WOrksheet#6

BRYCE KENDRIC SALADAR

2022-11-25

Use the dataset mpg

install.packages("dplyr")

```
library(ggplot2)
```

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

```
library(dplyr)
```

```
## Warning: package ’dplyr’ was built under R version 4.2.2
```

```
##
## 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
```

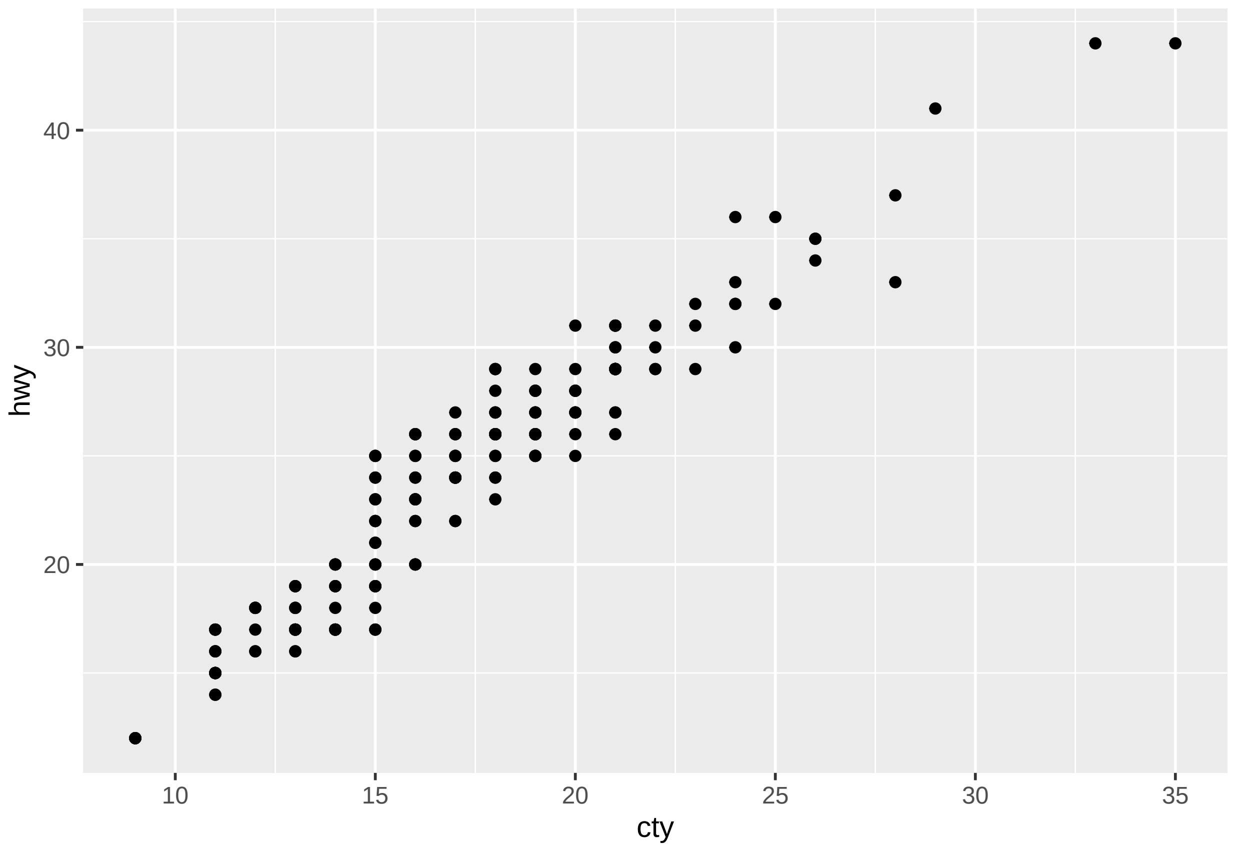
```
glimpse(mpg)
```

```
## Rows: 234
## Columns: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "audi", "~
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 quattro", "~
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0, 2.0, 2.~
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1999, 200~
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6, 8, 8, ~
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)", "auto~
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4", "4", "4~
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 17, 17, 1~
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 25, 25, 2~
## $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p~
## $ class        <chr> "compact", "compact", "compact", "compact", "compact", "c~
```

```
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" ...
```

```
ggplot(mpg,aes(cty, hwy)) +
  geom_point()
```



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

```
data(mpg)
dat <- as.data.frame(mpg)
nrow(mpg)
```

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

```
ncol(mpg)
```

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

```
#2 Which manufacturers has the most models in this dataset? Which model has the
#most variations? Ans:

modmt <- dat %>% group_by(manufacturer, model) %>%
  count()
modmt
```

```
## # 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     5
##  5 chevrolet    corvette               5
##  6 chevrolet    k1500 tahoe 4wd        4
##  7 chevrolet    malibu                 5
##  8 dodge        caravan 2wd           11
##  9 dodge        dakota pickup 4wd      9
## 10 dodge        durango 4wd            7
## # ... with 28 more rows
```

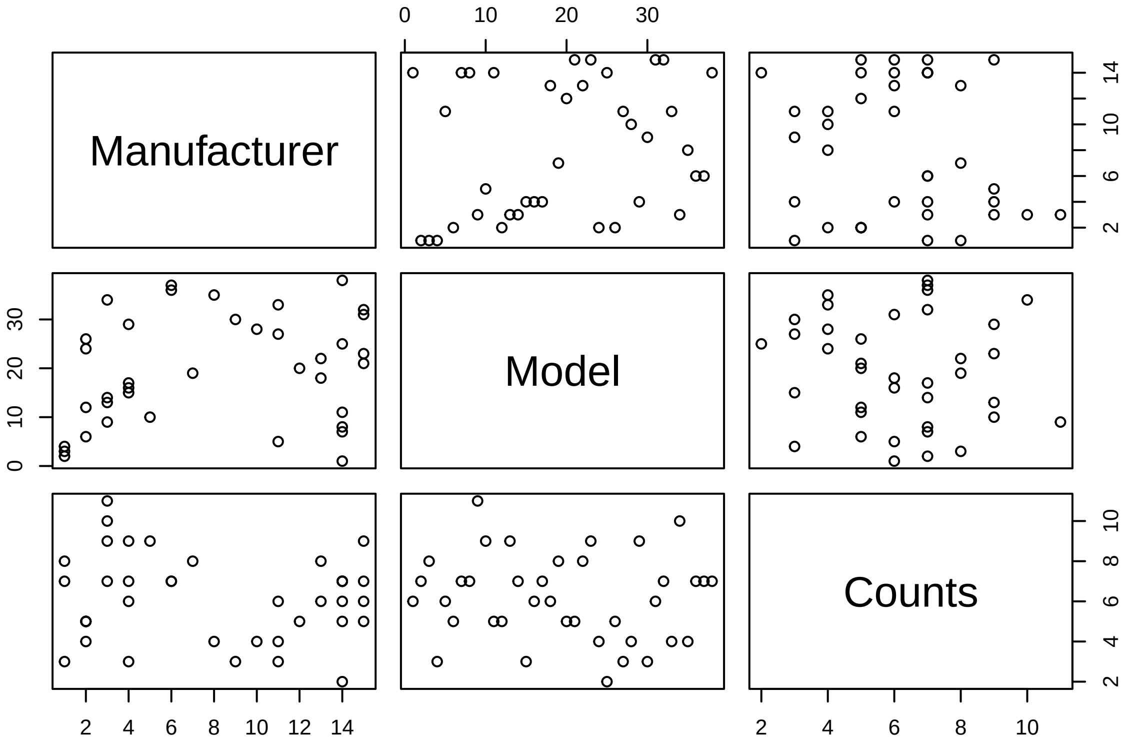
```
colnames(modmt) <- c("Manufacturer","Model","Counts")
```

```
# a. Group the manufacture and find the unique models.Copy the codes and results
uniqmod <- dat %>% group_by(manufacturer,model) %>%
  distinct() %>% count()
uniqmod
```

