Amazon Case Study

Vivek limbad sdbi

Amazon Case Study- Company Overview

- **Amazon is a Global E-Commerce Giant. It is an Internet-based company that sells electronic goods, apparel, movie books and every good that can be sold online on its Platform Amazon.com. Amazon was founded by Jeff Bezos in 1994.**
- **Mobile phones have revolutionized the way we purchase products online, making all the information available at our fingertips. As the access to information becomes easier, more and more consumers will seek product information from other consumers apart from the information provided by the seller. Reviews and ratings submitted by consumers are examples of such of type of information and they have already become an integral part of customer's buying-decision process. The review and ratings platform provided by eCommerce players creates transparent system for consumers to take informed decision and feel confident about it.**
- **Amazon.com is a treasure trove of product reviews and their review system is accessible across all channels presenting reviews in an easy-to-use format. The product reviewer submits a rating on a scale of 1 to 5 and provides own viewpoint according to the whole experience. The mean value is calculated from all the ratings to arrive at the final product rating. Others can also mark yes or no to a review depending on its helpfulness - adding credibility to the review and reviewer. In this study, we analysed more than 400 thousand reviews of unlocked mobile phones sold on Amazon.com to find insights with respect to reviews, ratings, price and their relationships.**

DATA

- **We extracted the following information from the 'unlocked phone' category of Amzon.com:**
- **Product Title**
- **Brand**
- **Price**
- **Rating**
- **Review text**
- **Number of people who found the review helpful**

OUR GOAL

This statistical analysis had the following goals:

- **Perform exploratory analysis of ratings and reviews**
- **Find out relationship between price and the number of reviews**
- **Find out relationship between helpfulness of review and length of review**
- **Find out relationship between review length and product price**
- **Find out relationship between review length and product rating**
- **Find out relationship between product price and product rating**
- **Word cloud of most-used words**
- **Sentiment analysis**

Loading the library

- **library(tidyverse) <- The tidyverse is an opinionated collection of R packages designed for data science**
- **library(ggplot2) <- ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics.**
- **library(ggthemes) <- Some extra themes, geoms, and scales for ‘ggplot2’.**
- **library(tidytext) <- Using tidy data principles can make many text mining tasks easier, more effective, and consistent with tools already in wide use.**
- **library(plotly) <- Plotly’s Python graphing library makes interactive, publication-quality graphs.**
- **library(readr) <- The goal of readr is to provide a fast and friendly way to read rectangular data (like csv, tsv, and fwf).**
- **library(extrafont) <- The extrafont package makes it easier to use fonts other than the basic PostScript fonts that R uses.**
- **library(stopwords) <- Provides multiple sources of stopwords, for use in text analysis and natural language processing.**

```
library(tidyverse)
library(ggplot2)
library(ggthemes)
library(tidytext)
library(plotly)
library(readr)
library(extrafont)
library(stopwords)

loadfonts(device="win")
```

Loading the datasets

```
items <- read_csv("C:\\Users\\Lenovo\\Downloads\\items.csv")
reviews <- read_csv("C:\\Users\\Lenovo\\Downloads\\20191226-reviews.csv\\reviews.csv")
```

Data

- Exploring the datasets

To find missing values * If the value is NA the is.na() function return the value of true, otherwise, return to a value of false.

```
sapply(items, function(x) sum(is.na(x)))
```

```
##          asin         brand         title           url         image
##             0             4             0             0             0
##        rating     reviewUrl  totalReviews         price originalPrice
##             0             0             0             0             0
```

```
sapply(reviews, function(x) sum(is.na(x)))
```

```
##         asin        name      rating        date    verified       title
##            0           1           0           0           0           2
##         body helpfulVotes
##           13       40771
```

- **Dropping only NA's in items because 4 have not brand names**

```
items <- na.omit(items)
max(items$rating)
```

```
## [1] 5
```

- **Renaming the columns**

```
names(reviews)[names(reviews)== "rating"] <- "reviewer_rating"
names(reviews)[names(reviews)=="title"] <- "review_title"
names(items)[names(items)=="rating"] <- "product_rating"
names(items)[names(items)=="title"] <- "product"
```

- **Merging dataset**

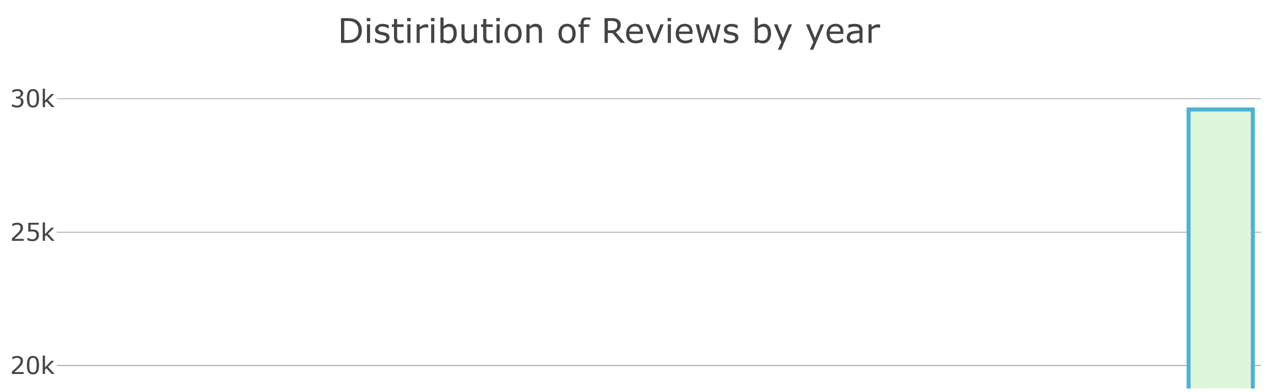
```
amazon$verified <- as.factor(amazon$verified)
```

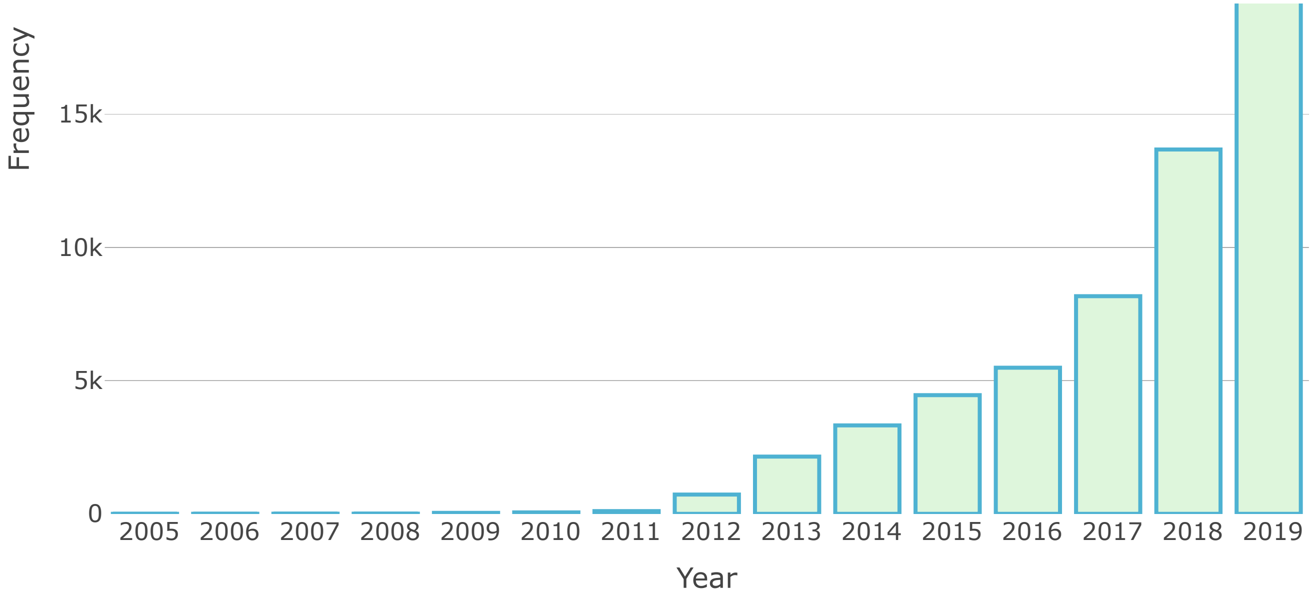
- **Data Column into Daymonth and year (2 Columns) years between 2005-2018**

Descriptive Analysis

Q.1 Distiribution of Reviews by year

```
fig <- plot_ly(amazon, x=~year,  type = "histogram",
      marker = list(color = "#def6dc",line = list(color = "#4eb2d2",width = 2))) %>%
      layout(title = "Distiribution of Reviews by year",
      yaxis = list(title = "Frequency",zeroline = FALSE),
      xaxis = list(title = "Year",zeroline = FALSE))
fig
```

