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*

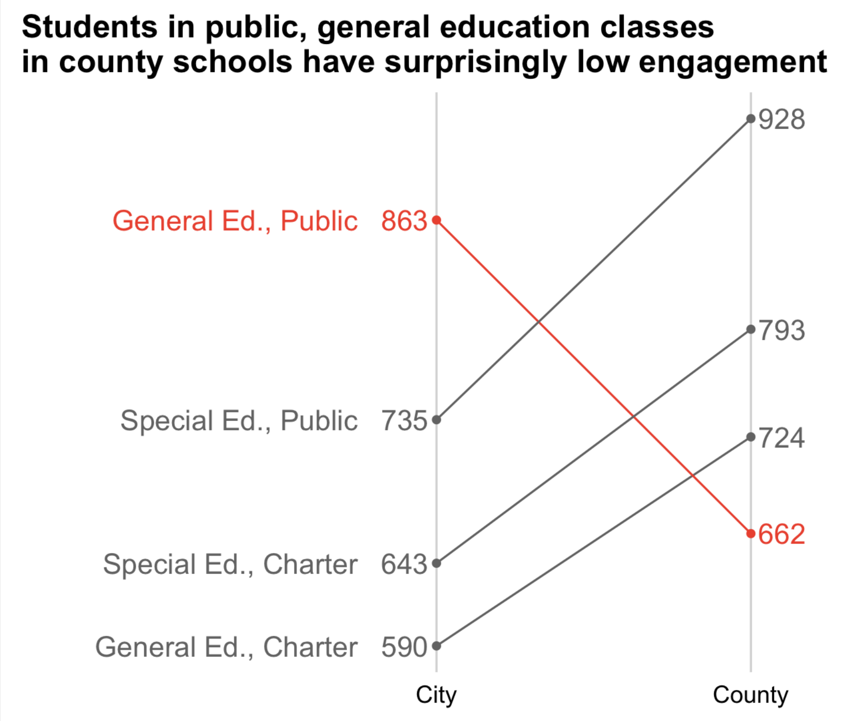
Class goal: translate *data* into *information*

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



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:

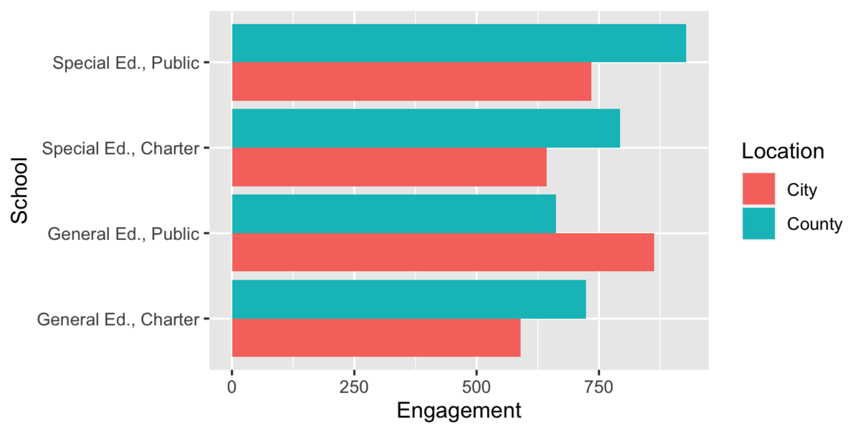
```
engagement_data <- engagement_data %>%
    gather(Location, Engagement, City:County) %>%
    mutate(Location = fct_relevel(
      Location, c('City', 'County')))
engagement_data
```

```
#>                 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
```

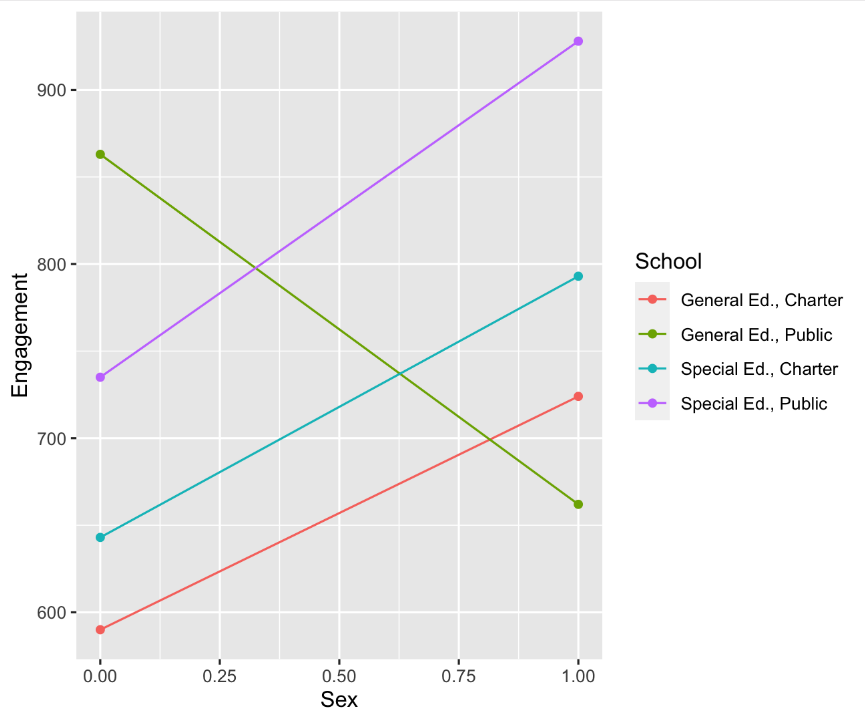
Data exploration: an iterative process

Initial exploratory plotting:

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

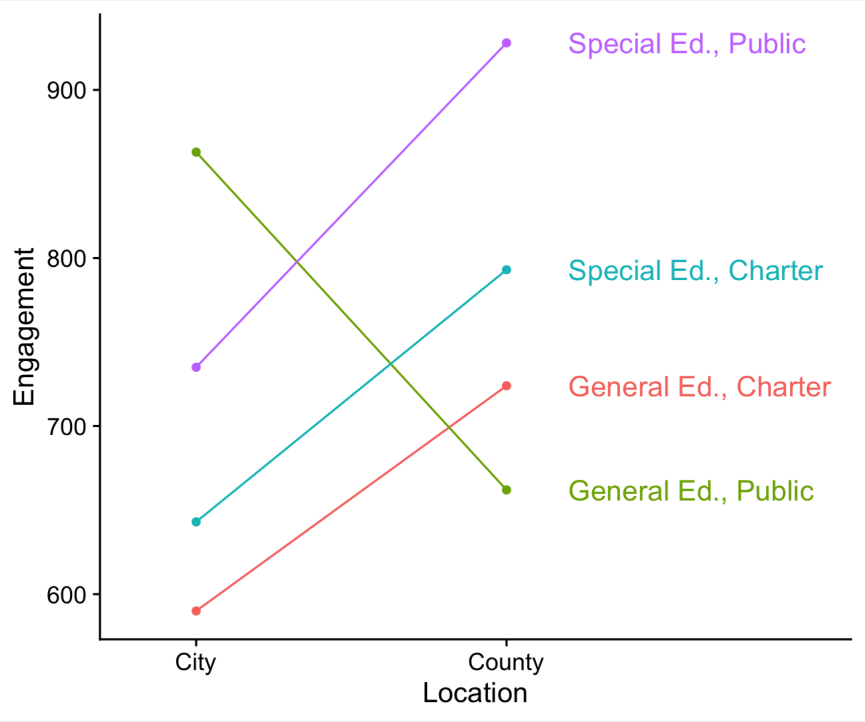


More exploratory plotting: highlight difference

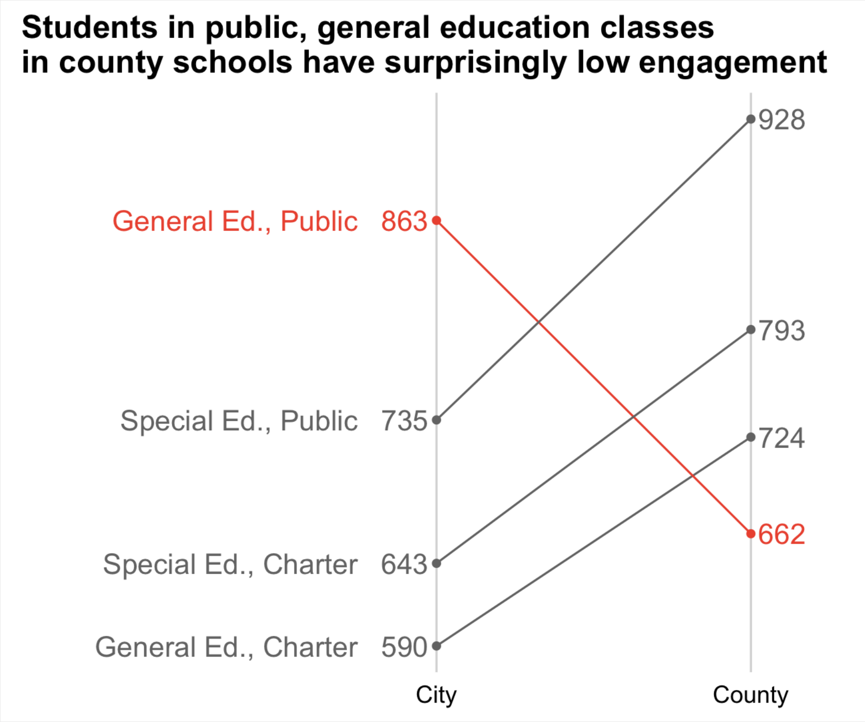


Data exploration: an iterative process

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

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

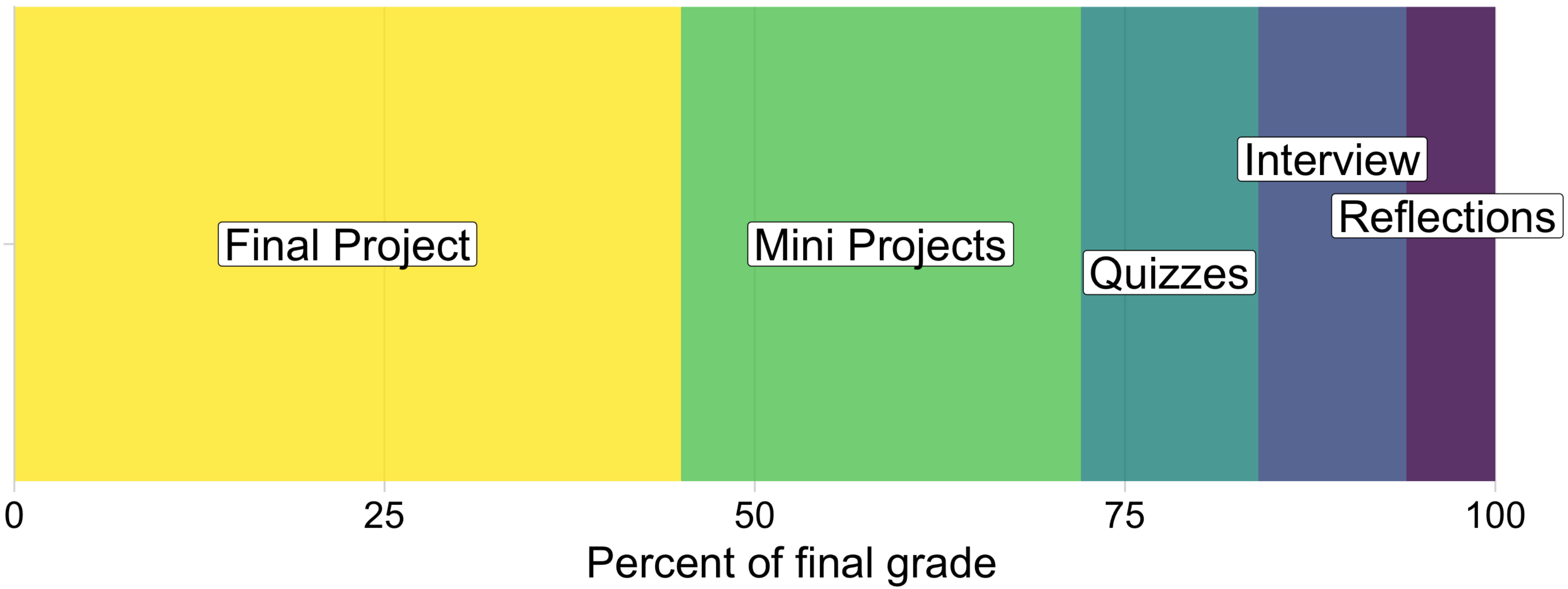
1) Weekly "reflections" on readings

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



Final Project
Mini Projects
Quizzes
Interview
Reflections
0
25
50
75
100
Percent of final grade

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`	`read_csv()`	**readr**
`.txt`	`read.table()`	**utils**
`.xlsx`	`read_excel()`	**readxl**

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", "22.5", "23.5", "55", "121.5",
#> $ `South Korea`   <chr> NA, NA, "NA", "NA", "NA", "NA", "NA", "NA", "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", "48.200000000000017", "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, 20
#> $ China           <chr> "NA", "NA", "NA", "NA", "NA", "2.5", "3", "10", "13", "40", "128.30000000000001", "34
#> $ Taiwan          <chr> "NA", "NA", "NA", "NA", "NA", "NA", "3.5", "8", "17", "39.299999999999997", "88", "16
#> $ 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,
#> $ Malaysia        <chr> "NA", "NA", "NA", "NA", "NA", "NA", "0", "0", "0", "0", "0", "0", "100.1", "397.9", "
#> $ Germany         <chr> "NA", "NA", "NA", "NA", "NA", "22.5", "23.5", "55", "121.5", "193", "339", "469.1", "
#> $ `South Korea`   <chr> "NA", "NA", "NA", "NA", "NA", "NA", "0", "0", "0", "0", "5.3", "13", "31.88393590567
#> $ `United States` <dbl> 34.7500, 38.8500, 51.0000, 53.7000, 60.8000, 75.0000, 100.3000, 120.6000, 103.0000, 1
#> $ Others          <chr> "NA", "NA", "NA", "NA", "NA", "48.200000000000017", "69.800000000000011", "97.299999
#> $ World           <dbl> 77.600, 88.600, 125.800, 154.900, 201.300, 276.800, 371.300, 542.000, 749.400, 1198.8
```

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.

🔍 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 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
```

