GoogleDataAnalyticsCapstone_Bellabeat

LorenFranco

2023-10-06

#Data Analysis: Bellabeat's Marketing Strategy

##–ASK–

Problem: Analyze smart device usage data from the FitBit dataset in order to gain insight into how consumers use non-Bellabeat smart devices. Then select one Bellabeat product to apply these insights to in your presentation.

Business Task: Find out what is the current trend seen in non-Bellabeat (competitor) smart devices, and understand how Bellabeat can apply this trend to a current product marketing strategy.

Stakeholders: CoFounders - Urska Srsen & Sando Mur

Mission & Business Goal Considerations: Empower women with knowledge about their own health and habits.

##–PREPARE–

Install & load tidyverse:

```
##always do this at the beginning
install.packages('tidyverse')
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## 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
```

Install & load additional packages for analysis:

```
##CLEAN DATASET
install.packages("here")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
##>>makes referencing files easier
install.packages("skimr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
##>>makes summarizing data easy, skim through more quickly
install.packages("janitor")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
##>>functions for cleaning data
##>##load packages
library("here")
```

```
## here() starts at /cloud/project
```

```
library("skimr")
library("janitor")
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

##–PROCESS–

Install datasets (CSV files):

```
##understand the columns present in ActivityDaily, to see where we can compare them to
↪  the other spreadsheets
library(readr)

ActivityDaily <- read_csv("DATA/Fitabase Data 4.12.16-5.12.16/dailyActivity_merged.csv")
```

```
## Rows: 940 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): ActivityDate
## dbl (14): Id, TotalSteps, TotalDistance, TrackerDistance, LoggedActivitiesDi...
##
## 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.
```

```
  colnames(ActivityDaily)
```

```
##  [1] "Id"                       "ActivityDate"
##  [3] "TotalSteps"               "TotalDistance"
##  [5] "TrackerDistance"          "LoggedActivitiesDistance"
##  [7] "VeryActiveDistance"       "ModeratelyActiveDistance"
##  [9] "LightActiveDistance"      "SedentaryActiveDistance"
## [11] "VeryActiveMinutes"        "FairlyActiveMinutes"
## [13] "LightlyActiveMinutes"     "SedentaryMinutes"
## [15] "Calories"
```

```
  na.strings=c("")

StepsDaily <- read_csv("DATA/Fitabase Data 4.12.16-5.12.16/dailySteps_merged.csv")
```

```
## Rows: 940 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityDay
## dbl (2): Id, StepTotal
##
## 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.
```

```
  na.strings=c("")

WeightLogInfo <- read_csv("DATA/Fitabase Data 4.12.16-5.12.16/weightLogInfo_merged.csv")
```

```
## Rows: 67 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Date
## dbl (6): Id, WeightKg, WeightPounds, Fat, BMI, LogId
## lgl (1): IsManualReport
##
## 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.
```

```
  na.strings=c("")

CaloriesDaily <- read_csv("DATA/Fitabase Data 4.12.16-5.12.16/dailyCalories_merged.csv")
```

```
## Rows: 940 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityDay
## dbl (2): Id, Calories
##
## 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.
```

```
  na.strings=c("")

SleepActivityDaily <- read_csv("DATA/Fitabase Data 4.12.16-5.12.16/sleepDay_merged.csv")
```

```
## Rows: 413 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): SleepDay
## dbl (4): Id, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed
##
## 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.
```

```
  na.strings=c("")
```

See the summaries of the dataframes using glimpse():

```
##summary of dataframe on ActivityDaily
glimpse(ActivityDaily)
```

```
## Rows: 940
## Columns: 15
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate             <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/~
## $ TotalSteps               <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 13019~
## $ TotalDistance            <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~
## $ TrackerDistance          <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8~
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveDistance       <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.5~
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.3~
## $ LightActiveDistance      <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.0~
## $ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveMinutes        <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4~
## $ FairlyActiveMinutes      <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21~
## $ LightlyActiveMinutes     <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 205, ~
## $ SedentaryMinutes         <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775, 818~
## $ Calories                 <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203~
```

