An Introduction to R

Why Learn to Code?

Julien Martin
University of Ottawa

Learning outcomes

this course (hopefully) will ...

- introduce you to using R ✔️
- help your research become more robust and reproducible ✔️
- show you how to import your data ✔️
- show you how to summarise, visualise and analyse your data ✔️

this course (definitely) won't ...

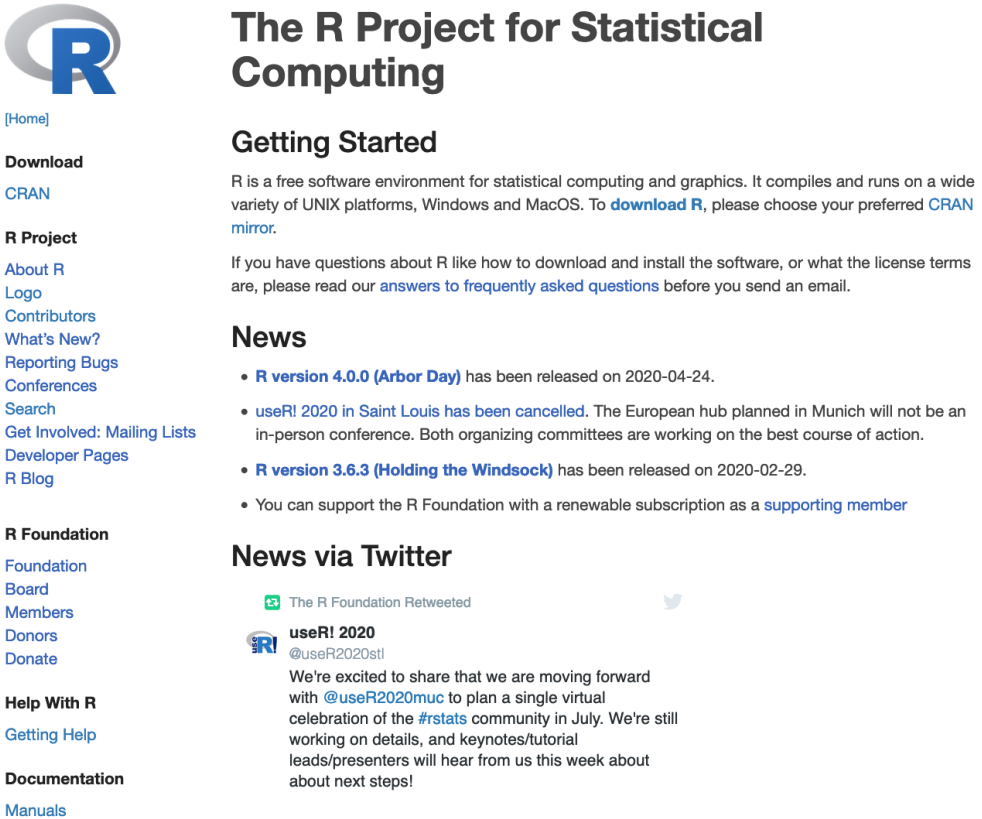
- teach you everything there is to know about R ❌
- make you feel completely comfortable with R ❌

What is R?

- answer depends on who you ask and what they use R for
- environment for statistical computing, graphics and programming
- originally created by Ross Ihaka and Robert Gentleman (1996)
- currently maintained by international R-core development team
- many, many people also contribute to R and the wider R community

extra R information

- more information can be found at https://www.r-project.com
- download R from the CRAN website https://cran.r-project.org/



why is R considered hard?

