Bellabeat_Smart_Device_Data_Analysis

Muhammad Anwar

2023-09-28

Introduction

Welcome to my Bellabeat Data Analysis Case Study! In this case study, I'll be performing real-world tasks as a Data Analyst. To answer important business questions, I'll be following the steps of the data analysis process, which include asking questions, preparing data, processing it, analyzing, sharing insights, and taking action.

About Bellabeat

Bellabeat is a successful small company that specializes in creating health-focused products designed for women. With the potential to expand in the global smart device market, Bellabeat was founded in 2013 by Urška Sršen and Sando Mur. They manufacture health-focused smart products that collect data on activity, sleep, stress, and reproductive health, empowering women with insights about their health and habits. Over the years, Bellabeat has rapidly grown and established itself as a tech-driven wellness company for women.

Study Scenario

In this study, my focus is on one of Bellabeat's products, and I will analyze smart device data to gain insights into how consumers are using these devices. These insights will play a crucial role in guiding the company's marketing strategy.

Questions for the Analysis (Ask Phase)

During this phase, my aim was to better understand the data and the problem I'm addressing. To achieve this, I conducted additional research and posed specific questions:

- What are the notable trends in smart device usage? How can these trends inform Bellabeat's marketing strategy? Initially, the company needs to tailor their marketing efforts to meet the specific needs of their customers based on their usage of fitness smart devices. Following that, I will provide high-level recommendations for how these trends can shape Bellabeat's marketing strategy.
- Who are the key stakeholders? The primary stakeholders include Urška Sršen, Bellabeat's co-founder and Chief Creative Officer, Sando Mur, the mathematician and co-founder of Bellabeat. Additionally, collaboration with the broader Bellabeat marketing analytics team is essential to this analysis.

Business Task

Now, after obtaining answers to all of my questions during the ask phase, I can clearly define the business task:

"To analyze how Bellabeat customers use their smart devices and identify potential growth opportunities and recommendations for the Bellabeat marketing team based on trends in smart device usage."

Preparing the Data (Prepare Phase)

In this phase, I will perform the following tasks:

Downloading the Data

I will download and import the dataset for analysis. It is essential to ensure that the data is well-organized and credible. Additionally, I will sort and filter the data.

Data Source: Bellabeat encourages the use of publicly available data that explores smart device users' daily habits from FitBit FitnessTracker Data (CC0: Public Domain, dataset provided through Mobius). This Kaggle dataset comprises data from thirty Fitbit users who consented to share their personal tracker data, including minute-level information on physical activity, heart rate, and sleep monitoring. It contains details about daily activity, steps, and heart rate, offering insights into users' habits. You can download the dataset from this link: FitBit Fitness Tracker Data.

About the Dataset: The data was collected through a survey distributed via Amazon Mechanical Turk between December 3, 2016, and December 5, 2016, and consists of 18 CSV files.

Loading Packages

I will install and load specific R packages to assist in the analysis. To save space and prevent the display of R code execution messages and warnings, I will use the options `message=FALSE` and `warning=FALSE`. Additionally, I will include some data cleaning packages for further analysis, the last three packages in the list.

```
# installing packages
install.packages("tidyverse")
```

```
## 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)
```

```
install.packages("dplyr")
```

```
## 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("tidyr")
```

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

```
install.packages("here")
```

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

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

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

```
install.packages("janitor")
```

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

Now, I'm going to load these packages. And I'm using in my code the options message=FALSE and warning=FALSE, to save space. And to prevent printing of the execution of the R code generated and the warning messages.

```
# loading the libraries
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
```

```
library(lubridate)
library(dplyr)
library(ggplot2)
library(tidyr)
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
```

Importing dataset

Now, I'm going to Import all dataset. Then VIEW, CLEAN, FORMAT, and ORGANIZE the data. After reviewing all the dataset, I decided to make some asumptions and work only with these data for my analysis:
- dailyActivity_merged.csv

```
Activity <- read.csv("dailyActivity_merged.csv")
head(Activity)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366    4/12/2016      13162          8.50            8.50
## 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
```

```
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               1.88                     0.55
## 2                        0               1.57                     0.69
## 3                        0               2.44                     0.40
## 4                        0               2.14                     1.26
## 5                        0               2.71                     0.41
## 6                        0               3.19                     0.78
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                4.71                       0                21
## 3                3.91                       0                30
## 4                2.83                       0                29
## 5                5.04                       0                36
## 6                2.51                       0                38
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  13                  328              728     1985
## 2                  19                  217              776     1797
## 3                  11                  181             1218     1776
## 4                  34                  209              726     1745
## 5                  10                  221              773     1863
## 6                  20                  164              539     1728
```

```
colnames(Activity)
```

