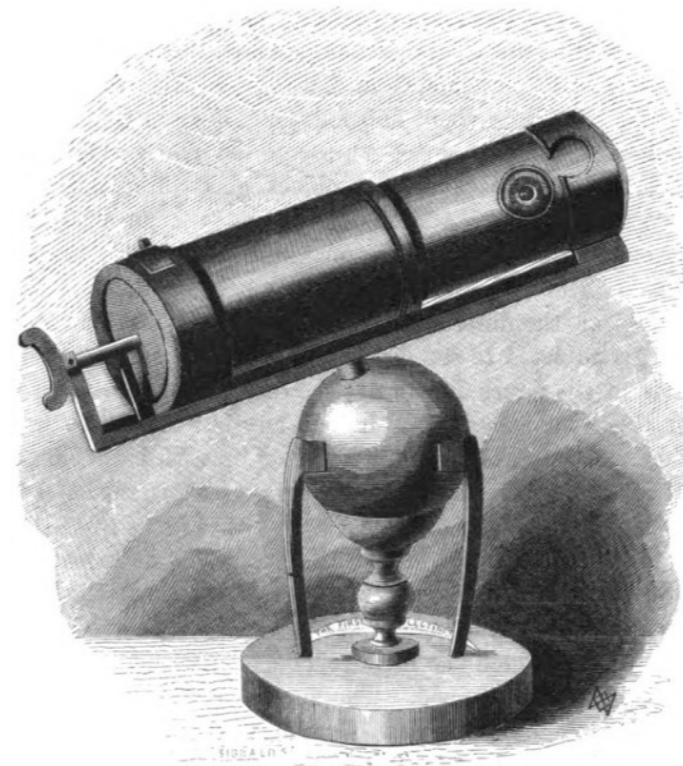


TidyBiology: An Introduction to Biological Data Science in R



Matthew Hirshey, Ph.D.

Duke University Medical Center

Department of Medicine, Pharmacology & Cancer Biology

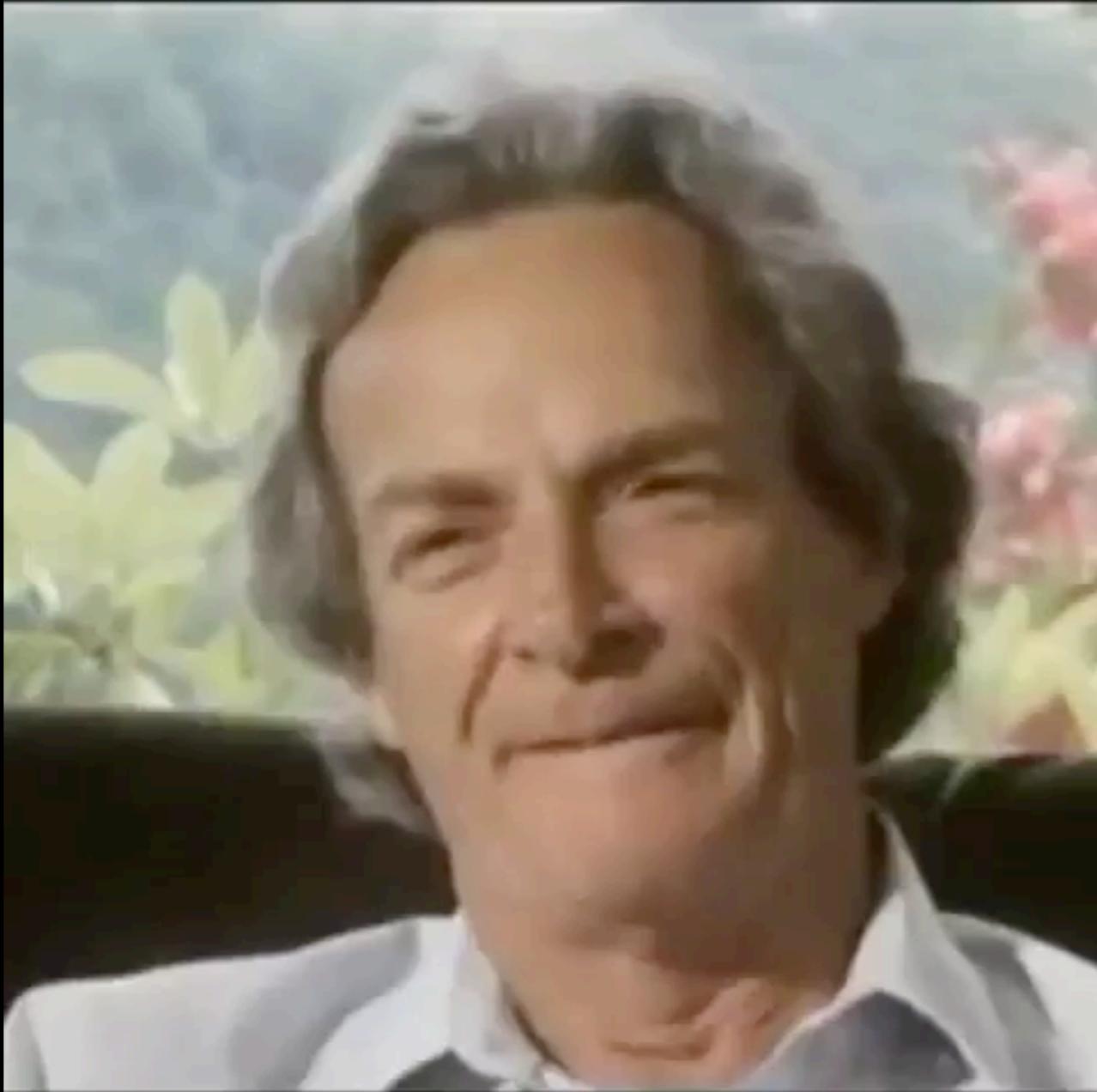
Duke Molecular Physiology Institute

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

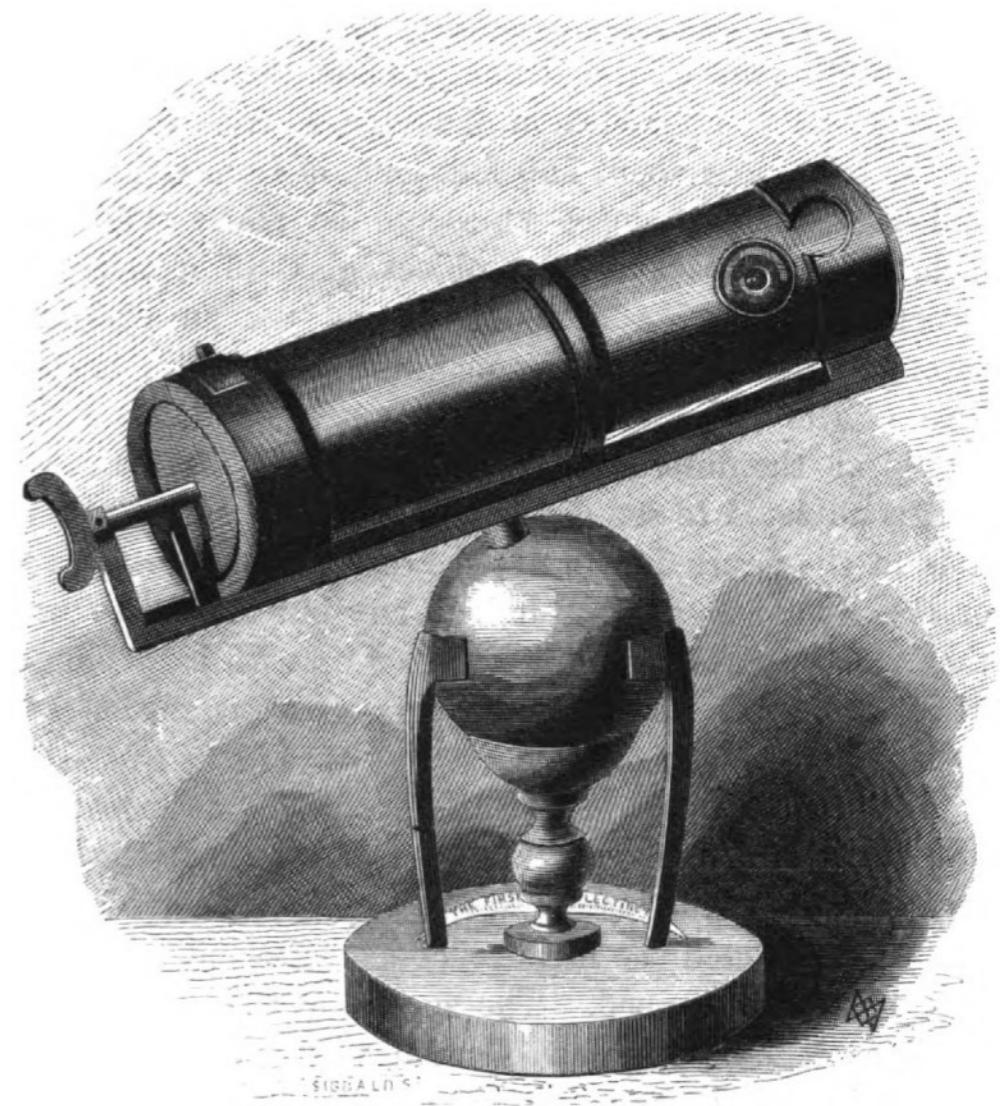
Goal of science is to know something

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)

