Mobile Data Analytics

2024-03-01

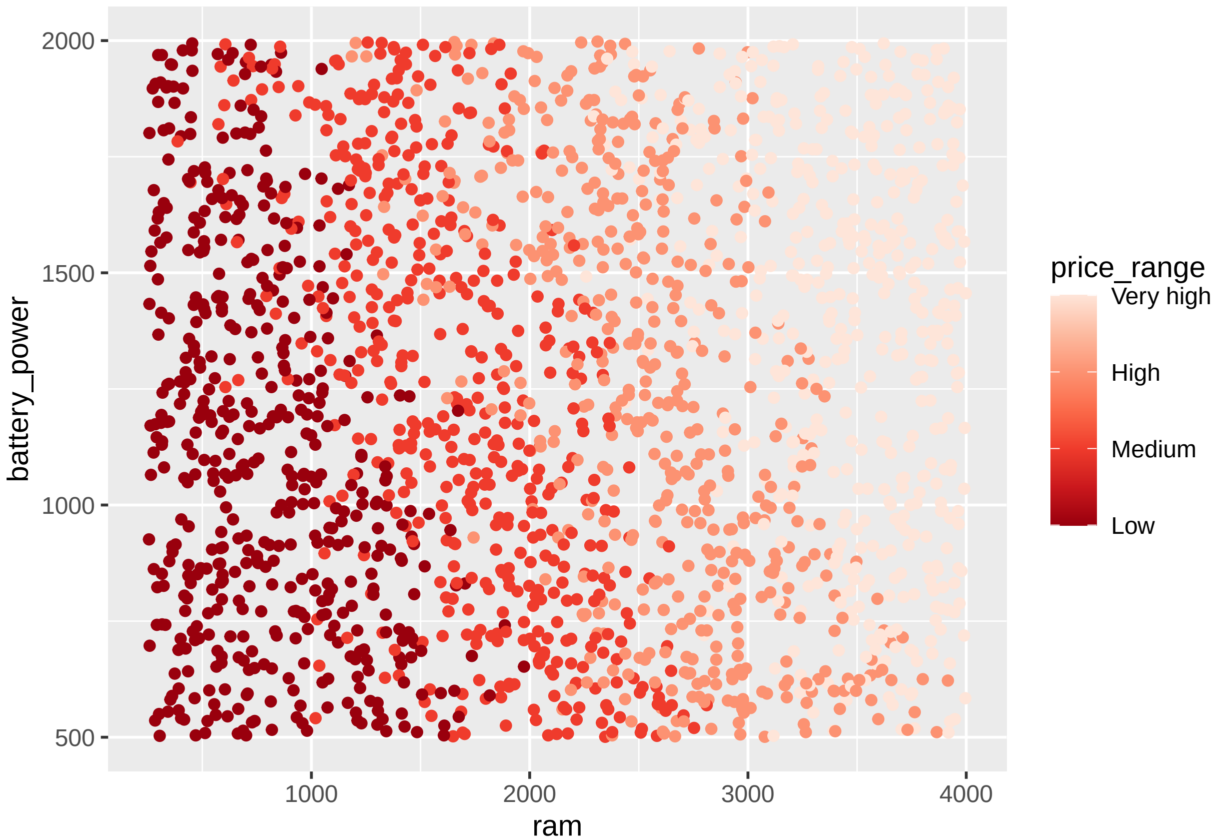
```r
# Question 1
mobile_data <- read.csv(file = "/Users/mihuynh/Downloads/Train Data Set/train.csv")
str(mobile_data)
```

```
## '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 ...
```

```r
# Turn the variable price range into a factor variable with levels:
# "0" for low, "1" for medium, "2" for high, and "3" for very high.
price_range <- factor(x = mobile_data$price_range, levels = c("0", "1", "2", "3"), labels = c("Low", "Me
str(price_range)
```

```
##  Factor w/ 4 levels "Low","Medium",..: 2 3 3 3 2 2 4 1 1 1 ...
```

```r
# Make a scatter plot between the variables battery power vs ram.
# Add colors based on price range.
library(ggplot2)
ggplot(data = mobile_data) +
  geom_point(aes(x = ram, y = battery_power, color = price_range)) +
  scale_color_distiller(palette = "Reds", labels = c("Low", "Medium", "High", "Very high"))
```




```r
# Find the Pearson correlation between the variables
# ram and battery power.
pearson <- cor(mobile_data$ram, mobile_data$battery_power, method = c("pearson"))
print(pearson)
```

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

```r
# Create four separate data sets by sub-setting the "mobile data"
# using the variable price range as
# "priceLow", "priceMedium", "priceHigh" and "priceVeryhigh".
priceLow <- subset(mobile_data, price_range == 0)
priceMedium <- subset(mobile_data, price_range == 1)
priceHigh <- subset(mobile_data, price_range == 2)
priceVeryhigh <- subset(mobile_data, price_range == 3)


# Calculate the Pearson correlation coefficient
# between the variable pair (ram , battery power) separately
# for each price range. Explain any correlations
# you might find in terms of how a cellphone operates.
LowCor <- cor(priceLow$ram, priceLow$battery_power, method = c("pearson"))
MedCor <- cor(priceMedium$ram, priceMedium$battery_power, method = c("pearson"))
HighCor <- cor(priceHigh$ram, priceHigh$battery_power, method = c("pearson"))
VeryhighCor <- cor(priceVeryhigh$ram, priceVeryhigh$battery_power, method = c("pearson"))
print(LowCor)
```

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

```r
print(MedCor)
```

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

```r
print(HighCor)
```

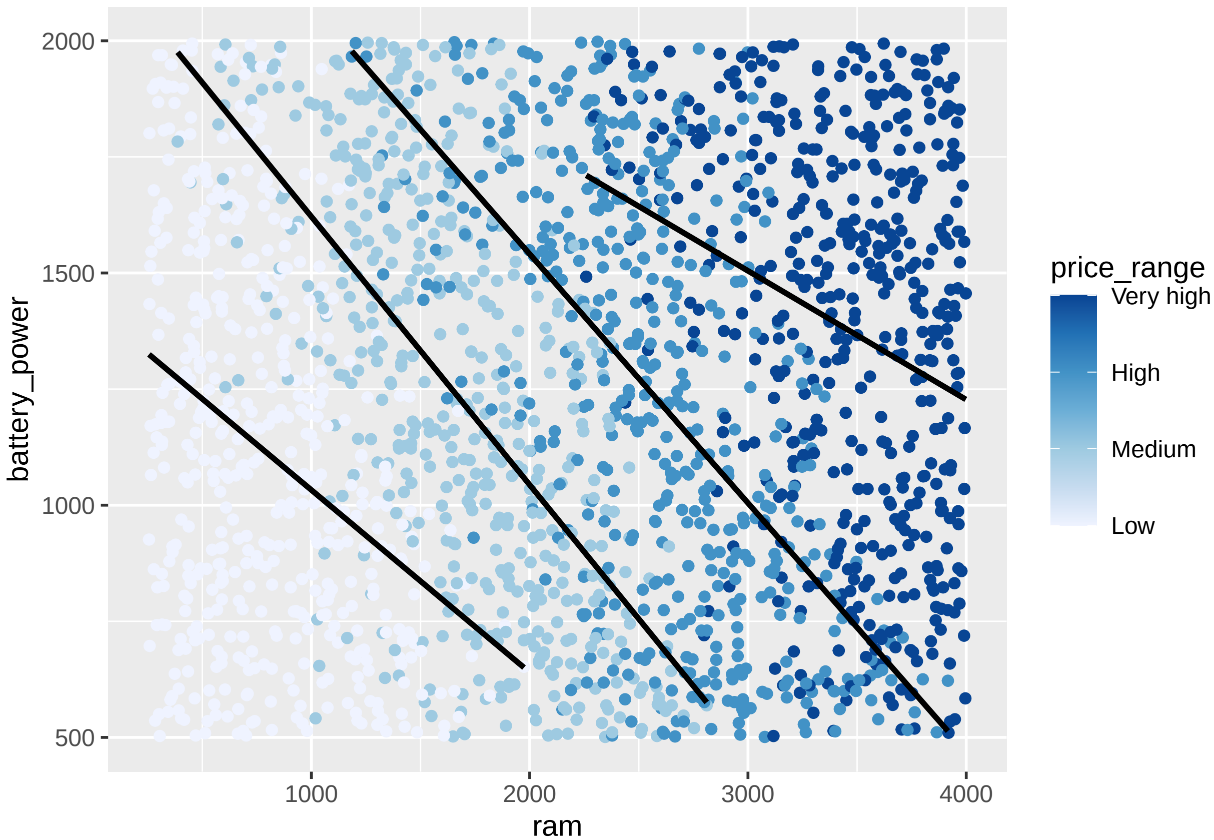
```
## [1] -0.5874086
```

```r
print(VeryhighCor)
```

