Data Wrangling in R

Look at the Data!

Prof. Bisbee

Vanderbilt University

Lecture Date: 2023/09/18

Slides Updated: 2023-09-18

Agenda

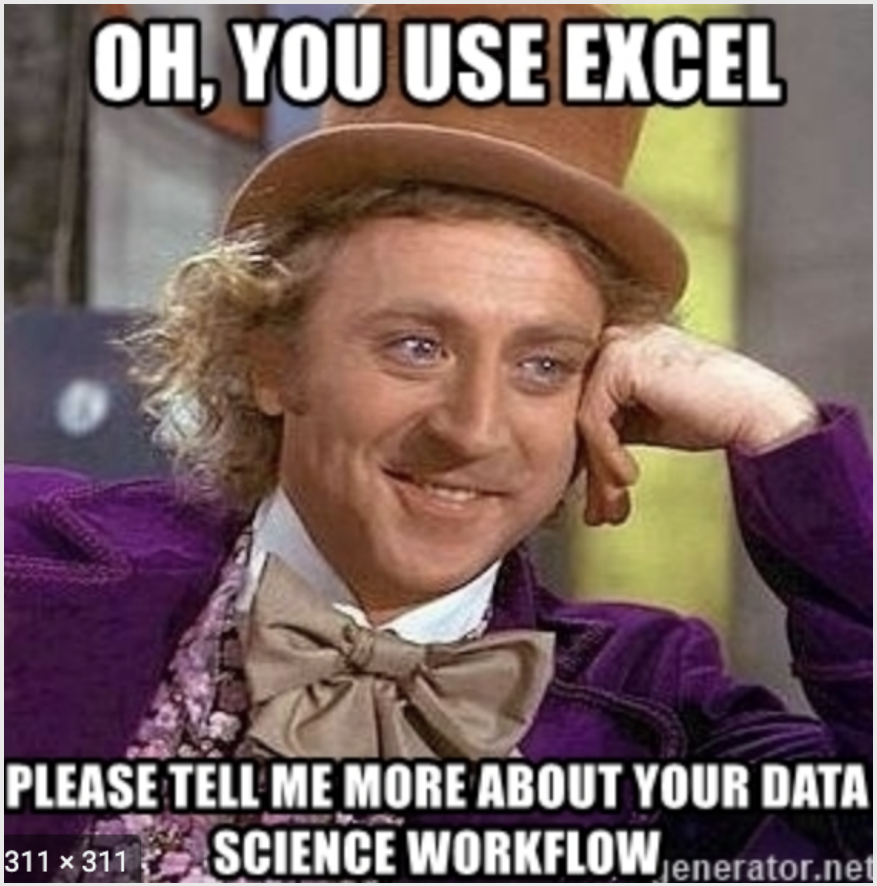
1. What is "data wrangling"?
2. Why `R`?
3. `tibbles` (table dataframes)
4. Political + data **science**
    - Michigan exit polls

What is "data wrangling"?

- Preparing data for analysis
    - DANGER: Most important but least appreciated step!

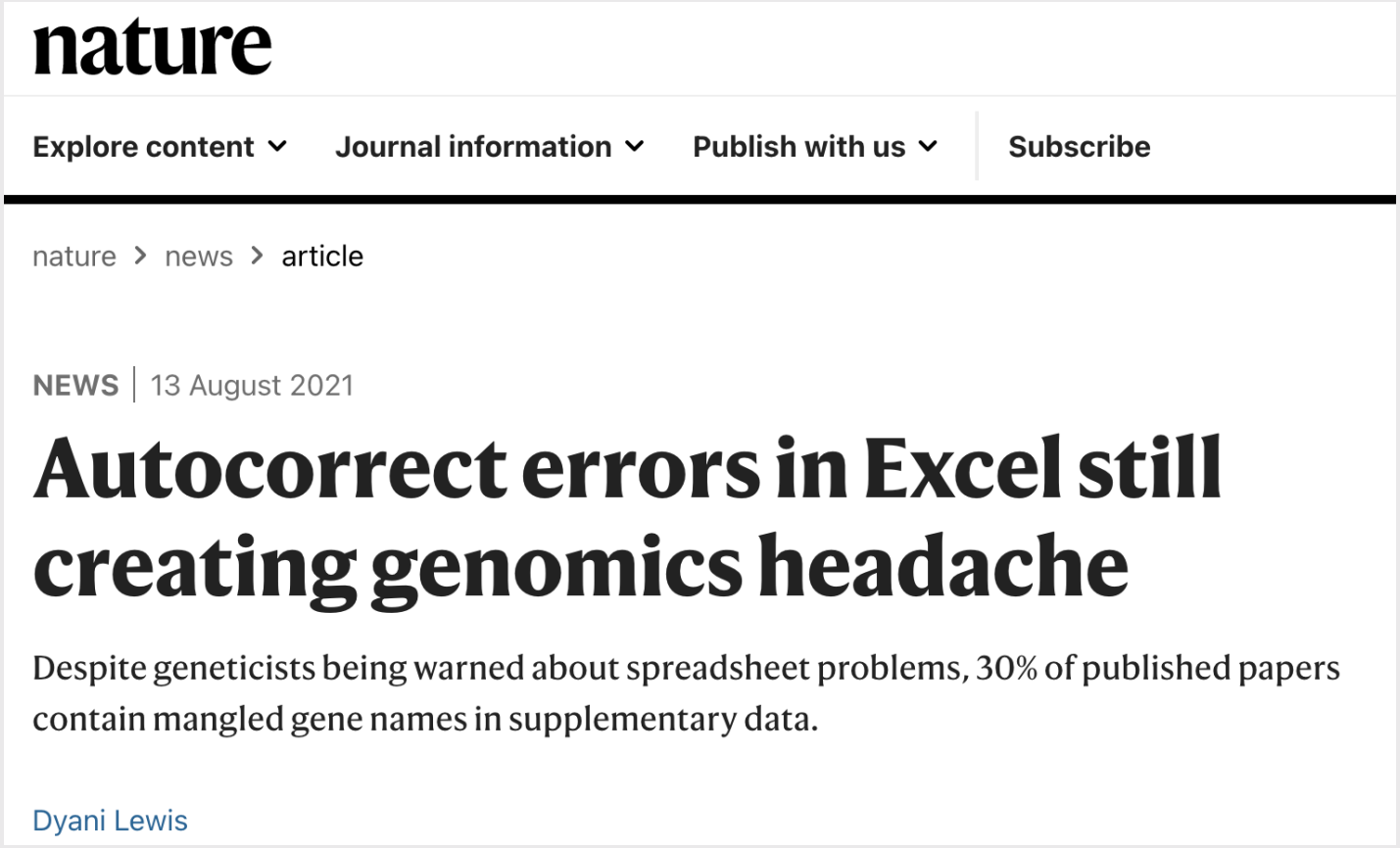
Danger!

- Data wrangling gone bad



Danger!

- Data wrangling gone bad



nature

Explore content | Journal information | Publish with us | Subscribe

nature > news > article

NEWS | 13 August 2021

Autocorrect errors in Excel still creating genomics headache

Despite geneticists being warned about spreadsheet problems, 30% of published papers contain mangled gene names in supplementary data.

Dyani Lewis

Danger!

- Data wrangling gone bad

Economic Policy

Is the evidence for austerity based on an Excel spreadsheet error?

By Brad Plumer

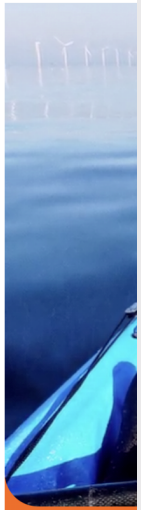
April 16, 2013

One of the more influential studies that's often used to argue for austerity has come in for an extensive new critique. (***Update:*** The authors respond down below.)

The paper in question is Carmen Reinhart and Kenneth Rogoff's famous 2010 study "Growth in a Time of Debt," which found that economic growth severely suffers when a country's public debt level reaches 90 percent of GDP. That 90 percent figure has often been cited in the past few years as one big reason why countries must trim their deficits — even if their economies are still weak.

But a new critique (pdf) by Thomas Herndon, Michael Ash and Robert Pollin claims that this result may need revision. For one, the economists argue that Reinhart and Rogoff excluded three episodes of high-debt, high-growth nations — Canada, New Zealand, and Australia in the late 1940s. Second, they argue, Reinhart and Rogoff made some contestable assumptions about weighting different historical episodes.

Now, those are two methodological objections. But there's also a third problem, as Mike Konczal details here. Reinhart and Rogoff appear to have made an error with one of their Excel spreadsheet formulas. By typing AVERAGE(L30:L44) at one point instead of AVERAGE(L30:L49), they left out Belgium, a key counterexample:



