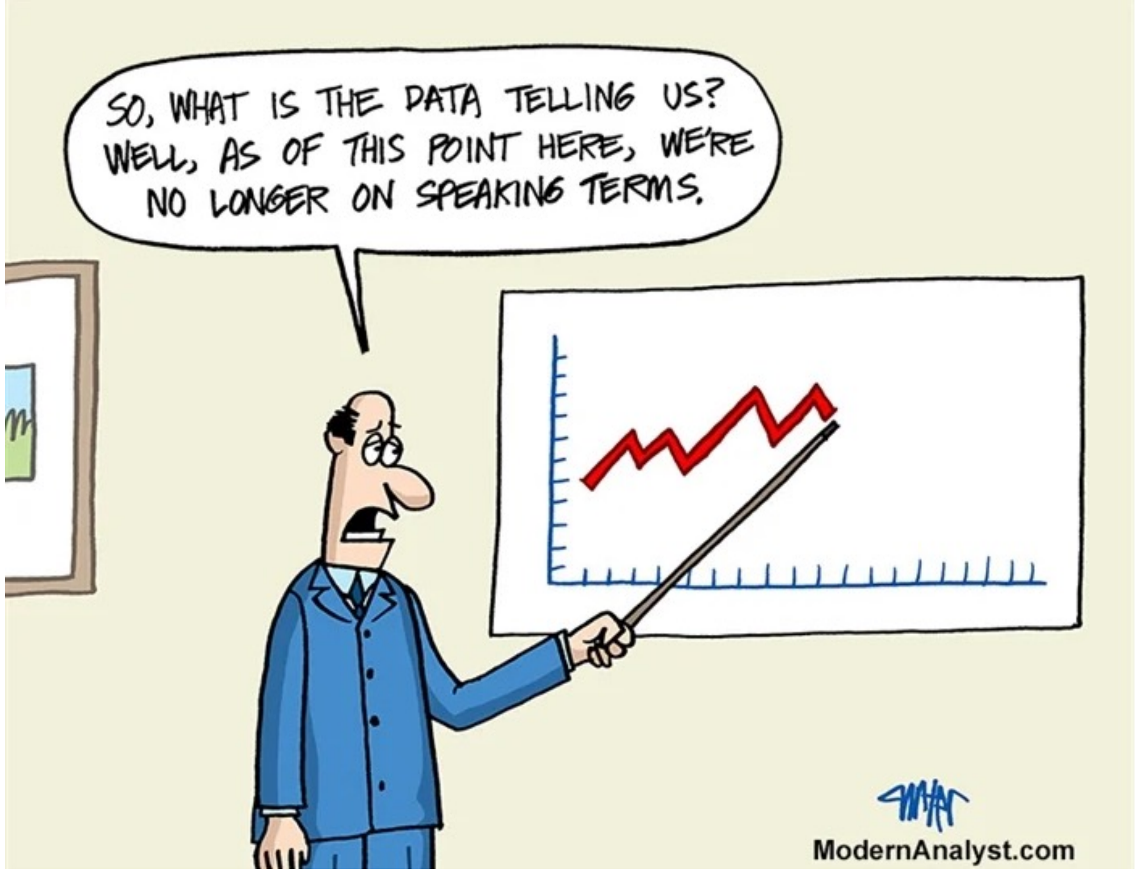
Data cleaning, ethics, and conclusions



Overview

Data cleaning/wrangling continued

- Pivoting data with tidyr
- Joining data frames

Brief mention: Shiny for interactive web applications

Ethics

Conclusions

Announcement

There will be no late penalty for the final project that are turned in before the end of reading period

I still highly recommend you turn it in at the original deadline so that you have plenty of time to study for the final exam

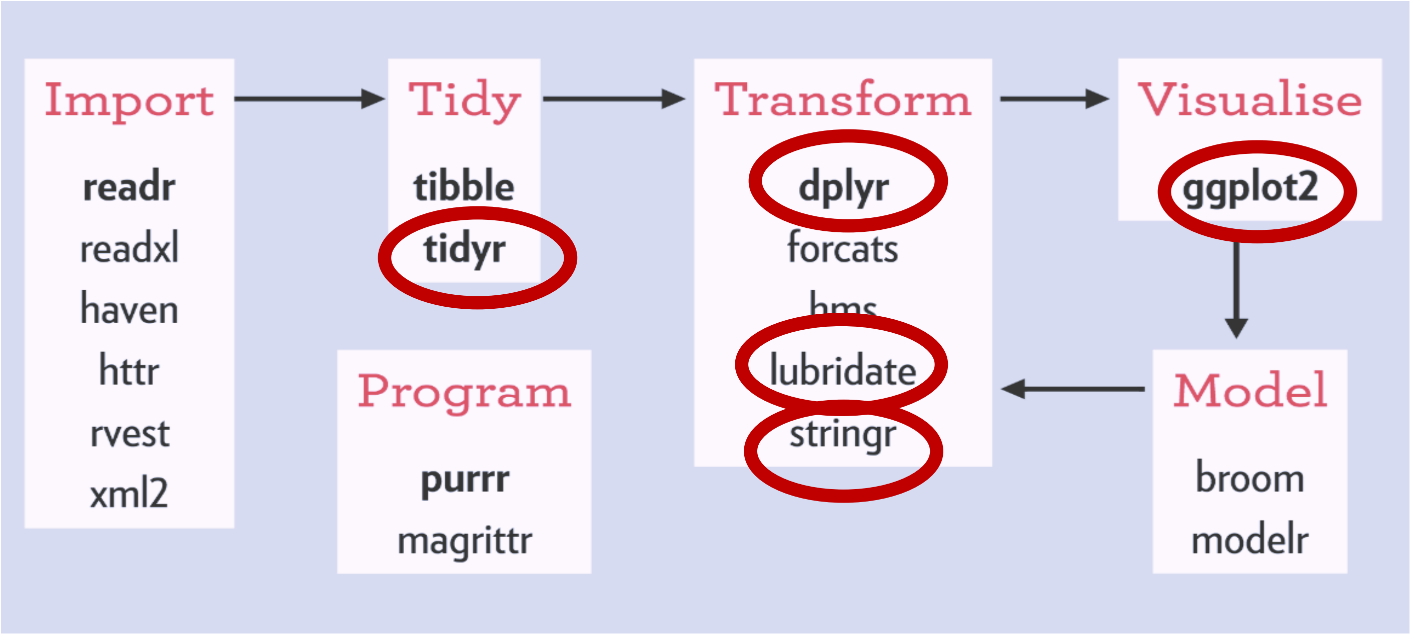
- The final exam is weighted significantly more than the final project

