RWorksheet #6

Jacklord Española

2022-11-24

```
#Use the dataset mpg
library(ggplot2)

#to get the mpg dataset, load the ggplot package first
data(mpg)
#to get the mpg dataset, load the ggplot package first
data(mpg)
data1 <- as.data.frame(data(mpg)) #converting from list to data frame

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" ...
```

```
#use of glimpse() - much tidier compared to str()
library(dplyr) #glimpse() is a function under dplyr package
```

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

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

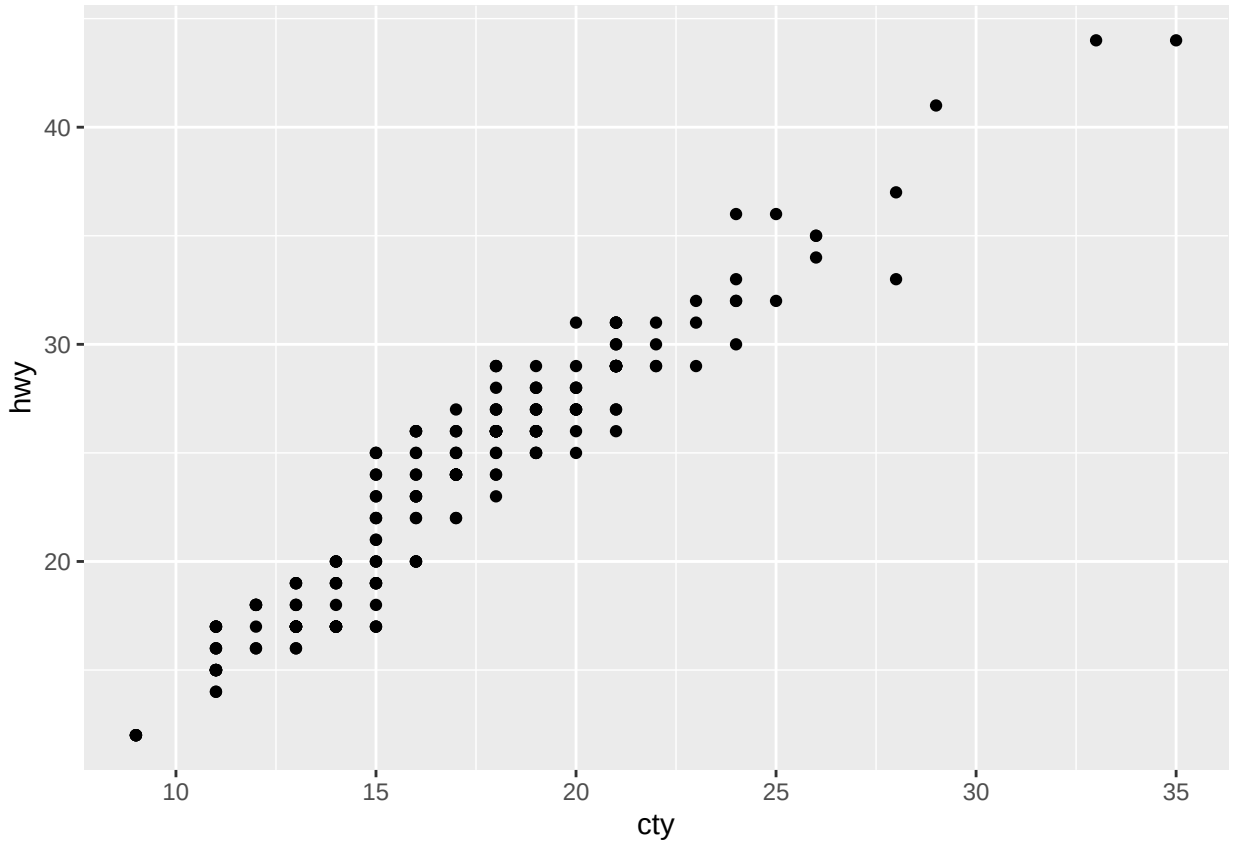
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
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~
```

```
#Example. graph using ggplot()
ggplot(mpg, aes(cty, hwy)) +
  geom_point()
```




```
#1. How many columns are in mpg dataset? How about the number of rows? Show the
#codes and its result.
nrow(mpg) #234
```

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

```
ncol(mpg) #11
```

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

```
#2. Which manufacturer has the most models in this data set? Which model has the
# most variations? Ans: dodge has 37 models

