# Bellabeat Capstone

Alaif Naima

2023-08-24

## Bellabeat Case Study

### Table of Contents



### Background

Bellabeat is a company that manufactures high-tech health focused products for women. They provide a variety of smart devices that accumulate data such as activity minutes, steps, heart rate and sleep. The focus of this study is to analyze the data from their products to gain insight into how consumers use their smart devices and learn to better target their marketing strategy.

### Ask

**Business Task** : Identify potential opportunities for growth and provide recommendations for Bellabeat marketing strategy based on trends in use of the smart devices.

**Stakeholders**:

- Urška Sršen - Bellabeat cofounder and Chief Creative Officer
- Sando Mur - Bellabeat cofounder and key member of Bellabeat executive team
- Bellabeat Marketing Analytics team

### Prepare

- *Dataset* : The data source used for our case study is FitBit Fitness Tracker Data. This dataset is stored in Kaggle and was made available through Mobius.
- *Accessibility & Privacy* : The FitBit Fitness Tracker Data is open-source, available in a public domain (Kaggle). FitBit users have consented to submitting their personal tracker data. The users are identified by a unique ID in the data provided to maintain anonymity.

- *About the data* : The FitBit Fitness Tracker Data contains personal fitness tracker data from 30 FitBit Users. Personal tracker data includes: * Minute-level output for physical activity, heart rate, and sleep monitoring. * It also includes information about daily activity, steps, and heart rate that may be used to analyze user habits.
- *Data Organization* : Data is organized as CSV files, into several tables focusing on different aspects of the data tracked by FitBit. It is organized in long format because each user (identified by a unique ID) has data in multiple rows corresponding to different activity dates and times.
  - dailyActivity_merged.csv : daily activity (steps, distance, intensities, calories, activity minutes) over 31 days (33 users)
  - dailyCalories_merged.csv : daily calories over 31 days (33 users)
  - dailyIntensities_merged.csv : daily intensity (in minutes and distance) over 31 days (33 users)
  - dailySteps_merged.csv : daily steps of over 31 days (33 users)
  - hourlyCalories_merged.csv : hourly calories burned over 31 days (33 users)
  - hourlyIntensities_merged.csv : hourly intensity total and average over 31 days
  - hourlySteps_merged.csv : hourly steps over 31 days (33 users)
  - minuteSleep_merged.csv : sleep logged by minute over 31 days (24 users)
  - sleepDay_merged.csv : daily sleep log (count of sleeps per day, total minutes asleep, and total time in bed) (24 users)
  - weightLogInfo_merged.csv : weight tracked by Kg & pounds over 30 days (8 users)
  - heartrate_seconds_merged.csv : day & time log for heartrate (7 users)
  - minuteCaloriesNarrow_merged.csv : calories burned every minute over 31 days (33 users), each row is one minute
  - minuteCaloriesWide_merged.csv : calories burned every minute over 31 days (33 users), each column is one minute
  - minuteIntensitiesNarrow_merged.csv : intensity by minute over 31 days (33 users), each row is one minute
  - minuteIntensitiesWide_merged.csv : intensity by minute over 31 days (33 users), each column is one minute
  - minuteMETsNarrow_merged.csv : ratio of energy used during activity to energy used during rest in minutes
  - minuteStepsNarrow_merged.csv : steps every minute over 31 days (33 users), each row is one minute
  - minuteStepsWide_merged.csv : steps every minute over 31 days (33 users), each column is one minute
- *Data Credibility & Integrity* : A possible limitation to our analysis is the low sample size (30 users). This low sample size could lead to a sampling bias. It is possible that the sample is not representative of the population as a whole.

## Process

Due to the size of the data set, I have chosen to complete my analysis in RStudio. After insuring the data's integrity, steps will be taken to clean the data and verify it.

```
utils::install.packages("tidyverse", repos = "https://mirror.csclub.uwaterloo.ca/CRAN/")
```

**Install the required packages**

```
##
## The downloaded binary packages are in
##  /var/folders/jy/kqj727n12z9b84897rx12skw0000gn/T//RtmpbKWO7w/downloaded_packages
```

```
utils::install.packages("janitor", repos = "https://mirror.csclub.uwaterloo.ca/CRAN/")
```

```
##
## The downloaded binary packages are in
##  /var/folders/jy/kqj727n12z9b84897rx12skw0000gn/T//RtmpbKWO7w/downloaded_packages
```

```
utils::install.packages("here", repos = "https://mirror.csclub.uwaterloo.ca/CRAN/")
```

```
##
## The downloaded binary packages are in
##  /var/folders/jy/kqj727n12z9b84897rx12skw0000gn/T//RtmpbKWO7w/downloaded_packages
```

```
utils::install.packages("ggpubr", repos = "https://mirror.csclub.uwaterloo.ca/CRAN/")
```

```
##
## The downloaded binary packages are in
##  /var/folders/jy/kqj727n12z9b84897rx12skw0000gn/T//RtmpbKWO7w/downloaded_packages
```

```
utils::install.packages("skimr", repos = "https://mirror.csclub.uwaterloo.ca/CRAN/")
```

```
##
## The downloaded binary packages are in
##  /var/folders/jy/kqj727n12z9b84897rx12skw0000gn/T//RtmpbKWO7w/downloaded_packages
```

```
library(tidyverse)
```

**Load packages**

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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)
```

```r
library(readr)
library(stringr)
library(tibble)
library(janitor)
```

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

```r
library(here)
```

```
## here() starts at /Users/alaifnaima
```

```r
library(ggpubr)
library(skimr)
library(ggrepel)
```

```r
library(readr)
daily_activity <- read_csv("Desktop/Fitabase Data 4.12.16-5.12.16/dailyActivity_merged.csv")
```

**Import Datasets (FOCUS: Daily Activity, Daily Sleep, Hourly Calories, Hourly Steps, Hourly Intensities)**

```
## 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
sleep <- read_csv("Desktop/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.
```

```
hourly_calories <- read_csv("Desktop/Fitabase Data 4.12.16-5.12.16/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.
```

```
hourly_steps <- read_csv("Desktop/Fitabase Data 4.12.16-5.12.16/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.
```

```
hourly_intensities <- read_csv("Desktop/Fitabase Data 4.12.16-5.12.16/hourlyIntensities_merged.csv")
```

```
## Rows: 22099 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityHour
## dbl (3): Id, TotalIntensity, AverageIntensity
##
## 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.
```

```
View(daily_activity)
View(sleep)
View(hourly_calories)
View(hourly_steps)
View(hourly_intensities)
```

**Let's take a look at our datasets**

**Cleaning & Formatting**

```
library(dplyr)
library(tidyverse)

sum(duplicated(daily_activity))
```

**Look for duplicates in the datasets**

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

```
sum(duplicated(sleep))
```

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

```
sum(duplicated(hourly_calories))
```

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

```
sum(duplicated(hourly_steps))
```

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

```
sum(duplicated(hourly_intensities))
```

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

