RWorksheet_Gerona#4c

Mariel M. Gerona

#1a: Importing and Exploring the mpg Dataset

```
# Load necessary libraries
library(ggplot2)
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(tidyr)

# Load the mpg dataset
data("mpg")

# Display first few rows
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
```

#1b. Which variables are categorical?

```
categorical_vars <- mpg %>% select(where(is.character))
categorical_vars
```

```
## # A tibble: 234 x 6
##    manufacturer model      trans      drv   fl    class
##    <chr>        <chr>      <chr>      <chr> <chr> <chr>
##  1 audi         a4         auto(l5)   f     p     compact
##  2 audi         a4         manual(m5) f     p     compact
##  3 audi         a4         manual(m6) f     p     compact
##  4 audi         a4         auto(av)   f     p     compact
##  5 audi         a4         auto(l5)   f     p     compact
```

```
##  6 audi         a4         manual(m5) f     p     compact
##  7 audi         a4         auto(av)   f     p     compact
##  8 audi         a4 quattro manual(m5) 4     p     compact
##  9 audi         a4 quattro auto(l5)   4     p     compact
## 10 audi         a4 quattro manual(m6) 4     p     compact
## # i 224 more rows
```

#1c. Which variables are continuous?

```
continuous_vars <- mpg %>% select(where(is.numeric))
continuous_vars
```

```
## # A tibble: 234 x 5
##    displ  year   cyl   cty   hwy
##    <dbl> <int> <int> <int> <int>
##  1   1.8  1999     4    18    29
##  2   1.8  1999     4    21    29
##  3   2    2008     4    20    31
##  4   2    2008     4    21    30
##  5   2.8  1999     6    16    26
##  6   2.8  1999     6    18    26
##  7   3.1  2008     6    18    27
##  8   1.8  1999     4    18    26
##  9   1.8  1999     4    16    25
## 10   2    2008     4    20    28
## # i 224 more rows
```

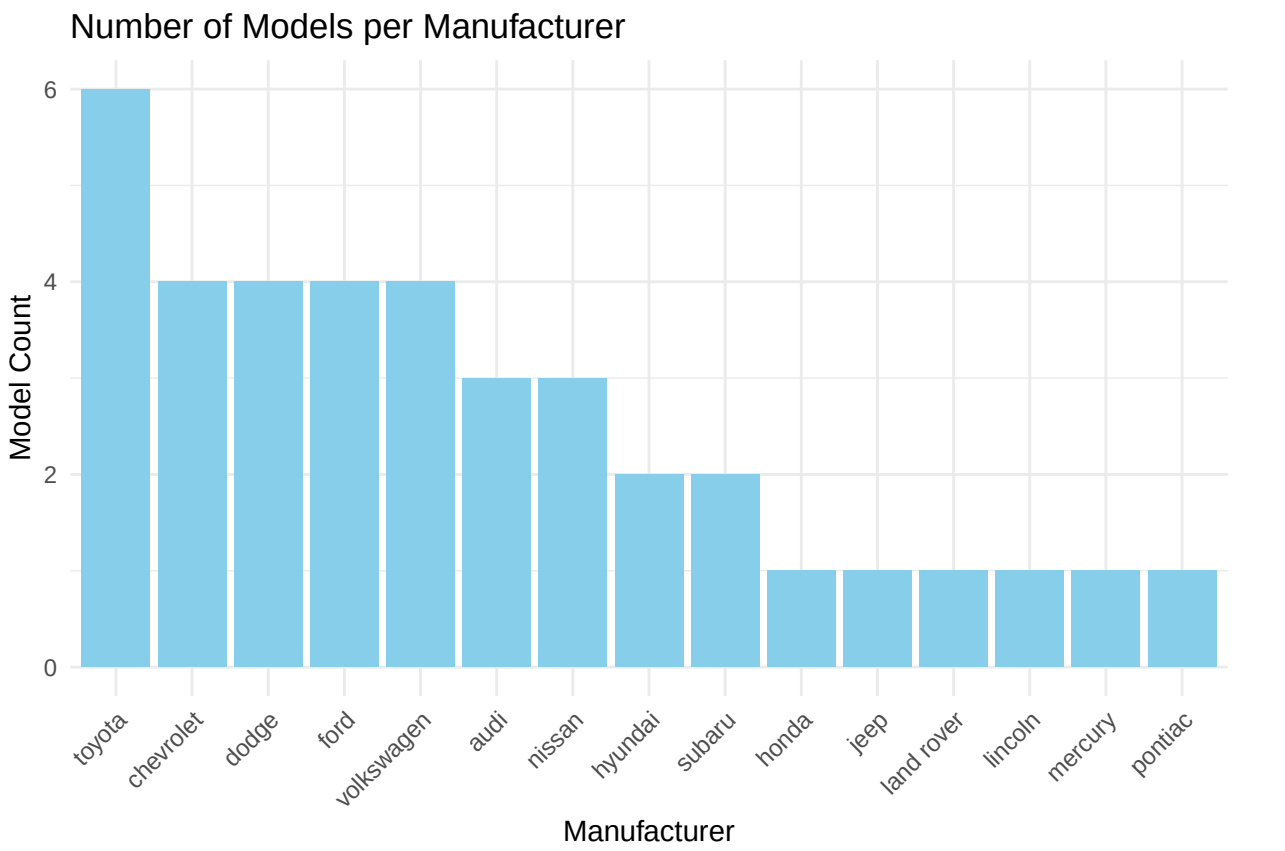
#2a. Group Manufacturers and Unique Models

