Week 6 – Further into Data Viz

Alex Lishinski and Joshua Rosenberg

February 25, 2021

Welcome!

Welcome to *week 6*!

Record the meeting

Breakout rooms!

Starting with whomever most wants to go first:

One question:

- What is a weird, unusual, or surprising situation that you encountered when using R in the last week?

One reflection/discussion:

- The Greenhalgh et al. (2020) chapter outlines six considerations related to conducting ethical research. Which of these six do you think is *important but insufficiently emphasized* in your area of research?

Review of last week's class

Why visualize data?

One answer:

"You should look at your data." (Healy, 2018)

To elaborate on this:

- Visualizations allow to *understand the structure and nature of your data*, and to begin to understand what might relate to what else
- Just like we want to be constantly looking at our data in its spreadsheet/table/data frame format (e.g., `str()`, `glimpse()`, and `View()`), visualizing our data can help us to make sure our data contains what we think it does–and it can alert us to when it does not

Review of last week's class

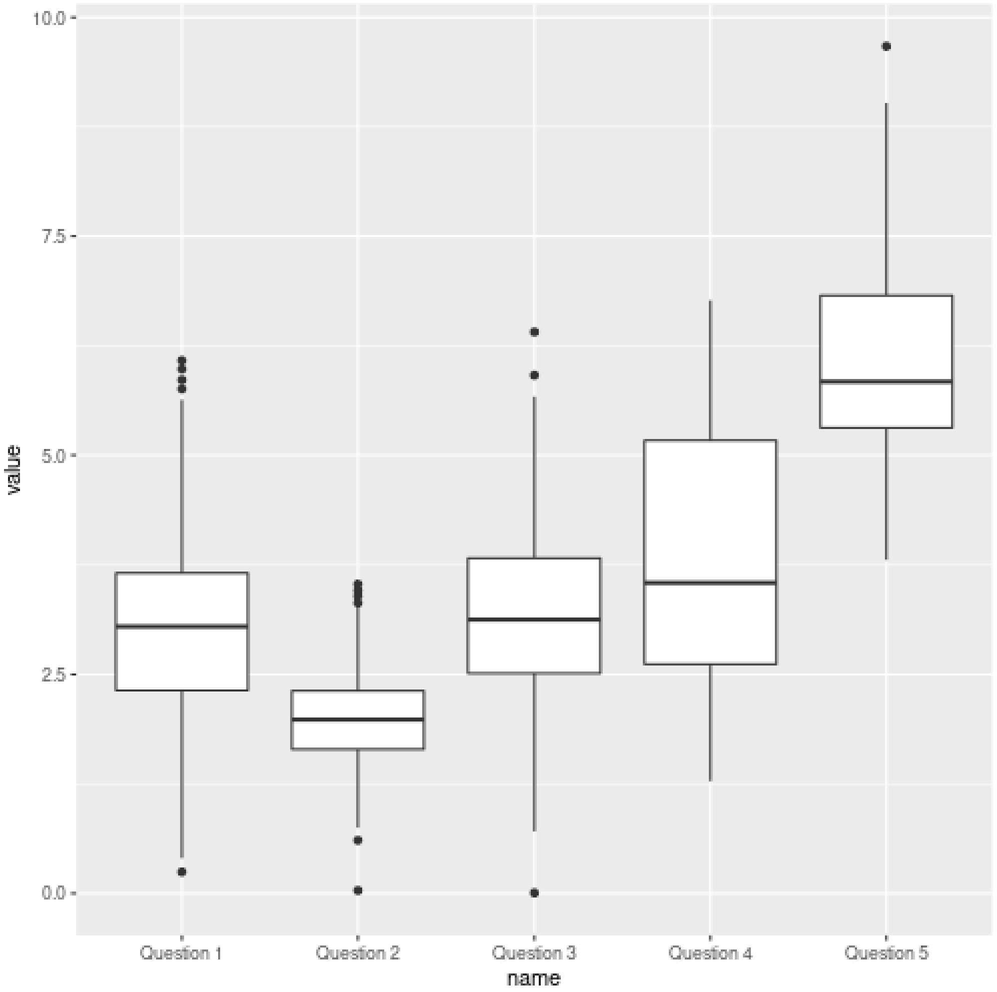
- Exploratory visualization and presentation visualization
- Basics of using base R plotting functions as well as ggplot

Review of last week's class

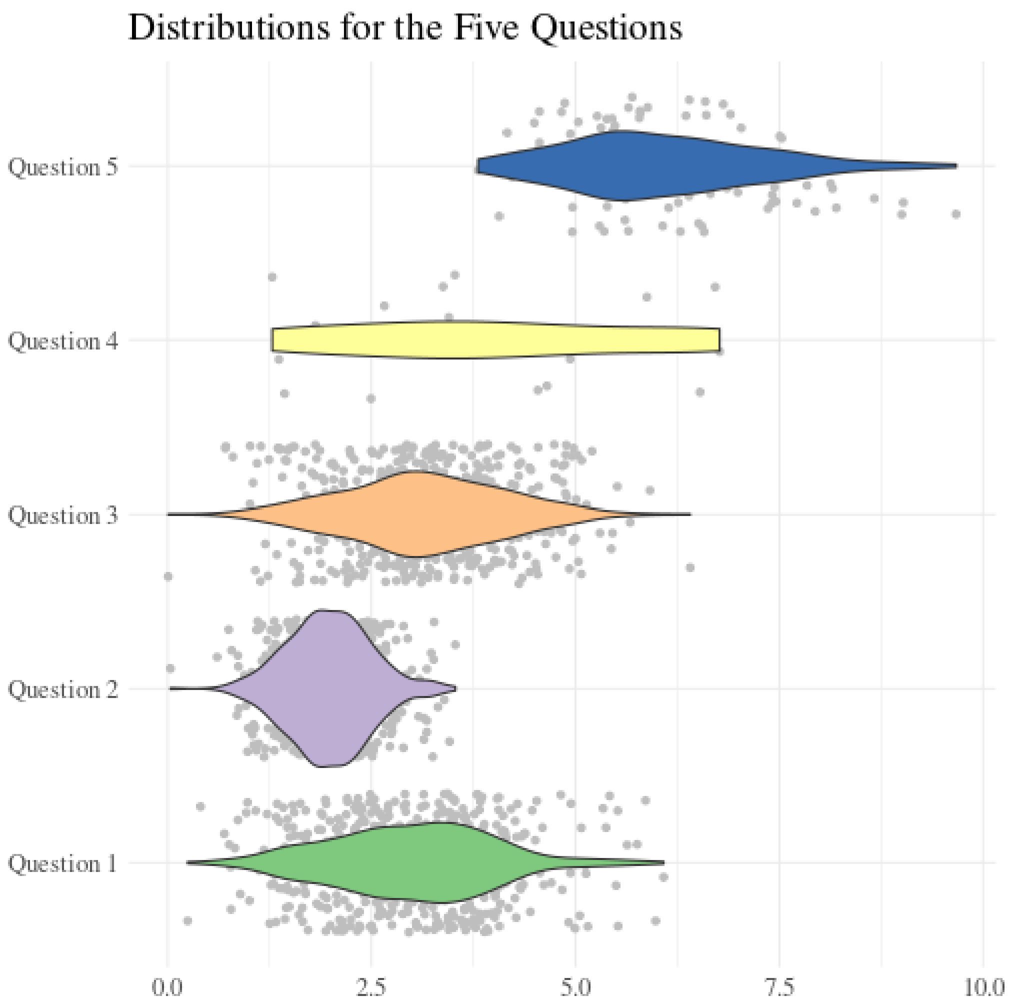
```r
data %>%
  ggplot(aes(x = name, y = value)) +
  geom_boxplot()
```

```r
data %>%
  ggplot(aes(x = value, y = name, fill = name)) +
  geom_jitter(color = "gray") +
  geom_violin() +
  theme_minimal() +
  scale_fill_brewer("", type = "qual") +
  ylab(NULL) +
  xlab(NULL) +
  theme(text = element_text(size = 16, family = "Times"),
        legend.position = "none") +
  ggtitle("Distributions for the Five Questions")
```

