Worksheet 4C

Bea Joyce C. Baylon

2024-10-30

1. Use the dataset mpg

```r
#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" ...
```

```r
#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.

```r
#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
```

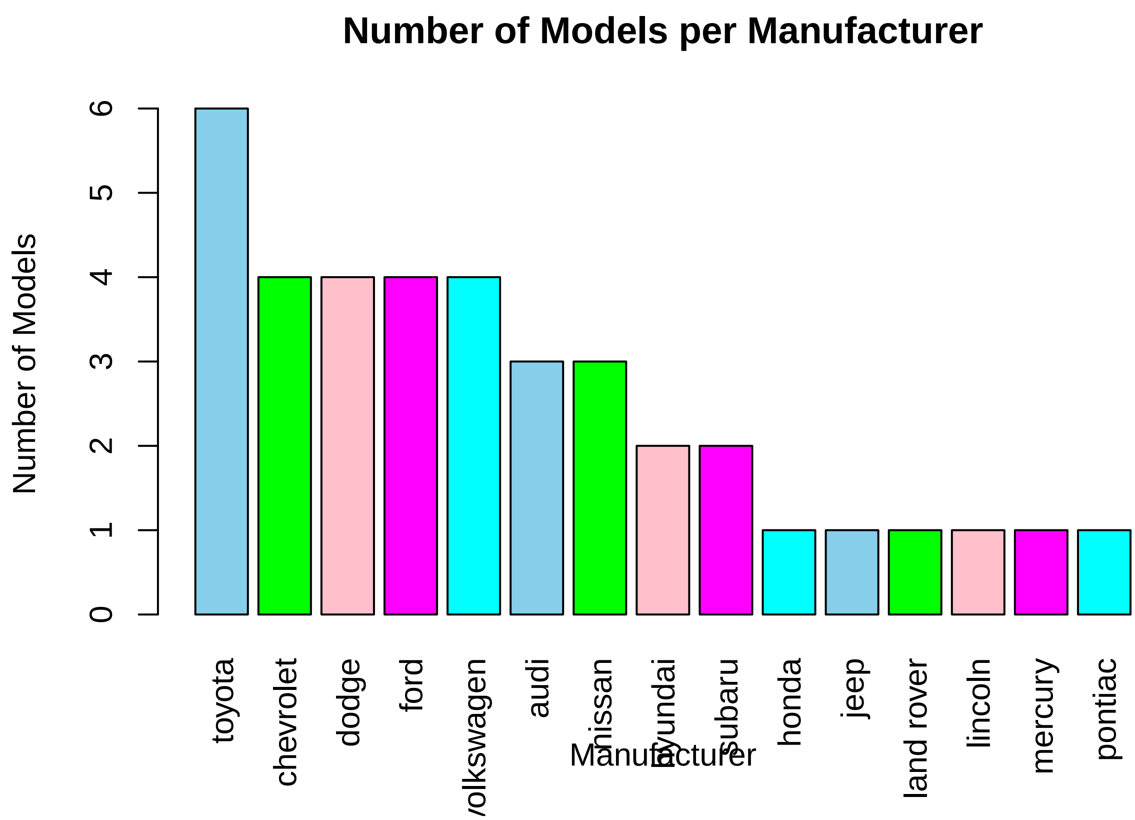
```r
variations_num <- table(mpg$model)
variations_num [variations_num == max(variations_num)]
```

```
## 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", "green", "pink", "magenta", "cyan"),
        las = 3)
```

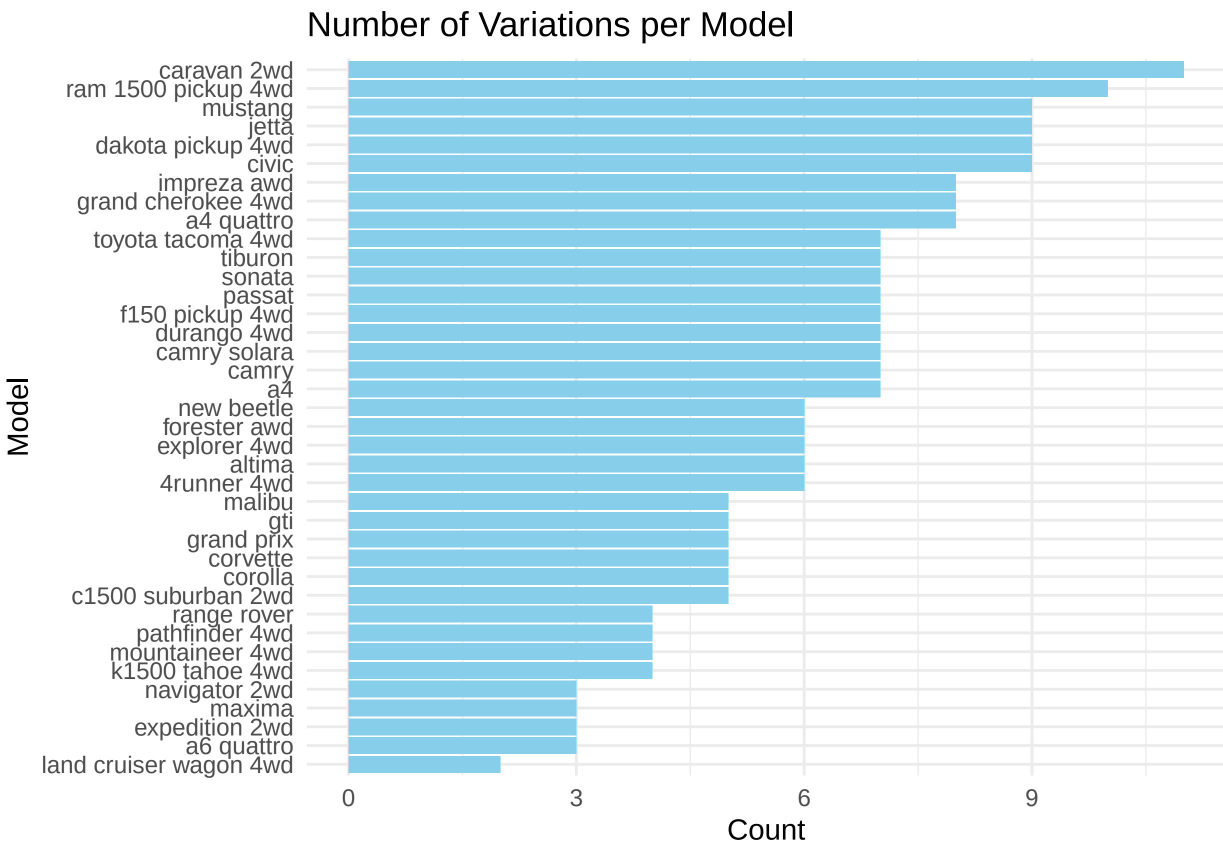



