Homework 3

Adyan Rahman

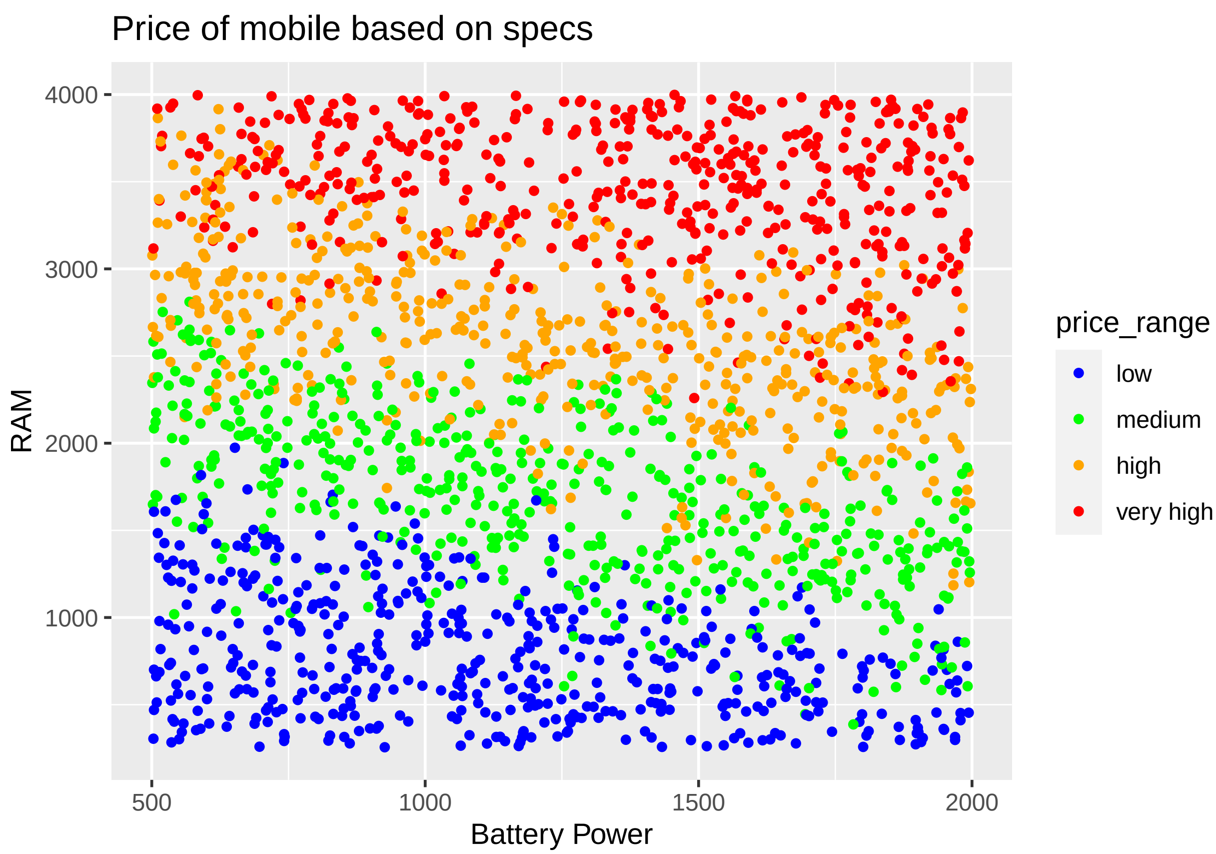
Problem 1

(a)

```r
# Opening the data set
setwd("C:\\Users\\ktzr1\\OneDrive\\Desktop\\STAT3355 Datasets")

mobile_data <- read.csv("train.csv")

# Convert "price_range" to a factor with specified levels
mobile_data$price_range <- factor(mobile_data$price_range,
                                  levels = c("0", "1", "2", "3"),
                                  labels = c("low", "medium", "high", "very high"))
```

(b)

```r
# Load the ggplot2 package
library(ggplot2)

# Scatter plot with colors based on price range
ggplot(mobile_data, aes(x = battery_power, y = ram, color = price_range)) +
  geom_point(shape = 16) +
  labs(title = "Price of mobile based on specs", x = "Battery Power", y = "RAM") +
  scale_color_manual(values = c("low" = "blue", "medium" = "green", "high" = "orange",
                                "very high" = "red"))
```



(c)

```r
# Calculate the Pearson correlation coefficient
correlation <- cor(mobile_data$ram, mobile_data$battery_power)

# Print the correlation coefficient
print(correlation)
```

```
## [1] -0.0006529264
```