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

manufacturer_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]>
```

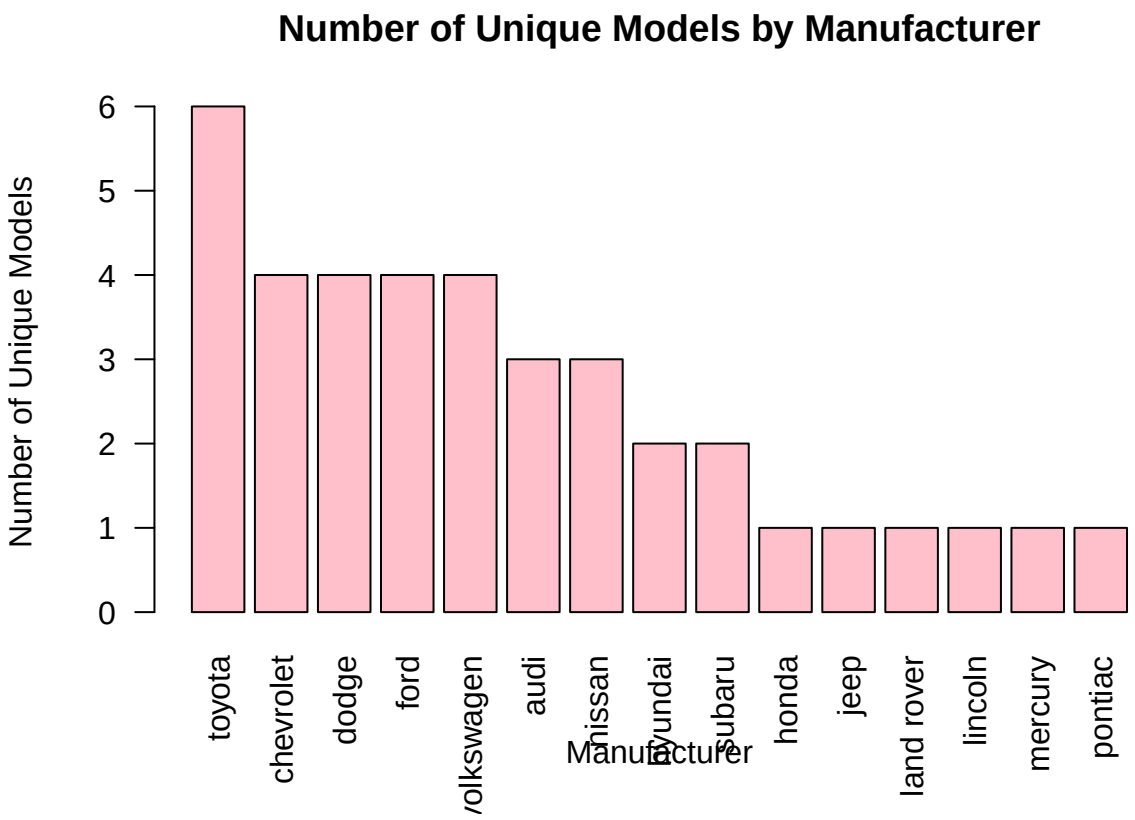
#2. Plot Manufacturer Model Counts

```r
# Count the models per manufacturer
manufacturer_counts <- mpg %>%
  group_by(manufacturer) %>%
  summarise(models = n_distinct(model))

# Plot the counts of models per manufacturer
ggplot(manufacturer_counts, aes(x = reorder(manufacturer, -models), y = models)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  theme_minimal() +
  labs(title = "Number of Models per Manufacturer", x = "Manufacturer", y = "Model Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

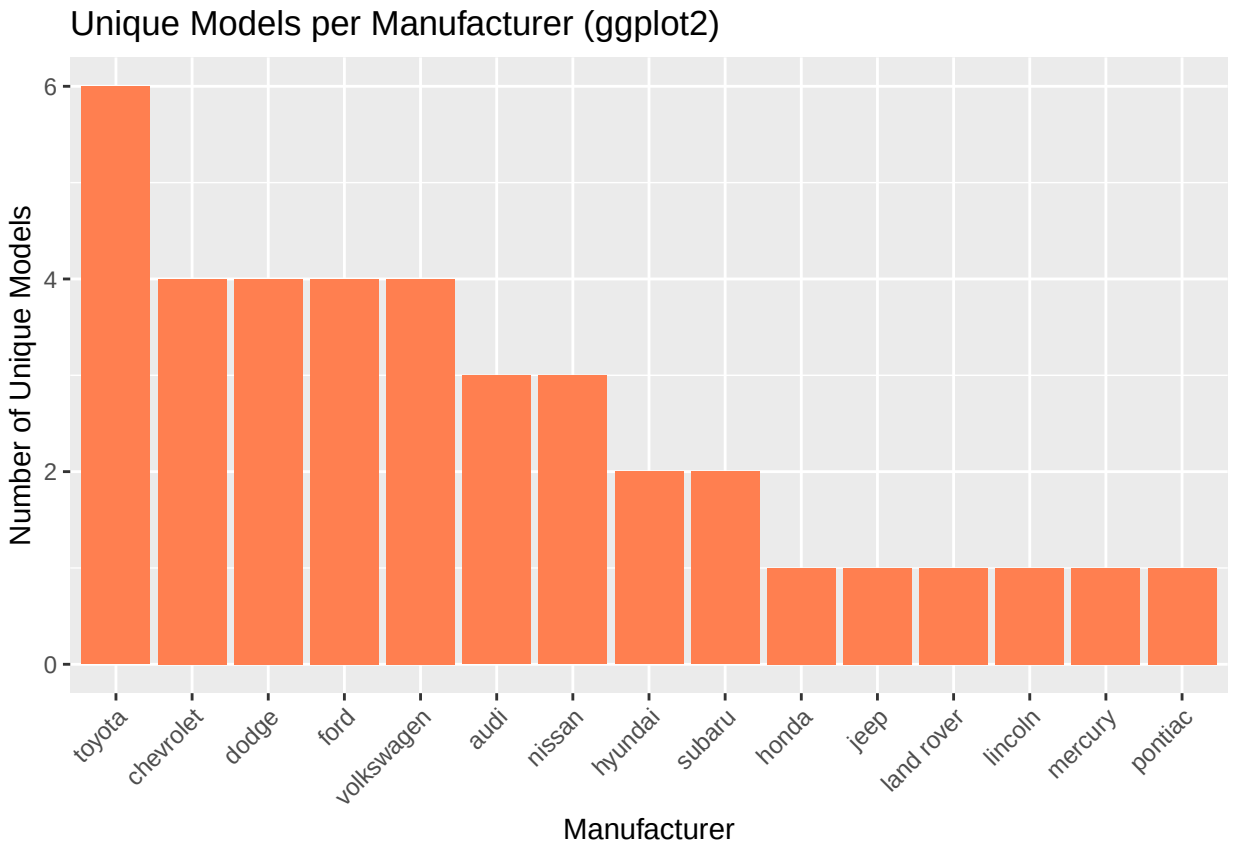


##2b. Graph Unique Models Using plot() and ggplot()

```r
sorted_data <- manufacturer_counts[order(manufacturer_counts$models, decreasing = TRUE),]
barplot(sorted_data$models, names.arg = sorted_data$manufacturer, las = 2, col = "Pink",
main = "Number of Unique Models by Manufacturer",
xlab = "Manufacturer", ylab = "Number of Unique Models")
```