```
##summary of dataframe on CaloriesDaily
glimpse(CaloriesDaily)
```

```
## Rows: 940
## Columns: 3
## $ Id          <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 1503960366~
## $ ActivityDay <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/2016", "4/16/~
## $ Calories    <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 2035, 1786, 1775~
```

```
##summary of dataframe on StepsDaily
glimpse(StepsDaily)
```

```
## Rows: 940
## Columns: 3
## $ Id          <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 1503960366~
## $ ActivityDay <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/2016", "4/16/~
## $ StepTotal   <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 13019, 15506, 1054~
```

How many unique client (Ids) are logged in each dataframe:

```
##how many clients in each dataframe // how many distinct IDs in each section
  n_distinct(CaloriesDaily$Id)
```

```
## [1] 33
```

```
  n_distinct(SleepActivityDaily$Id)
```

```
## [1] 24
```

```
  n_distinct(WeightLogInfo$Id)
```

```
## [1] 8
```

```
  n_distinct(StepsDaily$Id)
```

```
## [1] 33
```

```
  n_distinct(ActivityDaily$Id)
```

```
## [1] 33
```

The most participants are in the StepsDaily, CaloriesDaily, and ActivityDaily dataframes. The least participants are in the WeightLogInfo dataframe.

How many observations are there in each dataframe:

```
##how many observations are there in each dataframe?
nrow(CaloriesDaily)
```

```
## [1] 940
```

```
nrow(SleepActivityDaily)
```

```
## [1] 413
```

```
nrow(WeightLogInfo)
```

```
## [1] 67
```

```
nrow(StepsDaily)
```

```
## [1] 940
```

```
nrow(ActivityDaily)
```

```
## [1] 940
```

For futher analysis the following dataframes will be focused on: CaloriesDaily, StepsDaily, ActivityDaily

Change the column name that represents "Date" in ActivityDaily, StepsDaily, CaloriesDaily

```
ActivityDaily <- ActivityDaily %>%
    rename(Date=ActivityDate)
StepsDaily <- StepsDaily %>%
    rename(Date=ActivityDay)
CaloriesDaily <- CaloriesDaily %>%
     rename(Date=ActivityDay)

##view new column name
colnames(ActivityDaily)
```

```
##  [1] "Id"                       "Date"
##  [3] "TotalSteps"               "TotalDistance"
##  [5] "TrackerDistance"          "LoggedActivitiesDistance"
##  [7] "VeryActiveDistance"       "ModeratelyActiveDistance"
##  [9] "LightActiveDistance"      "SedentaryActiveDistance"
## [11] "VeryActiveMinutes"        "FairlyActiveMinutes"
## [13] "LightlyActiveMinutes"     "SedentaryMinutes"
## [15] "Calories"
```

```
colnames(StepsDaily)
```

```
## [1] "Id"        "Date"      "StepTotal"
```

```
colnames(CaloriesDaily)
```

```
## [1] "Id"       "Date"     "Calories"
```

What activity categories have the highest time recorded?

