How to Use R for Data Science

Lecture Notes

© Prof. Dr. Stephan Huber (Stephan.Huber@hs-fresenius.de)

Last compiled on 22 March, 2023

Contents

Preface

About the notes

- This script aims to support my lecture at the HS Fresenius. It is incomplete and no substitute for taking actively part in class.
- The old version of the notes can be found here PDF.
- I appreciate you reading it, and I appreciate any comments.
- This is work in progress so please check for updates regularly.
- The lecture notes are available online or you can download it as a pdf file: here
- These notes are published under a Creative Commons BY-SA license (CC BY-SA) version 4.0. This means it can be reused, remixed, retained, revised and redistributed as long as appropriate credit is given to the authors. If you remix, or modify the original version of this open textbook, you must redistribute all versions of this open textbook under the same license—CC BY-SA. This script is build on the work of Navarro (2020), Muschelli and Jaffe (2022), Thulin (2021), and Ismay and Kim (2022) which is also published under CC BY-SA.

About the author

Figure 1: Prof. Dr. Stephan Huber[1]

[1]Picture is taken from https://sites.google.com/view/stephanhuber

Prof. Dr. Stephan Huber is Professor of International Economics and Data Science at *HS Fresenius* and holds a Diploma in Economics from the *University of Regensburg* and a Doctoral Degree (summa cum laude) from the University of Trier. He completed postgraduate studies at the *Interdisciplinary Graduate Center of Excellence at the Institute for Labor Law and Industrial Relations in the European Union (IAAEU)* in Trier. He was a research assistant to Prof. Dr. Dr. h.c. Joachim Möller at the *University of Regensburg*, post-doc at the *Leibniz Institute for East and Southeast European Studies (IOS)* in Regensburg and freelancer at *Charles University* in Prague.

He has worked as a lecturer at various institutions including the *TU Munich*, the *University of Regensburg*, *Saarland University*, and the *Universities of Applied Sciences in Frankfurt and Augsburg*. He has also taught abroad for the *University of Cordoba* in Spain and the *University of Perugia*. Professor Huber has published his work in international journals such as the *Canadian Journal of Economics* and the *Stata Journal*. More on his work can be found on his private homepage www.t1p.de/stephanhuber.

Contact

Hochschule Fresenius für Wirtschaft & Medien GmbH
Im MediaPark 4c
50670 Cologne

Office: 4b OG-1 Bü01 (Office hour: Thursday 1-2 p.m.)
Telefon: +49 221 973199-523
Mail: stephan.huber@hs-fresenius.de
Private homepage: www.t1p.de/stephanhuber
Github: https://github.com/hubchev

Chapter 1

Getting Started with R

Before we can start exploring data in R, there are some key concepts to understand first:

1. Why R?
2. How to learn R?
3. What are R and RStudio?
4. How to use R and RStudio without installation
5. How to install R and RStudio
6. How to write and run code in R
7. What are R packages?

1.1 Why R?

R is a free and open-source programming language that provides a wide range of advanced statistics capabilities, state-of-the-art graphics, and powerful data manipulation capabilities. It supports larger data sets, reads any type of data, and runs on multiple platforms. R makes it easier to automate tasks, organize projects, ensure reproducibility, and find and fix errors, and anyone can contribute packages to improve its functionality. Moreover, the following points are worth to emphasize:

- **R is an artist!** Check out:
  - https://www.r-graph-gallery.com/
  - http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html

