Regacho_Worksheet#4c

1. Use the dataset mpg

```
# a. Show your solutions on how to import a csv file into the environment.

# First, load ggplot2
library(ggplot2)

# Method 1: Using read.csv (base R)
# mpg <- read.csv("mpg.csv")

# Method 2: Using read_csv from readr package (faster and more flexible)
 #library(readr)
 #mpg <- read_csv("mpg.csv")


# Check the structure after import
# str(mpg)
# head(mpg)

# b. Which variables from mpg data set are categorical?

categorical_vars <- c(
  "manufacturer",    # manufacturer name (factor)
  "model",           # model name (factor)
  "trans",           # type of transmission (factor)
  "drv",             # drive train type - f, r, 4 (factor)
  "fl",              # fuel type (factor)
  "class"            # type of car (factor)
)

# Verify by checking data types
sapply(mpg, class)
```

```
## manufacturer        model        displ         year          cyl        trans
##  "character"  "character"    "numeric"    "integer"    "integer"  "character"
##          drv          cty          hwy           fl        class
##  "character"    "integer"    "integer"  "character"  "character"
```

```
# Or check unique values
sapply(mpg[categorical_vars], function(x) length(unique(x)))
```

```
## manufacturer        model        trans          drv           fl        class
##           15           38           10            3            5            7
```

```r
# c. Which are continuous variables?

continuous_vars <- c(
  "displ",        # engine displacement in liters (numeric)
  "year",         # year of manufacture (numeric/integer)
  "cyl",          # number of cylinders (numeric/integer)
  "cty",          # city miles per gallon (numeric)
  "hwy"           # highway miles per gallon (numeric)
)

# Verify they are numeric
sapply(mpg[continuous_vars], class)
```

```
##     displ      year       cyl       cty       hwy
## "numeric" "integer" "integer" "integer" "integer"
```

```r
# Summary statistics for continuous variables
summary(mpg[continuous_vars])
```

```
##      displ            year           cyl             cty             hwy
##  Min.   :1.600   Min.   :1999   Min.   :4.000   Min.   : 9.00   Min.   :12.00
##  1st Qu.:2.400   1st Qu.:1999   1st Qu.:4.000   1st Qu.:14.00   1st Qu.:18.00
##  Median :3.300   Median :2004   Median :6.000   Median :17.00   Median :24.00
##  Mean   :3.472   Mean   :2004   Mean   :5.889   Mean   :16.86   Mean   :23.44
##  3rd Qu.:4.600   3rd Qu.:2008   3rd Qu.:8.000   3rd Qu.:19.00   3rd Qu.:27.00
##  Max.   :7.000   Max.   :2008   Max.   :8.000   Max.   :35.00   Max.   :44.00
```

2. Which manufacturer has the most models in this data set? Which model has the most variations? Show your answer.

```r
# a. Group the manufacturers and find the unique models. Show your codes and result.

# Load packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.6
## v forcats   1.0.1     v stringr   1.6.0
## v lubridate 1.9.4     v tibble    3.3.0
## v purrr     1.2.0     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(ggplot2)

# Manufacturer with the most models
```

```r
manufacturer_summary <- mpg %>%
  group_by(manufacturer) %>%
  summarise(
    num_models = n_distinct(model),
    total_cars = n()
  ) %>%
  arrange(desc(num_models))

print(manufacturer_summary)
```

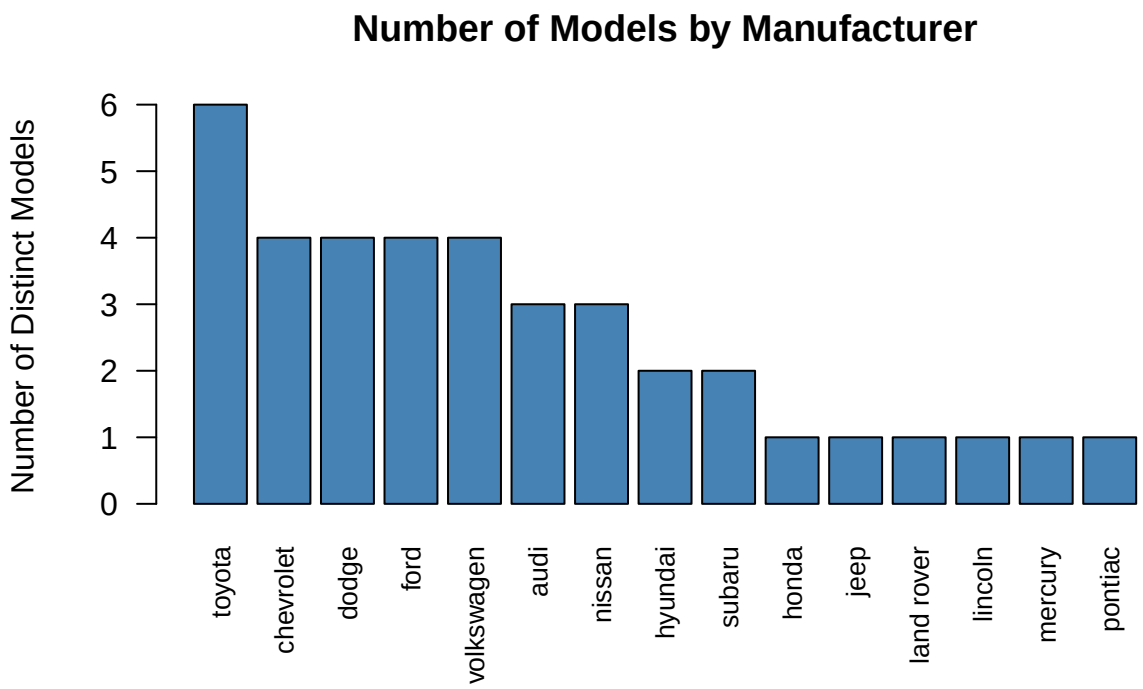
```
## # A tibble: 15 x 3
##    manufacturer num_models total_cars
##    <chr>             <int>      <int>
##  1 toyota                6         34
##  2 chevrolet             4         19
##  3 dodge                 4         37
##  4 ford                  4         25
##  5 volkswagen            4         27
##  6 audi                  3         18
##  7 nissan                3         13
##  8 hyundai               2         14
##  9 subaru                2         14
## 10 honda                 1          9
## 11 jeep                  1          8
## 12 land rover            1          4
## 13 lincoln               1          3
## 14 mercury               1          4
## 15 pontiac               1          5
```

```r
# Dodge has the most models (37 distinct models)

# Model with the most variations
model_variations <- mpg %>%
  group_by(model) %>%
  summarise(
    num_variations = n(),
    manufacturers = paste(unique(manufacturer), collapse = ", "),
    engine_types = n_distinct(displ),
    transmission_types = n_distinct(trans),
    drive_types = n_distinct(drv)
  ) %>%
  arrange(desc(num_variations))

print(head(model_variations, 10))
```

