RWorksheet_calvario#4c.Rmd

Jolien

2024-11-19

1. Use the dataset mpg

a. Solutions 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  12 variables:
##  $ X           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ 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. Which variables from mpg dataset are categorical?

The categorical variables in the mpg dataset are manufacturer, model, trans, drv, fl, and class. These represent distinct groups or categories, such as car brand, transmission type, drivetrain, fuel type, and car class.

c. Which are continuous variables?

The continuous variables in the mpg dataset are displ, cty, hwy, and cyl, as they represent measurable numerical values.

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

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

```r
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
```

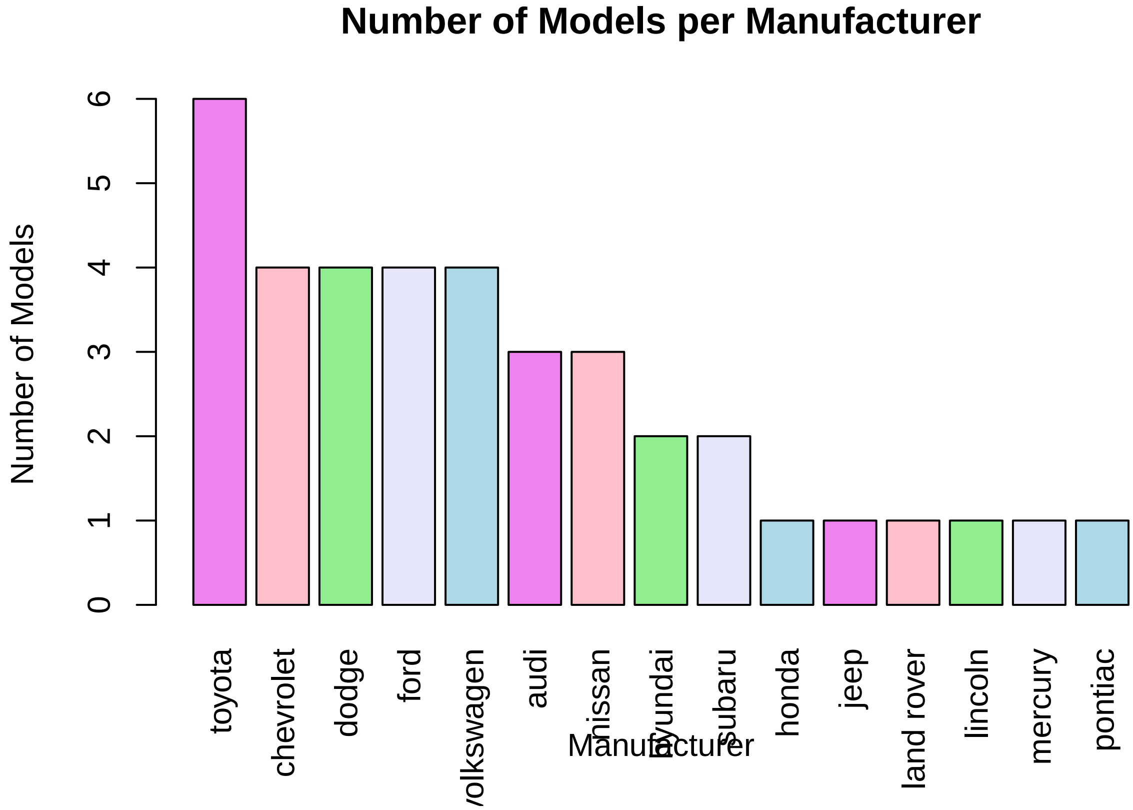
```
## # A tibble: 15 x 2
##    manufacturer model_num
##    <chr>            <int>
##  1 toyota               6
##  2 chevrolet            4
##  3 dodge                4
##  4 ford                 4
##  5 volkswagen           4
##  6 audi                 3
##  7 nissan               3
##  8 hyundai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

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

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

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

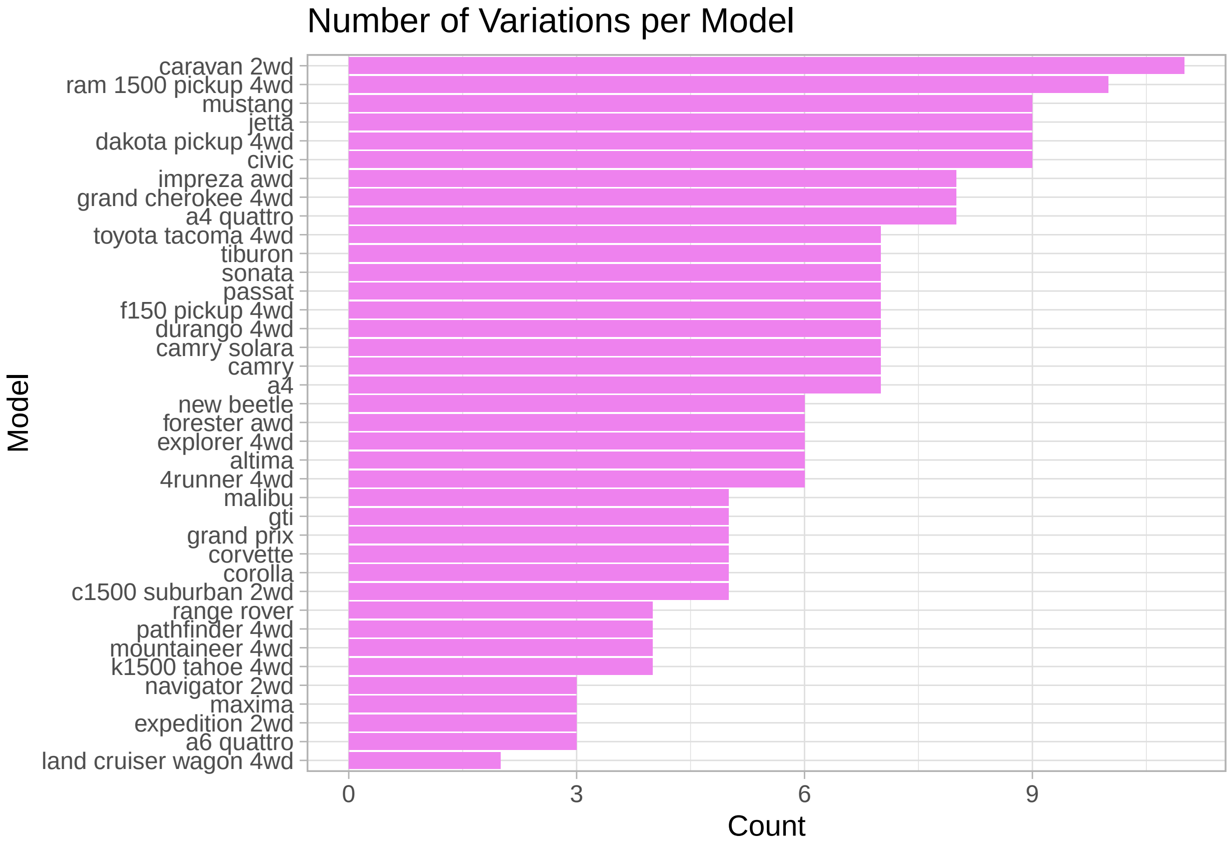
```
manufacturer_data <- setNames(
  manufacturer_model$model_num,
  manufacturer_model$manufacturer
  )

