

**Reproducibility**

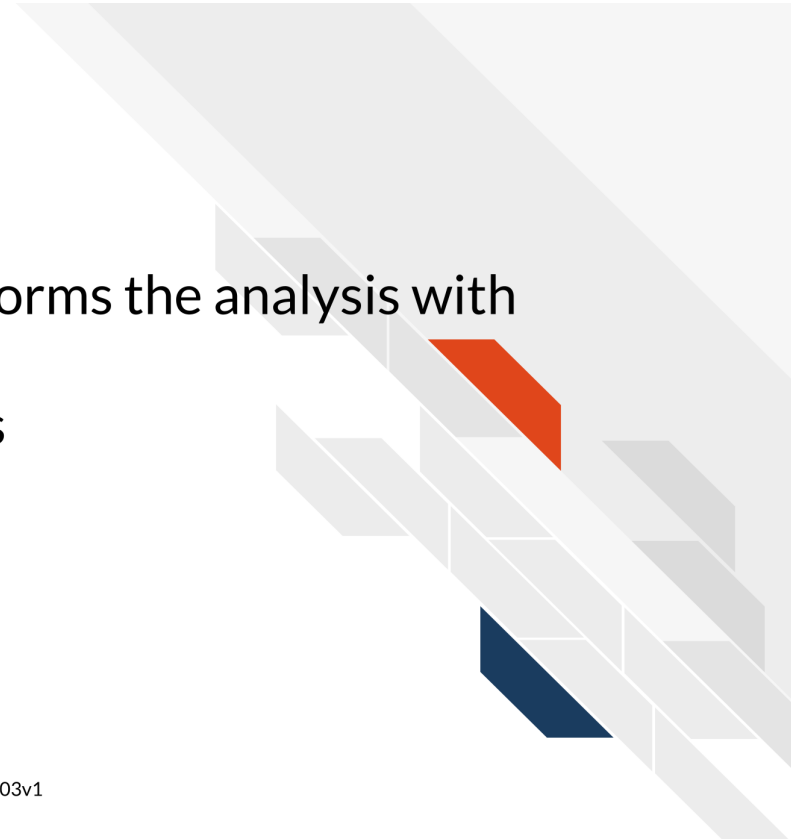
# What's Reproducibility

## Reproducibility:

a different analyst re-performs the analysis with  
the **same code** and  
the **same data** and obtains  
the **same result**.

Patil, Peng, Leek (2016) <https://www.biorxiv.org/content/10.1101/066803v1>

Content adapted from [Candace Savonen](#).



My data analysis is showing a pattern that is very informative for the ongoing research in my field.



Ruby the Researcher

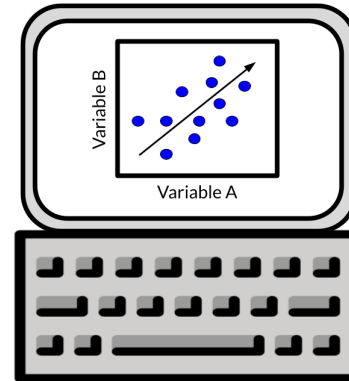
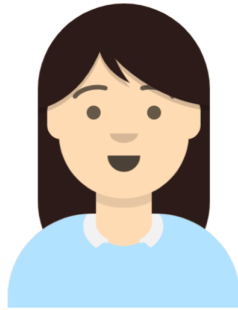


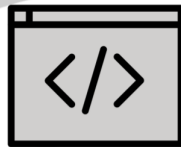
Image created by Candace Savonen using Avataars.

