STAT3355(HW-3)

Tulasi Janjanam

2024-10-01

Problem 1

(a)

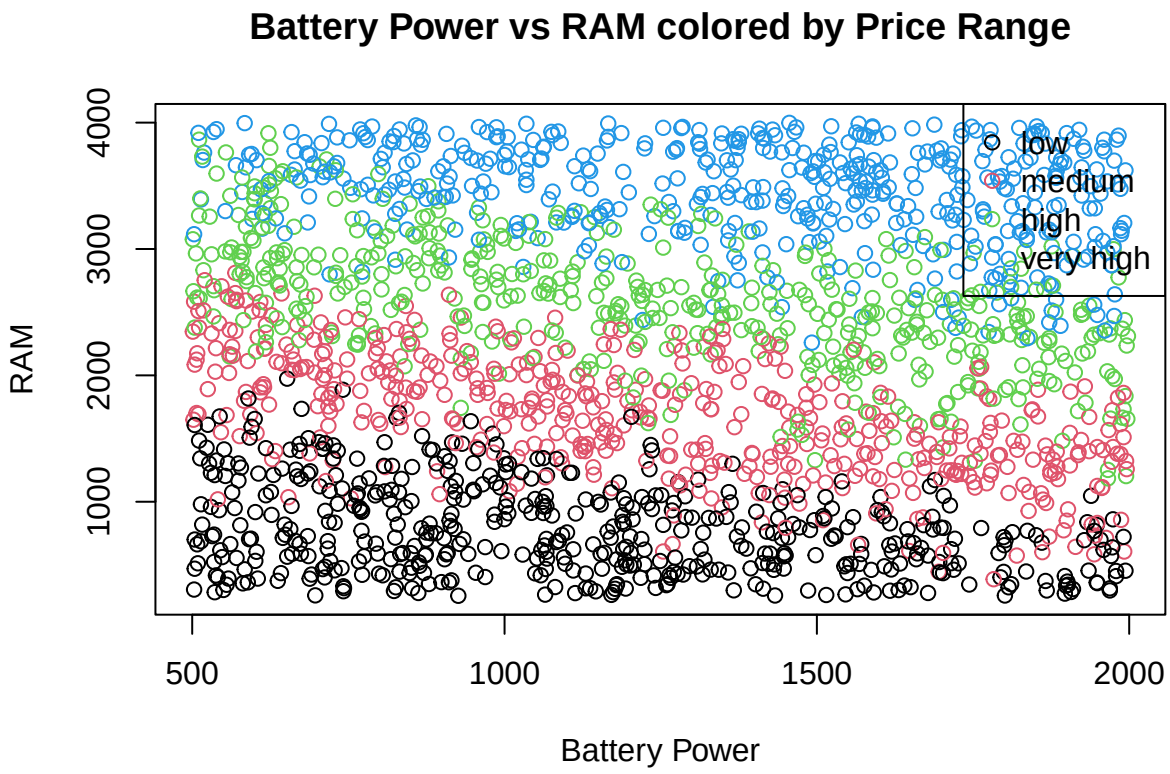
```
mobile_data <- read.csv("/Users/tulasijanjanam/Downloads/archive (2)/train.csv")
str(mobile_data)
```

```
## 'data.frame':    2000 obs. of  21 variables:
##  $ battery_power: int  842 1021 563 615 1821 1859 1821 1954 1445 509 ...
##  $ blue         : int  0 1 1 1 1 0 0 0 1 1 ...
##  $ clock_speed  : num  2.2 0.5 0.5 2.5 1.2 0.5 1.7 0.5 0.5 0.6 ...
##  $ dual_sim     : int  0 1 1 0 0 1 0 1 0 1 ...
##  $ fc           : int  1 0 2 0 13 3 4 0 0 2 ...
##  $ four_g       : int  0 1 1 0 1 0 1 0 0 1 ...
##  $ int_memory   : int  7 53 41 10 44 22 10 24 53 9 ...
##  $ m_dep        : num  0.6 0.7 0.9 0.8 0.6 0.7 0.8 0.8 0.7 0.1 ...
##  $ mobile_wt    : int  188 136 145 131 141 164 139 187 174 93 ...
##  $ n_cores      : int  2 3 5 6 2 1 8 4 7 5 ...
##  $ pc           : int  2 6 6 9 14 7 10 0 14 15 ...
##  $ px_height    : int  20 905 1263 1216 1208 1004 381 512 386 1137 ...
##  $ px_width     : int  756 1988 1716 1786 1212 1654 1018 1149 836 1224 ...
##  $ ram          : int  2549 2631 2603 2769 1411 1067 3220 700 1099 513 ...
##  $ sc_h         : int  9 17 11 16 8 17 13 16 17 19 ...
##  $ sc_w         : int  7 3 2 8 2 1 8 3 1 10 ...
##  $ talk_time    : int  19 7 9 11 15 10 18 5 20 12 ...
##  $ three_g      : int  0 1 1 1 1 1 1 1 1 1 ...
##  $ touch_screen : int  0 1 1 0 1 0 0 1 0 0 ...
##  $ wifi         : int  1 0 0 0 0 0 1 1 0 0 ...
##  $ price_range  : int  1 2 2 2 1 1 3 0 0 0 ...
```

```
mobile_data$price_range <- factor(mobile_data$price_range,
                                  levels = c(0, 1, 2, 3),
                                  labels = c("low", "medium", "high", "very high"))
```

(b)

```r
# Scatter plot
plot(mobile_data$battery_power, mobile_data$ram,
     col = mobile_data$price_range,
     xlab = "Battery Power", ylab = "RAM",
     main = "Battery Power vs RAM colored by Price Range")
legend("topright", legend = levels(mobile_data$price_range),
       col = 1:4, pch = 1)
```



(c)

```r
correlation <- cor(mobile_data$ram, mobile_data$battery_power, method = "pearson")
print(correlation)
```

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

(d)

```r
priceLow <- subset(mobile_data, price_range == "low")
priceMedium <- subset(mobile_data, price_range == "medium")
priceHigh <- subset(mobile_data, price_range == "high")
priceVeryhigh <- subset(mobile_data, price_range == "very high")

head(priceLow)
```

```
##    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
```

```
head(priceMedium)
```

```
##    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
```

```
head(priceHigh)
```

```
##    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
```

```
head(priceVeryhigh)
```

```
##    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)

```
# Low price
cor(priceLow$ram, priceLow$battery_power, method = "pearson")
```

```
## [1] -0.3465878
```

```
# Medium price
cor(priceMedium$ram, priceMedium$battery_power, method = "pearson")
```

