Review of Technology

STA6349: Applied Bayesian Analysis

Spring 2025

Introduction

- Welcome to Applied Bayesian Analysis - Spring 2025!
  - → Canvas set up
  - → Syllabus
  - → Discord
  - → R/RStudio
  - → Quarto
  - → GitHub
  - → Resources

Introduction

- General topics:
  - → Probability rules and distributions
  - → Bayes Theorem
  - → Prior distributions
  - → Posterior distributions
  - → Conjugate families
  - → Beta-Binomial, Normal-Normal, and Gamma-Poisson models
  - → Posterior simulation
  - → Posterior inference
  - → Linear regression
- **This is an applied class.**

GitHub

- Our course lectures and labs are posted on GitHub.
- Please bookmark the repository: GitHub for STA6349.
- You will want to look at my .qmd files for formatting / $\LaTeX$ purposes.
- Feel free to poke around my GitHub to see materials for other classes.

R/RStudio

- We will be using R in this course.
    - I use the RStudio IDE, however, if you would like to use another IDE, that is fine.
- It is okay if you have not used R before!
- Full disclosure: I am a **biostatistician** first, **programmer** second.
    - This means that I focus on the application of statistical methods and not on “understanding” the innerworkings of R.
        - R is a *tool* that we use, like how SAS, JMP, Stata, SPSS, Excel, etc. are tools.
    - Sometimes my code is not elegant/efficient, and that’s okay! Because our focus is on the application of methods, we are interested in the code working.
    - I have learned *so much* from my students since implementing R in the classroom.
        - Do not be afraid to teach me new things!
- **This is an applied class.**

R/RStudio

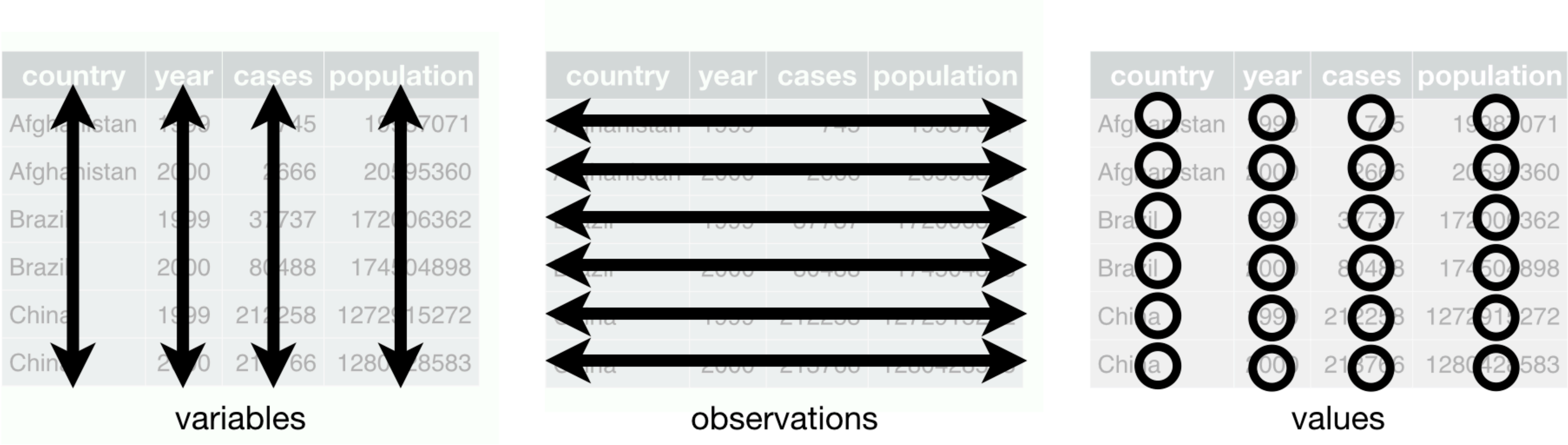
- You can install R and RStudio on your computer for **free**.
  - → R from CRAN
  - → RStudio from Posit
- Alternative to installing: RStudio Server hosted by UWF HMCSE
- **Do not use Citrix.**
- I encourage you to install R on your own machine if you are able.
  - → In the "real world," you will not have access to the server.
  - → Installing on your own machine will help your future self troubleshoot issues.