Review of last week's class

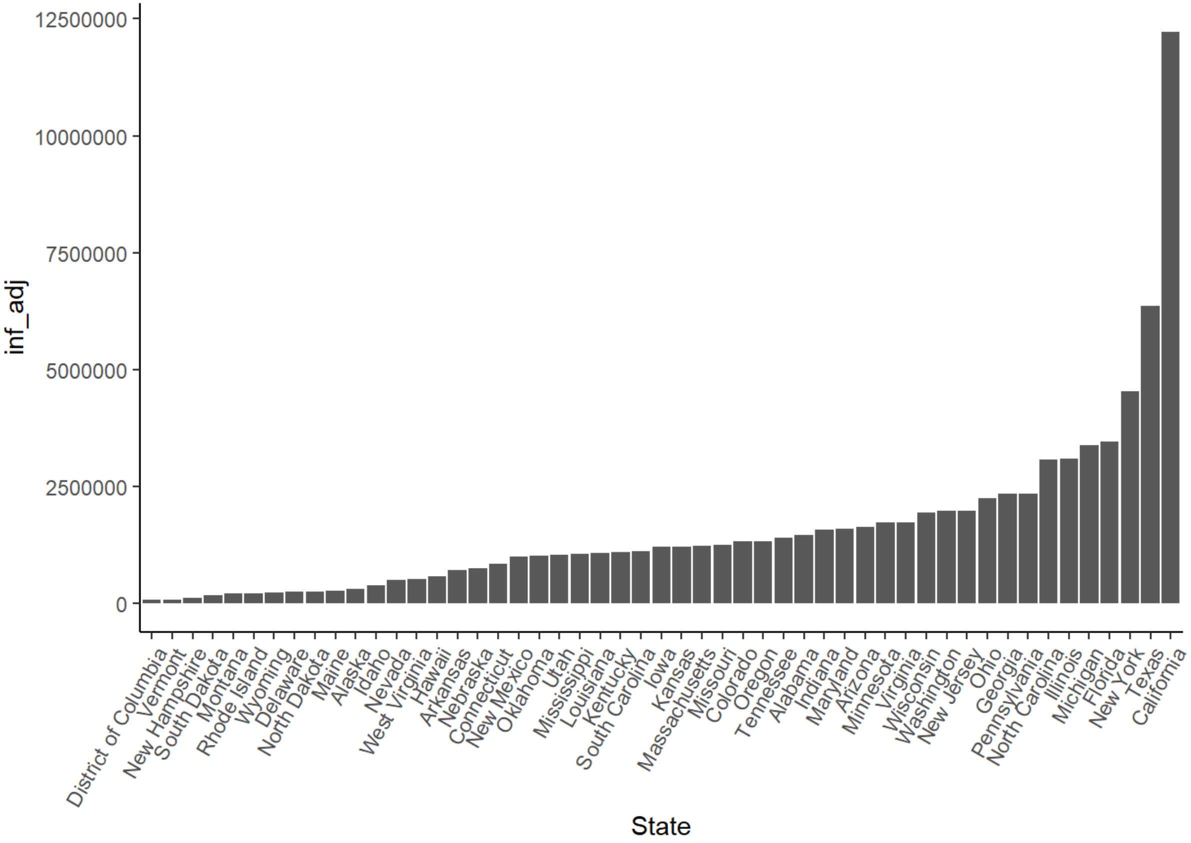


Review of last week's class



Homework highlights

What do you notice? What do you wonder about?



This week's topics

Overview

1. Data viz ideas and details
2. Data viz and tidying operations

2 overarching goals of learning data viz in R

- Conceptual framework of visualization
    - Grammar of graphics and different mappings of data onto visual elements
- Details of implementation
    - How to build and refine plots layer by layer
    - Eventually: Interactive data viz with ggviz and shiny

Part 1/2: Data Viz Ideas

1. Data Viz Ideas

Outline

A. Review of the grammar of graphics

B. Understanding visualizations by layers

C. Understanding mapping of data to geoms

1A: Grammar of Graphics

Another way to think about visualizing data is in terms of the elements that make up a plot.

The *grammar of graphics* (Wickham, 2010, Wilkinson, 2012) has a particular answer to the question of what a plot includes:

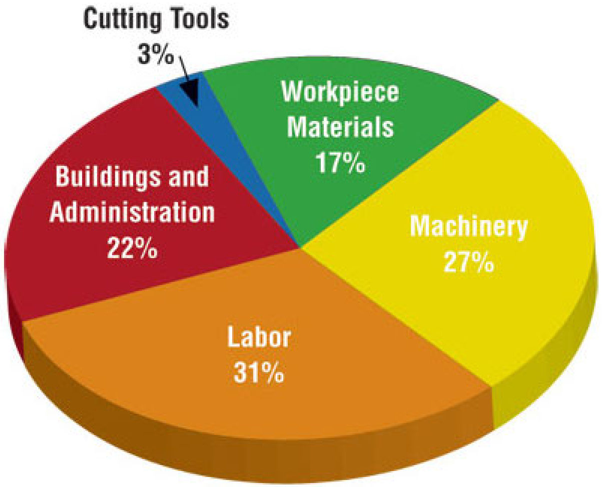
Why a grammar of graphics?

- gain insight into complex figures
- reveal deeper relationships between what may appear to be unrelated visualizations
- more flexibly and creatively visualize data––including in ways that do not fit well into one type of plot
- suggest what makes a good figure

