RWorksheet_guion#4c.Rmd.

Mikyla Grace Guion

2024-10-30

1.

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  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" ...
```

The categorical variables are manufacturer, model, trans, drv, fl, and class.

c. Which are continuous variables?

The continous variables are displ, year, cyl, cty, and 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
```

```
library(ggplot2)

manufacturer_count <- mpg %>%
  group_by(manufacturer) %>%
  summarise(num_models = n_distinct(model)) %>%
  arrange(desc(num_models))

manufacturer_count[1, ]
```

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

```
model_count <- mpg %>%
  group_by(model) %>%
  summarise(num_variations = n()) %>%
  arrange(desc(num_variations))

model_count[1, ]
```

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

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

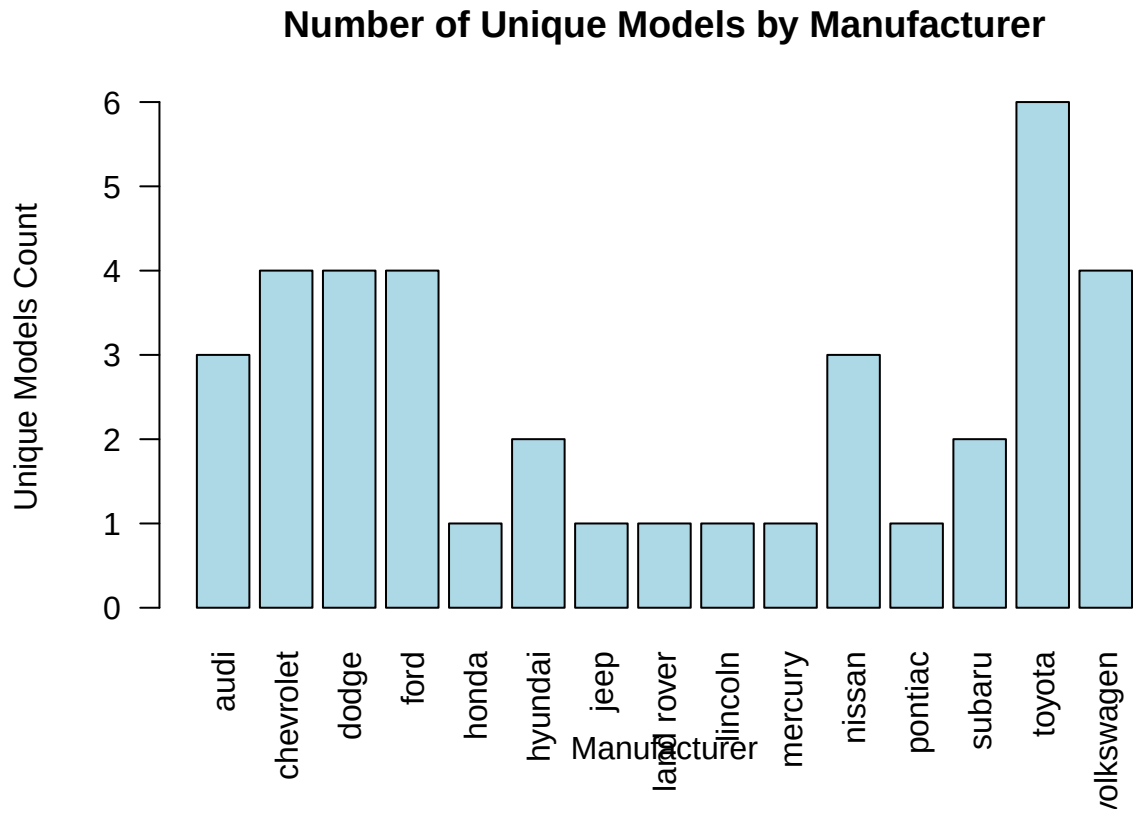
```
unique_models <- mpg %>%
  group_by(manufacturer) %>%
  summarise(unique_models_count = n_distinct(model))

unique_models
```

```
## # A tibble: 15 x 2
##    manufacturer unique_models_count
##    <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
```

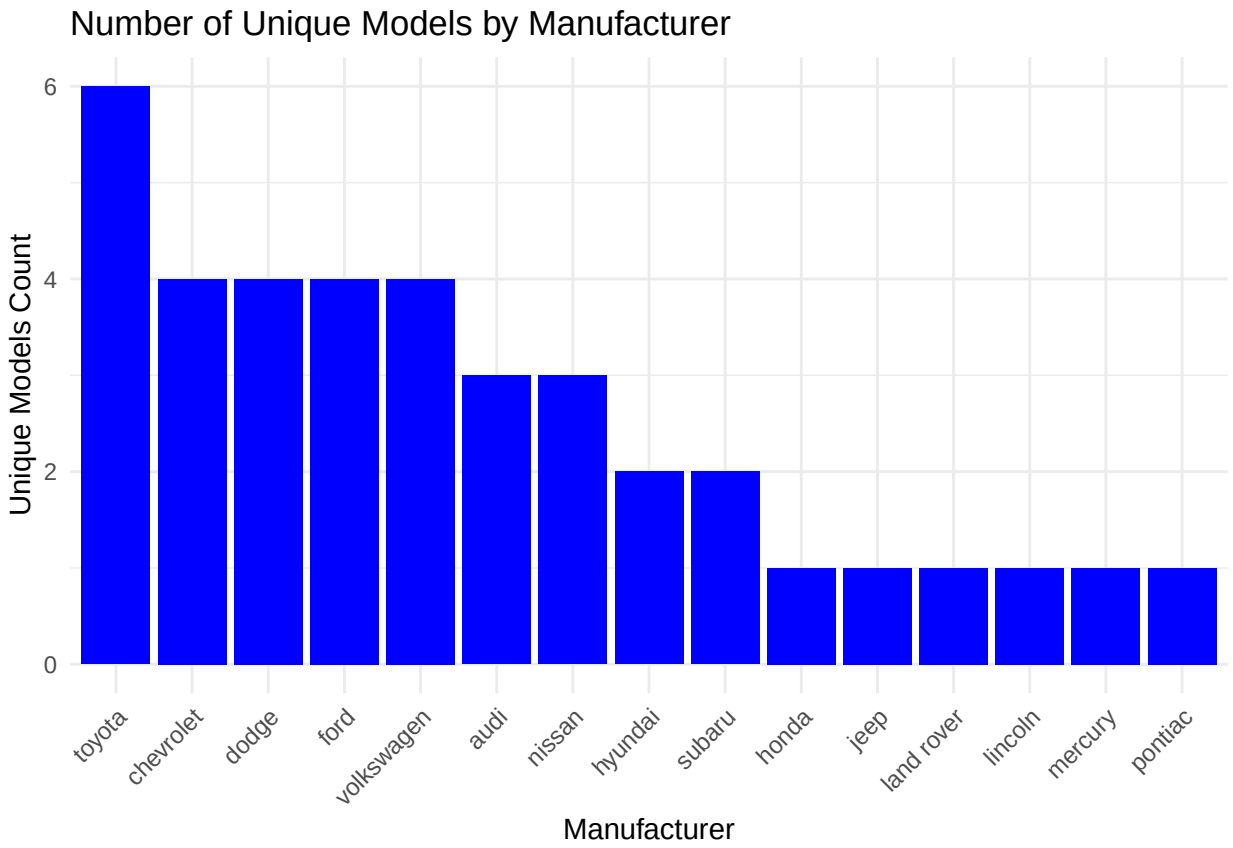
```
## 14 toyota                       6
## 15 volkswagen                   4
```

