RWorksheet_Celestra#4c

Kenneth Celestra

2024-11-03

1.

```
library(ggplot2)
data("mpg")
write.csv(mpg, "mpg.csv", row.names = FALSE)
```

a

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

```
##   manufacturer model displ year cyl      trans drv cty hwy fl   class
## 1         audi    a4   1.8 1999   4   auto(l5)   f  18  29  p compact
## 2         audi    a4   1.8 1999   4 manual(m5)   f  21  29  p compact
## 3         audi    a4   2.0 2008   4 manual(m6)   f  20  31  p compact
## 4         audi    a4   2.0 2008   4   auto(av)   f  21  30  p compact
## 5         audi    a4   2.8 1999   6   auto(l5)   f  16  26  p compact
## 6         audi    a4   2.8 1999   6 manual(m5)   f  18  26  p compact
```

b. The Categorical variables in the are:

manufacturer- manufacturer name model- model name trans- type of transmission drv- type of drive train fl- fuel type class- “type” of car

c. The Continuous variables in the mpg dataset are:

displ- engine displacement, in liters year- year of manufacture cyl- number of cylinders cty- city miles per gallon hwy- highway miles per gallon

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
```

```
mostModels <- mpg %>%
  group_by(manufacturer) %>%
  summarize(num_models = n_distinct(model)) %>%
  arrange(desc(num_models)) %>%
  slice(1)
```

```
mostModels
```

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

```
mostVariations<- mpg %>%
  group_by(model) %>%
  summarize(num_variations = n()) %>%
  arrange(desc(num_variations)) %>%
  slice(1)

mostVariations
```

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

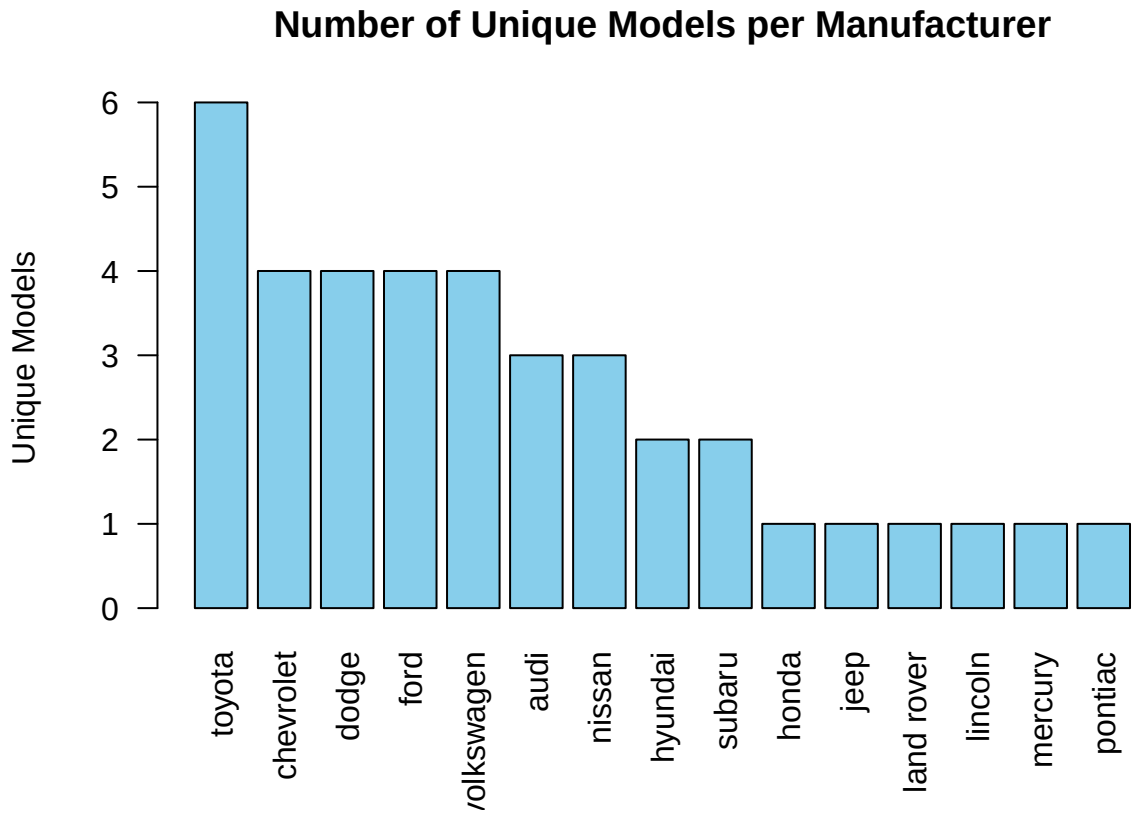
a

```
unique_models <- mpg %>%
  group_by(manufacturer) %>%
  summarize(unique_models = n_distinct(model)) %>%
  arrange(desc(unique_models))

unique_models
```

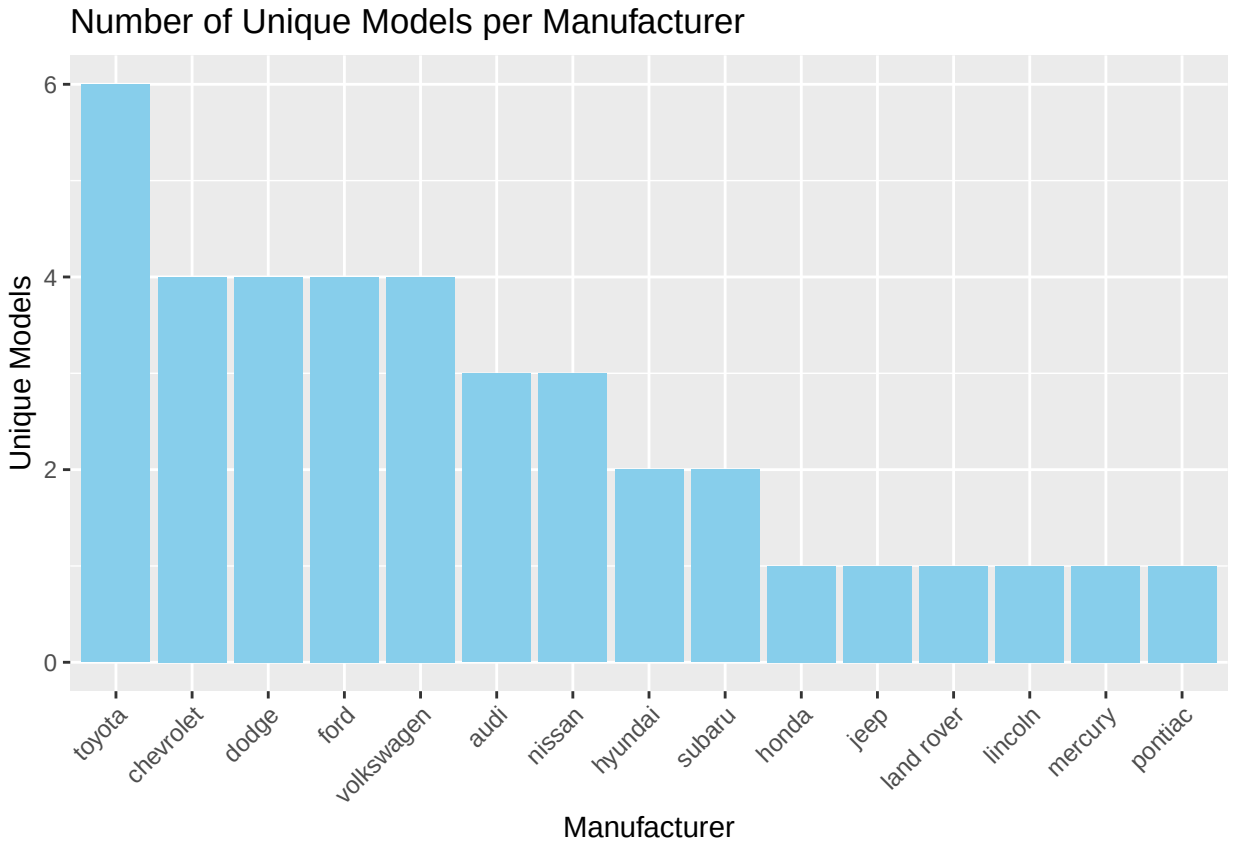
```
## # A tibble: 15 x 2
##    manufacturer unique_models
##    <chr>                <int>
##  1 toyota                   6
##  2 chevrolet                4
##  3 dodge                    4
##  4 ford                     4
##  5 volkswagen               4
##  6 audi                     3
##  7 nissan                   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
```

b