```r
#below is the barplot from the ggplot().
variations_num <- mpg %>%
  group_by(model) %>%
  summarize(count = n()) %>%
  arrange(desc(count))

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
```

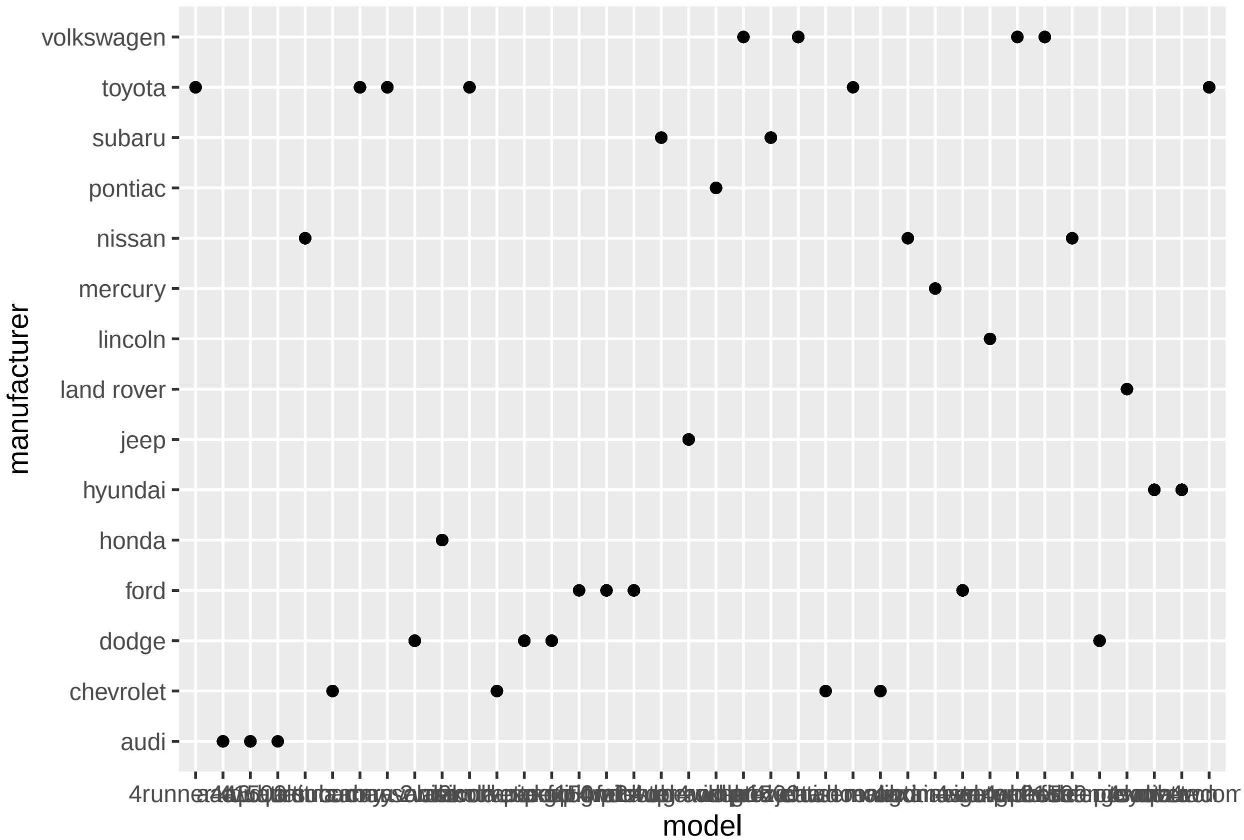
```r
ggplot(variations_num,
       aes(x = reorder(model, count), y = count)) +
       geom_bar(stat = "identity", fill = "skyblue") +  coord_flip() +
       labs(title = "Number of Variations per Model", x = "Model", y = "Count") +
       theme_minimal()
```



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. The scatter plot is not very useful for understanding the relationship between models and manufacturers because:

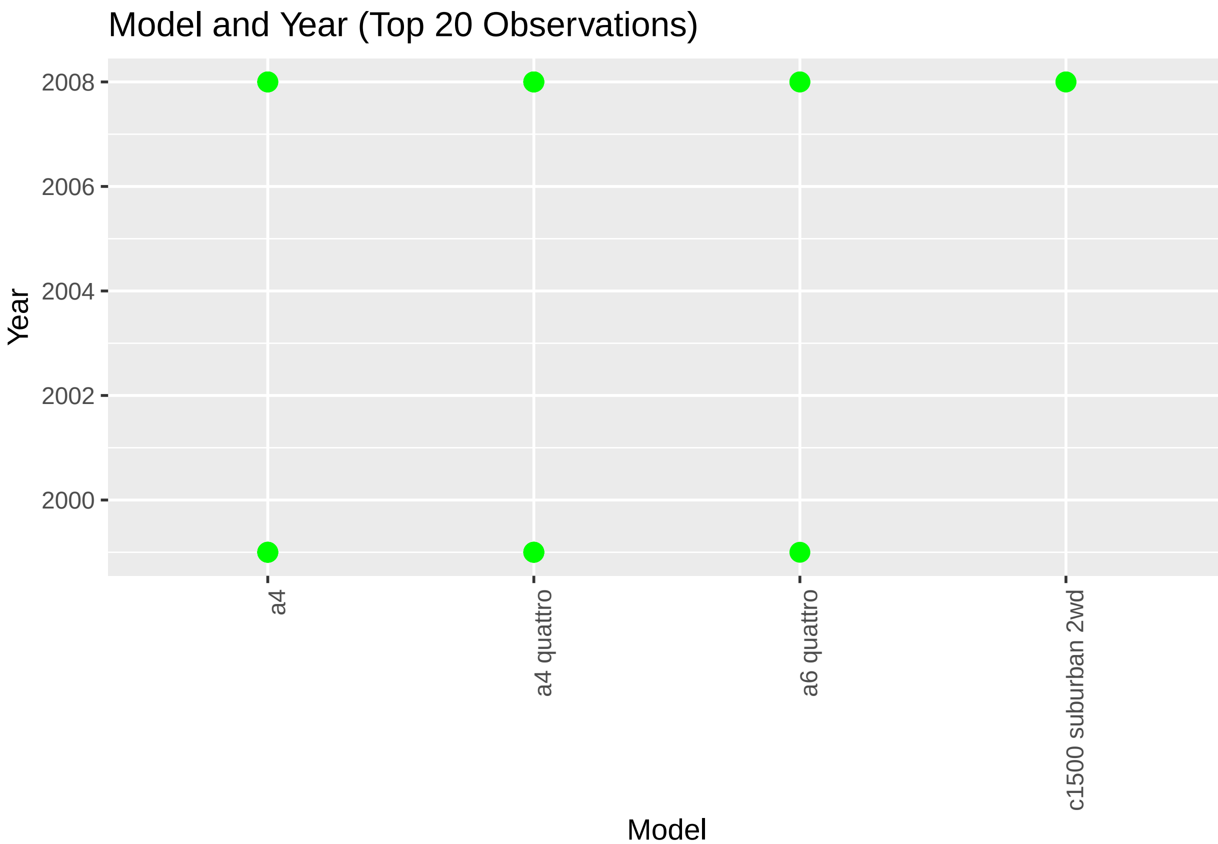
-The plot becomes cluttered with overlapping points when there are many models and manufacturers. Categorical data are better visualized using bar plots or heatmaps for easier interpretation. To make the data more informative:

-For me, Using a bar plot to show the number of models for each manufacturer. This provides a clear comparison between manufacturers.

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 = "green", size = 3) +
       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
```

