Worksheet _Tabladillo#6

Quennie Tabladillo

2022-11-25

R Markdown

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
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)
mpg
```

```
## # A tibble: 234 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~
## # ... with 224 more rows
```

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

```r
#Number of column in mpg dataset
ncol(mpg)
```

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

```r
#Number of rows in mpg dataset
nrow(mpg)
```

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

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

```r
#Manufacturer with the most models
data1 <- mpg %>% group_by(manufacturer) %>% count()
data1
```

```
## # A tibble: 15 x 2
## # Groups:   manufacturer [15]
##    manufacturer     n
##    <chr>        <int>
##  1 audi            18
##  2 chevrolet       19
##  3 dodge           37
##  4 ford            25
##  5 honda            9
##  6 hyundai         14
##  7 jeep             8
##  8 land rover       4
##  9 lincoln          3
## 10 mercury          4
## 11 nissan          13
## 12 pontiac          5
## 13 subaru          14
## 14 toyota          34
## 15 volkswagen      27
```

```r
#Model with the most variations
data2 <- mpg %>% group_by(model) %>% count()
data2
```

```
## # A tibble: 38 x 2
## # Groups:   model [38]
##    model                 n
##    <chr>             <int>
##  1 4runner 4wd           6
##  2 a4                    7
##  3 a4 quattro            8
##  4 a6 quattro            3
##  5 altima                6
##  6 c1500 suburban 2wd    5
```

```
##  7 camry                   7
##  8 camry solara            7
##  9 caravan 2wd            11
## 10 civic                   9
## # ... with 28 more rows
```

```
#Ans: The manufacturer with the most models is "dodge" and the model with the most
#variations is the "caravan 2wd"
```

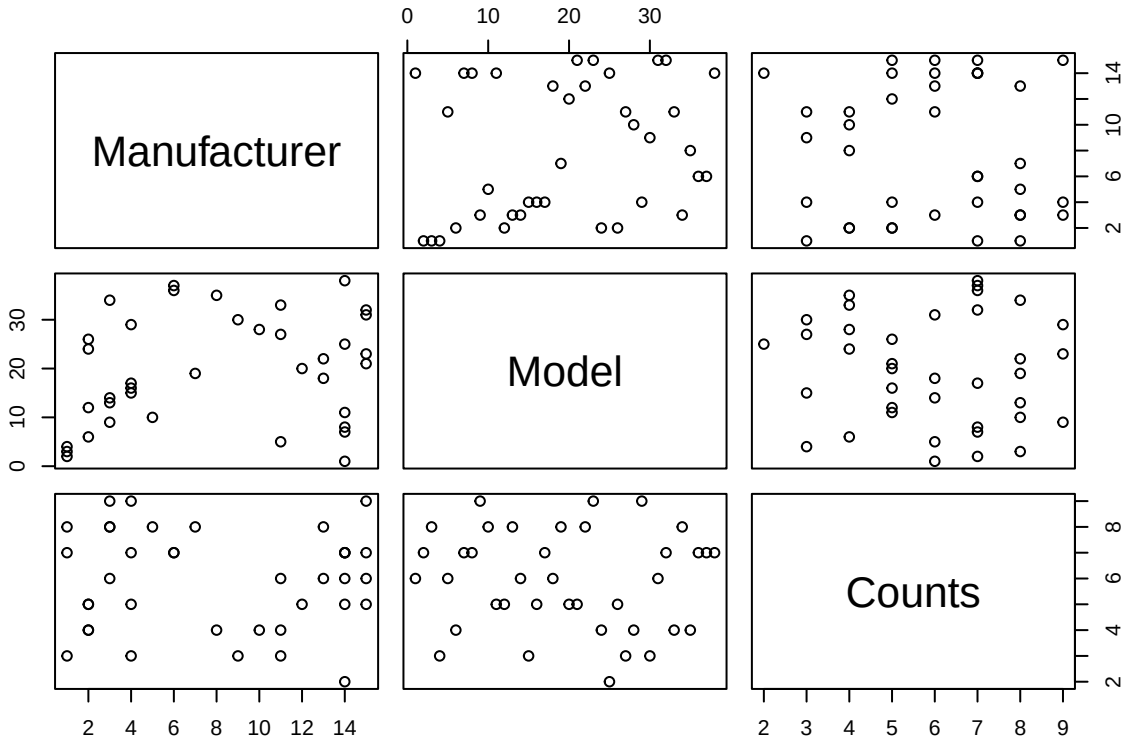
a.Group the manufacturers and find the unique models. Copy the codes and result.