```
  summary(ActivityDaily)
```

```
##        Id                 Date             TotalSteps    TotalDistance
##  Min.   :1.504e+09   Length:940         Min.   :    0   Min.   : 0.000
##  1st Qu.:2.320e+09   Class :character   1st Qu.: 3790   1st Qu.: 2.620
##  Median :4.445e+09   Mode  :character   Median : 7406   Median : 5.245
##  Mean   :4.855e+09                      Mean   : 7638   Mean   : 5.490
##  3rd Qu.:6.962e+09                      3rd Qu.:10727   3rd Qu.: 7.713
##  Max.   :8.878e+09                      Max.   :36019   Max.   :28.030
##  TrackerDistance  LoggedActivitiesDistance VeryActiveDistance
##  Min.   : 0.000   Min.   :0.0000           Min.   : 0.000
##  1st Qu.: 2.620   1st Qu.:0.0000           1st Qu.: 0.000
##  Median : 5.245   Median :0.0000           Median : 0.210
##  Mean   : 5.475   Mean   :0.1082           Mean   : 1.503
##  3rd Qu.: 7.710   3rd Qu.:0.0000           3rd Qu.: 2.053
##  Max.   :28.030   Max.   :4.9421           Max.   :21.920
##  ModeratelyActiveDistance LightActiveDistance SedentaryActiveDistance
##  Min.   :0.0000           Min.   : 0.000      Min.   :0.000000
##  1st Qu.:0.0000           1st Qu.: 1.945      1st Qu.:0.000000
##  Median :0.2400           Median : 3.365      Median :0.000000
##  Mean   :0.5675           Mean   : 3.341      Mean   :0.001606
##  3rd Qu.:0.8000           3rd Qu.: 4.782      3rd Qu.:0.000000
##  Max.   :6.4800           Max.   :10.710      Max.   :0.110000
##  VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes
##  Min.   :  0.00    Min.   :  0.00      Min.   :  0.0        Min.   :   0.0
##  1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:127.0        1st Qu.: 729.8
##  Median :  4.00    Median :  6.00      Median :199.0        Median :1057.5
##  Mean   : 21.16    Mean   : 13.56      Mean   :192.8        Mean   : 991.2
##  3rd Qu.: 32.00    3rd Qu.: 19.00      3rd Qu.:264.0        3rd Qu.:1229.5
##  Max.   :210.00    Max.   :143.00      Max.   :518.0        Max.   :1440.0
##     Calories
##  Min.   :   0
##  1st Qu.:1828
##  Median :2134
##  Mean   :2304
##  3rd Qu.:2793
##  Max.   :4900
```

Of the activity categories, most time is logged in the LightlyActiveMinutes & SedentaryMinutes categories.

After reviewing the data for activity, SedentaryMinutes has the most time logged. Also part of this dataset is the TotalSteps logged by each participant. What can the correlation between these two datapoints reveal?

```
# Create a group-means data set
mean_activity_id<- ActivityDaily%>%
  group_by(Id) %>%
  summarise(
    TotalSteps= mean(TotalSteps),
    SedentaryMinutes = mean(SedentaryMinutes)
  )
```

Set average BMI per each Id:

```
bmi_cat<- WeightLogInfo %>%
  group_by(Id) %>%
  summarise(BMI=mean(BMI))

bmi_mod <- bmi_cat %>%
  mutate(weight_class = case_when(
   BMI < 18.5 ~ 'underweight',
   between(BMI, 18.5, 24.9) ~ 'normal',
   between(BMI, 25, 29.9) ~ 'overweight',
   between(BMI, 30, 34.9) ~ 'obese',
   BMI > 35 ~ 'extreme',
   TRUE ~ 'unknown'))

bmi_mod <- bmi_mod %>%
  select(-BMI)
```

Change Dates to weekdays:

```
colnames(ActivityDaily)
```

```
##  [1] "Id"                       "Date"
##  [3] "TotalSteps"               "TotalDistance"
##  [5] "TrackerDistance"          "LoggedActivitiesDistance"
##  [7] "VeryActiveDistance"       "ModeratelyActiveDistance"
##  [9] "LightActiveDistance"      "SedentaryActiveDistance"
## [11] "VeryActiveMinutes"        "FairlyActiveMinutes"
## [13] "LightlyActiveMinutes"     "SedentaryMinutes"
## [15] "Calories"
```

```
##change date from MM/DD/YYYY TO YYY/DD/MM
weekday_ActivityDaily <- ActivityDaily %>%
  mutate(Date = as.Date(Date, format = "%m/%d/%Y"))
str(weekday_ActivityDaily)
```