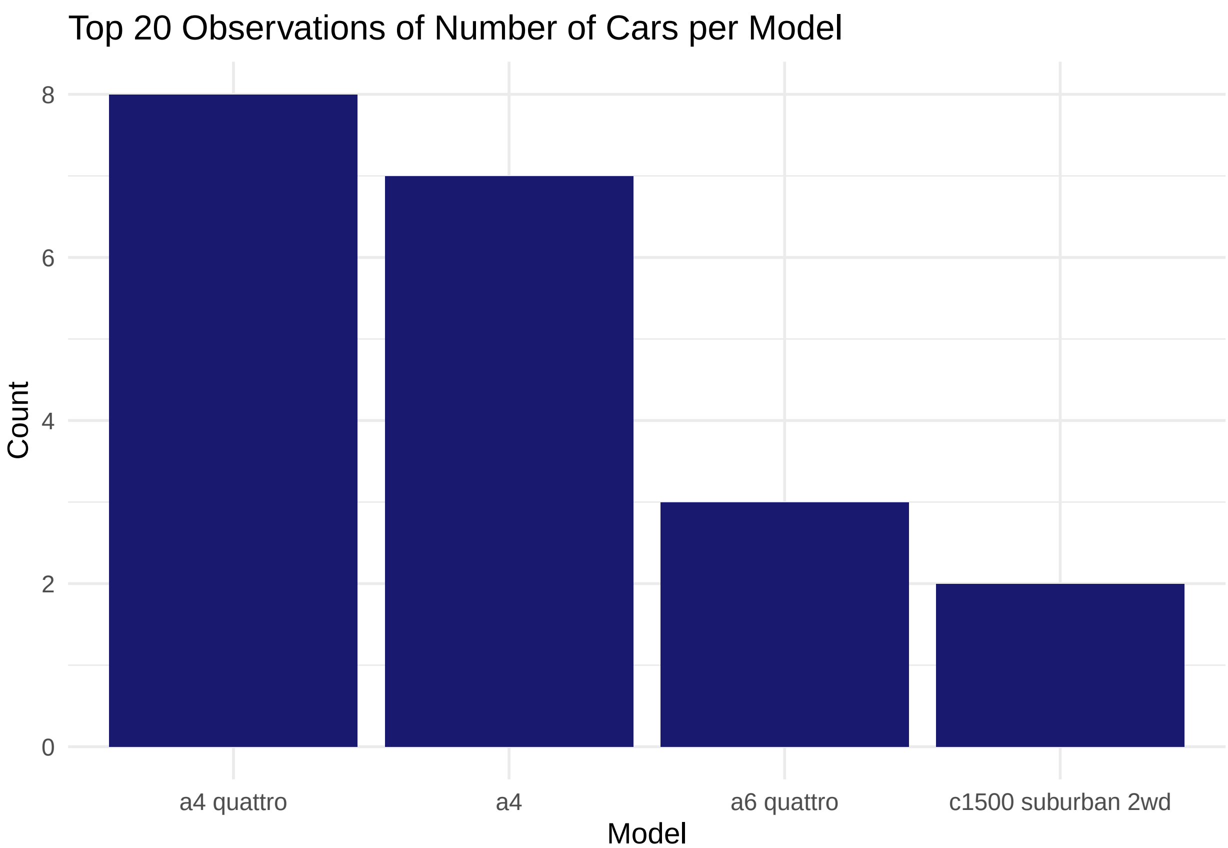
```
## # A tibble: 38 x 2
##    model              count
##    <chr>              <int>
##  1 4runner 4wd            6
##  2 a4                     7
##  3 a4 quattro             8
##  4 a6 quattro             3
##  5 altima                 6
##  6 c1500 suburban 2wd     5
##  7 camry               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

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

carNum20 <- obs20 %>%
```

