BellaBeat Capstone Project

Thapelo Mohaleamall

2026-01-06

About a company

Bellabeat, a high-tech manufacturer of health-focused products for women. Bellabeat is a successful small company, but they have the potential to become a larger player in the global smart device market. Urška Sršen, cofounder and Chief Creative Officer of Bellabeat, believes that analyzing smart device fitness data could help unlock new growth opportunities for the company

Ask Phase

Stakeholders:

- Urška Sršen: Bellabeat's cofounder and Chief Creative Officer
- Sando Mur: Mathematician and Bellabeat's cofounder; key member of the Bellabeat executive team
- Bellabeat marketing analytics team: A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat's marketing strategy.

Questions for the analysis

- What are some trends in smart device usage?
- How could these trends apply to Bellabeat customers?
- How could these trends help influence Bellabeat marketing strategy

Business task

Identify potential opportunities for growth and recommendations for the Bellabeat marketing strategy improvement based on trends in smart device usage.

Prepare Phase

1. Data Source Description

Dataset Name: FitBit Fitness Tracker Data (CC0: Public Domain)

Source: Hosted on Kaggle by user Mobius.

Method: collected via a distributed survey on Amazon Mechanical Turk.

Timeframe: March 12, 2016 – May 12, 2016.

Participants: 33 eligible Fitbit users.

2. ROCCC Analysis (Data Credibility Check)

Data Credibility and Limitations (ROCCC) In data analytics, I used ROCCC to check if data is good enough. Here is how this dataset scores:

Reliable (Low): The sample size is extremely small (33 users). It may not represent the broader population accurately.

Original (Low): This is third-party data collected via Amazon Mechanical Turk, not original first-party data collected by Bellabeat.

Comprehensive (Low): The dataset is missing key demographic information. Most importantly, we do not know the gender of the users. Since Bellabeat creates products specifically for women, this is a significant bias/limitation.

Current (Low): The data is from 2016. Smart device usage habits have changed significantly since then.

Cited (High): The dataset is well-documented in the public domain.

Licensing, Privacy, and Integrity

The dataset is released under the CC0: Public Domain license, ensuring it is open for use without copyright restrictions. Privacy is maintained as all personal data is anonymized, with users distinguished only by unique ID numbers. However, data integrity issues were identified during the initial review. There are inconsistencies in date formats across files, and the data contains many "0" values. These zeros are ambiguous, as they could represent sedentary time or periods when the device was not worn—a distinction that must be addressed during the cleaning process. Additionally, the minute-level data files are too large for standard spreadsheet processing, indicating a need for R for deeper analysis.

Description of all data sources

Table Name	File Type	Discription
dailyActivity_merged	csv	Daily Activity for 31 days of 31 users
dailyCalories_merged	csv	Daily Calories for 31 days for 33 users
dailyIntensities_merged	csv	Daily Intensity for 31 days of 33 users
heartrate_seconds_merged	csv	Date and time for heart rate logs for 7 users
hourlyCalories_merged	csv	Hourly Calories for 31 days of 33 users

Table Name	File Type	Discription
hourlyIntensities_merged	csv	Hourly Intensity for 31 days of 33 users
hourlySteps_merged	csv	Hourly steps for 31 days of 33 users
minuteCaloriesNarrow_merged	csv	Calories burnt every minute for 31 days of 33 users
minuteCaloriesWide_merged	csv	Calories burnt every minute for 31 days of 33 users for all rows
minuteIntensitiesNarrow_merged	csv	Intensities counted every minute for 31 days of 33 users
minuteIntensitiesWide_merged	csv	Intensities counted every minute for 31 days of 33 users for all rows
minuteMETsNarrow_merged	csv	Physical activity compared when active and resting
minuteSleep_merged	csv	Sleep log by Minute for 24 users over 31 days. Value column not specified
minuteStepsNarrow_merged	csv	steps tracked in minutes for 31 days for 33 users for every row
minuteStepsWide_merged	csv	steps tracked in minutes for 31 days for 33 users
sleepDay_merged	csv	Sleep logs
weightLogInfo_merged	csv	Weight in kg and pounds tracked for 31 days of 8 users

Process Phase

Analysis will be primarily be focused in R due to the amount of data and being able to create visualizations to share to stakeholders

