In Class Exercises 2

Tobias Boggess

2/7/2022

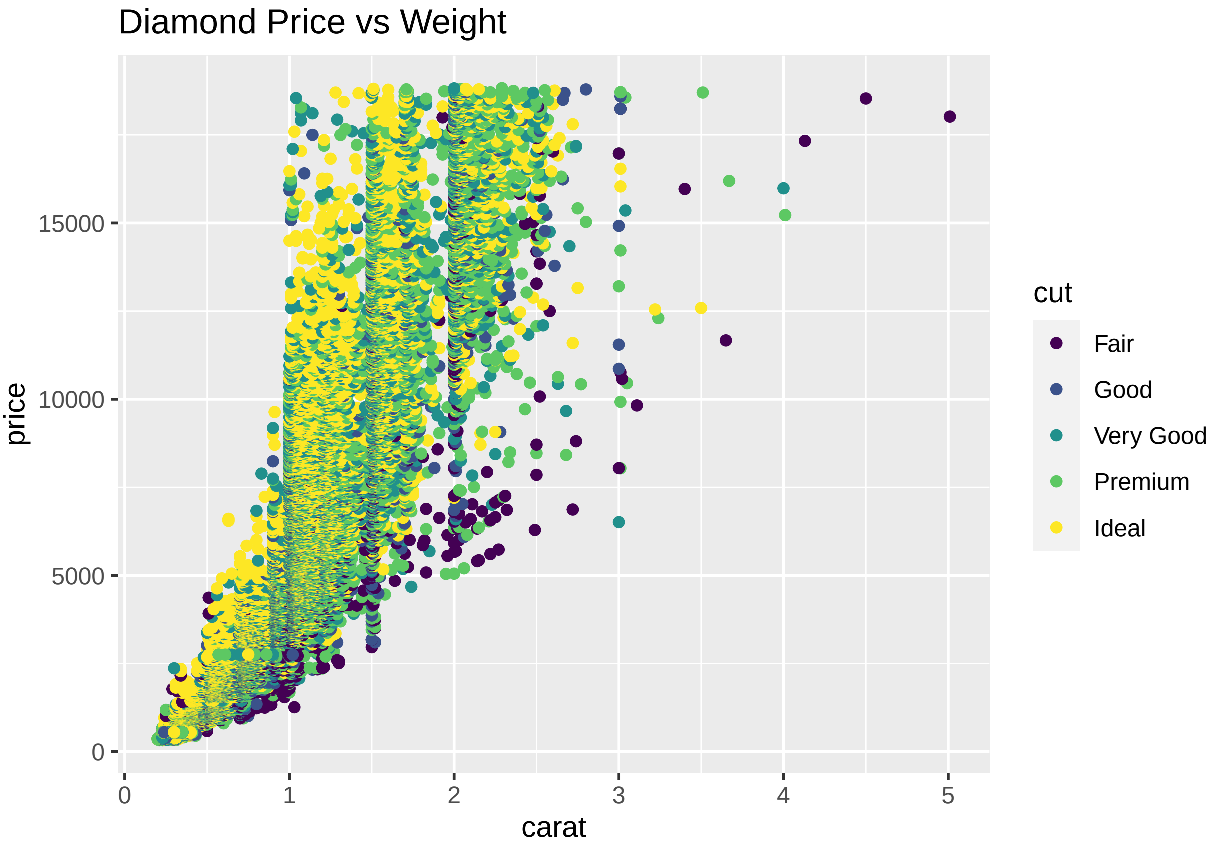
Section 3.2 Exercises

Exercise 1: Indicate teh visual cues, coordinate system, scales, and context is provided

a)

Results:

```
library(ggplot2)
ggplot(data = diamonds) +
  geom_point(mapping = aes(x = carat, y = price, color = cut)) +
  ggtitle("Diamond Price vs Weight")
```



Visual Cues: Position, Color

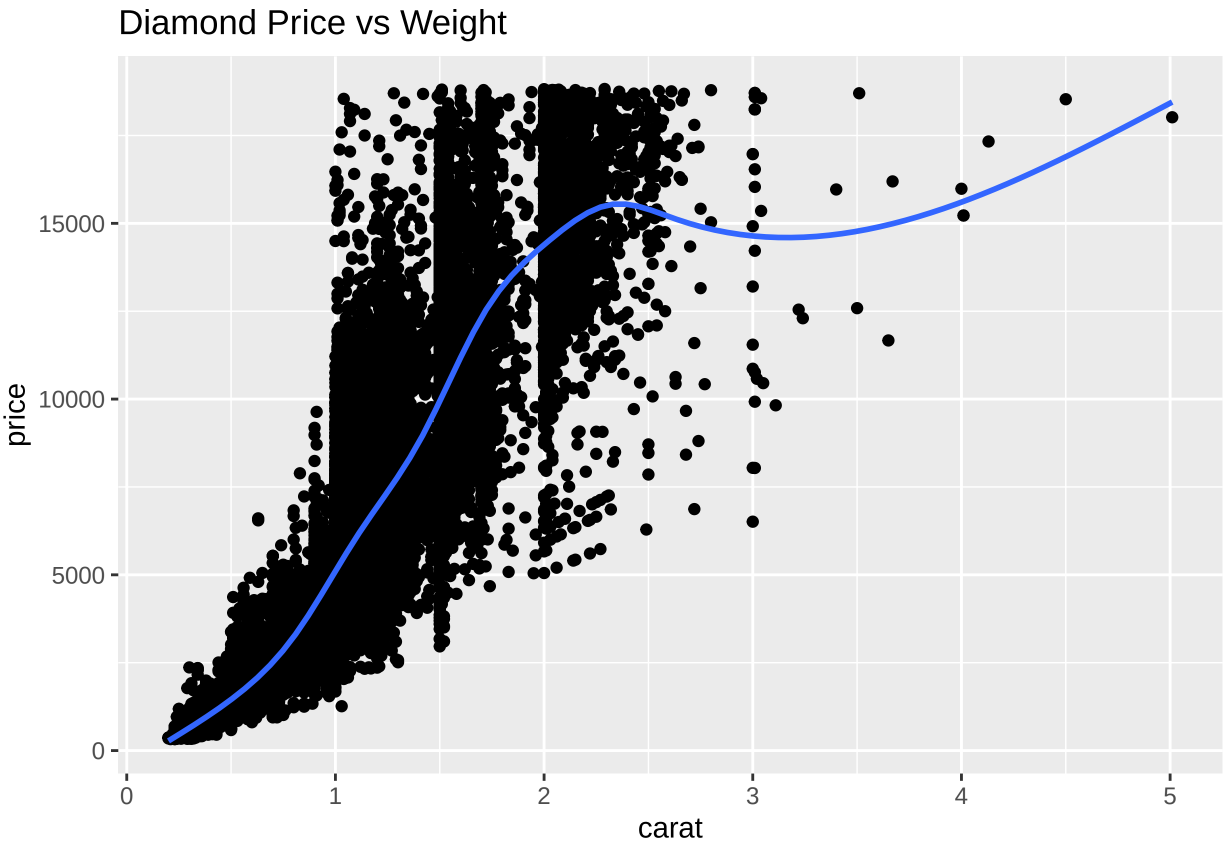
Coordinate System: Cartesian Coordinate System
Scales: x is numerical, Y is numerical, and a categorical scale
Context: X axis label, Y axis label, title, legend

b)

Results:

```
ggplot(data = diamonds) +
  geom_point(mapping = aes(x = carat, y = price)) +
  geom_smooth(mapping = aes(x = carat, y = price), se = FALSE)    +
  ggtitle("Diamond Price vs Weight")
```

```
## ‘geom_smooth()‘ using method = ’gam’ and formula ’y ~ s(x, bs = "cs")’
```



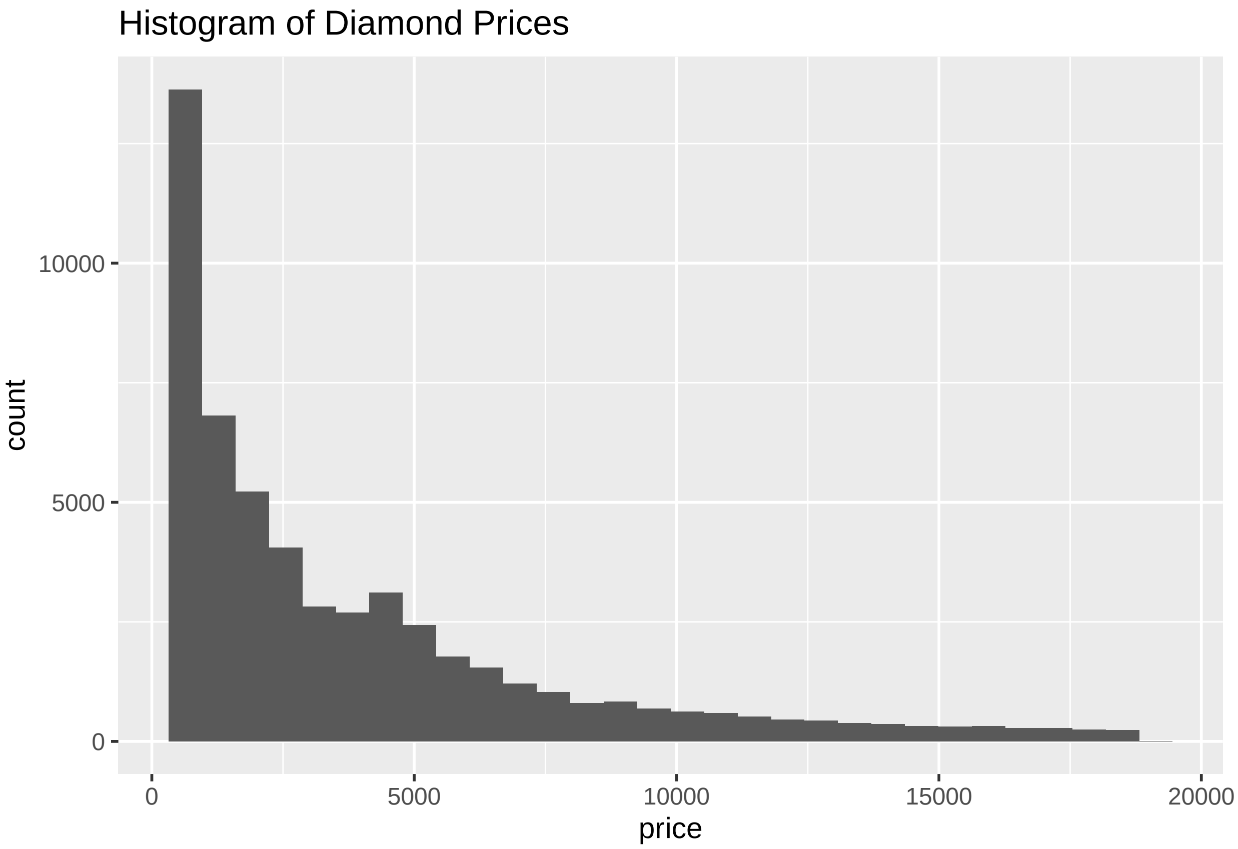
Visual Cues: Position, Direction
Coordinate System: Cartesian Coordinate System
Scales: Numerical
Context: X axis label, Y axis label, Title

c)

Results:

```r
ggplot(data = diamonds) +
geom_histogram(mapping = aes(x = price)) +
ggtitle("Histogram of Diamond Prices")
```

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

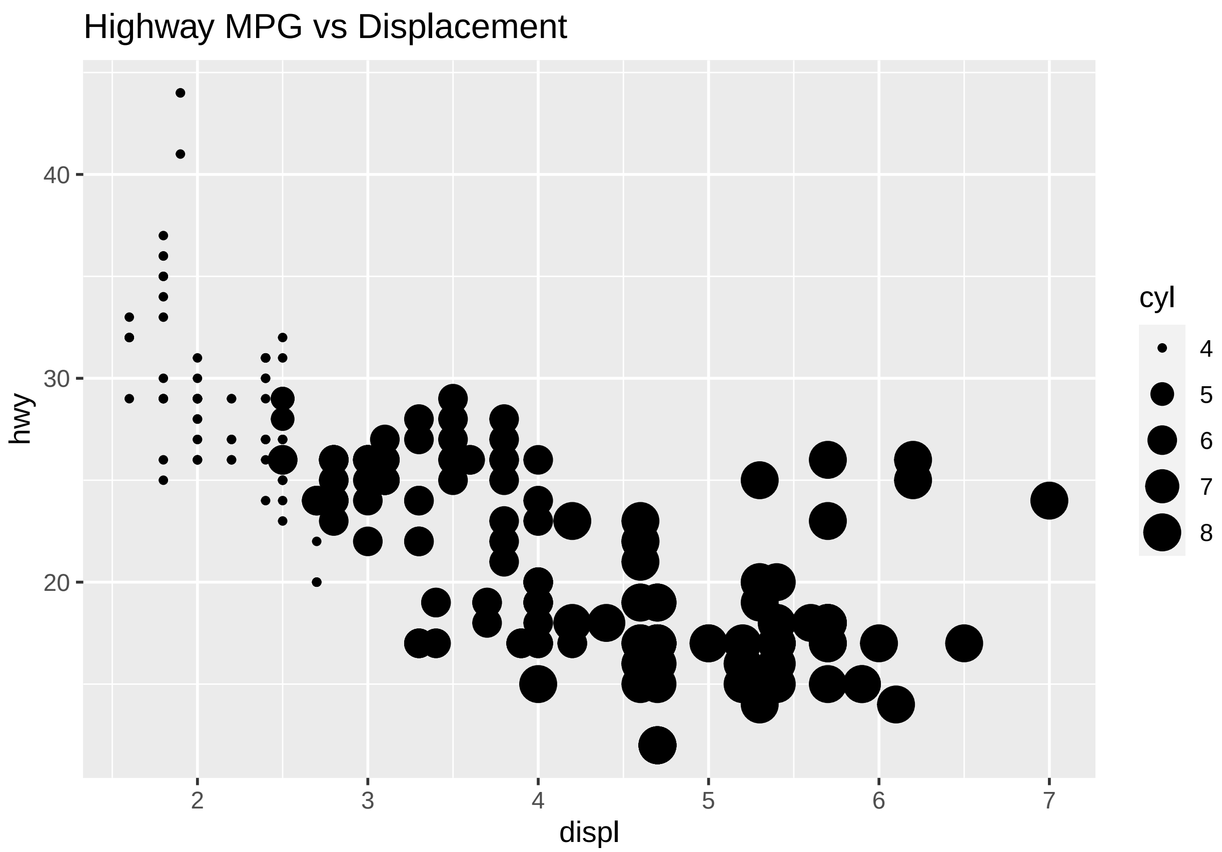


Visual Cues: Area or length
Coordinate System: Cartesian Coordinate System
Scale: X, Y axis are Numerical
Context: X axis label, Y axis label, Title

Exercise 2: Indicate the visual cues, coordinate system, scales, and context.

a)
Results:

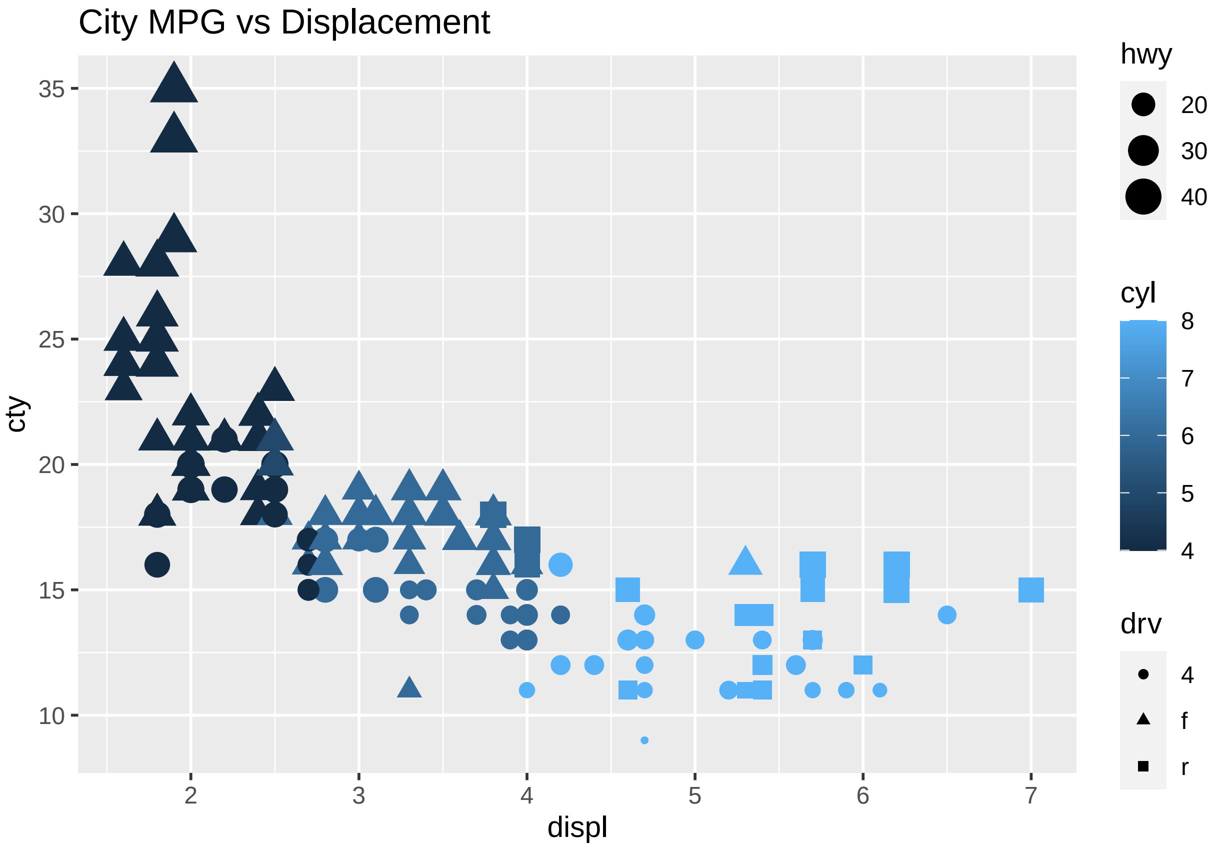
```r
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, size = cyl)) +
  ggtitle("Highway MPG vs Displacement")
```



Visual Cues: Position, Area
Coordinate System: Cartesian Coordinate System
Scales: X, Y, Area are numerical
Context: legend, x and Y axis labels, Title

b)
Results:

```
ggplot(data = mpg) +
  geom_point(mapping = aes(
    x = displ,
    y = cty,
    size = hwy,
    color = cyl,
    shape = drv
  )) +
  ggtitle("City MPG vs Displacement")
```



