Lab 8 Solved: Data Visualization – Tidyverse

Harris Coding Camp

Summer 2023

Data and background

```
library(tidyverse)
library(haven)
recent_college_grads <- haven::read_dta("../data/recent_college_grads.dta")
```

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.

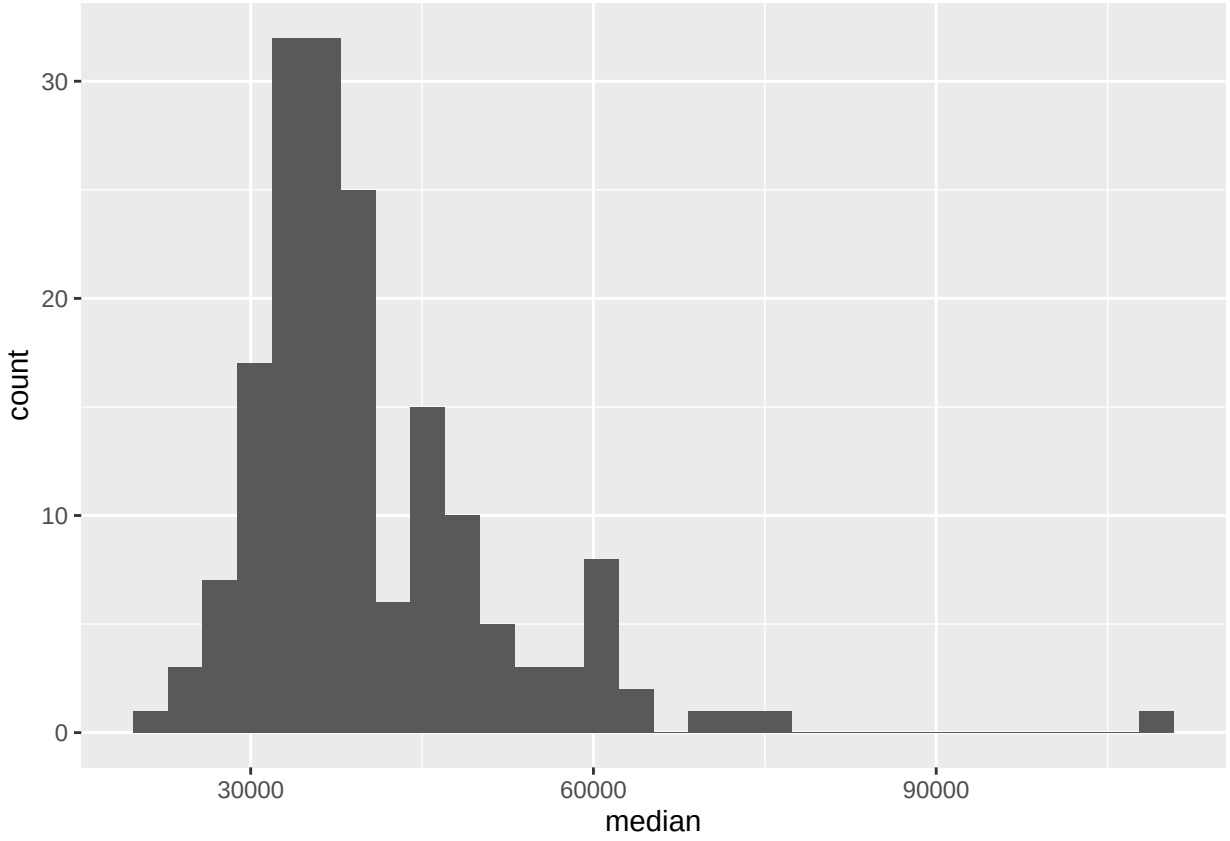
1. Let's first take a look at the distribution of all median incomes using `geom_histogram`, without considering the major categories.

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

SOLUTION:

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

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

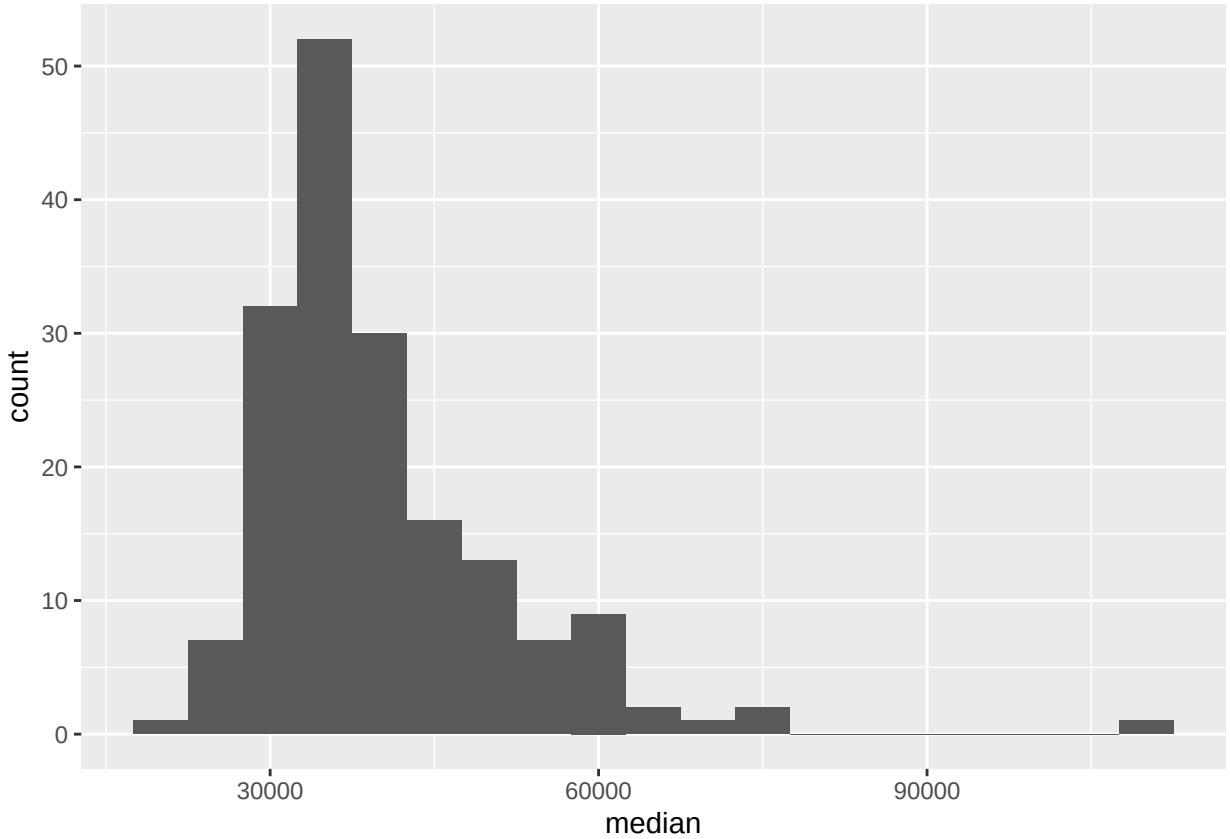


2. Try binwidths of 1000 and 5000 and choose one. Explain your reasoning for your choice.

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

SOLUTION:

```
ggplot(data = college_recent_grads,
       mapping = aes(x = median)) +
  geom_histogram(binwidth = 5000)
```



A bindwidth of 5,000 seems to better aggregate the data at hand. This allow us visualize the shape of the distribution easier.

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.

SOLUTION:

Median and the first and third quartile are useful for describing the distribution as it gives us an idea of the spread of the distribution and a range for which most of our data - 50% of it - lies.

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.

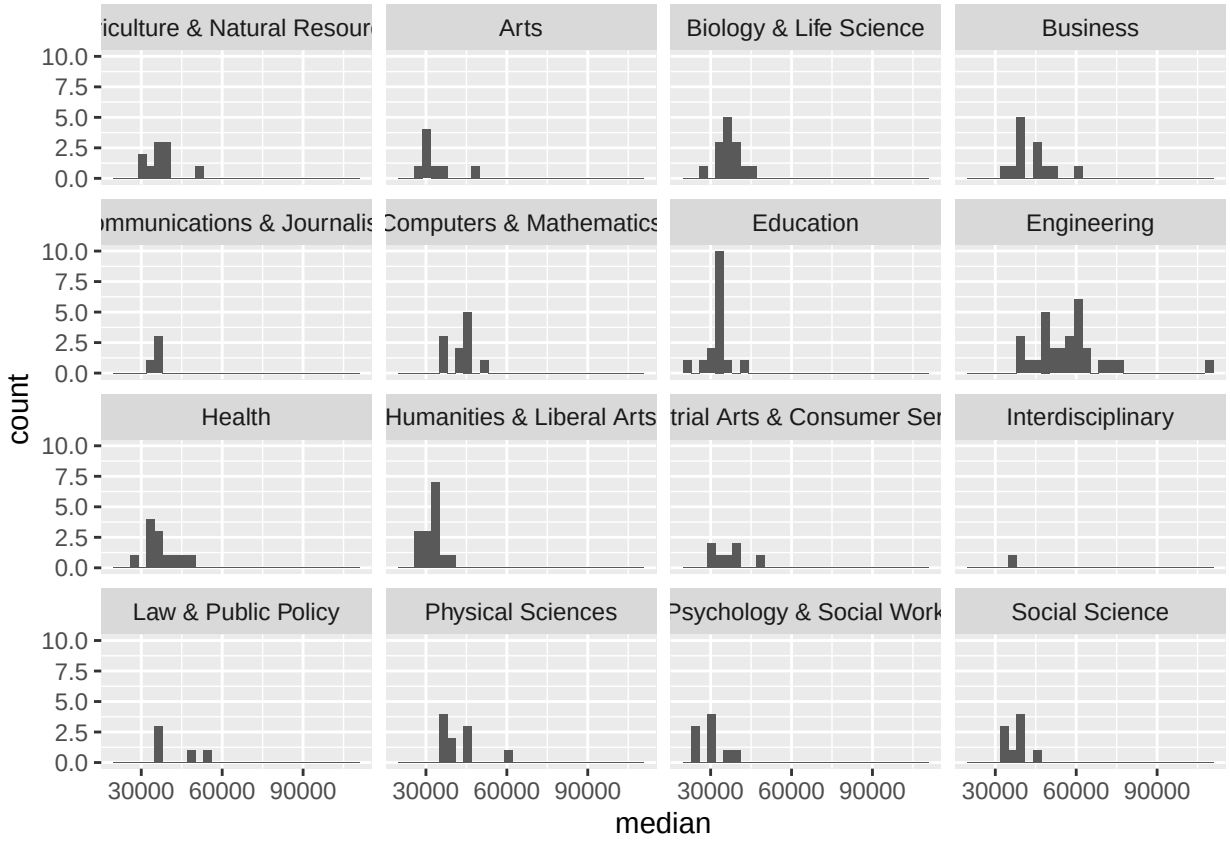
```
ggplot(data = ___, mapping = aes (x=median)) +
  geom_histogram(bindwidth = ___) +
  facet_wrap(.~major_category)
```

