Week 1: Exploratory Data Analysis

EMSE 4197 | John Paul Helveston | January 15, 2020

Meet your instructor!

Dr. John Helveston

Assistant Professor in Engineering Management & Systems Engineering

Background:

- 2016 PhD in Engineering & Public Policy at Carnegie Mellon University
- 2015 MS in Engineering & Public Policy at Carnegie Mellon University
- 2010 BS in Engineering Science & Mechanics at Virginia Tech

Research:

- Modeling conusmer preferences
- Electric vehicle adoption & diffusion
- China

Meet your tutors!

Yanjie He

Masters student in Data Analytics

Lingmei Zhao

Masters student in Statistics

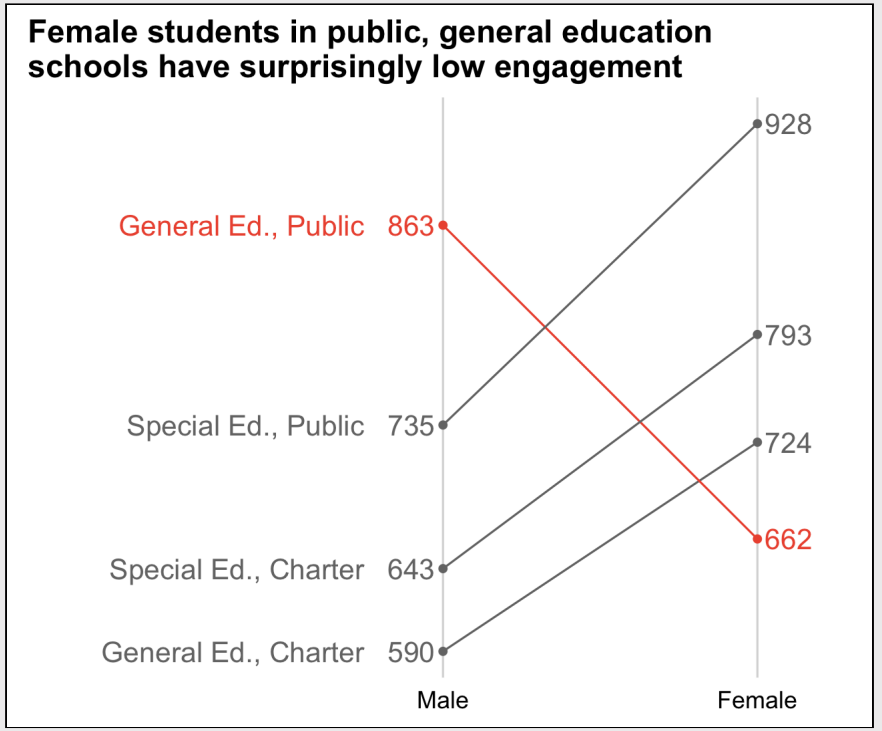
Class goal: translate *data* into *information*

Data

Student engagement scores

Class	School Type	Male	Female
Special Ed.	Charter	643	793
Special Ed.	Public	735	928
General Ed.	Charter	590	724
General Ed.	Public	863	662

Information



Data exploration: an iterative process

Encode data:

```
engagement_data <- data.frame(
    Male   = c(643, 735, 590, 863),
    Female = c(793, 928, 724, 662),
    School = c('Special Ed., Charter', 'Special Ed., Public',
               'General Ed., Charter', 'General Ed., Public'))
engagement_data
```

```
##   Male Female               School
## 1  643    793 Special Ed., Charter
## 2  735    928  Special Ed., Public
## 3  590    724 General Ed., Charter
## 4  863    662  General Ed., Public
```

Re-format data for plotting:

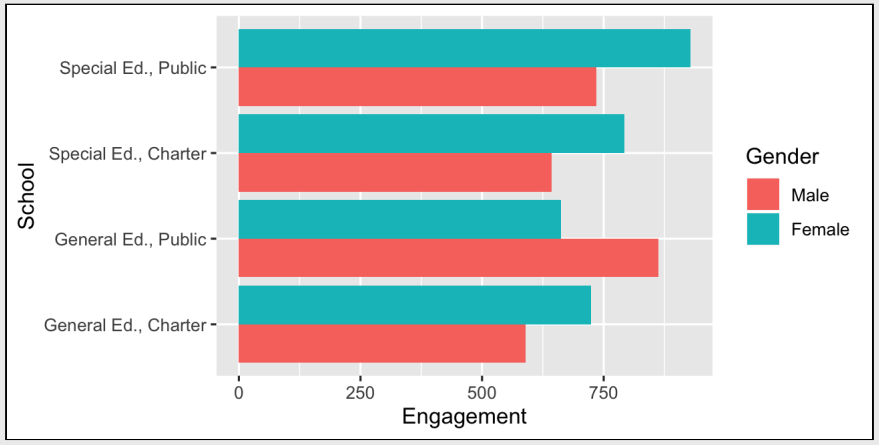
```
engagement_data <- engagement_data %>%
    gather(Gender, Engagement, Male:Female) %>%
    mutate(Gender = fct_relevel(
      Gender, c('Male', 'Female')))
engagement_data
```

```
##                 School Gender Engagement
## 1 Special Ed., Charter   Male        643
## 2  Special Ed., Public   Male        735
## 3 General Ed., Charter   Male        590
## 4  General Ed., Public   Male        863
## 5 Special Ed., Charter Female        793
## 6  Special Ed., Public Female        928
## 7 General Ed., Charter Female        724
## 8  General Ed., Public Female        662
```