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

```r
barplot(
  unique_models$unique_models_count,
  names.arg = unique_models$manufacturer,
  las = 2,                    # Make x-axis labels vertical for better readability
  col = "lightblue",
  main = "Number of Unique Models by Manufacturer",
  xlab = "Manufacturer",
  ylab = "Unique Models Count"
)
```




```r
ggplot(unique_models, aes(x = reorder(manufacturer, -unique_models_count), y = unique_models_count)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(title = "Number of Unique Models by Manufacturer", x = "Manufacturer", y = "Unique Models Count")
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



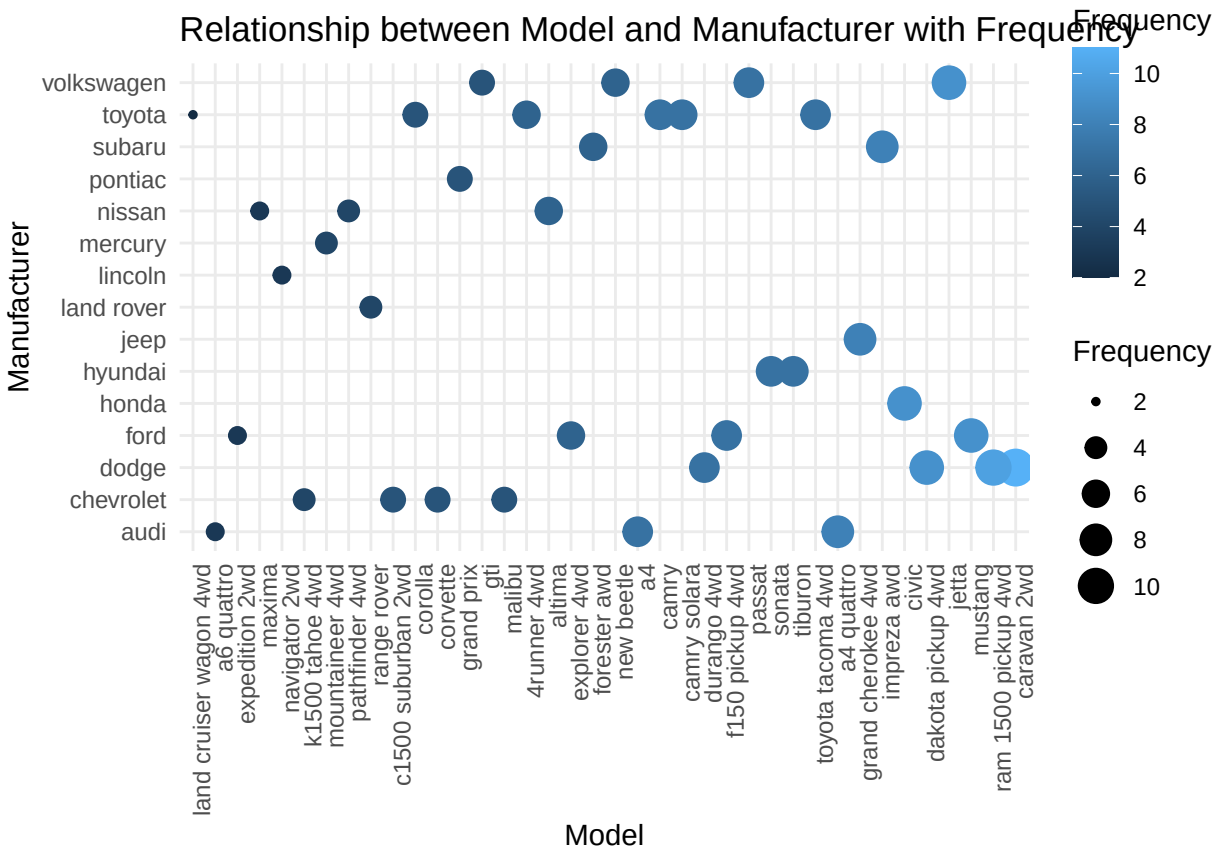
2. Same dataset will be used. You are going to show the relationship of the modeland the manufacturer.

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

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

