

Bellabeat Case Study

Eve Kudriavska

2025-03-26

Bellabeat Case Study

Introduction



Bellabeat is a high-tech manufacturer of health-focused products for women such as wellness devices which record sleep, stress, daily activities, water intake, menstrual cycle, ovulation, etc. Products are utilized through the application based on a membership program. Membership program provides guidance on nutrition, activity, sleep, health and beauty, and mindfulness through the App based on their lifestyle and goals. The company was founded in 2013.

Ask

Main stakeholders

Urška Sršen: Bellabeat's cofounder and Chief Creative Officer Sando Mur: Mathematician and Bellabeat's cofounder; key member of the Bellabeat executive team.

Business task

Comprehensive analysis of the usage of smart non- Bellabeat wellness devices by customers. Identify trends in the smart devices usage, which can be related to Bellabeat customers, and use the findings in building Bellabeat market strategy.

Deliverables

1. A clear summary of the business task
2. A description of all data sources used

3. Documentation of any cleaning or manipulation of data
4. A summary of data analysis
5. Supporting visualizations and key findings
6. Top high-level content recommendations based on the analysis.

Prepare

Data licensing, security, privacy and accessibility

The data we use is FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius (Owner)).

All participants gave their consent 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.

The source

Data source: <https://zenodo.org/records/53894#.X9oeh3Uzaao>
(<https://zenodo.org/records/53894#.X9oeh3Uzaao>).

Description: This dataset was generated by respondents to a distributed survey via Amazon Mechanical Turk between 03.12.2016-05.12.2016. The dataset consists of 29 files in csv format.

Data bias

The current data sample is quite small and might not be representative of the whole population.

Limitations

There is no personal data such as age, ethnicity, lifestyle, occupation, health conditions, geographical locations included in the sample.

Issues

The data was collected in 2016, therefore it might be outdated.

Prepare

Tools

For the initial dataset browsing I use Google sheets to quickly explore data. For the cleaning, preparing and analyzing data I will use R language in online RStudio via posit.cloud. The final report will be presented in RMarkdown file.

Exploration of data

I have checked the dataset in Google sheets and identified four main types of observations recorded by a tracker: sleep, calories, weight and steps. We will be looking into corresponding data tables, intensities table and activity_merged table for the further insights. I identified some issues with Date and Time, as well as zero values, which might screw the analysis. We will clean the dataset in the Studio.

Further exploration and cleaning

Loading the packages.

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats    1.0.0      ✓ stringr    1.5.1
## ✓ ggplot2    3.5.1      ✓ tibble     3.2.1
## ✓ lubridate  1.9.4      ✓ tidyr      1.3.1
## ✓ purrr      1.0.4
## — Conflicts — tidyverse_conflicts() —
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Loading the files.

```
activity <- read.csv("dailyActivity_merged.csv")
calories <- read.csv("hourlyCalories_merged.csv")
intensities <- read.csv("hourlyIntensities_merged.csv")
steps <- read.csv("hourlySteps_merged.csv")
sleep <- read.csv("sleepDay_merged.csv")
weight <- read.csv("weightLogInfo_merged.csv")
```

Getting the summary of the files to check for the initial issues and get some insights.

```
head(activity)
```

##	Id	ActivityDate	TotalSteps	TotalDistance	TrackerDistance
## 1	1503960366	4/12/2016	13162	8.50	8.50
## 2	1503960366	4/13/2016	10735	6.97	6.97
## 3	1503960366	4/14/2016	10460	6.74	6.74
## 4	1503960366	4/15/2016	9762	6.28	6.28
## 5	1503960366	4/16/2016	12669	8.16	8.16
## 6	1503960366	4/17/2016	9705	6.48	6.48
##	LoggedActivitiesDistance		VeryActiveDistance	ModeratelyActiveDistance	
## 1	0		1.88	0.55	
## 2	0		1.57	0.69	
## 3	0		2.44	0.40	
## 4	0		2.14	1.26	
## 5	0		2.71	0.41	
## 6	0		3.19	0.78	
##	LightActiveDistance		SedentaryActiveDistance	VeryActiveMinutes	
## 1	6.06		0	25	
## 2	4.71		0	21	
## 3	3.91		0	30	
## 4	2.83		0	29	
## 5	5.04		0	36	
## 6	2.51		0	38	
##	FairlyActiveMinutes		LightlyActiveMinutes	SedentaryMinutes	Calories
## 1	13		328	728	1985
## 2	19		217	776	1797
## 3	11		181	1218	1776
## 4	34		209	726	1745
## 5	10		221	773	1863
## 6	20		164	539	1728

Issues with the date format are identified. The date format is not consisted throughout the tables. For each table date and time will be separated and formatted consistently.

Process

Formatting Date and Time

Formatting the date/time to ““%m/%d/%Y”” in the activity table.

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

Checking results for activity table.

```
head(activity)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366 2016-04-12      13162          8.50          8.50
## 2 1503960366 2016-04-13      10735          6.97          6.97
## 3 1503960366 2016-04-14      10460          6.74          6.74
## 4 1503960366 2016-04-15       9762          6.28          6.28
## 5 1503960366 2016-04-16      12669          8.16          8.16
## 6 1503960366 2016-04-17       9705          6.48          6.48
## LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1              0              1.88              0.55
## 2              0              1.57              0.69
## 3              0              2.44              0.40
## 4              0              2.14              1.26
## 5              0              2.71              0.41
## 6              0              3.19              0.78
## LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1              6.06              0              25
## 2              4.71              0              21
## 3              3.91              0              30
## 4              2.83              0              29
## 5              5.04              0              36
## 6              2.51              0              38
## FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories      date
## 1              13              328              728      1985 04/12/16
## 2              19              217              776      1797 04/13/16
## 3              11              181             1218      1776 04/14/16
## 4              34              209              726      1745 04/15/16
## 5              10              221              773      1863 04/16/16
## 6              20              164              539      1728 04/17/16
```

Checking the calories table.

```
head(calories)
```

```
##           Id           ActivityHour Calories
## 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
```

Date and time need to be split and formatted.