1A: More Data Viz Ideas

Some general principles for effective data viz

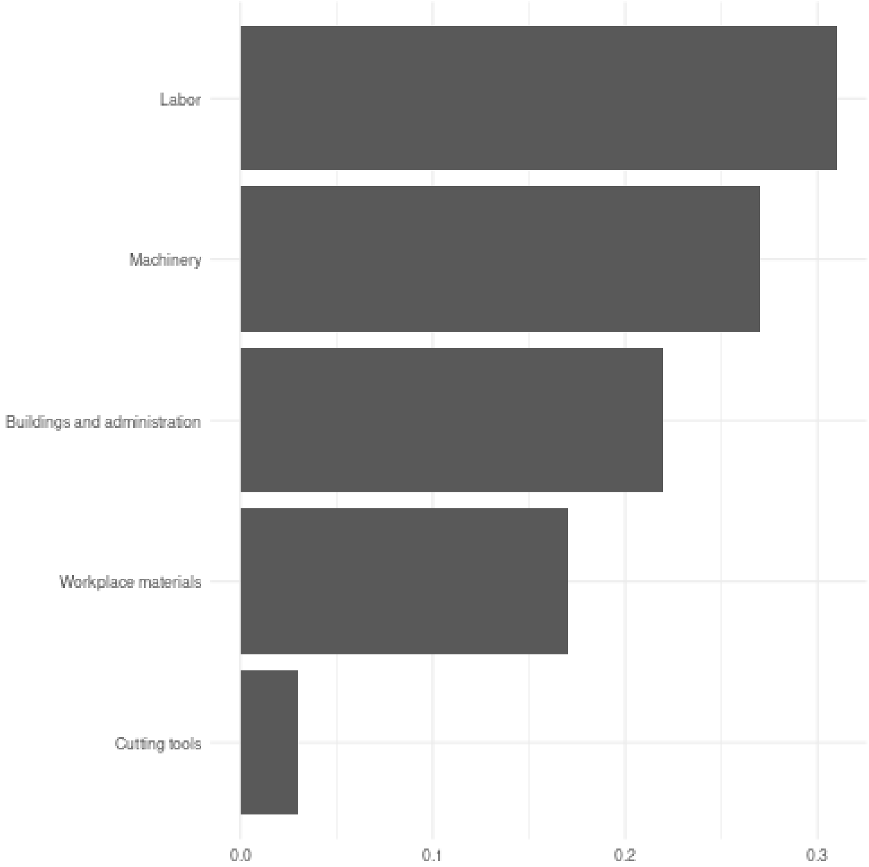
Keep it simple



1A: More Data Viz Ideas

Some general principles for effective data viz

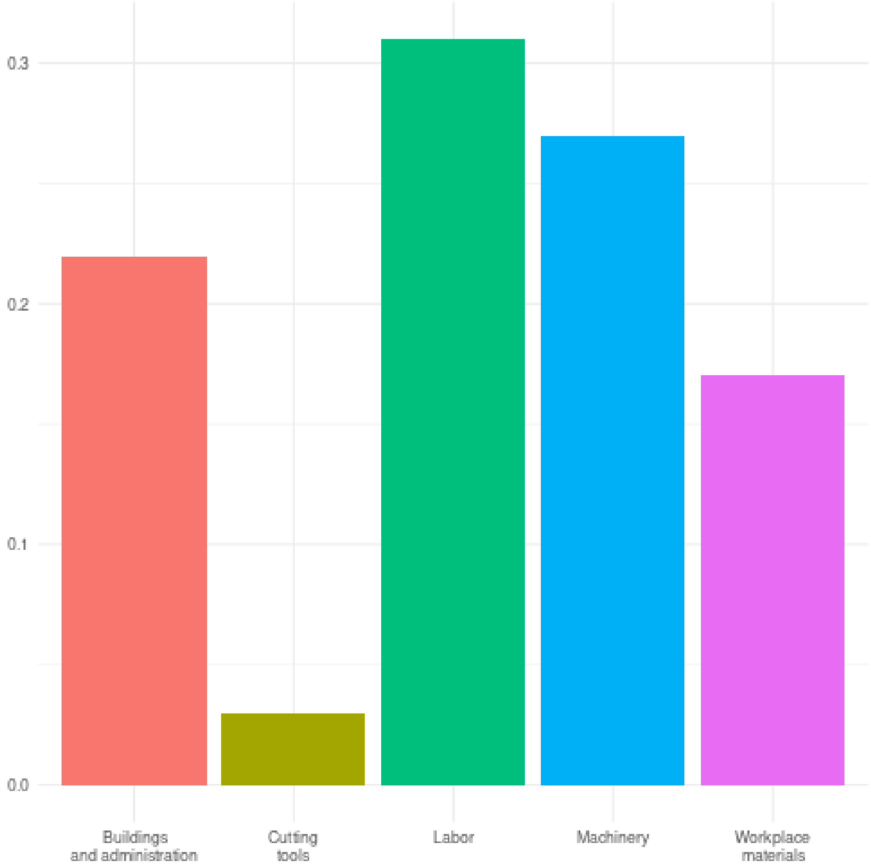
Keep it simple



1A: More Data Viz Ideas

Some general principles for effective data viz

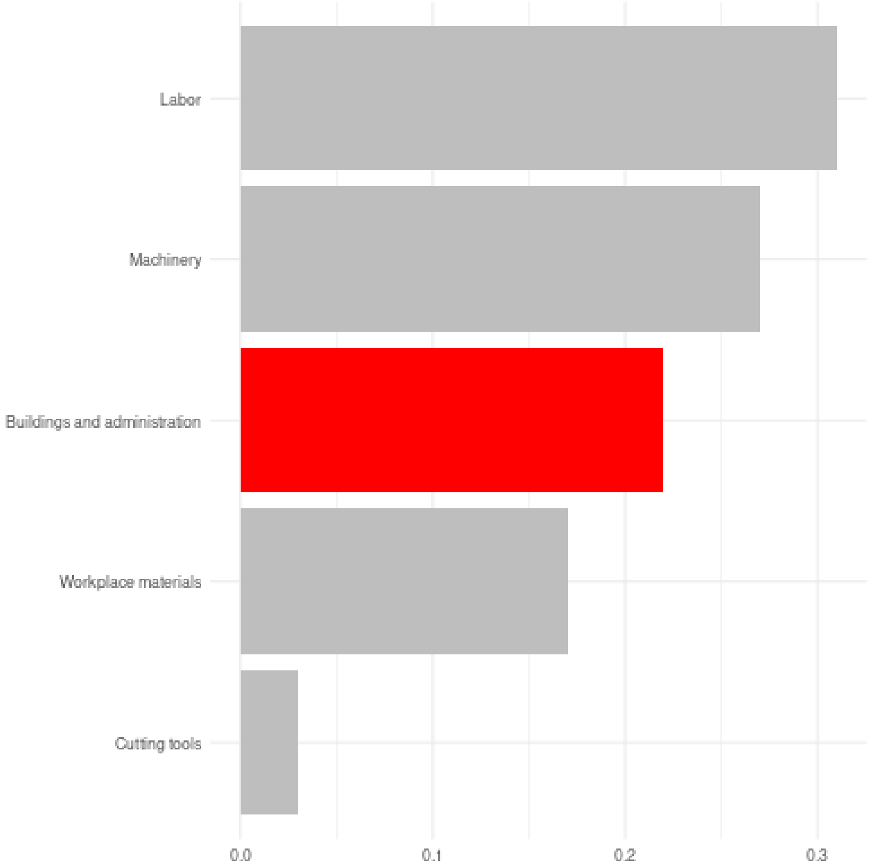
Use color to draw attention



1A: More Data Viz Ideas

Some general principles for effective data viz

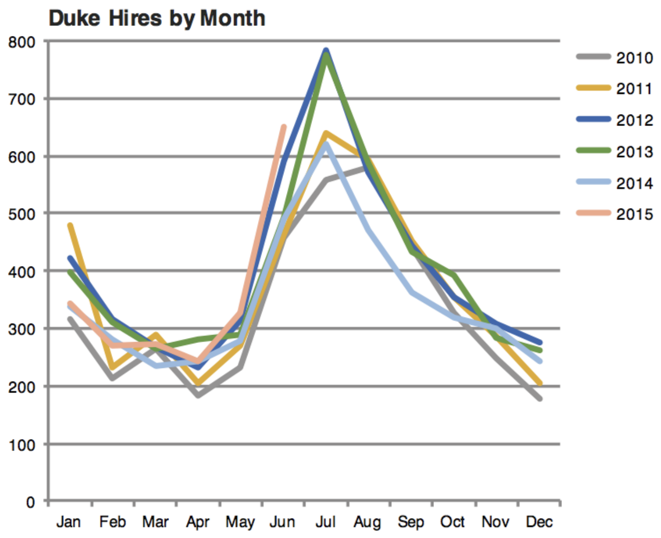
Use color to draw attention

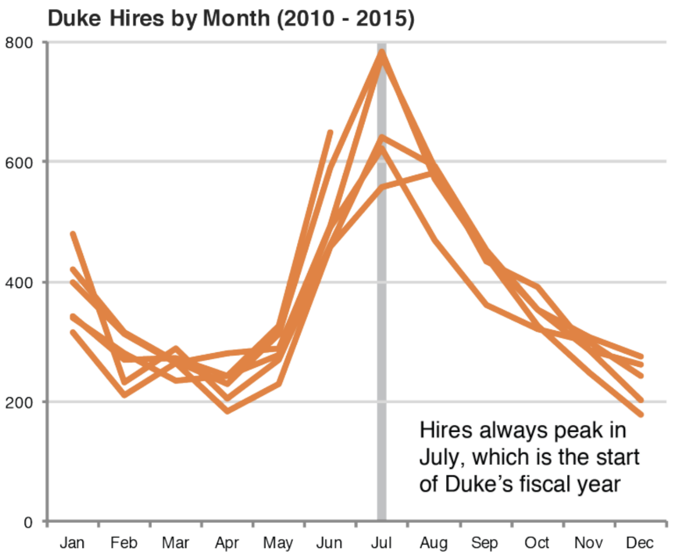


1A: More Data Viz Ideas

Some general principles for effective data viz

Tell a story





1B: Understanding visualizations by layer

