

JORDAN GOLDMEIER

DATA SMART

USING DATA SCIENCE TO TRANSFORM INFORMATION INTO INSIGHT

SECOND EDITION

WILEY

Data Smart

Second Edition

Data Smart

Using Data Science to Transform Information into Insight

Second Edition

Jordan Goldmeier

WILEY

Copyright © 2024 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada and the United Kingdom.

ISBNs: 9781119931386 (Paperback), 9781119931485 (ePDF), 9781119931393 (ePub)

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at www.wiley.com/go/permission.

Trademarks: WILEY, and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Further, readers should be aware that websites listed in this work may have changed or disappeared between when this work was written and when it is read. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

If you believe you've found a mistake in this book, please bring it to our attention by emailing our reader support team at wileysupport@wiley.com with the subject line "Possible Book Errata Submission."

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Control Number: 2023940016

Cover image: Courtesy of Greg Jones and Wiley
Cover design: Wiley

Dedicated to David and Terry

About the Author

Jordan Goldmeier is one of the leading global minds on data visualization and data science. His books include *Dashboards for Excel* (Apress), *Advanced Excel Essentials* (Apress), *Becoming a Data Head: How to Think, Speak, and Understand Data Science, Statistics, and Machine Learning* (Wiley). Jordan has received the prestigious Microsoft Most Valuable Professional Award many times over the years. He has consulted and provided training for Fortune 500 companies, NATO, and taught analytics for Wake Forest University. He runs multiple businesses as a digital nomad living in Lisbon, Portugal. You connect with Jordan on LinkedIn and on Instagram (@jordangoldmeier).

About the Technical Editors

Alex Gutman is a data scientist, corporate trainer, and Accredited Professional Statistician® who enjoys teaching a wide variety of data science topics to technical and nontechnical audiences. He's a former adjunct professor at the Air Force Institute of Technology and current adjunct at the University of Cincinnati. Alex is also the author of the book *Becoming a Data Head: How to Think, Speak, and Understand Data Science, Statistics, and Machine Learning* (Wiley). He received his BS and MS degrees in mathematics from Wright State University and his PhD in applied mathematics from the Air Force Institute of Technology.

Matthew Bernath is passionate about leveraging data to bolster economies and facilitate strategic dealmaking. Matthew has led the data analytics division of one of Africa's largest investment banks and is currently the head of data ecosystems for Africa's largest retailer. His diverse experience spans from structuring multibillion-rand project financing deals to utilizing data to uplift society, always driven by data-focused decision-making. Recognized as one of the "60 Data Changemakers to Know" by Narrative Science and a finalist for Data Analytics Leader of the Year in 2022, Matthew's achievements extend beyond his professional role. His contribution to community-building initiatives include hosting the Johannesburg Data Science and Financial Modelling meetup groups and the highly regarded *Financial Modelling Podcast*, which was awarded Financial Modelling Resource of the Year 2021. He also formerly hosted the RMB *Data Analytics* podcast. Prior to his investment banking and retail career, Matthew held leadership roles in various advisory and technology firms, bringing his data-driven approach to different industries.

Acknowledgments

Life has a weird way of coming full circle. I read the original *Data Smart* when it first came out in 2013. I had no imagination back then I would write the revised edition. Yet, here I am. If fate brought me to this place, it's because I love Excel. Therefore, it only makes sense to first acknowledge the Excel product team at Microsoft, who've managed to push Excel beyond the tool it was back in 2013.

As a Microsoft MVP, I've met some incredible folks at Microsoft over the years, who've really listened and understood the ways in which my community uses Excel. In particular, I would like to acknowledge David Gainer, Guy Lev, and Joe McDaid for continually expanding the product.

I would also like to acknowledge my peers in the Excel community who pushed the product beyond its limitations for the good of the whole. As it relates to the material in this book, I must mention George Mount, Oz du Soleil, Carlos Barboza, and Roberto Mensa for challenging the norm.

I also have to give major credit to the book's first author, John Foreman. If you weren't in the data space back in 2013, you should know it was a different world. In those days, people were enamored by the idea of "big data." Companies were rushing to implement technologies that could handle large datasets before they even had high-quality data.

But then there was John's book, which showed people how to do (or at the very least, teach) data science without big data technologies—you could just use Excel. John showed people that it wasn't about the technology, but, rather, one had to really think through the problem. And he did it without being boring. John's book served as major motivation and inspiration for my last book, *Becoming a Data Head*. It's a great honor to be working on this material.

I also have to acknowledge my technical editors, Alex Gutman and Mathew Bernath. Both are incredibly intelligent and esteemed in their fields. Alex and I wrote *Data Head* together, and it's amazing to once again have him on another project. Alex is thorough, humble, and deeply affable. He's often the smartest person in the room, but you would never know, as there's not an arrogant bone in his body. Alex's contributions are indelibly fused into the text of this book.

Mathew is perhaps the coolest data (and coffee) nerd I know. He knows his craft well and channels that knowledge into community building, bringing ideas and minds together to push the field forward. His technical advice on this book challenged many of the things I took for granted. This book is much better off for it, and I'm very grateful for his support.

I also want to acknowledge the team at Wiley. In particular, I would like to mention Jim Minatel who believed strongly in this project and really pushed to make it happen. I also want to thank John Sleeva, my development editor. John has my favorite working style—no news is good news. He’s always calm, thorough, and dependable. This is my second book with Wiley and Jim and John—and I couldn’t have asked for a better team.

I also have to mention Archana Pragash who worked tirelessly on proofing this book to my specifications. I often wondered when she slept. She always responded quickly—nights, weekends, etc. For a big project like this, Archana was a dependable pillar. The layout of this book is to her credit.

Finally, I would like to thank you, the reader. It’s your interest that makes this book happen. I hope you enjoy it.

Contents

Introduction.....	xix
1 Everything You Ever Needed to Know About Spreadsheets but Were Too Afraid to Ask	1
Some Sample Data	2
Accessing Quick Descriptive Statistics	3
Excel Tables.....	4
Filtering and Sorting	5
Table Formatting	7
Structured References	7
Adding Table Columns	10
Lookup Formulas.....	11
VLOOKUP.....	11
INDEX/MATCH.....	13
XLOOKUP.....	15
PivotTables	16
Using Array Formulas.....	19
Solving Stuff with Solver.....	20
2 Set It and Forget It: An Introduction to Power Query.....	27
What Is Power Query?	27
Sample Data	28
Starting Power Query	29
Filtering Rows.....	32
Removing Columns.....	33
Find & Replace	34
Close & Load to...Table.....	35

3	Naïve Bayes and the Incredible Lightness of Being an Idiot	39
	The World's Fastest Intro to Probability Theory	39
	Totaling Conditional Probabilities	40
	Joint Probability, the Chain Rule, and Independence	40
	What Happens in a Dependent Situation?	41
	Bayes Rule	42
	Separating the Signal and the Noise	43
	Using the Bayes Rule to Create an AI Model	44
	High-Level Class Probabilities Are Often Assumed to Be Equal	45
	A Couple More Odds and Ends	46
	Let's Get This Excel Party Started	47
	Cleaning the Data with Power Query	48
	Splitting on Spaces: Giving Each Word Its Due	50
	Counting Tokens and Calculating Probabilities	55
	We Have a Model! Let's Use It	58
4	Cluster Analysis Part 1: Using K-Means to Segment Your Customer Base	65
	Dances at Summer Camp	65
	Getting Real: K-Means Clustering Subscribers in Email Marketing	70
	The Initial Dataset	71
	Determining What to Measure	72
	Start with Four Clusters	75
	Euclidean Distance: Measuring Distances as the Crow Flies	76
	Solving for the Cluster Centers	80
	Making Sense of the Results	82
	Getting the Top Deals by Cluster	83
	The Silhouette: A Good Way to Let Different K Values Duke It Out	86
	How About Five Clusters?	95
	Solving for Five Clusters	96
	Getting the Top Deals for All Five Clusters	96
	Computing the Silhouette for 5-Means Clustering	99
	K-Medians Clustering and Asymmetric Distance Measurements	100
	Using K-Medians Clustering	100
	Getting a More Appropriate Distance Metric	100

