Worksheet 6

Kylene Joy Yanguas

2022-11-25

#1. How many columns are in mpg dataset? How about the number of rows? Show the #codes and its result.

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data(mpg)
as.data.frame(data(mpg))
```

```
##   data(mpg)
## 1       mpg
```

```
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

```
nrow(mpg)
```

```
## [1] 234
```

```
ncol(mpg)
```

```
## [1] 11
```

#The are 234 rows and 11 columns in mpg dataset.

#2. Which manufacturer has the most models in this data set? Which model has the most #variations? Ans:

```
manu1 <- mpg %>% group_by(manufacturer) %>%  tally(sort = TRUE)
```

#Dodge Manufacturer has the most models in this data set with 37 models.
#Toyota Manufacturer has 6 variation namely; 4runner 4wd,camry,camry solara,corolla,
#land cruiser wagon 4wd, toyota tacoma 4wd which has the most variation.

#a. Group the manufacturers and find the unique models. Copy the codes and result.

```
data <- mpg
data_mpg <- data %>% group_by(manufacturer, model) %>%
         distinct() %>% count()
   data_mpg
```

```
## # A tibble: 38 x 3
## # Groups:   manufacturer, model [38]
##    manufacturer model                  n
##    <chr>        <chr>              <int>
##  1 audi         a4                     7
##  2 audi         a4 quattro             8
##  3 audi         a6 quattro             3
##  4 chevrolet    c1500 suburban 2wd     4
##  5 chevrolet    corvette               5
##  6 chevrolet    k1500 tahoe 4wd        4
##  7 chevrolet    malibu                 5
##  8 dodge        caravan 2wd            9
##  9 dodge        dakota pickup 4wd      8
## 10 dodge        durango 4wd            6
## # ... with 28 more rows
```

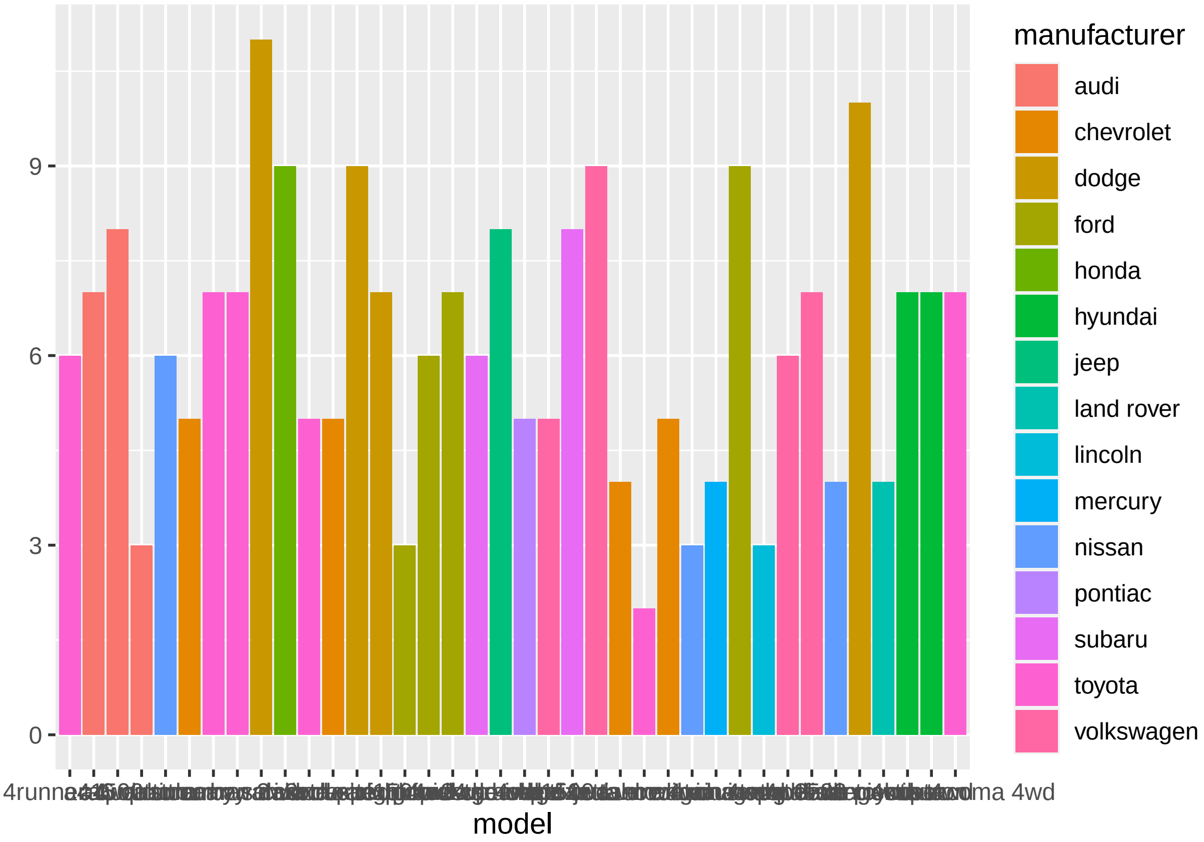
```
colnames(data_mpg) <- c("Manufacturer", "Model","Counts")
   data_mpg
```

```
## # A tibble: 38 x 3
## # Groups:   Manufacturer, Model [38]
##    Manufacturer Model              Counts
##    <chr>        <chr>               <int>
##  1 audi         a4                      7
##  2 audi         a4 quattro              8
##  3 audi         a6 quattro              3
##  4 chevrolet    c1500 suburban 2wd      4
##  5 chevrolet    corvette                5
##  6 chevrolet    k1500 tahoe 4wd         4
##  7 chevrolet    malibu                  5
##  8 dodge        caravan 2wd             9
##  9 dodge        dakota pickup 4wd       8
## 10 dodge        durango 4wd             6
## # ... with 28 more rows
```