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

Checking the Steps table.

```
head(steps)
```

```
##           Id           ActivityHour StepTotal
## 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
```

Activity Hour column should be split in Date and Time and formatted.

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

Checking date/time issues the sleep table.

```
head(sleep)
```

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

SleepDay should be split into Date and Time columns, and formatted accordingly.

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

Checking date/time issues in the intensities table.

```
head(intensities)
```

```
##           Id           ActivityHour TotalIntensity AverageIntensity
## 1 1503960366 4/12/2016 12:00:00 AM             20         0.333333
## 2 1503960366 4/12/2016 1:00:00 AM              8         0.133333
## 3 1503960366 4/12/2016 2:00:00 AM              7         0.116667
## 4 1503960366 4/12/2016 3:00:00 AM              0         0.000000
## 5 1503960366 4/12/2016 4:00:00 AM              0         0.000000
## 6 1503960366 4/12/2016 5:00:00 AM              0         0.000000
```

Activity hour should to be split in Date and Time, formatted.

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

Checking time/date issues in the weight table.

```
head(weight)
```

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

Date column needs to be split in Time and Date, formatted.

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

Removing zero values and blanks

We keep cleaning data. Lets get rid of blanks and zero values in some columns of the tables. Those values can influence summary of the tables, and further scew the analysis.

```
head(activity)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366 2016-04-12      13162          8.50          8.50
## 2 1503960366 2016-04-13      10735          6.97          6.97
## 3 1503960366 2016-04-14      10460          6.74          6.74
## 4 1503960366 2016-04-15       9762          6.28          6.28
## 5 1503960366 2016-04-16      12669          8.16          8.16
## 6 1503960366 2016-04-17       9705          6.48          6.48
## LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1              0              1.88              0.55
## 2              0              1.57              0.69
## 3              0              2.44              0.40
## 4              0              2.14              1.26
## 5              0              2.71              0.41
## 6              0              3.19              0.78
## LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1              6.06              0              25
## 2              4.71              0              21
## 3              3.91              0              30
## 4              2.83              0              29
## 5              5.04              0              36
## 6              2.51              0              38
## FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories      date
## 1              13              328              728      1985 04/12/16
## 2              19              217              776      1797 04/13/16
## 3              11              181             1218      1776 04/14/16
## 4              34              209              726      1745 04/15/16
## 5              10              221              773      1863 04/16/16
## 6              20              164              539      1728 04/17/16
```

Cleaning the activity table from zero and blanks in TotalSteps column.

```
activity_cleaned <- activity %>%
  filter(!is.na(activity$TotalSteps) & activity$TotalSteps != 0)
```

Analyse

Exploring cleaned dataset, identifying trends and insights.

Relationship between features

Exploring how many Unique ID use the trackers, and how they use the tracker.

```
n_distinct(activity$Id)
```

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

```
n_distinct(calories$Id)
```

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

```
n_distinct(steps$Id)
```



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

```
n_distinct(intensities$Id)
```

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

```
n_distinct(sleep$Id)
```

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

```
n_distinct(weight$Id)
```

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

The results are 33, 33, 33, 33, 24,8. Therefore we can conclude that we have 33 unique IDs for activity, steps, calories and intensities. All these observations are made automatically by a tracker. 24 unique IDs use sleep feature, which is also recorded automatically while users are wearing the device. And we know that 8 Unique Ids uses weight log feature, which in recorded manually by a user.

Checking amount of observation days per each table.

```
n_distinct(activity$date)
```

```
## [1] 31
```

```
n_distinct(calories$date)
```

```
## [1] 31
```

```
n_distinct(steps$date)
```

```
## [1] 31
```

```
n_distinct(intensities$date)
```

```
## [1] 31
```

```
n_distinct(sleep$date)
```

```
## [1] 31
```

```
n_distinct(weight$date)
```

```
## [1] 31
```

Each table has 31 days of observations, which proves the consistency of the record.

Therefore, we can conclude that the dataset contains observations of 33 users over 31 days.

Visuals

Lets create some visualisations to compare the usage of sleep, steps and weight features.

```
install.packages("ggvenn")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'  
## (as 'lib' is unspecified)
```

```
library(ggplot2)  
library(ggvenn)
```

```
## Loading required package: grid
```