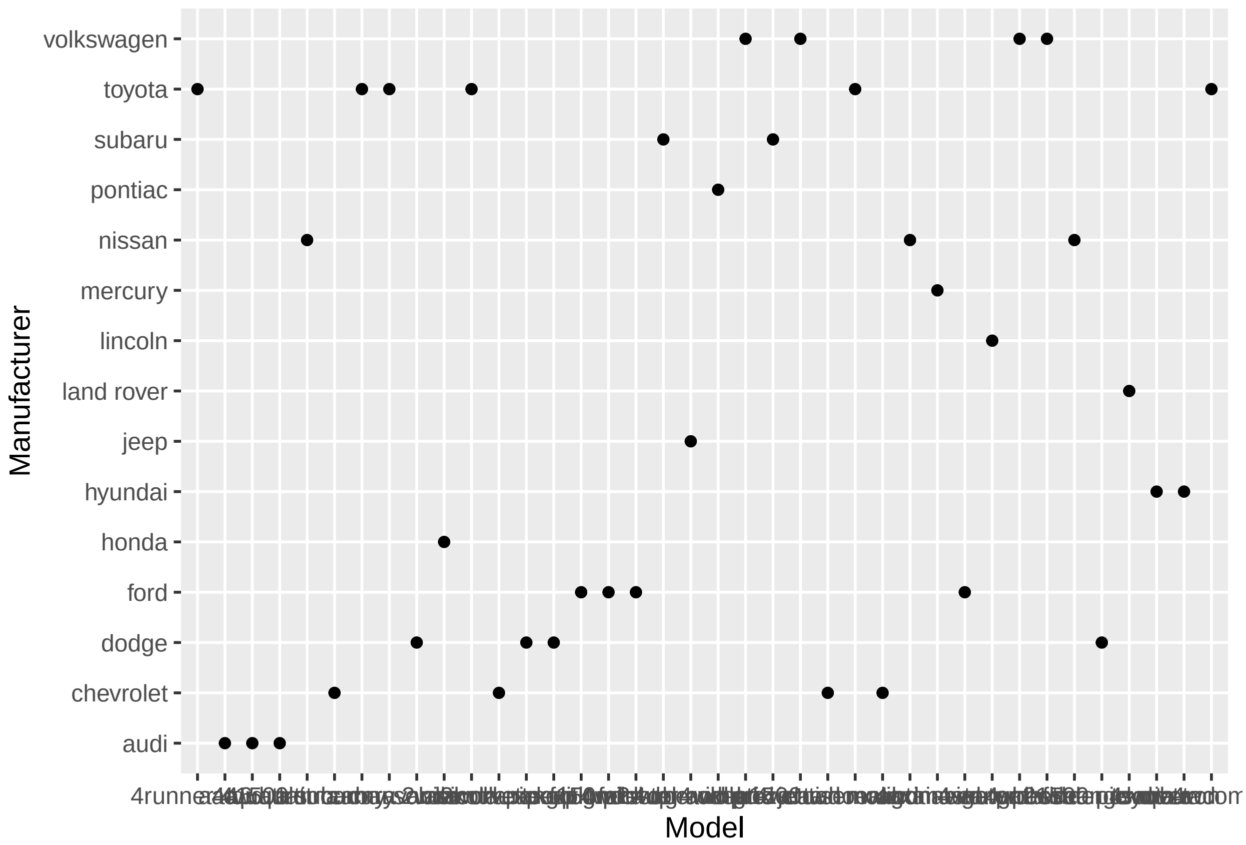
```
## # A tibble: 38 x 3
## # Groups:   manufacturer, model [38]
##    manufacturer model                  n
##    <chr>        <chr>              <int>
##  1 audi         a4                     7
##  2 audi         a4 quattro             8
##  3 audi         a6 quattro             3
##  4 chevrolet    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(uniqmod) <- c("Manufacturer", "Model", "Counts")
```

