Fitness Trackers

Omar Alamoudi

2022-09-25

Brief overview of the project:

In this project, we analyzed the fitness tracker product dataset and asked and answered questions in a structured and detailed manner. The dataset contains different products of different brands with their specifications, ratings and reviews for the Indian market.

The first step is to define the question:

We asked the following questions

1.Determine which is more popular (desired) smart watch or fitness bracelet?

2.What is the customer’s most favorite brand for both fitness watch and bracelet producers?

3.Determine the relationship between product specifications and customer evaluation.

The second step is to collect data:

After we set our goal of the analysis, which is to answer the questions asked.

We then fetch the data from Kaggle

https://www.kaggle.com/datasets/devsubhash/fitness-trackers-products-ecommerce?select=Fitness_trackers_updated.csv (https://www.kaggle.com/datasets/devsubhash/fitness-trackers-products-ecommerce?select=Fitness_trackers_updated.csv)

The data was collected from the e-commerce website (Flipkart) using Webscraping technology.

After downloading the data from Kaggle, we uploaded it to R-studio for analysis.

```
library(readr)

Fitness_trackers<- read_csv("D:/Projects/R-Fitness Trackers/Fitness_trackers_updated.csv")
```

```
## Rows: 610 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): Brand Name, Device Type, Model Name, Color, Display, Strap Material
## dbl (2): Rating (Out of 5), Average Battery Life (in days)
##
## 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.
```

We did data exploration to understand the structure and type of data and descriptive statistics of the data.

```
View(Fitness_trackers)

head(Fitness_trackers)  #Display the first 6 rows of the data set
```

```
## # A tibble: 6 x 11
##   `Brand Name` `Device Type` `Model Name` Color `Selling Price` `Original Price`
##   <chr>        <chr>         <chr>        <chr>           <dbl>            <dbl>
## 1 Xiaomi       FitnessBand   Smart Band 5 Black            2499             2999
## 2 Xiaomi       FitnessBand   Smart Band 4 Black            2099             2499
## 3 Xiaomi       FitnessBand   HMSH01GE     Black            1722             2099
## 4 Xiaomi       FitnessBand   Smart Band 5 Black            2469             2999
## 5 Xiaomi       FitnessBand   Band 3       Black            1799             2199
## 6 Xiaomi       FitnessBand   Band - HRX ~ Black            1299             1799
## # ... with 5 more variables: Display <chr>, `Rating (Out of 5)` <dbl>,
## #   `Strap Material` <chr>, `Average Battery Life (in days)` <dbl>,
## #   Reviews <dbl>
```

```
str(Fitness_trackers)  #Structure and type of data
```

```
## spec_tbl_df [610 x 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Brand Name                    : chr [1:610] "Xiaomi" "Xiaomi" "Xiaomi" "Xiaomi" ...
##  $ Device Type                   : chr [1:610] "FitnessBand" "FitnessBand" "FitnessBand"
"FitnessBand" ...
##  $ Model Name                    : chr [1:610] "Smart Band 5" "Smart Band 4" "HMSH01GE" "S
mart Band 5" ...
##  $ Color                         : chr [1:610] "Black" "Black" "Black" "Black" ...
##  $ Selling Price                 : num [1:610] 2499 2099 1722 2469 1799 ...
##  $ Original Price                : num [1:610] 2999 2499 2099 2999 2199 ...
##  $ Display                       : chr [1:610] "AMOLED Display" "AMOLED Display" "LCD Disp
lay" "AMOLED Display" ...
##  $ Rating (Out of 5)             : num [1:610] 4.1 4.2 3.5 4.1 4.3 4.2 4.3 4.4 4.4 4.2 ...
##  $ Strap Material                : chr [1:610] "Thermoplastic polyurethane" "Thermoplastic
polyurethane" "Leather" "Thermoplastic polyurethane" ...
##  $ Average Battery Life (in days): num [1:610] 14 14 14 14 7 20 7 14 14 7 ...
##  $ Reviews                       : num [1:610] NA NA NA NA NA NA NA 2 3 NA ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `Brand Name` = col_character(),
##   ..   `Device Type` = col_character(),
##   ..   `Model Name` = col_character(),
##   ..   Color = col_character(),
##   ..   `Selling Price` = col_number(),
##   ..   `Original Price` = col_number(),
##   ..   Display = col_character(),
##   ..   `Rating (Out of 5)` = col_double(),
##   ..   `Strap Material` = col_character(),
##   ..   `Average Battery Life (in days)` = col_double(),
##   ..   Reviews = col_number()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
sapply(Fitness_trackers, class)  # Print classes of  columns
```

```
##                     Brand Name                    Device Type
##                    "character"                    "character"
##                     Model Name                          Color
##                    "character"                    "character"
##                  Selling Price                 Original Price
##                      "numeric"                      "numeric"
##                        Display              Rating (Out of 5)
##                    "character"                      "numeric"
##                 Strap Material Average Battery Life (in days)
##                    "character"                      "numeric"
##                        Reviews
##                      "numeric"
```

```
summary(Fitness_trackers)  #Calculate Descriptive Statistics
```

```
##   Brand Name        Device Type         Model Name           Color
##  Length:610         Length:610         Length:610         Length:610
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Selling Price    Original Price     Display          Rating (Out of 5)
##  Min.   :   799   Min.   :  1599   Length:610         Min.   :2.000
##  1st Qu.:  6995   1st Qu.: 10249   Class :character   1st Qu.:4.000
##  Median : 14999   Median : 18995   Mode  :character   Median :4.200
##  Mean   : 20707   Mean   : 23978                      Mean   :4.196
##  3rd Qu.: 27468   3rd Qu.: 31417                      3rd Qu.:4.500
##  Max.   :122090   Max.   :122090                      Max.   :5.000
##                                                       NA's   :56
##  Strap Material     Average Battery Life (in days)    Reviews
##  Length:610         Min.   : 1.000                 Min.   :    2.0
##  Class :character   1st Qu.: 2.000                 1st Qu.:   79.5
##  Mode  :character   Median : 7.000                 Median :  287.5
##                     Mean   : 8.926                 Mean   : 1943.1
##                     3rd Qu.:14.000                 3rd Qu.:  904.5
##                     Max.   :45.000                 Max.   :23426.0
##                                                    NA's   :496
```

