Coding Lab: Visualizing data with `ggplot2`

Ari Anisfeld

Summer 2020

What will we cover?

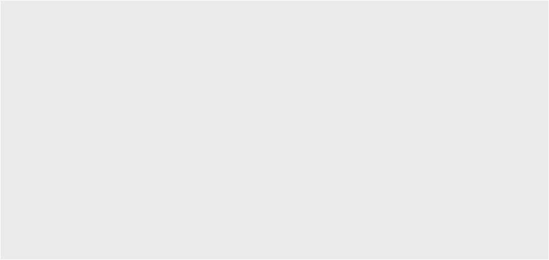
- What is the syntax of `ggplot`? What is `aes()`? What are `geom_xxx()`?
- How to use data visualization for exploration?
- How to make data visualization for communication?

Understanding ggplot()

By itself, ggplot() tells R to prepare to make a plot.

```
 texas_annual_sales <-
  texas_housing_data %>%
  group_by(year) %>%
  summarize(total_volume = sum(volume, na.rm = TRUE))

ggplot(data = texas_annual_sales)
```

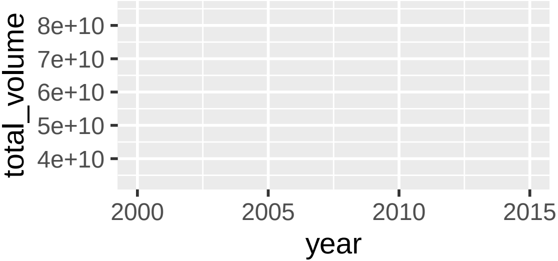


Adding a `mapping`

Adding `mapping = aes()` says how the data will map to "aesthetics".

- e.g. tell R to make x-axis `year` and y-axis `total_volume`.
- Each row of the data has (`year`, `total_volume`).
  - R will map that to the coordinate pair (x,y) .
  - Look at the data before moving on!

```r
ggplot(data = texas_annual_sales,
       mapping = aes(x = year, y = total_volume))
```



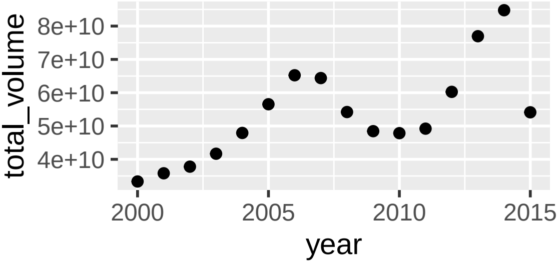
Visualizing the mapping with a geom

`geom_<name>` tells R what type of visualization to produce.

Here we see points.

- Each row of the data has (`year`, `total_volume`).
- R will map that to the coordinate pair (`x`,`y`).

```
ggplot(data = texas_annual_sales,
       mapping = aes(x = year, y = total_volume)) +
  geom_point()
```

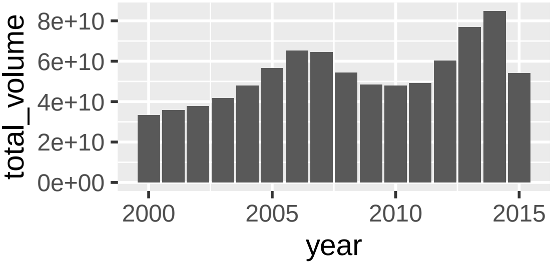


Visualizing the mapping with a geom

Here we see bars.

- Each row of the data has (year, total_volume).
- R will map that to the coordinate pair (x,y)

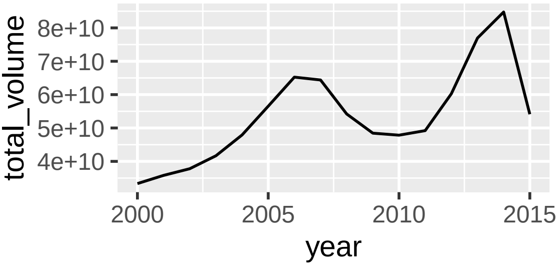
```
ggplot(data = texas_annual_sales,
       mapping = aes(x = year, y = total_volume)) +
  geom_col()
```



Visualizing the mapping with a geom

Here we see a line connecting each (x,y) pair.

```
ggplot(data = texas_annual_sales,
       mapping = aes(x = year, y = total_volume)) +
  geom_line()
```



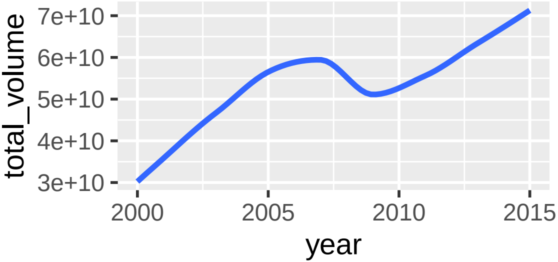
Visualizing the mapping with a geom

Here we see a smooth line. R does a statistical transformation!