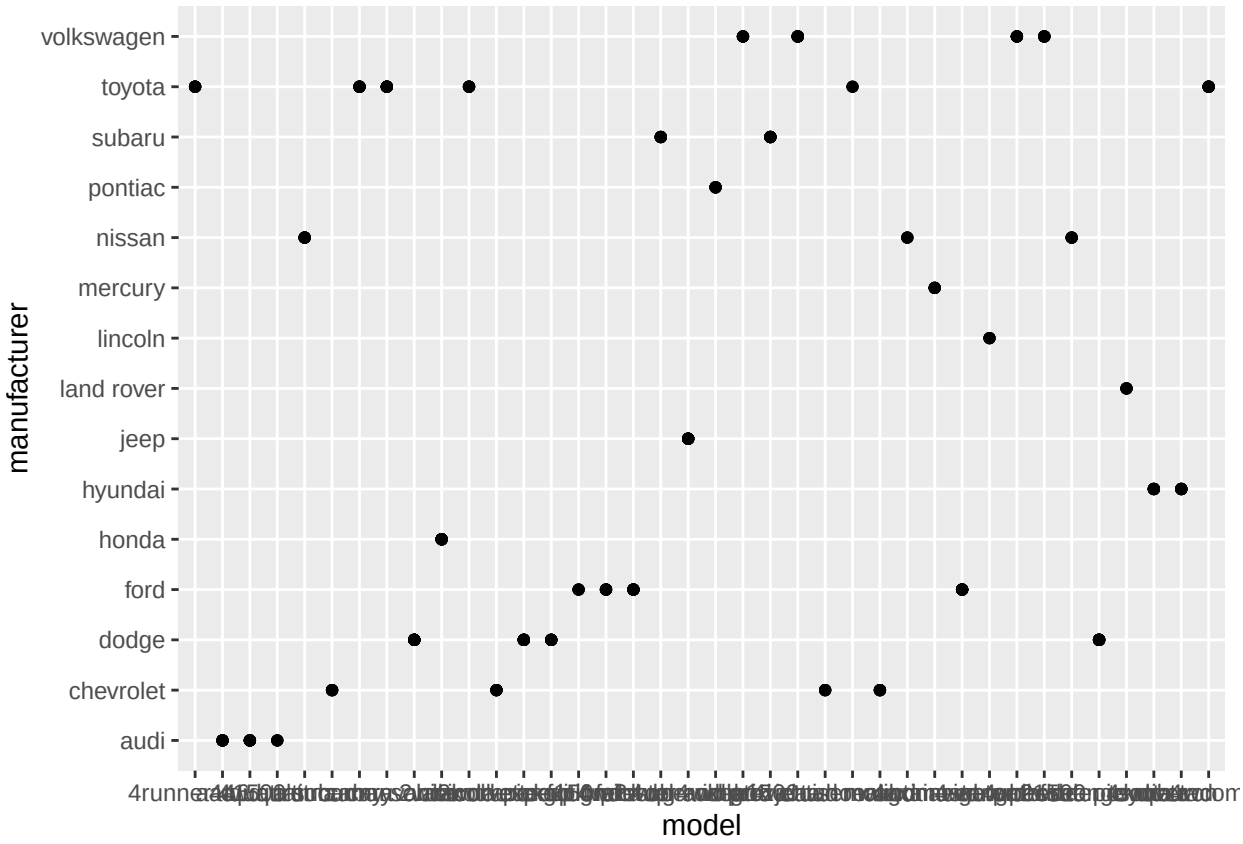



```
# Using ggplot2
ggplot(manufacturer_counts, aes(x = reorder(manufacturer, -models), y = models)) +
  geom_bar(stat = "identity", fill = "coral") +
  labs(title = "Unique Models per Manufacturer (ggplot2)",
       x = "Manufacturer",
       y = "Number of Unique Models") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



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

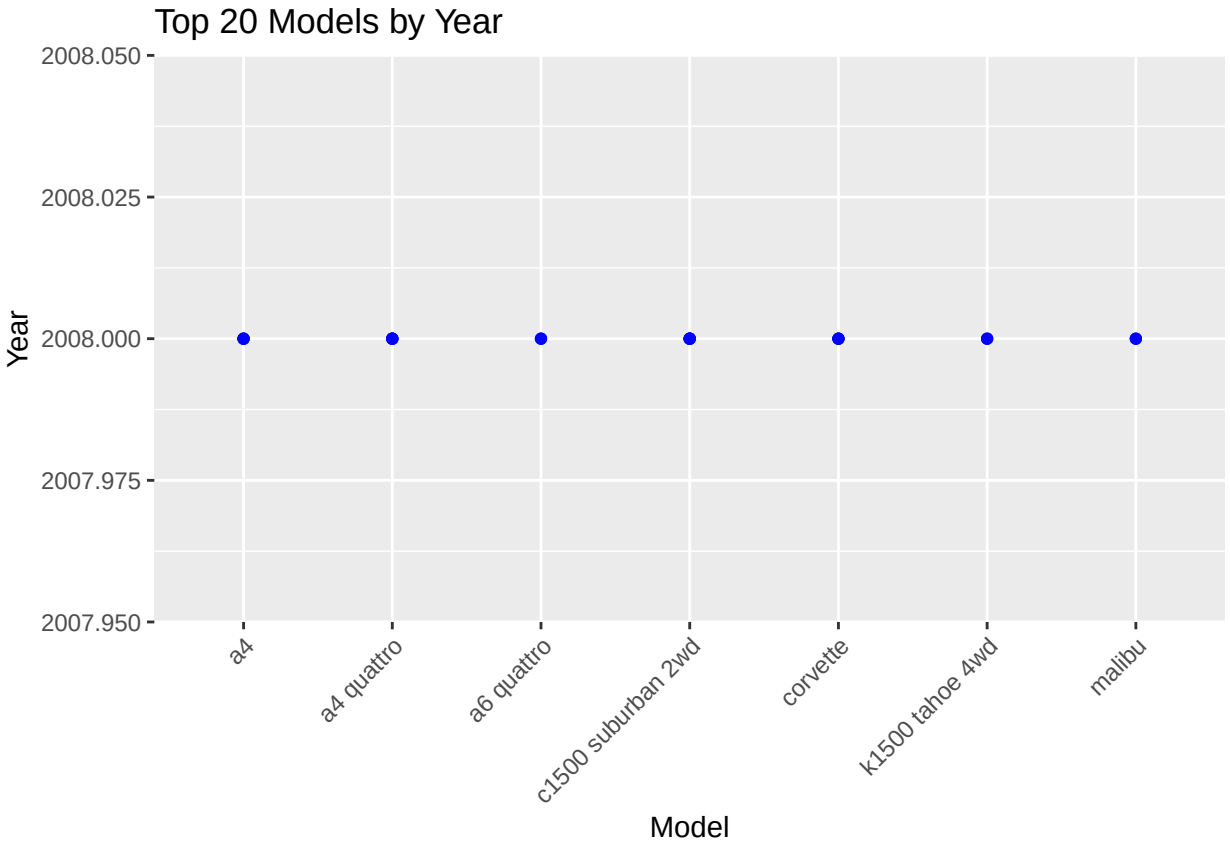
