

MH2010

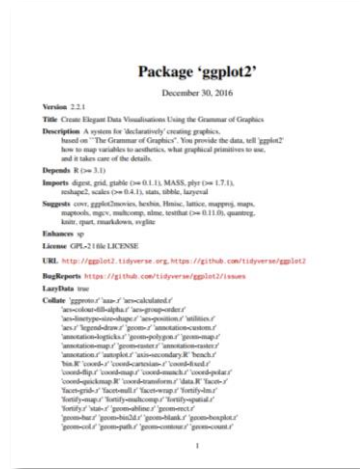
Practical session 29th Jan 2020

- Introduction to RStudio

Brendan Palmer,
Clinical Research Facility - Cork &
School of Public Health
 @B_A_Palmer

R is for Resources

Vignettes



Webpages



eBooks



Cheatsheets



Twitter



Mara Averick
@dataandme

tidyverse 🍷 @rstudio, 🧘 hoop head,
gnashgab, blatherskite, lesser 1/2 of
@batpigandme 🇺🇸 🇩🇪

📍 Massachusetts



One R Tip a Day
@RLangTip

One tip per day M-F on the R
programming language #rstats. Brought
to you by the R community team at
Microsoft.



Hadley Wickham 🌱
@hadleywickham

R, data, visualisation.

📍 Houston, TX

🌐 hadley.nz



David Robinson
@drob

Data Scientist at @StackOverflow, #rstats
fan/evangelist

📍 New York, NY

🌐 varianceexplained.org



Jenny Bryan
@JennyBryan

Software engineer @rstudio, humane
#rstats, adjunct prof @UBC where I
created @STAT545, part of @ropensci

📍 Vancouver, BC

🌐 jennybryan.org



Darren L Dahly
@statsepi Follows you

Principal Statistician, Epidemiologist, Sr
Lecturer | @HRBIreland Clinical Research
Facility @CRF_CORK | Cork #Rstats Users
Group meetup.com/Cork-Ireland-R...

📍 Cork, Ireland

🌐 darrendahly.github.io



Data Scientists IRL
@DataSci_Ireland Follows you

Promoting the Data Science professions
in Ireland.

📍 Ireland

🌐 facebook.com/DataScientists...

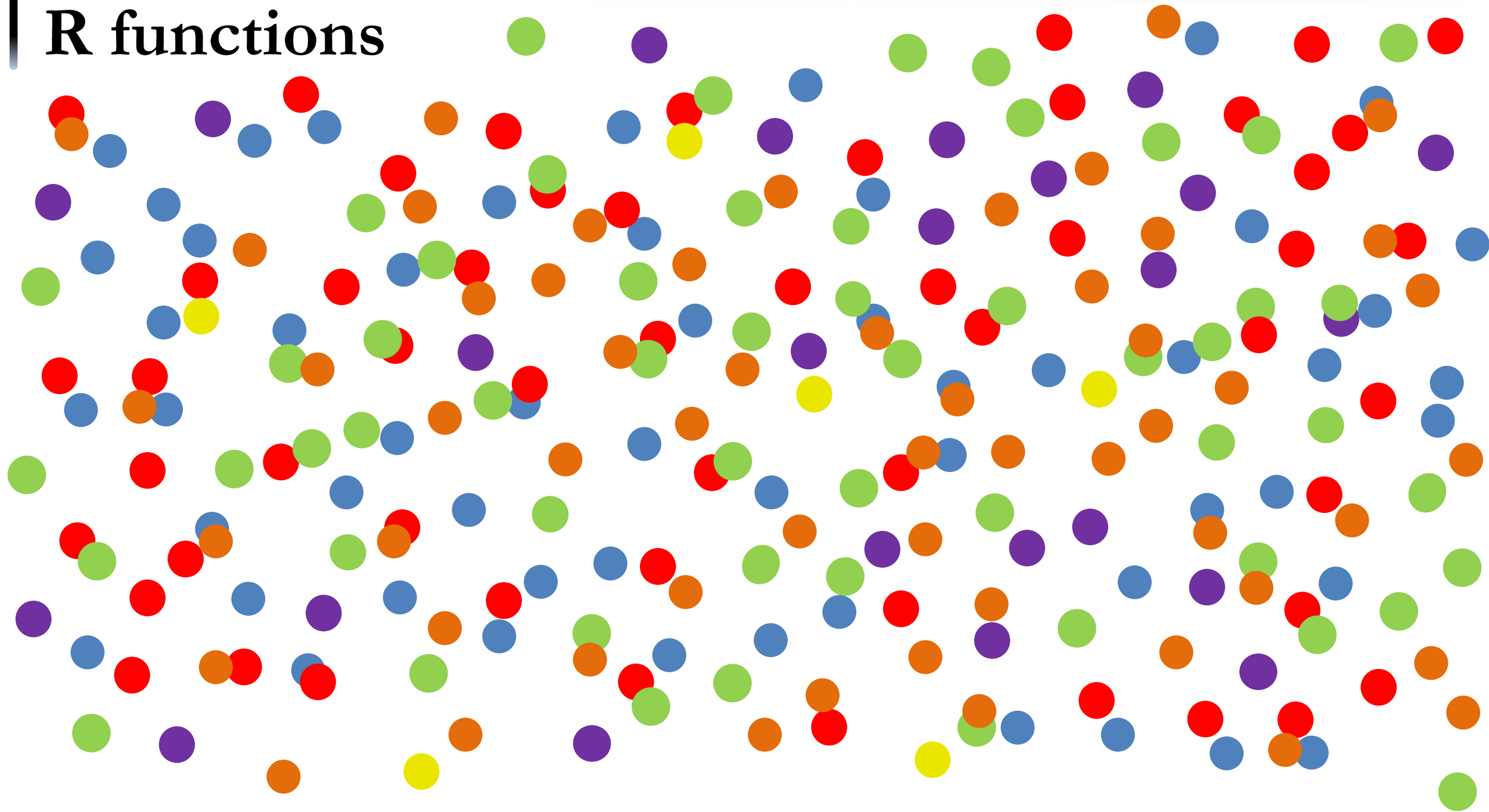


Kara Woo
@kara_woo

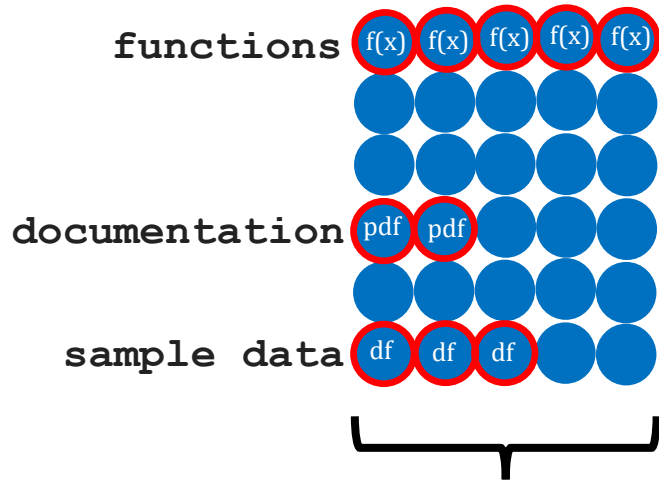
Research scientist at @sagebio. Data
curation, visualization, #rstats,
reproducibility, open science, ballet

🌐 karawoo.com

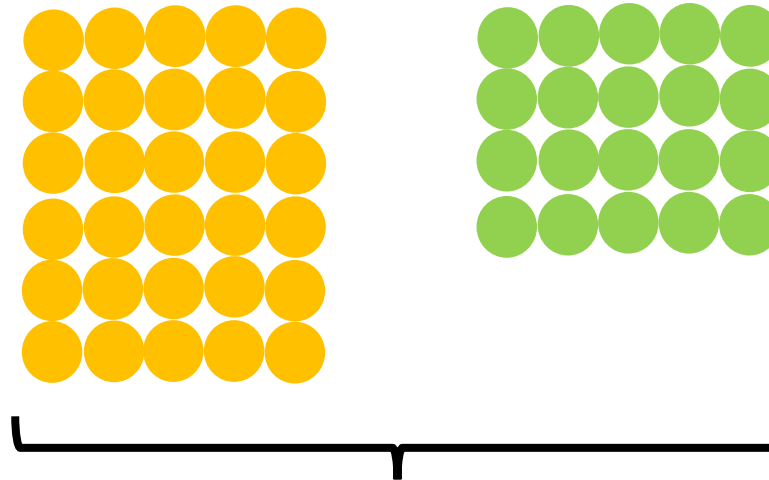
R functions



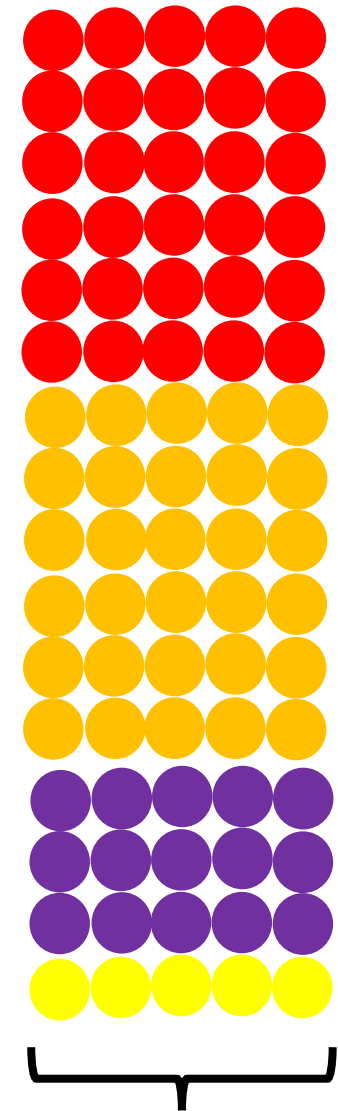
R packages



R comes pre-loaded with ~30 other packages (e.g. base, stats, graphics etc.)