(d)

```r
# Subset data into four separate data sets based on 'price_range'
price_low <- mobile_data[mobile_data$price_range == "low", ]
price_medium <- mobile_data[mobile_data$price_range == "medium", ]
price_high <- mobile_data[mobile_data$price_range == "high", ]
price_very_high <- mobile_data[mobile_data$price_range == "very high", ]

# Print the first few rows of each subset
head(price_low)
```

```
##    battery_power blue clock_speed dual_sim fc four_g int_memory m_dep mobile_wt
## 8           1954    0         0.5        1  0      0         24   0.8       187
## 9           1445    1         0.5        0  0      0         53   0.7       174
## 10           509    1         0.6        1  2      1          9   0.1        93
```

```
## 15          1866    0         0.5        0 13      1         52   0.7       185
## 16           775    0         1.0        0  3      0         46   0.7       159
## 24          1602    1         2.8        1  4      1         38   0.7       114
##    n_cores pc px_height px_width  ram sc_h sc_w talk_time three_g touch_screen
## 8        4  0       512     1149  700   16    3         5       1            1
## 9        7 14       386      836 1099   17    1        20       1            0
## 10       5 15      1137     1224  513   19   10        12       1            0
## 15       1 17       356      563  373   14    9         3       1            0
## 16       2 16       862     1864  568   17   15        11       1            1
## 24       3 20       466      788 1037    8    7        20       1            0
##    wifi price_range
## 8     1         low
## 9     0         low
## 10    0         low
## 15    1         low
## 16    1         low
## 24    0         low
```

```r
head(price_medium)
```

```
##    battery_power blue clock_speed dual_sim fc four_g int_memory m_dep mobile_wt
## 1            842    0         2.2        0  1      0          7   0.6       188
## 5           1821    1         1.2        0 13      1         44   0.6       141
## 6           1859    0         0.5        1  3      0         22   0.7       164
## 13          1815    0         2.8        0  2      0         33   0.6       159
## 19          1131    1         0.5        1 11      0         49   0.6       101
## 20           682    1         0.5        0  4      0         19   1.0       121
##    n_cores pc px_height px_width  ram sc_h sc_w talk_time three_g touch_screen
## 1        2  2        20      756 2549    9    7        19       0            0
## 5        2 14      1208     1212 1411    8    2        15       1            1
## 6        1  7      1004     1654 1067   17    1        10       1            0
## 13       4 17       607      748 1482   18    0         2       1            0
## 19       5 18       658      878 1835   19   13        16       1            1
## 20       4 11       902     1064 2337   11    1        18       0            1
##    wifi price_range
## 1     1      medium
## 5     0      medium
## 6     0      medium
## 13    0      medium
## 19    0      medium
## 20    1      medium
```

```r
head(price_high)
```

```
##    battery_power blue clock_speed dual_sim fc four_g int_memory m_dep mobile_wt
## 2           1021    1         0.5        1  0      1         53   0.7       136
## 3            563    1         0.5        1  2      1         41   0.9       145
## 4            615    1         2.5        0  0      0         10   0.8       131
## 14           803    1         2.1        0  7      0         17   1.0       198
## 26           961    1         1.4        1  0      1         57   0.6       114
## 29          1453    0         1.6        1 12      1         52   0.3        96
##    n_cores pc px_height px_width  ram sc_h sc_w talk_time three_g touch_screen
## 2        3  6       905     1988 2631   17    3         7       1            1
```

```
## 3         5  6      1263      1716 2603   11    2         9       1            1
## 4         6  9      1216      1786 2769   16    8        11       1            0
## 14        4 11       344      1440 2680    7    1         4       1            0
## 26        8  3       291      1434 2782   18    9         7       1            1
## 29        2 18       187      1311 2373   10    1        10       1            1
##    wifi price_range
## 2     0        high
## 3     0        high
## 4     0        high
## 14    1        high
## 26    1        high
## 29    1        high
```

```r
head(price_very_high)
```

```
##    battery_power blue clock_speed dual_sim fc four_g int_memory m_dep mobile_wt
## 7           1821    0         1.7        0  4      1         10   0.8       139
## 11           769    1         2.9        1  0      0          9   0.1       182
## 12          1520    1         2.2        0  5      1         33   0.5       177
## 17           838    0         0.5        0  1      1         13   0.1       196
## 18           595    0         0.9        1  7      1         23   0.1       121
## 21           772    0         1.1        1 12      0         39   0.8        81
##    n_cores pc px_height px_width  ram sc_h sc_w talk_time three_g touch_screen
## 7        8 10       381     1018 3220   13    8        18       1            0
## 11       5  1       248      874 3946    5    2         7       0            0
## 12       8 18       151     1005 3826   14    9        13       1            1
## 17       8  4       984     1850 3554   10    9        19       1            0
## 18       3 17       441      810 3752   10    2        18       1            1
## 21       7 14      1314     1854 2819   17   15         3       1            1
##    wifi price_range
## 7     1   very high
## 11    0   very high
## 12    1   very high
## 17    1   very high
## 18    0   very high
## 21    0   very high
```

(e)

```r
# Calculate Pearson correlation coefficient for each subset
correlation_low <- cor(price_low$ram, price_low$battery_power)
correlation_medium <- cor(price_medium$ram, price_medium$battery_power)
correlation_high <- cor(price_high$ram, price_high$battery_power)
correlation_veryhigh <- cor(price_very_high$ram, price_very_high$battery_power)

# Print the correlations
print(paste("Correlation for Low Price Range:", correlation_low))
```

```
## [1] "Correlation for Low Price Range: -0.346587767926678"
```

```r
print(paste("Correlation for Medium Price Range:", correlation_medium))
```

```
## [1] "Correlation for Medium Price Range: -0.613397054349082"
```

```r
print(paste("Correlation for High Price Range:", correlation_high))
```

```
## [1] "Correlation for High Price Range: -0.587408571267869"
```

```r
print(paste("Correlation for Very High Price Range:", correlation_veryhigh))
```

```
## [1] "Correlation for Very High Price Range: -0.262758864930475"
```

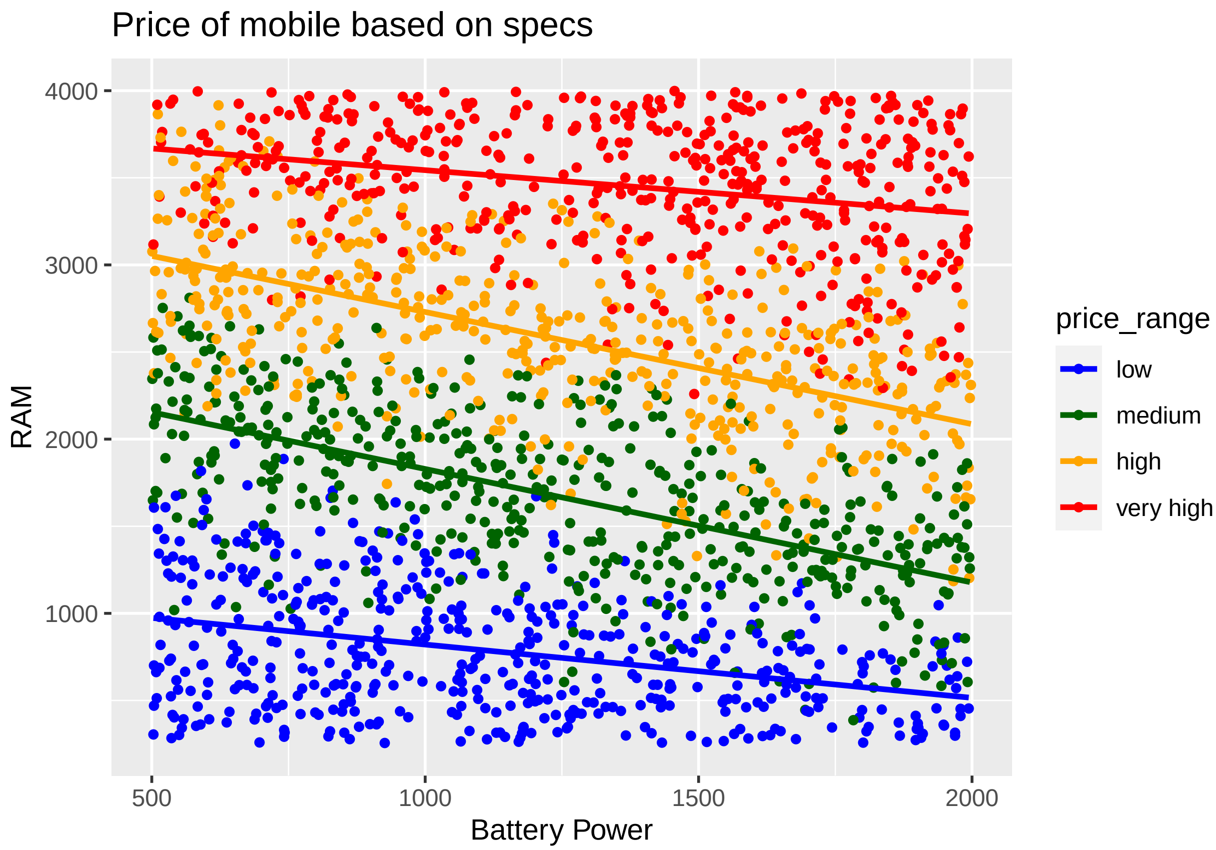
```r
# Explain any correlations you might find in terms of how a cellphone operates:

# Low Price Range: A higher correlation between RAM and battery power might
# suggest that phones in the low-price range often come with lower RAM and battery
# power, which may correlate with each other due to budget constraints or lower-end
# specifications.

# # Medium Price Range: The correlation might be moderate, indicating a somewhat
# consistent pattern of RAM and battery power as the phone price increases but not
# as stark as low-price ranges.

# High Price Range: In the high-price range, the correlation might be lower
# or negligible, suggesting that other factors become more influential in
# determining the specifications of the phone, such as camera quality,
# screen resolution, or brand reputation. Thus, RAM and battery power may not
# correlate strongly.

# Very High Price Range: Similar to the high-price range, the correlation
# might be even weaker as phones in this range often offer a wide variety of
# features, and consumers may prioritize different specifications over RAM and
# battery power.

# Why is this result so much different from the one that we found in Part c?

# The overall correlation might have been driven by a mix of different price ranges,
# leading to an average correlation across all data. When analyzing subsets based
# on price ranges, you're looking at more homogenous groups of phones with similar
# price points, features, and target markets, which can result in different patterns
# and correlations within each group.
```