```
## tibble [940 x 15] (S3: tbl_df/tbl/data.frame)
##  $ Id                       : num [1:940] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ Date                     : Date[1:940], format: "2016-04-12" "2016-04-13" ...
##  $ TotalSteps               : num [1:940] 13162 10735 10460 9762 12669 ...
##  $ TotalDistance            : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance          : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance       : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance      : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes        : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes      : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes     : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes         : num [1:940] 728 776 1218 726 773 ...
##  $ Calories                 : num [1:940] 1985 1797 1776 1745 1863 ...
```

```
##change date to weekdays
weekday_ActivityDaily <- weekday_ActivityDaily %>% mutate(weekday=weekdays(Date))
```

Join the Daily Activity to BMI by Id:

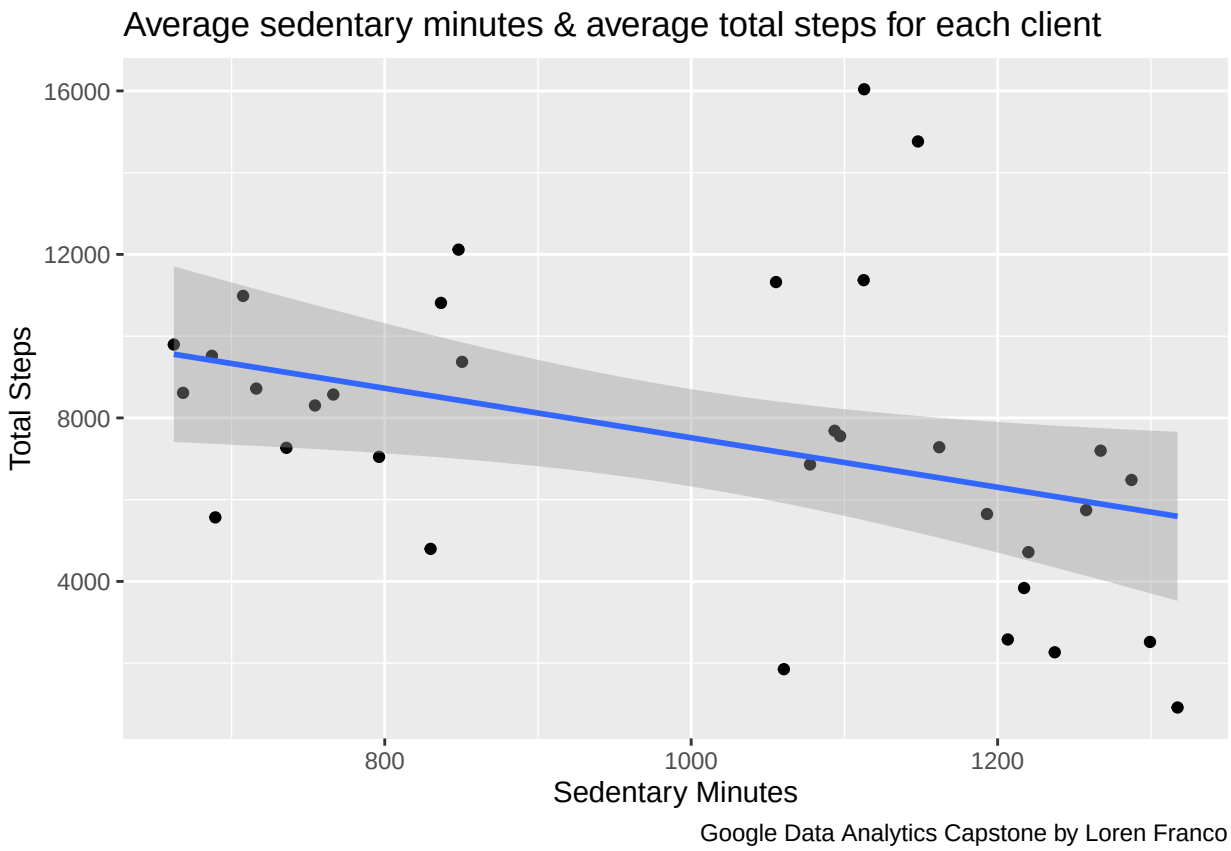
```
weeklyactivity_bmi <-
  left_join(ActivityDaily, bmi_mod, by=c('Id'='Id'))
```

##–ANALYZE–

What would a plot of SedentaryMinutes over TotalSteps reveal?

