Fall 2023

DIME Continuing Education

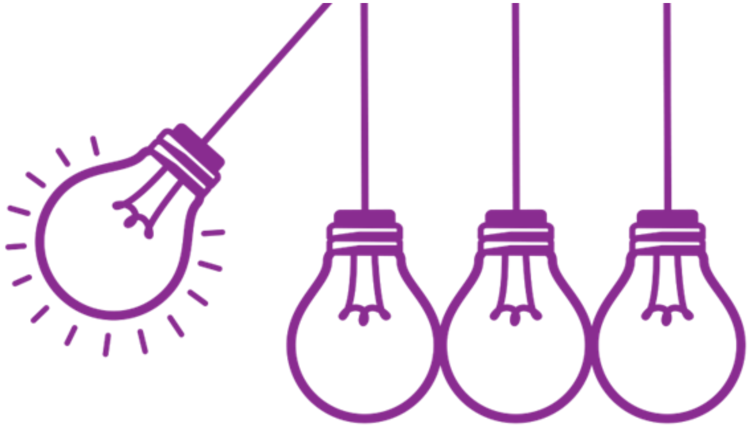
DATA ANALYSIS WITH TIDYVERSE

Mer Reyes Retana

December 7, 2023

THE WORLD BANK
IBRD • IDA | WORLD BANK GROUP

i2i DIME
TRANSFORM DEVELOPMENT



Data Analyis with Tidyverse

DIME Analytics
Mer Reyes Retana
December 7th, 2023

You can find this presentation and all the materials I will use for the training in this repository

Table of contents

1. Introduction
2. Tidy Workflow
3. Tidy Verbs
4. Data Visualization
5. Conclusion and References

Introduction

Motivation

The Critical Role of Data Preparation

Data Science and Economics:

- Significant time invested in data preparation
- Survey results indicate over 50% of time spent in cleaning and getting data ready

The Challenge

- Data preparation is often challenging and time-consuming
- A crucial step in achieving accurate and reliable analysis

Solution: Tidyverse in R

- `tidyverse`: A collection of R packages designed for data science
- Enhances efficiency and bearability in data preparation

Motivation

Who Should Attend?

- This session is open to **anyone who knows R** and is looking to **improve the style and efficiency of their code**.

Motivation

Who Should Attend?

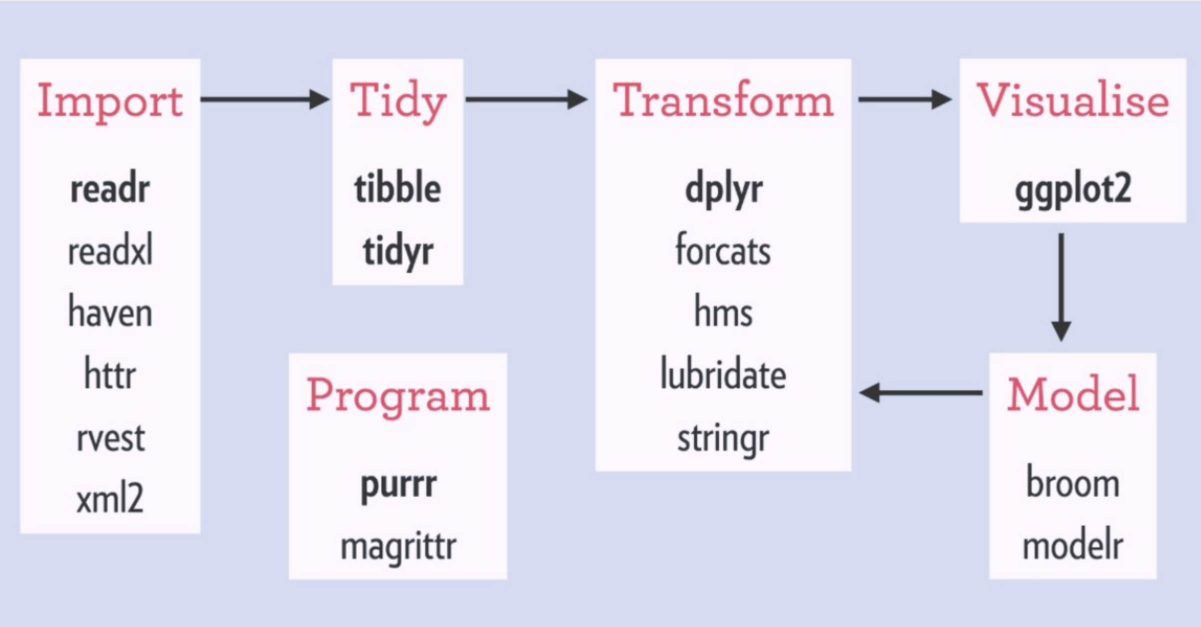
- This session is open to **anyone who knows R** and is looking to **improve the style and efficiency of their code**.

Requirements

- **Access to Materials**: Clone or download this repository to access the presentation materials.
- **No need to code along live**. Just sit back, and absorb the concepts.
- **After the session** you can work through the example (example/analysis_tidyverse_examples.R) at your own pace, enhancing your understanding of the Tidyverse.

What is the Tidyverse?

- **The best way to use R** is by using the multiple packages it offers.
- **One such set of packages** that we are focusing on is the `tidyverse`.



- A cohesive series of R packages
- Designed with a common philosophy
- Streamlines data import, tidying, transformation, visualization, and modeling

Tidy Workflow

Prepare your Workspace

- Install the `tidyverse` packages:

```
install.packages("tidyverse")
install.packages("dslabs")
```

- Download/Clone this repository in your computer.

- Ready to dive in.

Prepare your Workspace

Script Set-up

- We start by adding the libraries we will use in the project.

```r
library(dplyr)
library(tidyr)
library(ggplot2)
library(dslabs)
```

- Read the data

```r
data(polls_us_election_2016, package = "dslabs") # load data from package
polls_us_election_2016 <- as_tibble(polls_us_election_2016) # convert to a tibble
```

Data: 2016 US election polls from the `dslabs` package

- This dataset contains **real** data on polls made during the 2016 US Presidential elections and compiled by fivethirtyeight

```r
library(dslabs)
library(tidyverse)
data(polls_us_election_2016, package = "dslabs") # load data from package
polls_us_election_2016 <- as_tibble(polls_us_election_2016) # convert to a tibble
head(polls_us_election_2016) # show first 6 lines of first 6 variables
```

```
## # A tibble: 6 × 15
##   state startdate  enddate    pollster                 grade samplesize population
##   <fct> <date>     <date>     <fct>                    <fct>      <int> <chr>
## 1 U.S.  2016-11-03 2016-11-06 ABC News/Washington P… A+          2220 lv
## 2 U.S.  2016-11-01 2016-11-07 Google Consumer Surve… B          26574 lv
## 3 U.S.  2016-11-02 2016-11-06 Ipsos                    A-          2195 lv
## 4 U.S.  2016-11-04 2016-11-07 YouGov                   B           3677 lv
## 5 U.S.  2016-11-03 2016-11-06 Gravis Marketing         B-         16639 rv
## 6 U.S.  2016-11-03 2016-11-06 Fox News/Anderson Rob… A            1295 lv
```

Data: 2016 US election polls from the `dslabs` package

What variables does this dataset contain?

```
str(polls_us_election_2016) # Displays structures of R objects
```

```
## tibble [4,208 × 15] (S3: tbl_df/tbl/data.frame)
##  $ state           : Factor w/ 57 levels "Alabama","Alaska",..: 50 50 50 50 50 50 50 50 37 50 ...
##  $ startdate       : Date[1:4208], format: "2016-11-03" "2016-11-01" ...
##  $ enddate         : Date[1:4208], format: "2016-11-06" "2016-11-07" ...
##  $ pollster        : Factor w/ 196 levels "ABC News/Washington Post",..: 1 63 81 194 65 55 18 113 195
##  $ grade           : Factor w/ 10 levels "D","C-","C","C+",..: 10 6 8 6 5 9 8 8 NA 8 ...
##  $ samplesize      : int [1:4208] 2220 26574 2195 3677 16639 1295 1426 1282 8439 1107 ...
##  $ population      : chr [1:4208] "lv" "lv" "lv" "lv" ...
##  $ rawpoll_clinton : num [1:4208] 47 38 42 45 47 ...
##  $ rawpoll_trump   : num [1:4208] 43 35.7 39 41 43 ...
##  $ rawpoll_johnson : num [1:4208] 4 5.46 6 5 3 3 5 6 6 7.1 ...
##  $ rawpoll_mcmullin: num [1:4208] NA NA NA NA NA NA NA NA NA NA ...
##  $ adjpoll_clinton : num [1:4208] 45.2 43.3 42 45.7 46.8 ...
##  $ adjpoll_trump   : num [1:4208] 41.7 41.2 38.8 40.9 42.3 ...
```

The pipe operator

- Reading and understanding multiple operations can be difficult.
- Parameters are typically assigned after the function name using `()`.

```
summarise(
  group_by(
    filter(polls_us_election_2016, !is.na(section_type)),
    region,
    section_type
  ),
  n = n()
)
```

The pipe operator

- This approach can lead to complex and unreadable code.
- With the pipe code reads from left to right, top to bottom, which is more intuitive.

|> or %>% can be read as "then" and simplifies code structure.