(f)

```r
# Scatter plot with colors based on price range
graph <- ggplot(mobile_data, aes(x = battery_power, y = ram, color = price_range)) +
  geom_point(shape = 16) +
  labs(title = "Price of mobile based on specs", x = "Battery Power", y = "RAM") +
  scale_color_manual(values = c("low" = "blue", "medium" = "darkgreen", "high" = "orange",
                                "very high" = "red"))
```

```r
# Add trend lines for each price range separately
graph + geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



(g)

```r
# Subset the data for processors with 4, 6, and 8 cores
clock_sp_sub <- subset(mobile_data, n_cores %in% c(4, 6, 8))

# Calculate the average clock speed
average_clock_speed <- round(mean(clock_sp_sub$clock_speed), 2)

# Calculate the median clock speed
median_clock_speed <- round(median(clock_sp_sub$clock_speed), 2)

# Print the results
print(paste("Average Clock Speed:", average_clock_speed))
```

```
## [1] "Average Clock Speed: 1.53"
```

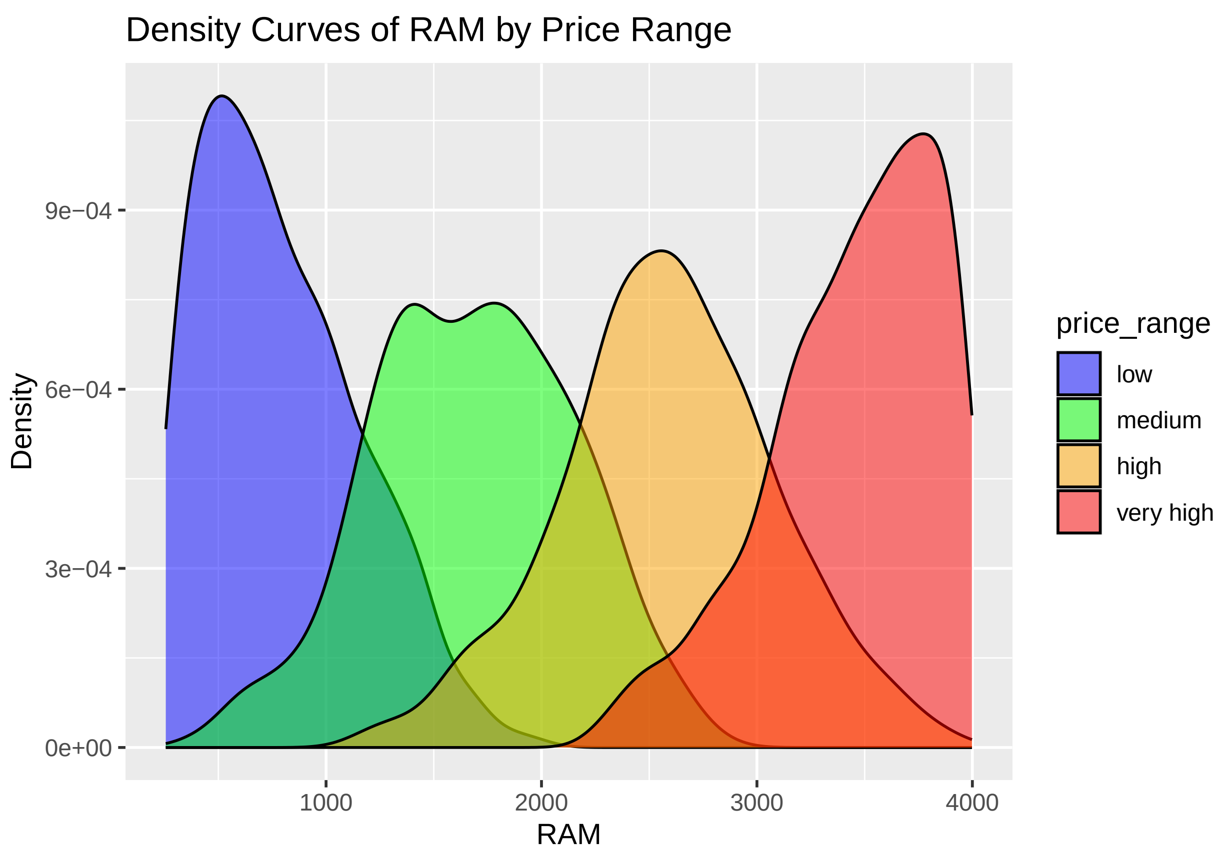
```r
print(paste("Median Clock Speed:", median_clock_speed))
```

