# RWorksheet_Cahutay#4c

Mark Anton Cahutay

2024-11-03

1. Use the dataset mpg

```
#A. Solution on how to import a csv file into the environment.
library(ggplot2)

mpg_data <- read.csv("mpg.csv")
str(mpg_data)
```

```
## 'data.frame':    234 obs. of  11 variables:
##  $ 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" ...
```

```
#B. The categorical variables from the mpg dataset are manufacture, model, year, cyl, trans, drv, fl, a

#C. The continuous variables from mpg are displ, cty, and hwy.
```

2.1: The manufacturer with the most models and the model with the most variations.

```
#A. Code for grouping the manufacturers and to look for their unique models.
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
```

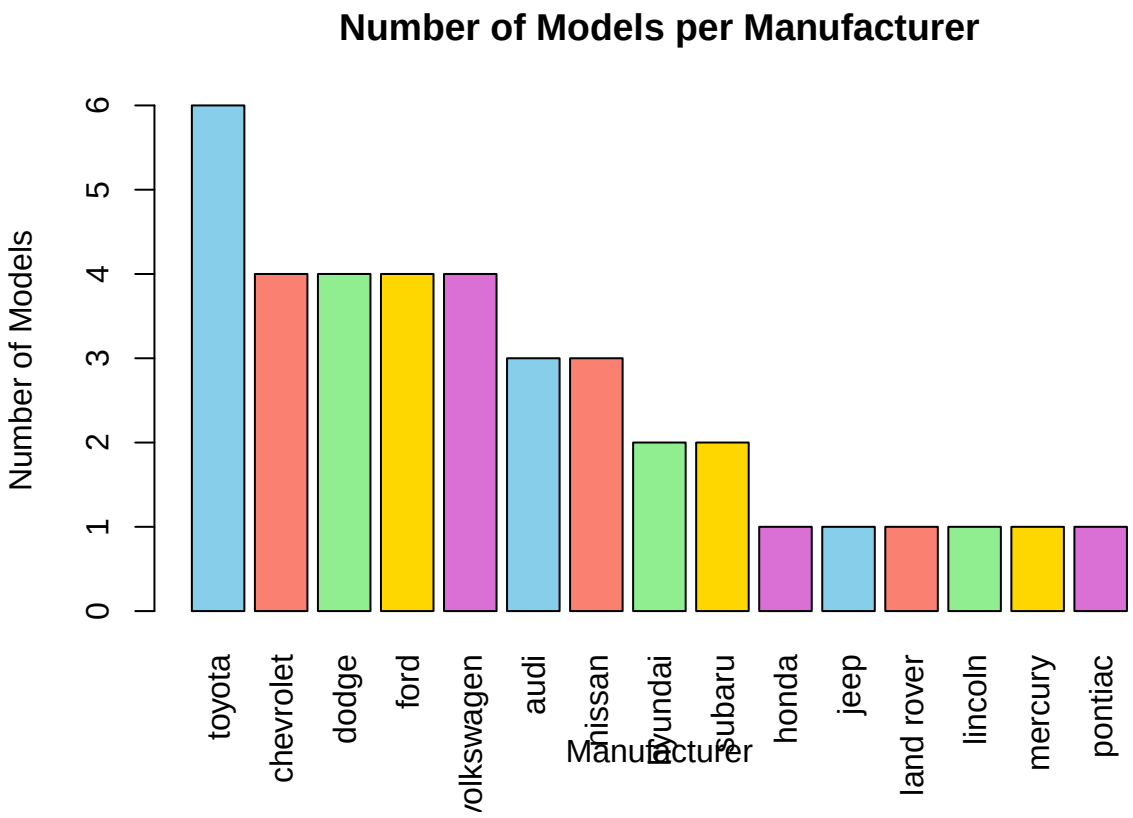
```r
manufacturer_model <- mpg %>%
  group_by(manufacturer) %>%
  summarize(model_num = n_distinct(model)) %>%
  arrange(desc(model_num))

manufacturer_model
```

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

```r
variations_num <- table(mpg$model)
variations_num [variations_num == max(variations_num)]
```

```
## caravan 2wd
##          11
```

```r
#B. Graph the result using plot() and ggplot().

#below is the barplot from plot() function
manufacturer_data <- setNames(
  manufacturer_model$model_num,
  manufacturer_model$manufacturer
  )

barplot(manufacturer_data,
        main = "Number of Models per Manufacturer",
        xlab = "Manufacturer",
        ylab = "Number of Models",
        col = c("skyblue", "salmon", "lightgreen", "gold", "orchid"),
        las = 3)
```

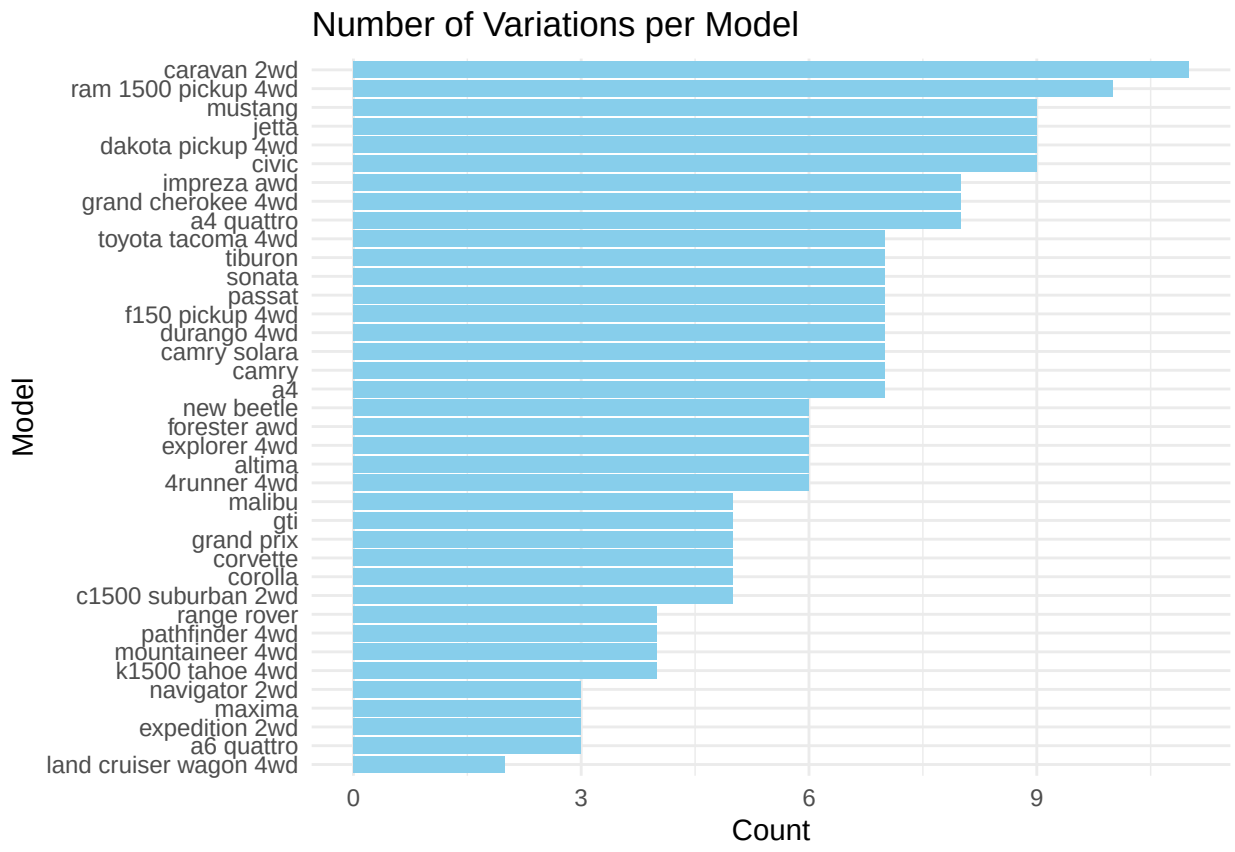



```
#below is the barplot from the ggplot().
variations_num <- mpg %>%
  group_by(model) %>%
  summarize(count = n()) %>%
  arrange(desc(count))

variations_num
```

