

# Week 1: *Getting Started*

☰ EMSE 4575: Exploratory Data Analysis

👤 John Paul Helveston

📅 January 12, 2021

Faculty trying to finish their courses for January 2021



# It's nice to see your faces 😊



If you're okay with it, please turn on your camera - it creates a more engaging discussion environment and an opportunity for us to get to know each other better.

Fun Zoom backgrounds encouraged 😊

(Your privacy is important, and I understand if you wish to keep cameras off. No pressure.)

# *Week 1: Getting Started*

1. Course Goal
2. Course Introduction
3. Break: Install Stuff
4. Workflow & Reading In Data
5. Data Provenance
6. Tidy Data

# *Week 1: Getting Started*

1. Course Goal
2. Course Introduction
3. Break: Install Stuff
4. Workflow & Reading In Data
5. Data Provenance
6. Tidy Data

# Course 1: Intro to 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 conducting an exploratory data analysis.
- Design principles for visualizing and communicating *information* extracted from data.
- Reproducibility: Reports that contain code, equations, visualizations, and narrative text.

**Class goal:** translate *data* into *information*

# Class goal: translate *data* into *information*

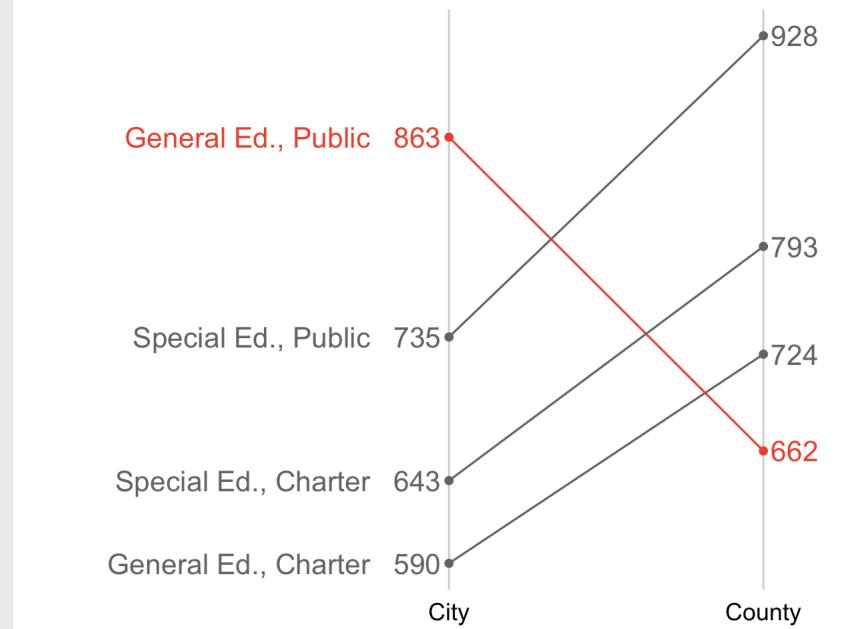
## Data

Average student engagement scores

Class	Type	City	County
Special Ed.	Charter	643	793
Special Ed.	Public	735	928
General Ed.	Charter	590	724
General Ed.	Public	863	662

## Information

Students in public, general education classes in county schools have surprisingly low engagement



# Data exploration: an iterative process

Encode data:

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

```
#>   City County           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:

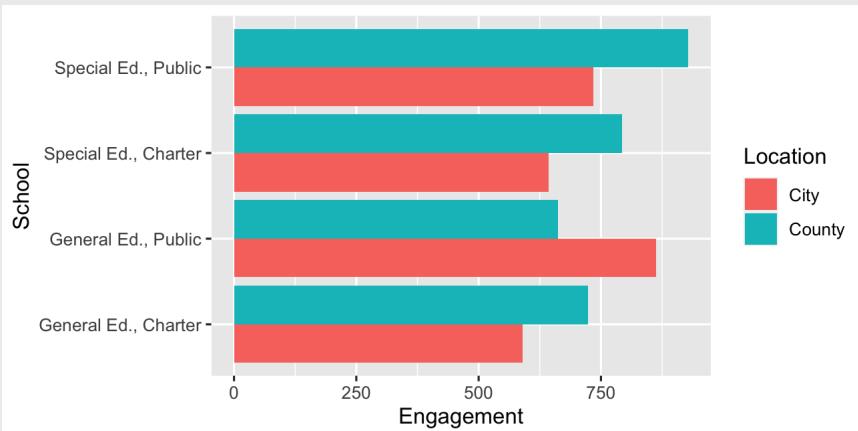
```
engagement_data <- engagement_data %>%  
  gather(Location, Engagement, City:County) %>%  
  mutate(Location = fct_relevel(  
    Location, c('City', 'County')))  
engagement_data
```

```
#>           School Location Engagement  
#> 1 Special Ed., Charter     City      643  
#> 2 Special Ed., Public     City      735  
#> 3 General Ed., Charter   City      590  
#> 4 General Ed., Public    City      863  
#> 5 Special Ed., Charter   County    793  
#> 6 Special Ed., Public    County    928  
#> 7 General Ed., Charter   County    724  
#> 8 General Ed., Public    County    662
```

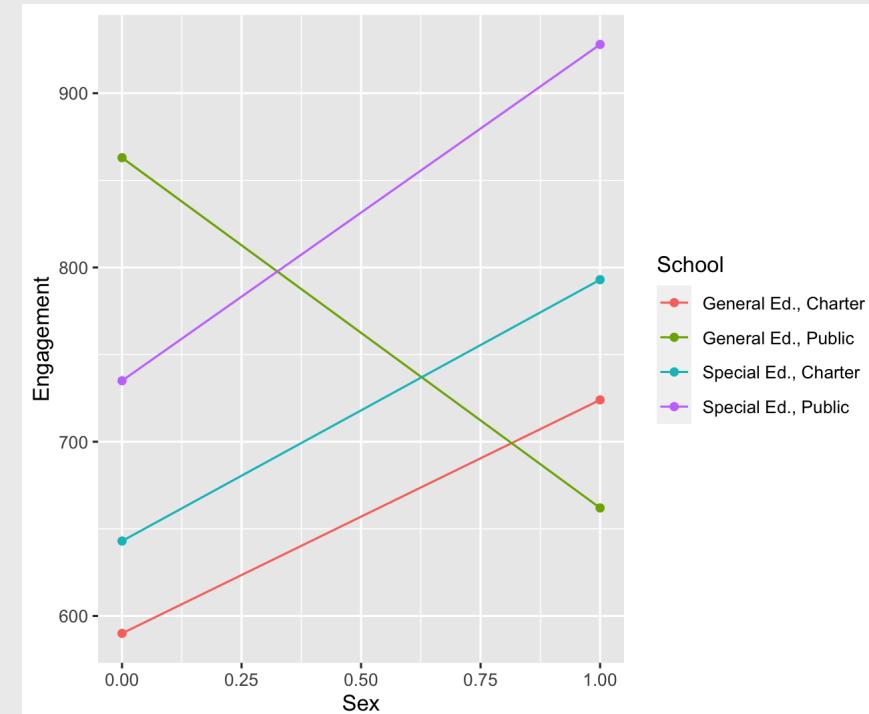
# Data exploration: an iterative process

Initial exploratory plotting:

```
engagement_data %>%
  ggplot() +
  geom_col(aes(x = Engagement, y = School,
               fill = Location),
            position = 'dodge')
```

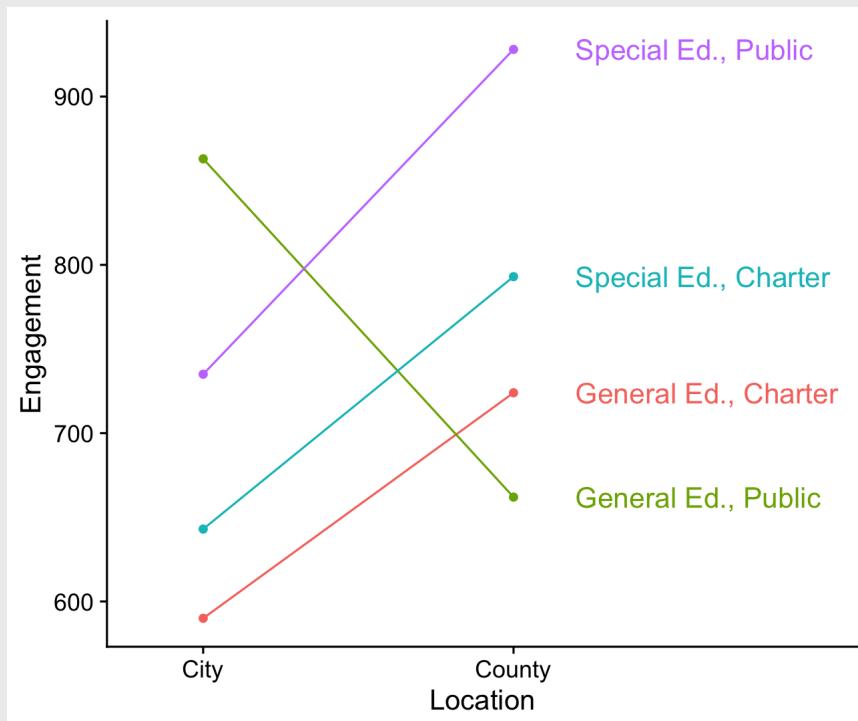


More exploratory plotting:  
highlight difference

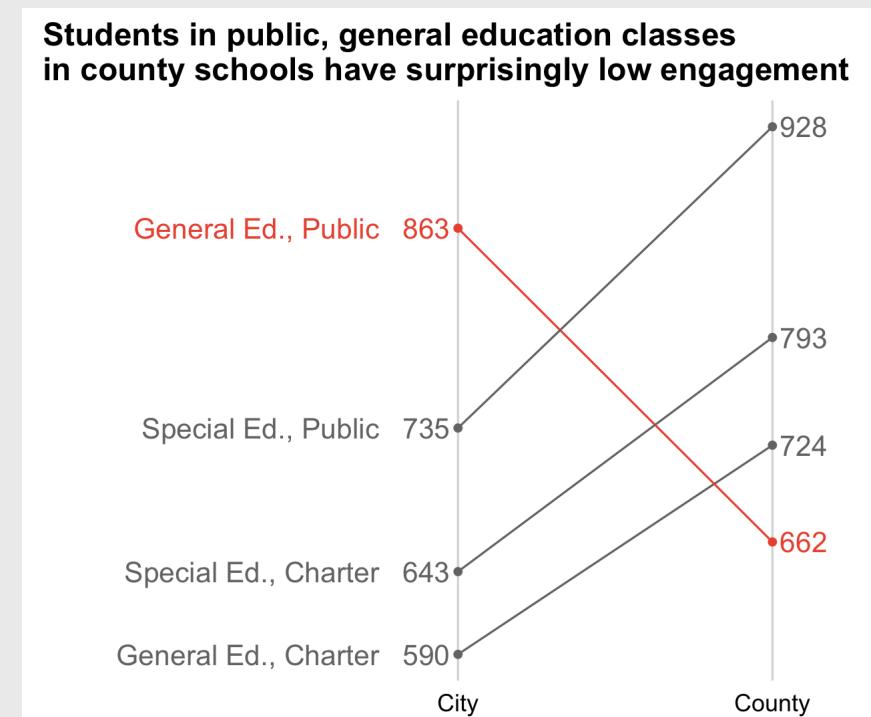


# Data exploration: an iterative process

Directly label figure:



Remove unnecessary axes, change colors, fix labels:



# A fully reproducible analysis

Code Plot