SOLUTION:

```
ggplot(data = college_recent_grads,
       mapping = aes(x=median)) +
  geom_histogram(bindwidth = 5000) +
  facet_wrap(.~major_category)
```

```
## Warning: Ignoring unknown parameters: bindwidth
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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.

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

SOLUTION:

```
college_recent_grads %>%
  filter(median == max(median) |
         median == min(median) |
         median == median(median)) %>%
  select(major, median)
```

```
## # A tibble: 8 x 2
##   major                                    median
##   <chr>                                     <dbl>
## 1 Petroleum Engineering                    110000
## 2 Human Resources And Personnel Management  36000
## 3 Pre-Law And Legal Studies                 36000
## 4 Miscellaneous Health Medical Professions  36000
## 5 Public Administration                     36000
## 6 Geosciences                               36000
## 7 Social Psychology                         36000
## 8 Library Science                           22000
```

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.

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

SOLUTION:

```
college_recent_grads %>%
  count(major_category) %>%
  arrange(desc(n))
```

```
## # A tibble: 16 x 2
##    major_category                        n
##    <chr>                             <int>
##  1 Engineering                          29
##  2 Education                            16
##  3 Humanities & Liberal Arts            15
##  4 Biology & Life Science               14
##  5 Business                             13
##  6 Health                               12
##  7 Computers & Mathematics              11
##  8 Agriculture & Natural Resources      10
```

```
##  9 Physical Sciences                       10
## 10 Psychology & Social Work                 9
## 11 Social Science                           9
## 12 Arts                                     8
## 13 Industrial Arts & Consumer Services      7
## 14 Law & Public Policy                      5
## 15 Communications & Journalism              4
## 16 Interdisciplinary                        1
```

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.

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

7. Then, we can use this to create a new variable in our data frame indicating whether a major is STEM or not. Complete the code.

```
college_recent_grads <- college_recent_grads %>%
  mutate(major_type = ifelse(...))
```

SOLUTION:

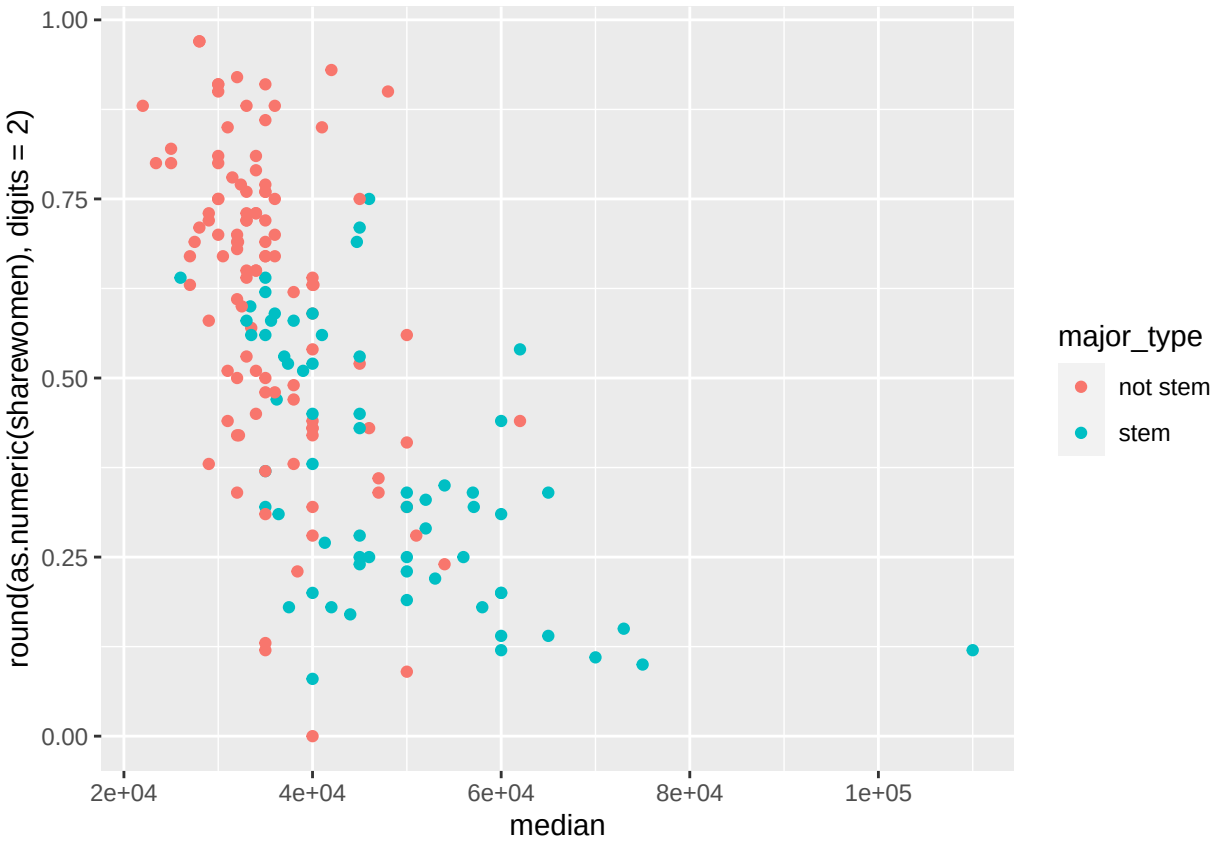
```
college_recent_grads <- college_recent_grads %>%
  mutate(major_type = ifelse(major_category %in%
                               stem_categories, "stem", "not stem"))
```

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.

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

SOLUTION:

```
ggplot(data = college_recent_grads,
       aes(x = median,
           y = round(as.numeric(sharewomen), digits = 2),
           color = major_type)) +
  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.

SOLUTION:

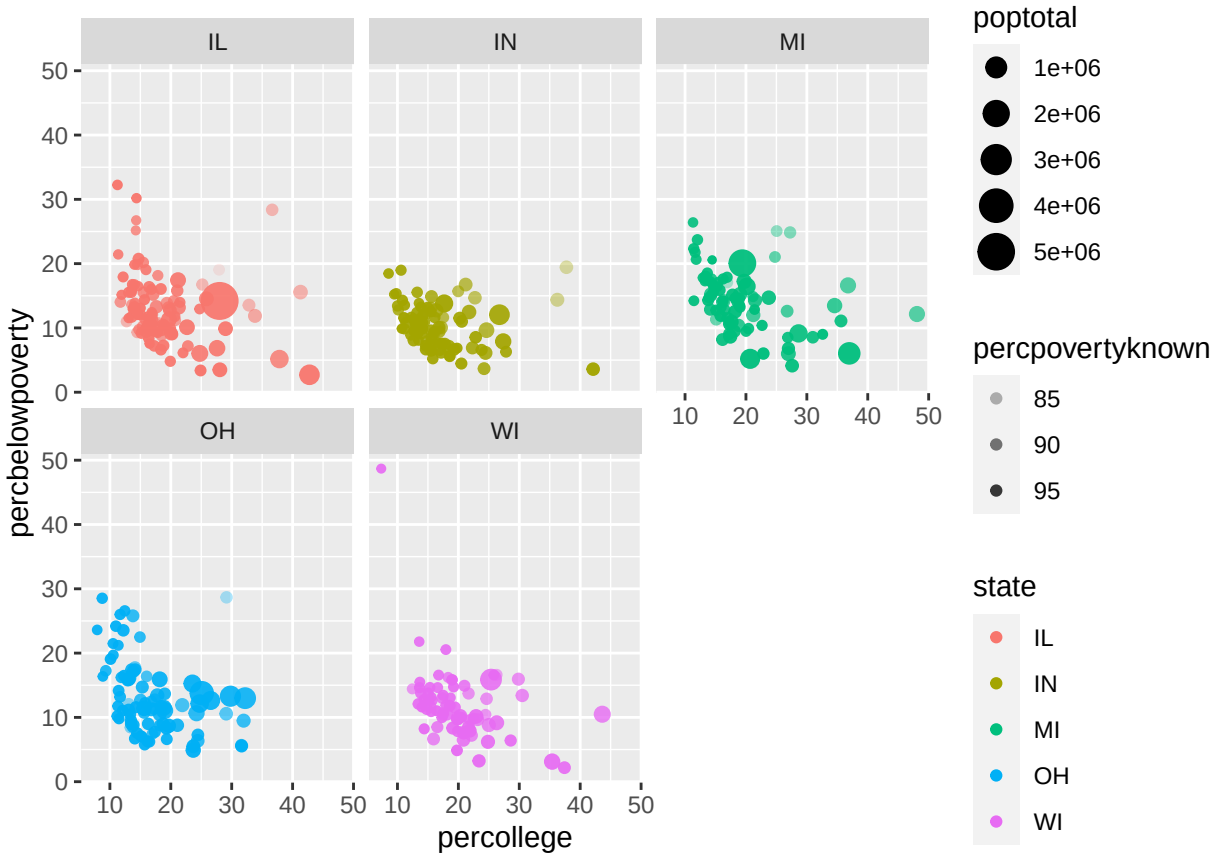
```
college_recent_grads %>%
  filter(major_type == 'stem', median < 36000) %>%
  arrange(median) %>%
  select(major, median, p25th, p75th)
```

```
## # A tibble: 10 x 4
##    major                                median p25th p75th
##    <chr>                                 <dbl> <dbl> <dbl>
##  1 Zoology                               26000 20000 39000
##  2 Ecology                               33000 23000 42000
##  3 Biology                               33400 24000 45000
##  4 Miscellaneous Biology                 33500 23000 48000
##  5 Multi-Disciplinary Or General Science 35000 24000 50000
##  6 Physiology                            35000 20000 50000
##  7 Communication Technologies            35000 25000 45000
##  8 Neuroscience                          35000 30000 44000
##  9 Atmospheric Sciences And Meteorology  35000 28000 50000
## 10 Environmental Science                 35600 25000 40200
```

