Worksheet-4c in R

Necel Kate Noblezada

2024-10-30

1. Use the dataset mpg

1.

a.

```
mpg_data <- read.csv("mpg.csv")
```

b.

```
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" ...
```

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

c.

The continous variables are displ, year, cyl, cty, and hwy.

2.

```
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.

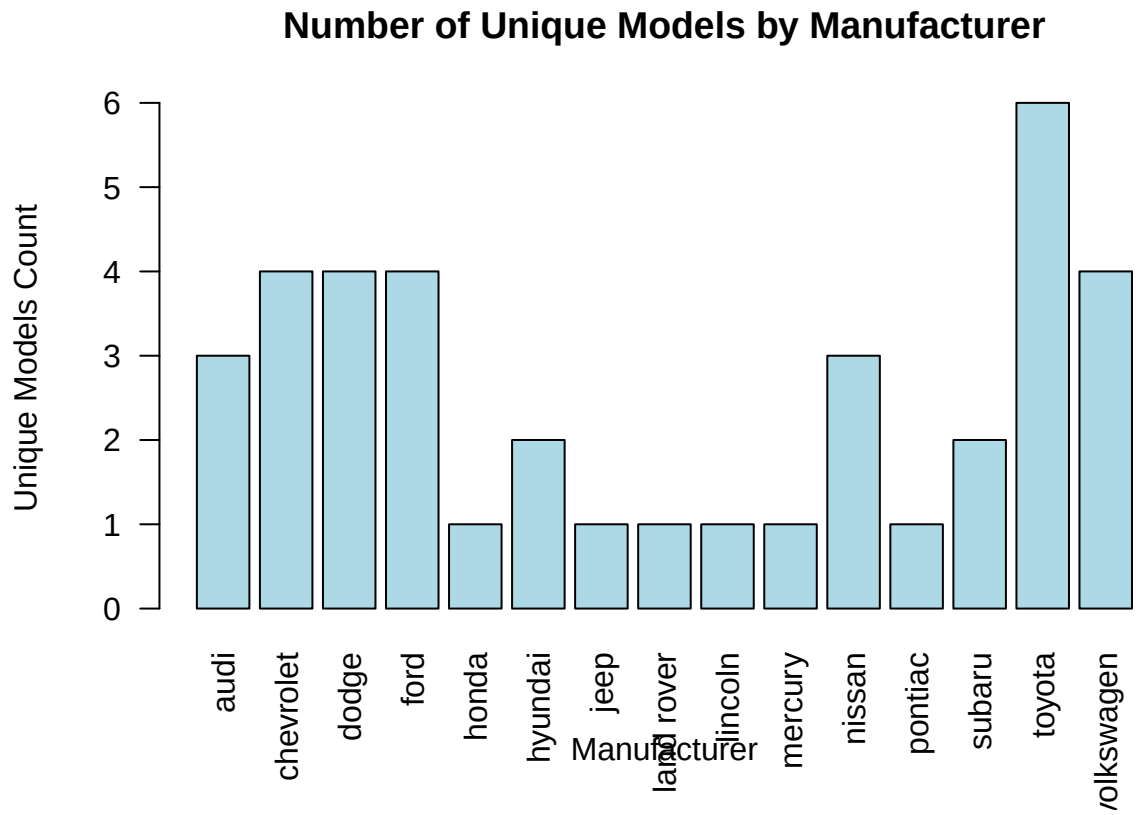
```
unique_models <- mpg %>%
  group_by(manufacturer) %>%
  summarise(unique_models_count = n_distinct(model))

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
```

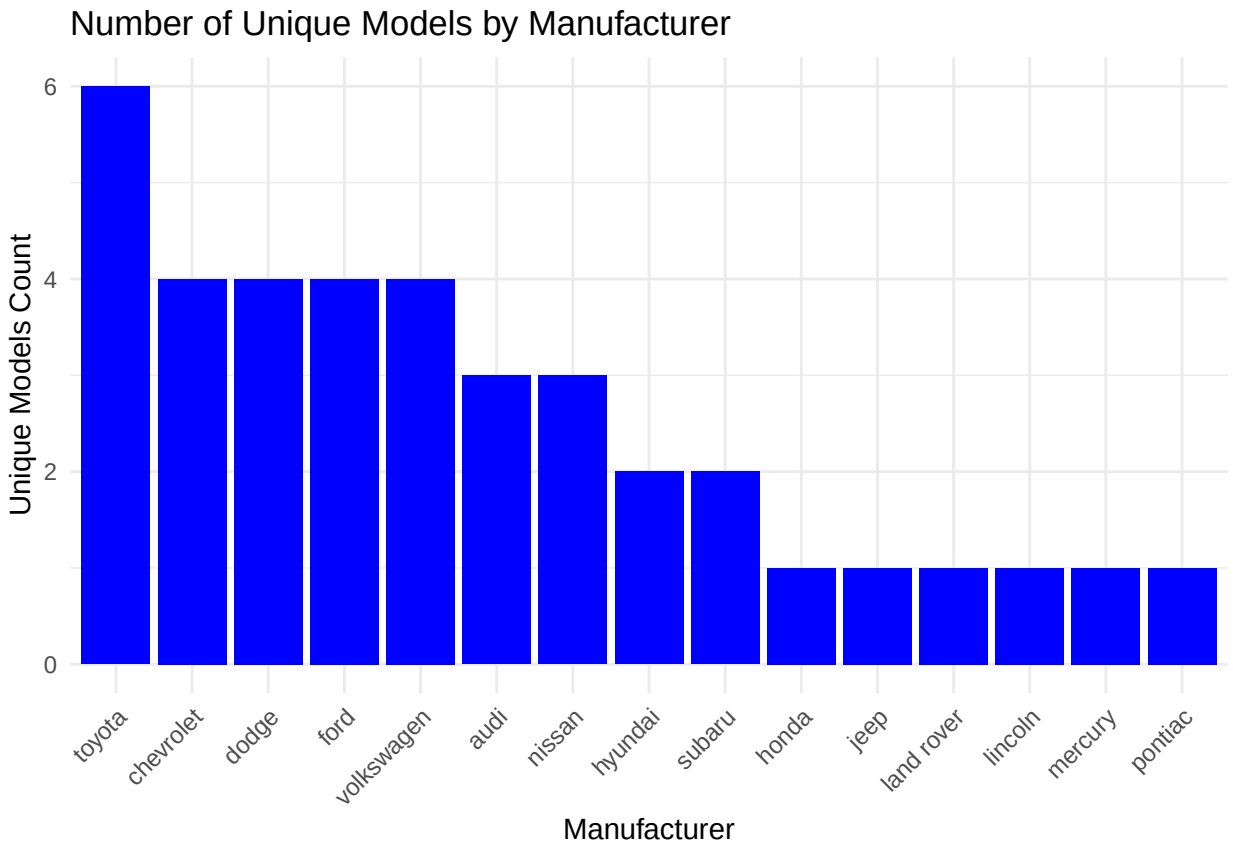
```
## 11 nissan                       3
## 12 pontiac                      1
## 13 subaru                       2
## 14 toyota                       6
## 15 volkswagen                   4
```

b.