barplot(manufacturer_data,
        main = "Number of Models per Manufacturer",
        xlab = "Manufacturer",
        ylab = "Number of Models",
        col = c("violet", "pink", "lightgreen", "lavender", "lightblue"),
        las = 3)
```



```r
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
```

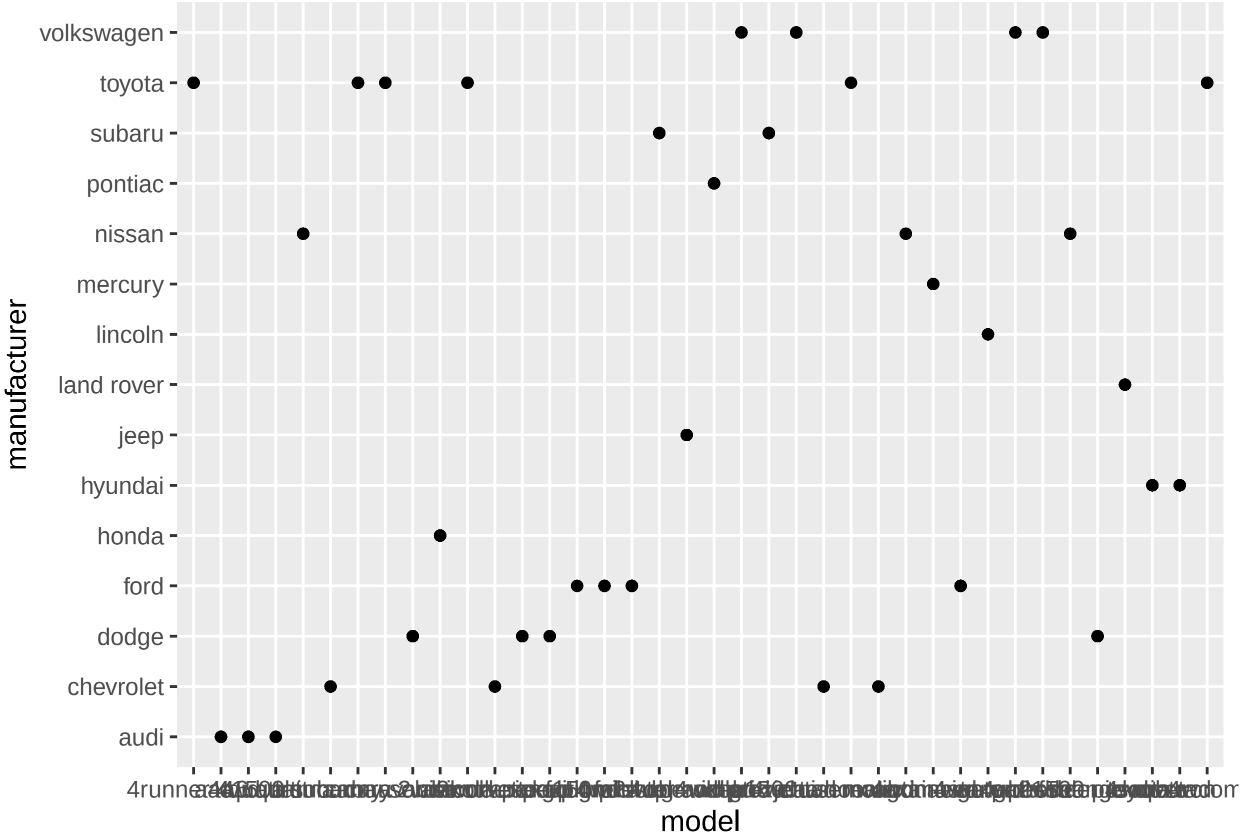
```r
ggplot(variations_num, aes(x = reorder(model, count), y = count)) +
  geom_bar(stat = "identity", fill = "violet") +
  coord_flip() +
  labs(title = "Number of Variations per Model", x = "Model", y = "Count") +
  theme_light()
```



2.2. 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?

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