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

```r
# Recreate the plot from Part (b), and add the trend lines
# for each price range separately.
ggplot(mobile_data, aes(x = ram, y = battery_power, color = price_range)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, aes(group = price_range), color = "black") +
  scale_color_distiller(palette = "Blues", direction = 1, labels = c("Low", "Medium", "High", "Very high
```

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



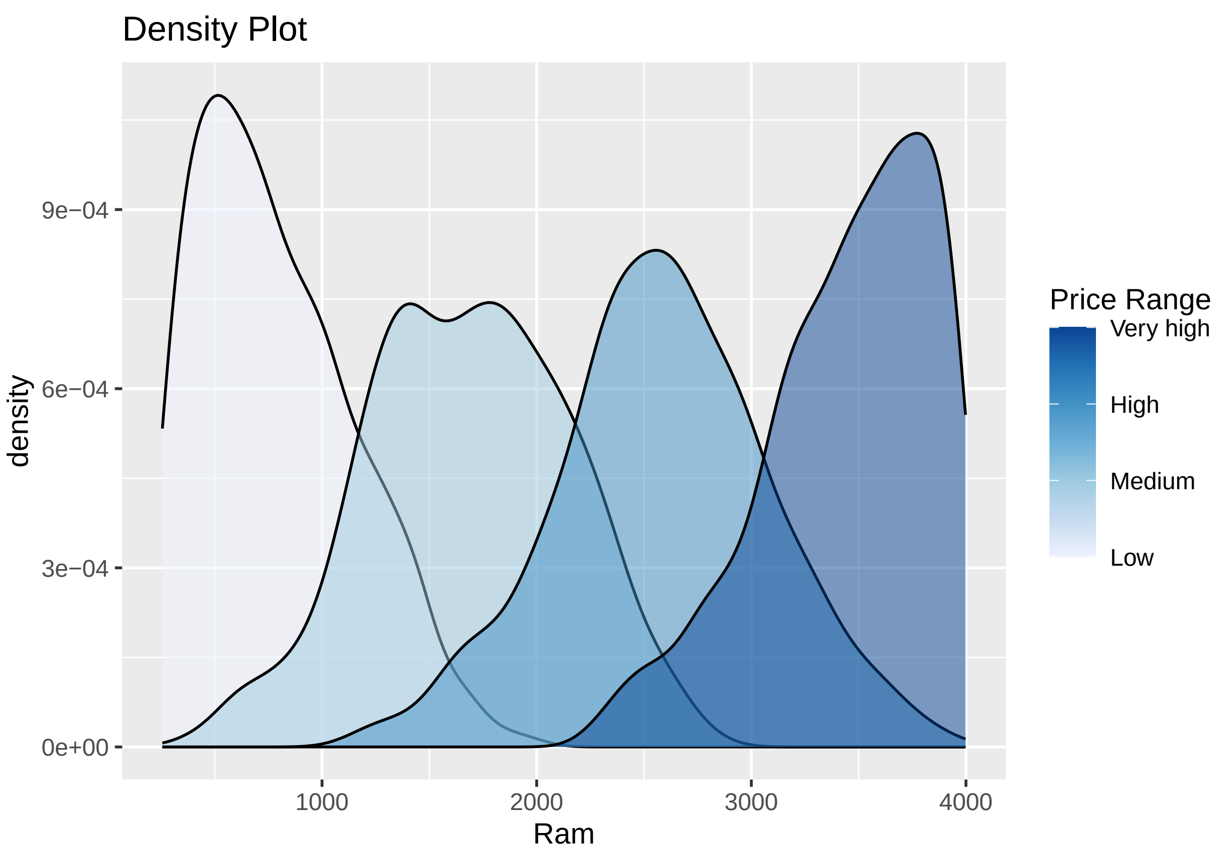

```r
# Find the average and the medium clock speed of the
# mobile phones which has 4, 6 and 8 cores in their
# processors. Round your answer to two decimal places.
filtered_data <- subset(mobile_data, n_cores %in% c(4, 6, 8))
avg_clock_speed <- mean(filtered_data$clock_speed)
median_clock_speed <- median(filtered_data$clock_speed)
round(avg_clock_speed, digits = 2)
```

```
## [1] 1.53
```

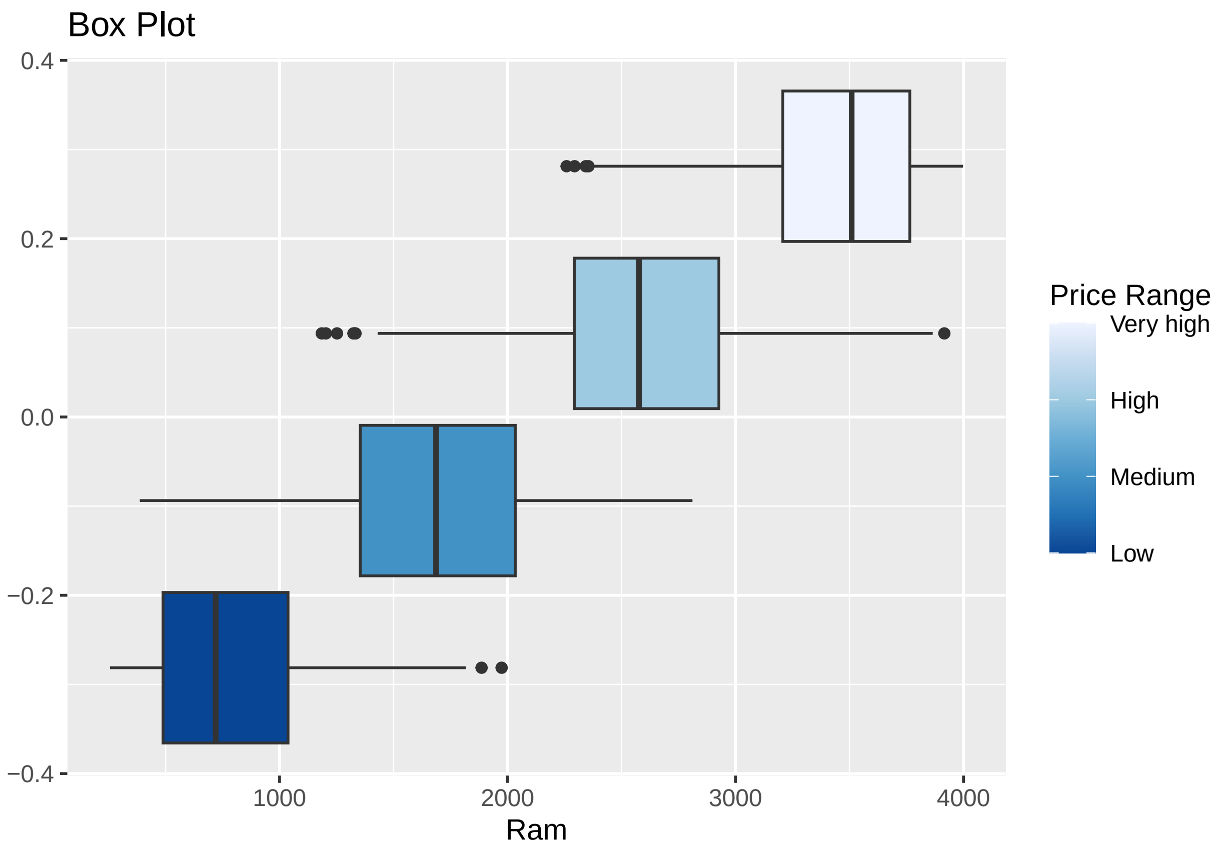
```r
round(median_clock_speed, digits = 2)
```

```
## [1] 1.5
```

