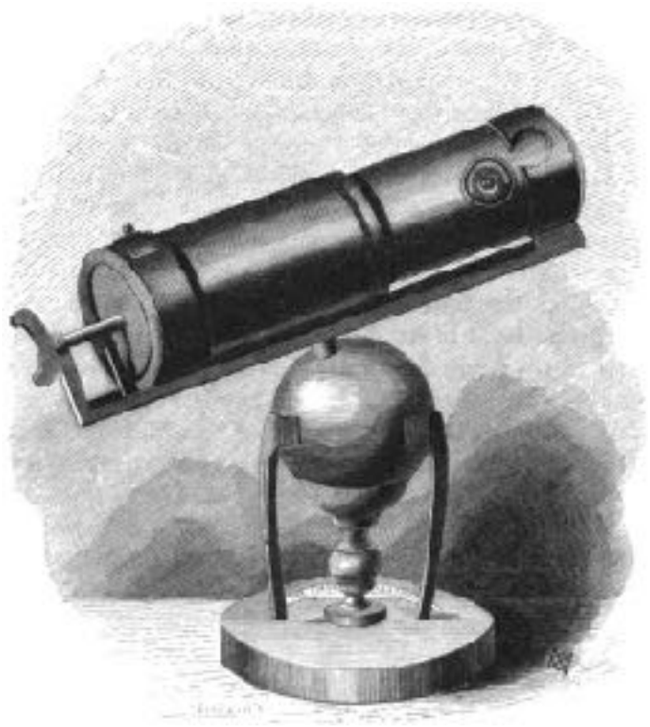
TidyBiology:

An Introduction to Biological Data Science in R



Matthew Hirschey, Ph.D.

Duke University Medical Center

Department of Medicine, Pharmacology & Cancer Biology

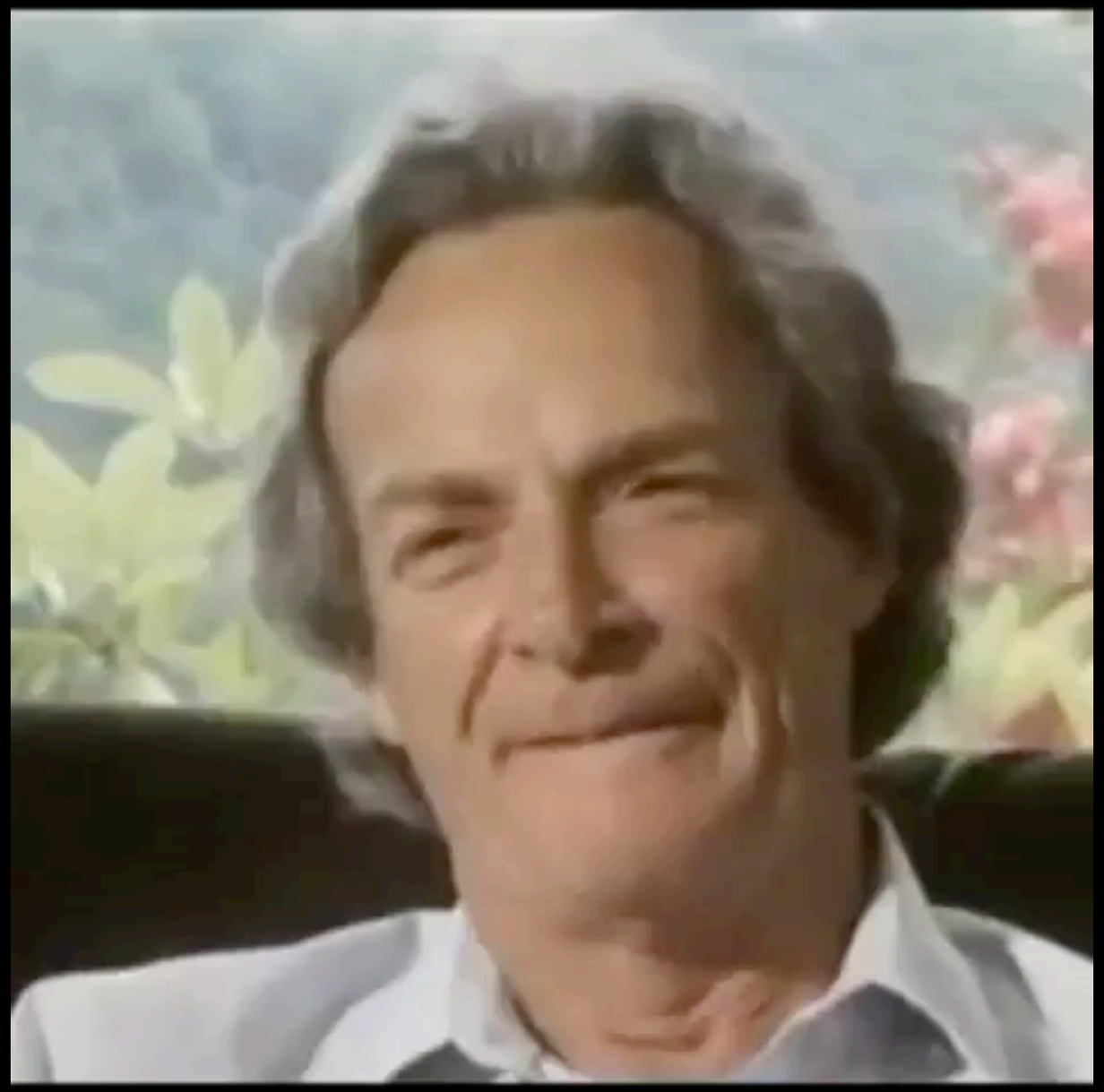
Duke Molecular Physiology Institute

Durham, NC, USA

RESULTS AND DISCUSSION

Identification of YFG as a Potential YFG-ase

Because [this metabolite is unique], we hypothesized that [enzymes] with [special] domains may have YFG-ase activity. To identify potential candidates, we first searched the Swiss-Prot database using the terms "YFG-ase" and "[metabolite]" as keywords and obtained 31 candidate proteins (Figure S1). We next excluded enzymes expressed in yeast, because yeast do not express [enzymes like YFG]. After these screening steps, we obtained nine candidates, including all YFG family members (Figure S1). Among the YFG family members, YFG1 is the most expressed isoform in the liver, where [we think it is important]…So we tested whether YFG1 is a YFG-ase.



“See that bird? It’s a brown-throated thrush, …and even if you know the name for it, you still know nothing about the bird. You only know something about people; what they call the bird.”

-Richard Feynman

Goal of science is to know something

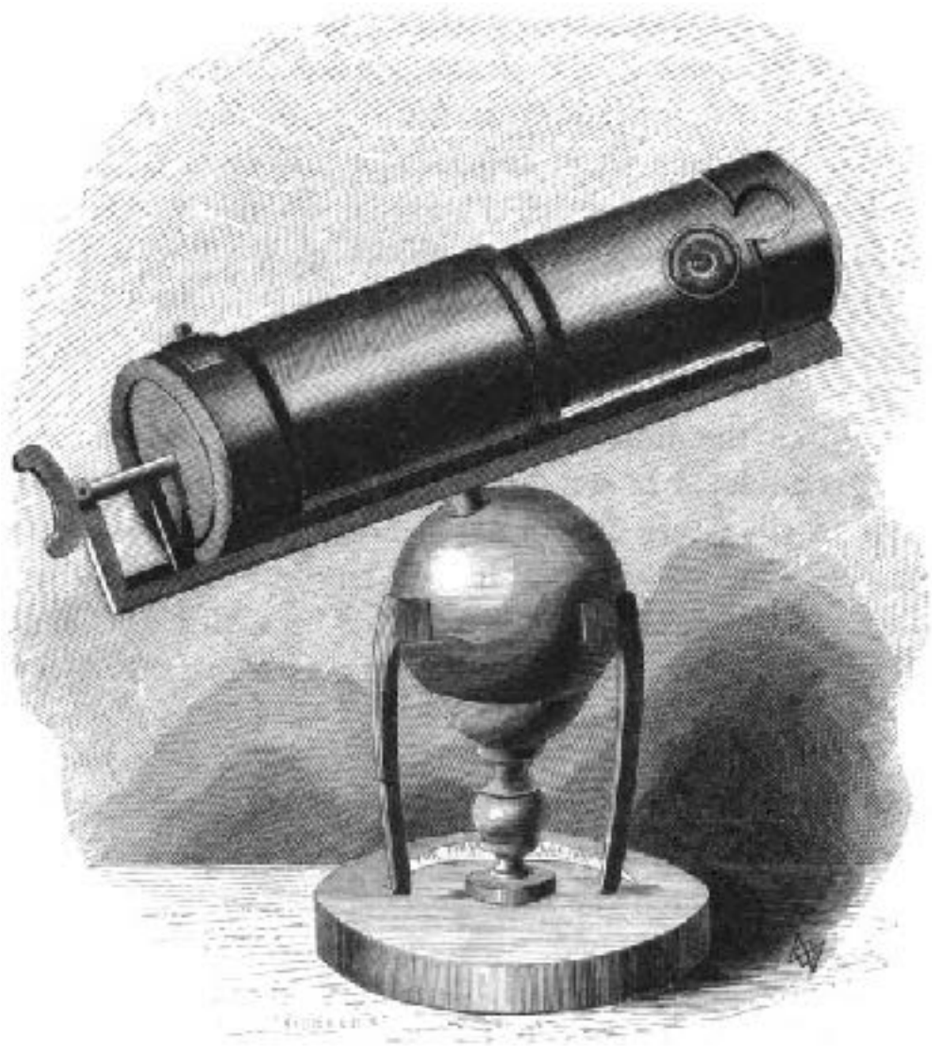
I don′t know what I may seem to the world, but as to myself, I seem to have been only like a boy playing on the sea-shore and diverting myself in now and then finding a smoother pebble or a prettier shell than ordinary, whilst the great ocean of truth lay all undiscovered before me.”

— Isaac Newton (1642-1727)

https://image01.cudl.lib.cam.ac.uk/content/images/MS-ADD-03975-000-00001.jpg

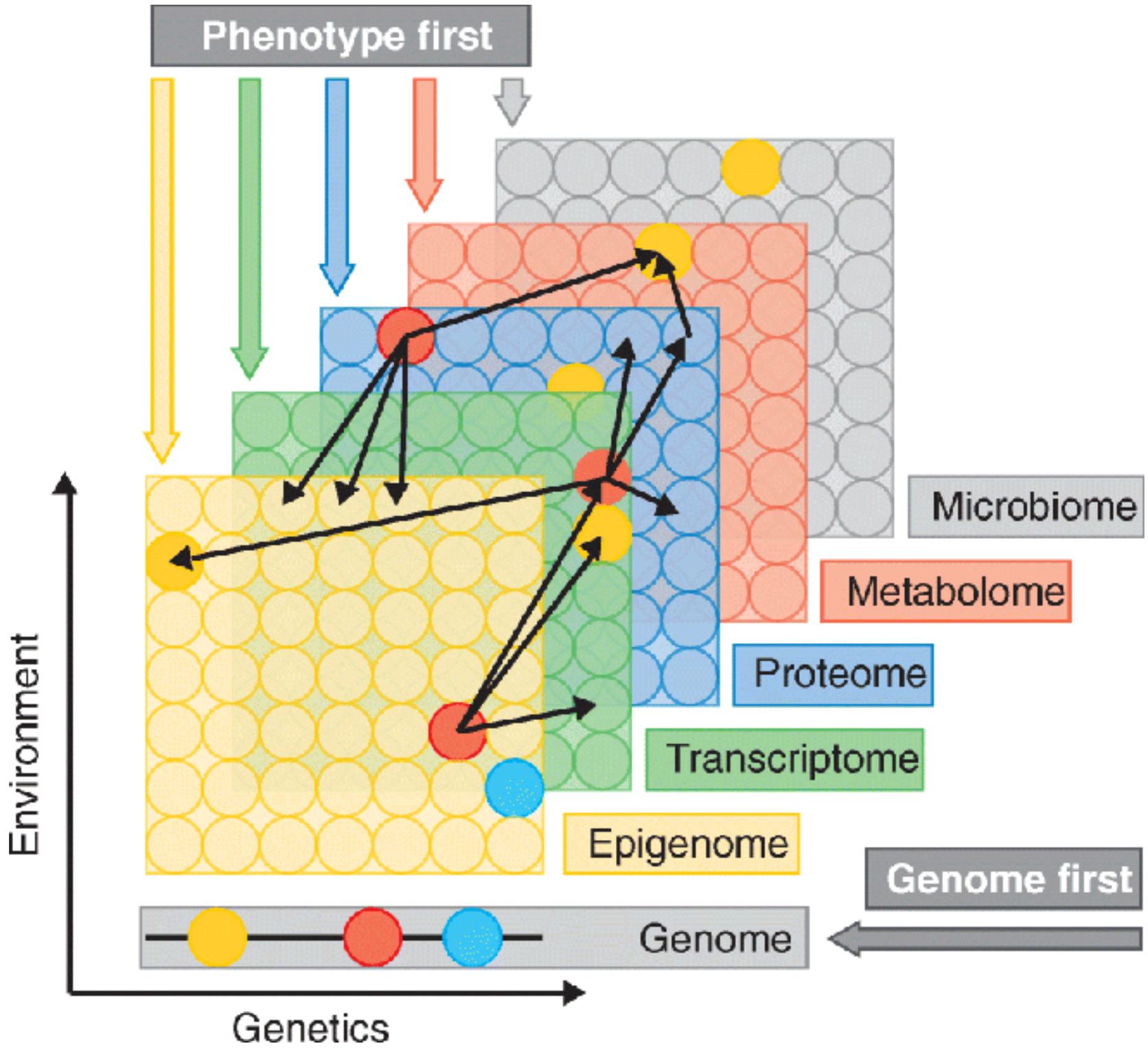
Scientific Method

1. Define a question
2. Observe (gather information and think)
3. Form a hypothesis
4. Test the hypothesis (performing an experiment and collecting data in a reproducible manner)
5. Analyze data (null-hypothesis significance test)
6. Interpret the data and draw conclusions (starting point for new hypotheses)



Is there enough evidence to reject the null hypothesis?

