Rworksheet.Sabando#4c.Rmd

Samantha L. Sabando

2025-12-07

```
##########################################
# 1. Import mpg dataset and identify variable types
##########################################
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)
library(tidyr)
library(readxl)

# Import dataset
data(mpg)
write.csv(mpg, "mpg.csv", row.names = FALSE)
mpgdata <- read.csv("mpg.csv", header = TRUE, stringsAsFactors = FALSE)

# a. Import CSV shown above
# b. Categorical variables: manufacturer, model, trans, drv, fl, class, year
# c. Continuous variables: displ, cty, hwy

##########################################
# 2. Which manufacturer has the most models? Which model has most variations?
##########################################
# a. Group manufacturers and find unique models
manumodels <- mpgdata %>%
  group_by(manufacturer) %>%
  summarise(totalmodels = n_distinct(model)) %>%
  arrange(desc(totalmodels))
manumodels
```

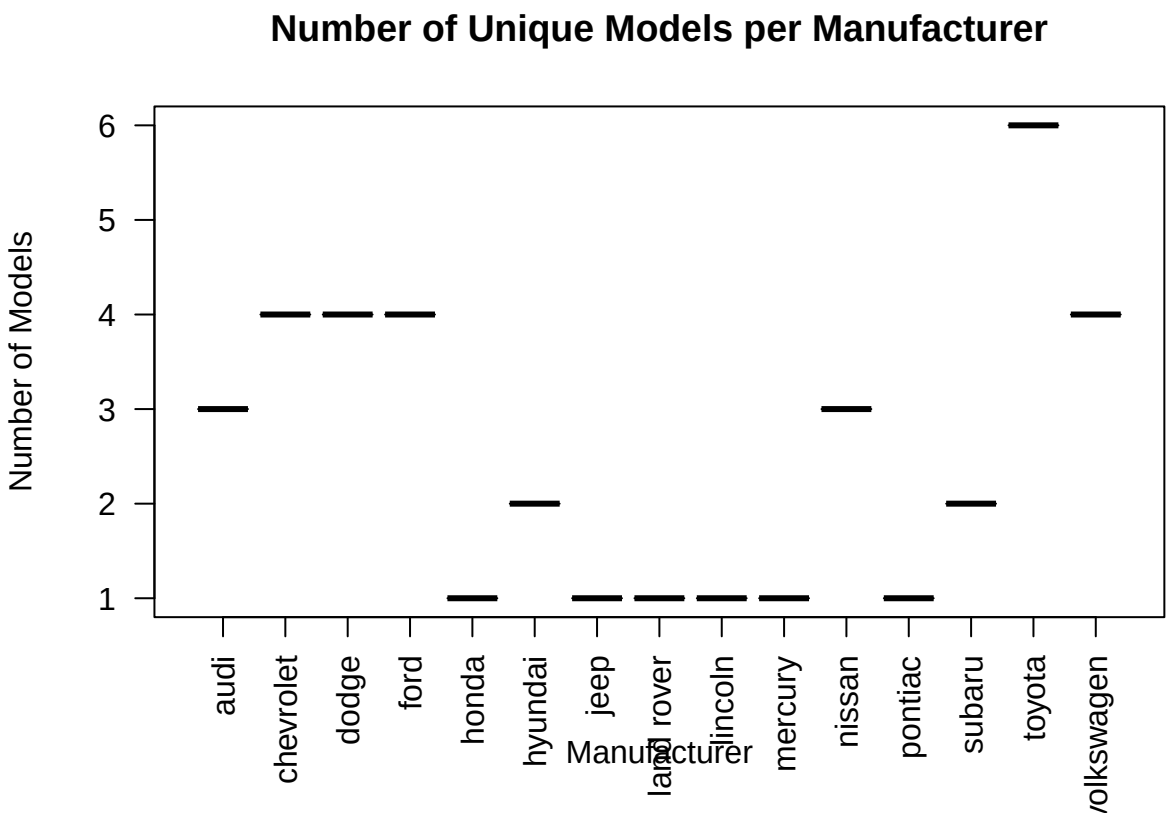
```
## # A tibble: 15 x 2
##    manufacturer totalmodels
##    <chr>              <int>
##  1 toyota                 6
##  2 chevrolet              4
##  3 dodge                  4
```

```
##  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
```

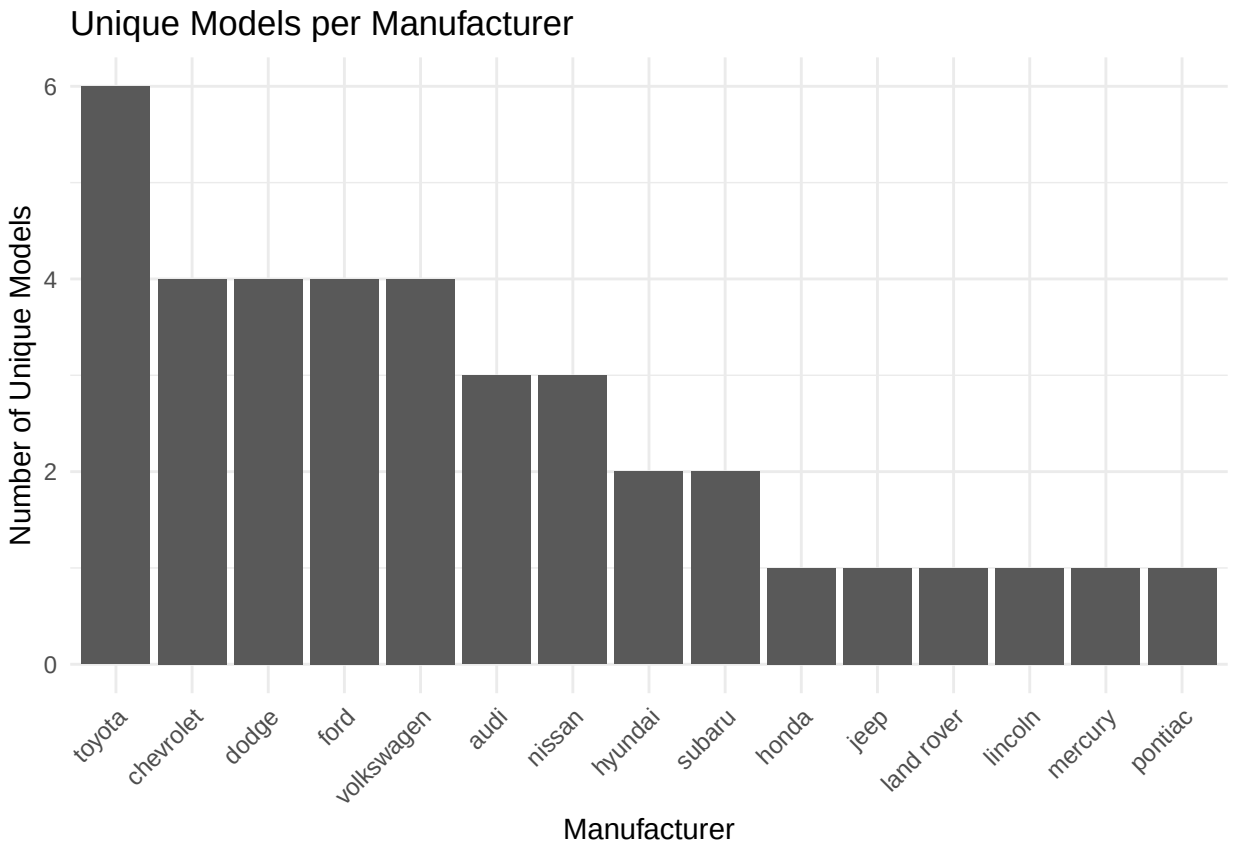
```
modelvars <- mpgdata %>%
  group_by(model) %>%
  summarise(totalvars = n()) %>%
  arrange(desc(totalvars))
modelvars
```

