Case Study Bellabeat

Elizabeth Hoang

2023-06-25

For this case study I'm a Junior Marketing Analyst for Bellabeat. Our business task is to find how the current trends for smart device usage will help our Bellabeat Marketing team's strategy and apply these trends to better target our products to Bellabeat's customers, in this case focusing on the Bellabeat App and trends we can use to improve it for users. We will then present our findings to the Marketing team and executive team which include the co-founders Urska Srsen and Sando Mur. These are the packages that I will need to download in order to clean, sort and analyze the data-sets.

Step 1: Load Packages

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

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

```
install.packages("readr")
```

```
## 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("dplyr")
```

```
## 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)
```

```
library("tidyverse")
```

```
## -- 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 error
```

```
library("readr")
library("tidyr")
library("dplyr")
library("skimr")
library("janitor")
```

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

The Fitbit data was retrieved from Kaggle a public database, the data set "FitBit Fitness Tracker Data" was created by Robert Furber, Julia Brinton, Michael Keating, and Alexa Ortiz and made available on Kaggle by Mobius. is a data-set collected of 30 participants who responded via a survey on Amazon Mechanical Turk from March 12, 2016 to May 12, 2016. It contains information that tracks data from physical activity, heart rate, and sleep monitoring. The files are in a downloaded zip folder with 18 CSV. files each including different categories of calories, daily activity, heart rate, steps and sleep measured by minutes and hour; including a data=set with weight changes and updates. Some files are long format and other files are wide format.

ROCCC- Validating Data

Reliability of the data provided is not reliable since there are only 30 participants.

Originality would be questionable since it was done via survey though Amazon Mechanical Turk, there could be bias or duplicate enters, or participants answers might not be accurate.

Comprehensiveness is good since most of the data sets are identical to Bellabeat's products and measurements each track for users, making it easy to understand,

Data is not current or up to date it is data focused specifically from March 12, 2016 to May 12, 2016; today's smart device usage trends are different than 2016.

Data is cited correctly with the information of the creators/authors of the database. The data base is cited as (Furberg, R., Brinton, J., Keating, M., & Ortiz, A. (2016). Crowd-sourced Fitbit datasets 03.12.2016-05.12.2016 [Data set]. Zenodo. https://doi.org/10.5281/zenodo.53894).

There's information on 30 participants that are FitBit users, it's only a small portion of the population, there could be some bias or limitations since it's not a good sample of the whole fitness population. The dataset provided is licensed by the Creative Commons Attriubtion 4.0 International Public License, making it public and accessible to anyone.

For these data sets we are only downloading three tables: dailyActivity_merged.csv sleepDay_merged.csv weightLogInfo_merged.csv

The other data-sets contain data that is already recorded in one of these three data-sets.

Step 2: Importing Data

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

```
## Rows: 940 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): ActivityDate
## dbl (14): Id, TotalSteps, TotalDistance, TrackerDistance, LoggedActivitiesDi...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
daily_sleep<- read_csv("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.
```

```
weight<- read_csv("weightLogInfo_merged.csv")
```

```
## Rows: 67 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Date
## dbl (6): Id, WeightKg, WeightPounds, Fat, BMI, LogId
## lgl (1): IsManualReport
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Step 3: Understanding our Data

Now we will take a look at each data-set to check for any errors that that the data was imported correctly.

```
head(activity_daily)
```

```
## # 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>
```

```
head(daily_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
```

```
head(weight)
```

```
## # A tibble: 6 x 8
##           Id Date       WeightKg WeightPounds   Fat   BMI IsManualReport   LogId
##        <dbl> <chr>         <dbl>        <dbl> <dbl> <dbl> <lgl>            <dbl>
## 1 1503960366 5/2/2016 ~     52.6         116.    22  22.6 TRUE           1.46e12
## 2 1503960366 5/3/2016 ~     52.6         116.    NA  22.6 TRUE           1.46e12
## 3 1927972279 4/13/2016~    134.          294.    NA  47.5 FALSE          1.46e12
## 4 2873212765 4/21/2016~     56.7         125.    NA  21.5 TRUE           1.46e12
## 5 2873212765 5/12/2016~     57.3         126.    NA  21.7 TRUE           1.46e12
## 6 4319703577 4/17/2016~     72.4         160.    25  27.5 TRUE           1.46e12
```

We'll run the str() function to take a quick look at the data frames for each data-set.

```
str(activity_daily)
```

```
## spc_tbl_ [940 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id                       : num [1:940] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate             : chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps               : num [1:940] 13162 10735 10460 9762 12669 ...
##  $ TotalDistance            : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance          : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance       : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance      : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes        : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes      : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes     : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes         : num [1:940] 728 776 1218 726 773 ...
##  $ Calories                 : num [1:940] 1985 1797 1776 1745 1863 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDate = col_character(),
##   ..   TotalSteps = col_double(),
##   ..   TotalDistance = col_double(),
##   ..   TrackerDistance = col_double(),
##   ..   LoggedActivitiesDistance = col_double(),
##   ..   VeryActiveDistance = col_double(),
##   ..   ModeratelyActiveDistance = col_double(),
##   ..   LightActiveDistance = col_double(),
##   ..   SedentaryActiveDistance = col_double(),
##   ..   VeryActiveMinutes = col_double(),
##   ..   FairlyActiveMinutes = col_double(),
##   ..   LightlyActiveMinutes = col_double(),
##   ..   SedentaryMinutes = col_double(),
```

```
##   ..   Calories = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(daily_sleep)
```

```
## spc_tbl_ [413 x 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id                : num [1:413] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : chr [1:413] "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:
##  $ TotalSleepRecords : num [1:413] 1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: num [1:413] 327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : num [1:413] 346 407 442 367 712 320 377 364 384 449 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   SleepDay = col_character(),
##   ..   TotalSleepRecords = col_double(),
##   ..   TotalMinutesAsleep = col_double(),
##   ..   TotalTimeInBed = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(weight)
```

```
## spc_tbl_ [67 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id            : num [1:67] 1.50e+09 1.50e+09 1.93e+09 2.87e+09 2.87e+09 ...
##  $ Date          : chr [1:67] "5/2/2016 11:59:59 PM" "5/3/2016 11:59:59 PM" "4/13/2016 1:08:52 AM" "
##  $ WeightKg      : num [1:67] 52.6 52.6 133.5 56.7 57.3 ...
##  $ WeightPounds  : num [1:67] 116 116 294 125 126 ...
##  $ Fat           : num [1:67] 22 NA NA NA NA 25 NA NA NA NA ...
##  $ BMI           : num [1:67] 22.6 22.6 47.5 21.5 21.7 ...
##  $ IsManualReport: logi [1:67] TRUE TRUE FALSE TRUE TRUE TRUE ...
##  $ LogId         : num [1:67] 1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   Date = col_character(),
##   ..   WeightKg = col_double(),
##   ..   WeightPounds = col_double(),
##   ..   Fat = col_double(),
##   ..   BMI = col_double(),
##   ..   IsManualReport = col_logical(),
##   ..   LogId = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Now the column names

```
colnames(activity_daily)
```

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

```
## [15] "Calories"
```

The Activity table is made up of 15 columns, two of them being the data for Total Steps and Calories, which are contained on two different CSV. files that we don't have to upload, since the information is also copied onto this data-set.

```
colnames(daily_sleep)
```

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

The Daily Sleep table is made of 5 columns containing the ID of each participant, the day, total sleep records, total minutes asleep, and total time spend in bed in minute form.

```
colnames(weight)
```

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

The Weight table, contains information of each participants ID, weight in Kg, fat, BMI, weight in pounds

Now we will view the data-sets to see how each one looks and some data we can analyze at first glance.

```
view(activity_daily)
view(daily_sleep)
view(weight)
```

At first glance, there a values with zeros for columns like Total Steps and Total Distance. This can mean that there are days when participants did not recorded or logged their steps and activity. This can cause the data to be skewed and not accurate.

