RWorksheet_Pabriaga#4c

John Martin S. Pabriaga

2024-10-30

```
#1.a
mpg_data <- read.table("mpg.csv", header = TRUE, sep = ",")
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" ...
```

#1.b In the mpg dataset, the manufacturer, model, trans (transmission type), drv (drive type), fl (fuel type) and class (vehicle class) are categorical variables.

#1.c

In the mpg dataset, the displ (engine displacement), cty and hwy are continuous variables.

```
#2.
manufacturer_counts <- table(mpg_data$manufacturer)
most_models_manufacturer <- names(which.max(manufacturer_counts))
most_models_count <- max(manufacturer_counts)

model_counts <- table(mpg_data$model)
most_variations_model <- names(which.max(model_counts))
most_variations_count <- max(model_counts)

most_models_manufacturer
```

```
## [1] "dodge"
```

```
most_models_count
```

```
## [1] 37
```

```
most_variations_model
```

```
## [1] "caravan 2wd"
```

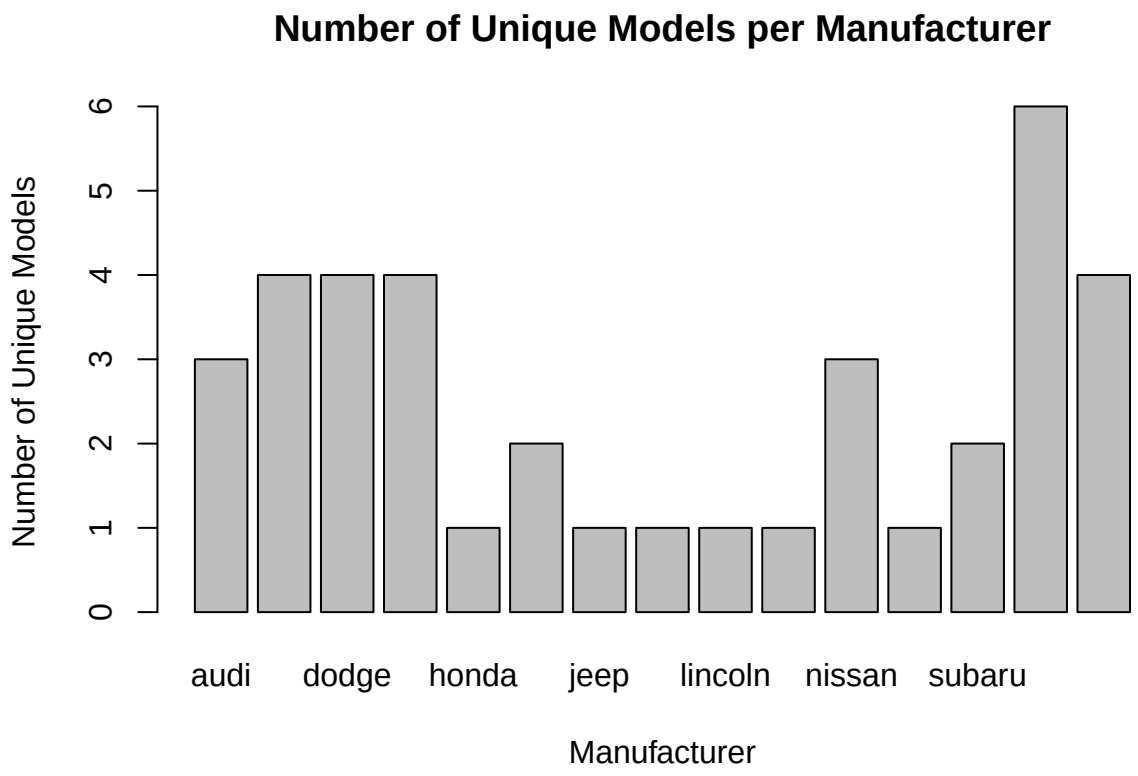
```r
most_variations_count
```

```
## [1] 11
```

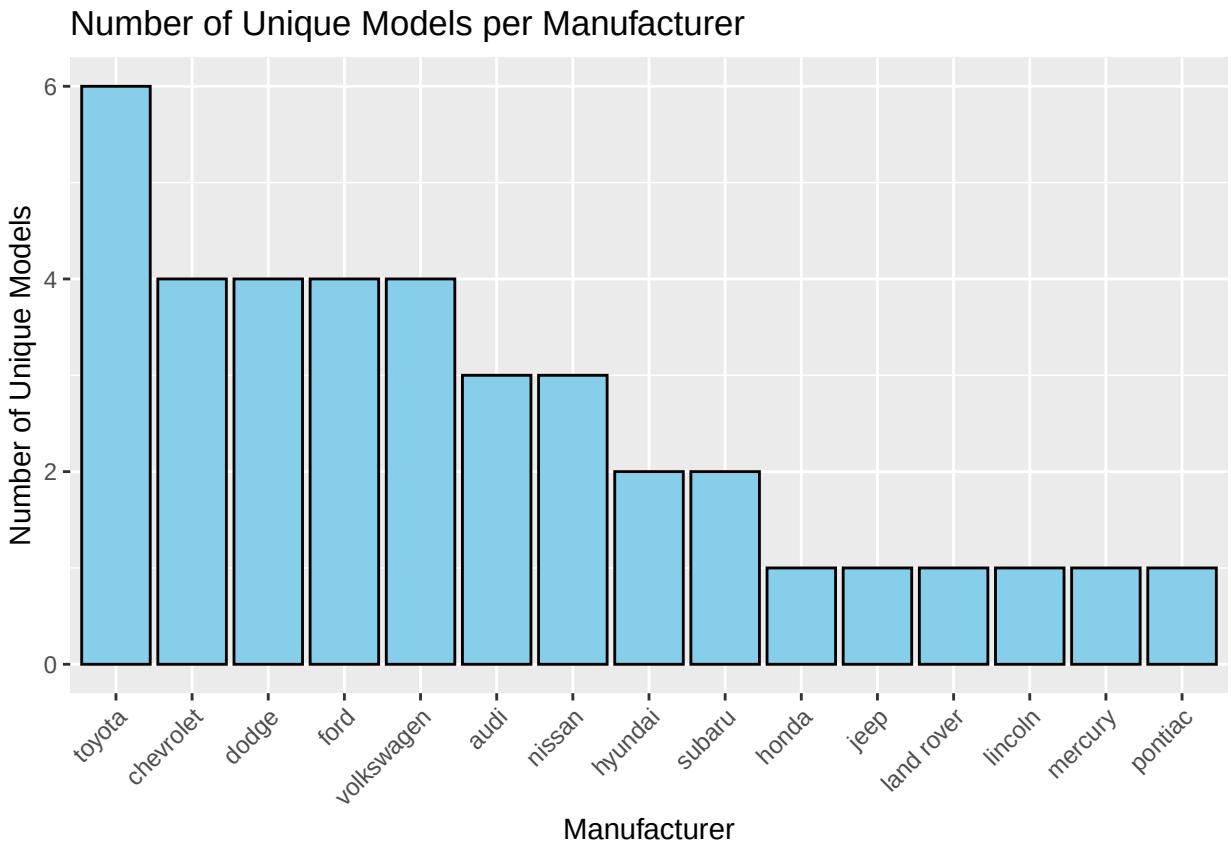
```r
#2.a
unique_models_by_manufacturer <- aggregate(model ~ manufacturer,
                                           data = mpg_data, function(x) length(unique(x)))

unique_models_by_manufacturer
```

```
##    manufacturer model
## 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
```