Continuation of tidyverse packages useful for your projects

Tidyverse packages useful for your projects

The tidyverse packages share a common design philosophy

- Most written by Hadley Wickham



The posit cheat sheets can be very useful

tidyr for pivoting data

Wide vs. Long data

Plotting data using ggplot requires that data is in the right format

- i.e., requires data transformations

Often this involves converting data from a **wide format** to **long format**

Wide data

Person	Age	Height
Bob	32	72
Alice	24	65
Steve	64	70

Long data

Person	name	value
Bob	Age	32
Bob	Height	72
Alice	Age	24
Alice	Height	65
Steve	Age	64
Steve	Height	70

library(tidyr)

tidyr::pivot_longer()

pivot_longer(df, cols) converts data from **wide** to **long**

- Takes multiple columns and converts them into two columns: name and value
  - Column names become categorical variable levels of a new variable called **name**
  - The data in rows become entries in a variable called **value**

Wide data

Person	Age	Height
Bob	32	72
Alice	24	65
Steve	64	70

→

Long data

Person	name	value
Bob	Age	32
Bob	Height	72
Alice	Age	24
Alice	Height	65
Steve	Age	64
Steve	Height	70

tidyr::pivot_wider()

pivot_wider(df, names_from, values_from) converts data from long to wide

- Turns the levels of categorical data into columns in a data frame

Narrow data

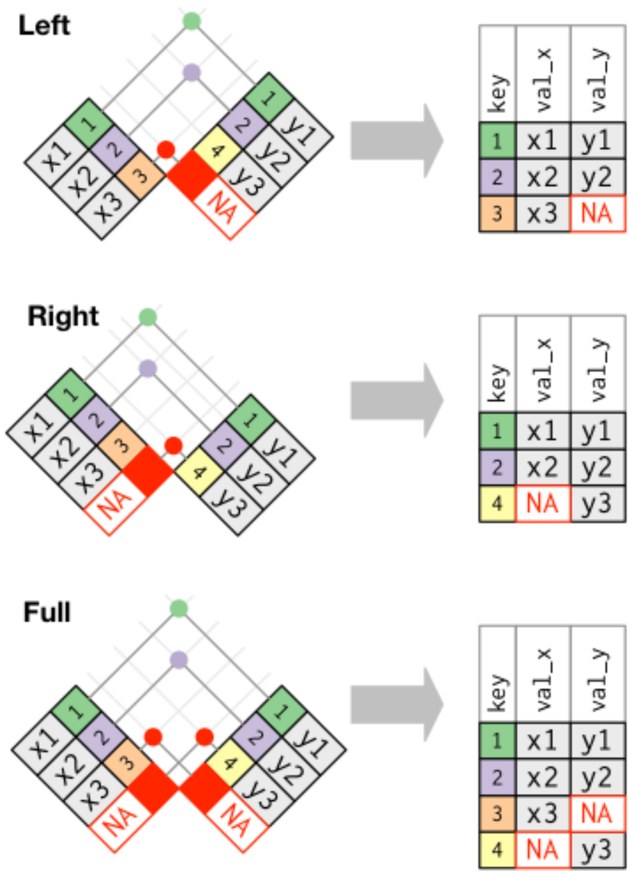
person	name	value
Bob	Age	32
Bob	Height	72
Alice	Age	24
Alice	Height	65
Steve	Age	64
Steve	Height	70

Wide data

Person	Age	Height
Bob	32	72
Alice	24	65
Steve	64	70

Let's try it in R…

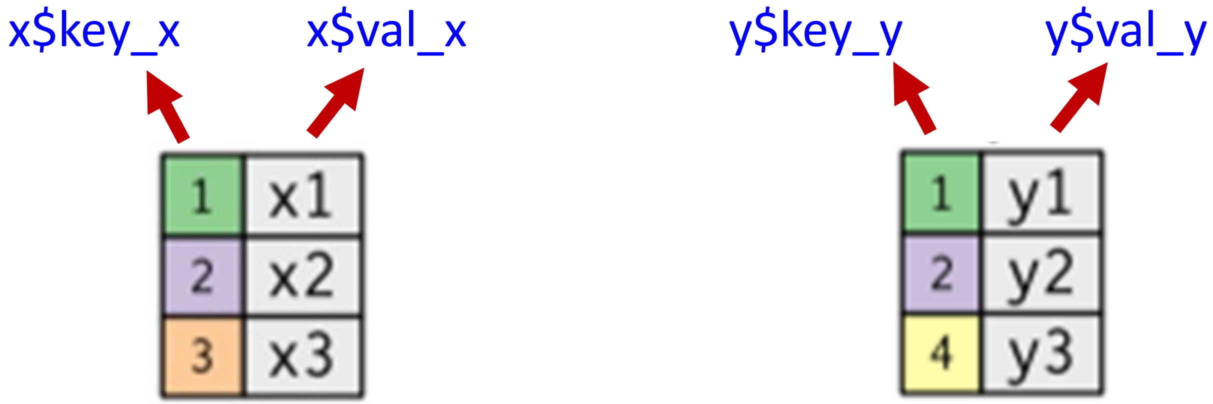
Joining data frames



Left and right tables

Suppose we have two data frames called x and y

- x have two variables called key_x, and val_x
- y has two variables called key_y and val_y



Data frame x

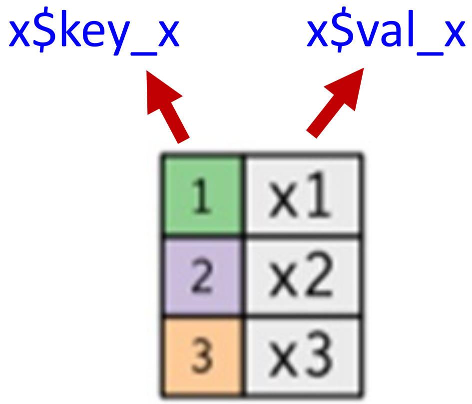
Data frame y

SDS230:download_data('x_y_join.rda')