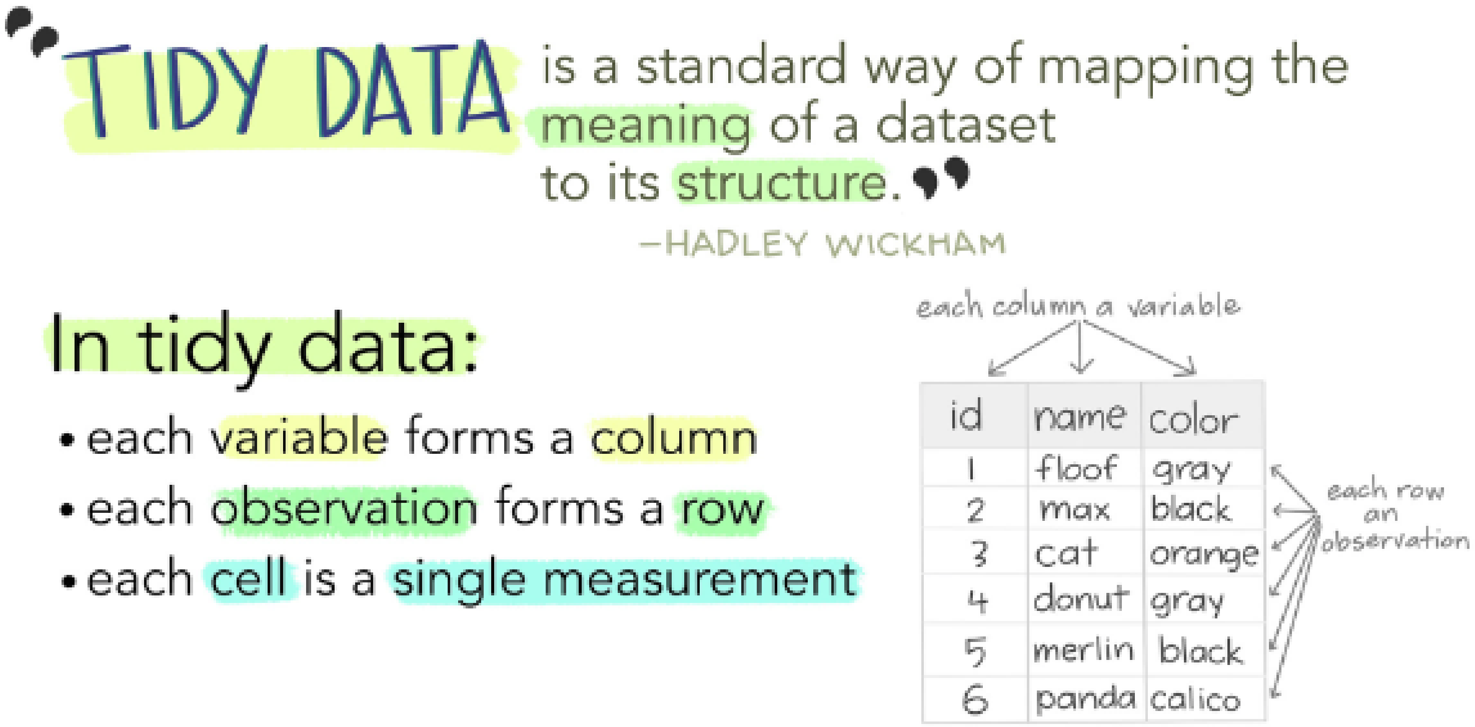
```
polls_us_election_2016 %>%
  filter(!is.na(section_type)) %>%
  group_by(region, section_type) %>%
  summarise(
    n = n()
  )
```

Tip: Use Shift + Ctrl/Cmd + M as a shortcut for the pipe operator.

Tidy Verbs

Tidy Verbs

The Tidyverse packages works better and should be ideally be used with tidy data or to produce tidy data.



> "TIDY DATA is a standard way of mapping the meaning of a dataset to its structure."
>
> —HADLEY WICKHAM

In tidy data:

- each variable forms a column
- each observation forms a row
- each cell is a single measurement

id	name	color
1	floof	gray
2	max	black
3	cat	orange
4	donut	gray
5	merlin	black
6	panda	calico

Tidy Verbs

The `tidyverse` ecosystem is composed of multiple packages, each equipped with specific "verbs" to streamline the data workflow process. We'll focus on the verbs from two pivotal packages: `tidyr` and `dplyr`.

- You are ***highly encouraged*** to read through Hadley Wickham's chapter. It's clear and concise.
- Also check out these great "cheatsheets" dplyr and tidyr.

Tidy Verbs

- The packages are organized around a set of **verbs**, i.e. *actions* to be taken.
- We operate on `data.frames` or `tibbles` (*nicer looking* data.frames.)

Tidy Verbs

- The packages are organized around a set of **verbs**, i.e. *actions* to be taken.
- We operate on `data.frames` or `tibbles` (*nicer looking* data.frames.)
- All *verbs* work as follows:

$$\text{verb}(\underbrace{\text{data.frame}}_{\text{1st argument}}, \underbrace{\text{what to do}}_{\text{2nd argument}})$$

Tidy Verbs

- The packages are organized around a set of **verbs**, i.e. *actions* to be taken.
- We operate on `data.frames` or `tibbles` (*nicer looking* data.frames.)
- All *verbs* work as follows:

$$\text{verb}(\underbrace{\text{data.frame}}_{\text{1st argument}}, \underbrace{\text{what to do}}_{\text{2nd argument}})$$

- Alternatively you can (**should**) use the `pipe` operator `%>%`:

$$\underbrace{\text{data.frame}}_{\text{1st argument}} \quad \underbrace{\%>\%}_{\text{"pipe" operator}} \quad \text{verb}(\underbrace{\text{what to do}}_{\text{2nd argument}})$$

Main `tidyr` Verbs

- `tidyr` is designed to tidy your data.

1. `pivot_longer()`: Lengthens data, increasing the number of rows and decreasing the number of columns.

Main `tidyr` Verbs

- `tidyr` is designed to tidy your data.

1. `pivot_longer()`: Lengthens data, increasing the number of rows and decreasing the number of columns.

2. `pivot_wider()`: Widens data, increasing the number of columns and decreasing the number of rows.

Main `tidyr` Verbs

- `tidyr` is designed to tidy your data.

1. `pivot_longer()`: Lengthens data, increasing the number of rows and decreasing the number of columns.

2. `pivot_wider()`: Widens data, increasing the number of columns and decreasing the number of rows.

3. `separate()`: Splits a column into multiple columns.

Main `tidyr` Verbs

- `tidyr` is designed to tidy your data.

1. `pivot_longer()`: Lengthens data, increasing the number of rows and decreasing the number of columns.
2. `pivot_wider()`: Widens data, increasing the number of columns and decreasing the number of rows.
3. `separate()`: Splits a column into multiple columns.
4. `unite()`: Combines multiple columns into a single column.

Main `tidyr` Verbs

- `tidyr` is designed to tidy your data.

