COMPSCIX 415.2 Homework 2

Rajat Jain

June 16th, 2018

Contents

Section 3.2.4 Exercises

1. **Run ggplot(data = mpg). What do you see?**

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


We see a blank plot because this statement just sets up the co-ordinate system without plotting anything. To plot, we need atleast two more layers - aesthetics(levels) and geometry (Eg. point).

2. **How many rows are in mpg? How many columns?**

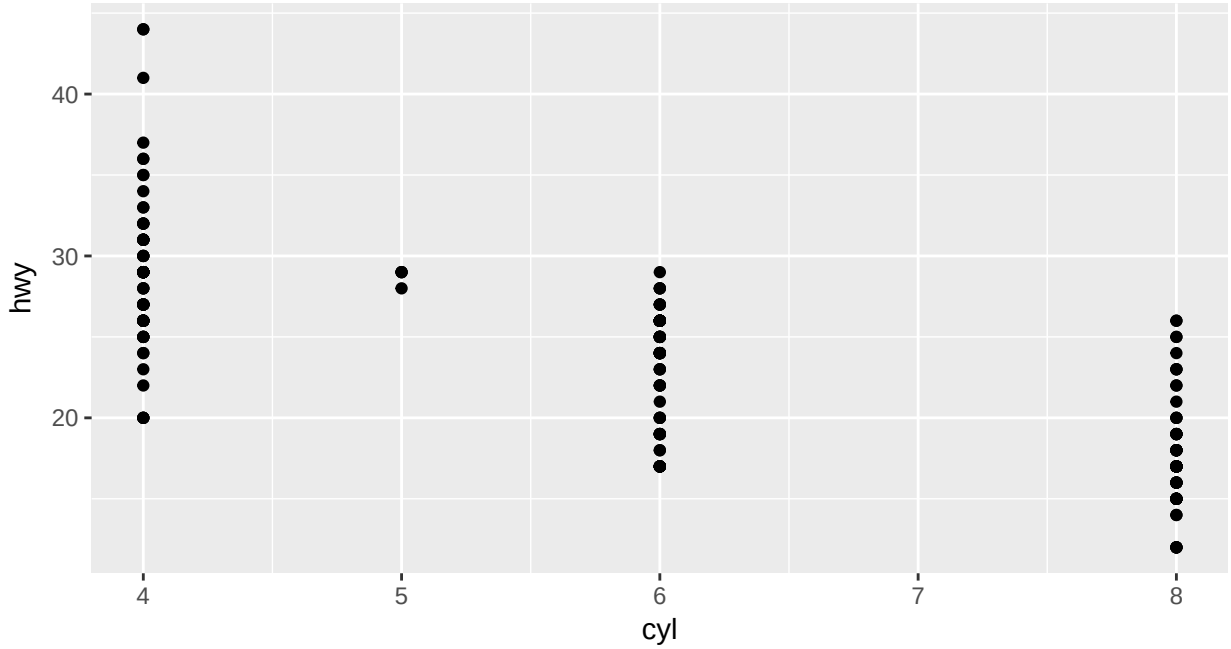
The `mpg` data set has 234 rows and 11 columns.

3. **What does the drv variable describe? Read the help for ?mpg to find out.**

Variable `drv` describes the drive terrain - f = front-wheel drive, r = rear wheel drive, 4 = 4wd.

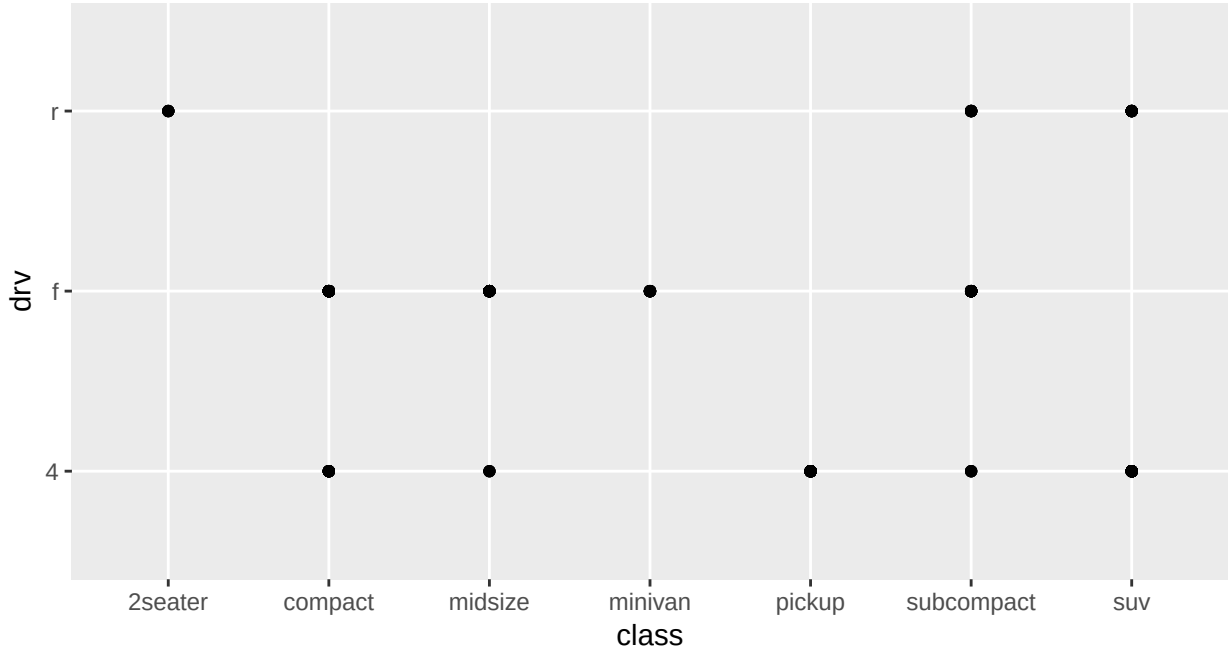
4. **Make a scatterplot of hwy vs cyl.**

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



5. **What happens if you make a scatterplot of class vs drv? Why is the plot not useful?**