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

```
manufc
```

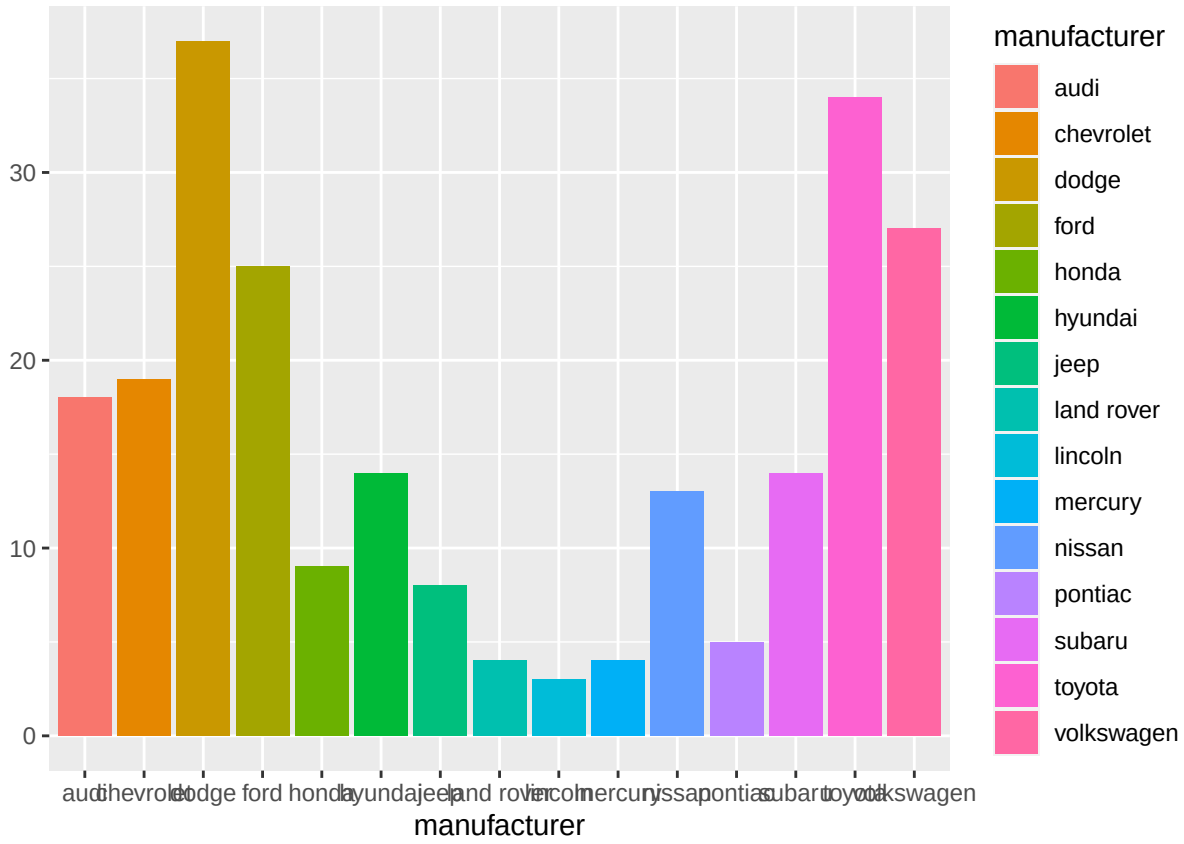
```
## # A tibble: 15 x 2
##    manufacturer     n
##    <chr>        <int>
##  1 dodge           37
##  2 toyota          34
##  3 volkswagen      27
##  4 ford            25
##  5 chevrolet       19
##  6 audi            18
##  7 hyundai         14
##  8 subaru          14
##  9 nissan          13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

```
#a. Group the manufacturers and find the unique models. Copy the codes and
#result.
mydata <- mpg
grp_manufc <- mydata %>% group_by(manufacturer, model) %>%
  distinct() %>% count()
grp_manufc
```

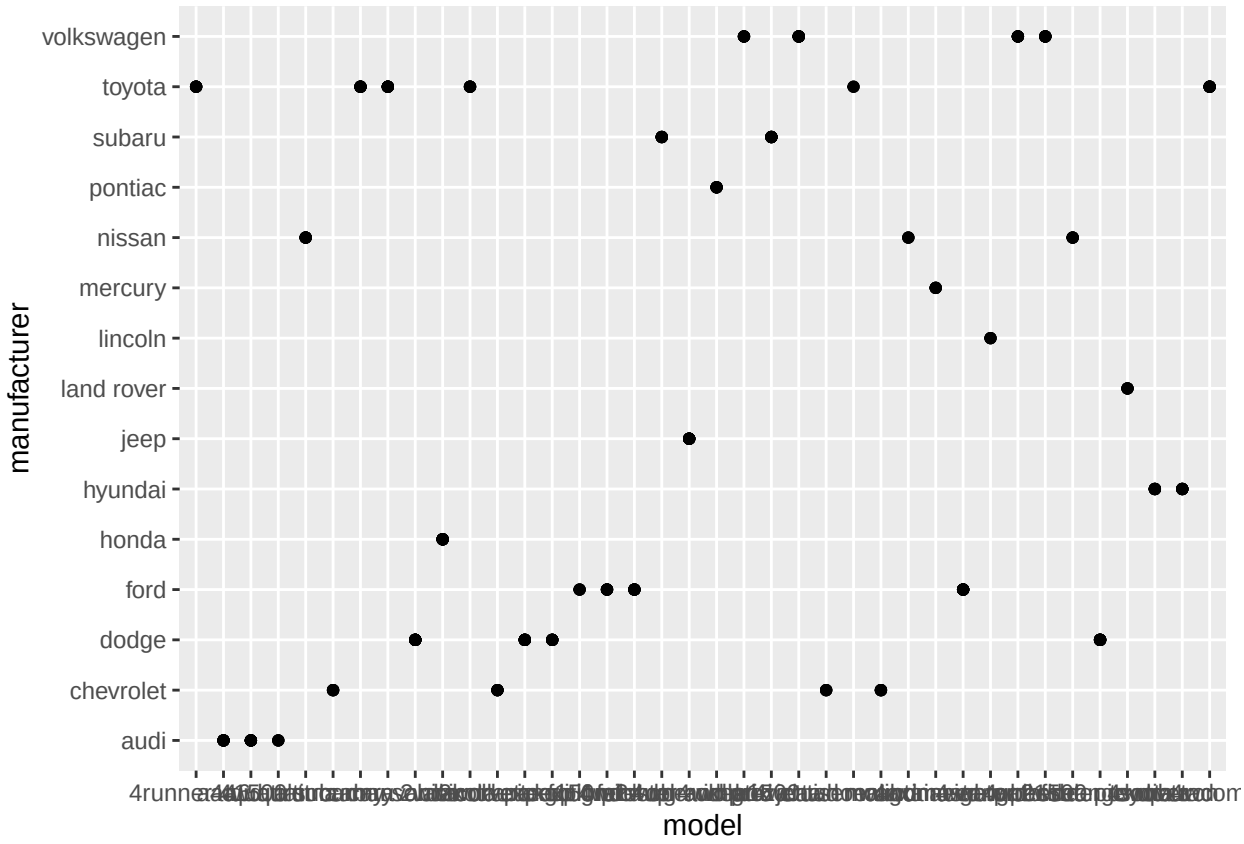
```
## # A tibble: 38 x 3
## # Groups:   manufacturer, model [38]
##    manufacturer model                  n
##    <chr>        <chr>              <int>
##  1 audi         a4                     7
##  2 audi         a4 quattro             8
##  3 audi         a6 quattro             3
##  4 chevrolet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

```
#b. Graph the result by using plot() and ggplot(). Write the codes and its
#result.
qplot(manufacturer, data=mpg, geom = "bar", fill=manufacturer )
```

