Google Data Analytics - Case Study: Bellabeat

How Can a Wellness Technology Company Play It Smart?

Introduction

Welcome to my Bellabeat data analysis case study. In this case study, I will perform the knowledge that I learned from Google Data Analytics Professional Certificate. In order to answer the key business questions, I will follow the steps of the data analysis process: ask, prepare, process, analyze, share, and act. Along the way, the Case Study Roadmap tables — including guiding questions and key tasks.

About the company

Urška Sršen and Sando Mur founded Bellabeat, a high-tech company that manufactures health-focused smart products. Since it was founded in 2013, Bellabeat has grown rapidly and quickly positioned itself as a tech-driven wellness company for women. By 2016, Bellabeat had opened offices around the world and launched multiple products. Bellabeat products became available through a growing number of online retailers in addition to their own e-commerce channel on their website.

Ask Phase

Identify the business task

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

The first thing is to recognize who are the potential customers of Bellabeat based on their usage of their fitness smart devices. Next is the answer is there any relationship between customer behaviors and data we have. After that, what is the effect of those trends on Bellabeat's marketing strategies.

Consider key stakeholders

The main stakeholders are Urška Sršen and Sando Mur, the founders of Bellabeat. The other stakeholders are Bellabeat marketing team and maybe there is also my manager.

Prepare Phase

Choosing the suitable dataset.

Sršen encourages to use public data that explores smart device users' daily habits. She points you to a specific data set: FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius):This Kaggle data set contains personal fitness tracker from thirty fitbit users. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. It includes information about daily activity, steps, and heart rate that can be used to explore users' habits.

I will use R programing language to work with this dataset.

Installing and loading common packages and libraries

```
install.packages("tidyverse")
install.packages("lubridate")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("tidyr")
install.packages("here")
install.packages("skimr")
install.packages("janitor")
```

```
library(tidyverse)
library(lubridate)
library(dplyr)
library(ggplot2)
library(tidyr)
library(here)
library(skimr)
library(janitor)
library(readr)
```

Importing dataset

In this step, I will import all datasets that I need to use for this project.

```
Activity <- read_csv("dailyActivity_merged.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.
```

```
head(Activity)
```

```
## # A tibble: 6 x 15
##        Id ActivityDate TotalSteps TotalDistance TrackerDistance LoggedActivitie~
##     <dbl> <chr>             <dbl>         <dbl>           <dbl>            <dbl>
## 1  1.50e9 4/12/2016         13162          8.5             8.5                 0
## 2  1.50e9 4/13/2016         10735          6.97            6.97                0
## 3  1.50e9 4/14/2016         10460          6.74            6.74                0
## 4  1.50e9 4/15/2016          9762          6.28            6.28                0
## 5  1.50e9 4/16/2016         12669          8.16            8.16                0
## 6  1.50e9 4/17/2016          9705          6.48            6.48                0
## # ... with 9 more variables: VeryActiveDistance <dbl>,
## #   ModeratelyActiveDistance <dbl>, LightActiveDistance <dbl>,
## #   SedentaryActiveDistance <dbl>, VeryActiveMinutes <dbl>,
## #   FairlyActiveMinutes <dbl>, LightlyActiveMinutes <dbl>,
## #   SedentaryMinutes <dbl>, Calories <dbl>
```

```
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)
```

```
## spec_tbl_df [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>
```

```
Heartrate <- read_csv("heartrate_seconds_merged.csv")
```

heartrate_seconds_merged.csv

```
## Rows: 2483658 Columns: 3
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Time
## dbl (2): Id, Value
##
## 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.
```

```
head(Heartrate)
```

```
## # A tibble: 6 x 3
##           Id Time                  Value
##        <dbl> <chr>                 <dbl>
## 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)
```

```
## spec_tbl_df [2,483,658 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id   : num [1:2483658] 2.02e+09 2.02e+09 2.02e+09 2.02e+09 2.02e+09 ...
##  $ Time : chr [1:2483658] "4/12/2016 7:21:00 AM" "4/12/2016 7:21:05 AM" "4/12/2016 7:21:10 AM" "4/12/
##  $ Value: num [1:2483658] 97 102 105 103 101 95 91 93 94 93 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   Time = col_character(),
##   ..   Value = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
Sleep <- read_csv("sleepDay_merged.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.
```

```
head(Sleep)
```

```
## # A tibble: 6 x 5
##           Id SleepDay           TotalSleepRecor~ TotalMinutesAsl~ TotalTimeInBed
##        <dbl> <chr>                         <dbl>            <dbl>          <dbl>
## 1 1503960366 4/12/2016 12:00:0~                1              327            346
```

```
## 2 1503960366 4/13/2016 12:00:0~                2              384            407
## 3 1503960366 4/15/2016 12:00:0~                1              412            442
## 4 1503960366 4/16/2016 12:00:0~                2              340            367
## 5 1503960366 4/17/2016 12:00:0~                1              700            712
## 6 1503960366 4/19/2016 12:00:0~                1              304            320
```

```
colnames(Sleep)
```