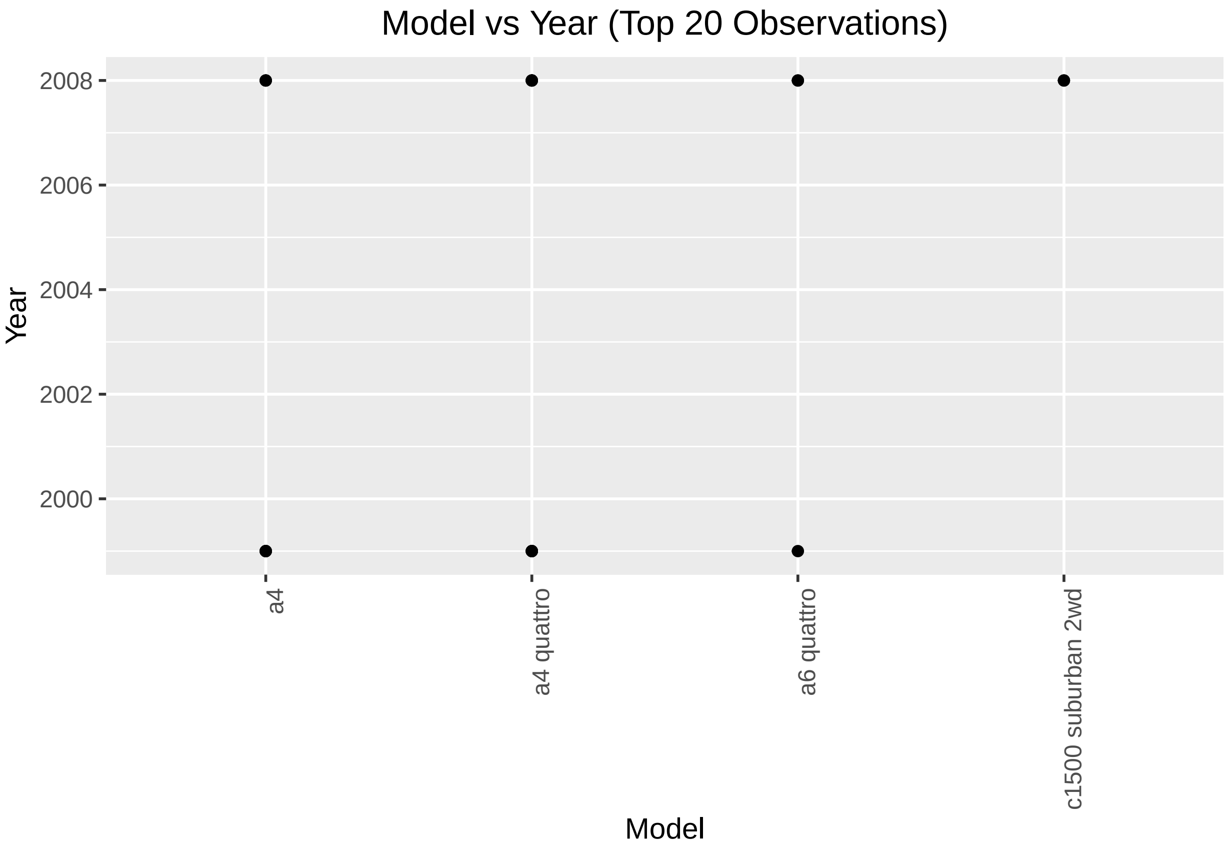
ggplot(mpg, aes(model, manufacturer)) + geom_point() creates a scatter plot showing the relationship between car models (model) on the x-axis and manufacturers (manufacturer) on the y-axis, with each point representing a car observation.

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

In its current form, this plot isn't very useful as it doesn't effectively visualize the relationship between car models and manufacturers. Since both variables are categorical, a scatter plot isn't the most appropriate way to represent this relationship.

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

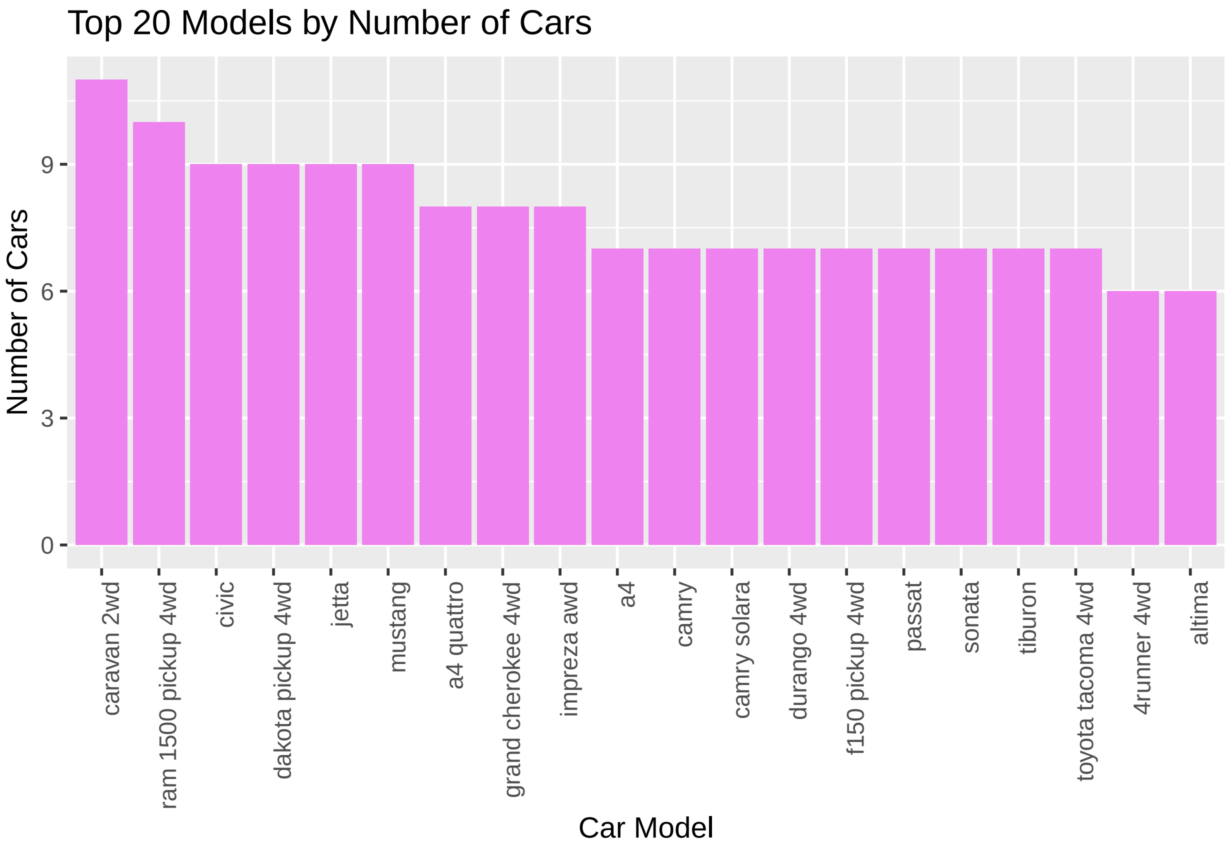
```r
library(ggplot2)

