homework3

2025-03-16

```
library(tidyverse)
```

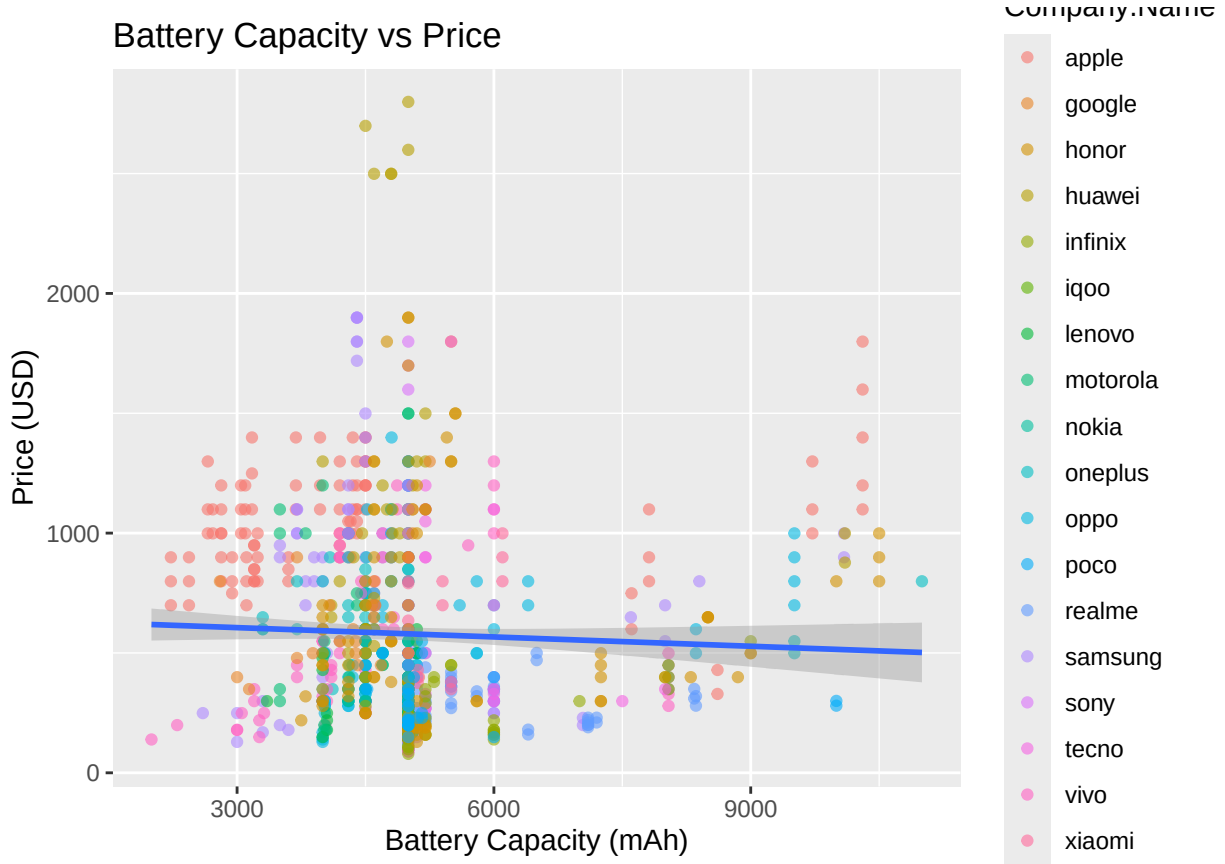
```
## Warning: package 'lubridate' was built under R version 4.4.1
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)

df <- read_csv("/Users/lusinegevorgyan/Desktop/mobiles_dataset.csv", show_col_types = FALSE) %>%
  mutate(

    across(starts_with("Launched.Price."),
           ~as.numeric(str_replace_all(as.character(.), "[^\\d.]", ""))),

    Price_Pakistan.PKR_USD = `Launched.Price.Pakistan.PKR` * 0.0036,
    Price_India.INR_USD = `Launched.Price.India.INR` * 0.011,
    Price_China.CNY_USD = `Launched.Price.China.CNY` * 0.14,
    Price_Dubai.AED_USD = `Launched.Price.Dubai.AED` * 0.27,
    Price_USD = `Launched.Price.USA.USD`
  ) %>%
  mutate(

    Company.Name = tolower(trimws(Company.Name)),
    RAM = as.numeric(str_replace_all(as.character(RAM), "[^\\d.]", "")),
    Battery.Capacity.mAh = as.numeric(str_replace_all(as.character(Battery.Capacity.mAh), "[^\\d.]", ""
  ) %>%
  drop_na(

    Battery.Capacity.mAh, RAM, Price_USD,
    Price_Pakistan.PKR_USD, Price_India.INR_USD,
    Price_China.CNY_USD, Price_Dubai.AED_USD
  )
```

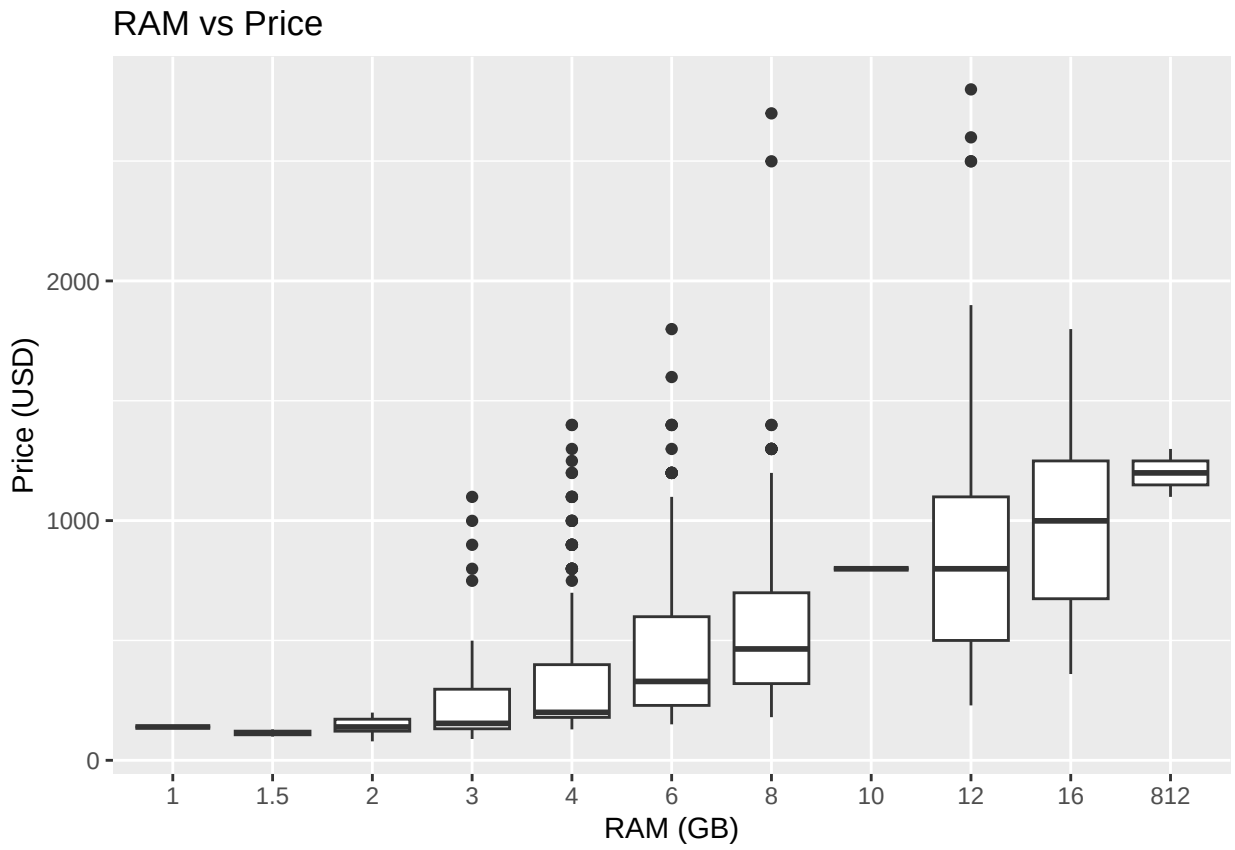
```r
battery_cor <- df %>%
  summarise(
    Global = cor(Battery.Capacity.mAh, Price_USD, use = "complete.obs"),
    Pakistan = cor(Battery.Capacity.mAh, Price_Pakistan.PKR_USD, use = "complete.obs"),
    India = cor(Battery.Capacity.mAh, Price_India.INR_USD, use = "complete.obs"),
    China = cor(Battery.Capacity.mAh, Price_China.CNY_USD, use = "complete.obs"),
    Dubai = cor(Battery.Capacity.mAh, Price_Dubai.AED_USD, use = "complete.obs"),
    USA = cor(Battery.Capacity.mAh, Price_USD, use = "complete.obs")
  )