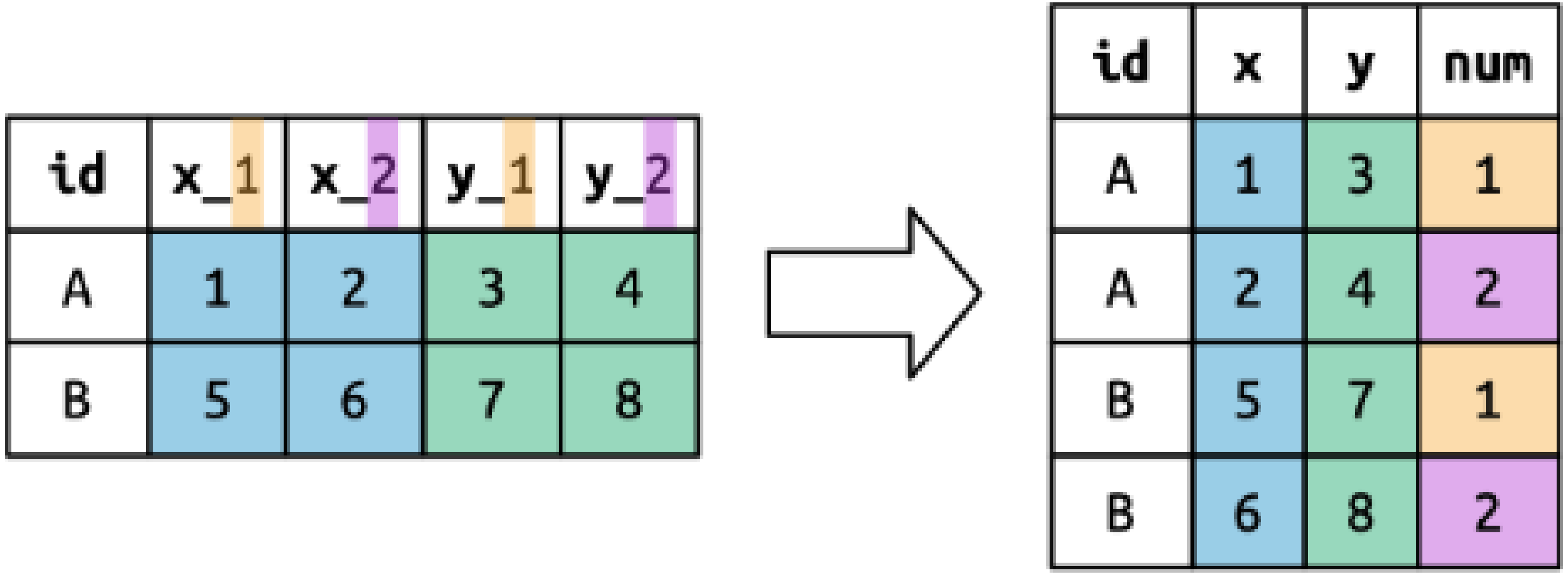
1. `pivot_longer()`: Lengthens data, increasing the number of rows and decreasing the number of columns.
2. `pivot_wider()`: Widens data, increasing the number of columns and decreasing the number of rows.
3. `separate()`: Splits a column into multiple columns.
4. `unite()`: Combines multiple columns into a single column.

- These verbs help transform data frames or tibbles to a tidy format, where each variable is a column, each observation is a row, and each type of observational unit forms a table.

tidyr::pivot_longer()

Some of the **most useful** verbs in the tidyverse package, and particularly important to achieve tidy data, are the reshape verbs.

The first reshape operation involves making datasets longer by increasing the number of rows and decreasing the number of columns.

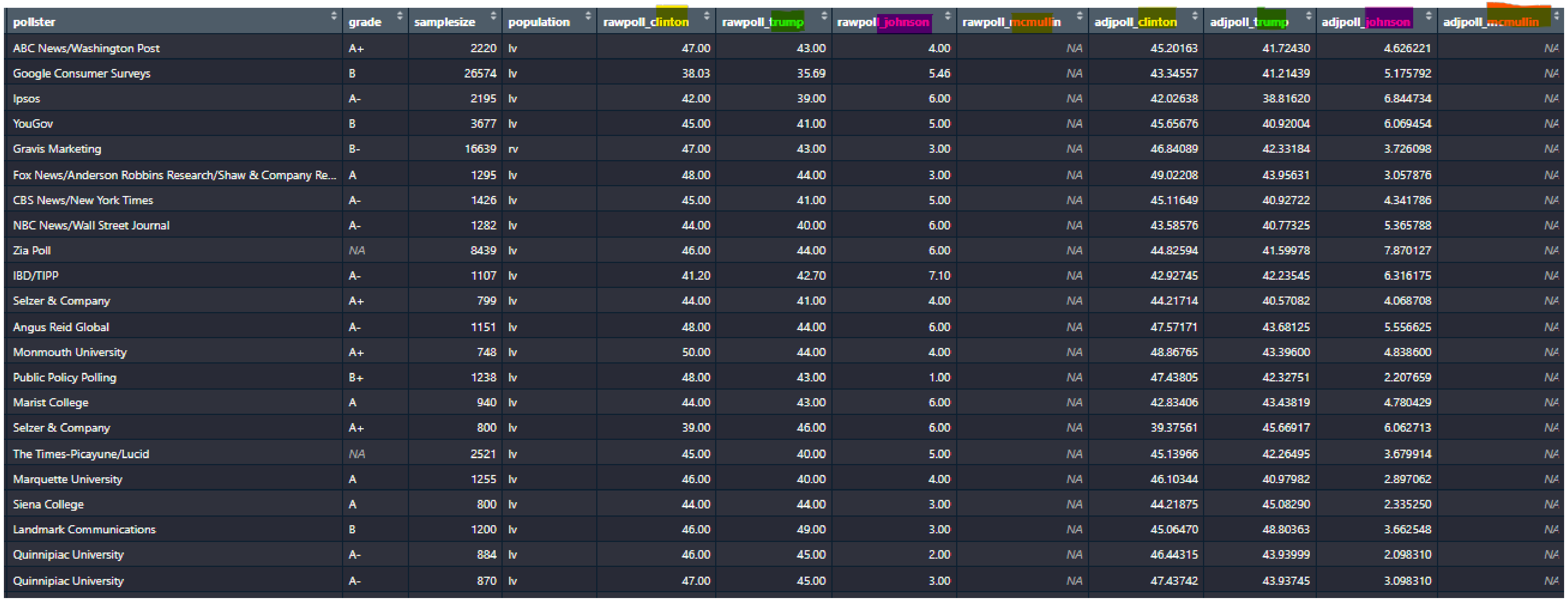


id	x_1	x_2	y_1	y_2
A	1	2	3	4
B	5	6	7	8

id	x	y	num
A	1	3	1
A	2	4	2
B	5	7	1
B	6	8	2

`tidyr::pivot_longer()`

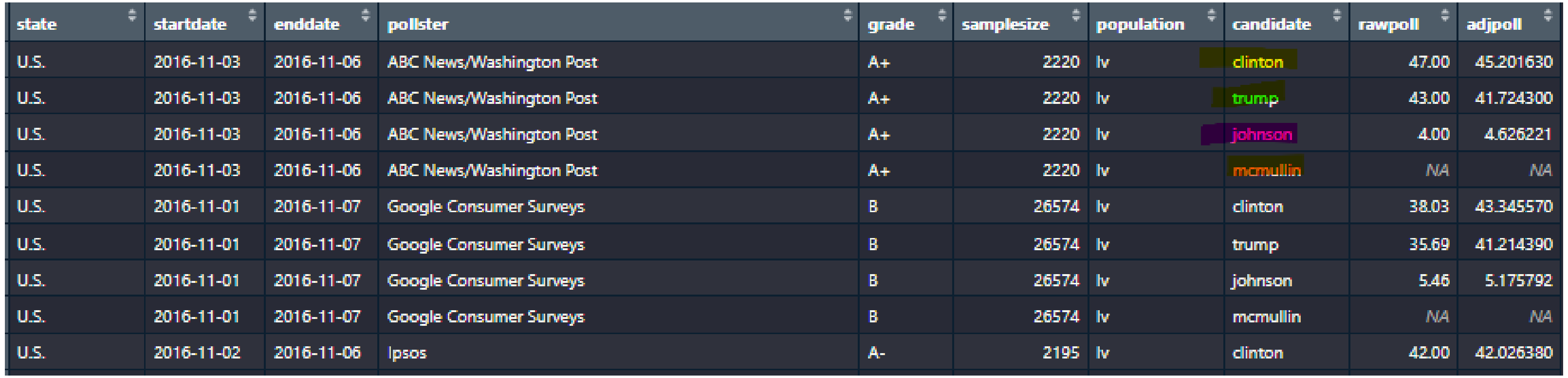
Example: we have rawpoll and adjpoll for every candidate in a wide format. Perhaps we want to have this information in two variables (instead of 8)