Loading packages

the following packages will be used to help for cleaning and visualization

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.6
## v forcats   1.0.1     v stringr   1.6.0
## v ggplot2   4.0.1     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.2.0
## -- 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 error
```

```
library(lubridate)
library(dplyr)
```

```r
library(ggplot2)
library(tidyr)
library(skimr)
library(janitor)
```

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

```r
library(lubridate)
```

based on the datasets we have the following will be used for our Analysis and answaer our business task. the following are:

- dailyActivity_merged
- sleepDay_merged
- dailySteps_merged
- hourlyCalories_merged
- hourlySteps_merged

```r
#importing datasets that will needed
dailyAcitviy <- read_csv("~/R project dataset/dailyActivity_merged.csv")
```

imorting and viewing of dataset

```
## 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.
```

```r
dailySteps <- read_csv("~/R project dataset/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
Sleep <- read_csv("~/R project dataset/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.
```

```r
HourlyCalories <- read_csv("~/R project dataset/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
HourlySteps <- read_csv("~/R project dataset/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
##viewing dataset

head(dailyAcitviy)
```

```
## # 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(dailySteps)
```

```
## # 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(Sleep)
```

```
## # A tibble: 6 x 5
##           Id SleepDay        TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##        <dbl> <chr>                       <dbl>              <dbl>          <dbl>
## 1 1503960366 4/12/2016 12:0~                 1                327            346
## 2 1503960366 4/13/2016 12:0~                 2                384            407
## 3 1503960366 4/15/2016 12:0~                 1                412            442
## 4 1503960366 4/16/2016 12:0~                 2                340            367
## 5 1503960366 4/17/2016 12:0~                 1                700            712
## 6 1503960366 4/19/2016 12:0~                 1                304            320
```

```r
head(HourlyCalories)
```

```
## # 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
head(HourlySteps)
```

```
## # 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
```

cleaning and formating of the dataset

in this section we will be cleaning the dataset that will be used for analysis and check for nulls,duplicates and necassary formating.

```r
#checking for unique values
n_unique(dailyAcitviy$Id)#33 users
```

cheking for unique values,duplicates and nulls

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

```r
n_unique(Sleep$Id)#24 users
```

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

```r
n_unique(dailySteps$Id)#33 users
```

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

```r
n_unique(HourlyCalories$Id)#33 users
```

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

```r
n_unique(HourlySteps$Id)#33 users
```

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

```r
#checking for duplicates
sum(duplicated(dailyAcitviy))#0 duplicates
```

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

```r
sum(duplicated(Sleep))#3 duplicates
```

```
## [1] 3
```

```r
sum(duplicated(dailySteps))#0 duplicates
```

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

```r
sum(duplicated(HourlyCalories))#0
```

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

```r
sum(duplicated(HourlySteps))#0
```

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

```r
#removing the duplicates and and NA
dailyActivity <- dailyAcitviy %>%
  distinct() %>%
  drop_na()

dailySleep <- Sleep %>%
  distinct() %>%
  drop_na()

HourlySteps <- HourlySteps %>%
  distinct() %>%
  drop_na()

HourlyCalories <- HourlyCalories %>%
  distinct() %>%
  drop_na()

dailySteps <- dailySteps %>%
  distinct() %>%
  drop_na()

sum(duplicated(dailySteps))#verifying removed duplicates
```

cleaning and formating

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

table names are acceptable and will continue to formating dates and time for datasets

```r
#sorting out the date formats
dailyAcitviy <- dailyAcitviy %>%
  mutate(ActivityDate = as_date(ActivityDate,format = "%m/%d/%Y"))

dailySteps <- dailySteps %>%
  mutate(ActivityDay = as_date(ActivityDay,format = "%m/%d/%Y"))

dailySleep <- dailySleep %>%
  mutate(SleepDay = as_datetime(SleepDay,format = "%m/%d/%Y %I:%M:%S %p"))

HourlySteps <- HourlySteps %>%
  mutate(ActivityHour = as_datetime(ActivityHour,format = "%m/%d/%Y %I:%M:%S %p"))