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

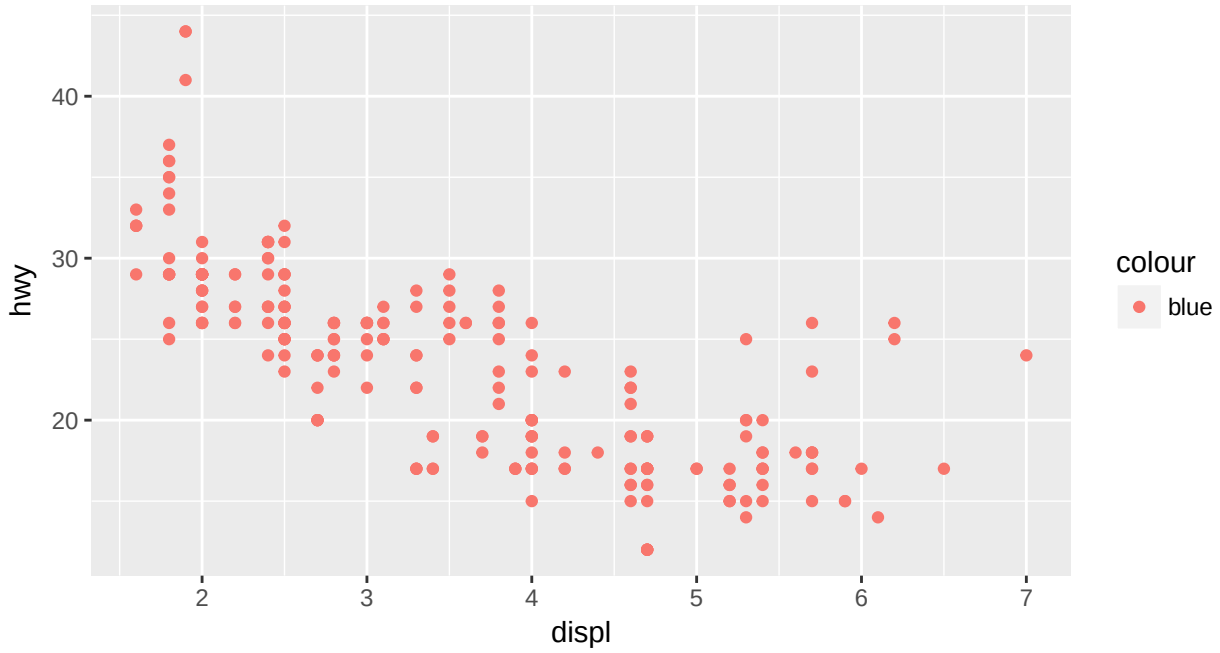


This plot is **not** very useful because both variables are categorical. So, it just shows the existence of overlapping observations with various combinations of `class` and `drv` without providing any meaningful insights.

Section 3.3.1 Exercises

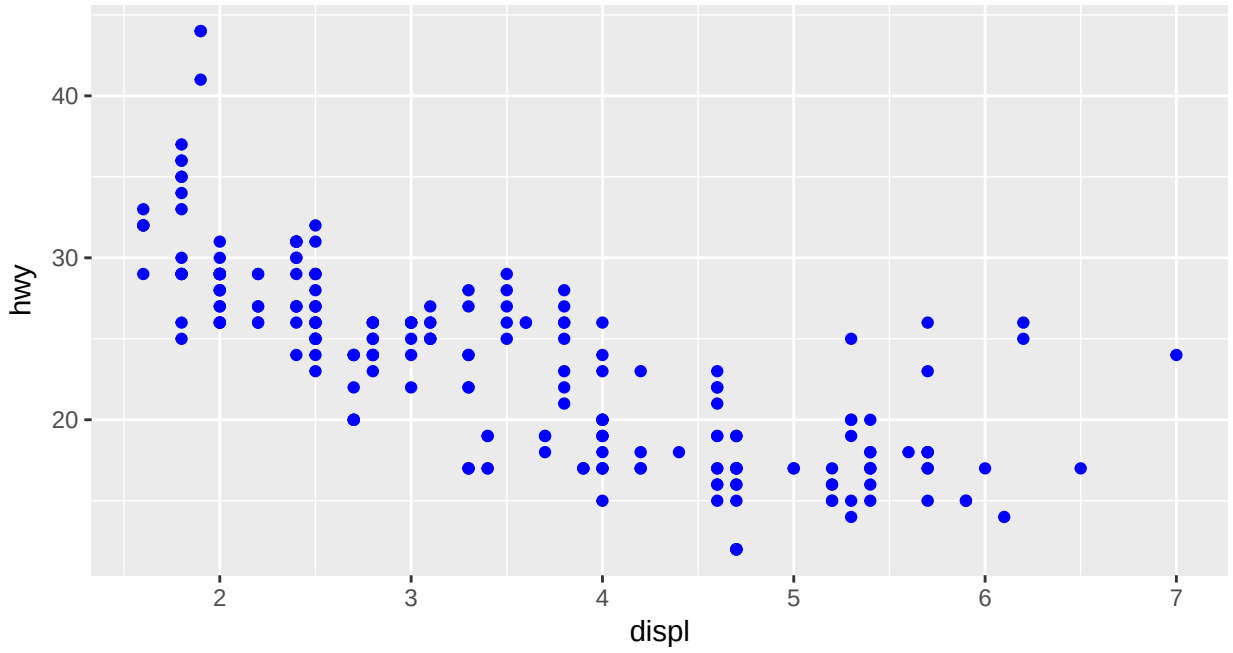
1. **What's gone wrong with this code? Why are the points not blue?**

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



In this code `color` aesthetic is being **mapped**. This kind of mapping should be used when `color` needs to be mapped to a variable. To **set** the color of the points, the `color` setting should be used outside of `aes()` Here is the fixed version:

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



2. **Which variables in mpg are categorical? Which variables are continuous? (Hint: type ?mpg to read the documentation for the dataset). How can you see this information when you run mpg?**

The following variables in `mpg` are categorical:

- manufacturer
- model
- trans
- drv
- fl
- class

Following variables are continuous:

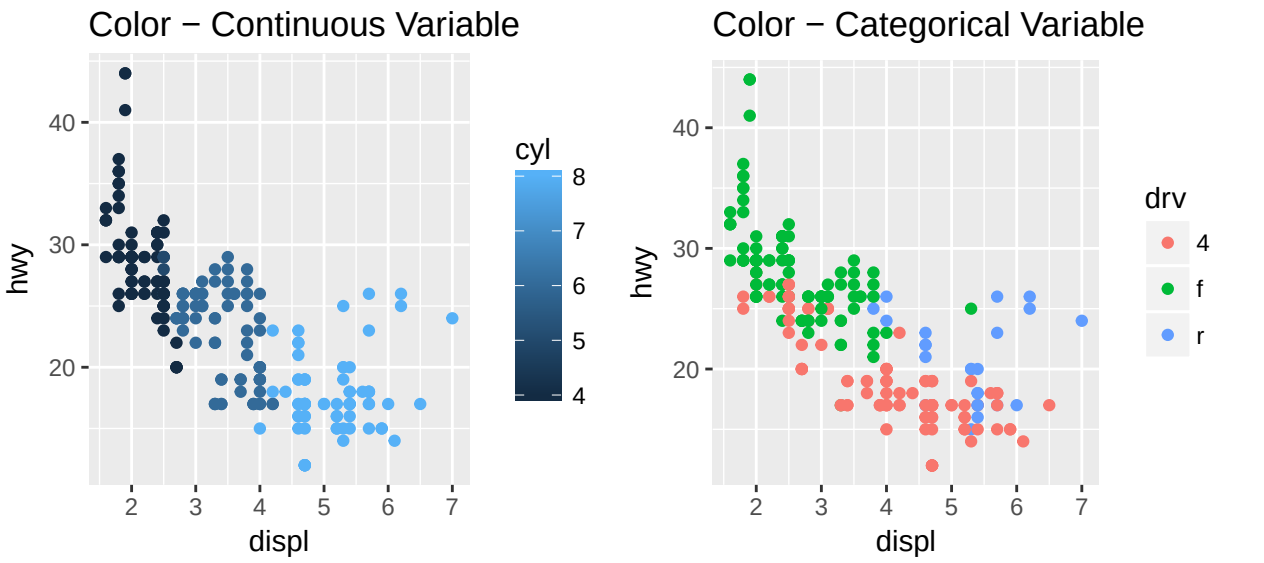
- displ
- year
- cyl
- cty
- hwy

`mpg` is a tibble object. So just running `mgp` in R will show the **type** of each variable.

3. **Map a continuous variable to color, size, and shape. How do these aesthetics behave differently for categorical vs. continuous variables?**

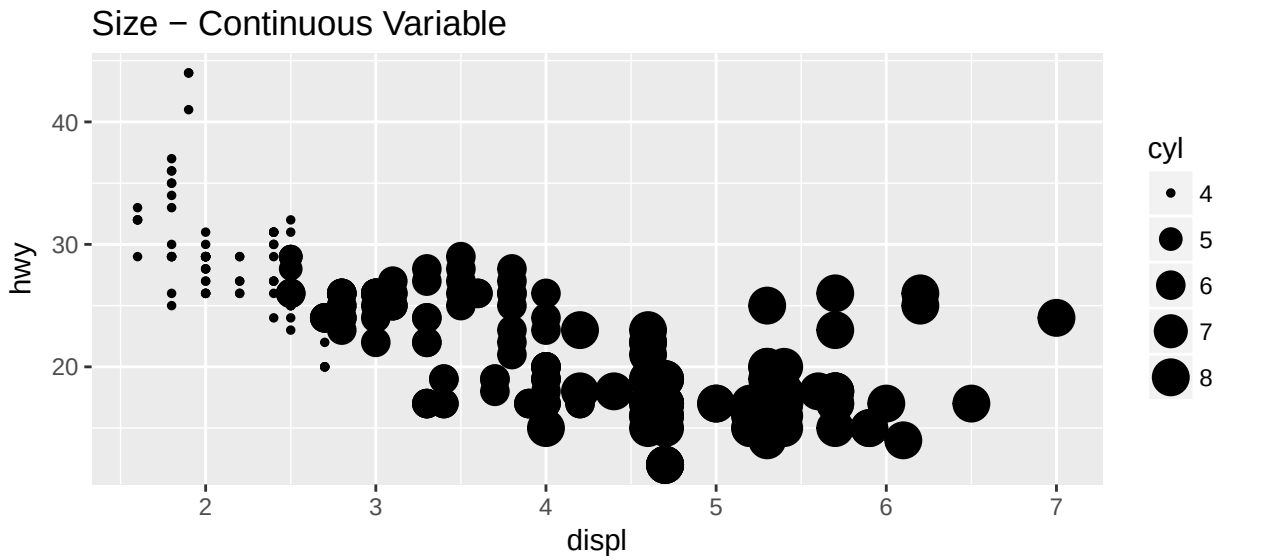
- `color` - maps **continuous** variables to the gradient of a single colour and **categorial** variables to different colors, as far apart from each other as possible. For example, the two plots below respectively show the continuous variable `cyl` and the categorical variable `drv` mapped to `color`.

```
p1 <- ggplot(data = mpg) + ggtitle("Color - Continuous Variable") +
  geom_point(mapping = aes(x = displ, y = hwy, color = cyl))

p2 <- ggplot(data = mpg) + ggtitle("Color - Categorical Variable") +
  geom_point(mapping = aes(x = displ, y = hwy, color = drv))

grid.arrange(p1, p2, ncol = 2)
```



- `size` - maps **continuous** variables to the size(area) of the mark. For example, the plot below shows the continuous variable `cyl` mapped to `size`. Mapping **categorical** variables to `size` works, although is not suggested.

```
ggplot(data = mpg) + ggtitle("Size - Continuous Variable") +
  geom_point(mapping = aes(x = displ, y = hwy, size = cyl))
```