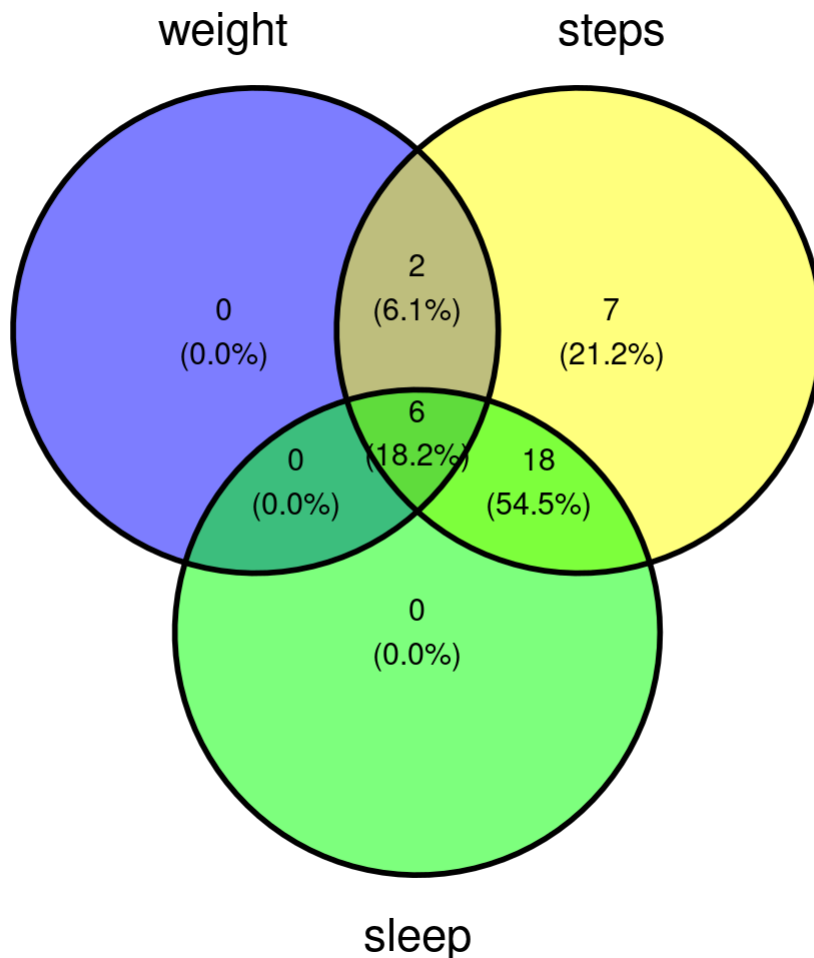
```
library(grid)
```

Creating Id list.

```
id_list <- list(  
  weight = weight$Id,  
  steps = steps$Id,  
  sleep = sleep$Id  
)
```

Creating a Venn Diagram

```
ggvenn(id_list)
```



The Venn diagram revealed that 33 lds use step count feature, 8 out of 33 also use weight monitor feature, 24 out of 33 uses sleep monitoring feature. Only 6 out of 33 uses all three feature. Venn diagram shows that 2/33 uses both steps and weight functions, 18/33 uses. We already can identify some trends emerging. The most popular feature is step count, which is automatically recorded. Sleep monitor is used less, but is also recorded automatically.

Only 24 users wear their monitor at night. And, we can clearly see that only 8 users monitor their weight, which is recorded manually. We might speculate that the reason for this is the way weight is logged in. It might not be easy or user friendly. Here is a possible gap in the market, which Bellabeat might want to explore.

Let's create a bar chart to further analyse the relationship between these features.

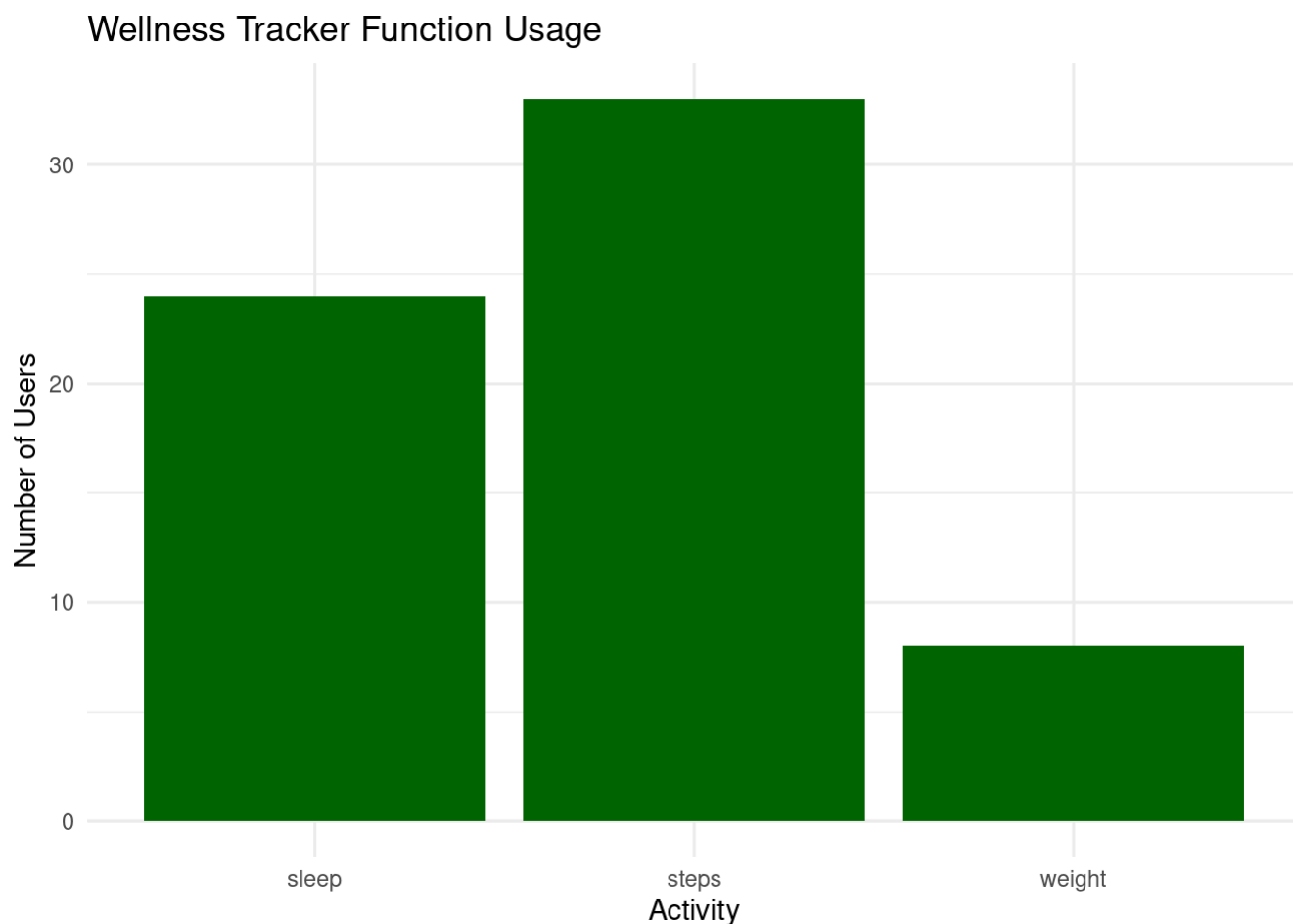
```
weight_count <- n_distinct(weight$Id)
steps_count <- n_distinct(steps$Id)
sleep_count <- n_distinct(sleep$Id)
```

