DA_CapstoneProject

Ashutosh Singh

2024-04-09

We will be following 6-steps approach to analyze the data that would be presented to the Key Stakeholders in order to make informed Business Decisions.

Those 6-steps are as follows: 1. Ask 2. Prepare 3. Process 4. Analyze 5. Share 6. Act

The Key Stakeholders would be: **Urška Sršen**: Bellabeat's cofounder and Chief Creative Officer, **Sando Mur**: Mathematician and Bellabeat's cofounder; key member of the Bellabeat executive team, and **Bellabeat marketing analytics team**

Background of the company

Bellabeat is a high-tech company which was founded in 2013 that manufactures health-focused smart products. Bellabeat has grown rapidly and quickly positioned itself as a tech-driven wellness company for women. They specialize in 5 home grown products which are Bellabeat app, Leaf, Time, Spring and Bellabeat memberships.

Ask Phase

Analyze smart device usage data in order to gain insight into how consumers use smart devices.

Prepare Phase

Stakeholder encourages to use public data that explores smart device users' daily habits. FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius): This Kaggle data set contains personal fitness tracker from thirty fitbit users. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. It includes information about daily activity, steps, and heart rate that can be used to explore users' habits. During this phase, I encountered few limitations while working with the datasets, which are briefed below: a. Data is unclear if it included all genders and diversity b. The dataset of 33 users' is fairly small and we might encounter sampling bias c. The dataset doesn't contain much of demographic details d. The survey was collected for short period of time (~2 months), which tells that the data may not be latest

Process Phase

To be able to present my analysis, I chose R to clean, process and analyze the data.

```r
install.packages("tidyverse")
```

Environment setup in R

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

```r
install.packages("skimr")
```

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

```
install.packages("ggplot2")
```

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

```
install.packages("lubridate")
```

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

```
library(tidyverse)
```

Loading the packages

```
## -- 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)
library(skimr)
library(janitor)
```

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

```
daily_activity <- read_csv ("dailyActivity_merged.csv")
```

Data Import

```
## 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.
```

```
daily_Steps <- read_csv ("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.
```

```r
hourly_Steps <- read_csv("hourlySteps_merged.csv")
```

```
## Rows: 22099 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityHour
## 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.
```

```r
hourly_Calories <- read_csv ("hourlyCalories_merged.csv")
```

```
## Rows: 22099 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityHour
## 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.
```

```r

str(daily_activity)
```

Check the Structure of the data

```
## spc_tbl_ [940 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id                      : num [1:940] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ 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 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDate = col_character(),
##   ..   TotalSteps = col_double(),
##   ..   TotalDistance = col_double(),
##   ..   TrackerDistance = col_double(),
##   ..   LoggedActivitiesDistance = col_double(),
##   ..   VeryActiveDistance = col_double(),
##   ..   ModeratelyActiveDistance = col_double(),
##   ..   LightActiveDistance = col_double(),
##   ..   SedentaryActiveDistance = col_double(),
##   ..   VeryActiveMinutes = col_double(),
##   ..   FairlyActiveMinutes = col_double(),
##   ..   LightlyActiveMinutes = col_double(),
##   ..   SedentaryMinutes = col_double(),
##   ..   Calories = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(daily_Steps)
```

```
## spc_tbl_ [940 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id         : num [1:940] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDay: chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ StepTotal  : num [1:940] 13162 10735 10460 9762 12669 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDay = col_character(),
##   ..   StepTotal = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(hourly_Steps)
```

```
## spc_tbl_ [22,099 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id          : num [1:22099] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour: chr [1:22099] "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00 AM"
##  $ StepTotal   : num [1:22099] 373 160 151 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityHour = col_character(),
##   ..   StepTotal = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(hourly_Calories)
```

```
## spc_tbl_ [22,099 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id          : num [1:22099] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour: chr [1:22099] "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00 AM"
##  $ Calories    : num [1:22099] 81 61 59 47 48 48 48 47 68 141 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityHour = col_character(),
##   ..   Calories = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
head(daily_activity)
```

