```r
#1.Use the dataset mpg

datmpg <- read.csv("mpg.csv")
```

```r
#1b. Which variables from mpg dataset are categorical?

categorical_vars <- c("manufacturer", "model", "trans", "drv", "fl", "class")
categorical_vars
```

```
## [1] "manufacturer" "model"        "trans"        "drv"          "fl"
## [6] "class"
```

```r
#1c. Which are continuous variables?
continuous_vars <- c("displ", "year", "cyl", "cty", "hwy")
continuous_vars
```

```
## [1] "displ" "year"  "cyl"   "cty"   "hwy"
```

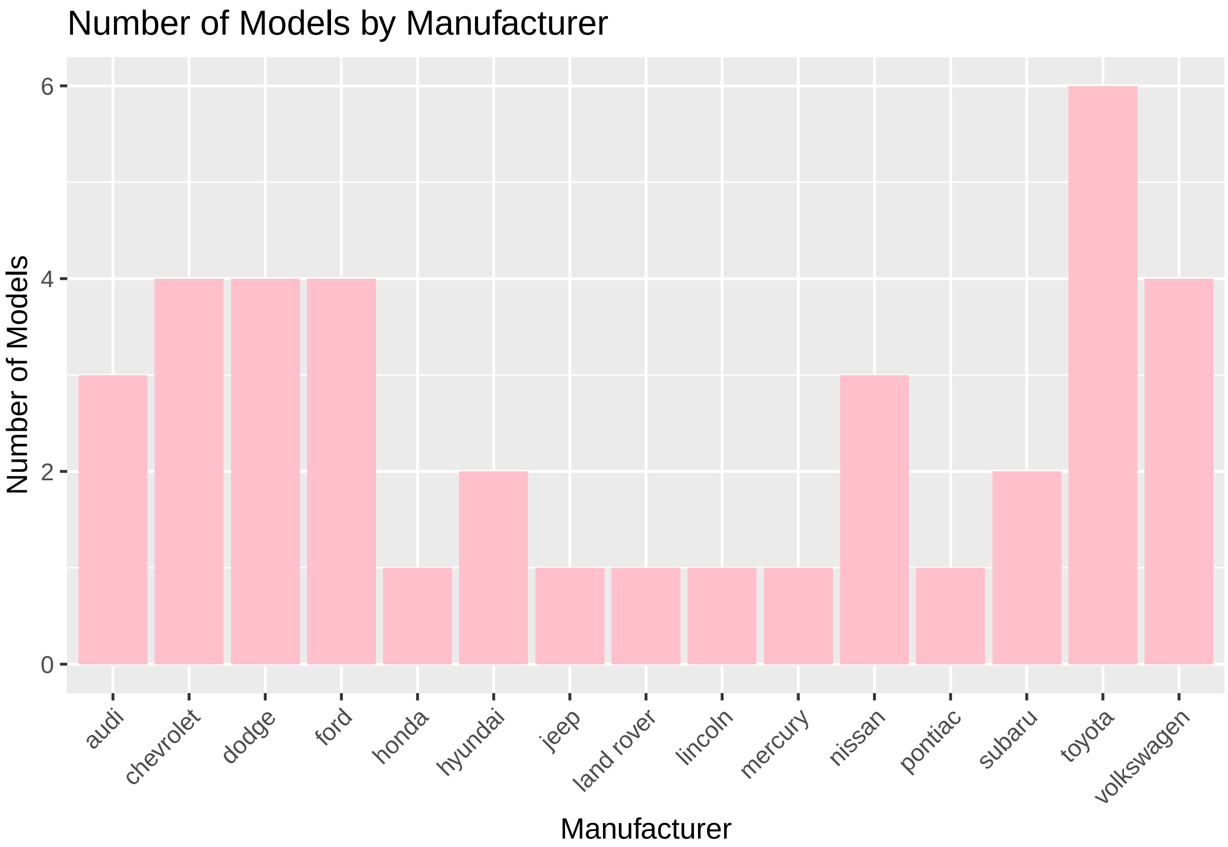
```r
#2. Which manufacturer has the most models in this data set? Which model has the most variations? Show y

#2a. Group the manufacturers and find the unique models. Show your codes and result.

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
```

```r
manufacturer_models <- datmpg %>%
  group_by(manufacturer) %>%
  summarise(num_models = n_distinct(model)) %>%
  arrange(desc(num_models))

manufacturer_models[1, ]
```