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

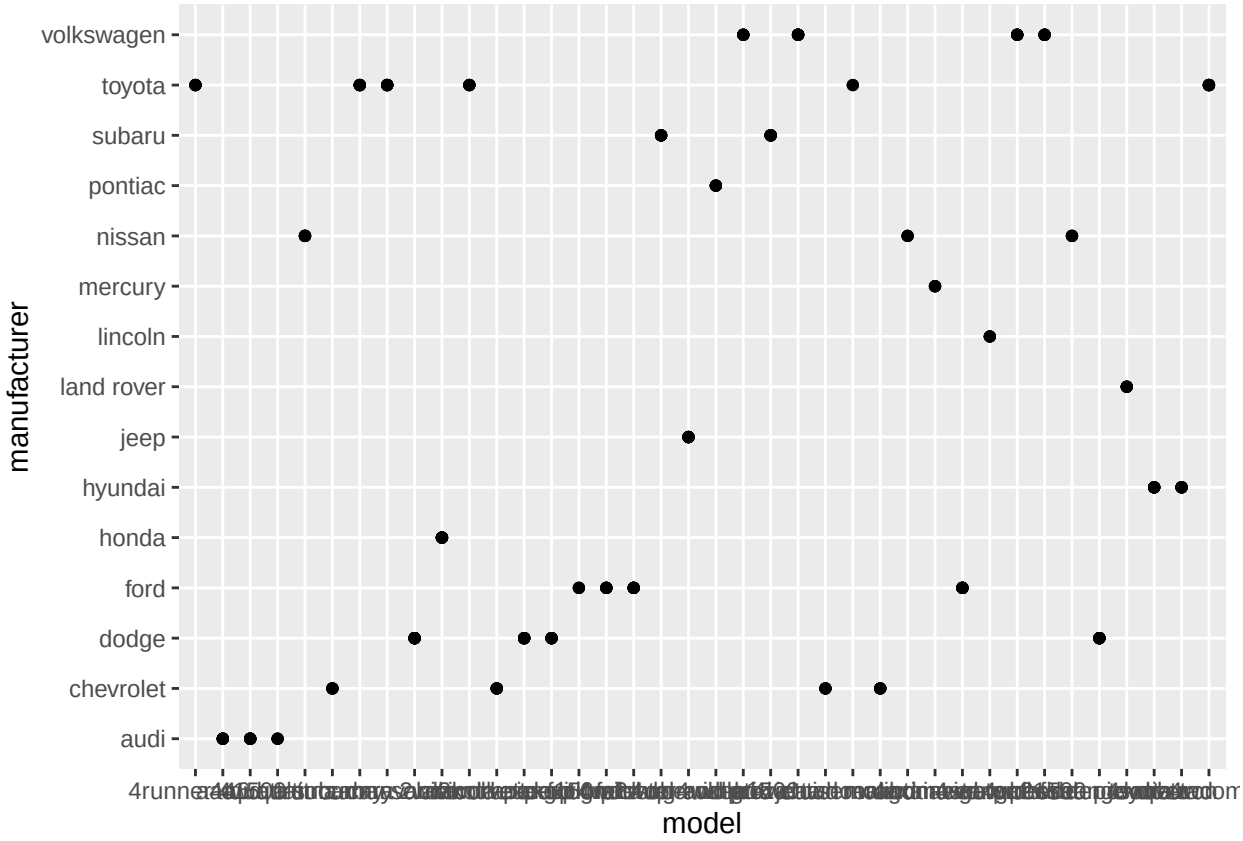
```
ggplot(variations_num,
       aes(x = reorder(model, count), y = count)) +
```

```
      geom_bar(stat = "identity", fill = "skyblue") +  coord_flip() +
      labs(title = "Number of Variations per Model", x = "Model", y = "Count") +
      theme_minimal()
```




2.2: Relationship of the model and manufacturer.

