

Week 01

Introduction

Data Wrangling

API209: Summer Math Camp

Rony Rodrigo Maximiliano Rodriguez-Ramirez

rrodriguezramirez@g.harvard.edu

Harvard University

August 15, 2024

Welcome

Plan for today

1. Getting to know each other
2. The layout of this summer camp (just the R part)
3. Why R, RStudio, Positron?
4. Data manipulation in R

Who am I?

My role

What should you expect from me?

At the end of this summer camp?

Course assistants for Math Camp

Shan



Ayush



Sara



The layout

The layout

What are we going during math camp (R Part)?

8 sessions over the next weeks:

1. 4 Lessons (2 hours)

- I will discuss about coding, strategies, and implementation

2. 4 Labs (1.5 hours)

- It will be a hands-on session. I will provide you with exercises and we will solve them together.

3. Optional: Office hours

The layout

There is a website for this summer camp:

WEBSITE

It is not up-to-date; but every week, you will have the materials for that respective week, i.e., you should have already your lab for tomorrow.

Coding

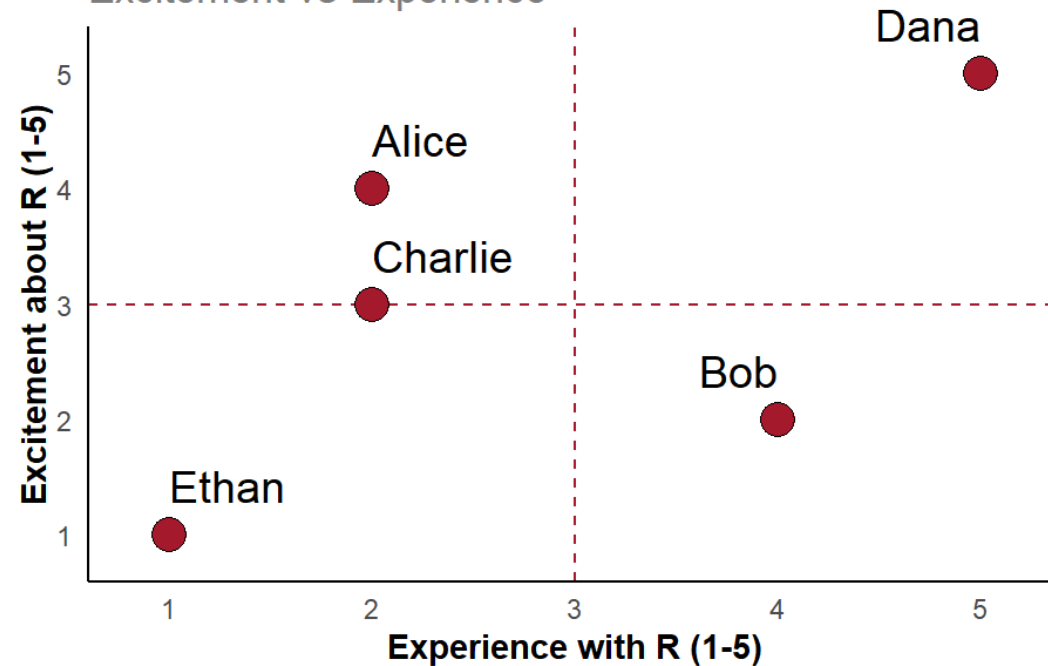
Coding is a *Skill*

It is most fun to practice a skill with people you know.

Discussion Activity: Archetypes

Student Archetypes:

Excitement vs Experience



Credit: @Dom | Former TF.

- **Intros:** Name, where you're from, favorite midday snack or superhero.
- **Experience:** With statistics, programming, and/or R? (Yes/No)
- **Which quadrant describes you?**

Classroom Norms

Class is a collective enterprise!

- Allow everyone the chance to speak.
- Be mindful of thoughts and actions.
- Understand differing levels of knowledge and experience.
- Help others in your group!
- Don't be afraid to ask questions!
- Respect others' opinions and suggestions.
- Try questions on your own first, then come together.
- Take breaks and have fun!

Using AI for Code Troubleshooting

- **AI assistants like ChatGPT and Claude can help debug code**
- **Steps to use AI for troubleshooting:**
 1. Paste your code snippet and error message
 2. Ask for help identifying and fixing the issue
 3. Review AI's suggestions critically
 4. Test proposed solutions in your environment
- **Benefits:**
 - Quick identification of common errors
 - Explanations of underlying issues
 - Suggestions for best practices

Why R, RStudio, Positron?