ggplot(df, aes(x = Battery.Capacity.mAh, y = Price_USD)) +
  geom_point(aes(color = Company.Name), alpha = 0.6) +
  geom_smooth(method = "lm") +
  labs(title = "Battery Capacity vs Price", x = "Battery Capacity (mAh)", y = "Price (USD)")
```

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




```r
ram_cor <- df %>%
  summarise(
    Global = cor(RAM, Price_USD, use = "complete.obs"),
    Pakistan = cor(RAM, Price_Pakistan.PKR_USD, use = "complete.obs"),
    India = cor(RAM, Price_India.INR_USD, use = "complete.obs"),
    China = cor(RAM, Price_China.CNY_USD, use = "complete.obs"),
    Dubai = cor(RAM, Price_Dubai.AED_USD, use = "complete.obs"),
```

```
    USA = cor(RAM, Price_USD, use = "complete.obs")
  )

ggplot(df, aes(x = factor(RAM), y = Price_USD)) +
  geom_boxplot() +
  labs(title = "RAM vs Price", x = "RAM (GB)", y = "Price (USD)")
```




```
apple_variation <- df %>%
  filter(Company.Name == "apple") %>%
  summarise(across(starts_with("Price_"), \(x) sd(x, na.rm = TRUE))) %>%
  rowMeans()


cat("Apple's average price variation across regions:\n")
```

```
## Apple’s average price variation across regions:
```

```
print(apple_variation)
```

```
## [1] 266.8182
```

```
other_variation <- df %>%
  filter(Company.Name != "apple") %>%
```

```
  group_by(Company.Name) %>%
  summarise(across(starts_with("Price_"), \(x) sd(x, na.rm = TRUE))) %>%
  mutate(avg_var = rowMeans(select(., starts_with("Price_")))) %>%
  summarise(mean(avg_var))

cat("\nOther brands' average price variation:\n")
```

```
##
## Other brands' average price variation:
```

```
print(other_variation)
```

```
## # A tibble: 1 x 1
##   `mean(avg_var)`
##             <dbl>
## 1            263.
```

```
apple_markup <- df %>%
  filter(Company.Name == "apple") %>%