View the data

```
## # A tibble: 6 x 15
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
##        <dbl> <chr>             <dbl>         <dbl>           <dbl>
## 1 1503960366 4/12/2016         13162          8.5             8.5 
## 2 1503960366 4/13/2016         10735          6.97            6.97
## 3 1503960366 4/14/2016         10460          6.74            6.74
## 4 1503960366 4/15/2016          9762          6.28            6.28
## 5 1503960366 4/16/2016         12669          8.16            8.16
## 6 1503960366 4/17/2016          9705          6.48            6.48
## # i 10 more variables: LoggedActivitiesDistance <dbl>,
## #   VeryActiveDistance <dbl>, ModeratelyActiveDistance <dbl>,
## #   LightActiveDistance <dbl>, SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <dbl>, FairlyActiveMinutes <dbl>,
## #   LightlyActiveMinutes <dbl>, SedentaryMinutes <dbl>, Calories <dbl>
```

```r
head(daily_Steps)
```

```
## # A tibble: 6 x 3
##           Id ActivityDay StepTotal
##        <dbl> <chr>           <dbl>
## 1 1503960366 4/12/2016       13162
## 2 1503960366 4/13/2016       10735
## 3 1503960366 4/14/2016       10460
## 4 1503960366 4/15/2016        9762
## 5 1503960366 4/16/2016       12669
## 6 1503960366 4/17/2016        9705
```

```r
head(hourly_Steps)
```

```
## # A tibble: 6 x 3
##           Id ActivityHour          StepTotal
##        <dbl> <chr>                     <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM       373
## 2 1503960366 4/12/2016 1:00:00 AM        160
## 3 1503960366 4/12/2016 2:00:00 AM        151
## 4 1503960366 4/12/2016 3:00:00 AM          0
## 5 1503960366 4/12/2016 4:00:00 AM          0
## 6 1503960366 4/12/2016 5:00:00 AM          0
```

```r
head(hourly_Calories)
```

```
## # A tibble: 6 x 3
##           Id ActivityHour          Calories
##        <dbl> <chr>                    <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM       81
## 2 1503960366 4/12/2016 1:00:00 AM        61
## 3 1503960366 4/12/2016 2:00:00 AM        59
## 4 1503960366 4/12/2016 3:00:00 AM        47
## 5 1503960366 4/12/2016 4:00:00 AM        48
## 6 1503960366 4/12/2016 5:00:00 AM        48
```

```r
colnames(daily_activity)
```

Verify the column names

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

```r
colnames(daily_Steps)
```

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

```r
colnames(hourly_Steps)
```

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

```r
colnames(hourly_Calories)
```

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

```r
sum(duplicated(daily_activity))
```

Check for Duplicate Values

```
## [1] 0
```

```r
sum(duplicated(daily_Steps))
```

```
## [1] 0
```

```r
sum(duplicated(hourly_Steps))
```

```
## [1] 0
```

```r
sum(duplicated(hourly_Calories))
```

```
## [1] 0
```

```r
daily_activity <- daily_activity %>%
  drop_na()
daily_Steps <- daily_Steps %>%
  drop_na()
hourly_Steps <- hourly_Steps %>%
  drop_na()
hourly_Calories <- hourly_Calories %>%
  drop_na()
```

Remove missing values (if any)

Analyze and Share Phase

```
daily_activity %>%
  select(TotalSteps,
         TotalDistance,
         SedentaryMinutes, Calories) %>%
  summary()
```