```
ggplot(mpg, aes(x = model, y = manufacturer)) + geom_point()
```



#2a. This plot shows the relationship between model and manufacturer. However, it may not be very informative due to the large number of model names, which can overlap and create a cluttered appearance, making the plot difficult to interpret.

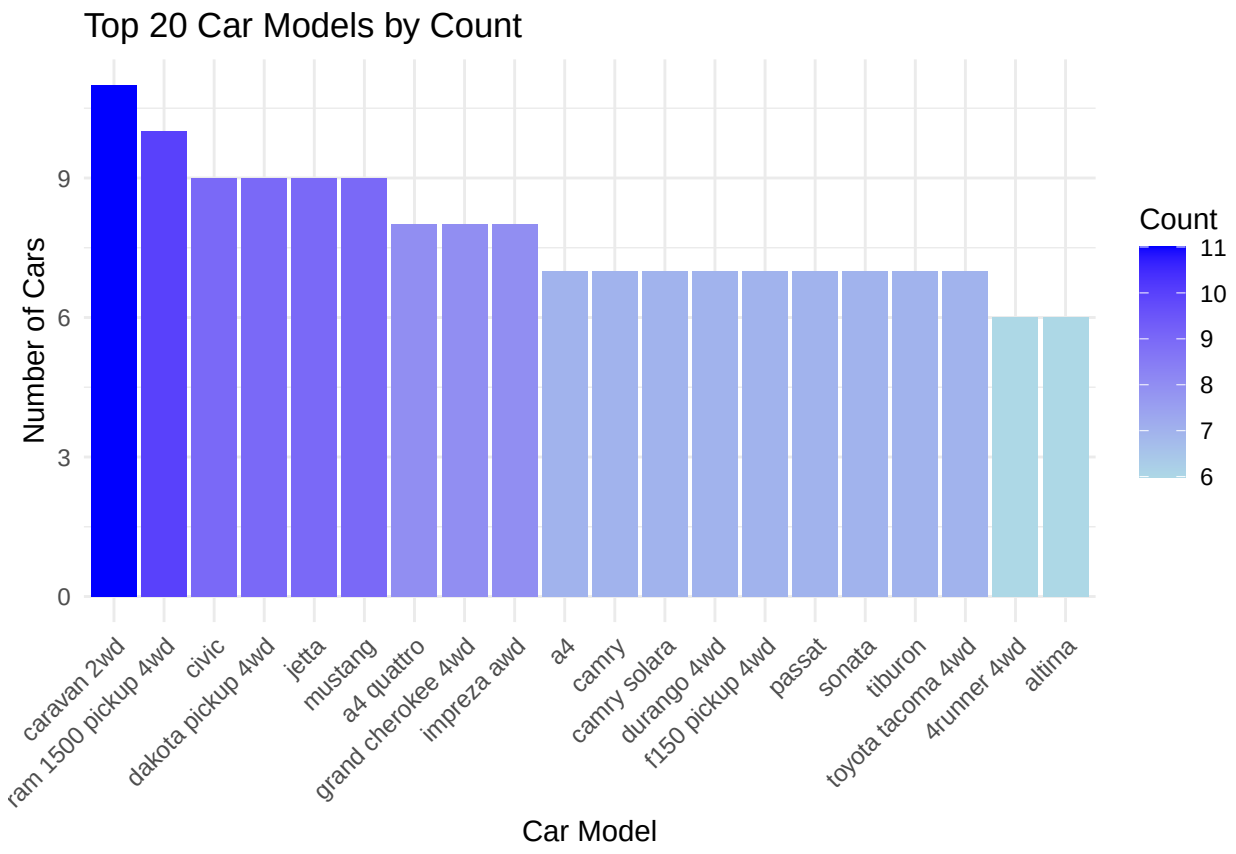
#3. Plot the model and the year using ggplot(). Use only the top 20 observations.

