Homework 2 - Data Exploration

Daniel Carpenter

August 2022

Table of contents

Packages

- Ideally, these packages will install automatically if you do not have them already

```
library(tidyverse) # get tidverse for piping
library(ggthemes) # themes for plots
```

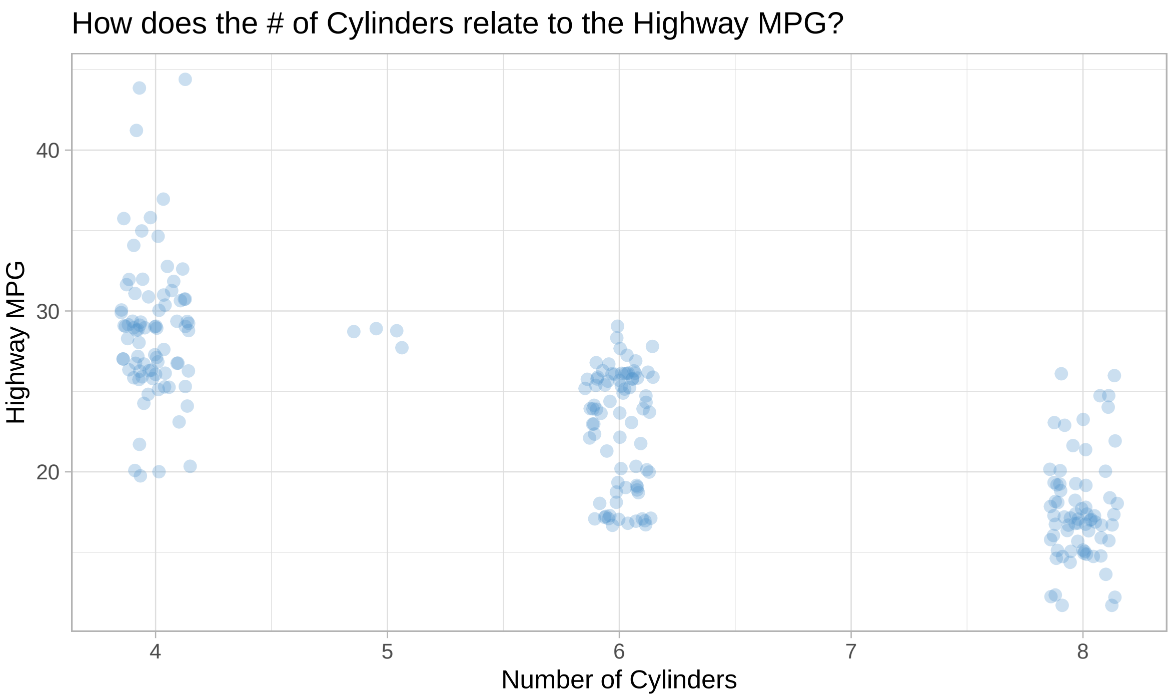
ggplot2

(a) | 3.2.4

Problem 4

Make a scatterplot of hwy vs cyl.

```
theme_set(theme_light()) # set the theme

# ?mpg
mpg %>%

  # hwy vs. cyl
  ggplot(aes(x = cyl,
             y = hwy)
         ) +

  # add points with a little but of jitter to see overlap
  # since discrete number of cylinders
  geom_jitter(color = 'steelblue3', size = 2, alpha = 0.3,
              width = 0.15) + # add points

  # Labels
  labs(title = 'How does the # of Cylinders relate to the Highway MPG?',
       x     = 'Number of Cylinders',
       y     = 'Highway MPG',
       caption = '\nNote small amount of jittering since number of cylinders is discrete')

  theme_get() # get the theme set before
```



