Jerry Cheng

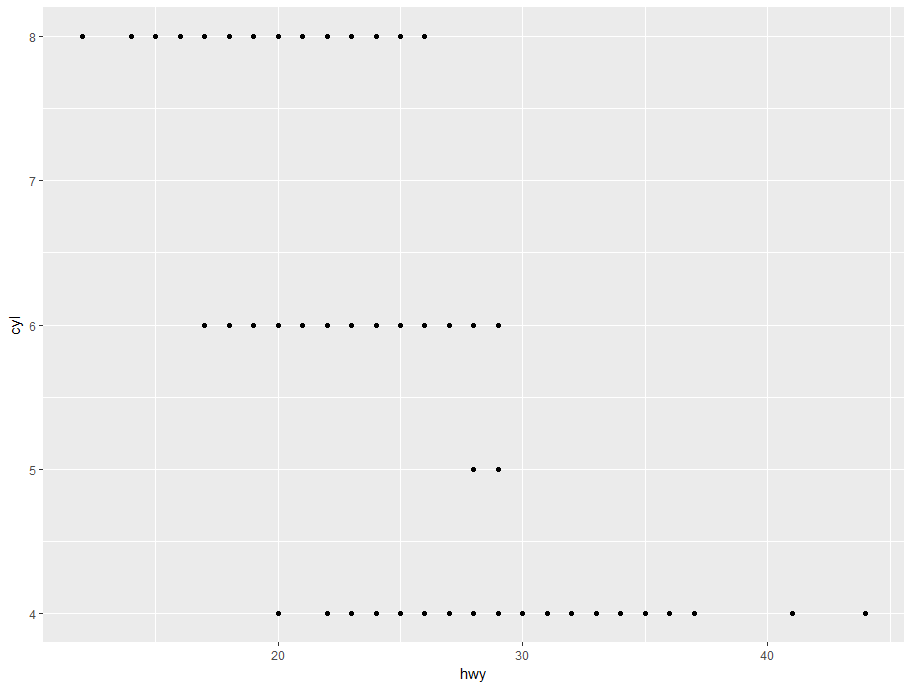
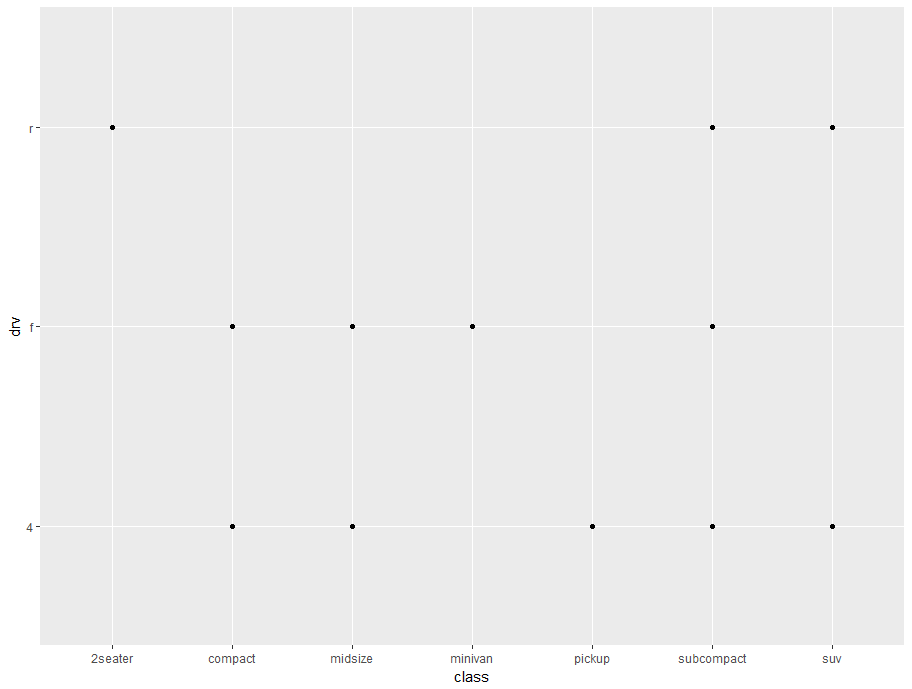
Computer Science with Data Analysis

