

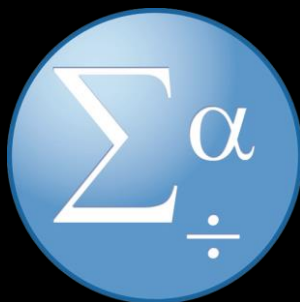
R for the Infrequent User

Shannon Dunnigan

GTMNERR

Thursday, September 23, 2021

Who am I?



PRIMER-e
empowering research



R Studio®



Why use R?

- It's FREE
- Integration
- Reproducibility
- Automation
- Community of support



Thomas Mock  @thomas_mock · Mar 21

Finalizing the slides on my [#rstats](#) presentation "Finding the YOU in the R community"

So I have a questions for the R community (at least on Twitter):

? "Why be an involved useR?"

15 7 43



Kim Cressman

@swmpkim

Following

Replying to @thomas_mock

Because it's fun and you can learn SO MUCH, without having to seek it out, just by watching what other people do. I have a lot of "OMG, you can do THAT?" moments.

I also learn a lot from what other people *ask*.

And sometimes I can help someone else, which is always nice.

8:42 PM - 21 Mar 2019

3 Likes



1 3 3

From Dr. Thomas Mock's presentation, "[A Gentle Introduction to Tidy Statistics in R](#)"



Thomas Mock  @thomas_mock · Mar 21

Finalizing the slides on my [#rstats](#) presentation "Finding the YOU in the R community"

So I have a questions for the R community (at least on Twitter):

? "Why be an involved useR?"

15 7 43



Luuuda

@ludmila_janda

Following

Replying to @thomas_mock

for all these feels:

👩 omg you can do that in R now?!

🙋 oh I think I can help with this issue

👏 oh man I've so been there

😊 thank you so much for helping me with this thing i've been going crazy trying to figure out

📊 whoa awesome data viz

📚 ohhh sweet resources

6:37 PM - 21 Mar 2019

3 Retweets 22 Likes



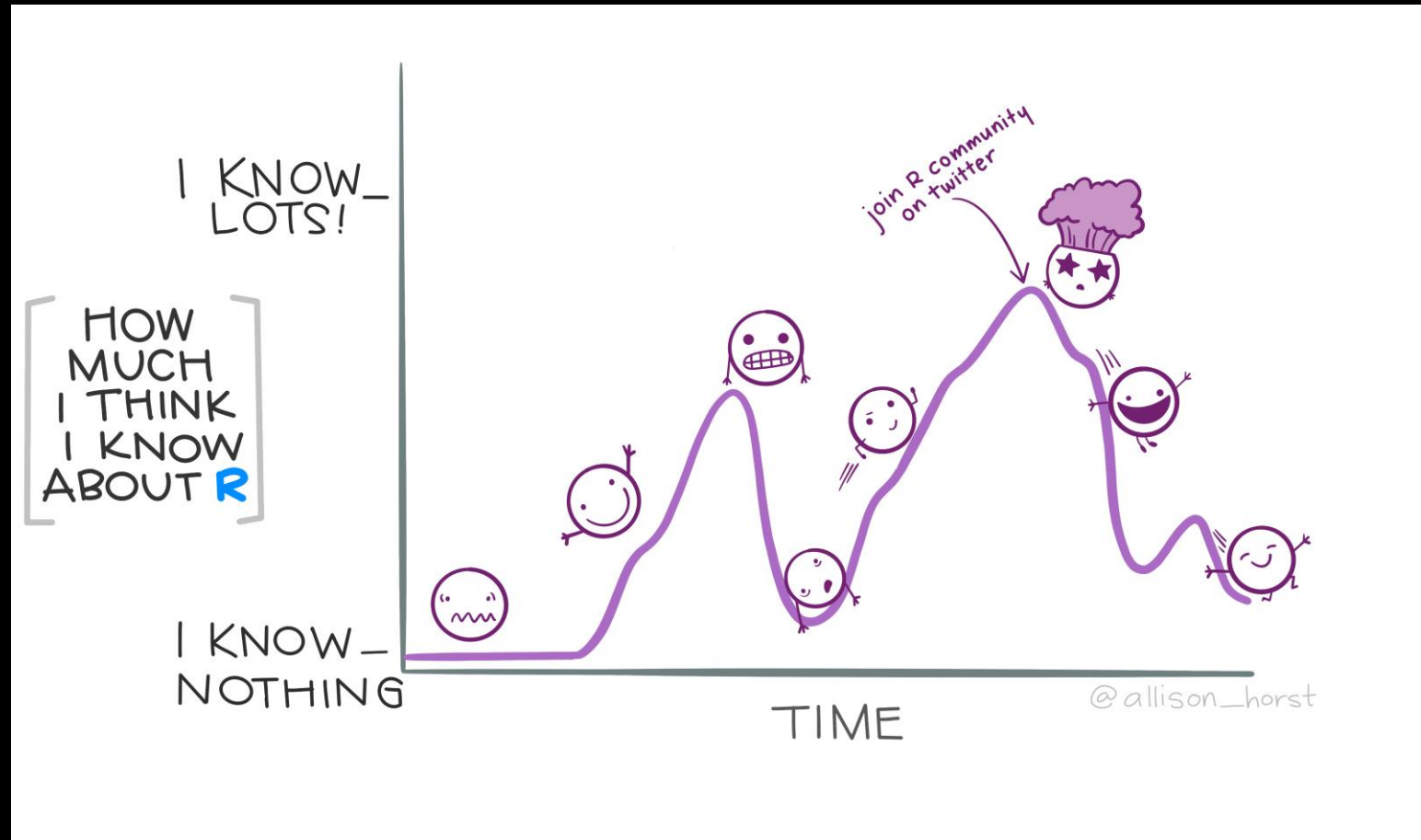
1 3 22

Who are you?

“The Infrequent UseR”



R Knowledge Rollercoaster by [Allison Horst](#)



This Workshop

What we will cover

- Reintroduction
- Best Practice for Reproducible Workflows
- Making a reproducible report with R Markdown and RStudio
- How to get and ask for help

What we will not cover

- Data tidying and wrangling
- Statistical analyses
- Customization of R Markdown outputs

but stay tuned...

HELLOOOO...IT'S ME

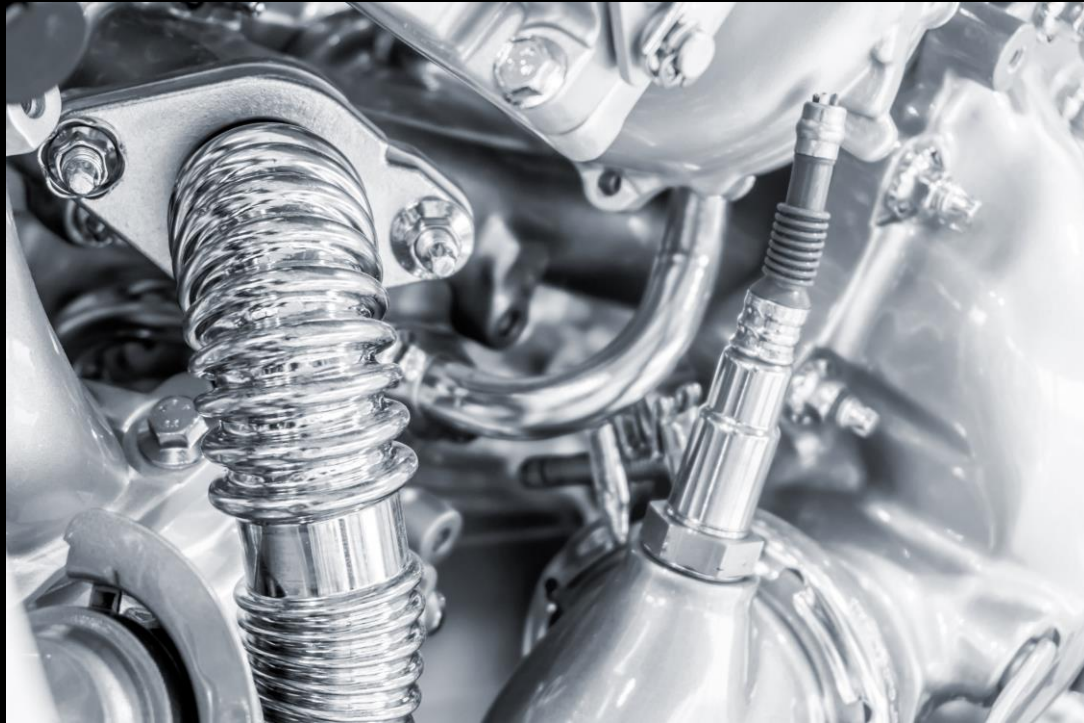


**“Hello, R, it’s me. I know it’s
been a while...”**

A Reintroduction to R

R vs. RStudio

R



RStudio



R vs R Packages

R



R Packages



Before you begin...

We recommend versions
R 4.0+ and RStudio 1.4+
for this workshop

- Check your versions of R and RStudio
- If you need to update:
 1. Manually install from CRAN (recommended if you do not care about old packages)
 2. Windows only: use ``installr`` package – run this through base R rather than RStudio
 3. MacOS only: use ``updateR`` - very similar to ``installr`` in Windows, but also requires the ``devtools`` package.