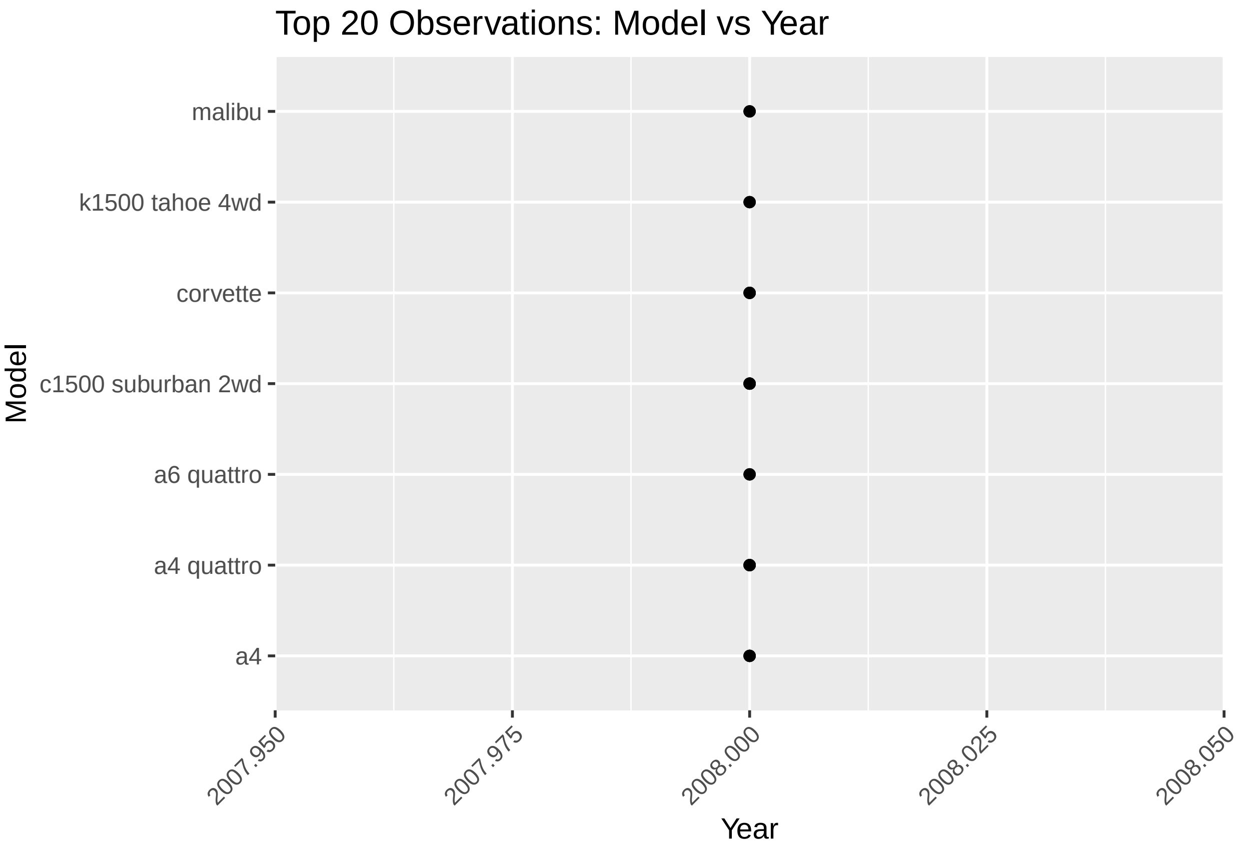
```
## # A tibble: 1 x 2
##   manufacturer num_models
##   <chr>             <int>
## 1 toyota                6
```

```r
#2b. Graph the result by using plot() and ggplot(). Write the codes and its result
library(ggplot2)

ggplot(manufacturer_models, aes(x = manufacturer, y = num_models)) +
  geom_bar(stat = "identity", fill = "pink") +
  labs(title = "Number of Models by Manufacturer", x = "Manufacturer", y = "Number of Models") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```r
#2.
#2a. What does ggplot(mpg, aes(model, manufacturer)) + geom_point() show?

#This ggplot command creates a scatter plot where each point represents a car model, positioned along t

#2b. Is it useful? If not, how could you modify the data to make it more informative?

#The plot may be useful for visualizing the distribution of models across manufacturers, but it could b
```

```r
#3.
top_20_data <- head(mpg[order(mpg$year, decreasing = TRUE), ], 20)
ggplot(top_20_data, aes(x = year, y = model)) +
  geom_point() +
  labs(title = "Top 20 Observations: Model vs Year", x = "Year", y = "Model") + theme(axis.text.x = eler
```

