JAY JACOBS + BOB RUDIS

Data Driven **Security**

Analysis, Visualization and Dashboards

WILEY

Data-Driven **Security**

Analysis, Visualization and Dashboards

JAY JACOBS + BOB RUDIS

WILEY

Data-Driven Security: Analysis, Visualization and Dashboards

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

Published by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-118-79372-5

ISBN: 978-1-118-79366-4 (ebk)

ISBN: 9789-1-118-79382-4 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Limit of Liability/Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or website may provide or recommendations it may make. Further, readers should be aware that Internet websites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at http://booksupport.wiley.com. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2013954100

About the Authors

Jay Jacobs has over 15 years of experience within IT and information security with a focus on cryptography, risk, and data analysis. As a Senior Data Analyst on the Verizon RISK team, he is a co-author on their annual Data Breach Investigation Report and spends much of his time analyzing and visualizing security-related data. Jay is a co-founder of the Society of Information Risk Analysts and currently serves on the organization's board of directors. He is an active blogger, a frequent speaker, a co-host on the *Risk Science* podcast and was co-chair of the 2014 Metricon security metrics/analytics conference. Jay can be found on twitter as `@jayjacobs`. He holds a bachelor's degree in technology and management from Concordia University in Saint Paul, Minnesota, and a graduate certificate in Applied Statistics from Penn State.

Bob Rudis has over 20 years of experience using data to help defend global Fortune 100 companies. As Director of Enterprise Information Security & IT Risk Management at Liberty Mutual, he oversees their partnership with the regional, multi-sector Advanced Cyber Security Center on large scale security analytics initiatives. Bob is a serial tweeter `(@hrbrmstr)`, avid blogger `(rud.is)`, author, speaker, and regular contributor to the open source community `(github.com/hrbrmstr)`. He currently serves on the board of directors for the Society of Information Risk Analysts (SIRA), is on the editorial board of the SANS Securing The Human program, and was co-chair of the 2014 Metricon security metrics/analytics conference. He holds a bachelor's degree in computer science from the University of Scranton.

About the Technical Editor

Russell Thomas is a Security Data Scientist at Zions Bancorporation and a PhD candidate in Computational Social Science at George Mason University. He has over 30 years of computer industry experience in technical, management, and consulting roles. Mr. Thomas is a long-time community member of Securitymetrics.org and a founding member of the Society of Information Risk Analysts (SIRA). He blogs at `http://exploringpossibilityspace.blogspot.com/` and is `@MrMeritology` on Twitter.

Credits

Executive Editor
Carol Long

Senior Project Editor
Kevin Kent

Technical Editor
Russell Thomas

Senior Production Editor
Kathleen Wisor

Copy Editor
Kezia Endsley

Editorial Manager
Mary Beth Wakefield

Freelancer Editorial Manager
Rosemarie Graham

Associate Director of Marketing
David Mayhew

Marketing Manager
Ashley Zurcher

Business Manager
Amy Knies

Vice President and Executive Group Publisher
Richard Swadley

Associate Publisher
Jim Minatel

Project Coordinator, Cover
Katie Crocker

Proofreader
Nancy Carrasco

Indexer
Johnna VanHoose Dinse

Cover Image
Bob Rudis

Cover Designer
Ryan Sneed

Acknowledgments

While our names are on the cover, this book represents a good deal of work by a good number of (good) people. A huge thank you goes out to Russell Thomas, our technical editor. His meticulous attention to detail has not only made this book better, but it's also saved us from a few embarrassing mistakes. Thank you for those of you who have taken the time to prepare and share data for this project: Symantec, AlienVault, Stephen Patton, and David Severski. Thank you to Wade Baker for his contagious passion, Chris Porter for his contacts, and the RISK team at Verizon for their work and contribution of VERIS to the community. Thank you to the good folks at Wiley—especially Carol Long, Kevin Kent, and Kezia Endsley—who helped shape this work and kept us on track and motivated.

Thank you also to the many people who have contributed by responding to our emails, talking over ideas, and providing your feedback. Finally, thanks to the many vibrant and active communities around R, Python, data visualizations, and information security; hopefully, we can continue to blur the lines between those communities.

Jay Jacobs

First and foremost, I would like to thank my parents. My father gave me his passion for learning and the confidence to try everything. My mother gave me her unwavering support, even when I was busy discovering which paths not to take. Thank you for providing a good environment to grow and learn. I would also like to thank my wife, Ally. She is my best friend, loudest critic, and biggest fan. This work would not be possible without her love, support, and encouragement. And finally, I wish to thank my children for their patience: I'm ready for that game now.

Bob Rudis

This book would not have been possible without the love, support, and nigh-unending patience through many a lost weekend of my truly amazing wife, Mary, and our three still-at-home children, Victoria, Jarrod, and Ian.

Thank you to Alexandre Pinto, Thomas Nudd, and Bill Pelletier for well-timed (though you probably didn't know it) messages of encouragement and inspiration. A special thank you to the open source community and reproducible research and open data movements who are behind most of the tools and practices in this text. Thank you, as well, to Josh Corman who came up with the spiffy title for the tome.

And, a final thank you—in recipe form—to those that requested one with the book:

Pan Fried Gnocchi with Basil Pesto

- 2 C fresh Marseille basil
- 1/2 C fresh grated Romano cheese
- 1/2 C + 2 tbsp extra virgin olive oil
- 1/4 C pine nuts

- 4 garlic scapes
- Himalayan sea salt; cracked pepper
- 1 lb. gnocchi (fresh or pre-made/vacuum sealed; gnocchi should be slightly dried if fresh)

Pulse (add in order): nuts, scapes, basil, cheese. Stream in 1/2 cup of olive oil, pulsing and scraping as needed until creamy, adding salt and pepper to taste. Set aside.

Heat a heavy-bottomed pan over medium-high heat; add remaining olive oil. When hot, add gnocchi, but don't crowd the pan or go above one layer. Let brown and crisp on one side for 3–4 minutes then flip and do the same on the other side for 2–3 minutes. Remove gnocchi from pan, toss with pesto, drizzle with saba and serve. Makes enough for 3–4 people.