```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))
```



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)
```

```
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
```

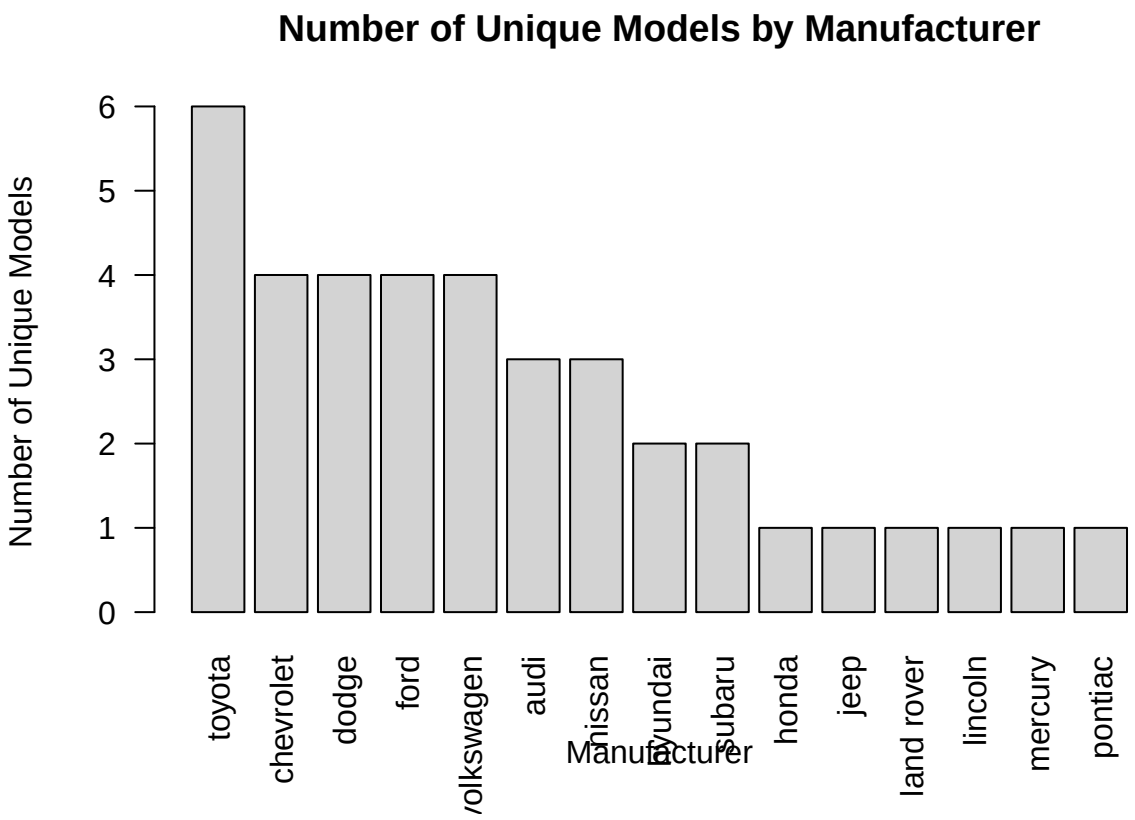
a. Group the manufacturers and find the unique models. Show your codes and result