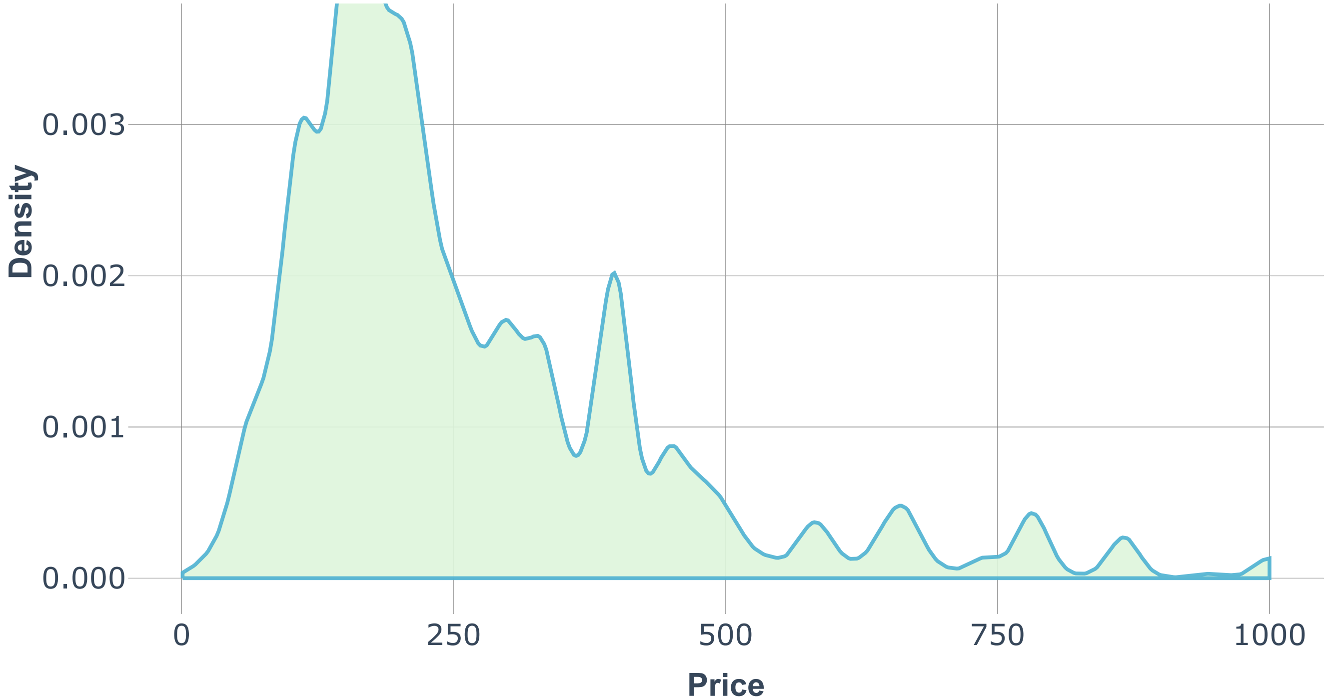




Q.2 Distribution of Price

```
amazonp <- amazon %>% filter(price != 0.00)

two <- ggplot(amazonp, aes(x = price)) +
       geom_density(alpha = 0.9,color="#4eb2d2", fill="#def6dc") +
       labs(x = "Price" , y = "Density") +
       theme_minimal() +
       theme(plot.title = element_text(hjust = 0.5 , color = "#37475A"),
       axis.title.x = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
       axis.title.y = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
       axis.text = element_text(size = 11 , color = "#37475A")) +
       ggtitle("Distribution of Price")
fig <- ggplotly(two)
fig
```

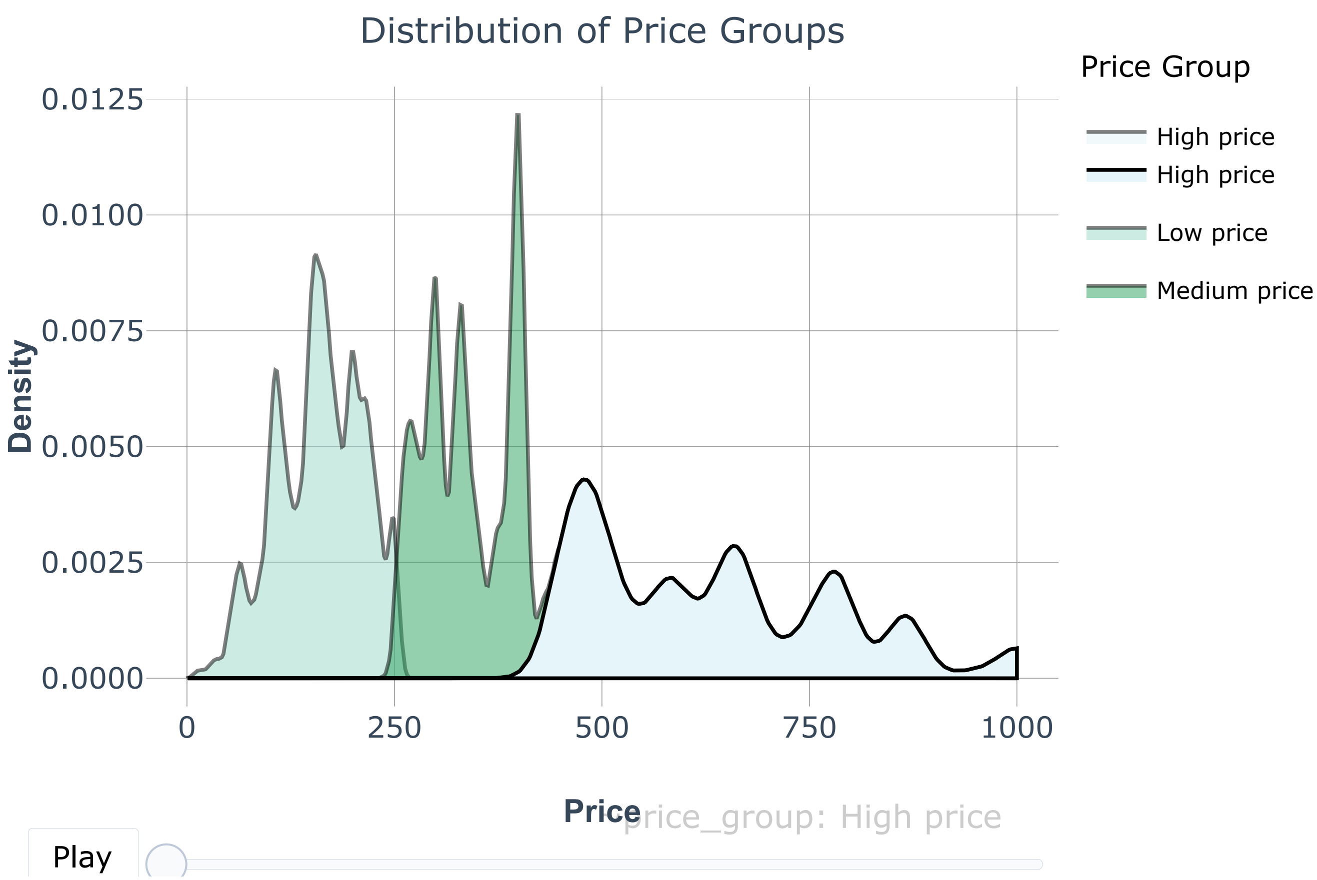


Q.3 Grouping Price

```
amazonp <- amazonp %>%
          mutate(price_group = if_else(between(price, 0, 250), "Low price",
          if_else(between(price, 250, 450), "Medium price",
          if_else(price > 450, "High price","Unknown price"))),
          price_group = if_else(is.na(price_group), "Unknown price", price_group)) %>%
          rownames_to_column(var = "id")
```