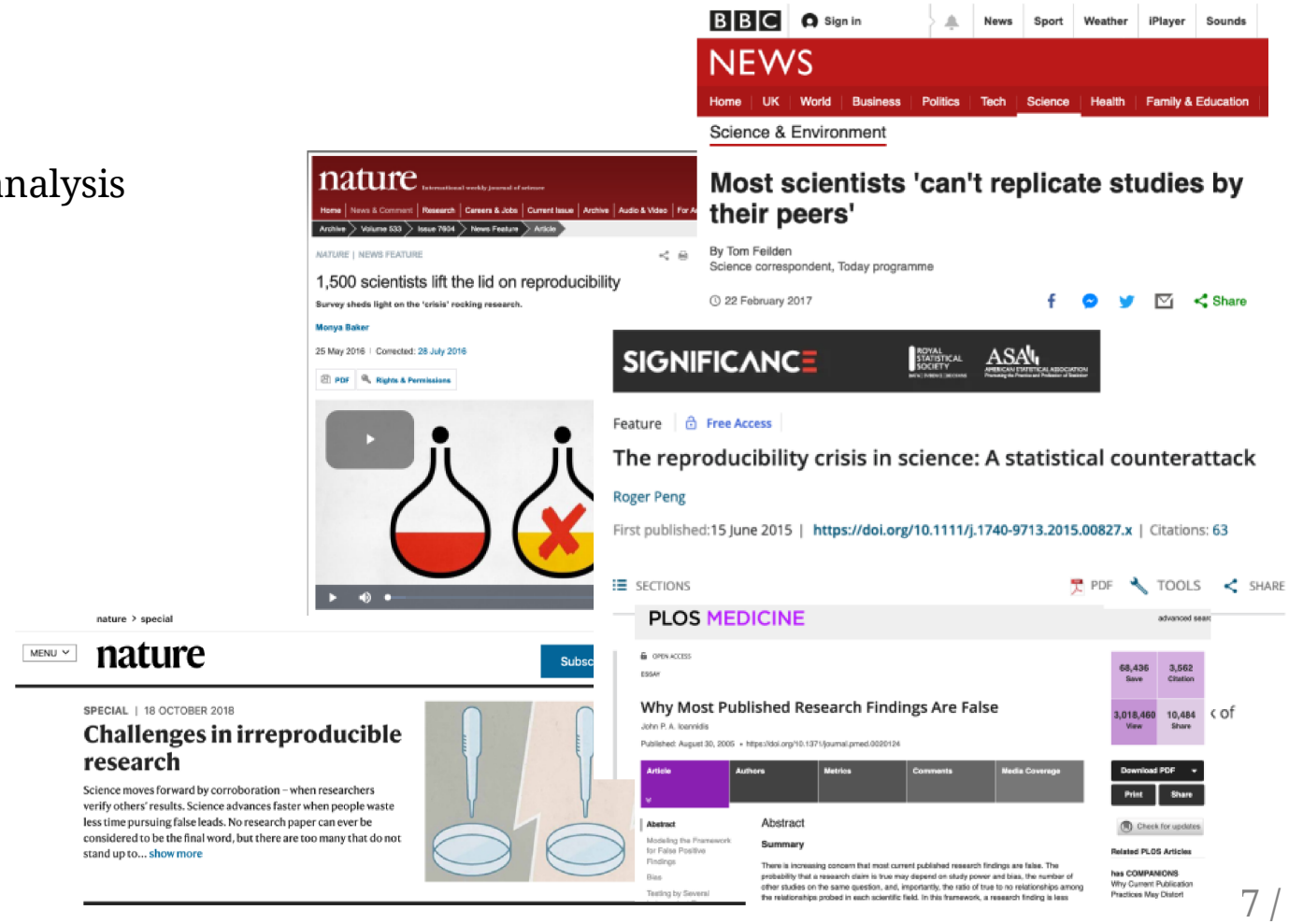
- high initial investment in time to learn R
- unfamiliar command line environment
- frustrating (and sometimes) inconsistent syntax
- analysis and figures can take longer (initially!)
- relatively steep learning curve
- it's not actually that hard, just unfamiliar

why you should learn R

- it’s free and platform independent
- it’s the software of choice for many students, academics, industries and charities worldwide
- highly flexible and extensive
- encourages you think about your research questions, data and analyses

why you should learn R

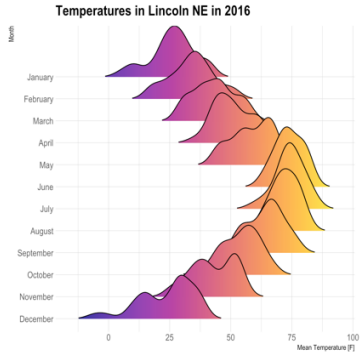
- it allows you to keep an exact and reproducible record of your analyses
- transparent
- other people* can reproduce your analysis
- easily share your code (GitHub)
- open science

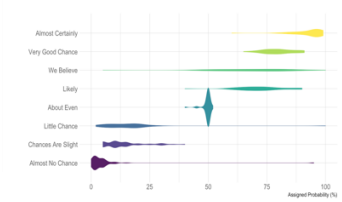


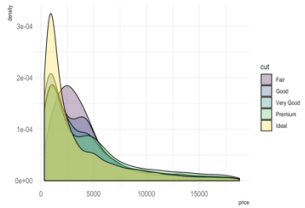
* including the future you!

