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
library(skimr)
library(knitr)
library(GGally) # pairs
library(scales)

# Ridge lines
library(ggridges)
library(viridis)
library(hrbrthemes)
```

ggplot2

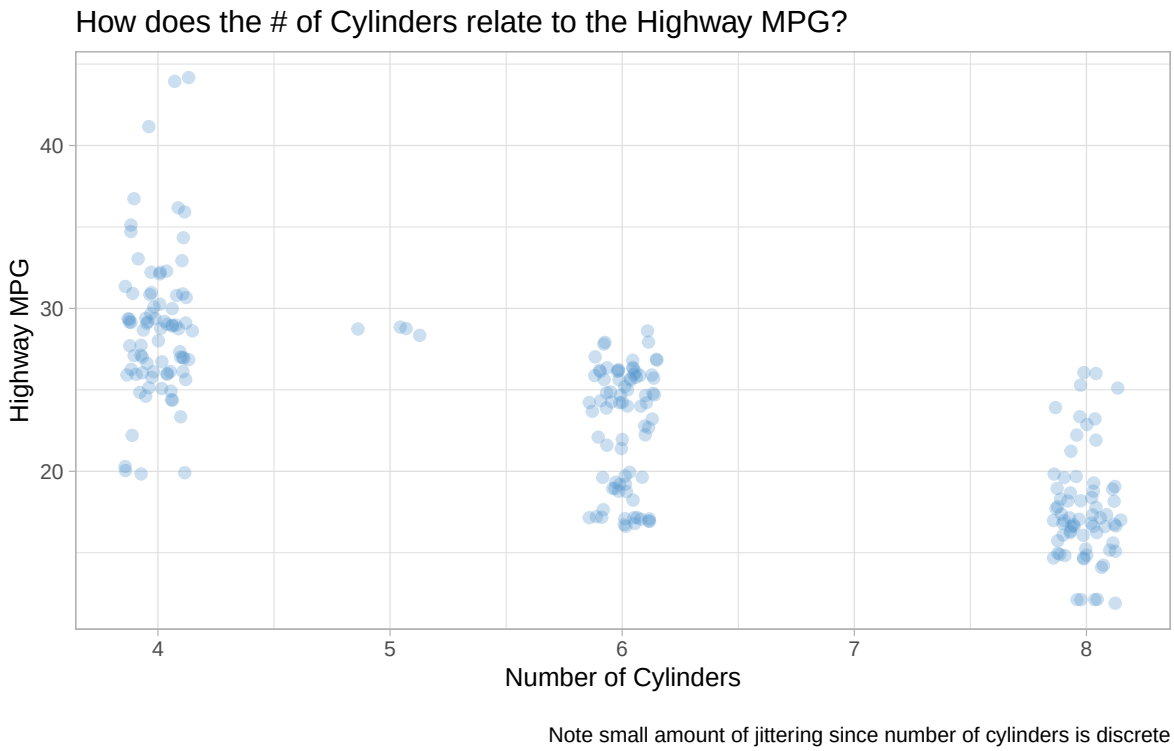
(a) | 3.2.4

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
```



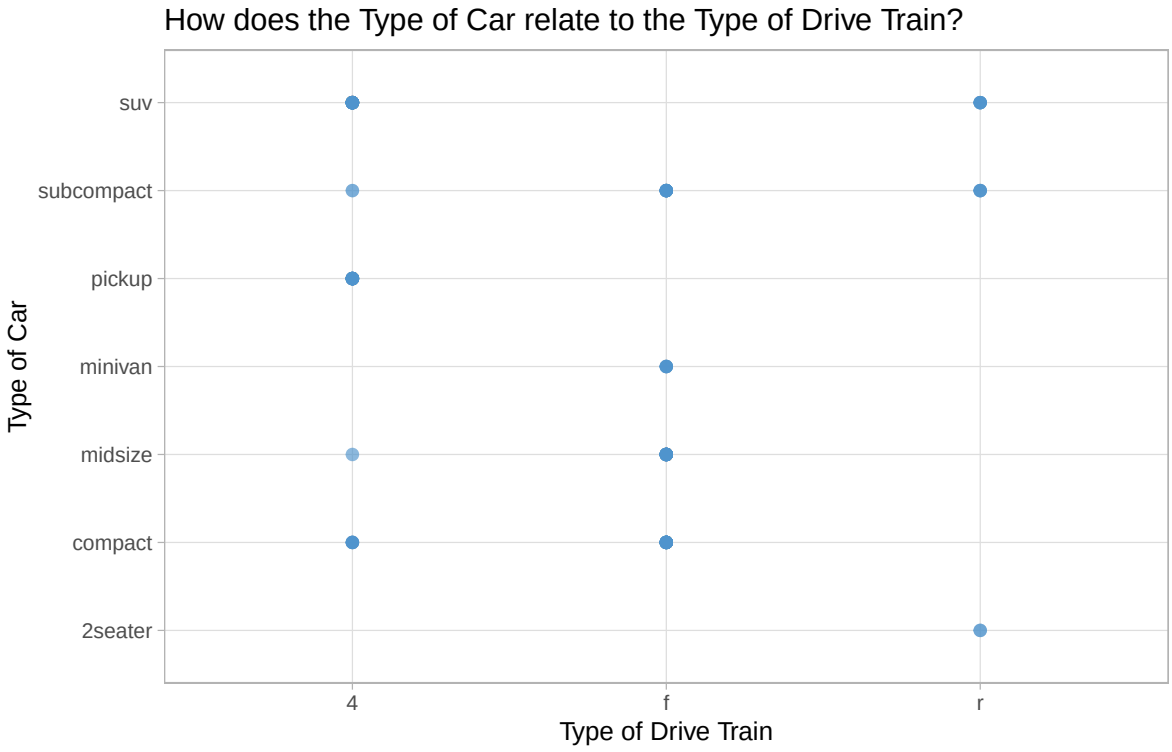
Problem 5

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:

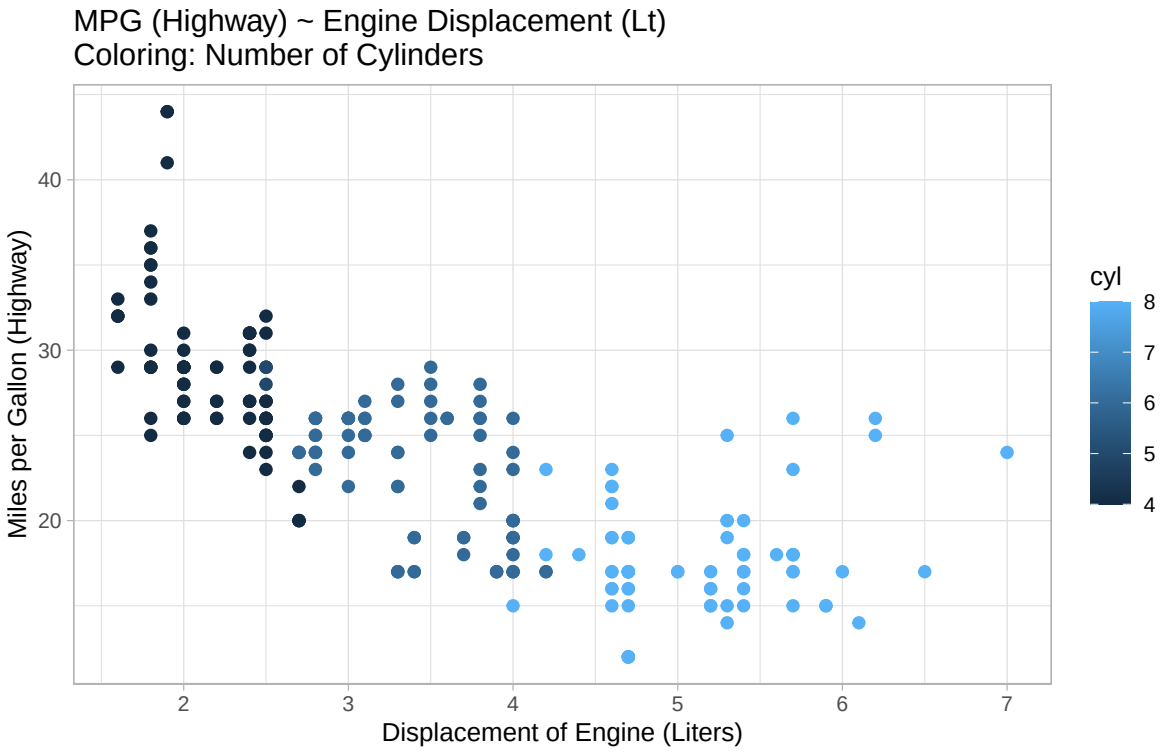
- size: continuous
- color: categorical
- shape: categorical

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
```

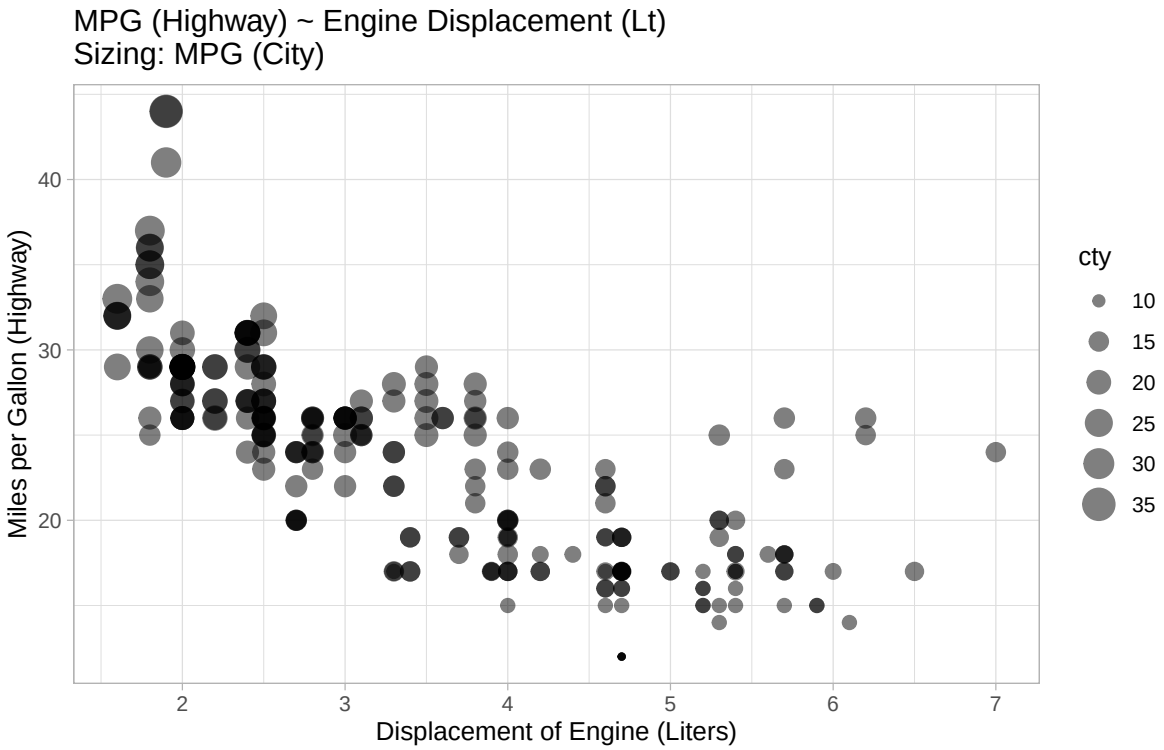
Map a `color`