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

plot(modmt)
```



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



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

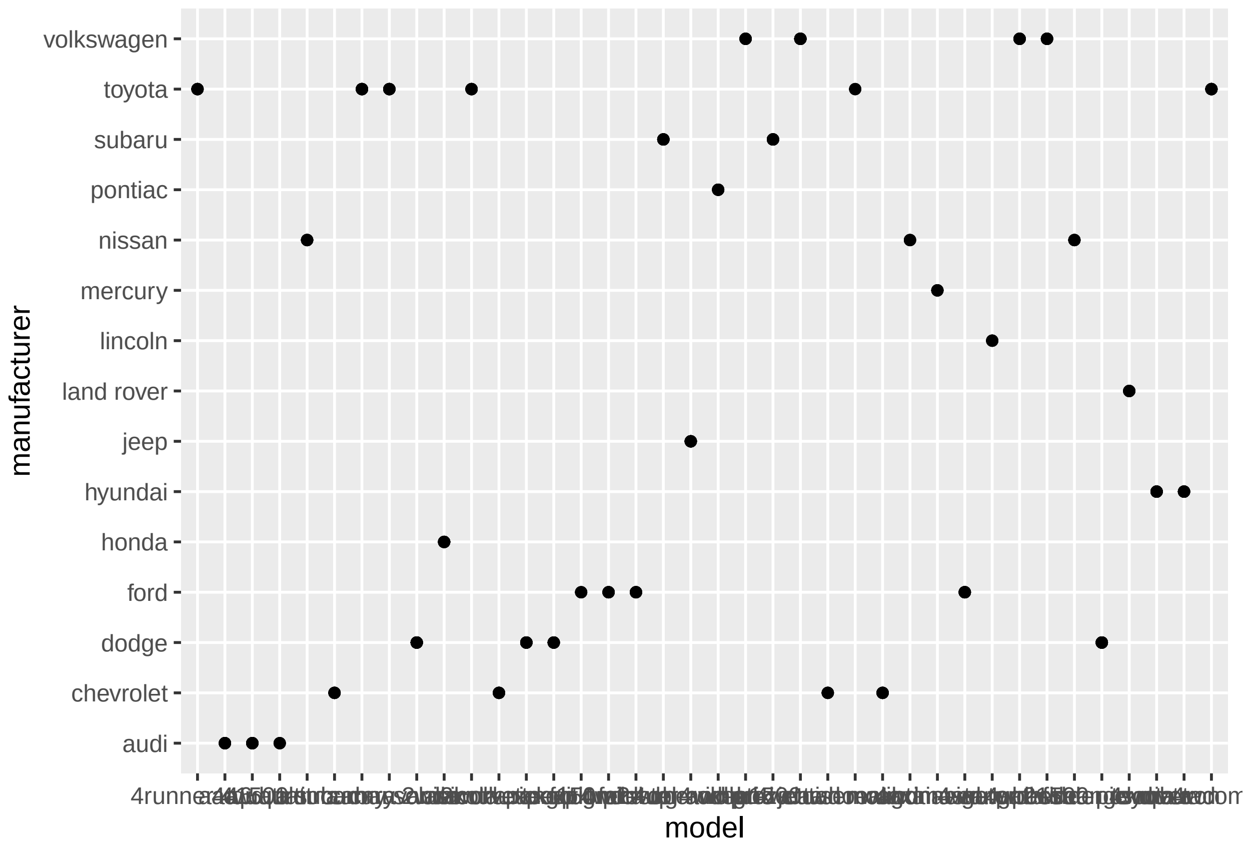
```
datampg <- mpg
dat_mod <- datampg %>% group_by(manufacturer, model) %>%
  distinct() %>% count()
dat_mod
```

```
## # 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(dat_mod) <- c("Manufacturer","Model")
dat_mod
```

```
## # 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()
```



```
# ANs: it reveals the exact graph of the geometric point of the Model and
#Manufacturer



#b. For you. is it useful? If not. how could you modify the data to make it more
# informative?
# ANS: after the plot was graph and arrange. It can be informative by adding
#some sort of legend for easier identification of the said data.
```

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

```
datmpg <- datampg %>% group_by(model) %>% count()
datmpg
```

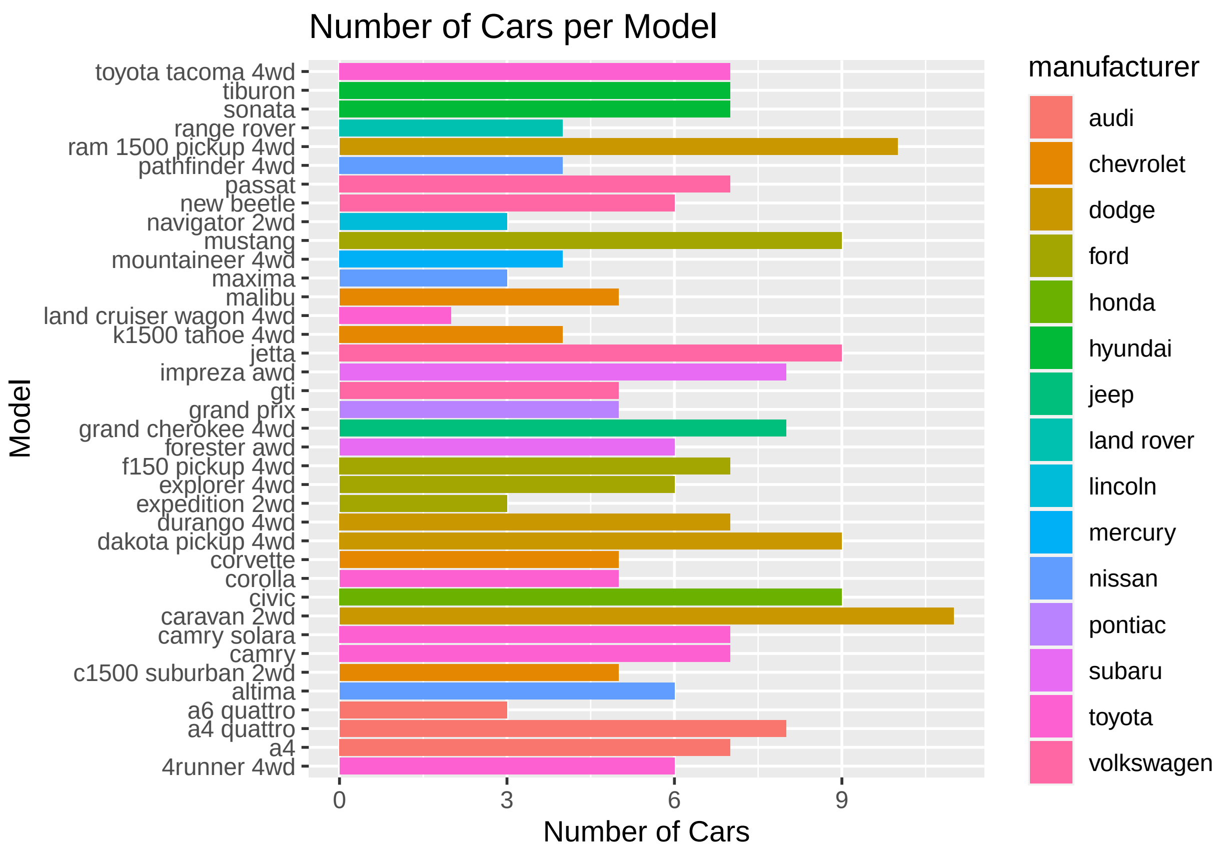
```
## # A tibble: 38 x 2
## # Groups:   model [38]
##    model                  n
##    <chr>              <int>
##  1 4runner 4wd            6
##  2 a4                     7
##  3 a4 quattro             8
##  4 a6 quattro             3
##  5 altima                 6
##  6 c1500 suburban 2wd     5
##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

```
colnames(datampg) <- c("Model","Counts")
datmpg
```

```
## # 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. 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.
```