```
barplot(unique_models$unique_models,
        names.arg = unique_models$manufacturer,
        las = 2,
        col = "skyblue",
        main = "Number of Unique Models per Manufacturer",
        ylab = "Unique Models")
```

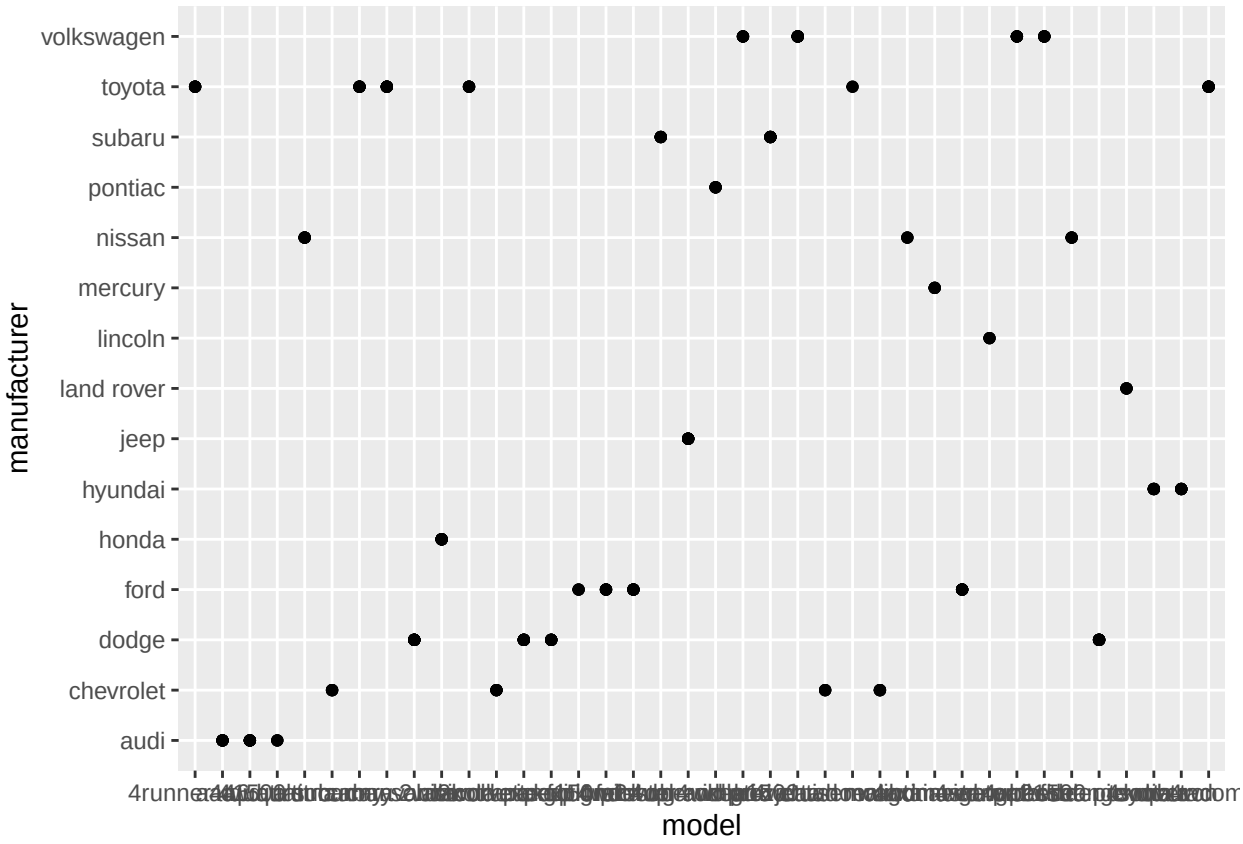



```r
library(ggplot2)

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



2. a It visually displays the distribution and overlap of models among different manufacturers, helping identify how many models each manufacturer has and any common models between them.

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



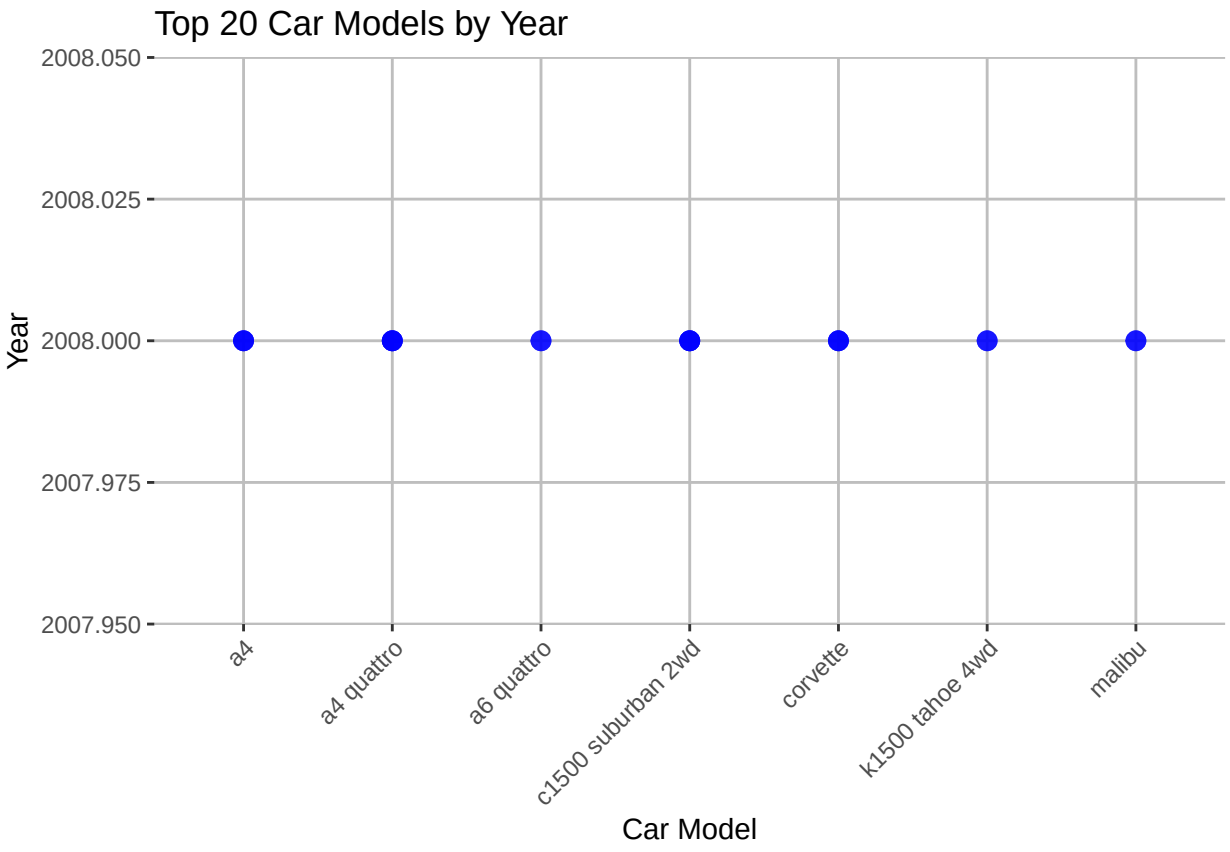
b. The scatter plot created by ggplot(mpg, aes(model, manufacturer)) + geom_point() has limitations, such as overlapping points and lack of context regarding important features like fuel efficiency or engine size. To make it more informative, you could add color to represent a variable like city miles per gallon (cty) and size for engine displacement (displ). Alternatively, using faceting with facet_wrap() can create separate panels for each manufacturer, enhancing clarity. Boxplots or violin plots could effectively show the distribution of continuous variables, such as city MPG, across manufacturers. Finally, employing interactive visualizations with packages like plotly allows users to explore the data dynamically, providing a richer understanding of the relationships in the dataset.