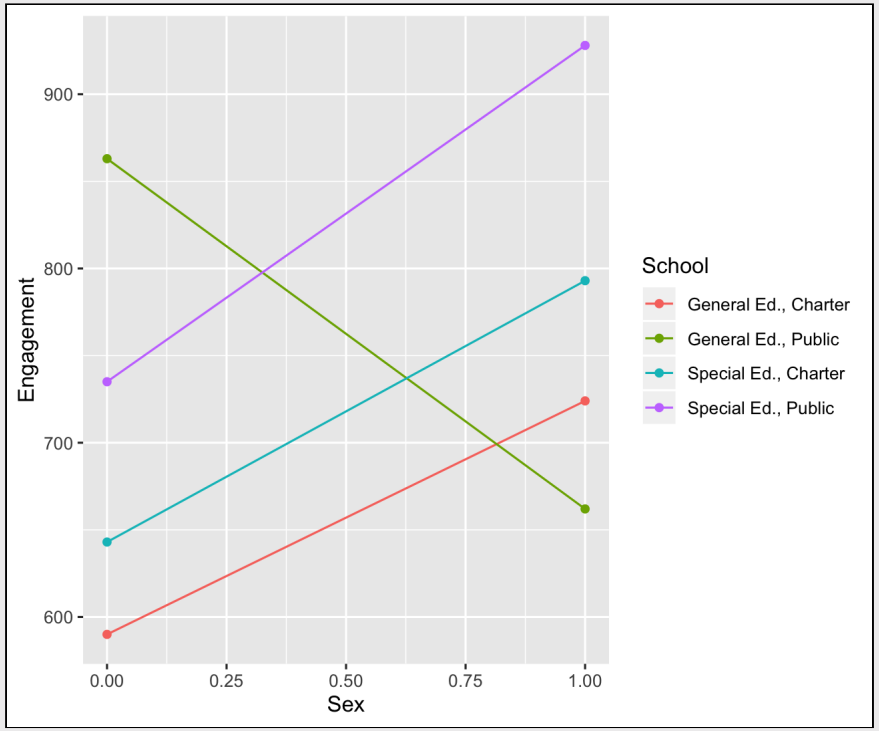
Data exploration: an iterative process

Initial exploratory plotting:

```
engagement_data %>%
    ggplot() +
    geom_bar(aes(x = School, y = Engagement,
                 fill = Gender), stat = 'identity',
             position = 'dodge') +
    coord_flip()
```

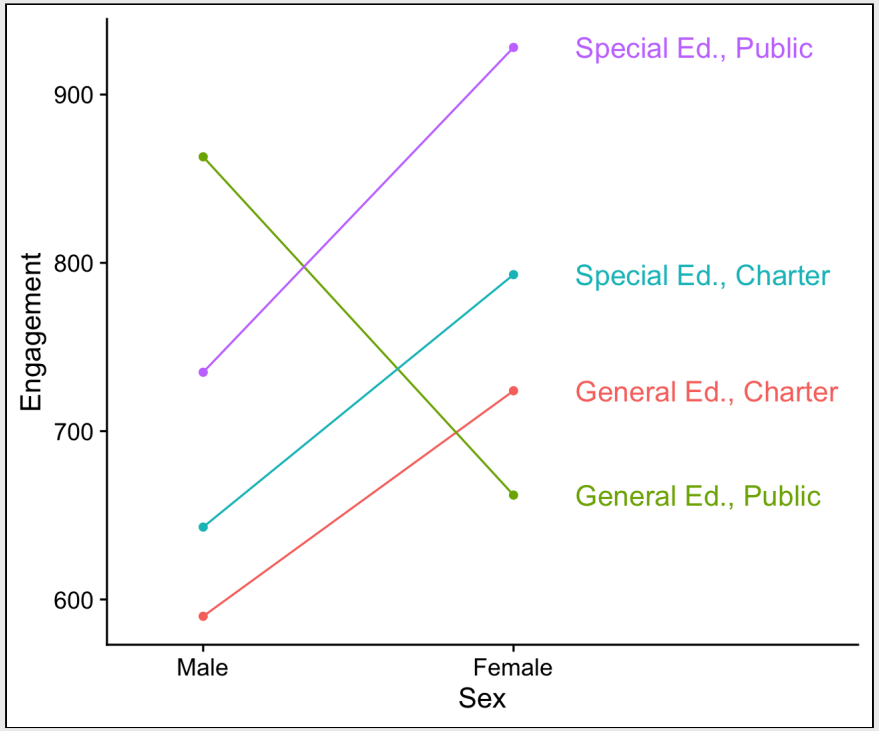


More exploratory plotting - highlight difference:

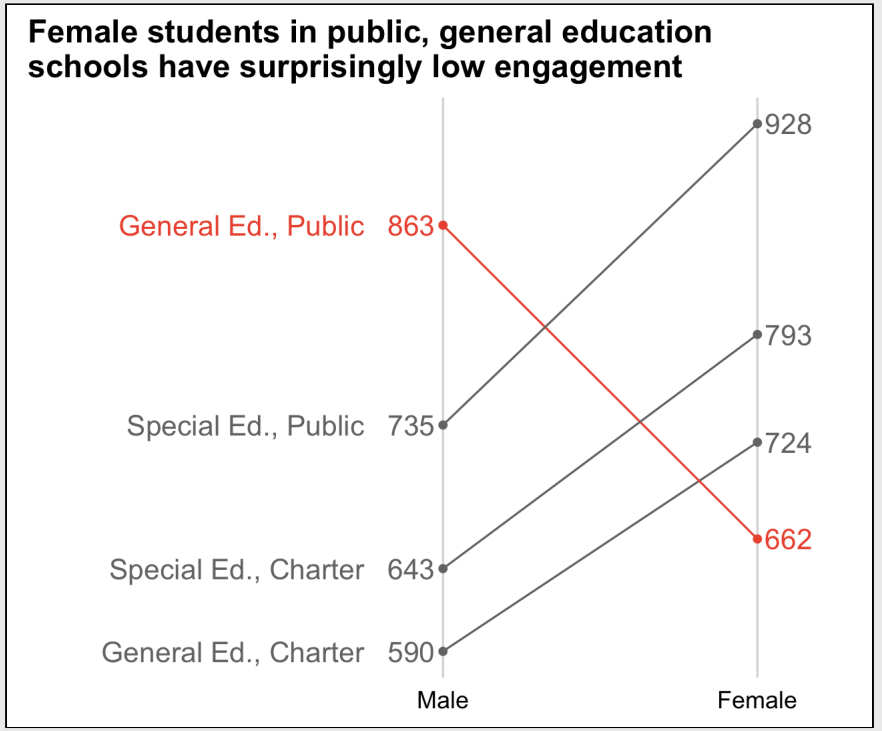


Data exploration: an iterative process

Directly label figure:



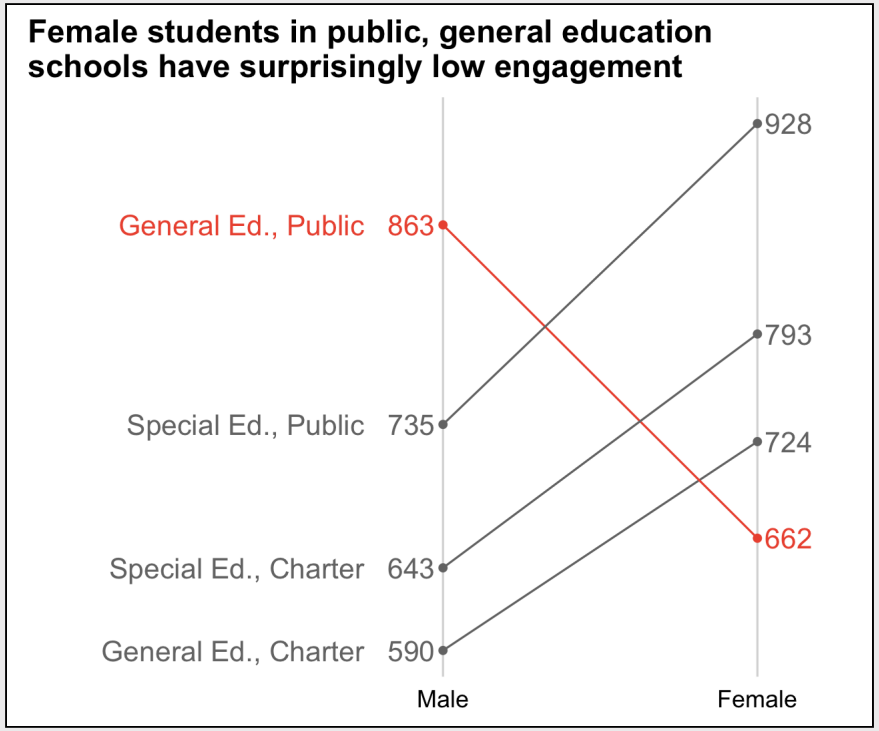
Remove unnecessary axes & format colors:



Code:

```
data.frame(
    Male   = c(643, 735, 590, 863),
    Female = c(793, 928, 724, 662),
    School = c('Special Ed., Charter', 'Special Ed., Public',
               'General Ed., Charter', 'General Ed., Public'),
    Highlight = c(0, 0, 0, 1)) %>%
    gather(Gender, Engagement, Male:Female) %>%
    mutate(
      Gender = fct_relevel(Gender, c('Male', 'Female')),
      Highlight = as.factor(Highlight),
      x = ifelse(Gender == 'Female', 1, 0)) %>%
    ggplot(aes(x = x, y = Engagement, group = School, color = Highlight)) +
    geom_point() +
    geom_line() +
    scale_color_manual(values = c('#757575', '#ed573e')) +
    labs(x = 'Sex', y = 'Engagement',
         title = paste0('Female students in public, general education\n',
                        'schools have surprisingly low engagement')) +
    scale_x_continuous(limits = c(-1.2, 1.2), labels = c('Male', 'Female'),
                       breaks = c(0, 1)) +
    geom_text_repel(aes(label = Engagement, color = as.factor(Highlight)),
                    data          = subset(engagement, Gender == 'Female'),
                    size          = 5,
                    nudge_x       = 0.1,
                    segment.color = NA) +
    geom_text_repel(aes(label = Engagement, color = as.factor(Highlight)),
                    data          = subset(engagement, Gender == 'Male'),
                    size          = 5,
                    nudge_x       = -0.1,
                    segment.color = NA) +
    geom_text_repel(aes(label = School, color = as.factor(Highlight)),
                    data          = subset(engagement, Gender == 'Male'),
                    size          = 5,
                    nudge_x       = -0.25,
                    hjust         = 1,
                    segment.color = NA) +
    theme_cowplot() +
    background_grid(major = 'x') +
    theme(axis.line = element_blank(),
          axis.title.x = element_blank(),
          axis.title.y = element_blank(),
          axis.text.y = element_blank(),
          axis.ticks = element_blank(),
          legend.position = 'none')
```

Chart:



A fully reproducible analysis

Course 1: Programming for Analytics

"Computational Literacy"

- Programming: Conditionals (if/else), loops, functions, testing, data types.
- Analytics: Data structures, import / export, basic data manipulation & visualization.