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

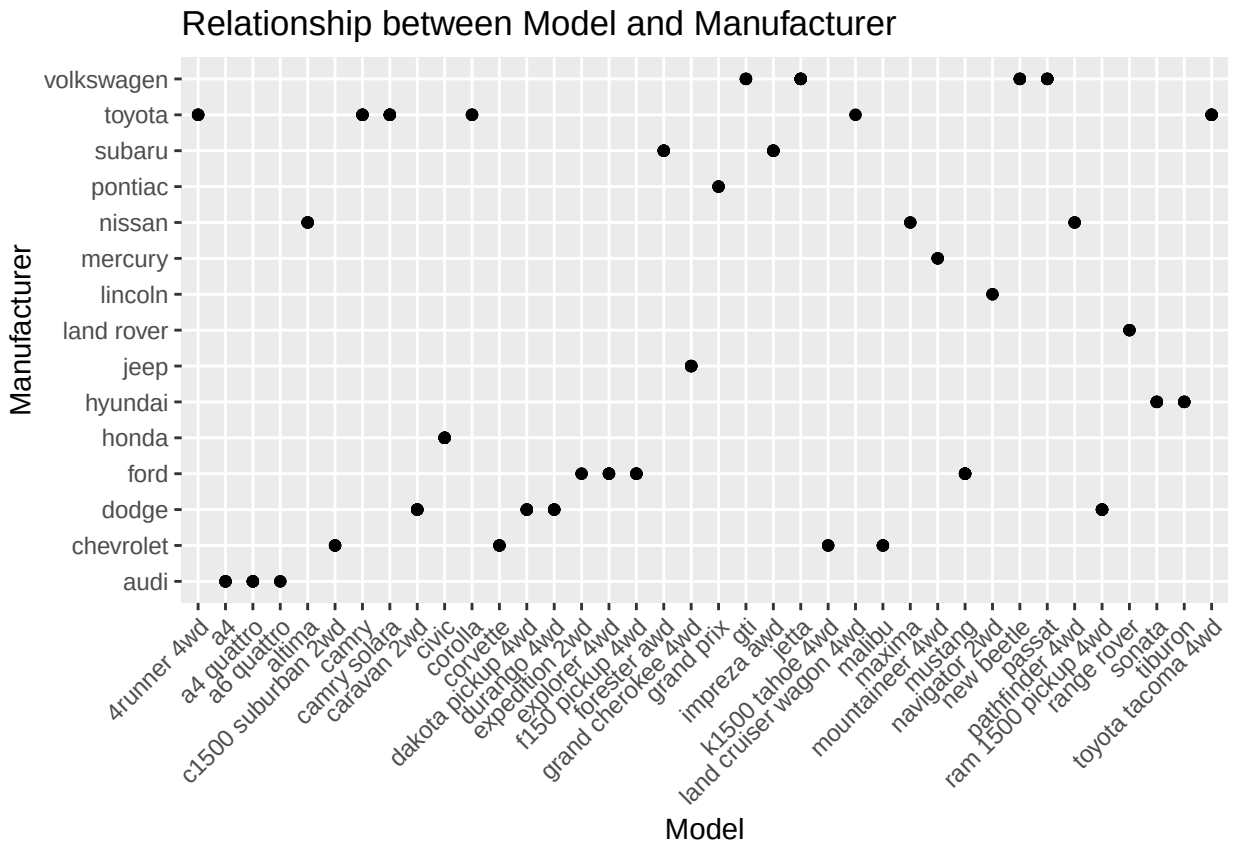
```
# b. Graph the result
plot(as.factor(manumodels$manufacturer), manumodels$totalmodels, las = 2,
     main = "Number of Unique Models per Manufacturer",
     xlab = "Manufacturer",
     ylab = "Number of Models")
```



