Intro: Why we wrote this

It seems like everywhere we look now, people are using data in beautiful and surprising ways to present their positions or shed light on new topics. We remember the first time we saw the impact data can have on storytelling. Hans Rosling, the Swedish physician/statistician, gave a talk displaying the gapminder dataset. Rosling perfectly his paired enthusiasm with a brilliant display on the screen. As he spoke, over a dozen colorful circles slid across the protection. His Ted profile description is perfect, "In Hans Rosling's hands, data sings."

Today, most primary sources of media use data as part of their evidence base. Check out the interactive data visualizations on the UpShot at the New York Times, the visual journalism data projects at the BBC, or the daily graphs in the Economist.

The massive amounts of data available have spawned new forms of media. Nate Silver's blog covering elections and politics has grown into multiple projects on fivethirtyeight. The Pudding is an example of an online data journalism site that covers non-conventional sources of data. Vox recently won an award for producing a graph that communicates a topic that pundits could've debated endlessly.

Now that we've shown you all this cool stuff, we want to tell you why we wrote this book,

"You've found something cool on the Internet, but you have no idea what it took to make it."

There are a ton of really great resources on the Internet right now for learning data science (see here, here, and here). Many of these courses are fantastic–they can teach you programming languages, website design, database management, statistics, and machine learning. But we sometimes found the sheer volume of these courses can be overwhelming for audiences who are wanting to understand how these technologies fit together.

We decided to take a step back and write a book that describes a data science workflow, or *shows how these tools work together*. We'll show you how R, RStudio, Git, & Github can be used to create elegant yet durable data analysis projects.

We chose to center this book around a particular use case, so there will be code files and tools we'll use that are specific to this project. But we've chosen not to spend too much time on the content of these files (we've documented them you want to look into the details). Instead, We're going to focus more on the "high level" ideas because these are topics you can take with you to your next project. We also encourage you to consult the articles and resource we've recommended throughout each chapter for more materials on each topic.

Our goal for anyone reading this book

We want to show you how to 1) take something neat you found on the Internet, 2) figure out what went into making it, and 3) see if you can reproduce the result.

We plan to include enough information to get you up and running and at the same time, not overwhelm you. If you've already Googled "Getting started in data science," you know there are a *ton* of resources. Figuring out where to start can feel like trying to get a drink of water from a fire hose.

Along the way, we will also cover some practical principles of programming, command-line tools, project file organization, and a few computer science topics.

Who this book is for

We've tried to keep the materials accessible to a broad audience, but we understand there are few useful data analysis texts written for everyone. Data tends to be very specific to the field they come from, and it's hard to find data that gets everyone excited. To try and help address this issue, we use data from multiple sources (Google trends, Twitter, Wikipedia, and Googlesheets).

We focus on the workflow and tools.

The next chapters outline a 'one-stop-shop' toolset that you can learn quickly and readily re-use (because we know your time is limited).

Technical assumptions

The reader we had in mind while writing this book was someone who,

1. Uses a computer every day at work
2. Understands how to navigate a web browser (Chrome, Safari, Firefox, etc.)
3. Has worked with a word processor (like Microsoft Word, Google Docs, or Apple Papers)

If you're an accountant, scientist, analyst, journalist, grad student, product manager, or decision-maker, this book is for you.

What this book covers

We will be covering R, RStudio, Git, and Github. We use these tools daily now, but we began our careers in other statistical programs (SPSS, Stata, SAS). We abandoned those tools (we know your pain) because of the sheer number of tasks we can accomplish, and that's what makes us recommend this toolkit to you. We've also reached out to our colleagues and included their lessons and insights.

What this book doesn't cover

We also understand there are alternative approaches to accomplishing the same goal, and we will try to mention these examples wherever possible. Jupyter Lab and Jupyter Notebooks, for example, offer reproducible scientific programming environments that can accomplish many of the same objectives we'll tackle in this book. However, we still think there are reasons you should use RStudio + Github instead, and we will outline these in the following chapters.