Note small amount of jittering since number of cylinders is discrete

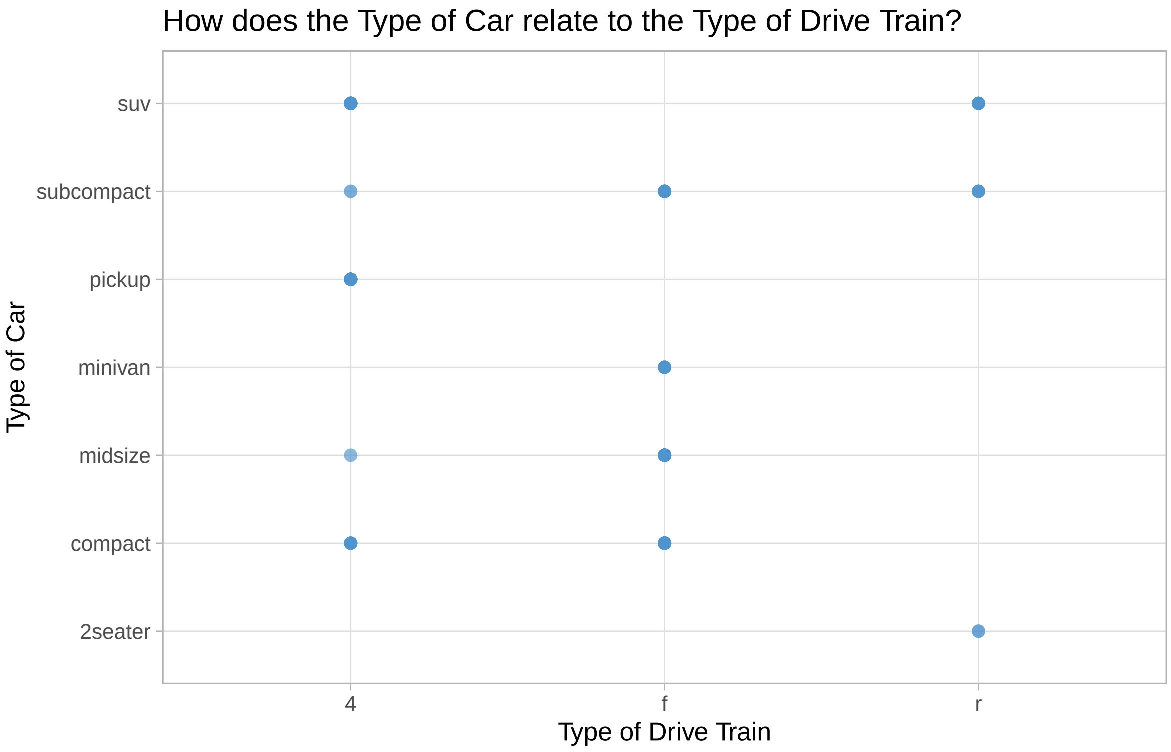
Problem 5

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

Answer: The below scatter is not useful since both the response and independant variables are discrete values (not continuous). This graph only shows the combinations between the dimensions. All data is overlapping.

```
# ?mpg
mpg %>%

  # hwy vs. cyl
  ggplot(aes(x = drv,
             y = class)
         ) +

  # add points
  geom_point(color = 'steelblue3', size = 2, alpha = 0.3) +

  # Labels
```

```
labs(title = 'How does the Type of Car relate to the Type of Drive Train?',
     x     = 'Type of Drive Train',
     y     = 'Type of Car') +

theme_get() # get the theme set before
```



(a) | 3.3.1

Problem 3

Map a continuous variable to color, size, and shape.

Assumptions:

1. Using same x and y variables as problem 1 of excercise 3.3.1
2. Assuming we are only mapping a variable one at a time, just because all three mappings at once could be confusing and lose effectiveness.

How do these aesthetics behave differently for categorical vs. continuous variables?

Answer: You need to be careful with continuous vs. categorical data when mapping. For example, you do not want to determine the size using a a categorical variable, since it will not provide much meaning on correlation. Generally, these will work well at telling a story:

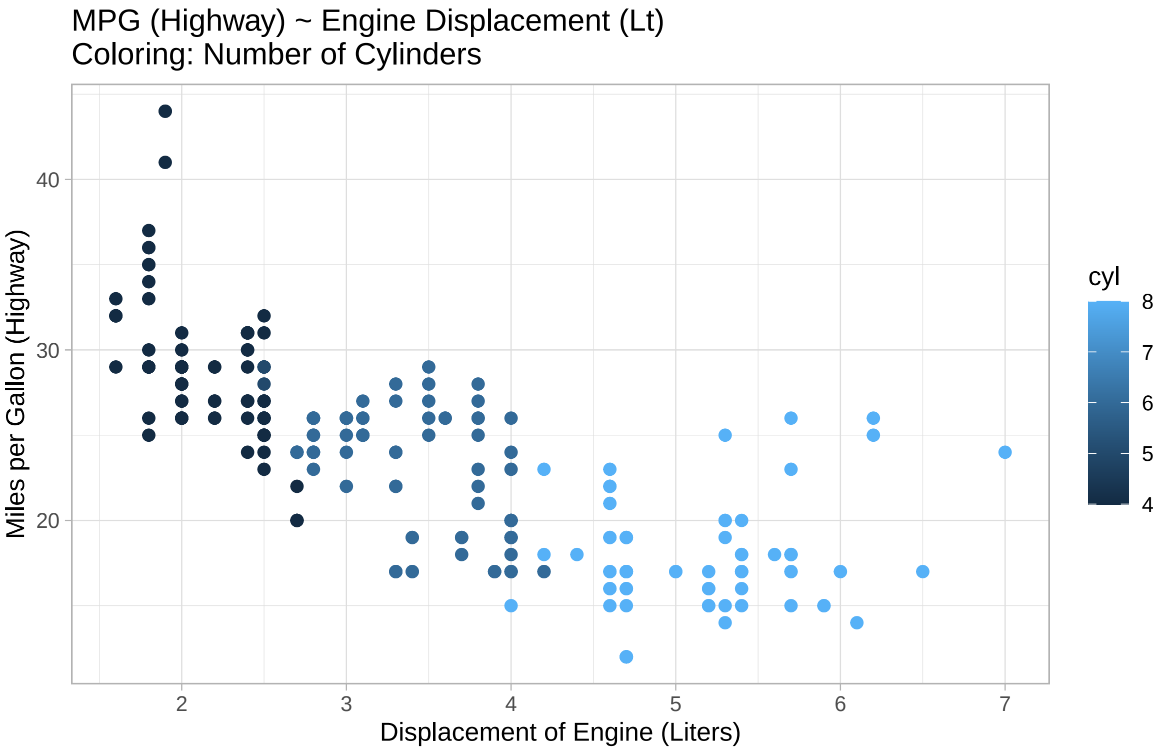
- size: continuous
- color: categorical
- shape: categorical

Create a base plot for reuse:

```
title_base = 'MPG (Highway) ~ Engine Displacement (Lt)\n'

# Create a base plot defined about with hwy ~ displ
plot_base <- mpg %>%

  # hwy vs. cyl
  ggplot(aes(x = displ,
             y = hwy
             )
         ) +

  # Labels
  labs(x     = 'Displacement of Engine (Liters)',
       y     = 'Miles per Gallon (Highway)' ) +

  theme_get() # get the theme set before
```