```
## [1] -0.6133971
```

```r
# High price
cor(priceHigh$ram, priceHigh$battery_power, method = "pearson")
```

```
## [1] -0.5874086
```

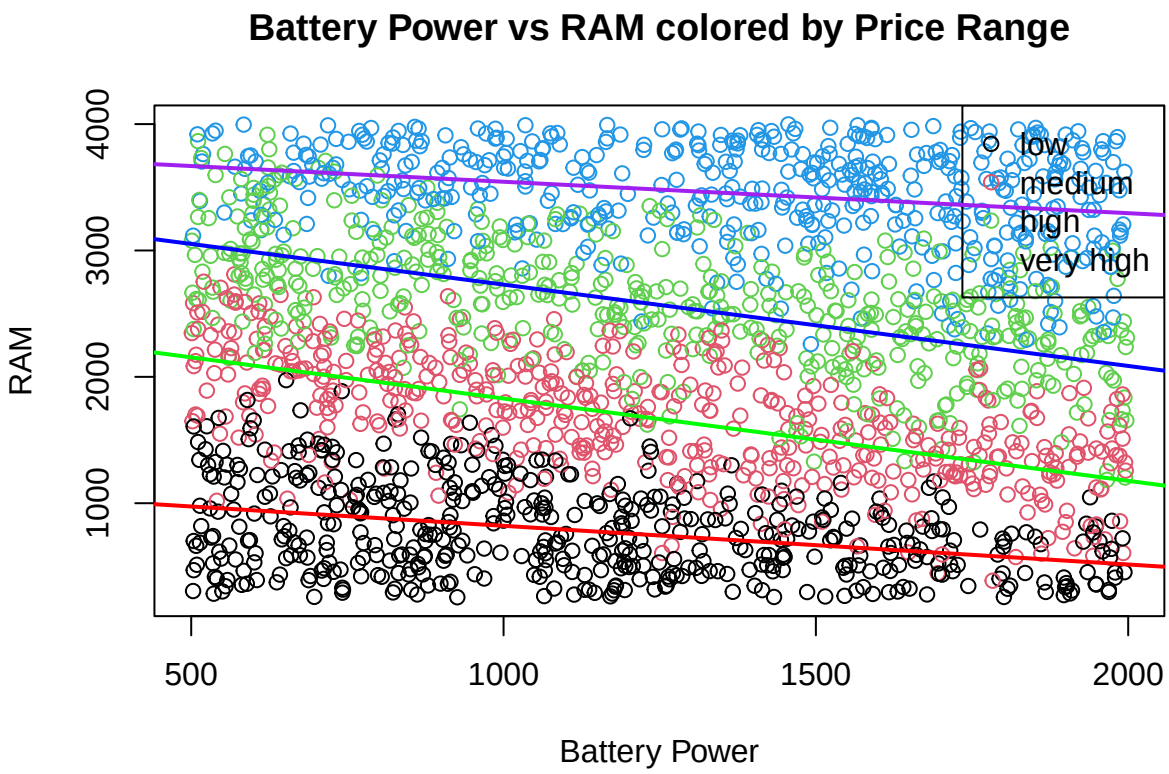
```r
# Very high price
cor(priceVeryhigh$ram, priceVeryhigh$battery_power, method = "pearson")
```

```
## [1] -0.2627589
```

In the lower price ranges, the correlation between RAM and battery power is weaker because low-end phones may not pair high RAM with larger batteries. In higher price ranges, the correlation is stronger, as premium phones tend to have more consistent designs, with higher RAM often paired with larger batteries. The overall correlation from Part (c) averages these relationships across all price ranges, which can mask differences between low-end and high-end phones. Analyzing by price range provides a clearer view of how RAM and battery power relate within each market segment.

(f)

```r
# Scatter plot
plot(mobile_data$battery_power, mobile_data$ram,
     col = mobile_data$price_range,
     xlab = "Battery Power", ylab = "RAM",
     main = "Battery Power vs RAM colored by Price Range")

# Add a legend
legend("topright", legend = levels(mobile_data$price_range),
       col = 1:4, pch = 1)

model_low <- lm(ram ~ battery_power, data = subset(mobile_data, price_range == "low"))
model_medium <- lm(ram ~ battery_power, data = subset(mobile_data, price_range == "medium"))
model_high <- lm(ram ~ battery_power, data = subset(mobile_data, price_range == "high"))
model_very_high <- lm(ram ~ battery_power, data = subset(mobile_data, price_range == "very high"))

# trend lines
abline(model_low, col = "red", lwd = 2)         # Trend line for low price range
abline(model_medium, col = "green", lwd = 2)    # Trend line for medium price range
abline(model_high, col = "blue", lwd = 2)       # Trend line for high price range
abline(model_very_high, col = "purple", lwd = 2) # Trend line for very high price range
```



(g)

```
subset_cores <- subset(mobile_data, n_cores %in% c(4, 6, 8))

average_clock_speed <- round(mean(subset_cores$clock_speed, na.rm = TRUE), 2)
median_clock_speed <- round(median(subset_cores$clock_speed, na.rm = TRUE), 2)