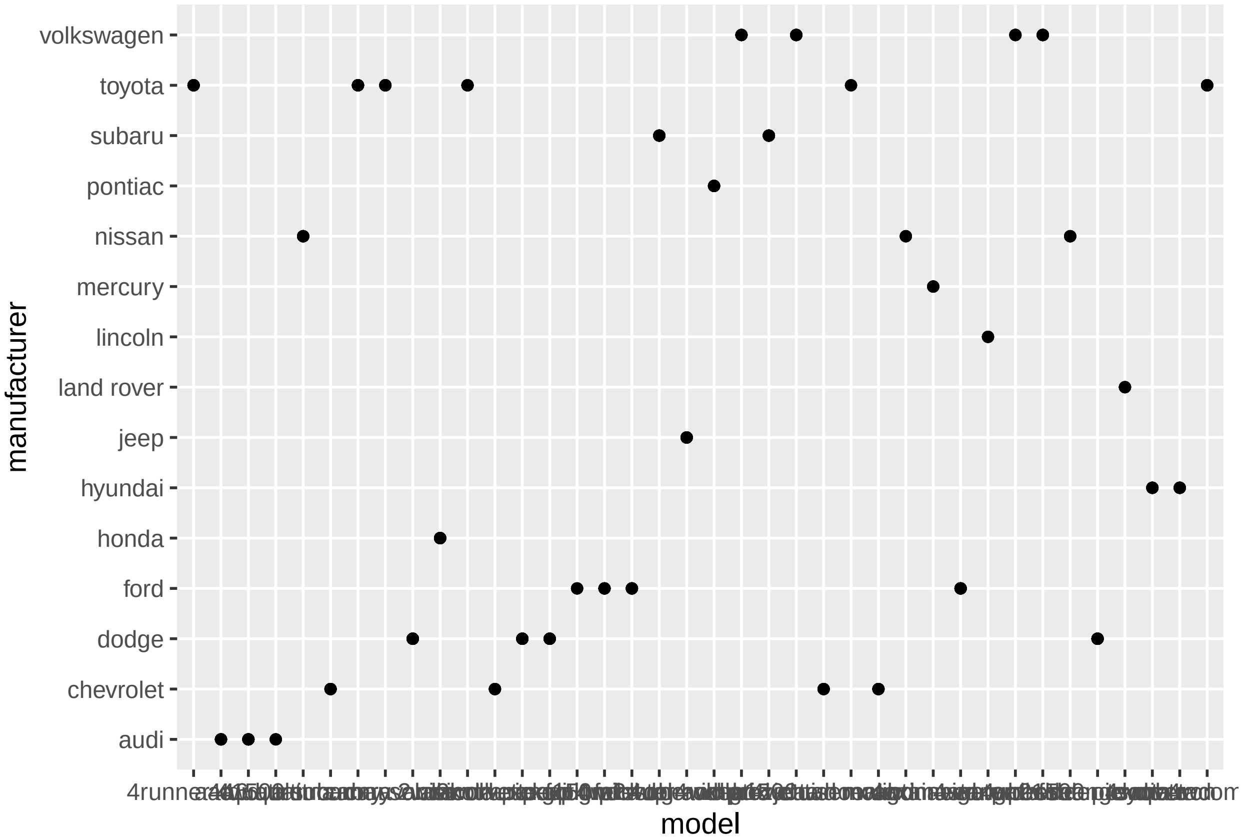
#b. Graph the result by using plot() and ggplot(). Write the codes and its result.

```
qplot(data = mpg, geom = "bar", model, fill=manufacturer)
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
```



```
ggplot(mpg, aes(model, manufacturer)) + geom_point()
```



#3. Same data set will be used. You are going to show the relationship of the model and #the manufacturer.

```
data <- mpg

data_mpg1 <- data %>% group_by(manufacturer, model) %>%
  distinct() %>% count()
data_mpg1
```