```
unique1 <- mpg %>% group_by(manufacturer, model) %>% distinct %>% count()
colnames(unique1) <- c("Manufacturer","Model","Counts")
unique1
```

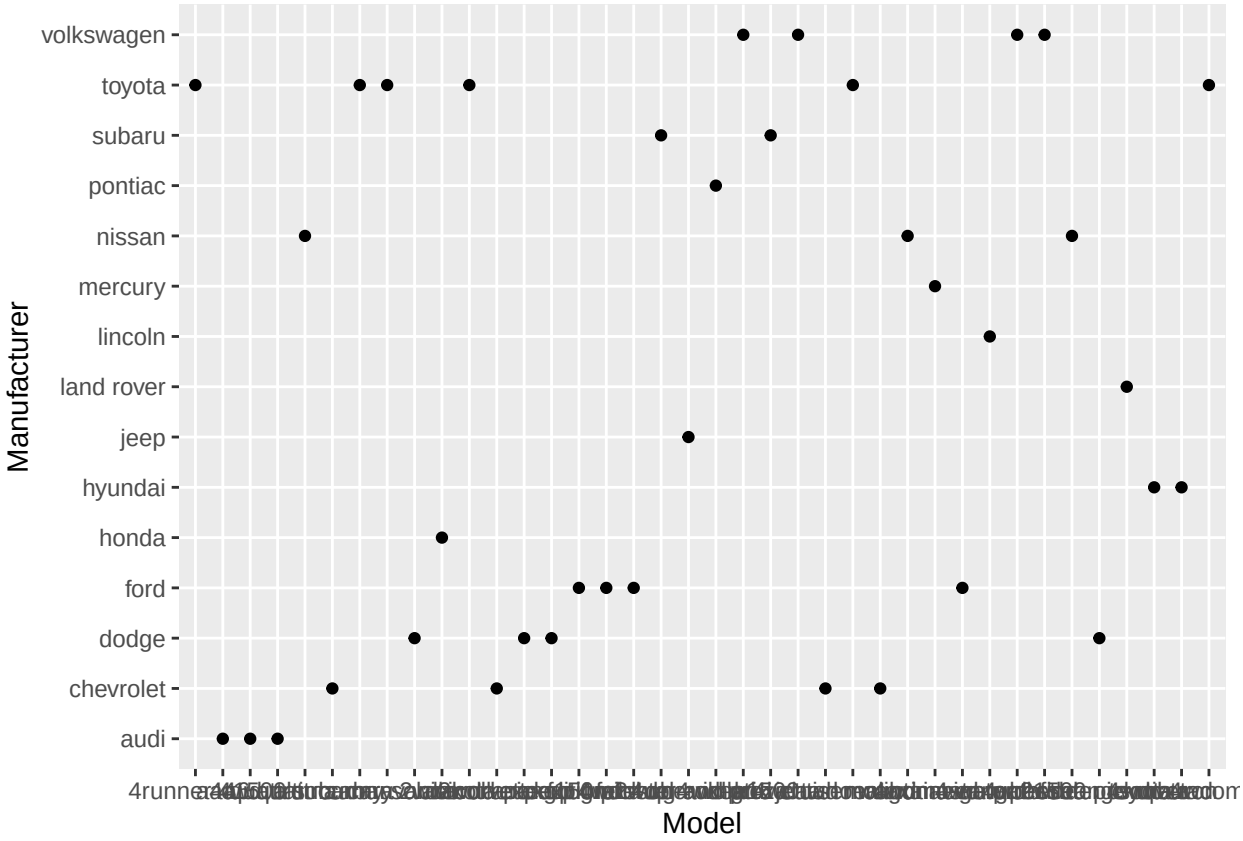
```
## # A tibble: 38 x 3
## # Groups:   Manufacturer, Model [38]
##    Manufacturer Model              Counts
##    <chr>        <chr>               <int>
##  1 audi         a4                      7
##  2 audi         a4 quattro              8
##  3 audi         a6 quattro              3
##  4 chevrolet    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

```
plot(unique1)
```



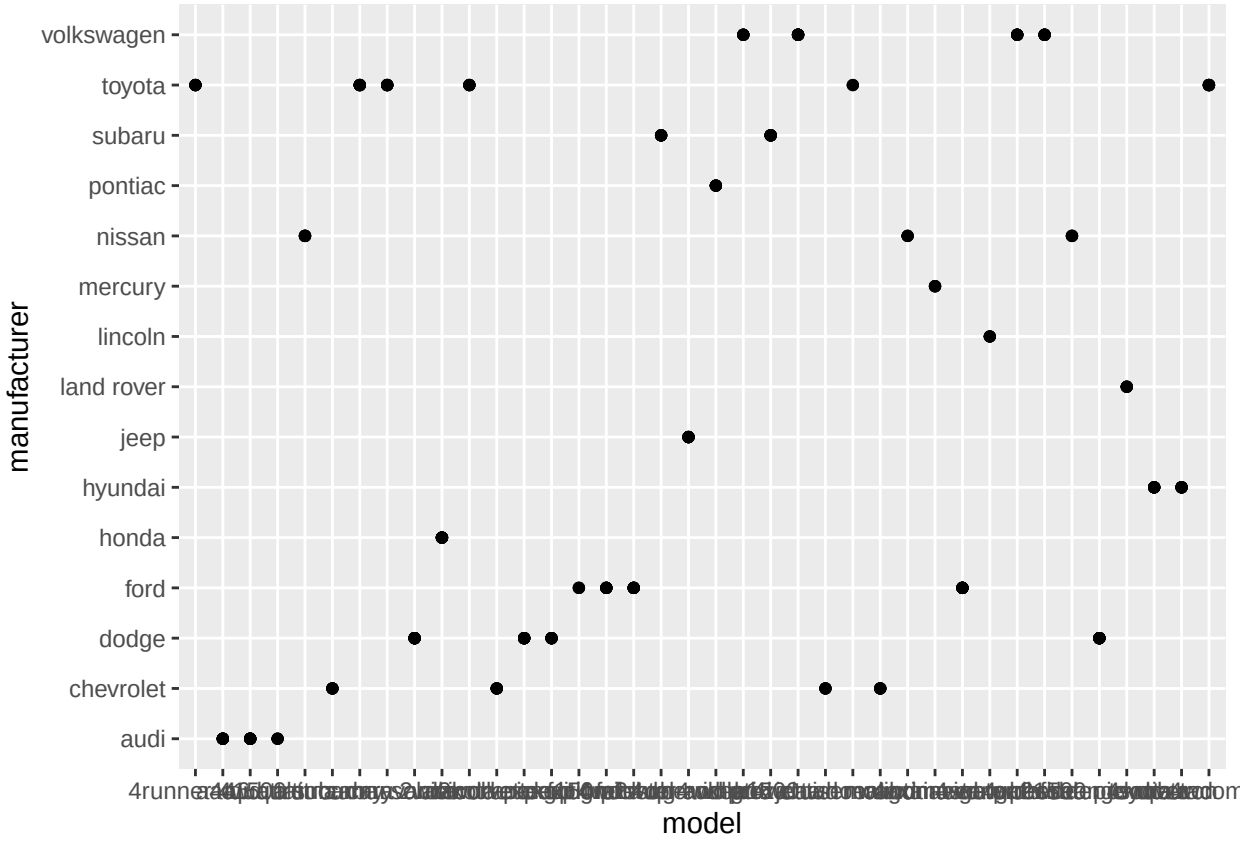

```
ggplot(unique1, aes(Model, Manufacturer)) + geom_point()
```