We also explored the names and number of columns and rows of data.

```
names(Fitness_trackers)   # Get column names
```

```
##  [1] "Brand Name"                     "Device Type"
##  [3] "Model Name"                     "Color"
##  [5] "Selling Price"                  "Original Price"
##  [7] "Display"                        "Rating (Out of 5)"
##  [9] "Strap Material"                 "Average Battery Life (in days)"
## [11] "Reviews"
```

```
dim(Fitness_trackers)            # Number of rows & columns
```

```
## [1] 610  11
```

The third step is to clean the data:

We first check the column names and modify them

```
colnames(Fitness_trackers)     # Print column names
```

```
##  [1] "Brand Name"                         "Device Type"                       
##  [3] "Model Name"                         "Color"                             
##  [5] "Selling Price"                      "Original Price"                    
##  [7] "Display"                            "Rating (Out of 5)"                 
##  [9] "Strap Material"                     "Average Battery Life (in days)"    
## [11] "Reviews"
```

```
#Modify Column Names

colnames(Fitness_trackers) <- c("Brand_Name", "Device_Type", "Model_Name","Color",
                                        "Selling_Price","Original_Price","Display","Rating",
                                "Strap_Material","Battery_Life_by_days","Reviews")
```

We checked for duplicates

```
Fitness_trackers <- unique(Fitness_trackers)          # Exclude duplicates
```

the number of smart watches VS number of Fitness Band

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v dplyr   1.0.8
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
Fitness_trackers %>%
  count(Device_Type) %>%
  group_by(Device_Type)
```

```
## # A tibble: 2 x 2
## # Groups:   Device_Type [2]
##   Device_Type     n
##   <chr>       <int>
## 1 FitnessBand    77
## 2 Smartwatch    529
```

We found the number of Smart Watch more than the number of Fitness Band, this leads to bias in the sample

To avoid bias, we equate the number of Smart Watch with the number of Fitness Band. The watch data contains 529, while the data contains 77 Fitness Bands.

Therefore, we delete part of the rows of smart watches so that their number equals the number of fitness band rows

first We convert data into numbers to facilitate the process of sorting and arranging

```
Fitness_trackers$Device_Type <- gsub("FitnessBand", "1", Fitness_trackers$Device_Type)
Fitness_trackers$Device_Type <- gsub("Smartwatch", "0", Fitness_trackers$Device_Type)
```

We arranged the data in ascending order to start from 0 where the value 0 means smart watches

```
dataframe<-Fitness_trackers %>% arrange(Device_Type)  # sort data
```

We then created a new, bias-free data set by deleting 452 smartwatch rows

```
Fitness_trackers2<-dataframe[-c(1:452), ]
```

We restore the data to its original state

```
Fitness_trackers2$Device_Type <- gsub("1", "FitnessBand", Fitness_trackers2$Device_Type)
Fitness_trackers2$Device_Type <- gsub("0", "Smartwatch", Fitness_trackers2$Device_Type)
```

Now we have the number of Smart Watch equal to the number of Fitness Band

dealing with messing data

See all missing values in the entire data set columns

```
Fitness_trackers2 %>% map(~sum(is.na(.)))
```

```
## $Brand_Name
## [1] 0
##
## $Device_Type
## [1] 0
##
## $Model_Name
## [1] 0
##
## $Color
## [1] 0
##
## $Selling_Price
## [1] 0
##
## $Original_Price
## [1] 0
##
## $Display
## [1] 0
##
## $Rating
## [1] 12
##
## $Strap_Material
## [1] 0
##
## $Battery_Life_by_days
## [1] 0
##
## $Reviews
## [1] 105
```

Number of NA values in the Rating column

```
Fitness_trackers2 %>%
  summarise(count=sum(is.na(Rating)))
```

```
## # A tibble: 1 x 1
##   count
##   <int>
## 1    12
```

Number of NA values in the Reviews column

```
sum(is.na(Fitness_trackers2$Reviews))
```

```
## [1] 105
```

We create a new data set and replace the missing values with zero

```
Fitness_trackers3<-Fitness_trackers2 %>%
  replace_na(list(
    Rating=0,
    Reviews=0))
```

Check missing values

```
colSums(is.na(Fitness_trackers3))
```

```
##          Brand_Name         Device_Type          Model_Name 
##                   0                   0                   0 
##               Color       Selling_Price      Original_Price 
##                   0                   0                   0 
##             Display              Rating      Strap_Material 
##                   0                   0                   0 
## Battery_Life_by_days             Reviews 
##                   0                   0
```

We delete the columns we don’t need in a new data set