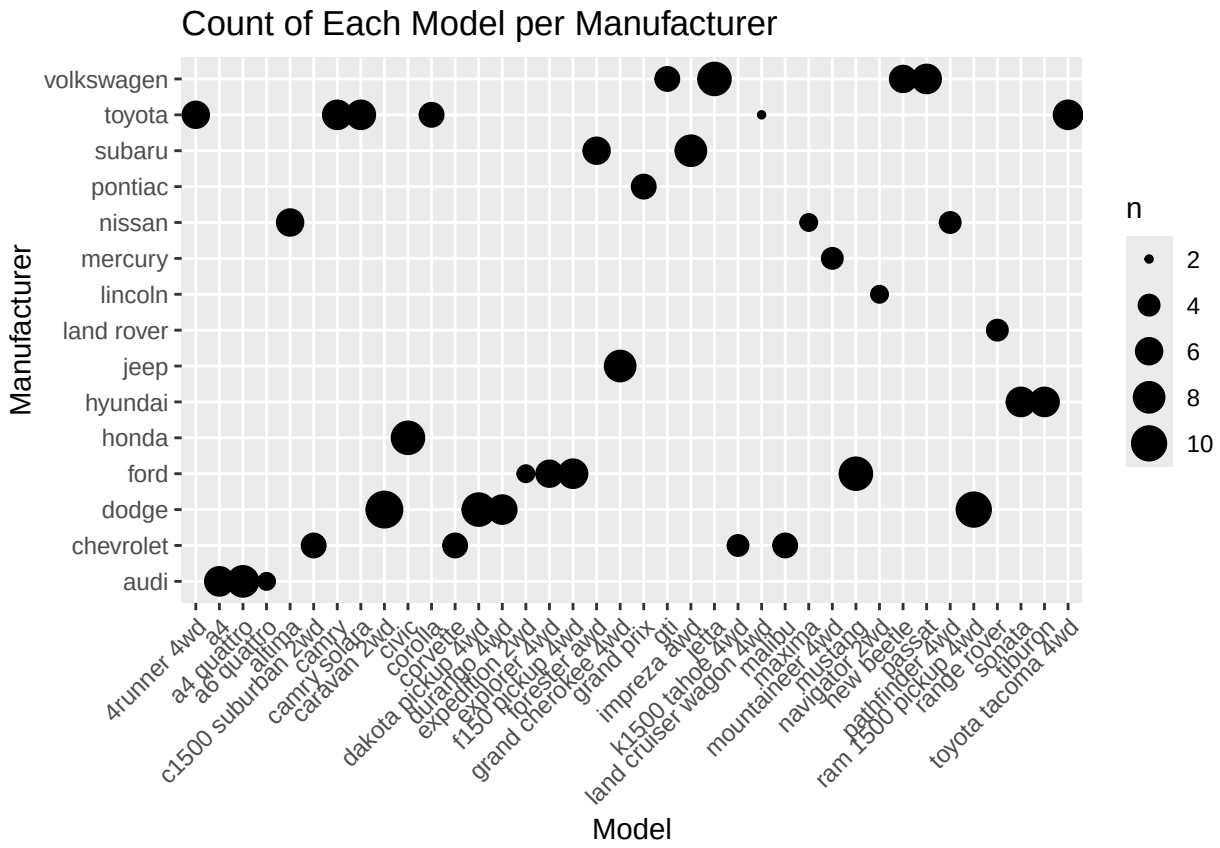

```
ggplot(manumodels, aes(x = reorder(manufacturer, -totalmodels), y = totalmodels)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Unique Models per Manufacturer",
       x = "Manufacturer",
       y = "Number of Unique Models") +
  theme(axis.text.x = element_text(angle = 45,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



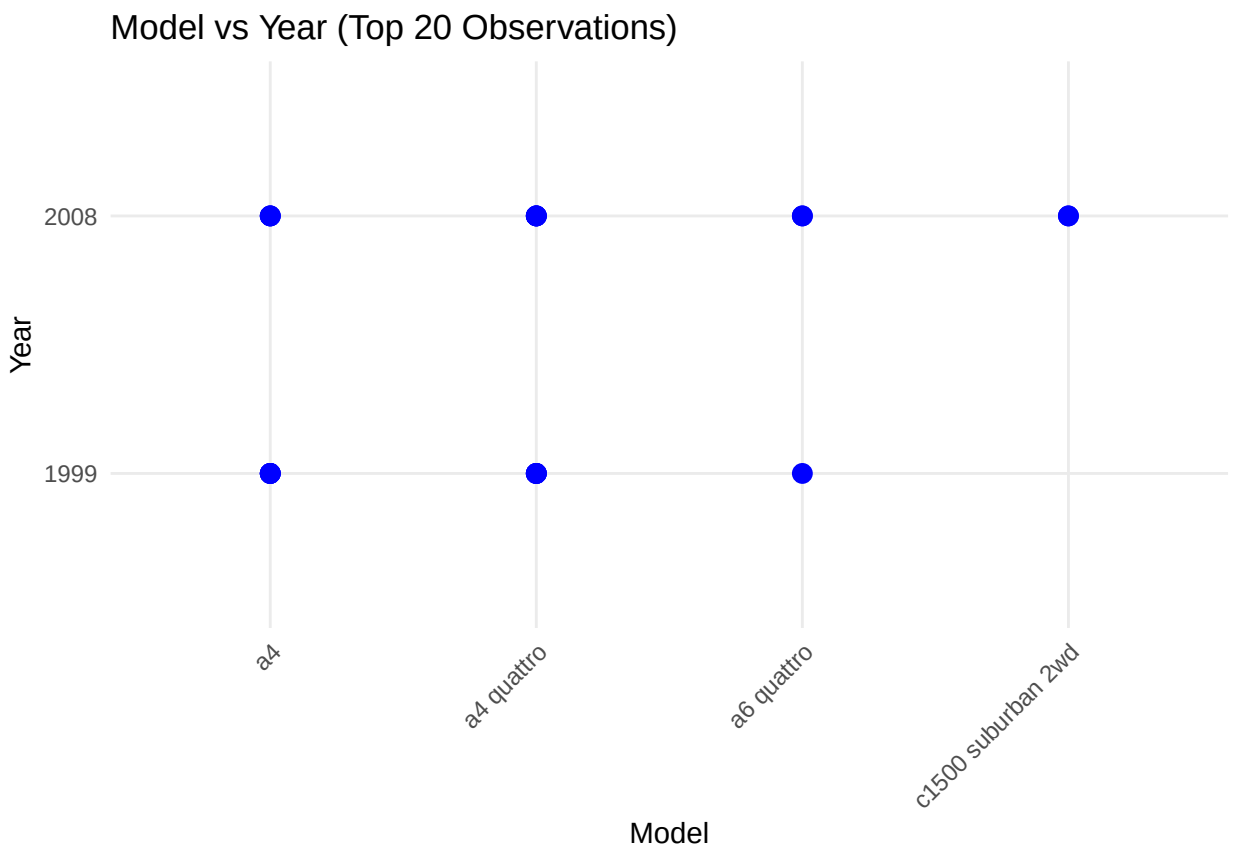