```
data <- data.frame(
  City    = c(643, 735, 590, 863),
  County  = 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(Location, Engagement, City:County) %>%
  mutate(
    Location = fct_relevel(Location, c('City', 'County')),
    Highlight = as.factor(Highlight),
    x = ifelse(Location == 'County', 1, 0))
```

```
plot <- ggplot(data, 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('Students in public, general education classes\n',
                     'in county schools have surprisingly low engagement')) +
  scale_x_continuous(limits = c(-1.2, 1.2), labels = c('City', 'County'),
                     breaks = c(0, 1)) +
  geom_text_repel(aes(label = Engagement, color = as.factor(Highlight)),
                 data      = subset(engagement, Location == 'County'),
                 size     = 5,
                 nudge_x = 0.1,
                 segment.color = NA) +
  geom_text_repel(aes(label = Engagement, color = as.factor(Highlight)),
                 data      = subset(engagement, Location == 'City'),
                 size     = 5,
                 nudge_x = -0.1,
                 segment.color = NA) +
  geom_text_repel(aes(label = School, color = as.factor(Highlight)),
                 data      = subset(engagement, Location == 'City'),
                 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')
```

# *Week 1: Getting Started*

1. Course Goal
2. Course Introduction
3. Break: Install Stuff
4. Workflow & Reading In Data
5. Data Provenance
6. Tidy Data

# Meet your instructor!



John Helveston, Ph.D.

- 2018 - Present Assistant Professor, Engineering Management & Systems Engineering
- 2016-2018 Postdoc at [Institute for Sustainable Energy](#), Boston University
- 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
- Website: [www.jhelvy.com](http://www.jhelvy.com)

# Meet your tutors!



**Saurav Pantha** (aka "The Firefighter")

- Graduate Assistant (GA)
- Masters student in EMSE

# Meet your tutors!



**Jennifer Kim** (aka "The Monitor")

- Learning Assistant (LA)
- EMSE Junior & P4A alumni

# Prerequisites

## EMSE 4574: Intro to Programming for Analytics

You should be able to:

- Use RStudio to write basic R commands.
- Know the distinctions between different R operators and data types, including numeric, string, and logical data.
- Use **tidyverse** functions to wrangle and manipulate data in R.
- Use the **ggplot2** library to create plots in R.

 [Check out R for Analytics Primer](#)

# Course website

🌐 Everything you need will be on the course website:  
<https://eda.seas.gwu.edu/2021-Spring/>

📅 The [schedule](#) is the best starting point

# Quizzes

🕒 In class every other week-ish (5 total, lowest dropped)

⌚ ~5 minutes

☰ Example quiz

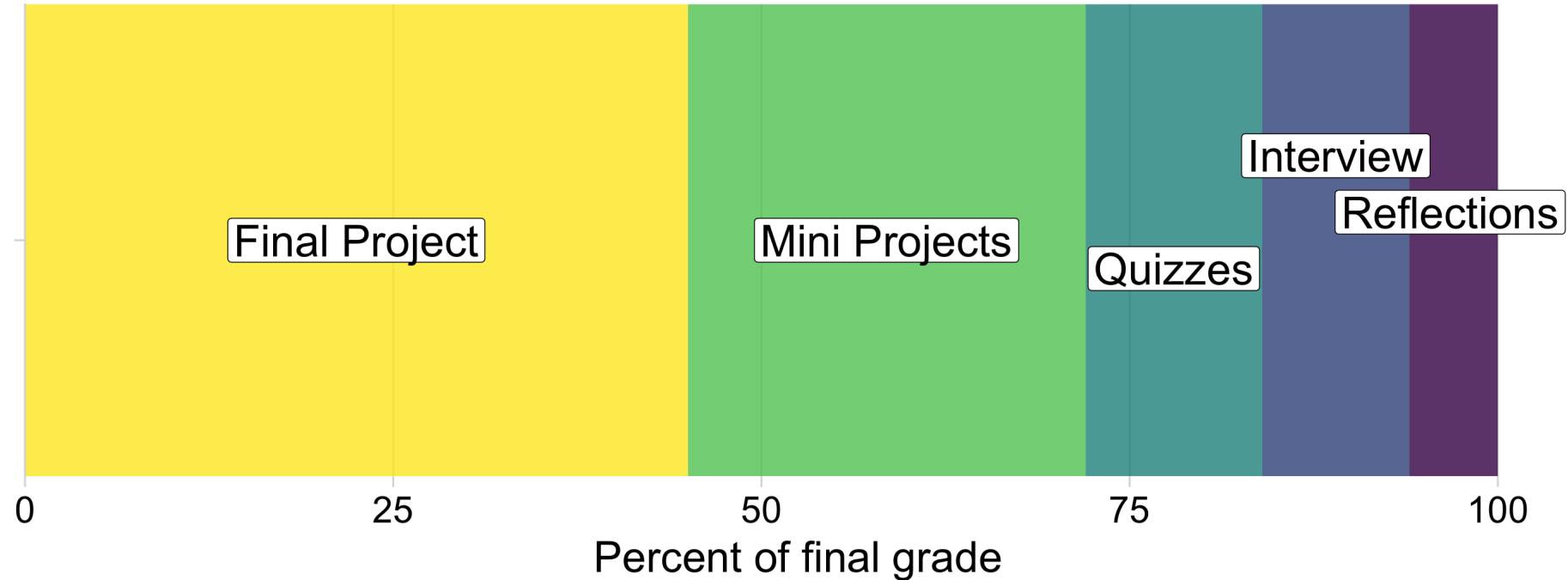
**Why quiz at all?** The "retrieval effect" - basically, you have to *practice* remembering things, otherwise your brain won't remember them (see the book "["Make It Stick: The Science of Successful Learning"](#)")

# Assignments

- 1)  Weekly "reflections" on **readings**
- 2)  3 Mini Projects (due 2 weeks from date assigned)
- 3)  **Final Project** (Teams of 2 - 3 students)

Item	Due Date
Proposal	March 12
Progress Report	April 16
Final Report	April 30
Presentation	May 03
Interview	Exam week

# Grades



# Grades

Item	Weight	Notes
Reflections	6 %	Weekly assignment (12 x 0.5%)
Quizzes	12 %	5 quizzes, lowest dropped
Mini Project 1	9 %	Individual projects
Mini Project 2	9 %	
Mini Project 3	9 %	
Final Project Proposal	10 %	Teams of 2-3 students
Final Project Progress Report	10 %	
Final Project Report	15 %	
Final Project Presentation	10 %	
Final Interview	10 %	Individual interview about your project

# Course policies

- BE NICE
- BE HONEST
- DON'T CHEAT

Copying is good, stealing is bad

"Plagiarism is trying to pass someone else's work off as your own. Copying is about reverse-engineering."

-- Austin Kleon, from [Steal Like An Artist](#)

# Late submissions

- **5** late days - use them anytime, no questions asked
- No more than **2** late days on any one assignment
- Contact me for special cases

# How to succeed in this class

- 👤 Participate during class!
- ☒ Start assignments early and **read carefully!**
- 📖 Actually read (before class)!
- 🛌 Get sleep and take breaks often!
- 🙋‍♂️ Ask for help!

# Getting Help

❖ Use [Slack](#) to ask questions.

❑ Meet with your tutors

❑ Schedule a meeting w/[Prof. Helveston](#):

- Mondays from 8:00-5:00pm
- Wednesdays from 3:20-5:00pm
- Thursdays from 12:00-5:00pm

[GW Coders](#)

## Course Software

 **Slack**: See bb for link to join;  
install on phone and **turn notifications on!**

 **R** & **RStudio** (Install both)

 Install **Cisco AnyConnect VPN Client** to use RStudio in  
the cloud: <https://rstudio.seas.gwu.edu/>

 **DataCamp**: sign up with your **@gwu.edu** email

*Break*

Install Stuff

05 : 00

# *Week 1: Getting Started*

1. Course Goal
2. Course Introduction
3. Break: Install Stuff
4. Workflow & Reading In Data
5. Data Provenance
6. Tidy Data

# Workflow for reading in data

- 1) Use R Projects (.Rproj files) to organize your analysis - **don't double-click .R files!**



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