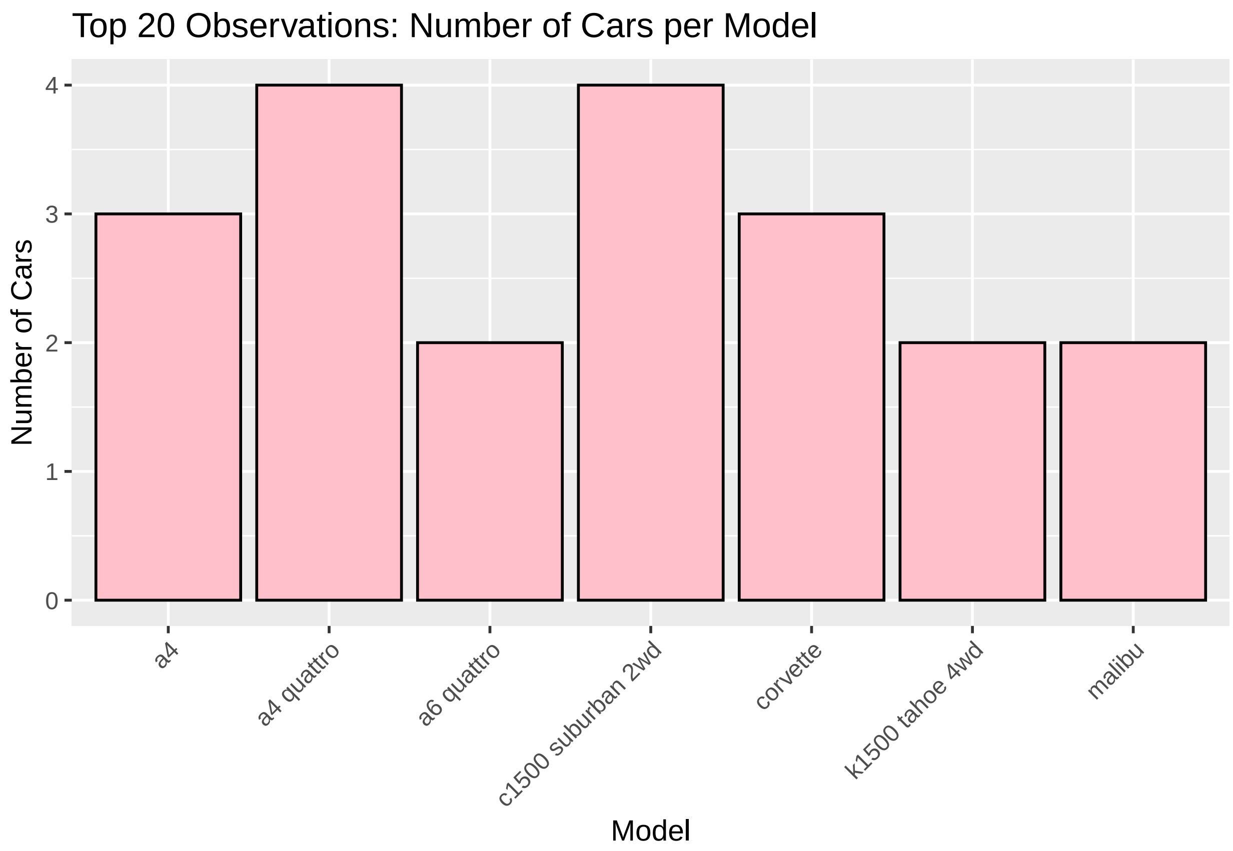


```
#4. Using the pipe (%>%), group the model and get the number of cars per model. Show codes and its resu

library(dplyr)

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

  print(cars_per_model)
```

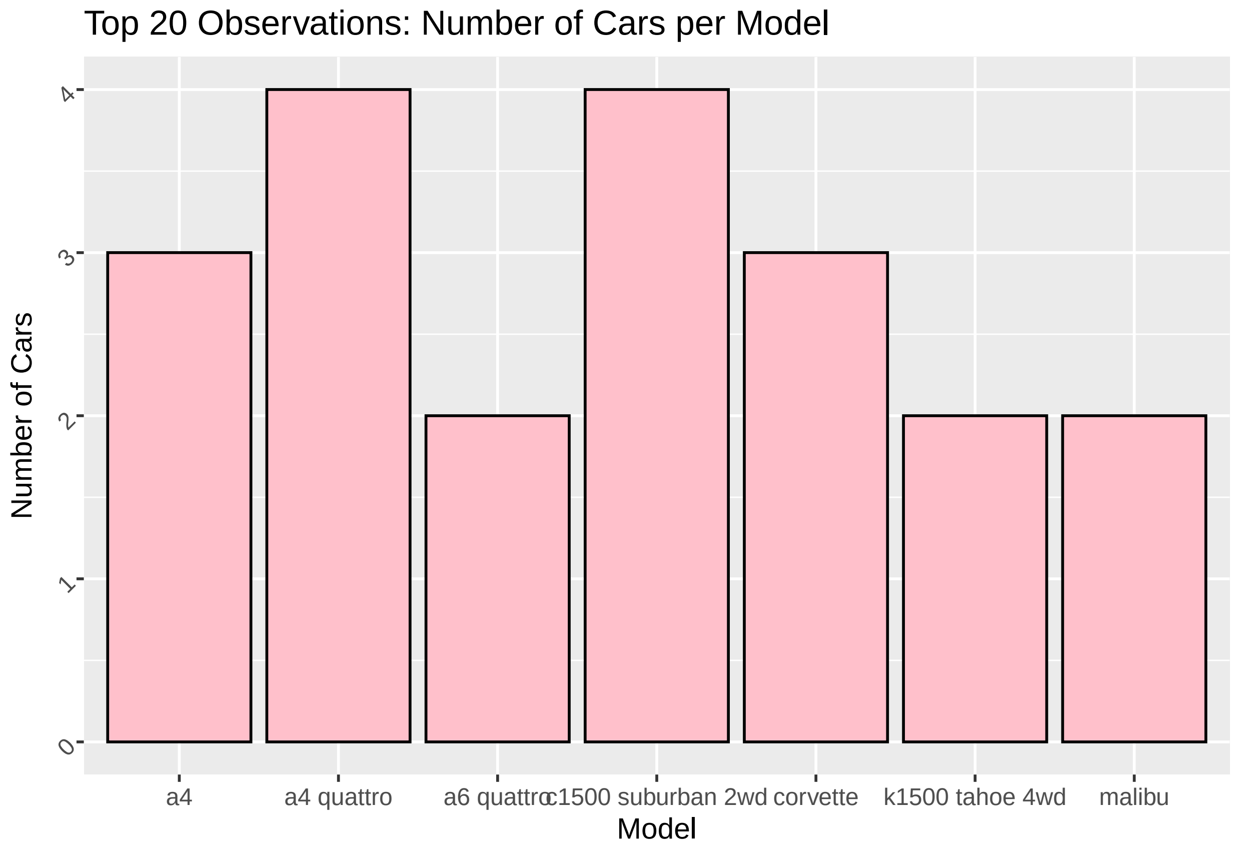
```
## # A tibble: 38 x 2
##    model              num_cars
##    <chr>                 <int>
##  1 4runner 4wd               6
##  2 a4                        7
##  3 a4 quattro                8
##  4 a6 quattro                3
##  5 altima                    6
##  6 c1500 suburban 2wd        5
##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