Creating a data frame for visualization

```
usage_summary <- data.frame(
  Activity = c("weight", "steps", "sleep"),
  UserCount = c(weight_count, steps_count, sleep_count)
)
```

Create a bar chart to compare how many lds are using sleep, steps and weight functions

```
ggplot(usage_summary, aes(x = Activity, y = UserCount)) +
  geom_bar(stat = "identity", fill = "darkgreen") +
  labs(
    title = "Wellness Tracker Function Usage",
    x = "Activity",
    y = "Number of Users"
  ) +
  theme_minimal()
```



The bar chart shows that steps count feature is the most popular, and all 33 IDs are using it. The sleep feature is the second popular. And weight is the last maybe due to the manual or unfriendly input of data.

Checking further summary of the tables.

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

##	TotalSteps	TotalDistance	SedentaryMinutes	Calories
##	Min. : 4	Min. : 0.00	Min. : 0.0	Min. : 52
##	1st Qu.: 4923	1st Qu.: 3.37	1st Qu.: 721.5	1st Qu.: 1856
##	Median : 8053	Median : 5.59	Median : 1021.0	Median : 2220
##	Mean : 8319	Mean : 5.98	Mean : 955.8	Mean : 2361
##	3rd Qu.: 11092	3rd Qu.: 7.90	3rd Qu.: 1189.0	3rd Qu.: 2832
##	Max. : 36019	Max. : 28.03	Max. : 1440.0	Max. : 4900

From the summary of the activity_cleaned we can divide all users into 3 groups, 1st group (lightly active) with average steps of 4923/day, 2nd group (fairly active) with average steps of 8053/day, 3rd group (very active) with average steps of 11092/day. The average amount of steps for all users is 8053. The average sedentary minutes is 995.8 which equates to almost 16 hours per day.

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

```
## VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes
## Min. : 0.00 Min. : 0.00 Min. : 0.0
## 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.:146.5
## Median : 7.00 Median : 8.00 Median :208.0
## Mean : 23.02 Mean : 14.78 Mean :210.0
## 3rd Qu.: 35.00 3rd Qu.: 21.00 3rd Qu.:272.0
## Max. :210.00 Max. :143.00 Max. :518.0
```

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

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

```
summary(steps)
```

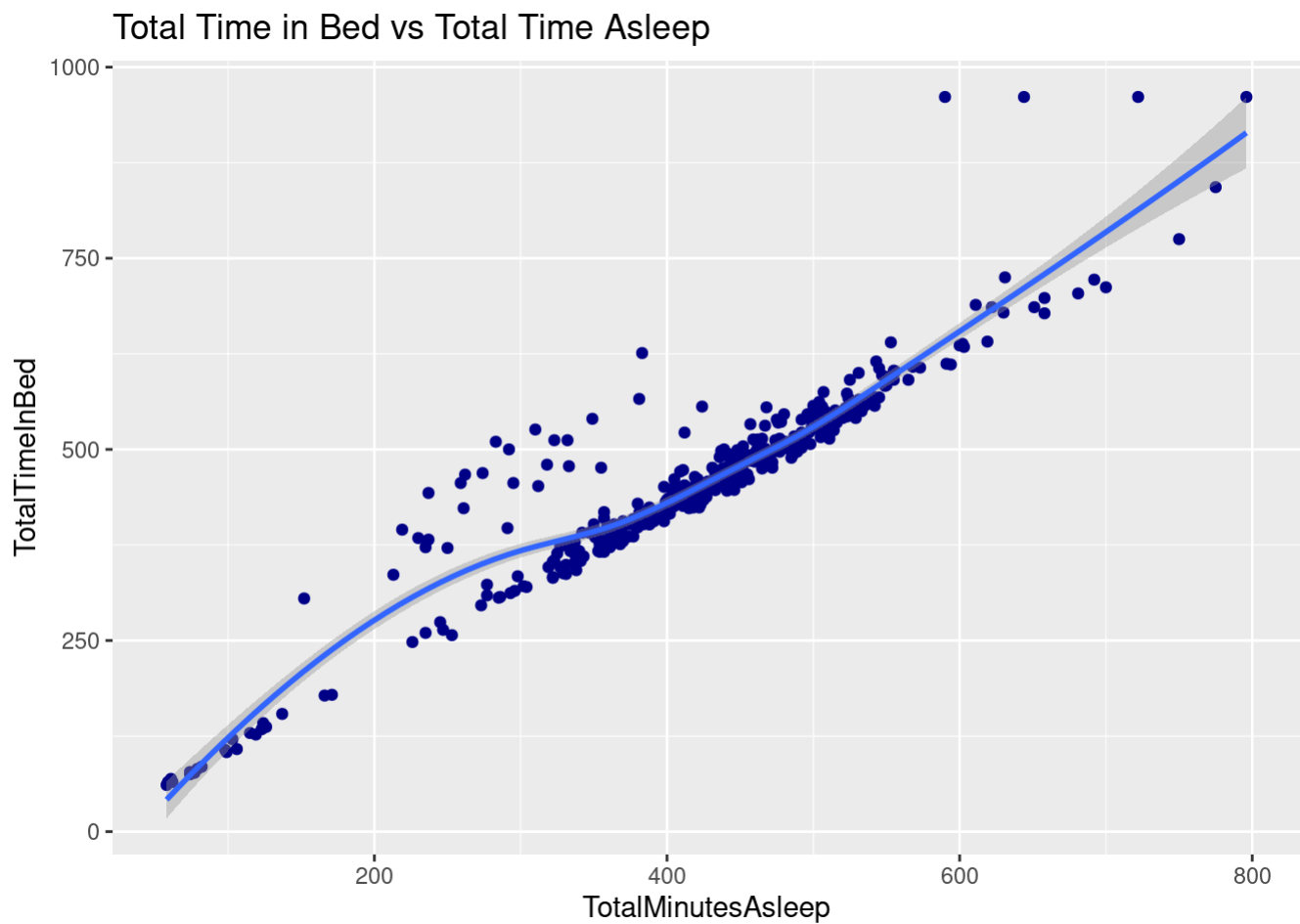
```
## Id ActivityHour StepTotal
## Min. :1.504e+09 Min. :2016-04-12 00:00:00.00 Min. : 0.0
## 1st Qu.:2.320e+09 1st Qu.:2016-04-19 01:00:00.00 1st Qu.: 0.0
## Median :4.445e+09 Median :2016-04-26 06:00:00.00 Median : 40.0
## Mean :4.848e+09 Mean :2016-04-26 11:46:42.58 Mean : 320.2
## 3rd Qu.:6.962e+09 3rd Qu.:2016-05-03 19:00:00.00 3rd Qu.: 357.0
## Max. :8.878e+09 Max. :2016-05-12 15:00:00.00 Max. :10554.0
## time date
## Length:22099 Length:22099
## Class :character Class :character
## Mode :character Mode :character
##
##
##
```

```
view(steps)
```