Summarize daily activity

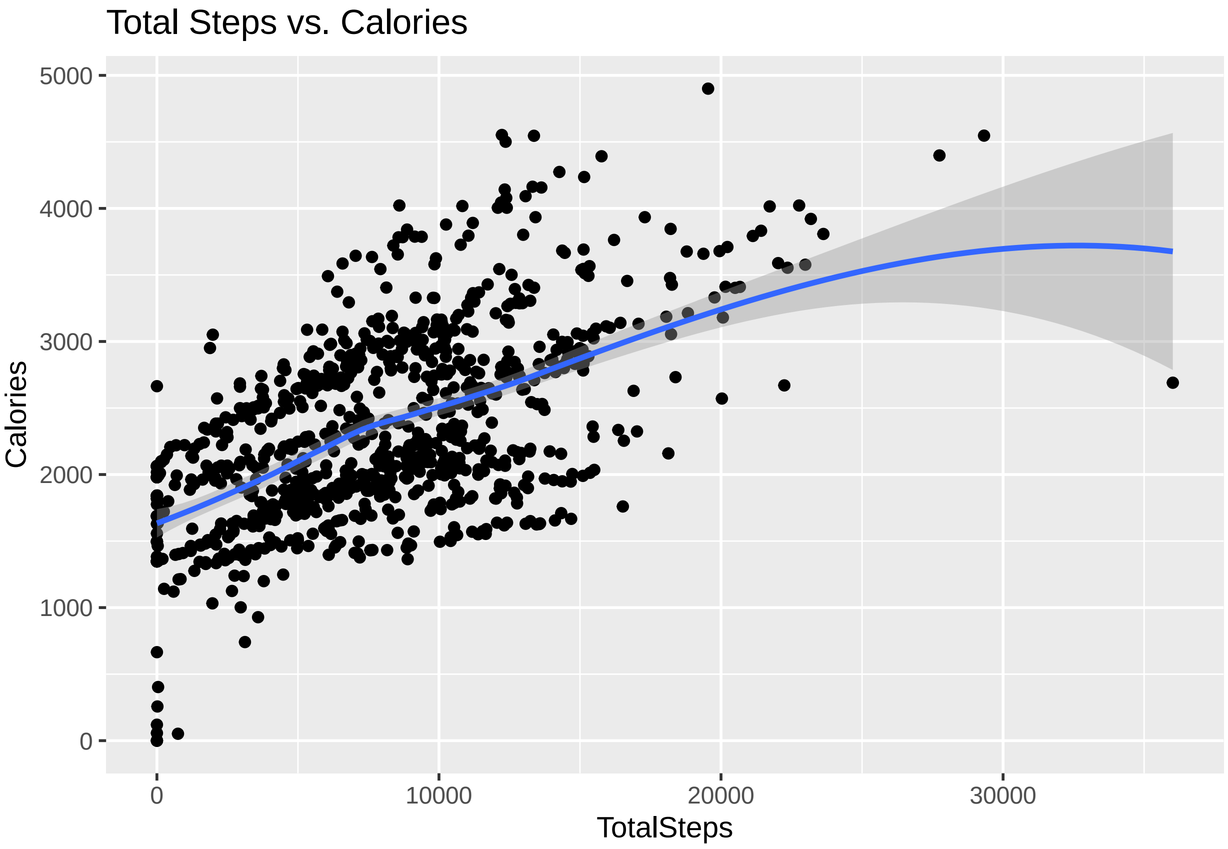
```
##    TotalSteps    TotalDistance    SedentaryMinutes    Calories   
##  Min.   :    0   Min.   : 0.000   Min.   :   0.0   Min.   :   0  
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 729.8   1st Qu.:1828  
##  Median : 7406   Median : 5.245   Median :1057.5   Median :2134  
##  Mean   : 7638   Mean   : 5.490   Mean   : 991.2   Mean   :2304  
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:1229.5   3rd Qu.:2793  
##  Max.   :36019   Max.   :28.030   Max.   :1440.0   Max.   :4900  
```

Outcome from the summary: 1. Average Total Steps taken is 7638 2. Average Sedentary time is 991 minutes, which is higher than expected. Recommend user to lower the sedentary time.

```
ggplot(data = daily_activity, aes(x = TotalSteps, y = Calories)) +
  geom_point() + geom_smooth() + labs(title ="Total Steps vs. Calories")
```

Total Steps vs Calories

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Hourly steps in a day I noticed that the Date and Time are merged in 1 column in the dataset, we will need to seperate date and time and 2 columns for this analysis

The below script will standardize the date and time

```
hourly_Steps<- hourly_Steps %>%
  rename(date_time = ActivityHour) %>%
  mutate(date_time = as.POSIXct(date_time, format ="%m/%d/%Y %I:%M:%S %p" , tz=Sys.timezone()))
```

Then we seperate date and time in 2 columns using below script:

```
hourly_Steps <- hourly_Steps %>%
  separate(date_time, into = c("date", "time"), sep= " ") %>%
  mutate(date = ymd(date))
```