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

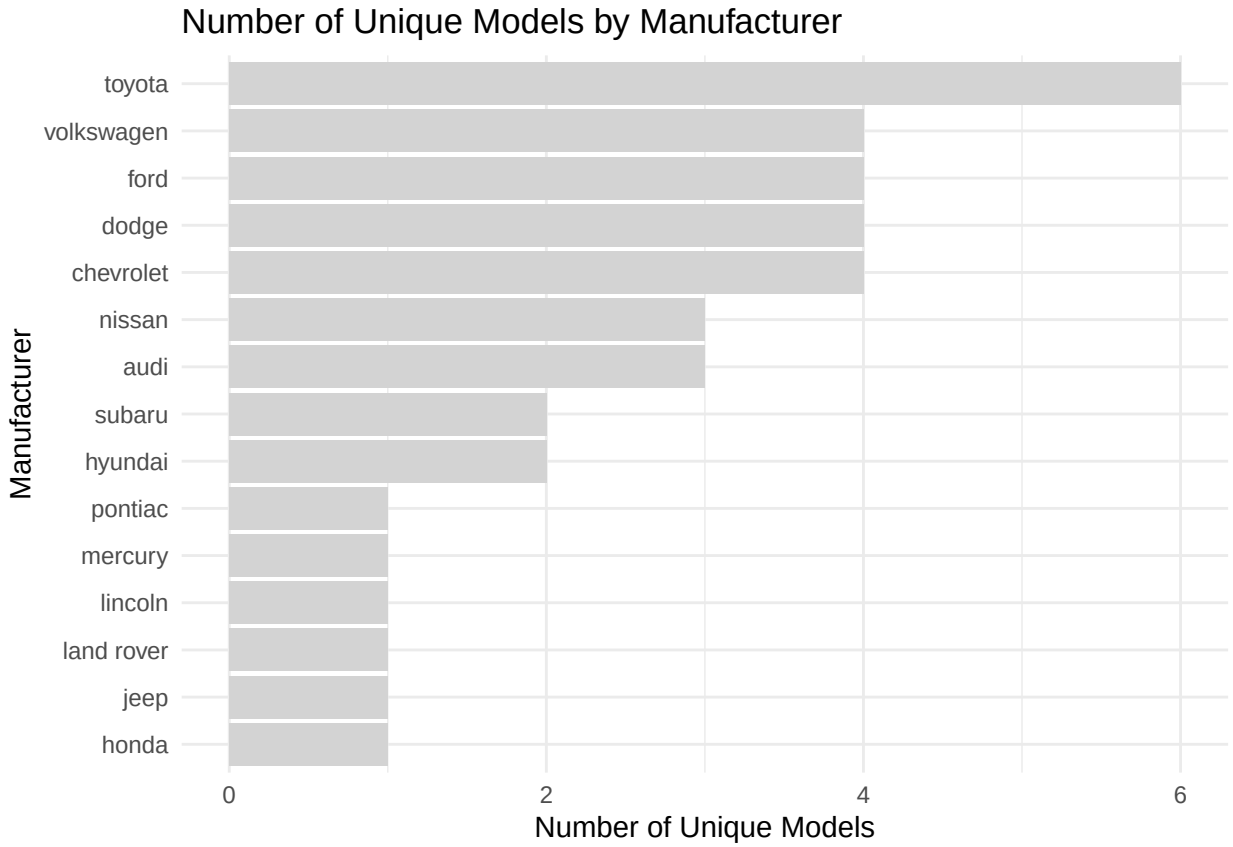
```
## # A tibble: 15 x 2
##    manufacturer unique_models
##    <chr>        <list>
##  1 audi         <chr [3]>
##  2 chevrolet    <chr [4]>
##  3 dodge        <chr [4]>
##  4 ford         <chr [4]>
##  5 honda        <chr [1]>
##  6 hyundai      <chr [2]>
##  7 jeep         <chr [1]>
##  8 land rover   <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.

```
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)

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 model and the manufacturer.

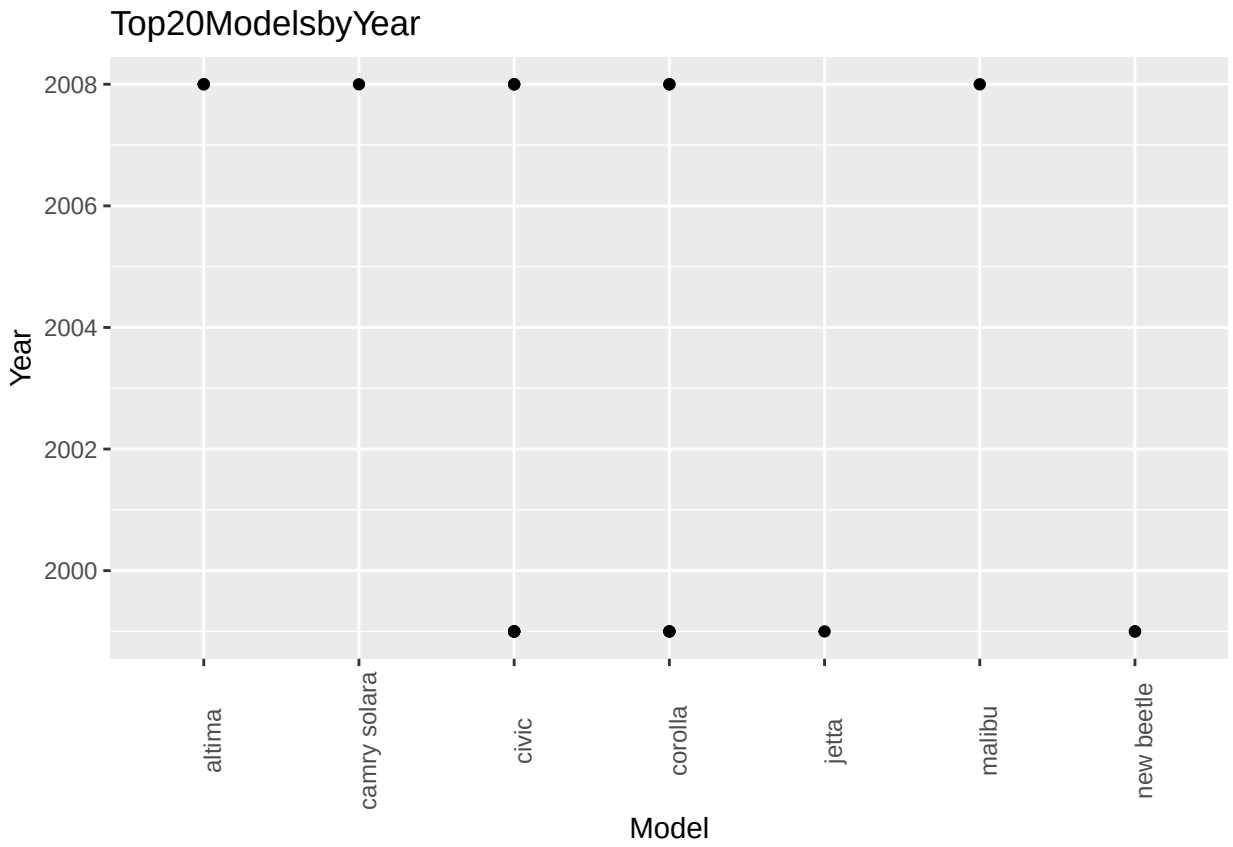
a. What does ggplot(mpg, aes(model, manufacturer))+geom_point() show?