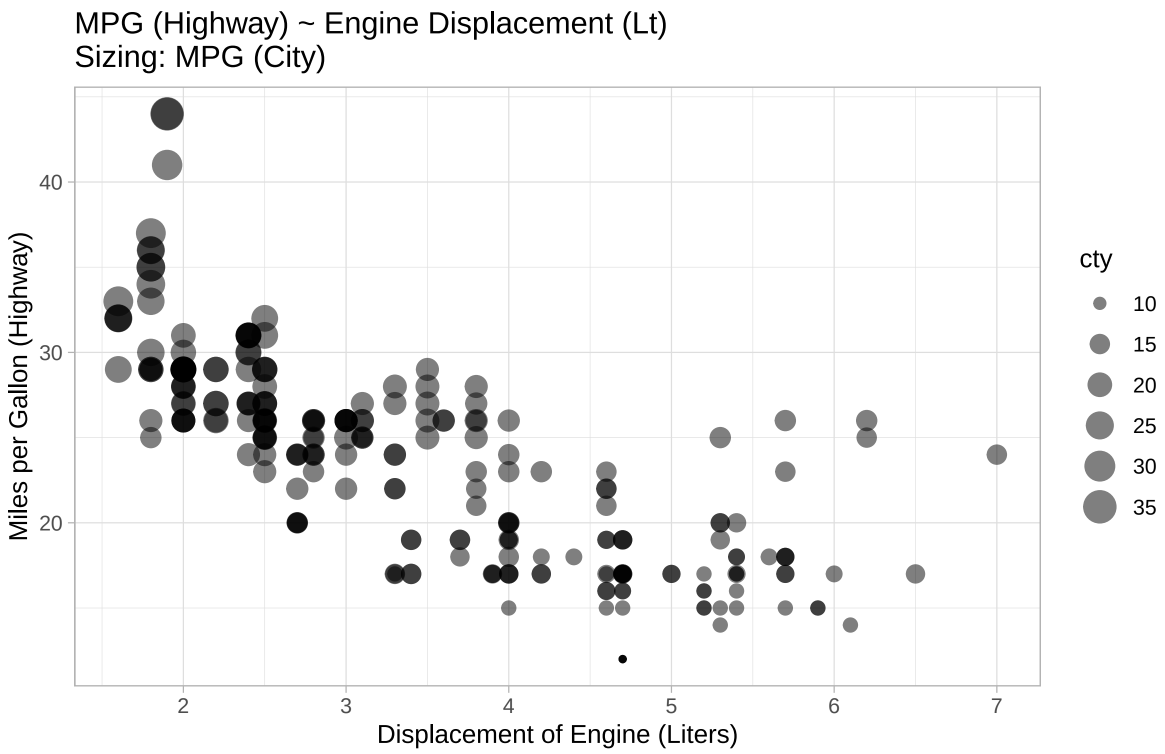
Map a color

```
plot_base + # Using a plot defined about with hwy ~ displ

  # Add mapping and other static aesthetics
  geom_point(aes(color = cyl), size=2) +

  # Update title
  ggtitle(paste0( title_base, 'Coloring: Number of Cylinders' ))
```



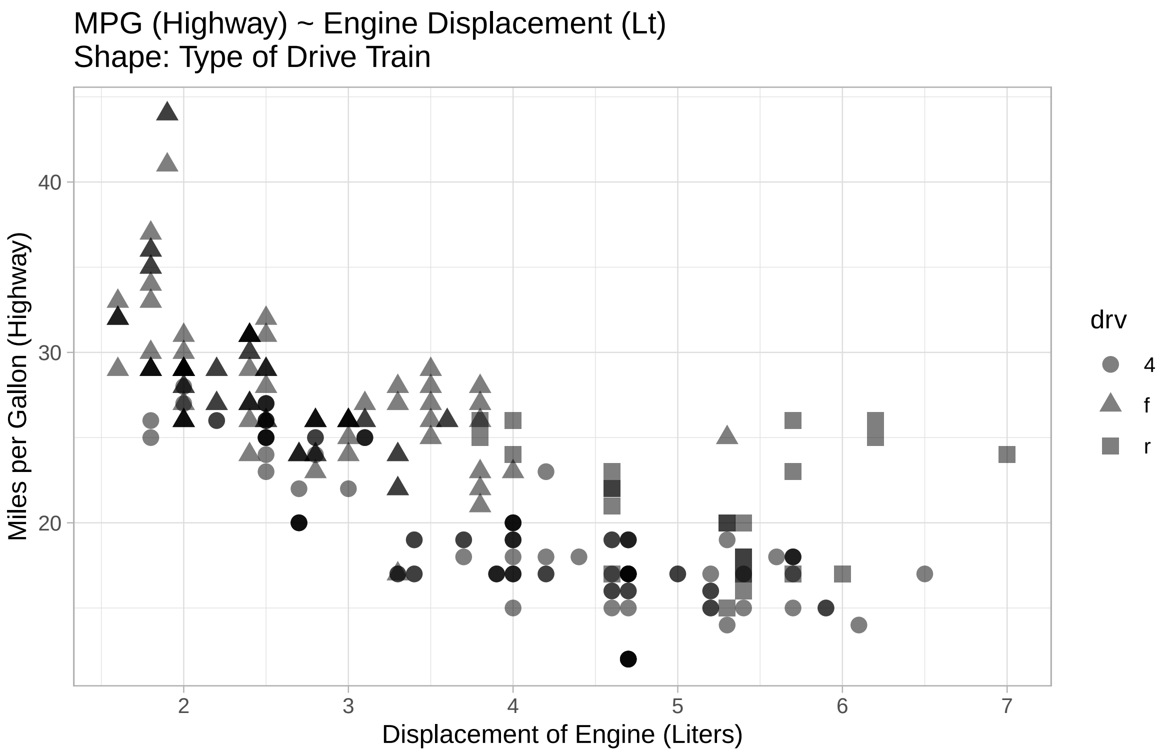
Map a size

```
plot_base + # Using a plot defined about with hwy ~ displ

  # Add mapping and other static aesthetics
  geom_point(aes(size = cty), alpha=0.5) +

  # Update title
  ggtitle(paste0( title_base, 'Sizing: MPG (City)' ))
```



