

COMPETITIVE ANALYSIS

Hamza

2023-08-10

MARKET ANALYSIS FOR PRODUCTS

This a product I have done for my client last year. * I have change some details for confidentiality of my client.

Introduction

I conduct a comprehensive competitive analysis of products from different brands, focusing on pricing, features, and market share. Visualize the findings to gain insights into the market landscape.

Data sets

There are three data sets used in this project **Products Competitors Product_Features**

Step 1:Load libraries and data sets

```
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(lubridate)
products<-read_csv("Products.csv")

## Rows: 10 Columns: 6
## -- Column specification -----
## Delimiter: ","
## chr  (3): product_name, brand, category
## dbl  (2): product_id, price
## date (1): launch_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

competitors<-read_csv("Competitors.csv")
product_features<-read_csv("Product_Features.csv")
```

```
## Rows: 20 Columns: 3
## -- Column specification -----
## Delimiter: ","
## chr (2): feature, feature_value
## dbl (1): product_id
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Step2: Explore the data

This chunk of code just give the overview of the data sets

```
glimpse(products)
```

```
## Rows: 10
## Columns: 6
## $ product_id    <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
## $ product_name  <chr> "Smartphone X", "Smartphone Y", "Smartwatch Z", "Laptop A~
## $ brand         <chr> "Brand A", "Brand B", "Brand A", "Brand C", "Brand A", "B~
## $ category      <chr> "Electronics", "Electronics", "Wearables", "Electronics",~
## $ price         <dbl> 699.99, 649.99, 299.99, 1199.99, 1099.99, 399.99, 349.99,~
## $ launch_date   <date> 2023-01-15, 2023-03-22, 2023-04-10, 2023-05-15, 2023-06-2~
```

```
glimpse(competitors)
```

```
## Rows: 5
## Columns: 4
## $ competitor_id <int> 1, 2, 3, 4, 5
## $ competitor_name <chr> "Competitor A", "Competitor B", "Competitor C", "Compe~
## $ industry       <chr> "Electronics", "Electronics", "Wearables", "Audio", "E~
## $ market_share   <dbl> 30.50, 25.75, 15.60, 12.40, 10.75
```

```
glimpse(product_features)
```

```
## Rows: 20
## Columns: 3
## $ product_id    <dbl> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10~
## $ feature       <chr> "Battery Life", "Screen Size", "Battery Life", "Screen S~
## $ feature_value <chr> "24 hours", "6.5 inches", "20 hours", "6.1 inches", "48 ~
```

Step3: Clean the data

This chunk of code will change the character/string data type to the date data type.

```
products<-products %>%
  mutate(launch_date=as.Date(launch_date))
```

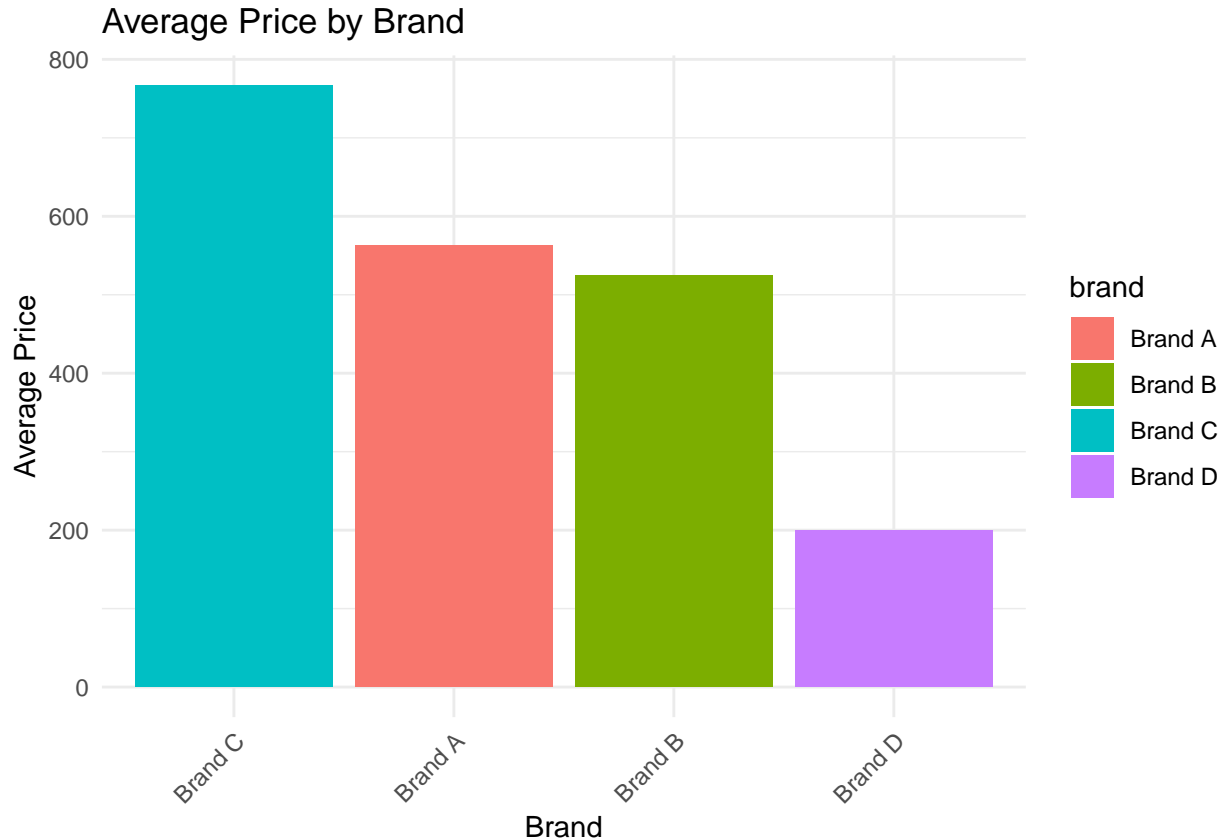
Step4: Analyze Pricing Strategies

Comparing the average price of products by brand