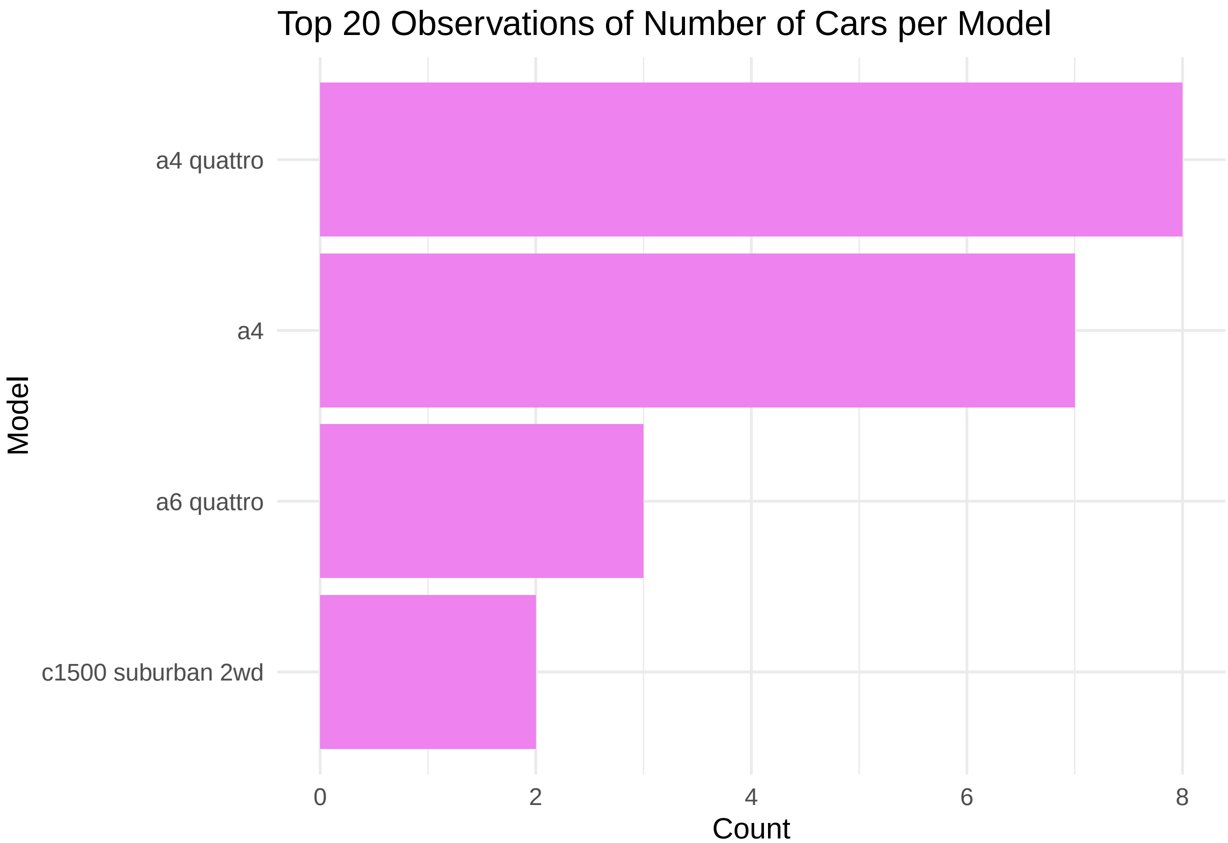
```r
  group_by(model) %>%
  summarise(count = n())

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



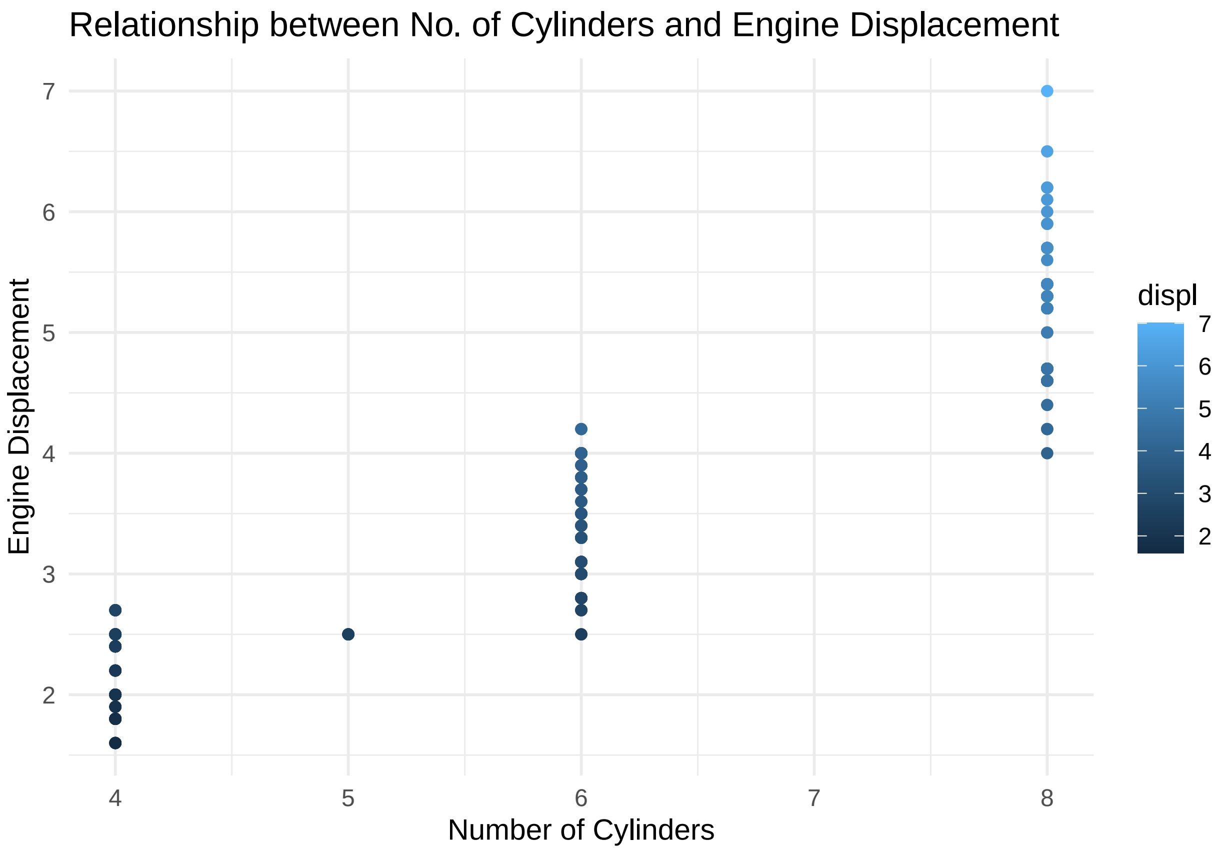

```r
#B. Plot using geom_bar() + coord_flip()
ggplot(
  carNum20,
  aes(x = reorder(model, count), y = count)
) +
  geom_bar(stat = "identity", fill = "violet") +
  labs(
    title = "Top 20 Observations of Number of Cars per Model",
    x = "Model",
    y = "Count"
  ) +
  coord_flip() +
```