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

```
str(Sleep)
```

```
## spec_tbl_df [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 12:00:0
##  $ 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>
```

```
Weight <- read_csv("weightLogInfo_merged.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.
```

```
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
```

```
colnames(Weight)
```

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

```
str(Weight)
```

```
## spec_tbl_df [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" "4
##  $ 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>
```

```
HourlyIntensities <- read_csv("hourlyIntensities_merged.csv")
```

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.
```

```
head(HourlyIntensities)
```

```
## # A tibble: 6 x 4
##           Id ActivityHour          TotalIntensity AverageIntensity
##        <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
```

```
colnames(HourlyIntensities)
```

```
## [1] "Id"               "ActivityHour"     "TotalIntensity"   "AverageIntensity"
```

```
str(HourlyIntensities)
```

```
## spec_tbl_df [22,099 x 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id              : num [1:22099] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour    : chr [1:22099] "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00
##  $ TotalIntensity  : num [1:22099] 20 8 7 0 0 0 0 0 13 30 ...
##  $ AverageIntensity: num [1:22099] 0.333 0.133 0.117 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityHour = col_character(),
##   ..   TotalIntensity = col_double(),
##   ..   AverageIntensity = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Process Phase

Cleaning the dataset

I used functions like skim_without_charts() to review the datasets. I also used clean_name() to clean names.

```
clean_names(Activity)
```

```
## # 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
## # ... with 930 more rows, and 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(Heartrate)
```

```
## # A tibble: 2,483,658 x 3
##            id time                  value
##         <dbl> <chr>                 <dbl>
##  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
##  7 2022484408 4/12/2016 7:22:10 AM    91
##  8 2022484408 4/12/2016 7:22:15 AM    93
##  9 2022484408 4/12/2016 7:22:20 AM    94
## 10 2022484408 4/12/2016 7:22:25 AM    93
## # ... with 2,483,648 more rows
```

```
clean_names(Sleep)
```

```
## # A tibble: 413 x 5
##            id sleep_day       total_sleep_rec~ total_minutes_a~ total_time_in_b~
##         <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
##  7 1503960366 4/20/2016 12:0~                1              360              377
##  8 1503960366 4/21/2016 12:0~                1              325              364
##  9 1503960366 4/23/2016 12:0~                1              361              384
## 10 1503960366 4/24/2016 12:0~                1              430              449
## # ... with 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
## # ... with 57 more rows
```

```
clean_names(HourlyIntensities)
```

```
## # 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
## # ... with 22,089 more rows
```

```
skim_without_charts(Activity)
```

Table 1: Data summary

Name	Activity
Number of rows	940
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	4.855407e+09	2.424805e+09	1503960366	2.320127e+09	4.445115e+09	6.962181e+09	8.877689e+09
TotalSteps	0	1	7.637910e+03	5.087150e+03	0	3.789750e+03	7.405500e+03	1.072700e+04	3.601900e+04
TotalDistance	0	1	5.490000e+00	3.920000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803000e+01
TrackerDistance	0	1	5.480000e+00	3.910000e+00	0	2.620000e+00	5.240000e+00	7.710000e+00	2.803000e+01
LoggedActivitiesDistance	0	1	1.100000e-01	6.200000e-01	0	0.000000e+00	0.000000e+00	0.000000e+00	4.940000e+00
VeryActiveDistance	0	1	1.500000e+00	2.660000e+00	0	0.000000e+00	1.000000e-01	2.050000e+00	2.192000e+01
ModeratelyActiveDistance	0	1	5.700000e-01	8.800000e-01	0	0.000000e+00	2.000000e-01	8.000000e-01	6.480000e+00
LightActiveDistance	0	1	3.340000e+00	2.040000e+00	0	1.950000e+00	3.360000e+00	4.780000e+00	1.071000e+01
SedentaryActiveDistance	0	1	0.000000e+00	0.000000e+00	0	0.000000e+00	0.000000e+00	0.000000e+00	1.100000e-01
VeryActiveMinutes	0	1	2.116000e+01	3.284000e+01	0	0.000000e+00	4.000000e+00	3.200000e+01	2.100000e+02
FairlyActiveMinutes	0	1	1.356000e+01	1.999000e+01	0	0.000000e+00	6.000000e+00	1.900000e+01	1.430000e+02
LightlyActiveMinutes	0	1	1.928100e+02	1.091700e+02	0	1.270000e+02	1.990000e+02	2.640000e+02	5.280000e+02
SedentaryMinutes	0	1	9.912100e+02	3.012700e+02	0	7.297500e+02	1.057500e+03	1.329500e+03	1.340000e+03
Calories	0	1	2.303610e+03	7.181700e+02	0	1.828500e+03	2.134000e+03	2.793250e+03	4.900000e+03

```
skim_without_charts(Heartrate)
```