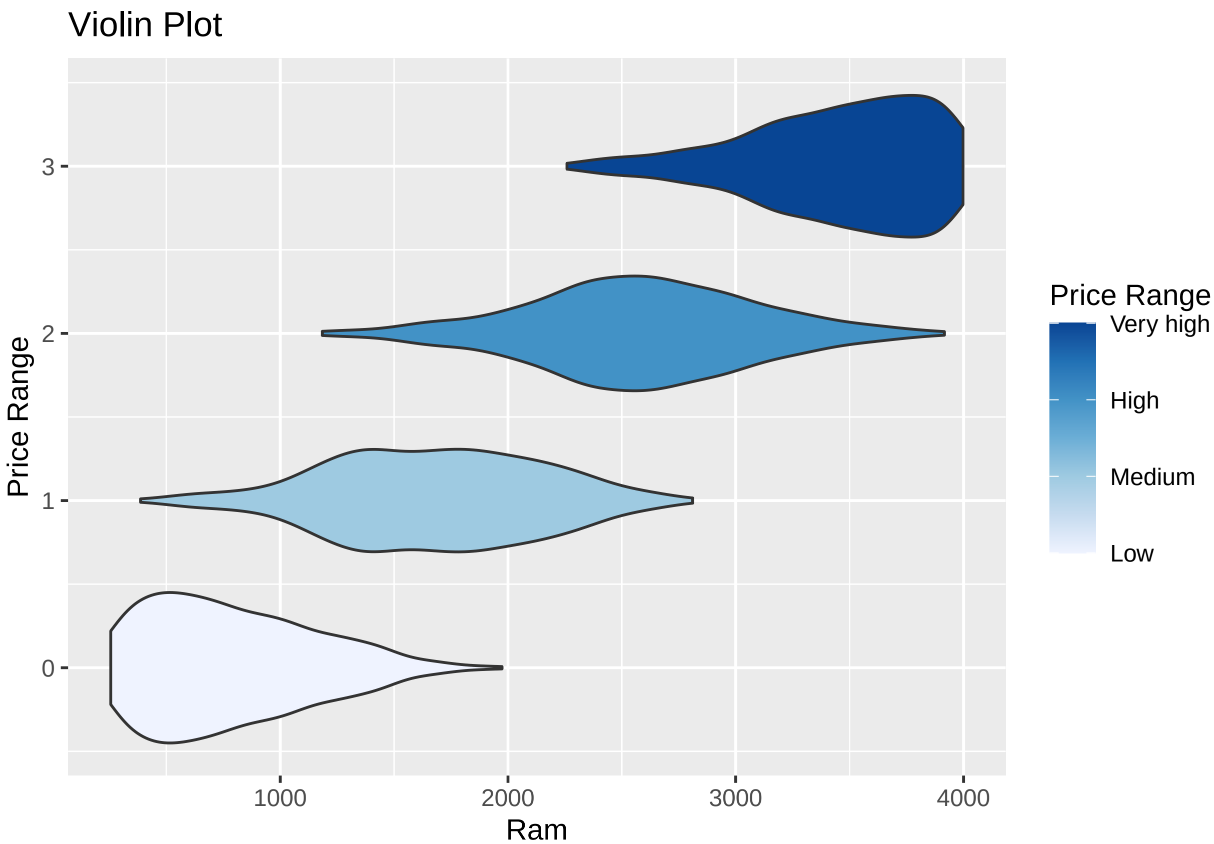
```r
# Make density curves of the ram where the 4 price ranges
# are in one plot and describe their shapes respectively.
ggplot(mobile_data, aes(x = ram, group = price_range, fill = price_range)) +
  geom_density(alpha = 0.5) +
  labs(title = "Density Plot", x = "Ram", fill = "Price Range") +
  scale_fill_distiller(palette = "Blues", direction = 1, labels = c("Low", "Medium", "High", "Very high
```




```r
# Make box plots of the ram where the 4 price ranges
# are in one plot and describe their shapes respectively
ggplot(mobile_data, aes(x = ram, group = price_range, fill = price_range)) +
  geom_boxplot() +
  labs(title = "Box Plot", x = "Ram", fill = "Price Range") +
  scale_fill_distiller(palette = "Blues", labels = c("Low", "Medium", "High", "Very high"))
```




```r
# Make a violin plot of the ram where the 4 price ranges
# are in one plot and describe their shapes respectively.
ggplot(mobile_data, aes(x = ram, y = price_range, group = price_range, fill = price_range)) +
  geom_violin() +
  labs(title = "Violin Plot", x = "Ram", y = "Price Range", fill = "Price Range") +
  scale_fill_distiller(palette = "Blues", direction = 1, labels = c("Low", "Medium", "High", "Very high
```