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

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

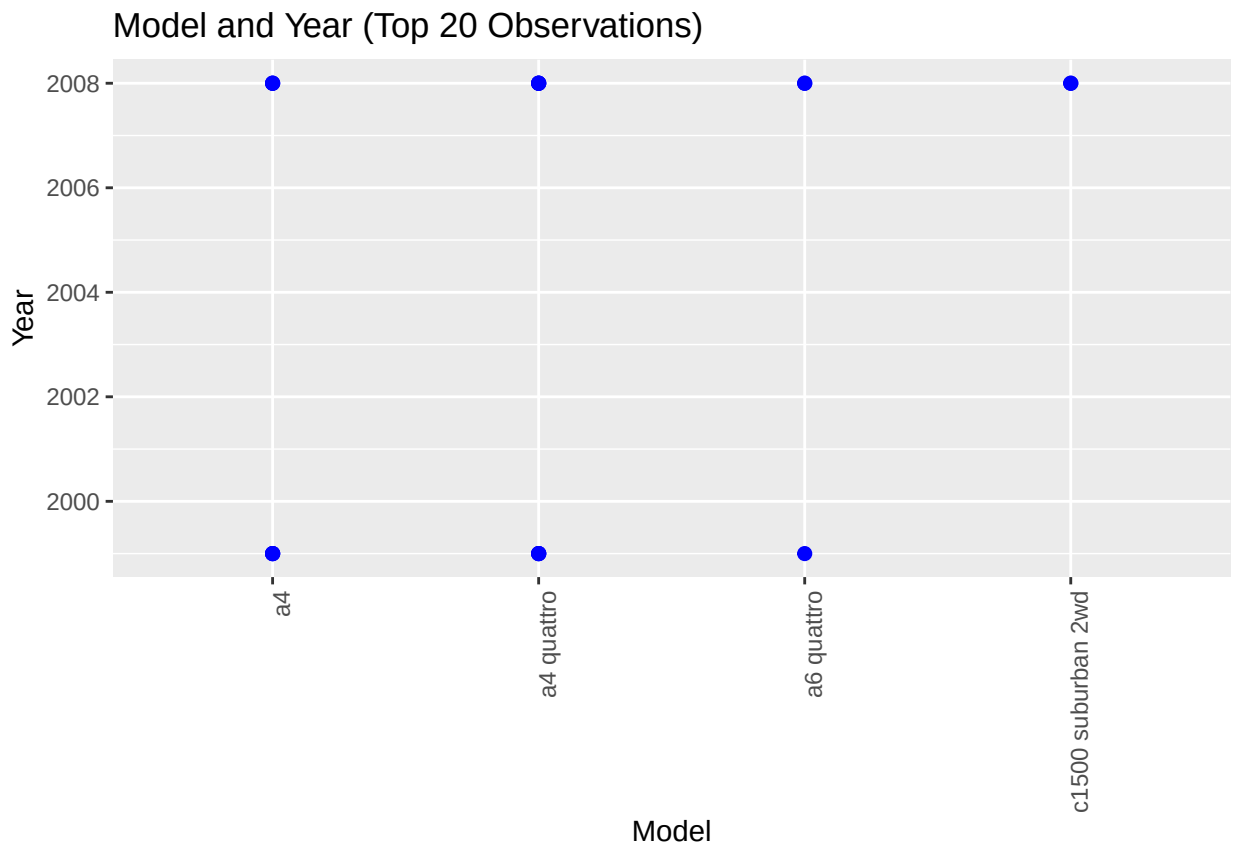



```
#This code displays a scatter plot of models and manufacturers.
```

B. For you, is it useful? If not, how could you modify the data to make it more informative? - For me, the scatter plot isn't useful, it is difficult to interpret because of the messy labels below it. To make it useful, I would change it to a bar plot because the data being used here are both categorical variable and bar plot works better with it. Setting the labels much clearer and readable would also be considered here.

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

```
obs20 <- mpg[1:20, ]

ggplot(obs20,
       aes(x = model, y = year)) +
       geom_point(color = "blue", size = 2) +
       labs(
          title = "Model and Year (Top 20 Observations)",
          x = "Model",
          y = "Year") +
       theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

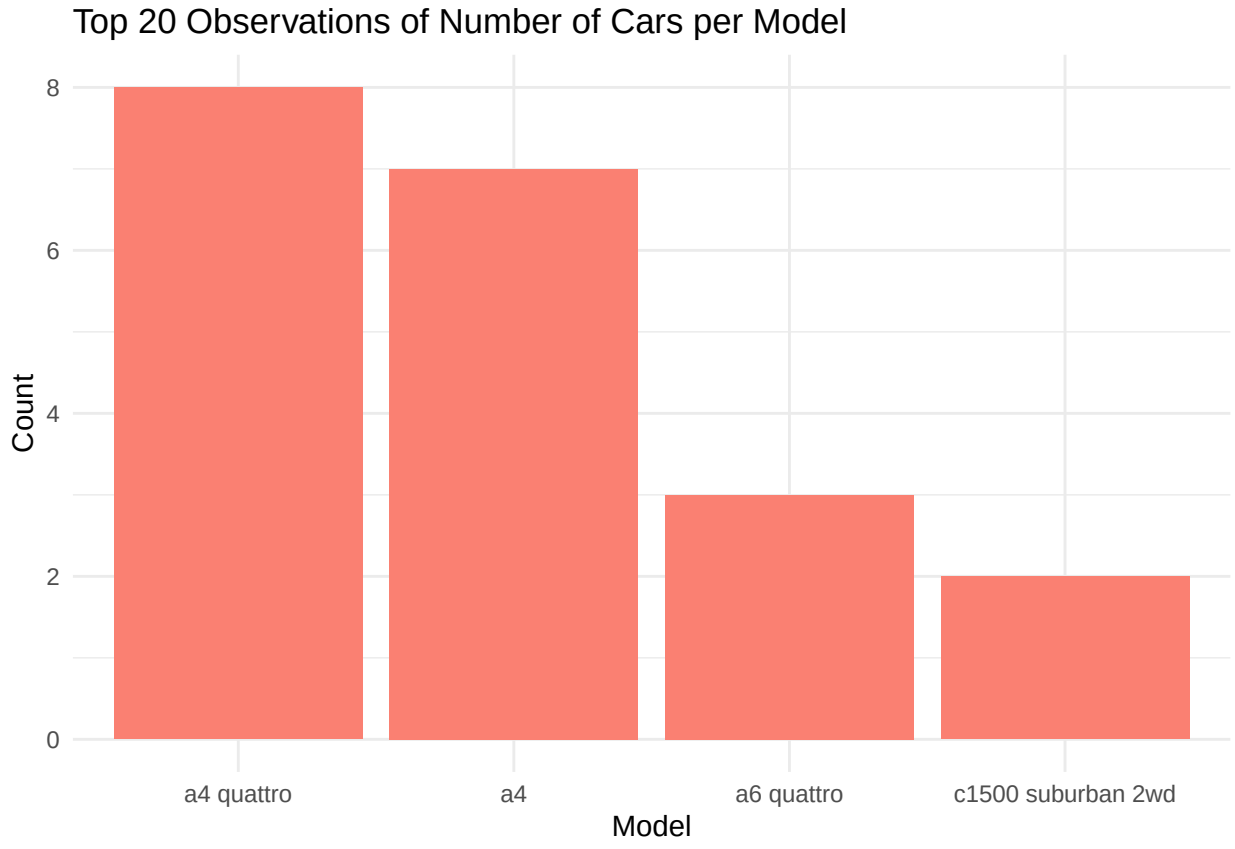



4. Using the pipe (%>%) to group the model and getting the number of cars per model.