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
```

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>    <
#> 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>    <
#> 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>    <
#> 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)

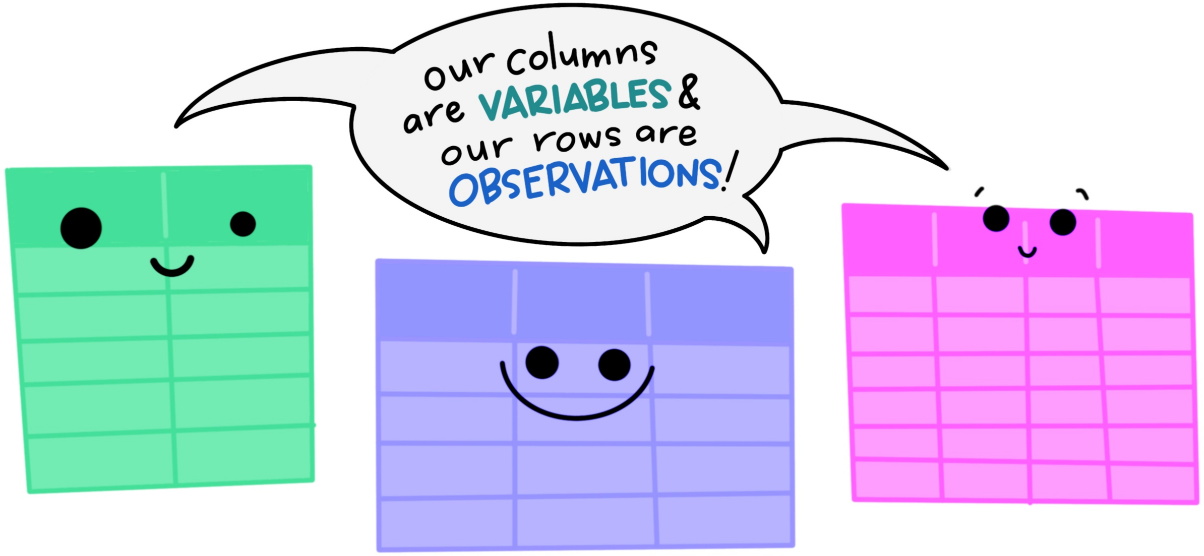
Description: Photovoltaic cell production by country

```
#> # 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?

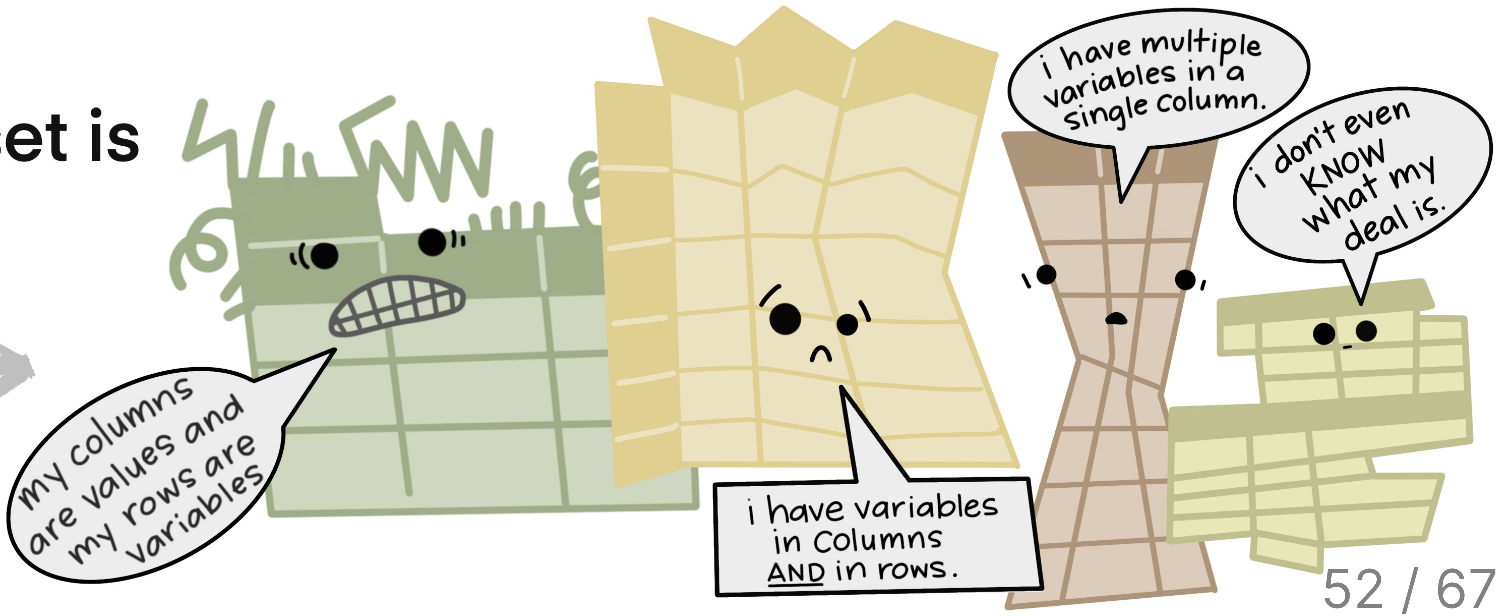
(a quick explanation with cute graphics, by [Allison Horst])