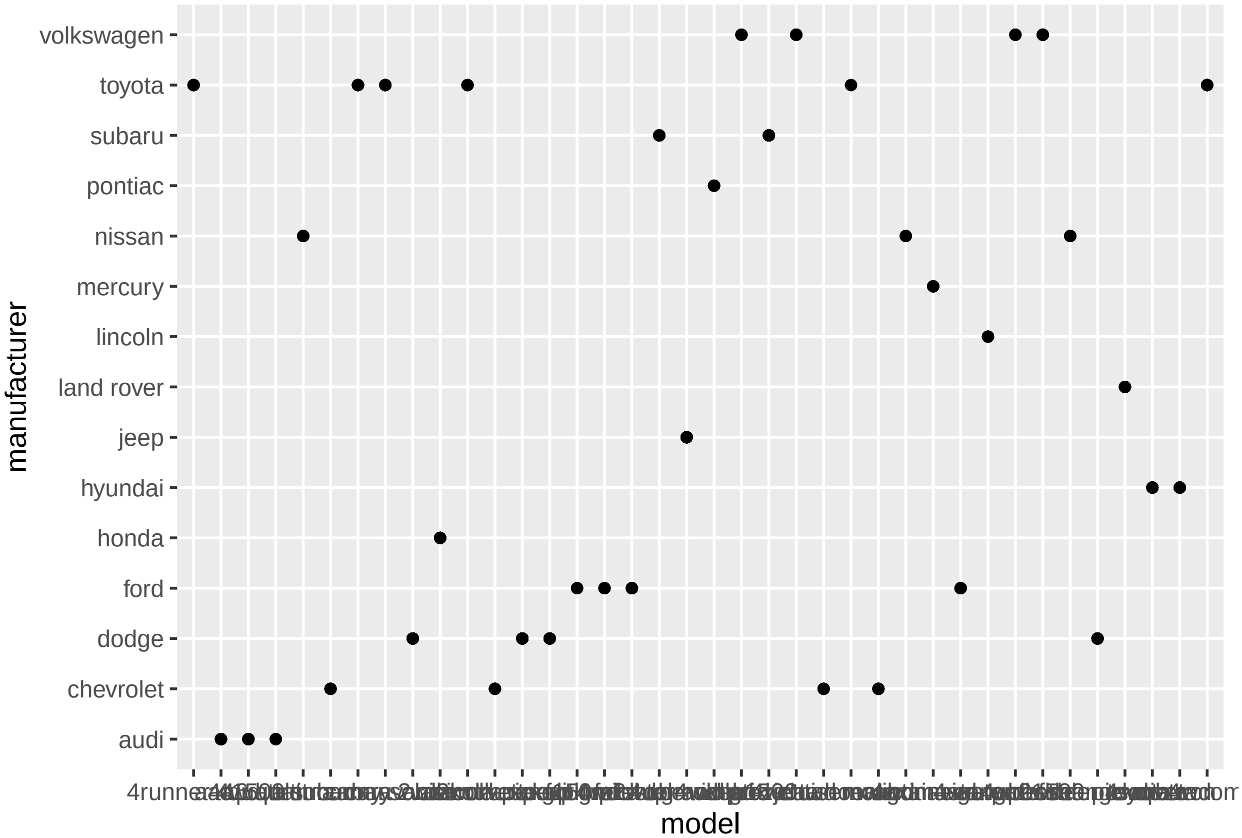
```
## # A tibble: 38 x 3
## # Groups:   manufacturer, model [38]
##    manufacturer model                  n
##    <chr>        <chr>              <int>
##  1 audi         a4                     7
##  2 audi         a4 quattro             8
##  3 audi         a6 quattro             3
##  4 chevrolet    c1500 suburban 2wd     4
##  5 chevrolet    corvette               5
##  6 chevrolet    k1500 tahoe 4wd        4
##  7 chevrolet    malibu                 5
##  8 dodge        caravan 2wd            9
##  9 dodge        dakota pickup 4wd      8
## 10 dodge        durango 4wd            6
## # ... with 28 more rows
```

```
colnames(data_mpg1) <- c("Manufacturer", "Model")
data_mpg1
```

```
## # A tibble: 38 x 3
## # Groups:   Manufacturer, Model [38]
##    Manufacturer Model                  ``
##    <chr>        <chr>               <int>
##  1 audi         a4                      7
##  2 audi         a4 quattro              8
##  3 audi         a6 quattro              3
##  4 chevrolet    c1500 suburban 2wd      4
##  5 chevrolet    corvette                5
##  6 chevrolet    k1500 tahoe 4wd         4
##  7 chevrolet    malibu                  5
##  8 dodge        caravan 2wd             9
##  9 dodge        dakota pickup 4wd       8
## 10 dodge        durango 4wd             6
## # ... with 28 more rows
```

#a. What does ggplot(mpg, aes(model, manufacturer)) + geom_point() show?

```
ggplot(mpg, aes(model, manufacturer)) + geom_point()
```



#It shows the scatter plot of the model and manufacturer of mpg data set.

#b. For you, is it useful? If not, how could you modify the data to make it more #informative? #It is useful, but for somehow the model name below isn't clear enough to modify #the data clearly and accurately. I preferred to use the bar graph to modify the data to make it more informative.

#4. Using the pipe (%>%), group the model and get the number of cars per model. Show #codes and its result.

```
data <- mpg
data_mpg2 <- data %>% group_by(manufacturer, model) %>%
  distinct() %>% count()
```

```r
data_mpg2
```