HourlyCalories <- HourlyCalories %>%
  mutate(ActivityHour = as_datetime(ActivityHour,format = "%m/%d/%Y %I:%M:%S %p"))

head(dailySleep)
```

```
## # A tibble: 6 x 5
##       Id SleepDay            TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##    <dbl> <dttm>                          <dbl>              <dbl>          <dbl>
## 1 1.50e9 2016-04-12 00:00:00                 1                327            346
```

```
## 2 1.50e9 2016-04-13 00:00:00                 2                384             407
## 3 1.50e9 2016-04-15 00:00:00                 1                412             442
## 4 1.50e9 2016-04-16 00:00:00                 2                340             367
## 5 1.50e9 2016-04-17 00:00:00                 1                700             712
## 6 1.50e9 2016-04-19 00:00:00                 1                304             320
```

```
head(HourlySteps)
```

```
## # A tibble: 6 x 3
##           Id ActivityHour        StepTotal
##        <dbl> <dttm>                  <dbl>
## 1 1503960366 2016-04-12 00:00:00       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
```

```
head(HourlyCalories)
```

```
## # A tibble: 6 x 3
##           Id ActivityHour        Calories
##        <dbl> <dttm>                 <dbl>
## 1 1503960366 2016-04-12 00:00:00       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
```

changing all date columns to be the same

```
dailyAcitviy <- dailyAcitviy %>%
  rename(Date = ActivityDate)
dailySteps <- dailySteps %>%
  rename(Date = ActivityDay)
dailySleep <- dailySleep %>%
  rename(Date = SleepDay)
HourlySteps <- HourlySteps %>%
  rename(Date = ActivityHour)
HourlyCalories <- HourlyCalories %>%
  rename(Date = ActivityHour)
```

all date and time sorted

now we will be merging tables specifically the daily activity and daily sleep will be merged

```
dailyActivitySleep <- merge(dailyAcitviy, dailySleep, by=c ("Id", "Date"))
glimpse(dailyActivitySleep)
```

```
## Rows: 410
## Columns: 18
```

```
## $ Id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ Date                     <date> 2016-04-12, 2016-04-13, 2016-04-15, 2016-04-~
## $ TotalSteps               <dbl> 13162, 10735, 9762, 12669, 9705, 15506, 10544~
## $ TotalDistance            <dbl> 8.50, 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6.3~
## $ TrackerDistance          <dbl> 8.50, 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6.3~
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveDistance       <dbl> 1.88, 1.57, 2.14, 2.71, 3.19, 3.53, 1.96, 1.3~
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 1.26, 0.41, 0.78, 1.32, 0.48, 0.3~
## $ LightActiveDistance      <dbl> 6.06, 4.71, 2.83, 5.04, 2.51, 5.03, 4.24, 4.6~
## $ SedentaryActiveDistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveMinutes        <dbl> 25, 21, 29, 36, 38, 50, 28, 19, 41, 39, 73, 3~
## $ FairlyActiveMinutes      <dbl> 13, 19, 34, 10, 20, 31, 12, 8, 21, 5, 14, 23,~
## $ LightlyActiveMinutes     <dbl> 328, 217, 209, 221, 164, 264, 205, 211, 262, ~
## $ SedentaryMinutes         <dbl> 728, 776, 726, 773, 539, 775, 818, 838, 732, ~
## $ Calories                 <dbl> 1985, 1797, 1745, 1863, 1728, 2035, 1786, 177~
## $ TotalSleepRecords        <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ TotalMinutesAsleep       <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361, ~
## $ TotalTimeInBed           <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384, ~
```

```
View(dailyActivitySleep)
```

Analyze Phase

We will analyze trends of the users of FitBit and determine if that can help us on BellaBeat's marketing strategy.

here is a summary for the following dataset:

```
dailyActivitySleep %>%
  select(TotalSteps,TotalDistance,Calories,TotalTimeInBed,TotalMinutesAsleep) %>%
  summary()
```