Checking Total time asleep vs Total time in bed for the insights. We can see that average total time asleep is 433 minutes or 7 hours 13 minutes. Therefore, since we know the correlation between time in bed and sleeping time, creating a 'Time to go to bed' reminder function might be useful to increase user's sleeping

time.

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



We can see a positive correlations between Total time in bed vs Total time asleep.

Merging data

Lets merge sleep and activity by ID and Date to create comprehensice visualisations.

```
merged_data <- merge(sleep, activity_cleaned, by=c('Id', 'date'))
head(merged_data)
```

```

##           Id      date    SleepDay TotalSleepRecords TotalMinutesAsleep
## 1 1503960366 04/12/16 2016-04-12                1             327
## 2 1503960366 04/13/16 2016-04-13                2             384
## 3 1503960366 04/15/16 2016-04-15                1             412
## 4 1503960366 04/16/16 2016-04-16                2             340
## 5 1503960366 04/17/16 2016-04-17                1             700
## 6 1503960366 04/19/16 2016-04-19                1             304
##   TotalTimeInBed      time ActivityDate TotalSteps TotalDistance TrackerDistance
## 1           346 00:00:00   2016-04-12     13162           8.50           8.50
## 2           407 00:00:00   2016-04-13     10735           6.97           6.97
## 3           442 00:00:00   2016-04-15      9762           6.28           6.28
## 4           367 00:00:00   2016-04-16     12669           8.16           8.16
## 5           712 00:00:00   2016-04-17      9705           6.48           6.48
## 6           320 00:00:00   2016-04-19     15506           9.88           9.88
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0                1.88                0.55
## 2                        0                1.57                0.69
## 3                        0                2.14                1.26
## 4                        0                2.71                0.41
## 5                        0                3.19                0.78
## 6                        0                3.53                1.32
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                      0                25
## 2                4.71                      0                21
## 3                2.83                      0                29
## 4                5.04                      0                36
## 5                2.51                      0                38
## 6                5.03                      0                50
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                13                328                728       1985
## 2                19                217                776       1797
## 3                34                209                726       1745
## 4                10                221                773       1863
## 5                20                164                539       1728
## 6                31                264                775       2035

```

Checking insights for steps

```
head(steps)
```

```

##           Id      ActivityHour StepTotal      time      date
## 1 1503960366 2016-04-12 00:00:00        373 00:00:00 04/12/16
## 2 1503960366 2016-04-12 01:00:00        160 01:00:00 04/12/16
## 3 1503960366 2016-04-12 02:00:00        151 02:00:00 04/12/16
## 4 1503960366 2016-04-12 03:00:00          0 03:00:00 04/12/16
## 5 1503960366 2016-04-12 04:00:00          0 04:00:00 04/12/16
## 6 1503960366 2016-04-12 05:00:00          0 05:00:00 04/12/16

```

Organising steps table, removing blanks, grouping steps by time.