```
path <- here::here("folder", "file.csv")
```

- 3) Import data with these functions:

File type	Function	Library
.csv	<code>read_csv()</code>	<code>readr</code>
.txt	<code>read.table()</code>	<code>utils</code>
.xlsx	<code>read_excel()</code>	<code>readxl</code>

# Importing Comma Separated Values (.csv)

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

```
library(tidyverse)
library(here)

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

head(milk_production)
```

```
#> # A tibble: 6 x 4
#>   region      state        year milk_produced
#>   <chr>       <chr>     <dbl>        <dbl>
#> 1 Northeast   Maine      1970    619000000
#> 2 Northeast   New Hampshire 1970    356000000
#> 3 Northeast   Vermont    1970    1970000000
#> 4 Northeast   Massachusetts 1970    658000000
#> 5 Northeast   Rhode Island 1970     75000000
#> 6 Northeast   Connecticut 1970    661000000
```

# Importing Text Files (.txt)

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

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

head(global_temps)
```

```
#>      V1      V2      V3
#> 1 1880 -0.18 -0.11
#> 2 1881 -0.10 -0.14
#> 3 1882 -0.11 -0.17
#> 4 1883 -0.19 -0.21
#> 5 1884 -0.28 -0.24
#> 6 1885 -0.31 -0.26
```

# Importing Text Files (.txt)

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

```
txtPath <- here('data', 'nasa_global_temps.txt')
global_temps <- read.table(txtPath, skip = 5, header = FALSE)
names(global_temps) <- c('year', 'no_smoothing', 'loess') # Add header

head(global_temps)
```

```
#>   year no_smoothing loess
#> 1 1880      -0.18 -0.11
#> 2 1881      -0.10 -0.14
#> 3 1882      -0.11 -0.17
#> 4 1883      -0.19 -0.21
#> 5 1884      -0.28 -0.24
#> 6 1885      -0.31 -0.26
```