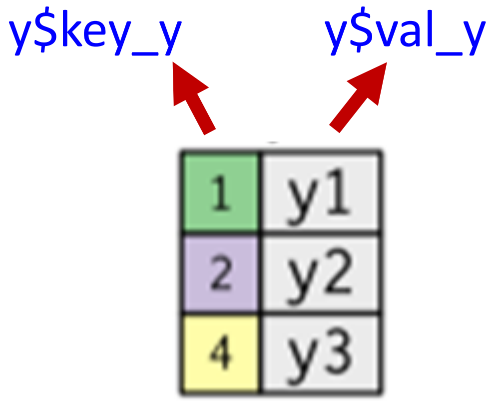
Left and right tables

Suppose we have two data frames called x and y

- x have two variables called key_x, and val_x
- y has two variables called key_y and val_y



Data frame x



Data frame y

Joins have the general form:

```
join(x, y, by = c("key_x" = "key_y"))
```

Left joins

Left joins keep all rows in the left table.

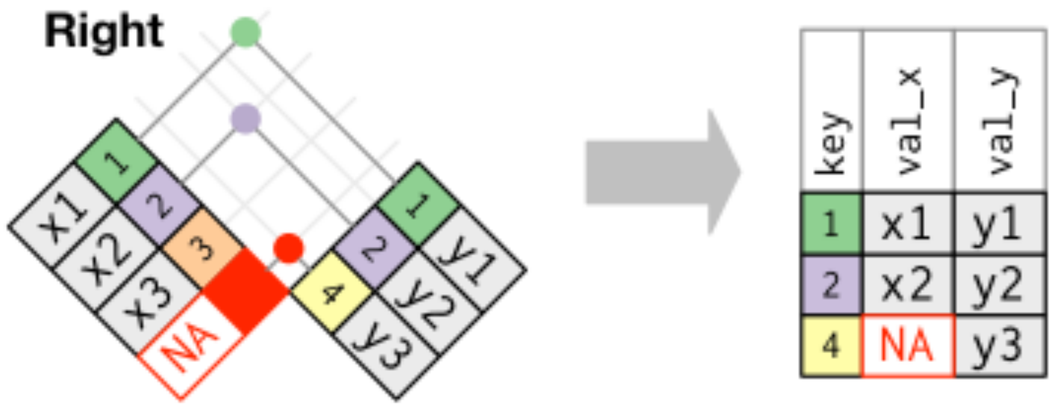
Data from right table is added when there is a matching key, otherwise NA as added.


```
> left_join(x, y, by = c("key_x" = "key_y"))
```

Right joins

Right joins keep all rows in the right table.

Data from left table added when there is a matching key, otherwise NA as added.



> right_join(x, y, by = c("key_x" = "key_y"))

Inner joins

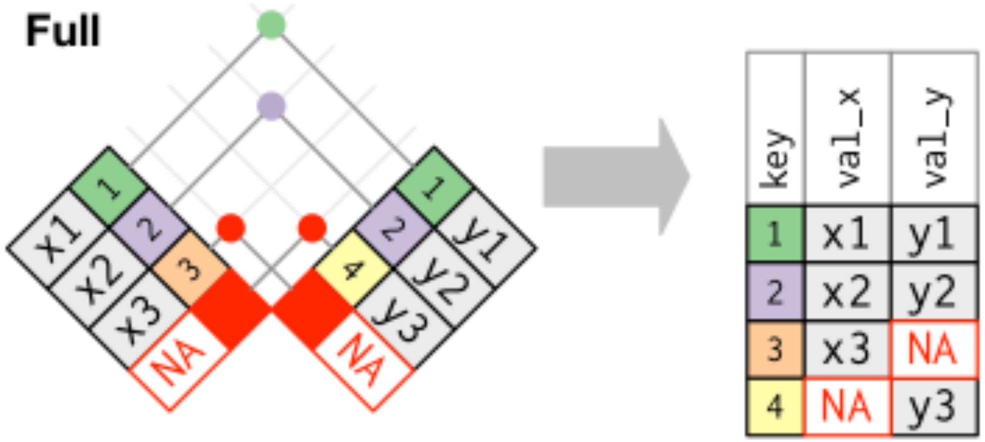
Inner joins only keep rows in which there are matches between the keys in both tables.


```
> inner_join(x, y, by = c("key_x" = "key_y"))
```

Full joins

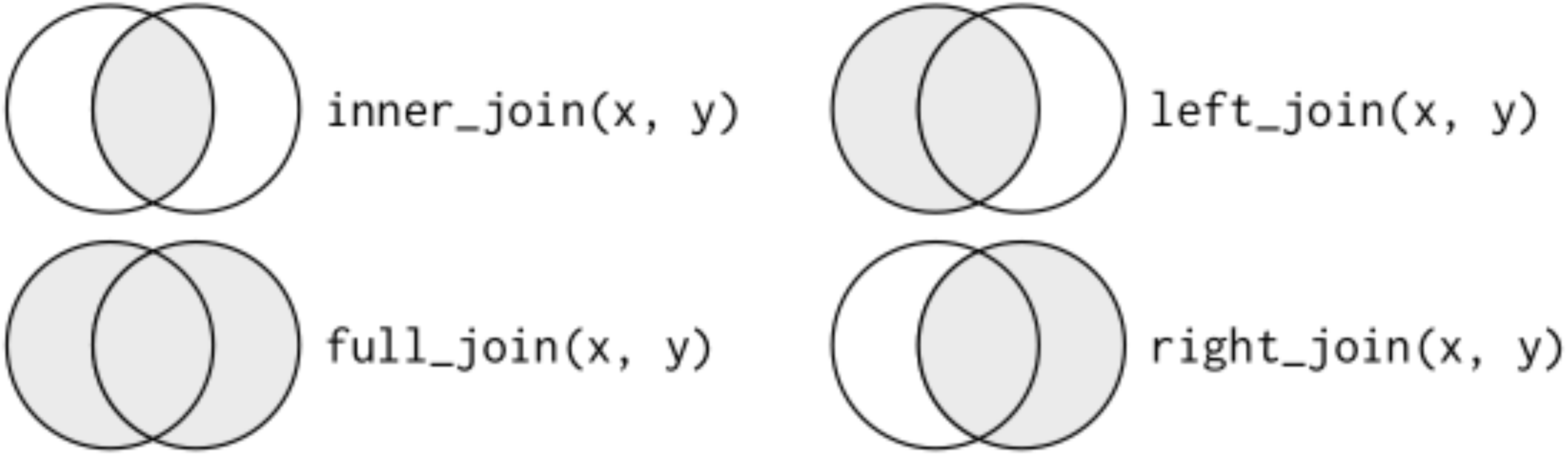
Full joins keep all rows in both table.