Putting It All in Excel	102
The Top Deals for the 5-Medians Clusters	104
5 Cluster Analysis Part II: Network Graphs and Community Detection	109
What Is a Network Graph?	110
Visualizing a Simple Graph	110
Beyond GiGraph and Adjacency Lists	115
Building a Graph from the Wholesale Wine Data	117
Creating a Cosine Similarity Matrix	118
Producing an R-Neighborhood Graph	121
Introduction to Gephi	123
Creating a Static Adjacency Matrix	124
Bringing in Your R-Neighborhood Adjacency Matrix into Gephi	124
Node Degree	128
Touching the Graph Data	130
How Much Is an Edge Worth? Points and Penalties in Graph Modularity	132
What's a Point, and What's a Penalty?	133
Setting Up the Score Sheet	136
Let's Get Clustering!	138
Split Number 1	138
Split 2: Electric Boogaloo	143
And... Split3: Split with a Vengeance	145
Encoding and Analyzing the Communities	146
There and Back Again: A Gephi Tale	151
6 Regression: The Granddaddy of Supervised Artificial Intelligence	157
Predicting Pregnant Customers at RetailMart Using Linear Regression	158
The Feature Set	159
Assembling the Training Data	161
Creating Dummy Variables	163
Let's Bake Our Own Linear Regression	165
Linear Regression Statistics: R-Squared, F-Tests, t-Tests	173
Making Predictions on Some New Data and Measuring Performance	182

Predicting Pregnant Customers at RetailMart Using Logistic Regression	192
First You Need a Link Function.	192
Hooking Up the Logistic Function and Reoptimizing	193
Baking an Actual Logistic Regression	196
7 Ensemble Models: A Whole Lot of Bad Pizza.	203
Getting Started Using the Data from Chapter 6	203
Bagging: Randomize, Train, Repeat.	204
Decision Stump is Another Name for a Weak Learner	204
Doesn't Seem So Weak to Me!	204
You Need More Power!	207
Let's Train It	208
Evaluating the Bagged Model	220
Boosting: If You Get It Wrong, Just Boost and Try Again.	223
Training the Model—Every Feature Gets a Shot.	224
Evaluating the Boosted Model	231
8 Forecasting: Breathe Easy: You Can't Win.	235
The Sword Trade Is Hopping	236
Getting Acquainted with Time-Series Data.	236
Starting Slow with Simple Exponential Smoothing	238
Setting Up the Simple Exponential Smoothing Forecast	240
You Might Have a Trend	249
Holt's Trend-Corrected Exponential Smoothing	250
Setting Up Holt's Trend-Corrected Smoothing in a Spreadsheet	252
So Are You Done? Looking at Autocorrelations	258
Multiplicative Holt-Winters Exponential Smoothing	266
Setting the Initial Values for Level, Trend, and Seasonality	268
Getting Rolling on the Forecast	274
And. . . Optimize!	280
Putting a Prediction Interval Around the Forecast	283
Creating a Fan Chart for Effect	287
Forecast Sheets in Excel	289

9	Optimization Modeling: Because That “Fresh-Squeezed” Orange Juice Ain’t Gonna Blend Itself	293
	Wait. . . Is This Data Science?	294
	Starting with a Simple Trade-Off	295
	Representing the Problem as a Polytope.	296
	Solving by Sliding the Level Set	297
	The Simplex Method: Rooting Around the Corners.	298
	Working in Excel	300
	Fresh from the Grove to Your Glass. . . with a Pit Stop Through a Blending Model.	305
	Let’s Start with Some Specs	307
	Coming Back to Consistency.	308
	Putting the Data into Excel	309
	Setting Up the Problem in Solver	311
	Lowering Your Standards.	314
	Dead Squirrel Removal: the Minimax Formulation	317
	If-Then and the “Big M” Constraint.	320
	Multiplying Variables: Cranking Up the Volume to 11,000.	324
	Modeling Risk.	330
	Normally Distributed Data	331
10	Outlier Detection: Just Because They’re Odd Doesn’t Mean They’re Unimportant	339
	Outliers Are (Bad?) People, Too.	340
	The Fascinating Case of <i>Hadlum v. Hadlum</i>	340
	Tukey’s Fences	341
	Applying Tukey’s Fences in a Spreadsheet.	342
	The Limitations of This Simple Approach	345
	Terrible at Nothing, Bad at Everything	346
	Preparing Data for Graphing	347
	Creating a Graph.	350
	Getting the k-Nearest Neighbors	351
	Graph Outlier Detection Method 1: Just Use the Indegree	352
	Graph Outlier Detection Method 2: Getting Nuanced with k-Distance	355
	Graph Outlier Detection Method 3: Local Outlier Factors Are Where It’s At	358

11	Moving on From Spreadsheets	363
	Getting Up and Running with R	364
	A Crash Course in R-ing	366
	Show Me the Numbers! Vector Math and Factoring	367
	The Best Data Type of Them All: the Dataframe	370
	How to Ask for Help in R	371
	It Gets Even Better...Beyond Base R	372
	Doing Some Actual Data Science	374
	Reading Data into R	374
	Spherical K-Means on Wine Data in Just a Few Lines	375
	Building AI Models on the Pregnancy Data	381
	Forecasting in R	389
	Looking at Outlier Detection	393
12	Conclusion	397
	Where Am I? What Just Happened?	397
	Before You Go-Go	397
	Get to Know the Problem	398
	We Need More Translators	398
	Beware the Three-Headed Geek-Monster: Tools, Performance, and Mathematical Perfection	399
	You Are Not the Most Important Function of Your Organization	401
	Get Creative and Keep in Touch!	402
	Index	403

Introduction

What Am I Doing Here?

If you're reading this book, it's because on some level you understand the importance of both data and data science in your business and career.

The original *Data Smart* was written more than a decade ago. John Foreman, the first book's author, exposed a new generation of readers to the supposed magic behind the curtain of data science. John proved that data science didn't have to be so mysterious. You could both understand and do data science in something as humble as the spreadsheet.

John's words severed as a prescient warning for what would come. He noted the "buzz about data science," and the pressure it created on businesses to take on data science projects and hire data scientists without even fully understanding why.

The truth is most people are going about data science all wrong. They're starting with buying the tools and hiring the consultants. They're spending all their money before they even know what they want, because a purchase order seems to pass for actual progress in many companies these days.

John's words still ring true today. Ten years after the first wave of interest in data science, the data science machine is still working in full force, churning out ideas faster than we can articulate the opportunities and challenges they present to business and society. In my last book, *Becoming a Data Head: How to Think, Speak and Understand Data Science, Statistics and Machine Learning* (Wiley, New York, NY, 2021), my coauthor and I called this the *data science industrial complex*.

To put it bluntly, despite the extensive interest in data and data science, projects still fail sometimes at alarming rates, even as data is supposed to be fact driven. In truth, as much as 87 percent data science projects won't make it into production.¹

What is and isn't a "data disaster" is perhaps up from some considerable debate. But it's fair to say the recent past is filled with examples in which technology, data, and the like were hailed as something magical before they ultimately came up short. Here are just a few examples worth considering:

¹"Why do 87% of data science projects never make it into production?" <https://designingforanalytics.com/resources/failure-rates-for-analytics-bi-iot-and-big-data-projects-85-yikes>

- An attorney used a generative AI chatbot for legal research, submitting a brief to the court with cases that did not exist, but perhaps sounded plausible.²
- The COVID-19 pandemic exposed major issues in forecasting across the board, from supply chain issues to understanding the spread of the virus.³
- When the original *Data Smart* came out, accurately predicting the outcome of the US presidential election seemed like an easy feat. In 2016, however, model after model inaccurately predicted a win for Hillary Clinton, despite increased money, time, and effort into the subject.⁴

Most data science projects and outcomes don't fail so spectacularly. Instead, data science projects die slow deaths, while management pours money and resources into chasing elusive numbers they don't entirely understand.

Yet, some of the greatest data achievements did not come from any particular technology. Rather, they came from human ingenuity. For instance, I used to lead projects for a nonprofit called DataKind, which leverages “data science and AI in service of Humanity.”

DataKind uses teams of volunteer data scientists to help mission-driven organizations design solutions to tough social problems in an ethical and socially responsible way. When I was there, we worked with major organizations like the United Nations and Habitat for Humanity.

