Kent Gener

Kent Hervey D. Gener BSIT2-B

2023-11-22

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

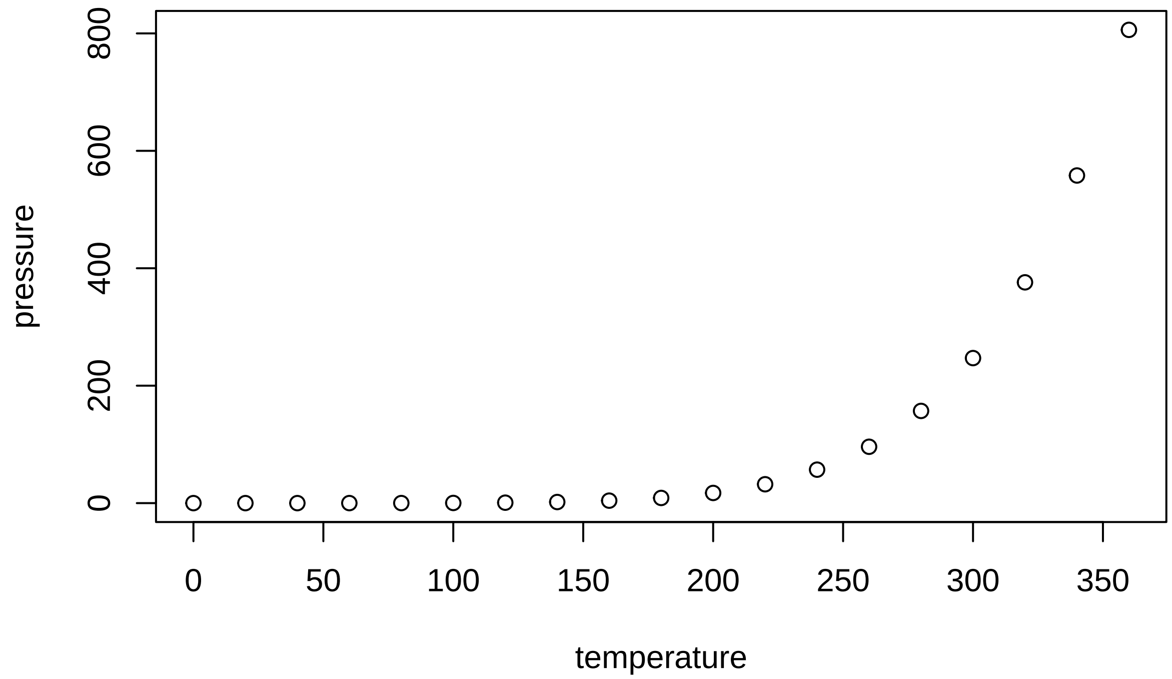
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

- Download and open the mpg file. Upload it to your OWN environment a. Show your solutions on how to import a csv file into the environment.

```
mpg_data <- read.csv('mpg.csv')
```

b. Which variables from mpg dataset are categorical?

```
str(mpg_data)
```

```
## 'data.frame':    234 obs. of  12 variables:
##  $ X           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
##  $ model       : chr  "a4" "a4" "a4" "a4" ...
##  $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int  1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int  4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr  "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr  "f" "f" "f" "f" ...
##  $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr  "p" "p" "p" "p" ...
##  $ class       : chr  "compact" "compact" "compact" "compact" ...
```

```r
categorical_vars <- c("manufacturer", "model", "year", "cyl", "trans", "drv", "fl", "class")
cat("Categorical variables:", categorical_vars, "\n")
```

```
## Categorical variables: manufacturer model year cyl trans drv fl class
```

c. Which are continuous variables?

```r
str(mpg_data)
```

```
## ’data.frame’:    234 obs. of  12 variables:
##  $ X           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
##  $ model       : chr  "a4" "a4" "a4" "a4" ...
##  $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int  1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int  4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr  "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr  "f" "f" "f" "f" ...
##  $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr  "p" "p" "p" "p" ...
##  $ class       : chr  "compact" "compact" "compact" "compact" ...
```

```r
continuous_vars <- c("displ", "cty", "hwy")
cat("Continuous variables:", continuous_vars, "\n")
```

```
## Continuous variables: displ cty hwy
```

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

```r
library(dplyr)
```

```
## Warning: package ’dplyr’ was built under R version 4.3.2

##
## 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
```

```r
models_per_manufacturer <- mpg_data %>%
  group_by(manufacturer) %>%
  summarise(unique_models = n_distinct(model))

models_per_manufacturer
```

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

a. Group the manufacturers and find the unique models. Show your codes and result.

```
library(dplyr)

models_per_manufacturer <- mpg_data %>%
  group_by(manufacturer) %>%
  summarise(unique_models = n_distinct(model))

models_per_manufacturer
```

