

HW3_Norayr_Sukiasyan

Norayr Sukiasyan

2025-03-16

```
library(tidyverse)
```

```
## -- 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  1.9.3      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(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##   discard
##
## The following object is masked from 'package:readr':
##
##   col_factor
```

```
mobiles <- read.csv("mobiles_dataset.csv", stringsAsFactors = FALSE)
```

```
mobiles <- mobiles %>%
  mutate(Price_Pakistan_USD = Launched.Price.Pakistan.PKR * 0.0036,
         Price_India_USD    = Launched.Price.India.INR * 0.011,
         Price_China_USD    = Launched.Price.China.CNY * 0.14,
         Price_USA_USD      = Launched.Price.USA.USD,      # Already in USD
         Price_Dubai_USD    = Launched.Price.Dubai.AED * 0.27)
```

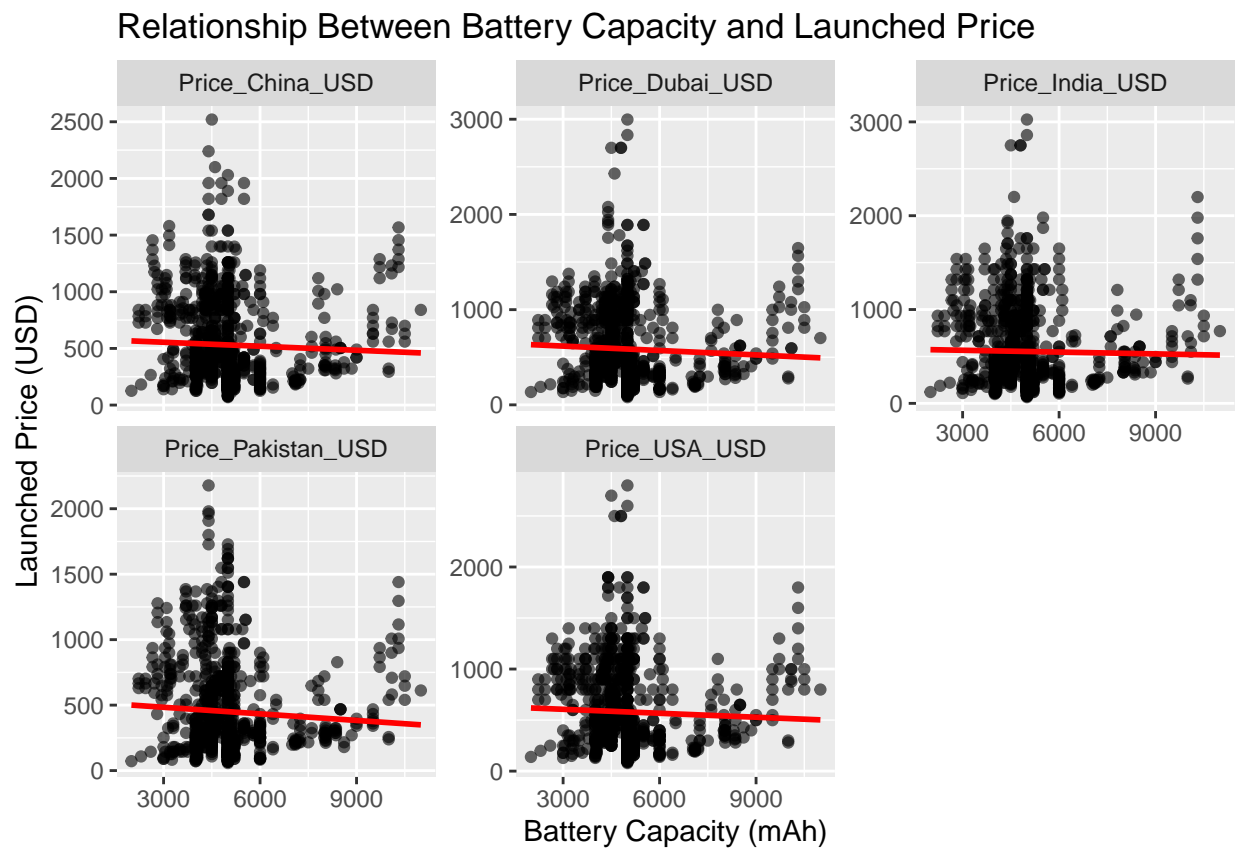
```
price_long <- mobiles %>%
  select(Model.Name, Battery.Capacity.mAh, RAM,
         Price_Pakistan_USD, Price_India_USD, Price_China_USD, Price_USA_USD, Price_Dubai_USD) %>%
  pivot_longer(cols = starts_with("Price_"),
               names_to = "Region",
               values_to = "Price_USD")
```

