RWorksheet_Gonzales#6

Mamerto F. Gonzales Jr.

2022-11-28

#WorkSheet-6

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
```

```
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
```

```
ncol(mpg)
```

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

```
nrow(mpg)
```

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

- There are 11 columns and 234 rows in the mpg data frame.

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

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

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

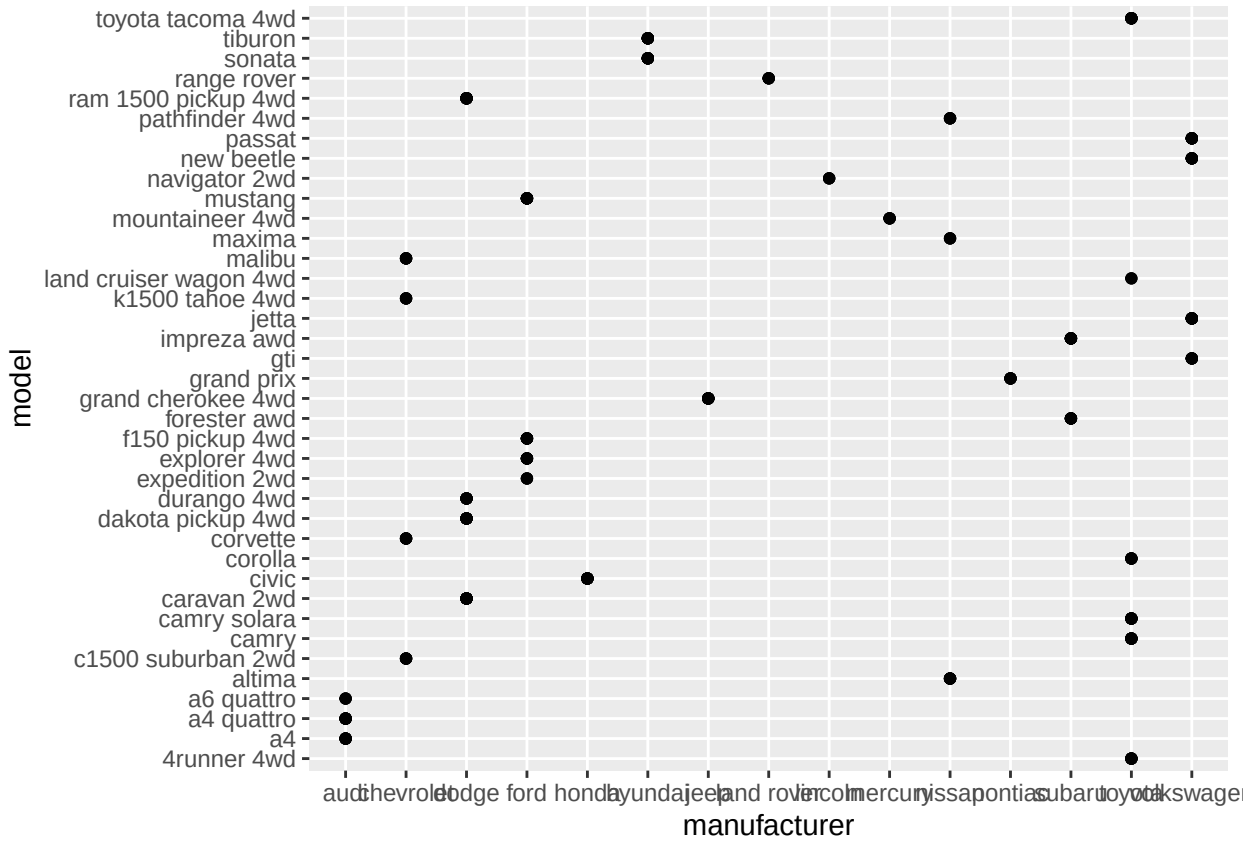
```
## # A tibble: 15 x 2
##    manufacturer     n
##    <chr>        <int>
##  1 dodge           37
##  2 toyota          34
##  3 volkswagen      27
##  4 ford            25
##  5 chevrolet       19
##  6 audi            18
##  7 hyundai         14
##  8 subaru          14
##  9 nissan          13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

```
unique(mpg$model)
```

```
##  [1] "a4"                    "a4 quattro"            "a6 quattro"
##  [4] "c1500 suburban 2wd"    "corvette"              "k1500 tahoe 4wd"
##  [7] "malibu"                "caravan 2wd"           "dakota pickup 4wd"
## [10] "durango 4wd"           "ram 1500 pickup 4wd"   "expedition 2wd"
## [13] "explorer 4wd"          "f150 pickup 4wd"       "mustang"
## [16] "civic"                 "sonata"                "tiburon"
## [19] "grand cherokee 4wd"    "range rover"           "navigator 2wd"
## [22] "mountaineer 4wd"       "altima"                "maxima"
## [25] "pathfinder 4wd"        "grand prix"            "forester awd"
## [28] "impreza awd"           "4runner 4wd"           "camry"
## [31] "camry solara"          "corolla"               "land cruiser wagon 4wd"
## [34] "toyota tacoma 4wd"     "gti"                   "jetta"
## [37] "new beetle"            "passat"
```