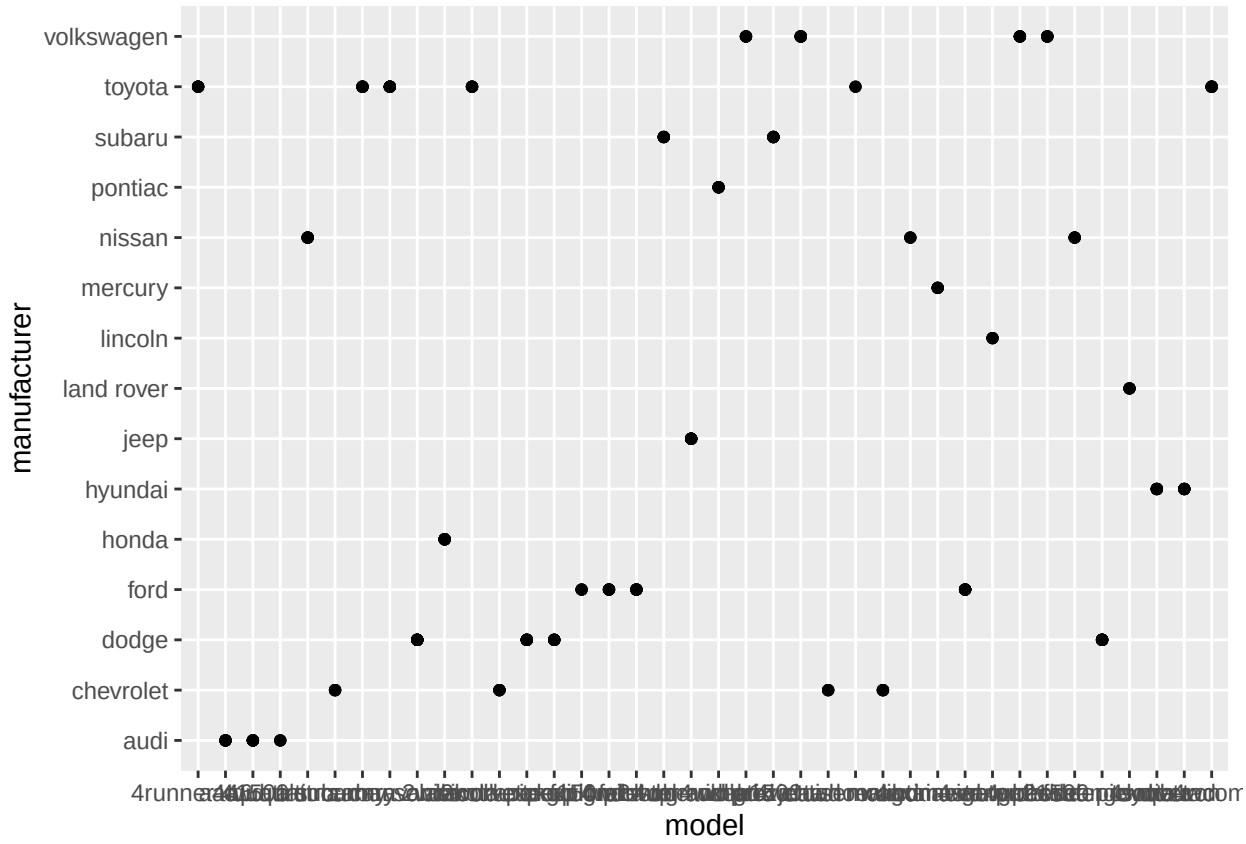
```r
#2.b
barplot(unique_models_by_manufacturer$model,
        names.arg = unique_models_by_manufacturer$manufacturer,
        main = "Number of Unique Models per Manufacturer",
        xlab = "Manufacturer", ylab = "Number of Unique Models")
```




```r
# Using ggplot2

library(ggplot2)
ggplot(unique_models_by_manufacturer, aes(x = reorder(manufacturer, -model), y = model)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  labs(title = "Number of Unique Models per Manufacturer", x = "Manufacturer",
       y = "Number of Unique Models") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
#2.a
ggplot(mpg, aes(model, manufacturer)) + geom_point()
```