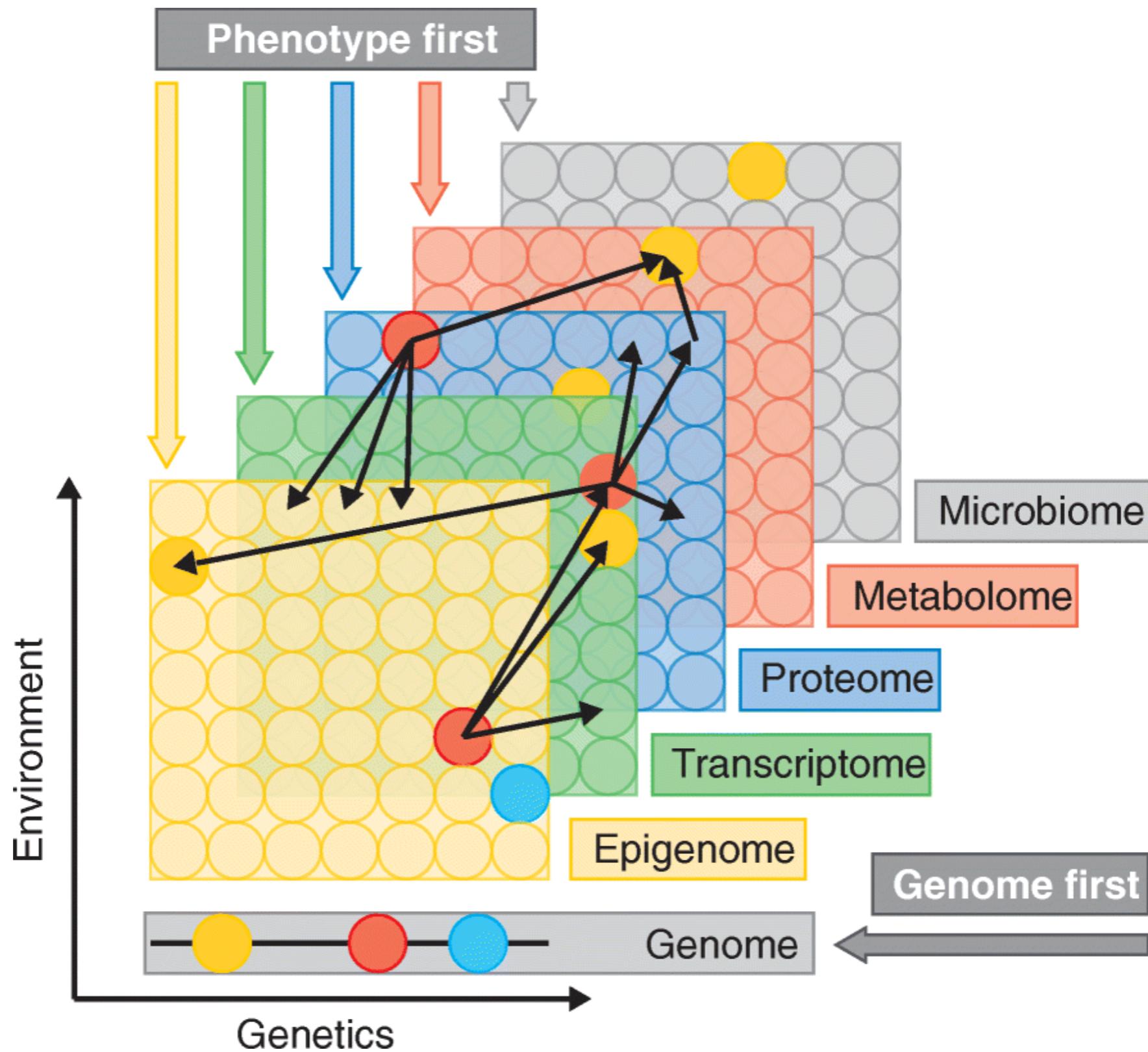
Scientific Method

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)



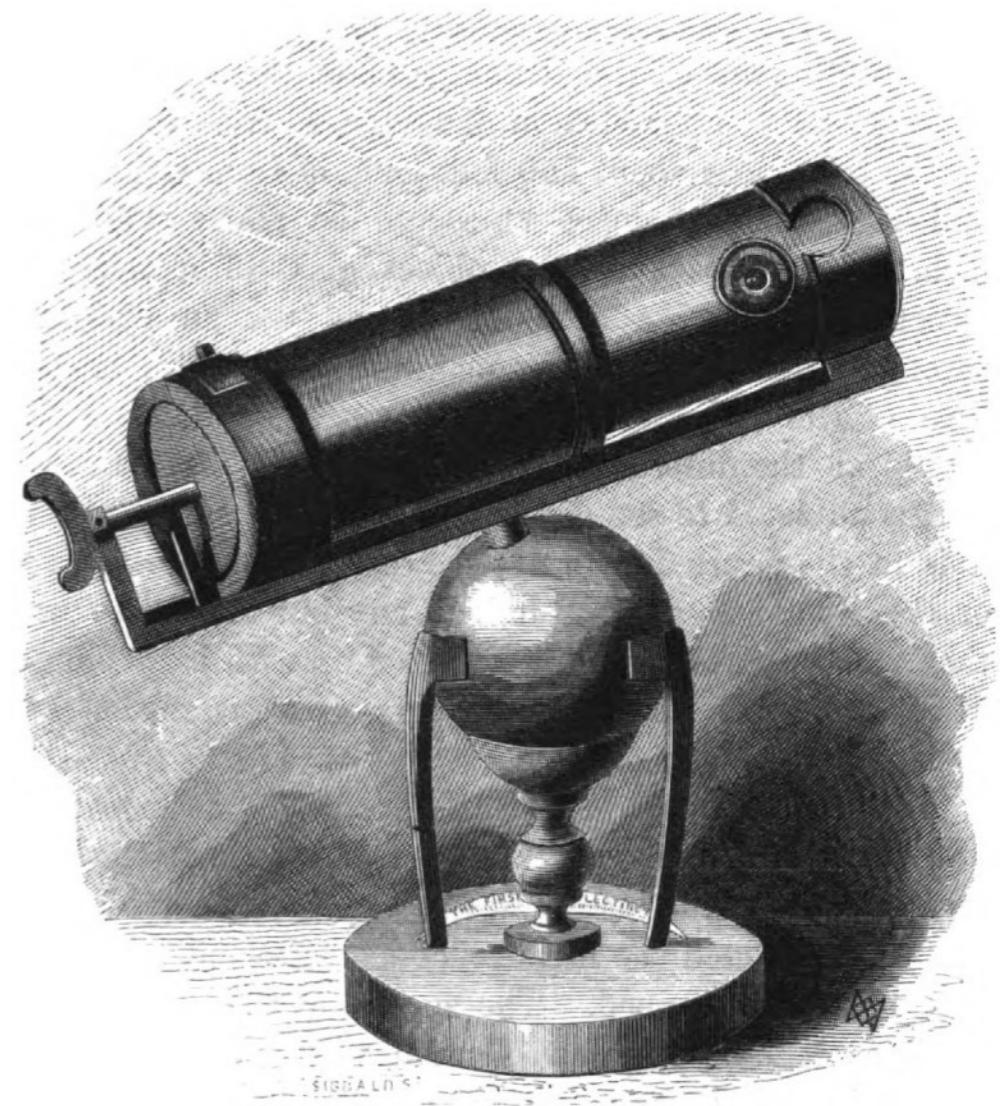
Is there enough evidence to reject the null hypothesis?

Screen

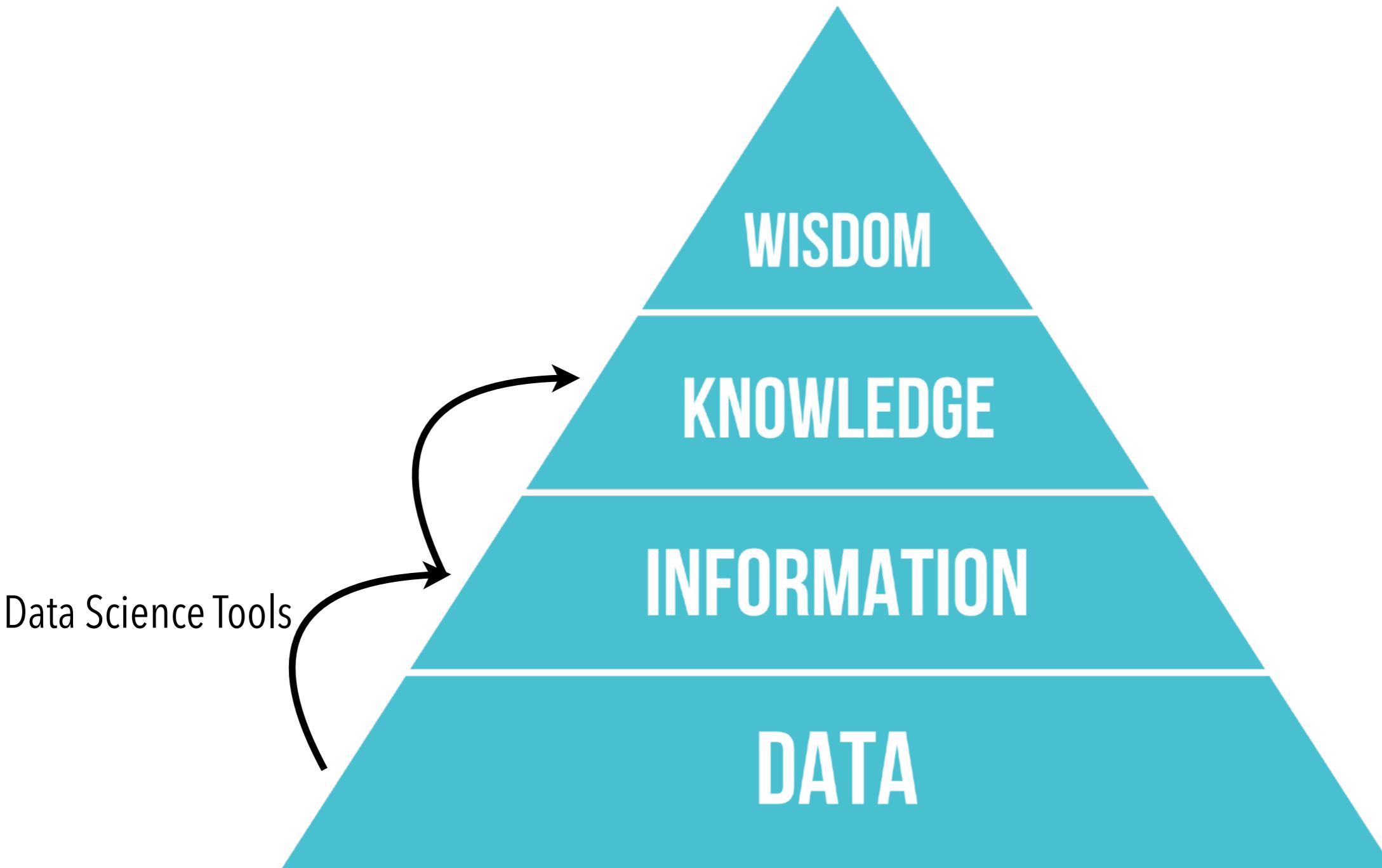


Scientific Method

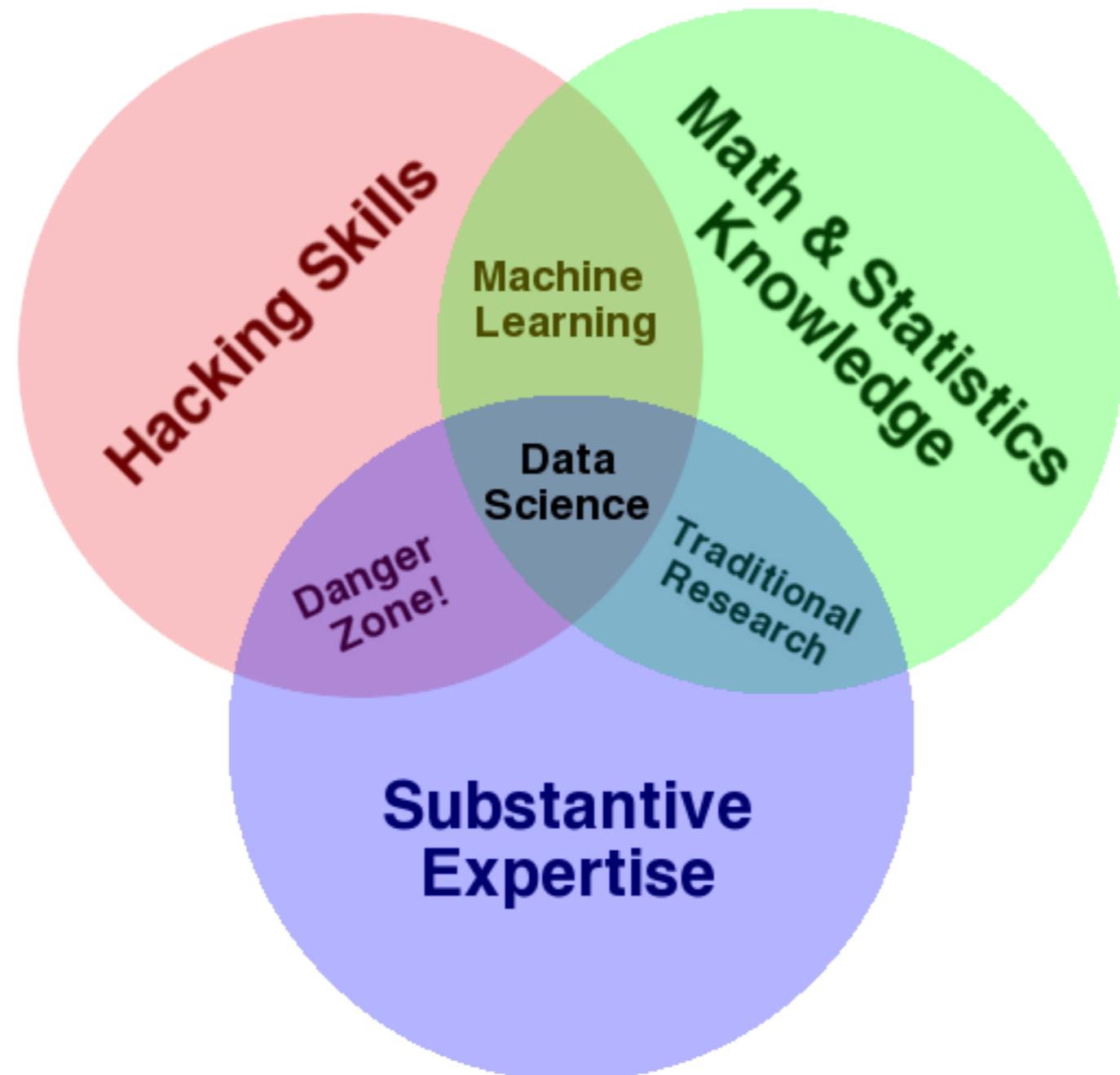
1. Define a question
2. Observe (gather information and think, **screen**)
3. Form a hypothesis
4. Test the hypothesis (performing an experiment and collecting data in a reproducible manner)
5. Analyze data (null-hypothesis significance test)
6. Interpret the data and draw conclusions (starting point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



