RWorksheet_Camarista#4c

John Lyxton Camarista

2024-11-01

##1. Use the dataset mpg - a. Show your solutions on how to import a csv file into the environment.

```
library(readr)


mpg_data <- read_csv("E:/Github/Data Science Worksheets/DataScience_Worksheets_Camarista/Worksheet#4/mpg
```

```
## New names:
## Rows: 234 Columns: 12
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (6): manufacturer, model, trans, drv, fl, class dbl (6): ...1, displ, year,
## cyl, cty, hwy
## 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.
## * `` -> `...1`
```

```
head(mpg_data)
```

```
## # A tibble: 6 x 12
##    ...1 manufacturer model displ  year   cyl trans drv     cty   hwy fl    class
##   <dbl> <chr>        <chr> <dbl> <dbl> <dbl> <chr> <chr> <dbl> <dbl> <chr> <chr>
## 1     1 audi         a4      1.8  1999     4 auto~ f        18    29 p     comp~
## 2     2 audi         a4      1.8  1999     4 manu~ f        21    29 p     comp~
## 3     3 audi         a4      2    2008     4 manu~ f        20    31 p     comp~
## 4     4 audi         a4      2    2008     4 auto~ f        21    30 p     comp~
## 5     5 audi         a4      2.8  1999     6 auto~ f        16    26 p     comp~
## 6     6 audi         a4      2.8  1999     6 manu~ f        18    26 p     comp~
```

- b. Which variables from mpg dataset are categorical?
  - The categorical variables are: manufacturer, model, year, trans, drv, fl, and class.
- c. Which are continuous variables?
  *The continous variables are: displ, cyl, cty, and hwy.

##2. a. Which manufacturer has the most models in this data set? Which model has the most variations? Show your answer. - a. Group the manufacturers and find the unique models. Show your codes and result.

```r
# Load necessary library
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
```

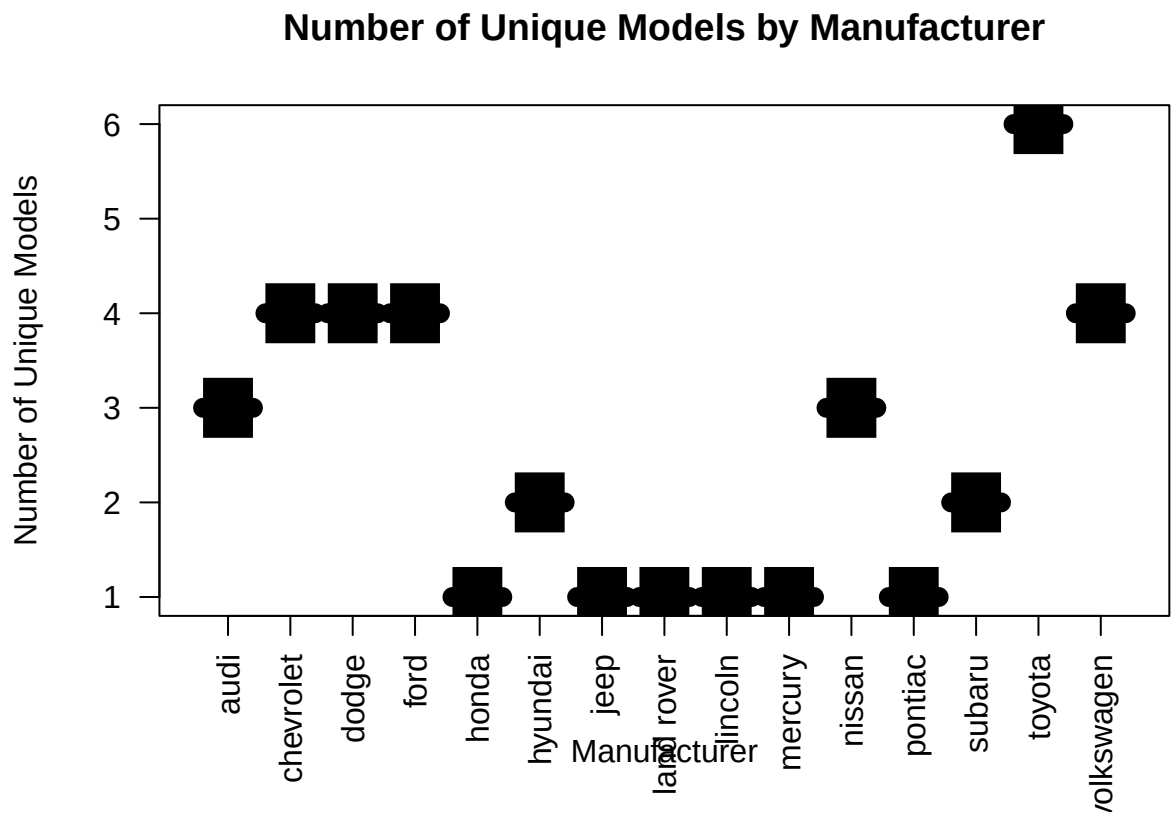
```r
# Get unique models for each manufacturer
unique_models <- mpg_data %>%
  select(manufacturer, model) %>%
  distinct() %>%
  arrange(manufacturer)

# Display the result
unique_models
```

```
## # A tibble: 38 x 2
##    manufacturer model
##    <chr>        <chr>
##  1 audi         a4
##  2 audi         a4 quattro
##  3 audi         a6 quattro
##  4 chevrolet    c1500 suburban 2wd
##  5 chevrolet    corvette
##  6 chevrolet    k1500 tahoe 4wd
##  7 chevrolet    malibu
##  8 dodge        caravan 2wd
##  9 dodge        dakota pickup 4wd
## 10 dodge        durango 4wd
## # i 28 more rows
```