# Top 20 observations from the mpg_data dataset
top_20_data <- head(mpg_data, 20)

# Plot using ggplot2
ggplot(top_20_data, aes(x = model, y = year)) +
  geom_point() +
  labs(
    title = "Model vs Year (Top 20 Observations)",
    x = "Model",
    y = "Year"
  ) +
  theme(
    axis.text.x = element_text(angle = 90, hjust = 1),
    plot.title = element_text(hjust = 0.5)  # Centering the title
  )
```



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

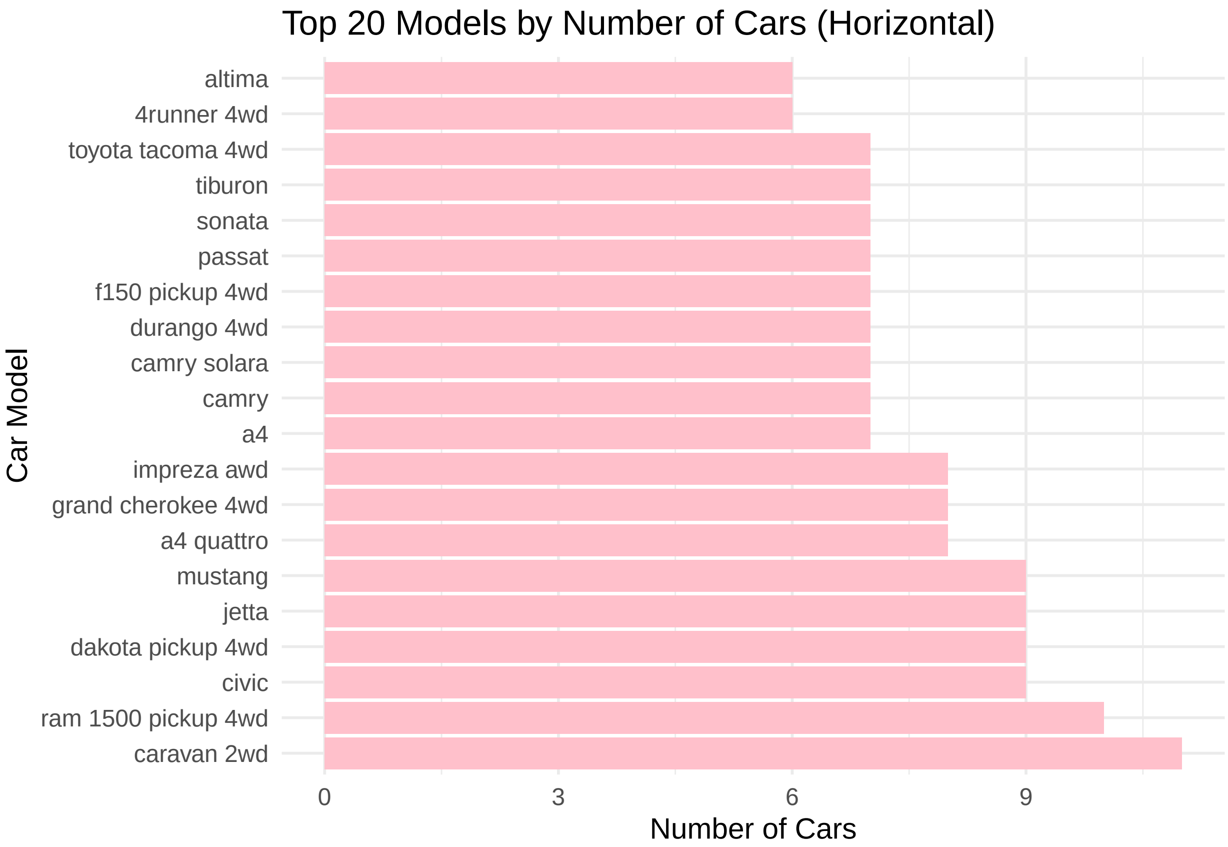
a. Plot using geom_bar() using the top 20 observations only. The graphs shoudl have a title, labels and colors. Show code and results.

```r
library(dplyr)
library(ggplot2)

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

top_20_models <- car_counts %>%
  arrange(desc(count)) %>%
  head(20)