```
## [1] "Median Clock Speed: 1.5"
```

```r
# The clock speed of processors with 4, 6, and 8 cores may not change
# significantly because the number of cores does not directly impact the clock
# speed of the processor models being compared. Therefore, the average and median
# clock speeds remain relatively stable across different core counts.
```

(h)

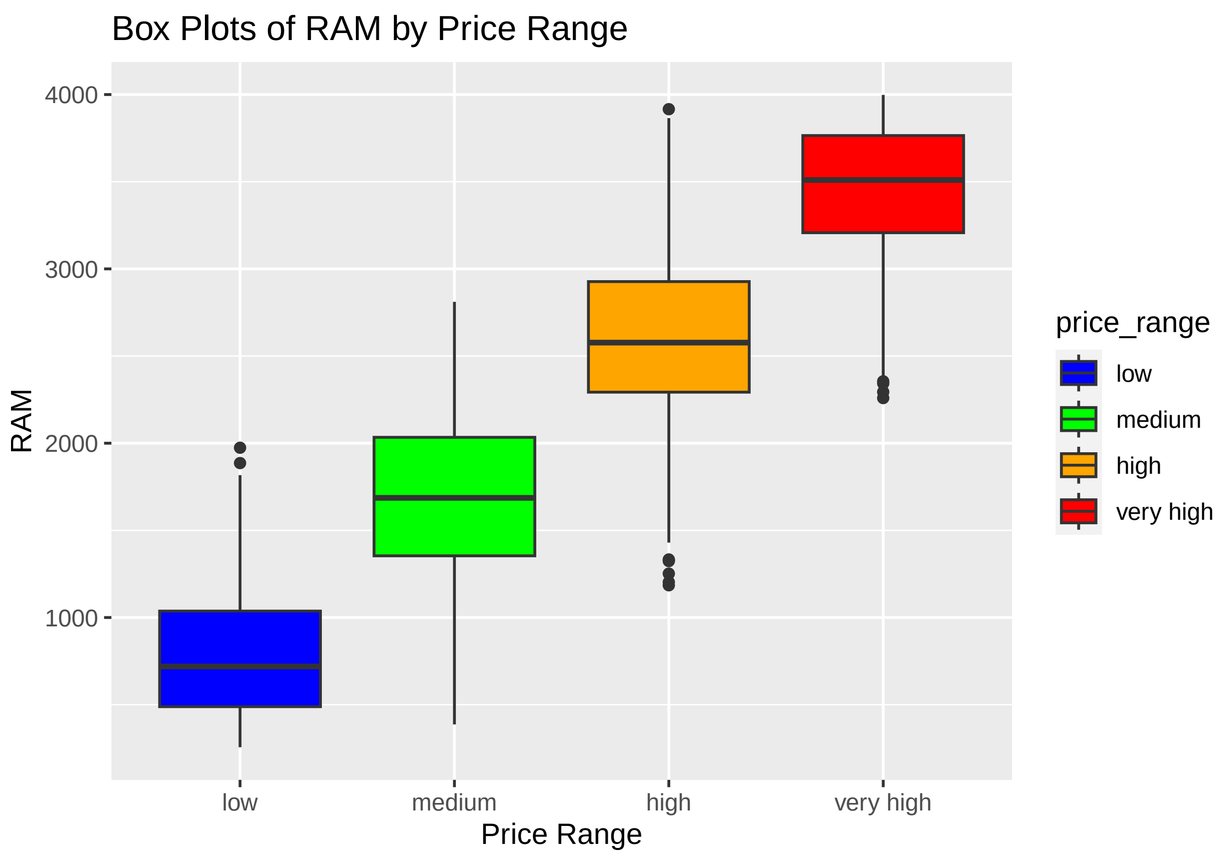
```r
# Create density curves for RAM by price range
density_plot <- ggplot(mobile_data, aes(x = ram, fill = price_range)) +
  geom_density(alpha = 0.5) +
  labs(title = "Density Curves of RAM by Price Range",
       x = "RAM",
       y = "Density") +
  scale_fill_manual(values = c("blue", "green", "orange", "red"))

# Print the plot
print(density_plot)
```



```
# Low Price Range: The density curve might be skewed to the right,
# indicating that there are more phones with lower RAM configurations in the low
# price range.

# Medium Price Range: The density curve may show a relatively normal
# distribution with a peak around the median RAM, indicating a balanced
# distribution of RAM configurations in the medium price range.

# High Price Range: The density curve might be skewed to the left or have
# a longer tail on the right, indicating that there are more phones with higher
# RAM configurations in the high price range.

# Very High Price Range: The density curve might be more symmetric or bimodal,
# indicating that there is a wider range of RAM configurations available in the
# very high price range, potentially catering to different market segments.
```

(i)

```
# Create box plots for RAM by price range
box_plot <- ggplot(mobile_data, aes(x = price_range, y = ram, fill = price_range)) +
  geom_boxplot() +
  labs(title = "Box Plots of RAM by Price Range",
       x = "Price Range",
       y = "RAM") +
  scale_fill_manual(values = c("blue", "green", "orange", "red"))