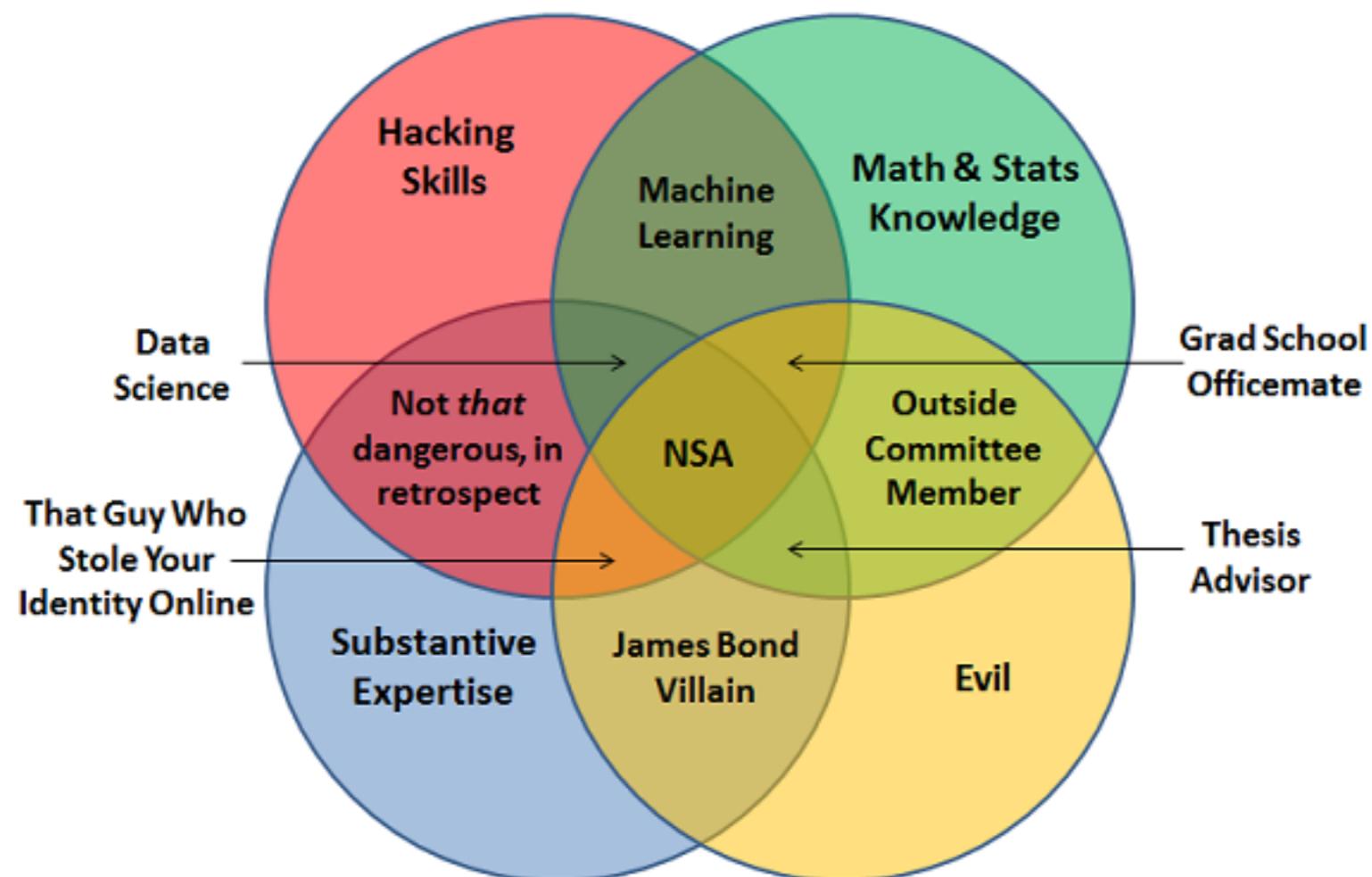
Hierarchy of Knowledge Gathering



Emerging field of Data Science



Venn Diagram of Data Science v2.0



Overall goal is to generate knowledge



Several approaches in data science

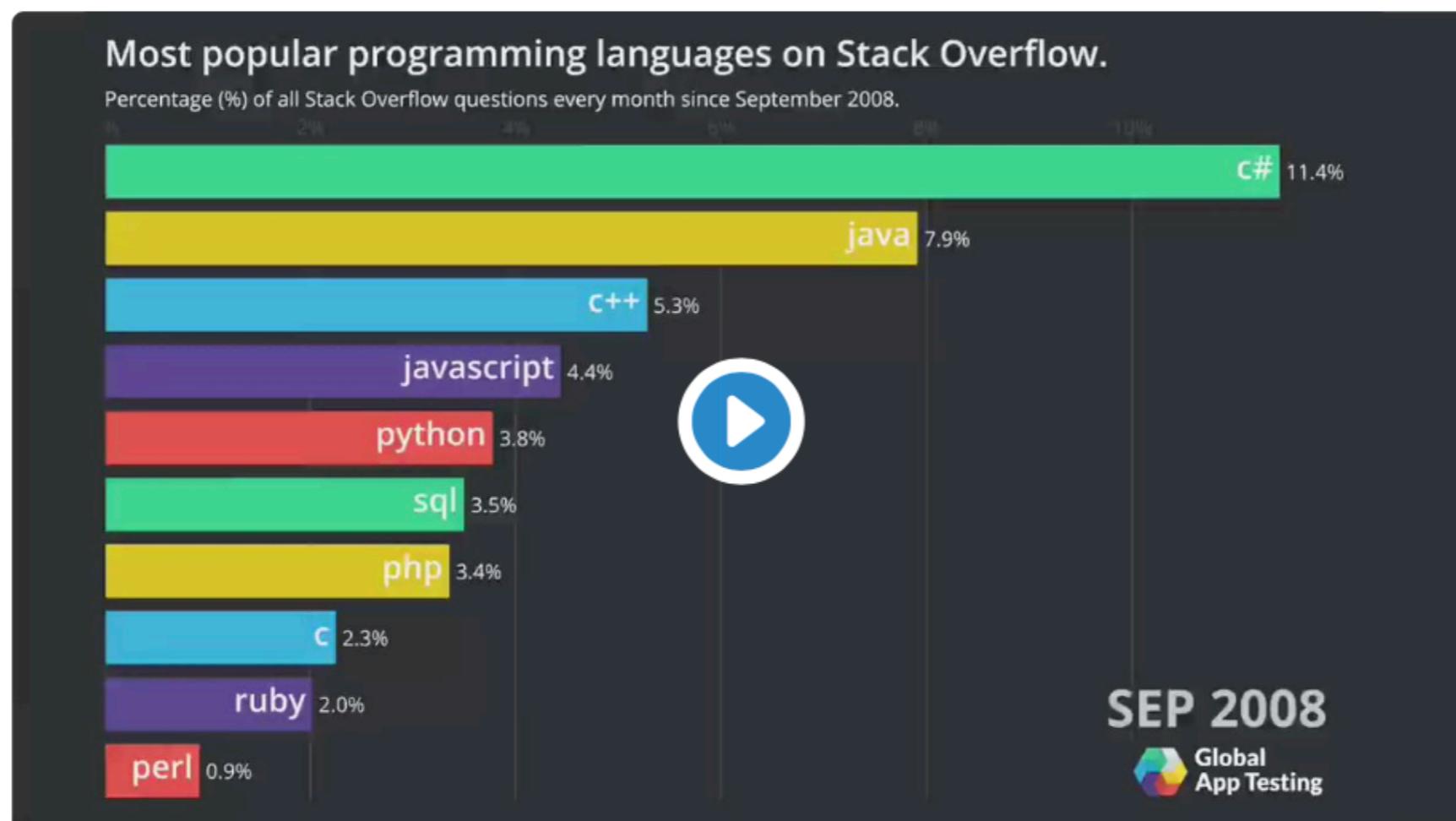


Veronika Romashkina

@vronnie911



StackOverflow language popularity through the years



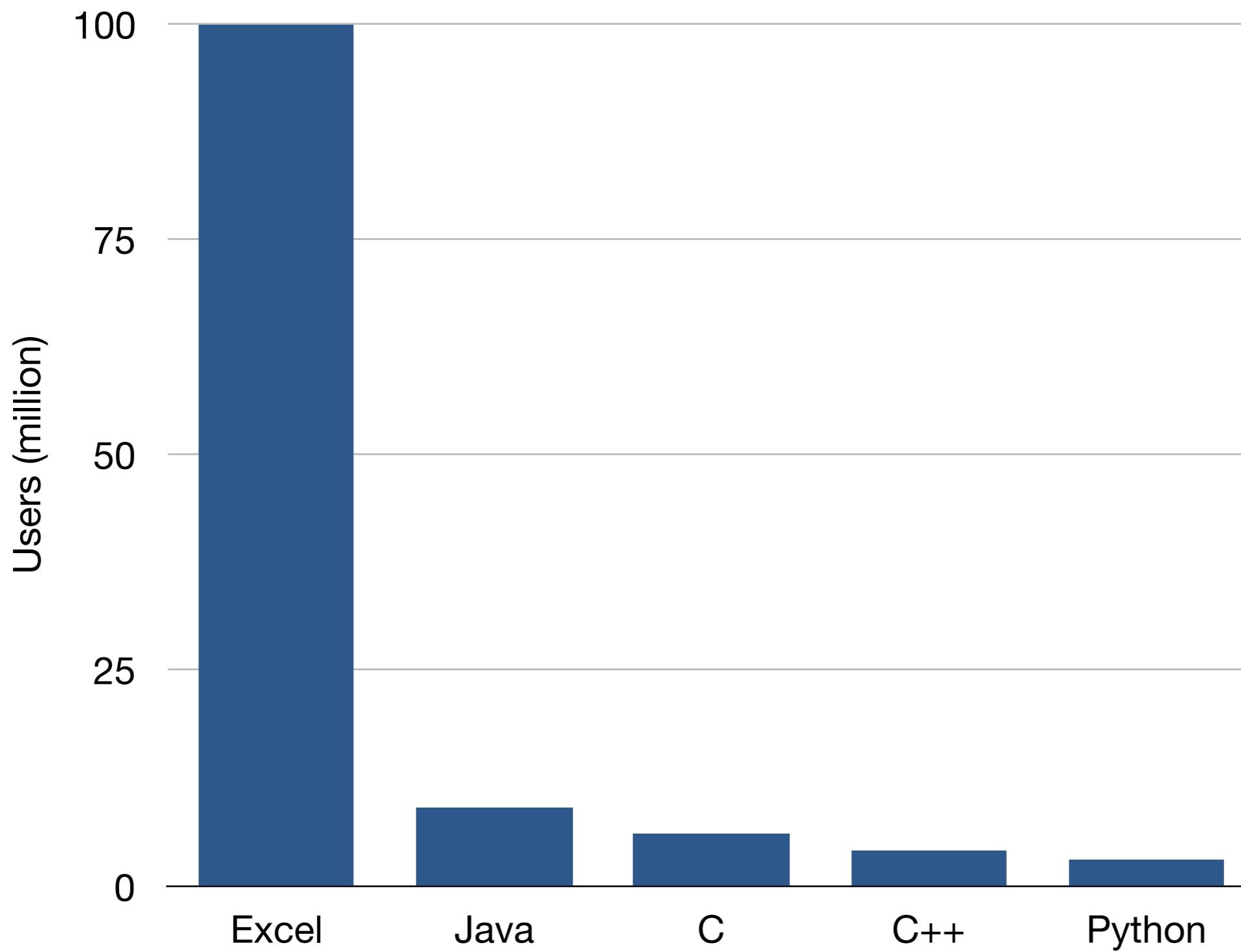
♥ 8,301 1:18 PM - May 6, 2019



4,838 people are talking about this

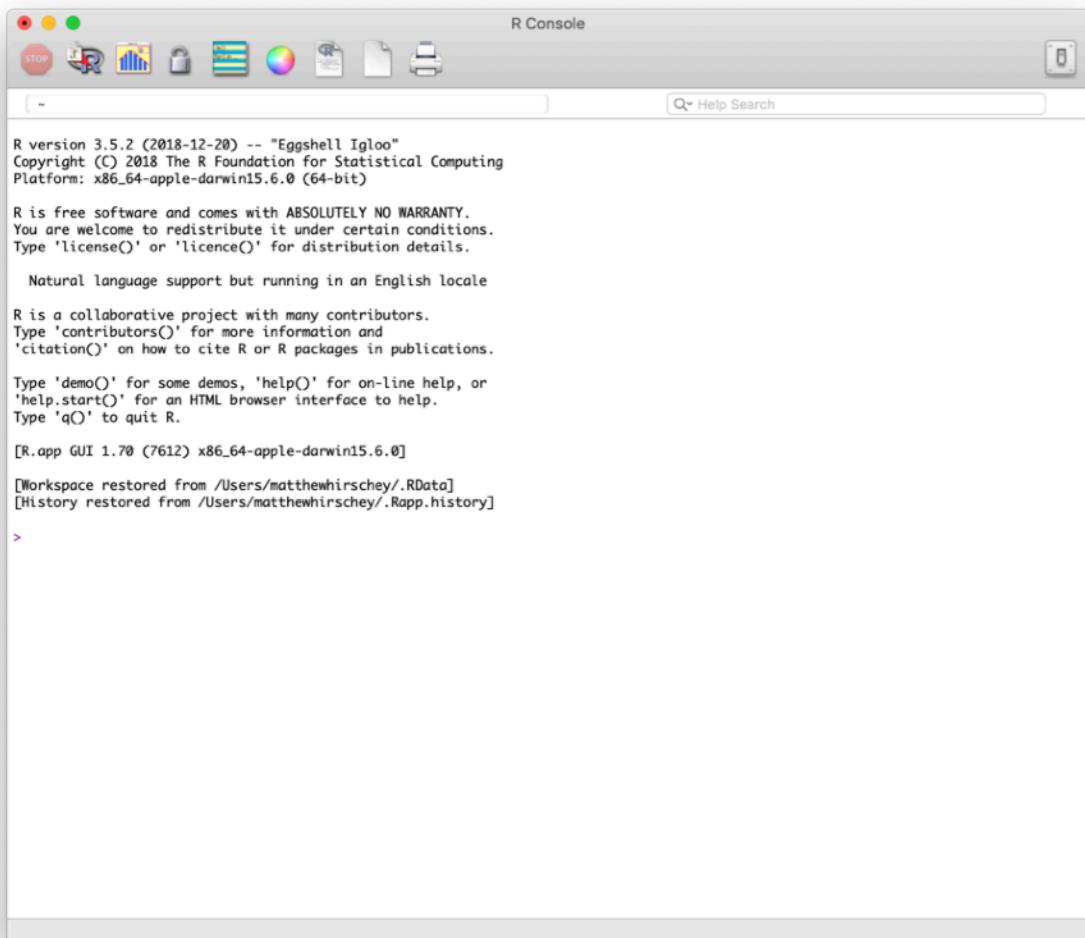


World's most popular programming languages



R

Statistical Programming Language



R version 3.5.2 (2018-12-20) -- "Eggshell Igloo"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

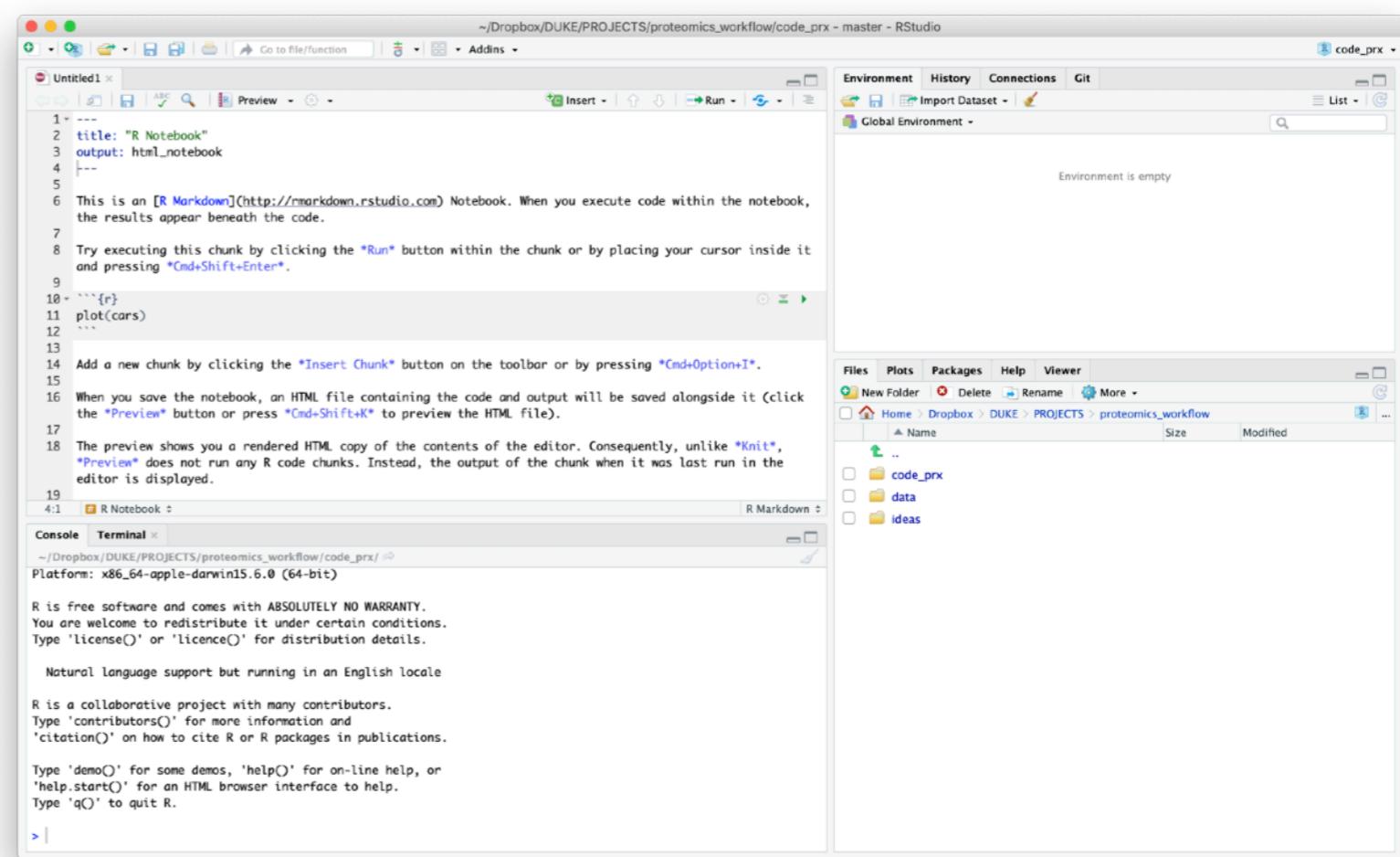
[R.app GUI 1.70 (7612) x86_64-apple-darwin15.6.0]

[Workspace restored from /Users/matthewhirshey/.RData]
[History restored from /Users/matthewhirshey/.Rapp.history]

>

RStudio

Integrated Developer Environment



Untitled1 x ~Dropbox/DUKE/PROJECTS/proteomics_workflow/code_prx - master - RStudio
1 ---
2 title: "R Notebook"
3 output: html_notebook
4 ---
5
6 This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook,
the results appear beneath the code.
7
8 Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it
and pressing *Cmd+Shift+Enter*.
9
10 ````{r}
11 plot(cars)
12 ````
13
14 Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.
15
16 When you save the notebook, an HTML file containing the code and output will be saved alongside it (click
the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).
17
18 The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*,
Preview does not run any R code chunks. Instead, the output of the chunk when it was last run in the
editor is displayed.
19
4:1 R Notebook :
Console Terminal x ~Dropbox/DUKE/PROJECTS/proteomics_workflow/code_prx/
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |





R 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.

Install the complete tidyverse with:

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

Packages contain functions, documentation, data

Install the complete tidyverse with:

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

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.

