RWorksheet_Malayas#4c

Andrew Miguel M. Malayas BSIT2A

2024-11-01

1. Use the dataset mpg

a. Show your solutions on how to import a csv file into the environment.

```
library(readr)
mpg <- read_csv("mpg.csv")
```

```
## New names:
## Rows: 234 Columns: 12
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (6): manufacturer, model, trans, drv, fl, class dbl (6): ...1, displ, year,
## cyl, cty, hwy
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
mpg
```

```
## # A tibble: 234 x 12
##     ...1 manufacturer model      displ  year   cyl trans drv     cty   hwy fl
##    <dbl> <chr>        <chr>      <dbl> <dbl> <dbl> <chr> <chr> <dbl> <dbl> <chr>
##  1     1 audi         a4           1.8  1999     4 auto~ f        18    29 p
##  2     2 audi         a4           1.8  1999     4 manu~ f        21    29 p
##  3     3 audi         a4           2    2008     4 manu~ f        20    31 p
##  4     4 audi         a4           2    2008     4 auto~ f        21    30 p
##  5     5 audi         a4           2.8  1999     6 auto~ f        16    26 p
##  6     6 audi         a4           2.8  1999     6 manu~ f        18    26 p
##  7     7 audi         a4           3.1  2008     6 auto~ f        18    27 p
##  8     8 audi         a4 quattro   1.8  1999     4 manu~ 4        18    26 p
##  9     9 audi         a4 quattro   1.8  1999     4 auto~ 4        16    25 p
## 10    10 audi         a4 quattro   2    2008     4 manu~ 4        20    28 p
## # i 224 more rows
## # i 1 more variable: class <chr>
```

b. Which variables from mpg dataset are categorical?

manufacturer - The car manufacturer (e.g., Audi, Chevrolet). - model - trans - drv - cyl - fl - class - manufacturer

c. Which are continuous variables?

- displ
- year
- cyt
- hwy

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

```
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
```

```
manufacturer_model_count <- mpg %>%
  group_by(manufacturer) %>%
  summarise(model_count = n_distinct(model)) %>%
  arrange(desc(model_count))

most_models <- manufacturer_model_count[1, ]

model_variation_count <- mpg %>%
  group_by(model) %>%
  summarise(variation_count = n()) %>%
  arrange(desc(variation_count))

most_variations_model <- model_variation_count[1, ]

most_models
```

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

```
most_variations_model
```

```
## # A tibble: 1 x 2
##   model       variation_count
##   <chr>                 <int>
## 1 caravan 2wd              11
```

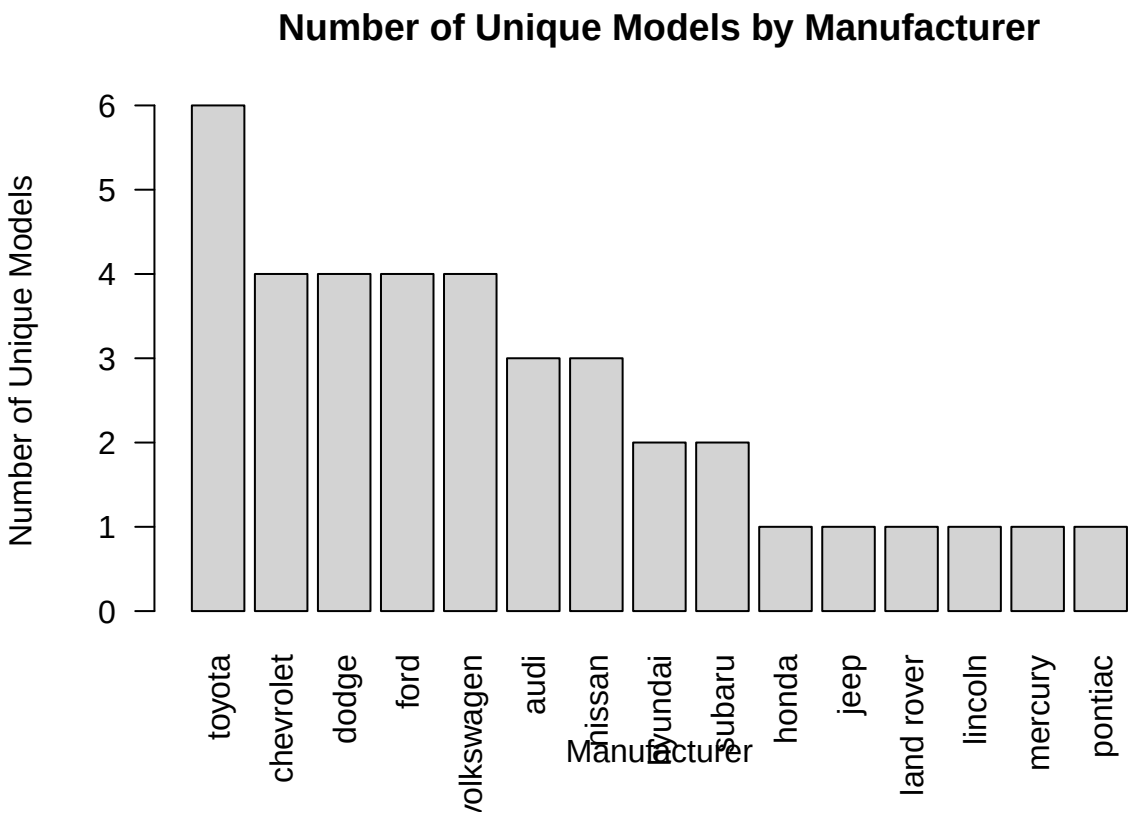
a. Group the manufacturers and find the unique models. Show your codes and result.

```r
unique_models <- mpg %>%
  group_by(manufacturer) %>%
  summarise(unique_models = list(unique(model))) %>%
  arrange(manufacturer)

print(unique_models)
```