Packages

- Installing packages

```
# install the ggplot2 package  
install.packages("ggplot2")
```

```
# install the ggplot2, here, readxl, and janitor packages  
install.packages(c("ggplot2", "here", "readxl", "janitor"))
```

```
# install using a vector object  
my_packages <- c("ggplot2", "here", "readxl", "janitor")  
install.packages(my_packages)
```

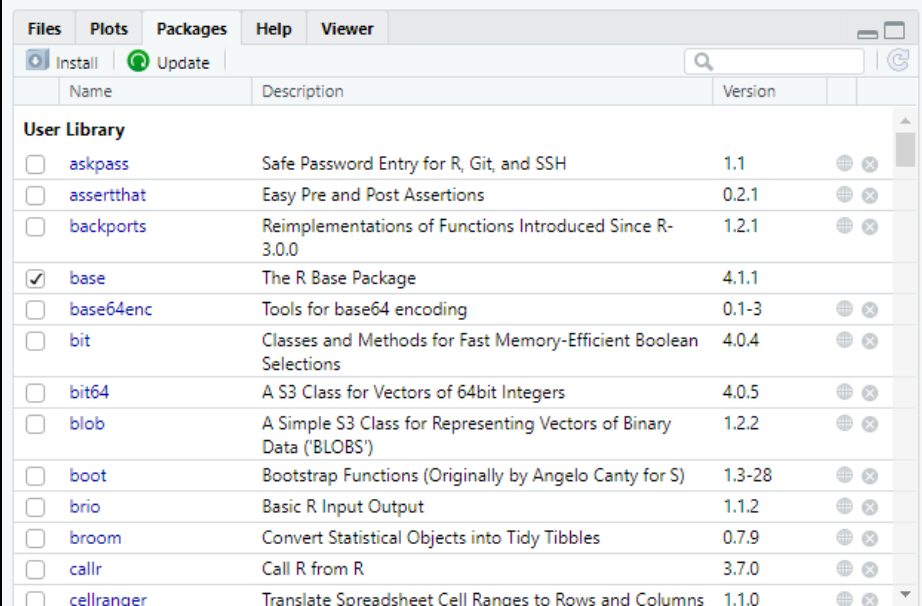
Or in RStudio, manage packages using Tools > Install Packages



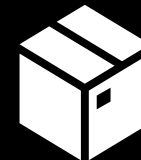
Packages

- Installing packages
- How to check what packages are installed already
check what packages are installed already
`installed.packages()`

- In RStudio, use the “Packages” tab
 - User Library (top): additional packages
 - System Library (bottom): base packages



Files	Plots	Packages	Help	Viewer
Install		Update		
Name	Description	Version		
User Library				
<input type="checkbox"/> askpass	Safe Password Entry for R, Git, and SSH	1.1		
<input type="checkbox"/> assertthat	Easy Pre and Post Assertions	0.2.1		
<input type="checkbox"/> backports	Reimplementations of Functions Introduced Since R-3.0.0	1.2.1		
<input checked="" type="checkbox"/> base	The R Base Package	4.1.1		
<input type="checkbox"/> base64enc	Tools for base64 encoding	0.1-3		
<input type="checkbox"/> bit	Classes and Methods for Fast Memory-Efficient Boolean Selections	4.0.4		
<input type="checkbox"/> bit64	A S3 Class for Vectors of 64bit Integers	4.0.5		
<input type="checkbox"/> blob	A Simple S3 Class for Representing Vectors of Binary Data ('BLOBS')	1.2.2		
<input type="checkbox"/> boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-28		
<input type="checkbox"/> brio	Basic R Input Output	1.1.2		
<input type="checkbox"/> broom	Convert Statistical Objects into Tidy Tibbles	0.7.9		
<input type="checkbox"/> callr	Call R from R	3.7.0		
<input type="checkbox"/> cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0		

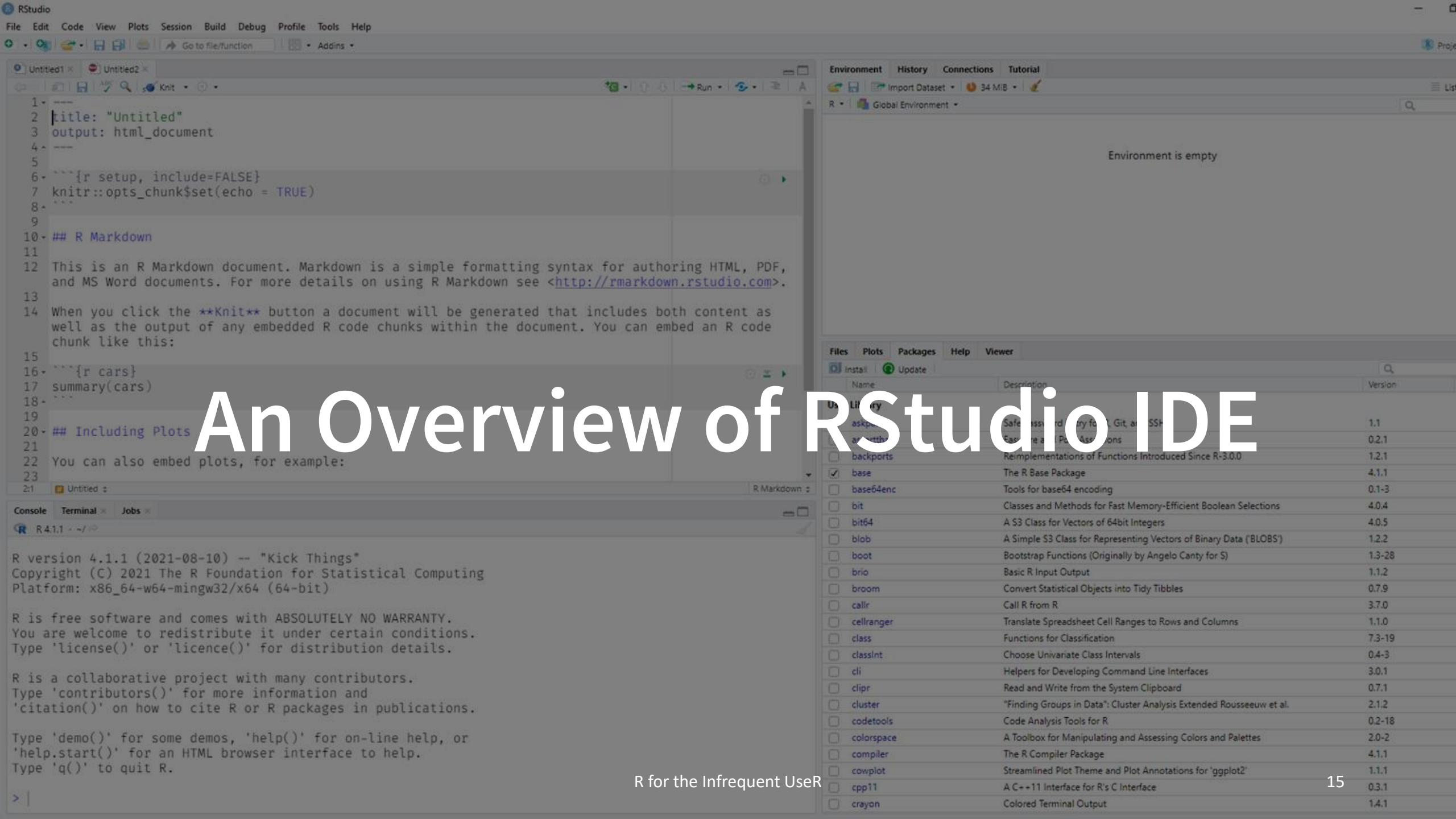


Packages

- Installing packages
- How to check what packages are installed already
- Updating packages
 - # list all packages where update is available
`old.packages()`

```
# update all available packages  
update.packages()
```

```
# update all without prompts for permission/clarification  
update.packages(ask = FALSE)
```



An Overview of RStudio IDE

RStudio IDE :: CHEAT SHEET

Documents and Apps

Open Shiny, R Markdown, knitr, Sweave, LaTeX, Rd files and more in Source Pane

Check spelling, Render output, Choose output format, Insert code chunk, Publish to server

Jump to previous chunk, Run selected code chunk, Show file outline, Visual Editor (reverse side), Run this and all previous code chunks, Run this code chunk, Set knitr chunk options

Access markdown guide at **Help > Markdown Quick Reference**. See reverse side for more on **Visual Editor**

RStudio recognizes that files named **app.R**, **server.R**, **ui.R**, and **global.R** belong to a shiny app

Run app, Choose location to shinyapps.io, Publish to shinyapps.io, Manage accounts

Source Editor

Navigate backwards/forwards, Open in new window, Save, Find and replace, Compile and run notebook, Run selected code

Re-run previous code, Source with or without Echo or as a local job, Show file outline, Multiple runways/column selection with **Alt + mouse drag**, Code diagnostics that appear in the margin, Hover over diagnostic symbols for details, Syntax highlighting based on your file's extension, Tab completion to finish function names, file paths, arguments, and more, Multi-language code snippets to quickly use common blocks of code, Jump to function in file, Change file type

Run scripts in separate sessions, Madmode, minimize panes, Ctrl/Cmd + R Markdown Build Log, Drag pane boundaries

Tab Panes

Import data with wizard, History of past commands to run/copy, Manage external databases, View memory usage, R tutorials

Load workspace, Save workspace, Clear R workspace, Search inside environment, Choose environment to display from list of parent environments, Display objects as list or grid, Functions, Function (op), View in data viewer, View function source code

Create, Delete, Rename, Path to displayed directory, A file browser keyed to your working directory. Click on file or directory name to open.

Version Control

Turn on at **Tools > Project Options > Git/SVN**

Added, Deleted, Modified, Untracked

Stage files, Commit staged files, Push/Pull to remote, View History, Current branch

Show file diff to view file differences

Debug Mode

Use **debug()**, **browser()**, or a breakpoint and execute your code to open the debugger mode.

Launch debugger mode from origin of error, Open traceback to examine the functions that R called before the error occurred

Package Development

Create a new package with **File > New Project > New Directory > R Package**

Enable roxygen documentation with **Tools > Project Options > Build Tools**

Roxygen guide at **Help > Roxygen Quick Reference**

See package information in the **Build Tab**

Install package, Run devtools::load_all() and reload changes, Clear output and rebuild, Run R CMD check, Customizable package build options

RStudio opens plots in a dedicated **Plots** pane

Navigate recent plots, Open in window, Export plot, Delete plot, Delete all plots

GUI **Package** manager lists every installed package

Install Packages, Update Packages, Browse package site, Click to load package with **library()**. (Un)click to detach package with **detach()**, Package version installed, Delete from library

RStudio opens documentation in a dedicated **Help** pane

Home page of helpful links, Search within help file, Search for help file

Viewer pane displays HTML content, such as Shiny apps, RMarkdown reports, and interactive visualizations

Stop Shiny app, Publish to shinyapps.io, rpubs, RStudioConnect, ...

View<data> opens spreadsheet like view of data set

Filter rows by value or value range, Sort by values, Search for value

Click next to line number to add/remove a breakpoint. Highlighted line shows when execution has paused

Run commands in environment where execution has paused, Examine variables in executing environment, Select function in traceback debug

Console, **Terminal**, **Jobs**

Step through code one line at a time, Step into and out of functions to run, Resume execution, Quit debug mode

<https://www.rstudio.com/resources/cheatsheets/>

Keyboard Shortcuts

	Windows/Linux	Mac
RUN CODE		
Search command history	Ctrl+↑	Cmd+↑
Interrupt current command	Esc	Esc
Clear console	Ctrl+L	Ctrl+L
NAVIGATE CODE		
Go to File/Function	Ctrl+.	Ctrl+.
WRITE CODE		
Attempt completion	Tab or Ctrl+Space	Tab or Ctrl+Space
Insert <- (assignment operator)	Alt+-	Option+-
Insert %>% (pipe operator)	Ctrl+Shift+M	Cmd+Shift+M
(Un)Comment selection	Ctrl+Shift+C	Cmd+Shift+C
MAKE PACKAGES		
Load All (devtools)	Ctrl+Shift+L	Cmd+Shift+L
Test Package (Desktop)	Ctrl+Shift+T	Cmd+Shift+T
Document Package	Ctrl+Shift+D	Cmd+Shift+D

DOCUMENTS AND APPS

Knit Document (knitr) Ctrl+Shift+K Cmd+Shift+K

Insert chunk (Sweave & Knitr) Cmd+Alt+I Cmd+Option+I

Run from start to current line Ctrl+Alt+B Cmd+Option+B

MORE KEYBOARD SHORTCUTS

Keyboard Shortcuts Help Alt+Shift+K Option+Shift+K

Show Command Palette Ctrl+Shift+P Cmd+Shift+P

View the Keyboard Shortcut Quick Reference with **Tools > Keyboard Shortcuts** or **Alt+Option + Shift + K**

Search for keyboard shortcuts with **Tools > Show Command Palette** or **Ctrl/Cmd + Shift + P**

Visual Editor

Block format, Check spelling, Render output, Choose output format, Choose output location, Insert code chunk, Jump to previous chunk, Jump to next chunk, Run selected lines, Publish to server, Show file outline, Back to Source Editor (front page)

Heading 2, Lists and block quotes, Links, Citations, Images, Insert blocks, citations, equations, and special characters, More formatting, R Markdown including Plots, File outline, Insert and edit tables

Clear formatting, Insert verbatim code, Add/Edit attributes

Jump to chunk or header

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.

```
{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

#> corrs
summary(corr)
```

RStudio Workbench

WHY RSTUDIO WORKBENCH?