```
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' ))
```



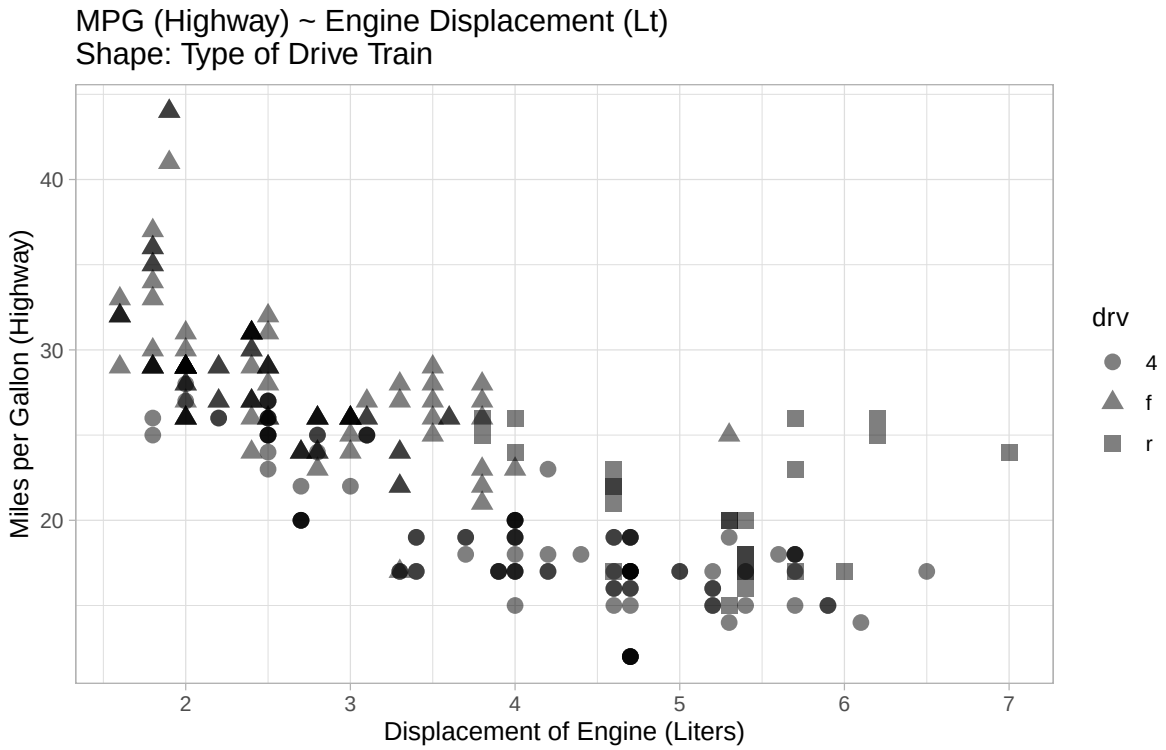
Map a size

```
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' ))
```

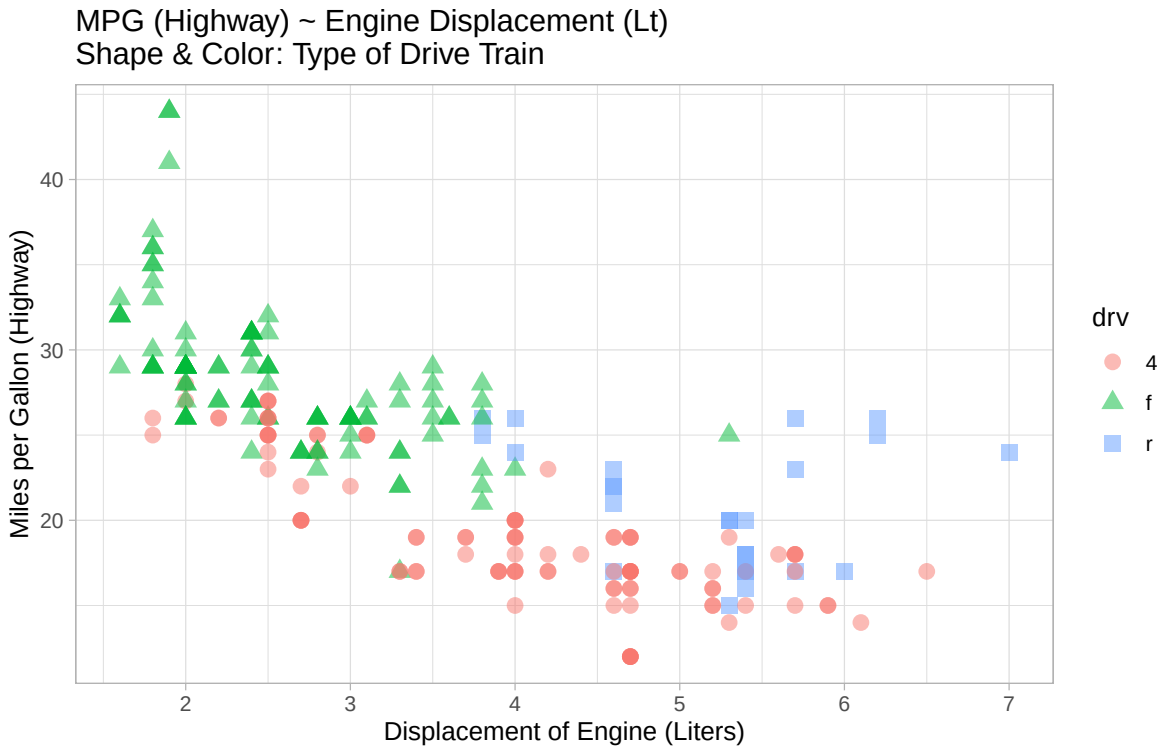


Problem 4

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' ))
```