```
#4a. Plot using geom_bar() using the top 20 observations only. The graphs shoudl have a title, labels a

top_20_data <- head(mpg[order(mpg$year, decreasing = TRUE), ], 20)
```

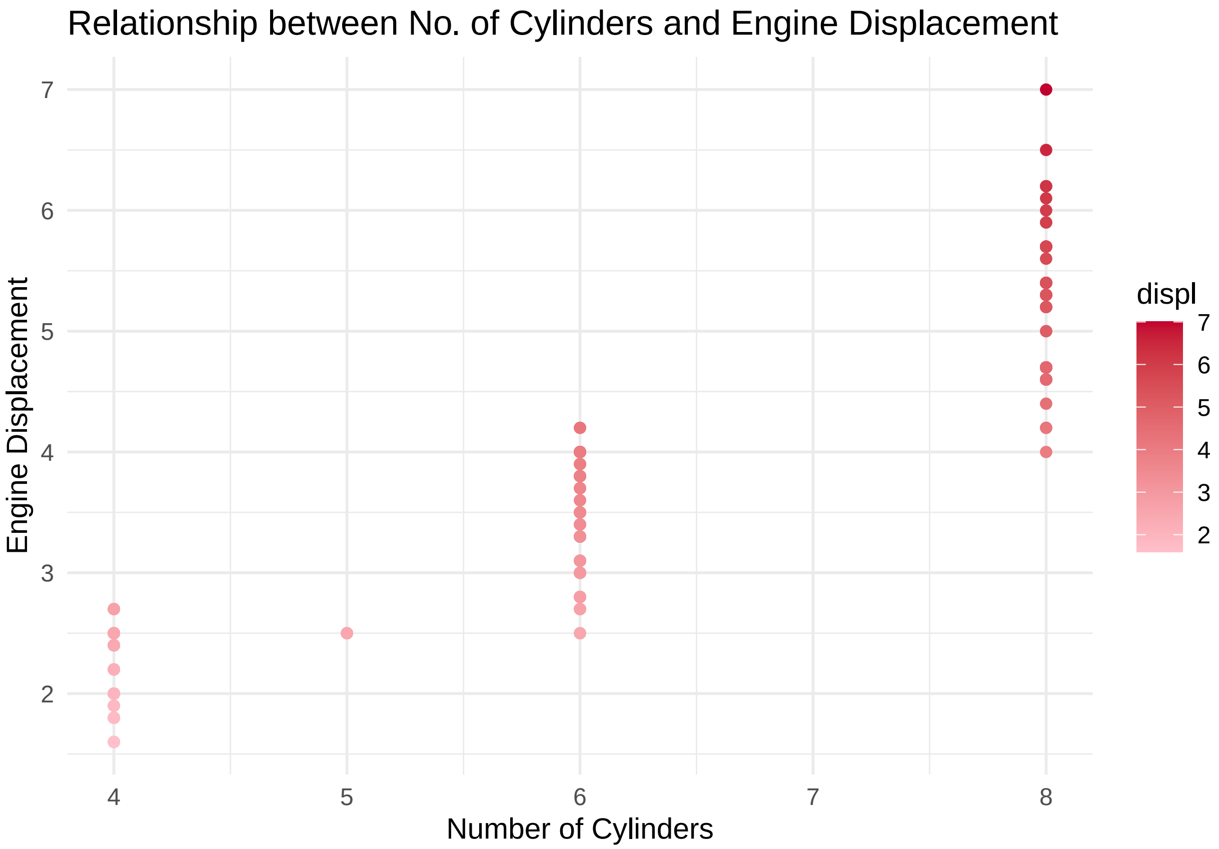
```
ggplot(top_20_data, aes(x = model)) +
  geom_bar(fill = "pink", color = "black") +
  labs(title = "Top 20 Observations: Number of Cars per Model",
       x = "Model", y = "Number of Cars") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
#4b.  Plot using the geom_bar() + coord_flip() just like what is shown below. Show codes and its result