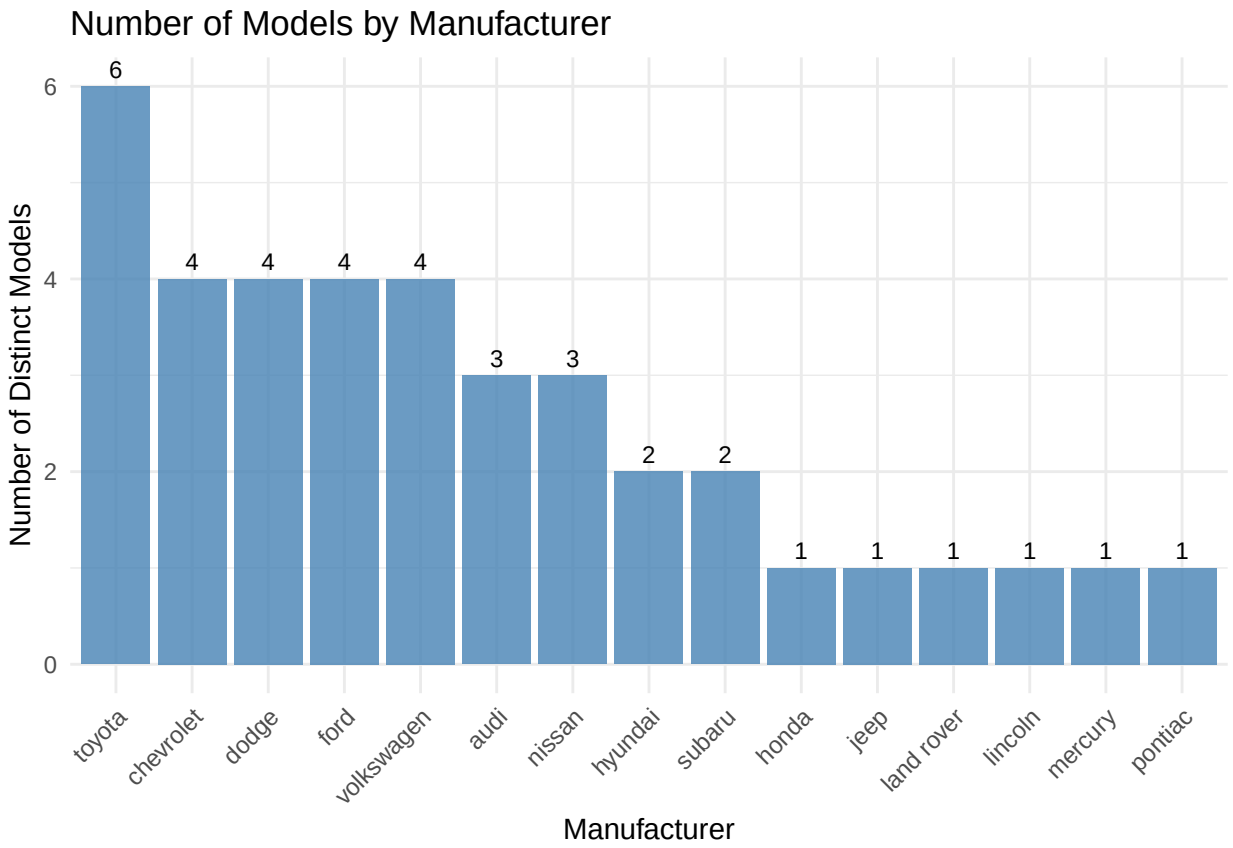
```
## # A tibble: 10 x 6
##    model            num_variations manufacturers engine_types transmission_types
##    <chr>                     <int> <chr>                 <int>              <int>
##  1 caravan 2wd                  11 dodge                     5                  3
##  2 ram 1500 pickup~             10 dodge                     4                  4
##  3 civic                         9 honda                     3                  4
##  4 dakota pickup 4~              9 dodge                     4                  4
```

```
##  5 jetta                          9 volkswagen             4                4
##  6 mustang                        9 ford                   4                4
##  7 a4 quattro                     8 audi                   4                4
##  8 grand cherokee ~               8 jeep                   6                2
##  9 impreza awd                    8 subaru                 2                3
## 10 a4                             7 audi                   4                4
## # i 1 more variable: drive_types <int>
```