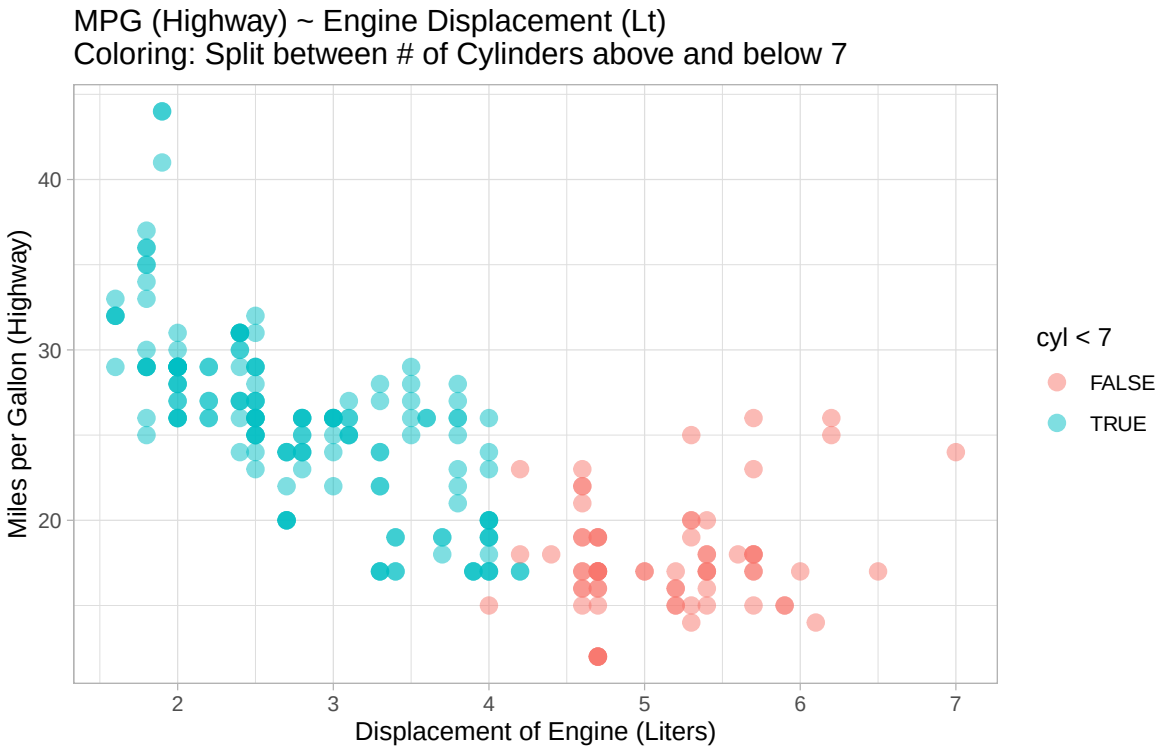


Problem 6

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:

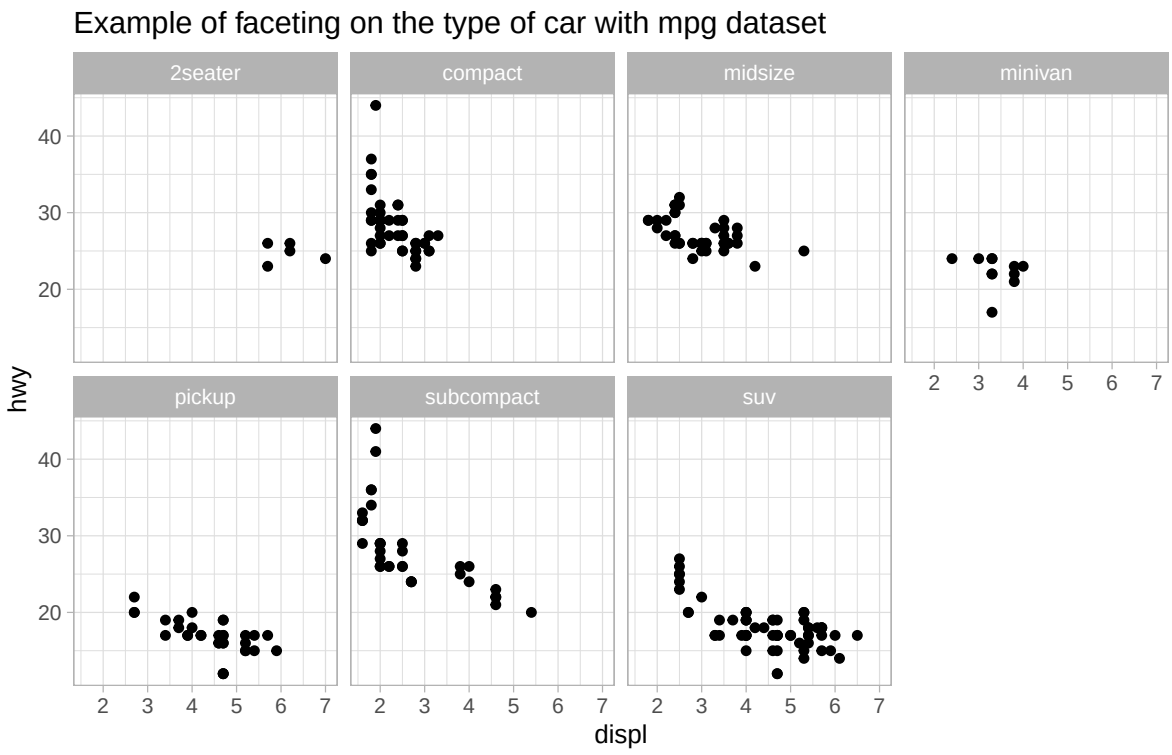
What are the advantages to using faceting instead of the colour aesthetic?

Advantages
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.

Disadvantages
You may want to compare the variables on the same plot. If the data does not overlap, then a facet may not be needed.

How might the balance change if you had a larger dataset?
If you have a lot of data, it may overlap or have disparate clusters. In that case having facets may be useful.

```
# 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()
```

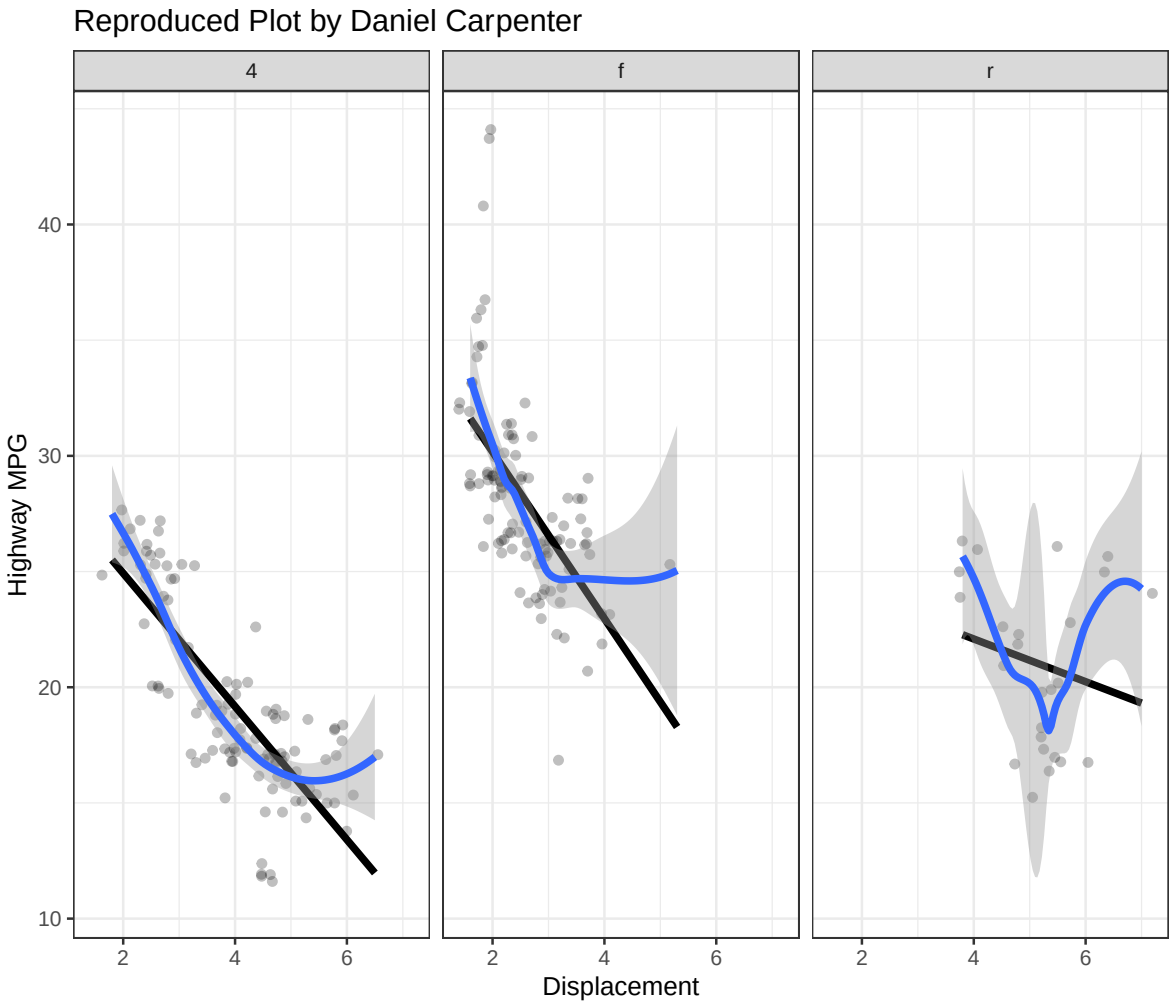


(b): Recreate the Plot

Please see the below plot recreated:

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

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

  # Labels
  labs(title = 'Reproduced Plot by Daniel Carpenter',
       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', size = 1.5) +

  # Loess smoother line
  geom_smooth(method = 'loess', size = 1.5)
```

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


Reproduced Plot by Daniel Carpenter
4
f
r
Highway MPG
40
30
20
10
2
4
6
Displacement

House prices data: Exploratory Data Analysis and Visualization

Pull in Data

```
housing <- read_csv('housingData.csv')
```

```
Rows: 1000 Columns: 74
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (38): MSZoning, Alley, LotShape, LandContour, LotConfig, LandSlope, Neig...
dbl (36): Id, MSSubClass, LotFrontage, LotArea, OverallQual, OverallCond, Ye...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Skimming Data

- Note looks like many `character` or `string` variables have limited number of unique values
- Some data not complete
- Here are some variables that I would imagine to have a large impact on the housing value. Let's look into each and see if we can reduce some of them if highly correlated
    1. LotArea: Lot size in square feet
    2. OverallQual: Rates the overall material and finish of the house
    3. OverallCond: Rates the overall condition of the house (might be correlated with qual)
    4. MSZoning: Identifies the general zoning classification of the sale.
    5. LandContour: Flatness of the property
    6. Condition1: Proximity to various conditions
    7. BldgType: Type of dwelling
    8. HouseStyle: Style of dwelling
    9. YearBuilt: Original construction date
    10. YearRemodAdd: Remodel date (same as construction date if no remodeling or additions)
    11. Foundation: Type of foundation

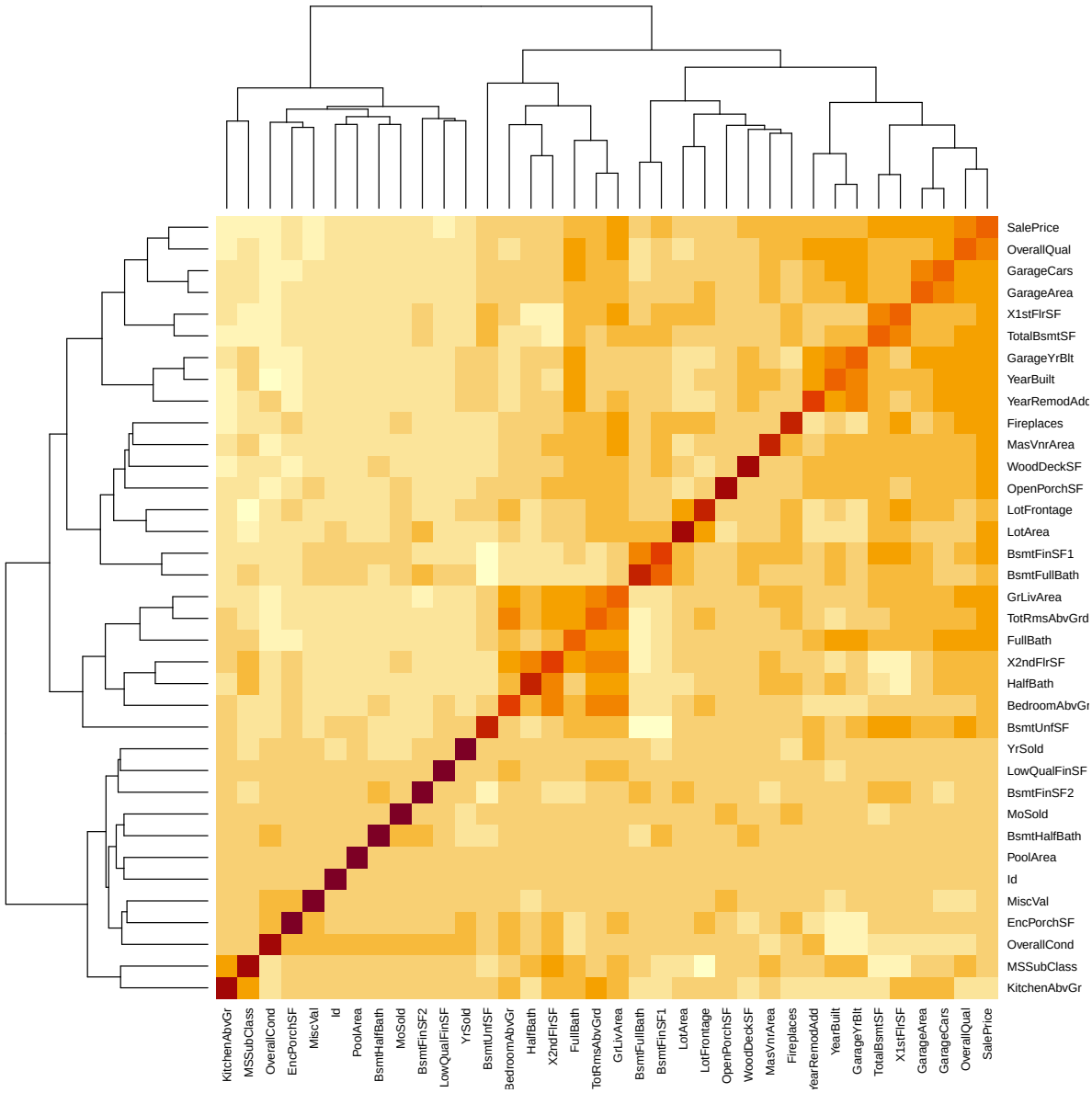
```
# Look only at what we assume may be important
housingImportant <- housing %>%

  # Select only the variables above mentioned
```

```
  select(Id,
         SalePrice,
         YrSold,
         LotArea,
         OverallQual,
         OverallCond,
         MSZoning,
         LandContour,
         Condition1,
         BldgType,
         HouseStyle,
         YearBuilt,
         YearRemodAdd,
         Foundation
         )

# Take a look at the data
skimmed <- skim(housing)

# Check out missing value fields
knitr::kable(skimmed %>% filter(n_missing > 0) )
```