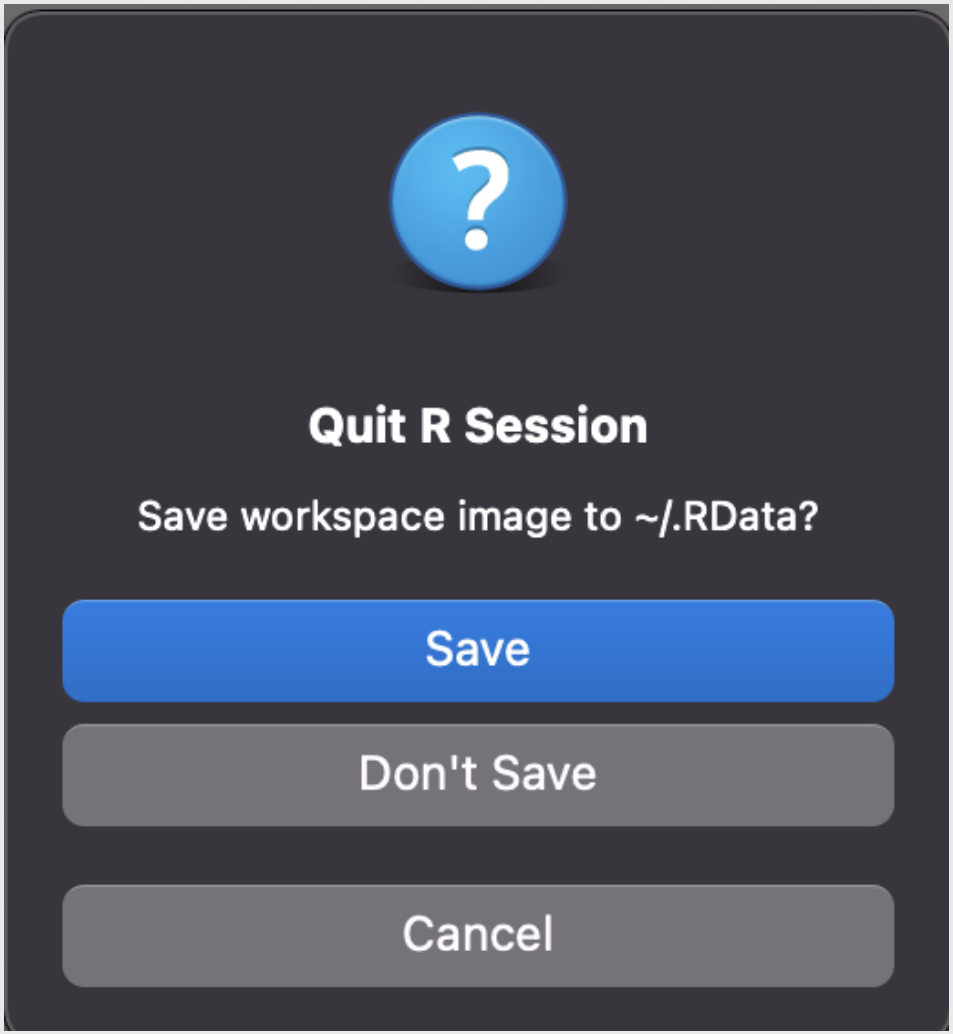
Danger!

- How do we avoid these mistakes?
- Three principles
    1. Replicability
    2. Understandability
    3. Robustness
- The big picture
    - Load **raw data** into an `RStudio` script
    - Wrangle the data within this script
    - Save a **new version** of the wrangled data with a **different file name**
- The point: Someone else should be able to recreate your work from scratch!

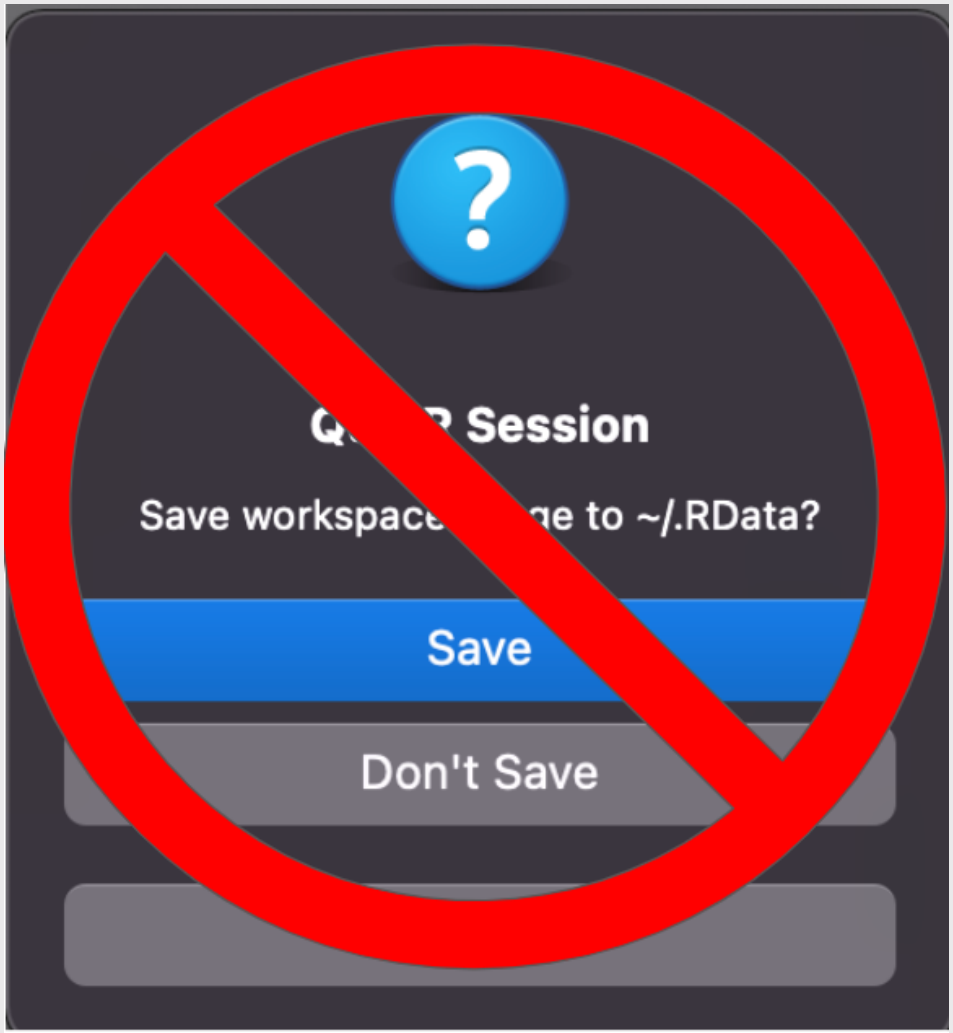
Why R?

- Checks all three boxes
- But even `R` can be corrupted...
- ...with `Save workspace image`
- **NEVER** **`Save workspace image`**
    - If you do, the next time you open `R`, it will load the modified data