Course 2: Exploratory Data Analysis

"Data Literacy"

- Strategies for systematically exploring data.
- Design principles for visualizing and communicating information contained in data.
- Reproducibility: Reports that contain code, equations, visualizations, and narrative text.

Course orientation

- Course website (link also in Blackboard): https://emse-eda-gwu.github.io/2020-Spring/index.html
- The schedule is the best starting point
- **Prerequisites**: Programming for Analytics - look at Assignment 0
- For help, look under the "Resources" tab
    - Use Slack to ask questions.
    - Go to Office hours / tutor sessions

Course policies

Basic policies

- **BE NICE**
- **BE HONEST**
- **DON'T CHEAT** (Translate: Write your own code!)

Late submissions

- **5** late days - use them however you want.
- You can't use more than **2** late days on any one assignment.

Assignments

1) Online exercises

- Example: Assignment 1
- DataCamp tour

2) Mini projects

3) Final Project

- Teams of 1 - 3 students
- 5 separate deliverables

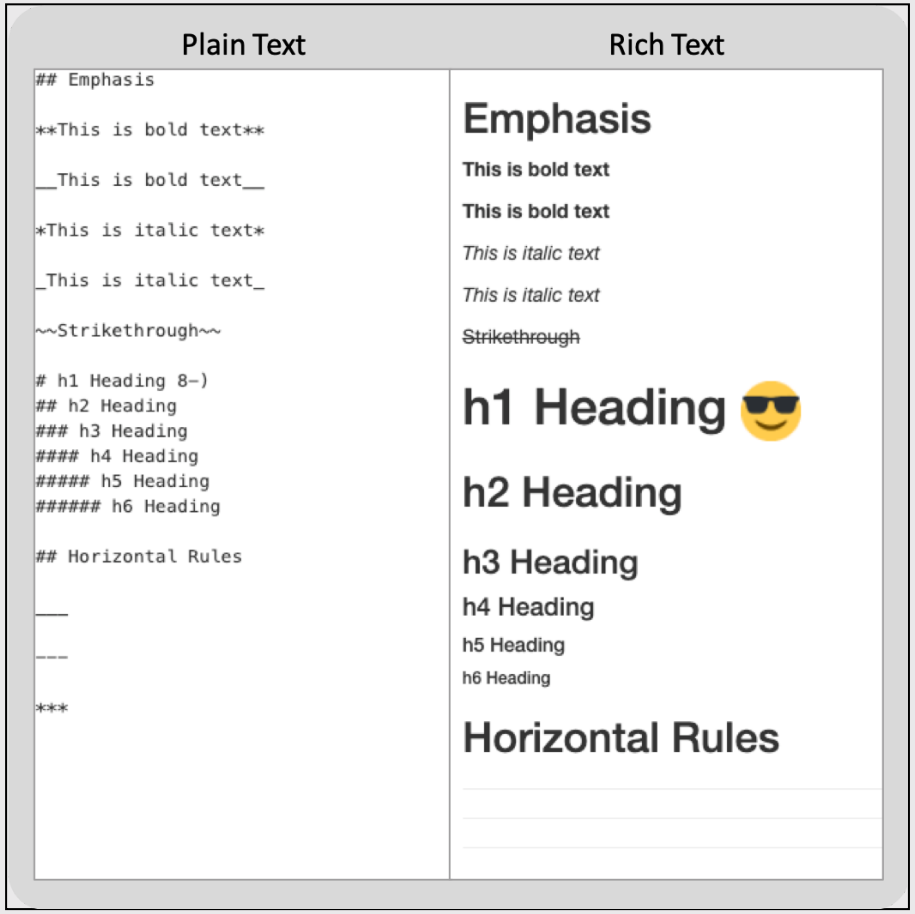
There is no final exam - the final project is the final exam

Grades

Item	Weight	Notes
Attendance & Participation	10 %	
Quizzes	15 %	Lowest quiz grade is dropped
Assignment 1	5 %	Exercises
Assignment 2	5 %	Exercises
Assignment 3	10 %	Mini Project
Assignment 4	10 %	Mini Project
Assignment 5	5 %	Exercises
Final Project Proposal	5 %	
Final Project Progress Report	5 %	
Final Project Peer Review	5 %	
Final Project Presentation	10 %	
Final Project Report	15 %	

Course Mantras

1) Embrace **plain text**



Plain Text

```
## Emphasis

**This is bold text**

__This is bold text__

*This is italic text*

_This is italic text_

~~Strikethrough~~

# h1 Heading 8-)
## h2 Heading
### h3 Heading
#### h4 Heading
##### h5 Heading
###### h6 Heading

## Horizontal Rules

___

---

***
```

Rich Text

Emphasis

This is bold text

This is bold text

This is italic text

This is italic text

~~Strikethrough~~

h1 Heading 😎

h2 Heading

h3 Heading

h4 Heading

h5 Heading

h6 Heading

Horizontal Rules

2) Embrace **reproducibility**.

RMarkdown -> HTML

Example:

This presentation was generated from R code!