Extend the open source server with a commercial license, support, and more:

- open and run multiple R sessions at once
- tune your resources to improve performance
- administrative tools for managing user sessions
- collaborate real-time with others in shared projects
- switch easily from one version of R to a different version
- integrate with your authentication, authorization, and audit practices
- work in the RStudio IDE, JupyterLab, Jupyter Notebooks, or VS Code

Download a free 45 day evaluation at www.rstudio.com/products/workbench/evaluation/

Share Projects

File > New Project

RStudio saves the call history, workspace, and working directory associated with a project. It reloads each when you re-open a project.

Start new R Session in current project, Close R Session in project, Active shared collaborators, Name of current project, Select R Version, Share Project with Collaborators

Run Remote Jobs

Run R on remote clusters (Kubernetes/Slurm) via the Job Launcher

Monitor launcher jobs, Launch a job, Add/Edit attributes, Set knitr chunk options, Run this and all previous code chunks, Run this code chunk

Job Name	Status	Location	Time
sleep.R	Running	Local	0:08
sleep.R	Succeeded 11:22 AM	Local	0:41
sleep.R	Idle	Kubernetes	Waiting

Run launcher jobs remotely



RStudio® is a trademark of RStudio, PBC • CC BY-SA RStudio • info@rstudio.com • 044-448-1212 • rstudio.com • Learn more at rstudio.com • Fort Awesome 5.15.3 • RStudio IDE 1.4.1717 • Updated: 2021-0



RStudio® is a trademark of RStudio, PBC • CC BY-SA RStudio • info@rstudio.com • 044-448-1212 • rstudio.com • Learn more at rstudio.com • Fort Awesome 5.15.3 • RStudio IDE 1.4.1717 • Updated: 2021-0

A background image showing a collaborative work environment. Two people are seated at a desk, looking at a laptop and various documents. The documents feature colorful charts, including a pie chart and bar graphs. A red pen and a glass of water are also visible on the desk. The overall scene is dimly lit, with the primary light source coming from the laptop screens.

A Reproducible Workflow

Adaptations from Jenny Bryan and Jim Hester's work
"What They Forgot to Teach You About R"