```
Fitness_trackers4<- subset(Fitness_trackers3, select = -c(Model_Name,Original_Price,Display,Strap_Material) )

colnames(Fitness_trackers4)
```

```
## [1] "Brand_Name"           "Device_Type"          "Color"               
## [4] "Selling_Price"        "Rating"               "Battery_Life_by_days"
## [7] "Reviews"
```

Step four Analyzing the data:

As mentioned before, the goal of the project is to answer the questions:

1.Determine which is more popular (desired) smart watch or fitness bracelet?

2.What is the customer’s most favorite brand for both fitness watch and bracelet producers?

3.Determine the relationship between product specifications and customer evaluation.

Let’s start with the first question

Determine which is more popular smart watch or fitness Band?

To answer the question, we will use descriptive analysis to analyze users’ desire, which devices do they prefer smart watches or Fitness Bands.

We use the ratings and reviews numbers to know which smartwatch or Fitness Bands is most desirable.

First, let’s see the number of ratings for watches and Fitness Bands The sum of ratings for smart watches and Fitness Band

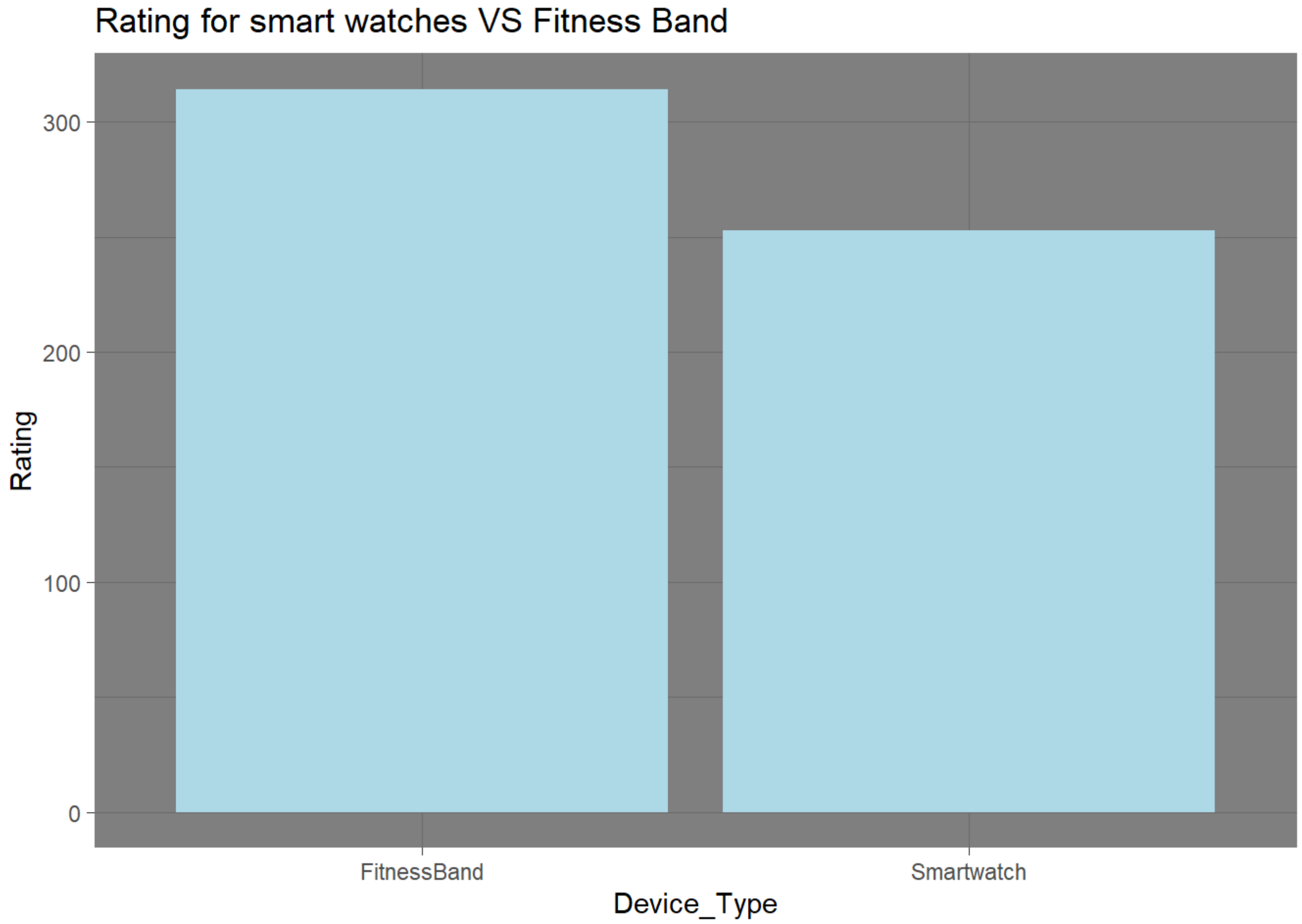
```
Fitness_trackers4 %>%
  group_by(Device_Type) %>%
  drop_na() %>%
  summarise(Rating=sum(Rating))
```

```
## # A tibble: 2 x 2
##   Device_Type Rating
##   <chr>        <dbl>
## 1 FitnessBand   314.
## 2 Smartwatch    253
```

Fitness Bands ratings are higher than smartwatch ratings, which means that customers prefer Fitness Bands more than smartwatches.

And here we see through the graph showing the ratings