pollster	grade	samplesize	population	rawpoll_clinton	rawpoll_trump	rawpoll_johnson	rawpoll_mcmullin	adjpoll_clinton	adjpoll_trump	adjpoll_johnson	adjpoll_mcmullin
ABC News/Washington Post	A+	2220	lv	47.00	43.00	4.00	NA	45.20163	41.72430	4.626221	NA
Google Consumer Surveys	B	26574	lv	38.03	35.69	5.46	NA	43.34557	41.21439	5.175792	NA
Ipsos	A-	2195	lv	42.00	39.00	6.00	NA	42.02638	38.81620	6.844734	NA
YouGov	B	3677	lv	45.00	41.00	5.00	NA	45.65676	40.92004	6.069454	NA
Gravis Marketing	B-	16639	rv	47.00	43.00	3.00	NA	46.84089	42.33184	3.726098	NA
Fox News/Anderson Robbins Research/Shaw & Company Re...	A	1295	lv	48.00	44.00	3.00	NA	49.02208	43.95631	3.057876	NA
CBS News/New York Times	A-	1426	lv	45.00	41.00	5.00	NA	45.11649	40.92722	4.341786	NA
NBC News/Wall Street Journal	A-	1282	lv	44.00	40.00	6.00	NA	43.58576	40.77325	5.365788	NA
Zia Poll	NA	8439	lv	46.00	44.00	6.00	NA	44.82594	41.59978	7.870127	NA
IBD/TIPP	A-	1107	lv	41.20	42.70	7.10	NA	42.92745	42.23545	6.316175	NA
Selzer & Company	A+	799	lv	44.00	41.00	4.00	NA	44.21714	40.57082	4.068708	NA
Angus Reid Global	A-	1151	lv	48.00	44.00	6.00	NA	47.57171	43.68125	5.556625	NA
Monmouth University	A+	748	lv	50.00	44.00	4.00	NA	48.86765	43.39600	4.838600	NA
Public Policy Polling	B+	1238	lv	48.00	43.00	1.00	NA	47.43805	42.32751	2.207659	NA
Marist College	A	940	lv	44.00	43.00	6.00	NA	42.83406	43.43819	4.780429	NA
Selzer & Company	A+	800	lv	39.00	46.00	6.00	NA	39.37561	45.66917	6.062713	NA
The Times-Picayune/Lucid	NA	2521	lv	45.00	40.00	5.00	NA	45.13966	42.26495	3.679914	NA
Marquette University	A	1255	lv	46.00	40.00	4.00	NA	46.10344	40.97982	2.897062	NA
Siena College	A	800	lv	44.00	44.00	3.00	NA	44.21875	45.08290	2.335250	NA
Landmark Communications	B	1200	lv	46.00	49.00	3.00	NA	45.06470	48.80363	3.662548	NA
Quinnipiac University	A-	884	lv	46.00	45.00	2.00	NA	46.44315	43.93999	2.098310	NA
Quinnipiac University	A-	870	lv	47.00	45.00	3.00	NA	47.43742	43.93745	3.098310	NA

tidyr::pivot_longer()

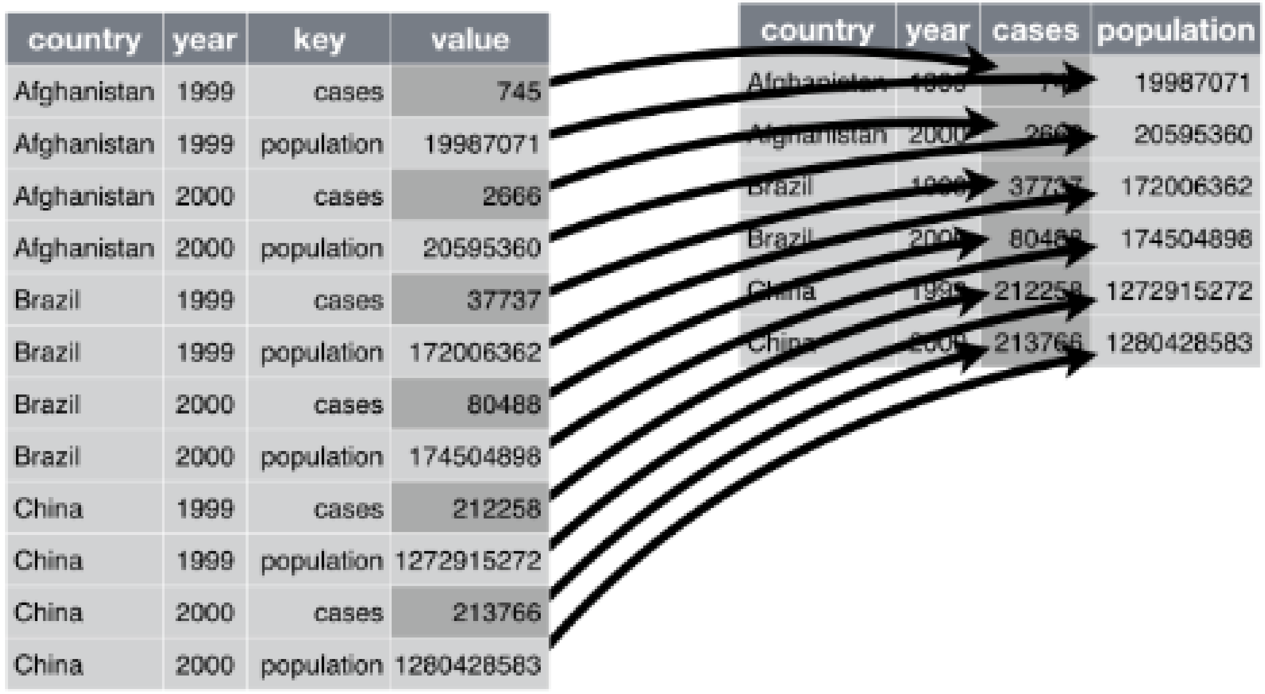
```
polls_us_election_2016 %>%
  pivot_longer(
    cols = c(rawpoll_clinton:adjpoll_trump),
    names_to = c(".value", "candidate"),
    names_pattern = "(rawpoll|adjpoll)_(.*)"
  )
```



state	startdate	enddate	pollster	grade	samplesize	population	candidate	rawpoll	adjpoll
U.S.	2016-11-03	2016-11-06	ABC News/Washington Post	A+	2220	lv	clinton	47.00	45.201630
U.S.	2016-11-03	2016-11-06	ABC News/Washington Post	A+	2220	lv	trump	43.00	41.724300
U.S.	2016-11-03	2016-11-06	ABC News/Washington Post	A+	2220	lv	johnson	4.00	4.626221
U.S.	2016-11-03	2016-11-06	ABC News/Washington Post	A+	2220	lv	mcmullin	NA	NA
U.S.	2016-11-01	2016-11-07	Google Consumer Surveys	B	26574	lv	clinton	38.03	43.345570
U.S.	2016-11-01	2016-11-07	Google Consumer Surveys	B	26574	lv	trump	35.69	41.214390
U.S.	2016-11-01	2016-11-07	Google Consumer Surveys	B	26574	lv	johnson	5.46	5.175792
U.S.	2016-11-01	2016-11-07	Google Consumer Surveys	B	26574	lv	mcmullin	NA	NA
U.S.	2016-11-02	2016-11-06	Ipsos	A-	2195	lv	clinton	42.00	42.026380

tidyr::pivot_wider()

This is the contrary of the previous command, there are cases where we have information in a long format and we would like to make it wider.



country	year	key	value
Afghanistan	1999	cases	745
Afghanistan	1999	population	19987071
Afghanistan	2000	cases	2666
Afghanistan	2000	population	20595360
Brazil	1999	cases	37737
Brazil	1999	population	172006362
Brazil	2000	cases	80488
Brazil	2000	population	174504898
China	1999	cases	212258
China	1999	population	1272915272
China	2000	cases	213766
China	2000	population	1280428583

table2

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

tidyr::pivot_wider()

Example: For simplicity, I will wide the previous dataframe, but here you can find multiple examples.

```
wider_dataframes <- longer_dataframe %>%
  pivot_wider(
    names_from = candidate,
    values_from = c(rawpoll, adjpoll),
    values_fn = list
  )
```

tidyr::separate()

Splits a column into multiple columns

Example: Divide enddate into year, month and day.

```r
polls_us_election_2016 %>%
  separate(enddate, into = c("year", "month", "day"), sep = "-")
```

```
## # A tibble: 4,208 × 17
##    state      startdate  year  month day   pollster  grade samplesize population
##    <fct>      <date>     <chr> <chr> <chr> <fct>     <fct>      <int> <chr>
##  1 U.S.       2016-11-03 2016  11    06    ABC News… A+          2220 lv
##  2 U.S.       2016-11-01 2016  11    07    Google C… B          26574 lv
##  3 U.S.       2016-11-02 2016  11    06    Ipsos     A-          2195 lv
##  4 U.S.       2016-11-04 2016  11    07    YouGov    B           3677 lv
##  5 U.S.       2016-11-03 2016  11    06    Gravis M… B-         16639 rv
##  6 U.S.       2016-11-03 2016  11    06    Fox News… A           1295 lv
##  7 U.S.       2016-11-02 2016  11    06    CBS News… A-          1426 lv
##  8 U.S.       2016-11-03 2016  11    05    NBC News… A-          1282 lv
##  9 New Mexico 2016-11-06 2016  11    06    Zia Poll  <NA>        8439 lv
```

tidyr::unite()

Combines multiple columns into a single column.