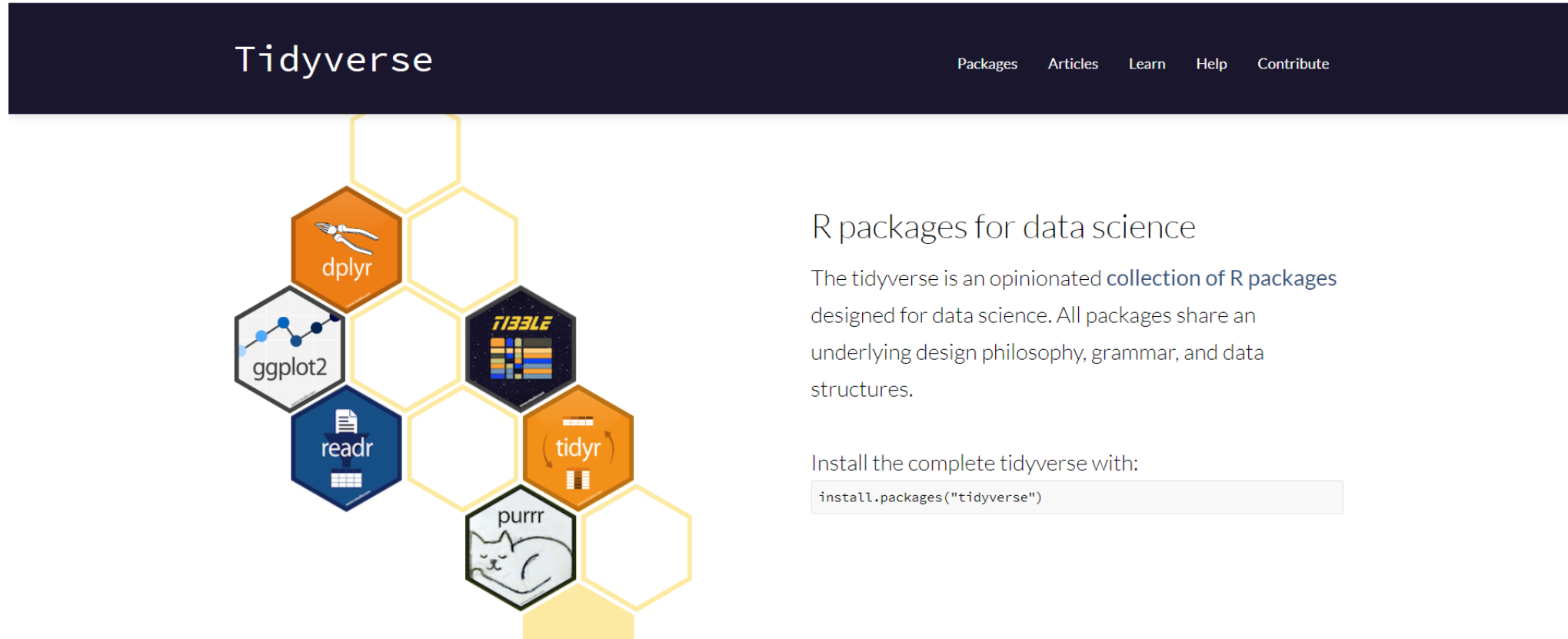


Other packages:
Install once
Update regularly
Load each session



tidyverse

What is the tidyverse?

A screenshot of the Tidyverse website. The header is dark blue with the word "Tidyverse" in white. To the right of the header are links for "Packages", "Articles", "Learn", "Help", and "Contribute". Below the header is a large graphic of a honeycomb grid. Several hexagons are filled with icons and package names: "dplyr" (orange, top left), "ggplot2" (grey, middle left), "readr" (blue, bottom left), "tidyr" (orange, middle right), "purrr" (grey, bottom right), and "TIBBLE" (dark blue, top right). To the right of the honeycomb graphic, the text "R packages for data science" is followed by a paragraph: "The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures." Below this is the text "Install the complete tidyverse with:" followed by a code block containing the command `install.packages("tidyverse")`.

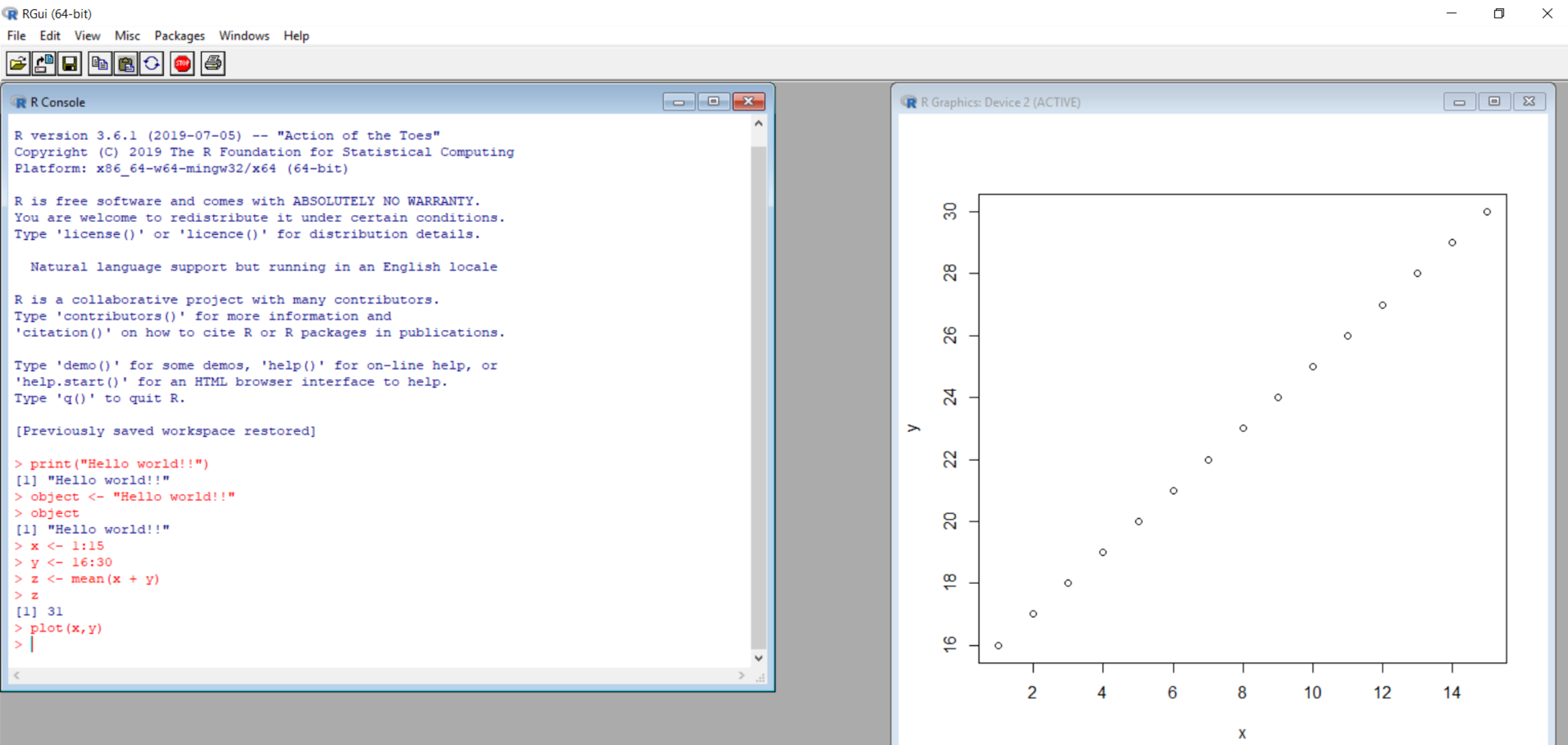
- Joined up collection of packages for data analysis
 - Consistent functions
 - Uses (tidy) data
 - Supports end-to-end workflows

What is the tidyverse?

```
> install.packages(c("broom", "cli2", "crayon",  
"dbplyr", "dplyr", "forcats", "ggplot2", "haven",  
"hms", "httr", "jsonlite", "lubridate",  
"magrittr", "modelr", "pillar", "purrr", "readr",  
"readxl", "reprex", "rlang", "rstudioapi",  
"rvest", "stringr", "tibble", "tidyr", "xml2"))
```

```
> install.packages("tidyverse")
```

R user interface versus RStudio



RStudio Cloud



[Log In](#)

[Sign Up](#)



Welcome to RStudio Cloud^{alpha}

Do, share, teach and learn data science with R.

[Get Started](#)

If you already have an RStudio shinyapps.io account, you can log in using your existing credentials.

[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

Set status

Brendan Palmer

bapalmer

[Edit profile](#)

Twitter: @B_A_Palmer

<https://crfcsdau.github.io/about/>

Organizations



[Overview](#) **[Repositories 14](#)** [Projects 0](#) [Stars 0](#) [Followers 10](#) [Following 10](#)

[Type: All](#)[Language: All](#)

[R-A_Hitchhikers_Guide_to_Reproducible_Research](#)



A 3-day R course given in University College Cork that encompasses RStudio projects, an introduction to the R tidyverse language, project workflows and reporting using R Markdown.