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

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

```r
model <- mpg %>%
  group_by(manufacturer) %>%
  summarise(unique_count = n_distinct(model)) %>%
  arrange(desc(unique_count))


barplot(model$unique_count,
        names.arg = model$manufacturer,
        las = 2,
        col = "lightgrey",
        main = "Number of Unique Models by Manufacturer",
        xlab = "Manufacturer",
        ylab = "Number of Unique Models")
```

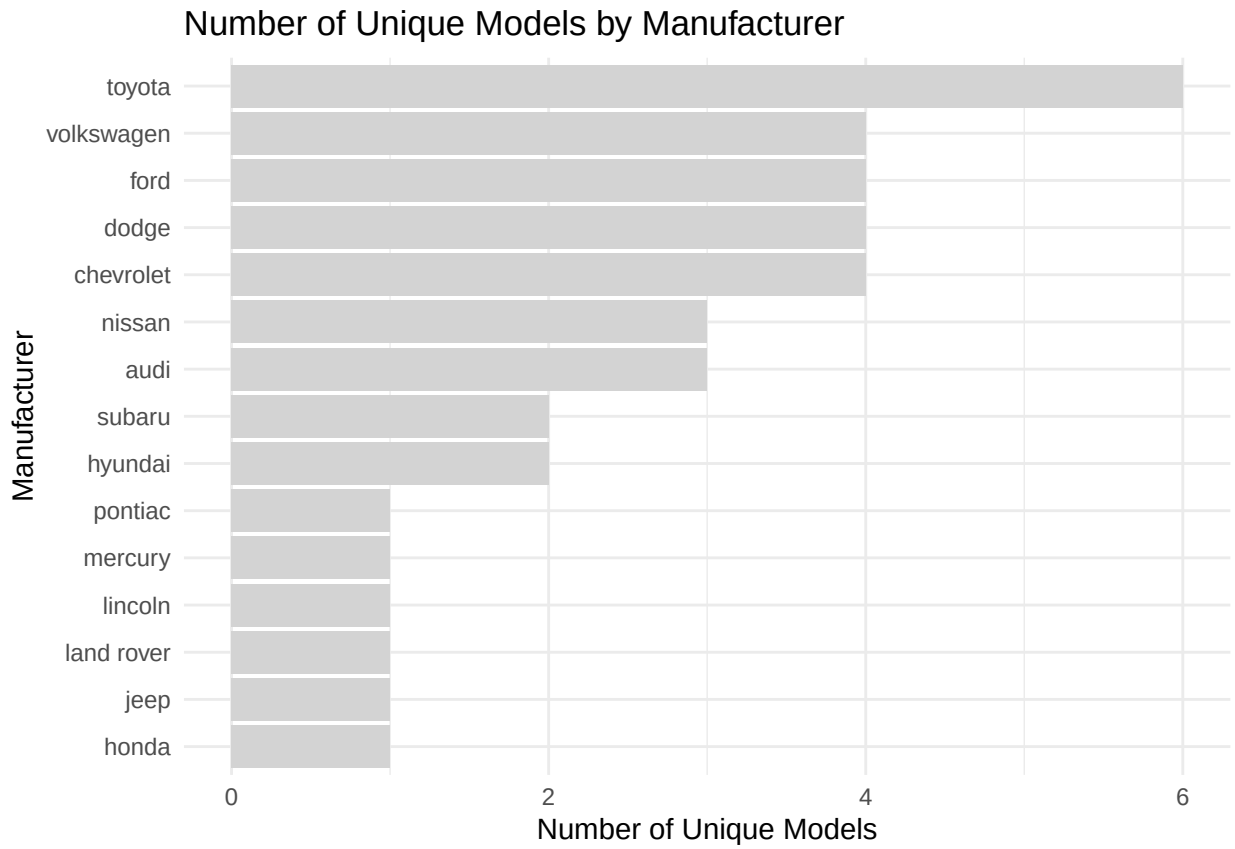



```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked _by_ '.GlobalEnv':
##
##     mpg
```

```
ggplot(model, aes(x = reorder(manufacturer, unique_count), y = unique_count)) +
  geom_bar(stat = "identity", fill = "lightgrey") +
  coord_flip() +
  labs(title = "Number of Unique Models by Manufacturer",
       x = "Manufacturer",
       y = "Number of Unique Models") +
  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?

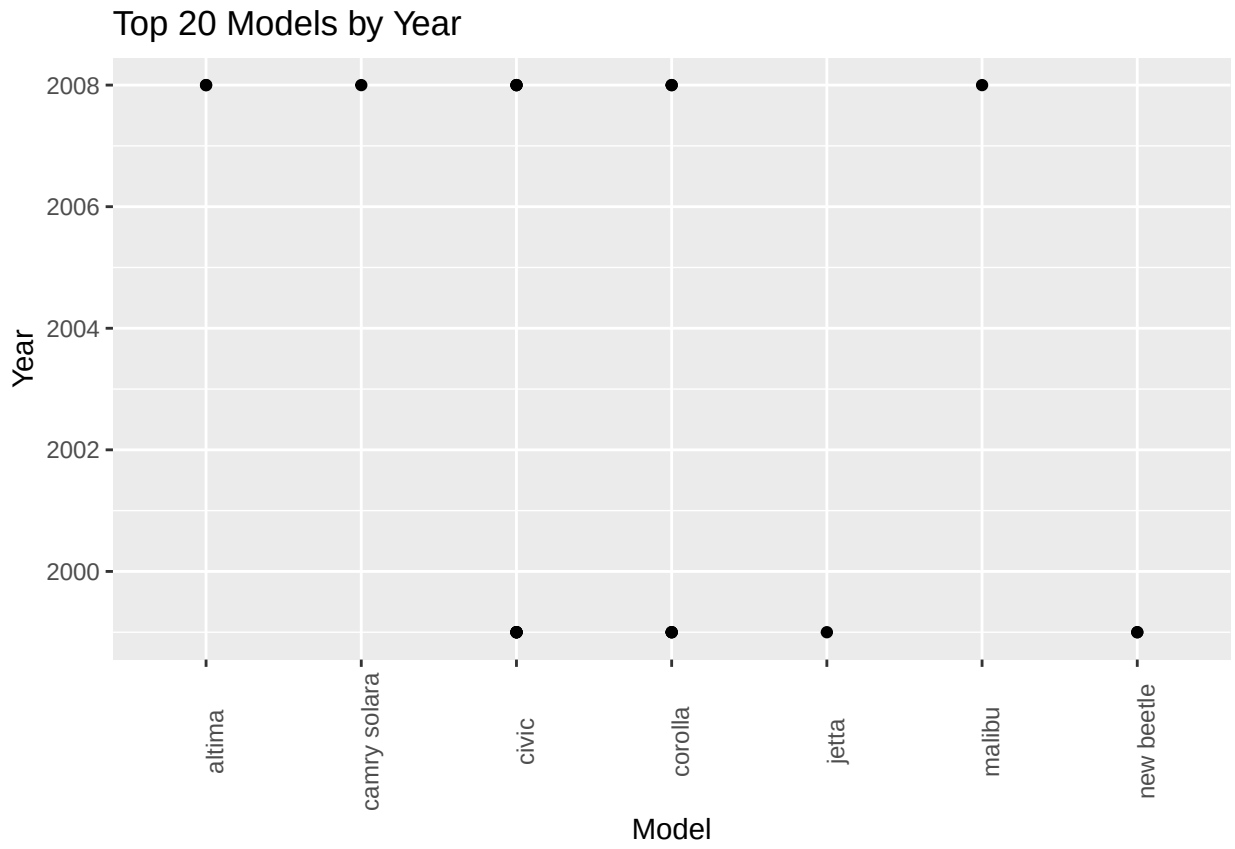
This code generates a scatter plot where each point represents a unique pairing of model and manufacturer. It illustrates which models are produced by each manufacturer, showing the distribution of models across different manufacturers.

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

As it stands, the plot isn't particularly helpful for analyzing the relationship between model and manufacturer because of the overplotting of points and the absence of numerical or continuous variables that could offer deeper insights.

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