ggplot(top_20_models, aes(x = reorder(model, -count), y = count)) +
  geom_bar(stat = "identity", fill = "violet") +
  labs(
    title = "Top 20 Models by Number of Cars",
    x = "Car Model",
    y = "Number of Cars"
  ) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))  # Rotate x-axis labels
```



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

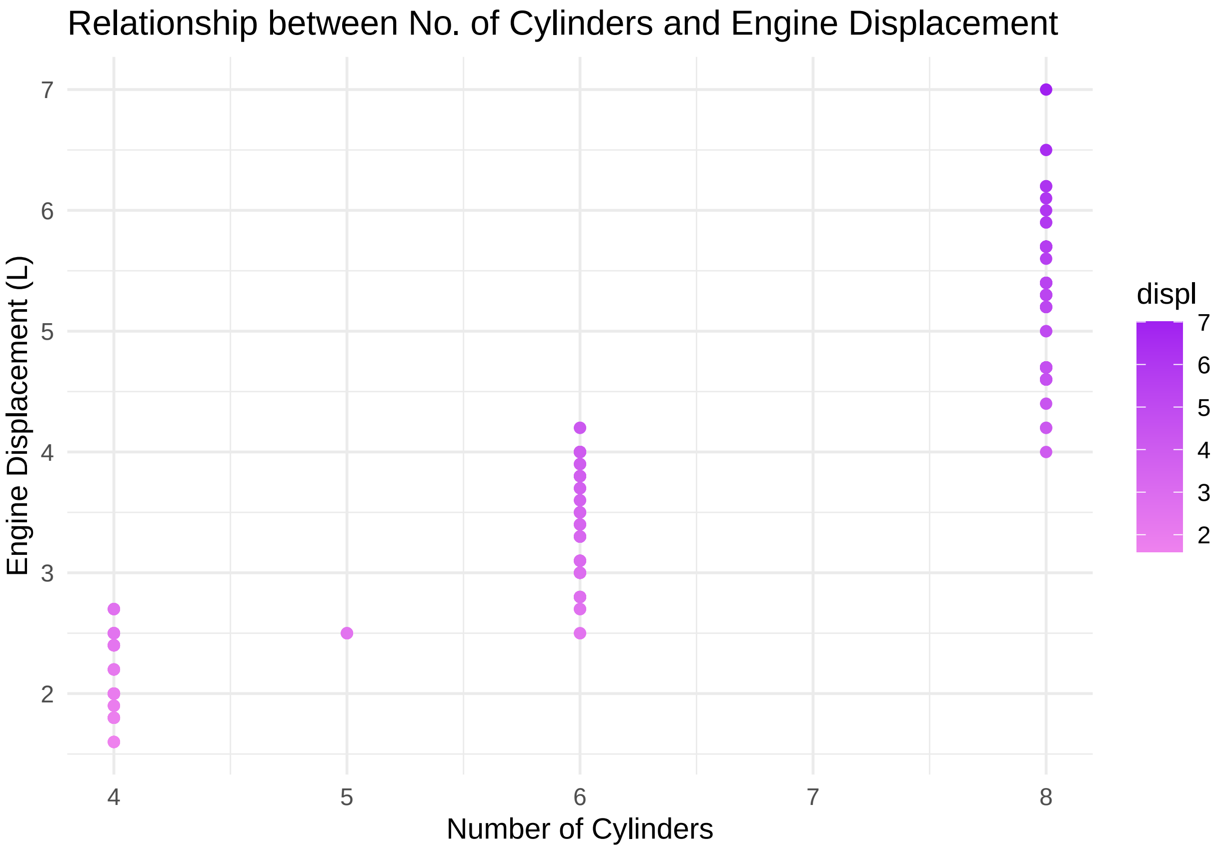
```
ggplot(top_20_models, aes(x = reorder(model, -count), y = count)) +
  geom_bar(stat = "identity", fill = "pink") +
  labs(
    title = "Top 20 Models by Number of Cars (Horizontal)",
    x = "Car Model",
    y = "Number of Cars"
  ) +
  coord_flip() +
  theme_minimal()
```



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

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

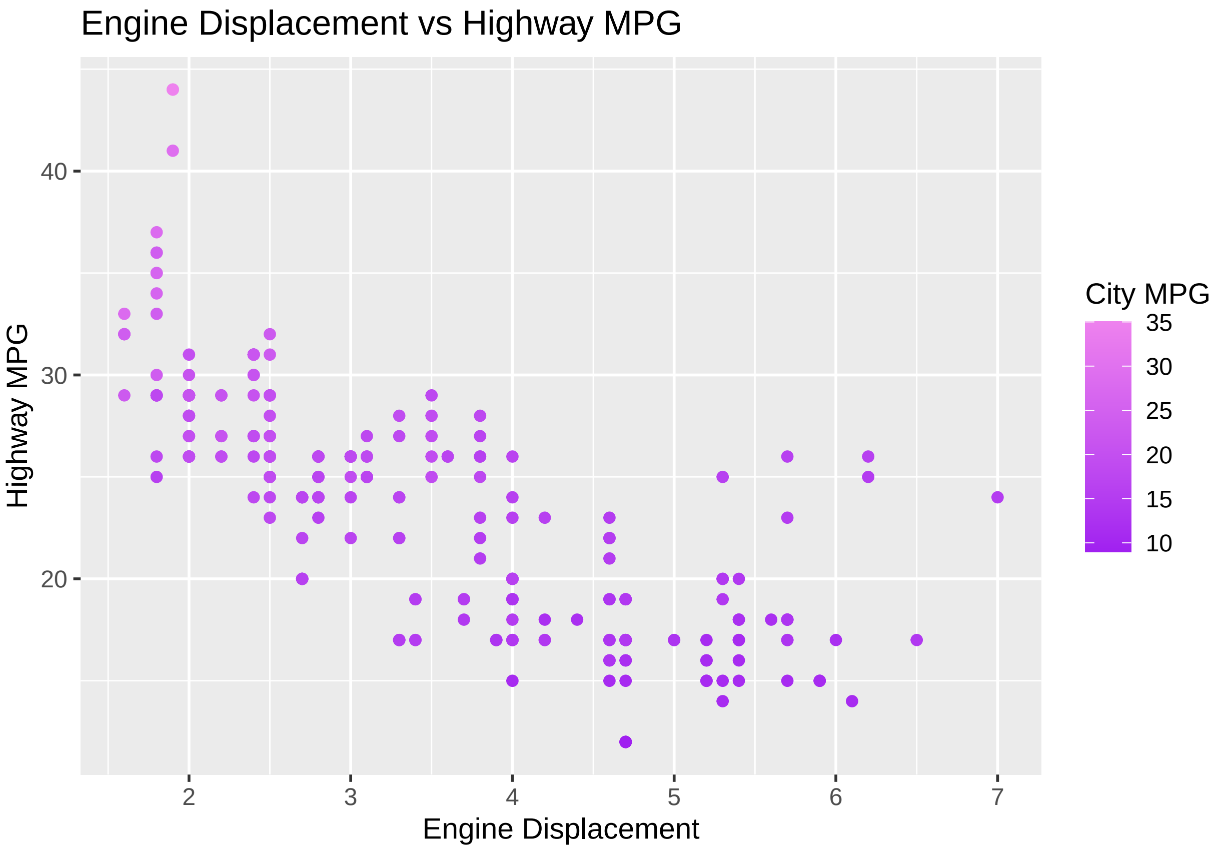
```r
library(ggplot2)

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 (L)"
  ) +
  scale_color_gradient(low = "violet", high = "purple") +
  theme_minimal()
```



The plot will show a scatter plot with the number of cylinders on the x-axis and the engine displacement on the y-axis. The points will be colored based on the engine displacement, where lower displacements will be shaded in violet, and higher displacements will be shaded in purple.

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.