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

```
str(Activity)
```

```
## 'data.frame':    940 obs. of  15 variables:
##  $ Id                       : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate             : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps               : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
##  $ TotalDistance            : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance          : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance       : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance      : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes        : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes      : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes     : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes         : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ Calories                 : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

- dailyCalories_merged.csv

```
Calories <- read.csv("dailyCalories_merged.csv")
head(Calories)
```

```
##           Id ActivityDay Calories
## 1 1503960366   4/12/2016     1985
## 2 1503960366   4/13/2016     1797
## 3 1503960366   4/14/2016     1776
## 4 1503960366   4/15/2016     1745
## 5 1503960366   4/16/2016     1863
## 6 1503960366   4/17/2016     1728
```

```
colnames(Calories)
```

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

```
str(Calories)
```

```
## 'data.frame':    940 obs. of  3 variables:
##  $ Id         : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDay: chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ Calories   : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

- dailyIntensities_merged.csv

```
Intensities <- read.csv("dailyIntensities_merged.csv")
head(Intensities)
```

```
##           Id ActivityDay SedentaryMinutes LightlyActiveMinutes
## 1 1503960366   4/12/2016              728                  328
## 2 1503960366   4/13/2016              776                  217
## 3 1503960366   4/14/2016             1218                  181
## 4 1503960366   4/15/2016              726                  209
## 5 1503960366   4/16/2016              773                  221
## 6 1503960366   4/17/2016              539                  164
##   FairlyActiveMinutes VeryActiveMinutes SedentaryActiveDistance
## 1                  13                25                       0
## 2                  19                21                       0
## 3                  11                30                       0
## 4                  34                29                       0
## 5                  10                36                       0
## 6                  20                38                       0
##   LightActiveDistance ModeratelyActiveDistance VeryActiveDistance
## 1                6.06                     0.55               1.88
## 2                4.71                     0.69               1.57
## 3                3.91                     0.40               2.44
## 4                2.83                     1.26               2.14
## 5                5.04                     0.41               2.71
## 6                2.51                     0.78               3.19
```

```
colnames(Intensities)
```

```
##  [1] "Id"                       "ActivityDay"
##  [3] "SedentaryMinutes"         "LightlyActiveMinutes"
##  [5] "FairlyActiveMinutes"      "VeryActiveMinutes"
##  [7] "SedentaryActiveDistance"  "LightActiveDistance"
##  [9] "ModeratelyActiveDistance" "VeryActiveDistance"
```

```
str(Intensities)
```

```
## 'data.frame':    940 obs. of  10 variables:
##  $ Id                       : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDay              : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
```

```
##  $ SedentaryMinutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ LightlyActiveMinutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ FairlyActiveMinutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ VeryActiveMinutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ LightActiveDistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ VeryActiveDistance      : num  1.88 1.57 2.44 2.14 2.71 ...
```

- heartrate_seconds_merged.csv

```
Heartrate <- read.csv("heartrate_seconds_merged.csv")
head(Heartrate)
```

```
##           Id                  Time Value
## 1 2022484408 4/12/2016 7:21:00 AM    97
## 2 2022484408 4/12/2016 7:21:05 AM   102
## 3 2022484408 4/12/2016 7:21:10 AM   105
## 4 2022484408 4/12/2016 7:21:20 AM   103
## 5 2022484408 4/12/2016 7:21:25 AM   101
## 6 2022484408 4/12/2016 7:22:05 AM    95
```

```
colnames(Heartrate)
```

```
## [1] "Id"    "Time"  "Value"
```

```
str(Heartrate)
```

```
## 'data.frame':    2483658 obs. of  3 variables:
##  $ Id   : num  2.02e+09 2.02e+09 2.02e+09 2.02e+09 2.02e+09 ...
##  $ Time : chr  "4/12/2016 7:21:00 AM" "4/12/2016 7:21:05 AM" "4/12/2016 7:21:10 AM" "4/12/2016 7:21:1
##  $ Value: int  97 102 105 103 101 95 91 93 94 93 ...
```