skim_type	skim_variable	n_missing	complete_rate	character.min	character.max	character.empty	character.n_unique	character.whitespace	numeric.mean	numeric.sd	numeric.p0	numeric.p25	numeric.p50	numeric.p75	numeric.p100	numeric.hist
character	Alley	938	0.062	4	4	0	2	0	NA	NA	NA	NA	NA	NA	NA	NA
character	MasVnrType	8	0.996	4	7	0	4	0	NA	NA	NA	NA	NA	NA	NA	NA
character	BsmtQual	31	0.979	2	8	0	4	0	NA	NA	NA	NA	NA	NA	NA	NA
character	BsmtCond	31	0.979	2	8	0	4	0	NA	NA	NA	NA	NA	NA	NA	NA
character	BsmtExposure	31	0.979	2	2	0	4	0	NA	NA	NA	NA	NA	NA	NA	NA
character	BsmtFinType1	31	0.979	3	3	0	6	0	NA	NA	NA	NA	NA	NA	NA	NA
character	BsmtFinType2	31	0.979	3	3	0	6	0	NA	NA	NA	NA	NA	NA	NA	NA
character	Electrical	1	0.999	5	5	0	4	0	NA	NA	NA	NA	NA	NA	NA	NA
character	FireplaceQu	466	0.534	2	8	0	5	0	NA	NA	NA	NA	NA	NA	NA	NA
character	GarageType	53	0.947	6	7	0	6	0	NA	NA	NA	NA	NA	NA	NA	NA
character	GarageFinish	53	0.947	3	3	0	3	0	NA	NA	NA	NA	NA	NA	NA	NA
character	GarageQual	53	0.947	2	8	0	5	0	NA	NA	NA	NA	NA	NA	NA	NA
character	GarageCond	53	0.947	2	8	0	5	0	NA	NA	NA	NA	NA	NA	NA	NA
character	PoolQC	998	0.002	2	2	0	2	0	NA	NA	NA	NA	NA	NA	NA	NA
character	Fence	805	0.195	4	5	0	4	0	NA	NA	NA	NA	NA	NA	NA	NA
character	MiscFeature	966	0.034	4	4	0	2	0	NA	NA	NA	NA	NA	NA	NA	NA
numeric	LotFrontage	207	0.793	NA	NA	NA	NA	NA	68.74523	23.19289	21	58	68	80.00313		
numeric	MasVnrArea	8	0.996	NA	NA	NA	NA	NA	95.41767	177.3182	0	0	0	146.25	1600	

skim_type	skim_variable	n_missing	complete_rate	character.min	character.max	character.empty	character.n_unique	character.whitespace	numeric.mean	numeric.sd	numeric.p0	numeric.p25	numeric.p50	numeric.p75	numeric.p100	numeric.hist
numeric	GarageYrBlt	53	0.96	NA	NA	NA	NA	NA	1976.23	25.57	1896	1960	1977	1999.00	2009	

```
# heatmap of the numeric data for non-null values
# Generally seems like these are for the categorical data that explains a
# Unique attribute of the house, like if the house has a basement, pool, fense or not.
housingNumeric <- housing %>% select_if(is.numeric) %>% drop_na()

correlationMatrix <- cor(housingNumeric )

heatmap(correlationMatrix)
```




```r
# get top 10 highest correlated variables

## Sort data on sale price descending
corMatrixSorted <- as.data.frame(correlationMatrix) %>% arrange(desc(SalePrice))

corVarsTop10 <- rownames(corMatrixSorted)[2:11] # 2:11 since exclude sale price variable
```

```r
# What are the top 10 (sorted by highest correlation)?
kable(corVarsTop10)
```