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

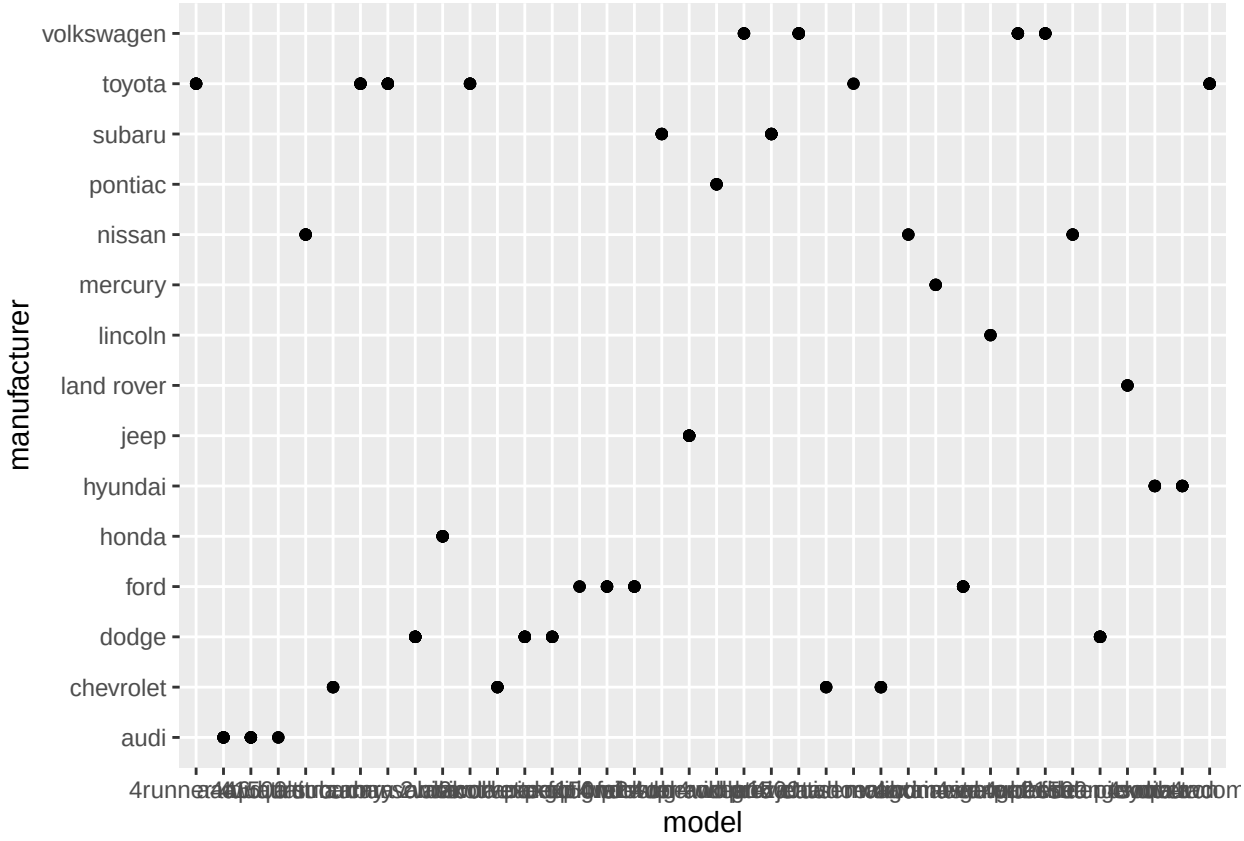
```
ggplot(mpg, aes(manufacturer, model)) +
geom_point()
```



3. Same dataset will be used. You are going to show the relationship of the modeland 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?

- Yes it is very useful because it is very easy to get information from.

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

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

```
## # 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
```

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

```
ggplot(CarsModel, aes(x = model, y = n, colour = "rainbow")) +
geom_bar(stat = "identity") + coord_flip()
```