```
##    TotalSteps    TotalDistance       Calories     TotalTimeInBed
##  Min.   :   17   Min.   : 0.010   Min.   : 257   Min.   : 61.0
##  1st Qu.: 5189   1st Qu.: 3.592   1st Qu.:1841   1st Qu.:403.8
##  Median : 8913   Median : 6.270   Median :2207   Median :463.0
##  Mean   : 8515   Mean   : 6.012   Mean   :2389   Mean   :458.5
##  3rd Qu.:11370   3rd Qu.: 8.005   3rd Qu.:2920   3rd Qu.:526.0
##  Max.   :22770   Max.   :17.540   Max.   :4900   Max.   :961.0
##  TotalMinutesAsleep
##  Min.   : 58.0
##  1st Qu.:361.0
##  Median :432.5
##  Mean   :419.2
##  3rd Qu.:490.0
##  Max.   :796.0
```

Visualisations

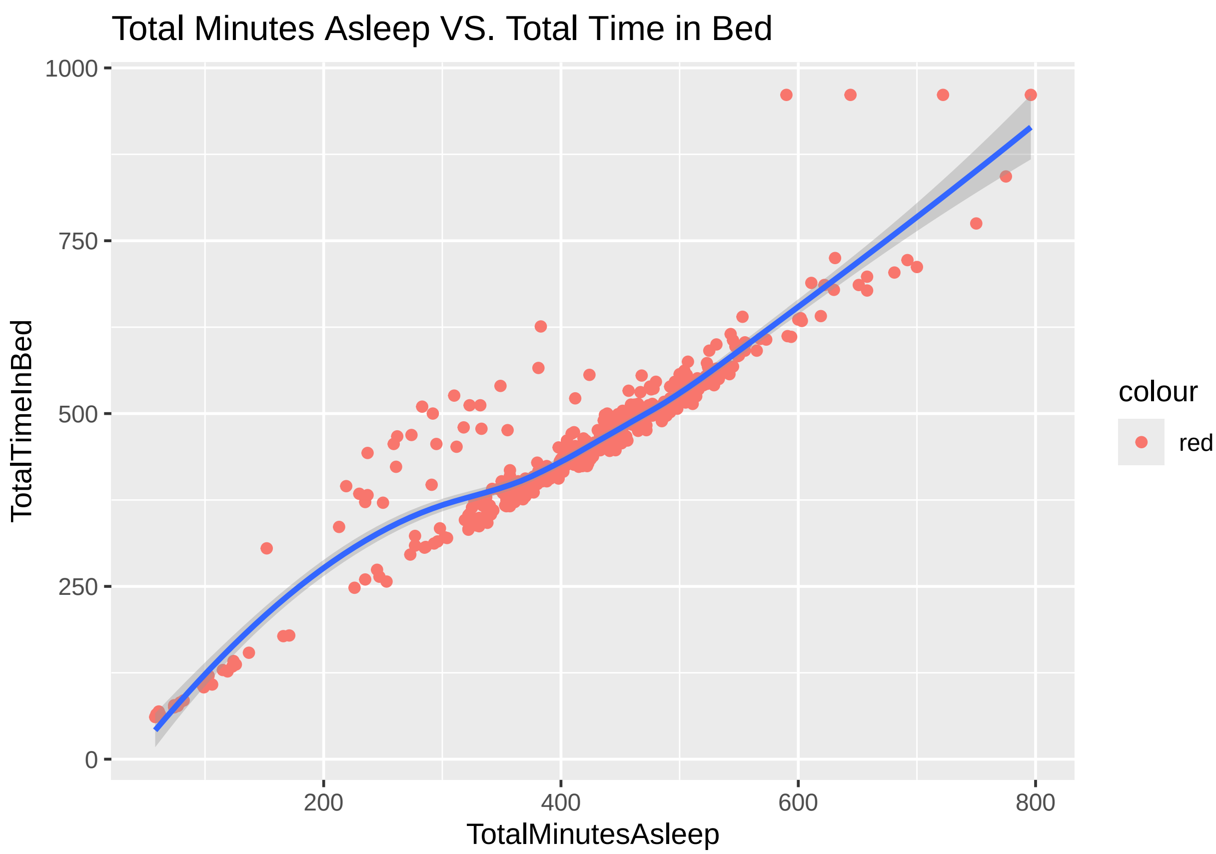
```r
ggplot(data = dailyActivitySleep)+
  geom_jitter(mapping = aes(x=TotalSteps,y=Calories,colour = "red"))+
  geom_smooth(mapping = aes(x=TotalSteps,y=Calories))+
  labs(title = "Total Steps VS. Calories")
```

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

The analysis reveals a positive correlation between daily steps and calorie expenditure. However, the data also highlights significant individual variance; users taking the same number of steps often show vastly different calorie burns, likely due to differences in weight or exercise intensity