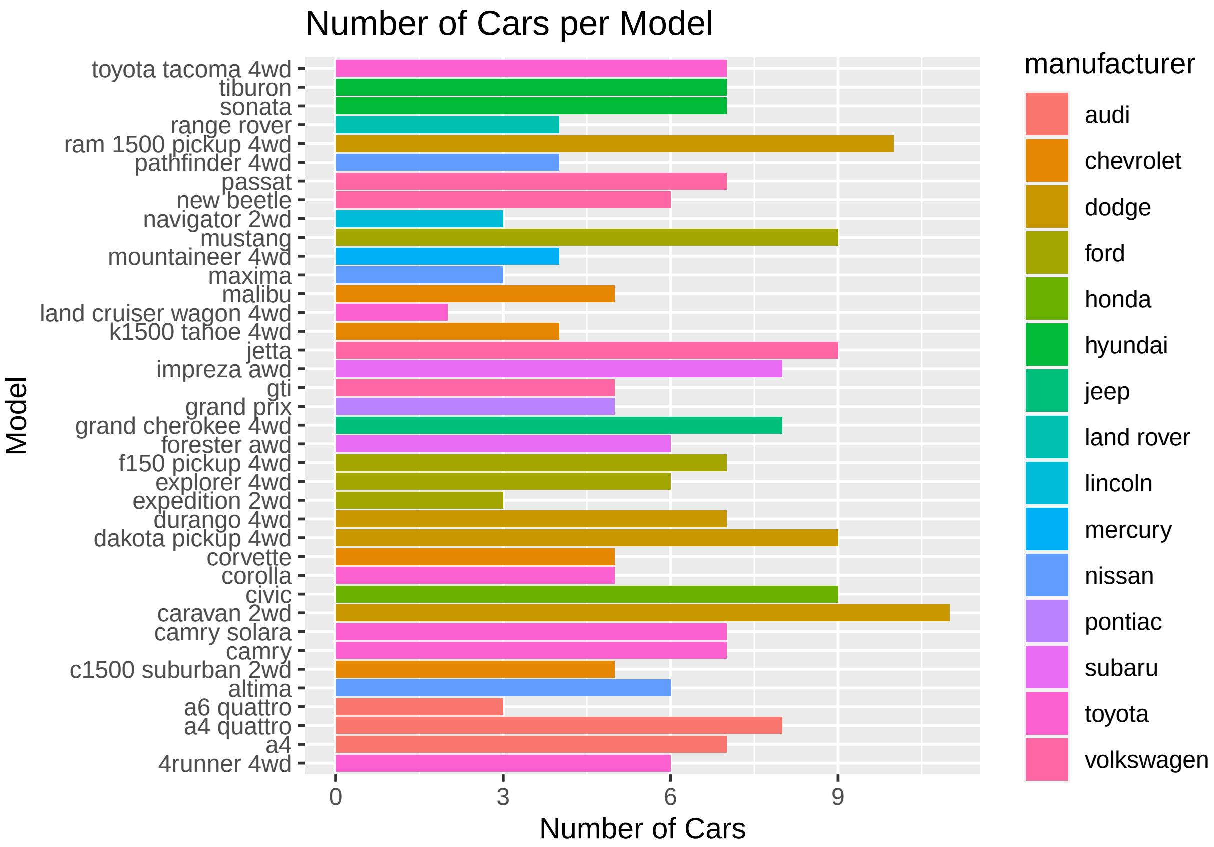
```
## # A tibble: 38 x 3
## # Groups:   manufacturer, model [38]
##    manufacturer model                  n
##    <chr>        <chr>              <int>
##  1 audi         a4                     7
##  2 audi         a4 quattro             8
##  3 audi         a6 quattro             3
##  4 chevrolet    c1500 suburban 2wd     4
##  5 chevrolet    corvette               5
##  6 chevrolet    k1500 tahoe 4wd        4
##  7 chevrolet    malibu                 5
##  8 dodge        caravan 2wd            9
##  9 dodge        dakota pickup 4wd      8
## 10 dodge        durango 4wd            6
## # ... with 28 more rows
```

```r
colnames(data_mpg2) <- c("Model", "Counts")
data_mpg2
```

```
## # A tibble: 38 x 3
## # Groups:   Model, Counts [38]
##    Model     Counts                ``
##    <chr>     <chr>              <int>
##  1 audi      a4                     7
##  2 audi      a4 quattro             8
##  3 audi      a6 quattro             3
##  4 chevrolet c1500 suburban 2wd     4
##  5 chevrolet corvette               5
##  6 chevrolet k1500 tahoe 4wd        4
##  7 chevrolet malibu                 5
##  8 dodge     caravan 2wd            9
##  9 dodge     dakota pickup 4wd      8
## 10 dodge     durango 4wd            6
## # ... with 28 more rows
```

#a. Plot using the geom_bar() + coord_flip() just like what is shown below. Show #codes and its result.

```r
qplot(model,
      data = mpg, main = "Number of Cars per Model",
      xlab = "Model",
      ylab = "Number of Cars",
      geom = "bar", fill = manufacturer) + coord_flip()
```