- sleepDay_merged.csv

```
Sleep <- read.csv("sleepDay_merged.csv")
head(Sleep)
```

```
##           Id              SleepDay TotalSleepRecords TotalMinutesAsleep
## 1 1503960366 4/12/2016 12:00:00 AM                 1                327
## 2 1503960366 4/13/2016 12:00:00 AM                 2                384
## 3 1503960366 4/15/2016 12:00:00 AM                 1                412
## 4 1503960366 4/16/2016 12:00:00 AM                 2                340
## 5 1503960366 4/17/2016 12:00:00 AM                 1                700
## 6 1503960366 4/19/2016 12:00:00 AM                 1                304
##   TotalTimeInBed
## 1            346
## 2            407
## 3            442
## 4            367
## 5            712
## 6            320
```

```
colnames(Sleep)
```

```
## [1] "Id"                 "SleepDay"           "TotalSleepRecords"
## [4] "TotalMinutesAsleep" "TotalTimeInBed"
```

```r
str(Sleep)
```

```
## 'data.frame':    413 obs. of  5 variables:
##  $ Id                : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : chr  "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:00 AM" "
##  $ TotalSleepRecords : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : int  346 407 442 367 712 320 377 364 384 449 ...
```

- weightLogInfo_merged.csv

```r
Weight <- read.csv("weightLogInfo_merged.csv")
head(Weight)
```

```
##           Id                  Date WeightKg WeightPounds Fat   BMI
## 1 1503960366  5/2/2016 11:59:59 PM     52.6     115.9631  22 22.65
## 2 1503960366  5/3/2016 11:59:59 PM     52.6     115.9631  NA 22.65
## 3 1927972279  4/13/2016 1:08:52 AM    133.5     294.3171  NA 47.54
## 4 2873212765 4/21/2016 11:59:59 PM     56.7     125.0021  NA 21.45
## 5 2873212765 5/12/2016 11:59:59 PM     57.3     126.3249  NA 21.69
## 6 4319703577 4/17/2016 11:59:59 PM     72.4     159.6147  25 27.45
##   IsManualReport        LogId
## 1           True 1.462234e+12
## 2           True 1.462320e+12
## 3          False 1.460510e+12
## 4           True 1.461283e+12
## 5           True 1.463098e+12
## 6           True 1.460938e+12
```

```r
colnames(Weight)
```

```
## [1] "Id"             "Date"           "WeightKg"       "WeightPounds"
## [5] "Fat"            "BMI"            "IsManualReport" "LogId"
```

```r
str(Weight)
```

```
## 'data.frame':    67 obs. of  8 variables:
##  $ Id            : num  1.50e+09 1.50e+09 1.93e+09 2.87e+09 2.87e+09 ...
##  $ Date          : chr  "5/2/2016 11:59:59 PM" "5/3/2016 11:59:59 PM" "4/13/2016 1:08:52 AM" "4/21/2
##  $ WeightKg      : num  52.6 52.6 133.5 56.7 57.3 ...
##  $ WeightPounds  : num  116 116 294 125 126 ...
##  $ Fat           : int  22 NA NA NA NA 25 NA NA NA NA ...
##  $ BMI           : num  22.6 22.6 47.5 21.5 21.7 ...
##  $ IsManualReport: chr  "True" "True" "False" "True" ...
##  $ LogId         : num  1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...
```

So now, we can see that everything were imported correctly.

Data Cleaning (Process Phase)

Basic Data Cleaning

Now, I'm in the process of preparing, cleaning, and organizing the dataset for analysis. To quickly assess the data, I employed functions like `glimpse()` and `skim_without_charts`. Additionally, I improved the data's readability by using `clean_names()` to clean up column names.

Here are the steps I took to clean the data:

- For the datasets (Activity, Calories, and Intensities): During data cleaning, I did not encounter spelling errors, misfield values, missing values, extra or blank spaces, nor did I find any duplicates. To enhance formatting, I applied clear formatting techniques. Some data types were converted to numeric, and date columns were adjusted to the appropriate date type.
- For Sleep data: I identified and removed three duplicate entries.
- For Weight data: In one column, I noticed a significant number of missing values. I made the decision to exclude that column from further analysis.

Fixing Data Formatting

I observed issues with the timestamp data. To facilitate analysis, I will convert it into date-time format and split it into date and time components.

```
# Activity
Activity$ActivityDate=as.POSIXct(Activity$ActivityDate, format="%m/%d/%Y", tz=Sys.timezone())
Activity$date <- format(Activity$ActivityDate, format = "%m/%d/%y")
Activity$ActivityDate=as.Date(Activity$ActivityDate, format="%m/%d/%Y", tz=Sys.timezone())
Activity$date=as.Date(Activity$date, format="%m/%d/%Y")


# Intensities
Intensities$ActivityDay=as.Date(Intensities$ActivityDay, format="%m/%d/%Y", tz=Sys.timezone())


# Sleep
Sleep$SleepDay=as.POSIXct(Sleep$SleepDay, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone())
Sleep$date <- format(Sleep$SleepDay, format = "%m/%d/%y")
Sleep$date=as.Date(Sleep$date, "% m/% d/% y")
```

Summarizing the dataset (Analyze Phase)