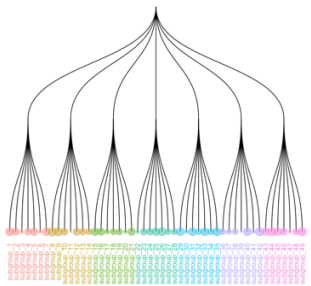
why you should learn R

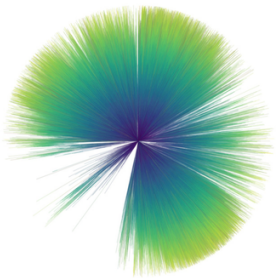
- excellent graphics capabilities

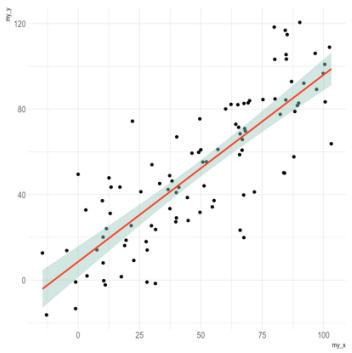


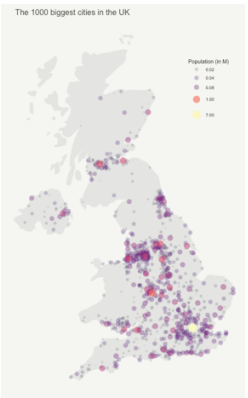










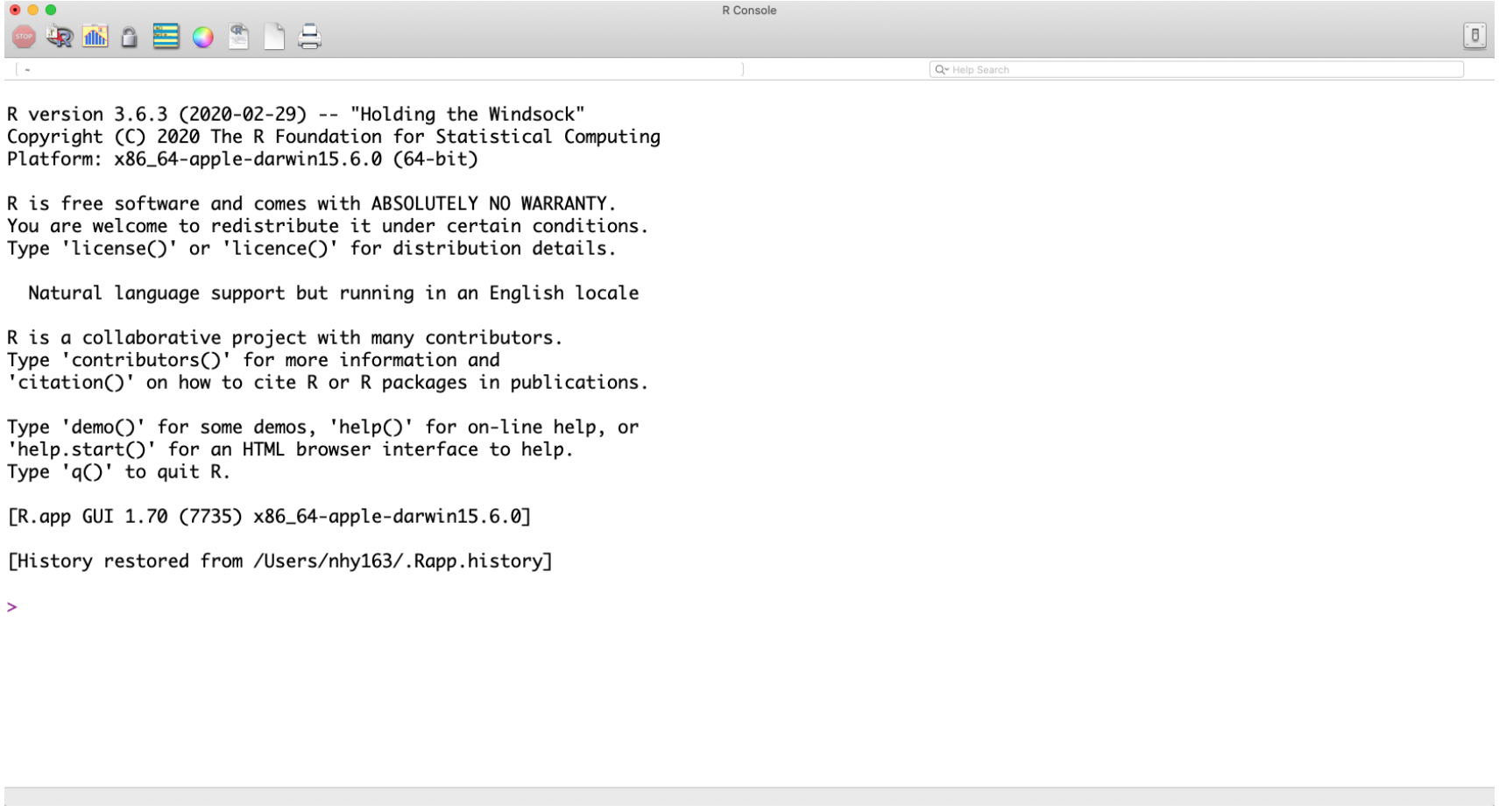


ggplot2

why you should learn R

- employability
- opportunity to get involved with a fantastic and supportive community

using R - GUI


R Console
Help Search
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (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.
[R.app GUI 1.70 (7735) x86_64-apple-darwin15.6.0]
[History restored from /Users/nhy163/.Rapp.history]
>

R

RStudio

using R - RStudio

RStudio

Go to file/function | Addins | Project: (None)

Untitled1* | Source on Save | Run | Source

R script

```
2  ```{r setup, include=FALSE}
3  options(htmltools.dir.version = FALSE)
4  ```
5
6  ```{r xaringan-themer, include=FALSE, warning=FALSE}
7  library(xaringanthemer)
8  style_mono_accent(
9    base_color = "#1F4257",