Part 1

1.1

```
ggplot(price_long, aes(x = Battery.Capacity.mAh, y = Price_USD)) +  
  geom_point(alpha = 0.6) +  
  facet_wrap(~ Region, scales = "free_y") +  
  geom_smooth(method = "lm", se = FALSE, col = "red") +  
  labs(title = "Relationship Between Battery Capacity and Launched Price",  
       x = "Battery Capacity (mAh)",  
       y = "Launched Price (USD)")
```

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



```
battery_corr <- price_long %>%  
  group_by(Region) %>%  
  summarise(correlation = cor(Battery.Capacity.mAh, Price_USD, use = "complete.obs"))  
print(battery_corr)
```

```
## # A tibble: 5 x 2  
##   Region      correlation  
##   <chr>          <dbl>
```

```
## 1 Price_China_USD      -0.0410
## 2 Price_Dubai_USD      -0.0489
## 3 Price_India_USD      -0.0191
## 4 Price_Pakistan_USD   -0.0609
## 5 Price_USA_USD        -0.0411
```

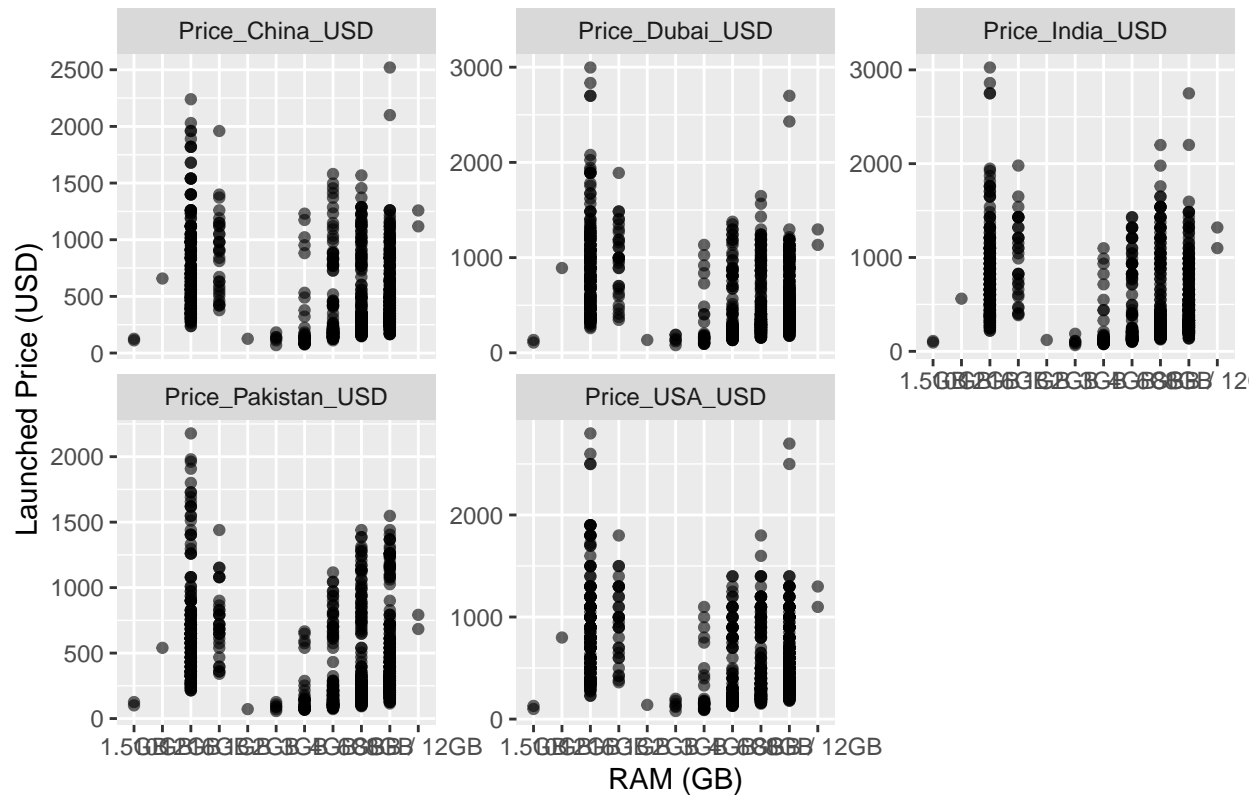
1.2

```
ram_long <- mobiles %>%
  select(Model.Name, RAM,
         Price_Pakistan_USD, Price_India_USD, Price_China_USD, Price_USA_USD, Price_Dubai_USD) %>%
  pivot_longer(cols = starts_with("Price_"),
               names_to = "Region",
               values_to = "Price_USD")

ggplot(ram_long, aes(x = RAM, y = Price_USD)) +
  geom_point(alpha = 0.6) +
  facet_wrap(~ Region, scales = "free_y") +
  geom_smooth(method = "lm", se = FALSE, col = "blue") +
  labs(title = "Relationship Between RAM Size and Launched Price",
       x = "RAM (GB)",
       y = "Launched Price (USD)")
```

