RWorksheet_Moquete#4c.Rmd

Renz Moquete

2025-12-12

1. Use the dataset mpg

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

Since mpg is already available in ggplot2, we'll load it directly

```
data(mpg)

# To demonstrate importing a CSV file (if we had one):
# mpg_csv <- read_csv("mpg.csv")  # For CSV file
# mpg_excel <- read_excel("mpg.xlsx")  # For Excel file

cat("First 6 rows of mpg dataset:\n")
```

```
## First 6 rows of mpg dataset:
```

```
print(head(mpg))
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
## 3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
## 4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
```

b. Which variables from mpg dataset are categorical?

```
cat("\nCategorical variables in mpg dataset:\n")
```

```
##
## Categorical variables in mpg dataset:
```

```
categorical_vars <- names(mpg)[sapply(mpg, is.character)]
print(categorical_vars)
```

```
## [1] "manufacturer" "model"        "trans"        "drv"          "fl"
## [6] "class"
```

```
# Also check factor variables
cat("\nFactor variables in mpg dataset:\n")
```

```
##
## Factor variables in mpg dataset:
```

```r
factor_vars <- names(mpg)[sapply(mpg, is.factor)]
print(factor_vars)
```

```
## character(0)
```

c. Which are continuous variables?

```r
cat("\nContinuous/numeric variables in mpg dataset:\n")
```

```
##
## Continuous/numeric variables in mpg dataset:
```

```r
numeric_vars <- names(mpg)[sapply(mpg, is.numeric)]
print(numeric_vars)
```

```
## [1] "displ" "year"  "cyl"   "cty"   "hwy"
```

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

a. Group the manufacturers and find the unique models

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

cat("Manufacturer with most models:\n")
```

```
## Manufacturer with most models:
```

```r
print(manufacturer_summary[1, ])
```

```
## # A tibble: 1 x 3
##   manufacturer num_models total_cars
##   <chr>             <int>      <int>
## 1 toyota                6         34
```

```r
# Model with most variations
model_summary <- mpg %>%
  group_by(model) %>%
  summarise(
    num_variations = n(),
    manufacturers = paste(unique(manufacturer), collapse = ", ")
  ) %>%
  arrange(desc(num_variations))

cat("\nModel with most variations:\n")
```

```
##
## Model with most variations:
```

```r
print(model_summary[1, ])
```

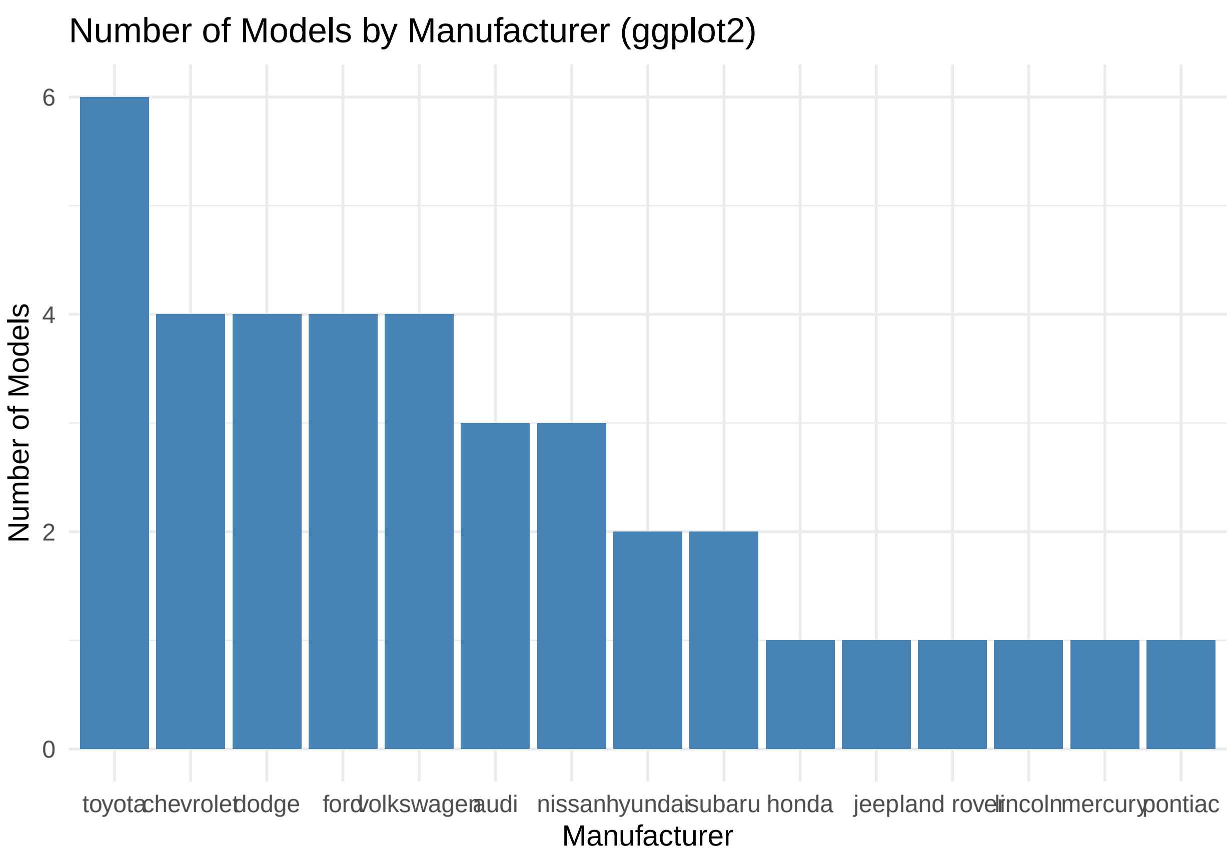
```
## # A tibble: 1 x 3
```

```
##   model        num_variations manufacturers
##   <chr>                 <int> <chr>
## 1 caravan 2wd              11 dodge
```

b. Graph the result by using plot() and ggplot()

```r
# Base R plot
par(mfrow = c(1, 2))
barplot(manufacturer_summary$num_models,
        names.arg = manufacturer_summary$manufacturer,
        main = "Number of Models by Manufacturer (Base R)",
        xlab = "Manufacturer",
        ylab = "Number of Models",
        col = "lightblue",
        las = 2,
        cex.names = 0.7)

