Lab #6: Data Visualization

Jacob Jameson

It is expected you watch the Module 6 material, here prior to this lab.

In this lab, you will work with `midwest.dta`, which you can download here.

General Guidelines:

You will encounter a few functions we did not cover in the lecture video. 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 too 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 lab questions

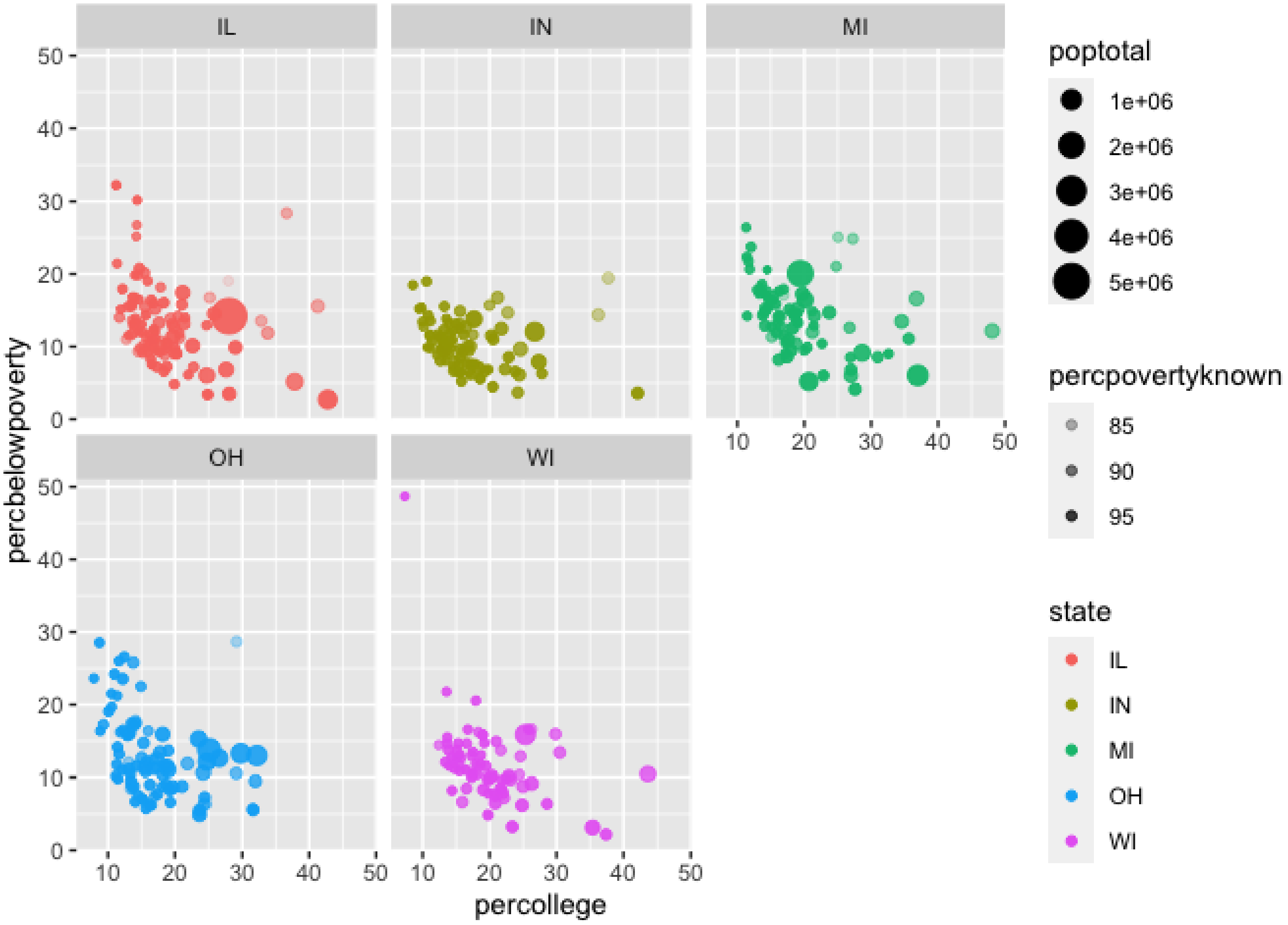
It is highly likely that you will encounter error messages while doing this lab Here are a few steps that might help get you through it.

