# INTRODUCTION TO R

**Introduction to Data Science DSC 105 Fall 2022**

## Table of Contents

Figure 1: RStudio Ball Logo (Source: rstudio.com)

# 1 OVERVIEW

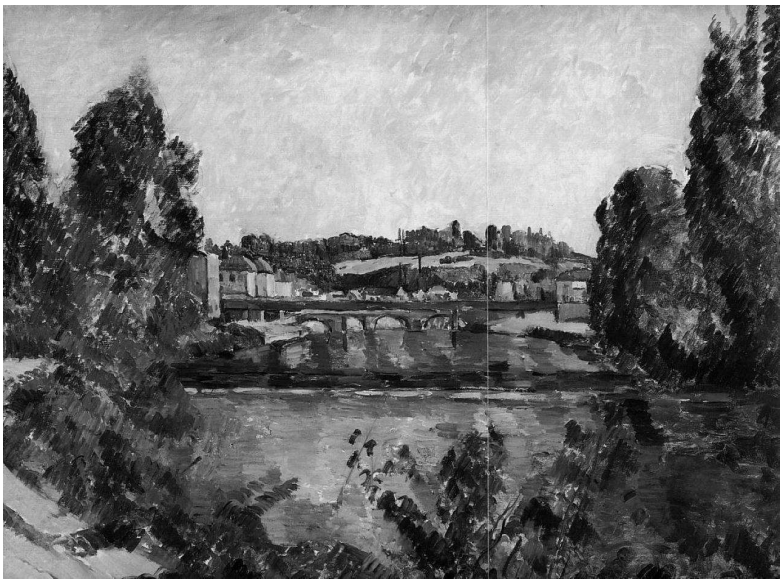


Figure 2: Bridge and Waterfall at Pontoise (Cezanne, 1881)

- Why are we using R?
- Getting in/out of R
- Installing R on Windows and Mac

- R Packages and libraries

  Inspiration and ideas especially from Davies(2016) and other places gratefully received (see references). At the end of some sections, you find challenges - things for you to think about or do something. You find solutions and tips regarding these challenges in a section at the end of the document.

# 2 WHY WE ARE USING R

| Programming Language | 2021 | 2016 | 2011 | 2006 | 2001 | 1996 | 1991 | 1986 |
|---|---|---|---|---|---|---|---|---|
| C | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |
| Java | 2 | 1 | 1 | 1 | 3 | 28 | - | - |
| Python | 3 | 5 | 6 | 7 | 23 | 16 | - | - |
| C++ | 4 | 3 | 3 | 3 | 2 | 2 | 2 | 8 |
| C# | 5 | 4 | 5 | 6 | 9 | - | - | - |
| JavaScript | 6 | 7 | 9 | 9 | 6 | 30 | - | - |
| PHP | 7 | 6 | 4 | 4 | 20 | - | - | - |
| R | 8 | 14 | 35 | - | - | - | - | - |
| SQL | 9 | - | - | - | - | - | - | - |
| Go | 10 | 56 | 15 | - | - | - | - | - |
| Perl | 14 | 8 | 7 | 5 | 4 | 3 | - | - |
| Lisp | 32 | 23 | 12 | 13 | 16 | 7 | 3 | 2 |
| Ada | 34 | 22 | 20 | 15 | 15 | 5 | 9 | 3 |

- One of the 'big three' (Python, R, SQL)
- FOSS and especially open to non-programmers
- Strong on analysis and visualization

Image Source: TIOBE.com/index - Check some of these languages out! 2020-2022, R has dropped again from no. 8 to 12 to 19.

When it comes to "self-service" data analysis, three languages are mentioned most often: R, Python and SQL. All three have their relative merits and issues.

I chose R as the programming language for this introductory course. The choice is partly **personal** and partly **professional**. *Personal:* I like it and it's new for me (I've only taught it since early 2020), so I am still excited about it. It's good if your instructor is excited about the material! *Professional:* as business professionals, you don't want to have to be programmers. At the same time, you need to be able to speak with experts and do and extend your own analyses (not be restricted for example by dashboards).

On a *practical* note, R has a very large, diverse user and developer community. Unlike Python, many of the users do not have a technology background. This means that the "world of R" is more easily accessible if digital technologies and programming aren't your main interests. The SQL community is probably even larger and even more diverse (databases being a more general interest than even statistical analysis), but the language SQL itself is hardly extensible, very focused on querying and less on visualization.

In reality, as a data scientist, or even as a business practitioner with serious, systematic data analytics interests, you need to know all of these - R, SQL, and Python. Here, we'll start with R.

For a direct comparison of Python and R for data cleaning and exploratory analysis with examples, see e.g. Radecic (2020), Uprety (2020) and Shotwell (2020). To see how R outperforms Python, see Grogan (2020). To see some equivalents of SQL in R, check ODSC (2018). And for an overview of data science tools beyond Python, R, and SQL, see Gallatin (2018). And here's a neat infographic from datacamp comparing both for data analysis.

There are downsides to using R as well, of course, and it has been called "hard to learn", too (Muenchen 2017), partly and paradoxically because the language is so flexible and extensible. Also, some innovations, like the Tidyverse, aren't necessarily good for beginners (Matloff 2019).

Of course, there's also always an index - in this case the "TIOBE" index of programming language popularity (based on the languages people search for), see figure 3. As you can see, R improved its position in one year from 20th to 8th. That's by far the strongest improvement of any language among the top 10. Still, Python is three times more search-successful. Neither Python nor SQL have changed their position compared to one year ago. The popularity of R quite likely rides on the popularity of statistics due to the interest in COVID-19 data analysis.

# 3 MATLOFF'S 10 REASONS

```
1. Public domain implementation of S
2. De facto standard among professional statisticians
3. Superior to comparable commercial products
4. Available for Windows, MacOS, and Linux
5. Extensible through library packaging
6. Has OOP and functional programming features
7. Saves data and command history between sessions
8. Has a large and helpful user community
9. Allows for interactive data exploration via command-line
10. Superior graphics capabilities
```

Source: The Art of R Programming (2011)

# 4 OBTAINING AND INSTALLING R FROM CRAN

URL: https://cran.r-project.org/mirrors.html

USA

| | |
|---|---|
| https://mirror.las.iastate.edu/CRAN/ | Iowa State University, Ames, IA |
| http://ftp.ussg.iu.edu/CRAN/ | Indiana University |
| https://rweb.crmda.ku.edu/cran/ | University of Kansas, Lawrence, KS |
| https://repo.miserver.it.umich.edu/cran/ | MBNI, University of Michigan, Ann Arbor, MI |
| http://cran.wustl.edu/ | Washington University, St. Louis, MO |
| https://archive.linux.duke.edu/cran/ | Duke University, Durham, NC |
| https://cran.case.edu/ | Case Western Reserve University, Cleveland, OH |
| https://ftp.osuosl.org/pub/cran/ | Oregon State University |
| http://lib.stat.cmu.edu/R/CRAN/ | Statlib, Carnegie Mellon University, Pittsburgh, PA |
| https://cran.mirrors.hoobly.com/ | Hoobly Classifieds, Pittsburgh, PA |
| https://mirrors.nics.utk.edu/cran/ | National Institute for Computational Sciences, Oak Ridge, TN |
| https://cran.microsoft.com/ | Revolution Analytics, Dallas, TX |

- CRAN = "Comprehensive R Archive Network"x at r-project.org
- Use *mirror sites* (**what's that?**) for download
- PRACTICE: **DOWNLOAD THE *INSTALLER* PROGRAM FOR YOUR OPERATING SYSTEM**

Download the installer for your operating system from your local CRAN ("**Comprehensive R Archive Network**") mirror here: https://cran.r-project.org/mirrors.html.