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

Relationship Between RAM Size and Launched Price



1.3

```
apple <- mobiles %>% filter(Company.Name == "Apple")

apple_long <- apple %>%
  select(Model.Name, Price_Pakistan_USD, Price_India_USD, Price_China_USD, Price_USA_USD, Price_Dubai_USD) %>%
  pivot_longer(cols = starts_with("Price_"),
               names_to = "Region",
               values_to = "Price_USD")

apple_range <- apple %>%
  mutate(Price_Range = pmax(Price_Pakistan_USD, Price_India_USD, Price_China_USD, Price_USA_USD, Price_Dubai_USD, na.rm = TRUE),
         Price_Range = pmin(Price_Pakistan_USD, Price_India_USD, Price_China_USD, Price_USA_USD, Price_Dubai_USD, na.rm = TRUE)) %>%
  select(Model.Name, Price_Range)
print(apple_range)
```

```
##           Model.Name Price_Range
## 1      iPhone 16 128GB  124.2590
## 2      iPhone 16 256GB  125.2590
## 3      iPhone 16 512GB  126.2590
## 4  iPhone 16 Plus 128GB  126.2590
## 5  iPhone 16 Plus 256GB  135.1290
## 6  iPhone 16 Plus 512GB  183.2590
## 7      iPhone 16 Pro 128GB  155.2590
## 8      iPhone 16 Pro 256GB  161.1290
## 9      iPhone 16 Pro 512GB  215.1290
```