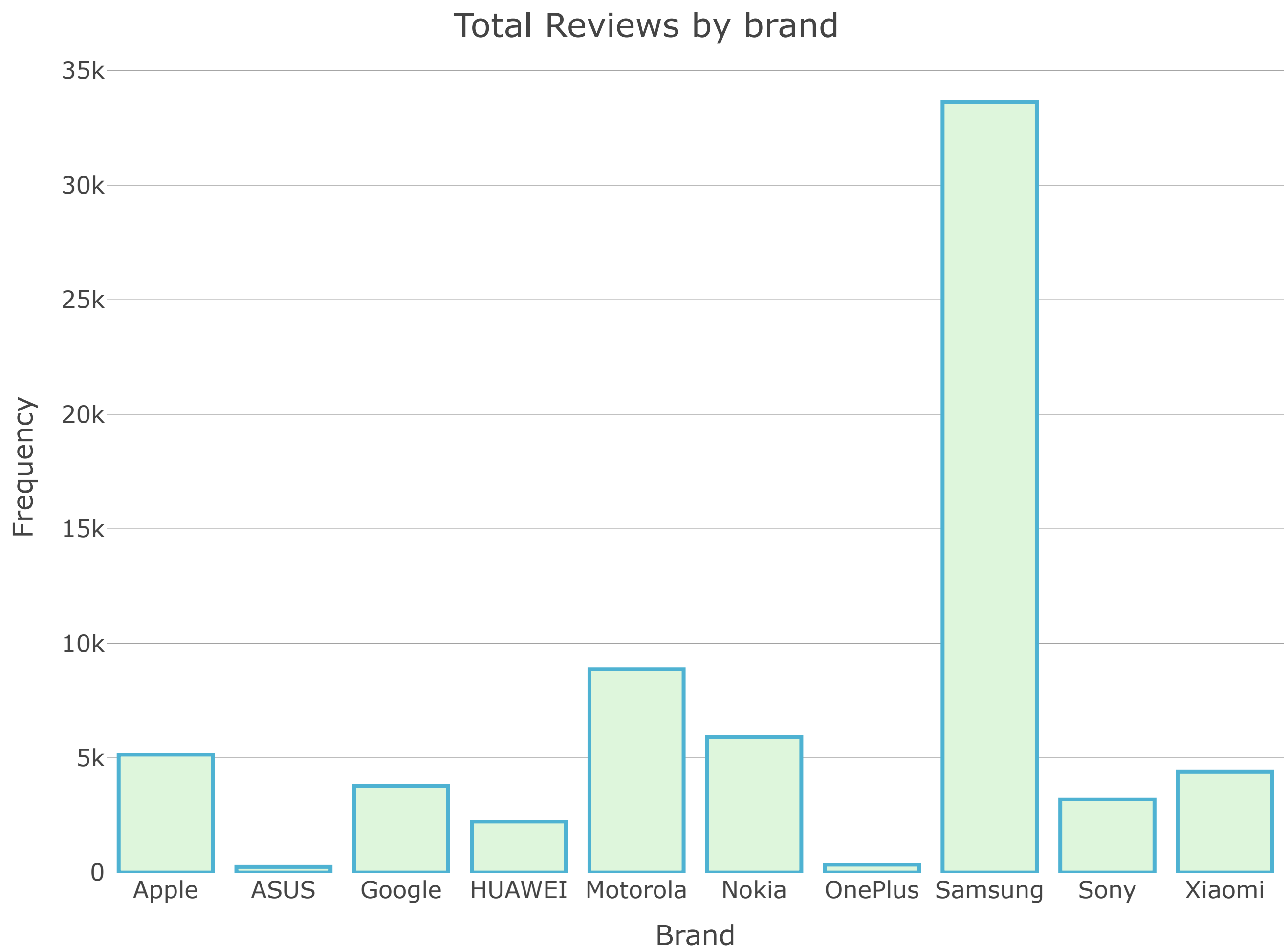
```
p_group_cl <- amazonp %>% dplyr::filter(!is.na(price_group)) %>%
              ggplot(aes(x = price,fill = price_group)) +
              labs(x = "Price" , y = "Density" , fill = "Price Group") +
              geom_density(alpha = 0.5) +
      scale_fill_brewer(palette="BuGn") +
              theme_minimal() +
              theme(plot.title = element_text(hjust = 0.5 , color = "#37475A"),
              axis.title.x = element_text(color = "#37475A", size = 12,face ="bold",family="Arial"),
              axis.title.y = element_text(color = "#37475A", size = 12,face ="bold",family="Arial"),
              axis.text = element_text(size = 11 , color = "#37475A")) +
              geom_density(aes(frame = price_group))  +
              labs(title = "Distribution of Price Groups")

fig <- ggplotly(p_group_cl)
fig
```



Q.4 Distribution of total reviews by brand

```
fig <- plot_ly(amazon, x=~brand,  type = "histogram",
       marker = list(color = "#def6dc",line = list(color = "#4eb2d2",width = 2))) %>%
       layout(title = "Total Reviews by brand",
       yaxis = list(title = "Frequency",zeroline = FALSE),
       xaxis = list(title = "Brand",zeroline = FALSE))
fig
```



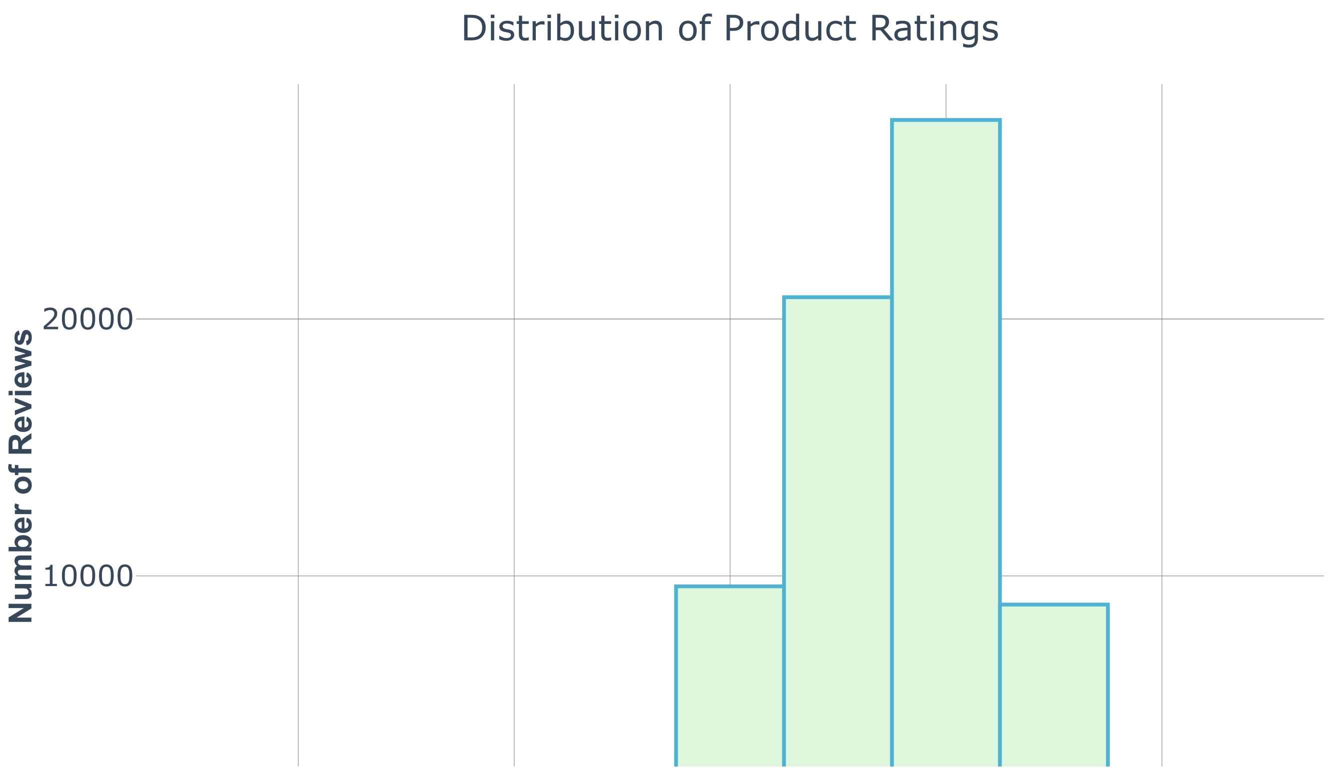
Exploratory Analysis based on Rating Distribution

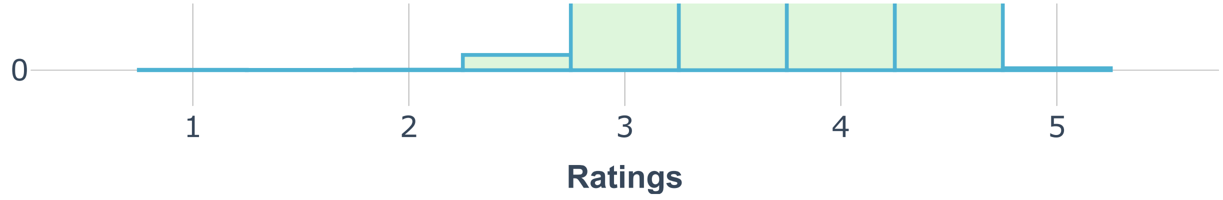
Q.5 Ratings vs. Number of reviews.