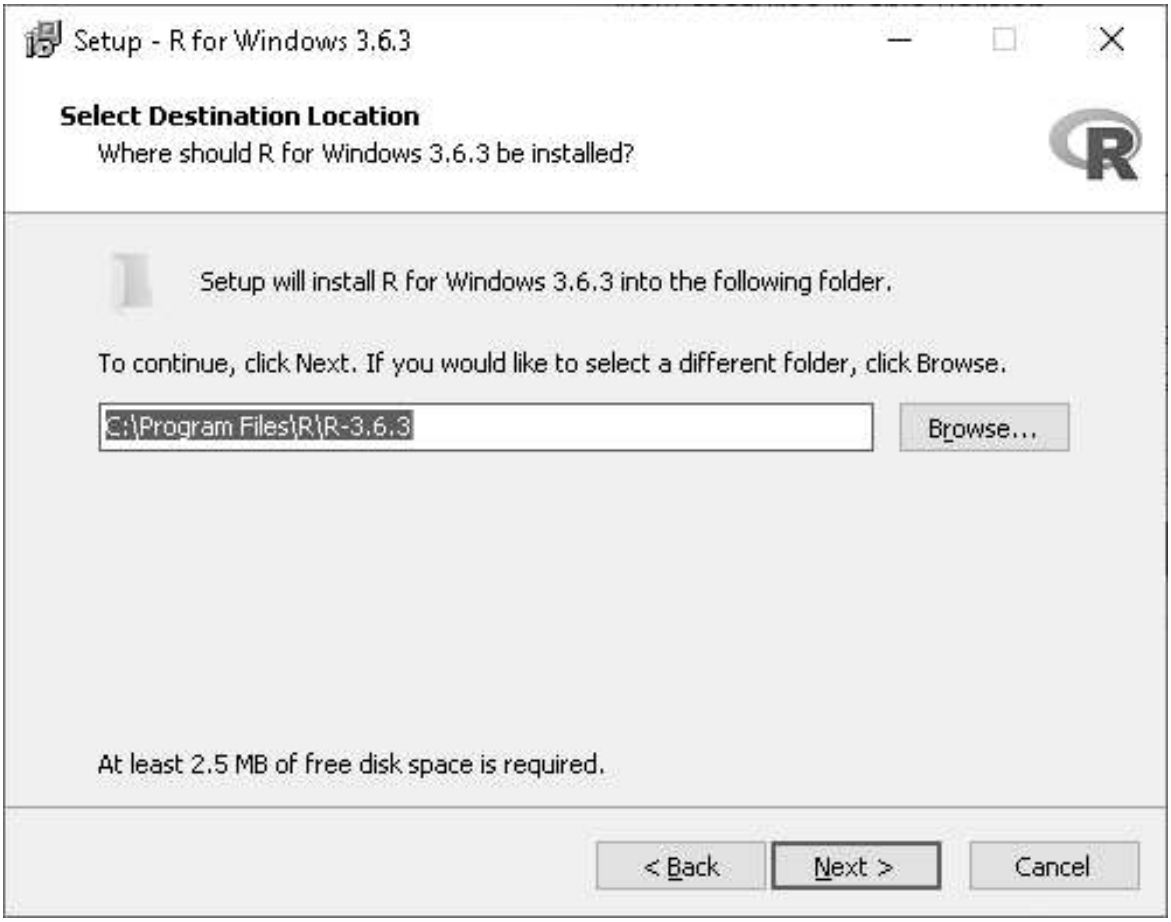
For example, if you are in Berlin, the Nürnberg server is closest: https://ftp.fau.de/cran/.

**Challenge:** Which server would you use if you were in Russia? Does the download page for that server look any different? Check it out! (Hint)

USA: notice that the TX server is at "revolutionanalytics.com", which used to be another R IDE bought by Microsoft. Microsoft embraced R so fiercely that they even started their own subset of it, Microsoft R Open, which you can get from MRAN (Microsoft R Application Network). **Can you discern the strategy here?** You can get it by reading this series of news flashes from Microsoft.

**Which other open source related platforms are now Microsoft?** Answer: GitHub

# 5 HOW THIS LOOKS UNDER WINDOWS




I tried this on Lenovo and Dell laptops running Windows 10 and it worked:

1. After opening the `R..win.exe` file, a popup asks you if you will let this pogram modify your hard disk. Say "yes" (why is this necessary?[1])

2. In the installation dialog, accept all settings and check the options for establishing a desktop shortcut and a quick launch icon.
3. The location of your R program files will be `C:\Program Files\R`. Once the installation is finished, you should have an icon on your desktop named `Rx64 4.0.2` (or whatever your version is).
4. Double click it to open the R console for the first time. At the `>` prompt, type `1+1` and `RETURN` to see if R can compute. Then type `demo(graphics)` and hit `RETURN` ("`Enter`") repeatedly to see a few R plots.
5. I also switched from my integrated (default) graphics card to a "High Performance NVIDIA" graphics card (which I did not know I had!).
6. To leave, type `q()` at the prompt or leave with the `File > Exit` graphical menu. When asked if you wish to save the workspace, say "no".

7. When installing a program, a dialog was opened offering me to install packages in a local folder (accept this with "yes").

   See this datacamp blog post (March 11, 2020) for installation instruction for Windows, MacOS X and Ubuntu (Linux).

---

*(If you have other troubles with R + MacOS, let me know. I have a Mac available and may be able to figure something out.)*

# 6 HOW THIS LOOKS ON A MAC


New installation & reconfiguration (2020)

I did this on a MacMini (2014) running MacOS 10.13.6 without too many problems (see below). Essentially the only problem occurred when trying to install packages (discussed later) and I could fix it easily by changing a system setting.

1. To download and install R for MacOS, go to `r-project.org`, and click on `CRAN` right below the `Download` headline. The CRAN mirror page opens. Scroll down to find a German mirror site and click to download the `.DMG` installer file, which will install the program.

    1. There were system-level error messages though the program installed alright. But I could not install CRAN packages because of this error: `tar: Failed to set default locale`. This refers to a problem with the `tar` unzip program. I checked `stackoverflow.com` and found a fix that in turn directed me back to a CRAN helpfile with lots (too much, really) information for Mac users.

        1. To fix the problem, close R, open a terminal and type: ~defaults

        write org.R-project.R force.LANG en$_{US.UTF}$-8~. Then restart R and the problem should have disappeared (it did for me and never came back).

        See also this datacamp blog post (March 11, 2020) for installation instruction for Windows, MacOS X and Ubuntu (Linux).

        ---

        *(If you have other troubles with R + MacOS, let me know. I have a Mac available and may be able to figure something out.)*

# 7 TODO PRACTICE: DOWNLOAD PRACTICE FILES


1. Open the course directory in GitHub, https://github.com/birkenkrahe/ds1
2. Open `/org/3_practice.org`
3. Open the `raw` version of the file
4. Save file as `3_practice.org`
5. Right click on the file in Explorer
6. Change `Opens with:` property to Emacs
7. Open file with Emacs from the Explorer

Summary:

# 8 TODO PRACTICE: INSTALL R

- Windows people: help each other!
- MacOS people: help each other!
- Linux people: you're good!