  summarise(across(starts_with("Price_"), \(x) mean(x, na.rm = TRUE))) %>%
  pivot_longer(everything()) %>%
  slice_max(value)

cat("\nCountry with highest Apple markup:\n")
```

```
##
## Country with highest Apple markup:
```

```
print(apple_markup)
```

```
## # A tibble: 1 x 2
##   name                value
##   <chr>               <dbl>
## 1 Price_India.INR_USD 1133.
```

```
stable_brands <- df %>%
  group_by(Company.Name) %>%
  summarise(across(starts_with("Price_"), \(x) sd(x, na.rm = TRUE))) %>%
  mutate(avg_var = rowMeans(select(., starts_with("Price_")))) %>%
  slice_min(avg_var, n = 3)

cat("\nTop 3 most stable brands:\n")
```

```
##
## Top 3 most stable brands:
```

```
print(stable_brands)
```

```
## # A tibble: 3 x 7
##   Company.Name Price_Pakistan.PKR_USD Price_India.INR_USD Price_China.CNY_USD
##   <chr>                         <dbl>               <dbl>               <dbl>
## 1 nokia                          53.8                49.7                43.4
## 2 iqoo                           36                  66                  85.5
## 3 infinix                        79.7                88.9                95.0
## # i 3 more variables: Price_Dubai.AED_USD <dbl>, Price_USD <dbl>, avg_var <dbl>
```

```
df <- df %>%
  mutate(Price_Category = case_when(
    Price_USD < 300 ~ "Budget",
    Price_USD >= 300 & Price_USD <= 700 ~ "Mid-Range",
    Price_USD > 700 ~ "Premium"
  ))

brand_segments <- df %>%
  count(Company.Name, Price_Category) %>%
  pivot_wider(
    names_from = Price_Category,
    values_from = n,
    values_fill = 0
  )
cat("\nBrand Segmentation Matrix:\n")
```

```
##
## Brand Segmentation Matrix:
```

```
print(brand_segments)
```

```