```r
  theme_minimal()
```



5. Plot the relationship between cyl - number of cylinders and displ - engine displacement using geom_point with aesthetic color = engine displacement.

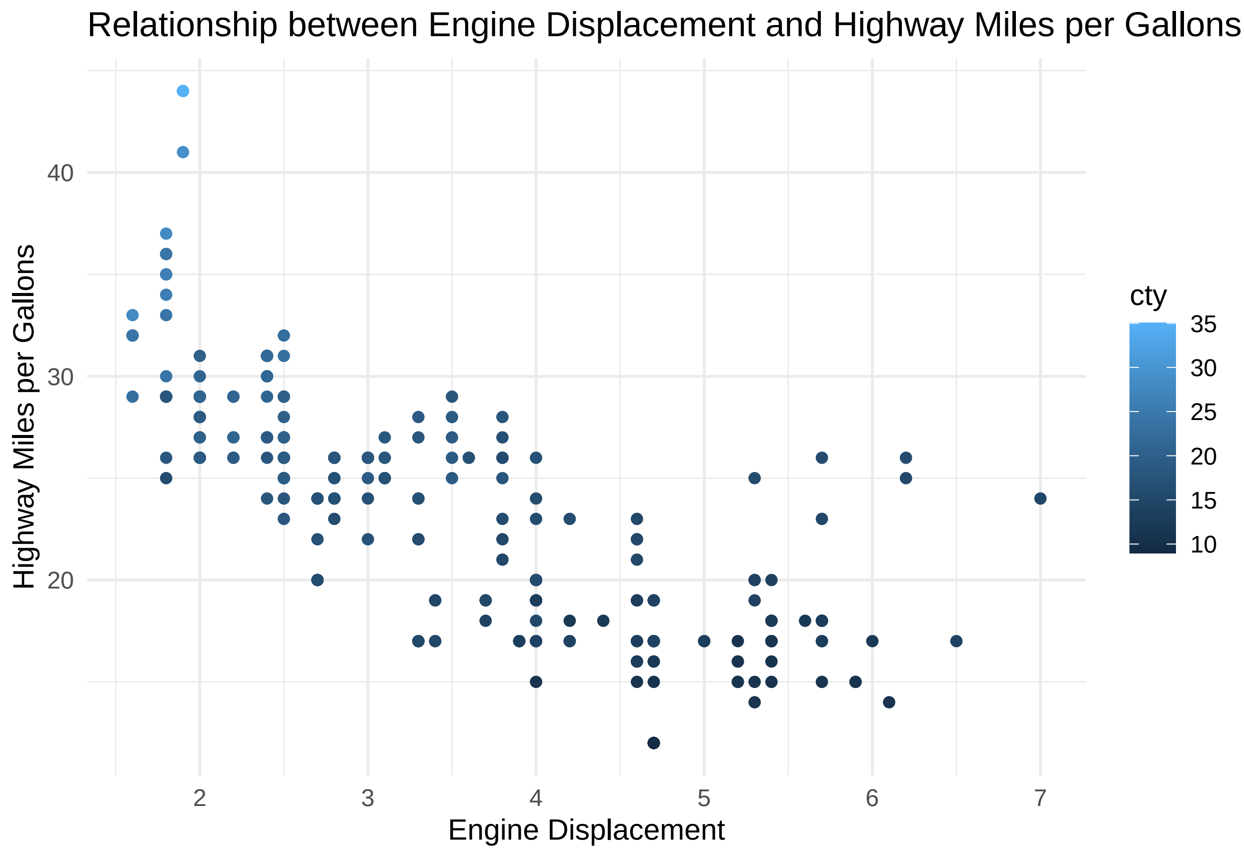
```r
#A. How would you describe its relationship? Show the codes and its result.
ggplot(mpg_data,
       aes(x = cyl, y = displ, color = displ)) +
       geom_point() +
       labs(
          title = "Relationship between No. of Cylinders and Engine Displacement",
          x = "Number of Cylinders",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



Observation: The scatter plot demonstrates an inverse relationship between engine displacement and highway miles per gallon. Vehicles with larger engines (displ) tend to have lower fuel efficiency (hwy). The color aesthetic adds cty (city miles per gallon) as a third dimension, further illustrating that vehicles with higher city MPG also tend to have higher highway MPG.

Reason for Output: The result is expected as larger engines typically consume more fuel that tends to result in lower efficiency. Also, the color mapping enhances the plot by showing how city MPG aligns with this trend.