Table 4: Data summary

Name	Heartrate

Table 4: Data summary

Number of rows	2483658
Number of columns	3
Column type frequency:	
character	1
numeric	2
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Time	0	1	19	21	0	961274	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	5.513765e+09	2.950223e+09	1.502248e+09	4.388162e+09	4.558396e+09	6.962181e+09	8.877689e+09
Value	0	1	7.733000e+01	19.400000e+00	36	63	73	88	203

```
skim_without_charts(Sleep)
```

Table 7: Data summary

Name	Sleep
Number of rows	413
Number of columns	5
Column type frequency:	
character	1
numeric	4
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
SleepDay	0	1	20	21	0	31	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	5.000979e+09	2.906036e+09	1.503960e+09	2.977333e+09	4.702922e+09	6.962181e+09	8.792010e+09
TotalSleepRecords	0	1	1.120000e+00	3.500000e-01	1	1	1	1	3

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
TotalMinutesAsleep	0	1	4.194700e+02	1.218340e+02	58	361	433	490	796
TotalTimeInBed	0	1	4.586400e+02	1.227100e+02	61	403	463	526	961

```
skim_without_charts(Weight)
```

Table 10: Data summary

Name	Weight
Number of rows	67
Number of columns	8
_____	
Column type frequency:	
character	1
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	19	21	0	56	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	6.962181e+09	6.962181e+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	1.129600e+02	1.348600e+02	1.350300e+02	1.875000e+02	2.943200e+02
Fat	65	0.03	2.350000e+01	1.120000e+01	2.200000e+01	2.275000e+01	2.350000e+01	2.425000e+01	2.500000e+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

```
skim_without_charts(HourlyIntensities)
```

Table 14: Data summary

Name	HourlyIntensities
Number of rows	22099
Number of columns	4
_____	

Table 14: Data summary

Column type frequency:	
character	1
numeric	3
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ActivityHour	0	1	19	21	0	736	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Id	0	1	4.848235e+09	2.424225e+09	1503960366	2320127002	4.445115e+09	6.962181e+09	8877689391
TotalIntensity	0	1	1.204000e+02	1.111130e+01	0	0	3.000000e+00	1.600000e+01	180
AverageIntensity	0	1	2.000000e-01	3.5000e-01	0	0	5.000000e-02	2.700000e-01	3

- There are 65 missing values in total 67 values of column fat in Weight dataframe.
- For Activity dataframe: I did not find any Spelling error, Misfield value, Missing value, Extra or blank space.

To find and remove duplicate values, I identified them by duplicated() and distinct():

```
get_dupes(Sleep)
```

```
## No variable names specified - using all columns.
```

```
## # A tibble: 6 x 6
##          Id SleepDay TotalSleepRecor~ TotalMinutesAsl~ TotalTimeInBed dupe_count
##       <dbl> <chr>               <dbl>            <dbl>          <dbl>      <int>
## 1    4.39e9 5/5/201~                1              471            495          2
## 2    4.39e9 5/5/201~                1              471            495          2
## 3    4.70e9 5/7/201~                1              520            543          2
## 4    4.70e9 5/7/201~                1              520            543          2
## 5    8.38e9 4/25/20~                1              388            402          2
## 6    8.38e9 4/25/20~                1              388            402          2
```

- There are 3 duplicate rows in Sleep dataframe, so that I removed them with distinct() function:

```
distinct(Sleep)
```

```
## # A tibble: 410 x 5
##            Id SleepDay          TotalSleepRecor~ TotalMinutesAsl~ TotalTimeInBed
##         <dbl> <chr>                        <dbl>            <dbl>          <dbl>
##  1 1503960366 4/12/2016 12:00:~                1              327            346
##  2 1503960366 4/13/2016 12:00:~                2              384            407
##  3 1503960366 4/15/2016 12:00:~                1              412            442
##  4 1503960366 4/16/2016 12:00:~                2              340            367
##  5 1503960366 4/17/2016 12:00:~                1              700            712
```

```
##  6 1503960366 4/19/2016 12:00:~                1              304            320
##  7 1503960366 4/20/2016 12:00:~                1              360            377
##  8 1503960366 4/21/2016 12:00:~                1              325            364
##  9 1503960366 4/23/2016 12:00:~                1              361            384
## 10 1503960366 4/24/2016 12:00:~                1              430            449
## # ... with 400 more rows
```