```
daily_activity %>% distinct(Id) %>%  drop_na()
```

```
## # A tibble: 33 x 1
##            Id
##         <dbl>
##  1 1503960366
##  2 1624580081
##  3 1644430081
##  4 1844505072
##  5 1927972279
##  6 2022484408
##  7 2026352035
##  8 2320127002
##  9 2347167796
## 10 2873212765
## # i 23 more rows
```

```
sleep %>% distinct(Id) %>%  drop_na()
```

```
## # A tibble: 24 x 1
##            Id
##         <dbl>
##  1 1503960366
##  2 1644430081
##  3 1844505072
##  4 1927972279
##  5 2026352035
##  6 2320127002
##  7 2347167796
##  8 3977333714
##  9 4020332650
## 10 4319703577
## # i 14 more rows
```

```
hourly_calories %>% distinct (Id) %>% drop_na()
```

```
## # A tibble: 33 x 1
##            Id
##         <dbl>
##  1 1503960366
##  2 1624580081
##  3 1644430081
##  4 1844505072
##  5 1927972279
##  6 2022484408
##  7 2026352035
##  8 2320127002
##  9 2347167796
## 10 2873212765
## # i 23 more rows
```

```
hourly_steps %>% distinct (Id) %>% drop_na()
```

```
## # A tibble: 33 x 1
##            Id
##         <dbl>
##  1 1503960366
##  2 1624580081
##  3 1644430081
##  4 1844505072
##  5 1927972279
##  6 2022484408
##  7 2026352035
##  8 2320127002
##  9 2347167796
## 10 2873212765
## # i 23 more rows
```

```
hourly_intensities %>% distinct (Id) %>% drop_na()
```

```
## # A tibble: 33 x 1
##            Id
##         <dbl>
##  1 1503960366
##  2 1624580081
##  3 1644430081
##  4 1844505072
##  5 1927972279
##  6 2022484408
##  7 2026352035
##  8 2320127002
##  9 2347167796
## 10 2873212765
## # i 23 more rows
```

```
daily_sleep <- sleep %>% distinct(Id, .keep_all = TRUE)
```

**Remove duplicates from the sleep data**

```
sum(duplicated(daily_sleep))
```

**Verify that duplicates have been removed from the sleep data**

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

```
library(janitor)

clean_names(daily_activity)
```

**Clean names**

```
## # A tibble: 940 x 15
##            id activity_date total_steps total_distance tracker_distance
##         <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
##  7 1503960366 4/18/2016           13019           8.59             8.59
##  8 1503960366 4/19/2016           15506           9.88             9.88
##  9 1503960366 4/20/2016           10544           6.68             6.68
## 10 1503960366 4/21/2016            9819           6.34             6.34
## # i 930 more rows
## # i 10 more variables: logged_activities_distance <dbl>,
## #   very_active_distance <dbl>, moderately_active_distance <dbl>,
## #   light_active_distance <dbl>, sedentary_active_distance <dbl>,
## #   very_active_minutes <dbl>, fairly_active_minutes <dbl>,
## #   lightly_active_minutes <dbl>, sedentary_minutes <dbl>, calories <dbl>
```

```
clean_names(daily_sleep)
```

```
## # A tibble: 24 x 5
##          id sleep_day total_sleep_records total_minutes_asleep total_time_in_bed
##       <dbl> <chr>                   <dbl>                <dbl>             <dbl>
##  1   1.50e9 4/12/201~                   1                  327               346
##  2   1.64e9 4/29/201~                   1                  119               127
##  3   1.84e9 4/15/201~                   1                  644               961
```

```
##  4   1.93e9 4/12/201~                 3                 750                 775
##  5   2.03e9 4/12/201~                 1                 503                 546
##  6   2.32e9 4/23/201~                 1                  61                  69
##  7   2.35e9 4/13/201~                 1                 467                 531
##  8   3.98e9 4/12/201~                 1                 274                 469
##  9   4.02e9 4/12/201~                 1                 501                 541
## 10   4.32e9 4/14/201~                 1                 535                 557
## # i 14 more rows
```

```
clean_names(hourly_calories)
```

```
## # A tibble: 22,099 x 3
##            id activity_hour         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
##  7 1503960366 4/12/2016 6:00:00 AM        48
##  8 1503960366 4/12/2016 7:00:00 AM        47
##  9 1503960366 4/12/2016 8:00:00 AM        68
## 10 1503960366 4/12/2016 9:00:00 AM       141
## # i 22,089 more rows
```

```
clean_names(hourly_steps)
```

```
## # A tibble: 22,099 x 3
##            id activity_hour         step_total
##         <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
##  7 1503960366 4/12/2016 6:00:00 AM           0
##  8 1503960366 4/12/2016 7:00:00 AM           0
##  9 1503960366 4/12/2016 8:00:00 AM         250
## 10 1503960366 4/12/2016 9:00:00 AM        1864
## # i 22,089 more rows
```

```
clean_names(hourly_intensities)
```

```
## # A tibble: 22,099 x 4
##            id activity_hour         total_intensity average_intensity
##         <dbl> <chr>                          <dbl>             <dbl>
##  1 1503960366 4/12/2016 12:00:00 AM             20             0.333
##  2 1503960366 4/12/2016 1:00:00 AM               8             0.133
##  3 1503960366 4/12/2016 2:00:00 AM               7             0.117
##  4 1503960366 4/12/2016 3:00:00 AM               0             0
```

```
##  5 1503960366 4/12/2016 4:00:00 AM                0             0
##  6 1503960366 4/12/2016 5:00:00 AM                0             0
##  7 1503960366 4/12/2016 6:00:00 AM                0             0
##  8 1503960366 4/12/2016 7:00:00 AM                0             0
##  9 1503960366 4/12/2016 8:00:00 AM               13             0.217
## 10 1503960366 4/12/2016 9:00:00 AM               30             0.5
## # i 22,089 more rows
```

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

daily_sleep$SleepDay=as.POSIXct(daily_sleep$SleepDay, format="%m/%d/%Y", tz=Sys.timezone())
daily_sleep$date <- format(daily_sleep$SleepDay, format = "%m/%d/%y")

hourly_calories$ActivityHour=as.POSIXct(hourly_calories$ActivityHour, format="%m/%d/%Y", tz=Sys.timezone
hourly_calories$date <- format(hourly_calories$ActivityHour, format = "%m/%d/%y")

hourly_steps$ActivityHour=as.POSIXct(hourly_steps$ActivityHour, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.ti
hourly_steps$time <- format(hourly_steps$ActivityHour, format = "%H:%M:%S")
hourly_steps$date <- format(hourly_steps$ActivityHour, format= "%m/%d/%y")

hourly_intensities$ActivityHour=as.POSIXct(hourly_intensities$ActivityHour, format="%m/%d/%Y %I:%M:%S %
hourly_intensities$time <- format(hourly_intensities$ActivityHour, format = "%H:%M:%S")
hourly_intensities$date <- format(hourly_intensities$ActivityHour, format= "%m/%d/%y")
```

**Fix date & time formats**

```
n_distinct(daily_activity$Id)
```

**Verify the number of users for our datasets**

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

```
n_distinct(daily_sleep$Id)
```

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

```
n_distinct(hourly_calories$Id)
```

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

```
n_distinct(hourly_steps$Id)
```

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

```
n_distinct(hourly_intensities$Id)
```

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

We have verified that all our data sets have 33 users, except the sleep data set that has 24 users.