**Repeatable:** keeping everything the same but repeating the analysis -  
do we get the same results?

Ruby the  
Researcher



Code



Data



Results

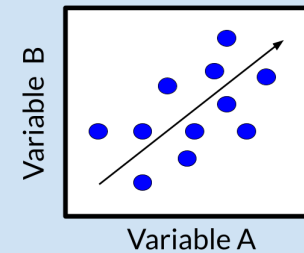
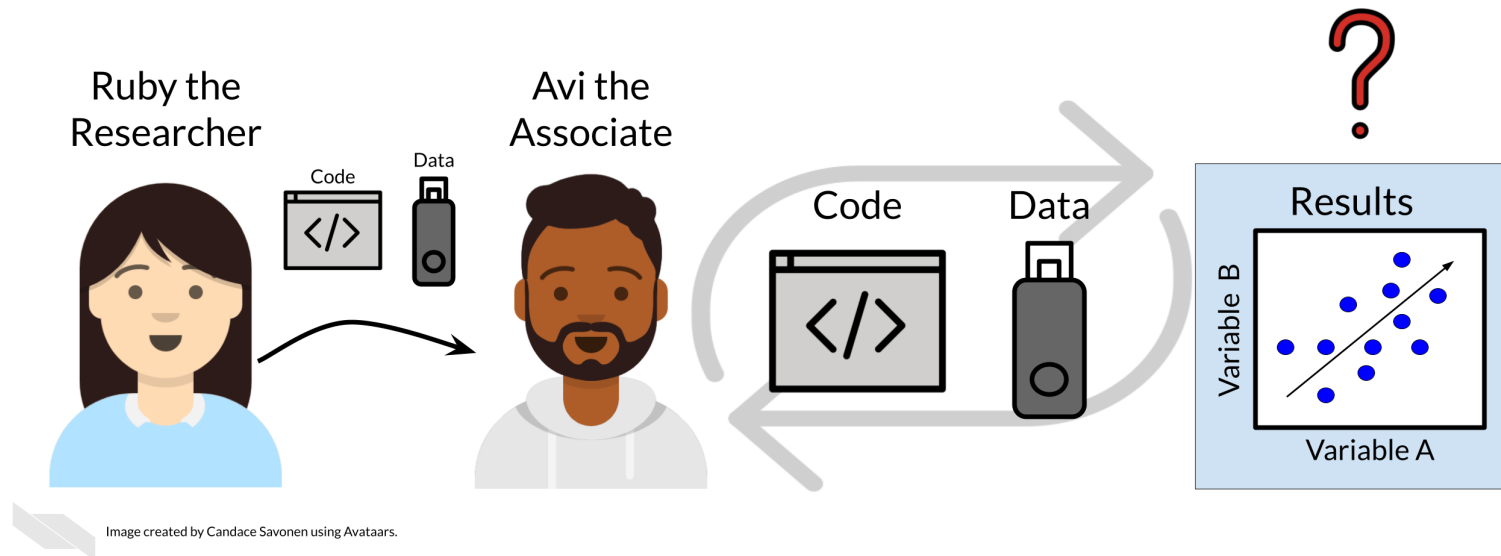
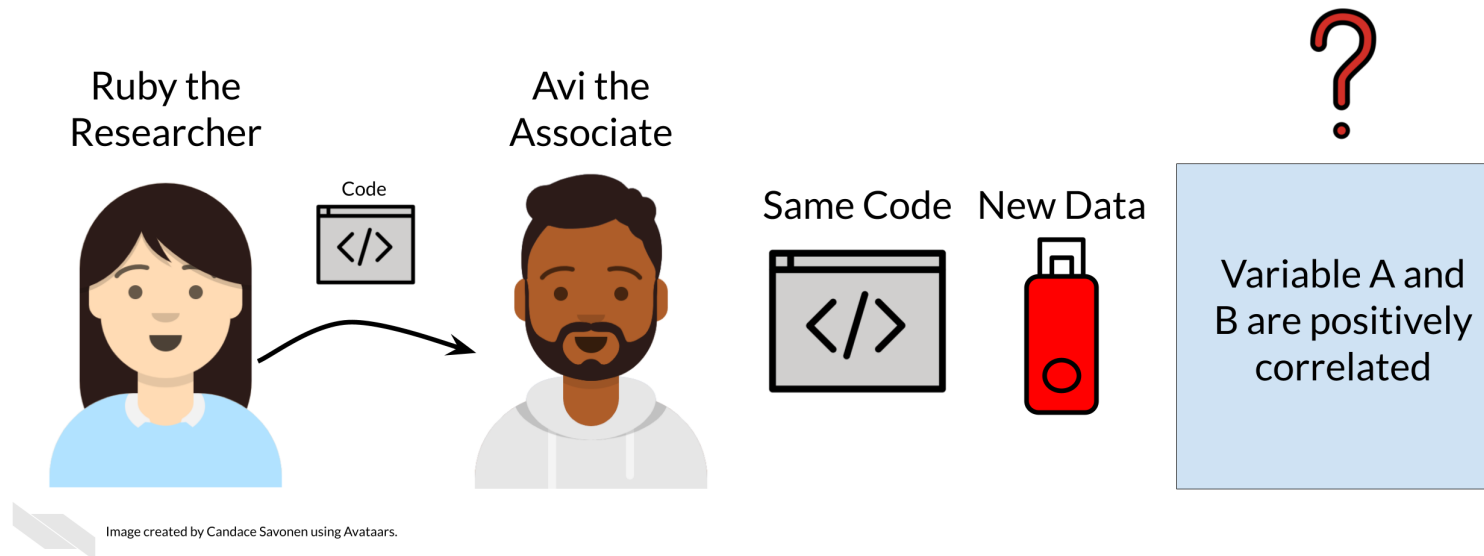


Image created by Candace Savonen using Avataars.