Formatting dataset:

I am going to change the data type from character to date time and split to date and time

```
Activity$date <- mdy(Activity$ActivityDate)
glimpse(Activity$date)
```

Activity dataframe:

```
##  Date[1:940], format: "2016-04-12" "2016-04-13" "2016-04-14" "2016-04-15" "2016-04-16" ...
```

```
Heartrate$Date_Time <- mdy_hms(Heartrate$Time,tz=Sys.timezone())
glimpse(Heartrate$Date_Time)
```

Heartrate dataframe:

```
##  POSIXct[1:2483658], format: "2016-04-12 07:21:00" "2016-04-12 07:21:05" "2016-04-12 07:21:10" ...
```

```
Heartrate$Date <- as.Date(Heartrate$Date_Time)
glimpse(Heartrate$Date)
```

```
##  Date[1:2483658], format: "2016-04-12" "2016-04-12" "2016-04-12" "2016-04-12" "2016-04-12" ...
```

```
Sleep$Date_Time <- mdy_hms(Sleep$SleepDay,tz=Sys.timezone())
glimpse(Sleep$Date_Time)
```

Sleep dataframe:

```
##  POSIXct[1:413], format: "2016-04-12" "2016-04-13" "2016-04-15" "2016-04-16" "2016-04-17" ...
```

```
Sleep$Date <- as.Date(Sleep$Date_Time)
glimpse(Sleep$Date)
```

```
##  Date[1:413], format: "2016-04-12" "2016-04-13" "2016-04-15" "2016-04-16" "2016-04-17" ...
```

```
Weight$Date_Time <- mdy_hms(Weight$Date,tz=Sys.timezone())
glimpse(Weight$Date_Time)
```

Weight dataframe:

```
##  POSIXct[1:67], format: "2016-05-02 23:59:59" "2016-05-03 23:59:59" "2016-04-13 01:08:52" ...
```

```
Weight$Day <- as.Date(Weight$Date_Time)
glimpse(Weight$Day)
```

```
##  Date[1:67], format: "2016-05-02" "2016-05-03" "2016-04-13" "2016-04-21" "2016-05-12" ...
```

```
HourlyIntensities$Date_Time <- mdy_hms(HourlyIntensities$ActivityHour,tz=Sys.timezone())
glimpse(HourlyIntensities$Date_Time)
```

HourlyIntensities dataframe

```
##  POSIXct[1:22099], format: "2016-04-12 00:00:00" "2016-04-12 01:00:00" "2016-04-12 02:00:00" ...
```

```
HourlyIntensities$Time <- format(as.POSIXct(HourlyIntensities$Date_Time),format = "%H:%M:%S")
glimpse(HourlyIntensities$Time)
```

```
##  chr [1:22099] "00:00:00" "01:00:00" "02:00:00" "03:00:00" "04:00:00" ...
```

Analyze Phase

The total number of participants in each data set

```
n_distinct(Activity$Id)
```

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

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

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

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

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

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

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

There are 33 participants in Activity dataframes. 24 participants in Sleep dataframe. Heartrate and Weight dataframes only have 14 and 8 participants.

Sumary dataset

```
Activity %>%
  select(TotalSteps,
         TotalDistance,
         SedentaryMinutes,
         VeryActiveMinutes,
         FairlyActiveMinutes,
         LightlyActiveMinutes,
         SedentaryMinutes,
         Calories) %>%
  summary()
```

Activity dataframe:

```
##    TotalSteps    TotalDistance    SedentaryMinutes VeryActiveMinutes
##  Min.   :    0   Min.   : 0.000   Min.   :   0.0   Min.   :  0.00
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 729.8   1st Qu.:  0.00
##  Median : 7406   Median : 5.245   Median :1057.5   Median :  4.00
##  Mean   : 7638   Mean   : 5.490   Mean   : 991.2   Mean   : 21.16
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:1229.5   3rd Qu.: 32.00
```

```
##  Max.   :36019   Max.   :28.030   Max.   :1440.0   Max.   :210.00
##  FairlyActiveMinutes LightlyActiveMinutes    Calories
##  Min.   :  0.00      Min.   :  0.0        Min.   :   0
##  1st Qu.:  0.00      1st Qu.:127.0        1st Qu.:1828
##  Median :  6.00      Median :199.0        Median :2134
##  Mean   : 13.56      Mean   :192.8        Mean   :2304
##  3rd Qu.: 19.00      3rd Qu.:264.0        3rd Qu.:2793
##  Max.   :143.00      Max.   :518.0        Max.   :4900
```