Now that all the data is stored appropriately and has been prepared for analysis, I can start putting it to work. Let's look at the total number of participants in each data sets:

```
Activity %>%
summarise(Activity_participants = n_distinct(Activity$Id))
```

```
##   Activity_participants
## 1                    33
```

```
n_distinct(Calories$Id)
```

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

```
n_distinct(Intensities$Id)
```

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

```
n_distinct(Heartrate$Id)
```

```
## [1] 14
```

```
n_distinct(Sleep$Id)
```

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

```
n_distinct(Weight$Id)
```

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

Participants Overview

In the dataset, there are distinct participant counts across various categories:

- **Activity, Calories, and Intensities Data:** These datasets include a total of 33 participants.
- **Sleep Data:** The sleep dataset contains information from 24 participants.
- **Heart Rate Data:** For the heart rate dataset, there are records from 14 participants.
- **Weight Data:** The weight dataset comprises data from just 8 participants.

Given that the groups with 8 and 14 participants are relatively small, drawing meaningful recommendations and conclusions from these datasets may be challenging. Therefore, my primary focus for analysis will be on the following datasets: Activity, Calories, Intensities, and Sleep.

Below, you'll find brief summary statistics for each of these data frames.

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

```
##    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
```

Exploring the number of Intense active participants :

```
# Explore number of active minutes per category
Intensities %>%
select(VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, SedentaryMinutes) %>%
summary()
```

```
##  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
```

For the Calories dataframe:

```
# Calories
Calories %>%
select(Calories) %>%
summary()
```

```
##     Calories
##  Min.   :   0
##  1st Qu.:1828
##  Median :2134
##  Mean   :2304
##  3rd Qu.:2793
```

```
##  Max.   :4900
```

For the Sleep dataframe:

```
# Sleep
Sleep %>%
select(TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed) %>%
summary()
```

```
##  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.000     Min.   : 58.0      Min.   : 61.0
##  1st Qu.:1.000     1st Qu.:361.0      1st Qu.:403.0
##  Median :1.000     Median :433.0      Median :463.0
##  Mean   :1.119     Mean   :419.5      Mean   :458.6
##  3rd Qu.:1.000     3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :3.000     Max.   :796.0      Max.   :961.0
```

For the Weight dataframe:

```
# Weight
Weight %>%
select(WeightKg, Fat) %>%
summary()
```

```
##     WeightKg           Fat
##  Min.   : 52.60   Min.   :22.00
##  1st Qu.: 61.40   1st Qu.:22.75
##  Median : 62.50   Median :23.50
##  Mean   : 72.04   Mean   :23.50
##  3rd Qu.: 85.05   3rd Qu.:24.25
##  Max.   :133.50   Max.   :25.00
##                   NA's   :65
```

Key Insights

Summary of Key Findings

Here are the key insights derived from the analysis:

- **High Sedentary Time:** The average sedentary time exceeds 16 hours, signaling the urgent need for a robust marketing strategy to reduce this high sedentary behavior.
- **Light Activity:** The majority of participants exhibit light activity levels, often accompanied by a significant amount of sedentary time.
- **Sleep Patterns:** On average, participants sleep once a day for approximately 7 hours.
- **Step Count:** The average daily step count stands at 7,638, slightly below the CDC-recommended threshold. According to CDC research, taking 8,000 steps daily is associated with a 51% lower risk of all-cause mortality, while taking 12,000 steps daily is associated with a 65% lower risk compared to taking 4,000 steps.

Data Merging

Before proceeding with data visualization, I will merge two datasets, namely Activity and Sleep data, based on the "Id" column. It's important to note that there are more participant IDs in the Activity dataset than in the Sleep dataset. To ensure consistency, I will use the "inner_join" merge option, retaining the number of participants from the Sleep dataset.

```
Combined_data_inner <- merge(Sleep, Activity, by="Id")
n_distinct(Combined_data_inner$Id)
```

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

For the analysis, I'm thinking about using an "outer join" to make sure we keep all participants in the dataset. I can do this by adding the extra argument "all=TRUE" in my code.

```
Combined_data_outer <- merge(Sleep, Activity, by="Id", all = TRUE)
n_distinct(Combined_data_outer$Id)
```

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

Data Visualization (Share and Act Phases)