3.

```
top20 <- mpg %>%
  arrange(desc(year)) %>%
  head(20)

ggplot(top20, aes(x = model, y = year)) +
  geom_point(color = "blue", size = 3, alpha = 0.7) +
  labs(title = "Top 20 Car Models by Year",
       x = "Car Model",
       y = "Year") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        panel.background = element_blank(),
        panel.grid.major = element_line(color = "gray", size = 0.5),
        panel.grid.minor = element_blank())
```

```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
```

generated.



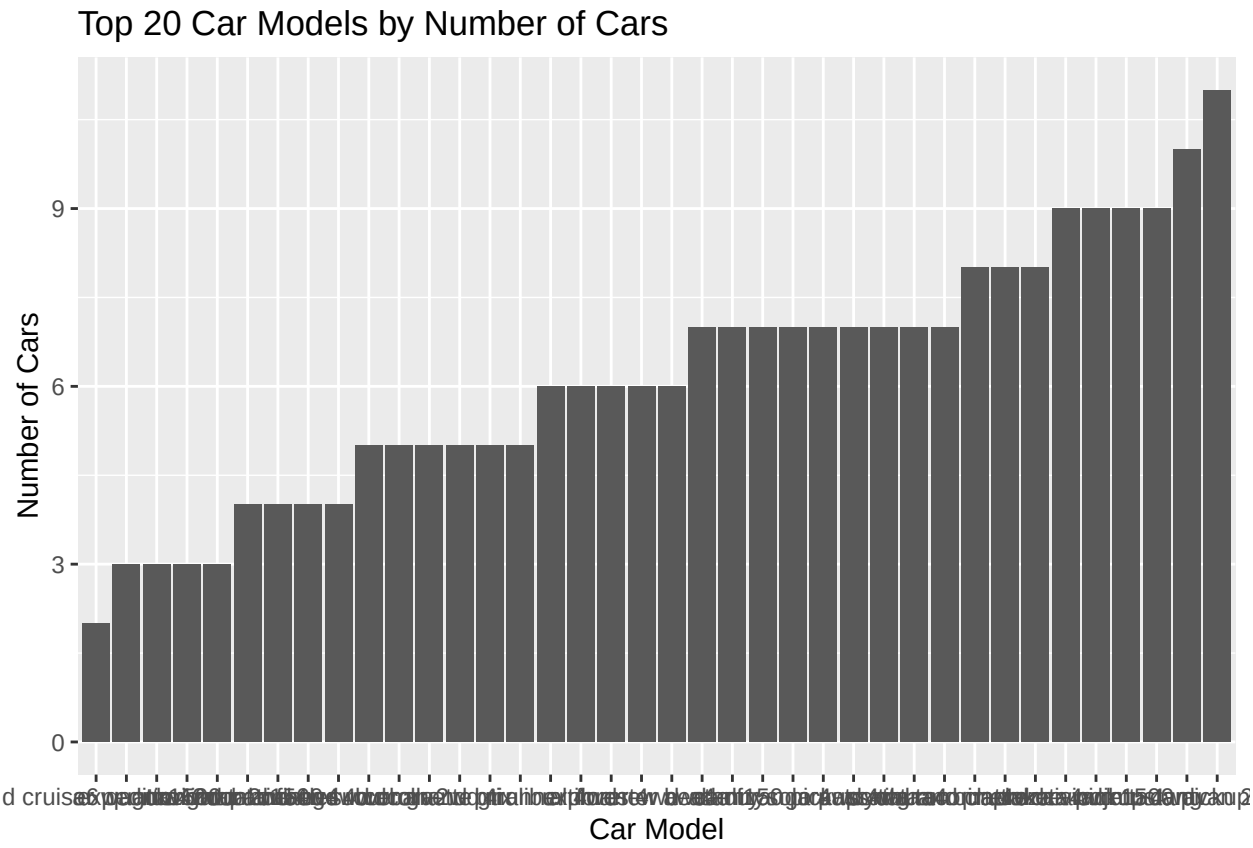
4.

```
carcount<- mpg %>%
  group_by(model) %>%
  summarize(num_cars = n())
carcount
```

```
## # A tibble: 38 x 2
##    model              num_cars
##    <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
```