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

Questions

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))
```

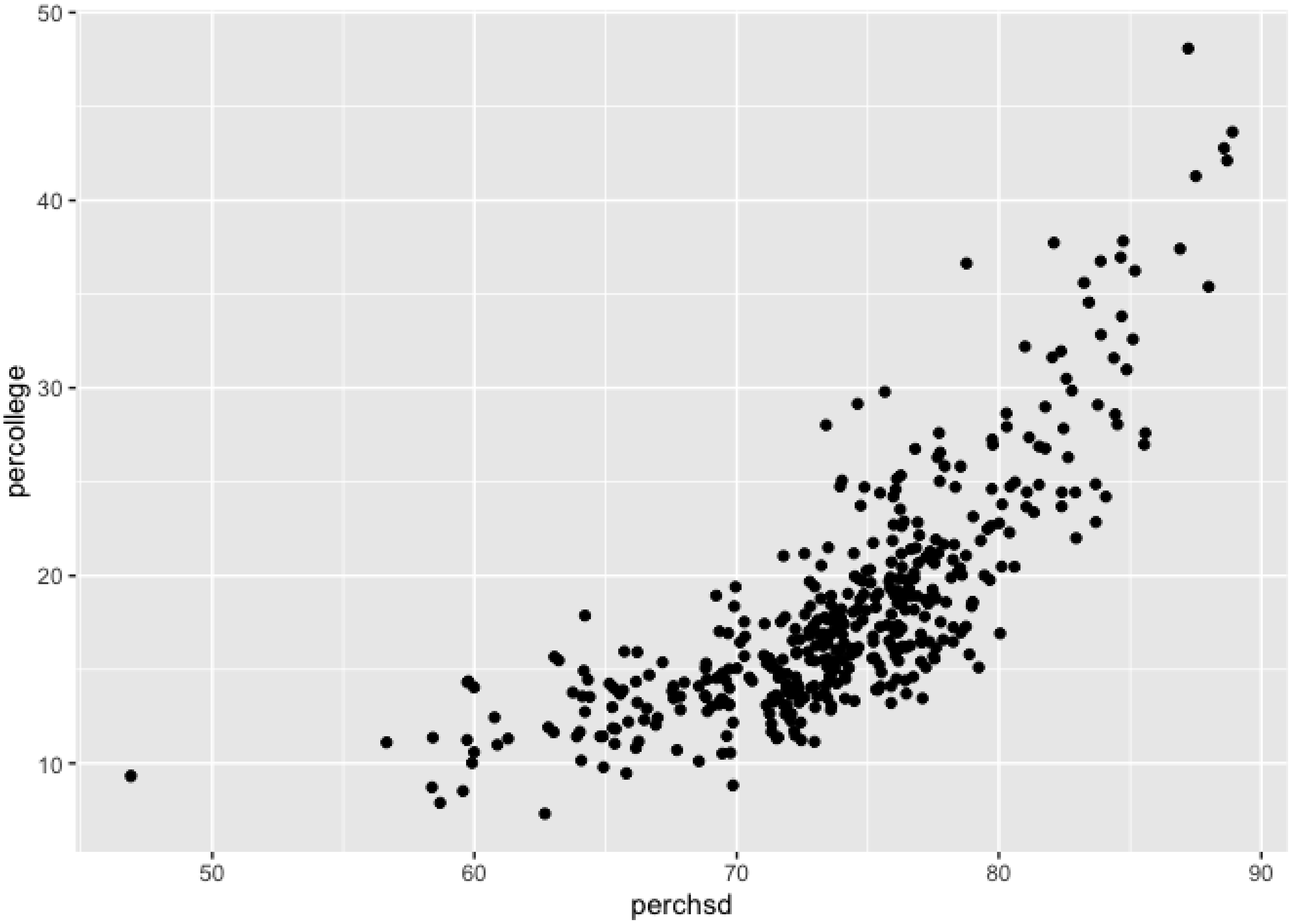


1. Which is more highly correlated with poverty at the county level, college completion rates or high school completion rates? Is