## # A tibble: 18 x 4
##    Company.Name `Mid-Range` Premium Budget
##    <chr>              <int>   <int>  <int>
##  1 apple                  8      89      0
##  2 google                12       9      0
##  3 honor                 37      25     29
##  4 huawei                15      27      0
##  5 infinix               15       0     41
##  6 iqoo                   3       0      0
##  7 lenovo                 5       0     10
##  8 motorola              33       7     22
##  9 nokia                  0       0     10
## 10 oneplus               23      20     10
## 11 oppo                  59      24     46
## 12 poco                  15       0     17
## 13 realme                26       0     43
## 14 samsung               19      39     26
## 15 sony                   3       6      0
## 16 tecno                 12       9     18
## 17 vivo                  37      16     33
## 18 xiaomi                12       9      6
```

```r
full_coverage <- brand_segments %>%
  filter(Budget > 0 & `Mid-Range` > 0 & Premium > 0)

cat("\nBrands Covering All Segments:\n")
```

```
##
## Brands Covering All Segments:
```

```r
print(full_coverage)
```

```
## # A tibble: 8 x 4
##   Company.Name `Mid-Range` Premium Budget
##   <chr>              <int>   <int>  <int>
## 1 honor                 37      25     29
## 2 motorola              33       7     22
## 3 oneplus               23      20     10
## 4 oppo                  59      24     46
## 5 samsung               19      39     26
## 6 tecno                 12       9     18
## 7 vivo                  37      16     33
## 8 xiaomi                12       9      6
```

```r
df <- df %>%
  mutate(Price_Category = case_when(
    Price_USD < 300 ~ "Budget",
    Price_USD >= 300 & Price_USD <= 700 ~ "Mid-Range",