NEVER SAVE WORKSPACE IMAGE

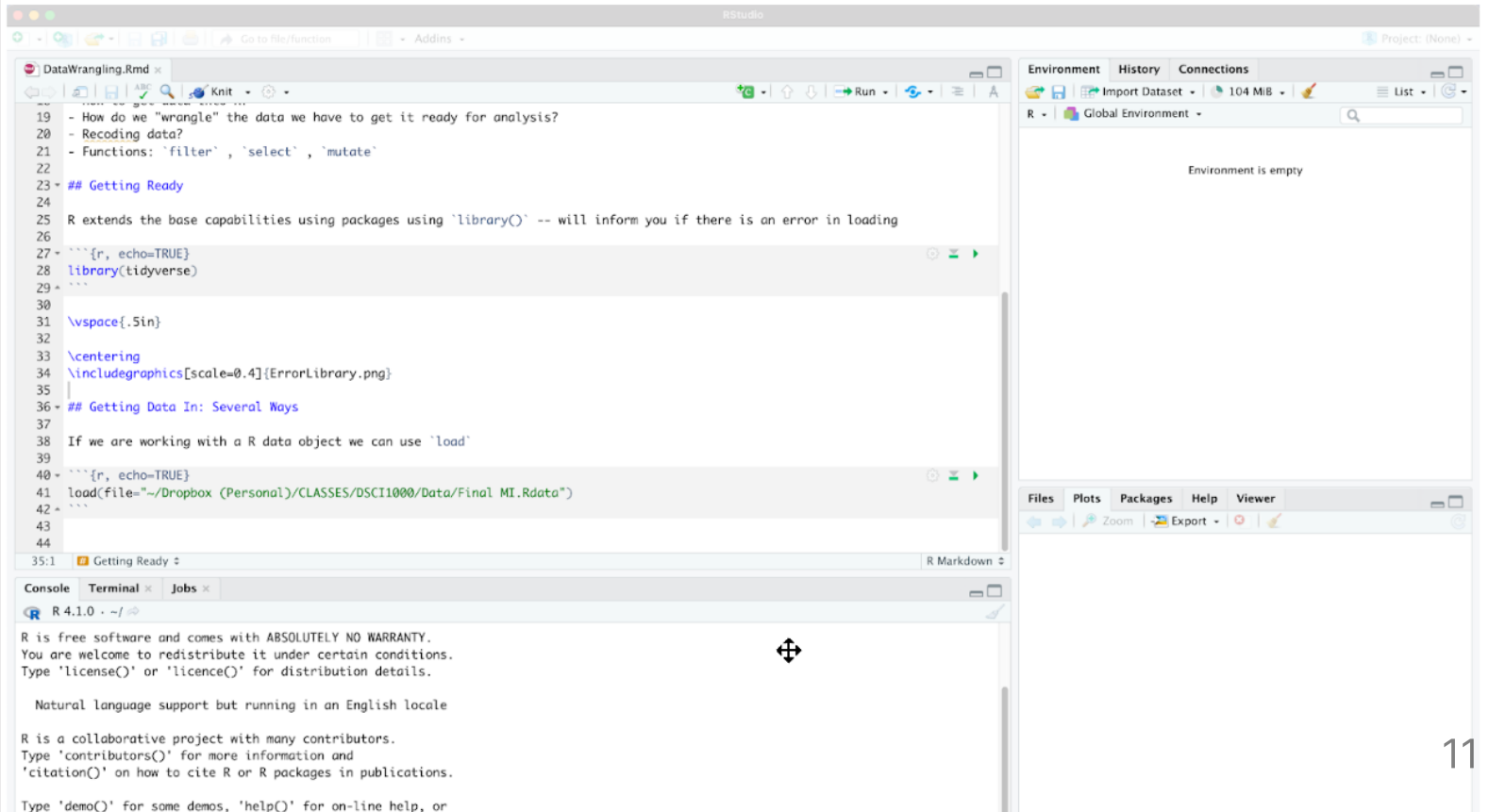


NEVER SAVE WORKSPACE IMAGE



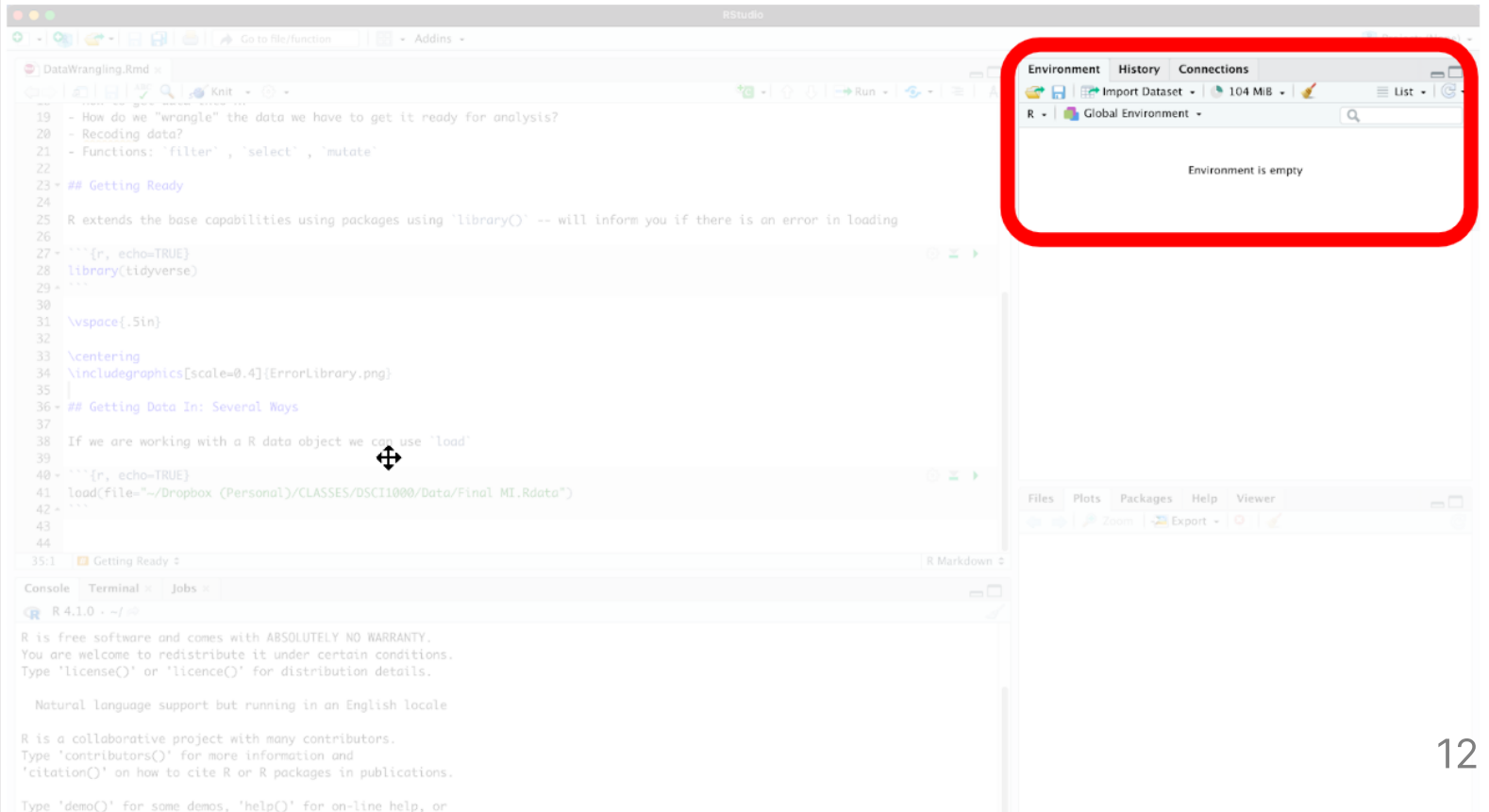
Why R?

- **ALWAYS** start with an `empty environment`
  - How do you know if the environment is empty?



Why R?

- **ALWAYS** start with an `empty environment`
    - How do you know if the environment is empty?

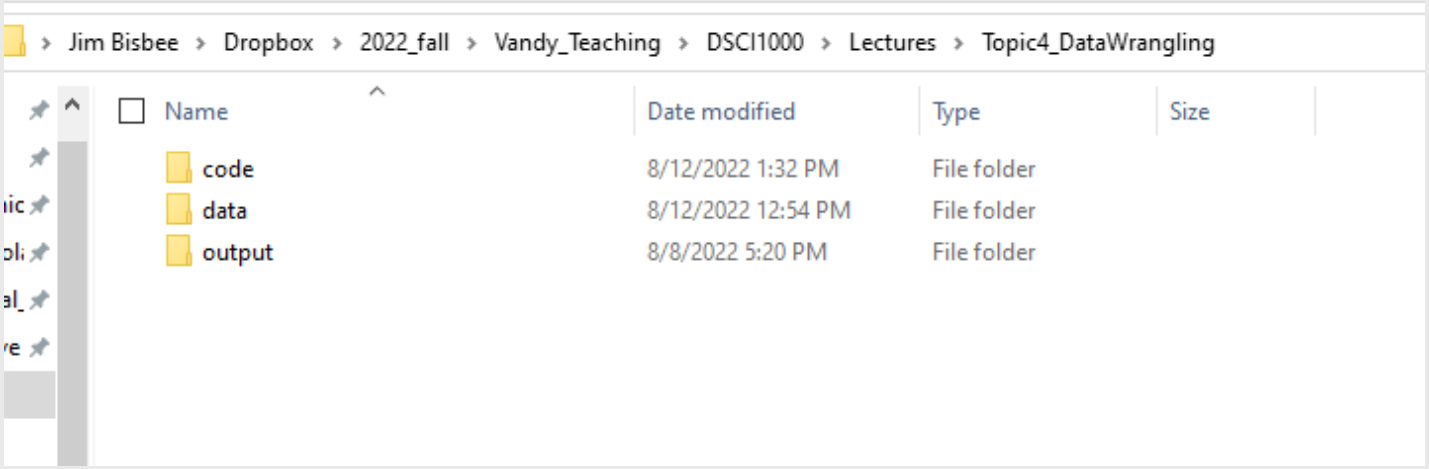


Why R?

- R scripts or RMarkDown scripts should be:
    1. Single purpose (but definitions can vary)
    2. Well-commented (make it easy for anyone to understand)
    3. Iteratively built (constant bug checking)

Let's get started

- Set up three subfolders



- Download `MI2020_ExitPoll.rds` to your `data` folder

Pausing for science

- What are these data?
- "Exit poll" data from Michigan in the fall of 2020
  - Polls fielded either on or just before an election
  - Used to understand *why* voters chose a candidate...
  - ...not to *predict* who will win (voting data is used for that)

Michigan

- A "swing state" in 2020, following unexpected support for Trump in 2016

Politics • Analysis

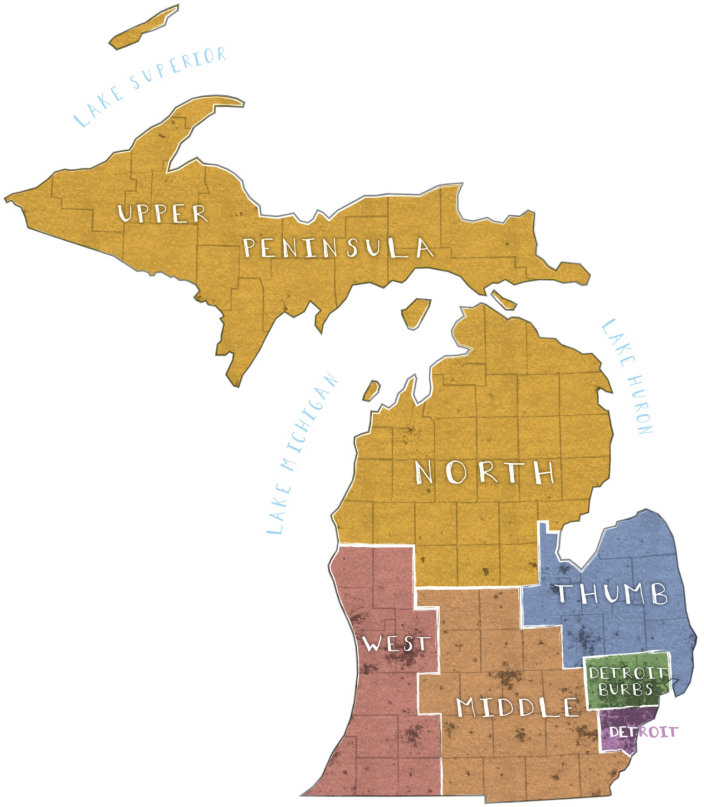
The six political states of Michigan

Story by **David Weigel**
Map by **Lauren Tierney**
Aug. 9, 2020

First in a series on swing states

How did Michigan become a defining swing state of the Trump era? Some of it had to do with history, both the state's 20-year stretch of Democratic wins and the revived conservatism of Detroit's suburbs. Plenty had to do with the margin — fewer than 11,000 votes out of nearly 4.8 million cast and counted. Donald Trump filled in the gaps, arguing that his opposition to the NAFTA free-trade deal had broken the old Democratic coalition, as the state's Democrats seemed to warn it could when they gave Bernie Sanders a narrow primary win over Hillary Clinton.