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




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




```
#3. Same dataset will be used. You are going to show the relationship of the
#model and the manufacturer.
data1 <- mpg
data2 <- data1 %>% group_by(manufacturer, model) %>%
  distinct() %>% count()
data2
```

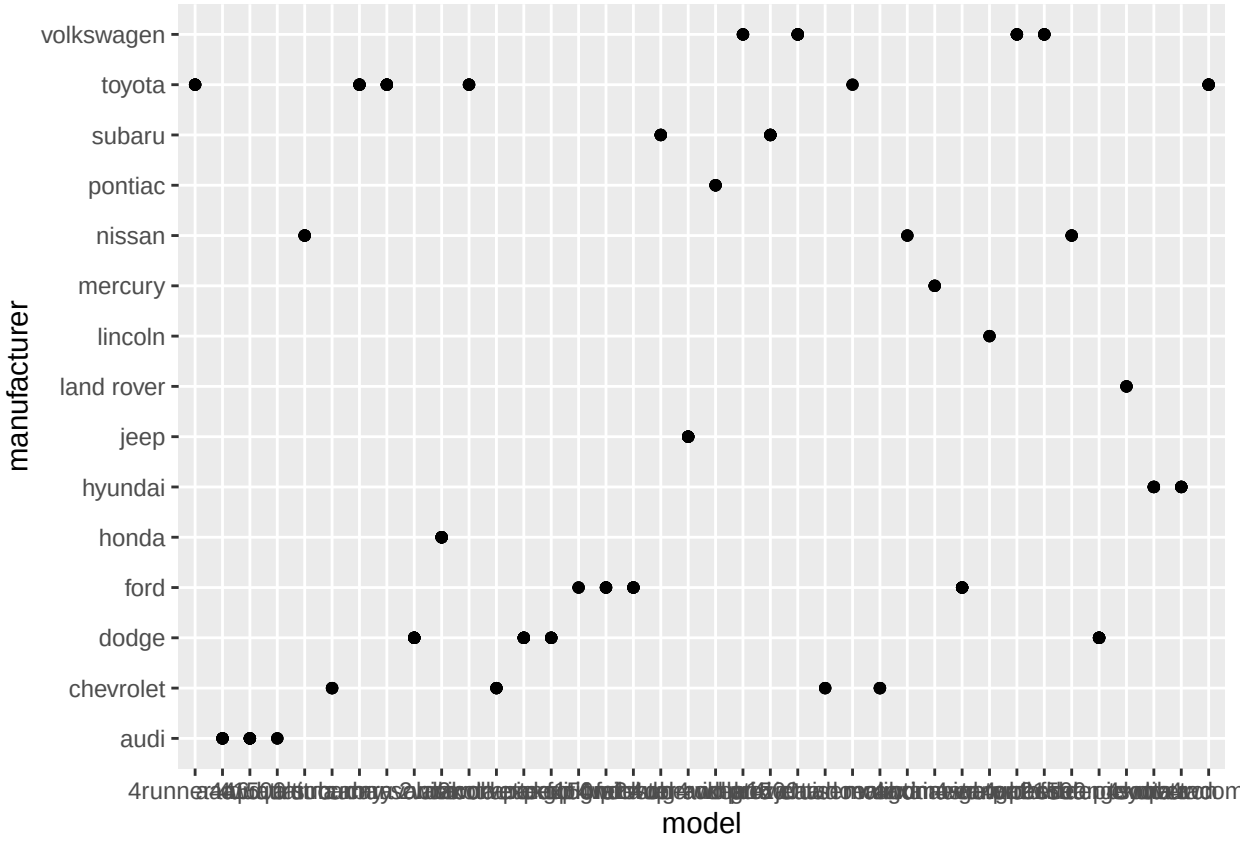
```
## # A tibble: 38 x 3
## # Groups:   manufacturer, model [38]
##    manufacturer model                 n
##    <chr>        <chr>             <int>
##  1 audi         a4                    7
##  2 audi         a4 quattro            8
##  3 audi         a6 quattro            3
##  4 chevrolet    c1500 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(data2) <- c("Manufacturer", "Model")
data2
```

```
## # 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()
```

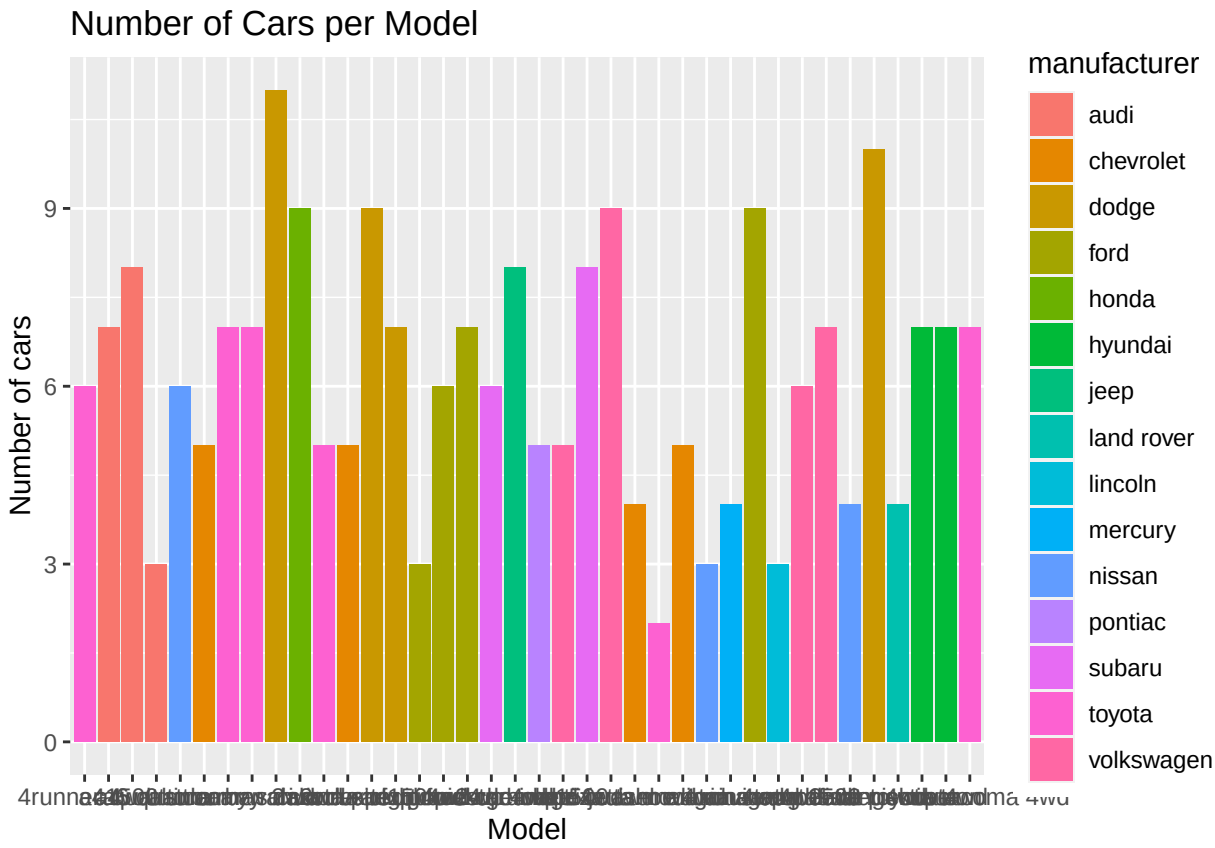



```
#Answer: It shows the geometric point graph of model and manufacturer in mpg.

#b. For you, is it useful? If not, how could you modify the data to make it more
#informative
#Answer: Yes, it is useful because you can track down the data for each model of
#the manufacturer and modify it.
```

```
#4. Using the pipe (%>%), group the model and get the number of cars per model.
#Show codes and its result.
grp_model <- grp_manufc %>% group_by(model) %>% count()
grp_model
```

