

RWorksheet_lauron#4c.Rmd.

Mary Ghale C. Lauron

2025-12-02

```
#(1.)

#a import a csv file

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)
data(mpg)

write.csv(mpg, "mpg.csv", row.names = FALSE)

mpgdata <- read.csv("mpg.csv", header = TRUE, stringsAsFactors = FALSE)
str(mpgdata)

## '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 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" ...

#b. variables from mpg dataset are categorical
#The variables from mpg dataset that are categorical manufacturer, model, trans, #drv, fl, class.

#c. continuous variables
#These are the continuous variables are year,displ, cty, hwy, cyl.
```

#(2.) manufacturer that has the most models.Model that has the most variations.

```
manu_models_vars <- mpgdata %>%  
  group_by(model) %>%  
  summarise(total_manumovars = n()) %>%  
  arrange(desc(total_manumovars))
```

manu_models_vars

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

#a. Group the manufacturers and find the unique models.

```
manu_model <- mpgdata %>%  
  group_by(manufacturer) %>%  
  summarise(uniqumodels = n_distinct(model)) %>%  
  arrange(desc(uniqumodels))
```

manu_model

```
## # A tibble: 15 x 2  
##   manufacturer uniqumodels  
##   <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
```

#OUTPUT:

```
# A tibble: 15 × 2  
#manufacturer total_models  
#<chr>          <int>
```

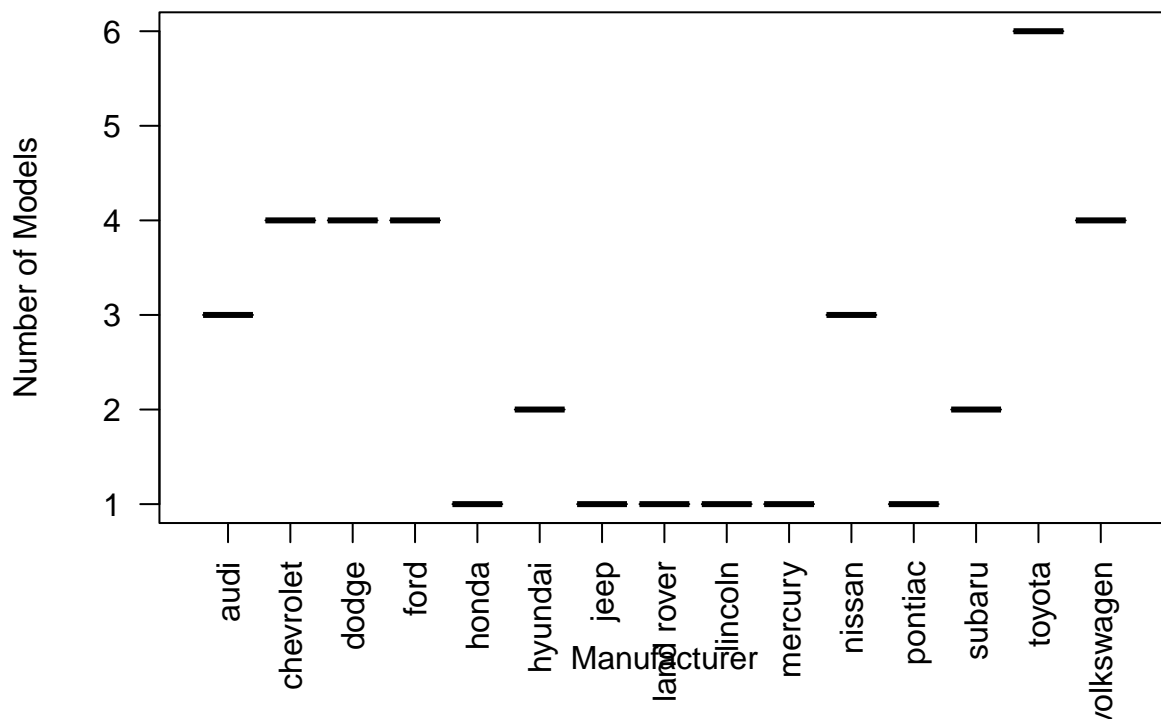
```
# 1 toyota          6
#2 chevrolet       4
#3 dodge           4
#4 ford            4
#5 volkswagen      4
#6 audi            3
#7 nissan           3
#8 hyundai         2
#9 subaru          2
#10 honda          1
#11 jeep           1
#12 land rover     1
#13 lincoln        1
#14 mercury        1
#15 pontiac        1
```

```
#b. plot() and ggplot().
```

```
#b1. plot()
```

