Accelerated TA Session 9: Data Visualization – Tidyverse

Harris Coding Camp

Summer 2022

General Guidelines

You may encounter some functions we did not cover in the lectures. This will give you some practice on how to use a new function for the first time. You can try following steps:

1. Start by typing `?new_function` in your Console to open up the help page
2. Read the help page of this `new_function`. The description might be a bit technical for now. That's OK. Pay attention to the Usage and Arguments, especially the argument `x` or `x,y` (when two arguments are required)
3. At the bottom of the help page, there are a few examples. Run the first few lines to see how it works
4. Apply it in your questions

It is highly likely that you will encounter error messages while doing this exercise. Here are a few steps that might help get you through it:

1. Locate which line is causing this error first
2. Check if you have a typo in the code. Sometimes your group members can spot a typo faster than you.
3. If you enter the code without any typo, try googling the error message. Scroll through the top few links see if any of them helps
4. Try working on the next few questions while waiting for help by TAs

Data and background

1. We'll work with data sets from `recent_college_grads.dta`, which you can download here. This is data on college majors and earnings, specifically the data behind the FiveThirtyEight story "The Economic Guide To Picking A College Major".

2. Load the packages you'll need: `tidyverse` and the package with code to read `dta` files)[1].

3. Load the data and examine it. By now, you should be familiar with the introductory questions. What are the column names? Some of them are not entirely obvious and well work through them! (In general, you want to get your hands on a code book.) How many rows are there? Is the data tidy? Is a row a single observation and if so what's an observation in this data?

[1] `haven`

I. Manipulating College Data

How do the distributions of median income compare across major categories?

A percentile is a measure used in statistics indicating the value below which a given percentage of observations in a group of observations fall. For example, the 20th percentile is the value below which 20% of the observations may be found.

There are three types of incomes reported in this data frame: `p25th`, `median`, and `p75th`. These correspond to the 25th, 50th, and 75th percentiles of the income distribution of sampled individuals for a given major.

We need to do a few things to answer this question "How do the distributions of median income compare across major categories?". First, we need to group the data by `major_category`. Then, we need a way to summarize the distributions of median income within these groups. This decision will depend on the shapes of these distributions. So first, we need to visualize the data.

1. Let's first take a look at the distribution of all median incomes using `geom_histogram`, without considering the major categories.

```
ggplot(data = ____,
       mapping = aes(x = median)) +
  geom_histogram()
```

2. Try binwidths of 1000 and 5000 and choose one. Explain your reasoning for your choice.

```
ggplot(data = ___,
       mapping = aes(x = median)) +
  geom_histogram(binwidth = ___)
```

We can also calculate summary statistics for this distribution using the `summarize` function:

```
college_recent_grads %>%
  summarize(min = min(median), max = max(median),
            mean = mean(median), med = median(median),
            sd = sd(median),
            q1 = quantile(median, probs = 0.25),
            q3 = quantile(median, probs = 0.75))
```

```
## # A tibble: 1 x 7
##     min    max   mean   med     sd    q1    q3
##   <dbl>  <dbl>  <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1 22000 110000 40151. 36000 11470. 33000 45000
```

3. Based on the shape of the histogram you created in the previous part, determine which of these summary statistics above (min, max, mean, med, sd, q1, q3) is/are useful for describing the distribution. Write up your description and include the summary statistic output as well. You can pick single/multiple statistics and briefly explain why you pick it/them.

4. Next, we facet the plot by major category. Plot the distribution of `median` income using a histogram, faceted by `major_category`. Use the binwidth you chose in part 4.

```r
ggplot(data = ___,
       mapping = aes (x=median)) +
  geom_histogram(bindwidth = ___) +
  # To specify the variable to facet on
  # ~var_name and vars(var_name) both work!
  facet_wrap(~major_category)
```

5. Use `filter` to find out which major has the highest median income? lowest? Which major has the median income? Hint: refer to the statistics in part 4.

```r
college_recent_grads %>%
  ____(median == ____)
```

6. Which major category is the most popular in this sample? To answer this question, we use a new function called `count`, which first groups the data, then counts the number of observations in each category and store the counts into a column named `n`. Add to the pipeline appropriately to arrange the results so that the major with the highest observations is on top.

```r
college_recent_grads %>%
  count(major_category) %>%
  ___(___(n))
```

What types of majors do women tend to major in?