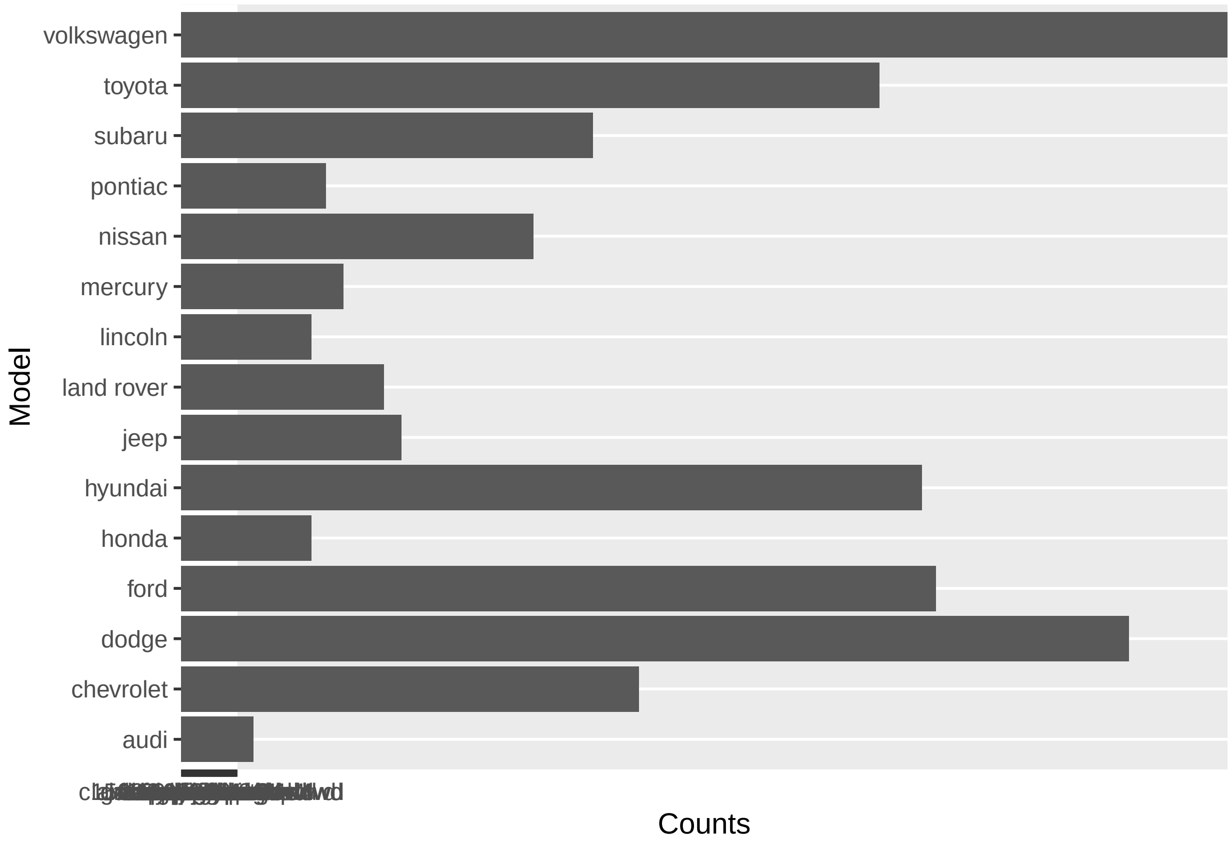
```
#b.  Use only the top 20 observations. Show code and results.

mp <- datmpg[1:20,] %>% top_n(2)
```

```
## Selecting by n
```

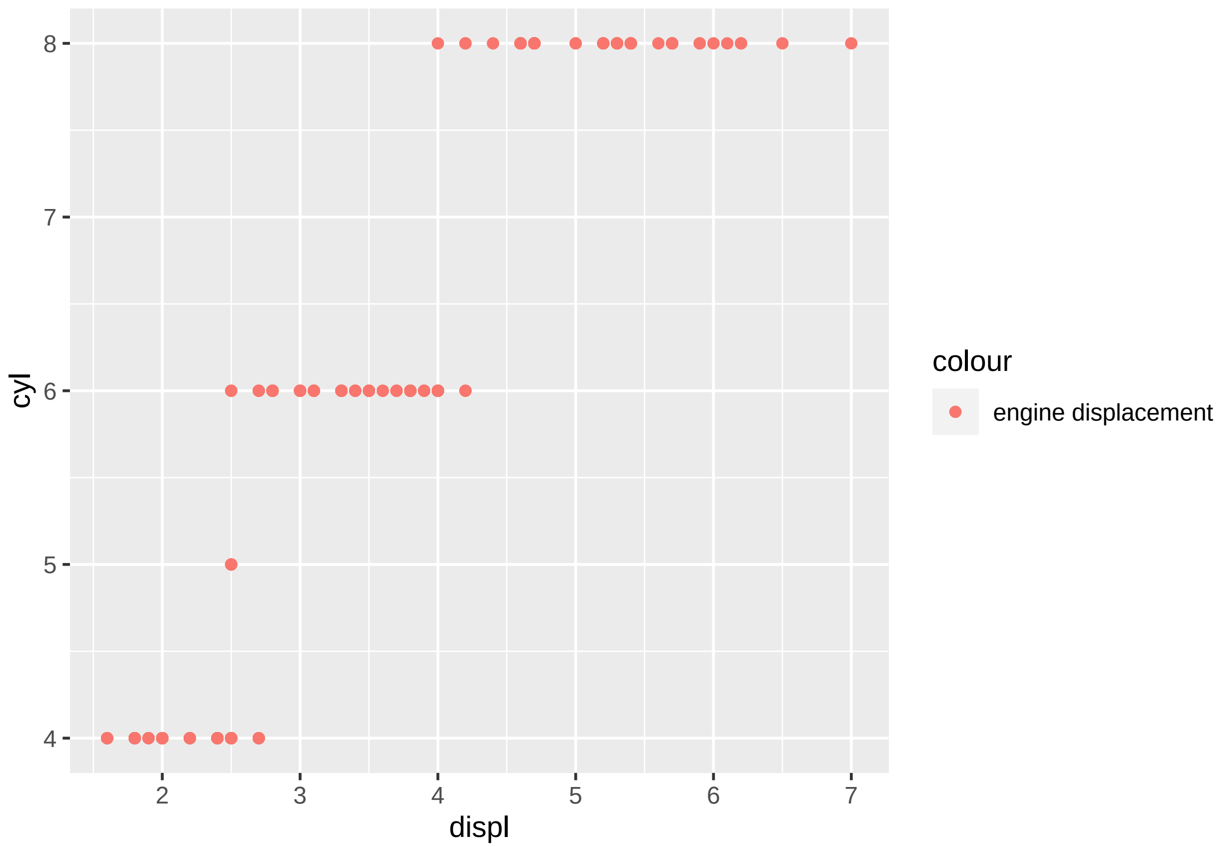
```
mp
```

```
## # A tibble: 20 x 2
## # Groups:   model [20]
##    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
## 11 corolla                5
## 12 corvette             5
## 13 dakota pickup 4wd    9
## 14 durango 4wd          7
## 15 expedition 2wd       3
## 16 explorer 4wd         6
## 17 f150 pickup 4wd      7
## 18 forester awd         6
## 19 grand cherokee 4wd   8
## 20 grand prix           5
```

```
ggplot(datampg, aes(x = Model, y = Counts)) +
  geom_bar(stat = "Identity") +coord_flip()
```