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

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

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



b.For you, is it useful? If not, how could you modify the data to make it more informative?

```
#Ans: The plot is not useful for a number of reasons
#Each dot represents a different manufacturer-model combination that we observe in the data.
#Some of the data is hidden as some manufacturer-model combinations appear
#more than once in the data (e.g. `audi a4 quattro`)
```

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

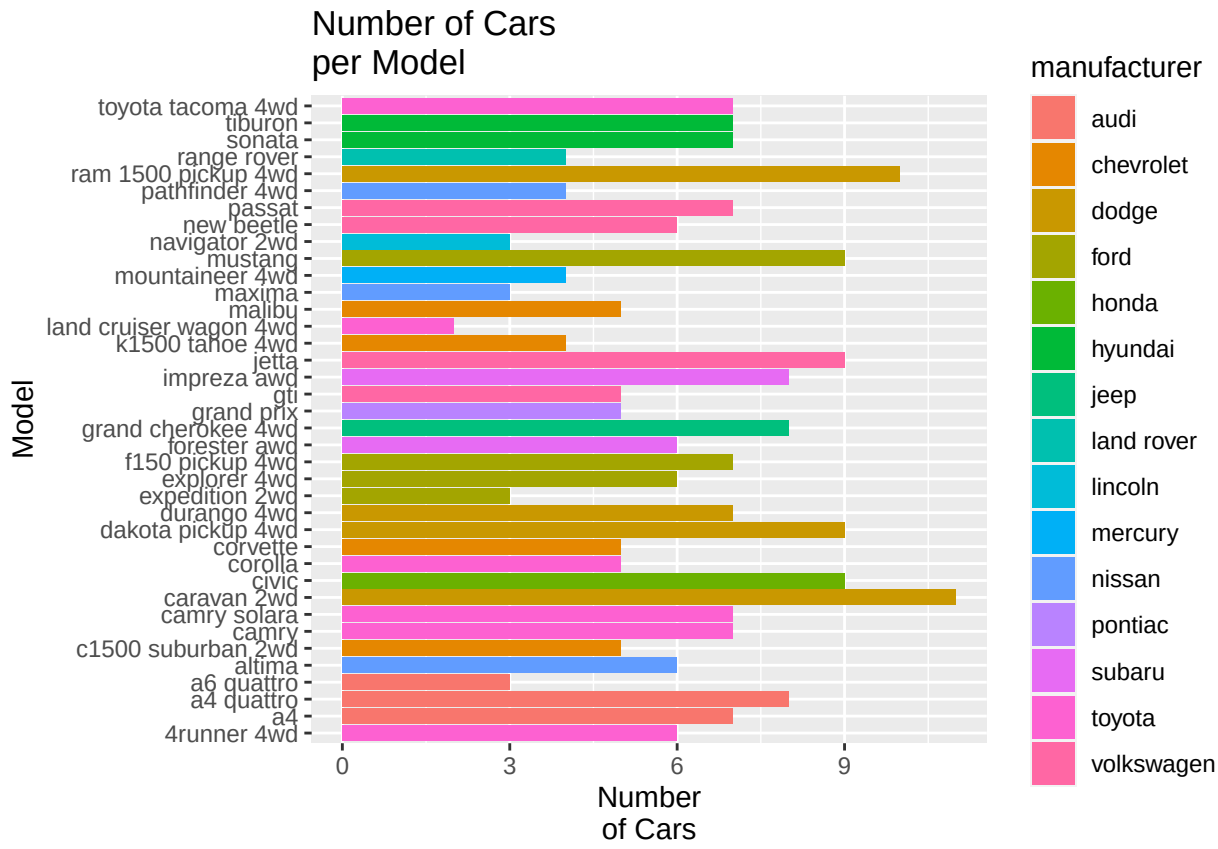
```
## # A tibble: 38 x 2
## # Groups:   model [38]
##    model                  n
##    <chr>              <int>
##  1 4runner 4wd            6
##  2 a4                     7
##  3 a4 quattro             8
##  4 a6 quattro             3
##  5 altima                 6
##  6 c1500 suburban 2wd     5
##  7 camry                  7
##  8 camry solara           7
##  9 caravan 2wd           11
## 10 civic                  9
## # ... with 28 more rows
```

```
## # A tibble: 38 x 2
## # Groups:   Model [38]
##    Model               Counts
##    <chr>                <int>
##  1 4runner 4wd              6
##  2 a4                       7
##  3 a4 quattro               8
##  4 a6 quattro               3
##  5 altima                   6
##  6 c1500 suburban 2wd       5
##  7 camry                    7
##  8 camry solara             7
##  9 caravan 2wd             11
## 10 civic                    9
## # ... with 28 more rows
```