- Now R doesn't visualize the mapping (`year`, `total_volume`) to each (x,y) pair
- Instead it fits a model to the (x,y) and then plots the "smooth" line

```r
ggplot(data = texas_annual_sales,
       mapping = aes(x = year, y = total_volume)) +
  geom_smooth(se = FALSE)
```

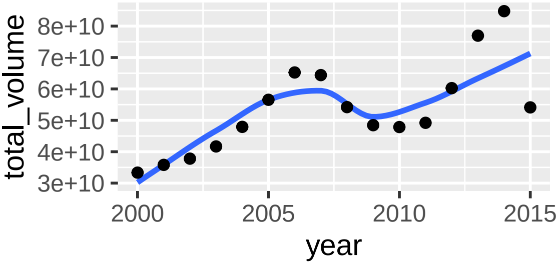
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Visualizing the mapping with a geom

We can overlay several geom.

```
ggplot(data = texas_annual_sales,
       mapping = aes(x = year, y = total_volume)) +
  geom_smooth(se = FALSE) +
  geom_point()
```

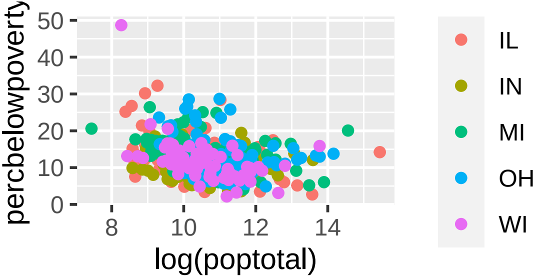


Using aesthetics to explore data.

Now we'll look at aesthetics that go beyond x and y axes so we can understand our data better.

- color maps data to the color of points or lines.
  - Each state is assigned a color.
  - This works with discrete data and continuous data.

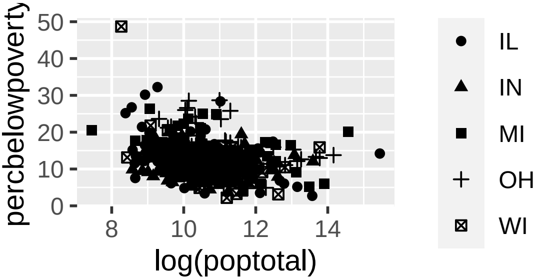
```
midwest %>%
  ggplot(aes(x = log(poptotal),
             y = percbelowpoverty,
             color = state)) +
    geom_point()
```



Using aesthetics to explore data.

- shape maps data to the shape of points.
  - Each state is assigned a shape.
  - This works with discrete data only.

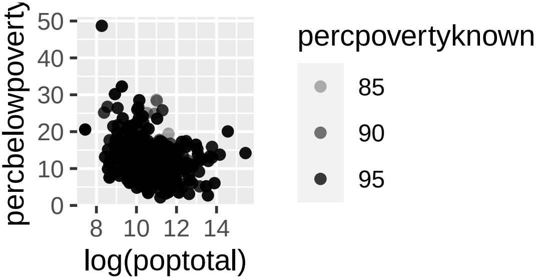
```
midwest %>%
   ggplot(aes(x = log(poptotal),
              y = percbelowpoverty,
              shape = state)) +
     geom_point()
```



Using aesthetics to explore data.

- alpha maps data to the transparency of points.
  - Here we map the percentage of people within a known poverty status to alpha[1]

```
midwest %>%
   ggplot(aes(x = log(poptotal),
              y = percbelowpoverty,
              alpha = percpovertyknown)) +
      geom_point()
```

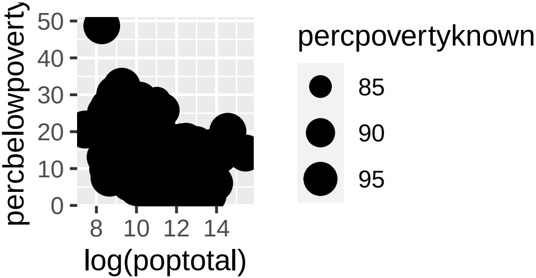


[1] Using alpha for a discrete variable is not advised.

Using aesthetics to explore data.

- size maps data to the size of points and width of lines.
    - Here we map the percentage of people within a known poverty status to size[2]

```
midwest %>%
  ggplot(aes(x = log(poptotal),
             y = percbelowpoverty,
             size = percpovertyknown)) +
    geom_point()
```



[2] Using size for a discrete variable is not advised.