    Price_USD > 700 ~ "Premium"
  ))

brand_segments <- df %>%
  count(Company.Name, Price_Category) %>%
  pivot_wider(names_from = Price_Category, values_from = n, values_fill = 0)

full_coverage <- brand_segments %>%
  filter(Budget > 0 & `Mid-Range` > 0 & Premium > 0)
```

```r
df <- df %>%
  mutate(Price_Category = case_when(
    Price_USD < 300 ~ "Budget",
    Price_USD >= 300 & Price_USD <= 700 ~ "Mid-Range",
    Price_USD > 700 ~ "Premium"
  ))

brand_segments <- df %>%
  count(Company.Name, Price_Category) %>%
  pivot_wider(
    names_from = Price_Category,
    values_from = n,
    values_fill = 0
  )

full_coverage <- brand_segments %>%
```

```
  filter(Budget > 0 & `Mid-Range` > 0 & Premium > 0)


cat("\nBrand Segmentation Matrix:\n")
```

```
##
## Brand Segmentation Matrix:
```

```
print(brand_segments)
```

```
## # A tibble: 18 x 4
##    Company.Name `Mid-Range` Premium Budget
##    <chr>              <int>   <int>  <int>
##  1 apple                  8      89      0
##  2 google                12       9      0
##  3 honor                 37      25     29
##  4 huawei                15      27      0
##  5 infinix               15       0     41
##  6 iqoo                   3       0      0
##  7 lenovo                 5       0     10
##  8 motorola              33       7     22
##  9 nokia                  0       0     10
## 10 oneplus               23      20     10
## 11 oppo                  59      24     46
## 12 poco                  15       0     17
## 13 realme                26       0     43
## 14 samsung               19      39     26
## 15 sony                   3       6      0
## 16 tecno                 12       9     18
## 17 vivo                  37      16     33
## 18 xiaomi                12       9      6
```