Contents

Introduction. xv

Chapter 1 • The Journey to Data-Driven Security 1

A Brief History of Learning from Data 2

Nineteenth Century Data Analysis 2

Twentieth Century Data Analysis 3

Twenty-First Century Data Analysis 4

Gathering Data Analysis Skills 5

Domain Expertise 6

Programming Skills 8

Data Management 10

Statistics 12

Visualization (a.k.a. Communication) 14

Combining the Skills 15

Centering on a Question 16

Creating a Good Research Question 17

Exploratory Data Analysis 18

Summary 18

Recommended Reading 19

Chapter 2 • Building Your Analytics Toolbox: A Primer on Using R and Python for Security Analysis 21

Why Python? Why R? And Why Both? 22

Why Python? 23

Why R? 23

Why Both? 24

Jumpstarting Your Python Analytics with Canopy 24

Understanding the Python Data Analysis and Visualization Ecosystem 25

Setting Up Your R Environment 29

Introducing Data Frames 33

Organizing Analyses 36

Summary 37

Recommended Reading 38

Chapter 3 • Learning the “Hello World” of Security Data Analysis. 39

Solving a Problem. 40

Getting Data 41

Reading In Data 43

Exploring Data 47

Homing In on a Question . . . 58
Summary . . . 70
Recommended Reading . . . 70

Chapter 4 • Performing Exploratory Security Data Analysis . . . 71
Dissecting the IP Address . . . 73
Representing IP Addresses . . . *73*
Segmenting and Grouping IP Addresses . . . *75*
Locating IP Addresses . . . *77*
Augmenting IP Address Data . . . 80
Association/Correlation, Causation, and Security Operations Center Analysts Gone Rogue . . . *86*
Mapping Outside the Continents . . . 90
Visualizing the ZeuS Botnet . . . *92*
Visualizing Your Firewall Data . . . *98*
Summary . . . 100
Recommended Reading . . . 101

Chapter 5 • From Maps to Regression . . . 103
Simplifying Maps . . . 105
How Many ZeroAccess Infections per Country? . . . *108*
Changing the Scope of Your Data . . . *111*
The Potwin Effect . . . *113*
Is This Weird? . . . *117*
Counting in Counties . . . *120*
Moving Down to Counties . . . *122*
Introducing Linear Regression . . . 125
Understanding Common Pitfalls in Regression Analysis . . . *130*
Regression on ZeroAccess Infections . . . *131*
Summary . . . 136
Recommended Reading . . . 136

Chapter 6 • Visualizing Security Data . . . 137
Why Visualize? . . . 138
Unraveling Visual Perception . . . *139*
Understanding the Components of Visual Communications . . . 144
Avoiding the Third Dimension . . . *144*
Using Color . . . *146*
Putting It All Together . . . *148*
Communicating Distributions . . . *154*
Visualizing Time Series . . . *156*
Experiment on Your Own . . . *157*
Turning Your Data into a Movie Star . . . 158
Summary . . . 159
Recommended Reading . . . 160

Chapter 7 • Learning from Security Breaches 161

Setting Up the Research 162

Considerations in a Data Collection Framework 164

Aiming for Objective Answers *164*

Limiting Possible Answers *164*

Allowing "Other," and "Unknown" Options *164*

Avoiding Conflation and Merging the Minutiae *165*

An Introduction to VERIS 166

Incident Tracking *168*

Threat Actor *168*

Threat Actions *169*

Information Assets *173*

Attributes *173*

Discovery/Response *176*

Impact *176*

Victim *177*

Indicators *179*

Extending VERIS with Plus *179*

Seeing VERIS in Action 179

Working with VCDB Data 181

Getting the Most Out of VERIS Data *185*

Summary 189

Recommended Reading 189

Chapter 8 • Breaking Up with Your Relational Database 191

Realizing the Container Has Constraints 195

Constrained by Schema *196*

Constrained by Storage *198*

Constrained by RAM *199*

Constrained by Data *200*

Exploring Alternative Data Stores 200

BerkeleyDB *201*

Redis *203*

Hive *207*

MongoDB *210*

Special Purpose Databases *214*

Summary 215

Recommended Reading 216

Chapter 9 • Demystifying Machine Learning 217

Detecting Malware 218

Developing a Machine Learning Algorithm *220*

Validating the Algorithm *221*

Implementing the Algorithm *222*

Benefiting from Machine Learning 226

Answering Questions with Machine Learning 226

Measuring Good Performance. 227

Selecting Features 228

Validating Your Model 230

Specific Learning Methods. 230

Supervised 231

Unsupervised. 234

Hands On: Clustering Breach Data. 236

Multidimensional Scaling on Victim Industries 238

Hierarchical Clustering on Victim Industries 240

Summary 242

Recommended Reading 243

Chapter 10 • Designing Effective Security Dashboards. 245

What Is a Dashboard, Anyway? 246

A Dashboard Is Not an Automobile 246

A Dashboard Is Not a Report 248

A Dashboard Is Not a Moving Van 251

A Dashboard Is Not an Art Show. 253

Communicating and Managing "Security" through Dashboards 258

Lending a Hand to Handlers. 258

Raising Dashboard Awareness 260

The Devil (and Incident Response Delays) Is in the Details 262

Projecting "Security" 263

Summary 267

Recommended Reading 267

Chapter 11 • Building Interactive Security Visualizations 269

Moving from Static to Interactive 270

Interaction for Augmentation 271

Interaction for Exploration 274

Interaction for Illumination 276

Developing Interactive Visualizations 281

Building Interactive Dashboards with Tableau 281

Building Browser-Based Visualizations with D3. 284

Summary 294

Recommended Reading 295

Chapter 12 • Moving Toward Data-Driven Security 297

Moving Yourself toward Data-Driven Security 298

The Hacker 299

The Statistician 302

The Security Domain Expert. 302

The Danger Zone 303

Moving Your Organization toward Data-Driven Security 303
Ask Questions That Have Objective Answers 304
Find and Collect Relevant Data 304
Learn through Iteration 305
Find Statistics. 306
Summary 308
Recommended Reading 308

Appendix A • Resources and Tools 309

Appendix B • References 313

Index • 321

Introduction

“It’s a dangerous business, Frodo, going out your door. You step onto the road, and if you don’t keep your feet, there’s no knowing where you might be swept off to.”