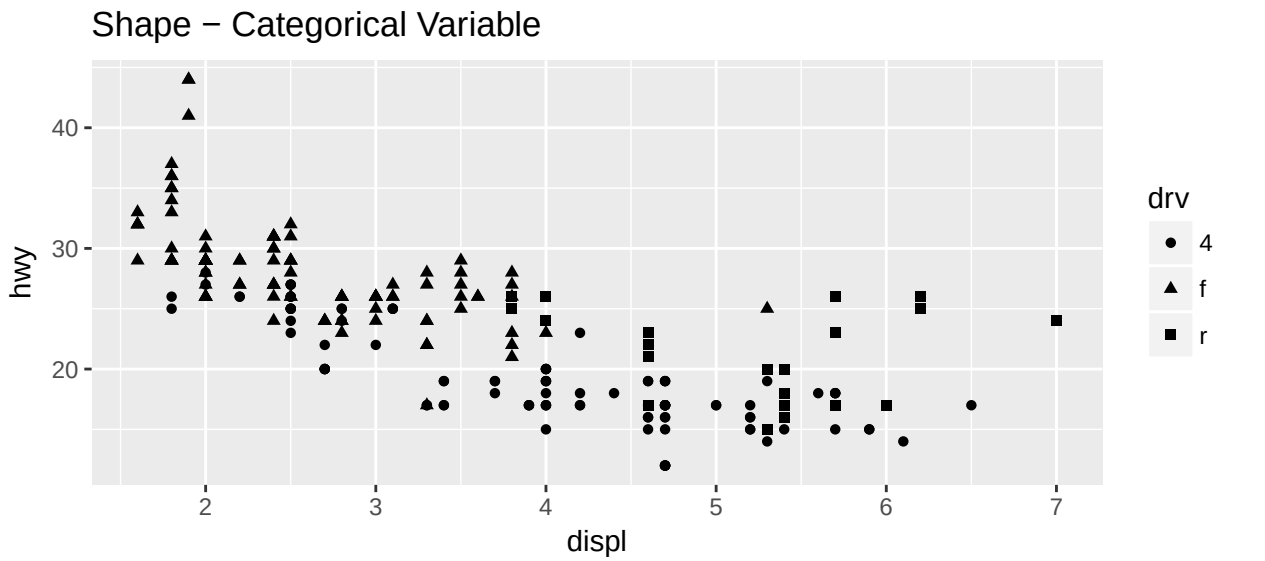


- `shape` - doesn't work with a **continuous** variable. For example, this code tries to map continuous variable `cyl` to `shape`. It will throw an error on execution.

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

When **categorical** variables are mapped to `shape`, the resultant plot has different shapes each representing one class of the mapped categorical variable. For example, in this plot, categorical variable `drv` is mapped to `shape`.

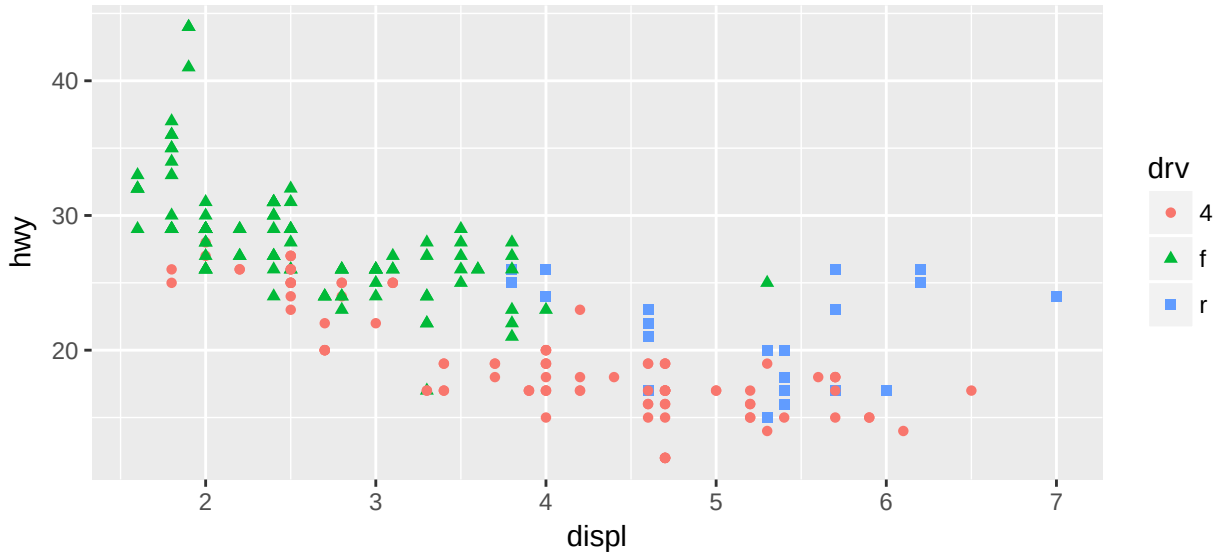
```
ggplot(data = mpg) + ggtitle("Shape - Categorical Variable") +
  geom_point(mapping = aes(x = displ, y = hwy, shape = drv))
```



4. **What happens if you map the same variable to multiple aesthetics?**

Mapping the same variable to multiple aesthetics works fine. For example, variable `drv` is mapped to both `shape` and `color` in the plot below.

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



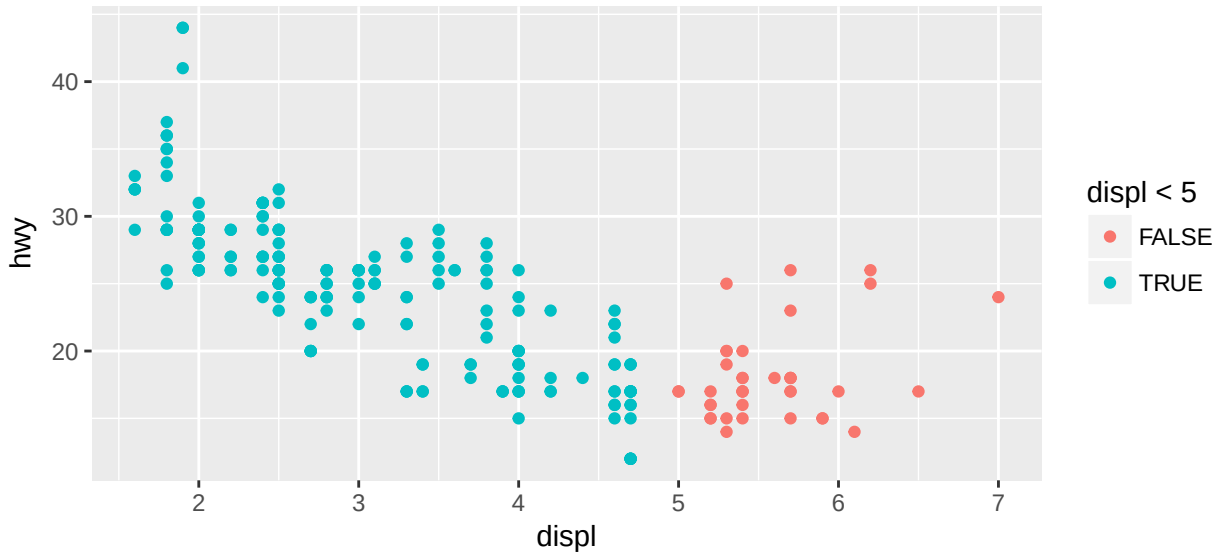
5. **What does the stroke aesthetic do? What shapes does it work with? (Hint: use ?geom_point)**

The `stroke` aesthetic for `geom_point()` is used to change border size of points. It works only with those shapes which have a border. For example, shapes between 21 and 25.