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))
```

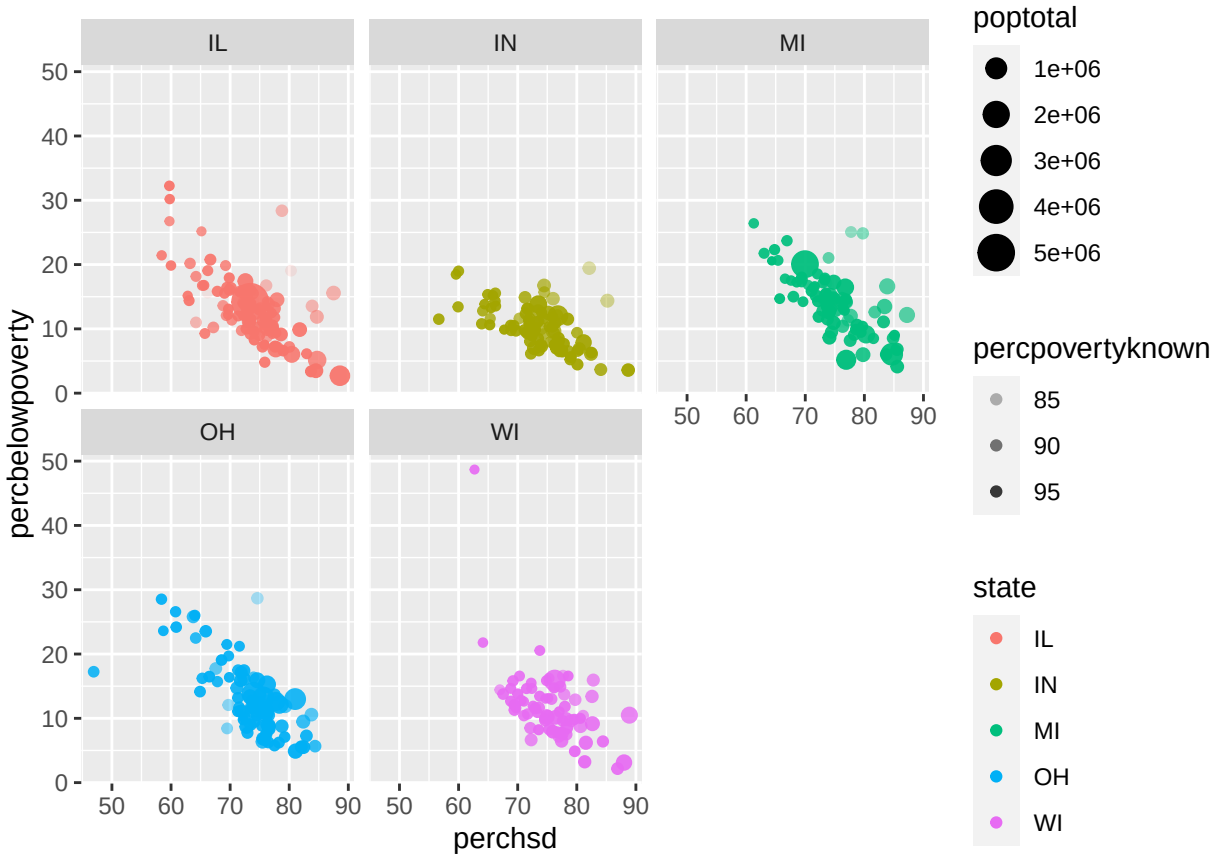


1. Which is more highly correlated with poverty at the county level, college completion rates or high school completion rates? Is it consistent across states? Change one line of code in the above graph.

SOLUTION:

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

```
geom_point() +
facet_wrap(vars(state))
```



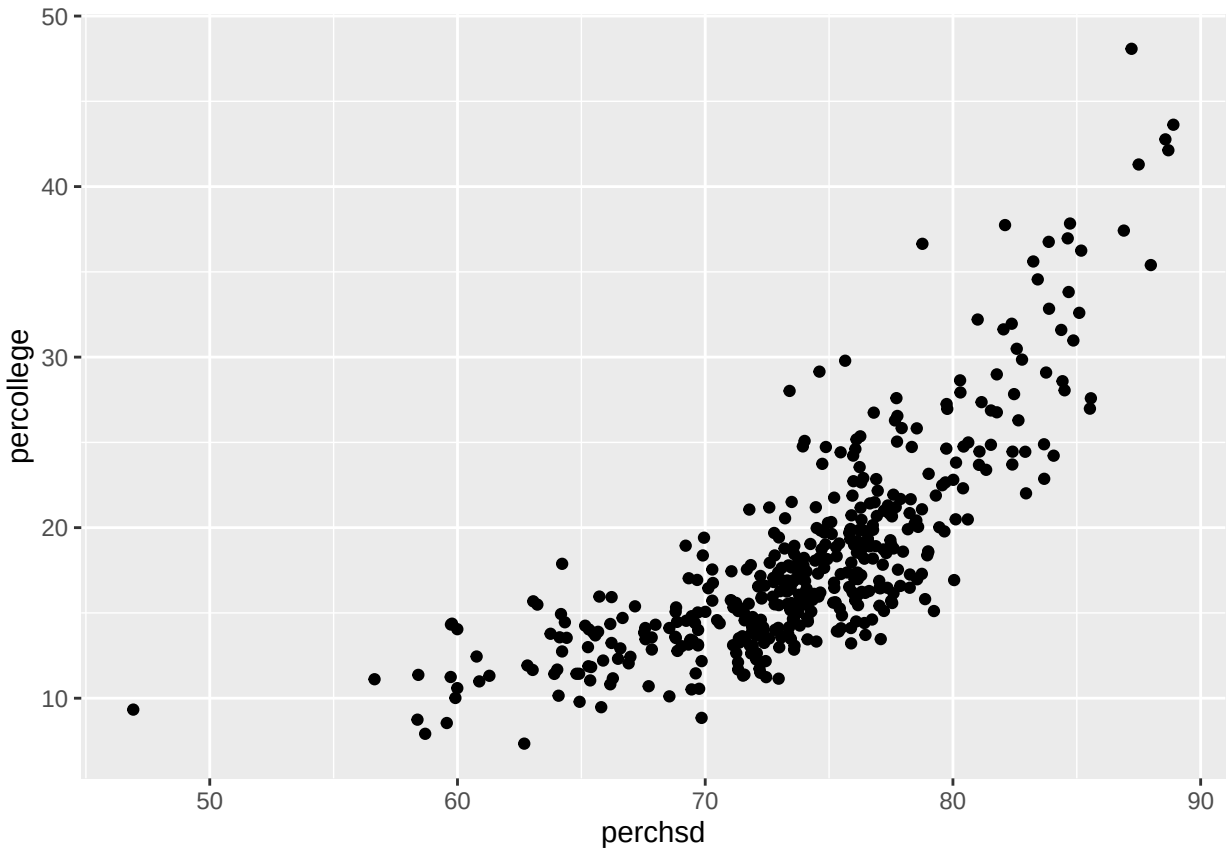
It appears that high school degree attainment is more strongly correlated with poverty rates at the county level.

geoms

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

1. SOLUTION:

```
midwest %>%
  ggplot(aes(x = perchsd, y = percollege)) +
  geom_point()
```

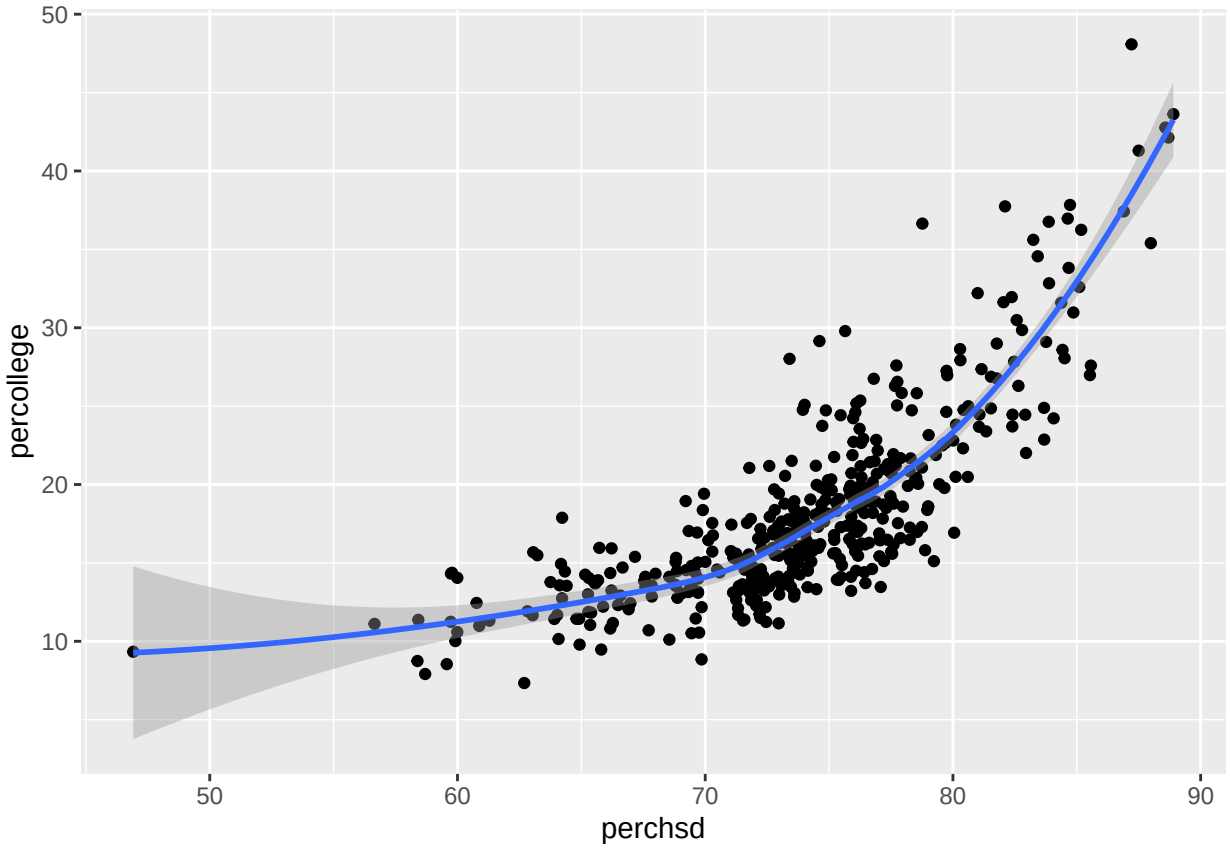


2.

SOLUTION:

```
midwest %>%
  ggplot(aes(x = perchsd, y = percollege)) +
  geom_point() +
  geom_smooth()
```

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

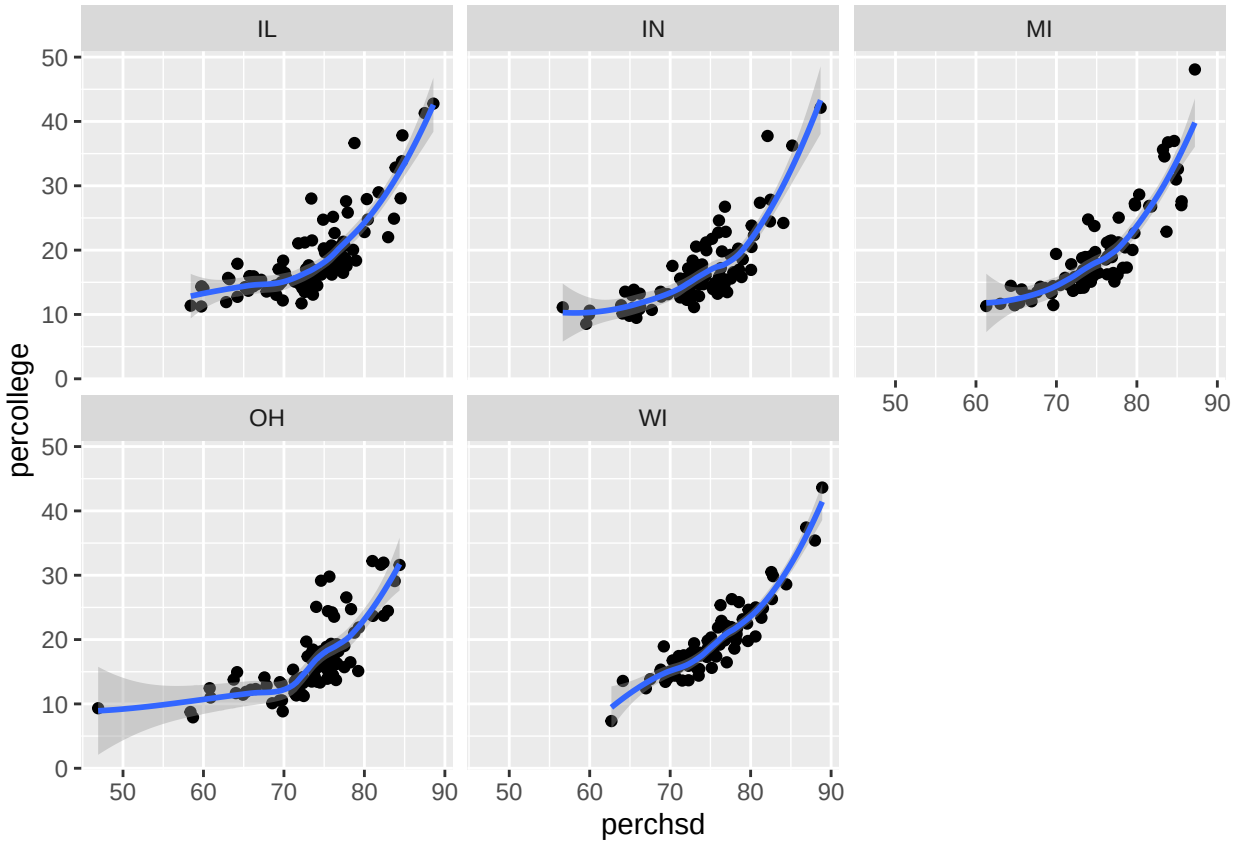


3

SOLUTION:

```
midwest %>%
  ggplot(aes(x = perchsd, y = percollege)) +
  geom_point() +
  geom_smooth() +
  facet_wrap(vars(state))
```

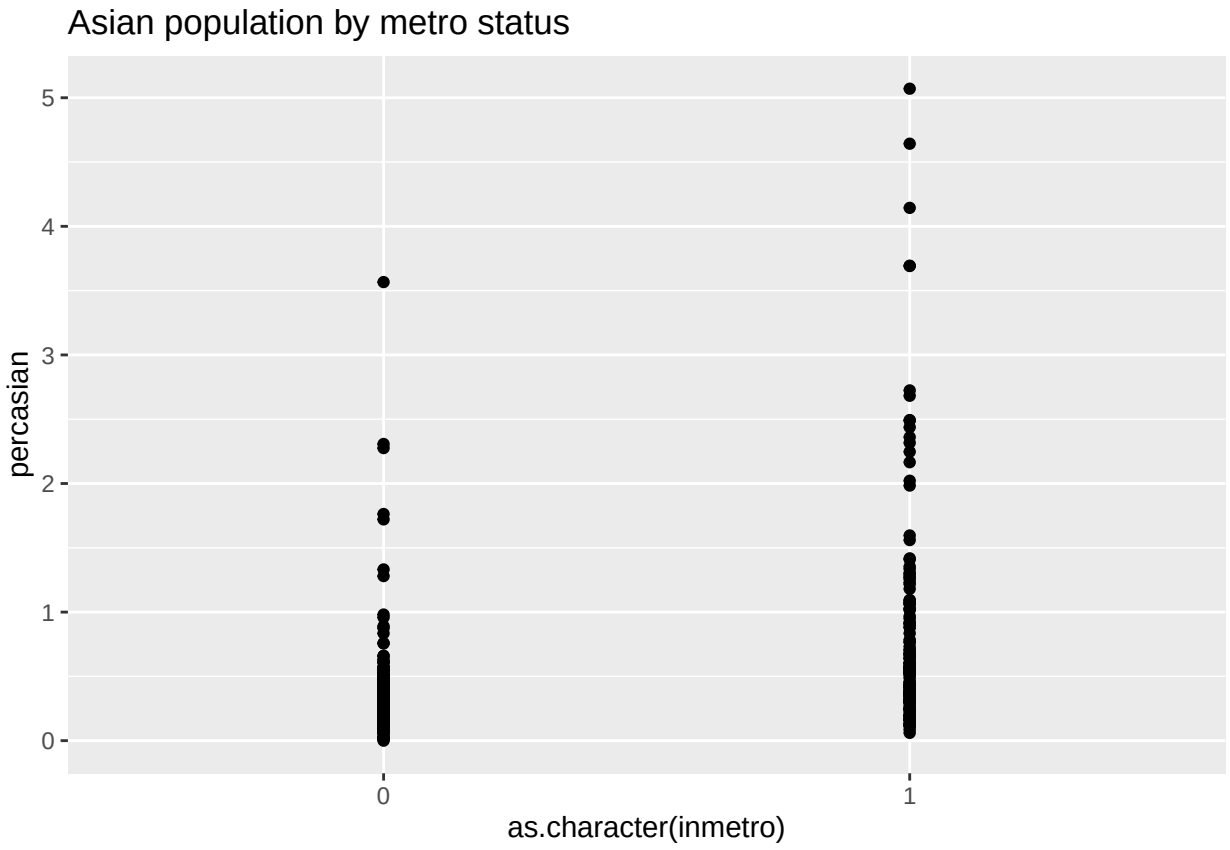
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



4.

SOLUTION:

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

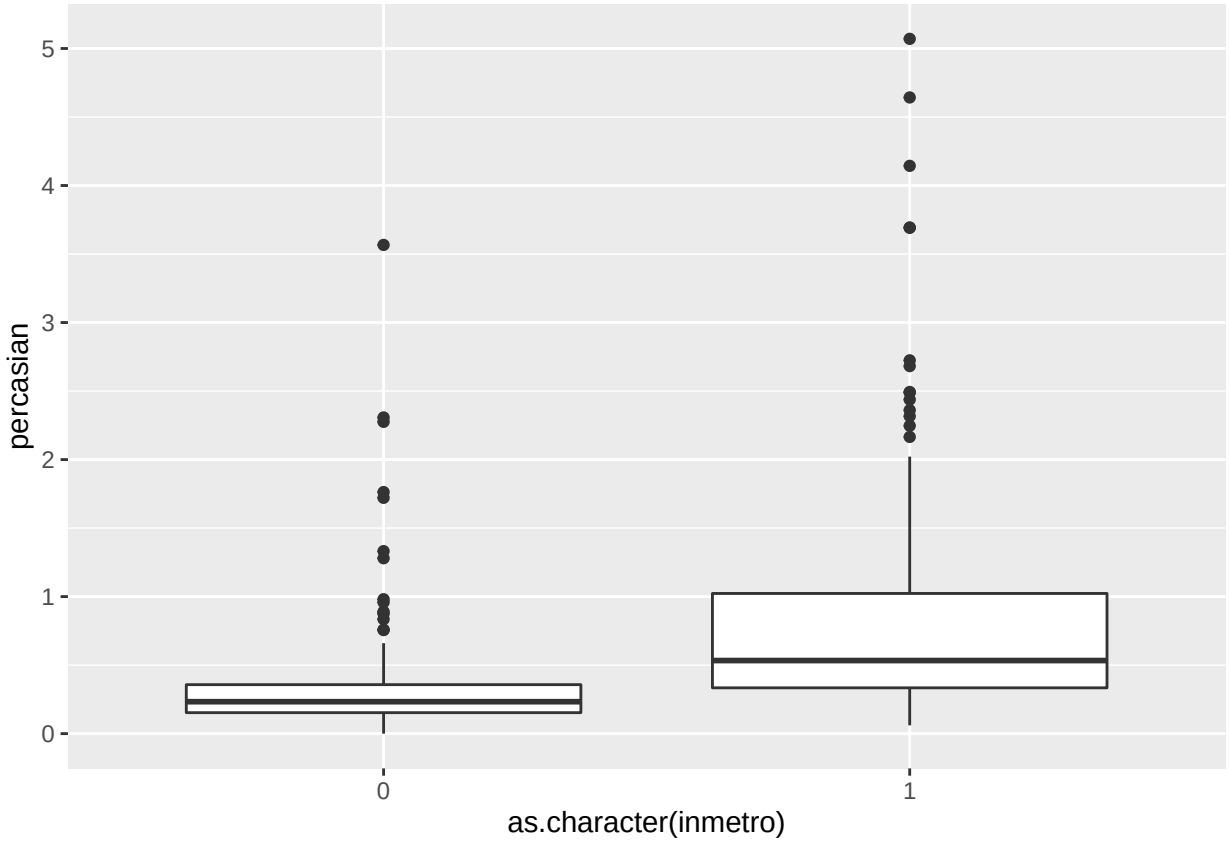


Notice that `inmetro` is numeric, but I want it to behave like a discrete variable so I use `x = as.character(inmetro)`. Complete the code above for part 4.

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

SOLUTION:

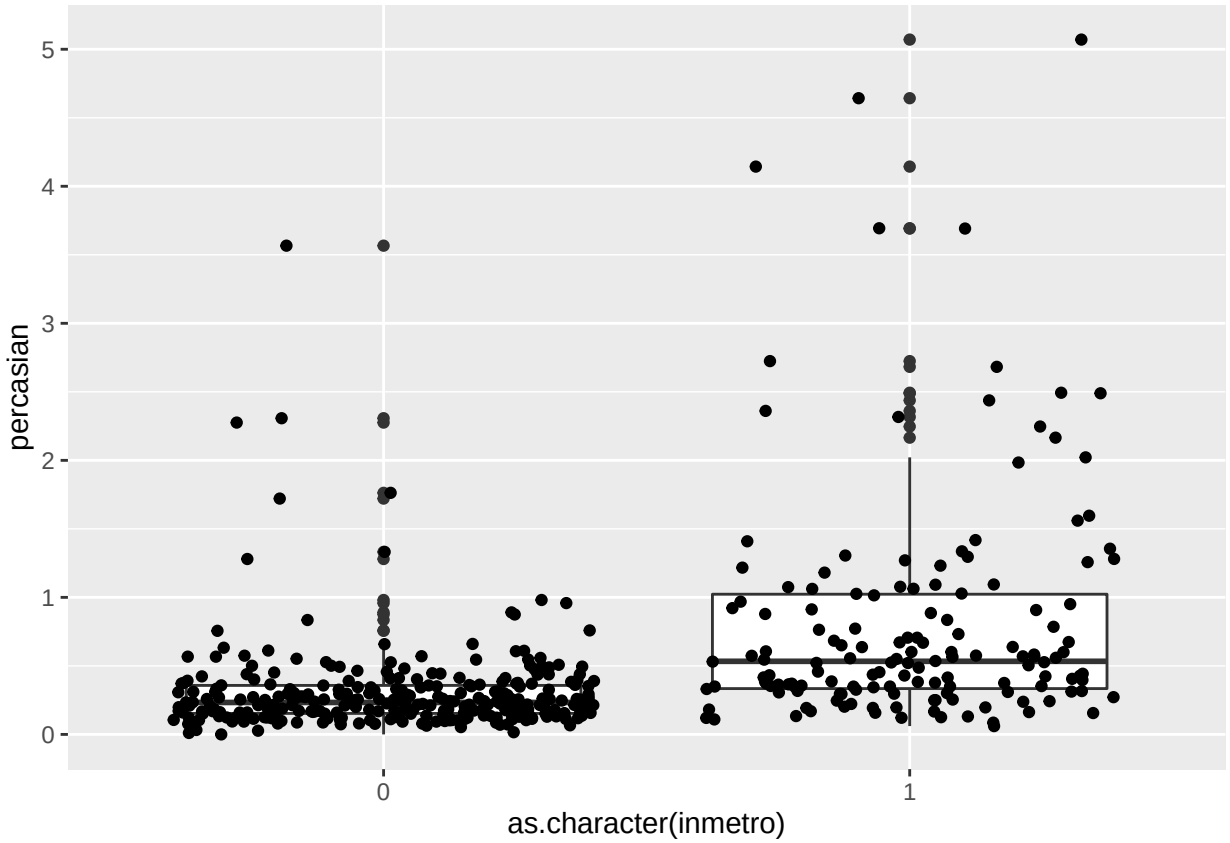
```
midwest %>%
  ggplot(aes(x = as.character(inmetro), y = percasian)) +
  geom_boxplot()
```



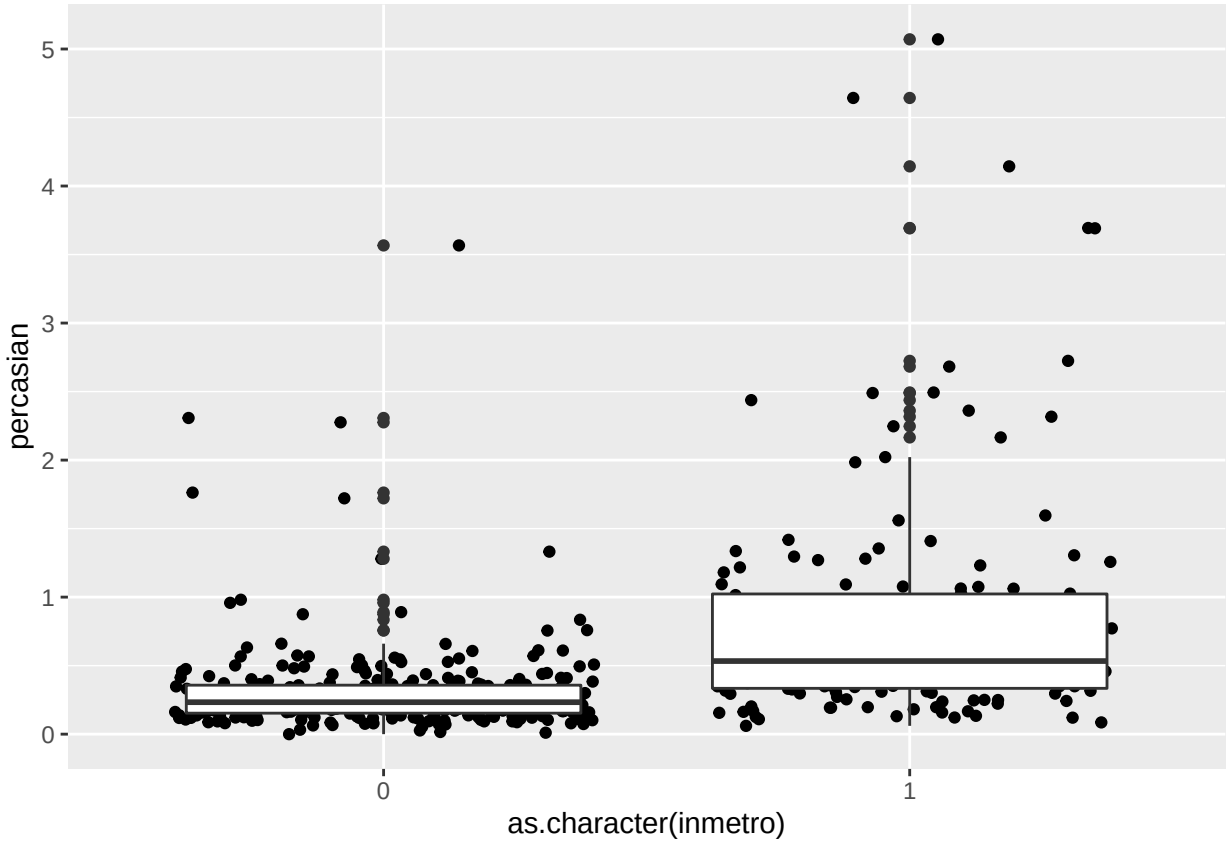
6. Use `geom_jitter()` and `geom_boxplot()` at the same time for "Asian population by metro status". Does order matter?

SOLUTION:

```
midwest %>%
  ggplot(aes(x = as.character(inmetro), y = percasian)) +
  geom_boxplot() +
  geom_jitter()
```

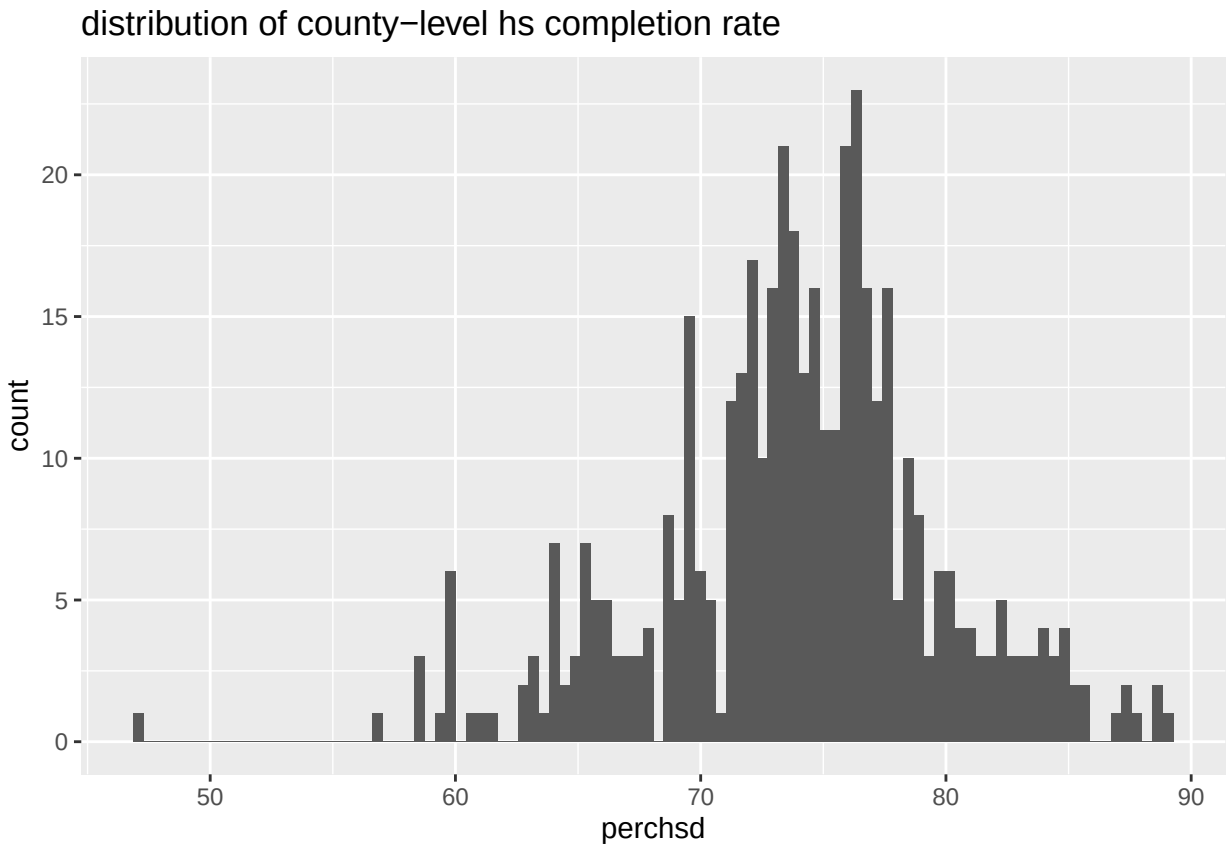



