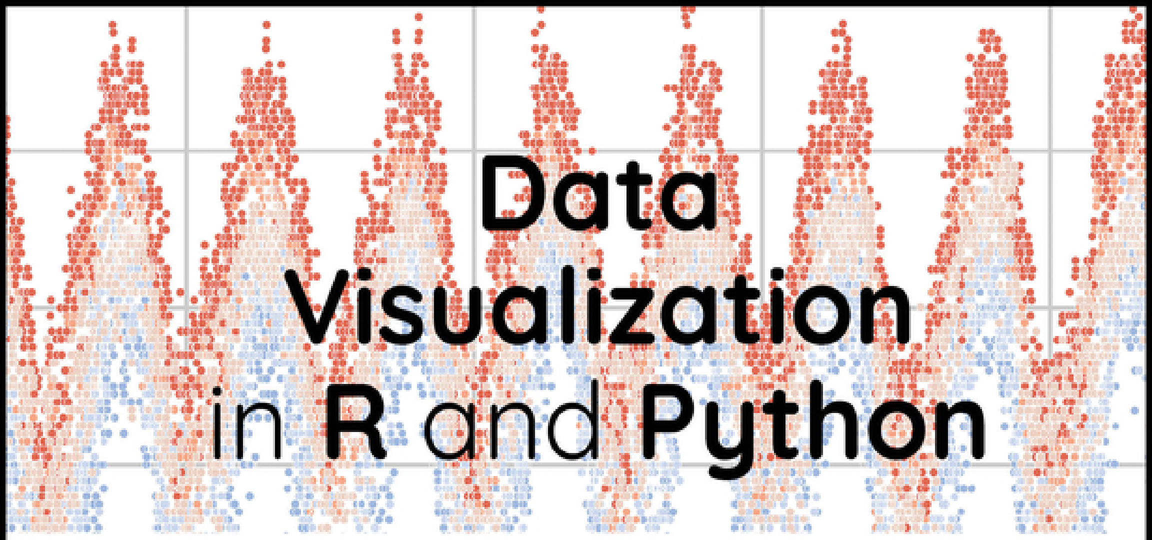
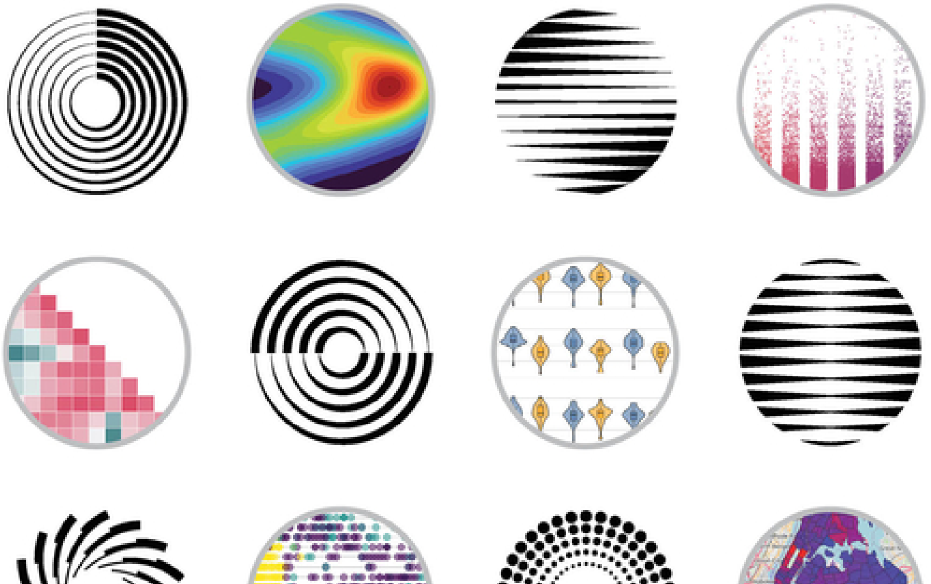


# Data Visualization in R and Python

MARCO CREMONINI



WILEY

# Data Visualization in R and Python

# Data Visualization in R and Python

*Marco Cremonini*
University of Milan, Italy

WILEY

***Library of Congress Cataloging-in-Publication Data is applied for:***

Hardback ISBN 9781394289486

Cover Design: Manuela Ruggeri
Cover Images: Courtesy of Marco Cremonini, © ulimi/Getty Images