## 10	iPhone 16 Pro Max	128GB	184.2590
## 11	iPhone 16 Pro Max	256GB	185.2590
## 12	iPhone 16 Pro Max	512GB	241.2590
## 13	iPhone 15	128GB	96.2590
## 14	iPhone 15	256GB	105.9926
## 15	iPhone 15	512GB	179.9926
## 16	iPhone 15 Plus	128GB	137.1290
## 17	iPhone 15 Plus	256GB	177.1290
## 18	iPhone 15 Plus	512GB	203.1290
## 19	iPhone 15 Pro	128GB	128.2590
## 20	iPhone 15 Pro	256GB	188.1290
## 21	iPhone 15 Pro	512GB	228.1290
## 22	iPhone 15 Pro Max	128GB	212.2590
## 23	iPhone 15 Pro Max	256GB	241.2590
## 24	iPhone 15 Pro Max	512GB	297.2590
## 25	iPhone 14	128GB	133.0036
## 26	iPhone 14	256GB	147.0036
## 27	iPhone 14	512GB	211.0036
## 28	iPhone 14 Plus	128GB	161.0036
## 29	iPhone 14 Plus	256GB	225.0036
## 30	iPhone 14 Plus	512GB	239.0036
## 31	iPhone 14 Pro	128GB	211.2590
## 32	iPhone 14 Pro	256GB	270.1290
## 33	iPhone 14 Pro	512GB	365.1290
## 34	iPhone 14 Pro Max	128GB	295.2590
## 35	iPhone 14 Pro Max	256GB	351.2590
## 36	iPhone 14 Pro Max	512GB	380.2590
## 37	iPhone 13 mini	128GB	138.9036
## 38	iPhone 13 mini	256GB	157.9036
## 39	iPhone 13 mini	512GB	231.9036
## 40	iPhone 13	128GB	121.9036
## 41	iPhone 13	256GB	195.9036
## 42	iPhone 13	512GB	233.9036
## 43	iPhone 13 Pro	128GB	319.9000
## 44	iPhone 13 Pro	256GB	351.0400
## 45	iPhone 13 Pro	512GB	391.0400
## 46	iPhone 13 Pro Max	128GB	329.9000
## 47	iPhone 13 Pro Max	256GB	351.1700
## 48	iPhone 13 Pro Max	512GB	407.1700
## 49	iPhone 12 mini	64GB	138.9036
## 50	iPhone 12 mini	128GB	157.9036
## 51	iPhone 12 mini	256GB	231.9036
## 52	iPhone 12	64GB	121.9036
## 53	iPhone 12	128GB	195.9036
## 54	iPhone 12	256GB	233.9036
## 55	iPhone 12 Pro	128GB	319.9000
## 56	iPhone 12 Pro	256GB	351.0400
## 57	iPhone 12 Pro	512GB	391.0400
## 58	iPhone 12 Pro Max	128GB	329.9000
## 59	iPhone 12 Pro Max	256GB	351.1700
## 60	iPhone 12 Pro Max	512GB	407.1700
## 61	iPhone 11	64GB	193.8636
## 62	iPhone 11	128GB	213.8636
## 63	iPhone 11	256GB	285.9036

```
## 64      iPhone 11 Pro 64GB      306.9036
## 65      iPhone 11 Pro 256GB     344.9036
## 66      iPhone 11 Pro 512GB     382.9036
## 67      iPhone 11 Pro Max 64GB  236.9036
## 68      iPhone 11 Pro Max 256GB 274.9036
## 69      iPhone 11 Pro Max 512GB 312.9036
## 70      iPhone X 64GB          580.3236
## 71      iPhone X 256GB         564.3236
## 72      iPhone XS 64GB         494.3236
## 73      iPhone XS 256GB        506.3236
## 74      iPhone XS 512GB        518.3236
## 75      iPhone XS Max 64GB      512.3236
## 76      iPhone XS Max 256GB     524.3236
## 77      iPhone XS Max 512GB     536.3236
## 78      iPhone XR 64GB         341.8636
## 79      iPhone XR 128GB        375.8636
## 80      iPhone XR 256GB        373.8636
## 81      iPad Air 10.9-inch 64GB 315.9036
## 82      iPad Air 10.9-inch 256GB 425.0036
## 83      iPad 10.2-inch 32GB     149.0036
## 84      iPad 10.2-inch 128GB    222.9036
## 85      iPad Mini 7.9-inch 64GB 237.8636
## 86      iPad Mini 7.9-inch 256GB 260.9036
## 87      iPad Pro 11-inch 128GB   355.8636
## 88      iPad Pro 11-inch 256GB   395.8636
## 89      iPad Pro 11-inch 512GB   524.9036
## 90      iPad Pro 12.9-inch 128GB 327.8636
## 91      iPad Pro 12.9-inch 256GB 353.8636
## 92      iPad Pro 12.9-inch 512GB 382.9036
## 93      iPad Pro 13-inch 128GB   382.9036
## 94      iPad Pro 13-inch 256GB   530.9036
## 95      iPad Pro 13-inch 512GB   642.9036
## 96      iPad Pro 13-inch 1TB     682.9036
## 97      iPad Pro 13-inch 2TB     758.9036
```

```
all_brands_range <- mobiles %>%
  mutate(Price_Range = pmax(Price_Pakistan_USD, Price_India_USD, Price_China_USD, Price_USA_USD, Price_Indonesia_USD, Price_Dubai_USD),
         Price_Range = pmin(Price_Pakistan_USD, Price_India_USD, Price_China_USD, Price_USA_USD, Price_Indonesia_USD, Price_Dubai_USD),
         group_by(Company.Name) %>%
         summarise(Avg_Price_Range = mean(Price_Range, na.rm = TRUE),
                   Model_Count = n()))
print(all_brands_range)
```

```
## # A tibble: 19 x 3
##   Company.Name Avg_Price_Range Model_Count
##   <chr>         <dbl>         <int>
## 1 Apple         293.             97
## 2 Google        261.             21
## 3 Honor         189.             91
## 4 Huawei        519.             42
## 5 Infinix        89.9            56
## 6 Lenovo        110.             15
## 7 Motorola       138.             62
## 8 Nokia         41.0             10
```

```
## 9 OnePlus                204.         53
## 10 Oppo                  220.        129
## 11 POCO                   114.         30
## 12 Poco                   60.5          2
## 13 Realme                 48.9         69
## 14 Samsung                259.         84
## 15 Sony                   371.          9
## 16 Tecno                  192.         39
## 17 Vivo                   241.         86
## 18 Xiaomi                 195.         27
## 19 iQOO                   198.          3
```

```
apple_region_avg <- apple_long %>%
  group_by(Region) %>%
  summarise(Avg_Price = mean(Price_USD, na.rm = TRUE)) %>%
  arrange(desc(Avg_Price))
print(apple_region_avg)
```

```
## # A tibble: 5 x 2
##   Region      Avg_Price
##   <chr>      <dbl>
## 1 Price_India_USD    1133.
## 2 Price_USA_USD     1028.
## 3 Price_China_USD   1005.
## 4 Price_Dubai_USD    995.
## 5 Price_Pakistan_USD 891.
```

1.4

```
mobiles <- mobiles %>%
  mutate(avg_price = rowMeans(select(., Price_Pakistan_USD, Price_India_USD, Price_China_USD, Price_USA_USD)),
         Segment = case_when(
           avg_price < 300 ~ "Budget",
           avg_price >= 300 & avg_price <= 700 ~ "Mid-range",
           avg_price > 700 ~ "Premium"
         ))
```

```
brand_segments <- mobiles %>%
  group_by(Company.Name, Segment) %>%
  summarise(Model_Count = n()) %>%
  arrange(Company.Name, Segment)
```