But Trump's appeal in the state may have been



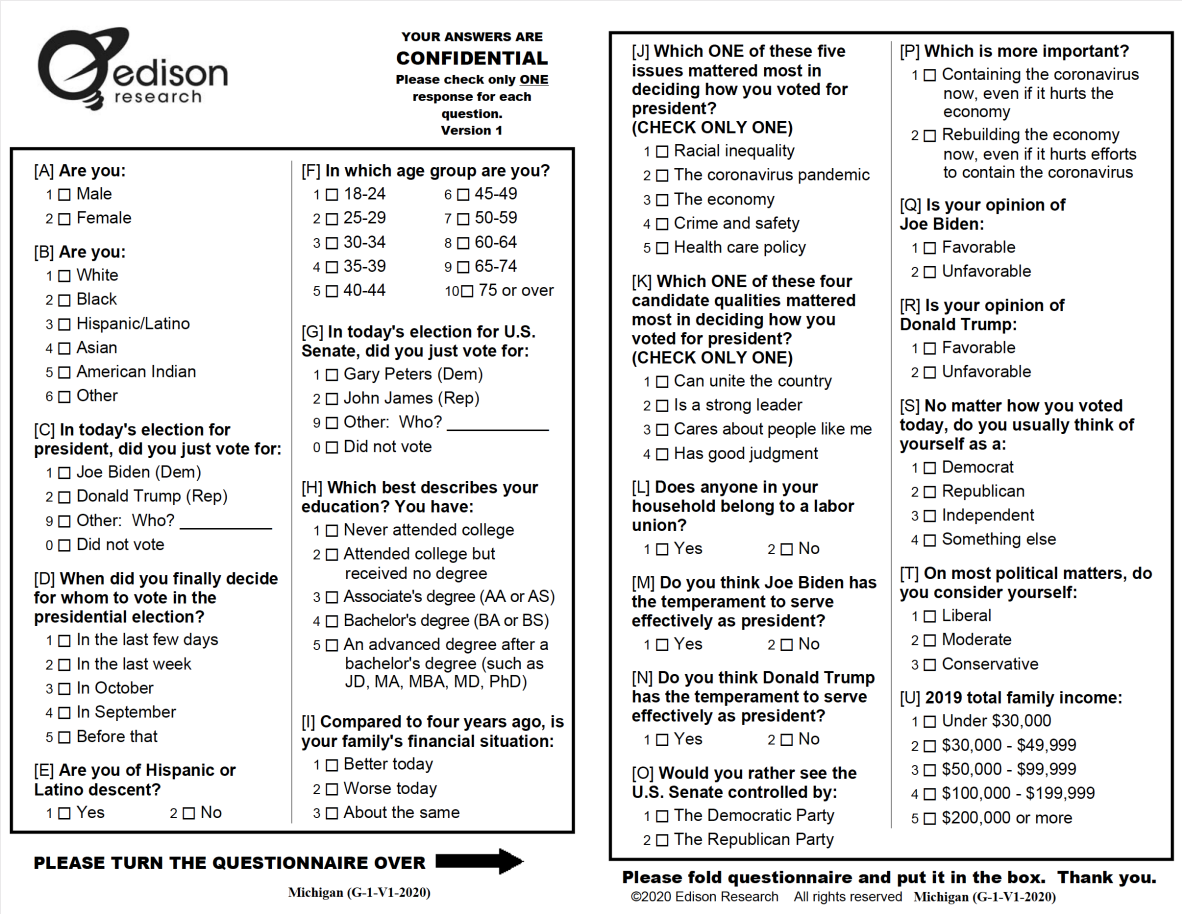
Exit Polls

- These polls ask voters to check a box in response to a question

Voter Joe Cieslo fills out an exit poll for the AP, CNN, Fox, NBC, CBS and ABC after voting at the Community Arts Center in Johnstown, Pennsylvania, on April 26, 2016. (Todd Berkey/The Tribune-Democrat via AP)

Exit Polls

- These polls ask voters to check a box in response to a question



edison research

YOUR ANSWERS ARE CONFIDENTIAL
Please check only ONE response for each question.
Version 1

[A] Are you:
1 ☐ Male
2 ☐ Female

[B] Are you:
1 ☐ White
2 ☐ Black
3 ☐ Hispanic/Latino
4 ☐ Asian
5 ☐ American Indian
6 ☐ Other

[C] In today's election for president, did you just vote for:
1 ☐ Joe Biden (Dem)
2 ☐ Donald Trump (Rep)
9 ☐ Other: Who? _____
0 ☐ Did not vote

[D] When did you finally decide for whom to vote in the presidential election?
1 ☐ In the last few days
2 ☐ In the last week
3 ☐ In October
4 ☐ In September
5 ☐ Before that

[E] Are you of Hispanic or Latino descent?
1 ☐ Yes 2 ☐ No

[F] In which age group are you?
1 ☐ 18-24 6 ☐ 45-49
2 ☐ 25-29 7 ☐ 50-59
3 ☐ 30-34 8 ☐ 60-64
4 ☐ 35-39 9 ☐ 65-74
5 ☐ 40-44 10 ☐ 75 or over

[G] In today's election for U.S. Senate, did you just vote for:
1 ☐ Gary Peters (Dem)
2 ☐ John James (Rep)
9 ☐ Other: Who? _____
0 ☐ Did not vote

[H] Which best describes your education? You have:
1 ☐ Never attended college
2 ☐ Attended college but received no degree
3 ☐ Associate's degree (AA or AS)
4 ☐ Bachelor's degree (BA or BS)
5 ☐ An advanced degree after a bachelor's degree (such as JD, MA, MBA, MD, PhD)

[I] Compared to four years ago, is your family's financial situation:
1 ☐ Better today
2 ☐ Worse today
3 ☐ About the same

PLEASE TURN THE QUESTIONNAIRE OVER

Michigan (G-1-V1-2020)

[J] Which ONE of these five issues mattered most in deciding how you voted for president? (CHECK ONLY ONE)
1 ☐ Racial inequality
2 ☐ The coronavirus pandemic
3 ☐ The economy
4 ☐ Crime and safety
5 ☐ Health care policy

[K] Which ONE of these four candidate qualities mattered most in deciding how you voted for president? (CHECK ONLY ONE)
1 ☐ Can unite the country
2 ☐ Is a strong leader
3 ☐ Cares about people like me
4 ☐ Has good judgment

[L] Does anyone in your household belong to a labor union?
1 ☐ Yes 2 ☐ No

[M] Do you think Joe Biden has the temperament to serve effectively as president?
1 ☐ Yes 2 ☐ No

[N] Do you think Donald Trump has the temperament to serve effectively as president?
1 ☐ Yes 2 ☐ No

[O] Would you rather see the U.S. Senate controlled by:
1 ☐ The Democratic Party
2 ☐ The Republican Party

[P] Which is more important?
1 ☐ Containing the coronavirus now, even if it hurts the economy
2 ☐ Rebuilding the economy now, even if it hurts efforts to contain the coronavirus

[Q] Is your opinion of Joe Biden:
1 ☐ Favorable
2 ☐ Unfavorable

[R] Is your opinion of Donald Trump:
1 ☐ Favorable
2 ☐ Unfavorable

[S] No matter how you voted today, do you usually think of yourself as a:
1 ☐ Democrat
2 ☐ Republican
3 ☐ Independent
4 ☐ Something else

[T] On most political matters, do you consider yourself:
1 ☐ Liberal
2 ☐ Moderate
3 ☐ Conservative

[U] 2019 total family income:
1 ☐ Under $30,000
2 ☐ $30,000 - $49,999
3 ☐ $50,000 - $99,999
4 ☐ $100,000 - $199,999
5 ☐ $200,000 or more

Please fold questionnaire and put it in the box. Thank you.

 Michigan (G-1-V1-2020)

Exit Polls

- **Predictive**: Use data to *predict* an outcome of interest.

    1. How many voters report voting for each candidate?
    2. What predicts support for Trump? For Biden?

- **Descriptive**: Use data to *describe* an event.

    1. How did Trump support vary by: gender? Age? Education?
    2. When did voters make up their minds?
    3. Why did voters support Trump or Biden?
    4. How do Trump and Biden voters vary in their opinions on: Covid? Race relations?

Getting Started

- Open `RStudio` and `require(tidyverse)`

```
require(tidyverse)
```

- Load the data

```
MI_raw <- readRDS('../data/MI2020_ExitPoll.rds')
# OR
MI_raw  <-
read_rds('https://github.com/jbisbee1/DS1000_F2023/raw/main/Lectures/3_
```

- And take a look

```
MI_raw
```

```
## # A tibble: 1,231 × 63
##       ID WEIGHT LALVOTERID     GROUP       ZIP DISTR…¹    Z1
##    <dbl>  <dbl> <chr>          <dbl+lbl> <dbl>   <dbl> <dbl>
##  1     9  0.405 LALMI6290066   3 [3]     49327       2    NA
##  2    66  1.81  LALMI2492492   1 [1]     48234      14    NA
```

Tabular Data

- Remember that we are using tabular data, where rows are observations (i.e., survey respondents) and columns are variables (i.e., vote choice)
- What is the **unit of observation** in these data?
  - Voters...which voters?
  - Voters in Michigan...all of them?
  - No just a random sample of those leaving the ballot box in 2020

The Process: Steps 1-3

- Step 1: **Look** at the data
- Step 2: **Wrangle** the data
- Step 3: **Analyze** the data

Step 1: Look at the data

- Goals:

1. Understand variable "types"
    - Most important: continuous versus categorical
2. Identify **missingness**: either `NA` or "unit non-response"
    - Unit non-response: observations who didn't provide information
3. Identify **skew** or other phenomena that require wrangling
    - Highly skewed data should be logged
    - Some variables should be transformed to **rates** or **proportions**

Step 1: Look at the data

- Methods:

  1. Just look: see the first few rows and the first few columns
  2. `glimpse()`: see the first few rows for every column
  3. `summary()`: see the lowest, highest, mean, median, and quartiles, along with missingness (**better for continuous data**)
  4. `count()`: see the number of observations in each category (**categorical data only**)