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

Sleep dataframe:

```
##  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
```

```
Heartrate %>%
  select(Value) %>%
  summary()
```

Heartrate dataframe:

```
##      Value
##  Min.   : 36.00
##  1st Qu.: 63.00
##  Median : 73.00
##  Mean   : 77.33
##  3rd Qu.: 88.00
##  Max.   :203.00
```

```
Weight %>%
  select(WeightKg,
         BMI) %>%
  summary()
```

Weight dataframe:

```
##     WeightKg           BMI
##  Min.   : 52.60   Min.   :21.45
##  1st Qu.: 61.40   1st Qu.:23.96
##  Median : 62.50   Median :24.39
##  Mean   : 72.04   Mean   :25.19
##  3rd Qu.: 85.05   3rd Qu.:25.56
##  Max.   :133.50   Max.   :47.54
```

Key finding from these summar

Average steps per day is 7638.

People consumed 2304 calories a day.

Participants' average sleep time is 6.98 hours a day.

The average light activity (192.8 minutes) is considerably higher than very active (21.16 minutes) and fairly active (13.56 minutes).

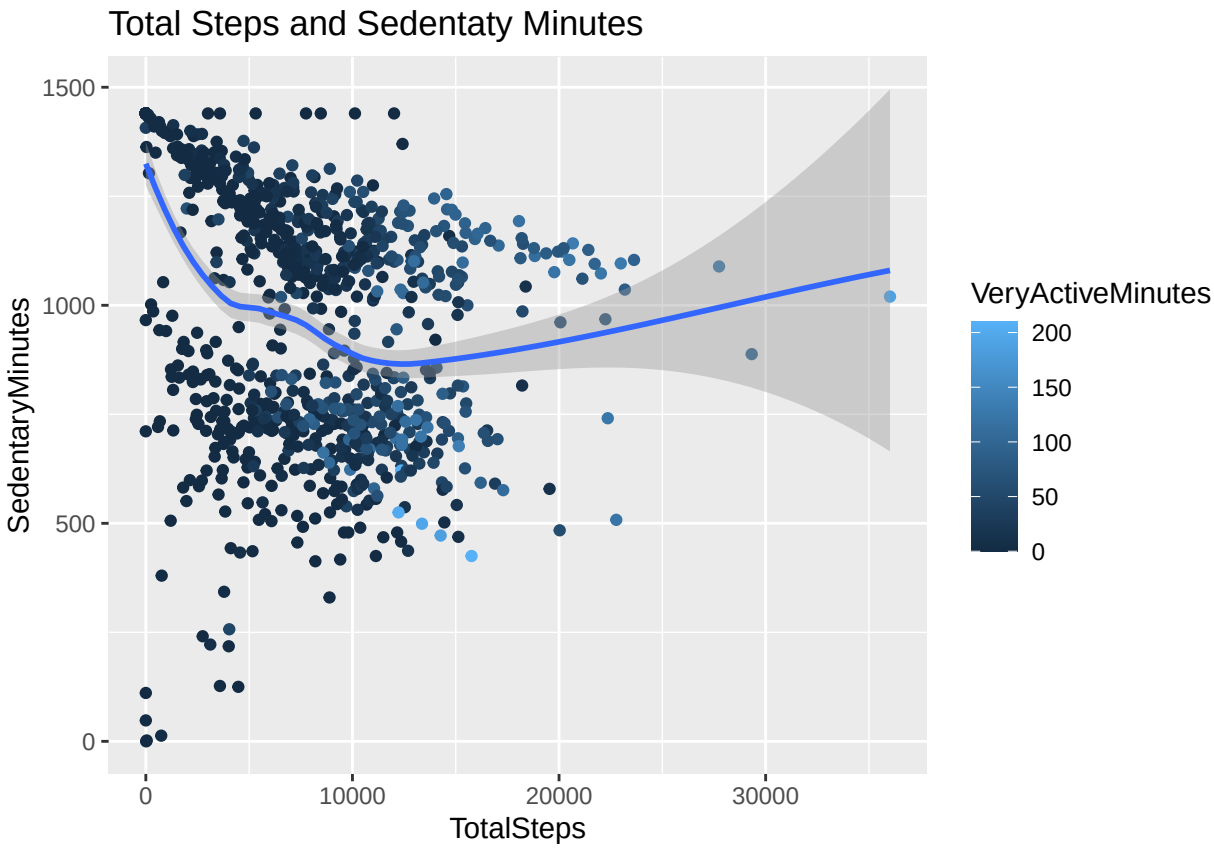
The number of sedentary time is very high more than 16 hours.