```
Top_20 <- mpg %>%
  arrange(desc(cty)) %>%
  head(20)
ggplot(Top_20, aes(x = model, y = year)) +
  geom_point() +
  labs(title = "Top 20 Models by Year", x = "Model", y = "Year") +
  theme(axis.text.x = element_text(angle = 90, hjust = 0.5))
```



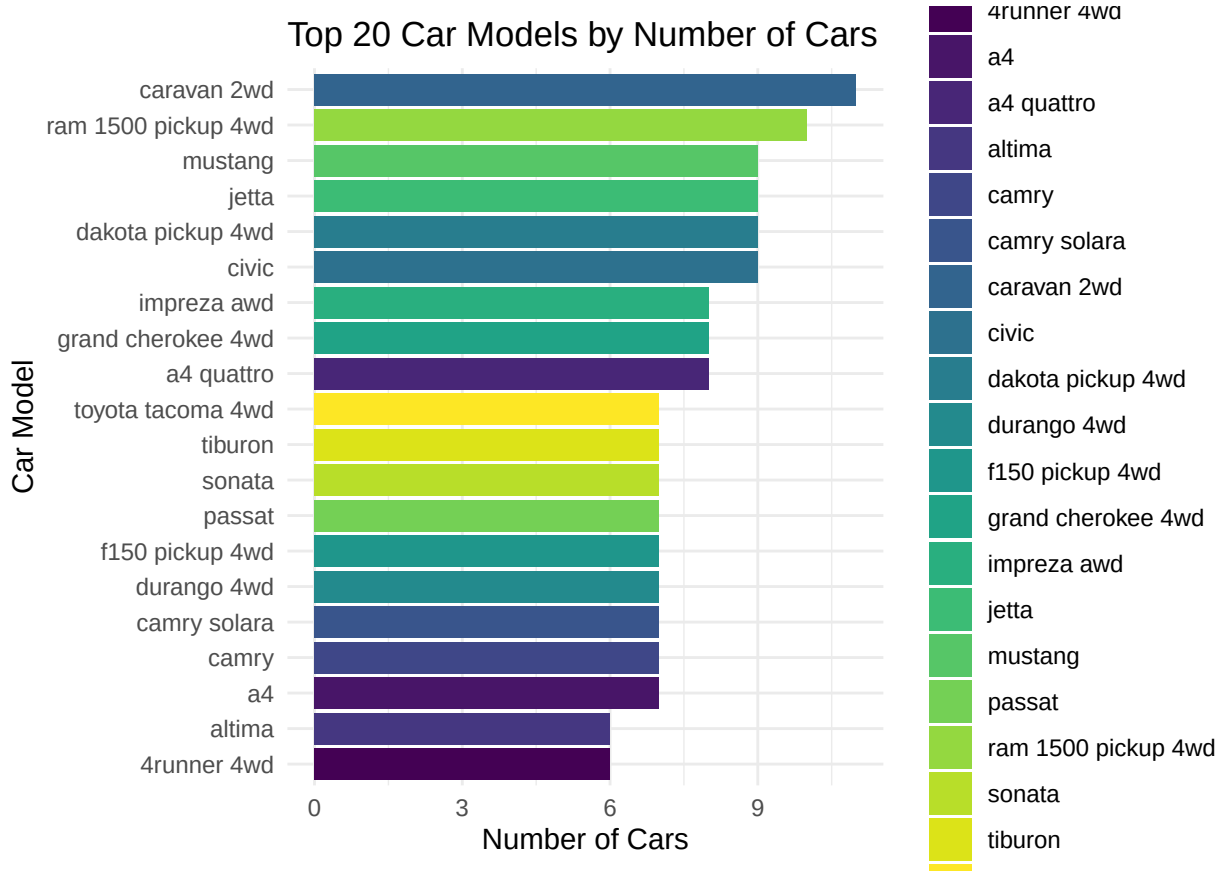
4. Using the pipe (%>%), group the model and get the number of cars per model. Show codes and its result