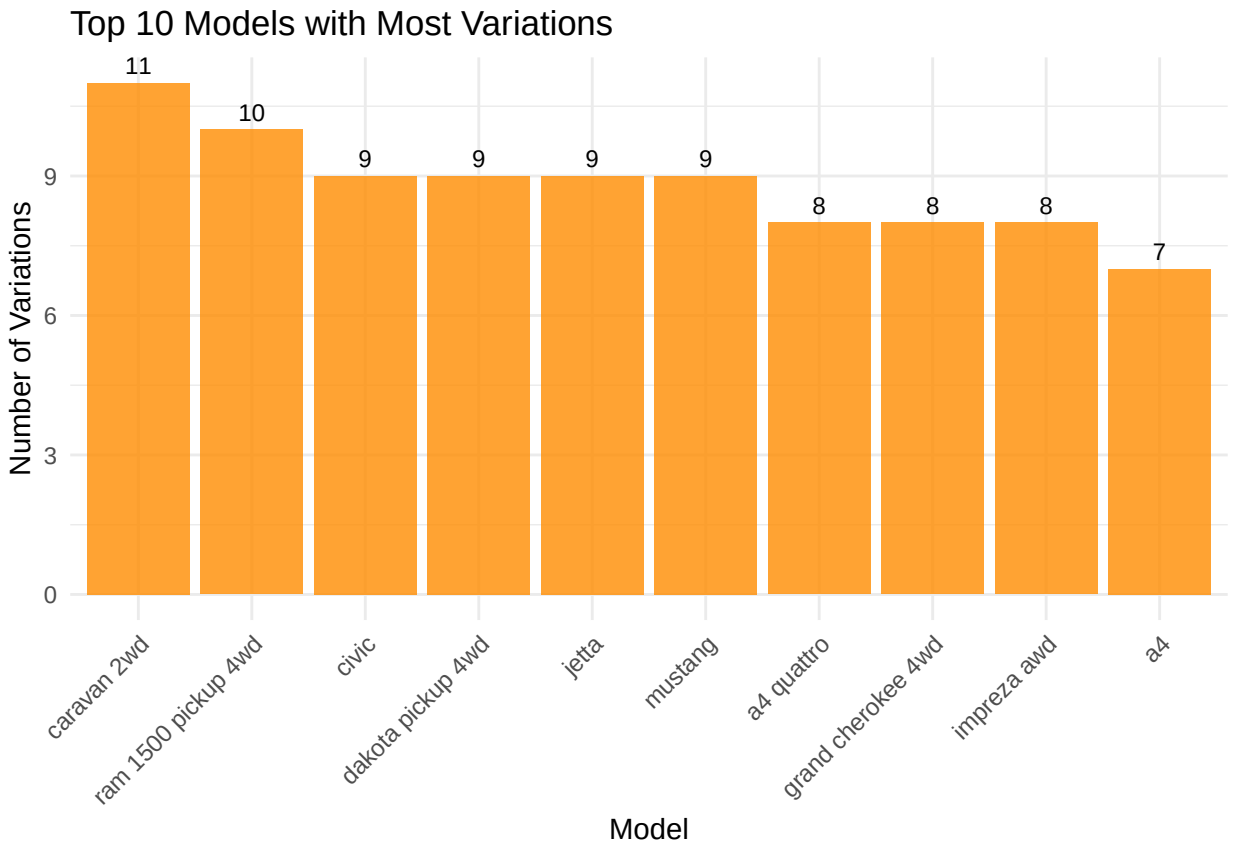
```r
# Caravan 2wd, Mustang, and Ram 1500 Pickup 4wd are tied for the most variations (5 variations each)

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

# Base R plot for manufacturers
par(mar = c(8, 4, 4, 2))  # Adjust margins for labels
barplot(manufacturer_summary$num_models,
        names.arg = manufacturer_summary$manufacturer,
        las = 2,  # Vertical labels
        col = "steelblue",
        main = "Number of Models by Manufacturer",
        ylab = "Number of Distinct Models",
        cex.names = 0.8)
```




```r
# ggplot for manufacturers
ggplot(manufacturer_summary, aes(x = reorder(manufacturer, -num_models), y = num_models)) +
  geom_bar(stat = "identity", fill = "steelblue", alpha = 0.8) +
  geom_text(aes(label = num_models), vjust = -0.5, size = 3) +
  labs(title = "Number of Models by Manufacturer",
       x = "Manufacturer",
       y = "Number of Distinct Models") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



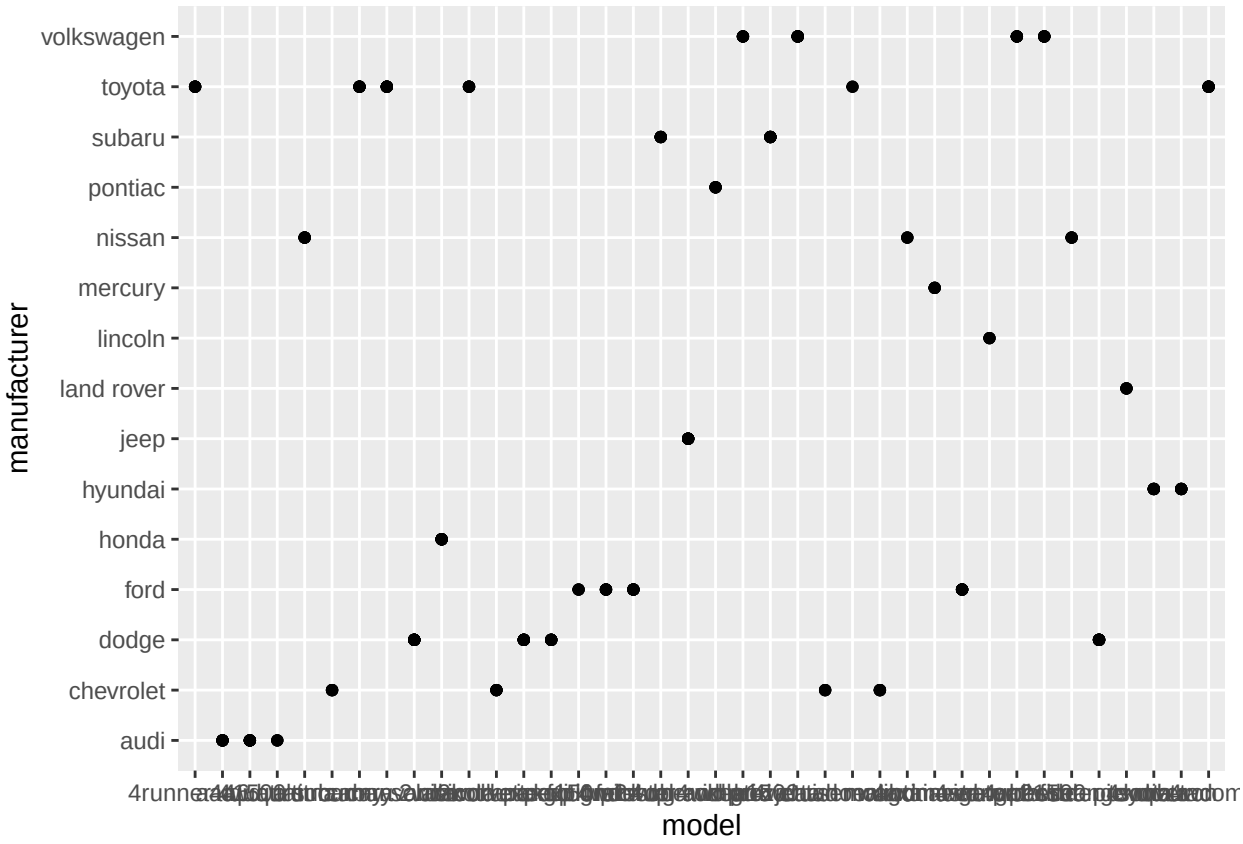

```
# Plot for top models with most variations
top_models <- head(model_variations, 10)