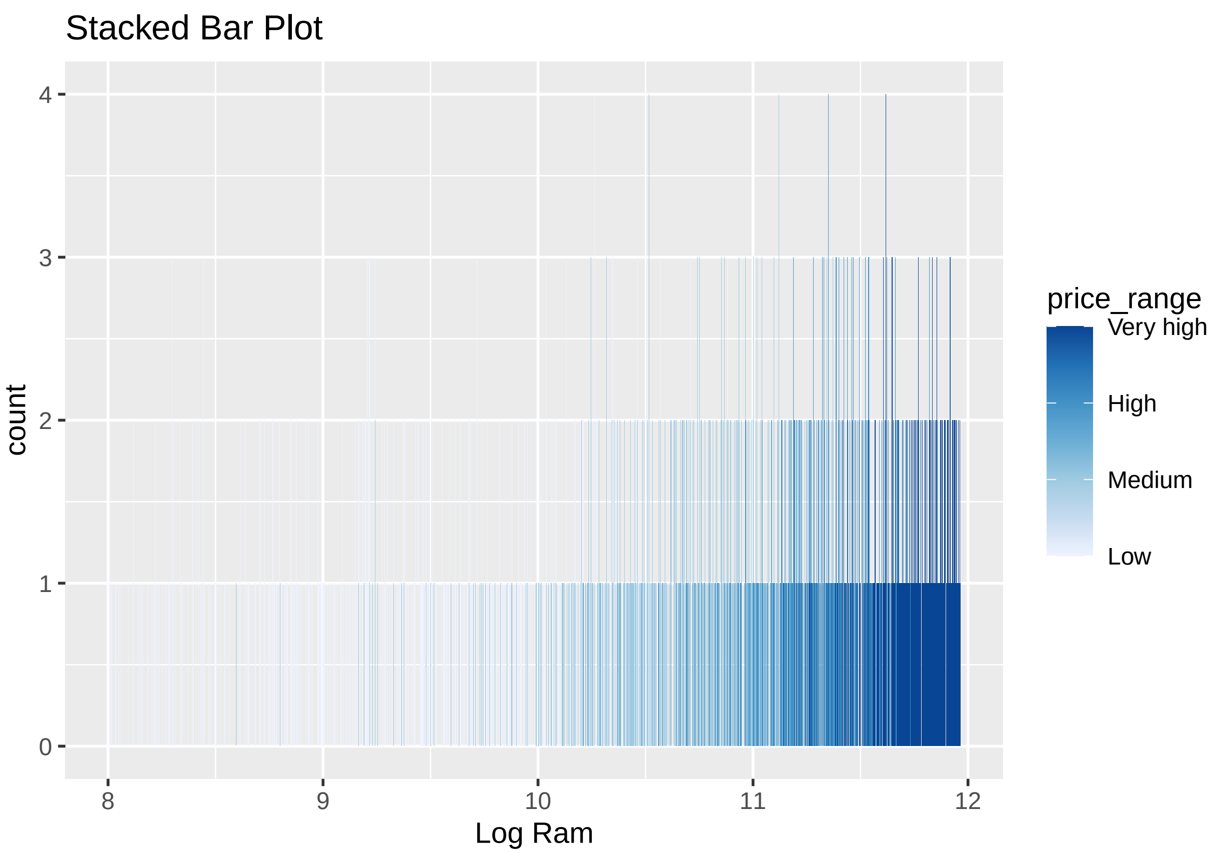

```r
# Make a factor variable out of ram by taking the log2 (ram)
# and rounding that value to the nearest whole number.
log_ram <- log2(mobile_data$ram)
round(log_ram)
```

```
##   [1] 11 11 11 11 10 10 12  9 10  9 12 12 11 11  9  9 12 12 11 11 11 12 10 10
##  [25] 11 11 12 12 11  9 12  9 11 12 10 10 11 12 12 11 11 12 11 11  9 10 12 11
##  [49] 10 12 11 11 12 10 12 11 10 12 11 12 11  9  9 11 12 12 12 12 10 12 12 11
##  [73]  9 12  9 11 10 11 12 11 12 11 12 11 11 10 10 12 11 11 11 11 10  8 10 12
##  [97] 10 11 10 11 11 10 11 10 10 11 11 11 12 11 11 11 10 12 12 11 12 11  9 10
## [121] 11 12 12 12 11 12 11 11 12  9 12 10 12 11 11 11 11 12  8 10 10  9 12 11
## [145]  8 12 11 12 11 12 11 12 10 11  9 10  9  9 12 10 12 12  9  9 11 12 11 11
## [169] 11 10 12 11 12 11 12 11 10 11 11 11 11 10 11 12 12 10 11 12 11 12 11 11
## [193] 12 12 11 11 11 12 11 10 10 10  9 11 11 11 12 10 12 12 11 11 10 11  9 12
## [217] 10 11 10 12  9  9 12 11 12 12 12 11 11 11 12 12  9 11 12  9 11 12 11  9
## [241] 12 11 12 11 12 12  8 11 11 10 12 11 11 10  9 12  9 11 10 12 10 11  9  9
## [265] 12  9 11 11 11 10  9 10 12 11 11 11 12 11  9  8 12 12 10 11 11 11 12 12
## [289] 11 11 12 12 10 11  9 11 12 10  9 11 12 10 10 12  8 10 10 12 10 11 12 11
## [313] 12  9 11 12  9  9  9 11 10 12 12  9 10 10 12 10 11 11 11 10 11 10 11 11
## [337] 12 10 12 11 12 11 11 12 10 10 12 11 11 12 11 11 10 11 12 10 11 10  9 12
## [361] 10  9 12 12 11 10 11 12 11 10 12 11 12  9 11 11  8 11  9 12 10 12 11 12
## [385] 12 12 12 11 11 12 12 11 12 12 10 12 12 10 11 12 11 12 12 11 11 12 12 11
## [409] 11 11  9 12 10 11 12 11 11 11 11 10 11 11 11 10 12 12  9 12 11 11 12 11
## [433] 11 10  9  9 12 10 12 10 10 10 12 10  8  9 11 11 11 11 11 11 12 10 12 11
## [457] 10 12 11 10 11 11 11 10  9 12 12  8 10 11 12 12 11 11 11 12 12 10 12 12
## [481]  9 12  9 12 11  9 11 11 10 11 10 12 12  8 11 11 12 11 11 12 10 12 10 10
## [505]  9 10 11 11 11 11 12 12  9 11 11 12 11 12 12 12 11 10 12 12 12 12 11 11
```

```
##  [529] 11 11 10 10 12 11 11 11 10 12 12 11  9 12 11 11 11 11 11 11 12 11 12 10
##  [553] 10 10  9 10 11 12 11 10 12 11 11 10 12  9 10  9 12 12 12 12 12 12 10 11
##  [577] 11 10 10 11 12 12 10 10 12  9 10 10 10  9 12 12 10 12 11  9 12 12  8  9
##  [601] 11 12  9 11 10 11 11 10 12 11 10 10 12 10 12 12 12 11 12 12 12  9 11 11
##  [625] 10 11  9 10 10  9 10 12 11 11 12 11 12 11 10 11 11 11 10 11  9 11 10 11
##  [649] 12 11 12 11 10 11 10 12 11 11 12 11 10  8  9 11 10  9 10 12 12 11 11 11
##  [673]  9 10 10 12 12 11 12 11 12 11 11 11 11 10 11 10  9 11 12  8 12 11 10 11
##  [697] 12 11 11  9  9 12 10 12 12 11 12 12 12  8 12 10 11 10 11 11  9 10 12 11
##  [721] 11 12 11 10 11 10 12 10 12 12  9 11 10 11 11 10  8 12 11 11 12 11 11 11
##  [745] 11 12 12 10 12 12 12 10 10 11 10 12  9 12  9 11  9  9 10 12 12 10 11 11
##  [769] 12 11 12  8 11 11  9  9 11 11 11 10 11 11 11 10 11 10 12 12 11 12  9 12
##  [793] 11 12 12 11 11  9 12 12 11  9  8 12 11 10 12 12 12 12  9 10  8 11  9 12
##  [817]  9 11 11 10  9 12 12 10 11 12 11 12 12 11 11 11  9 12 12 12 12 12 11 12
##  [841] 11 11 12 11 12  8 10 12  9  9 12 12 11  9 11 10 12 11 10 12 11 11 12  9
##  [865] 11 12 11 12 11 10 12 12 11  9 10  9  9 11 12 12  9 11 11 10 12 11 12  9
##  [889] 12 11 11 12 12 11 10 12 11 11 10 12 12 10 11 11 10 12 12  9  9 11 10 11
##  [913] 12 11 12 11 10 10 12 11  9 10  9 11 10 11 11 12 11 10 12 11 12 11 12 12
##  [937] 12 12 11 11 12  8 10 11  9 12  9  9 11 11 11 10 12 10 11 11 12 12 10 12
##  [961] 11 12 11  9 12 12  8  8 11 10 11  9 12 12  9 10 12 10 12 10 10 12 12 11
##  [985]  8  9 10 12 12 11 11 11 10  9  9 12 12 10 11 12 12 10 11  8 12 12 10  9
## [1009] 11 10 11 10 10 11  9 12 10 11 12 10 12 12 11  8 10  8 10 11 11 11 11 11
## [1033] 10 11  8  9 12 10 10 11  9 11 11 10 11 12 12 11 12 11 11 11 11 12 10 10
## [1057] 12 11 11 11 12 10 11  9  9 10 11 10  9  8  9 11 11 10 11 10 10 12 12 11
## [1081] 11 10 11 11 11 11 10 11  9 11 11 11 11 10  9 11 12 11 11  9 11 10  9 11
## [1105] 11 10  9 11 12 12 11 12 12 11 12 12  9 11 11 11 12 12 11 11 11 11 11 11
## [1129] 10 11 12 11 11 10  9 11 11 10 11 10 12 11 11 11 11 11 11 10 11 10 12 11
## [1153] 11 12 11 11 12  9 11  9 11 12 11 11 11 10 10 12 11 11 10 12 11 12 11 12
## [1177] 11 11  9 12 12 10  9  8 12 11 12 10 12 12 10 12 11  8 10 11 11 11 10 12
## [1201] 10 11 11 11 11 11 12 10 11 10 12  9 12 11 12 11 11 12 11 10 12 11 11 12
## [1225]  8  9 12 11  9 11  8  9 11 12 12  9  9 10 11 12  9 11 11 11 11 10 11  9
## [1249]  9 11 11 12 12  9  9 12 12 11 12 11  9 12 12 10 11 12 10 10 11 12 10 11
## [1273]  8 11 11 12 12 10 10 12 10 12 11 11 11 12  9 12 10 11 10 12 10 10  8 11
## [1297] 11 12 11  9 12 10  9 12 12 11 10 11 12 11 12 11 10 12  9 10 12 12 12 12
## [1321] 11 12 11 11 11 10 11 12 11 10 11 10 11 10 11 12 11 11 12 12 11 11 10 11
## [1345] 11 12 12 11  9 12 12 10 12 11 11 11 10 11 11 10 10  9  9 10 11 11 10 12
## [1369]  9 11 10 11 12  9 12 11 10 12  8 11 11 12 10 10 11 10 12 11 12  8 11 11
## [1393] 10  9 10  8 12  9 10 10 12 10 12 11  9 11 10 11  9 10 11 11 10 11 11 11
## [1417]  9 10 12 10  9 12 11 12 10 10 11 11 11 10 10 12 10 12  8 10 10 10  9 10
## [1441] 12 12 11  8 11 11  9 10 10 11  9 11 10 11 11 11 10 11 11 12 12 10 12 11
## [1465]  9 11 11 12 12 10 10 10 10 10 12 11 11 10 12 12 11 12 11 11 11  8 10 11
## [1489] 12  9 11 12  9 10 10 10  8 12 11 11 11 11 11 10 12  8 11 11 11 11 11 11
## [1513] 12  9 11 11 11 12 12 12 12 10 11  8 10 11 11 11 10 10  9 12 12  9 12 12
## [1537] 11 11 11  9 11 12 12 12 11 11 11 12 11 10 11 10 11 10 10  9 12 11 10 11
## [1561] 11 11 11 11  9 10  9 11 12 10  8 12 12 10 12 12  9 11 11 10  9  9  8 11
## [1585]  9 12 11 11 11 12 12 11 11 11 11 11 12  9 11 12 11 10 12 11  9 12 11 11
## [1609] 10 12 11 11 12 11 12 12 12 12 11 11 12 12 11 10 11 10 10 12 12 11  8 12
## [1633] 12 11 10 12 11 11 11 12 11 11 12 11 11 10 11 10 12 10  8 11 10 12  9 12
## [1657] 11 11 12 12 11 10 10 12 11 11 11 11 11 11 12  8 11 11 12 11 11 11 12 12
## [1681] 11 11 10 11 12 12 12  9 12 11 12 11 12 11 10 11 11 12  9 10  8 11 11  9
## [1705] 11 10 11 10 12 12  9 11 11 10 12  8 12 12 11  9 11 10 11 10 11 11 11 12
## [1729] 10 12 10 10 11 12 12 12 10 11 11 11 12 11 10 12 11  8 10 11 11 12 10 11
## [1753] 12  9 12 12 12 11  8 12 10 11 10 12 11 11 10 12 12 10 12 11 12 10 12 10
## [1777] 11 11 11 12 10 12 11 10 12 11 11 12 10 11 12 11 11 11 12 10 11  9 11 12
## [1801]  9 11 12 12  9 11 10 11  9 11  9 11  8  9 11 12 12 11  9 11  9 12 11 11
```

```
## [1825] 11 11 11 12 12  8 10 11 12 11 12 11 11 11 10 11 12 11  9 12 11 12 10 10
## [1849] 11 12 10 12 10  9  9  9 12 12 12 10  9 10 10 12 10 11 12 11 12 12 12 11
## [1873] 10 10 12 11  9 11 11 10 11 11  9 10 12 10 11 11  9 12 11  8  8 12 10 11
## [1897]  9 12 12 10 11 11 11 11 12 10 10 12 12 11 11  8 11  8 10 11 11  9 11 10
## [1921] 12 11 10 11 11 11 11 12 10 11 12 11 11 11 11 11 12 11 11 11 11 12 10 10
## [1945] 11 10 11 11 11 12 10 10 11 12 11 12  8 12 11 10 11 11 11 11 10 10 11  9
## [1969]  9 11 11 11 10 12 11 11  9 11 11 11 11 10 11 12 11 12 11 10 11 11  8 12
## [1993] 10 12 12  9 11 12 10 12
```

```
log_ram_factor <- as.factor(mobile_data$log_ram)

# Make a stacked bar plot to show the relationship between
# price range and log2(ram)
ggplot(mobile_data, aes(x = log_ram, fill = price_range, group = price_range)) +
  geom_bar() +
  labs(title = "Stacked Bar Plot", x = "Log Ram") +
  scale_fill_distiller(palette = "Blues", direction = 1, labels = c("Low", "Medium", "High", "Very high
```




```
# MPG DATASET

# Problem 2a
# Turn the variable cyl to an ordered factor variable with levels
# "4", "5", "6", and "8"
library(ggplot2)
data(mpg)
cyl <- factor(x = mpg$cyl, levels = c("4", "5", "6", "8"), ordered = is.ordered(c))
levels(cyl)
```

```
## [1] "4" "5" "6" "8"
```

```r
# Problem 2b
# Turn the variable trans to a factor variable,
# of which unique values are "auto" and "manu"
trans <- factor(substr(mpg$trans, 1, 4), levels = c("auto", "manu"))
levels(trans)
```

```
## [1] "auto" "manu"
```

```r
# Problem 2c
# Turn the variable drv to an ordered factor variable
# with levels "f", "r", and "4"
drv <- factor(mpg$drv, ordered = TRUE, levels = c("f", "r", "4"))
levels(drv)
```

