

# **WEEK 1**

# **INTRO TO TIDYVERSE AND GGPLOT**

**DATA VISUALIZATION FOR SOCIAL SCIENTISTS**

**LECTURER: JEFFREY ZIEGLER, PhD**

**TEACHING FELLOW: SHEKHAR KEDIA**

**ASDS - TRINITY COLLEGE DUBLIN**

**SPRING 2026**

# ROAD MAP FOR TODAY

- Today:

- ▶ Welcome
- ▶ Mapping data to aesthetics
- ▶ Intro for tidyverse and ggplot in R

- By next week, please...

- ▶ Fork GitHub repository
- ▶ Problem set #1

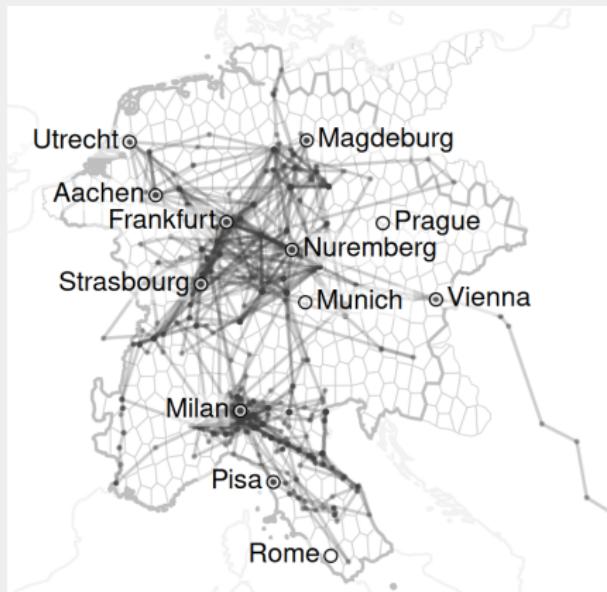
# MAPPING DATA TO AESTHETICS

- Aesthetic
  - ▶ Visual property of graph
  - ▶ Position, shape, color, etc.

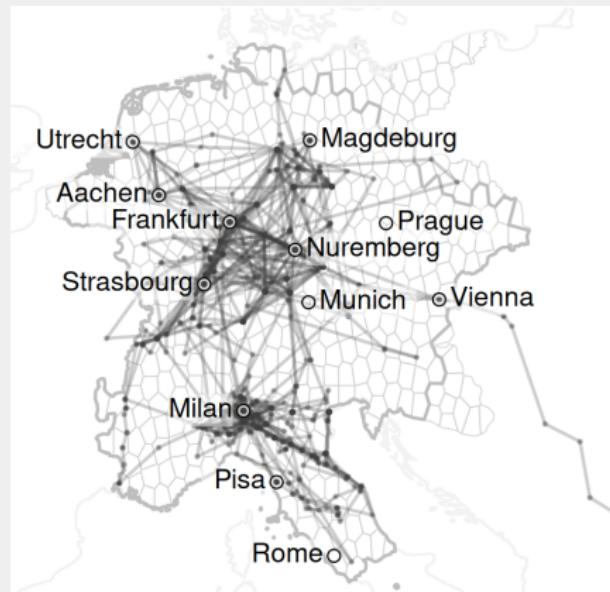
- Data
  - ▶ Column in dataset

# Ex: RULERS' ITINERARIES OF ROMAN EMPIRE

Frederick I Barbarossa (1152-1190)



Louis IV (1314-1347)



Müller-Crepon, C., Neupert-Wentz, C., Kokkonen, A., & Möller, J. (2025). Rulers on the Road: Itinerant Rule in the Holy Roman Empire, AD 919–1519. American Journal of Political Science.