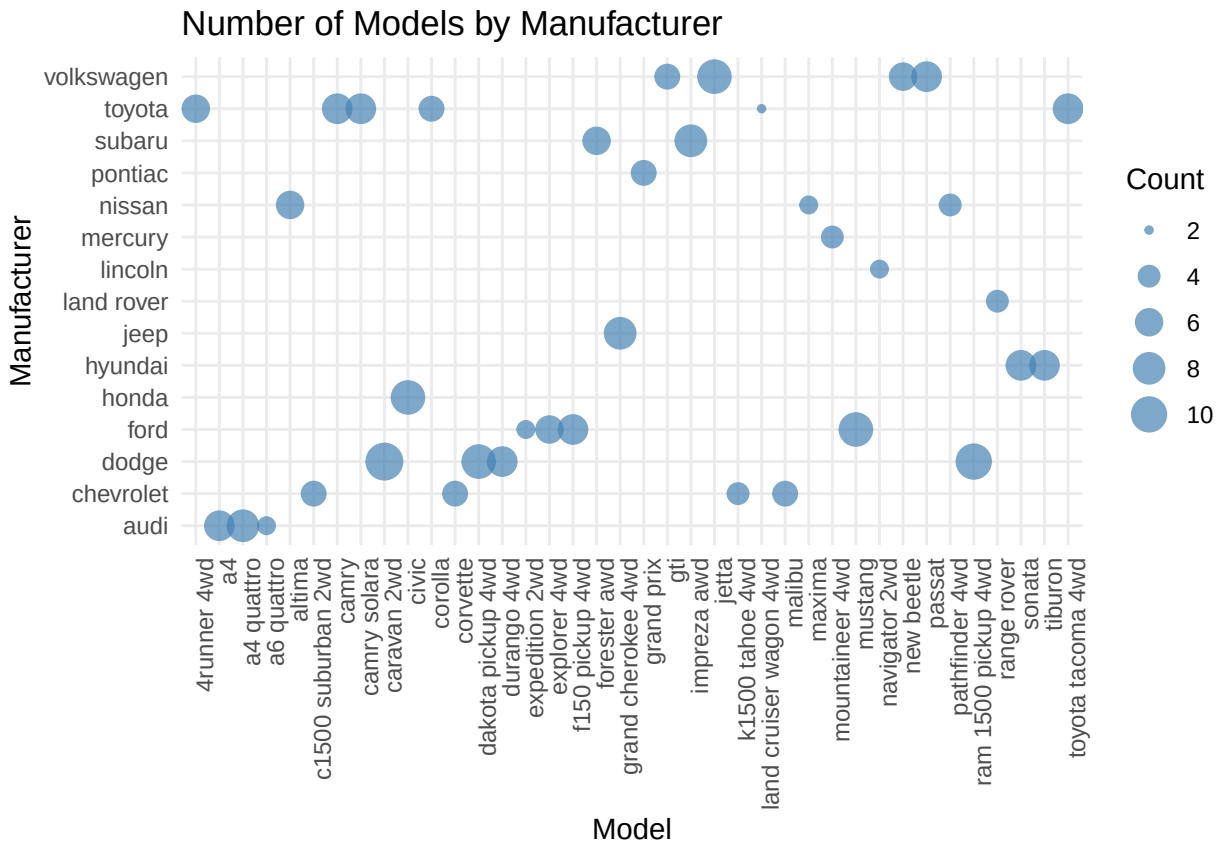

```
#2.b
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

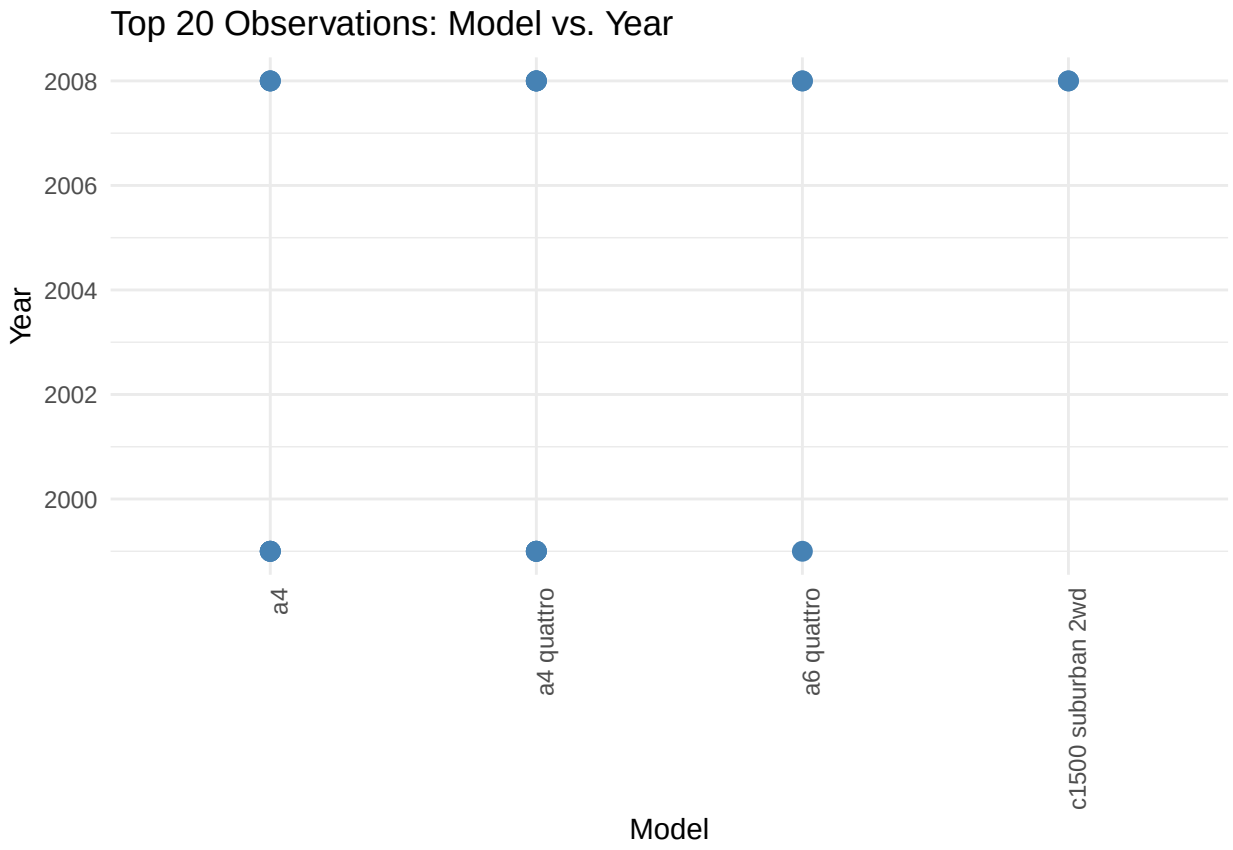
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
model_counts <- mpg_data %>%
  group_by(manufacturer, model) %>%
  summarise(count = n(), .groups = "drop")

ggplot(model_counts, aes(x = model, y = manufacturer, size = count)) +
  geom_point(color = "steelblue", alpha = 0.7) +
  theme_minimal() +
  labs(title = "Number of Models by Manufacturer",
       x = "Model",
       y = "Manufacturer",
       size = "Count") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```r
#3.
top_20_data <- mpg_data %>% slice_head(n = 20)

ggplot(top_20_data, aes(x = model, y = year)) +
  geom_point(color = "steelblue", size = 3) +
  theme_minimal() +
  labs(title = "Top 20 Observations: Model vs. Year",
       x = "Model",
       y = "Year") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

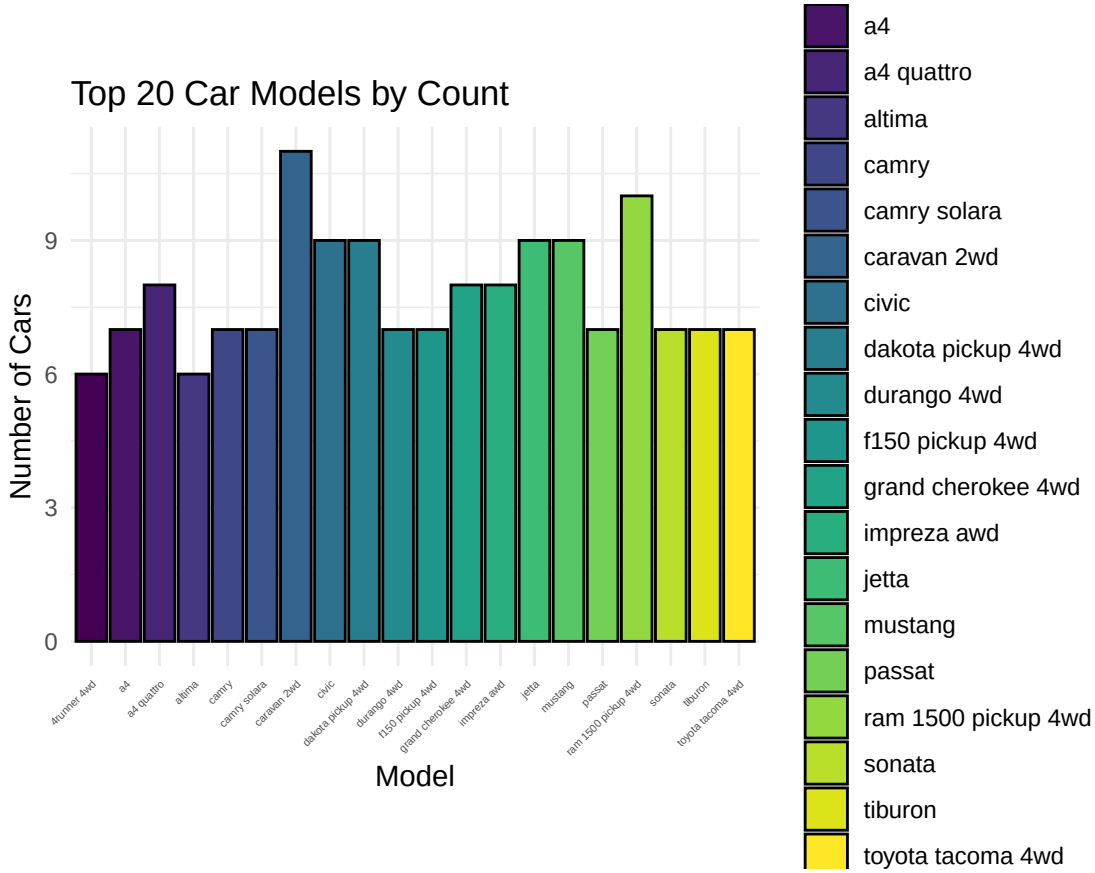