# Print the plot
print(box_plot)
```

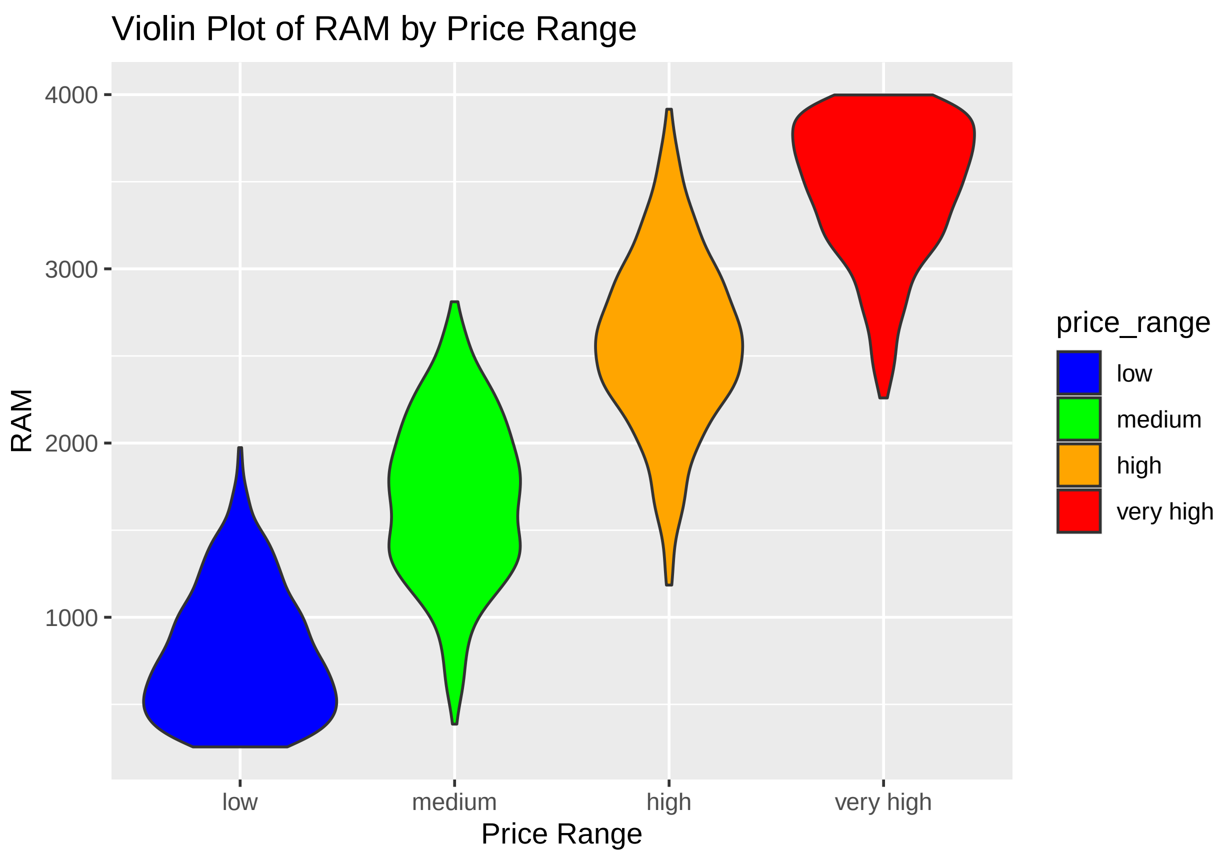



```r
# Low Price Range: The box plot might have a lower median and shorter
# interquartile range (IQR), indicating that phones in the low price range tend
# to have lower RAM configurations.

# Medium Price Range: The box plot may have a moderate median and a
# balanced distribution of RAM configurations, with an average IQR.

# High Price Range: The box plot might have a higher median and longer IQR,
# indicating that phones in the high price range tend to have higher
# RAM configurations.

# Very High Price Range: The box plot might have the highest median and the widest
# IQR, indicating that phones in the very high price range offer a wide range of
# RAM configurations, catering to diverse consumer needs.
```

(j)

```r
# Create a violin plot of RAM by price range
violin_plot <- ggplot(mobile_data, aes(x = price_range, y = ram, fill = price_range)) +
  geom_violin() +
  labs(title = "Violin Plot of RAM by Price Range",
       x = "Price Range",
       y = "RAM") +
  scale_fill_manual(values = c("blue", "green", "orange", "red"))
```

```r
# Print the plot
print(violin_plot)
```




```r
# Low Price Range: The violin plot might be narrower and shorter, indicating
# that there is less variability in RAM configurations for lower-priced mobile phones.
#
# Medium Price Range: The violin plot might be wider and taller, suggesting a
# broader range of RAM configurations for medium-priced mobile phones.
#
# High Price Range: The violin plot might be narrower and taller, indicating that
# there is less variability but higher RAM configurations for high-priced mobile phones.
#
# Very High Price Range: The violin plot might be wider and flatter, showing a wide
# range of RAM configurations and potentially indicating more diversity in the
# types of phones available in this price range.
```

(k)

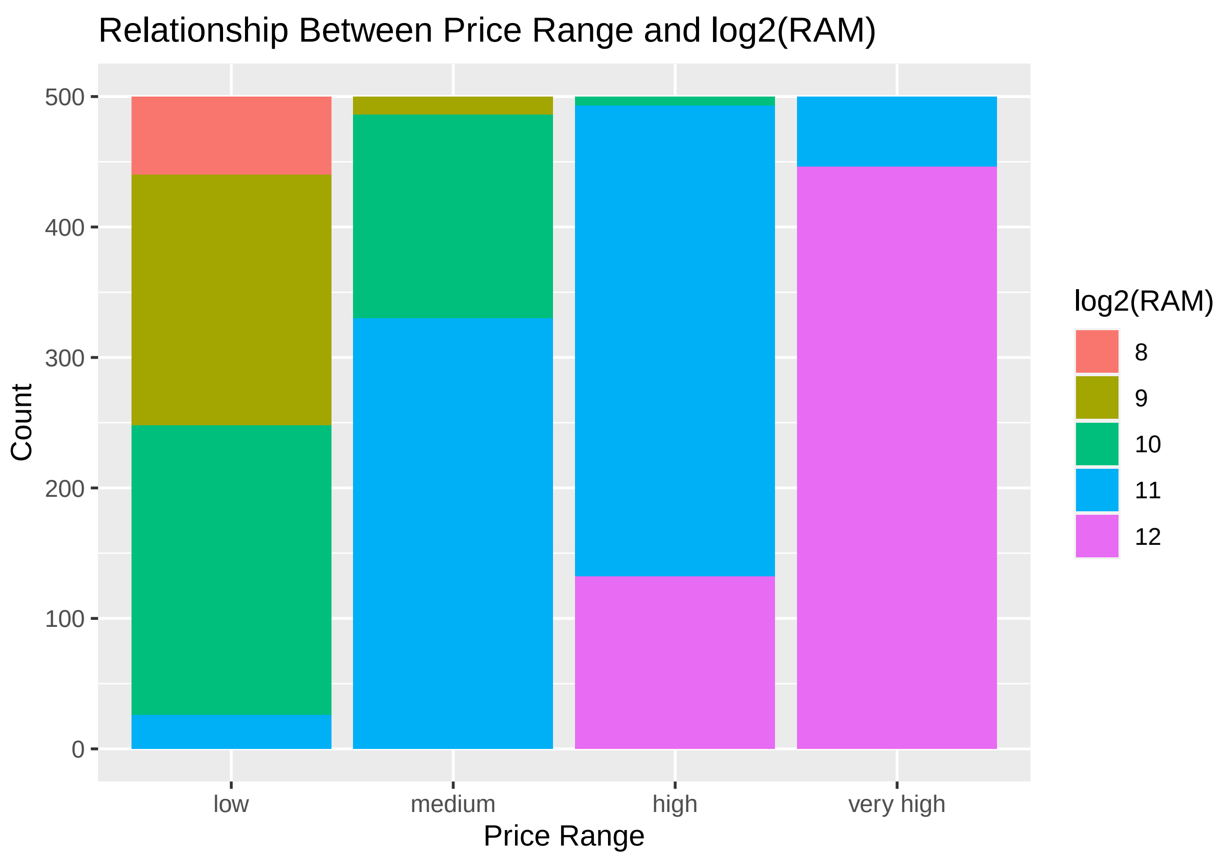
```r
# Create a factor variable by taking the log2(ram) and rounding to the nearest
# whole number
ram_log <- as.factor(round(log2(mobile_data$ram)))

# Print the unique values of the factor variable
print(unique(ram_log))
```

```
## [1] 11 10 12 9  8
## Levels: 8 9 10 11 12
```

```r
# Creating a factor variable out of RAM by taking the log2 of RAM is a sensible
# approach because it helps to normalize the distribution of RAM sizes and
# facilitates the interpretation of RAM sizes in a categorical manner, making
# it easier to identify patterns and relationships in the data.
```

(l)

```r
# Create the stacked bar plot
ggplot(mobile_data, aes(x = price_range, fill = ram_log)) +
  geom_bar(position = "stack") +
  labs(title = "Relationship Between Price Range and log2(RAM)",
       x = "Price Range",
       y = "Count") +
  scale_fill_discrete(name = "log2(RAM)")
```



Problem 2

(a)