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



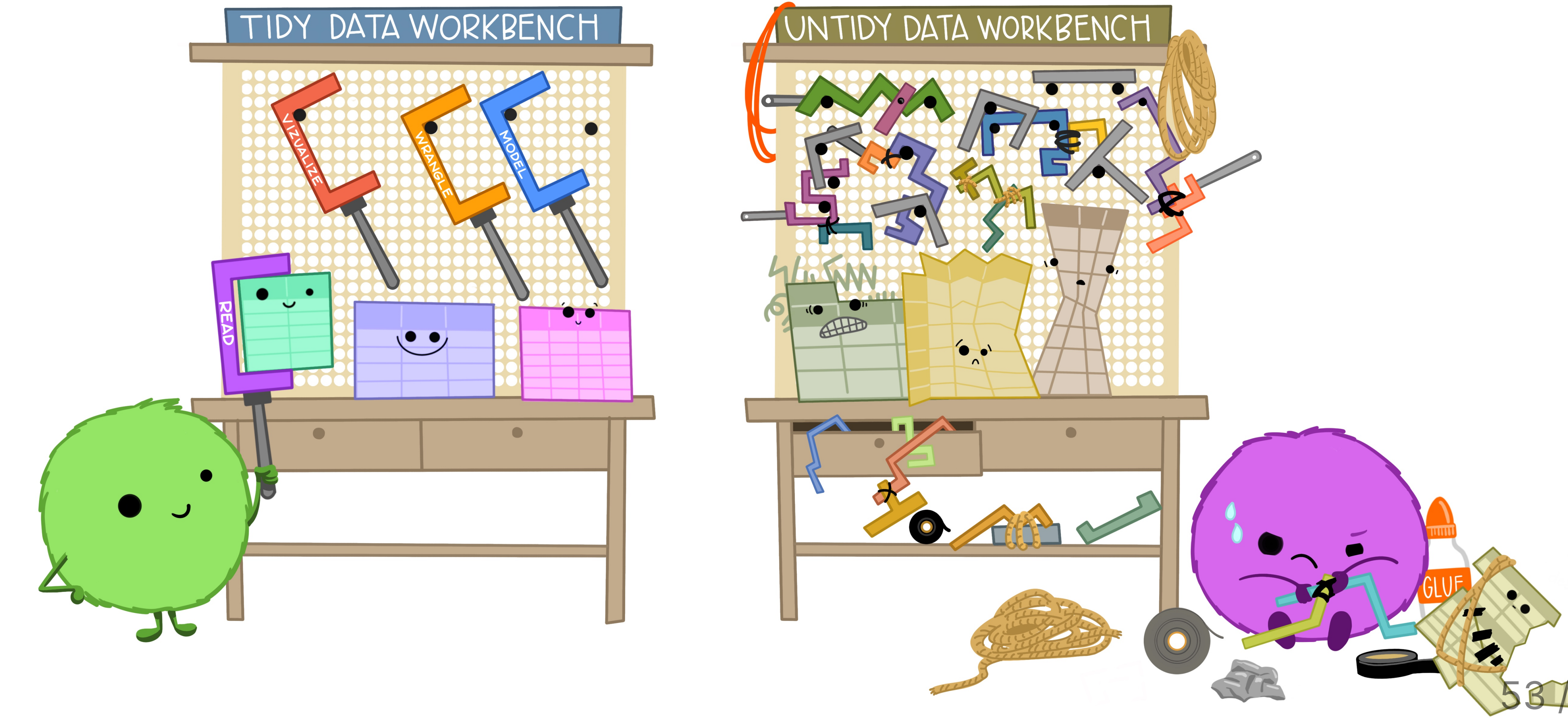
"...but every messy dataset is messy in its own way."