First, let's create a new vector called `stem_categories` that lists the major categories that are considered STEM fields.

```r
stem_categories <- c("Biology & Life Science",
                     "Computers & Mathematics",
                     "Engineering",
                     "Physical Sciences")
```

7. Then, we can use this to add a new variable indicating whether a major is STEM or not.
8. Create a scatterplot of median income vs. proportion of women in that major, colored by whether the major is in a STEM field or not. Describe the association between these three variables.

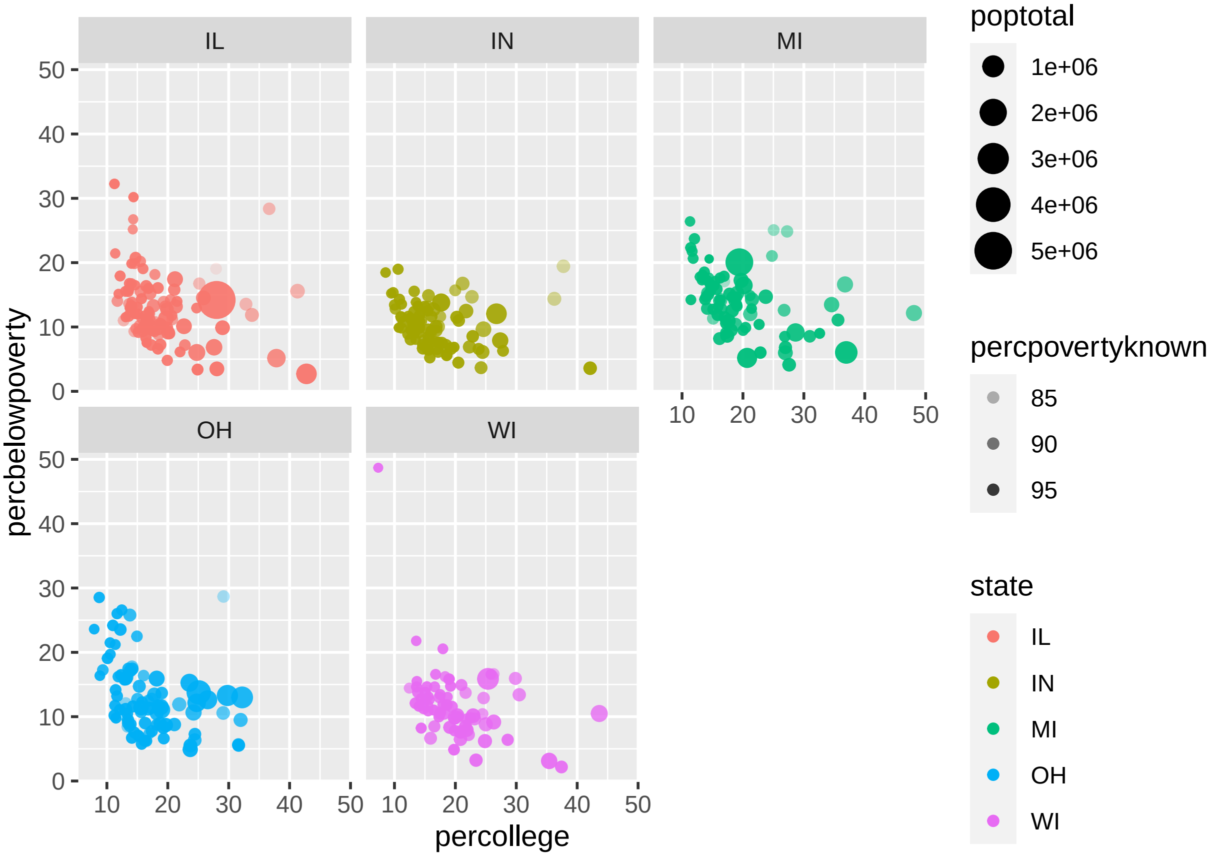
```r
ggplot(data = ___,
       mapping = aes(x = ___,
                     y = ___,
                     color = ___)) +
       geom_point()
```

9. We can use the logical operators to also `filter` our data for STEM majors whose median earnings is less than median for all majors's median earnings, which we found to be $36,000 earlier. Your output should only show the major name and median, 25th percentile, and 75th percentile earning for that major and should be sorted such that the major with the lowest median earning is on top.