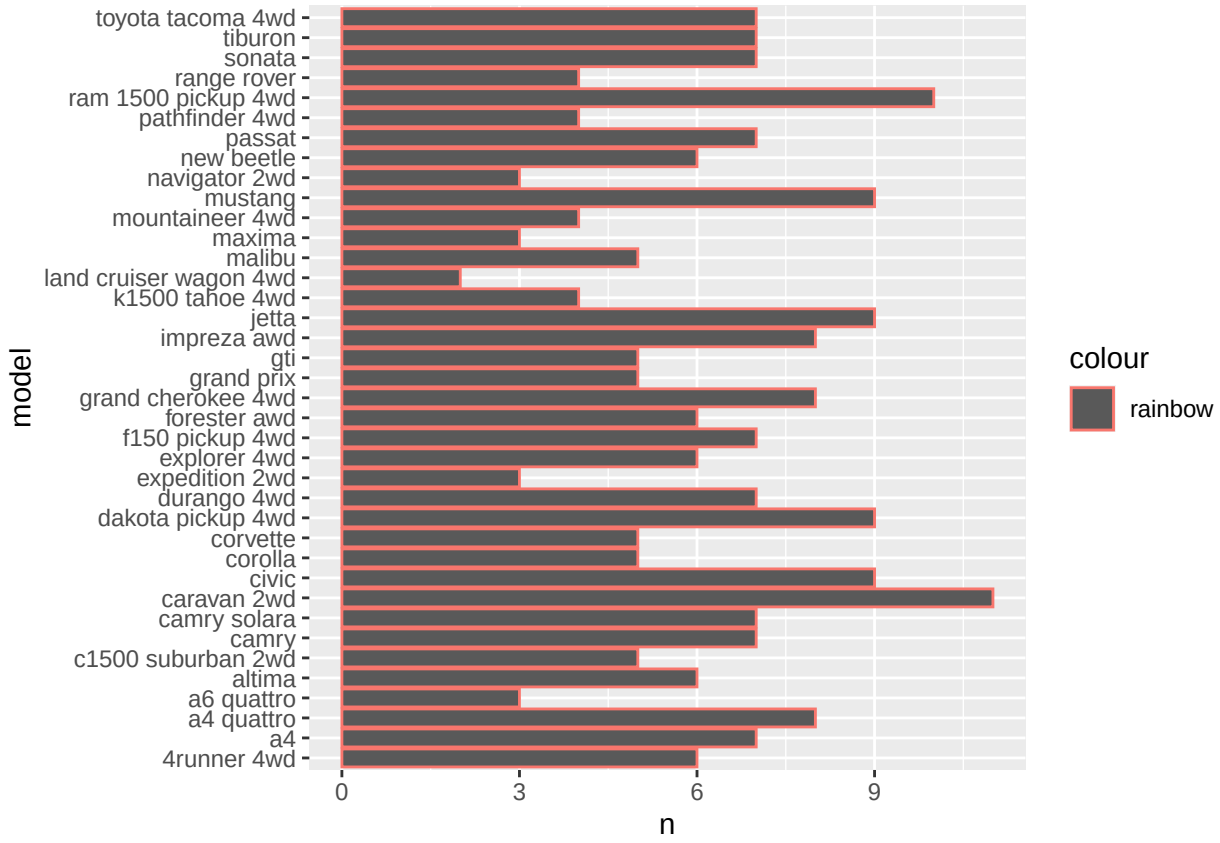
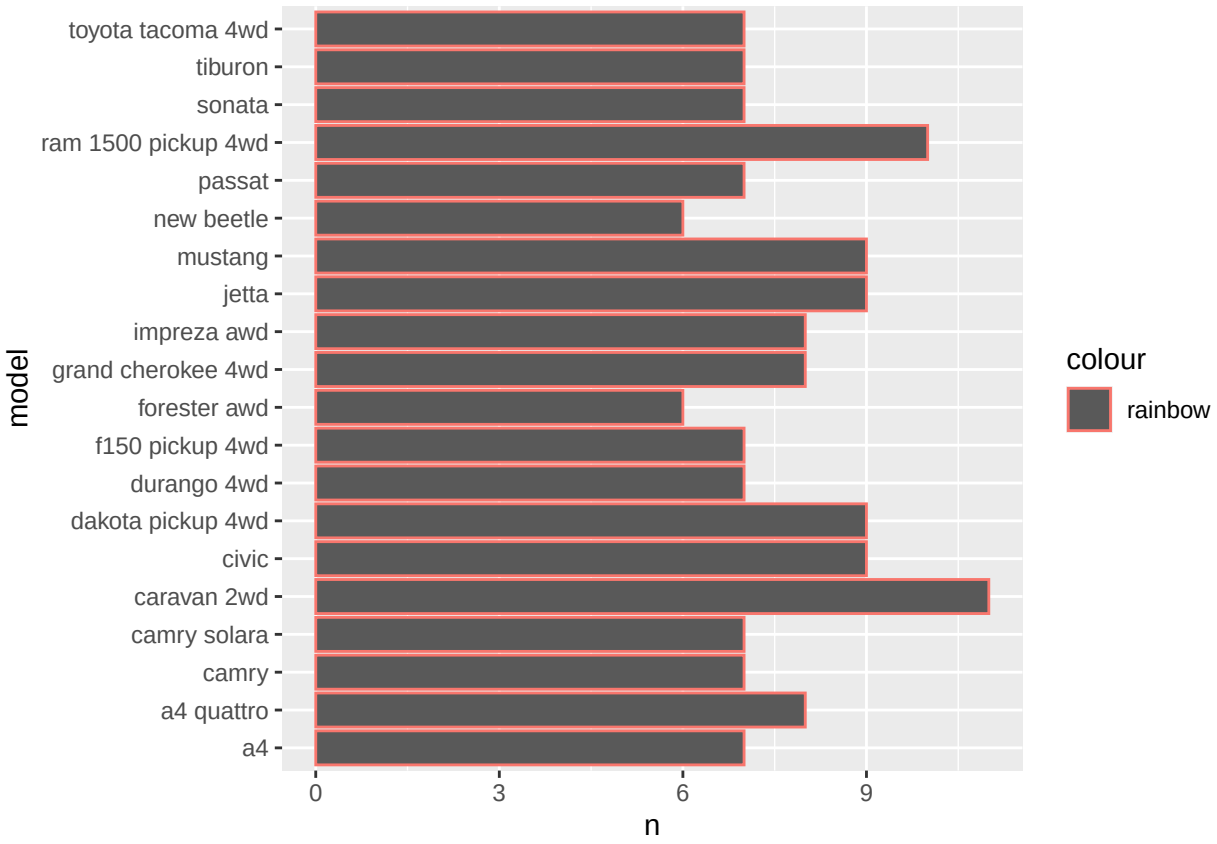


Figure 1: Car Models b. Use only the top 20 observations. Show code and results.