```r
ggplot(data = dailyActivitySleep)+
  geom_point(mapping = aes(x=TotalMinutesAsleep,y=TotalTimeInBed,colour = "red"))+
  geom_smooth(mapping = aes(x=TotalMinutesAsleep,y=TotalTimeInBed))+
  labs(title = "Total Minutes Asleep VS. Total Time in Bed")
```

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



The analysis of sleep data shows a strong linear correlation between time in bed and actual sleep duration. However, significant outliers exist where users spend substantially more time in bed than asleep, indicating potential issues with sleep or restlessness.

```
hourly_Calories <- HourlyCalories %>%
  separate(Date,into = c("Date","Time"),sep = " ") %>%
  mutate(Date = ymd(Date))
```

histogram showing the average calories burnt throughout the day

```
## 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,
## ...].
```

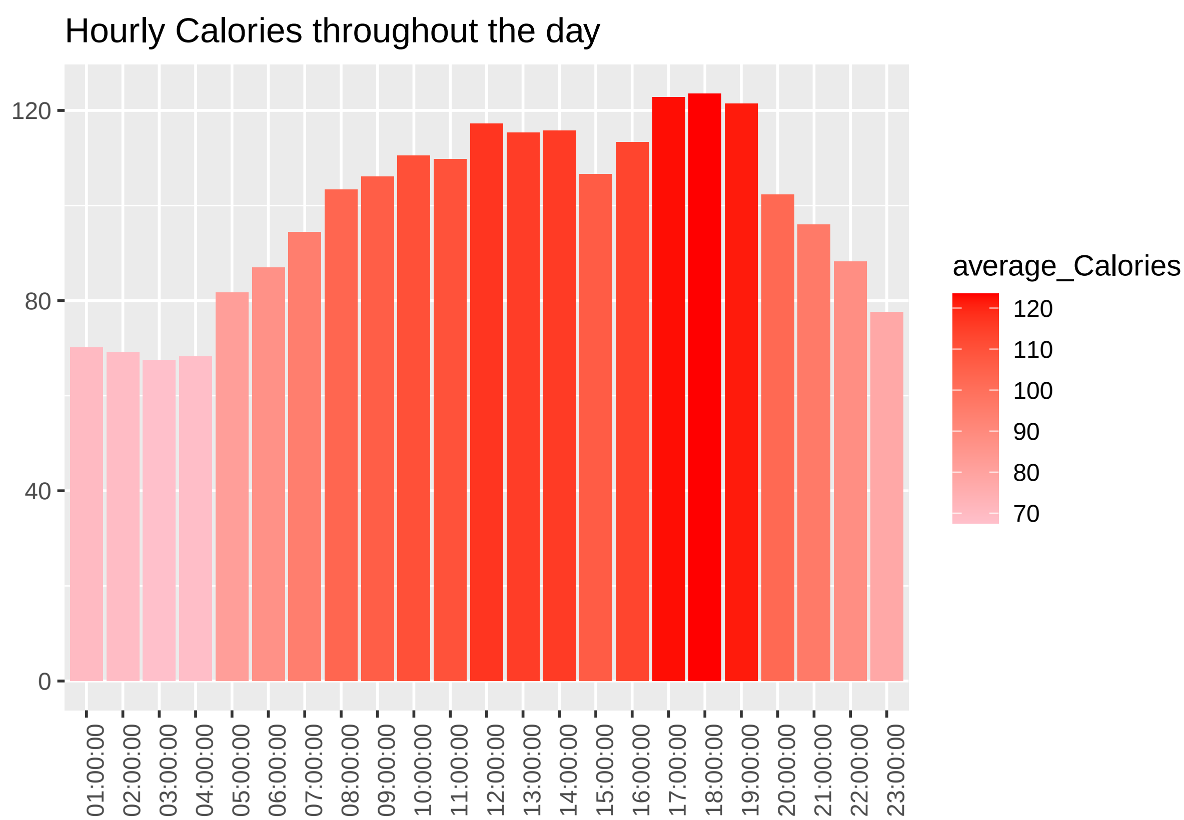
```
hourly_Calories <- hourly_Calories %>%
  drop_na()

head(hourly_Calories)
```

```
## # A tibble: 6 x 4
##           Id Date       Time     Calories
##        <dbl> <date>     <chr>       <dbl>
```