II. Manipulating Midwest Data

Recall ggplot works by mapping data to aesthetics and then telling ggplot how to visualize the aesthetic with geoms. Like so:

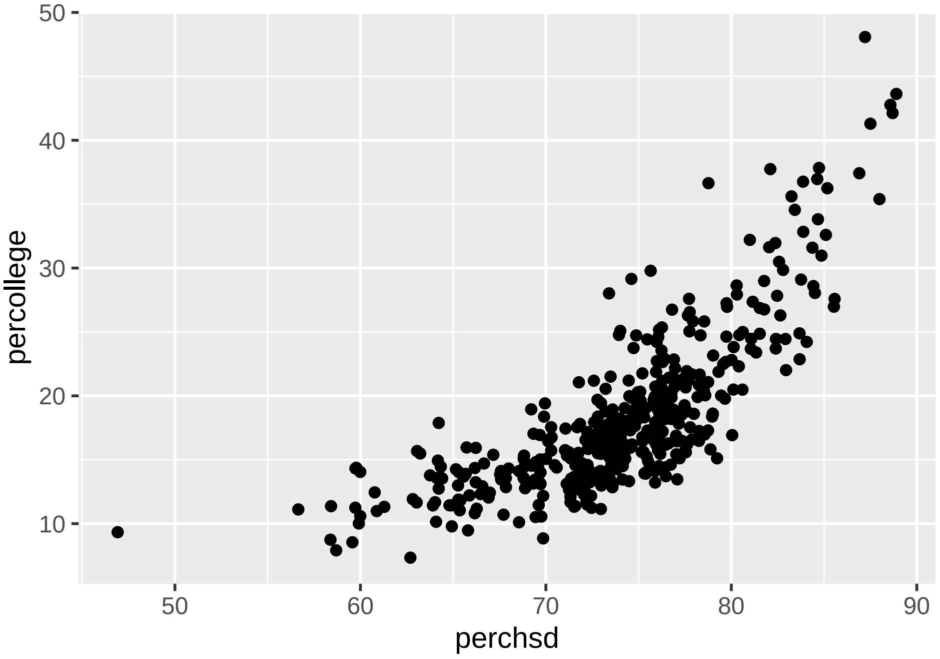
```
midwest %>%
  ggplot(aes(x = percollege,
             y = percbelowpoverty,
             color = state,
             size = poptotal,
             alpha = percpovertyknown)) +
  geom_point() +
  facet_wrap(vars(state))
```



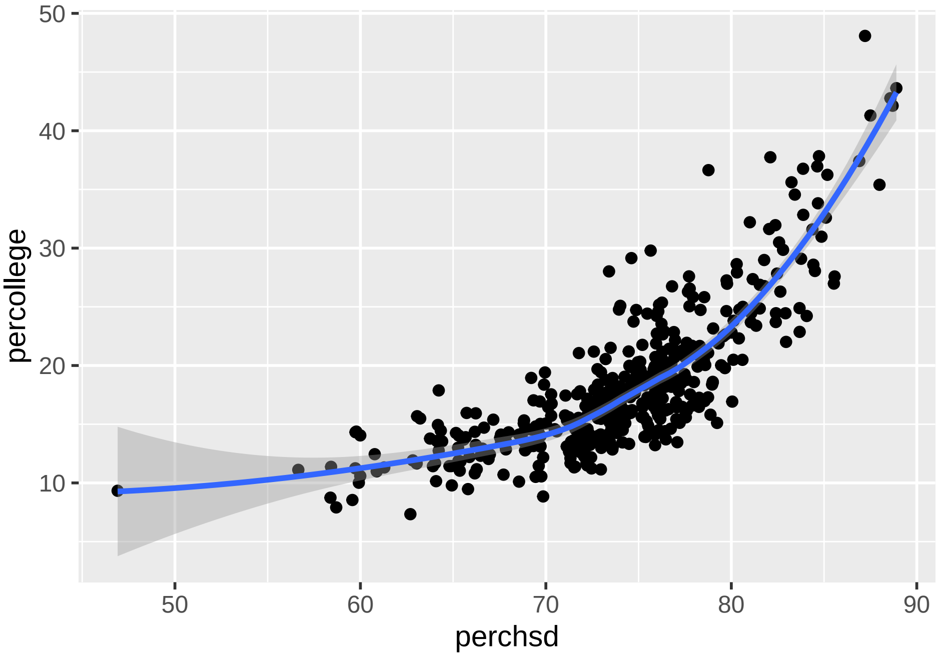
1. Which is more highly correlated with poverty at the county level, college completion rates or high school completion rates? Is it consistent across states? Change one line of code in the above graph.