Map a shape

```
plot_base + # Using a plot defined about with hwy ~ displ

  # Add mapping and other static aesthetics
  geom_point(aes(shape = drv), size=3, alpha=0.5) +

  # Update title
  ggtitle(paste0( title_base, 'Shape: Type of Drive Train' ))
```

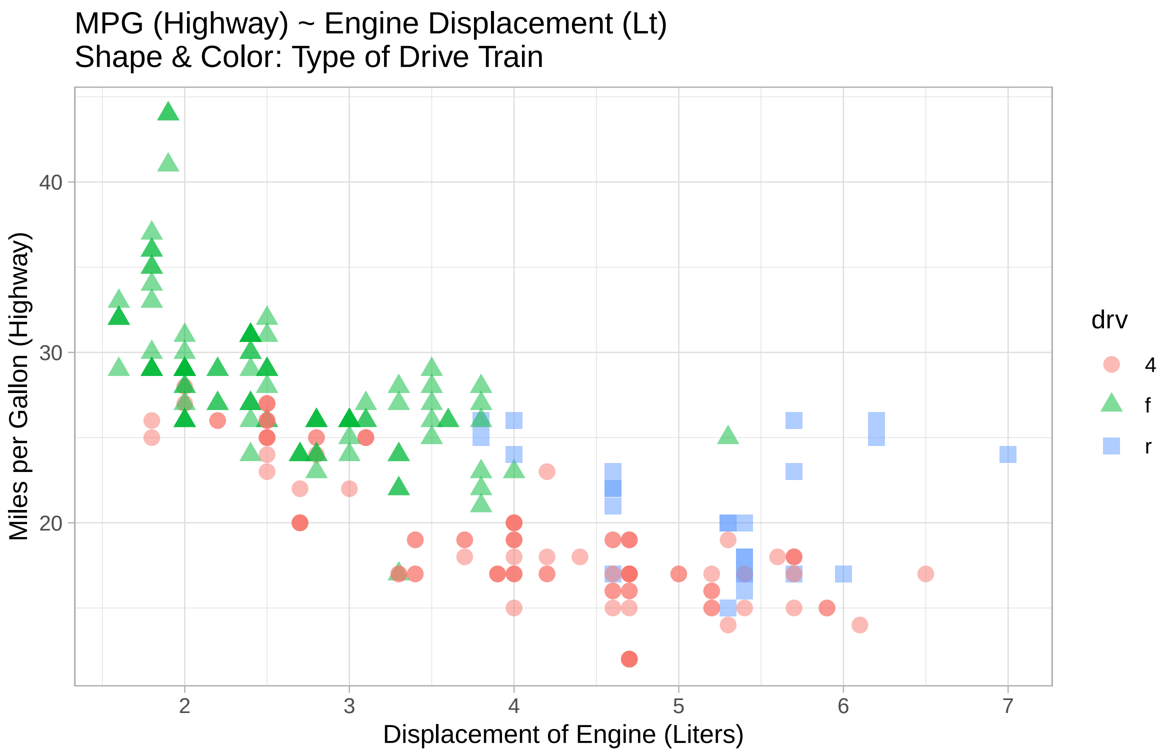


Problem 4

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

Answer: It will condense the legend and it makes it much easier to read. This would be a useful way to analyze the information.

```
plot_base + # Using a plot defined about with hwy ~ displ

  # Add mapping and other static aesthetics
  geom_point(aes(shape = drv,
                 color = drv
                 ), size=3, alpha=0.5) +

  # Update title
  ggtitle(paste0( title_base, 'Shape & Color: Type of Drive Train' ))
```