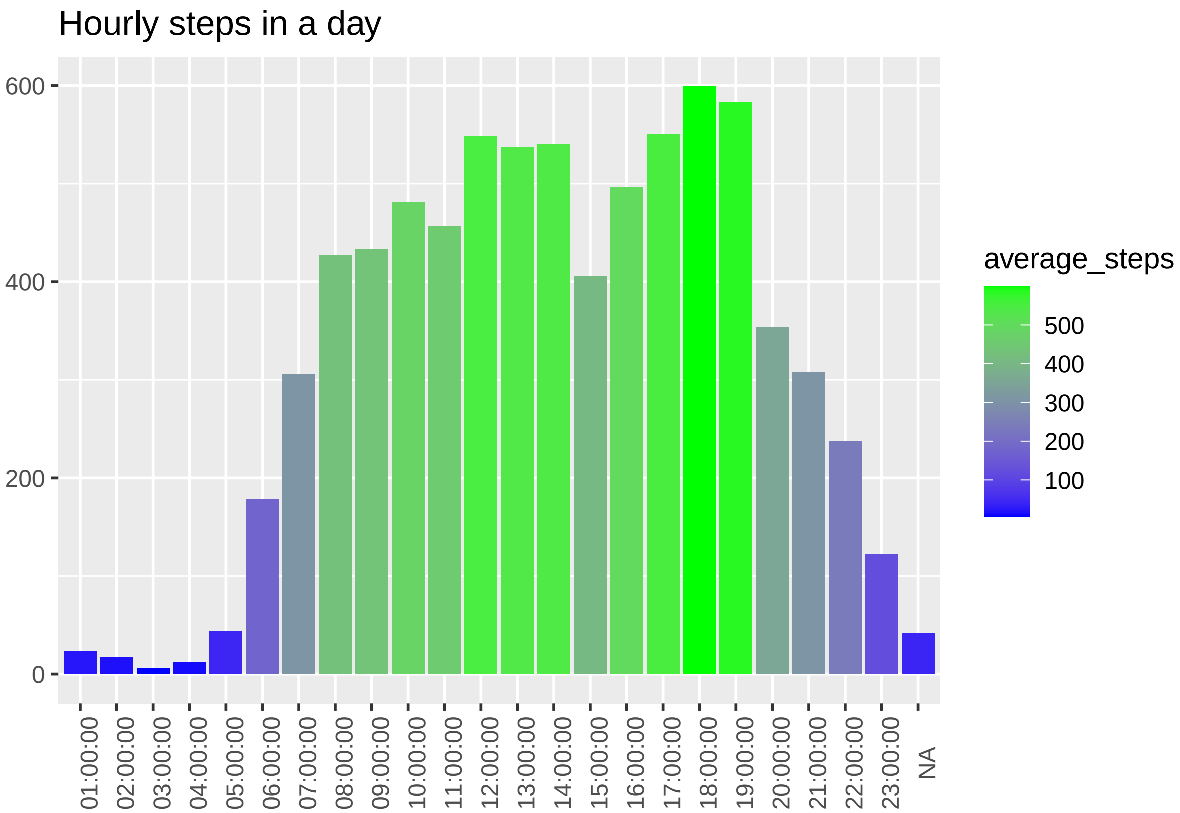
```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 934 rows [1, 25, 49, 73,
## 97, 121, 145, 169, 193, 217, 241, 265, 289, 313, 337, 361, 385, 409, 433, 457,
## ...].
```

```
head(hourly_Steps)
```

```
## # A tibble: 6 x 4
##           Id date       time     StepTotal
##        <dbl> <date>     <chr>        <dbl>
## 1 1503960366 2016-04-12 <NA>           373
## 2 1503960366 2016-04-12 01:00:00       160
## 3 1503960366 2016-04-12 02:00:00       151
## 4 1503960366 2016-04-12 03:00:00         0
## 5 1503960366 2016-04-12 04:00:00         0
## 6 1503960366 2016-04-12 05:00:00         0
```

Visualization of Hourly Steps in a day through graph

```
hourly_Steps %>%
  group_by(time) %>%
  summarize(average_steps = mean(StepTotal)) %>%
  ggplot() +
  geom_col(mapping = aes(x=time, y = average_steps, fill = average_steps)) +
  labs(title = "Hourly steps in a day", x="", y="") +
  scale_fill_gradient(low = "blue", high = "green")+
  theme(axis.text.x = element_text(angle = 90))
```



Observation People are more active between 8 AM to 5 PM as they walk more steps, while 3 PM being rest time for most of them.

Hourly Calories in a day The below script will standardize the date and time

```
hourly_Calories<- hourly_Calories %>%
  rename(date_time = ActivityHour) %>%
  mutate(date_time = as.POSIXct(date_time, format ="%m/%d/%Y %I:%M:%S %p" , tz=Sys.timezone()))
```

Then we seperate date and time in 2 columns using below script:

```
hourly_Calories <- hourly_Calories %>%
  separate(date_time, into = c("date", "time"), sep= " ") %>%
  mutate(date = ymd(date))
```