```
#4.

cars_per_model <- mpg_data %>%
  group_by(model) %>%
  summarise(count = n(), .groups = "drop")

cars_per_model
```

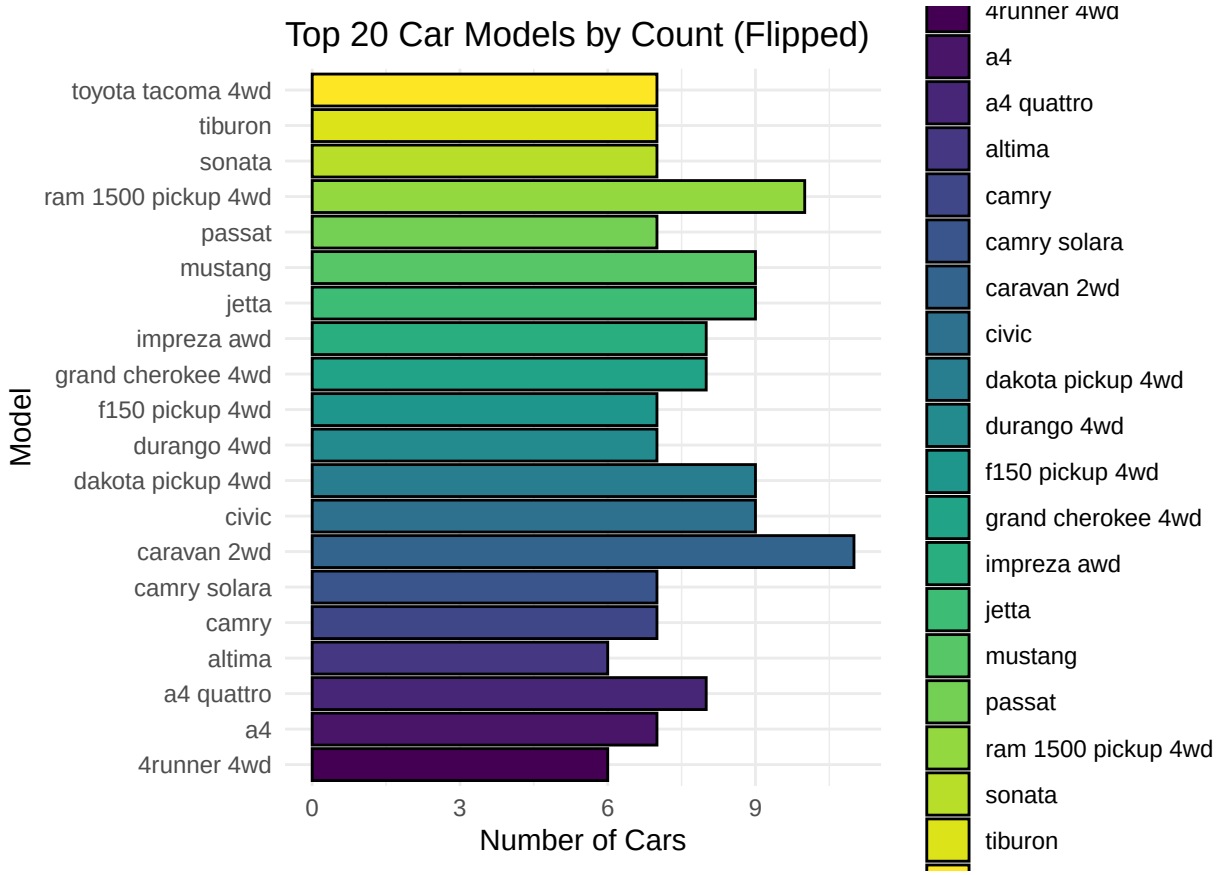
```
## # A tibble: 38 x 2
##    model              count
##    <chr>              <int>
##  1 4runner 4wd            6
##  2 a4                     7
##  3 a4 quattro             8
##  4 a6 quattro             3
##  5 altima                 6
##  6 c1500 suburban 2wd     5
##  7 camry               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

```
#4.a

top_20_models <- cars_per_model %>%
  arrange(desc(count)) %>%
  slice_head(n = 20)
```

