
# to simplify the function calls, assign
x <- sales$sales_total
y <- sales$num_of_orders

cor(x,y)                          # returns 0.7508015 (correlation)
cov(x,y)                          # returns 345.2111 (covariance)
IQR(x)                            # returns 215.21 (interquartile range)
mean(x)                           # returns 249.4557 (mean)
median(x)                         # returns 151.65 (median)
range(x)                          # returns 30.02 7606.09 (min max)
sd(x)                             # returns 319.0508 (std. dev.)
var(x)                            # returns 101793.4 (variance)
```

The `IQR()` function provides the difference between the third and the first quartiles. The other functions are fairly self-explanatory by their names. The reader is encouraged to review the available help files for acceptable inputs and possible options.

The function `apply()` is useful when the same function is to be applied to several variables in a data frame. For example, the following R code calculates the standard deviation for the first three variables in *sales*. In the code, setting `MARGIN=2` specifies that the `sd()` function is applied over the columns. Other functions, such as `lapply()` and `sapply()`, apply a function to a list or vector. Readers can refer to the R help files to learn how to use these functions.

```
apply(sales[,c(1:3)], MARGIN=2, FUN=sd)
      cust_id   sales_total num_of_orders
  2886.895680    319.050782      1.441119
```

Additional descriptive statistics can be applied with user-defined functions. The following R code defines a function, `my_range()`, to compute the difference between the maximum and minimum values returned by the `range()` function. In general, user-defined functions are useful for any task or operation that needs to be frequently repeated. More information on user-defined functions is available by entering `help("function")` in the console.

```
# build a function to provide the difference between
# the maximum and the minimum values
my_range <- function(v) {range(v)[2] - range(v)[1]}
my_range(x)
7576.07
```

3.2 Exploratory Data Analysis

So far, this chapter has addressed importing and exporting data in R, basic data types and operations, and generating descriptive statistics. Functions such as `summary()` can help analysts easily get an idea of the magnitude and range of the data, but other aspects such as linear relationships and distributions are more difficult to see from descriptive statistics. For example, the following code shows a summary view of a data frame *data* with two columns *x* and *y*. The output shows the range of *x* and *y*, but it's not clear what the relationship may be between these two variables.