6. **What happens if you map an aesthetic to something other than a variable name, like aes(colour = displ < 5)?**

It works fine. In the example above when `color` aesthetic is mapped to `displ < 5`, it gets treated as if mapped to a categorical variable with values `TRUE` and `FALSE`.

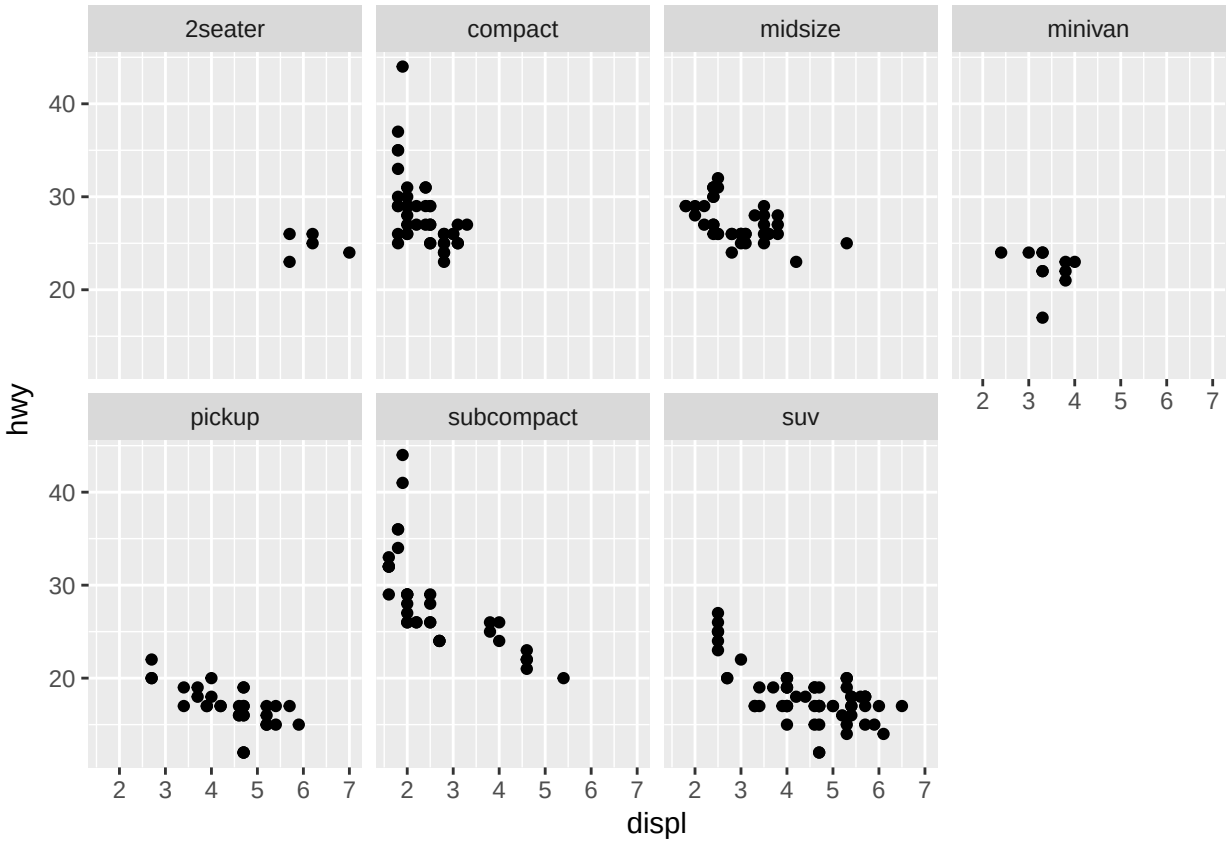
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy, color = displ < 5)) + geom_point()
```



Section 3.5.1 Exercises

4. **Take the first faceted plot in this section:**