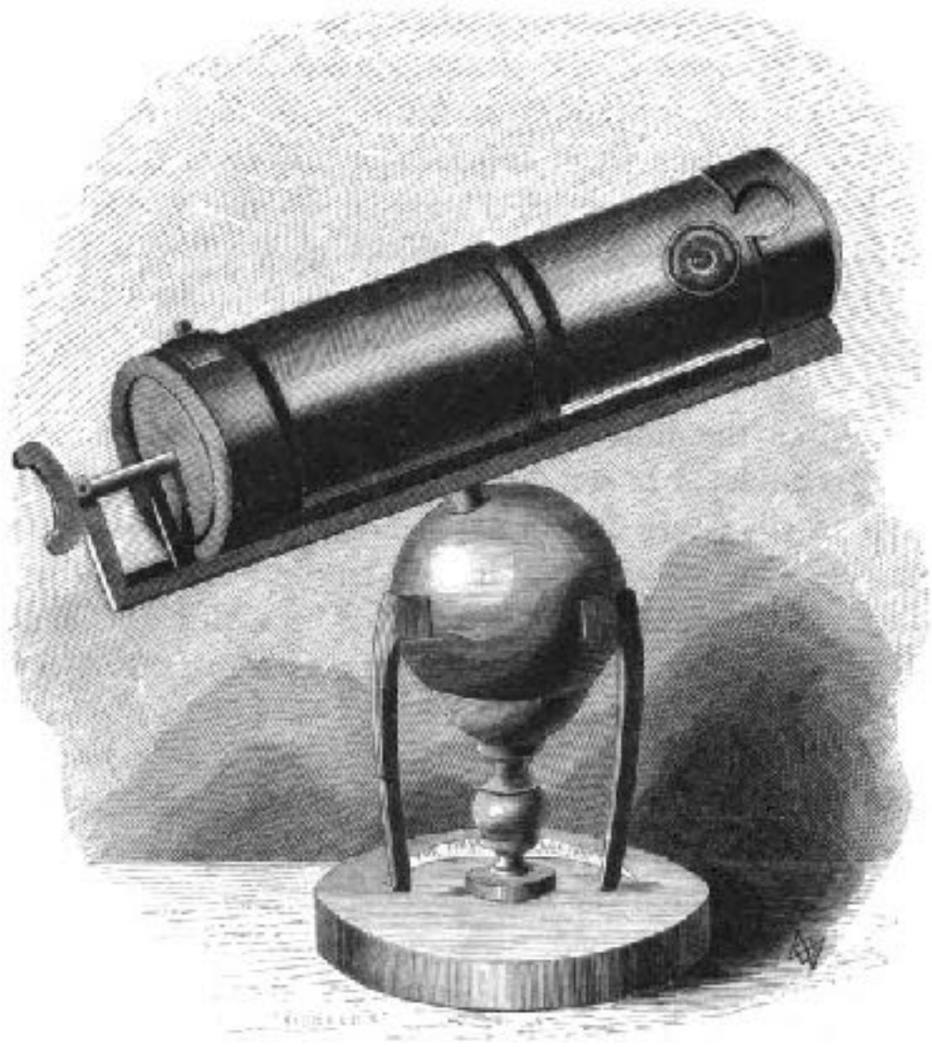
Screen



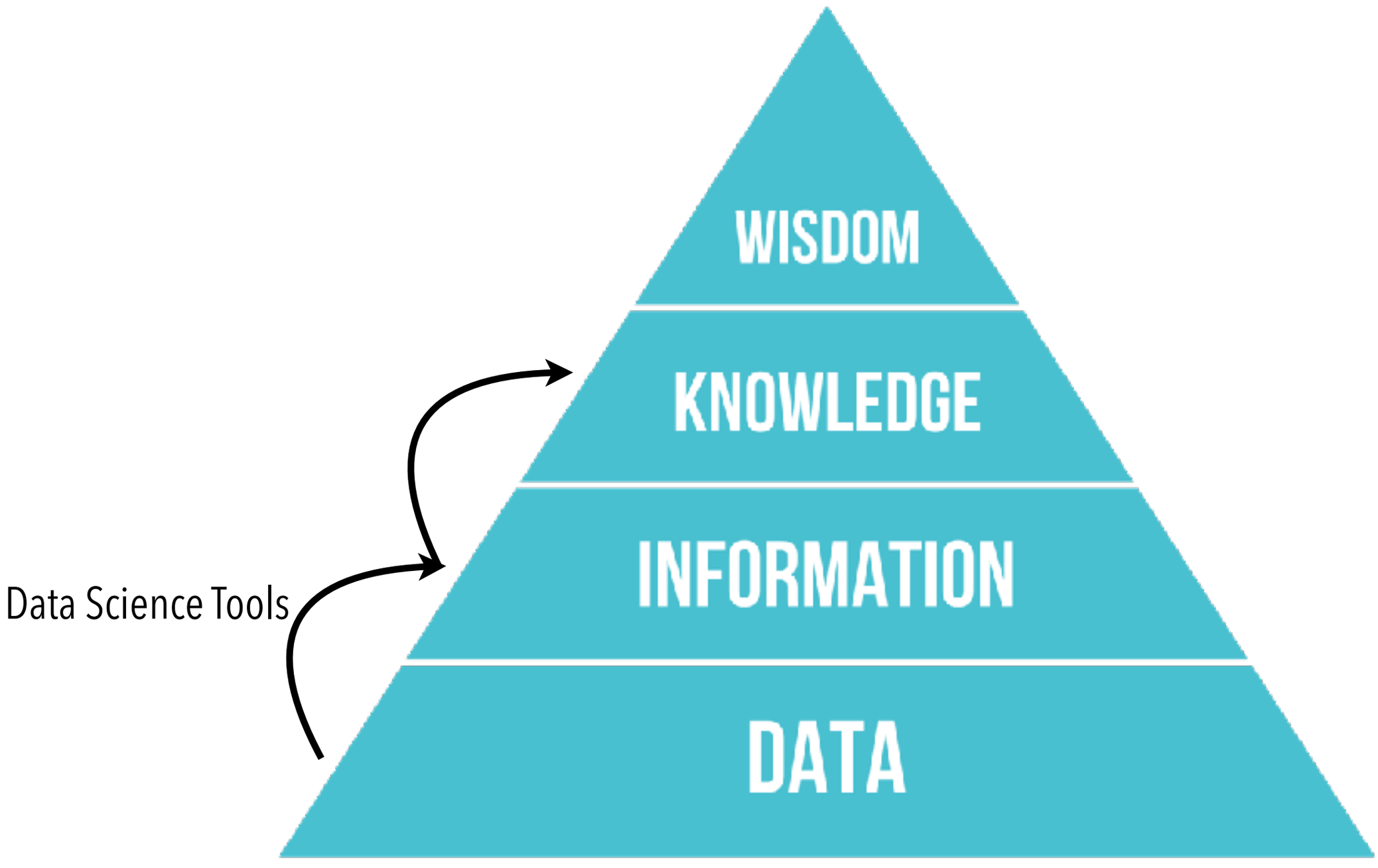
Yehudit Hasin, Marcus Seldin and Aldons Lusis. Multi-omics approaches to disease. Genome Biology (2017) 18:83

Scientific Method

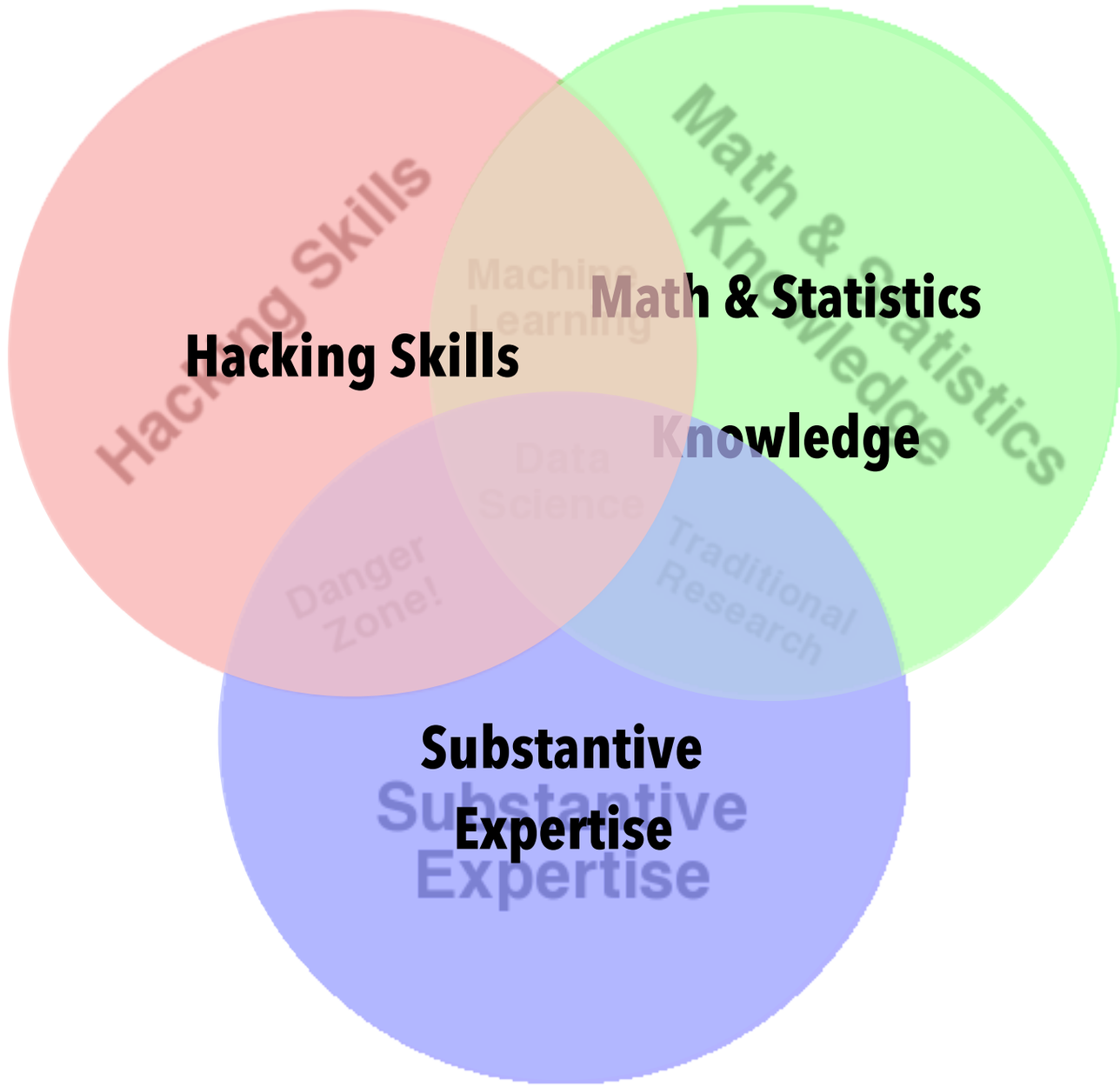
1. Define a question
2. Observe (gather information and think, **screen**)
3. Form a hypothesis
4. Test the hypothesis (performing an experiment and collecting data in a reproducible manner)
5. Analyze data (null-hypothesis significance test)
6. Interpret the data and draw conclusions (starting point for new hypotheses)