```
rating<-ggplot(data=Fitness_trackers4, aes(x=Device_Type, y=Rating))+
  geom_bar(stat="identity",fill="lightblue")+
  theme_dark()+
  labs(title = "Rating for smart watches VS Fitness Band")

rating
```



Second, we see the number of reviews of smart watches and fitness bands

The sum of reviews for smart watches and Fitness Band

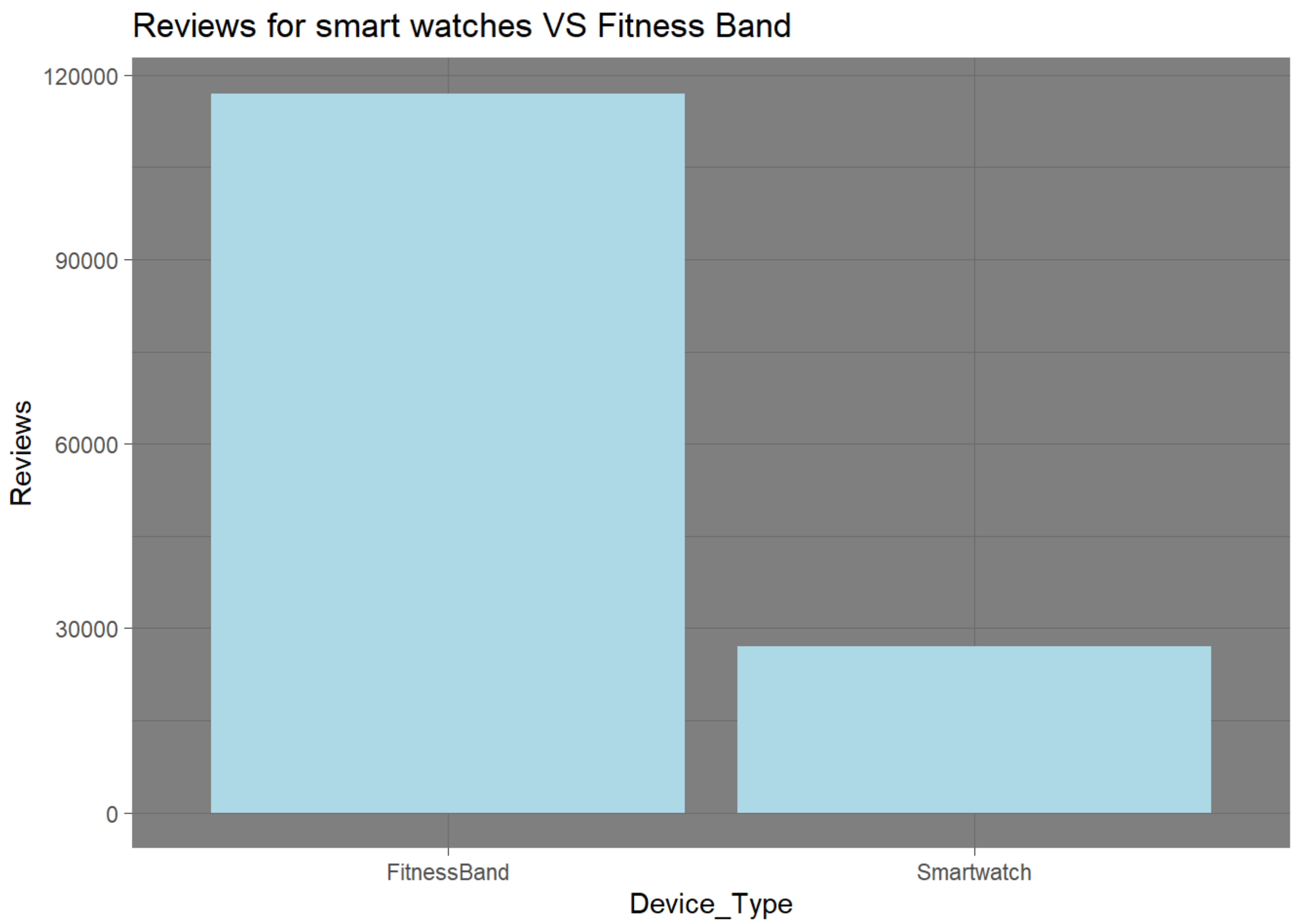
```
Fitness_trackers4 %>%
  group_by(Device_Type) %>%
  drop_na() %>%
  summarise(Reviews=sum(Reviews))
```

```
## # A tibble: 2 x 2
##   Device_Type Reviews
##   <chr>         <dbl>
## 1 FitnessBand  117009
## 2 Smartwatch    27062
```

Here, we note through the graph that the fitness band reviews are higher than the smart watch reviews, which indicates that there are more comments on the fitness band than on the smart watch.

Chart total Reviews for smart watches and Fitness Band

```
Review<-ggplot(data=Fitness_trackers4, aes(x=Device_Type, y=Reviews))+
  geom_bar(stat="identity",fill="lightblue")+
  theme_dark()+
  labs(title = "Reviews for smart watches VS Fitness Band")

Review
```



The data does not contain what the reviews include, are they positive or negative

then we move to the second question

What is the most preferred brand for customers for both smart watches and fitness bands?

We’ll start with descriptive data analytics

Brand names in the data set

```
unique(Fitness_trackers4$Brand_Name)
```

```
##  [1] "GARMIN"         "OnePlus"        "Huawei"         "FOSSIL"
##  [5] "boAt"           "Crossbeats"     "dizo by realme" "Noise"
##  [9] "Ptron"          "Zebronics"      "Fire-Boltt"     "Xiaomi"
## [13] "FitBit"         "realme"         "Honor"          "GOQii"
## [17] "Infinix"        "LCARE"          "LAVA"           "Fastrack"
## [21] "SAMSUNG"
```

Calculate the number of devices for each Brand