Volunteers built all sorts of models and tools, from forecasting water demand in California to using satellite imagery to identify villages in need with machine learning. The work we did had impact, so it's not all doom and gloom. When you're done with this book, you might consider giving back in your own way.⁵ Remember: Humans solve problems not machines.

What Is Data Science?

In my last book, *Becoming a Data Head*, Alex Gutman (my coauthor) and I actually don't define data science. One reason is that the space is too hard to pin down. And we didn't want folks to get caught up in trying to justify whether or not they were data scientists. In the original *Data Smart*, John Foreman offers this working definition:

Data science is the transformation of data using mathematics and statistics into valuable insights, decisions, and products.

²“8 famous analytics and AI disasters.” [www.cio.com/article/190888/](http://www.cio.com/article/190888/5-famous-analytics-and-ai-disasters.html)

³“Forecasting for COVID-19 has failed.” www.ncbi.nlm.nih.gov/pmc/articles/PMC7447267

⁴“The Real Story Of 2016.” <https://fivethirtyeight.com/features/the-real-story-of-2016>

⁵To see the impact DataKind has had, take a look at their case studies - www.datakind.org/what-we-do

John takes a broad, business-centric view. He's quick to note it's a "catchall buzzword for [everything] analytics today." Ten years later, I and the rest of the industry are still struggling to define exactly what data science is and isn't. So rather than proffer a definition as if that will get us closer to the truth, I'd rather describe what a data scientist does.

- *Data scientists identify relevant questions that can be solved with data.* This may sound obvious, but many questions can't be solved with data and technology. A good data scientist can tease out the problems in which algorithms and analyses make the most sense.
- *Data scientists extract meaningful patterns and insights from data.* Anyone can eyeball a set of numbers and draw their own conclusions. On the other hand, data scientists focus on what can be said statistically and verifiably. They separate speculation from science, focusing instead on what the data says.
- Finally, *data scientists convey results using data visualization and clear communication.* In many cases, a data scientist will have to explain how an algorithm works and what it does. Historically, this has been a challenge for many in the field. But a recent crop of books (like this one) aims at giving data scientists a way to explain how they came to their results without being too stuck into the weeds.

Incredibly, some of the techniques mentioned in the following pages are as old as World War II. They were invented at the dawn of the modern computer, long before you could easily spin up a new instance of R. The hype machine won't tell you these "new" algorithms were first developed on punch cards.

And some of the techniques in this book were invented recently, taking advantage of the wealth of data, self-service analytics, cloud computers, and new graphical processing units developed in the last 10 years.

Again, we're reminded that human ingenuity is what drives this field forward.

Age has no bearing on difficulty or usefulness. All these techniques whether or not they're currently the rage are equally useful in the right business context. It's up to you to use them correctly. That's why you need to understand how they work, how to choose the right technique for the right problem, and how to prototype with them.

Do Data Scientists Actually Use Excel?

Many (but not all) veteran data scientists will tell you they loathe spreadsheets and Excel in particular. They will say that Excel isn't the best place to create a data science model. To some extent, they're right.

But before you throw this book away, let's understand why they say this. You see, there was a time before R and before Python. It was a time when MATLAB and SPSS reigned supreme. The latter tools were expensive and often required a computer with some major

horsepower to run a model. Moreover, the files that these tools generated were not easily distributable. And, in a secure corporate or institutional environment, sending files with code in them over email would trip the unsafe-email alarms.

As a result, many in the industry began building their work in Excel. This was particularly true of models that helped support executive decision-making. Excel was the secret way around these email systems. It was a way to build a mini data application without having to get approval from the security team.

Many executive teams relied on Excel. Unfortunately, this also created a myopic view among executives who didn't really understand data science. For them, Excel was the only place to do this type of work. It was where they were most comfortable.

They knew the product. They could see what the analyst created. And the analyst could walk them through each step. In fact, that's why we're using Excel in this book.

But Excel (at the time) was limited. Limited by how much it could process at any moment. Limited by the amount of data it could store. The macro language behind Excel, Visual Basic for Applications (VBA), is still hailed by many executives as an advanced feature. But VBA is based on Visual Basic 6.0, which was deprecated in 1999. The Excel version of this language has received only the barest of updates. When today's data scientists point out that VBA can't do what R or Python can, it's hard to disagree.

On the flipside, however, Microsoft has paid attention over the last few years. The Excel product team has come to understand how data scientists use their tool. They've poured more research into some very specific use cases. For instance, we'll talk about an entirely new data wrangling tool in Excel called Power Query. Power Query can do the same data wrangling tasks as in Python and R, often more quickly. And we'll talk about new Excel functions that make data science in Excel a whole lot easier. Today, there is renewed interest in using Excel for data science problems beyond what was possible only a few years ago.

But if there's a place where Excel shines, it's in explaining and understanding data science concepts. Before getting a "yes" to your new data science project, you'll need to get buy-in from management. You can fire up an advanced algorithm in R, pull out lines of code, and explain what each function does step-by-step. Or you can walk management through the algorithm in Excel and even give them the ability to filter results and ask questions of the data.

In fact, Excel is great for prototyping. You're not running a production AI model for your online retail business out of Excel, but that doesn't mean you can't look at purchase data, experiment with features that predict product interest, and prototype a targeting model.

At the end of this book, I'll show you how to implement what we've built in Excel in R. In fact, this follows my own path in building data science tools for companies. First, I would lay out my ideas in Excel. Use the spreadsheet as a way to validate my ideas and make sure I understand exactly what the algorithms do. Then, usually, when I'm ready, I move it to R or Python.

But sometimes I don't. Because in some instances Excel just gets the job done, and the problem doesn't need more complication. As you will see, knowing how to do these techniques in Excel will give you a major advantage, whether or not you end up implementing them in something more powerful.

Conventions

To help you get the most from the text and keep track of what's happening, I've used a number of conventions throughout the book.

Frequently in this text I'll reference little snippets of Excel code like this:

```
=IF(A1 = 1, "I love Excel", "I REALLY love Excel")
```

SIDEBARS

Sidebars touch upon some side issue related to the text in detail.

WARNING

Warnings hold important, not-to-be-forgotten information that is directly relevant to the surrounding text.

NOTE

Notes cover tips, hints, tricks, or asides to the current discussion.

- We bold technical objects, when introducing them for the first time, or when it makes sense to set them off. We also use the bold font to refer to specific fields and buttons.
- We *italicize* new concepts and important words when we introduce them.
- We show filenames, URLs, and formulas within the text like so: `www.linkedin.com/in/jordangoldmeier`

Let's Get Going

A new generation of data scientists is learning how to implement work that was only theoretical when I first started. The industry is undergoing a serious reflection on what's

important. Businesses are starting to realize their most important assets aren't data, algorithms, or technology—it's people. People just like you.

As you go along your data journey, you will likely encounter more than your fair share of bad decision-making, a lack of critical thinking, ignorant management, and even some imposter syndrome. Sadly, as with many of the data successes, these are part of the legacy. But with the knowledge contained herein, you'll be set up for success. You'll understand the algorithms. You'll know how and what they do. And, you won't be fooled by buzzwords. When it comes to doing real data science work, you'll already know how to identify the data science opportunities within your own organization.

By reading this book, you're going to have a leg up on the next generation of data problems. Whether you're a veteran of the field or a student in school, by reading this book, you will become a better data scientist.

In Chapter 1, "Everything You Ever Needed to Know About Spreadsheets but Were Too Afraid to Ask," I'm going to fill in a few holes in your Excel knowledge. And, in Chapter 2, "Set & Forget it! An introduction to Power Query," I'm going to show you Power Query. After that, you'll move right into use cases. By the end of this book, you'll have experience implementing from scratch the following techniques:

- Optimization using linear and integer programming.
- Working with time-series data, detecting trends, and seasonal patterns, and forecasting with exponential smoothing.
- Using Monte Carlo simulation in optimization and forecasting scenarios to quantify and address risk.
- Applying Artificial intelligence using the general linear model, logistic link functions, ensemble methods, and naïve Bayes.
- Measuring distances between customers using cosine similarity, creating kNN graphs, calculating modularity, and clustering customers.
- Detecting outliers in a single dimension with Tukey fences or in multiple dimensions with local outlier factors.
- Using R packages to implement data science techniques quickly.

It's now time for our journey to begin. I'll see you in the next chapter!