Hierarchy of Knowledge Gathering

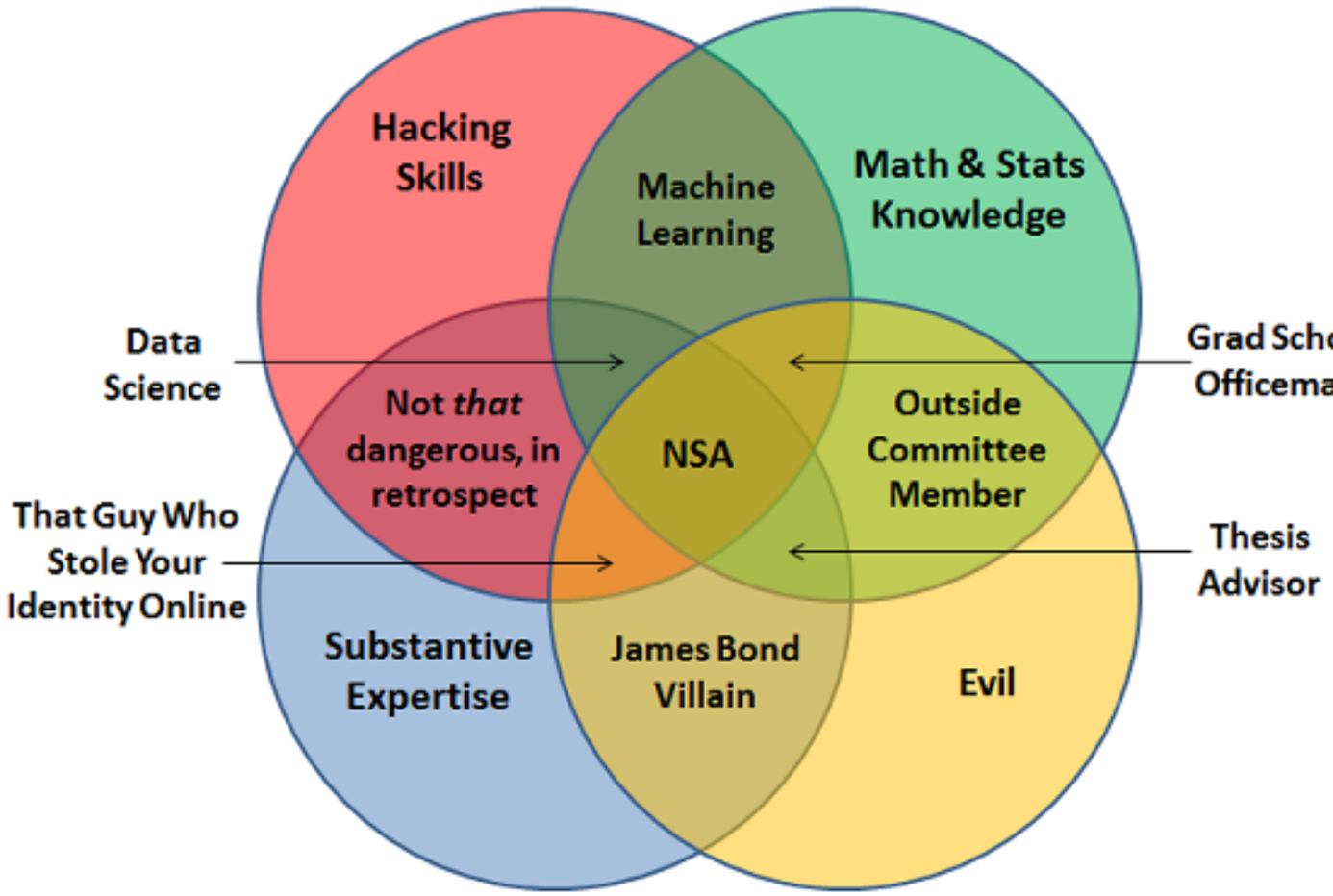


Emerging field of Data Science



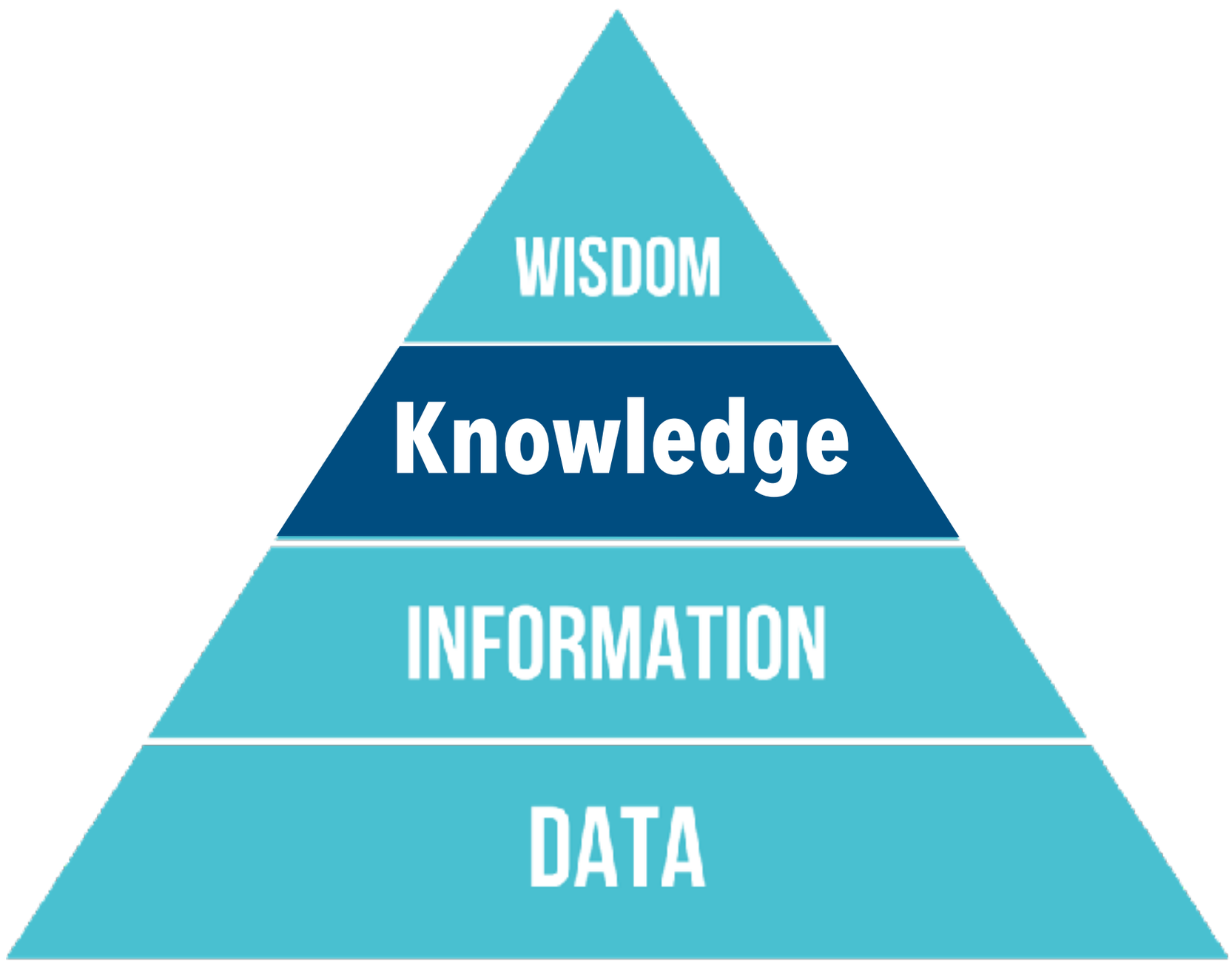
http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

Venn Diagram of Data Science v2.0



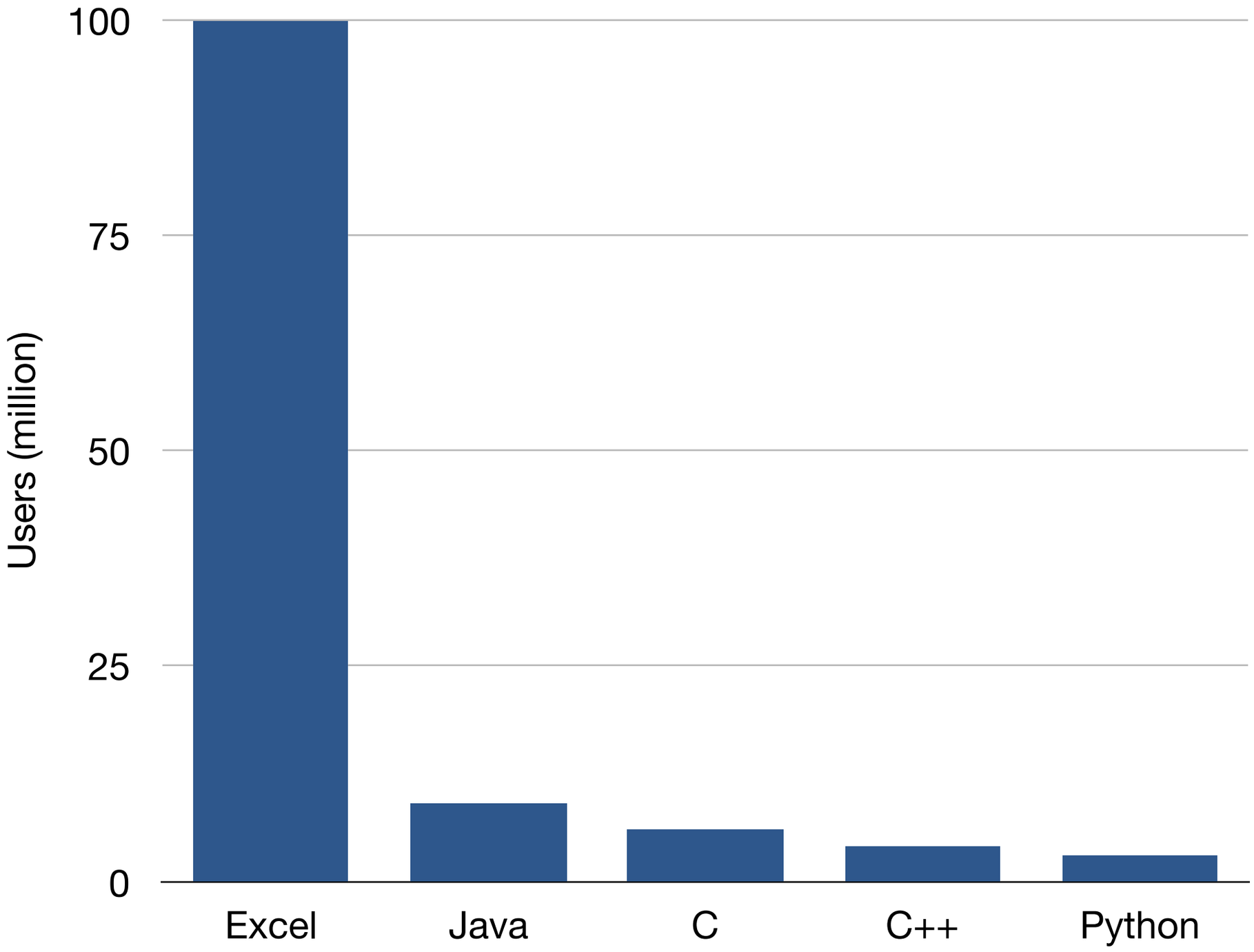
https://www.kdnuggets.com/2016/10/battle-data-science-venn-diagrams.html

Overall goal is to generate knowledge



Several approaches in data science

World's most popular programming languages



R

Language

Values: 1, "Florida", "2010-01-25"

R the language

Values: 1, "Florida", "2010-01-25"

Objects: x <- 22/7

A name without quotes

< followed by -

(it looks like an arrow)

A value, object, or function result

R the language

Values - 1, "Florida", "2010-01-25"

Objects - x <- c(22/7, 0.99, 3)

To put multiple values in an object, combine the values with c()

R the language

Values - 1, "Florida", "2010-01-25"

Objects - x <- c(22/7, 0.99, 3)

Functions - round(x, digits = 3)

A name without quotes

followed by () to run the function

Arguments: values, objects, or function results

Which of these are numbers?

1 "1" "one" one

Which of these are numbers?

$\frac{1}{\text{number}}$	"1"	"one"	one

Which of these are numbers?

1	"1"	"one"	one
number	words (strings)		

Which of these are numbers?

1	"1"	"one"	one
number	words (strings)		object

Which of these will work?

Suppose one <- 1.

log(1) log("1") log("one") log(one)

Data are stored in a dataframe

Data stored in a dataframe are conceptually equivalent to a spreadsheet with rows and columns (chromosome$id)

Filter

	id	length_mm	basepairs	variations	protein_codinggenes	pseudo_genes	totallongnc_rna	totals
1	1	85	248956422	12151146	2058	1220	1200	496
2	2	83	242193529	12945965	1309	1023	1037	375
3	3	67	198295559	10638715	1078	763	711	298
4	4	65	190214555	10165685	752	727	657	228
5	5	62	181538259	9519995	876	721	844	235
6	6	58	170805979	9130476	1048	801	639	234
7	7	54	159345973	8613298	989	885	605	208
8	8	50	145138636	8221520	677	613	735	214
9	9	48	138394717	6590811	786	661	491	190
10	10	46	133797422	7223944	733	568	579	204
11	11	46	135086622	7535370	1298	821	710	233
12	12	45	133275309	7228129	1034	617	848	227
13	13	39	114364328	5082574	327	372	397	104
14	14	36	107043718	4865950	830	523	533	239
15	15	35	101991189	4515076	613	510	639	250
16	16	31	90338345	5101702	873	465	799	187