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

```
colnames(grp_model) <- c("model","Counts")

#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)
```

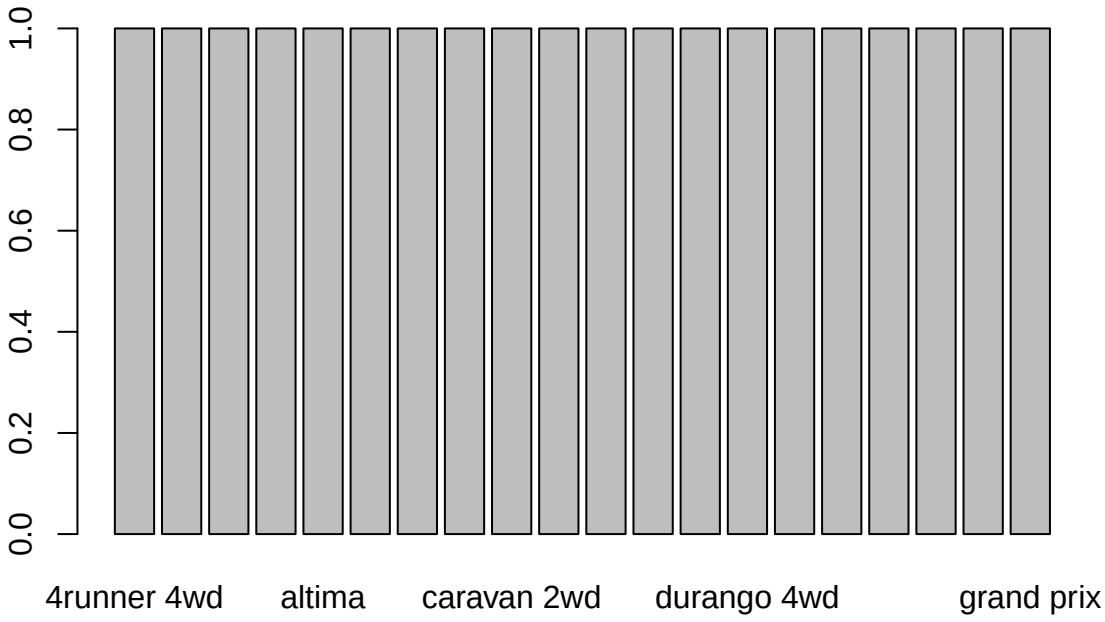



```
      coord_flip()
```

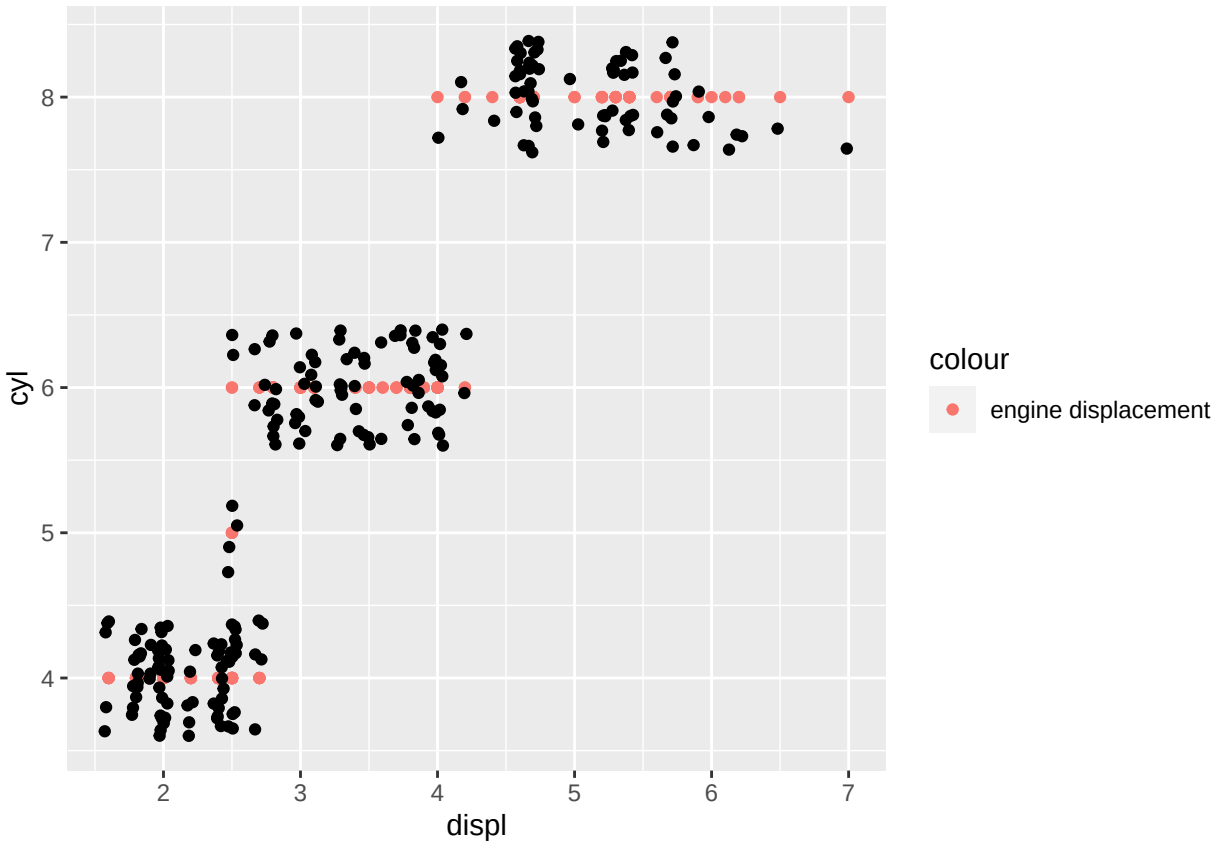
```
## <ggproto object: Class CoordFlip, CoordCartesian, Coord, gg>
```

```
##     aspect: function
##     backtransform_range: function
##     clip: on
##     default: FALSE
##     distance: function
##     expand: TRUE
##     is_free: function
##     is_linear: function
##     labels: function
##     limits: list
##     modify_scales: function
##     range: function
##     render_axis_h: function
##     render_axis_v: function
##     render_bg: function
##     render_fg: function
##     setup_data: function
##     