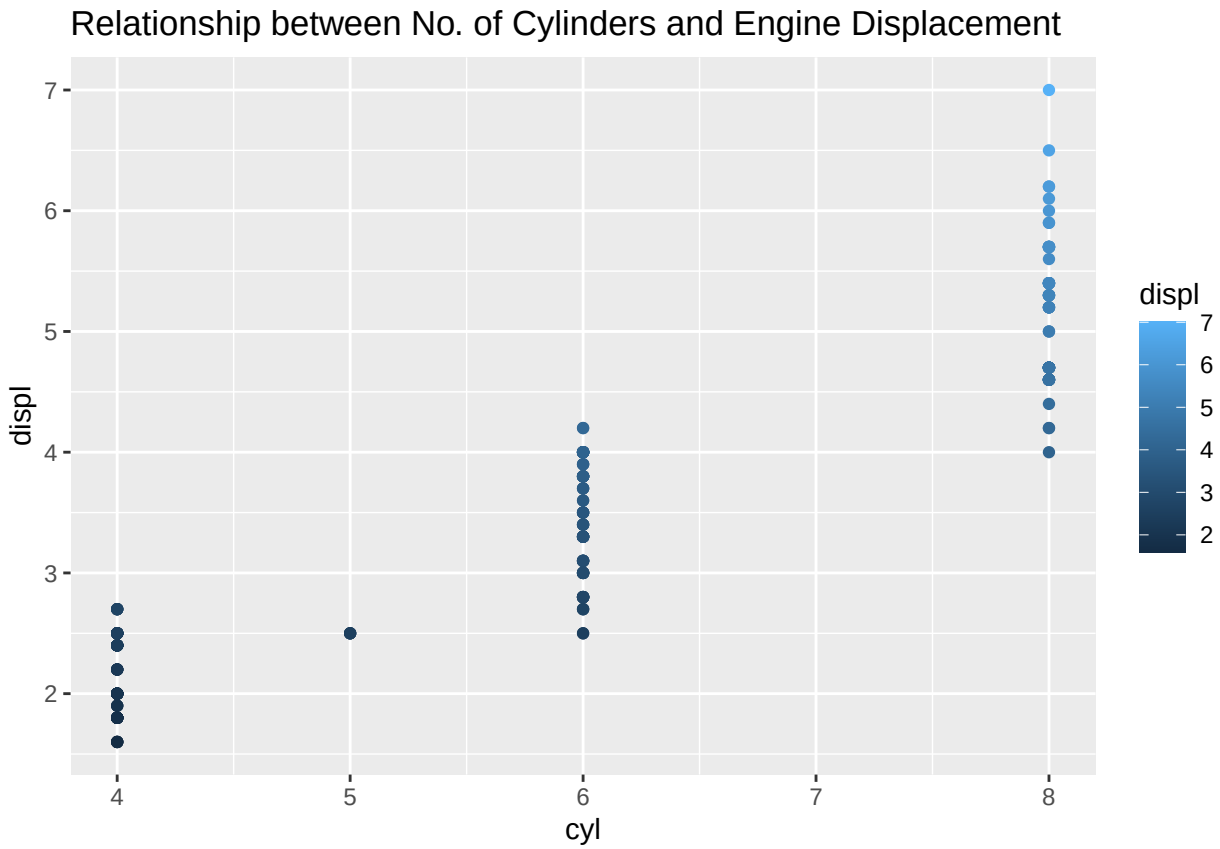
```
ggplot(CarsModel[tail(order(CarsModel$n), 20), ], ) +
     aes(model, n, colour = "rainbow") + geom_bar(stat = "identity") + coord_flip()
```



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.

```
CylVsDispl <- ggplot(mpg, aes(x = cyl, y = displ, color = displ)) +
geom_point()
print(CylVsDispl + ggtitle("Relationship between No. of Cylinders and Engine Displacement"))
```