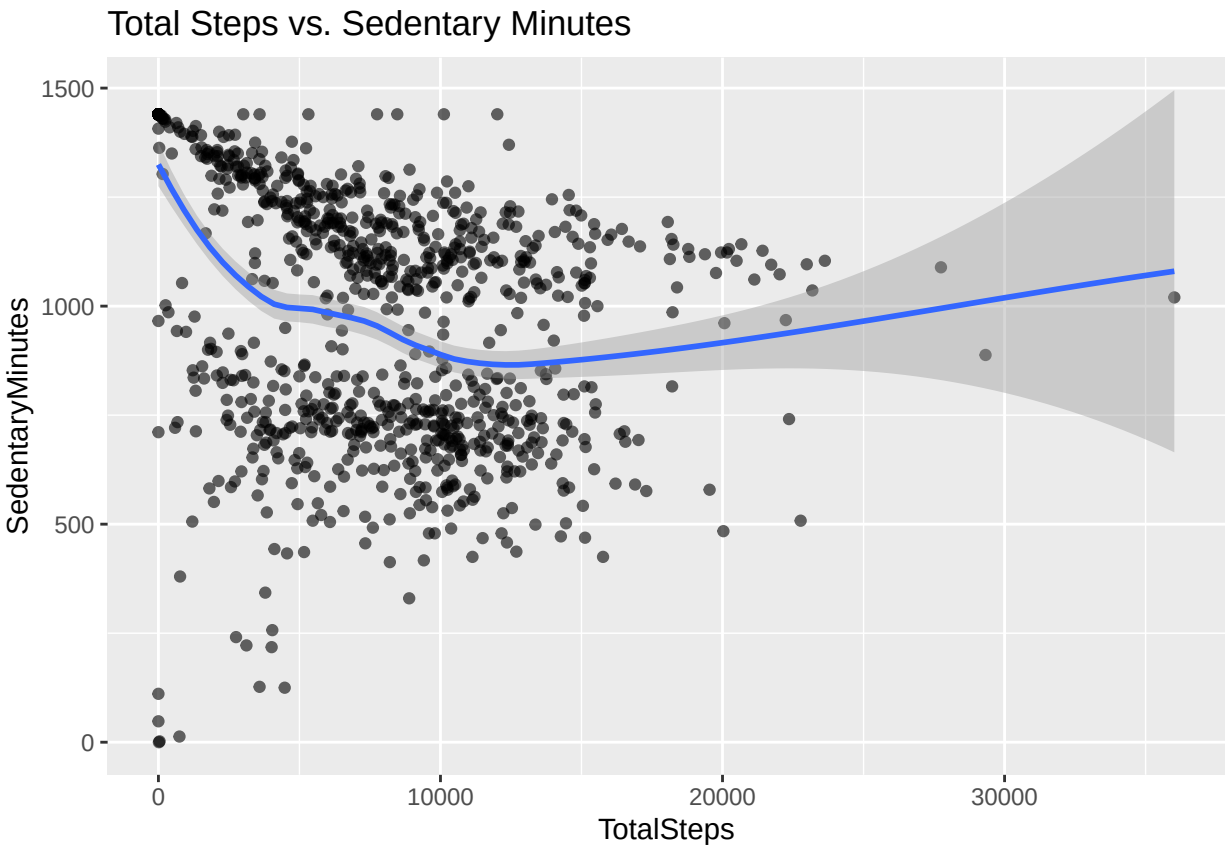
Now, let's move on to creating visual representations to explore some important aspects.

Relationship Between Daily Steps and Sedentary Time

In this section, we aim to understand the connection between the number of steps taken in a day and the amount of time spent in sedentary activities.

```
ggplot(data=Activity, aes(x=TotalSteps, y=SedentaryMinutes)) + geom_point(alpha = 0.6) + geom_smooth() +
```

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



I observe a clear inverse relationship between the number of steps taken and the amount of sedentary time. When individuals spend more time in sedentary activities, they tend to take fewer steps throughout the day. This data suggests that there is an opportunity for the company to focus its marketing efforts on customer