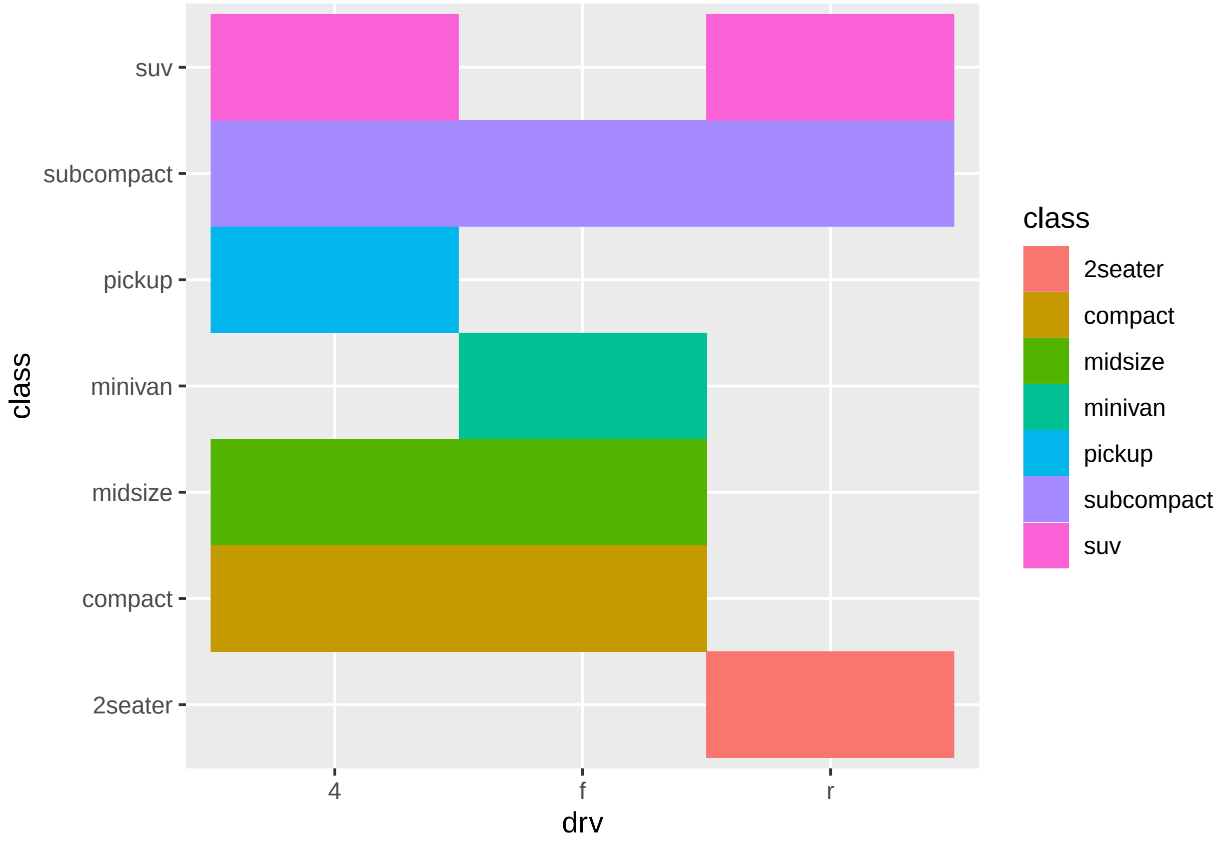
```
#5. Plot the relationship between cyl - number of cylinders and displ - engine
#displacement using geom_point with aesthetic colour = engine displacement.
#Title should be "Relationship between No. of Cylinders and Engine Displacement".

#a. Show thw 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"))
```



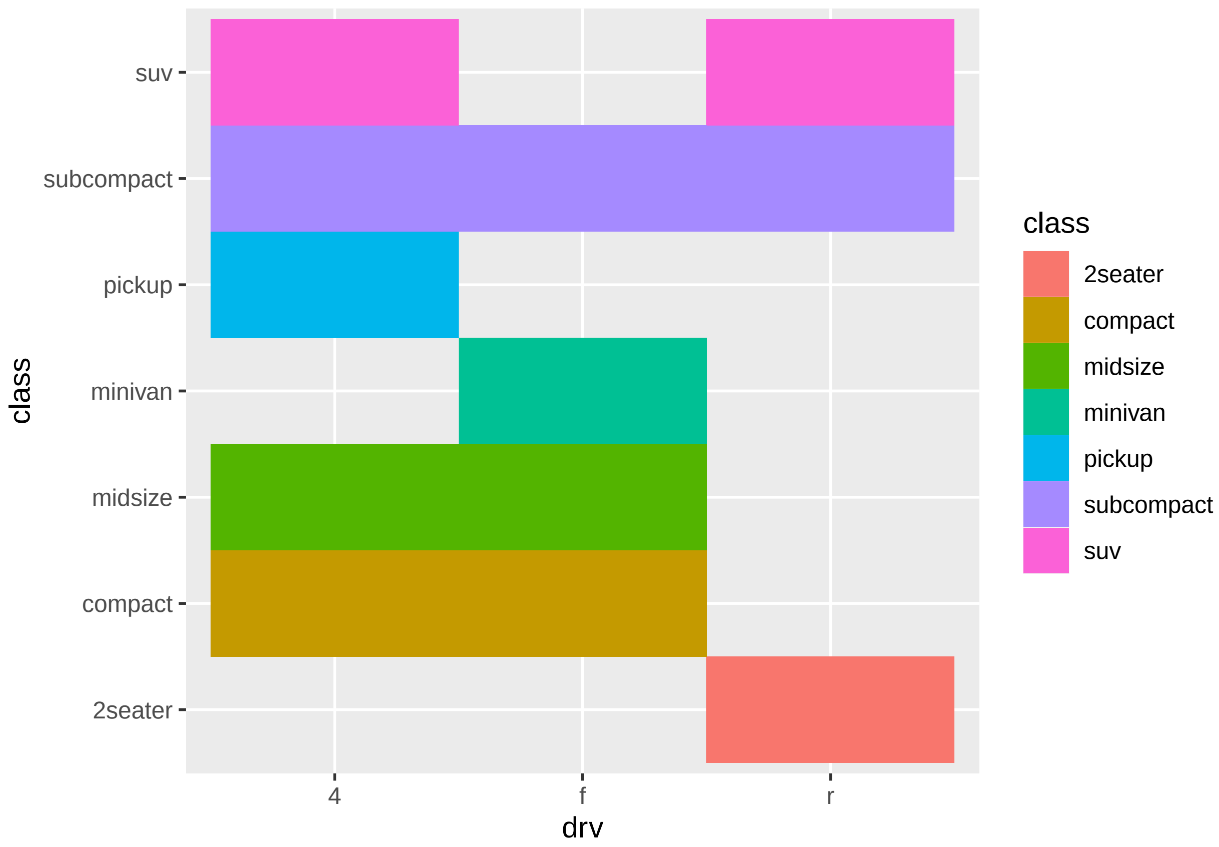
```
#b. How would you describe its relationship?

#ANS: Graph that was based on the plot reveals that cyl was on y axis and
#the displ are on x axis, which easily shows the indication of the engine
#displacement.
```