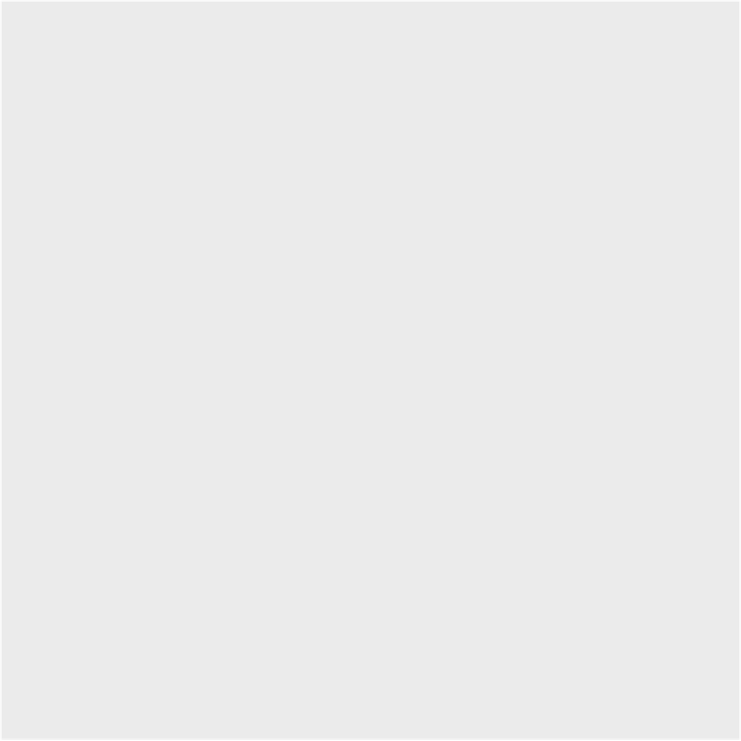
Layers:

1. Data
2. One or more geometric objects (shape, point, line, etc.)
3. A mapping between variables in the data and the geometric objects and their characteristics (including their size and color)
4. A theme

1B: Understanding visualizations by layer

Data

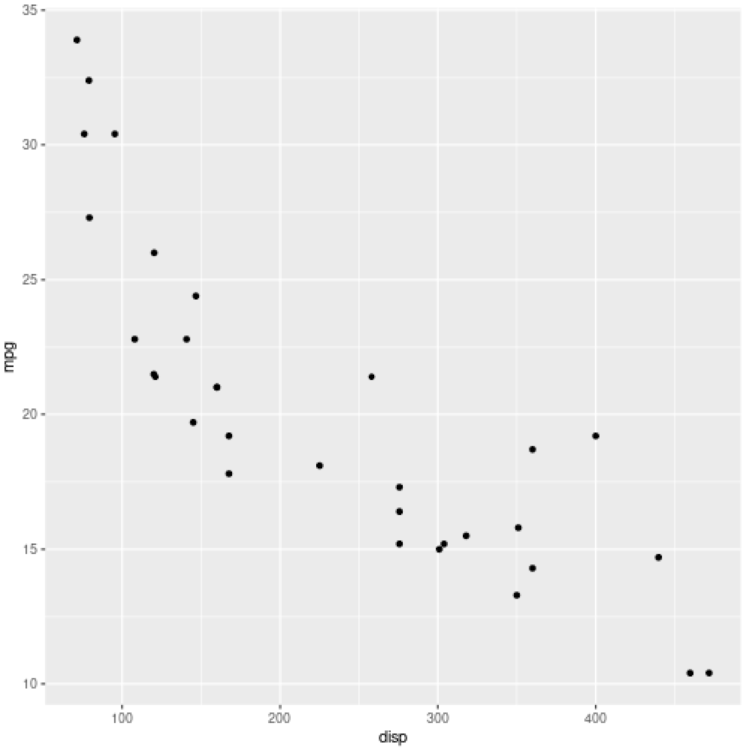
```
ggplot(mtcars)
```



1B: Understanding visualizations by layer

One geom

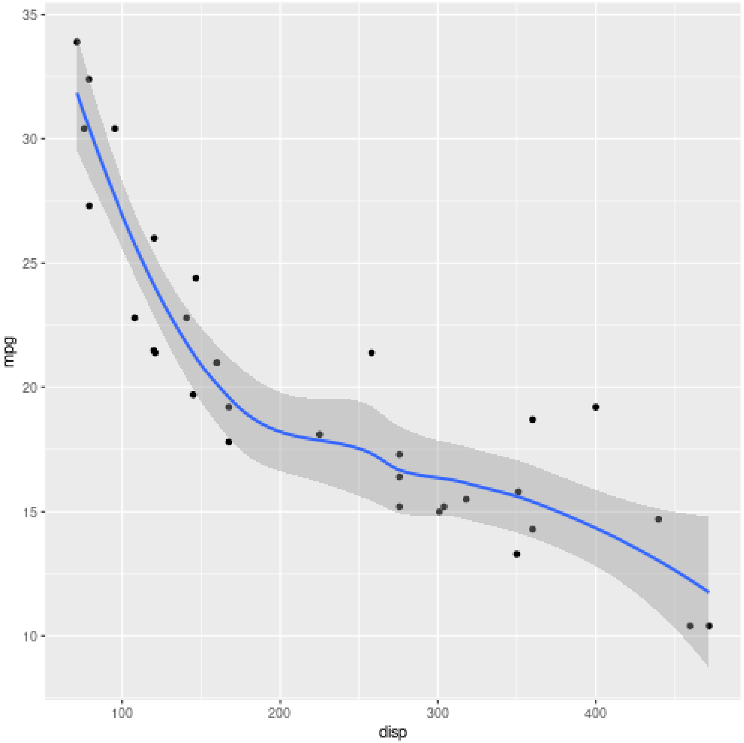
```
ggplot(mtcars) +
  geom_point(aes(x = disp, y = mpg))
```



1B: Understanding visualizations by layer

Additional Geoms

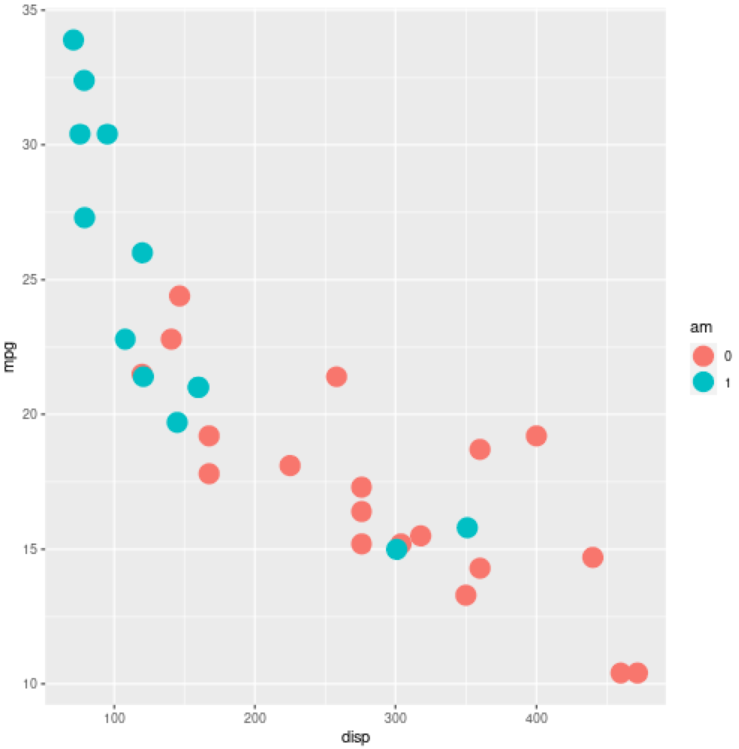
```
ggplot(mtcars) +
  geom_point(aes(x = disp, y = mpg)) +
  geom_smooth(aes(x = disp, y = mpg), method = "loess")
```