Data Visualization (Share Phase)

```
ggplot(data=Activity, aes(x=TotalSteps, y=SedentaryMinutes, color = VeryActiveMinutes)) +
  geom_point() +
  geom_smooth() +
  labs(title="Total Steps and Sedentaty Minutes")
```

Relationship between steps taken in a day and sedentary minutes:

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

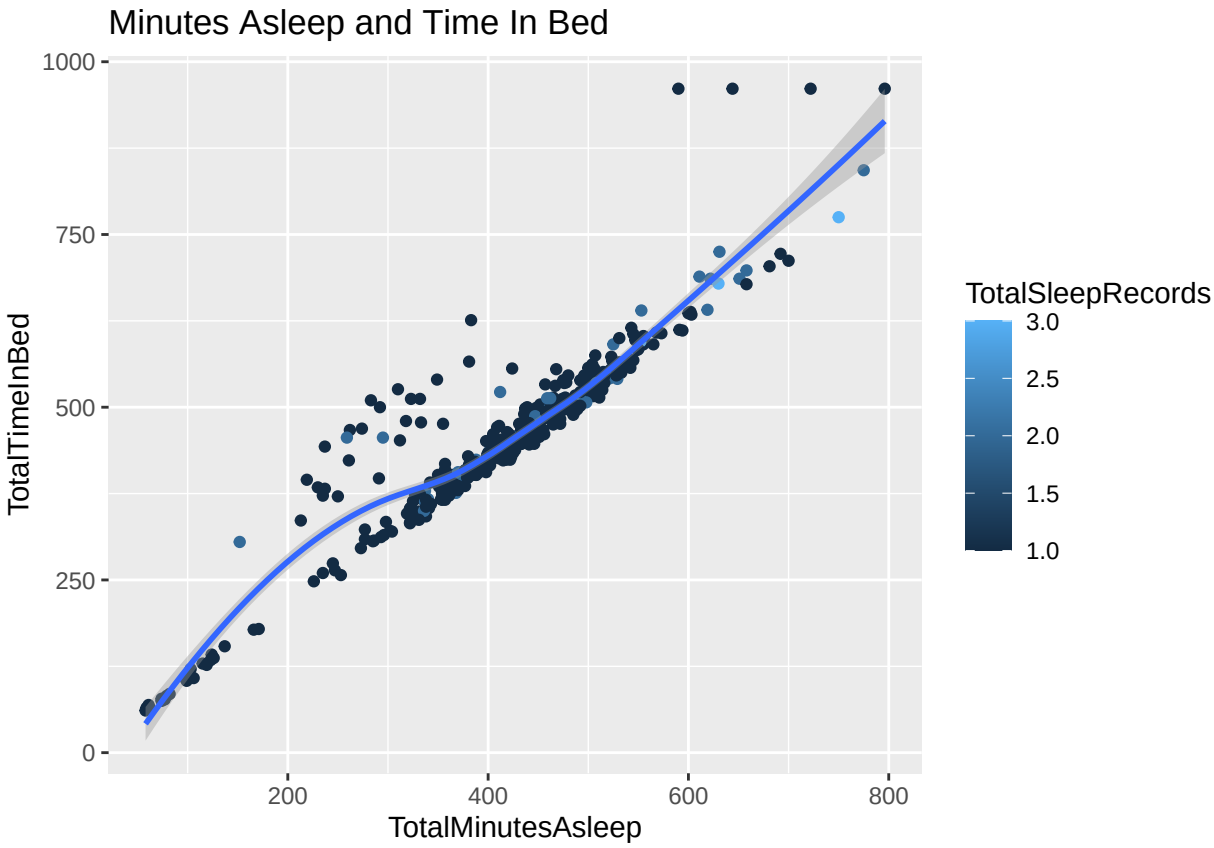


There is a negative relationship between two figures. The more sedentary time participants have, the less steps they take a day.

```
ggplot(data=Sleep, aes(x=TotalMinutesAsleep, y=TotalTimeInBed, color = TotalSleepRecords)) +
  geom_point() +
  geom_smooth() +
  labs(title="Minutes Asleep and Time In Bed")
```