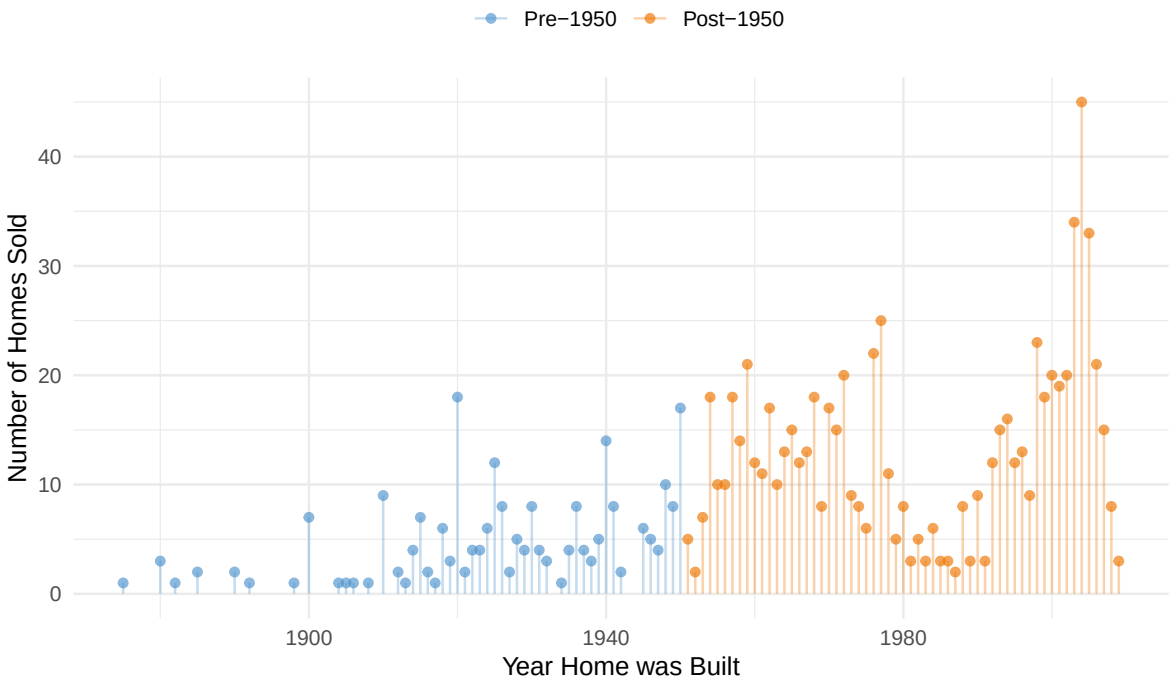
x
OverallQual
GrLivArea
TotalBsmtSF
GarageCars
X1stFlrSF
GarageArea
FullBath
TotRmsAbvGrd
YearBuilt
YearRemodAdd

Reviewing Potential Visualizations

Are newer homes more popular?

```r
# unique(housing$YrSold)
YEAR_THRESHOLD = 1950

housing %>% # using the housing data

  # Get a count of homes sold by year built
  group_by(YearBuilt) %>%
  summarise(NumSold = n() ) %>%

  # Start ggplot with x axis being yearbuilt
  ggplot(aes(x = YearBuilt,
             color = YearBuilt > YEAR_THRESHOLD
             )) +

      # Labels and Titles
      labs(title = paste('Most homes Sold in Sample were Built after', YEAR_THRESHOLD),
           x = 'Year Home was Built',
           y = 'Number of Homes Sold') +

      # Build a lollipop chart
```

```r
# Basics here: https://r-graph-gallery.com/300-basic-lollipop-plot.html
geom_segment(aes(x=YearBuilt, xend=YearBuilt, y=0, yend=NumSold),
             alpha = 0.33) +
geom_point(aes(y = NumSold),
           alpha = 0.66) +

# Diverge on colors based on the YEAR_THRESHOLD variable
# Splits based on the year built
scale_color_manual(values = c('steelblue3', 'darkorange2'),
                   labels = paste0(c('Pre-', 'Post-'), YEAR_THRESHOLD) ) +

# Themes
theme_minimal() +
theme(legend.title    = element_blank(), # Format the legend nicer
      legend.position = 'top')
```



Are there any changes happening to sale price overtime?

```
base_ridgeline <- housing %>%
  ggplot(aes(y     = YrSold,
             group = YrSold,
             x     = SalePrice,
             color = YearBuilt > YEAR_THRESHOLD,
             fill  = YearBuilt > YEAR_THRESHOLD
             )
         ) +

  # Labels
  labs(title    = 'Distribution of Yearly Home Prices at Sale Date Remain Steady',
       subtitle = paste('Note Homes Built after', YEAR_THRESHOLD, 'sell for Less'),
       x        = 'Sale Price of Home (USD)',
       y        = 'Year Home Sold') +

  # Ridge Line Density Plots
  # More here: https://r-graph-gallery.com/294-basic-ridgeline-plot.html#color
  geom_density_ridges_gradient(scale = 3, rel_min_height = 0.01,
                               alpha = 0.5) +

  # Formatting of axis as comma
  scale_x_continuous(labels = comma) +

  # Themes
  theme_minimal() +
    theme(
      legend.position = "top",
      legend.title    = element_blank(),
      panel.spacing   = unit(0.1, "lines"),
      strip.text      = element_blank()
    ) +

  # Facet on the year threshold
  facet_grid(. ~ YearBuilt > YEAR_THRESHOLD) +

  # Diverge on colors based on the YEAR_THRESHOLD variable
  # Splits based on the year built
  scale_color_manual(values = c('steelblue3', 'darkorange2'),
                     labels = paste0(c('Pre-', 'Post-'), YEAR_THRESHOLD) ) +
```

```
  scale_fill_manual( values = c('lightsteelblue2', 'sandybrown'),
                     labels = paste0(c('Pre-', 'Post-'), YEAR_THRESHOLD) )


base_ridgeline # display
```

```
Picking joint bandwidth of 13700
```

```
Picking joint bandwidth of 18700
```