# Importing Excel Files (.xlsx)

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)
```

```
glimpse(pv_cells)
```

```
#> Rows: 25
#> Columns: 10
#> 
#> $ Year <chr> NA, NA, "1995", "1996", "1997", "1998", "1999", "2000", "2001", "2002"
#> $ China <chr> "Megawatts", NA, "NA", "NA", "NA", "NA", "NA", "2.5", "3", "10", "13"
#> $ Taiwan <chr> NA, NA, "NA", "NA", "NA", "NA", "NA", "NA", "3.5", "8", "17", "39.299
#> $ Japan <dbl> NA, NA, 16.4, 21.2, 35.0, 49.0, 80.0, 128.6, 171.2, 251.1, 363.9, 601
#> $ Malaysia <chr> NA, NA, "NA", "NA", "NA", "NA", "NA", "NA", "0", "0", "0", "0", "0",
#> $ Germany <chr> NA, NA, "NA", "NA", "NA", "NA", "NA", "NA", "22.5", "23.5", "55", "121.5",
#> $ `South Korea` <chr> NA, NA, "NA", "NA", "NA", "NA", "NA", "NA", "0", "0", "0", "0", "0", "5.3"
#> $ `United States` <dbl> NA, NA, 34.7500, 38.8500, 51.0000, 53.7000, 60.8000, 75.0000, 100.300
#> $ Others <chr> NA, NA, "NA", "NA", "NA", "NA", "NA", "NA", "48.20000000000017", "69.80000
#> $ World <dbl> NA, NA, 77.600, 88.600, 125.800, 154.900, 201.300, 276.800, 371.300
```

# Importing Excel Files (.xlsx)

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) %>%
  mutate(Year = as.numeric(Year)) %>% # Convert "non-years" to NA
  filter(!is.na(Year)) # Drop NA rows in Year
```

```
glimpse(pv_cells)
```

```
#> Rows: 19
#> Columns: 10
#> $ Year              <dbl> 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013
#> $ China             <chr> "NA", "NA", "NA", "NA", "NA", "2.5", "3", "10", "13", "40", "128.3000000000001", "341.0", "341.0", "341.0", "341.0", "341.0", "341.0", "341.0", "341.0", "341.0", "341.0"
#> $ Taiwan            <chr> "NA", "NA", "NA", "NA", "NA", "NA", "3.5", "8", "17", "39.29999999999997", "88", "169.0", "169.0", "169.0", "169.0", "169.0", "169.0", "169.0", "169.0", "169.0", "169.0"
#> $ Japan              <dbl> 16.4, 21.2, 35.0, 49.0, 80.0, 128.6, 171.2, 251.1, 363.9, 601.5, 833.0, 926.4, 937.5, 950.0, 960.0, 970.0, 980.0, 990.0, 1000.0, 1010.0
#> $ Malaysia           <chr> "NA", "NA", "NA", "NA", "NA", "NA", "0", "0", "0", "0", "0", "100.1", "397.9", "397.9", "397.9", "397.9", "397.9", "397.9", "397.9", "397.9", "397.9", "397.9"
#> $ Germany            <chr> "NA", "NA", "NA", "NA", "NA", "NA", "22.5", "23.5", "55", "121.5", "193", "339", "469.1", "469.1", "469.1", "469.1", "469.1", "469.1", "469.1", "469.1", "469.1", "469.1"
#> $ `South Korea`      <chr> "NA", "NA", "NA", "NA", "NA", "NA", "0", "0", "0", "5.3", "13", "31.8839359056746", "31.8839359056746", "31.8839359056746", "31.8839359056746", "31.8839359056746", "31.8839359056746", "31.8839359056746", "31.8839359056746", "31.8839359056746", "31.8839359056746"
#> $ `United States`    <dbl> 34.7500, 38.8500, 51.0000, 53.7000, 60.8000, 75.0000, 100.3000, 120.6000, 103.0000, 115.0000, 130.0000, 145.0000, 160.0000, 175.0000, 190.0000, 205.0000, 220.0000, 235.0000, 250.0000, 265.0000, 280.0000, 295.0000, 310.0000, 325.0000, 340.0000, 355.0000, 370.0000, 385.0000, 400.0000, 415.0000, 430.0000, 445.0000, 460.0000, 475.0000, 490.0000, 505.0000, 520.0000, 535.0000, 550.0000, 565.0000, 580.0000, 595.0000, 610.0000, 625.0000, 640.0000, 655.0000, 670.0000, 685.0000, 700.0000, 715.0000, 730.0000, 745.0000, 760.0000, 775.0000, 790.0000, 805.0000, 820.0000, 835.0000, 850.0000, 865.0000, 880.0000, 895.0000, 910.0000, 925.0000, 940.0000, 955.0000, 970.0000, 985.0000, 1000.0000, 1015.0000, 1030.0000, 1045.0000, 1060.0000, 1075.0000, 1090.0000, 1105.0000, 1120.0000, 1135.0000, 1150.0000, 1165.0000, 1180.0000, 1195.0000, 1210.0000, 1225.0000, 1240.0000, 1255.0000, 1270.0000, 1285.0000, 1300.0000, 1315.0000, 1330.0000, 1345.0000, 1360.0000, 1375.0000, 1390.0000, 1405.0000, 1420.0000, 1435.0000, 1450.0000, 1465.0000, 1480.0000, 1495.0000, 1510.0000, 1525.0000, 1540.0000, 1555.0000, 1570.0000, 1585.0000, 1600.0000, 1615.0000, 1630.0000, 1645.0000, 1660.0000, 1675.0000, 1690.0000, 1705.0000, 1720.0000, 1735.0000, 1750.0000, 1765.0000, 1780.0000, 1795.0000, 1810.0000, 1825.0000, 1840.0000, 1855.0000, 1870.0000, 1885.0000, 1900.0000, 1915.0000, 1930.0000, 1945.0000, 1960.0000, 1975.0000, 1990.0000, 2005.0000, 2020.0000, 2035.0000, 2050.0000, 2065.0000, 2080.0000, 2095.0000, 2110.0000, 2125.0000, 2140.0000, 2155.0000, 2170.0000, 2185.0000, 2200.0000, 2215.0000, 2230.0000, 2245.0000, 2260.0000, 2275.0000, 2290.0000, 2305.0000, 2320.0000, 2335.0000, 2350.0000, 2365.0000, 2380.0000, 2395.0000, 2410.0000, 2425.0000, 2440.0000, 2455.0000, 2470.0000, 2485.0000, 2500.0000, 2515.0000, 2530.0000, 2545.0000, 2560.0000, 2575.0000, 2590.0000, 2605.0000, 2620.0000, 2635.0000, 2650.0000, 2665.0000, 2680.0000, 2695.0000, 2710.0000, 2725.0000, 2740.0000, 2755.0000, 2770.0000, 2785.0000, 2800.0000, 2815.0000, 2830.0000, 2845.0000, 2860.0000, 2875.0000, 2890.0000, 2905.0000, 2920.0000, 2935.0000, 2950.0000, 2965.0000, 2980.0000, 2995.0000, 3010.0000, 3025.0000, 3040.0000, 3055.0000, 3070.0000, 3085.0000, 3100.0000, 3115.0000, 3130.0000, 3145.0000, 3160.0000, 3175.0000, 3190.0000, 3205.0000, 3220.0000, 3235.0000, 3250.0000, 3265.0000, 3280.0000, 3295.0000, 3310.0000, 3325.0000, 3340.0000, 3355.0000, 3370.0000, 3385.0000, 3400.0000, 3415.0000, 3430.0000, 3445.0000, 3460.0000, 3475.0000, 3490.0000, 3505.0000, 3520.0000, 3535.0000, 3550.0000, 3565.0000, 3580.0000, 3595.0000, 3610.0000, 3625.0000, 3640.0000, 3655.0000, 3670.0000, 3685.0000, 3700.0000, 3715.0000, 3730.0000, 3745.0000, 3760.0000, 3775.0000, 3790.0000, 3805.0000, 3820.0000, 3835.0000, 3850.0000, 3865.0000, 3880.0000, 3895.0000, 3910.0000, 3925.0000, 3940.0000, 3955.0000, 3970.0000, 3985.0000, 3995.0000, 4010.0000, 4025.0000, 4040.0000, 4055.0000, 4070.0000, 4085.0000, 4100.0000, 4115.0000, 4130.0000, 4145.0000, 4160.0000, 4175.0000, 4190.0000, 4205.0000, 4220.0000, 4235.0000, 4250.0000, 4265.0000, 4280.0000, 4295.0000, 4310.0000, 4325.0000, 4340.0000, 4355.0000, 4370.0000, 4385.0000, 4400.0000, 4415.0000, 4430.0000, 4445.0000, 4460.0000, 4475.0000, 4490.0000, 4505.0000, 4520.0000, 4535.0000, 4550.0000, 4565.0000, 4580.0000, 4595.0000, 4610.0000, 4625.0000, 4640.0000, 4655.0000, 4670.0000, 4685.0000, 4700.0000, 4715.0000, 4730.0000, 4745.0000, 4760.0000, 4775.0000, 4790.0000, 4805.0000, 4820.0000, 4835.0000, 4850.0000, 4865.0000, 4880.0000, 4895.0000, 4910.0000, 4925.0000, 4940.0000, 4955.0000, 4970.0000, 4985.0000, 4995.0000, 5010.0000, 5025.0000, 5040.0000, 5055.0000, 5070.0000, 5085.0000, 5100.0000, 5115.0000, 5130.0000, 5145.0000, 5160.0000, 5175.0000, 5190.0000, 5205.0000, 5220.0000, 5235.0000, 5250.0000, 5265.0000, 5280.0000, 5295.0000, 5310.0000, 5325.0000, 5340.0000, 5355.0000, 5370.0000, 5385.0000, 5400.0000, 5415.0000, 5430.0000, 5445.0000, 5460.0000, 5475.0000, 5490.0000, 5505.0000, 5520.0000, 5535.0000, 5550.0000, 5565.0000, 5580.0000, 5595.0000, 5610.0000, 5625.0000, 5640.0000, 5655.0000, 5670.0000, 5685.0000, 5700.0000, 5715.0000, 5730.0000, 5745.0000, 5760.0000, 5775.0000, 5790.0000, 5805.0000, 5820.0000, 5835.0000, 5850.0000, 5865.0000, 5880.0000, 5895.0000, 5910.0000, 5925.0000, 5940.0000, 5955.0000, 5970.0000, 5985.0000, 5995.0000, 6010.0000, 6025.0000, 6040.0000, 6055.0000, 6070.0000, 6085.0000, 6100.0000, 6115.0000, 6130.0000, 6145.0000, 6160.0000, 6175.0000, 6190.0000, 6205.0000, 6220.0000, 6235.0000, 6250.0000, 6265.0000, 6280.0000, 6295.0000, 6310.0000, 6325.0000, 6340.0000, 6355.0000, 6370.0000, 6385.0000, 6400.0000, 6415.0000, 6430.0000, 6445.0000, 6460.0000, 6475.0000, 6490.0000, 6505.0000, 6520.0000, 6535.0000, 6550.0000, 6565.0000, 6580.0000, 6595.0000, 6610.0000, 6625.0000, 6640.0000, 6655.0000, 6670.0000, 6685.0000, 6695.0000, 6710.0000, 6725.0000, 6740.0000, 6755.0000, 6770.0000, 6785.0000, 6795.0000, 6810.0000, 6825.0000, 6840.0000, 6855.0000, 6870.0000, 6885.0000, 6895.0000, 6910.0000, 6925.0000, 6940.0000, 6955.0000, 6970.0000, 6985.0000, 6995.0000, 7010.0000, 7025.0000, 7040.0000, 7055.0000, 7070.0000, 7085.0000, 7095.0000, 7110.0000, 7125.0000, 7140.0000, 7155.0000, 7170.0000, 7185.0000, 7195.0000, 7210.0000, 7225.0000, 7240.0000, 7255.0000, 7270.0000, 7285.0000, 7295.0000, 7310.0000, 7325.0000, 7340.0000, 7355.0000, 7370.0000, 7385.0000, 7395.0000, 7410.0000, 7425.0000, 7440.0000, 7455.0000, 7470.0000, 7485.0000, 7495.0000, 7510.0000, 7525.0000, 7540.0000, 7555.0000, 7570.0000, 7585.0000, 7595.0000, 7610.0000, 7625.0000, 7640.0000, 7655.0000, 7670.0000, 7685.0000, 7695.0000, 7710.0000, 7725.0000, 7740.0000, 7755.0000, 7770.0000, 7785.0000, 7795.0000, 7810.0000, 7825.0000, 7840.0000, 7855.0000, 7870.0000, 7885.0000, 7895.0000, 7910.0000, 7925.0000, 7940.0000, 7955.0000, 7970.0000, 7985.0000, 7995.0000, 8010.0000, 8025.0000, 8040.0000, 8055.0000, 8070.0000, 8085.0000, 8095.0000, 8110.0000, 8125.0000, 8140.0000, 8155.0000, 8170.0000, 8185.0000, 8195.0000, 8210.0000, 8225.0000, 8240.0000, 8255.0000, 8270.0000, 8285.0000, 8295.0000, 8310.0000, 8325.0000, 8340.0000, 8355.0000, 8370.0000, 8385.0000, 8395.0000, 8410.0000, 8425.0000, 8440.0000, 8455.0000, 8470.0000, 8485.0000, 8495.0000, 8510.0000, 8525.0000, 8540.0000, 8555.0000, 8570.0000, 8585.0000, 8595.0000, 8610.0000, 8625.0000, 8640.0000, 8655.0000, 8670.0000, 8685.0000, 8695.0000, 8710.0000, 8725.0000, 8740.0000, 8755.0000, 8770.0000, 8785.0000, 8795.0000, 8810.0000, 8825.0000, 8840.0000, 8855.0000, 8870.0000, 8885.0000, 8895.0000, 8910.0000, 8925.0000, 8940.0000, 8955.0000, 8970.0000, 8985.0000, 8995.0000, 9010.0000, 9025.0000, 9040.0000, 9055.0000, 9070.0000, 9085.0000, 9095.0000, 9110.0000, 9125.0000, 9140.0000, 9155.0000, 9170.0000, 9185.0000, 9195.0000, 9210.0000, 9225.0000, 9240.0000, 9255.0000, 9270.0000, 9285.0000, 9295.0000, 9310.0000, 9325.0000, 9340.0000, 9355.0000, 9370.0000, 9385.0000, 9395.0000, 9410.0000, 9425.0000, 9440.0000, 9455.0000, 9470.0000, 9485.0000, 9495.0000, 9510.0000, 9525.0000, 9540.0000, 9555.0000, 9570.0000, 9585.0000, 9595.0000, 9610.0000, 9625.0000, 9640.0000, 9655.0000, 9670.0000, 9685.0000, 9695.0000, 9710.0000, 9725.0000, 9740.0000, 9755.0000, 9770.0000, 9785.0000, 9795.0000, 9810.0000, 9825.0000, 9840.0000, 9855.0000, 9870.0000, 9885.0000, 9895.0000, 9910.0000, 9925.0000, 9940.0000, 9955.0000, 9970.0000, 9985.0000, 9995.0000, 10010.0000, 10025.0000, 10040.0000, 10055.0000, 10070.0000, 10085.0000, 10095.0000, 10110.0000, 10125.0000, 10140.0000, 10155.0000, 10170.0000, 10185.0000, 10195.0000, 10210.0000, 10225.0000, 10240.0000, 10255.0000, 10270.0000, 10285.0000, 10295.0000, 10310.0000, 10325.0000, 10340.0000, 10355.0000, 10370.0000, 10385.0000, 10395.0000, 10410.0000, 10425.0000, 10440.0000, 10455.0000, 10470.0000, 10485.0000, 10495.0000, 10510.0000, 10525.0000, 10540.0000, 10555.0000, 10570.0000, 10585.0000, 10595.0000, 10610.0000, 10625.0000, 10640.0000, 10655.0000, 10670.0000, 10685.0000, 10695.0000, 10710.0000, 10725.0000, 10740.0000, 10755.0000, 10770.0000, 10785.0000, 10795.0000, 10810.0000, 10825.0000, 10840.0000, 10855.0000, 10870.0000, 10885.0000, 10895.0000, 10910.0000, 10925.0000, 10940.0000, 10955.0000, 10970.0000, 10985.0000, 10995.0000, 11010.0000, 11025.0000, 11040.0000, 11055.0000, 11070.0000, 11085.0000, 11095.0000, 11110.0000, 11125.0000, 11140.0000, 11155.0000, 11170.0000, 11185.0000, 11195.0000, 11210.0000, 11225.0000, 11240.0000, 11255.0000, 11270.0000, 11285.0000, 11295.0000, 11310.0000, 11325.0000, 11340.0000, 11355.0000, 11370.0000, 11385.0000, 11395.0000, 11410.0000, 11425.0000, 11440.0000, 11455.0000, 11470.0000, 11485.0000, 11495.0000, 11510.0000, 11525.0000, 11540.0000, 11555.0000, 11570.0000, 11585.0000, 11595.0000, 11610.0000, 11625.0000, 11640.0000, 11655.0000, 11670.0000, 11685.0000, 11695.0000, 11710.0000, 11725.0000, 11740.0000, 11755.0000, 11770.0000, 11785.0000, 11795.0000, 11810.0000, 11825.0000, 11840.0000, 11855.0000, 11870.0000, 11885.0000, 11895.0000, 11910.0000, 11925.0000, 11940.0000, 11955.0000, 11970.0000, 11985.0000, 11995.0000, 12010.0000, 12025.0000, 12040.0000, 12055.0000, 12070.0000, 12085.0000, 12095.0000, 12110.0000, 12125.0000, 12140.0000, 12155.0000, 12170.0000, 12185.0000, 12195.0000, 12210.0000, 12225.0000, 12240.0000, 12255.0000, 12270.0000, 12285.0000, 12295.0000, 12310.0000, 12325.0000, 12340.0000, 12355.0000, 12370.0000, 12385.0000, 12395.0000, 12410.0000, 12425.0000, 12440.0000, 12455.0000, 12470.0000, 12485.0000, 12495.0000, 12510.0000, 12525.0000, 12540.0000, 12555.0000, 12570.0000, 12585.0000, 12595.0000, 12610.0000, 12625.0000, 12640.0000, 12655.0000, 12670.0000, 12685.0000, 12695.0000, 12710.0000, 12725.0000, 12740.0000, 12755.0000, 12770.0000, 12785.0000, 12795.0000, 12810.0000, 12825.0000, 12840.0000, 12855.0000, 12870.0000, 12885.0000, 12895.0000, 12910.0000, 12925.0000, 12940.0000, 12955.0000, 12970.0000, 12985.0000, 12995.0000, 13010.0000, 13025.0000, 13040.0000, 13055.0000, 13070.0000, 13085.0000, 13095.0000, 13110.0000, 13125.0000, 13140.0000, 13155.0000, 13170.0000, 13185.0000, 13195.0000, 13210.0000, 13225.0000, 13240.0000, 13255.0000, 13270.0000, 13285.0000, 
```