```
library(dplyr)

carNum <- mpg %>%
  group_by(model) %>%
  summarize(count = n())

carNum
```

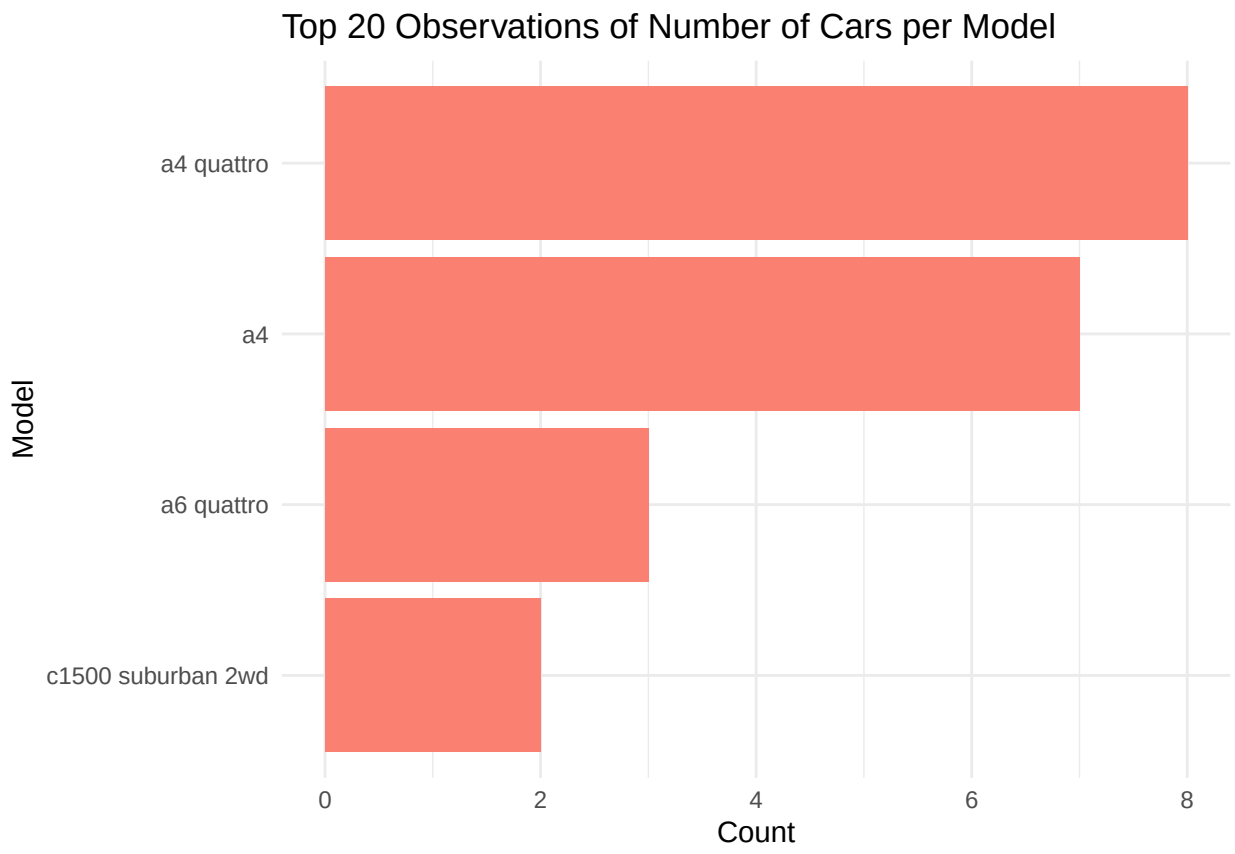
```
## # A tibble: 38 x 2
##    model              count
##    <chr>              <int>
##  1 4runner 4wd            6
##  2 a4                     7
##  3 a4 quattro             8
##  4 a6 quattro             3
##  5 altima                 6
##  6 c1500 suburban 2wd     5
##  7 camry               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

```r
#A. Plot using geom_bar() using the top 20 observations only.

carNum20 <- obs20 %>%
  group_by(model) %>%
  summarise(count = n())

ggplot(
  carNum20,
  aes(x = reorder(model, -count), y = count)
) +
  geom_bar(stat = "identity", fill = "salmon") +
  labs(
    title = "Top 20 Observations of Number of Cars per Model",
    x = "Model",
    y = "Count"
  ) +
  theme_minimal()
```