## Ex: MAPPING DATA TO AESTHETICS

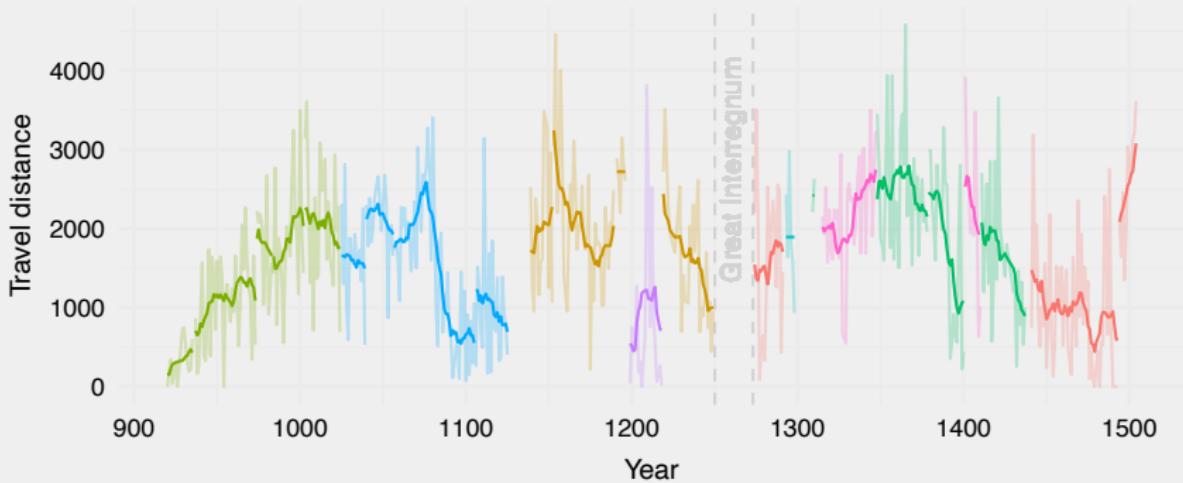
| Data        | Aesthetic         | Graphic/Geometry |
|-------------|-------------------|------------------|
| Longitude   | Position (x-axis) | Grid             |
| Latitude    | Position (y-axis) | Grid             |
| Itinerary   | Color             | Path             |
| Destination | Color             | Point            |

In ggplot...

| Data        | aes() | geom         |
|-------------|-------|--------------|
| Longitude   | x     | aes(x=)      |
| Latitude    | y     | aes(y=)      |
| Itinerary   | x, y  | geom_line()  |
| Destination | x, y  | geom_point() |

## Ex: MAPPING DATA TO AESTHETICS

Count of *Regesta Imperii* entries by ruler and year (running mean in bold)



Müller-Crepon, C., Neupert-Wentz, C., Kokkonen, A., & Møller, J. (2025). Rulers on the Road: Itinerant Rule in the Holy Roman Empire, AD 919–1519. American Journal of Political Science.

## Ex: MAPPING DATA TO AESTHETICS

| Data     | aes() | geom        |
|----------|-------|-------------|
| Time     | x     | aes(x=)     |
| Distance | y     | aes(y=)     |
| Leader   | x, y  | geom_line() |
| Mean     | x, y  | geom_line() |

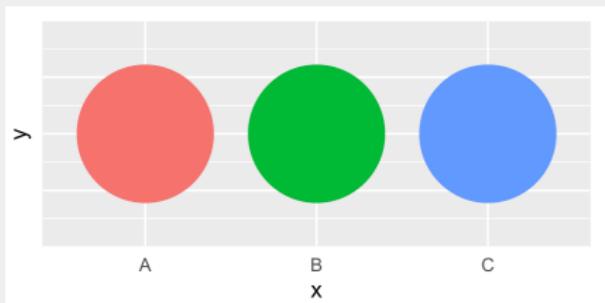
## COMPONENTS AS LAYERS

- So far we've introduced data, aesthetics, and geometries
- Think of these components as "layers"
- There are many layers we can use
  - ▶ We sequentially add layers onto foundational `ggplot()` plot to create complex figures
  - ▶ Add them to initial plot in `ggplot()` with "+"

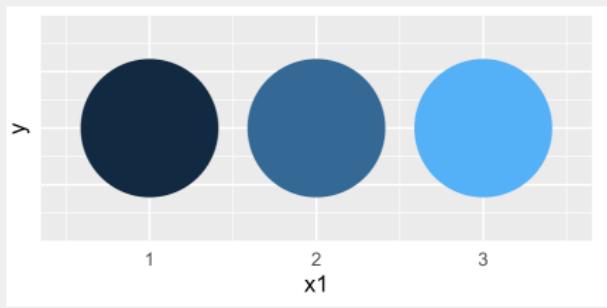
# POSSIBLE AESTHETICS

Color - aes(color=)