1B: Understanding visualizations by layer

Additional Aesthetic Parameters: Color

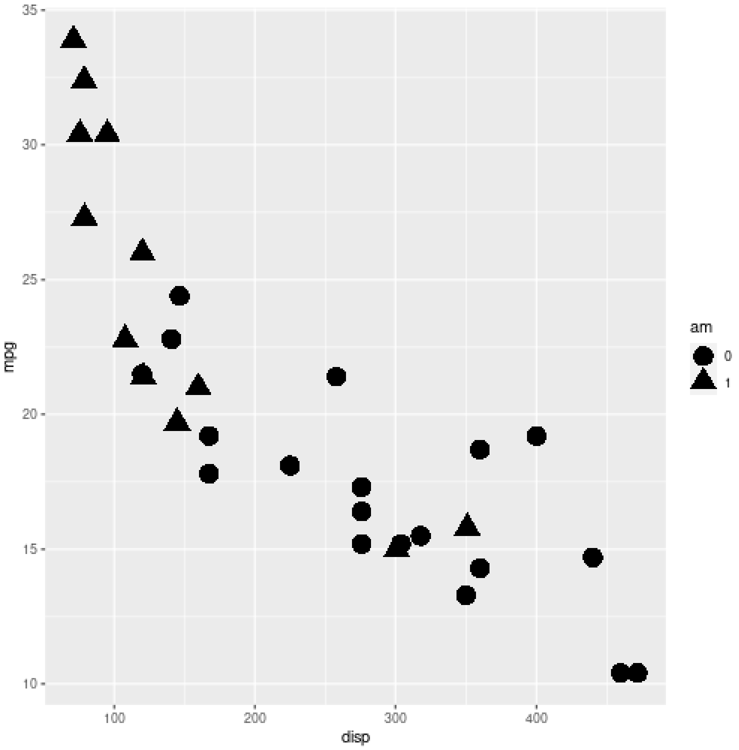
```
ggplot(mtcars) +
  geom_point(aes(x = disp, y = mpg, color = am), size = 6)
```



1B: Understanding visualizations by layer

Additional Aesthetic Parameters: Shape

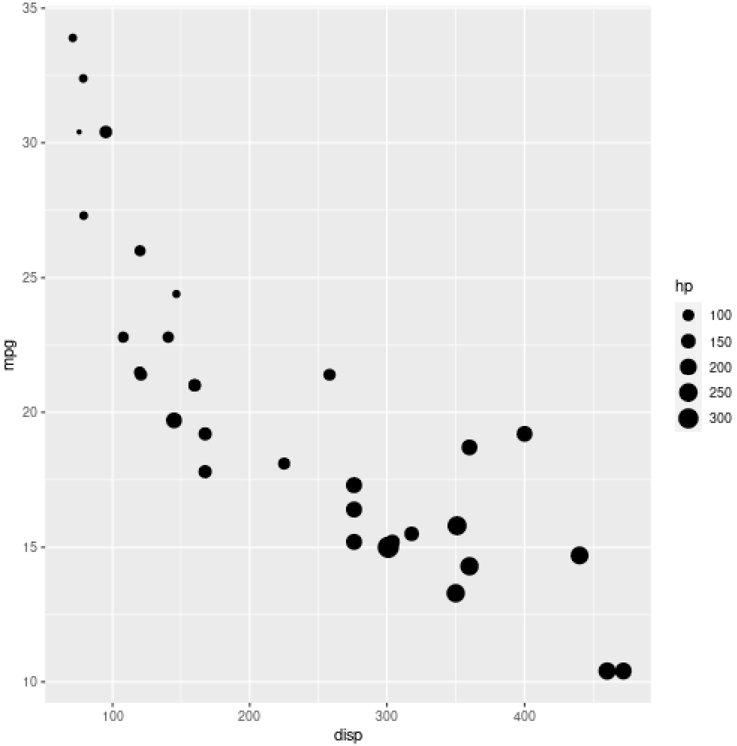
```
ggplot(mtcars) +
  geom_point(aes(x = disp, y = mpg, shape = am), size = 6)
```



1B: Understanding visualizations by layer

Additional Aesthetic Parameters: Size

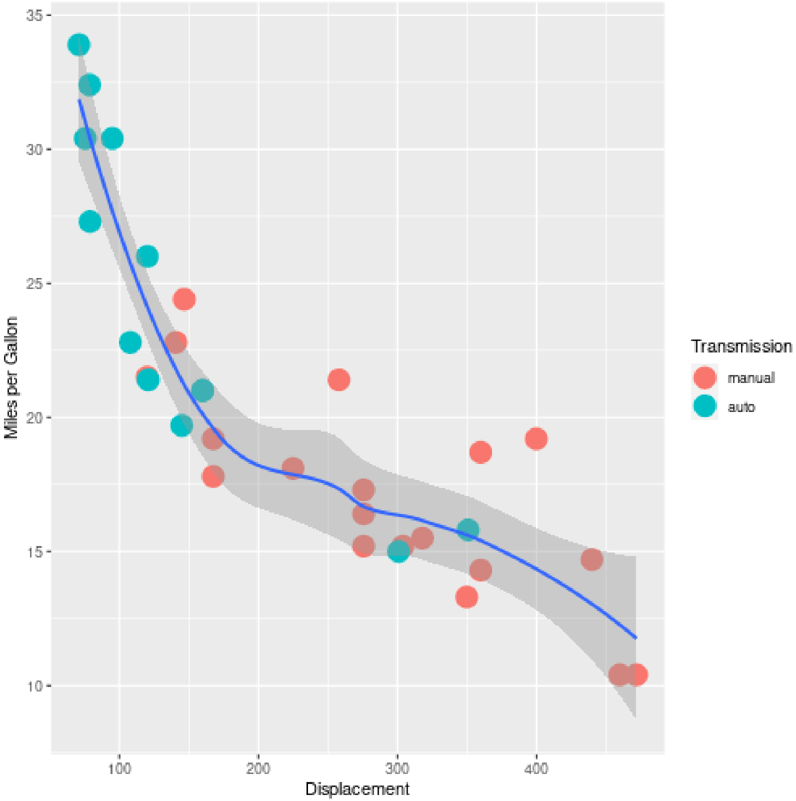
```
ggplot(mtcars) +
  geom_point(aes(x = disp, y = mpg, size = hp))
```