Step 4: Analyzing some of the data

Now we will run a clean name function to make sure there are only number, letters and underscores in the names of each column for each table.

```
clean_names(activity_daily)
```

```
## # 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: 413 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   1.50e9 4/13/201~                   2                  384                407
##  3   1.50e9 4/15/201~                   1                  412                442
##  4   1.50e9 4/16/201~                   2                  340                367
##  5   1.50e9 4/17/201~                   1                  700                712
##  6   1.50e9 4/19/201~                   1                  304                320
##  7   1.50e9 4/20/201~                   1                  360                377
##  8   1.50e9 4/21/201~                   1                  325                364
##  9   1.50e9 4/23/201~                   1                  361                384
## 10   1.50e9 4/24/201~                   1                  430                449
## # i 403 more rows
```

```
clean_names(weight)
```

```
## # A tibble: 67 x 8
##            id date  weight_kg weight_pounds   fat   bmi is_manual_report  log_id
##         <dbl> <chr>     <dbl>         <dbl> <dbl> <dbl> <lgl>              <dbl>
##  1 1503960366 5/2/~      52.6          116.    22  22.6 TRUE             1.46e12
##  2 1503960366 5/3/~      52.6          116.    NA  22.6 TRUE             1.46e12
##  3 1927972279 4/13~     134.           294.    NA  47.5 FALSE            1.46e12
##  4 2873212765 4/21~      56.7          125.    NA  21.5 TRUE             1.46e12
##  5 2873212765 5/12~      57.3          126.    NA  21.7 TRUE             1.46e12
##  6 4319703577 4/17~      72.4          160.    25  27.5 TRUE             1.46e12
##  7 4319703577 5/4/~      72.3          159.    NA  27.4 TRUE             1.46e12
##  8 4558609924 4/18~      69.7          154.    NA  27.2 TRUE             1.46e12
##  9 4558609924 4/25~      70.3          155.    NA  27.5 TRUE             1.46e12
## 10 4558609924 5/1/~      69.9          154.    NA  27.3 TRUE             1.46e12
## # i 57 more rows
```

We will take a look at the Total Steps column to see how the steps are recorded and in order to see if the zeros will be consistent and an issue for our analysis. First we will arrange it by ascending order of total steps. Then we will do a descending order of total steps to see the difference.

```
activity_daily %>%  arrange(TotalSteps)
```

```
## # A tibble: 940 x 15
##            Id ActivityDate TotalSteps TotalDistance TrackerDistance
##         <dbl> <chr>              <dbl>         <dbl>           <dbl>
##  1 1503960366 5/12/2016              0             0               0
##  2 1844505072 4/24/2016              0             0               0
##  3 1844505072 4/25/2016              0             0               0
##  4 1844505072 4/26/2016              0             0               0
##  5 1844505072 5/2/2016               0             0               0
##  6 1844505072 5/7/2016               0             0               0
##  7 1844505072 5/8/2016               0             0               0
##  8 1844505072 5/9/2016               0             0               0
##  9 1844505072 5/10/2016              0             0               0
## 10 1844505072 5/11/2016              0             0               0
## # i 930 more rows
## # i 10 more variables: LoggedActivitiesDistance <dbl>,
## #   VeryActiveDistance <dbl>, ModeratelyActiveDistance <dbl>,
```

```
## #   LightActiveDistance <dbl>, SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <dbl>, FairlyActiveMinutes <dbl>,
## #   LightlyActiveMinutes <dbl>, SedentaryMinutes <dbl>, Calories <dbl>
```

```
activity_daily %>% arrange(-TotalSteps)
```

```
## # A tibble: 940 x 15
##            Id ActivityDate TotalSteps TotalDistance TrackerDistance
##         <dbl> <chr>             <dbl>         <dbl>           <dbl>
##  1 1624580081 5/1/2016          36019          28.0            28.0
##  2 8877689391 4/16/2016         29326          25.3            25.3
##  3 8877689391 4/30/2016         27745          26.7            26.7
##  4 8877689391 4/27/2016         23629          20.6            20.6
##  5 8877689391 4/12/2016         23186          20.4            20.4
##  6 8053475328 4/24/2016         22988          18.0            18.0
##  7 4388161847 5/7/2016          22770          17.5            17.5
##  8 8053475328 4/23/2016         22359          17.2            17.2
##  9 2347167796 4/16/2016         22244          15.1            15.1
## 10 8053475328 5/8/2016          22026          17.6            17.6
## # i 930 more rows
## # i 10 more variables: LoggedActivitiesDistance <dbl>,
## #   VeryActiveDistance <dbl>, ModeratelyActiveDistance <dbl>,
## #   LightActiveDistance <dbl>, SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <dbl>, FairlyActiveMinutes <dbl>,
## #   LightlyActiveMinutes <dbl>, SedentaryMinutes <dbl>, Calories <dbl>
```

Running these functions, we can see the smallest number of steps tracked are 0 in ascending order, then when we run a function to descend the number of steps we can see max number of steps taken by a participants was 36,019. There are to count so we need to get rid of all of the zeros to get an accurate analysis.

I will create data frames for each one, so it'll be easier to view later on in the data analysis.

```
stepsasc <- activity_daily %>%  arrange(TotalSteps)
stepsdes <- activity_daily %>% arrange(-TotalSteps)
view(stepsasc)
view(stepsdes)
```

We will run a function to get summarized information about of participants and the average steps they took.When we run this we get a data frame of all of the participants and the average mean steps they took across every day.

```
activity_daily %>% group_by(Id) %>% drop_na() %>% summarize(mean_totalsteps=mean(TotalSteps))
```

```
## # A tibble: 33 x 2
##            Id mean_totalsteps
##         <dbl>           <dbl>
##  1 1503960366          12117.
##  2 1624580081           5744.
##  3 1644430081           7283.
##  4 1844505072           2580.
##  5 1927972279            916.
##  6 2022484408          11371.
##  7 2026352035           5567.
##  8 2320127002           4717.
##  9 2347167796           9520.
## 10 2873212765           7556.
## # i 23 more rows
```

When we run this we get a data frame of all of the participants and the average mean each individual took across every day.

```
activity_daily %>% group_by(Id) %>% drop_na() %>% summarize(max_totalsteps=max(TotalSteps))
```

```
## # A tibble: 33 x 2
##            Id max_totalsteps
##         <dbl>          <dbl>
##  1 1503960366          18134
##  2 1624580081          36019
##  3 1644430081          18213
##  4 1844505072           8054
##  5 1927972279           3790
##  6 2022484408          18387
##  7 2026352035          12357
##  8 2320127002          10725
##  9 2347167796          22244
## 10 2873212765           9685
## # i 23 more rows
```

We ran a function to see the highest amount of steps taken by a participants, which we see was done by ID(1624580081) for the amount of 36019 steps.

Next we'll check if there's any correlation between Total Steps taken and calories with the participants.

```
activity_daily %>%  group_by(Id, TotalSteps) %>% drop_na() %>% summarize(mean_calories=mean(Calories))
```

```
## `summarise()` has grouped output by 'Id'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 876 x 3
## # Groups:   Id [33]
##            Id TotalSteps mean_calories
##         <dbl>      <dbl>         <dbl>
##  1 1503960366          0             0
##  2 1503960366       9705          1728
##  3 1503960366       9762          1745
##  4 1503960366       9819          1775
##  5 1503960366      10039          1788
##  6 1503960366      10060          1740
##  7 1503960366      10460          1776
##  8 1503960366      10544          1786
##  9 1503960366      10602          1820
## 10 1503960366      10735          1797
## # i 866 more rows
```

```
activity_daily %>%  group_by(Id, TotalSteps) %>% drop_na() %>%
summarize(mean_calories=mean(Calories), max_calories=max(Calories))
```

```
## `summarise()` has grouped output by 'Id'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 876 x 4
## # Groups:   Id [33]
##            Id TotalSteps mean_calories max_calories
##         <dbl>      <dbl>         <dbl>        <dbl>
##  1 1503960366          0             0            0
##  2 1503960366       9705          1728         1728
```

```
##  3 1503960366       9762          1745          1745
##  4 1503960366       9819          1775          1775
##  5 1503960366      10039          1788          1788
##  6 1503960366      10060          1740          1740
##  7 1503960366      10460          1776          1776
##  8 1503960366      10544          1786          1786
##  9 1503960366      10602          1820          1820
## 10 1503960366      10735          1797          1797
## # i 866 more rows
```

I kept getting an errors every time I ran this function. This information can be wrong so I will remove all the zeros now and work with a new set of dataframes made for each data-set.

Step 5: Cleaning our Data

```
activity_daily2<-activity_daily %>% filter(TotalSteps !=0)
view(activity_daily2)
```

By creating a new data frame for Activity Daily, are data is much cleaner and easier to make analysis from it.

Next we see that the weight and daily sleep tables contain dates and time in the same column, we need to separate these in order to be able to have a clear reading of the table.

```
daily_sleep2<-daily_sleep %>%
  separate(SleepDay, c("Date","Time"), " ")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 413 rows [1, 2, 3, 4, 5, 6,
## 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
weight2<-weight %>% separate(Date, c("Date","Time")," ")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 67 rows [1, 2, 3, 4, 5, 6, 7,
## 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
view(daily_sleep2)
view(weight2)
```

Next we need to distinguish how many Ids or participants are unique for each data-set in order to make there are no duplicates or missing values.

```
n_distinct(activity_daily2$Id)
```

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

```
n_distinct(daily_sleep2$Id)
```

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

```
n_distinct(weight2$Id)
```

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

In distinguishing how many IDs there are present for each table, we can see three different outcomes for the data that was tracked and logged. There's suppose to be only 30 participants in the project from the data collect through, yet when we run the function, it shows there's 33 participants, this puts in the question the credibility and accuracy of the data-set. Further only 24 people recorded their sleep scheduled and only 8 people of the 30 recorded their weights.