it consistent across states? Change one line of code in the above graph.

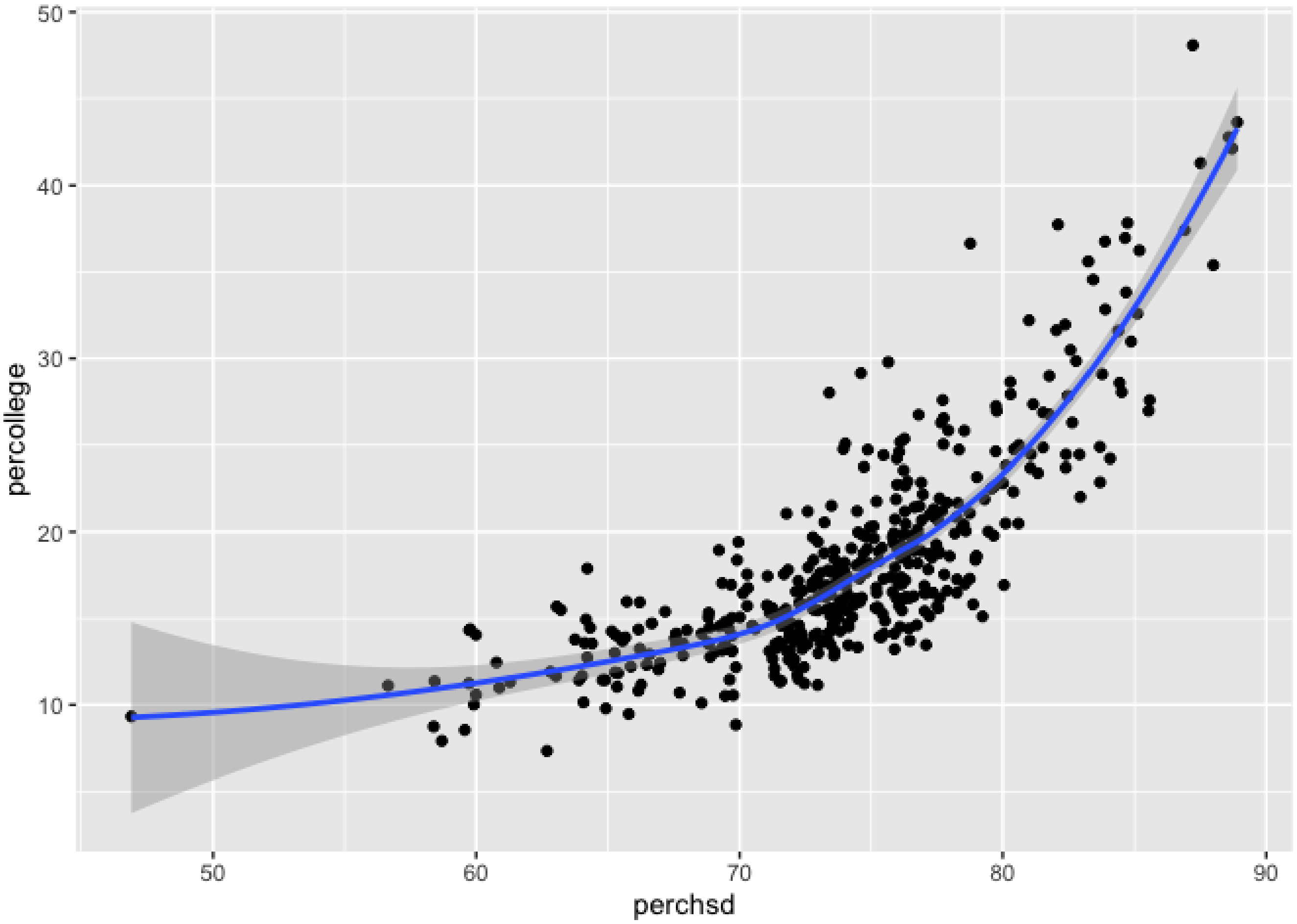
geoms

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

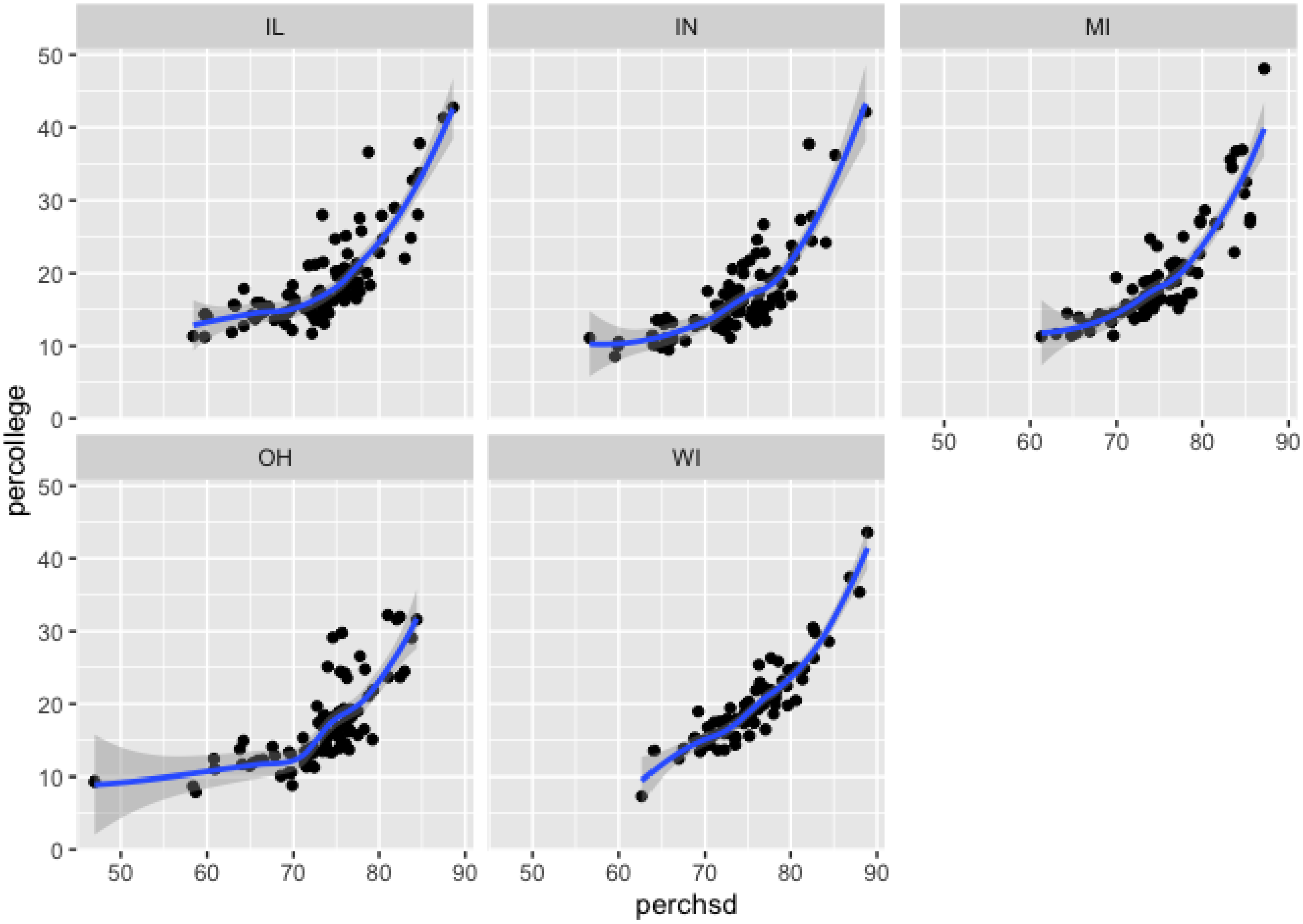
1.



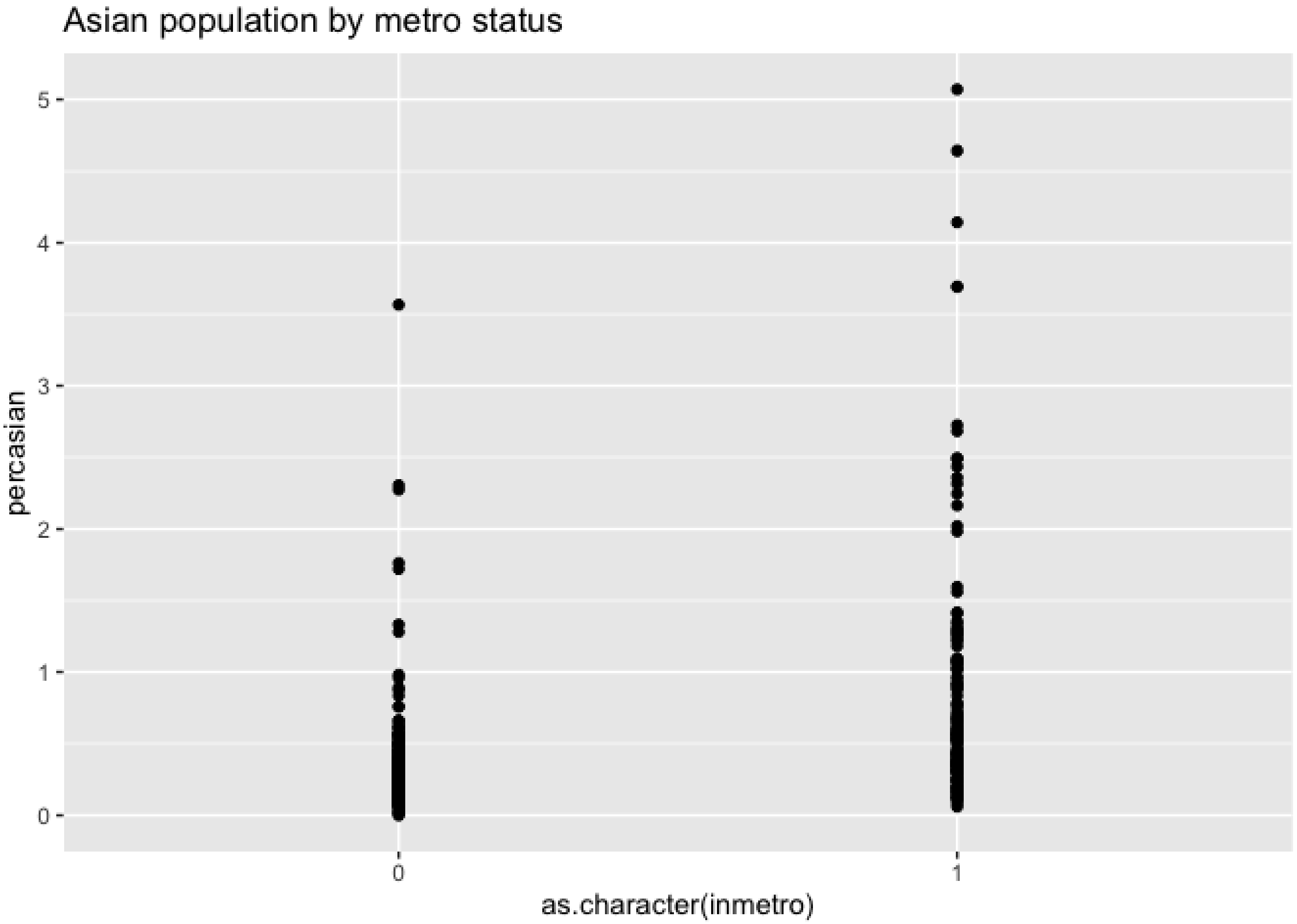
2.



3.