```
## 1 1503960366 2016-04-12 01:00:00       61
## 2 1503960366 2016-04-12 02:00:00       59
## 3 1503960366 2016-04-12 03:00:00       47
## 4 1503960366 2016-04-12 04:00:00       48
## 5 1503960366 2016-04-12 05:00:00       48
## 6 1503960366 2016-04-12 06:00:00       48
```

```r
hourly_Calories %>%
  group_by(Time) %>%
  summarise(average_Calories = mean(Calories)) %>%
  ggplot()+
  geom_col(mapping = aes(x=Time,y= average_Calories,fill = average_Calories))+
  labs(title = "Hourly Calories throughout the day",x="",y="")+
  scale_fill_gradient(low = "pink",high = "red")+
  theme(axis.text.x = element_text(angle = 90))
```



The analysis of hourly calorie data reveals a distinct 'evening peak' in user activity, with the highest energy expenditure occurring between 5:00 PM and 7:00 PM. In contrast, mornings show a slower, gradual ramp-up. Additionally, a noticeable dip in activity occurs around 3:00 PM.

```r
hourly_Steps <- HourlySteps %>%
  separate(Date,into = c("Date", "Time"), sep= " ") %>%
  mutate(Date = ymd(Date)) %>%
  drop_na()
```

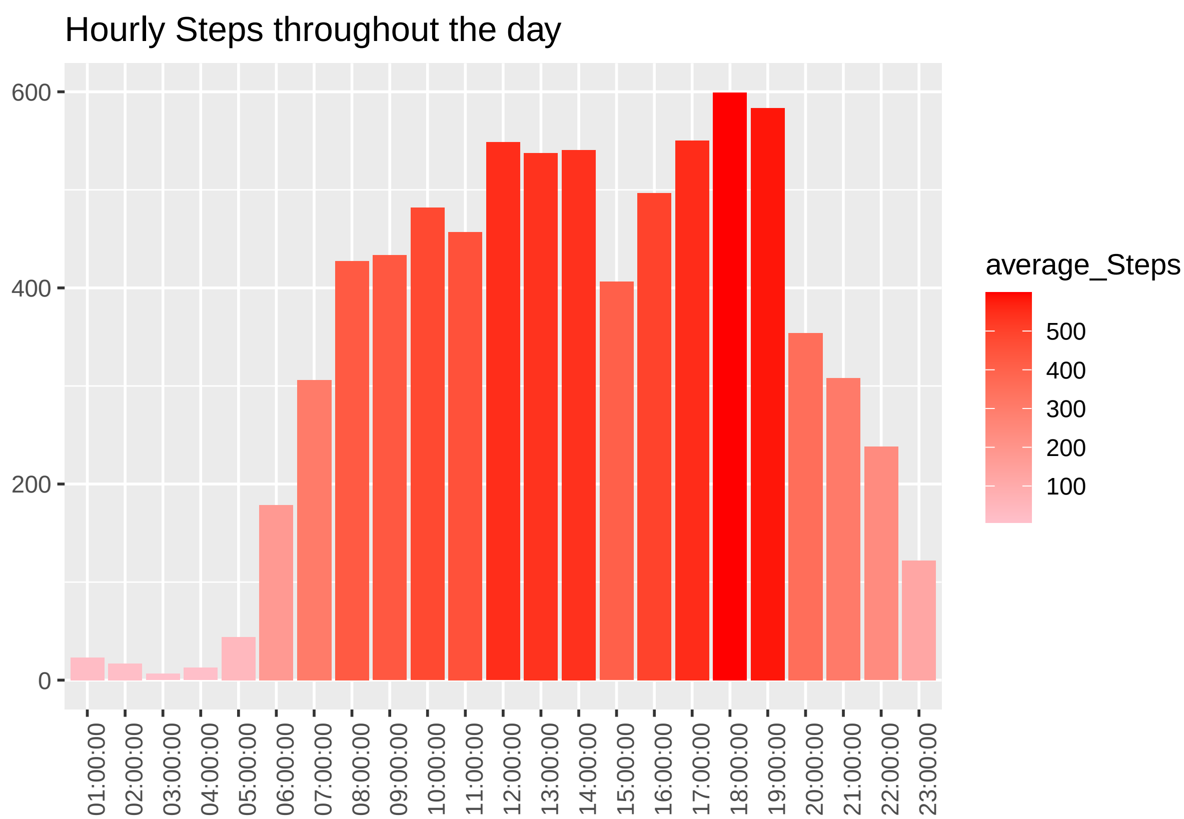
histogram for average hourly steps

