Sales Analysis Report

Mahmoud Hafez

2023-08-12

This Sales analysis Markdown document involves the analysis of sales and product data to gain insights into various aspects of a business's performance. The dataset used for this project contains information about sales transactions and products, and it aims to provide valuable insights for decision-making and business improvement. You can find the dataset here.

first let's load some libraries and read The data # Load required libraries

```
library(tidyr)
library(readxl)
library(ggplot2)
library(scales)
library(dplyr)
```

Read the dataset

```
data <- read_xlsx("D:/My job/Sales_analysis/excel/Project
Data/Combined_Data.xlsx")
head(data)
```

```
## # A tibble: 6 × 6
##   `Order ID` Product         `Quantity Ordered` `Price Each` `Order Date`
##        <dbl> <chr>                        <dbl>        <dbl> <dttm>
## 1     319670 Bose SoundSpor…                  1       100.   2019-12-21
21:45:00
## 2     319669 Wired Headphon…                  1        12.0  2019-12-03
10:39:00
## 3     319668 Vareebadd Phone                  1       400    2019-12-09
06:43:00
## 4     319667 AA Batteries (…                  2         3.84 2019-12-01
12:01:00
## 5     319666 Lightning Char…                  1        15.0  2019-12-11
20:58:00
## 6     319665 iPhone                           1       700    2019-12-15
11:13:00
## # ℹ 1 more variable: `Purchase Address` <chr>
```

Data Cleaning and Preparation

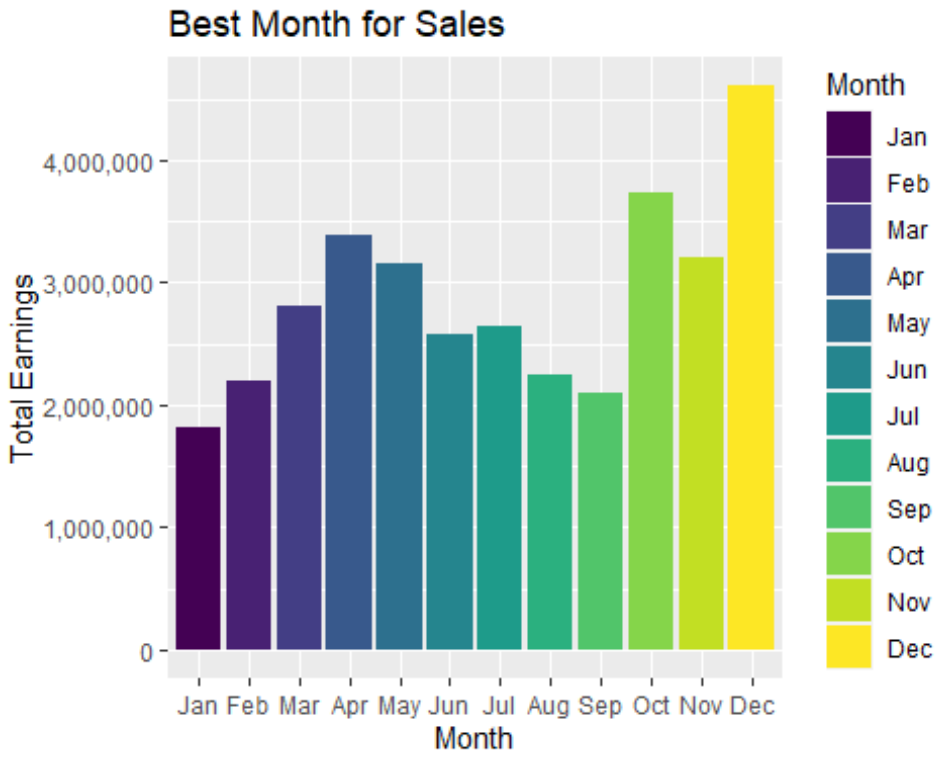
```r
# Remove blanks and duplicates
cleaned_data <- data %>%
  na.omit() %>%
  distinct()

# Split address column into street, city, and zip code
cleaned_data <- cleaned_data %>%
  separate(`Purchase Address`, into = c("Street", "City", "ZipCode"), sep =
", ")

# Calculate total price
cleaned_data <- cleaned_data %>%
  mutate(TotalPrice = `Quantity Ordered` * `Price Each`)
```

Q1: Find the best month for sales

```r
# Group data by year and month, then calculate total earnings
monthly_earnings <- cleaned_data %>%
  group_by( Month = lubridate::month(`Order Date`, label = TRUE)) %>%
  summarize(TotalEarnings = sum(TotalPrice)) %>%
  arrange(desc(TotalEarnings))
#The Best month
best_month <- monthly_earnings %>%
  slice(1)
print(best_month)
```

```
## # A tibble: 1 × 2
##   Month TotalEarnings
##   <ord>         <dbl>
## 1 Dec        4608296.
```

```r
# Visualize best months for sales

ggplot(monthly_earnings, aes(x = Month, y = TotalEarnings, fill = Month)) +
  geom_bar(stat = "identity") +
  labs(title = "Best Month for Sales",
       x = "Month",
       y = "Total Earnings")+
  scale_y_continuous(labels = scales::comma)  # Use comma separator for large numbers
```