Now I will check the uniqueness of the amount of rows, I'll check to see if there are any duplicated rows, since there are more than 30 users showing up, we want to make sure we get rid of these duplicated rows and

numbers that can skew our analysis.

```
nrow(activity_daily2)
```

```
## [1] 863
```

```
nrow(daily_sleep2)
```

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

```
nrow(weight2)
```

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

After running our function we see that the daily sleep table has 3 more duplicated rows, therefore we will get rid of these to create a new data frame.

```
daily_sleep3<-unique(daily_sleep2)
```

Now that are data tables are finally clean and filtered we can begin to analyze it. We will start by looking a detailed summary of each data-set.

```
skim_without_charts(activity_daily2)
```

Table 1: Data summary

Name	activity_daily2
Number of rows	863
Number of columns	15
_____	
Column type frequency:	
character	1
numeric	14
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityDate	0	1	8	9	0	31	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.857542e+09	2.418405e+09	1503960366	2.320127e+09	4.445115e+09	6.962181e+09	8.877689e+09
TotalSteps	0	1	8.319390e+03	4.744970e+03	4	4.923000e+03	8.053000e+03	1.109250e+04	3.601900e+04
TotalDistance	0	1	5.980000e+00	3.720000e+00	0	3.370000e+00	5.690000e+00	7.900000e+00	2.803000e+01
TrackerDistance	0	1	5.960000e+00	3.700000e+00	0	3.370000e+00	5.690000e+00	7.880000e+00	2.803000e+01
LoggedActivitiesDistance	0	1	1.200000e-01	6.500000e-01	0	0.000000e+00	0.000000e+00	0.000000e+00	4.940000e+00
VeryActiveDistance	0	1	1.640000e+00	2.740000e+00	0	0.000000e+00	0.000000e+00	2.270000e+00	2.092000e+01
ModeratelyActiveDistance	0	1	6.200000e-01	9.100000e-01	0	0.000000e+00	0.000000e+00	8.700000e-01	6.480000e+00
LightActiveDistance	0	1	3.640000e+00	2.060000e+00	0	2.340000e+00	3.580000e+00	4.890000e+00	1.071000e+01

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
SedentaryActiveDistance	0	1	0.000000e+00	0.000000e-02	0	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e-01
VeryActiveMinutes	0	1	2.302000e+01	3.365000e+01	0	0.000000e+00	7.000000e+00	3.600000e+01	2.100000e+02
FairlyActiveMinutes	0	1	1.478000e+01	2.043000e+01	0	0.000000e+00	8.000000e+00	2.000000e+01	1.430000e+02
LightlyActiveMinutes	0	1	2.100200e+02	9.278000e+01	0	1.465000e+02	2.028000e+02	2.722000e+02	5.128000e+02
SedentaryMinutes	0	1	9.557500e+02	3.802900e+02	0	7.215000e+02	1.021000e+03	1.389000e+03	1.734000e+03
Calories	0	1	2.361300e+03	7.027100e+02	52	1.855500e+03	2.320000e+03	2.832000e+03	4.900000e+03

```
skim_without_charts(daily_sleep3)
```

Table 4: Data summary

Name	daily_sleep3
Number of rows	410
Number of columns	6
_____	
Column type frequency:	
character	2
numeric	4
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Date	0	1	8	9	0	31	0
Time	0	1	8	8	0	1	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.994963e+09	2.060863e+09	1503960366	4.977334e+09	4702921684	6962181067	8792009665
TotalSleepRecords	0	1	1.120000e+00	3.500000e-01	1	1.000000e+00	1.0	1	3
TotalMinutesAsleep	0	1	4.191700e+02	1.186400e+02	58	3.610000e+02	432.5	490	796
TotalTimeInBed	0	1	4.584800e+02	1.274600e+02	61	4.037500e+02	463.0	526	961

```
skim_without_charts(weight2)
```

Table 7: Data summary

Name	weight2
Number of rows	67
Number of columns	9
_____	
Column type frequency:	
character	2

Table 7: Data summary

logical	1
numeric	6
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Date	0	1	8	9	0	31	0
Time	0	1	7	8	0	26	0

Variable type: logical

skim_variable	n_missing	complete_rate	mean	count
IsManualReport	0	1	0.61	TRU: 41, FAL: 26

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1.00	7.009282e+09	2.950322e+09	1.503960e+09	4.962181e+09	4.962181e+09	8.877689e+09	8.877689e+09
WeightKg	0	1.00	7.204000e+01	1.392000e+01	5.126000e+01	6.114000e+01	6.125000e+01	8.505000e+01	1.335000e+02
WeightPounds	0	1.00	1.588100e+02	3.070000e+01	1.159600e+02	1.353600e+02	1.377900e+02	1.875000e+02	2.943200e+02
Fat	65	0.03	2.350000e+01	2.120000e+01	2.200000e+00	1.275000e+01	2.350000e+01	3.425000e+01	9.500000e+01
BMI	0	1.00	2.519000e+01	3.070000e+00	2.145000e+01	2.396000e+01	2.439000e+01	2.556000e+01	4.754000e+01
LogId	0	1.00	1.461772e+12	2.829948e+08	1.460444e+12	1.461079e+12	1.461802e+12	1.462375e+12	1.463098e+12

I will make these data-sets easier to read by focusing only the columns I want to see for each. I will make new dataframes for each containing only the important information or data I will use to analyze FitBits trends.

```
activity_daily_new<-activity_daily2 %>%
  select(Id, ActivityDate, TotalSteps, VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, Se
  rename(Date=ActivityDate)
```

```
daily_sleep_new<-daily_sleep3 %>%
  select(Id, Date, TotalMinutesAsleep, TotalTimeInBed)
```

```
weight_new<-weight2 %>%
  select(Id, Date, BMI, WeightPounds, IsManualReport)
```

Now let us get a quick detailed summary of each new table.

```
summary(activity_daily_new)
```

```
##        Id                 Date              TotalSteps    VeryActiveMinutes
##  Min.   :1.504e+09   Length:863         Min.   :    4   Min.   :  0.00
##  1st Qu.:2.320e+09   Class :character   1st Qu.: 4923   1st Qu.:  0.00
##  Median :4.445e+09   Mode  :character   Median : 8053   Median :  7.00
##  Mean   :4.858e+09                      Mean   : 8319   Mean   : 23.02
```

```
##  3rd Qu.:6.962e+09                     3rd Qu.:11092   3rd Qu.: 35.00
##  Max.   :8.878e+09                     Max.   :36019   Max.   :210.00
##  FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes    Calories
##  Min.   :  0.00      Min.   :  0.0        Min.   :   0.0   Min.   :  52
##  1st Qu.:  0.00      1st Qu.:146.5        1st Qu.: 721.5   1st Qu.:1856
##  Median :  8.00      Median :208.0        Median :1021.0   Median :2220
##  Mean   : 14.78      Mean   :210.0        Mean   : 955.8   Mean   :2361
##  3rd Qu.: 21.00      3rd Qu.:272.0        3rd Qu.:1189.0   3rd Qu.:2832
##  Max.   :143.00      Max.   :518.0        Max.   :1440.0   Max.   :4900
```

```
summary(daily_sleep_new)
```

```
##        Id               Date           TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.504e+09   Length:410         Min.   : 58.0      Min.   : 61.0
##  1st Qu.:3.977e+09   Class :character   1st Qu.:361.0      1st Qu.:403.8
##  Median :4.703e+09   Mode  :character   Median :432.5      Median :463.0
##  Mean   :4.995e+09                      Mean   :419.2      Mean   :458.5
##  3rd Qu.:6.962e+09                      3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :8.792e+09                      Max.   :796.0      Max.   :961.0
```

```
summary(weight_new)
```

```
##        Id               Date                BMI         WeightPounds
##  Min.   :1.504e+09   Length:67          Min.   :21.45   Min.   :116.0
##  1st Qu.:6.962e+09   Class :character   1st Qu.:23.96   1st Qu.:135.4
##  Median :6.962e+09   Mode  :character   Median :24.39   Median :137.8
##  Mean   :7.009e+09                      Mean   :25.19   Mean   :158.8
##  3rd Qu.:8.878e+09                      3rd Qu.:25.56   3rd Qu.:187.5
##  Max.   :8.878e+09                      Max.   :47.54   Max.   :294.3
##  IsManualReport
##  Mode :logical
##  FALSE:26
##  TRUE :41
##
##
##
```

Right away we can see the Activity_Daily table that minimum total of steps taken was 4, this is impossible unless the individual was in bed all day. Again, this is a concern for error in the data. The max number of step was 36,019. The average number of steps taken was 8,319. In regards to calories, the minimum calories burned was 52, the max was 2,220 and the average overall was 2,361.

According to the article "How Many Steps Should I Take a Day?" by Zoe Weiner, released on the Well and Good website, the average steps someone should take to be "considered active" is 7,500 steps a day, but to be healthy, boost metabolism and weight management, 15,000 steps are day would need to be met. With that in mind, most of the FitBit users are reaching the average steps they should take but not the necessary steps to stay healthy and have a healthy weight management.