## Analyze & Visualize

We will begin our analysis by getting a summary of statistics from the different data sets.

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

### Daily Activity Statistics

```
##    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
```

From this data set, we can see that: * Mean Steps: 7638 * Mean Distance: 5.4 km * Mean Sedentary Minutes: 991.2 min * Mean Calories burned: 2304

```
daily_activity %>%
  select(VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes) %>%
  summary ()
```

### Activity Minutes per Category Statistics

```
##  VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes
##  Min.   :  0.00    Min.   :  0.00      Min.   :  0.0
##  1st Qu.:  0.00    1st Qu.:  0.00      1st Qu.:127.0
##  Median :  4.00    Median :  6.00      Median :199.0
##  Mean   : 21.16    Mean   : 13.56      Mean   :192.8
##  3rd Qu.: 32.00    3rd Qu.: 19.00      3rd Qu.:264.0
##  Max.   :210.00    Max.   :143.00      Max.   :518.0
```

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

**Calorie Summary Statistics**

```
##     Calories
##  Min.   :   0
##  1st Qu.:1828
##  Median :2134
##  Mean   :2304
##  3rd Qu.:2793
##  Max.   :4900
```

```
daily_sleep %>%
  select(TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed) %>%
  summary()
```

**Sleep Summary Statistics**

```
##  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.000     Min.   : 61.0      Min.   : 69.0
##  1st Qu.:1.000     1st Qu.:313.8      1st Qu.:353.5
##  Median :1.000     Median :427.0      Median :460.5
##  Mean   :1.125     Mean   :390.7      Mean   :432.4
##  3rd Qu.:1.000     3rd Qu.:499.5      3rd Qu.:527.2
##  Max.   :3.000     Max.   :750.0      Max.   :961.0
```

**Merge Data** It would be useful for our analysis to merge the daily activity data with the daily sleep data to look for trends between sleep and activity. The 'Id' and 'date' columns can be used to join the data.

```
merged_data <- merge(daily_sleep, daily_activity, by=c('Id', 'date'))

head(merged_data)
```

```
##           Id     date   SleepDay TotalSleepRecords TotalMinutesAsleep
## 1 1503960366 04/12/16 2016-04-12                 1                327
## 2 1644430081 04/29/16 2016-04-29                 1                119
## 3 1844505072 04/15/16 2016-04-15                 1                644
## 4 1927972279 04/12/16 2016-04-12                 3                750
## 5 2026352035 04/12/16 2016-04-12                 1                503
## 6 2320127002 04/23/16 2016-04-23                 1                 61
##   TotalTimeInBed ActivityDate TotalSteps TotalDistance TrackerDistance
## 1            346   2016-04-12      13162          8.50            8.50
## 2            127   2016-04-29       3176          2.31            2.31
## 3            961   2016-04-15       3844          2.54            2.54
## 4            775   2016-04-12        678          0.47            0.47
```

```
## 5            546   2016-04-12       4414          2.74           2.74
## 6             69   2016-04-23       5079          3.42           3.42
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0              1.88                     0.55
## 2                        0              0.00                     0.00
## 3                        0              0.00                     0.00
## 4                        0              0.00                     0.00
## 5                        0              0.19                     0.35
## 6                        0              0.00                     0.00
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                2.31                       0                 0
## 3                2.54                       0                 0
## 4                0.47                       0                 0
## 5                2.20                       0                 3
## 6                3.42                       0                 0
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  13                  328              728     1985
## 2                   0                  120             1193     2498
## 3                   0                  176              527     1725
## 4                   0                   55              734     2220
## 5                   8                  181              706     1459
## 6                   0                  242             1129     1804
```

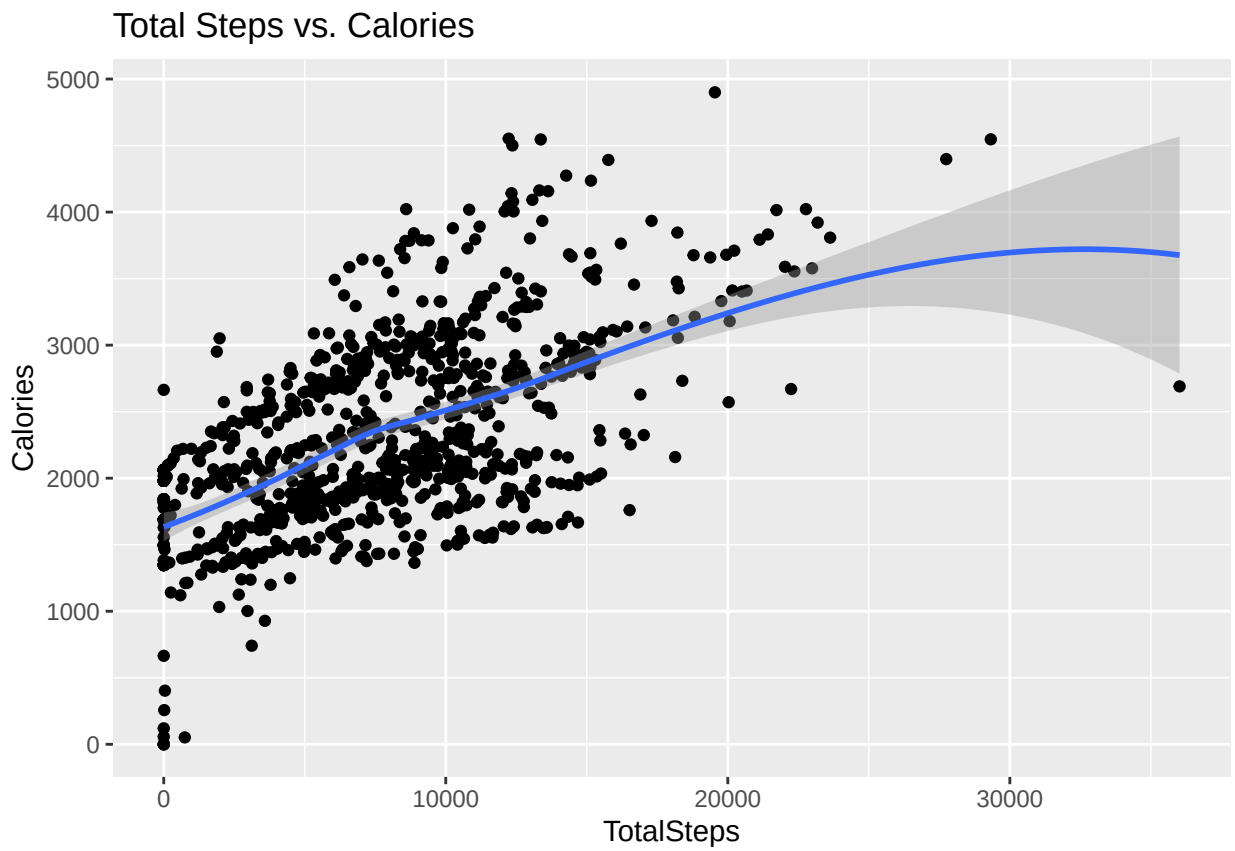
### Visualization of Data

```
library(ggplot2)
```