# Your turn

10:00

Download [today's class notes](#)

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

- `lotr_words.csv`
- `north_america_bear_killings.txt`
- `uspto_clean_energy_patents.xlsx`

# *Week 1: Getting Started*

1. Course Goal
2. Course Introduction
3. Break: Install Stuff
4. Workflow & Reading In Data
5. Data Provenance
6. Tidy Data

# 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?
- Is this the *right* data to answer your question?

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

# Q Document your source like a museum curator

**Example:** View `README.md` file in the `data` folder

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

- The name of the file you are describing.
- The date you downloaded it.
- The original name of the downloaded file (in case you renamed it).
- The url to the site you downloaded it from.
- The source of the *original* data (sometimes different from the site you downloaded it from).
- A short description of the data, maybe how they were collected (if available).
- A dictionary for the data (e.g. a simple markdown table describing each variable).

# Your turn

10:00

Documentation in the "data/README.md" file is missing for the following data sets:

- `wildlife_impacts.csv`: [source](#) (Breakout Rooms 1 & 2)
- `north_america_bear_killings.txt`: [source](#) (Breakout Rooms 3 & 4)
- `uspto_clean_energy_patents.xlsx`: [source](#) (Breakout Rooms 5 & 6)

Go to the above sites and add the following information to the "data/README.md" file:

- The name of the downloaded file.
- The web address to the site you downloaded the data from.
- The source of the *original* data (if different from the website).
- A short description of the data and how they were collected.
- A dictionary for the data (hint: the site might already have this!).

# *Week 1: Getting Started*

1. Course Goal
2. Course Introduction
3. Break: Install Stuff
4. Workflow & Reading In Data
5. Data Provenance
6. Tidy Data

# Variables, values, and observations

- **Variable**: Something you can measure
- **Value**: The measurement of a variable
- **Observation**: A set of associated measurements across different variables

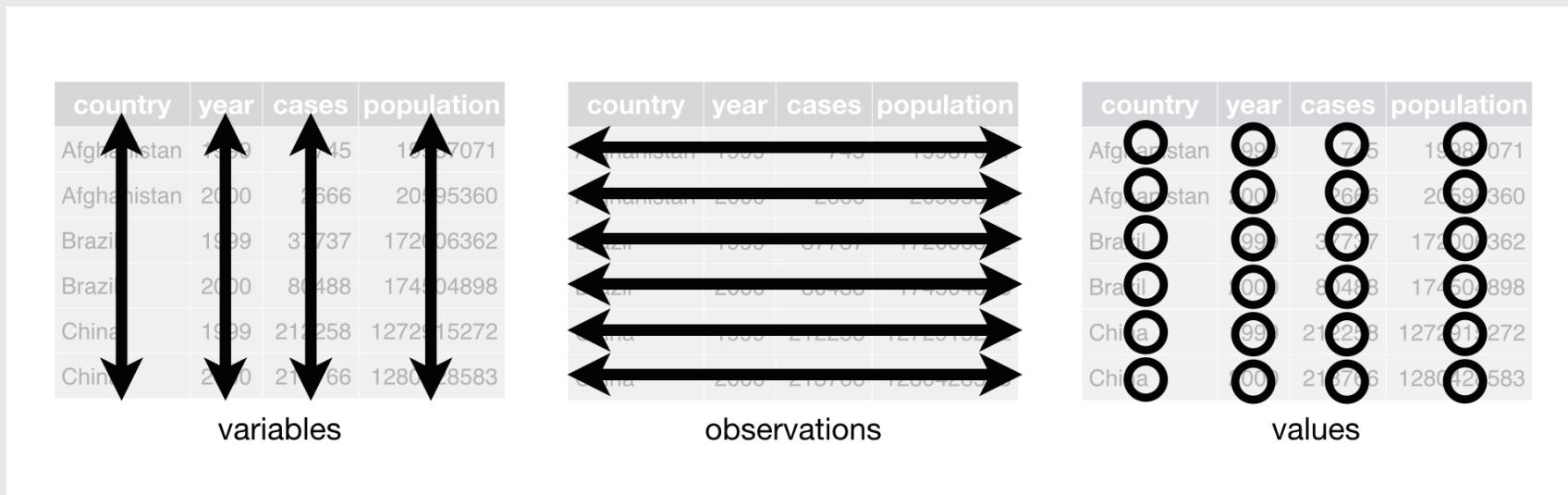
```
head(fed_spend_long)
```

```
#> # A tibble: 6 × 3
#>   department    year rd_budget_mil
#>   <chr>        <dbl>      <dbl>
#> 1 DOD          1976      35696
#> 2 NASA         1976      12513
#> 3 DOE          1976      10882
#> 4 HHS          1976      9226
#> 5 NIH          1976      8025
#> 6 NSF          1976      2372
```

# 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

```
#> # A tibble: 6 x 3
#>   department  year rd_budget_mil
#>   <chr>       <dbl>        <dbl>
#> 1 DOD         1976      35696
#> 2 NASA        1976      12513
#> 3 DOE          1976      10882
#> 4 HHS          1976      9226
#> 5 NIH          1976      8025
#> 6 NSF          1976      2372
```

country	year	cases	population
Afghanistan	1990	745	1637071
Afghanistan	2000	2666	20995360
Brazil	1999	37737	172006362
Brazil	2000	80488	17404898
China	1999	212258	1272015272
China	2000	21666	1280428583

variables

country	year	cases	population
Afghanistan	1990	745	1637071
Afghanistan	2000	2666	20995360
Brazil	1999	37737	172006362
Brazil	2000	80488	17404898
China	1999	212258	1272015272
China	2000	21666	1280428583

observations

country	year	cases	population
Afghanistan	1990	745	1637071
Afghanistan	2000	2666	20995360
Brazil	1999	37737	172006362
Brazil	2000	80488	17404898
China	1999	212258	1272015272
China	2000	21666	1280428583

values

# Tidy ("long")

```
head(fed_spend_long)
```

```
#> # A tibble: 6 x 3
#>   department  year rd_budget_mil
#>   <chr>      <dbl>      <dbl>
#> 1 DOD        1976      35696
#> 2 NASA       1976      12513
#> 3 DOE        1976      10882
#> 4 HHS        1976      9226
#> 5 NIH        1976      8025
#> 6 NSF        1976      2372
```

# Untidy ("wide")

```
head(fed_spend_wide)
```

```
#> # A tibble: 6 x 15
#>   year    DHS    DOC    DOD    DOE    DOT    EPA    HHS    Inte
#>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
#> 1 1976     0    819  35696  10882   1142    968   9226
#> 2 1977     0    837  37967  13741   1095    966   9507
#> 3 1978     0    871  37022  15663   1156   1175  10533
#> 4 1979     0    952  37174  15612   1004   1102  10127
#> 5 1980     0    945  37005  15226   1048    903  10045
#> 6 1981     0    829  41737  14798    978    901  9644
```

# Identifying tidy data

1. Pick a cell in a column
2. Ask "is **cell** a value of **column**?"
3. Repeat for each column

```
head(fed_spend_long)
```

```
#> # A tibble: 6 x 3
#>   department    year rd_budget_mil
#>   <chr>        <dbl>      <dbl>
#> 1 DOD          1976      35696
#> 2 NASA         1976      12513
#> 3 DOE          1976      10882
#> 4 HHS          1976      9226
#> 5 NIH          1976      8025
#> 6 NSF          1976      2372
```

```
head(fed_spend_wide)
```

```
#> # A tibble: 6 x 15
#>   year    DHS    DOC    DOD    DOE    DOT    EPA    HHS    Inte<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 1976     0    819  35696  10882   1142    968   9226
#> 2 1977     0    837  37967  13741   1095    966   9507
#> 3 1978     0    871  37022  15663   1156   1175  10533
#> 4 1979     0    952  37174  15612   1004   1102  10127
#> 5 1980     0    945  37005  15226   1048   903   10045
#> 6 1981     0    829  41737  14798    978   901   9644
```