```
## 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 01:00:00       160
## 2 1503960366 2016-04-12 02:00:00       151
## 3 1503960366 2016-04-12 03:00:00         0
## 4 1503960366 2016-04-12 04:00:00         0
## 5 1503960366 2016-04-12 05:00:00         0
## 6 1503960366 2016-04-12 06:00:00         0
```

```
hourly_Steps %>%
  group_by(Time) %>%
  summarise(average_Steps = mean(StepTotal)) %>%
  ggplot()+
  geom_col(mapping = aes(x=Time,y= average_Steps,fill = average_Steps))+
  labs(title = "Hourly Steps throughout the day",x="",y="")+
  scale_fill_gradient(low = "pink",high = "red")+
  theme(axis.text.x = element_text(angle = 90))
```



The analysis of hourly patterns confirms a strong correlation between step counts and calorie expenditure. Both metrics follow an identical daily rhythm: rising gradually in the morning, plateauing during lunch (12:00–14:00), dipping significantly at 3:00 PM, and reaching peak intensity between 5:00 PM and 7:00 PM.

device usage creating a pie chart representing the user types from low use to high use based on how they're usage of they're device giving us insight on the usage distribution

```
dailyUserType <- dailyActivitySleep %>%
  group_by(Id) %>%
  summarise(daysUsed=sum(n())) %>%
  mutate(usage = case_when(
    daysUsed >= 1 & daysUsed <= 10 ~"Low Use",
    daysUsed >= 11 & daysUsed <= 20 ~"Moderate Use",
    daysUsed >= 21 & daysUsed <= 31 ~"High Use",
  ))
head(dailyUserType)
```

```
## # A tibble: 6 x 3
##           Id daysUsed usage
##        <dbl>    <int> <chr>
## 1 1503960366       25 High Use
## 2 1644430081        4 Low Use
## 3 1844505072        3 Low Use
## 4 1927972279        5 Low Use
## 5 2026352035       28 High Use
## 6 2320127002        1 Low Use
```

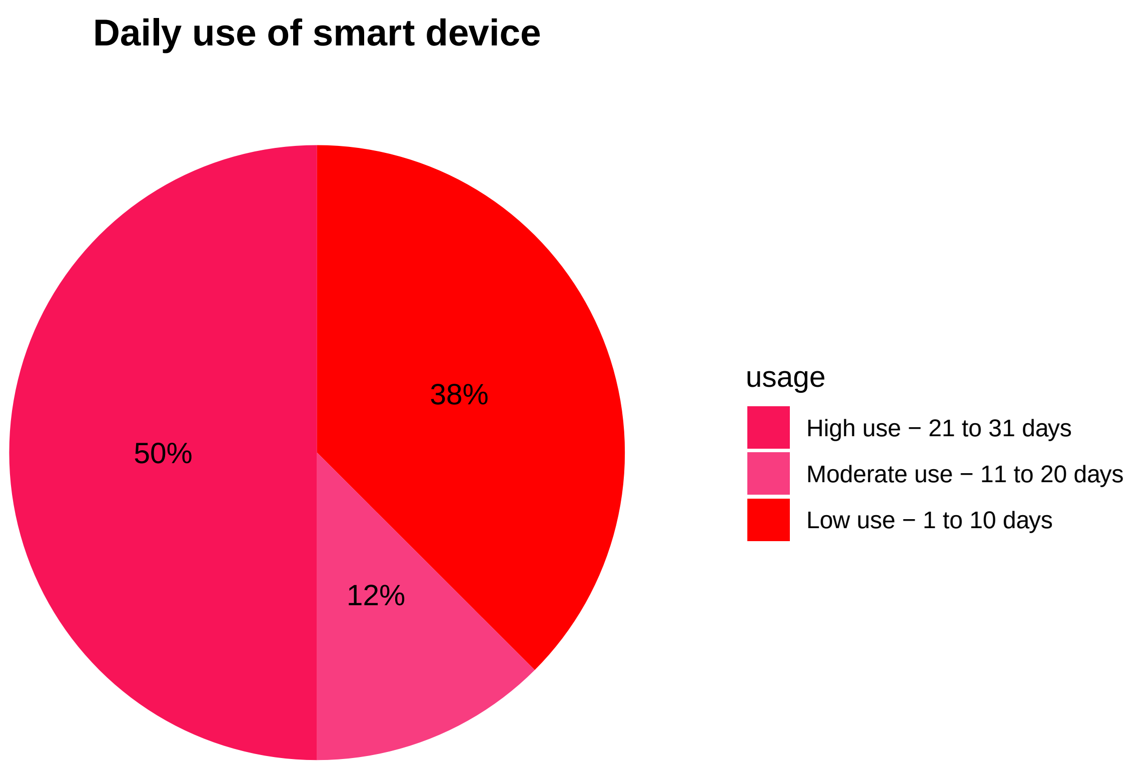
```
##creating a percentage dataframe that will be used to creat a barchart
dailyUserTypePercent <- dailyUserType %>%
  group_by(usage) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(usage) %>%
  summarise(totalPercent = total/totals) %>%
  mutate(labels = scales :: percent(totalPercent))