print(paste("Average Clock Speed:", average_clock_speed))
```

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

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

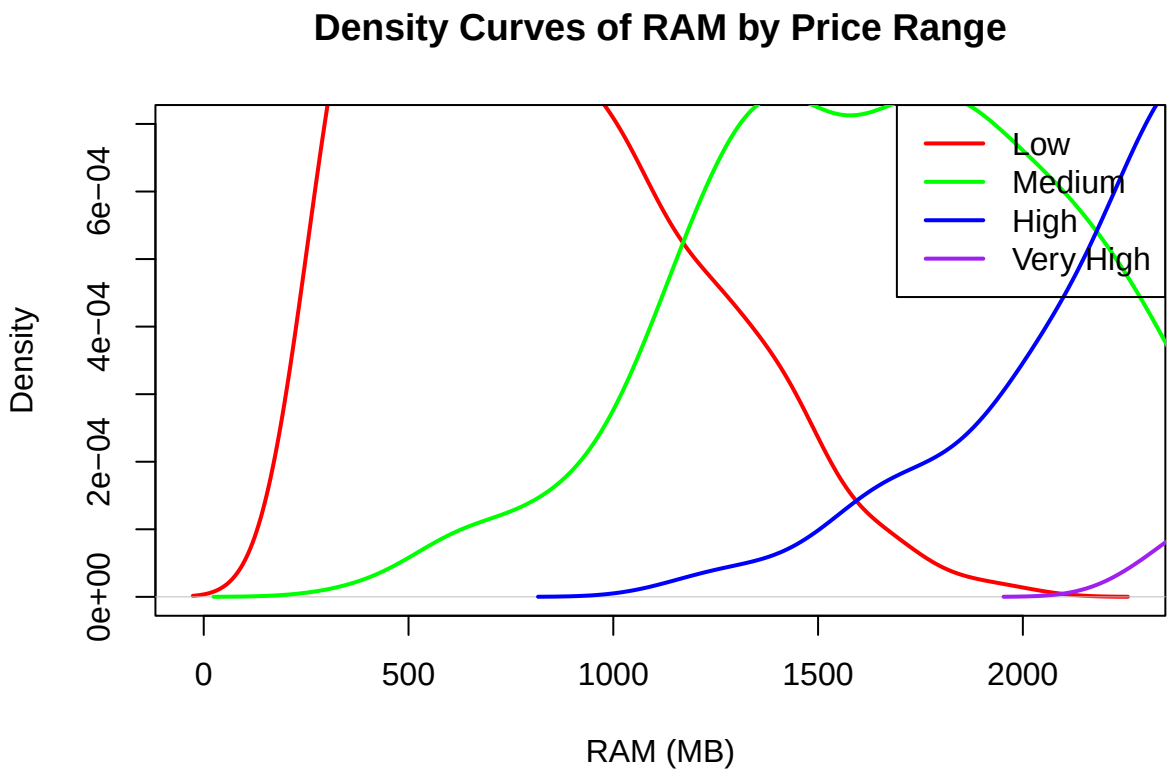
```
## [1] "Median Clock Speed: 1.5"
```

The average and median clock speeds for phones with 4, 6, and 8 cores are similar because these phones tend to have uniform, high-performance hardware. Since they are all modern, advanced devices, the clock speeds are consistent, with no major outliers affecting the results.

(h)

```
plot(density(subset(mobile_data, price_range == "low")$ram),
     col = "red", lwd = 2,
     main = "Density Curves of RAM by Price Range",
     xlab = "RAM (MB)", ylim = c(0, 0.0007))
```

```r
lines(density(subset(mobile_data, price_range == "medium")$ram), col = "green", lwd = 2)
lines(density(subset(mobile_data, price_range == "high")$ram), col = "blue", lwd = 2)
lines(density(subset(mobile_data, price_range == "very high")$ram), col = "purple", lwd = 2)

legend("topright", legend = c("Low", "Medium", "High", "Very High"),
       col = c("red", "green", "blue", "purple"), lwd = 2)
```



The curves representing RAM distribution across price ranges shift to the right as the price increases: low-cost phones (red) peak at lower RAM values, medium-cost phones (green) show moderate RAM, high-cost phones (blue) have significantly more RAM, and premium phones (purple) peak at the highest RAM values. Each curve illustrates the trend of increasing RAM availability with higher price categories.

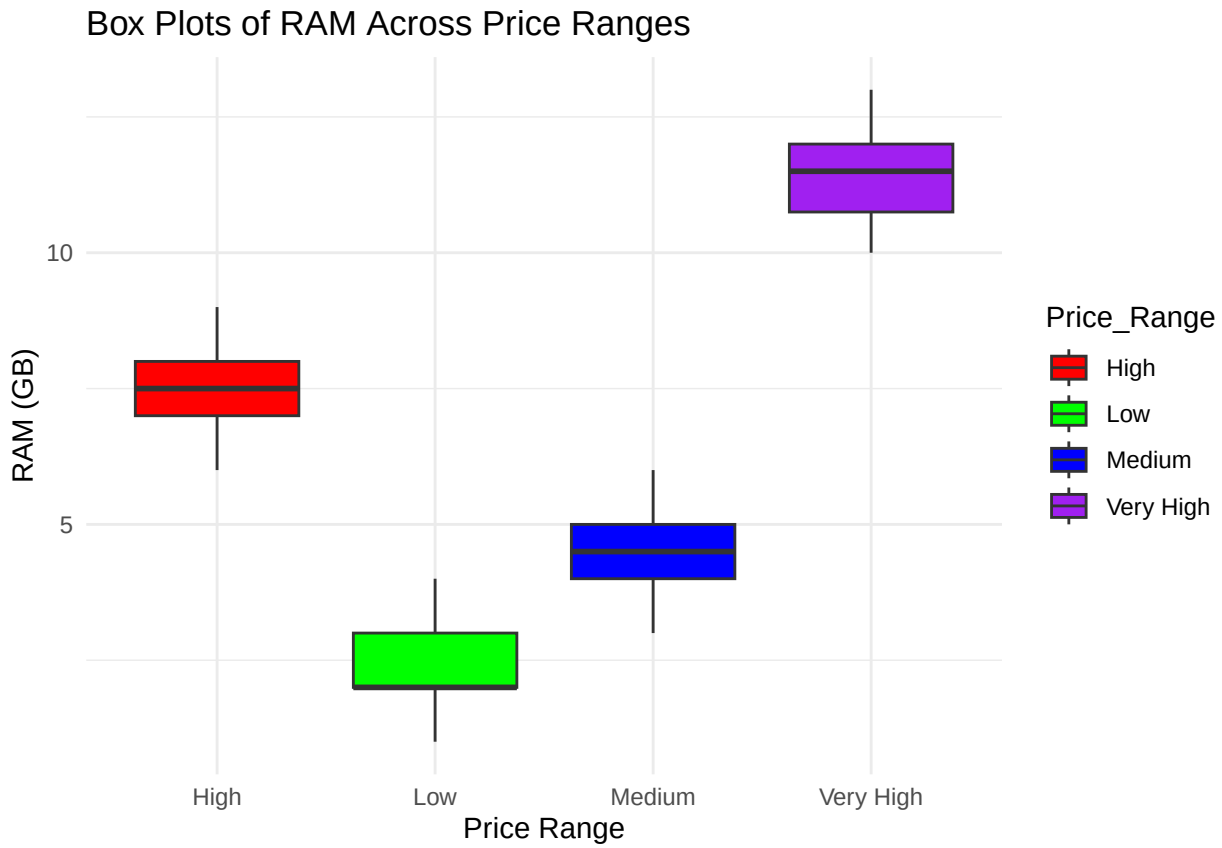
(i)

```r
options(repos = c(CRAN = "https://cloud.r-project.org/"))
install.packages("ggplot2")
```

```
##
## The downloaded binary packages are in
##  /var/folders/21/75_sfw8d6mb3b6s8mftqhsm00000gn/T//Rtmph4uw58/downloaded_packages
```