Bilbo Baggins, *The Fellowship of the Ring*

In recent years, cybersecurity has taken center stage in the personal and professional lives of the majority of the global population. Data breaches are a daily occurrence, and intelligent adversaries target consumers, corporations, and governments with practically no fear of being detected or facing consequences for their actions. This is all occurring while the systems, networks, and applications that comprise the backbones of commerce and critical infrastructure are growing ever more complex, interconnected, and unwieldy.

Defenses built solely on the elements of faith-based security—unaided intuition and "best" practices—are no longer sufficient to protect us. The era of the security shaman is rapidly fading, and it's time to adopt the proven tools and techniques being used in other disciplines to take an evolutionary step into *Data-Driven Security*.

Overview of the Book and Technologies

Data-Driven Security: Analysis, Visualization and Dashboards has been designed to take you on a journey into the world of security data science. The start of the journey looks a bit like the word cloud shown in Figure 1, which was created from the text in the chapters of this book. You have a great deal of information available to you, and *may* be able to pick out a signal or two within the somewhat hazy noise on your own. However, it's like looking for a needle in a haystack without a magnet.

project http relatively
representation code make combination require
attributes SIMPLE REGRESSION following algorithm FIGURE know
really BREACH MEASURE interactive little
report VERIS validation set communicate work perform machine
present TEST features
EVENTS CLUSTERING HELP DASHBOARD POINTS bit USE line categories SECTION framework indicators multiple
graphic COLOR NEED TOOLS output significant
definitely level understand executive
SYSTEMS development value type DATA different answer start NODES
CLASSIFICATION important possible focus fields Python correlation
STATE infections information quantitative RESEARCH selection interesting
malware support number variables LIST count
OPTIONS COMPARE FUNCTION available ORGANIZATION natural generate actually
display CALCULATE DRIVEN questions want apply simply
begin based elements PROVIDE relationship DISTRIBUTION
QUERY APPLICATION OPERATIONS come security THINGS MAP EXPLORE book
installed people identify skills analysts linear BETTER STANDARD
benefit DESCRIPTIVE SINGLE VISUALIZATION quite internet population
movements observations example chapter good country plot AlienVault
VARIATION basic group way STATISTICS record graph build
going vulnerabilities
think network analysis LEARNING model complex objects ATTEMPT
completely consumer details time designed especially
change effective LOOK prediction INCIDENT 11 SPECIFIC include NORMAL
new structure experience run CREATE process coefficient chart column large
domain language database order limited MEAN reputation management
problem consider CHALLENGE PATTERNS
probably industry TECHNIQUES ADDRESSES APPROACH
world ACTION results individual methods collect environment hosts
represent STRAIGHTFORWARD
table program TRAINING
components relational

FIGURE 1

You'll have much more success identifying what matters (see Figure 2) if you apply the right tools in the most appropriate way possible.

project http relatively
representation make
attributes SIMPLE code combination
really BREACH MEASURE REGRESSION following algorithm FIGURE know require
report VERIS validation interactive little
present
TEST features set communicate work perform machine
EVENTS CLUSTERING line categories
graphic COLOR HELP DASHBOARD POINTS bit USE NEED indicators SECTION framework multiple
definitely level TOOLS output significant
SYSTEMS development understand value type executive DATA different answer start NODES
CLASSIFICATION important possible focus fields Python correlation
STATE selection interesting
malware support infections information quantitative variables RESEARCH
OPTIONS number LIST count
display CALCULATE COMPARE FUNCTION available ORGANIZATION natural generate actually
begin based elements questions want apply simply
QUERY APPLICATION PROVIDE DRIVEN relationship DISTRIBUTION
OPERATIONS come security THINGS MAP EXPLORE book
installed people identify skills BETTER STANDARD
benefit DESCRIPTIVE VISUALIZATION quite analysts linear population
movements SINGLE internet good country
observations example chapter plot AlienVault
VARIATION basic group way STATISTICS record graph build
going vulnerabilities complex
think network analysis model objects ATTEMPT
completely consumer details LEARNING designed especially
change time prediction INCIDENT
effective LOOK run 11 SPECIFIC include NORMAL
new structure experience CREATE process coefficient chart column large
domain language database order consider limited MEAN reputation management
problem TECHNIQUES ADDRESSES APPROACH CHALLENGE PATTERNS
probably industry hosts
world ACTION results individual methods collect environment
STRAIGHTFORWARD
represent table program TRAINING
components relational

Figure 2

This book focuses on Python and R as the foundational data analysis tools, but also introduces the design and creation of modern static and interactive visualizations with HTML5, CSS, and JavaScript. It also provides background on and security use cases for modern NoSQL databases.

How This Book Is Organized

Rather than have you gorge at an all-you-can-eat buffet, the chapters are more like tapas—each with their own distinct flavor profiles and textures. Like the word *tapas* itself suggests, each chapter covers a different foundational topic within security data science and provides plenty of pointers for further study.

Chapter 1 lays the foundation for the journey and provides examples of how other disciplines have evolved into data-driven practices. It also provides an overview of the skills a security data scientist needs.

Chapters 2, 3, and 4 dive right into the tools, technologies, and basic techniques that should be part of every security data scientists' toolbox. You'll work with AlienVault's IP Reputation database (one of the most thorough sources of malicious nodes publicly available) and take a macro look at the ZeuS and ZeroAccess botnets. We introduce the analytical side of Python in Chapters 2 and 3. Then we thrust you into the world of statistical analysis software with a *major* focus on the R language in the remainder of the book. Unlike traditional introductory texts in R (or statistics in general), we use security data throughout the book to help make the concepts as real and practical as possible for the information security professional.