'summarise()' has grouped output by 'Company.Name'. You can override using the
'.groups' argument.

```
print(brand_segments)
```

```
## # A tibble: 44 x 3
## # Groups:   Company.Name [19]
```

```
##   Company.Name Segment   Model_Count
##   <chr>         <chr>         <int>
## 1 Apple        Budget         1
## 2 Apple        Mid-range        6
## 3 Apple        Premium        90
## 4 Google       Mid-range        11
## 5 Google       Premium        10
## 6 Honor        Budget         33
## 7 Honor        Mid-range        35
## 8 Honor        Premium        23
## 9 Huawei       Mid-range        17
## 10 Huawei      Premium        25
## # i 34 more rows
```

```
brand_coverage <- brand_segments %>%
  group_by(Company.Name) %>%
  summarise(Segments_Covered = paste(Segment, collapse = ", "),
            Num_Segments = n())
print(brand_coverage)
```

```
## # A tibble: 19 x 3
##   Company.Name Segments_Covered   Num_Segments
##   <chr>         <chr>         <int>
## 1 Apple        Budget, Mid-range, Premium      3
## 2 Google       Mid-range, Premium            2
## 3 Honor        Budget, Mid-range, Premium      3
## 4 Huawei       Mid-range, Premium            2
## 5 Infinix      Budget, Mid-range            2
## 6 Lenovo       Budget, Mid-range            2
## 7 Motorola     Budget, Mid-range, Premium      3
## 8 Nokia       Budget              1
## 9 OnePlus      Budget, Mid-range, Premium      3
## 10 Oppo        Budget, Mid-range, Premium      3
## 11 POCO        Budget, Mid-range            2
## 12 Poco        Budget              1
## 13 Realme      Budget, Mid-range            2
## 14 Samsung     Budget, Mid-range, Premium      3
## 15 Sony        Mid-range, Premium            2
## 16 Tecno       Budget, Mid-range, Premium      3
## 17 Vivo        Budget, Mid-range, Premium      3
## 18 Xiaomi      Budget, Mid-range, Premium      3
## 19 iQOO        Mid-range              1
```

1.5

```
region_avg <- mobiles %>%
  summarise(
    Pakistan = mean(Price_Pakistan_USD, na.rm = TRUE),
    India    = mean(Price_India_USD, na.rm = TRUE),
    China    = mean(Price_China_USD, na.rm = TRUE),
    USA      = mean(Price_USA_USD, na.rm = TRUE),
```



```

    Dubai    = mean(Price_Dubai_USD, na.rm = TRUE)
  )
print(region_avg)

```

```

##   Pakistan   India   China   USA   Dubai
## 1 449.9342 552.8237 530.7414 579.6238 586.029

```

```

brand_regional <- mobiles %>%
  select(Company.Name, Price_Pakistan_USD, Price_India_USD, Price_China_USD, Price_USA_USD, Price_Dubai_USD)
  pivot_longer(cols = starts_with("Price_"), names_to = "Region", values_to = "Price_USD") %>%
  group_by(Company.Name, Region) %>%
  summarise(Avg_Price = mean(Price_USD, na.rm = TRUE)) %>%
  ungroup()

```

'summarise()' has grouped output by 'Company.Name'. You can override using the
'.groups' argument.

```

brand_variation <- brand_regional %>%
  group_by(Company.Name) %>%
  summarise(Price_Range = max(Avg_Price) - min(Avg_Price))
print(brand_variation)