“Verbs” (i.e. functions) act on data

These “verbs” act on data

```
do_this(to_that)
```

```
do_this(to_that, using_these)
```

Data are stored in a dataframe, which is conceptually equivalent to a spreadsheet with rows and columns (data\$entry_name)

| | entry_name | protein_names | gene_name | gene_name_alt | sequence |
|----|-------------|-----------------------|-----------|--------------------|------------------------------------|
| 1 | AARD_HUMAN | Alanine and argini... | AARD | C8orf85 | MGPDFRRCRERISQGLQGLPRAELWFPPRPAC |
| 2 | ABHD1_HUMAN | Protein ABHD1 (EC... | ABHD1 | LABH1 | MLSSFLSPQNGTWADTFSLLLALAVALYLGYYWA |
| 3 | AAKG2_HUMAN | 5'-AMP-activated ... | PRKAG2 | NA | MGSAVMDTKKKDVSSPGGSGKKNASQKRRS |
| 4 | ABCG8_HUMAN | ATP-binding casse... | ABCG8 | NA | MAGKAAEERGLPKATPQDTSGLQDRLFSSES |
| 5 | ABCA7_HUMAN | ATP-binding casse... | ABCA7 | NA | MAFWTQLMLLLWKNFMYRRRQPVQLLVELLWP |
| 6 | ABCAA_HUMAN | ATP-binding casse... | ABCA10 | NA | MNKMALASFMKGRTVIGTPDEETMDIELPKKYH |
| 7 | ABCB5_HUMAN | ATP-binding casse... | ABCB5 | NA | MENSERAEMQENYQRNGTAEEQPKLRKEAVGS |
| 8 | ABLM1_HUMAN | Actin-binding LIM ... | ABLIM1 | ABLIM KIAA0059 ... | MPAFLGLKCLGKLCSEKSKVTSSERTSARGSNRI |
| 9 | ABHDA_HUMAN | Mycophenolic acid... | ABHD10 | NA | MAVARLAAVAAWVPCRSWGWAAPFGPHRG |
| 10 | ACOC_HUMAN | Cytoplasmic aconi... | ACO1 | IREB1 | MSNPFAHLAEPLDPVQPGKKFFNLNKLEDSRYG |
| 11 | ACH10_HUMAN | Neuronal acetylch... | CHRNA10 | NACHRA10 | MGLRSHHLSLGLLLFLLPAECLGAEGRLALKFR |

Extract or create new objects

You can call a single part of the data frame

```
data$entry_name
```

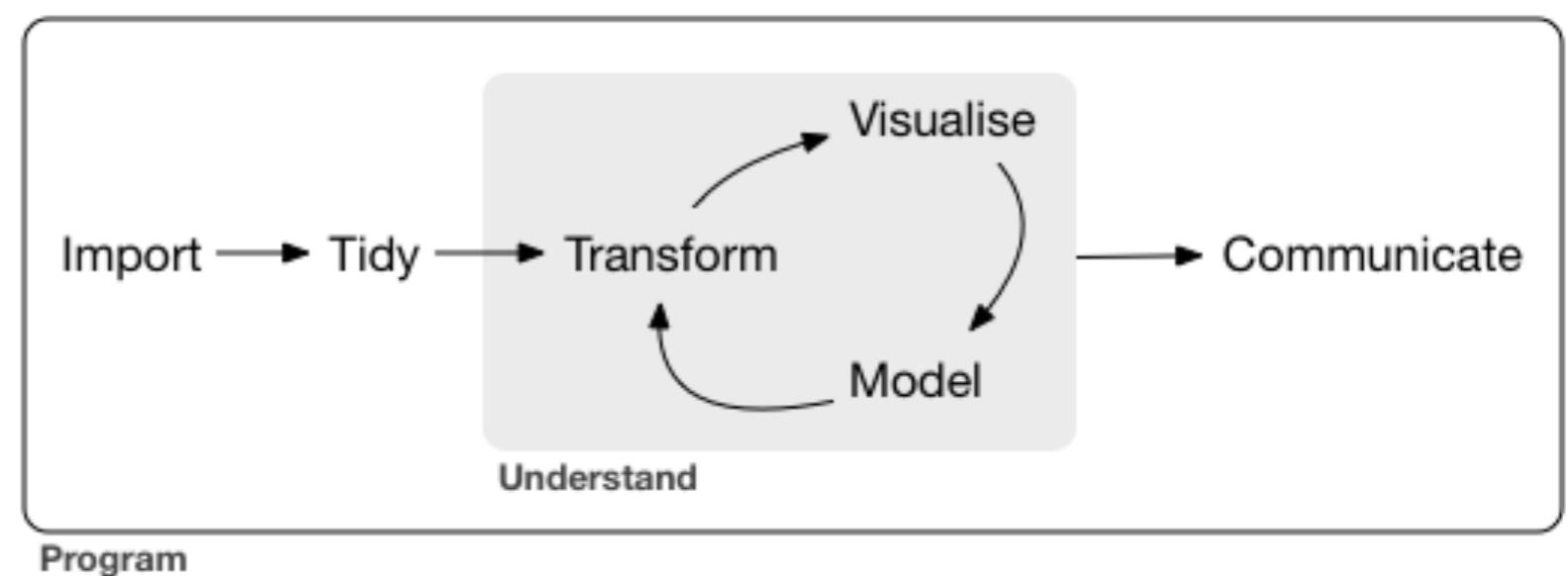
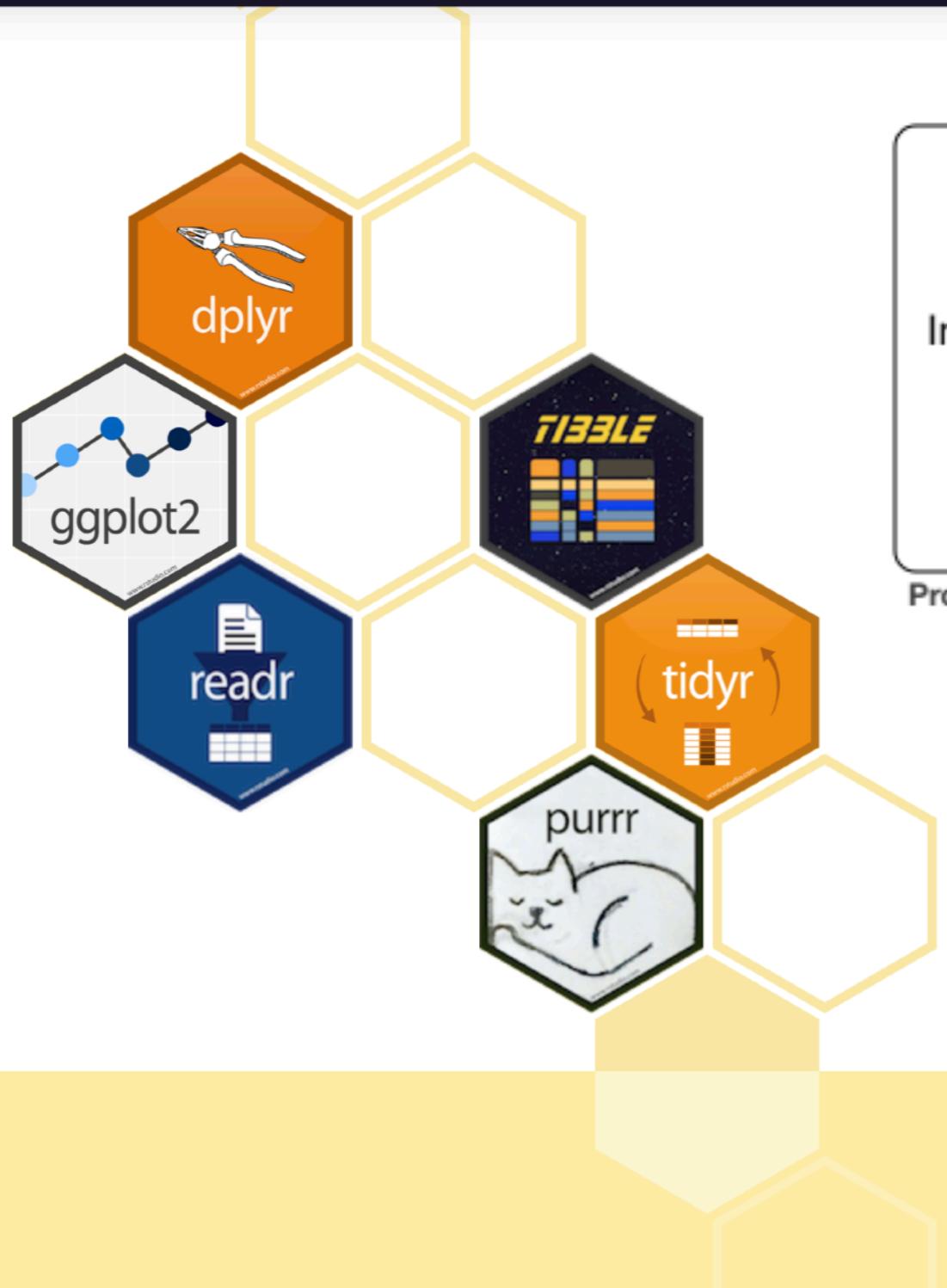
| | entry_name |
|----|-------------|
| 1 | AARD_HUMAN |
| 2 | ABHD1_HUMAN |
| 3 | AAKG2_HUMAN |
| 4 | ABCG8_HUMAN |
| 5 | ABCA7_HUMAN |
| 6 | ABCAA_HUMAN |
| 7 | ABCB5_HUMAN |
| 8 | ABLM1_HUMAN |
| 9 | ABHDA_HUMAN |
| 10 | ACOC_HUMAN |