ggplot(top_20_data, aes(y = model)) +
  geom_bar(fill = "pink", color = "black") +
  labs(title = "Top 20 Observations: Number of Cars per Model",
       x = "Number of Cars", y = "Model") +
  coord_flip() +
  theme(axis.text.y = element_text(angle = 45, hjust = 1))
```

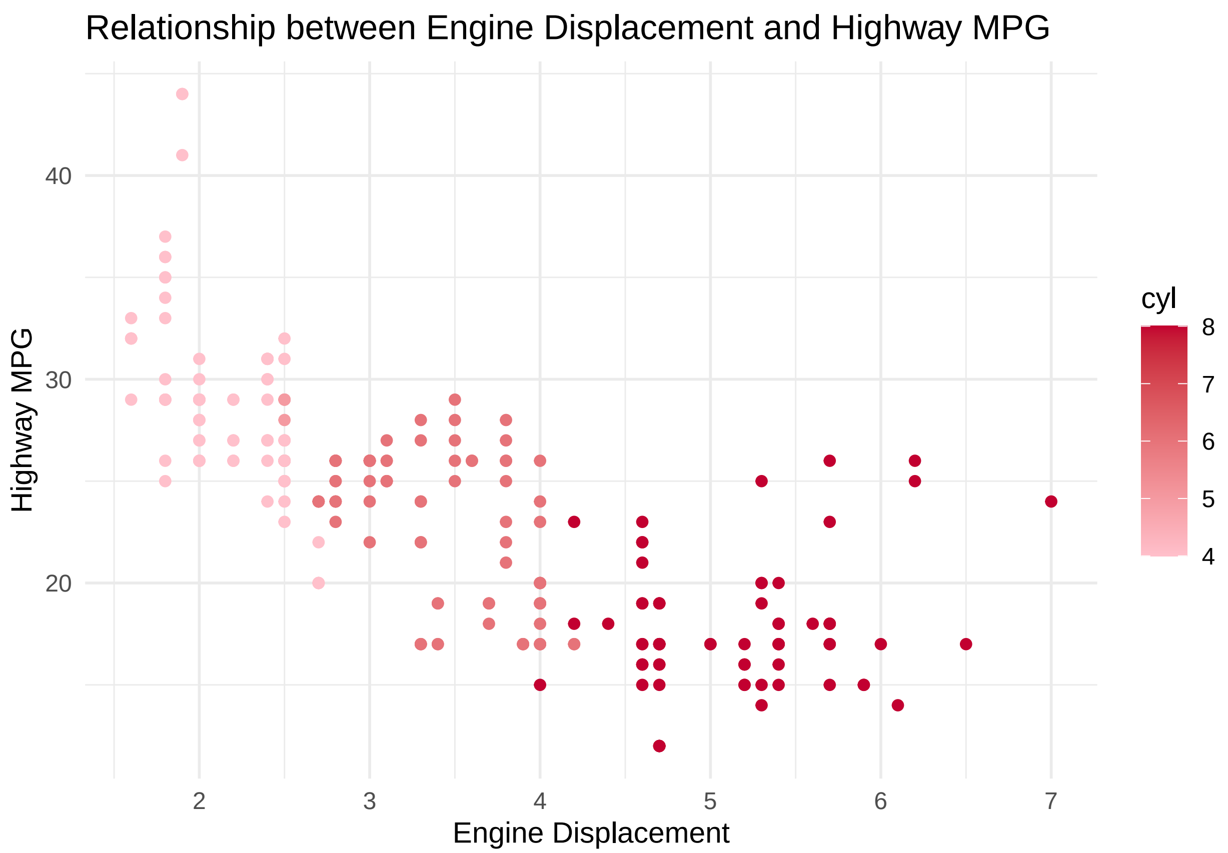


```
#5. Plot the relationship between cyl - number of cylinders and displ - engine displacement using geom_


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_gradient(low = "pink", high = "#C20030") +
  theme_minimal()
```



```
#5a. How would you describe its relationship? Show the codes and its result. The plot shows that there
```

```
#5a. How would you describe its relationship? Show the codes and its result.

ggplot(mpg, aes(x = displ, y = hwy, color = cyl)) +
  geom_point() +
  labs(title = "Relationship between Engine Displacement and Highway MPG",
       x = "Engine Displacement", y = "Highway MPG") +
  scale_color_gradient(low = "pink", high = "#C20030") +
  theme_minimal()
```



```
#6. Import the traffic.csv onto your R environment.
traffic <- read.csv("traffic.csv")
```

```
#6a. How many numbers of observation does it have? What are the variables of the traffic dataset the Sh

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

number_obs <- nrow(trafficdata)
var_traffic <- ncol(trafficdata)
varnames <- names(trafficdata)

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

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

```
cat("Number of Variables:", var_traffic, "\n")
```

```
## Number of Variables: 4
```