```r
library(ggplot2)
# Sample data
Low <- c(2, 3, 2, 4, 2, 3, 1, 2)
Medium <- c(4, 5, 6, 5, 4, 5, 3, 4)
```

```
High <- c(6, 8, 7, 8, 9, 7, 8, 7)
VeryHigh <- c(10, 12, 11, 13, 12, 11, 10, 12)

ram_data <- data.frame(
  RAM = c(Low, Medium, High, VeryHigh),
  Price_Range = rep(c("Low", "Medium", "High", "Very High"), each=length(Low))
)

ggplot(ram_data, aes(x = Price_Range, y = RAM, fill = Price_Range)) +
  geom_boxplot() +
  scale_fill_manual(values = c("red", "green", "blue", "purple")) +
  labs(title = "Box Plots of RAM Across Price Ranges",
       x = "Price Range",
       y = "RAM (GB)") +
  theme_minimal()
```



Low Price Range (Red): Peaks at lower RAM values, indicating limited RAM options. Medium Price Range (Green): Shows a moderate amount of RAM with a wider interquartile range. High Price Range (Blue): Displays a larger spread with higher median RAM availability. Very High Price Range (Purple): Peaks at the highest RAM values, indicating premium options.

(j)

```
install.packages("vioplot")
```

##

```
## The downloaded binary packages are in
##  /var/folders/21/75_sfw8d6mb3b6s8mftqhsm00000gn/T//Rtmph4uw58/downloaded_packages
```

```
library(vioplot)
```

```
## Loading required package: sm
```

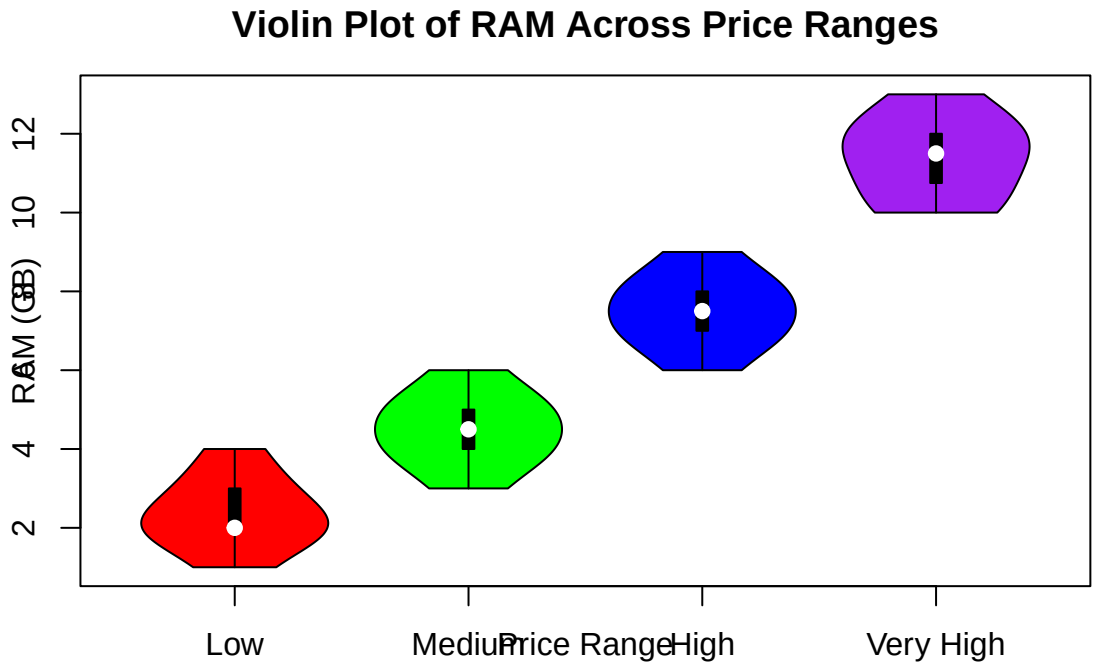
```
## Package 'sm', version 2.2-6.0: type help(sm) for summary information
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
# Sample data for RAM in different price ranges
Low <- c(2, 3, 2, 4, 2, 3, 1, 2)
Medium <- c(4, 5, 6, 5, 4, 5, 3, 4)
High <- c(6, 8, 7, 8, 9, 7, 8, 7)
VeryHigh <- c(10, 12, 11, 13, 12, 11, 10, 12)

ram_data <- list(Low, Medium, High, VeryHigh)
vioplot(ram_data,
        names = c("Low", "Medium", "High", "Very High"),
        col = c("red", "green", "blue", "purple"),
        main = "Violin Plot of RAM Across Price Ranges",
        xlab = "Price Range",
        ylab = "RAM (GB)")
```



The Low Price Range (Red) exhibits a narrow shape, peaking at lower RAM values, indicating that low-cost phones generally have limited RAM options with a concentrated distribution around 2-4 GB. In contrast, the Medium Price Range (Green) presents a wider shape, with a slight peak around moderate RAM values (4-6 GB), suggesting that mid-tier phones offer a broader range of RAM options compared to low-cost ones. The High Price Range (Blue) shows a wider violin with a peak shifted further right, reflecting increased RAM availability (6-9 GB), indicating that high-cost phones have substantially more RAM, catering to high-performance needs. Finally, the Very High Price Range (Purple) is the widest, peaking at the highest RAM values (10-13 GB), demonstrating that premium phones provide the most RAM, with a significant concentration at these elevated levels.

(k)

```
# Sample RAM data
ram_values <- c(2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)
ram_log2 <- round(log2(ram_values))
ram_factor <- as.factor(ram_log2)
data.frame(RAM = ram_values, Log2 = ram_log2, Factor = ram_factor)
```

```
##    RAM Log2 Factor
## 1    2    1      1
## 2    3    2      2
## 3    4    2      2
## 4    5    2      2
## 5    6    3      3
## 6    7    3      3
## 7    8    3      3
## 8    9    3      3
```