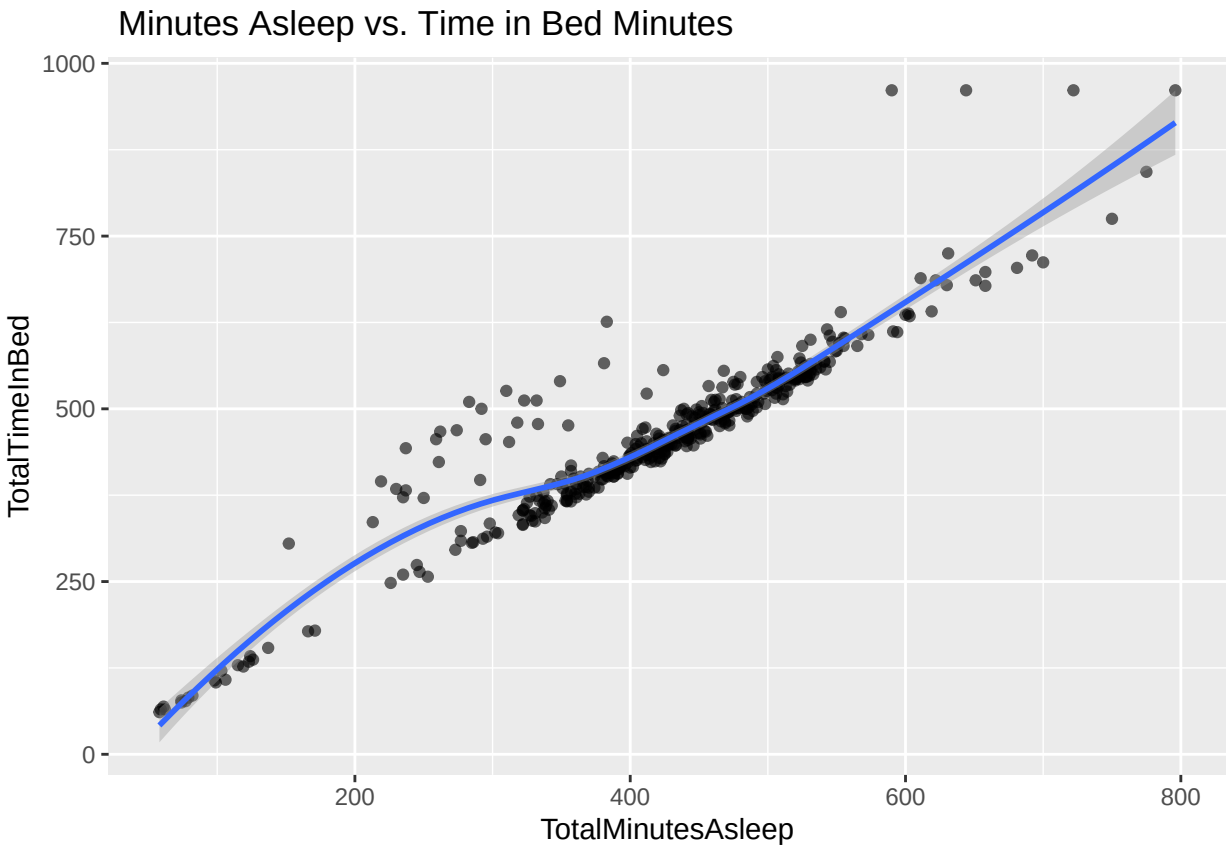
segments with high sedentary behavior. To achieve this, the company should explore strategies to encourage customers to engage in more walking and also monitor their daily step counts.

Exploring the Connection Between Sleep Duration and Time Spent in Bed

We're interested in understanding how the duration of sleep (measured in minutes asleep) is related to the total time spent in bed.

```
ggplot(data=Sleep, aes(x=TotalMinutesAsleep, y=TotalTimeInBed)) + geom_point(alpha = 0.6)+ geom_smooth()
```