```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  facet_wrap(~ class, nrow = 2)
```



What are the advantages to using faceting instead of the colour aesthetic? What are the disadvantages? How might the balance change if you had a larger dataset?

The major advantage of using facets is that they can be used to cleanly visualize data in more than 2 dimensions. While the color aesthetic can help convey similar insights, they are plotted on the same canvas space and often become too difficult to differentiate when we have more than a few classes. The disadvantage of using facets is that the comparison between classes becomes difficult as they are plotted on separate small plots. We have to rely on the alignment of various plots to compare correctly. With larger datasets, if the number of classes being plotted is large, facets might be a better visualization option.

5. **Read ?facet_wrap. What does nrow do? What does ncol do? What other options control the layout of the individual panels? Why doesn't facet_grid() have nrow and ncol argument?**

`nrow` defines the number of rows in the 2d sequence of facets. Similarly, `ncol` defines the number of columns in the 2d sequence of facets. Other options which control the layout of the individual panels are `as.table` and `dir`. `facet_grid()` doesn't have `nrow` and `ncol` arguments because the size of matrix of facet_grid panels is defined by the row and column facetting variables.

Section 3.6.1 Exercises

1. **What geom would you use to draw a line chart? A boxplot? A histogram? An area chart?**

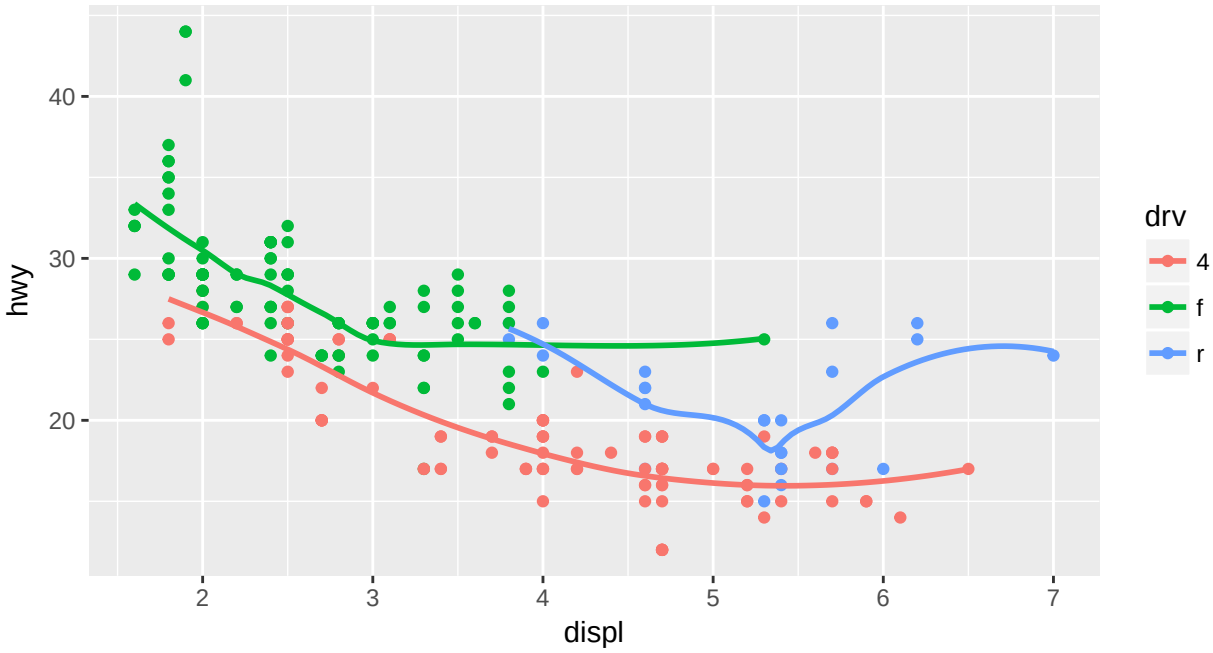
Following geom could be used for each of these charts:

- Line Chart - `geom_line()`
- Boxplot - `geom_boxplot()`
- Histogram - `geom_histogram()`
- Area Chart - `geom_area()`

2. **Run this code in your head and predict what the output will look like. Then, run the code in R and check your predictions.**

This code should generate a scatter plot of `displ` v/s `hwy` with values of `drv` represented by individual colors. Each class of `drv` should have their own colored smooth line without displaying the confidence intervals.

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



3. **What does show.legend = FALSE do? What happens if you remove it? Why do you think I used it earlier in the chapter?**

It drops the legend created automatically by `ggplot2`. If we remove it, the legend will start showing up whenever applicable. It was used earlier in the chapter because the bar chart produced in the example had the same variable `cut` mapped to both aesthetics `x` and `fill`. The values corresponding to `cut` were already displayed on the `x` axis as **labels**. Having them display as the **legend** for `fill` added no value.

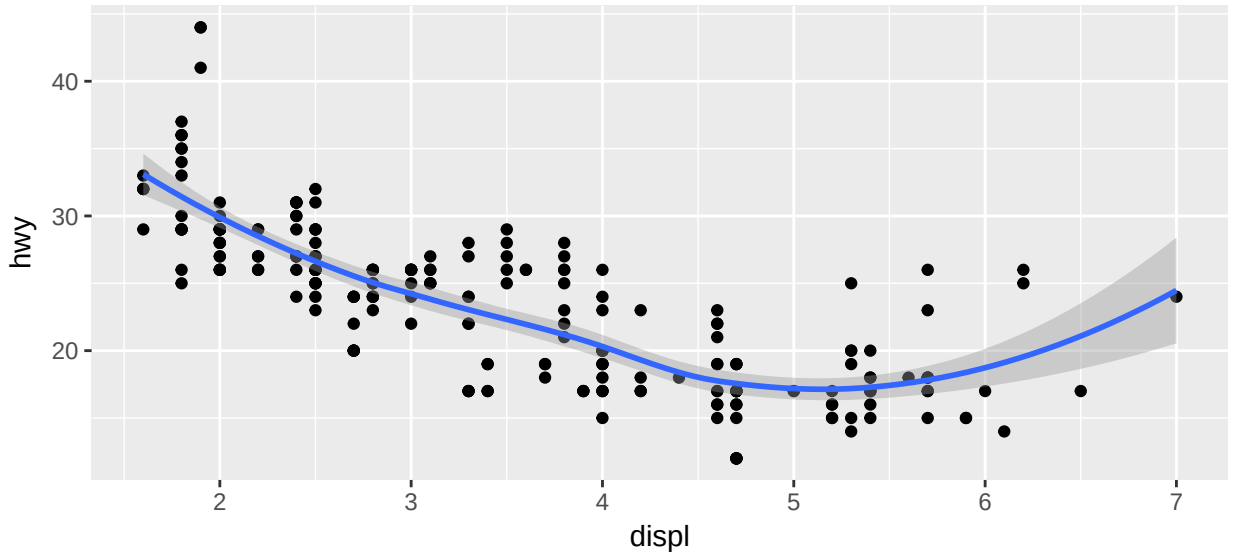
4. **What does the se argument to geom_smooth() do?**

The `se` argument to `geom_smooth()` toggles displaying confidence interval around smooth On and Off. It is set to `TRUE` (On) by default.

5. **Will these two graphs look different? Why/why not?**

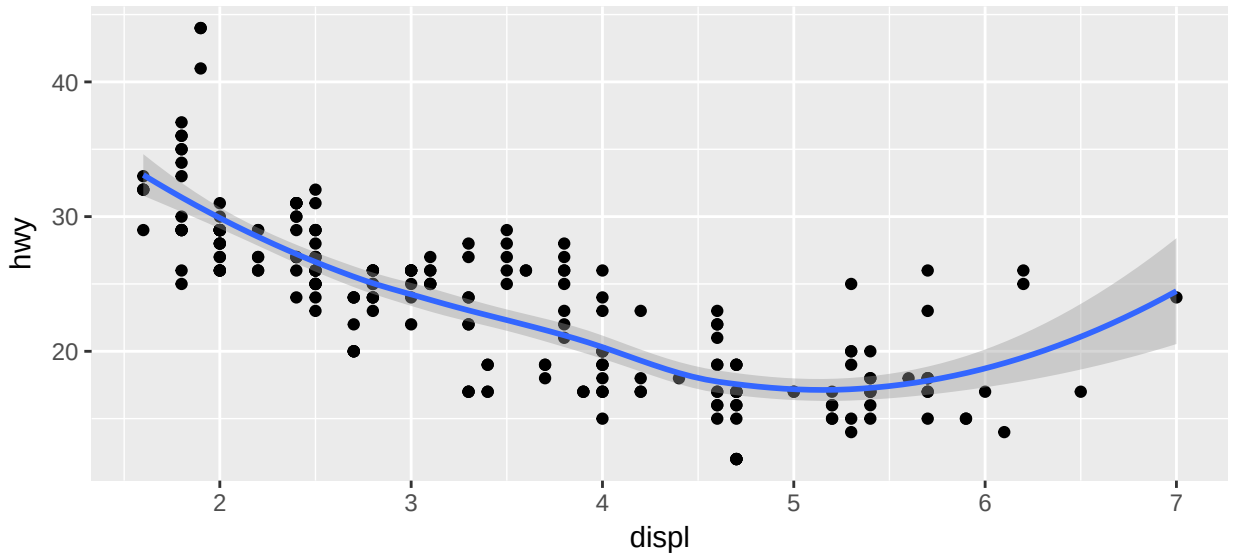
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess'
```