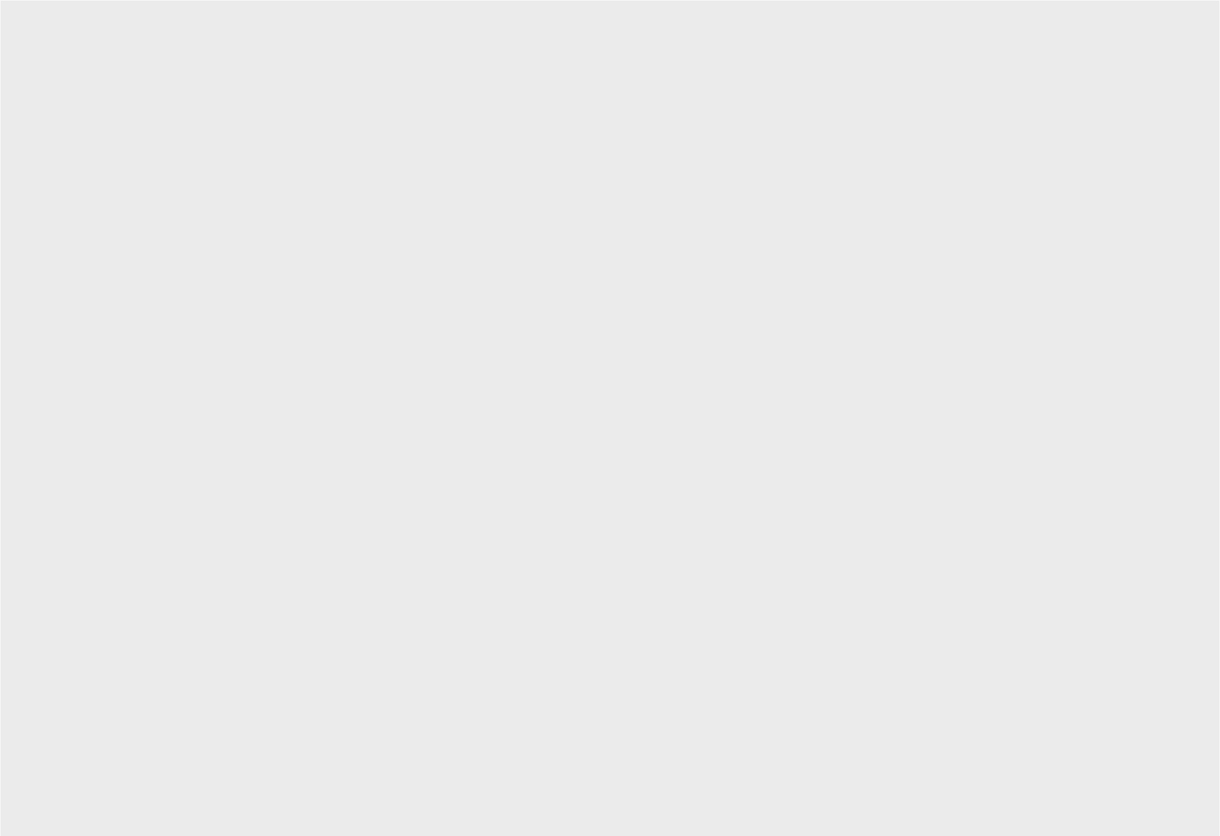
Visual Cues: Shapes, Position, Shade, Area
Coordinate System: Cartesian Coordinate System
Scales: X and Y axis are numerical, hwy is numerical, cyl is numerical, and drive terrain is categorical
Context: X and Y labels, Title, Legend

Section 4.1 Exercises

Exercise 3: Describe what is seen

Results:

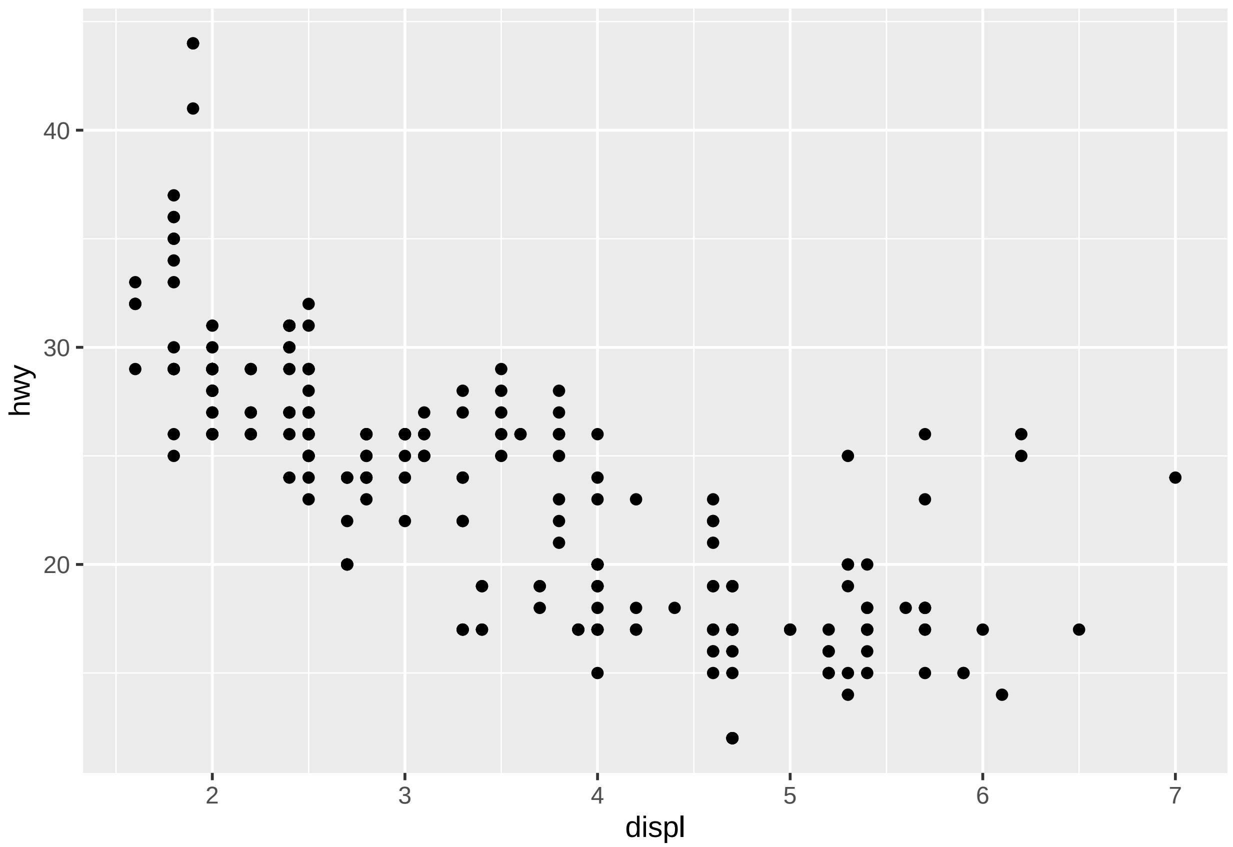
```
ggplot(data = mpg)
```



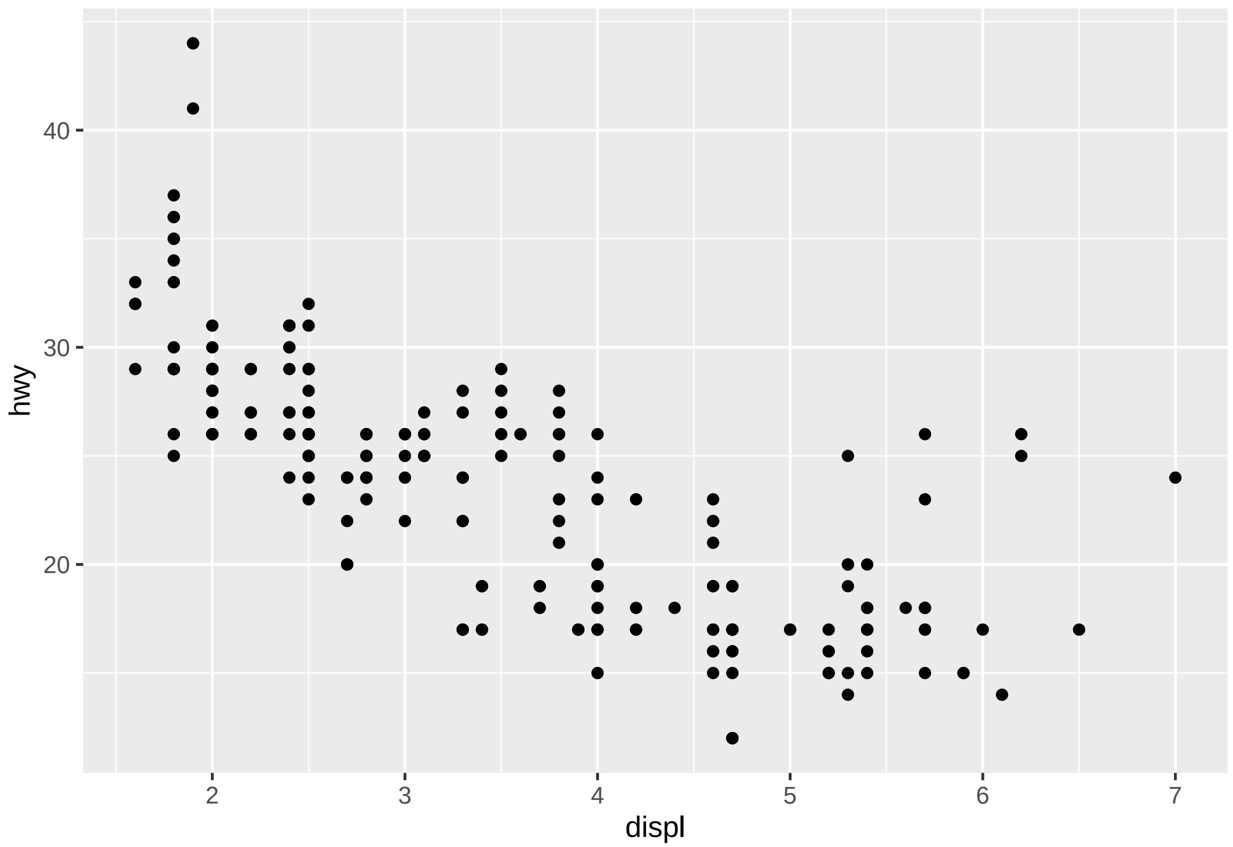
No graph is showing because it is missing some parameters that are needed.

Exercise 4: Guess if both scatter plots make the same scatterplot.
Results:

```
## Specify data in ggplot():
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```



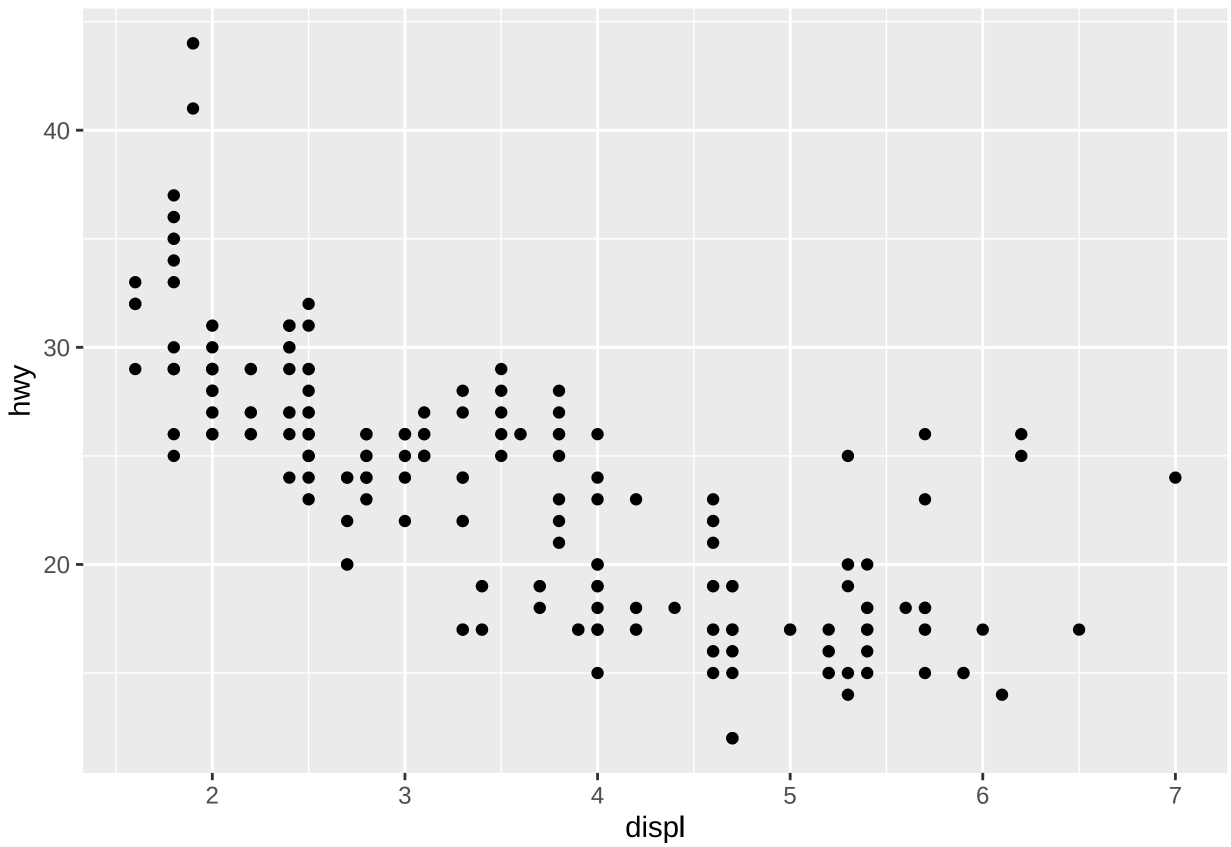
```
## Specify data in geom_*() function:
ggplot() +
  geom_point(data = mpg, mapping = aes(x = displ, y = hwy))
```



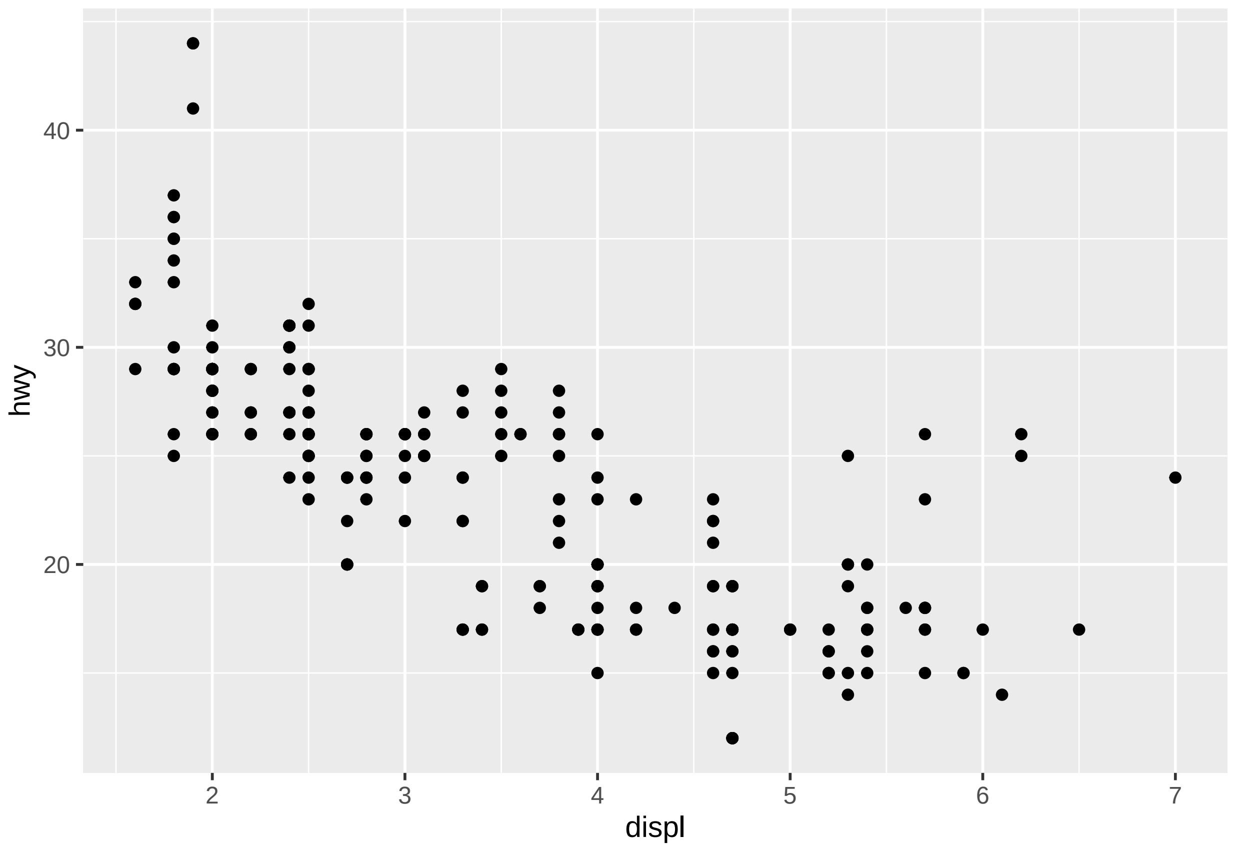
Both will make the same scatter plot.

Exercise 5: Guess if both scatter plots make the same graph.
Result:

```
## Specify aesthetics in geom_*() function:
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```



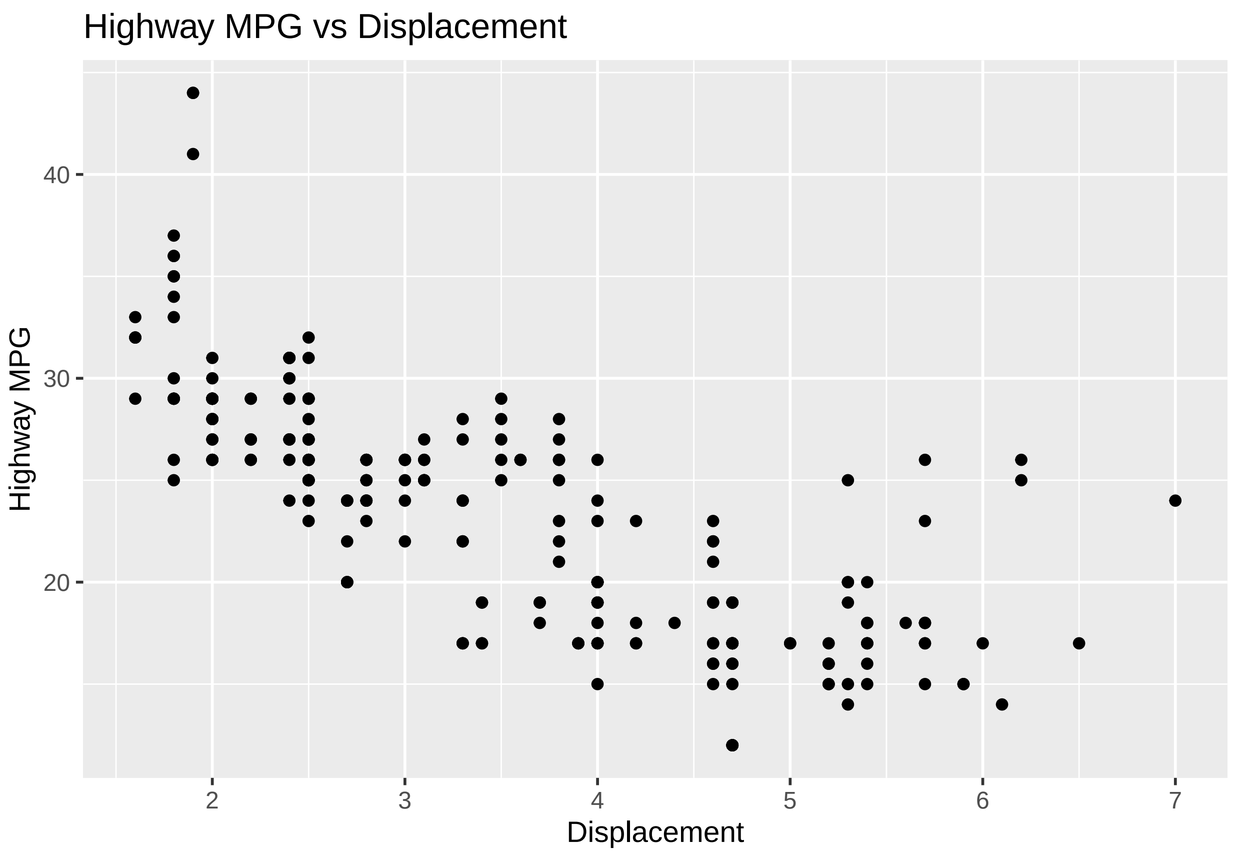
```
## Specify aesthetics in ggplot():
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point()
```



Both commands will make the same scatter plot.