How to succeed in this class

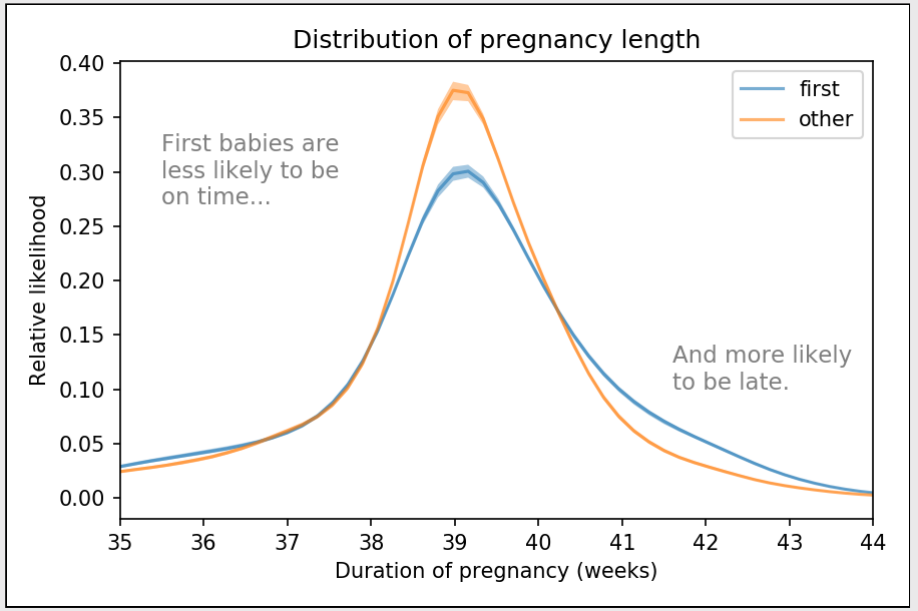
- **Take notes** - in class, doing assignments...basically all the time :)
- Start assignments early!
- Don't cheat!
- Come to office hours / tutor sessions!

Life is complicated

No class on 4/15 & 4/22...

...because this is happening on 4/13

...but nothing in life in certain



Graph from this analysis

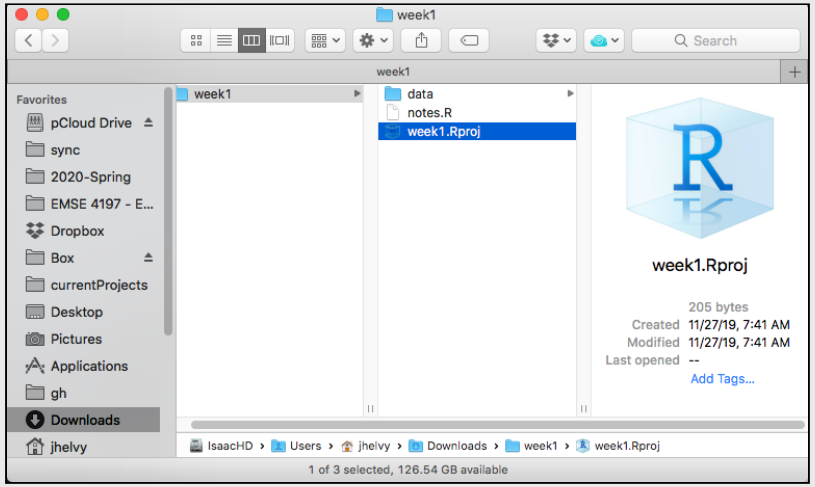
Data includes 31,906 births in the U.S.

Getting Started

1. Sign up for Slack - link in Blackboard announcement
2. Install Slack app and **turn notifications on**
3. Sign up for DataCamp - **you must use your @gwu.edu email**
4. Sign up for RStudio Cloud.
5. Download and install R
6. Download and install RStudio (Desktop version)

Workflow

1) Use R Projects to organize your analysis



First example: go to the "classroom" channel in Slack

2) Use the `here` package to create file paths

3) Use these functions to import data:

Data file type	Import function	Library
`.csv`	`read_csv()`	**readr**
`.txt`	`read.table()`	**readr**
`.xlsx`	`read_excel()`	**readxl**

Data Import Examples

Read in `.csv` files with `read_csv()`:

```
library(tidyverse)
library(here)

csvPath <- here('data', 'milk_production.csv')
milk_production <- read_csv(csvPath)
```

Read in `.txt` files with `read.table()`:

```
txtPath <- here('data', 'nasa_global_temps.txt')
global_temps <- read.table(txtPath, skip = 5)
```

Read in `.xlsx` files with `read_excel()`:

```
library(readxl)

xlsxPath <- here('data', 'pv_cell_production.xlsx')
pv_cells <- read_excel(xlsxPath, sheet = 'Cell Prod by Country', skip = 2)
```

Your turn

Write code to import the following data files from the "`data`" folder:

- `wildlife_impacts.csv`
- `north_america_bear_killings.txt`
- `pc_sales_2018.xlsx`

Data provenance - It matters where you get your data

- **Validity**:
  - Is this data trustworthy? Is it authentic?
  - Where did the data come from?
  - How has the data been changed / managed over time?
  - Is the data complete?

- **Comprehension**:
  - Is this data accurate?
  - Can you explain your results?
  - Are you using the *right* data to answer your question?

- **Reproducibility**: The data source is the start of the reproducibility chain.

Document your source like a museum curator

[**Example**: View "`data_sources.txt`" file]

Whenever you download data, you should **at a minimum** record the following:

- The name of the file you are describing.
- The date you downloaded the file (i.e. today's date).
- The original name of the downloaded file (often times you'll rename the original file name).
- The url to the site you downloaded the data from.
- The source of the *original* data (often different from where you downloaded the data).
- A short description of the data and how they were collected.
- A dictionary for the data.