```
midwest %>%
  ggplot(aes(x = as.character(inmetro), y = percasian)) +
  geom_jitter() +
  geom_boxplot()
```


5
4
3
2
1
0
percasian
0
1
as.character(inmetro)

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")
```



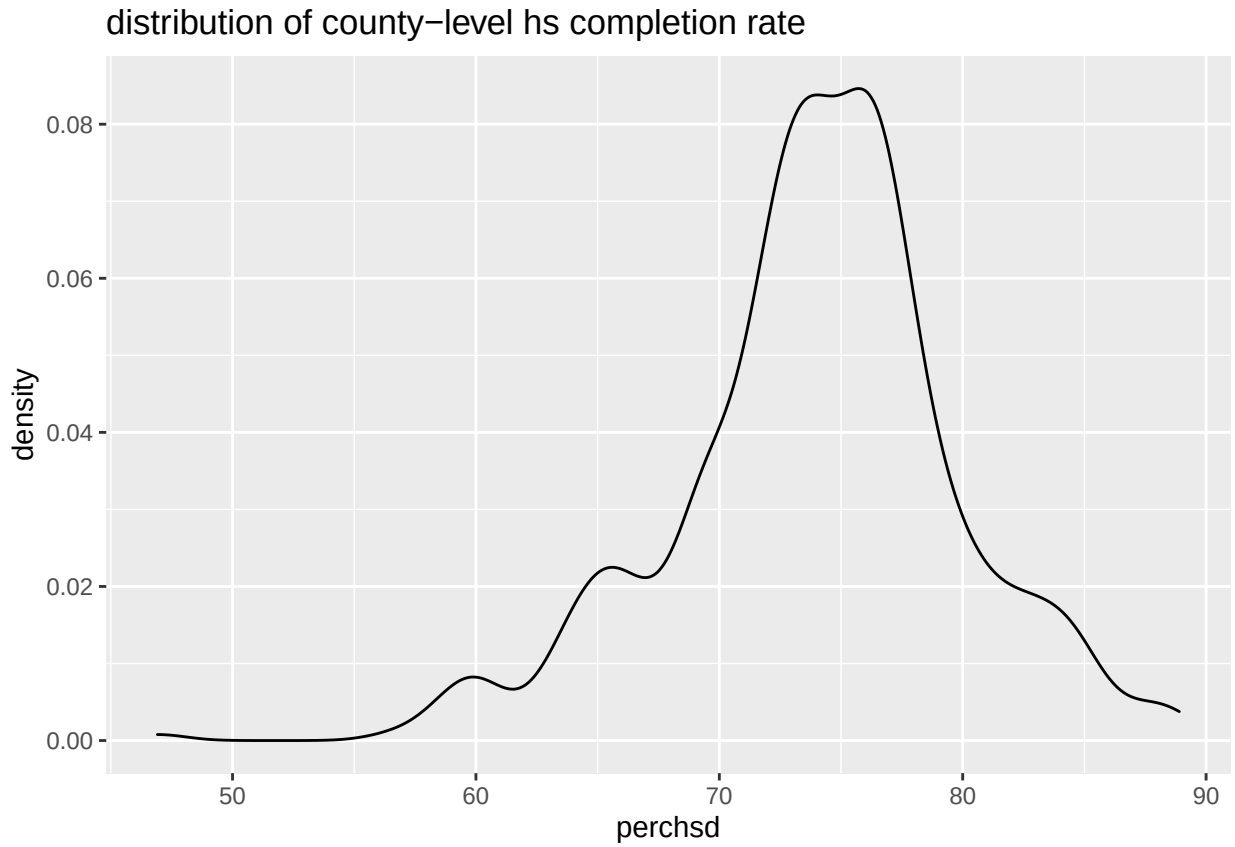
SOLUTION:

`bins` determine the number of bins to divide the data into. E.g. midwest has 437 obs, so if we use 40 bins each bin will contain 437/40 = roughly 11 counties. By default, there are 30 bins and ggplot gives you a warning, because it's an arbitrary default.

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

SOLUTION:

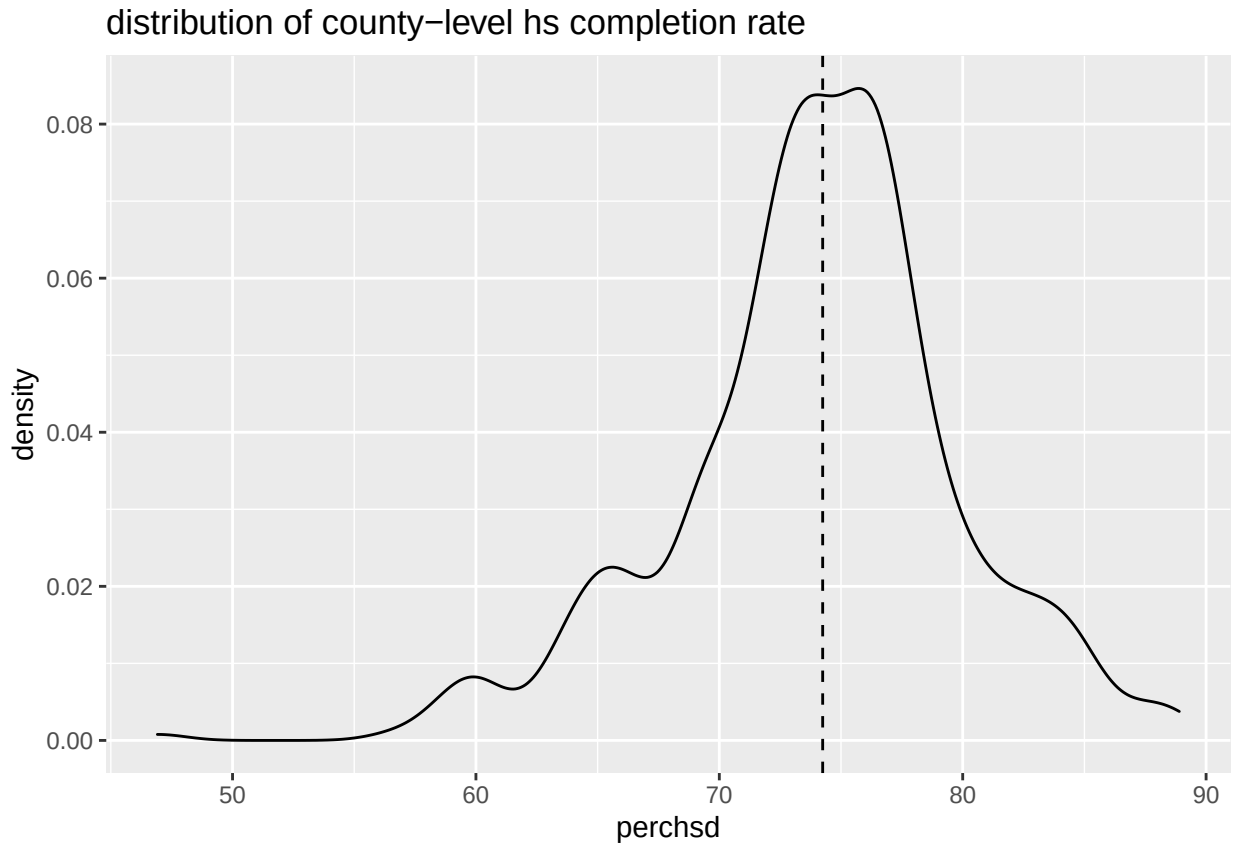
```
midwest %>%
  ggplot(aes(x = perchsd)) +
  geom_density() +
  labs(title = "distribution of county-level hs completion rate")
```



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

SOLUTION:

```
midwest %>%
  ggplot(aes(x = perchsd)) +
  geom_density() +
  geom_vline(aes(xintercept = median(perchsd)), linetype = "dashed") +
  labs(title = "distribution of county-level hs completion rate")
```