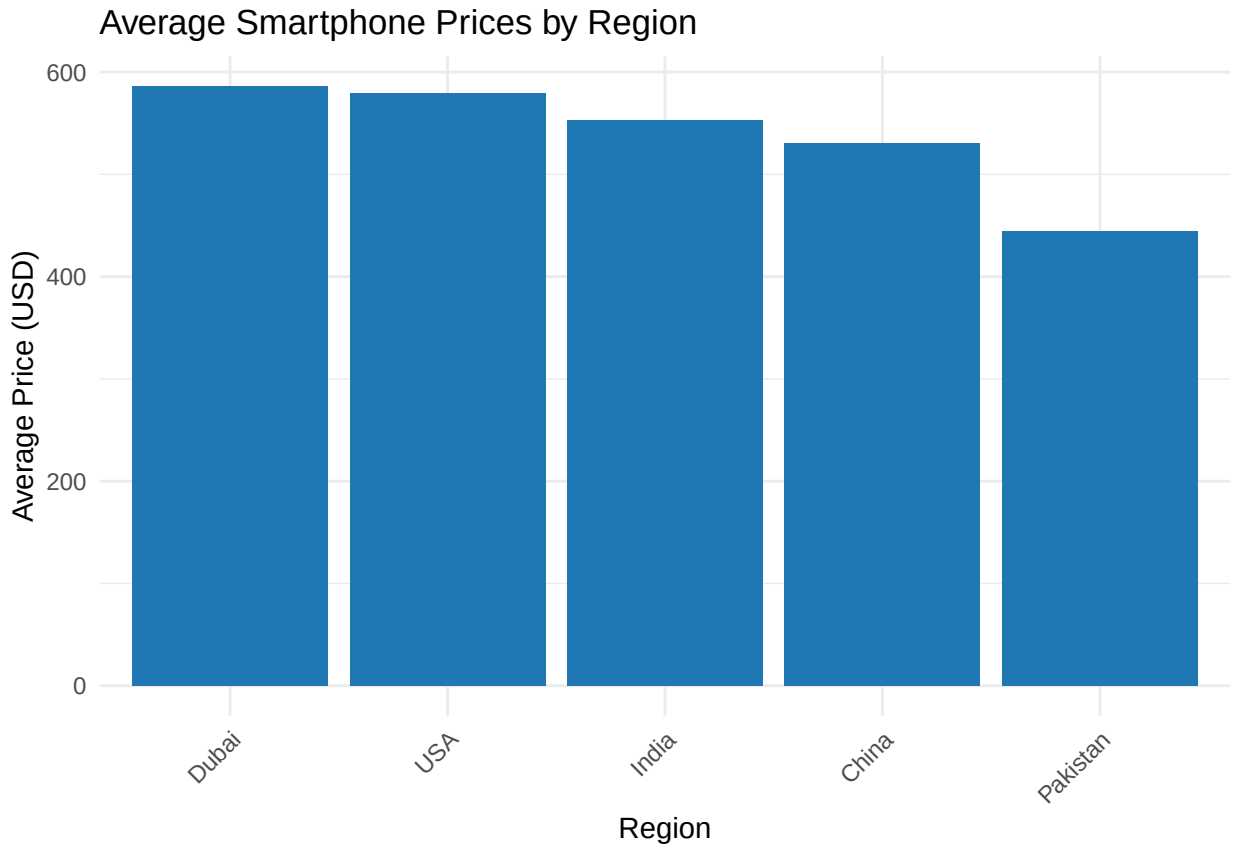
```
cat("\nBrands Covering All Segments:\n")
```

```
##
## Brands Covering All Segments:
```

```
print(full_coverage)
```

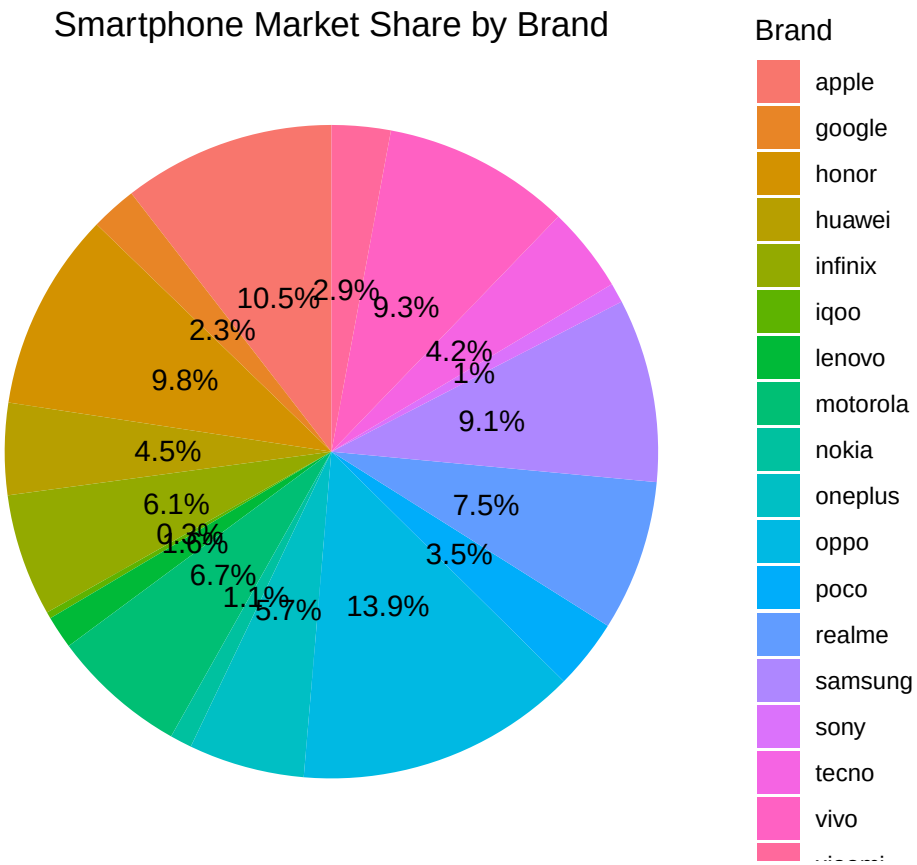
```
## # A tibble: 8 x 4
##   Company.Name `Mid-Range` Premium Budget
##   <chr>              <int>   <int>  <int>
## 1 honor                 37      25     29
## 2 motorola              33       7     22
## 3 oneplus               23      20     10
## 4 oppo                  59      24     46
## 5 samsung               19      39     26
## 6 tecno                 12       9     18
## 7 vivo                  37      16     33
## 8 xiaomi                12       9      6
```

```r
avg_prices <- df %>%
  summarise(
    Pakistan = mean(Price_Pakistan.PKR_USD, na.rm = TRUE),
    India = mean(Price_India.INR_USD, na.rm = TRUE),
    China = mean(Price_China.CNY_USD, na.rm = TRUE),
    Dubai = mean(Price_Dubai.AED_USD, na.rm = TRUE),
    USA = mean(Price_USD, na.rm = TRUE)
  ) %>%
  pivot_longer(everything(), names_to = "Region", values_to = "Average_Price")


ggplot(avg_prices, aes(x = reorder(Region, -Average_Price), y = Average_Price)) +
  geom_col(fill = "#1f77b4") +
  labs(title = "Average Smartphone Prices by Region",
       x = "Region",
       y = "Average Price (USD)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
market_share <- df %>%
  count(Company.Name, name = "Count") %>%
  mutate(Percentage = Count / sum(Count) * 100,
         Company.Name = fct_lump(Company.Name, prop = 0.02)) %>%
  group_by(Company.Name) %>%
  summarise(Percentage = sum(Percentage))
```

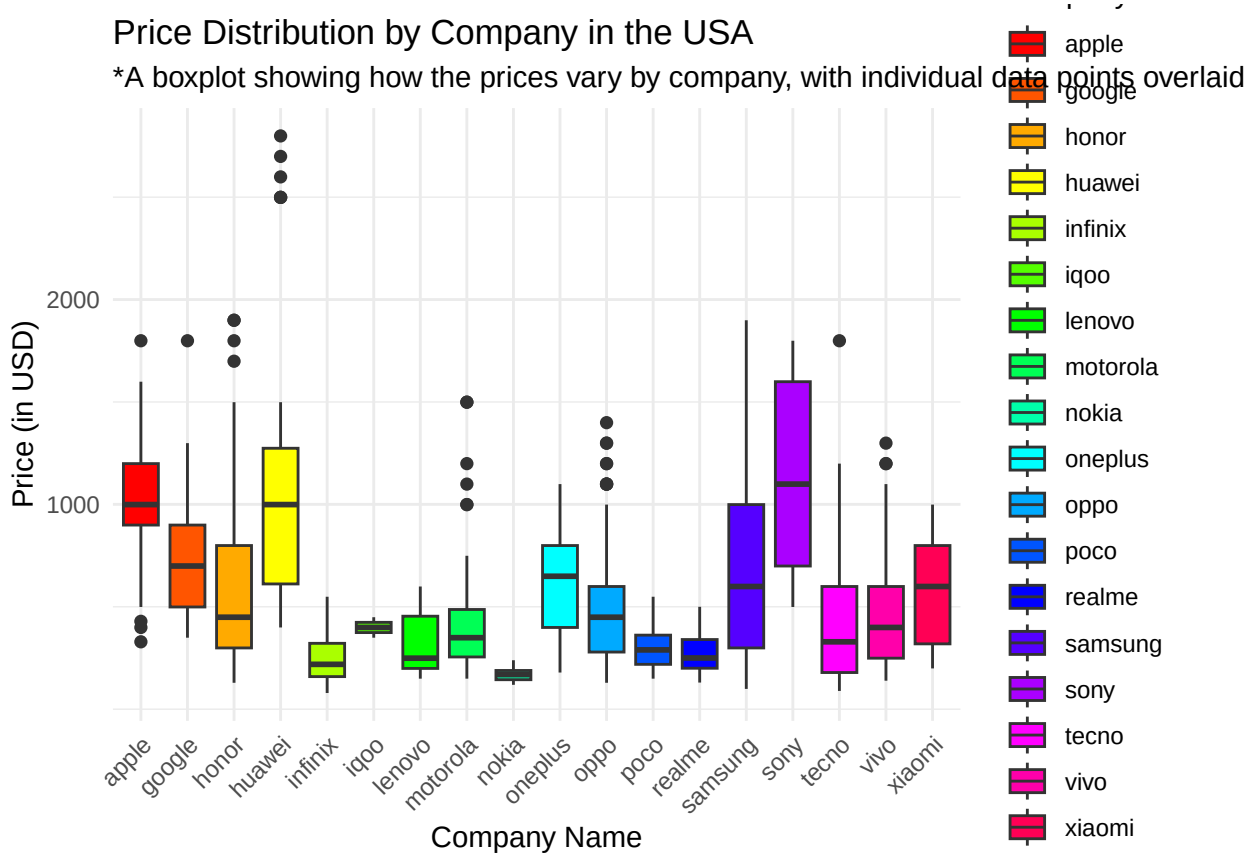
```