# 9 OPENING R FOR THE FIRST TIME

- R is an *interpreted language* - instructions are entered via CLI[2]
- On your windows box, you have R as a terminal program and as a GUI
- In Emacs, you can execute R code blocks and display graphs together

# 10 TODO PRACTICE: FIND R / RUN R SCRIPTS

Summary:

- R is an interpreted program with a shell (CLI/console)
- On Windows, there are a GUI and a terminal program
- You can run R scripts in the foreground or in the background

# 11 R SHELL: VERSION AND PLATFORM

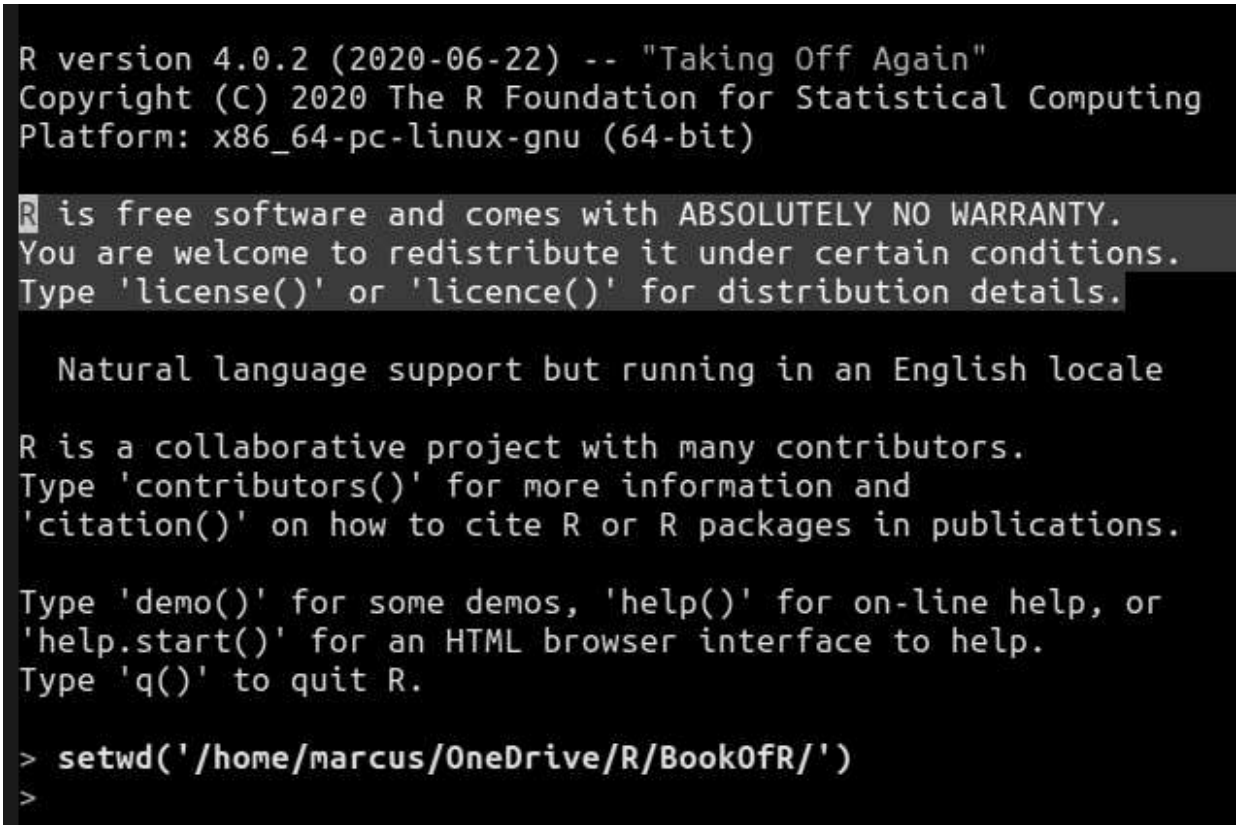
```
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

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.

> setwd('/home/marcus/OneDrive/R/BookOfR/')
>
```

What type of bit-architecture do you have?

This is the first screen you see (figure 11) after starting R on the command-line. The highlighted section shows the current (June 2020) version of Base-R, as the core R program is officially called. Versions get their own names, like operating systems (my Ubuntu Linux operating system e.g. has the version number 18.04-LTS and the name "Bionic Beaver"). R 4.0.2 is also called "Taking Off Again". Lastly, the platform of the operating system on which the R program runs, is shown - a 64-bit version of Linux using the x86 computer architecture.

**Challenge:** what type of computer architecture does your computer have (most importantly: 64-bit)? (Hint)

# 12 R SHELL: DISTRIBUTION LICENSE




Type `license()`. What is "GNU"?

As you'll find out when following the instructions in figure 12 by entering `license()` at the prompt, the R software is distributed "under the terms of the GNU General Public License" (GPL). Popular software also distributed under the GPL include the Linux "kernel" (the core of the operating system), and the GNU compiler collection. You may have heard of the term "open source", which essentially means the same thing, though one may quibble (and people do, a lot). What's important to remember: use of the GPL (= making R "free software") has contributed enormously to the success of this language.

**Challenge:** what is "GNU software" exactly? Which programs belong to it? Are there any programs that you have used before? (Hint)

# 13 R SHELL: THE R PROJECT

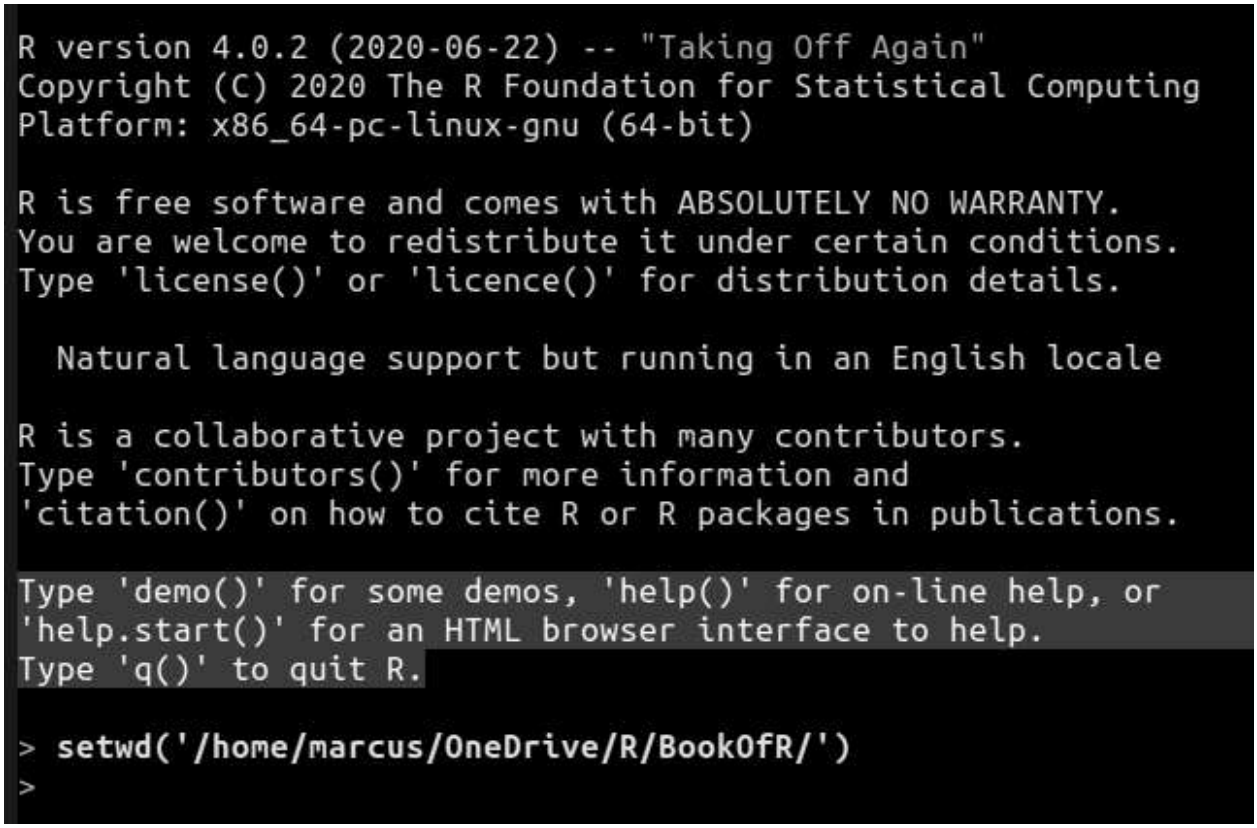
```
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

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.

> setwd('/home/marcus/OneDrive/R/BookOfR/')
>
```

- Enter `citation()`. Why cite software?
- Enter `contributors()`. Who can contribute?