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

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

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

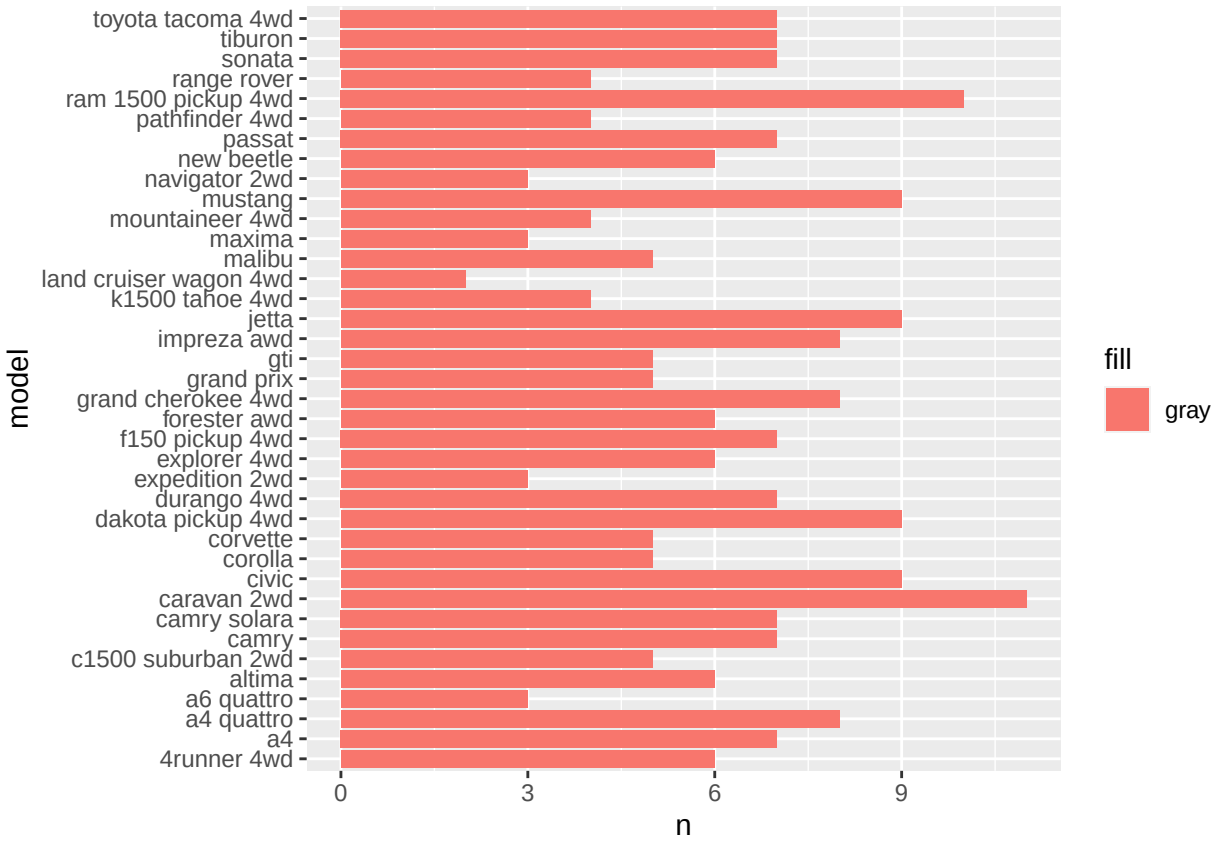


b.Use only the top 20 observations. Show code and results

```
topObserve <- mpg %>% group_by(model) %>% tally(sort = TRUE)
topObserve
```

```
## # A tibble: 38 x 2
##    model                  n
##    <chr>              <int>
##  1 caravan 2wd           11
##  2 ram 1500 pickup 4wd   10
##  3 civic                  9
##  4 dakota pickup 4wd      9
##  5 jetta                  9
##  6 mustang                9
##  7 a4 quattro             8
##  8 grand cherokee 4wd     8
##  9 impreza awd            8
## 10 a4                     7
## # ... with 28 more rows
```