MIT License Updated 6 days ago

[lunchtime_sessions](#)



Short 1 hour introductions to R-related topics such as creating R projects, using GitHub through RStudio and more

HTML 1 MIT License Updated 12 days ago

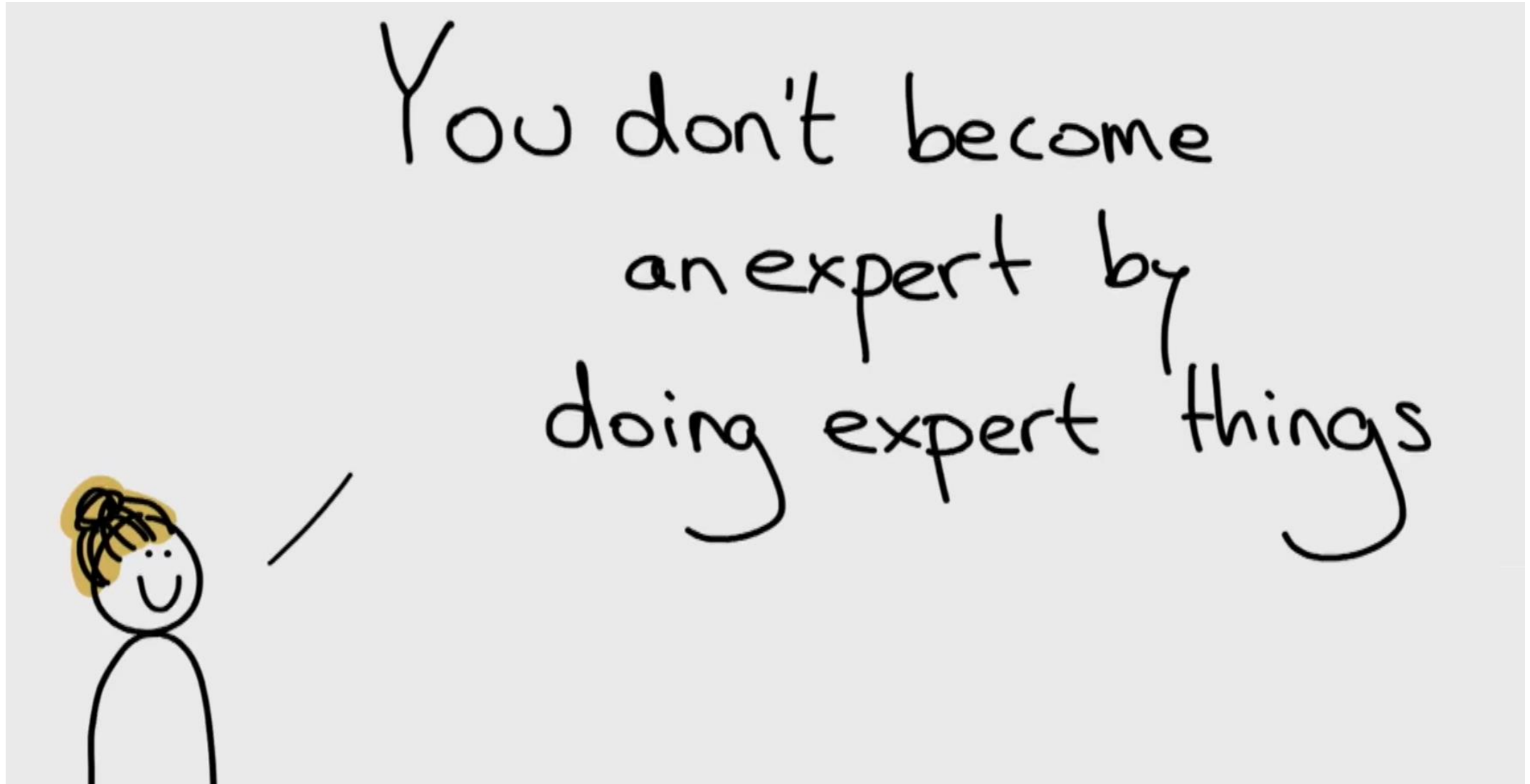
[RCR](#)



Section of the UCC Reproducible Conduct of Research digital badge dedicated to exposing researchers to reproducible research practices.

HTML MIT License Updated 23 days ago

Central theme of this session



test_script.R

Source on Save

```

31
32 gather(sample, expression, G0.05:U0.3) %>%
33
34 separate(sample, c("nutrient", "rate"), sep = 1, convert = TRUE) %>%
35
36 mutate(nutrient = plyr::revalue(nutrient, nutrient_names)) %>%
37
38 filter(!is.na(expression), systematic_name != "")
39
40 # Plot the clean data
41
42 cleaned_genes_tbl %>%
43
44 filter(BP == "leucine biosynthesis") %>%
45
46 ggplot(mapping = aes(x = rate, y = expression, color = nutrient)) +
47   geom_point() +
48   geom_smooth(method = "lm", se = FALSE) +
49   facet_wrap(~ name)

```

49:21 Does this work on your system? ↕

```

/R_Users_Workshop/PG_module/course_notes/R-A_Hitchhikers_Guide_to_Reproducible_Research/Pre-workshop/
GID = col_character(),
YORF = col_character(),
NAME = col_character()

```

see spec(...) for full column specifications.

```
> cleaned_genes_tbl %>%
```

```
  filter(BP == "leucine biosynthesis") %>%
```

```

ggplot(mapping = aes(x = rate, y = expression, color = nutrient)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  facet_wrap(~ name)

```

>

Code editor

R console

Environment History Connections Git

Import Dataset

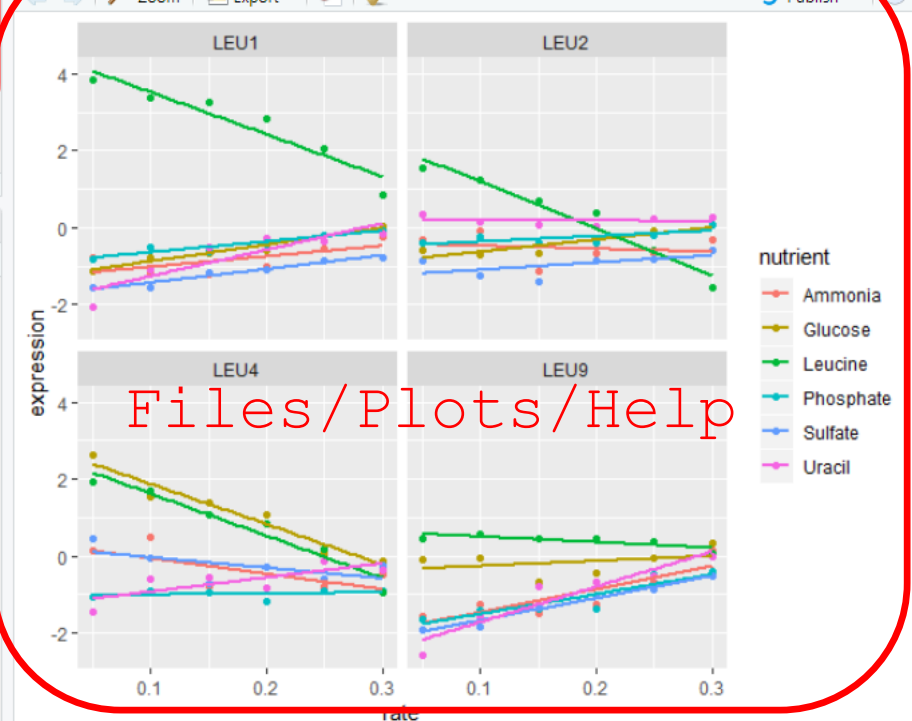
Global Environment

<input type="checkbox"/>	Name	Type	Length	Size	Value
<input type="checkbox"/>	cleaned_g...	tbl_df	7	11.3 ...	198430 obs. of 7...
<input type="checkbox"/>	nutrient_...	charac...	6	984 B	Named chr [1:6] "G...
<input type="checkbox"/>	url	charac...	1	168 B	"http://varianceex...

Environment

Files Plots Packages Help Viewer

Zoom Export



Files/Plots/Help

R Markdown

```
~/Open_Science/Digital_Badge/RCR - master - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function Addins
lettuce_report.Rmd
1 ---
2 title: "This is a reproducible document"
3 author: "Dr. Brendan Palmer"
4 date: "18th June 2019"
5 output:
6   word_document:
7     fig_height: 4
8     fig_width: 6
9 ---
10 # This is the beginning of the project
11
12 Our initial reports might be restricted to lab meetings etc. We can use `R
13 Markdown` to show the code we are using, so that the meetings are not just a
14 demonstration of the results, but also an examination of the `code` used to obtain
15 them.
16
17 ## Data overview
18 {r packages and setup, include = FALSE}
19 knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE)
20 # Load your packages here
21 library(tidyverse)
22 library(knitr)
23
24 The plot below is call from the ggplot object entitled `report_plot` created in
25 the script `03_final_analysis.R`.
26 {r Plots from script, echo = FALSE}
27
28 source("scripts/03_final_analysis.R")
29
30 # The location of the Rmd file dictates whether the path to other files is intact
```

This is a reproducible document

Dr. Brendan Palmer

18th June 2019

This is the beginning of the project

Our initial reports might be restricted to lab meetings etc. We can use R Markdown to show the code we are using, so that the meetings are not just a demonstration of the results, but also an examination of the code used to obtain them.

Data overview

The plot below is call from the ggplot object entitled report_plot created in the script 03_final_analysis.R.