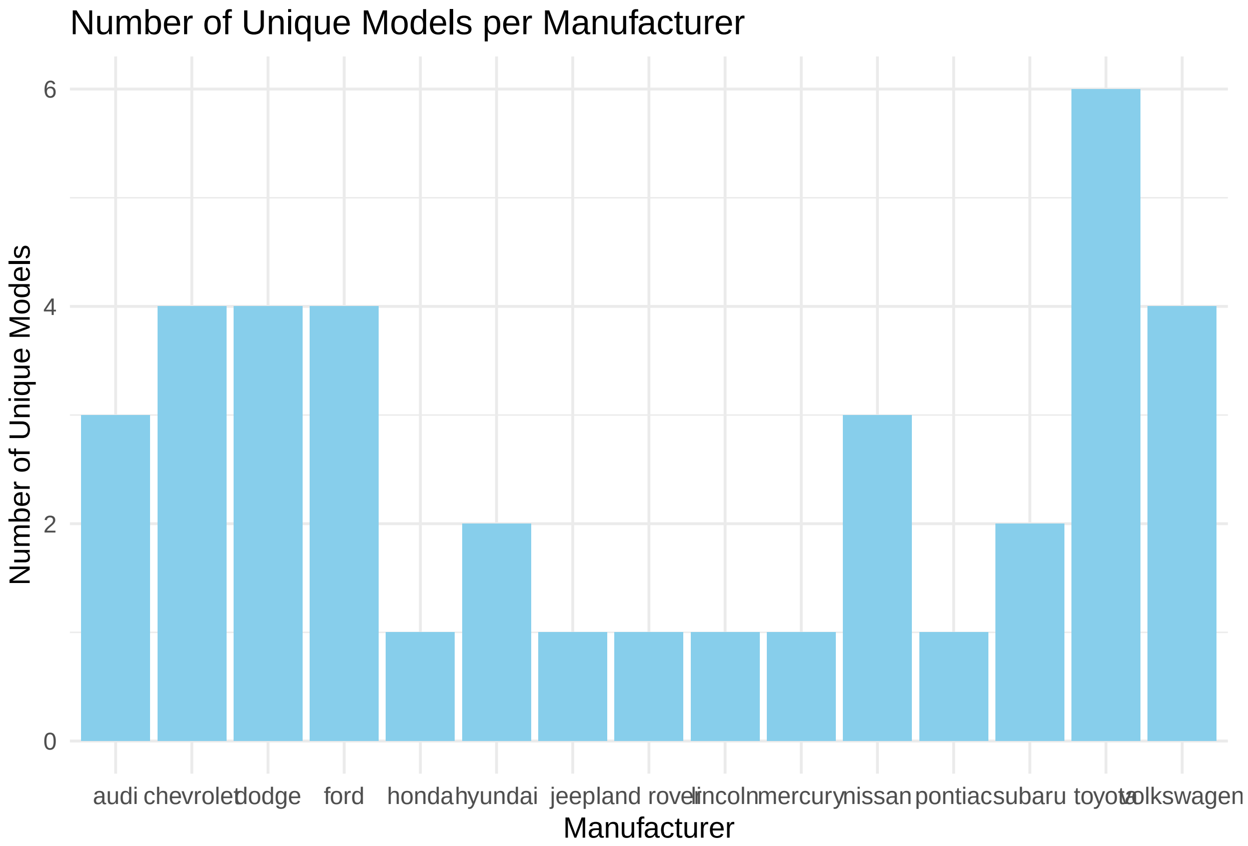
```
## # A tibble: 15 x 2
##    manufacturer unique_models
##    <chr>                <int>
##  1 audi                     3
##  2 chevrolet                4
##  3 dodge                    4
##  4 ford                     4
##  5 honda                    1
##  6 hyundai                  2
##  7 jeep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

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

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

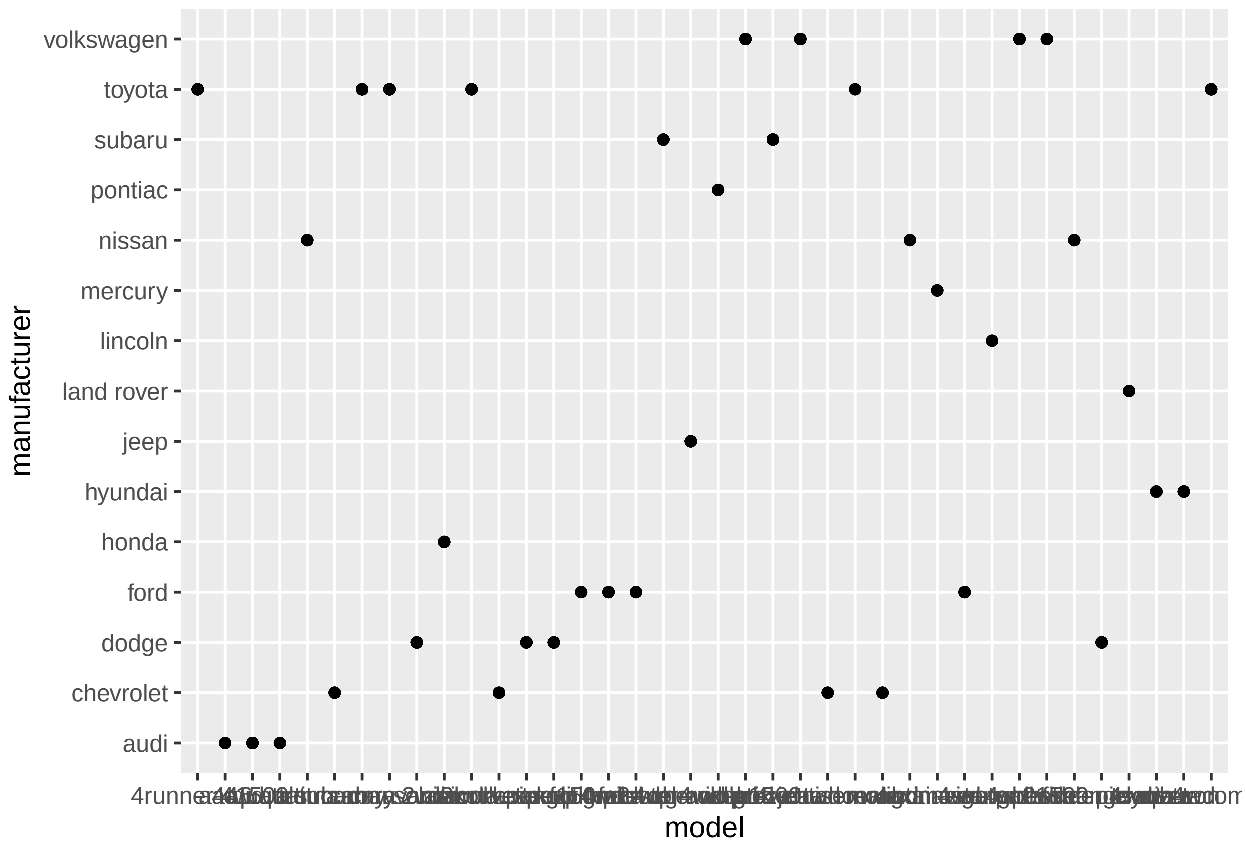
```r
ggplot(models_per_manufacturer, aes(x = manufacturer, y = unique_models)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Number of Unique Models per Manufacturer", x = "Manufacturer", y = "Number of Unique Mo
  theme_minimal()
```



2. 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?