```

```

## # A tibble: 19 x 2
##   Company.Name Price_Range
##   <chr>         <dbl>
## 1 Apple         242.
## 2 Google        228.
## 3 Honor         175.
## 4 Huawei        470.
## 5 Infinix        87.8
## 6 Lenovo         89.7
## 7 Motorola      104.
## 8 Nokia         39.6
## 9 OnePlus       178.
## 10 Oppo         194.
## 11 POCO         113.
## 12 Poco         60.5
## 13 Realme       38.7
## 14 Samsung      82.1
## 15 Sony        345.
## 16 Tecno        190.
## 17 Vivo        209.
## 18 Xiaomi       146.
## 19 iQOO         196.

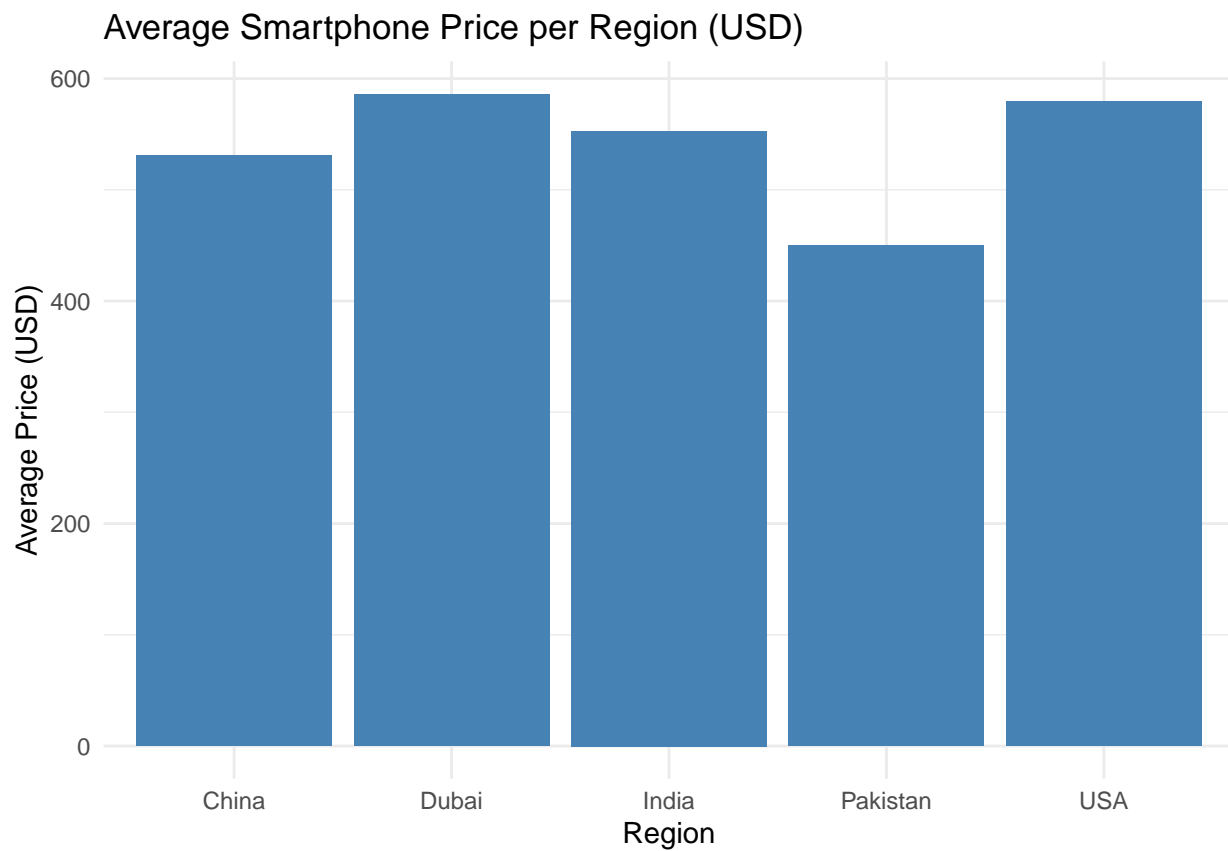
```

Part 2

2.1

```
## 1. Bar Chart for Average Price per Region (USD)
region_avg <- mobiles %>%
  summarise(
    Pakistan = mean(Price_Pakistan_USD, na.rm = TRUE),
    India    = mean(Price_India_USD, na.rm = TRUE),
    China    = mean(Price_China_USD, na.rm = TRUE),
    USA      = mean(Price_USA_USD, na.rm = TRUE),
    Dubai    = mean(Price_Dubai_USD, na.rm = TRUE)
  ) %>%
  pivot_longer(cols = everything(), names_to = "Region", values_to = "Avg_Price")

# Plot the bar chart
ggplot(region_avg, aes(x = Region, y = Avg_Price)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  labs(title = "Average Smartphone Price per Region (USD)",
       x = "Region",
       y = "Average Price (USD)") +
  theme_minimal()
```