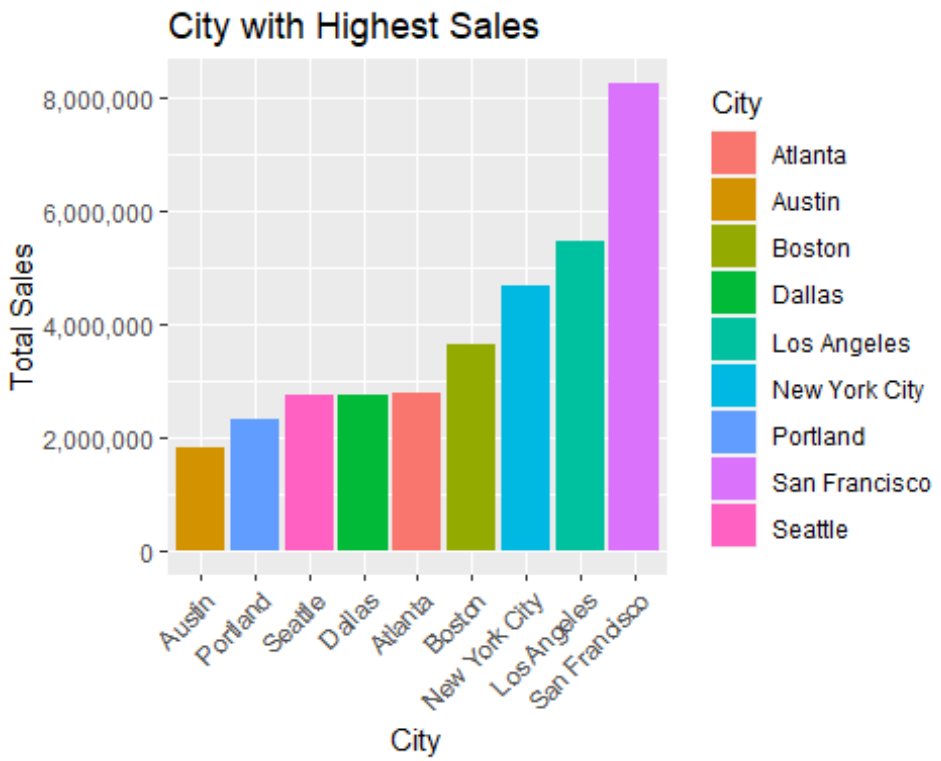
Q2: City with the highest number of sales

```r
highest_sales_city <- cleaned_data %>%
  group_by(City) %>%
  summarize(TotalSales = sum(TotalPrice)) %>%
  arrange(desc(TotalSales))
print(highest_sales_city)
```

```
## # A tibble: 9 × 2
##   City          TotalSales
##   <chr>              <dbl>
## 1 San Francisco   8254744.
## 2 Los Angeles     5448304.
## 3 New York City   4661867.
## 4 Boston          3658628.
## 5 Atlanta         2794199.
## 6 Dallas          2765374.
## 7 Seattle         2745046.
## 8 Portland        2319332.
## 9 Austin          1818044.
```

Visualize city with highest sales

```
ggplot(highest_sales_city, aes(x = reorder(City, TotalSales), y = TotalSales,
fill = City)) +
  geom_bar(stat = "identity") +
  labs(title = "City with Highest Sales",
       x = "City",
       y = "Total Sales") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  scale_y_continuous(labels = scales::comma)  # Use comma separator for large
numbers
```