Example: Using the previous example, unite the "year", "month", "day" back into "enddate"

```
separated_data %>%
  unite("enddate", c("year", "month", "day"), sep = "-")
```

```
## # A tibble: 4,208 × 15
##    state startdate  enddate pollster grade samplesize population rawpoll_clinton
##    <fct> <date>     <chr>   <fct>    <fct>       <int> <chr>                <dbl>
##  1 U.S.  2016-11-03 2016-1… ABC New… A+           2220 lv                    47
##  2 U.S.  2016-11-01 2016-1… Google … B           26574 lv                    38.0
##  3 U.S.  2016-11-02 2016-1… Ipsos    A-           2195 lv                    42
##  4 U.S.  2016-11-04 2016-1… YouGov   B            3677 lv                    45
##  5 U.S.  2016-11-03 2016-1… Gravis … B-          16639 rv                    47
##  6 U.S.  2016-11-03 2016-1… Fox New… A            1295 lv                    48
##  7 U.S.  2016-11-02 2016-1… CBS New… A-           1426 lv                    45
##  8 U.S.  2016-11-03 2016-1… NBC New… A-           1282 lv                    44
##  9 New…  2016-11-06 2016-1… Zip Poll <NA>         8439 lv                    46
```

Main `dplyr` Verbs

- `dplyr` helps you with the main data manipulation challenges.

1. `select()`: Choose certain variables by name

Main `dplyr` Verbs

- `dplyr` helps you with the main data manipulation challenges.

1. `select()`: Choose certain variables by name

2. `filter()`: Subset observations based on a certain condition.

Main `dplyr` Verbs

- `dplyr` helps you with the main data manipulation challenges.

1. `select()`: Choose certain variables by name
2. `filter()`: Subset observations based on a certain condition.
3. `arrange()`: Reorder rows based on a certain condition.

Main `dplyr` Verbs

- `dplyr` helps you with the main data manipulation challenges.

1. `select()`: Choose certain variables by name
2. `filter()`: Subset observations based on a certain condition.
3. `arrange()`: Reorder rows based on a certain condition.
4. `mutate()`: Create new variables.

Main `dplyr` Verbs

- `dplyr` helps you with the main data manipulation challenges.

1. `select()`: Choose certain variables by name
2. `filter()`: Subset observations based on a certain condition.
3. `arrange()`: Reorder rows based on a certain condition.
4. `mutate()`: Create new variables.
5. `summarise()`: Collapse data to a single summary

Main `dplyr` Verbs

- `dplyr` helps you with the main data manipulation challenges.

1. `select()`: Choose certain variables by name
2. `filter()`: Subset observations based on a certain condition.
3. `arrange()`: Reorder rows based on a certain condition.
4. `mutate()`: Create new variables.
5. `summarise()`: Collapse data to a single summary
6. `group_by()`: All the above can be used in conjunction with `group_by()` to use function on groups rather than entire data

dplyr::select()

Example: Only keep the variables `state, startdate, enddate, pollster, rawpoll_clinton, rawpoll_trump`

```
polls_us_election_2016 %>%
  select(state,startdate,enddate,pollster,rawpoll_clinton,rawpoll_trump)
```

```
## # A tibble: 4,208 × 6
##    state      startdate  enddate    pollster       rawpoll_clinton rawpoll_trump
##    <fct>      <date>     <date>     <fct>                    <dbl>         <dbl>
##  1 U.S.       2016-11-03 2016-11-06 ABC News/Wash…            47            43
##  2 U.S.       2016-11-01 2016-11-07 Google Consum…            38.0          35.7
##  3 U.S.       2016-11-02 2016-11-06 Ipsos                     42            39
##  4 U.S.       2016-11-04 2016-11-07 YouGov                    45            41
##  5 U.S.       2016-11-03 2016-11-06 Gravis Market…            47            43
##  6 U.S.       2016-11-03 2016-11-06 Fox News/Ande…            48            44
##  7 U.S.       2016-11-02 2016-11-06 CBS News/New …            45            41
##  8 U.S.       2016-11-03 2016-11-05 NBC News/Wall…            44            40
##  9 New Mexico 2016-11-06 2016-11-06 Zia Poll                  46            44
## 10 U.S.       2016-11-04 2016-11-07 IBD/TIPP                  41.2          42.7
```

`dplyr::filter()`

Example: Which polls had a sample size of at least 2,000 people?

dplyr::filter()

Example: Which polls had a sample size of at least 2,000 people?

```
polls_us_election_2016
```

```
## # A tibble: 4,208 × 15
##    state      startdate  enddate    pollster         grade samplesize population
##    <fct>      <date>     <date>     <fct>            <fct>      <int> <chr>
##  1 U.S.       2016-11-03 2016-11-06 ABC News/Washin… A+          2220 lv
##  2 U.S.       2016-11-01 2016-11-07 Google Consumer… B          26574 lv
##  3 U.S.       2016-11-02 2016-11-06 Ipsos            A-          2195 lv
##  4 U.S.       2016-11-04 2016-11-07 YouGov           B           3677 lv
##  5 U.S.       2016-11-03 2016-11-06 Gravis Marketing B-         16639 rv
##  6 U.S.       2016-11-03 2016-11-06 Fox News/Anders… A           1295 lv
##  7 U.S.       2016-11-02 2016-11-06 CBS News/New Yo… A-          1426 lv
##  8 U.S.       2016-11-03 2016-11-05 NBC News/Wall S… A-          1282 lv
##  9 New Mexico 2016-11-06 2016-11-06 Zia Poll         <NA>        8439 lv
## 10 U.S.       2016-11-04 2016-11-07 IBD/TIPP         A-          1107 lv
## # ℹ 4,198 more rows
```

dplyr::filter()

Example: Which polls had a sample size of at least 2,000 people?

```
polls_us_election_2016 %>%
  filter(samplesize > 2000)
```

```
## # A tibble: 403 × 15
##    state      startdate  enddate    pollster          grade samplesize population
##    <fct>      <date>     <date>     <fct>             <fct>      <int> <chr>
##  1 U.S.       2016-11-03 2016-11-06 ABC News/Washin… A+          2220 lv
##  2 U.S.       2016-11-01 2016-11-07 Google Consumer… B          26574 lv
##  3 U.S.       2016-11-02 2016-11-06 Ipsos             A-          2195 lv
##  4 U.S.       2016-11-04 2016-11-07 YouGov            B           3677 lv
##  5 U.S.       2016-11-03 2016-11-06 Gravis Marketing B-         16639 rv
##  6 New Mexico 2016-11-06 2016-11-06 Zia Poll          <NA>        8439 lv
##  7 U.S.       2016-11-05 2016-11-07 The Times-Picay… <NA>        2521 lv
##  8 U.S.       2016-11-01 2016-11-07 USC Dornsife/LA… <NA>        2972 lv
##  9 Georgia    2016-11-03 2016-11-06 Gravis Marketing B-          2002 rv
## 10 Virginia   2016-11-01 2016-11-02 Remington         <NA>        3076 lv
```

`dplyr::filter()`

Standard suite of comparison operators:

- `>`: greater than,
- `<`: smaller than,
- `>=`: greater than or equal to,
- `<=`: smaller than or equal to,
- `!=`: not equal to,
- `==`: equal to.