Set in 9.5/12.5pt STIXTwoText by Straive, Chennai, India

# Contents

# Preface

The idea of this handbook came to me when I noticed something that made me pause and reflect. What I saw was that when I mentioned *data visualization* to a person who know just a little about it, perhaps adding that it involves representing data and the results of data analysis with *figures*, sometimes even *interactive one*, the reaction was often of curiosity with a shade of perplexity, the name sounded nice, but what is it, exactly? After all, if we have a table with data and we want to produce a graph, isn't it enough to search in a menu, choose the stylized figure of the graph you want to create and click? Is there so much to say to fill an entire book? When I also add that what I was talking about were completely different graphic tools from those of office automation and that, to tell the truth, it doesn't even stop at the graphics, even if they are interactive, but there are also *dashboards*, i.e. the latest evolution of data visualization, when real dynamic web applications are created, then the expression of the interlocutor was generally crossed by a shadow of concern. At that moment, I typically threw the ace up the sleeve by saying that in data visualization there are also maps, geographical maps – why not? – those are data too, they are spatial data, geographical data, and the maps are produced with the zoom, the flags, colored areas, and also cartographic maps, you may work with maps of New York, Tokyo, Paris, Rome, New Delhi, you name it.

At that point the interlocutors were usually looking puzzled, the references they had from the common experience were lost and doesn't really know what this data visualization is about, only that there actually seems to be a lot to say, enough to fill an entire book.

If anyone recognizes themselves in this interlocutor, be assured that you are in good company. Good in a literal not figurative sense, because data visualization is the Cinderella of data science that many admire but always from a certain distance, it arrives last and at the best moment it is forced to step back because there is no longer enough time to teach, study, or practice it. Yet, it frequently happens that those who, given the right opportunity to study and practice it, sense that it could be decidedly interesting, certainly prove useful and applicable in an infinite number of fields. This is due to a property that data visualization has and is instead absent in data analysis or code development: *it stimulates visual creativity together with logic*. Even statisticians and programmers use creativity, those who deny it have never been neither of them, but that is logical creativity. With data visualization, another dimension of otherwise neglected data science comes into play, *the visual language combined with computational logic*, meaning that data are represented with an expressive form that is no longer just logical and symbolic, but also perceptive, sensorial, shapes together with colors come into play, the once passive observer starts interacting, or projections of geographical areas suddenly become artifacts to use in a visual communication. Data visualization conveys different knowledge and logic for an expressive form that always has a double nature: computational for the data that feeds it, visual and sometimes interactive for the language it uses to communicate with the observer. There is enough to fill not a single book, in fact, what is contained in this book is a part of the discourse on data visualization, the one more practical and operative, other publications approach data visualization considering complementary aspects, such as the aesthetical composition of graphics, the storytelling behind a visual communication, and the syntax and semantic of a visual language together with the sensorial perception and psychology, and there is a lot to say for each one of these topics. All of them are essential for a complete understanding of the aim and extent of data visualization, but together they just don't fit in one single handbook, unless presented in a truly superficial fashion, for this reason almost every book on data visualization focuses more explicitly on a few of those aspects. This book is dedicated to the more operational and computational issues, because you have to know the low-level logic behind modern data visualization artifacts and you have to know and practice with tools, they are not all alike, "just pick the easiest to use and you're all set" is definitely not a good advice and, given the liveliness of the proprietary data visualization tools' market, it is easy to forget about open-source ones, which instead rival and often surpass what proprietary tools are able to offer; may be with a little more of initial efforts, but not much.

To conclude, data visualization probably deserves better consideration in educational programs and a recognition as a coherent and evolving discipline. It could be a lot of fun to study and practice it, it could make also you pause and reflect about tools for communicating data science results with a visual language, and it includes many different aspects from diverse disciplines, both theoretical and practical, all converging and enmeshing in a coherent body of knowledge. These are all good characteristics for curious persons. The Cinderella role of data visualization can be overcome by recognizing its educational and professional value and, no less important, its creative stimulus.