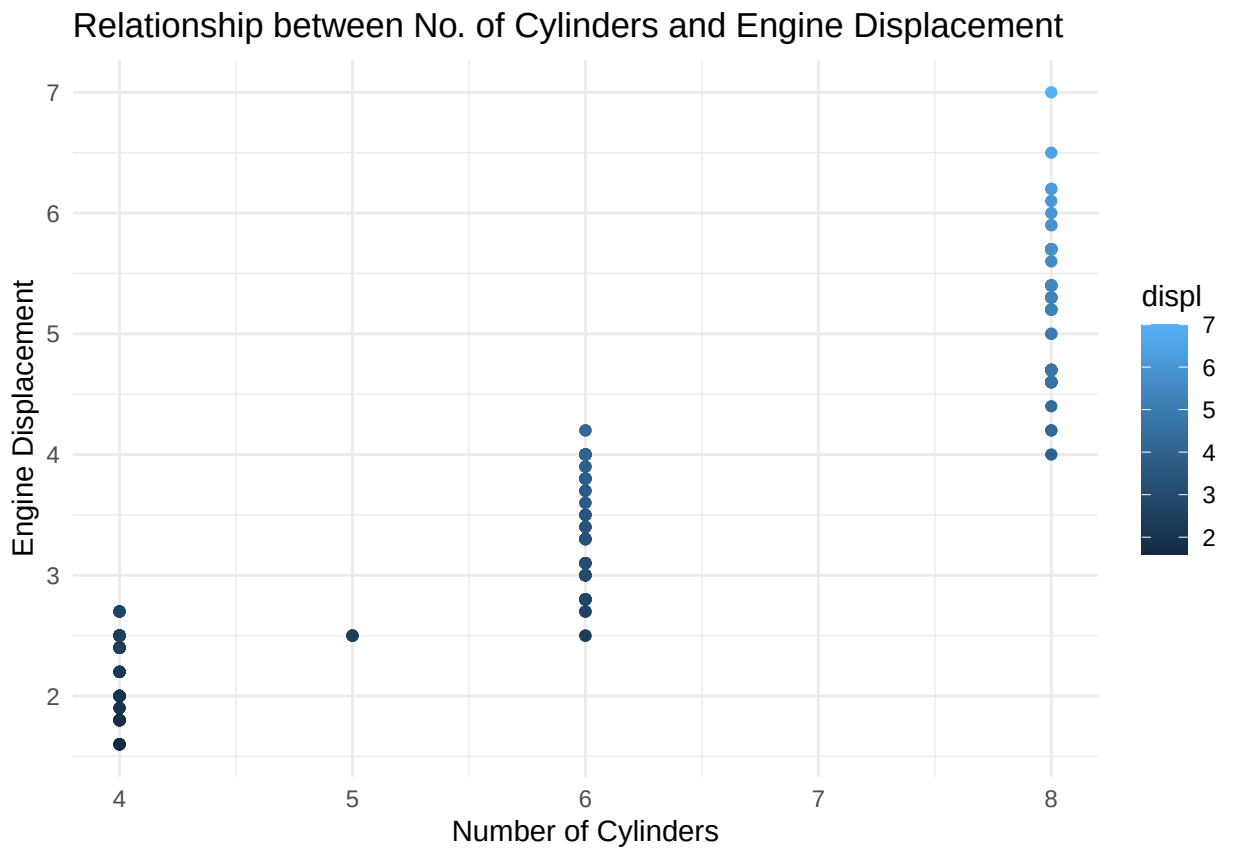

```r
#B. Plot using geom_bar() + coord_flip()
ggplot(
  carNum20,
  aes(x = reorder(model, count), y = count)
) +
  geom_bar(stat = "identity", fill = "salmon") +
  labs(
```

```
    title = "Top 20 Observations of Number of Cars per Model",
    x = "Model",
    y = "Count"
  ) +
  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.

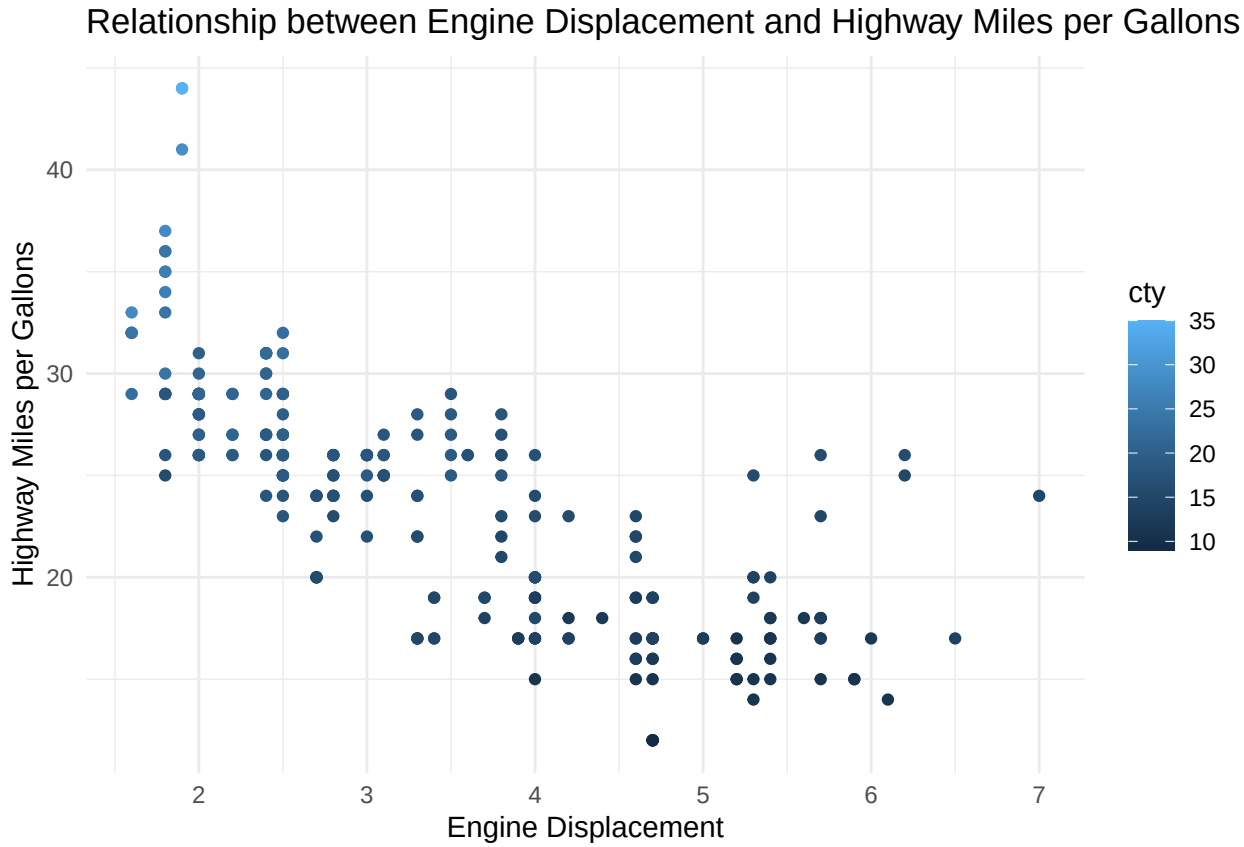
```
#A. How would you describe its relationship? Show the codes and its result.
ggplot(mpg_data,
       aes(x = cyl, y = displ, color = displ)) +
       geom_point() +
       labs(
          title = "Relationship between No. of Cylinders and Engine Displacement",
          x = "Number of Cylinders",
          y = "Engine Displacement"
       ) +
  theme_minimal()
```




- From my own observations, the cars with higher number of cylinders often comes with higher engine displacement.

6.1: 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?