Tidy Data

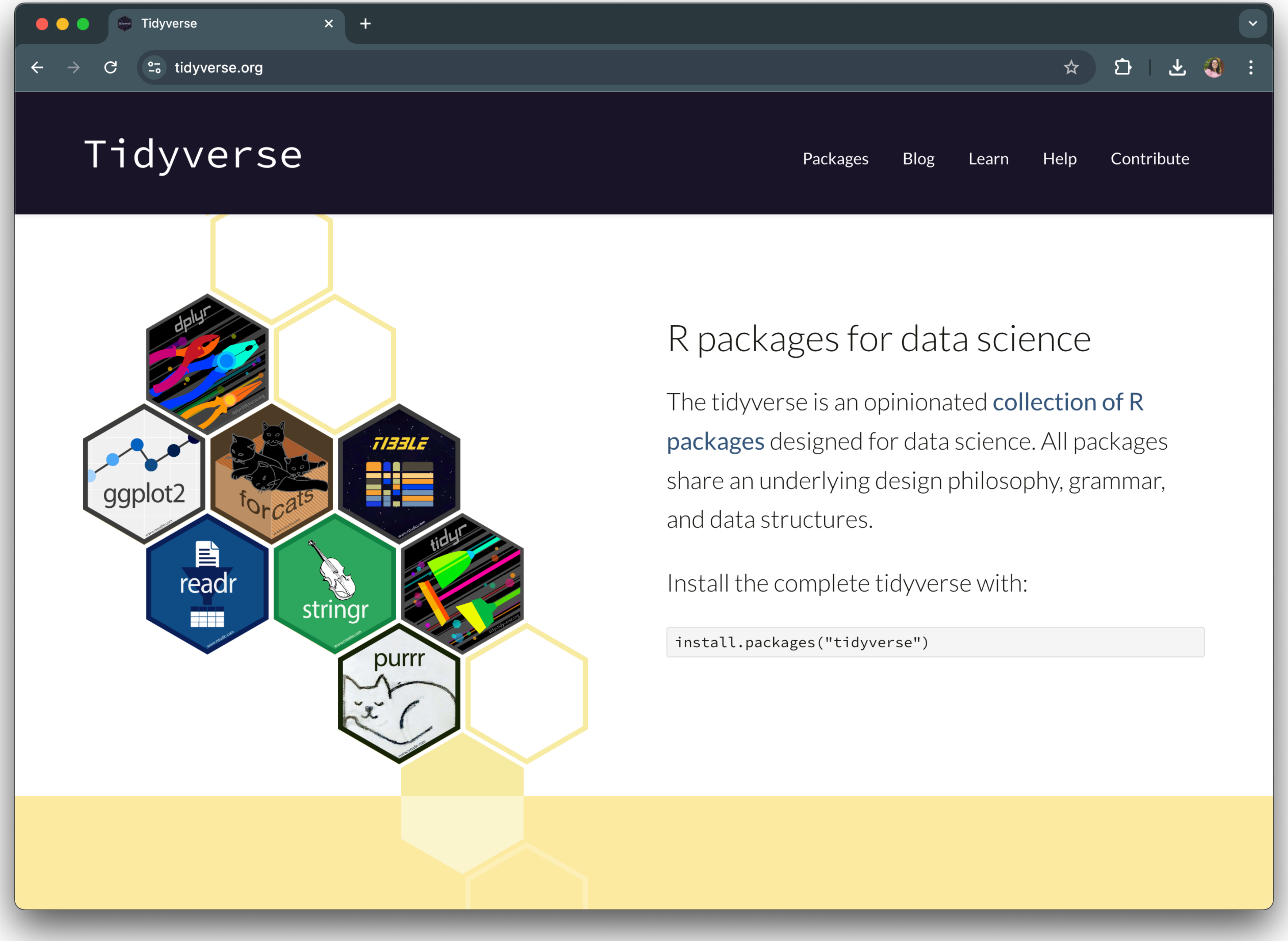
Journal article: *Tidy Data* by Wickham (2014, *Journal of Statistical Software*)