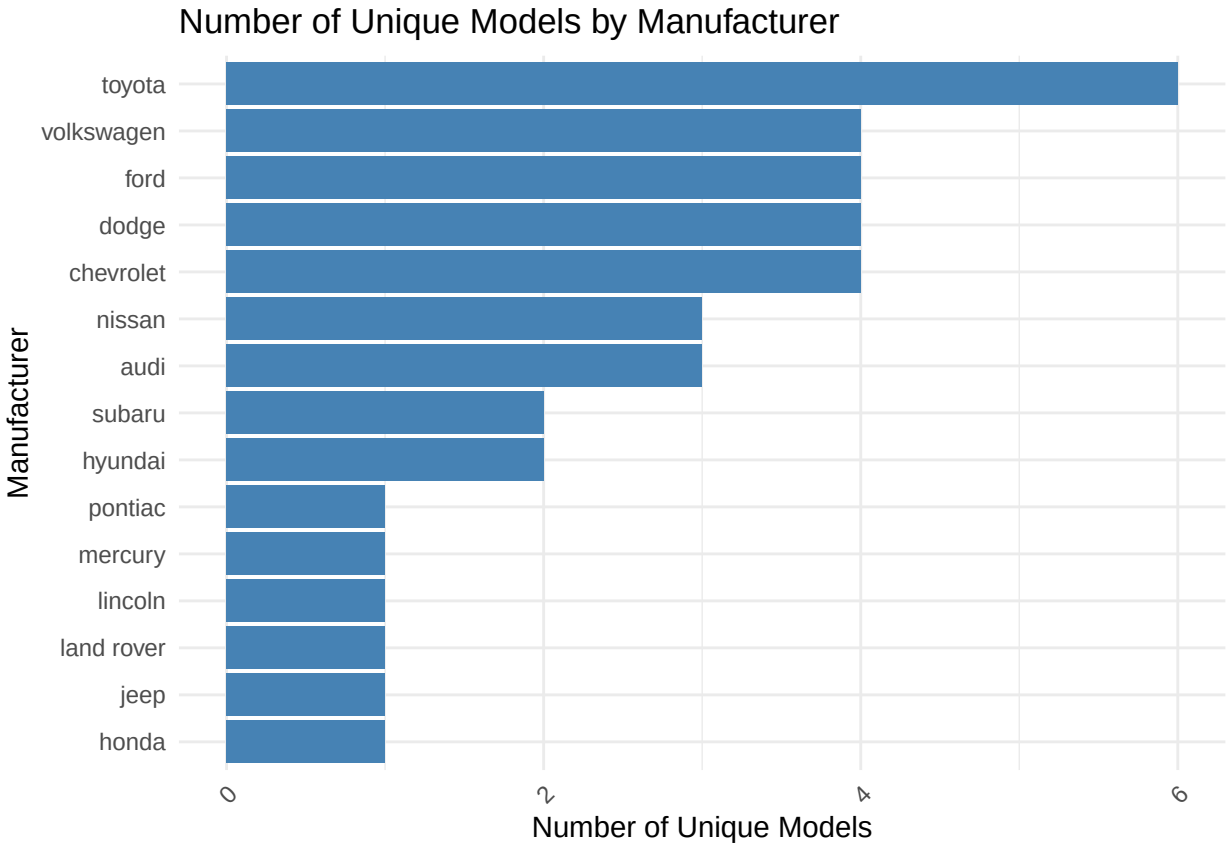
- b. Graph the result by using plot() and ggplot(). Write the codes and its result.

```r
library(dplyr)

model_count <- mpg_data %>%
  select(manufacturer, model) %>%
  distinct() %>%
  group_by(manufacturer) %>%
  summarise(model = n())

model_count$manufacturer <- factor(model_count$manufacturer, levels = unique(model_count$manufacturer))

plot(
```

```
  model_count$manufacturer,
  model_count$model,
  type = "h",  lines,
  main = "Number of Unique Models by Manufacturer",
  xlab = "Manufacturer",
  ylab = "Number of Unique Models",
  col = "skyblue",
  las = 2,
  lwd = 10
)
```

