The basics: 05 ggplot

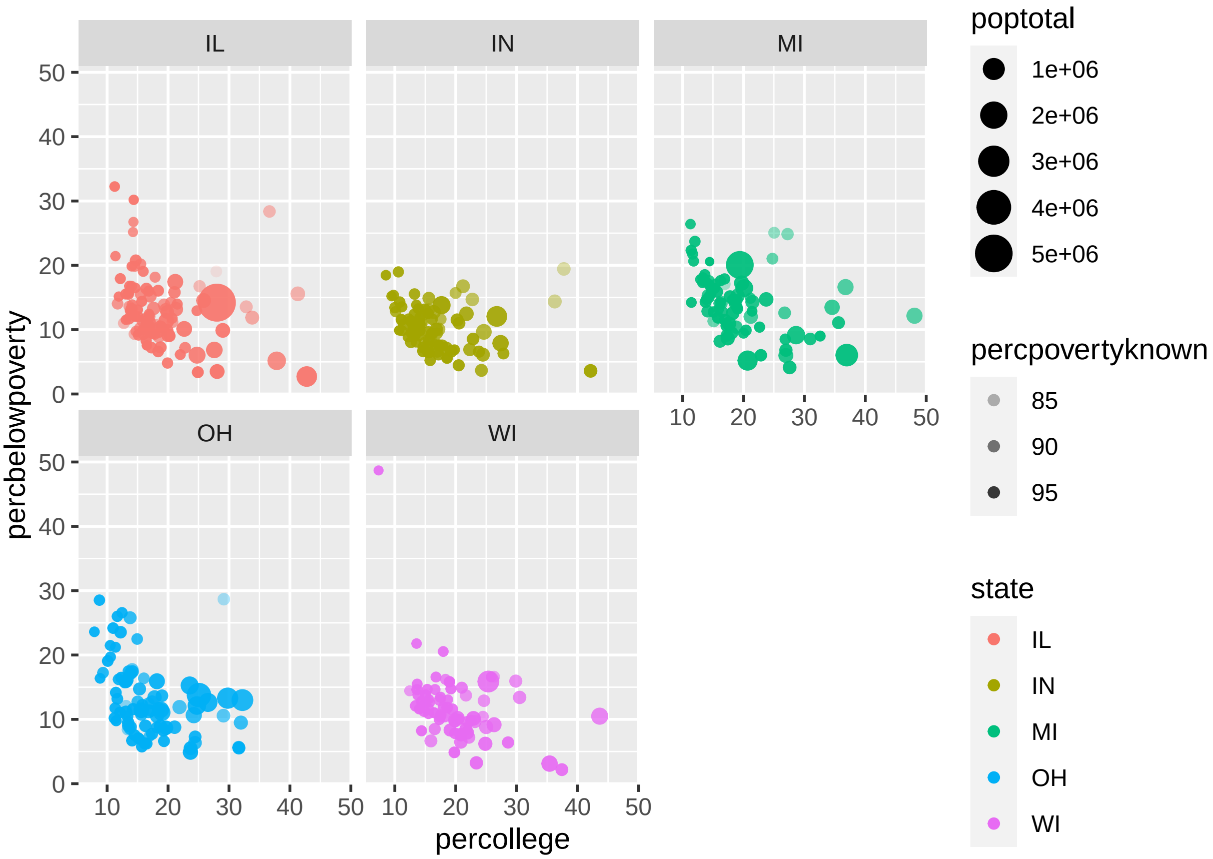
Ari Anisfeld

9/8/2020

Questions

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

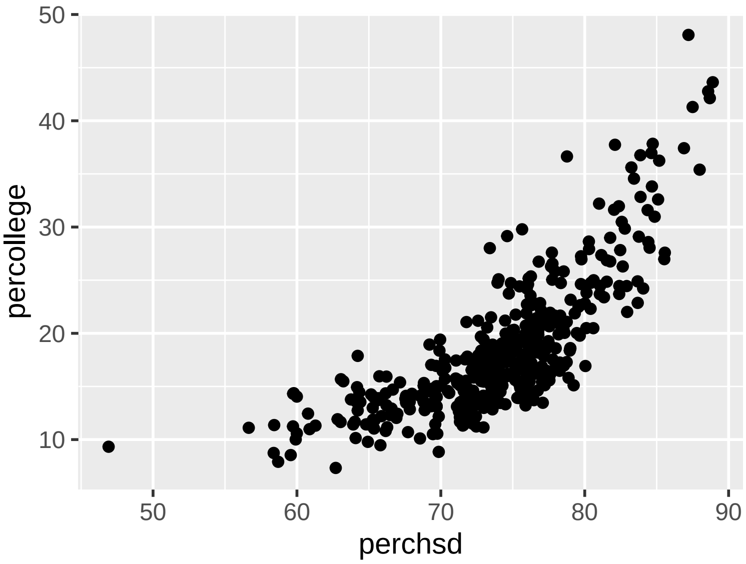
```
midwest %>%
  ggplot(aes(x = percollege,
             y = percbelowpoverty,
             color = state,
             size = poptotal,
             alpha = percpovertyknown)) +
  geom_point() +
  facet_wrap(vars(state))
```



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.

