

R for Non-Programmers: A Guide for Social Scientists

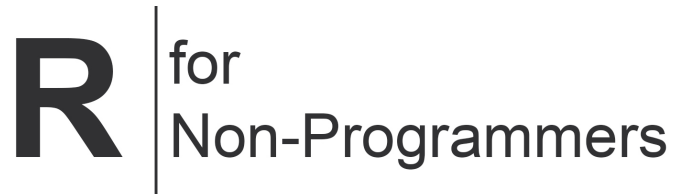
Daniel Dauber

2021-07-15

Contents

Welcome	5
Acknowledgments	7
1 Why learn a programming language as a non-programmer?	9
1.1 Learning new tools to analyse your data is always essential . . .	10
1.2 Programming languages enhance your conceptual thinking	10
1.3 Programming languages allow you to look at your data from a different angle	11
1.4 Learning any programming language will help you learn other programming languages.	11
2 Setting up R and RStudio	13
2.1 Installing R	13
2.2 Installing RStudio	14
2.3 When you first start RStudio	14
2.4 Updating R and RStudio: Living at the pulse of innovation . . .	15
2.5 RStudio Cloud	15
3 RStudio Interface	17

Welcome



Welcome to *R for Non-Programmers: A guide for Social Scientists*. This book is intended to be of help to everyone who wishes to enter the world of R Programming, but not necessarily for the purpose to become a programmer. Instead, this book intends to convey key concepts in data analysis, especially quantitative research.

[Add more]

Acknowledgments

Special thanks are due to my wife, who supported me in so many ways to get this book completed. Also, I would like to thank my son, who patiently watched me sitting at the computer type this book up.

Chapter 1

Why learn a programming language as a non-programmer?

‘*R*’, it is not just a letter you learn in primary school, but a powerful programming language. While it is used for a lot of quantitative data analysis, it has grown over the years to become a powerful tool that excels (#no-pun-intended) in handling data and performing customised computations with quantitative and qualitative data.

R is now one of my core tools to perform various types of analysis because I can use it in many different ways, for example,

- statistical analysis,
- corpus analysis,
- development of online dashboards to dynamically generate interactive data visualisations,
- connection to social media APIs for data collection,
- Creation of reporting systems to provide individualised feedback to research participants,
- Drafting and writing research articles, etc.

Learning R is like learning a foreign language. If you like learning languages, then ‘R’ is just another one.

While *R* has become a comprehensive tool for data scientists, it has yet to find its way into the mainstream field of Social Sciences. Why? Well, learning programming languages is not necessarily something that feels comfortable to

everyone. It is not like Microsoft Word, where you can open the software and explore it through trial and error. Learning a programming language is like learning a foreign language: You have to learn vocabulary, grammar and syntax. Similar to learning a new language, programming languages also have steep learning curves and require quite some commitment.

For this reason, most people do not even dare to learn it because it is time-consuming and often not considered a ‘*core method*’ in Social Sciences disciplines. Apart from that, tools like SPSS have very intuitive interfaces, which seem much easier to use (or not?). However, the feeling of having ‘*mastered*’ *R* (although one might never be able to claim this) can be extremely rewarding.

I guess this introduction was not necessarily helpful in convincing you to learn any programming language. However, despite those initial hurdles, there are a series of advantages to consider. Below I list some good reasons to learn a programming language as they pertain to my own experiences.

1.1 Learning new tools to analyse your data is always essential

Theories change over time, and new insights into certain social phenomena are published every day. Thus, your knowledge might get outdated quite quickly. This is not so much the case for research methods knowledge. Typically, analytical techniques remain over many years. We still use the mean, mode, quartiles, standard deviation, etc., to describe our quantitative data. Still, there are always new computational methods that help us to crunch the numbers even more. *R* is a tool that allows you to venture into new analytical territory because it is open source. Thousands of developers provide cutting-edge research methods free of charge for you to try with your data. You can find them on platforms like GitHub. *R* is like a giant supermarket, where all products are available for free. However, to read the labels on the product packaging and understand what they are, you have to learn the language used in this supermarket.

1.2 Programming languages enhance your conceptual thinking

While I have no empirical evidence for this, I am very certain it is true. While I would argue that my conceptual thinking is quite good, I would not necessarily say that I was born with it. Programming languages are very logical. Any error in your code will make you fail to execute it properly. Sometimes you face challenges in creating the correct code to solve a problem. Through creative abstract thinking (I should copyright this term), you start to approach your problems differently, whether it is a coding problem or a problem in any other

1.3. PROGRAMMING LANGUAGES ALLOW YOU TO LOOK AT YOUR DATA FROM A DIFFERENT ANGLE

context. For example, I know many students enjoy the process of qualitative coding. However, they often struggle to detach their insights from the actual data and synthesise ideas on an abstract and more generic level. Qualitative researchers might refer to this as challenges in '*second-order deconstruction of meaning*'. This process of abstraction is a skill that needs to be honed, nurtured and practised. From my experience, programming languages are one way to achieve this, but they might not be recognised for this just yet.