`geoms`

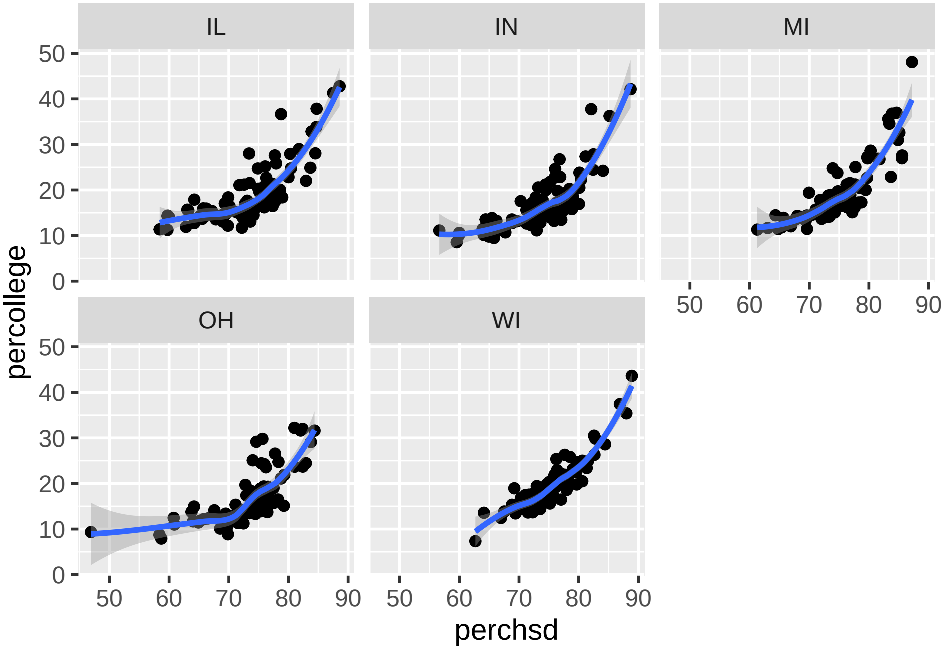
For the following, write code to reproduce each plot using `midwest`:



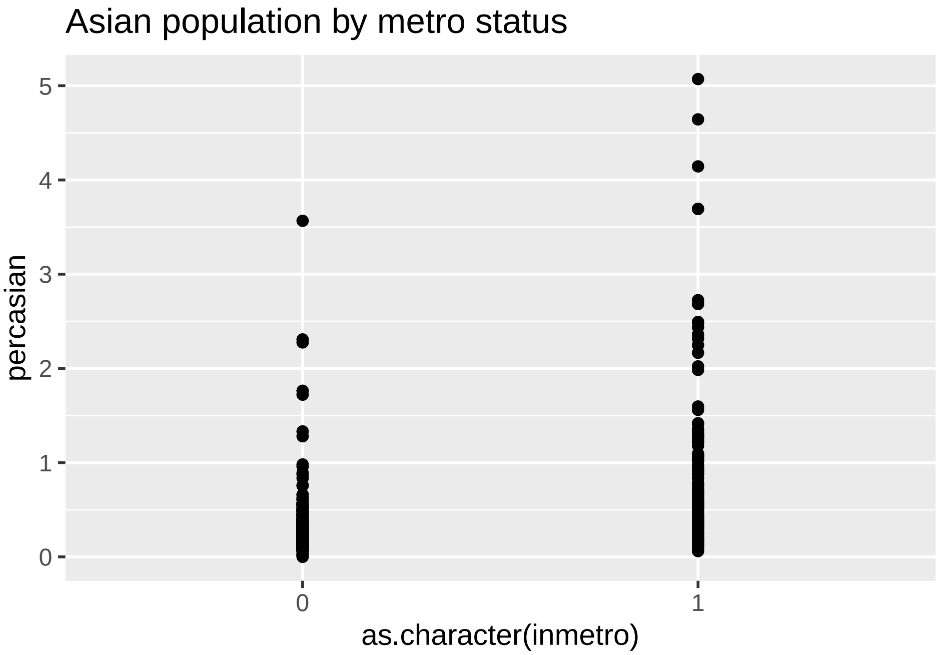
1.



