# Preface

If you are a scientist, an analyst, a consultant, or anybody else who has to prepare technical documents or reports, one of the most important skills you need to have is the ability to make compelling data visualizations, generally in the form of figures. Figures will typically carry the weight of your arguments. They need to be clear, attractive, and convincing. The difference between good and bad figures can be the difference between a highly influential or an obscure paper, a grant or contract won or lost, a job interview gone well or poorly. And yet, there are surprisingly few resources to teach you how to make compelling data visualizations. Few colleges offer courses on this topic, and there are not that many books on this topic either. (Some exist, of course.) Tutorials for plotting software typically focus on how to achieve specific visual effects rather than explaining why certain choices are preferred and others not. In your day-to-day work, you are simply expected to know how to make good figures, and if you're lucky you have a patient adviser who teaches you a few tricks as you're writing your first scientific papers.

In the context of writing, experienced editors talk about "ear," the ability to hear (internally, as you read a piece of prose) whether the writing is any good. I think that when it comes to figures and other visualizations, we similarly need "eye," the ability to look at a figure and see whether it is balanced, clear, and compelling. And just as is the case with writing, the ability to see whether a figure works or not can be learned. Having eye means primarily that you are aware of a larger collection of simple rules and principles of good visualization, and that you pay attention to little details that other people might not.

In my experience, again just as in writing, you don't develop eye by reading a book over the weekend. It is a lifelong process, and concepts that are too complex or too subtle for you today may make much more sense five years from now. I can say for myself that I continue to evolve in my understanding of figure preparation. I routinely try to expose myself to new approaches, and I pay attention to the visual and design choices others make in their figures. I'm also open to changing my mind. I might today consider a given figure great, but next month I might find a reason to criticize it. So with this in mind, please don't take anything I say as gospel. Think critically about my reasoning for certain choices and decide whether you want to adopt them or not.

While the materials in this book are presented in a logical progression, most chapters can stand on their own, and there is no need to read the book cover to cover. Feel free to skip around, to pick out a specific section that you're interested in at the moment, or one that covers a particular design choice you're pondering. In fact, I think you will get the most out of this book if you don't read it all at once, but rather read it piecemeal over longer stretches of time, try to apply just a few concepts from the book in your figuremaking, and come back to read about other concepts or reread sections on concepts you learned about a while back. You may find that the same chapter tells you different things if you reread it after a few months have passed.

Even though nearly all of the figures in this book were made with R and ggplot2, I do not see this as an R book. I am talking about general principles of figure preparation. The software used to make the figures is incidental. You can use any plotting software you want to generate the kinds of figures I'm showing here. However, ggplot2 and similar packages make many of the techniques I'm using much simpler than other plotting libraries. Importantly, because this is not an R book, I do not discuss code or programming techniques anywhere in this book. I want you to focus on the concepts and the figures, not on the code. If you are curious about how any of the figures were made, you can check out the book's source code at its GitHub repository.

## Thoughts on Graphing Software and Figure-Preparation Pipelines

I have over two decades of experience preparing figures for scientific publications and have made thousands of figures. If there has been one constant over these two decades, it's been the change in figure preparation pipelines. Every few years, a new plotting library is developed or a new paradigm arises, and large groups of scientists switch over to the hot new toolkit. I have made figures using gnuplot, Xfig, Mathematica, Matlab, matplotlib in Python, base R, ggplot2 in R, and possibly others I can't currently remember. My current preferred approach is ggplot2 in R, but I don't expect that I'll continue using it until I retire.

This constant change in software platforms is one of the key reasons why this book is not a programming book and why I have left out all code examples. I want this book to be useful to you regardless of which software you use, and I want it to remain valuable even once everybody has moved on from ggplot2 and is using the next new thing. I realize that this choice may be

frustrating to some ggplot2 users who would like to know how I made a given figure. However, anybody who is curious about my coding techniques can read the source code of the book. It is available. Also, in the future I may release a supplementary document focused just on the code.

One thing I have learned over the years is that automation is your friend. I think figures should be autogenerated as part of the data analysis pipeline (which should also be automated), and they should come out of the pipeline ready to be sent to the printer, with no manual post-processing needed. I see a lot of trainees autogenerate rough drafts of their figures, which they then import into Illustrator for sprucing up. There are several reasons why this is a bad idea. First, the moment you manually edit a figure, your final figure becomes irreproducible. A third party cannot generate the exact same figure you did. While this may not matter much if all you did was change the font of the axis labels, the lines are blurry, and it's easy to cross over into territory where things are less clear-cut. As an example, let's say you want to manually replace cryptic labels with more readable ones. A third party may not be able to verify that the label replacement was appropriate. Second, if you add a lot of manual post-processing to your figure-preparation pipeline, then you will be more reluctant to make any changes or redo your work. Thus, you may ignore reasonable requests for change made by collaborators or colleagues, or you may be tempted to reuse an old figure even though you've actually regenerated all the data. Third, you may yourself forget what exactly you did to prepare a given figure, or you may not be able to generate a future figure on new data that exactly visually matches your earlier figure. These are not made-up examples. I've seen all of them play out with real people and real publications.