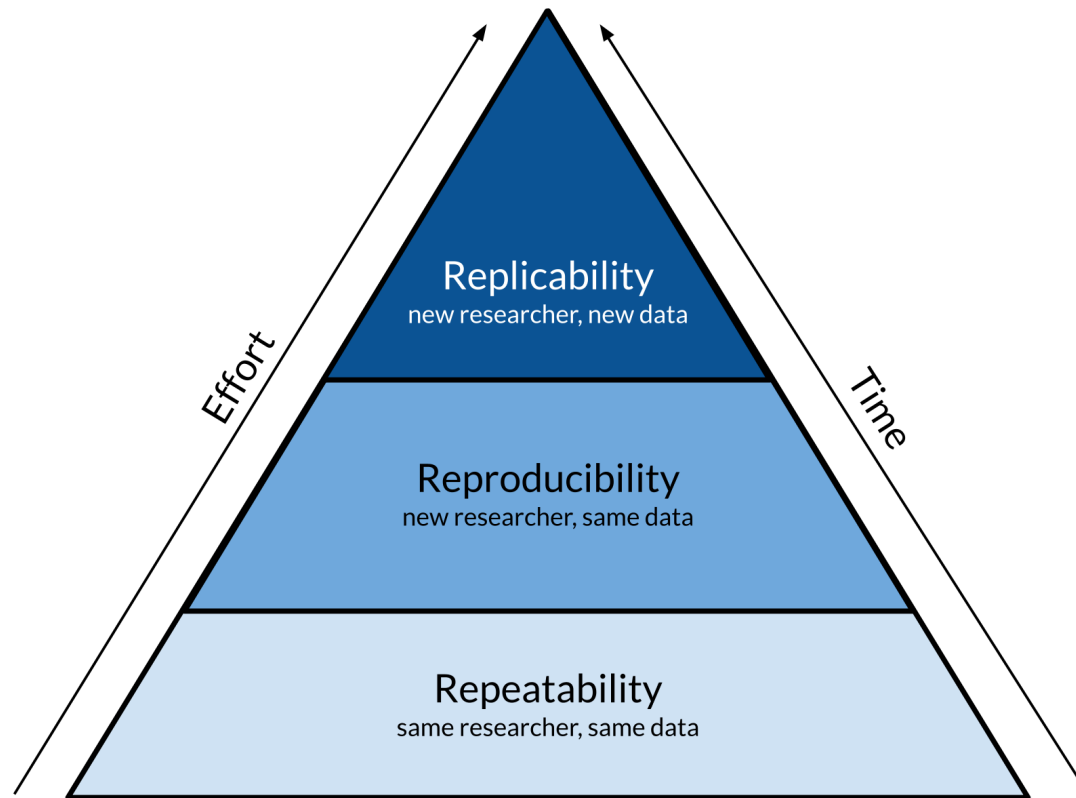
**Reproducible:** using the same data and analysis but in the hands of *another researcher* - do we get the same results?



# Replicable: with new data do we obtain the same inferences?



# Reproducibility vs Repeatability vs Replicability



Based off of a figure from Essawy et al, 2020 <https://doi.org/10.1016/j.envsoft.2020.104753>