  - https://www.r-bloggers.com/2020/05/7-useful-interactive-charts-in-r/

- **R is an employment insurance.** If you are good in R programming or if you are good in writing programming code in general, you have plenty of opportunities to earn a decent salary.
- **R uses the computer and computers are great!** Doing statistics on a computer is faster, easier and more powerful than doing it by hand. Computers excel at mindless repetitive tasks. For most people, the only reason to ever do statistical calculations with pencil and paper is for learning purposes.
- **Excel is bad!** Doing statistics in a spreadsheet (e.g., Microsoft Excel) is often a bad idea. Although many people are likely feel more familiar with them, spreadsheets are very limited in terms of what analyses they allow you to do. You can easily lose the overview and it is hard to keep track of what you have done and in comparison with command line driven programs. In particular, the ability to make your analysis *replicable* is limited.
- **R is good, proprietary software is bad!** Avoiding proprietary software is a very good idea because it costly, support is exclusively provided by the owner of the software (if they stop supporting your version you are lost), security issues cannot be checked as the source code is not available, and possibilities for customization are limited.
- **R is big!** Something that you might not appreciate now, but will love later on if you do anything involving data analysis, is the fact that R is highly extensible. When you download and install R, you get all the basic packages, and those are very powerful on their own. However, because R is so open and so widely used, it's become something of a standard tool in statistics, and so lots of people write their own packages that extend the system. And these are freely available too. One of the consequences of this, I've noticed, is that if you open up an advanced textbook (a recent one, that is) rather than introductory textbooks, is that a *lot* of them use R. In other words, if you learn how to do your basic statistics in R, then you're a lot closer to being able to use the state of the art methods than you would be if you'd started out with a "simpler" system: so if you want to become a genuine expert in data analysis, learning R is a very good use of your time.
- **R is the future!** Programming is a core skill in research, economics, and business. R is one of the most widely used programming languages in the world today. It is used in almost every industry such as finance, banking, medicine or manufacturing. R is used for portfolio management, risk analytics in finance and banking industries.

1.2 How to learn R

There are many different approaches to learning R. It pretty much depends on your preferences, needs, goals, prerequisites and limitations. It is up to you to search and find a suitable way to achieve the learning goals. However, I offer these notes and if you are in one of my classes, you can ask at any time for help.

The notes should walk you through many of the things that are important when working in R, and it should help you dig deeper and learn more if you want to. In particular, I recommend my swirl courses. However, there are thousands of other resources for learning R: textbooks, online courses, videos, guided tutorials, and so (see section 1.3).

Below, I'll give you a list of resources that are worth a look. You might find what you're looking for there. If not, just keep reading this book. Above all, those who have personally taken one of my courses are welcome to contact me if they think I can help them.

Warning: R is not without its weaknesses: It's not easy to learn, it has some very annoying quirks that we all have to deal with, it's slower than other languages (Phyton, MATLAB), and R's algorithms and sources are spread across many packages (since there's no big company behind it that wants you to buy it). This sometimes makes it very difficult for beginners to find what they are looking for. In simple words: you can get lost!

Tips on learning to code: Learning to code/program is quite similar to learning a foreign language. It can be daunting and frustrating at first. Such frustrations are common and it is normal to feel discouraged as you learn. However, just as with learning a foreign language, if you put in the effort and are not afraid to make mistakes, anybody can learn and improve.

Here are a few useful tips to keep in mind as you learn to program:

- **Remember that computers are not actually that smart**: You may think your computer or smartphone is "smart," but really people spent a lot of time and energy designing them to appear "smart." In reality, you have to tell a computer everything it needs to do. Furthermore, the instructions you give your computer can't have any mistakes in them, nor can they be ambiguous in any way.
- **Take the "copy, paste, and tweak" approach**: Especially when you learn your first programming language or you need to understand particularly complicated code, it is often much easier to take existing code that you know works and modify it to suit your ends. This is as opposed to trying to type out

the code from scratch. We call this the *"copy, paste, and tweak"* approach. So early on, we suggest not trying to write code from memory, but rather take existing examples we have provided you, then copy, paste, and tweak them to suit your goals. After you start feeling more confident, you can slowly move away from this approach and write code from scratch. Think of the "copy, paste, and tweak" approach as training wheels for a child learning to ride a bike. After getting comfortable, they won't need them anymore.

- **The best way to learn to code is by doing**: Rather than learning to code for its own sake, we find that learning to code goes much smoother when you have a goal in mind or when you are working on a particular project, like analyzing data that you are interested in and that is important to you.
- **Practice is key**: Just as the only method to improve your foreign language skills is through lots of practice and speaking, the only method to improving your coding skills is through lots of practice. Don't worry, however, we'll give you plenty of opportunities to do so!