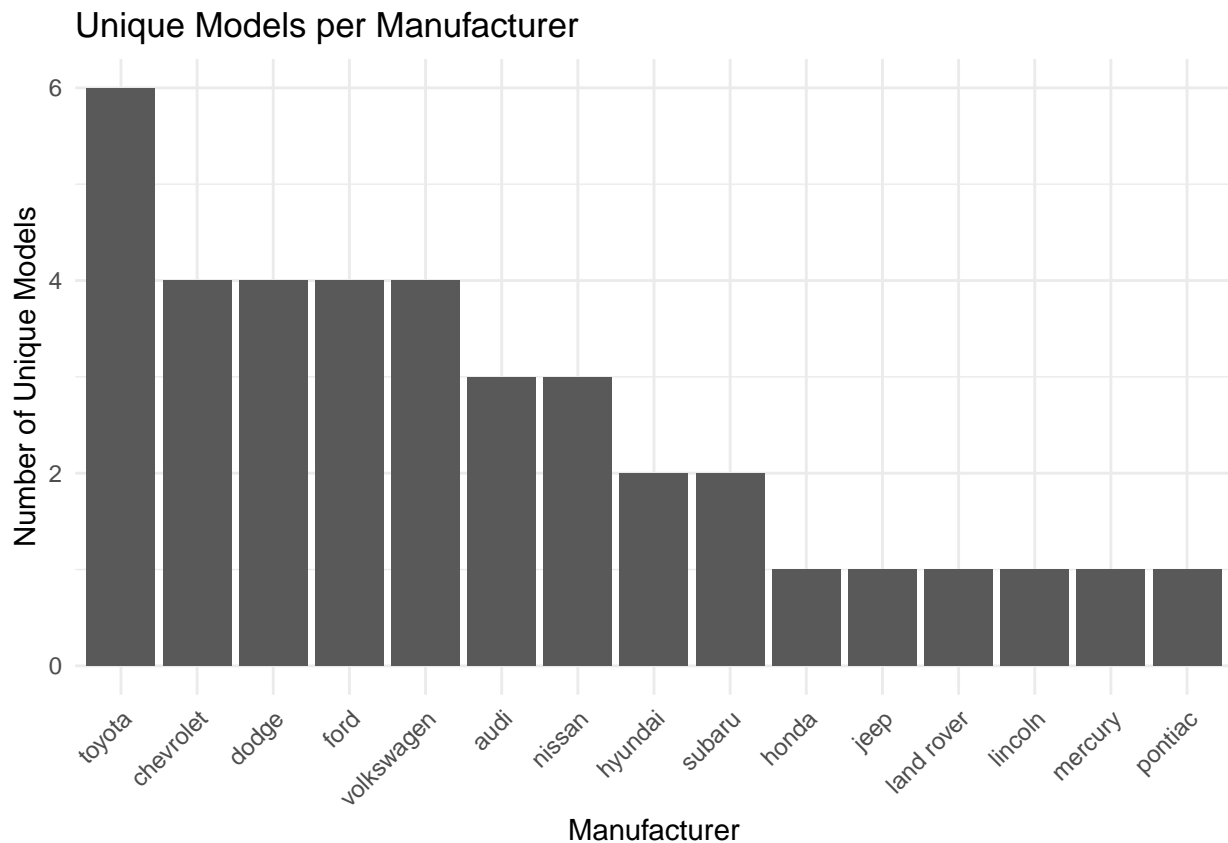
```
plot(as.factor(manu_model$manufacturer),
     manu_model$Uniquemodels,
     las = 2,
     main = "Number of Unique Models per Manufacturer",
     xlab = "Manufacturer",
     ylab = "Number of Models")
```

Number of Unique Models per Manufacturer



```
#b2. ggplot()
```

```
ggplot(manu_model, aes(x = reorder(manufacturer, -uniquemodels), y = uniquemodels)) +  
  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))
```



#(2.) Relationship of the model and the manufacturer.

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

#The ggplot shows the number of models per manufacturer: on the y axis it shows that number of models and on the x axis it shows the manufacturer listing a models in vertical way. The whole graph shows the relationship of models to manufacturer.

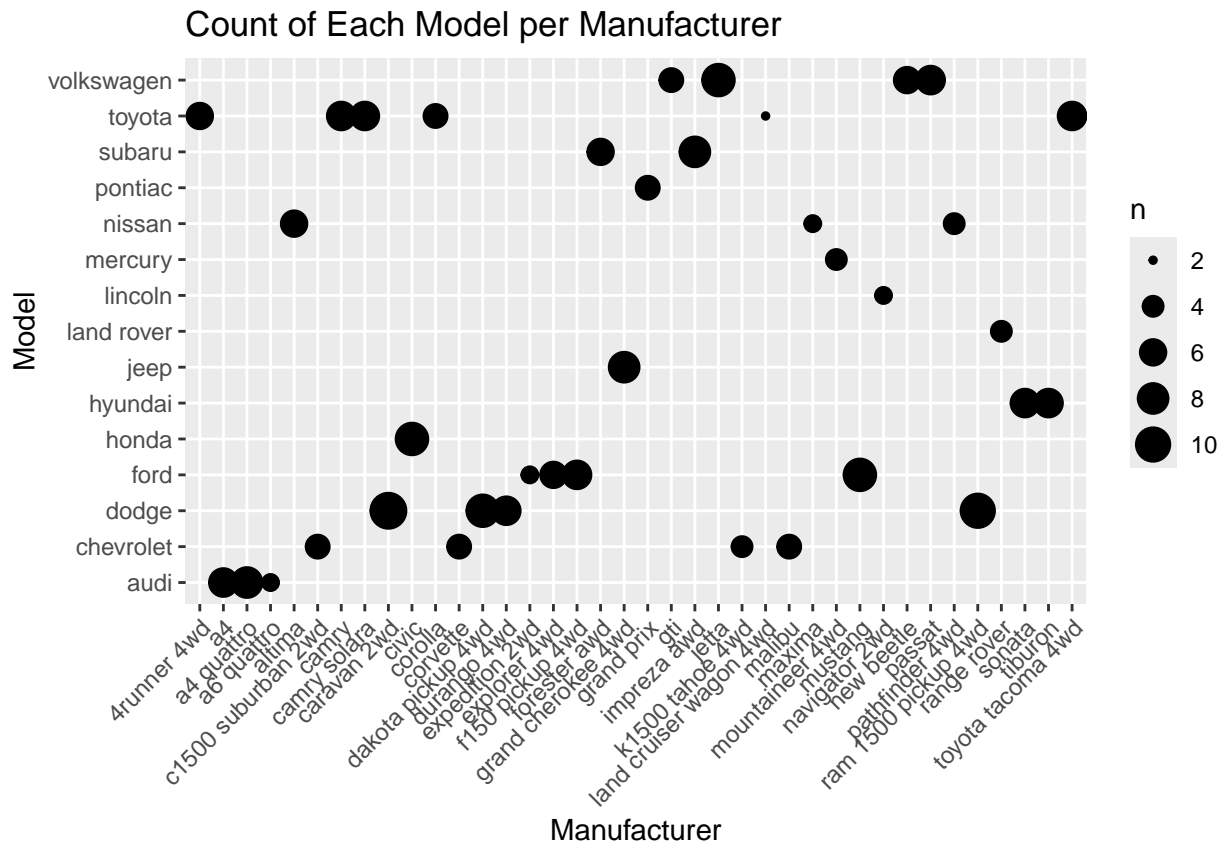
#b. For you, is it useful? If not, how could you modify the data to make it more informative?

#It is already useful but not really an informative as it lacks some key points.