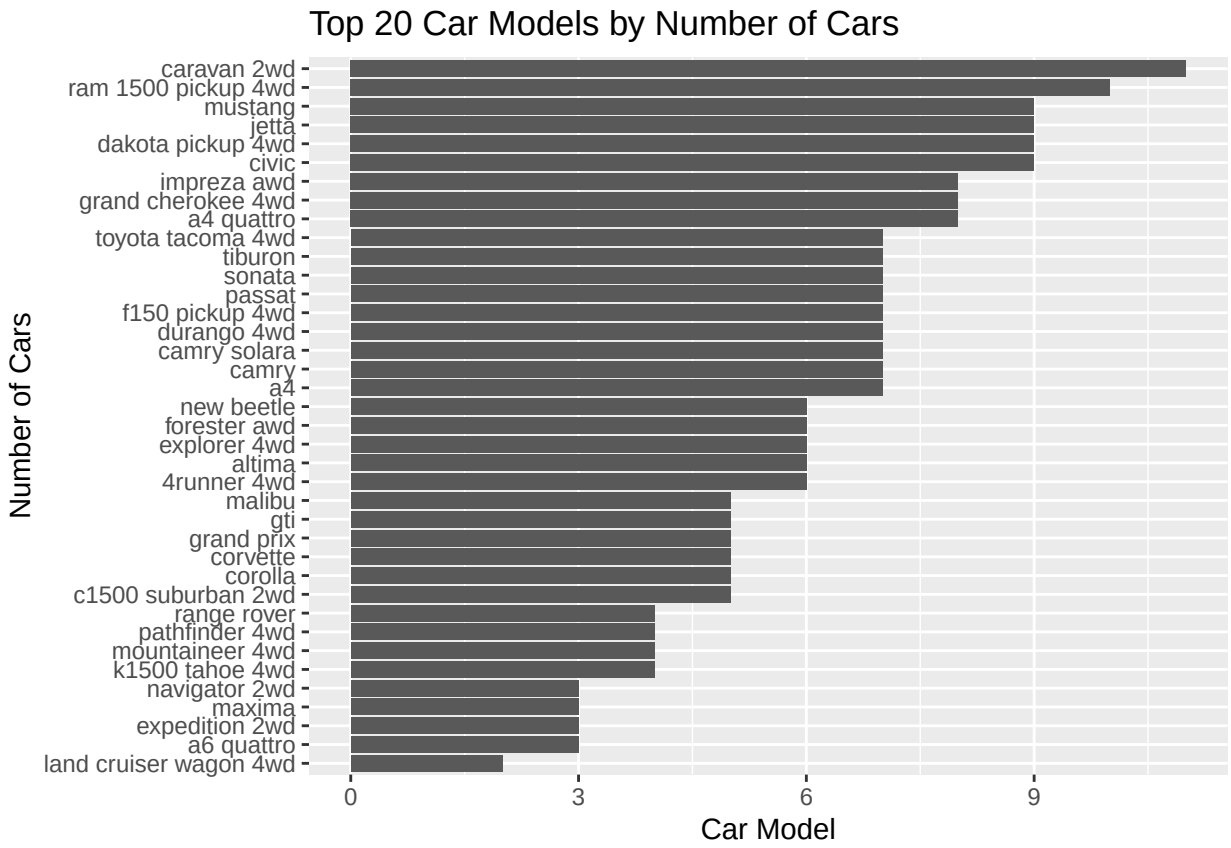
a

```
ggplot(carcount, aes(x = reorder(model, num_cars), y = num_cars)) +
  geom_bar(stat = "identity") +
  labs(title = "Top 20 Car Models by Number of Cars",
       y = "Number of Cars",
       x = "Car Model")
```



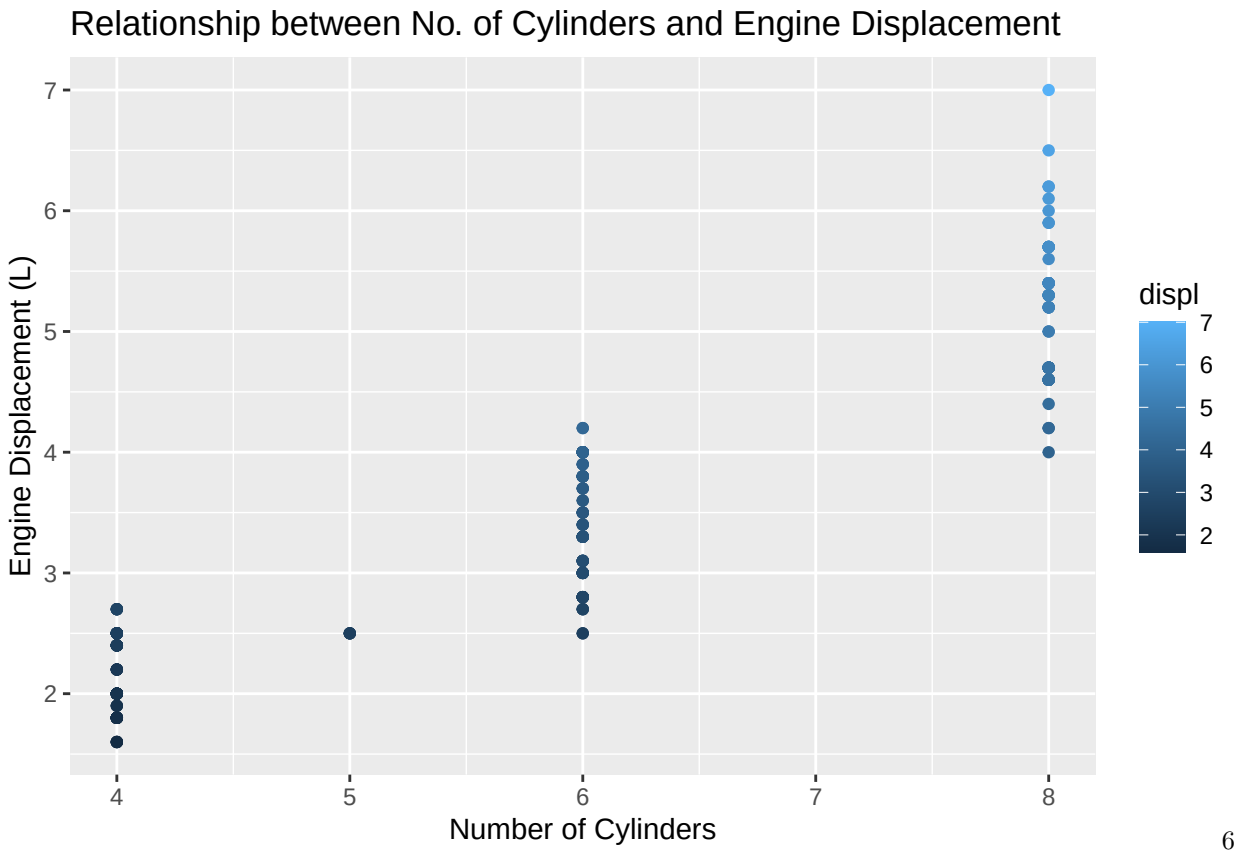
b

```
ggplot(carcount, aes(x = reorder(model, num_cars), y = num_cars)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 20 Car Models by Number of Cars",
       x = "Number of Cars",
       y = "Car Model")
```