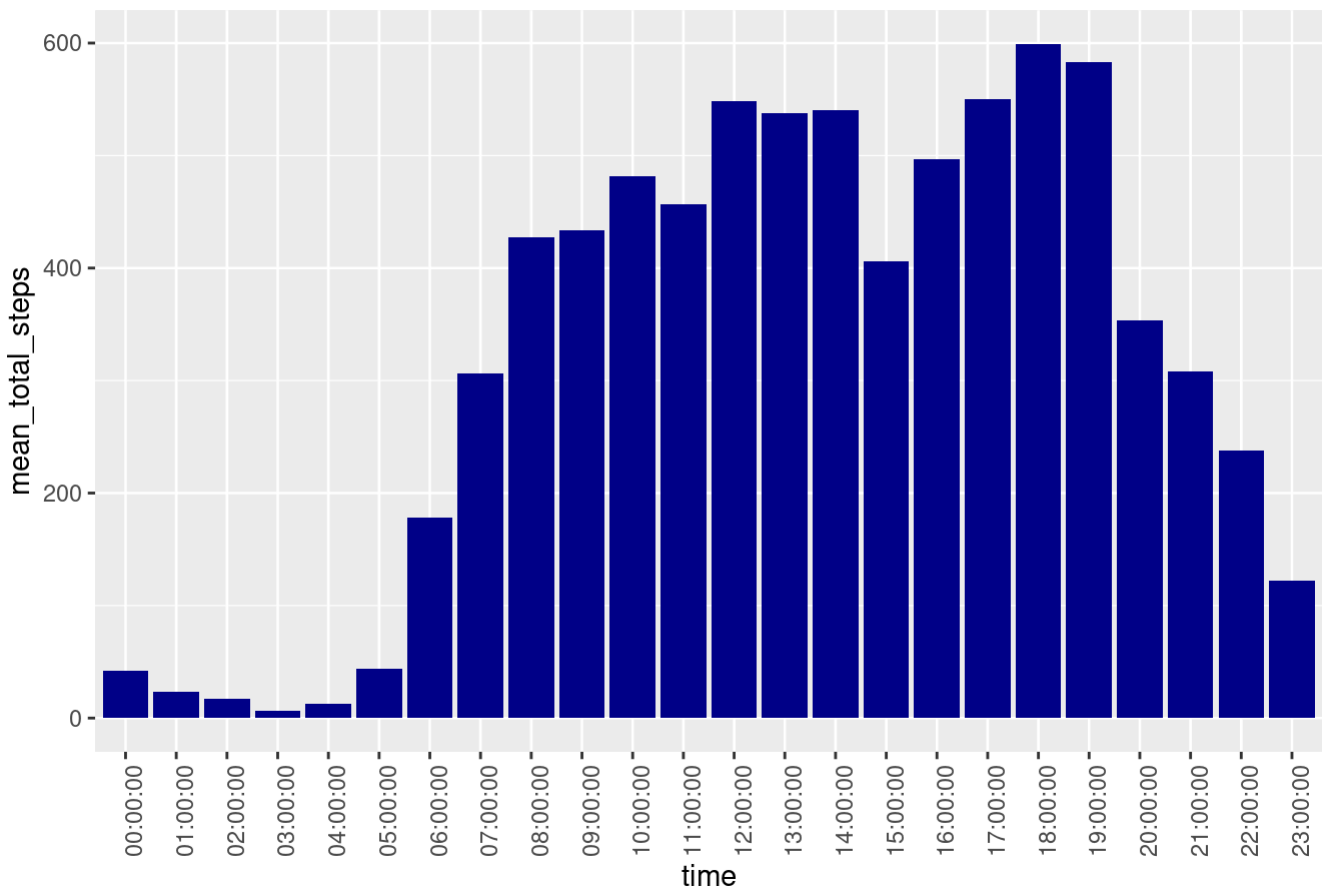
```
steps_new <- steps %>%
  select(StepTotal,time)%>%
  group_by(time) %>%
  drop_na() %>%
  summarise(mean_total_steps = mean(StepTotal))
```

Creating histogram to see mean_steps during the day.

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

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

Average Total Steps vs. Time



Let's take a look at intensities table to cross check intensities level vs time of the day. We will create a bar chart to illustrate this relationship.

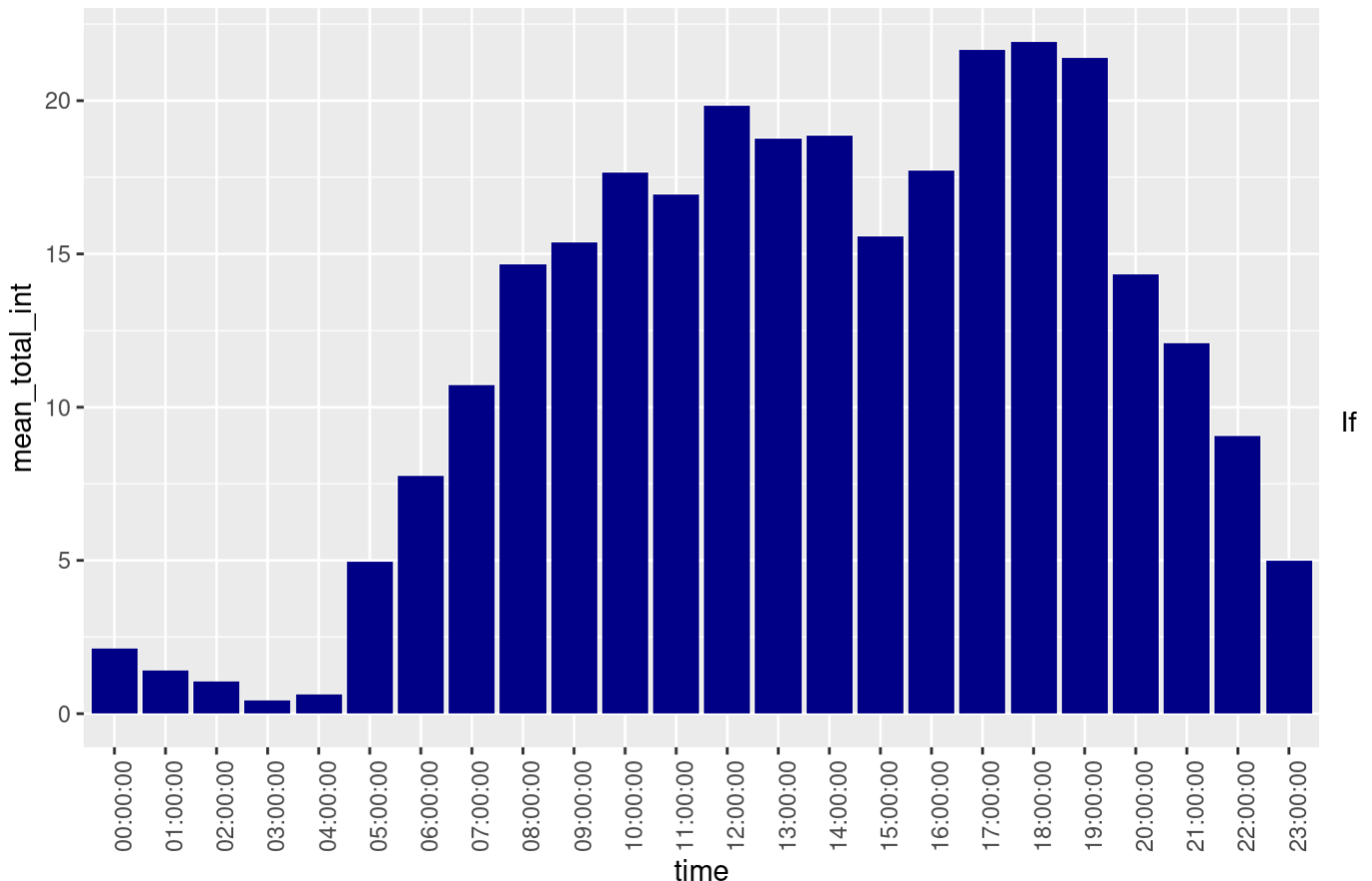
```
intensities_new <- intensities %>%
  group_by(time) %>%
  drop_na() %>%
  summarise(mean_total_int = mean(TotalIntensity))
```



```
ggplot(data=intensities_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")
```