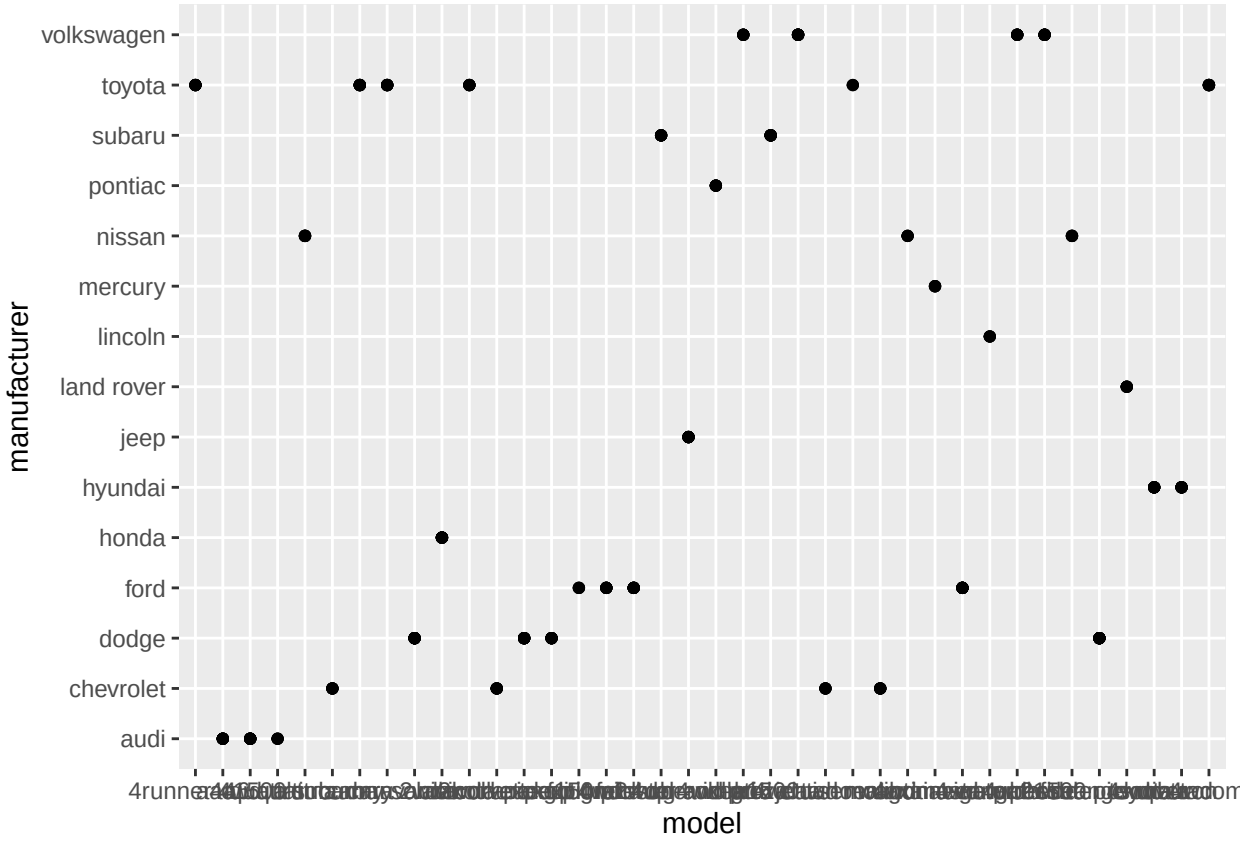



```
library(ggplot2)

ggplot(model_count, aes(x = reorder(manufacturer, model), y = model)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Number of Unique Models by Manufacturer",
       x = "Manufacturer",
       y = "Number of Unique Models") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



##2. b. Same dataset will be used. You are going to show the relationship of the modeland the manufacturer. - a. What does ggplot(mpg, aes(model, manufacturer)) + geom_point() show?
* Each point represents an individual observation in the mpg dataset, with each combination of model and manufacturer plotted as a point.