ggplot(top_models, aes(x = reorder(model, -num_variations), y = num_variations)) +
  geom_bar(stat = "identity", fill = "darkorange", alpha = 0.8) +
  geom_text(aes(label = num_variations), vjust = -0.5, size = 3) +
  labs(title = "Top 10 Models with Most Variations",
       x = "Model",
       y = "Number of Variations") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# Manufacturer with most models: Dodge (37 distinct models)
# Models with most variations: Caravan 2wd, Mustang, and Ram 1500 Pickup 4wd (5 variations each)
```

#2. 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?

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

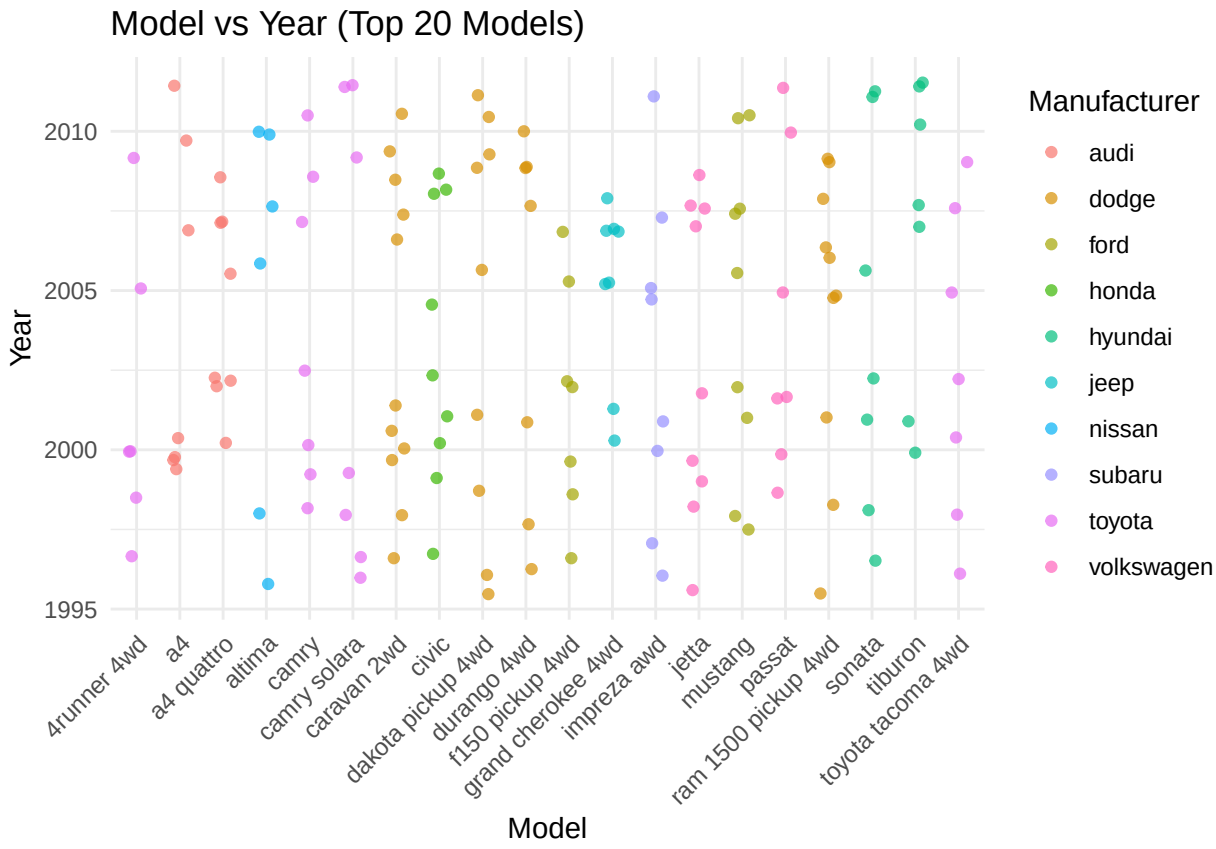



```
# This creates a scatter plot where: x-axis: car models (all 234 individual entries) and y-axis: manufa

# b. For you, is it useful? If not, how could you modify the data to make it more informative? No becau
```

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

```
# Get top 20 models by frequency and plot with year
top_20_models <- mpg %>%
  count(model) %>%
  arrange(desc(n)) %>%
  head(20) %>%
  pull(model)

mpg_top_20 <- mpg %>%
  filter(model %in% top_20_models)

ggplot(mpg_top_20, aes(x = model, y = year)) +
  geom_jitter(aes(color = manufacturer), alpha = 0.7, width = 0.2) +
  labs(title = "Model vs Year (Top 20 Models)",
       x = "Model",
       y = "Year",
       color = "Manufacturer") +
```

```
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



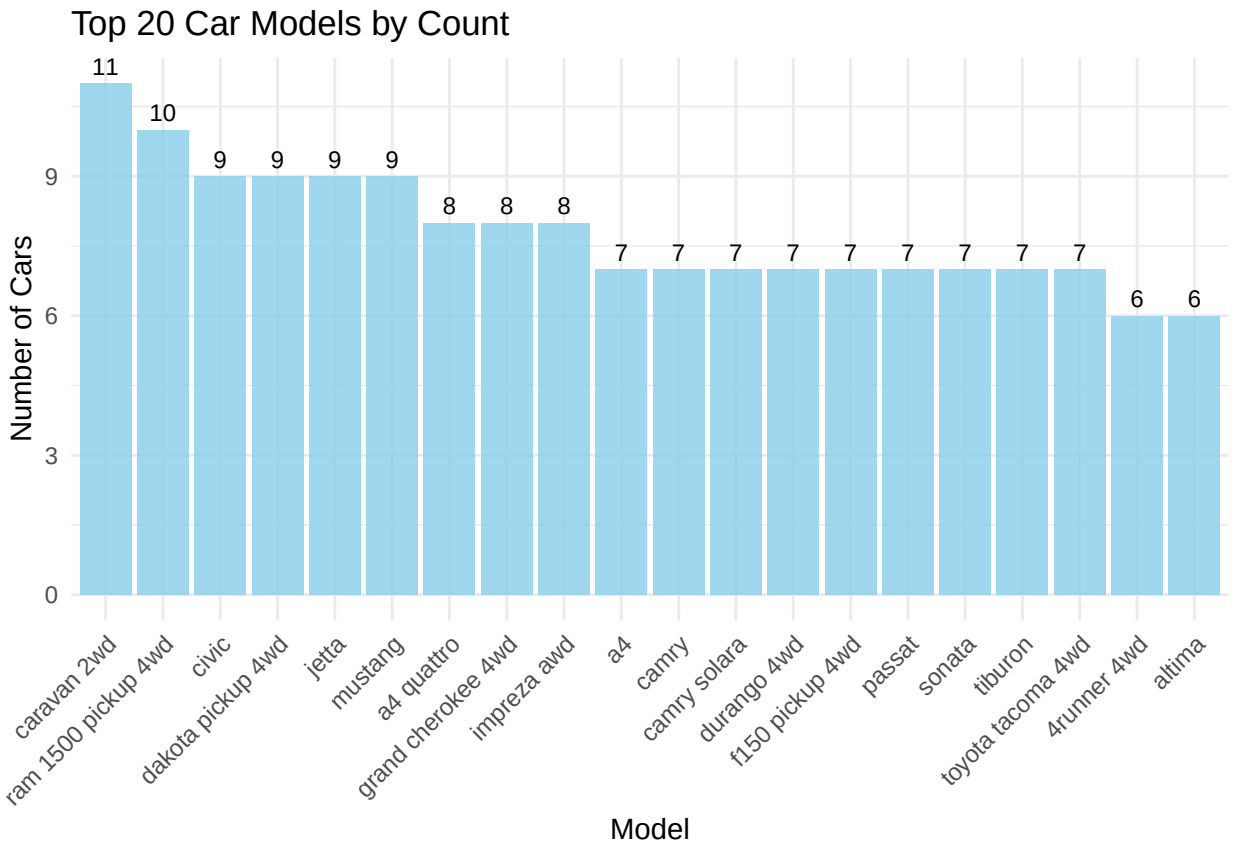
4. Using the pipe (%>%), group the model and get the number of cars per model. Show codes and its result.

```
# Group by model and count cars
model_counts <- mpg %>%
  group_by(model) %>%
  summarise(num_cars = n()) %>%
  arrange(desc(num_cars))

print(model_counts)
```

```
## # A tibble: 38 x 2
##    model               num_cars
##    <chr>                  <int>
##  1 caravan 2wd               11
##  2 ram 1500 pickup 4wd       10
##  3 civic                      9
##  4 dakota pickup 4wd          9
##  5 jetta                      9
```