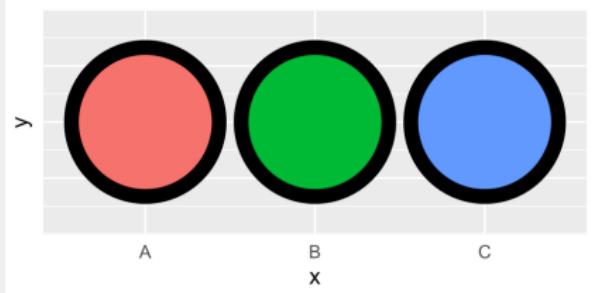
Continuous



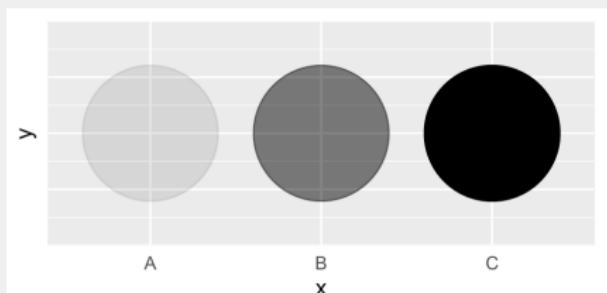
Discrete



Fill - aes(fill=)

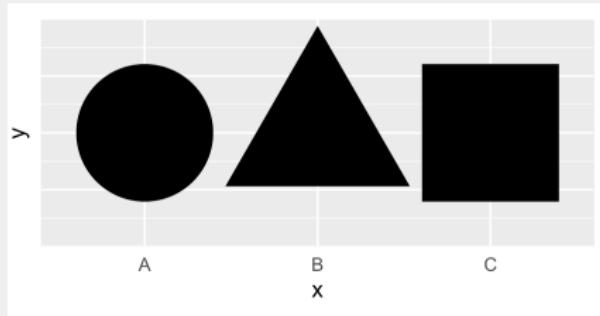


Alpha - aes(alpha=)

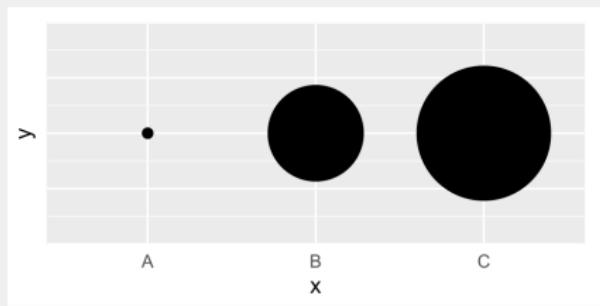


# POSSIBLE AESTHETICS

Shape - geom\_point()



Size - geom\_point(size=)



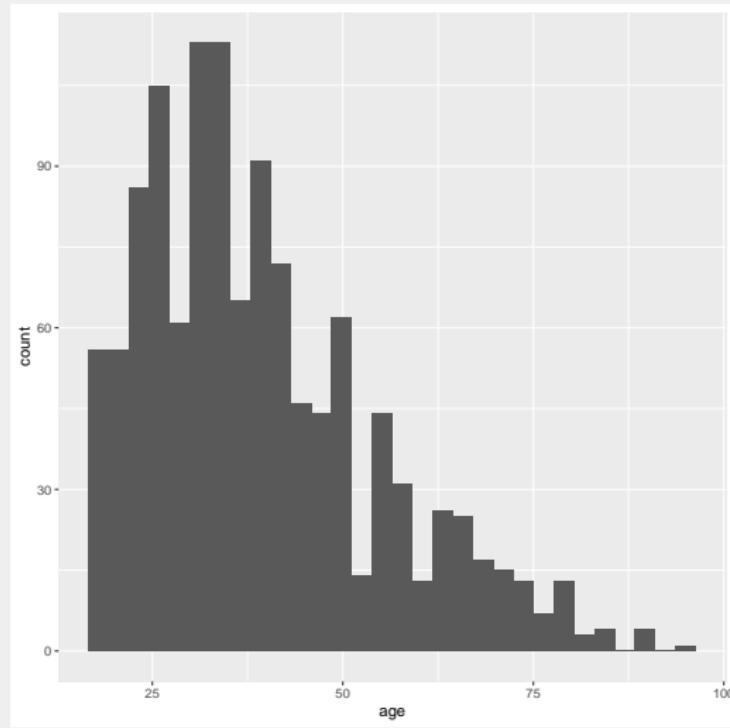
# POSSIBLE GEOMS

| Example geom   | What it makes                        |
|--|--------------------------------------|
|  | <code>geom_col()</code> Bar charts   |
|  | <code>geom_text()</code> Text        |
|  | <code>geom_point()</code> Points     |
|  | <code>geom_boxplot()</code> Boxplots |
|  | <code>geom_sf()</code> Maps          |