```
library(ggplot2)

ggplot(data=mean_activity_id,aes(x=SedentaryMinutes,y=TotalSteps)) +
  geom_point() +
 geom_smooth(formula = y ~ x, method = "lm") +
  labs(y= "Total Steps", x = "Sedentary Minutes",
       title="Average sedentary minutes & average total steps for each client",
  caption="Google Data Analytics Capstone by Loren Franco")
```




```
##on average, as Sedentary Minutes increase, the tota steps decrease for users
##this might provide a pocket for innovation, where we can notify the user as they are
↪  approaching a specified threshold to get their steps in
```

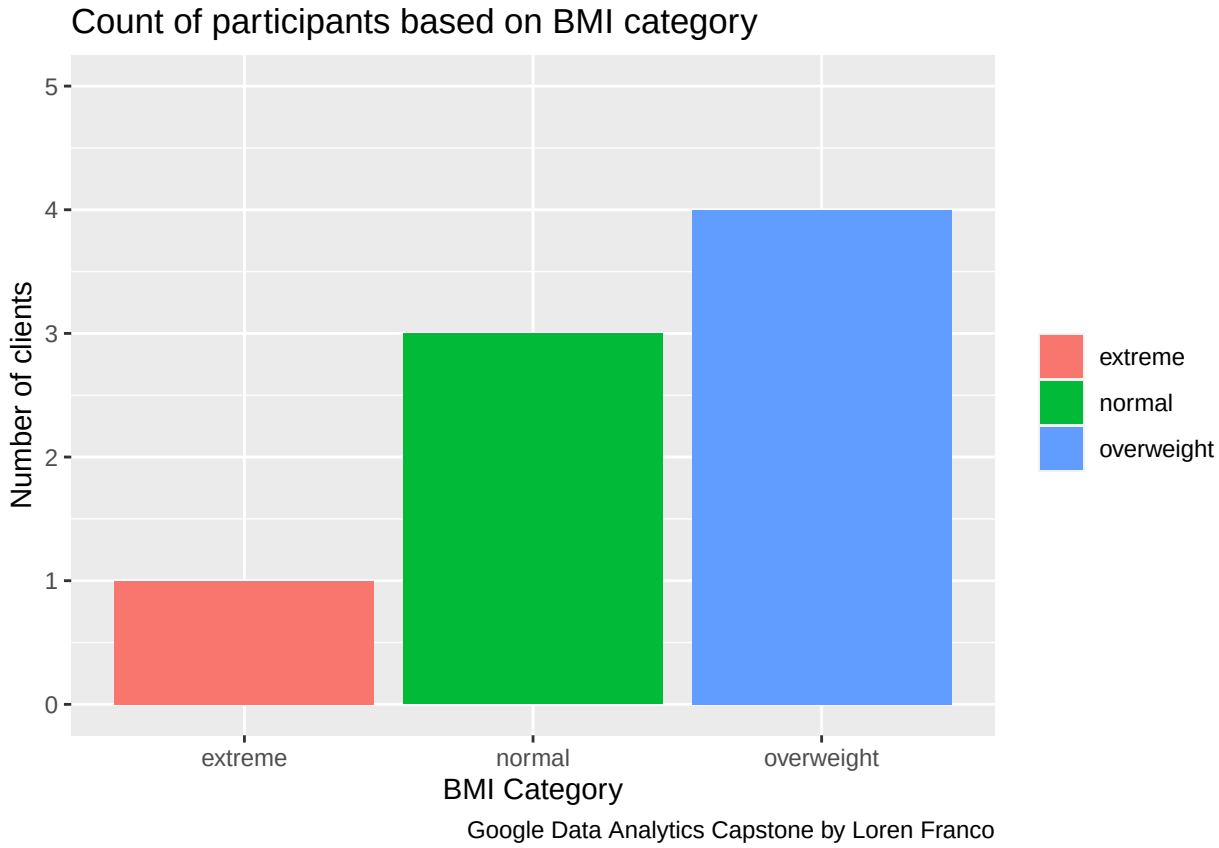
On average, as Sedentary Minutes increase, the Total Steps decrease.

This might provide a pocket for innovation, where Bellabeat can use the activity tracked in the Leaf to notify