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

```r
# Problem 2d
# Turn the variable fl to a factor variable, of
# which unique values are "gasoline", "diesel", and "other"
fl <- factor(ifelse(mpg$fl %in% c("d", "x"), "diesel",
                ifelse(mpg$fl %in% c("e", "c"), "other", "gasoline")))
levels(fl)
```

```
## [1] "diesel"   "gasoline" "other"
```

```r
# Problem 2e
# Turn the variable class to an ordered factor variable
# with levels "2seater", "subcompact", "compact",
# "midsize", "suv", "minivan", and "pickup"
class <- factor(mpg$class, ordered = TRUE, levels = c("2seater", "subcompact", "compact", "midsize", "su
levels(class)
```

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

```r
# Problem 2f
# Create a new variable of country to indicate the
# manufacturer base location
country_lookup <- data.frame(manufacturer = c("audi", "chevrolet", "dodge", "ford", "honda", "hyundai",
mpg <- merge(mpg, country_lookup, by.x = "manufacturer", by.y = "manufacturer", all.x = TRUE)
head(mpg)
```

```
##   manufacturer model displ year cyl      trans drv cty hwy fl   class country
## 1         audi    a4   1.8 1999   4   auto(l5)   f  18  29  p compact Germany
## 2         audi    a4   1.8 1999   4 manual(m5)   f  21  29  p compact Germany
## 3         audi    a4   2.0 2008   4 manual(m6)   f  20  31  p compact Germany
## 4         audi    a4   2.0 2008   4   auto(av)   f  21  30  p compact Germany
## 5         audi    a4   2.8 1999   6   auto(l5)   f  16  26  p compact Germany
## 6         audi    a4   2.8 1999   6 manual(m5)   f  18  26  p compact Germany
```