Exercise 6: Guess what the ggtitle(), xlab(), and ylab() commands do.
Results:

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  ggtitle(label = "Highway MPG vs Displacement") +
  xlab(label = "Displacement") +
  ylab(label = "Highway MPG")
```

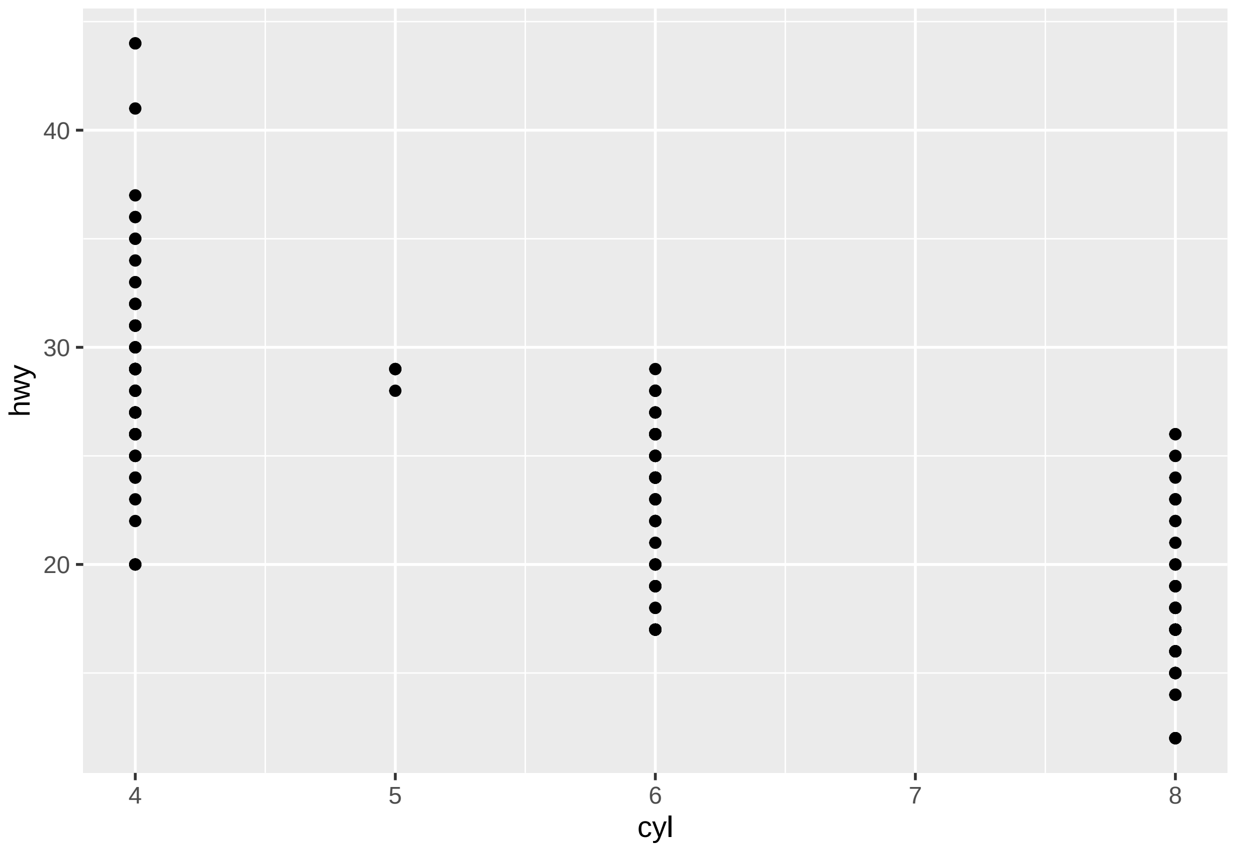


Title: Highway MPG VS Displacement
X Label: Displacement
Y Label: Highway MPG

Exercise 7: Do the following

a) Make a scatter plot and show R commands. Results:

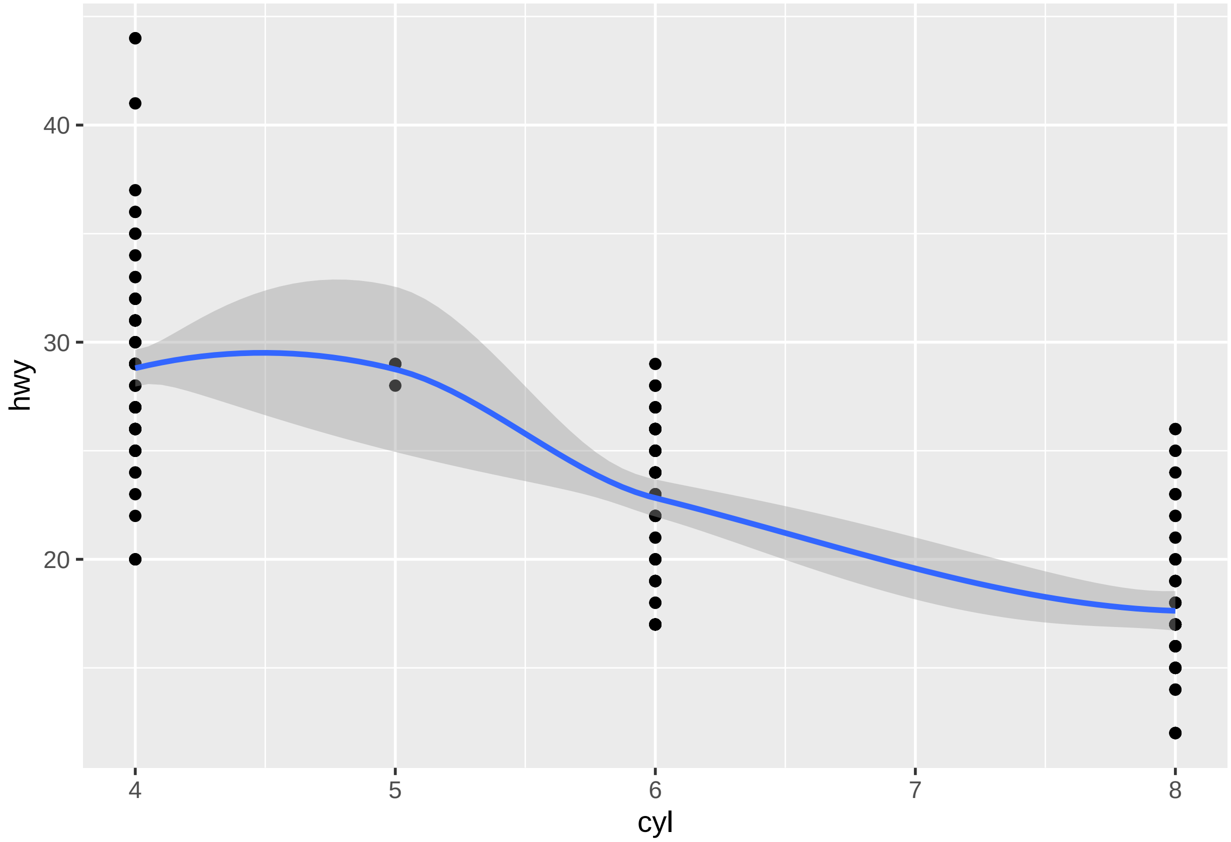
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = cyl, y = hwy))
```



b) Add geom_smooth() to scatter plot made in part *a* Results:

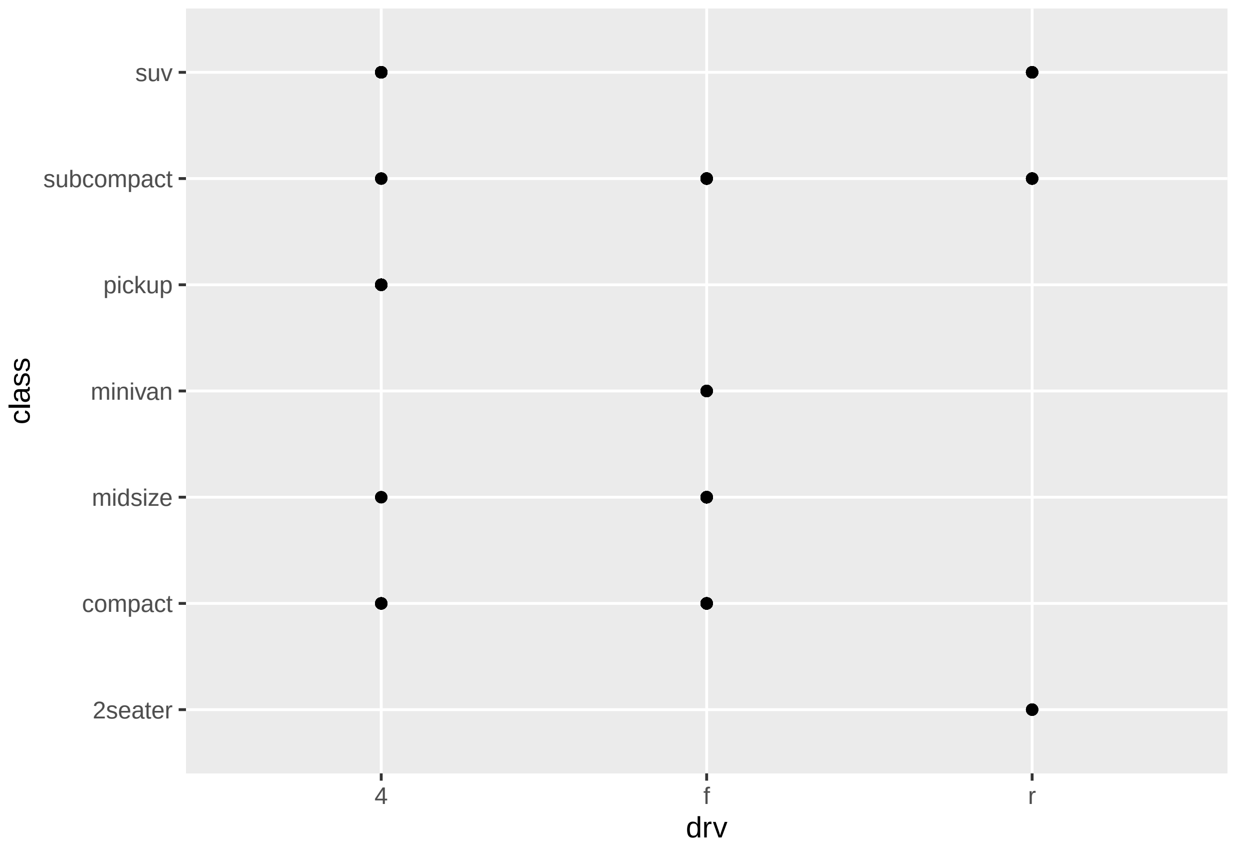
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = cyl, y = hwy)) +
  geom_smooth(mapping = aes(x = cyl, y = hwy))
```

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

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 6

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 2

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 3.2687e-015

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at 6

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius 2
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition
## number 3.2687e-015
```



c) Make a scatterplot of class vs drv. What happens? Why is it not useful?
Results:

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = drv, y = class))
```



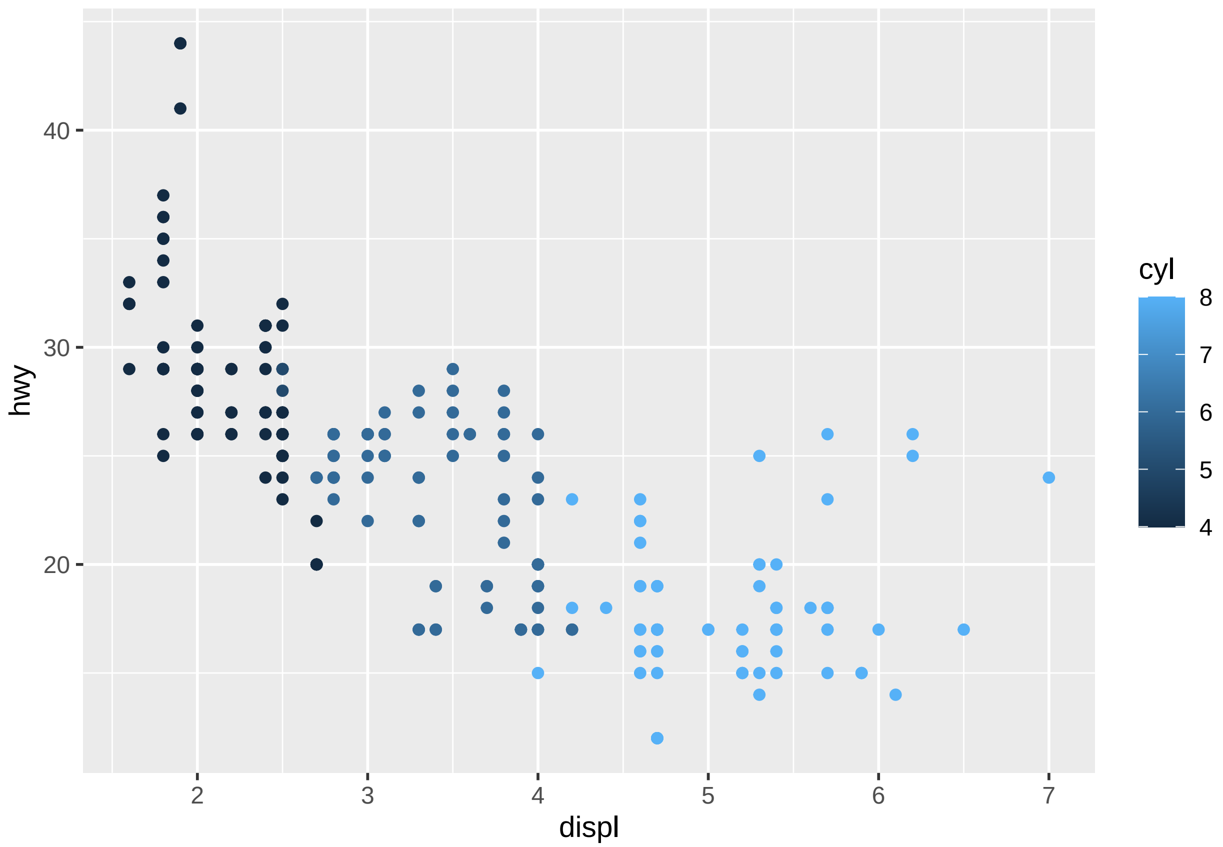
The plot is not useful because it just tells what type of drive terrains (4 wheel drive, rear wheel drive, front wheel drive) is available to different types of cars. This is particularly useful in analyzing the differences between car types. Both the x axis and y axis are categorical so the information isn't useful.

Exercise 4.2 Exercises

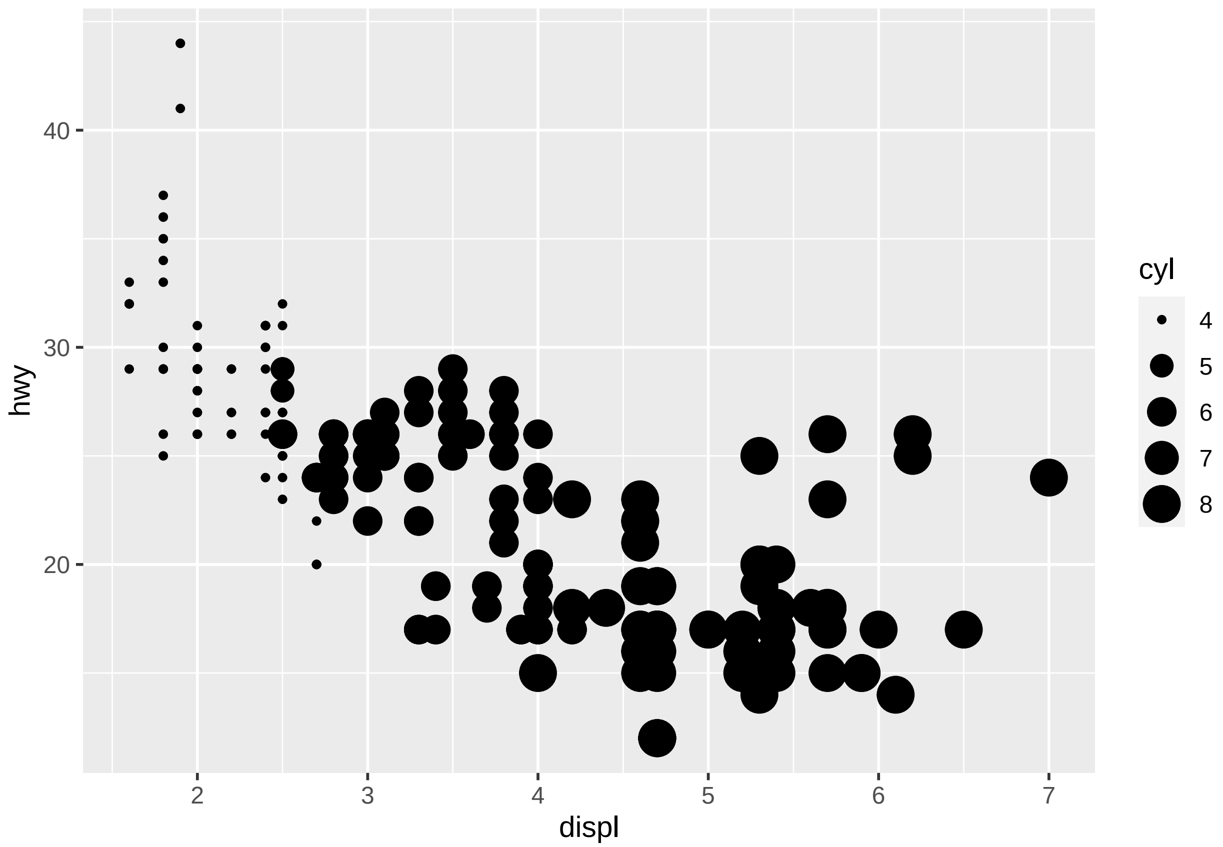
Exercise 8: Do the following

a) How does the plot differ when cyl is mapped to size instead of color?
Results:

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



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

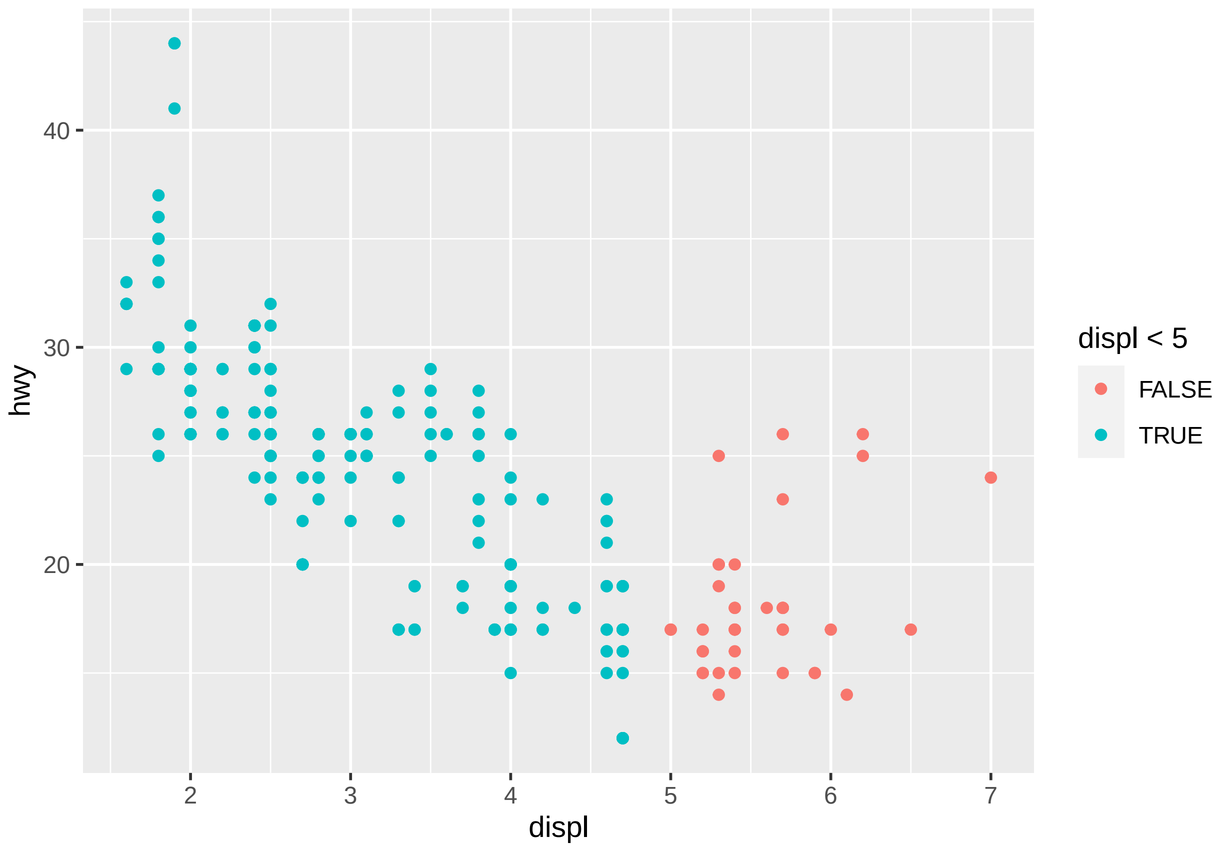


The plot turns different when cyl changes from color to size. As the number of cylinders increases, the area also increases whereas with color the area stayed the same but changed to a different shade of blue.
##### b) What happens when cyl is mapped to shape?
Results:

```
#ggplot(data = mpg) +
#  geom_point(mapping = aes(x = displ, y = hwy, shape = cyl))
```

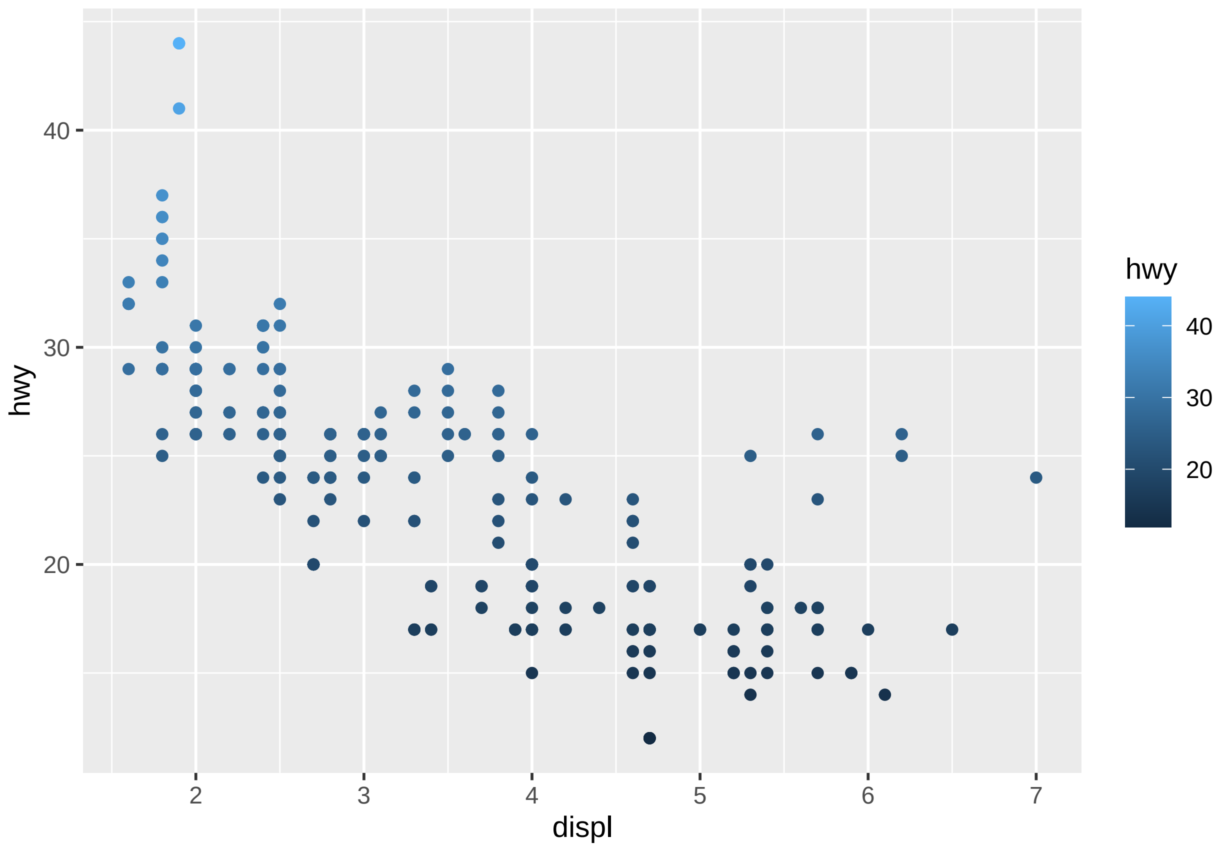
When cyl is mapped to shape, this produces an error:
"Error in `scale_f()`: ! A continuous variable can not be mapped to shape Backtrace: 1. base `<fn>`(x) 2. ggplot2:::print.ggplot(x) 4. ggplot2:::ggplot_build.ggplot(x) 5. ggplot2 by_layer(function(l, d) l$compute_aesthetics(d, plot))6.ggplot2f(l = layers[[i]], d = data[[i]])$7.lcompute_aesthetics(d, plot) 8. ggplot2 f(…, self = self) 9. ggplot2:::scales_add_defaults(…) 12. ggplot2:::find_scale(aes, datacols[[aes]], env) 13. ggplot2 scale_f()".
#### c) Run the command below and describe what happens.
Results:

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = displ < 5))
```



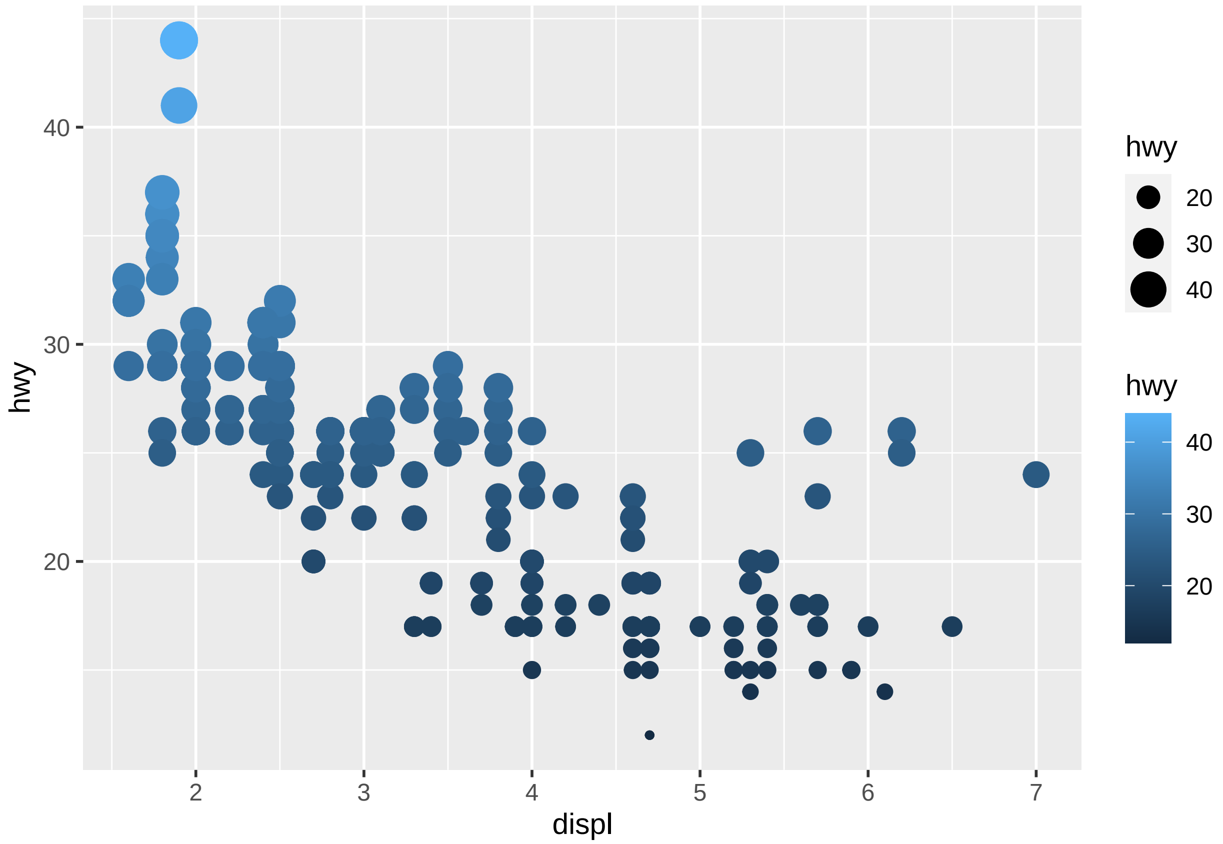
The plot will show two different colors depending on the value of the displacement variable. The blue represents them as true and the red will display false.
#### d) Describe what happens to the following.
* Command for mapping hwy to both y and color

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



The higher the number for highway, the lighter shade of blue the dots will present.
* Command for mapping hwy to y, color, and size

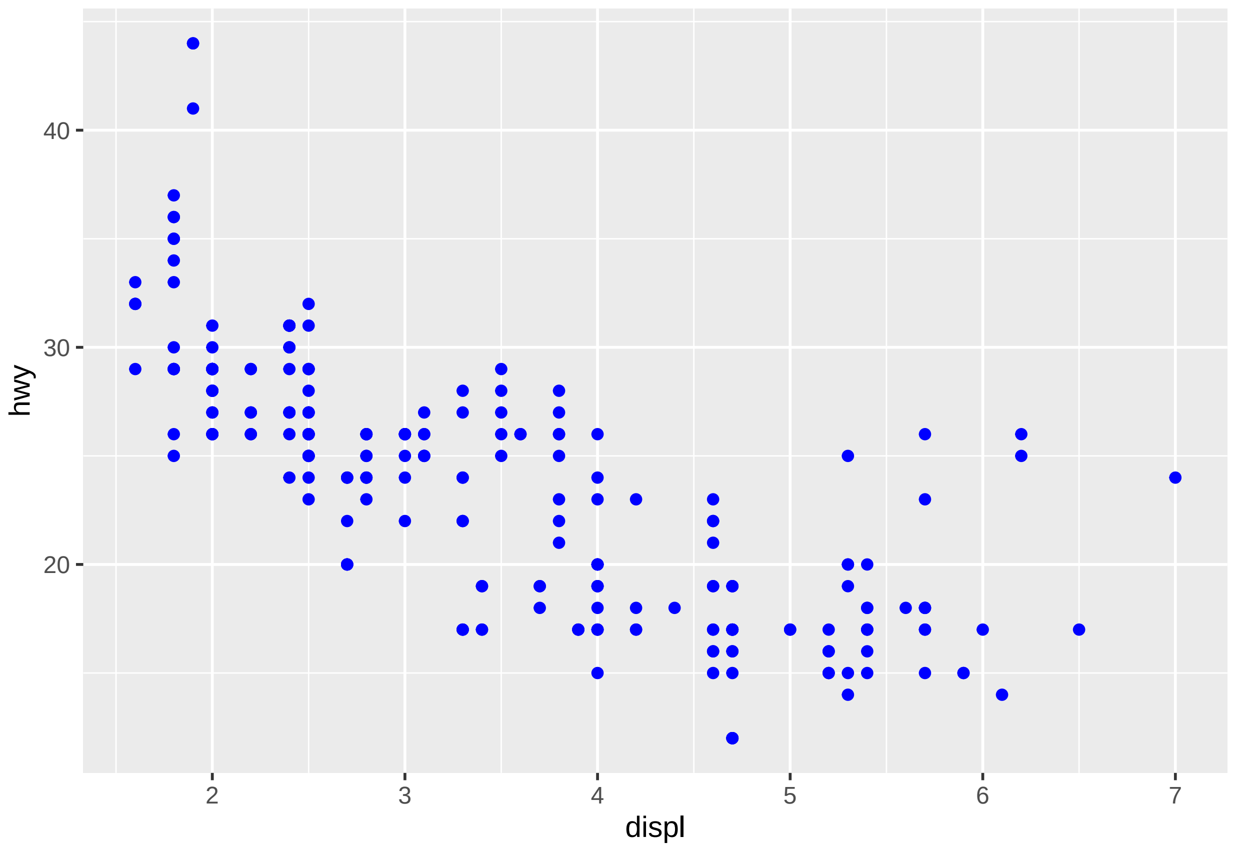
```
ggplot(data = mpg) +
  geom_point(mapping = aes(
    x = displ,
    y = hwy,
    color = hwy,
    size = hwy
  ))
```



This graph will increase the size of the dots based on the variable highway and the shade will also change based on the highway variable.

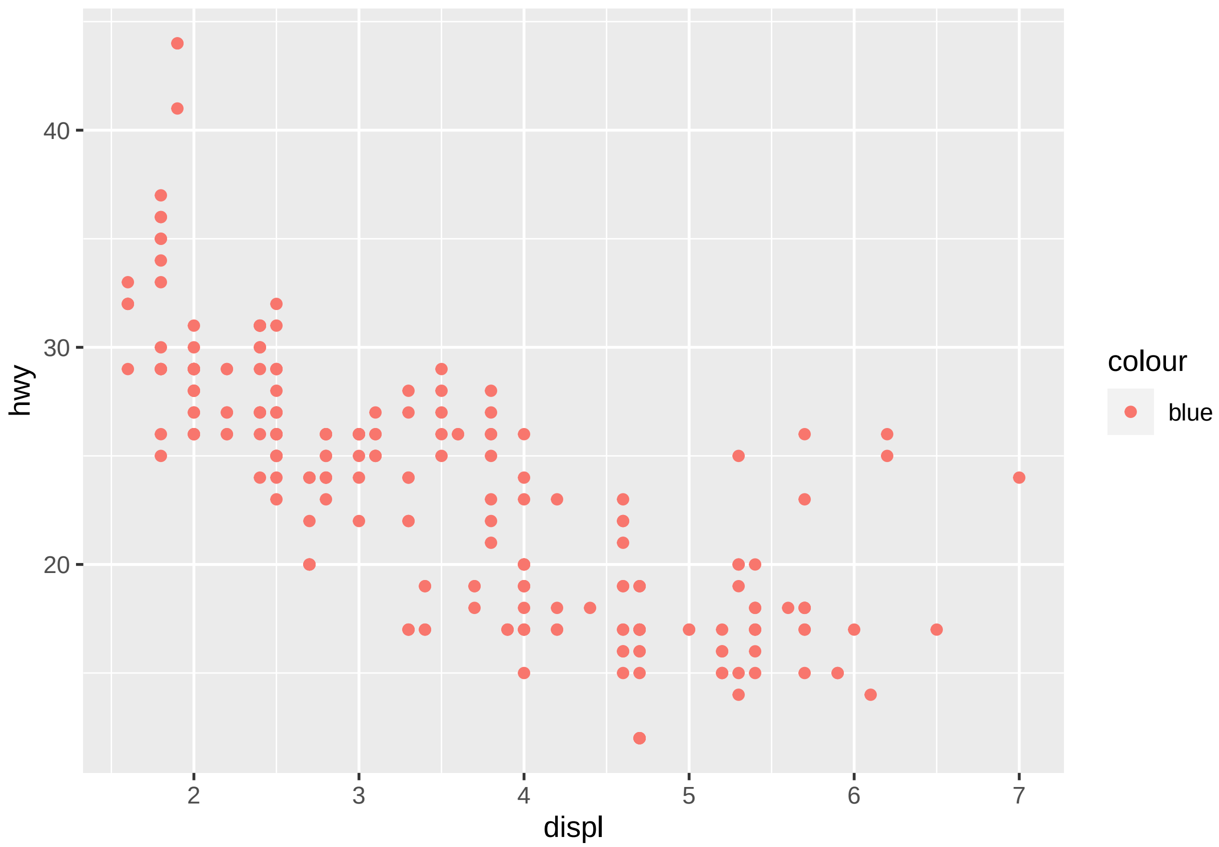
Exercise 9: The following code changes the color of the points in the graph to blue.

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), color = "blue")
```



What happens in the following code?

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, color = "blue"))
```



If color = blue is inside the aes, the color will be more of a red instead of blue.