#alternatives:

```
ggplot(mpgdata, aes(model, manufacturer)) +  
  geom_count() +
```

```
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
labs(title = "Count of Each Model per Manufacturer",
     x = "Manufacturer",
     y = "Model")
```

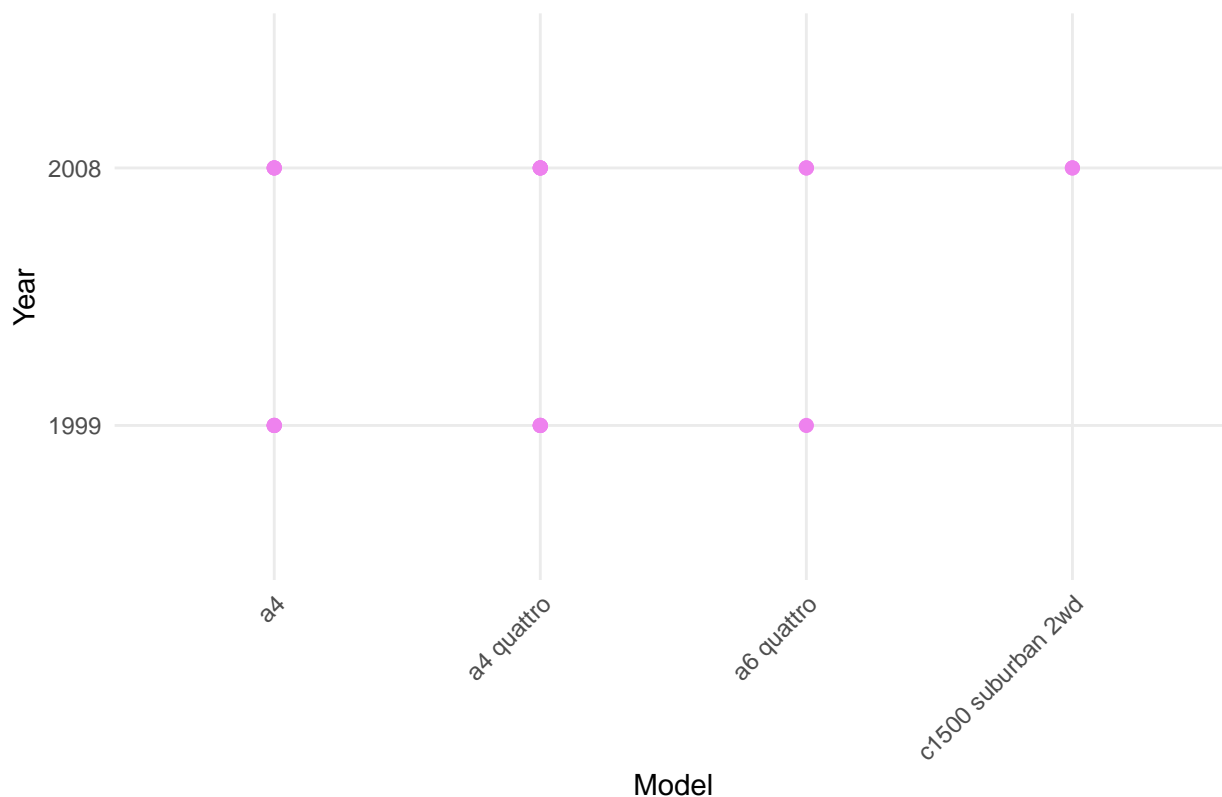


#(3.) Plot the model and the year using ggplot() (top 20 observations).

```
twenny_obser <- mpgdata[1:20, ]
```

```
ggplot(twenny_obser, aes(x = model, y = factor(year))) +
  geom_point(color = "violet", size = 2) +
  labs(title = "Model vs Year (Top 20 Observations)",
       x = "Model",
       y = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Model vs Year (Top 20 Observations)



#(4.) Using the pipe (%>%), group the model and get the number of cars per model.

```
model_g <- mpgdata %>%
  group_by(model) %>%
  summarise(carnum = n()) %>%
  arrange(desc(carnum))
```

model_g

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

#OUTPUT:

```
# A tibble: 38 x 2
#model                number_of_cars
#<chr>                <int>
```

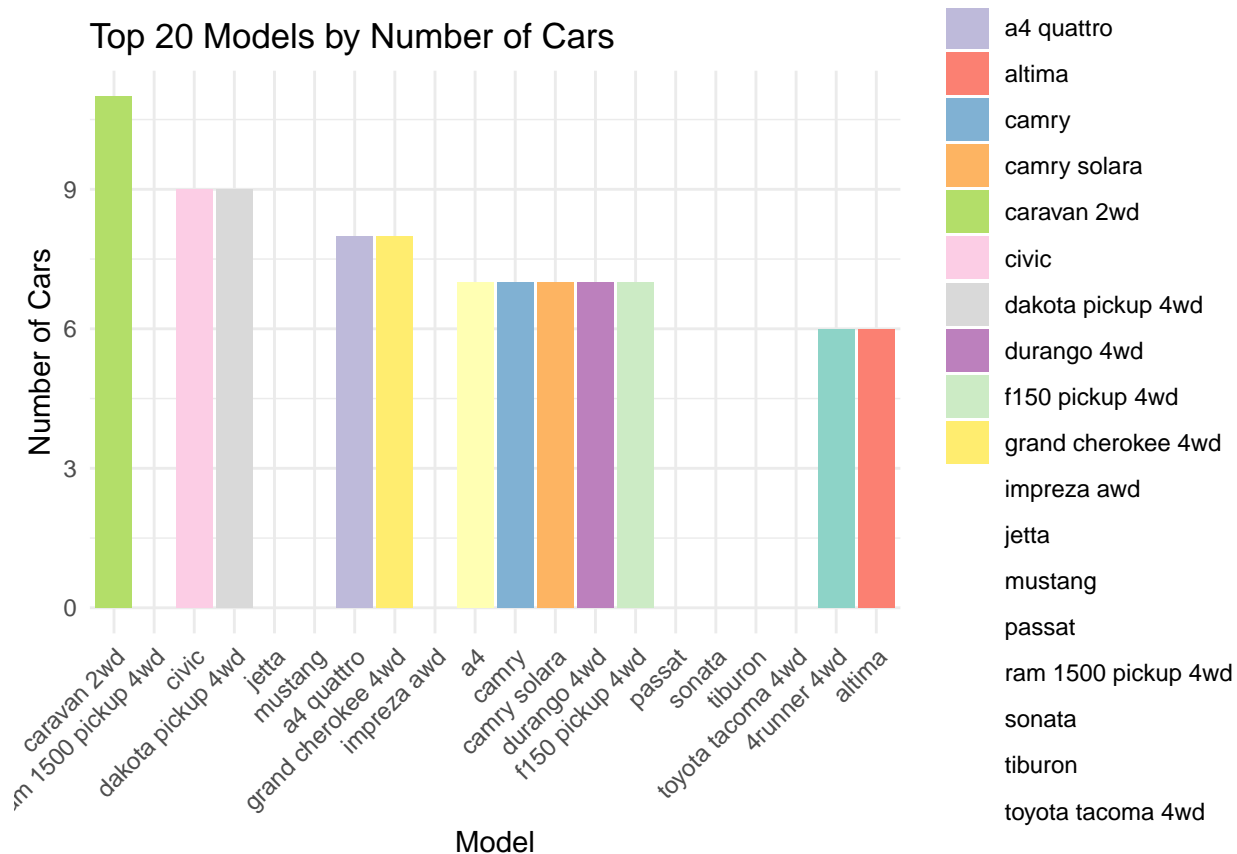
```
# 1 caravan 2wd          11
#2 ram 1500 pickup 4wd    10
#3 civic                 9
#4 dakota pickup 4wd      9
#5 jetta                 9
#6 mustang               9
#7 a4 quattro            8
#8 grand cherokee 4wd     8
#9 impreza awd           8
#10 a4                   7
```

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