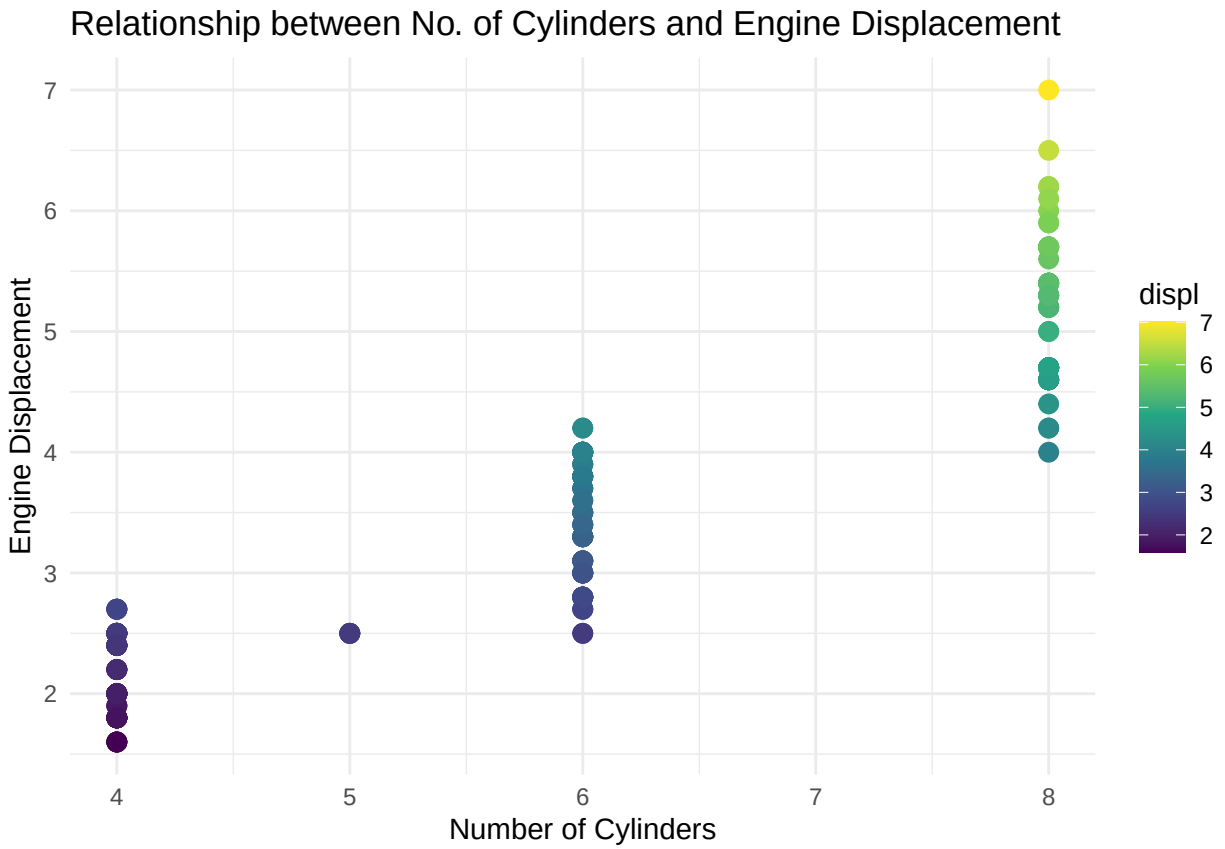
```
ggplot(top_20_models, aes(x = model, y = count, fill = model)) +
  geom_bar(stat = "identity", color = "black") +
  theme_minimal() +
  labs(title = "Top 20 Car Models by Count",
       x = "Model",
       y = "Number of Cars") +
theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 4),
      plot.margin = margin(1, 1, 1, 1, "cm"))+   scale_fill_viridis_d()
```