Your turn - in 3 groups

Go to the site listed in the `notes.R` file and add the following information about the data to the "`data_sources.txt`" file:

- The name of the downloaded file in the "`data`" folder.
- The date you downloaded the file (i.e. today's date).
- The url to the site you downloaded the data from.
- The source of the *original* data (if different from where you downloaded the data).
- A short description of the data and how they were collected.
- A dictionary for the data (if available).

5 minute break!

Stand up

Move around

Stretch!

Variables, values, and observations

- **Variable**: A quantity, quality, or property that you can measure.
- **Value**: The state of a variable when you measure it.
- **Observation**: A set of measurements that are made under similar conditions

Table 1

```
## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>       <int>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

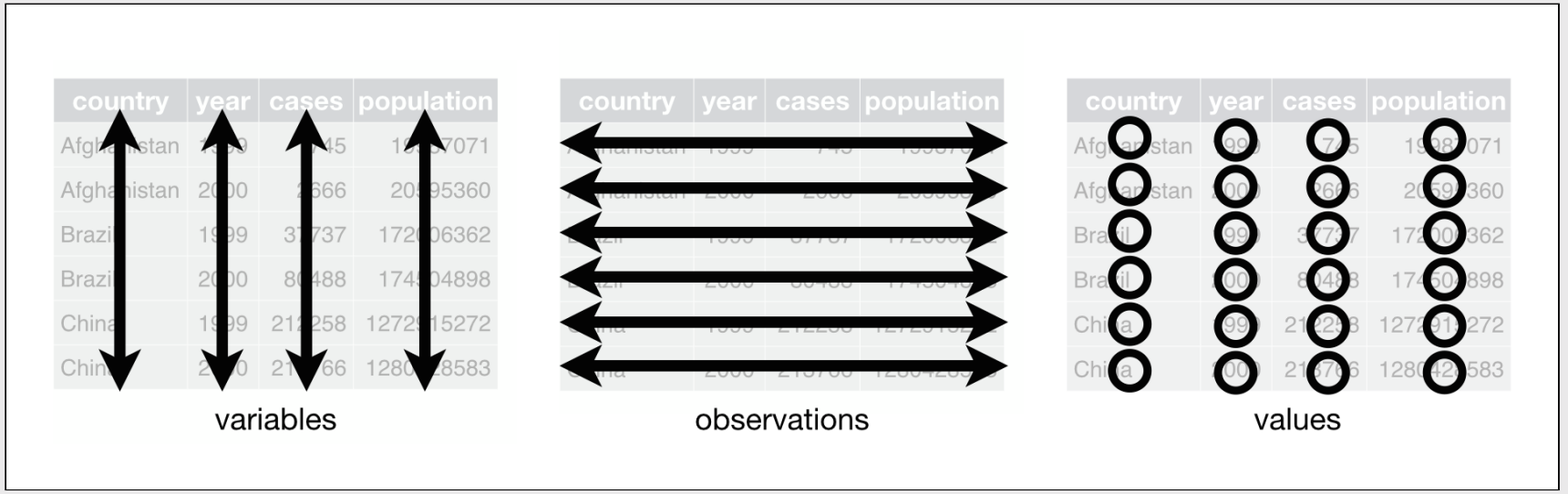
Table 2

```
## # A tibble: 6 x 4
##   country      year type           count
##   <chr>       <int> <chr>          <int>
## 1 Afghanistan  1999 cases            745
## 2 Afghanistan  1999 population  19987071
## 3 Afghanistan  2000 cases           2666
## 4 Afghanistan  2000 population  20595360
## 5 Brazil       1999 cases          37737
## 6 Brazil       1999 population 172006362
```

Tidy data

Tidy data follows the following three rules:

- Each **variable** has its own **column**.
- Each **observation** has its own **row**.
- Each **value** has its own **cell**.



Tidy data

- **Variable**: A quantity, quality, or property that you can measure.
- **Value**: The state of a variable when you measure it.
- **Observation**: A set of measurements that are made under similar conditions

Tidy

```
## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>       <int>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

Un-tidy

```
## # A tibble: 6 x 4
##   country      year type           count
##   <chr>       <int> <chr>          <int>
## 1 Afghanistan  1999 cases            745
## 2 Afghanistan  1999 population  19987071
## 3 Afghanistan  2000 cases           2666
## 4 Afghanistan  2000 population  20595360
## 5 Brazil       1999 cases           37737
## 6 Brazil       1999 population 172006362
```

Tidy data:

- **Variable**: A quantity, quality, or property that you can measure.
- **Value**: The state of a variable when you measure it.
- **Observation**: A set of measurements that are made under similar conditions

Example: PV Solar Cell Production Data

Un-tidy format:

```
head(pv_cells)
```

```
## # A tibble: 6 x 10
##   Year  China Taiwan Japan Malaysia Germany `South Korea` `United States`
##   <chr> <chr> <chr>  <dbl> <chr>    <chr>   <chr>                   <dbl>
## 1 1995  NA    NA      16.4 NA       NA      NA                       34.8
## 2 1996  NA    NA      21.2 NA       NA      NA                       38.8
## 3 1997  NA    NA      35   NA       NA      NA                       51
## 4 1998  NA    NA      49   NA       NA      NA                       53.7
## 5 1999  NA    NA      80   NA       NA      NA                       60.8
## 6 2000  2.5   NA     129.  NA       22.5    NA                       75
## # … with 1 more variable: World <dbl>
```

Tidy format:

```
head(pv_cells_tidy)
```

```
## # A tibble: 6 x 3
##   Year  Country nPVCells
##   <chr> <chr>   <chr>
## 1 1995  China   NA
## 2 1996  China   NA
## 3 1997  China   NA
## 4 1998  China   NA
## 5 1999  China   NA
## 6 2000  China   2.5
```