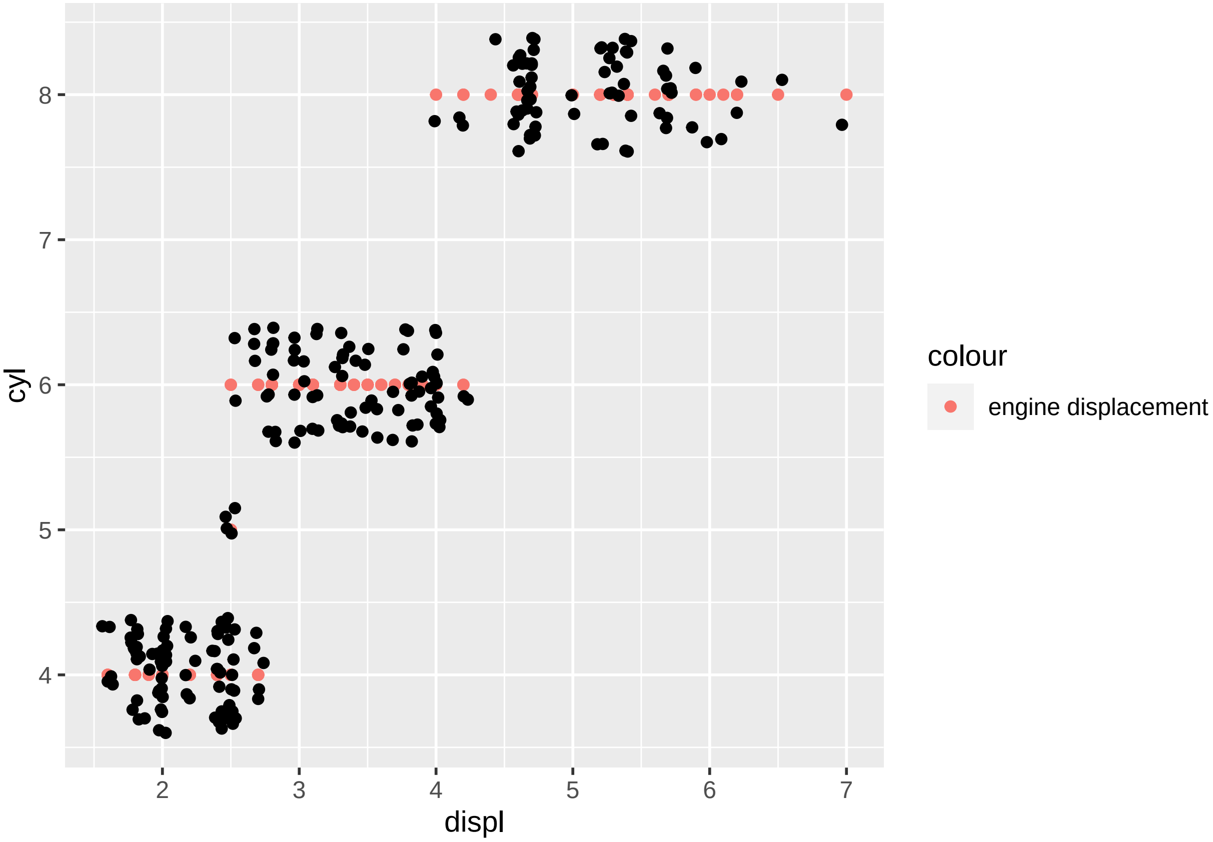
#b. Use only the top 20 observations. Show code and results.

```
head(mpg, n=20)
```

```
## # A tibble: 20 x 11
##    manufacturer model       displ  year   cyl trans drv     cty   hwy fl    class
##    <chr>        <chr>       <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
##  1 audi         a4            1.8  1999     4 auto~ f        18    29 p     comp~
##  2 audi         a4            1.8  1999     4 manu~ f        21    29 p     comp~
##  3 audi         a4            2    2008     4 manu~ f        20    31 p     comp~
##  4 audi         a4            2    2008     4 auto~ f        21    30 p     comp~
##  5 audi         a4            2.8  1999     6 auto~ f        16    26 p     comp~
##  6 audi         a4            2.8  1999     6 manu~ f        18    26 p     comp~
##  7 audi         a4            3.1  2008     6 auto~ f        18    27 p     comp~
##  8 audi         a4 quattro    1.8  1999     4 manu~ 4        18    26 p     comp~
##  9 audi         a4 quattro    1.8  1999     4 auto~ 4        16    25 p     comp~
## 10 audi         a4 quattro    2    2008     4 manu~ 4        20    28 p     comp~
## 11 audi         a4 quattro    2    2008     4 auto~ 4        19    27 p     comp~
## 12 audi         a4 quattro    2.8  1999     6 auto~ 4        15    25 p     comp~
## 13 audi         a4 quattro    2.8  1999     6 manu~ 4        17    25 p     comp~
## 14 audi         a4 quattro    3.1  2008     6 auto~ 4        17    25 p     comp~
## 15 audi         a4 quattro    3.1  2008     6 manu~ 4        15    25 p     comp~
## 16 audi         a6 quattro    2.8  1999     6 auto~ 4        15    24 p     mids~
## 17 audi         a6 quattro    3.1  2008     6 auto~ 4        17    25 p     mids~
## 18 audi         a6 quattro    4.2  2008     8 auto~ 4        16    23 p     mids~
## 19 chevrolet    c1500 sub~    5.3  2008     8 auto~ r        14    20 r     suv
## 20 chevrolet    c1500 sub~    5.3  2008     8 auto~ r        11    15 e     suv
```

#5. Plot the relationship between cyl - number of cylinders and displ - engine displacement using geom_point with aesthetic colour = engine displacement. #Title should be “Relationship between No. of Cylinders and Engine Displacement”. #a. Show the codes and its result.

```
ggplot(data = mpg , mapping = aes(x = displ, y = cyl,
                              main = "Relationship between No of Cylinders and Engine Displacement")) +
geom_point(mapping=aes(colour = "engine displacement")) + geom_jitter()
```