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

### Steps vs Calories

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

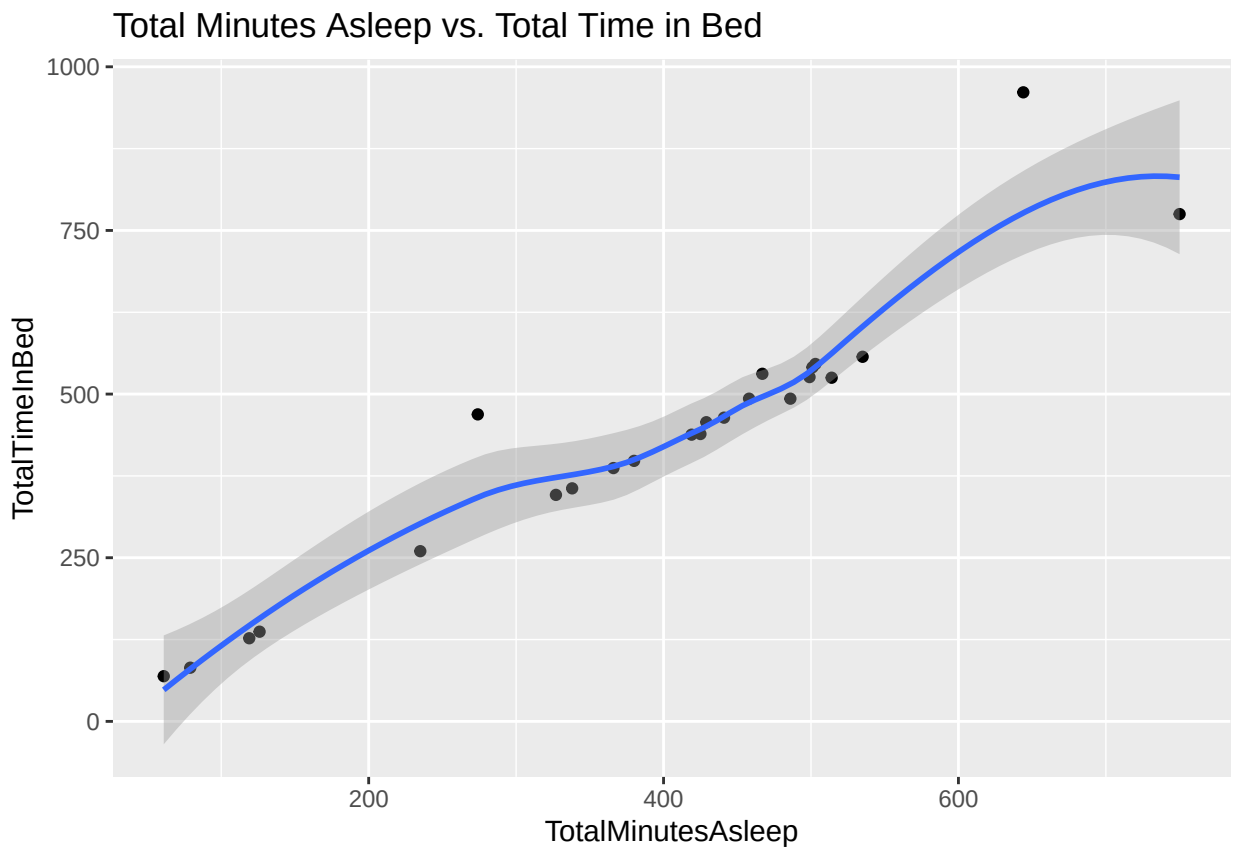



Trend: Calories increase as steps increase

```
ggplot(data=daily_sleep, mapping=aes(x=TotalMinutesAsleep, y=TotalTimeInBed))+
  geom_point()+ geom_smooth() + labs(title="Total Minutes Asleep vs. Total Time in Bed")
```