Why R, RStudio, and Positron?

Why R?

- Open-source and free.
- Extensive ecosystem for statistical analysis.
- Wide range of packages for data manipulation and visualization.
- Active and supportive community.

RStudio and Positron: The IDEs for R

Why RStudio?

- Integrated development environment (IDE) that simplifies coding in R.
- Built-in tools for code development, debugging, and collaboration.
- Seamless integration with RMarkdown for dynamic report generation.
- Powerful tools for data visualization and manipulation.

Why Positron?

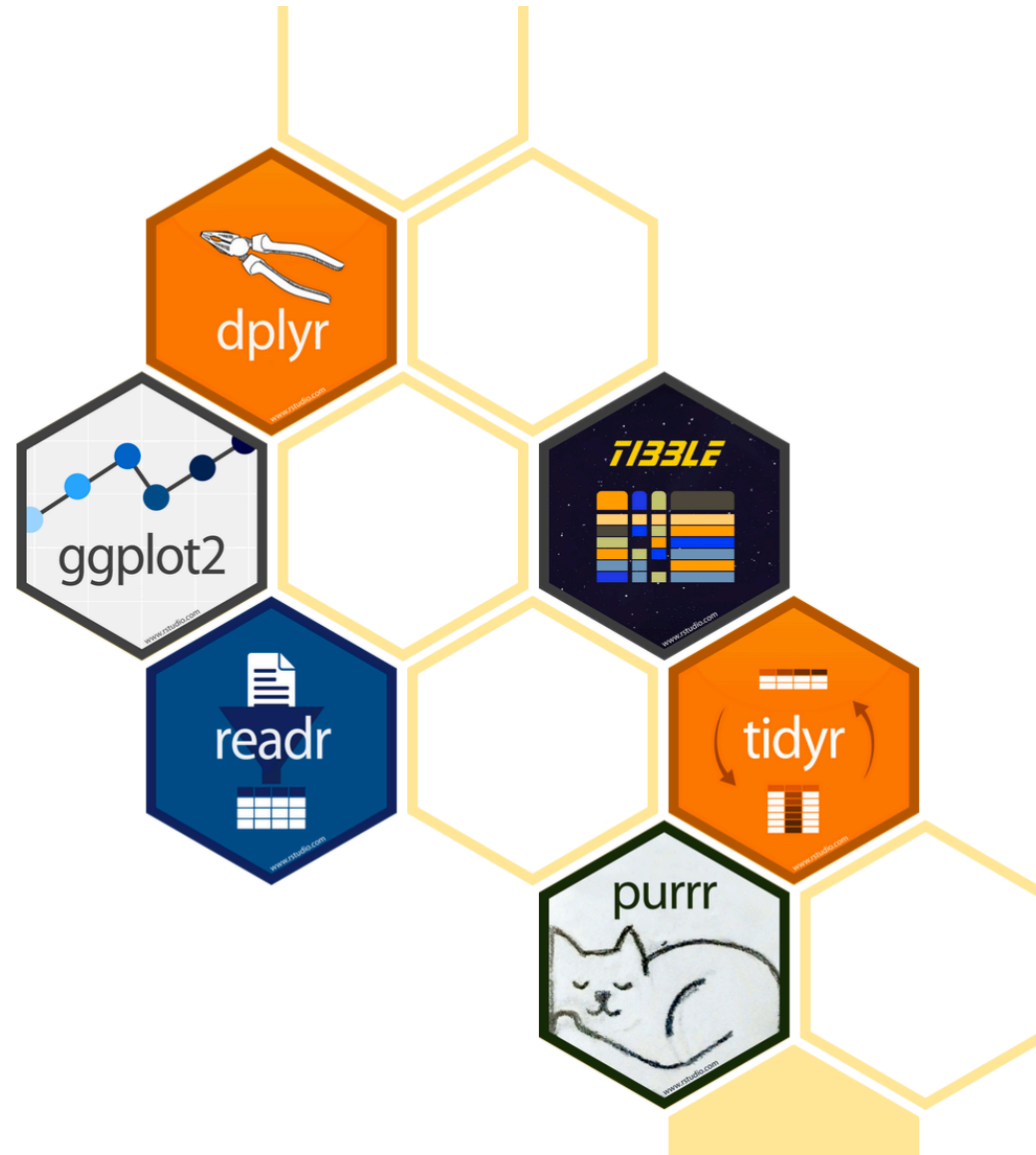
- New, modern IDE designed to enhance the R programming experience.
- Sleeker interface with enhanced performance and features.
- Supports the latest R packages and workflows.
- Focuses on integrating modern development tools and practices.