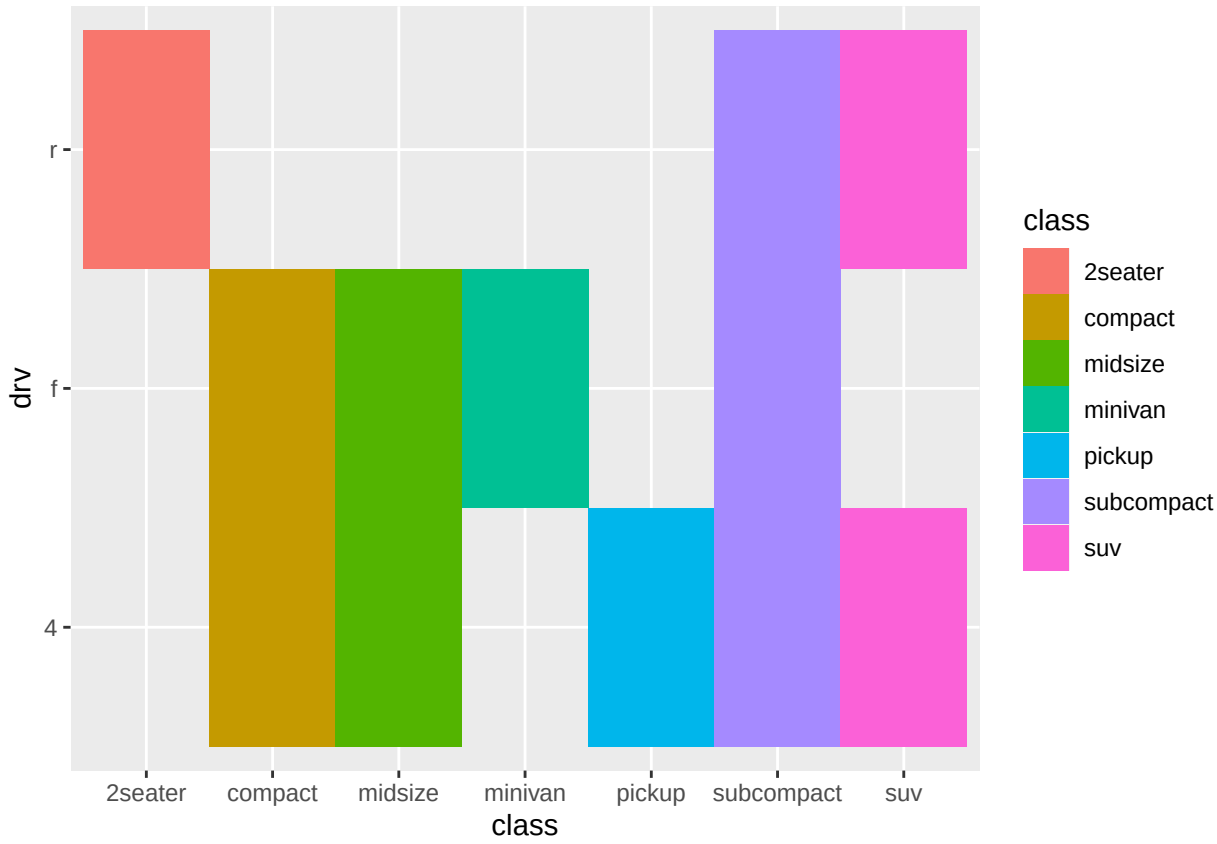
b. How would you describe its relationship?

- The higher the number of Cylinders, the engine displacement also rises.

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.

```
mpg %>%
count(class, drv) %>%
ggplot(aes(x = class, y = drv)) +
geom_tile(mapping = aes(fill = class))
```