```
## `summarise()` has grouped output by 'manufacturer'. You can override using the
## `.groups` argument.
```

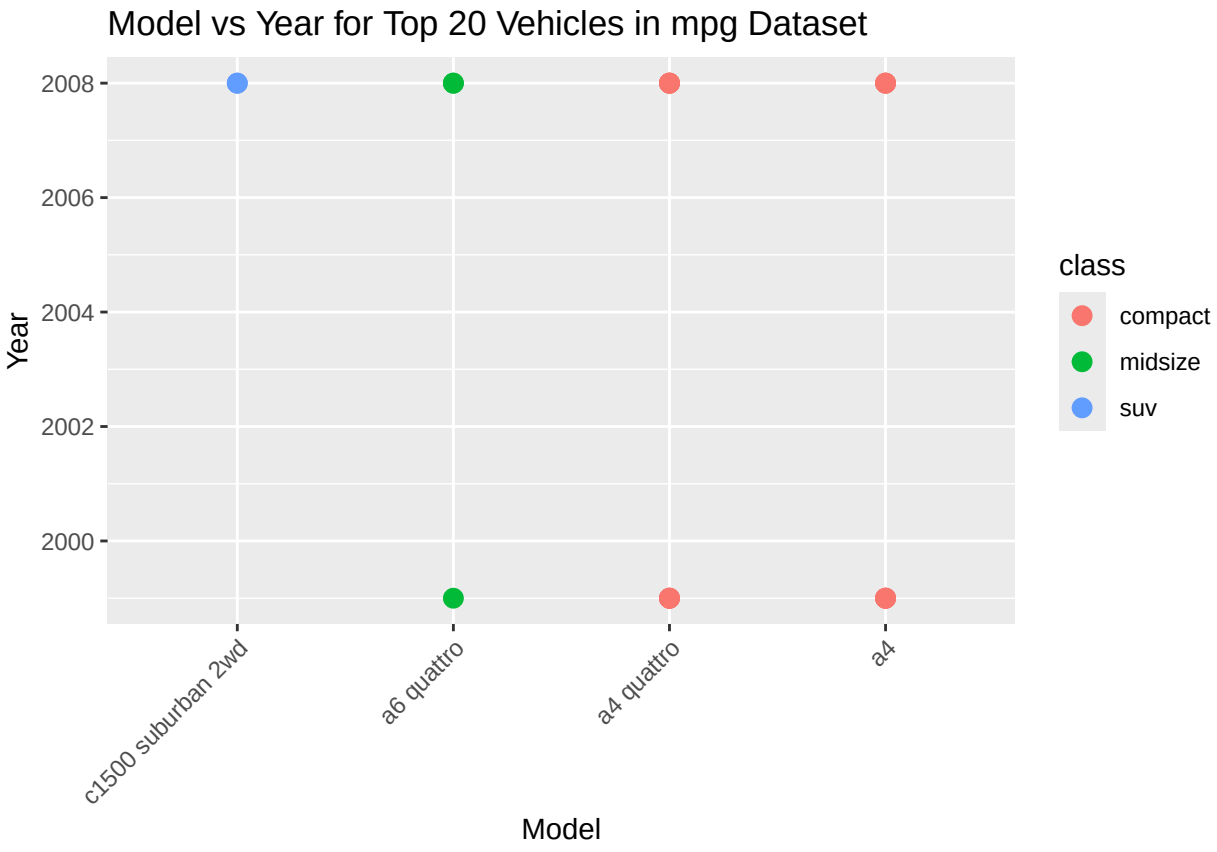
```
ggplot(mpg_summary, aes(x = reorder(model, count), y = manufacturer)) +
  geom_point(aes(size = count, color = count)) +
  labs(title = "Relationship between Model and Manufacturer with Frequency",
       x = "Model",
       y = "Manufacturer",
       size = "Frequency",
       color = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



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

For me, it's not really informative, it's hard to compare the sizes to the frequency legend plus some points are overlapping. To change these, maybe add the number to the y axis and use geom_jitter() to make data points more visible. # 3. Plot the model and the year using ggplot(). Use only the top 20 observations. Write the codes and its results.

```
top_20_mpg <- head(mpg, 20)

ggplot(top_20_mpg, aes(x = reorder(model, -year), y = year)) +
  geom_point(aes(color = class), size = 3) +  # Use points colored by class
  labs(title = "Model vs Year for Top 20 Vehicles in mpg Dataset",
       x = "Model",
       y = "Year") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

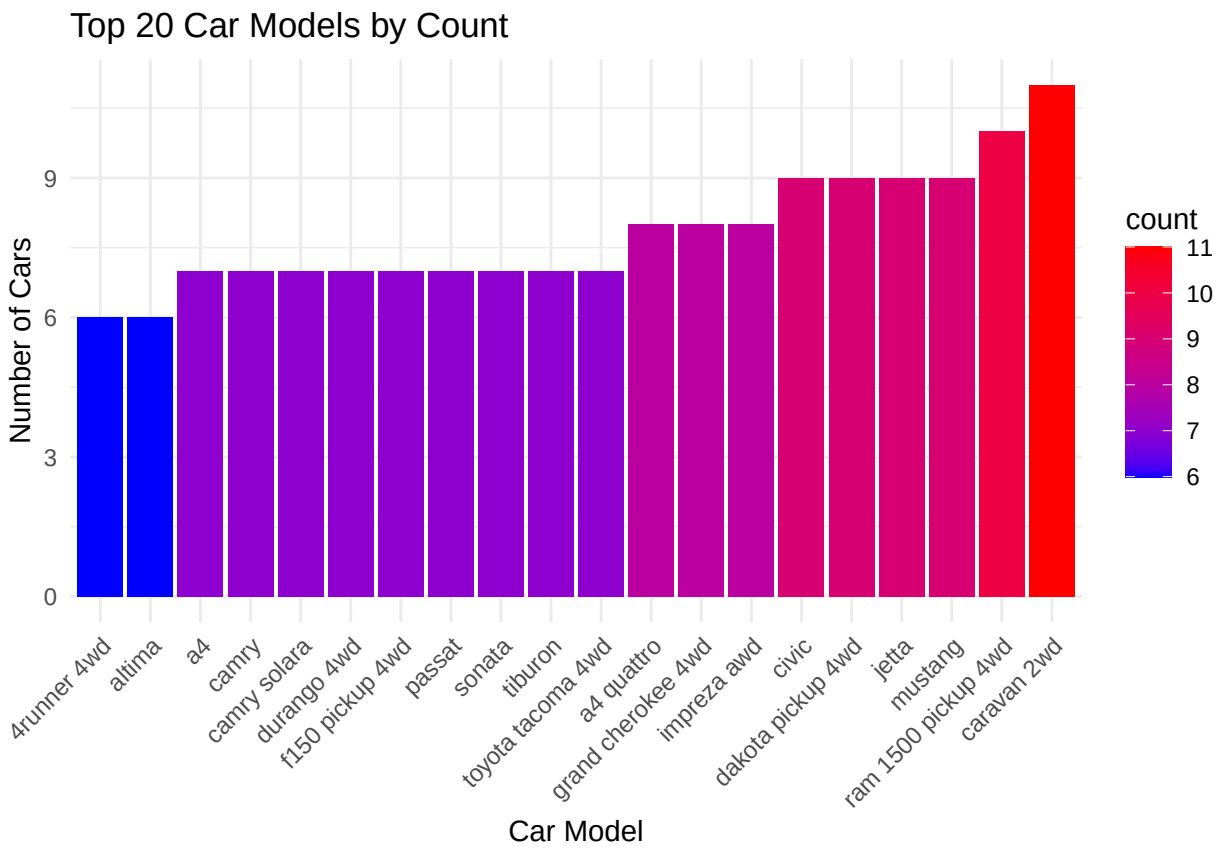


4. Using the pipe (%>%), group the model and get the number of cars per model. Show codes and its result