Logical operators:

1. `x & y`: `x` **and** `y`
2. `x | y`: `x` **or** `y`
3. `!y`: **not** `y`

`dplyr::filter()`

Example: Which A graded poll with at least 2,000 people had Trump win at least 45% of the vote?

```
polls_us_election_2016 %>%
  filter(grade == "A" & samplesize > 2000 & rawpoll_trump > 45)
```

```
## # A tibble: 1 × 15
##   state   startdate  enddate    pollster       grade samplesize population
##   <fct>   <date>     <date>     <fct>          <fct>      <int> <chr>
## 1 Indiana 2016-04-26 2016-04-28 Marist College A           2149 rv
## # ℹ 8 more variables: rawpoll_clinton <dbl>, rawpoll_trump <dbl>,
## #   rawpoll_johnson <dbl>, rawpoll_mcmullin <dbl>, adjpoll_clinton <dbl>,
## #   adjpoll_trump <dbl>, adjpoll_johnson <dbl>, adjpoll_mcmullin <dbl>
```

dplyr::arrange()

Example: Sort the dataframe in terms of the sample size.

```
polls_us_election_2016 %>%
  arrange(samplesize) # this will do it from smaller to larger by default
```

```
## # A tibble: 4,208 × 15
##    state          startdate  enddate    pollster      grade samplesize population
##    <fct>          <date>     <date>     <fct>         <fct>      <int> <chr>
##  1 Wyoming        2016-10-04 2016-10-09 Google Consu… B             35 lv
##  2 Maine          2016-10-04 2016-10-09 Google Consu… B             37 lv
##  3 New Hampshire  2016-09-21 2016-09-26 Google Consu… B             39 lv
##  4 Hawaii         2016-09-14 2016-09-20 Google Consu… B             42 lv
##  5 Wyoming        2016-09-27 2016-10-03 Google Consu… B             43 lv
##  6 Rhode Island   2016-10-10 2016-10-14 Google Consu… B             45 lv
##  7 Vermont        2016-10-04 2016-10-09 Google Consu… B             47 lv
##  8 North Dakota   2016-09-27 2016-10-03 Google Consu… B             49 lv
##  9 Rhode Island   2016-09-14 2016-09-20 Google Consu… B             50 lv
## 10 Rhode Island   2016-10-04 2016-10-09 Google Consu… B             51 lv
```

dplyr::mutate()

Example: What was...

1. the combined vote share of Trump and Clinton for each poll?
2. the difference between Trump's raw poll vote share and 538's adjusted vote share?

```
polls_us_election_2016 %>%
  mutate(trump_clinton_tot = rawpoll_trump + rawpoll_clinton,
         trump_raw_adj_diff = rawpoll_trump - adjpoll_trump)
```

```
## # A tibble: 4,208 × 17
##    state      startdate  enddate    pollster         grade samplesize population
##    <fct>      <date>     <date>     <fct>            <fct>      <int> <chr>
##  1 U.S.       2016-11-03 2016-11-06 ABC News/Washin… A+          2220 lv
##  2 U.S.       2016-11-01 2016-11-07 Google Consumer… B          26574 lv
##  3 U.S.       2016-11-02 2016-11-06 Ipsos            A-          2195 lv
##  4 U.S.       2016-11-04 2016-11-07 YouGov           B           3677 lv
##  5 U.S.       2016-11-03 2016-11-06 Gravis Marketing B-         16639 rv
##  6 U.S.       2016-11-03 2016-11-06 Fox News/Anders… A           1295 lv
```

dplyr::mutate()

Example: What was...

1. the combined vote share of Trump and Clinton for each poll?
2. the difference between Trump's raw poll vote share and 538's adjusted vote share?

```
polls_us_election_2016 %>%
  mutate(trump_clinton_tot = rawpoll_trump + rawpoll_clinton,
         trump_raw_adj_diff = rawpoll_trump - adjpoll_trump) %>%
  names()
```

```
##  [1] "state"              "startdate"          "enddate"
##  [4] "pollster"           "grade"              "samplesize"
##  [7] "population"         "rawpoll_clinton"    "rawpoll_trump"
## [10] "rawpoll_johnson"    "rawpoll_mcmullin"   "adjpoll_clinton"
## [13] "adjpoll_trump"      "adjpoll_johnson"    "adjpoll_mcmullin"
## [16] "trump_clinton_tot"  "trump_raw_adj_diff"
```

dplyr::summarise()

Compute statistics

Example: What is the maximum vote share for Trump?

```
polls_us_election_2016 %>%
  summarise(max_trump = max(rawpoll_trump))
```

```
## # A tibble: 1 × 1
##   max_trump
##       <dbl>
## 1        68
```

dplyr::group_by()

Example: What is the average vote share for Clinton by poll grade?

```r
polls_us_election_2016 %>%
  group_by(grade)
```

```
## # A tibble: 4,208 × 15
## # Groups:   grade [11]
##    state      startdate  enddate    pollster         grade samplesize population
##    <fct>      <date>     <date>     <fct>            <fct>      <int> <chr>
##  1 U.S.       2016-11-03 2016-11-06 ABC News/Washin… A+          2220 lv
##  2 U.S.       2016-11-01 2016-11-07 Google Consumer… B          26574 lv
##  3 U.S.       2016-11-02 2016-11-06 Ipsos            A-          2195 lv
##  4 U.S.       2016-11-04 2016-11-07 YouGov           B           3677 lv
##  5 U.S.       2016-11-03 2016-11-06 Gravis Marketing B-         16639 rv
##  6 U.S.       2016-11-03 2016-11-06 Fox News/Anders… A           1295 lv
##  7 U.S.       2016-11-02 2016-11-06 CBS News/New Yo… A-          1426 lv
##  8 U.S.       2016-11-03 2016-11-05 NBC News/Wall S… A-          1282 lv
##  9 New Mexico 2016-11-06 2016-11-06 Zia Poll         <NA>        8439 lv
```

dplyr::group_by()

Example: What is the average vote share for Clinton by poll grade?

```
polls_us_election_2016 %>%
  group_by(grade)
```

```
## # A tibble: 4,208 × 15
## # Groups:   grade [11]
##    state      startdate  enddate    pollster         grade samplesize population
##    <fct>      <date>     <date>     <fct>            <fct>      <int> <chr>
##  1 U.S.       2016-11-03 2016-11-06 ABC News/Washin… A+          2220 lv
##  2 U.S.       2016-11-01 2016-11-07 Google Consumer… B          26574 lv
##  3 U.S.       2016-11-02 2016-11-06 Ipsos            A-          2195 lv
##  4 U.S.       2016-11-04 2016-11-07 YouGov           B           3677 lv
##  5 U.S.       2016-11-03 2016-11-06 Gravis Marketing B-         16639 rv
##  6 U.S.       2016-11-03 2016-11-06 Fox News/Anders… A           1295 lv
##  7 U.S.       2016-11-02 2016-11-06 CBS News/New Yo… A-          1426 lv
##  8 U.S.       2016-11-03 2016-11-05 NBC News/Wall S… A-          1282 lv
##  9 New Mexico 2016-11-06 2016-11-06 Zia Poll         <NA>        8439 lv
```

dplyr::group_by()

Example: What is the average vote share for Clinton by poll grade?

```
polls_us_election_2016 %>%
  group_by(grade) %>%
  summarise(mean_vote_clinton = mean(rawpoll_clinton))
```

```
## # A tibble: 11 × 2
##    grade mean_vote_clinton
##    <fct>             <dbl>
##  1 D                  46.7
##  2 C-                 43.2
##  3 C                  41.8
##  4 C+                 44.2
##  5 B-                 43.9
##  6 B                  37.3
##  7 B+                 44.1
##  8 A-                 43.0
##  9 A                  45.3
```

Chaining commands

With these verbs and the pipe chaining commands helps tidy your workflow, you can do multiple of this operations at the same time. (for tidyverse verbs and others)

```r
polls_us_election_2016 %>%
  pivot_longer(
    cols = c(rawpoll_clinton:adjpoll_mcmullin),
    names_to = c(".value", "candidate"),
    names_pattern = "(rawpoll|adjpoll)_(.*)"
  ) %>% # reshape
  select(-enddate) %>%  # everything except enddate
  filter(samplesize>=2000) %>%  # filter sample size greater than 2,000
  group_by(state, candidate) %>% #group by state and candidate
  summarise(rawpoll = mean(rawpoll, na.rm = TRUE)) # summarise
```

```
## # A tibble: 104 × 3
## # Groups:   state [26]
##    state      candidate rawpoll
##    <fct>      <chr>       <dbl>
```

Chaining commands

With these verbs and the pipe chaining commands helps tidy your workflow, you can do multiple of this operations at the same time. (for tidyverse verbs and others)

```
polls_us_election_2016 %>%
  pivot_longer(
    cols = c(rawpoll_clinton:adjpoll_mcmullin),
    names_to = c(".value", "candidate"),
    names_pattern = "(rawpoll|adjpoll)_(.*)"
  )  # reshape
```

```
## # A tibble: 16,832 × 10
##    state startdate  enddate    pollster     grade samplesize population candidate
##    <fct> <date>     <date>     <fct>        <fct>      <int> <chr>      <chr>
##  1 U.S.  2016-11-03 2016-11-06 ABC News/W… A+          2220 lv         clinton
##  2 U.S.  2016-11-03 2016-11-06 ABC News/W… A+          2220 lv         trump
##  3 U.S.  2016-11-03 2016-11-06 ABC News/W… A+          2220 lv         johnson
##  4 U.S.  2016-11-03 2016-11-06 ABC News/W… A+          2220 lv         mcmullin
##  5 U.S.  2016-11-01 2016-11-07 Google Con… B          26574 lv         clinton
```