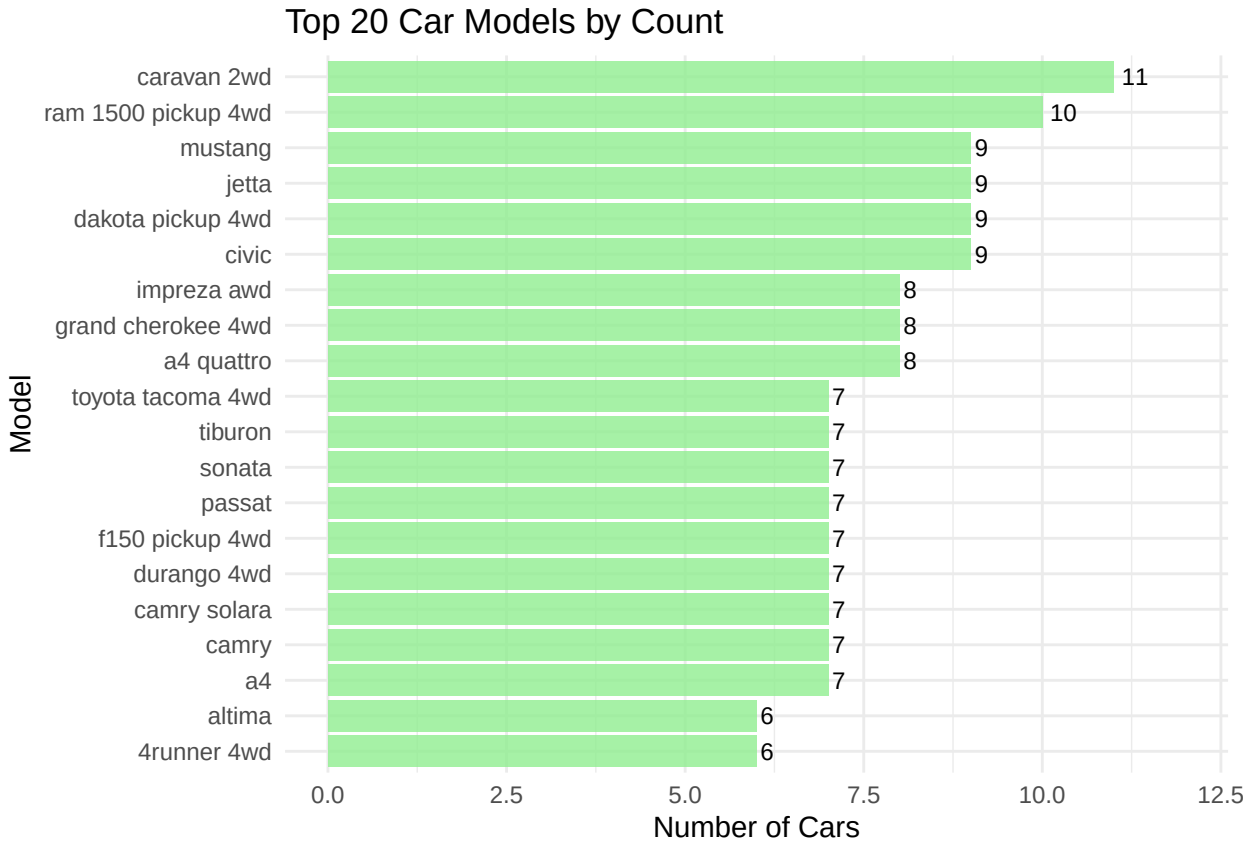
```
##  6 mustang                   9
##  7 a4 quattro                8
##  8 grand cherokee 4wd        8
##  9 impreza awd               8
## 10 a4                        7
## # i 28 more rows
```

```r
# a. Plot using geom_bar() using the top 20 observations only. The graphs shoudl have a title, labels a

model_counts_top20 <- head(model_counts, 20)

ggplot(model_counts_top20, aes(x = reorder(model, -num_cars), y = num_cars)) +
  geom_bar(stat = "identity", fill = "skyblue", alpha = 0.8) +
  geom_text(aes(label = num_cars), vjust = -0.5, size = 3) +
  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))
```




```r
# b. Plot using the geom_bar() + coord_flip() just like what is shown below. Show codes and its result.

ggplot(model_counts_top20, aes(x = reorder(model, num_cars), y = num_cars)) +
  geom_bar(stat = "identity", fill = "lightgreen", alpha = 0.8) +
  geom_text(aes(label = num_cars), hjust = -0.3, size = 3) +
  coord_flip() +
```