```r
# mpg is includes in ggplot2, so we can read it like as such
data(mpg)

# Convert the 'cyl' variable to an ordered factor variable with specified levels
mpg$cyl <- factor(mpg$cyl, ordered = TRUE, levels = c("4", "5", "6", "8"))

# View the structure of the mpg dataset to confirm the change
str(mpg$cyl)
```

```
##  Ord.factor w/ 4 levels "4"<"5"<"6"<"8": 1 1 1 1 3 3 3 1 1 1 ...
```

(b)

```r
# Extract substrings from the 'trans' variable
trans_substr <- substr(mpg$trans, 1, 4)

# Convert the extracted substrings to a factor variable with unique values "auto"
# and "manu"
mpg$trans <- factor(trans_substr, levels = c("auto", "manu"))

# View the unique values of the 'trans' variable to verify the change
unique(mpg$trans)
```

```
## [1] auto manu
## Levels: auto manu
```

(c)

```r
# Convert the 'drv' variable to an ordered factor variable with specified levels
mpg$drv <- factor(mpg$drv, ordered = TRUE, levels = c("f", "r", "4"))

# View the structure of the 'drv' variable to confirm the change
str(mpg$drv)
```

```
##  Ord.factor w/ 3 levels "f"<"r"<"4": 1 1 1 1 1 1 1 3 3 3 ...
```

(d)

```r
# Replacing values
mpg$fl[mpg$fl == "e" | mpg$fl == "c"] <- "other"
mpg$fl[mpg$fl == "p" | mpg$fl == "r"] <- "gasoline"
mpg$fl[mpg$fl == "d"] <- "diesel"

# Converting to factor
mpg$fl <- factor(mpg$fl)


# View the unique values of the 'fl' variable to verify the change
unique(mpg$fl)
```

```
## [1] gasoline other    diesel
## Levels: diesel gasoline other
```

(e)

```r
# Convert the 'class' variable to an ordered factor variable with specified levels
mpg$class <- factor(mpg$class, ordered = TRUE, levels = c("2seater", "subcompact",
                                                          "compact", "midsize",
                                                          "suv", "minivan", "pickup"))

# View the structure of the 'class' variable to confirm the change
str(mpg$class)
```

```
##  Ord.factor w/ 7 levels "2seater"<"subcompact"<..: 3 3 3 3 3 3 3 3 3 3 ...
```

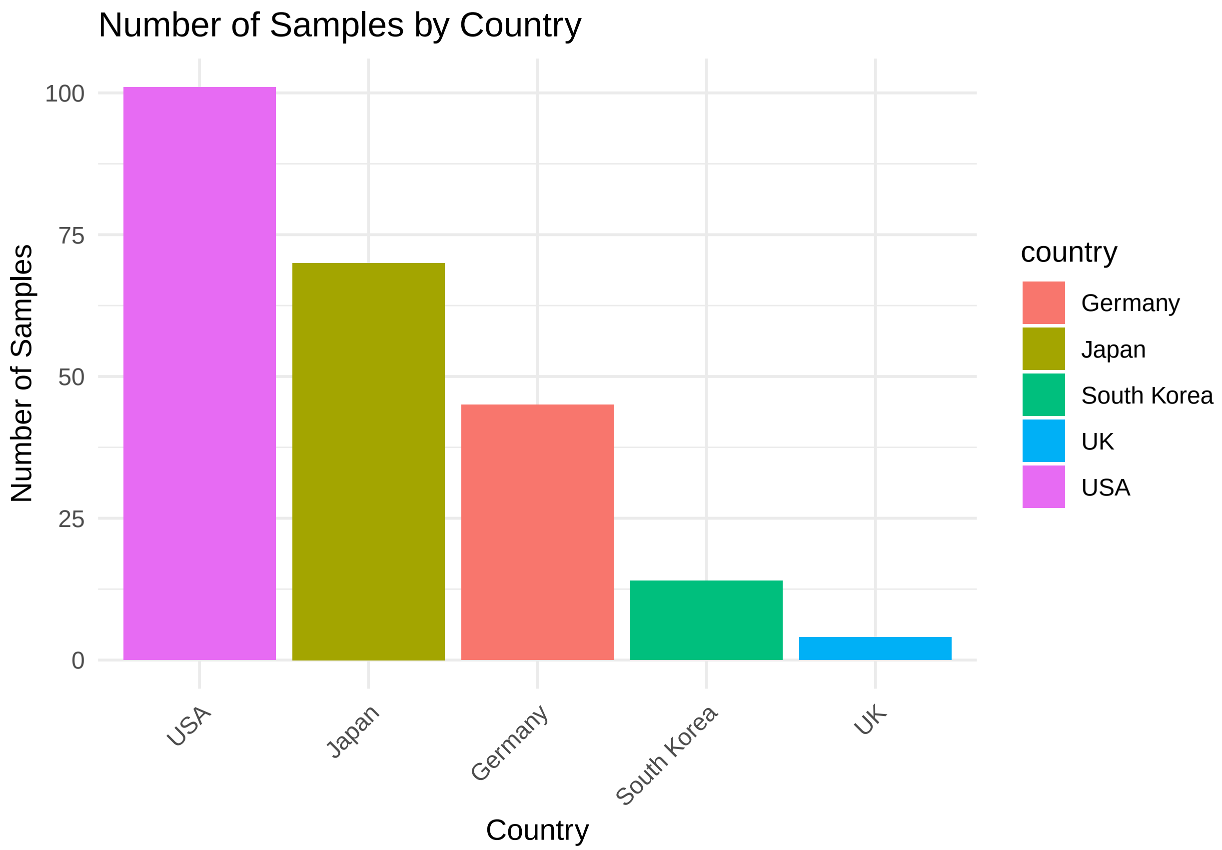
(f)

```r
# Create a new variable 'country' indicating the manufacturer's base location
mpg$country <- NA  # Initialize the 'country' variable with NA values

# Define a lookup table for manufacturer and corresponding country
country_lookup <- list(
  "audi" = "Germany",
  "chevrolet" = "USA",
  "dodge" = "USA",
  "ford" = "USA",
  "honda" = "Japan",
  "hyundai" = "South Korea",
  "jeep" = "USA",
  "land rover" = "UK",
  "lincoln" = "USA",
  "mercury" = "USA",
  "nissan" = "Japan",
  "pontiac" = "USA",
  "subaru" = "Japan",
  "toyota" = "Japan",
  "volkswagen" = "Germany"
)