October 8, 2024

*Marco Cremonini*
University of Milan

# Introduction

When you mention *data visualization* to a person who doesn't know it, perhaps adding that it involves data and the results of data analysis with *figures,* sometimes even *interactive ones,* the reaction you observe is often that the person in front of you looks intrigued but doesn't know exactly what it consists of. After all, if we have a table with data and we want to produce a graph, isn't it enough to open the usual application, go to a certain drop-down menu, choose the stylized figure of the graph you want to create and click? Is there so much to say to fill an entire book? At that moment, when you perceive that the interlocutor is thinking of the well-known spreadsheet product, you may add that those described in the book are graphic tools completely different from those of office automation and, to tell the truth, we don't even stop at the graphics, even if interactive, but there are also *dashboards*, namely the latest evolution of data visualization, when it is transformed into dynamic web applications, and to obtain dashboards it is not sufficient to click on menus but you have to go deeper into the inner logic and mechanisms. It's then that the expression of the interlocutor is generally crossed by a shadow of concern and you can play the ace up your sleeve by saying that in data visualization there are also *maps*, geographical maps, sure, those are made by data too: spatial data and geographical data, and the maps can be produced with the many available widgets such as zoom, flags, and colored areas; and we even go beyond simple maps, because there are also cartographic maps with layers of cartographic quality, such as maps of Rome, of Venice, of New York, of the most famous, and also not-so-famous cities and places, possibly with very detailed geographical information.

At that point the interlocutor has likely lost the references she or he had from the usual experience with office automation products and doesn't really know what this data visualization is, only that there seems to be a lot to say, enough to fill an entire book. If anyone recognizes themselves in this imaginary interlocutor (imaginary up to a certain point, to be honest), know that you are in good company. Good in a literal not figurative sense, because data visualization is a little like the Cinderella of data science that many admire from a certain distance, it arrives last in a project and sometimes it does not receive the attention it deserves. Yet there are many who, given the right opportunity to study and practice it, sense that it could be interesting and enjoyable, it could certainly prove useful and applicable in an infinite number of areas, situations, and results. This is due to a property that data visualization has and is instead absent in traditional data analysis or code development: *it stimulates visual creativity together with logic*. Even statisticians and programmers use creativity, those who deny it have never really practiced one of those disciplines, but that is logical creativity. With data visualization, another dimension of data science that is otherwise neglected comes into play, *the visual language combined with computational logic*, the data represented with an expressive form that is no longer just logical and formal, but also perceptive, and sensorial, comes into play with shapes, colors, use and projections of space, and it is always accompanied with meaning that the originator wish to convey and the observers will interpret, often subjectively. Data visualization conveys different knowledge and logic for an expressive form that always has a double soul: computational for the data that feeds it, visual and sometimes interactive for the language it uses to communicate with the observer. Data visualization has always a double nature: it is a key part of data science for its methods, techniques, and tools, and it is storytelling; who produces visual representations from data tells a story that may have different guises and may produce different reactions. There is enough to fill not just a single book.

## Organization of the Work: Foundations and Advanced Contents

The text is divided into four parts already mentioned in the previous introduction. The ***first part*** presents the *fundamentals* of data visualization with Python and R, the two reference languages and environments for data science, employed to create *static graphs* as a direct result of a previous data wrangling (import, transformation) and analysis activity. The reference libraries for this first part are *Seaborn* for Python and *ggplot2* for R. They are both modern open-source graphics libraries and in constant evolution, both produced by the *core developers* and with the contributions of the respective *communities,* very large and lively in engaging in continuous innovations. *Seaborn* is the more recent of the two and partly represents an evolved interface of Python's traditional *matplotlib graphics library*, made more functional and enriched with features and graph types popular in modern data visualization. *Ggplot2* is the traditional graphic library for R, unanimously recognized as one of the best ever, both in the open-source and proprietary world. Ggplot is full of high-level features and constantly evolving, it receives contributions from researchers and developers from various scientific and application fields. A simply unavoidable tool for anyone approaching data visualization. The two have different settings, more traditional Seaborn, with a collection of functions and options for the different types of charts supported. Instead, ggplot is organized by overlapping graphic levels, according to a setting that goes by the name of *grammar of graphics,* shared by some of the most widespread digital graphics tools, and suitable for developing even unconventional types of graphics, thanks to the extreme flexibility it allows. This first part covers about a third of the work.

The ***second part*** introduces *Altair,* a Python library capable of producing interactive graphics in HTML and JSON format, as well as static versions in bitmap (PNG and JPG) and vector (SVG) formats. Altair is a young but solid graphic library because in all respects it represents a modern interface of *Vega-Lite,* a graphic library with an established tradition for web applications thanks to the declarative syntax in JSON format. Altair offers the same web-oriented functionality as Vega-Lite for typical data science use, with a syntax that supports the definition of overlapping graphical layers and aesthetics composed with a syntax that is easy to use and common to similar tools. This second part presents a higher level of difficulty than the first, but certainly within reach for those who have acquired the fundamental knowledge given by the first part. The first and second parts cover approximately half of the work.