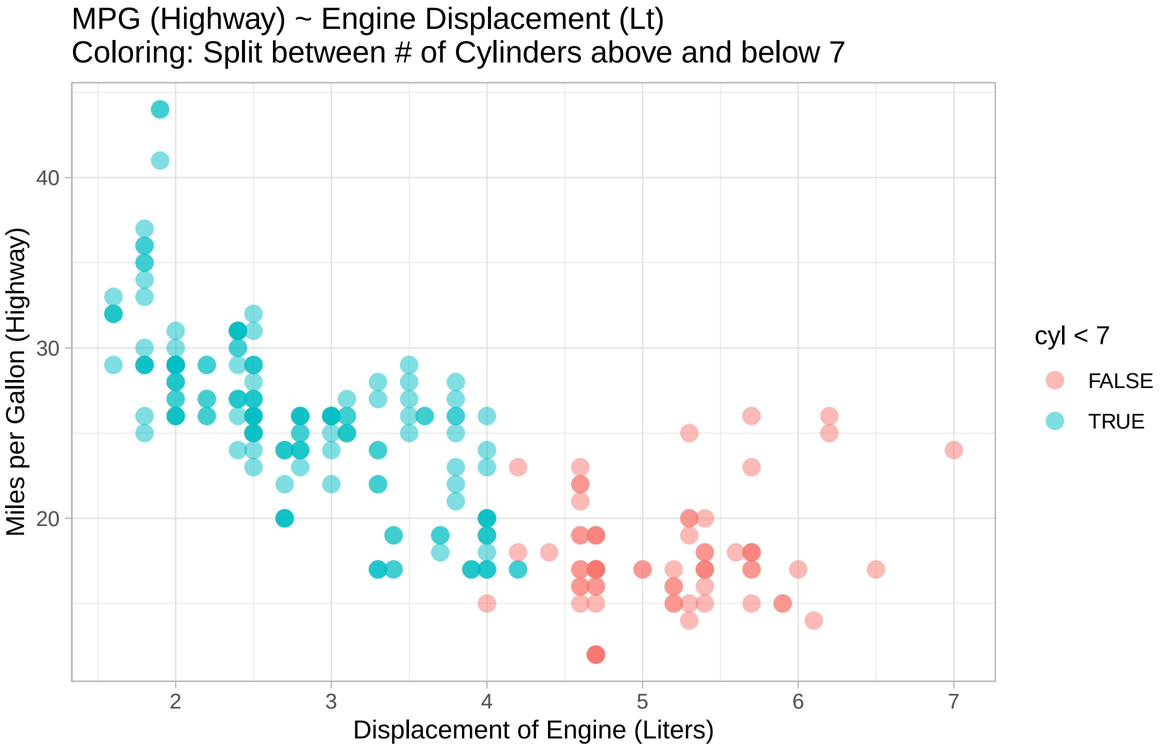


Problem 6

What happens if you map an aesthetic to something other than a variable name, like aes(colour = displ < 5)? Note, you'll also need to specify x and y.