- **let's look at the distribution of ratings among the reviews. Most of the reviewers have given 4-star and 3-star rating with relatively very few giving 1-star rating.**

```
five <- ggplot(amazon, aes(x=product_rating)) + geom_histogram(binwidth = 0.5,color="#4eb2d2", fill="#def6dc") +         coo
rd_cartesian(xlim = c(.5, 5.5))+
        labs(title ="Distribution of Product Ratings ",
        x = "Ratings", y = "Number of Reviews") +
        theme_minimal() +
        theme(plot.title = element_text(hjust = 0.5 , color = "#37475A"),
        axis.title.x = element_text(color = "#37475A", size = 12,face ="bold",family="Arial"),
        axis.title.y = element_text(color = "#37475A", size = 12,face ="bold",family="Arial"),
        axis.text = element_text(size = 11 , color = "#37475A"))

fig <- ggplotly(five)
fig
```





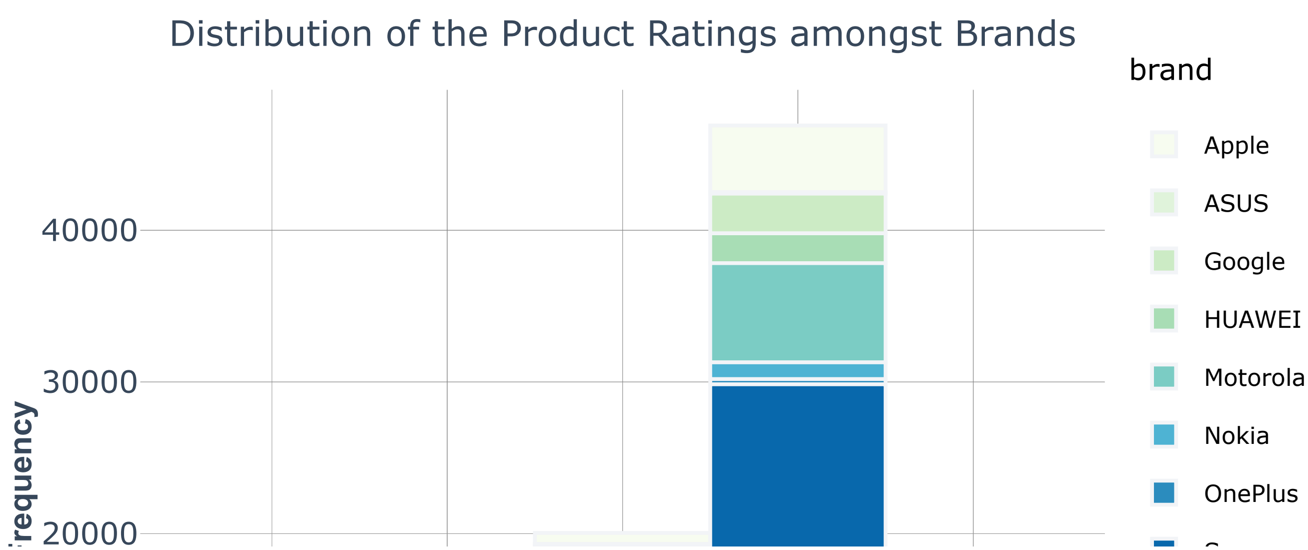
Conclusion

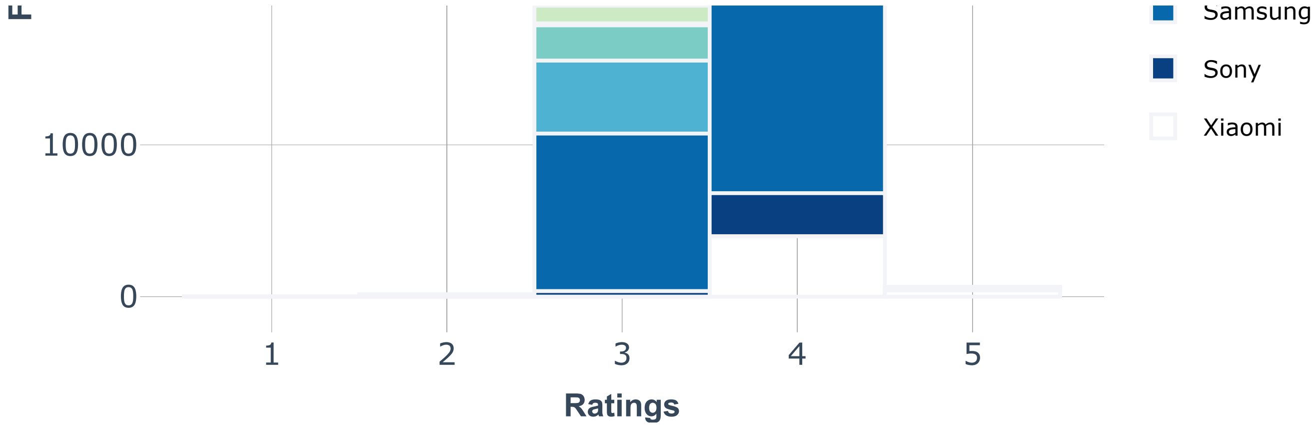
The mean value of all the ratings comes to 3.62.

Q.6 Ratings fill by brand

```
six <- ggplot(amazon, aes(x = product_rating,fill = `brand` )) +
       geom_histogram (binwidth = 1, col = "#f2f4f7" , stat="bin") +
       labs(x = "Ratings", y = "Frequency",title = "Distribution of the Product Ratings amongst Brands") +
       scale_x_continuous(breaks = c(1,2, 3,4, 5)) +
    scale_fill_brewer(palette="GnBu") +
       theme_minimal() +
       theme(plot.title = element_text(hjust = 0.5 , color = "#37475A"),
       axis.title.x = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
       axis.title.y = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
       axis.text = element_text(size = 11.5 , color = "#37475A"))

fig <- ggplotly(six)
fig
```



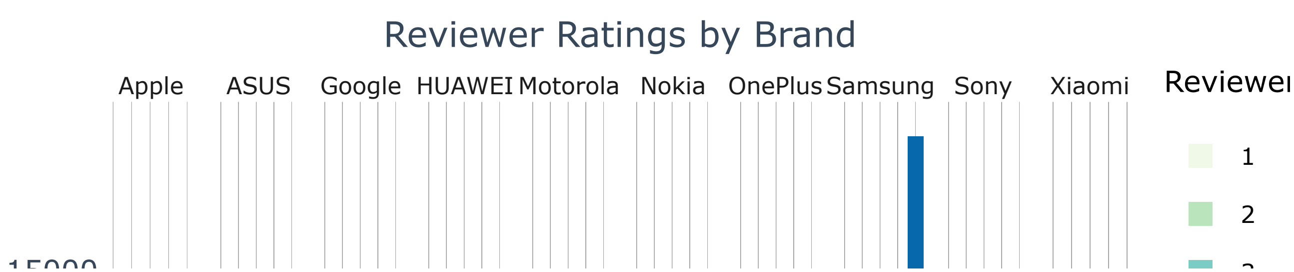


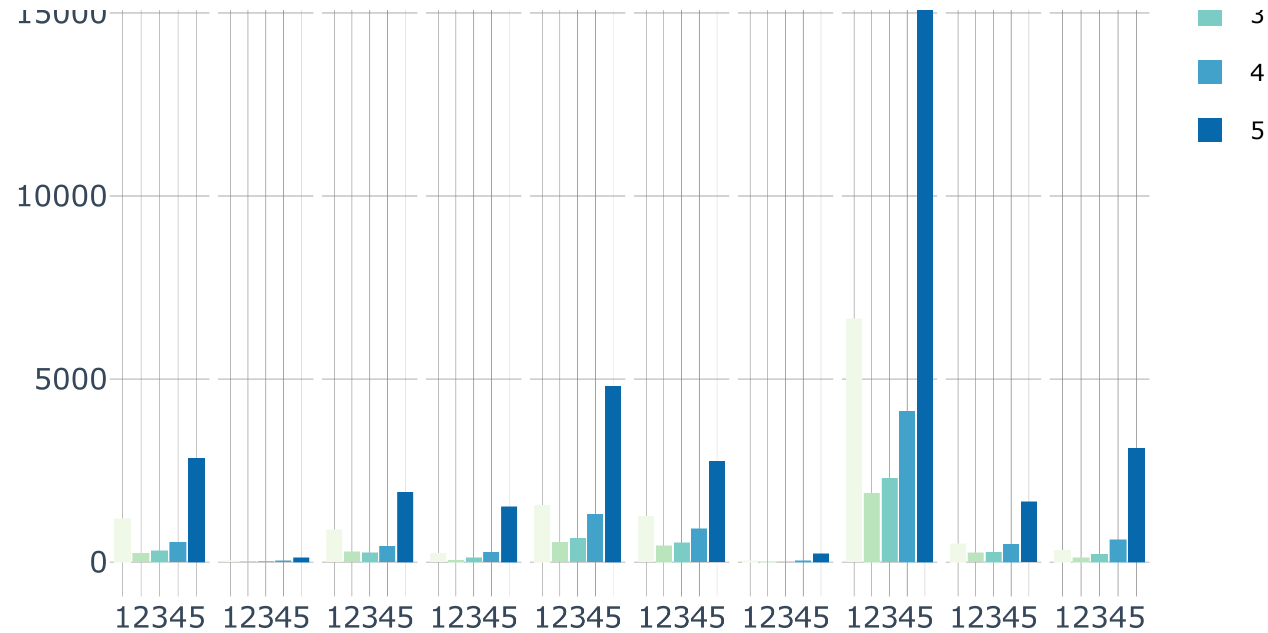
"IF YOU WANT TO CONTROL YOUR BRAND PRESENCE ONLINE YOU HAVE TO CONTROL WHAT IT LOOKS LIKE ON AMAZON."