How this book is structured

We structured this book somewhat like an Army Field Manual, which means each topic was chosen using the following criteria:

- (a) *Relative importance*. Which activities contribute most to successful training?

    *This book contains **brief descriptions** of the tools we recommend, with **diagrams and figures** outlining how they work, and **examples** for using them.*

- (b) *Need*. Which training activities will benefit the most from guidance? Which activities have received little attention in the past or which have previously required improvement?

    We'll expand on a few tools we felt are harder to grasp (Git and version control). We will also go over topics typical college courses overlook or neglect (file naming, project organization).

- (c) *Time*. How much time is available? Which activities can be effectively taught in that time?

Time is the real enemy of any data project. All computational work comes down to keystrokes and neurons. This book is trying to narrow the gap between 1) seeing a data product (neurons) and 2) translating what you see into commands on a computer that can be used to recreate that product.

- (d) *Personnel*. What are the known or suspected levels of expertise among individuals receiving training?

    We assume everyone reading this text has very little exposure to the tools we'll be covering (R, RStudio, Git, or Github). We do expect you are comfortable using a computer.

The secret to the Army's training abilities is the Field Manual (FM). Army FMs are amazing–they cover almost any topic you can imagine and are well illustrated. For example, watch this video of the drill and ceremony movement called the "counter column".

Now, look at the same thing in a figure.