```
twenny_obser <- model_g[1:20, ]

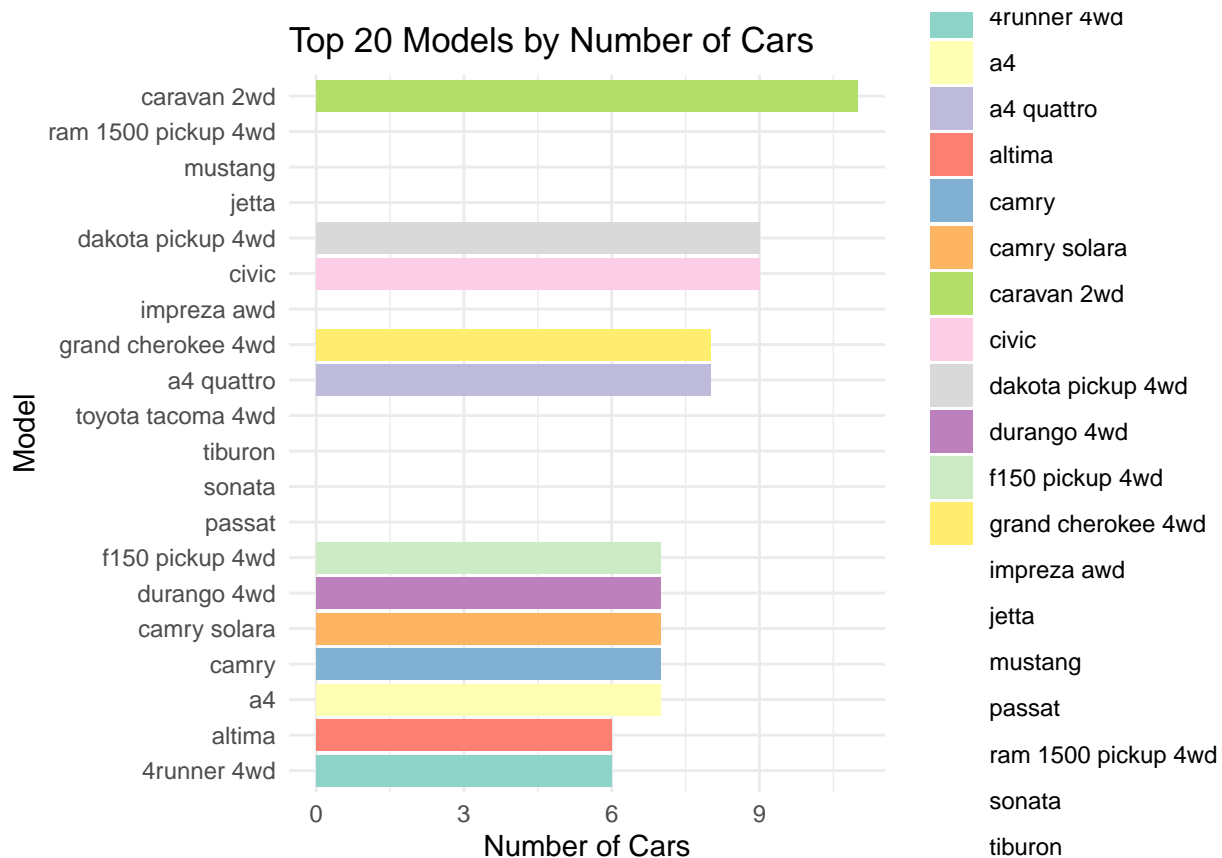
ggplot(twenny_obser, aes(x = reorder(model, -carnum), y = carnum, fill = model)) +
  geom_bar(stat = "identity") +
  labs(title = "Top 20 Models by Number of Cars",
       x = "Model",
       y = "Number of Cars") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "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. Plot using the geom_bar() + coord_flip()
ggplot(twenney_obser, 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")
```

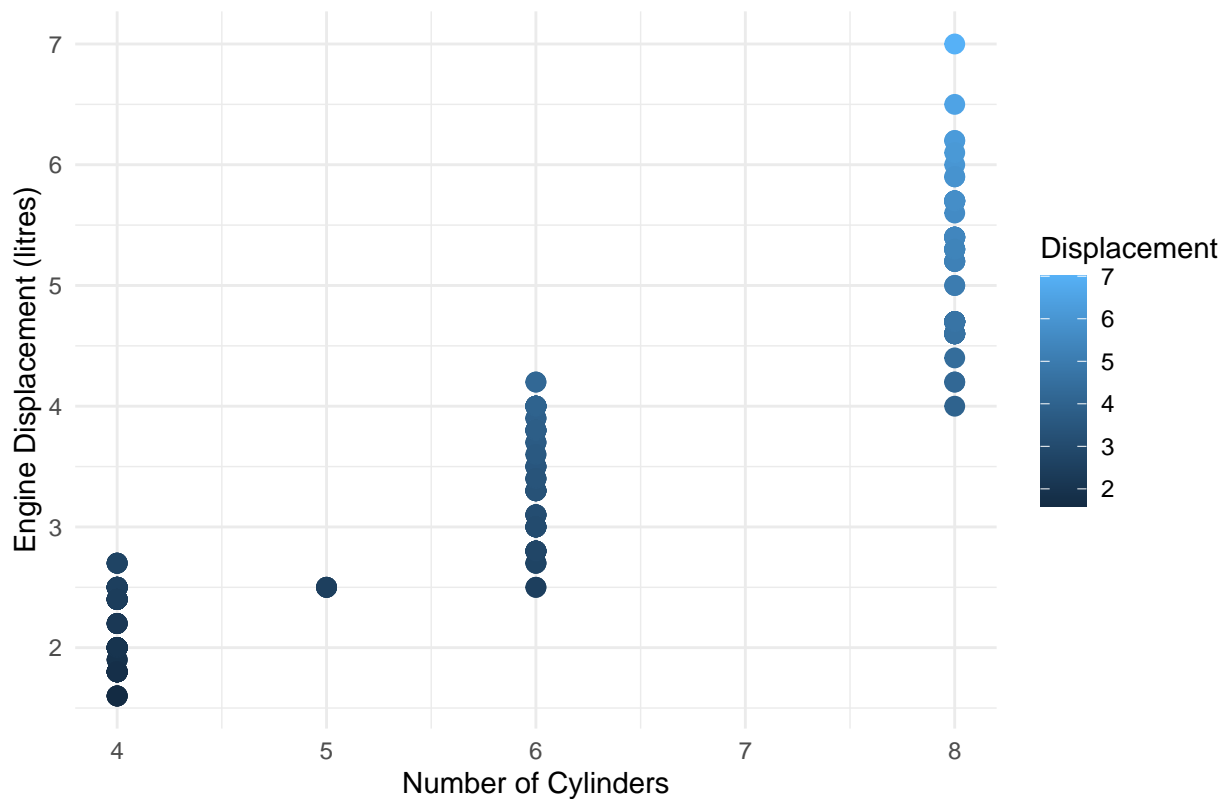
```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Set3 is 12
## Returning the palette you asked for with that many colors
```

#(5.) Plot the relationship between cyl - number of cylinders and displ - engine displacement using geom_point