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

```
## `geom_smooth()` using method = 'loess'
```

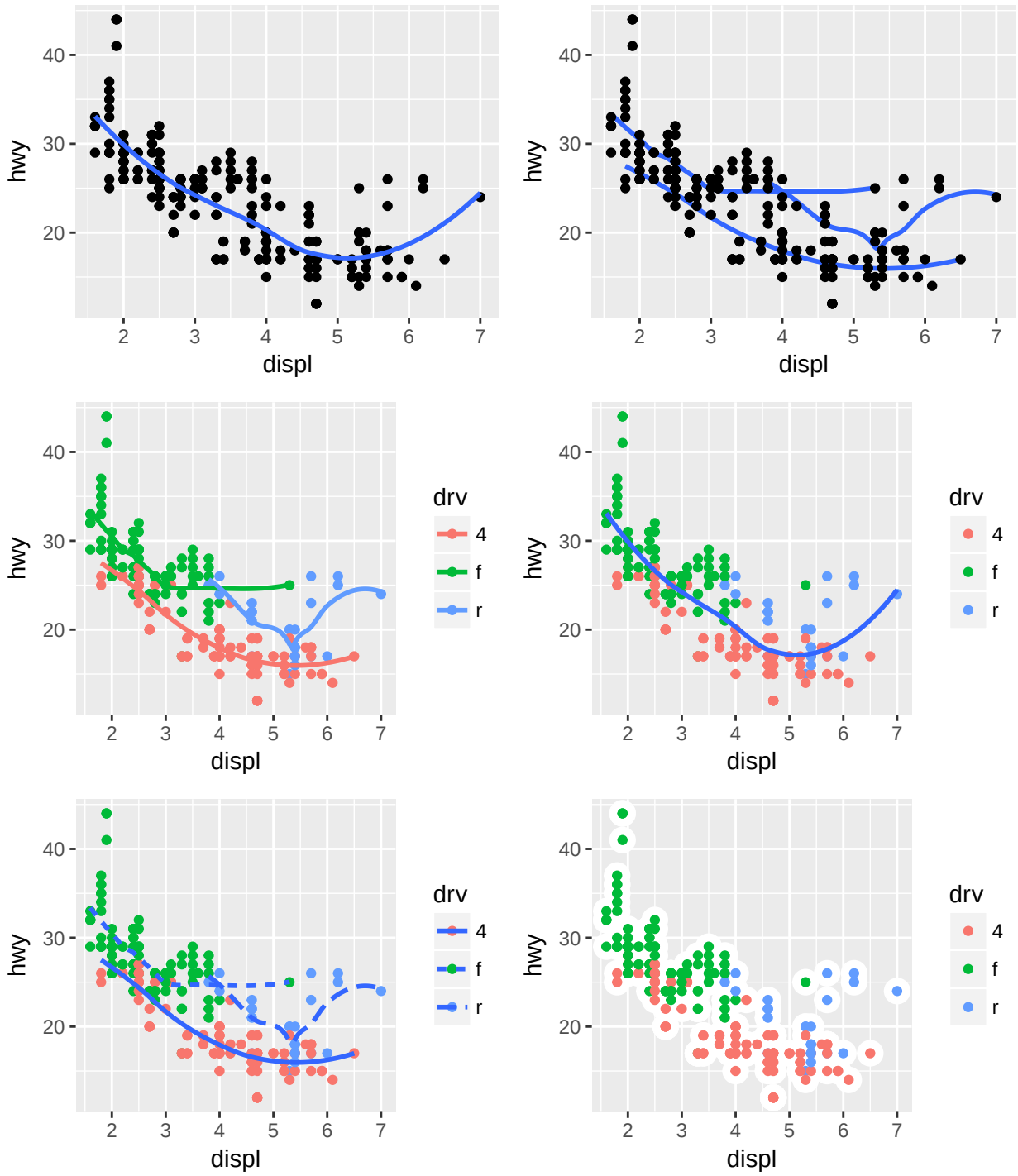


No. They will look exactly the same. It does not matter at which level the aesthetic is added as long as it is same. In the first case here, the aesthetics is being added at the top before adding the geometry layers. Whereas, in the second case the aesthetic is being added individually for both the geometry layers. However, the aesthetic is exactly the same. So the output plot of the two statements will look exactly the same.

6. **Recreate the R code necessary to generate the following graphs.**

Here is the code necessary to generate the graphs shown:

```
# Define a common ggplot object with common x and
# y aesthetic to avoid repeating it over and over.
plot <- ggplot(data = mpg,
               mapping = aes(x = displ, y = hwy))

# First Plot.
plot.1 <- plot +
  geom_point() +
  geom_smooth(se = FALSE)

# Second Plot.
plot.2 <- plot +
  geom_smooth(aes(group = drv),
              se = FALSE) +
  geom_point()

# Third Plot.
plot.3 <- plot +
  aes(color = drv) +
  geom_point() +
  geom_smooth(se = FALSE)

# Forth Plot.
plot.4 <- plot +
  geom_point(aes(color = drv)) +
  geom_smooth(se = FALSE)

# Fifth Plot.
plot.5 <- plot +
  geom_point(aes(color = drv)) +
  geom_smooth(aes(linetype = drv),
              se = FALSE)

# Sixth Plot.
plot.6<- plot +
  geom_point(size = 5, color='white') +
  geom_point(aes(color = drv))


# Arrange the plots on a 3x2 grid.
grid.arrange(plot.1, plot.2,
             plot.3, plot.4,
             plot.5, plot.6,
             nrow = 3, ncol = 2)
```