Which IDE Should You Use?

RStudio vs. Positron

- **RStudio** is well-established with a large user base and extensive support.
- **Positron** offers cutting-edge features for those looking to adopt the latest tools.
- Consider trying both to see which fits your workflow best.
- During the pre-summer assignment, we used **Posit Cloud**.
 - For those who haven't installed, either RStudio nor Positron in your computer, there is a Posit Cloud Project [here](#)

The tidyverse

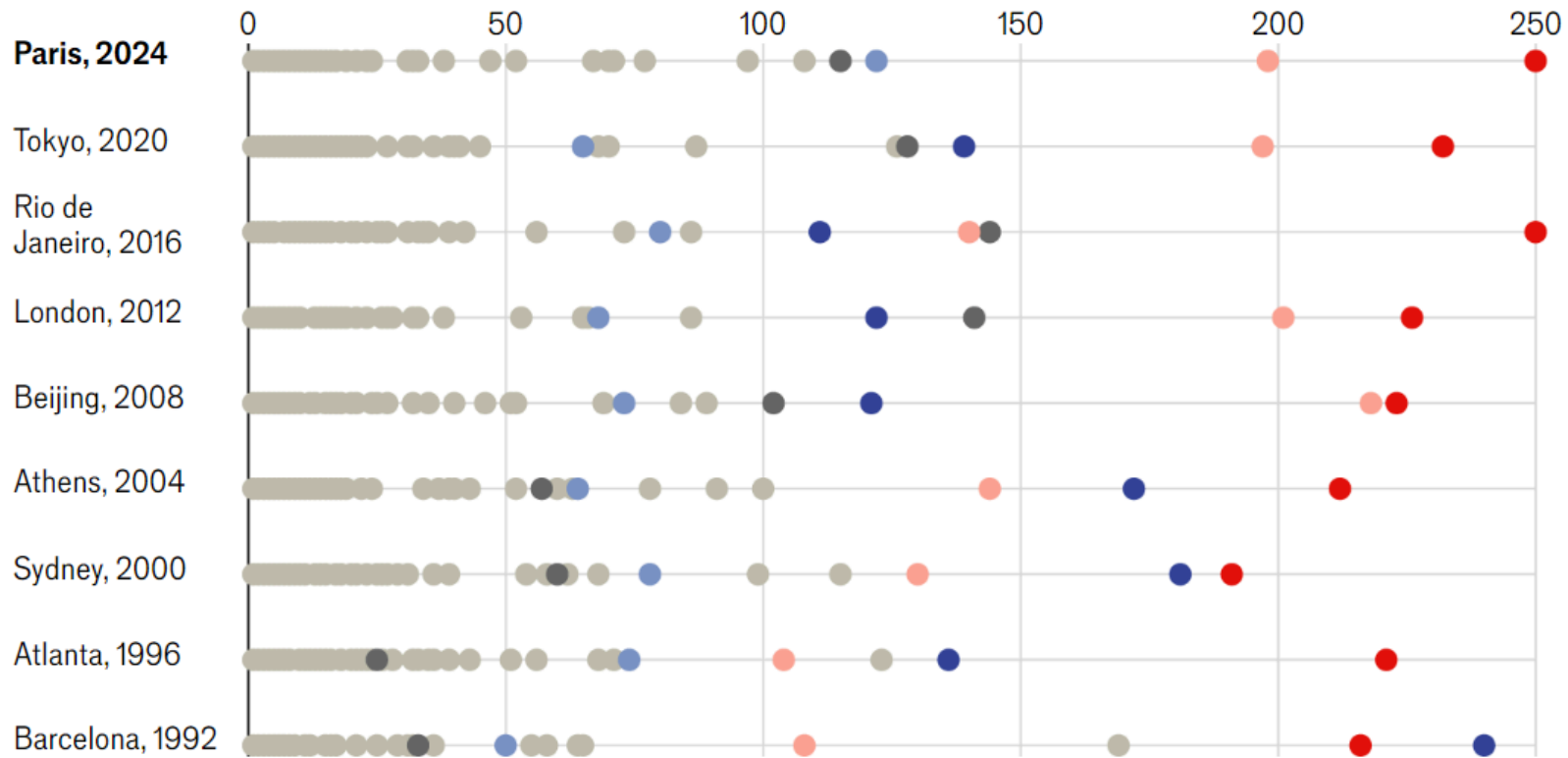


Original Plot

American exceptionalism

Summer Olympics, medal points*

● United States ● China ● France ● Britain ● Russia[†] ● Other



*Gold=3 points, Silver=2, Bronze=1 †Includes Unified Team in 1992 and Russian Olympic Committee in 2020

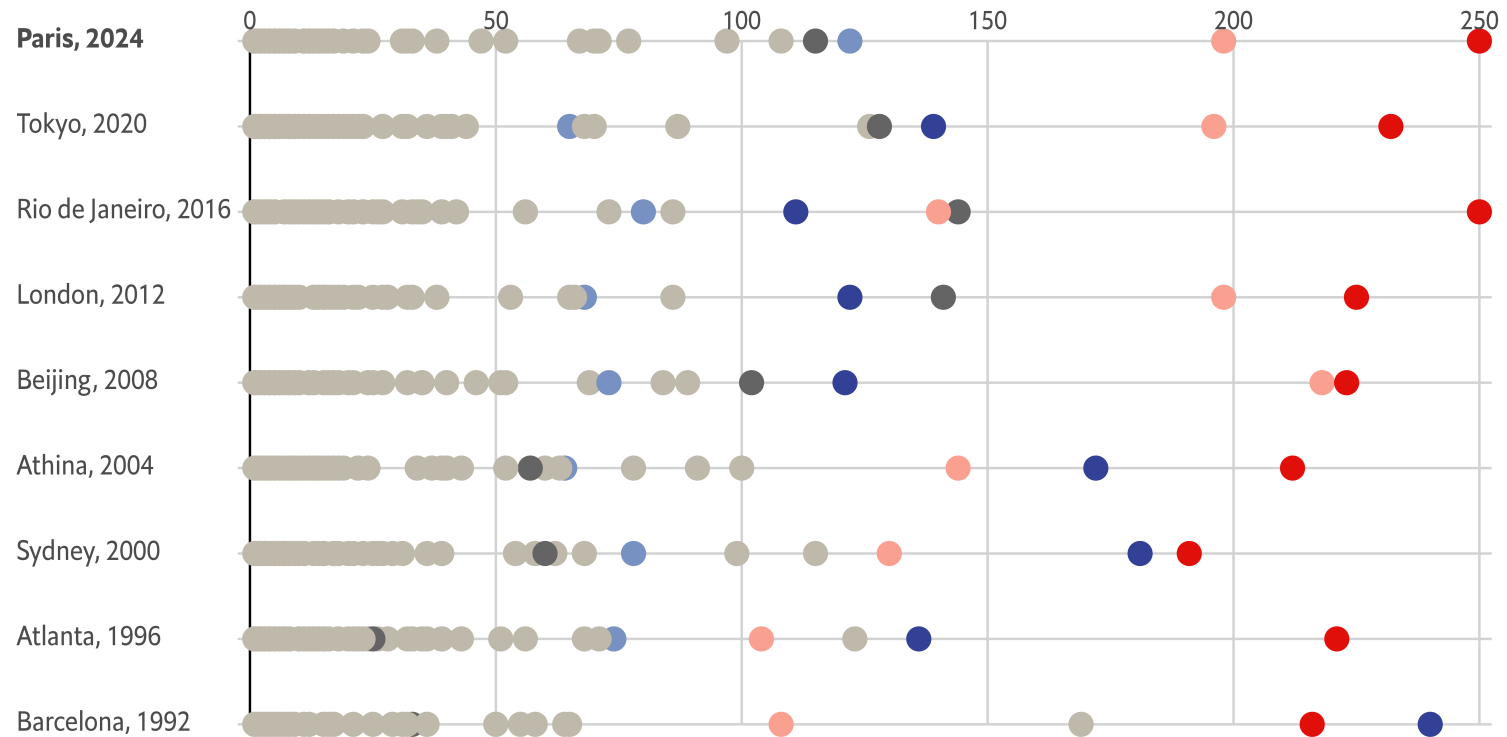
Sources: International Olympic Committee; Olympedia