The relationship between minutes asleep and time in bed

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

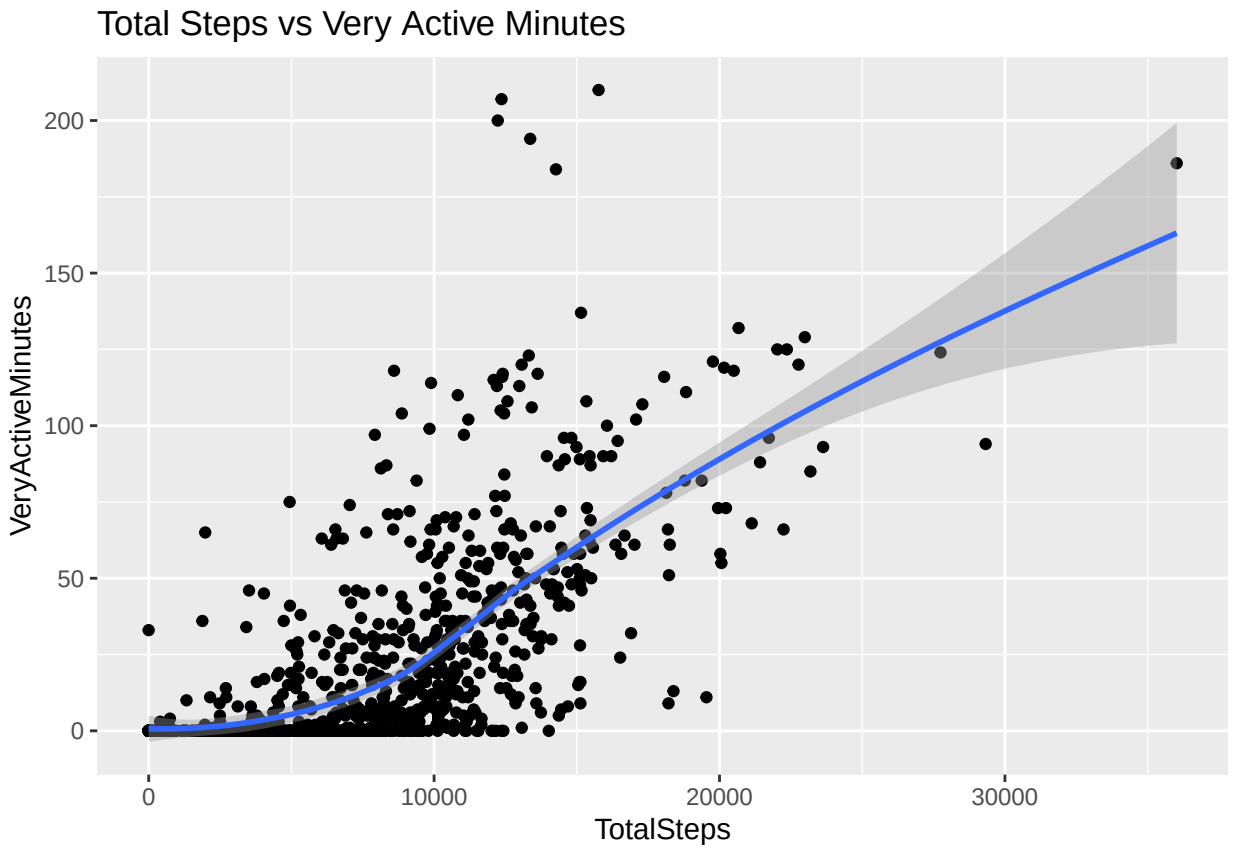


Time in bed and Total minutes asleep have a nearly perfect positive correlation.

```
ggplot(data=Activity, aes(x=TotalSteps, y=VeryActiveMinutes)) +
  geom_point() +
  geom_smooth() +
  labs(title="Total Steps vs Very Active Minutes ")
```