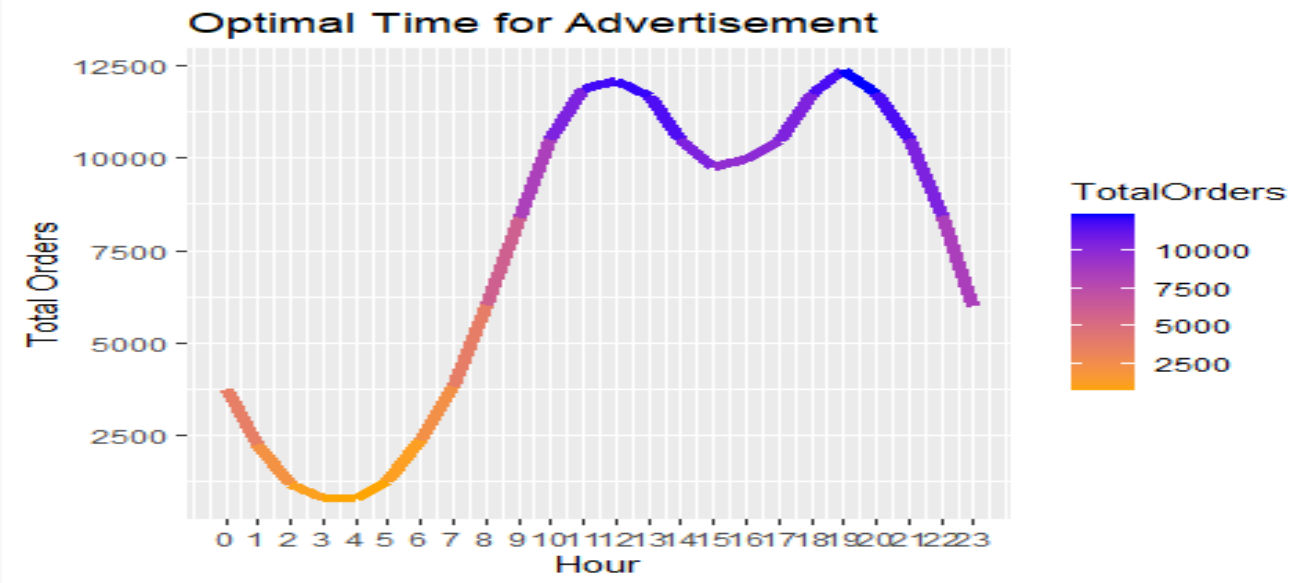
Q3: Optimal time for advertisement

```
optimal_ad_time <- cleaned_data %>%
  mutate(Hour = lubridate::hour(`Order Date`)) %>%
  group_by(Hour) %>%
  summarize(TotalOrders = n_distinct(`Order ID`)) %>%
  arrange(desc(TotalOrders))
print(optimal_ad_time)
```

```
## # A tibble: 24 × 2
##     Hour TotalOrders
##    <int>       <int>
##  1    19       12377
##  2    12       12082
##  3    11       11882
##  4    20       11763
##  5    18       11761
##  6    13       11682
##  7    14       10522
##  8    21       10499
##  9    10       10492
## 10    17       10476
## # ℹ 14 more rows
```

Visualize optimal advertisement time

```
# Visualize optimal advertisement time with a single color gradient
ggplot(optimal_ad_time, aes(x = Hour, y = TotalOrders, color = TotalOrders))
+
  geom_line(linewidth=2) +
  labs(title = "Optimal Time for Advertisement",
       x = "Hour",
       y = "Total Orders") +
  scale_color_gradient(low = "orange", high = "blue")+
  scale_x_continuous(breaks = seq(0, 23,by = 1))
```



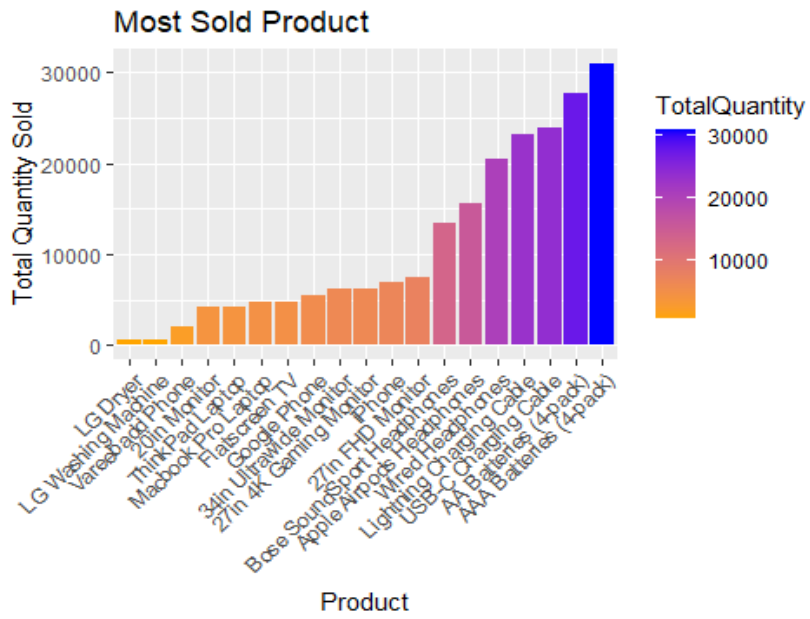
Q4: What product sold the most?

```
product_quantity <- cleaned_data %>%
  group_by(Product) %>%
  summarize(TotalQuantity = sum(`Quantity Ordered`)) %>%
  arrange(desc(TotalQuantity))
print(product_quantity)
```

```
## # A tibble: 19 × 2
##    Product                    TotalQuantity
##    <chr>                              <dbl>
##  1 AAA Batteries (4-pack)             30986
##  2 AA Batteries (4-pack)              27615
##  3 USB-C Charging Cable               23931
##  4 Lightning Charging Cable           23169
##  5 Wired Headphones                   20524
##  6 Apple Airpods Headphones           15637
##  7 Bose SoundSport Headphones         13430
##  8 27in FHD Monitor                    7541
##  9 iPhone                              6847
## 10 27in 4K Gaming Monitor              6239
## 11 34in Ultrawide Monitor              6192
## 12 Google Phone                        5529
## 13 Flatscreen TV                       4813
## 14 Macbook Pro Laptop                  4725
## 15 ThinkPad Laptop                     4128
## 16 20in Monitor                        4126
## 17 Vareebadd Phone                     2068
## 18 LG Washing Machine                   666
## 19 LG Dryer                             646
```