The third and fourth parts represent ***advanced*** data visualization contents. The difficulty increases and so does the commitment required, on the other hand, we face two real worlds: that of *web dashboards* and of *spatial data* and *maps*. The term *dashboard* may be new to many, but dashboards are not. Whenever you access environments on the web that show menus and configurable graphic objects according to user's choices and content in the form of data or graphs, what you are using is most likely a dashboard. If you access Open Data of a large institution, such as the Organisation for Economic Co-operation and Development (OECD) or the United Nations, or even an internal company application that displays graphs and statistics, you are most likely using a dashboard. Numerous systems and products for creating dashboards with different technologies are available, it is a vast market. In data science environments with Python and R, there are two formidable tools, *Plotly/Dash* and *Shiny*, respectively. They are professional tools, and the list of relevant organizations using them is long. They are also irreplaceable teaching tools for learning the logic and basic mechanisms of a dashboard, which, in its final form, is a web application, therefore integrated with the typical technology of pages and websites. However, a Dash or Shiny dashboard is also something else, it is the terminal point of a *pipeline* that begins with the fundamentals of data science, data import, data wrangling, data analysis, and then static and dynamic graphs. The dashboard is the final end in which everything is concentrated and integrated: logic, mechanisms, requirements, and creativity. Technically they are challenging due to the presence of reactive logic which allows them to be dynamic and interactive and due to the integration of various components. The text discusses and develops examples of medium complexity, with different solutions, from web scraping of online content to the integration of Altair interactive graphics.

The second world that opens up, that of *geographical maps*, is undeniably fascinating. *Spatial data*, choropleth *maps*, the simplest ones with the colored areas (such as maps with areas colored according to the coalition that won the elections or the rate of unemployment by province, region, or nation), but also *maps* based on *cartography data* are the declination given by data science of a discipline that has very ancient roots and still constitutes an almost independent environment composed of high-resolution maps and geographic information systems (GIS), with its specializations and professional skills. Until a few years ago, data science tools could not even touch that world, but today they have come surprisingly close. This is thanks to extraordinary progress in open-source systems and tools, Python but above all R, which is now offering formidable tools capable of also using shape files from a technical cartography and geographic coordinate systems according to international standards. In the examples presented, geographic and cartographic files from Venice, Rome, and New York were used with the aim of showing the impressive potential offered by the Python and R tools.

## Who is it Aimed at?

It is simple to specify to whom this text is addressed: it is addressed to everyone. Anyone who finds data visualization interesting, and images useful for their work, study, and the skills they are building, will find a learning path that starts from the fundamentals and goes up to cartography and web applications. Of course, saying "it's aimed at everyone" is simple, then doubts may arise in the reader, "but am I also part of those *everyone*?" Trying to make a list of those included in this "everyone" will inevitably leave out someone, but we could certainly mention students, researchers, and instructors of social, political, and economic sciences. In addition to many generic data, they may have spatial data to represent (e.g. movement of people and goods, global supply chains, logistics, spatial or ethnographic analyses). Next, students, researchers, and instructors of marketing, communication, public relations, journalism, media, and advertising, for whom interactive representations via the web and graphics in general are important, as products and skills. Also, students, researchers, and instructors of scientific and medical disciplines could be interested, they often deal with sophisticated graphic representations, for example in biology or epidemiology, without forgetting that the graphic contributions from the genomics and molecular biology community are among the most numerous. Students, researchers, and instructors of engineering, management, or bioengineering, for example, use data science tools and visualization as an integral part of their analyses. Historians, archaeologists, and paleontologists produce high-quality graphic representations, so the text can be useful for them too. Well, the list is already long, and I'm definitely forgetting someone who should be mentioned instead.

In general, undergraduate and graduate students, teachers, researchers, and Ph.D. students will find many examples and explanations to help them graphically present content and organize exercises. Likewise, professionals and companies, for their corporate and institutional communication and training, may enjoy the material presented in the text.