Chapter 5 introduces some techniques for creating maps and introduces some core statistical concepts, along with a lesson or two about extraterrestrial visitors.

Chapter 6 delves into the biological and cognitive science foundations of visual communication (data visualization) and even shows you how to animate your security data.

This lays a foundation for learning how to analyze and visualize security breaches in **Chapter 7**, where you'll also have an opportunity to work with real incident data.

Chapter 8 covers modern database concepts with new tricks for traditional database deployments and new tools with a range of NoSQL solutions discussed. You'll also get tips on how to answer the question, "Have we seen this IP address on our network?"

Chapter 9 introduces you to the exciting and relatively new world of machine learning. You'll learn about the core concepts and explore a handful of machine-learning techniques and develop a new appreciation for how algorithms can pick up patterns that your intuition might never recognize.

Chapters 10 and 11 give you practical advice and techniques for building effective visualizations that will both communicate and (hopefully) impress your consumers. You'll use everything from Microsoft Excel to state of the art tools and libraries, and be able to translate what you've learned outside of security. Visualization concepts are made even more tangible through "makeovers" of security dashboards that many of you may be familiar with.

Finally, we show you how to apply what you've learned at both a personal and organizational level in **Chapter 12**.

Who Should Read This Book

We wrote this book because we both thoroughly enjoy working with data and wholeheartedly believe that we can make significant progress in improving cybersecurity if we take the time to understand how to ask the right questions, perform accurate and reproducible analyses on data, and communicate the results in the most compelling ways possible.

Readers will get the most out of this book if they come to it with some security domain experience and the ability to do basic coding or scripting. If you are already familiar with Python, you can skip the introduction to it in Chapter 2 and can skim through much of Chapter 3. We level the field a bit by introducing and focusing on R, but you would do well to make your way through all the examples and listings that use R throughout the book, as it is an excellent language for modern data science. If you are new to programming, Chapters 2, 3, and 4 will provide enough of an immersive experience to help you see if it's right for you.

We place emphasis on statistical and machine learning across many chapters and do not recommend skipping any of that content. However, you *can* hold off on Chapter 9 (which discusses machine learning) until the very end, as it will not detract significantly from the flow of the book.

If you know databases well, you need only review the use cases in Chapter 8 to ensure you're thinking about all the ways you can use modern and specialized databases in security use cases.

Unlike many books that discuss dashboards, the only requirements for Chapter 10 are Microsoft Excel or OpenOffice Calc, as we made no assumptions about the types of tools and restrictions you have to work with in your organization. You can also save Chapter 11 for future reading if you have no desire to build interactive visualizations.

In short, though we are writing to Information Technology and Information Security professionals, students, consultants, and anyone looking for more about the how-to of analyzing data and making it understandable for protecting networks will find what they need in this book.

Tools You Will Need

Everything you need to follow along with the exercises is freely available:

- **The R project** (`http://www.r-project.org`)—Most of the examples are written in R, and with the wide range of community developed packages like ggplot2 (`http://ggplot2.org`) almost anything is possible.
- **RStudio** (`http://www.rstudio.com/`)—It will be *much* easier to get to know R and run the examples if you use the RStudio IDE.
- **Python** (`http://www.python.org`)—A few of the examples leverage Python and with add-on packages like pandas (`http://pandas.pydata.org`) makes this a very powerful platform.
- **Sublime Text** (`http://www.sublimetext.com/`)—This, or another robust text editor, will come in very handy especially when working with HTML/CSS/JavaScript examples.
- **D3.js** (`http://d3js.org/`)—Grabbing a copy of D3 and giving the basics a quick read through ahead of Chapter 11 will help you work through the examples in that chapter a bit faster.
- **Git** (`http://git-scm.com/`)—You'll be asked to use git to download data at various points in the book, so installing it now will save you some time later.
- **MongoDB** (`http://www.mongodb.org/`)—MongoDB is used in Chapter 8, so getting it set up early will make those examples less cumbersome.
- **Redis** (`http://redis.io/`) —This, too, is used in some examples in Chapter 8.
- **Tableau Public** (`http://www.tableausoftware.com/`)—If you intend to work with the survey data in Chapter 11, having a copy of Tableau Public will be useful.

Additionally, all of the code, examples, and data used in this book are available through the companion website for this book (`www.wiley.com/go/datadrivensecurity`).

We recommend using Linux or Mac OS, but all of the examples should work fine on modern flavors of Microsoft Windows as well.

What's on the Website

As mentioned earlier, you'll want to check out the companion website `www.wiley.com/go/datadrivensecurity` for the book, which has the full source code for all code listings, the data files used in the examples, and any supporting documents (such as Microsoft Excel files).

The Journey Begins!

You have everything you need to start down the path to *Data-Driven Security*. We hope your journey will be filled with new insights and discoveries and are confident you'll be able to improve your security posture if you successfully apply the principles you're about to learn.

1

The Journey to Data-Driven Security

"It ain't so much the things we don't know that get us into trouble. It's the things we know that just ain't so."

Josh Billings, Humorist

This book isn't really about data analysis and visualization.

Yes, almost every section is focused on those topics, but being able to perform good data analysis and produce informative visualizations is just a means to an end. You never (okay, rarely) analyze data for the sheer joy of analyzing data. You analyze data and create visualizations to gain new perspectives, to find relationships you didn't know existed, or to simply discover new information. In short, you do data analysis and visualizations to learn, and that is what this book is about. You want to learn how your information systems are functioning, or more importantly how they are failing and what you can do to fix them.

The cyber world is just too large, has too many components, and has grown far too complex to simply rely on intuition. Only by augmenting and supporting your natural intuition with the science of data analysis will you be able to maintain and protect an ever-growing and increasingly complex infrastructure. We are not advocating replacing people with algorithms; we are advocating arming people with algorithms so that they can learn more and do a better job. The data contains information, and you can learn better with the information in the data than without it.