```
Fitness_trackers4 %>%
  count(Brand_Name) %>%
  print(n=21)
```

```
## # A tibble: 21 x 2
##    Brand_Name         n
##    <chr>          <int>
##  1 boAt               1
##  2 Crossbeats         8
##  3 dizo by realme     2
##  4 Fastrack           2
##  5 Fire-Boltt        14
##  6 FitBit            27
##  7 FOSSIL             1
##  8 GARMIN            12
##  9 GOQii              4
## 10 Honor             13
## 11 Huawei            26
## 12 Infinix            1
## 13 LAVA               1
## 14 LCARE              1
## 15 Noise              5
## 16 OnePlus            3
## 17 Ptron              5
## 18 realme             4
## 19 SAMSUNG            4
## 20 Xiaomi             8
## 21 Zebronics         12
```

Create a dataset to rank the devices owned by each brand

```
newdf<-Fitness_trackers4 %>%count(Brand_Name)
```

Change the device number column name

```
names(newdf)[names(newdf) == 'n'] <- 'number_of_devices'
```

Arrange the number of devices for each brand

```
newdf %>%
  arrange(-number_of_devices) %>%
  print(n=21)
```

```
## # A tibble: 21 x 2
##    Brand_Name     number_of_devices
##    <chr>                      <int>
##  1 FitBit                        27
##  2 Huawei                        26
##  3 Fire-Boltt                    14
##  4 Honor                         13
##  5 GARMIN                        12
##  6 Zebronics                     12
##  7 Crossbeats                     8
##  8 Xiaomi                         8
##  9 Noise                          5
## 10 Ptron                          5
## 11 GOQii                          4
## 12 realme                         4
## 13 SAMSUNG                        4
## 14 OnePlus                        3
## 15 dizo by realme                 2
## 16 Fastrack                       2
## 17 boAt                           1
## 18 FOSSIL                         1
## 19 Infinix                        1
## 20 LAVA                           1
## 21 LCARE                          1
```

We calculate the total ratings obtained by each company

We created a data set to collect and rank the most rated Brands

```
newdf2<-aggregate(Fitness_trackers4$Rating,list(Fitness_trackers4$Brand_Name),sum,na.rm=T)

colnames(newdf2)<-c("Brand_Name","number_of_Rating")  #Change column names
```

Create a new data set to rank the ratings

```
newdf3<-newdf2 %>% arrange(-number_of_Rating)
```

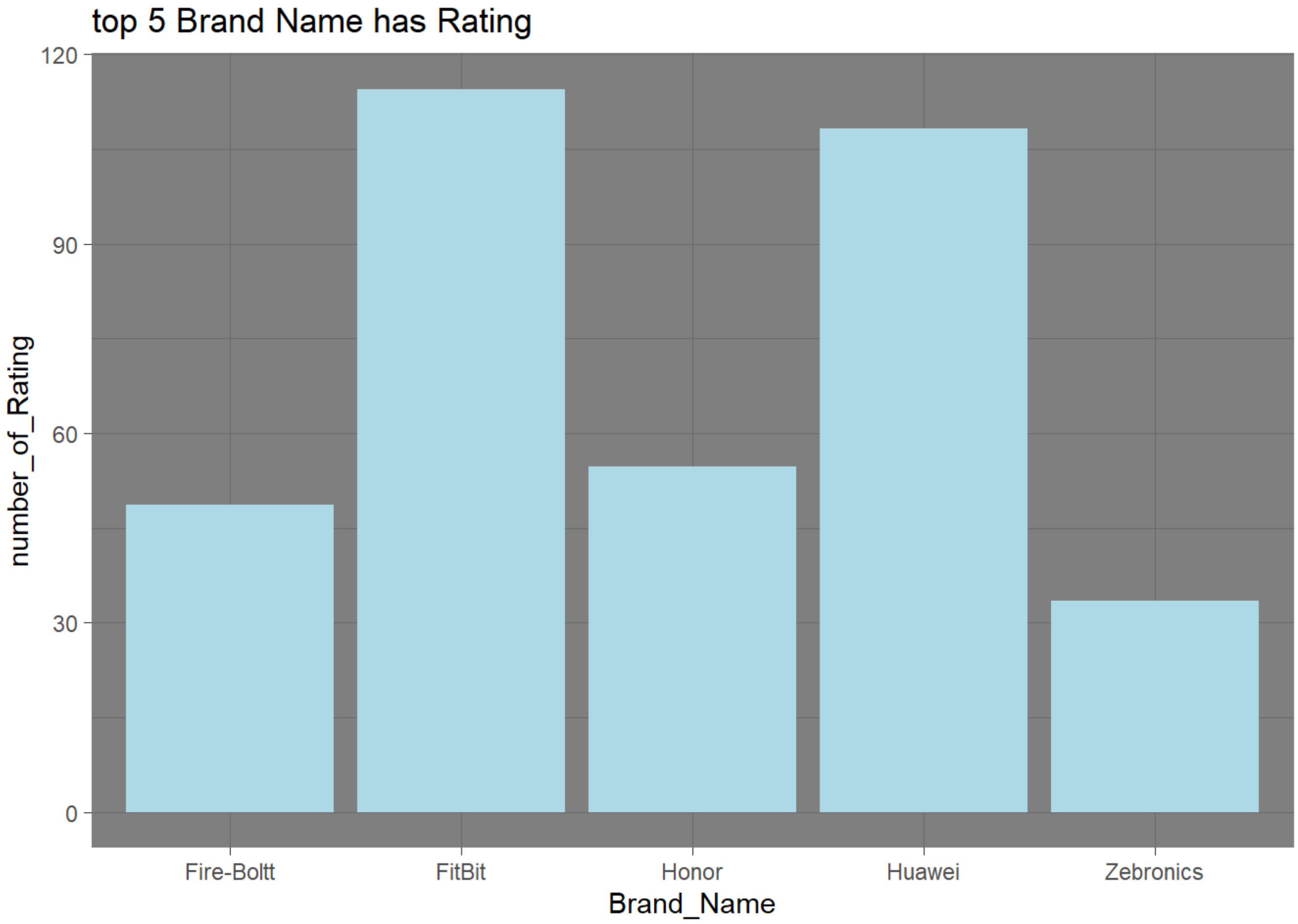
We are now choosing the 5 most rated Brands