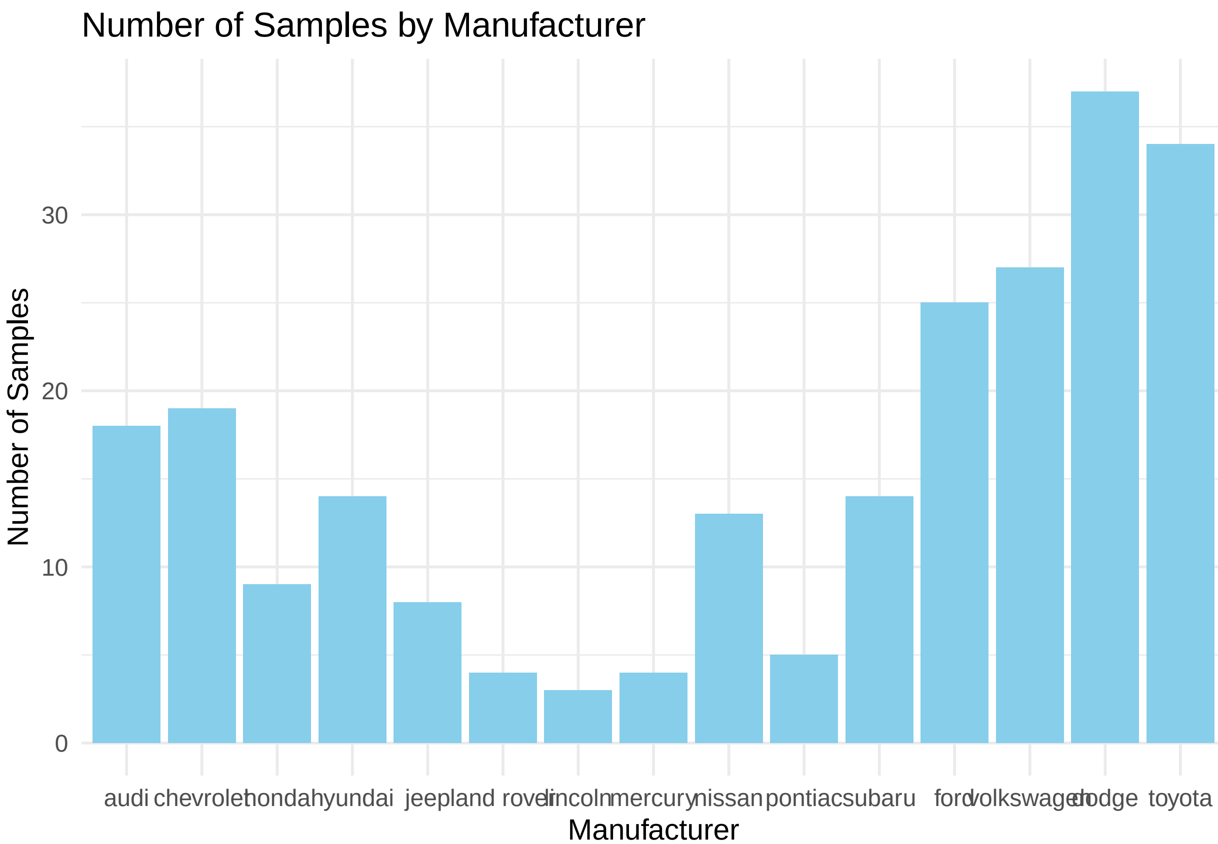
```r
# Problem 2g
# Draw a bar plot of the variable country and
# arrange the country in decreasing order in terms of the
# number of samples.
library(magrittr)
library(dplyr)
```

```
##
## 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
manufacturer_counts <- mpg %>%
  count(manufacturer) %>%
  arrange(desc(n))
mpg$manufacturer <- reorder(mpg$manufacturer, mpg$manufacturer, function(x) sum(x == manufacturer_counts
```

```
## Warning in x == manufacturer_counts$manufacturer: longer object length is not a
## multiple of shorter object length

## Warning in x == manufacturer_counts$manufacturer: longer object length is not a
## multiple of shorter object length

## Warning in x == manufacturer_counts$manufacturer: longer object length is not a
## multiple of shorter object length

## Warning in x == manufacturer_counts$manufacturer: longer object length is not a
## multiple of shorter object length

## Warning in x == manufacturer_counts$manufacturer: longer object length is not a
## multiple of shorter object length

## Warning in x == manufacturer_counts$manufacturer: longer object length is not a
## multiple of shorter object length

## Warning in x == manufacturer_counts$manufacturer: longer object length is not a
## multiple of shorter object length

## Warning in x == manufacturer_counts$manufacturer: longer object length is not a
## multiple of shorter object length

## Warning in x == manufacturer_counts$manufacturer: longer object length is not a
## multiple of shorter object length

## Warning in x == manufacturer_counts$manufacturer: longer object length is not a
## multiple of shorter object length

## Warning in x == manufacturer_counts$manufacturer: longer object length is not a
## multiple of shorter object length

## Warning in x == manufacturer_counts$manufacturer: longer object length is not a
## multiple of shorter object length

## Warning in x == manufacturer_counts$manufacturer: longer object length is not a
## multiple of shorter object length
```

```r
ggplot(mpg, aes(x = manufacturer)) +
  geom_bar(fill = "skyblue") +
  labs(title = "Number of Samples by Manufacturer", x = "Manufacturer", y = "Number of Samples") +
  theme_minimal()
```