Step 1: Look at the data

```
glimpse(MI_raw)
```

```
## Rows: 1,231
## Columns: 63
## $ ID             <dbl> 9, 66, 225, 243, 286, 293, 365, 367…
## $ WEIGHT         <dbl> 0.4045421, 1.8052619, 0.8601966, 0.…
## $ LALVOTERID     <chr> "LALMI6290066", "LALMI2492492", "LA…
## $ GROUP          <dbl+lbl> 3, 1, 4, 1, 1, 1, 1, 1, 1, 1, 1…
## $ ZIP            <dbl> 49327, 48234, 48301, 48130, 49946, …
## $ DISTRICT       <dbl> 2, 14, 9, 7, 1, 4, 4, 2, 5, 12, 7, …
## $ Z1             <dbl> NA, NA, 48322, 48130, NA, NA, 48813…
## $ S1             <dbl+lbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…
## $ S2A            <dbl+lbl> 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2…
## $ S2B            <dbl+lbl> NA,  1,  1,  1,  2,  1,  1,  1,…
## $ S3             <dbl+lbl>  1, NA, NA, NA, NA, NA, NA, NA,…
## $ S4             <dbl+lbl>  1, NA, NA, NA, NA, NA, NA, NA,…
## $ VERSION        <dbl+lbl> 1, 2, 2, 1, 1, 2, 2, 2, 1, 1, 1…
## $ PRSMI20        <dbl+lbl> 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2…
## $ SENMI20        <dbl+lbl> 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2…
## $ TIME16         <dbl+lbl>  5, NA, NA,  5,  5, NA, NA, NA,…
## $ ISSUE20        <dbl+lbl>  5, NA, NA,  9,  1, NA, NA, NA,…
## $ QLT20          <dbl+lbl>  4, NA, NA,  4,  3, NA, NA, NA,…
```

Step 1-1: Variable Types

- **Continuous Variables**
  - `dbl`: numeric data stored with great precision. Can be manipulated with mathematical functions.
  - `int`: numeric data stored as integers. `R` typically treats `dbl` and `int` as interchangeable.
- **Categorical Variables**
  - `chr`: string data, consisting of letters (and/or numbers). **Cannot** be manipulated with mathematical functions.
  - `fct`: string data that is stored with a number. Typically used to define group membership.
- **Mixtures**
  - `lbl`: string data that is stored with a number. Similar to `fct` but less commonly occurring in base `R`.

Step 1-2: Missingness

- Two ways to indicate an observation is missing data

    1. `NA` code

    2. Some bespoke code for "unit non-response" (often `9`, or `99`, or some large value ending in `9` that is dissimilar from the rest of the data)

Step 1-2: Missingness

- To identify `NA`-style missingness, use `summary()`

```
summary(MI_raw %>% select(LALVOTERID,SEX,AGE10,PARTYID,LGBT,QLT20))
```

```
##   LALVOTERID             SEX            AGE10
##  Length:1231        Min.   :1.00   Min.   : 1.000
##  Class :character   1st Qu.:1.00   1st Qu.: 6.000
##  Mode  :character   Median :2.00   Median : 8.000
##                     Mean   :1.53   Mean   : 8.476
##                     3rd Qu.:2.00   3rd Qu.: 9.000
##                     Max.   :2.00   Max.   :99.000
##
##     PARTYID           LGBT           QLT20
##  Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :2.000   Median :2.000   Median :3.000
##  Mean   :2.236   Mean   :2.224   Mean   :2.956
##  3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:4.000
##  Max.   :9.000   Max.   :9.000   Max.   :9.000
##                  NA's   :615     NA's   :616
```

Step 1-2: Missingness

- To identify unit non-response, can use `count()`

```
MI_raw %>%
  count(PARTYID)
```

```
## # A tibble: 5 × 2
##   PARTYID                                    n
##   <dbl+lbl>                              <int>
## 1 1 [Democrat]                             425
## 2 2 [Republican]                           280
## 3 3 [Independent]                          416
## 4 4 [Something else]                        94
## 5 9 [[DON'T READ] Don’t know/refused]       16
```

- The number `9` indicates unit non-response for `PARTYID`

Step 1-2: Missingness

- To identify unit non-response, can use `count()`

```
MI_raw %>%
  count(AGE10)
```

```
## # A tibble: 11 × 2
##    AGE10                          n
##    <dbl+lbl>                  <int>
##  1  1 [18 and 24,]               33
##  2  2 [25 and 29,]               28
##  3  3 [30 and 34,]               42
##  4  4 [35 and 39,]               46
##  5  5 [40 and 44,]               78
##  6  6 [45 and 49,]               83
##  7  7 [50 and 59,]              274
##  8  8 [60 and 64,]              143
##  9  9 [65 and 74,]              290
## 10 10 [75 or over?]             199
## 11 99 [[DON'T READ] Refused]     15
```

- The number `99` indicates unit non-response for `AGE10`

Step 1-2: Missingness

- To identify unit non-response, can use `count()`

```
MI_raw %>%
  count(LGBT)
```

```
## # A tibble: 4 × 2
##   LGBT                                      n
##   <dbl+lbl>                             <int>
## 1  1 [Yes]                                 23
## 2  2 [No]                                 570
## 3  9 [[DON'T READ] Don’t know/Refused]     23
## 4 NA                                      615
```

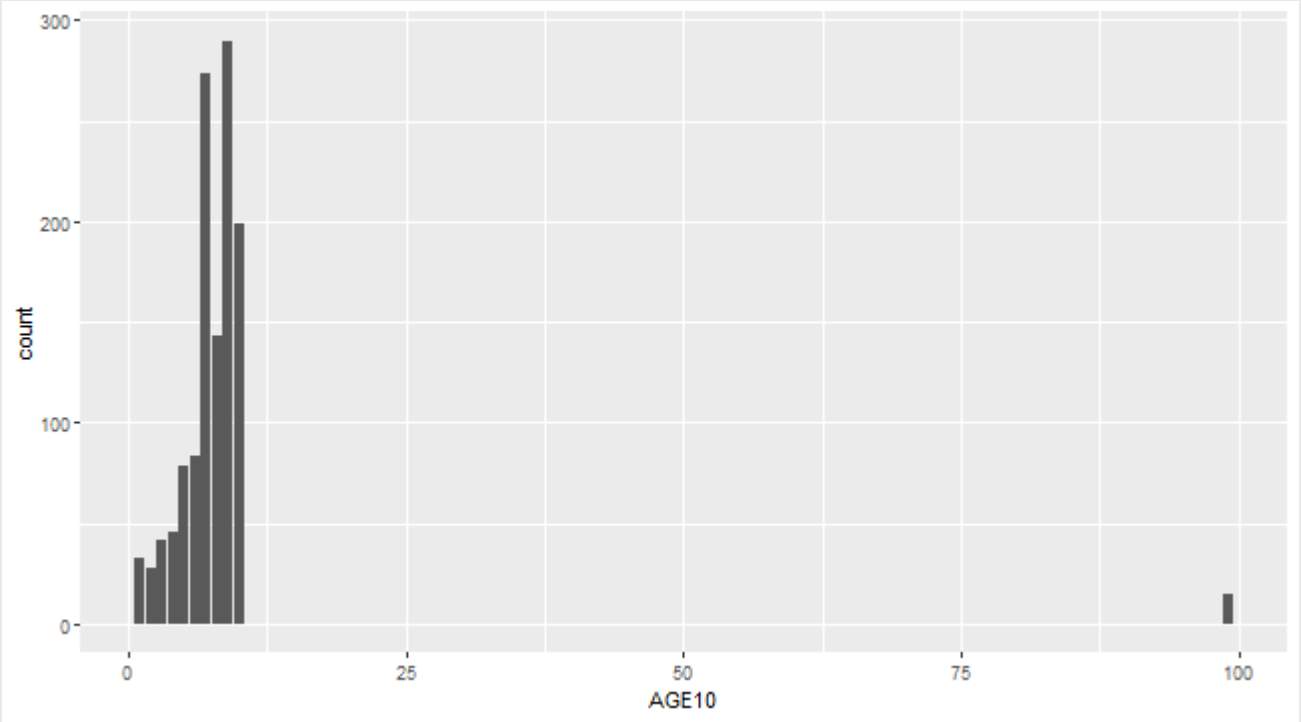
- The number `9` indicates unit non-response for `LGBT` **AND** this variable also has `NA`

Step 1-2: Missingness

- If the data isn't labelled, can still **look** to identify unit non-response
- Re-introducing...`ggplot()`
    - Only interested in **single variable**
- For now, we need two parts:
    1. `aes(x = [x-axis variable])`
    2. `geom_histogram()` (for continuous) or `geom_bar()` (for categorical)

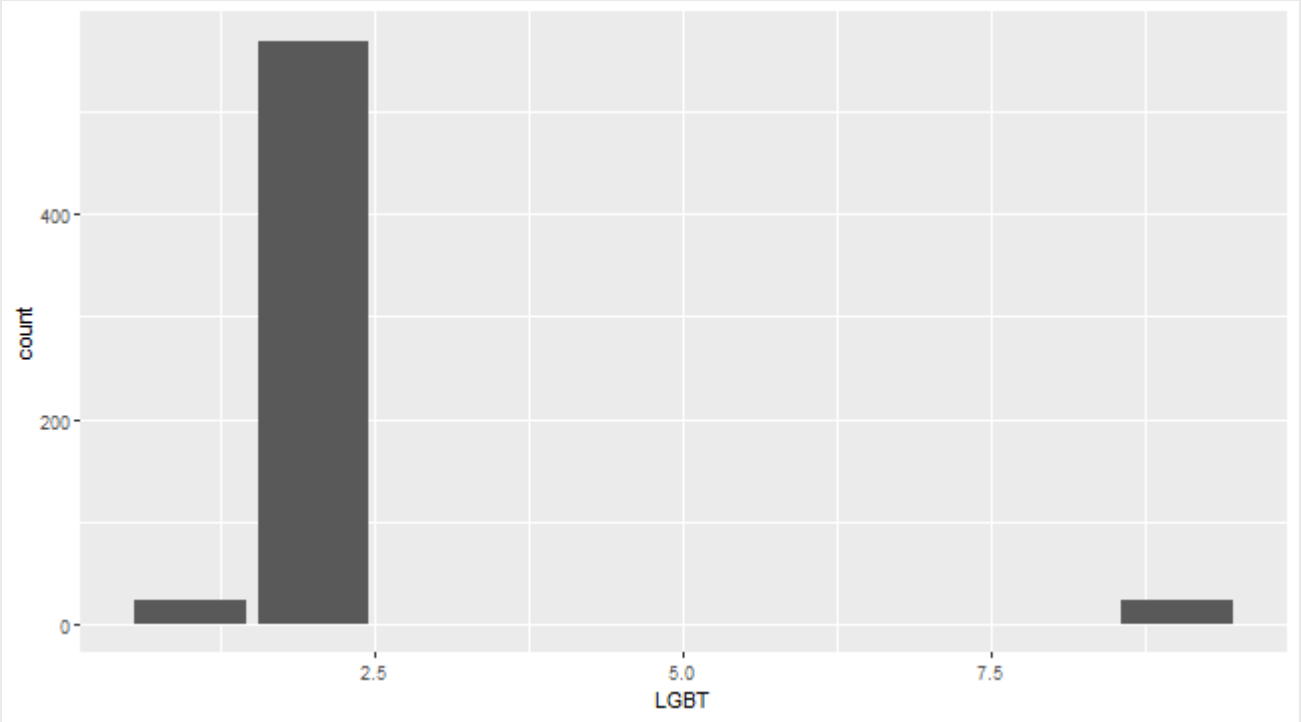
ggplot() Intro

```
MI_raw %>%
  ggplot(aes(x = AGE10)) +
  geom_bar()
```