  Behind R is a large project of volunteers (figure 13. At it centre is the "R Core Group" of developers. Because R is part of the "GNU suite" of programs, and because its predecessor was called S, it is also sometimes called "GNU S". Becker (2004) has written an interesting historical account of S. When using R for analysis in a thesis, a paper, an essay or a blog post, one should cite it as a source. This is what the code `citation()` is for. Same goes for specific packages (more on this later) like "`data.table`" that are not part of Base-R. The citation alternatives may also prompt you to check out `LaTeX` and `BibTeX`, which are quasi-standards for the professional (and beautiful!) formatting of scientific papers.

  **Challenge:** is there any connection between R and LaTeX? Or more general between the programming language R und markup languages (like HTML or LaTeX)? (Hint)

# 14 R SHELL: DEMO AND HELP




- Enter `demo(graphics)` and marvel.
- Enter `help.start()` - where is this page?

  The section higlighted in figure 14 suggests a few commands that you ought to try for yourself:

  `help()` is a function to get help for whatever you put in between the brackets. A quick win is `help(help)`, or help about the help function. The format of the help pages is borrowed from the Unix man[ual] pages. An alternative to `help()` is `?` followed by the term you need help with, e.g. `?help`, which is the same as `help(help)` but much shorter. Lastly, `help.start()` opens a browser window with help in HTML format. Very useful access to a wealth of systematic information. If you don't know the exact name, you can also search across all documentation using `help.search()` or the shortcut `??`. Try entering `??cars` if you are looking for datasets on cars. You'll find that there are four known datasets with cars in different packages.

  Via the dataset search, you can also find out that functions like `help()` or `demo()` are part of the `utils` package - respective functions are listed as `utils::[function]`. It contains all sorts of functions for housekeeping and administration.

  The R help system is however not written for beginners. Personally, I more often go to textbooks or, preferably, to stackoverflow.com if I have a question or need to remind myself of a command or a way of doing things.

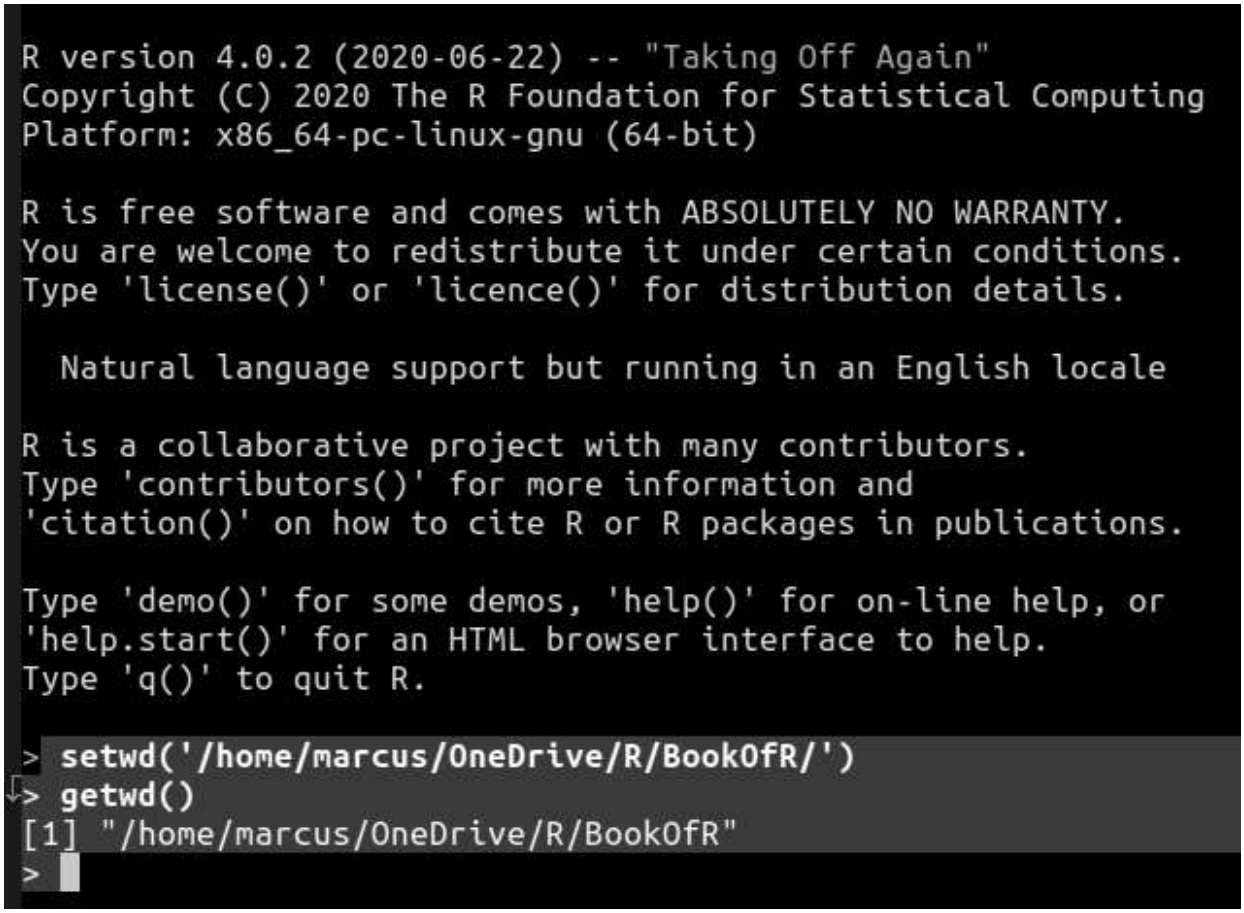
  There are a few interactive demo programs available, too. You should try `demo(graphics)` and marvel at the various possibilities of R to create plots with your data. Notice how few lines of code are sufficient to create great effects! The window that opens when you execute the demo commands is the standard graphics output when using R in command-line mode.

# 15 TODO PRACTICE: EXPLORING THE R SHELL

Summary:

- Contents of the R startup screen: R's version, license, project, citation, how to get help and demos, and how to quit R
- GPL is the GNU Public License (important for FOSS)
- BibTeX and LaTeX for scientific document processing

# 16 WORKING DIRECTORY




- Enter `getwd()` ("get working dir")
- Use `setwd()` to change directory

  When you start R, you may be asked, which working directory you wish to use. This is where all files created (e.g. plots) will be put and where R will look first to load scripts with R commands for execution.

  The `setwd()` command in figure 15 allows you to set any directory as working directory. To check which one is used right now, you can use `getwd()`.