NAs are added where there are no matches.



> full_join(x, y, by = c("key_x" = "key_y"))

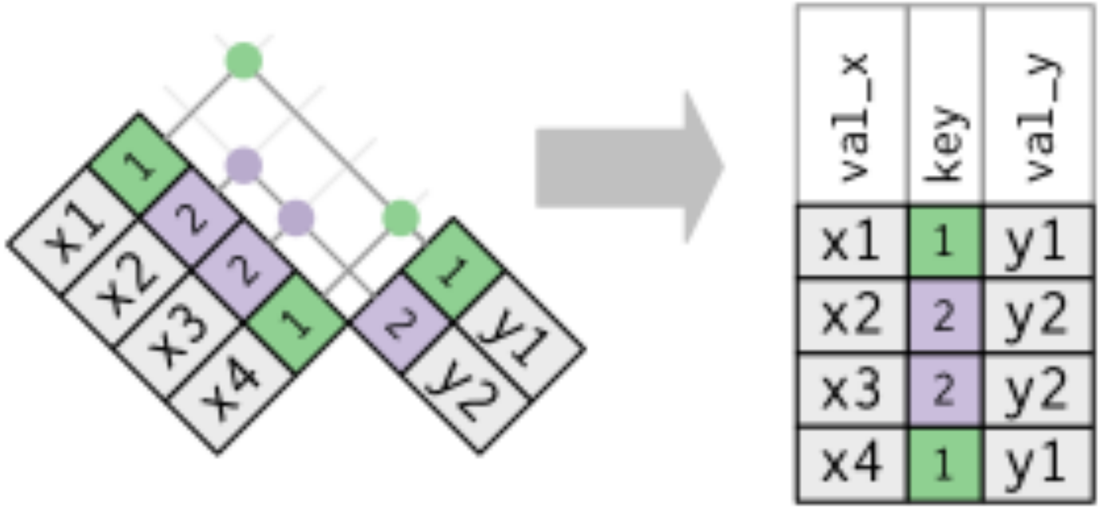
Summary



Duplicate keys

Duplicate keys are useful if there is a many-to-one relationship

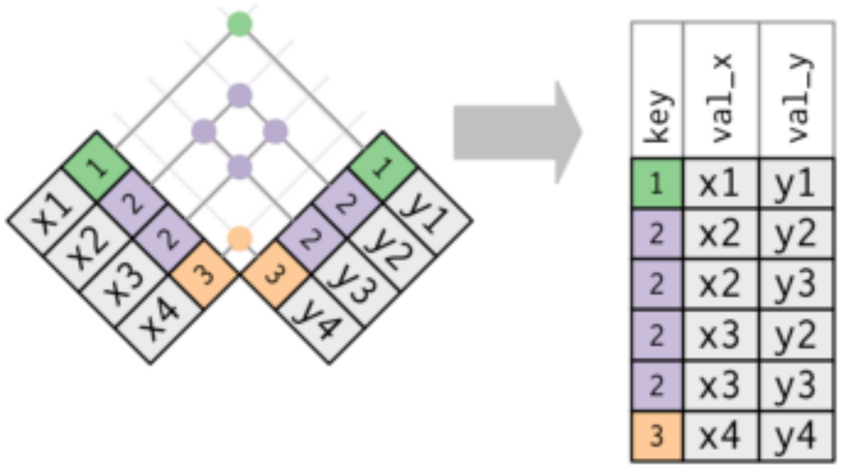
- e.g., duplicates are useful in the left table when doing a left join



Duplicate keys

If both tables have duplicate keys you get all possible combinations of joined values (Cartesian product).

- **This is usually an error!**



Always check the output size after you join a table because even if there is not a syntax error you might not get the table you are expecting!

- You can check how many rows a data frame has using the nrow() function

Duplicate keys

To deal with duplicate keys in both tables, we can join the tables using multiple keys in order to make sure that each row is uniquely specified.

We can do this using the syntax:

```
join(x2, y2, by = c("key1_x" = "key1_y", "key2_x" = "key2_y"))
```

Duplicate keys

```
>  x2 <- data.frame(key1_x = c(1, 2, 2),
                   key2_x = c("a", "a", "b"),
                   val_x = c("y1", "y2", "y3"))

>  y2 <- y2 <- data.frame(key1_y = c(1, 2, 2, 3, 3),
                   key2_y = c("a", "a", "b", "a", "b"),
                   val_y = c("y1", "y2", "y3", "y4", "y5"))

> left_join(x2, y2, c("key1_x" = "key1_y"))

> left_join(x2, y2, c("key1_x" = "key1_y", "key2_x" = "key2_y"))
```