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

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="Top20ModelsbyYear",x="Model",y="Year") +
```

```
theme(axis.text.x=element_text(angle=90,hjust=0.5))
```

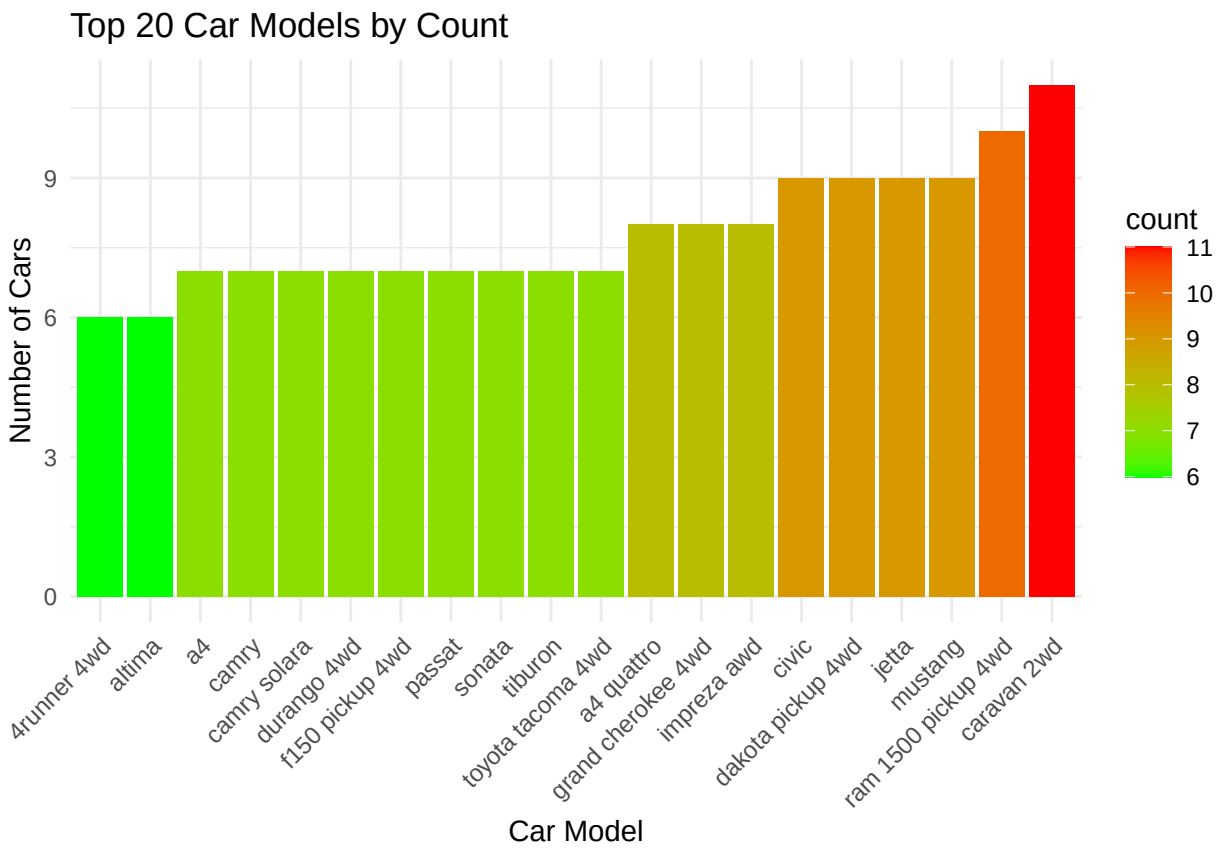


4.

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

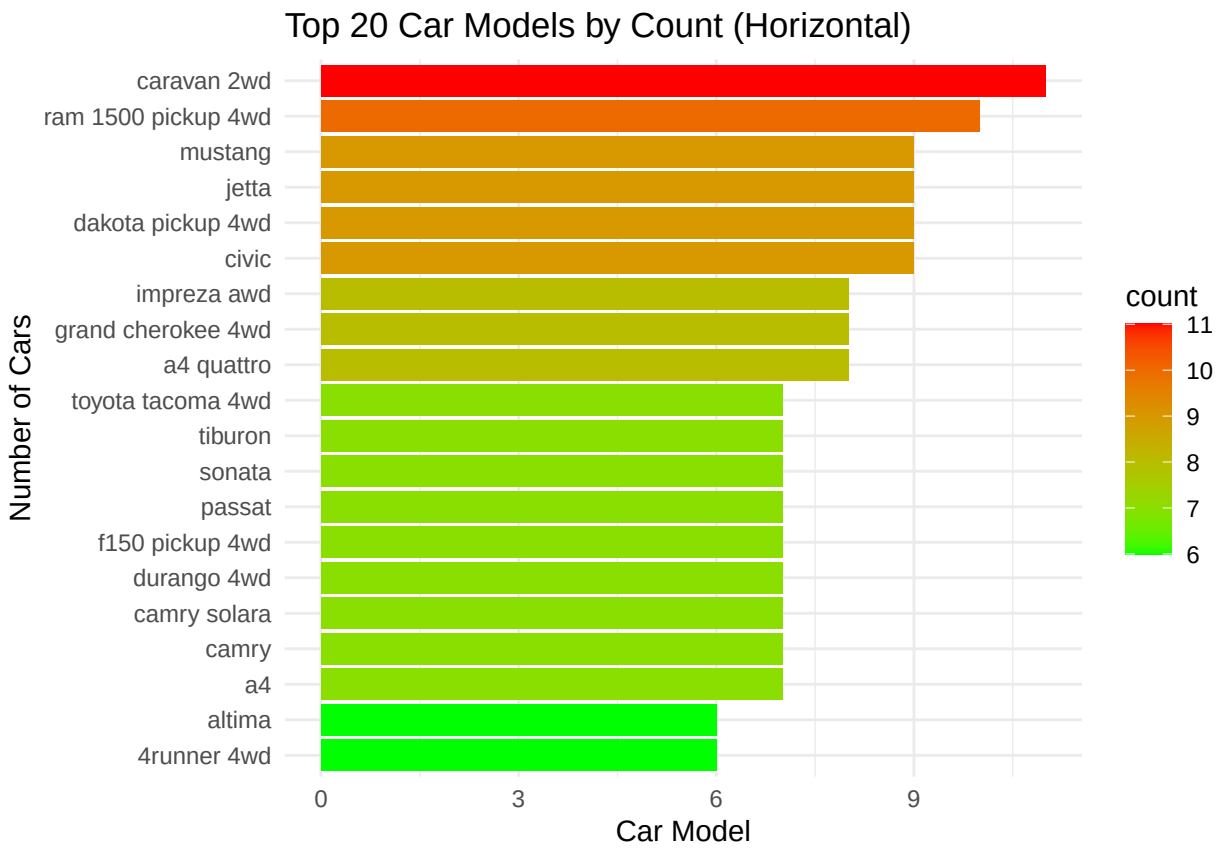
a.