```
model_counts <- mpg %>%
  group_by(model) %>%
  summarise(num_of_cars = n()) %>%
  arrange(desc(num_of_cars))

print(model_counts)
```

```
## # A tibble: 38 x 2
##    model               num_of_cars
##    <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
```

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

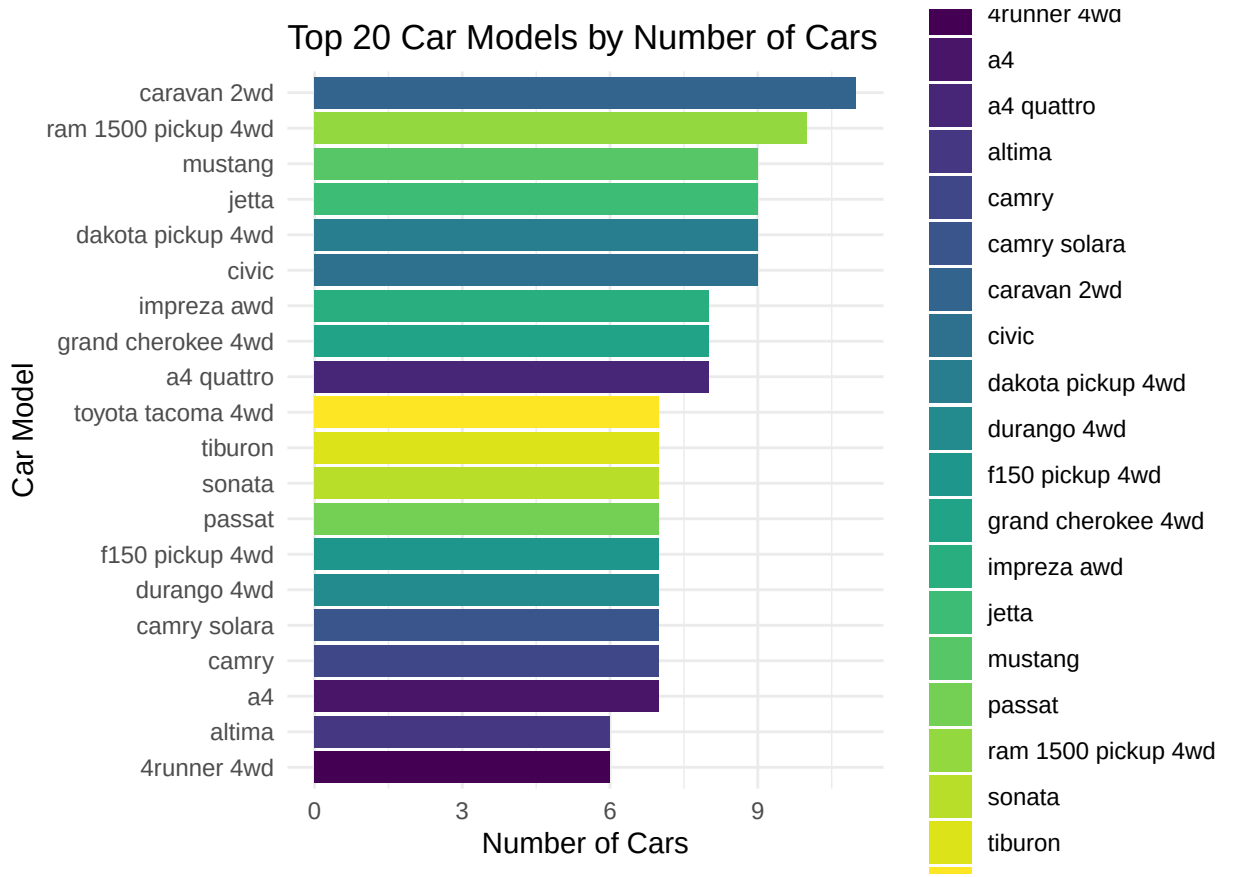
```
top_20_models <- mpg %>%
  group_by(model) %>%
  summarise(num_of_cars = n()) %>%
  arrange(desc(num_of_cars)) %>%
  slice_head(n = 20)