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 ...
```

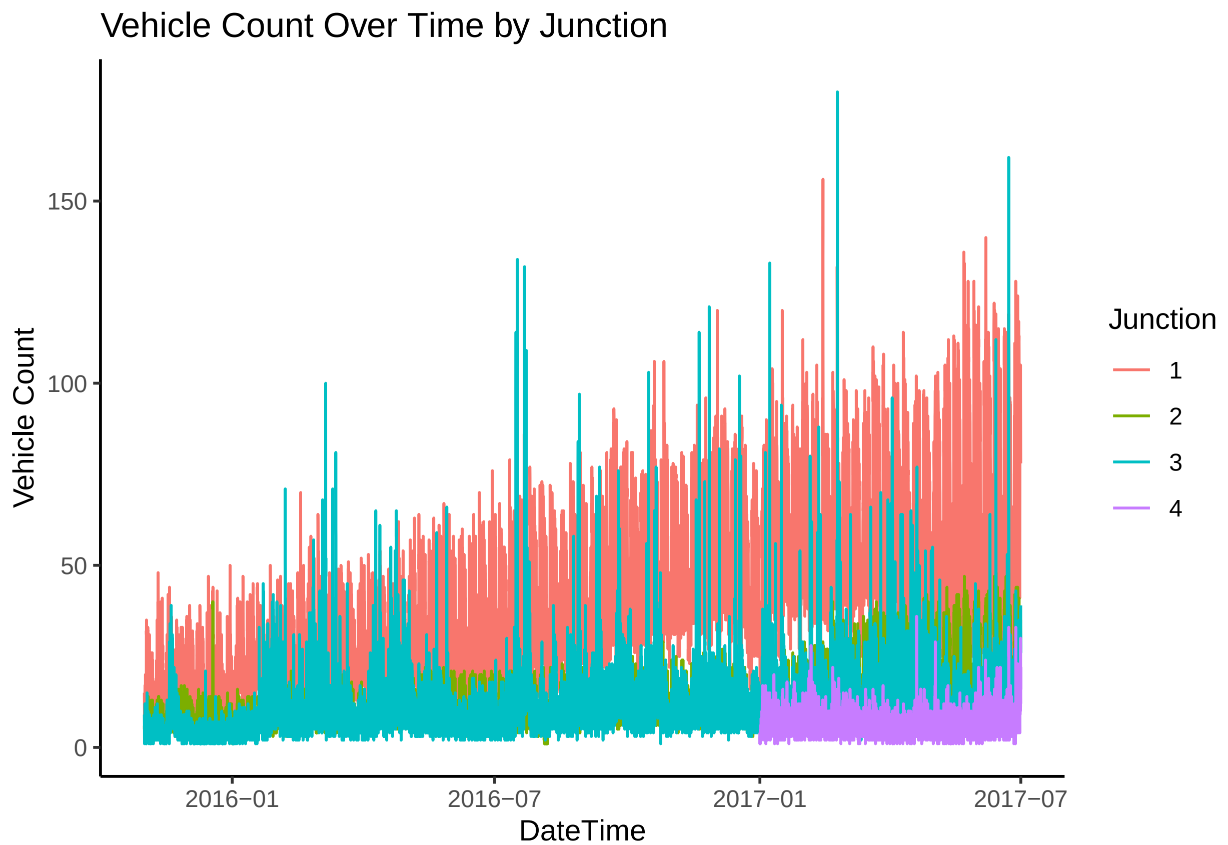
- The number of observations of traffic.csv is 48,120. The variables on the other is 4 which are named DateTime, Junction, Vehicles, and ID.

```
#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 1 ...
```

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

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

alexa_variations
```

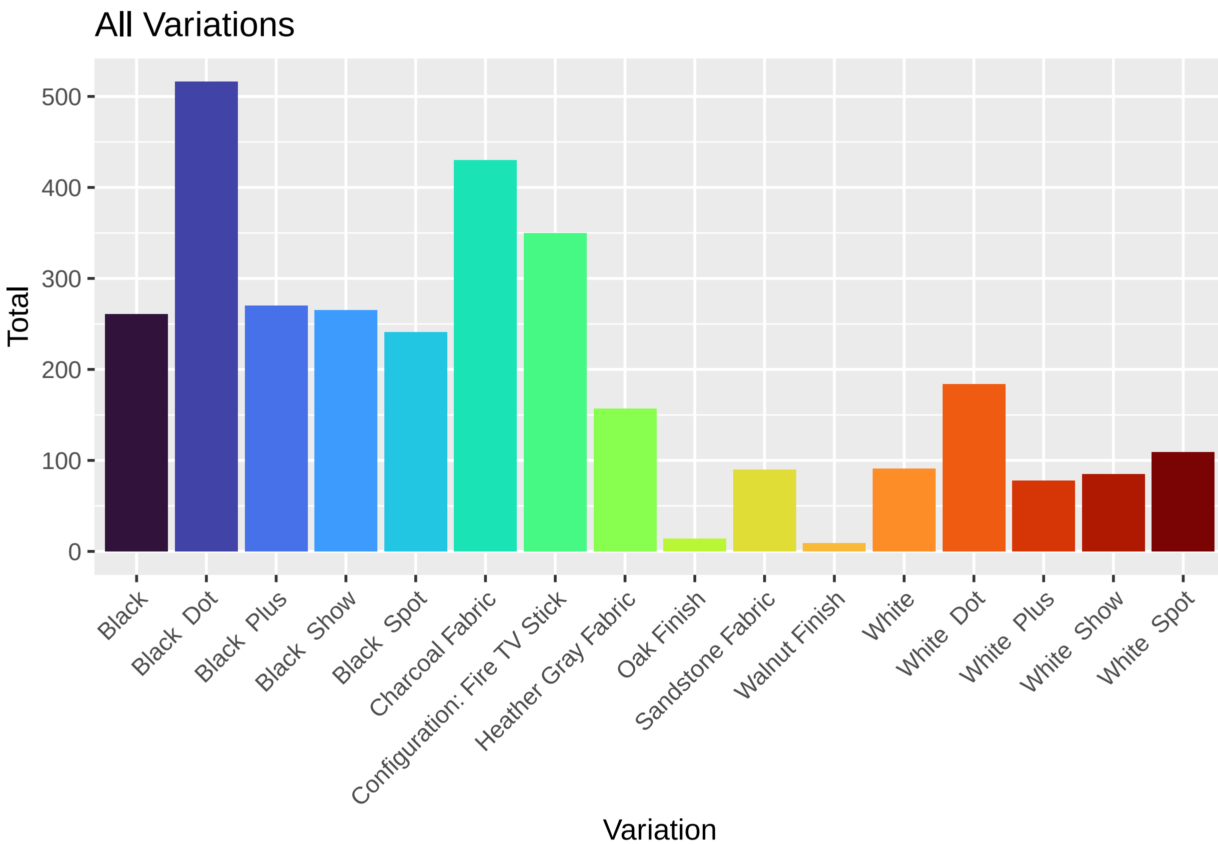
```
## # A tibble: 16 x 2
##    variation                    total
##    <chr>                        <int>
##  1 Black                          261
##  2 Black  Dot                     516
##  3 Black  Plus                    270
##  4 Black  Show                    265
##  5 Black  Spot                    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
```

```r
#C. Plot the variations using the ggplot() function.
library(viridis)
```