Q.7 Reviewer Ratings by Brand

```
nine2 <- ggplot(amazon, aes(x=reviewer_rating, group=brand))+
  geom_bar(aes(fill=factor(..x..)),stat="count")+
  facet_grid(~brand)+
  labs(x=NULL,y=NULL,title="Reviewer Ratings by Brand")+
  ylab(NULL) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5 , color = "#37475A"),
        axis.title.x = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
        axis.title.y = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
        axis.text = element_text(size = 11 , color = "#37475A")) +
  scale_fill_brewer(name="Reviewer Ratings",palette="GnBu",label=c("1", "2", "3", "4", "5"))

fig <- ggplotly(nine2)
fig
```





Conclusion

- **When consumers pay more for a product, they also expect better quality and sellers need to meet this expectation.**
- **It can be considered that with cost the product quality increases, which in turn leads to higher rating.**

```
price <- amazonp %>%
        mutate(rating_group = if_else(between(product_rating, 0, 1), "1",
        if_else(between(product_rating, 1, 2), "2",
        if_else(between(product_rating, 2, 3), "3",
        if_else(between(product_rating, 3, 4), "4",
        if_else(product_rating > 5, "5","5"))))),
        rating_group = if_else(is.na(rating_group), "", rating_group)) %>%
        rownames_to_column(var = "idd")
```

```
price <- na.omit(price)
```

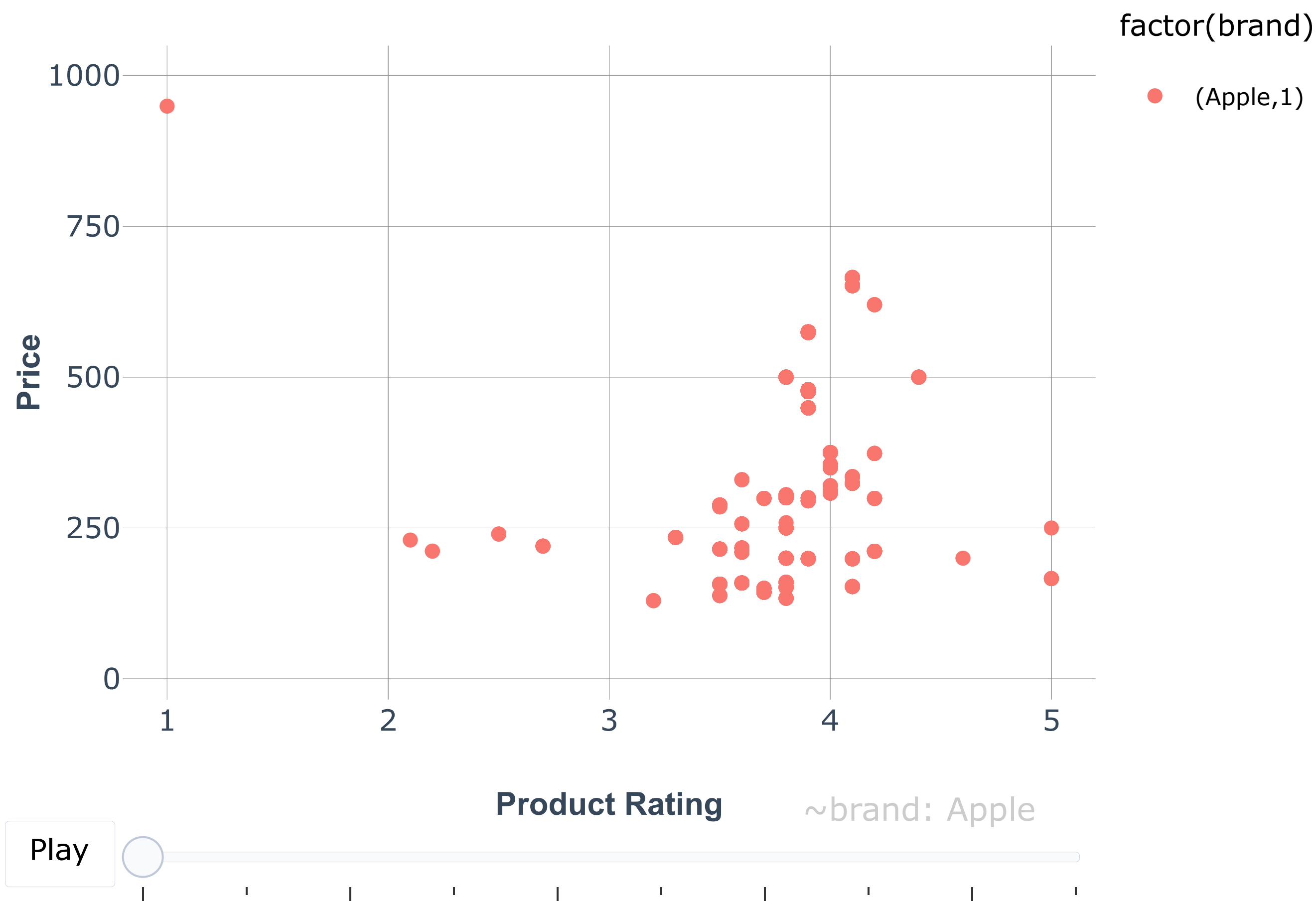
Q.8 Price and Brand**

- **Let’s now try to explore correlation between product price and number of reviews.**

- **This will help us answer questions like: Do expensive products receive more number of reviews?**

```
tanb <- ggplot(price, aes(product_rating, price, colour=factor(brand)), na.rm=TRUE) +
        theme_minimal() +
        labs(x="Product Rating",y="Price")+
        theme(plot.title = element_text(hjust = 0.5 , color = "#37475A"),
        axis.title.x = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
        axis.title.y = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
        axis.text = element_text(size = 11 , color = "#37475A")) +
        geom_point(aes(frame = brand , colour=factor(brand))) +
        scale_fill_discrete(name="Brand")

fig <- ggplotly(tanb)
fig
```



Apple Google Motorola OnePlus Sony

Conclusion

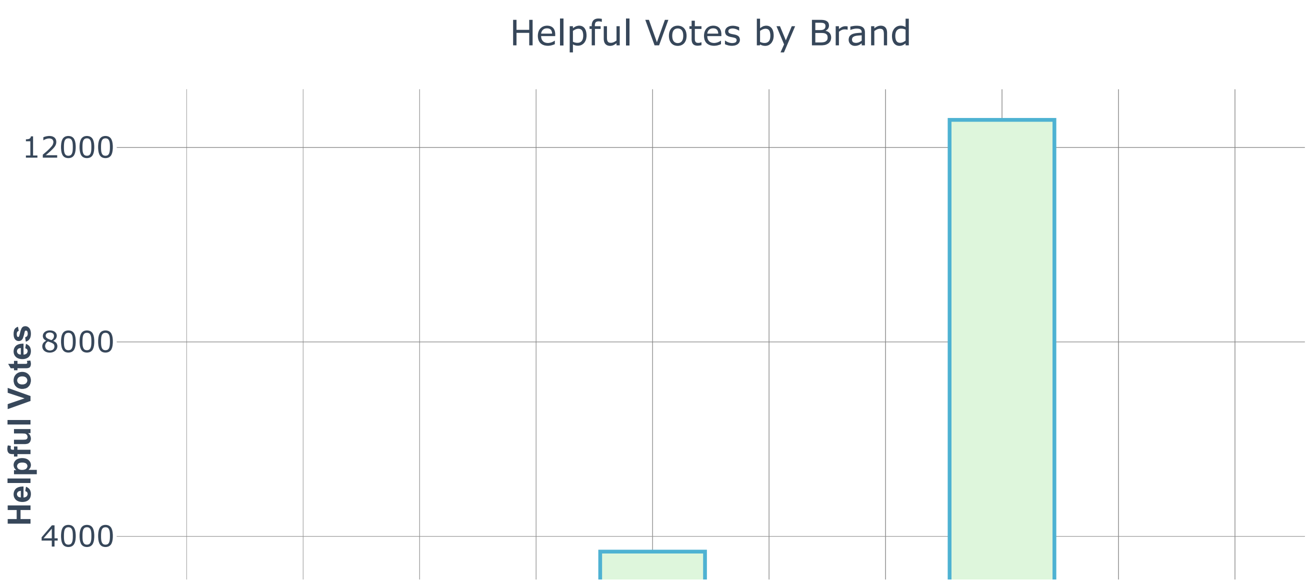
- **The scatter above says not necessarily.**
- **So there is no relationship between price and the number of reviews it gets.**