ggplot(market_share, aes(x = "", y = Percentage, fill = Company.Name)) +
  geom_col(width = 1) +
  coord_polar("y", start = 0) +
  geom_text(aes(label = paste0(round(Percentage, 1), "%")),
            position = position_stack(vjust = 0.5)) +
  labs(title = "Smartphone Market Share by Brand",
       fill = "Brand") +
  theme_void() +
  theme(legend.position = "right",
        plot.title = element_text(hjust = 0.5))
```




```
colnames(df)
```

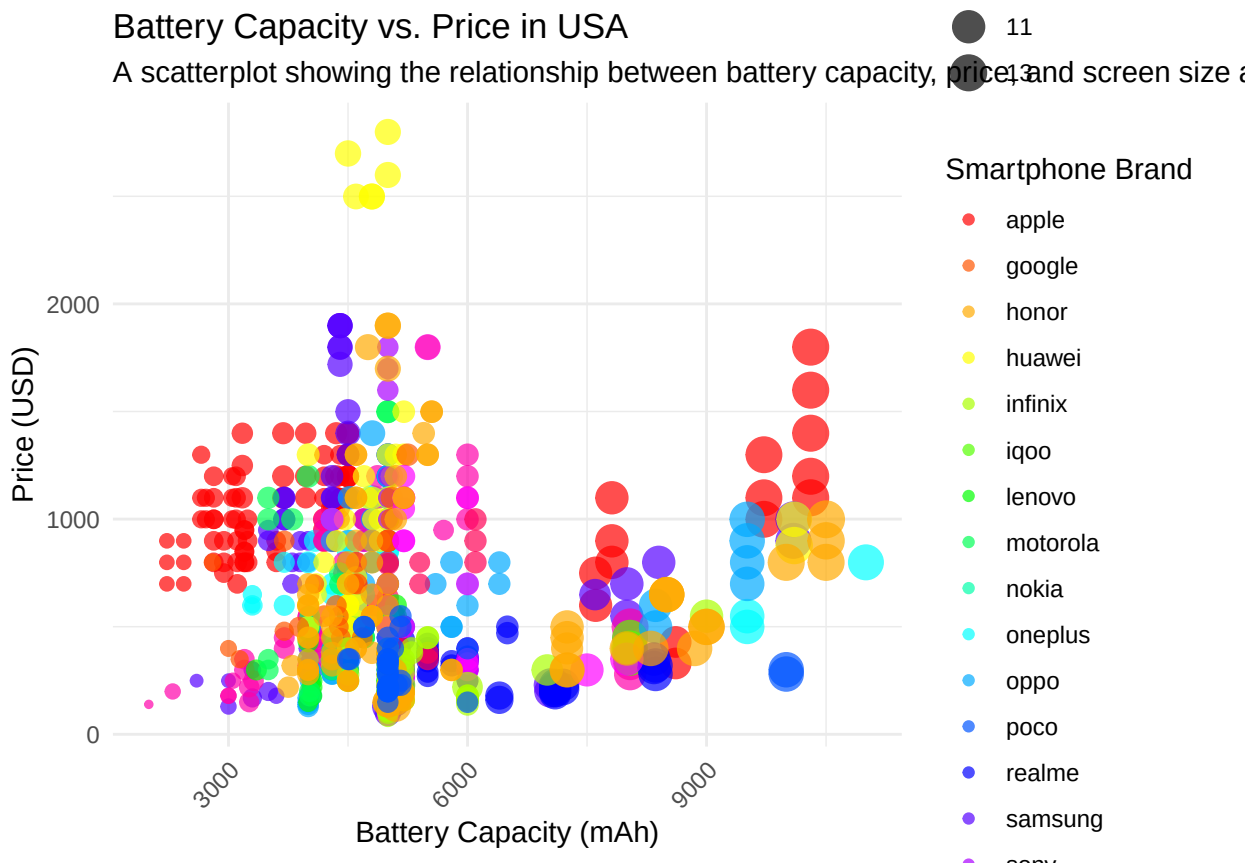
```
##  [1] "Company.Name"                "Model.Name"
##  [3] "Mobile.Weight"               "RAM"
##  [5] "Front.Camera"                "Back.Camera"
##  [7] "Processor"                   "Battery.Capacity.mAh"
##  [9] "Screen.Size.inches"          "Launched.Price.Pakistan.PKR"
## [11] "Launched.Price.India.INR"    "Launched.Price.China.CNY"
## [13] "Launched.Price.USA.USD"      "Launched.Price.Dubai.AED"
## [15] "Launched.Year"               "Price_Pakistan.PKR_USD"
## [17] "Price_India.INR_USD"         "Price_China.CNY_USD"
## [19] "Price_Dubai.AED_USD"         "Price_USD"
## [21] "Price_Category"
```

```
ggplot(df, aes(x = Company.Name, y = Price_USD, fill = Company.Name)) +
  geom_boxplot(outlier.shape = 16, outlier.size = 2) +
  labs(
    title = "Price Distribution by Company in the USA",
    subtitle = "*A boxplot showing how the prices vary by company, with individual data points overlaid
    x = "Company Name",
    y = "Price (in USD)"
  ) +
  scale_y_continuous(breaks = c(1000, 2000)) +
  scale_fill_manual(values = rainbow(length(unique(df$Company.Name)))) +
  scale_color_manual(values = rainbow(length(unique(df$Company.Name)))) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "right"
  )
```




```