Visualize most sold product

```
ggplot(product_quantity, aes(x = reorder(Product, TotalQuantity), y =
TotalQuantity, fill = TotalQuantity)) +
  geom_bar(stat = "identity") +
  labs(title = "Most Sold Product",
       x = "Product",
       y = "Total Quantity Sold") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  scale_fill_gradient(low = "orange", high = "blue")
```

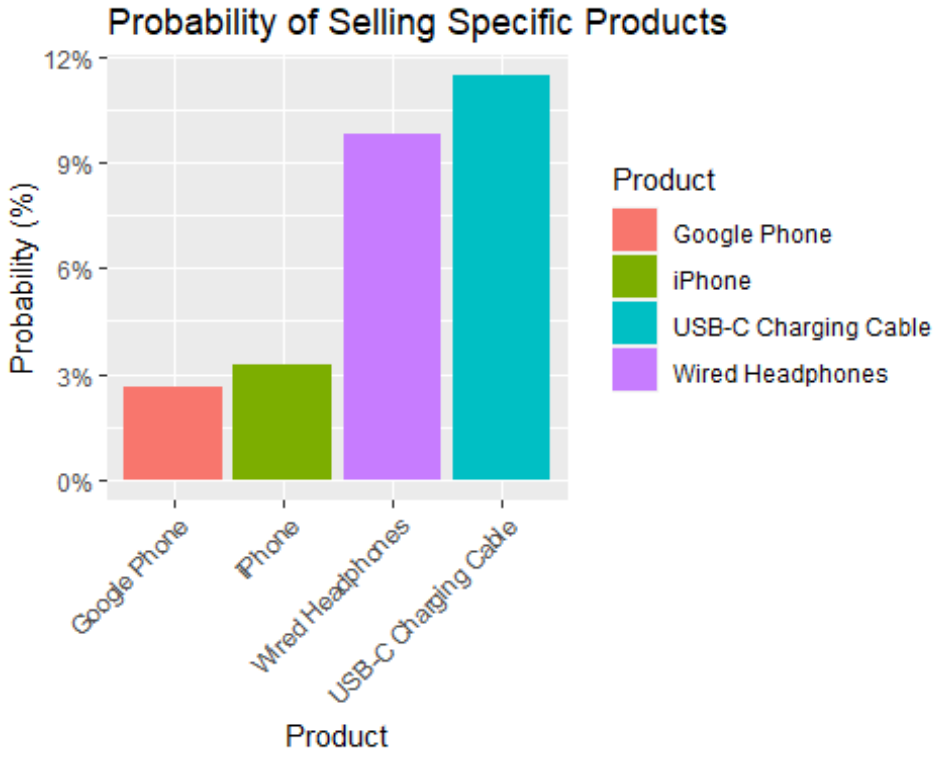


Q5: How Much Probability?

```
specific_orders <- c("USB-C Charging Cable", "iPhone", "Google Phone", "Wired
Headphones")

product_probs <- cleaned_data %>%
  filter(Product %in% specific_orders) %>%
  group_by(Product) %>%
  summarize(Probability = paste0(round(sum(`Quantity Ordered`) /
sum(cleaned_data$`Quantity Ordered`) * 100, 2), "%"))
print(product_probs)

## # A tibble: 4 × 2
##   Product              Probability
##   <chr>                <chr>
## 1 Google Phone         2.65%
## 2 USB-C Charging Cable 11.46%
## 3 Wired Headphones     9.83%
## 4 iPhone               3.28%
```

Visualize probabilities for specific products

```
# Clean the Probability column by removing percentage signs and converting to numeric
product_probs$Probability <- as.numeric(gsub("%", "", product_probs$Probability))

# Visualize probabilities for specific products with a color gradient
ggplot(product_probs, aes(x = reorder(Product, Probability), y = Probability, fill = Product)) +
  geom_bar(stat = "identity") +
  labs(title = "Probability of Selling Specific Products",
       x = "Product",
       y = "Probability (%)") +
  scale_y_continuous(labels = percent_format(scale = 1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
```