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

Relationship between No. of Cylinders and Engine Displacement



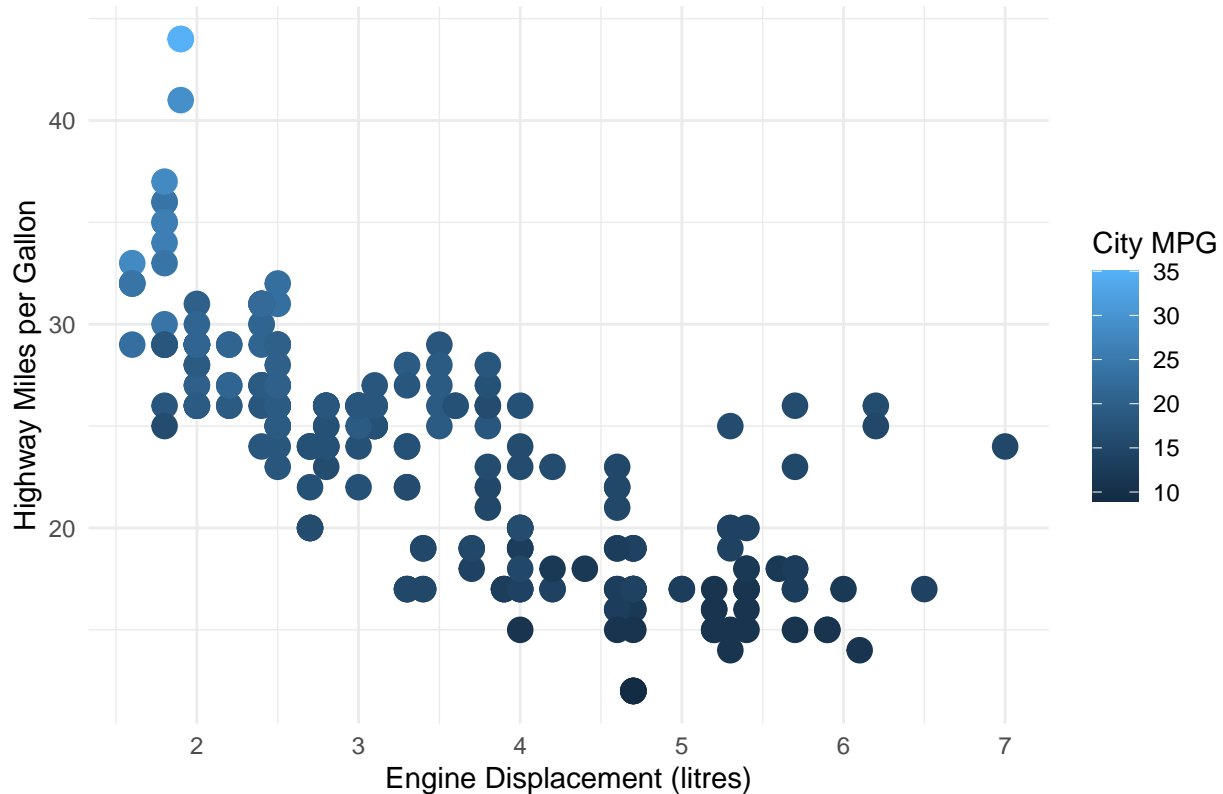
#a. How would you describe its relationship?

I would describe its relationship a clear positive relationship between the number of cylinders and engine displacement. Vehicles with fewer cylinders, such as 4-cylinder engines, have smaller displacements, typically ranging from about 1 to 3 liters. As the number of cylinders increases to six, engine displacement also increases, falling around 2.5 to 4 liters. Cars with eight cylinders have the higher displacements, reaching between 4 to 7 liters.

#6. Plot the relationship between displ (engine displacement) and hwy(highway miles per gallon).