And save it as an object for later use

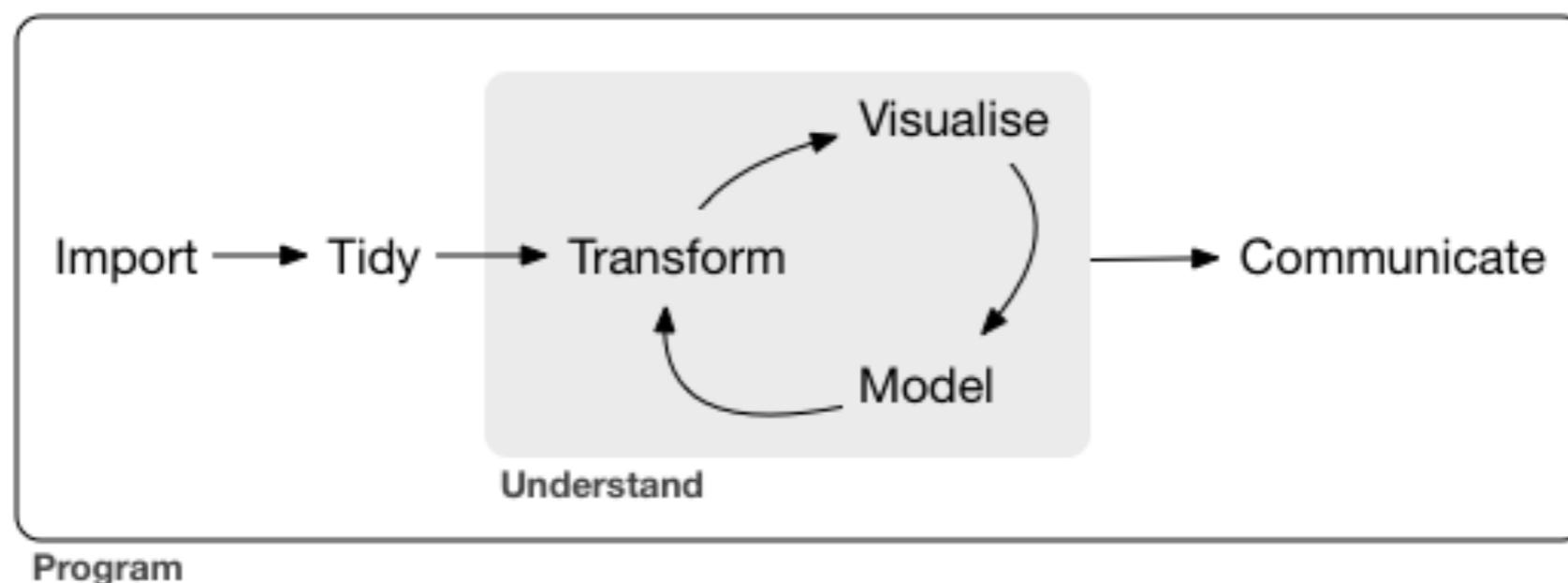
```
entries <- data$entry_name
```

Tidyverse

Packages Articles Learn Help Contribute



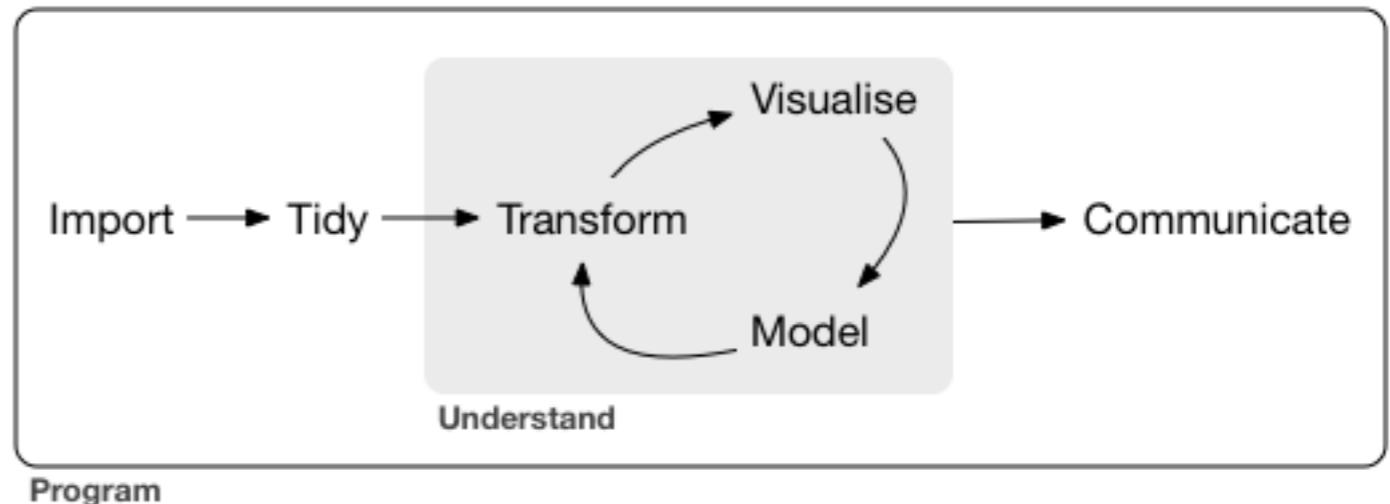
Exploratory Data Analysis (EDA)



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.

%>%
magrittr

Ceci n'est pas un pipe.

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

“and then”

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?

```
data %>%  
  select("gene", "expression_level_control", "expression_level_experimental") %>%  
  mutate(differential_expression_level = expression_level_experimental - expression_level_control) %>%  
  arrange(desc(differential_expression_level))
```

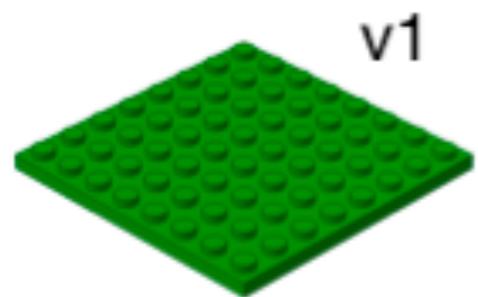
```
ranked_dge <- data %>%  
  select("gene", "expression_level_control", "expression_level_experimental") %>%  
  mutate(differential_expression_level = expression_level_experimental - expression_level_control) %>%  
  arrange(desc(differential_expression_level))
```

Demonstration:

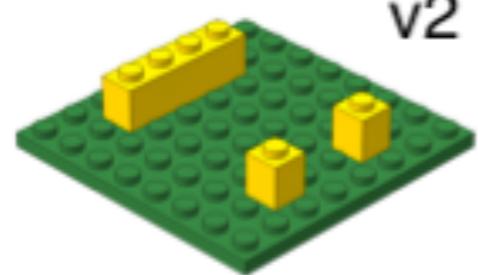
Human protein characterization

- Import
- Tidy data
- Exploratory Data Analysis (EDA): an approach to analyzing data sets to summarize their main characteristics, often with visual methods.