```
## 9   10    3      3
## 10  11    3      3
## 11  12    4      4
## 12  13    4      4
```

Taking the logarithm base 2 of RAM values and rounding to the nearest whole number normalizes the distribution and aligns the data with the common powers of 2 used in computing. This transformation converts continuous RAM values into discrete categories, enhancing interpretability and facilitating comparisons across different devices and price ranges.

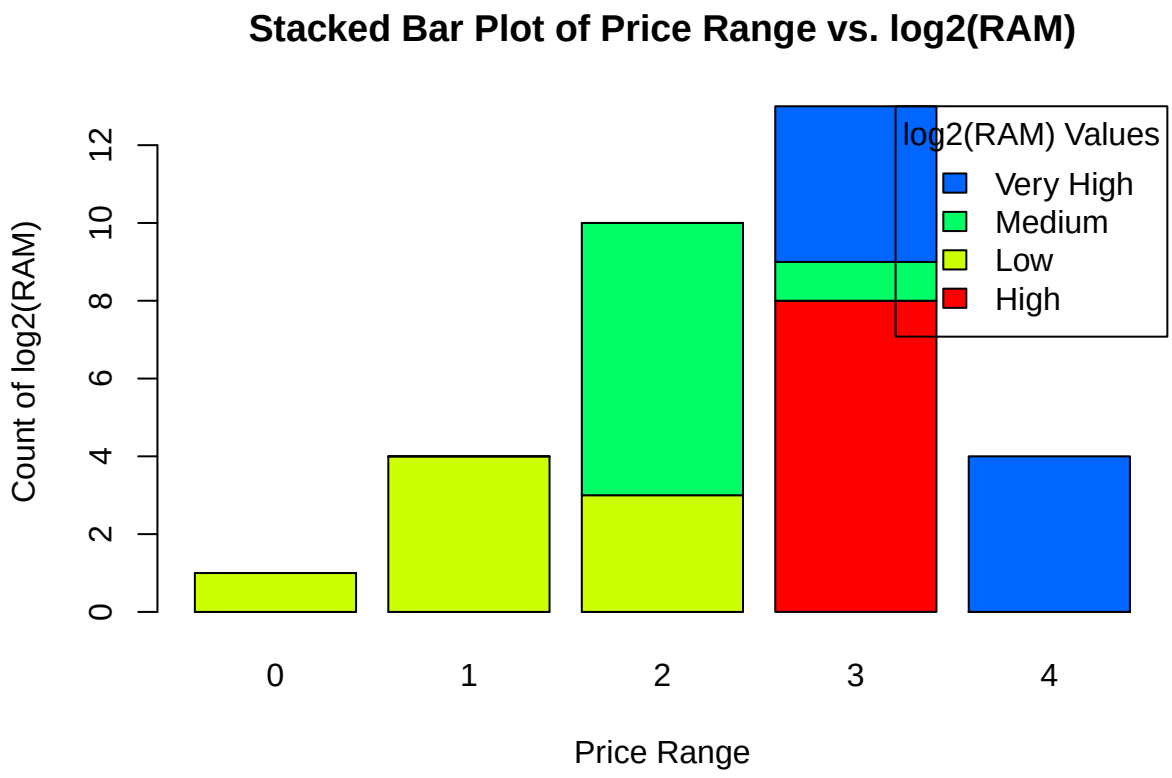
(l)

```r
ram_values_low <- c(2, 3, 2, 4, 2, 3, 1, 2)
ram_values_medium <- c(4, 5, 6, 5, 4, 5, 3, 4)
ram_values_high <- c(6, 8, 7, 8, 9, 7, 8, 7)
ram_values_very_high <- c(10, 12, 11, 13, 12, 11, 10, 12)


ram_values <- c(ram_values_low, ram_values_medium, ram_values_high, ram_values_very_high)
price_range <- c(rep("Low", length(ram_values_low)),
                 rep("Medium", length(ram_values_medium)),
                 rep("High", length(ram_values_high)),
                 rep("Very High", length(ram_values_very_high)))


data <- data.frame(
  RAM = ram_values,
  Price_Range = price_range
)
data$Log2_RAM <- round(log2(data$RAM))
print(data)
```

```
##    RAM Price_Range Log2_RAM
## 1    2         Low        1
## 2    3         Low        2
## 3    2         Low        1
## 4    4         Low        2
## 5    2         Low        1
## 6    3         Low        2
## 7    1         Low        0
## 8    2         Low        1
## 9    4      Medium        2
## 10   5      Medium        2
## 11   6      Medium        3
## 12   5      Medium        2
## 13   4      Medium        2
## 14   5      Medium        2
## 15   3      Medium        2
## 16   4      Medium        2
## 17   6        High        3
## 18   8        High        3
## 19   7        High        3
## 20   8        High        3
## 21   9        High        3
```

```
## 22   7         High         3
## 23   8         High         3
## 24   7         High         3
## 25  10    Very High         3
## 26  12    Very High         4
## 27  11    Very High         3
## 28  13    Very High         4
## 29  12    Very High         4
## 30  11    Very High         3
## 31  10    Very High         3
## 32  12    Very High         4
```

```r
count_data <- table(data$Price_Range, data$Log2_RAM)


barplot(count_data,
        beside = FALSE,
        col = rainbow(ncol(count_data)),  # Use rainbow colors for different log2(RAM) types
        main = "Stacked Bar Plot of Price Range vs. log2(RAM)",
        xlab = "Price Range",
        ylab = "Count of log2(RAM)",
        legend.text = TRUE,  # Include a legend
        args.legend = list(title = "log2(RAM) Values", x = "topright"))
```



Problem 2

(a)

```
options(repos = c(CRAN = "https://cloud.r-project.org/"))
# Install the package if you haven't done it before
install.packages("ggplot2")
```

```
##
## The downloaded binary packages are in
##  /var/folders/21/75_sfw8d6mb3b6s8mftqhsm00000gn/T//Rtmph4uw58/downloaded_packages
```

```
# Load the package
library(ggplot2)
data("mpg")
mpg$cyl <- factor(mpg$cyl, levels = c("4", "5", "6", "8"), ordered = TRUE)
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : Ord.factor w/ 4 levels "4"<"5"<"6"<"8": 1 1 1 1 3 3 3 1 1 1 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

(b)

```
options(repos = c(CRAN = "https://cloud.r-project.org/"))
library(ggplot2)
data("mpg")
unique(mpg$trans)
```

```
##  [1] "auto(l5)"   "manual(m5)" "manual(m6)" "auto(av)"   "auto(s6)"
##  [6] "auto(l4)"   "auto(l3)"   "auto(l6)"   "auto(s5)"   "auto(s4)"
```

```
mpg$trans <- factor(substr(mpg$trans, 1, 3), levels = c("aut", "man"), labels = c("auto", "manu"))
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : Factor w/ 2 levels "auto","manu": 1 2 2 1 1 2 1 2 1 2 ...
```

```
##  $ drv          : chr [1:234] "f" "f" "f" "f" ...
##  $ cty          : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy          : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl           : chr [1:234] "p" "p" "p" "p" ...
##  $ class        : chr [1:234] "compact" "compact" "compact" "compact" ...
```

```
unique(mpg$trans)
```