Extract or create new objects

You can call a single part of the data frame

```
chromosome$id
```

id
1
2
3
4
5
6
7
8
9

And save it as an object for later use

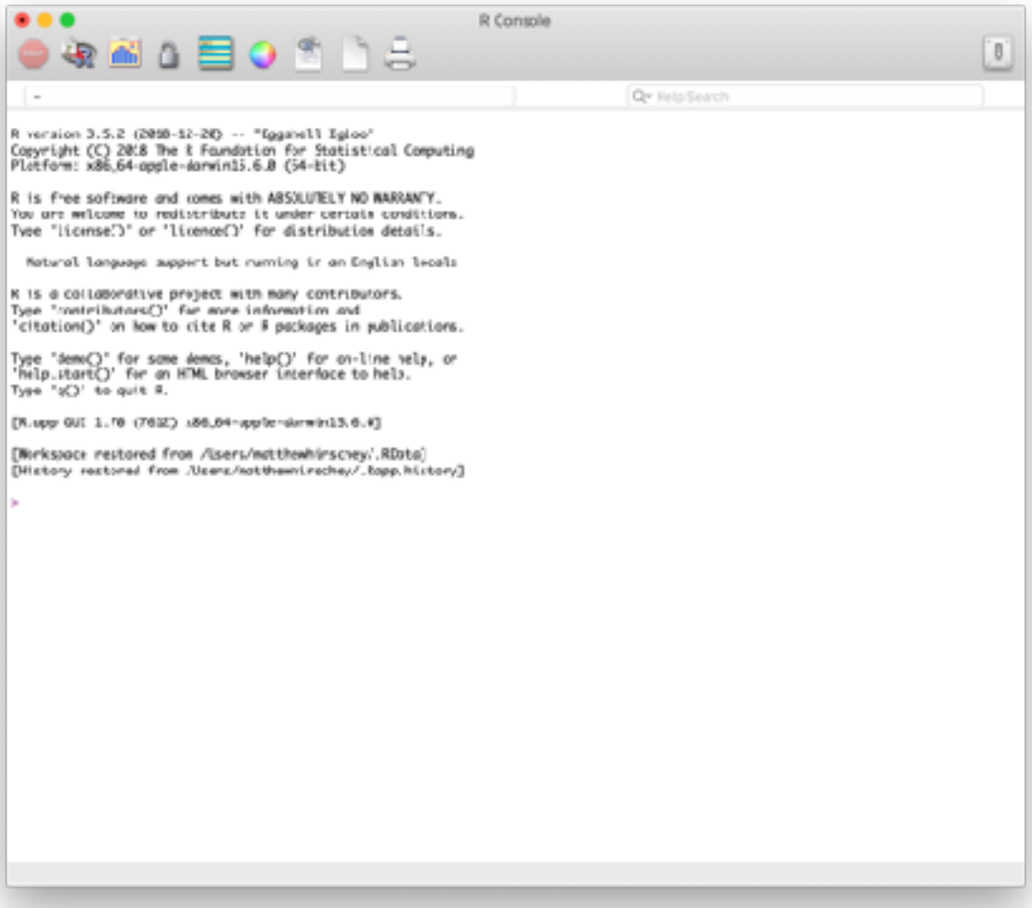
```
ids <- chromosome$id
```

R

IDE

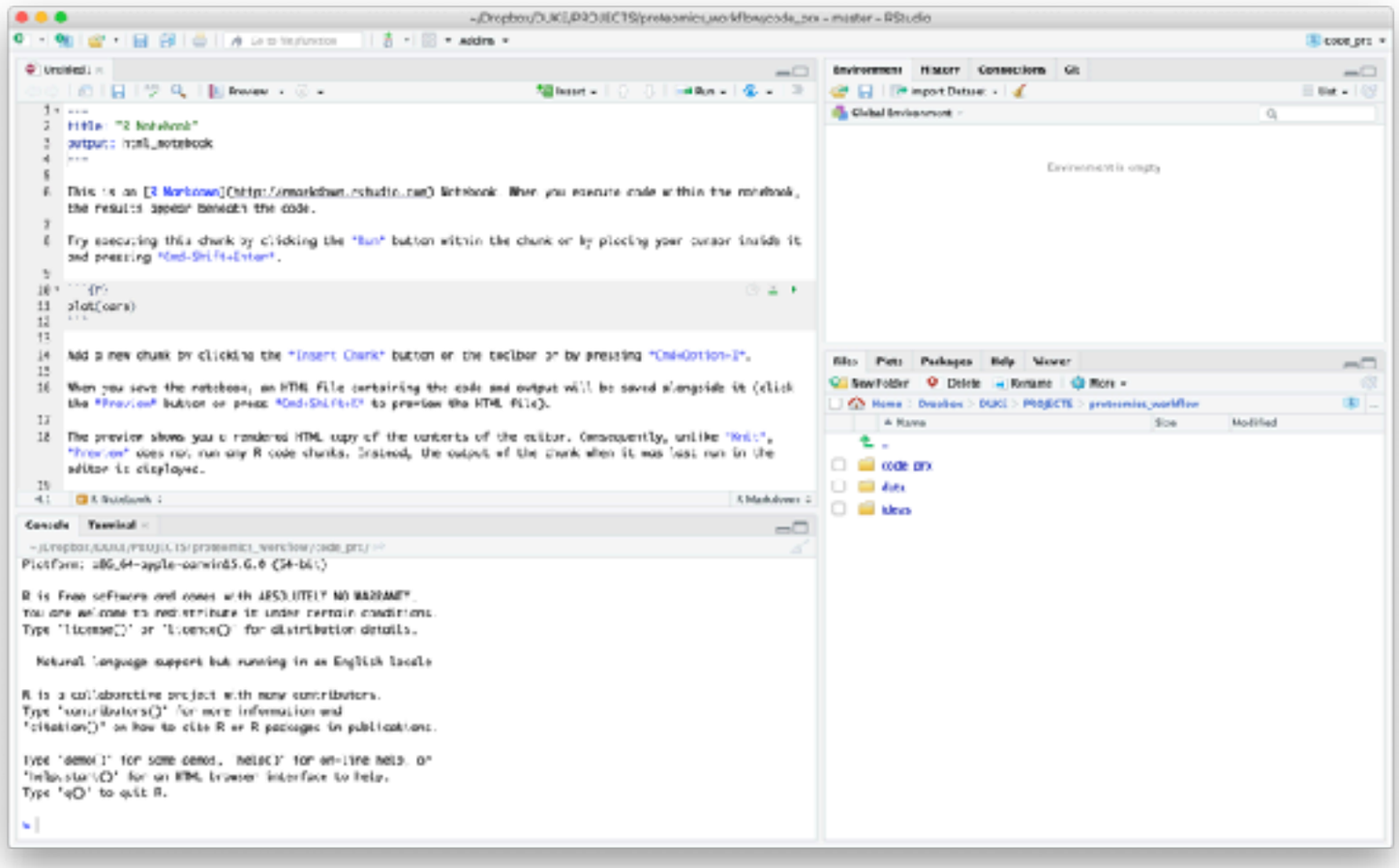
R

Statistical Programming Language



RStudio

Integrated Developer Environment



Source: Modern Dive; h/t datasciencebox.org

Demo

What am I looking at?

R v. Rstudio

Go to 01_intro

Open 01_demo.Rmd

Follow along if you'd like.

R Markdown

R Markdown Introduction

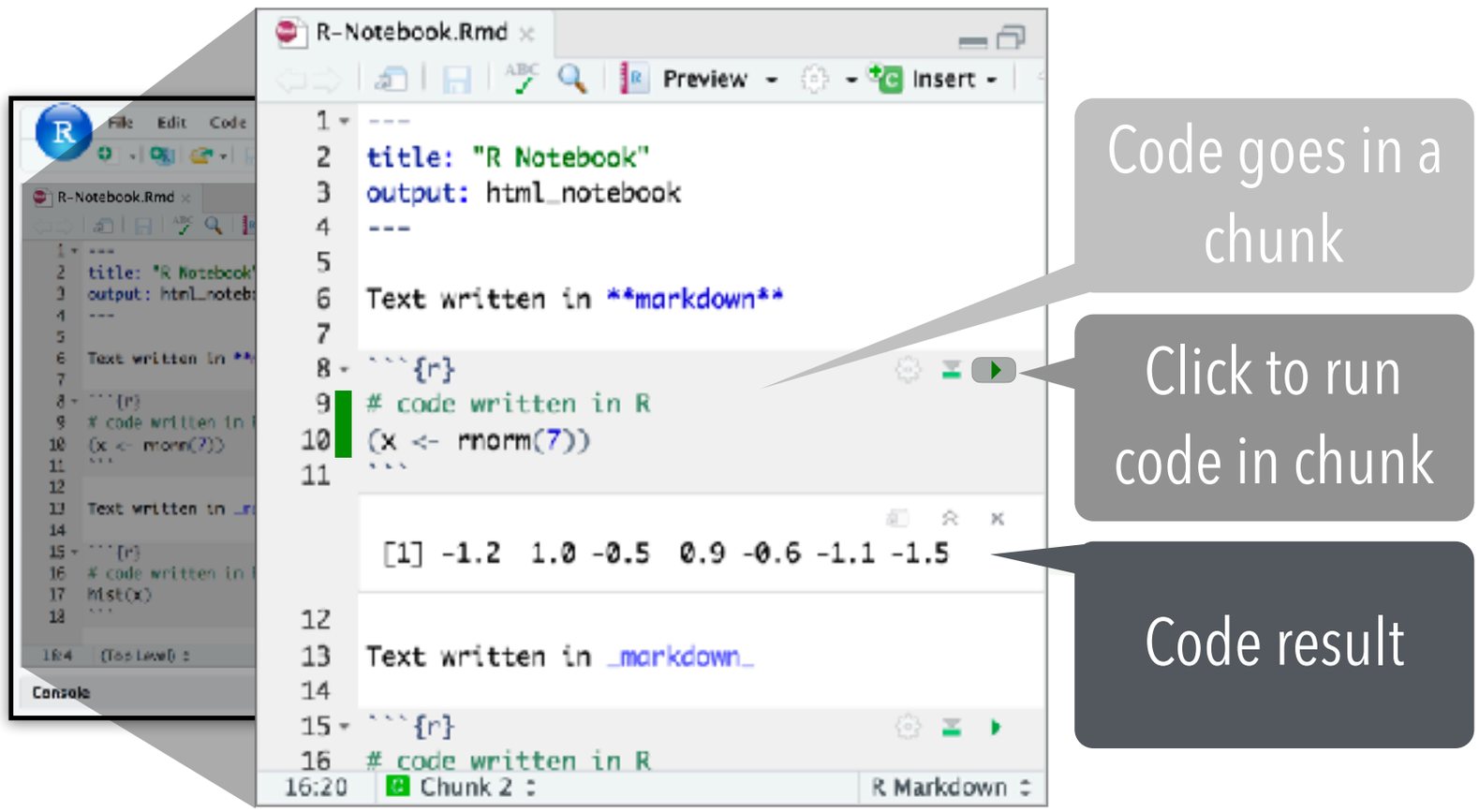
Go to 01_intro

Open 01_exercise.Rmd

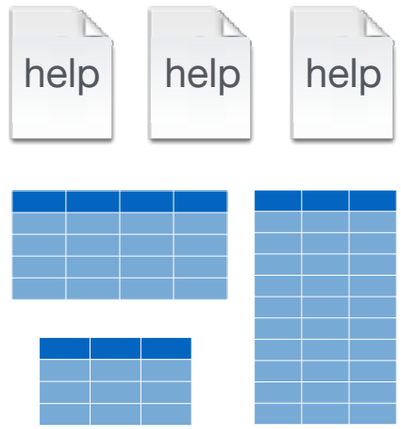
Read through the file and do everything it tells you to do.

R Markdown

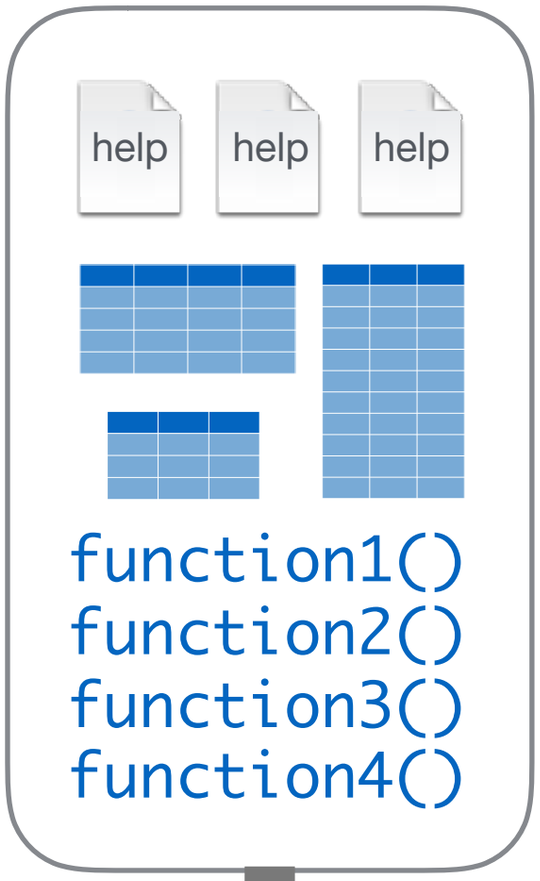
An authoring format for Data Science.

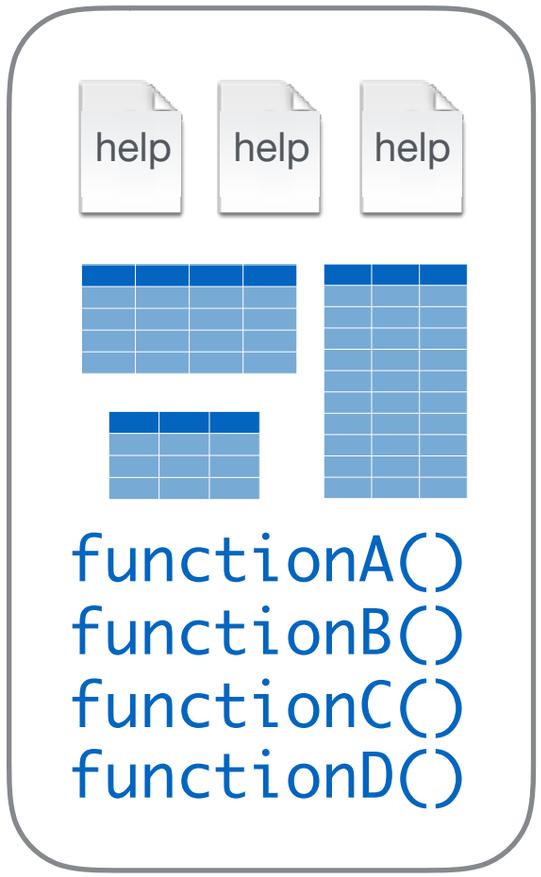


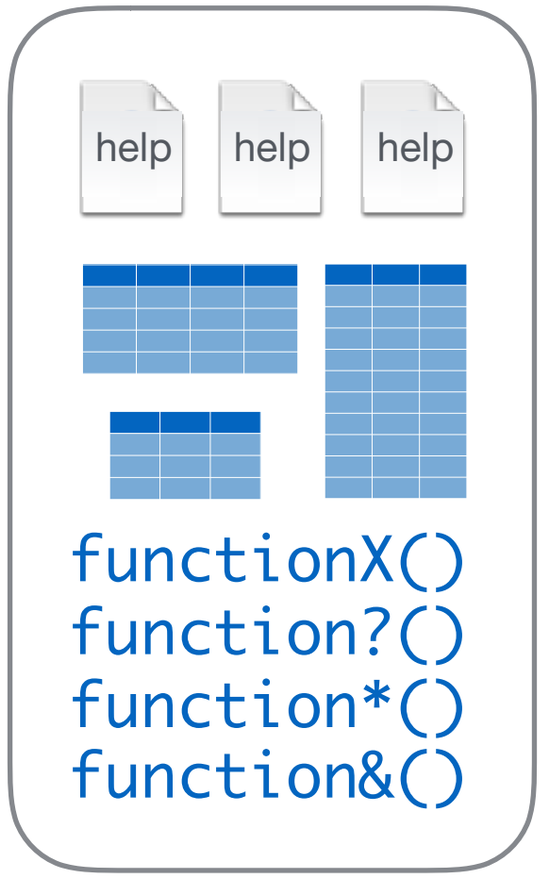
R Packages



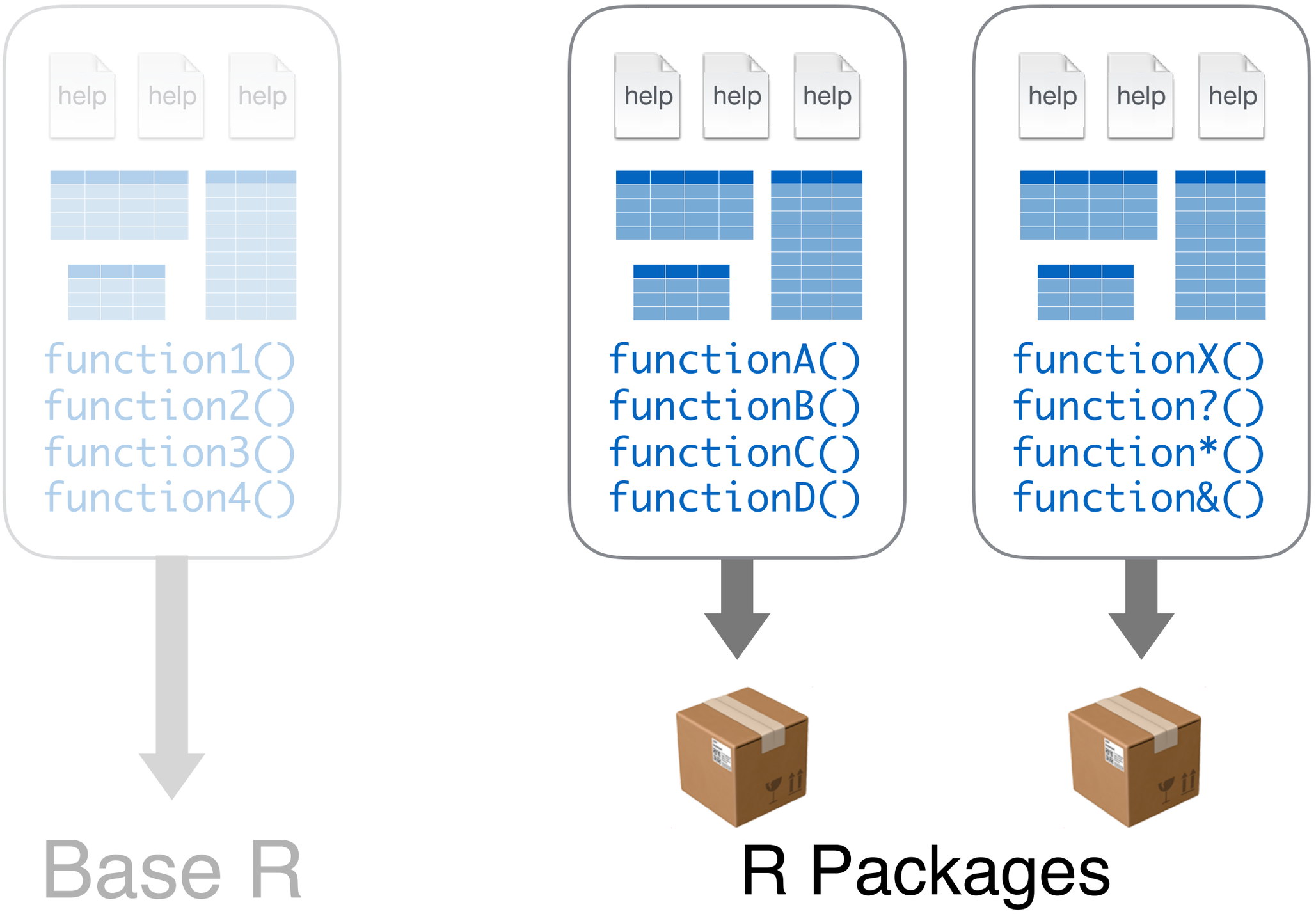

```
function1()
function2()
function3()
function4()
```


help
help
help
function1()
function2()
function3()
function4()


help
help
help
functionA()
functionB()
functionC()
functionD()


help
help
help
functionX()
function?()
function*()
function&()