# Why Reproducibility is important...

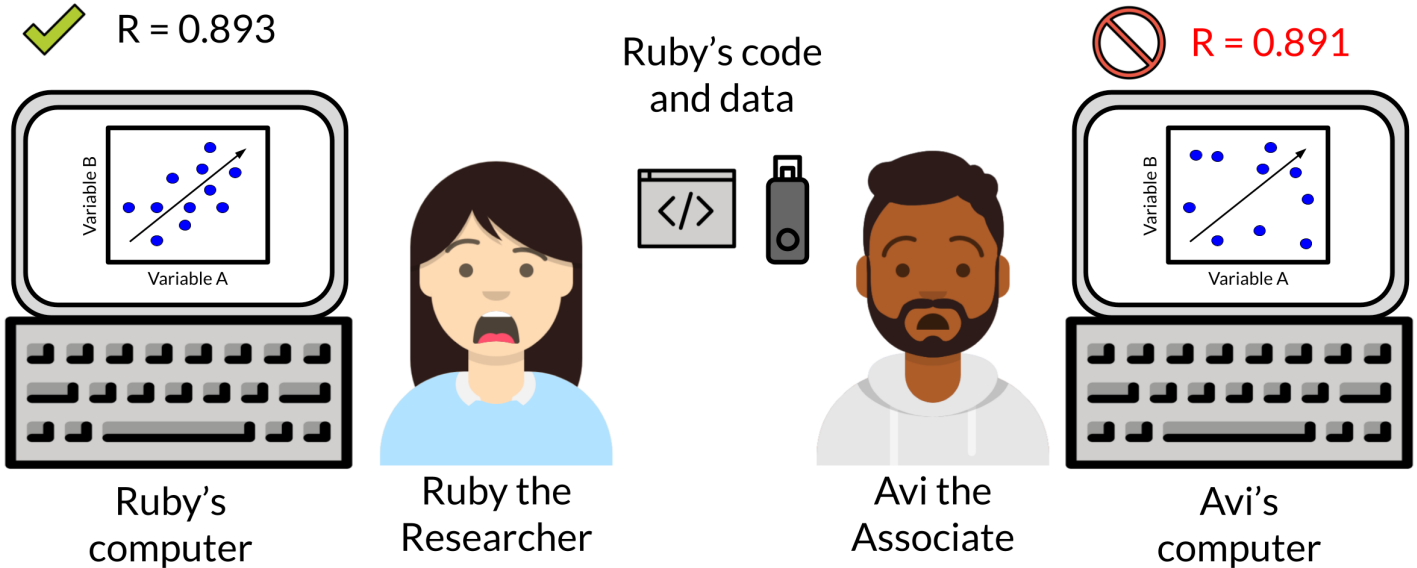
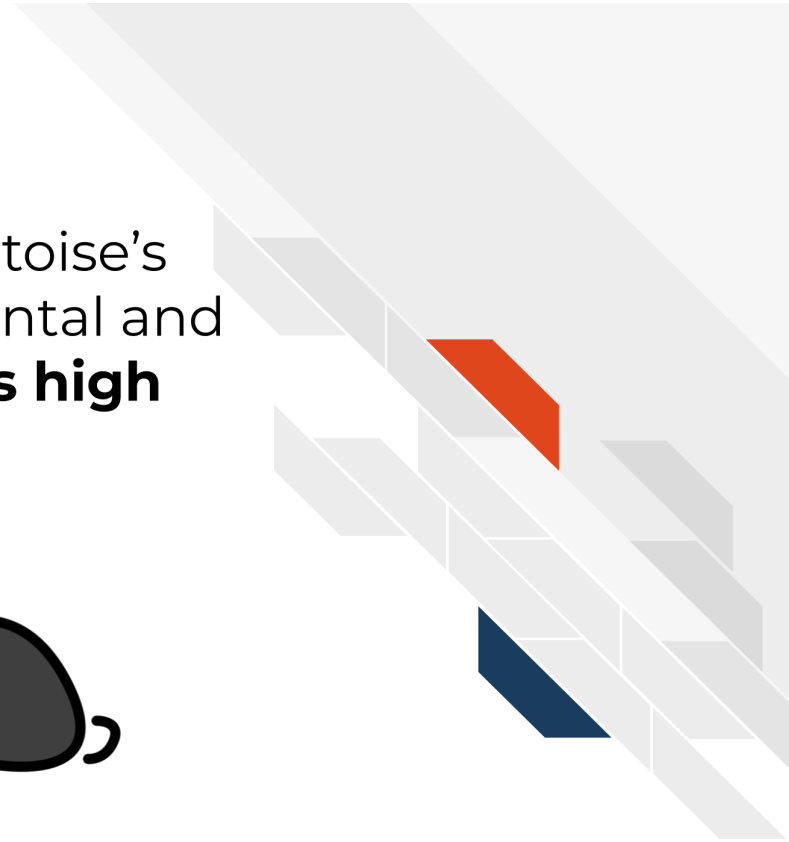
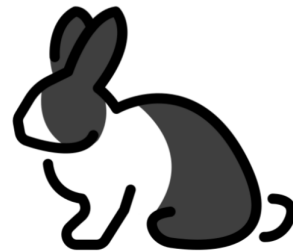
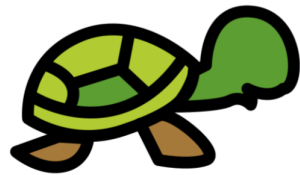


Image created by Candace Savonen using Avataars.