```
# Get the top 20 observations based on year
top_20_mpg <- mpg %>%
  arrange(desc(year)) %>%
  head(20)

# Create the plot
ggplot(top_20_mpg, aes(x = model, y = year)) +
  geom_point(color = "blue") +
  labs(title = "Top 20 Models by Year",
       x = "Model",
       y = "Year") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



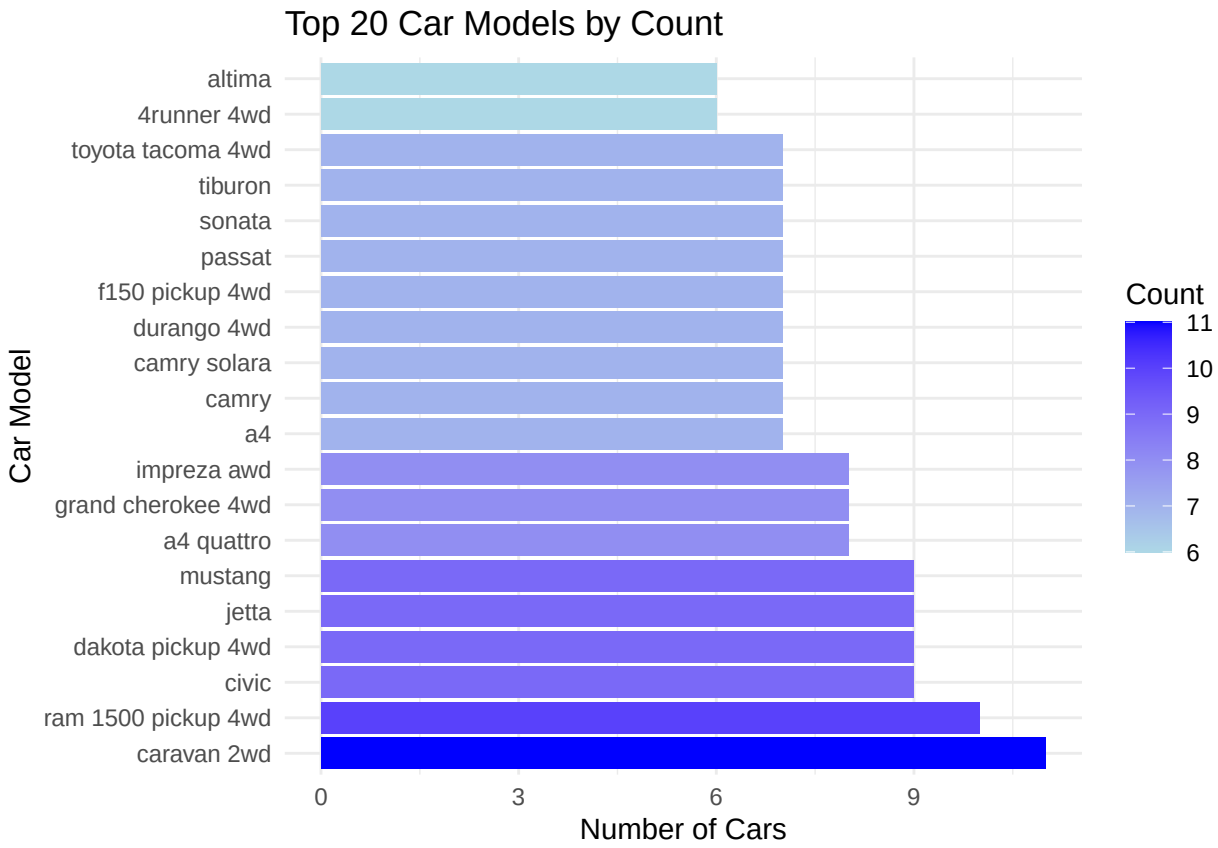
#4a. Plot using geom_bar() using the top 20 observations only. The graphs should have a title, labels, and colors.

```
# Group the data by model and count the number of cars per model
model_counts <- mpg %>%
  group_by(model) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  head(20)

# Plot the top 20 models with geom_bar()
ggplot(model_counts, 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",
       fill = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_gradient(low = "lightblue", high = "blue")
```



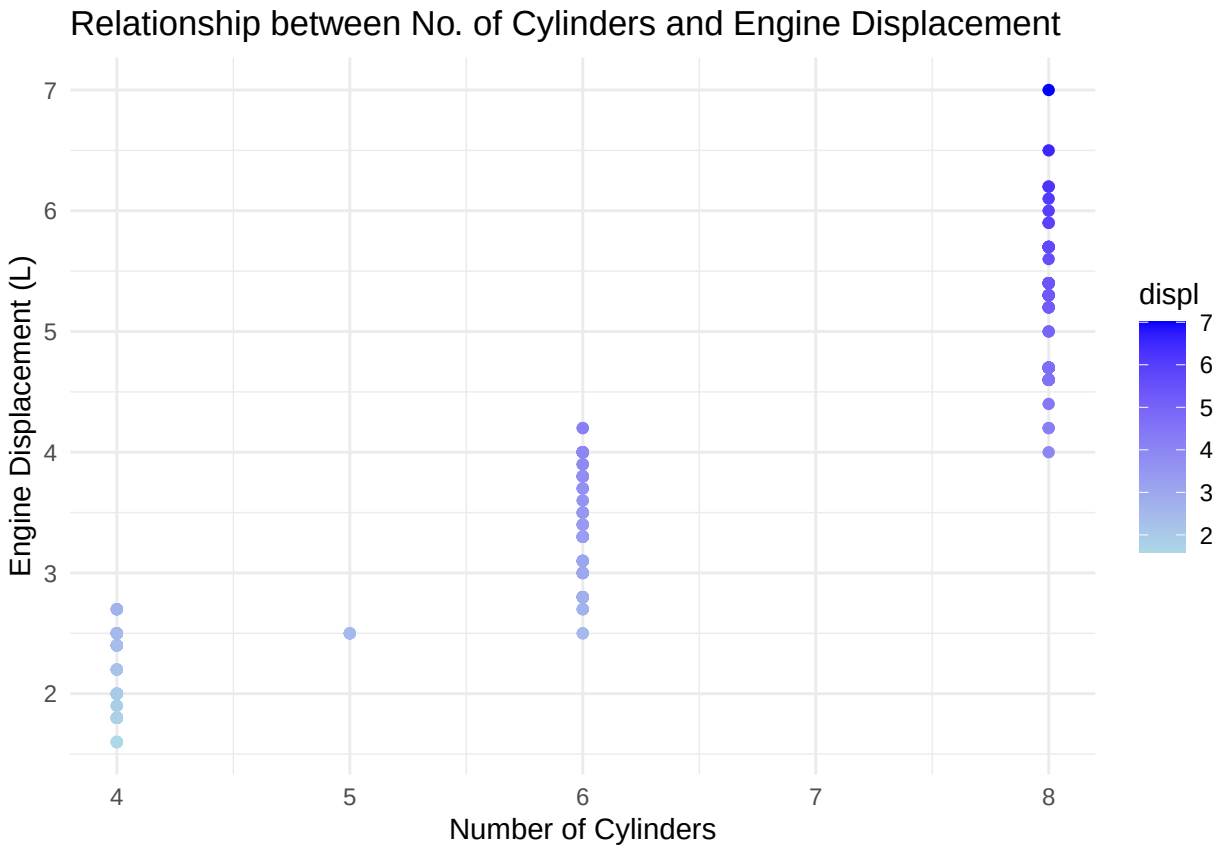
#4b. Plot using the geom_bar() + coord_flip().

```
# Plot the top 20 models with geom_bar() and coord_flip()
ggplot(model_counts, aes(x = reorder(model, -count), y = count, fill = count)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 20 Car Models by Count",
       x = "Car Model",
       y = "Number of Cars",
       fill = "Count") +
  theme_minimal() +
  scale_fill_gradient(low = "lightblue", high = "blue")
```