Base R


help
help
help
function1()
function2()
function3()
function4()
help
help
help
functionA()
functionB()
functionC()
functionD()
help
help
help
functionX()
function?()
function*()
function&()
Base R
R Packages

“Verbs” (i.e. functions) act on data

These “verbs” act on data

```
do_this(to_that)

do_this(to_that, using_these)
```

Packages contain functions, documentation, data

Package

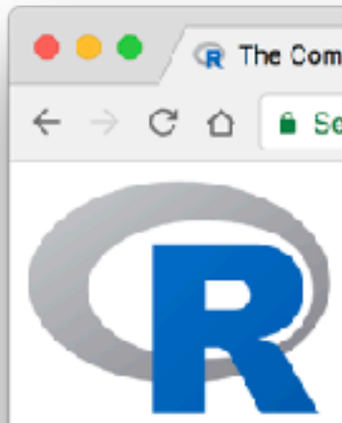
dplyr part of the tidyverse

0.8.3

Overview

dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges:

- `mutate()` adds new variables that are functions of existing variables
- `select()` picks variables based on their names.
- `filter()` picks cases based on their values.
- `summarise()` reduces multiple values down to a single summary.
- `arrange()` changes the ordering of the rows.

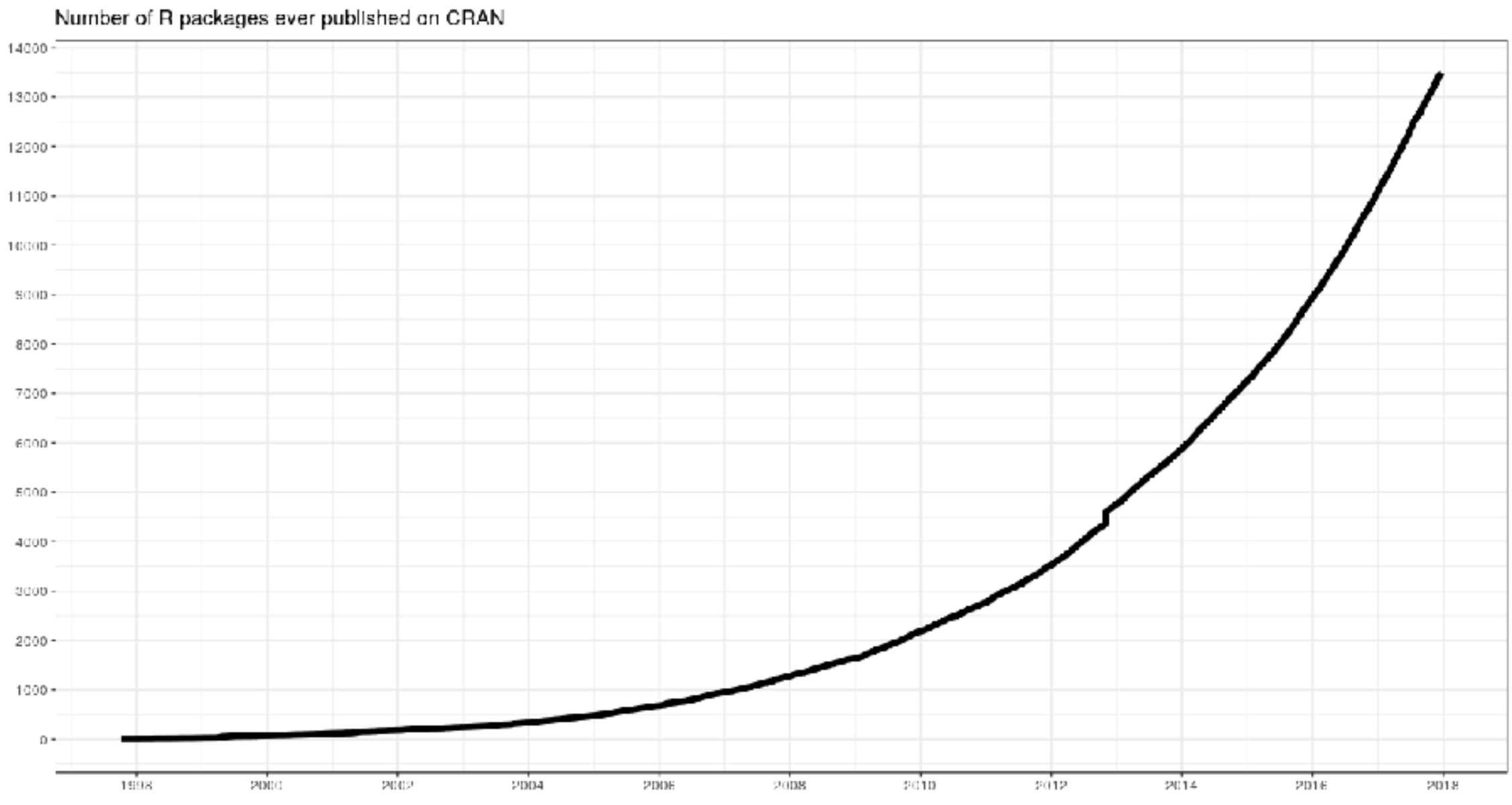


Available CRAN Packages By Name

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A3	Accurate, Adaptable, and Accessible Error Metrics for Predictive Models
abbyyR	Access to Abbyy Optical Character Recognition (OCR) API
abc	Tools for Approximate Bayesian Computation (ABC)
ABCanalysis	Computed ABC Analysis
abc.data	Data Only: Tools for Approximate Bayesian Computation (ABC)
abcdeFBA	ABCDE_FBA: A-Biologist-Can-Do-Everything of Flux Balance Analysis with this package
ABCoptim	Implementation of Artificial Bee Colony (ABC) Optimization
ABCp2	Approximate Bayesian Computational Model for Estimating P2
ABC.RAP	Array Based CpG Region Analysis Pipeline
abcrf	Approximate Bayesian Computation via Random Forests
abctools	Tools for ABC Analyses
abd	The Analysis of Biological Data
abf2	Load Gap-Free Axon ABF2 Files
ABHgenotypeR	Easy Visualization of ABH Genotypes
abind	Combine Multidimensional Arrays
abjutils	Useful Tools for Jurimetrical Analysis Used by the Brazilian Jurimetrics Association
abn	Modelling Multivariate Data with Additive Bayesian Networks
abodOutlier	Angle-Based Outlier Detection
AbsFilterGSEA	Improved False Positive Control of Gene-Permuting GSEA with Absolute Filtering
AbSim	Time Resolved Simulations of Antibody Repertoires
abundant	High-Dimensional Principal Fitted Components and Abundant Regression
ACA	Abrupt Change-Point or Aberration Detection in Point Series
acc	Exploring Accelerometer Data
accelerometry	Functions for Processing Minute-to-Minute Accelerometer Data
accelmissing	Missing Value Imputation for Accelerometer Data
AcceptanceSampling	Creation and Evaluation of Acceptance Sampling Plans

14108
packages
as of 5/2019



Graph by https://besteamperu.org

Using packages

Step1

```
install.packages(“tidyverse")
```

Downloads files to computer

1 x per computer

Tidyverse

Packages Articles Learn Help Contribute

R packages for data science

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Pop Quiz!

The tidyverse contains the following packages. How would you install them?

ggplot2

dplyr

tidyr

readr

purrr

tibble

hms

stringr

lubridate

forcats

DBI

haven

httr

jsonlite

readxl

rvest

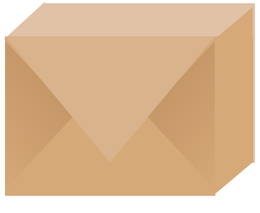
xml2

modelr

tidyverse

```
install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyr")
install.packages("readr")
install.packages("purrr")
install.packages("tibble")
install.packages("hms")
install.packages("stringr")
install.packages("lubridate")
install.packages("forcats")
install.packages("DBI")
install.packages("haven")
install.packages("httr")
install.packages("jsonlite")
install.packages("readxl")
install.packages("rvest")
install.packages("xml2")
install.packages("modelr")
install.packages("broom")
```

tidyverse



An R package that serves as a short cut for installing and loading the components of the tidyverse.

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

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

does the equivalent of

```
install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyr")
install.packages("readr")
install.packages("purrr")
install.packages("tibble")
install.packages("hms")
install.packages("stringr")
install.packages("lubridate")
install.packages("forcats")
install.packages("DBI")
install.packages("haven")
install.packages("httr")
install.packages("jsonlite")
install.packages("readxl")
install.packages("rvest")
install.packages("xml2")
install.packages("modelr")
install.packages("broom")
```

Using packages

Step1

```
install.packages("tidybiology")
```

Downloads files to computer

1 x per computer

Step2

```
library("tidybiology")
```

Loads package

1 x per R Session

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

does the equivalent of

```
install.packages("ggplot2")
install.packages("dplyr")
install.packages("tidyr")
install.packages("readr")
install.packages("purrr")
install.packages("tibble")
install.packages("stringr")
install.packages("forcats")
install.packages("lubridate")
install.packages("hms")
install.packages("DBI")
install.packages("haven")
install.packages("httr")
install.packages("jsonlite")
install.packages("readxl")
install.packages("rvest")
install.packages("xml2")
install.packages("modelr")
install.packages("broom")
```

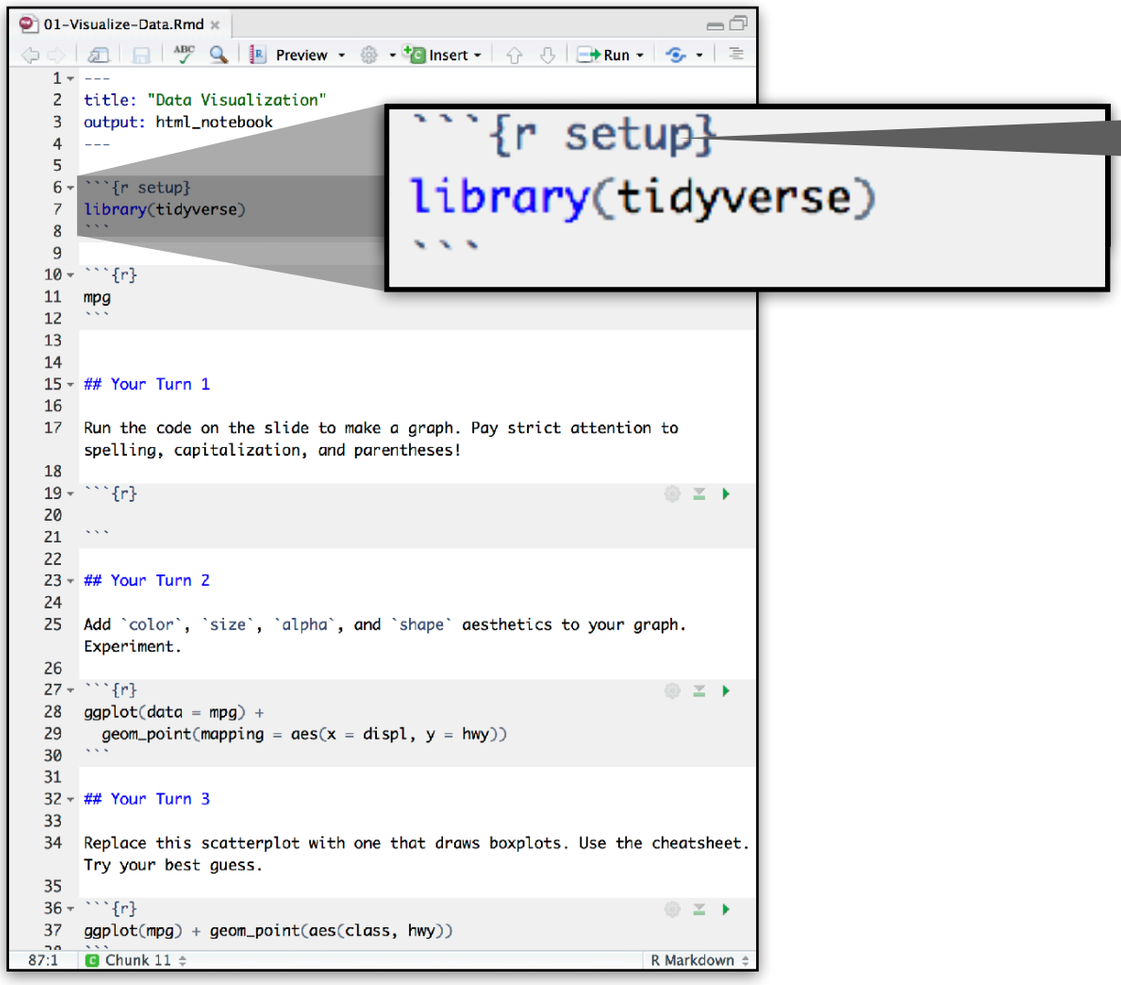
```
library("tidyverse")
```

does the equivalent of

```
library("ggplot2")
library("dplyr")
library("tidyr")
library("readr")
library("purrr")
library("tibble")
library("stringr")
library("forcats")
```