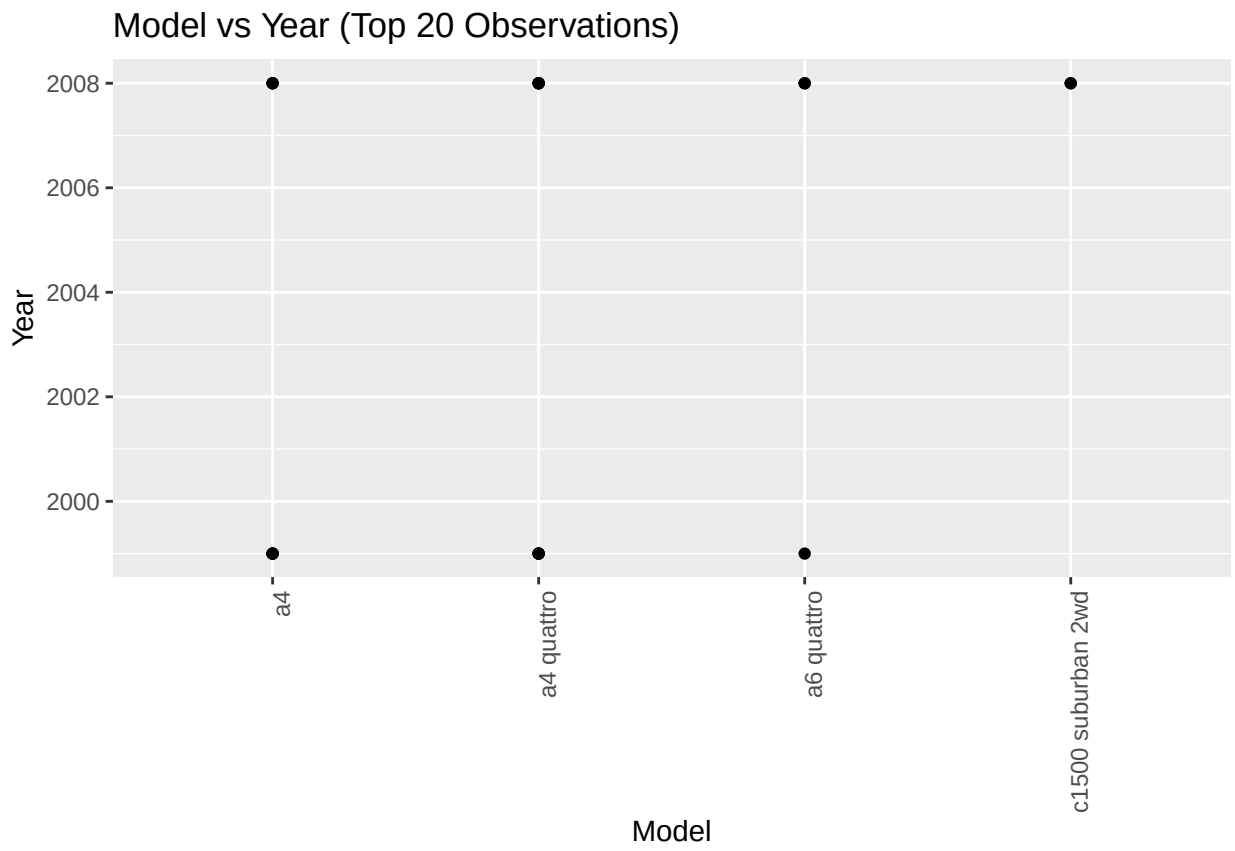
```
ggplot(mpg, aes(model, manufacturer)) + geom_point()
```



- b. For you, is it useful? If not, how could you modify the data to make it more informative?
* Is not particularly useful in its current form because it's cluttered due to the large number of unique models on the x-axis.
* A bar chart or heatmap showing counts or averages would provide a clearer, more informative view than the original scatter plot.

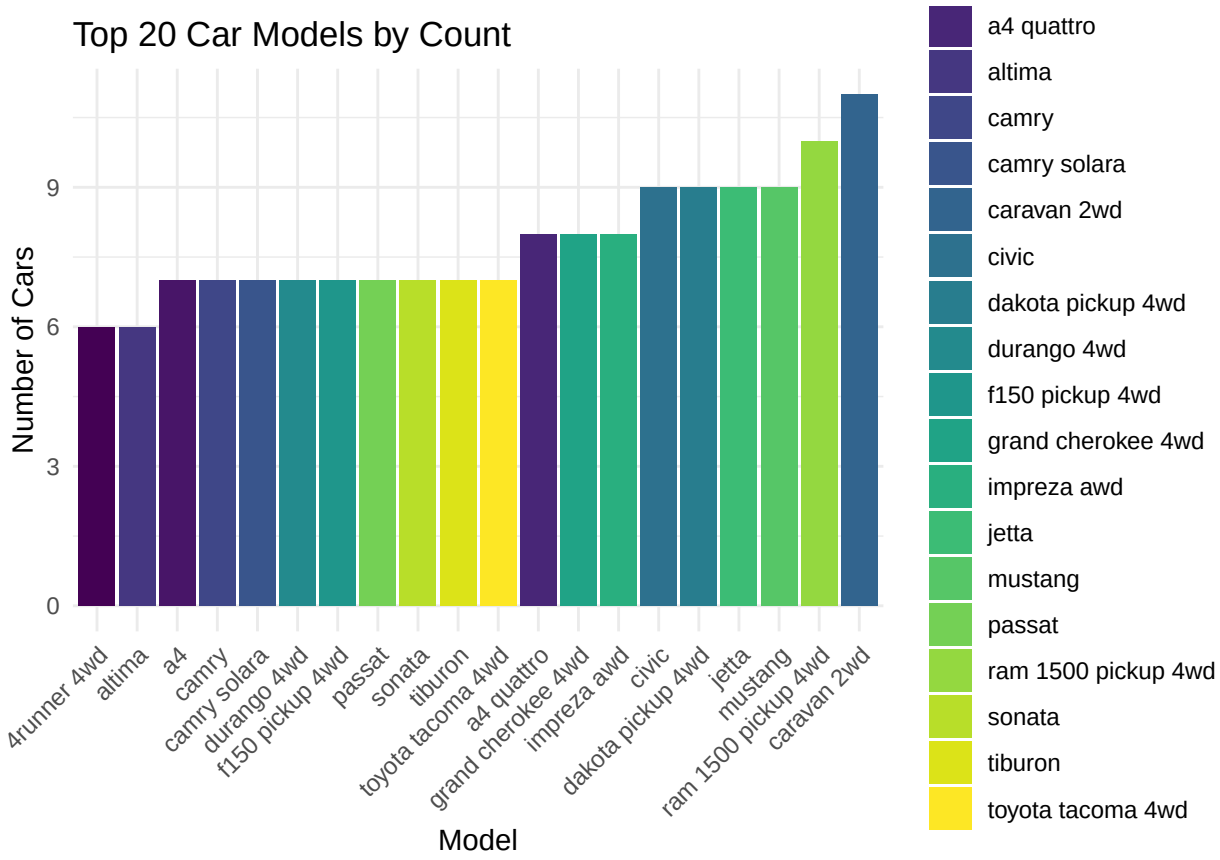
##3. Plot the model and the year using ggplot(). Use only the top 20 observations. Write the codes and its results.