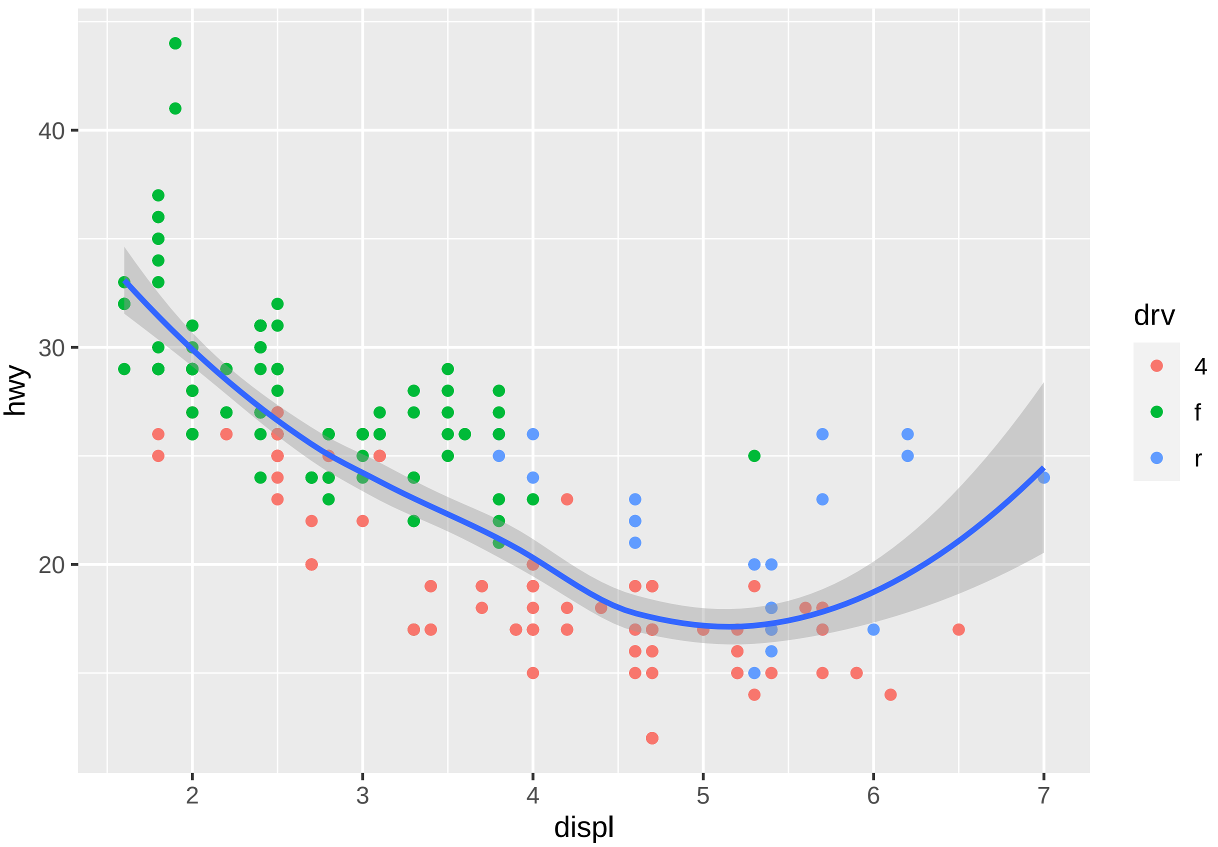
Section 4.3 Exercises

Exercise 10: Do the following.

a) Try to predict what the following graph will look like, then check your answer. Report your findings.

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = drv)) +
  geom_smooth()
```

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

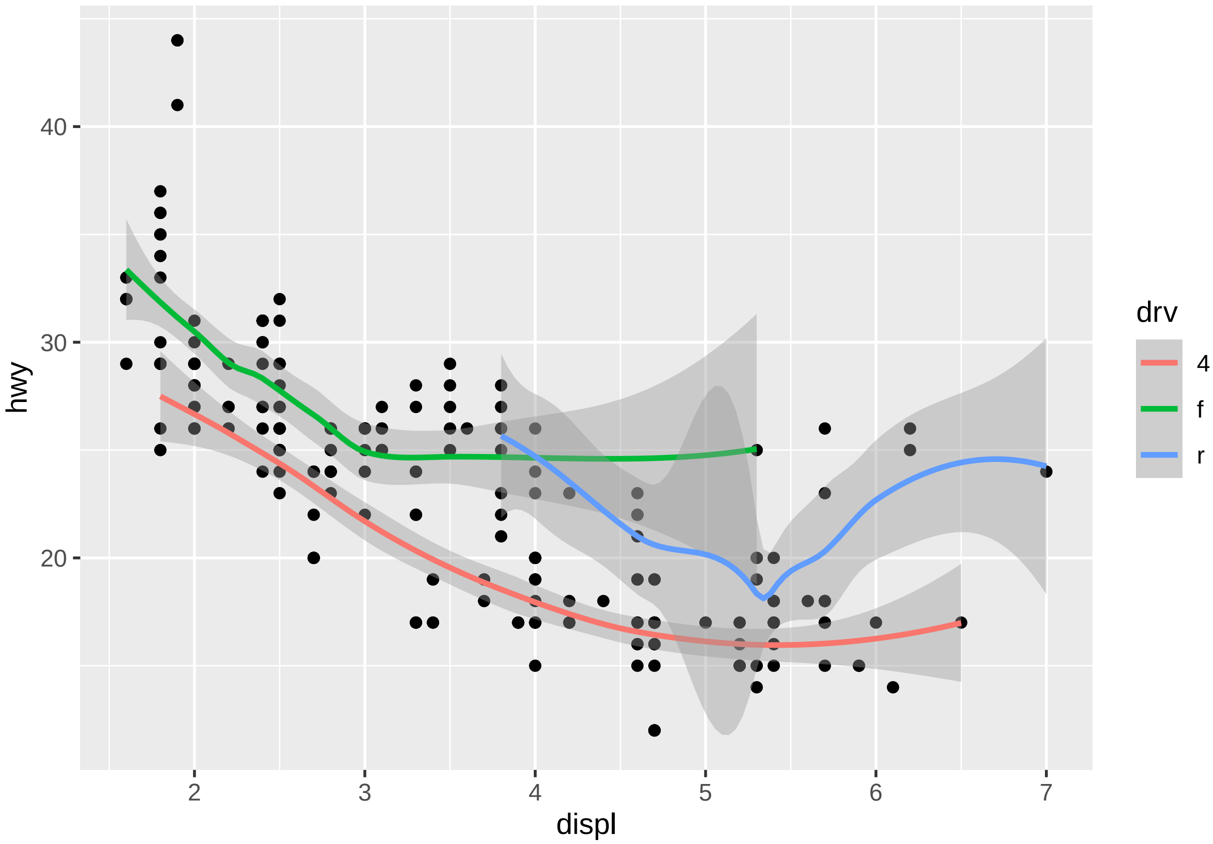


I believe the code above will have different colors to represent each drive terrain type. Then, the graph will show the generic trend line of all drive terrain types.

b) Now try to predict what the following graph will look like, then check your answer. Report your findings.

```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth(mapping = aes(color = drv))
```

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

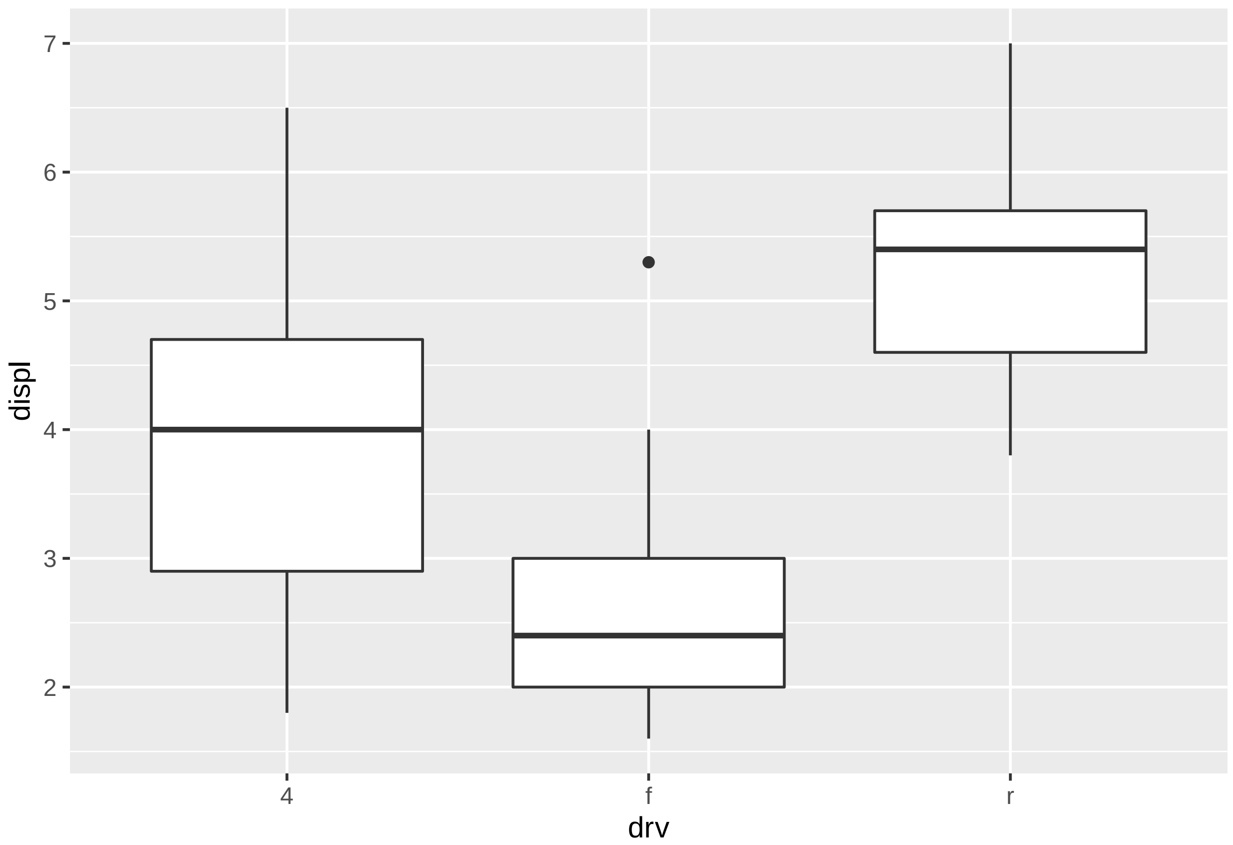


This graph will show the trends of each drive terrain type in different colors. The points are just going to show a black point instead of a color.

Exercise 11: Do the following:

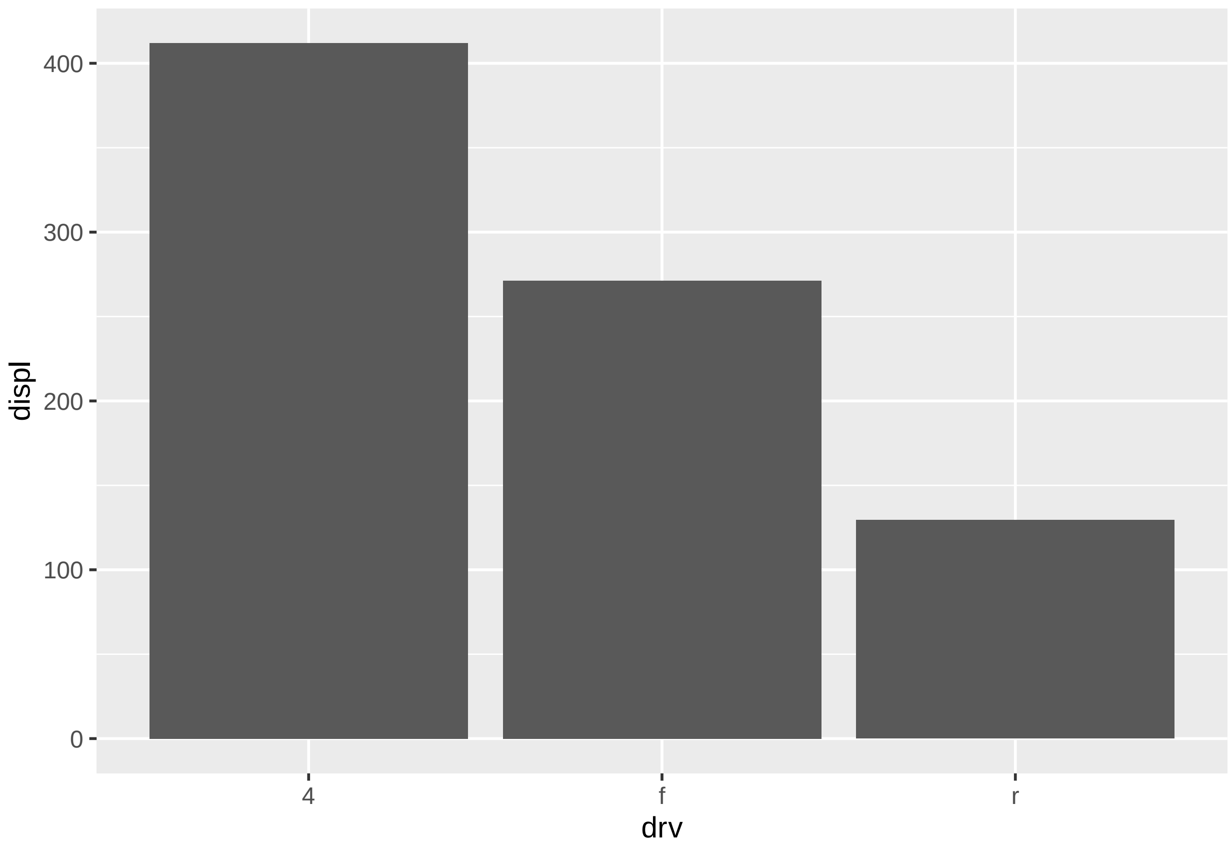
#### a) Modify the command so that the following layer is added to the plot. Report what you see.

```
ggplot(data = mpg) + geom_boxplot(mapping = aes(x = drv, y = displ))
```



#### b) What happens if you use geom_col() in place of geom_boxplot() in Part b? Try it and report what you see.

```
ggplot(data = mpg) + geom_col(mapping = aes(x = drv, y = displ))
```

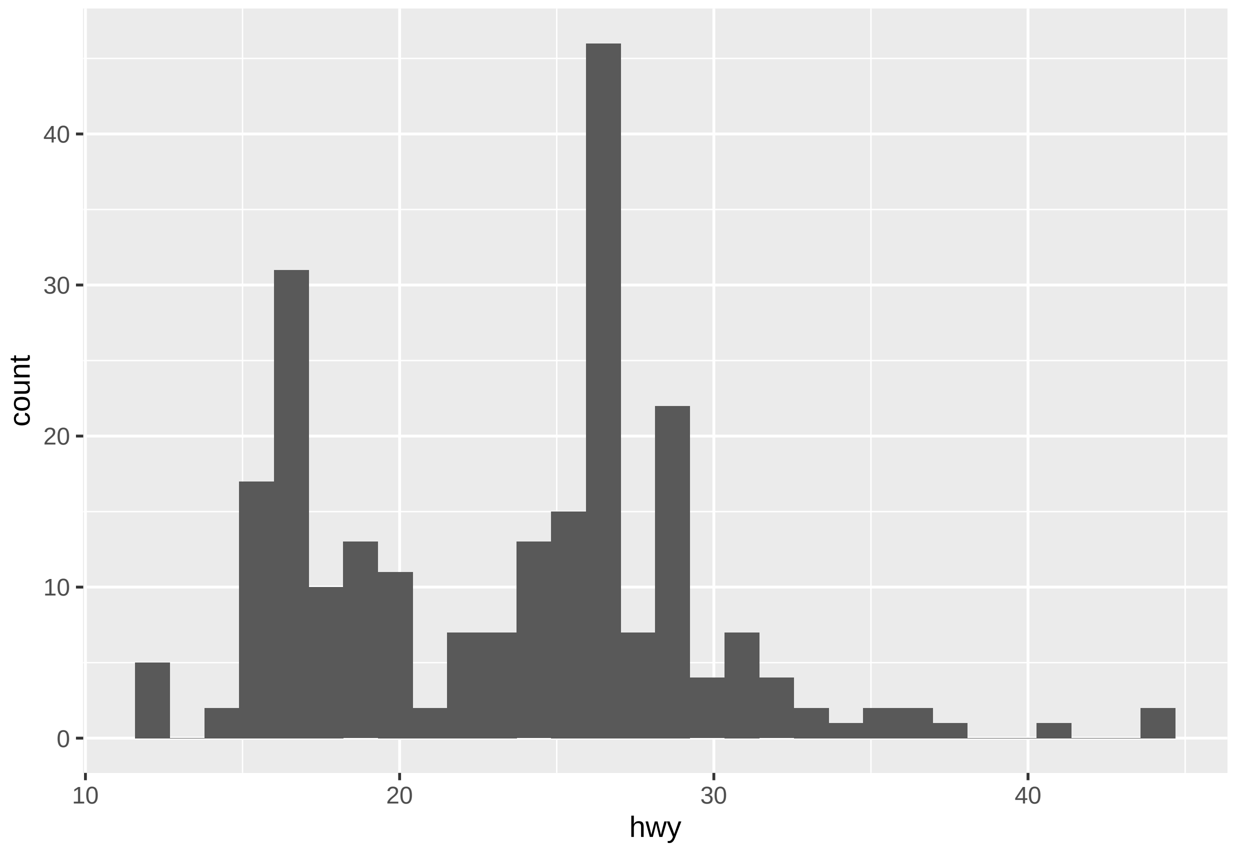


Exercise 12: Do the following.

a) Use ggplot() and geom_histogram() to make a histogram of hwy (from the mpg data set). Report your R command(s).

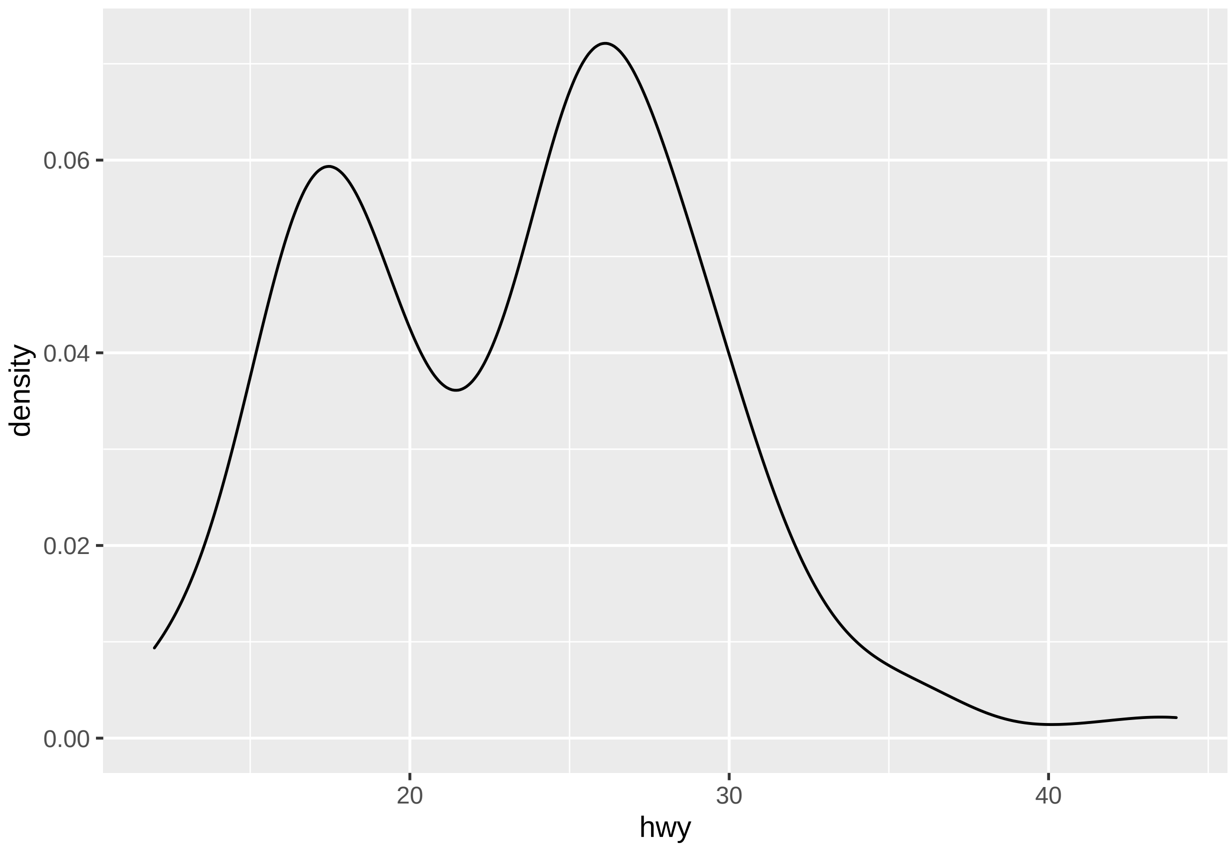
```
ggplot(data = mpg) + geom_histogram(mapping = aes(x = hwy))
```

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



b) Replace geom_histogram() in your command from Part a by geom_density() to make a density plot hwy. Report your R command(s).

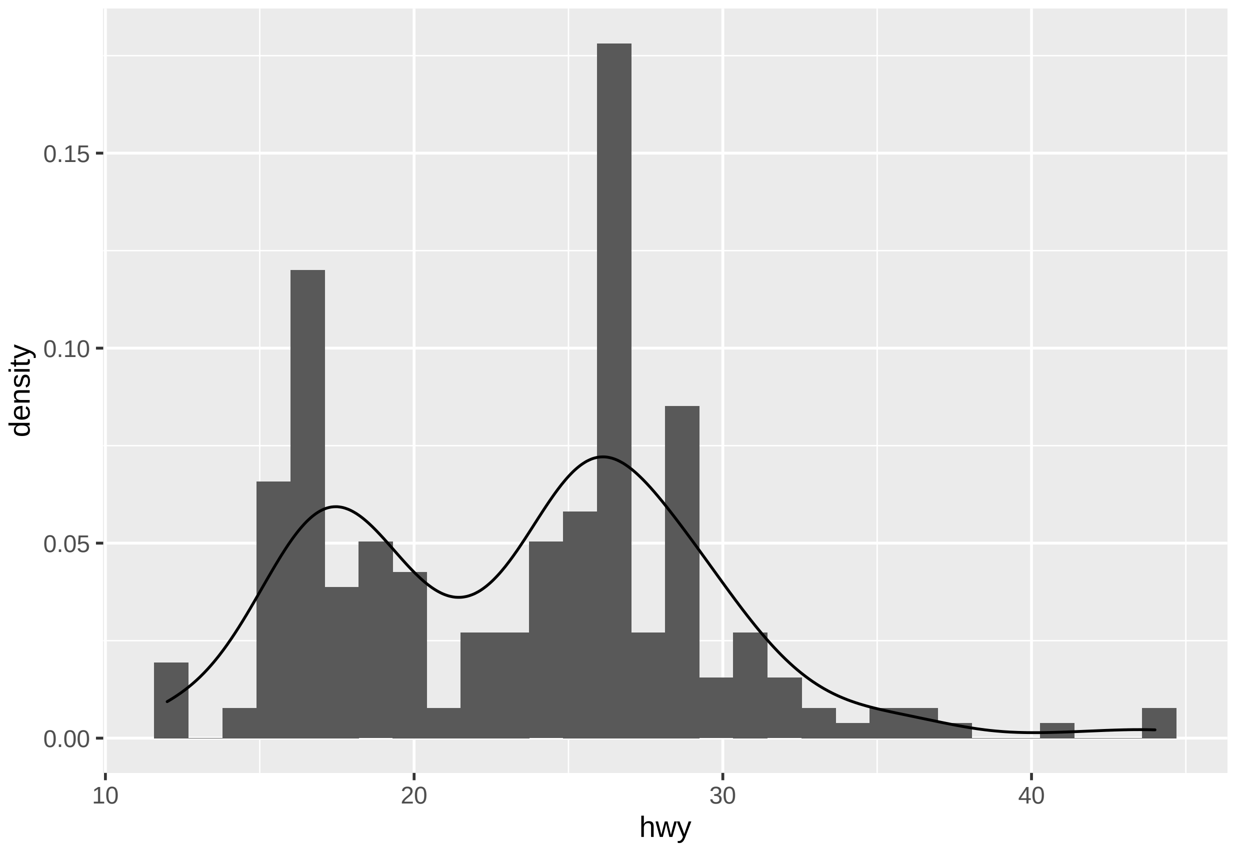
```
ggplot(data = mpg) + geom_density(mapping = aes(x = hwy))
```



#### c) Make the following graph and describe the result:

```
ggplot(mpg, mapping = aes(x = hwy)) +
  geom_histogram(mapping = aes(y = stat(density))) +
  geom_density()
```

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



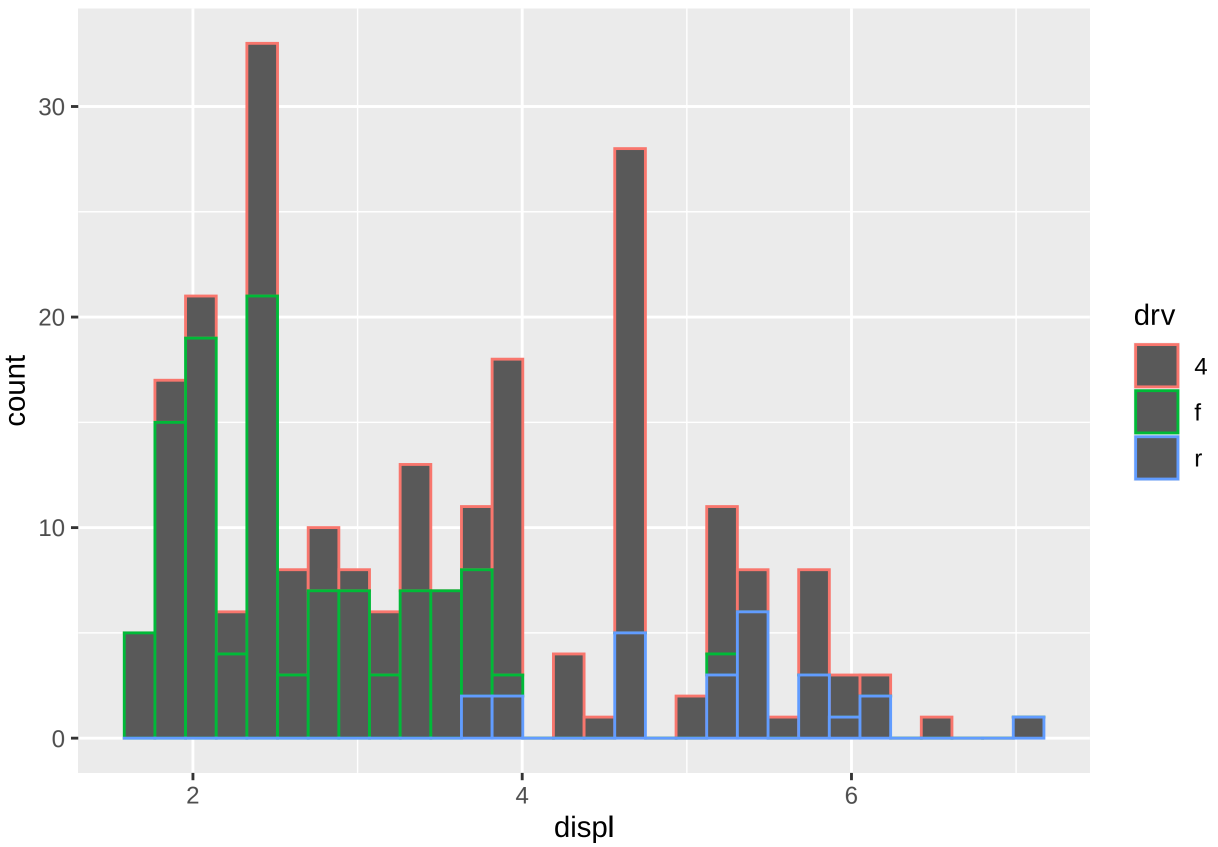
Section 4.4 Exercises

Exercise 13: Do the following.

a) Which graph do you prefer?
Graph 1:

```
ggplot(data = mpg) +
  geom_histogram(mapping = aes(x = displ, color = drv))
```

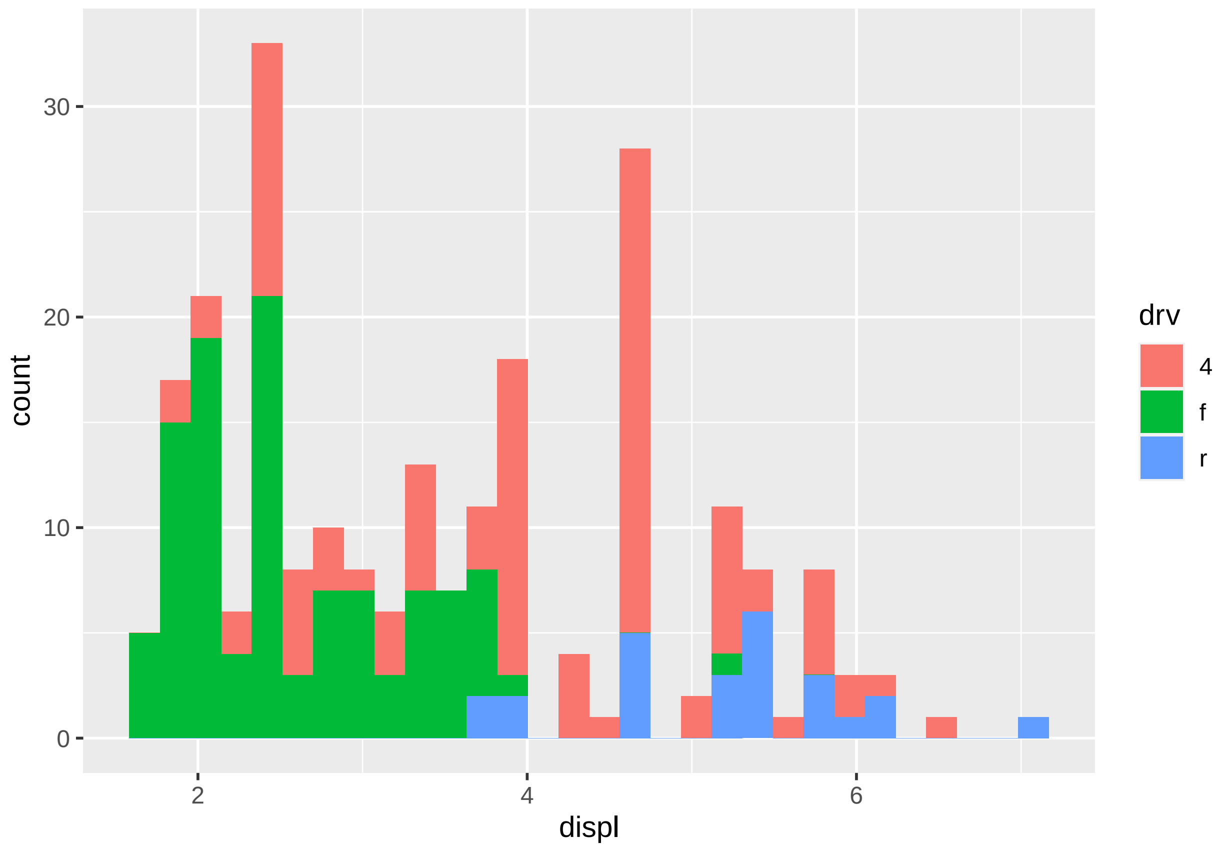
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Graph 2:

```
ggplot(data = mpg) +
  geom_histogram(mapping = aes(x = displ, fill = drv))
```

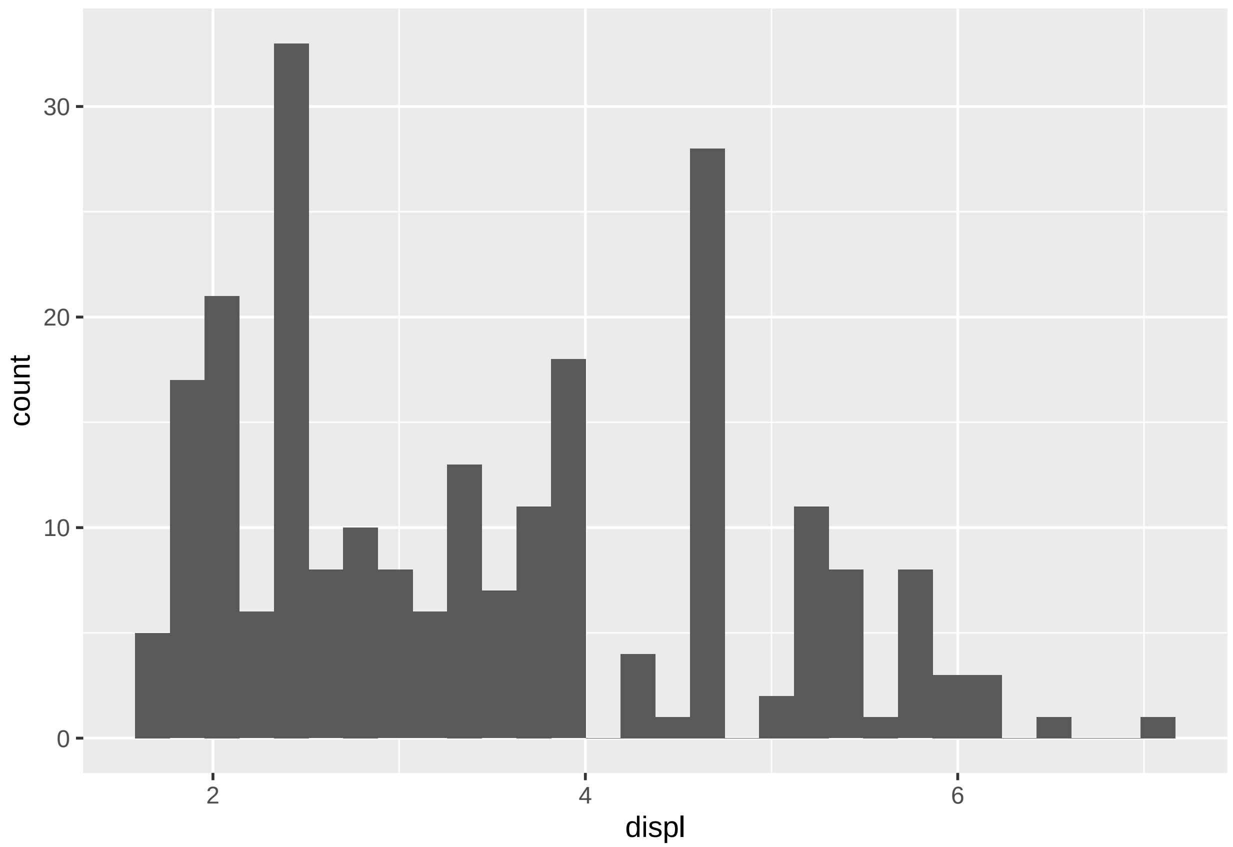
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



I prefer the first graph because it is easier for me to read rather than staring at a variety of different colors. #### b) Alter the code using facet_wrap()
Original Graph:

```
ggplot(data = mpg) +
  geom_histogram(mapping = aes(x = displ))
```

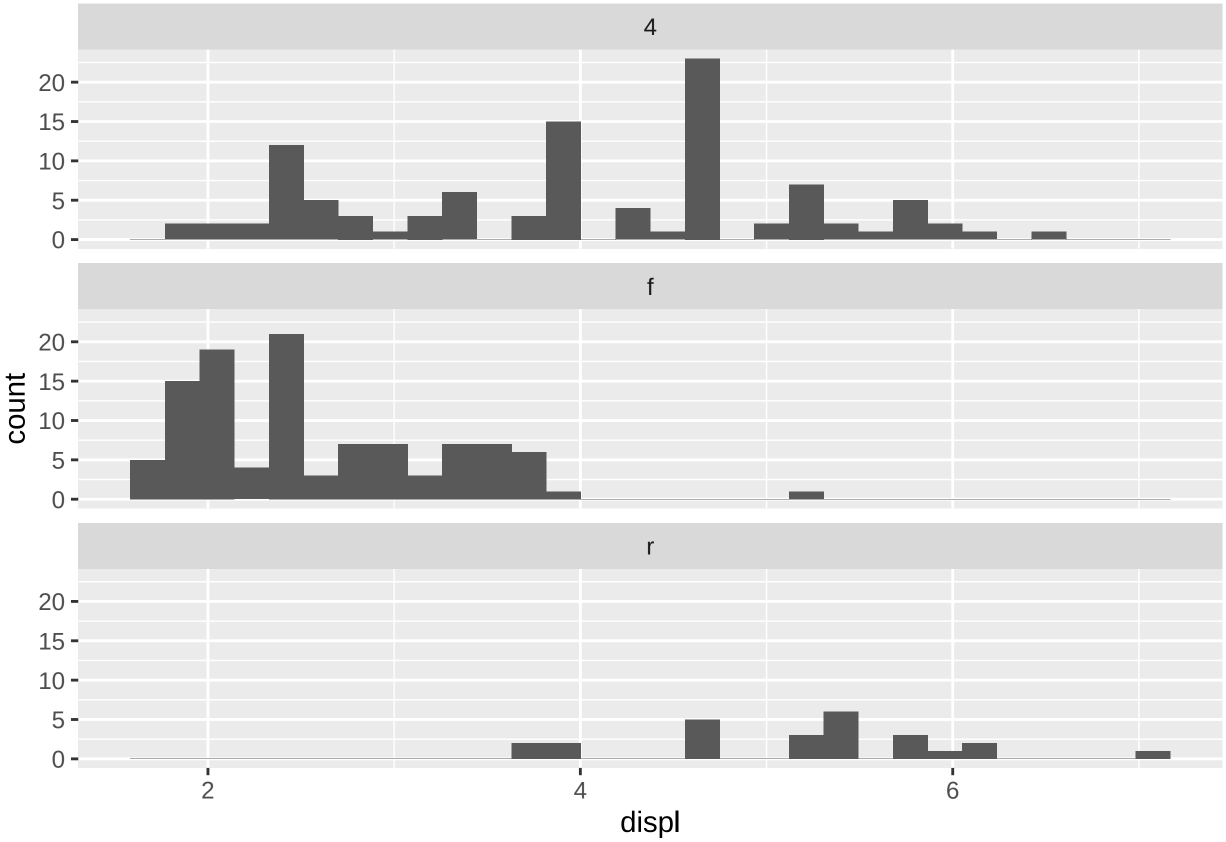
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



Modified Graph:

```
ggplot(data = mpg) +
  geom_histogram(mapping = aes(x = displ)) +
  facet_wrap(facets = ~ drv,
             nrow = 3,
             ncol = 1)
```

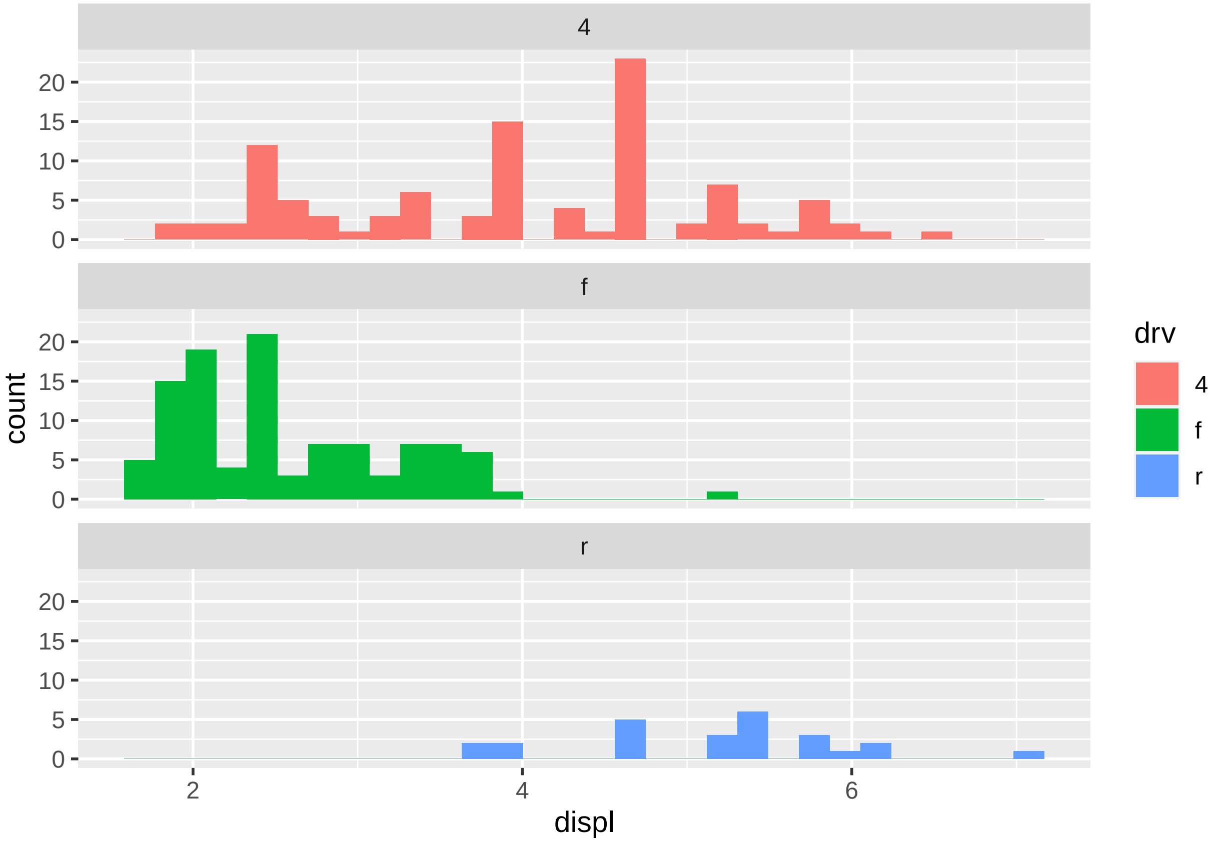
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



#### c) Duplicate Part *b* but use another aesthetic *fill* = *drv*

```
ggplot(data = mpg) +
  geom_histogram(mapping = aes(x = displ, fill = drv)) +
  facet_wrap(facets = ~ drv,
             nrow = 3,
             ncol = 1)
```

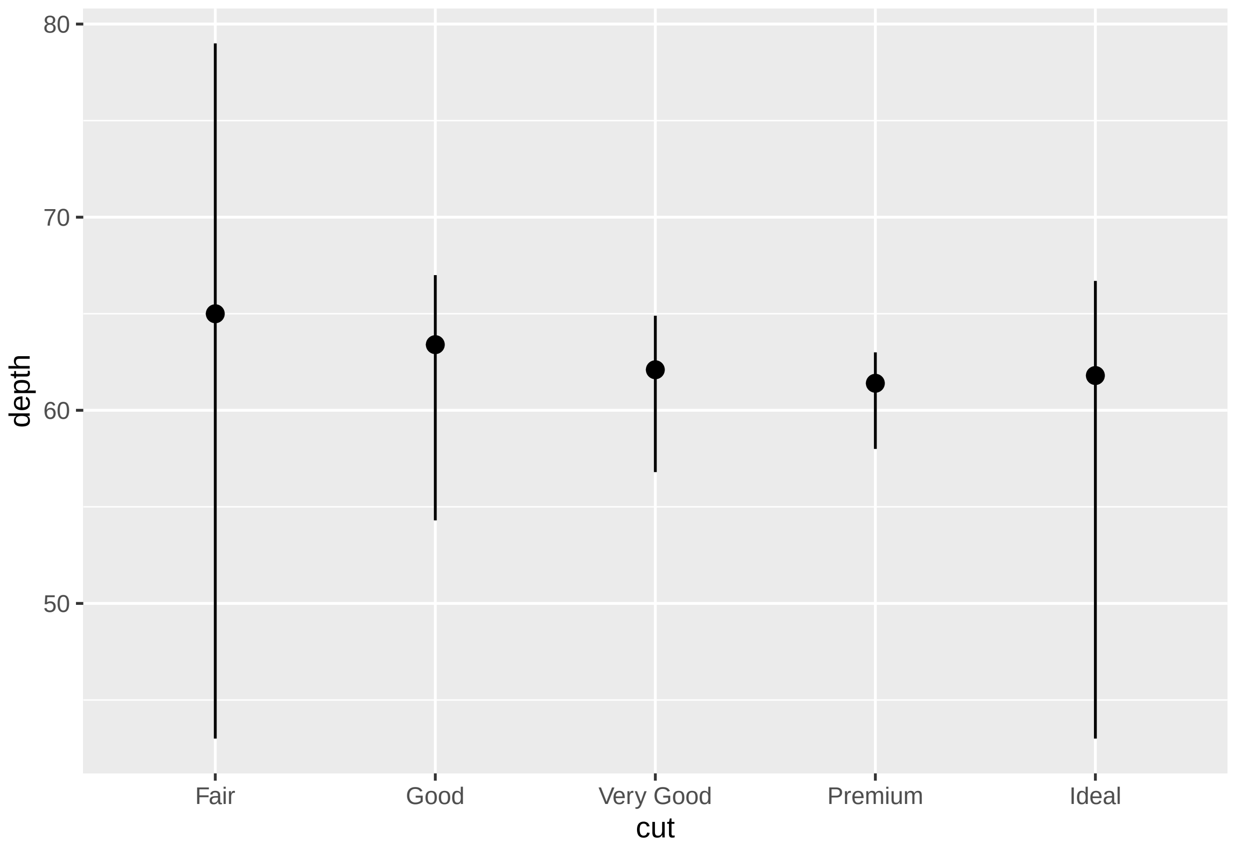
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



## Section 4.5 Exercises ### Exercise 14: Do the following.
Figure Code:

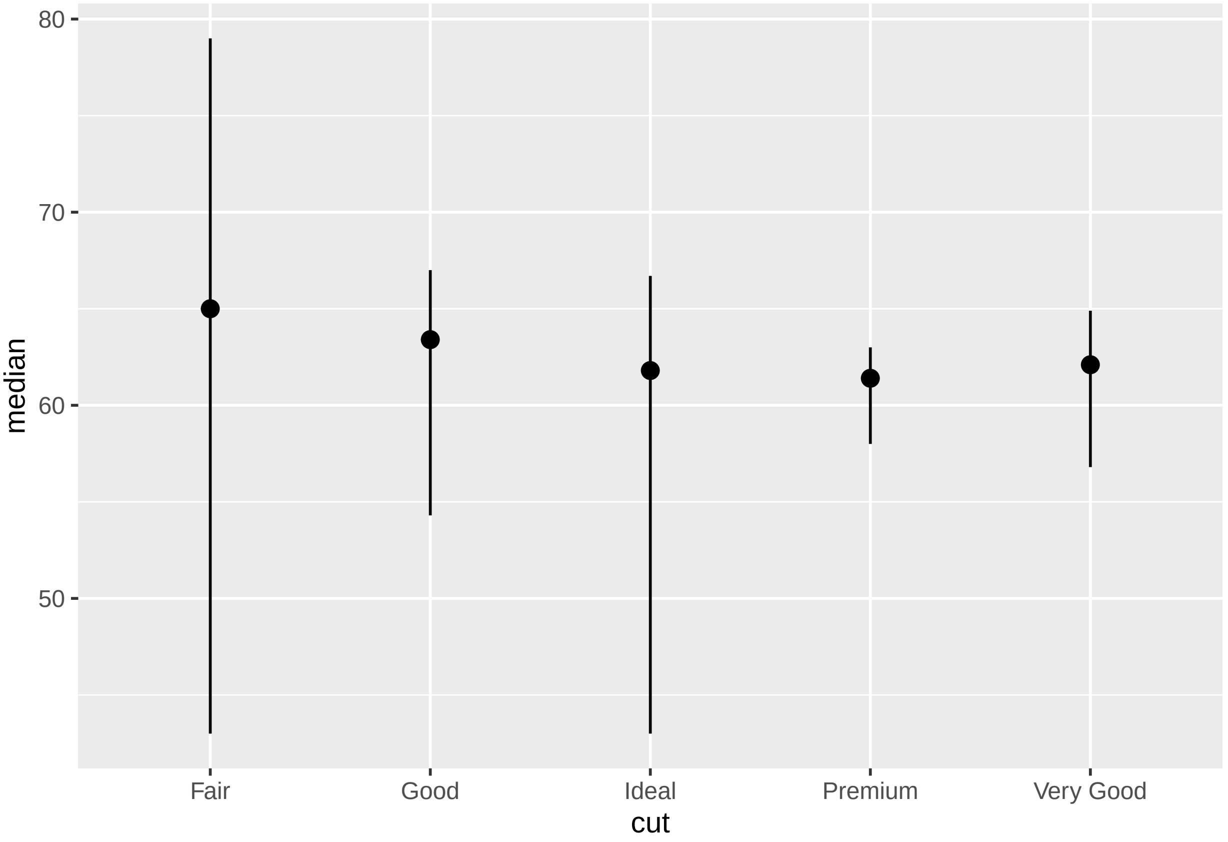
```
ggplot(data = diamonds) +
  stat_summary(
    mapping = aes(x = cut, y = depth),
    fun.ymin = min,
    fun.ymax = max,
    fun.y = median
  )
```

```
## Warning: 'fun.y' is deprecated. Use 'fun' instead.

## Warning: 'fun.ymin' is deprecated. Use 'fun.min' instead.

## Warning: 'fun.ymax' is deprecated. Use 'fun.max' instead.
```



#### a) What’s its default type of geometric object (i.e. its default for the geom argument)?
The default type of geometric object is “pointrange”.
#### b) Replicate and show the following code.
Code:

```
grouped_by_cut <- data.frame(
  cut = c("Fair", "Good", "Very Good",
          "Premium", "Ideal"),
  lower = c(43.0, 54.3, 56.8, 58.0, 43.0),
  upper = c(79.0, 67.0, 64.9, 63.0, 66.7),
  median = c(65.0, 63.4, 62.1, 61.4, 61.8)
)

ggplot(data = grouped_by_cut) +
  geom_pointrange(mapping = aes(
    x = cut,
    y = median,
    ymin = lower,
    ymax = upper
  ))
```



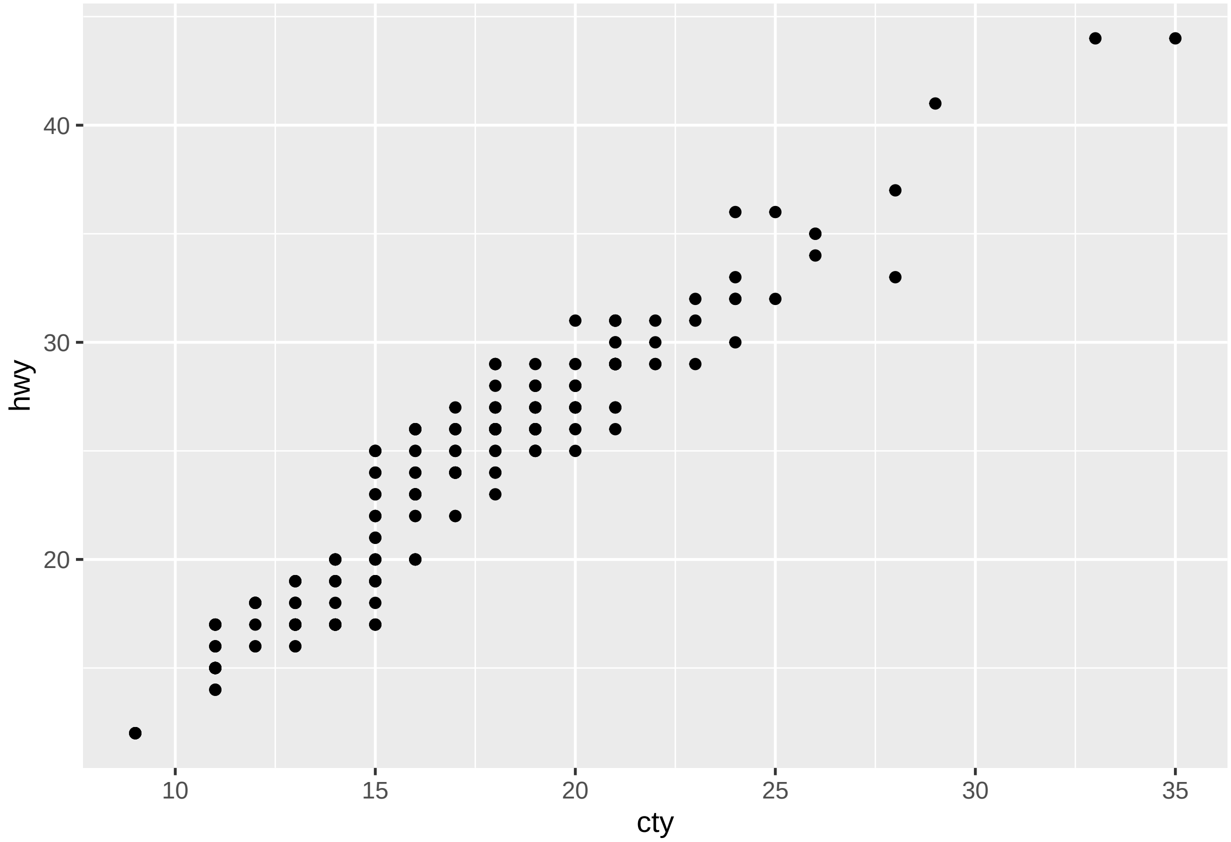
### Exercise 15: What statistical values does it compute?
The statistical values ? stat_smooth() computes is ymin(minimum y) or xmin(minimum x), ymax(maximum y) or xmax(maximum x), and se(standard error)
### Exercise 16: What does geom_col() do? How does it differ from geom_bar()?
The *geom_bar()* sets the height of the bar proportional to the number of cases in each group and *geom_col()* sets the height of the bar so it represents the values in the data.

Section 4.6 Exercises

Exercise 17: What's its default position adjustment (i.e. its default for the position argument)? What's the default position adjustment for geom_point()?

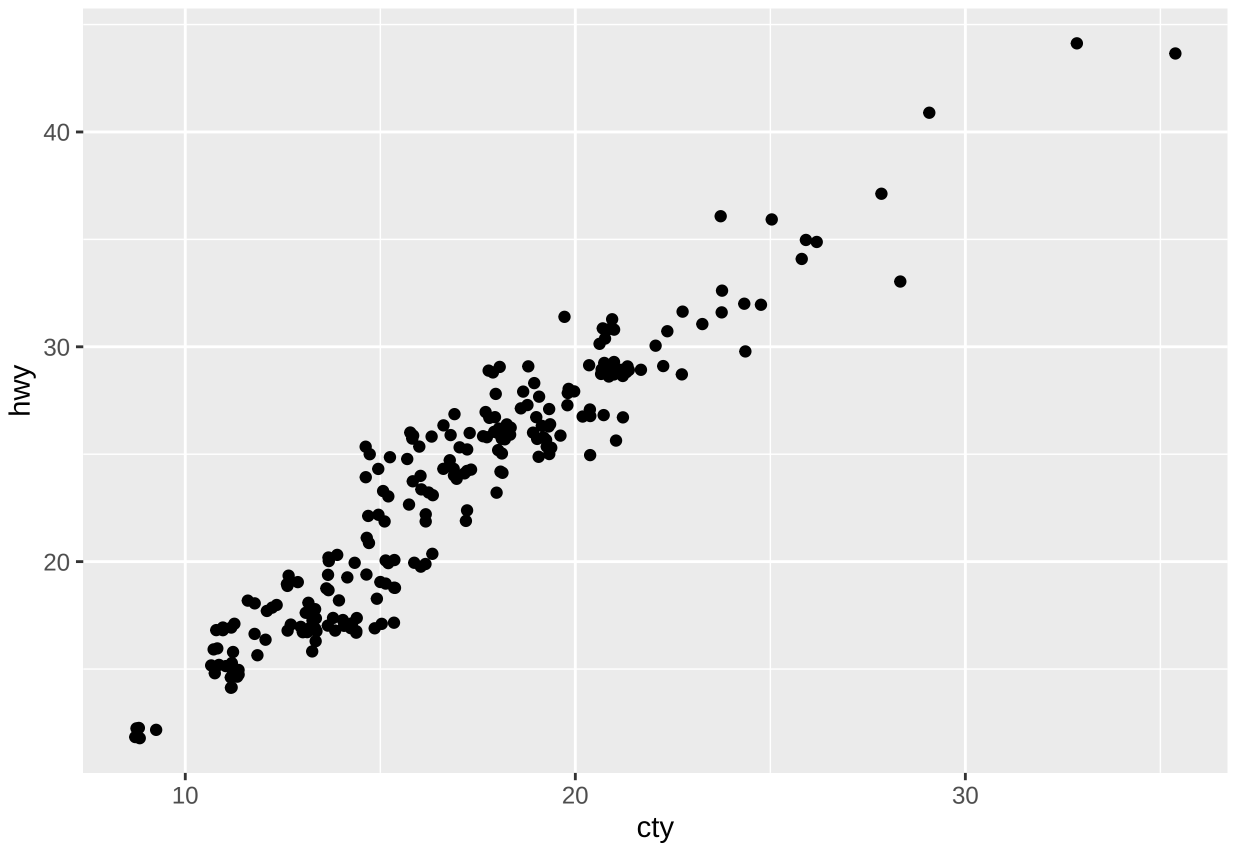
The default position adjustment is *"stack"* for the *geom_bar*. The default position adjustment for *geom_point* is *"identity"*.
### Exercise 18: Answer the following. #### a) What's the problem with the following plot?
Code:

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



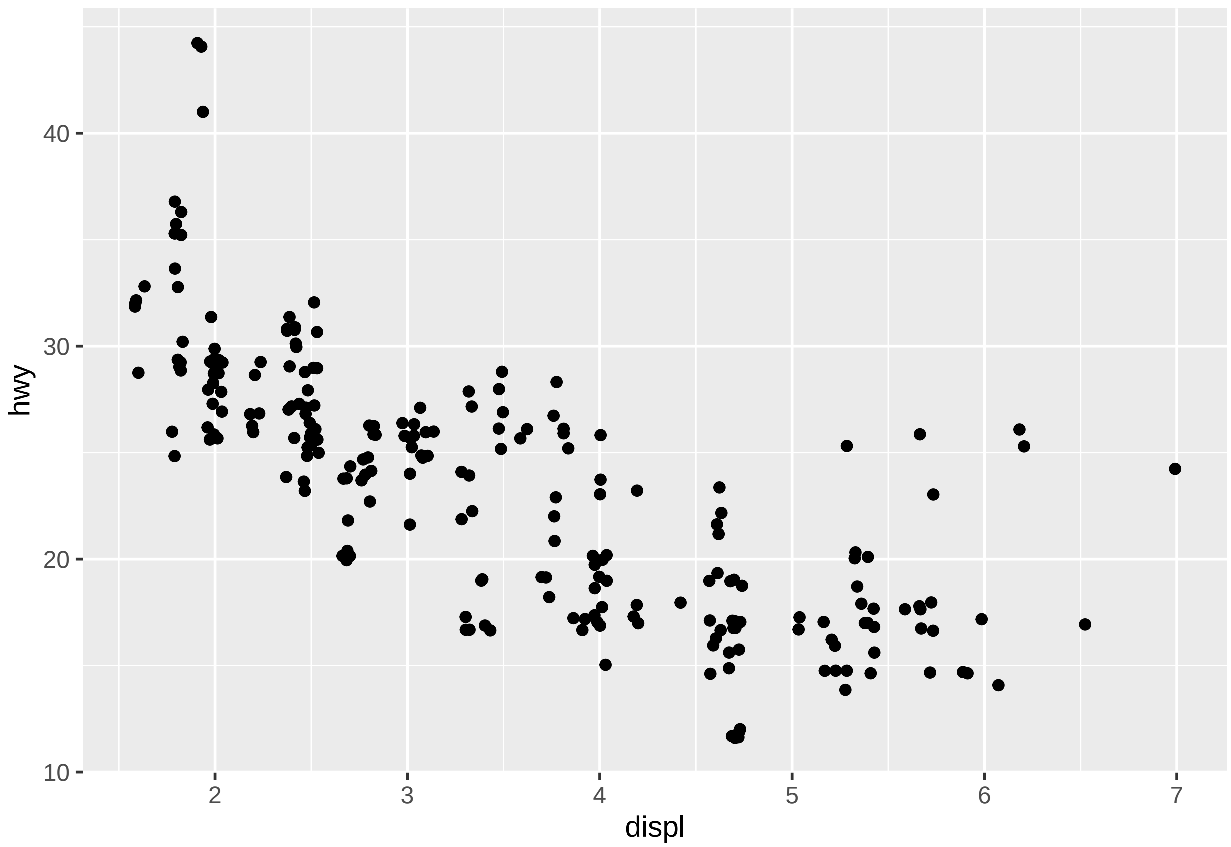
The problem with the plot above is that the graph is confusing because it has multiple dots per each city value. Basically, it shows conflicting information.
#### b) Re-run the command above using position = "jitter" in geom_point(), and describe the improvement in the plot.
Updated Code:

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = cty, y = hwy), position = "jitter")
```