For all these reasons, interactive plot programs are a bad idea. They inherently force you to manually prepare your figures. In fact, it's probably better to autogenerate a figure draft and spruce it up in Illustrator than to make the entire figure by hand in some interactive plot program. Please be aware that Excel is an interactive plot program as well and is not recommended for figure preparation (or data analysis).

One critical component in a book on data visualization is the feasibility of the proposed visualizations. It's nice to invent some elegant new type of visualization, but if nobody can easily generate figures using this visualization then there isn't much use to it. For example, when Tufte first proposed sparklines nobody had an easy way of making them. While we need visionaries who move the world forward by pushing the envelope of what's possible, I envision this book to be practical and directly applicable to working data scientists preparing figures for their publications. Therefore, the visualizations I propose in the subsequent chapters can be generated with a few lines of R code via ggplot2 and readily available extension packages. In fact, nearly every figure in this book, with the exception of a few figures in Chapters 26, 27, and 28, was autogenerated exactly as shown.

## Conventions Used in This Book

The following typographical conventions are used in this book:

*Italic*

Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`

Used to refer to program elements such as variable or function names, statements, and keywords.

> **TIP**
>
> This element signifies a tip or suggestion.

> **NOTE**
>
> This element signifies a general note.

**WARNING**

This element indicates a warning or caution.

## Using Code Examples

Supplemental material is available for download at *https://github.com/clauswilke/dataviz*.

This book is here to help you get your job done. In general, if example code is offered with this book, you may use it in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Fundamentals of Data Visualization* by Claus O. Wilke (O'Reilly). Copyright 2019 Claus O. Wilke, 978-1-492-03108-6."

You may find that additional uses fall within the scope of fair use (for example, reusing a few figures from the book). If you feel your use of code examples or other content falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

## O'Reilly Online Learning

**NOTE**

For almost 40 years, *O'Reilly Media* has provided technology and business training, knowledge, and insight to help companies succeed.

Our unique network of experts and innovators share their knowledge and expertise through books, articles, conferences, and our online learning platform. O'Reilly's online learning platform gives you on-demand access to live training courses, in-depth learning paths, interactive coding environments, and a vast collection of text and video from O'Reilly and 200+ other publishers. For more information, please visit *http://oreilly.com*.

## How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

800-998-9938 (in the United States or Canada)

707-829-0515 (international or local)

- 707-829-0515 (international or local)
- 707-829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at *http://bit.ly/fundamentals-of-data-visualization*.

To comment or ask technical questions about this book, send email to *bookquestions@oreilly.com*.

For more information about our books, courses, conferences, and news, see our website at *http://www.oreilly.com*.

Find us on Facebook: *http://facebook.com/oreilly*

Follow us on Twitter: *http://twitter.com/oreillymedia*

Watch us on YouTube: *http://www.youtube.com/oreillymedia*

## Acknowledgments

This project would not have been possible without the fantastic work the RStudio team has put into turning the R universe into a first-rate publishing platform. In particular, I have to thank Hadley Wickham for creating ggplot2, the plotting software that was used to make all the figures throughout this book. I would also like to thank Yihui Xie for creating R Markdown and for writing the `knitr` and `bookdown` packages. I don't think I would have started this project without these tools ready to go. Writing R Markdown files is fun, and it's easy to collect material and gain momentum. Special thanks go to Achim Zeileis and Reto Stauffer for `colorspace`, Thomas Lin Pedersen for `ggforce` and `gganimate`, Kamil Slowikowski for `ggrepel`, Edzer Pebesma for `sf`, and Claire McWhite for her work on `colorspace` and `colorblindr` to simulate color-vision deficiency in assembled R figures.

Several people have provided helpful feedback on draft versions of this book. Most importantly, Mike Loukides, my editor at O'Reilly, and Steve Haroz have both read and commented on every chapter. I also received helpful comments from Carl Bergstrom, Jessica Hullman, Matthew Kay, Tristan Mahr, Edzer Pebesma, Jon Schwabish, and Hadley Wickham. Len Kiefer's blog and Kieran Healy's book and blog postings have provided numerous inspirations for figures to make and datasets to use. A number of people pointed out minor issues or typos, including Thiago Arrais, Malcolm Barrett, Jessica Burnett, Jon Calder, Antônio Pedro Camargo, Daren Card, Kim Cressman, Akos Hajdu, Thomas Jochmann, Andrew Kinsman, Will Koehrsen, Alex Lalejini, John Leadley, Katrin Leinweber, Mikel Madina, Claire McWhite, S'busiso Mkhondwane, Jose Nazario, Steve Putman, Maëlle Salmon, Christian Schudoma, James Scott-Brown, Enrico Spinielli, Wouter van der Bijl, and Ron Yurko.