```
ggplot(head(mpg_data, 20), aes(x = model, y = year)) +
  geom_point() +
  labs(title = "Model vs Year (Top 20 Observations)", x = "Model", y = "Year") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



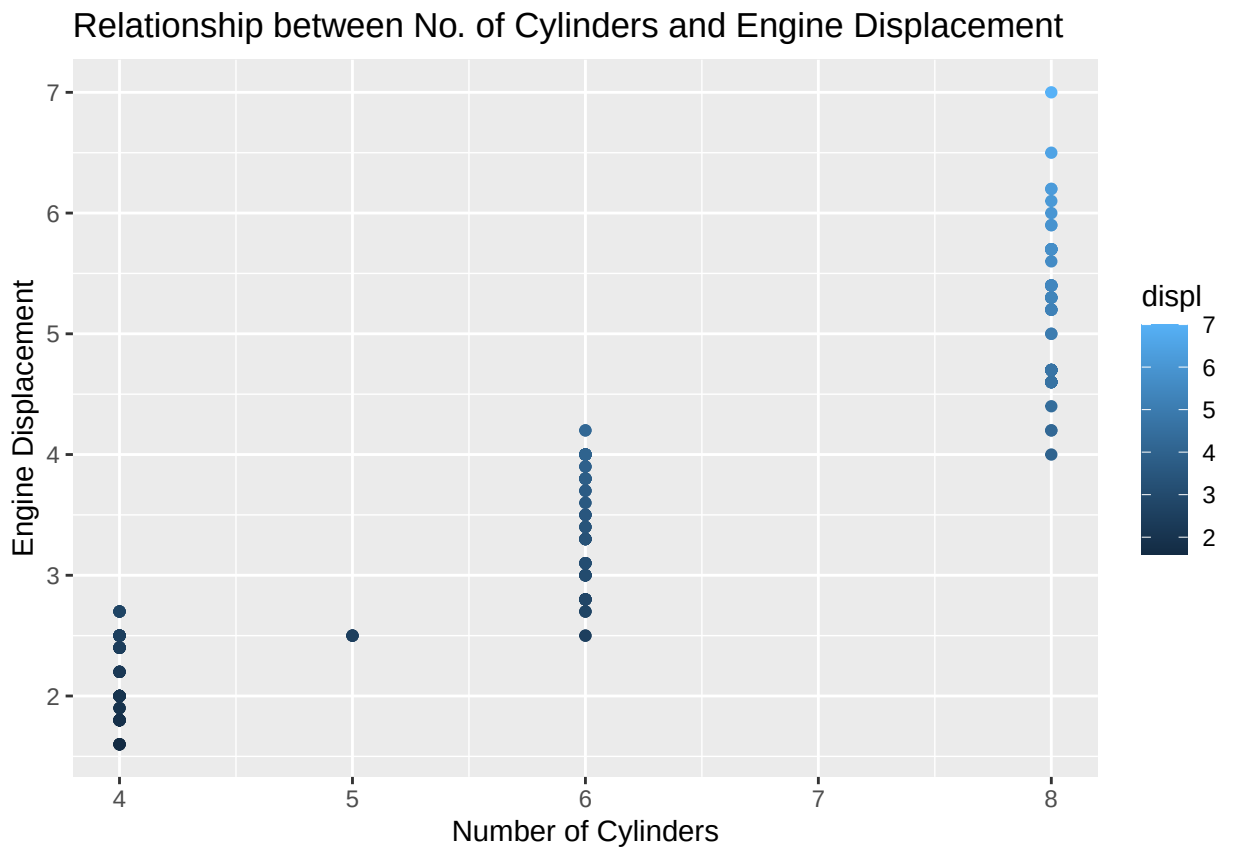
##4. Using the pipe (%>%), group the model and get the number of cars per model. Show codes and its result.
- a. Plot using geom_bar() using the top 20 observations only. The graphs should have a title, labels and colors. Show code and results.

```
library(dplyr)
library(ggplot2)

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

top_20_models <- model_counts %>%
  head(20)

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



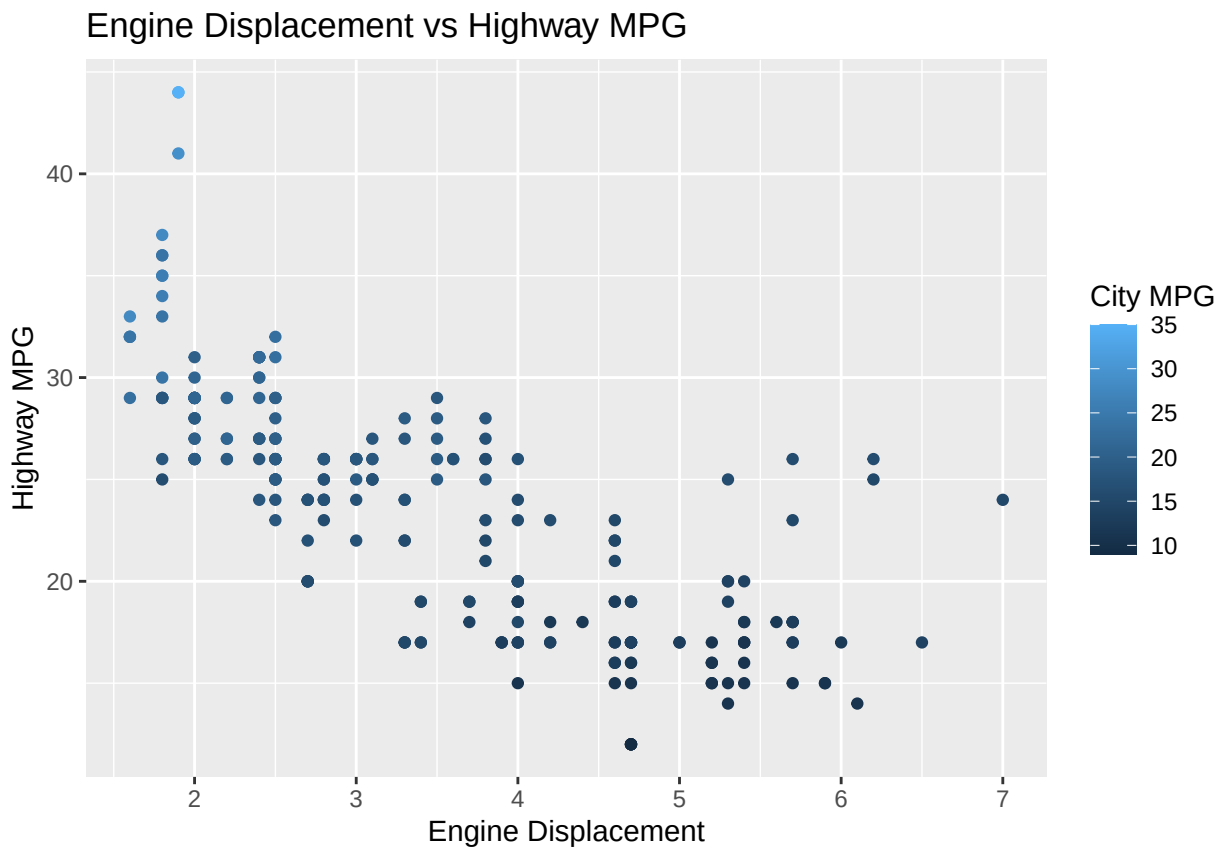
##5. Plot the relationship between cyl - number of cylinders and displ - engine displacement using geom_point with aesthetic color = engine displacement. Title should be "Relationship between No. of Cylinders and Engine Displacement".
- a. How would you describe its relationship? Show the codes and its result.

```
ggplot(mpg_data, 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")
```



##6. a. Plot the relationship between displ (engine displacement) and hwy(highway miles per gallon). Mapped it with a continuous variable you have identified in #1-c. What is its result? Why it produced such output?

```
ggplot(mpg_data, aes(x = displ, y = hwy, color = cty)) +
  geom_point() +
  labs(title = "Engine Displacement vs Highway MPG",
       x = "Engine Displacement", y = "Highway MPG", color = "City MPG")
```



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

```
library(readr)

traffic_data <- read_csv("E:/Github/Data Science Worksheets/DataScience_Worksheets_Camarista/Worksheet#
```

```
## Rows: 48120 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): DateTime
## dbl (3): Junction, Vehicles, ID
##
## 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.
```

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

- a. How many numbers of observation does it have? What are the variables of the traffic dataset the Show your answer.