R

**Chapter 3:**

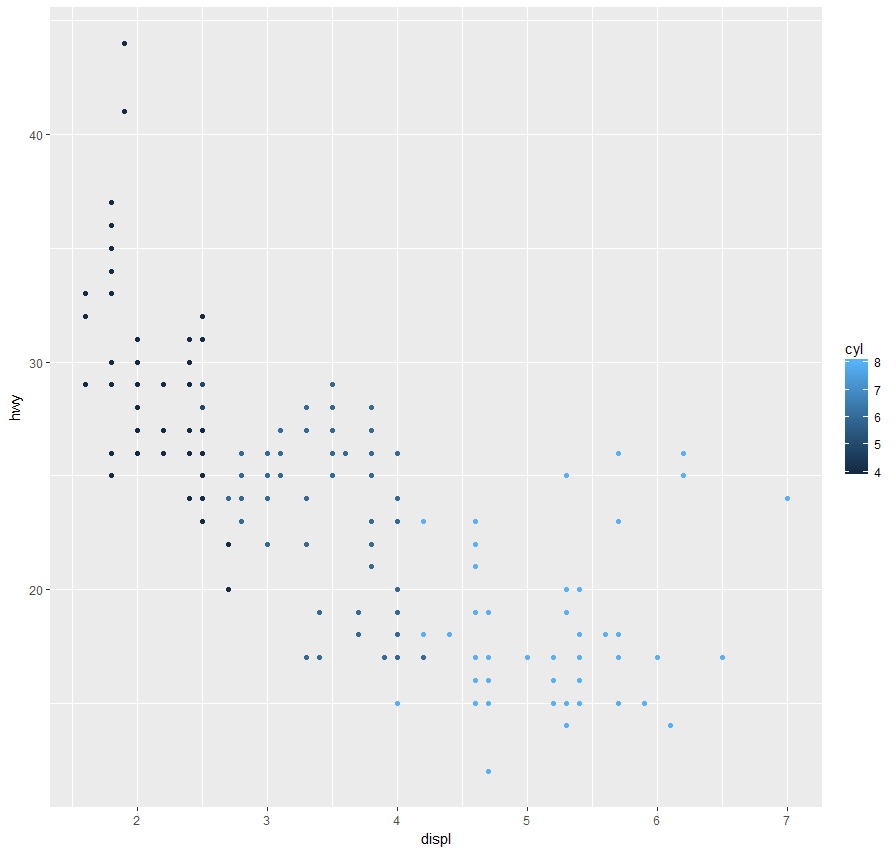
* install.packages(“tidyverse”) – Installs the tidyverse package
* library(tidyverse) – let’s you access the data sets under tidyverse
* format: package::function()
  + ggplot2::ggplot() – explicitly tells you that we’re using the ggplot() function from the ggplot2 package
* To display a data frame, type it in – mpg
* ggplot(data = mpg) +
  + geom\_point(mapping = aes(x = displ, y = hwy)) #Creates a ggplot
* The graphing template:
* ggplot(data = <DATA>) +
  + <GEOM\_FUNCTION>(mapping = aes(<MAPPINGS>))