Chaining commands

With these verbs and the pipe chaining commands helps tidy your workflow, you can do multiple of this operations at the same time. (for tidyverse verbs and others)

```
polls_us_election_2016 %>%
  pivot_longer(
    cols = c(rawpoll_clinton:adjpoll_mcmullin),
    names_to = c(".value", "candidate"),
    names_pattern = "(rawpoll|adjpoll)_(.*)"
  ) %>% # reshape
  select(-enddate) # everything except enddate
```

```
## # A tibble: 16,832 × 9
##    state startdate  pollster       grade samplesize population candidate rawpoll
##    <fct> <date>     <fct>          <fct>      <int> <chr>      <chr>       <dbl>
##  1 U.S.  2016-11-03 ABC News/Wash… A+          2220 lv         clinton        47
##  2 U.S.  2016-11-03 ABC News/Wash… A+          2220 lv         trump          43
##  3 U.S.  2016-11-03 ABC News/Wash… A+          2220 lv         johnson         4
##  4 U.S.  2016-11-03 ABC News/Wash… A+          2220 lv         mcmullin       NA
```

Chaining commands

With these verbs and the pipe chaining commands helps tidy your workflow, you can do multiple of this operations at the same time. (for tidyverse verbs and others)

```
polls_us_election_2016 %>%
  pivot_longer(
    cols = c(rawpoll_clinton:adjpoll_mcmullin),
    names_to = c(".value", "candidate"),
    names_pattern = "(rawpoll|adjpoll)_(.*)"
  ) %>% # reshape
  select(-enddate) %>%  # everything except enddate
  filter(samplesize>=2000) # filter sample size greater than 2,000
```

```
## # A tibble: 1,612 × 9
##    state startdate  pollster       grade samplesize population candidate rawpoll
##    <fct> <date>     <fct>          <fct>      <int> <chr>      <chr>       <dbl>
##  1 U.S.  2016-11-03 ABC News/Wash… A+          2220 lv         clinton        47
##  2 U.S.  2016-11-03 ABC News/Wash… A+          2220 lv         trump          43
##  3 U.S.  2016-11-03 ABC News/Wash… A+          2220 lv         johnson         4
```

Chaining commands

With these verbs and the pipe chaining commands helps tidy your workflow, you can do multiple of this operations at the same time. (for tidyverse verbs and others)

```
polls_us_election_2016 %>%
  pivot_longer(
    cols = c(rawpoll_clinton:adjpoll_mcmullin),
    names_to = c(".value", "candidate"),
    names_pattern = "(rawpoll|adjpoll)_(.*)"
  ) %>% # reshape
  select(-enddate) %>%  # everything except enddate
  filter(samplesize>=2000) %>%   # filter sample size greater than 2,000
  group_by(state, candidate) %>% #group by state and candidate
  summarise(rawpoll = mean(rawpoll, na.rm = TRUE)) # summarise
```

```
## # A tibble: 104 × 3
## # Groups:   state [26]
##    state      candidate rawpoll
##    <fct>      <chr>       <dbl>
```

Data Visualization

Data Visualization

- There is an extremely powerful tool in the `tidyverse`: `ggplot2`

gg is for Grammar of Graphics

How do we express visual in words?

- **Data** to be visualized
- **Aes**thetics mappings from data to visual component
- **Geom**etric objects that appear on the plot

(and more but we will keep it simple) for more you can see this presentation

Data

Tidy Data

1. Each variable forms a ***column***

2. Each observation forms a ***row***

3. Each observational unit forms a table

Start by asking

1. What information do I want to use in my visualization?

2. Is that data contained in ***one column/row*** for a given data point?

Data

We will use our same dataframe to show this.

```
longer_dataframe %>%
ggplot()
```

or

```
ggplot(longer_dataframe)
```

Aesthetics

Map data to visual elements or parameters

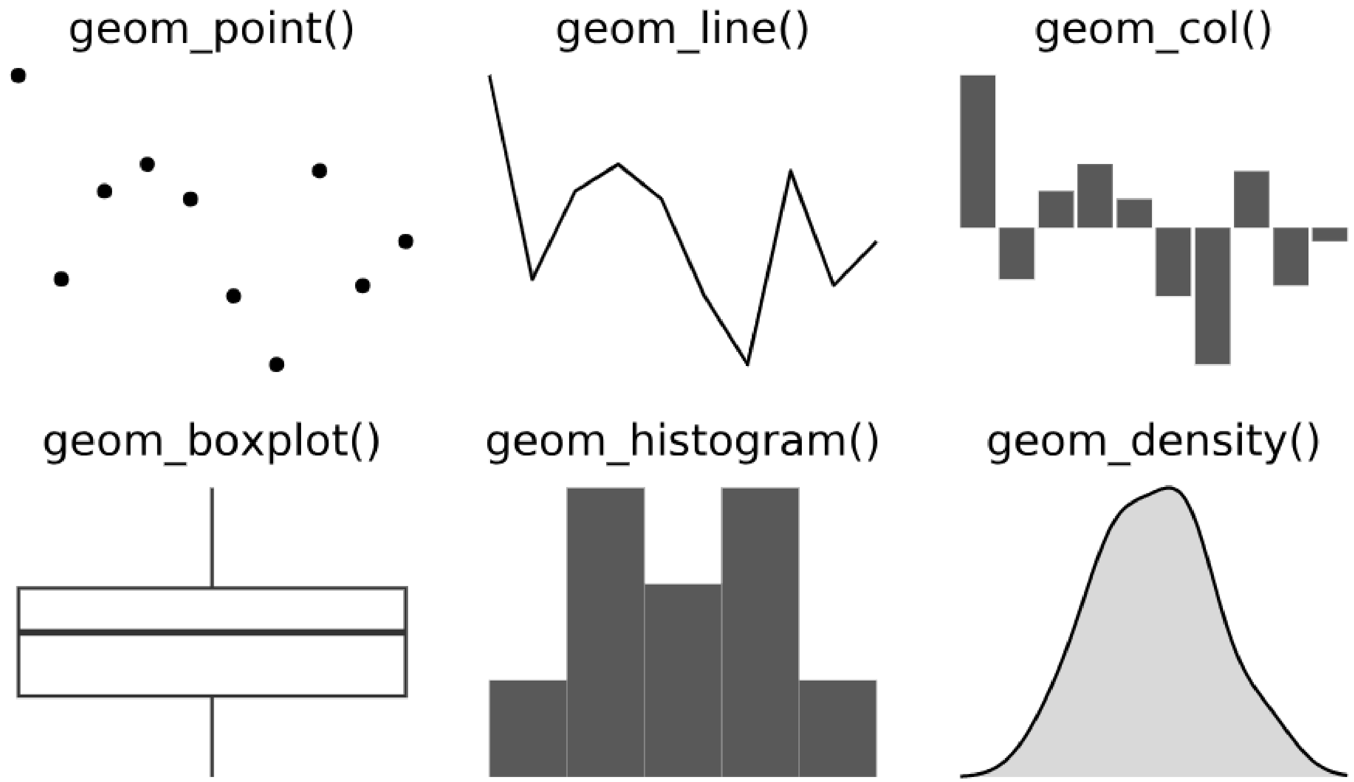
- date → **x**
- vote → **y**
- candidate → *shape*, *color*, etc.