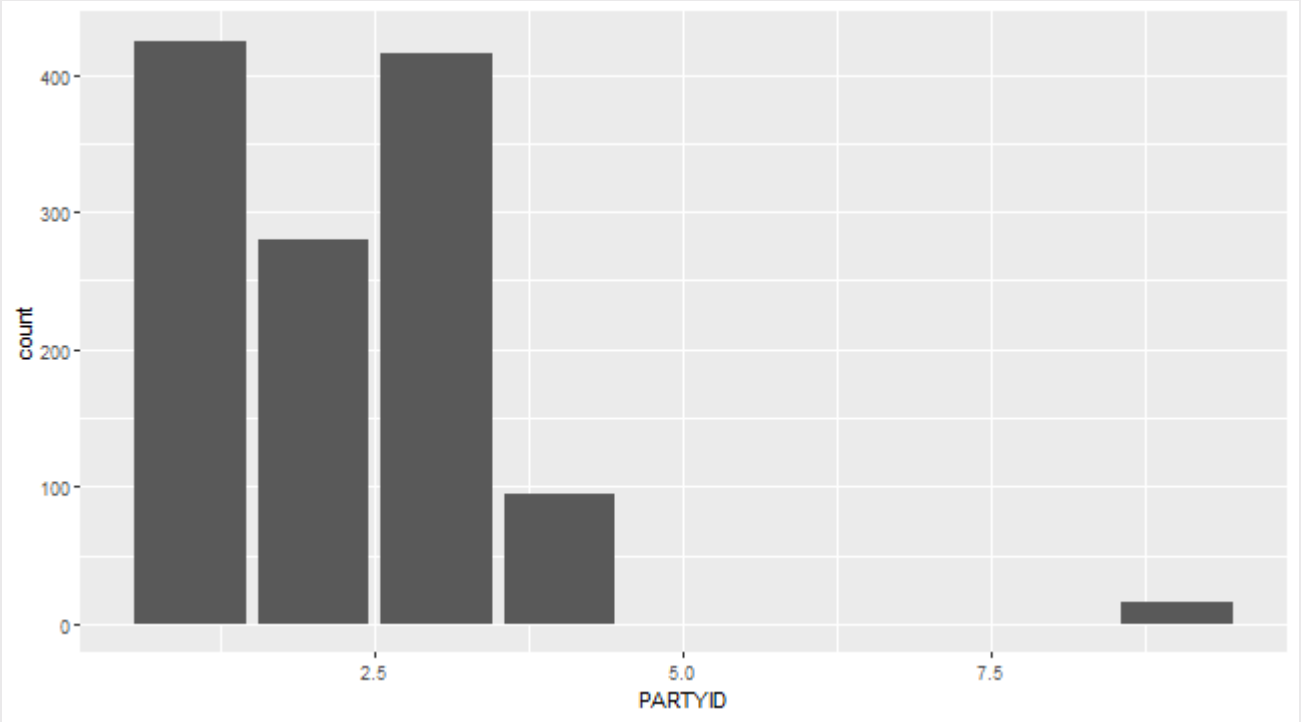
ggplot() Intro

```
MI_raw %>%
  ggplot(aes(x = LGBT)) +
  geom_bar()
```



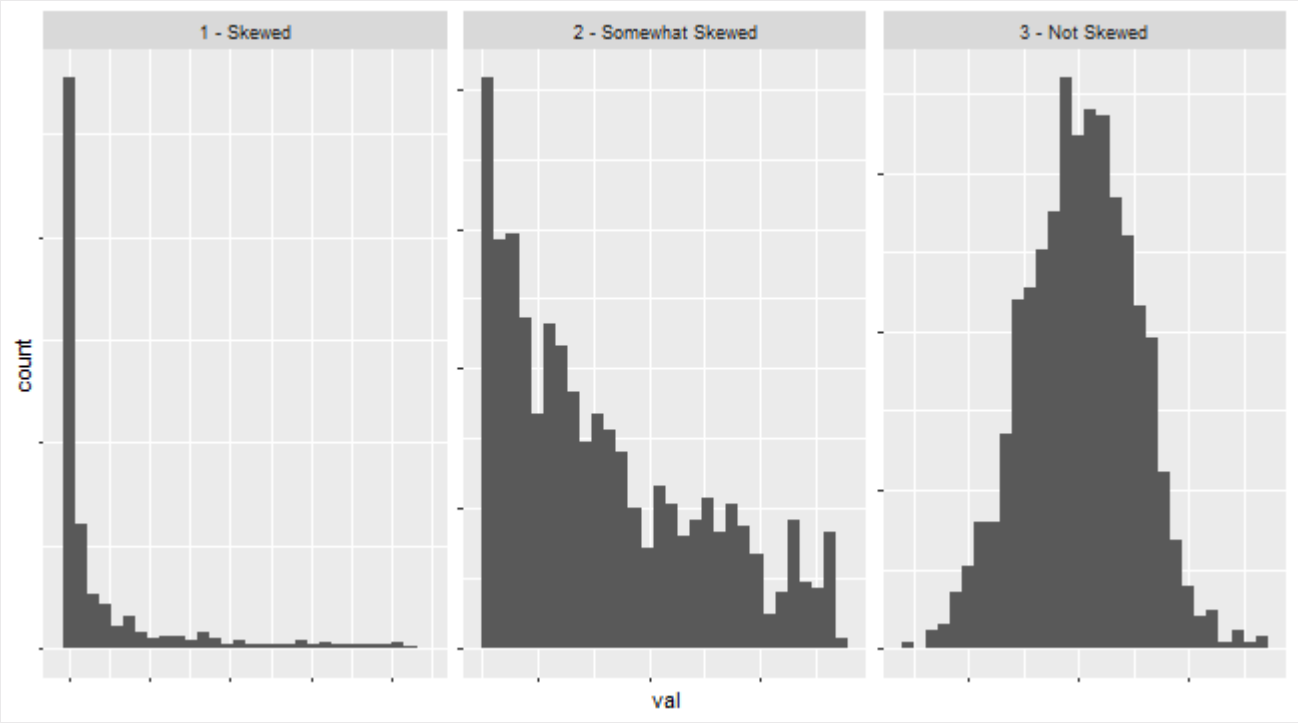
ggplot() Intro

```
MI_raw %>%
  ggplot(aes(x = PARTYID)) +
  geom_bar()
```



Step 1-3: Skew & Transformations

- To identify **skew**, we will also need `ggplot()`



- We will return to this later

Step 2: Wrangle

- Why wrangle data?

```
MI_raw %>%
  summarise(avgAge = mean(AGE10,na.rm=T))
```

```
## # A tibble: 1 × 1
##   avgAge
##    <dbl>
## 1   8.48
```

Step 2: Wrangle

- We over-estimate the average age category because R doesn't realize these are codes!

```
MI_raw %>%
  count(AGE10)
```

```
## # A tibble: 11 × 2
##    AGE10                          n
##    <dbl+lbl>                  <int>
##  1  1 [18 and 24,]               33
##  2  2 [25 and 29,]               28
##  3  3 [30 and 34,]               42
##  4  4 [35 and 39,]               46
##  5  5 [40 and 44,]               78
##  6  6 [45 and 49,]               83
##  7  7 [50 and 59,]              274
##  8  8 [60 and 64,]              143
##  9  9 [65 and 74,]              290
## 10 10 [75 or over?]             199
## 11 99 [[DON'T READ] Refused]     15
```

Step 2: Wrangle

- We need to convert **codes** for missing data to `NA`
- Use `mutate()` and `ifelse()` to replace `99` with `NA`
- `mutate()`:
  - Either creates a new column
  - Or changes an existing column
- `ifelse()`:
  - Does exactly as it says
  - `ifelse([LOGIC],[VALUE IF TRUE],[VALUE IF FALSE])`

Step 2: Wrangle

- If: `AGE10` is `99` (i.e., the unit non-response code)
- Then: give me an `NA` value
- Otherwise: give me whatever `AGE10` is

```
MI_raw %>%
  mutate(AGE10_new = ifelse(AGE10 == 99,NA,AGE10)) %>%
  select(AGE10,AGE10_new) %>%
  slice(c(35,1:20)) # Don't need to know slice() yet, ignore
```

```
## # A tibble: 21 × 2
##    AGE10                        AGE10_new
##    <dbl+lbl>                        <dbl>
##  1 99 [[DON'T READ] Refused]           NA
##  2  2 [25 and 29,]                      2
##  3 10 [75 or over?]                    10
##  4  7 [50 and 59,]                      7
##  5  9 [65 and 74,]                      9
##  6  8 [60 and 64,]                      8
##  7  7 [50 and 59,]                      7
##  8  9 [65 and 74,]                      9
```