# ggplot2
ggplot(manufacturer_summary, aes(x = reorder(manufacturer, -num_models), y = num_models)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Number of Models by Manufacturer (ggplot2)",
       x = "Manufacturer",
       y = "Number of Models") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme_minimal()
par(mfrow = c(1, 1))
```



c. Same dataset will be used. Show relationship of model and manufacturer

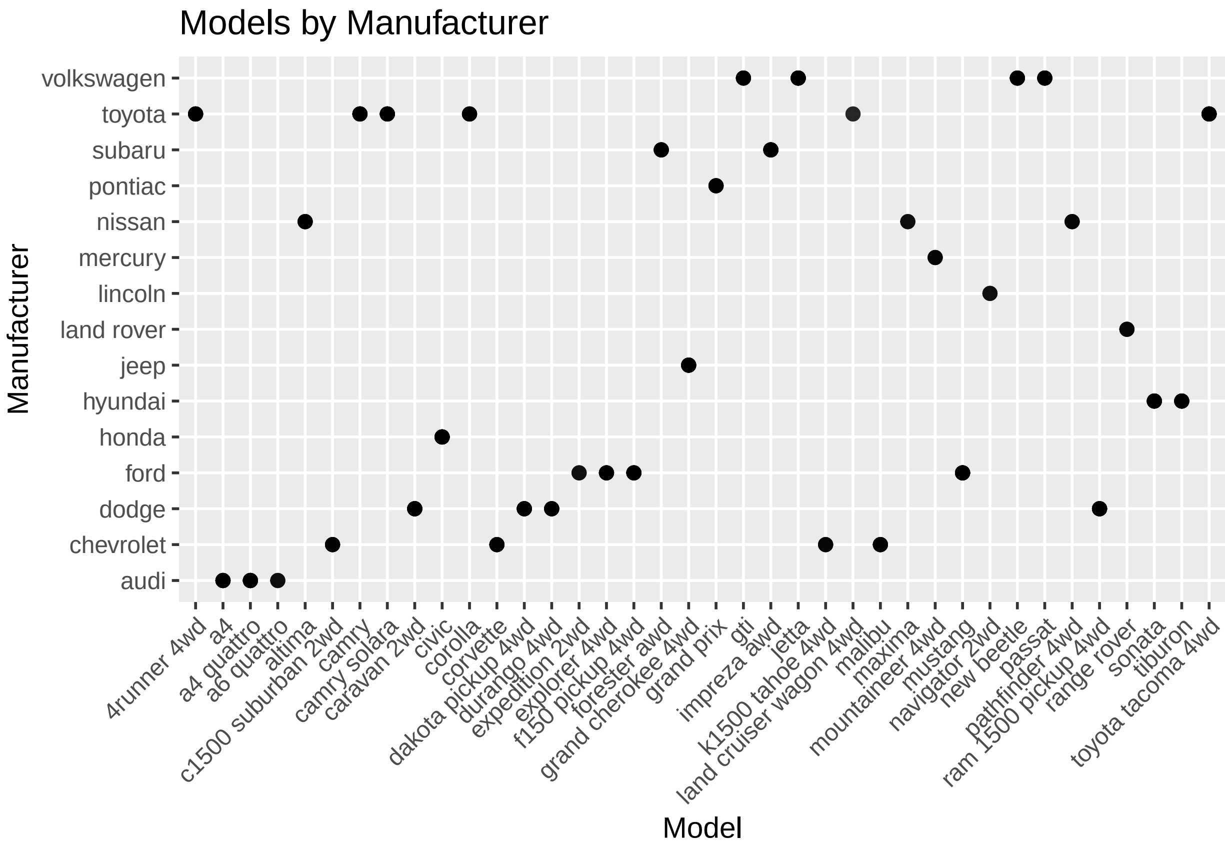
```r
# a. What does ggplot(mpg, aes(model, manufacturer)) + geom_point() show?
cat("This plot shows each car model as a point, positioned by model (x-axis) and manufacturer (y-axis).
```

```
## This plot shows each car model as a point, positioned by model (x-axis) and manufacturer (y-axis).
```

```r
cat("It visualizes which manufacturers produce which models.\n\n")
```

```
## It visualizes which manufacturers produce which models.
```

```r
# Create the plot
p1 <- ggplot(mpg, aes(x = model, y = manufacturer)) +
  geom_point(alpha = 0.6, size = 2) +
  labs(title = "Models by Manufacturer",
       x = "Model",
       y = "Manufacturer") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
print(p1)
```




```r
# b. For you, is it useful? If not, how could you modify the data to make it more informative?
cat("\nb. This plot could be improved by:\n")
```

```
##
## b. This plot could be improved by:
```

```r
cat("1. Adding color to show year or other variables\n")
```

```
## 1. Adding color to show year or other variables
```

```r
cat("2. Adding jitter to reduce overplotting\n")
```

```
## 2. Adding jitter to reduce overplotting
```

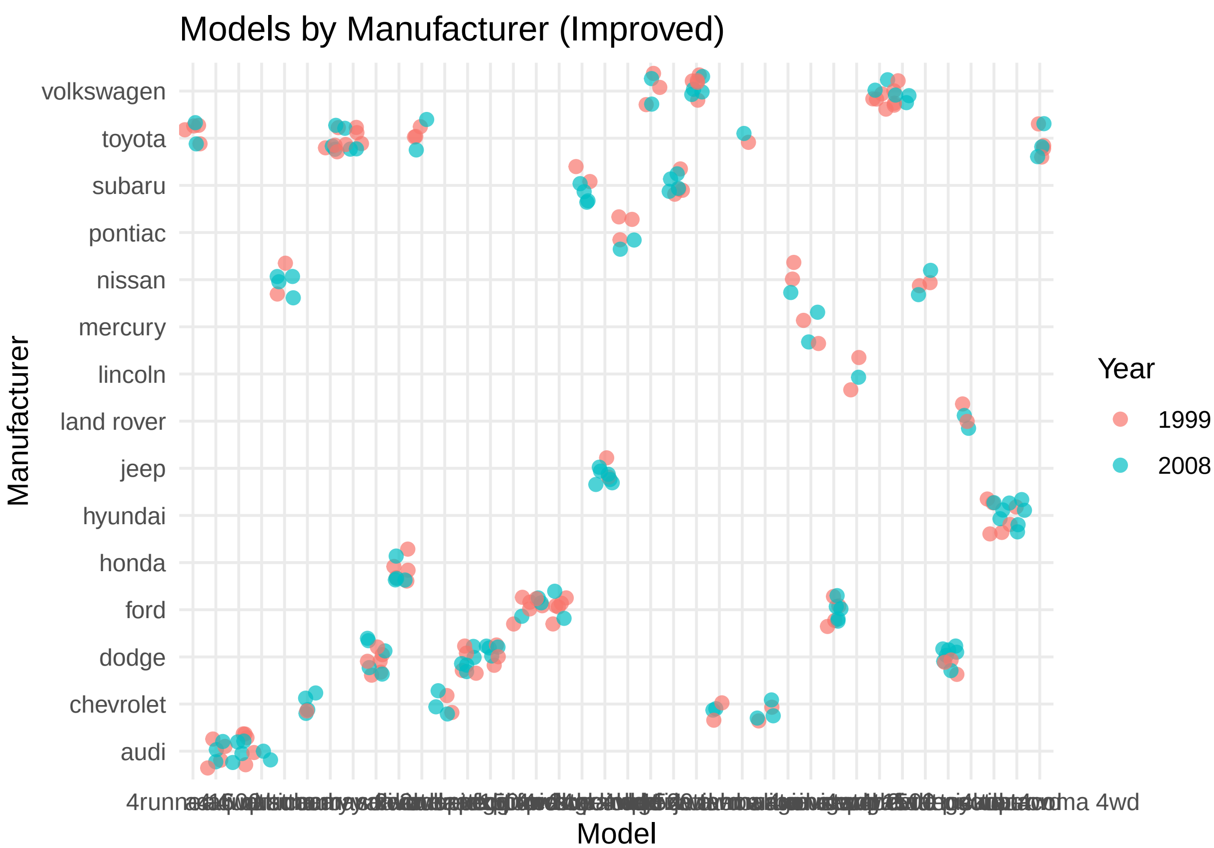
```r
cat("3. Reordering or aggregating data\n")
```