Using ggplot

American exceptionalism

Summer Olympics, medal points*

● United States ● China ● France ● Great Britain ● Russian Federation ● Other



*Gold=3 points, Silver=2, Bronze=1 †Includes Unified Team in 1992 and Russian Olympic Committee in 2020
Sources: International Olympic Committee; Olympedia

Data wrangling

Data Manipulation vs Data Wrangling

Data Manipulation

- Focused on changing data structure
- Often uses specific functions or methods
- Usually works with structured data
- Examples: sorting, filtering, aggregating
- Typically a part of data wrangling
- More straightforward, less time-consuming

Data Wrangling

- Broader process including cleaning and transformation
- May involve multiple tools and techniques
- Handles both structured and unstructured data
- Examples: merging datasets, handling missing values, format conversion
- Encompasses the entire data preparation process
- Can be complex and time-intensive

Advanced Data Wrangling

Mastering Data Manipulation in R

- **Advanced Filtering and Selection:**
 - Use of conditional filtering and dynamic column selection.
- **Complex Mutate Operations:**
 - Creating conditional columns, using lag and lead.
- **Data Reshaping:**
 - Pivoting data, advanced grouping.
- **Efficient Data Handling:**
 - Joining datasets, parallel processing.

Recap: The Tidyverse

The tidyverse is a collection of R packages designed for data science.

They share an underlying design philosophy, grammar, and data structures.

Core Packages:

- `ggplot2` - Data visualization
- `dplyr` - Data manipulation
- `tidyr` - Data tidying
- `readr` - Data import
- `purrr` - Functional programming
- `tibble` - Modern data frames
- `stringr` - String manipulation
- `forcats` - Categorical data

dplyr: Key Functions

Commonly Used Functions:

- `filter()` - Subset rows based on conditions
- `select()` - Choose columns by names
- `mutate()` - Create new columns or modify existing ones
- `arrange()` - Reorder rows
- `summarize()` - Aggregate data
- `group_by()` - Group data for summary operations

dplyr: Example

```
1 library(dplyr)
2
3 # Filter and select
4 filtered_data ← starwars ▷
5   filter(height > 180) ▷
6   select(name, height, hair_color)
7
8 filtered_data
```

```
# A tibble: 39 × 3
  name          height hair_color
  <chr>         <int> <chr>
1 Darth Vader   202 none
2 Biggs Darklighter 183 black
3 Obi-Wan Kenobi 182 auburn, white
4 Anakin Skywalker 188 blond
5 Chewbacca     228 brown
6 Boba Fett     183 black
7 IG-88        200 none
8 Bossk        190 none
9 Qui-Gon Jinn 193 brown
10 Nute Gunray  191 none
# i 29 more rows
```

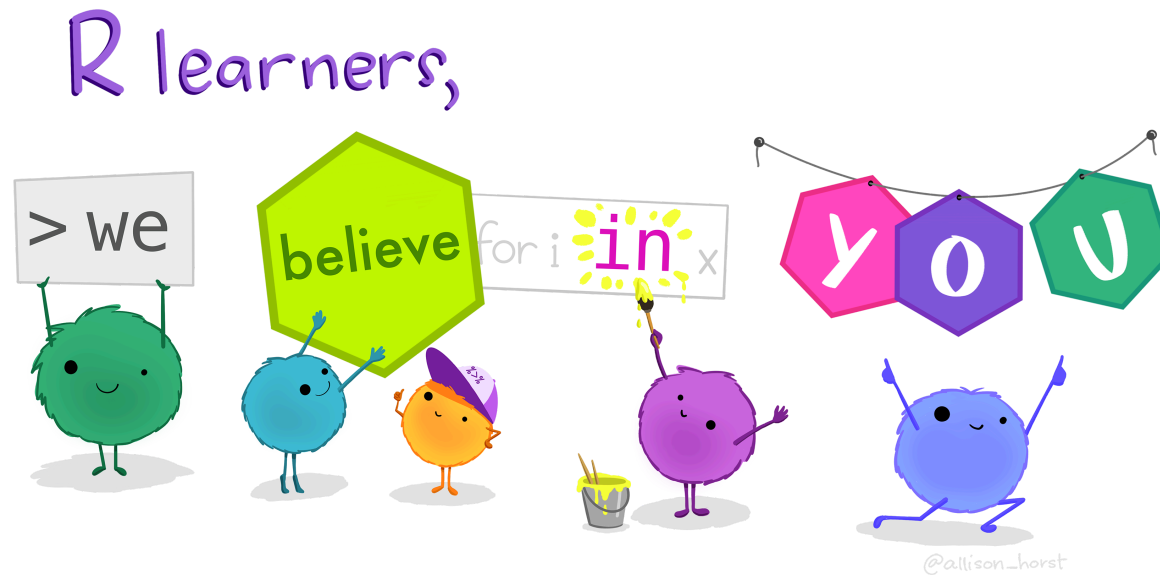
dplyr: Example

```
1 # Mutate
2 starwars ▷
3   mutate(
4     tatooine = ifelse( #<<
5       homeworld = "Tatooine", #<<
6       "Tatooine", #<<
7       "Others" #<<
8     ) #<<
9   ) ▷
10  group_by(tatooine) ▷
11  summarize(
12    mean = mean(height)
13  )
```

```
# A tibble: 3 × 2
  tatooine mean
  <chr>    <dbl>
1 Others  177.
2 Tatooine 170.
3 <NA>    NA
```