The Daily_Sleep table shows that the minimum minutes of sleep were 58 minutes, the max was 796 minutes and on average it was 419.2 minutes. While the total time in bed, the minimum was 61 minutes, maximum was 961 minutes and on average 458 minutes.

The Centers for Disease Control and Prevention in the article "How Much Sleep Do I need?" states that the average adult needs to sleeps between 7-8 hours a night, that's between 420 to 480 minutes a night. Based on the data gather for FitBit users, the average participant is sleeping the right amount of sleep, but there are also users that are sleeping only an hour and other users that sleep more than the average time they should be sleeping, this can cause lack of activity.

The Weight table shows that the BMI minimum was 21.45, the maximum was 47.54 and on average it was 25.19. While for the weight in pounds, the minimum was 116 pounds, maximum was 294.3 pounds and on average the weight was 158.8 pounds.

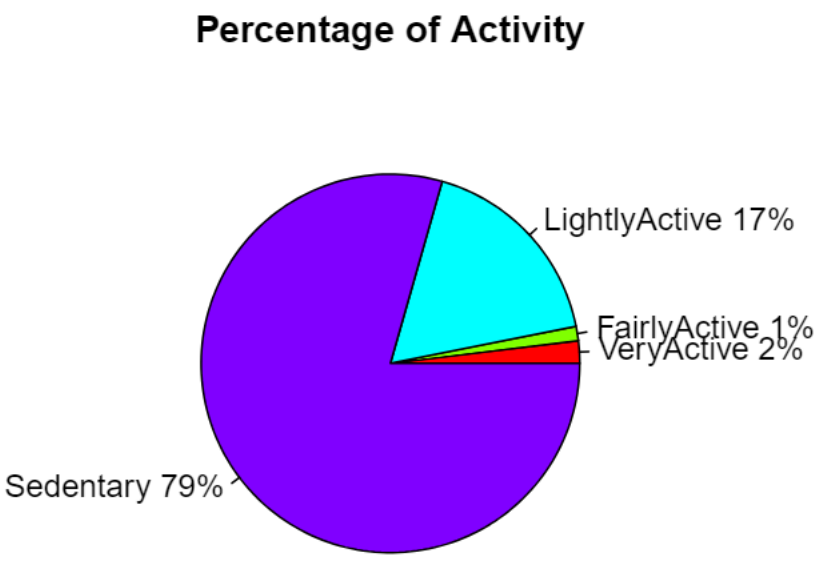
Also according to the CDC for "Accessing Your Weight," the healthy weight BMI falls between 18.5 and 24.9. A BMI between 25.0 and 29.9 is considered to be overweight. With this in mind from the Weight data-set analysis, most of the FitBit users fall in the overweight category. The minimum BMI is 21.45 this means that even the users with the minimum BMI are closer to being overweight.

Step 6: Visualizations & Findings

The first graph is a pie chart out of 100 percent that shows how much the participants tended to workout, as you can see 79% of them were in Sedentary, meaning that they were never active or didn't record how active they had been. Only 20% of the participants were actively from lightly, fairy, to very active.

```
VeryActiveMin<-sum(activity_daily_new$VeryActiveMinutes)
FairlyActiveMin<-sum(activity_daily_new$FairlyActiveMinutes)
LightlyActiveMin<-sum(activity_daily_new$LightlyActiveMinutes)
SedentaryMin<-sum(activity_daily_new$SedentaryMinutes)
TotalMin<-VeryActiveMin + FairlyActiveMin + LightlyActiveMin + SedentaryMin
```

```
slices<-c(VeryActiveMin, FairlyActiveMin, LightlyActiveMin, SedentaryMin)
lbls<-c("VeryActive", "FairlyActive", "LightlyActive", "Sedentary")
pct <-round(slices/sum(slices)*100)
lbls <-paste(lbls, pct)
lbls<-paste(lbls, "%", sep = "")
pie(slices, labels = lbls, col = rainbow(length(lbls)),main ="Percentage of Activity")
```



This scatter plot shows how the more steps people took the more calories they would burn. Again as we can wee most of the participants didn't take as much steps or record most of their steps. Therefore the data can be inaccurate or there is missing information.'

```
ggplot(data=activity_daily_new) +
  geom_point(mapping=aes(x=TotalSteps, y=Calories), color="yellow") +
  geom_smooth(mapping=aes(x=TotalSteps, y=Calories)) +
  labs(title="Relationship Between Total Steps and Calories Burned", x="Total Steps", y="Calories Burne
```

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

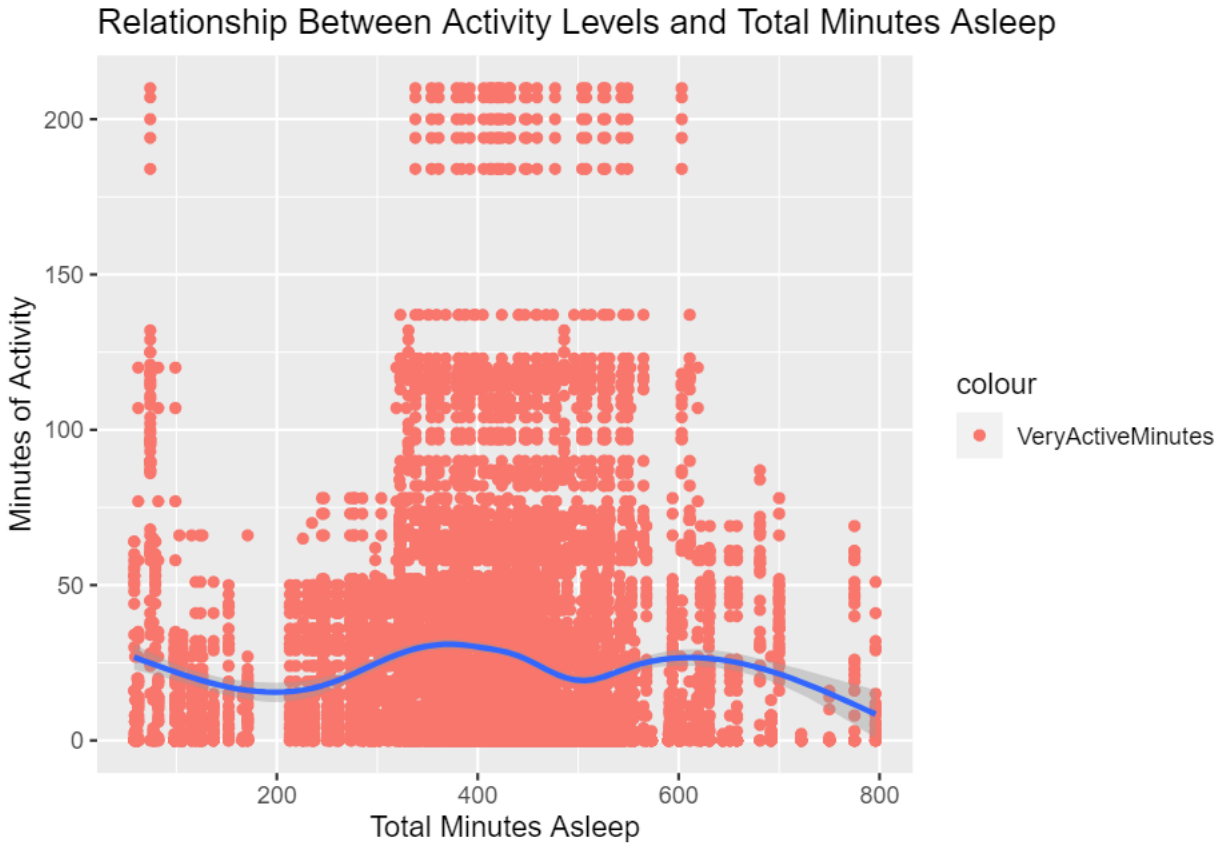
Now lets combine the data of Activity Daily and Daily Sleep to see if there is any correlation between both factors.

```
combined_data<- merge(activity_daily_new, daily_sleep_new, by="Id")
```

```
ggplot(data=combined_data) +
  geom_point(mapping=aes(x=TotalMinutesAsleep, y=VeryActiveMinutes, color="VeryActiveMinutes")) +
  geom_smooth(mapping=aes(x=TotalMinutesAsleep, y=VeryActiveMinutes, regLineColor="blue"))+
  labs(title="Relationship Between Activity Levels and Total Minutes Asleep", x="Total Minutes Asleep",
```

```
## Warning in geom_smooth(mapping = aes(x = TotalMinutesAsleep, y =
## VeryActiveMinutes, : Ignoring unknown aesthetics: regLineColor
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

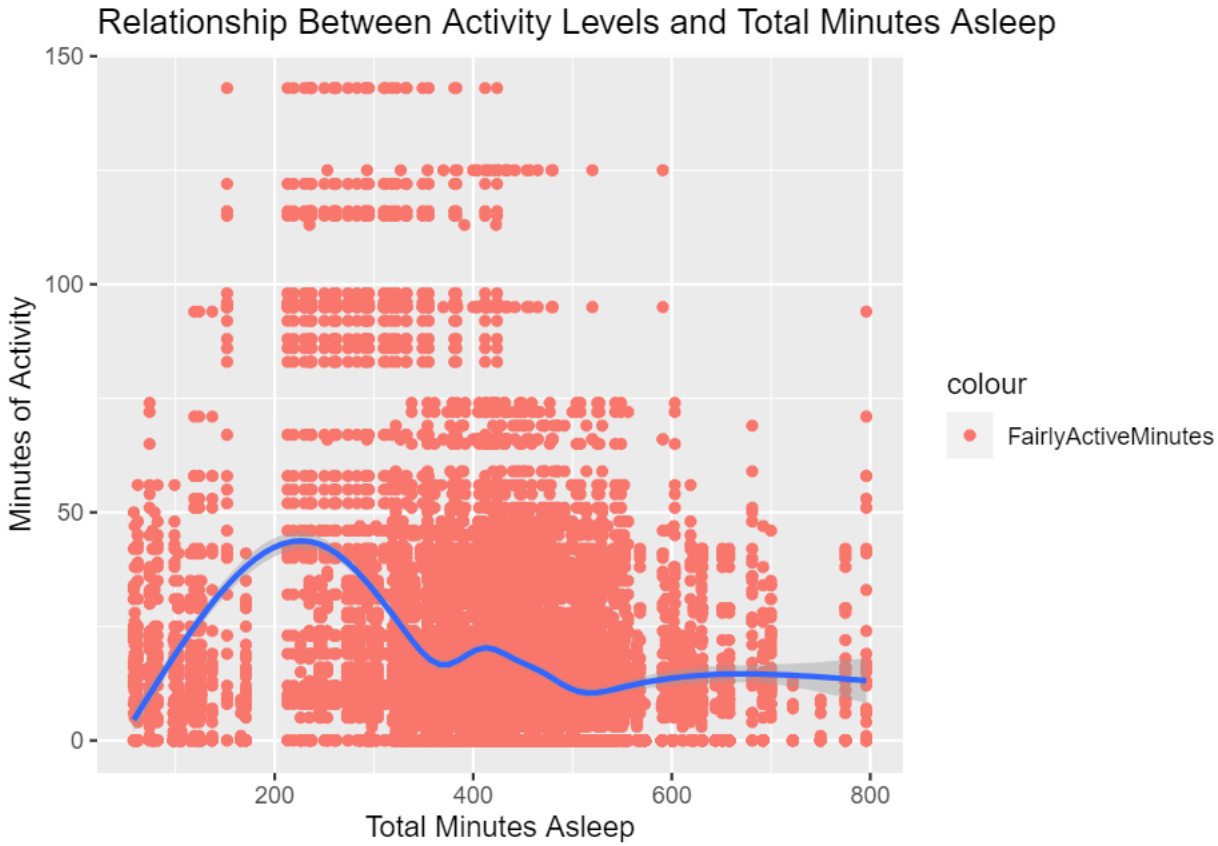


This table shows that the less minutes very active participants slept, the more minutes of activity they would have.