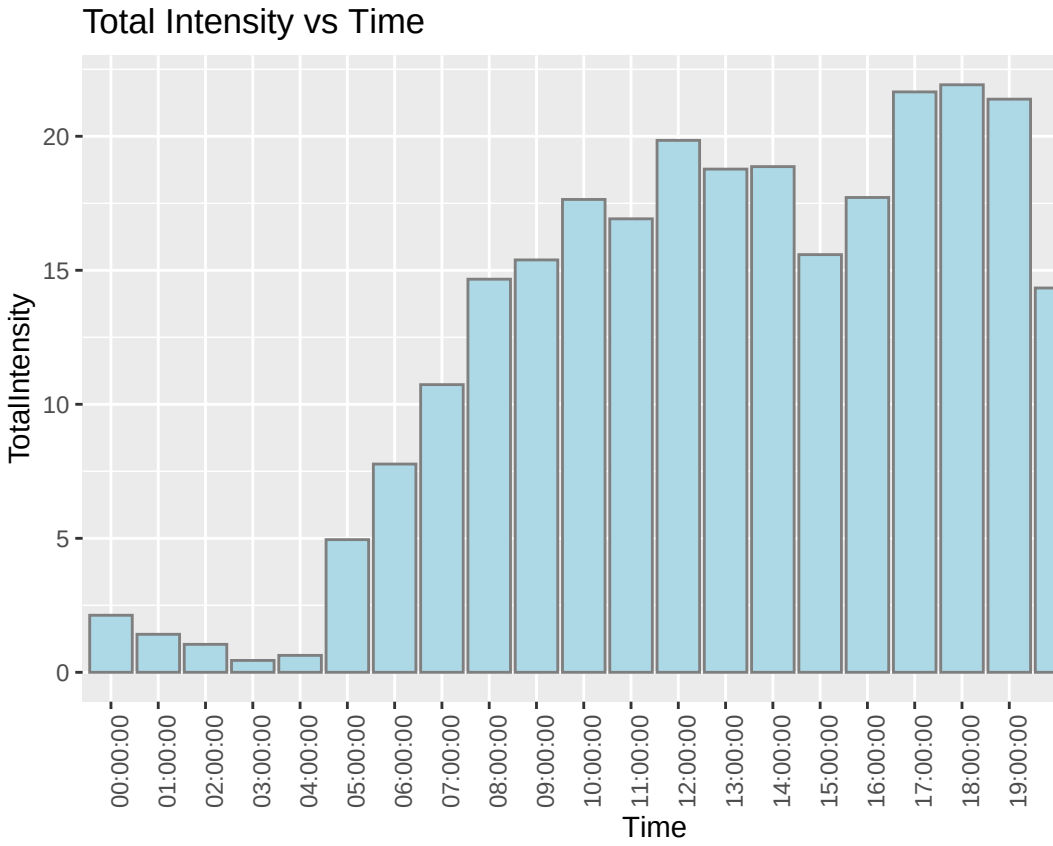
Relationship between Total Steps and Very Active Minutes:

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



The graph indicate that there is a positive correlation between Total Steps and Very Active Minutes a day by participants. That means the more steps they walk per day, the more likely the sleep time to increase.

```
ggplot(HourlyIntensities) +
  stat_summary(aes(x=Time,y=TotalIntensity),
               fun=mean,geom="bar",
               fill="lightblue",col="grey50") +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(title="Total Intensity vs Time ")
```



Most Intensity hour in date:

Look at the chart, I notice that participants use the app mostly after work from 5pm to 7pm.

Act Phase

Conclusion

After analyzing and visualizing FitBit Fitness Tracker Data set, I found some insights that would help Bellabeat improve their business:

- The negative relationship between sedentary time and Total Steps, and the number of sedentary time per day. This figure shows that the company needs to promote more about the benefit of the number of steps should take a, day especially for people who have high sedentary time.
- Time in bed and Total minutes asleep have a nearly perfect positive correlation. Through that, the company may consider adding more features to remind customers what time they need to go to bed and what sleep time they should take.
- The graph indicates that there is a positive correlation between Total Steps and Very Active Minutes a day by participants. That means the company can use it to encourage app users to do intense activity in order to increase the number of steps because 10,000 steps per day are good for people's health.
- Look at the chart, I notice that participants use the app mainly after work from 5 pm to 7 pm. The company app can use this figure to remind and motivate users to go for exercise.

Target Audiences

People who want to use the app remind or encourage to do exercise for keeping weight or for health purposes. Especially people who have a full-time job and want to run or walk after work.

Recommendation

The average number of steps per day is 7638, this figure needs to increase. According to CDC research, people should walk 10,000 steps a day. That is a number said to help reduce certain health conditions, such as high blood pressure and heart disease.

People consumed 2304 calories a day. Regarding NHS, the recommended daily calorie intake is 2,000 calories a day for women and 2,500 for men. We could use BMR method to calculate the calories need to consume a day. For example, Women: BMR = 655 + (9.6 x wt in kg) + (1.8 x ht in cm) - (4.7 x age in years). For a 30 year old female, 167.6 cm tall and weigh 54.5 kg, she need 1339 calories/day.

Participants' average sleep time is 6.98 hours a day, it is a little bit lower than the recommendation. For adults, getting less than seven hours of sleep a night on a regular basis has been linked with poor health, including weight gain, having a body mass index of 30 or higher, diabetes, high blood pressure, heart disease, stroke, and depression.

The average light activity is considerably higher than very and fairly active. A study by Dr. Maria Hagströmer confirms that replacing sedentary time with moderate- or higher-intensity physical activity has an even greater effect on reducing deaths linked to cardiovascular disease.

The number of sedentary time is very high more than 16 hours. The study by Dr.Maria Hagströmer also recommend that replacing sedentary time with just 10 minutes of either moderate- or vigorous-intensity activity each day was linked to a 38 percent reduced risk of death from cardiovascular disease, while 30 minutes per day was linked to a 77 percent reduction.