- Dozens of possible geoms, each class session will cover different ones)
- See [ggplot2 documentation](#) for complete examples of all geom layers

# Ex: AFROBAROMETER - ZIMBABWE (R10, 2024)

## Distribution of respondents by age



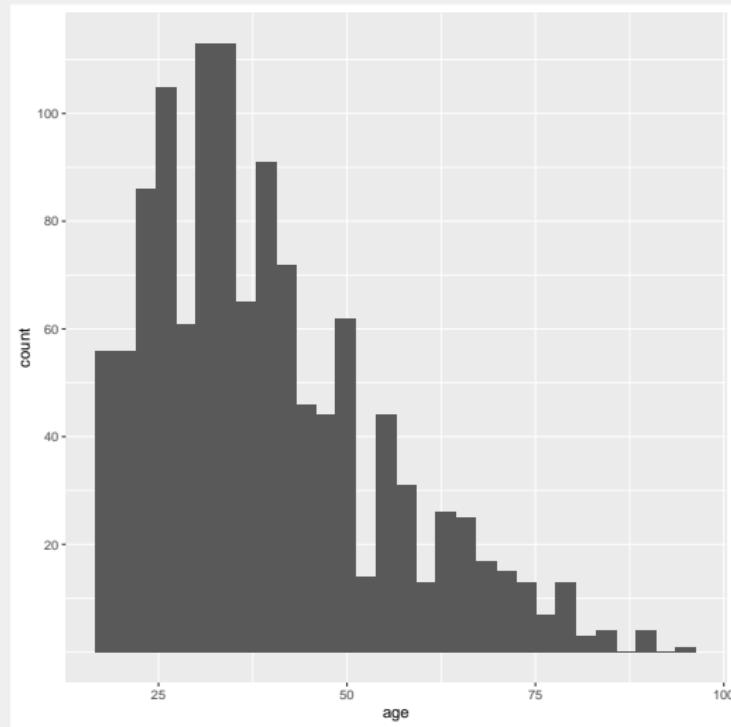
```
1 ggplot(data = AB_ZIM ,  
         mapping = aes(x=  
                         age)) +  
2 geom_histogram()
```

# SCALES

Scales change properties of variable mapping

| Example layer                                 | What it does                      |
|---|-----------------------------------|
| <code>scale_x_continuous()</code>             | Make the x-axis continuous        |
| <code>scale_x_continuous(breaks = 1:5)</code> | Manually specify axis ticks       |
| <code>scale_x_log10()</code>                  | Log x-axis                        |
| <code>scale_color_gradient()</code>           | Use gradient                      |
| <code>scale_fill_viridis_d()</code>           | Fill with discrete viridis colors |

## EX: ADJUST Y-AXIS SCALE



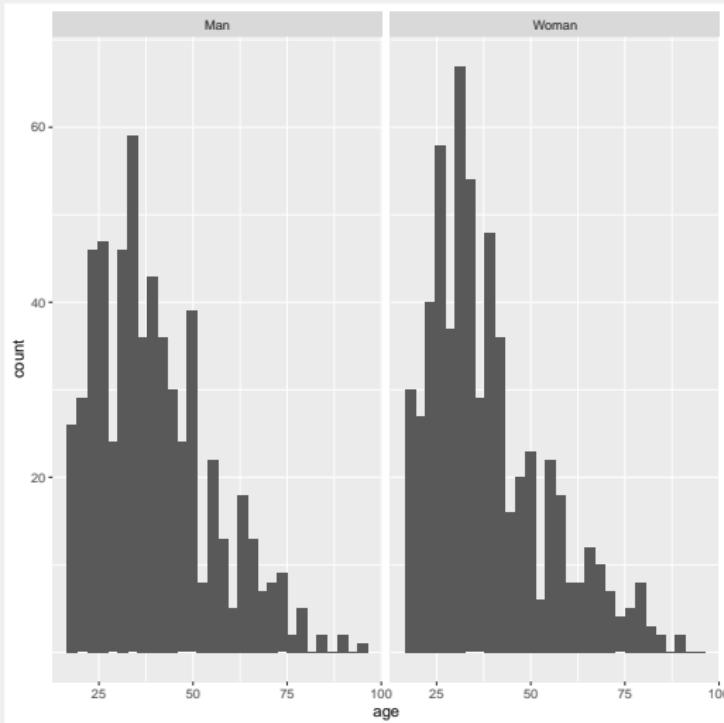
```
1 ggplot(data = AB_ZIM,  
         mapping = aes(x=  
                         age)) +  
2   geom_histogram() +  
3   scale_y_continuous(  
                           breaks = c(20,  
                                     40, 60, 80, 100))
```