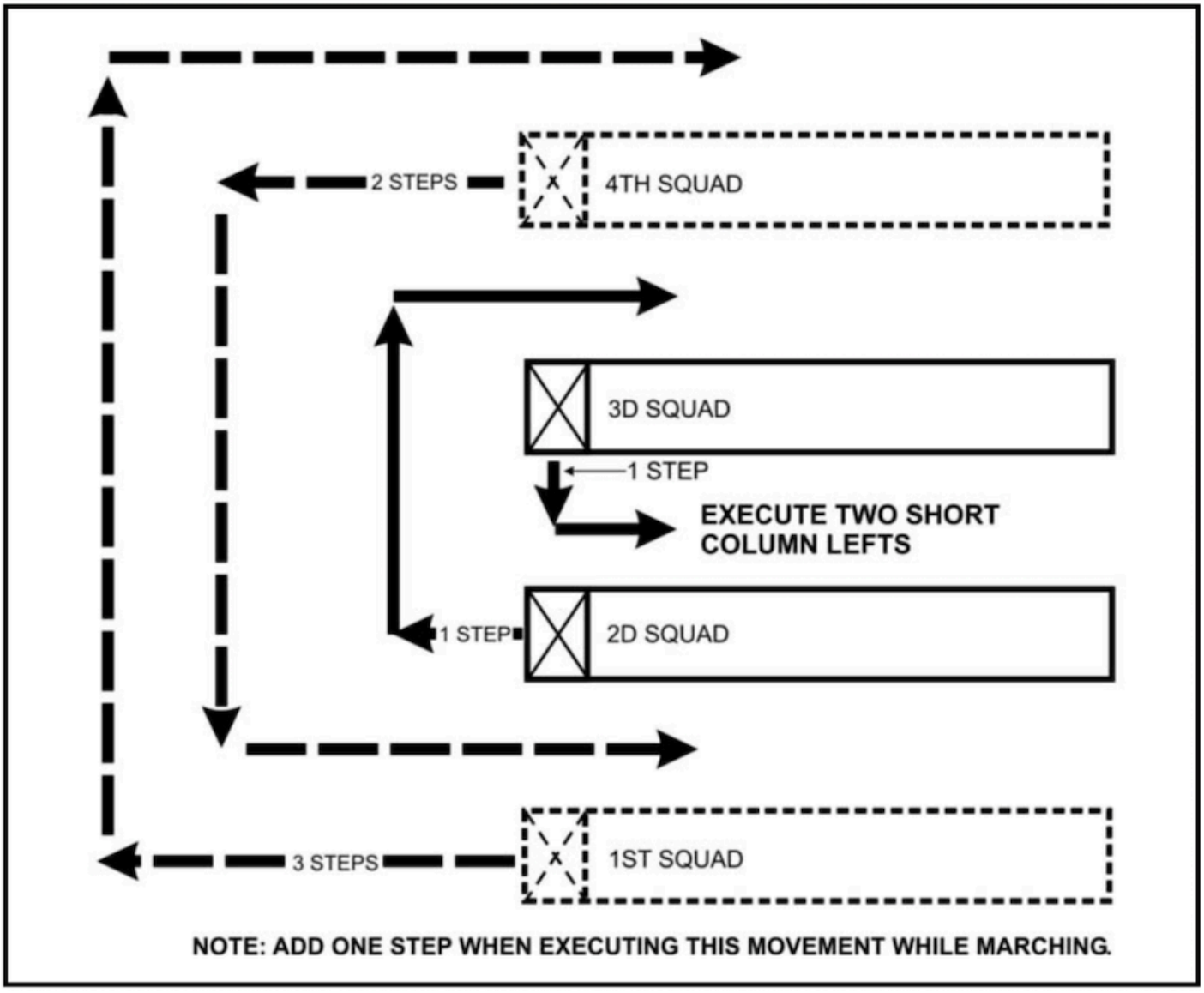

Figure 7-2. Counter-Column March at the Halt.

As you can see, executing a counter column is complicated. But the Army has taught hundreds of thousands of soldiers on this movement for decades. How? They give soldiers a field manual (FM 22–5) to read and dedicated time to practice.

The strength of the FMs is how they present information: they gave the material in everyday language (usually between a 6th–8th-grade reading level) with an emphasis on diagrams, pictures, and simple drawings.

We've found so much data science and statistical information on the internet has a ton of acronyms, jargon, and equations. We've actively avoided using technical verbiage, and focused on using figures and graphics.

> *"…there are lots of other books that explain what things are called. This book explains what they do."*

The quote above comes from Randall Munroe, author of the xkcd comic. In his book "Thing Explainer", Munroe uses pictures and plain language to describe multiple complex systems (rocket ships, the periodic table, laptops, etc.).

The subtitle of "*Thing Explainer*" is *Complicated stuff in simple words*, which is what we're trying to replicate here. Wherever possible, we've dropped unnecessary technical jargon and spelled out any acronyms.

What you'll walk away with

You will have a working project (cool visualizations, lots of code, data) a ton of resources, and a book for reproducing this process again.

Language and style guide

We use the plural 'we' throughout the book based on the excellent advice from Donald Knuth, Tracy Larabee, and Paul Roberts, "*think of a dialog between author and reader.*."

As with most written works, the topics in this book are the result of many conversations, emails, comment threads, and communications that could not have happened in isolation. We want to thank everyone who's contributed to these ideas over the years.

The text uses the following style guide:

`this is code.`

```
# this a code chunk
```

> *some quoted text*

click on hyperlinks

plain text for our thoughts

Learn more

- Practical Data Science for Stats is a resource you should bookmark in your browser. The articles in this collection will come up again in future sections, but we found we use these resources so much it's nice to have them somewhere handy.
- The R for data science community and R for Data Science book are excellent resources to help you started.
- **Collaboration and reproducibility** - there's a direct connection between collaboration and reproducibility. The better your collaborators can reproduce your work, the better they'll understand your results.

- This text is also an *opinionated technical manual*, and covers topics left out of typical statistics texts, "*Statisticians have long shied away from teaching process, with the complaint that it might limit the creativity necessary to tackle different analytical problems. However, by not teaching opinionated analysis development, we subject fledgling data to each individually spin their wheels in coming up with process for avoiding common and generalized problems*."
- We recommend RStudio and Github for anyone looking to get started with data science, visualization, reproducible reporting, dashboards development, or website/blog creation. By suggesting these particular tools, we're not saying there aren't other ways or workflows capable of accomplishing the same activities. These are the tools we've found success with, so they're what we recommend.