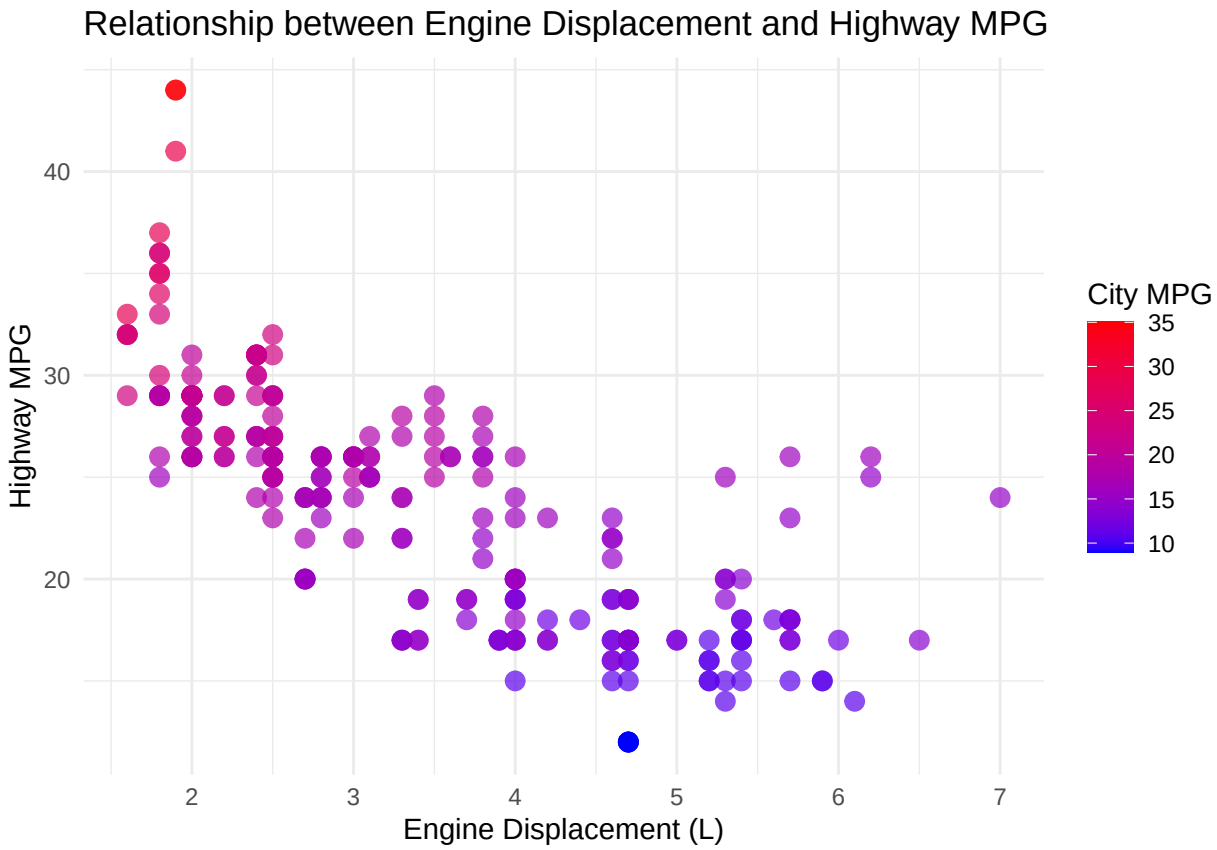
```
  labs(title = "Top 20 Car Models by Count",
       x = "Model",
       y = "Number of Cars") +
  theme_minimal() +
  expand_limits(y = max(model_counts_top20$num_cars) + 1)
```



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?
 library(ggplot2)

ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = cty), size = 3, alpha = 0.7) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(
    title = "Relationship between Engine Displacement and Highway MPG",
    x = "Engine Displacement (L)",
    y = "Highway MPG",
    color = "City MPG"
```

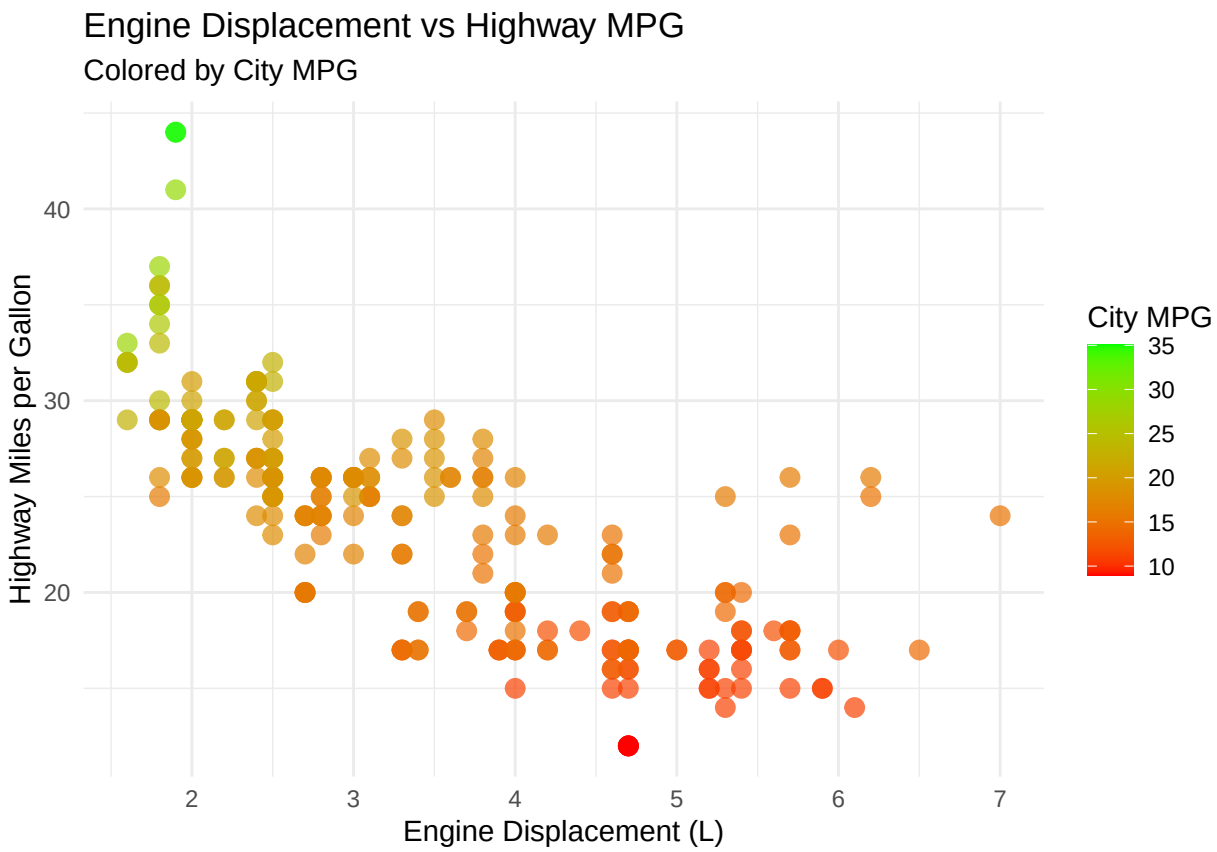
```
  ) +
  theme_minimal()
```




```
# The plot shows that as engine displacement increases, highway MPG decreases. Cars with higher city MP

# Why: Larger engines burn more fuel, resulting in lower efficiency. Since city MPG is also a measure o
```

6. 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?