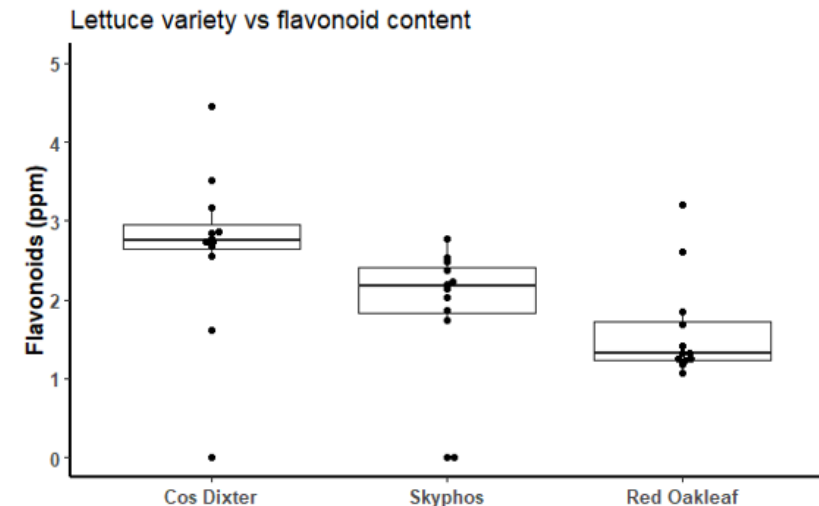
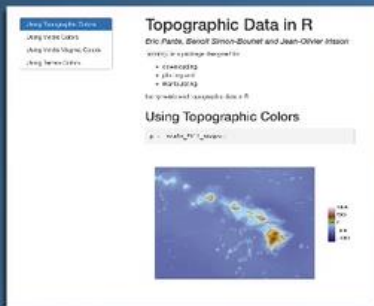


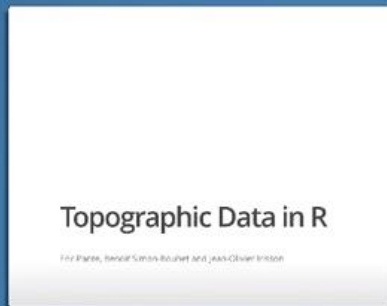
Fig. 1. Flavonoid content of three lettuce varieties under three experimental conditions.

Or we can also recreate the code within the R Markdown document as seen below.

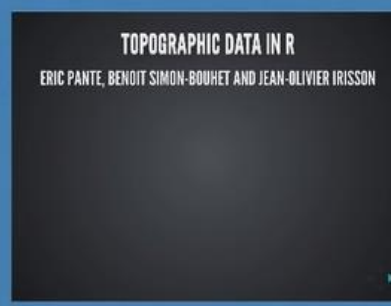
What has R Markdown ever done for us?



html



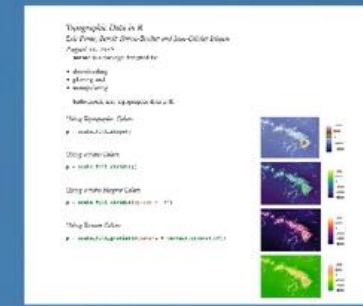
ioslides



reveal.js



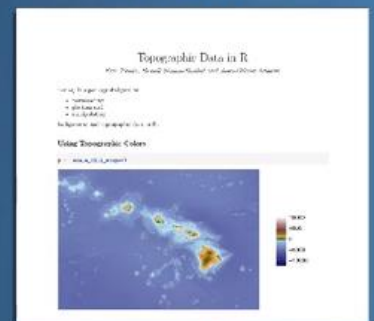
rtf



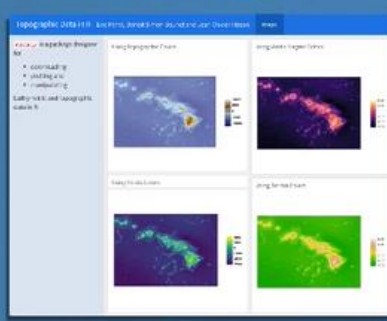
tuftes handout



book



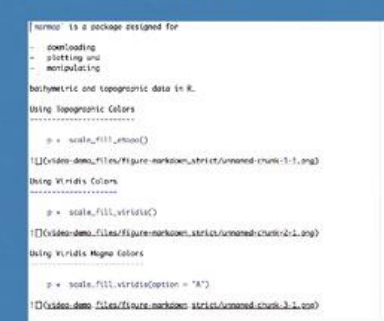
pdf



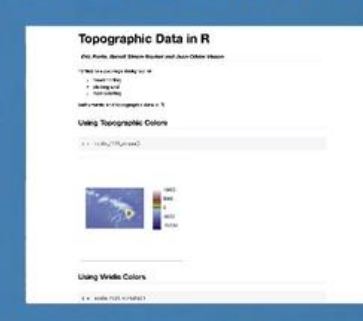
dashboard



slidy



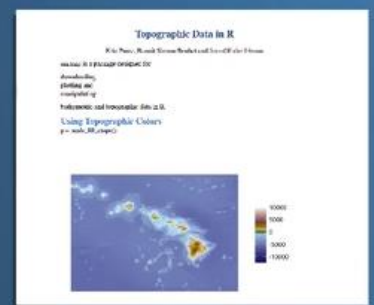
markdown



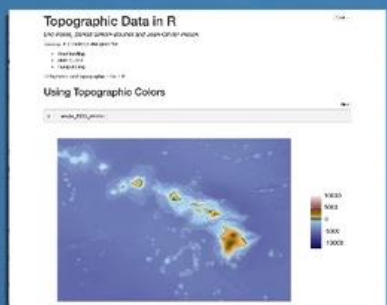
package vignette



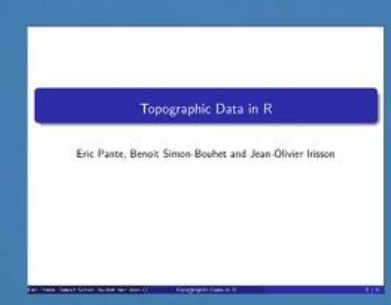
website



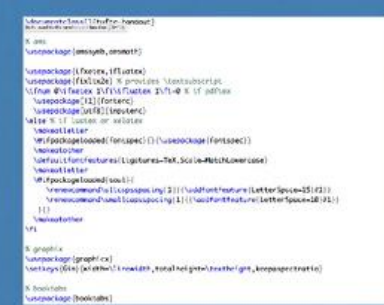
Word



notebook



beamer



latex



custom template



shiny app

R Markdown

YAML header

```
---  
title: "Diamond sizes"  
date: 2016-08-25  
output: html_document  
---
```

Chunks of code

```
```{r setup, include = FALSE}  
library(ggplot2)
library(dplyr)
smaller <- diamonds %>%
 filter(carat <= 2.5)
```
```

Plain text with data
outputs from R code

```
We have data about `r nrow(diamonds)`  
diamonds. Only  
`r nrow(diamonds) - nrow(smaller)` are  
larger than  
2.5 carats. The distribution of the  
remainder is shown below:
```

Chunks of code

```
```{r, echo = FALSE}  
smaller %>%
 ggplot(aes(carat)) +
 geom_freqpoly(binwidth = 0.01)
```
```

R Markdown

Knit the document

Insert new chunk

A YAML
header

Text formatted
with Markdown

Code
chunk

```
example_report.Rmd* x
[Navigation icons] Knit [Settings icon]
Insert [Up arrow] [Down arrow] Run [Refresh icon] [List icon]

1 ---
2 title: "This is a reproducible document"
3 author: "Dr. Brendan Palmer"
4 date: "2nd August 2019"
5 output:
6   word_document:
7     fig_height: 4
8     fig_width: 6
9 ---
10
11 # This is the beginning of the project
12
13 Our initial reports might be restricted to lab meetings etc. We can use `R Markdown`
14 show the code we are using, so that the meetings are not just a demonstration of t
15 results, but also an examination of the `code` used to obtain them.
16
17 ## Data overview
18
19 The plot below is call from the ggplot object entitled `report_plot` created in the
20 script `03_final_analysis.R`.
21
22 ```{r Plots from script, echo = FALSE}
23
24 library(tidyverse)
25 library(knitr)
26
27 source("scripts/03_final_analysis.R")
28
29 # The location of the Rmd file dictates whether the path to other files is intact
30
31 report_plot
32
33 ```
```

Click to run all
code chunks
above

Run code in the
chunk

R Markdown - Headers

```
# Header 1  
## Header 2  
### Header 3  
#### Header 4  
##### Header 5  
##### Header 6
```



Header 1
Header 2
Header 3
Header 4
Header 5
Header 6

R Markdown - Formatting

Text

italics

__bold__

``code``



Text

italics

bold

code

R Markdown - Lists

Bullets

- * bullet 1
- * bullet 2

Numbered list

1. item 1
2. item 2



Bullets

- bullet 1
- bullet 2

Numbered list

1. item 1
2. item 2

R Markdown - Hyperlinks

This is a
`[link](www.git.com)`.



This is a [link](#).