#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.

```
ggplot(data = dat, mapping = aes(x = drv, y = class)) +
  geom_tile(aes(fill=class))
```



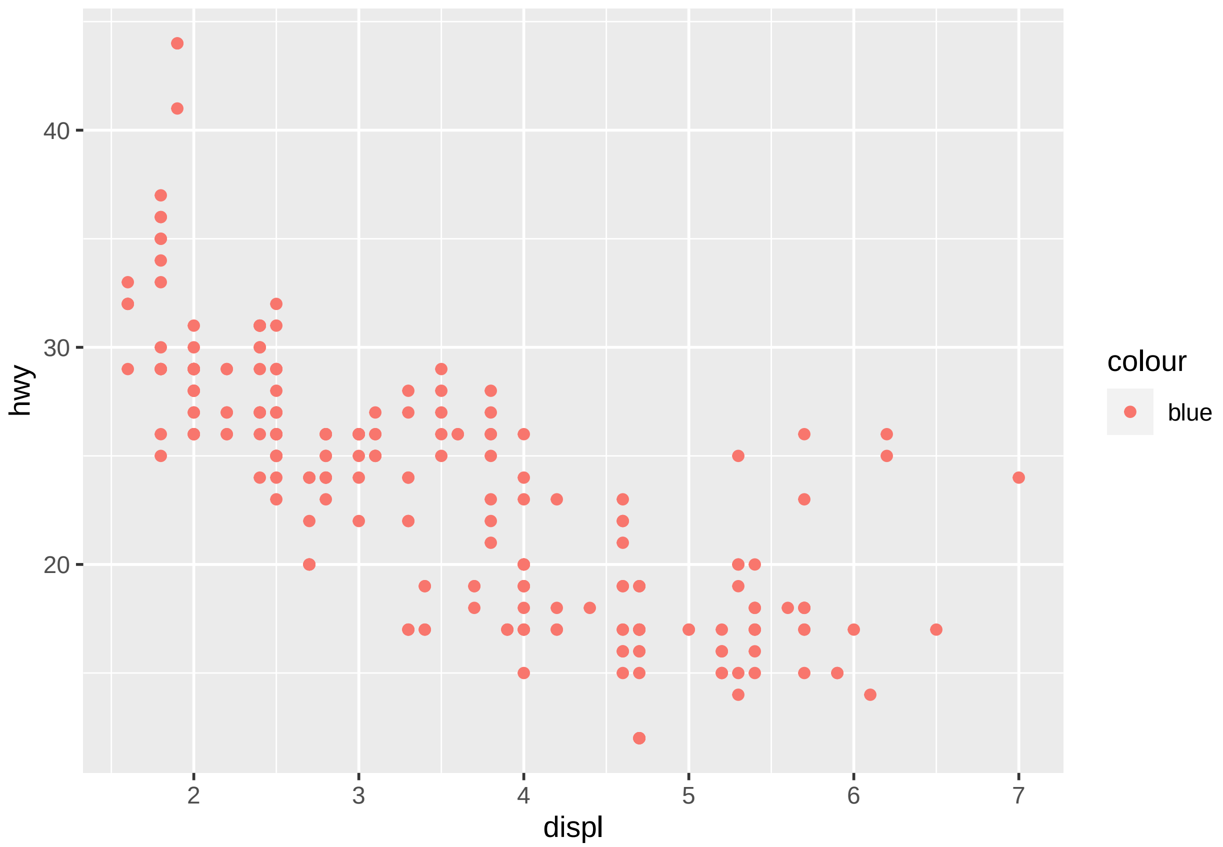
```
#a. Show the codes and its result for the narrative in #6.

ggplot(data = dat, mapping = aes(x = drv, y = class)) +
  geom_tile(aes(fill=class))
```



```
#b. Interpret the result.

#ANS: it graphs into sa tile graph that shows X axis(drv) and y axis(class).
```