What I'm trying to say is that data visualization, like data science as a whole, is not a sectoral discipline for which you need to have a specific background, such as a statistician, computer scientist, engineer, or graphic designer. It is not necessary at all, in fact the opposite is needed, that is, that data visualization and data science be as transversal as possible, being studied and used by all those who, for their formation and work interests, in their specific field, from economics to paleontology, from psychology to molecular biology, find themselves working with data, whether numerical, textual, or spatial and find useful to obtain high-quality visual representations from those data, perhaps interactive or structured in dashboards.

## What is Required and What is Learned

To follow and learn the contents of the text it is necessary to know the fundamentals of data science with Python and R, meaning those concerned with importing and reading operations of datasets and the typical data wrangling operations (sorting, aggregations, shape and type transformations, selections, and so on). Numerous examples are presented in the text which include the data wrangling part (the cases where it is longer can be found in the Supplementary Material), so to replicate a visualization all the necessary code is available, starting from reading the Open Data. Therefore, it is not required to independently produce the preliminary part of operations on the data, but it is necessary to be able to interpret the logic and the operations that are performed. Hence the need to know the fundamentals, as well as the possibility of producing variations of the examples.

Another aspect that may appear problematic is knowledge of the fundamentals of data science with both Python and R because often one only knows one of the two environments and languages. In this regard, I would like to reassure anyone who finds themselves in this situation. If you know data wrangling operations with R or Python, interpreting the logic of those carried out with the other language requires little effort, at most the details of the syntax will need some specific attention and learning efforts. But here a second consideration comes into play: knowledge of both Python and R is particularly useful in modern data science, those who know only one of the two probably just need a good opportunity to learn the second, discovering that the learning curve is much smoother than could have been imagined and the effort will be certainly reasonable. The advantage will be considerable in terms of new features and tools that will become available.

The organization into parts also suggests a progression and division in learning and teaching. The *first part* is also suitable for those who have just learned the fundamentals of data science and can be carried out in parallel with the study of those fundamentals. Most static graphs require basic data wrangling operations and generating graphs can be a great educational tool for demonstrating the logic and use of data wrangling operations. The presentation of the different types of static graphs follows an order of increasing complexity, from the first intuitive and easily modifiable in infinite variations, up to the last ones which require knowledge of some important properties of statistical analysis. The difficulty level of code is generally low. The *second part* is a natural continuation of the first. The Altair library has a linear and clear syntax, so the greater difficulty introduced by the interactive features, especially in terms of computational logic, is completely within reach for anyone who has learned the fundamentals contained in the first part. The result will be motivating, the Altair interactive graphics are of excellent quality, allowing various configurations and alternative solutions.

Between these two parts and the subsequent *third* and *fourth parts*, there is a gap in terms of what is required and what is learned, for this reason in the initial introductory part the last two parts were presented as *advanced content*. It is necessary to have acquired a good familiarity with the fundamentals, confidence in searching for information in the documentation of libraries, and knowing how to patiently and methodically manage errors. In other words, you need to have done a good number of exercises with the fundamental part.

For the *third part on dashboards,* it is necessary to have basic knowledge of HTML, CSS, and in general how a traditional web page is made. They are not difficult notions, but it may take some time to acquire them. You don't need more advanced knowledge, such as JavaScript or web application frameworks. You also need to have gained some confidence in writing scripts in Python and R. In both cases you learn the basic reactive mechanisms to manage interactivity, it is a different logic from the traditional one.

For the *fourth part on maps,* it is necessary to learn the fundamental notions of geographic coordinate systems, the form of geographic data with the typical organization in geometries, and the often-necessary coordinate transformations. The tools used are partly known, *ggplot* for R and *pandas* for Python, but many new ones will be encountered because in any case, not only in the world of cartography but also in that of data science, the logic, methods, and tools to use spatial data have specificities that distinguish them. As mentioned initially, there are some initial difficulties to overcome and it is required to go into the details of the shape of the spatial data, but the use of these data and the production of geographical maps is fascinating, right from the first and simple choropleth maps. However, it is right after those initial maps that there's the real beauty of working with spatial data and geographic maps.

## What is Excluded

As always, or almost always, much remains excluded from the content of a book, sometimes simply due to the need not to exceed a certain number of pages, sometimes out of pure forgetfulness, and often due to a conscious choice by the Author. All three motifs also exist in this work. For the first, that is simply how a publisher works; for the second, other than apologizing I don't know what to say since those are things I've forgotten; for the third however, there is something to comment on, if for no other reason than to give some explanation of the motives for exclusions by choice.