```
car_counts <- mpg_data %>%
  group_by(model) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
```

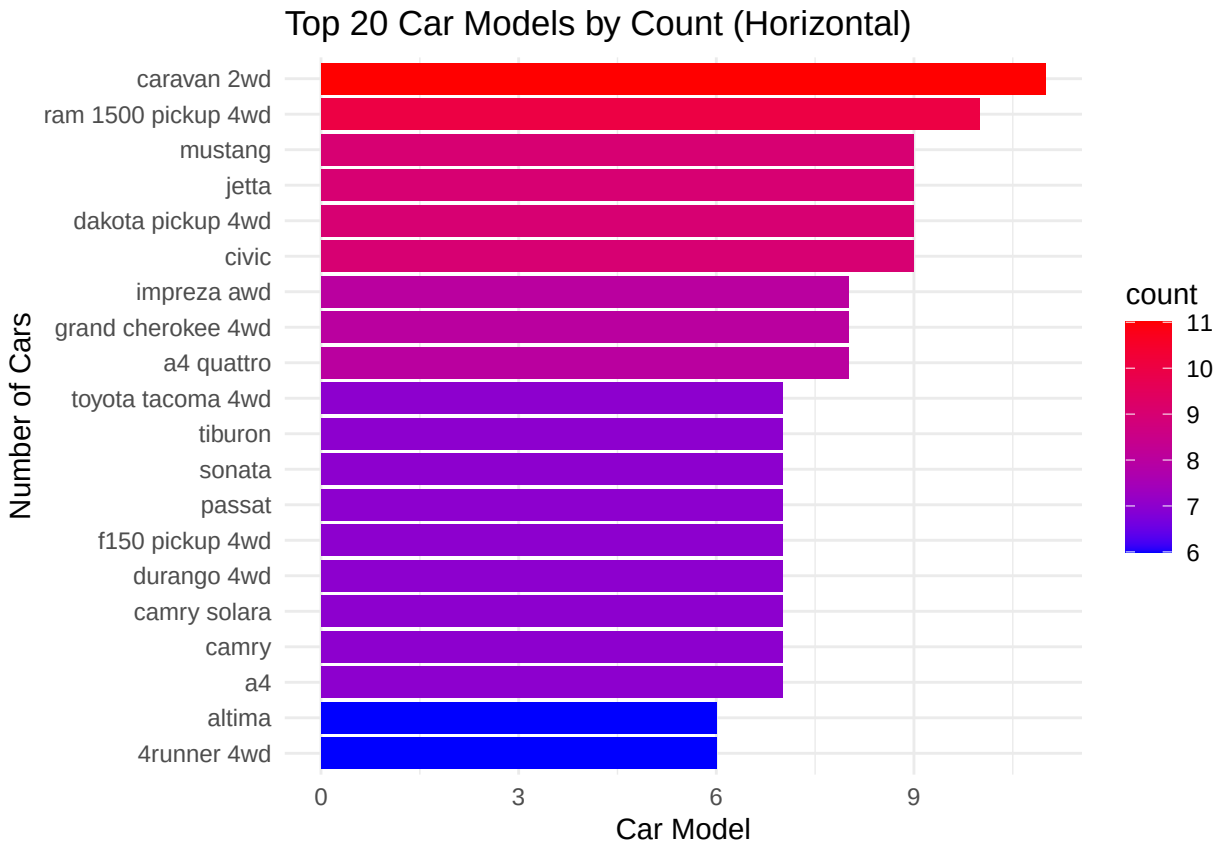
a. Plot using geom_bar() using the top 20 observations only. The graphs shoudl have a title, labels and colors. Show code and results.

```
top_20 <- head(car_counts, 20)

ggplot(top_20, aes(x = reorder(model, count), y = count, fill = count)) +
  geom_bar(stat = "identity") +
  labs(title = "Top 20 Car Models by Count",
       x = "Car Model",
       y = "Number of Cars") +
  scale_fill_gradient(low = "blue", high = "red") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