the user via their phone app they should get up and walk around to increase their steps. This notification would then fire off a specified threshold (ex: Sedentary Mintutes or Total Steps threshold).

What is the frequency of BMI categories across all clients?

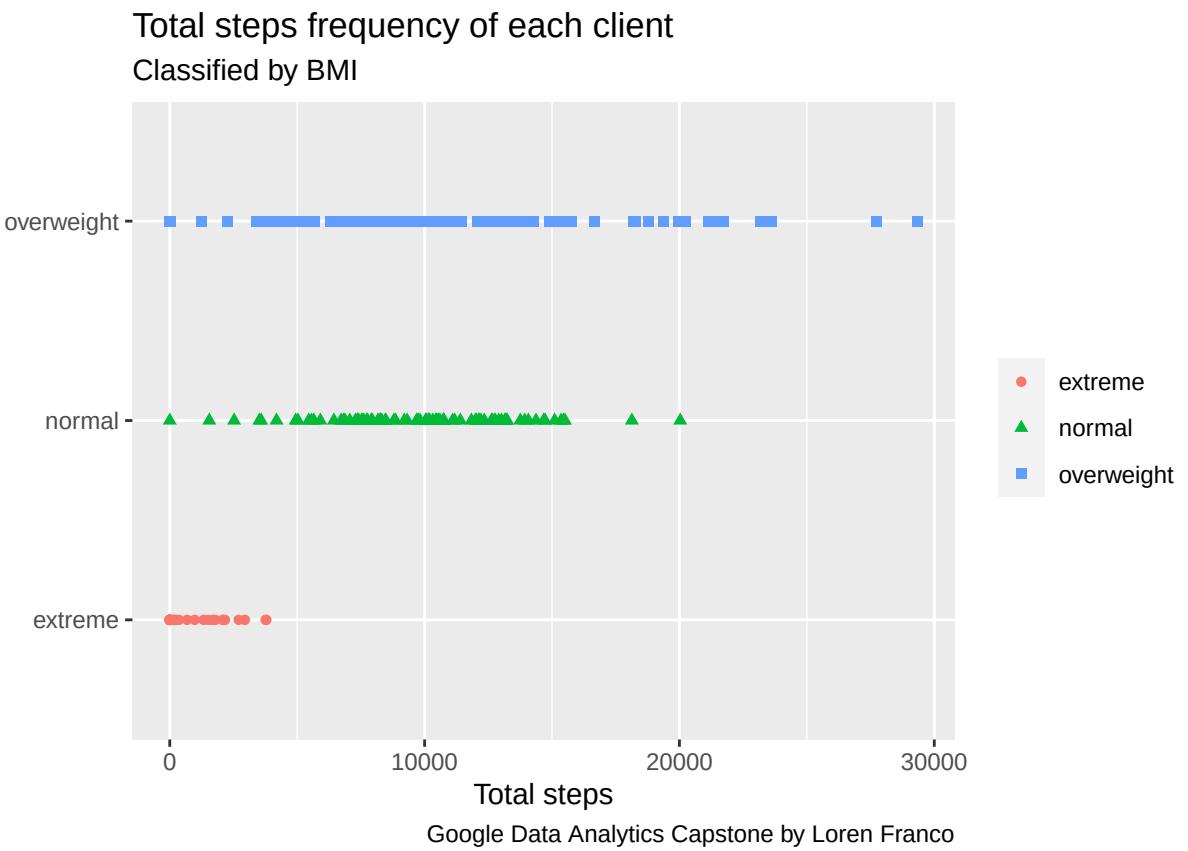
```
library(ggplot2)
ggplot(data=bmi_mod,aes(x=weight_class,fill=weight_class)) +
        geom_bar() +
      theme(legend.title=element_blank()) +
  scale_y_continuous(limits=c(0,5)) +
        labs(y= "Number of clients", x = "BMI Category",
             title="Count of participants based on BMI category",
            caption="Google Data Analytics Capstone by Loren Franco")
```



There are more participants within the normal and overweight categories, what are their TotalSteps?