```r
ggplot(mpg_data, aes(model, manufacturer)) + geom_point()
```

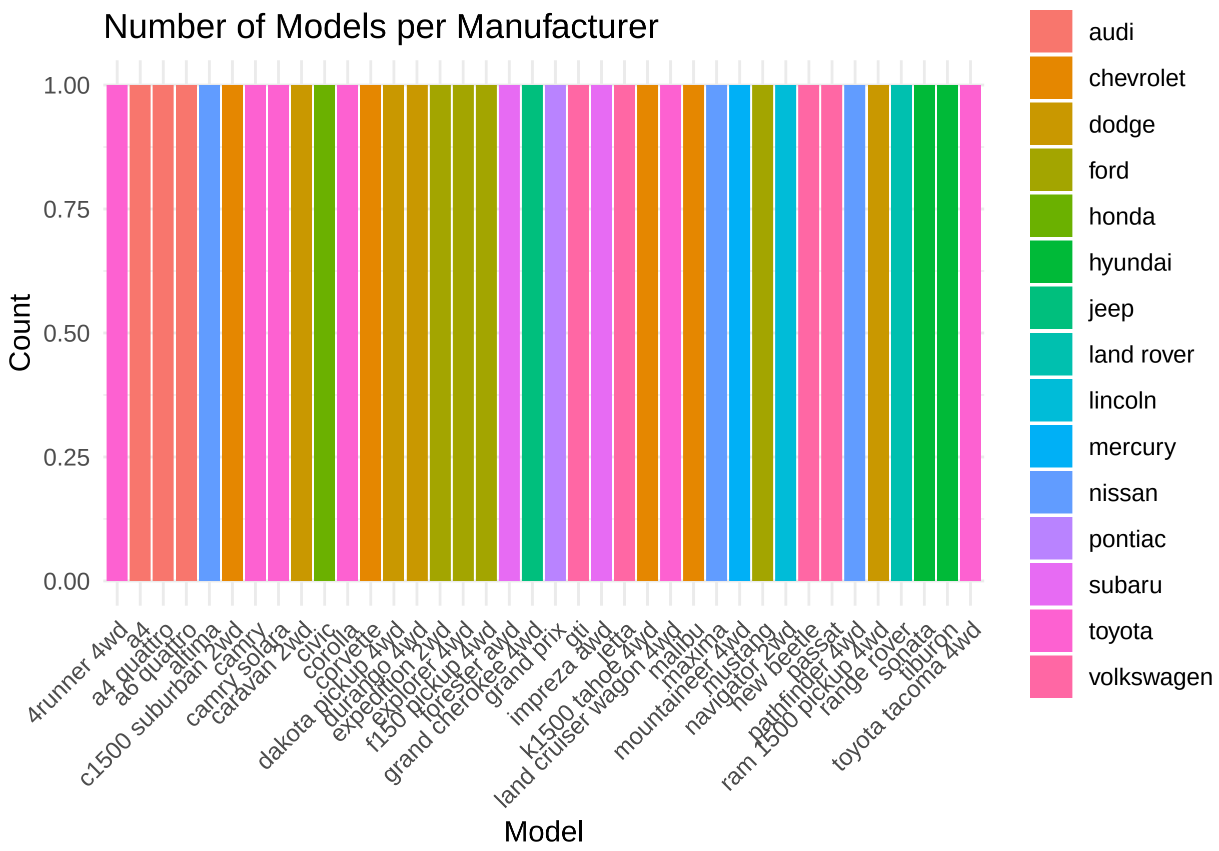


b. For you, is it useful? If not, how could you modify the data to make it more informative?