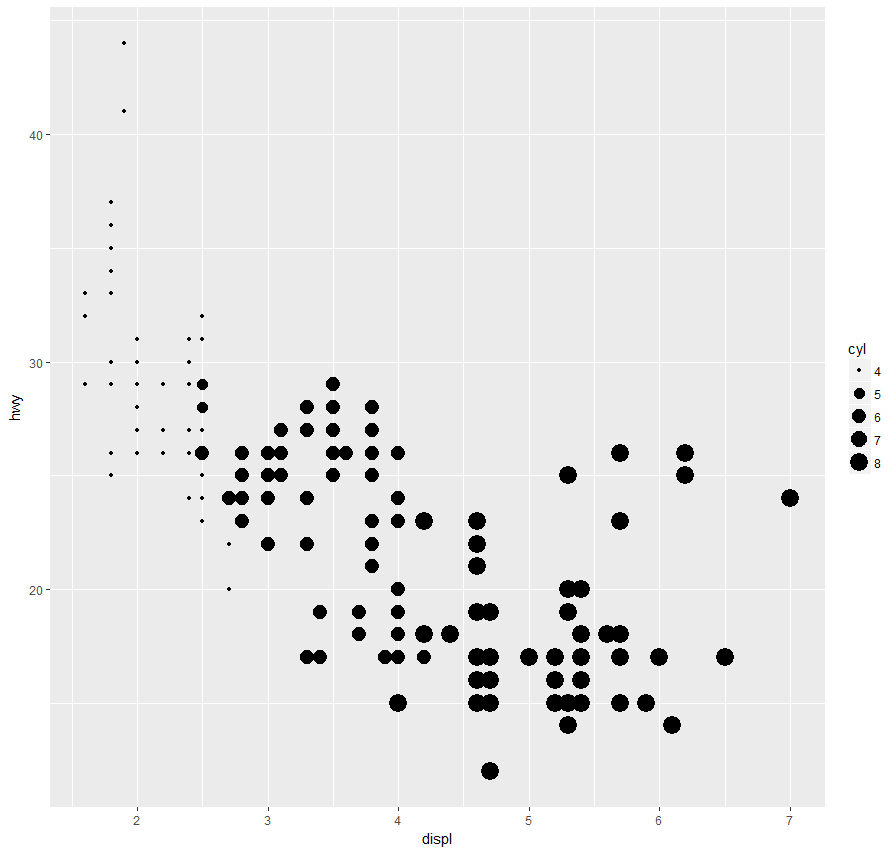
3.2.4 Exercies:

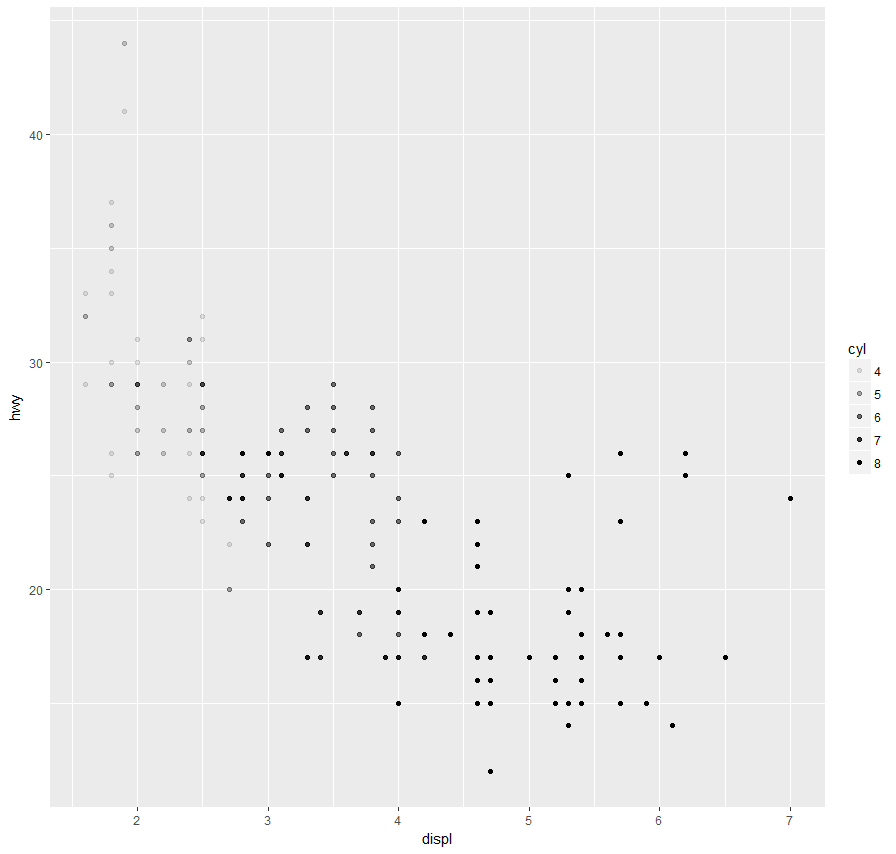
1. When you run ggplot(data = mpg), nothing happens.
2. In mpg, there is 234 rows and 11 columns (variables).
3. Drv represents what kind of drive the car is (full wheel drive, 2 wheel drive…)
4. 
5.  The two data points don’t have any real significance to them, therefore making the graph useless. There is also no trend in the data.