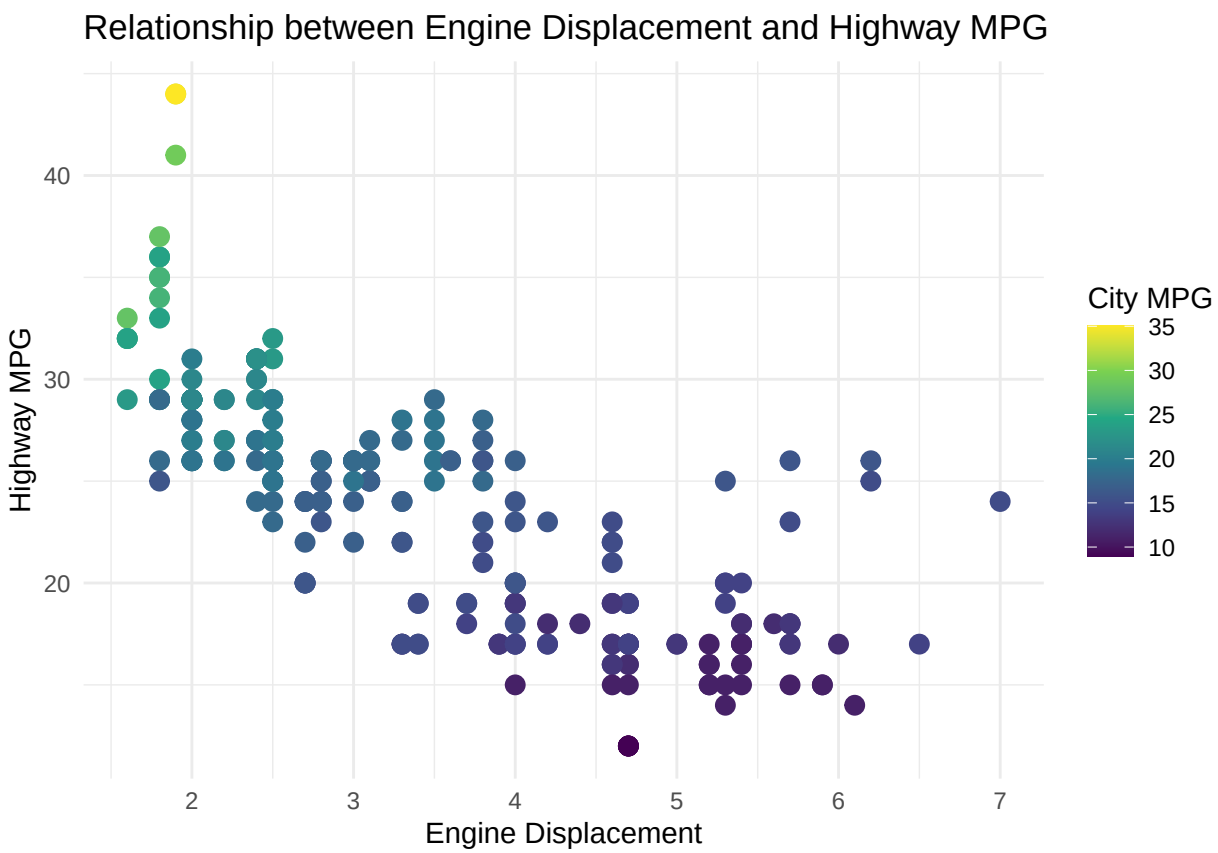

```
#4.b
ggplot(top_20_models, aes(x = model, y = count, fill = model)) +
  geom_bar(stat = "identity", color = "black") +
  coord_flip() +
  theme_minimal() +
  labs(title = "Top 20 Car Models by Count (Flipped)",
       x = "Model",
       y = "Number of Cars") +
  scale_fill_viridis_d()
```




```r
#5.
ggplot(mpg_data, aes(x = cyl, y = displ, color = displ)) +
  geom_point(size = 3) +
  theme_minimal() +
  labs(title = "Relationship between No. of Cylinders and Engine Displacement",
       x = "Number of Cylinders",
       y = "Engine Displacement") +
  scale_color_viridis_c()
```




```r
#6.
library(ggplot2)

ggplot(mpg_data, aes(x = displ, y = hwy, color = cty)) +
  geom_point(size = 3) +
  theme_minimal() +
  labs(title = "Relationship between Engine Displacement and Highway MPG",
       x = "Engine Displacement",
       y = "Highway MPG",
       color = "City MPG") +
  scale_color_viridis_c()  # Adds a continuous color scale
```



Negative Correlation Between displ and hwy: The plot demonstrates that as engine displacement increases, highway miles per gallon decrease. This relationship is expected, as larger engines (higher displ values) generally consume more fuel, which results in lower fuel efficiency (lower hwy values).

Larger Engines Are Less Efficient: Vehicles with high engine displacement typically have more cylinders or larger cylinder volumes, both of which require more fuel, resulting in lower MPG. Efficiency in City and Highway Are Related: Generally, cars designed for fuel efficiency show better performance in both city and highway settings. This correlation is why we see the color gradient for cty align with trends in hwy and displ.

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

#6. a
num_observations <- nrow(traffic_data)
num_variables <- ncol(traffic_data)
variable_names <- names(traffic_data)