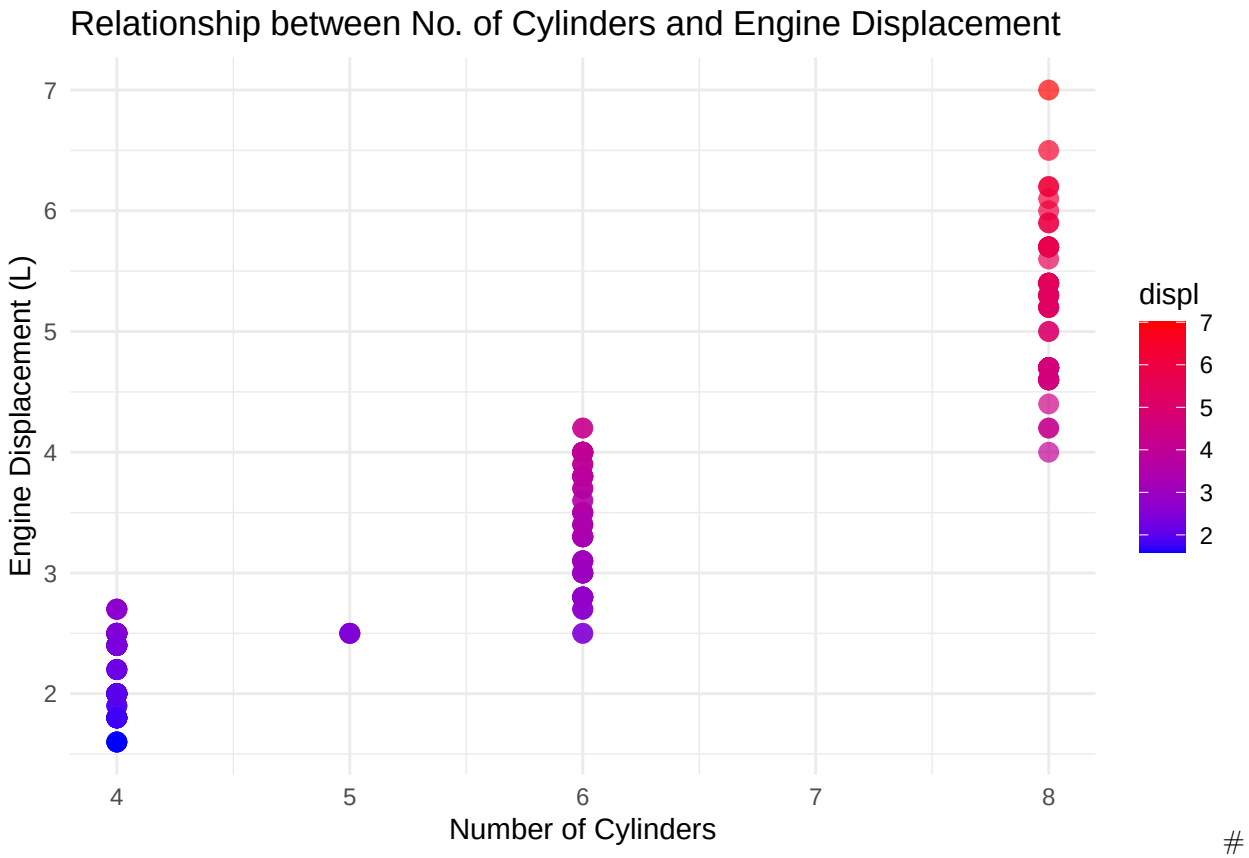
b. Plot using the geom_bar() + coord_flip() just like what is shown below. Show codes and its result.

```
ggplot(top_20, aes(x = reorder(model, count), y = count, fill = count)) +
  geom_bar(stat = "identity") +
  labs(title = "Top 20 Car Models by Count (Horizontal)",
       x = "Number of Cars",
       y = "Car Model") +
  scale_fill_gradient(low = "blue", high = "red") +  # Color gradient
  theme_minimal() +
  coord_flip()
```



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".

```
ggplot(mpg_data, aes(x = cyl, y = displ, color = displ)) +
  geom_point(size = 3, alpha = 0.7) +  # Adjust point size and transparency
  labs(title = "Relationship between No. of Cylinders and Engine Displacement",
       x = "Number of Cylinders",
       y = "Engine Displacement (L)") +
  scale_color_gradient(low = "blue", high = "red") +
  theme_minimal()
```

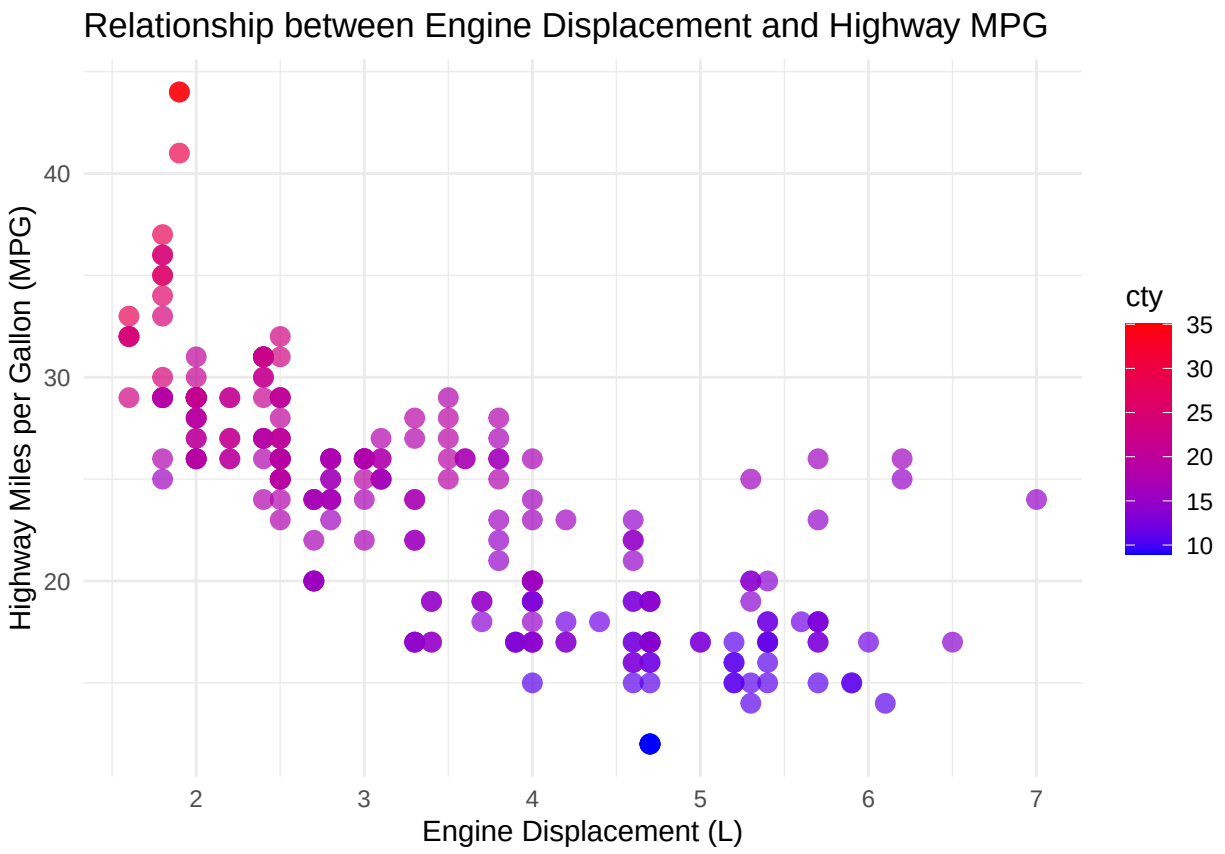


#

a. How would you describe its relationship? Show the codes and its result. The more number of cylinders the higher the displacement. It shows a positive relationship.