Setup

The setup chunk is always run once before anything else



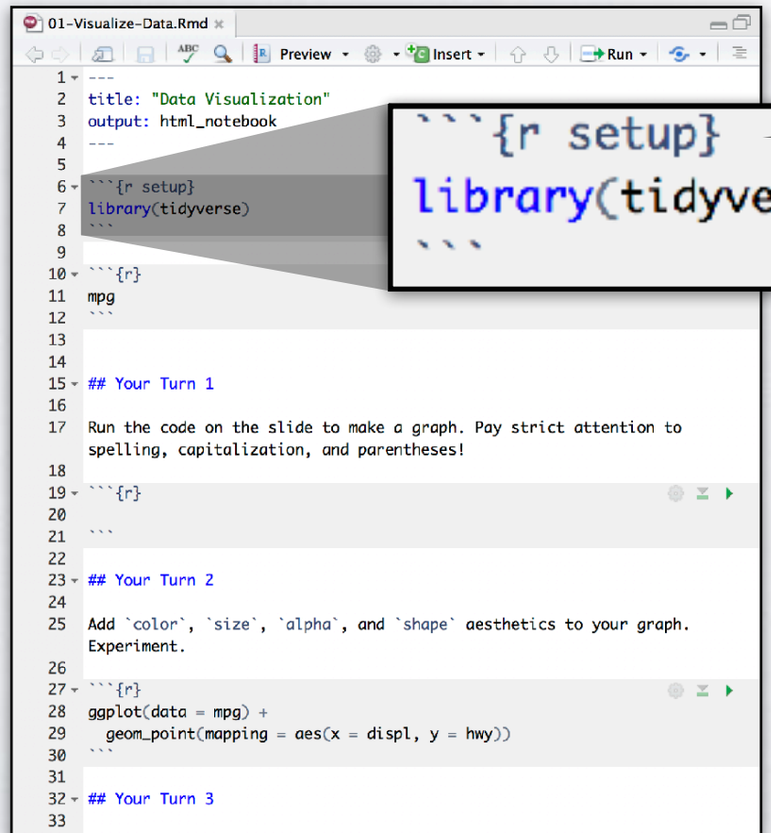
chunk labels are optional, the setup label is special

Exercise

Add a setup chunk (as shown below) to the top of 01_exercise.Rmd.

Use it to load the tidyverse package (*remember to run this chunk)

Then uncomment and run the final code chunk at the bottom of your file.

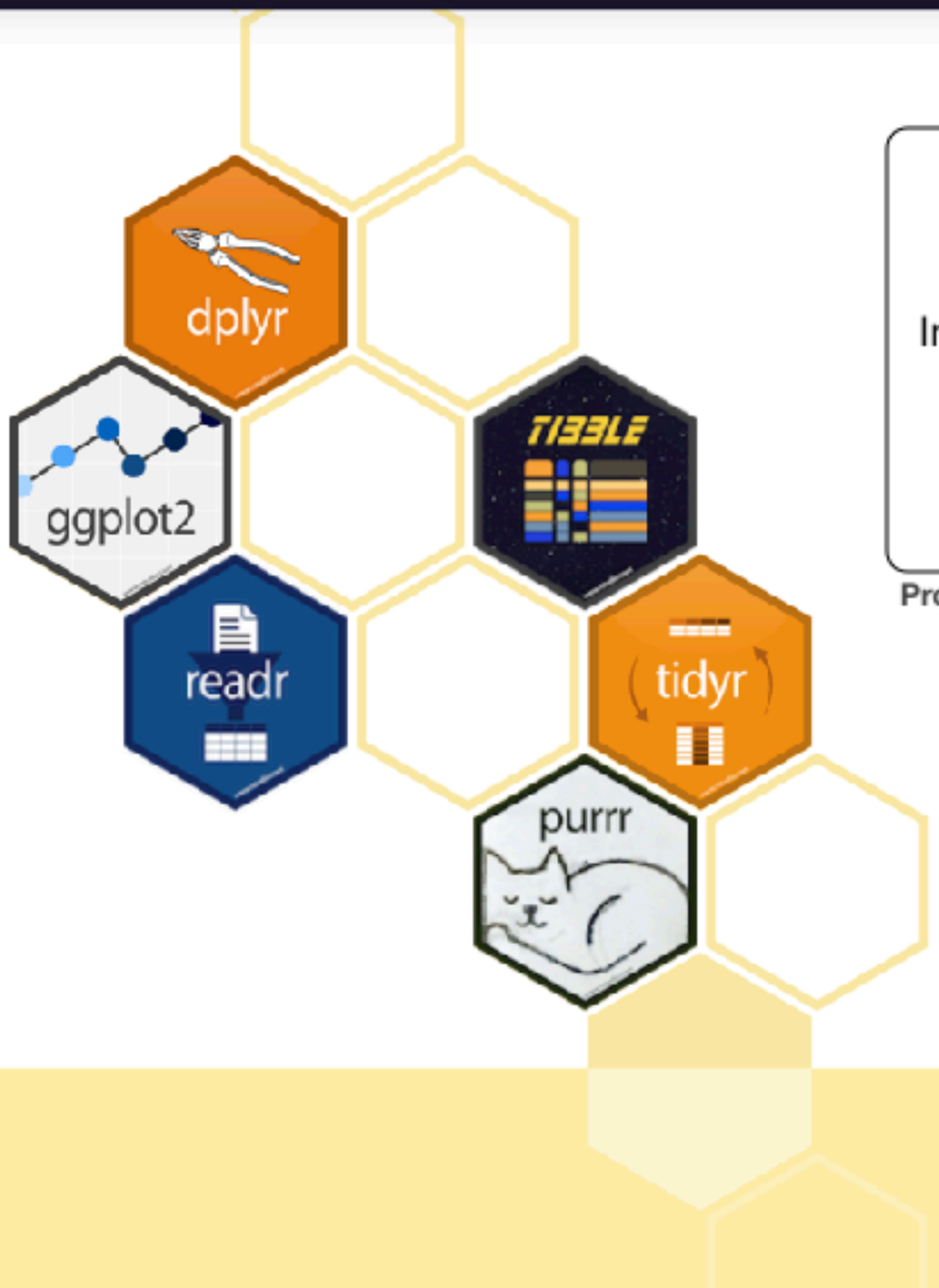


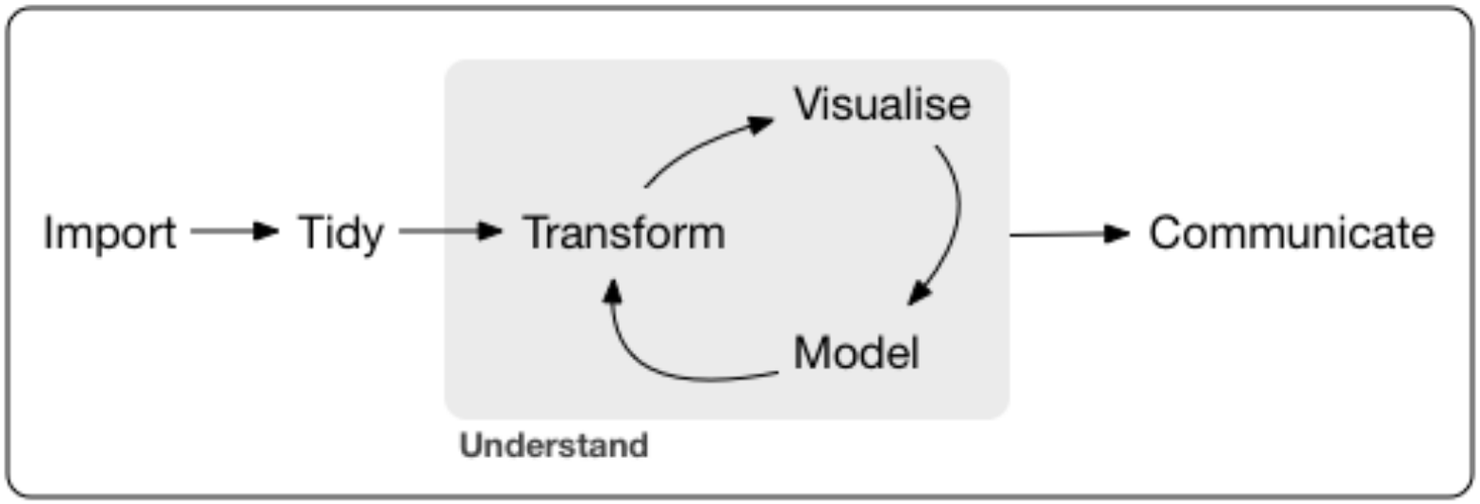

```{r setup}
library(tidyverse)
```

chunk labels are optional, the setup label is special

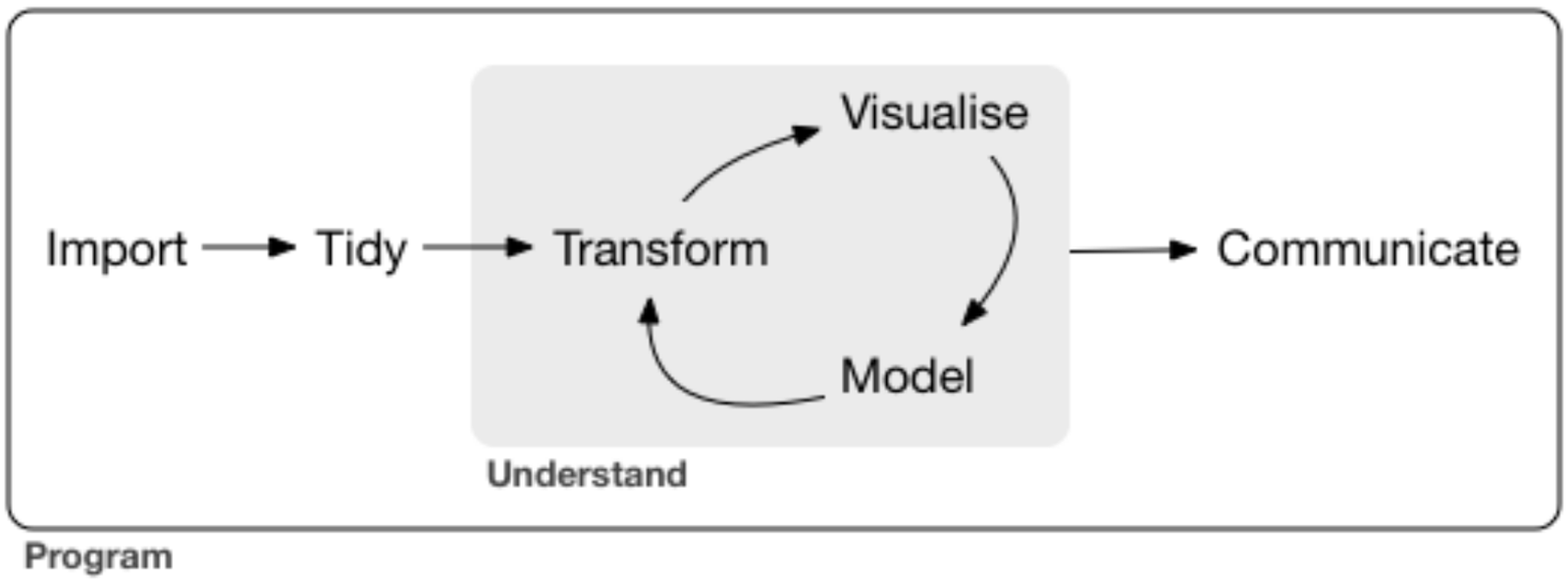
Tidyverse

Packages Articles Learn Help Contribute

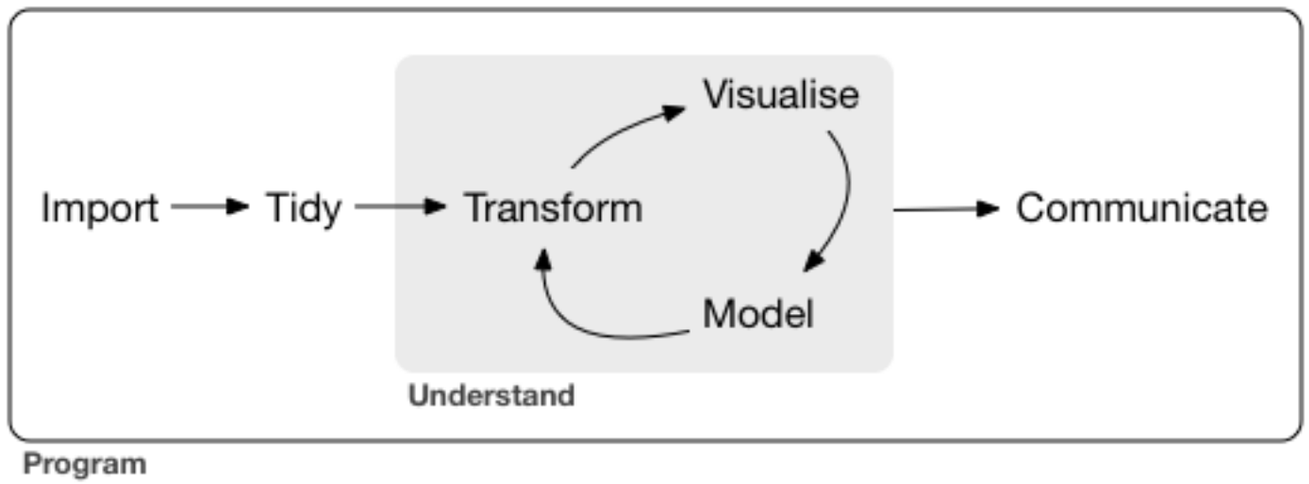




Exploratory Data Analysis (EDA)



Basic Tidyverse Principles



IMPORT (readr):

- » read_csv()
- » read_tsv()
- » read_delim()

TIDY & TRANSFORM (dplyr):

- » mutate() adds new variables that are functions of existing variables
- » select() picks variables based on their names.
- » filter() picks cases based on their values.
- » summarise() reduces multiple values down to a single summary.
- » arrange() changes the ordering of the rows.

VISUALISE (ggplot): creating graphics, based on 'The Grammar of Graphics'

- » aes()
- » geom_x() + layers

MODEL (broom):

- » tidy(), glance(), augment()
- » lm(~)

Ceci n'est pas une pipe.