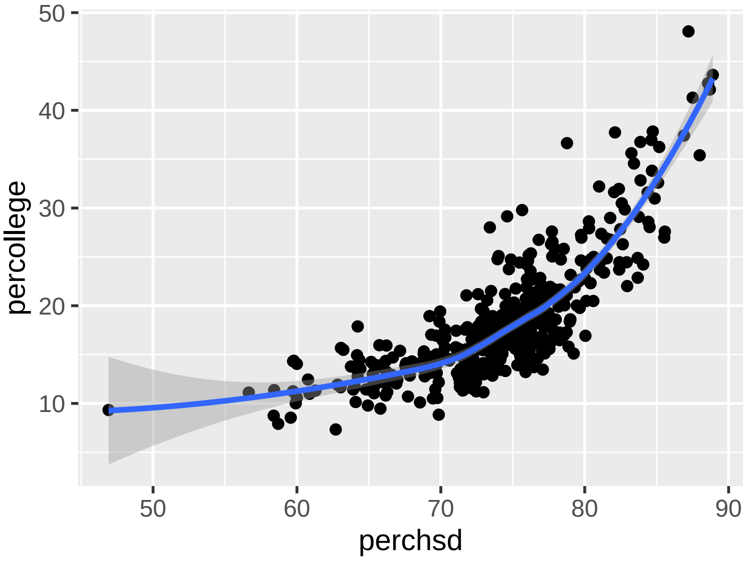
`geoms`

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

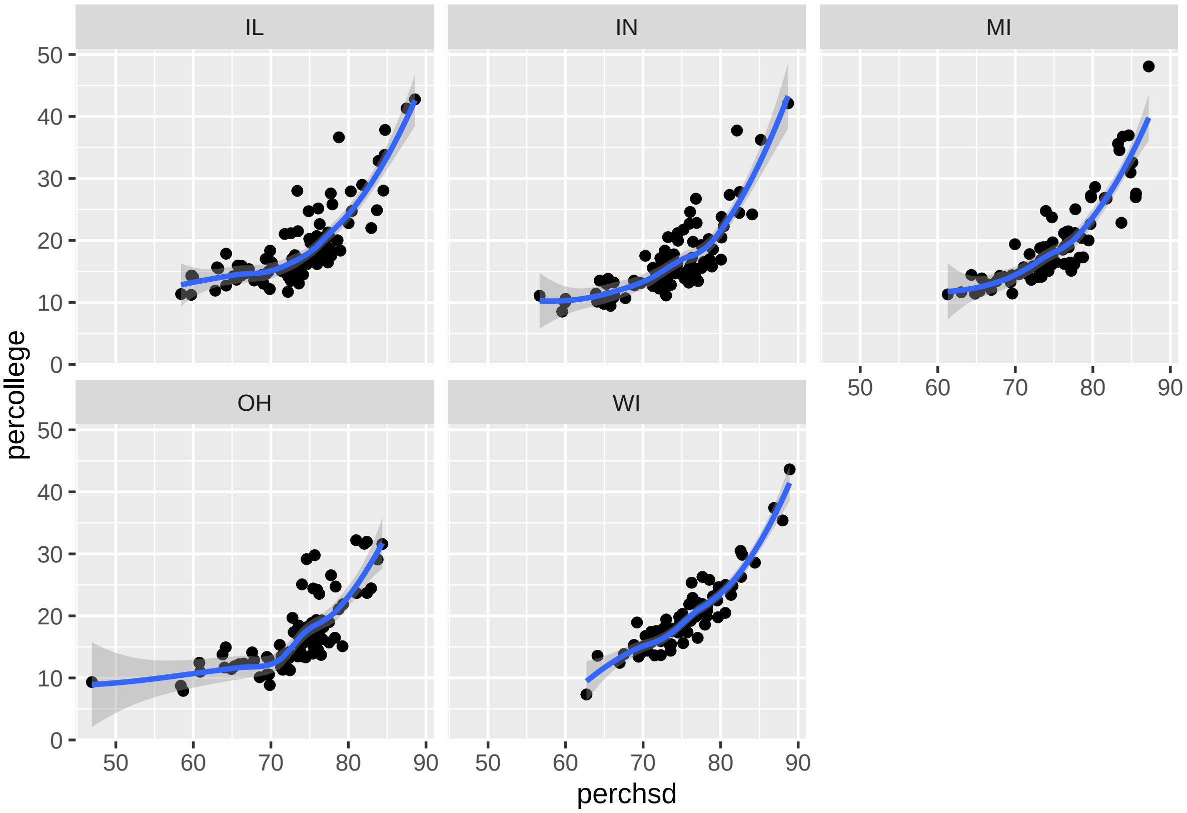


1.

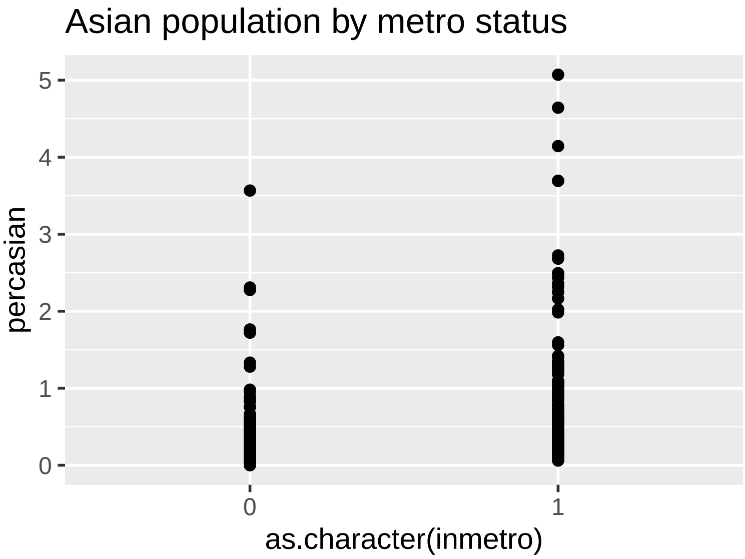
2. `` ## `geom_smooth()` using method = 'loess' and formula 'y ~ x' ``



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



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”.