I would also more broadly like to thank all the other contributors to the tidyverse and the R community in general. There truly is an R package for any visualization challenge one may encounter. All these packages have been developed by an extensive community of thousands of data scientists and statisticians, and many of them have in some form contributed to the making of this book.

Finally, I would like to thank my wife Stefania for patiently enduring many evenings and weekends during which I spent hours in front of the computer writing ggplot2 code, obsessing over minute details of certain figures, and fleshing out chapter details.

# Chapter 1. Introduction

Data visualization is part art and part science. The challenge is to get the art right without getting the science wrong, and vice versa. A data visualization first and foremost has to accurately convey the data. It must not mislead or distort. If one number is twice as large as another, but in the visualization they look to be about the same, then the visualization is wrong. At the same time, a data visualization should be aesthetically pleasing. Good visual presentations tend to enhance the message of the visualization. If a figure contains jarring colors, imbalanced visual elements, or other features that distract, then the viewer will find it harder to inspect the figure and interpret it correctly.

In my experience, scientists frequently (though not always!) know how to visualize data without being grossly misleading. However, they may not have a well-developed sense of visual aesthetics, and they may inadvertently make visual choices that detract from their desired message. Designers, on the other hand, may prepare visualizations that look beautiful but play fast and loose with the data. It is my goal to provide useful information to both groups.

This book attempts to cover the key principles, methods, and concepts required to visualize data for publications, reports, or presentations. Because data visualization is a vast field, and in its broadest definition could include topics as varied as schematic technical drawings, 3D animations, and user interfaces, I necessarily had to limit my scope. I am specifically covering the case of static visualizations presented in print, online, or as slides. The book does not cover interactive visuals or movies, except in one brief section in Chapter 16. Therefore, throughout this book, I will use the words “visualization” and “figure” somewhat interchangeably. The book also does not provide any instruction on *how* to make figures with existing visualization software or programming libraries. The annotated bibliography at the end of the book includes pointers to appropriate texts covering these topics.

The book is divided into three parts. The first, “From Data to Visualization,” describes different types of plots and charts, such as bar graphs, scatterplots, and pie charts. Its primary emphasis is on the science of visualization. In this part, rather than attempting to provide encyclopedic coverage of every conceivable visualization approach, I discuss a core set of visuals that you will likely encounter in publications and/or need in your own work. In organizing this part, I have attempted to group visualizations by the type of message they convey rather than by the type of data being visualized. Statistical texts often describe data analysis and visualization by type of data, organizing the material by number and type of variables (one continuous variable, one discrete variable, two continuous variables, one continuous and one discrete variable, etc.). I believe that only statisticians find this organization helpful. Most other people think in terms of a message, such as how large something is, how it is composed of parts, how it relates to something else, and so on.

The second part, “Principles of Figure Design,” discusses various design issues that arise when assembling data visualizations. Its primary but not exclusive emphasis is on the aesthetic aspect of data visualization. Once we have chosen the appropriate type of plot or chart for our dataset, we have to make aesthetic choices about the visual elements, such as colors, symbols, and font sizes. These choices can affect both how clear a visualization is and how elegant it looks. The chapters in this second part address the most common issues that I have seen arise repeatedly in practical applications.

The third part, “Miscellaneous Topics,” covers a few remaining issues that didn’t fit into the first two parts. It discusses file formats commonly used to store images and plots, provides thoughts about the choice of visualization software, and explains how to place individual figures into the context of a larger document.



## Ugly, Bad, and Wrong Figures

Throughout this book, I frequently show different versions of the same figures, some as examples of how to make a good visualization and some as examples of how not to. To provide a simple visual guideline of which examples should be emulated and which should be avoided, I am labeling problematic figures as “ugly,” “bad,” or “wrong” (Figure 1-1):

*Ugly*

A figure that has aesthetic problems but otherwise is clear and informative

*Bad*

A figure that has problems related to perception; it may be unclear, confusing, overly complicated, or deceiving