cat("Number of observations:", num_observations, "\n")
```

```
## Number of observations: 48120
```

```
cat("Number of variables:", num_variables, "\n")
```

```
## Number of variables: 4
```

```
cat("Variables:", variable_names, "\n")
```

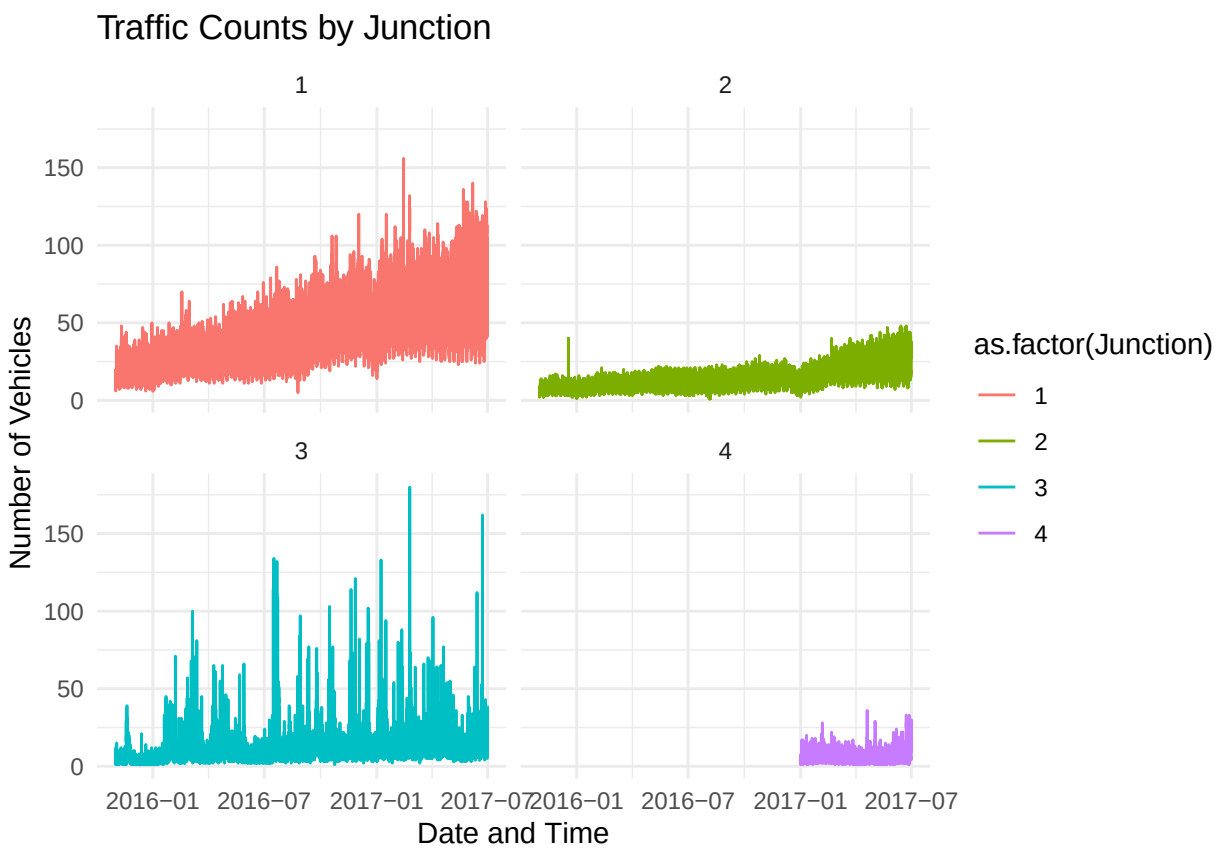
```
## Variables: DateTime Junction Vehicles ID
```

```
#6.b
junction_list <- split(traffic_data, traffic_data$Junction)
```

```r
for (junction in names(junction_list)) {
  cat("Data for junction:", junction, "\n")
  print(head(junction_list[[junction]]))
  cat("\n")
}
```

```
## Data for junction: 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
##
## Data for junction: 2
##                   DateTime Junction Vehicles          ID
## 14593 2015-11-01 00:00:00        2        6 20151101002
## 14594 2015-11-01 01:00:00        2        6 20151101012
## 14595 2015-11-01 02:00:00        2        5 20151101022
## 14596 2015-11-01 03:00:00        2        6 20151101032
## 14597 2015-11-01 04:00:00        2        7 20151101042
## 14598 2015-11-01 05:00:00        2        2 20151101052
##
## Data for junction: 3
##                   DateTime Junction Vehicles          ID
## 29185 2015-11-01 00:00:00        3        9 20151101003
## 29186 2015-11-01 01:00:00        3        7 20151101013
## 29187 2015-11-01 02:00:00        3        5 20151101023
## 29188 2015-11-01 03:00:00        3        1 20151101033
## 29189 2015-11-01 04:00:00        3        2 20151101043
## 29190 2015-11-01 05:00:00        3        2 20151101053
##
## Data for junction: 4
##                   DateTime Junction Vehicles          ID
## 43777 2017-01-01 00:00:00        4        3 20170101004
## 43778 2017-01-01 01:00:00        4        1 20170101014
## 43779 2017-01-01 02:00:00        4        4 20170101024
## 43780 2017-01-01 03:00:00        4        4 20170101034
## 43781 2017-01-01 04:00:00        4        2 20170101044
## 43782 2017-01-01 05:00:00        4        1 20170101054
```

```r
#6.c

library(ggplot2)

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

ggplot(traffic_data, aes(x = DateTime, y = Vehicles, color = as.factor(Junction))) +
  geom_line() +
  facet_wrap(~ Junction) +
  labs(title = "Traffic Counts by Junction", x = "Date and Time",
       y = "Number of Vehicles") +
  theme_minimal()
```




```
library(readxl)
alexa_data <- read_excel("alexa_file.xlsx")

#7.a

num_observations <- nrow(alexa_data)
num_columns <- ncol(alexa_data)
column_names <- names(alexa_data)

cat("Number of observations:", num_observations, "\n")
```

```
## Number of observations: 3150
```

```
cat("Number of columns:", num_columns, "\n")
```

```
## Number of columns: 5
```

```
cat("Column names:", column_names, "\n")
```

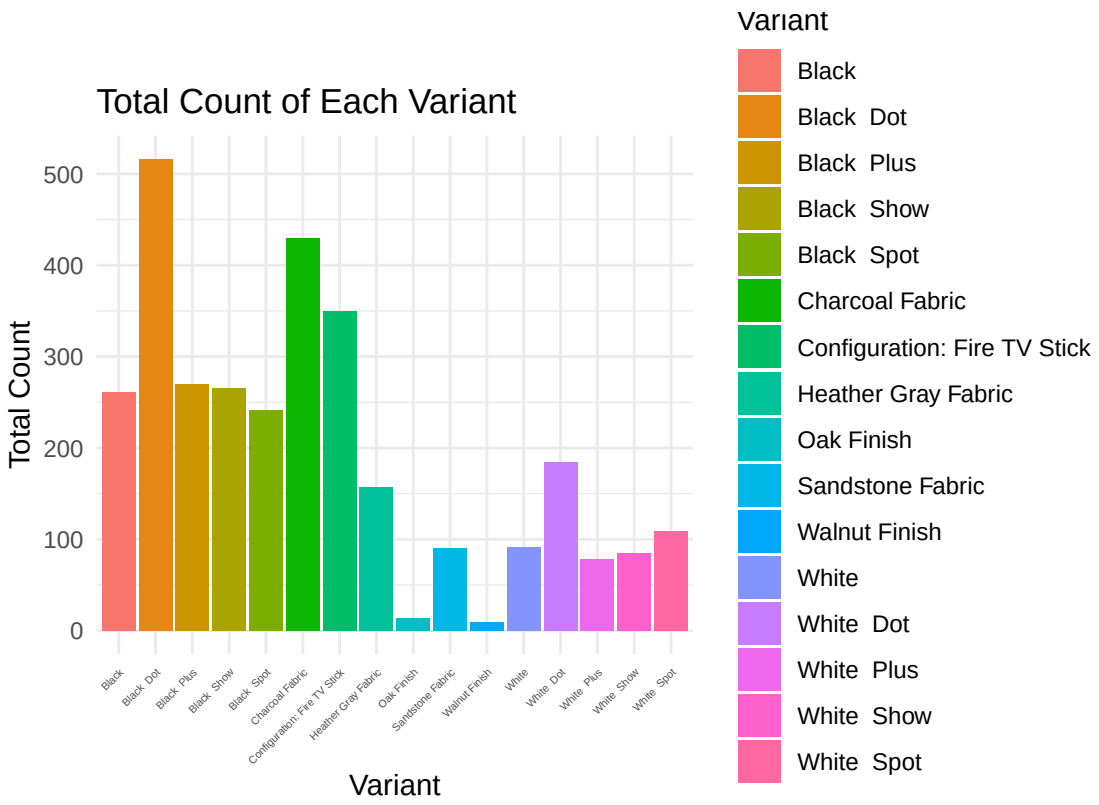
```
## Column names: rating date Variant verified_reviews feedback
```