**Total Minutes Asleep vs Total Time in Bed**

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



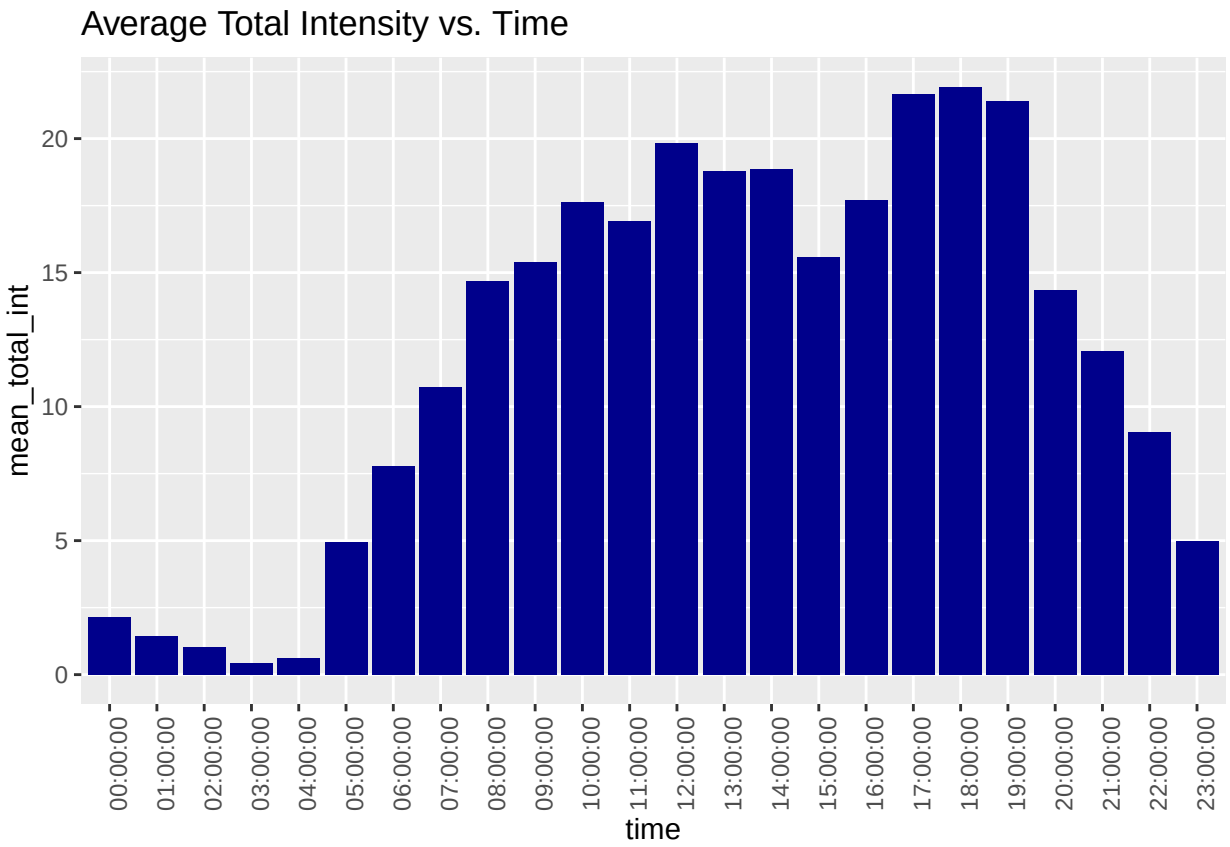

Trend: Total minutes asleep increase as total time in bed increase

```
int_new <- hourly_intensities %>%
  group_by(time) %>%
  drop_na() %>%
  summarise(mean_total_int=mean(TotalIntensity))

ggplot(data=int_new, aes(x=time, y=mean_total_int)) +
  geom_histogram(stat = "identity", fill='darkblue')+
  theme(axis.text.x = element_text(angle = 90))+
  labs(title="Average Total Intensity vs. Time")
```

**Intensity vs Time**

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

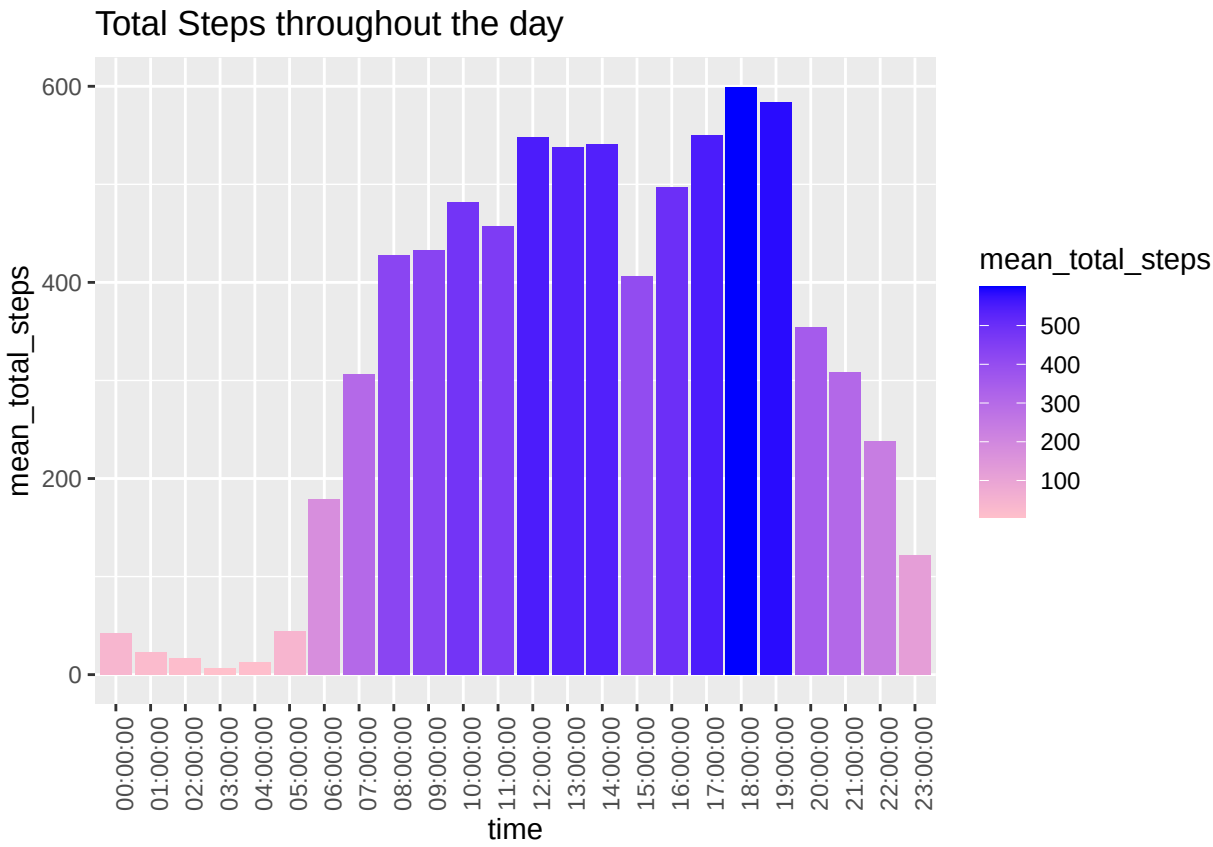



Trend: People are at their highest activity intensities between 8am-8pm

```
View(hourly_steps)

steps_new <- hourly_steps %>%
  group_by(time) %>%
  drop_na() %>%
  summarise(mean_total_steps=mean(StepTotal))

ggplot(data=steps_new)+ geom_col(mapping=aes(x=time, y=mean_total_steps, fill=mean_total_steps))+
  scale_fill_gradient(low='pink', high='blue')+
  theme(axis.text.x = element_text(angle = 90))+
  labs(title="Total Steps throughout the day")
```