```
ggplot(mpg_data,
       aes(x = displ, y = hwy, color = cty)
       ) +
       geom_point() +
       labs(
         title = "Relationship between Engine Displacement and Highway Miles per Gallons",
         x = "Engine Displacement",
         y = "Highway Miles per Gallons"
       ) +
       theme_minimal()
```




- From my observation of the relationship between engine displacement and highway miles per gallon, cars with higher engine displacements tend to be less fuel efficient. In contrast, vehicles with lower engine displacements are generally more fuel efficient, as they consume less fuel per mile traveled.

6.2: Import traffic.csv

```
#A. Number of observations of traffic.csv
traffic_data <- read.csv("traffic.csv")

str(traffic_data)
```

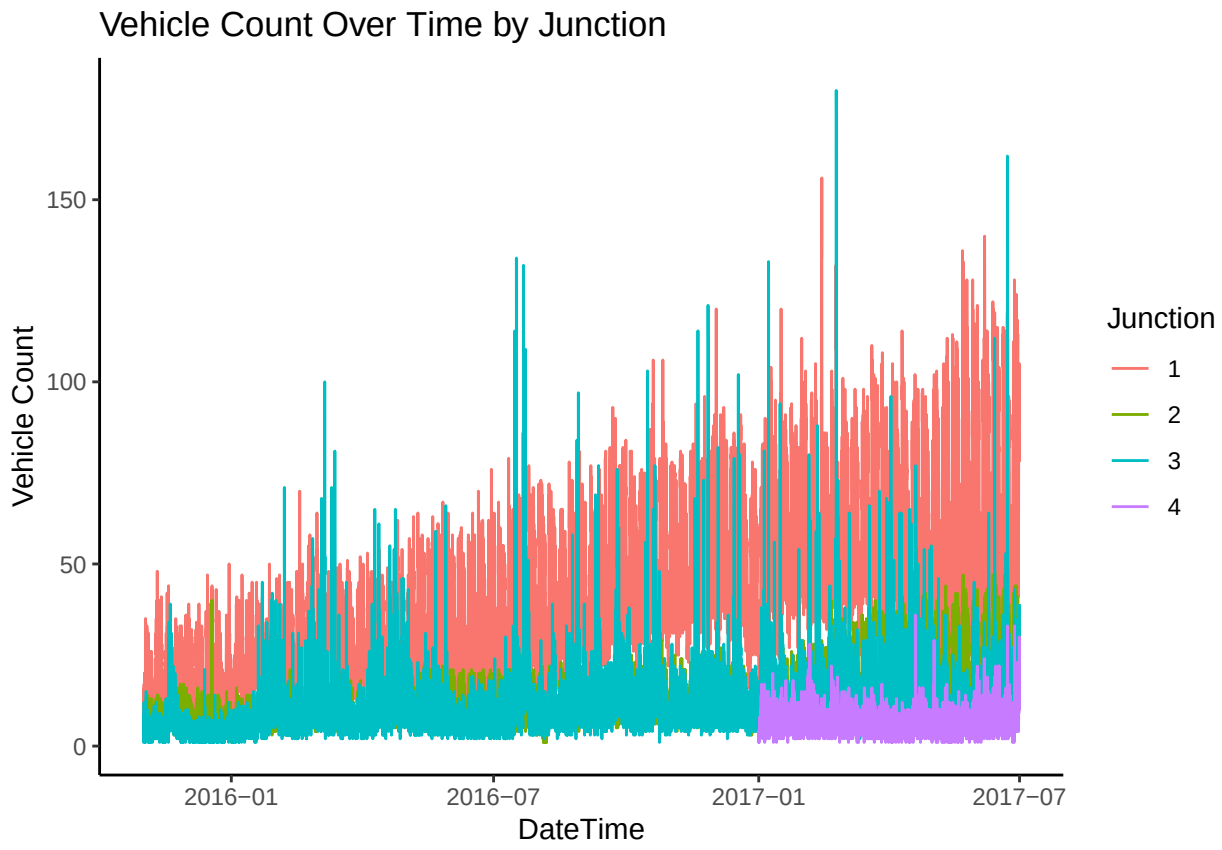
```
## 'data.frame':    48120 obs. of  4 variables:
##  $ DateTime: chr  "2015-11-01 00:00:00" "2015-11-01 01:00:00" "2015-11-01 02:00:00" "2015-11-01 03:0
##  $ Junction: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ Vehicles: int  15 13 10 7 9 6 9 8 11 12 ...
##  $ ID      : num  2.02e+10 2.02e+10 2.02e+10 2.02e+10 2.02e+10 ...
```

- The number of observations of traffic.csv is 48,120. The variables on the other is 4 which are named DateTime, Junction, Vehicles, and ID.

```
#B. Subset of the traffic dataset into junctions.
traffic_junction <- traffic_data$Junction
```

```
#C. Plot junction in a geom_line()
junction_plot <- traffic_data %>% select(DateTime, Junction, Vehicles)
```

```r
junction_plot$DateTime <- as.POSIXct(junction_plot$DateTime, format="%Y-%m-%d %H:%M:%S")

ggplot(junction_plot, aes(x = DateTime, y = Vehicles, color = factor(Junction))) +
  geom_line() +
  labs(title = "Vehicle Count Over Time by Junction",
       x = "DateTime",
       y = "Vehicle Count",
       color = "Junction") +
  theme_classic()
```




7. Import alexa_file.xlsx

```r
library(readxl)
alexa_data <- read_xlsx("alexa_file.xlsx")