Aesthetics

```r
longer_dataframe %>%
  filter(state == "District of Columbia") %>%
  ggplot() +
  aes(x = enddate,
      y = rawpoll,
      color = candidate)
```

Geoms

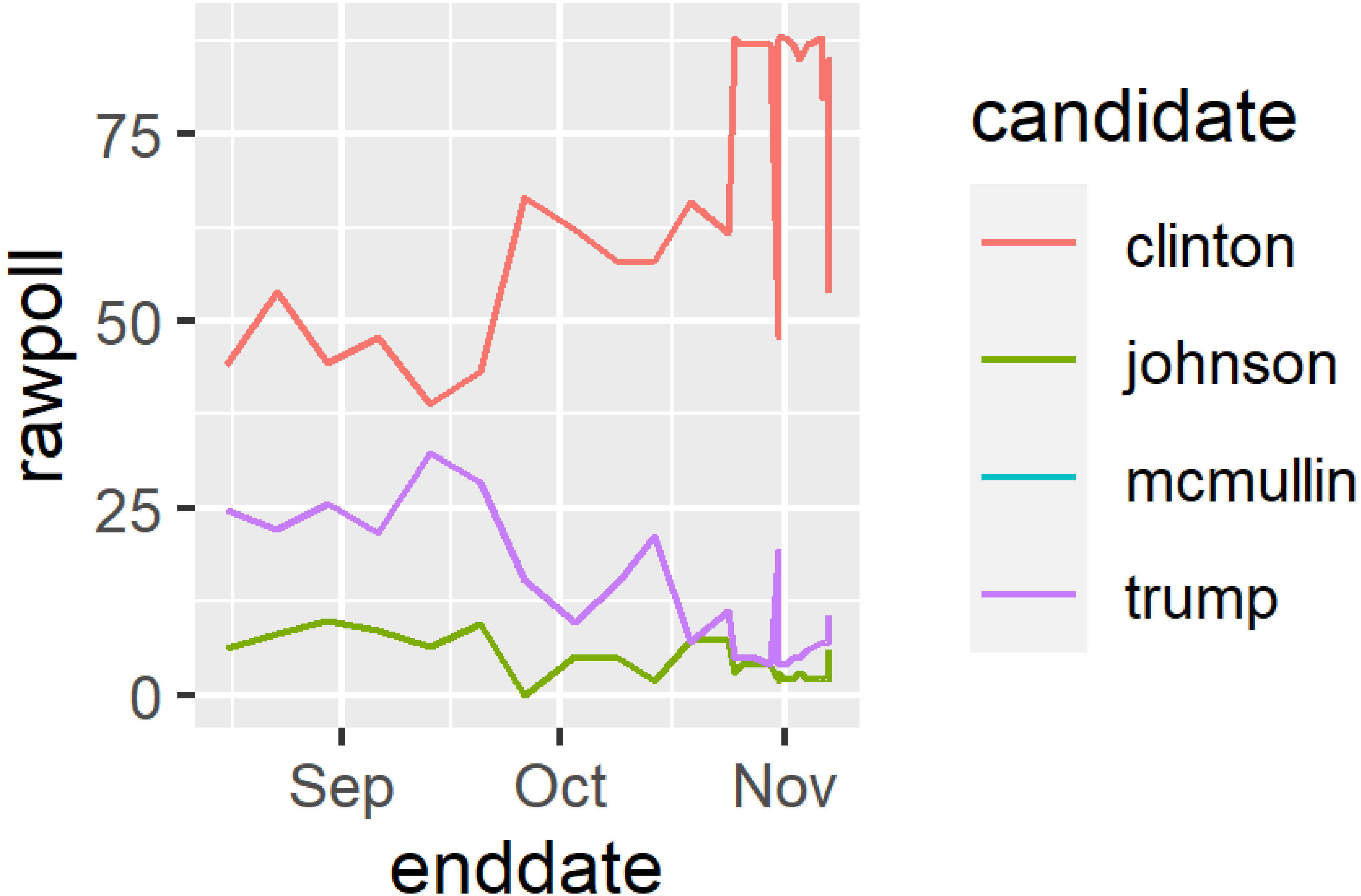
Geometric objects displayed on the plot



Geoms

```r
longer_dataframe %>%
  filter(state == "District of Columbia") %>%
  ggplot() +
  aes(x = enddate,
      y = rawpoll,
      color = candidate) +
  geom_line()
```

Geoms



rawpoll
75
50
25
0
Sep
Oct
Nov
enddate
candidate
clinton
johnson
mcmullin
trump

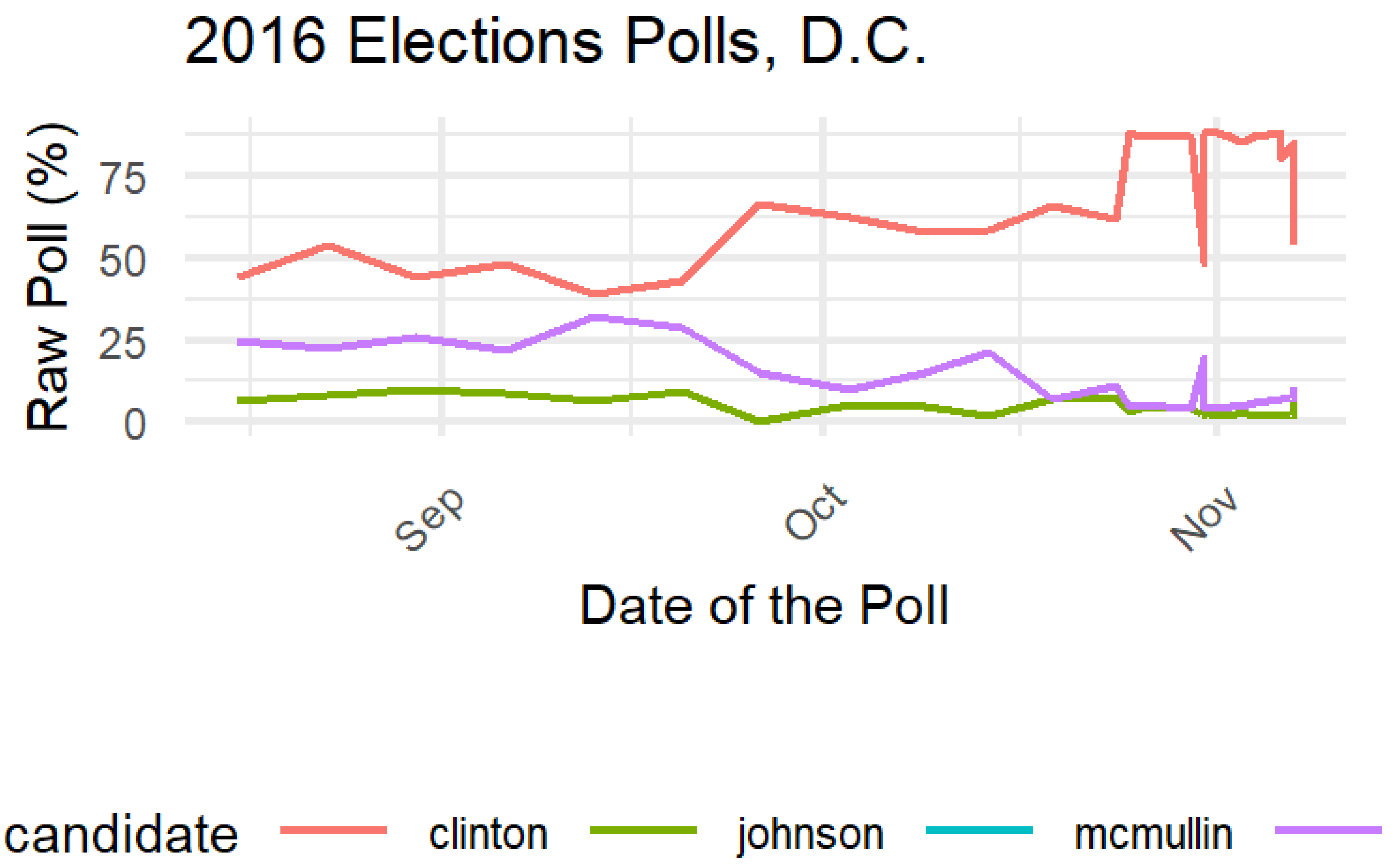
Extra

We can make it prettier, and do many more things with ggplot, please explore on your own.

```r
longer_dataframe %>%
  filter(state == "District of Columbia") %>%
  ggplot() +
  aes(x = enddate,
      y = rawpoll,
      color = candidate) +
  geom_line() +
  theme_minimal() + # minimal theme (delete extra lines)
  theme(
  legend.position = "bottom" # position of legend
  ) +
  labs( # Define title and labels
    y = "Raw Poll",
    x = "Date of the Poll",
    title = "2016 Elections Polls in District of Columbia"
  )
```

Extra

We can make it prettier, and do many more things with ggplot, please explore on your own.



Conclusion

Do the best you can until you know better, when you know better do better...

Thank you!

References

I largely based this slides on the presentation by Florian Oswald and Mylene Feuillade that you can find here and here

More resources

- You are ***highly encouraged*** to read through Hadley Wickham's chapter. It's clear and concise.
- Also check out this great dplyr "cheatsheet" here.
- As well as this tidyr "cheatsheet" here.
- Garrick Aden-Buie ggplot presentation here.
- The R Graph Gallery here.
- Illustrations by Allison Horst.