```
library(ggplot2)

model_manufacturer_count <- mpg %>%
  group_by(model, manufacturer) %>%
  summarise(count = n())
```

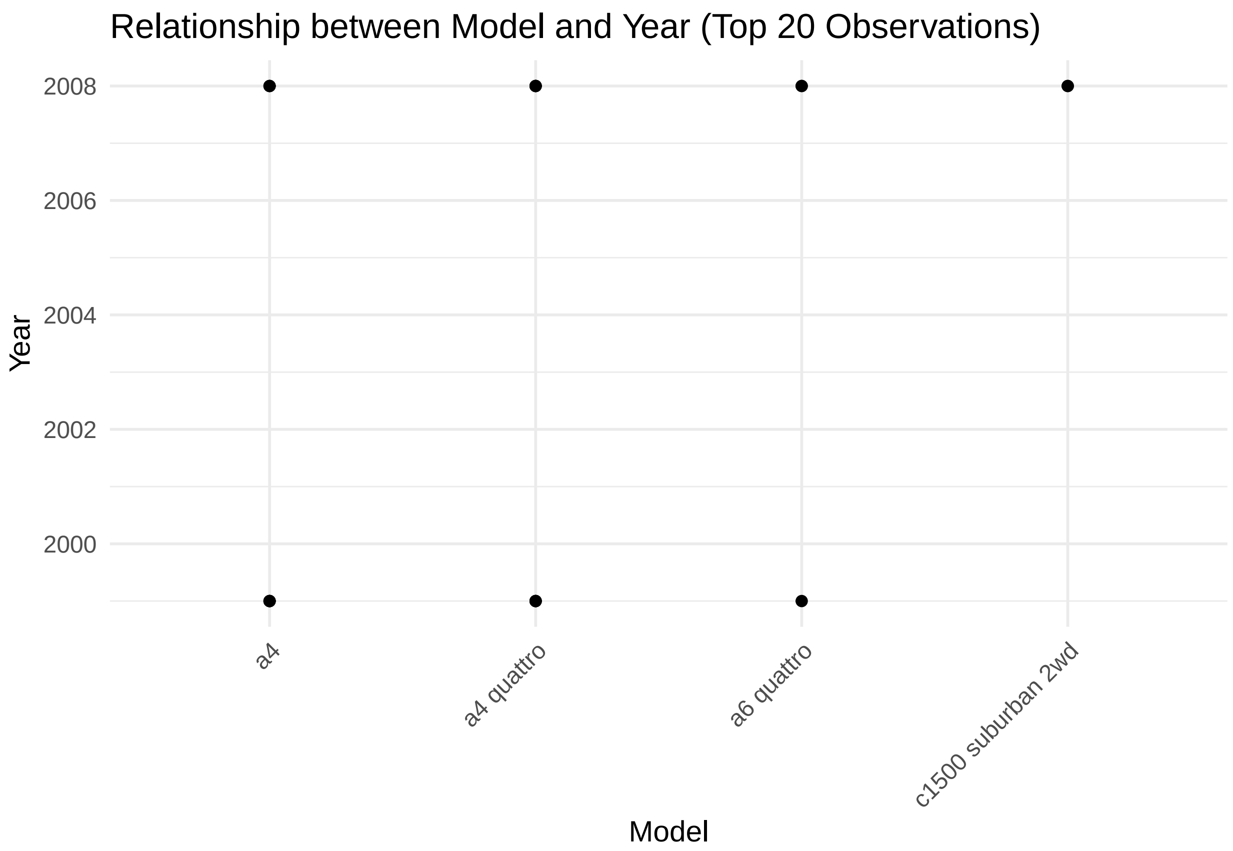
```
## ‘summarise()‘ has grouped output by ’model’. You can override using the
## ‘.groups‘ argument.
```

```
ggplot(model_manufacturer_count, aes(x = model, fill = manufacturer)) +
  geom_bar() +
  labs(title = "Number of Models per Manufacturer",
       x = "Model", y = "Count",
       fill = "Manufacturer") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



3. Plot the model and the year using ggplot(). Use only the top 20 observations. Write the codes and its results.

```
library(ggplot2)

top_20 <- head(mpg, 20)

ggplot(top_20, aes(x = model, y = year)) +
  geom_point() +
  labs(title = "Relationship between Model and Year (Top 20 Observations)",
       x = "Model", y = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



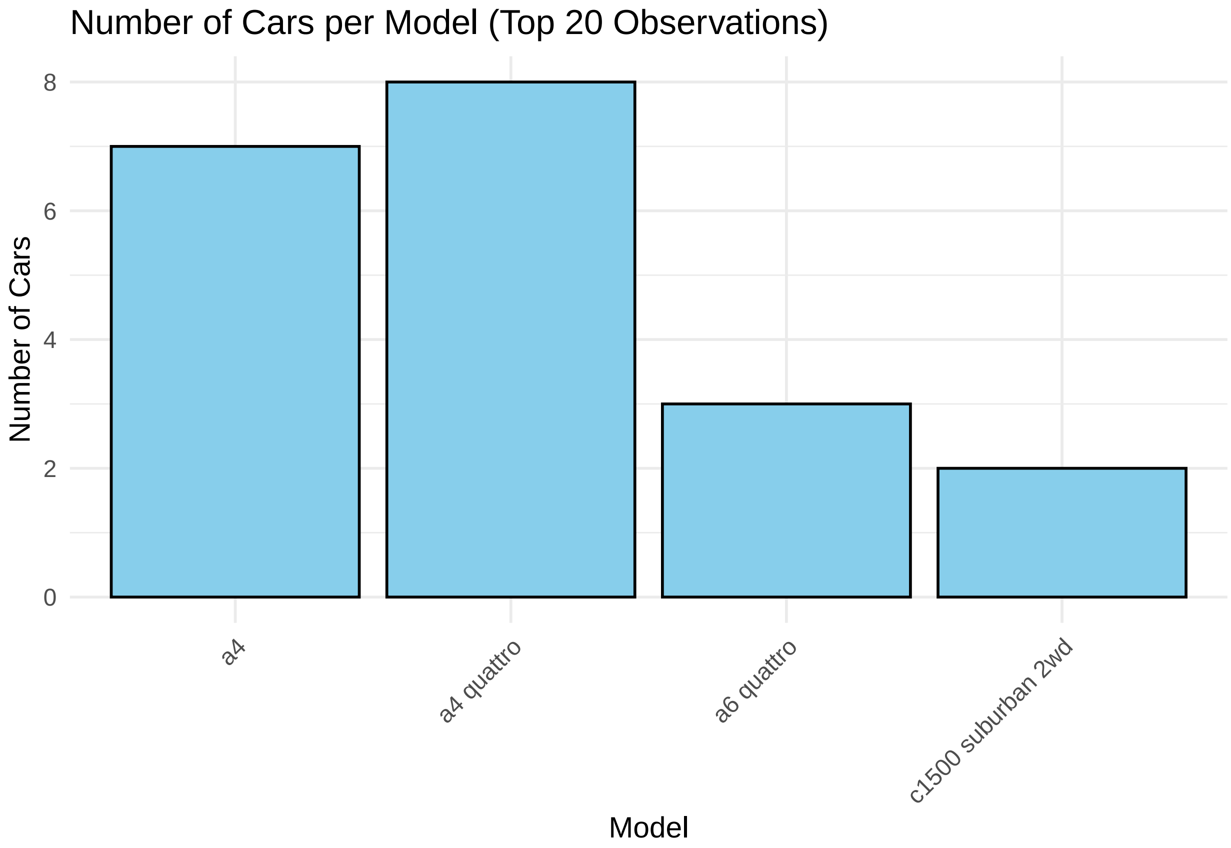
4. Using the pipe (%>%), group the model and get the number of cars per model. Show codes and its result