1.3 Learning resources

AWESOME

R Learning Resources

Thousand of freely available books and resources exist. On (https://bookdown.org/) and in the Big Book of R is a big collection of links to R books that verifies my claim. Another nice collection of learning ressources can be found here: AWESOME R Learning-Resources

In Rstudio you find in the left panel at the bottom a panel that is called *Help*. There you find a lot of links, manuals, and references that offer you tons of resources to learn R for free including: (https://education.rstudio.com/) and (https://support.rstudio.com/hc/en-us/articles/200552336-Getting-Help-with-R)

Since you may feel overwhelmed by the number of resources, I would like to highlight four books:

Timbers, Campbell, and Lee (2022): **Data Science: A First Introduction** is a free and up to date book that comes with exercises with worksheets that are available on UBC-DSCI GitHub repository

Wickham and Grolemund (2023): **R for Data Science: Import, Tidy, Transform, Visualize, and Model Data** is the most popular source to learn R. It focuses on introducing the tidyverse package and is freely available online.

Irizarry (2022): **Introduction to Data Science: Data Analysis and Prediction Algorithms With R** is a complete, up to date, and applied introduction.

Venables, Smith, and R Core Team (2022) **An Introduction to R: Notes on R: A Programming Environment for Data Analysis and Graphics** is a manual from the R Core Development Team that shows how to use R without having to install and load additional packages.

Some other sources that are worth mentioning are these:

- The search engine www.rseek.org is R specific and often better than www.google.com as it only searches for content that has to do with the programming language R.
- On www.rdocumentation.org you can find the complete documentation of all R packages.
- Many find these cheatsheets helpful.

1.4 What are R and RStudio?

Throughout this book, we will assume that you are using R via RStudio. First time users often confuse the two. At its simplest, R is like a car's engine while RStudio is like a car's dashboard as illustrated in Figure 1.1.

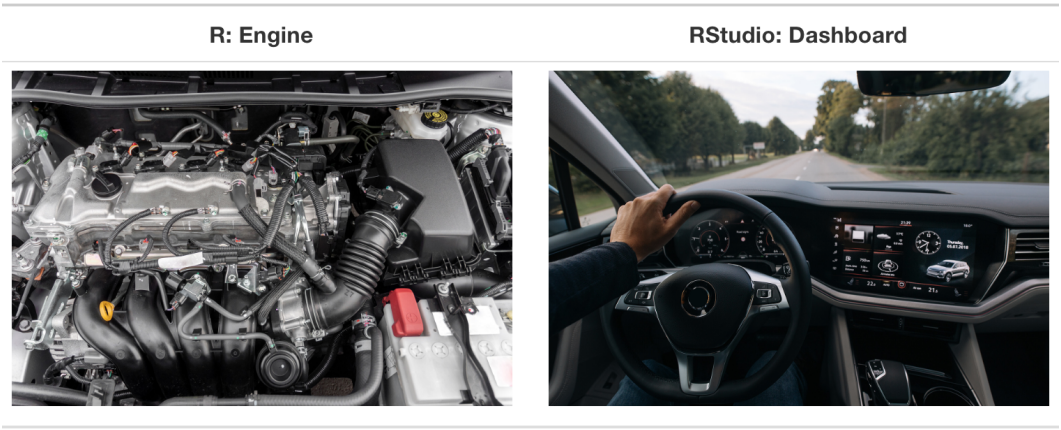

Figure 1.1: Analogy of difference between R and RStudio.

More precisely, R is a programming language that runs computations, while RStudio is an *integrated development environment (IDE)* that provides an interface by adding many convenient features and tools. So just as the way of having access to a speedometer, rearview mirrors, and a navigation system makes driving much easier, using RStudio's interface makes using R much easier as well.

Much as we don't drive a car by interacting directly with the engine but rather by interacting with elements on the car's dashboard, we won't be using R directly but rather we will use RStudio's interface. After you install R and RStudio on your computer, you'll have two new *programs* (also called *applications*) you can open. We'll always work in RStudio and not in the R application. Figure 1.2 shows what icon you should be clicking on your computer.