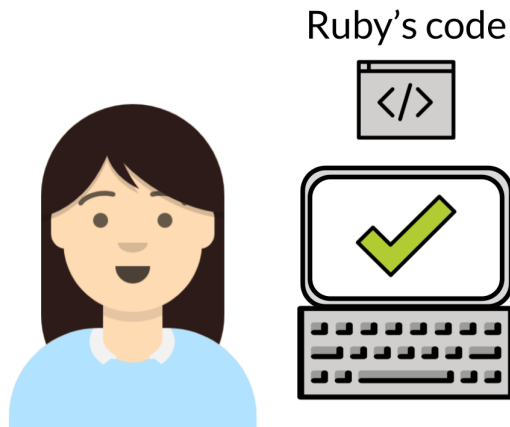
# It's worth the wait

Reproducibility is a tortoise's game - it's an incremental and slow process *but* **it has high payoffs!**

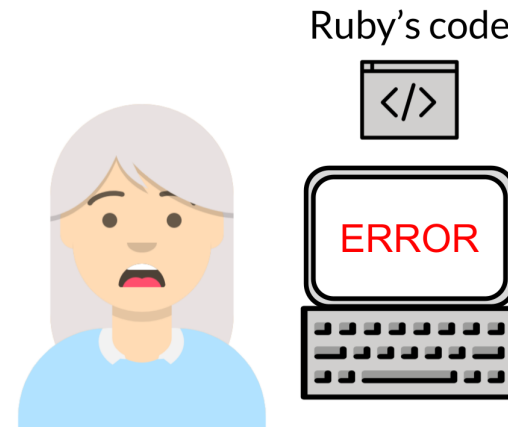



# Reproducibility can also be for your future self!

Now Ruby



Future Ruby



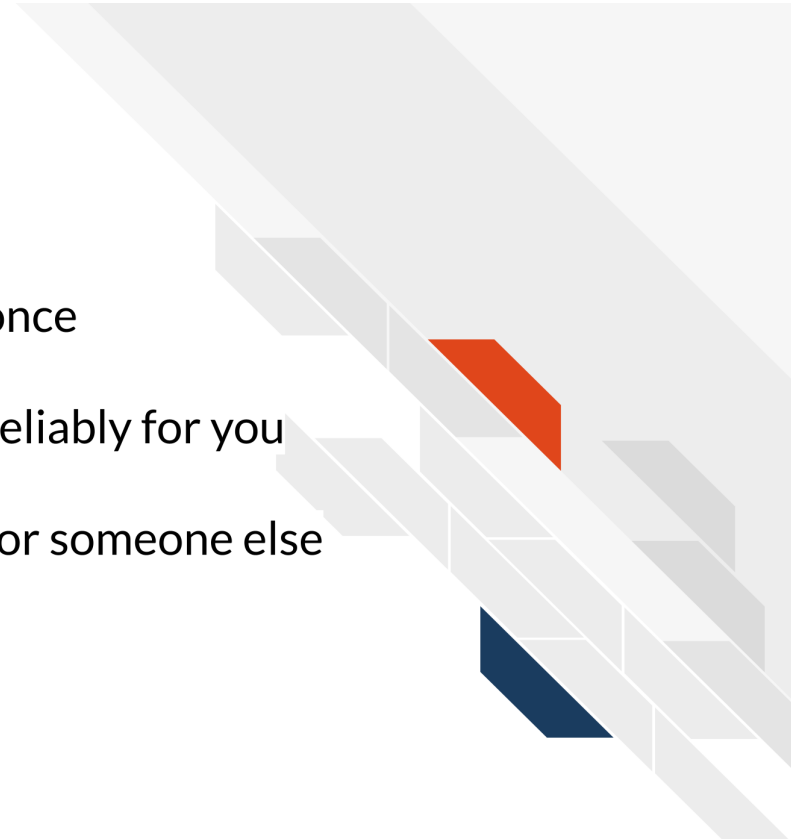
 Image created by Candace Savonen using Avataars.

# The process

Step 1) Get your code to work once

Step 2) Get your code to work reliably for you

Step 3) Get your code to work for someone else



# R Markdown

R Markdown notebooks are a handy tool for reproducibility!

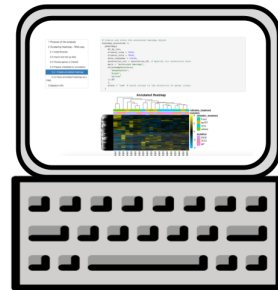


# R Markdown lets you test your work

Working from this notebook allows me to interactively develop on my data analysis and write down my thoughts about the process all in one place!



Ruby the Researcher



**RMarkdown is conducive to interactive development!**



Image created by Candace Savonen using Avataars.