1B: Understanding visualizations by layer

Theme: labels

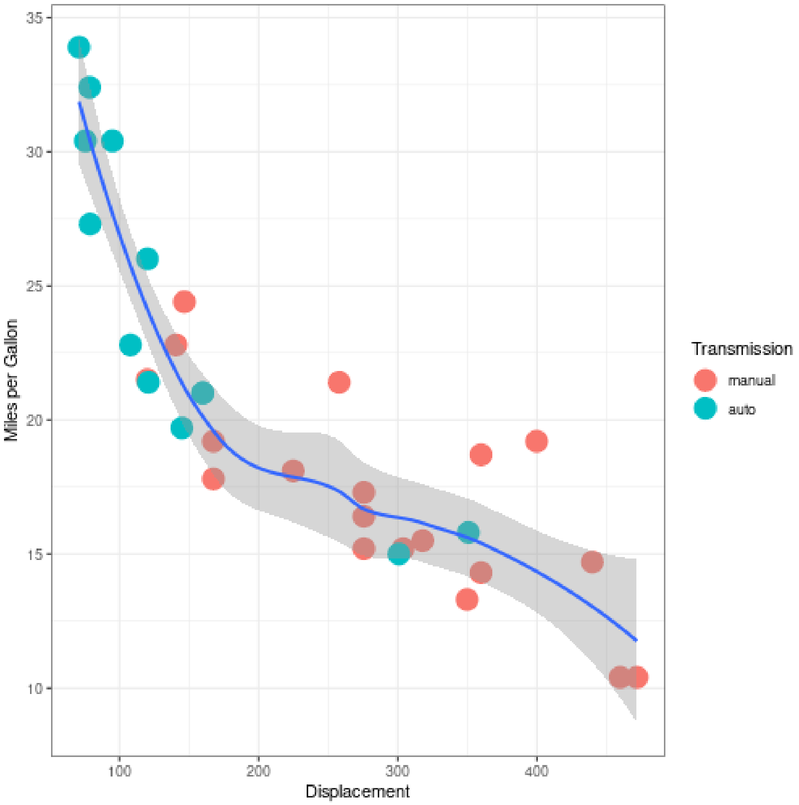
```
# code chunk here
ggplot(mtcars) +
  geom_point(aes(x = disp, y = mpg, col
  geom_smooth(aes(x = disp, y = mpg), m
  xlab("Displacement") +
  ylab("Miles per Gallon") +
  scale_color_discrete(name = "Transmis
```



1B: Understanding visualizations by layer

Theme: overall

```
# code chunk here
ggplot(mtcars) +
  geom_point(aes(x = disp, y = mpg, col
  geom_smooth(aes(x = disp, y = mpg), m
  xlab("Displacement") +
  ylab("Miles per Gallon") +
  scale_color_discrete(name = "Transmis
  theme_bw()
```



1B: Understanding visualizations by layer

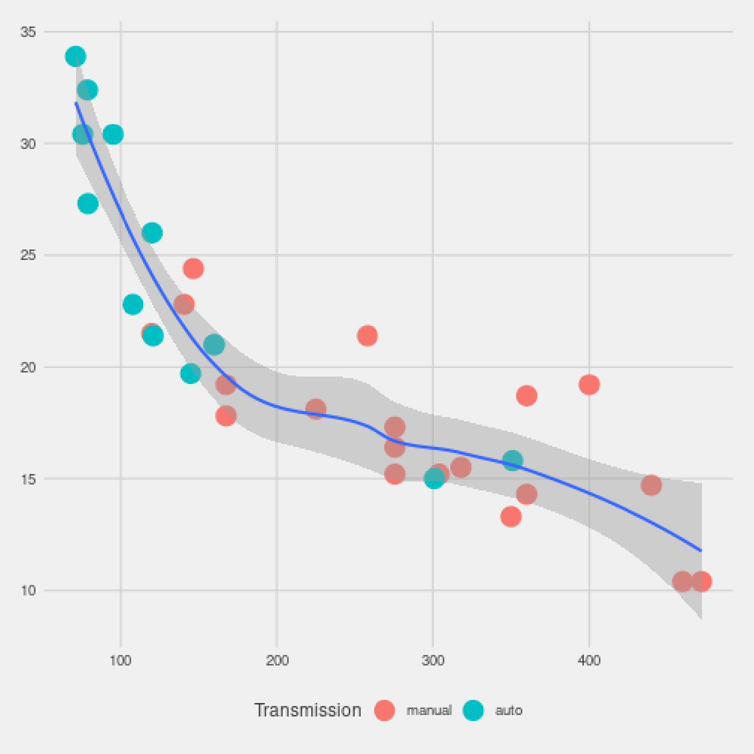
ggthemes package

```r
library(ggthemes)

base_plot <- ggplot(mtcars) +
  geom_point(aes(x = disp, y = mpg, color = am), size = 6) +
  geom_smooth(aes(x = disp, y = mpg), method = "loess") +
  xlab("Displacement") +
  ylab("Miles per Gallon") +
  scale_color_discrete(name = "Transmission", labels = c("manual", "auto"))
```

1B: Understanding visualizations by layer

Fivethirtyeight style

```
base_plot + theme_fivethirtyeight()
```



1C: Understanding mapping data to geoms

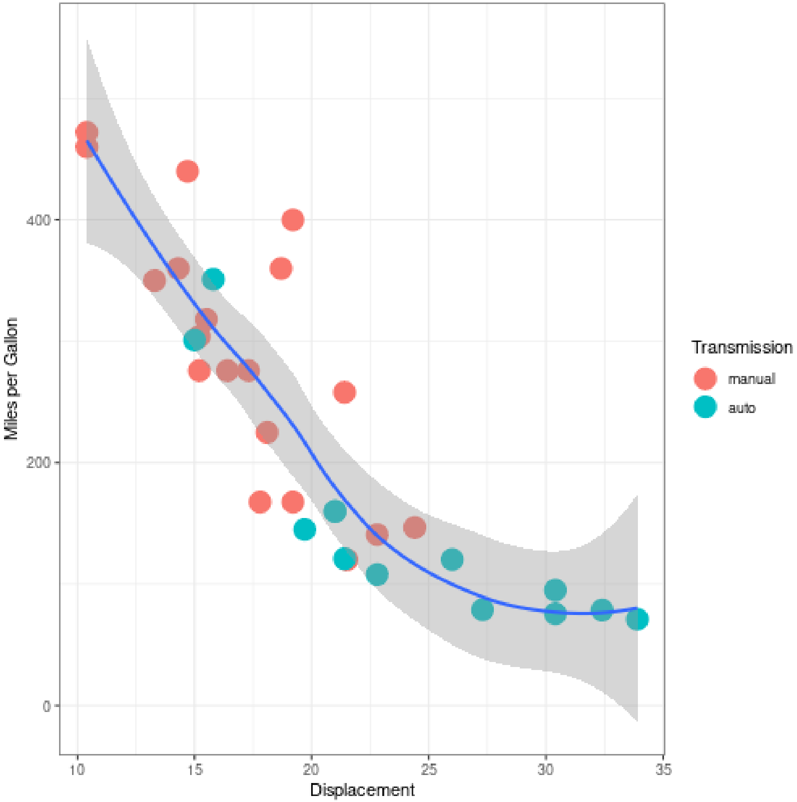
You can create different plots by:

- Changing the aesthetic *mapping* between variables in the data and geometric objects
- Changing the geometric objects