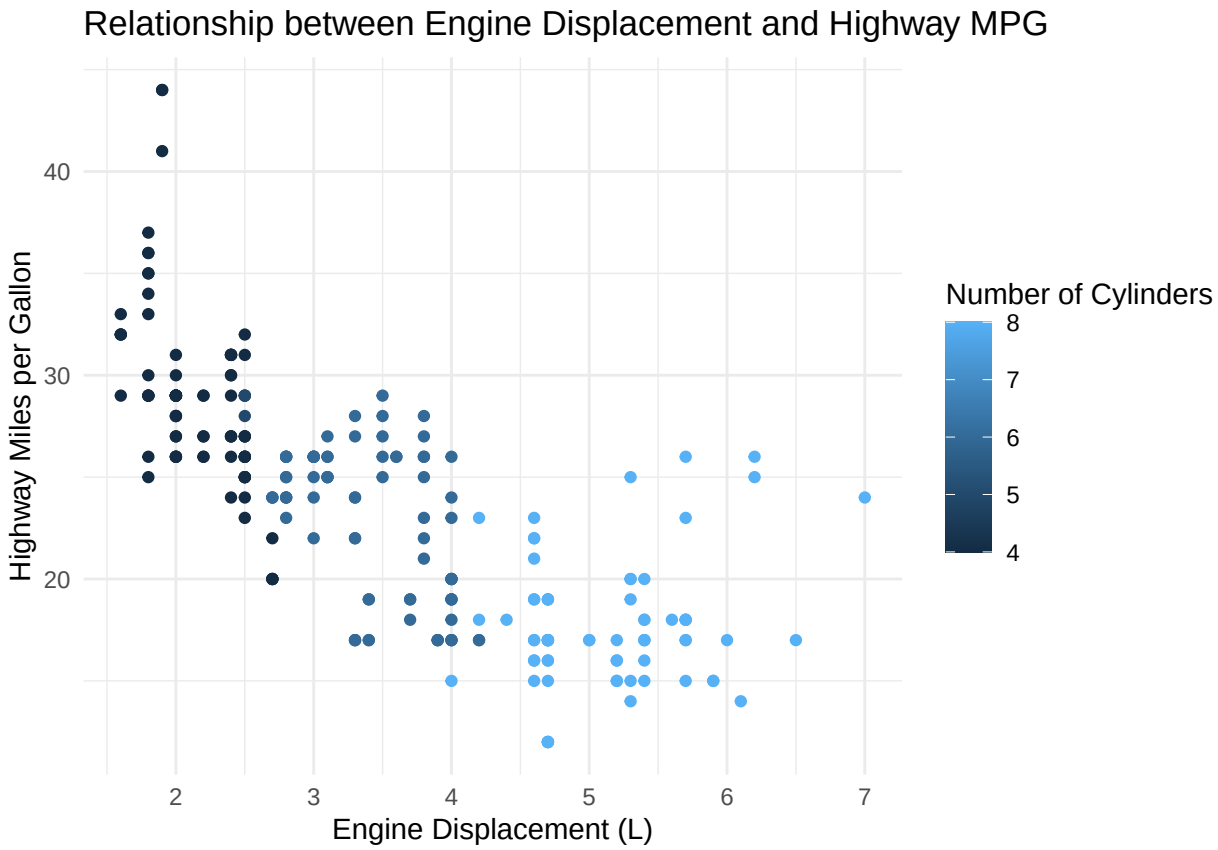
#5. Plot the relationship between cyl - number of cylinders and displ - engine displacement using geom_point with aesthetic color = engine displacement. Title should be "Relationship between No. of Cylinders and Engine Displacement".

```
# Create the plot
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 (L)") +
  theme_minimal() +
  scale_color_gradient(low = "lightblue", high = "blue")
```



#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.

```
# Create the plot
ggplot(mpg, aes(x = displ, y = hwy, color = cyl)) +
  geom_point() +
  labs(title = "Relationship between Engine Displacement and Highway MPG",
       x = "Engine Displacement (L)",
       y = "Highway Miles per Gallon",
       color = "Number of Cylinders") +
  theme_minimal()
```



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

```
# Load necessary library for reading Excel files
library(readxl)

# Import the traffic.xlsx file
traffic_data <- read_excel("/cloud/project/Worksheet#4/Worksheet#4c/traffic.xlsx")

# Check the number of observations and variables
n_obs <- nrow(traffic_data)  # Number of observations
variables <- colnames(traffic_data)  # Variables in the dataset

# Show results
n_obs
```

```
## [1] 10
```

```
variables
```

```
## [1] "Date"       "Junction_A" "Junction_B" "Junction_C"
```

#6b. Subset the traffic dataset into junctions.

```
# Subset the dataset based on the identified column (e.g., location)
junctions_data <- traffic_data %>%
  filter(!is.na(location))  # Replace 'location' with the actual column name
```

```
## Warning: There was 1 warning in `filter()`.
## i In argument: `!is.na(location)`.
## Caused by warning in `is.na()`:
```