# Identifying tidy data

Are the column names *values* of a variable?

```
head(fed_spend_long)
```

```
#> # A tibble: 6 x 3
#>   department  year rd_budget_mil
#>   <chr>      <dbl>        <dbl>
#> 1 DOD         1976       35696
#> 2 NASA        1976       12513
#> 3 DOE          1976       10882
#> 4 HHS          1976        9226
#> 5 NIH          1976       8025
#> 6 NSF          1976        2372
```

```
head(fed_spend_wide)
```

```
#> # A tibble: 6 x 15
#>   year    DHS    DOC    DOD    DOE    DOT    EPA    HHS    Inte
#>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
#> 1 1976     0    819  35696  10882   1142    968   9226
#> 2 1977     0    837  37967  13741   1095    966   9507
#> 3 1978     0    871  37022  15663   1156   1175  10533
#> 4 1979     0    952  37174  15612   1004   1102  10127
#> 5 1980     0    945  37005  15226   1048   903   10045
#> 6 1981     0    829  41737  14798    978   901   9644
```

# Quick practice 1: Is this data frame "tidy"?

Decide [here](#) (link also in #classroom)

**Description:** Tuberculosis cases in various countries

```
#> # A tibble: 6 x 4
#>   country     year   cases population
#>   <chr>       <dbl>   <dbl>        <dbl>
#> 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
```

# Quick practice 2: Is this data frame "tidy"?

Decide [here](#) (link also in #classroom)

**Description:** Word counts by character type in "Lord of the Rings" trilogy

```
#> # A tibble: 9 x 4
#>   Film          Race   Female   Male
#>   <chr>        <chr>    <dbl>    <dbl>
#> 1 The Fellowship Of The Ring Elf      1229     971
#> 2 The Fellowship Of The Ring Hobbit    14     3644
#> 3 The Fellowship Of The Ring Man       0     1995
#> 4 The Return Of The King Elf      183      510
#> 5 The Return Of The King Hobbit    2     2673
#> 6 The Return Of The King Man       268     2459
#> 7 The Two Towers   Elf      331      513
#> 8 The Two Towers   Hobbit    0     2463
#> 9 The Two Towers   Man       401     3589
```

# Quick practice 3: Is this data frame "tidy"?

Decide [here](#) (link also in #classroom)

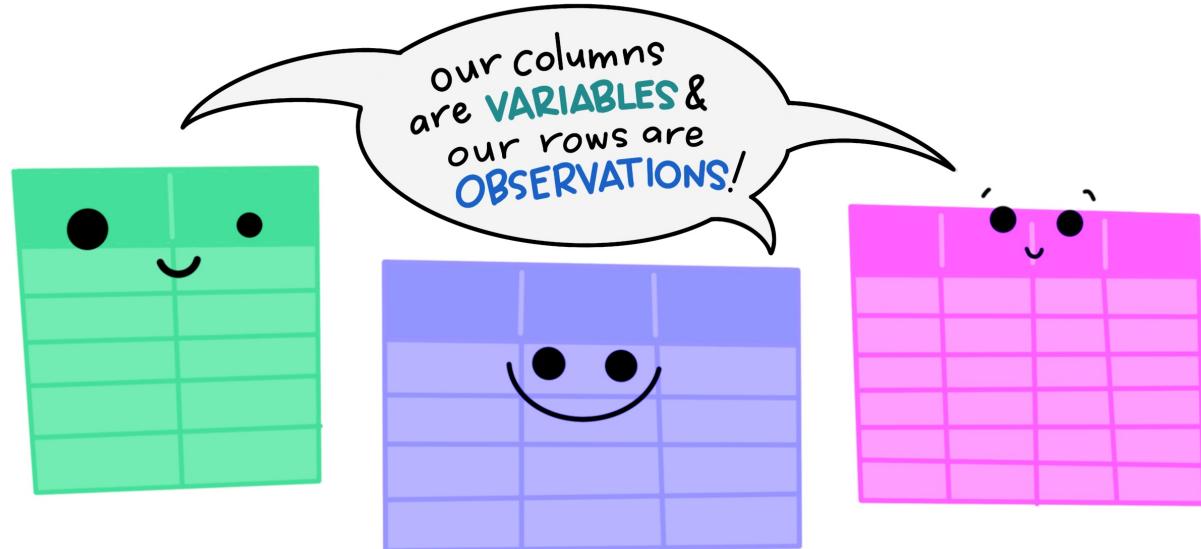
**Description:** Photovoltaic cell production by country

```
#> # A tibble: 6 x 10
#>   Year China Taiwan Japan Malaysia Germany `South Korea` `United States` 
#>   <dbl> <chr> <chr>  <dbl> <chr>    <chr>    <chr>    <dbl>
#> 1 1995 NA      NA        16.4 NA       NA       NA      34.
#> 2 1996 NA      NA        21.2 NA       NA       NA      38.
#> 3 1997 NA      NA        35    NA       NA       NA      51.
#> 4 1998 NA      NA        49    NA       NA       NA      53.
#> 5 1999 NA      NA        80    NA       NA       NA      60.
#> 6 2000 2.5     NA        129.  NA      22.5     NA      75.
```

# Why do we need tidy data?

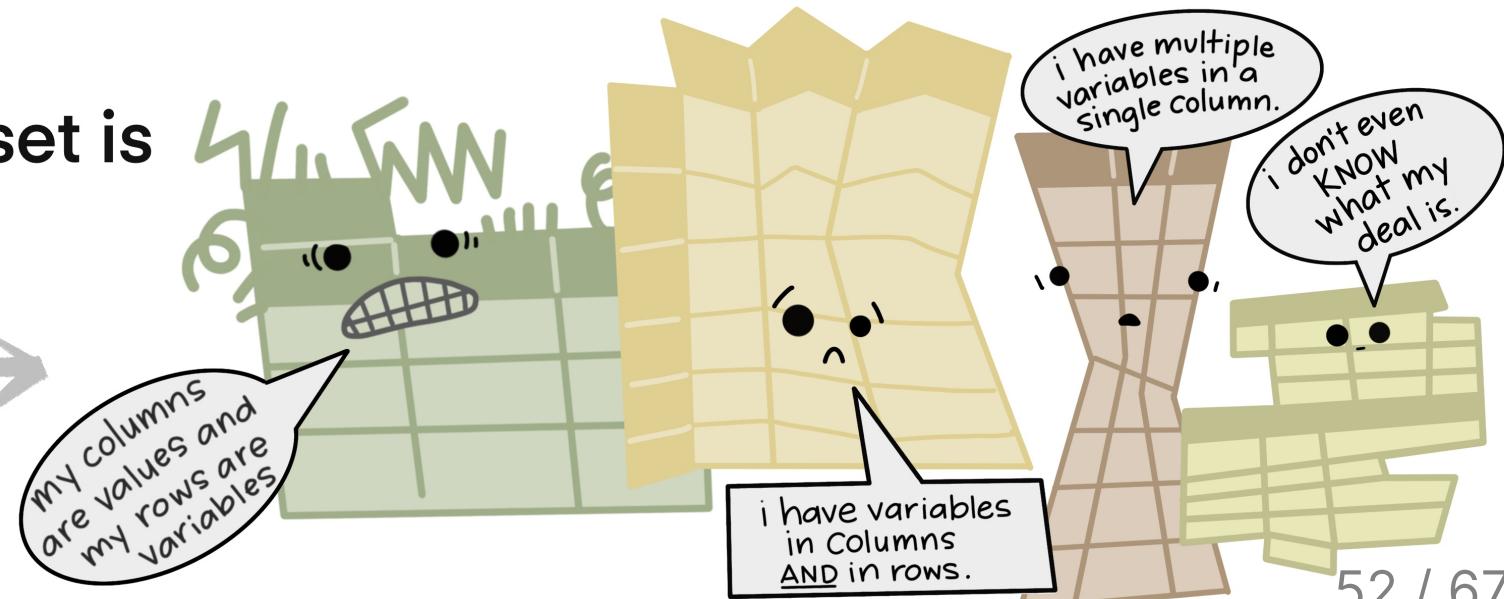
(a quick explanation with cute graphics, by [Allison Horst](#))

The standard structure of  
tidy data means that  
“tidy datasets are all alike...”