After you open RStudio, you should see something similar to Figure 1.3 where three or four panels dividing the screen.

1. The *Environment* panel, where a list of the data you have imported and created can be found.
2. The *Files*, *Plots* and *Help* panel, where you can see a list of available files, will be able to view graphs that you produce, and can find help documents for different parts of R.

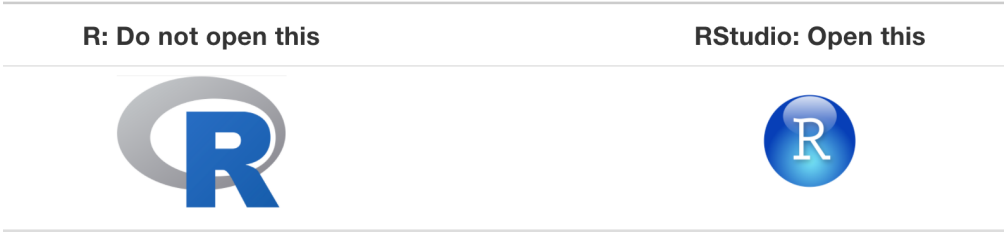

Figure 1.2: Icons of R versus RStudio on your computer.

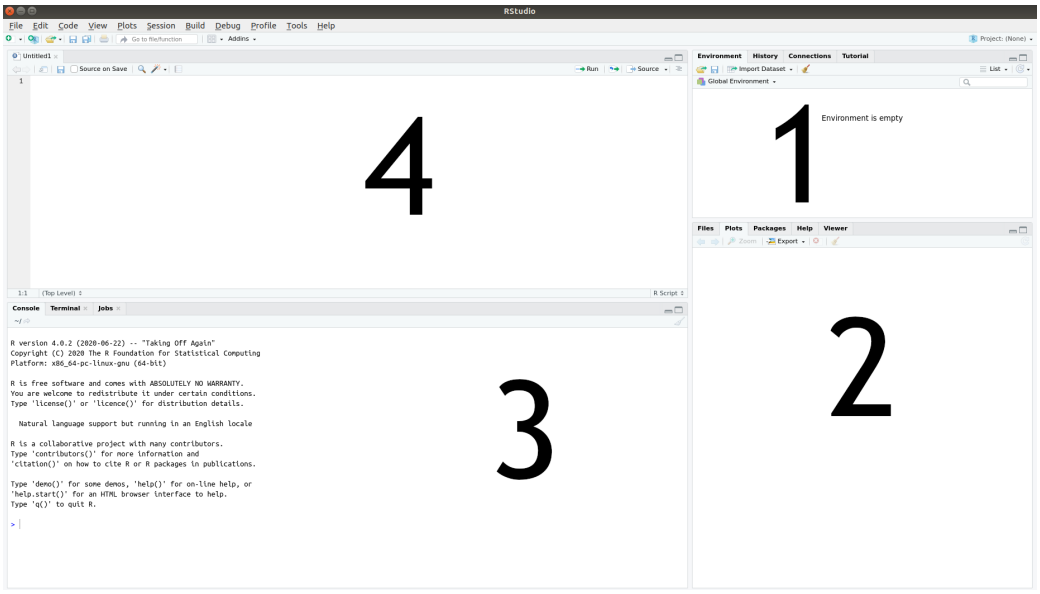

Figure 1.3: RStudio interface to R.

3. The *Console* panel, used for running code. This is where we'll start with the first few examples.
4. The *Script* panel, used for writing code. This is where you'll spend most of your time working.

The *Console* panel will contain R's startup message, which shows information about which version of R you're running. My startup message at the time of writing was as follows:

> R version 4.1.2 (2021-11-01) – "Bird Hippie" Copyright (C) 2021 The R Foundation for Statistical Computing Platform: x86_64-pc-linux-gnu (64-bit)
>
> R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under certain conditions. Type 'license()' or 'licence()' for distribution details.
>
> Natural language support but running in an English locale
>
> R is a collaborative project with many contributors. Type 'contributors()' for more information and 'citation()' on how to cite R or R packages in publications.
>
> Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help. Type 'q()' to quit R.