4. Notice here `inmetro` is numeric, but I want it to behave like a discrete variable so I use `x = as.character(inmetro)`. Use `labs(title = "Asian population by metro status")` to create the title.



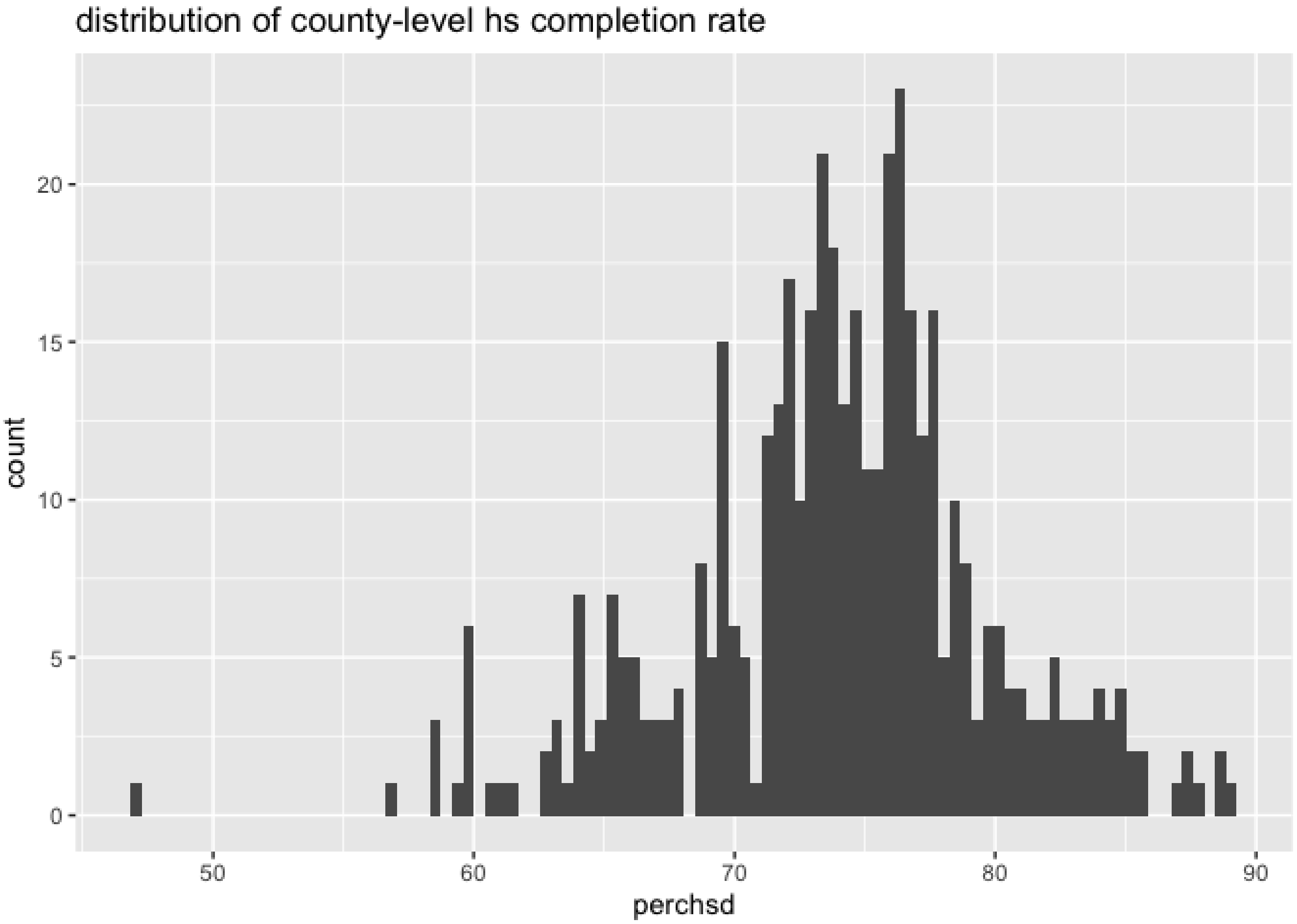
5. Use `geom_boxplot()` instead of `geom_point()` for **“Asian population by metro status”**

6. Use `geom_jitter()` instead of `geom_point()` for **“Asian population by metro status”**

7. Use `geom_jitter()` and `geom_boxplot()` at the same time for **“Asian population by metro status”**. Does order matter?

8. 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")
```