1.3 Programming languages allow you to look at your data from a different angle

There are certainly commonly known and well-established techniques regarding how you should analyse your data rigorously. However, it can be quite some fun to try techniques outside your disciplines. This does not only apply to programming languages, of course. Sometimes, learning about a new research method enables you to look at your current tools in very different ways too. One of the biggest challenges for any researcher is to reflect on your work. Learning new and maybe even '*strange*' tools can help with this. Admittedly, sometimes you might find out that some new tools are also a dead-end. Still, you might have learned something valuable through the process of engaging with your data differently. So shake off the rust of your analytical routine and blow some fresh air into your research methods.

1.4 Learning any programming language will help you learn other programming languages.

Once you understand the logic of one language, you will find it relatively easy to understand new programming languages. Of course, if you wanted to, you could become the next '*Neo*' (from 'The Matrix') and change the reality of your research forever. On a more serious note, though, if you know any programming language already, learning R will be easier because you have accrued some basic understanding of these particular types of languages.

Having considered everything of the above, do you feel ready for your next foreign language?

Chapter 2

Setting up R and RStudio

Every journey starts with gathering the right equipment. This intellectual journey is not much different. The first step that every 'R novice has to face is to set everything up to get started. There are essentially two strategies:

- Install *R* and RStudio

or

- Run RStudio in a browser via RStudio Cloud

While installing *R* and Studio requires more time and effort, I strongly recommend it, especially if you want to work offline or make good use of your computer's CPU. However, if you are not sure yet whether you enjoy learning *R*, you might wish to look at RStudio Cloud first. Either way, you can follow the examples of this book no matter which choice you make.

2.1 Installing R

The core module of our programming is R itself, and since it is an open-source project, it is available for free on Windows, Mac and Linux computers. Here is what you need to do to install it properly on your computer of choice:

1. Go to www.r-project.org
2. Click on 'CRAN' where it says download.
3. Choose a server in your country (all of them work, but downloads will perform quicker)

4. Select the version you want to install (I recommend the latest version)
5. Open the downloaded file and follow the installation instructions. (I recommend leaving the suggested settings as they are).

(INSERT IMAGES ABOUT HERE)

This was relatively easy. You now have *R* installed. Technically you can start using *R* for your research, but there is one more tool I strongly advise installing: RStudio.

2.2 Installing RStudio

R by itself is just the *‘beating heart’* of *R* programming, but it has no particular user interface. If you want buttons to click and actually ‘see’ what you are doing, there is no better way than RStudio. It will be the primary tool to interact with *R* and the only software you need to do all the fun parts. To install RStudio perform the following steps:

1. Go to www.rstudio.com.
2. Go to Products > RStudio
3. On this page, select RStudio Desktop
4. Select the *‘free’* option and begin the download
5. Open the downloaded file and follow the installation instructions (again, keep it to the default settings as much as possible)

Congratulations, you are all set up to learn R.

2.3 When you first start RStudio

You are all set up to get started, but if you want the best experience, you might want to make some tweaks to your settings. I recommend at least the following two by clicking on *RStudio > Preferences > Code*:

1. In the *‘Editing’* tab, make sure to have at least the first four options ticked, especially the ‘Auto-indent code after paste’. This setting will save time when trying to format your coding appropriately, making it easier to read.

(INSERT IMAGE)

2. In the `*Display*` tab, you might want to have the first three options selected. In particular, `*Highlight selected line*` is helpful because, in more complicated code, it is helpful to see where your cursor is.

Finally, everything is ready, and we can get to the most exciting part of this post.

2.4 Updating R and RStudio: Living at the pulse of innovation

While not strictly something that helps you become a better programmer, this advice might come in handy to avoid becoming a frustrated programmer. When you update your software, you need to update R and RStudio separately from each other. While both R and RStudio work closely with each other, they still constitute separate pieces of software. Thus, it is important to keep in mind that updating RStudio will not automatically update R. This can become problematic if specific packages you installed via RStudio (like a fancy learning algorithm) might not be compatible with earlier versions of R. Also, additional R packages developed by other people are separate pieces and are updated too, independently from R and RStudio.

I know what you are thinking: This already sounds complicated and cumbersome. However, rest assured, we take a look at how you can easily update all your packages with RStudio itself. Thus, all you need to remember is: *R* needs to be updated separately from everything else. Enough of spoiling future posts, back to *R* and RStudio. It is time to take a closer look at it by finally opening RStudio.

2.5 RStudio Cloud

Chapter 3

RStudio Interface