ggplot(top_20_models, aes(x = reorder(model, num_of_cars), y = num_of_cars, fill = model)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 20 Car Models by Number of Cars",
       x = "Car Model",
       y = "Number of Cars",
       fill = "Model") +
  theme_minimal() +
  scale_fill_viridis_d()
```



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

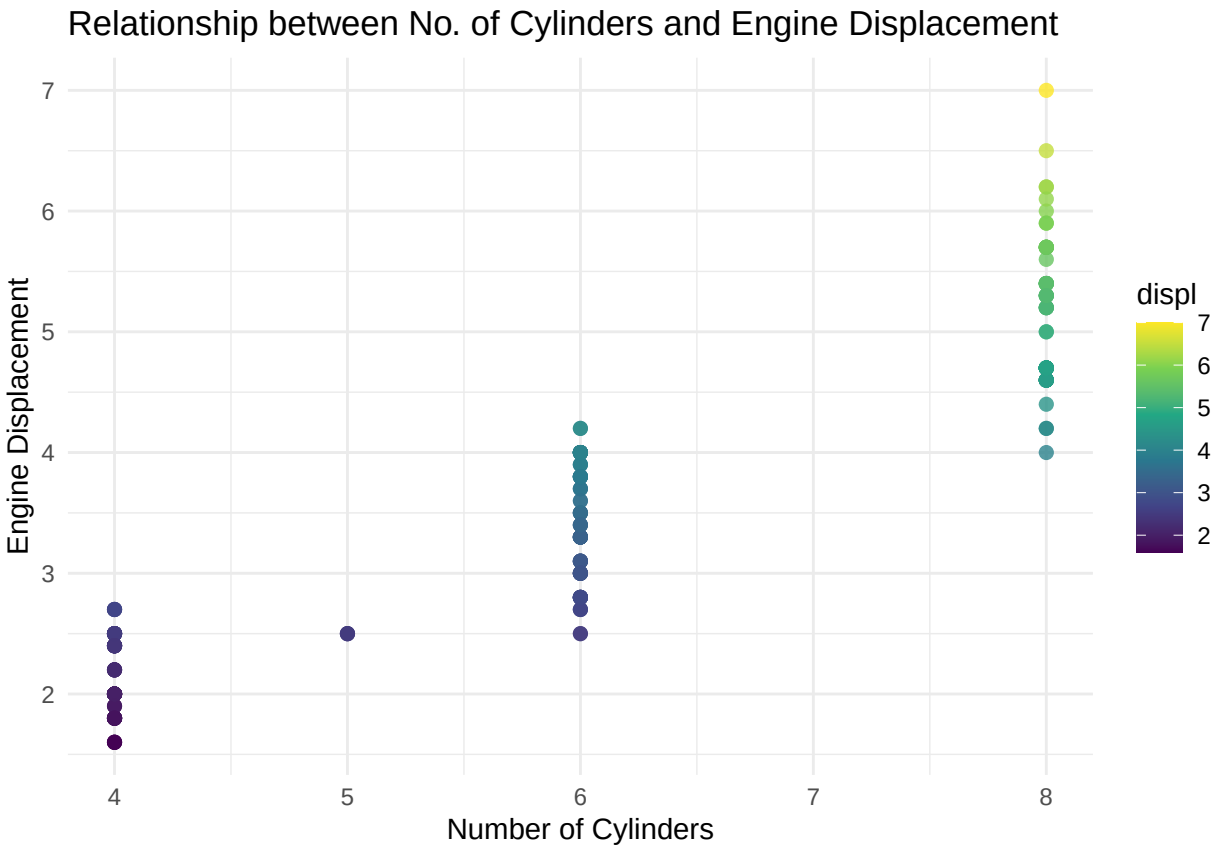
```
top_20_models <- mpg %>%
  group_by(model) %>%
  summarise(num_of_cars = n()) %>%
  arrange(desc(num_of_cars)) %>%
  slice_head(n = 20)
```

```r
ggplot(top_20_models, aes(x = reorder(model, num_of_cars), y = num_of_cars, fill = model)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 20 Car Models by Number of Cars",
       x = "Car Model",
       y = "Number of Cars") +
  theme_minimal() +
  scale_fill_viridis_d()
```



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
ggplot(mpg, aes(x = cyl, y = displ, color = displ)) +
  geom_point(size = 2, alpha = 0.8) +
  labs(title = "Relationship between No. of Cylinders and Engine Displacement",
       x = "Number of Cylinders",
       y = "Engine Displacement") +
  theme_minimal() +
  scale_color_viridis_c()
```



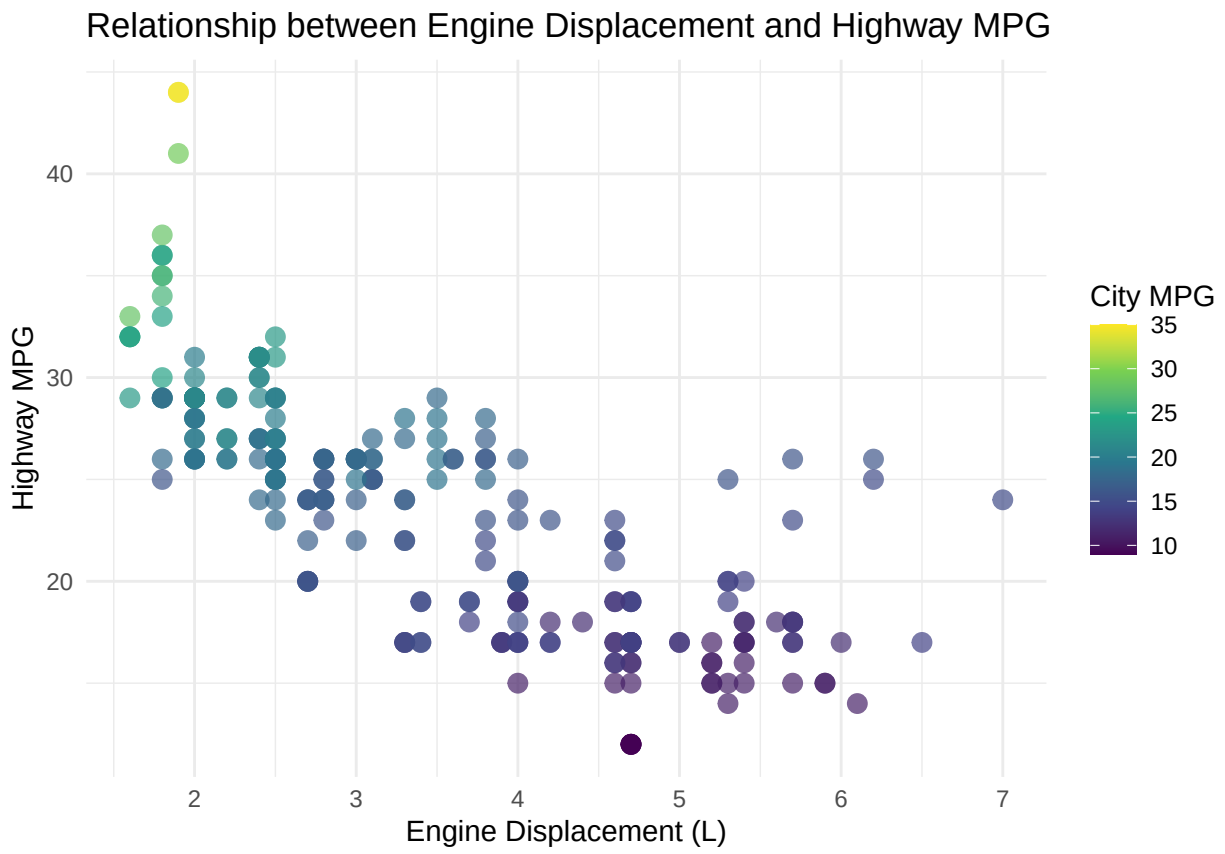
a. How would you describe its relationship? Show the codes and its result.