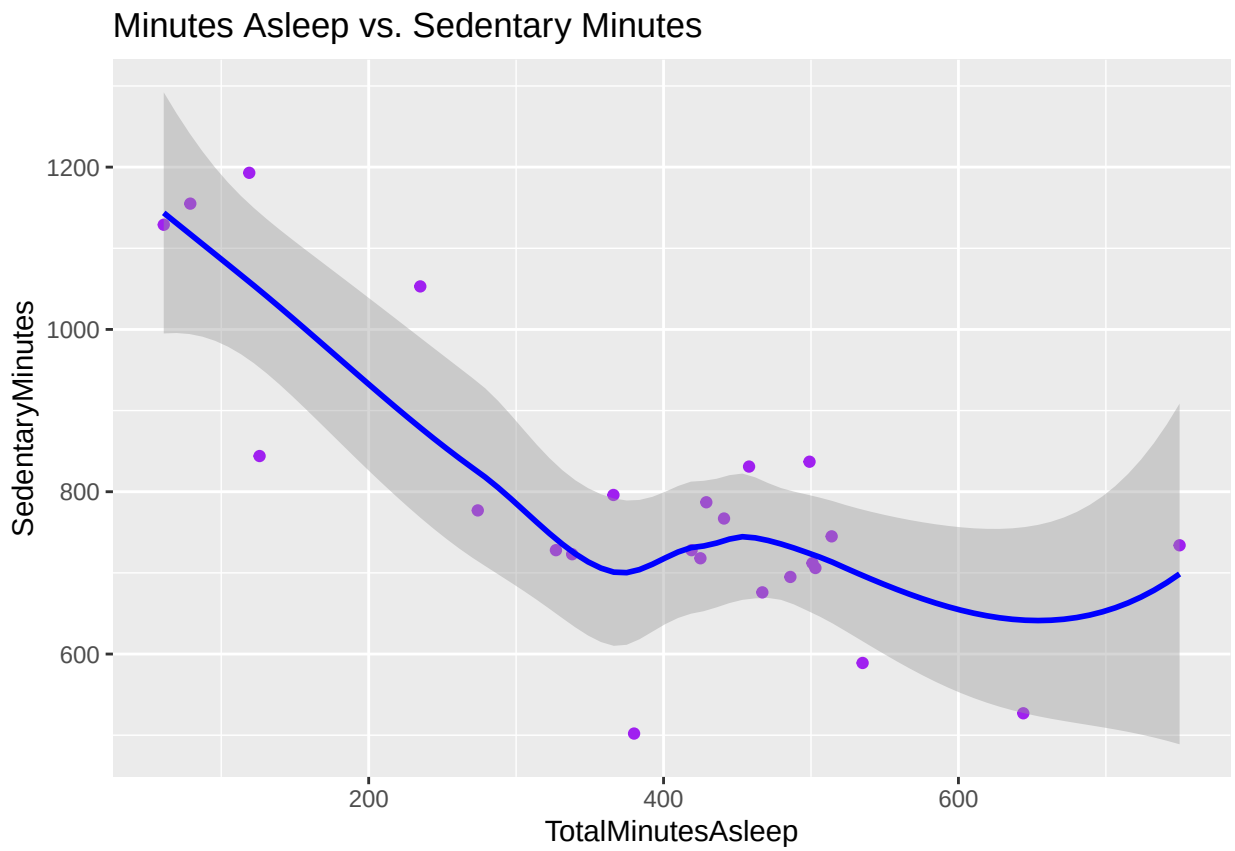

**Steps vs Time**

Trend: Most steps are taken from 8am-8pm

```
ggplot(data=merged_data, aes(x=TotalMinutesAsleep, y=SedentaryMinutes))+
  geom_point(color='purple')+ geom_smooth(color='blue')+
  labs(title="Minutes Asleep vs. Sedentary Minutes")
```

**Total Minutes Asleep vs Sedentary Minutes**

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




Trend: As total minutes asleep increases, total sedentary minutes decreases. Therefore, as you sleep better, you are more active or if you are more active, you sleep better.

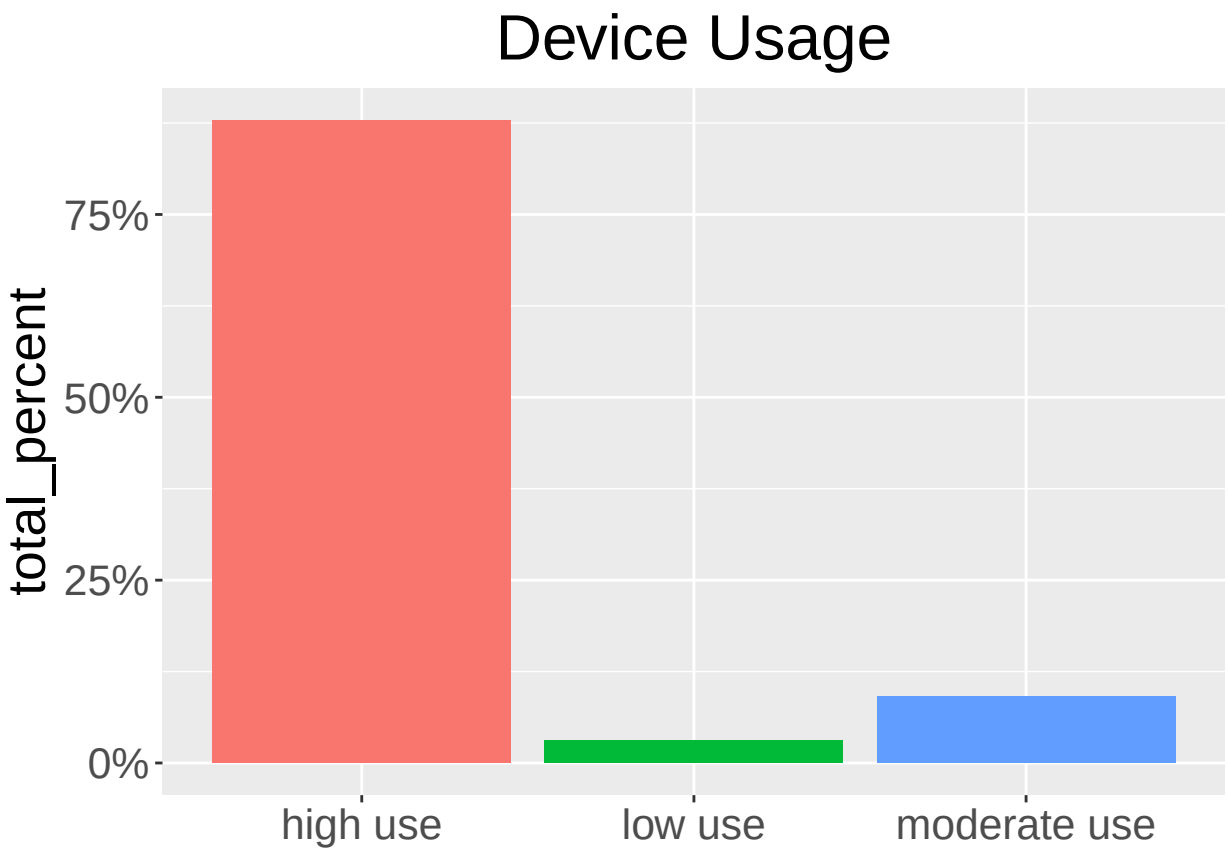
**Smart Device Usage** We can look at how often users wore their trackers by categorizing them into "low use", "moderate use", and "high use":

```
daily_use <- daily_activity %>%
  group_by(Id) %>%
  drop_na() %>%
  summarise(days_used=sum(n())) %>%
  mutate(usage = case_when(
    days_used >= 1 & days_used <= 10 ~ "low use",
    days_used >= 11 & days_used <= 20 ~ "moderate use",
    days_used >= 21 & days_used <= 31 ~ "high use",
  ))

View(daily_use)

daily_use %>%
  group_by(usage) %>%
  summarise(total=n()) %>%
  mutate(totals=sum(total)) %>%
  group_by(usage) %>%
  summarise(total_percent=total/totals) %>%
  ggplot(daily_use, mapping=aes(usage, y=total_percent, fill=usage))+
  geom_col()+scale_y_continuous(labels=scales::percent)+
```

```
  theme(legend.position="none")+
  labs(title="Device Usage", x=NULL)+
  theme(legend.position="none", text= element_text(size=20), plot.title=element_text(hjust=0.5))
```




Trend: Majority of tracker usage is categorized as "high use". So many of the users wore the tracker for 21-31 days.

