CWRU DSCI351-351M-451: Homework 3

Prof.:Roger French, TA: JiQi Lim

21 September, 2018

Contents

3.1.0.1 HW2, 5 points, 2 questions.

- Question 1 = 3 points
- Question 2 = 2 points

Details

- Due Tuesday September 25th
  - Before Class
- The grading is done on how you show your thinking,
  - explain yourself and
  - show your Rcode and
  - the output you got from your code.
- Code style is important
  - Follow Rstudio code diagnostics notices
  - And the Google R Style Guide

To be done as an Rmd file,

- where you turn in
  - the Rmd file and
  - the compiled pdf showing your work.
  - and the R script of IntroR.R

You will want to produce a report type format

- (html and pdf type document) to turn in.
- And not an ioslides or beamer (slide type) compiled output.
  - These are presentation formats, and can be fussy

Also are you backing up your git repo

- in a second and third location,
- to avoid corruption problems?

3.1.0.2 1. Pipelines

We have seen pipeline %>% notation,

- You can read about them in
  - In dplyr package help
  - In Hadley Wickham's "R 4 Data Science" book
- This is a practice for you to begin using pipelines
- All work must be done using only one dplyr pipeline per question
  - with no intermediate variables,

  - including using summarise() to get results
- For reference, here are examples of
  - correct and incorrect answers for a problem

Example: What is the average petal length of setosa irises?

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)

# incorrect
data("iris")
df <- iris %>%
  filter(Species == "setosa")

mean(df$Petal.Length)
```

```
## [1] 1.462
```

```
# correct
iris %>%
  filter(Species == "setosa") %>%
  summarise(mean(Petal.Length))
```

```
##   mean(Petal.Length)
## 1              1.462
```

1a: What is the max, min, and average price of diamonds with a "Very Good" cut?

```
data("diamonds")
head(diamonds)
```

```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1 0.23  Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2 0.21  Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
## 3 0.23  Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4 0.290 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5 0.31  Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6 0.24  Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

```
str(diamonds)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    53940 obs. of  10 variables:
##  $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
##  $ cut    : Ord.factor w/ 5 levels "Fair"<"Good"<..: 5 4 2 4 2 3 3 3 1 3 ...
##  $ color  : Ord.factor w/ 7 levels "D"<"E"<"F"<"G"<..: 2 2 2 6 7 7 6 5 2 5 ...
##  $ clarity: Ord.factor w/ 8 levels "I1"<"SI2"<"SI1"<..: 2 3 5 4 2 6 7 3 4 5 ...
```

```
##  $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