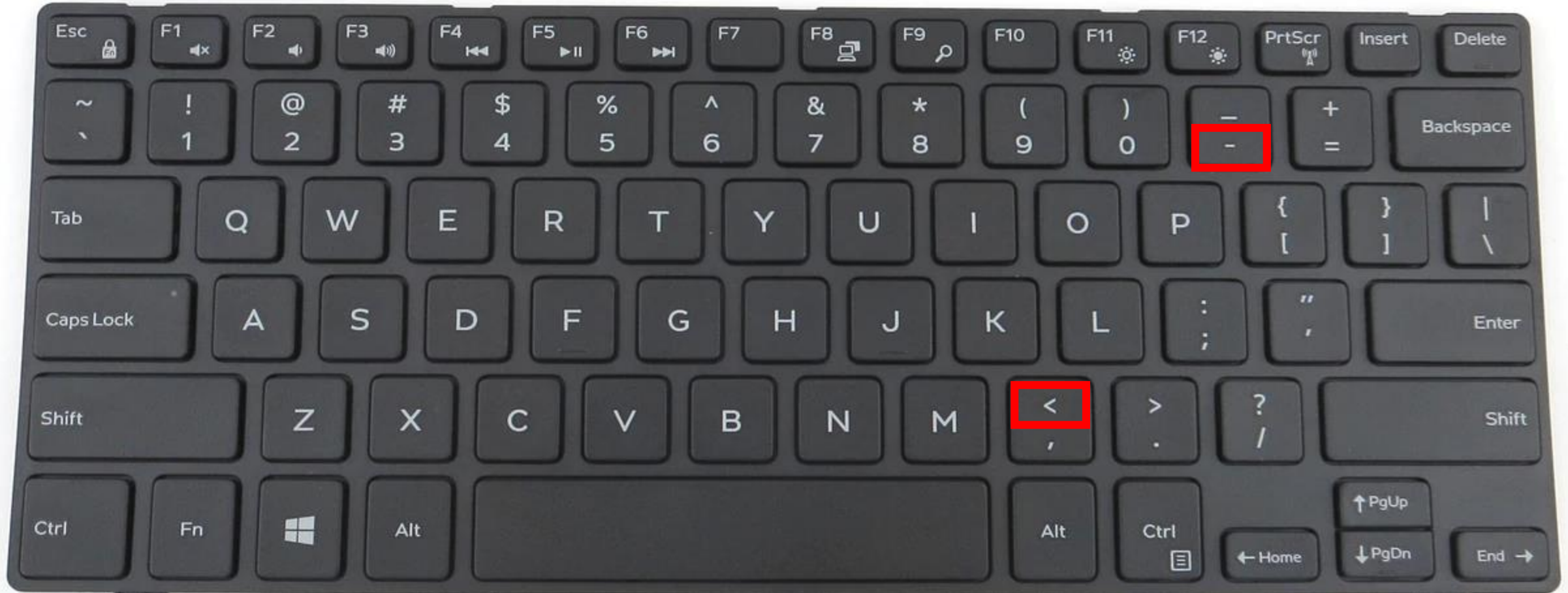
Basics of R code

| Symbol | What it does | Example 1 | Example 2 |
|---------------------|--|---|---|
| <code><-</code> | Assign operator
Creates new objects | <pre>> x <- 5 > x [1] 5</pre> | <pre>> y <- "This" > y [1] "This"</pre> |
| <code>c()</code> | Helps create
objects with more
than one element | <pre>> v <- c(5,6,7,8) > v [1] 5 6 7 8</pre> | <pre>> w <- c("This", "is", "easy! ") > w [1] "This" "is" "easy!"</pre> |
| <code>#</code> | Computer ignores
what is written.
Used for adding
notes to code | <pre>> # print("hello") ></pre> | <pre>> print("hello") [1] "hello"</pre> |
| <code>%>%</code> | Literally
translates as "then
do this" | <pre>> data %>% do_something_to(data)</pre> | <pre>> data %>% do_something_to(data) %>% do_something_else_to(data)</pre> |
| <code>%in%</code> | returns a logical
vector indicating
if there is a match | <pre>> "x" %in% c("x", "y", "z") [1] TRUE</pre> | <pre>> c("x", "y", "z") %in% "x" [1] TRUE FALSE FALSE</pre> |
| <code>?</code> | Access help | <pre>> ?mean()</pre> | <pre>> ?geom_point()</pre> |

FYI: R is case sensitive!! Name.of.data ≠ name.of.data

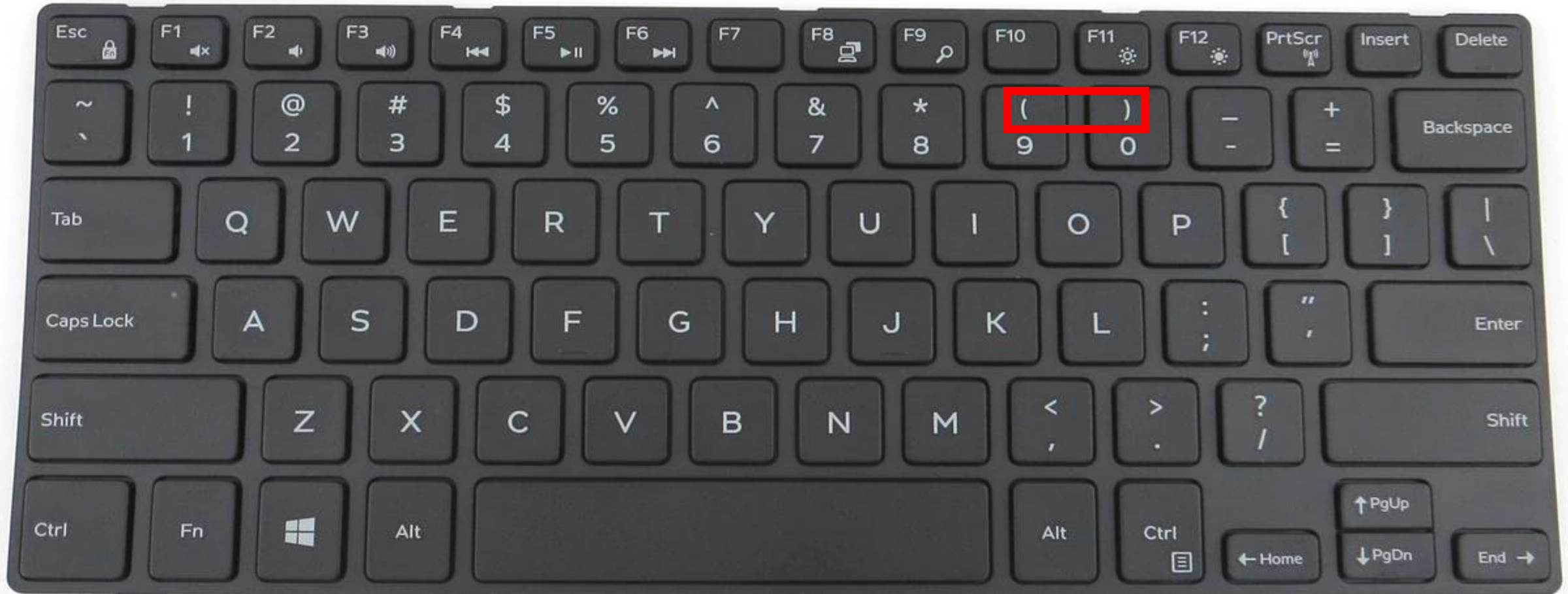
Say hello to the lesser known keyboard keys

Assignment operator: <-



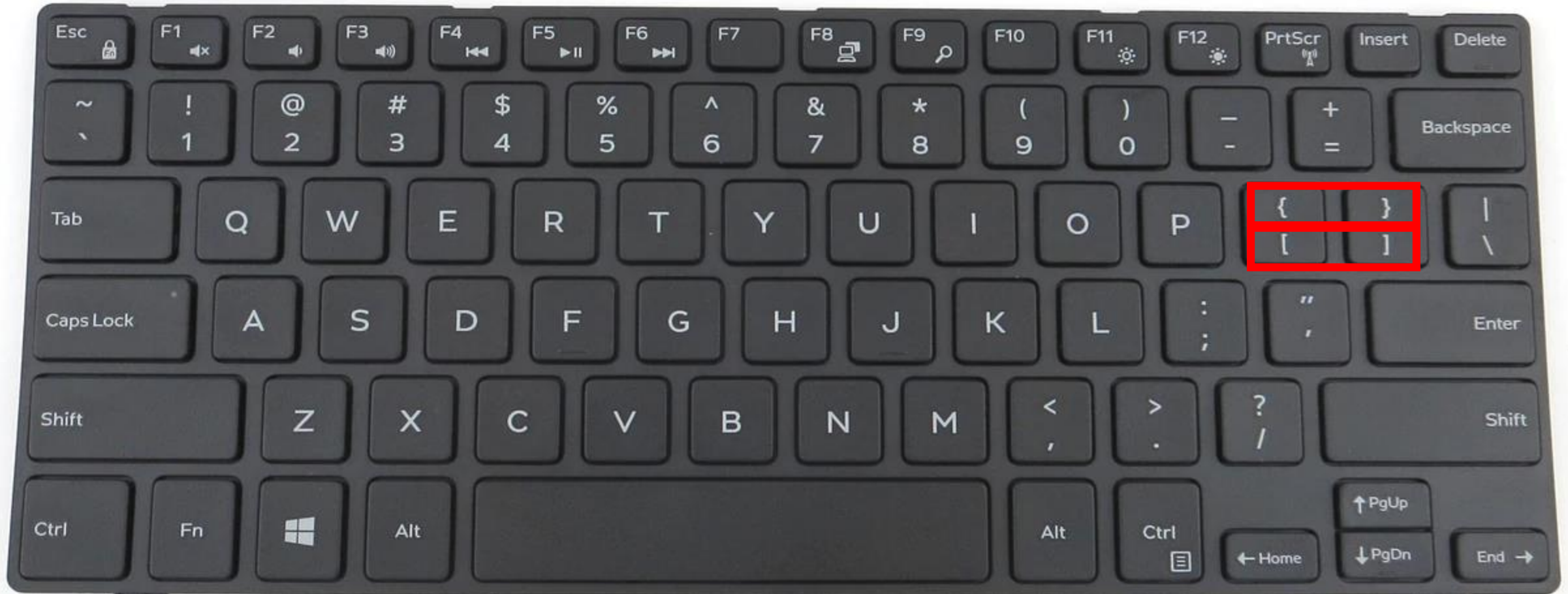
Say hello to the lesser known keyboard keys