```
top_Brand<-newdf3[-c(6:21), ] # To delete data

top_Brand
```

```
##   Brand_Name number_of_Rating
## 1     FitBit            114.5
## 2     Huawei            108.2
## 3      Honor             54.7
## 4 Fire-Boltt             48.7
## 5  Zebronics             33.5
```

Representation in a graph of the most rated brands of smart devices

```
Brand<-ggplot(data=top_Brand, aes(x=Brand_Name, y=number_of_Rating))+
  geom_bar(stat="identity",fill="lightblue")+
  theme_dark()+
  labs(title = "top 5 Brand Name has Rating")

Brand
```



And through the chart, we found that the brand fitBit got the highest rating of 114.5

then we move to the third question

Determine the relationship between product specifications, prices and customer evaluation.

We analyze the correlation between specifications by comparing the classification and some specifications by creating a drawing

We compare the color - the life of the battery - and the selling price compared to the classifications

First we see which colors got the highest rating

We created a new dataset to put the colors that got the top 10 rating in order

```
top_color<-aggregate(Fitness_trackers4$Rating,list(Fitness_trackers4$Color),sum,na.rm=T)
```

Edit column names

```
colnames(top_color)<-c("color","number_of_Rating")
```

Create a new dataset for color rating in order

```
top_color2<-top_color %>%
  arrange(-number_of_Rating)
```

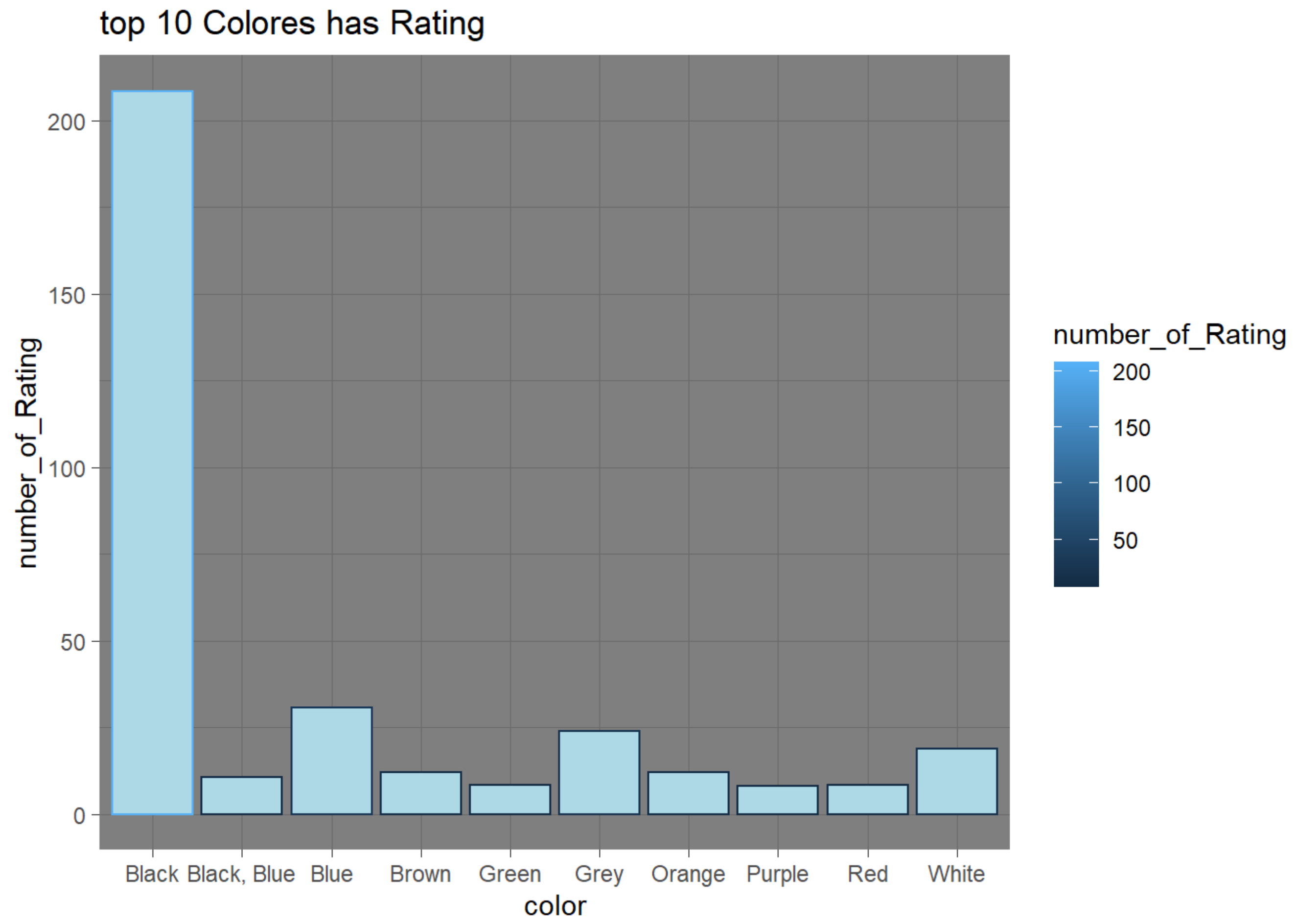
the colors that got the top 10 rating