```
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 = "green", high = "red") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



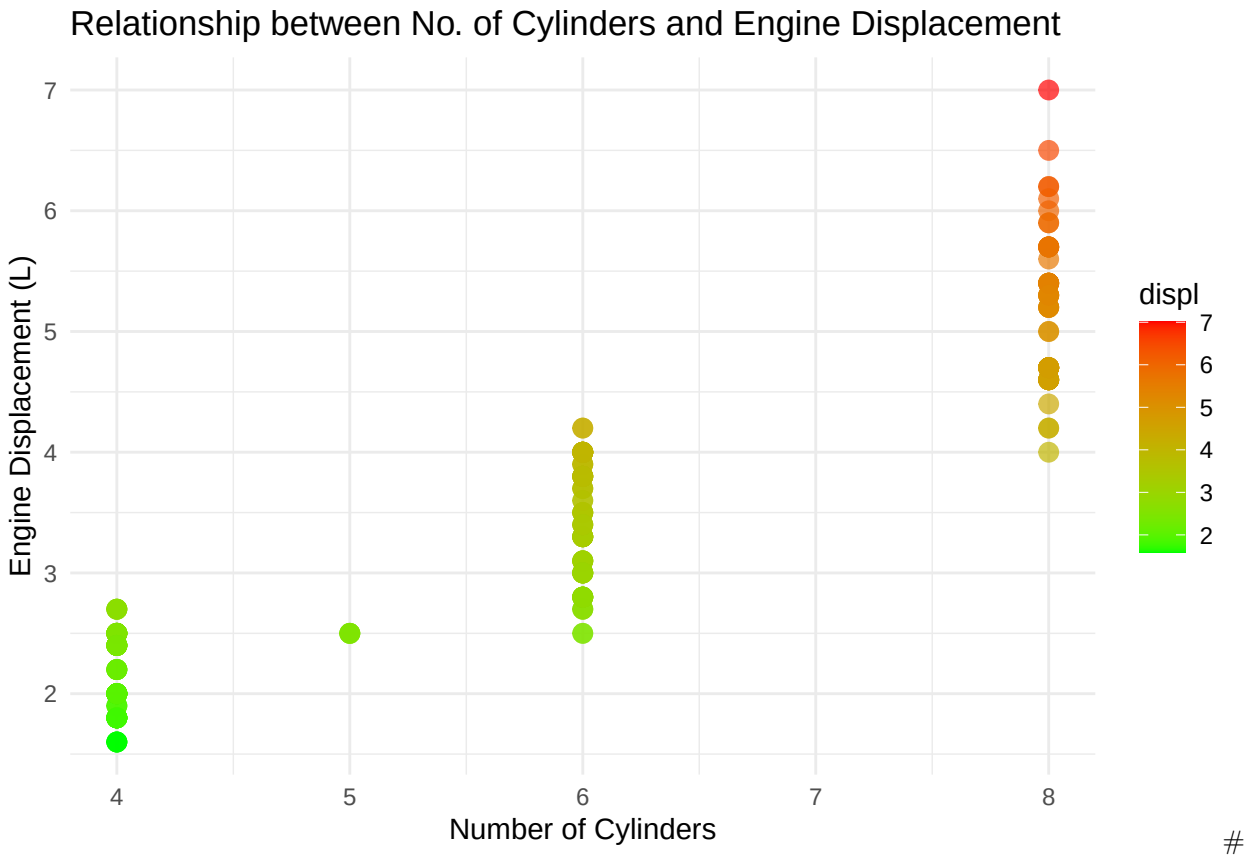
b.

```
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 = "green", high = "red") +  # Color gradient
  theme_minimal() +
  coord_flip()
```



5.

```
ggplot(mpg, 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 = "green", high = "red") +
  theme_minimal()
```