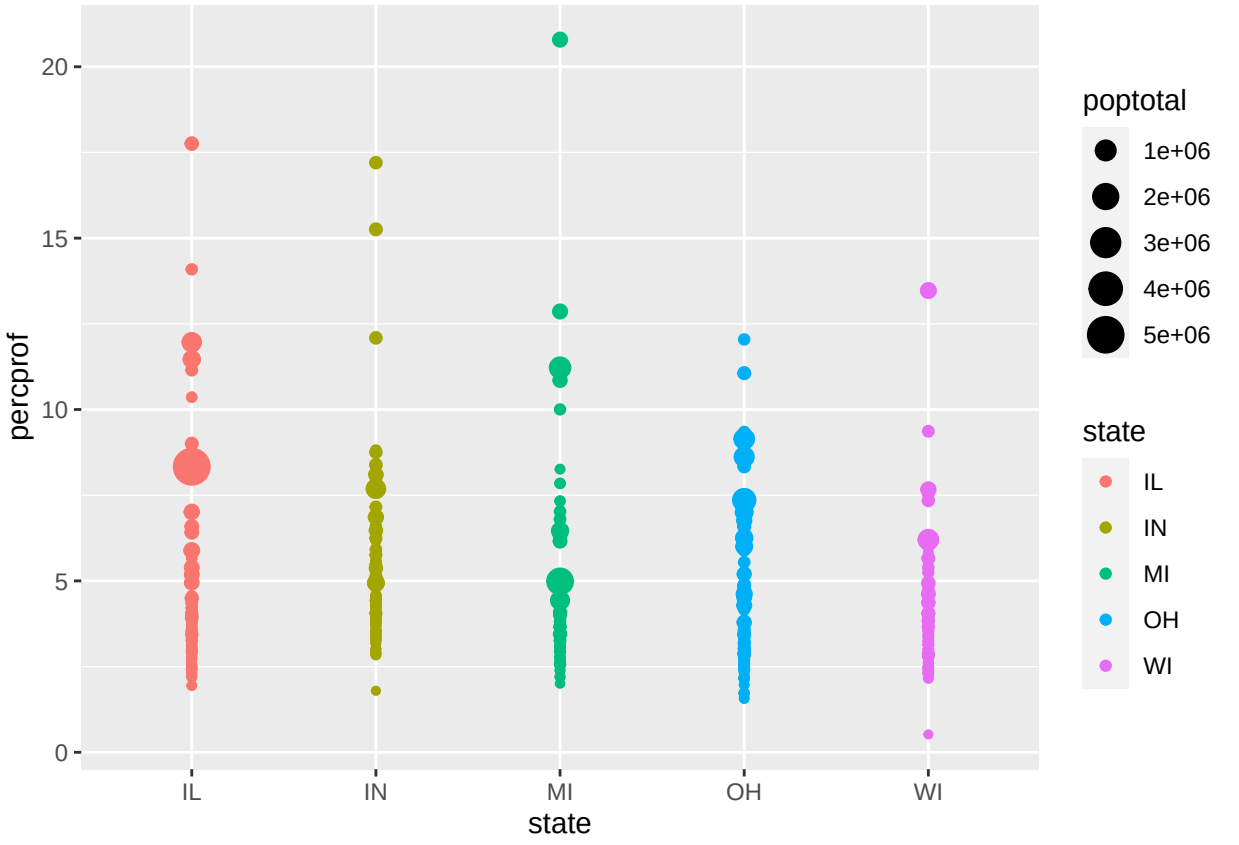
Aesthetics

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

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

SOLUTION:

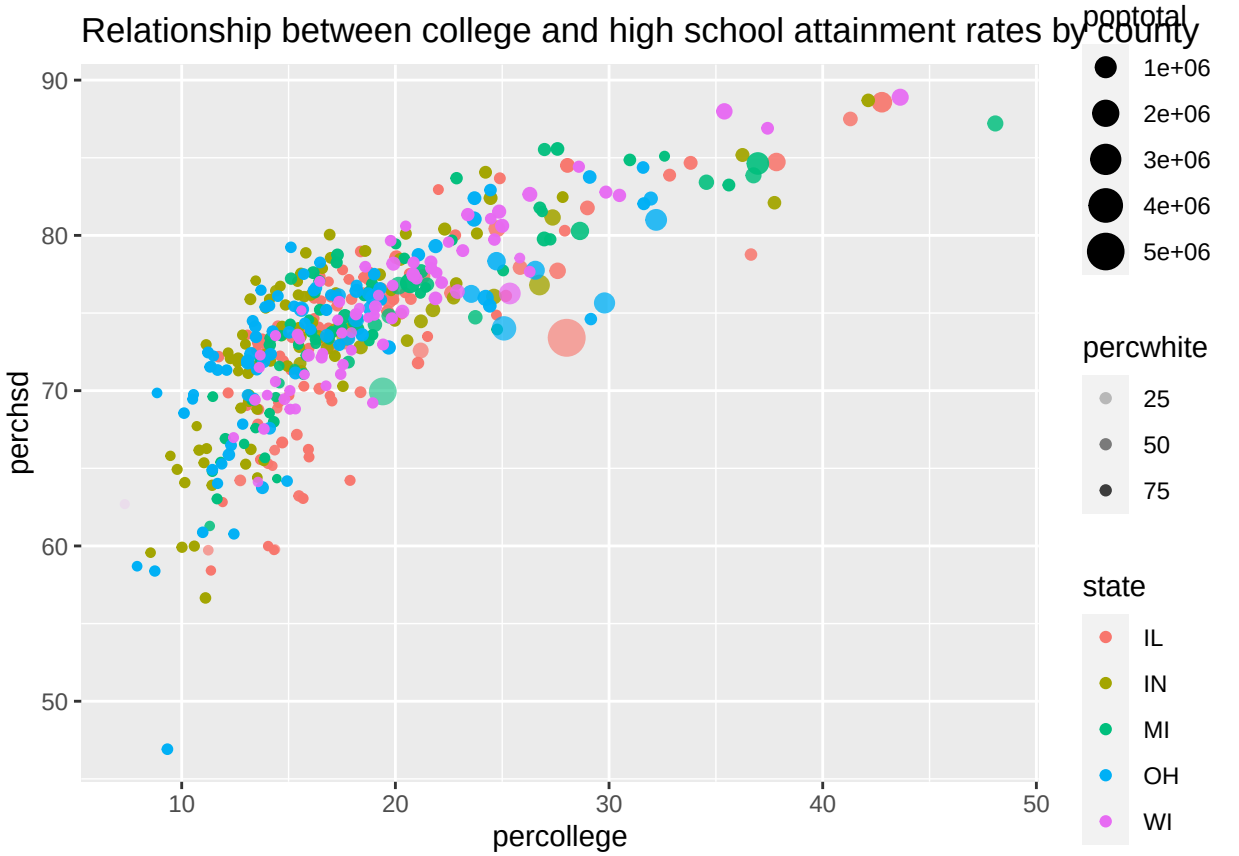
```
midwest %>%
  ggplot(aes(x = state, y = percprof, color = state, size = poptotal)) +
  geom_point()
```



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

SOLUTION:

```
midwest %>%
  ggplot(aes(x = percollege,
             y = perchsd,
             color = state,
             size = poptotal,
             alpha = percwhite)) +
  geom_point() +
  labs(title = "Relationship between college and high school attainment rates by county")
```

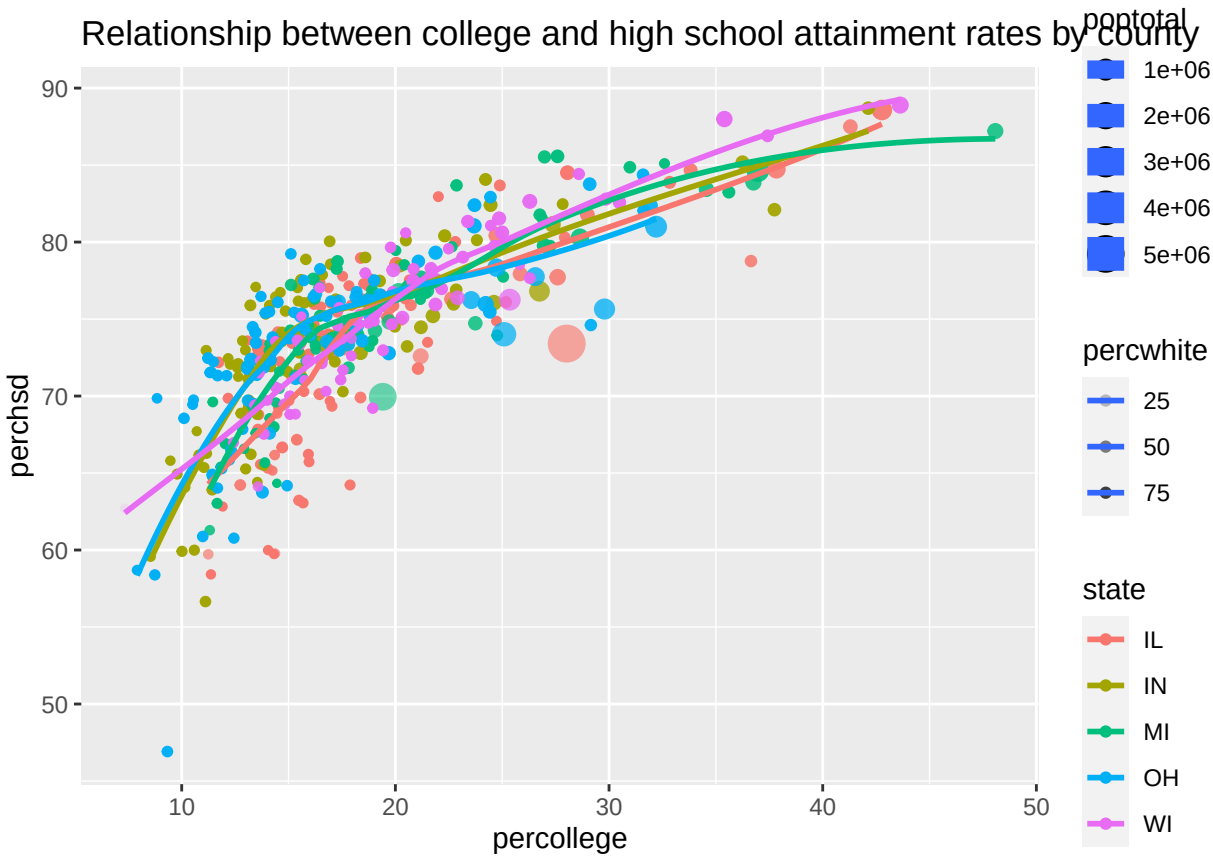


3. Add smooth lines. Get rid of the error around your smooth lines by adding the argument `se = FALSE`.

SOLUTION:

```
midwest %>%
  ggplot(aes(x = percollege,
             y = perchsd,
             color = state,
             size = poptotal,
             alpha = percwhite)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  labs(title = "Relationship between college and high school attainment rates by county")
```