# Assign the country based on manufacturer's name
for (i in 1 : nrow(mpg)) {
  manufacturer <- tolower(mpg$manufacturer[i])  # Convert to lowercase for case-insensitivity
  if (manufacturer %in% names(country_lookup)) {
    mpg$country[i] <- country_lookup[[manufacturer]]
  } else {
    mpg$country[i] <- "Unknown"  # Assign 'Unknown' for missing or unmatched manufacturers
  }
}

# View the unique values of the 'country' variable
unique(mpg$country)
```

```
## [1] "Germany"     "USA"         "Japan"       "South Korea" "UK"
```

(g)

```
# Draw a bar plot of the variable 'country'
ggplot(mpg, aes(x = reorder(country, -table(country)[country]), fill = country)) +
  geom_bar() +
  labs(title = "Number of Samples by Country",
       x = "Country",
       y = "Number of Samples") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better visibilit
```



(h)

```
# Filter the dataset to include only U.S. cars
us_cars <- subset(mpg, manufacturer %in% c("chevrolet",
                                           "dodge", "ford", "jeep", "lincoln",
                                           "mercury", "pontiac"))

# Summary statistics for engine displacement (displ)
summary(us_cars$displ)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.400   3.900   4.700   4.572   5.300   7.000
```

```r
# Summary statistics for number of cylinders (cyl)
summary(us_cars$cyl)
```

```
##  4  5  6  8
##  3  0 37 61
```

```r
# Summary of transmission types (trans)
table(us_cars$trans)
```

```
##
## auto manu
##   83   18
```

```r
# Summary of drive types (drv)
table(us_cars$drv)
```

```
##
##  f  r  4
## 21 25 55
```

```r
# Summary of fuel types (fl)
table(us_cars$fl)
```

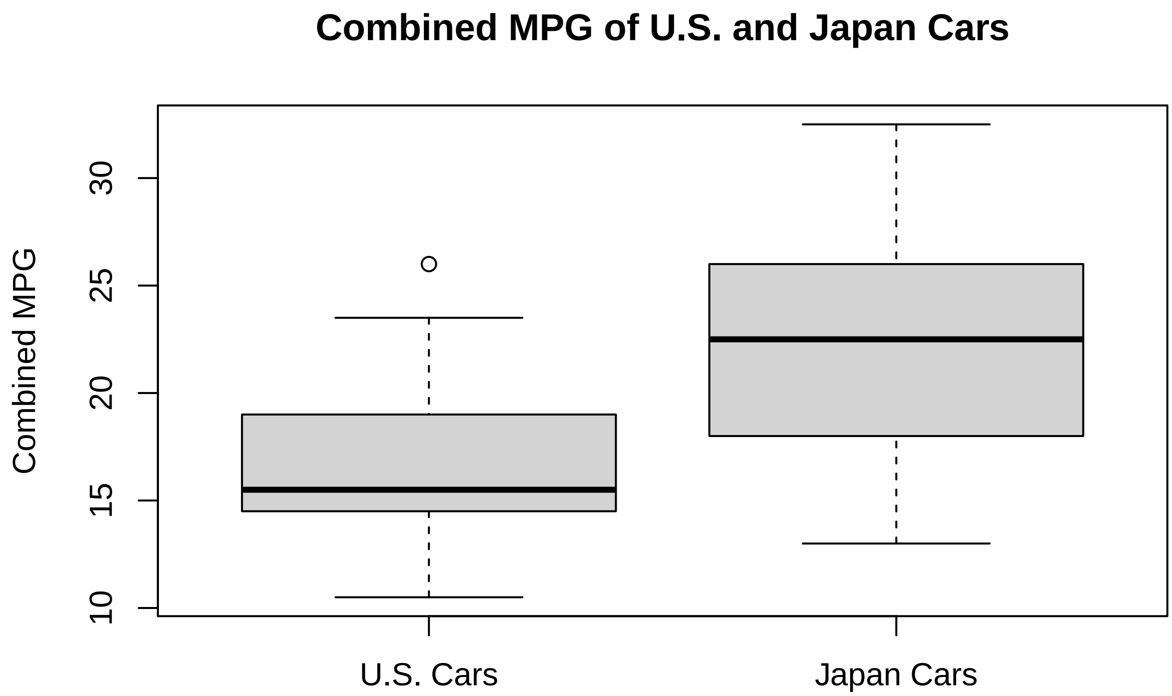
```
##
##   diesel gasoline    other
##        2       91        8
```

```r
# Summary of car types (class)
table(us_cars$class)
```

```
##
##    2seater subcompact    compact    midsize        suv    minivan     pickup
##          5          9          0         10         40         11         26
```

(i)

```r
# Create a new variable for combined miles per gallon (mpg)
mpg$combined_mpg <- (mpg$cty + mpg$hwy) / 2

# Filter the dataset for U.S. and Japan cars
us_cars <- subset(mpg, manufacturer %in% c("chevrolet", "dodge", "ford", "jeep",
                                           "lincoln", "mercury", "pontiac"))
japan_cars <- subset(mpg, manufacturer %in% c("honda", "nissan", "subaru", "toyota"))

# Create a boxplot of combined mpg for U.S. and Japan cars
boxplot(us_cars$combined_mpg, japan_cars$combined_mpg, names = c("U.S. Cars",
                                                                 "Japan Cars"),
        main = "Combined MPG of U.S. and Japan Cars", ylab = "Combined MPG")
```




```
# Calculate statistics for U.S. cars
us_mean <- mean(us_cars$combined_mpg)
us_median <- median(us_cars$combined_mpg)
us_sd <- sd(us_cars$combined_mpg)
us_iqr <- IQR(us_cars$combined_mpg)

# Calculate statistics for Japan cars
japan_mean <- mean(japan_cars$combined_mpg)
japan_median <- median(japan_cars$combined_mpg)
japan_sd <- sd(japan_cars$combined_mpg)
japan_iqr <- IQR(japan_cars$combined_mpg)