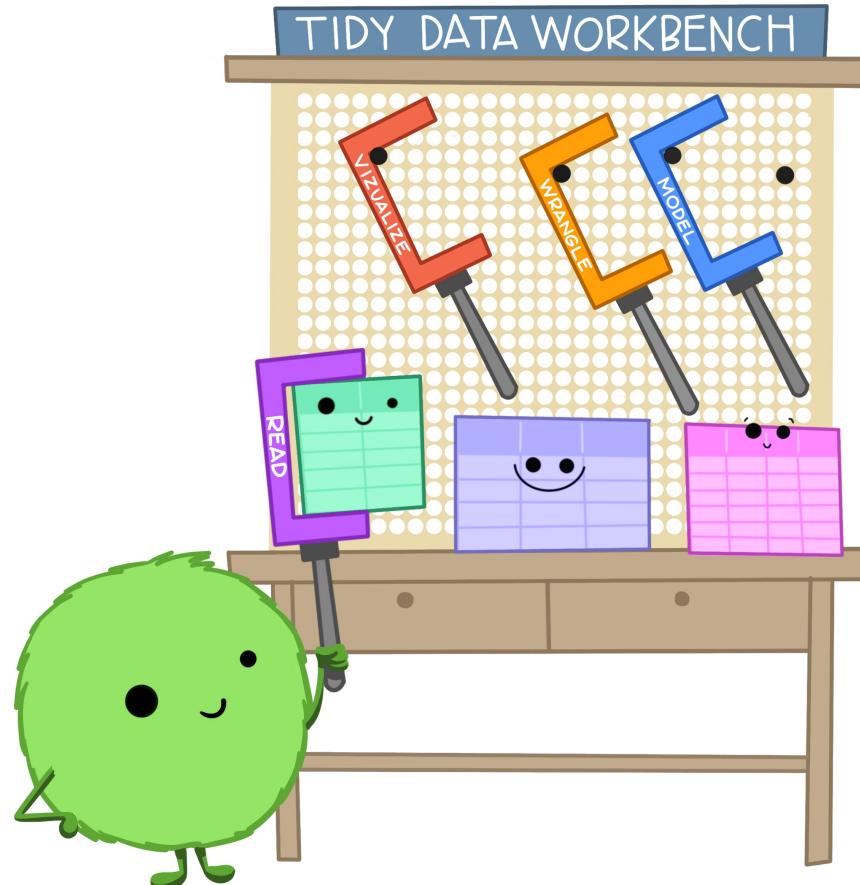


“...but every messy dataset is  
messy in its own way.”