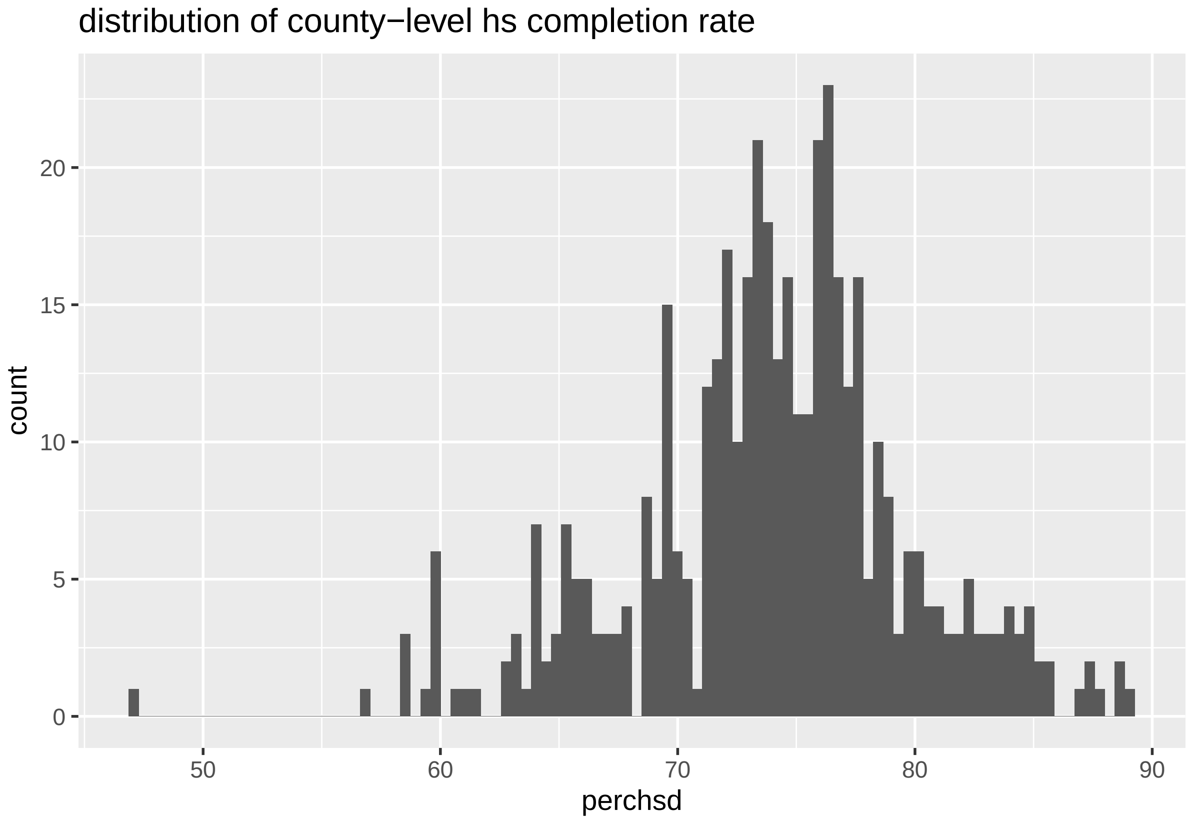
6. Use `geom_jitter()` instead of `geom_point()` for “Asian population by metro status”