setup_layout: function
##     setup_panel_guides: function
##     setup_panel_params: function
##     setup_params: function
##     train_panel_guides: function
##     transform: function
##     super:  <ggproto object: Class CoordFlip, CoordCartesian, Coord, gg>
```

```
#b. Use only the top 20 observations. Show code and results
barplot(grp_model$Counts[1:20],names.arg=grp_model$model[1:20])
```

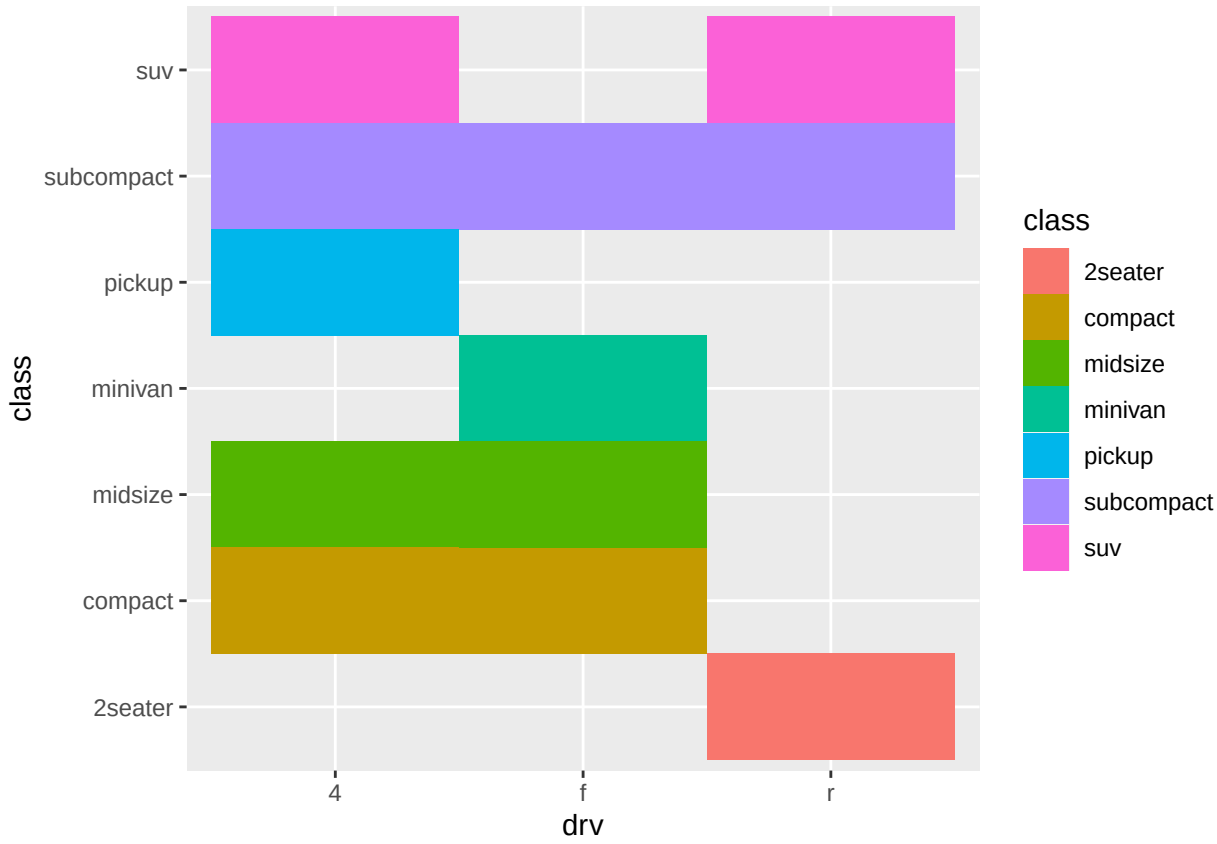



```
#5. Plot the relationship between cyl - number of cylinders and displ - #engine displacement using geom
#a. Show the codes and its result.
ggplot(data = mpg , mapping = aes(x = displ, y = cyl,
      main = "Relationship between No. of Cylinders and Engine Displacement")) +
      geom_point(mapping=aes(colour = "engine displacement")) + geom_jitter()
```