```
ggplot(mpgdata, aes(x = displ, y = hwy, color = cty)) +
  geom_point(size = 4) +
  labs(title = "Relationship between Engine Displacement and Highway MPG",
       x = "Engine Displacement (litres)",
       y = "Highway Miles per Gallon",
       color = "City MPG") +
  theme_minimal()
```

Relationship between Engine Displacement and Highway MPG



#Result and why produced such output.

*# It produced such output beacuse it shows the relationship of hwy and displ in
#numeric value because both are continous. The engine displacement has a higher
#hwy in the first displ and gradually lower.*

#6. Import the traffic.csv onto your R environment.

```
library(readr)
traffic <- read_csv("/cloud/project/traffic.csv")
```

```
## Rows: 9 Columns: 4
```

```
## -- Column specification -----
```

```
## Delimiter: ","
```

```
## chr (1): Date
```

```
## dbl (2): Junction, Vehicles
```

```
## time (1): Time
```

```
##
```

```
## i Use `spec()` to retrieve the full column specification for this data.
```

```
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

#a. How many numbers of observation does it have? What are the variables of the traffic dataset.

```
nrow(traffic)
```

```
## [1] 9
```

```

names(traffic)

## [1] "Date"      "Time"      "Junction" "Vehicles"
#b. subset the traffic dataset into junctions.

traffic_2junc <- traffic[, c("Junction")]

traffic_2junc

## # A tibble: 9 x 1
##   Junction
##   <dbl>
## 1       1
## 2       3
## 3       1
## 4       2
## 5       1
## 6       3
## 7       1
## 8       3
## 9       2

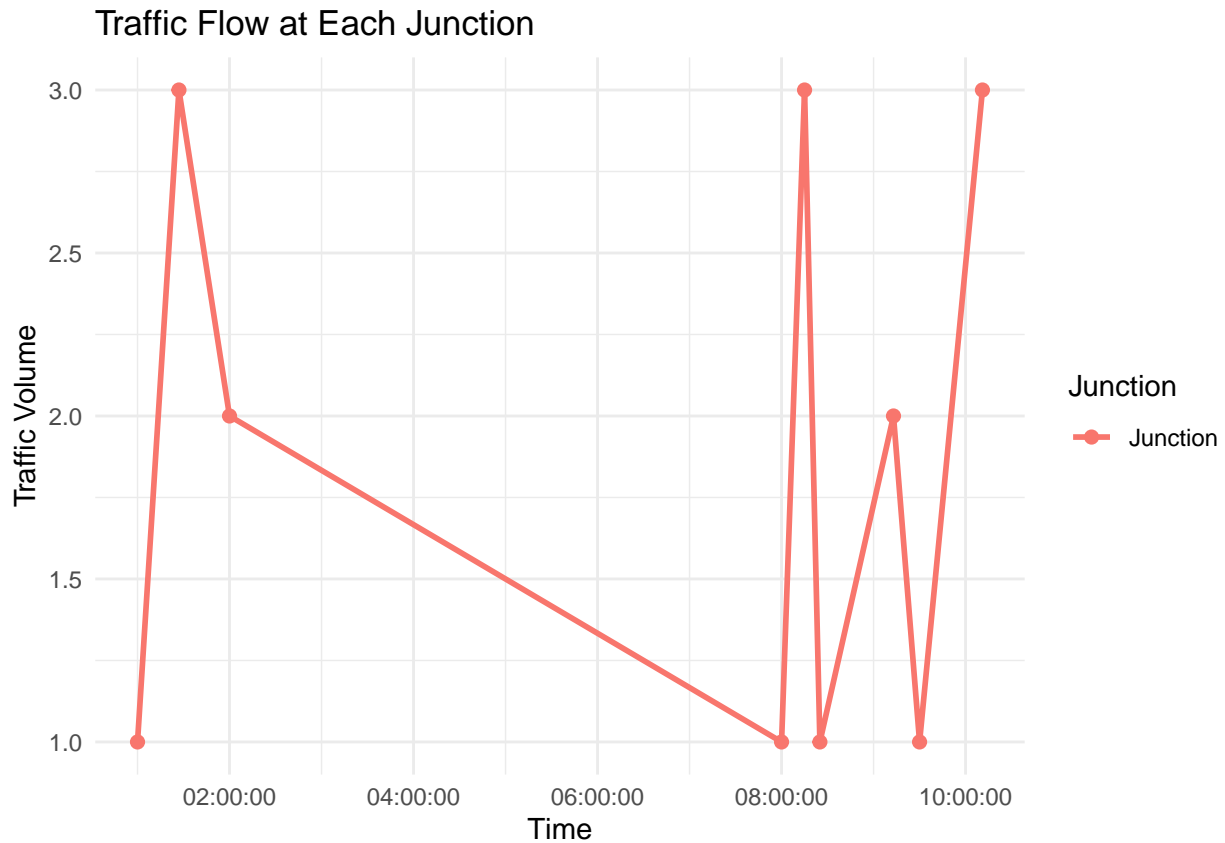
#c. Plot each junction in a using geom_line().
library(tidyr)

traffic_junc <- traffic %>%
  pivot_longer(cols = starts_with("Junction"),
               names_to = "Junction",
               values_to = "TrafficFlow")

ggplot(traffic_junc, aes(x = Time, y = TrafficFlow, color = Junction)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  labs(title = "Traffic Flow at Each Junction",
       x = "Time",
       y = "Traffic Volume") +
  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.