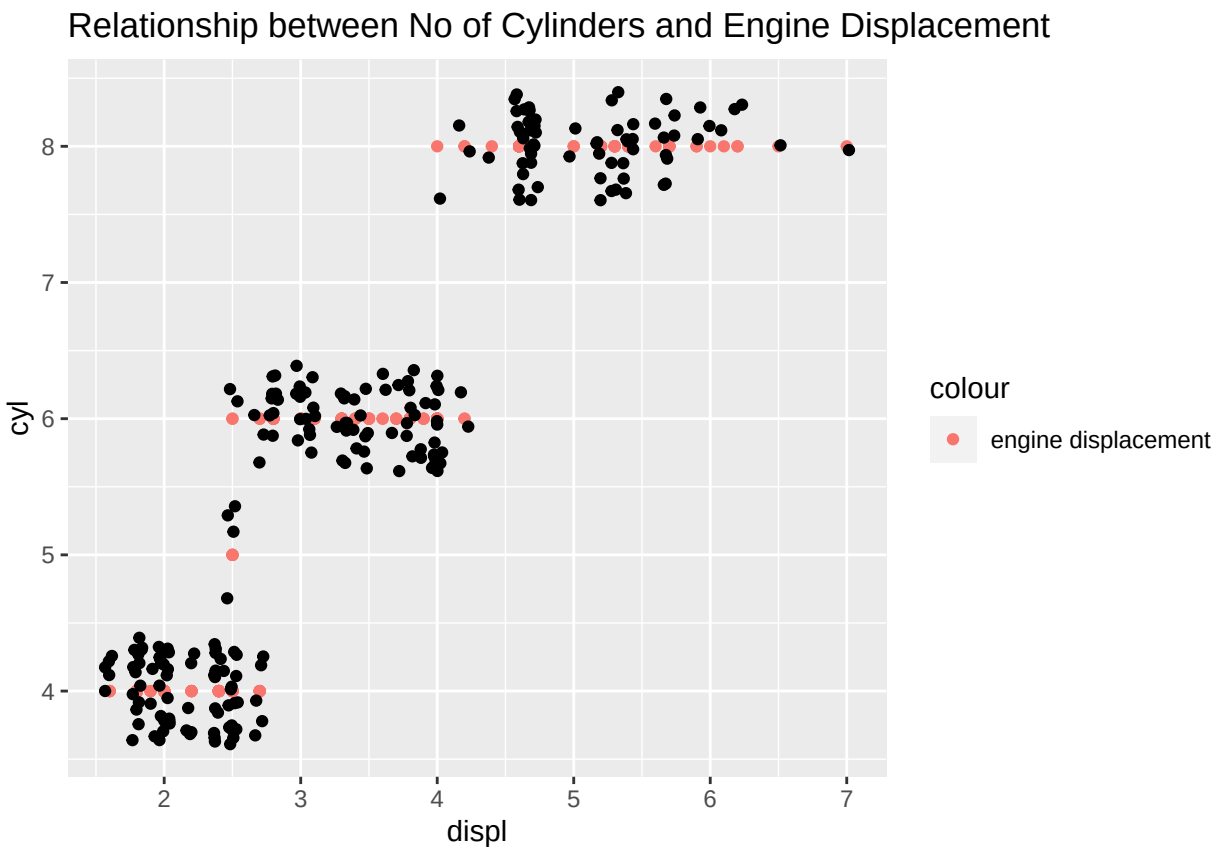
```
ggplot(topObserve, aes(x = model, y = n, fill = "gray")) +
  geom_bar(stat = "identity") + coord_flip()
```



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



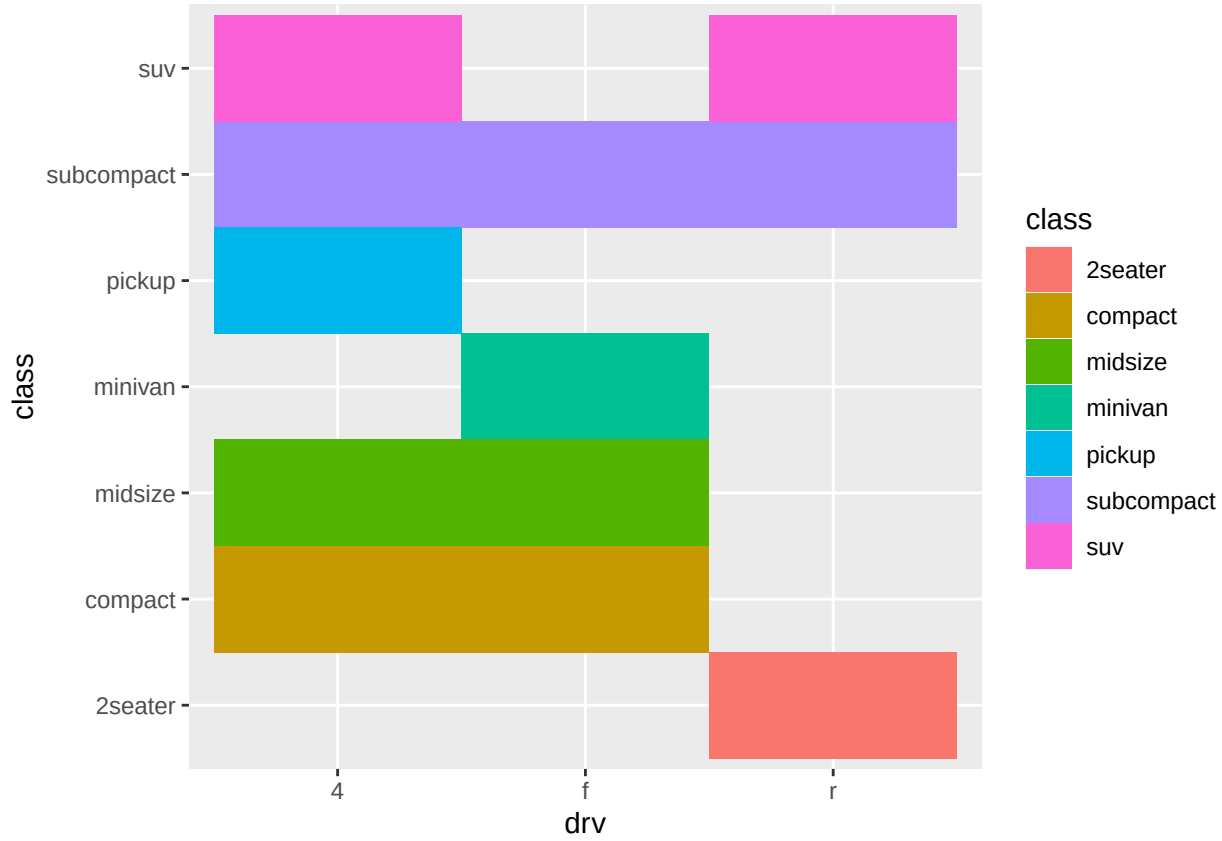
b. How would you describe its relationship?

```
#As the number of  cylinder increases, the engine displacement also increases.
```