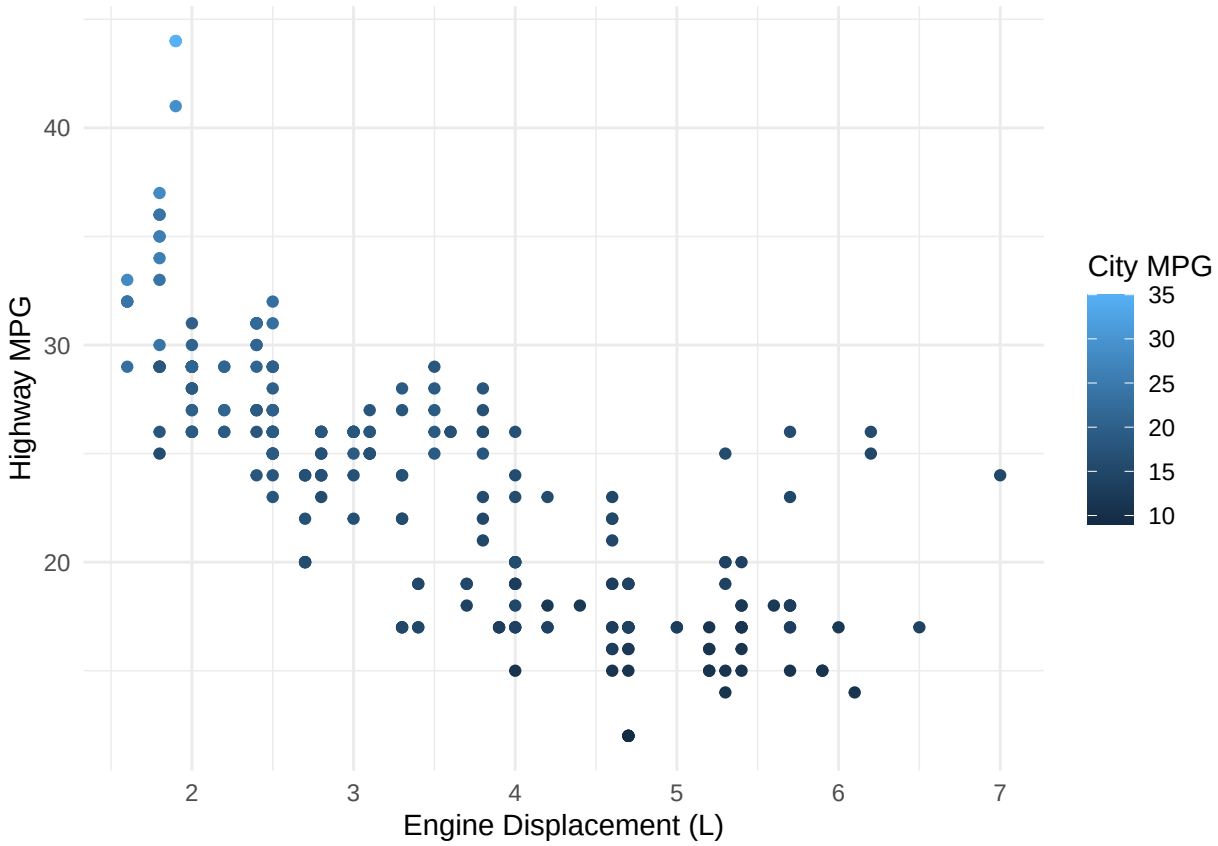
5. a The scatter plot between the number of cylinders (cyl) and engine displacement (displ) generally shows a positive correlation, indicating that as the number of cylinders increases, engine displacement also tends to rise. Vehicles with more cylinders typically have larger engines, which is reflected in the clustering of points. While most vehicles follow this trend, there may be outliers—such as those with fewer cylinders but larger displacement—due to advanced engineering or turbocharging. Overall, this relationship underscores how engine design commonly associates a higher cylinder count with increased displacement, suggesting greater power and fuel consumption potential.

```
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)")
```



6

```
library(ggplot2)

ggplot(mpg, aes(x = displ, y = hwy, color = cty)) +
  geom_point() +
  labs(
       x = "Engine Displacement (L)",
       y = "Highway MPG",
       color = "City MPG") +
  theme_minimal()
```



There is negative correlation between displ and hwy. As engine displacement increases, highway miles per gallon (MPG) tends to decrease. This trend reflects that larger engines generally consume more fuel and have lower MPG.

This output occurs because larger engines (higher displacement) usually require more fuel, resulting in lower MPG values. By mapping cty as the color, we gain additional insight into fuel efficiency: vehicles with better highway MPG often have better city MPG as well. This correlation is a result of engine design and efficiency standards that impact fuel consumption across both highway and city driving conditions.