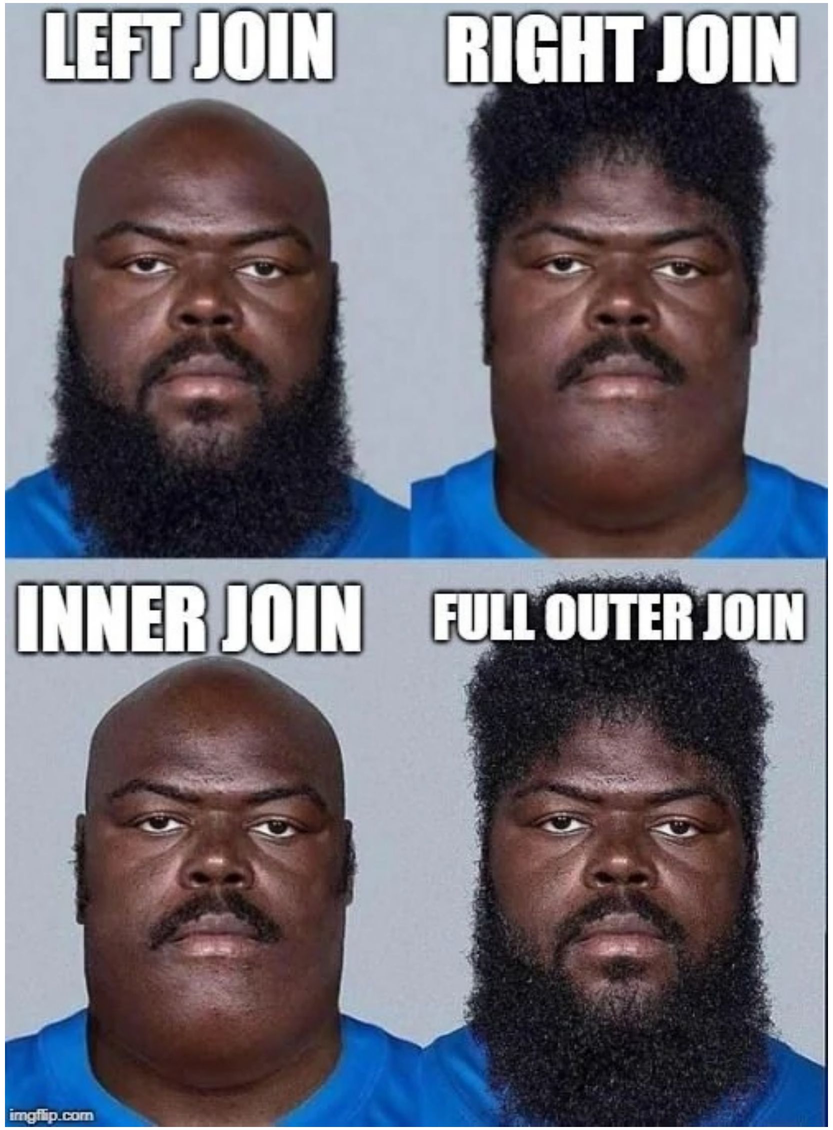
Structured Query Language

Having multiple tables that can be joined together is common in Relational Database Systems (RDBS).

- A common language used by RDBS is Structured Query Language (SQL)

dplyr	SQL
`inner_join(x, y, by = "z")`	`SELECT * FROM x INNER JOIN y USING (z)`
`left_join(x, y, by = "z")`	`SELECT * FROM x LEFT OUTER JOIN y USING (z)`
`right_join(x, y, by = "z")`	`SELECT * FROM x RIGHT OUTER JOIN y USING (z)`
`full_join(x, y, by = "z")`	`SELECT * FROM x FULL OUTER JOIN y USING (z)`

Let’s try it in R…



Brief mention: Interactive applications using Shiny

Shiny

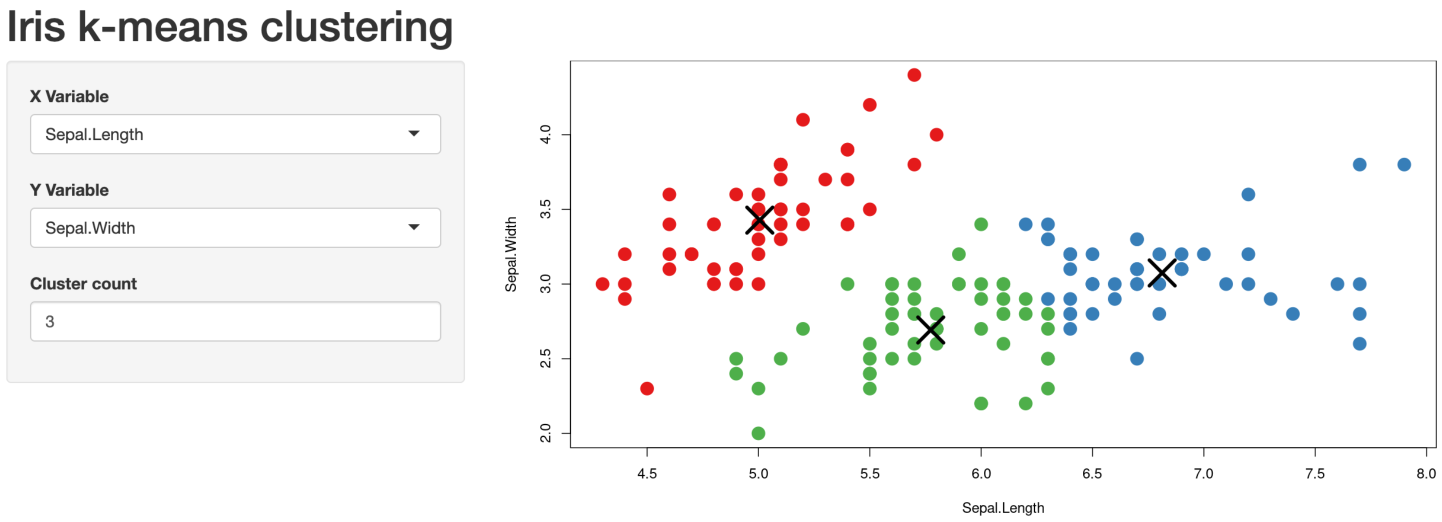
Shiny is an R package that makes it easy to build interactive web apps