```
## 3. Reordering or aggregating data
```

```r
cat("4. Using geom_count() to show frequency\n")
```

```
## 4. Using geom_count() to show frequency
```

```r
# Improved version
p2 <- ggplot(mpg, aes(x = model, y = manufacturer, color = as.factor(year))) +
  geom_jitter(alpha = 0.7, size = 2) +
  labs(title = "Models by Manufacturer (Improved)",
       x = "Model",
       y = "Manufacturer",
       color = "Year") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme_minimal()
print(p2)
```



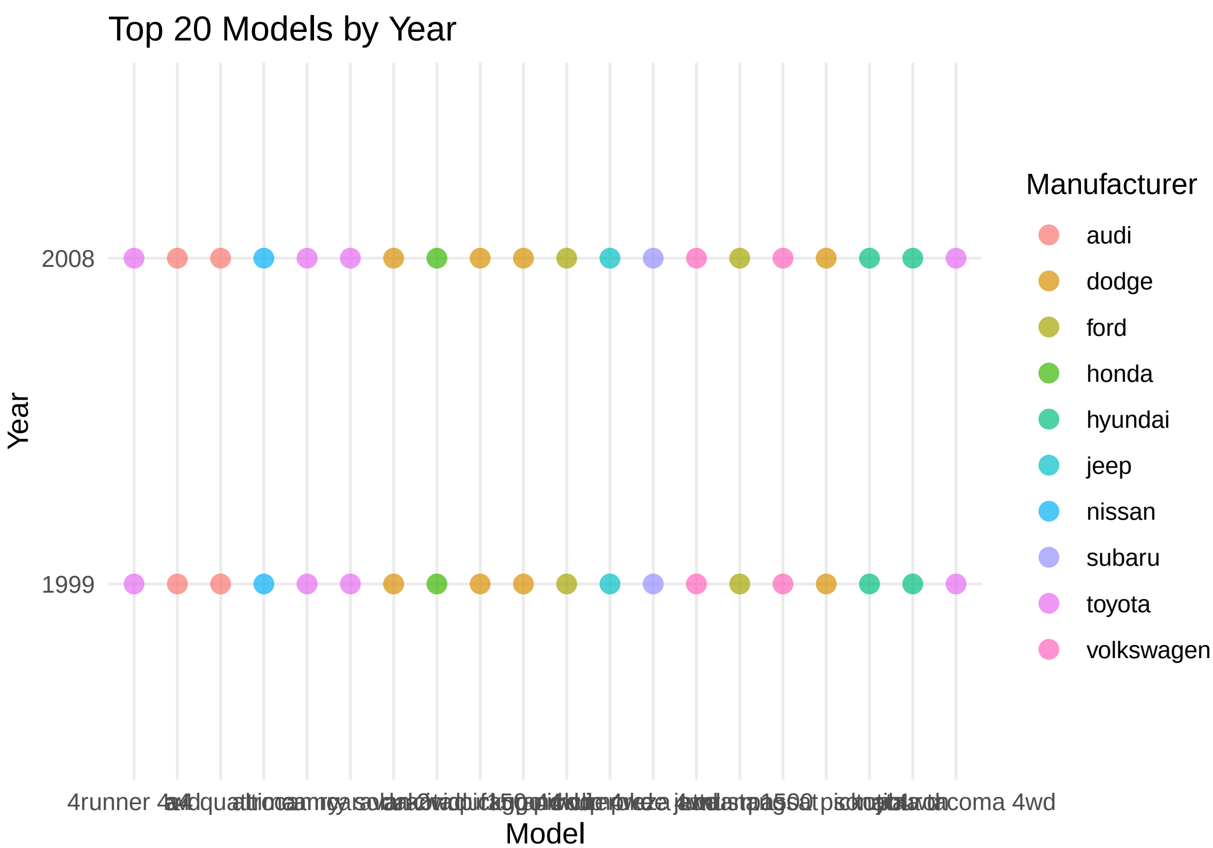
c. Plot the model and the year using ggplot(). Use only the top 20 observations.

```r
top_20_models <- mpg %>%
  group_by(model) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  head(20) %>%
  pull(model)
```

```r
mpg_top20 <- mpg %>%
  filter(model %in% top_20_models)

# Count unique combinations instead of using distinct
mpg_top20_summary <- mpg_top20 %>%
  group_by(model, year, manufacturer) %>%
  summarise(count = n(), .groups = 'drop')

p3 <- ggplot(mpg_top20_summary, aes(x = model, y = as.character(year))) +
  geom_point(aes(color = manufacturer), size = 3, alpha = 0.7) +
  labs(title = "Top 20 Models by Year",
       x = "Model",
       y = "Year",
       color = "Manufacturer") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme_minimal()
print(p3)
```



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

```r
model_counts <- mpg %>%
  group_by(model) %>%
  summarise(num_cars = n()) %>%
  arrange(desc(num_cars))