```r
library(dplyr)

cars_per_model <- mpg %>%
  group_by(model) %>%
  summarise(num_cars = n())

cars_per_model
```

```
## # A tibble: 38 x 2
##    model              num_cars
##    <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
## # i 28 more rows
```

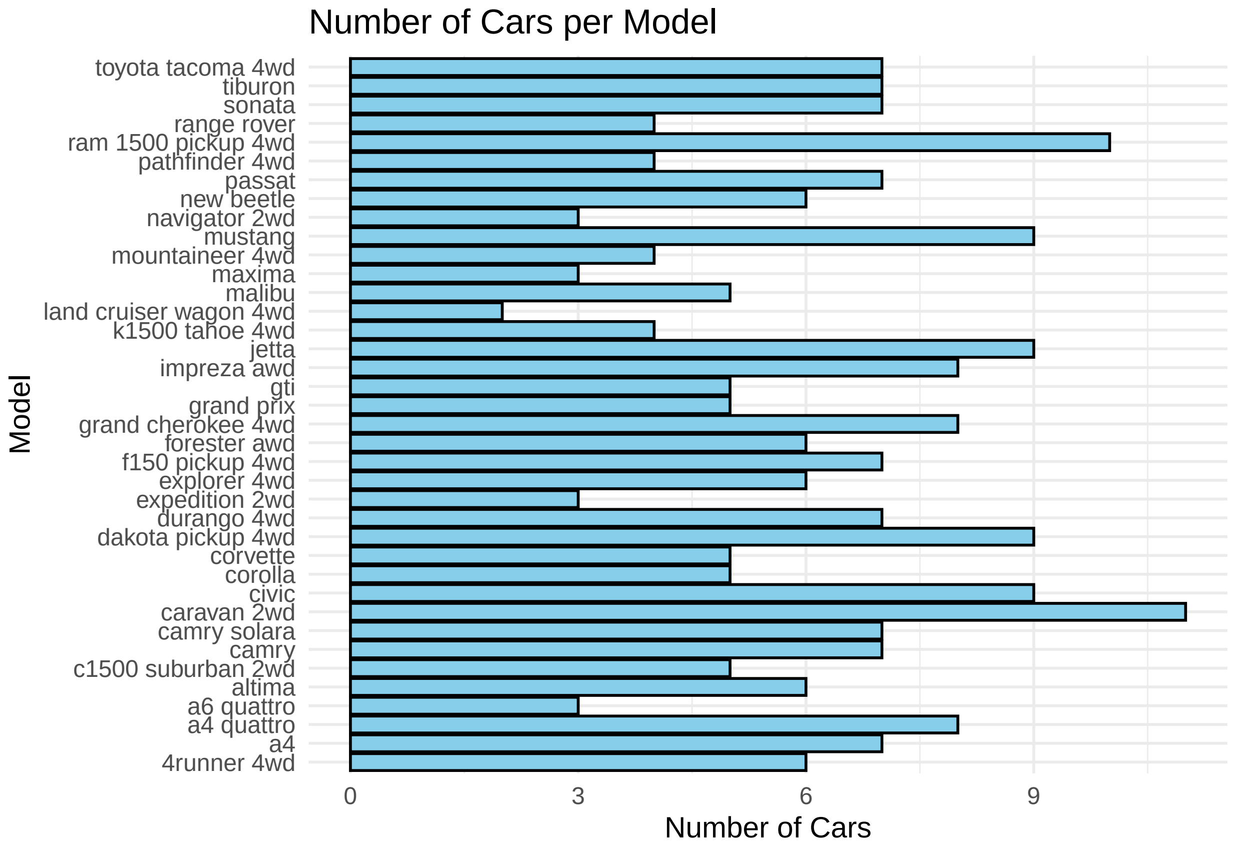
a. Plot using geom_bar() using the top 20 observations only. The graphs shoudl have a title, labels and colors. Show code and results.