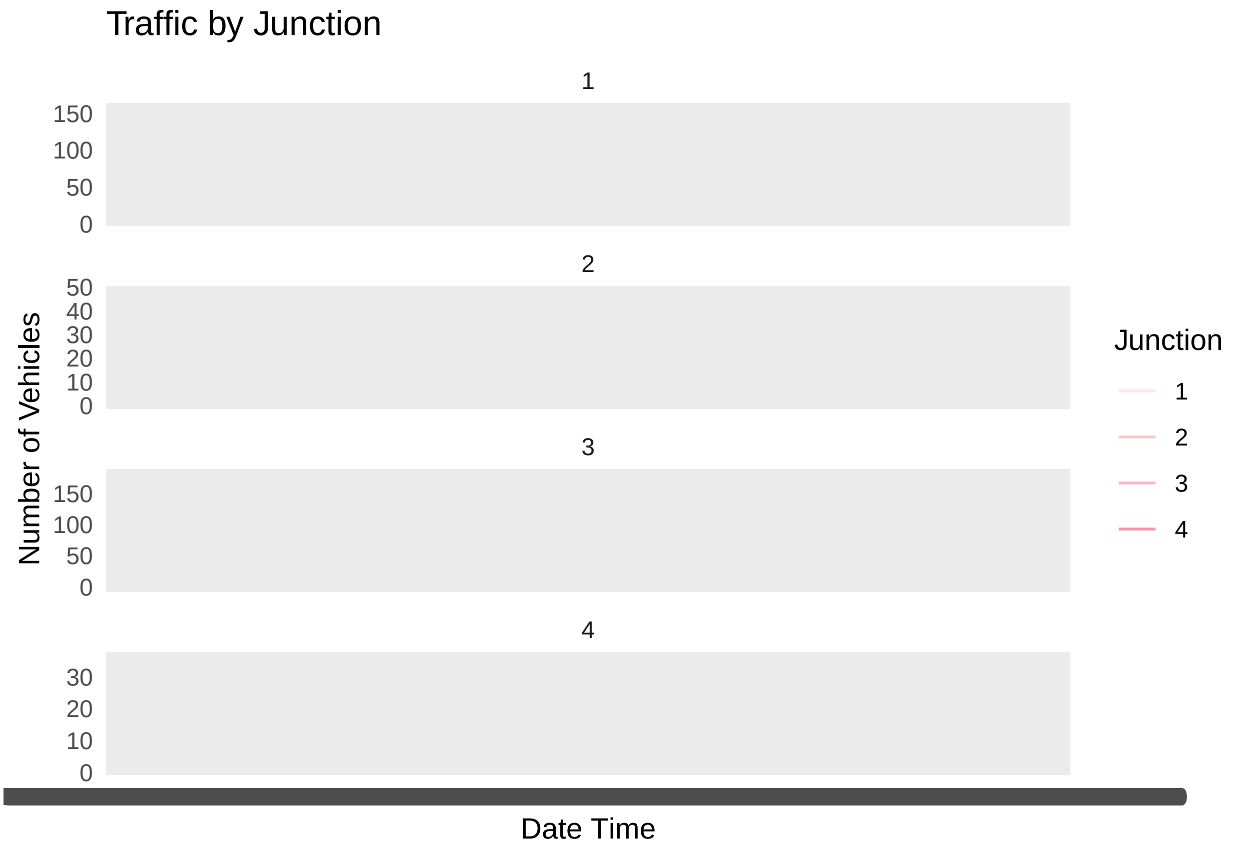
```
cat("Variables:", paste(varnames, collapse = ", "), "\n")
```

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

```
#6b.subset the traffic dataset into junctions. What is the R codes and its output?

junction_traffic <- c(1,2,3,4)


junction_subset <- trafficdata[trafficdata$Junction %in% junction_traffic, ]
```

```
#6c.Plot each junction in a using geom_line(). Show your solution and output.
library(ggplot2)

ggplot(trafficdata, aes(x = DateTime, y = Vehicles, color = as.factor(Junction))) +
  geom_line() +
  scale_color_manual(values = c("#ffe5ec", "#ffc2d1", "#ffb3c6", "#ff8fab")) +
  labs(title = "Traffic by Junction",
       x = "Date Time",
       y = "Number of Vehicles") +
  theme_minimal() +
  facet_wrap(~Junction, scales = "free_y", ncol = 1) +
  guides(color = guide_legend(title = "Junction"))
```

```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```



```
#7.From alexa_file.xlsx, import it to your environment
library(readxl)

alexafile <- read_excel("alexa_file.xlsx")
```

```
#7a.How many observations does alexa_file has? What about the number of columns? Show your solution and

alexa_obser <- nrow(alexafile)

alexacol_obser <- ncol(alexafile)

cat("The number of observations on alexa is:",alexa_obser,"\n")
```

```
## The number of observations on alexa is: 3150
```

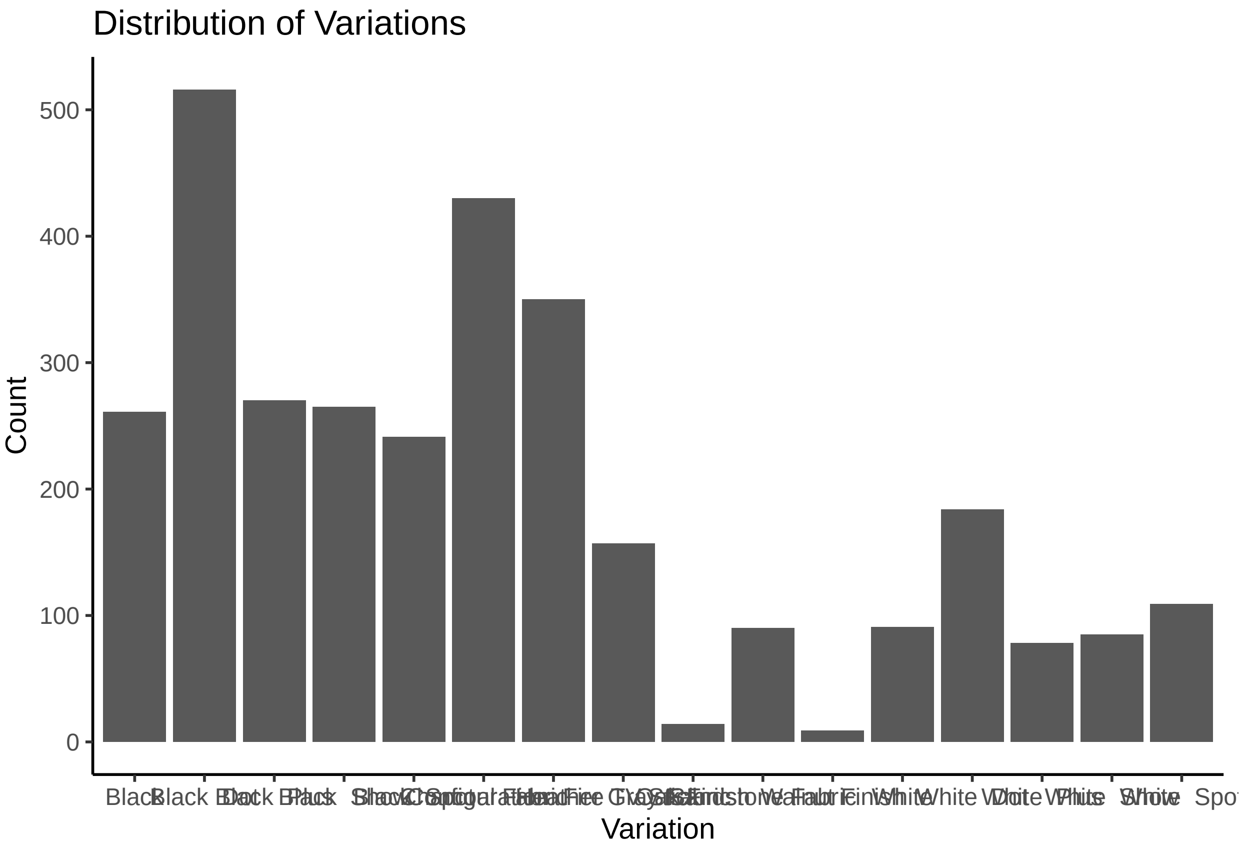
```
cat("The number of coloumns on alexa is:",alexacol_obser,"\n")
```