```
ggplot(data=combined_data) +
  geom_point(mapping=aes(x=TotalMinutesAsleep, y=FairlyActiveMinutes, color="FairlyActiveMinutes")) +
  geom_smooth(mapping=aes(x=TotalMinutesAsleep, y=FairlyActiveMinutes, regLineColor="blue")) +
  labs(title="Relationship Between Activity Levels and Total Minutes Asleep", x="Total Minutes Asleep",
```

```
## Warning in geom_smooth(mapping = aes(x = TotalMinutesAsleep, y =
## FairlyActiveMinutes, : Ignoring unknown aesthetics: regLineColor
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

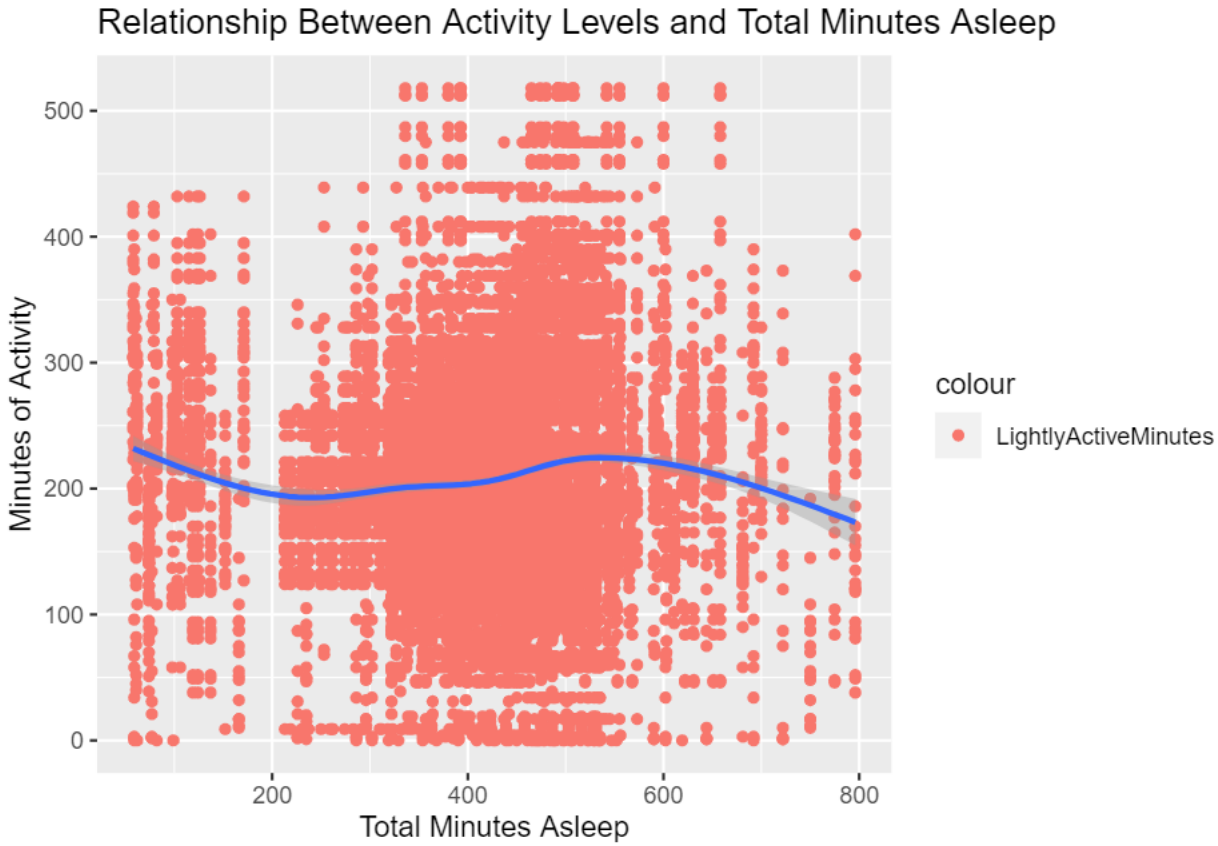


While when participants were Fairly Active their total minutes of sleep shows that the more minutes they slept, the less active they would be.

```
ggplot(data=combined_data) +
  geom_point(mapping=aes(x=TotalMinutesAsleep, y=LightlyActiveMinutes, color="LightlyActiveMinutes")) +
  geom_smooth(mapping=aes(x=TotalMinutesAsleep, y=LightlyActiveMinutes, regLineColor="blue")) +
  labs(title="Relationship Between Activity Levels and Total Minutes Asleep", x="Total Minutes Asleep",
```

```
## Warning in geom_smooth(mapping = aes(x = TotalMinutesAsleep, y =
## LightlyActiveMinutes, : Ignoring unknown aesthetics: regLineColor
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