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")
```

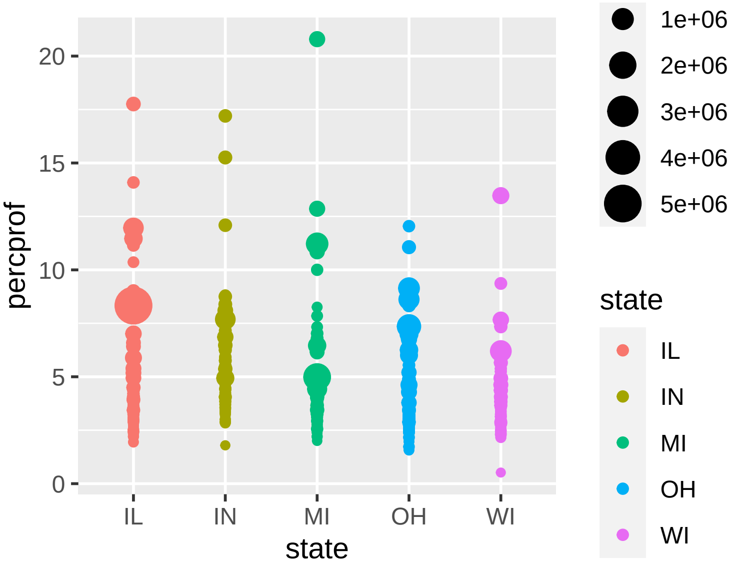


9. Remake "distribution of county-level hs completion rate" with `geom_density()` instead of `geom_histogram()`.
10. Add a vertical line at the median `perchsd` using `geom_vline`. You can calculate the median directly in the ggplot code.

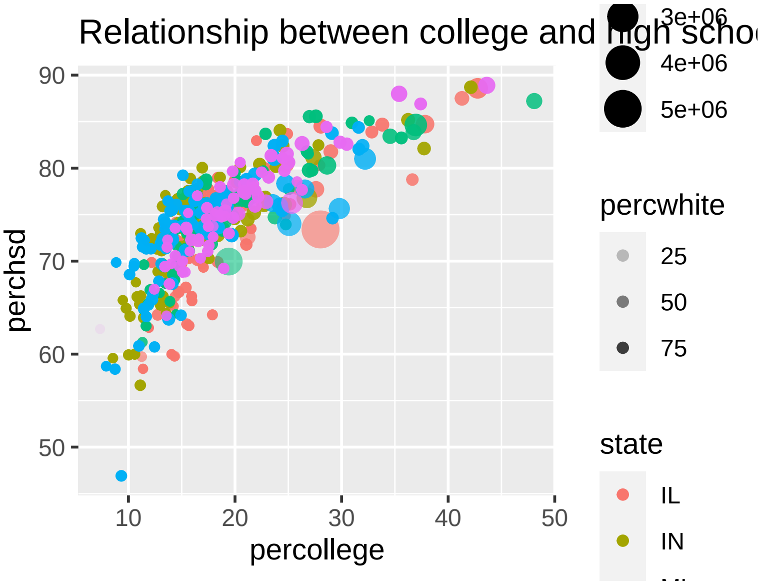
Aesthetics

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

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

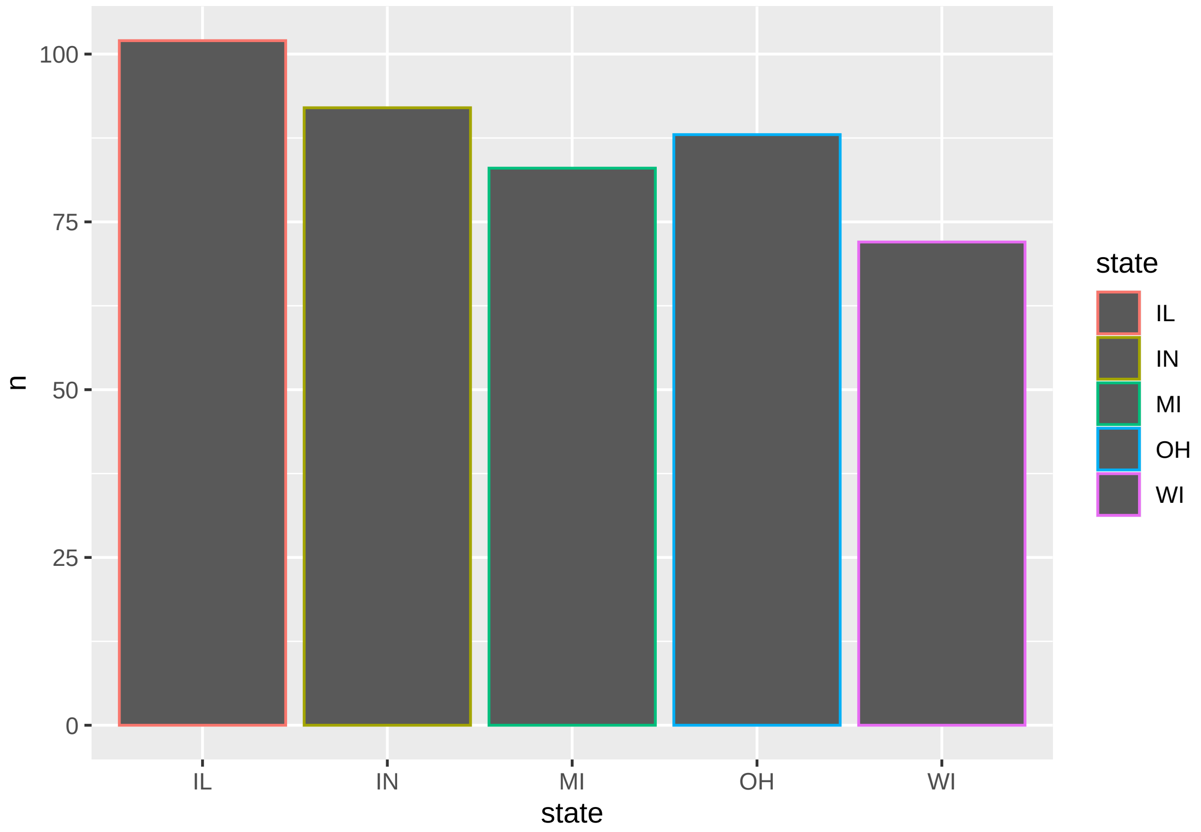