```
## The number of coloumns on alexa is: 5
```

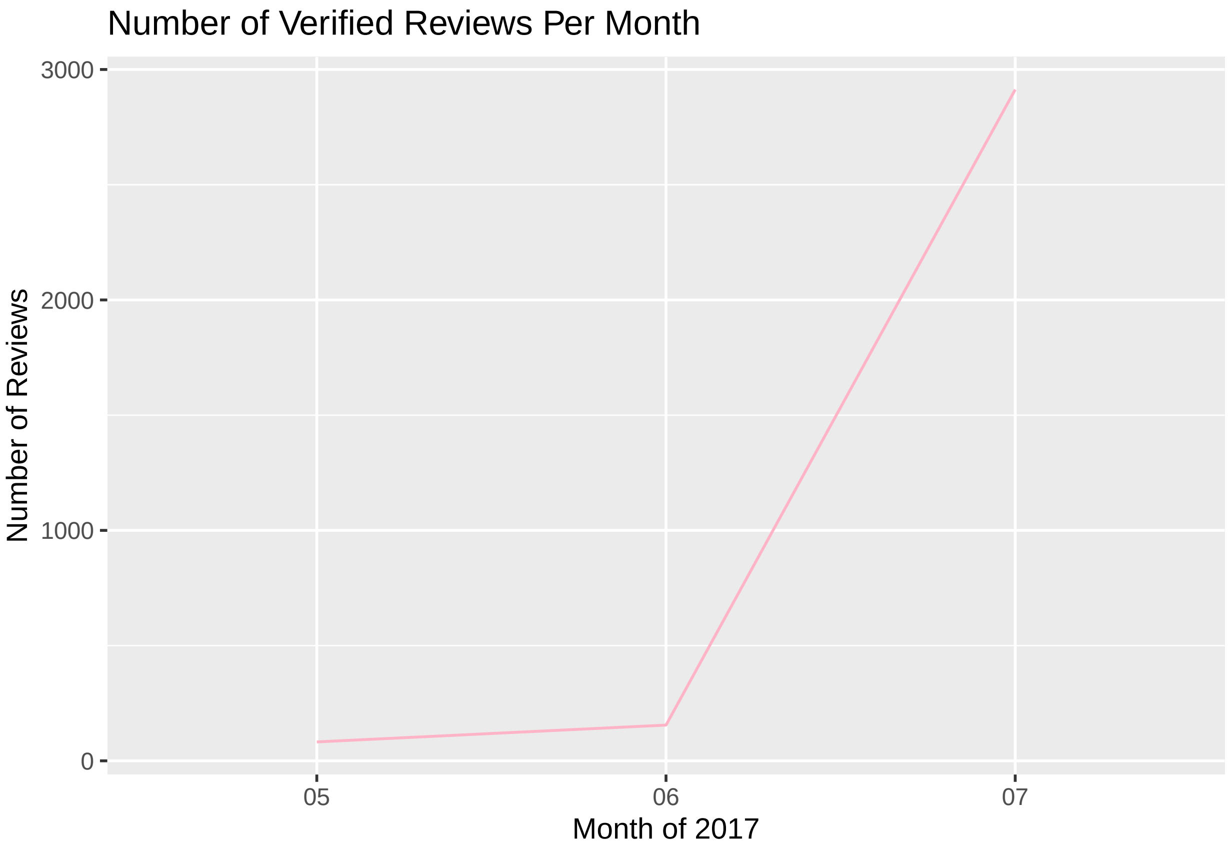
```
#7b.group the variations and get the total of each variations. Use dplyr package. Show solution and ans

library(dplyr)
groupvariations <- alexafile%>%
  group_by(variation)%>%
  summarise(totalcount_ = n())