Answer: It will map the points above and below the right hand side of the inequality. For example, below shows when the number of cylinders is < 7. It also makes a note in the legend

```
plot_base + # Using a plot defined about with hwy ~ displ

  # Add mapping and other static aesthetics
  geom_point(aes(color = cyl < 7), size=3, alpha=0.5) +

  # Update title
  ggtitle(paste0( title_base, 'Coloring: Split between # of Cylinders above and below 7' )
```



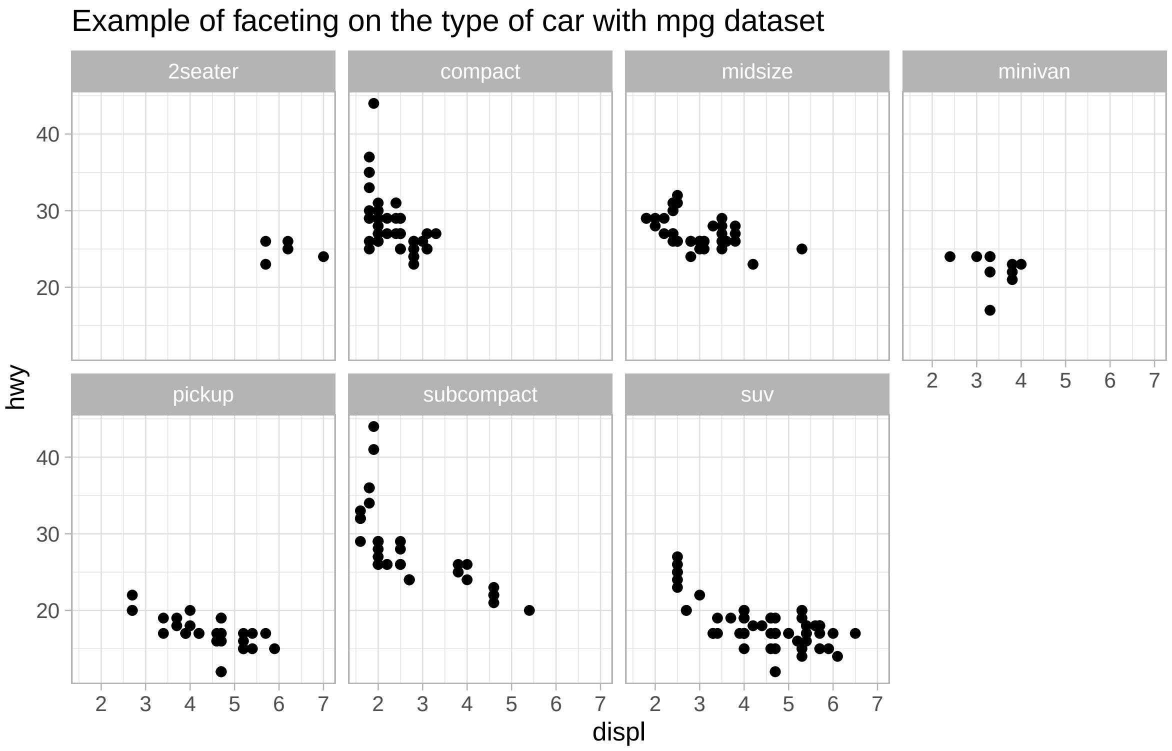
(a) | 3.5.1

Problem 4:

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?

Answer: Faceting allows you to see trends within certain subgroups of a variable. For example, the below graph shows the relationships between the x and y variables given the type of car. You can see clear trends within some of the sub-groups.

```
# Code from website
ggplot(data = mpg) +

  # Create the x/y mapping
  geom_point(mapping = aes(x = displ, y = hwy)) +

  # Facet on type of car
  facet_wrap(~ class, nrow = 2) +
  # Title
  ggtitle('Example of faceting on the type of car with mpg dataset') +
  theme_get()
```