This book focuses on using real data—the types of data you have probably come across in your work. But rather than focus on huge discoveries in the data, this book focuses more on the process and less on the result. As a result of that decision, the use cases are intended to be exemplary and introductory rather than knock-your-socks-off cool. The goal here is to teach you new ways of looking at and learning from data. Therefore, the analysis is intended to be new ground in terms of technique, not necessarily in conclusion.

A Brief History of Learning from Data

One of the best ways of appreciating the power of statistical data analysis and visualization is to look back in history to a time when these methods were first put to use. The following cases provide a vivid picture of "before" versus "after," demonstrating the dramatic benefits of the then-new methods.

Nineteenth Century Data Analysis

Prior to the twentieth century, the use of data and statistics was still relatively undeveloped. Although great strides were made in the eighteenth century, much of the scientific research of the day used basic descriptive statistics as evidence for the validity of the hypothesis. The inability to draw clear conclusions from noisy data (and almost all real data is more or less noisy) made much of the scientific debates more about opinions of the data than the data itself. One such fierce debate[1] in the nineteenth century was between two medical professionals in which they debated (both with data) the cause of cholera, a bacterial infection that was often fatal.

The cholera outbreak in London in 1849 was especially brutal, claiming more than 14,000 lives in a single year. The cause of the illness was unknown at that time and two competing theories from two researchers emerged. Dr. William Farr, a well-respected and established epidemiologist, argued that cholera was caused by air pollution created by decomposing and unsanitary matter (officially called the ***miasma*** theory). Dr. John Snow, also a successful epidemiologist who was not as widely known as Farr, put forth the theory that cholera was spread by consuming water that was contaminated by a "special animal poison" (this was prior to the discovery of bacteria and germs). The two debated for years.

Farr published the "Report on the Mortality of Cholera in England 1848–49" in 1852, in which he included a table of data with eight possible explanatory variables collected from the 38 registration districts of London.

[1] And worthy of a bona fide Hollywood plot as well. See `http://snowthemovie.com/`

In the paper, Farr presented some relatively simple (by today's standards) statistics and established a relationship between the average elevation of the district and cholera deaths (lower areas had more deaths). Although there was also a relationship between cholera deaths and the source of drinking water (another one of the eight variables he gathered), he concluded that it was not nearly as significant as the elevation. Farr's theory had data and logic and was accepted by his peers. It was adopted as fact of the day.

Dr. John Snow was passionate and vocal about his disbelief in Farr's theory and relentless in proving his own. It's said he even collected data by going door to door during the cholera outbreak in the Soho district of 1854. It was from that outbreak and his collected data that he made his now famous map in Figure 1-1. The hand-drawn map of the Soho district included little tick marks at the addresses where cholera deaths were reported. Overlaying the location of water pumps where residents got their drinking water showed a rather obvious clustering around the water pump on Broad Street. With his map and his passionate pleas, the city did allow the pump handle to be removed and the epidemic in that region subsided. However, this wasn't enough to convince his critics. The cause of cholera was heavily debated even beyond John Snow's death in 1858.

The cholera debate included data and visualization techniques (long before computers), yet neither had been able to convince the opposition. The debate between Snow and Farr was re-examined in 2003 when statisticians in the UK evaluated the data Farr published in 1852 with modern methods. They found that the data Farr pointed to as proof of an airborne cause actually supported Snow's position. They concluded that if modern statistical methods were available to Farr, the data he collected would have changed his conclusion. The good news of course, is that these statistical methods are available today to you.

Twentieth Century Data Analysis

A few years before Farr and Snow debated cholera, an agricultural research station north of London at Rothamsted began conducting experiments on the effects of fertilizer on crop yield. They spent decades conducting experiments and collecting data on various aspects such as crop yield, soil measurements, and weather variables. Following a modern-day logging approach, they gathered the data and diligently stored it, but they were unable to extract the full value from it. In 1919 they hired a brilliant young statistician named Ronald Aylmer Fisher to pore through more than 70 years of data and help them understand it. Fisher quickly ran into a challenge with the data being confounded, and he found it difficult to isolate the effect of the fertilizer from other effects, such as weather or soil quality. This challenge would lead Fisher toward discoveries that would forever change not just the world of statistics, but almost every scientific field in the twentieth century.

What Fisher discovered (among many revolutionary contributions to statistics) is that if an experiment was designed correctly, the influence of various effects could not just be separated, but also could be measured and their influence calculated. With a properly designed experiment, he was able to isolate the effects of weather, soil quality, and other factors so he could compare the effects of various fertilizer mixtures. And this work was not limited to agriculture; the same techniques Fisher developed at Rothamsted are still used widely today in everything from medical trials to archaeology dig sites. Fisher's work, and the work of his peers, helped revolutionize science in the twentieth century. No longer could scientists simply collect and present their data as evidence of their claim as they had in the eighteenth century. They now had the tools to design robust experiments and the techniques to model how the variables affected their experiment and observations.