```
ggplot(mpg_data, aes(x = displ, y = hwy, color = cty)) +
  geom_point() +
  labs(title = "Engine Displacement vs Highway MPG",
       x = "Engine Displacement", y = "Highway MPG", color = "City MPG") +
  scale_color_gradient(low = "purple", high = "violet")
```



What is its result?

The results of this plot help explain how engine displacement and vehicle weight both affect fuel efficiency, revealing the trade-offs between engine size, vehicle weight, and fuel economy.

Why it produced such output?

The plot shows that as engine displacement (displ) increases, highway miles per gallon (hwy) decreases, with heavier vehicles (mapped by weight) generally having larger engines and lower fuel efficiency.

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

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

head(traffic)
```

```
##              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
```

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

```
n_obs <- nrow(traffic)

variables <- colnames(traffic)

n_obs
```

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

```
variables
```

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

There are 48120 observations. The variables in the traffic dataset are: "DateTime" "Junction" "Vehicles" "ID".

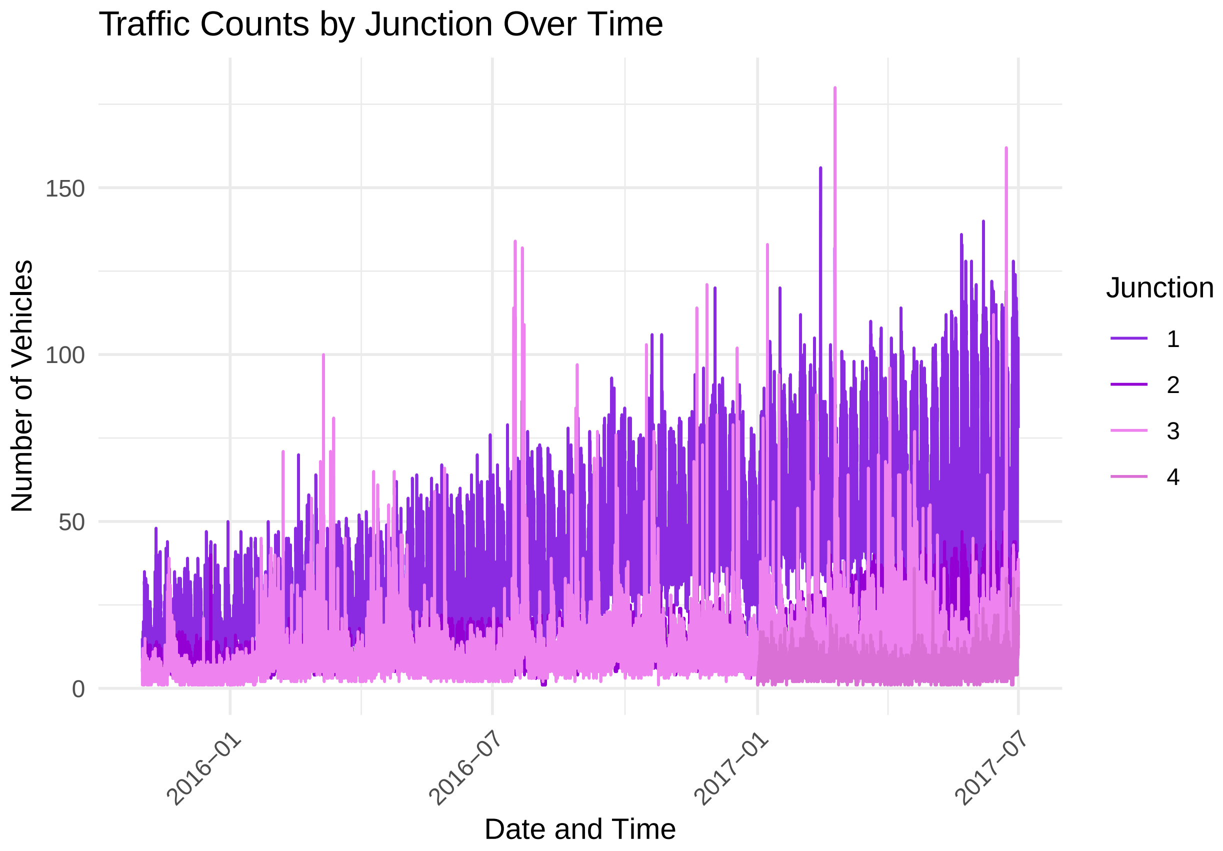
b. subset the traffic dataset into junctions. What is the R codes and its output?

```
library(dplyr)

junction_data <- traffic %>%
  group_by(Junction) %>%
  group_split()

print(junction_data[[1]])
```

```
## # A tibble: 14,592 x 4
##    DateTime            Junction Vehicles          ID
##    <chr>                  <int>    <int>       <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
##  7 2015-11-01 06:00:00        1        9 20151101061
##  8 2015-11-01 07:00:00        1        8 20151101071
##  9 2015-11-01 08:00:00        1       11 20151101081
## 10 2015-11-01 09:00:00        1       12 20151101091
## # i 14,582 more rows
```

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

```
library(ggplot2)

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

ggplot(traffic, aes(x = DateTime, y = Vehicles, color = factor(Junction))) +
  geom_line() +  # Create the line plot
  labs(title = "Traffic Counts by Junction Over Time",
       x = "Date and Time",
       y = "Number of Vehicles",
       color = "Junction") +
  scale_color_manual(values = c("1" = "#8A2BE2",    # Violet
                               "2" = "#9400D3",    # Dark Violet
                               "3" = "#EE82EE",    # Light Violet
                               "4" = "#DA70D6",    # Orchid (light violet shade)
                               "5" = "#E6E6FA")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better readabili
```



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_data <- read_excel("alexa_file.xlsx")

dimensions <- dim(alexa_data)
number_of_observations <- dimensions[1]
number_of_columns <- dimensions[2]

number_of_observations
```

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