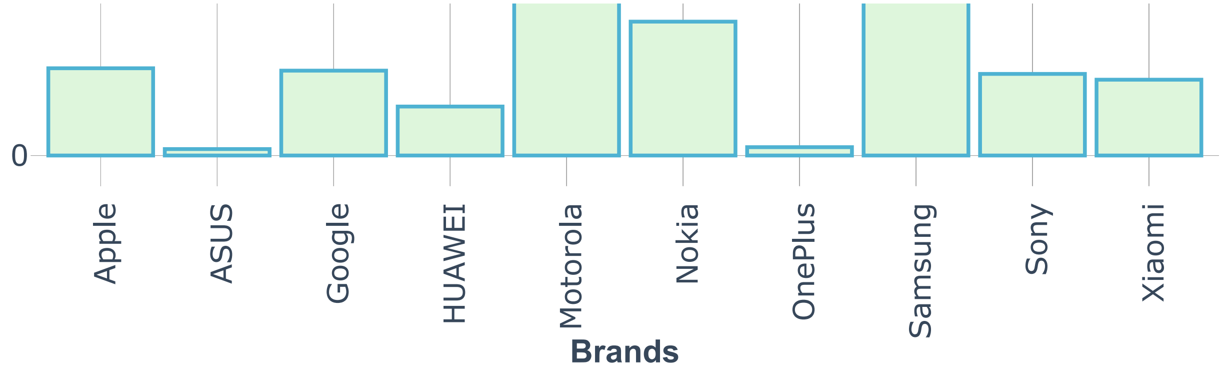
Q.9 Helpful Votes by Brand

```
a <- amazon %>% select(brand, helpfulVotes) %>% na.omit()
```

```
elev <- ggplot(a, aes(factor(brand))) +
        geom_bar(position = "dodge", color="#4eb2d2", fill="#def6dc") +
        ylab("Helpful Votes")+ xlab("Brands") + labs(title = "Helpful Votes by Brand")+
        theme_minimal() +
        theme(plot.title = element_text(hjust = 0.5 , color = "#37475A"),
        axis.title.x = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
        axis.title.y = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
        axis.text.y = element_text(size = 11 , color = "#37475A"),
        axis.text.x = element_text(size = 11 , color = "#37475A",angle = 90))

fig <- ggplotly(elev)
fig
```






```
data(stop_words)
stopwords_phone <- c(stopwords("english"),"phone","Samsung","Nokia","Apple","ASUS","OnePlus","Motorola","HUAWEI","Sony","Goo
gle","Xiaomi","great","like","good",
                     "samsung","nokia","iphone","apple","asus","oneplus","motorola","huawei","sony","google","xiaomi", "phon
es")
```

```
amazon_clean<- amazon %>% select(brand,body) %>% unnest_tokens(input=body,output=word) %>%
               count(brand,word,sort=T) %>% filter(nchar(word)>3) %>%
               filter(!word %in% stopwords_phone) %>%
               group_by(brand)
```

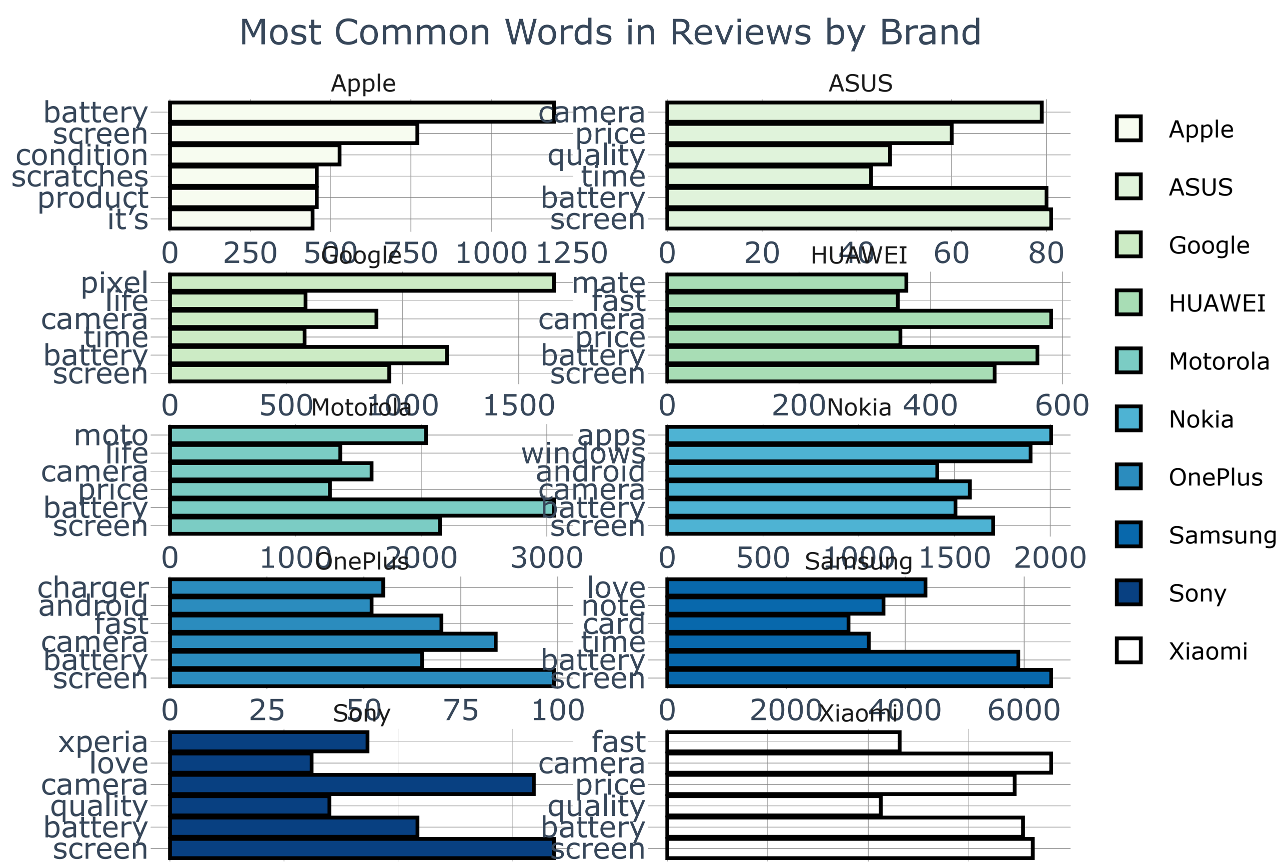
```
amazon_clean2 <- amazon_clean %>% anti_join(stop_words)
```

```
## Joining, by = "word"
```

Q.10 Most common words by brand

```
twl <- amazon_clean2 %>% top_n(n=6,n)%>% mutate(word = reorder(word, n)) %>%
       ggplot(aes(x=word, y=n,fill=brand)) +
       geom_col(show.legend=F,col="black")+
       facet_wrap(~brand, ncol=2,scales="free")+
       xlab(NULL)+ ylab("Count")+
       ggtitle("Most Common Words in Reviews by Brand") +
       scale_fill_brewer(palette="GnBu") +
       theme_minimal() +
       theme(plot.title = element_text(hjust = 0.5 , color = "#37475A"),
       axis.title.x = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
       axis.title.y = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
       axis.text = element_text(size = 11 , color = "#37475A")) + coord_flip()

fig <- ggplotly(twl)
fig
```




```
amazon_clean_rank<- amazon %>% select(reviewer_rating,body) %>%  unnest_tokens(input=body,output=word) %>%
                    count(reviewer_rating,word,sort=T) %>%
                    filter(nchar(word)>3) %>%
                    filter(!word %in% stopwords_phone) %>%
                    group_by(reviewer_rating)
```

```
amazon_clean_rank2 <- amazon_clean_rank %>% anti_join(stop_words)
```