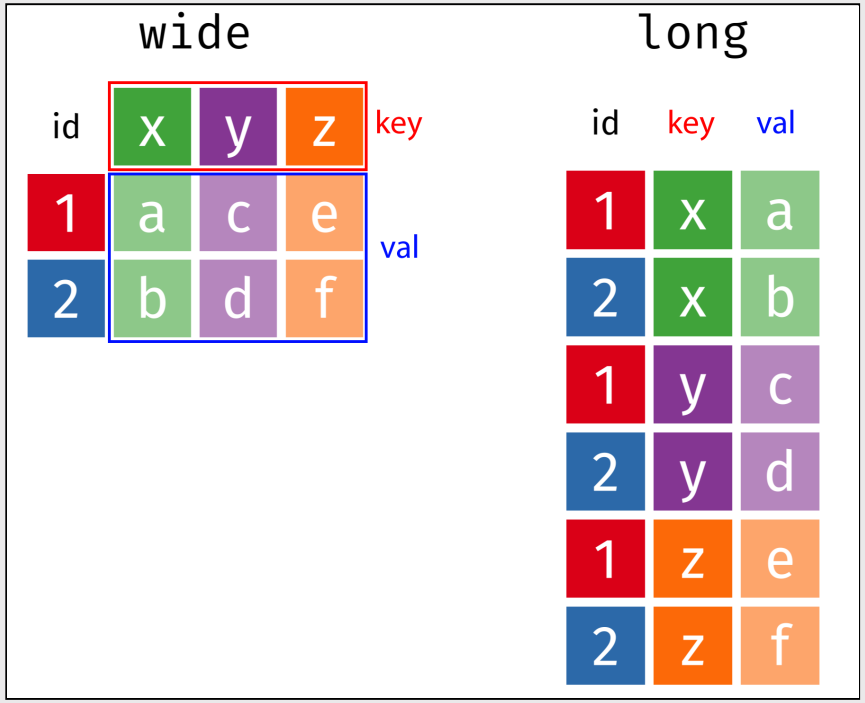
Re-shaping from "wide" to "long" (tidy)

`gather(data, key = "", value = "", ...)`

```
## # A tibble: 6 x 10
##   Year  China Taiwan Japan Malaysia Germany `South Korea` `United States`
##   <chr> <chr> <chr>  <dbl> <chr>    <chr>   <chr>                    <dbl>
## 1 1995  NA    NA      16.4 NA       NA      NA                        34.8
## 2 1996  NA    NA      21.2 NA       NA      NA                        38.8
## 3 1997  NA    NA      35   NA       NA      NA                        51
## 4 1998  NA    NA      49   NA       NA      NA                        53.7
## 5 1999  NA    NA      80   NA       NA      NA                        60.8
## 6 2000  2.5   NA     129.  NA       22.5    NA                        75
## # … with 1 more variable: World <dbl>
```

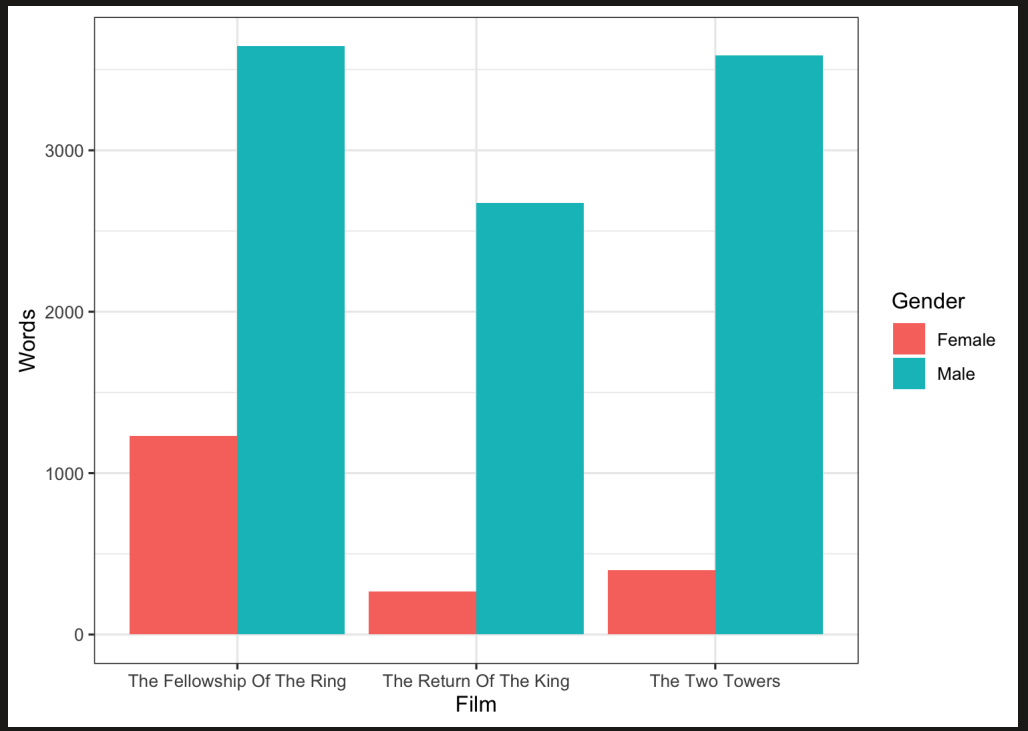
```
pv_cells_long <- pv_cells %>%
    gather(key = 'Country', value = 'nPVCells', China:World)
```

```
## # A tibble: 6 x 3
##   Year  Country nPVCells
##   <chr> <chr>   <chr>
## 1 1995  China   NA
## 2 1996  China   NA
## 3 1997  China   NA
## 4 1998  China   NA
## 5 1999  China   NA
## 6 2000  China   2.5
```



Your turn

1. Read in the `lotr_words.csv` data file.
2. Use `gather()` to "tidy" the data into four columns: `Film`, `Race`, `Gender`, `Words`.
3. Use `write_csv()` and `here()` to save your file at `lotr_words_tidy.csv` in the `data` folder.
4. Use your "tidy" formatted data to create the plot to the right.