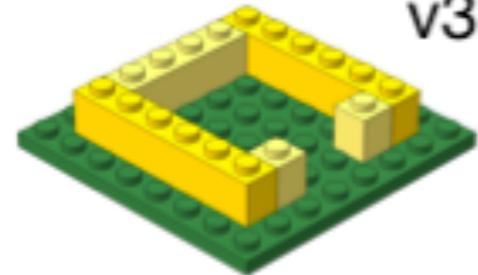
Writing code is a step-wise process



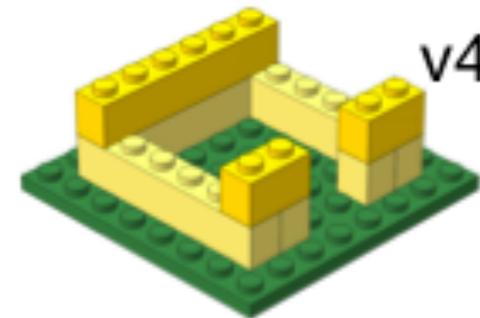
v1



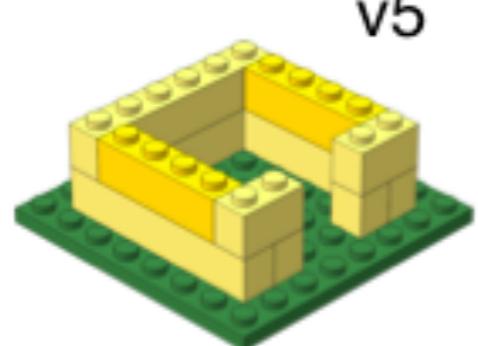
v2



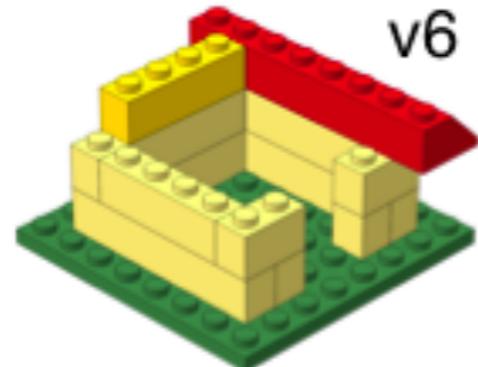
v3



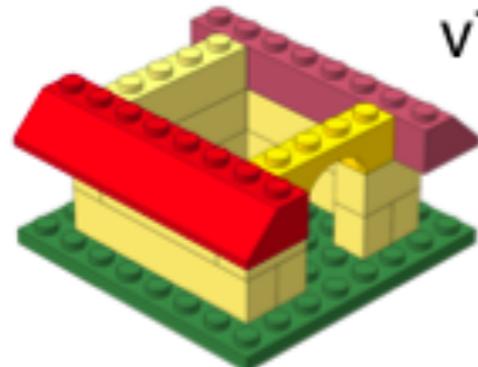
v4



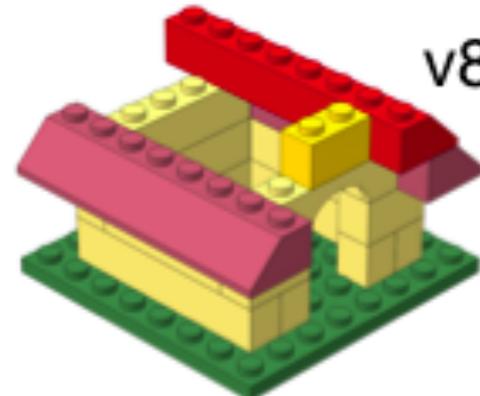
v5



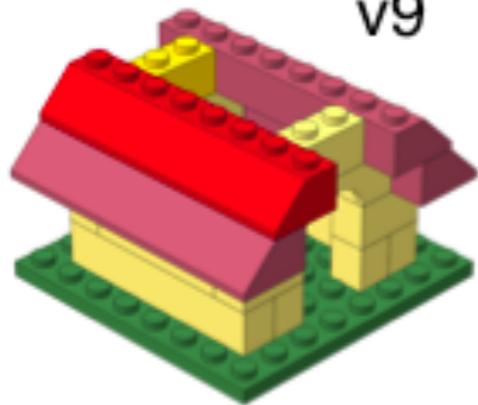
v6



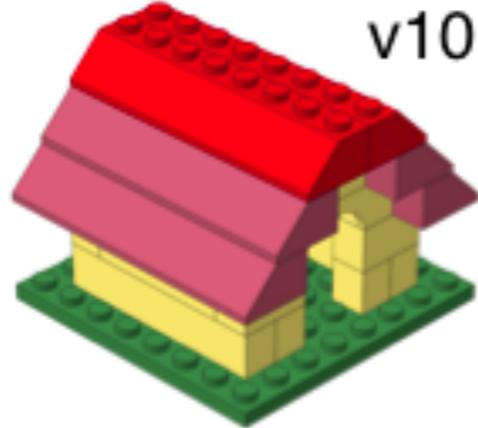
v7



v8



v9



v10

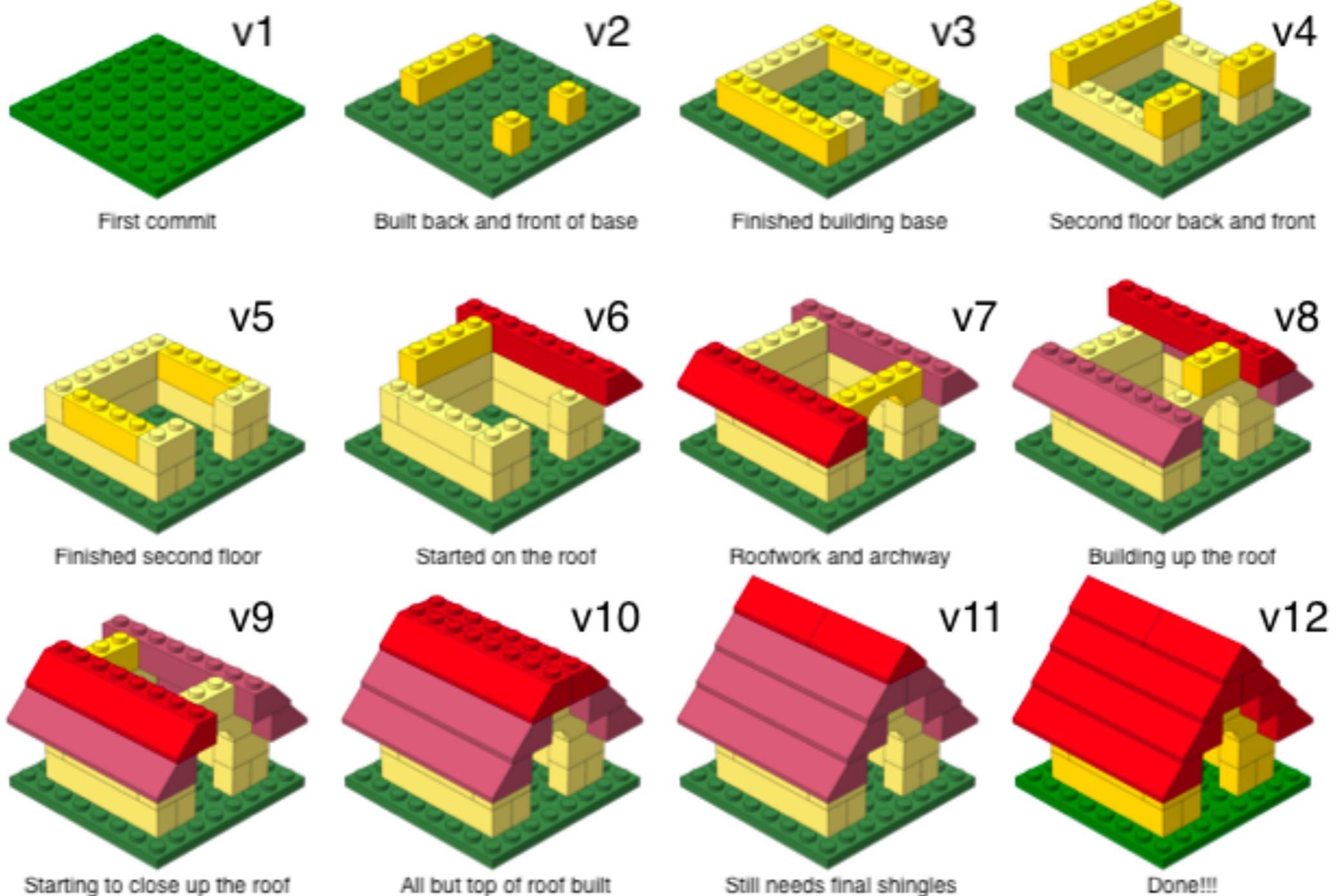


v11



v12

Writing code is a step-wise process



Demonstrations

- Gene co-expression analysis (SIRT5)
- Correlations of DepMap dependencies?
- Correlations of AA signatures? Clustering

Demonstration: PTM analysis

hirscheylab / ptm2019

Watch ▾ 0

Star 0

Fork 0

Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

Settings

Branch: master ▾

ptm2019 / ptm.Rmd

Find file Copy path



hirscheylab updated figure numbers, major code update, minor header update

617f57d on Jun 10

1 contributor

464 lines (385 sloc) | 19.3 KB

Raw

Blame

History



```
1 ---  
2 title: "The Growing Landscape of Protein Modifications"  
3 author: "E. Keith Keenan & Matthew D. Hirschey"  
4 date: "`r format(Sys.time(), '%B %d, %Y')`"  
5 output:  
6   pdf_document:  
7     toc: yes  
8   html_notebook:  
9     df_print: paged  
10    toc: yes  
11 ---  
12 This is an [R Markdown](http://rmarkdown.rstudio.com) notebook accompanying a review on protein modifications. When you execute  
13  
14 ##Load libraries  
15 ```{r load_block, warning=FALSE, echo=TRUE}  
16 library(tidyverse)  
17 library(janitor)  
18 library(viridis)  
19 library(XML)
```

Conclusions

Data science enables **data-driven** information gathering and hypothesis generation

➡ Scientific Research

➡ Reviews

Data science enables the ability to ask new types of questions

Process centric, not necessarily gene centric

Making things computable makes them actionable at zero marginal cost.

Workflows save time, achieve reproducibility

Resources: Cheat Sheets

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

Data Import :: CHEAT SHEET

R's tidyverse is built around `tidy data` stored in `tibbles`, which are enhanced data frames. The front side of this sheet shows how to read files into R with `readr`. The reverse side shows how to create tibbles with `tibble` and layout tidy data with `tidyR`.

OTHER TYPES OF DATA
Try one of the following packages to import other types of files

- `haven` - SPSS, Stata, and SAS files
- `readxl` - excel files (.xls and .xlsx)
- `DBI` - databases
- `jsonlite` - JSON
- `xml2` - XML
- `httr` - Web APIs
- `rvest` - HTML (Web Scraping)

Save Data
Save x, an R object, to path, a file path, as:

```
write_csv(x, path) #> "NA", append = FALSE, col_names = (append)
write_xls(x, path) #> "NA", append = FALSE, col_names = (append)
write_rds(x, path) #> "NA", append = FALSE, col_names = (append)
write_json(x, path) #> "NA", append = FALSE, col_names = (append)
write_xml(x, path) #> "NA", append = FALSE, col_names = (append)
write_html(x, path) #> "NA", append = FALSE, col_names = (append)
```

Read Tabular Data These functions share the common arguments:

```
read(*file, col_names = TRUE, col_types = NULL, locale = default_locale(), na = "", NA, quoted_na = TRUE, comment = "", trim_ws = TRUE, skip = 0, n_max = min(1000, n_max), progress = interactive())
```

Comma Delimited Files
`read_csv("file.csv")`
`to_csv(x, file = "file.csv")`
`write_file = "file.csv"`

Semi-colon Delimited Files
`read_csv2("file.csv")`
`write_file = "file.csv"`

With Any Delimiter
`read_delim("file.txt", delin = "")`
`write_file = "file.txt"`

Fixed Width Files
`read_fwf("file.fwf", col_positions = c(1, 3, 5))`
`write_file = "file.fwf"`

Tab Delimited Files
`read_tsv("file")` Also read `table`:
`write_tsv(x, path = "file.tsv")`

USEFUL ARGUMENTS

Example file
`write_csv(a, b, c, n1, 1, 2, 3, n4, 5, "file.csv")`

No header
`read_csv(a, b, c, n1, 1, 2, 3, n4, 5, "file.csv")`

Provide header
`read_csv(a, b, c, n1, 1, 2, 3, n4, 5, "file.csv")`

Missing Values
`read_csv(a, na = c("1", "?"))`

Read Non-Tabular Data

Read a file into a single string
`read_file(file, locale = default_locale())`

Read each line into its own string
`read_lines(file, skip = 0, n_max = -1L, na = character(), locale = default_locale(), progress = interactive())`

Read Apache log file
`read_log(file, col_names = FALSE, n_max = 0, n_max = 1, progress = interactive())`

R Studio

Data Transformation with dplyr :: CHEAT SHEET

dplyr

Data types
`readr` functions guess the types of each column and convert where appropriate (but will NOT convert strings to factors automatically). A message shows the type of each column in the result.

Parsed with column specification:
`## Parsed with column specification:`
`## age = col_integer(),`
`## sex = col_character(),`
`## earn = col_double()`

Each variable is in its own column
`x %>% fly becomes (x, y)`

Each observation, or case, is in its own row

Manipulate Cases
`Row` functions return a subset of rows as a new table.

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

EXTRACT CASES

EXTRACT VARIABLES

Manipulate Variables

EXTRACT VARIABLES

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

pull:
`pull(data, var = 1)` Extract column values as a vector. Choose by name or index.
`pulliris, Sepal.Length`

select:
`select(data, ...)` Extract columns as a table. Also `select_if`.
`selectiris, Sepal.Length, Species`

Use these helpers with select(...):
`e.g. select_iris, stars, with("Sepal.Length")`

contains:
`contains(match, num_range(prefix, range))` e.g. `mpg$ cyl`

ends_with:
`ends_with(match, one_of(...))` e.g. `Species`

matches:
`matches(match, starts_with(match))`

MAKE NEW VARIABLES

vectorized function

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

Logical and boolean operators to use with filter()

Group Cases

ARRANGE CASES

ADD CASES

R Studio

Data Visualization with ggplot2 :: CHEAT SHEET

Basics

Geoms Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

TWO VARIABLES

CONTINUOUS

DISCRETE

ONE VARIABLE

CONTINUOUS

DISCRETE

THREE VARIABLES

MAPS

R Studio

R Markdown Cheat Sheet

learn more at rmarkdown.rstudio.com

1. Workflow R Markdown is a format for writing reproducible, dynamic reports with R. Use it to embed R code and results into slideshows, pdfs, html documents, Word files and more. To make a report:

- i. Open Open a file that uses the .Rmd extension.
- ii. Write Write content with the easy to use R Markdown syntax.
- iii. Embed Embed R code that creates output to include in the report.
- iv. Render Replace R code with its output and transform the report into a slideshow, pdf, html or ms Word file.

2. Open File Start by saving a text file with the extension .Rmd, or open an RStudio Rmd template

3. Markdown Next, write your report in plain text. Use markdown syntax to describe how to format text in the final report.