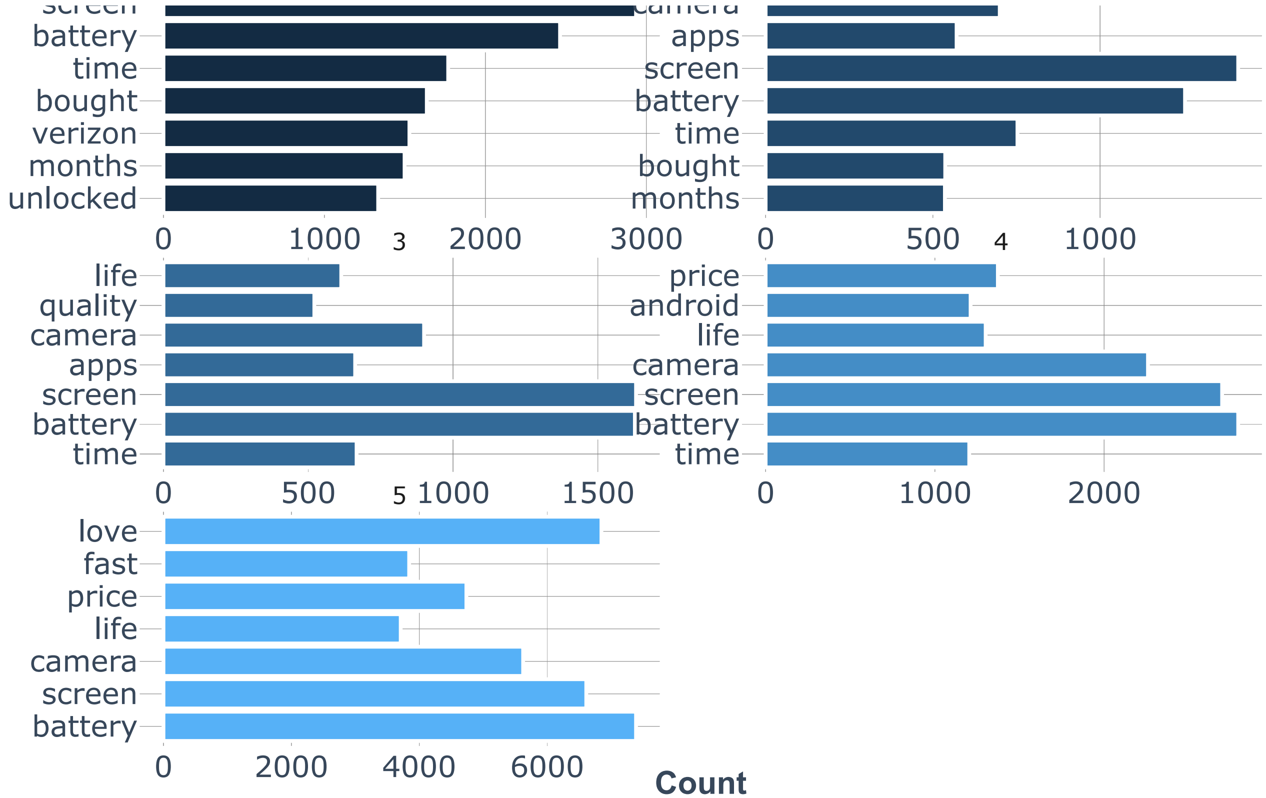
```
## Joining, by = "word"
```

Q.11 Most common words all reviews

- **We segregated the reviews according to their ratings - positive reviews (4 or 5 star) and negative reviews (1 or 2 star).**
- **In both type of reviews there are certain common words like “work”, “battery” and “screen”. The most frequently used words in positive reviews are: “great”, “good”, “camera”, “price”, “excellent”, etc. In case of negative reviews words such as “return”, “back”, “problem”, “charge” are prevalent.**

```
fif <- amazon_clean_rank2 %>% top_n(n=7,n)%>% mutate(word = reorder(word, n)) %>%
       ggplot(aes(x=word, y=n,fill=reviewer_rating)) +
       geom_col(show.legend=F,col="white")+
       facet_wrap(~reviewer_rating, ncol=2,scales="free")+
       xlab(NULL) + ylab("Count")+
       ggtitle("Most Common Words in Reviews by Rating") +
       theme_minimal() +
       theme(plot.title = element_text(hjust = 0.5 , color = "#37475A"),
       axis.title.x = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
       axis.text = element_text(size = 11 , color = "#37475A")) + coord_flip()

fig <- ggplotly(fif)
fig
```



CONCLUSION

- **Amazon's product review platform shows that most of the reviewers have given 4-star and 3-star ratings to unlocked mobile phones.**
- **We also uncovered that lengthier reviews tend to be more helpful and there is a positive correlation between price & rating. Sentiment analysis shows that positive sentiment is prevalent among the reviews and in terms of emotions, 'trust', 'anticipation' and 'joy' have highest scores.**
- **It'd be interesting to perform further analysis based on the brand (example: Samsung vs. Apple).**
- **We can also look at building a model to predict the helpfulness of the review and the rating based on the review text.**
- **Corpus-based and knowledge-based methods can be used to determine the semantic similarity of review text.**
- **There are many insights to be unveiled from the Amazon reviews.**

Samsung Vs. Apple

```
samsung <- amazon %>% filter(brand == "Samsung")

samsung2 <- amazon %>% filter(year >= 2017)

samsung_apple <- amazon %>% filter(brand == "Samsung"  | brand== "Apple")

samsung_apple %>% filter(year >= 2017)
```

```
##  [1] asin            name            reviewer_rating daynmonth
##  [5] year            verified        review_title    body
##  [9] helpfulVotes    brand           product         url
## [13] image           product_rating  reviewUrl       totalReviews
## [17] price           originalPrice
## <0 rows> (or 0-length row.names)
```

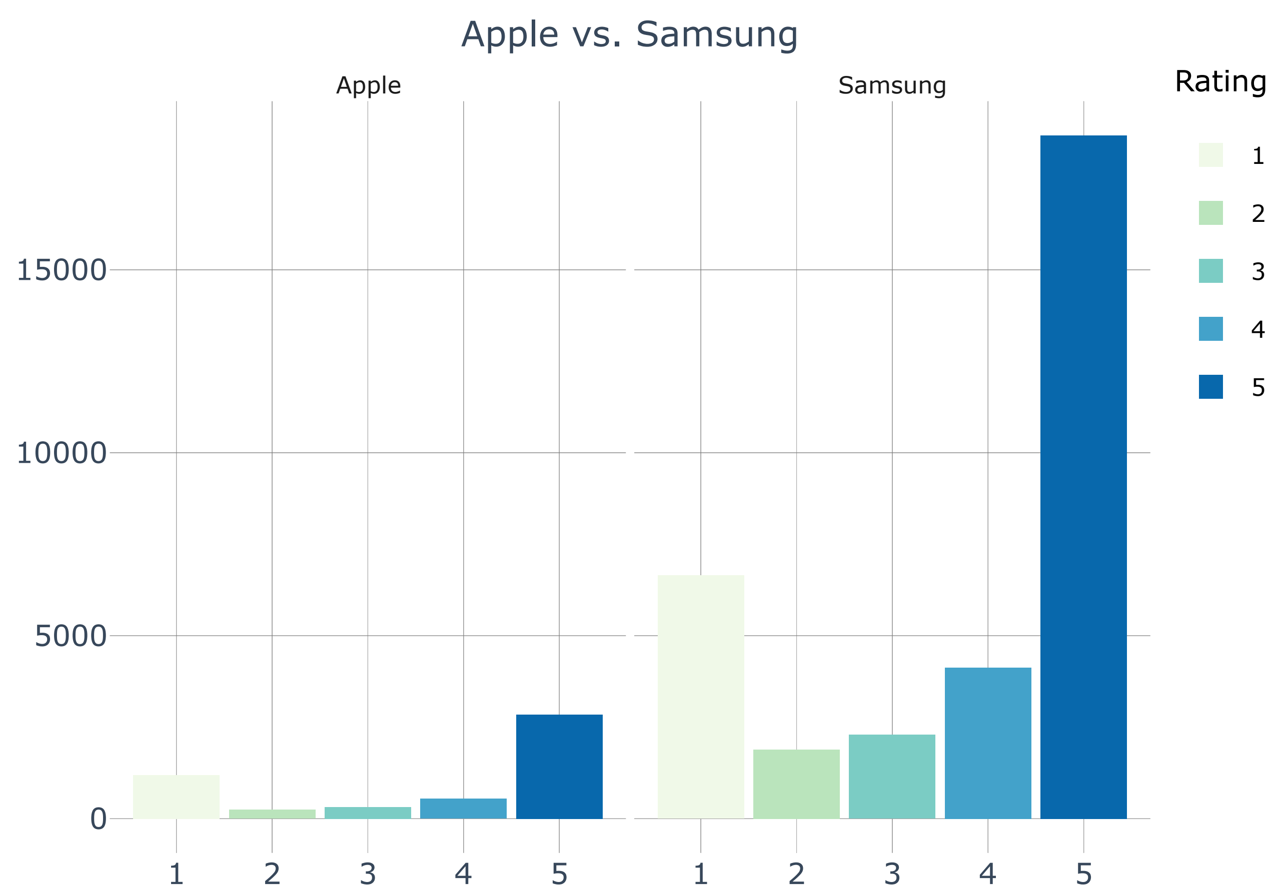
```
samsung_apple %>% select(brand) %>% distinct()
```

```
##     brand
## 1 Samsung
## 2   Apple
```

```
samsung_apple<- samsung_apple %>%
  mutate(price_group = if_else(between(price, 0, 250), "Low price",
  if_else(between(price, 250, 450), "Medium price",
  if_else(price > 450, "High price","Unknown price"))),
  price_group = if_else(is.na(price_group), "Unknown price", price_group)) %>%
  rownames_to_column(var = "id")
```