3.3.1 Exercises

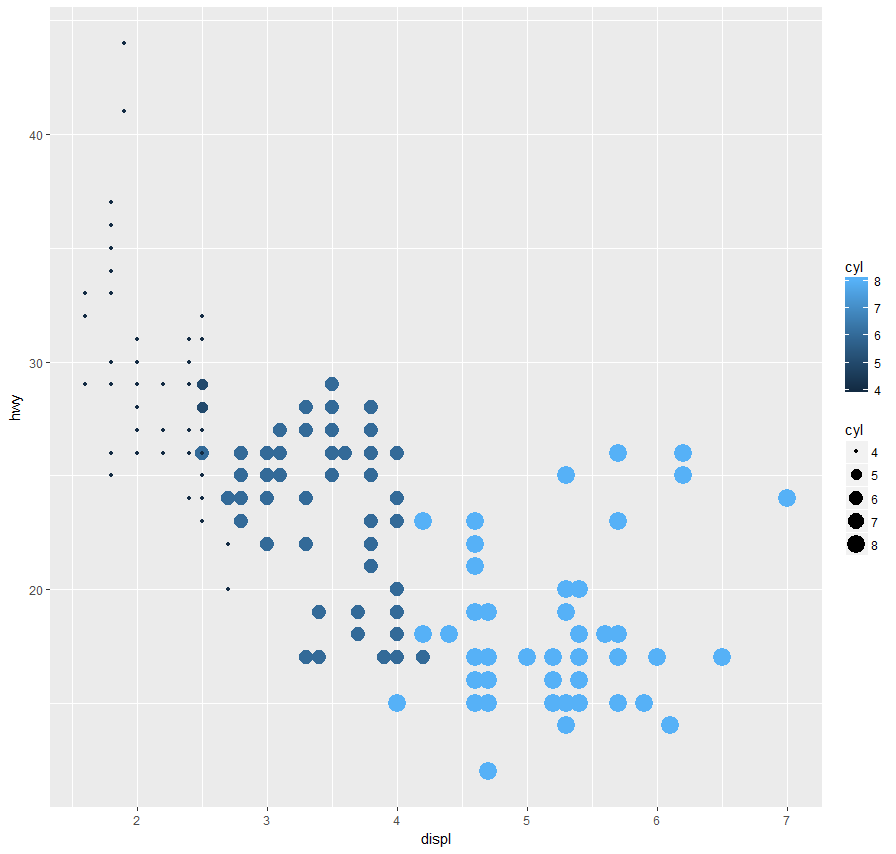
1. The reason why the plot isn’t displaying in blue is because of a parenthesis error.
2. When you run ?mpg, it will display if the data is numerical, telling you if it is continuous or not
   1. Categorial: model, trans, drv, ft, class
   2. Continuous: displ, year, cyl, cty, hwy
3. Categorial variables will have certain traits binded to specific types, whereas continuous will have a scale.

Color

Size

Alpha

1. Using the same variable for different aesthetics will work fine. Trying to use the same aesthetic twice will result in R using the first one and dropping an error message.