The first obvious exclusion is the absence of proprietary technologies and tools. For data visualization there are many proprietary solutions, from very specialized ones produced by small companies to generalist ones produced by big players. Manufacturers of data visualization software will say that their tools are better than those presented in this book. For some aspects, it might be true, but almost always it is false and in general, to define itself as better than the open-source tools of Python and R would require several distinctions and clarifications that are rarely presented. One of the main reasons is the ease of use of the graphical interfaces of proprietary tools compared to the low-level programming of open-source ones. An old, worn out, and now out-of-date issue that is slowly, perhaps, starting to be overcome. It is obvious that learning to click sequences of buttons and menus or drag graphic icons is initially simpler than writing code with a programming language. The initial learning curve is different in the two cases. The point, however, lies in that adjective, "initial." What happens next? What is the purpose of learning to use these tools? If the purpose is educational, teaching and learning the fundamentals and advanced contents of data visualization, there is practically no choice, only the environments and tools that exhibit low-level details are teaching tools. The others simply aren't. They are suitable for professional training courses on that particular instrument, but not for basic teaching or learning. This is enough to exclude any proprietary instrument from this text. It should be noted that some of the most modern proprietary tools (or perhaps made by intelligent manufacturers) are integrating the open-source technologies of Python and R into their frameworks, with the idea of offering both possibilities.

Then there is a specific and perhaps surprising exclusion among the basic chart types, and not one of the exotic kind that very few use, on the contrary of the most widespread, very widespread indeed. The excluded is *pie charts* and reason is simply that *it is not useful* in the true sense of data visualization in data science. The statement will seem surprising, in what sense are pie charts, ubiquitous and used millions of times, not useful? I will briefly explain the reason, which is also shared by many who deal with data visualization. A graph is produced to visually represent the information contained in certain data and this representation is based on at least two conditions: (1) that the visual representation is clear and interpretable in an unambiguous way and (2) that with the graph, the information contained in the data is easier to understand than the tabular form (or at least of equal difficulty). Pie charts satisfy neither condition. They are ambiguous because the relative size of the slices is often unclear and above all they make it more difficult to interpret the data than the equivalent table. In other words, if the table with the values is presented instead of the pie chart, the reader has easier, clearer, and more understandable information. On the contrary, bar charts are one of the fundamental type of graphics, despite the fact that pie charts are simply the polar coordinate representation of a bar chart. So why this difference and why pie charts are so common? The reason for the difference is that visually evaluating angles is considerably more difficult than comparing linear heights. Pie charts are mostly used because they just give a touch of color to an otherwise monotonous text, not for their informative content. And what about the difficulty of evaluating the slice proportions? Well, the numerical values are often added to the slices, that is, in practice, to rewrite the data table right over the graphic.

To conclude, data visualization deserves more space in educational programs and clearer recognition as a coherent and evolving discipline and body of knowledge. The Cinderella role of data science can be overcome by recognizing its educational value and, no less importantly, its creative stimulus.

# About the Companion Website

This book is accompanied by a companion website:

**https://www.wiley.com/go/Cremonini/DataVisualization1e**

This website includes:

- Codes
- Figures
- Datasets

# Part I

# Static Graphics with ggplot (R) and Seaborn (Python)

## Grammar of Graphics

The grammar of graphics was cited in the Introduction and will continue to be mentioned in the rest of the text. We see a brief summary here. The concept of grammar of graphics was proposed by Leland Wilkinson in the early 2000s with the idea of creating grammatical, mathematical, and aesthetic rules to define the graphics that were produced by statistical analysis. The different approach, with respect to the fixed definition of chart types composed of stylized reference schemes, is that a graph's grammar would instead have allowed previously unknown flexibility. In Wilkinson's definition, seven fundamental components were identified, but the construction by overlapping layers was not yet highlighted. It is Hadley Wickham, core developer of R and ggplot, who in 2010 introduced the *layered* grammar of graphics, with which Wilkinson's approach was updated by reviewing the fundamental elements. The definition by levels provides the representation of the data, combining statistics and geometries, two of the fundamental elements, together with *positions*, *aesthetics*, *scales*, a *coordinate system*, and possibly *facets*. We will find all these elements in ggplot and Altair, the two graphic libraries organized according to the grammar of graphics considered in this book, as well as in the recent but still preliminary Seaborn Objects interface of Seaborn, the reference graphic library for Python.

## References

Leland Wilkinson, *The Grammar of Graphics*, 2nd Ed., Springer-Verlag New York 2005, https://doi.org/10.1007/0-387-28695-0.