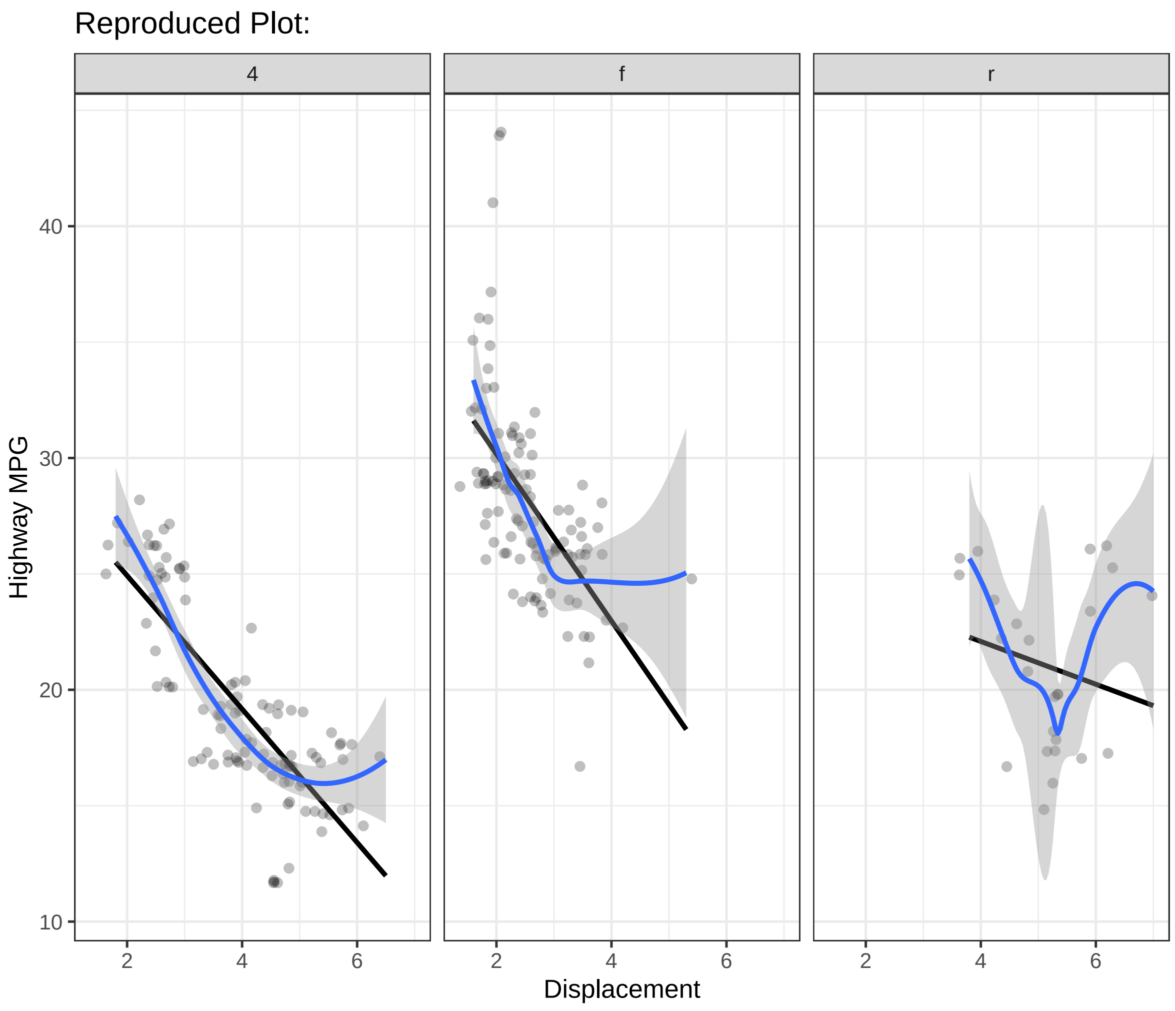


(b): Recreate the Plot

Please see the below plot recreated:

```r
# Create a base plot defined about with hwy ~ displ
mpg %>%

  # hwy vs. cyl
  ggplot( aes(x = displ, y = hwy) ) +

  # Labels
  labs(title = 'Reproduced Plot:',
       x     = 'Displacement',
       y     = 'Highway MPG' ) +

  # Color theme: black an white
  theme_bw() +

  # The jittered points
  geom_jitter(alpha = 0.25,    # Transparency
              width = 0.25) + # Jittering amount

  # Facet on Drive Shaft Type
  facet_grid(. ~ drv) +

  # Linear model line
  geom_smooth(method = lm, fill = NA, color = 'black') +

  # Loess smoother line
  geom_smooth(method = 'loess')
```

```
`geom_smooth()` using formula 'y ~ x'
`geom_smooth()` using formula 'y ~ x'
```


Reproduced Plot:

“‘

Housing Data