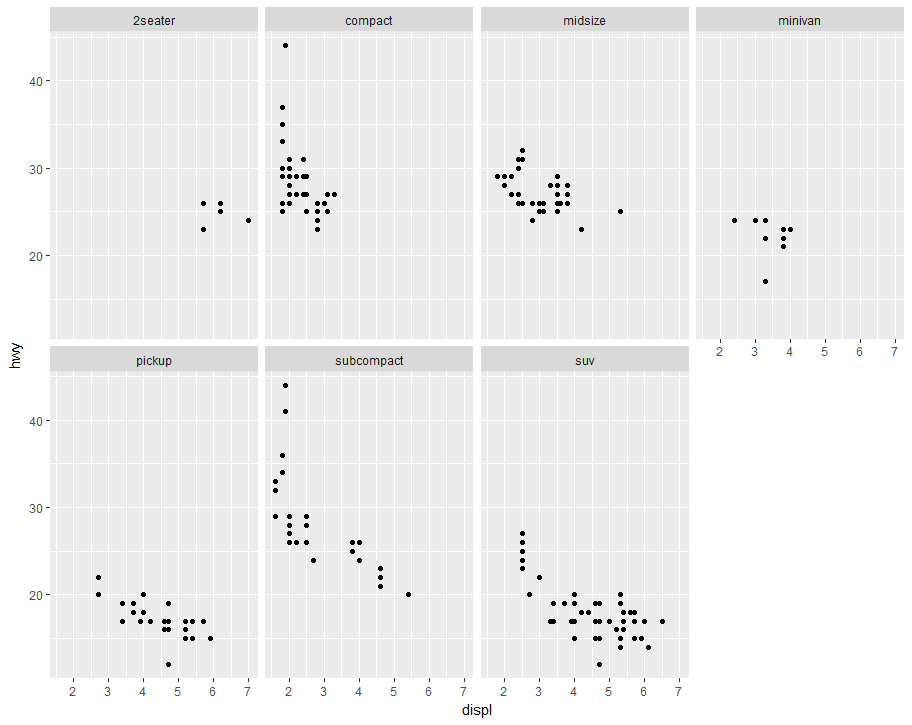


1. The stroke aesthetic will control the width of the border, working with shapes that are compatible with the stroke aesthetic. It will only work with shapes that have a different fill color – if a shape is a static filled color, you can’t change the border width.
2. It will evaluate the expression and plot the result if possible. It may or may not work.

3.4

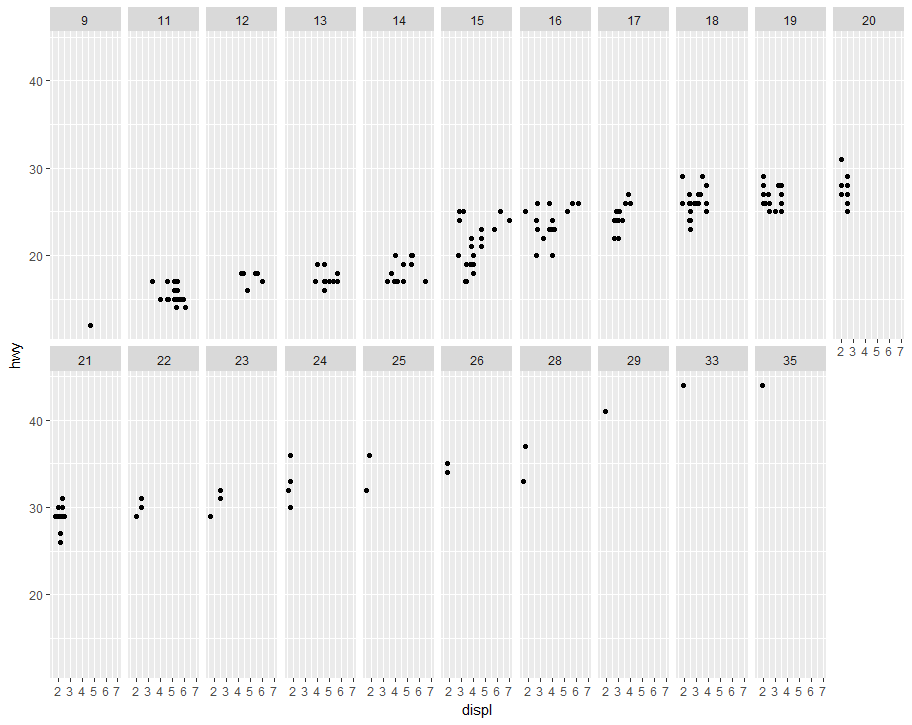
* Common errors: “”, (), stuff like that