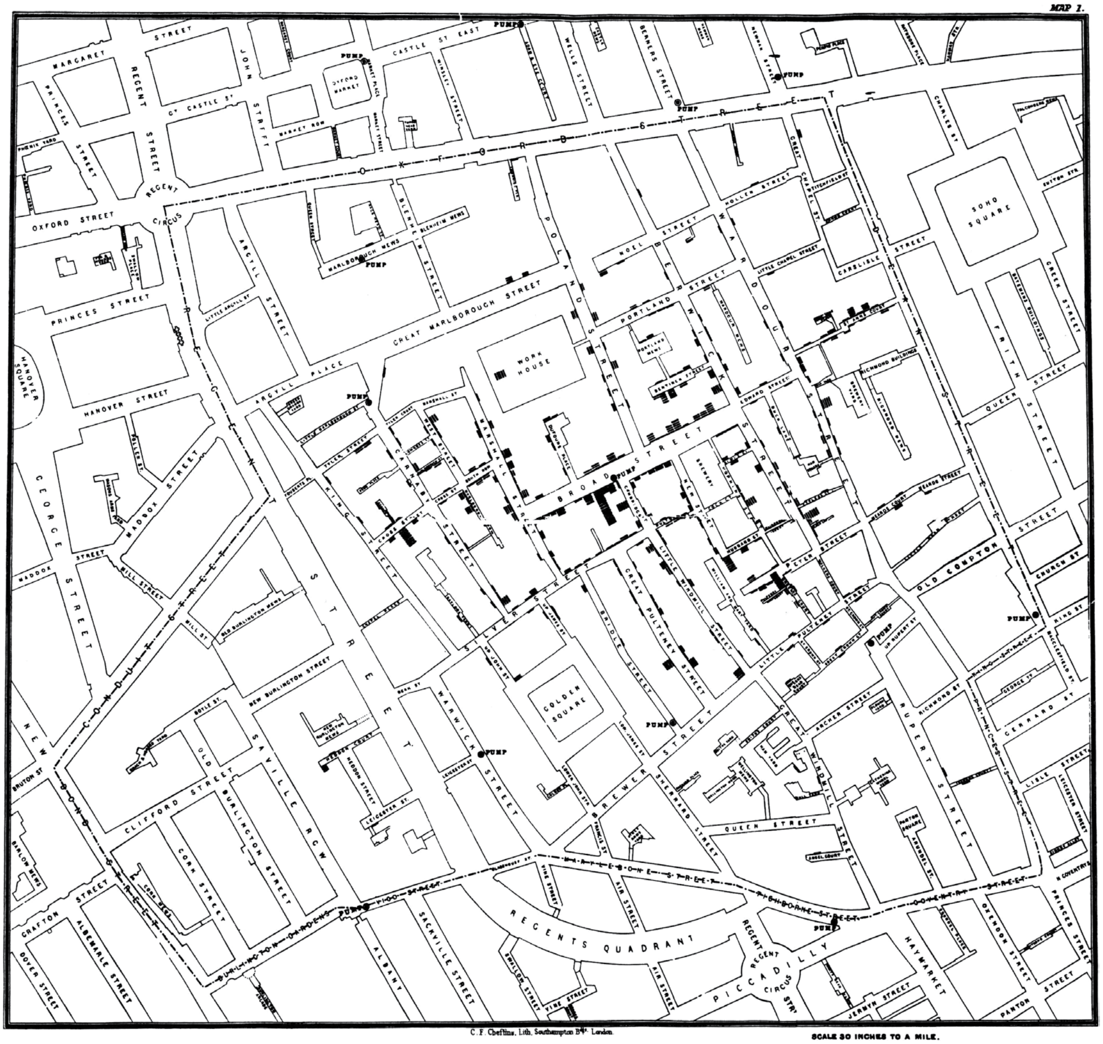

FIGURE 1-1 *Hand-drawn map of the areas affected by cholera*

At this point, the world of science included statistical models. Much of the statistical and science education focused on developing and testing these models and the assumptions behind them. Nearly every statistical problem started with the question—"What's the model?"—and ended with the model populated to allow description and even prediction using the model. This represented a huge leap forward and enabled research never before possible. If it weren't for computers, the world would probably still consider these techniques to be modern. But computers are ubiquitous and they have enabled a whole new approach to data analysis that was both impossible and unfathomable prior to their development.

Twenty-First Century Data Analysis

It's difficult to pull out any single person or event that captures where data analysis is today like Farr and Fisher captured the previous stages of data analysis. The first glimpse at what was on the horizon came from John Tukey, who wrote in 1962 that data analysis should be thought of as different from statistics (although analysis leveraged statistics). He stated that data analysis must draw from science more than mathematics (can you see the term "data science" in there?). Tukey was not only an accomplished statistician, having contributed numerous procedures and techniques to the field, but he was also an early proponent of visualization techniques for the purpose of describing and exploring the data. You will come back to some of Tukey's work later in this chapter.

Let's jump ahead to a paper written in 2001 by Leo Breiman, a statistician who focused on machine learning algorithms (which are discussed in Chapter 9). In the paper he describes a new culture of data analysis that does not focus on defining a data model ***of nature*** but instead derives an algorithmic model ***from nature***. This new culture has evolved within computer science and engineering largely outside (or perhaps alongside) traditional statistics. New approaches are born from the practical problems created by the information age, which created large quantities of complex and noisy data. The revolutionary idea that Breiman outlined in this paper is that models should be judged on their predictive accuracy instead of validating the model with traditional statistical tests (which are not without value by the way).

At face value you may think of testing "predictive accuracy" by gathering data today and determining how it predicts the world of tomorrow, but that's not what the idea is about. The idea is about splitting the data of today into two data sets, using the first data set to generate (or "train") an algorithm and then validating (or "test") its predictive accuracy on the second data set. To increase the power of this approach, you can iterate through this process multiple times, splitting the data into various training and test sets, generating and validating as you go. This approach is not well suited to small data sets, but works remarkably well with modern data sets.

There are several main differences between data analysis in the modern information age and the agricultural fields of Rothamsted. First, there is a large difference in the available sample size. "Classic" statistical techniques were largely limited by what the computers of the day could handle ("computers" were the people hired to "compute" all day long). With generally smaller samples, generating a training and test was impractical. However, modern environments are recording hundreds of variables generated across thousands of systems. Large sample sizes are the norm, not the exception.

Second, for many environments and industries, a properly designed experiment is unlikely if not completely impossible. You cannot divide your networks into control and test groups, nor would you want to test the efficacy of a web application firewall by only protecting a portion of a critical application. One effect of these environmental limits is a much higher noise-to-signal ratio in the data. The techniques of machine learning (and the related field of data mining) have evolved with the challenges of modern data in mind.

Finally, knowledge of statistics is just one skill of many that contributes to successful data analysis in the twenty-first century. With that in mind, the next section spends some time looking at the various skills and attributes that support a good data analysis.