```
##############################################
# 3. Relationship of model and manufacturer
##############################################
ggplot(mpgdata, aes(x = model, y = manufacturer)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Relationship between Model and Manufacturer",
       x = "Model",
       y = "Manufacturer")
```




```
# Improved plot with counts
ggplot(mpgdata, aes(x = model, y = manufacturer)) +
  geom_count() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Count of Each Model per Manufacturer",
       x = "Model",
       y = "Manufacturer")
```

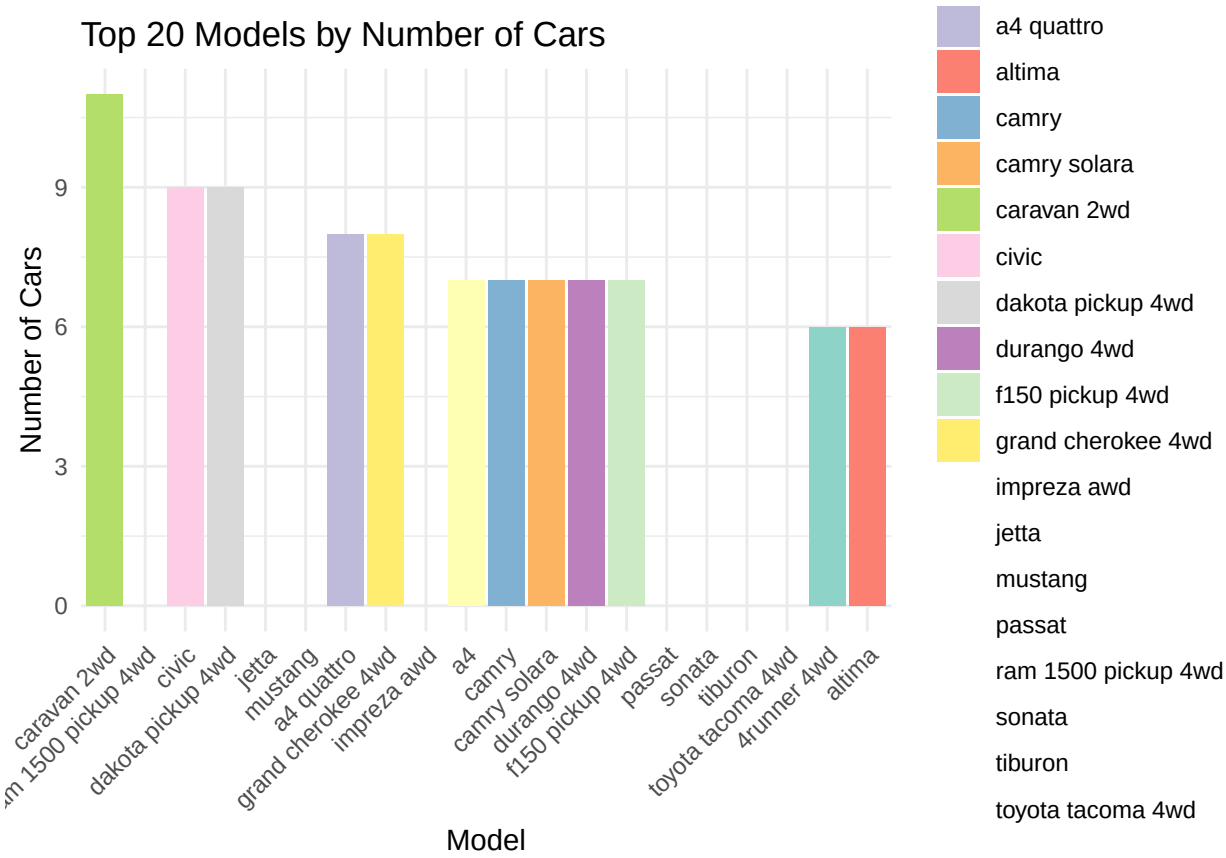



```
##############################################
# 4. Plot model vs year (top 20 observations)
##############################################
top20data <- mpgdata[1:20, ]
ggplot(top20data, aes(x = model, y = factor(year))) +
  geom_point(color = "blue", size = 3) +
  labs(title = "Model vs Year (Top 20 Observations)",
       x = "Model",
       y = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