If you don't have panel number 4, open it by opening an existing R-script or creating a new one. You can create a new on by clicking *Ctrl+Shift+N* (alternatively, you can use the menu: File→New File→R Script).

You can resize the panels as you like, either by clicking and dragging their borders or using the minimise/maximise buttons in the upper right corner of each panel. Clicking *Ctrl++* and *Ctrl+-* allows to make the fonts larger or smaller.

When you exit RStudio, you will be asked if you wish to *save your workspace*, meaning that the data that you've worked with will be stored so that it is available the next time you run R. That might sound like a good idea, but in general, I recommend that you don't save your workspace, as that often turns out to cause problems down the line. It is almost invariably a much better idea to simply rerun the code you worked with in your next R session.

1.5 How to use R and RStudio without installation

If you don't want to install R on your PC or you don't have admin rights to do so, you can use RStudio online doing *cloud computing* on https://posit.cloud/. Posit Cloud (formerly RStudio Cloud) is a cloud-based solution that allows anyone to do, share, teach and learn data science online. It is free for individuals with some restrictions and limited capacities.

1.6 Installing R and RStudio

You will first need to download and install both R and RStudio (Desktop version) on your computer. It is important that you install R first and then install RStudio.

1. **You must do this first:** Download and install R by going to https://cloud.r-project.org/.

- If you are a Windows user: Click on "Download R for Windows", then click on "base", then click on the Download link.
- If you are macOS user: Click on "Download R for (Mac) OS X", then under "Latest release:" click on R-X.X.X.pkg, where R-X.X.X is the version number. For example, the latest version of R as of November 25, 2019 was R-3.6.1.
- If you are a Linux user: Click on "Download R for Linux" and choose your distribution for more information on installing R for your setup.

1. **You must do this second:** Download and install RStudio at https://www.rstudio.com/products/rstudio/download/.

- Scroll down to "Installers for Supported Platforms" near the bottom of the page.
- Click on the download link corresponding to your computer's operating system.

1.7 What are R packages?

A package is basically just a big collection of functions, data sets and other R objects that are all grouped together under a common name. Some packages are already installed when you put R on your computer, but the vast majority of them of R packages are out there on the internet, waiting for you to download, install and use them. R packages are collections of functions and data sets developed by the community. They increase the power of R by improving existing base R functionalities, or by adding new ones. For example, if you are usually working with data frames, probably you will have heard about *dplyr* or *data.table*, two of the most popular R packages. More than 10,000 packages are available at the official repository (CRAN) and many more are publicly available through the internet.

In this section, I'll describe how to work with packages using the Rstudio tools. Along the way, you'll see that whenever you get Rstudio to do something (e.g., install a package), you'll actually see the R commands that get created.

However, before we get started, there's a critical distinction that you need to understand, which is the difference between having a package **installed** on your computer, and having a package **loaded** in R. When you install R on your computer only a small number of packages come bundled with the basic R installation. The installed packages are on your computer. The critical thing to remember is that just because something is on your computer doesn't mean R can use it. In order for R to be able to *use* one of your installed packages, that package must also be *loaded*. Generally, when you open up R, only a few of these packages (about 7 or 8) are actually loaded. Basically what it boils down to is this:

1. A package must be installed before it can be loaded.
2. A package must be loaded before it can be used.

We only need to install a package once on our computer. However, to use the package, we need to load the library every time we start a new R environment. You can think of this as installing a bulb versus turning on the light.

The two step process might seem a little odd at first, but the designers of R had very good reasons to do it this way. That is, there are more than 10.000 packages, and probably about 8000 authors of packages, and no-one really knows what all of them do. Keeping the installation separate from the loading minimizes the chances that two packages will interact with each other in a nasty way. Moreover having installed all available packages would probably blow your hard disk.