Leland Wilkinson, The Grammar of Graphics, Chapter 13, *Handbook of Computational Statistics, Concepts and Methods*, 2nd Ed., Gentle J.E., Härdle W. K. and Mori Y. (eds.), Springer-Verlag Berlin Heidelberg 2012. https://doi.org/10.1007/978-3-642-21551-3.

Wickham, H. (2010). A Layered Grammar of Graphics. *Journal of Computational and Graphical Statistics* 19 (1): 3–28. http://dx.doi.org/10.1198/jcgs.2009.07098.

# 1

# Scatterplots and Line Plots

Scatterplots, with the main variant represented by line plots, are the fundamental type of graphic for pairs of continuous variables or for a continuous variable and a categorical variable and, in addition to representing the most common type of graphic together with bar plots (or bar charts/bar graphs), form the basis for numerous variations. The logic that guides a scatterplot graphic is to represent with *markers* (e.g., dots or other symbols) the values that two quantities (variables and attributes) take on during a certain number of observations. The pairs of variables are conventionally associated with the Cartesian axes $x$ and $y$, with scales ordered in ascending order, according to units of measurement, which may be different. By convention, the variable associated with the $x$-axis is said to be the *independent* variable and put in relation with the *dependent* variable on the $y$-axis, meaning that what is shown is implicitly a correlation between the two variables. This traditional interpretation of the meaning of the representation of variables on Cartesian axes must be put in the right context to avoid all too frequent errors. The result of a scatterplot graphic, in no case, demonstrates the existence of a cause–effect relationship between two variables. The cause–effect relationship must already be known in order to consider one variable as truly dependent on the other. Or it must be demonstrated, extending the graphic analysis with considerations about the phenomenon observed and the reasons in favor of the existence of such a cause–effect relationship. Conversely, a scatterplot simply shows how pairs of values from two variables are distributed for a sample of observations, nothing is said about the reasons. A typical example that is often presented considers the height and weight of a certain sample of people. Each person represents a single observation, the two quantities have different units of measurement and for each person, the intersection between the coordinates of height ($x$-axis) and weight ($y$-axis) is marked with a dot (or other marker). In this case, we know that there is a cause–effect relationship between the two physical characteristics: a greater height tends to correspond to a greater weight for purely physiological reasons regarding body size. This does not mean that a tall person always weighs more than a short one, which is obviously false, but only that this tends to be true, given a homogeneous sample of the population.

### Dataset

*Inflation, consumer prices (annual %)*, World Bank Open Data (https://data.worldbank.org/indicator/FP.CPI.TOTL.ZG).

*Copyright*: Creative Commons CC BY-4.0 (https://creativecommons.org/licenses/by/4.0/)

*Compiled historical daily temperature and precipitation data for selected 210 U.S. cities*, Yuchuan Lai and David Dzombak, Carnegie Mellon University, April 2022, DOI: 10.1184/R1/7890488.v5 (https://kilthub.cmu.edu/articles/dataset/Compiled_daily_temperature_and_precipitation_data_for_the_U_S_cities/7890488).

*Copyright*: Creative Commons CC0 1.0 DEED.

*GDP growth (annual %)*, World Bank Open Data (https://data.worldbank.org/indicator/NY.GDP.MKTP.KD.ZG).*Copyright*: Creative Commons CC BY-4.0 (https://creativecommons.org/licenses/by/4.0/).

## 1.1 R: ggplot

### 1.1.1 Scatterplot

Let us start with the just mentioned relation between height and weight of a sample of people. For this, we can use the dataset *heights*, predefined into package *modelr*, which is part of the *tidyverse* package. For simplicity, we always assume to load the *tidyverse* package for all R examples. The dataset refers to a sample of US citizens collected in a 2012 study of the *U.S. Bureau of Labor Statistics*. Values are expressed as centimeters and kilograms, for readers familiar with the Imperial system, they could be visualized simply by omitting the two transformations into centimeters and kilograms with the conversion coefficients shown in the code.

```
library(tidyverse)

df= modelr::heights
df$height_cm= 2.54*df$height
df$weight_kg= 0.45359237*df$weight

df
# A tibble: 7,006 × 10
   income height weight   age marital  sex    height_cm weight_kg
    <int>  <dbl>  <int> <int> <fct>    <fct>   <dbl>     <dbl>
 1  19000     60    155    53 married  female  152.       70.3
```