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_data, 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 Miles per Gallon (MPG)") +
  scale_color_gradient(low = "blue", high = "red") +
  theme_minimal()
```



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

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

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

```
length(traffic_data)
```

```
## [1] 4
```

```
variable_names <- names(traffic_data)

variable_names
```

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

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

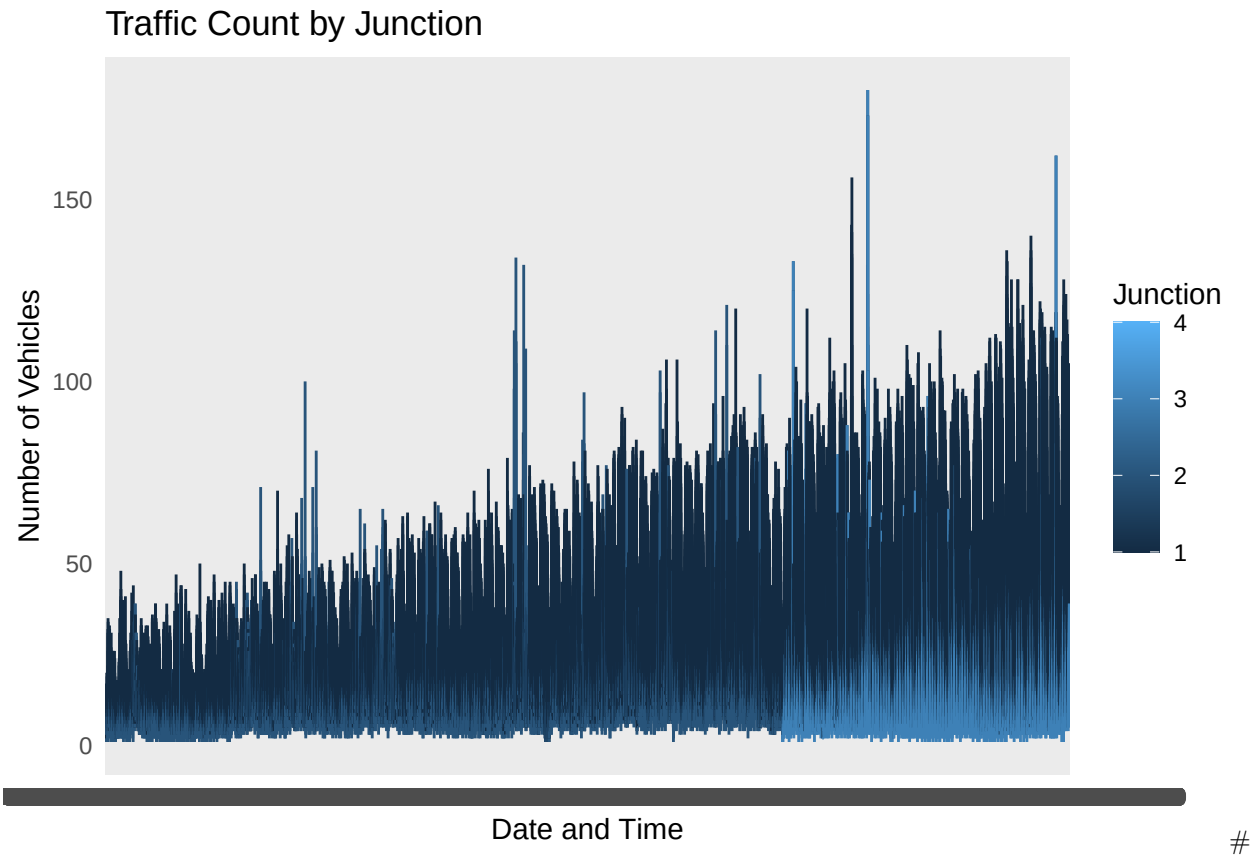
```
unique_junctions <- unique(traffic_data$Junction)

junctions_dataframes <- list()
```

```r
for (junction in unique_junctions) {
  junctions_dataframes[[junction]] <- traffic_data %>%
    filter(Junction == junction)
}

head(junctions_dataframes[[1]])
```

```
##              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
```

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

```r
ggplot(traffic_data, aes(x = DateTime, y = Vehicles, color = Junction)) +
  geom_line() +
  labs(title = "Traffic Count by Junction",
       x = "Date and Time",
       y = "Number of Vehicles") +
  theme_minimal() +
  theme(legend.position = "right")
```



#

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

```
library(readxl)

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

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)

num_observations
```

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

```
num_columns
```

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

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

```
alexa_data$rating <- as.numeric(as.character(alexa_data$rating))
alexa_data$verified_reviews <- as.numeric(as.character(alexa_data$verified_reviews))
```

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