cat("Number of cars per model (top 10):\n")
```

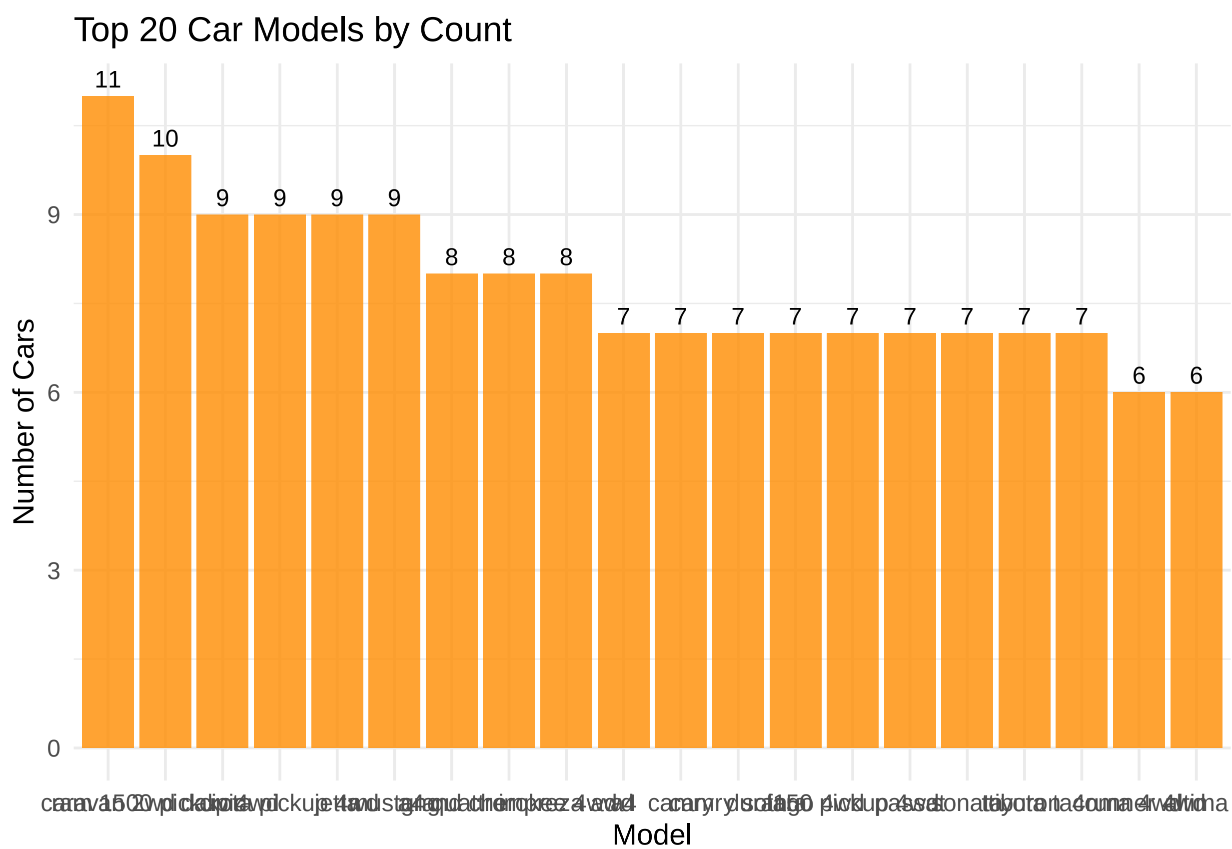
```
## Number of cars per model (top 10):
```

```r
print(head(model_counts, 10))
```

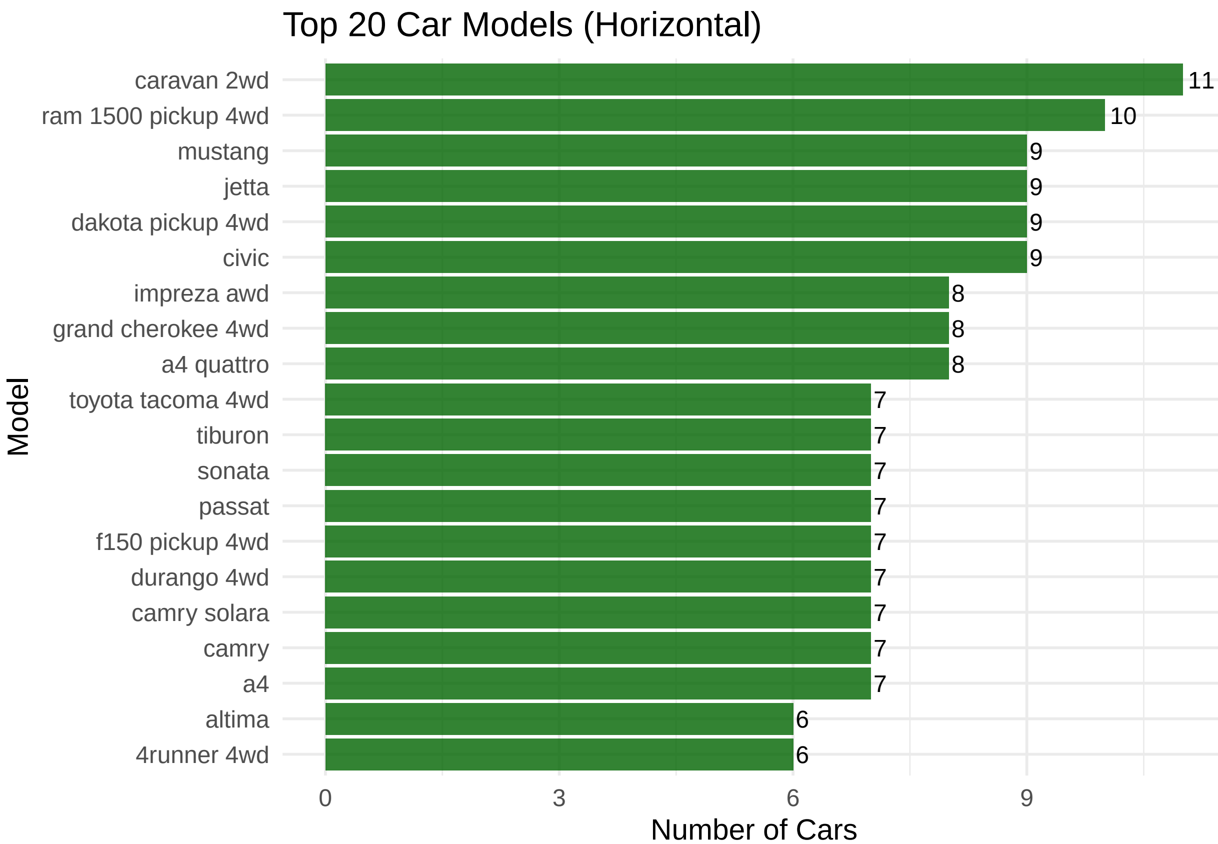
```
## # A tibble: 10 x 2
##    model              num_cars
##    <chr>                 <int>
##  1 caravan 2wd              11
##  2 ram 1500 pickup 4wd      10
##  3 civic                     9
##  4 dakota pickup 4wd         9
##  5 jetta                     9
##  6 mustang                   9
##  7 a4 quattro                8
##  8 grand cherokee 4wd        8
##  9 impreza awd               8
## 10 a4                        7
```