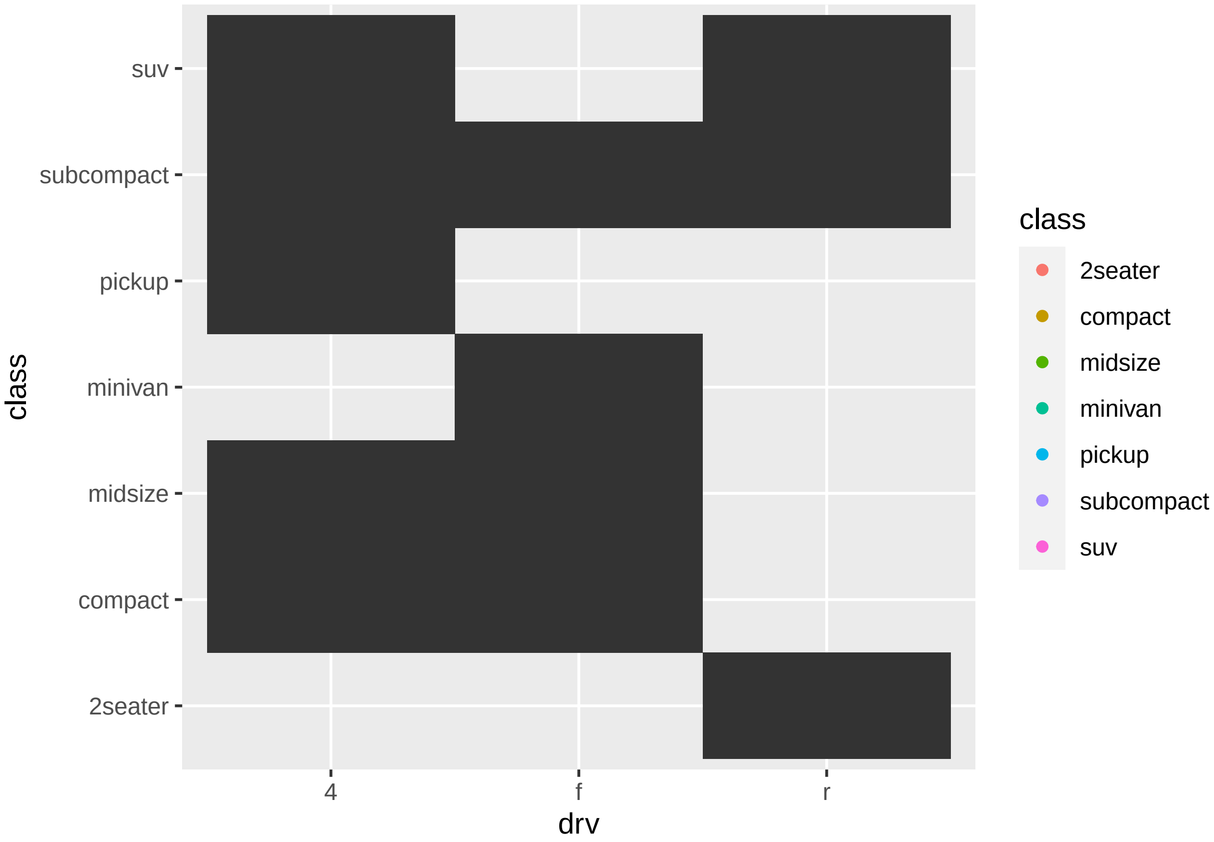


#b. How would you describe its relationship? #The data shows that the graph is jittered. The color pink indicates as engine displacement which on a straight horizontal position.

#6. Get the total number of observations for drv - type of drive train (f = front-wheel drive, r = rear wheel drive, 4 = 4wd) and class - type of class (Example: suv, 2 seater, etc.). #Plot using the geom_tile() where the number of observations for class be used as a fill for aesthetics.

#a. Show the codes and its result for the narrative in #6.

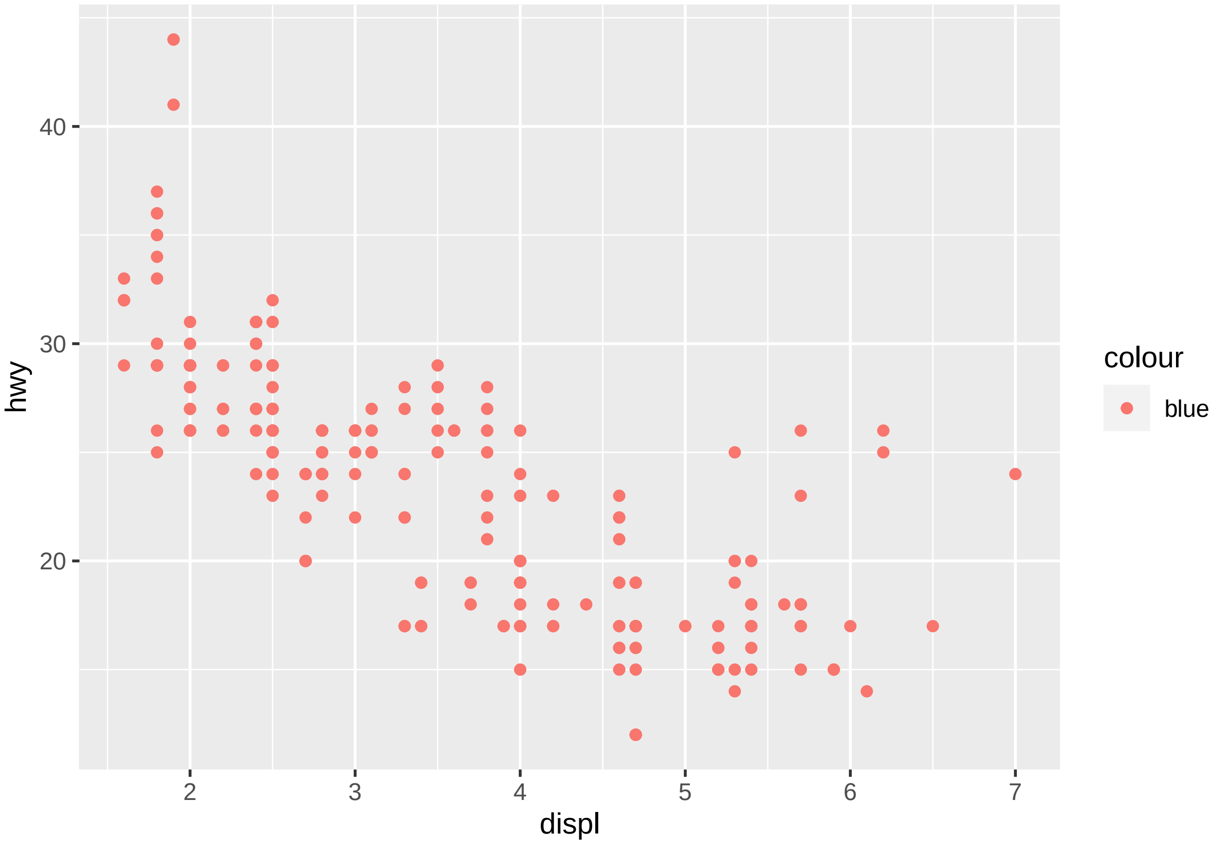
```
ggplot(data = mpg, mapping = aes(x = drv, y = class)) +
  geom_point(mapping=aes(color=class)) +
  geom_tile()
```