Functions take arguments inside round brackets: `function()`



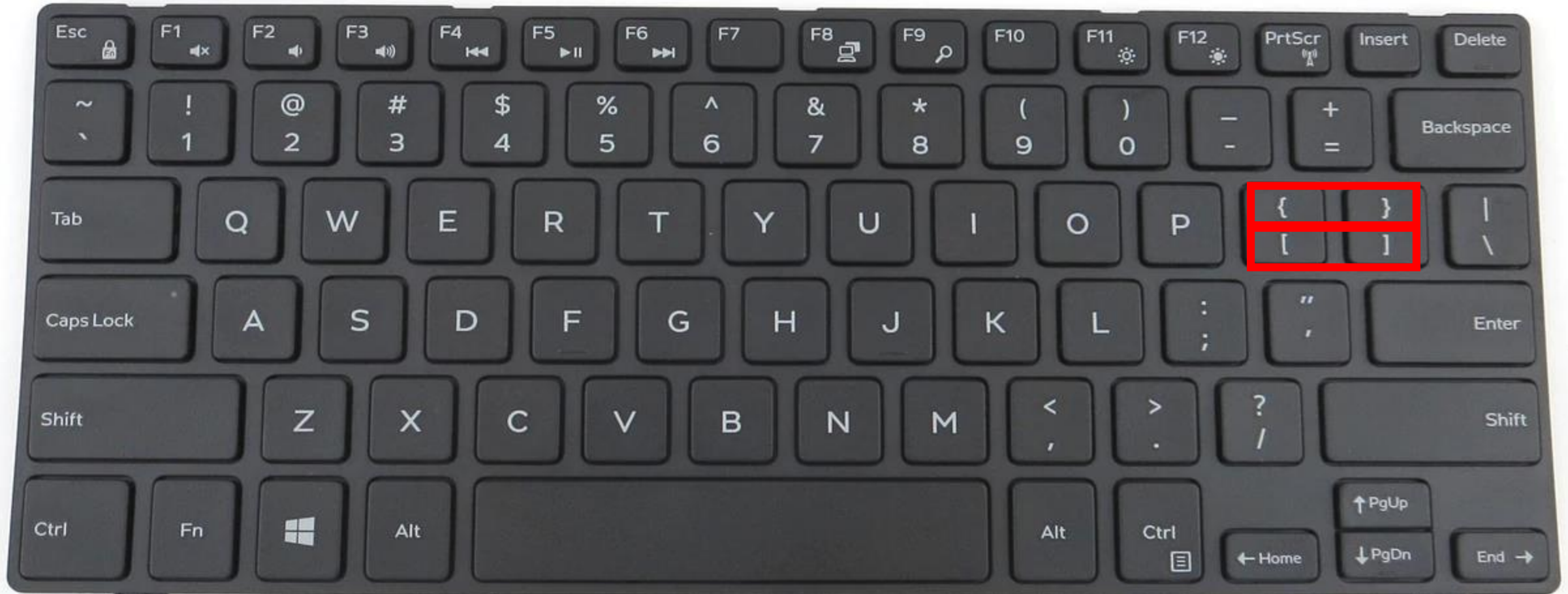
Say hello to the lesser known keyboard keys

Indexing occurs inside square brackets: **[]**



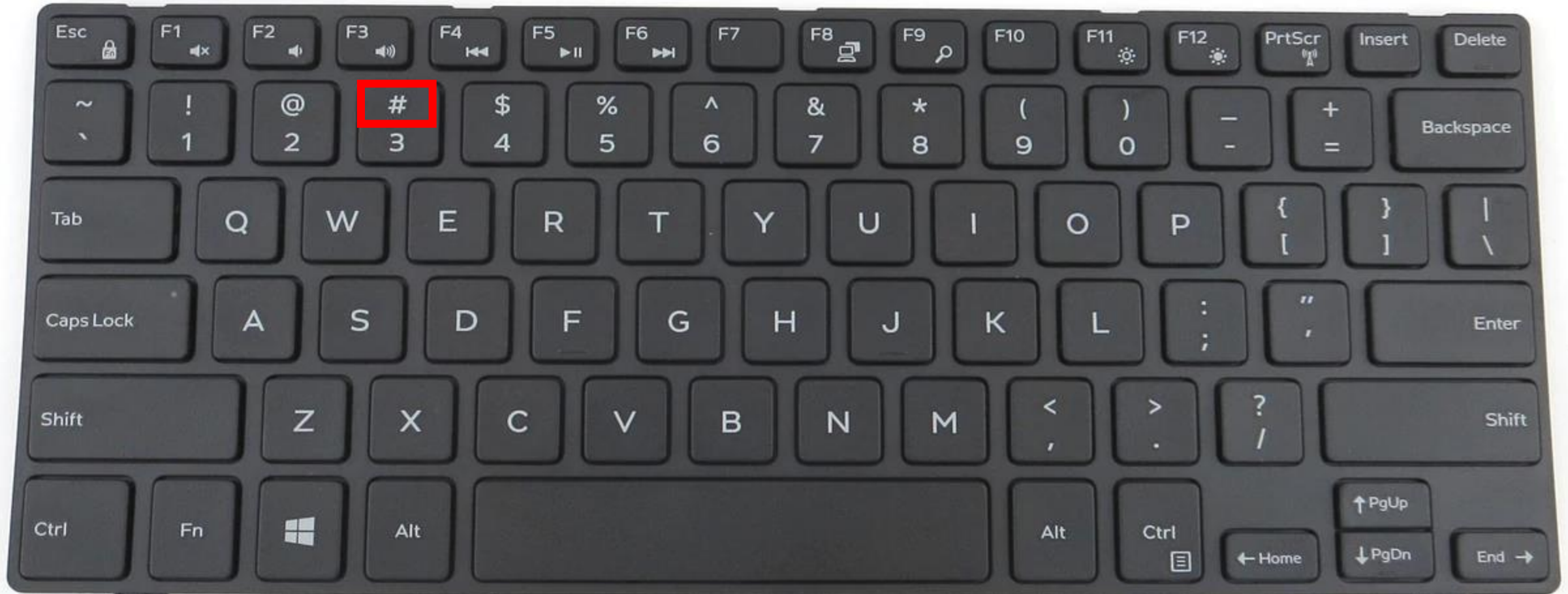
Say hello to the lesser known keyboard keys

Functions are defined inside curly brackets: **{ }**



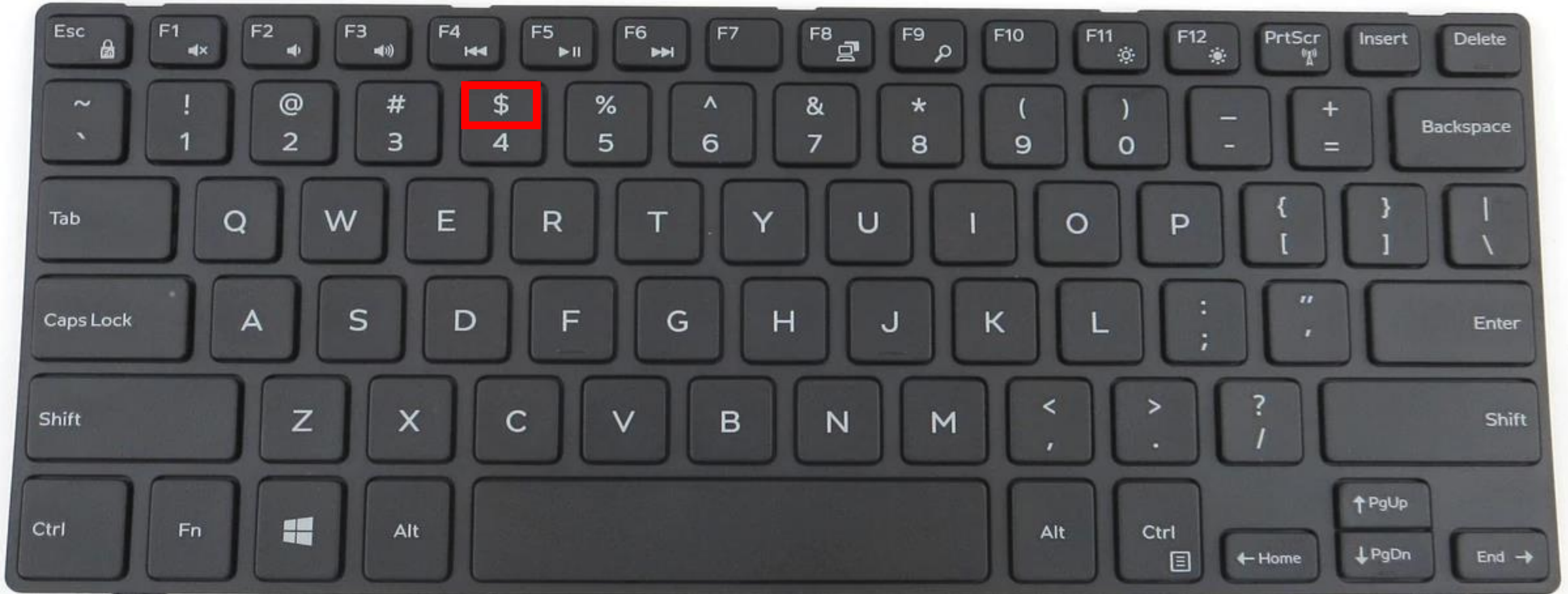
Say hello to the lesser known keyboard keys