```r
# Plot using geom_bar() using the top 20 observations only.
top_20_counts <- head(model_counts, 20)

p4 <- ggplot(top_20_counts, aes(x = reorder(model, -num_cars), y = num_cars)) +
  geom_bar(stat = "identity", fill = "darkorange", alpha = 0.8) +
  labs(title = "Top 20 Car Models by Count",
       x = "Model",
       y = "Number of Cars") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme_minimal() +
  geom_text(aes(label = num_cars), vjust = -0.5, size = 3)
print(p4)
```

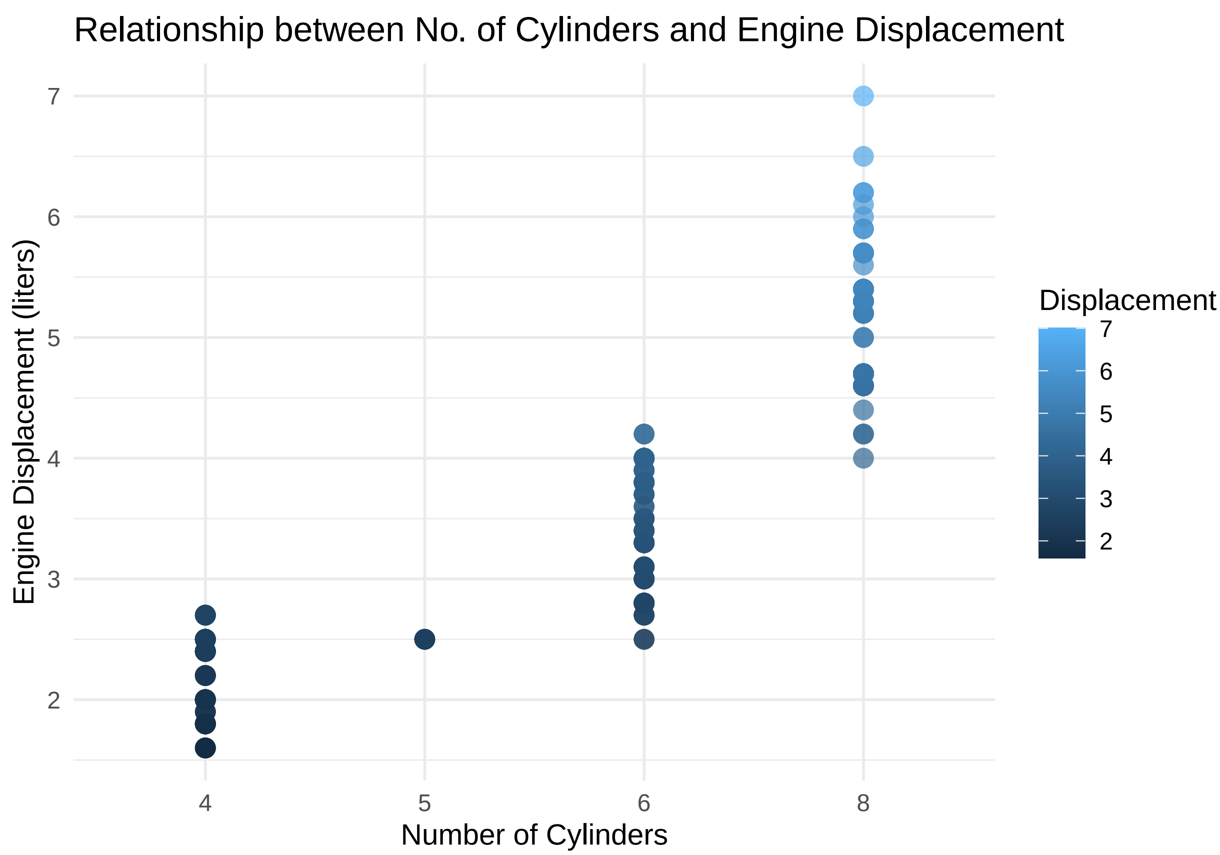



```r
# Plot using geom_bar() + coord_flip()
p5 <- ggplot(top_20_counts, aes(x = reorder(model, num_cars), y = num_cars)) +
  geom_bar(stat = "identity", fill = "darkgreen", alpha = 0.8) +
  coord_flip() +
  labs(title = "Top 20 Car Models (Horizontal)",
       x = "Model",
       y = "Number of Cars") +
  theme_minimal() +
  geom_text(aes(label = num_cars), hjust = -0.2, size = 3)
print(p5)
```



3. Plot the relationship between cyl and displ

```
p6 <- ggplot(mpg, aes(x = as.factor(cyl), y = displ, color = displ)) +
  geom_point(size = 3, alpha = 0.7) +
  labs(title = "Relationship between No. of Cylinders and Engine Displacement",
       x = "Number of Cylinders",
       y = "Engine Displacement (liters)",
       color = "Displacement") +
  theme_minimal()
print(p6)
```




```
cat("\nRelationship description: Generally, as the number of cylinders increases,\n")
```

```
##
## Relationship description: Generally, as the number of cylinders increases,
```

```
cat("the engine displacement also tends to increase. However, there is some overlap\n")
```

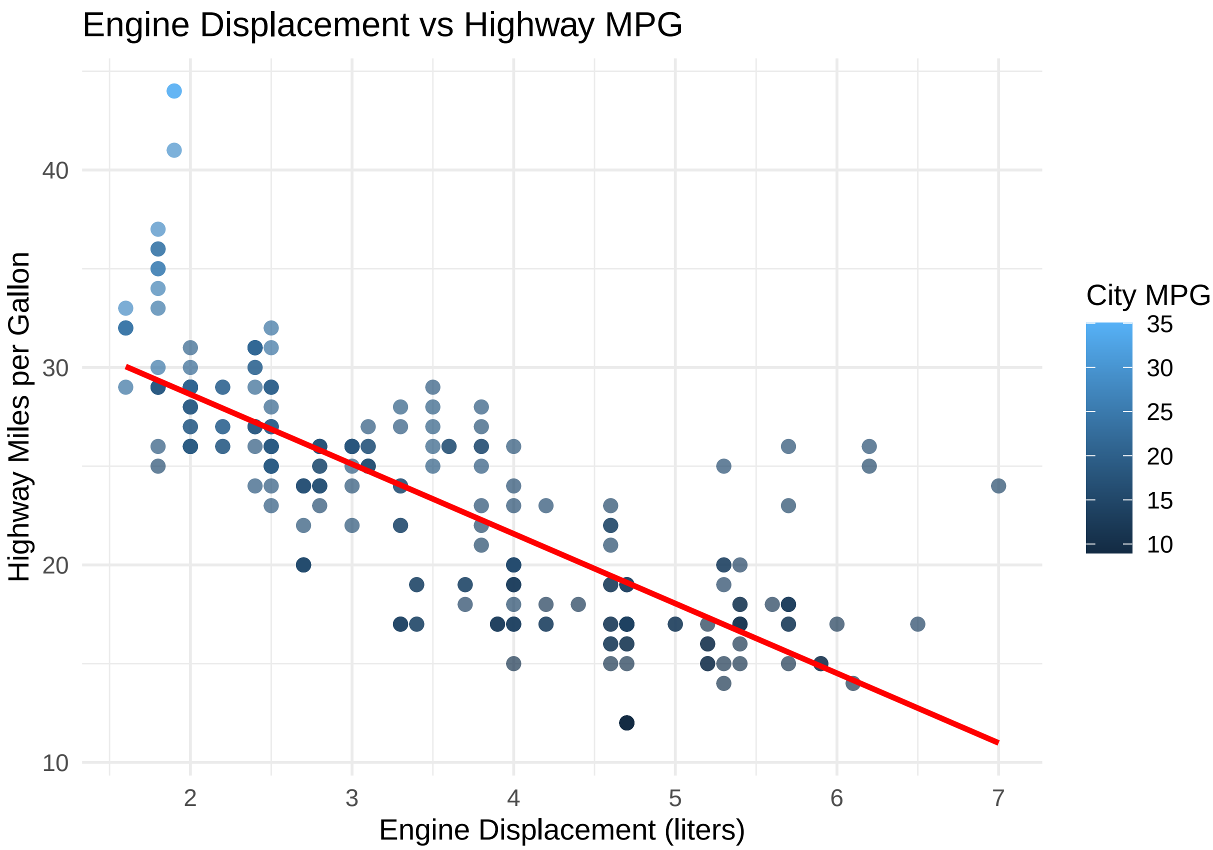
```
## the engine displacement also tends to increase. However, there is some overlap
```