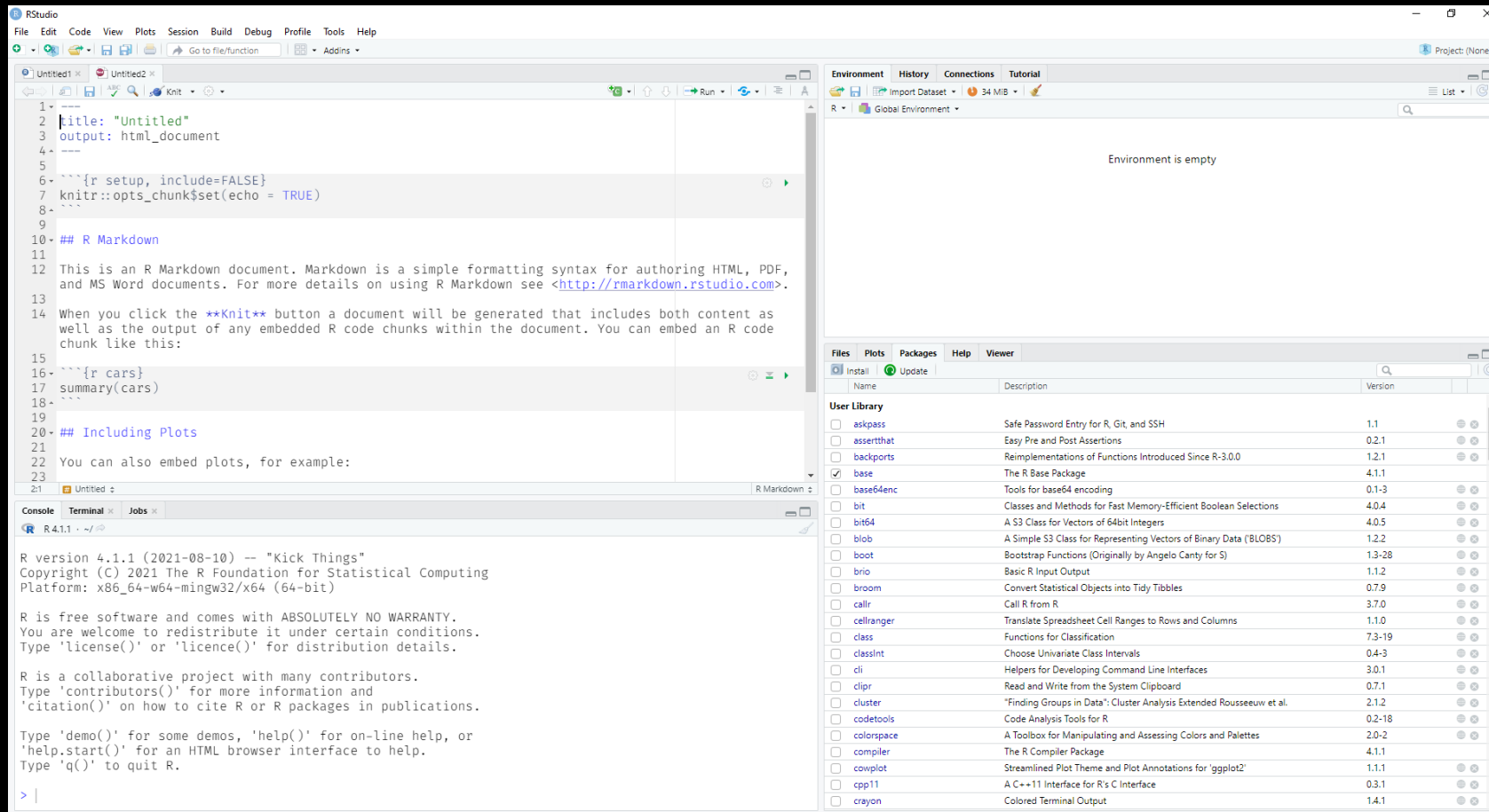
<https://rstats.wtf/>

Reproducibility: Use an IDE

Environment

Build and Git
integration

Source Code



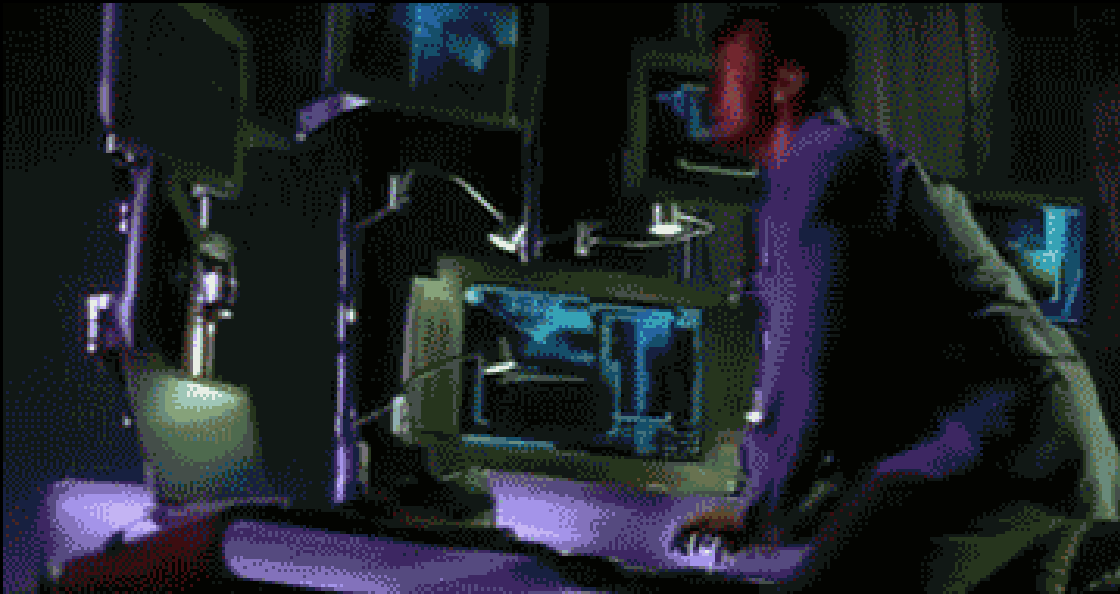
Files

Plots

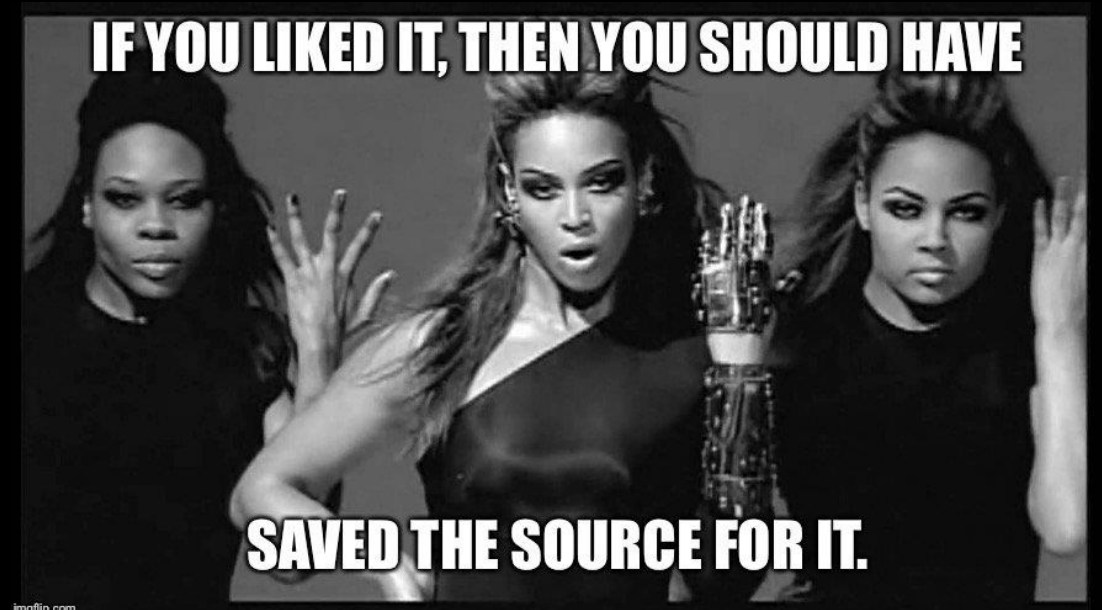
Viewer

Console

Reproducibility: Save the Source, not the workspace



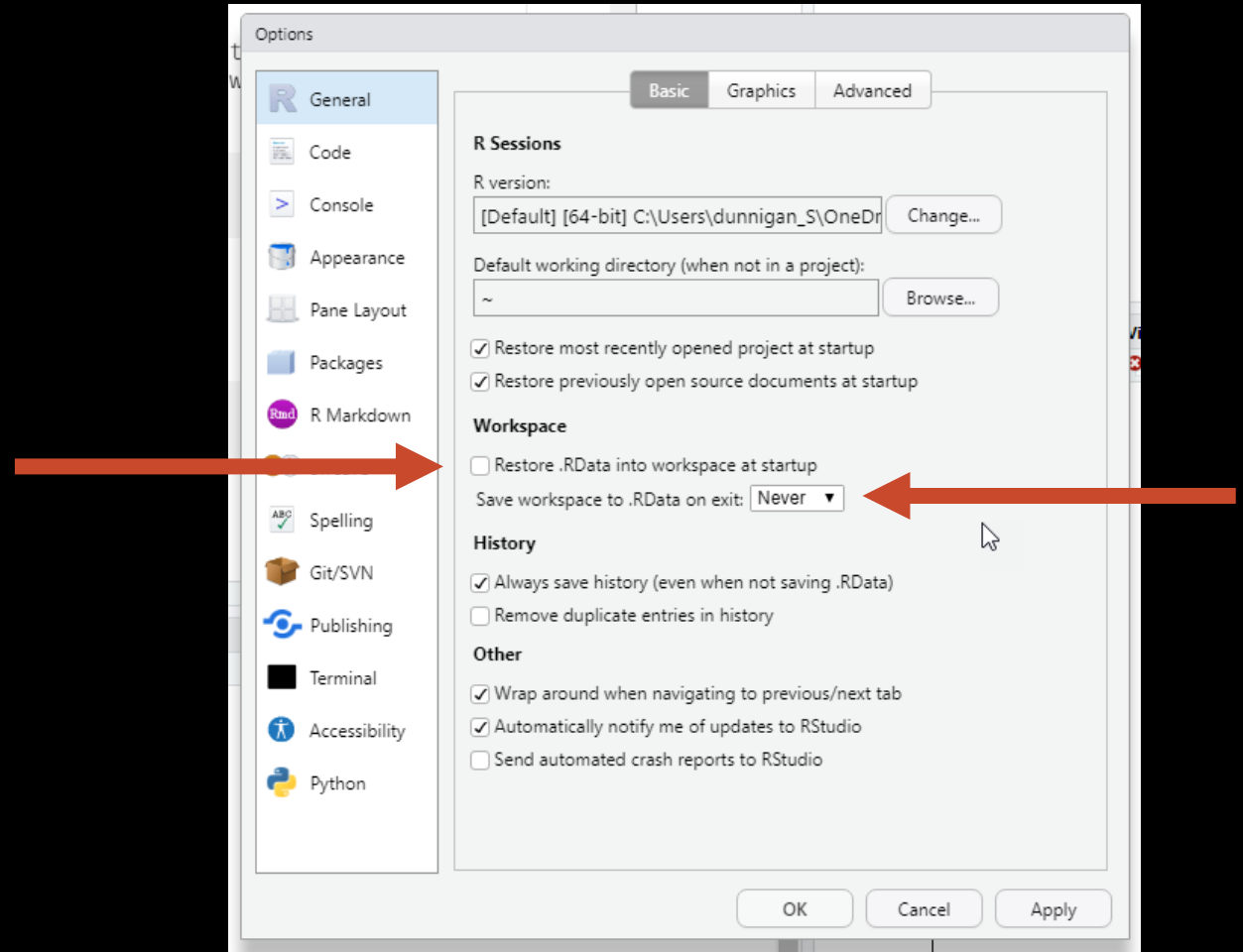
Hugh Jackman in “Swordfish” (2001) – [great scene](#)



[From WTF](#) (too good not to reshare)

Reproducibility: Save the Source, not the workspace

- In RStudio go to Tools > Global Options
- Or ``usethis::use_blank_slate()``




Reproducibility: Saving Objects

- Some analyses take a long time to execute
- Break analysis into natural phases
- Isolate computationally demanding steps: Script -> outputs
- Write objects to file

```
saveRDS(object, here("output", "my_object.rds"))  
# load these objects in subsequent scripts  
my_object <- readRDS(here("output", "my_object.rds"))
```

```
# Or use save() and load() with .RData  
save(object, here("output", "my_object.RData"))  
load(here("output", "my_object.RData"))
```

Reproducibility: Project-oriented Workflow



Name	Status	Date modified	Type	Size
.git	☁	9/9/2021 11:00 AM	File folder	
.Rproj.user	☁	9/9/2021 11:00 AM	File folder	
backgroundinfo	☁	9/9/2021 11:00 AM	File folder	
data	☁	9/9/2021 11:00 AM	File folder	
images	☁	9/9/2021 11:00 AM	File folder	
output	☁	9/9/2021 11:00 AM	File folder	
R	☁	9/9/2021 11:00 AM	File folder	
reportdocs	☁	9/9/2021 11:00 AM	File folder	
.gitignore	☁	2/10/2021 12:03 PM	Text Document	1 KB
.Rhistory	☁	7/8/2021 4:28 PM	RHISTORY File	18 KB
_config.yml	☁	6/18/2019 8:06 AM	YML File	1 KB
guana	☁	7/8/2021 3:55 PM	R Project	1 KB
my-styles	☁	8/1/2019 11:59 AM	Microsoft Word D...	18 KB
README	☁	2/10/2021 12:02 PM	Chrome HTML Do...	618 KB
README.md	☁	2/17/2021 6:06 PM	MD File	1 KB

Reproducibility: Project-oriented Workflow

Name	Status	Date modified	Type	Size
.git	☁	9/9/2021 11:00 AM	File folder	
.Rproj.user	☁	9/9/2021 11:00 AM	File folder	
backgroundinfo	☁	9/9/2021 11:00 AM	File folder	
data	☁	9/9/2021 11:00 AM	File folder	
images	☁	9/9/2021 11:00 AM	File folder	
output	☁	9/9/2021 11:00 AM	File folder	
R	☁	9/9/2021 11:00 AM	File folder	
reportdocs	☁	9/9/2021 11:00 AM	File folder	
.gitignore	☁	2/10/2021 12:03 PM	Text Document	1 KB
.Rhistory	☁	7/8/2021 4:28 PM	RHISTORY File	18 KB
_config.yml	☁	6/18/2019 8:06 AM	YML File	1 KB
guana	☁	7/8/2021 3:55 PM	R Project	1 KB
my-styles	☁	8/1/2019 11:59 AM	Microsoft Word D...	18 KB
README	☁	2/10/2021 12:02 PM	Chrome HTML Do...	618 KB
README.md	☁	2/17/2021 6:06 PM	MD File	1 KB

Reproducibility: Use Safe Paths

*“If the first line of your #rstats script is
`setwd(“C:\Users\jenny\path\that\only\I\have”)`,
I will come into your lab and SET YOUR COMPUTER ON FIRE.”*

[Rage tweets by @jennybc and @tpoi](#)

Reproducibility: Project-oriented Workflow

Name	Status	Date modified	Type	Size
.git	☁	9/9/2021 11:00 AM	File folder	
.Rproj.user	☁	9/9/2021 11:00 AM	File folder	
backgroundinfo	☁	9/9/2021 11:00 AM	File folder	
data	☁	9/9/2021 11:00 AM	File folder	
images	☁	9/9/2021 11:00 AM	File folder	
output	☁	9/9/2021 11:00 AM	File folder	
R	☁	9/9/2021 11:00 AM	File folder	
reportdocs	☁	9/9/2021 11:00 AM	File folder	
.gitignore	☁	2/10/2021 12:03 PM	Text Document	1 KB
.Rhistory	☁	7/8/2021 4:28 PM	RHISTORY File	18 KB
_config.yml	☁	6/18/2019 8:06 AM	YML File	1 KB
guana	☁	7/8/2021 3:55 PM	R Project	1 KB
my-styles	☁	8/1/2019 11:59 AM	Microsoft Word D...	18 KB
README	☁	2/10/2021 12:02 PM	Chrome HTML Do...	618 KB
README.md	☁	2/17/2021 6:06 PM	MD File	1 KB

here: find your
PATH!



Artwork by
Allison Horst
(<https://github.com/allisonhorst>)

So, what does that look like?

```
# load here package
```

```
library(here)
```

```
here() starts at C:/Users/Dunnigan_S/2021-infrequent-useR
```

```
# run here
```

```
here::here('data', 'fun.xlsx')
```

```
[1] "C:/Users/Dunnigan_S/2021-infrequent-useR/data/fun.xlsx"
```

```
# read in 'fun' data
```

```
dat <- readxl::read_xlsx(here::here('data', 'fun.xlsx'))
```

Reproducibility:

Use standardized naming conventions

- **TL;DR** - machine and human readable and plays well with default ordering (put something numeric first).

Really great presentation by Jenny Bryan about naming files that she gave at the Reproducible Science Workshop:
<https://speakerdeck.com/jennybc/how-to-name-files>

“plays well with default ordering”

```
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H01.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H02.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H03.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_platefile.csv
2014-02-26_BRAFWTNEGASSAY_FFPEDNA-CRC-1-41_A01.csv
2014-02-26_BRAFWTNEGASSAY_FFPEDNA-CRC-1-41_A02.csv
2014-02-26_BRAFWTNEGASSAY_FFPEDNA-CRC-1-41_A03.csv
2014-02-26_BRAFWTNEGASSAY_FFPEDNA-CRC-1-41_A04.csv
```

```
01_marshall-data.r
02_pre-dea-filtering.r
03_dea-with-limma-voom.r
04_explore-dea-results.r
90_limma-model-term-name-fiasco.r
helper01_load-counts.r
helper02_load-exp-des.r
helper03_load-focus-statinf.r
helper04_extract-and-tidy.r
```

put something numeric first

From Jenny Bryan’s [“How to name things”](#) presentation.