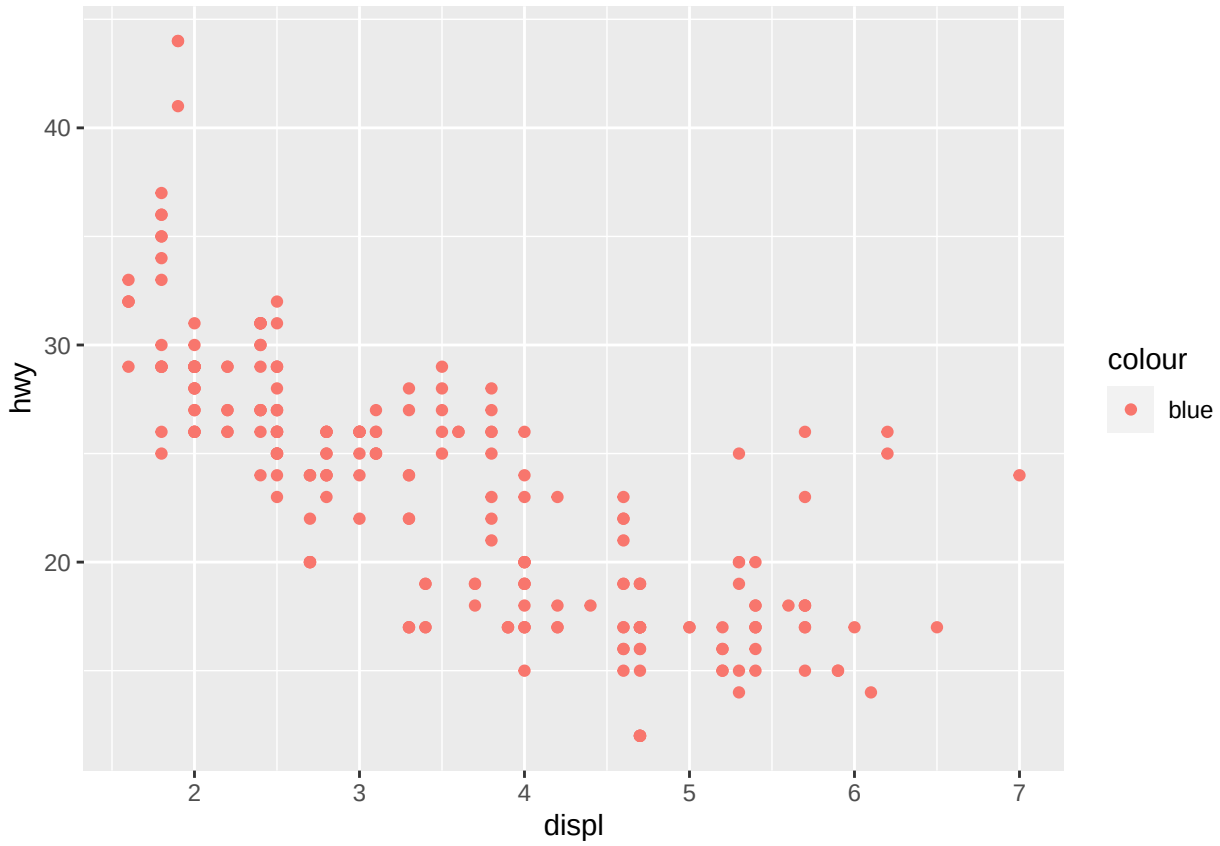

```
#b. How would you describe its relationship?
#Answer: Looking at the graph, the cyl represents the y, the graph is #gittered and the pink dots repre
```

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

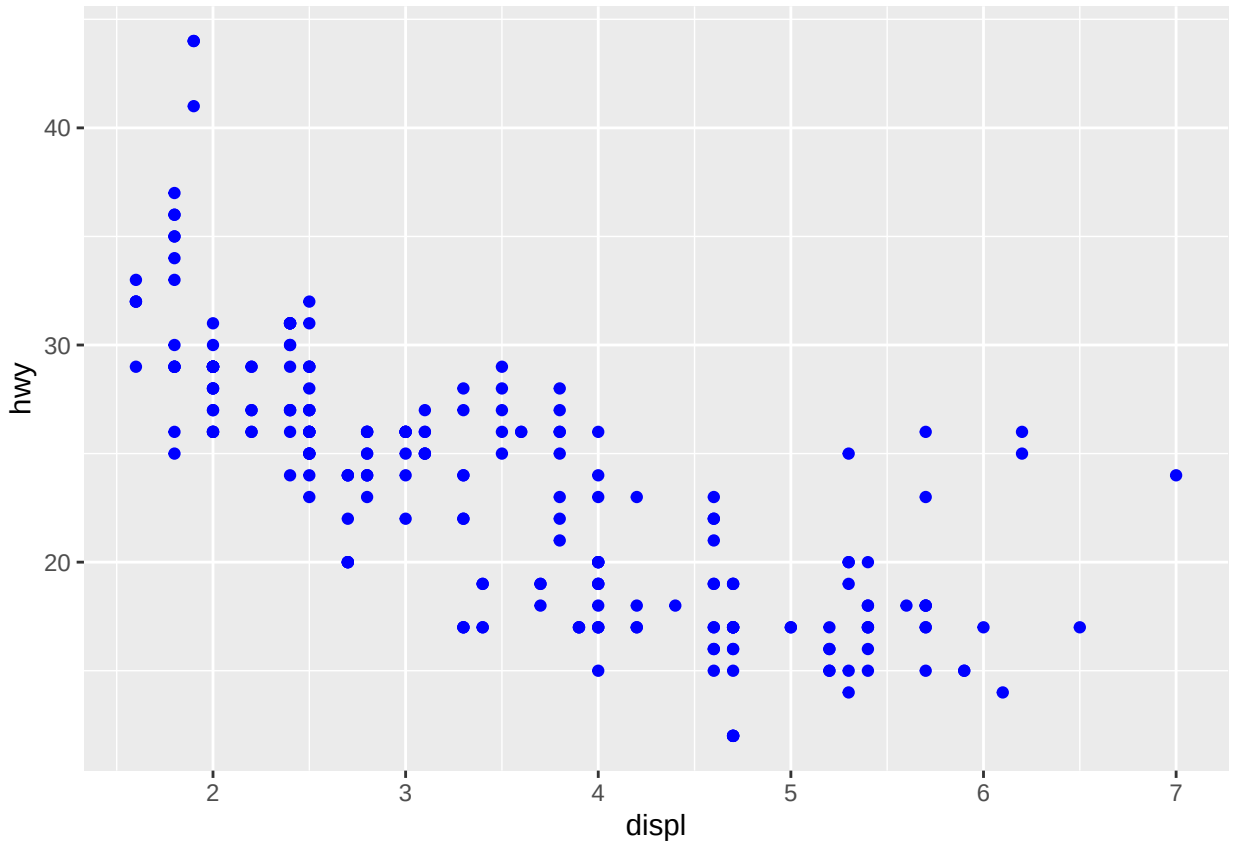



```
#b. Interpret the result.
#Answer: Areas covered with black are "mapped" using the mapping geometric point
#graph. The y as class and x as drv.
```

```
#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")
```