```
# Using a continuous variable from #1-c (cty - city MPG)
library(ggplot2)
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = cty), size = 3, alpha = 0.7) +
  scale_color_gradient(low = "red", high = "green", name = "City MPG") +
  labs(title = "Engine Displacement vs Highway MPG",
       subtitle = "Colored by City MPG",
       x = "Engine Displacement (L)",
       y = "Highway Miles per Gallon") +
  theme_minimal()
```




```
#The plot shows a negative correlation between engine displacement and highway MPG, with color revealin

# Smaller engines (low displacement) have higher MPG (green points)
# Larger engines (high displacement) have lower MPG (red points)
# City MPG pattern: The color gradient shows that city MPG follows the same trend as highway MPG - both

# Why this output:
# Larger engines consume more fuel, both city and highway MPG are affected similarly by engine size and
```

6. Import the traffic.csv file

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

```
##   time junction cars
## 1    1        A   20
## 2    2        A   25
## 3    3        A   30
## 4    1        B   10
## 5    2        B   12
## 6    3        B   15
## 7    1        C   30
## 8    2        C   28
## 9    3        C   26
```

```r
dim(traffic)      # number of rows and columns
```

```
## [1] 9 3
```

```r
names(traffic)    # variable names
```

```
## [1] "time"     "junction" "cars"
```

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

traffic_A <- subset(traffic, junction == "A")
traffic_B <- subset(traffic, junction == "B")
traffic_C <- subset(traffic, junction == "C")

traffic_A
```

```
##   time junction cars
## 1    1        A   20
## 2    2        A   25
## 3    3        A   30
```

```r
traffic_B
```

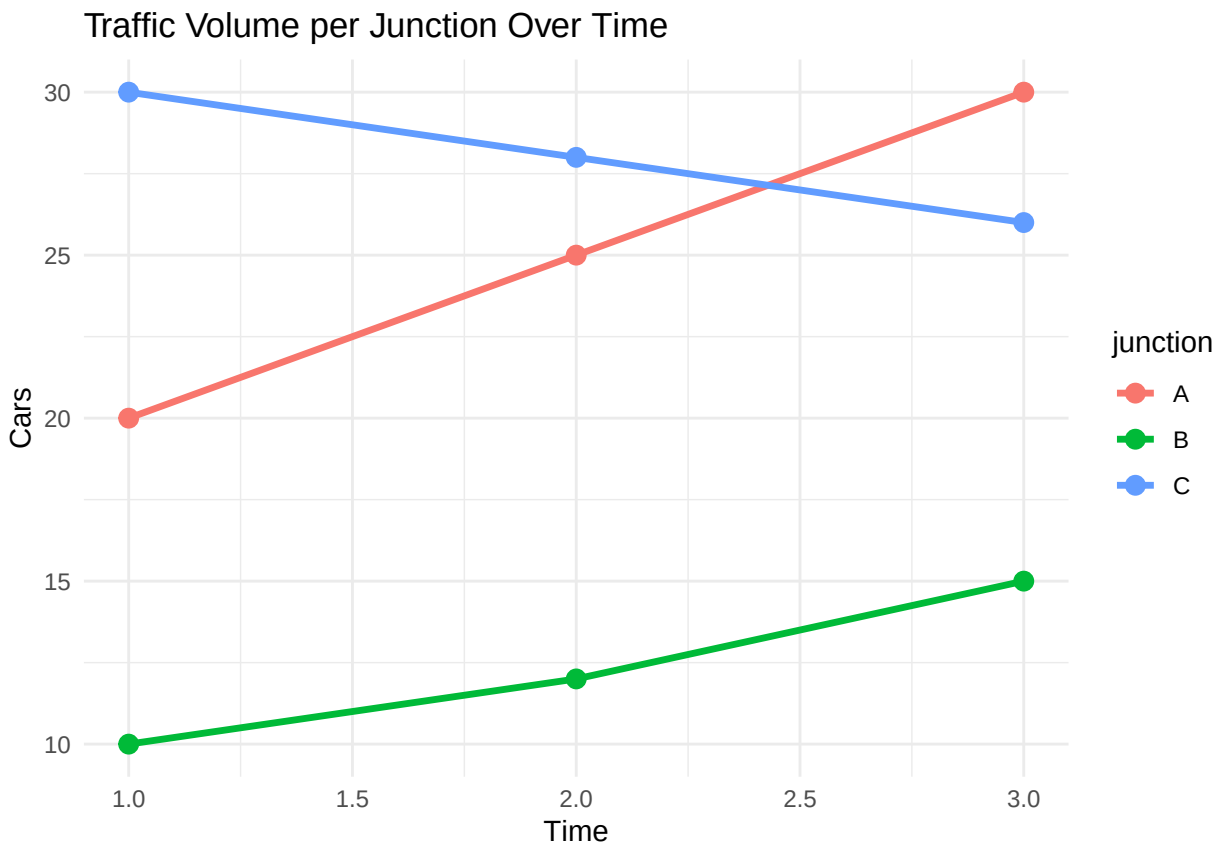
```
##   time junction cars
## 4    1        B   10
## 5    2        B   12
## 6    3        B   15
```

```r
traffic_C
```

```
##   time junction cars
## 7    1        C   30
## 8    2        C   28
## 9    3        C   26
```

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

library(ggplot2)

ggplot(traffic, aes(x = time, y = cars, color = junction)) +
  geom_line(size = 1.2) +
  geom_point(size = 3) +
  labs(
    title = "Traffic Volume per Junction Over Time",
    x = "Time",
    y = "Cars"
  ) +
  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)