```
cat("and variation, especially in the 4, 6, and 8 cylinder categories.\n")
```

```
## and variation, especially in the 4, 6, and 8 cylinder categories.
```

4. Plot the relationship between displ and hwy

```
p7 <- ggplot(mpg, aes(x = displ, y = hwy, color = cty)) +  # Using cty as continuous variable
  geom_point(size = 2, alpha = 0.7) +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Engine Displacement vs Highway MPG",
       x = "Engine Displacement (liters)",
       y = "Highway Miles per Gallon",
       color = "City MPG") +
  theme_minimal()
print(p7)
```




```
cat("\nResult: There is a negative relationship between engine displacement and highway MPG.\n")
```

```
##
## Result: There is a negative relationship between engine displacement and highway MPG.
```

```
cat("Larger engines (higher displacement) tend to have lower fuel efficiency (lower hwy mpg).\n")
```

```
## Larger engines (higher displacement) tend to have lower fuel efficiency (lower hwy mpg).
```

```
cat("The color gradient (city MPG) shows a similar pattern - cars with better city MPG\n")
```

```
## The color gradient (city MPG) shows a similar pattern - cars with better city MPG
```

```
cat("also tend to have better highway MPG.\n")
```

```
## also tend to have better highway MPG.
```

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

```
set.seed(123)
n_days <- 5  # Use 5 days instead of 7
n_hours <- n_days * 24

traffic <- data.frame(
  DateTime = seq(from = as.POSIXct("2024-01-01"),
                 length.out = n_hours,
                 by = "hour"),
  Junction1 = rpois(n_hours, lambda = 50) + rnorm(n_hours, 0, 10),
  Junction2 = rpois(n_hours, lambda = 60) + rnorm(n_hours, 0, 15),
```

```r
  Junction3 = rpois(n_hours, lambda = 40) + rnorm(n_hours, 0, 8),
  Junction4 = rpois(n_hours, lambda = 70) + rnorm(n_hours, 0, 12)
)

cat("Traffic dataset created with", nrow(traffic), "observations\n")
```

```
## Traffic dataset created with 120 observations
```

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

```r
cat("Number of observations in traffic dataset:", nrow(traffic), "\n")
```

```
## Number of observations in traffic dataset: 120
```

```r
cat("Variables in traffic dataset:\n")
```

```
## Variables in traffic dataset:
```

```r
print(names(traffic))
```

```
## [1] "DateTime"  "Junction1" "Junction2" "Junction3" "Junction4"
```

b. Subset the traffic dataset into junctions.

```r
junctions_data <- traffic %>%
  select(DateTime, starts_with("Junction"))

cat("First 6 rows of junctions data:\n")
```

```
## First 6 rows of junctions data:
```

```r
print(head(junctions_data))
```

```
##              DateTime Junction1 Junction2 Junction3 Junction4
## 1 2024-01-01 00:00:00  30.27856 67.383429  22.30455  77.42595
## 2 2024-01-01 01:00:00  42.85332 74.017525  20.29518  87.41759
## 3 2024-01-01 02:00:00  21.98464 86.798865  42.50956  69.58904
## 4 2024-01-01 03:00:00  44.69093 69.159370  40.16202  70.68030
## 5 2024-01-01 04:00:00  47.38244 51.794852  44.77553  51.76900
## 6 2024-01-01 05:00:00  59.87917  6.352766  39.00202  53.86500
```

c. Plot each junction using geom_line()

```r
# Reshape data for ggplot
traffic_long <- junctions_data %>%
  pivot_longer(cols = -DateTime,
               names_to = "Junction",
               values_to = "Traffic_Volume")

p8 <- ggplot(traffic_long, aes(x = DateTime, y = Traffic_Volume, color = Junction)) +
  geom_line(linewidth = 1) +
  labs(title = "Traffic Volume by Junction",
       x = "Date and Time",
       y = "Traffic Volume",
       color = "Junction") +
  theme_minimal() +
  facet_wrap(~Junction, ncol = 2, scales = "free_y")
print(p8)
```

7. From alexa_file.xlsx, import it to your environment

```
# Create sample data for demonstration
set.seed(123)
alexa <- data.frame(
  Variations = sample(c("Black Dot", "Black Plus", "Black Show", "Black Spot",
                        "White Dot", "White Plus", "White Show", "White Spot"),
                      200, replace = TRUE),
  Ratings = sample(1:5, 200, replace = TRUE, prob = c(0.05, 0.1, 0.15, 0.3, 0.4)),
  Verified_Reviews = sample(10:500, 200, replace = TRUE),
  Date = sample(seq(as.Date('2024-01-01'), as.Date('2024-06-30'), by="day"), 200, replace = TRUE)
)

cat("Sample alexa data created for demonstration.\n")
```

```
## Sample alexa data created for demonstration.
```

```
cat("In your actual work, use: alexa <- read_excel('alexa_file.xlsx')\n\n")
```

```
## In your actual work, use: alexa <- read_excel('alexa_file.xlsx')
```

a. How many observations does alexa_file has? What about the number of columns?

```
cat("Number of observations in alexa dataset:", nrow(alexa), "\n")
```

```
## Number of observations in alexa dataset: 200
```

```
cat("Number of columns in alexa dataset:", ncol(alexa), "\n")
```

```
## Number of columns in alexa dataset: 4
```

```r
cat("\nDataset structure:\n")
```

```
##
## Dataset structure:
```

```r
str(alexa)
```

```
## 'data.frame':    200 obs. of  4 variables:
##  $ Variations      : chr  "White Show" "White Show" "Black Show" "White Plus" ...
##  $ Ratings         : int  5 1 4 4 4 2 5 5 5 5 ...
##  $ Verified_Reviews: int  120 288 460 402 326 304 231 296 231 82 ...
##  $ Date            : Date, format: "2024-04-24" "2024-06-06" ...
```

b. Group the variations and get the total of each variations.