Are we good here?

As of now, you should have the tools to understand the last code. More resources are available in our website. Now, it's time to make some mistakes!



Artwork by @allison_horst

Takeaways

Sucking

Originally from: Dr. Andrew Heiss

“There is no way of knowing nothing about a subject to knowing something about a subject without going through a period of much frustration and suckiness.”

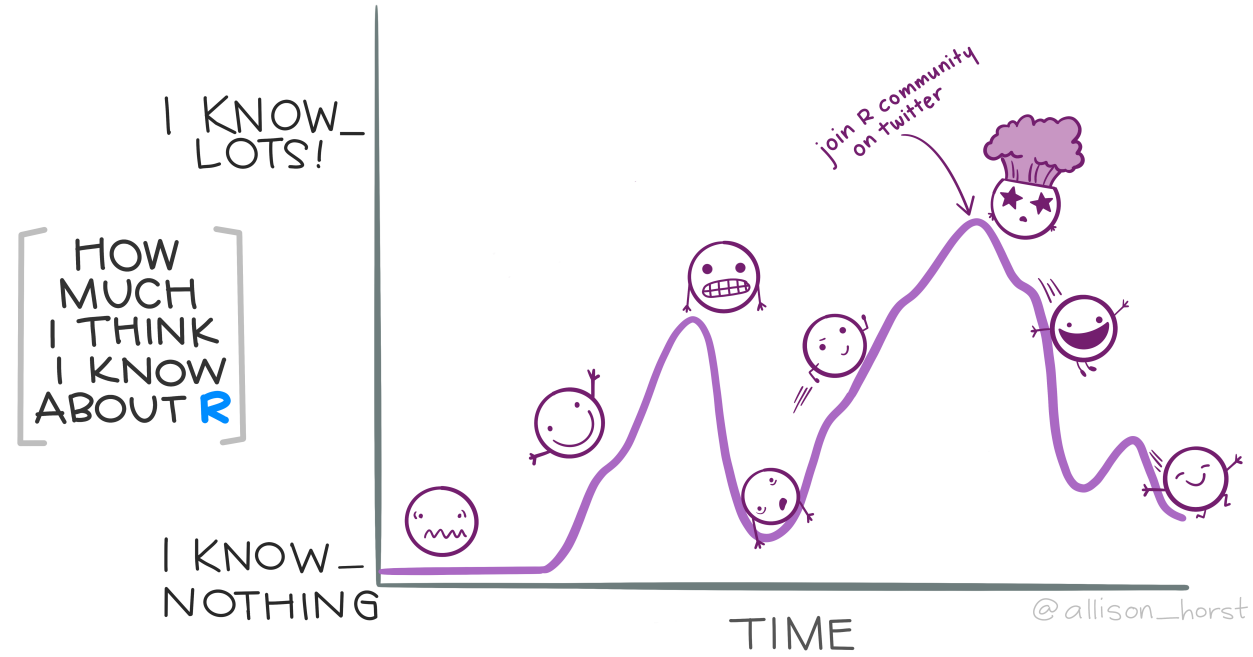
“Push through. You’ll suck less.”

Hadley Wickham, author of {ggplot2}

Sucking



Sucking



Artwork by @allison_horst