1

Everything You Ever Needed to Know About Spreadsheets but Were Too Afraid to Ask

This book assumes you have some experience working with spreadsheets. You won't need to be a spreadsheet expert, but if this is your first-time opening Excel, you might find this chapter a bit challenging. If that's you, I would recommend pairing this chapter with a *For Dummies* book or a beginner-level online class.

Even so, what follows in this chapter might still surprise the most seasoned, self-professed Excel pros. So, regardless of your Excel experience, this chapter should not be skipped! In the following pages, we'll describe a wide variety of Excel features that we'll use throughout the book.

Before moving forward, let's talk about the different versions of Excel out there and how they might affect you. First, everything in this book will work seamlessly in Excel 365 and Excel 2016 and beyond for Windows.

This book is going to use Excel 365 desktop for Windows. Excel 365 generally represents the latest versions of Excel, to which Microsoft pushes monthly updates. Some institutions still use enterprise versions of Excel such as Excel 2016 and Excel 2019. These versions will work, too. To ensure you are using the latest version of Excel, call the IT department at your school or your office and let them know you'd like to get the latest build. They'll know what you mean.

The story for Mac is a bit different. If you're on a Mac, some of keystrokes will be different. There are different icons and button locations. Power Query, Excel's data wrangling powerhouse, has fewer features in the Mac version as of this writing. Still, you should be able to get by just fine.

This book requires a desktop version of Excel. Though you can work in Excel through their online platform and through SharePoint, neither of these environments is suitable for this book as of this writing. That may change in time, but for now, assume everything from here on out is for Excel on the desktop.

Some Sample Data

NOTE

The Excel workbook used in this chapter, `Concessions.xlsx`, is available for download at the book's website at www.wiley.com/go/datasmart2e.

Let's start with some sample data.

You don't know this about me, but I love hot dogs. (Seriously, I have a Chicago-style hot dog tattoo.) A dream of mine is to one day run a hot dog stand. Let's say that dream happens, and I open up a concession stand to serve the sporting events of a local high school. If you've already opened `Concessions.xlsx`, let's start on the first tab, Basketball Game Sales.

At the end of each night, the point-of-sale system spits out the day's takings. It looks like in Figure 1.1.

	A	B	C	D	E
1	Item	Category	Price	Profit	Actual Profit
2	Beer	Beverages	\$ 4.00	50%	\$ 2.00
3	Hamburger	Hot Food	\$ 3.00	67%	\$ 2.00
4	Popcorn	Hot Food	\$ 5.00	80%	\$ 4.00
5	Pizza	Hot Food	\$ 2.00	25%	\$ 0.50
6	Bottled Water	Beverages	\$ 3.00	83%	\$ 2.50
7	Hot Dog	Hot Food	\$ 1.50	67%	\$ 1.00
8	Chocolate Dipped Cone	Frozen Treat	\$ 3.00	50%	\$ 1.50
9	Soda	Beverages	\$ 2.50	80%	\$ 2.00
10	Chocolate Bar	Candy	\$ 2.00	75%	\$ 1.50
11	Hamburger	Hot Food	\$ 3.00	67%	\$ 2.00
12	Beer	Beverages	\$ 4.00	50%	\$ 2.00
13	Hot Dog	Hot Food	\$ 1.50	67%	\$ 1.00
14	Licorice Rope	Candy	\$ 2.00	50%	\$ 1.00
15	Chocolate Dipped Cone	Frozen Treat	\$ 3.00	50%	\$ 1.50
16	Nachos	Hot Food	\$ 3.00	50%	\$ 1.50
17	Pizza	Hot Food	\$ 2.00	25%	\$ 0.50
18	Beer	Beverages	\$ 4.00	50%	\$ 2.00
19	Soda	Beverages	\$ 2.50	80%	\$ 2.00
20	Beer	Beverages	\$ 4.00	50%	\$ 2.00

Figure 1.1: Concession stand sales

This data is laid out in *tabular format*. This is likely something you're very familiar with. In Excel it's made up of *rows*, *columns*, and *cells*.

Some areas of data science may call these by different names. For instance, a row might be called a *record*, *observation*, or *tuple*. A column might be called a *field*, *feature*,

or *dimension*. In truth, it doesn't matter what you call them so long as you use them well. However, you should take note that those around you might use different terms depending upon their field.

Accessing Quick Descriptive Statistics

Excel has the ability to instantly provide summary statistics—such as average, sum, min, and max—in the status bar. However, most of these measures aren't enabled by default. You'll likely want to refer to these continuously along your data journey.

To see what I mean, select cell E2 and then press Ctrl+Shift+Down (⌘+Shift+Down on a Mac). This will automatically highlight the data region of the entire column, spanning from E2:E200.

TIP

If you're the type who loves keyboard shortcuts, my friend, David Bruns, has put together a very handy list of shortcuts for both Mac and PC on his website, Excel Jet. See <https://exceljet.net/shortcuts>.

Look at the lower-right portion of your status bar. It should show an average of \$1.79. Right-click the average label in your status bar, and you'll see multiple measures you can select (see Figure 1.2). Go ahead and select Average, Count, Numerical Count, Minimum, Maximum, and Sum. Once complete, you'll see they're now all available in the status bar. You'll appreciate having these measures at a moment's glance.

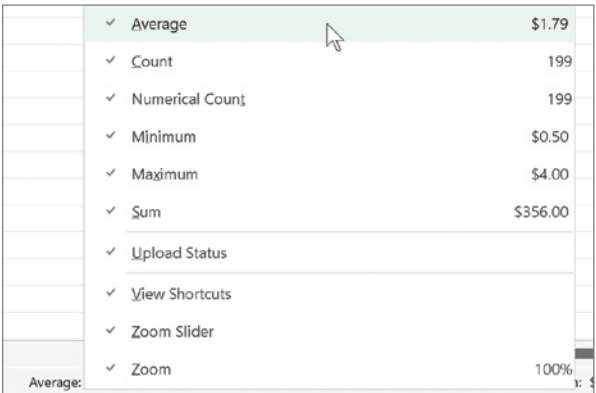


Figure 1.2: When you right-click the status bar, you have the option to have additional descriptive statistics reported to you. Select all of them.

Excel Tables

Perhaps you have experience with Excel formulas. You know, for instance, we could place a formula like `=AVERAGE(E2:E200)` in a blank cell to find the average.

Unfortunately, the cell reference (E2:E200) inside the `AVERAGE` formula is a bit of problem for the data scientist. What if we want to add 50 records? We would have to remember to update the `AVERAGE` formula to the new cell address of E2:E250. What if we moved the data from column E to column F? We would *again* have to ensure the `AVERAGE` formula pulls from F2:F250. And when you think about it, what does E2:E200 or F2:F250 really tell us about the data it represents?

You may have accepted that clunky formulas and misaligned references are just part of Excel. But I'm here to tell you there's a better way. Excel tables were created to meet the challenges described.

To apply an Excel table, place the cursor anywhere inside the data region. On the Insert tab, click Table (see Figure 1.3).

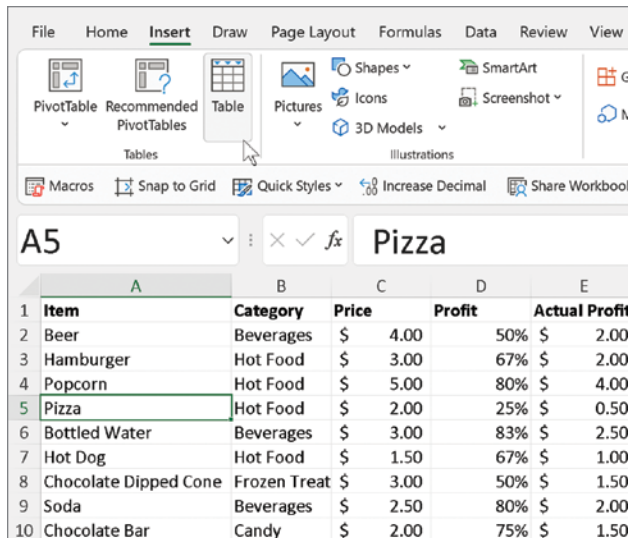


Figure 1.3: To insert an Excel table, place your cursor anywhere in the table region. Then, on the Insert tab, click the Table button.