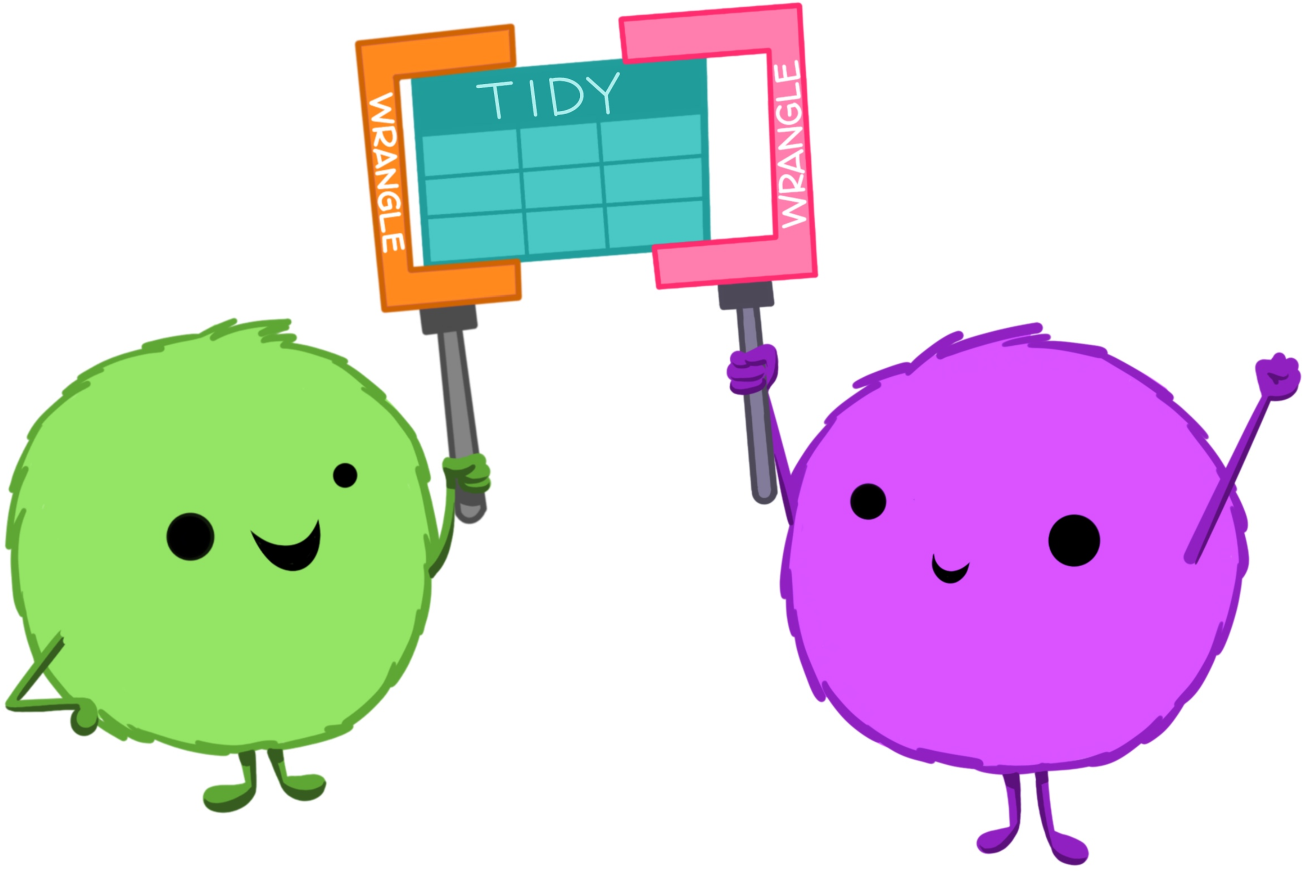
—HADLEY WICKHAM

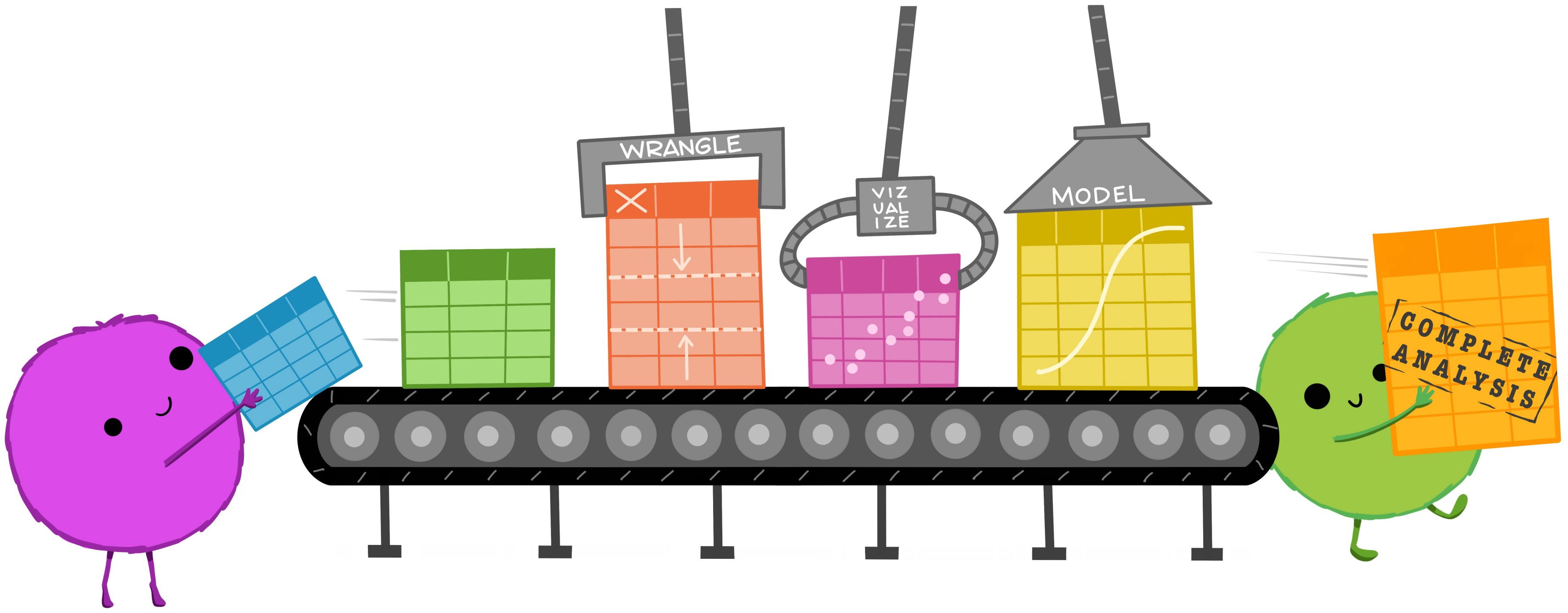


When working with tidy data, we can use the same tools in similar ways for different datasets...

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




WRANGLE
TIDY
WRANGLE


WRANGLE
VIZ
UAL
IZE
MODEL
COMPLETE
ANALYSIS

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> <db
#> 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