```
library(ggplot2)

top_20 <- head(mpg, 20)

ggplot(top_20, aes(x = model)) +
  geom_bar(fill = "skyblue", color = "black") +
  labs(title = "Number of Cars per Model (Top 20 Observations)",
       x = "Model", y = "Number of Cars") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



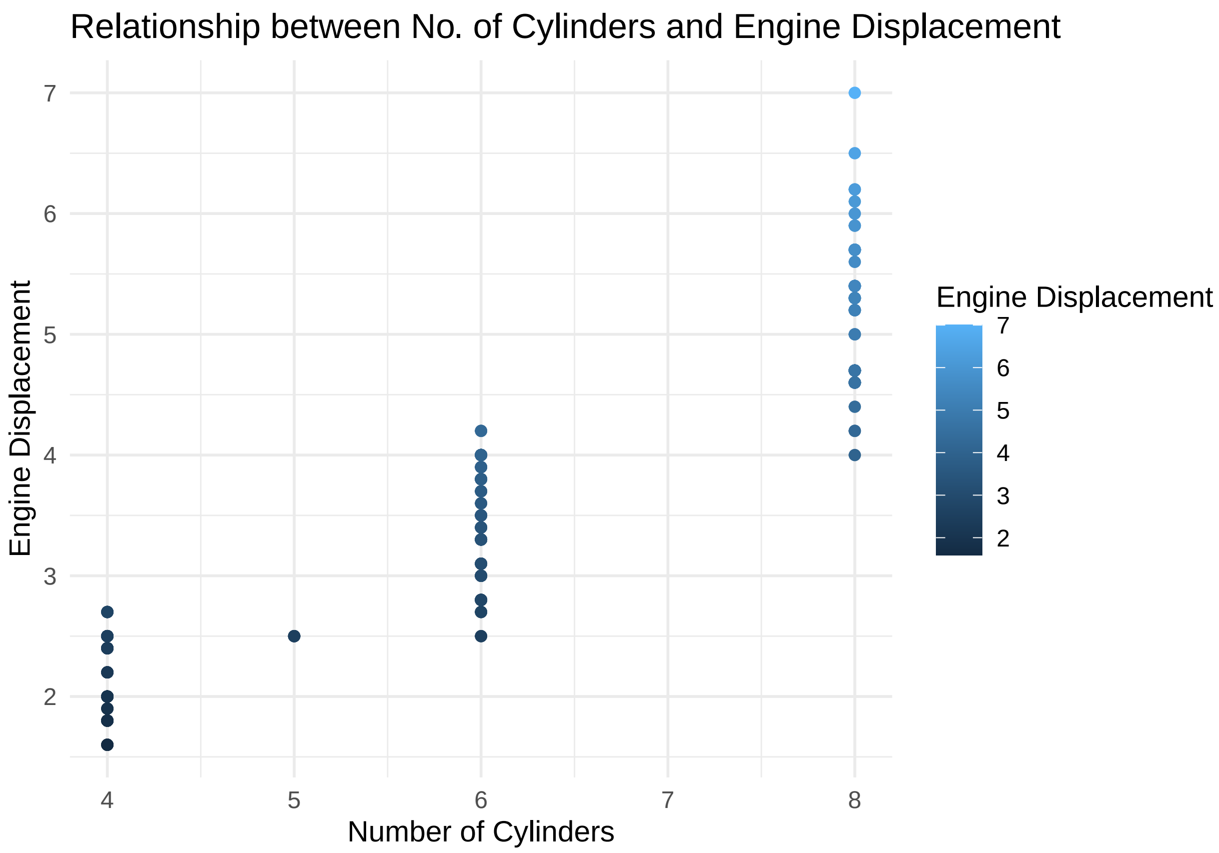
b. Plot using the geom_bar() + coord_flip() just like what is shown below. Show codes and its result.

```
library(ggplot2)

ggplot(cars_per_model, aes(x = model, y = num_cars)) +
  geom_bar(fill = "skyblue", color = "black", stat = "identity") +
  labs(title = "Number of Cars per Model",
       x = "Model", y = "Number of Cars") +
  coord_flip() +
  theme_minimal()
```



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

```r
library(ggplot2)

ggplot(mpg, aes(x = cyl, y = displ, color = displ)) +
  geom_point() +
  labs(title = "Relationship between No. of Cylinders and Engine Displacement",
       x = "Number of Cylinders", y = "Engine Displacement") +
  scale_color_continuous(name = "Engine Displacement") +
  theme_minimal()
```



a. How would you describe its relationship? Show the codes and its result.

```
#The relationship between cylinder count and engine displacement is often #positive, with larger engine
```

6. Plot the relationship between displ (engine displacement) and hwy(highway miles per gallon). Mapped it with a continuous variable you have identified in #1-c. What is its result? Why it produced such output?