```
number_of_columns
```

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

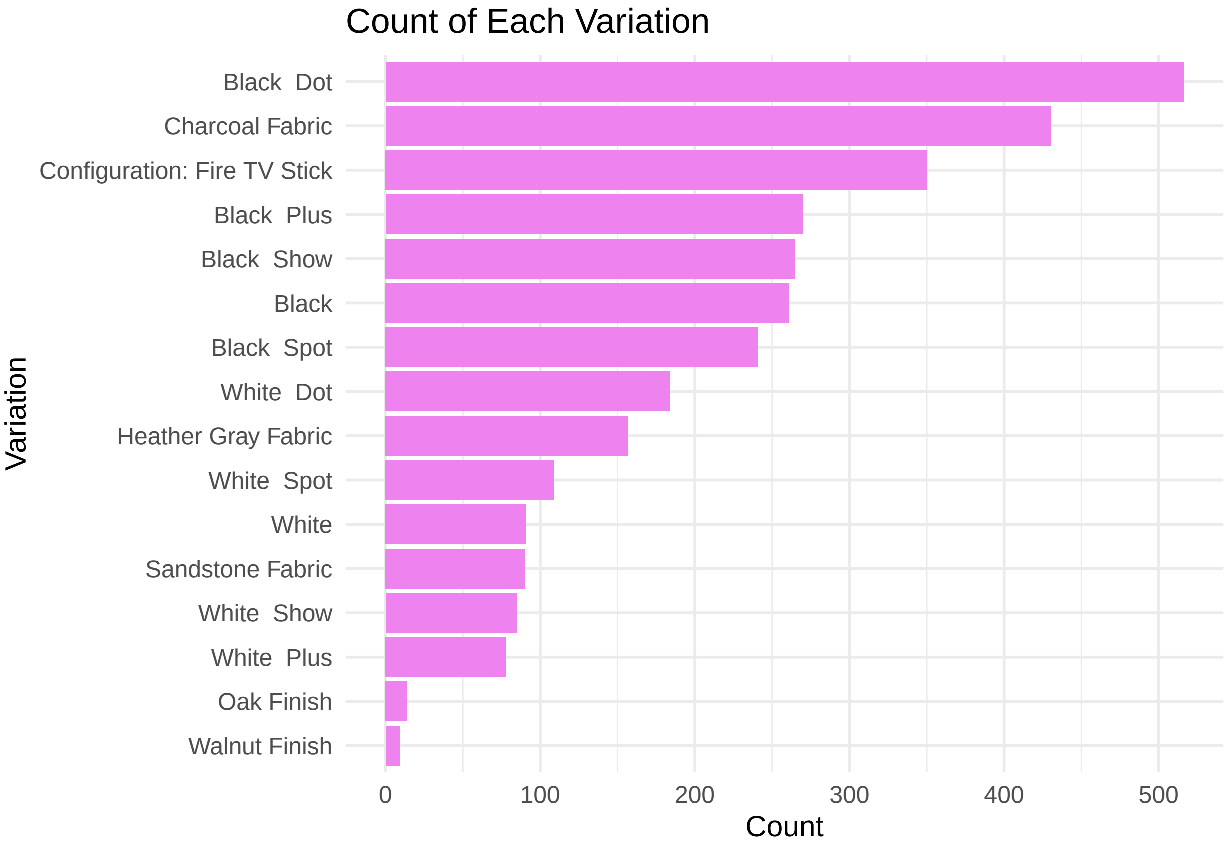
b. group the variations and get the total of each variations. Use dplyr package. Show solution and answer.

```
library(dplyr)

variation_counts <- alexa_data %>%
  group_by(variation) %>%
  summarize(Count = n())
print(variation_counts)
```

```
## # A tibble: 16 x 2
##    variation                      Count
##    <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.

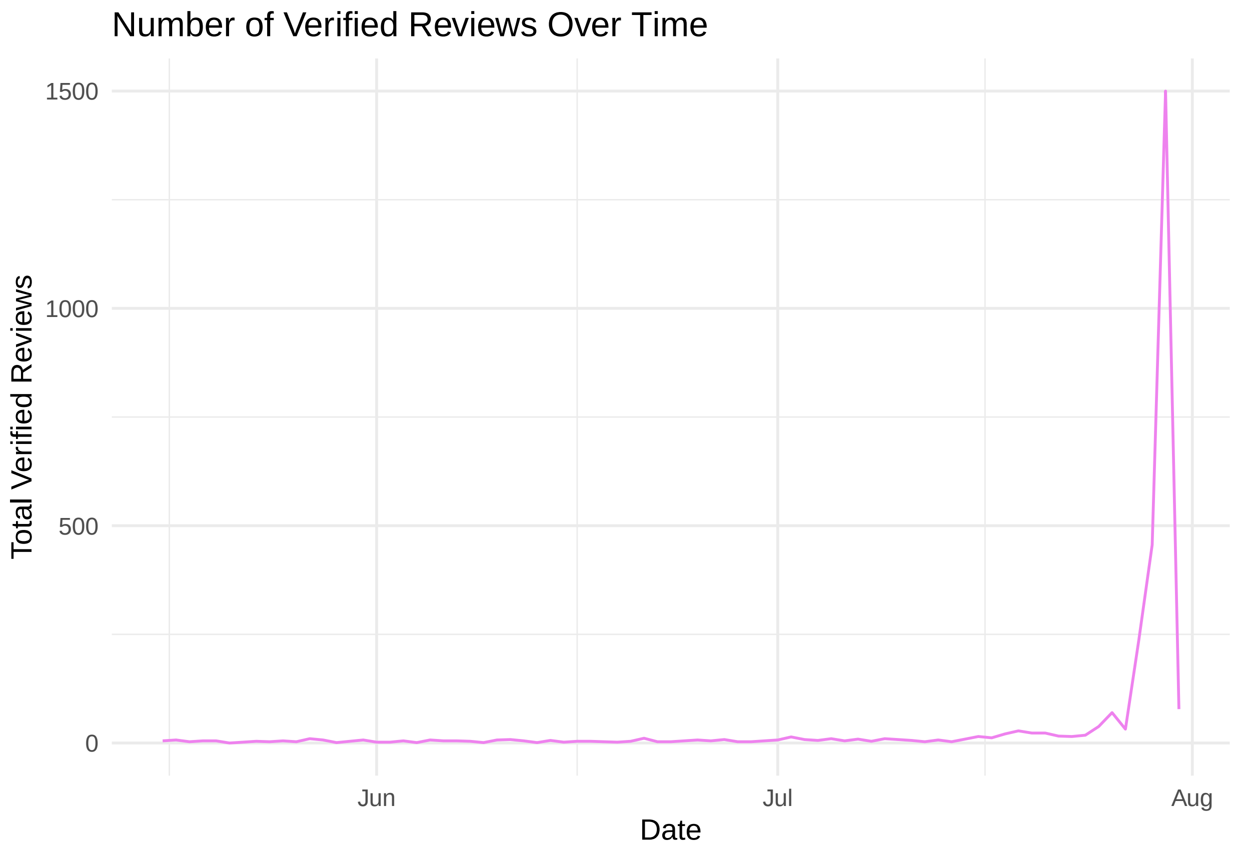
```r
library(ggplot2)

ggplot(variation_counts, aes(x = reorder(variation, Count), y = Count)) +
  geom_bar(stat = "identity", fill = "violet") +
  labs(title = "Count of Each Variation",
       x = "Variation",
       y = "Count") +
  theme_minimal() +
  coord_flip()
```