Use the ISO 8601
standard for dates
(YYYY-MM-DD)

Deliberate use of “_” and “-” allows us to recover meta-data from the filenames.

```
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H01.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H02.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_H03.csv
2013-06-26_BRAFWTNEGASSAY_Plasmid-Cellline-100-1MutantFraction_platefile.csv
```

```
> flist <- list.files(pattern = "Plasmid") %>% head

> stringr::str_split_fixed(flist, "[_\\.]", 5)
      [,1]      [,2]      [,3]      [,4]      [,5]
[1,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "A01" "csv"
[2,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "A02" "csv"
[3,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "A03" "csv"
[4,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "B01" "csv"
[5,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "B02" "csv"
[6,] "2013-06-26" "BRAFWTNEGASSAY" "Plasmid-Cellline-100-1MutantFraction" "B03" "csv"

      date      assay      sample set      well
```

This happens to be R but also possible in the shell, Python, etc.

From Jenny Bryan’s [“How to name things”](#) presentation.

Examples

NO

fig 2.png

Jim's master of all data.xlsx

Plankton 2021 raw*edited.csv

abstract.docx

YES

fig02_linegraph-chla-timeseries.png

jims-filename-is-better-master.xlsx

2021-plankton-edited.csv

2021-09-23_abstract-for-wksp.docx

The `janitor` package

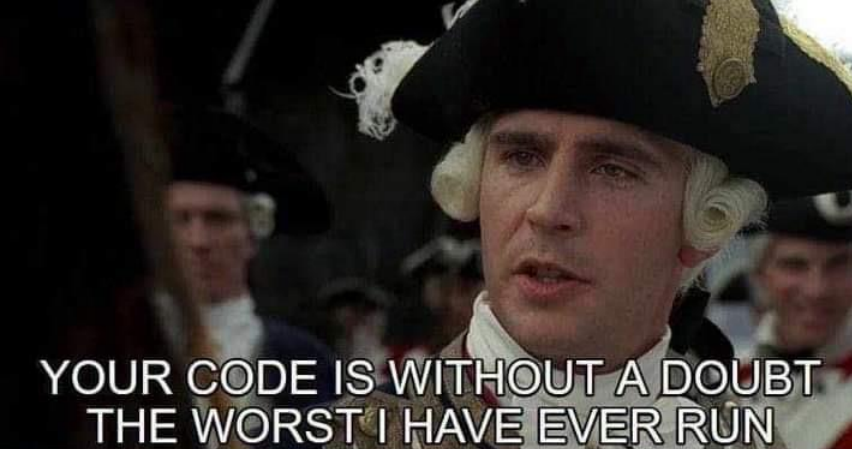


Artwork by
Allison Horst
(<https://github.com/allisonhorst>)

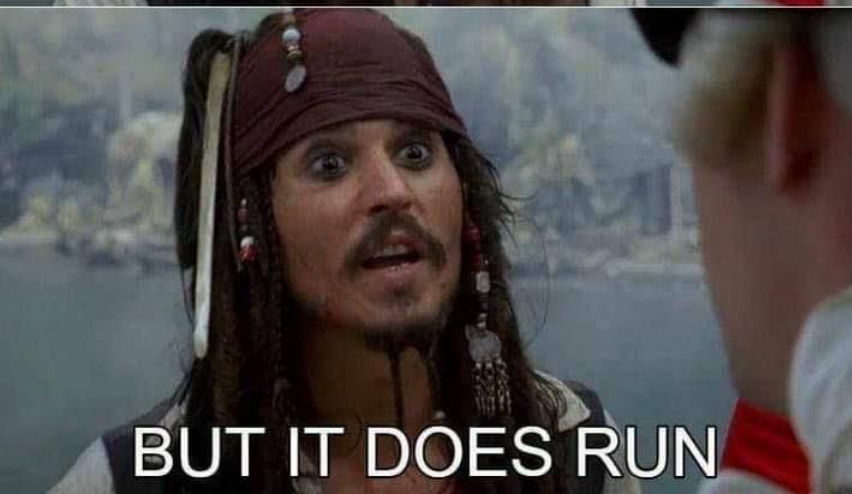
The `janitor` package

```
> head(env)
# A tibble: 6 x 16
  Site      StationCode Date           `Time (24 hr)` ActivityType ComponentShort ComponentLong
  <chr>      <chr>      <dtm>          <chr>          <chr>      <chr>      <chr>
1 PINE ISLAND gtmpinut1.1 2021-01-12 00:00:00 13:06      Field      Depth      Water depth
2 PINE ISLAND gtmpinut1.1 2021-01-12 00:00:00 13:06      Field      Depth_S    Sample depth
3 PINE ISLAND gtmpinut1.1 2021-01-12 00:00:00 13:06      Field      SECCHI     Secchi Disk
4 PINE ISLAND gtmpinut1.1 2021-01-12 00:00:00 13:06      Field      WIND_S     Wind Speed
```

```
> clean_names(env) %>% head()
# A tibble: 6 x 16
  site      station_code date           time_24_hr activity_type component_short component_long
  <chr>      <chr>      <dtm>          <chr>          <chr>      <chr>      <chr>
1 PINE ISLAND gtmpinut1.1 2021-01-12 00:00:00 13:06      Field      Depth      Water depth
2 PINE ISLAND gtmpinut1.1 2021-01-12 00:00:00 13:06      Field      Depth_S    Sample depth
3 PINE ISLAND gtmpinut1.1 2021-01-12 00:00:00 13:06      Field      SECCHI     Secchi Disk
4 PINE ISLAND gtmpinut1.1 2021-01-12 00:00:00 13:06      Field      WIND_S     Wind Speed
```

YOUR CODE IS WITHOUT A DOUBT
THE WORST I HAVE EVER RUN



BUT IT DOES RUN

Reproducibility: Coding style

“Good coding style is like using correct punctuation. You can manage without it, but it sure makes things easier to read.”

– Hadley Wickham, *Advanced R*

- Style Guides exist (like [Hadley's](#) or [Google](#))
- You may write it, but others will read it.
- Be consistent.
- Check out the ``formatR`` package by Yihui Xie

Image is shared from an R users Facebook group,
and I wish I could give the original creator credit.

Reproducibility: Coding Style

- Lowercase variable and object names with underscores (_) to separate words within a name.
- Spaces! Code is already difficult to read. `giveyoureyesabreak`
- Closing curly braces ``}`` go on their own line
- Use `<-` and not `=`, for assignment
- Pipes! ``%>%`` from the ``magrittr`` package

Additional Best Practice Tips

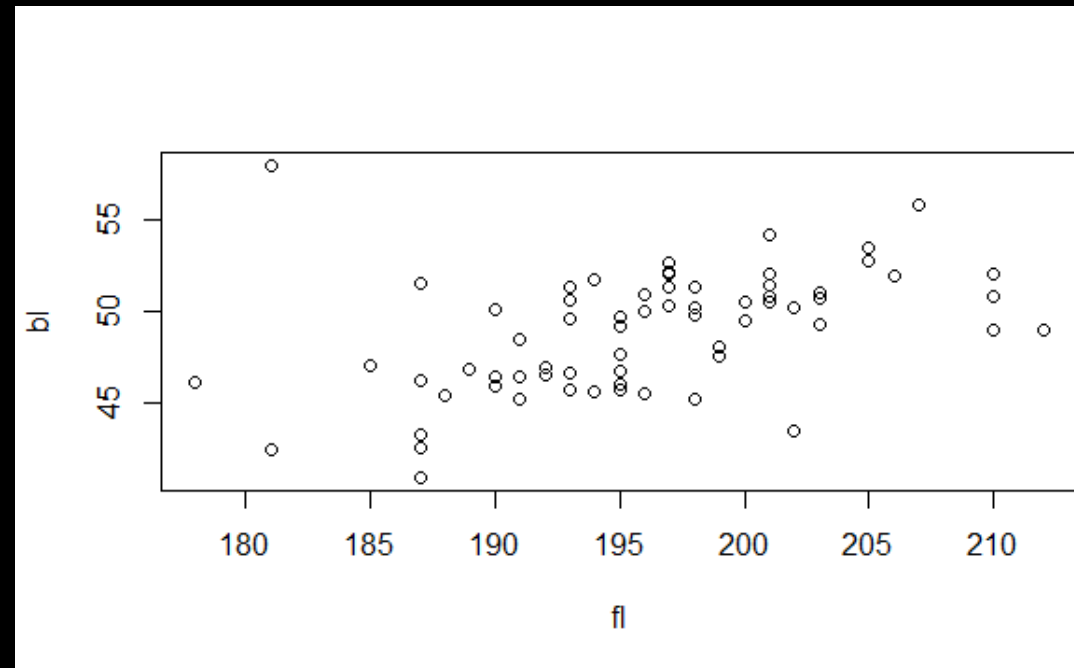
Tidy data, dataframes, and moving away from the spreadsheet

Leave it in the data frame

Adapted from Jenny Bryan's RStudio webinar "*Thinking inside the box: you can do that inside a dataframe?!*"

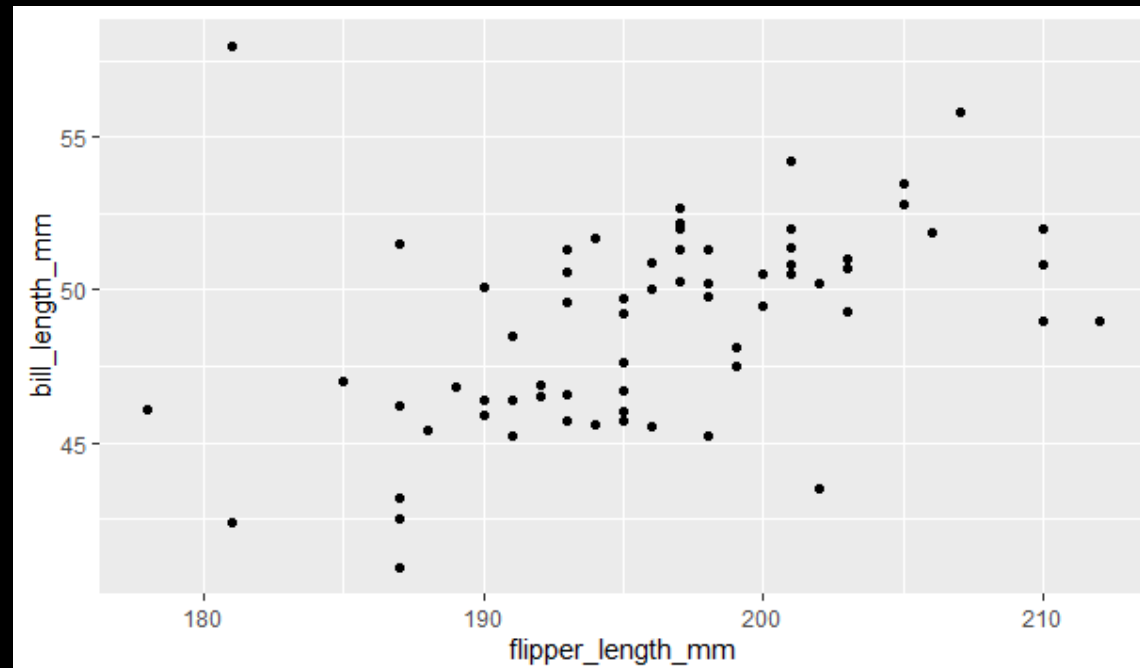
- Don't create little excerpts and copies of your data

```
library(palmerpenguins) # for the dataset  
bl <- penguins[277:344, 3]  
fl <- penguins[277:344, 5]  
plot(bl ~ fl)
```