You can comment out your code using the hash key: #



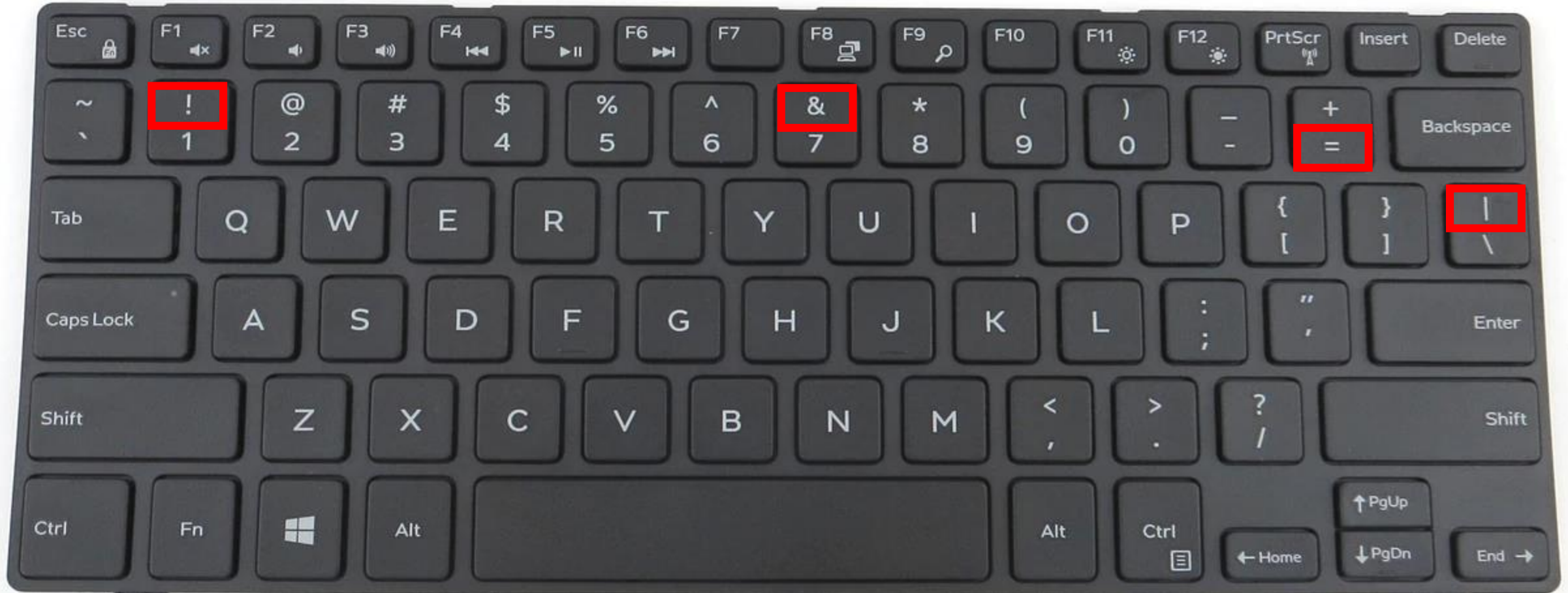
Say hello to the lesser known keyboard keys

Dollar sign allows you extract elements by name: \$



Say hello to the lesser known keyboard keys

Logical TRUE/FALSE operators equals, not equals, and, or:
==, !=, &, |



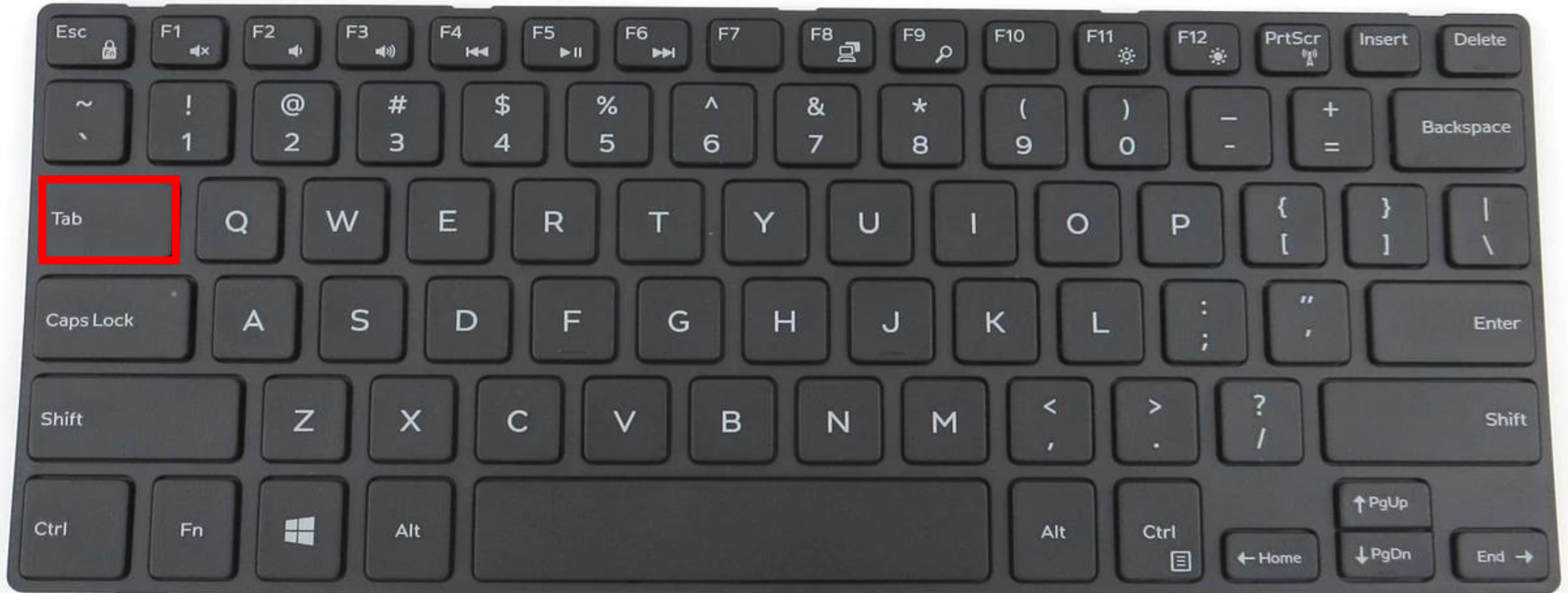
Say hello to the lesser known keyboard keys

Tilde operator for use in formulas: ~



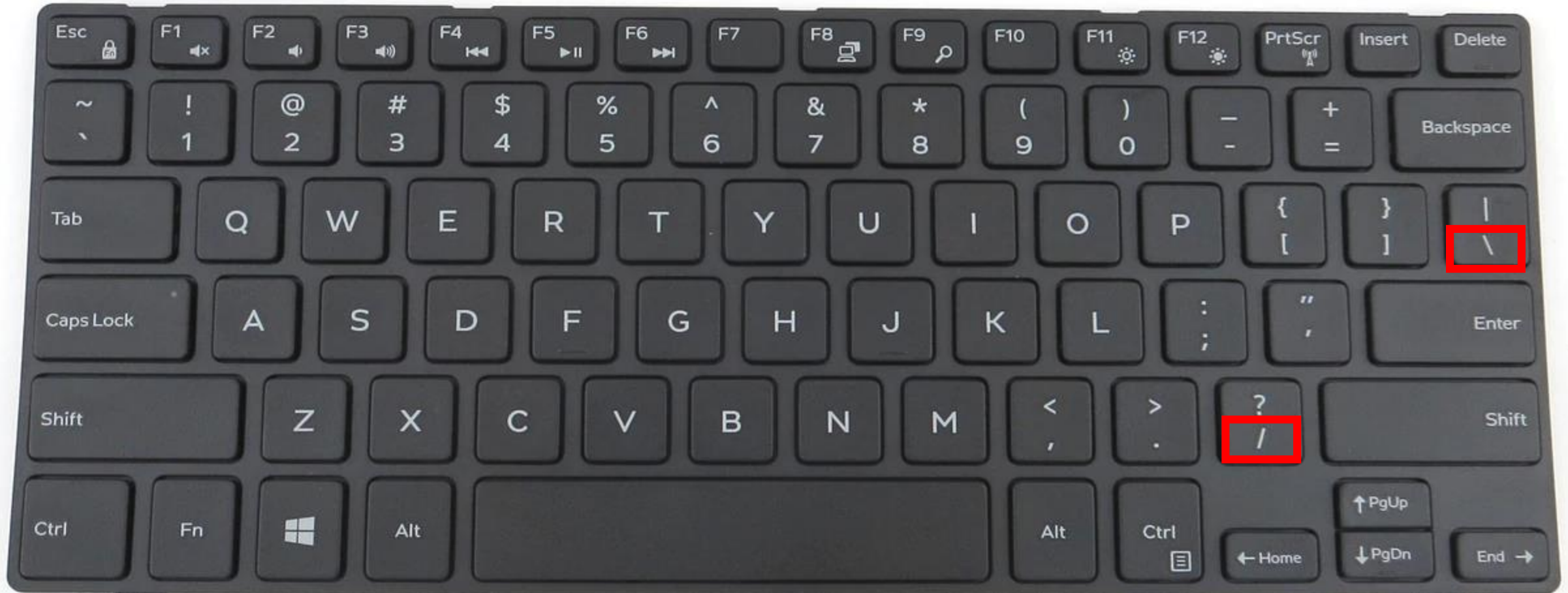
Say hello to the lesser known keyboard keys