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, 2seater, 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(mpg, aes(x = drv, y = class)) + geom_tile(aes(fill = class))
```

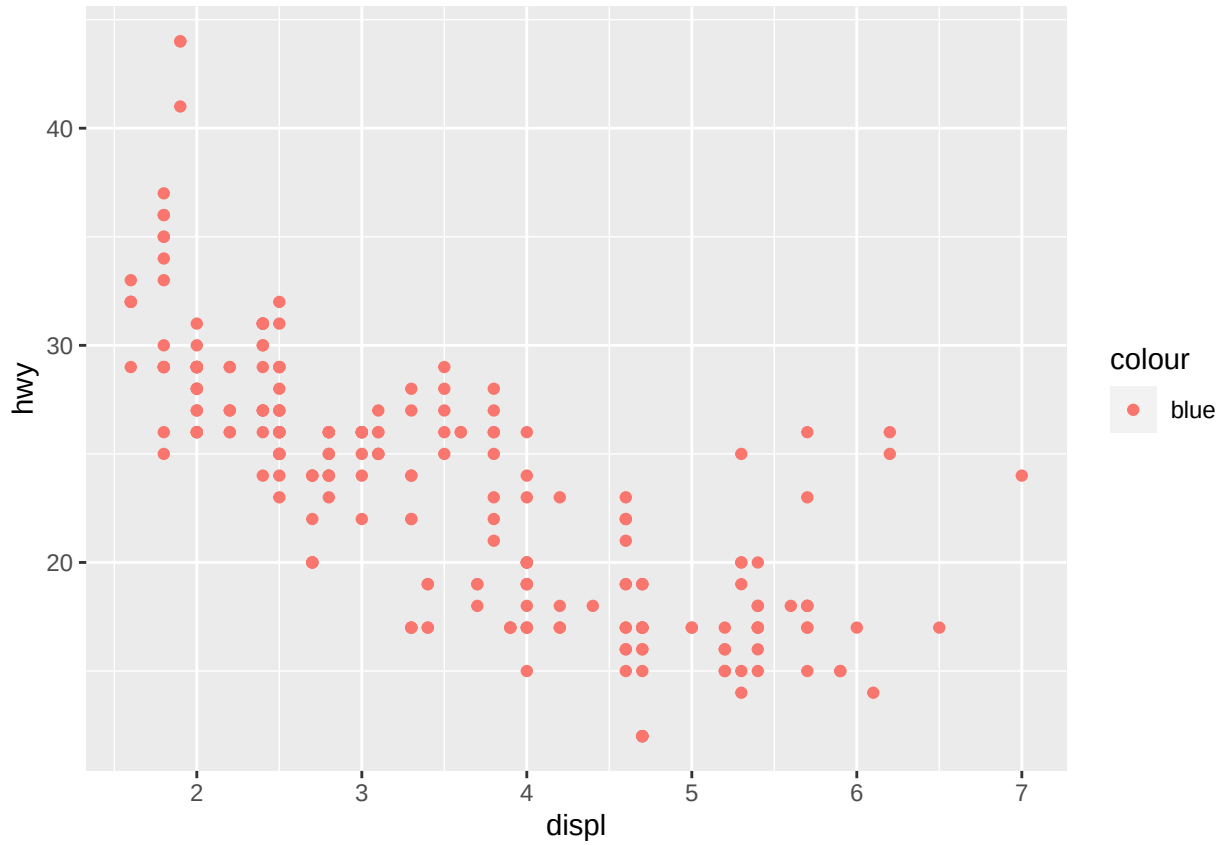


b.Interpret the result.

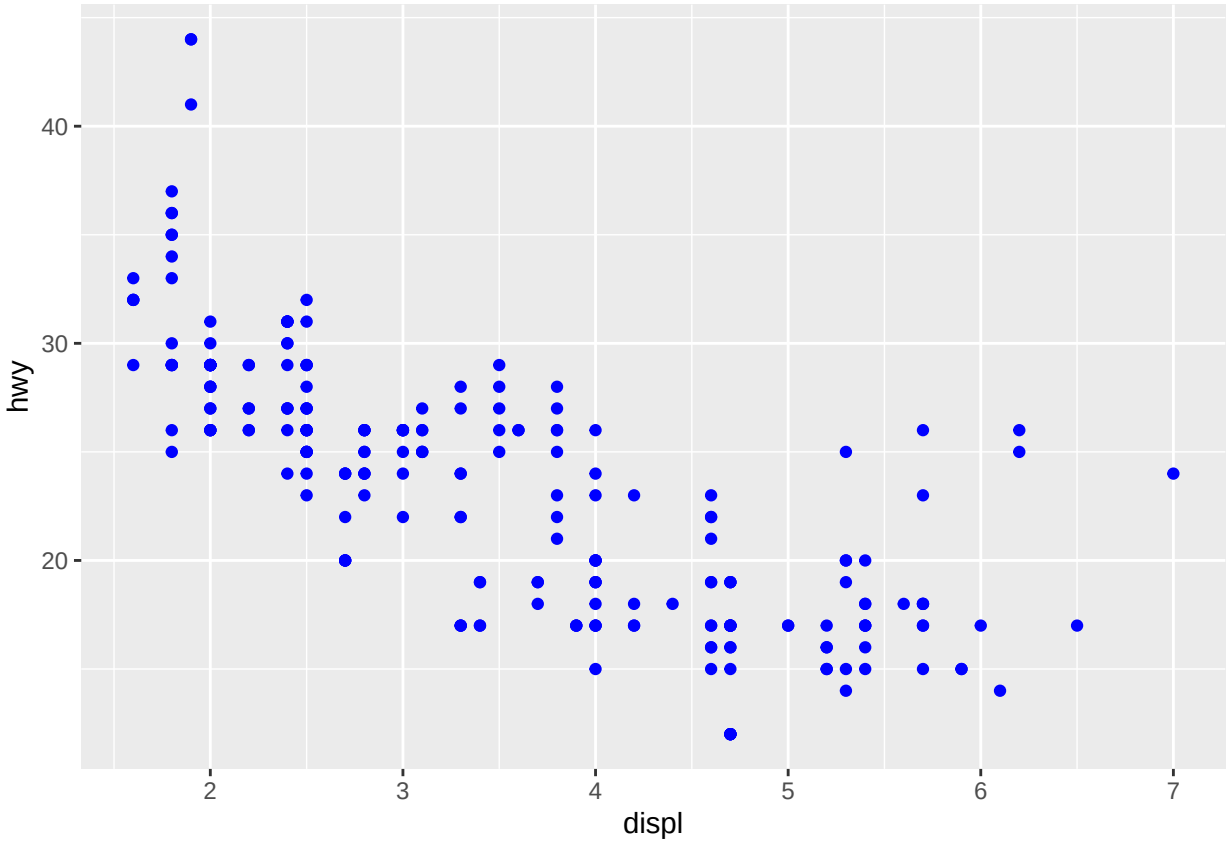
```
#Ans: The suv are both under the 4 wheel drive and the rear wheel drive, while
#the subcompact,pick up, midsize and the compact fall under the 4 wheel drive only.
#The minivan fall under the front wheel drive and the 2seater fall under the rear wheel drive.
```

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