- Tutorial: https://shiny.rstudio.com/tutorial/

Example: k-means clustering on Fisher's Iris data set

Setosa

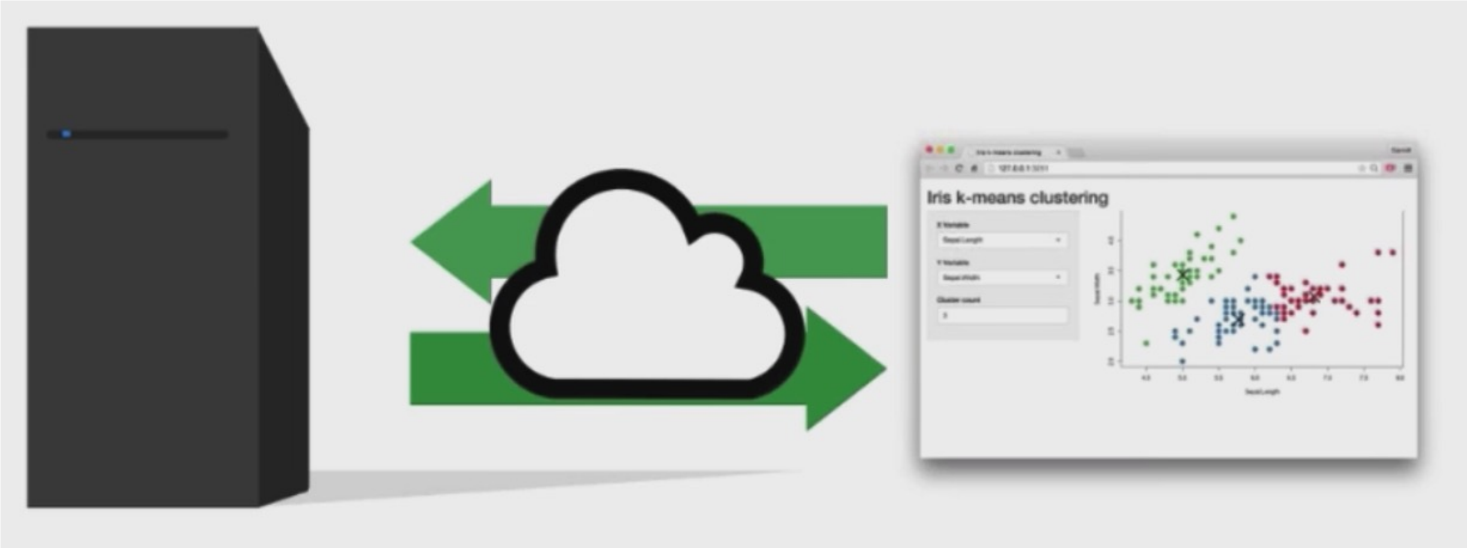
Virginica



Shiny applications

Server runs R code,
creates results

Client uses a web-based
GUI to interact with code



You need to write 2 pieces of code to create a Shiny app:

- server: for the code that is run on the server
- ui: for the web interface shown to the user

Shiny application template

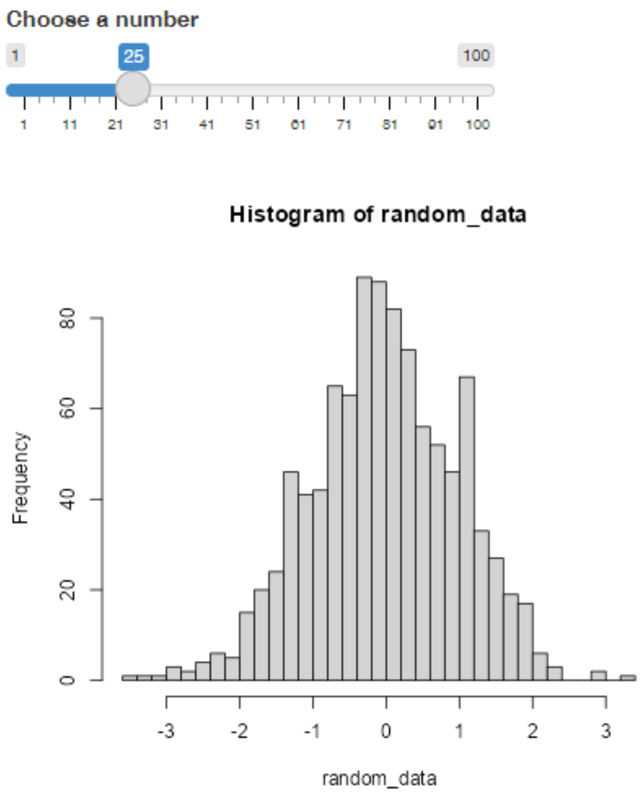
```
# include the shiny package
library(shiny)

# the function to create the user interface
ui <- fluidPage()

# the function to create the server
server <- function(input, output) {}

# putting them together to run
shinyApp(ui = ui, server = server)
```

UI code converted to HTML for web browsers

Example of a full app...

An app that displays 1000 random points in a histogram

- The number of bins in the histogram can be changed with a slider input




```
random_data <- rnorm(1000)

ui <- fluidPage(

        sliderInput(inputId = "num",
          label = "Choose a number",
          val = 25, min = 1, max = 100),

        plotOutput("my_plot")
)

server <- function(input, output) {

        output$my_plot <-   renderPlot({
                hist(random_data, breaks = input$num))
        })
}

shinyApp(ui = ui, server = server)
```

Let's quickly explore this in R…

Ethics in Statistics and Data Science

Ethics in Data Science

Ethics of:

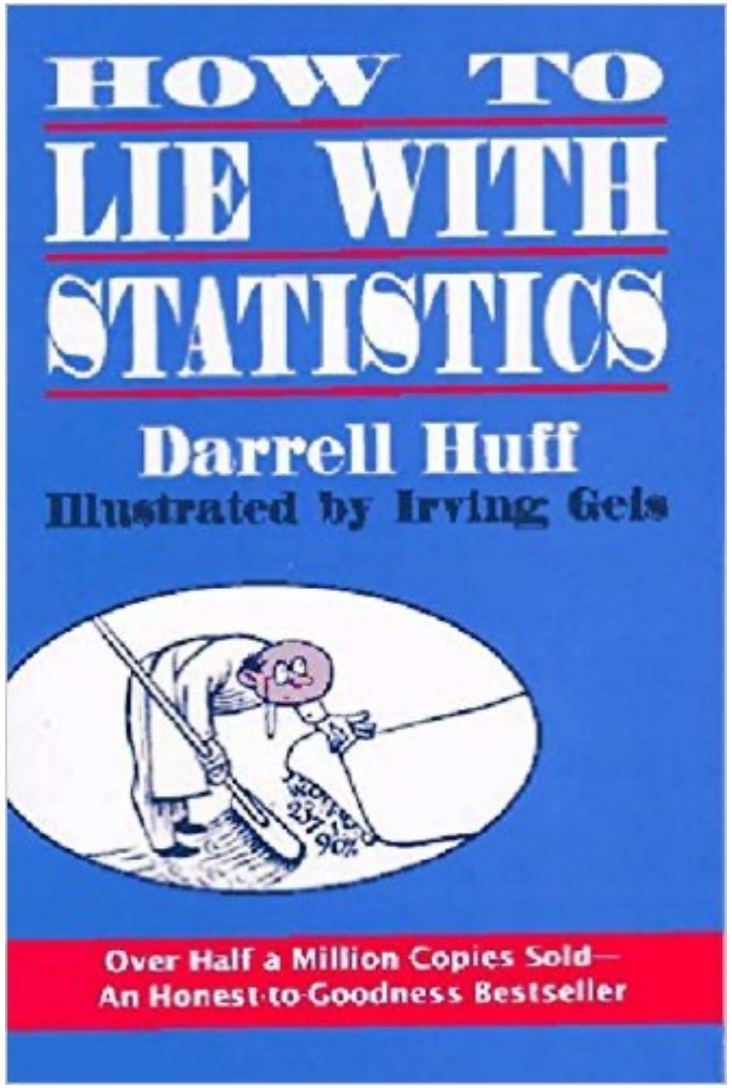
1. Data presentation
2. Using valid data
3. Data scraping TOS and privacy
4. Reproducibility
5. Citations/peer review
6. Disclosure
7. Honestly conducting inferential analyses
8. Ethics of creating powerful machine learning tools