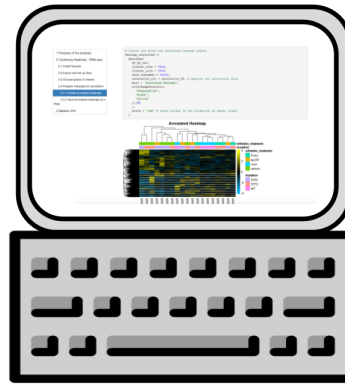
# R Markdown allows you to more clearly show what you did

Avi, here's some output from this scientific notebook I've been developing from!

This is so easy to follow and read, even though I didn't write the code. Thanks for sharing your exciting results!



Ruby the  
Researcher



Avi the  
Associate

**RMarkdown creates easily  
shareable output!**

Image created by Candace Savonen using Avataars.

# R Markdown makes it easier to update code and see results


Yay! I just got the data for 5 more samples. Because of my handy notebook set up, I can easily call one command and re-run the analysis so it is updated with the new samples included!



Ruby the  
Researcher



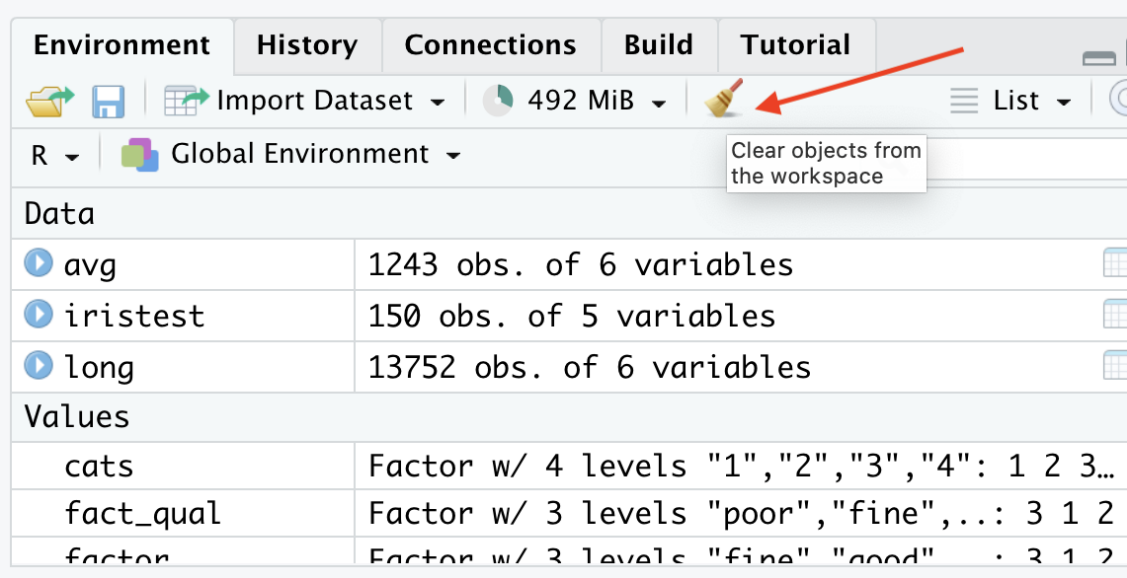
**RMarkdown is handy for  
creating updateable reports!**

 Image created by Candace Savonen using Avataars.

# Clean your environment

Regularly cleaning your environment and trying your code again, can help ensure that your code is running as expected.

Occasionally we might forget to save a step of our code in our R Markdown file that we ran only in the console. This will help us figure that out.



The screenshot shows the RStudio Environment pane. At the top, there are tabs for 'Environment', 'History', 'Connections', 'Build', and 'Tutorial'. Below the tabs, there are icons for file operations and a memory usage indicator showing '492 MiB'. A red arrow points to a trash can icon, which has a tooltip that reads 'Clear objects from the workspace'. Below this, the 'Global Environment' is selected. The pane is divided into 'Data' and 'Values' sections. The 'Data' section lists three objects: 'avg' (1243 obs. of 6 variables), 'iristest' (150 obs. of 5 variables), and 'long' (13752 obs. of 6 variables). The 'Values' section lists three factors: 'cats' (Factor w/ 4 levels "1","2","3","4": 1 2 3...), 'fact\_qual' (Factor w/ 3 levels "poor","fine",...: 3 1 2), and 'factor' (Factor w/ 3 levels "fine" "good" : 3 1 2).