# Import file
alexa <- read_excel("alexa_file.xlsx")

# a. How many observations does alexa_file has? What about the number of columns? Show your solution an

# Number of observations (rows)
nrow(alexa)
```

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

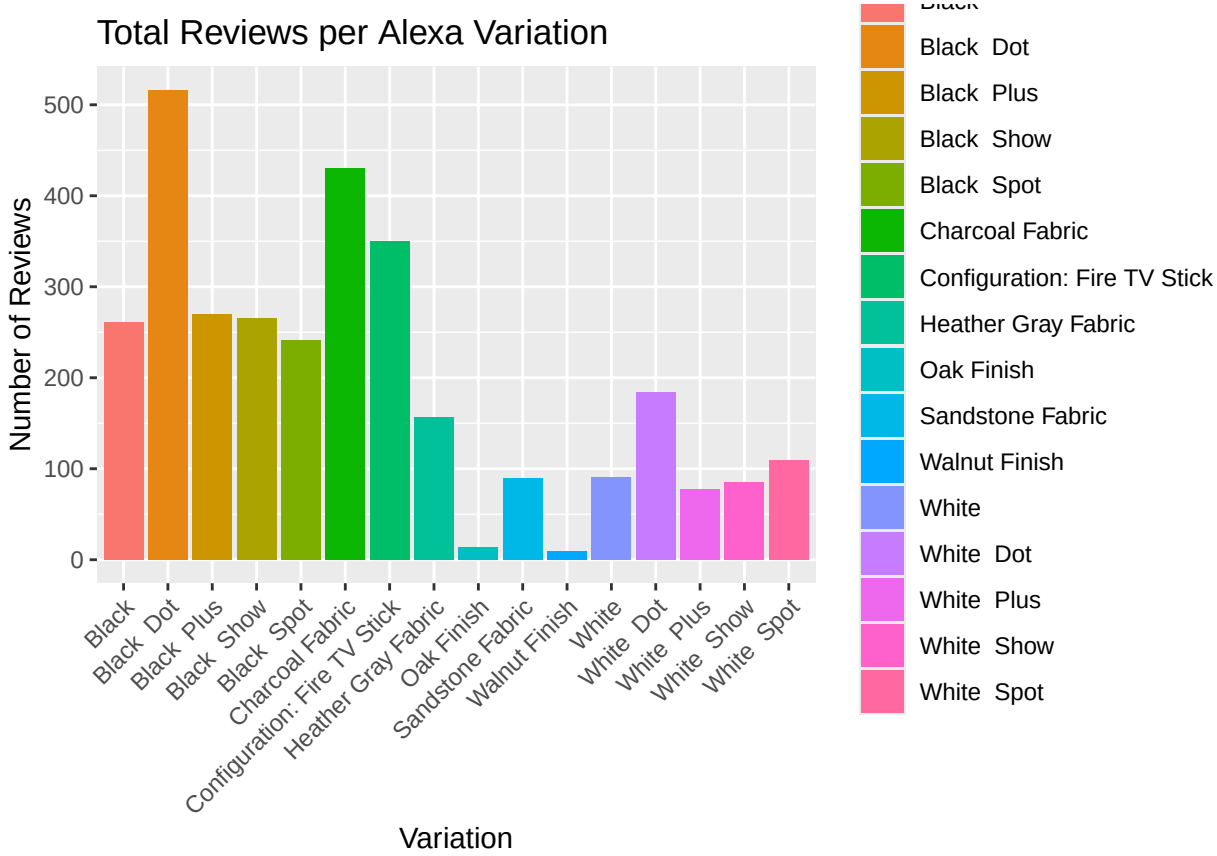
```
# Number of columns
ncol(alexa)
```

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

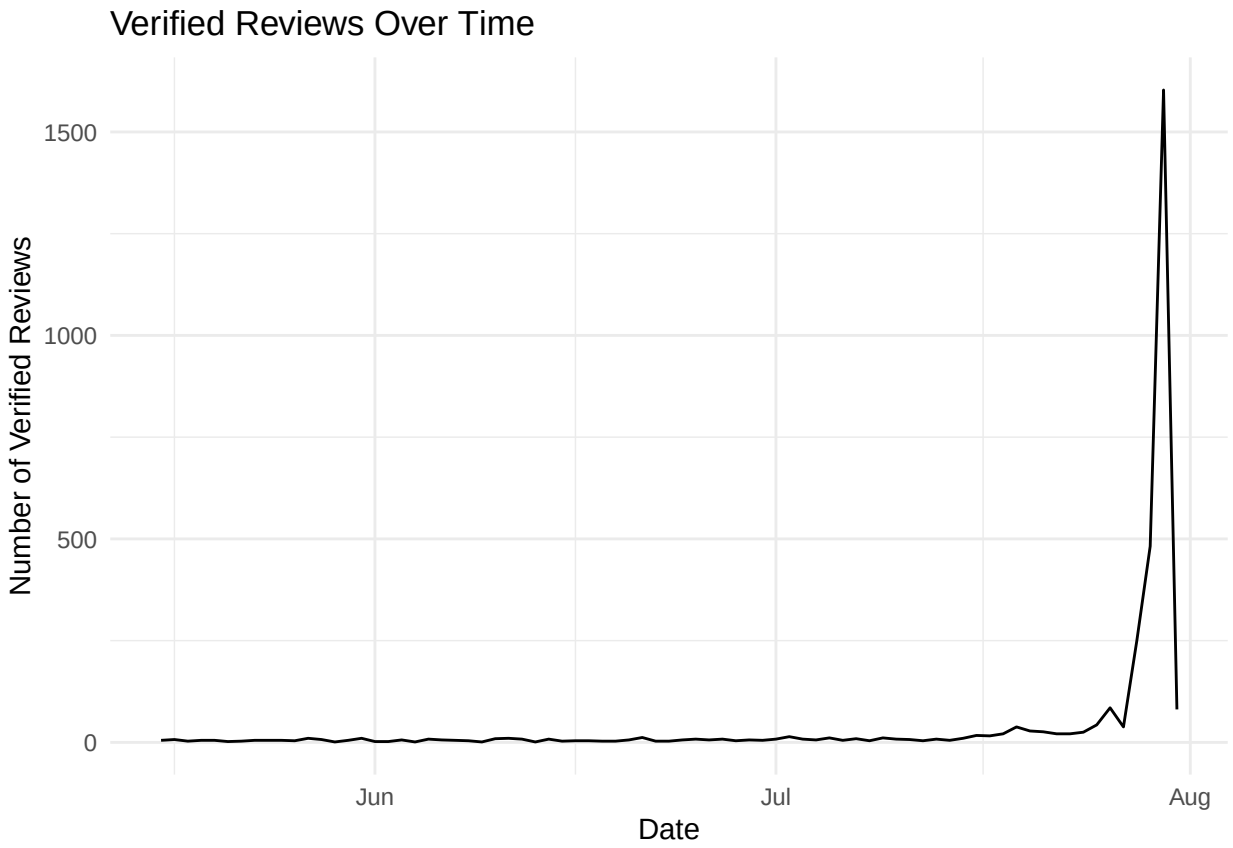
```r
# b. group the variations and get the total of each variations. Use dplyr package. Show solution and an

library(dplyr)

variation_total <- alexa %>%
  group_by(variation) %>%
  summarise(total = n())

variation_total
```

```
## # 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
```

```r
# c. Plot the variations using the ggplot() function. What did you observe? Complete the details of the

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




```r
# d. Plot a geom_line() with the date and the number of verified reviews. Complete the details of the g
date_counts <- alexa %>%
  group_by(date) %>%
  summarise(total_reviews = n())

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




```
# e. Get the relationship of variations and ratings. Which variations got the most highest in rating? P

rating_variation <- alexa %>%
  group_by(variation) %>%
  summarise(avg_rating = mean(rating))

rating_variation
```

```
## # 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
```

```r
ggplot(rating_variation, aes(x = variation, y = avg_rating, fill = variation)) +
  geom_col() +
  labs(
    title = "Average Rating per Alexa Variation",
    x = "Variation",
    y = "Average Rating"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```