  How you specify the path to the current working directory depends on your operating system, e.g. `/home/marcus` for my home directory on MacOS/Linux, or `C:\Users\Marcus` under Windows. Especially as a Windows user, you should look at your file organisation - this will pay off as soon as you use the terminal or command-line. The Bash shell that I use on my Linux computer (and that most MacOS users will use) is also available within Windows 10 (Posey 2018).

# 17 TODO PRACTICE: CHANGE WORKING DIRECTORY

Summary:

- Function without arguments: `getwd()`
- Function with arguments: `setwd('...')`
- Absolute pathname like `'c:/Users/birkenkrahe/'`
- Relative pathnames like `'../../'` ("go up by 2 levels")

# 18 THE R SHELL PROMPT

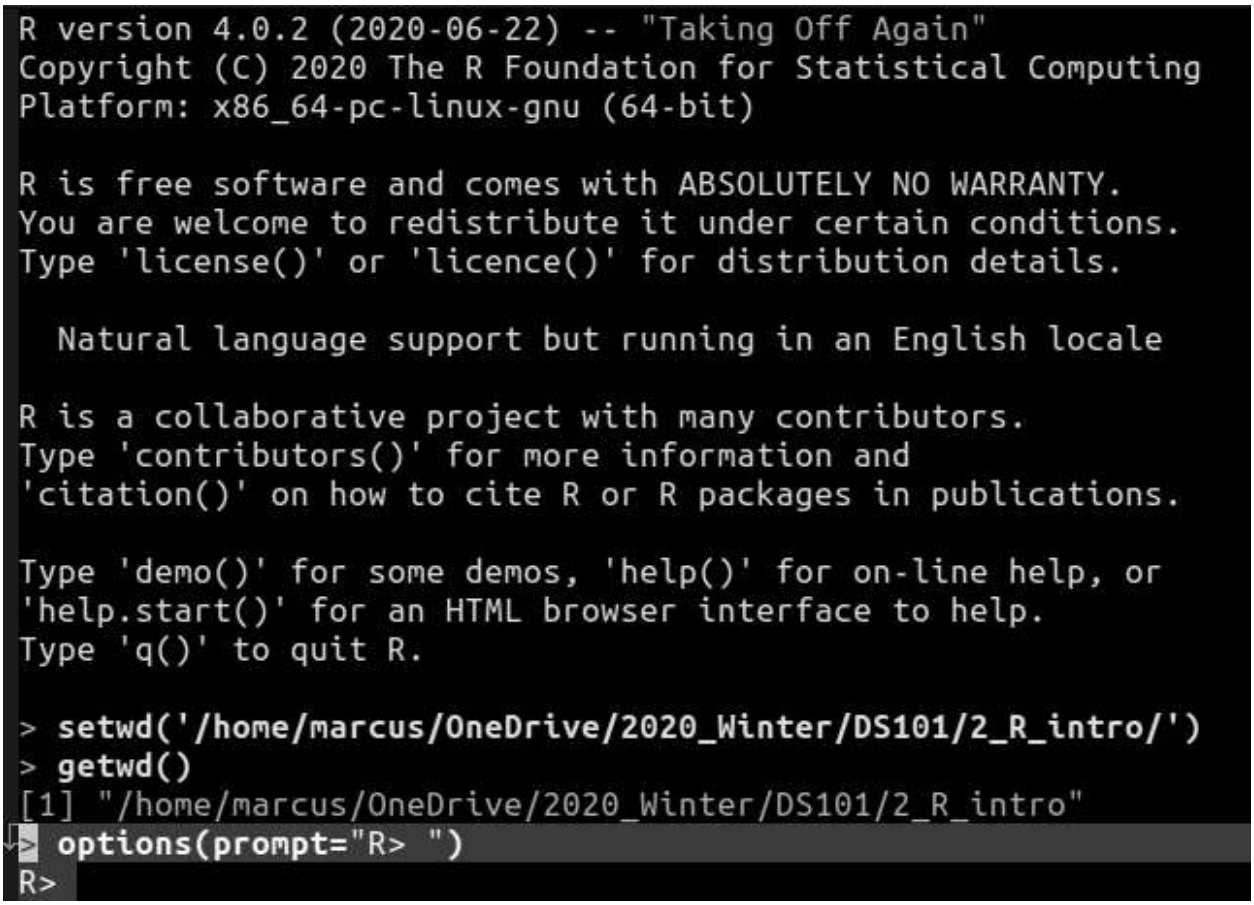


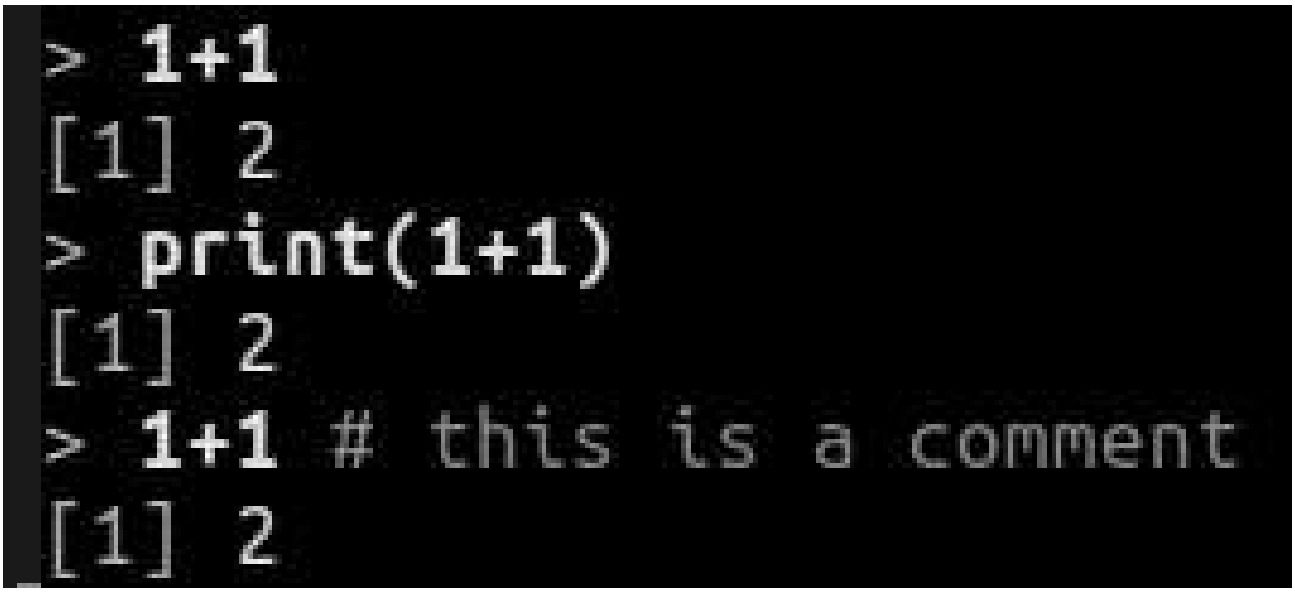

Figure 16 shows a new utility command, `options()`, that you can use to change the identifying prompt at the beginning of the command line. You don't have to do this but it's nice to know that and how you can do it. One of the advantages of working on the command-line is that you experience how you can adapt your working environment to your personal needs - something that most graphical environments do not allow you do to (at least not without a lot more effort). Freedom of extensibility is the name of the command-line game.