```
sum(is.na(alexa_data$rating))
```

```
## [1] 0
```

```
sum(is.na(alexa_data$verified_reviews))
```

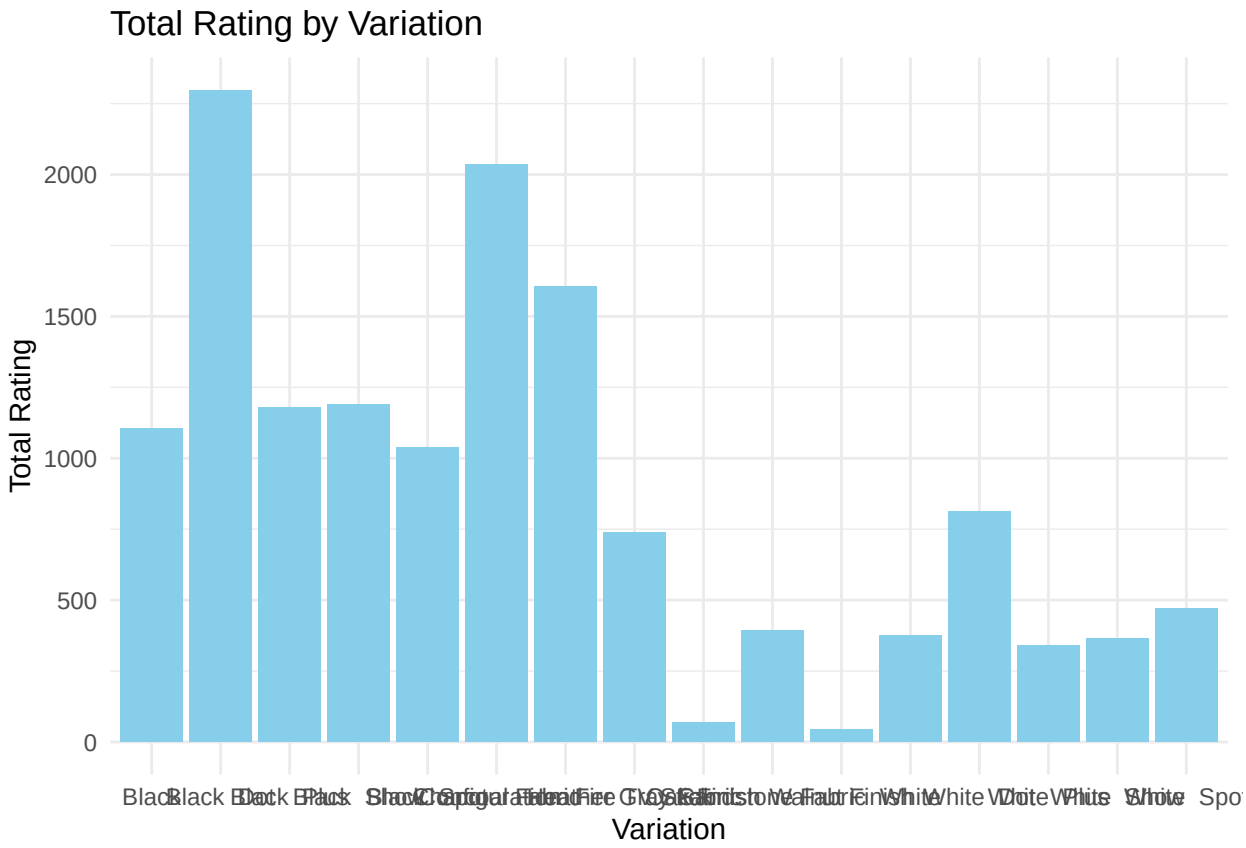
```
## [1] 3150
```

```
variation_totals <- alexa_data %>%
  group_by(variation) %>%
  summarize(Total_Rating = sum(rating, na.rm = TRUE),
            Total_Verified_Reviews = sum(verified_reviews, na.rm = TRUE))

print(variation_totals)
```

```
## # A tibble: 16 x 3
##    variation                    Total_Rating Total_Verified_Reviews
##    <chr>                               <dbl>                  <dbl>
##  1 Black                                1105                      0
##  2 Black  Dot                          2298                      0
##  3 Black  Plus                         1180                      0
##  4 Black  Show                         1190                      0
##  5 Black  Spot                         1039                      0
##  6 Charcoal Fabric                     2034                      0
##  7 Configuration: Fire TV Stick        1607                      0
##  8 Heather Gray Fabric                  737                      0
##  9 Oak Finish                            68                      0
## 10 Sandstone Fabric                     392                      0
## 11 Walnut Finish                         44                      0
## 12 White                                377                      0
## 13 White  Dot                           814                      0
## 14 White  Plus                          340                      0
## 15 White  Show                          364                      0
## 16 White  Spot                          470                      0
```

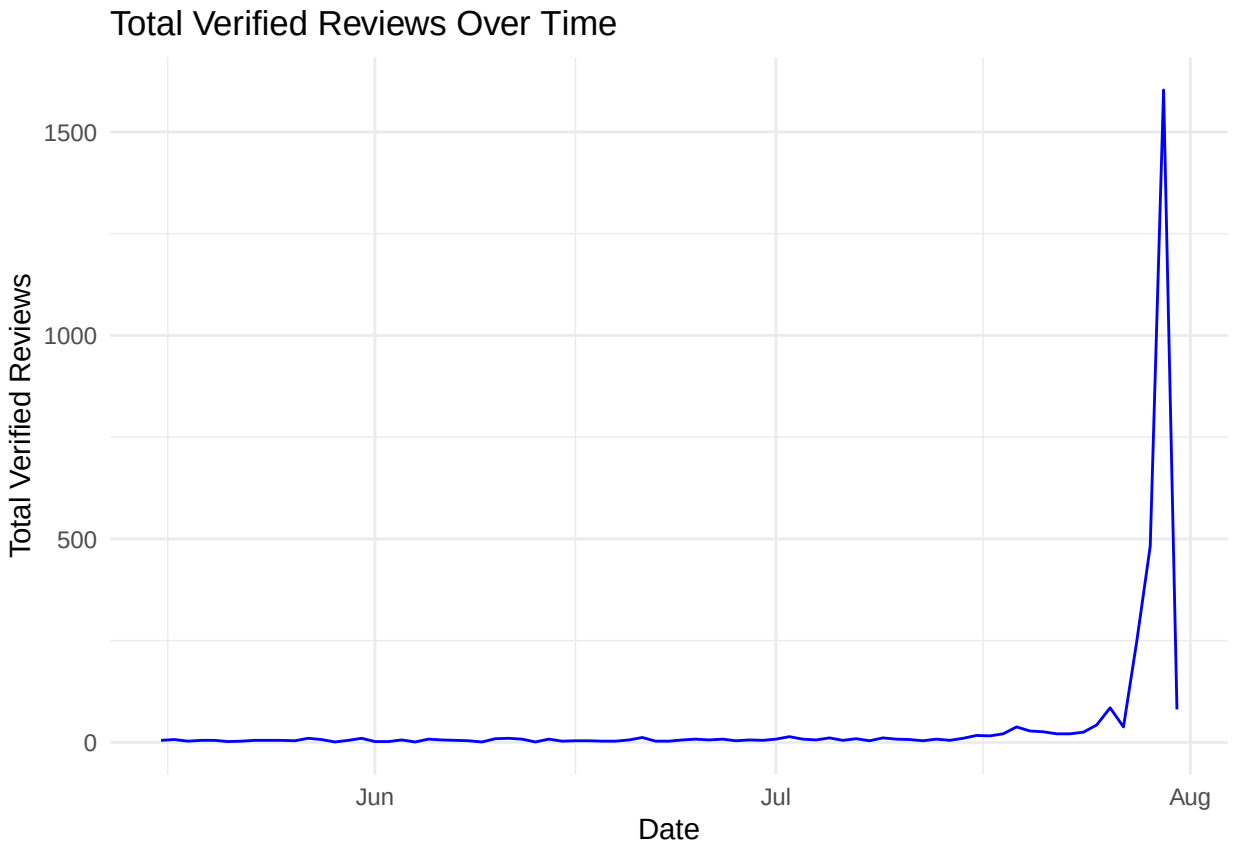
c. Plot the variations using the ggplot() function. What did you observe? Complete the details of the graph. Show solution and answer.