```
library(ggplot2)
 weeklyactivity_bmi2 <- weeklyactivity_bmi %>% select(Id,TotalSteps,weight_class) %>%
↪  na.omit(weight_class)

ggplot(data=weeklyactivity_bmi2,aes(x=TotalSteps, y=weight_class, shape=weight_class,
↪  color=weight_class)) +
  geom_point() +
   theme(legend.title=element_blank()) +
  labs(y= "", x = "Total steps",
       title="Total steps frequency of each client", subtitle = "Classified by BMI",
      caption="Google Data Analytics Capstone by Loren Franco")
```



For those in the extreme BMI category, they are logging less than 5000 total steps each day. Adults are recomended to reach about 10,000 steps per day and for records of <50000 steps per day, the client is considered to be living a sedentary lifestyle.

It may be helpful for Bellabeat to focus on empowering their population in the extreme BMI range, and encourage them to log more steps. Perhaps these individuals can sign up for daily step challenges, be notified (during certain times of the day, maybe client provided?) that they should get some steps in.

What days of the week are clients more active according to their product tracking? The below bar chart will determine the frequency of the client's Activity data (includes total steps, sedentary/active minutes, calories, and total distance)

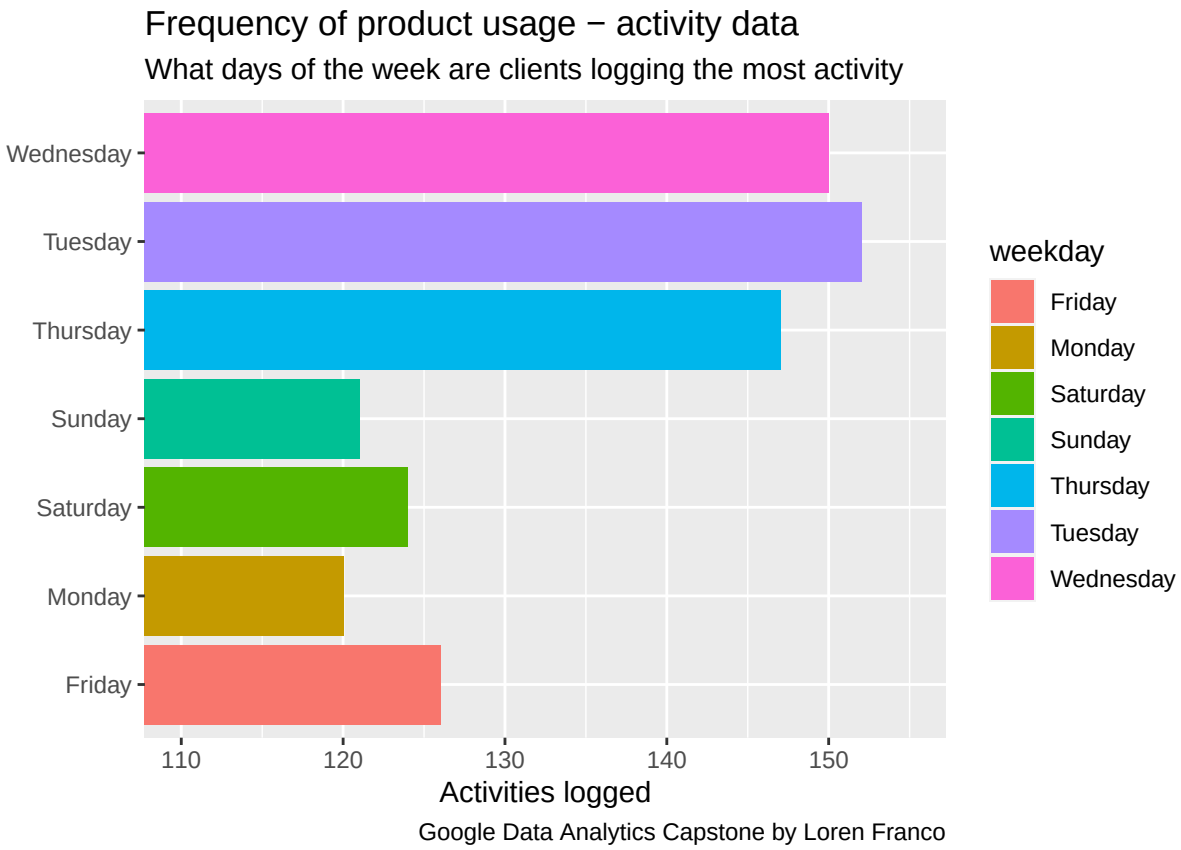
```
weekdays_count <- weekday_ActivityDaily %>%
  count(weekday, name="countof_weekday")