Book chapter: *Data Tidying* by Wickham, Çetinkaya-Rundel, and Grolemund

- There are three interrelated rules that make a dataset tidy:
    1. Each variable is a column; each column is a variable.
    2. Each observation is a row; each row is an observation.
    3. Each value is a cell; each cell is a single value.



Tidyverse



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")
```

Tidyverse

- `tibble` for modern data frames.
- `readr` and `haven` for data import.
  - → `readr` is pulled in with `tidyverse`
  - → `haven` needs to be called in on its own
- `tidyr` for data tidying.
- `dplyr` for data manipulation.
- `ggplot2` for data visualization.
- It is not possible for me to teach you everything you will ever need to know about programming in R.
  - → Good resource for `tidyverse`: data science in a box

Tidyverse

- A major advantage of using `tidyverse` is the common “language” between the functions.
- Another advantage: the pipe operator, `%>%`.
  - Yes, there is a pipe operator now included in base R. No, I do not use it.
    - Here is a discussion of similarities and differences from Hadley himself.
  - By default, `%>%` deposits everything that came before into the first argument of the next function.
    - If we want to insert it elsewhere, we can indicate that with a “.” in the function.

```r
lm(body_mass_g ~ flipper_length_mm, data = penguins)

penguins %>% lm(body_mass_g ~ flipper_length_mm, data = .)
```

Tidyverse

- If we try to use a function before calling its package in, we will see an error.

```r
sw <- tibble(starwars) %>% filter(mass < 100)
```

```
Error in tibble(starwars) %>% filter(mass < 100): could not find function "%>%"
```

- We are good to go after calling in `tidyverse`.

```r
library(tidyverse)
sw <- tibble(starwars) %>% filter(mass < 100)
head(sw, n=3)
```

name	height	mass	hair_color	skin_color	eye_color	birth_year	sex	gender
<chr>	<int>	<dbl>	<chr>	<chr>	<chr>	<dbl>	<chr>	<chr>
Luke Skywalker	172	77	blond	fair	blue	19	male	masculine
C-3PO	167	75	NA	gold	yellow	112	none	masculine
R2-D2	96	32	NA	white, blue	red	33	none	masculine

3 rows | 1-9 of 14 columns

Importing Data

- Let’s import data from the Jackson Heart Study.

```r
jhs_csv <- read_csv("/path/to/folder/analysislong.csv")
head(jhs_csv)
```

subjid	visit	VisitDate	DaysFromV1	YearsFromV1	ARIC	recruit	ageIneligible	FastHours	age
<dbl>	<dbl>	<chr>	<dbl>	<dbl>	<chr>	<chr>	<chr>	<dbl>	<dbl>
2054	1	06/30/2003	0	0	JHS-Only	Random	No	16.47	63.4
2054	2	07/17/2007	1478	4	JHS-Only	Random	No	16.87	67.5
2054	3	07/17/2010	2574	7	JHS-Only	Random	No	15.53	70.5
2013	1	09/30/2003	0	0	JHS-Only	Random	No	15.33	56.0
2013	2	07/04/2008	1739	5	JHS-Only	Random	No	14.02	60.8
2013	3	12/26/2010	2644	7	JHS-Only	Random	No	2.33	63.3

6 rows | 1-10 of 204 columns

Importing Data

- Be comfortable with Googling for help with code to import data.
- As a collaborative statistician, I have received the following file types:
  - .sas7bdat
  - .sav
  - .dat
  - .csv
  - .xls
  - .xlsx
  - .txt
  - Google Sheet
  - hand written

Importing Data

- There have been times where I have received data as a .xlsx, but I can’t get it to import properly.
  - Usually, the issue is that there is a character variable with too much text.
  - Sometimes, it’s that the variable type changes mid-dataset.
    - i.e., both a number and a character stored in the same vector.
- Sometimes the solution is saving it as a different file type (I default to .csv).
- Get comfortable Googling error messages.
  - I am still consulting Dr. Google for assistance on a daily basis!
- Try not to do any data management within the original file type!
  - We want to be able to retrace our steps.
  - Reproducible research!

Data Manipulation

- Functions:
  - → `select()`: Selecting columns.
  - → `filter()`: Filtering the observations.
  - → `mutate()`: Adding or transforming columns.
  - → `summarise()`: Summarizing data.
  - → `group_by()`: Grouping data for summary operations.
  - → `%>%`: Pipelines.

Data Manipulation

- `select()`: Selecting columns.