syntax

YAML

Header 1

Header 2

Header 3

Header 4

Header 5

Header 6

Header 7

Header 8

Header 9

Header 10

Header 11

Header 12

Header 13

Header 14

Header 15

Header 16

Header 17

Header 18

Header 19

Header 20

Header 21

Header 22

Header 23

Header 24

Header 25

Header 26

Header 27

Header 28

Header 29

Header 30

Header 31

Header 32

Header 33

Header 34

Header 35

Header 36

Header 37

Header 38

Header 39

Header 40

Header 41

Header 42

Header 43

Header 44

Header 45

Header 46

Header 47

Header 48

Header 49

Header 50

Header 51

Header 52

Header 53

Header 54

Header 55

Header 56

Header 57

Header 58

Header 59

Header 60

Header 61

Header 62

Header 63

Header 64

Header 65

Header 66

Header 67

Header 68

Header 69

Header 70

Header 71

Header 72

Header 73

Header 74

Header 75

Header 76

Header 77

Header 78

Header 79

Header 80

Header 81

Header 82

Header 83

Header 84

Header 85

Header 86

Header 87

Header 88

Header 89

Header 90

Header 91

Header 92

Header 93

Header 94

Header 95

Header 96

Header 97

Header 98

Header 99

Header 100

Header 101

Header 102

Header 103

Header 104

Header 105

Header 106

Header 107

Header 108

Header 109

Header 110

Header 111

Header 112

Header 113

Header 114

Header 115

Header 116

Header 117

Header 118

Header 119

Header 120

Header 121

Header 122

Header 123

Header 124

Header 125

Header 126

Header 127

Header 128

Header 129

Header 130

Header 131

Header 132

Header 133

Header 134

Header 135

Header 136

Header 137

Header 138

Header 139

Header 140

Header 141

Header 142

Header 143

Header 144

Header 145

Header 146

Header 147

Header 148

Header 149

Header 150

Header 151

Header 152

Header 153

Header 154

Header 155

Header 156

Header 157

Header 158

Header 159

Header 160

Header 161

Header 162

Header 163

Header 164

Header 165

Header 166

Header 167

Header 168

Header 169

Header 170

Header 171

Header 172

Header 173

Header 174

Header 175

Header 176

Header 177

Header 178

Header 179

Header 180

Header 181

Header 182

Header 183

Header 184

Header 185

Header 186

Header 187

Header 188

Header 189

Header 190

Header 191

Header 192

Header 193

Header 194

Header 195

Header 196

Header 197

Header 198

Header 199

Header 200

Header 201

Header 202

Header 203

Header 204

Header 205

Header 206

Header 207

Header 208

Header 209

Header 210

Header 211

Header 212

Header 213

Header 214

Header 215

Header 216

Header 217

Header 218

Header 219

Header 220

Header 221

Header 222

Header 223

Header 224

Header 225

Header 226

Header 227

Header 228

Header 229

Header 230

Header 231

Header 232

Header 233

Header 234

Header 235

Header 236

Header 237

Header 238

Header 239

Header 240

Header 241

Header 242

Header 243

Header 244

Header 245

Header 246

Header 247

Header 248

Header 249

Header 250

Header 251

Header 252

Header 253

Header 254

Header 255

Header 256

Header 257

Header 258

Header 259

Header 260

Header 261

Header 262

Header 263

Header 264

Header 265

Header 266

Header 267

Header 268

Header 269

Header 270

Header 271

Header 272

Header 273

Header 274

Header 275

Header 276

Header 277

Header 278

Header 279

Header 280

Header 281

Header 282

Header 283

Header 284

Header 285

Header 286

Header 287

Header 288

Header 289

Header 290

Header 291

Header 292

Header 293

Header 294

Header 295

Header 296

Header 297

Header 298

Header 299

Header 300

Header 301

Header 302

Header 303

Header 304

Header 305

Header 306

Header 307

Header 308

Header 309

Header 310

Header 311

Header 312

Header 313

Header 314

Header 315

Header 316

Header 317

Header 318

Header 319

Header 320

Header 321

Header 322

Header 323

Header 324

Header 325

Header 326

Header 327

Header 328

Header 329

Header 330

Header 331

Header 332

Header 333

Header 334

Header 335

Header 336

Header 337

Header 338

Header 339

Header 340

Header 341

Header 342

Header 343

Header 344

Header 345

Header 346

Header 347

Header 348

Header 349

Header 350

Header 351

Header 352

Header 353

Header 354

Header 355

Header 356

Header 357

Header 358

Header 359

Header 360

Header 361

Header 362

Header 363

Header 364

Header 365

Header 366

Header 367

Header 368

Header 369

Header 370

Header 371

Header 372

Header 373

Header 374

Header 375

Header 376

Header 377

Header 378

Header 379

Header 380

Header 381

Header 382

Header 383

Header 384

Header 385

Header 386

Header 387

Header 388

Header 389

Header 390

Header 391

Header 392

Header 393

Header 394

Header 395

Header 396

Header 397

Header 398

Header 399

Header 400

Header 401

Header 402

Header 403

Header 404

Header 405

Header 406

Header 407

Header 408

Header 409

Header 410

Header 411

Header 412

Header 413

Header 414

Header 415

Header 416

Header 417

Header 418

Header 419

Header 420

Header 421

Header 422

Header 423

Header 424

Header 425

Header 426

Header 427

Header 428

Header 429

Header 430

Header 431

Header 432

Header 433

Header 434

Header 435

Header 436

Header 437

Header 438

Header 439

Header 440

Header 441

Header

Resources: Stackoverflow

stack overflow Products Customers Use cases Log in Sign up

Home PUBLIC Stack Overflow Tags Users Jobs TEAMS What's this? Join Private Q&A

Search Results

Results for removing legend title ggplot

removing legend title ggplot

20 results

99 votes [Q: remove legend title in ggplot](#)
I'm trying to **remove** the **title** of a **legend** in ggplot2: df <- data.frame(g = rep(letters[1:2], 5), x = mnorm(10), y = mnorm(10)) library(ggplot2) **ggplot**(df, aes(x, y, colour=g ...
r ggplot2 asked Feb 8 '13 by [smillig](#)

0 votes [Q: How to change the position of plotly figures using ggplotly](#)
I am trying to **remove** the **legend title** using ggplotly without success. I'm sure there is an easy fix, but I cannot find the documentation for it - and **removing** the **legend title** (or changing the ...)) + geom_boxplot() + theme(legend.title=element_blank()) a # No **Legend Title** # plotly puts back the **legend title** ggplotly(a) Any ideas how to change / **remove** the **title** of the graph? Should it be done using ggplotly or **ggplot**? ...
r ggplot2 plotly asked Dec 28 '15 by [Nick](#)

1 vote [Q: Remove legend in ggplot in Python](#)
The **title** says it all: Is there any way to **remove** the **legend** in **ggplot** in Python? I tried to google but could only find the solution for R ...
python ggplot2 visualization python-ggplot asked Jul 25 '18 by [Giang Do](#)

0 votes [Q: ggplot legend / label change with various guides](#)
already used to **remove** the titles and also **remove legend** for a 3rd aes, but I do not manage to do it. Do you have a solution? Here is my code : p <- **ggplot**(data, aes(x = Nb)) p + geom_ribbon(aes(ymin ... I am trying to

SIEMENS **Siemens AG** Singapore Electronics Public 10k+ people

Senior Data Scientist machine-learning sql

HTML5 Code King html5 css3

Business Architect enterprise

[View all 14 job openings!](#)

Hot Network Questions

When does Haskell complain about incorrect typing in functions?

How much were the LMs maneuvered to their landing points?

Why didn't Britain or any other European power

Resources

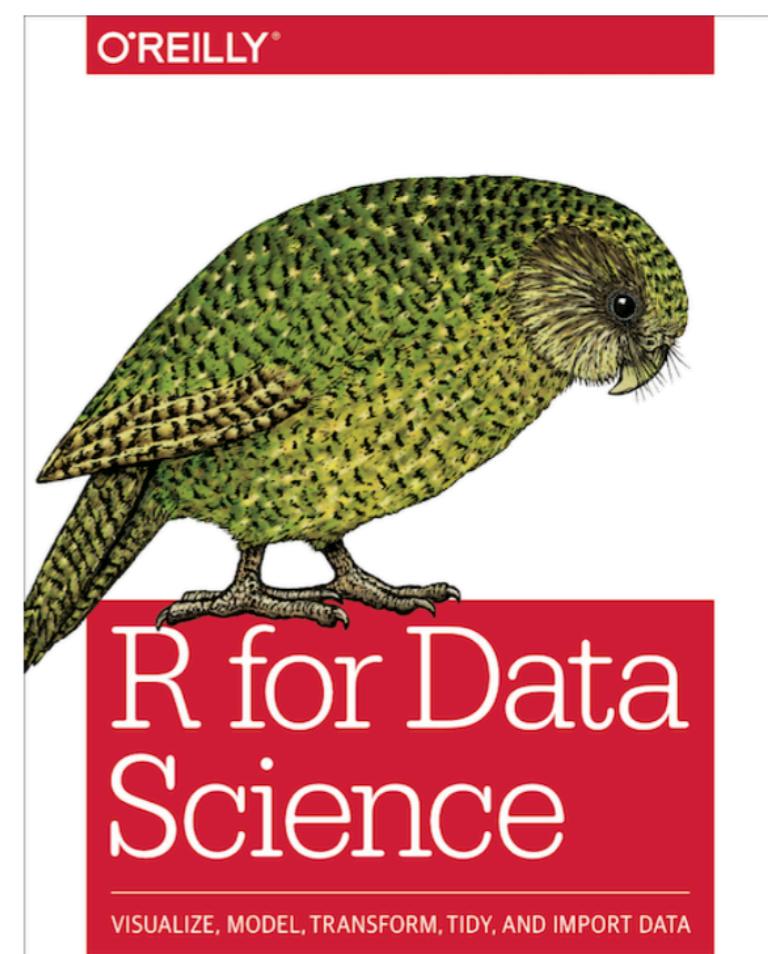
R for Data Science

Garrett Grolemund

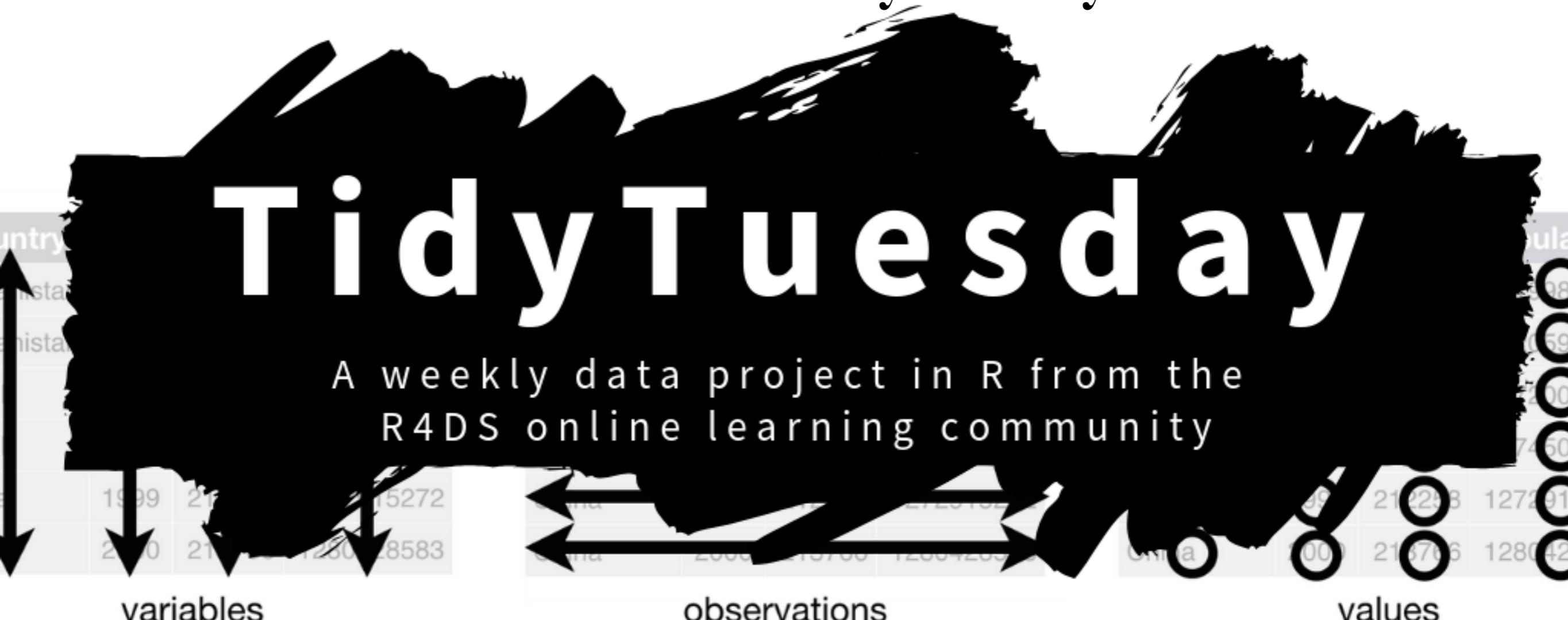
Hadley Wickham

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: #tidytuesday



[rfordatascience / tidytuesday](#)

[Unwatch](#) 258

[Unstar](#) 960

[Fork](#) 457

[Code](#)

Issues 48

Pull requests 0

Projects 0

Wiki

Security

Insights

Official repo for the #tidytuesday project

380 commits

2 branches

11 releases

8 contributors

MIT

Branch: master ▾

New pull request

Create new file

Upload files

Find File

Clone or download ▾

Acknowledgements

Funding

National Institute of Aging (NIA)
National Institute of Diabetes and
Digestive and Kidney Diseases
(NIDDK)

Glenn Foundation (Glenn Award)

Hirshey Lab (Current)

Alex Adams, Kristin Anderson, Jon
Haldeman, E. Keith Keenan, Shreyas
Kulkarni, Zhihong Lin, Allie Mills,
Darren Stuart, Alec Trub, Derek
Zachman

Hirshey Lab (Alum)

Dhaval Bhatt, Frank Huynh, Michelle
Green, Kathleen Hershberger, Angel
Martin, Eoin McDonnell, Brett
Peterson, Gregg Wagner, Dongning
Wang

Inspiration, ideas, packages, code

Tidyverse, datasciencebox.org (Mine
Çetinkaya-Rundel), Modern Dive,