```
#Code 1
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy, colour = "blue"))
```




```
#Code 2
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy), colour = "blue")
```




```
#Ans: The difference between the two codes are the color of the dots in the
#plot although, based on the two codes the color that was declared was blue.
```

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

```
#Ans: The result was the R documentation of a dataset about the fuel economy
#data from 1999 to 2008 for 38 popular models of car
```

a.Which variables from mpg dataset are categorical?

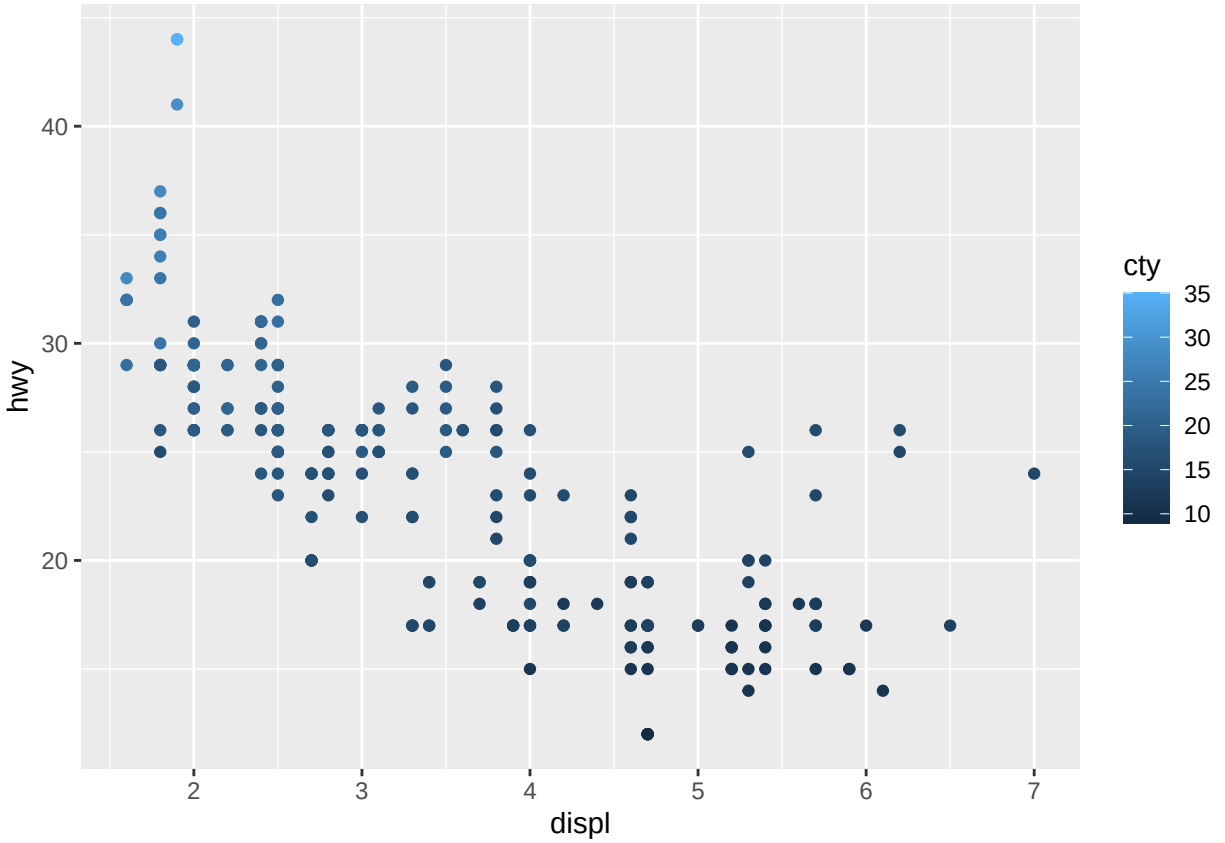
```
#Ans: Categorical variables in mpg include: manufacturer, model, trans (type of transmission), drv (fro
```

b. Which are continuous variables?