Step 2: Wrangle

- The Assignment Operator (`<-`): `R`'s version of "Save As..."
- If we don't either:
    1. Overwrite the `MI_raw` object...
    2. Or create a new object...
    - `R` will not remember this new `AGE10_new` variable we created

```
MI_raw %>% select(AGE10_new)
```

```
## Error in `select()`:
## ! Can't subset columns that don't exist.
## ✕ Column `AGE10_new` doesn't exist.
```

Step 2: Wrangle

- Thus we need to use the **assignment operator** (`<-`)

```
MI_final <- MI_raw %>%
  mutate(AGE10_new = ifelse(AGE10 == 99,NA,AGE10))

MI_final %>%
  select(AGE10,AGE10_new) %>%
  slice(c(35,1:20)) # Don't need to know slice() yet, ignore
```

```
## # A tibble: 21 × 2
##    AGE10                        AGE10_new
##    <dbl+lbl>                        <dbl>
##  1 99 [[DON'T READ] Refused]           NA
##  2  2 [25 and 29,]                      2
##  3 10 [75 or over?]                    10
##  4  7 [50 and 59,]                      7
##  5  9 [65 and 74,]                      9
##  6  8 [60 and 64,]                      8
##  7  7 [50 and 59,]                      7
##  8  9 [65 and 74,]                      9
##  9  8 [60 and 64,]                      8
## 10  6 [45 and 49,]                      6
```

Step 2: Wrangle

- We can now get a more accurate measure of the average age category

```
MI_final %>%
  summarise(avgAge = mean(AGE10_new,na.rm=T))
```

```
## # A tibble: 1 × 1
##   avgAge
##    <dbl>
## 1   7.36
```

Variable Classes

- **How should we interpret this number?**
  - Does this mean that the average age of respondents is 7?
  - **NO**: we want to keep the definitions!

Converting to `chr`

- There is a helpful package called `haven` which will extract these labels as `factors`
- Install it with `install.packages("haven")` but DON'T `require()` it
- We can call on useful functions without `require()` by using two `::`

```
haven::as_factor(MI_final$AGE10) %>% head()
```

```
## [1] 25 and 29,  75 or over? 50 and 59,  65 and 74,
## [5] 60 and 64,  50 and 59,
## 11 Levels: 18 and 24, 25 and 29, 30 and 34, ... [DON'T READ]
Refused
```

Step 2: Wrangle

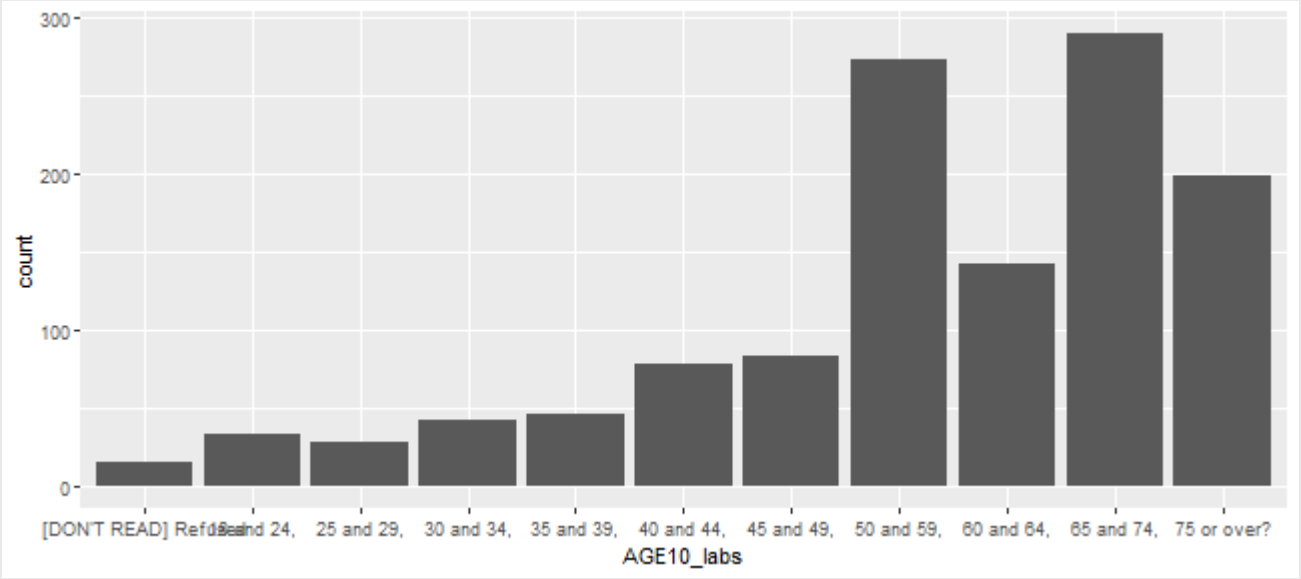
- Goal: Create a new column that converts the value for `AGE10` to a label
- To do this, use `mutate()` + the `haven::as_factor()` function + the `as.character()` function
  - Step 1: `as_factor()` function (from `haven`) converts to `fct` class
  - Step 2: `as.character()` function (from base `R`) converts to `chr` class

```
MI_final <- MI_final %>%
  mutate(AGE10_labs = as.character(haven::as_factor(AGE10)))
```

- **NB:** we **overwrite** `MI_final` to **add** the new column `AGE10_labs`

Step 2: Wrangle

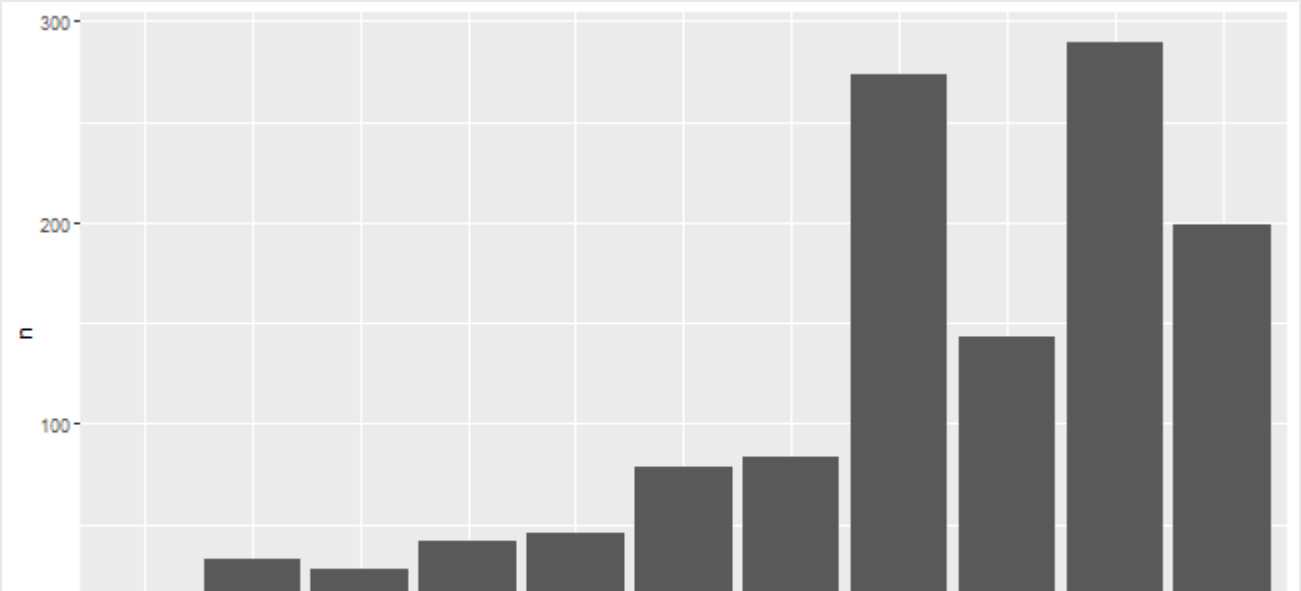
```
MI_final %>%
  ggplot(aes(x = AGE10_labs)) +
  geom_bar(stat = 'count')
```



Aside on `geom_bar()`

- Default: `stat = "count"` means show total observations in each category
- Override: `stat = "identity"` forces to show a specific value set in `aes()`

```
MI_final %>%
  count(AGE10_labs) %>%
  ggplot(aes(x = AGE10_labs,y = n)) +
  geom_bar(stat = 'identity')
```



Step 2: Wrangle

- Say we want the category labels for `PRSMI20`, `QLT20`, and `LGBT`

```
MI_final <- MI_final %>%
  mutate(preschoice = as.character(haven::as_factor(PRSMI20)),
         Qlty = as.character(haven::as_factor(QLT20)),
         LGBT_lab = as.character(haven::as_factor(LGBT)))

MI_final %>%
  select(preschoice,Qlty,LGBT_lab)
```

```
## # A tibble: 1,231 × 3
##    preschoice                  Qlty                  LGBT_…¹
##    <chr>                       <chr>                 <chr>
##  1 Joe Biden, the Democrat     Has good judgment     <NA>
##  2 Joe Biden, the Democrat     <NA>                  No
##  3 Joe Biden, the Democrat     <NA>                  No
##  4 Joe Biden, the Democrat     Has good judgment     <NA>
##  5 Joe Biden, the Democrat     Cares about people …  <NA>
##  6 Joe Biden, the Democrat     <NA>                  No
##  7 Joe Biden, the Democrat     <NA>                  No
##  8 Joe Biden, the Democrat     <NA>                  No
##  9 Donald Trump, the Republican Cares about people … <NA>
```

Preparing a new `tibble`

- Science guides us toward which variables to focus on

```
names(MI_final)
```

```
##  [1] "ID"             "WEIGHT"         "LALVOTERID"
##  [4] "GROUP"          "ZIP"            "DISTRICT"
##  [7] "Z1"             "S1"             "S2A"
## [10] "S2B"            "S3"             "S4"
## [13] "VERSION"        "PRSMI20"        "SENMI20"
## [16] "TIME16"         "ISSUE20"        "QLT20"
## [19] "TEMPBIDEN"      "TEMPTRUMP"      "CONTROLSEN"
## [22] "FINSIT"         "ECONVCORONA20"  "FAVBIDEN2"
## [25] "FAVTRUMP"       "FORCAND"        "NEWVOTER"
## [28] "NEC"            "HANDLEECON20"   "HANDLECORONA20"
## [31] "RACISM20"       "VOTE2016"       "COUNTACC"
## [34] "TRUMP4"         "CONTAINCOVID"   "COVIDHARDSHIP"
## [37] "AGE10"          "SEX"            "EDUC18"
## [40] "QRACEAI"        "LATINOS"        "PARTYID"
## [43] "PHIL3"          "INCOME20"       "BRNAGAIN"
## [46] "CHILD12"        "LGBT"           "UNIONHH12"
## [49] "QN5"            "QN6A"           "QN6B"
## [52] "QN6C"           "County"         "SMPFIPS"
```

Let's Wrangle!