6. a

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

#number of observations
dim(traffic)
```

```
## [1] 48120     4
```

```
#variables in the dataset
names(traffic)
```

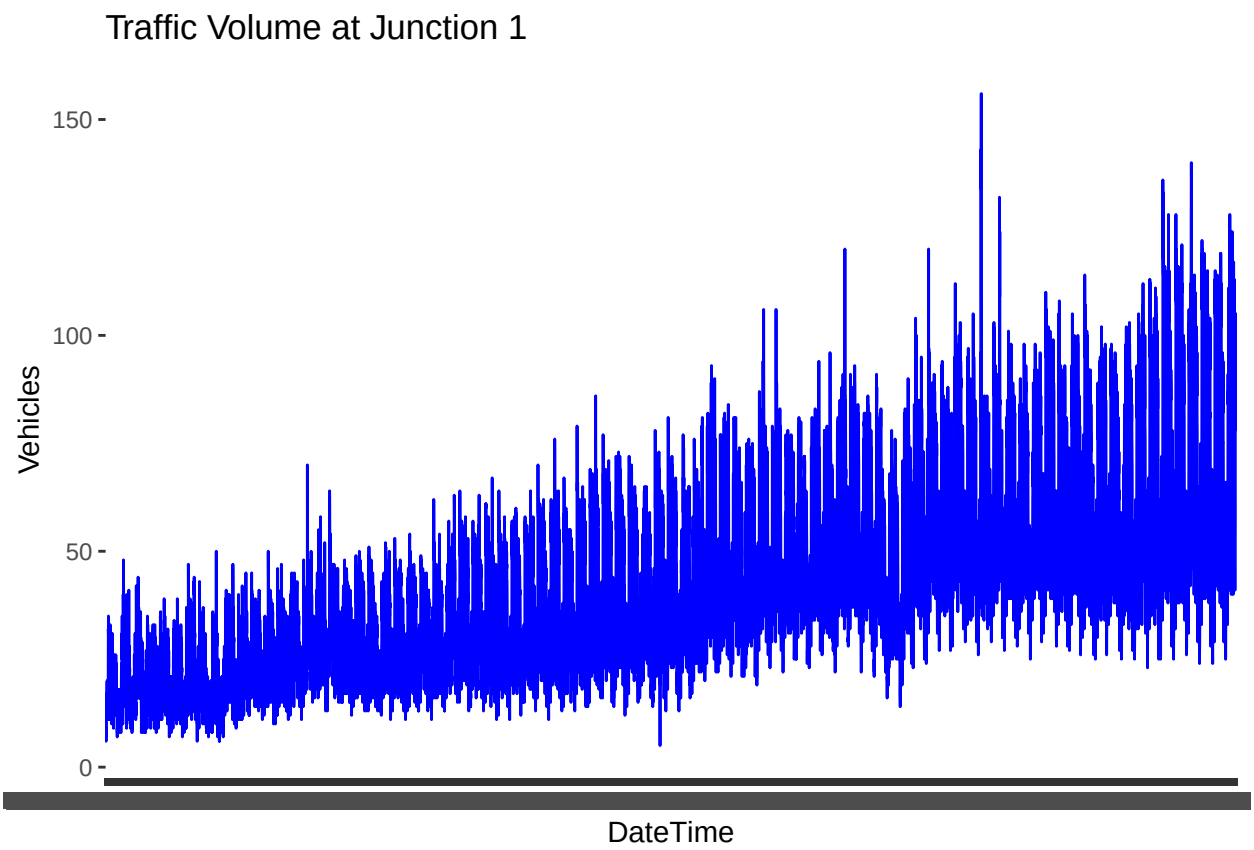
```
## [1] "DateTime" "Junction" "Vehicles" "ID"
```

b

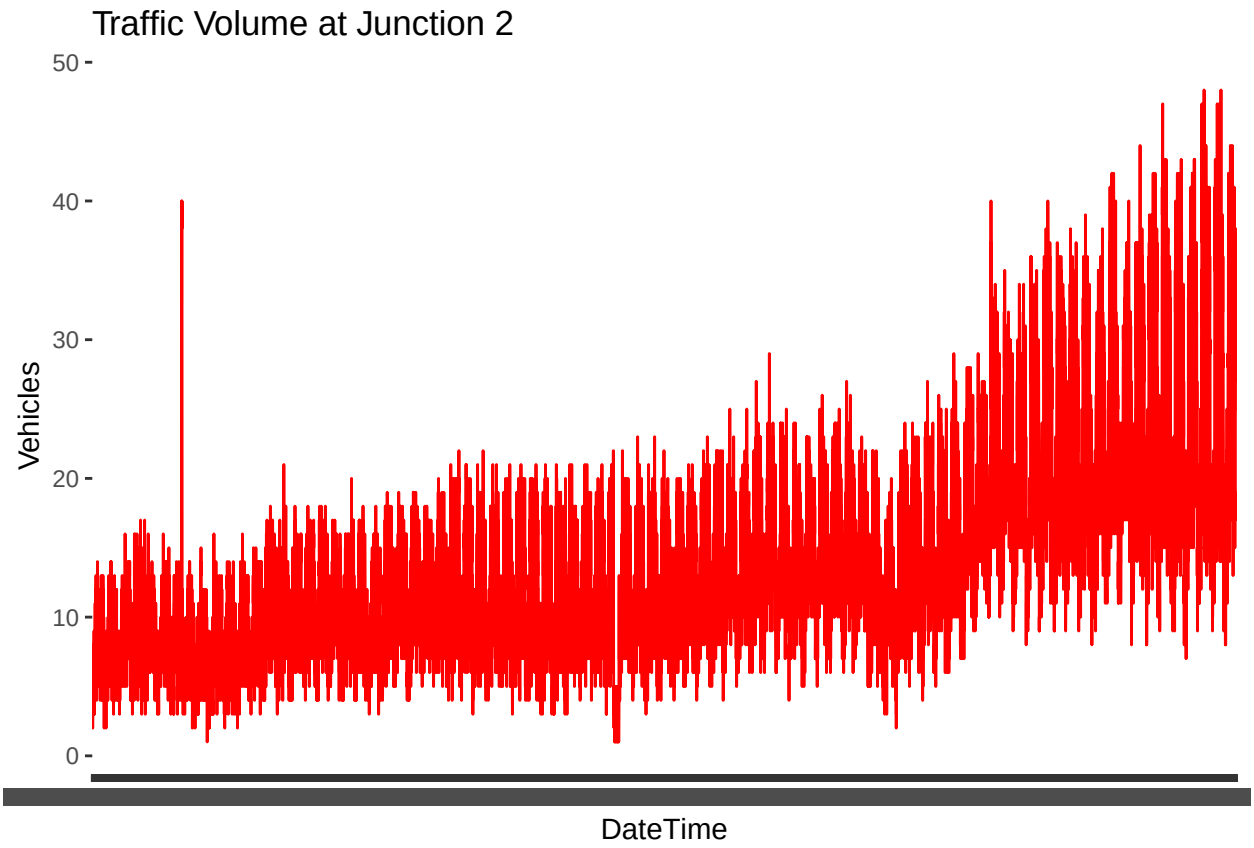
```
junction_data1 <- subset(traffic, Junction == "1")
junction_data2 <- subset(traffic, Junction == "2")
junction_data3 <- subset(traffic, Junction == "3")
junction_data4 <- subset(traffic, Junction == "4")
```

c

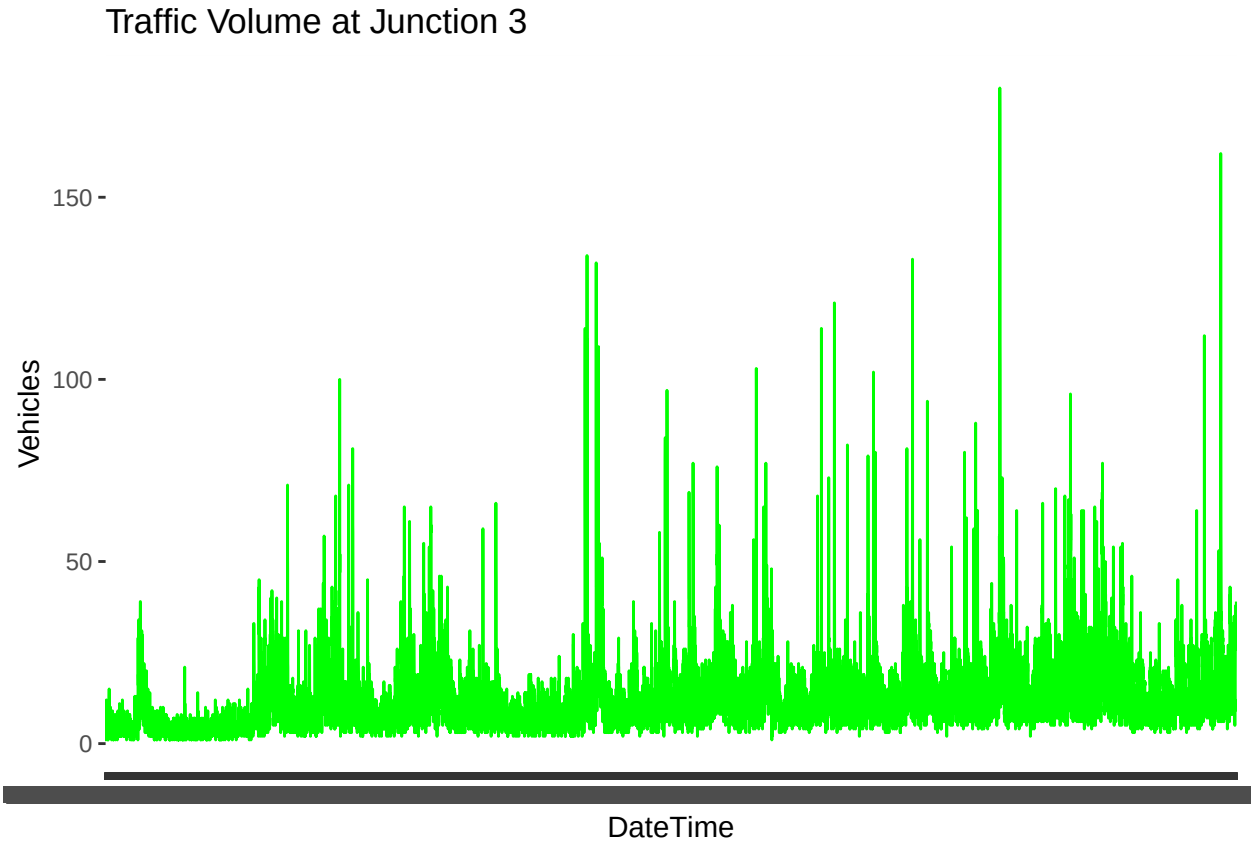
```r
# Plot for Junction 1
plot1 <- ggplot(junction_data1, aes(x = DateTime, y = Vehicles, group = Junction)) +
  geom_line(color = "blue") +
  labs(title = "Traffic Volume at Junction 1", x = "DateTime", y = "Vehicles")

# Plot for Junction 2
plot2 <- ggplot(junction_data2, aes(x = DateTime, y = Vehicles, group = Junction)) +
  geom_line(color = "red") +
  labs(title = "Traffic Volume at Junction 2", x = "DateTime", y = "Vehicles")

# Plot for Junction 3
plot3 <- ggplot(junction_data3, aes(x = DateTime, y = Vehicles, group = Junction)) +
  geom_line(color = "green") +
  labs(title = "Traffic Volume at Junction 3", x = "DateTime", y = "Vehicles")