10   header_font_google = google_font("Josefin Sans"),
11   text_font_google   = google_font("Montserrat", "300", "300i"),
12   code_font_google   = google_font("Droid Mono")
13 )
14 ```
15
16 class: title-slide, center, middle
17
18 # `r rmarkdown::metadata$title`
```

20:36 (Top Level) R Script

Environment | History | Connections | Tutorial

Import Dataset | Grid | Global Environment

Environment/History

Name	Type	Length	Size	Value
opar	list	1	392 B	List of 1
pie.sales	numeric	6	664 B	Named num [1:6] 0.12 0.3 0.26 0.…
pin	numeric	2	64 B	num [1:2] 5.68 1.81
scale	numeric	1	56 B	0.00302460317460317
usr	numeric	4	80 B	num [1:4] -573 1453 -14 634
x	numeric	87	744 B	num [1:87] 10 20 30 40 50 60 70 …
xadd	numeric	1	56 B	508.41511414327
xdelta	numeric	1	56 B	860
xscale	numeric	1	56 B	0.00660077519379845
xx	integer	202	856 B	int [1:202] 0 1 2 3 4 5 6 7 8 9 …
y	numeric	61	536 B	num [1:61] 10 20 30 40 50 60 70 …
yadd	numeric	1	56 B	0
ydelta	numeric	1	56 B	600
yscale	numeric	1	56 B	0.00302460317460317