Data	
avg	1243 obs. of 6 variables
iristest	150 obs. of 5 variables
long	13752 obs. of 6 variables

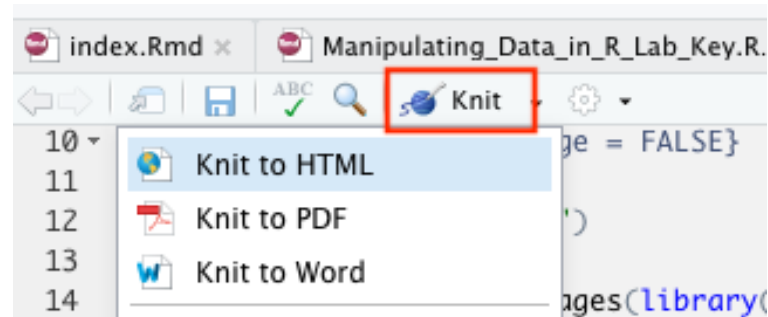
  

Values	
cats	Factor w/ 4 levels "1","2","3","4": 1 2 3...
fact_qual	Factor w/ 3 levels "poor","fine",...: 3 1 2
factor	Factor w/ 3 levels "fine" "good" : 3 1 2



## Check if your file knits regularly

Regularly checking if your file knits will help you spot a missing step or error earlier when you have less code to try to identify where your code might have gone wrong.



## **Tell your future self and others what you did!**

Provide sufficient detail so that you can understand what you did.

## Need random numbers to stay consistent?

Use `set.seed()` : sets the starting state for the random number generator (RNG) in R.

```
[1] 10 7 6 3 1 2 5 9 4 8
```

# R Markdown syntax

Before:

```
# Header - biggest font created by hashtag and space
## SubHeader Second Biggest created by 2 hashtags and space

bold text
italicized text

`code` referenced outside of a chunk needs backticks
```

After knit:

**Header - biggest font created by hashtag and space**

**SubHeader Second Biggest created by 2 hashtags and space**

**bold** text *italicized* text

`code` referenced outside of a chunk needs backticks

# R Markdown syntax

Go to Help > Cheat Sheets > R Markdown Cheat Sheet (which will download it)

Or checkout Help > Cheat Sheets > R Markdown Reference Guide

Or checkout the [Class Website!](#)

**SOURCE EDITOR**

The image shows a screenshot of an R Markdown source editor window titled "report.Rmd". The editor contains R Markdown code with several callouts pointing to specific features:

- 1. New File**: Points to the plus icon in the top-left corner of the editor.
- 2. Embed Code**: Points to the code chunk delimiters ````{r cars}` on line 20.
- 3. Write Text**: Points to the text content of the code chunk on lines 21-22.
- 4. Set Output Format(s) and Options**: Points to the `output:` section on lines 4-6, specifically the `toc: TRUE` option.
- 5. Save and Render**: Points to the "Run" button in the top-right toolbar.

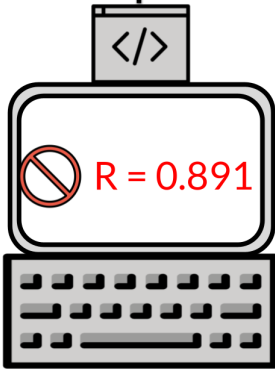
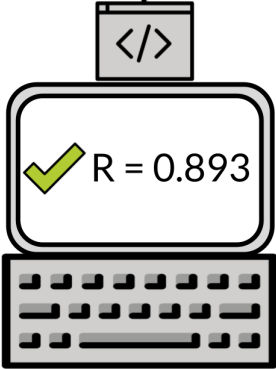
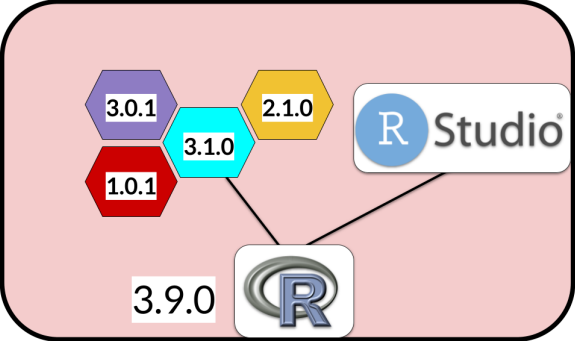
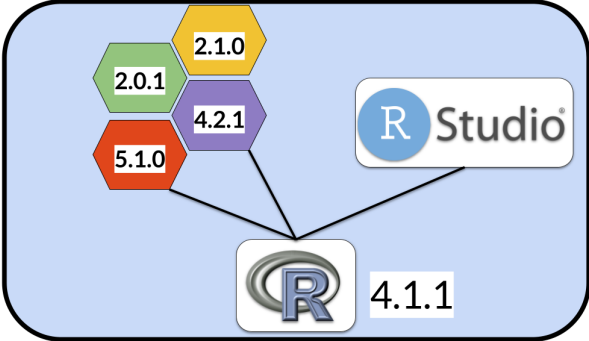
Additional callouts in the image include:

- "set preview location" pointing to the "set" icon in the toolbar.
- "insert code chunk" pointing to the "insert code chunk" icon in the toolbar.
- "go to code chunk" pointing to the "go to code chunk" icon in the toolbar.
- "run code chunk(s)" pointing to the "run code chunk(s)" icon in the toolbar.
- "modify chunk options" pointing to the gear icon in the toolbar.
- "run all previous chunks" pointing to the "run all previous chunks" icon in the toolbar.

```
1 ---
2 title: "Document Title"
3 author: "Author Name"
4 output:
5   html_document:
6     toc: TRUE
7 ---
8
9 ```{r setup, include=FALSE}
10 knitr::opts_chunk$set(echo = TRUE)
11 ```
12
13 ## R Markdown
14
15 This is an R Markdown document.
16 Markdown is a simple formatting
17 syntax for authoring HTML, PDF,
18 and MS Word documents.
19
20 ```{r cars}
21 summary(cars)
22 ```
```

# Versions matter

Ruby's local computing environment    Avi's local computing environment



Created by Candace Savonen

# Session info can help

## Ruby's session info print out

```
R version 4.0.2 (2020-06-22)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 20.04.2 LTS

Matrix products: default
BLAS/LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblas-p0.3.8.so

locale:
 [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C              LC_TIME=en_US.UTF-8
 [4] LC_COLLATE=en_US.UTF-8   LC_MONETARY=C             LC_MESSAGES=C
 [7] LC_PAPER=en_US.UTF-8     LC_NAME=C                 LC_MEASUREMENT=C
[10] LC_TELEPHONE=C

attached base packages:
[1] stats      graphics  grDevices  utils      datasets  methods   base

other attached packages:
[1] rmarkdown_2.4

loaded via a namespace (and not attached):
 [1] rstudioapi_0.11  knitr_1.30      magrittr_1.5    hms_0.5.3      tidyselect_1.
 [6] R6_2.4.1         rlang_0.4.7     fansi_0.4.1     dplyr_1.0.2    tools_4.0.2
[11] xfun_0.18        sessioninfo_1.1.1 tinytex_0.26    cli_2.0.2      withr_2.3.0
[16] htmltools_0.5.0 ellipsis_0.3.1  assertthat_0.2.1 yaml_2.2.1     digest_0.6.25
[21] tibble_3.0.3     lifecycle_0.2.0 crayon_1.3.4    evaluate_0.14
[26] vctrs_0.3.4      glue_1.4.2      ncaranfin_2.0.3
```

## Avi's session info print out

```
R version 4.0.5 (2021-03-31)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Big Sur 10.16

Matrix products: default
LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib

locale:
 [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats      graphics  grDevices  utils      datasets  methods   base

other attached packages:
[1] rmarkdown_2.10

loaded via a namespace (and not attached):
 [1] leanbuild_0.1.2  BiocManager_1.30.16 compiler_4.0.5  magrittr_2.0.1
 [5] fastmap_1.1.0   htmltools_0.5.2  tools_4.0.5     yaml_2.2.1
 [9] tinytex_0.33    knitr_1.33       digest_0.6.27   xfun_0.25
[13] rlang_0.4.11    evaluate_0.14
```

R version 4.0.2 vs 4.0.5

Different operating systems!

rmarkdown 2.4 vs 2.10

If Avi and Ruby have discrepancies in their results, the session info print out gives a record which may have clues to why that might be!



# Lab 1

- [Class Website](#)
- [Lab](#)



## More resources

These are just some quick tips, for more information:

- [Reproducibility in Cancer Informatics course](#)
- [Advanced Reproducibility in Cancer Informatics course](#)
- [The RMarkdown book](#)
- [Jenny Bryan's organizational strategies.](#)
- [Write efficient R code for science.](#)

# Summary

To help make your work more reproducible:

- Use RMarkdown
- Clean your environment regularly
- Check the knit of your RMarkdown regularly
- Tell your future self and others what you did!
- Print session info!

▢ [Class Website](#)



Image by [Gerd Altmann](#) from [Pixabay](#)