# Plot for Junction 4
plot4 <- ggplot(junction_data4, aes(x = DateTime, y = Vehicles, group = Junction)) +
  geom_line(color = "purple") +
  labs(title = "Traffic Volume at Junction 4", x = "DateTime", y = "Vehicles")

plot1
```



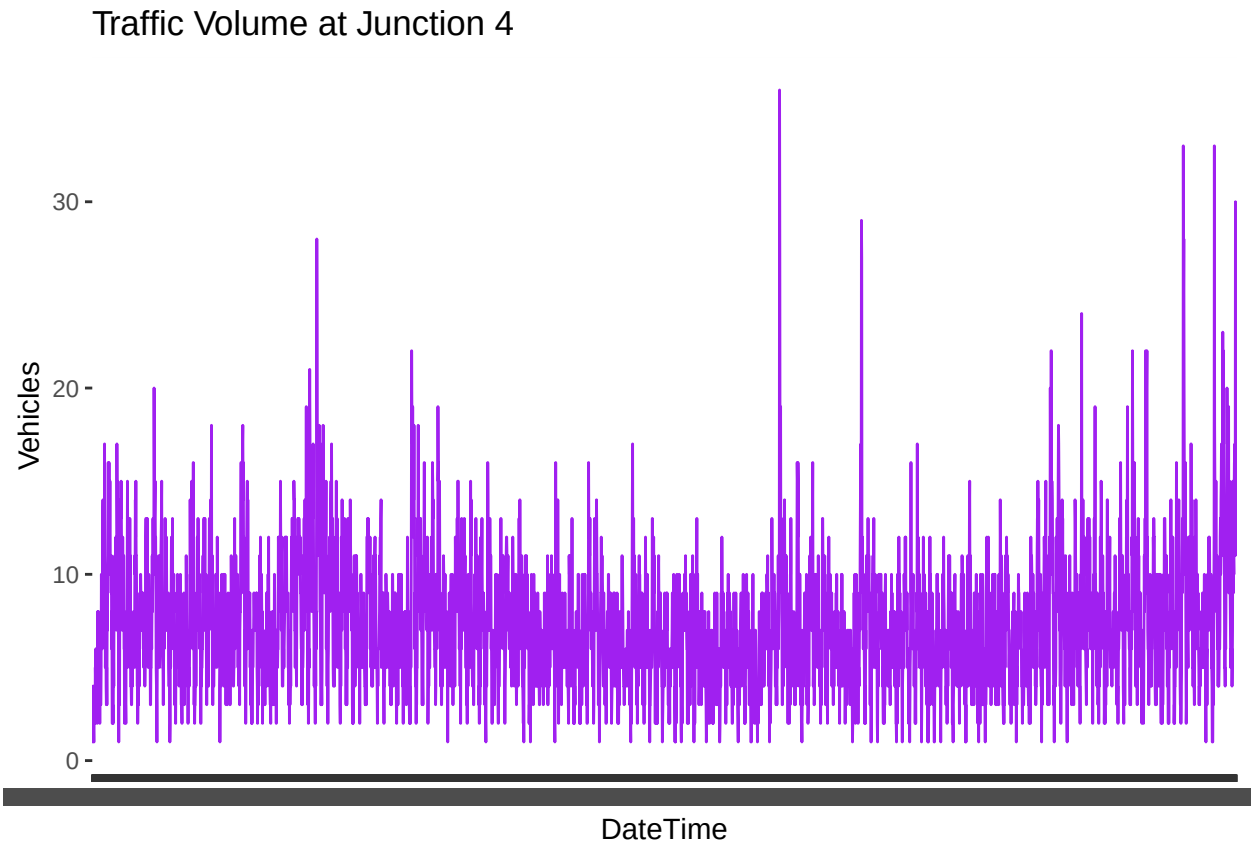

```r
plot2
```


Traffic Volume at Junction 2
50
40
30
20
10
0
Vehicles
DateTime

```
plot3
```




```
plot4
```



7. a

```
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.

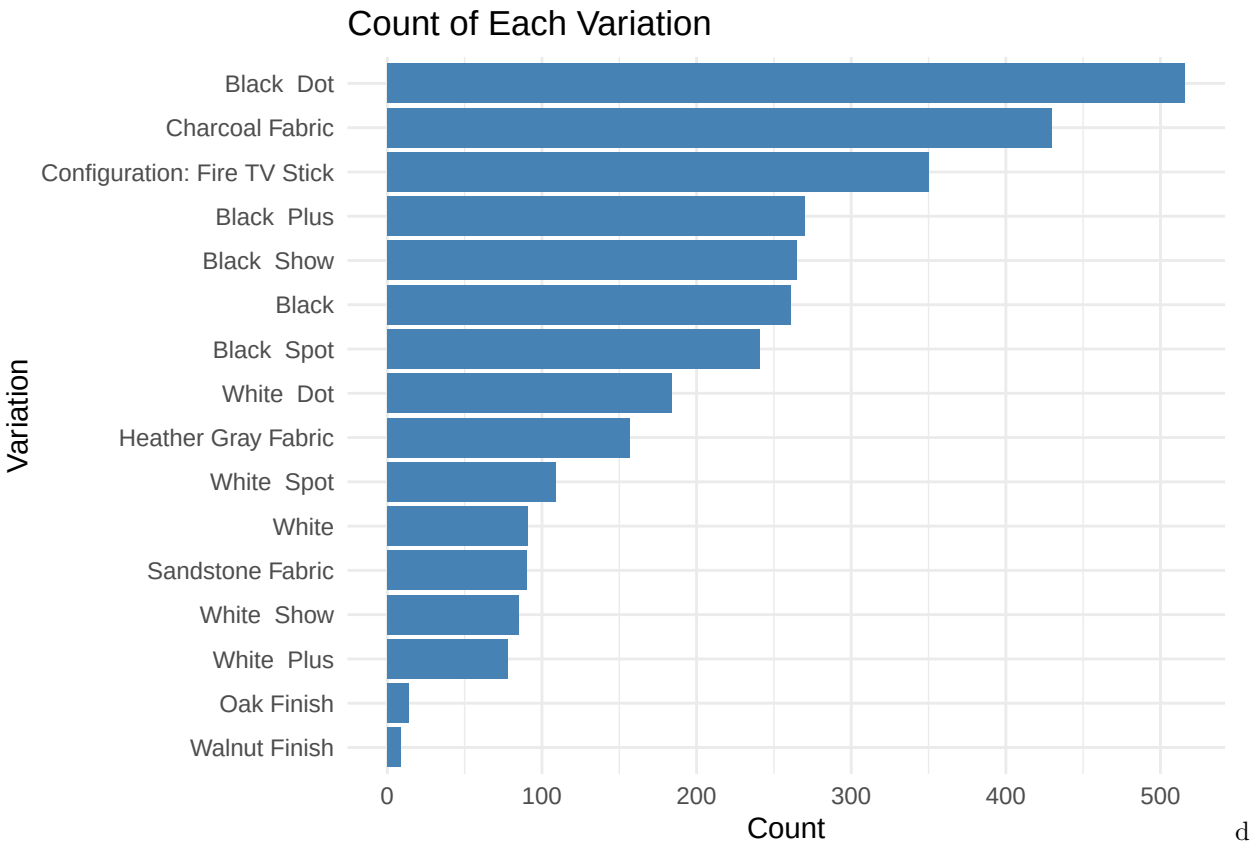
```
library(dplyr)

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
```