```
?diamonds
```

1b: What is the average carat weight for each cut of diamond?

1c: Add a variable that is a ratio of the price per weight for each diamond.

- Which cut of diamond has the highest average price to weight ratio and what is it?

1d: What is the 100th most expensive diamond in each color group of at least 0.30 carets?

- (show only price and color variables)

3.1.0.3 2. ggplot2

ggplot2 is a package for making plots from data.

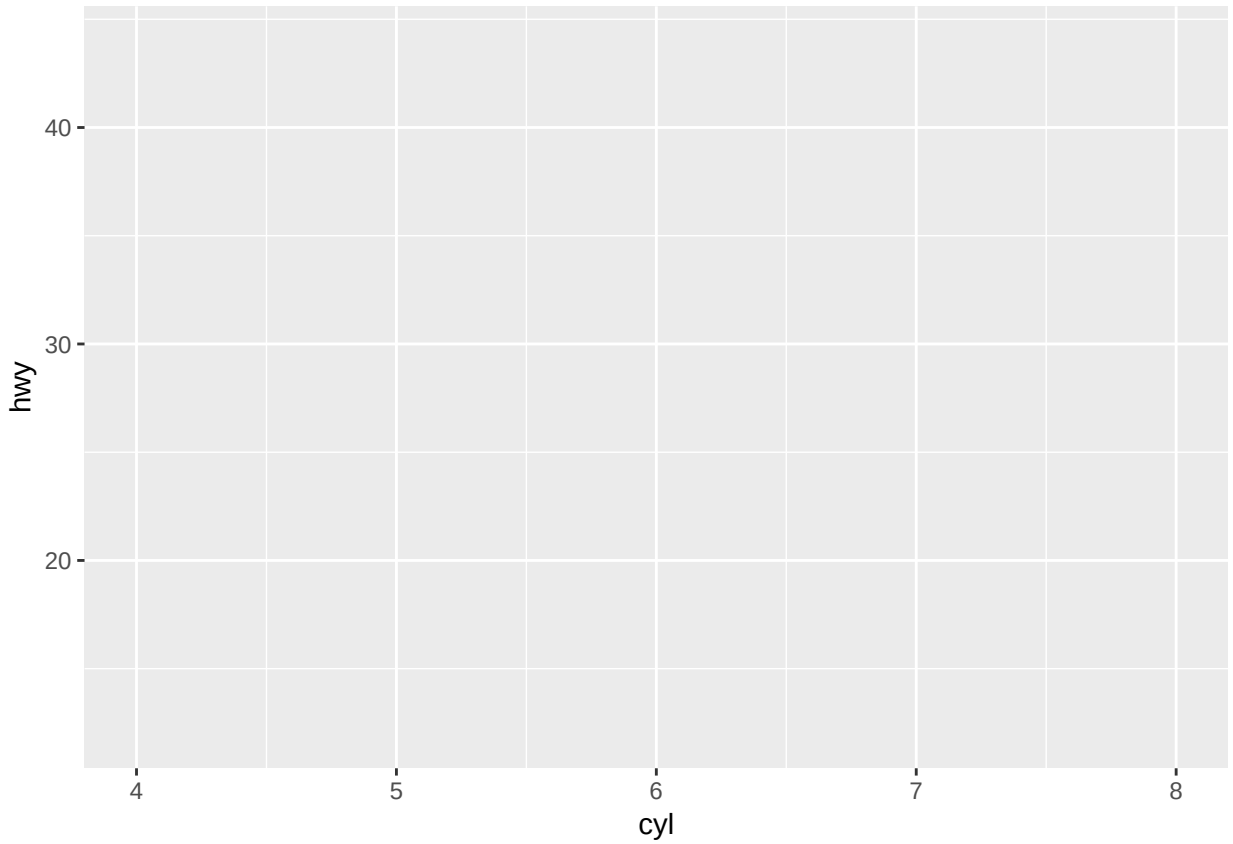
It provides tools for making complex and detailed graphs.

ggplot2 builds graphs in layers,

- where first the graph must be specified,
- then layers are added to the plot using the '+' operator.

In this example nothing appears in the plot

```
library(ggplot2)
data("mpg")