Bring intent into your code

```
library(tidyverse)
penguins %>%
  dplyr::filter(species == "Chinstrap") %>%
  ggplot(aes(x = flipper_length_mm,
             y = bill_length_mm)) +
  geom_point()
```



Tidy data

country	year	cases	pop
AFG	1999	745	1999
AFG	2000	745	1999
BAN	1999	745	1999
BAN	2000	745	1999
CHN	1999	745	1999
CHN	2000	745	1999

A data set is **tidy** iff:

1. Each **variable** is in its own **column**
2. Each **observation** is in its own **row**
3. Each **value** is in its own **cell**

Highly recommend these papers:

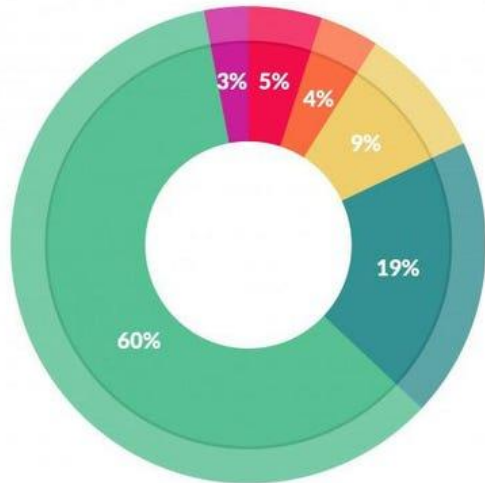
[Wickham, 2014: Tidy Data](#)

[Broman and Woo, 2017: Data Organization in Spreadsheets](#)

Tidy data

country	year	cases	pop
Afghanistan	2009	184336	22660130
Algeria	2009	143219	24154979
Algeria	2010	146674	24154979
Algeria	2011	149988	24154979
Algeria	2012	153302	24154979
Algeria	2013	156616	24154979
Algeria	2014	159930	24154979
Algeria	2015	163244	24154979
Algeria	2016	166558	24154979
Algeria	2017	169872	24154979
Algeria	2018	173186	24154979
Algeria	2019	176500	24154979
Algeria	2020	179814	24154979
Algeria	2021	183128	24154979
Algeria	2022	186442	24154979
Algeria	2023	189756	24154979
Algeria	2024	193070	24154979
Algeria	2025	196384	24154979
Algeria	2026	199698	24154979
Algeria	2027	203012	24154979
Algeria	2028	206326	24154979
Algeria	2029	209640	24154979
Algeria	2030	212954	24154979
Algeria	2031	216268	24154979
Algeria	2032	219582	24154979
Algeria	2033	222896	24154979
Algeria	2034	226210	24154979
Algeria	2035	229524	24154979
Algeria	2036	232838	24154979
Algeria	2037	236152	24154979
Algeria	2038	239466	24154979
Algeria	2039	242780	24154979
Algeria	2040	246094	24154979
Algeria	2041	249408	24154979
Algeria	2042	252722	24154979
Algeria	2043	256036	24154979
Algeria	2044	259350	24154979
Algeria	2045	262664	24154979
Algeria	2046	265978	24154979
Algeria	2047	269292	24154979
Algeria	2048	272606	24154979
Algeria	2049	275920	24154979
Algeria	2050	279234	24154979
Algeria	2051	282548	24154979
Algeria	2052	285862	24154979
Algeria	2053	289176	24154979
Algeria	2054	292490	24154979
Algeria	2055	295804	24154979
Algeria	2056	299118	24154979
Algeria	2057	302432	24154979
Algeria	2058	305746	24154979
Algeria	2059	309060	24154979
Algeria	2060	312374	24154979
Algeria	2061	315688	24154979
Algeria	2062	319002	24154979
Algeria	2063	322316	24154979
Algeria	2064	325630	24154979
Algeria	2065	328944	24154979
Algeria	2066	332258	24154979
Algeria	2067	335572	24154979
Algeria	2068	338886	24154979
Algeria	2069	342200	24154979
Algeria	2070	345514	24154979
Algeria	2071	348828	24154979
Algeria	2072	352142	24154979
Algeria	2073	355456	24154979
Algeria	2074	358770	24154979
Algeria	2075	362084	24154979
Algeria	2076	365398	24154979
Algeria	2077	368712	24154979
Algeria	2078	372026	24154979
Algeria	2079	375340	24154979
Algeria	2080	378654	24154979
Algeria	2081	381968	24154979
Algeria	2082	385282	24154979
Algeria	2083	388596	24154979
Algeria	2084	391910	24154979
Algeria	2085	395224	24154979
Algeria	2086	398538	24154979
Algeria	2087	401852	24154979
Algeria	2088	405166	24154979
Algeria	2089	408480	24154979
Algeria	2090	411794	24154979
Algeria	2091	415108	24154979
Algeria	2092	418422	24154979
Algeria	2093	421736	24154979
Algeria	2094	425050	24154979
Algeria	2095	428364	24154979
Algeria	2096	431678	24154979
Algeria	2097	434992	24154979
Algeria	2098	438306	24154979
Algeria	2099	441620	24154979
Algeria	2100	444934	24154979
Algeria	2101	448248	24154979
Algeria	2102	451562	24154979
Algeria	2103	454876	24154979
Algeria	2104	458190	24154979
Algeria	2105	461504	24154979
Algeria	2106	464818	24154979
Algeria	2107	468132	24154979
Algeria	2108	471446	24154979
Algeria	2109	474760	24154979
Algeria	2110	478074	24154979
Algeria	2111	481388	24154979
Algeria	2112	484702	24154979
Algeria	2113	488016	24154979
Algeria	2114	491330	24154979
Algeria	2115	494644	24154979
Algeria	2116	497958	24154979
Algeria	2117	501272	24154979
Algeria	2118	504586	24154979
Algeria	2119	507900	24154979
Algeria	2120	511214	24154979
Algeria	2121	514528	24154979
Algeria	2122	517842	24154979
Algeria	2123	521156	24154979
Algeria	2124	524470	24154979
Algeria	2125	527784	24154979
Algeria	2126	531098	24154979
Algeria	2127	534412	24154979
Algeria	2128	537726	24154979
Algeria	2129	541040	24154979
Algeria	2130	544354	24154979
Algeria	2131	547668	24154979
Algeria	2132	550982	24154979
Algeria	2133	554296	24154979
Algeria	2134	557610	24154979
Algeria	2135	560924	24154979
Algeria	2136	564238	24154979
Algeria	2137	567552	24154979
Algeria	2138	570866	24154979
Algeria	2139	574180	24154979
Algeria	2140	577494	24154979
Algeria	2141	580808	24154979
Algeria	2142	584122	24154979
Algeria	2143	587436	24154979
Algeria	2144	590750	24154979
Algeria	2145	594064	24154979
Algeria	2146	597378	24154979
Algeria	2147	600692	24154979
Algeria	2148	604006	24154979
Algeria	2149	607320	24154979
Algeria	2150	610634	24154979
Algeria	2151	613948	24154979
Algeria	2152	617262	24154979
Algeria	2153	620576	24154979
Algeria	2154	623890	24154979
Algeria	2155	627204	24154979
Algeria	2156	630518	24154979
Algeria	2157	633832	24154979
Algeria	2158	637146	24154979
Algeria	2159	640460	24154979
Algeria	2160	643774	24154979
Algeria	2161	647088	24154979
Algeria	2162	650402	24154979
Algeria	2163	653716	24154979
Algeria	2164	657030	24154979
Algeria	2165	660344	24154979
Algeria	2166	663658	24154979
Algeria	2167	666972	24154979
Algeria	2168	670286	24154979
Algeria	2169	673600	24154979
Algeria	2170	676914	24154979
Algeria	2171	680228	24154979
Algeria	2172	683542	24154979
Algeria	2173	686856	24154979
Algeria	2174	690170	24154979
Algeria	2175	693484	24154979
Algeria	2176	696798	24154979
Algeria	2177	700112	24154979
Algeria	2178	703426	24154979
Algeria	2179	706740	24154979
Algeria	2180	710054	24154979
Algeria	2181	713368	24154979
Algeria	2182	716682	24154979
Algeria	2183	719996	24154979
Algeria	2184	723310	24154979
Algeria	2185	726624	24154979
Algeria	2186	729938	24154979
Algeria	2187	733252	24154979
Algeria	2188	736566	24154979
Algeria	2189	739880	24154979
Algeria	2190	743194	24154979
Algeria	2191	746508	24154979
Algeria	2192	749822	24154979
Algeria	2193	753136	24154979
Algeria	2194	756450	24154979
Algeria	2195	759764	24154979
Algeria	2196	763078	24154979
Algeria	2197	766392	24154979
Algeria	2198	769706	24154979
Algeria	2199	773020	24154979
Algeria	2200	776334	24154979
Algeria	2201	779648	24154979
Algeria	2202	782962	24154979
Algeria	2203	786276	24154979
Algeria	2204	789590	24154979
Algeria	2205	792904	24154979
Algeria	2206	796218	24154979
Algeria	2207	799532	24154979
Algeria	2208	802846	24154979
Algeria	2209	806160	24154979
Algeria	2210	809474	24154979
Algeria	2211	812788	24154979
Algeria	2212	816102	24154979
Algeria	2213	819416	24154979
Algeria	2214	822730	24154979
Algeria	2215	826044	24154979
Algeria	2216	829358	24154979
Algeria	2217	832672	24154979
Algeria	2218	835986	24154979
Algeria	2219	839300	24154979
Algeria	2220	842614	24154979
Algeria	2221	845928	24154979
Algeria	2222	849242	24154979
Algeria	2223	852556	24154979
Algeria	2224	855870	24154979
Algeria	2225	859184	24154979
Algeria	2226	862498	24154979
Algeria	2227	865812	24154979
Algeria	2228	869126	24154979
Algeria	2229	872440	24154979
Algeria	2230	875754	24154979
Algeria	2231	879068	24154979
Algeria	2232	882382	24154979
Algeria	2233	885696	24154979
Algeria	2234	889010	24154979
Algeria	2235	892324	24154979
Algeria	2236	895638	24154979
Algeria	2237	898952	24154979
Algeria	2238	902266	24154979
Algeria	2239	905580	24154979
Algeria	2240	908894	24154979
Algeria	2241	912208	24154979
Algeria	2242	915522	24154979
Algeria	2243	918836	24154979
Algeria	2244	922150	24154979
Algeria	2245	925464	24154979
Algeria	2246	928778	24154979
Algeria	2247	932092	24154979
Algeria	2248	935406	24154979
Algeria	2249	938720	24154979
Algeria	2250	942034	24154979
Algeria	2251	945348	24154979
Algeria	2252	948662	24154979
Algeria	2253	951976	24154979
Algeria	2254	955290	24154979
Algeria	2255	958604	24154979
Algeria	2256	961918	24154979
Algeria	2257	965232	24154979
Algeria	2258	968546	24154979
Algeria	2259	971860	24154979
Algeria	2260	975174	24154979
Algeria	2261	978488	24154979
Algeria	2262	981802	24154979
Algeria	2263	985116	24154979
Algeria	2264	988430	24154979
Algeria	2265	991744	24154979
Algeria	2266	995058	24154979
Algeria	2267	998372	24154979
Algeria	2268	1001686	24154979
Algeria	2269	1005000	24154979
Algeria	2270	1008314	24154979
Algeria	2271	1011628	24154979
Algeria	2272	1014942	24154979
Algeria	2273	1018256	24154979
Algeria	2274	1021570	24154979
Algeria	2275	1024884	24154979
Algeria	2276	1028198	24154979
Algeria	2277	1031512	24154979
Algeria	2278	1034826	24154979
Algeria	2279	1038140	24154979
Algeria	2280	1041454	241549