2.



3.



4.

```
midwest %>%
  ggplot(aes(x = ..., y = ...)) +
  geom_point() +
  labs(title = "Asian population by metro status")
```

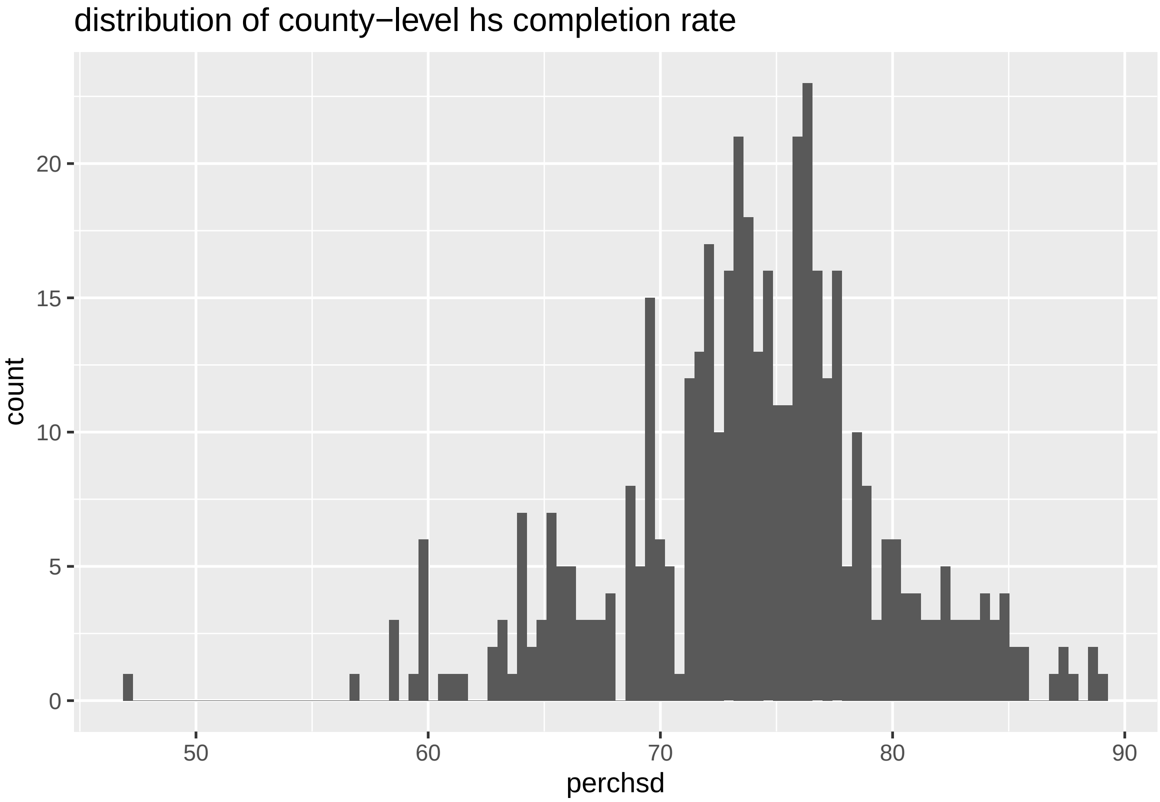
Notice that `inmetro` is numeric, but I want it to behave like a discrete variable so I use `x = as.character(inmetro)`. Complete the code above for part 4.

5. Use `geom_boxplot()` instead of `geom_point()` for "Asian population by metro status".

6. Use geom_jitter() and geom_boxplot() at the same time for "Asian population by metro status". Does order matter?

7. Histograms are used to visualize distributions. What happens when you change the bins argument? What happens if you leave the bins argument off?

```
midwest %>%
  ggplot(aes(x = perchsd)) +
  geom_histogram(bins = 100) +
  labs(title = "distribution of county-level hs completion rate")
```