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

Average Total Intensity vs. Time



we examine two bar charts we can conclude that they mainly overlap, peaking between 5pm and 8 pm, with a significant increase during the lunch hours. It might mean that most of the users are employed and have relatively sedentary jobs like office jobs, considering the amount of sedentary hours. The users tend to do more steps during lunch and after 5 pm, which further confirms our theory regarding their occupations and lifestyle.

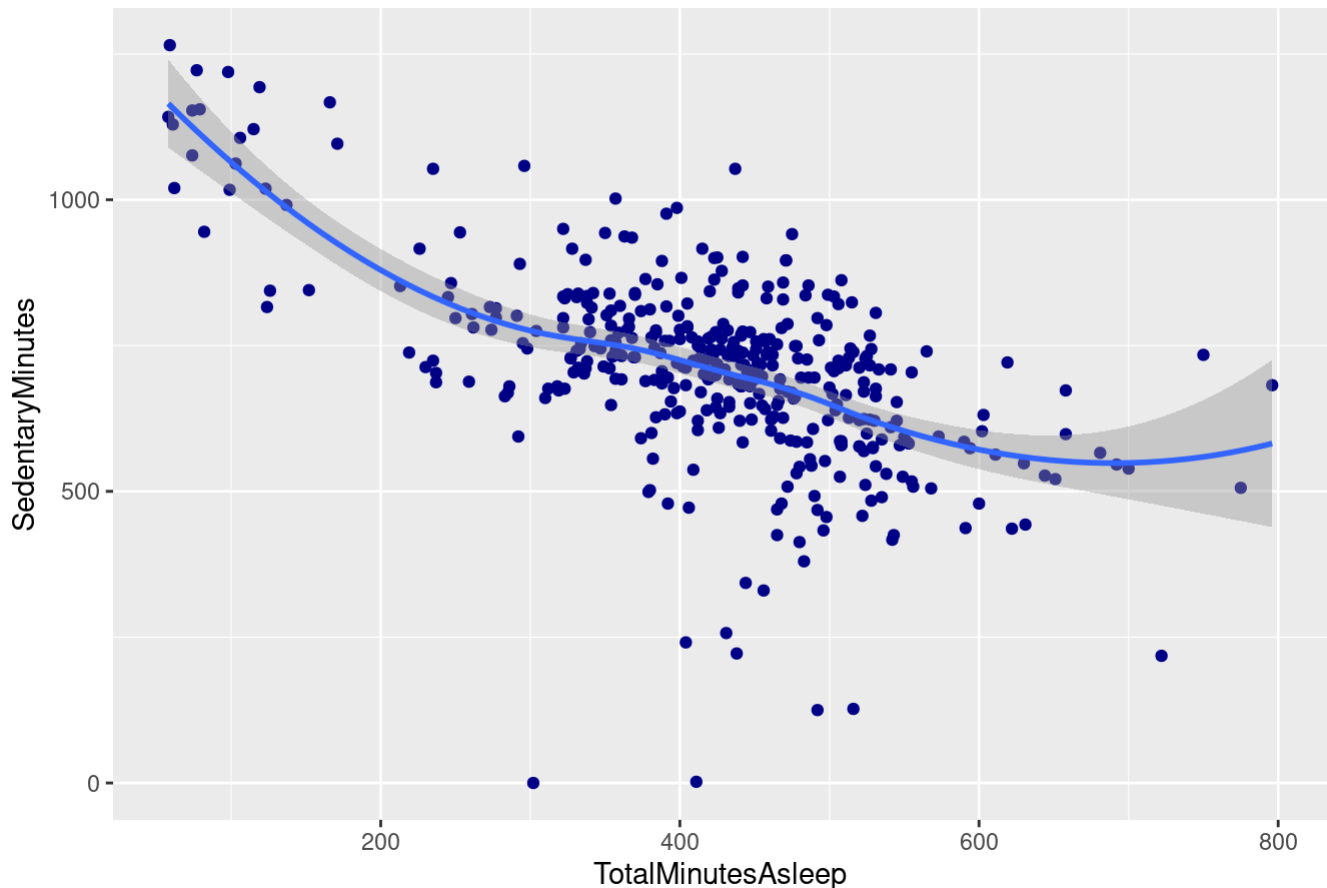
Let's explore quality of sleep our users are getting. Is there a correlation between sedentary time and sleep duration?