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

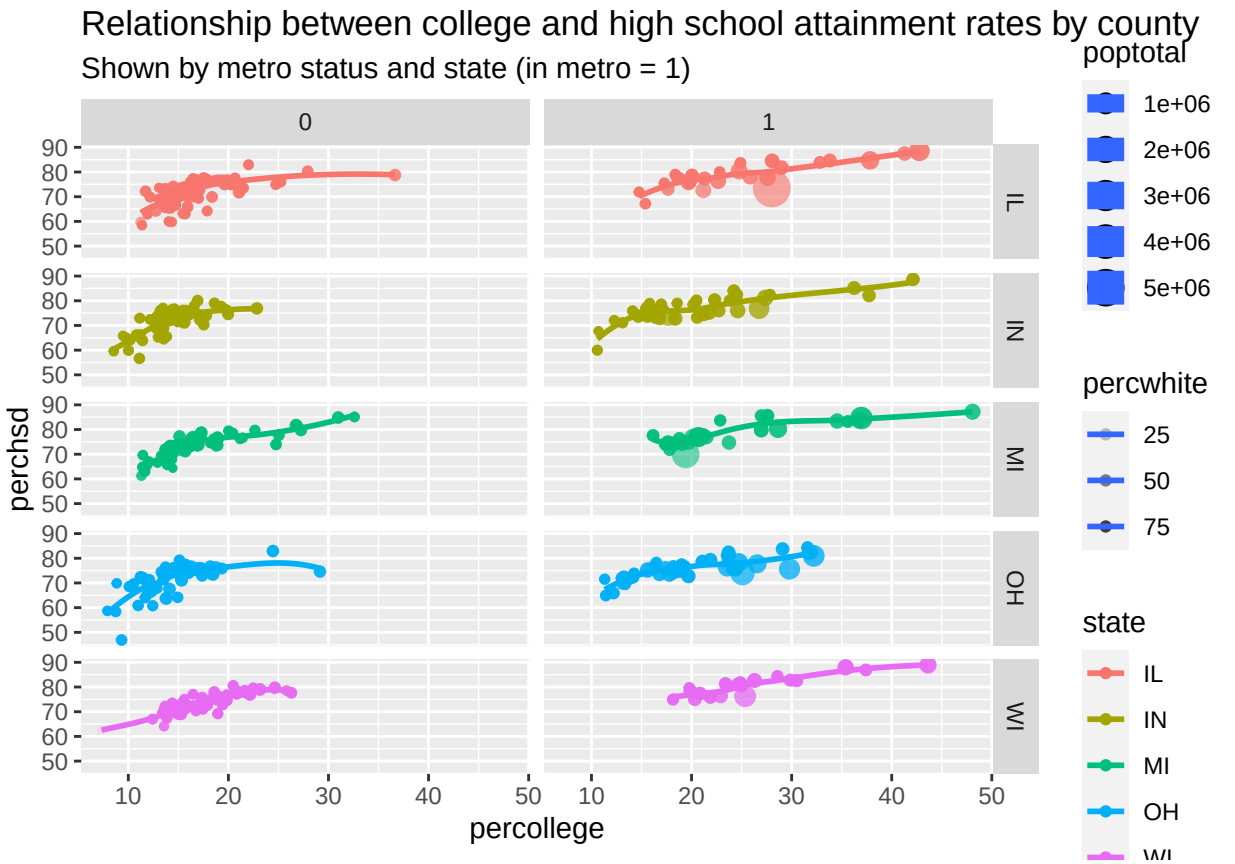


4. Now try faceting with `facet_grid` and the code `facet_grid(col = vars(inmetro), rows = vars(state))` to your plot.

SOLUTION:

```
midwest %>%
  ggplot(aes(x = percollege,
             y = perchsd,
             color = state,
             size = poptotal,
             alpha = percwhite)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  facet_grid(col = vars(inmetro), rows = vars(state)) +
  labs(title = "Relationship between college and high school attainment rates by county",
       subtitle = "Shown by metro status and state (in metro = 1)")
```

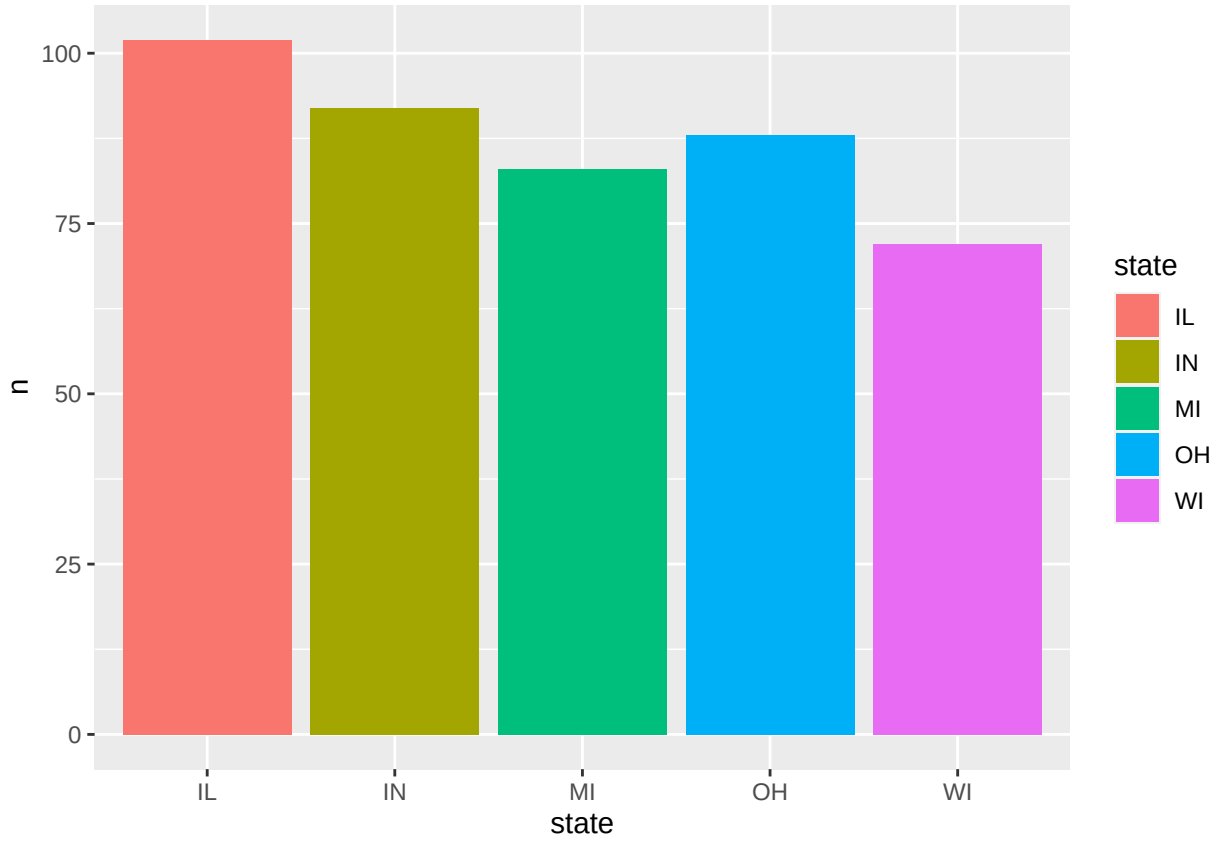
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



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

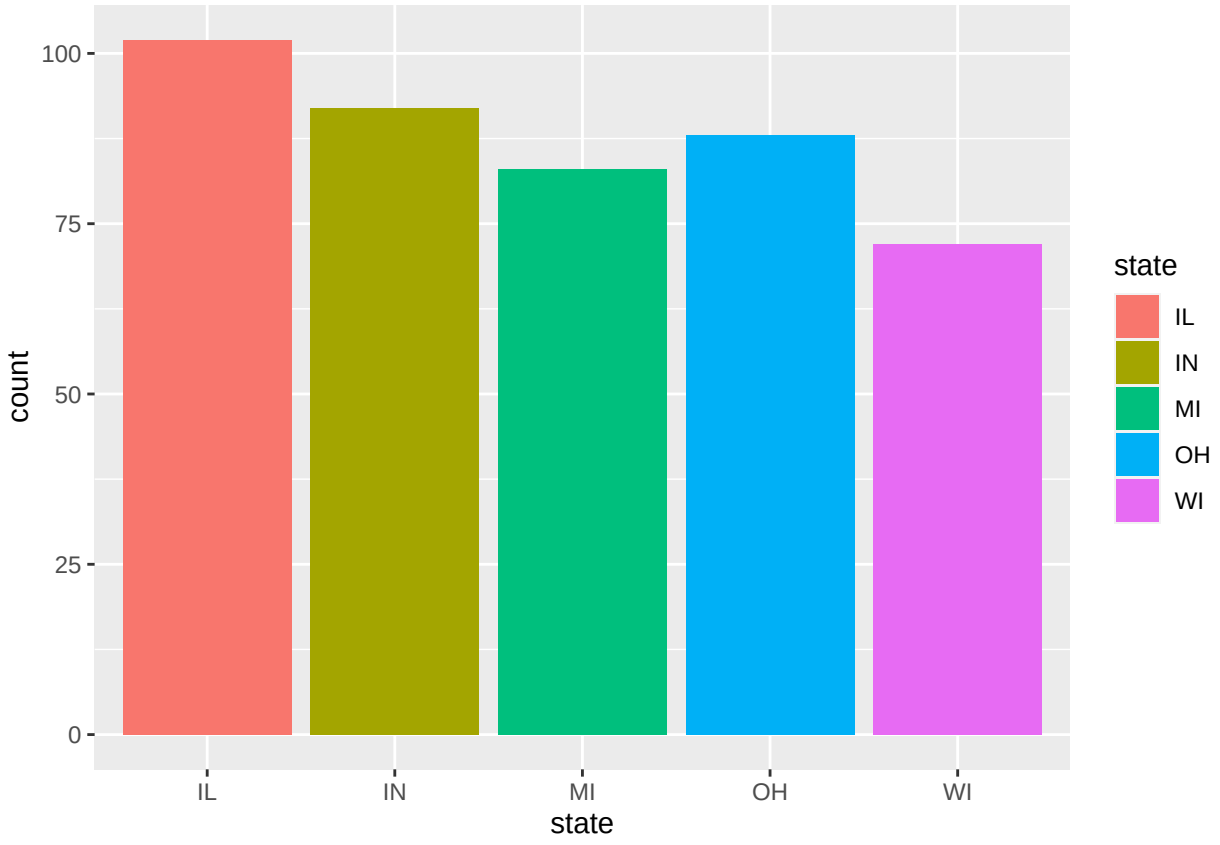
SOLUTION:

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



SOLUTION:

`geom_bar` does a statistical transformation where it calculates the number of rows per group (`x` value) and makes that the height of the bar. This is the same as using `count` on the data and then using `geom_col`. By default, `geom_bar()` has `stat = "count"` where `stat` is an argument that tells `geom_bar()` what kind of statistical transformation to do. We can get the `geom_col` behavior with `geom_bar(stat = "identity")`, `stat = "identity"` means we just take the `y` value from `n` directly.