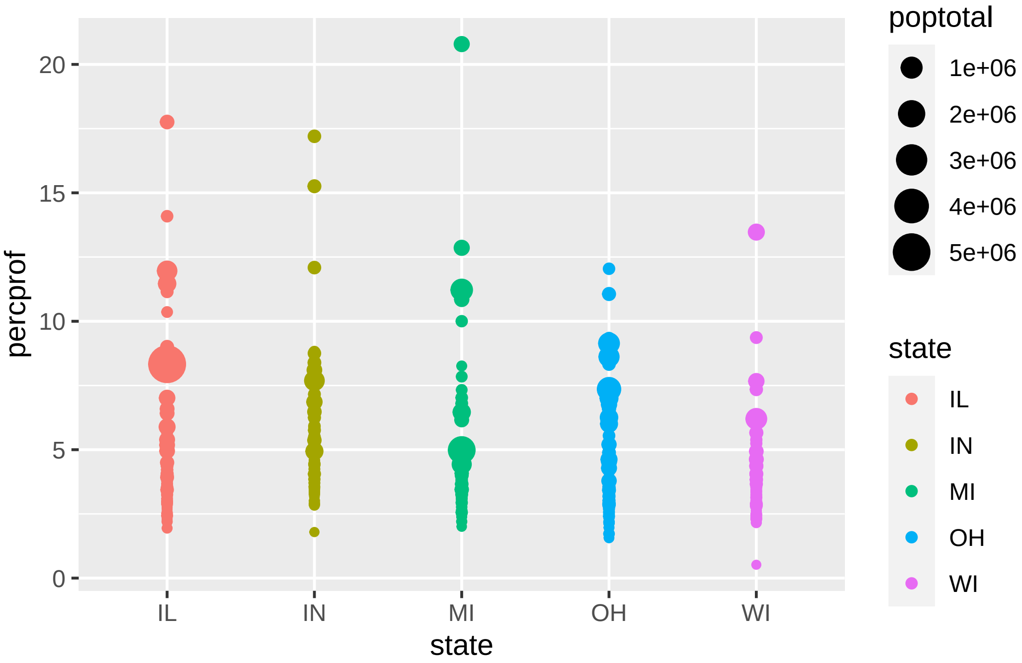
8. Remake "distribution of county-level hs completion rate" with geom_density() instead of geom_histogram().