ggplot(data = mpg, aes(x = cyl, y = hwy))
```



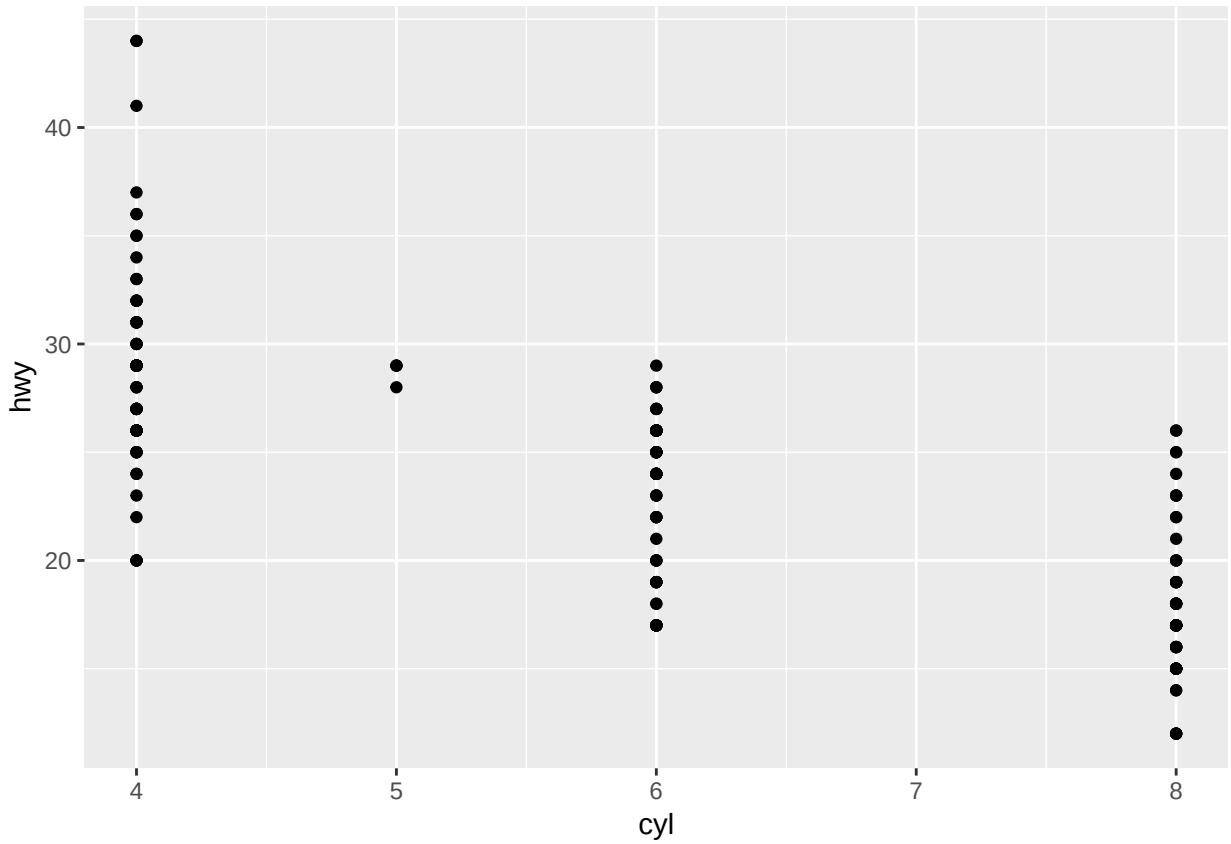
This is because I did not define the next layer,

- all I did was define some kind of graph between cylinders and highway mpg

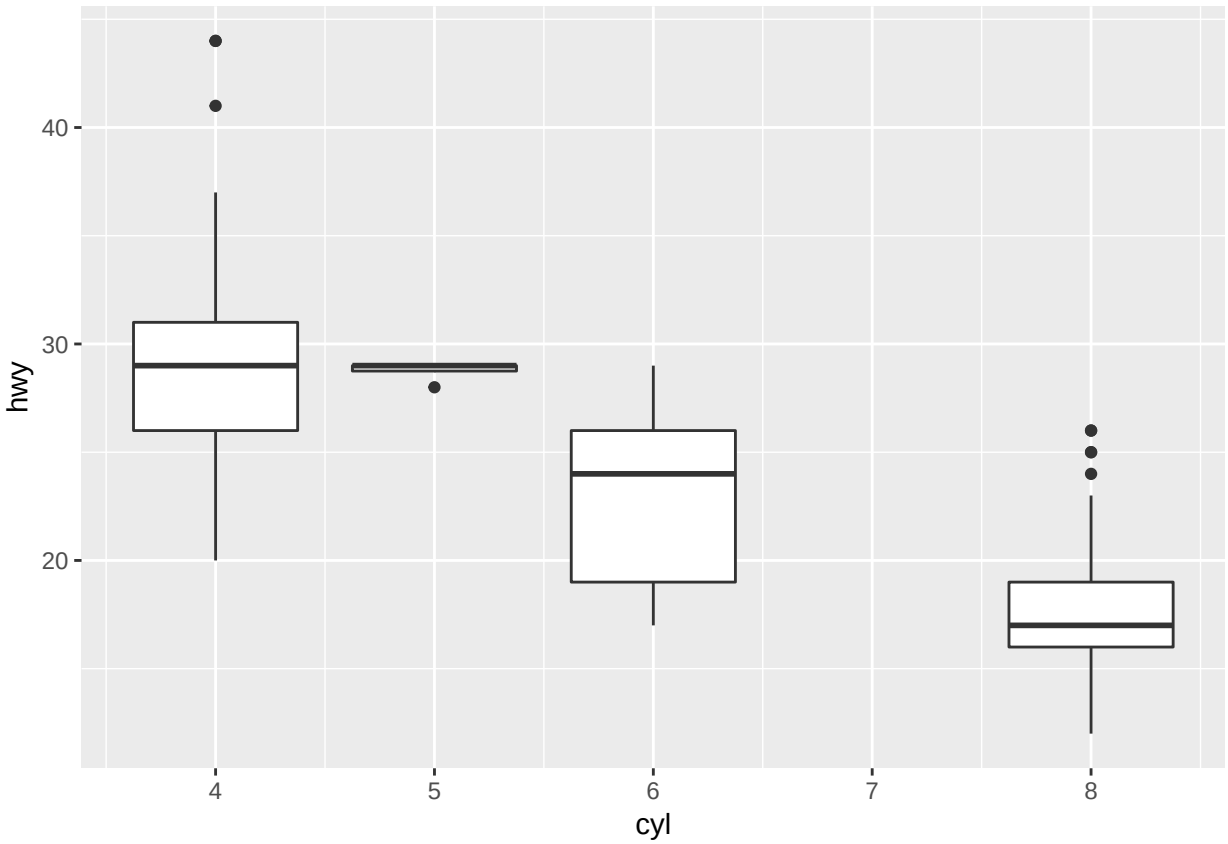
Since we have already defined the data at the beginning,

- we don't need to specify it in the layer