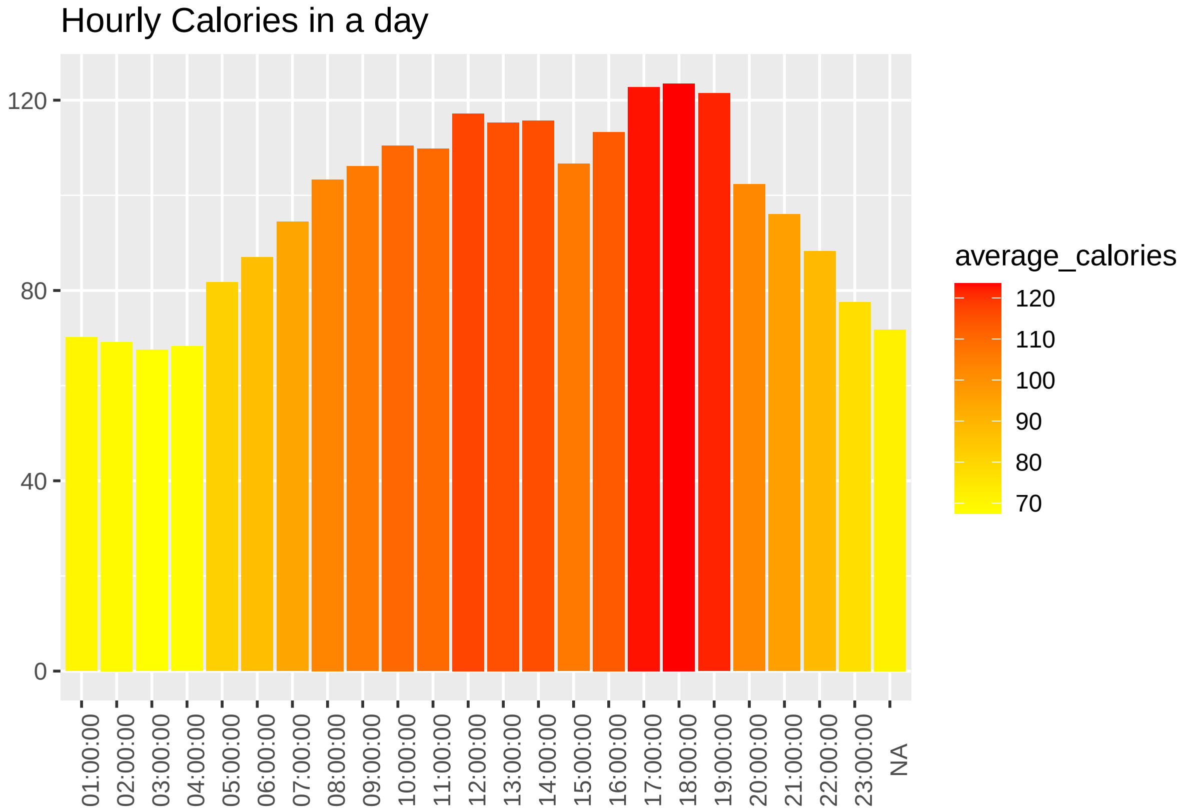
```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 934 rows [1, 25, 49, 73,
## 97, 121, 145, 169, 193, 217, 241, 265, 289, 313, 337, 361, 385, 409, 433, 457,
## ...].
```

```
head(hourly_Calories)
```

```
## # A tibble: 6 x 4
##           Id date       time     Calories
##        <dbl> <date>     <chr>       <dbl>
## 1 1503960366 2016-04-12 <NA>           81
## 2 1503960366 2016-04-12 01:00:00       61
## 3 1503960366 2016-04-12 02:00:00       59
## 4 1503960366 2016-04-12 03:00:00       47
## 5 1503960366 2016-04-12 04:00:00       48
## 6 1503960366 2016-04-12 05:00:00       48
```

Visualization of Hourly Steps in a day through graph

```
hourly_Calories %>%
  group_by(time) %>%
  summarize(average_calories = mean(Calories)) %>%
  ggplot() +
  geom_col(mapping = aes(x=time, y = average_calories, fill = average_calories)) +
  labs(title = "Hourly Calories in a day", x="", y="") +
  scale_fill_gradient(low = "yellow", high = "red")+
  theme(axis.text.x = element_text(angle = 90))
```



Observation More Calories are burnt between 8 AM to 5 PM as they walk more steps, while 3 PM being rest time for most of them.

Act Phase

Here are some recommendations for company to utilize the data in improving their marketing strategy and take better business decisions. 1. Notifications in the app or device. For example: Remind the customer to drink water, go for a short walk, overall daily progress, etc. 2. Feature on app to be able to create a community and share the progress with others (avoid personal details). This might encourage people to perform better in the areas they're falling behind. 3. Ranking or VIP level status based on customer's achievements.

Thanks for reading