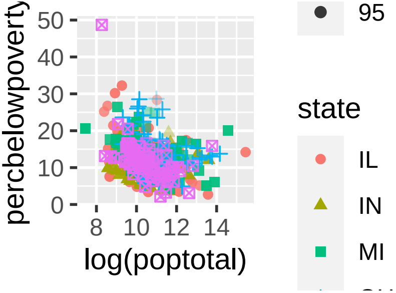
Using aesthetics to explore data.

We can combine any and all aesthetics, and even map the same variable to multiple aesthetics

```
midwest %>%
   ggplot(aes(x = log(poptotal),
              y = percbelowpoverty,
              alpha = percpovertyknown,
              color = state,
              shape = state))+
      geom_point()
```

Using aesthetics to explore data.

We can combine any and all aesthetics, and even map the same variable to multiple aesthetics



Using aesthetics to explore data

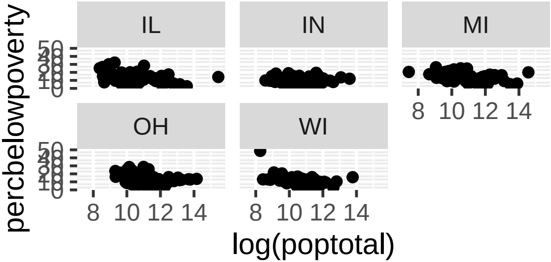
Different geoms have specific aesthetics that go with them.

- use `?` to see which aesthetics a geom accepts (e.g `?geom_point`)
  - the bold aesthetics are required.
- the ggplot cheatsheet shows all the `geoms` with their associated aesthetics

Facets

Facets provide an additional tool to explore multidimenional data

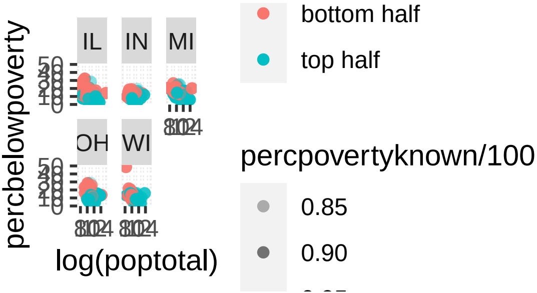
```
midwest %>%
   ggplot(aes(x = log(poptotal),
              y = percbelowpoverty)) +
      geom_point() +
      facet_wrap(vars(state))
```



Using aesthetics to explore data

Here's an example of how we can

```r
midwest %>%
  mutate(pc = ifelse(perchsd > median(perchsd), "top half",
  ggplot(aes(x = log(poptotal),
             y = percbelowpoverty,
             color = pc,
             alpha = percpovertyknown / 100)) +
  geom_point() +
  facet_wrap(~state)
```



Key take aways

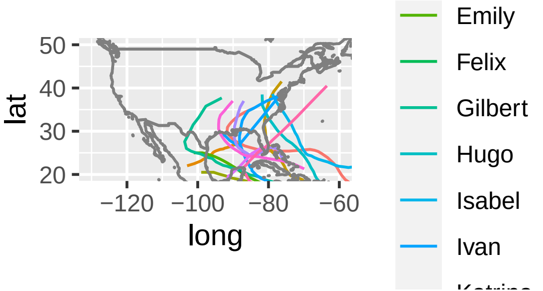
- ggplot starts by mapping data to "aesthetics".
  - e.g. What data shows up on x and y axes and how color, size and shape appear on the plot.
  - We need to be aware of 'continuous' vs. 'discrete' variables.
- Then, we use geoms to create a visualization based on the mapping.
  - Again we need to be aware of 'continuous' vs. 'discrete' variables.
- Making quick plots helps us understand data and makes aware of data issues
- To communicate effectively with data visualizations, we . . .

? What is aes()? What are geom_xxx()? - How to use data visualization for exploration? - How to make data visualization for communication?

```
big_cities <- c("Dallas", "Austin", "San Antonio", "Housto
texas_housing_cities <-
  texas_housing_data %>%
   # I'm taking out 2015, since it only has 7 months
```

Appendix: Some graphs you made along the way

```
storms %>%
  group_by(name, year) %>%
  filter(max(category) == 5) %>%
ggplot(aes(x = long, y = lat, color = name)) +
  geom_path() +
  borders("world") +
  coord_quickmap(xlim = c(-130, -60), ylim = c(20, 50))
```


lat
50
40
30
20
−120
−100
−80
−60
long
Emily
Felix
Gilbert
Hugo
Isabel
Ivan
Katrina