```
ggplot(variation_totals, aes(x = variation, y = Total_Rating)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Total Rating by Variation",
       x = "Variation",
       y = "Total Rating") +
  theme_minimal()
```



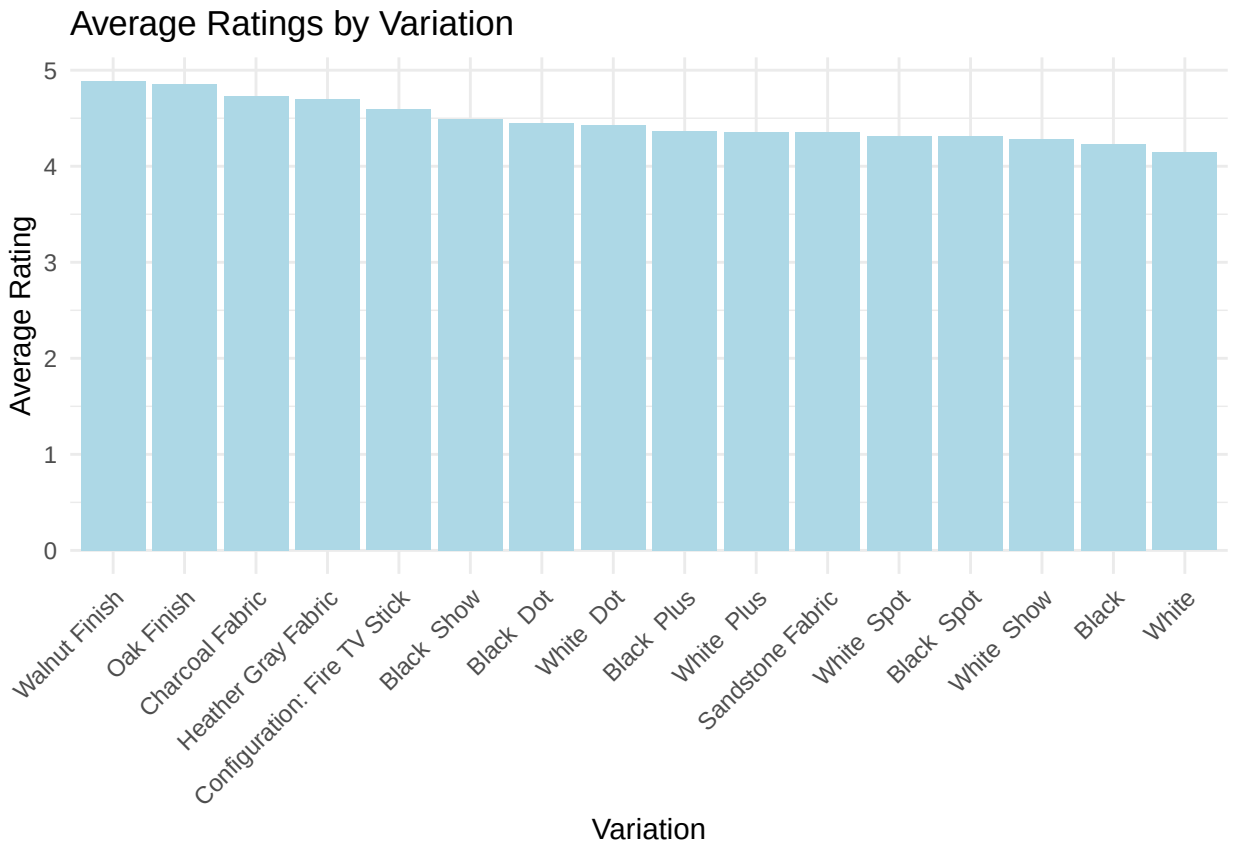
The graph shows that each bar corresponds to the total rating for a specific variation. Although the names on the x axis overlaps. # d. Plot a geom_line() with the date and the number of verified reviews. Complete the details of the graphs. Show your answer and solution.

```
reviews_by_date <- alexa_data %>%
  group_by(date) %>%  # Group by date
  summarize(Total_Verified_Reviews = n())

ggplot(reviews_by_date, aes(x = date, y = Total_Verified_Reviews)) +
  geom_line(color = "blue") +
  labs(title = "Total Verified Reviews Over Time", x = "Date", y = "Total 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.

```
average_ratings <- alexa_data %>%
  group_by(variation) %>%
  summarize(Average_Rating = mean(rating, na.rm = TRUE))

highest_rating_variation <- average_ratings %>%
  filter(Average_Rating == max(Average_Rating))

ggplot(average_ratings, aes(x = reorder(variation, -Average_Rating), y = Average_Rating)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  labs(title = "Average Ratings by Variation",
       x = "Variation",
       y = "Average Rating") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


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