The plot indicates that as the number of cylinders rises, engine displacement generally follows an upward trend. This suggests a positive correlation, where vehicles with a higher cylinder count tend to have greater engine displacement.

The color gradient adds another layer of information, with darker colors representing lower engine displacement values and lighter colors representing higher values. This helps illustrate how engine displacement varies across different cylinder counts.

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?

```
ggplot(mpg, aes(x = displ, y = hwy, color = cty)) +
  geom_point(size = 3, alpha = 0.7) +
  labs(title = "Relationship between Engine Displacement and Highway MPG",
       x = "Engine Displacement (L)",
       y = "Highway MPG",
       color = "City MPG") +
  theme_minimal() +
  scale_color_viridis_c()
```



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

```
library(readr)
traffic_info <- read_csv("traffic.csv")
```

```
## Rows: 48120 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (3): Junction, Vehicles, ID
## dttm (1): DateTime
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(traffic_info)
```

```
## # A tibble: 6 x 4
##   DateTime            Junction Vehicles          ID
##   <dttm>                 <dbl>    <dbl>       <dbl>
## 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_of_observations <- nrow(traffic_info)
variables <- colnames(traffic_info)
num_of_observations
```

```
## [1] 48120
```

```
variables
```

```
## [1] "DateTime" "Junction" "Vehicles" "ID"
```

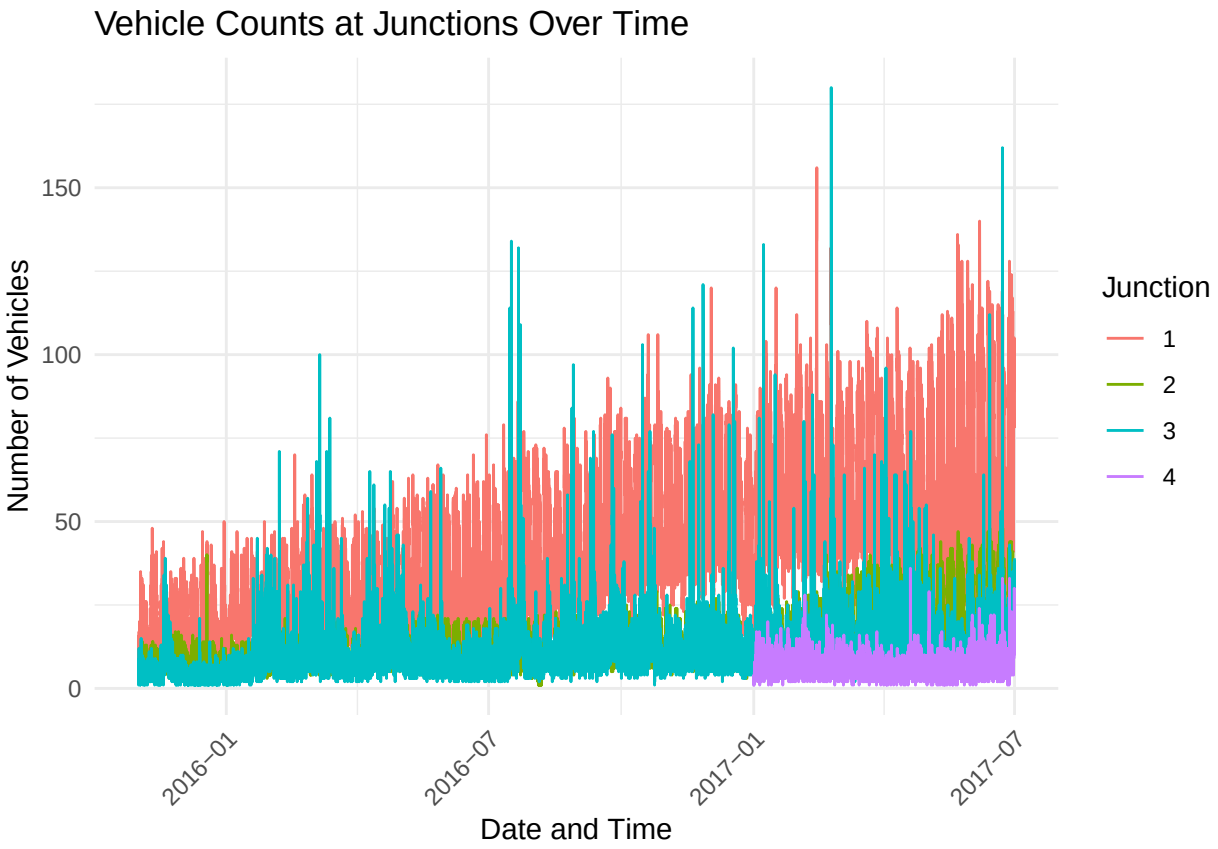
b. subset the traffic dataset into junctions. What is the R codes and its output?

```
data_to_junctions <- subset(traffic_info, Junction == TRUE)
head(data_to_junctions)
```

```
## # A tibble: 6 x 4
##   DateTime            Junction Vehicles          ID
##   <dttm>                 <dbl>    <dbl>       <dbl>
## 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
```

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

```
ggplot(traffic_info, aes(x = DateTime, y = Vehicles, color = factor(Junction))) +
  geom_line() +
  labs(title = "Vehicle Counts at Junctions Over Time",
       x = "Date and Time",
       y = "Number of Vehicles",
       color = "Junction") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



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

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