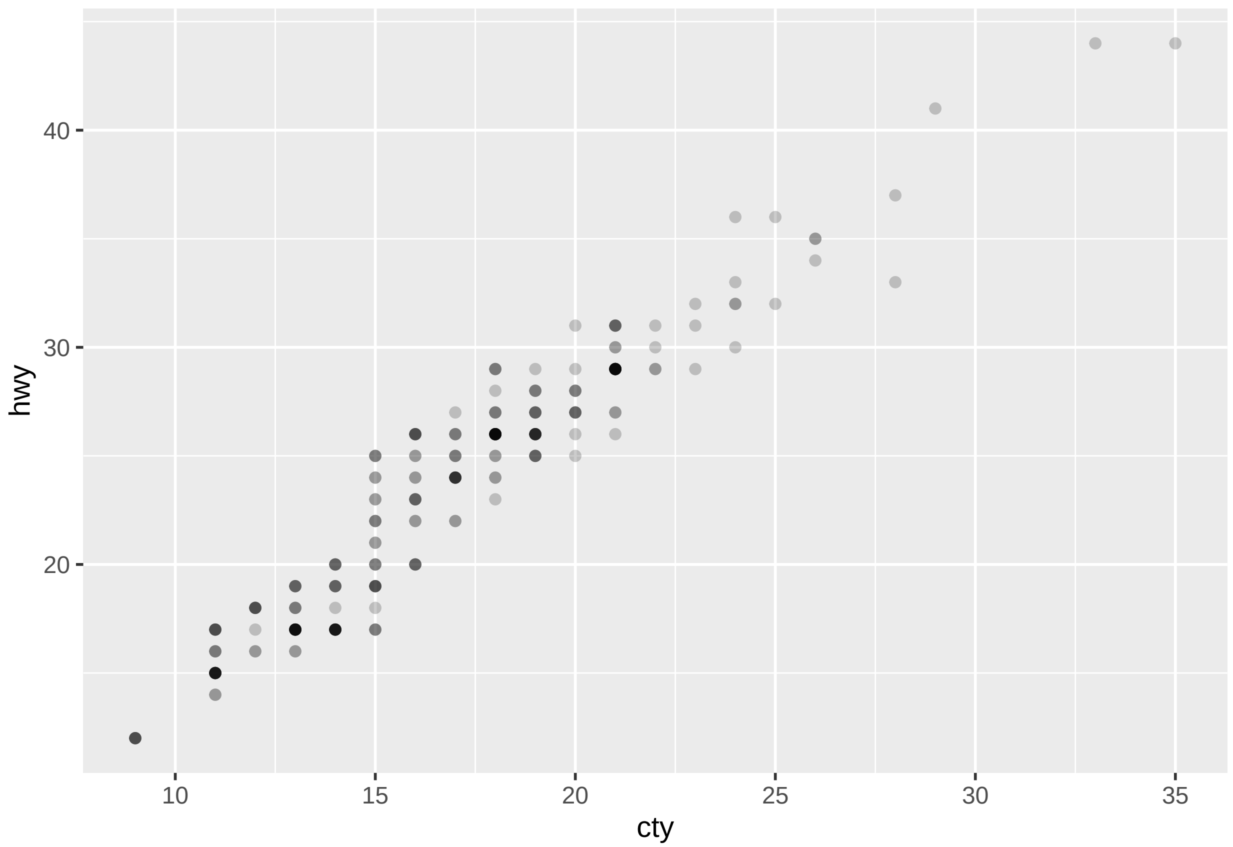
This graph shows the data better since each city value has a separate and clear trend for the highway values. #### c) Verify the following code shows the same graph as in part *b*.
Graph:

```
ggplot(data = mpg) +
  geom_jitter(mapping = aes(x = displ, y = hwy))
```



d) Describe the improvement in the following graph.
Graph:

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = cty, y = hwy), alpha = 0.2)
```



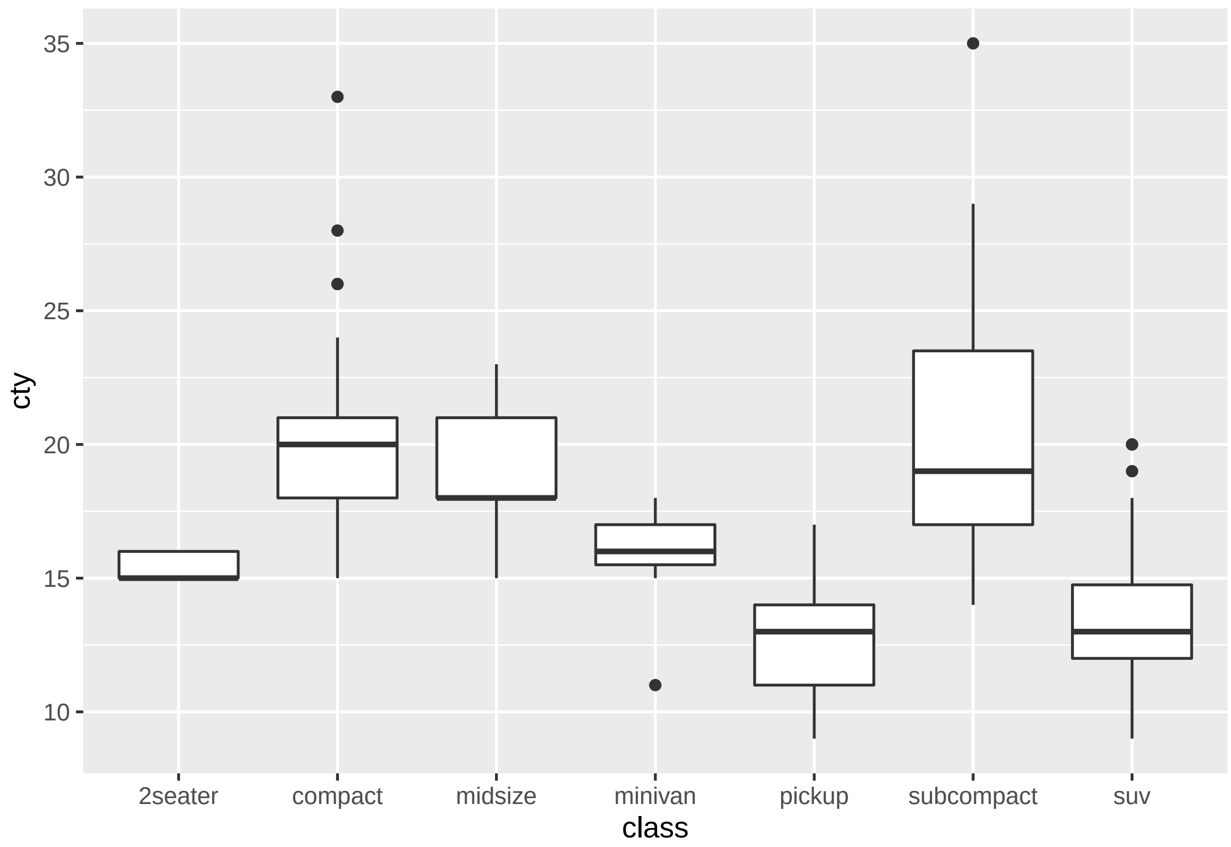
This graph is a lot clearer than the first graph in part *a* because it shows where the majority of the vehicles tested lie when they tested the city and highway mpg.

Section 4.7 Exercises

Exercise 19: Alter the code given above using coord_flip() to produce the following plot. Report your R command(s).

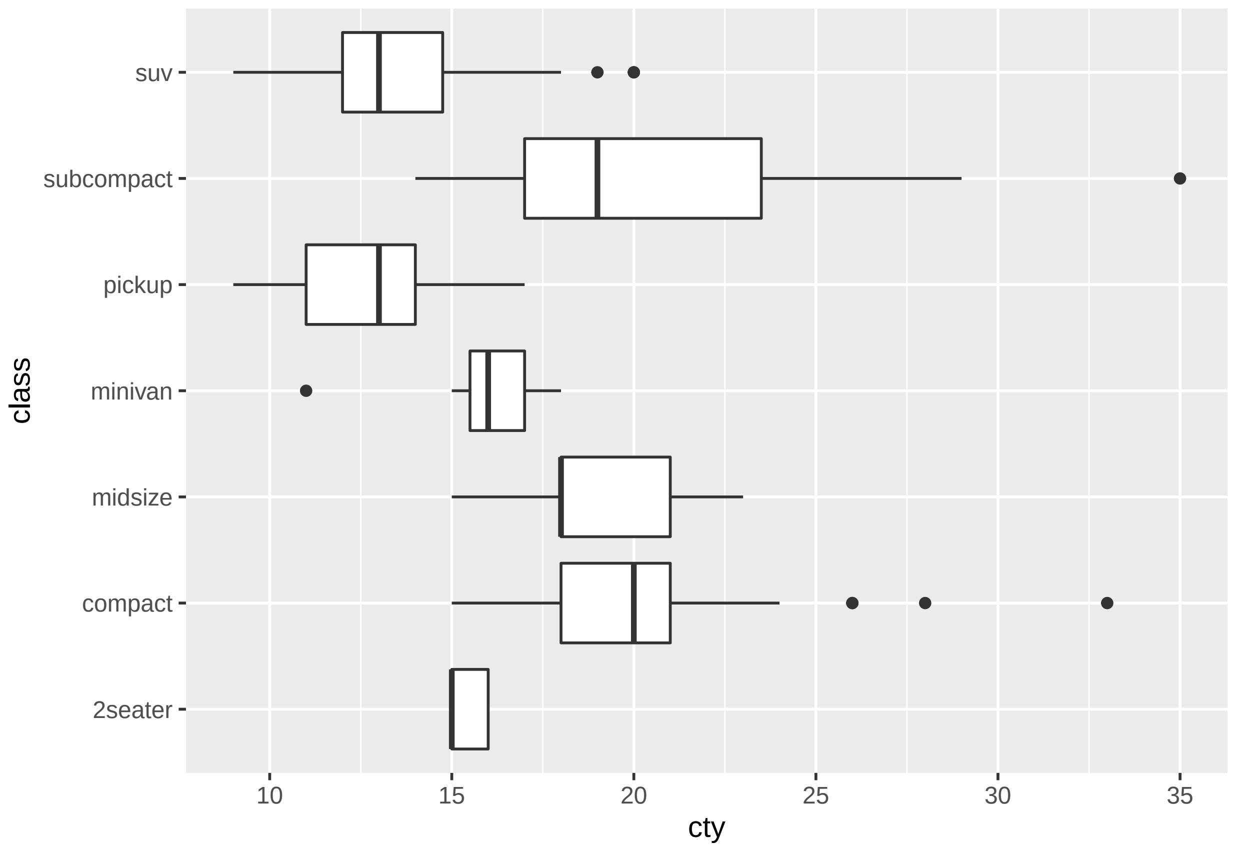
Original Code:

```
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = class, y = cty))
```



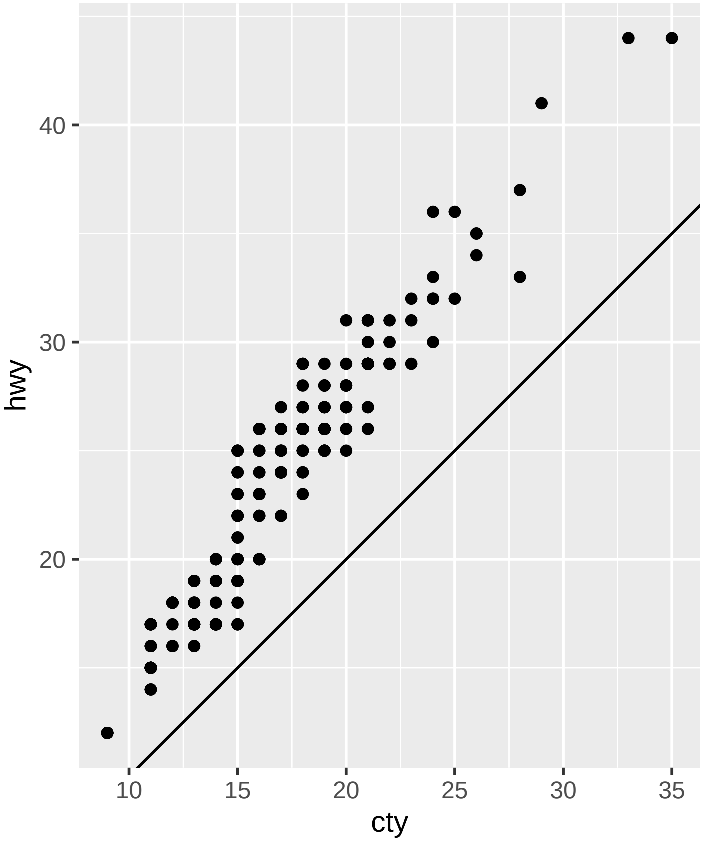
Coord Flip Code:

```
ggplot(data = mpg) +
  geom_boxplot(mapping = aes(x = class, y = cty)) +
  coord_flip()
```



### Exercise 20: Run the following commands. What does the plot tell you about city and highway mpg? Why is coord_fixed() important? What does geom_abline() do?
Code:

```
ggplot(data = mpg, mapping = aes(x = cty, y = hwy)) +
  geom_point() +
  coord_fixed() +
  geom_abline()
```



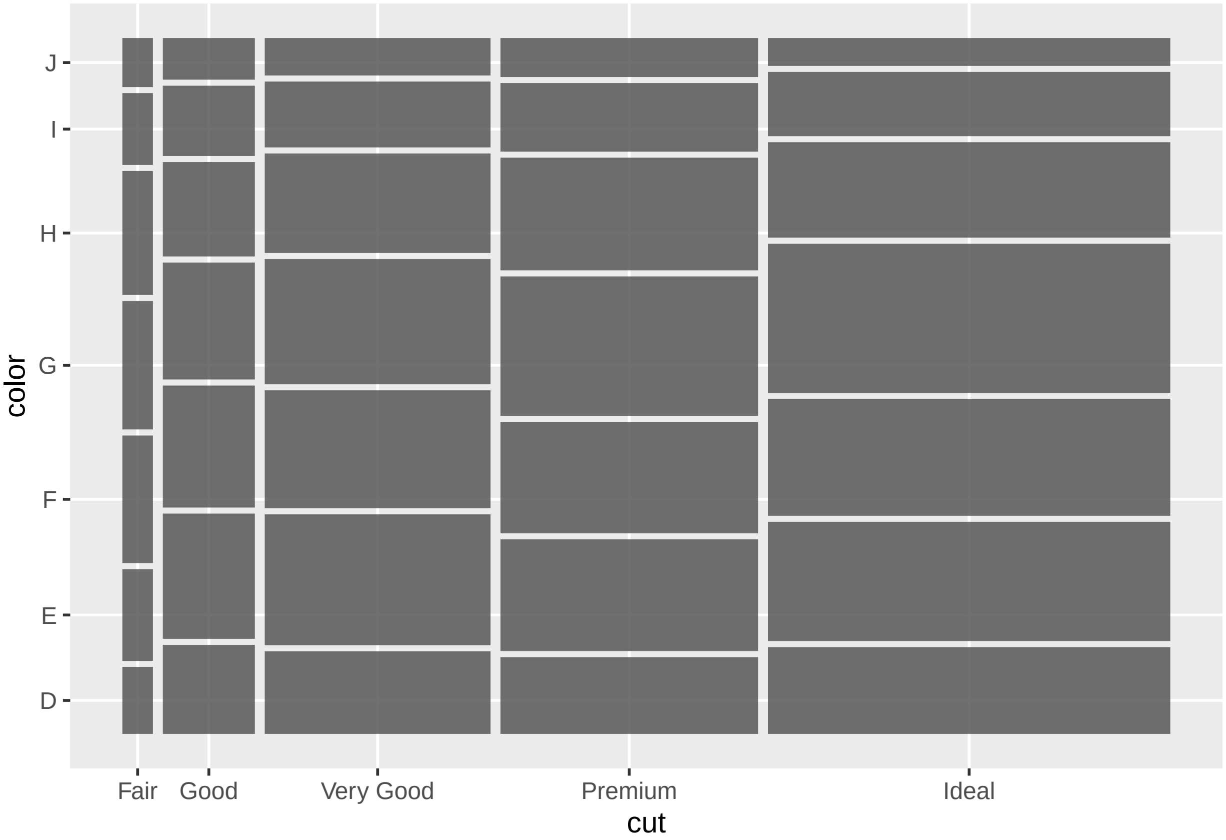
Based on the graph, highway mpg is always going to be higher or the same than the city mpg. The *coord_fixed()* command sets both the x and y scales to a 1:1 ratio in this case. The *geom_abline()* command gives a diagonal line to add as a reference so annotating plots is easier.
## Section 4.10 Exercises ### Exercise 21: Load the "ggmosaic" package and do the following.
Loading ggmosaic:

```
library(ggmosaic)
```

#### a) Make a mosaic plot of the cut and color variables. Report your R command(s).
Code:

```
ggplot(data = diamonds) +
  geom_mosaic(mapping = aes(x = product(color, cut)))
```

```
## Warning: ‘unite_()‘ was deprecated in tidyr 1.2.0.
## Please use ‘unite()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was generated.
```



#### b) Based on the plot, which combination of cut and color do you think is most prevalent? Which is least prevalent?
Based on the above graph, I would say the most prevalent is the ideal cut with the g color. The least prevalent is the fair cut with the j color.
#### c) Is your answer to Part b consistent with the values in the contingency table?
Code:

```
table(diamonds$color, diamonds$cut)
```

```
##
##      Fair Good Very Good Premium Ideal
##   D  163  662      1513    1603  2834
##   E  224  933      2400    2337  3903
##   F  312  909      2164    2331  3826
##   G  314  871      2299    2924  4884
##   H  303  702      1824    2360  3115
##   I  175  522      1204    1428  2093
##   J  119  307       678     808   896
```

My answer to part *b* is consistent to the table above.