7. Import the traffic.csv onto your R environment.

```
traffic <- read.csv("traffic.csv")

head(traffic)
```

```
##              DateTime Junction Vehicles          ID
## 1 2015-11-01 00:00:00        1       15 20151101001
## 2 2015-11-01 01:00:00        1       13 20151101011
## 3 2015-11-01 02:00:00        1       10 20151101021
## 4 2015-11-01 03:00:00        1        7 20151101031
## 5 2015-11-01 04:00:00        1        9 20151101041
## 6 2015-11-01 05:00:00        1        6 20151101051
```

a. How many numbers of observation does it have? What are the variables of the traffic dataset the Show your answer.

```
num_observations <- nrow(traffic)

variables <- names(traffic)

cat("Number of Observations:", num_observations, "\n")
```

```
## Number of Observations: 48120
```

```
cat("Variables:", paste(variables, collapse = ", "), "\n")
```

```
## Variables: DateTime, Junction, Vehicles, ID
```

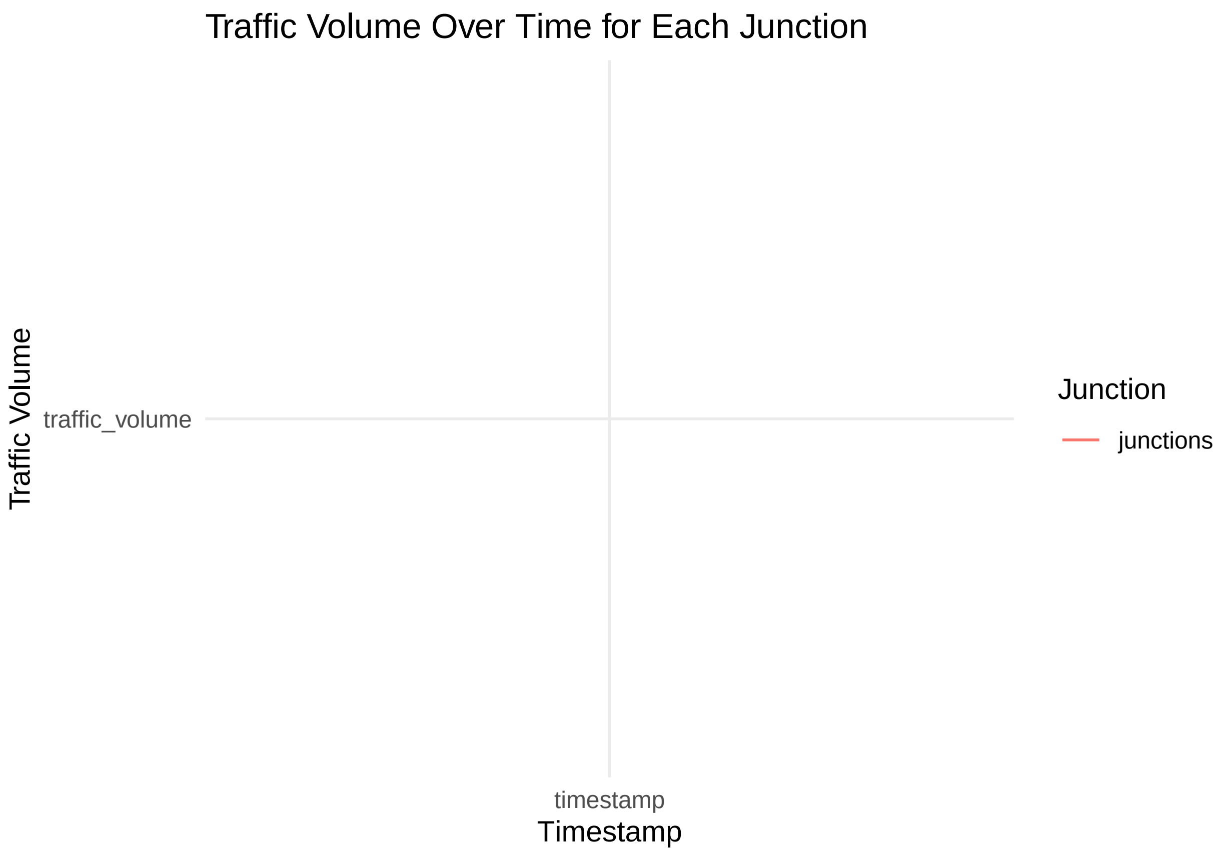
b. subset the traffic dataset into junctions. What is the R codes and its output?

```
junctions <- unique(traffic$junction)

junctions
```

```
## NULL
```

c. Plot each junction in a using geom_line(). Show your solution and output.

```
library(ggplot2)

# Plot each junction using geom_line()
ggplot(traffic, aes(x = 'timestamp', y = 'traffic_volume', color = 'junctions')) +
  geom_line() +
  labs(title = "Traffic Volume Over Time for Each Junction",
       x = "Timestamp", y = "Traffic Volume",
       color = "Junction") +
  theme_minimal()
```



7. From alexa_file.xlsx, import it to your environment

```
if (!requireNamespace("readxl", quietly = TRUE)) {
  install.packages("readxl")
}
library(readxl)
```

```
## Warning: package ’readxl’ was built under R version 4.3.2
```

```
file_path <- "alexa_file.xlsx"

alexa_data <- read_excel(file_path)

head(alexa_data)
```

```
## # A tibble: 6 x 5
##   rating date                variation          verified_reviews       feedback
##    <dbl> <dttm>              <chr>              <chr>                     <dbl>
## 1      5 2018-07-31 00:00:00 Charcoal Fabric    Love my Echo!                 1
## 2      5 2018-07-31 00:00:00 Charcoal Fabric    Loved it!                     1
## 3      4 2018-07-31 00:00:00 Walnut Finish      Sometimes while playi~        1
## 4      5 2018-07-31 00:00:00 Charcoal Fabric    I have had a lot of f~        1
## 5      5 2018-07-31 00:00:00 Charcoal Fabric    Music                         1
## 6      5 2018-07-31 00:00:00 Heather Gray Fabric I received the echo a~        1
```

a. How many observations does alexa_file has? What about the number of columns? Show your solution and answer.

```
num_observations <- nrow(alexa_data)

num_columns <- ncol(alexa_data)

cat("Number of Observations:", num_observations, "\n")
```

```
## Number of Observations: 3150
```

```
cat("Number of Columns:", num_columns, "\n")
```

```
## Number of Columns: 5
```