#A. Number of observations and columns of alexa_file
str(alexa_data)
```

```
## tibble [3,150 x 5] (S3: tbl_df/tbl/data.frame)
##  $ rating          : num [1:3150] 5 5 4 5 5 5 3 5 5 5 ...
##  $ date            : POSIXct[1:3150], format: "2018-07-31" "2018-07-31" ...
##  $ variation       : chr [1:3150] "Charcoal Fabric" "Charcoal Fabric" "Walnut Finish" "Charcoal Fabri
##  $ verified_reviews: chr [1:3150] "Love my Echo!" "Loved it!" "Sometimes while playing a game, you ca
##  $ feedback        : num [1:3150] 1 1 1 1 1 1 1 1 1 ...
```

- The alexa_file has 3,150 number of observations and 5 number of variables or columns, these are the rating, date, variation, verified_reviews, and feedback.

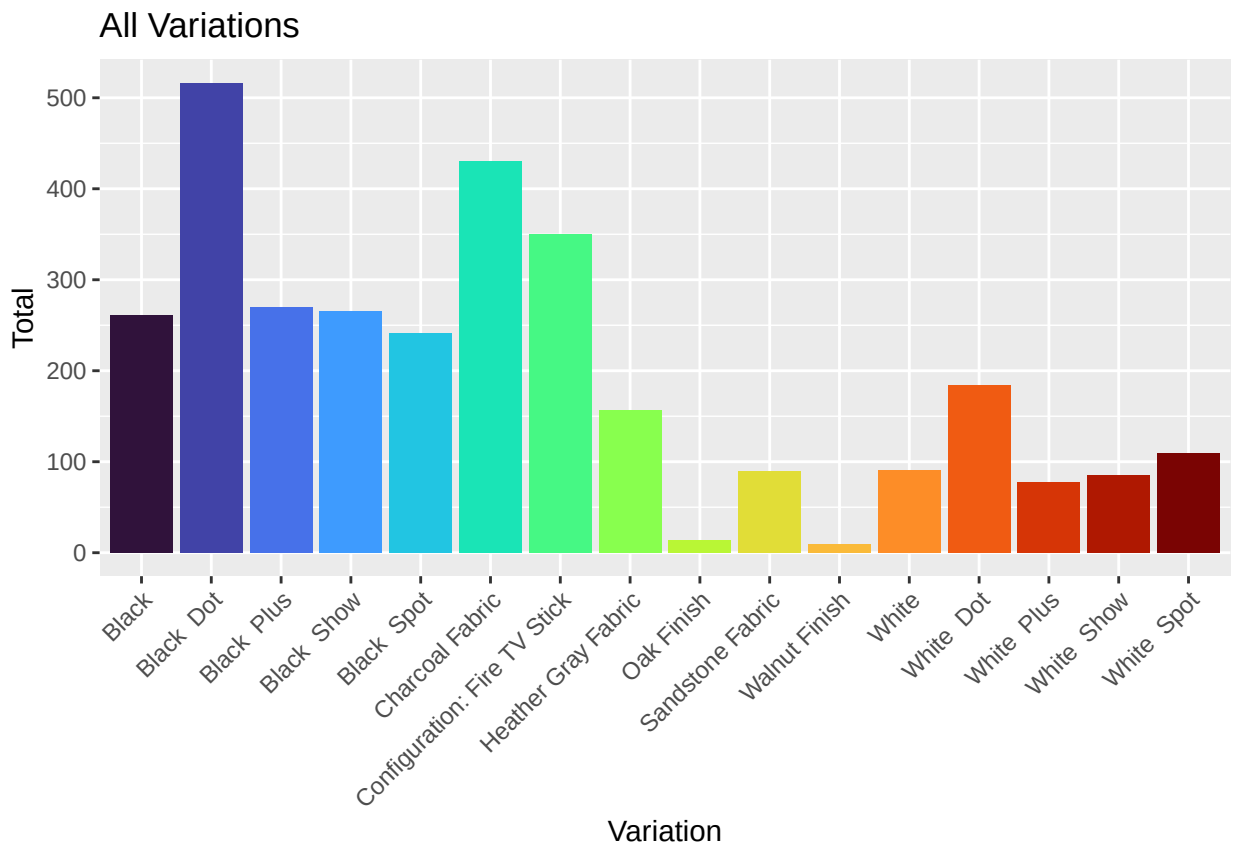
```
#B. Grouping and getting the total of each variations
alexa_variations <- alexa_data %>%
  group_by(variation) %>%
   summarise(total = n())

alexa_variations
```

```
## # 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.
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(ggplot2)

ggplot(alexa_variations, aes(x = variation, y = total, fill = variation)) +
  geom_bar(stat = "identity") +
  labs(title = "All Variations",
       x = "Variation",
       y = "Total") +
       theme(legend.position = "none") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_viridis_d(option = "turbo")
```



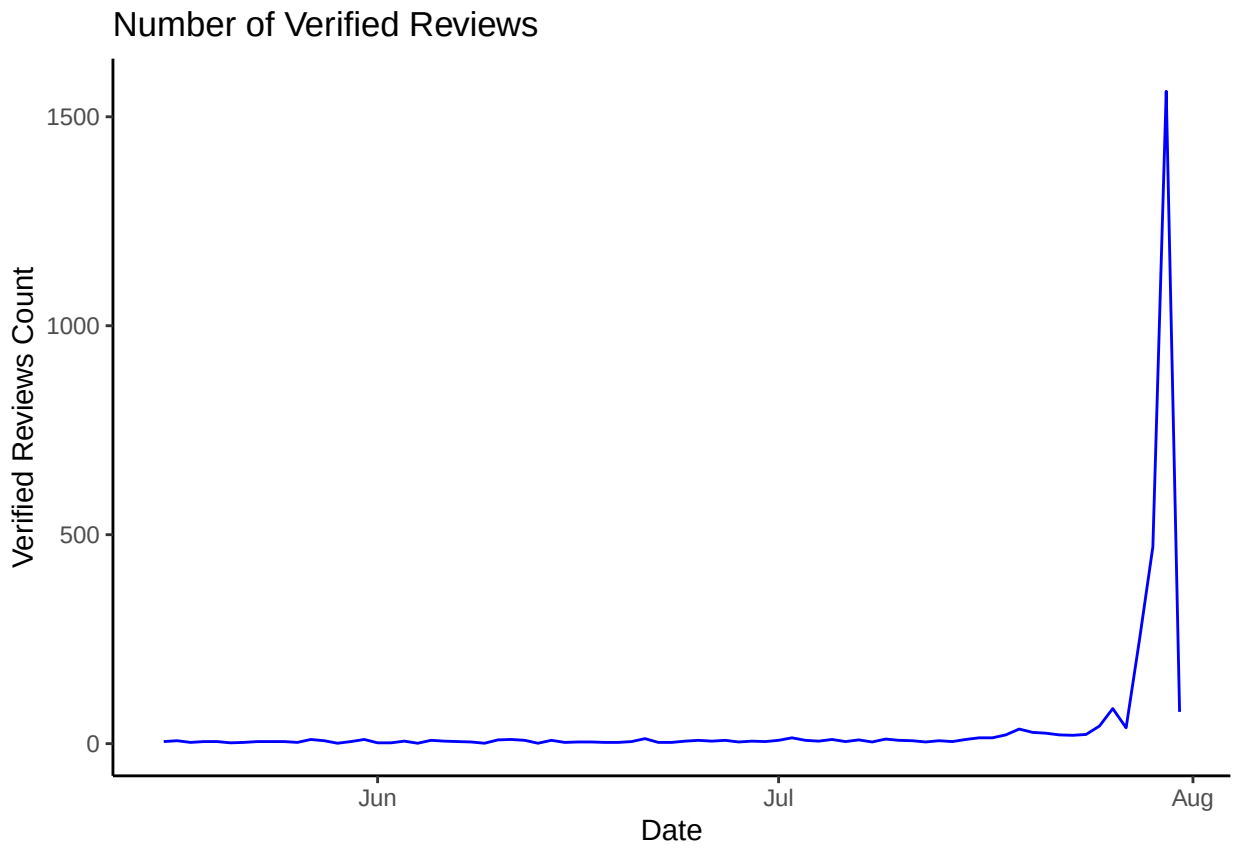

- Based on my observation, the dark colored variations are the most dominant ones, most of it have higher total than those in the white or light colored variations.