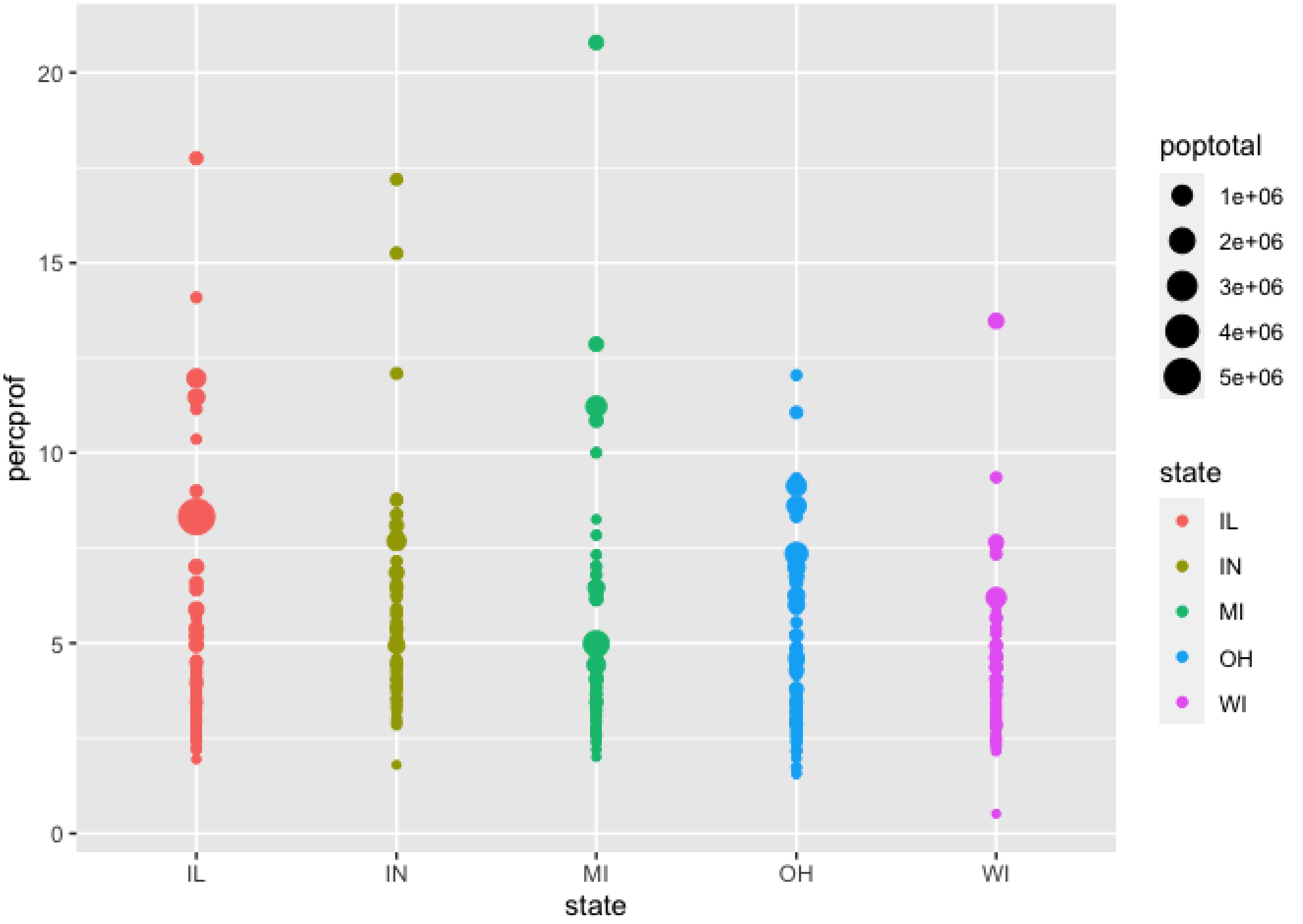
9. Remake **"distribution of county-level hs completion rate"** with `geom_density()` instead of `geom_histogram()`.