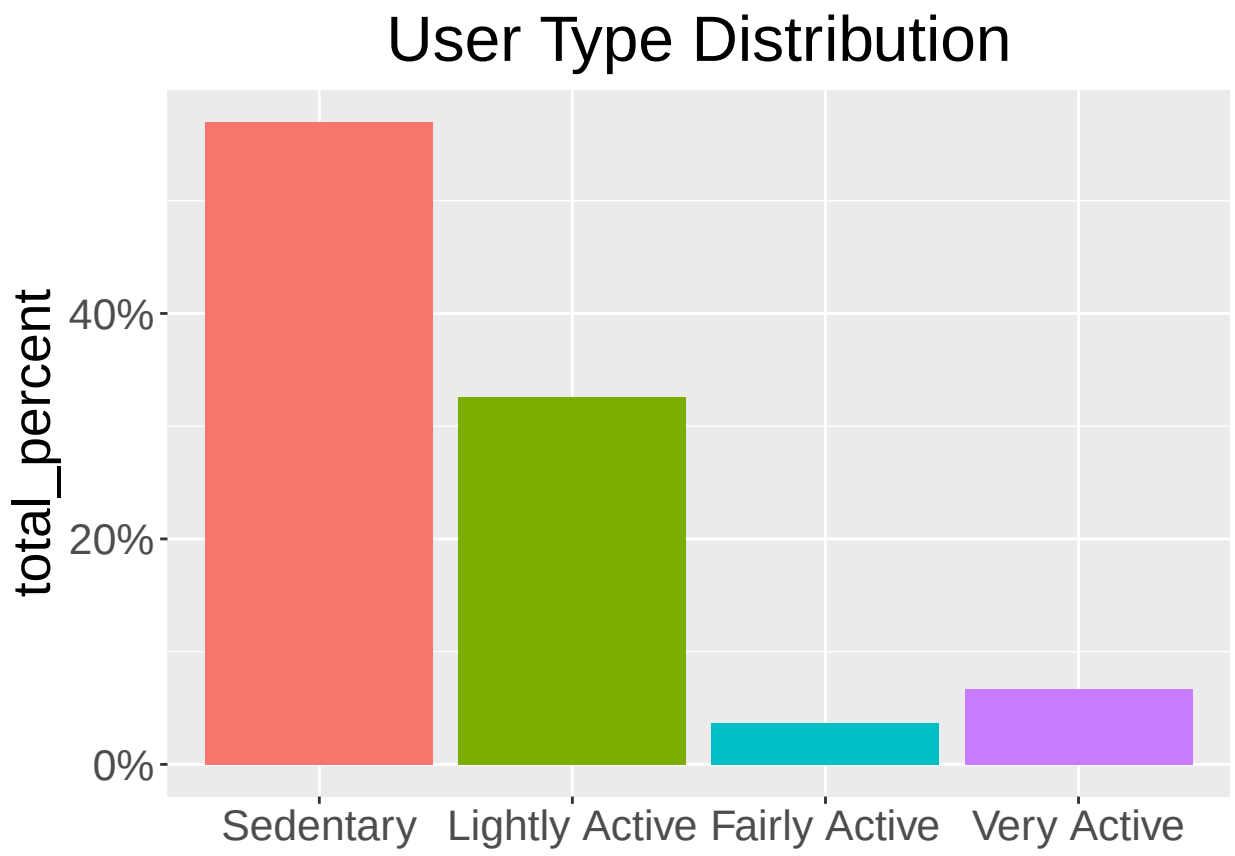
```
data_by_usertype <- daily_activity %>%
  reframe(
    user_type = factor(case_when(
      SedentaryMinutes > mean(SedentaryMinutes) & LightlyActiveMinutes < mean(LightlyActiveMinutes) & Fa
      SedentaryMinutes < mean(SedentaryMinutes) & LightlyActiveMinutes > mean(LightlyActiveMinutes) & Fa
      SedentaryMinutes < mean(SedentaryMinutes) & LightlyActiveMinutes < mean(LightlyActiveMinutes) & Fa
      SedentaryMinutes < mean(SedentaryMinutes) & LightlyActiveMinutes < mean(LightlyActiveMinutes) & Fa
    ),levels=c("Sedentary", "Lightly Active", "Fairly Active", "Very Active")), Calories, .group=Id) %>%
  drop_na()
```

**Understanding Users based on Activity Levels** We have now created a new table based on User Type.

```
View(data_by_usertype)
```

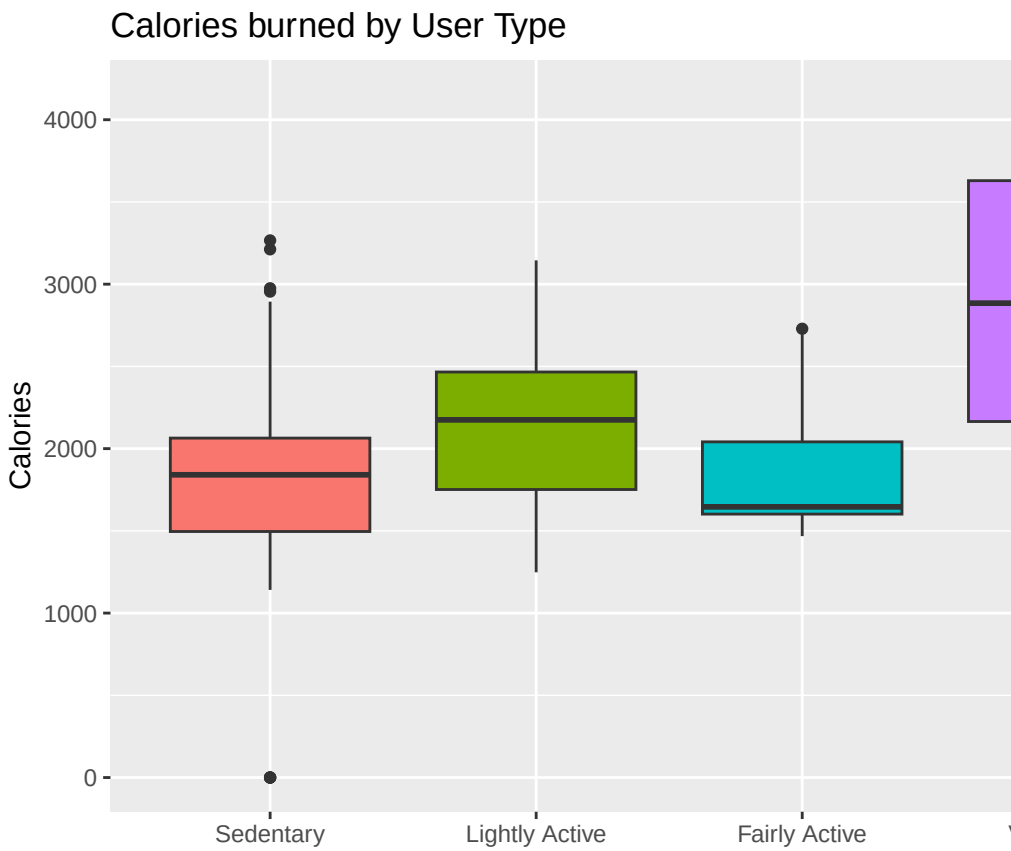
Let us take a look at the distribution of user types:

```r
data_by_usertype %>%
  group_by(user_type) %>%
  summarise(total= n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(user_type) %>%
  summarise(total_percent = total/totals) %>%
  ggplot(data_by_usertype, mapping=aes(user_type, y=total_percent, fill=user_type))+
  geom_col() + scale_y_continuous(labels = scales::percent) +
  theme(legend.position ="none")+
  labs(title="User Type Distribution", x=NULL)+
  theme(legend.position="none", text = element_text(size = 20), plot.title = element_text(hjust=0.5))
```