```
traffic_obs <- nrow(traffic_data)
print(paste("There are", traffic_obs, "observations"))
```

```
## [1] "There are 48120 observations"
```

```r
print("The variables in the traffic dataset are: ")
```

```
## [1] "The variables in the traffic dataset are: "
```

```r
names(traffic_data)
```

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

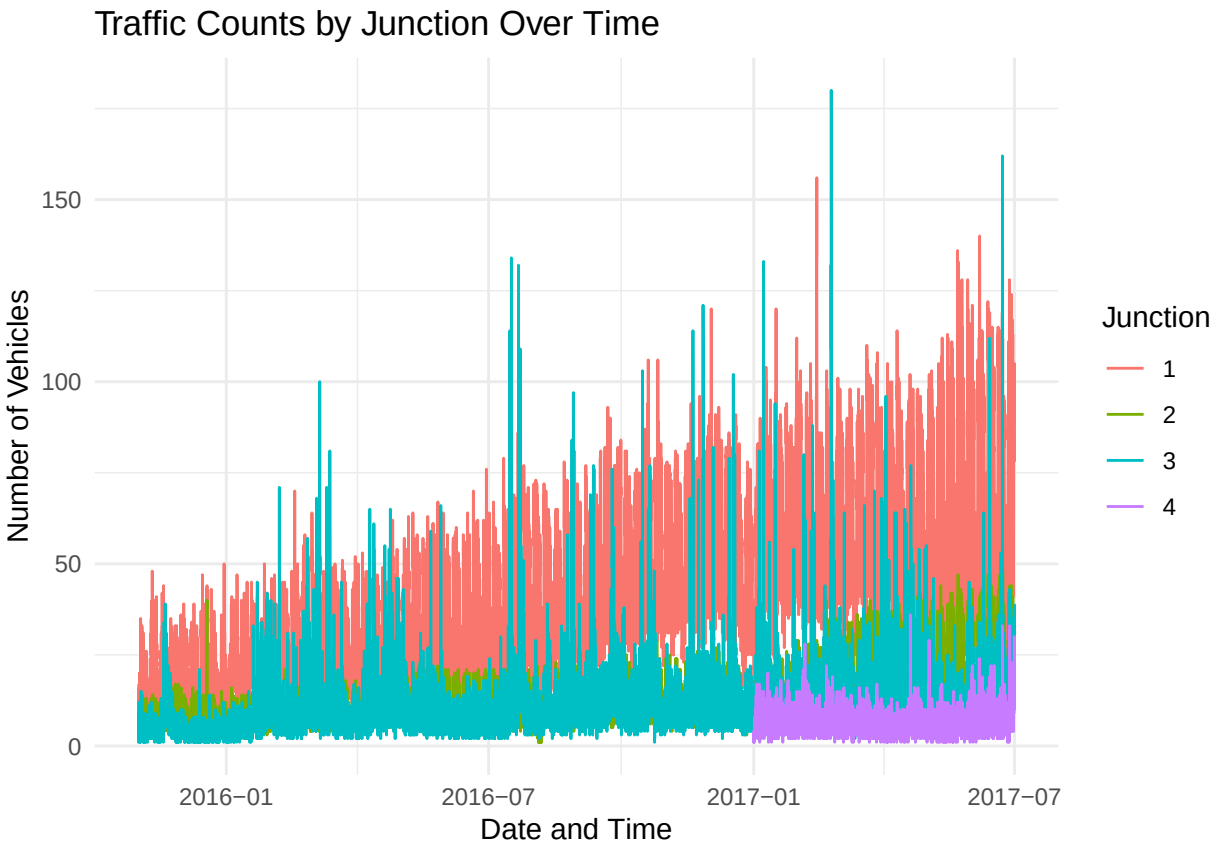
- b. subset the traffic dataset into junctions. What is the R codes and its output?

```r
library(dplyr)

# Create subsets for each junction
junction_data <- traffic_data %>%
  group_by(Junction) %>%
  group_split()
print(junction_data[[1]])
```

```
## # A tibble: 14,592 x 4
##    DateTime            Junction Vehicles          ID
##    <dttm>                 <dbl>    <dbl>       <dbl>
##  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
##  7 2015-11-01 06:00:00        1        9 20151101061
##  8 2015-11-01 07:00:00        1        8 20151101071
##  9 2015-11-01 08:00:00        1       11 20151101081
## 10 2015-11-01 09:00:00        1       12 20151101091
## # i 14,582 more rows
```

- c. Plot each junction in a using geom_line(). Show your solution and output.

```r
library(ggplot2)
  ggplot(traffic_data, aes(x = DateTime, y = Vehicles, color = factor(Junction))) +
    geom_line() +
    labs(title = "Traffic Counts by Junction Over Time",
         x = "Date and Time",
         y = "Number of Vehicles",
         color = "Junction") +
    theme_minimal()
```



##7. From alexa_file.xlsx, import it to your environment - a. How many observations does alexa_file has? What about the number of columns? Show your solution and answer.

```
library(readxl)

alexa_data <- read_excel("E:/Github/Data Science Worksheets/DataScience_Worksheets_Camarista/Worksheet#4

alexa_obs <- nrow(alexa_data)
alexa_num_cols <- ncol(alexa_data)