*Wrong*

A figure that has problems related to mathematics; it is objectively incorrect

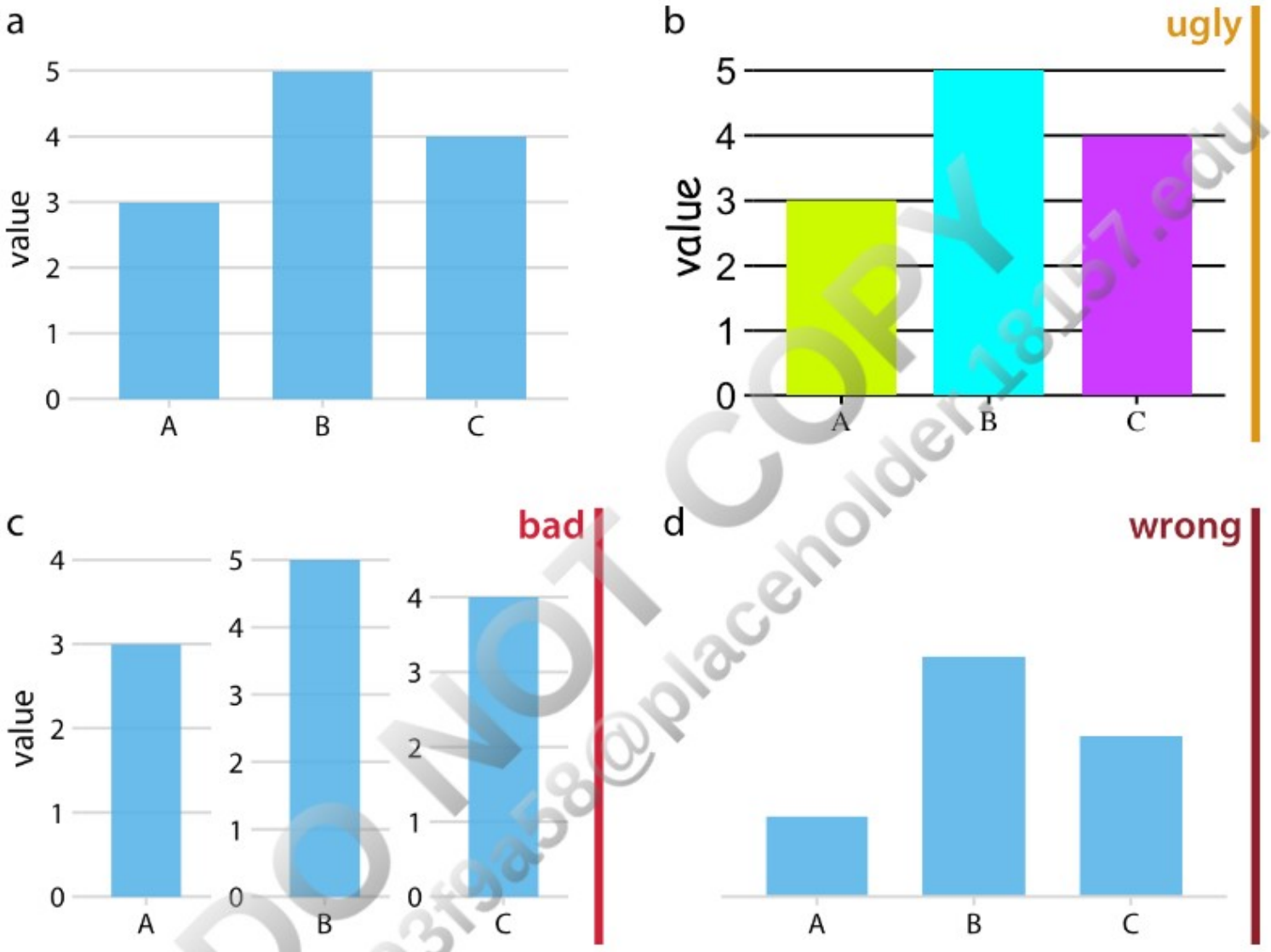


*Figure 1-1. Examples of ugly, bad, and wrong figures. (a) A bar plot showing three values (A = 3, B = 5, and C = 4). This is a reasonable visualization with no major flaws. (b) An ugly version of part (a). While the plot is technically correct, it is not aesthetically pleasing. The colors are too bright and not useful. The background grid is too prominent. The text is displayed using three different fonts in three different sizes. (c) A bad version of part (a). Each bar is shown with its own y axis scale. Because the scales don't align, this makes the figure misleading. One can easily get the impression that the three values are closer together than they actually are. (d) A wrong version of part (a). Without an explicit y axis scale, the numbers represented by the bars cannot be ascertained. The bars appear to be of lengths 1, 3, and 2, even though the values displayed are meant to be 3, 5, and 4.*

I am not explicitly labeling good figures. Any figure that isn't labeled as flawed should be assumed to be at least acceptable. It is a figure that is informative, looks appealing, and could be printed as is. Note that among the good figures, there will still be differences in quality, and some good figures will be better than others.

I generally provide my rationale for specific ratings, but some are a matter of taste. In general, the "ugly" rating is more subjective than the "bad" or "wrong" rating. Moreover, the boundary between "ugly" and "bad" is somewhat fluid. Sometimes poor design choices can interfere with human perception to the point where a "bad" rating is more appropriate than an "ugly" rating. In any case, I encourage you to develop your own eye and to critically evaluate my choices.