```
#D. Plot a geom_line() with the date and the number of verified reviews.
library(ggplot2)
library(dplyr)

reviews <- alexa_data %>%
  filter(!is.na(verified_reviews)) %>%
  group_by(date) %>%
  summarise(reviews_num = n())

reviews
```

```
## # A tibble: 77 x 2
##    date                reviews_num
##    <dttm>                    <int>
##  1 2018-05-16 00:00:00           5
##  2 2018-05-17 00:00:00           7
##  3 2018-05-18 00:00:00           3
##  4 2018-05-19 00:00:00           5
##  5 2018-05-20 00:00:00           5
##  6 2018-05-21 00:00:00           2
##  7 2018-05-22 00:00:00           3
##  8 2018-05-23 00:00:00           5
##  9 2018-05-24 00:00:00           5
## 10 2018-05-25 00:00:00           5
```

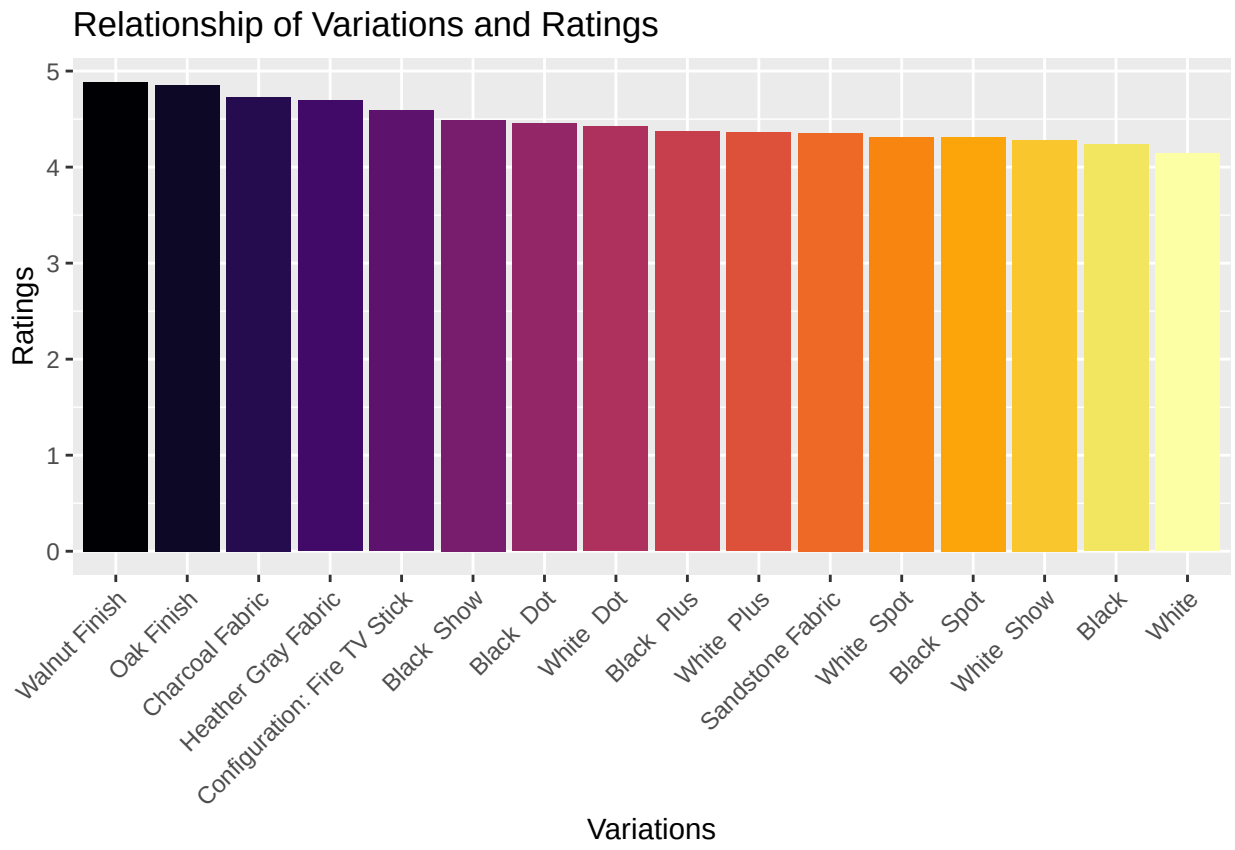
```
## # i 67 more rows
```

```
ggplot(reviews, aes(x = date, y = reviews_num)) +
  geom_line(color = "blue") +
  labs(title = "Number of Verified Reviews",
       x = "Date",
       y = "Verified Reviews Count") +
  theme_classic()
```




```
#E. Get the relationship of variations and ratings. Which variations got the most highest in rating? Pl
library(forcats)
ratings_data <- alexa_data %>%
  group_by(variation) %>%
  summarise(avg_rating = mean(rating))

ratings_data <- ratings_data %>%
  mutate(variation = fct_reorder(variation, avg_rating, .desc = TRUE))

ggplot(ratings_data, aes(x = variation, y = avg_rating, fill = variation)) +
  geom_bar(stat = "identity") +
  labs(
    title = "Relationship of Variations and Ratings",
    x = "Variations",
    y = "Ratings"
  ) +
```

```
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
theme(legend.position = "none") +
scale_fill_viridis_d(option = "inferno")
```




- The top 3 variations that got highest ratings are the Walnut Finish followed by Oak Finish and Charcoal Fabric.