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

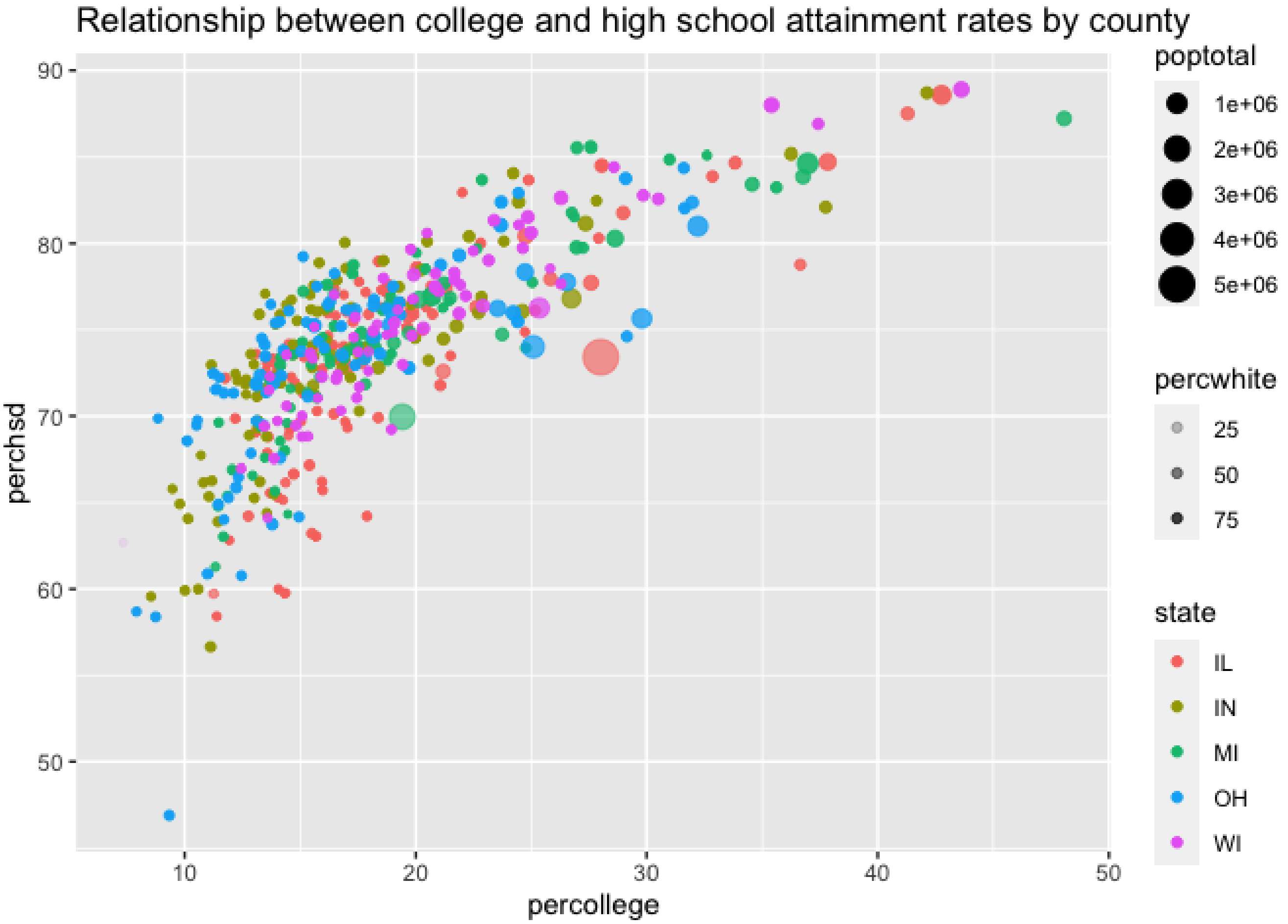
Aesthetics

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`

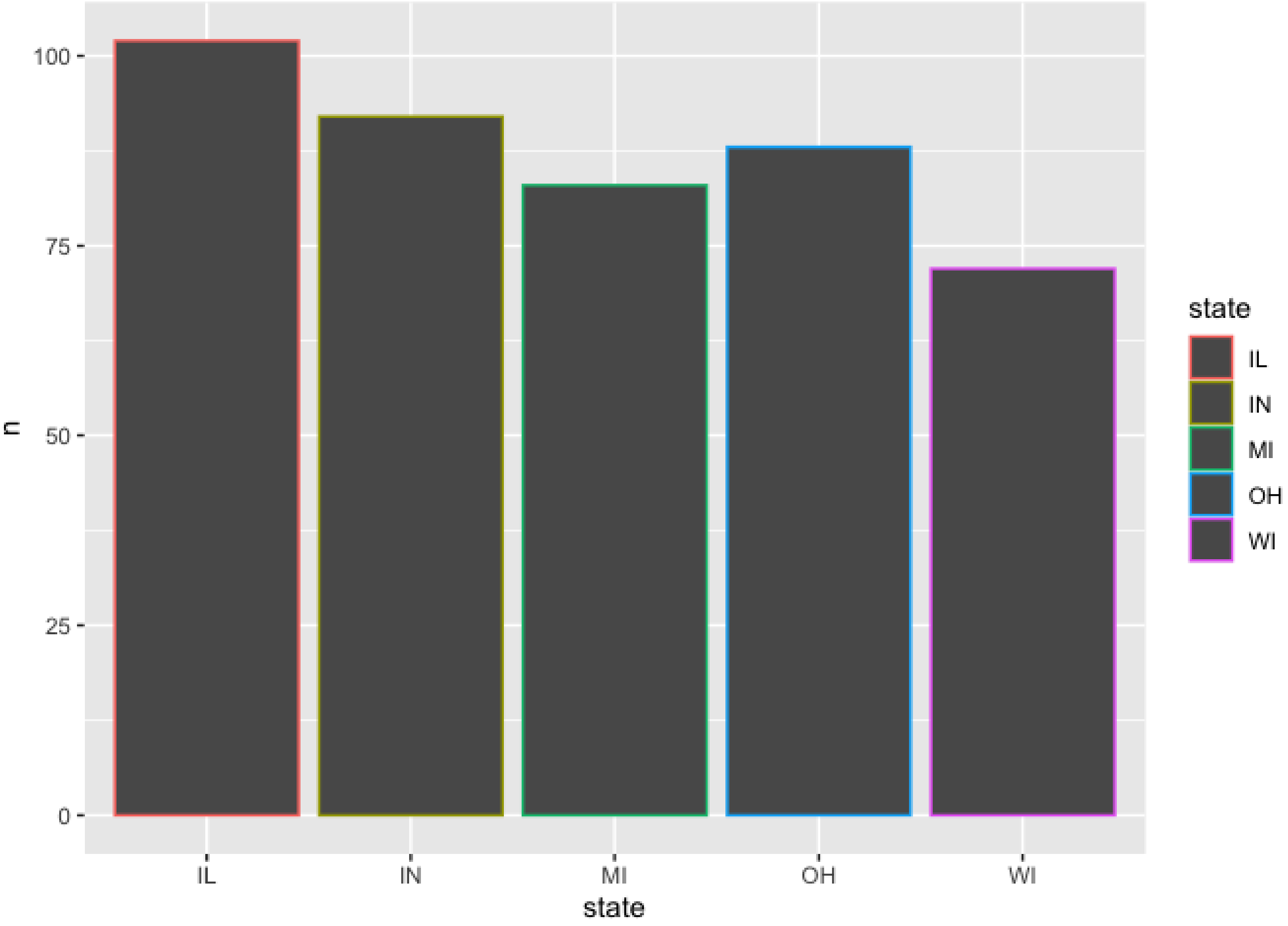


3. Add smooth lines. Get rid of the error around your smooth lines by adding the argument `se = FALSE`.

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

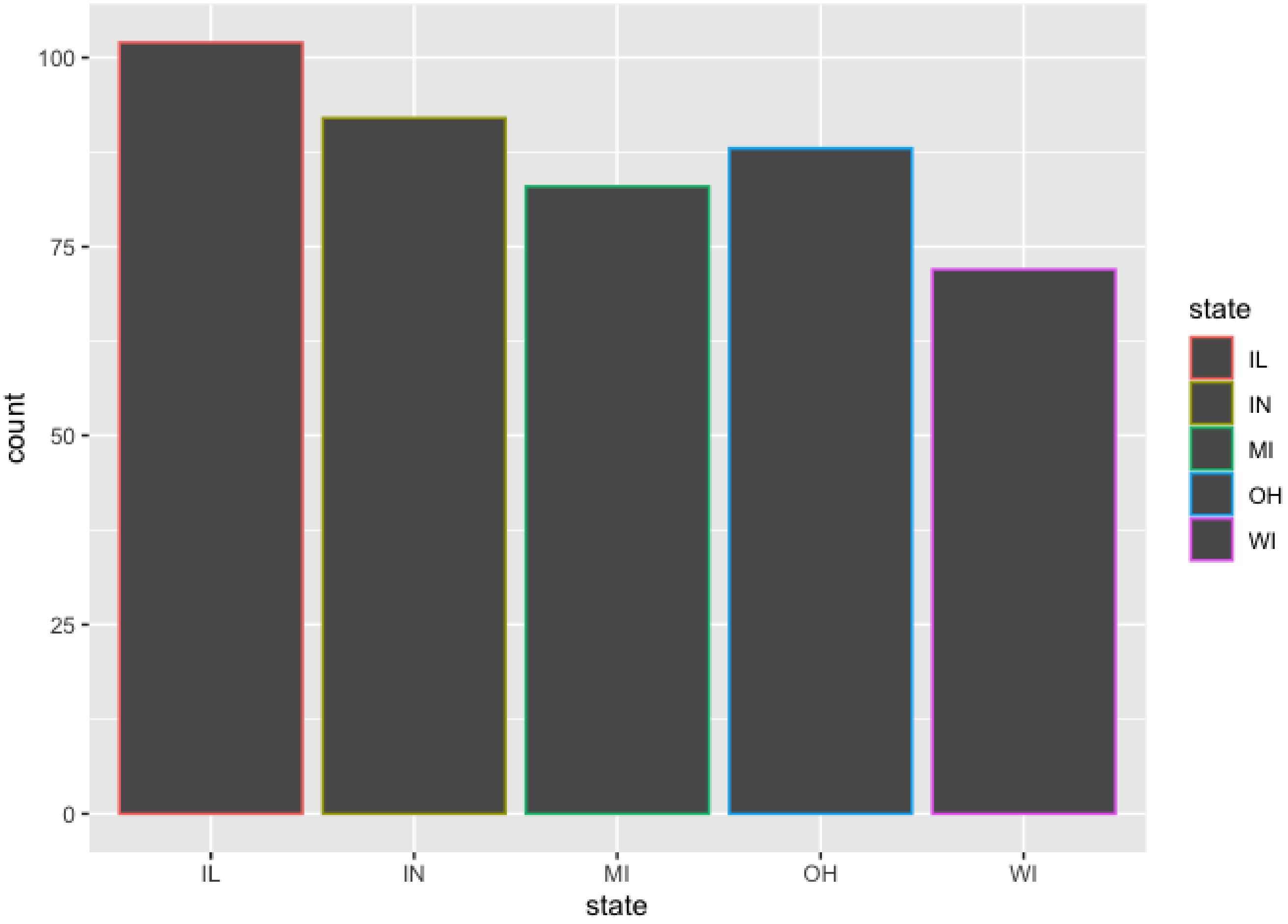
5. When making bar graphs, `color` only changes the outline of the bar. Change the aestethic name to fill to get the desired result

```
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()
```



Well done! You’ve learned how to work with R to create some awesome looking visuals!

Want to improve this tutorial? Report any suggestions/bugs/improvements on here! We’re interested in learning from you how we can make this tutorial better.