# Print the statistics
cat("Statistics for U.S. Cars:\n")
```

```
## Statistics for U.S. Cars:
```

```
cat("Mean:", us_mean, "\n")
```

```
## Mean: 16.63861
```

```
cat("Median:", us_median, "\n")
```

```
## Median: 15.5
```

```r
cat("Standard Deviation:", us_sd, "\n")
```

```
## Standard Deviation: 3.302362
```

```r
cat("Interquartile Range (IQR):", us_iqr, "\n\n")
```

```
## Interquartile Range (IQR): 4.5
```

```r
cat("Statistics for Japan Cars:\n")
```

```
## Statistics for Japan Cars:
```

```r
cat("Mean:", japan_mean, "\n")
```

```
## Mean: 22.66429
```

```r
cat("Median:", japan_median, "\n")
```

```
## Median: 22.5
```

```r
cat("Standard Deviation:", japan_sd, "\n")
```

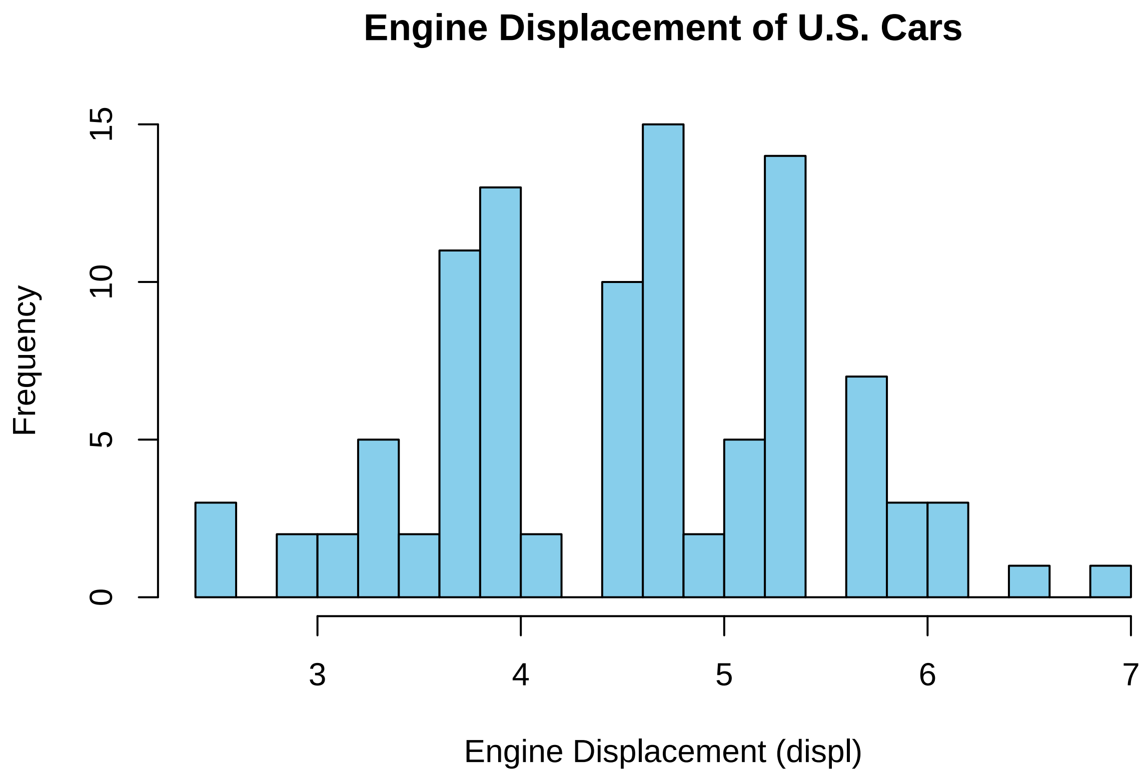
```
## Standard Deviation: 4.60208
```

```r
cat("Interquartile Range (IQR):", japan_iqr, "\n")
```

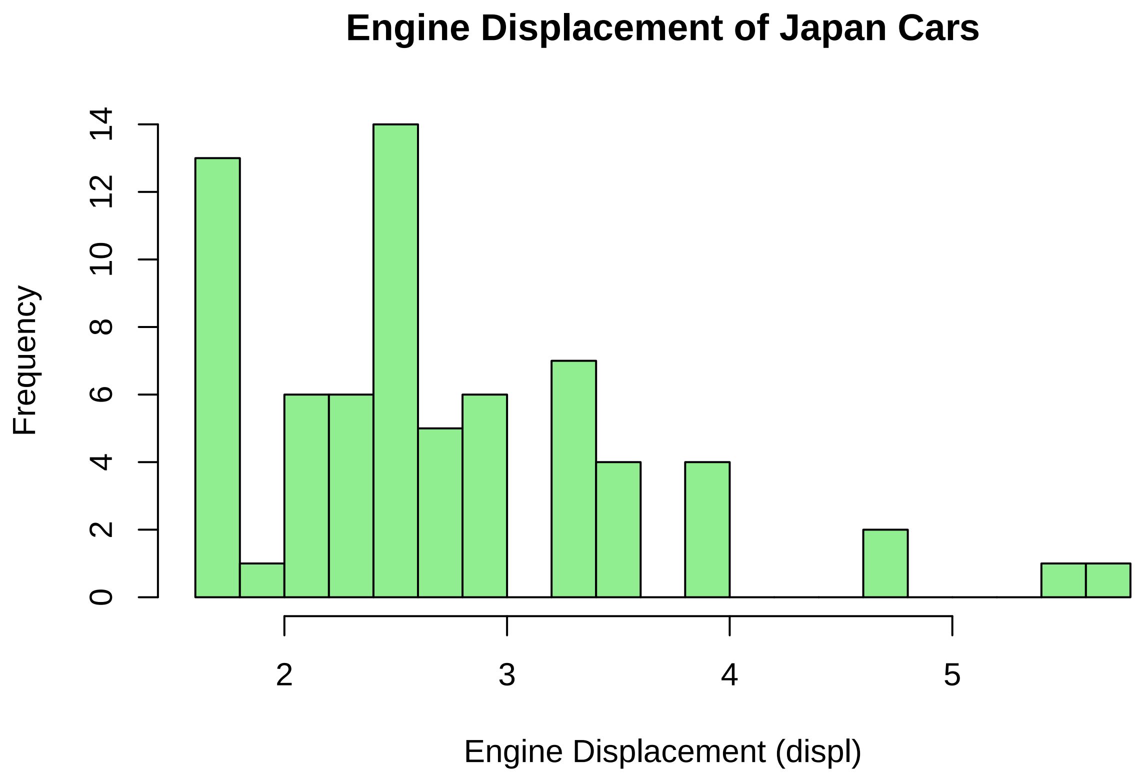
```
## Interquartile Range (IQR): 7.625
```

(j)

```r
# Filter the dataset for U.S. and Japan cars
us_cars <- subset(mpg, manufacturer %in% c("chevrolet", "dodge", "ford", "jeep",
                                            "lincoln", "mercury", "pontiac"))
japan_cars <- subset(mpg, manufacturer %in% c("honda", "nissan", "subaru", "toyota"))

# Create a histogram of engine displacement for U.S. cars
hist(us_cars$displ, breaks = 20, col = "skyblue", main = "Engine Displacement of U.S. Cars",
     xlab = "Engine Displacement (displ)", ylab = "Frequency")
```




```r
# Create a histogram of engine displacement for Japan cars
hist(japan_cars$displ, breaks = 20, col = "lightgreen", main = "Engine Displacement of Japan Cars",
     xlab = "Engine Displacement (displ)", ylab = "Frequency")
```



Problem 3

(a) Team Name: Team 22

Team Member's names and majors: Adyan Rahman - Major: Data Science Jimmy Harvin - Major: Computer Science Ashish Adhikari - Major: Math