```
top_color3<-top_color2[-c(11:66), ]

top_color3
```

```
##          color number_of_Rating
## 1        Black             208.5
## 2         Blue              30.6
## 3         Grey              23.8
## 4        White              18.8
## 5        Brown              12.1
## 6       Orange              11.9
## 7  Black, Blue              10.6
## 8        Green               8.3
## 9          Red               8.3
## 10      Purple               8.2
```

Representation in a graph to see which colors are preferred by customers

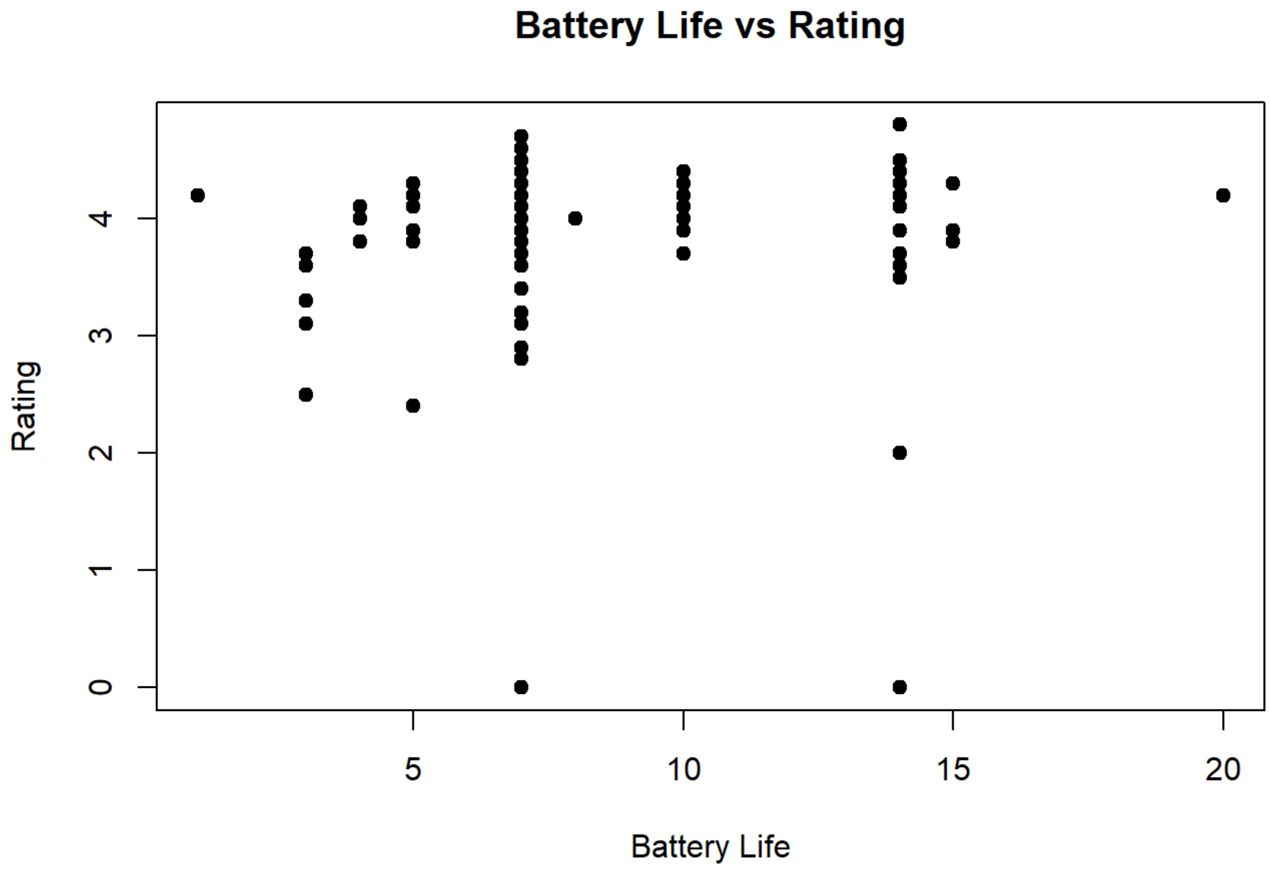
```
colors<-ggplot(data=top_color3, aes(x=color, y=number_of_Rating,color=number_of_Rating))+
  geom_bar(stat="identity",fill="lightblue")+
  theme_dark()+
  labs(title = "top 10 Colores has Rating")