Gathering Data Analysis Skills

We know there is a natural allure to data science and everyone wants to achieve that sexy mystique surrounding security data analysis. Although we have focused on this concept of data analysis so far, it takes more than just analytic skills to create the mystique that everyone is seeking. You need to combine statistics and data analysis with visualization techniques, and then leverage the computing power and mix with a healthy dose of domain (information security) knowledge. All of this begins not with products or tools but with your own skills and abilities.

Before getting to the skills, there are a couple underlying personality traits we see in data analysts that we want to discuss: curiosity and communication. Working with data can at times be a bit like an archeological dig—spending hour after hour with small tools in the hope of uncovering even the tiniest of insights. So it is with data analysis—pearls of wisdom are nestled deep within data just waiting to be discovered and presented to an eagerly awaiting audience. It is only with that sense of wonder and curiosity that the hours spent cleaning and preparing data are not just tolerable, but somehow exciting and worth every moment. Because there is that moment, when you're able to turn a light on in an otherwise dark room, when you can describe some phenomenon or explain some pattern, when it all becomes worth it. That's what you're after. You are uncovering those tiny moments of enlightenment hidden in plain sight if you know where to look.

Once you turn that light on, you have to bring others into the room for the discovery; otherwise, you will have constructed a house that nobody lives in. It's not enough to point at your work and say, "see!" You have to step back and think of the best way to communicate your discovery. The complexity present in the systems and the analysis makes it difficult to convey the results in a way that everyone will understand what you have discovered. Often times it takes a combination of words, numbers, and pictures to communicate the data's insights. Even then, some people will take away nothing, and others will take away too much. But there is still a need to condense this complexity into a paragraph, table, or graphic.

Although we could spend an entire book creating an exhaustive list of skills needed to be a good security data scientist, this chapter covers the following skills/domains that a data scientist will benefit from knowing within information security:

- **Domain expertise**—Setting and maintaining a purpose to the analysis
- **Data management**—Being able to prepare, store, and maintain data
- **Programming**—The glue that connects data to analysis
- **Statistics**—To learn from the data
- **Visualization**—Communicating the results effectively

It might be easy to label any one of these skills as the most important, but in reality, the whole is greater than the sum of its parts. Each of these contributes a significant and important piece to the workings of security data science.

Domain Expertise

The fact that a data scientist needs domain expertise should go without saying and it may seem obvious, but data analysis is only meaningful when performed with a higher purpose in mind. It's your experience with information security that will guide the direction of the analysis, provide context to the data, and help apply meaning to the results. In other words, domain expertise is beneficial in the beginning, middle, and end of all your data analysis efforts.

And Why Expertise Shouldn't Get in the Way

We are probably preaching to the choir here. If you are reading this book, it is probably safe to assume that you have domain expertise and see value in moving toward a data-driven approach in information security. Therefore, rather than spend the effort discussing the benefits of domain expertise in data analysis, this section covers some objections you might encounter as other domain experts (or skeptical leadership) are brought into the data analysis effort.

People are smarter than models. There are those who hold the opinion that people will always outperform algorithms (or statistics, or models) and there is some truth to this. Teaching a machine, for example, to catch a fly ball is remarkably challenging. As Kahneman and Klein point out in their 2009 paper titled *Conditions for Intuitive Expertise: a Failure to Disagree,* however, determining when people will outperform algorithms is heavily dependent on the environment of the task. If the environment is complex and feedback is delayed or ambiguous, algorithms will generally and relatively consistently outperform human judgment. So, the question then becomes, how complex is the security of the information systems and how clear is the feedback? When you make a change or add a security control, how much feedback do you receive on how well it is actually protecting the information asset?

The result is that information security occurs in a very complex environment, but that doesn't mean you put all your eggs in the algorithm basket. What it does mean is that you should have some healthy skepticism about any approach that relies purely on human judgment, and you should seek ways to augment and support that expertise. That's not to compare algorithms to human judgment. It's not wise to set up an either-or choice. You do, however, want to compare human judgment combined with algorithms and data analysis against human judgment alone. You do not want to remove the human element, but you should be skeptical of unsupported opinion. In a complex environment, it is the combination of human intuition and data analysis that will produce the best results and create the best opportunity for learning and securing the infrastructure.

It's just lying with statistics. This expresses a general distrust in statistics and data analysis, which are often abused and misused (and in some cases flat out made up) for the sake of serving some ulterior motive. In a way, this distrust is grounded in a collective knowledge of just how easy it is to social-engineer people. However, you are in a different situation since your motive is to learn from the data. You are sitting on mounds of data that hold information and patterns just waiting to be discovered. Not leveraging data analysis because statistics are misused is like not driving a car because they are sometimes used as get-away vehicles. You need to be comfortable with adding statistics to your information security toolkit.

This is not to say that data analysis is infallible. There may be times when the analysis provides the wrong answer, perhaps through poor data collection, under-trained analysts, a mistake in the process, or simply using Excel (couldn't resist). But what you should see is simply fewer mistakes when you apply the rigor of data analysis combined with your expertise. Again, the key is combining data analysis and expertise.

This ain't rocket science. This statement has two insinuations. First, it says that whatever the problem is you're trying to solve, you should be able to solve it with common sense. But this concern goes back to the first point, which is thinking that people outperform algorithms consistently and a group of people around a conference table looking at a complex environment can solve the (complex) problem without the need for data analysis. But as we discussed, you should pull a chair up to the conference table for the data analysis because you are generally better off with it than without it.

The second implication of the statement is that data analysis is too complicated and will cost too much (in time, money, or resources). This view is simply misinformed and the objection is more likely to be a concern about an uncomfortable change in practices than a concern about time spent with data analysis. Many of the tools are open source (if the organization is averse to open source, there are plenty of commercial solutions out there as well) and the only real commitment is in the time to learn some of the basic techniques and methods in this book. The actual analysis itself can be fairly quick, and with the right combination of tools and experience, it can be done in real time.