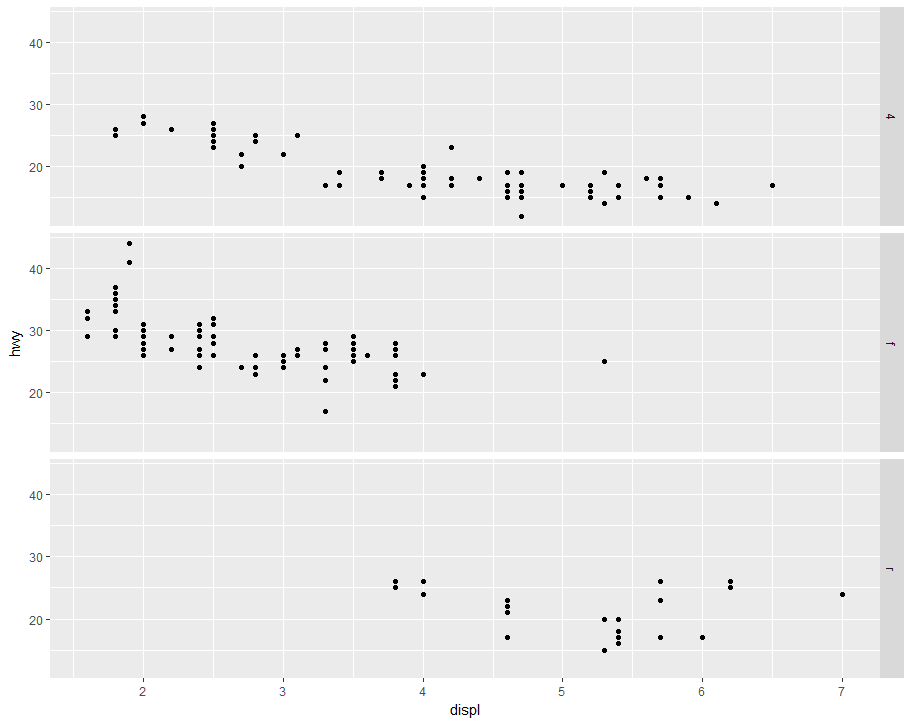
3.5

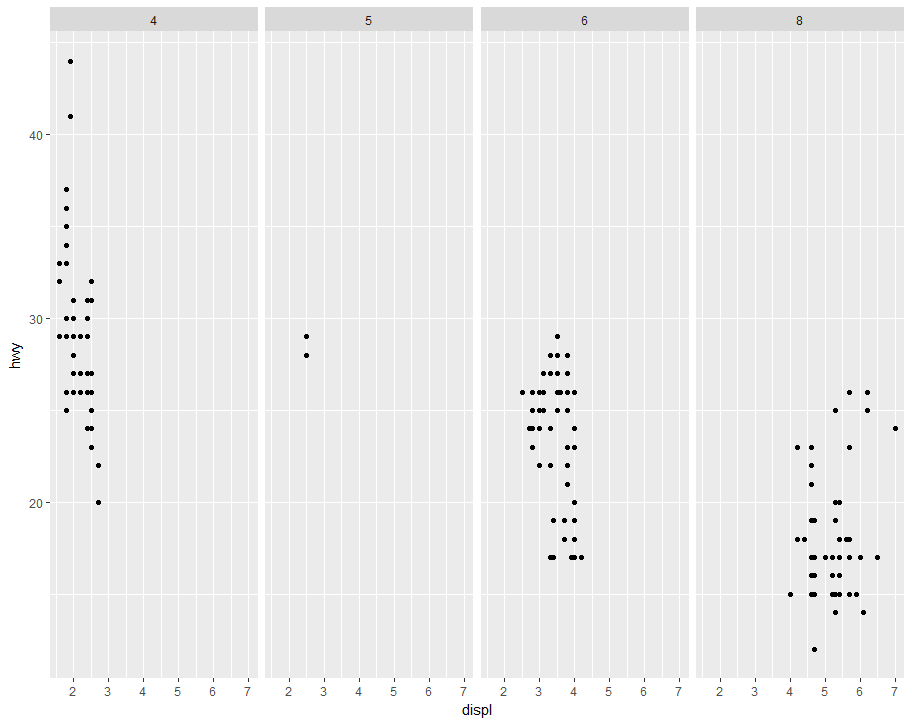
* By splitting your graph into facets, you can have numerous subplots
* **ggplot**(data = mpg) +
* **geom\_point**(mapping = **aes**(x = displ, y = hwy)) +
* **facet\_wrap**(~ class, nrow = 2)
* This will allow you to facet an additional variable (being class) to the plot
* Result: 
* To facet using the combination of two variables, use the following code:
* **ggplot**(data = mpg) +
* **geom\_point**(mapping = **aes**(x = displ, y = hwy)) +
* **facet\_grid**(drv ~ cyl)
* It will facet drv and cyl

3.5.1 Exercises

1. If you facet a continuous variable, it will work.



1. Empty means that there was no data that would satisfy what boundaries you gave it. It would imply that there’s no data.
   1. 



* 1. The . is a placeholder, allowing us to just have a facet in a singular dimension.