```
jhs_csv %>%
  select(subjid, visit, age, sex) %>%
  head(n=4)
```

subjid <dbl>	visit <dbl>	age <dbl>	sex <chr>
2054	1	63.4	Male
2054	2	67.5	Male
2054	3	70.5	Male
2013	1	56.0	Female

4 rows

Data Manipulation

- `filter()`: Filtering rows.

```r
jhs_csv %>%
  filter(visit == 1) %>%
  head(n=3)
```

subjid	visit	VisitDate	DaysFromV1	YearsFromV1	ARIC	recruit	ageIneligible	FastHours	age
<dbl>	<dbl>	<chr>	<dbl>	<dbl>	<chr>	<chr>	<chr>	<dbl>	<dbl>
2054	1	06/30/2003	0	0	JHS-Only	Random	No	16.47	63.4
2013	1	09/30/2003	0	0	JHS-Only	Random	No	15.33	56.0
455	1	01/03/2004	0	0	JHS-Only	Volunteer	No	15.17	56.5

3 rows | 1-10 of 204 columns

Data Manipulation

- `mutate()`: Adding or transforming columns.

```r
jhs_csv %>%
  filter(visit == 1) %>%
  select(subjid, sex) %>%
  mutate(male = if_else(sex == "Male", 1, 0)) %>%
  head(n=3)
```

subjid	sex	male
<dbl>	<chr>	<dbl>
2054	Male	1
2013	Female	0
455	Female	0

3 rows

Data Manipulation

- `summarise()`: Summarizing data.

```r
jhs_csv %>%
  filter(visit == 1) %>%
  summarize(n = n(),
            mean_BMI = round(mean(BMI, na.rm = TRUE),2),
            sd_BMI = round(sd(BMI, na.rm = TRUE),2),
            n_female = sum(sex == "Female", na.rm = TRUE),
            pct_female = round(sum(sex == "Female", na.rm = TRUE)*100/n(),2))
```

n	mean_BMI	sd_BMI	n_female	pct_female
<int>	<dbl>	<dbl>	<int>	<dbl>
2653	31.86	6.97	1673	63.06

1 row

Data Manipulation

- `group_by()`: Grouping data for summary operations.

```r
jhs_csv %>%
  filter(visit == 1) %>%
  group_by(HTN) %>%
  summarize(n = n(),
            mean_BMI = round(mean(BMI, na.rm = TRUE),2),
            sd_BMI = round(sd(BMI, na.rm = TRUE),2),
            n_female = sum(sex == "Female", na.rm = TRUE),
            pct_female = round(sum(sex == "Female", na.rm = TRUE)*100/n(),2))
```

HTN	n	mean_BMI	sd_BMI	n_female	pct_female
<chr>	<int>	<dbl>	<dbl>	<int>	<dbl>
No	1237	30.76	6.84	742	59.98
Yes	1416	32.81	6.94	931	65.75

2 rows

Wrap Up

- Today we have gently introduced data management in R.
- I do not expect you to become an expert R programmer, but the more you practice, the easier it becomes.
- Today’s activity: Assignment 0