```
library(readxl)

alexa <- read_excel("alexa_file.xlsx")
dimensions <- dim(alexa)
num_rows <- dimensions[1]
num_columns <- dimensions[2]

num_rows
```

```
## [1] 3150
```

```
num_columns
```

```
## [1] 5
```

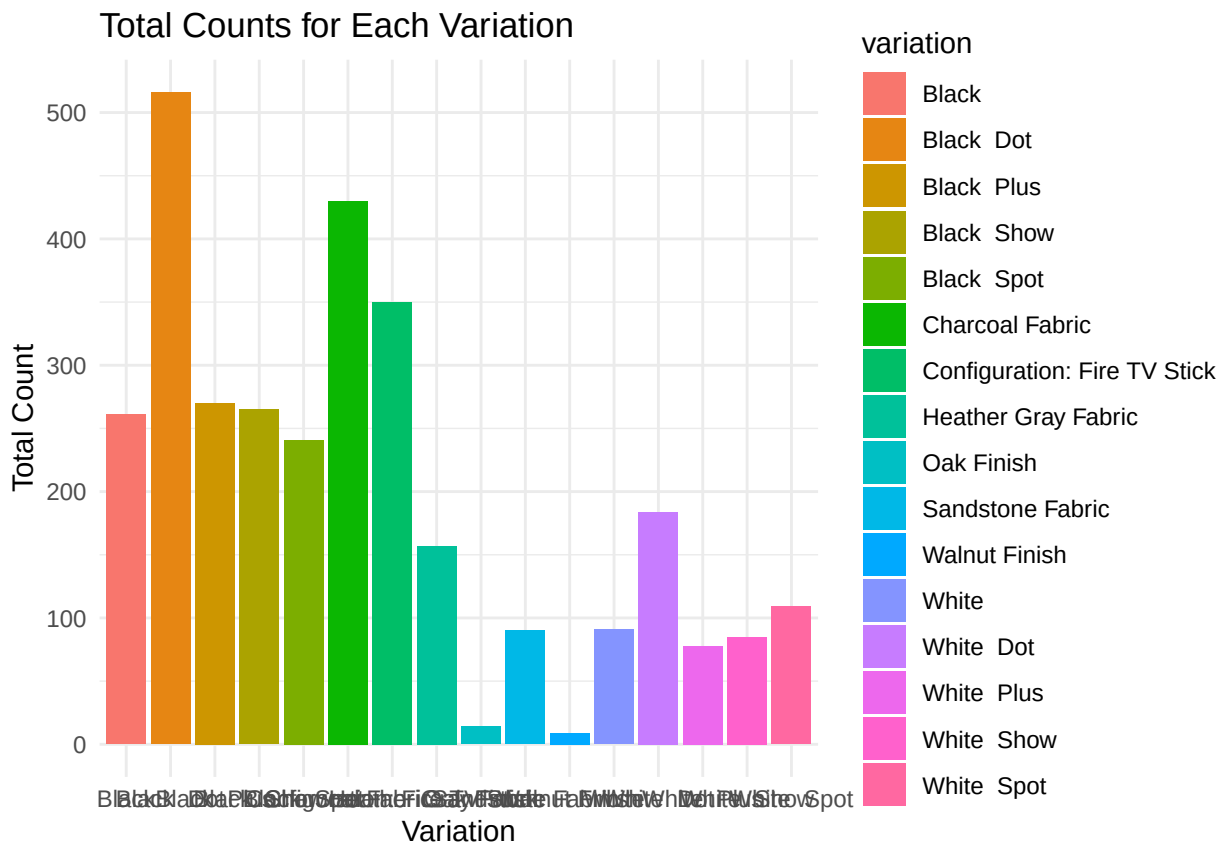
b. group the variations and get the total of each variations. Use dplyr package. Show solution and answer.

```r
total_variations <- alexa %>%
  group_by(variation) %>%
  summarise(total = n())
total_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. What did you observe? Complete the details of the graph. Show solution and answer.

```r
ggplot(total_variations, aes(x = variation, y = total, fill = variation)) +
  geom_bar(stat = "identity") +
  labs(title = "Total Counts for Each Variation", x = "Variation", y = "Total Count") +
  theme_minimal()
```