```

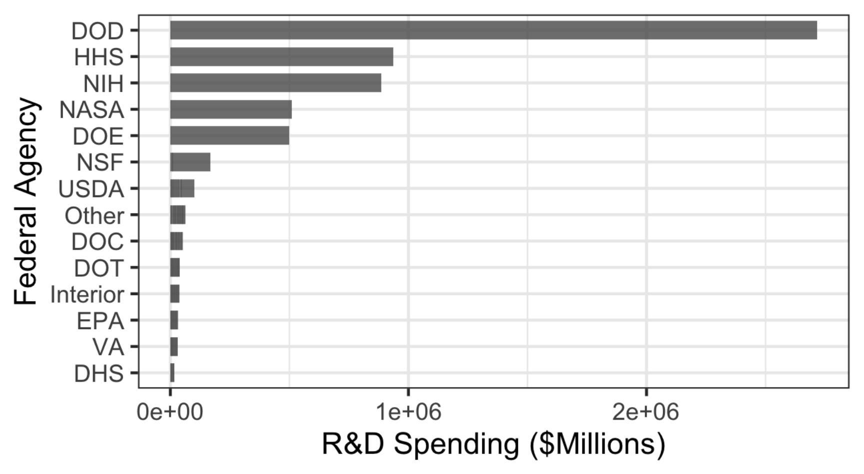
Some tidy examples: plotting

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
```

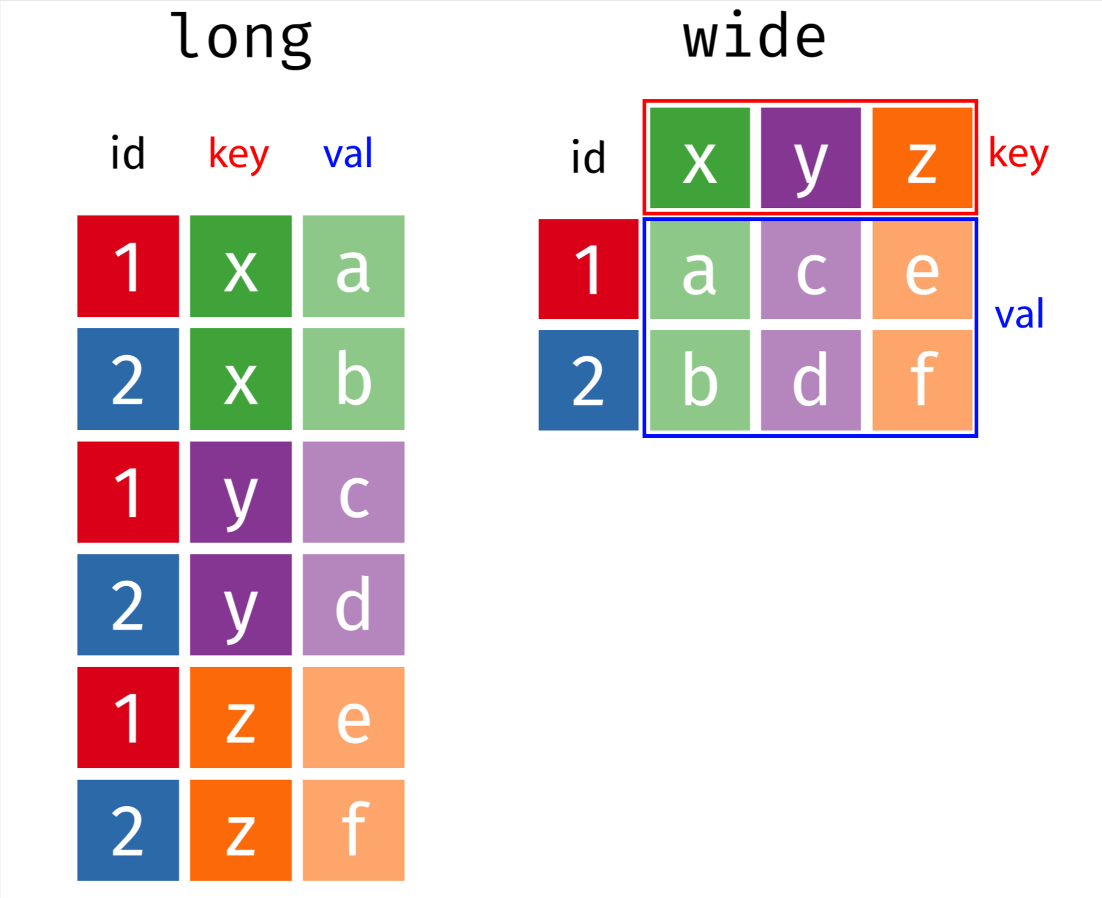
```
ggplot(fed_spend_long) +
  geom_col(aes(x = rd_budget_mil, y = reorder(de
           width = 0.7, alpha = 0.8) +
  theme_bw(base_size = 15) +
  labs(x = "R&D Spending ($Millions)",
       y = "Federal Agency")
```