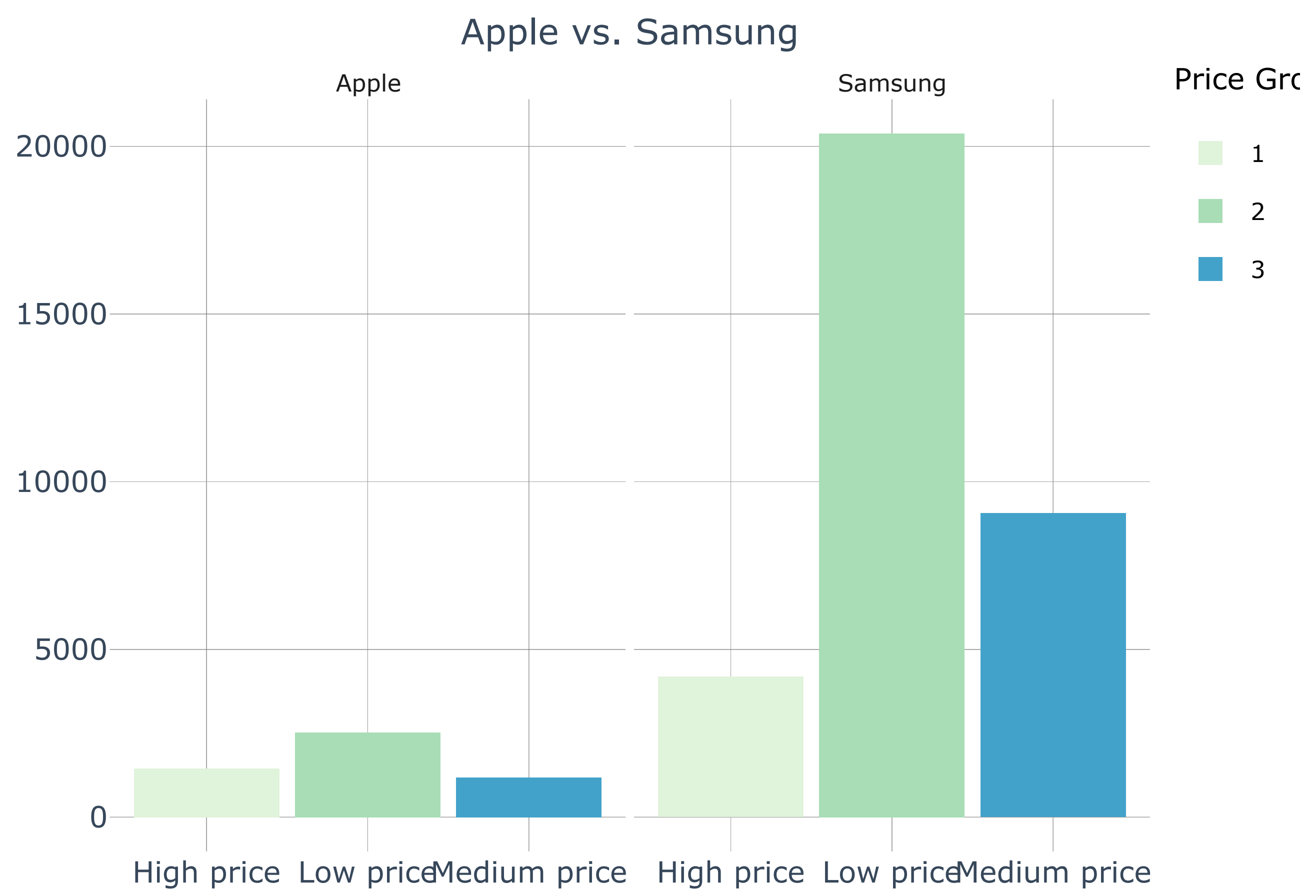
Product Rating

```
sa <- ggplot(samsung_apple,aes(x=reviewer_rating,group=brand))+
  geom_bar(aes(fill=factor(..x..)),stat="count")+
  facet_grid(~brand)+
  labs(x=NULL,y=NULL,title="Apple vs. Samsung ")+
  scale_fill_brewer(palette="GnBu",name="Rating") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5 , color = "#37475A"),
        axis.title.x = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
        axis.title.y = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
        axis.text = element_text(size = 11 , color = "#37475A"))
fig <- ggplotly(sa)
fig
```



Distribution of Price Groups

```
s_a <- ggplot(samsung_apple,aes(x=price_group,group=brand))+
  geom_bar(aes(fill=factor(..x..)),stat="count")+
  facet_grid(~brand)+
  labs(title="Apple vs. Samsung",x=NULL,y=NULL)+
  scale_fill_brewer(name="Price Group",palette="GnBu") +
theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5 , color = "#37475A"),
        axis.title.x = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
        axis.title.y = element_text(color = "#37475A", size = 12,face = "bold",family="Arial",margin = margin(r = 50)),
        axis.text = element_text(size = 11 , color = "#37475A"))
fig <- ggplotly(s_a)
fig
```



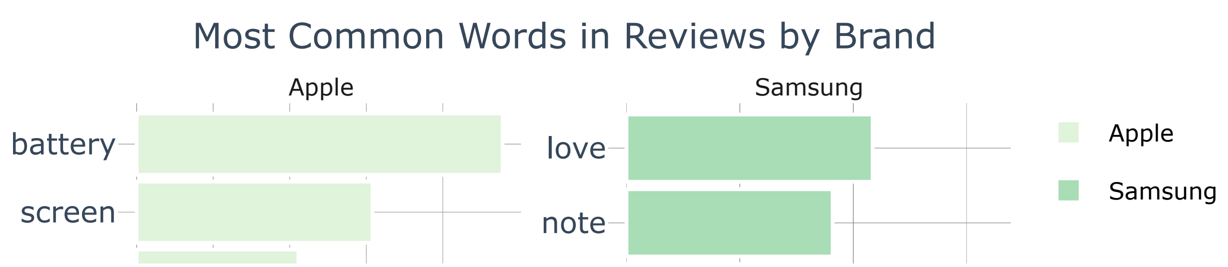
```
samsung_apple_clear<- samsung_apple %>%
  select(brand,body) %>%
  unnest_tokens(input=body,output=word) %>%
  count(brand,word,sort=T) %>%
  filter(nchar(word)>3) %>%
  filter(!word %in% stopwords_phone) %>%
  group_by(brand)
```

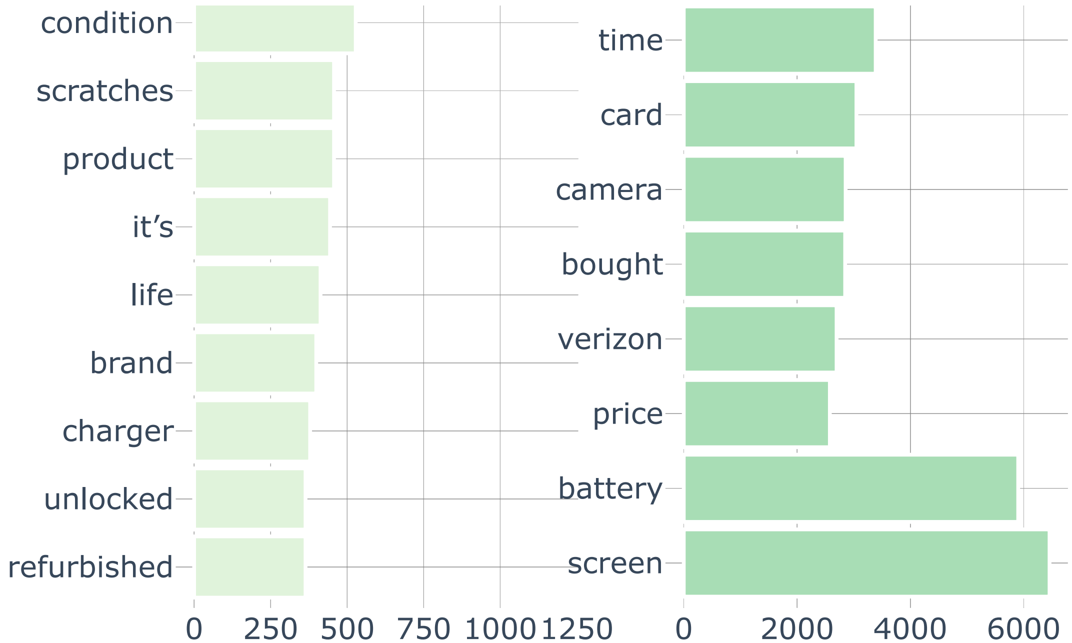
```
samsung_apple_clear2 <- samsung_apple_clear %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

Most Common Words in Reviews by Brand

```
sam_app <- samsung_apple_clear2 %>%
  top_n(n=10,n)%>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(x=word, y=n,fill=brand)) + geom_col(show.legend=F,col="white")+ facet_wrap(~brand, ncol=2,scales="free")+
  xlab(NULL) + ylab(NULL)+ ggtitle("Most Common Words in Reviews by Brand") + coord_flip() +
  scale_fill_brewer(palette="GnBu") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5 , color = "#37475A"),
        axis.title.x = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
        axis.title.y = element_text(color = "#37475A", size = 12,face = "bold",family="Arial"),
        axis.text = element_text(size = 11 , color = "#37475A"))
fig <- ggplotly(sam_app)
fig
```




condition
scratches
product
it’s
life
brand
charger
unlocked
refurbished
0
250
500
750
1000
1250
time
card
camera
bought
verizon
price
battery
screen
0
2000
4000
6000