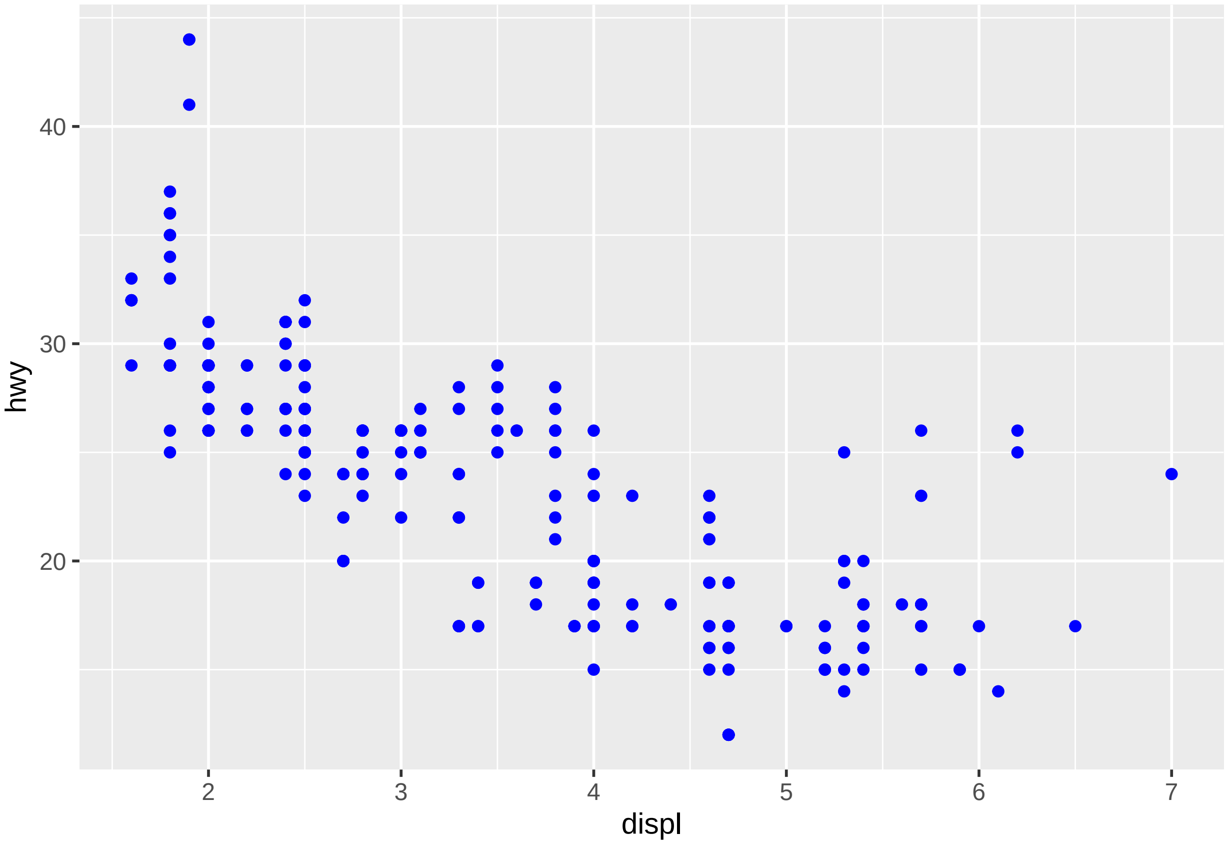
#b. Interpret the result. #The data shows that the total number of observations for drv - type of drive train are covered with black #were mapped using the mapping geometric point graph.

#7. Discuss the difference between these codes. Its outputs for each are shown below.

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



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



#8. Try to run the command ?mpg. What is the result of this command?

```
?mpg
```

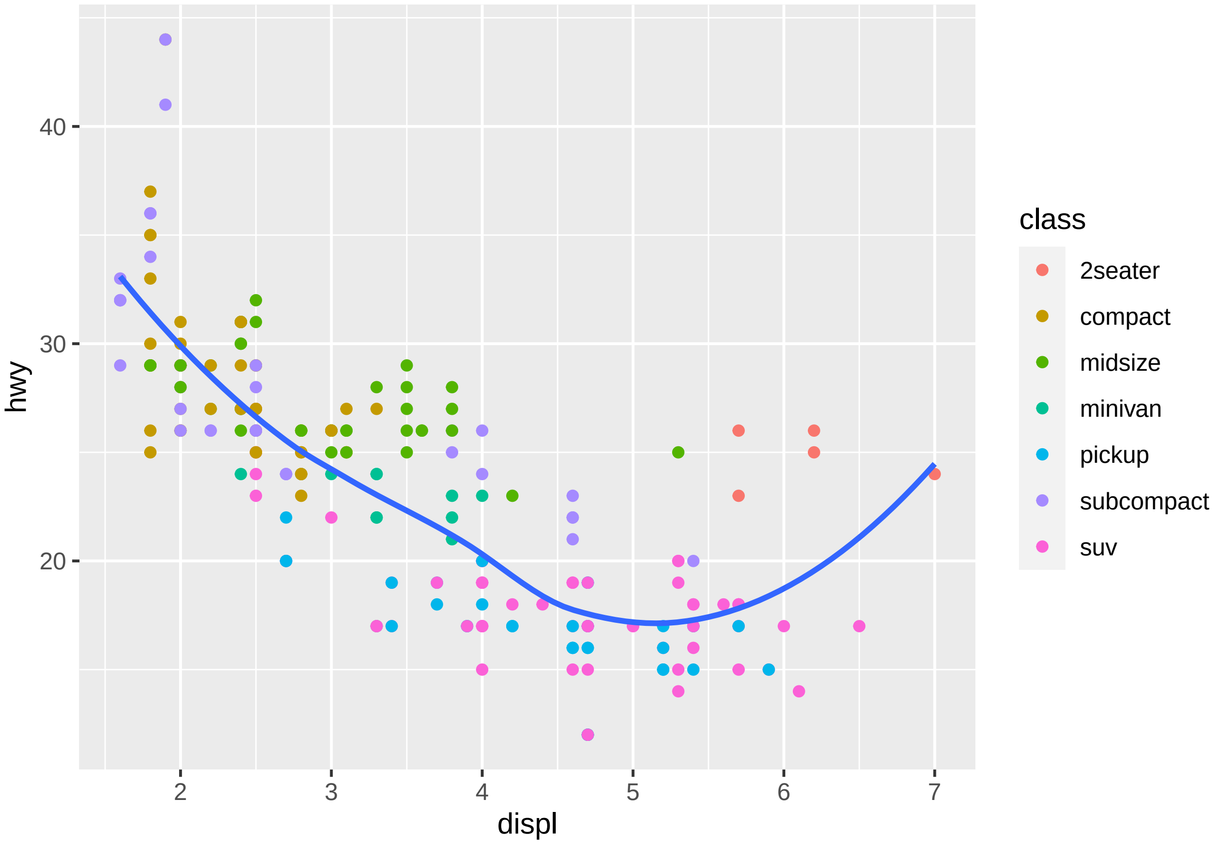
#This command shows about the mpg dataset.