“...most time-consuming, least enjoyable data science task...”



What data scientists spend the most time doing

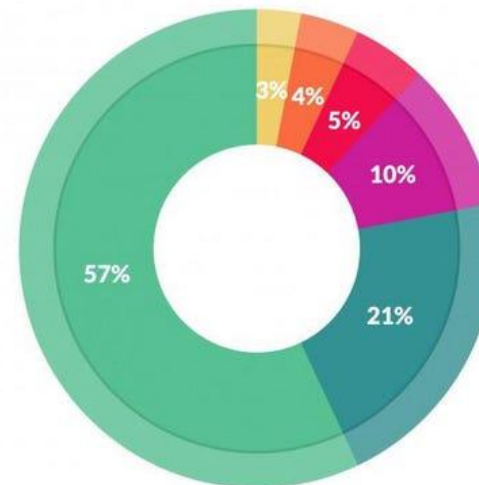
- Building training sets: 3%
- Cleaning and organizing data: 60%
- Collecting data sets: 19%
- Mining data for patterns: 9%
- Refining algorithms: 4%
- Other: 5%

Forbes

Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says

 **Gil Press** Senior Contributor 
Enterprise & Cloud
I write about technology, entrepreneurs and innovation.

[Follow](#)

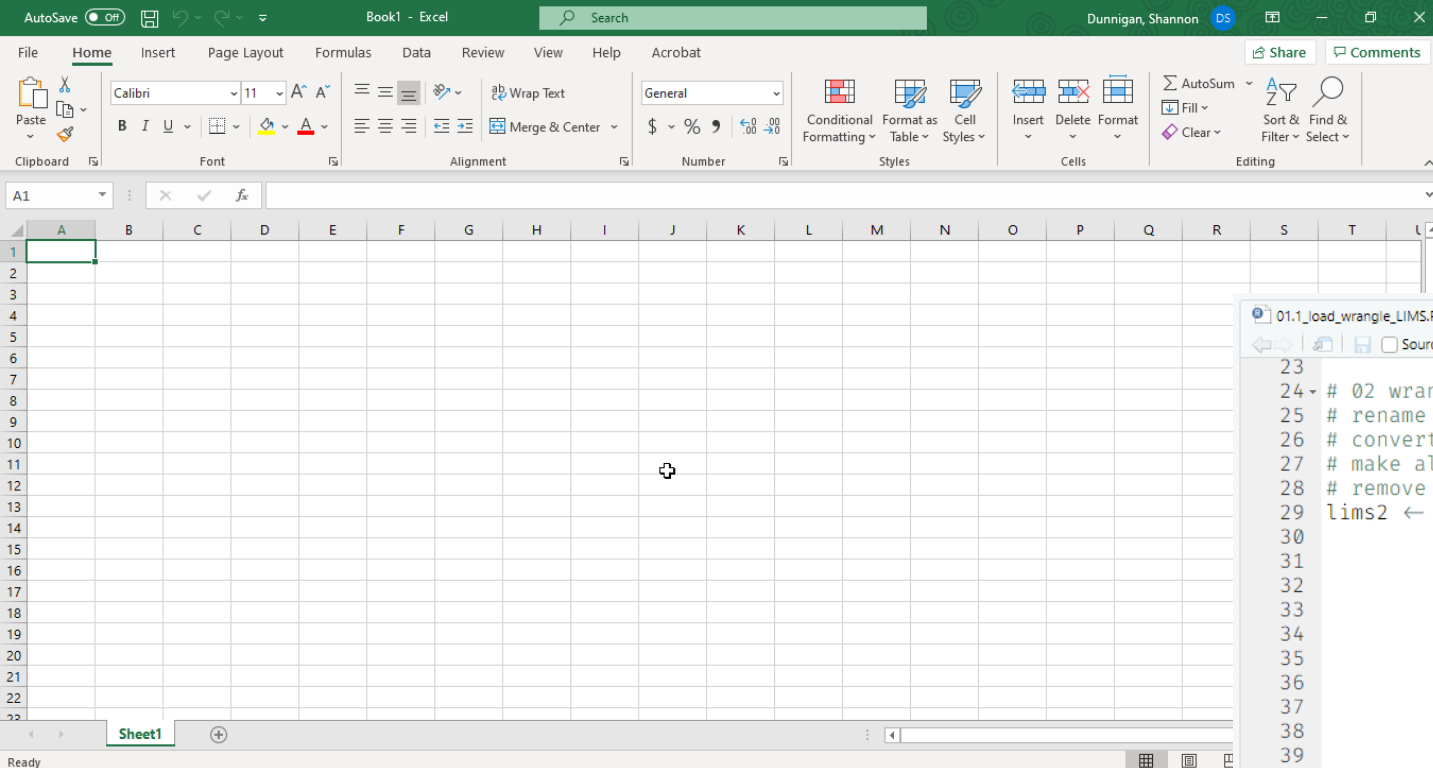


What's the least enjoyable part of data science?

- Building training sets: 10%
- Cleaning and organizing data: 57%
- Collecting data sets: 21%
- Mining data for patterns: 3%
- Refining algorithms: 4%
- Other: 5%



Overcoming the spreadsheet hurdle

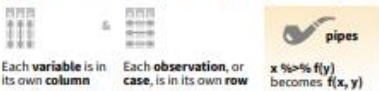


```
01.1_load_wrangle_LIMS.R
23
24 # 02 wrangle-tidy data
25 # rename some columns in lims to what we use in SWMP
26 # convert datetimes into POSIXct format
27 # make all entries in station_code and component_long columns lowercase (easier coding)
28 # remove field blanks
29 lims2 <- lims %>%
30   dplyr::rename(station_code = field_id,
31                 component_long = component,
32                 datetimestamp = date_sampled) %>%
33   dplyr::mutate(datetimestamp = as.POSIXct(strptime(datetimestamp,
34                                                    "%d-%b-%Y %H:%M", tz='EST')),
35                date_analyzed = as.POSIXct(strptime(date_analyzed,
36                                                    "%d-%b-%Y %H:%M", tz='EST')),
37                station_code = tolower(station_code),
38                component_long = tolower(component_long)) %>%
39   dplyr::filter(station_code != "field blank")
40
41 # correct so that TKN and TKN-F are different
42 # fixing the LIMS entry so that kjeldahl nitrogen, dissolved is different from kjeldahl nitrogen
43 tkn_f <- lims2 %>%
44   dplyr::filter(analysis == "W-TKN-F") %>%
45   dplyr::mutate(component_long = "kjeldahl nitrogen, dissolved")
46
47 # merge the renamed TKNF data with all the other data and then join with the `names` df to get t
48 lims3 <- lims2 %>%
49   dplyr::filter(analysis != "W-TKN-F") %>%
50   dplyr::bind_rows(tkn_f) %>%
51   dplyr::left_join(names, by = "component_long") %>%
52   dplyr::mutate(cdm_name = forcats::as_factor(cdm_name)) # I think conv. to factor hel
53
54 ## clean up environment ---
55 rm(tkn_f, lims2, names)
56
57 # 03 make LIMS data wide
58 # to make the LIMS data in wide format for entry into in-house datafile
59
60 lims_wide_results <- lims3 %>%
61   dplyr::select(station_code, datetimestamp, cdm_name, component_long, re
62
```

Data transformation with dplyr : CHEAT SHEET

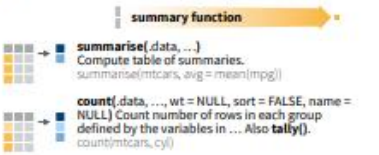


dplyr functions work with pipes and expect tidy data. In tidy data:



Summarise Cases

Apply **summary functions** to columns to create a new table of summary statistics. Summary functions take vectors as input and return one value (see back).



Group Cases

Use **group_by(data, ...)**, add = FALSE, drop = TRUE) to create a "grouped" copy of a table grouped by columns in ... dplyr functions will manipulate each "group" separately and combine the results.



Use **rowwise(data, ...)** to group data into individual rows. dplyr functions will compute results for each row. Also apply functions to list-columns. See tidy cheat sheet for list-column workflow.

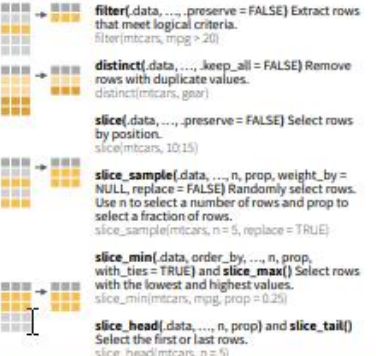


ungroup(x, ...) Returns ungrouped copy of table. `ungroup(mtcars)`

Manipulate Cases

EXTRACT CASES

Row functions return a subset of rows as a new table.