```r
variation_totals <- alexa %>%
  group_by(Variations) %>%
  summarise(
    Count = n(),
    Avg_Rating = mean(Ratings, na.rm = TRUE),
    Total_Reviews = sum(Verified_Reviews, na.rm = TRUE)
  ) %>%
  arrange(desc(Count))

cat("Variation totals:\n")
```

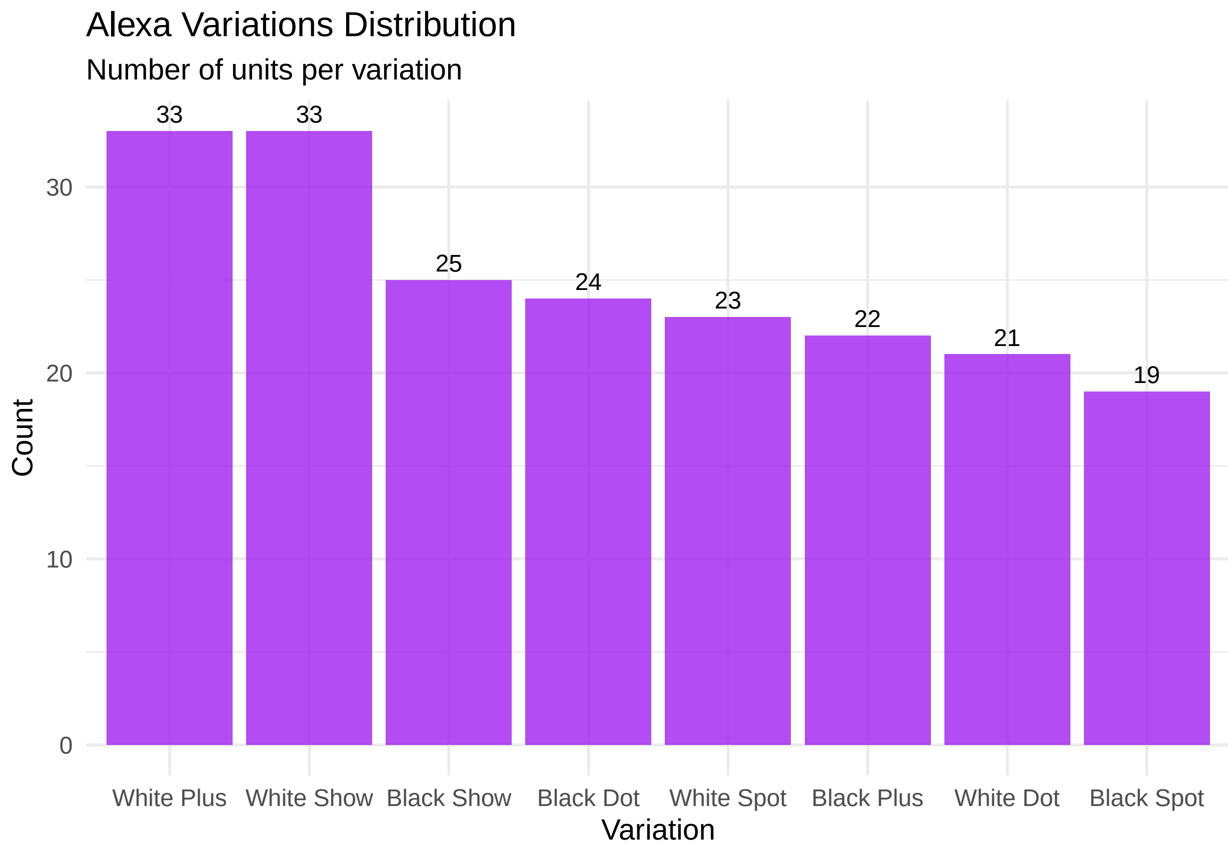
```
## Variation totals:
```

```r
print(variation_totals)
```

```
## # A tibble: 8 x 4
##   Variations Count Avg_Rating Total_Reviews
##   <chr>      <int>      <dbl>         <int>
## 1 White Plus    33       3.76          9379
## 2 White Show    33       3.88          8499
## 3 Black Show    25       4.08          6035
## 4 Black Dot     24       4.29          5677
## 5 White Spot    23       3.87          6765
## 6 Black Plus    22       3.68          5560
## 7 White Dot     21       3.81          5311
## 8 Black Spot    19       3.95          4228
```

c. Plot the variations using ggplot()

```r
p9 <- ggplot(variation_totals, aes(x = reorder(Variations, -Count), y = Count)) +
  geom_bar(stat = "identity", fill = "purple", alpha = 0.8) +
  labs(title = "Alexa Variations Distribution",
       x = "Variation",
       y = "Count",
       subtitle = "Number of units per variation") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme_minimal() +
  geom_text(aes(label = Count), vjust = -0.5, size = 3)
print(p9)
```




```
cat("\nObservation: The bar plot shows the distribution of different Alexa variations.\n")
```

```
##
## Observation: The bar plot shows the distribution of different Alexa variations.
```

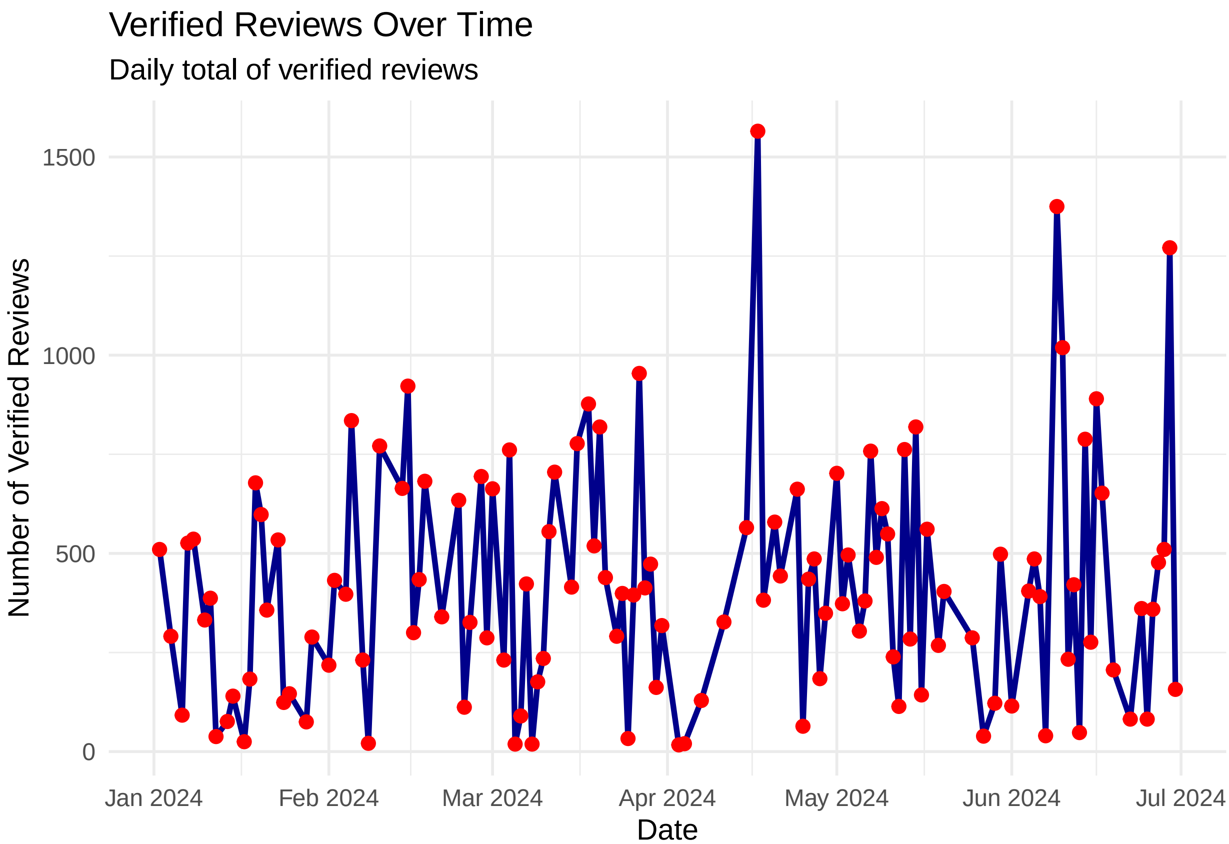
```
cat("We can see which variations are most/least common in the dataset.\n")
```