print(paste("There are ", alexa_obs, "observations in the data"))
```

```
## [1] "There are  3150 observations in the data"
```

```
print(paste("There are ", alexa_num_cols, "columns int the data"))
```

```
## [1] "There are  5 columns int the data"
```

- b. group the variations and get the total of each variations. Use dplyr package. Show solution and answer.

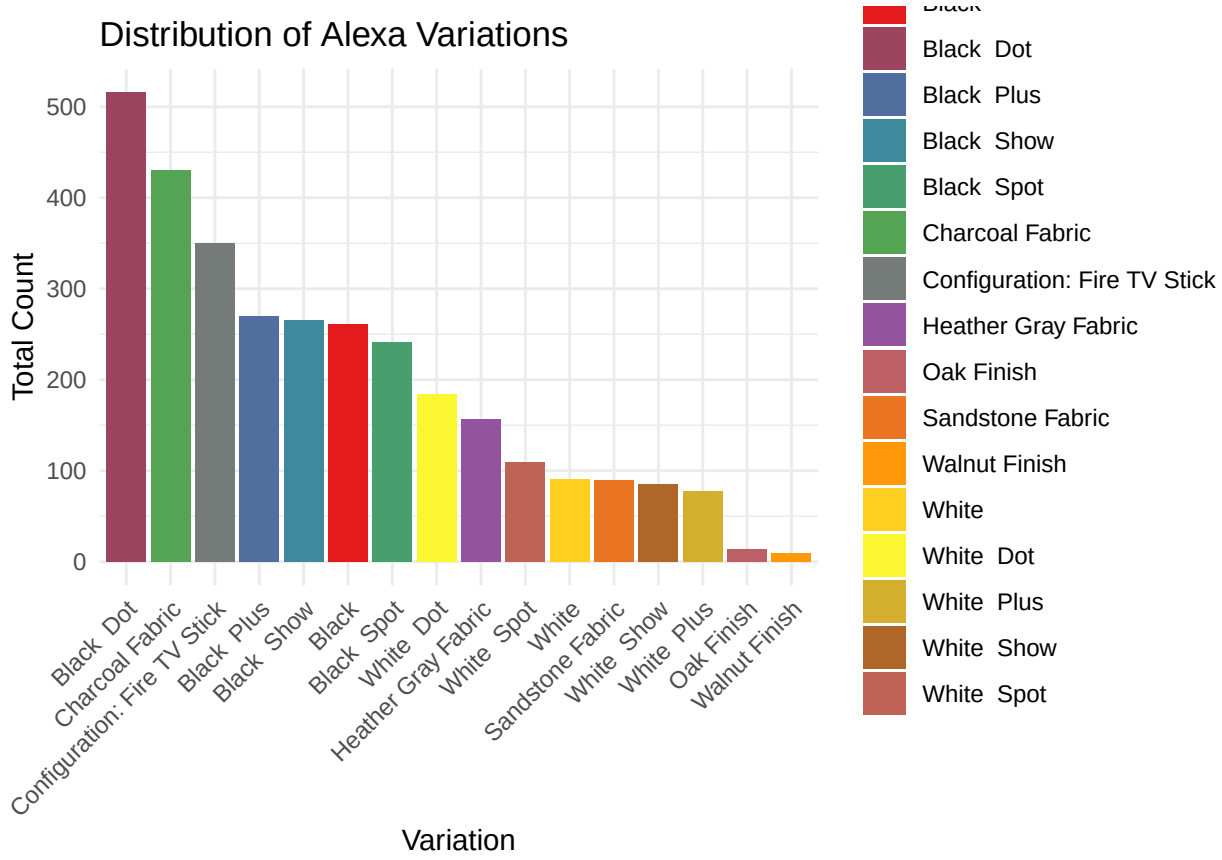
```
library(dplyr)

alexa_variation_counts <- alexa_data %>%
```

```
  group_by(variation) %>%
  summarise(Total = n())

print(alexa_variation_counts)
```

```
## # 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
```

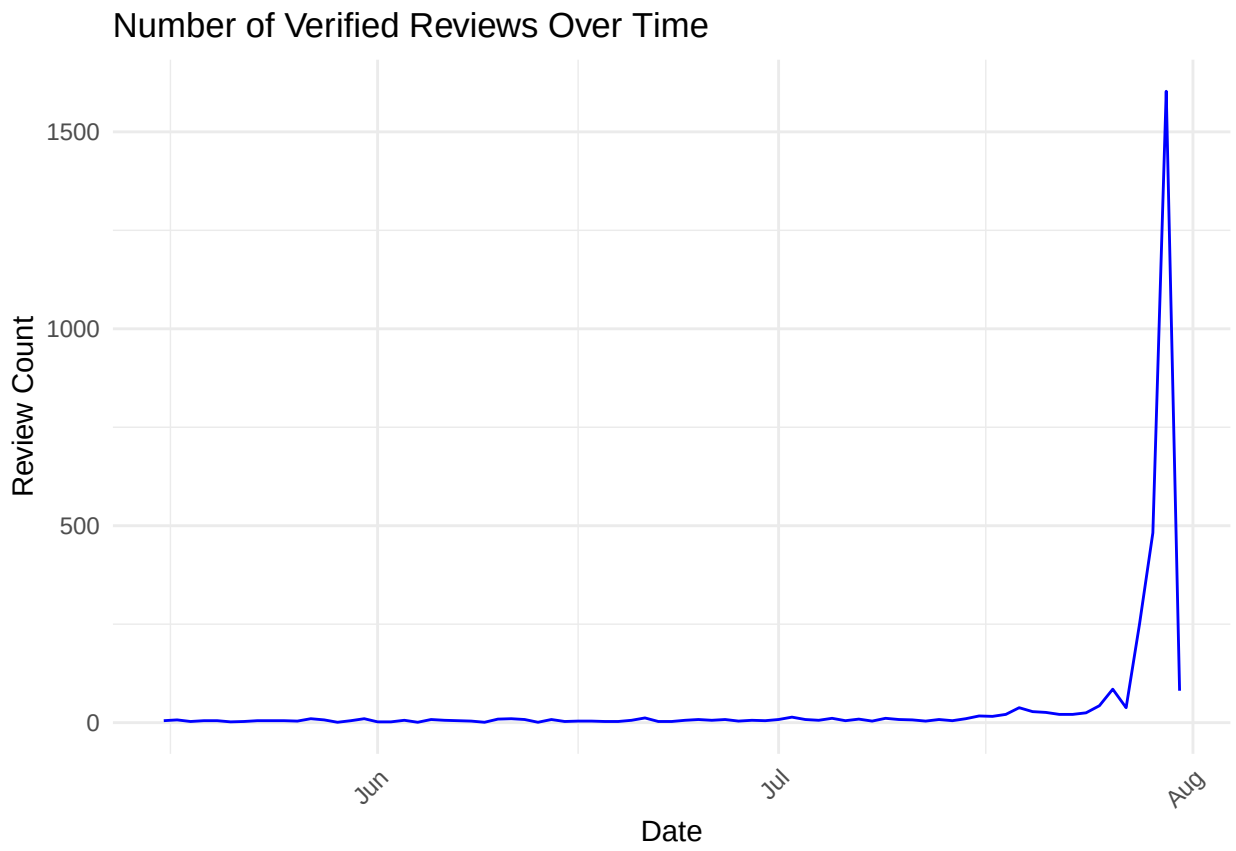
- c. Plot the variations using the ggplot() function. What did you observe? Complete the details of the graph. Show solution and answer.