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

(c)

```
options(repos = c(CRAN = "https://cloud.r-project.org/"))
library(ggplot2)
data("mpg")
unique(mpg$drv)
```

```
## [1] "f" "4" "r"
```

```
mpg$drv <- factor(mpg$drv, levels = c("f", "r", "4"), ordered = TRUE)
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : Ord.factor w/ 3 levels "f"<"r"<"4": 1 1 1 1 1 1 1 3 3 3 ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

```
unique(mpg$drv)
```

```
## [1] f 4 r
## Levels: f < r < 4
```

(d)

```
options(repos = c(CRAN = "https://cloud.r-project.org/"))
library(ggplot2)
data("mpg")
unique(mpg$fl)
```

```
## [1] "p" "r" "e" "d" "c"
```

```r
mpg$fl <- factor(
  ifelse(mpg$fl %in% c("e", "c"), "other", mpg$fl),
  levels = c("gas", "diesel", "other"),
  labels = c("gasoline", "diesel", "other")
)
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : Factor w/ 3 levels "gasoline","diesel",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

```r
unique(mpg$fl)
```

```
## [1] <NA>  other
## Levels: gasoline diesel other
```

(e)

```r
options(repos = c(CRAN = "https://cloud.r-project.org/"))
library(ggplot2)
data("mpg")
unique(mpg$class)
```

```
## [1] "compact"    "midsize"    "suv"        "2seater"    "minivan"
## [6] "pickup"     "subcompact"
```

```r
mpg$class <- factor(mpg$class,
                    levels = c("2seater", "subcompact", "compact", "midsize", "suv", "minivan", "pickup
                    ordered = TRUE)
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : Ord.factor w/ 7 levels "2seater"<"subcompact"<..: 3 3 3 3 3 3 3 3 3 3 ...
```

```r
unique(mpg$class)
```

```
## [1] compact    midsize    suv        2seater    minivan    pickup     subcompact
## 7 Levels: 2seater < subcompact < compact < midsize < suv < ... < pickup
```

(f)

```r
options(repos = c(CRAN = "https://cloud.r-project.org/"))
library(ggplot2)
data("mpg")
mpg$country <- NA  # Initialize the country variable with NA
mpg$country[mpg$manufacturer %in% c("chevrolet", "dodge", "ford", "jeep", "lincoln", "mercury", "pontia
mpg$country[mpg$manufacturer %in% c("honda", "nissan", "subaru", "toyota")] <- "japan"
mpg$country[mpg$manufacturer %in% c("audi", "volkswagen")] <- "germany"
mpg$country[mpg$manufacturer == "hyundai"] <- "south korea"
mpg$country[mpg$manufacturer == "land rover"] <- "great britain"
str(mpg)
```

```
## tibble [234 x 12] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
##  $ country     : chr [1:234] "germany" "germany" "germany" "germany" ...
```

```r
head(mpg)
```

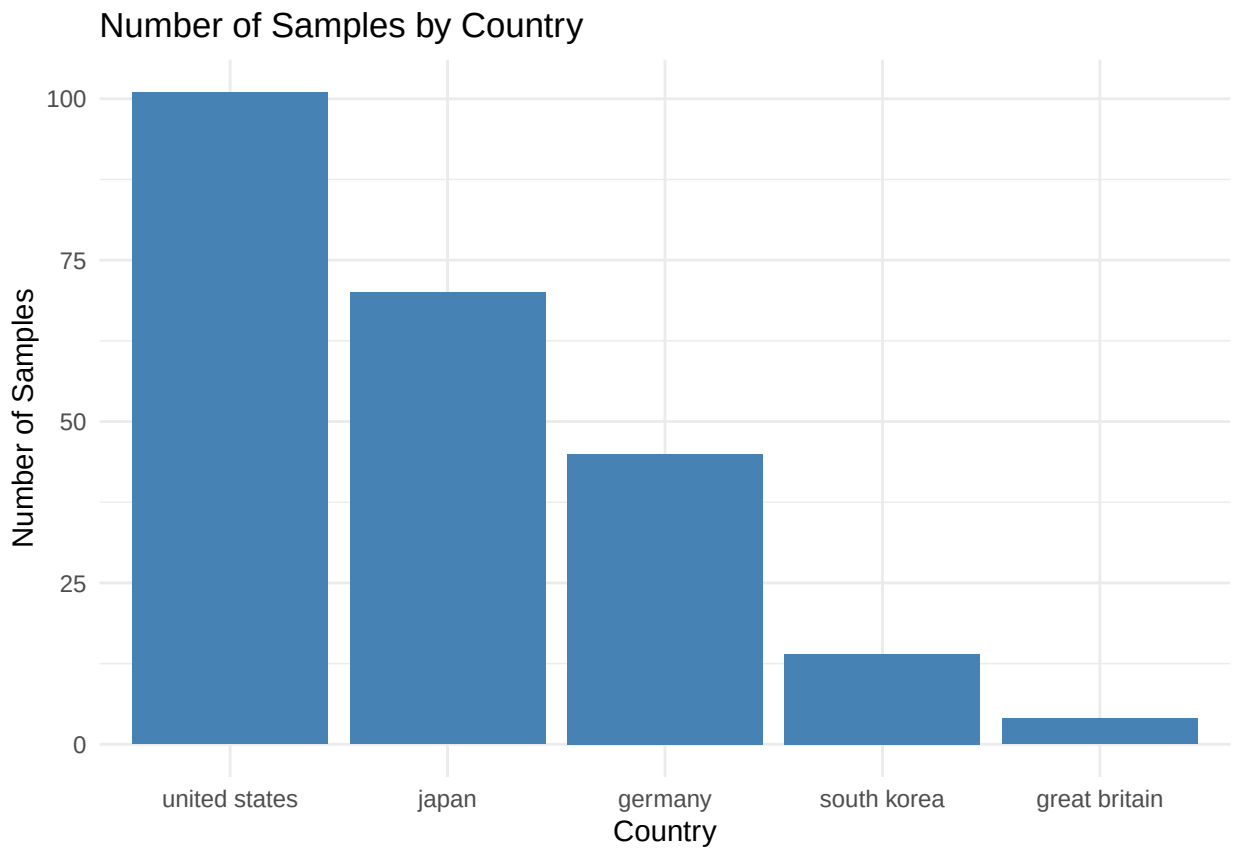
```
## # A tibble: 6 x 12
##   manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
## 2 audi         a4      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~
## # i 1 more variable: country <chr>
```