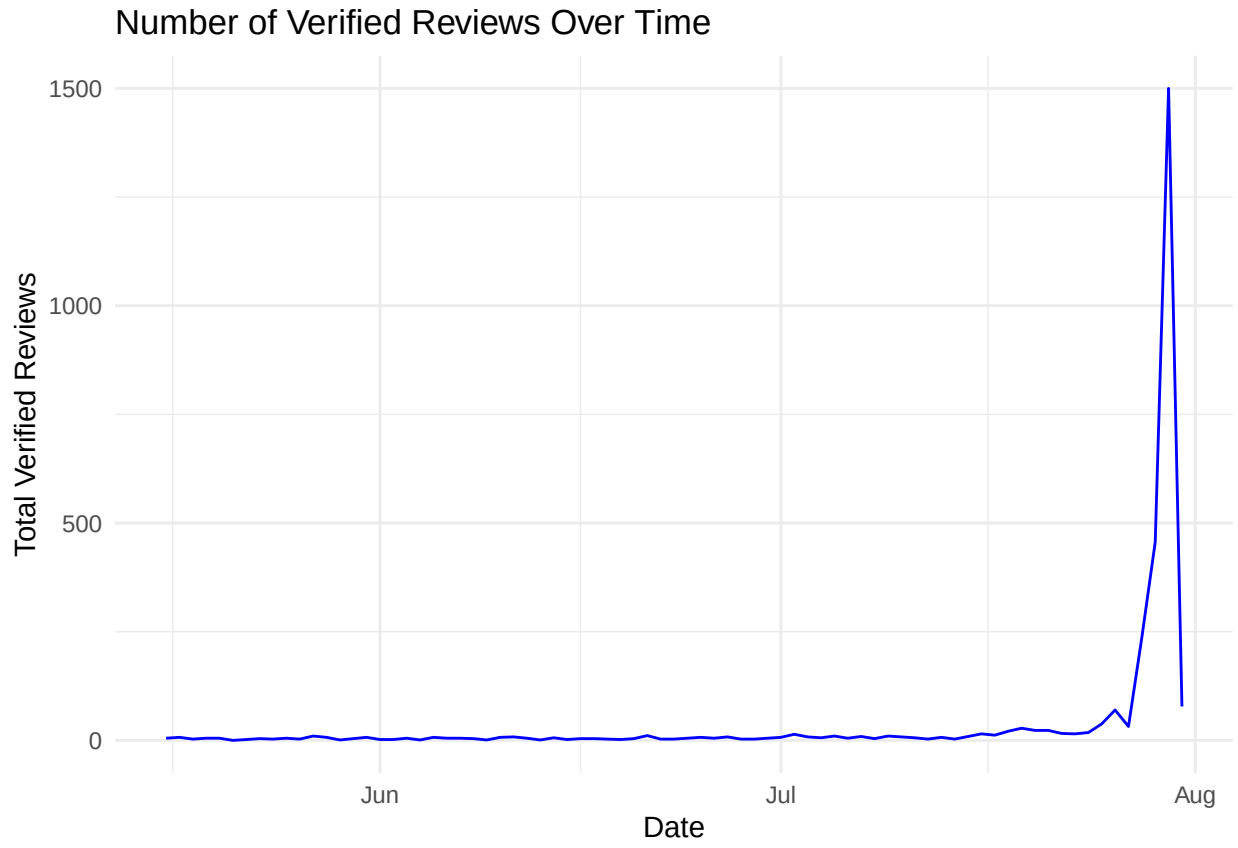
c

```r
library(ggplot2)

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



d

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

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

# Plot the data
ggplot(daily_reviews, aes(x = date, y = total_verified_reviews)) +
  geom_line(color = "blue") +
  labs(title = "Number of Verified Reviews Over Time",
       x = "Date",
       y = "Total Verified Reviews") +
  theme_minimal()
```



e

```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)
```

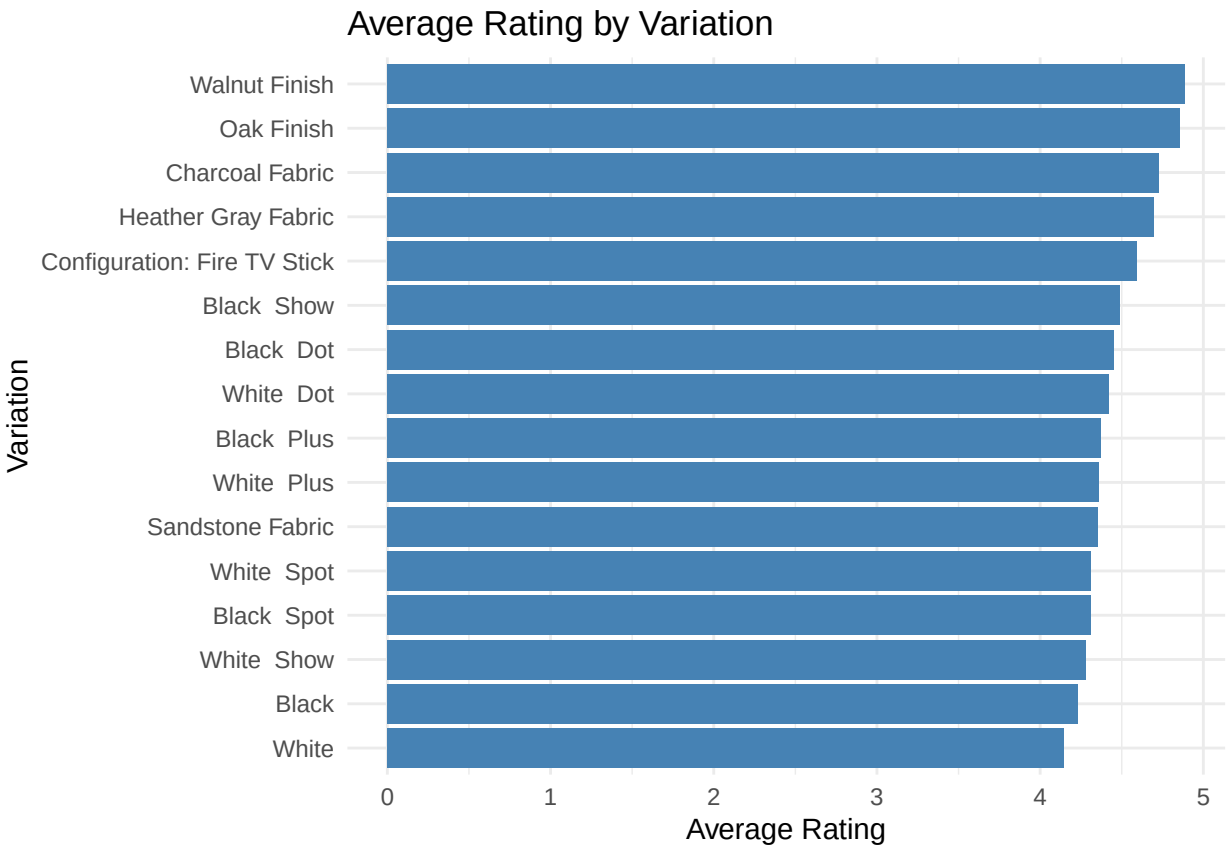
```
## # A tibble: 16 x 2
##    variation                    Average_Rating
##    <chr>                                 <dbl>
##  1 Walnut Finish                          4.89
##  2 Oak Finish                             4.86
##  3 Charcoal Fabric                        4.73
```

```
##  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
```

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

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

```r
ggplot(variation_ratings, aes(x = reorder(variation, Average_Rating), y = Average_Rating)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  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