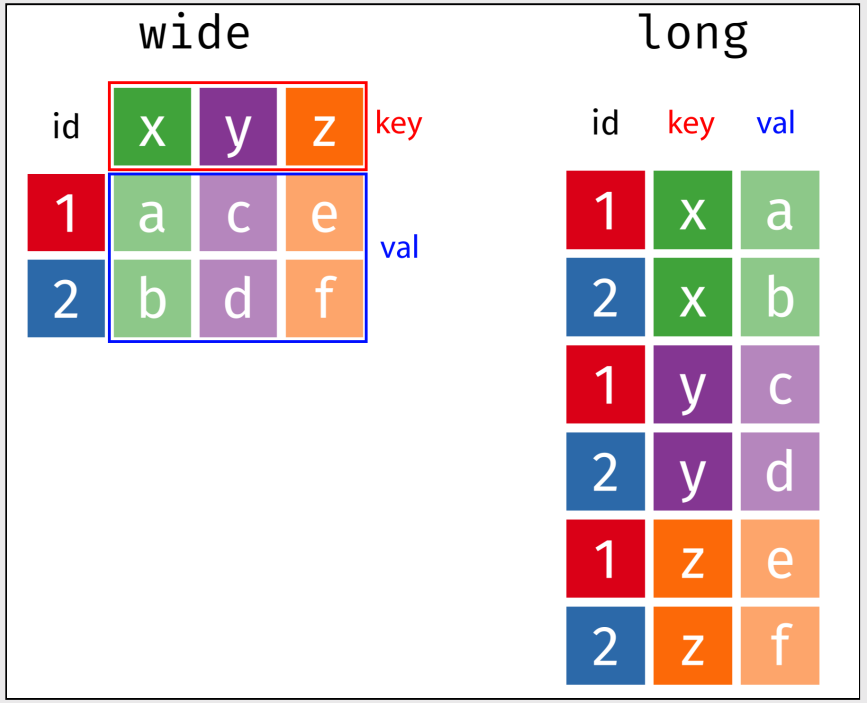
Re-shaping from "long" to "wide" (un-tidy)

```
spread(data, key = "", value = "")
```

```
## # A tibble: 6 x 3
##   Year  Country nPVCells
##   <chr> <chr>   <chr>
## 1 1995  China   NA
## 2 1996  China   NA
## 3 1997  China   NA
## 4 1998  China   NA
## 5 1999  China   NA
## 6 2000  China   2.5
```

```
pv_cells_wide <- pv_cells_long %>%
    spread(key = Country, value = nPVCells)
```

```
## # A tibble: 6 x 10
##   Year  China Germany Japan Malaysia Others `South Korea` Taiwan `United S
##   <chr> <chr> <chr>   <chr> <chr>    <chr>  <chr>         <chr>  <chr>
## 1 1995  NA    NA      16.4  NA       NA     NA            NA     34.75
## 2 1996  NA    NA      21.2  NA       NA     NA            NA     38.85
## 3 1997  NA    NA      35    NA       NA     NA            NA     51
## 4 1998  NA    NA      49    NA       NA     NA            NA     53.7
## 5 1999  NA    NA      80    NA       NA     NA            NA     60.8
## 6 2000  2.5   22.5    128.6 NA       48.20… NA            NA     75
## # … with 1 more variable: World <chr>
```



Your turn

1. Read in the `lotr_words_tidy.csv` data file.
2. Use `spread()` to convert the "tidy" data back into it's untidy format with the columns: `Film`, `Race`, `Female`, `Male`
3. Use your "wide" format data and to compute the percentage of words spoken by Female characters (regardless of race) in each film:

```
## # A tibble: 3 x 4
##   Film                      Female  Male percentFemale
##   <chr>                      <dbl> <dbl>         <dbl>
## 1 The Fellowship Of The Ring   1243  6610         15.8
## 2 The Return Of The King        453  5642          7.43
## 3 The Two Towers                732  6565         10.0
```

Writing a research question

Follow these guidelines.

Your question should be:

- **Clear**: your audience can easily understand its purpose without additional explanation.
- **Focused**: it is narrow enough that it can be addressed thoroughly with the data available and within the limits of the final project report.
- **Concise**: it is expressed in the fewest possible words.
- **Complex**: it is not answerable with a simple "yes" or "no," but rather requires synthesis and analysis of data.
- **Arguable**: its potential answers are open to debate rather than accepted facts (do others care about it?)

Writing a research question

Bad question: Why are social networking sites harmful?

- Unclear: it does not specify *which* social networking sites or state what harm is being caused; assumes that "harm" exists.

Improved question: How are online users experiencing or addressing privacy issues on such social networking sites as Facebook and Twitter?

- Specifies the sites (Facebook and Twitter), type of harm (privacy issues), and who is harmed (online users).

Other good examples: See the Final Project Assignment page