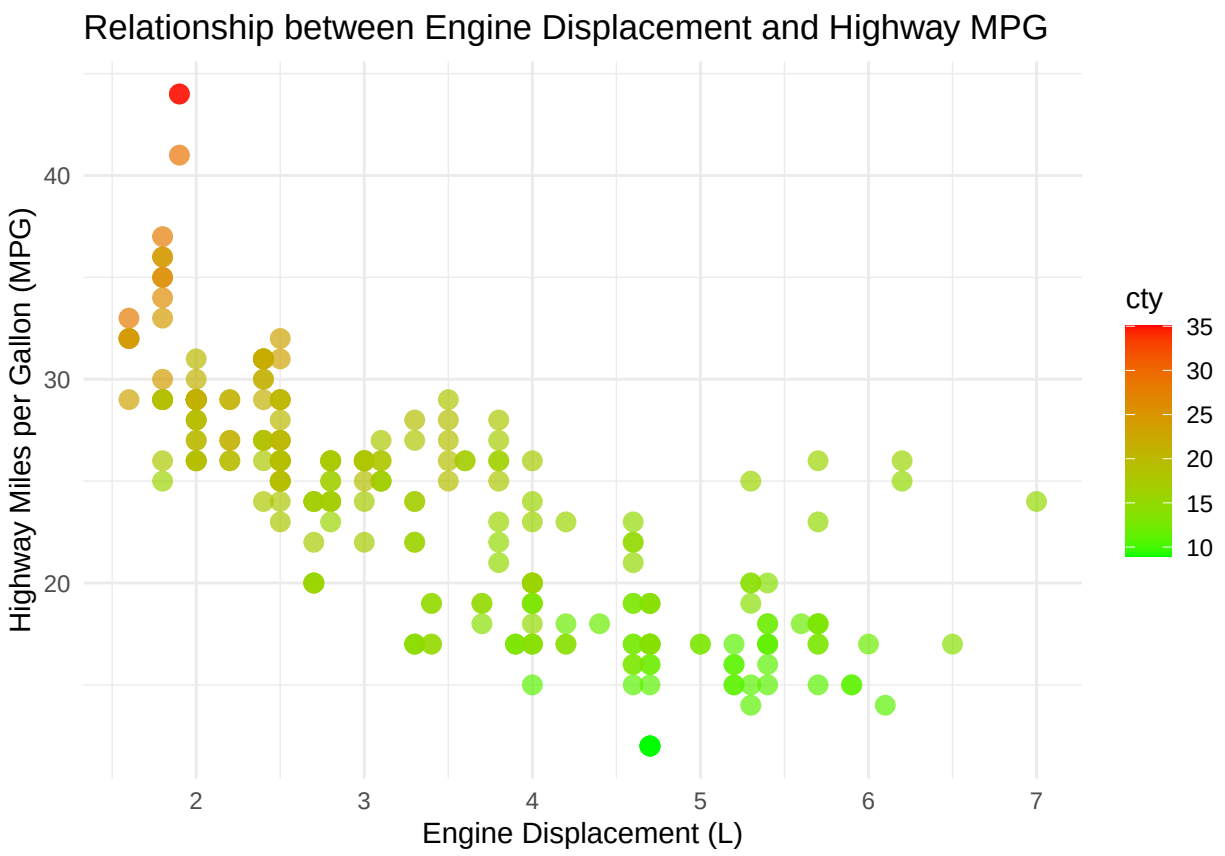


#

a. The more number of cylinders the higher the displacement. It shows a positive relationship.

6.

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



6.

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

a.

```
length(traffic_data)
```

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

```
variable_names <- names(traffic_data)

variable_names
```

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

b.

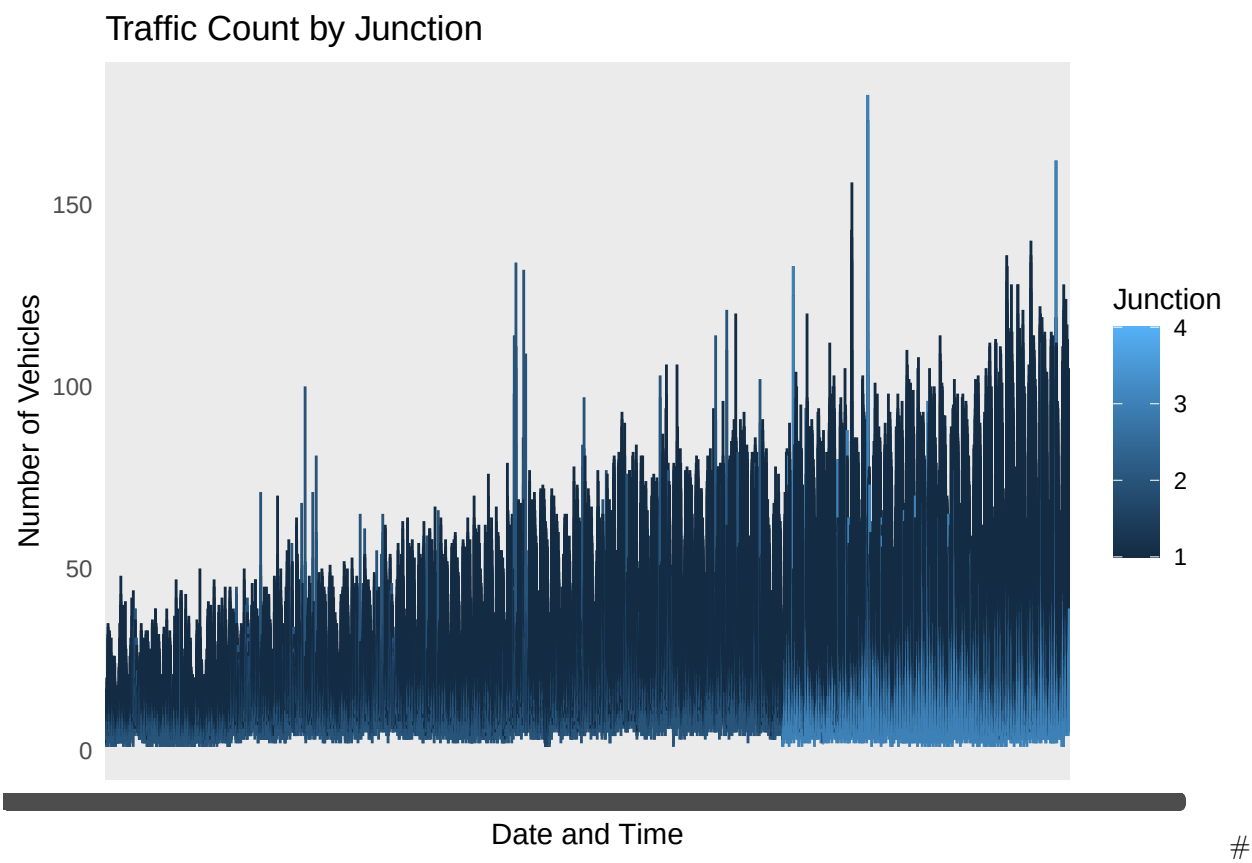
```
unique_junctions <- unique(traffic_data$Junction)

junctions_dataframes <- list()

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.