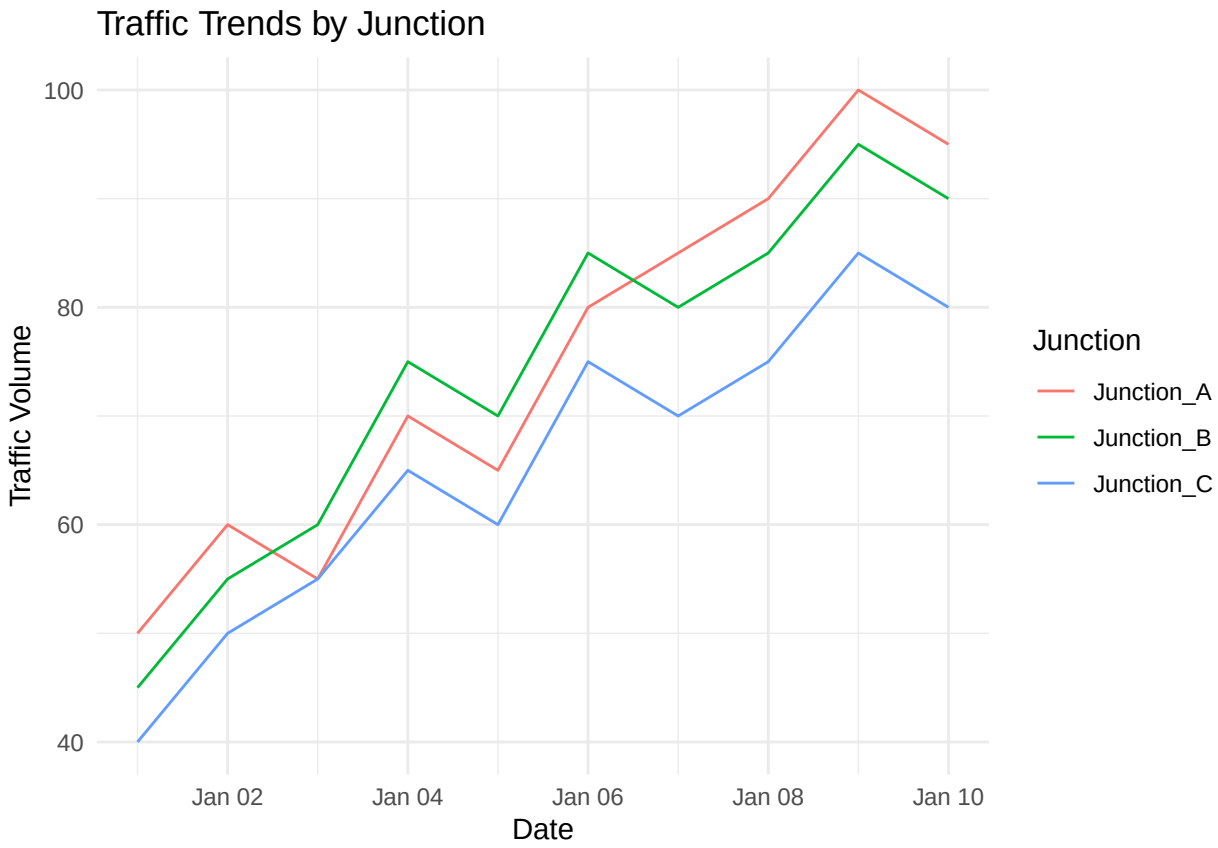
```
## ! is.na() applied to non-(list or vector) of type 'closure'
```

```
# View the first few rows of the junctions dataset
head(junctions_data)
```

```
## # A tibble: 6 x 4
##   Date                Junction_A Junction_B Junction_C
##   <dttm>                   <dbl>      <dbl>      <dbl>
## 1 2024-01-01 00:00:00         50         45         40
## 2 2024-01-02 00:00:00         60         55         50
## 3 2024-01-03 00:00:00         55         60         55
## 4 2024-01-04 00:00:00         70         75         65
## 5 2024-01-05 00:00:00         65         70         60
## 6 2024-01-06 00:00:00         80         85         75
```

#6c. Plot each junction using geom_line.

```
# Reshape data to long format using pivot_longer
traffic_data_long <- traffic_data %>%
  pivot_longer(cols = starts_with("Junction"),  # Select columns starting with "Junction"
               names_to = "Junction",            # New column to hold the junction names
               values_to = "Traffic")           # New column to hold the traffic data

# Plotting the traffic data using geom_line
ggplot(traffic_data_long, aes(x = Date, y = Traffic, color = Junction, group = Junction)) +
  geom_line() +
  labs(title = "Traffic Trends by Junction",
       x = "Date",
       y = "Traffic Volume") +
  theme_minimal()
```



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

```
# Load necessary libraries
library(readxl)

# Set the correct file path
file_path <- "/cloud/project/Worksheet#4/Worksheet#4c/alexa_file.xlsx"

# Import the alexa_file.xlsx
alexa_file <- read_excel(file_path)
```

#7a. How many observations does alexa_file has? What about the number of columns?

```
# Check the number of observations and columns
num_observations <- nrow(alexa_file)
num_columns <- ncol(alexa_file)

# Show results
num_observations
```

```
## [1] 10
```

```
num_columns
```

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

#7b. Group the variations and get the total of each variation using dplyr.

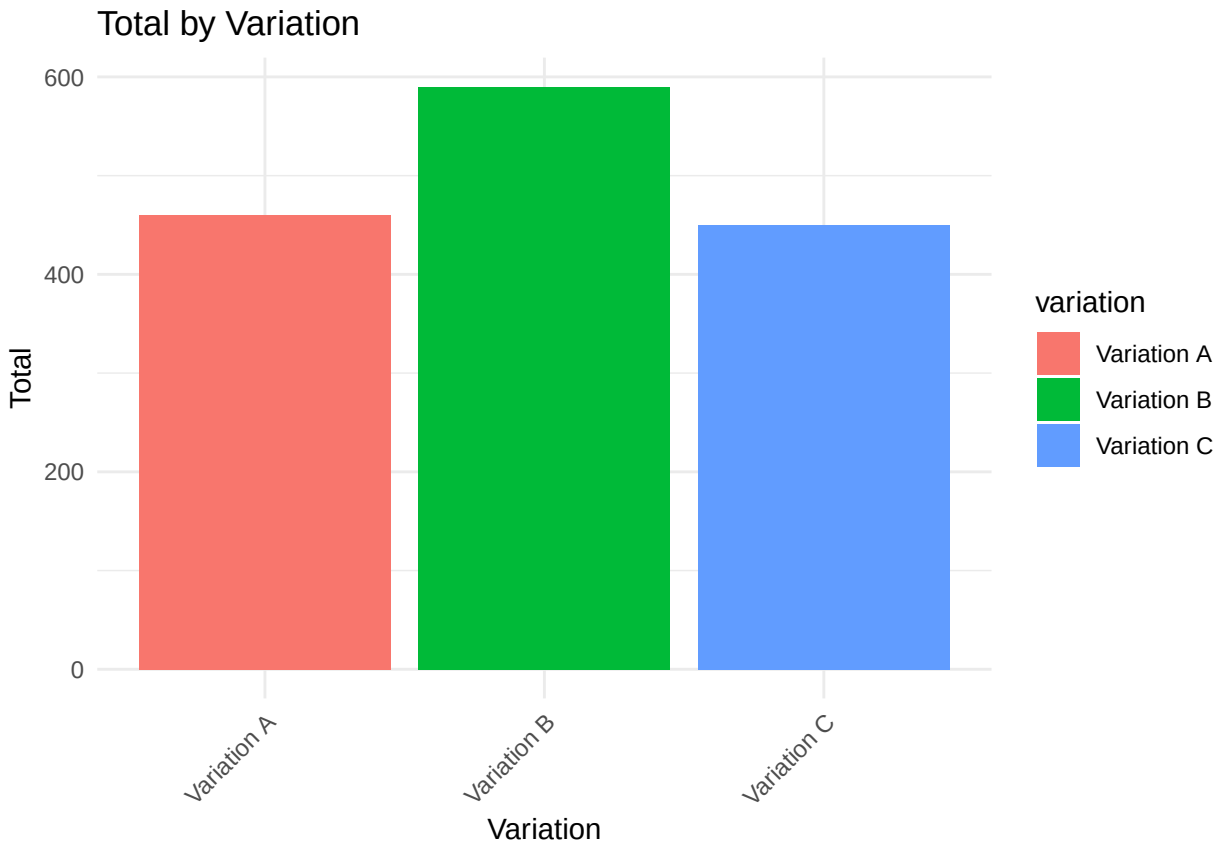
```
# Group by variations and get the total of each variation
variation_totals <- alexa_file %>%
  group_by(variation) %>%
```

```
  summarise(total = sum(value, na.rm = TRUE))

# Show the results
variation_totals
```

```
## # A tibble: 3 x 2
##   variation   total
##   <chr>       <dbl>
## 1 Variation A   460
## 2 Variation B   590
## 3 Variation C   450
```