ggplot(df, aes(x = Battery.Capacity.mAh, y = Price_USD, color = Company.Name, size = Screen.Size.inches)
  geom_point(alpha = 0.7) +
  labs(
    title = "Battery Capacity vs. Price in USA",
    subtitle = "A scatterplot showing the relationship between battery capacity, price, and screen size
    x = "Battery Capacity (mAh)",
    y = "Price (USD)",
    color = "Smartphone Brand",
```

```
    size = "Screen Size (inches)"
  ) +
  theme_minimal() +
  theme(
    legend.position = "right",
    axis.text.x = element_text(angle = 45, hjust = 1)
  ) +
  scale_color_manual(values = rainbow(length(unique(df$Company.Name))))
```




```
custom_shapes <- c("apple" = 16,
                   "honor" = 17,
                   "oppo" = 18,
                   "samsung" = 15,
                   "vivo" = 16)

ggplot(df, aes(x = Battery.Capacity.mAh, y = Price_USD)) +
  geom_point(aes(shape = Company.Name), alpha = 0.7, size = 3, color = ifelse(df$Battery.Capacity.mAh <
  scale_x_continuous(breaks = seq(2000, 10000, by = 2000), limits = c(200, 10000)) +
  scale_y_continuous(breaks = seq(500, 1500, by = 500), limits = c(500, 1500)) +
  scale_shape_manual(values = custom_shapes) +
  labs(
    title = "Battery Capacity vs. Price in USA",
    subtitle = "*Scatterplot with custom axis ranges and two-tone point colors*",
    x = "Battery Capacity (mAh)",
    y = "Price (USD)",
```

```
    shape = "Smartphone Brand"
  ) +
  theme_minimal() +
  theme(
    legend.position = "right",
    plot.title = element_text(size = 16, face = "bold"),
    plot.subtitle = element_text(size = 12, face = "italic"),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

```
## Warning: Removed 694 rows containing missing values or values outside the scale range
## (`geom_point()`).
```