groupvariations
```

```
## # A tibble: 16 x 2
##    variation                    totalcount_
##    <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
```

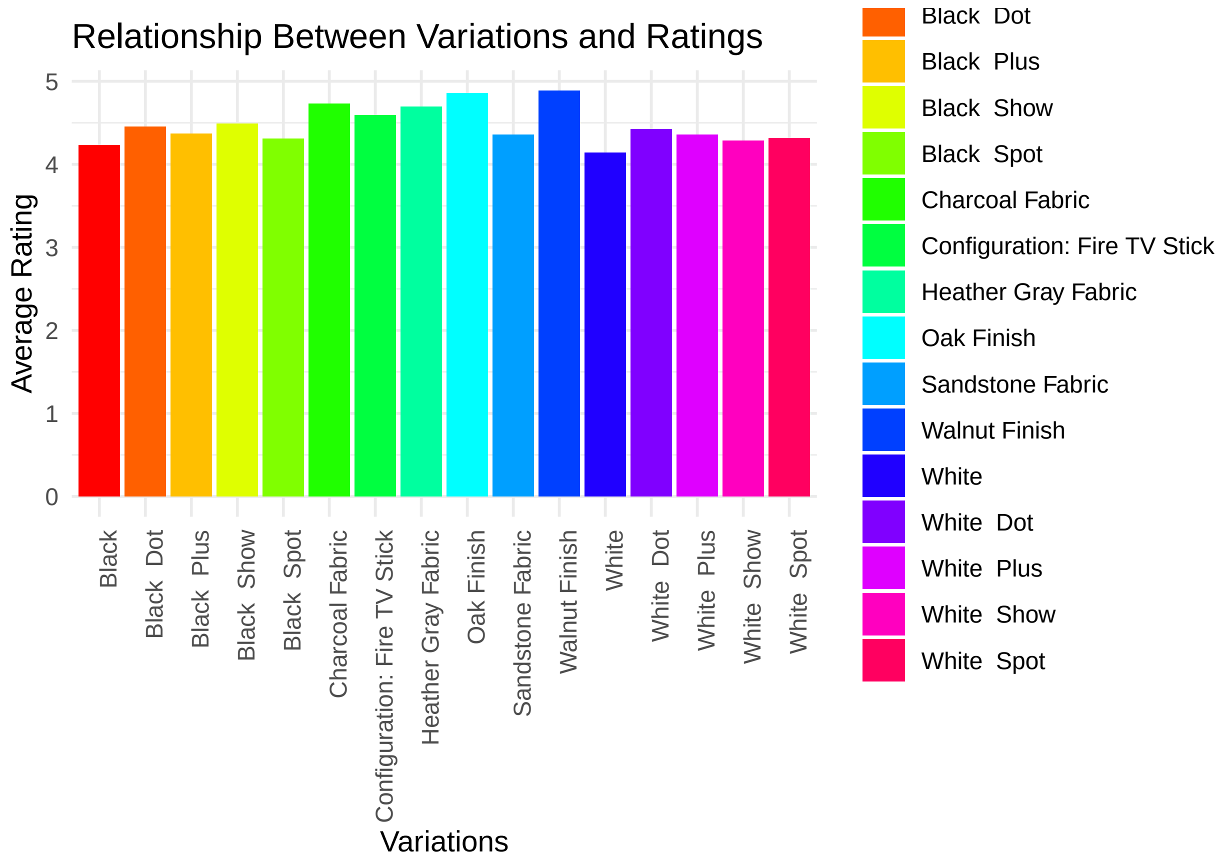
```
#7c. Plot the variations using the ggplot() function. What did you observe? Complete the details of the

library(ggplot2)

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



```
#7d. Plot a geom_line() with the date and the number of verified reviews. Complete the details of the g

alexafile$date <- as.Date(alexafile$date)

alexafile$month <- format(alexafile$date, "%m")

monthscount <- alexafile %>%
  group_by(month) %>%
  summarise(num_reviews = n())


monthlyrev <- table(monthscount)

ggplot(monthscount, aes(x = month, y = num_reviews, group = 1)) +
  geom_line(color = "#ffb3c6") +
  labs(title = "Number of Verified Reviews Per Month",
       x = "Month of 2017", y = "Number of Reviews")
```



```
#7e. Get the relationship of variations and ratings. Which variations got the most h ghest in rating? P

library(dplyr)

ggplot(alexafile, aes(x = variation, y = rating, fill = variation)) +
  geom_bar(stat = "summary", fun = "mean", position = "dodge") +
  scale_fill_manual(values = rainbow(n = length(unique(alexafile$variation)))) +
  labs(title = "Relationship Between Variations and Ratings",
       x = "Variations",
       y = "Average Rating") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
var_ratings <-  alexafile %>%
  group_by(variation)%>%
  summarise(avearage_rating = mean(rating, na.rm = TRUE))

max_rating <- max(var_ratings$average_rating, na.rm = TRUE)
```

```
## Warning: Unknown or uninitialised column: `average_rating`.
```

```
## Warning in max(var_ratings$average_rating, na.rm = TRUE): no non-missing
## arguments to max; returning -Inf
```

```
highrate <- alexafile %>%
  filter(rating == max_rating)
print(highrate)
```

```
## # A tibble: 0 x 6
## # i 6 variables: rating <dbl>, date <date>, variation <chr>,
## #   verified_reviews <chr>, feedback <dbl>, month <chr>
```