```
#7.b

library(dplyr)

variation_totals <- alexa_data %>%
  group_by(Variant) %>%
  summarise(total = n())
```

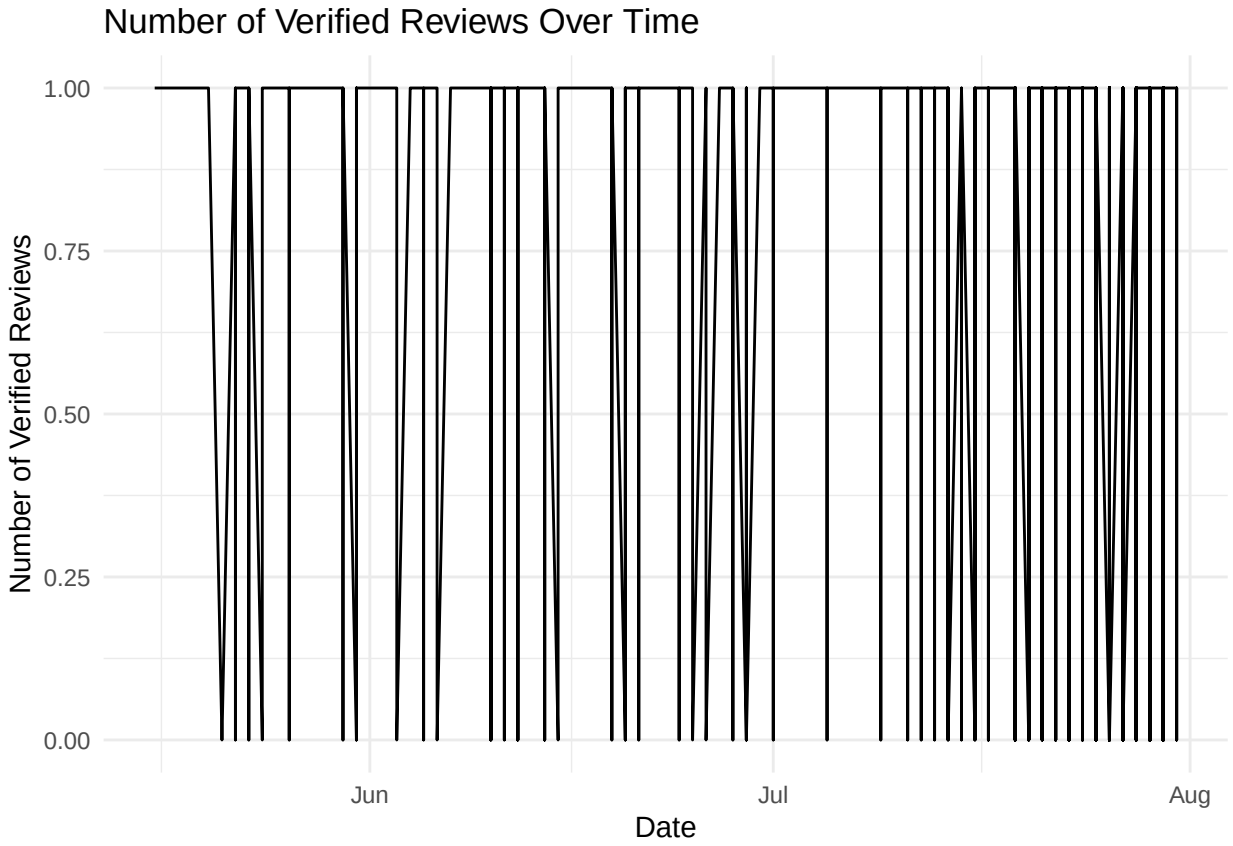
```r
print(variation_totals)
```

```
## # A tibble: 16 x 2
##    Variant                       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
#7.c

library(ggplot2)

ggplot(variation_totals, aes(x = Variant, y = total, fill = Variant)) +
  geom_bar(stat = "identity") +
  labs(title = "Total Count of Each Variant", x = "Variant",
       y = "Total Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 4),
        plot.margin = margin(1, 1, 1, 1, "cm"))
```




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

ggplot(alexa_data, aes(x = date, y = feedback)) +
  geom_line() +
  labs(title = "Number of Verified Reviews Over Time", x = "Date",
       y = "Number of Verified Reviews") +
  theme_minimal()
```




```
#7.e

rating_by_variant <- alexa_data %>%
  group_by(Variant) %>%
  summarise(avg_rating = mean(rating, na.rm = TRUE))

print(rating_by_variant)
```

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

```r
ggplot(rating_by_variant, aes(x = Variant, y = avg_rating, fill = Variant)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Rating by Variant", x = "Variant", y = "Average Rating") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 4),
        plot.margin = margin(1, 1, 1, 1, "cm"))
```