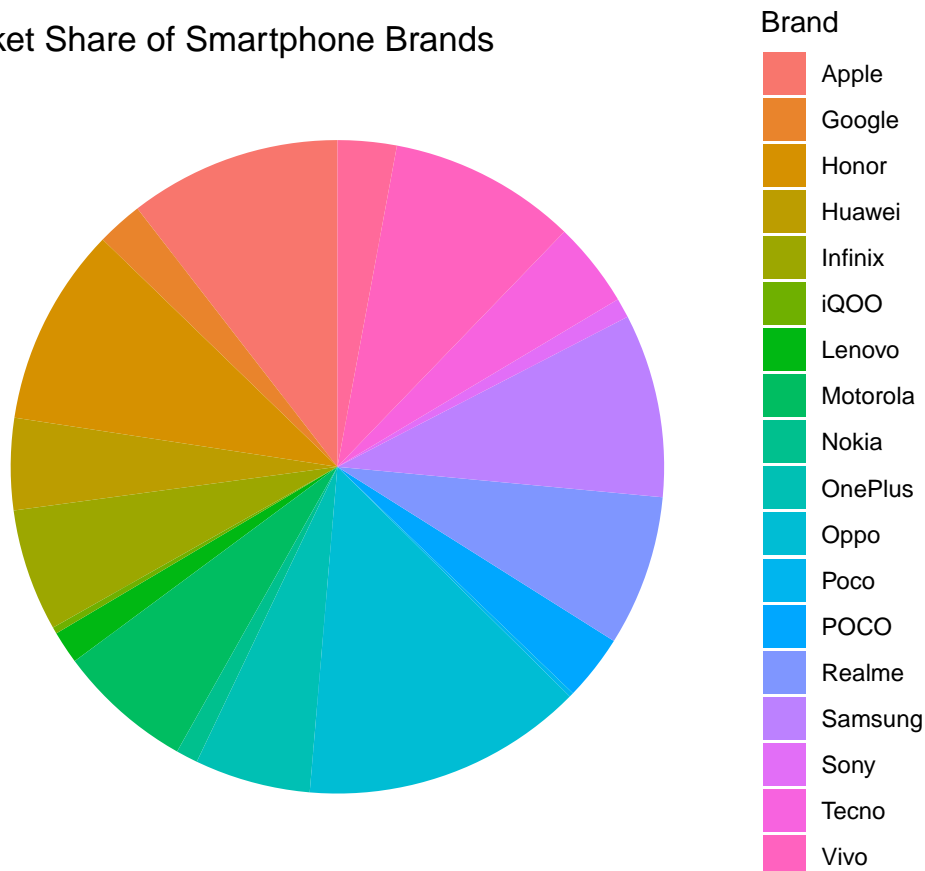


2.2

```
## 2. Pie Chart of the Market Share of Smartphone Brands
brand_share <- mobiles %>%
  group_by(Company.Name) %>%
  summarise(Count = n())

ggplot(brand_share, aes(x = "", y = Count, fill = Company.Name)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Market Share of Smartphone Brands",
       fill = "Brand") +
  theme_void() +
  theme(legend.position = "right")
```