```
## We can see which variations are most/least common in the dataset.
```

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

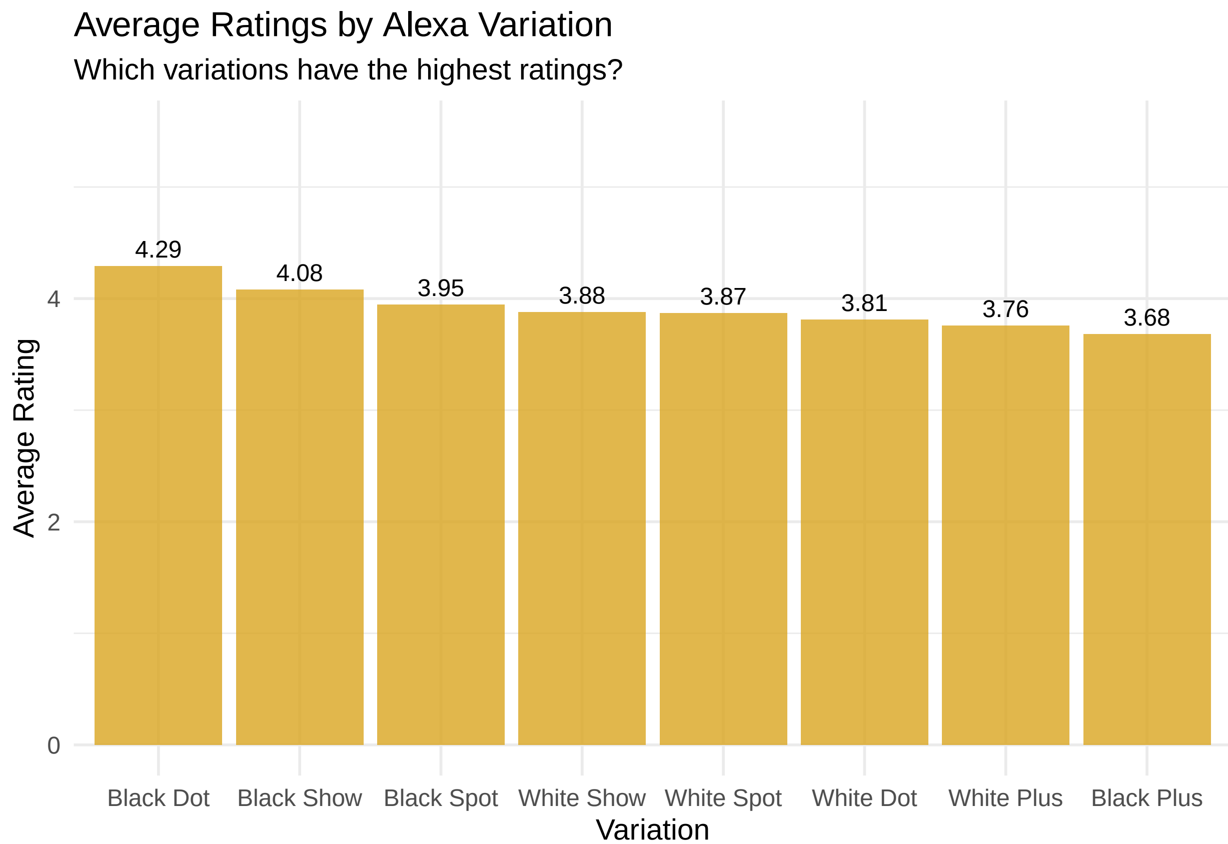
```
# First, aggregate data by date
daily_reviews <- alexa %>%
  group_by(Date) %>%
  summarise(Total_Verified_Reviews = sum(Verified_Reviews, na.rm = TRUE))

p10 <- ggplot(daily_reviews, aes(x = Date, y = Total_Verified_Reviews)) +
  geom_line(color = "darkblue", linewidth = 1) +
  geom_point(color = "red", size = 2) +
  labs(title = "Verified Reviews Over Time",
       x = "Date",
       y = "Number of Verified Reviews",
       subtitle = "Daily total of verified reviews") +
  theme_minimal() +
  scale_x_date(date_breaks = "1 month", date_labels = "%b %Y")
print(p10)
```



e. Get the relationship of variations and ratings.

```
# Which variations got the most highest in rating?
p11 <- ggplot(variation_totals, aes(x = reorder(Variations, -Avg_Rating), y = Avg_Rating)) +
  geom_bar(stat = "identity", fill = "goldenrod", alpha = 0.8) +
  labs(title = "Average Ratings by Alexa Variation",
       x = "Variation",
       y = "Average Rating",
       subtitle = "Which variations have the highest ratings?") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme_minimal() +
  geom_text(aes(label = round(Avg_Rating, 2)), vjust = -0.5, size = 3) +
  ylim(0, 5.5)
print(p11)
```




```r
# Find variation with highest average rating
highest_rating <- variation_totals %>%
  arrange(desc(Avg_Rating)) %>%
  slice(1)

cat("\nVariation with highest average rating:\n")
```

```
##
## Variation with highest average rating:
```

```r
print(highest_rating)
```

```
## # A tibble: 1 x 4
##   Variations Count Avg_Rating Total_Reviews
##   <chr>      <int>      <dbl>         <int>
## 1 Black Dot     24       4.29          5677
```