```
#Ans: cty , city highway miles per gallon
```

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()
```

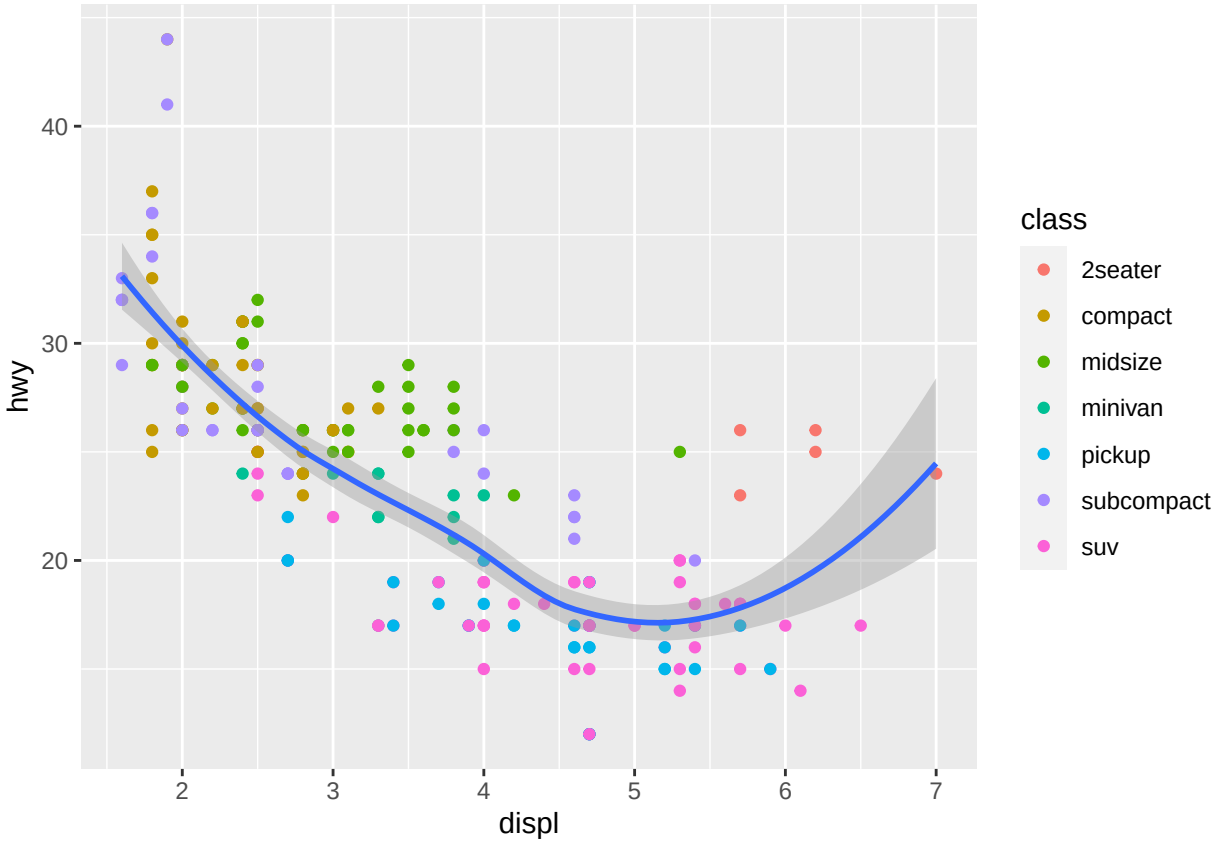



```
#Ans: The plot shows a negative relationship between engine size (displ) and
#fuel efficiency (hwy). In other words, cars with big engines use more fuel.
```

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(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color=class)) +
  geom_smooth()
```

```
## `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 these = FALSE to remove the confidence interval and method = lm to check for linear modeling

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