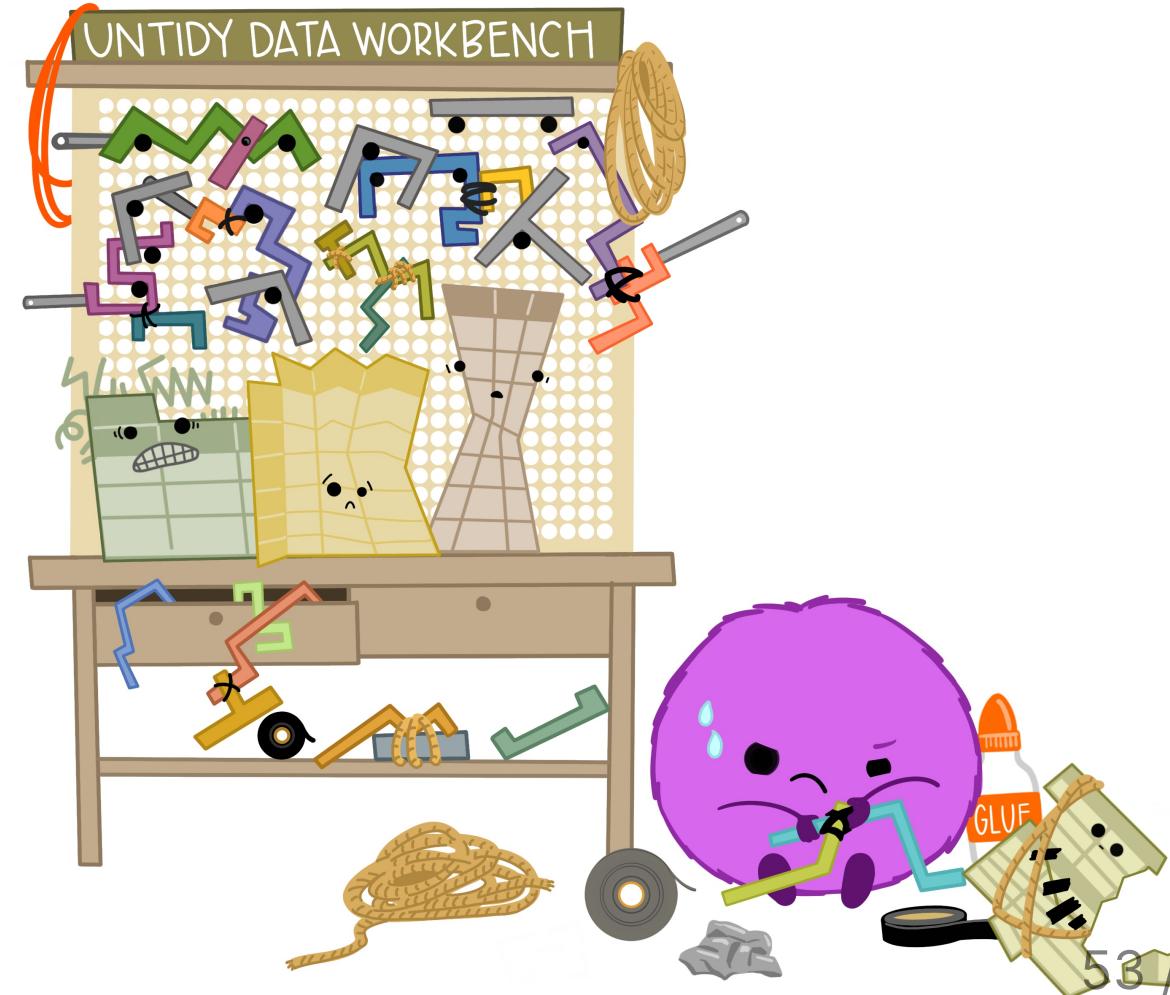
-HADLEY WICKHAM

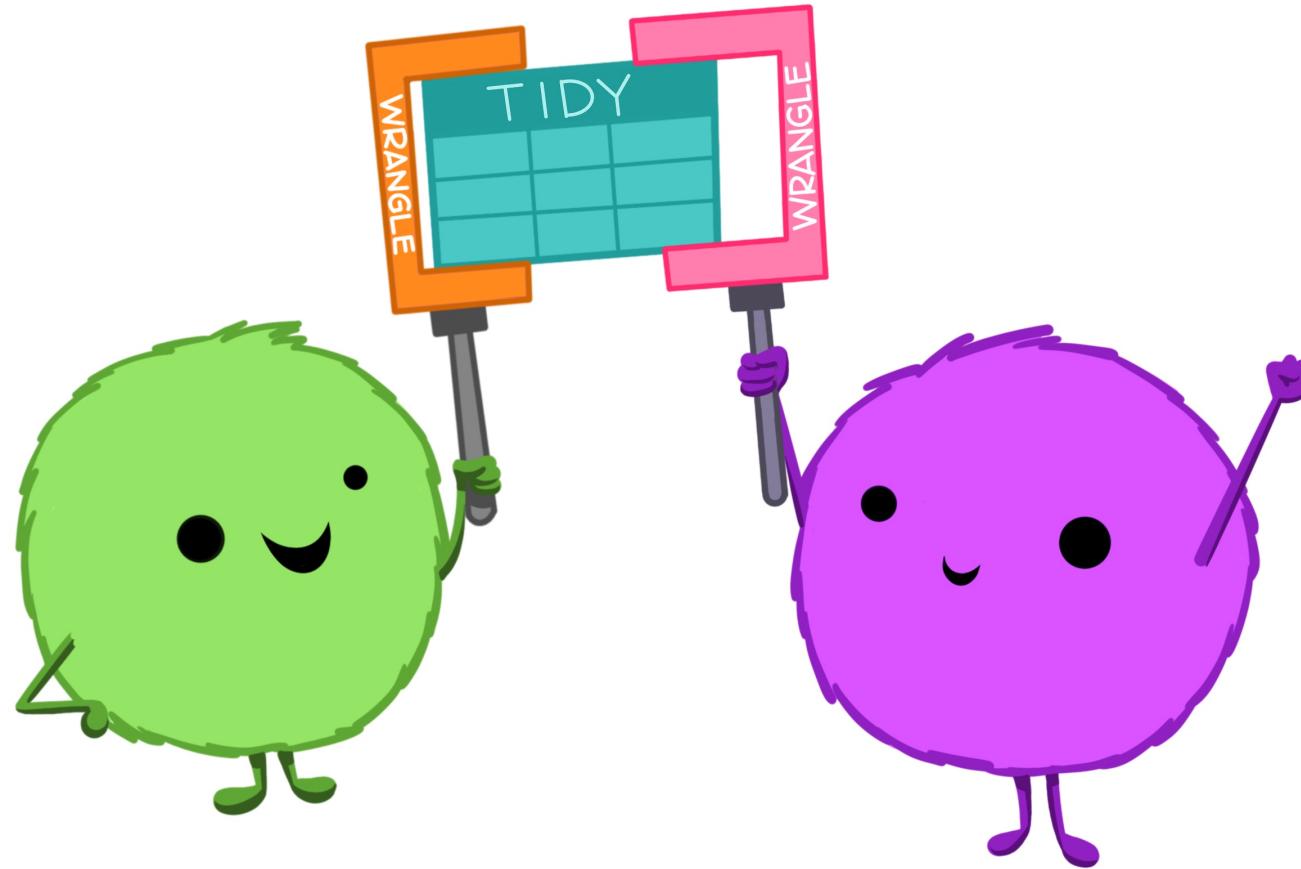


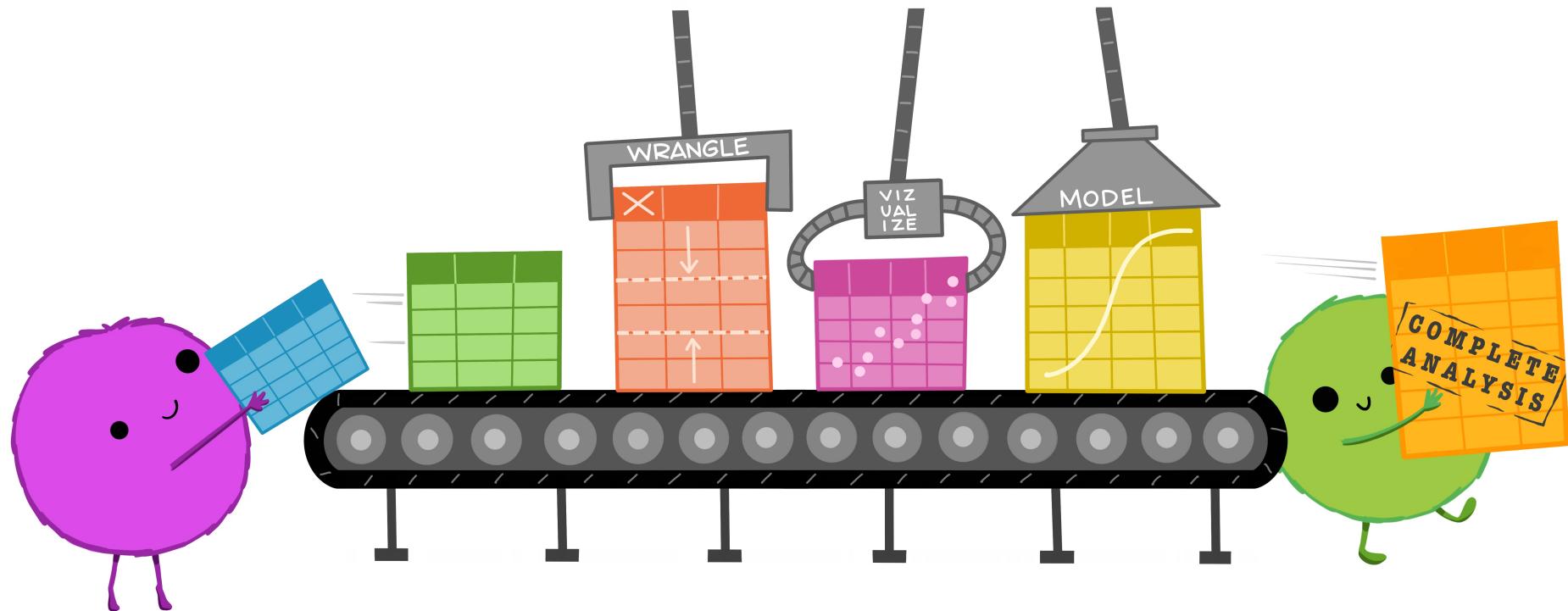
When working with tidy data,  
we can use the **same tools** in  
**similar ways** for different datasets...



...but working with untidy data often means  
reinventing the wheel with **one-time**  
**approaches** that are hard to iterate or reuse.







# Some tidy examples: data wrangling

Compute the total R&D spending in each year

```
head(fed_spend_long)
```

```
#> # A tibble: 6 x 3
#>   department    year rd_budget_mil
#>   <chr>        <dbl>      <dbl>
#> 1 DOD           1976     35696
#> 2 NASA          1976     12513
#> 3 DOE           1976     10882
#> 4 HHS           1976      9226
#> 5 NIH           1976      8025
#> 6 NSF           1976      2372
```

```
fed_spend_long %>%
  group_by(year) %>%
  summarise(total = sum(rd_budget_mil))
```

```
#> # A tibble: 42 x 2
#>   year   total
#>   <dbl>   <dbl>
#> 1 1976  86227
#> 2 1977  91807
#> 3 1978  94864
#> 4 1979  96601
#> 5 1980  96305
#> 6 1981  98304
#> 7 1982  95448
#> 8 1983  95010
#> 9 1984 105371
#> 10 1985 114818
```

# Some tidy examples: data wrangling

Compute the total R&D spending in each year

```
head(fed_spend_wide)
```

```
#> # A tibble: 6 x 15
#>   year    DHS    DOC    DOD    DOE    DOT    E
#>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 1976     0    819  35696  10882   1142    9
#> 2 1977     0    837  37967  13741   1095    9
#> 3 1978     0    871  37022  15663   1156   11
#> 4 1979     0    952  37174  15612   1004   11
#> 5 1980     0    945  37005  15226   1048    9
#> 6 1981     0    829  41737  14798    978    9
```

```
fed_spend_wide %>%
  mutate(total = DHS + DOC + DOD + DOE + DOT) %>%
  select(year, total)
```

```
#> # A tibble: 42 x 2
#>   year  total
#>   <dbl> <dbl>
#> 1 1976  86227
#> 2 1977  91807
#> 3 1978  94864
#> 4 1979  96601
#> 5 1980  96305
#> 6 1981  98304
#> 7 1982  95448
#> 8 1983  95010
#> 9 1984 105371
#> 10 1985 114818
```

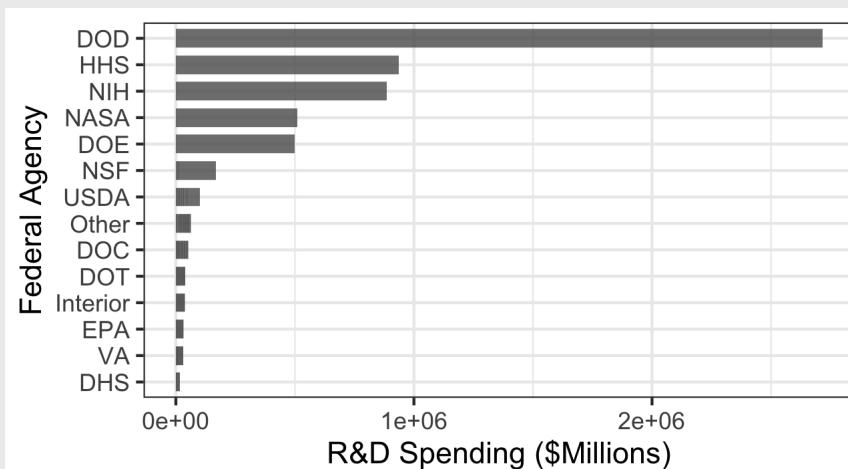
# Some tidy examples: plotting

Make a bar chart of total R&D spending by agency

```
head(fed_spend_long)
```

```
#> # A tibble: 6 x 3
#>   department    year rd_budget_mil
#>   <chr>        <dbl>          <dbl>
#> 1 DOD           1976          35696
#> 2 NASA          1976          12513
#> 3 DOE           1976          10882
#> 4 HHS           1976          9226
#> 5 NIH           1976          8025
#> 6 NSF           1976          2372
```

```
ggplot(fed_spend_long) +
  geom_col(aes(x = rd_budget_mil, y = reorder(department, rd_budget_mil)))
  width = 0.7, alpha = 0.8) +
  theme_bw(base_size = 15) +
  labs(x = "R&D Spending ($Millions)",
       y = "Federal Agency")
```



# Tidying and Untidying your data with `spread()` and `gather()`

# spread( ): from tidy ("long") to untidy ("wide")

**key** = column names, **value** = cells

long			wide		
id	key	val	id	x	y
1	x	a	1	a	e
2	x	b			
1	y	c	2	b	f
2	y	d			
1	z	e			
2	z	f			

# spread( ): from tidy ("long") to untidy ("wide")

key = column names, value = cells

```
head(fed_spend_long)
```

```
#> # A tibble: 6 x 3
#>   department    year rd_budget_mil
#>   <chr>        <dbl>      <dbl>
#> 1 DOD          1976     35696
#> 2 NASA         1976     12513
#> 3 DOE          1976     10882
#> 4 HHS          1976      9226
#> 5 NIH          1976      8025
#> 6 NSF          1976      2372
```

```
fed_spend_wide <- fed_spend_long %>%
  spread(key = department,
         value = rd_budget_mil)
```

```
head(fed_spend_wide)
```

```
#> # A tibble: 6 x 15
#>   year   DHS   DOC   DOD   DOE   DOT   EPA
#>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <
#> 1 1976     0    819  35696  10882   1142   968
#> 2 1977     0    837  37967  13741   1095   966
#> 3 1978     0    871  37022  15663   1156   1175  1
#> 4 1979     0    952  37174  15612   1004   1102  1
#> 5 1980     0    945  37005  15226   1048   903   1
#> 6 1981     0    829  41737  14798   978   901
```

# gather( ): from untidy ("wide") to tidy ("long")

**key** = column names, **value** = cells

wide				long			
id	x	y	z	key	id	key	val
1	a	c	e	val	1	x	a
2	b	d	f		2	x	b
1				val	1	y	c
2					2	y	d
1				val	1	z	e
2					2	z	f

# gather( ): from untidy ("wide") to tidy ("long")

key = column names, value = cells

```
#> # A tibble: 6 x 15
#>   year    DHS    DOC    DOD    DOE    DOT    EPA
#>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
#> 1 1976     0    819  35696  10882   1142    968
#> 2 1977     0    837  37967  13741   1095    966
#> 3 1978     0    871  37022  15663   1156   1175  1
#> 4 1979     0    952  37174  15612   1004   1102  1
#> 5 1980     0    945  37005  15226   1048    903  1
#> 6 1981     0    829  41737  14798    978    901
```

```
fed_spend_long <- fed_spend_wide %>%
  gather(key = "department",
         value = "rd_budget_mil",
         DHS:VA)

head(fed_spend_long)
```

```
#> # A tibble: 6 x 3
#>   year department rd_budget_mil
#>   <dbl> <chr>          <dbl>
#> 1 1976 DHS              0
#> 2 1977 DHS              0
#> 3 1978 DHS              0
#> 4 1979 DHS              0
#> 5 1980 DHS              0
#> 6 1981 DHS              0
```

# Your turn: Tidy <--> Untidy

10:00

We already read in the following two data frames:

- `pv_cells`
- `milk_production`

Now we'll modify the format of each:

1. Use `spread()` to "untidy" the `milk_production` data into a format where the columns are state names and the values are the milk produced in each state.
2. Use `gather()` to "tidy" the `pv_cells` data into a data frame with three names: `year`, `country`, `numCells`

Start thinking about research questions

# 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 [Example Projects Page](#) page