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

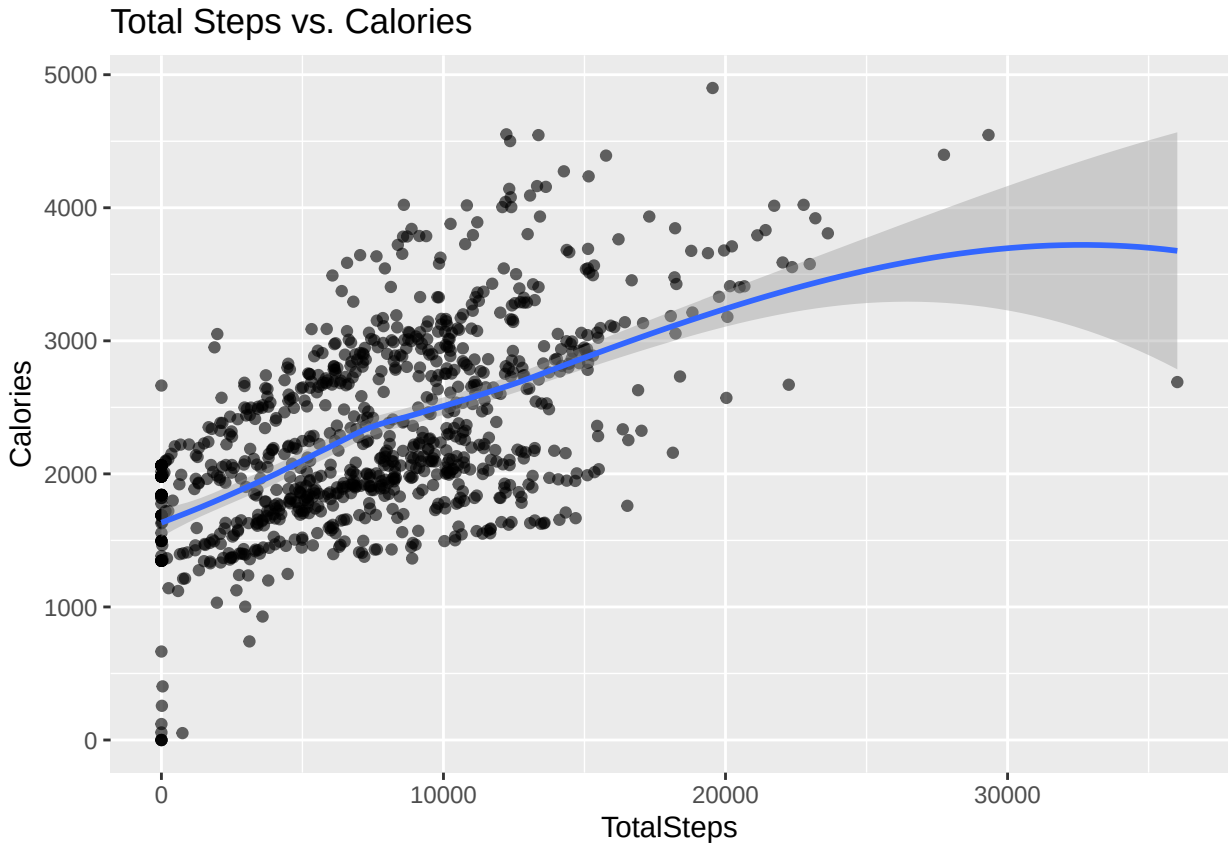


As anticipated, we notice a nearly straight-line trend when examining the relationship between minutes asleep and time in bed. To assist users in enhancing their sleep quality, the company should contemplate implementing notifications to encourage a regular sleep schedule.

Analyzing the Link Between Daily Steps and Caloric Expenditure We want to investigate the association between the number of steps taken and the calories burned (caloric expenditure).

```
ggplot(data=Activity, aes(x=TotalSteps, y=Calories)) + geom_point(alpha = 0.6) + geom_smooth() + labs(t
```

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



We observe a clear positive relationship between the total number of steps taken and the calories burned. When we are more physically active, our calorie expenditure increases.

Intensities Data

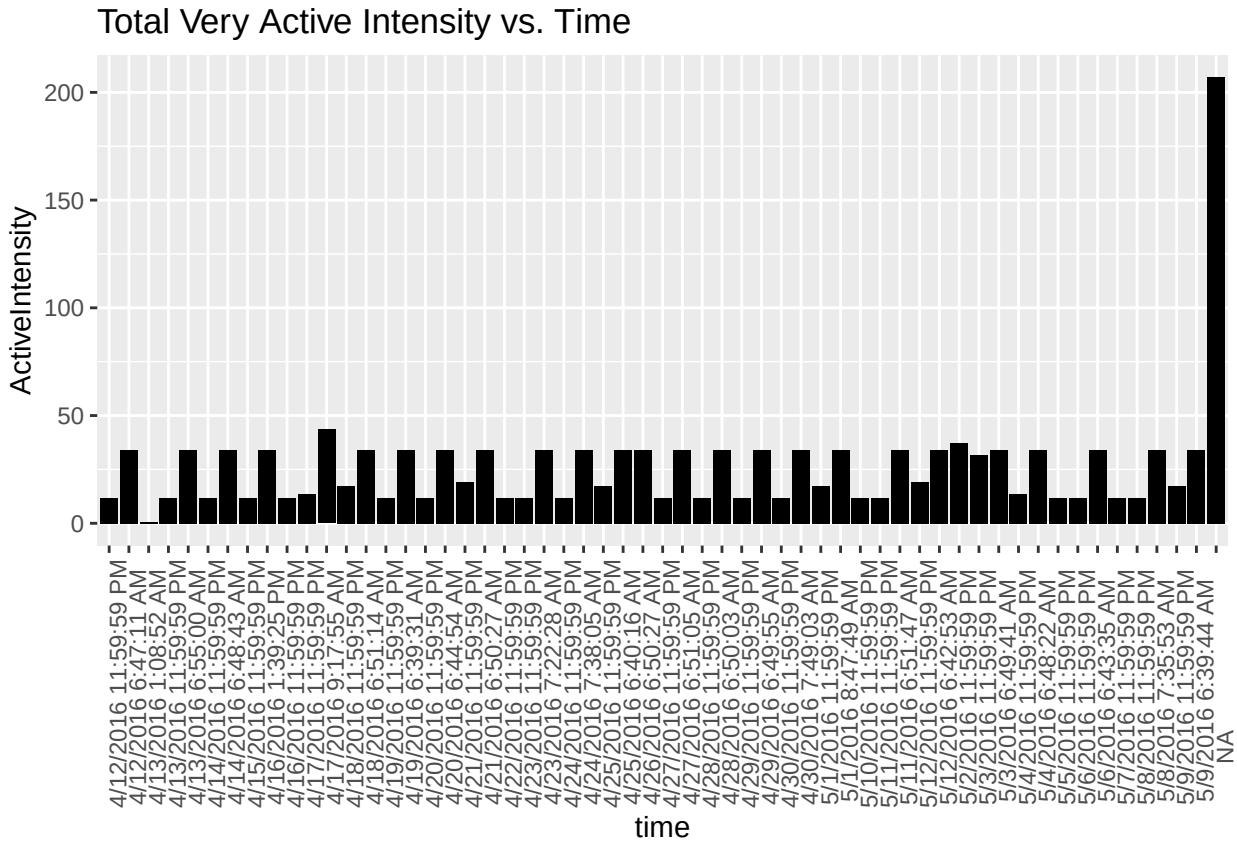
Now, let's shift our focus to examine some data related to activity intensities over a period of time.

```
Intensities$ActiveIntensity <- (Intensities$VeryActiveMinutes)/60

Combined_data <- merge(Weight, Intensities, by="Id", all=TRUE)
Combined_data$time <- format(Combined_data$Date, format = "%H:%M:%S")

ggplot(data=Combined_data, aes(x=time, y=ActiveIntensity)) + geom_histogram(stat = "identity", fill='bla
  theme(axis.text.x = element_text(angle = 90)) +
  labs(title="Total Very Active Intensity vs. Time ")
```

```
## Warning in geom_histogram(stat = "identity", fill = "black"): Ignoring unknown
## parameters: `binwidth`, `bins`, and `pad`
```