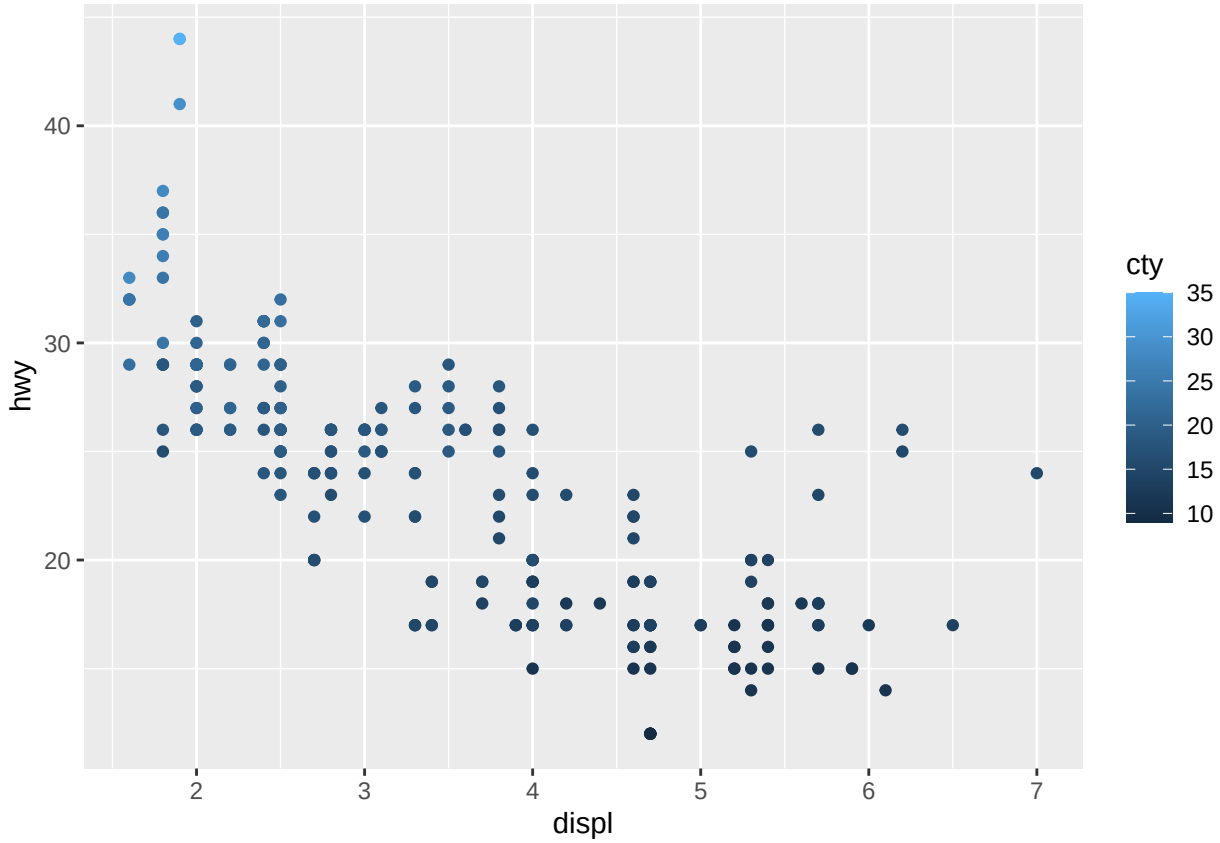

```
#In the expression, colour = "blue", "blue" is interpreted as a categorical
#variable which only takes a single value "blue". Considering how colour = 1:234
#and colour = 1 are interpreted by aes().
```

```
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
```

```
#a. Which variables from mpg dataset are categorical?
#Categorical variables in mpg include: manufacturer, model, trans
#(type of transmission), drv (front-wheel drive, rear-wheel, 4wd),
#fl (fuel type), and class (type of car).

#b. Which are continuous variables?
#Continuous variables in mpg include: displ (engine displacement in litres),cyl
```