1C: Understanding mapping data to geoms

Changing the mapping

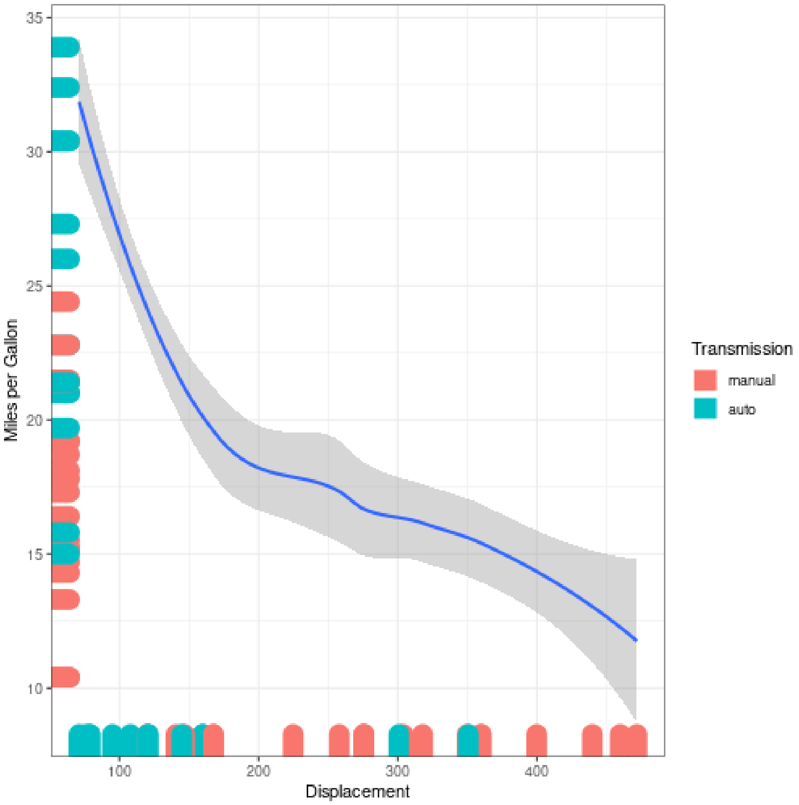
```
# code chunk here
ggplot(mtcars) +
  geom_point(aes(y = disp, x = mpg, col
  geom_smooth(aes(y = disp, x = mpg), m
  xlab("Displacement") +
  ylab("Miles per Gallon") +
  scale_color_discrete(name = "Transmis
  theme_bw()
```



1C: Understanding mapping data to geoms

Changing geoms

```
# code chunk here
ggplot(mtcars) +
  geom_rug(aes(x = disp, y = mpg, color
  geom_smooth(aes(x = disp, y = mpg), m
  xlab("Displacement") +
  ylab("Miles per Gallon") +
  scale_color_discrete(name = "Transmis
  theme_bw()
```



Part 2/2: Data Viz and Tidying

2: How does tidying data relate to data viz?

Often, we have to make changes to our data frame in order to create the visualization we would like to create.

Making a new variable prior to plotting the data

Other data tidying steps we might take prior to visualizing data:

- **recoding** variables
- **creating a factor** (so that we can order elements of a plot as we wish for them to be ordered)
- **grouping** and **summarizing** to plot a summary statistic
- realizing that your data processing and tidying was not quite sufficient, so **returning to those stages** before finalizing your visualization
- **re-running our analysis** (`.Rmd` file) because we discovered an issue with our data

2: How does tidying data relate to data viz?

Sometimes we need to recode a variable or add a new one

```r
tidykids <- read_csv(here("data", "tidykids.csv"))
```

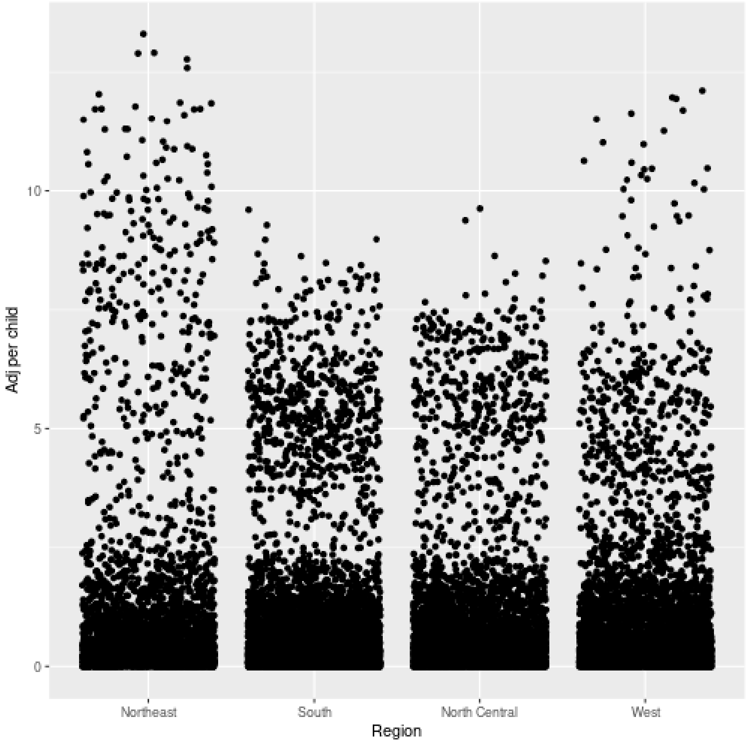
```
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   state = col_character(),
##   variable = col_character(),
##   year = col_double(),
##   raw = col_double(),
##   inf_adj = col_double(),
##   inf_adj_perchild = col_double()
## )
```

```r
state_region <- data.frame(state.name, state.region)

tidykids_reg <- left_join(tidykids, state_region, by = c("state" = "state.name"))

tidykids_reg$timeblock <- recode(tidykids_reg$year,
       `1997` = "1997-2001", `1998` = "1997-2001", `1999` = "1997-2001", `2000` = "1997-2001", `20
       `2002` = "2002-2006", `2003` = "2002-2006", `2004` = "2002-2006", `2005` = "2002-2006", `20
       `2007` = "2007-2011", `2008` = "2007-2011", `2009` = "2007-2011", `2010` = "2007-2011", `20
       `2012` = "2012-2016", `2013` = "2012-2016", `2014` = "2012-2016", `2015` = "2012-2016", `20
```

2: How does tidying data relate to data viz?

Sometimes we need to recode a variable for plotting

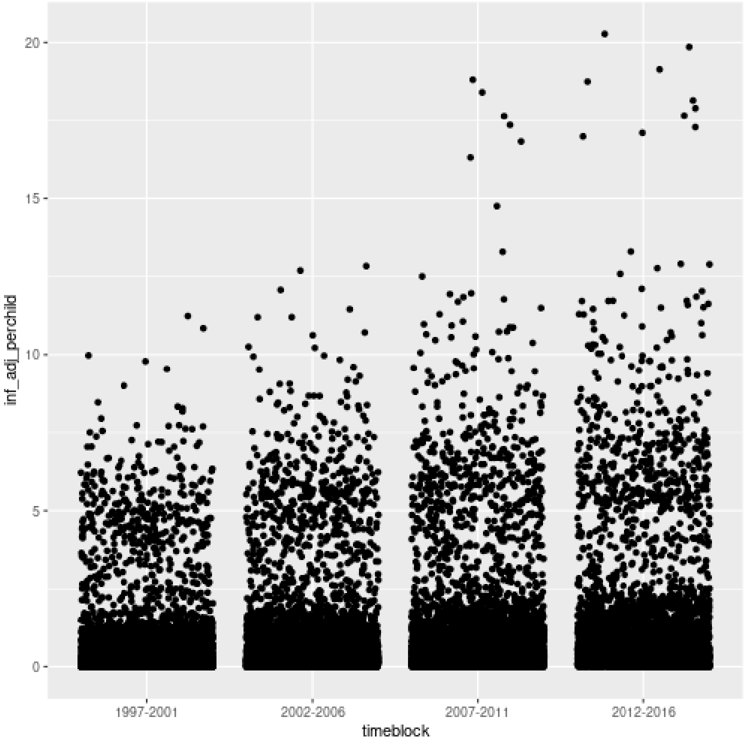
```
ggplot(na.omit(tidykids_reg)) +
  geom_jitter(aes(x = state.region, y = inf_adj_perchild)) +
  xlab("Region") +
  ylab("Adj per child")
```



2: How does tidying data relate to data viz?