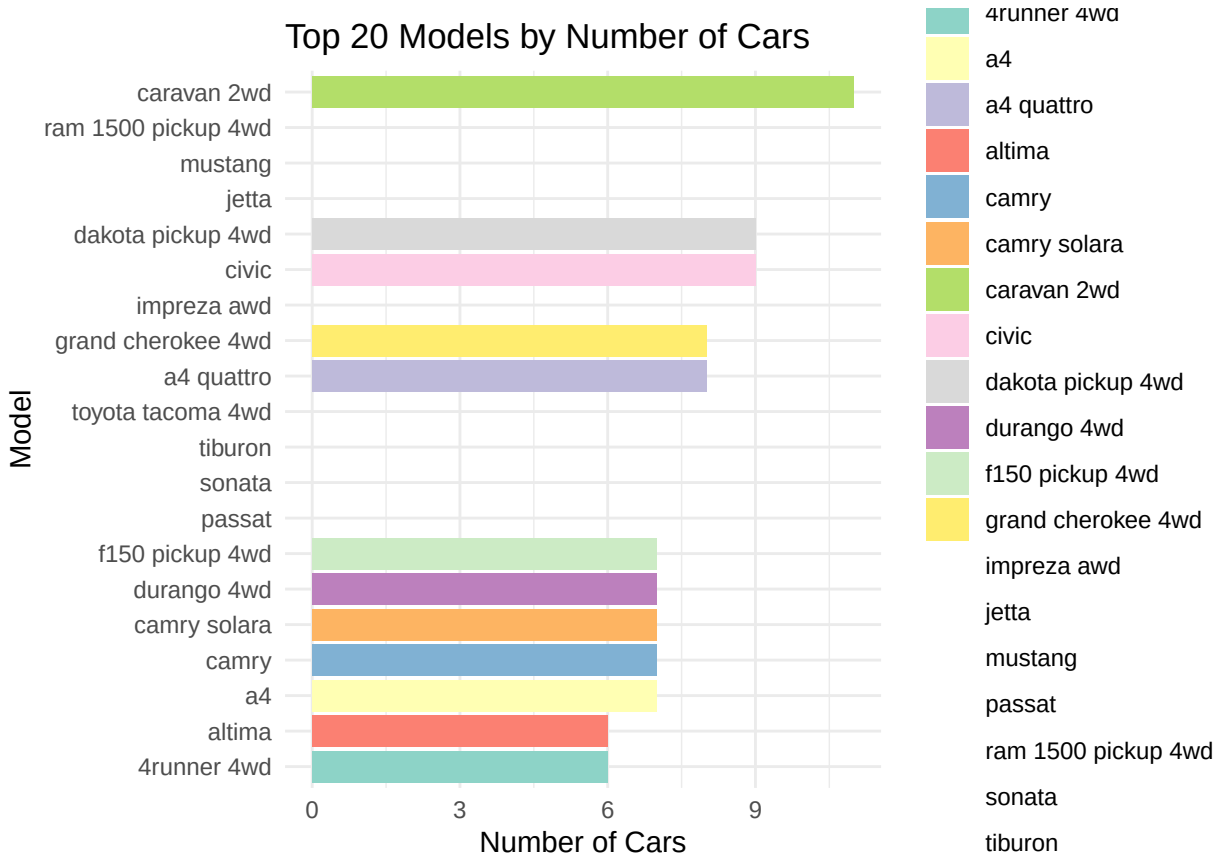

```
##############################################
# 5. Number of cars per model and plots
##############################################
modelcount <- mpgdata %>%
  group_by(model) %>%
  summarise(carnum = n()) %>%
  arrange(desc(carnum))
top20models <- modelcount[1:20, ]

# a. Bar plot
ggplot(top20models, aes(x = reorder(model, -carnum), y = carnum, fill = model)) +
  geom_bar(stat = "identity") +
  labs(title = "Top 20 Models by Number of Cars",
       x = "Model",
       y = "Number of Cars") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set3")
```

```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Set3 is 12
## Returning the palette you asked for with that many colors
```




```
# b. Flipped bar plot
ggplot(top20models, aes(x = reorder(model, carnum), y = carnum, fill = model)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 20 Models by Number of Cars",
       x = "Model",
       y = "Number of Cars") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set3")
```

```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Set3 is 12
## Returning the palette you asked for with that many colors
```




```
##############################################
# 6. Relationship between cylinders and engine displacement
##############################################
##############################################
# 6. TRAFFIC DATASET
##############################################
# Import traffic data
traffic <- read.csv("traffic.csv", header = TRUE, stringsAsFactors = FALSE)

# a. Number of observations and variables
cat("Traffic dataset dimensions (rows, cols):", dim(traffic), "\n")
```

```
## Traffic dataset dimensions (rows, cols): 6 3
```

```
cat("Traffic dataset variables:", names(traffic), "\n\n")
```

```
## Traffic dataset variables: Time Junction1 Junction2
```

```
# b. Subset the dataset into junctions (long format)
traffic_long <- traffic %>%
  pivot_longer(cols = starts_with("Junction"),
               names_to = "Junction",
               values_to = "Vehicles")

# Example: show first few rows of Junction1
junction1 <- traffic_long %>% filter(Junction == "Junction1")
cat("Subset of Junction1:\n")
```

```
## Subset of Junction1:
```

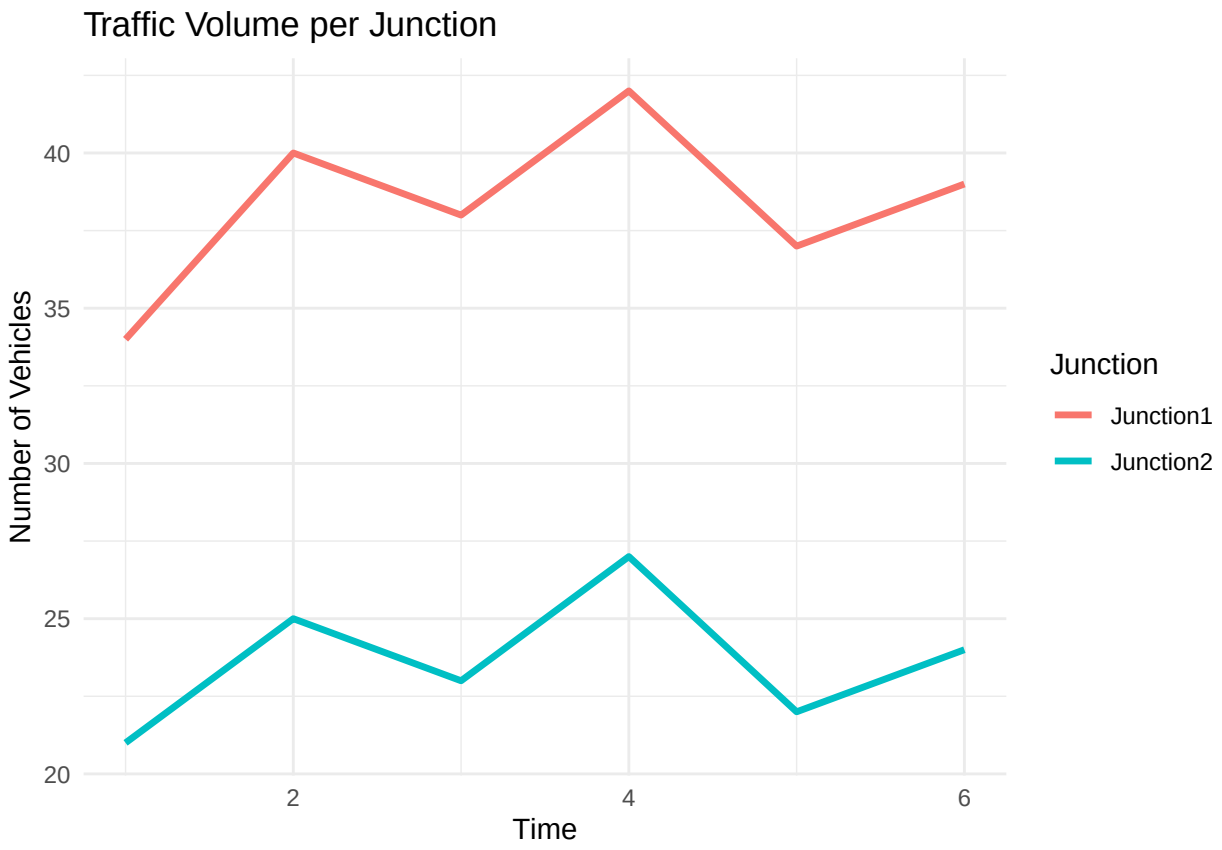
```
print(head(junction1))
```

```
## # A tibble: 6 x 3
##    Time Junction  Vehicles
##   <int> <chr>        <int>
## 1     1 Junction1       34
## 2     2 Junction1       40
## 3     3 Junction1       38
## 4     4 Junction1       42
## 5     5 Junction1       37
## 6     6 Junction1       39
```