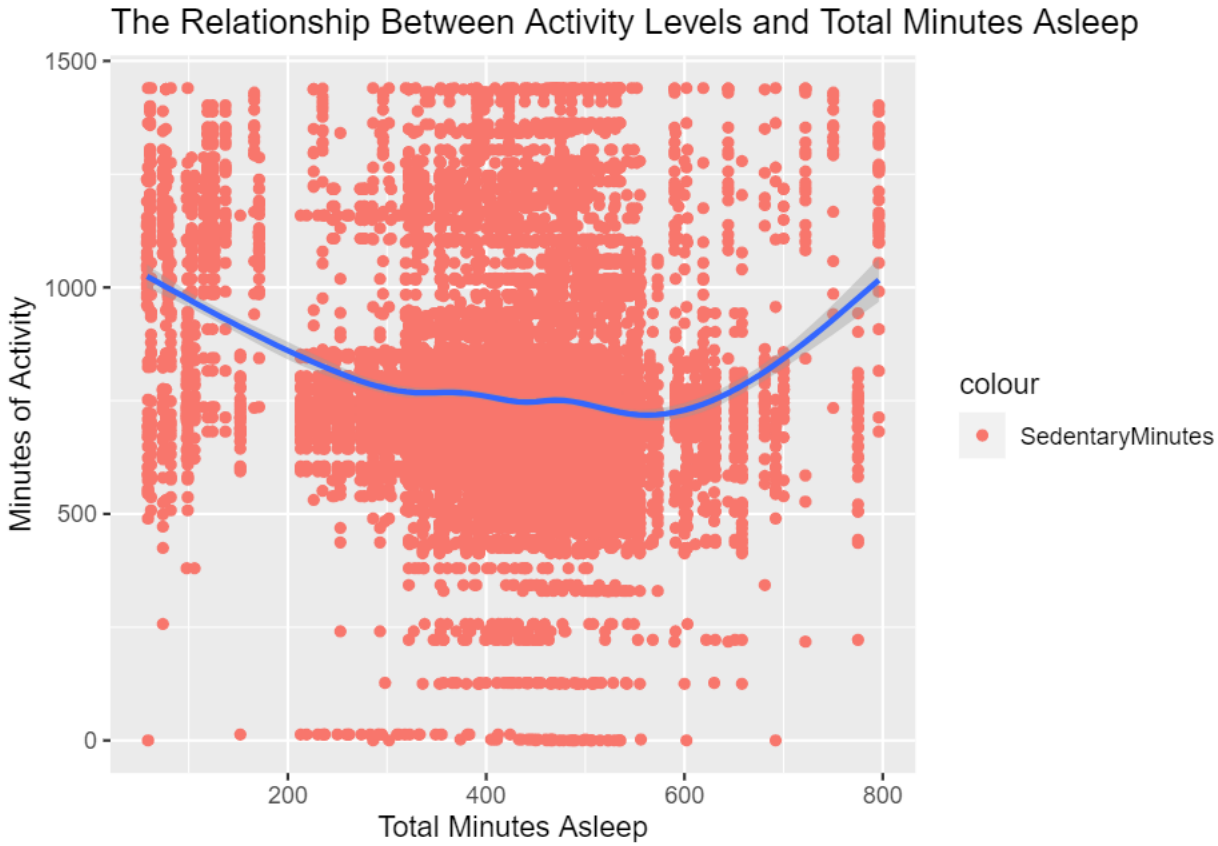


The table for Lightly Active participants stayed almost constant, they more minutes they sleep the more their activity stayed the same as their total minutes.

```
ggplot(data=combined_data) +
  geom_point(mapping=aes(x=TotalMinutesAsleep, y=SedentaryMinutes, color="SedentaryMinutes")) +
  geom_smooth(mapping=aes(x=TotalMinutesAsleep, y=SedentaryMinutes, regLineColor="blue")) +
  labs(title="The Relationship Between Activity Levels and Total Minutes Asleep", x="Total Minutes Asle
```

```
## Warning in geom_smooth(mapping = aes(x = TotalMinutesAsleep, y =
## SedentaryMinutes, : Ignoring unknown aesthetics: regLineColor
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

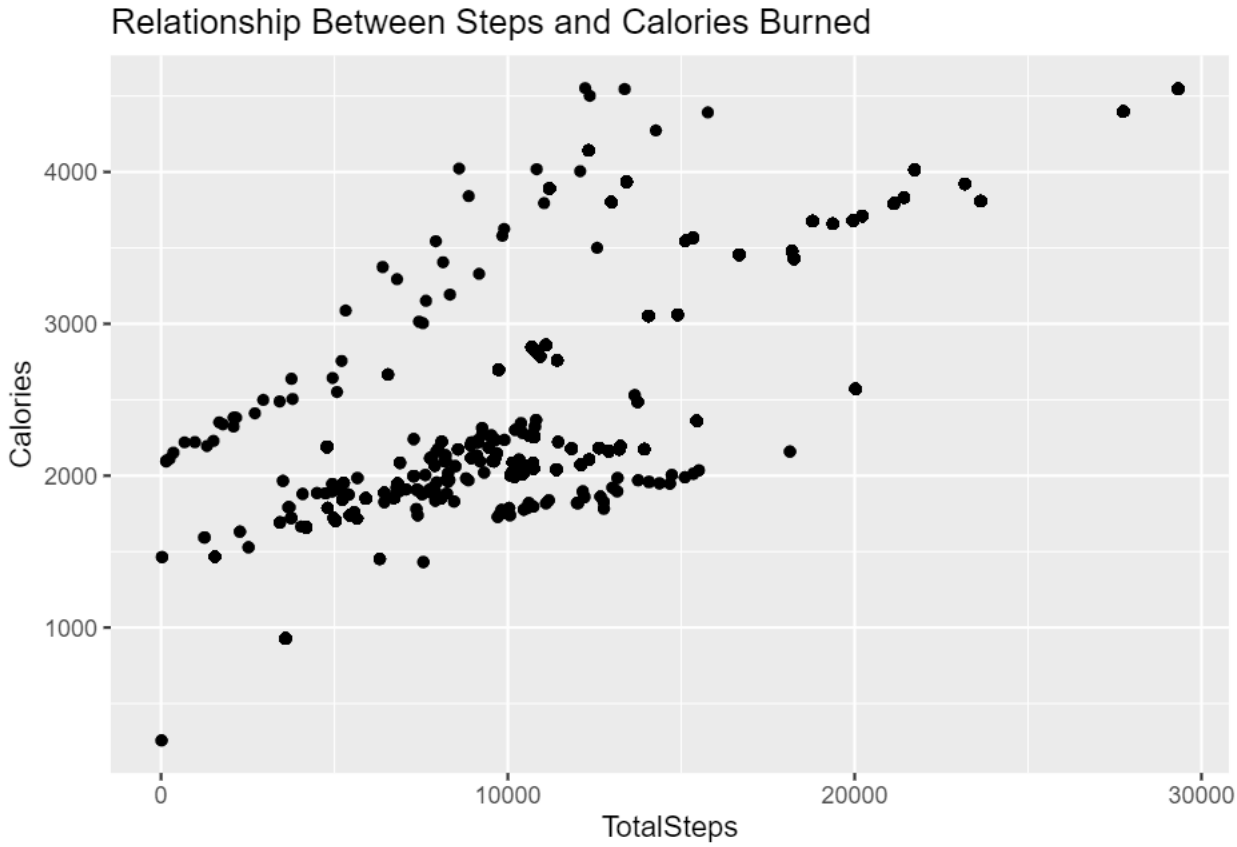


Now I'll see if there's any correlation between the activity daily and weight by each participant. I will merge both tables to get to see the correlation between them.

```
weight_activitydaily<- merge(activity_daily_new, weight_new, by="Id")
```

The sedentary participants table shows that the more they slept, the more sedentary they would be.

```
ggplot(data=weight_activitydaily)+
  geom_point(mapping=aes(x=TotalSteps, y=Calories))+
  labs(title="Relationship Between Steps and Calories Burned")
```