1. Ethics of data presentation

Data should be displayed in an honest way that gives an accurate picture of trends

Darrell Huff wrote a classic book in the 1950’s pointing out ways that people lie with statistics

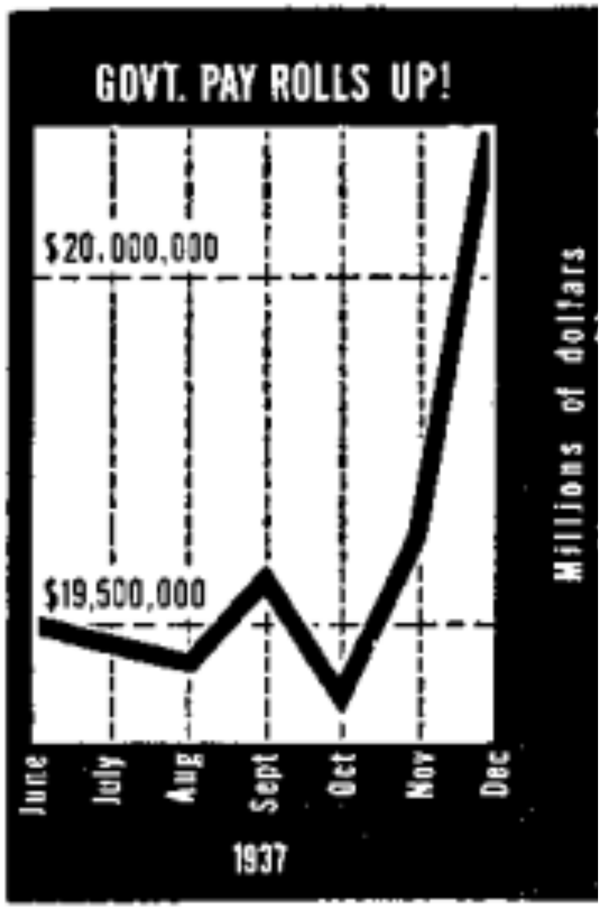
[The book was banned as training material at the VA]



Ethics of data presentation

What is potentially misleading with this figure?

Only a 4% increase in payroll



From a 1938 article in Dun's Review titled 'GOVERNMENT PAY ROLLS UP!'

3. Data scraping, terms of service and privacy

Scraping publicly available data is fine (e.g., Wikipedia) but what about scraping data if:

- It violates a website's Terms of Service?
- User privacy?

Kirkegaard and Bjerrekaer scraped okcupid and data on 68,371 users publicly available including usernames, dating preferences, etc.

- Is this ok?

Submitted: 8th of May 2016
Published: 3rd of November 2016

The OKCupid dataset: A very large public dataset of dating site users

Emil O. W. Kirkegaard* Julius D. Bjerrekær†

Open Differential Psychology

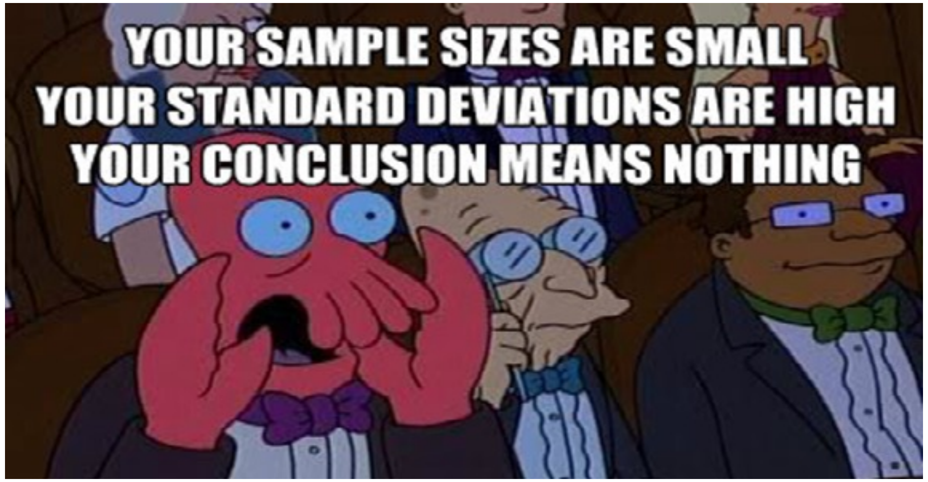
7. Ethics in statistical analyses

P-hacking (data dreging):

Trying many different hypothesis tests on a data set until you reach ‘statistical significance’ ($p < 0.05$)

File drawer effect:

- Try a million studies until one is significant

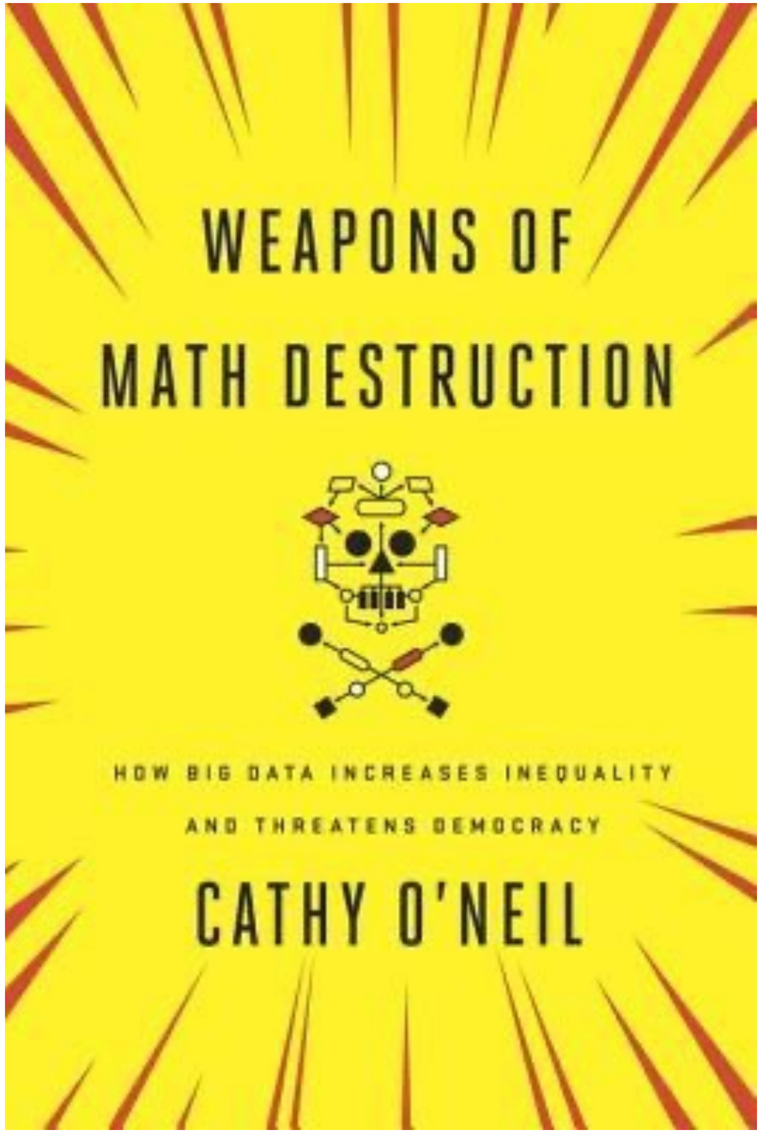


8. Ethics in machine learning

Care must be taken when interpreting the results from machine learning algorithms

To learn more…

Take S&DS 150: Data Science Ethics



Wrap up and conclusions

Topics we will cover

~~**R and descriptive statistics/plots**: Base R, fundamental concepts in Statistics~~

~~**Review confidence intervals:** Sampling and bootstrap distributions~~

~~**Review of hypothesis tests:** Permutation and parametric tests, theories of testing~~

~~**Data wrangling:** filtering and summarizing data, joining data sets, reshaping data~~

~~**Data visualization:** grammar of graphics~~

~~**Regression:** simple/multiple, non-linear terms, logistic regression~~

~~**ANOVA:** one-way/factorial, interactions~~

~~**Statistical learning:** cross-validation, logistic regression~~

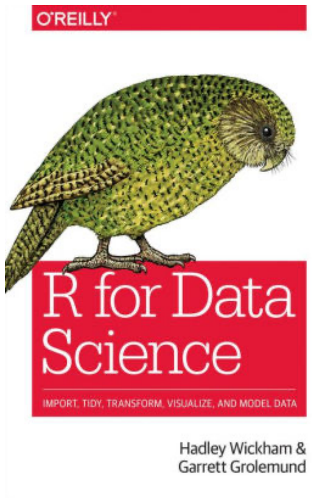
Course objectives

- ✓ Extend and solidify concepts and method learned in intro stats
- ✓ Learn how to use the R programming language to analyze, visualize and wrangle data
- ✓ Gain experience extracting insights from real data
- ✓ **Learn how to find patterns in a large noisy data sets and convincingly convey the results to others!**

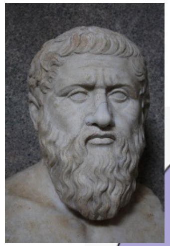
Next steps

Take more advanced Statistics and Data Science classes offered at Yale!

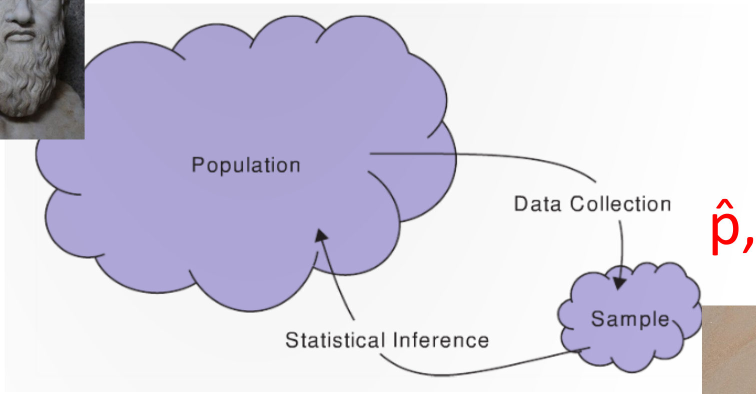
There are many good online resources to learn more R



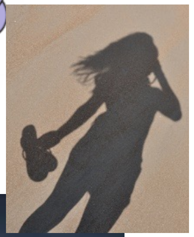
A final take away message…



$\pi, \mu, \sigma, \rho, \beta$



$\hat{p}, \overline{x}, s, r, b$



Thanks to the teaching assistants!!!

Teaching Fellows (TF)

- Zihan Chen chen@yale.edu
- Sahil Singh singh@yale.edu
- William Zhang william.j.zhang@yale.edu

Undergraduate Learning Assistants (ULA)

- Sohum Kapadia kapadia@yale.edu
- Adia Keene keene@yale.edu
- Selma Mazioud mazioud@yale.edu
- Ryan Vaughn vaughn@yale.edu

Course manager

- Abby Spears abby.spears@yale.edu

Good luck with the end of the semester!

Good luck finishing your final projects!

The final exam is on Saturday December 16 at 7pm