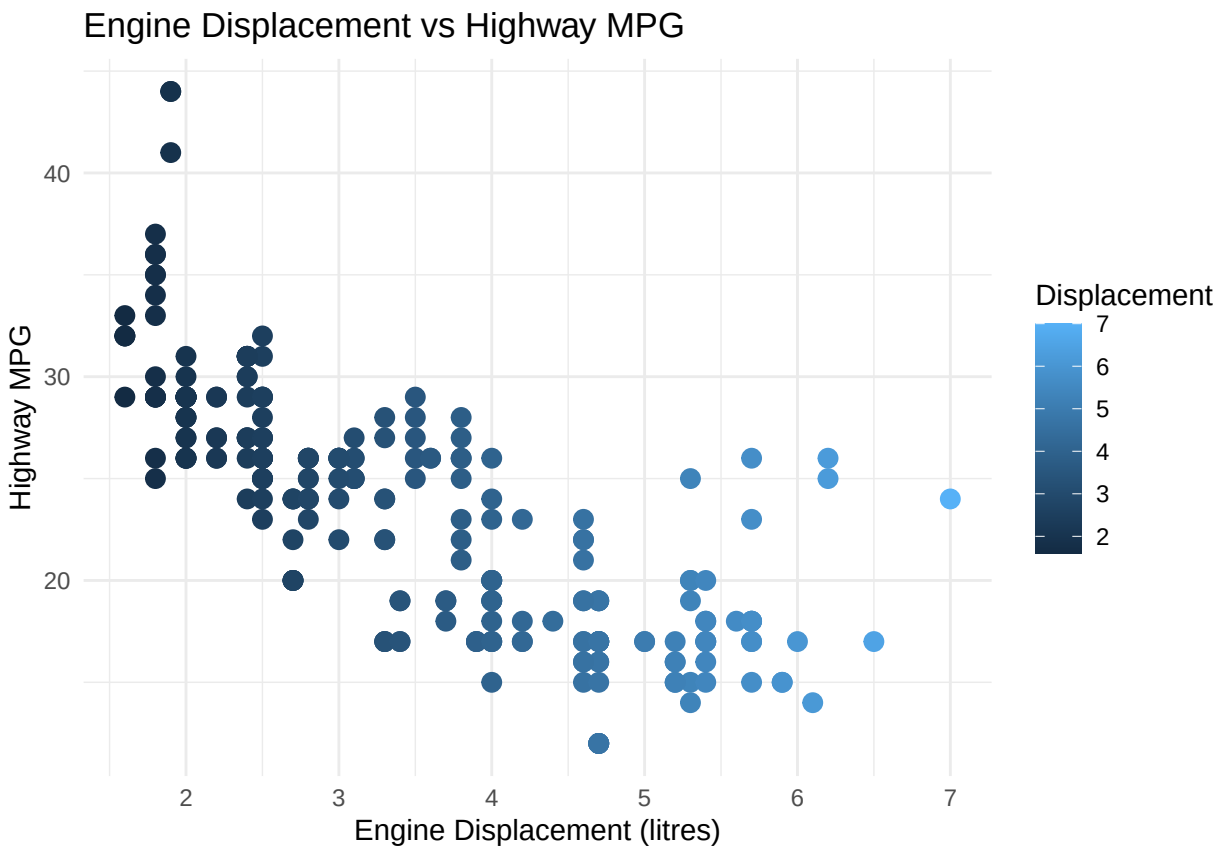
```
cat("\n")
```

```
# c. Plot each junction using geom_line()
ggplot(traffic_long, aes(x = Time, y = Vehicles, color = Junction)) +
  geom_line(size = 1.2) +
  labs(title = "Traffic Volume per Junction",
       x = "Time",
       y = "Number of Vehicles") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```r
# Relationship between displ and hwy
ggplot(mpgdata, aes(x = displ, y = hwy, color = displ)) +
  geom_point(size = 3) +
  labs(title = "Engine Displacement vs Highway MPG",
       x = "Engine Displacement (litres)",
       y = "Highway MPG",
       color = "Displacement") +
  theme_minimal()
```




```r
# Shows negative correlation: larger engines → lower highway MPG

##########################################
# 7. Traffic dataset
##########################################
traffic <- read.csv("traffic.csv", header = TRUE, stringsAsFactors = FALSE)
cat("Traffic dataset dimensions (rows, cols):", dim(traffic), "\n")
```

```
## Traffic dataset dimensions (rows, cols): 6 3
```

```r
cat("Traffic dataset variables:", names(traffic), "\n")
```