# FACETS

Facets show subplots for different subsets of data

| Example layer                                  | What it does                  |
|--|-------------------------------|
| <code>facet_wrap(vars(continent))</code>       | Plot for each continent       |
| <code>facet_wrap(vars(continent, year))</code> | iPlot for each continent/year |
| <code>acet_wrap(..., ncol = 1)</code>          | Put all facets in one column  |
| <code>facet_wrap(..., nrow = 1)</code>         | Put all facets in one row     |

## EX: ADD FACETS FOR GENDER



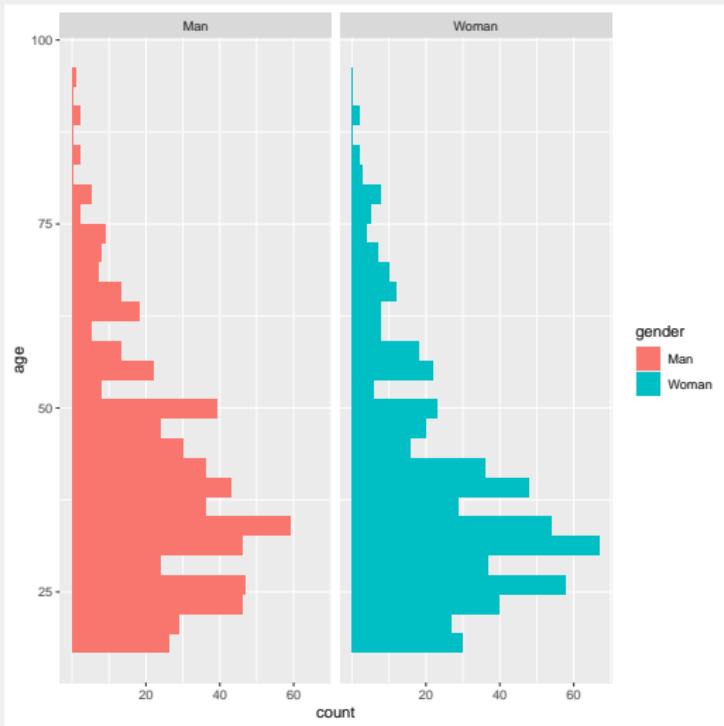
```
1 ggplot(data = AB_ZIM ,  
         mapping = aes(x=  
                        age)) +  
2   geom_histogram() +  
3   scale_y_continuous(  
                           breaks = c(20,  
                                 40, 60, 80, 100))  
+  
4   facet_wrap(vars(  
                      gender))
```

# COORDINATES

Change coordinate system

| Example layer                                 | What it does              |
|---|---------------------------|
| <code>coord_cartesian()</code>                | Plot for each continent   |
| <code>coord_cartesian(ylim = c(1, 10))</code> | Zoom in where y is 1–10   |
| <code>coord_flip()</code>                     | Switch x and y            |
| <code>coord_polar()</code>                    | Use circular polar system |