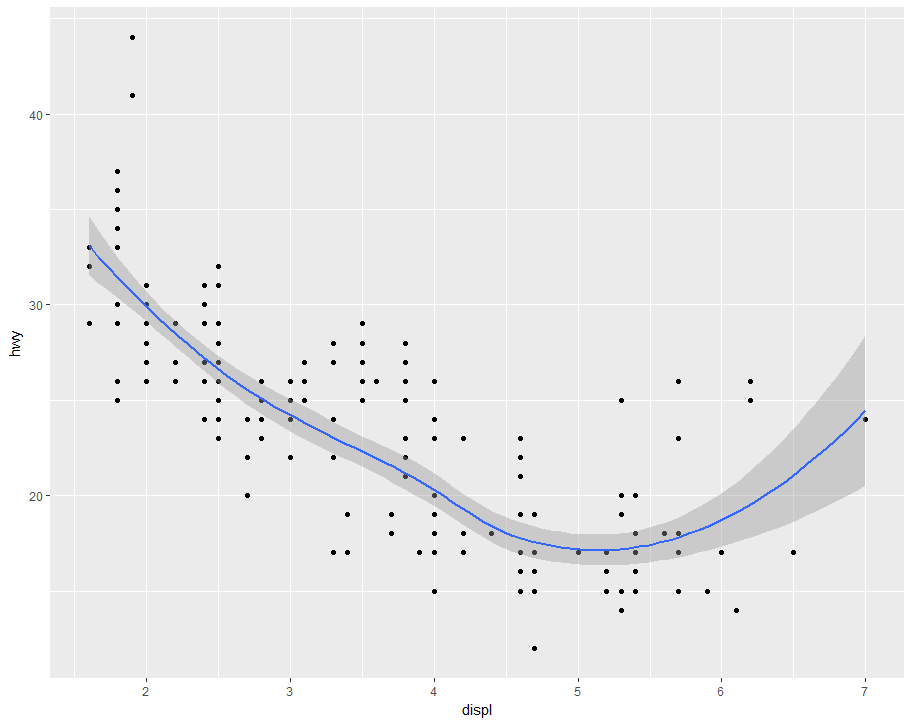
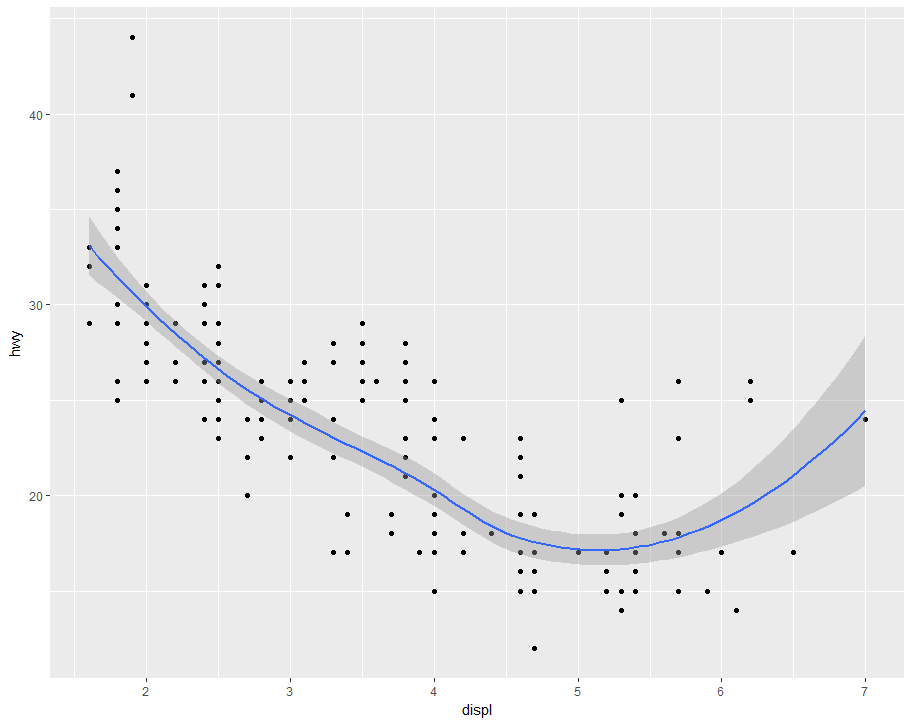
1. With a large data set, just using colors may get to the point where it is too confusing to use the colors anymore. However, facets are harder to use with a single glance. With more data comes the bigger necessity to use more facets.
2. Nrow adjusted the number of rows, and ncol adjusts the number of columns. Facet grid doesn’t have this option because through the use of facet grid, it is already implying that you have both rows and columns.
3. You put the greater number of columns on the wider side because most modern resolutions are wider than they are tall. 1920x1080, 3124x2448, etc.

3.6

* The geom function allows you to adjust the kind of plot that is displayed –
  + Geom\_point will do a dot graph, geom\_smooth will make a smooth graph
* Every geom function takes a mapping argument- however, not all aesthetics work for all types of graphs.

3.6.1 Exercises

1. A
   1. Line Chart: geom\_line
   2. Boxplot: geom\_boxplot
   3. Histogram: geom\_histogram
   4. Area Chart: geom\_area
2. In my head, I imagine two graphs, a line graph and a dot graph, being on top of each other, separated by numerous colors. 
3. The show\_legend = false probably gets rid of the legend. It was probably used earlier in the chapter because the legend was too complex/confusing.
4. The se argument changes whether or not there is a confidence interval around the smoothing line.

They look the same because they are the same graphs with the same commands but formatted differently.

* 1. Ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
     1. Geom\_point()+
     2. Geom\_smooth(se = FALSE)
  2. Ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
     1. Geom\_point() +
     2. Geom\_smooth(se = FALSE)
  3. Ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color = drv)) +
     1. Geom\_point() +
     2. Geom\_smooth(se = FALSE)
  4. Ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
     1. Geom\_point(aes(color=drv)) +
     2. Geom\_smooth(aes(linetype = drv), se = FALSE)
  5. Ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
     1. Geom\_point(size = 4, color = “white”) +
     2. Geom\_point(aes(color = drv))