9. Add a vertical line at the median perchsd using geom_vline. You can calculate the median directly in the ggplot code.

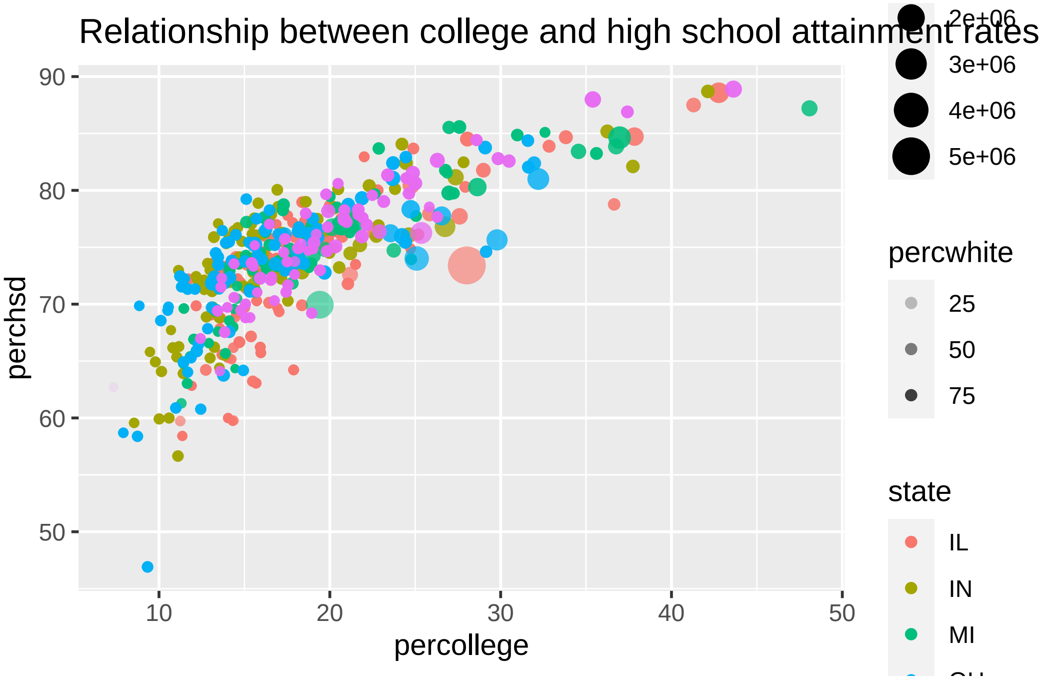
Aesthetics

For the following, write code to reproduce each plot using `midwest`

1. Use `x`, `y`, `color` and `size`.



2. Use `x`, `y`, `color` and `size`.

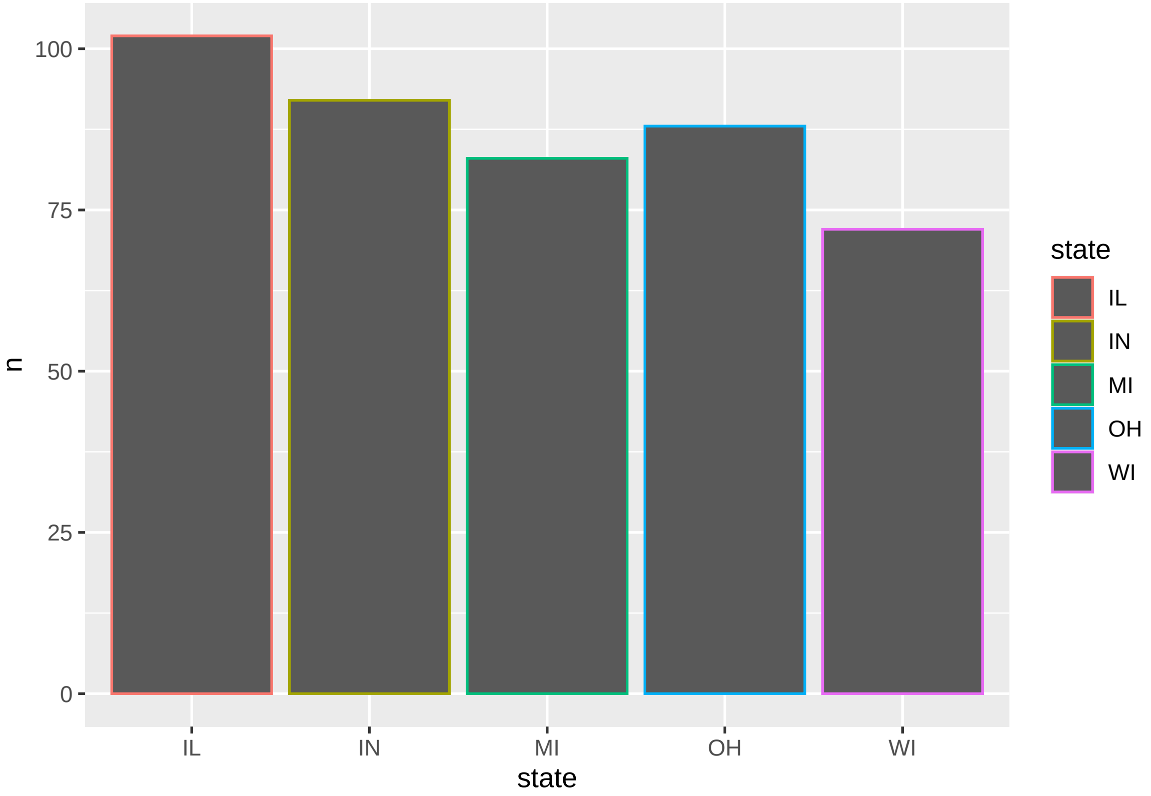


3. Add smooth lines. Get rid of the error around your smooth lines by adding the argument `se = FALSE`.

4. Now try faceting with `facet_grid` and the code `facet_grid(col = vars(inmetro), rows = vars(state))` to your plot.

5. When making bar graphs, `color` only changes the outline of the bar. Change the aesthetic name to `fill` to get the desired result.

```
midwest %>%
  count(state) %>%
  ggplot(aes(x = state, y = n, color = state)) +
  geom_col()
```



6. There's a `geom` called `geom_bar` that takes a dataset and calculates the count. Read the following code and compare it to the `geom_col` code above. Describe how `geom_bar()` is different than `geom_col`.

```
midwest %>%
 ggplot(aes(x = state, color = state)) +
 geom_bar()
```