##heres the visual

library(scales)
```

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

```
ggplot(data=weekdays_count,aes(x=weekday,y=countof_weekday,fill=weekday)) +
  geom_col() +
    scale_y_continuous(limits=c(110,155),oob = rescale_none) +
  coord_flip() +
  labs(y= "Activities logged", x = "",
       title="Frequency of product usage - activity data",
       subtitle = "What days of the week are clients logging the most activity",
      caption="Google Data Analytics Capstone by Loren Franco")
```



The weekdays with the highest product usage are Tuesdays & Wednesdays, while the weekdays with the lowest product usage are Sunday's and Monday's.

##–CONCLUSION–

To continue the Bellabeat's mission and goals, which is about empowering women of their own health and habits, the aforementioned data can create an understanding of how to market Bellabeat products to capture the market from competitors. After studying the competitor data, it is clear that Bellabeats currently has a valid product competitor to entice new customers and maintain current ones.

The Bellabeat app and the Leaf wellness tracker are two products that are able to capture multiple types of activity logged by the client. Upon reviewing the data, it is clear that the most utilization was logged in the Activity dataset which includes activity minutes/type, distance, and calories. Within this marketing strategy it is important to emphasize the Leaf's capability to track steps and activity minutes - all of which will be conveniently logged into the Bellabeat app.

As mentioned earlier, to couple with this new marketing strategy - additional product features should accompany this release. Areas of improvement were noted in the competitor product's inability to motivate their clientele rated in the extreme BMI category, as they were ranked lowest in their total steps tracked. By

evaluating the days of the week that clients are emitting more or less activity, a behavior can be created in the Bellabeat app to emit notifications based on this 'lack' of activity logged. This notification can fire off according to a client-specific threashold calculated by their average total steps, current BMI category, and active utilization of Bellabeat products (activity tracking). In addition or in lieu of the notification feature, perhaps clients can sign up for daily step challenges, with higher step counts being challenged for the days they are less likely to log activity (based on user habits).

Overall Bellabeats is in a valuable position to capture competitor market shares and continue to improve on their product by analyzing competitor data as well as their in-house data.