Logical and boolean operators to use with filter()

== < <= > >= !is.na() %in% | xor() != > != is.na() ! &

ARRANGE CASES



ADD CASES



Manipulate Variables

EXTRACT VARIABLES

Column functions return a set of columns as a new vector or table.



Use these helpers with select() and across() e.g. select(mtcars, mpg:cyl)

contains(match) num_range(prefix, range) i.e. mpg:cyl ends_with(match) all_of()/any_of(x, ..., vars) e.g. gear starts_with(match) matches(match) everything()

MANIPULATE MULTIPLE VARIABLES AT ONCE



MAKE NEW VARIABLES

Apply **vectorized functions** to columns. Vectorized functions take vectors as input and return vectors of the same length as output (see back).



Vectorized Functions

TO USE WITH MUTATE()

mutate() and **transmute()** apply vectorized functions to columns to create new columns. Vectorized functions take vectors as input and return vectors of the same length as output.

vectorized function

OFFSET

dplyr::lag() - offset elements by 1
dplyr::lead() - offset elements by -1

CUMULATIVE AGGREGATE

dplyr::cumall() - cumulative all()
dplyr::cumany() - cumulative any()
dplyr::cummax() - cumulative max()
dplyr::cummean() - cumulative mean()
dplyr::cummin() - cumulative min()
dplyr::cumprod() - cumulative prod()
dplyr::cumsum() - cumulative sum()

RANKING

dplyr::cume_dist() - proportion of all values <=
dplyr::dense_rank() - rank w ties = min, no gaps
dplyr::min_rank() - rank with ties = min
dplyr::ntble() - bins into n bins
dplyr::percent_rank() - min_rank scaled to [0,1]
dplyr::row_number() - rank with ties = "first"

MATH

+, -, *, /, ^, %, %%, %%% - arithmetic ops
log(), **log2()**, **log10()** - logs
dplyr::between() - x >= left & x <= right
dplyr::near() - safe == for floating point numbers

MISCELLANEOUS

dplyr::case_when() - multi-case if_else()
`stars %>% mutate(type = case_when(height > 200 ~ "large", species == "Droid" ~ "robot", TRUE ~ "other"))`

dplyr::coalesce() - first non-NA values by element across a set of vectors
dplyr::if_else() - element-wise if() + else()
dplyr::na_if() - replace specific values with NA
dplyr::pmax() - element-wise max()
dplyr::pmin() - element-wise min()

Summary Functions

TO USE WITH SUMMARISE()

summarise() applies summary functions to columns to create a new table. Summary functions take vectors as input and return single values as output.

summary function

COUNT

dplyr::n() - number of values/rows
dplyr::n_distinct() - # of uniques
sum(is.na()) - # of non-NA's

POSITION

mean() - mean, also **mean(is.na())**
median() - median

LOGICAL

sum() - proportion of TRUE's
sum() - # of TRUE's

ORDER

dplyr::first() - first value
dplyr::last() - last value
dplyr::nth() - value in nth location of vector

RANK

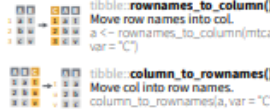
quantile() - nth quantile
min() - minimum value
max() - maximum value

SPREAD

IQR() - Inter-Quartile Range
mad() - median absolute deviation
sd() - standard deviation
var() - variance

Row Names

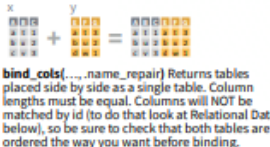
Tidy data does not use rownames, which store a variable outside of the columns. To work with the rownames, first move them into a column.



Also **tibble::has_rownames()** and **tibble::remove_rownames()**

Combine Tables

COMBINE VARIABLES



RELATIONAL DATA

Use a **"Mutating Join"** to join one table to columns from another, matching values with the rows that they correspond to. Each join retains a different combination of values from the tables.

left_join(x, y, by = NULL, copy = FALSE, suffix = c("x", "y"), ..., keep = FALSE, na_matches = "na") Join matching values from x to y.

right_join(x, y, by = NULL, copy = FALSE, suffix = c("x", "y"), ..., keep = FALSE, na_matches = "na") Join matching values from x to y.

inner_join(x, y, by = NULL, copy = FALSE, suffix = c("x", "y"), ..., keep = FALSE, na_matches = "na") Join data. Retain only rows with matches.

full_join(x, y, by = NULL, copy = FALSE, suffix = c("x", "y"), ..., keep = FALSE, na_matches = "na") Join data. Retain all values, all rows.

COLUMN MATCHING FOR JOINS

Use **by = c("col1", "col2", ...)** to specify one or more common columns to match on. `left_join(x, y, by = "A")`

Use a named vector, **by = c("col1" = "col2")**, to match on columns that have different names in each table. `left_join(x, y, by = c("C" = "D"))`

Use **suffix** to specify the suffix to give to unmatched columns that have the same name in both tables. `left_join(x, y, by = c("C" = "D"), suffix = c("1", "2"))`

COMBINE CASES



RELATIONAL DATA

Use a **"Filtering Join"** to filter one table against the rows of another.

semi_join(x, y, by = NULL, copy = FALSE, suffix = c("x", "y"), ..., keep = FALSE, na_matches = "na") Return rows of x that have a match in y. Use to see what will be included in a join.

anti_join(x, y, by = NULL, copy = FALSE, suffix = c("x", "y"), ..., keep = FALSE, na_matches = "na") Return rows of x that do not have a match in y. Use to see what will not be included in a join.

Use a **"Nest Join"** to inner join one table to another into a nested data frame.

nest_join(x, y, by = NULL, copy = FALSE, keep = FALSE, name = NULL, ...) Join data, nesting matches from y in a single new data frame column.

SET OPERATIONS

intersect(x, y, ...) Rows that appear in both x and y.

setdiff(x, y, ...) Rows that appear in x but not y.

union(x, y, ...) Rows that appear in x or y. (Duplicates removed). **union_all()** retains duplicates.

Use **setequal()** to test whether two data sets contain the exact same rows (in any order).





Your turn: What kinds of things have you found helpful?



A dark blue background featuring a faint, semi-transparent financial candlestick chart. The chart shows price fluctuations over time, with various numerical values and lines indicating trends. The overall aesthetic is professional and data-oriented.

A Reproducible Report

[https://github.com/skdunnigan/
2021-infrequent-useR](https://github.com/skdunnigan/2021-infrequent-useR)

What is R Markdown?

- File format for making document using R
- Written in markdown (easy, plain text format)
- Chunks of embedded code
- Designed to be used with the `rmarkdown` package
- “render” combines knit and convert to produce the file

rmarkdown :: CHEAT SHEET

What is rmarkdown?



Workflow

1. Open a new .Rmd file in the RStudio IDE by going to File > New File > R Markdown.
2. Embed code in chunks. Run code by line, by chunk, or all at once.
3. Write text and add tables, figures, images, and citations. Format with Markdown syntax or the RStudio Visual Markdown Editor.
4. Set output format(s) and options in the YAML header. Customize themes or add parameters to execute or add interactivity with Shiny.
5. Save and render the whole document. Knit periodically to preview your work as you write.
6. Share your work!

Embed Code with knitr

CODE CHUNKS

Surround code chunks with ````{r}` and ````` or use the Insert Code Chunk button. Add a chunk label and/or chunk options inside the curly braces after `r`.

```
```{r chunk-label, include=FALSE}
summary(mtcars)
```
```

SET GLOBAL OPTIONS

Set options for the entire document in the first chunk.

```
```{r include=FALSE}
knitr::opts_chunk$set(message = FALSE)
```
```

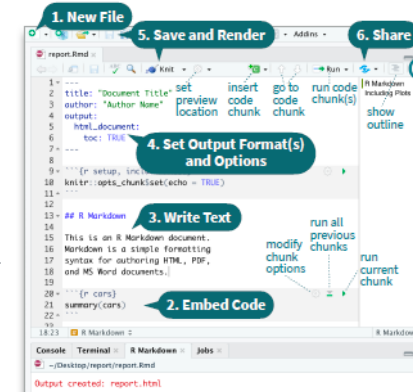
INLINE CODE

Insert ``r`` into text sections. Code is evaluated at render and results appear as text.

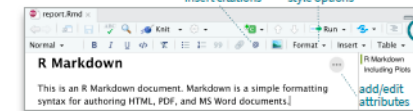
*Built with ``r` getRversion()` * *Built with 4.1.0*



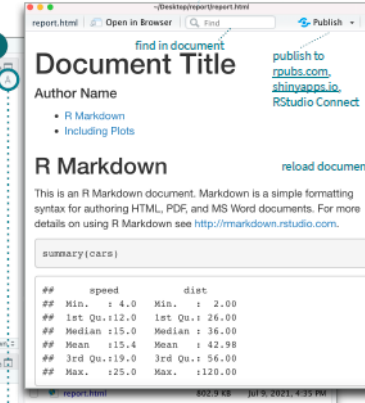
SOURCE EDITOR



VISUAL EDITOR



RENDERED OUTPUT



Insert Citations

Create citations from a bibliography file, a Zotero library, or from DOI references.

BUILD YOUR BIBLIOGRAPHY

- Add BibTeX or CSL bibliographies to the YAML header.

```
title: "My Document"
bibliography: references.bib
link-citations: TRUE
```

- If Zotero is installed locally, your main library will automatically be available.

- Add citations by DOI by searching "from DOI" in the Insert Citation dialog.

INSERT CITATIONS

- Access the Insert Citations dialog in the Visual Editor by clicking the @ symbol in the toolbar or by clicking Insert > Citation.

- Add citations with markdown syntax by typing `@cite` or `@cite`.

Insert Tables

Output data frames as tables using `kable(data, caption)`.

```
```{r}
data <- faithful[1:4,]
knitr::kable(data,
 caption = "Table with kable")
```
```

Other table packages include `flextable`, `gt`, and `kableExtra`.

Write with Markdown

The syntax on the left renders:

Plain text.

End a line with two spaces to

start a new paragraph.

Also end with a backslash

to make a new line.

Italics and **bold**

superscript² and subscript₂

~strikethrough~

escaped: _ _ _

endash: ---, emdash: ---

Header 1

Header 2

Header 6

- item 1

- item 2a (indent 1 tab)

- item 2b

1. ordered list

2. item 2

- item 2a (indent 1 tab)

- item 2b

<link url>

[This is a link.](link url)

[This is another link](id).

At the end of the document:

![]: link url

![]: link url

At the end of the document:

![]: image.png

![]: image.png

verbatim code

...

multiple lines

of verbatim code

> block quotes

equation: $\pi \times r^2 \times h$

equation block: $\pi \times r^2 \times h$

horizontal rule: ---

Right | Left | Default | Center

12 | 12 | 12 | 12

123 | 123 | 123 | 123

1 | 1 | 1 | 1

HTML Tablesets

Results {tabset}

Plots text

text

Tables

more text