In the Create Table dialog box, ensure that My Table Has Headers is selected, and then click OK.

You have now applied an Excel table. Your screen should look like in Figure 1.4.

The screenshot shows an Excel spreadsheet with a table named 'Sales'. The table has five columns: Item, Category, Price, Profit, and Actual Profit. The data is as follows:

Item	Category	Price	Profit	Actual Profit
Beer	Beverages	\$ 4.00	50%	\$ 2.00
Hamburger	Hot Food	\$ 3.00	67%	\$ 2.00
Popcorn	Hot Food	\$ 5.00	80%	\$ 4.00
Pizza	Hot Food	\$ 2.00	25%	\$ 0.50
Bottled Water	Beverages	\$ 3.00	83%	\$ 2.50
Hot Dog	Hot Food	\$ 1.50	67%	\$ 1.00
Chocolate Dipped Cone	Frozen Treat	\$ 3.00	50%	\$ 1.50
Soda	Beverages	\$ 2.50	80%	\$ 2.00
Chocolate Bar	Candy	\$ 2.00	75%	\$ 1.50
Hamburger	Hot Food	\$ 3.00	67%	\$ 2.00
Beer	Beverages	\$ 4.00	50%	\$ 2.00
Hot Dog	Hot Food	\$ 1.50	67%	\$ 1.00
Licorice Rope	Candy	\$ 2.00	50%	\$ 1.00
Chocolate Dipped Cone	Frozen Treat	\$ 3.00	50%	\$ 1.50

Figure 1.4: Concession stand data with an Excel table applied

Whenever you create a new table, the Table Design ribbon tab appears, allowing you to interact with the table. As a first step for working with Excel tables, give the table a good name.

Excel will attempt to name the table for you with names like Table1, Table2, and mysteriously, Table1_2. You should never accept these default names (it's tacky!), but instead set a proper name reflecting the underlying dataset.

In the upper-left corner of the Table Design tab you can set the table's name. In Figure 1.4, I've set it to Sales. You'll quickly see why this is important.

Tables provide tons of features, akin to the data frames of Python and R that make doing data science in Excel that much easier. For one, as you scroll down an Excel table, the normal alphabetical column headers are replaced with the table's fields. This allows you to work with a table and know which column you're working with without freezing the top row. Take a look at Figure 1.5.

Filtering and Sorting

Tables have filtering and sorting already baked in (no need to apply the filtering feature on the Home or Data tab). For instance, if I want to simply look at the sales of hot dogs, I can filter the Item column by clicking the drop-down button in the header and selecting the item of interest (see Figure 1.6).

6 Data Smart

	Item	Category	Price	Profit	Actual Profit
2	Beer	Beverages	\$ 4.00	50%	\$ 2.00
3	Hamburger	Hot Food	\$ 3.00	67%	\$ 2.00
4	Popcorn	Hot Food	\$ 5.00	80%	\$ 4.00
5	Pizza	Hot Food	\$ 2.00	25%	\$ 0.50
6	Bottled Water	Beverages	\$ 3.00	83%	\$ 2.50
7	Hot Dog	Hot Food	\$ 1.50	67%	\$ 1.00
8	Chocolate Dipped Cone	Frozen Treat	\$ 3.00	50%	\$ 1.50
9	Soda	Beverages	\$ 2.50	80%	\$ 2.00
10	Chocolate Bar	Candy	\$ 2.00	75%	\$ 1.50
11	Hamburger	Hot Food	\$ 3.00	67%	\$ 2.00
12	Beer	Beverages	\$ 4.00	50%	\$ 2.00
13	Hot Dog	Hot Food	\$ 1.50	67%	\$ 1.00
14	Licorice Rope	Candy	\$ 2.00	50%	\$ 1.00
15	Chocolate Dipped Cone	Frozen Treat	\$ 3.00	50%	\$ 1.50

Figure 1.5: Tables will replace the column headers with the column names. This means you can work with the table without having to freeze the header row.

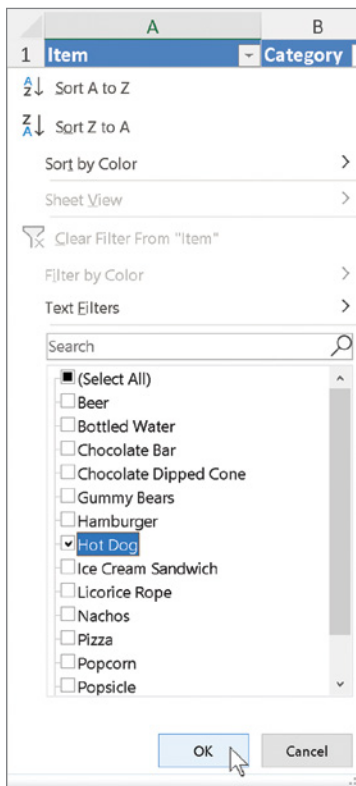


Figure 1.6: Tables already have filters built-in. To filter a specific column, press the gray drop-down button next to the column header.

To clear the filter, again, simply click the drop-down in the header of Items and select Clear Filter From Item.

Before moving on, take a look at the different options available in the drop-downs. Note that there are many ways to sort and filter your data.

Table Formatting

Excel's default formatting of tables is hideous. (Sorry, that's just how I feel.) I can't abide by the tacky overcolored defaults. For ease of reading the data in your tables, my recommendation is to use the Table style menu and select a table style from the Light category that does not include banded rows (see Figure 1.7).

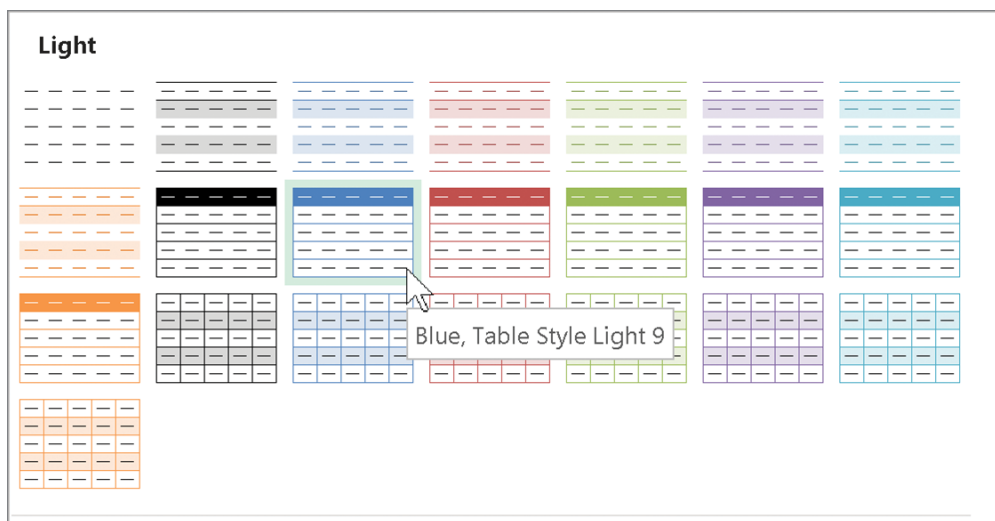


Figure 1.7: The default table formatting is overly colorful and distracting. However, the lighter styles will make your data easy to read and comprehend.

Going forward, I will always adjust the table style to one of the easier-to-read options even when I haven't directed you to do so.

Structured References

Structured references are the single most important feature of Excel tables. Remember, at the start of the chapter, we focused on the issues of using cell references in formulas. Let's see how Excel tables switch things up.

In cell H2 on the Basketball Game Sales tab, I've created a label called Average Profit. To the right of it, in cell I2, I'll write my Average formula. You can set this up like in Figure 1.8.

Item	Category	Price	Profit	Actual Profit
Beer	Beverages	\$ 4.00	50%	\$ 2.00
Hamburger	Hot Food	\$ 3.00	67%	\$ 2.00
Popcorn	Hot Food	\$ 5.00	80%	\$ 4.00
Pizza	Hot Food	\$ 2.00	25%	\$ 0.50
Bottled Water	Beverages	\$ 3.00	83%	\$ 2.50
Hot Dog	Hot Food	\$ 1.50	67%	\$ 1.00
Chocolate Dipped Cone	Frozen Treat	\$ 3.00	50%	\$ 1.50

Figure 1.11: When referring to a table's column field in Excel, you will see the selected column highlighted like a cell reference.