# 19 TODO PRACTICE: CHANGE R SHELL PROMPT

Summary:

- The functions `options` controls display options
- You can extract display options with `$`, e.g. `options()$prompt`
- You can get help with the `help` function (or `?`)

# 20 COMPUTING AND COMMENTING




One of the advantages of the interactive command-line is the ability to perform arithmetic operations. In figure 17 we begin with a simple addition. We'll do a lot more of this in the next section. When you type the command and click `ENTER`, R responds by printing out the result without the need to explicit instruct it using a `print` command (though as you can see, this works as well). You also see here that `#` is the R sign for a comment (which is ignored upon execution). The ominous `[1]` at the beginning of each output line indicates the number of columns printed. R does this because it is strongest when manipulating tabular data - data ordered in columns and rows.

# 21 TODO PRACTICE: COMPUTE AND COMMENT

Summary:

- You can print results with or without `print`
- Create (inline) comments with `#`
- `eshell` is a Linux-type shell in Emacs (doc)

# 22 R packages

- Packages contain functions and data sets
- Most packages must be installed and loaded first
- Default data sets are pre-loaded: `?datasets`

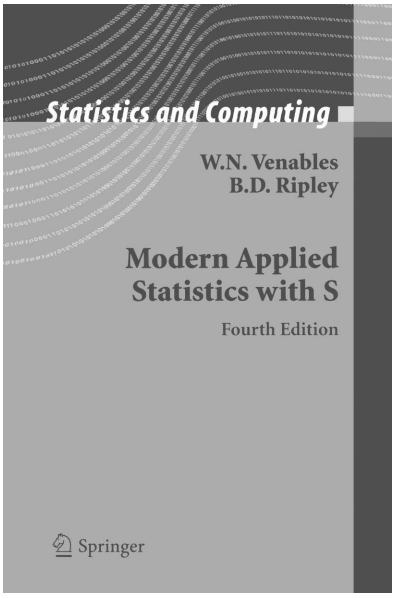



Figure 18: MASS is from the book by Venables/Ripley (2002)

# 23 INSTALL PACKAGES

```
> install.packages("MASS")
Installing package into ‘/home/marcus/R/x86_64-pc-linux-gnu-library/4.1’
(as ‘lib’ is unspecified)
trying URL 'https://ftp.fau.de/cran/src/contrib/MASS_7.3-54.tar.gz'
Content type 'application/x-gzip' length 506246 bytes (494 KB)
==================================================
downloaded 494 KB
```

- To install package "`MASS`": enter `install.packages("MASS")`
- Installation includes identifying location on your computer
- Installation downloads compressed *tarball* from a CRAN mirror site
- `md5sum` is a GNU utility program that checks correct file transfer

# 24 MISCELLANEOUS PACKAGE COMMANDS

- To uninstall a package, use `remove.packages(package="[pkgname]")`
- To see all installed packages: `installed.packages()`

- To update packages: `update.packages()` (this can take a while)

```
> update.packages(package="MASS")
Warning: package 'MASS' in library 'C:/Program Files/R/R-4.1.3/library' will not be updated
blob :
 Version 1.2.2 installed in C:/Users/birkenkrahe/R/win-library/4.1
 Version 1.2.3 available at https://mirrors.nics.utk.edu/cran
Update? (Yes/no/cancel)
```

Figure 20: Updating the R package MASS (R session screenshot)

- For a short package description: `packageDescription("...")`
- To see all datasets in a package: `data(package="...")`
- `data()` will list all datasets for all installed packages
- To load a package into current R session only: `library("...")`
- For a list of currently loaded packages: `search()`

- For a list of search paths (to find pkgs): `searchpaths()`

```
> searchpaths()
 [1] ".GlobalEnv"
 [2] "C:/Users/birkenkrahe/R/win-library/4.1/MASS"
 [3] "ESSR"
 [4] "C:/Program Files/R/R-4.1.3/library/stats"
 [5] "C:/Program Files/R/R-4.1.3/library/graphics"
 [6] "C:/Program Files/R/R-4.1.3/library/grDevices"
 [7] "C:/Program Files/R/R-4.1.3/library/utils"
 [8] "C:/Program Files/R/R-4.1.3/library/datasets"
 [9] "C:/Program Files/R/R-4.1.3/library/methods"
[10] "Autoloads"
[11] "c:/PROGRA~1/R/R-41~1.3/library/base"
>
```

Figure 21: Search paths for R packages on my Windows box

# 25 LOAD DATASETS

- After loading a package that contains data sets, you must load them
- To load a data set contained in package, use `data([name])`.

- You can (often) get help on datasets with `?` or `help([name])`[3]

```
> library(MASS)
> data(phones)
> ls()
[1] "phones"
> rm(list=ls())
> ls()
character(0)
>
```

Figure 22: Loading MASS, MASS::phones, listing and delisting

# 26 EXPLORE DATA

- When you've loaded a data set, you should take a look at it
- Most useful: `str` to see the data structure, `head` and `tail` to see the first and last few rows
- These functions have many different attributes (check the help)

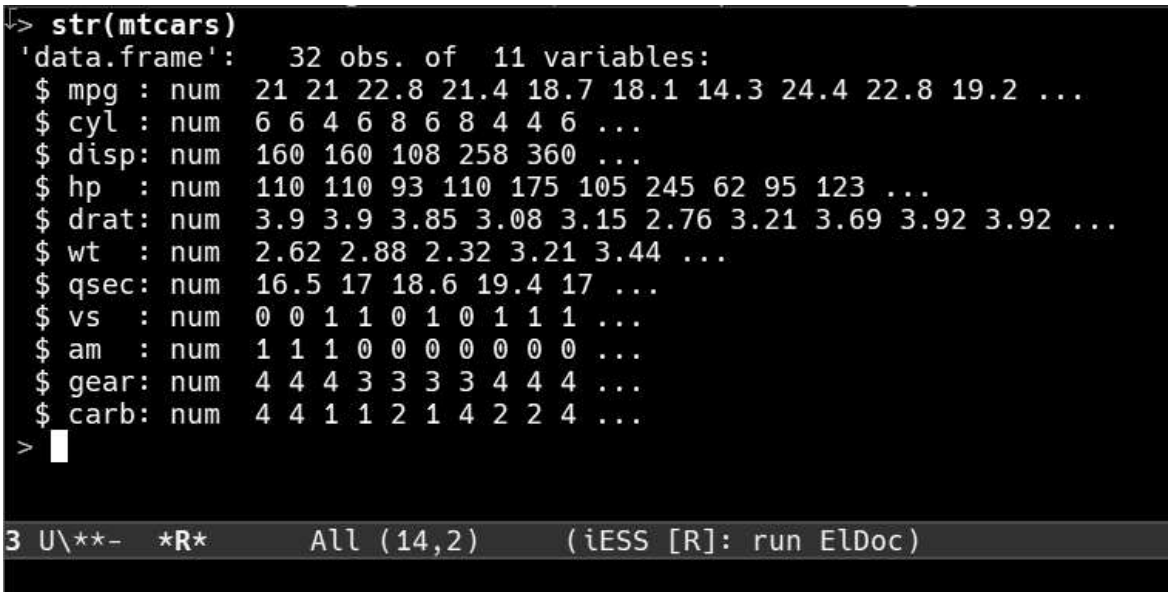



Figure 23: structure of the built-in data set mtcars

# 27 TODO PRACTICE: R PACKAGE COMMANDS

Summary:

- You can install, uninstall packages and data sets in them
- You must load packages and data sets before using them
- Your current R session keeps track of all loaded objects
- Display structure, head and tail of loaded data sets

# 28 SAVING YOUR WORKSPACE

- When you quit an R session with `q()` or `quit()`, you're asked if you want to save the *workspace image*.
- The workspace image includes all objects that were defined in the session, like loaded libraries, datasets, variables etc.
- In the current directory, R saves your command history (in a readable text file `.Rhistory`), and all data (in a machine-readable file `.RData`).

```
> q()
Save workspace image? [y/n/c]: y

Process R finished at Thu Jul  8 20:25:09 2021
```