```



```
#7. From alexa_file.xlsx, import it to your environment
library(readxl)

alexa_file <- read_excel("/cloud/project/alexa_file.xlsx")
```

```
#a. alexa_file observation and number of columns.
```

```
alexa_obs <- nrow(alexa_file)

alexa_col <- ncol(alexa_file)

alexa_obs
```

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

```
alexa_col
```

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

```
#b. group the variations and get the total of each variations.
```

```
varg_each <- alexa_file %>%
  group_by(variation) %>%
  summarise(total = n())
```

```
varg_each
```

```
## # 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
```

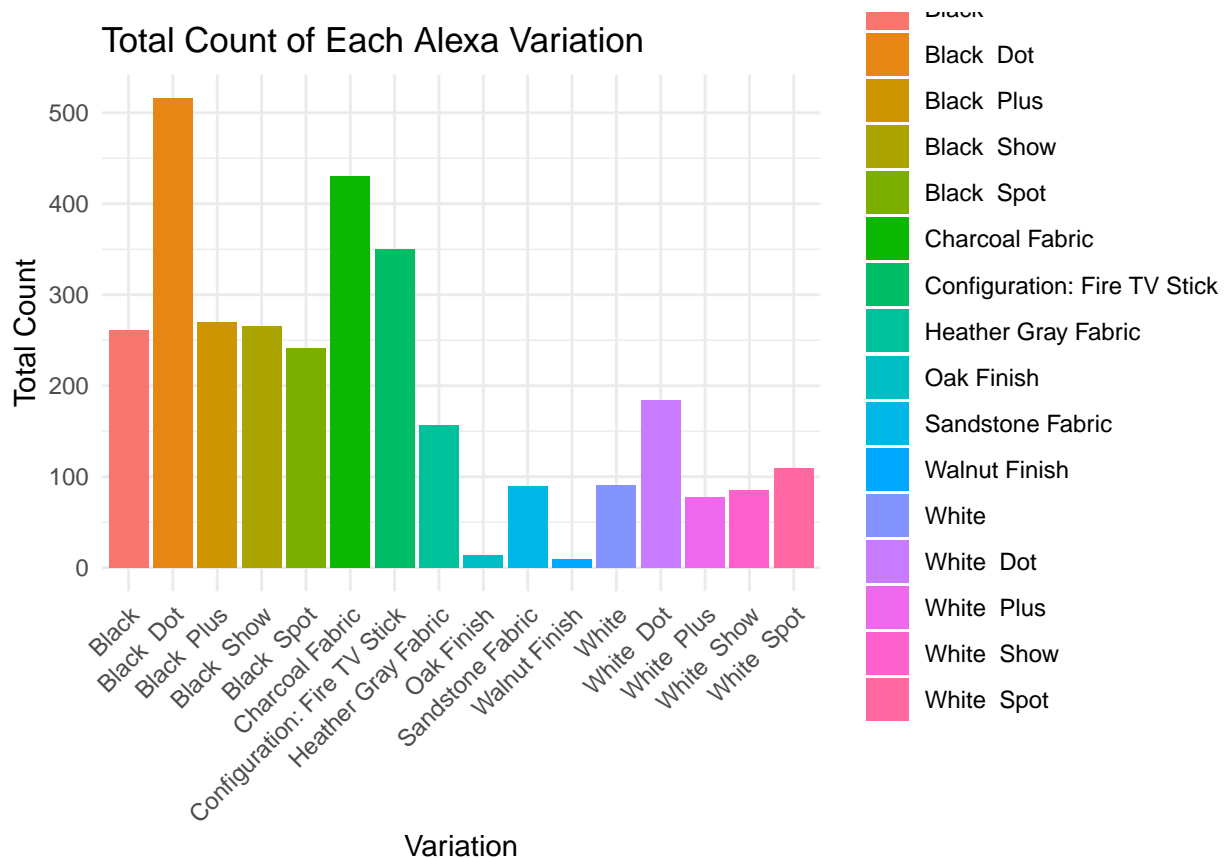
#OUTPUT:

A tibble: 16 × 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
```

#c. Plot the variations using the ggplot() function.

```
ggplot(varg_each, aes(x = variation, y = total, fill = variation)) +
  geom_bar(stat = "identity") +
  labs(
    title = "Total Count of Each Alexa Variation",
    x = "Variation",
    y = "Total Count"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
#d. Plot a geom_line() with the date and the number of verified reviews.

alex_file$date <- as.Date(alex_file$date, format="%Y-%m-%d")

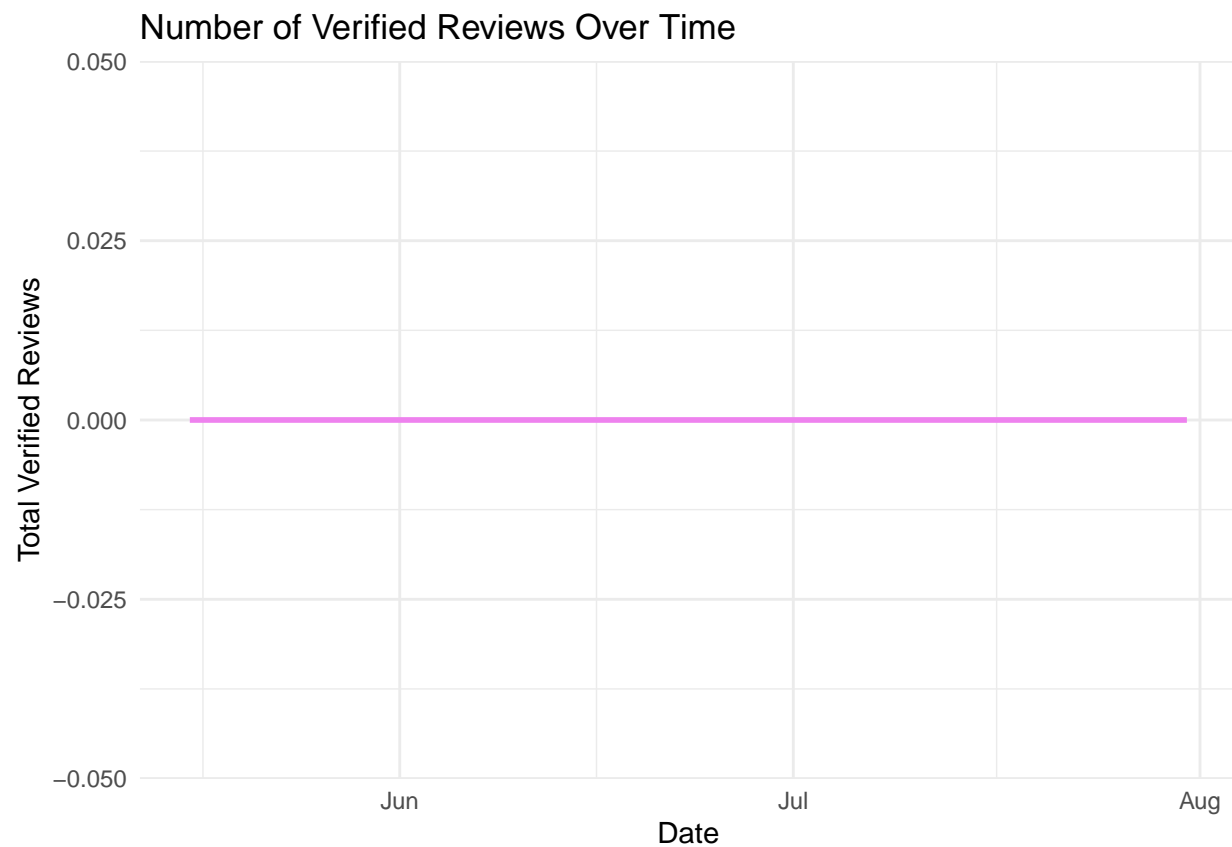
alex_file$verifiedreviews <- as.numeric(gsub(",", "", alex_file$verified_reviews))