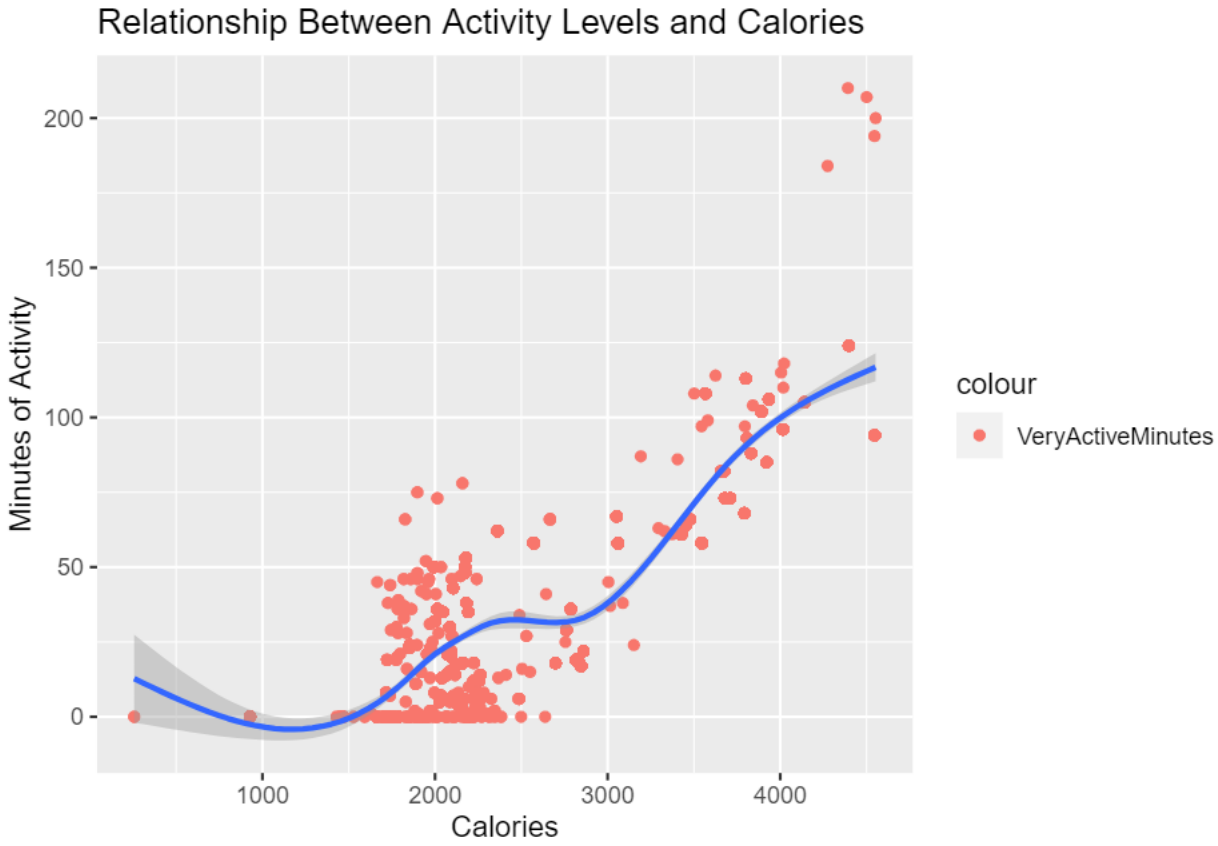


As expected like the previous tables, the more steps participants took, the more calories they would burn.

```
ggplot(data=weight_activitydaily) +
  geom_point(mapping=aes(x=Calories, y=VeryActiveMinutes, color="VeryActiveMinutes")) +
  geom_smooth(mapping=aes(x=Calories, y=VeryActiveMinutes, regLineColor="blue"))+
  labs(title="Relationship Between Activity Levels and Calories", x="Calories", y="Minutes of Activity"
```

```
## Warning in geom_smooth(mapping = aes(x = Calories, y = VeryActiveMinutes, :
## Ignoring unknown aesthetics: regLineColor
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

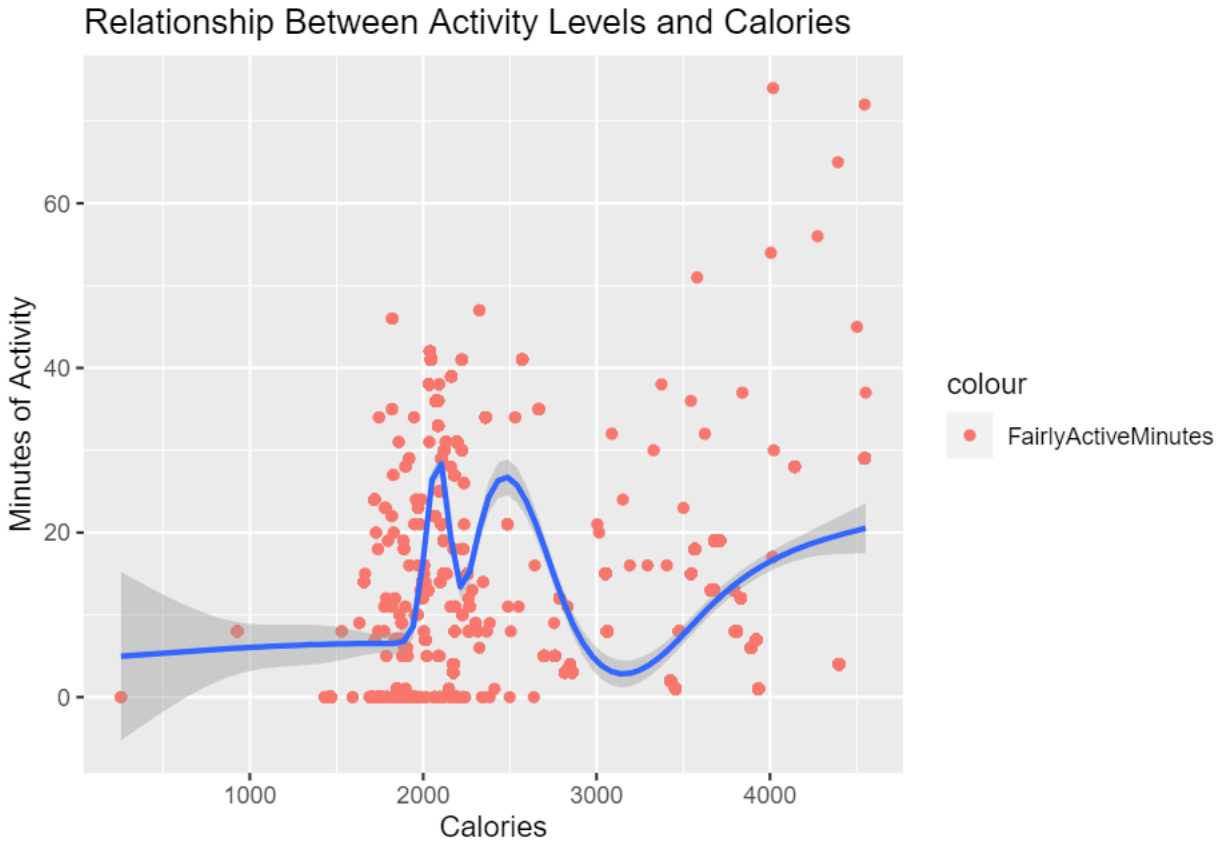


This table shows the relationship between Activity Level for Very Active participants and the calories burned, as we can see, the more minutes they were active the more calories were burned over time.

```
ggplot(data=weight_activitydaily) +
  geom_point(mapping=aes(x=Calories, y=FairlyActiveMinutes, color="FairlyActiveMinutes")) +
  geom_smooth(mapping=aes(x=Calories, y=FairlyActiveMinutes, regLineColor="blue"))+
  labs(title="Relationship Between Activity Levels and Calories", x="Calories", y="Minutes of Activity"
```

```
## Warning in geom_smooth(mapping = aes(x = Calories, y = FairlyActiveMinutes, :
## Ignoring unknown aesthetics: regLineColor
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