```
ggplot(data = mpg, aes(x = cyl, y = hwy)) +
  geom_point()
```



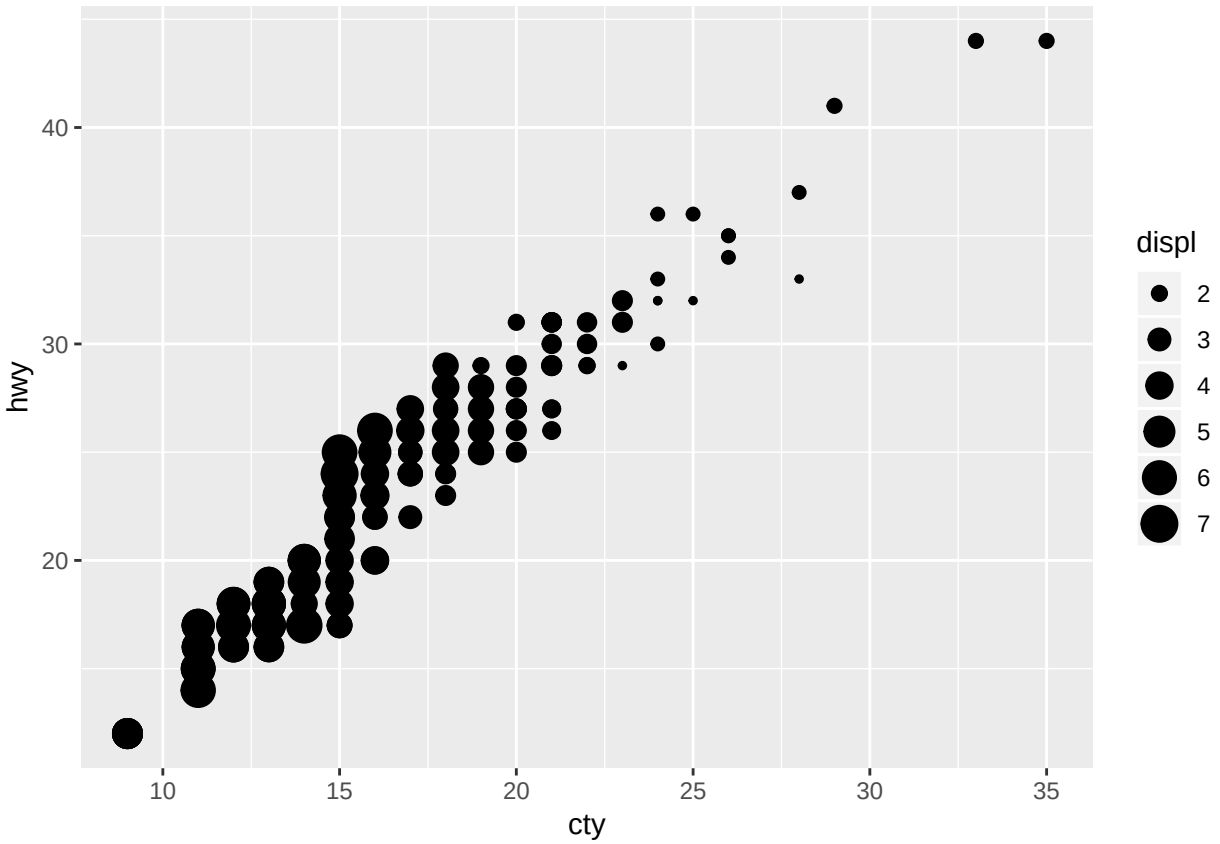