# a. Which variables from mpg data set are categorical? #The variable that is categorical are the following: #manufacturer, model, year of manufacturer, trans, dtr, cyl, drv, cty, highwat miles per gallon, and fluel type # b. Which are continuous variables? #The continuous variable is displ(engine displacement, in liters). #c. Plot the relationship between displ (engine displacement) and hwy(highway miles #per gallon). Mapped it with a continuous variable you have identified in #5-b. What is its result? Why it produced such output? ggplot(mpg, aes(x = displ, y = hwy, colour = cty)) + geom_point()

#9. Plot the relationship between displ (engine displacement) and hwy(highway miles #per gallon) using geom_point(). Add a trend line over the existing plot using #geom_smooth() with se = FALSE. Default method is "loess".

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

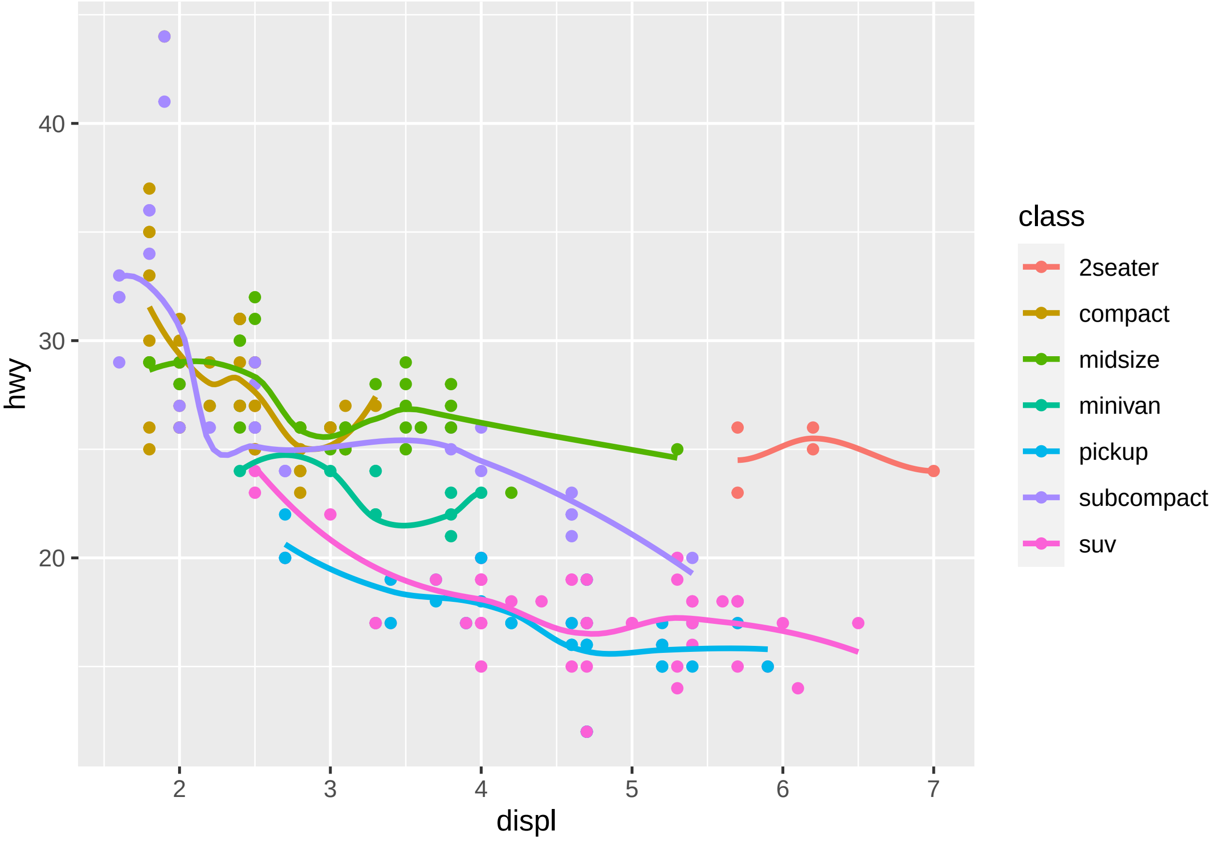
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



#10. Using the relationship of displ and hwy, add a trend line over existing plot. Set the #se = FALSE to remove the confidence interval and method = lm to check for linear modeling.

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

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

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 5.6935

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.5065

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 0.65044

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 4.008

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.708

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 0.25
```


hwy
displ
class
2seater
compact
midsize
minivan
pickup
subcompact
suv