Tab key for autocomplete

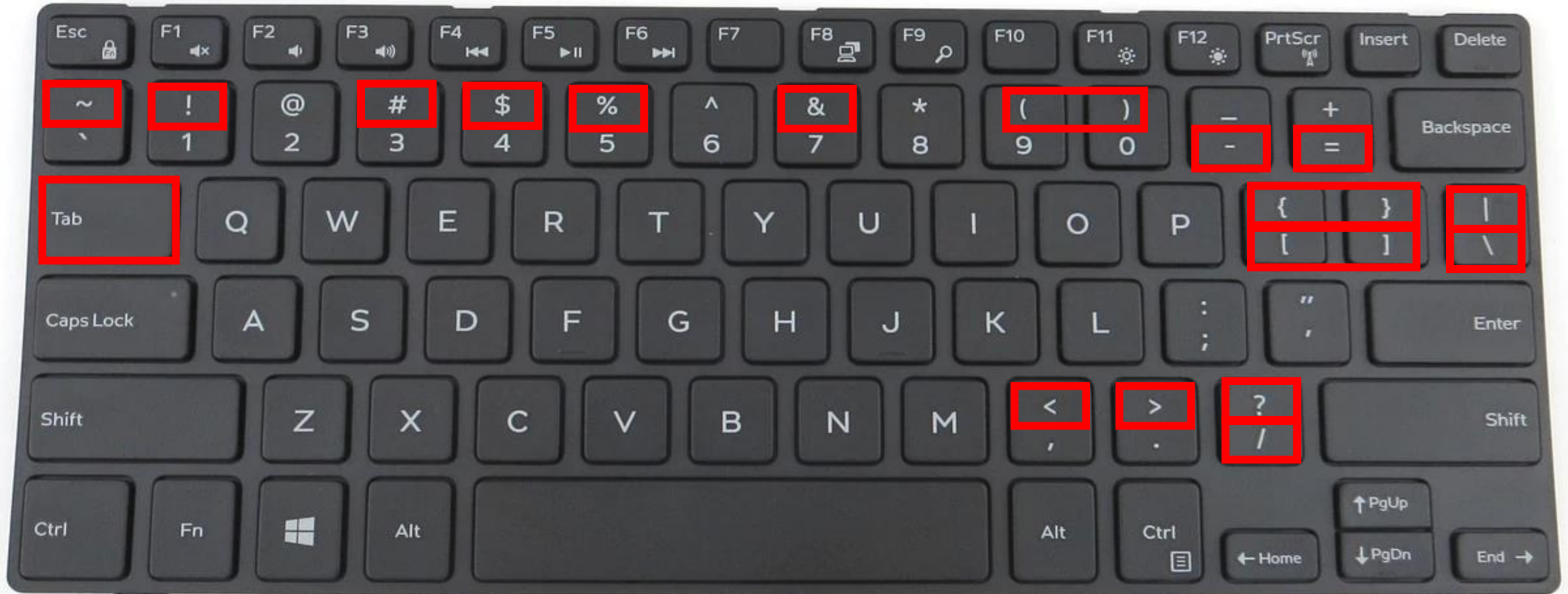


Say hello to the lesser known keyboard keys

Backslash and forward slash



Say hello to the lesser known keyboard keys



To understand R, remember the following

- Everything that exists is an object
- Everything that happens is a function

Creating objects

For most of us, R is simply the creation of and manipulation of objects:

```
new_object <- c(1, 2, 3)
```

- the objects are then fed into functions to create amazing new objects

```
amazing_new_object <- function(new_object)
```

Broadly speaking the following is true in R:

- information

```
> data_frame <- function(information)
```

```
> plot <- function(data_frame)
```

```
> model <- function(data_frame)
```

Naming objects

There are a few simple rules to follow initially:

- Object names must start with a letter and can only contain letters, numbers, `'_'` and `'.'`
- Certain characters should not be used, e.g:
 - `c` is the concatenate function `'c()'`
 - `T` is used as shorthand for `TRUE`
 - `F` is used as shorthand for `FALSE`

Three main types of data structure

The main data types are;

```
# double (for double precision floating point numbers)  
typeof(1.23)
```

```
# character  
typeof("string")
```

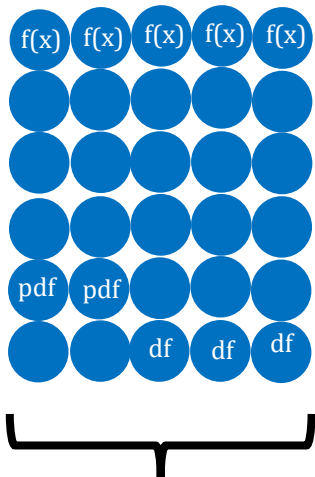
```
# logical  
typeof(FALSE)
```

```
# missing values are represented by NA  
example <- c(1, 2, NA, 4)
```

Walkthrough example script

- `scripts/01_baseR_introduction.R`

Package contents



Base R:
Comes
pre-
loaded

Functions

```
> base::|
```

| | |
|---------------|--------|
| max.col | {base} |
| mean | {base} |
| mean.Date | {base} |
| mean.default | {base} |
| mean.difftime | {base} |
| mean.POSIXct | {base} |
| mean.POSIXlt | {base} |
| mem.limits | {base} |
| memCompress | {base} |

mean(x, ...)

Generic function for the (trimmed) arithmetic mean.

Press F1 for additional help

Data sets

```
> data()
```

| |
|----------|
| faithful |
| freeny |
| infert |
| iris |
| iris3 |
| islands |
| lh |
| longley |

iris

This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are *Iris setosa*, *versicolor*, and *virginica*.

Press F1 for additional help

Conflicts

```
> filter|
```

| | |
|------------|---------|
| filter | {dplyr} |
| filter_ | {dplyr} |
| filter_all | {dplyr} |
| filter_at | {dplyr} |
| filter_if | {dplyr} |
| Filter | {base} |
| Filters | |

filter(x, filter, method = c("convolution", "recursive"), sides = 2L, circular = FALSE, init = NULL)

Applies linear filtering to a univariate time series or to each series separately of a multivariate time series.

Press F1 for additional help

Walkthrough example script

- `scripts/02_navigating_R_packages.R`

Practical notes

- docs/frequentist_inference.Rmd

Types of data structure

- Vectors come in two forms

A: Atomic vectors contain exactly one type of data

```
all_numbers      <- c(1, 2, 0.5, -0.5, 3.4)
all_characters   <- c("One", "too", "3")
all_logical      <- c(TRUE, FALSE) # NOTE: Type it out
```

B: Lists allow combinations of different types of data

```
this_is_a_list   <- list(1, TRUE, "Three", "4")
typeof(this_is_a_list)
[1] "list"
```

```
this_is_also_a_list <- list(all_numbers, all_characters)
```


OPTIONAL EXTRAS

Types of data structure

Matrices/Arrays:

- You can have a matrix of two or more dimensions

```
a_matrix <- matrix(1:9, 3, 3)
```

- Vectors and matrices can only contain **one** type of data
- **VERY VERY VERY NB:** If you try to create a vector with more than one data type, then it will undergo coercion to the least common denominator
- The coercion rule goes:
logical -> integer -> numeric -> complex -> character
- You can perform coercions yourself on vectors

Types of data structures

Data frames:

- These are a special type of list
- Observations are in rows
- Variables are in columns
- Labels or other metadata may also be present

```
> a_data_frame <- data.frame(number = 1:10,  
                             char = sample(letters, 10),  
                             this_really_a_col_name = rep(c(TRUE, FALSE), 5))
```

Indexing

- Indexing can occur in one or two dimensions

- One dimension:

```
new_object <- c(1, 2, 3)
```

```
new_object[1]
```

```
[1] 1
```

- Two dimensions

```
a_data_frame[1, 1] # i.e [Row number 1, Column number 1]
```

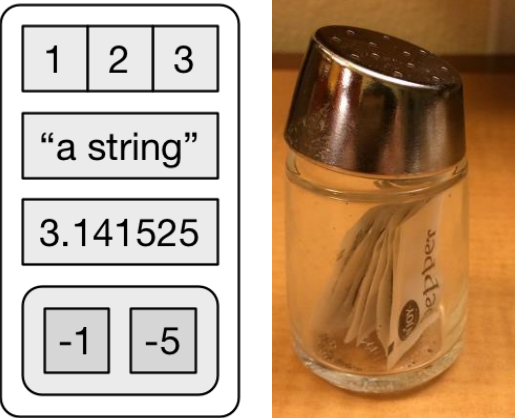
```
a_data_frame$number[1] # i.e. Column called number, row 1
```

Indexing

```
# Recall
```

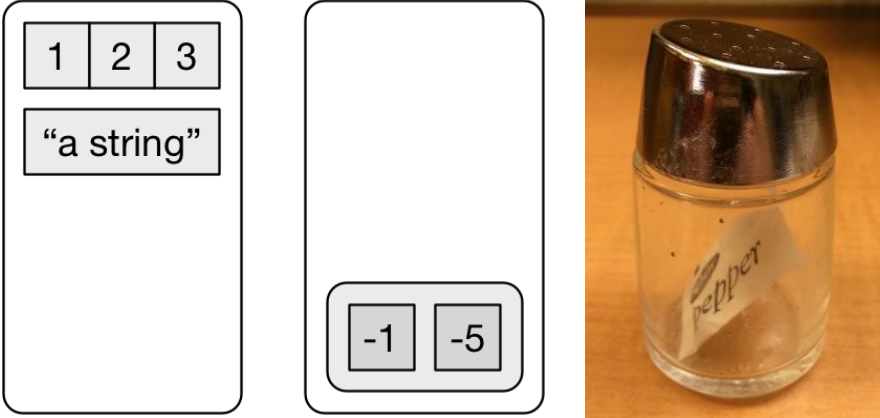
```
- this_is_also_a_list <- list(all_numbers, all_characters, all_logical)
```

a




list

a[1:2] a[4]



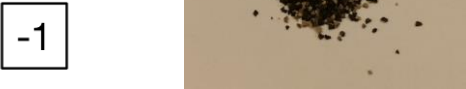
list

a[[4]] a[[4]][1]



Vector

a[[4]][[1]]



Vector

```
# Important
```

- `[` extracts a sublist, results will be a list
- `[[` extracts a single component

Try it out for yourself

- `scripts/03_practice_worksheet.R`