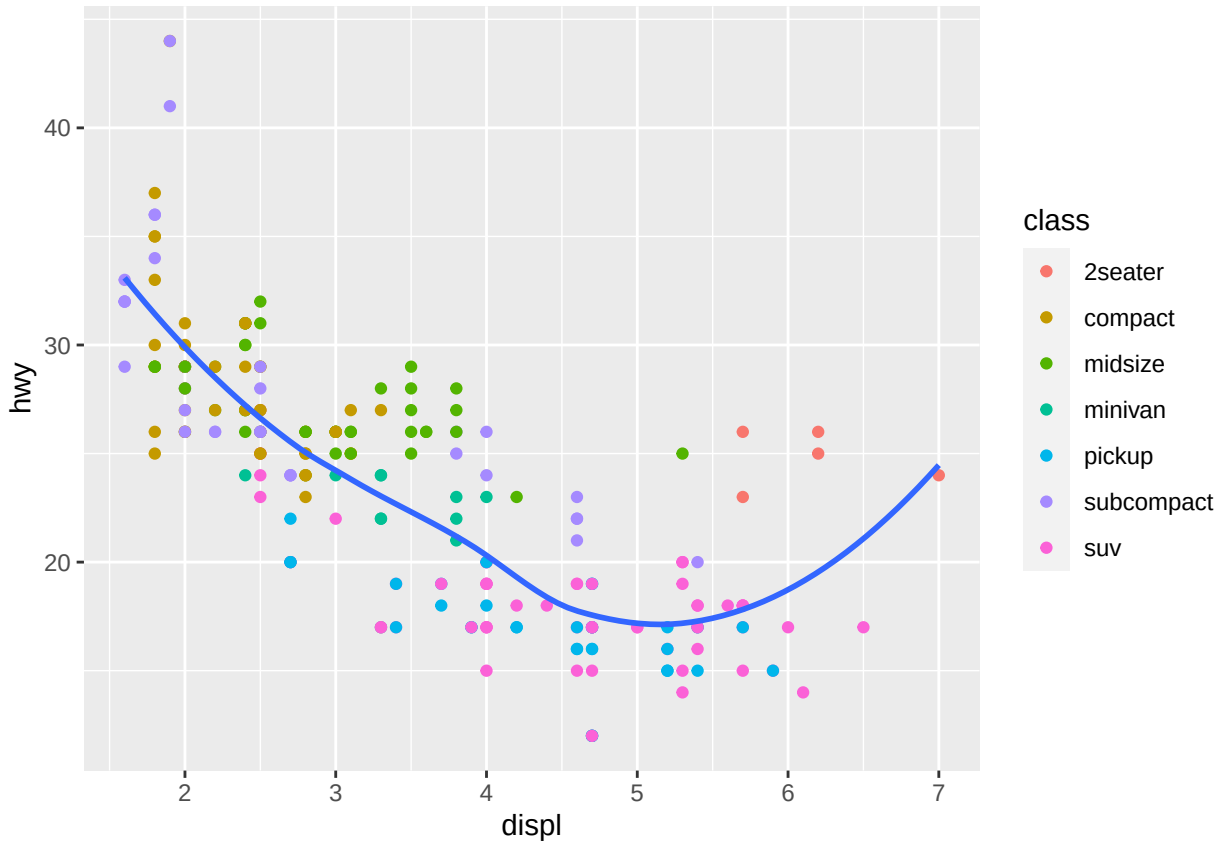
```
#(number of cylinders), cty (city miles/gallon), and hwy (highway gallons/mile).

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




```
#What is its result? Why it produced such output?
#Answer: data tracks the cty by placing cty(city miles per gallon) at color
#having a variation or hues of blue.
```

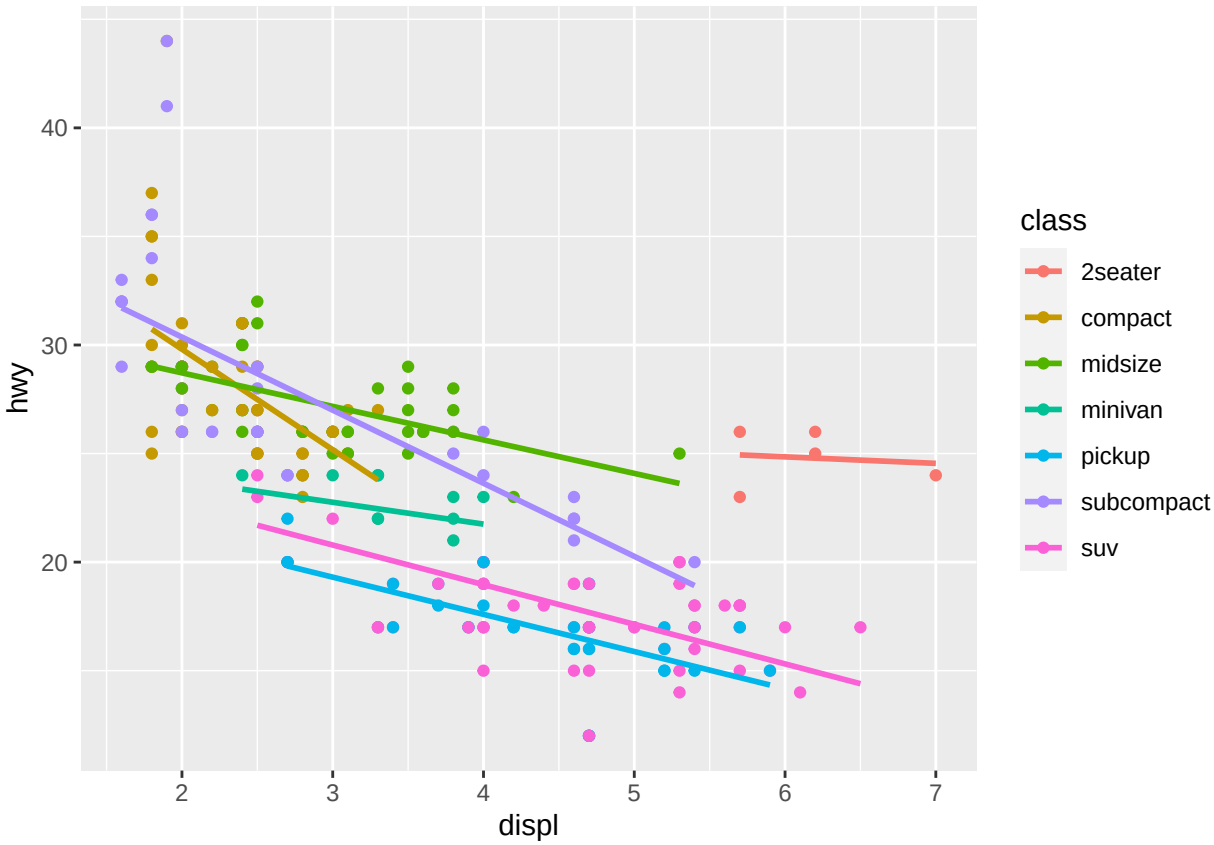
```
#9. Plot the relationship between displ (engine displacement) and hwy(highway
#miles per gallon) using geom_point(). Add a trend line over the existing plot
#using geom_smooth() with se = FALSE. Default method is "loess".
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping=aes(color=class)) +
  geom_smooth(se = FALSE,  method = loess)
```

```
## `geom_smooth()` using 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, method = lm)
```

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


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