Another good analogy for R packages is they are like apps you can download onto a mobile phone:

Figure 1.4: Installing packages

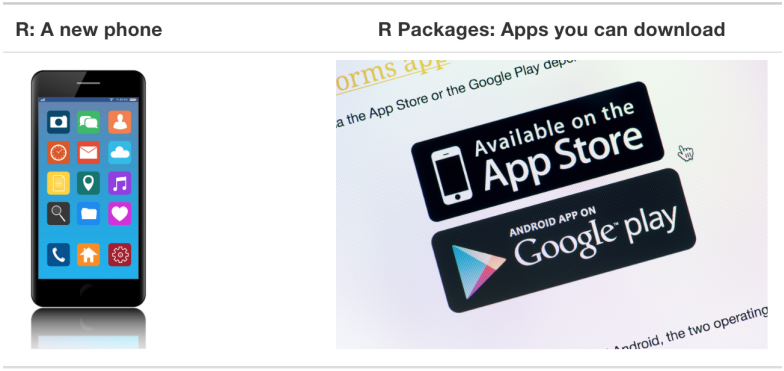

Figure 1.5: Analogy of R versus R packages.

So R is like a new mobile phone: while it has a certain amount of features when you use it for the first time, it doesn't have everything. R packages are like the apps you can download onto your phone from Apple's App Store or Android's Google Play.

1.7.1 Package installation

There are two ways to install an R package: an easy way and a very easy way. Let's install the `ggplot2` package the easy way first as shown in Figure 1.6. In the Files pane of RStudio:

a) Click on the "Packages" tab.
b) Click on "Install" next to Update.
c) Type the name of the package under "Packages (separate multiple with space or comma):" In this case, type `ggplot2`.
d) Click "Install."

An alternative way to install a package is by typing `install.packages("ggplot2")`

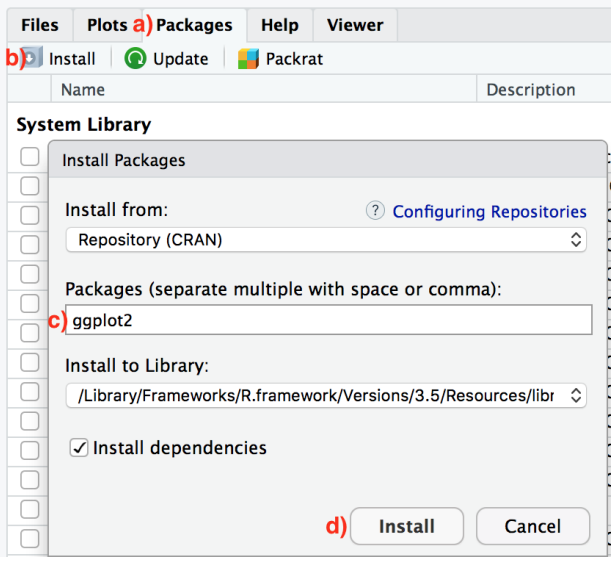

Figure 1.6: Installing packages in R the easy way.

in the console pane of RStudio and pressing Return/Enter on your keyboard. Note you must include the quotation marks around the name of the package.

Much like an app on your phone, you only have to install a package once. However, if you want to update a previously installed package to a newer version, you need to reinstall it by repeating the earlier steps.

1.7.2 Package loading

Recall that after you've installed a package, you need to *load it*. In other words, you need to *open it*. We do this by using the `library()` command.

For example, to load the `ggplot2` package, run the following code in the console pane. What do we mean by "run the following code"? Either type or copy-and-paste the following code into the console pane and then hit the Enter key.

```
library(ggplot2)
```

If after running the earlier code, a blinking cursor returns next to the `>` "prompt" sign, it means you were successful and the `ggplot2` package is now loaded and ready to use. If, however, you get a red "error message" that reads ...

```
Error in library(ggplot2) : there is no package called 'ggplot2'
```

. . . it means that you didn't successfully install it. If you get this error message, go back to section 1.7.1 on R package installation and make sure to install the `ggplot2` package before proceeding.