```
library(ggplot2)
library(RColorBrewer)

custom_palette <- colorRampPalette(brewer.pal(9, "Set1"))(20)

ggplot(alexa_variation_counts, aes(x = reorder(variation, -Total), y = Total, fill = variation)) +
  geom_bar(stat = "identity") +
  labs(
    title = "Distribution of Alexa Variations",
    x = "Variation",
    y = "Total Count"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = custom_palette)
```



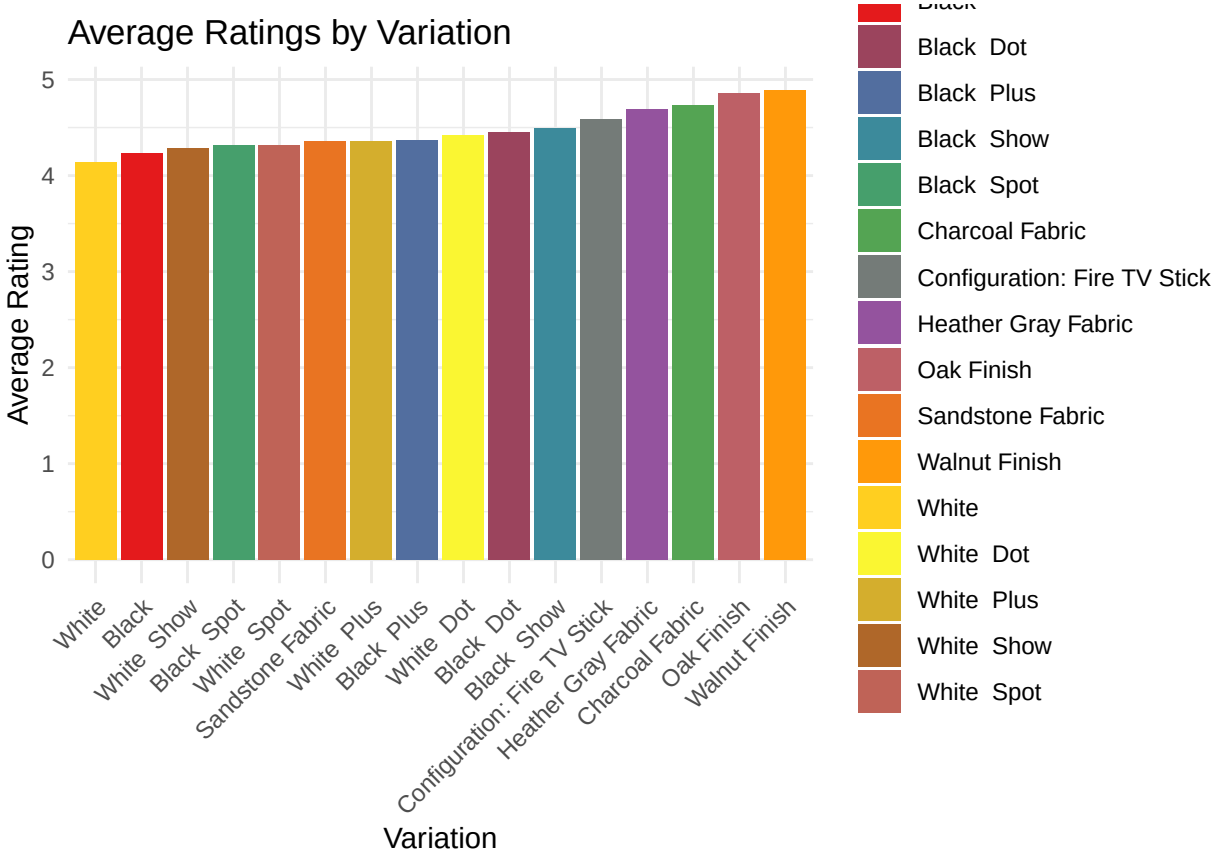
- d. Plot a geom_line() with the date and the number of verified reviews. Complete the details of the graphs. Show your answer and solution.

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

daily_reviews <- alexa_data %>%
  group_by(date) %>%
  summarise(review_count = n(), .groups = 'drop')

ggplot(daily_reviews, aes(x = date, y = review_count)) +
  geom_line(color = "blue") +
  labs(
    title = "Number of Verified Reviews Over Time",
    x = "Date",
    y = "Review Count"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



- e. Get the relationship of variations and ratings. Which variations got the most highest in rating? Plot a graph to show its relationship. Show your solution and answer.

```
# Calculate the average rating by variation
variation_ratings <- alexa_data %>%
  group_by(variation) %>%
  summarise(Average_Rating = mean(rating, na.rm = TRUE), .groups = 'drop') %>%
  arrange(desc(Average_Rating))

# Plot the average ratings by variation
custom_palette <- colorRampPalette(brewer.pal(9, "Set1"))(20)

ggplot(variation_ratings, aes(x = reorder(variation, Average_Rating), y = Average_Rating, fill = variati
  geom_bar(stat = "identity") +
  labs(
    title = "Average Ratings by Variation",
    x = "Variation",
    y = "Average Rating"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = custom_palette)
```


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