```
# or a different layer

# here we have to define cyl as the group for each box
ggplot(data = mpg, aes(x = cyl, y = hwy)) +
  geom_boxplot(aes(group = cyl))
```



We can add additional information about showing data in our plot

- by adding parameters into the aesthetics (aes()) function

```
ggplot(data = mpg, aes(x = cty, y = hwy)) +
  geom_point(aes(size = displ))
```



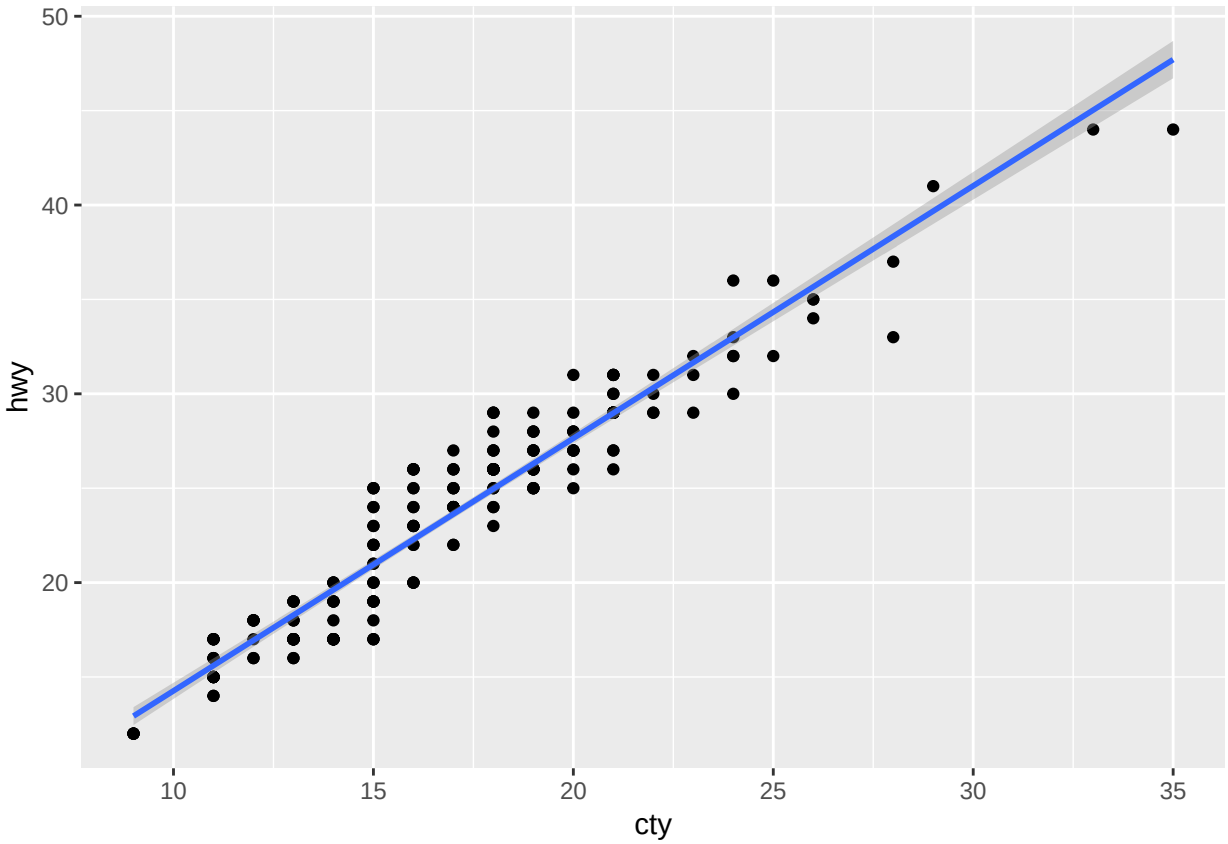
We can also add on additional layers if we want to,

- keep in mind ordering is important.

The data for each layer can be defined per layer

- this is important if you're trying to add multiple data sets to a plot