## Ex: FLIP COORDINATES



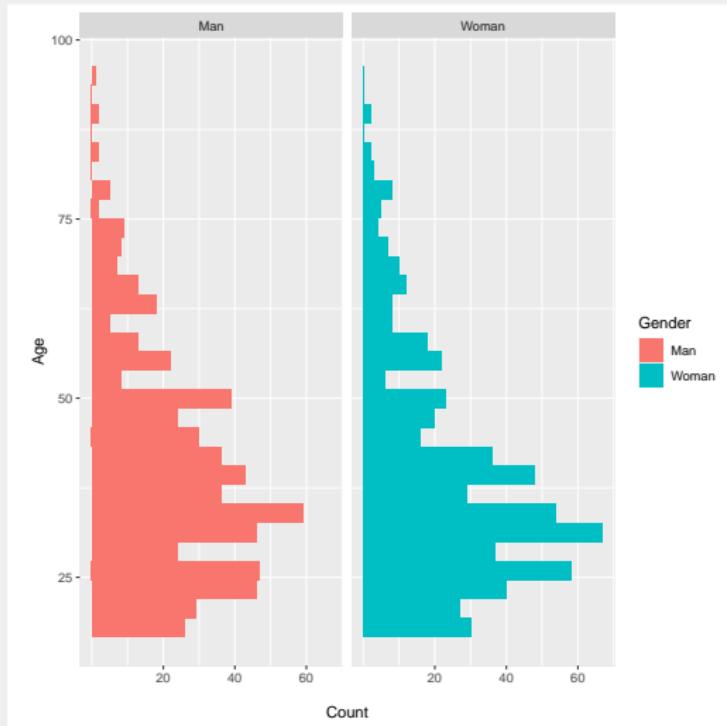
```
1 ggplot(data = AB_ZIM,  
         mapping = aes(x=  
                         age, fill=gender)  
         ) +  
2 geom_histogram() +  
3 scale_y_continuous(  
                     breaks = c(20,  
                               40, 60, 80, 100))  
+  
4 facet_wrap(vars(  
                  gender)) +  
5 coord_flip()
```

## LABELS

Add labels to plot with a single `labs()` layer

| Example layer                            | What it does         |
|--|----------------------|
| <code>labs(title = "Neat title")</code>  | Title                |
| <code>labs(caption = "Something")</code> | Caption              |
| <code>labs(y = "Something")</code>       | y-axis               |
| <code>labs(size = "Population")</code>   | Title of size legend |

## EX: ADD LABELS



```
1 ggplot(data = AB_ZIM,
         mapping = aes(x=
                       age, fill=gender)
         ) +
2 geom_histogram() +
3 scale_y_continuous(
        breaks = c(20,
                  40, 60, 80, 100))
+
4 facet_wrap(vars(
                gender)) +
5 coord_flip() +
6 labs(x="\nAge", y="\nCount", fill="Gender")
```

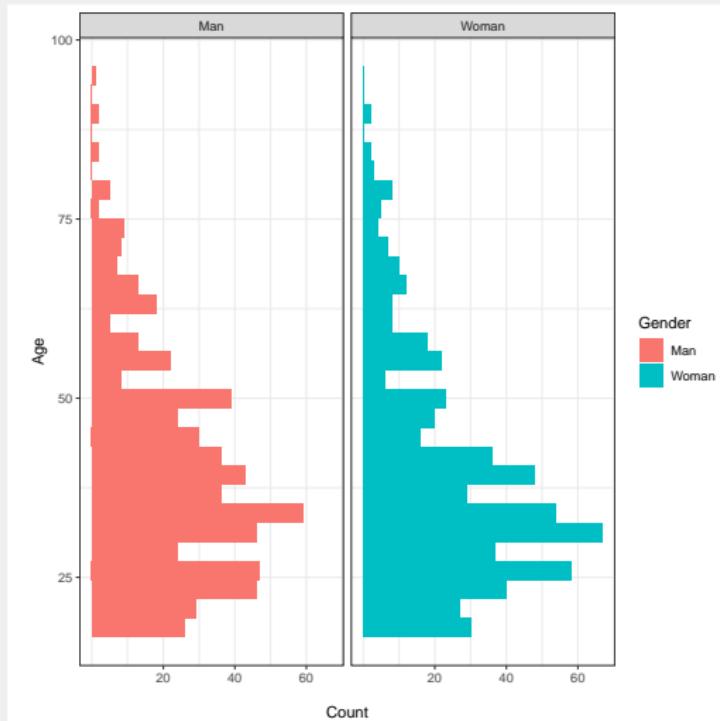
# THEME

Change appearance of anything in plot

- Many built-in themes
- Pre-built themes online, like `ggthemes` package
- Make theme adjustments with `theme()`
- So many options, we have a whole class session dedicated to this

| Example layer                | What it does            |
|------------------------------|-------------------------|
| <code>theme_grey()</code>    | Default grey background |
| <code>ltheme_bw()</code>     | Black and white         |
| <code>theme_dark()</code>    | Dark                    |
| <code>theme_minimal()</code> | Minimal                 |