Magritte

magrittr package by Stefan Milton Bache developed the concept of the pipe, which is used heavily in the tidy verse

"and then"

https://cran.r-project.org/web/packages/magrittr/vignettes/magrittr.html

The “pipe” is a sequence of functions, that are sequentially applied to an object

```
wakeup(self) %>%
  put_on("clothes") %>%
  eat("breakfast") %>%
  go(to = "work")
```

Alternative nested code

```
go(eat(put_on(wakeup(self), "clothes"), "brekfast"), to = "work")
```

What does this code do?

```
wakeup(self) %>%
  put_on("clothes") %>%
  eat("breakfast") %>%
  fmk() %>%
  go(to = "work")
```

```
morning_routine <- wakeup(self) %>%
  put_on("clothes") %>%
  eat("breakfast") %>%
  fmk() %>%
  go(to = "work")
```

The “pipe” is a sequence of functions, that are sequentially applied to an object

What does this code do?

```
chromosome %>%
  select("id", "variations", "basepairs") %>%
  mutate(percent_var = variations/basepairs) %>%
  arrange(desc(percent_var))
```

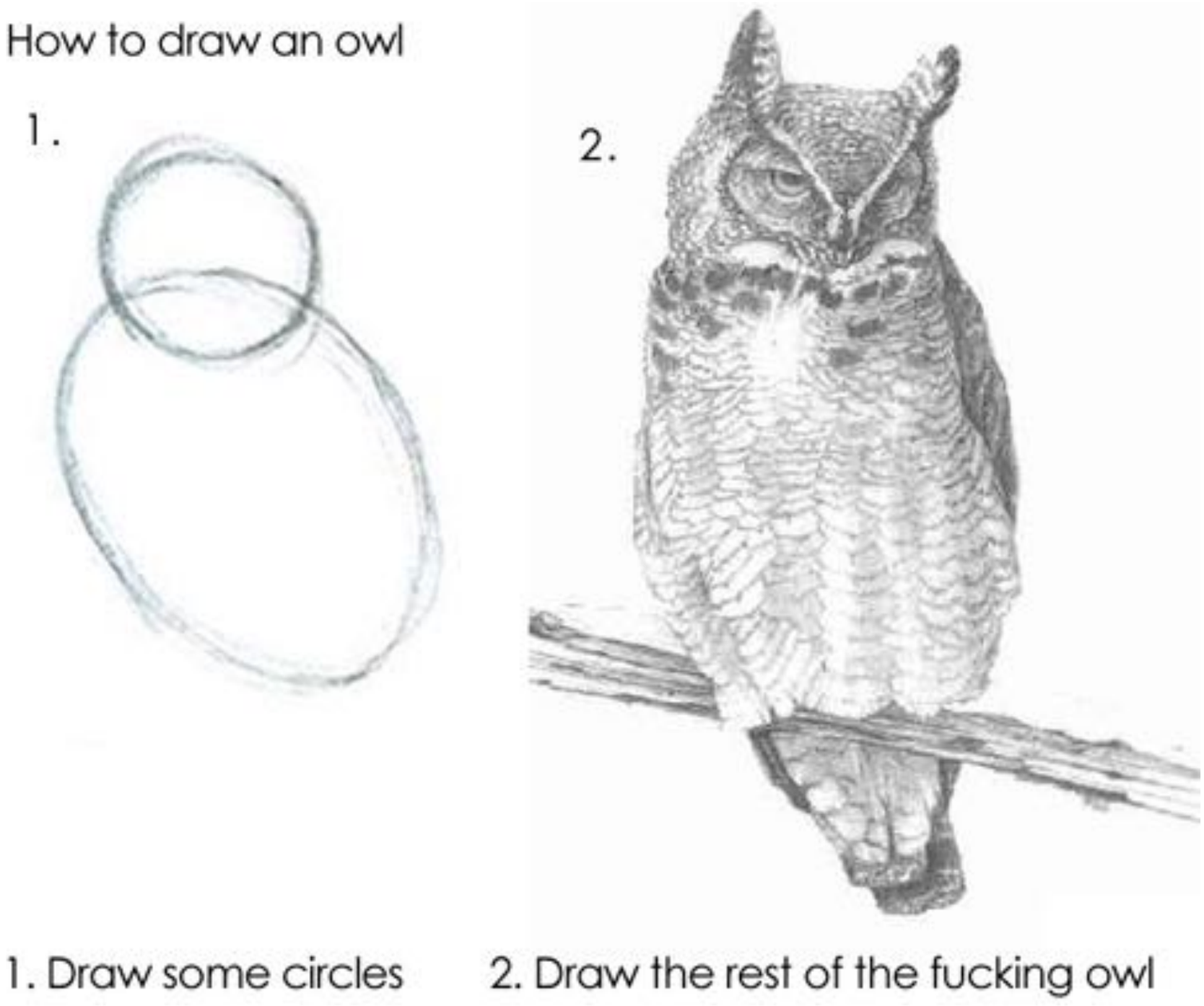
Project

Go to 01_Intro

Open 01_homework.Rmd

Read through the file and the code, to do everything it tells you to do.

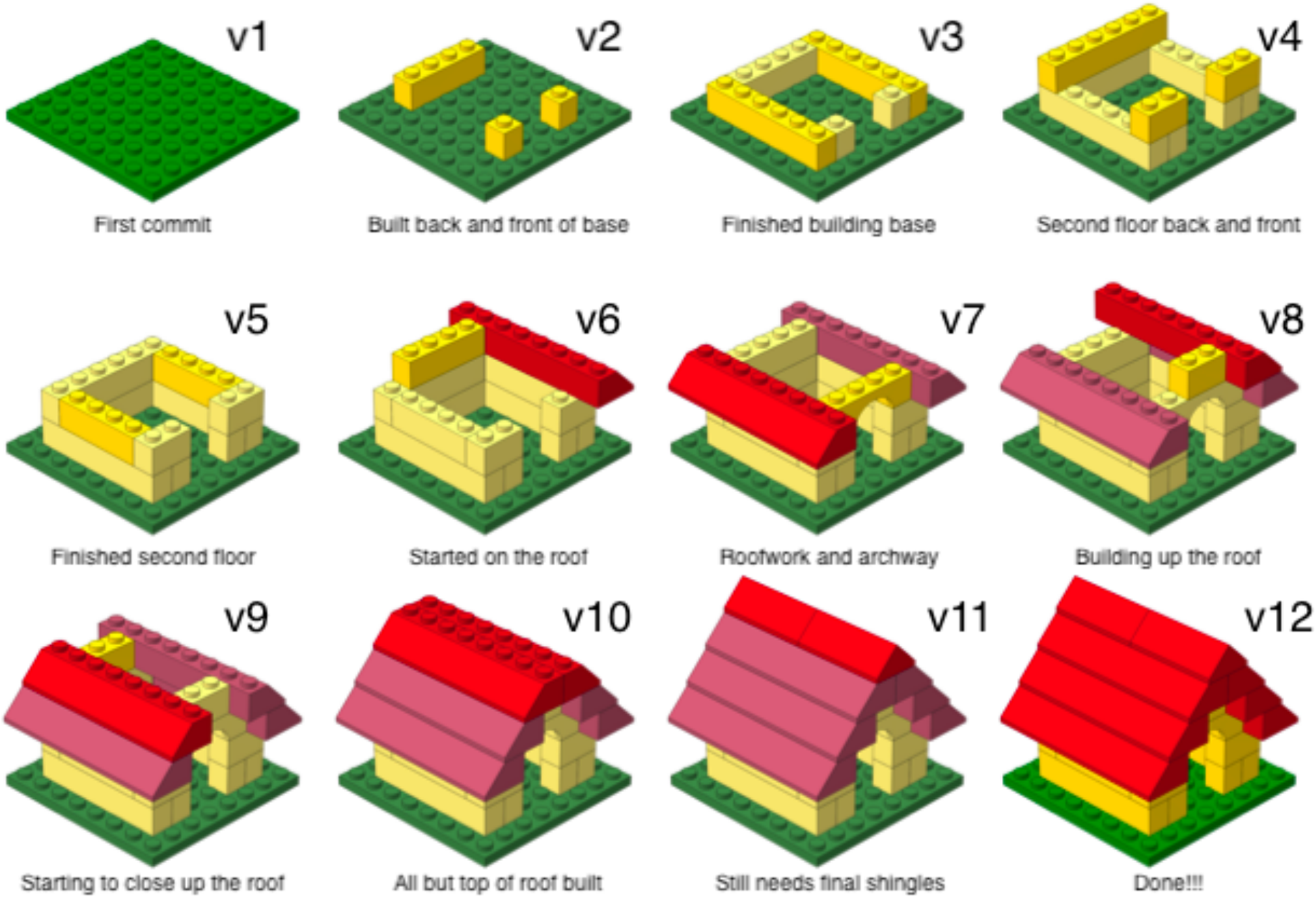
Writing code is NOT like drawing an owl



1. Draw some circles

2. Draw the rest of the fucking owl

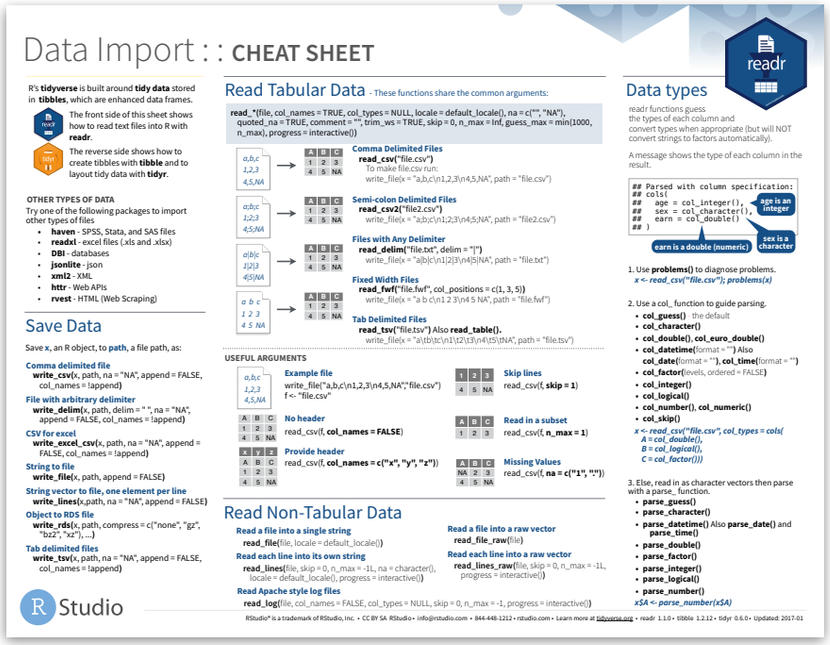
Writing code is a step-wise process

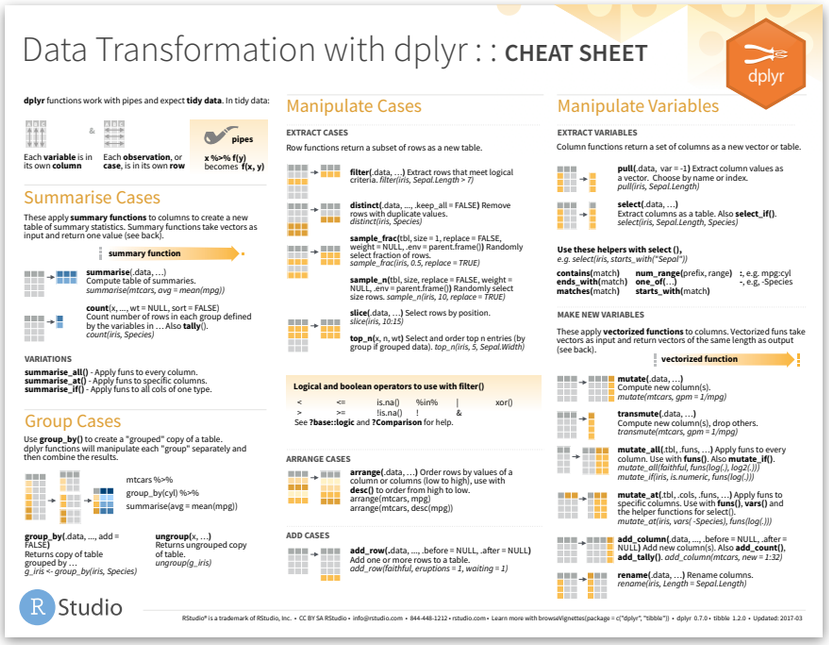


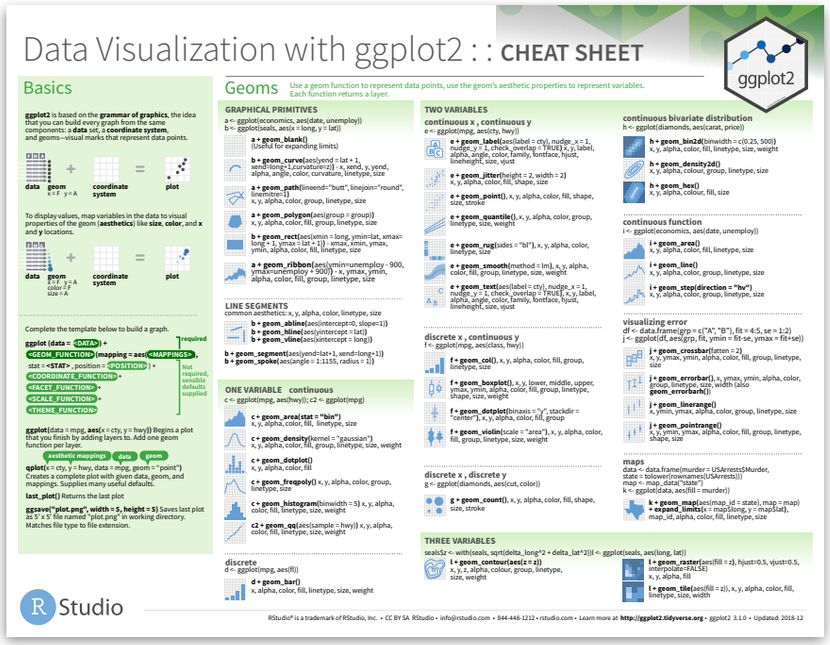
Slide images from: datasciencebox.org

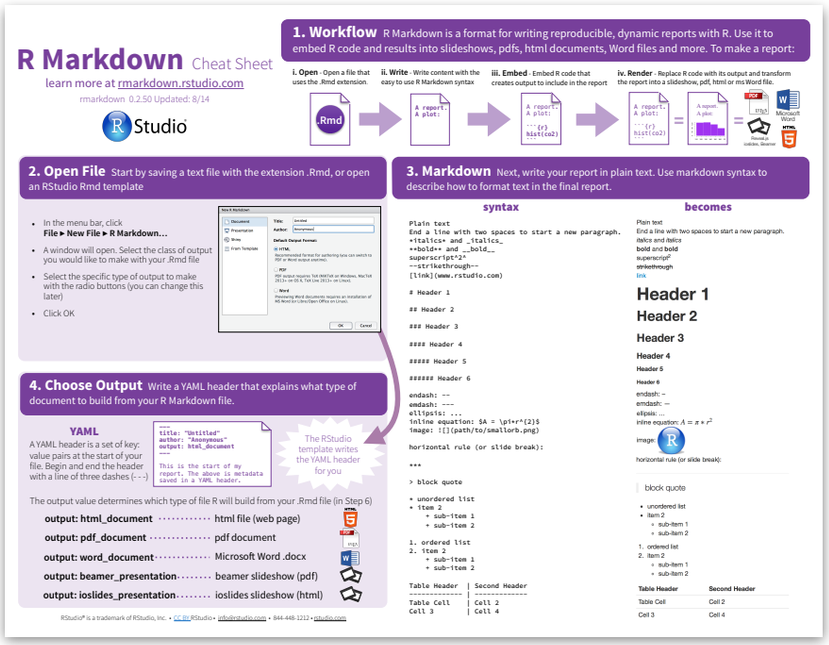
Resources: Cheat Sheets

Cheat sheets make it easy to learn about and quickly refer to function in some of the common packages.