## Warning: NAs introduced by coercion
sum(is.na(alex_file$verifiedreviews))

## [1] 3150

veri_reviews <- alex_file %>%
  group_by(date) %>%
  summarise(totalverified = sum(verifiedreviews, na.rm = TRUE))

ggplot(veri_reviews, aes(x = date, y = totalverified)) +
  geom_line(color = "violet", linewidth = 1) +
  labs(
    title = "Number of Verified Reviews Over Time",
    x = "Date",
    y = "Total Verified Reviews"
  ) +
  theme_minimal()
```



#e.relationship of variations and ratings and who got the most highest in rating.

```
high_rating <- alexa_file %>%
  group_by(variation) %>%
  summarise(avg_rating = mean(rating, na.rm = TRUE)) %>%
  arrange(desc(avg_rating))
```

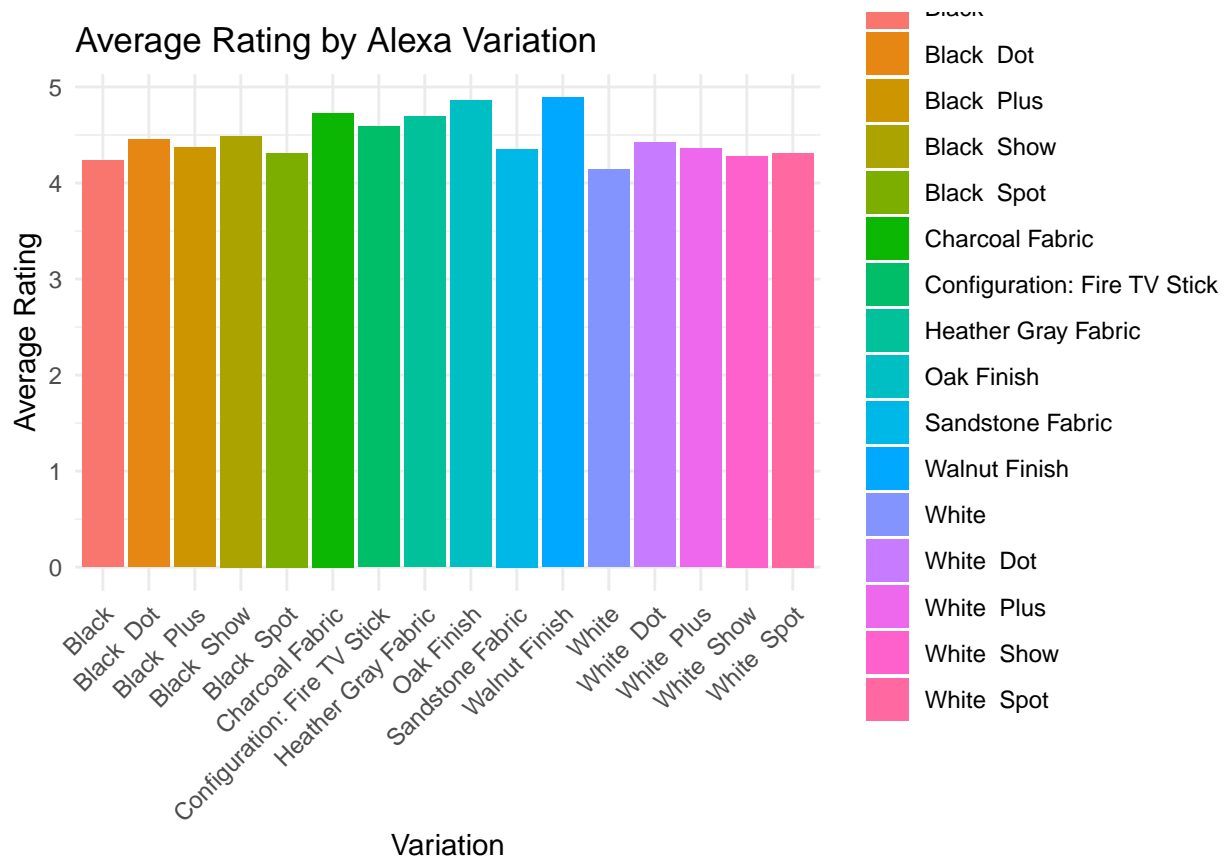
```
high_rating
```

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

```
ggplot(high_rating, aes(x = variation, y = avg_rating, fill = variation)) +  
  geom_bar(stat = "identity") +  
  labs(  
    title = "Average Rating by Alexa Variation",  
    x = "Variation",  
    y = "Average Rating"  
  ) +  
  theme_minimal() +  
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



#The variation that got the most highest in rating is the Walnut Finish as it has the highest bar on t