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.

```r
alexa_data$date <- as.Date(alexa_data$date)

daily_reviews <- alexa_data %>%
  group_by(date) %>%
  summarise(total_verified_reviews = sum(feedback))
```

```r
# Plot the data
ggplot(daily_reviews, aes(x = date, y = total_verified_reviews)) +
  geom_line(color = "violet") +
  labs(title = "Number of 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.

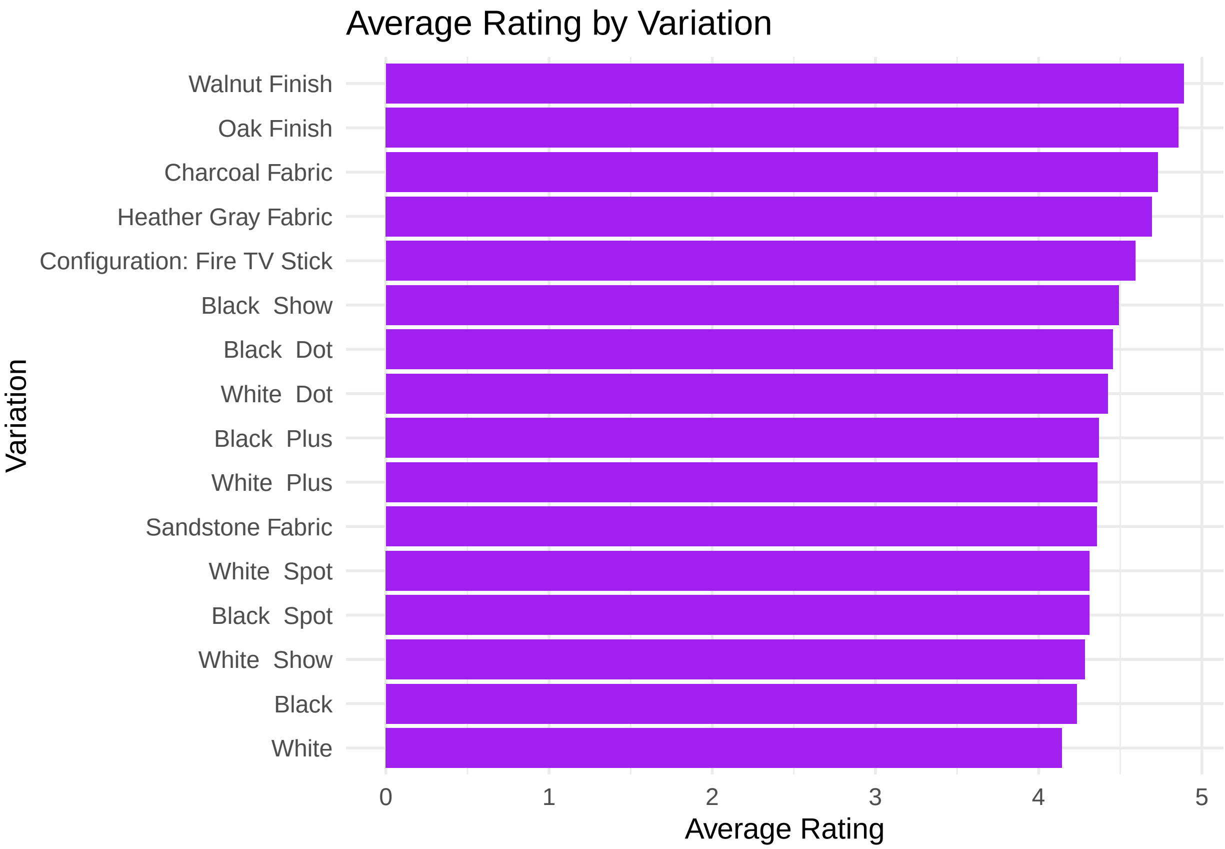
```r
variation_ratings <- alexa_data %>%
  group_by(variation) %>%
  summarize(Average_Rating = mean(as.numeric(rating), na.rm = TRUE)) %>%
  arrange(desc(Average_Rating))
print(variation_ratings)
```

```
## # A tibble: 16 x 2
##    variation                        Average_Rating
##    <chr>                                     <dbl>
##  1 Walnut Finish                              4.89
##  2 Oak Finish                                 4.86
##  3 Charcoal Fabric                            4.73
##  4 Heather Gray Fabric                     4.69
##  5 Configuration: Fire TV Stick            4.59
##  6 Black  Show                             4.49
##  7 Black  Dot                              4.45
##  8 White  Dot                              4.42
##  9 Black  Plus                             4.37
## 10 White  Plus                             4.36
## 11 Sandstone Fabric                        4.36
## 12 White  Spot                             4.31
## 13 Black  Spot                             4.31
## 14 White  Show                             4.28
## 15 Black                                   4.23
## 16 White                                   4.14
```

```
highest_variation <- variation_ratings %>%
  slice(1)
print(highest_variation)
```

```
## # A tibble: 1 x 2
##   variation     Average_Rating
##   <chr>                  <dbl>
## 1 Walnut Finish           4.89
```

```
ggplot(variation_ratings, aes(x = reorder(variation, Average_Rating), y = Average_Rating)) +
  geom_bar(stat = "identity", fill = "purple") +
  labs(title = "Average Rating by Variation",
       x = "Variation",
       y = "Average Rating") +
  theme_minimal() +
  coord_flip()
```


Average Rating by Variation
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
0
1
2
3
4
5
Average Rating