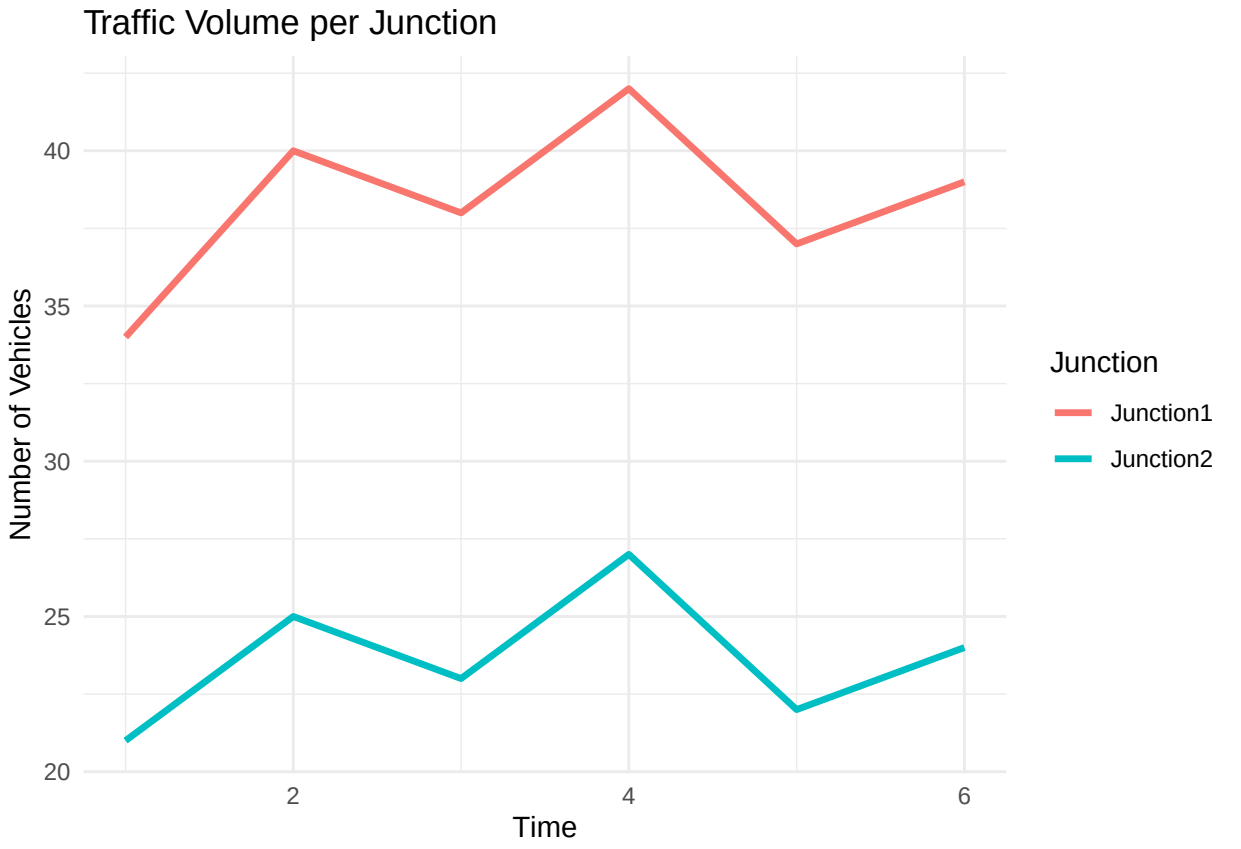
```
## Traffic dataset variables: Time Junction1 Junction2
```

```r
# Reshape wide → long for plotting
traffic_long <- traffic %>%
  pivot_longer(cols = starts_with("Junction"),
               names_to = "Junction",
               values_to = "Vehicles")
```

```r
head(traffic_long)
```

```
## # A tibble: 6 x 3
##    Time Junction  Vehicles
##   <int> <chr>        <int>
## 1     1 Junction1       34
## 2     1 Junction2       21
## 3     2 Junction1       40
## 4     2 Junction2       25
## 5     3 Junction1       38
## 6     3 Junction2       23
```

```r
# Plot each junction using geom_line
ggplot(traffic_long, aes(x = Time, y = Vehicles, color = Junction)) +
  geom_line(size = 1.2) +
  labs(title = "Traffic Volume per Junction",
       x = "Time",
       y = "Number of Vehicles") +
  theme_minimal()
```




```r
##############################################
# 7. Alexa dataset
##############################################
alexa <- read_excel("alexa_file.xlsx")
cat("Alexa dataset dimensions (rows, cols):", dim(alexa), "\n")
```

```
## Alexa dataset dimensions (rows, cols): 3150 5
```

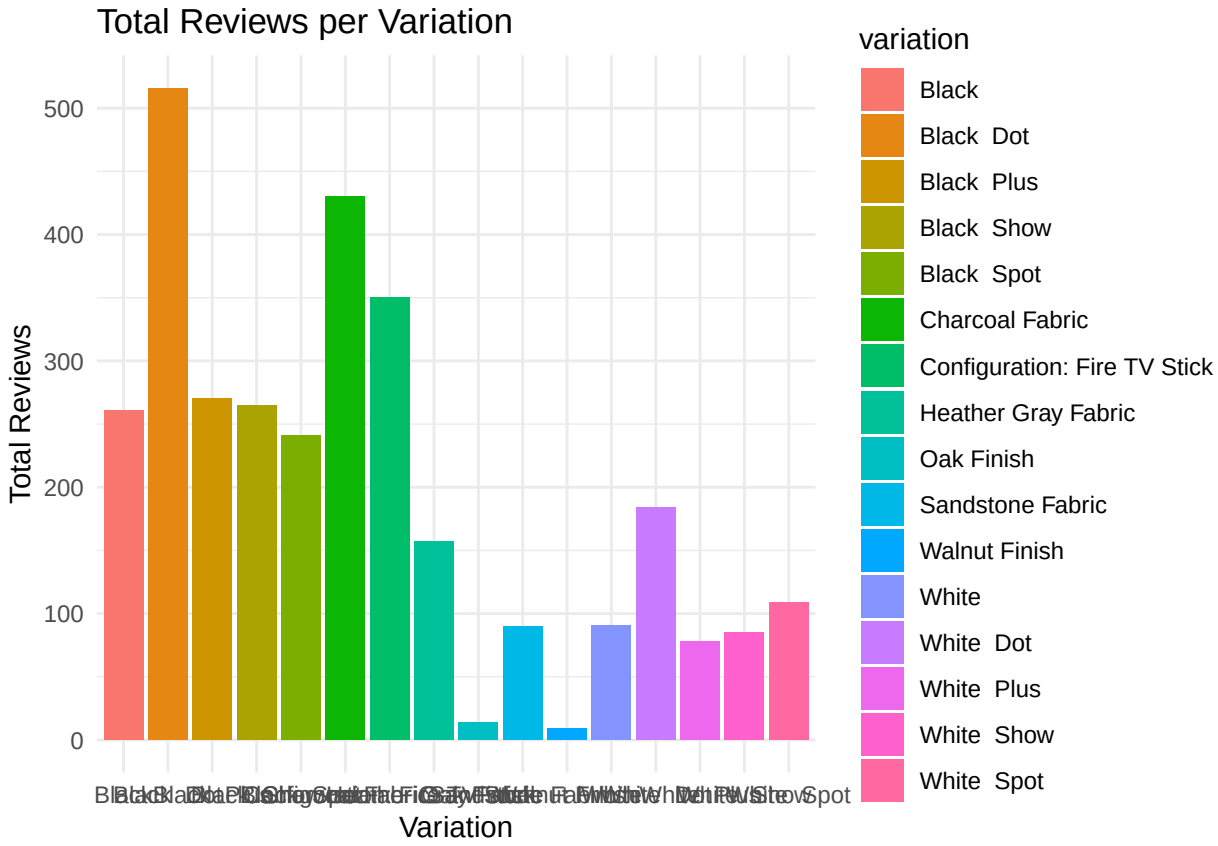
```r
# Group variations and total per variation
variation_summary <- alexa %>%
  group_by(variation) %>%
  summarise(total_reviews = n())
cat("Total reviews per variation:\n")
```

```
## Total reviews per variation:
```

```r
print(variation_summary)
```

```
## # A tibble: 16 x 2
##    variation                     total_reviews
##    <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
# Plot variations
ggplot(variation_summary, aes(x = variation, y = total_reviews, fill = variation)) +
  geom_bar(stat = "identity") +
  labs(title = "Total Reviews per Variation",
       x = "Variation",
       y = "Total Reviews") +
  theme_minimal()
```




```
# Verified reviews over time
ggplot(alexa, aes(x = date, y = verified_reviews)) +
  geom_line(color = "blue") +
  labs(title = "Verified Reviews Over Time",
       x = "Date",
       y = "Number of Verified Reviews") +
  theme_minimal()
```