2. Use `x`, `y`, `color` and `size`.



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

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



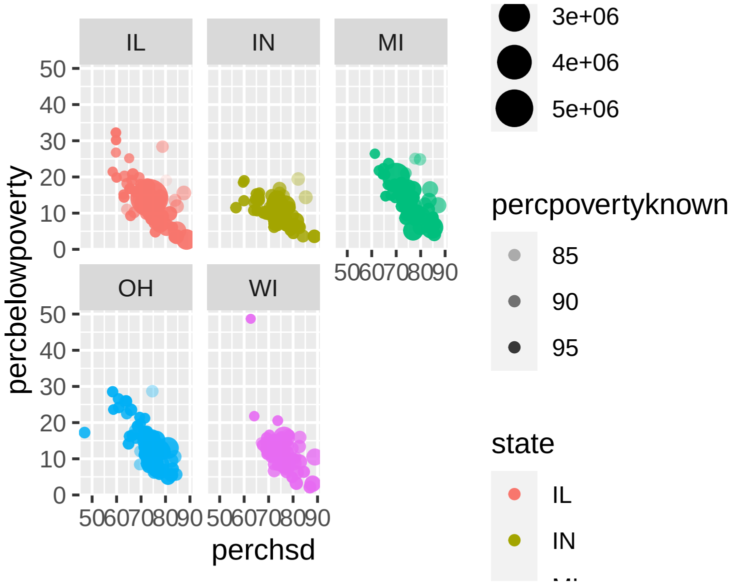
6. There's a `geom` called `geom_bar` that takes a dataset and calculates the count. Read the following code and compare it to the `geom_col` code above. Describe how `geom_bar()` is different than `geom_col`

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

Solutions

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