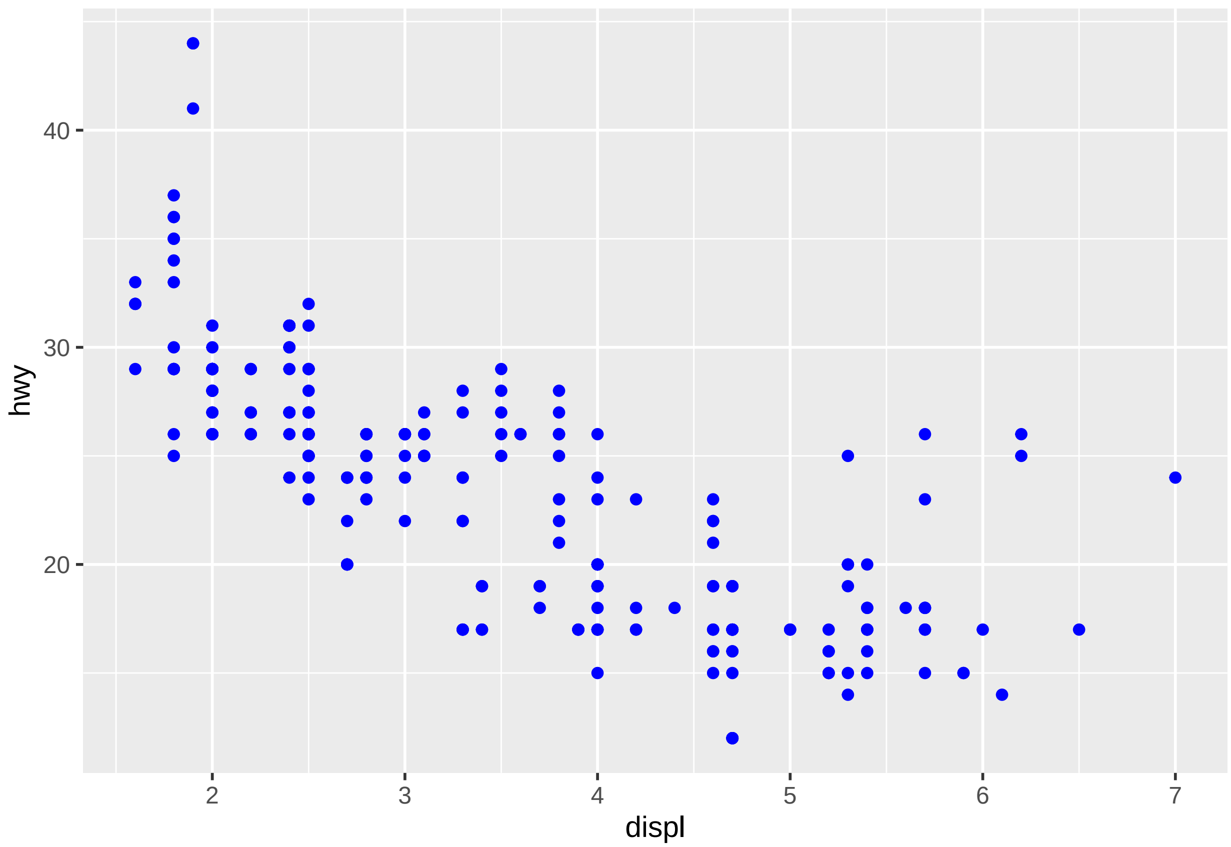
#7.Discuss the difference between these codes. Its outputs for each are shown below. # • Code #1

```
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")
```



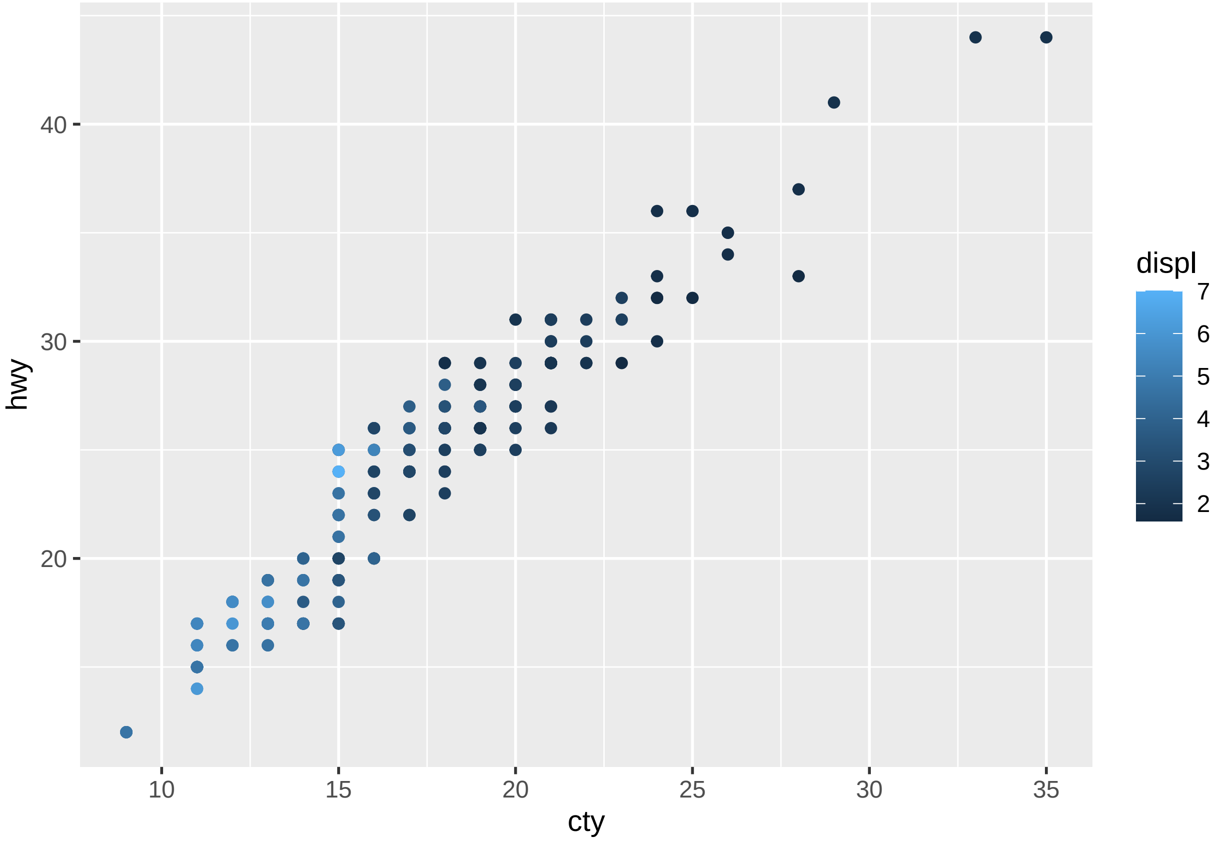
#8. Try to run the command ?mpg. What is the result of this command?

```
md <-?mpg
md
```

```
## starting httpd help server ... done
```

```
#a. Which variables from mpg dataset are categorical?

# ANS: the manufacturer, model, trans, drv, fl, class
#are the categorical variables from the dataset of mpg.



#b. Which are continuous variables?

#ANS: displ,year,cty,cyl,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?
```