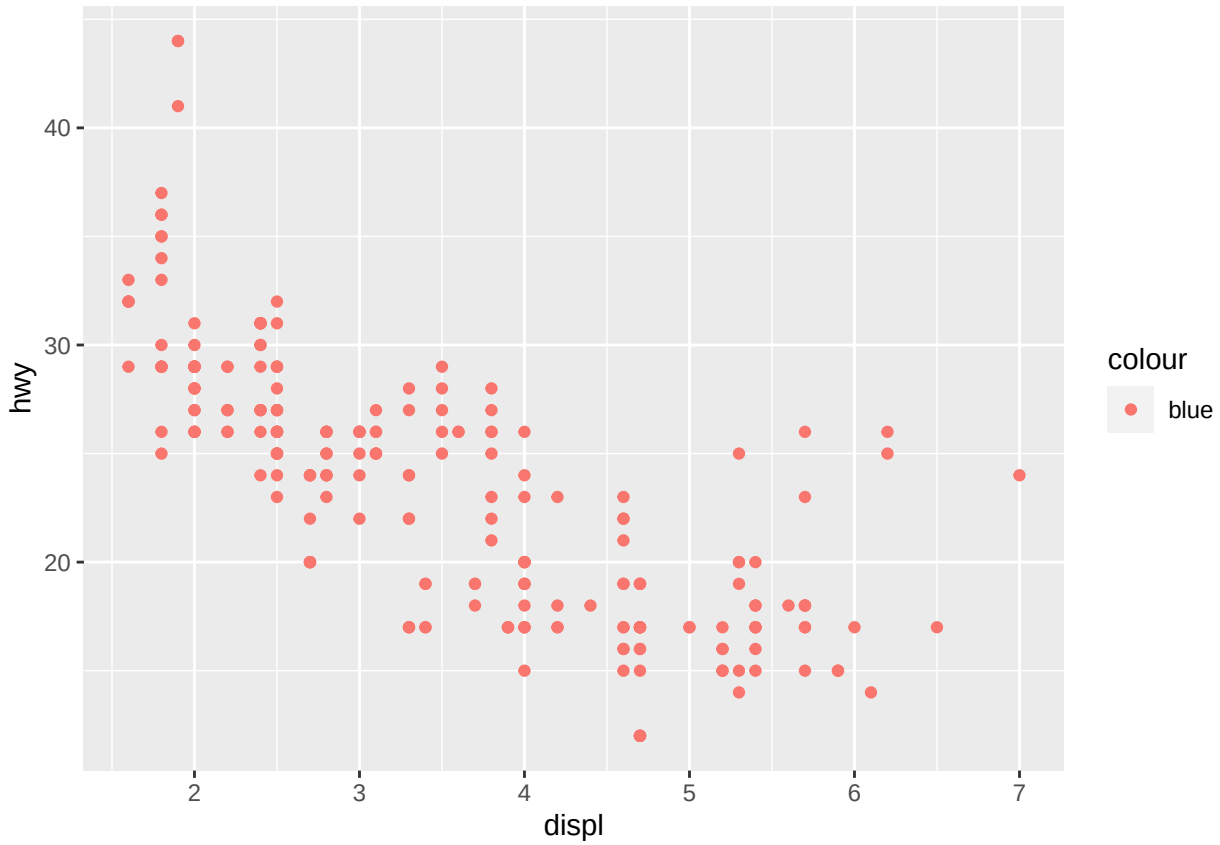


b. Interpret the result.

- Different types of cars have different types of Driving.

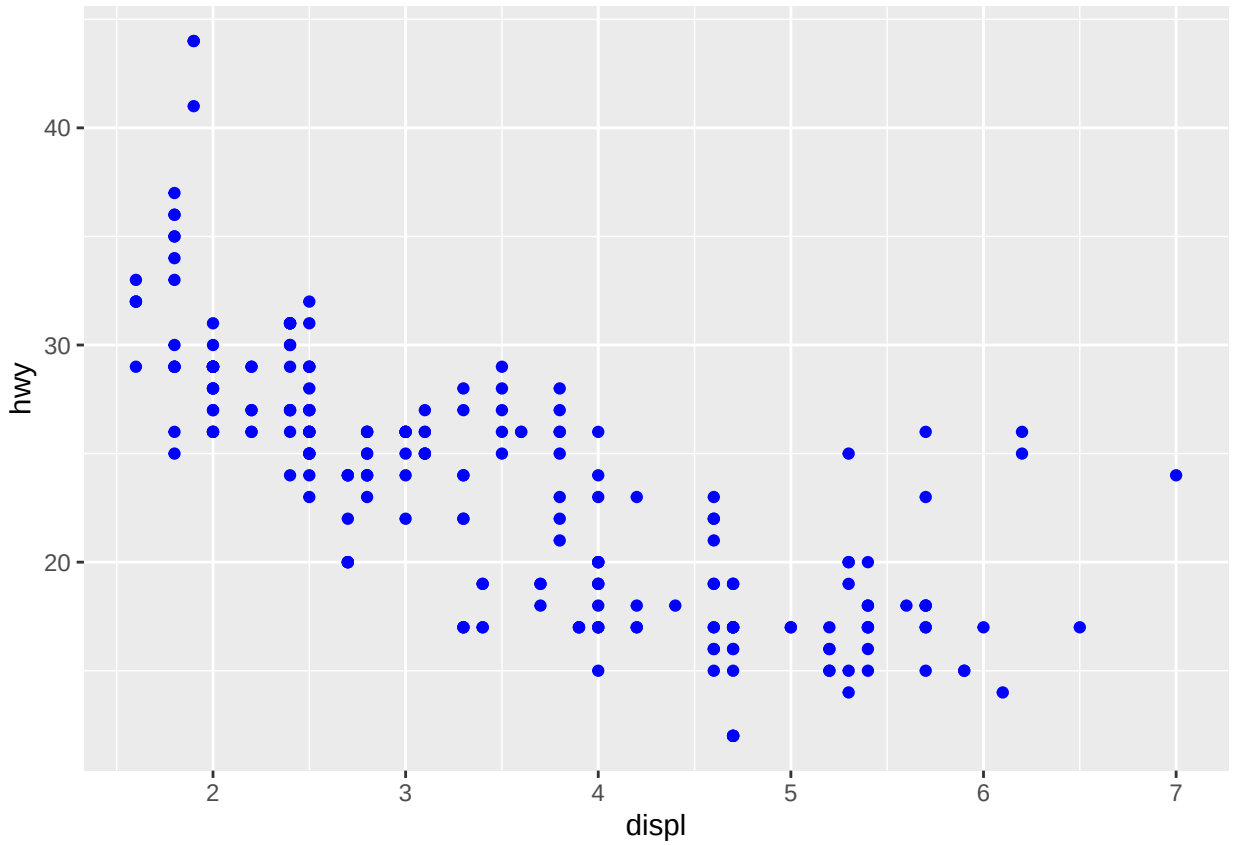
7. Discuss the difference between these codes. Its outputs for each are shown below. • Code #1

```
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")
```



- Putting the values inside the aes generates a legend and makes the color with that while putting it outside aes ggplot2 did not make the legend automatically and inputted your value.

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

```
?mpg
```

- It scours the internet and shows its description and usage

a. Which variables from mpg dataset are categorical?