```
ggplot() +
  geom_point(data = mpg, aes(x = cty, y = hwy)) +
  geom_smooth(data = mpg, aes(x = cty, y = hwy), method = "lm")
```



Now it's your turn to make some plots

- All plotting must be done using ggplot2,
- Any data manipulation must be done with dplyr pipelines
  - running into the ggplot funciton

2a: Use the mtcars data set,

- plot mpg vs displacement and color by cylinders

```
data("mtcars")
?mtcars
```

2b: Create a boxplot of the horsepower readings for each cylinder count,

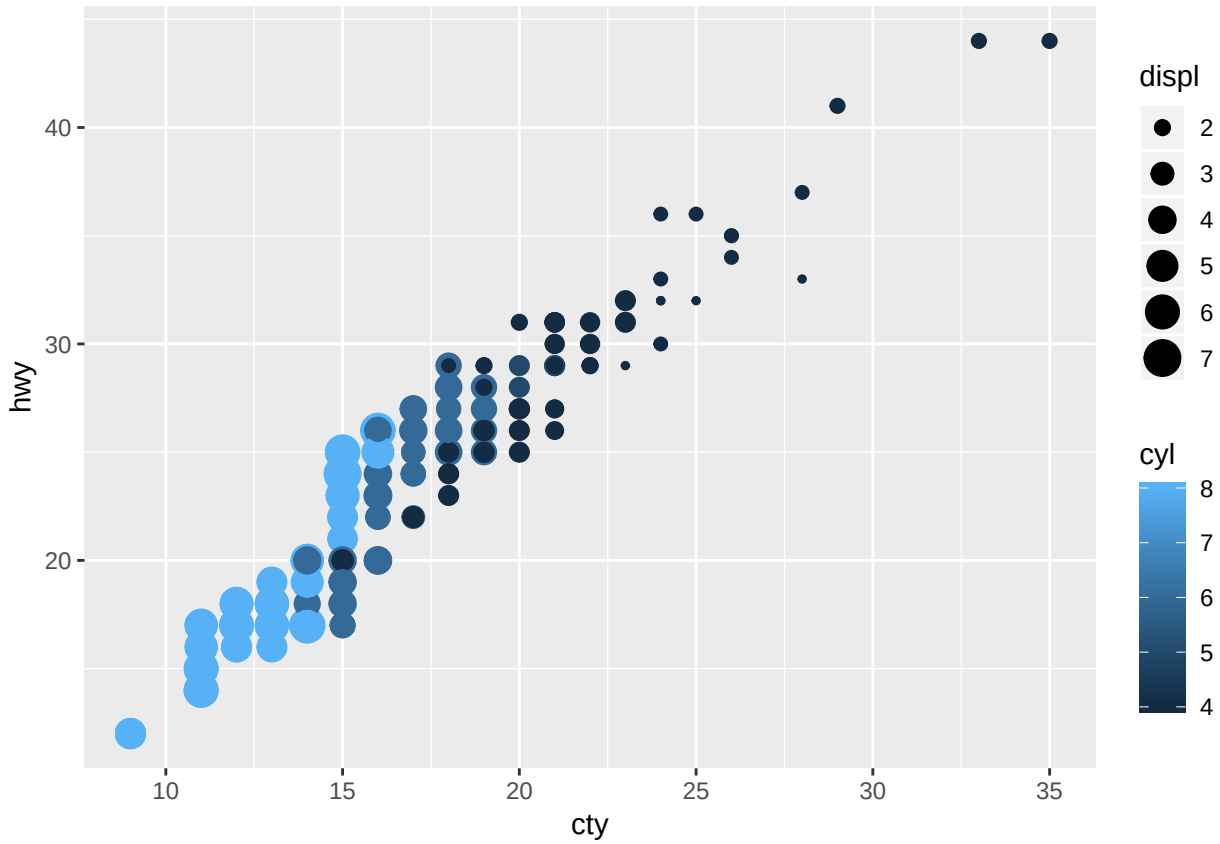