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

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

`key` = column names, `value` = cells



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

key = column names, value = cells

```r
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
```

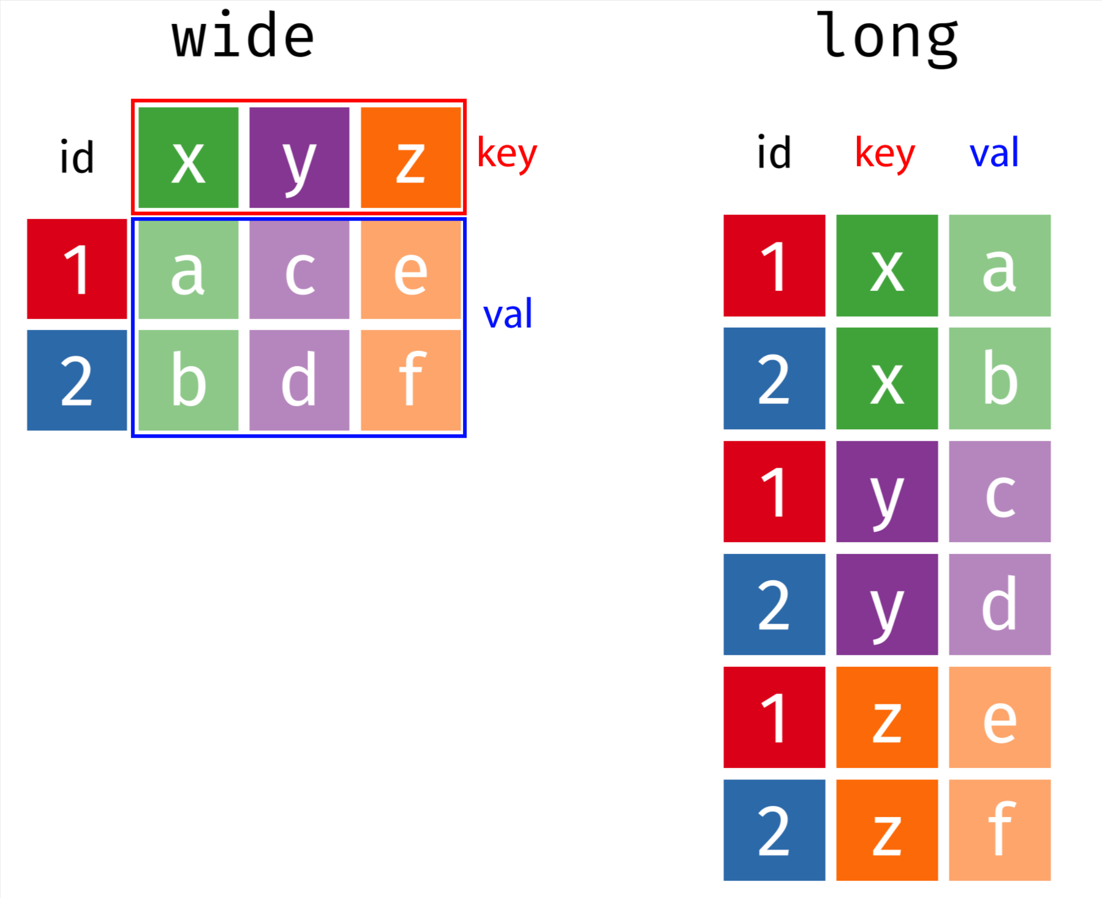
```r
fed_spend_wide <- fed_spend_long %>%
    spread(key = department,
           value = rd_budget_mil)

head(fed_spend_wide)
```

```
#> # A tibble: 6 x 15
#>     year   DHS   DOC   DOD   DOE   DOT   EPA <
#>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <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



`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
```

```r
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