```
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.

```
library(readxl)
```

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

a.

```r
num_observations <- nrow(alexa_data)

num_columns <- ncol(alexa_data)

num_observations
```

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

```r
num_columns
```

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

b.

```r
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
```

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

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

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

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

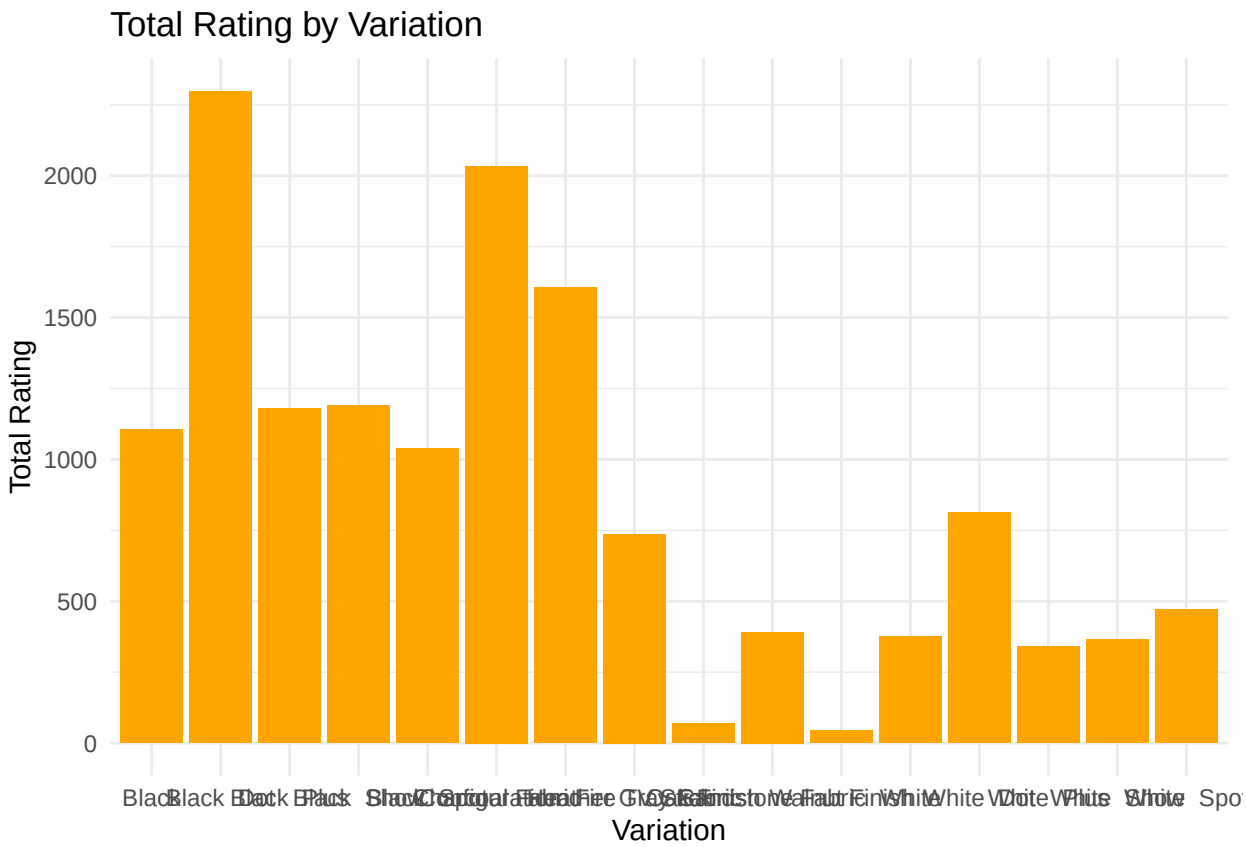
```r
variation_totals <- alexa_data %>%
  group_by(variation) %>%
  summarize(Total_Rating = sum(rating, na.rm = TRUE),
            Total_Verified_Reviews = sum(verified_reviews, na.rm = TRUE))

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
```