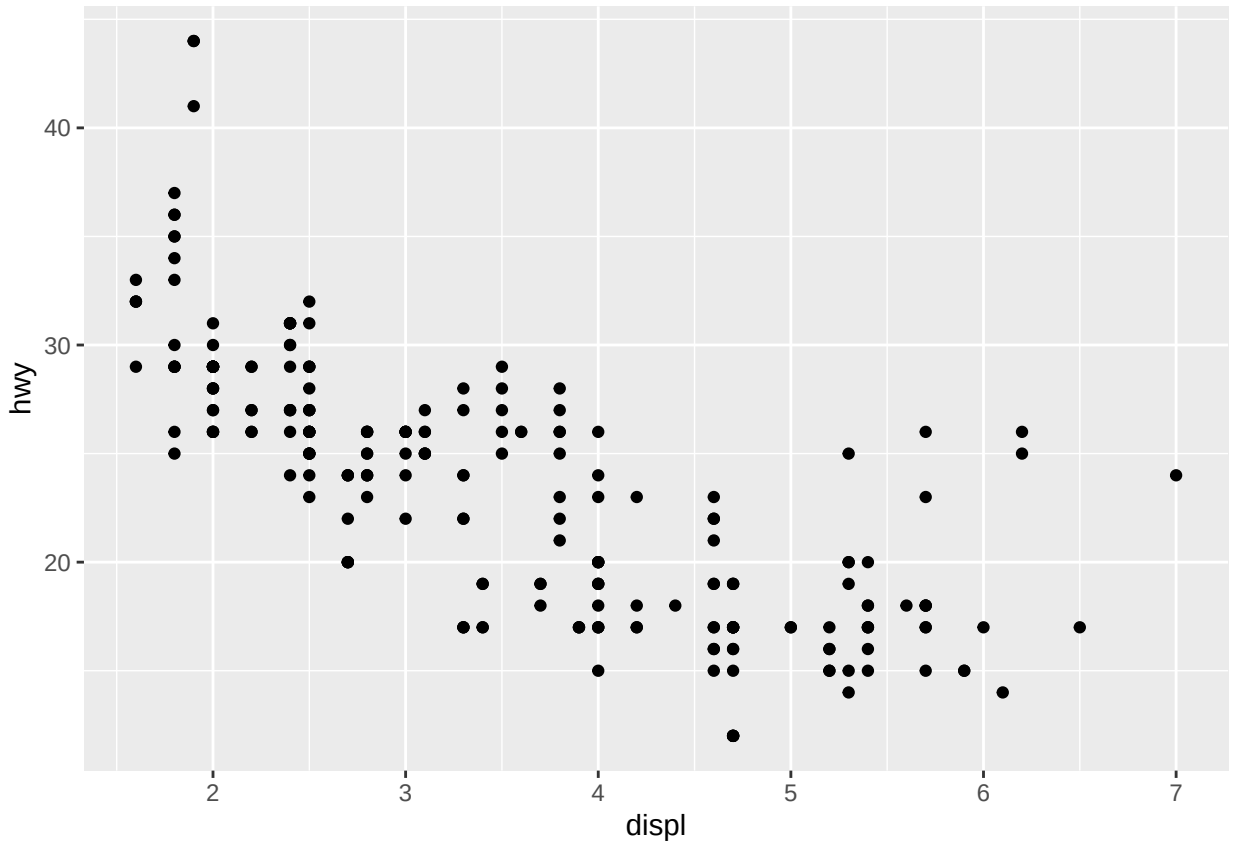
- The variables that are categorical in mpg dataset are manufacturer, model, trans, drv, fl, and class.

b. Which are continuous variables?

- The continuous variables in mpg dataset are displ, year, cyl, cty, and hwy

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?

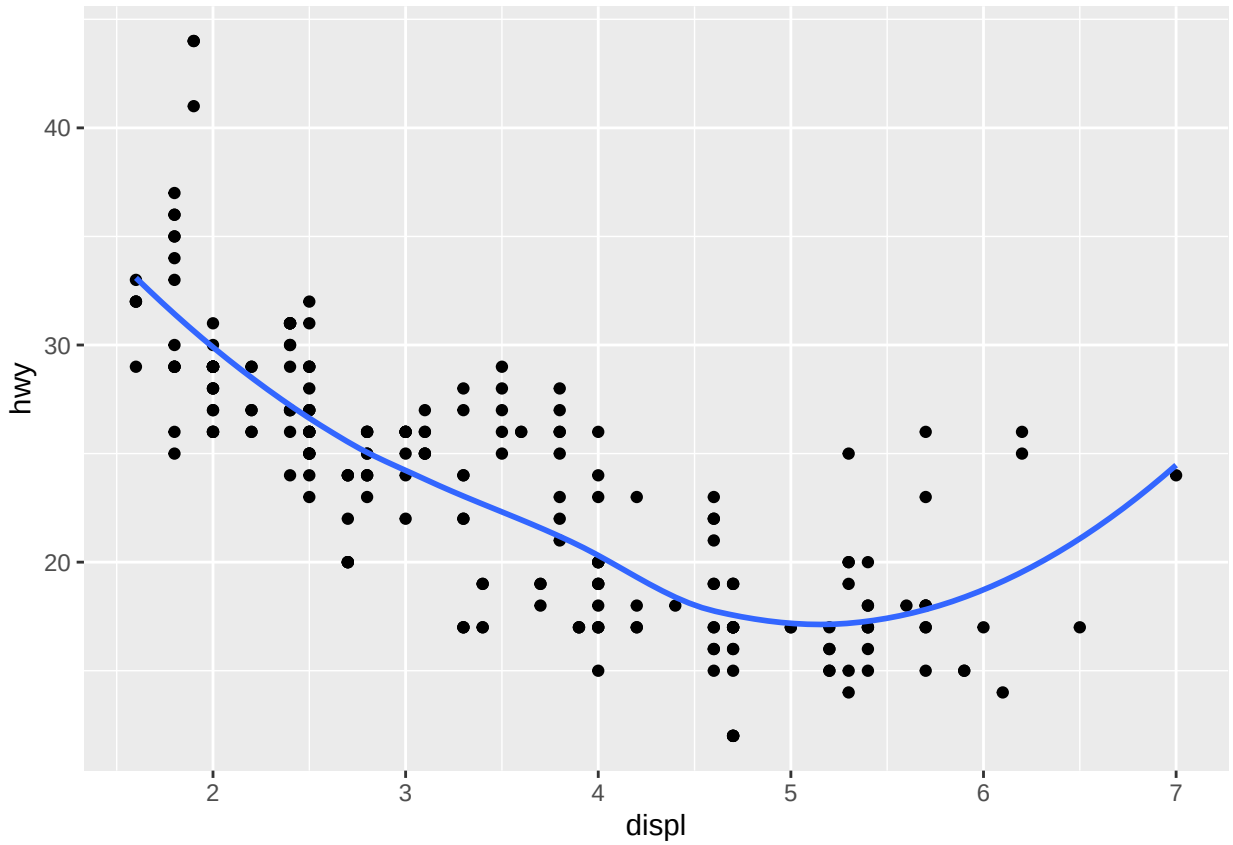
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
geom_point()
```



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(mpg, aes(displ, hwy)) +
geom_point() + geom_smooth(se = FALSE)
```