Once you're happy with the formula, press Enter. The average profit spend is 1.79ish. To format this cell to a dollar amount, you can click the dollar sign icon (\$) on the Home ribbon tab in the Number group to turn it into a two-decimal dollar amount.

Now, I want to draw your attention to the magic of tables and structured references. Structured references can grow and shrink based on how much data is contained in the table without having to adjust the formulas that use them. Let's see this in action.

Scroll all the way down to the bottom of the table and place your cell in the leftmost cell at the bottom of the table. In this case, that's cell A201 (see Figure 1.12).

	A	B	C	D	E
199	Bottled Water	Beverages	\$ 3.00	83%	\$ 2.50
200	Popsicle	Frozen Treat	\$ 3.00	83%	\$ 2.50
201					

Figure 1.12: To automatically add information to an Excel table, place your cursor in the cells directly under the last record and add your new data.

Let's add a new record in cell A201 by typing Popsicle. Then, press Enter. Note that the table has now grown to consume this new record. In the Profit field, add a large dollar amount like \$2000. Your table should now look like in Figure 1.13.

	Item	Category	Price	Profit	Actual Profit
190	Popsicle	Frozen Treat	\$ 3.00	83%	\$ 2.50
191	Gummy Bears	Candy	\$ 2.00	50%	\$ 1.00
192	Pizza	Hot Food	\$ 2.00	25%	\$ 0.50
193	Bottled Water	Beverages	\$ 3.00	83%	\$ 2.50
194	Popcorn	Hot Food	\$ 5.00	80%	\$ 4.00
195	Beer	Beverages	\$ 4.00	50%	\$ 2.00
196	Pizza	Hot Food	\$ 2.00	25%	\$ 0.50
197	Popsicle	Frozen Treat	\$ 3.00	83%	\$ 2.50
198	Chocolate Bar	Candy	\$ 2.00	75%	\$ 1.50
199	Bottled Water	Beverages	\$ 3.00	83%	\$ 2.50
200	Popsicle	Frozen Treat	\$ 3.00	83%	\$ 2.50
201	Popsicle				\$2,000

Figure 1.13: When you add new data to the bottom of a table, it will automatically grow to consume the new information. However, you won't need to change any of the formulas that refer to it.

Don't worry if the dollar amount is so large as to be an outlier; we'll end up deleting this record in a second. I simply want to demonstrate that the average actual profit in cell I2 has changed dramatically to 11.78ish! The new data has become part of the table. And the formulas used on it do not need to be changed as data is added or deleted. The best part is that you don't need to type these new records—you can simply copy and paste in new data to the bottom of the table as it becomes available.

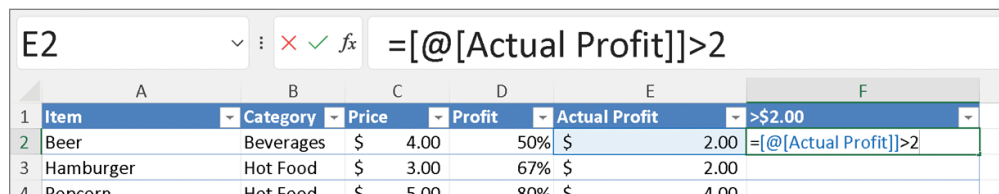
To delete the record, simply right-click the row label, 201, and select Delete. Do this and then double-check that I2 has changed back to its previous amount.

Adding Table Columns

In addition to being able to add records easily to the bottom of the table, you can also add custom column fields directly to the right of the table. Say, for instance, you would like to see which items generated more than \$2 in actual profit.

In cell F1 (the topmost cell adjacent to the right of the table), type >\$2.00. This will be your new column header. See how the table automatically grows to incorporate the new header! The best part: this new header will also appear in the IntelliSense drop-down.

Now, in cell F2, type = and then select the cell directly to the left of it. When you do this, Excel automatically fills in the formula with the structured reference `[@[Actual Profit]]`. Complete the formula `=[@[Actual Profit]]>2`, like in Figure 1.14. Press Enter.



	A	B	C	D	E	F
1	Item	Category	Price	Profit	Actual Profit	>\$2.00
2	Beer	Beverages	\$ 4.00	50%	\$ 2.00	=[@[Actual Profit]]>2
3	Hamburger	Hot Food	\$ 3.00	67%	\$ 2.00	
4	Pancake	Hot Food	\$ 5.00	80%	\$ 4.00	

Figure 1.14: Structured references allow you to create column calculations that work on data as records and not cell references.

Presto! Note what happens—the formula automatically populates the entire length of the table. No dragging necessary!

I know what you're thinking: *what's with the @?* This symbol means “at this row.” In other words, rather than working on the entire column range (which we did when calculating the average), the @ symbol directs the table formula to work only on the cells in that row. This makes sense when you think about it: each record contains a particular piece of information, and you might want to know information only with respect to the fields of that record.

To delete this column, simply right-click the column header of Column F and select Delete.

Excel tables are a wonderful tool in Excel, and as you can see, they fix many of the issues described at the start of this section. For the rest of this book, we'll work with a mix of cell references and the structured references of tables. We'll choose the one that makes the most sense given the situation. As we go through the book, you'll see how best to use each type.

TIP

I would love to further extol Excel tables, but unfortunately there's a page limit from the publisher. If you'd like to master Excel tables, see these books:

- *Excel Tables: A Complete Guide for Creating, Using and Automating Lists and Tables* by Zack Barresse and Kevin Jones (Holy Macro Books, 2014).
- *Excel 2019 Bible* by Michael Alexander, Richard Kusleika, and John Walkenbach (Wiley, 2018)

Lookup Formulas

Lookup formulas are very important in data analysis. They allow us to find out information about one or more records as well as join information from different tables.

There are multiple lookup functions in Excel. However, the ones we're most concerned with are `VLOOKUP`, `INDEX/MATCH`, and the venerable `XLOOKUP`. To understand each function, let's understand why they exist and the workflows they are designed to solve.

VLOOKUP

Let's start with `VLOOKUP`. `VLOOKUPS` exist because of a specific data workflow. For instance, sometimes you will receive data that looks like in Figure 1.15.

Key	Value1	Value2	Value3
4524
2352
1834
3271

Figure 1.15: A table layout where the unique identifier the “key” is the leftmost column

The leftmost column is the *key*. A key column contains a unique identification for each record. Keys can be anything from a simple record number to a Social Security number, student ID, or SKU. As you'll see, `VLOOKUP` requires a table of this layout.

The breakdown of `VLOOKUP` looks like this:

`VLOOKUP`(what you're looking for, the entire table of where you're looking, the column(s) of interest, whether to perform an exact or approximate match)

Let's take a look at each of these:

- **what you're looking for:** *This is your search term. In the case of `VLOOKUP`, it'll be the specific key you're looking for.*
- **The entire table of where you're looking:** *This is a reference to the entire table where you're looking, including the key column.*
- **the column(s) of interest:** *This is where you identify the columns you'd like to return.*
- **whether to perform an exact or approximate match:** *`VLOOKUP` allows you to either match the key exactly or approximately match it. In this book, we won't be dealing with approximate match. So, whenever you use `VLOOKUP` from examples in this book, make sure the last parameter is always a zero or a `FALSE` (which mean the same thing in Excel). This tells Excel we always want exact matching.*

`VLOOKUP` works by matching the search term to the key along the table's left side. Once the match is found, the key effectively "unlocks" the record. You can tell `VLOOKUP` what to return from the record by specifying the desired return column numbers in the column of interest parameter.

Enough talk—let's see it in Excel. In the Concessions file, select the Calories Worksheet tab. Note there is already a table there named `NutritionInfo`. In Figure 1.16, I've implemented the following formula in cell B19:

```
=VLOOKUP("Hot Dog", NutritionInfo, 2, FALSE)
```

As you can see, `Hot Dog` is the unique key. Once the `VLOOKUP` finds the key in the table, it returns the value associated with the column number provided. In this case, the 2 passed into the third parameter of the formula indicates we want the information from the second column.