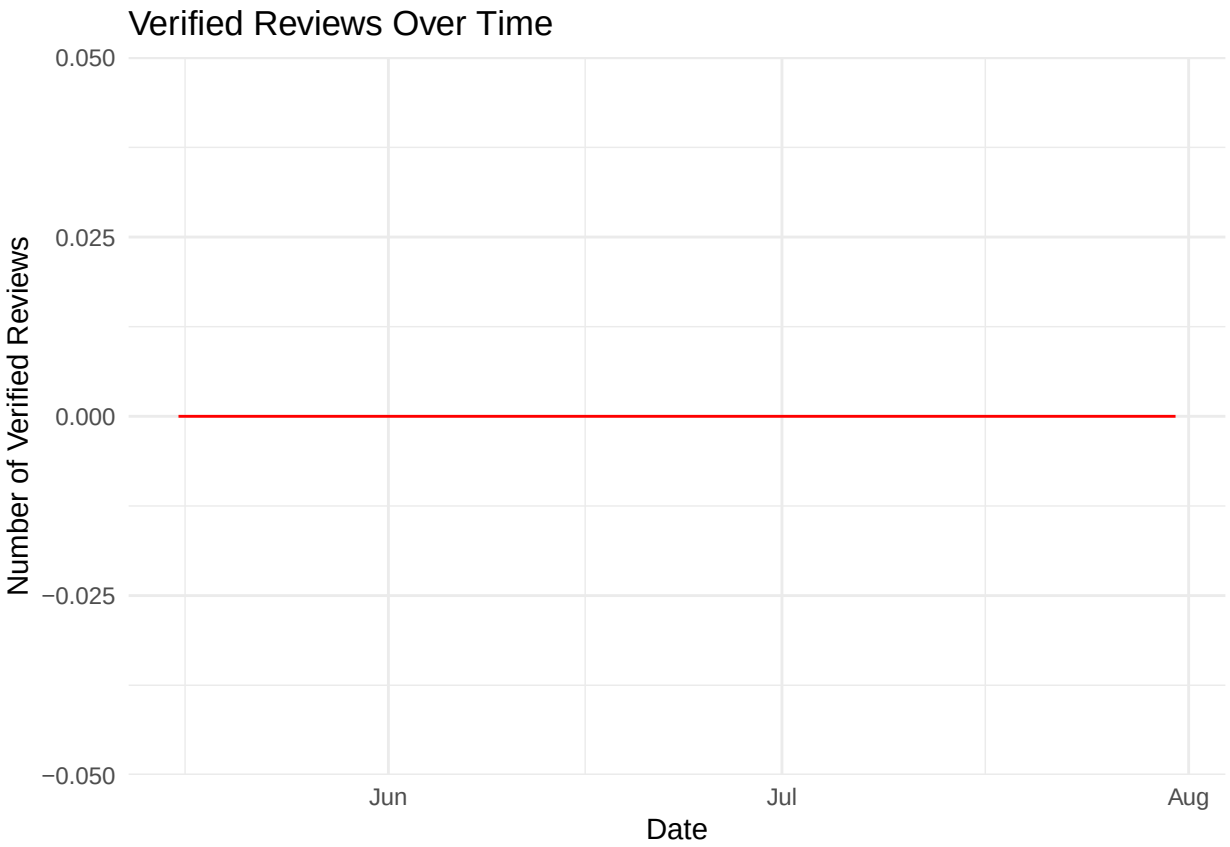


d. Plot a geom_line() with the date and the number of verified reviews. Complete the details of the graphs. Show your answer and solution.

```
alexa$verified_reviews <- as.numeric(alexa$verified_reviews)
```

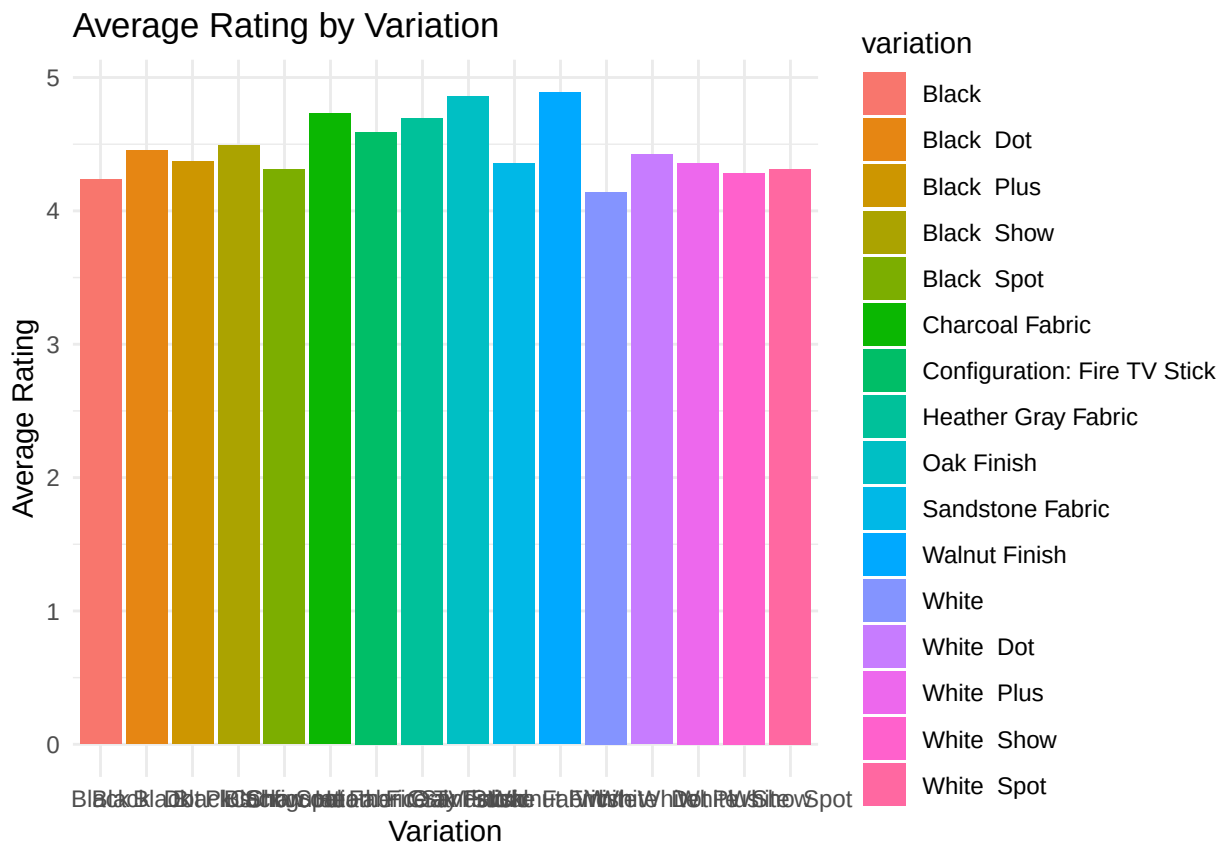
```
## Warning: NAs introduced by coercion
```

```
date_and_num_reviews <- alexa %>%
  group_by(date) %>%
  summarise(verified_reviews_total = sum(verified_reviews, na.rm = TRUE))

ggplot(date_and_num_reviews, aes(x = date, y = verified_reviews_total)) +
  geom_line(color = "red") +
  labs(title = "Verified Reviews Over Time", x = "Date", y = "Number of Verified Reviews") +
  theme_minimal()
```



e. Get the relationship of variations and ratings. Which variations got the most highest in rating? Plot a graph to show its relationship. Show your solution and answer.

```
variation_ratings <- alexa %>%
  group_by(variation) %>%
  summarise(average_rating = mean(rating, na.rm = TRUE))
ggplot(variation_ratings, aes(x = variation, y = average_rating, fill = variation)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Rating by Variation", x = "Variation", y = "Average Rating") +
  theme_minimal()
```


Average Rating by Variation
Average Rating
0
1
2
3
4
5
Variation
variation
Black
Black Dot
Black Plus
Black Show
Black Spot
Charcoal Fabric
Configuration: Fire TV Stick
Heather Gray Fabric
Oak Finish
Sandstone Fabric
Walnut Finish
White
White Dot
White Plus
White Show
White Spot