```r
midwest %>%
  ggplot(aes(x = perchsd,
             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.

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

2.
   ```r
   midwest %>%
     ggplot(aes(x = perchsd, y = percollege)) +
     geom_point() +
     geom_smooth()
   ```

3.
   ```r
   midwest %>%
     ggplot(aes(x = perchsd, y = percollege)) +
   ```

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

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

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

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

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

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

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

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

```
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()
```

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

   `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.

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

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

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

```
midwest %>%
  ggplot(aes(x = perchsd)) +
  geom_density() +
```

```r
  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`

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

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

   ```r
   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")
   ```

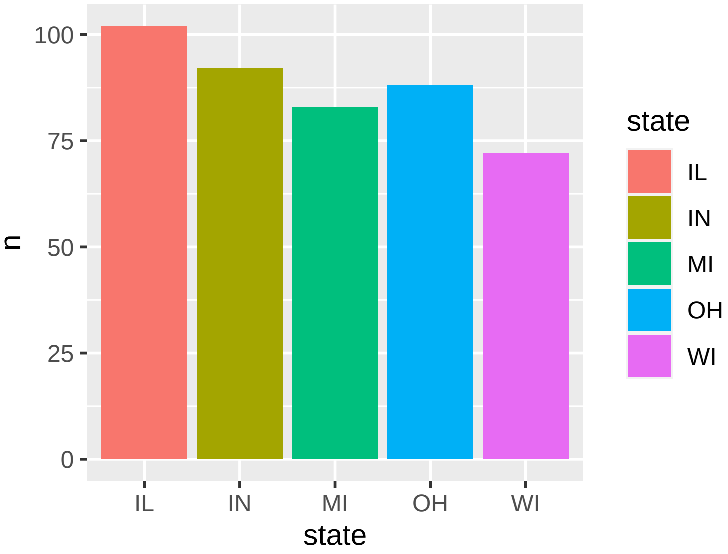
3. Add smooth lines. Get rid of the error around your smooth lines by adding the argument `se = FALSE`.

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

   ```r
   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)")
   ```

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

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



1. 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`

   `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.