Analyzing Intensity Data Over Time

By examining trends in intensity data over time, the company can gain valuable insights into how customers use their products throughout the day. It's evident that many users are active before and after their work hours, presenting an opportunity for the Bellabeat app to send reminders and motivate users to engage in activities like running or walking during these times.

Conclusions and Recommendations for the Business

Empowering Customers with Data

Collecting data on activity, sleep, stress, and more empowers Bellabeat to provide customers with valuable insights into their health and daily routines. Bellabeat is experiencing rapid growth and has positioned itself as a technology-driven wellness company.

Target Audience

Bellabeat should focus its efforts on individuals with full-time jobs who spend a significant amount of time at their desks or in offices and are in need of fitness and daily activity to maintain their health. Users engage in light physical activity to stay healthy, but there is room for improvement in their daily activity levels. Providing knowledge on developing healthy habits and offering motivation can be beneficial.

Message to the Company

The Bellabeat app should aim to be a unique fitness and activity companion, akin to a friendly guide. It should assist users in balancing their personal and professional lives while promoting healthy habits.

Recommendations for the Bellabeat Marketing Team

Here are some recommendations based on the analysis:

1. **Addressing High Sedentary Time:**
   - The data indicates that users of the app have an average sedentary time exceeding 16 hours, which is a concern. Bellabeat should implement a strong marketing strategy targeting segments with high sedentary behavior. Encouraging users to increase daily steps through step tracking and notifications can be an effective approach.
2. **Promoting Better Sleep Habits:**
   - On average, users sleep for 7 hours a day. To help users improve their sleep quality, Bellabeat should consider sending app notifications to encourage bedtime routines. Additionally, the app can recommend reducing sedentary time to enhance sleep patterns.
3. **Encouraging Higher Daily Step Counts:**
   - The average daily step count of 7,638 falls slightly below the CDC's recommended threshold. Bellabeat can educate users about the health benefits of achieving at least 8,000 steps per day, as suggested by CDC research. This can be a compelling way to motivate users to increase their physical activity.
4. **Utilizing Intensity Data:**
   - Analyzing intensity data over time can provide valuable insights into user engagement patterns throughout the day. Notably, many users are active before and after work hours. Leveraging this information, the Bellabeat app can send timely reminders and motivation to encourage users to engage in physical activities like running or walking during these periods.
5. **Supporting Weight Management:**
   - For customers aiming to lose weight, Bellabeat can offer features to track daily calorie consumption. Additionally, the app can provide suggestions for low-calorie, healthy meal options for lunch and dinner to aid in weight management.

Thank you for your interest in my Bellabeat Case Study! Your comments and recommendations for further improvement are highly appreciated. Please take care, and goodbye!