(g)

```r
options(repos = c(CRAN = "https://cloud.r-project.org/"))

library(ggplot2)
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
data("mpg")
mpg$country <- NA  # Initialize the country variable with NA
mpg$country[mpg$manufacturer %in% c("chevrolet", "dodge", "ford", "jeep", "lincoln", "mercury", "pontia
mpg$country[mpg$manufacturer %in% c("honda", "nissan", "subaru", "toyota")] <- "japan"
mpg$country[mpg$manufacturer %in% c("audi", "volkswagen")] <- "germany"
mpg$country[mpg$manufacturer == "hyundai"] <- "south korea"
mpg$country[mpg$manufacturer == "land rover"] <- "great britain"
country_counts <- mpg %>%
  group_by(country) %>%
  summarise(count = n(), .groups = "drop") %>%
  arrange(desc(count))

ggplot(country_counts, aes(x = reorder(country, -count), y = count)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Number of Samples by Country",
       x = "Country",
       y = "Number of Samples") +
  theme_minimal()
```




```
most_samples <- country_counts[1, ]
least_samples <- country_counts[nrow(country_counts), ]

cat("Country with the most samples:", most_samples$country, "with", most_samples$count, "samples.\n")
```

```
## Country with the most samples: united states with 101 samples.
```

```
cat("Country with the least samples:", least_samples$country, "with", least_samples$count, "samples.\n")
```

```
## Country with the least samples: great britain with 4 samples.
```

(h)

```
options(repos = c(CRAN = "https://cloud.r-project.org/"))
library(ggplot2)
data("mpg")
typical_displ <- as.numeric(names(sort(table(mpg$displ), decreasing = TRUE)[1]))
typical_cyl <- as.numeric(names(sort(table(mpg$cyl), decreasing = TRUE)[1]))
typical_trans <- names(sort(table(mpg$trans), decreasing = TRUE)[1])
typical_drv <- names(sort(table(mpg$drv), decreasing = TRUE)[1])
typical_fl <- names(sort(table(mpg$fl), decreasing = TRUE)[1])
typical_class <- names(sort(table(mpg$class), decreasing = TRUE)[1])

cat("Typical U.S. Car Summary:\n")
```

```
## Typical U.S. Car Summary:
```

```
cat("Engine Displacement:", typical_displ, "L\n")
```

```
## Engine Displacement: 2 L
```

```
cat("Number of Cylinders:", typical_cyl, "\n")
```

```
## Number of Cylinders: 4
```

```
cat("Type of Transmission:", typical_trans, "\n")
```

```
## Type of Transmission: auto(l4)
```

```
cat("Drive Type:", typical_drv, "\n")
```

```
## Drive Type: f
```

```
cat("Fuel Type:", typical_fl, "\n")
```

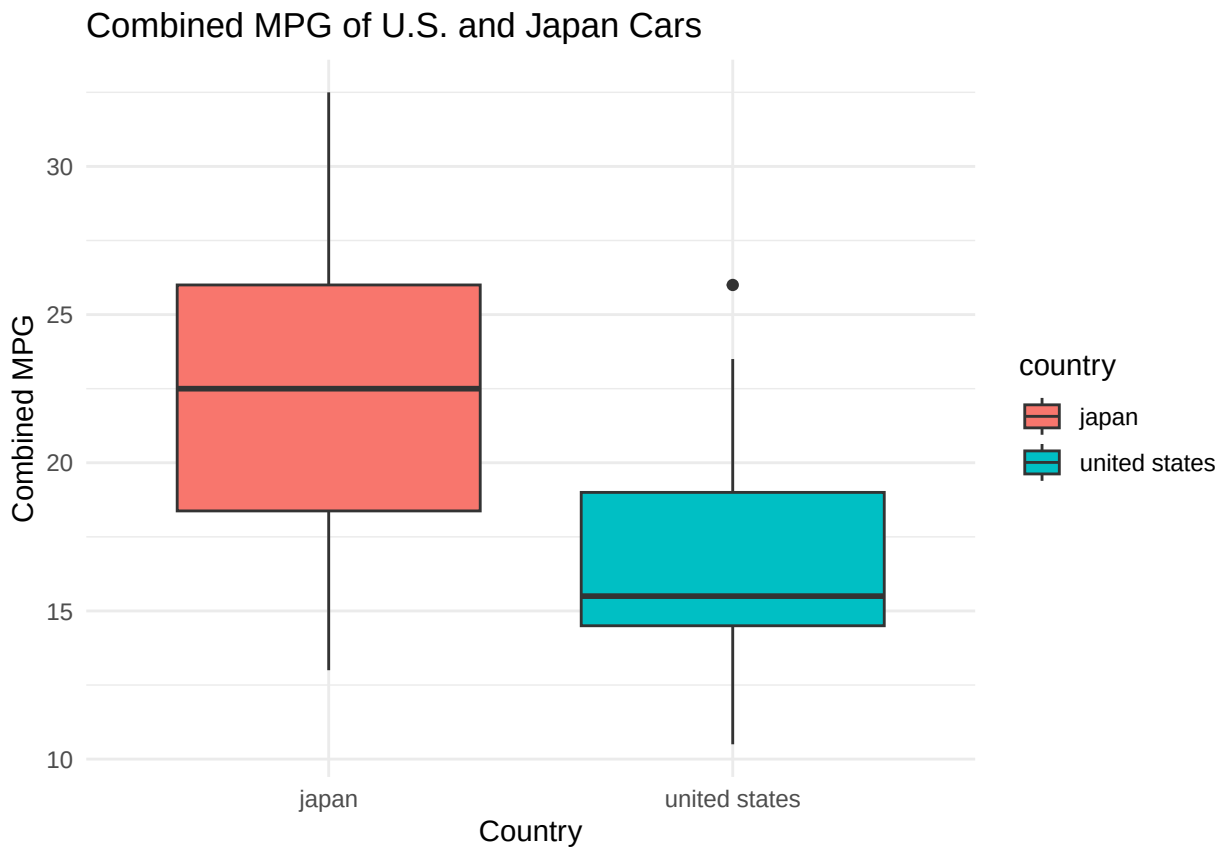
```
## Fuel Type: r
```

```
cat("Type of Car:", typical_class, "\n")
```

```
## Type of Car: suv
```