	A	B	C	D	E	F
1	Item	Calories				
2	Beer	200				
3	Bottled Water	0				
4	Chocolate Bar	255				
5	Chocolate Dipped Cone	300				
6	Gummy Bears	300				
7	Hamburger	320				
8	Hot Dog	265				
9	Ice Cream Sandwich	240				
10	Licorice Rope	280				
11	Nachos	560				
12	Pizza	480				
13	Popcorn	500				
14	Popsicle	150				
15	Soda	120				
16						
17						
18	How many calories are in a Hot Dog?					
19	VLOOKUP	265 =VLOOKUP("Hot Dog", NutritionInfo, 2,FALSE)				

Figure 1.16: A VLOOKUP has been implemented in cell B19. In this case, the formula looks for the unique record associated with Hot Dog and returns information in the cell (the second column) to its right.

INDEX/MATCH

VLOOKUP solves one type of workflow for a specific data layout. But sometimes you're looking for something different—for instance, you want to find the key associated with a specific data point. Figure 1.17 shows a representation of this potential workflow.

Key	Value1	Value2	Value3
4524
2352	... ? ← known info → ? ...		
1834
3271

Figure 1.17: In some workflows, you know a value stored in another field, and you would like to know the associated key and other data from other columns.

In this case, you could use the function combination of INDEX/MATCH—so named because it combines the two functions together to achieve its aim. To understand how it works, let's take a look at these two functions:

= MATCH(what you're searching for, where you want to look, the match type)

- **what you're searching for:** This is the search parameter.
- **where you want to look:** This is the row or column in which you want to look.
- **the match type:** This is for whether you want it to be an exact match or an approximate match. In this book, we'll only be dealing with exact matches, so simply set that to 0 or FALSE.

Importantly, MATCH returns the record location where it finds its match.

This works out really well for the associated function, INDEX. INDEX takes in a row, column, or table of data, and lets you supply the coordinates of where you want to look. In that way, it functions like a reverse lookup. Let's take a look at INDEX:

=INDEX(table or column of interest, row of interest, column of interest)

- **table or column of interest:** This is a cell or table reference to a table or column.
- **row of interest:** If you supply a 4 here, the fourth record within the column or table is returned. In this case, we'll place the results of our MATCH function here since that returns the record location.
- **column of interest:** This is an optional parameter. In this case, we're using data of only one dimension (rows or columns), so we don't need this. However, if you would like to look up values across an entire table, this parameter is very useful.

Once we know where the record is located using MATCH, we can figure out the key to the associated person using INDEX. Let's see this in action in Excel.

Let's say we want to know which item has the most calories. In cells B22:24, I have broken down the steps to complete this formula (see Figure 1.18). In B22, we find the maximum calories from the NutritionInfo[Calories] column. In B23, we use MATCH to find where this value is located. Excel returns a 10. Finally, in B24, we use INDEX to look up the 10th record from NutritionInfo's Item column field. In B25, I've combined the formula into one.

=INDEX(NutritionInfo[Item], MATCH(MAX(B22), NutritionInfo[Calories], 0))

	A	B	C	D	E	F	G	H	I
1	Item	Calories							
2	Beer	200							
3	Bottled Water	0							
4	Chocolate Bar	255							
5	Chocolate Dipped Cone	300							
6	Gummy Bears	300							
7	Hamburger	320							
8	Hot Dog	265							
9	Ice Cream Sandwich	240							
10	Licorice Rope	280							
11	Nachos	560							
12	Pizza	480							
13	Popcorn	500							
14	Popsicle	150							
15	Soda	120							
16									
17									
18	How many calories are in a Hot Dog?								
19	VLOOKUP	265	=VLOOKUP("Hot Dog", NutritionInfo, 2, FALSE)						
20									
21	Which item has the most calories?								
22	MAX	560	=MAX(NutritionInfo[Calories])						
23	MATCH	10	=MATCH(B22, NutritionInfo[Calories], 0)						
24	INDEX	Nachos	=INDEX(NutritionInfo[Item], B23)						
25	INDEX/MATCH	Nachos	=INDEX(NutritionInfo[Item], MATCH(MAX(NutritionInfo[Calories]), NutritionInfo[Calories], 0))						
26									

Figure 1.18: INDEX/MATCH has been across helper cells B22:B24. It's also implemented as one formula in B25.

TIP

Combining lots of functions together is hard to read, so press Alt+Enter (Option+Return on a Mac) while in the formula bar to go to a new line. It will look like the following code. We'll be doing that for long formulas in this book.

```
=INDEX(  
    NutritionInfo[Item],  
    MATCH(MAX(B21),NutritionInfo[Calories],0)  
)
```

USE LET TO AVOID REPEATING MULTIPLE COLUMNS

Take a look at the INDEX/MATCH function we wrote:

```
=INDEX(NutritionInfo[Item], MATCH(MAX(NutritionInfo[Calories]),  
    NutritionInfo[Calories], 0))
```

Notice that we have to repeat the column `NutritionInfo[Calories]` for this problem. It's not uncommon to have to repeat multiple references and functions in Excel, especially when working with complicated datasets.

If you have the most updated version of Excel 365, you can use the `LET` function to help you stay more organized. `LET` allows you to declare a variable that exists only within the formula. For instance, we could use this instead to make the formula more readable:

```
= LET(  
    CaloriesColumn, NutritionInfo[Calories],  
    INDEX(NutritionInfo[Item],  
        MATCH(MAX(CaloriesColumn),CaloriesColumn, 0)  
    )  
)
```

Notice how we replace `NutritionInfo[Calories]` info with the `CaloriesColumn` variable. We'll be using `LET` quite a bit in this book.

XLOOKUP

For many years, `VLOOKUP` and `INDEX/MATCH` were the mainstays of Excel developers. But as you can see, both have issues: with a `VLOOKUP` you need to know the column index of interest. And `INDEX/MATCH` just feels needlessly complicated. As a result, Microsoft released a new function that can do what `VLOOKUP` and `INDEX/MATCH` do but better.

You might be wondering why I'm even bringing up `VLOOKUP` and `INDEX/MATCH` if a better alternative is out there. The answer is that while `XLOOKUP` is considered generally available,

not everyone has it yet. `XLOOKUP` is technically slower than `INDEX/MATCH`, so sometimes you might prefer the latter to save calculation time.

For those with `XLOOKUP`, this section is for you. For those who don't have it yet (you'll know you don't have it when it doesn't work), make sure you're using the latest version of Excel or Office 365.

`XLOOKUP` is effectively the best of both worlds. While there are many options in the `XLOOKUP` function, you really need to know only three things to get started: what you're looking for, the column of where you want to look, and the return column where the cell you return resides. Take a look at the representation in Figure 1.19.

What I'm looking for = ★

Key	Value1	Value2	Value3
4534
2352	...	★	...
1834
3271

Return column Search column

Figure 1.19: The elements required to implement an `XLOOKUP`

Let's implement `XLOOKUP` in Excel. Take a look at cell B28 in Figure 1.20. In this case, the goal is again to find the item with the most calories, this time using `XLOOKUP`. The following is the formula we use:

```
=XLOOKUP (MAX (NutritionInfo [Calories]), NutritionInfo [Calories],  
NutritionInfo [Item])
```

Notice that `XLOOKUP` contains many of the same elements of the `INDEX/MATCH` solution—it's just more condensed.

Once you understand lookup formulas, you're ready to move on to PivotTables.

PivotTables

What if you wanted to know the total counts of each item type you sold? Or you wanted to know revenue totals by item?

These questions are akin to “aggregate” or “group by” queries, which are a key element of descriptive analytics. The goal of descriptive analytics is to take large columns of data and push them through aggregator functions to get just a few descriptive numbers. PivotTables help us do this by grouping together different elements of a dataset and applying some type of aggregation function like `SUM`, `MIN`, `COUNT`, and others.