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

```r
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")
```



The bar chart highlights the total reviews for each variation. Variations with darker bars indicate higher popularity based on the number of reviews.

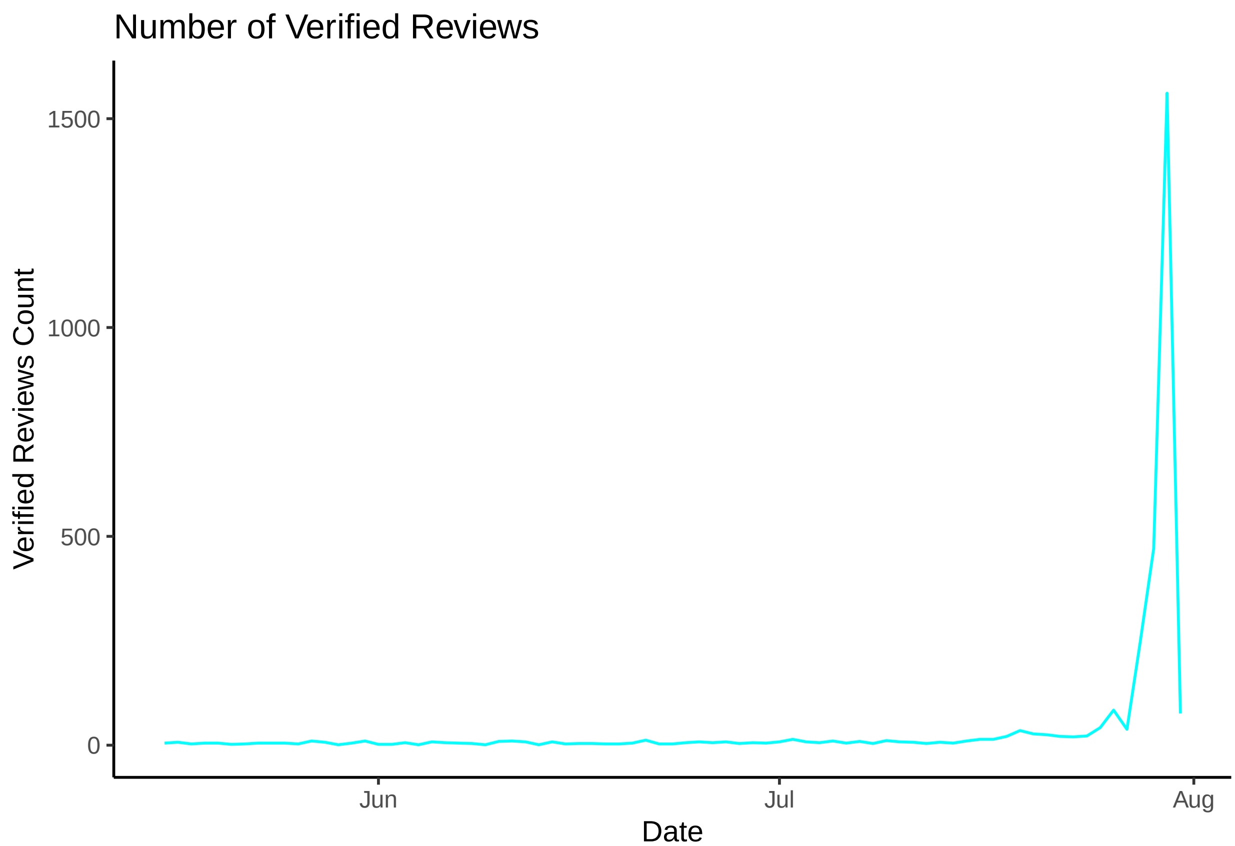
```r
#D. Plot a geom_line() with the date and the number of verified reviews.
library(ggplot2)
library(dplyr)

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
```

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



The line plot shows the number of verified reviews over time. Peaks may indicate promotional events, holidays, or product launches.

```r
#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 = 50, hjust = 2)) +
  theme(legend.position = "none") +
```

```r
scale_fill_viridis_d(option = "inferno")
```