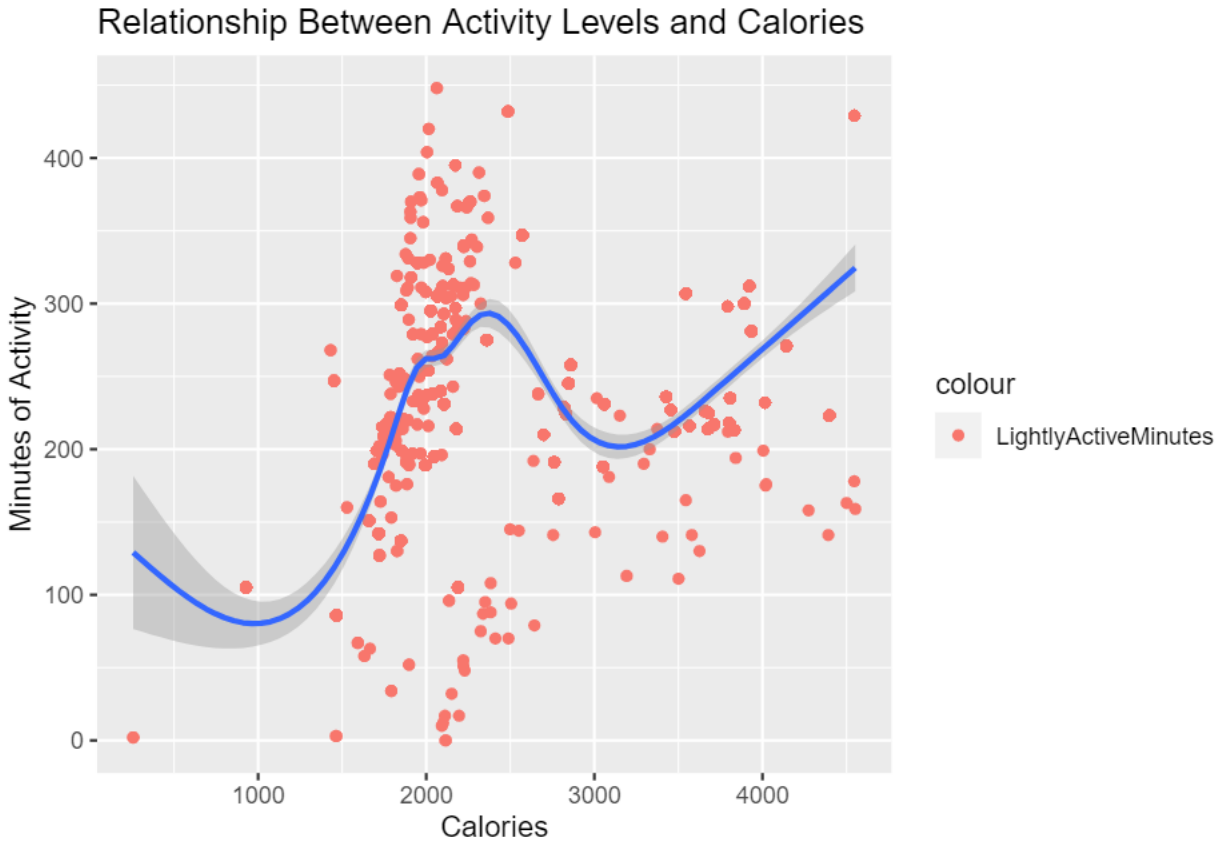


This table shows the relationship between the participants that were fairly Active and the calories that they burned. As we can see they mostly worked on average 30 to 40 minutes a day so their calories burned stayed around the same amount, expect for those few that worked out more and burned more calories.

```
ggplot(data=weight_activitydaily) +
  geom_point(mapping=aes(x=Calories, y=LightlyActiveMinutes, color="LightlyActiveMinutes")) +
  geom_smooth(mapping=aes(x=Calories, y=LightlyActiveMinutes, regLineColor="blue"))+
  labs(title="Relationship Between Activity Levels and Calories", x="Calories", y="Minutes of Activity"
```

```
## Warning in geom_smooth(mapping = aes(x = Calories, y = LightlyActiveMinutes, :
## Ignoring unknown aesthetics: regLineColor
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



This table shows the relationship between the participants that were Lightly Active and the calories burned, it was a positive correlation in two points. Most calories were burned when they worked out between 200 to 400 minutes. The calories stayed consistent around those minutes.

```
ggplot(data=weight_activitydaily) +
  geom_point(mapping=aes(x=Calories, y=SedentaryMinutes, color="SedentaryMinutes"))+
  geom_smooth(mapping=aes(x=Calories, y=SedentaryMinutes, regLineColor="blue"))+
  labs(title="Relationship Between Activity Levels and Calories", x="Calories", y="Minutes of Activity"
```

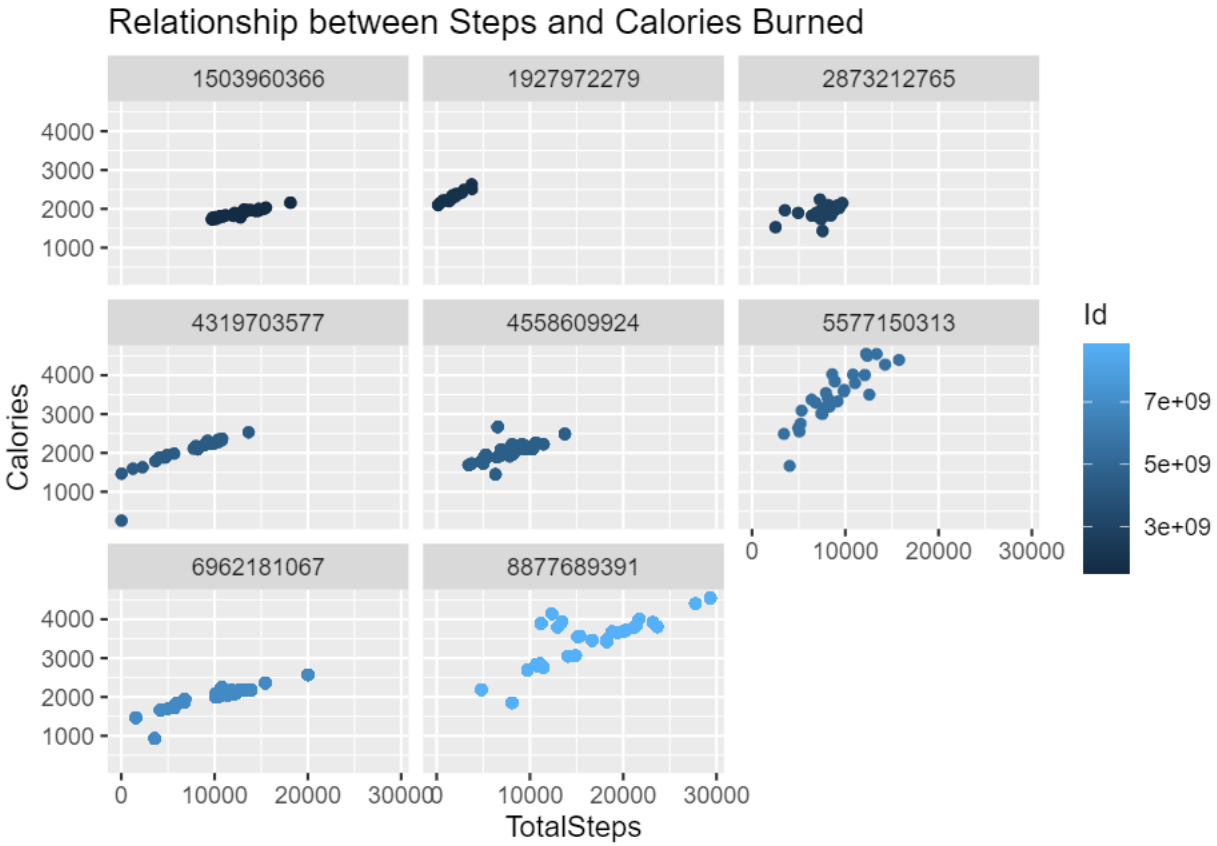
```
## Warning in geom_smooth(mapping = aes(x = Calories, y = SedentaryMinutes, :
## Ignoring unknown aesthetics: regLineColor
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

This table shows the relationship between participants who were Sedentary and the calories burned, it starts with light positive correlation and begins to go downward. This means that the less activity they began to have the less calories they would burn. They were mostly stagnant and didn't lose much calories as the other participants did.

Next we I ran this function with facet wrap to see more in detail each participant and the calories and steps they took.

```
ggplot(data=weight_activitydaily, aes(x=TotalSteps, y=Calories))+
  geom_point(aes(color=Id))+ facet_wrap(~Id)+
  labs(title = "Relationship between Steps and Calories Burned")
```



We only got back 8 tables, reminder that only 8 participants recorded their calories and weight change, this is why instead of getting all 30 participants and a table for each, we only receive back 8. This further proves that the data collected is not accurate and a good data set to make analysis. The lack of information from the remaining participants shows that the information is already skewed and will lack the necessary information that Bellabeat will need to answer its question of the marketing trends.

From the analysis and tables created, we can see that there is a lack of information and data that is still needed, we can try to answer the questions of the device usage current trends, but our analysis will be skewed and not current. Many participants didn't record their data or logged in steps, weight and activity for some days. When they did have their FitBits on, we noticed that the more active they were the more calories they would burn and lose weight. The same can be said for when they were barely active, the less active they were the less calories they would burn. Reasons why data might not have collected or logged by participants can be for various reasons, the Fitbit could have run out of battery, or they simply could have forgotten to put it on.

Step 7: Act

That being said, based on the analysis of FitBit, Bellabeat, can try to implement ways to have it's consumers log and track their information via the app or devices by giving incentives or reminders. Having daily or hourly reminders can be the perfect way to have consumers be able to record their steps, weight, and calories. Nowadays, our devices are able to detect when we are moving, based on this, Bellabeat can try to find a way to utilize this new trend by recognizing when consumers are moving or being active and sending a reminder to them to record their time or steps. This would create consistency and reliability in the data that can be collected from Bellabeat consumers in order to find ways to make our devices easier to use and exciting for consumers.

In regards to the data sets, one recommendation I would have that would make this data more accurate, reliable, and unique, would be to collect data for the exact 2 months or 60 days. Along with having a bigger sample size of the population; 30 participants isn't enough to create an analysis, the data is inaccurate, not

consistent and creates issues when trying to figure out the business task. There's also the potential of it being biased. When there's a small sample size there's limitations to the data and chance of error can occur. The data should also be more current and up to date. This data-sets are from 2016, that is 8 years ago, the device usage trends have definitely changed along with user's habits. This means that the data provided was inaccurate and not current, leading to potential errors or analysis in the final finding of the data-sets.

Bibliography:

CDC. "Assessing Your Weight." Centers for Disease Control and Prevention, 9 June 2023, https://www.cdc.gov/healthyweight/assessing/index.html.

—. "How Much Sleep Do I Need?" Centers for Disease Control and Prevention, 14 Sept. 2022, https://www.cdc.gov/sleep/about_sleep/how_much_sleep.html.

Furberg, R., Brinton, J., Keating, M., & Ortiz, A. (2016). Crowd-sourced Fitbit datasets 03.12.2016-05.12.2016 [Data set]. Zenodo., https://doi.org/10.5281/zenodo.53894

Weiner, Zoe. "How Many Steps Should I Take a Day?" Well+Good, 22 June 2022, https://www.wellandgood.com/how-many-steps-should-i-take-a-day/.