# 29 CUSTOMIZING AT STARTUP

- When you install packages, you do not need administrative rights, even if R is installed in a read-only portion of your computer. The OS will offer you to install packages in a user directory.
- When downloading the package as part of the installation or updating process, Windows forces you to pick a mirror. You can disable this by creating your own `~/.Rprofile` file and specifying a download mirror.

```
/home/marcus/OneDrive/2021_Fall/ds101:
total used in directory 13596 available 240.9 GiB
drwxrwxr-x  6 marcus marcus     4096 Jul  8 20:25 .
-rw-rw-r--  1 marcus marcus     2564 Jul  8 20:25 .Rhistory
-rw-rw-r--  1 marcus marcus 12891303 Jul  8 20:25 .RData
```

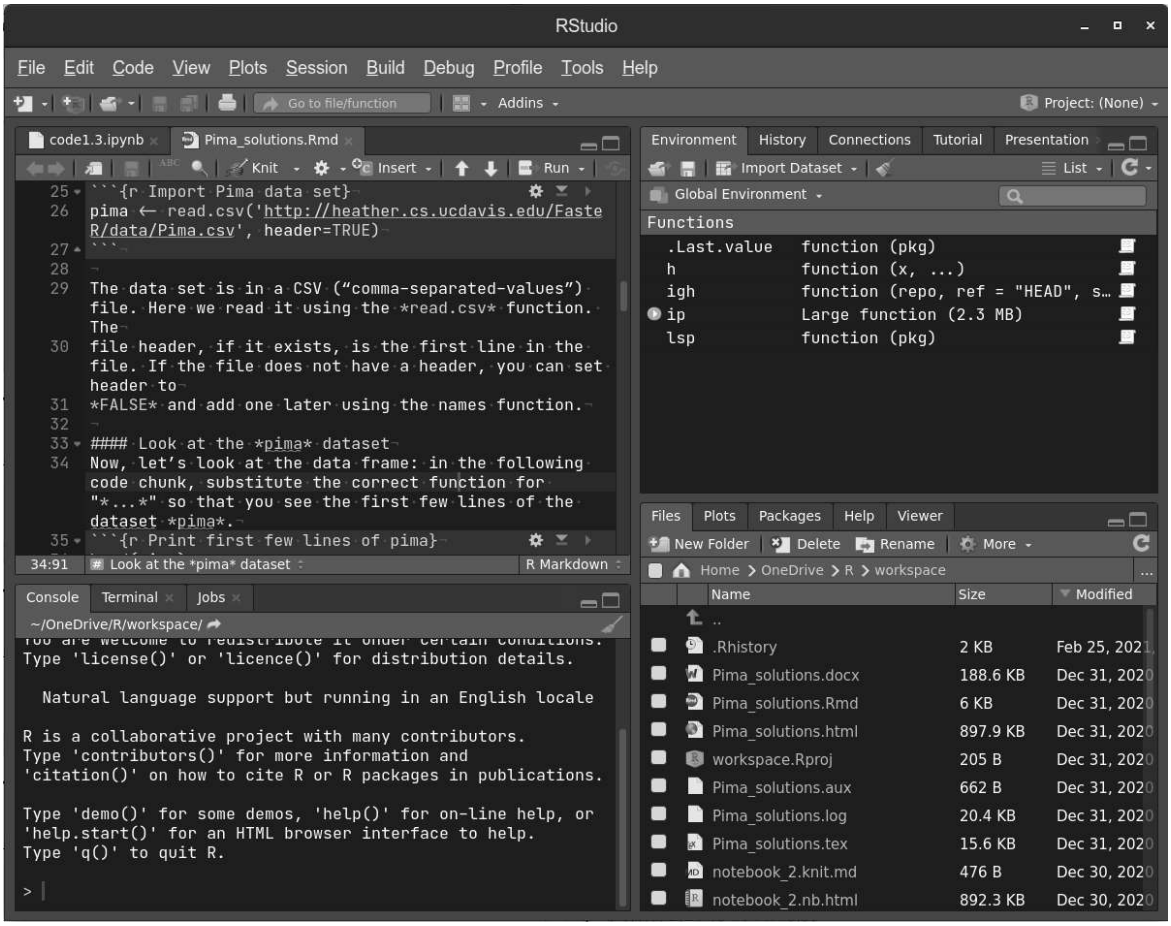
- Saved R commands: `.Rhistory`
- Saved R variables: `.RData`
- R profile settings: `.Rprofile`
- See also: "Fun with .Rprofile and customizing R startup" (Fischetti, 2014)

# 30 TODO PRACTICE: CUSTOMIZING AT STARTUP

Summary:

- Emacs and R have a home directory (`~/`) for startup files[4]
- You can determine R's startup behavior in `~/.Rprofile`
- `~/.Rprofile` is read every time a new R shell is started

# 31 The RStudio IDE




- RStudio is a popular (FOSS) IDE for R with literate programming capabilities (it supports interactive R Notebooks)
- We're not using RStudio (why) but Emacs + ESS + Org-mode instead
- You can download RStudio from here - perhaps you learn to like it[5]

# 32 Concept Summary

- R is an easy to **learn** language to quickly and interactively analyse datasets. R is especially strong on visualization.
- R can be downloaded from `r-project.org` and installed on your computer.
- There is plenty of **help** on R available from within the program, or on the Internet using the wider community of practitioners.
- When you open R, you establish a working **environment**, which includes packages, functions and variables.

# 33 Code summary

| TERM | MEANING |
|---|---|
| `license()`, `licence()` | License info |
| `help()`, `?help` | get help |
| `??[name]` | check occurrences |

| TERM | MEANING |
|---|---|
| `demo()` | R demos |
| `getwd()`, `setwd()` | get/set working dir |
| `options(prompt=)` | set prompt |
| `options(repos=)` | set download repo |
| `options()$prompt` | display prompt |
| `options()$repos` | display download repo |
| `print(1+1)` | result of `1+1` |
| `quit()`, `q()` | leave R |
| `# ...` | comment |
| `library("MASS")` | load |
| `detach("package:[name]")` | unload package |
| `install.packages("MASS")` | install |
| `installed.packages()` | list all packages |
| `update.packages()` | update |
| `packageDescription("MASS")` | describe |
| `help(package="MASS")` | show |
| `data()` | built-in datasets |
| `search()` | list loaded pkgs |
| `searchpaths()` | list pkg search paths |
| `ls()` | list loaded objects |
| `rm(list=ls())` | unload objects |

# 34 What next?

Figure 27: HAL 9000 interface (Kubrick's 2001 Space Odyssey)

See also: HAL 9000: "I'm sorry Dave, I'm afraid I can't do that."