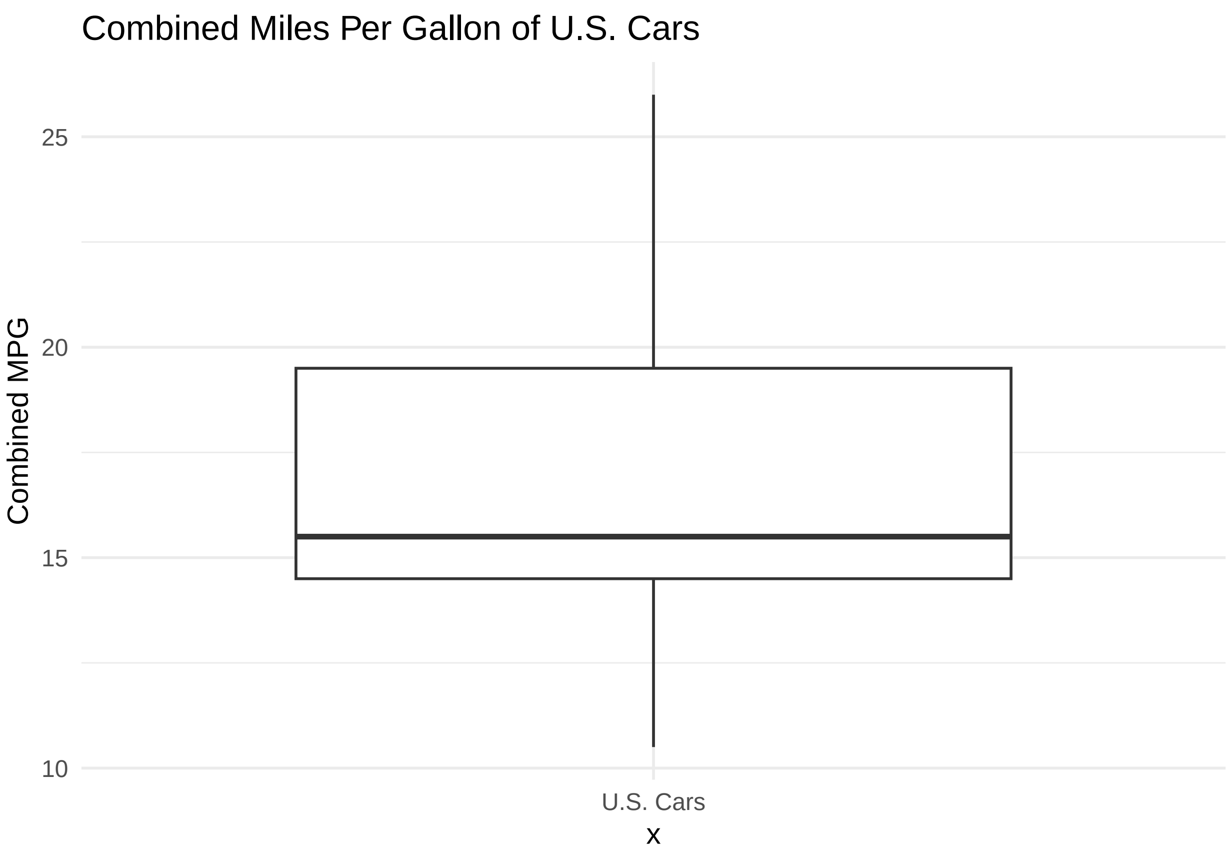



```r
# Problem 2h
# Summarize what a typical U.S. car looks like,
# in terms of engine displacement (i.e. displ), number of
# cylinders (i.e. cyl), type of transmission (i.e. trans),
# drive type (i.e. drv), fuel type (i.e. fl), and type
# of car (i.e. class)?
us_cars <- subset(mpg, manufacturer == "ford" | manufacturer == "chevrolet" | manufacturer == "dodge" |
summary_us_cars <- summary(us_cars[, c("displ", "cyl", "trans", "drv", "fl", "class")])
print(summary_us_cars)
```

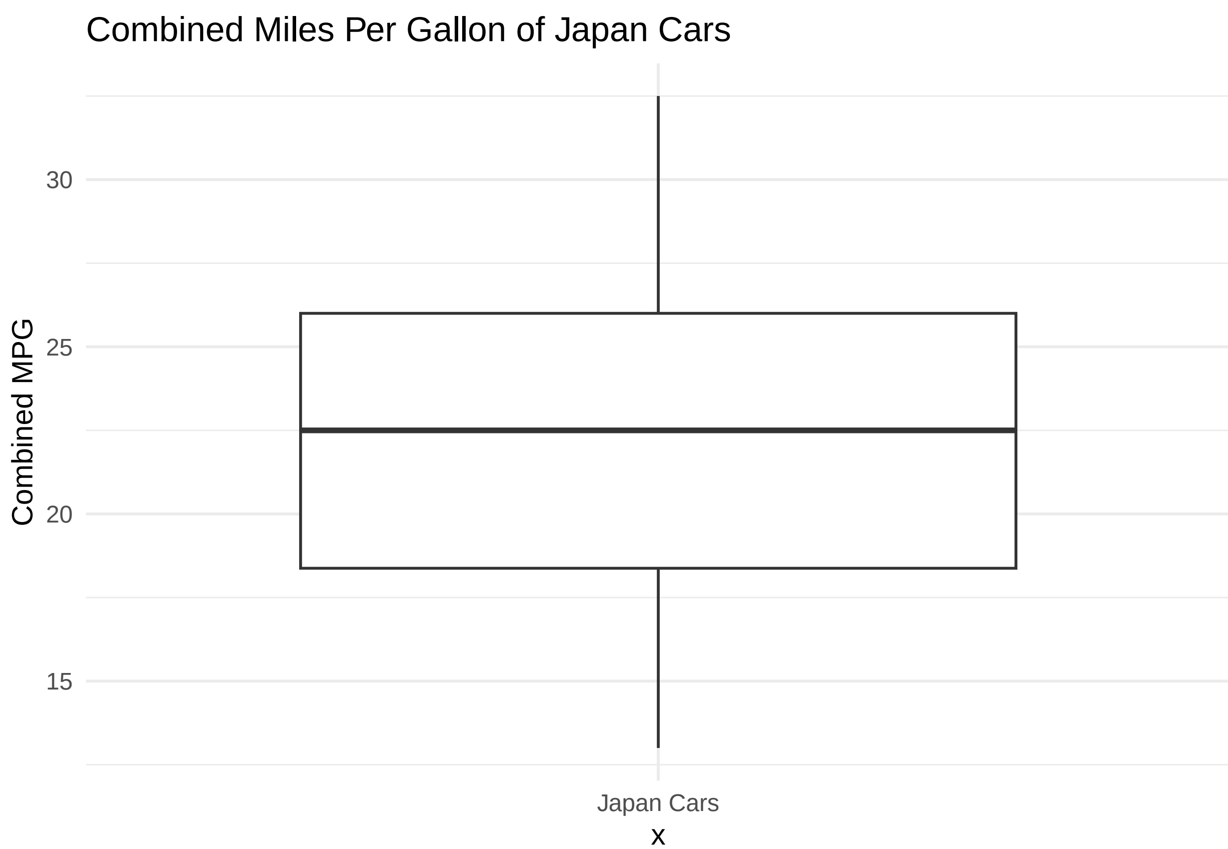
```
##      displ            cyl           trans               drv
##  Min.   :2.400   Min.   :4.00   Length:93          Length:93
##  1st Qu.:3.900   1st Qu.:6.00   Class :character   Class :character
##  Median :4.600   Median :8.00   Mode  :character   Mode  :character
##  Mean   :4.572   Mean   :7.14
##  3rd Qu.:5.300   3rd Qu.:8.00
##  Max.   :7.000   Max.   :8.00
##       fl               class
##  Length:93          Length:93
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
```

```
##
##
```

```
# Problem 2i
# Make a boxplot of the combined miles per gallon
# (i.e. (cty + hwy)/2) of U.S. cars and Japan cars,
# respectively, and report their means, medians,
# standard deviations, and IQRs.
mpg$combined_mpg <- (mpg$cty + mpg$hwy) / 2
us_cars <- subset(mpg, manufacturer %in% c("ford", "chevrolet", "dodge", "mercury", "pontiac", "lincoln
japan_cars <- subset(mpg, manufacturer %in% c("honda", "toyota", "nissan", "subaru", "mazda", "mitsubis
ggplot(mapping = aes(x = "U.S. Cars", y = combined_mpg)) +
  geom_boxplot(data = us_cars) +
  labs(title = "Combined Miles Per Gallon of U.S. Cars",
       y = "Combined MPG") +
  theme_minimal()
```




```
ggplot(mapping = aes(x = "Japan Cars", y = combined_mpg)) +
  geom_boxplot(data = japan_cars) +
  labs(title = "Combined Miles Per Gallon of Japan Cars",
       y = "Combined MPG") +
  theme_minimal()
```




```r
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)
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)
cat("Summary statistics for U.S. cars: \"")
```

```
## Summary statistics for U.S. cars: "
```

```r
cat("Mean: ", us_mean, "")
```

```
## Mean:  16.73118
```

```r
cat("Median: ", us_median, "")
```

```
## Median:  15.5
```

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

```
## Standard Deviation:  3.335057
```

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

```
## Interquartile Range (IQR):  5
```