Market Share of Smartphone Brands

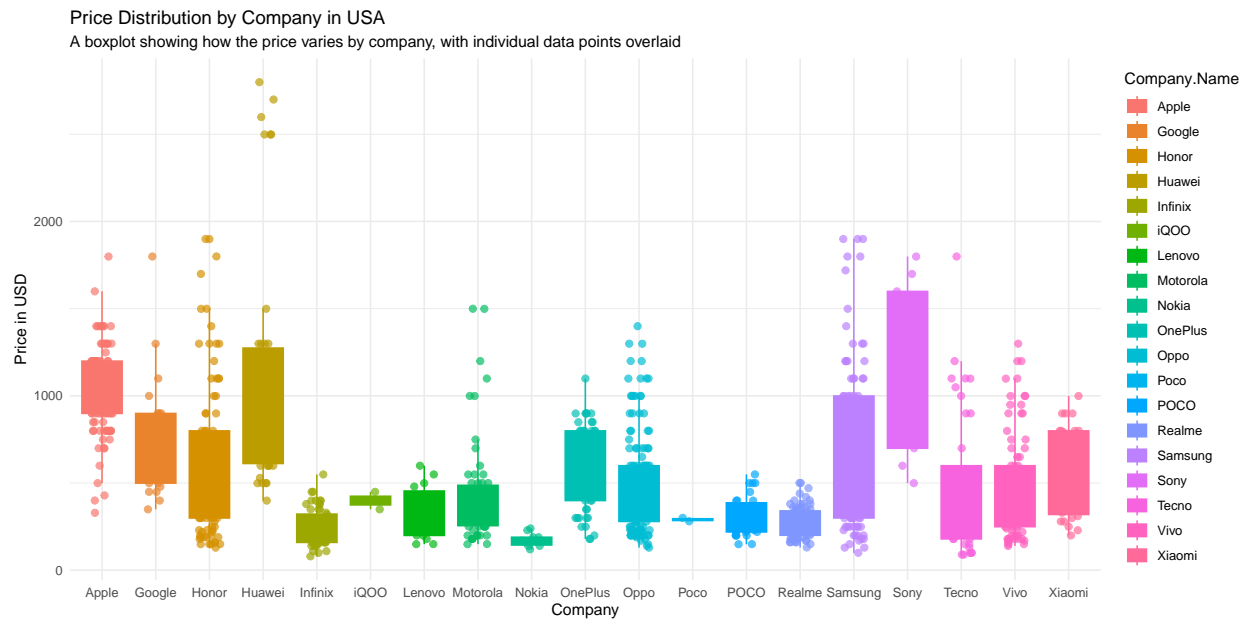


Part 3

3.1

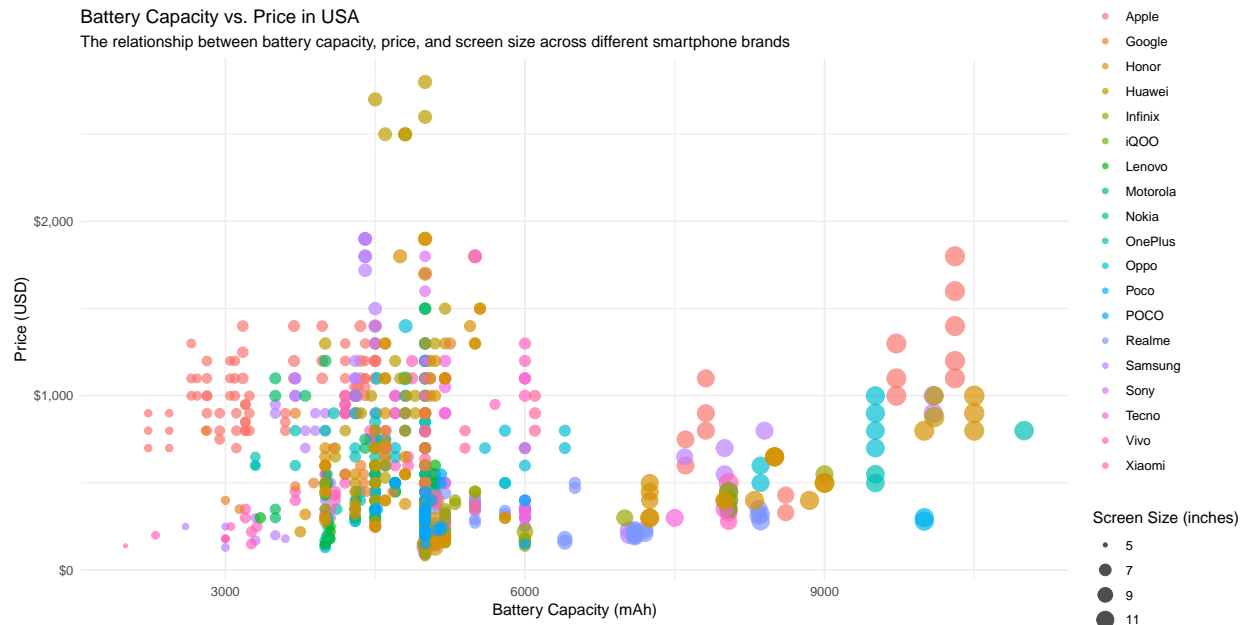
```
ggplot(mobiles, aes(x = Company.Name,
                    y = Price_USA_USD,
                    color = Company.Name,
                    fill = Company.Name)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, size = 2) +
```

```
labs(title = "Price Distribution by Company in USA",
      subtitle = "A boxplot showing how the price varies by company, with individual data points overlaid",
      x = "Company",
      y = "Price in USD") +
theme_minimal() +
theme(legend.position = "right")
```



3.2

```
ggplot(mobiles, aes(x = Battery.Capacity.mAh,
                    y = Price_USA_USD,
                    color = Company.Name,
                    size = Screen.Size.inches)) +
geom_point(alpha = 0.7) +
labs(title = "Battery Capacity vs. Price in USA",
      subtitle = "The relationship between battery capacity, price, and screen size across different s",
      x = "Battery Capacity (mAh)",
      y = "Price (USD)",
      color = "Brand",
      size = "Screen Size (inches)") +
scale_y_continuous(labels = dollar_format(prefix = "$")) +
theme_minimal() +
theme(legend.position = "right")
```



3.3

```
top_5_brands <- c("Apple", "Honor", "Oppo", "Samsung", "Vivo")

mobiles_top5 <- mobiles %>%
  filter(Company.Name %in% top_5_brands)

mobiles_top5 <- mobiles_top5 %>%
  mutate(
    Battery.Capacity.mAh = as.numeric(Battery.Capacity.mAh),
    Price_USA_USD       = as.numeric(Price_USA_USD),
    Screen.Size.inches  = as.numeric(Screen.Size.inches)
  )

ggplot(mobiles_top5, aes(x = Battery.Capacity.mAh,
                        y = Price_USA_USD,
                        shape = Company.Name,
                        color = Screen.Size.inches)) +
  geom_point(size = 3, alpha = 0.7) +
  labs(title = "Battery Capacity vs. Price for Top 5 Brands",
       subtitle = "Different shapes for each brand, color by screen size (USA)",
       x = "Battery Capacity (mAh)",
       y = "Price (USD)",
       shape = "Brand",
       color = "Screen Size (inches)") +
  # Format y-axis as dollar amounts if desired
  scale_y_continuous(labels = dollar_format(prefix = "$")) +
  # Minimal theme
  theme_minimal() +
  theme(legend.position = "right")
```