We don't have the data. An alternate form of this objection is saying that we don't have actuarial-quality data (which is more prevalent when you start talking about risk analysis). Data detractors argue that anything less than perfect data is worthless and prevents you from creating well-designed experiments. This statement is untrue and quite harmful. If you were to wait around for perfect data, you would always be waiting and many learning opportunities would be missed. More importantly and to the heart of this objection, you don't ***need*** perfect data. You just need methods to learn from the messy data you do have. As Douglas Hubbard wrote in 2010 in his book *How to Measure Anything,* "The fact is that we often have more data than we think, we need less data than we think, and getting more data through observation is simpler than we think." So, generally speaking, data for security analysis absolutely exists; often times it is just waiting to be collected. You can, with a few alterations, collect and accurately analyze even sketchy data. Modern data analysis methods have evolved to work with the noisy, incomplete, and imperfect data you have.

But we will fall off the edge of the world. There is one last point to consider and it's not so much an objection to data analysis, but an obstacle in data analysis. When you are seen as a domain expert, you are expected to provide answers with confidence. The conflict arises when confidence is confused with certainty. Data analysis requires just enough self-awareness and humility to create space for doubt in the things you think you know. Even though you may confidently state that passwords should be so many characters long with a certain amount of complexity, the reality is you just don't know where the balance is between usability and security. Confidence needs to be balanced with humility and the ability to update your beliefs based on new evidence. This obstacle in data analysis is not just limited to the primary analyst. Other domain experts involved in the analysis will have to come face to face with their own humility. Not everyone will want to hear that his or her world isn't flat.

Programming Skills

As much as we'd like to portray data science as a glamorous pursuit of truth and knowledge, as we've said, it can get a little messy. Okay, that's an understatement. Working with data is a great deal more uncertain and unkempt than people think and, unfortunately, the mess usually appears early on when you're attempting to collect and prepare the data. This is something that many classes in statistics never prepare their students for. Professors hand out rather nice and neat data sets ready to be imported into the analysis tool *du jour*. Once you leave the comfort of the classroom, you quickly realize that the world is a disorganized and chaotic place and data (and its subsequent analyses) are a reflection of that fact.

This is a cold, hard lesson in data science: Data comes to you in a wide range of formats, states, and quality. It may be embedded in unstructured or semi-structured log files. It may need to be scraped from a website. Or, in extreme cases, data may come in an overly complex and thoroughly frustrating format known as XML. Somehow, you must find a way to collect, coax, combine, and massage what you're given into a format that supports further analysis. Although this could be done with a lot of patience, a text editor, and judicious use of summer interns, the ability to whip together a script to do the work will provide more functionality, flexibility, and efficiency in the long run. Learning even basic programming skills opens up a whole range of possibilities when you're working with data. It frees you to accept multiple forms of data and manipulate it into whatever formats work best with the analysis software you have. Although there is certainly a large collection of handy data conversion tools available, they cannot anticipate or handle everything you will come across. To be truly effective while working with data, you need to adapt to the data in your world, not vice versa.

AES-256-Bit Keys Are Twice as Good as AES-128, Right?

One natural assumption about AES-256-bit keys is that because they are twice as long as AES-128-bit keys, they are twice as secure. We've been around information security people when they force a project to use 256-bit keys because they are "twice as good." Well, let's look into the math. First, you are talking about bits here, and although 256 bits is twice as many bits as 128, 256-bit keys actually have 2^{128} ***times*** more keys. Break out your slide rules and work through an exercise to try to answer a simple question: If you had access to the world's fastest super-computer, how many 128-bit keys could you crack?

The world's fastest super computer (at the time of this writing) is the ***Tianhe-2*** in China, which does around 34 petaflops (34 x 10^{15} floating point operations) per second. If you assume it takes one operation to generate a key and one operation to test it (this is an absurd and conservative assumption), you can test an amazing 17 x 10^{15} keys per second. But a 128-bit key has 3.4 x 10^{38} possibilities, which means after a full year of cracking 128-bit keys, you will have exhausted 1.6 x 10^{-13} percent of the key space. Even if you run the super-computer for 1,000 years, you will only have searched 0.0000000000016 percent of all the possible keys (and spent a fortune on electricity).

To put this simply, ***the probability of brute-force cracking a 128-bit key is already so infinitesimally small that you could easily round off that probability to zero***. But let's be professional here and say, "Moving from a 128-bit key to a 256 is moving the probability from really-super-duper-infinitesimally-small to really-super-duper-infinitesimally-small x 2^{128}."

Any modern language will support basic data manipulation tasks, but scripting languages such as Python and R appear to be used slightly more often in data analysis than their compiled counterparts (Java and C). However, the programming language is somewhat irrelevant. The end results (and a happy analyst) are more important than picking any "best" language. Whatever gets the job done with the least amount of effort is the best language to use. We generally flip between Python (pandas) and R for cleaning and converting data (or perhaps some Perl if we're feeling nostalgic) and then R or pandas for the analysis and visualization. Learning web-centric languages like HTML, CSS, and JavaScript will help create interactive visualizations for the web, as you'll see in Chapter 11, but web languages are not typically involved in the preparation and analysis of data.

There is a tool worth mentioning in this section—the "gateway tool" between a text editor and programming—known as the ***spreadsheet*** (such as Microsoft Excel or OpenOffice Calc). Spreadsheets allow non-programmers to do some amazing things and get some quick and accessible results. Although spreadsheets have their own sets of challenges and drawbacks, they also have some benefits. If the data is not too large or complex and the task is not deciding the future of the world economy (see the following sidebar), Excel may be the best tool for the job. We strongly suggest seeing Excel as a temporary solution. It does well at quick one-shot tasks. But if you have a repeating analytic task or model that is used repeatedly, it's best to move to some type of structured programming language.

As a cleaning tool, spreadsheets seem like a very good solution at first (especially for those who have developed some skill with them). But spreadsheets are event-driven, meaning they work through clicking, typing, and dragging. If you want to apply a conversion to a row of data, you have to click to select the row and apply a conversion. This works for small data sets or quick tasks, but trust us, you will (more often than