# 35 What now? read!

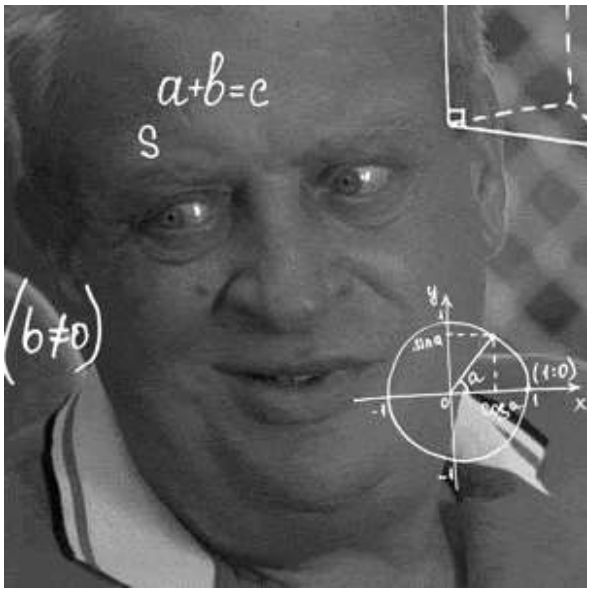
- Read frequently and widely
- Go both deep and stay shallow: You've seen that I don't just cite peer-reviewed papers but blog posts, too. The truth is that I have personally learnt a lot more from them than from scientific papers. However, this is partly a function of my experience and skill. Without these, it might be hard to distinguish what's good and bad - just like when you google any topic you don't know anything about yet. But even if you're a bloody beginner, I recommend reading widely and both deeply (with a lot of focus, e.g. when looking up terms, repeating analyses and retyping code) and shallowly (skimming articles, reading comments), because you build an associative network of terms, arguments and practices. I follow a bunch of data science experts on Twitter for the same reason. If you do this for any topic that is being discussed on a factual (rather than an overly political or emotional) basis, you'll learn more faster[6].
- For example: take a look at "R Weekly" for a weekly, curated collection of articles from the R community. This will give you an idea of the spread of information.

# 36 What now? play!

Read: Data Scientists Should Learn Through Play

To understand why you should play (see figure 29), check the article by an active blogger and professional in the R-blogosphere, Keith McNulty, who leads data science at the global strategy consulting firm McKinsey & Co. He argues that "learning through playing around" with the software is a good way to learn (McNulty 2020) - I agree. Though I am often distracted by having to create teaching material for you, playing around on or off the command-line, looking at interesting data and combing through them using the analytical tools R offers, or checking other people's plots or inferences, is the most fun way of learning R. There's nothing wrong with reading or working through a course, watching teaching videos, of course, either. #+end$_{notes}$

# 37 What's the next topic?




Arithmetic with R

# 38 References


- Adolfo Alvarez (25 Mar 2019). R Packages: A Beginner's Guide. Online: datacamp.com.
- Robert Becker (2004). A Brief History of S. Online: sas.waterloo.ca.
- Tilman M. Davies (2016). The Book of R. No Starch Press.
- Tony Fischetti (September 17, 2014). Fun with .Rprofile and customizing R startup. URL: R-bloggers.com.
- Kyle Gallatin (1 Nov 2018). Some Important Data Science Tools that aren't Python, R, SQL or Math. Online: towardsdatascience.com.
- Michael Grogan (23 Jul 2020). How R Still Excels Compared To Python. Online: towardsdatascience.com.
- Knuth D (1992). Literate Programming. Stanford, Center for the Study of Language and Information Lecture Notes 27.
- Norman Matloff (2019). TidyverseSceptic. Online: github.com.
- Keith McNulty (23 Jun 2020). Data Scientists Should Learn Through Play. Online: drkeithmcnulty.com.
- Robert A. Muenchen (2017). Why R is Hard to Learn. Online: r4stats.com.
- Brien Posey (5 Feb 2018). How To Navigate the File System in Windows 10's Bash Shell. Online: redmondmag.com.
- Dario Radecic (10 Sept 2020). Trying R for the First Time. Online: towardsdatascience.com.
- Gordon Shotwell (30 Dec 2019). Why I use R. Online: blog.shotwell.ca.
- Sagar Uprety (23 Jul 2020). Data Cleaning and Exploratory Analysis in Python and R. Online: towardsdatascience.com.
- Venables/Ripley (2002). Modern Applied Statistics with S. Springer. Online: researchgate.net.
- Yuleng Zeng (28 Aug 2018). An Introduction to R and LaTeX. Online: bookdown.org.


# 39 Hints

## 39.1 Download from CRAN

Mirror sites are called that way because they are actual identical copies of the original site. The quality of the cloned page is monitored. The result looks interesting (to me). You can see how well maintained a particular mirror site is.

## 39.2 Opening R for the first time

The projects listed here (by no means a complete list!) are divided in applications and infrastructure projects. **Applications** of R include bioinformatics (e.g. in the medical sciences or in genomics), geospatial statistics (anything related to maps), and finance (R is strong with this one!). **Infrastructure** includes incorporation of R in Wikis (like Wikipedia) - for example to generate plots on the fly - and ESS ("Emacs Speaks Statistics"), which is the interface to the extensible text editor that I'm using (e.g. to create all documentation for this course - essentially from one text file). An alternative to ESS is the highly popular IDE (Integrated Development Environment) RStudio. We will not be using it in this course but I encourage you to check it out, try it and see if you like it, especially if my teaching tempo is too slow for you!

## 39.3 Version and platform

See here to find out details of your CPU and computer architecture for Windows or MacOS.

## 39.4 Distribution license

Go to GNU Software to see a list of all programs distributed under the GPL. These programs constitute the GNU system of free software. Looking through the list, I noticed the following programs that I have used: Chess (chess game implementation), Emacs (extensible text editor that I am using in this very moment), Gimp (image manipulation), Gnome (desktop for my operating system, Ubuntu Linux), and so on…425 programs are listed here alone (29 Aug 2020).

### 39.5 The R Project

There is no special connection between LaTeX and R, except that both are free software programs, one for formatting (especially when mathematical formulas need to be presented), the other one for statistical calculations and visualisation. However, to communicate data analysis results and to make the analysis process itself reproducible, a combination between these two goals (formatting/programming) is desirable. This is exactly what "literate programming" (Knuth 1984) does. There is also a program called "R Markdown" to create documents that enables you e.g. to created HTML, PDF, ePUB and Kindle books with only one source. You can find examples at bookdown.org. See also Zeng (2018) for a brief introduction to both R and LateX - sufficient to get started - written apparently as a minimal example for bookdown. For LaTeX there are also cloud editors like overleaf.com.

### 39.6 R Packages

You can directly search for this dataset - I usually take the search string "`r doc [name]`, in this case `r doc MASS boston`, which gets me straight to this page. At the top, you can read that "The `Boston` data frame has 506 rows and 14 columns". There's also an R Notebook, which shows various aspects of this dataset.

Another way to find the answer is by using the command `str()` that you already know: `str(Boston` contains the answer in the first line - as long as `MASS` has been loaded. (Check out what happens if not by closing the R session with `q()` (don't save the workspace) and reopening it again.

The simplest way is to type `help(Boston)` (again, only after loading the `MASS` package).

# Footnotes:

[1] To open the R console, and direct plots to the correct device, the R program needs to be "plugged into" your operating system, as it were. You could still run it otherwise but e.g. you'd have to always type the exact program path.

[2] In fact, you can also save R instructions as a script and then run them using the program `Rscript` or in batch mode with the command `R CMD BATCH`. We'll practice these commands in class.

[3] Strictly speaking, the availability of help depends on the package design - well written packages and data sets are well documented and are accompanied by short and detailed descriptions, or even papers (so-called "vignettes"). An example is the `Rcpp` package that interfaces R and C++.

[4] You can also re-set this home directory - this FAQ explains how.

[5]

[6] Data science is a mixed affair when it comes to this last tip: because of the importance of statistics and models for COVID-19, public discussions e.g. on Twitter are often instantly politicized and emotionally charged. However, to be able to navigate these waters and still extract the common good, is an important ability that is, for me, also part of "data literacy". Learning how to read and discern different views, focus on facts and problem-solving, while not ignoring the wider problem setting, is my working definition of the scientific method.

Author: R Installation and First Steps
Created: 2022-07-31 Sun 19:40