(i)

```
options(repos = c(CRAN = "https://cloud.r-project.org/"))

library(ggplot2)
library(dplyr)

data("mpg")
mpg$country <- NA  # Initialize the country variable with NA
mpg$country[mpg$manufacturer %in% c("chevrolet", "dodge", "ford", "jeep", "lincoln", "mercury", "pontia
mpg$country[mpg$manufacturer %in% c("honda", "nissan", "subaru", "toyota")] <- "japan"
mpg$country[mpg$manufacturer %in% c("audi", "volkswagen")] <- "germany"
mpg$country[mpg$manufacturer == "hyundai"] <- "south korea"
mpg$country[mpg$manufacturer == "land rover"] <- "great britain"
mpg$combined_mpg <- (mpg$cty + mpg$hwy) / 2

mpg_filtered <- mpg %>% filter(country %in% c("united states", "japan"))
ggplot(mpg_filtered, aes(x = country, y = combined_mpg, fill = country)) +
  geom_boxplot() +
  labs(title = "Combined MPG of U.S. and Japan Cars",
       x = "Country",
       y = "Combined MPG") +
  theme_minimal()
```



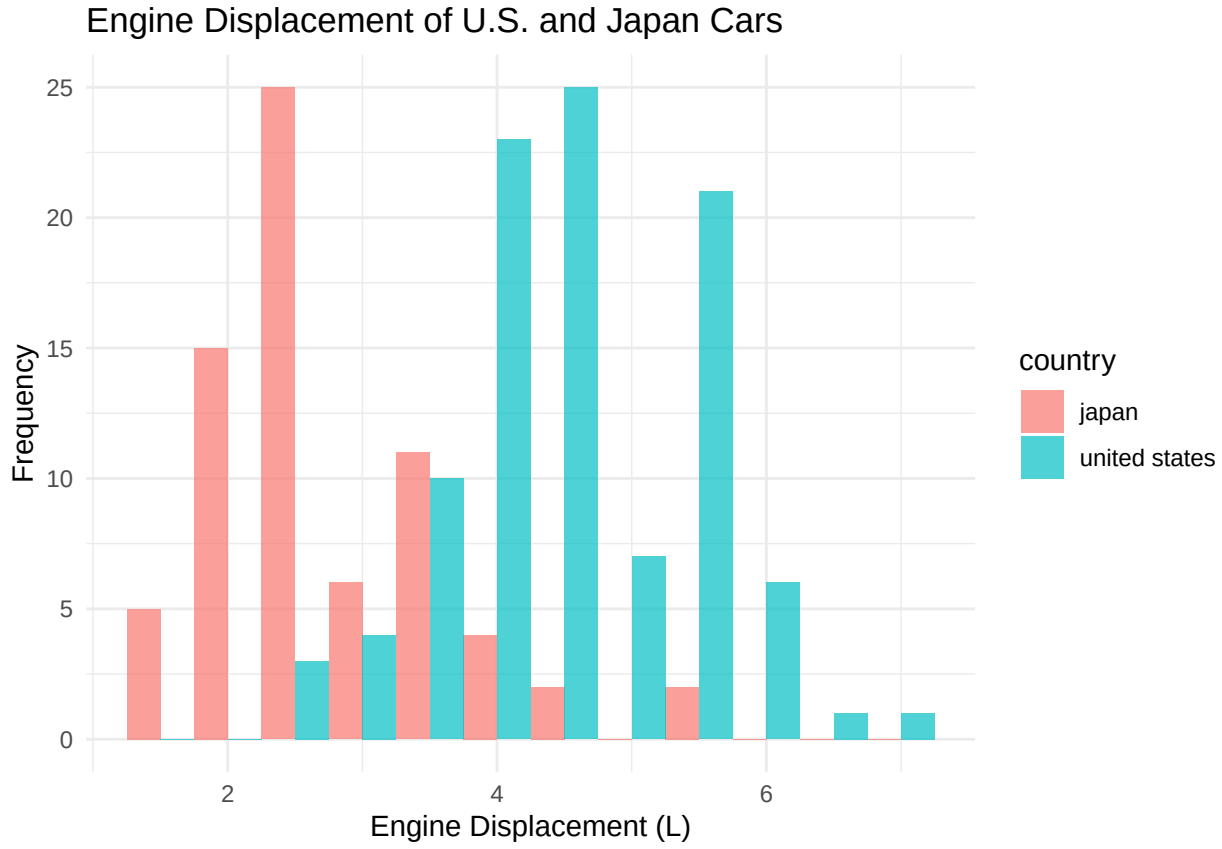

```
summary_stats <- mpg_filtered %>%
  group_by(country) %>%
  summarise(
    mean_mpg = mean(combined_mpg, na.rm = TRUE),
    median_mpg = median(combined_mpg, na.rm = TRUE),
    sd_mpg = sd(combined_mpg, na.rm = TRUE),
    IQR_mpg = IQR(combined_mpg, na.rm = TRUE),
    .groups = "drop"
  )

print(summary_stats)
```

```
## # A tibble: 2 x 5
##   country       mean_mpg median_mpg sd_mpg IQR_mpg
##   <chr>            <dbl>      <dbl>  <dbl>   <dbl>
## 1 japan             22.7       22.5   4.60    7.62
## 2 united states     16.6       15.5   3.30    4.5
```

(j)

```
options(repos = c(CRAN = "https://cloud.r-project.org/"))
library(ggplot2)
library(dplyr)
data("mpg")
```

```r
mpg$country <- NA  # Initialize the country variable with NA


mpg$country[mpg$manufacturer %in% c("chevrolet", "dodge", "ford", "jeep", "lincoln", "mercury", "pontia
mpg$country[mpg$manufacturer %in% c("honda", "nissan", "subaru", "toyota")] <- "japan"
mpg$country[mpg$manufacturer %in% c("audi", "volkswagen")] <- "germany"
mpg$country[mpg$manufacturer == "hyundai"] <- "south korea"
mpg$country[mpg$manufacturer == "land rover"] <- "great britain"

mpg_filtered <- mpg %>% filter(country %in% c("united states", "japan"))

ggplot(mpg_filtered, aes(x = displ, fill = country)) +
  geom_histogram(binwidth = 0.5, position = "dodge", alpha = 0.7) +
  labs(title = "Engine Displacement of U.S. and Japan Cars",
       x = "Engine Displacement (L)",
       y = "Frequency") +
  theme_minimal()
```



U.S. Cars: The histogram for U.S. cars typically shows a right-skewed distribution, with a higher frequency of cars having larger engine displacements, particularly around 3.0 to 5.0 L. There may be a significant number of cars with displacements over 5.0 L, indicating the presence of larger vehicles like SUVs and trucks. Japanese Cars: The histogram for Japanese cars often displays a more uniform or slightly left-skewed distribution, with most cars concentrated around 1.5 to 3.0 L. This suggests a preference for smaller, more fuel-efficient engines, common in sedans and compact cars.