- We are interested in presidential vote choice (`PRSMI20` or `preschoice`) and the quality of the candidate the respondent likes most (`QLT20` or `Qlty`)
- Look at these first

```
MI_final %>% select(preschoice,Qlty)
```

```
## # A tibble: 1,231 × 2
##    preschoice                  Qlty
##    <chr>                       <chr>
##  1 Joe Biden, the Democrat     Has good judgment
##  2 Joe Biden, the Democrat     <NA>
##  3 Joe Biden, the Democrat     <NA>
##  4 Joe Biden, the Democrat     Has good judgment
##  5 Joe Biden, the Democrat     Cares about people like me
##  6 Joe Biden, the Democrat     <NA>
##  7 Joe Biden, the Democrat     <NA>
##  8 Joe Biden, the Democrat     <NA>
##  9 Donald Trump, the Republican Cares about people like me
## 10 Joe Biden, the Democrat     Cares about people like me
## # … with 1,221 more rows
```

Let's Wrangle!

- We can get some preliminary descriptive info with `count()`

```
MI_final %>%
  count(preschoice)
```

```
## # A tibble: 6 × 2
##   preschoice                          n
##   <chr>                           <int>
## 1 Another candidate                  25
## 2 Donald Trump, the Republican      459
## 3 Joe Biden, the Democrat           723
## 4 Refused                            14
## 5 Undecided/Don’t know                4
## 6 Will/Did not vote for president     6
```

Let's Wrangle!

- We can get some preliminary descriptive info with `count()` (science!)

```
MI_final %>%
  count(preschoice,SEX)
```

```
## # A tibble: 12 × 3
##    preschoice                          SEX            n
##    <chr>                               <dbl+lbl>  <int>
##  1 Another candidate                   1 [Male]      17
##  2 Another candidate                   2 [Female]     8
##  3 Donald Trump, the Republican        1 [Male]     247
##  4 Donald Trump, the Republican        2 [Female]   212
##  5 Joe Biden, the Democrat             1 [Male]     304
##  6 Joe Biden, the Democrat             2 [Female]   419
##  7 Refused                             1 [Male]       7
##  8 Refused                             2 [Female]     7
##  9 Undecided/Don’t know                1 [Male]       3
## 10 Undecided/Don’t know                2 [Female]     1
## 11 Will/Did not vote for president     1 [Male]       1
## 12 Will/Did not vote for president     2 [Female]     5
```

Choosing the right function

- But `count()` is less useful for continuous variables

```
MI_final %>%
  count(WEIGHT)
```

```
## # A tibble: 411 × 2
##    WEIGHT     n
##     <dbl> <int>
##  1  0.100     1
##  2  0.113     1
##  3  0.119     1
##  4  0.133     2
##  5  0.141     1
##  6  0.142     1
##  7  0.144     1
##  8  0.146     1
##  9  0.147     1
## 10  0.149     5
## # … with 401 more rows
```

Choosing the right function

- Use `summary()` instead

```
MI_final %>%
  select(WEIGHT) %>%
  summary(WEIGHT)
```

```
##      WEIGHT
##  Min.   :0.1003
##  1st Qu.:0.3775
##  Median :0.8020
##  Mean   :1.0000
##  3rd Qu.:1.4498
##  Max.   :5.0853
```

Let's Wrangle!

- Recall some of the other helpful functions in `tidyverse`

```
MI_final %>%
  count(preschoice) %>%
  arrange(desc(n))
```

```
## # A tibble: 6 × 2
##   preschoice                          n
##   <chr>                           <int>
## 1 Joe Biden, the Democrat           723
## 2 Donald Trump, the Republican      459
## 3 Another candidate                  25
## 4 Refused                            14
## 5 Will/Did not vote for president     6
## 6 Undecided/Don’t know                4
```

Let's Wrangle!

- Let's look for missing data

```
MI_final %>%
  count(Qlty)
```

```
## # A tibble: 6 × 2
##   Qlty                                  n
##   <chr>                             <int>
## 1 [DON'T READ] Don’t know/refused    26
## 2 Can unite the country             125
## 3 Cares about people like me        121
## 4 Has good judgment                 205
## 5 Is a strong leader                138
## 6 <NA>                              616
```

Let's Wrangle!

- Drop rows where the respondent didn't answer the "quality" question

```
MI_final %>%
  drop_na(Qlty) %>%
  count(Qlty)
```

```
## # A tibble: 5 × 2
##   Qlty                                n
##   <chr>                           <int>
## 1 [DON'T READ] Don’t know/refused    26
## 2 Can unite the country             125
## 3 Cares about people like me        121
## 4 Has good judgment                 205
## 5 Is a strong leader                138
```

Let's Wrangle!

- But we still have the `[DON'T READ] Don't know/refused` respondents
- Convert "unit non-response" codes to `NA`
    - Need to look at them first!

```
MI_final %>%
  count(Qlty)
```

```
## # A tibble: 6 × 2
##   Qlty                                n
##   <chr>                           <int>
## 1 [DON'T READ] Don’t know/refused    26
## 2 Can unite the country             125
## 3 Cares about people like me        121
## 4 Has good judgment                 205
## 5 Is a strong leader                138
## 6 <NA>                              616
```

Let's Wrangle!

- Convert "unit non-response" codes to NA

```
MI_final <- MI_final %>%
  mutate(Qlty = ifelse(grepl("DON'T READ",Qlty),NA,Qlty))
MI_final %>%
  count(Qlty)
```

```
## # A tibble: 5 × 2
##   Qlty                          n
##   <chr>                     <int>
## 1 Can unite the country       125
## 2 Cares about people like me  121
## 3 Has good judgment           205
## 4 Is a strong leader          138
## 5 <NA>                        642
```

Let's Wrangle!

- Drop rows where the respondent didn't answer the "quality" question

```
MI_final %>%
  drop_na(Qlty) %>%
  count(Qlty)
```

```
## # A tibble: 4 × 2
##   Qlty                          n
##   <chr>                     <int>
## 1 Can unite the country       125
## 2 Cares about people like me  121
## 3 Has good judgment           205
## 4 Is a strong leader          138
```

Let's Wrangle!

- This is equivalent to `filter(!is.na(Qlty))`

```
MI_final %>%
  filter(!is.na(Qlty)) %>%
  count(Qlty)
```

```
## # A tibble: 4 × 2
##   Qlty                          n
##   <chr>                     <int>
## 1 Can unite the country       125
## 2 Cares about people like me  121
## 3 Has good judgment           205
## 4 Is a strong leader          138
```

Aside on filtering NA

- Make sure to specify which column has the missing data!
- Otherwise get "complete cases" (respondents with no missing data)

```
MI_final %>%
  drop_na() %>%
  count(QLT20)
```

```
## # A tibble: 0 × 2
## # … with 2 variables: QLT20 <dbl+lbl>, n <int>
```

- Same as

```
MI_final %>%
  filter(complete.cases(.)) %>%
  count(QLT20)
```

```
## # A tibble: 0 × 2
## # … with 2 variables: QLT20 <dbl+lbl>, n <int>
```

Let's Wrangle!

- Finally, let's subset the data to focus only on the other variables we are interested in

```
MI_final <- MI_final %>%
  select(SEX,AGE10,PARTYID,
         WEIGHT,QRACEAI,EDUC18,LGBT,
         BRNAGAIN,LATINOS,RACISM20,
         QLT20,Qlty,PRSMI20,preschoice)
```

- This is our **wrangled** data

```
MI_final
```

```
## # A tibble: 1,231 × 14
##    SEX          AGE10             PARTYID     WEIGHT QRACEAI
##    <dbl+lbl>    <dbl+lbl>         <dbl+lbl>    <dbl> <dbl+lbl>
##  1 2 [Female]    2 [25 and 29,]  3 [Indepen…  0.405  1 [White]
##  2 2 [Female]   10 [75 or over…  1 [Democra…  1.81   2 [Black]
##  3 2 [Female]    7 [50 and 59,]  1 [Democra…  0.860  1 [White]
##  4 1 [Male]      9 [65 and 74,]  3 [Indepen…  0.199  1 [White]
##  5 2 [Female]    8 [60 and 64,]  3 [Indepen…  0.177  1 [White]
```

Finishing up

- And now save both the raw data and our prepared data together

```
save(MI_raw,MI_final,file = '../output/MI_prepped.RData')
```

- An `.RData` file can contain multiple objects
- When we load it, we get access to all the objects we've created

```
rm(MI_raw,MI_final)

load('../output/MI_prepped.RData')
```

- **NB: different file extensions require different functions!**
  - `load` for `.RData` files
  - `read_rds` for `.rds` files
  - `read_csv` for `.csv` files

Conclusion

- What you need to know from today
    1. Why `RStudio` + `R` are great for **wrangling**
    2. `mutate()`
    3. `ifelse()`
    4. Assignment Operator ("Save As..."): `<-`

Quiz & Homework

- Go to Brightspace and take the **5th** quiz
    - The password to take the quiz is ####
- **Homework:**
    1. Work through Data_Wrangling_hw.Rmd
    2. Problem Set 2 (on Brightspace)