colors
```



We see that the black color got the highest rating. This indicates the black color is the most desirable color in Fitness tracking devices

Then we compare the battery life and ratings and see the relationship between them

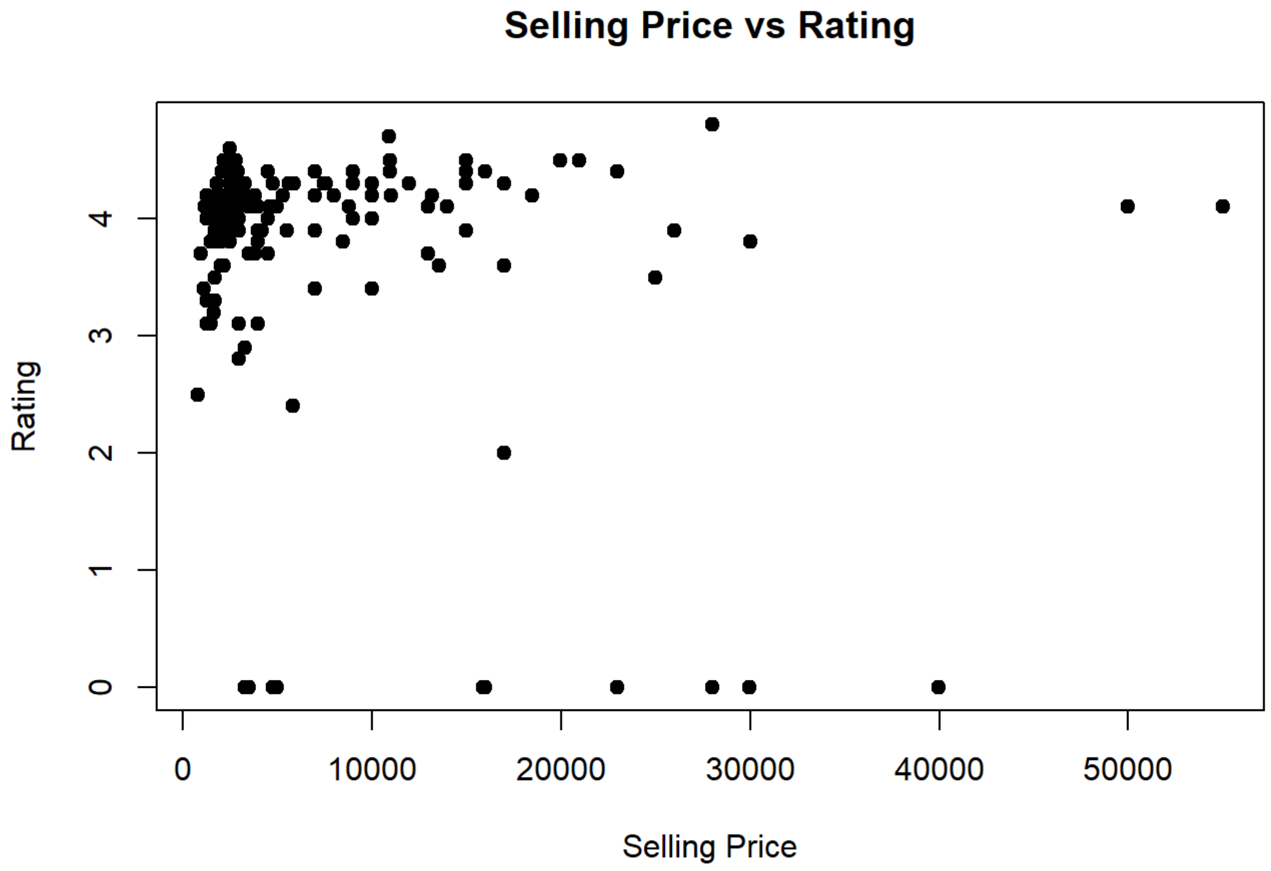
```
plot(x = Fitness_trackers4$Battery_Life_by_days,y = Fitness_trackers4$Rating,
     xlab = "Battery Life",
     ylab = "Rating",
     main = "Battery Life vs Rating"
     , pch=19
)
```



Through the graph, the relationship between the device’s battery life and the evaluation shows us that the relationship between them or the correlation is weak or zero, meaning that there is no correlation or relationship between the battery life and customer evaluation.

Determine the relationship between selling price and customer rating

```
plot(x = Fitness_trackers4$Selling_Price,y = Fitness_trackers4$Rating,
     xlab = "Selling Price",
     ylab = "Rating",
     main = "Selling Price vs Rating"
     , pch=19
)
```



Through the graph, it shows us that the relationship or correlation between the selling price and customer evaluation is a negative relationship, meaning that the cheaper or less the price, the higher the customer evaluation.

Step five Interpretation of results:

We also know that the goal of the project is to answer the following questions….

1.Determine which is more popular smart watch or fitness Band?

2.What is the most preferred brand for customers for both smart watches and fitness bands?

3.Determine the relationship between product specifications, prices and customer rating

1.which is more popular smart watch or fitness Band?

Through the results of ratings and reviews, the FitnessBand is the most requested and desired by customers, perhaps because the FitnessBand is cheaper compared to smart watches, as well as sufficient specifications to track sports activities.

2.What is the most preferred brand for customers for both smart watches and fitness bands?

The brand fitBit is the most famous in the world of sports trackers for having the most rating of 114.5Perhaps the reason for this is that it has 27 devices, which is the most brand with devices in the data set.

3.Determine the relationship between product specifications, prices and customer rating

We analyzed the relationships or correlation between specifications and customer rating Where we found that the black color is the most common, and we found the correlation between battery life and customer rating no relationship. We also found an inverse relationship between the selling price and customer rating , where the lower the price, the higher the rating