```
pricing_comparison<-products %>%
  group_by(brand) %>%
  summarise(average_price=mean(price,na.rm=TRUE)) %>%
  arrange(desc(average_price))

pricing_comparison %>%
```

```
ggplot(aes(x = reorder(brand, -average_price), y = average_price, fill = brand)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Price by Brand", x = "Brand", y = "Average Price") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

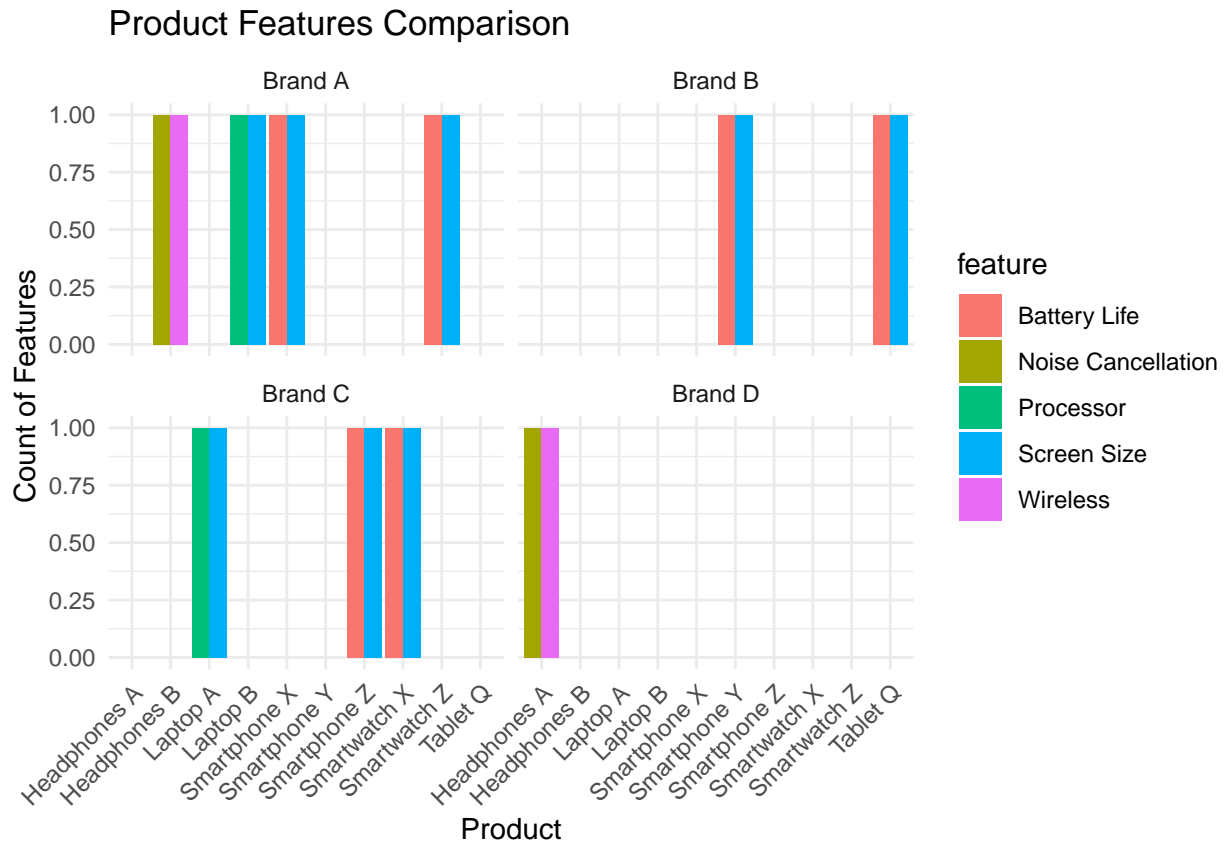


Step5: Feature Comparison

Comparing the features across products

```
Feature_comparison <- product_features %>%
  inner_join(products, by = "product_id") %>%
  select(product_name, brand, feature, feature_value) %>%
  arrange(brand, feature)

Feature_comparison %>%
  ggplot(aes(x = product_name, fill = feature)) +
  geom_bar(position = "dodge") +
  labs(title = "Product Features Comparison", x = "Product", y = "Count of Features") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_wrap(~brand)
```



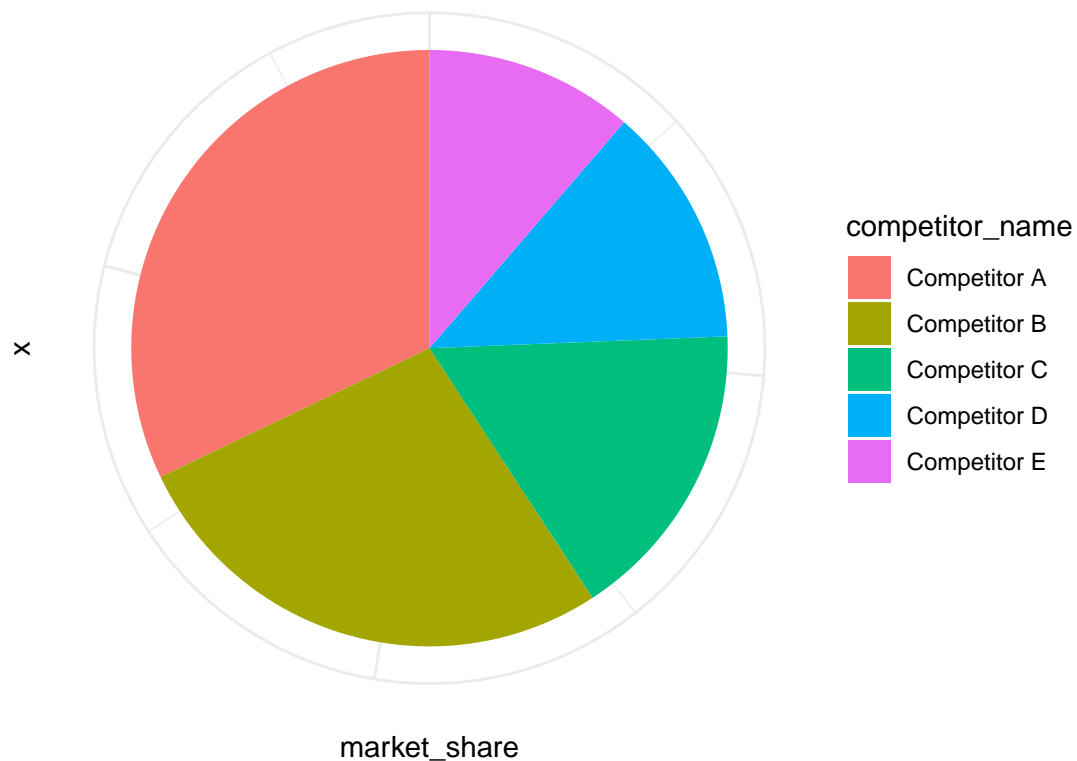
Step6: Market Share Analysis

Analyzing the market share of competitors

```
market_share_analysis <- competitors %>%
  arrange(desc(market_share))

market_share_analysis %>%
  ggplot(aes(x = "", y = market_share, fill = competitor_name)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar(theta = "y") +
  labs(title = "Market Share Distribution by Competitor") +
  theme_minimal() +
  theme(axis.text.x = element_blank())
```

Market Share Distribution by Competitor



Step7: Product Launch Trends

Analyzing product launch trends by year

```
launch_trends <- products %>%
  mutate(launch_year = year(launch_date)) %>%
  group_by(launch_year) %>%
  summarise(product_launched = n()) %>%
  arrange(launch_year)

launch_trends %>%
  ggplot(aes(x = launch_year, y = product_launched)) +
  geom_line(color = "blue") +
  geom_point(color = "blue") +
  labs(title = "Product Launch Trends by Year", x = "Year", y = "Number of Products Launched") +
  theme_minimal()
```

`geom_line()`: Each group consists of only one observation.
 ## i Do you need to adjust the group aesthetic?