	A	B	C	D	E	F	G	H	I	J
1	Item	Calories								
2	Beer	200								
3	Bottled Water	0								
4	Chocolate Bar	255								
5	Chocolate Dipped Cone	300								
6	Gummy Bears	300								
7	Hamburger	320								
8	Hot Dog	265								
9	Ice Cream Sandwich	240								
10	Licorice Rope	280								
11	Nachos	560								
12	Pizza	480								
13	Popcorn	500								
14	Popsicle	150								
15	Soda	120								
16										
17										
18	How many calories are in a Hot Dog?									
19	VLOOKUP	265	=VLOOKUP("Hot Dog", NutritionInfo, 2, FALSE)							
20										
21	Which item has the most calories?									
22	MAX	560	=MAX(NutritionInfo[Calories])							
23	MATCH	10	=MATCH(B22, NutritionInfo[Calories], 0)							
24	INDEX	Nachos	=INDEX(NutritionInfo[Item], B23)							
25	INDEX/MATCH	Nachos	=INDEX(NutritionInfo[Item], MATCH(MAX(NutritionInfo[Calories]), NutritionInfo[Calories], 0))							
26										
27	Which item has the most calories?									
28	XLOOKUP	Nachos	=XLOOKUP(MAX(NutritionInfo[Calories]), NutritionInfo[Calories], NutritionInfo[Item])							
29										

Figure 1.20: XLOOKUP is implemented in cell B28. Notice the XLOOKUP formula uses many of the same cell references as INDEX/MATCH but gets to the answer in fewer steps.

To apply a PivotTable, go back to your Basketball Game Sales sheet and click into the Sales table. On the Insert tab, click PivotTable. Click OK to create a new sheet for the PivotTable.¹

In this new sheet, the PivotTable Builder will appear on the right side of the screen. The builder allows you to take the columns from the original selected data and use them as report filters, column and row labels for grouping, or values for aggregating.

Now, say you wanted to know total revenue by item. To get at that, you drag the Item tile in the PivotTable Builder into the Rows section and the Price tile into the Values section. This means that you'll be operating on revenue grouped by item name. This is shown in Figure 1.21.

¹ A PivotTable joke: A PivotTable walks into a bar. The bar tender says, "should I start a new tab?"

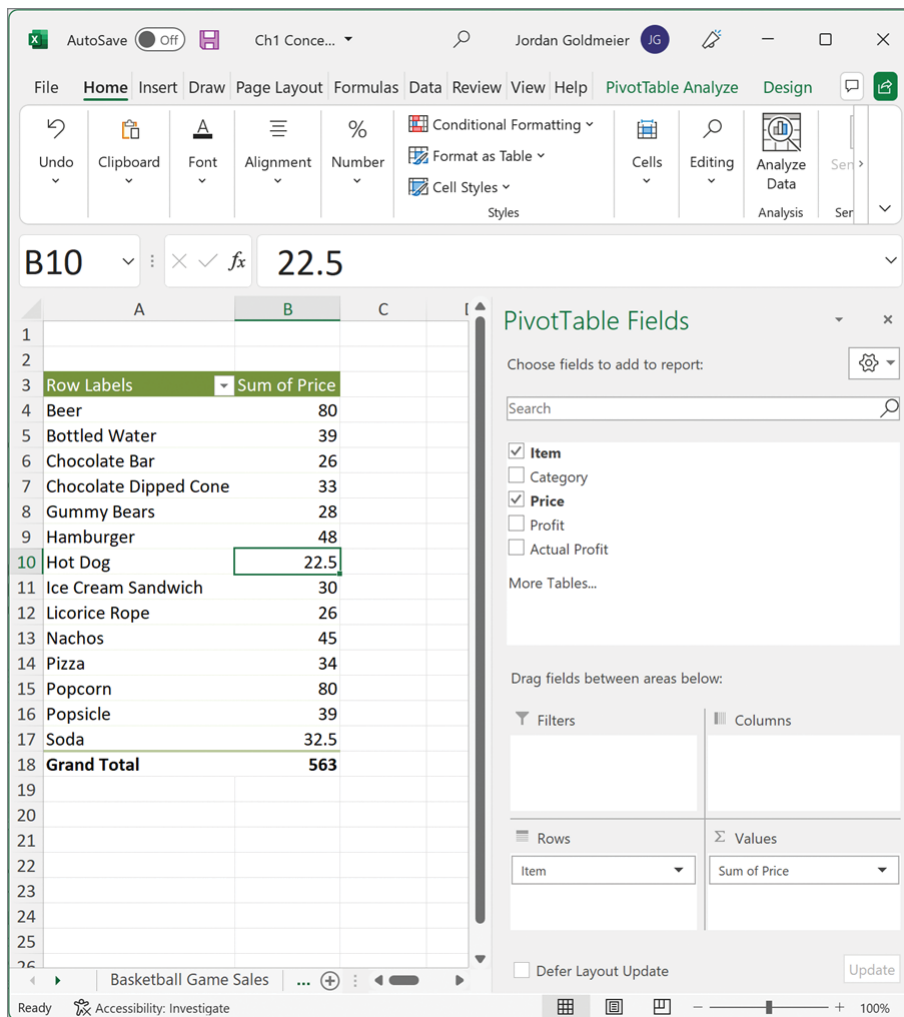


Figure 1.21: The PivotTable Builder and a count of sales by item

What if you wanted to break out these sums by category? Simply drag the Category tile into the Columns section of the builder. This gives the table shown in Figure 1.22. Note that the PivotTable in the figure automatically totals up rows and columns for you.

If you want to ever get rid of something from the table, just uncheck it or grab the tile from the section it's in and drag it out of the builder as if you were tossing it away. Go ahead and drop the Category tile. Feel free to swap in various row and column labels until you get the hang of what's going on. For instance, try to get a total calorie count sold by category using a PivotTable. When you're done with the PivotTable, change the worksheet tab name to "Pivot" to stay organized.

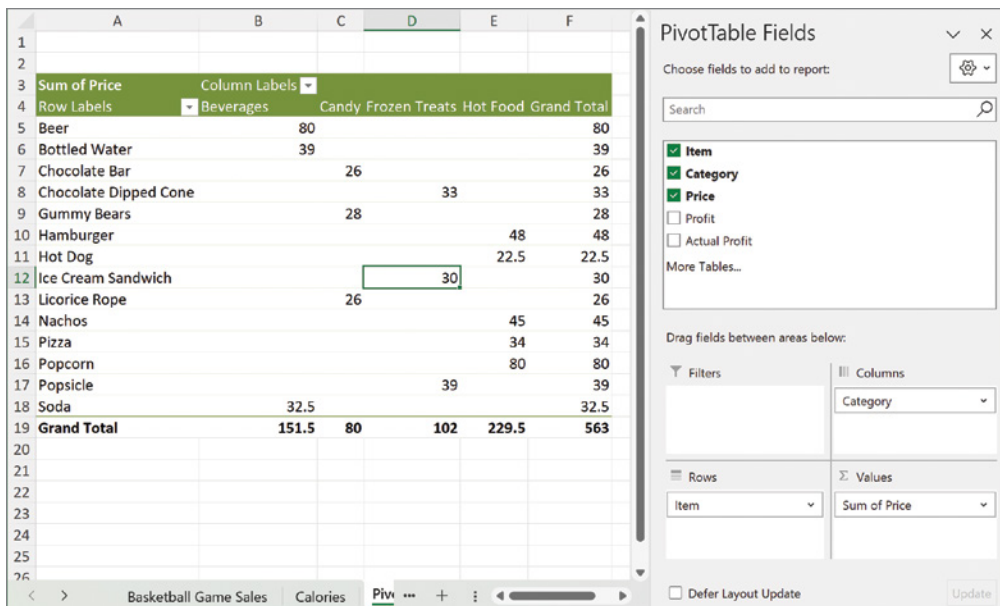


Figure 1.22: Revenue by item and category

Using Array Formulas

Array formulas in Excel might be new to you, but they've been around for a while and have gone through some major evolutions since the original *Data Smart* was published.

In the early years of Excel, array formulas required a multistep process to implement. In the last several years, however, Microsoft introduced dynamic array formulas that no longer require the complications of old. The original *Data Smart* featured the older method of implementing array formulas. However, your version of Excel likely has the newer implementation.

Still, because some Excel versions don't yet have dynamic array formulas, I will explain how to do both in the section that follows.

Let's say you owe a percentage of your sales to the athletic program. In the Concessions workbook, there is a tab called Fee Schedule. The Fee Schedule is how much from each item will go to support the local school's athletic program. Note that this data on this tab is listed horizontally. We'll need to implement the `TRANSPOSE` formula to make it vertical.

So, how much money do you owe for last night's game? To answer that question, you need to multiply the total revenue of each item by the fee schedule and then sum them all up. In other words, you need to find the dot product between the two.