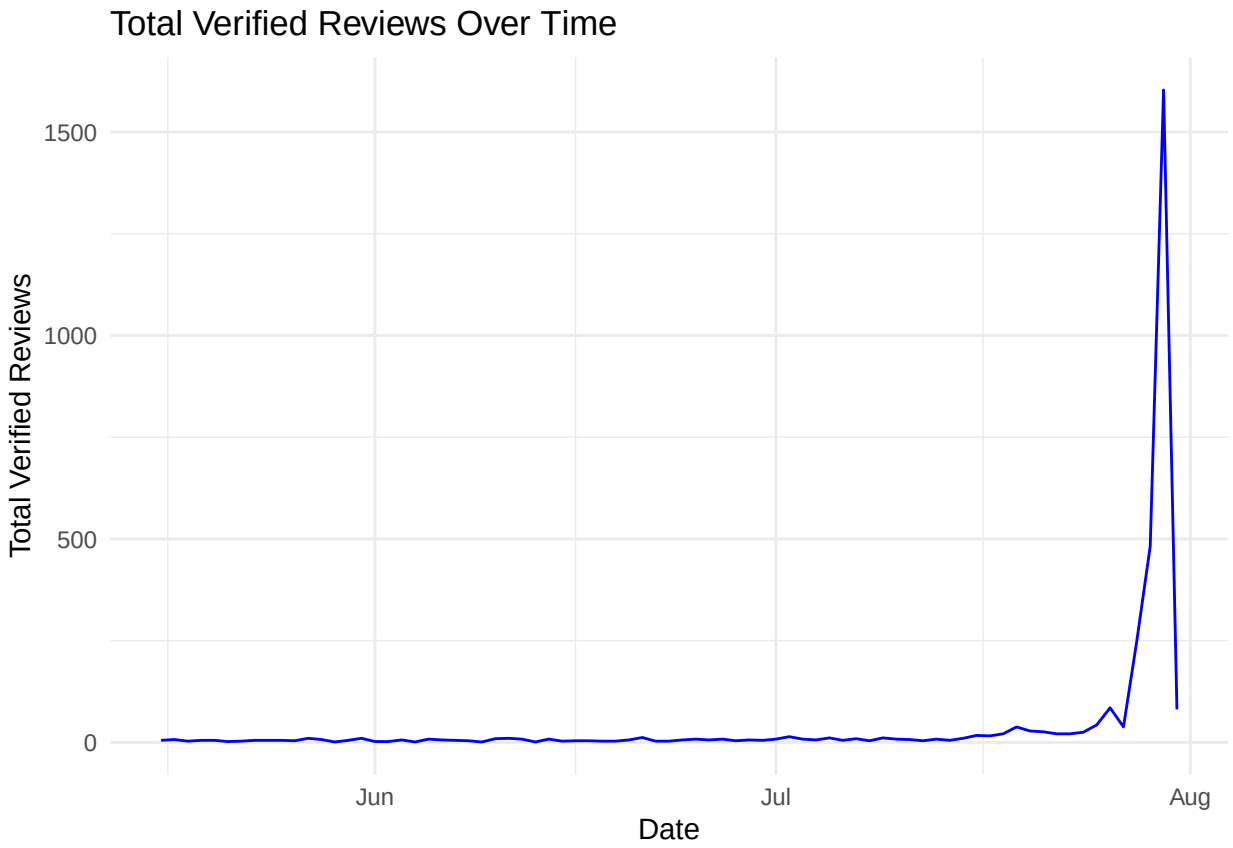
c.

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



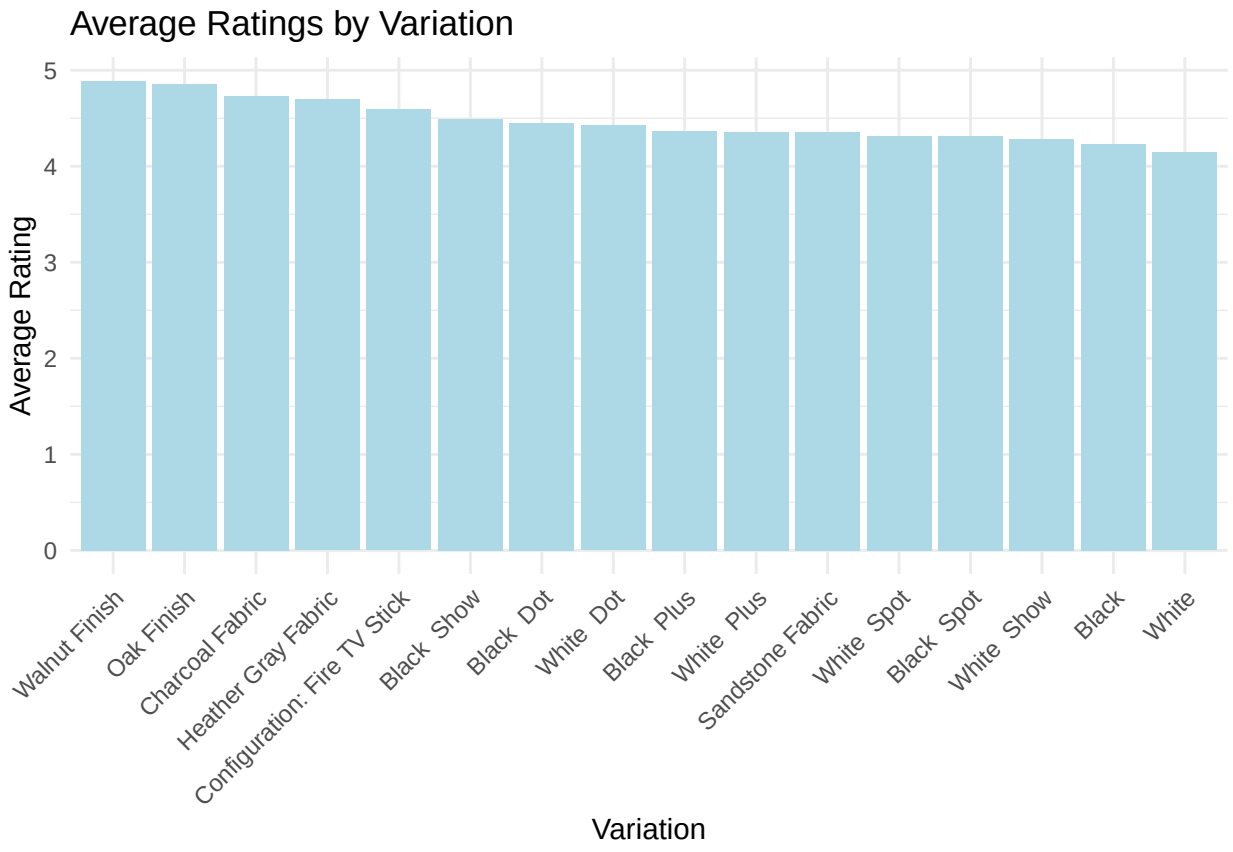
The graph shows that each bar corresponds to the total rating for a specific variation. Although the names on the x axis overlaps. # d.

```
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.

```
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