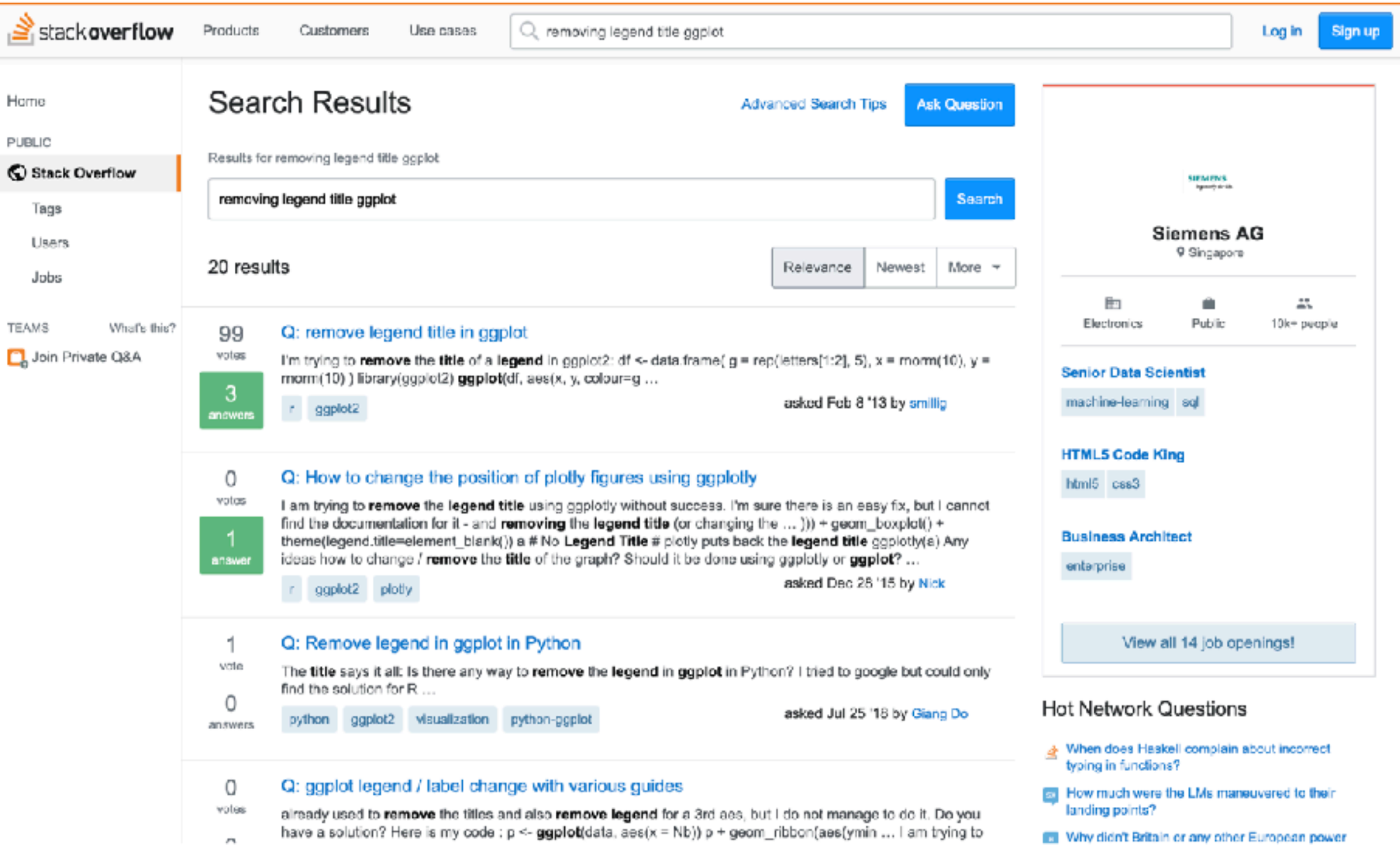








Resources: Stackoverflow



https://r4ds.had.co.nz/index.html

Resources

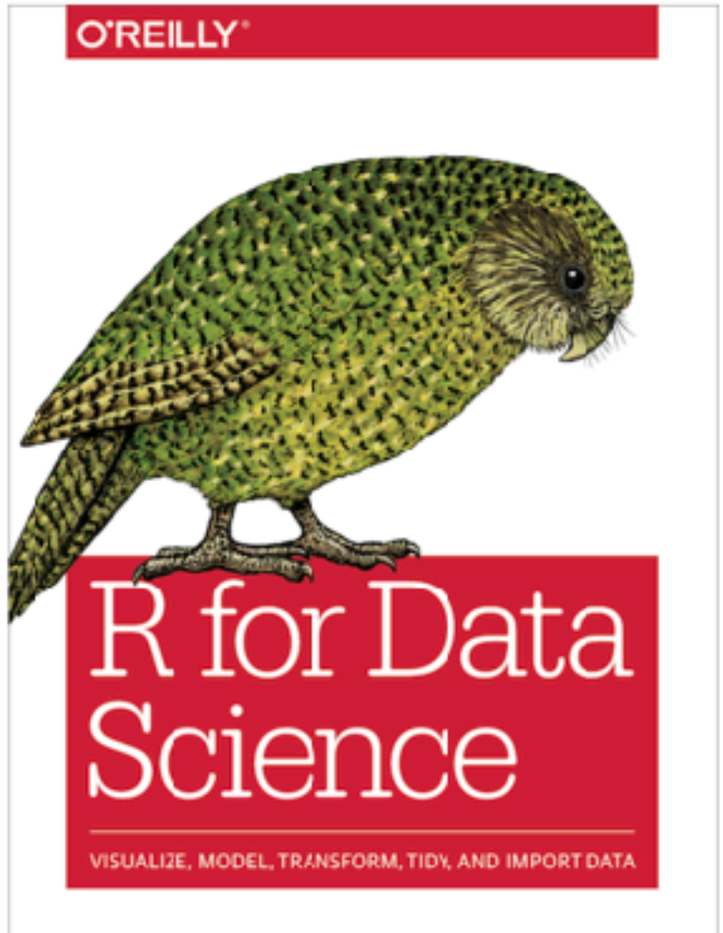
R for Data Science

Garrett Grolemund

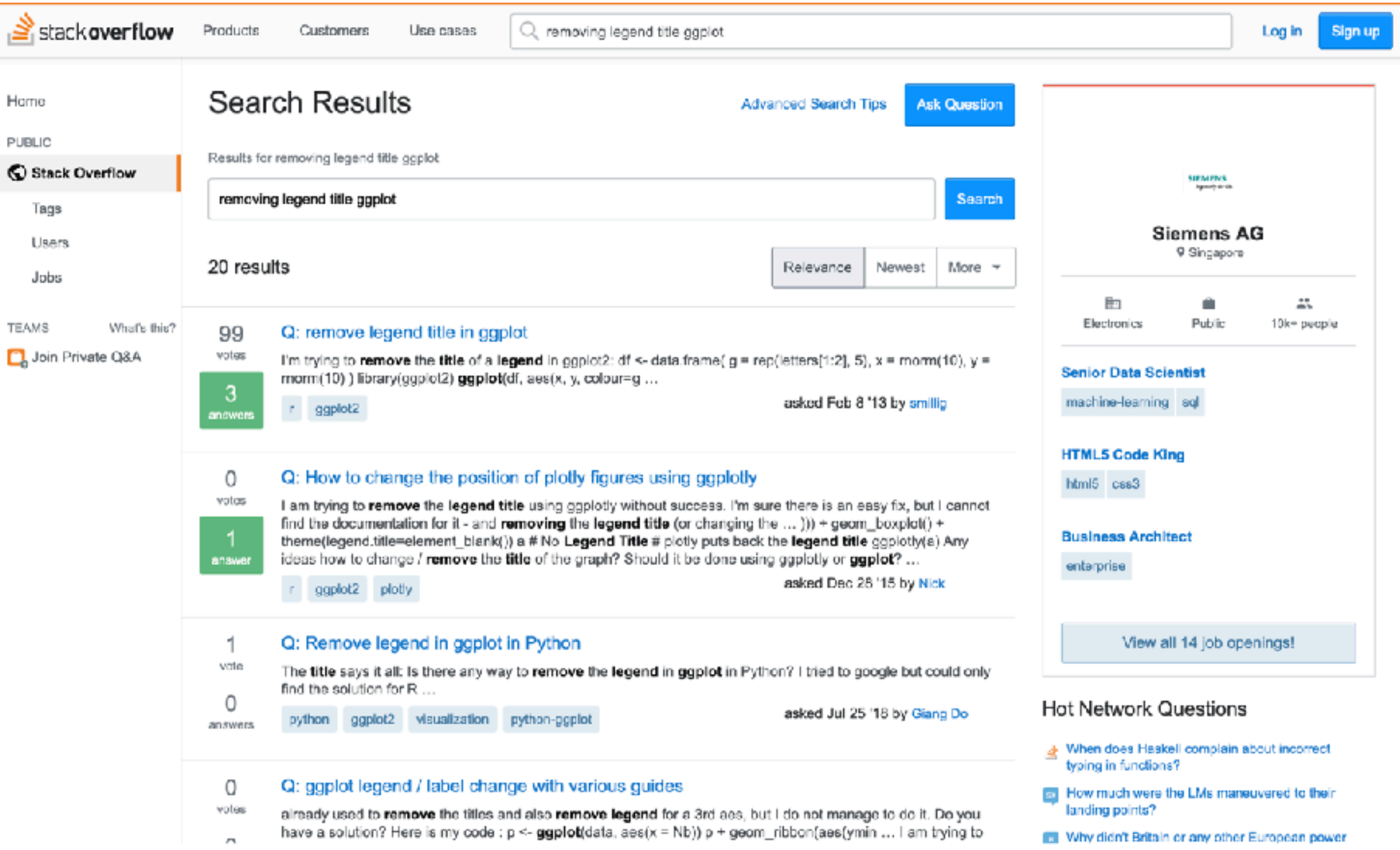
Hadley Wickham

Welcome

This is the website for **"R for Data Science"**. This book will teach you how to do data science with R: You'll learn how to get your data into R, get it into the most useful structure, transform it, visualise it and model it. In this book, you will find a practicum of skills for data science. Just as a chemist learns how to clean test tubes and stock a lab, you'll learn how to clean data and draw plots —and many other things besides. These are the skills that allow data science to happen, and here you will find the best practices for doing each of these things with R. You'll learn how to use the grammar of graphics, literate programming, and reproducible research to save time. You'll also learn how to manage cognitive resources to facilitate discoveries when wrangling, visualising, and exploring data.



Resources: Stackoverflow



Resources: Help file documentation

Type in the console >

```
>
>
> ?mean
>
```

Or navigate to the Help tab and search there, to see this

mean {base} R Documentation

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)
```

Resources: Help file documentation

The name of the function, and the library it is in.

What it does.

More details on each named argument. This will tell you what class of thing each argument has to be—an object, a number, a data frame, a logical value, etc.

What the function returns—i.e., the result of whatever operation or calculation it performs. This can be a single number, as here, or a multi-part object such as a list, a data frame, a plot, or a model.

mean {base} R Documentation

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

```
mean(x, ...)

## Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)
```

The function's name, and in the parentheses the named arguments it expects, in the order it expects them. If an argument has a default value, it is shown. Arguments without default values (e.g. x) must be provided by you.

Arguments

x	An R object. Currently there are methods for numeric/logical vectors and date, date-time and time interval objects. Complex vectors are allowed for `trim = 0`, only.
trim	the fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

The ellipsis allows other arguments to be passed to and from the function.

Value

If `trim` is zero (the default), the arithmetic mean of the values in `x` is computed, as a numeric or complex vector of length one. If `x` is not logical (coerced to numeric), numeric (including integer) or complex, `NA_real_` is returned, with a warning.
If `trim` is non-zero, a symmetrically trimmed mean is computed with a fraction of `trim` observations deleted from each end before the mean is computed.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

`weighted.mean`, `mean.POSIXct`, `colMeans` for row and column means.

Other related functions

Examples

```
x <- c(0:10, 50)
xm <- mean(x)
c(xm, mean(x, trim = 0.10))
```

Self-contained examples that you can run at the console. These may use built-in datasets or other R functions.

[Package *base* version 3.4.3 Index]

Visit the package's Index page to look for Demos and Vignettes detailing how it works.

src: *A.1.1 How to read an R help page* in "Data Visualization: A practical introduction" Kieran Healy (2018). <http://socviz.co/>

Help files

Go to helpfile_vignette in the tidybiology-2019 workspace

Open students.Rmd, run all code, and View (students) to see your assigned vignette and associated function.

Open pivot_longer.Rmd, and knit the code; be inspired.

Open helpfile_template.Rmd, and see the pre-populated code template for you to work on your own function

Each student will generate a 'help vignette' as part of the final project.