#7c. Plot the variations using ggplot2.

```
ggplot(variation_totals, aes(x = variation, y = total, fill = variation)) +
  geom_bar(stat = "identity") +
  labs(title = "Total by Variation",
       x = "Variation",
       y = "Total") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



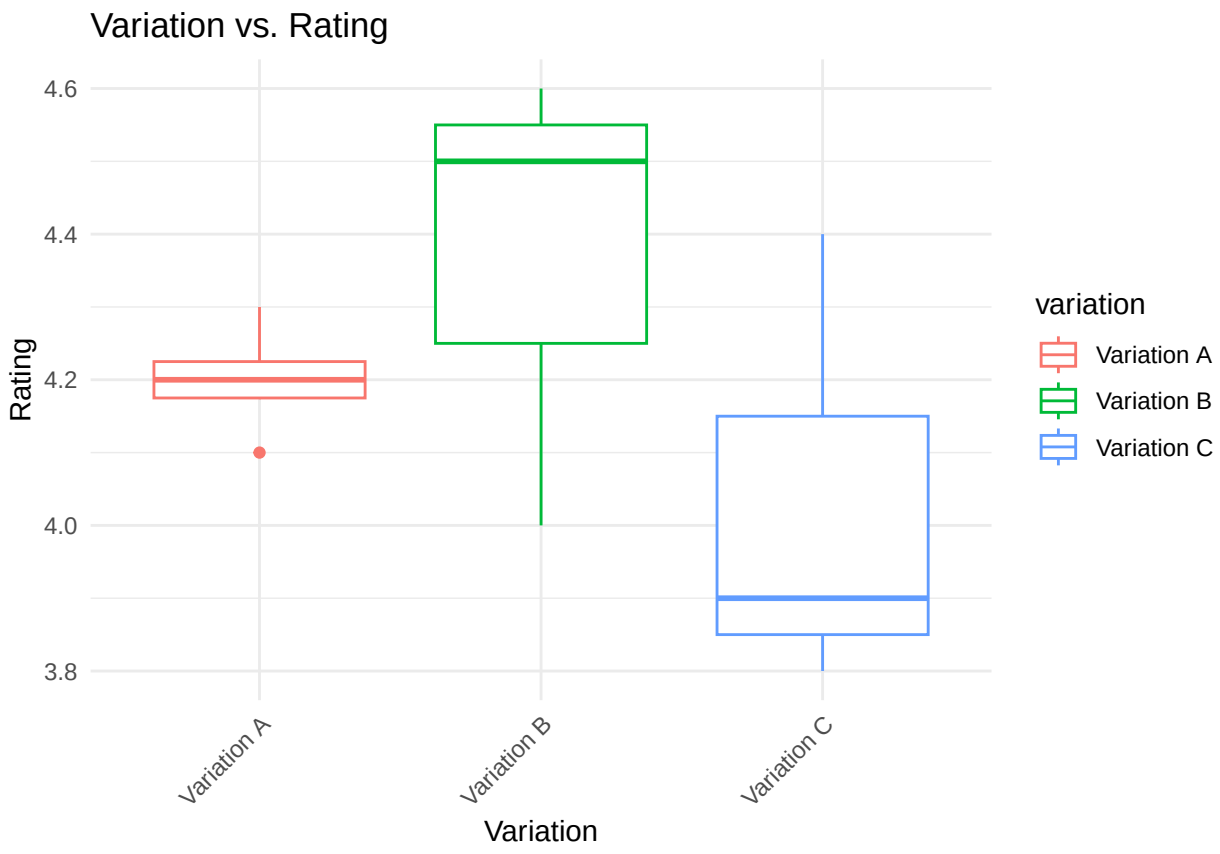
#7d. Plot a geom_line() with the date and the number of verified reviews.

```
ggplot(alexa_file, aes(x = date, y = verified_reviews)) +
  geom_line(color = "blue") +
  labs(title = "Verified Reviews Over Time",
       x = "Date",
       y = "Number of Verified Reviews") +
```

```
  theme_minimal()
```

#7e. Get the relationship of variations and ratings.

```
ggplot(alexa_file, aes(x = variation, y = rating, color = variation)) +
  geom_boxplot() +
  labs(title = "Variation vs. Rating",
       x = "Variation",
       y = "Rating") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Variation vs. Rating
4.6
4.4
4.2
4.0
3.8
Rating
Variation A
Variation B
Variation C
Variation
variation
Variation A
Variation B
Variation C