- show the data points on top of the plot

2c: Plot a histogram of the number cars in each carburetor count group

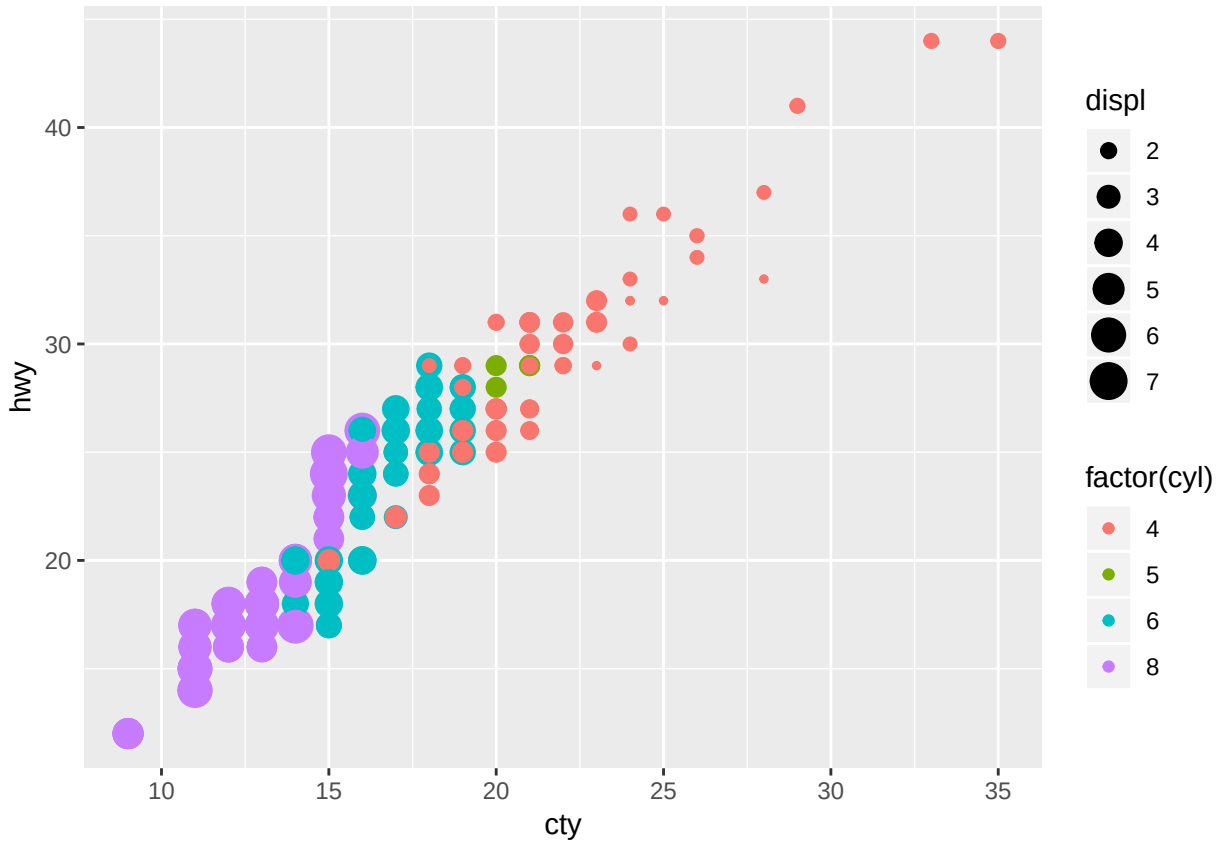
2d: Explain why these two plots look different,

- why does the color and key change between them?

```
ggplot(data = mpg, aes(x = cty, y = hwy)) +
  geom_point(aes(color = cyl, size = displ))
```




```
ggplot(data = mpg, aes(x = cty, y = hwy)) +
  geom_point(aes(color = factor(cyl), size = displ))
```


hwy
40
30
20
10
15
20
25
30
35
cty
displ
2
3
4
5
6
7
factor(cyl)
4
5
6
8