Console | Terminal | Jobs

~/

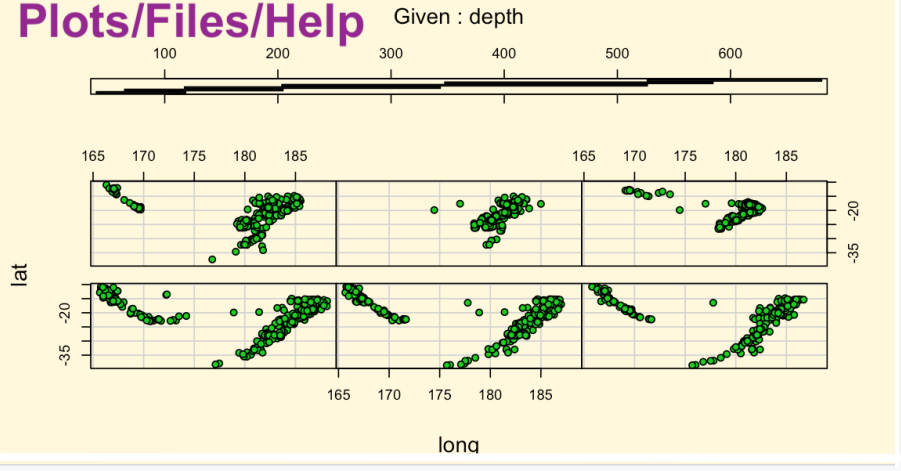
Console

```
> contour(x, y, volcano, levels = lev, col="yellow", lty="solid", add=TRUE)

> box()

> title("A Topographic Map of Maunga Whau", font= 4)

> title(xlab = "Meters North", ylab = "Meters West", font= 3)

> mtext("10 Meter Contour Spacing", side=3, line=0.35, outer=FALSE,
+       at = mean(par("usr")[1:2]), cex=0.7, font=3)

> ## Conditioning plots
>
> par(bg="cornsilk")

> coplot(lat ~ long | depth, data = quakes, pch = 21, bg = "green3")
Hit <Return> to see next plot:

> par(opar)
>
```

Files | Plots | Help | Viewer

Zoom | Export | Publish

Plots/Files/Help



using R - objects

- commands are typed in the editor and then sourced into the console
  at the > prompt

```
> 2 + 2
## [1] 4
```

- R is object orientated. You can create variable and assign value(s) to them

```
> a <- 2 + 2
> a
## [1] 4
```

using R - objects

- once created, operations can be performed on variables

```
> a <- 2 + 2
> b <- 3 * 2
>
> a + b
## [1] 10
```

- very powerful and flexible

using R - functions

- functions contain a set of instructions that allow you to perform a specific task(s)
- you can use functions that are inbuilt in R

```
> numbers <- c(2, 3, 4, 5, 6)   # use the concatenate function
> numbers
## [1] 2 3 4 5 6
>
> mean(numbers)                 # use mean function
## [1] 4
>
> var(numbers)                  # use variance function
## [1] 2.5
```

- comments (#) are ignored by R

using R - functions

- or you can install user contributed packages to increase versatility and power
- there are packages for almost anything

19,665
indexed packages

2,996,148
indexed functions

- install packages from CRAN, Bioconductor and GitHub
- packages are easy to install in R

RDocumentation (2019)

using R - functions

- or write your own functions
- example: function to calculate standard error

```
> se.fnc <- function(x){          # start function for se
+  std.x <- sd(x)                      # calculate SD
+  nos.x <- length(x)                  # calculate number obs
+  se.x <- std.x/(sqrt(nos.x))    # calculate SE
+  print(se.x)
+ }
>
> se.fnc(c(2, 3, 4, 5, 6)) # use your new function
## [1] 0.7071068
```

using R - syntax

- R is case sensitive `A` is not the same as `a`
- commands are generally separated by a new line, but you can also use a `;` (rare)
- anything that follows the hash symbol (#) will be ignored by R. Use this to comment your code
- a series of commands can be grouped using braces `{ }`
- write pretty code by following a code style guide

using R - help

- R has extensive help facilities
- when you need help for a function use `help()` or `?`

```
> help("plot")    # open help file for the function plot
```

or equivalently

```
> ?plot
```

- or to search across all help files use `help.search("plot")` or `??plot`

using R - help

- Duckduckgo_it: Try online search of any error messages you get. It's not cheating and everyone does it! You'll be surprised how many other people have probably had the same problem and solved it.
- Stack Overflow: There are many thousands of questions relevant to R on Stack Overflow
- Here are the most popular ones, ranked by vote. Make sure you search for similar questions before asking your own
- make sure you include a reproducible example to get the most useful advice. A reproducible example is a minimal example that lets others who are trying to help you to see the error themselves.

tips for learning R

- use R often and use it regularly
- learning R is not a memory test, you have your R scripts
- you don’t need to know everything there is to know about R to use it productively
- don't stare at code for hours.
- there are many ways to tackle a particular problem. See what others have done
- when stuck, and frustrated don't throw your computer by the window

Thanks!

I created these slides with xaringan and R Markdown using the rutgers css that I slightly modified.

Credit: I borrowed slides from Alex Douglas.