Trend: Most users fall into the Sedentary and Lightly Active Categories

```r
ggplot(data_by_usertype, mapping=aes(user_type, Calories, fill=user_type))+
  geom_boxplot()+
  theme(legend.position="none")+
  labs(title="Calories burned by User Type", x=NULL)
```




**Calories Burned by User Type**

Trend: Most calories burned by the very active users

**Using Total Steps to Categorize Users** First we will need to find out the daily averages of the users:

```
daily_avg <- merged_data %>%
  group_by(Id) %>%
  summarise(mean_daily_steps = mean(TotalSteps), mean_daily_calories= mean(Calories), mean_daily_sleep=

head(daily_avg)
```

```
## # A tibble: 6 x 4
##           Id mean_daily_steps mean_daily_calories mean_daily_sleep
##        <dbl>            <dbl>               <dbl>            <dbl>
## 1 1503960366            13162                1985              327
## 2 1644430081             3176                2498              119
## 3 1844505072             3844                1725              644
## 4 1927972279              678                2220              750
## 5 2026352035             4414                1459              503
## 6 2320127002             5079                1804               61
```

Now we will classify the users by their average daily steps:

```
usertype_bysteps <- daily_avg %>%
  mutate(usertype_bysteps = case_when(mean_daily_steps < 5000 ~ "sedentary", mean_daily_steps >= 5000 &
```

```r
head(usertype_bysteps)
```

```
## # A tibble: 6 x 5
##        Id mean_daily_steps mean_daily_calories mean_daily_sleep usertype_bysteps
##     <dbl>            <dbl>               <dbl>            <dbl> <chr>
## 1  1.50e9            13162                1985              327 very active
## 2  1.64e9             3176                2498              119 sedentary
## 3  1.84e9             3844                1725              644 sedentary
## 4  1.93e9              678                2220              750 sedentary
## 5  2.03e9             4414                1459              503 sedentary
## 6  2.32e9             5079                1804               61 lightly active
```

```r
View(usertype_bysteps)
```

Time to figure out the percent of user type based on steps:

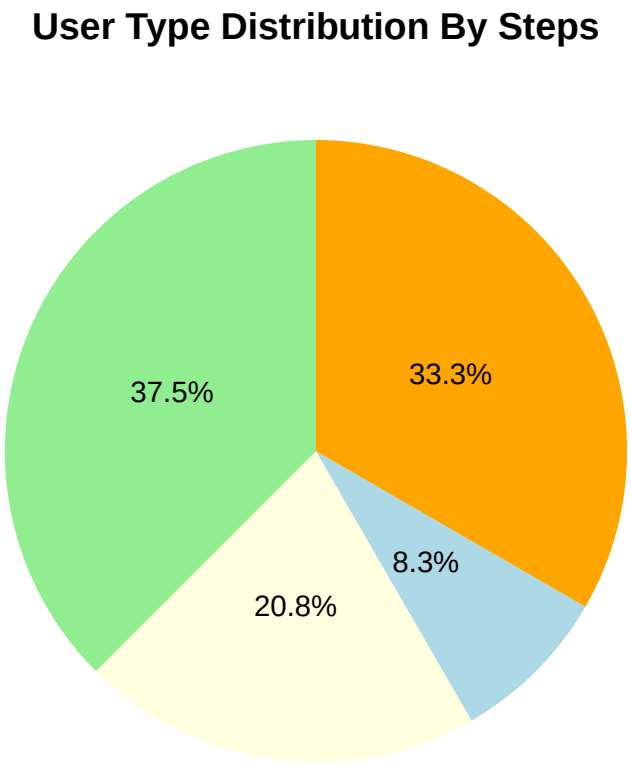
```r
usertype_step_percent <- usertype_bysteps %>%
  group_by(usertype_bysteps) %>%
  summarise(total=n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(usertype_bysteps) %>%
  summarise(total_percent= total/totals) %>%
  mutate(labels= scales::percent(total_percent))

usertype_step_percent$usertype_bysteps <- factor(usertype_step_percent$usertype_bysteps, levels = c("ve

head(usertype_step_percent)
```

```
## # A tibble: 4 x 3
##   usertype_bysteps total_percent labels
##   <fct>                    <dbl> <chr>
## 1 fairly active           0.208  20.8%
## 2 lightly active          0.0833 8.3%
## 3 sedentary               0.333  33.3%
## 4 very active             0.375  37.5%
```

```r
usertype_step_percent %>%
  ggplot(aes(x="", y=total_percent, fill=usertype_bysteps)) +
  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"))+
  scale_fill_manual(values= c("lightgreen", "lightyellow", "lightblue", "orange")) +
  geom_text(aes(label= labels), position=position_stack(vjust = 0.5)) + labs(title="User Type Distributi
```




**Visualization of User Type By Steps**

Trend: Based on average daily steps, users as mainly sedentary or very active.

So, categorizing user by steps and by total active minutes give us different stories. However, both categorizations indicate that a big chunk of users are sedentary.

## Conclusion

By providing us with our fitness data, Bellabeat can give us the opportunity to empower ourselves by improving our health.

How our insights can be applied:

- Steps/Sleep/Activity Notification

  Can encourage user step count by sending notifications to remind them to get up and take a walk throughout the day. Increased steps was shown to be correlated with increased calories burned

  Allow them to set alarm and a notification sent to help them to get to bed on time so they are well rested. People who sleep a healthy amount (approx. 8 hours), are less sedentary.

  Increased activity minutes was correlated with increased calories burned.

- Challenges and Competitions to encourage users to get more active:

  * Ex. Challenge: getting to 10,000 steps in a day

  * Ex. Competition : between friends, co-workers, family to increase activity minutes and distance

  As our analysis showed that a big percentage of tracker users are sedentary, we need to look for ways to increase their motivation to get active and get healthy.

## Recommendations

According to the analysis, it would be most strategic to focus marketing on individuals who are interested in improving their overall health, achieving weight loss, and getting better quality sleep. In future case studies, it may be possible to use Bellabeat's own data, as well as FitBit data from a bigger sample size to get a more thorough analysis.