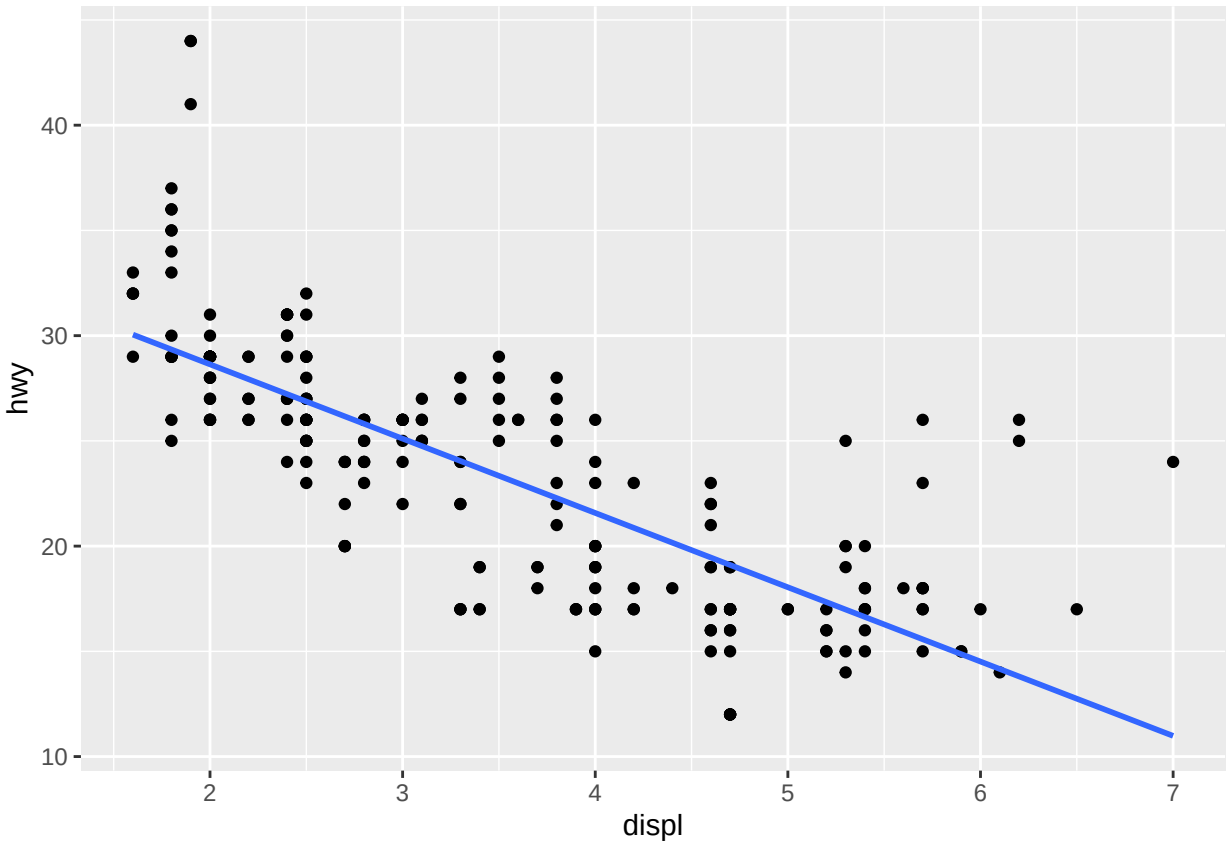
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



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(mpg, aes(displ, hwy)) +
geom_point() + geom_smooth(method = "lm", se = FALSE)
```

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


hwy
40
30
20
10
2
3
4
5
6
7
displ