dailyUserTypePercent$usage <- factor(dailyUserTypePercent$usage,levels
                                     =c("High Use","Moderate Use","Low Use"))
head(dailyUserTypePercent)
```

```
## # A tibble: 3 x 3
##   usage        totalPercent labels
##   <fct>               <dbl> <chr>
## 1 High Use            0.5   50%
## 2 Low Use             0.375 38%
## 3 Moderate Use        0.125 12%
```

```
##creating the pie chart
dailyUserTypePercent %>%
  ggplot(aes(x="",y=totalPercent, fill=usage)) +
  geom_bar(stat = "identity", width = 1)+
  coord_polar("y", start=0)+
```

```
  theme_minimal()+
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5, size=14, face = "bold")) +
  geom_text(aes(label = labels),
            position = position_stack(vjust = 0.5))+
  scale_fill_manual(values = c("#F81458","#F83D80","red"),
                    labels = c("High use - 21 to 31 days",
                               "Moderate use - 11 to 20 days",
                               "Low use - 1 to 10 days"))+
  labs(title="Daily use of smart device")
```



the analysis shows Users tend to either commit fully to the device or abandon it quickly. There is very little middle ground. The fact that nearly 40% of users wear the device for less than 10 days suggests a significant issue with user retention or device comfort.

#Act Phase(Recomendations)

1. Notification Strategy The analysis showed a massive spike in activity between 5:00 PM – 7:00 PM, but a dip at 3:00 PM. Bellabeat should program the app to send customized notifications based on time of day.

for Example:

- 3:00 PM: "Energy dip detected? Try a 5-minute breathing exercises." (Promotes wellness).
- 4:30 PM: "The workday is almost over! You're 2,000 steps away from your goal. Let's crush it tonight!" (Promotes activity).

2. Focus on "Hidden" Sleep Issues the sleep scatter plot showed a disconnect between "Time in Bed" and "Actual Sleep," indicating many users lie awake in bed (insomnia/restlessness).This gap presents a strategic opportunity for Bellabeat to introduce 'Sleep Hygiene' features, helping users reduce the time they spend awake in bed through targeted mindfulness or relaxation content such as breathing or meditation strategies.

3. Retention Campaign The pie chart revealed that 38% of users stop wearing the device after a few days ("Low Use").Address the "drop-off" especially in marketing.such as giving a slogan like "The tracker you actually want to wear.",to address the usage issue ,in the marketing treat the device as jewlery by showcasing it in formal and casual clothing.

anther strategy is implementing challenges and games that women can play in they're phones connected to they're devices such creating a score board with other women especially friends wearing the device to track each others steps and see who reaches the highest score. another is giving rewards to those wearing they're device for 30 days