Creating and reordering factors is often useful

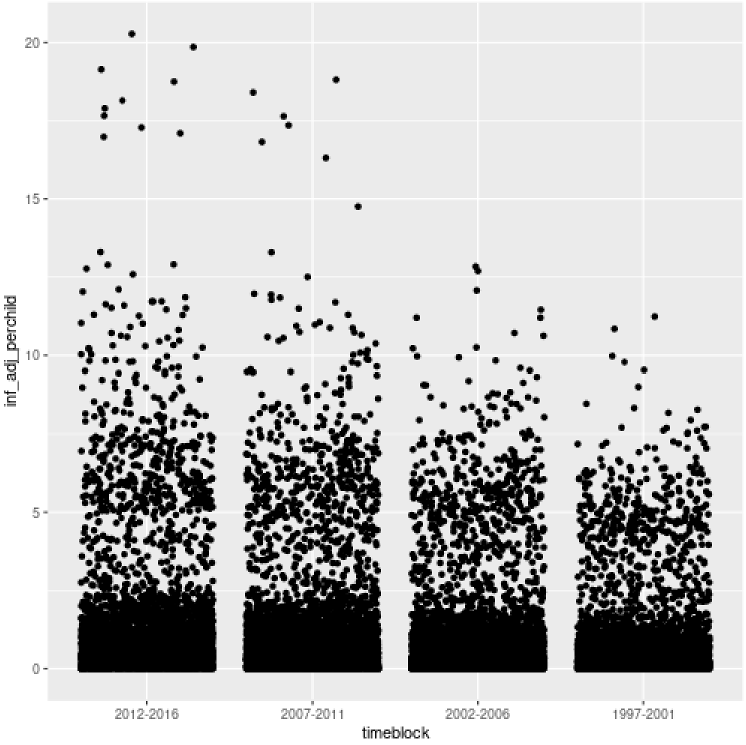
```
tidykids_reg <- tidykids_reg %>%
  mutate(timeblock = factor(timeblock))
ggplot(tidykids_reg) +
  geom_jitter(aes(timeblock, inf_adj_perchild))
```



2: How does tidying data relate to data viz?

Creating and reordering factors is often useful

```
tidykids_reg$timeblock <- fct_relevel(tidykids_reg$timeblock, c("2012-2016", "2007-2011", "2002-2(

ggplot(tidykids_reg) +
  geom_jitter(aes(timeblock, inf_adj_perchild))
```



2: How does tidying data relate to data viz?

When we do `group_by()` and `summarize()` we can plot summary statistics

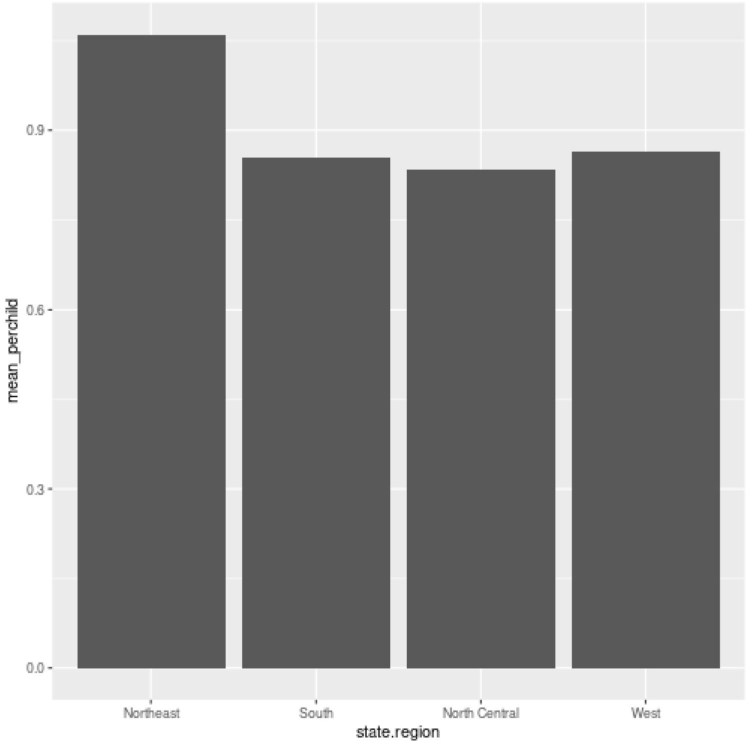
```
summm_df <- na.omit(tidykids_reg) %>%
  group_by(state.region) %>%
  summarize(mean_perchild = mean(inf_adj_perchild, na.rm = T))

summ_df
```

```
## # A tibble: 4 x 2
##   state.region  mean_perchild
## * <fct>                 <dbl>
## 1 Northeast             1.06
## 2 South                 0.855
## 3 North Central         0.834
## 4 West                  0.865
```

2: How does tidying data relate to data viz?

When we do `group_by()` and `summarize()` we can plot summary statistics

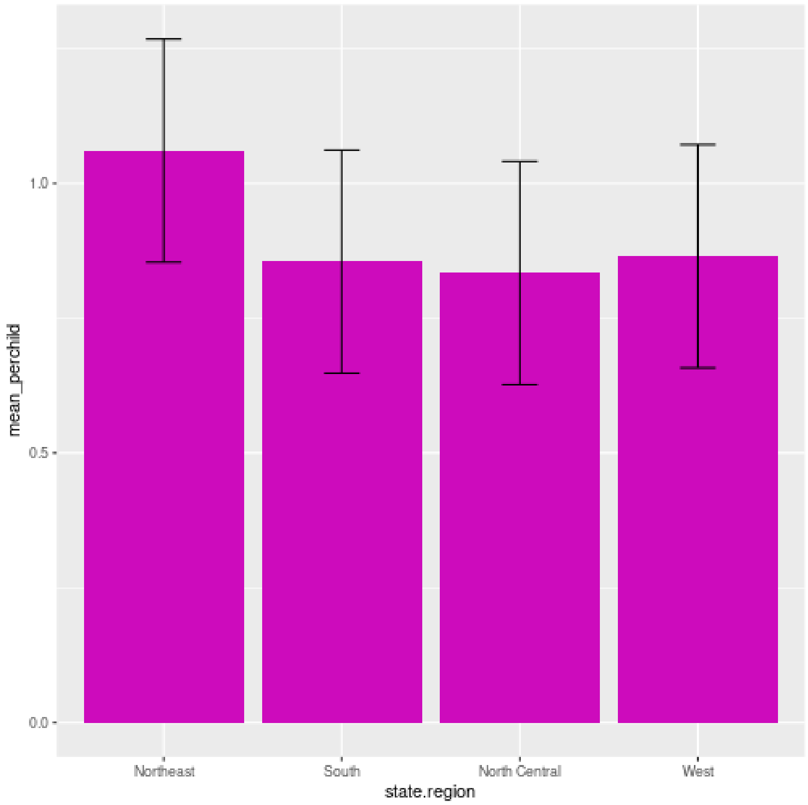
```
summm_df %>%
  ggplot() +
    geom_col(aes(state.region, mean_perchild))
```



2: How does tidying data relate to data viz?

When we do `group_by()` and `summarize()` we can plot summary statistics

```
# code chunk here
  summ_df %>%
    ggplot() +
      geom_bar(aes(state.region, mean_p
      geom_errorbar(aes(x = state.regic
                 position=position_dodge
```



Course Logistics

This week

- Homework 6: Available tommorow by noon; Due by Thursday, 3/4
- Readings
  - 1: https://clauswilke.com/dataviz/histograms-density-plots.html
  - 2: https://clauswilke.com/dataviz/visualizing-proportions.html

Coming up

- Data ethics
- *Just begin* to think and to ask questions about what you may want to do for a final project; something that will advance your research and allow you to exhibit and extend what you do in class

Random

- https://educationdata.urban.org/documentation/schools.html
- https://leanpub.com/tidyverseskillsdatascience

Wrapping up

In your base group's Slack channel:

- What is one thing you learned today?
- What is something you want to learn more about?
- *Also*, in GIF form (type `/giphy` in Slack, and then a random term), summarize how you are feeling about R