## Ex: APPLY theme\_bw



```
1 ggplot(data = AB_ZIM,  
         mapping = aes(x=  
                        age, fill=gender)  
         ) +  
2 geom_histogram() +  
3 scale_y_continuous(  
         breaks = c(20,  
                    40, 60, 80, 100))  
        +  
4 facet_wrap(vars(  
                      gender)) +  
5 coord_flip() +  
6 labs(x="\nAge", y="\nCount", fill="Gender") +  
7 theme_bw()
```

## CLEANING DATA FOR PLOTTING - TIDYVERSE

- For `ggplot()` to work, your data needs to be in a tidy format
- This doesn't mean that it's "clean", refers to structure of data
- All the packages in the **tidyverse** work best to tidy data

# VISUALIZING TIDY DATA

| country     | year | cases | population |
|-------------|------|-------|------------|
| Afghanistan | 1980 | 345   | 1607071    |
| Afghanistan | 2000 | 2666  | 2095360    |
| Brazil      | 1989 | 31737 | 17206362   |
| Brazil      | 2000 | 80488 | 17404898   |
| China       | 1989 | 21258 | 127215272  |
| China       | 2000 | 27666 | 12803583   |



variables

| country     | year | cases | population |
|-------------|------|-------|------------|
| Afghanistan | 1980 | 345   | 1607071    |
| Afghanistan | 2000 | 2666  | 2095360    |
| Brazil      | 1989 | 31737 | 17206362   |
| Brazil      | 2000 | 80488 | 17404898   |
| China       | 1989 | 21258 | 127215272  |
| China       | 2000 | 27666 | 12803583   |



observations

| country     | year | cases | population |
|-------------|------|-------|------------|
| Afghanistan | 1980 | 345   | 1607071    |
| Afghanistan | 2000 | 2666  | 2095360    |
| Brazil      | 1989 | 31737 | 17206362   |
| Brazil      | 2000 | 80488 | 17404898   |
| China       | 1989 | 21258 | 127215272  |
| China       | 2000 | 27666 | 12803583   |



values

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

# WIDE VS. LONG DATA

Ex: Employment rate by country

Wide

| Country  | 2017 | 2018 | 2019 | 2020 | 2021 |
|----------|------|------|------|------|------|
| Belgium  | 69,8 | 71   | 71,8 | 71,5 | 71,9 |
| Bulgaria | 71,4 | 72,4 | 75   | 73,4 | 73,2 |
| Czechia  | 78,4 | 79,8 | 80,2 | 79,6 | 79,8 |
| Denmark  | 77,8 | 78,7 | 79,4 | 78,8 | 79,8 |

Long

| Country  | Year | Value |
|----------|------|-------|
| Belgium  | 2017 | 69,8  |
| Belgium  | 2018 | 71    |
| Belgium  | 2019 | 71,8  |
| Belgium  | 2020 | 71,5  |
| Belgium  | 2021 | 71,9  |
| Bulgaria | 2017 | 71,4  |
| Bulgaria | 2018 | 72,4  |
| Bulgaria | 2019 | 75    |
| Bulgaria | 2020 | 73,4  |
| Bulgaria | 2021 | 73,2  |
| Czechia  | 2017 | 78,4  |
| Czechia  | 2018 | 79,8  |
| Czechia  | 2019 | 80,2  |
| Czechia  | 2020 | 79,6  |
| Czechia  | 2021 | 79,8  |

# REVIEW

- Mapping data to aesthetics
- ggplot
  - ▶ aes()
  - ▶ geom
  - ▶ Layers
  - ▶ Scales
  - ▶ Facets
  - ▶ Coordinates
  - ▶ Labels
  - ▶ Themes

## CLASS BUSINESS

- Read required (and suggested) online materials
- Fork GitHub repository
- Problem set # 1 is up on GitHub