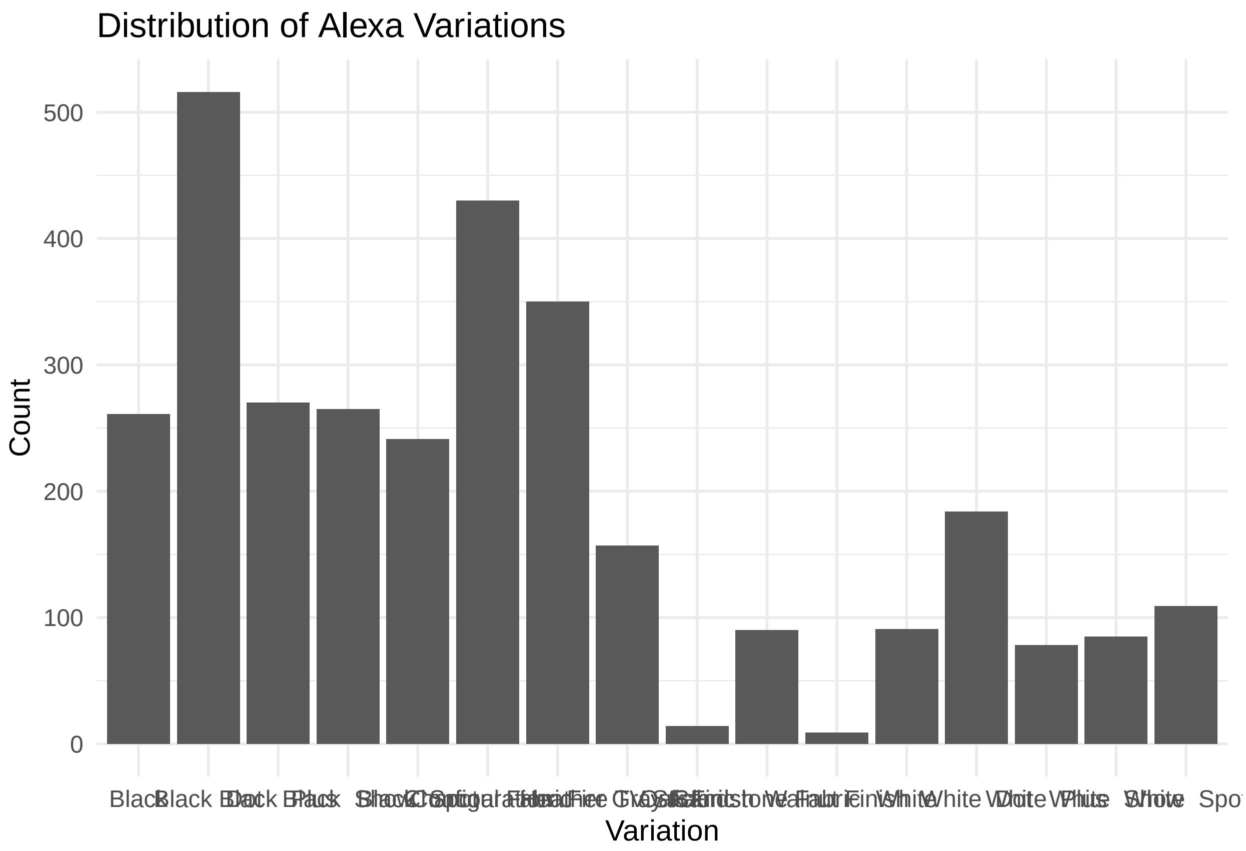
b. group the variations and get the total of each variations. Use dplyr package. Show solution and answer.

```
library(dplyr)

variation_totals <- alexa_data %>%
  group_by(variation) %>%
  summarise(total = n())

variation_totals
```

```
## # A tibble: 16 x 2
##    variation                    total
##    <chr>                        <int>
##  1 Black                          261
##  2 Black  Dot                     516
##  3 Black  Plus                    270
##  4 Black  Show                    265
##  5 Black  Spot                    241
##  6 Charcoal Fabric                430
##  7 Configuration: Fire TV Stick   350
##  8 Heather Gray Fabric            157
##  9 Oak Finish                      14
## 10 Sandstone Fabric                90
## 11 Walnut Finish                    9
## 12 White                           91
## 13 White  Dot                     184
## 14 White  Plus                     78
## 15 White  Show                     85
## 16 White  Spot                    109
```

c. Plot the variations using the ggplot() function. What did you observe? Complete the details of the graph. Show solution and answer.

```
library(ggplot2)

ggplot(alexa_data, aes(x = variation)) +
  geom_bar() +
  labs(title = "Distribution of Alexa Variations",
       x = "Variation",
       y = "Count") +
  theme_minimal()
```


Distribution of Alexa Variations
Count
Black
Variation