are some serious flaws, particularly if you are the last one to bed or the first to wake. It doesn't seem like the engineer

expensive alternative option to fill the gap. Ordered the Amazon Fire Stick from Best Buy. Instructions were short and

one of the lights by saying "Alexa, turn off the second light". In the Alexa app, I created a 'Group' with &#34
, but lately I've been getting terrible support. The guy that took my call just rambled off a (completely unhelpful) script a

igh multiple pages. The one thing that I am not a fan of is the home screen cards do not really rotate that much - they a

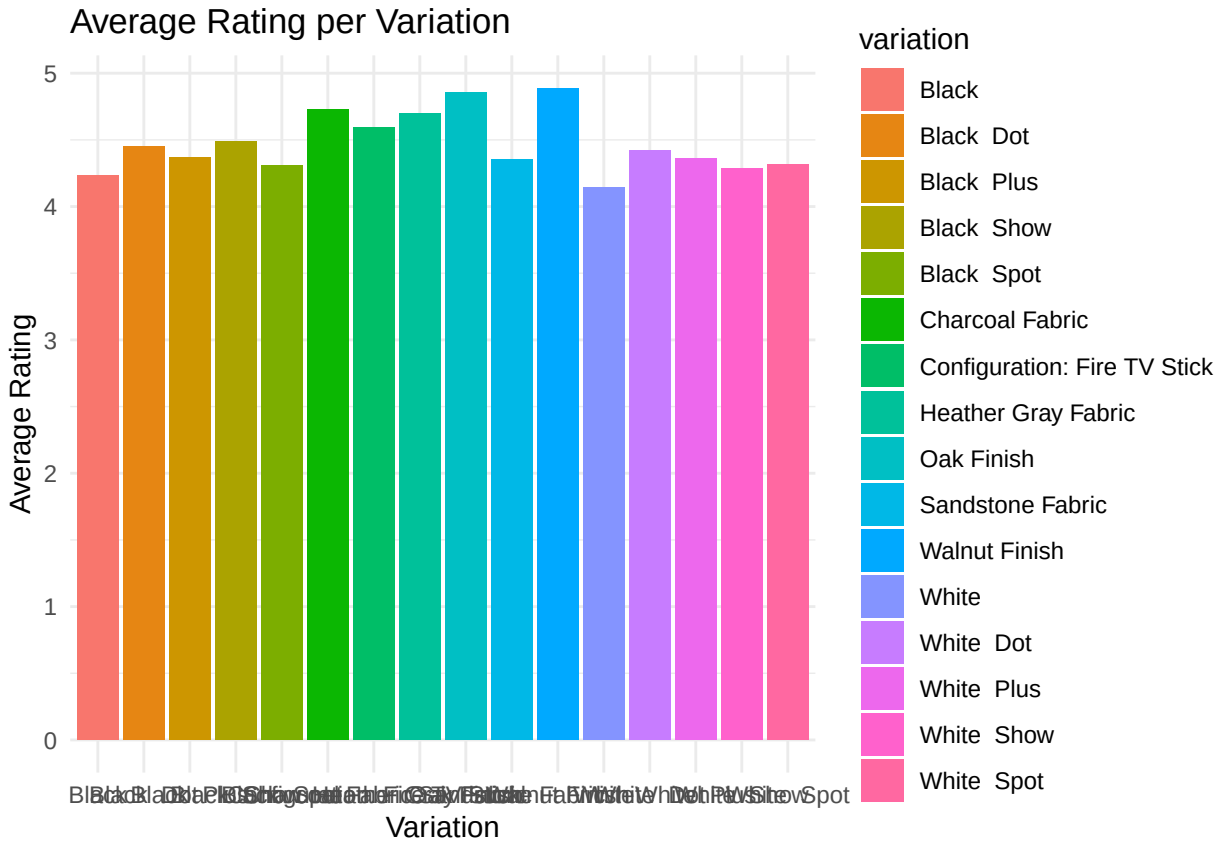
```
# Relationship between variations and ratings
variation_rating <- alexa %>%
  group_by(variation) %>%
  summarise(avg_rating = mean(rating, na.rm = TRUE))
cat("Average rating per variation:\n")
```

```
## Average rating per variation:
```

```
print(variation_rating)
```

```
## # A tibble: 16 x 2
##    variation                    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
```

```
# Plot average rating per variation
ggplot(variation_rating, aes(x = variation, y = avg_rating, fill = variation)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Rating per Variation",
       x = "Variation",
       y = "Average Rating") +
  theme_minimal()
```


Average Rating per Variation
Average Rating
0
1
2
3
4
5
Variation
variation
Black
Black Dot
Black Plus
Black Show
Black Spot
Charcoal Fabric
Configuration: Fire TV Stick
Heather Gray Fabric
Oak Finish
Sandstone Fabric
Walnut Finish
White
White Dot
White Plus
White Show
White Spot