hwy
displ
drv
4
f
r

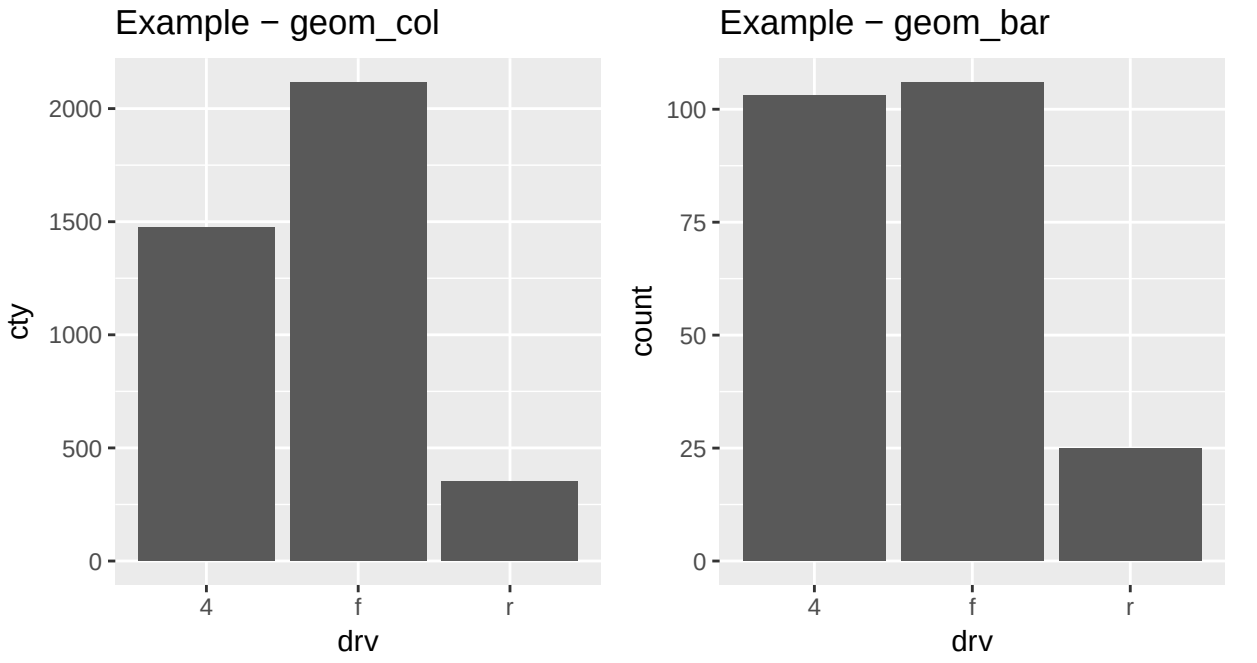
Section 3.7.1 Exercises

2. **What does geom_col() do? How is it different to geom_bar()?**

There are two types of bar charts: `geom_bar()` makes the height of the bar proportional to the number of cases in each group (or if the weight aethetic is supplied, the sum of the weights). If you want the heights of the bars to represent values in the data, use `geom_col()` instead. `geom_bar` uses `stat_count` by default: it counts the number of cases at each `x` position. `geom_col` uses `stat_identity`: it leaves the data as is. Courtesy: `?geom_col`

So, `geom_bar()` requires only `x` aesthetic whereas, `geom_col()` requires both `x` and `y` aesthetic. Here is an example:

```
# Setup common ggplot object.
my_plot <- ggplot(data = mpg)

# Both x and y aesthetic supplied.
my_col <- my_plot + ggtitle("Example - geom_col") +
  geom_col(mapping = aes(x = drv, y = cty))

# Only x aesthetic supplied.
my_bar <- my_plot + ggtitle("Example - geom_bar") +
  geom_bar(mapping = aes(x = drv))

# Arrange output in a grid.
grid.arrange(my_col, my_bar, ncol = 2)
```



Additional Question

Look at the data graphics at the following link: What is a Data Scientist. Please briefly critique the designer's choices. What works? What doesn't work? What would you have done differently?

Here is a section by section review of the visualization choices.

1. **Over 2/3 believe demand for talent will outpace the supply of data scientists**

Author has used a "Doughnut chart". Similar to pie-charts they are difficult to read. A simpler and much easier option would be to represent it using a bar chart.

2. **Only 12% see today's BI professional as the best source for new data scientists**

The visualization used, uses "area" of the block to represent percentages, very difficult to interpret. Color selection is not good. Grey color jumps in between all shades of blue.

3. **Lack of training and resources are the biggest obstacle to data science in organizations**

In this visualization, the author uses a Color saturation scale - darkest to represent the biggest obstacle, lightest to represent the smallest. This is a good idea.

4. **Data scientists are significantly more likely to have advanced degrees than BI professionals**

Tiled view makes it difficult to visualize the complete picture at once. A single chart with side-by-side bars would have been better. Alternatively, a side-by-side comparison of percentage numbers in the form of a table like the one shown below could have also worked.

Degrees	Data Scientist	Business Intelligence
High School	5%	11%
Technical School	5%	4%
Some College	14%	25%
College Graduate	37%	48%
Masters/Professional Degree	31%	12%
Doctoral Degree	9%	1%

5. **Business Intelligence professionals overwhelmingly studied Business in university. Data scientists have more varied backgrounds, especially in hard sciences**

Bubble charts are again "area" based visualizations which are very difficult to read. A side-by-side comparison table with just plain numbers, similar to the one above could have worked. For example:

Field of Study	Data Scientist	Business Intelligence
Computer Sciences	24%	13%
Engineering	17%	6%
Natural Sciences	11%	5%
Business	10%	37%
Mathematics	8%	7%
Humanities	7%	11%
Management Information Systems	6%	4%
Statistics	5%	1%
Social Sciences	5%	6%
Other	8%	10%

6. **Data scientists believe that new technology will create a demand for more data scientists**

Again a "Doughnut" chart.

7. **Characteristics of data scientists**

The Color scale on visualization is random. Could have used a sequential color scale increasing from Normal Data Science to Big Data Science.

8. **Data scientists are more likely to be involved across the data lifecycle**

Again a table with side-by-side comparison or a single chart with side by side bars for each stage of data lifecycle could have worked better. Tiled/facetted view makes it difficult to read.

9. **Who does a data scientist work with?**

It seems like the area of blocks is supposed to represent the values. Although, it doesn't. All blocks are of same size.