1.7.3 Package use

One very common mistake new R users make when wanting to use particular packages is they forget to *load* them first by using the `library()` command we just saw. Remember: *you have to load each package you want to use every time you start RStudio.* If you don't first *load* a package, but attempt to use one of its features, you'll see an error message similar to:

```
Error: could not find function
```

This is a different error message than the one you just saw on a package not having been installed yet. R is telling you that you are trying to use a function in a package that has not yet been *loaded*. R doesn't know where to find the function you are using. Almost all new users forget to do this when starting out, and it is a little annoying to get used to doing it. However, you'll remember with practice and after some time it will become second nature for you.

Chapter 2

Learn interactively with *swirl*

The *swirl* R package makes it fun and easy to learn R programming and data science. *swirl* teaches you R programming and data science interactively, at your own pace, and right in the R console! If you are new to R, have no fear. *swirl* will walk you through each of the steps required to employ Rstudio and R for your purpose. To start it, please follow my instructions precisely:

Open Rstudio and type in the console the following:

```
install.packages("swirl")
library("swirl")
ls()
rm(list=ls())
install_course_github("hubchev", "swirl-it")
swirl()
```

The above lines of code do the following:

- Install the *swirl* package.
- Load the *swirl* package.
- List the content of the environment.
- Remove everything from the environment.
- Start *swirl*.
- Install my *swirl* course that is hosted on GitHub.
- With *swirl* you start *swirl* and your learning experience.

If the course has failed to install, you can try to download the file `swirl-it.swc` from https://github.com/hubchev/swirl-it and install the course with `install\_course()`

Please choose the course *swirl-it* and the learning module *huber-intro-1*. You can exit *swirl* at any time by typing `bye()` or by clicking the *Esc* on your keyboard.

After you have successfully finished learning module *huber-intro-1* please go ahead with the learning module *huber-intro-2* that is also part of my swirl course *swirl-it*.

swirl modules on data analytical basics

In my swirl modules *huber-data-1*, *huber-data-2*, and *huber-data-3* I introduce some very basic statistical principles on how to analyse data.

swirl module on the `tidyverse` package

I compiled a short *swirl* module to introduce the *tidyverse* universe. This is a powerful collection of packages which I discuss later on. The learning module is also part of my *swirl-it* course.

Other *swirl* modules

You can also install some other courses. You find a list of courses here http://swirlstats.com/scn/index.html or here https://github.com/swirldev/swirl_courses

I can recommend the following:

```
swirl::install_course_github("swirldev", "R_Programming_E")
swirl::install_course_github("matt-dray", "tidyswirl")
swirl::install_course("Getting and Cleaning Data")
swirl::install_course_github("sysilviakim", "swirl-tidy")
swirl::install_course("Regression Models")
```

Literature

References

Irizarry, R. A. (2022). *Introduction to data science: Data analysis and prediction algorithms with R*. CRC Press. Accessed January 30, 2023. Retrieved from https://rafalab.github.io/dsbook/

Ismay, C., & Kim, A. Y. (2022). *Statistical inference via data science: A moderndive into R and the tidyverse*. CRC Press. Accessed January 30, 2023. Retrieved from https://moderndive.com/

Muschelli, J., & Jaffe, A. (2022). *Introduction to r for public health researchers* (This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.) GitHub. (https://github.com/muschellij2/intro_to_r)

Navarro, D. (2020). *Learning statistics with r* (Version 0.6 ed.). (https://learningstatisticswithr.com/)

Thulin, M. (2021). *Modern statistics with r: From wrangling and exploring data to inference and predictive modelling*. Eos Chasma Press. Accessed February 30, 2023. Retrieved from https://www.modernstatisticswithr.com/

Timbers, T., Campbell, T., & Lee, M. (2022). *Data science: A first introduction*. CRC Press. Accessed January 30, 2023. Retrieved from https://datasciencebook.ca/

Venables, W. N., Smith, D. M., & R Core Team. (2022). *An introduction to R: Notes on R: A programming environment for data analysis and graphics* (This manual is for R, version 4.1.3 (2022-03-10) ed.). (http://cran.r-project.org/doc/manuals/R-intro.pdf (retrieved on 2022/04/06))

Wickham, H., & Grolemund, G. (2023). *R for data science (2e)*. Accessed January 30, 2023. Retrieved from https://r4ds.hadley.nz/