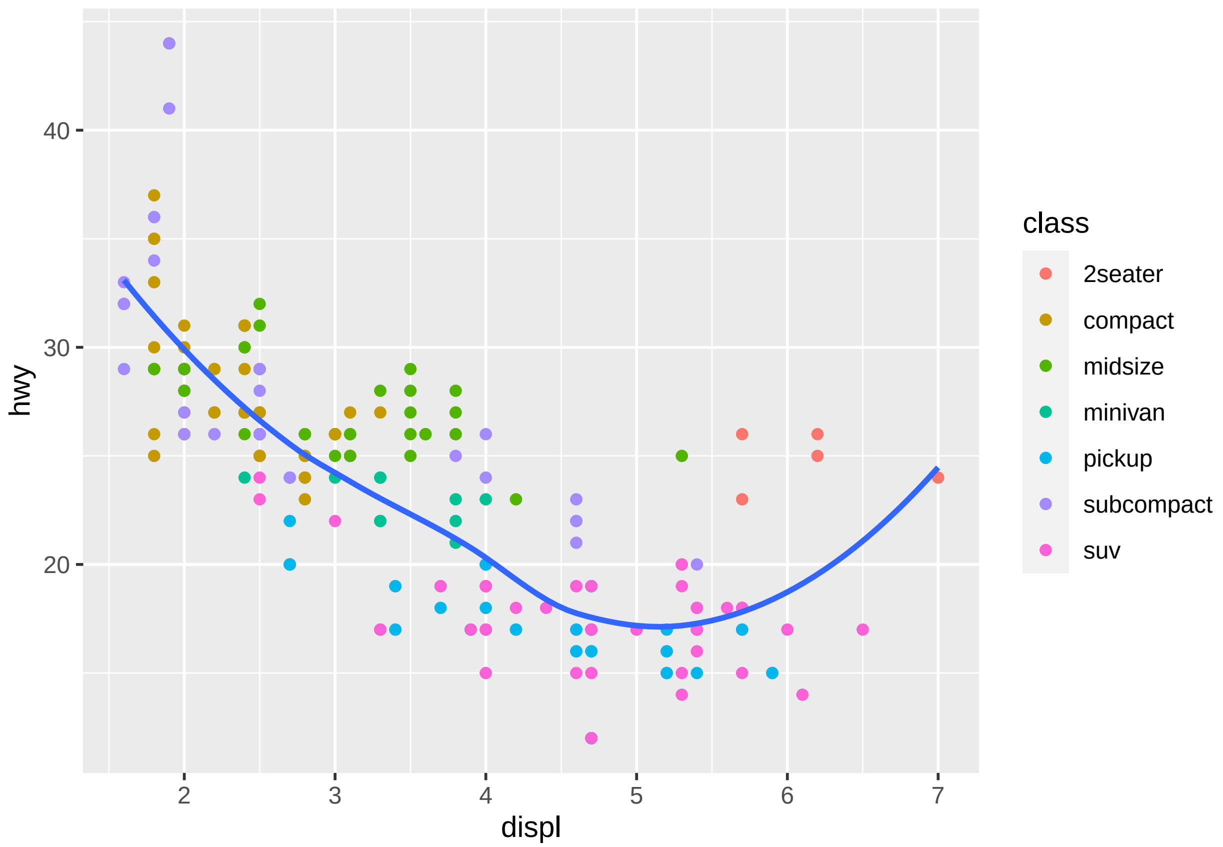
```
#ANS: the plot shows a productive rating of hwy and cty .

ggplot(mpg, aes(x = cty, y = hwy, colour = displ)) + 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)
```

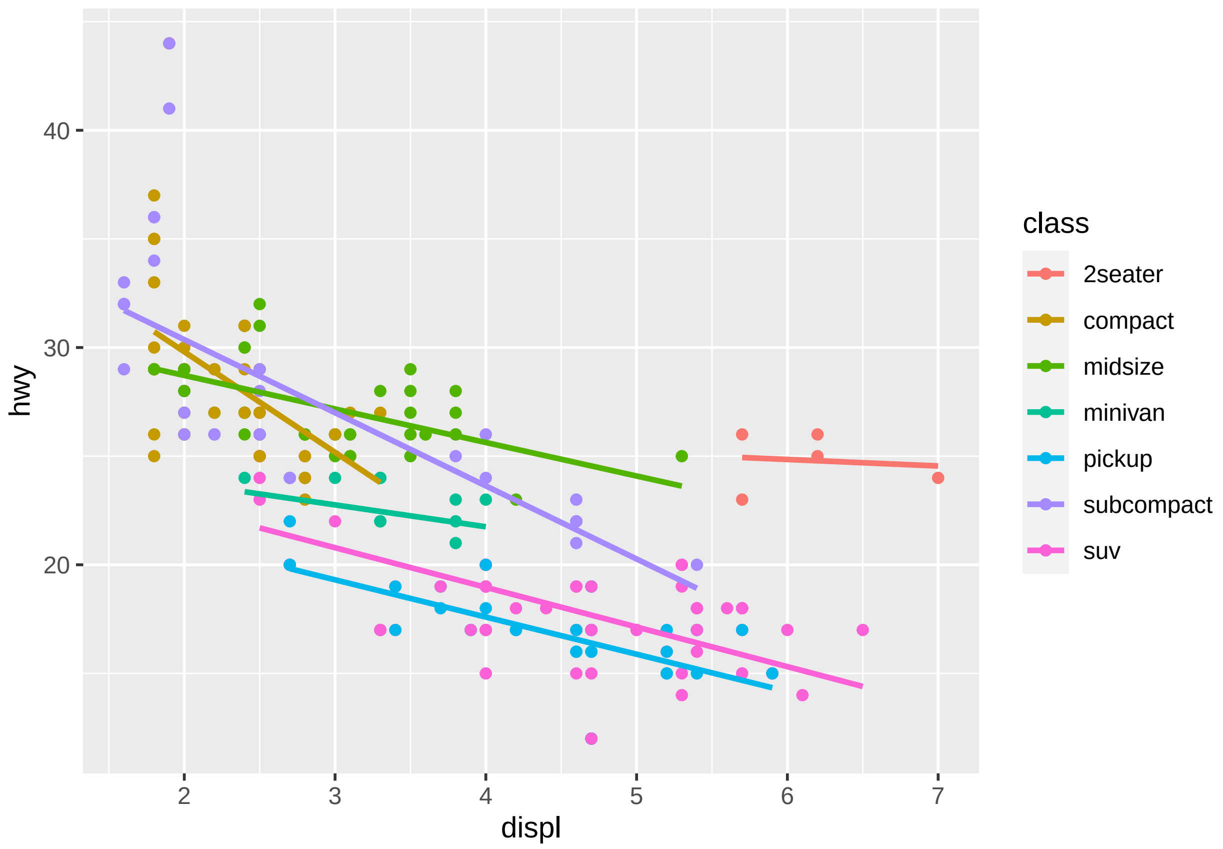
```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```



#10. Using the relationship of displ and hwy, add a trend line over existing #plot. Set these = 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, method = lm)
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```


40
30
20
hwy
2
3
4
5
6
7
displ
class
2seater
compact
midsize
minivan
pickup
subcompact
suv