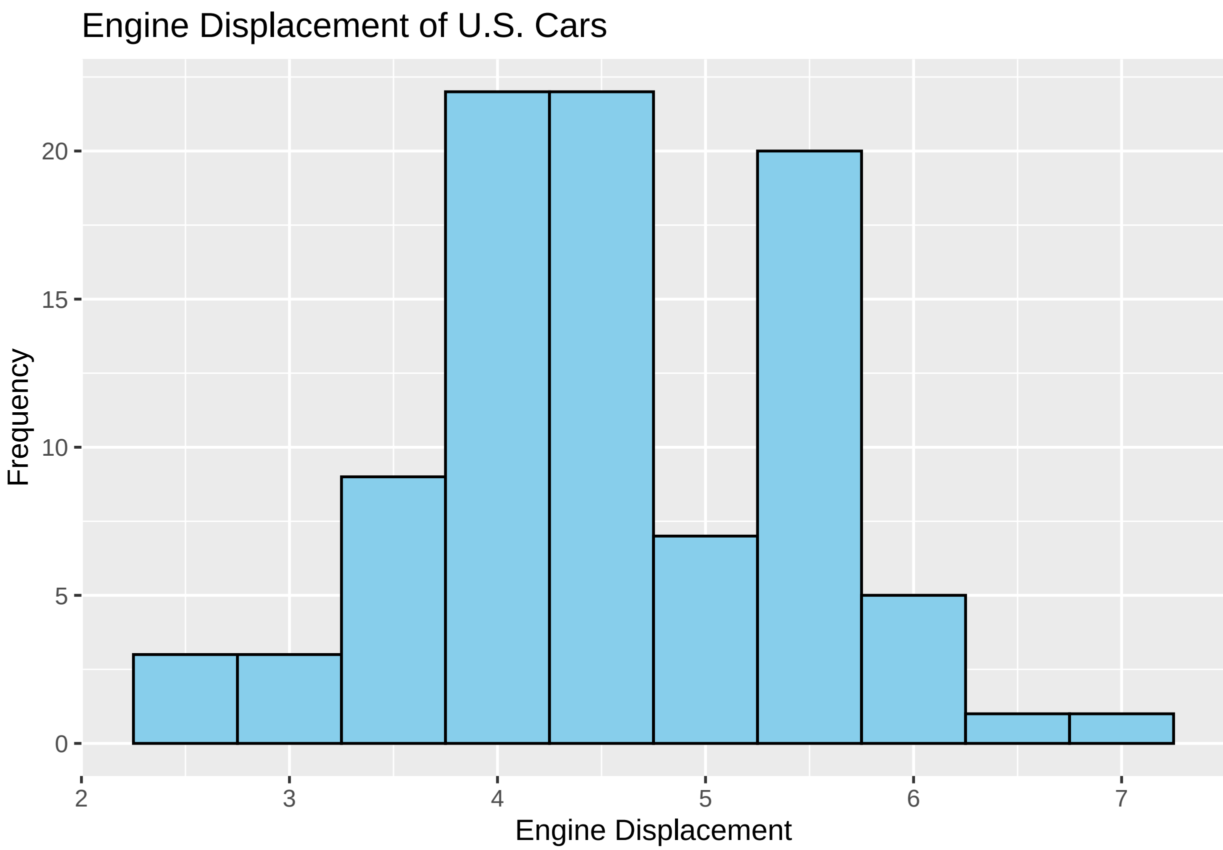
```r
cat("Summary statistics for Japan cars: \"")
```

```
## Summary statistics for Japan cars: "
```

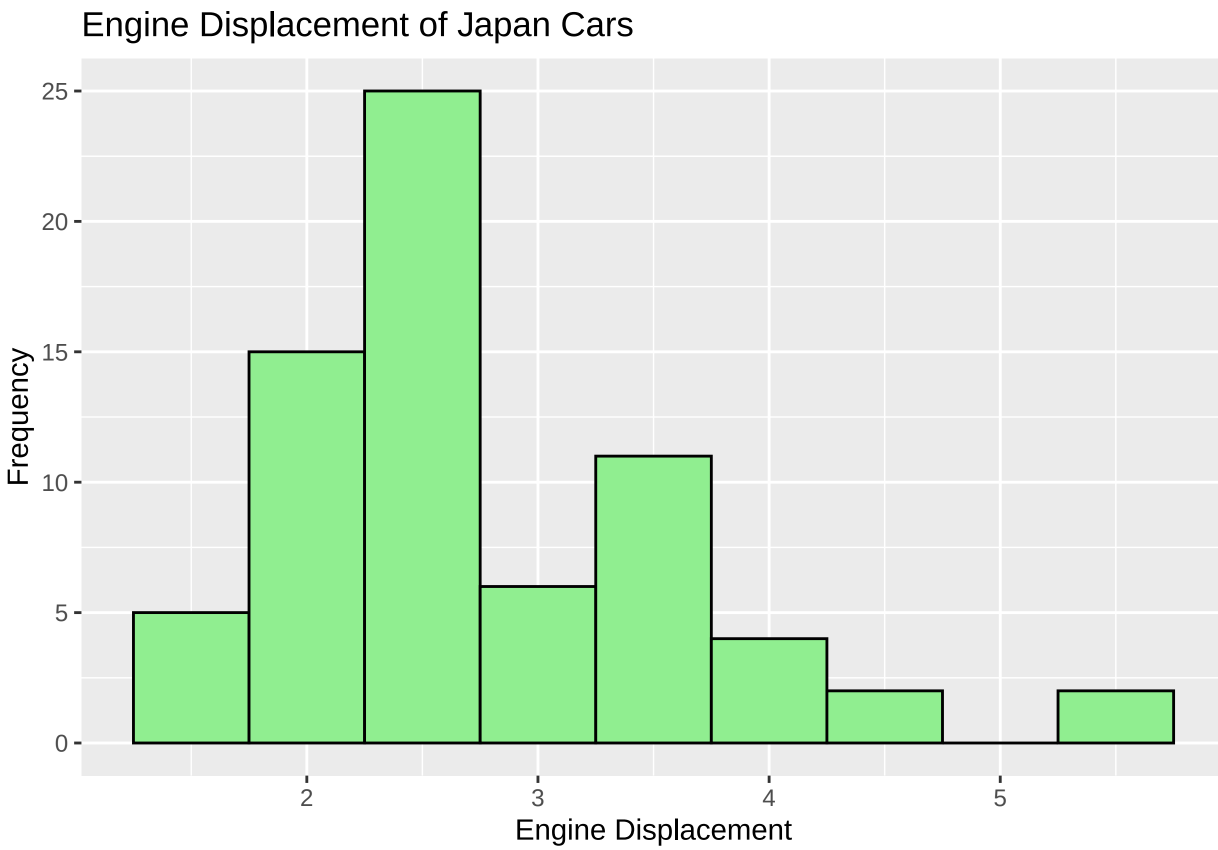
```
cat("Mean:", japan_iqr, "\n")
```

```
## Mean: 7.625
```

```
# Problem 2j
# Make a histogram of the engine displacement
# (i.e. displ) of U.S. cars and Japan cars, respectively,
# and describe their shape
us_cars <- subset(mpg, manufacturer %in% c("ford", "chevrolet", "dodge", "mercury", "pontiac", "lincoln
japan_cars <- subset(mpg, manufacturer %in% c("honda", "toyota", "nissan", "subaru", "mazda", "mitsubis
ggplot(us_cars, aes(x = displ)) +
  geom_histogram(binwidth = 0.5, fill = "skyblue", color = "black") +
  labs(title = "Engine Displacement of U.S. Cars",
       x = "Engine Displacement",
       y = "Frequency")
```




```
ggplot(japan_cars, aes(x = displ)) +
  geom_histogram(binwidth = 0.5, fill = "lightgreen", color = "black") +
  labs(title = "Engine Displacement of Japan Cars",
       x = "Engine Displacement",
       y = "Frequency")
```


Engine Displacement of Japan Cars
Frequency
25
20
15
10
5
0
2
3
4
5
Engine Displacement