Creating a line chart to explore correlation between Sedentary Minutes and Total minutes asleep

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

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

Minutes Asleep vs. Sedentary Minutes



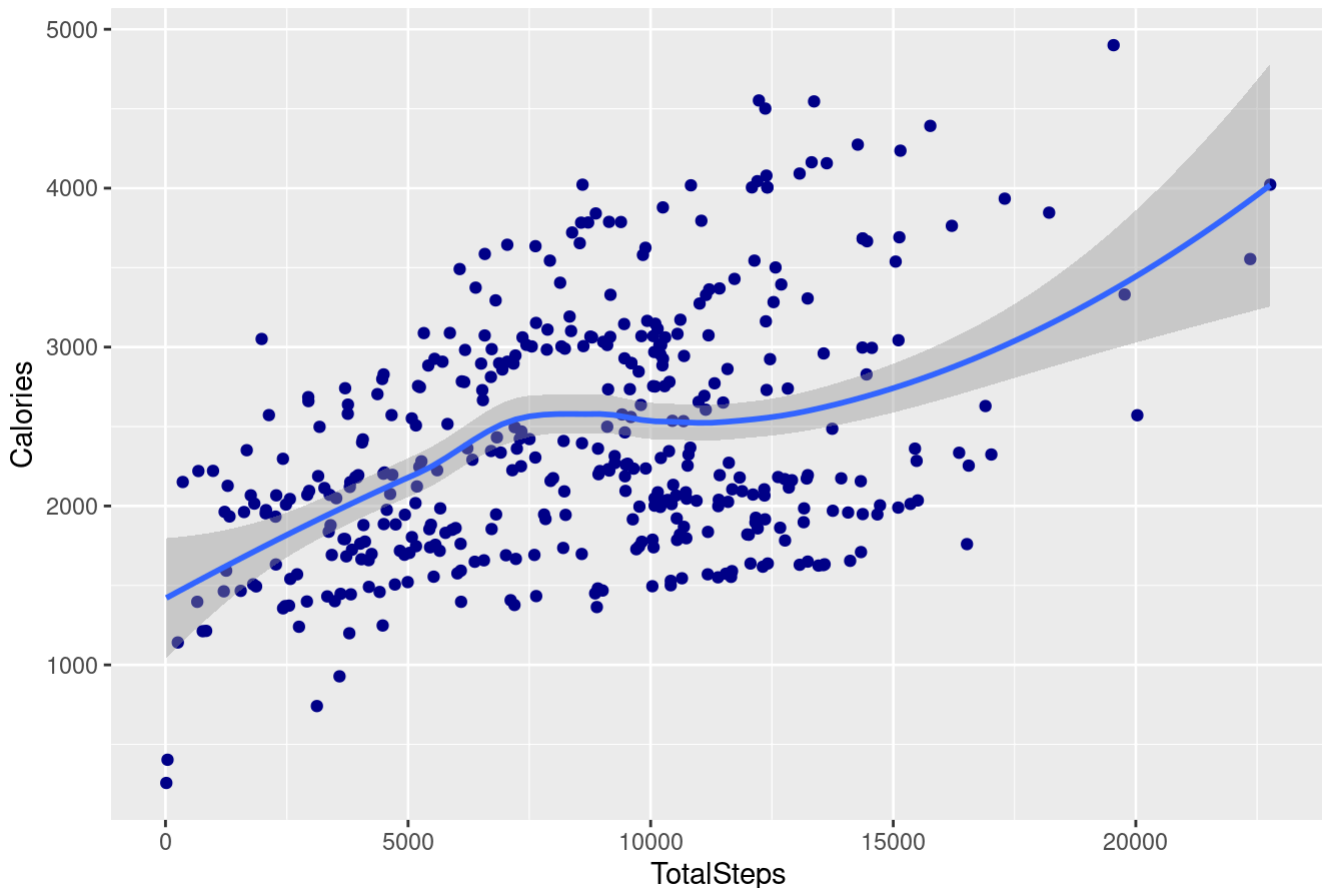
From the “Minutes Asleep vs Sedentary Minutes” we can clearly see that there is a negative correlation between sedentary time and sleeping time. This insight might be used in the marketing strategy to target people who want improve their sleep by reducing sedentary time. Particularly people who spend a lot of time in front of the computer or desk and are not aware of the impact it has on their sleep.

Lets look into relationship between Calories and Total steps.

```
ggplot(data=merged_data, aes(x=TotalSteps, y=Calories)) +  
  geom_point(color='darkblue') + geom_smooth() +  
  labs(title="Total Steps vs. Calories")
```

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

Total Steps vs. Calories



We can see some positive correlation, meaning the more steps are taken the more calories are burnt. Again that might be an incentive for people who would like to watch their weight, improve sleep and reduce sedentary time. We already know that there is a gap in the market for weight watching feature.

Summary

We have analysed dataset containing observations of 33 participants over duration of 31 days. We have looked at the daily activity, intensities, sleep, steps, calories tables to get comprehensive insights and to understand trends of the device usage.

Key findings

- We have concluded that there are three main features that consumers use the wellness tracker to monitor: daily steps, sleep and weight(calories are calculated automatically with total steps count). However, even though the device is able to monitor all three main activities, the features are not used equally.
- We know that all 33 users monitor steps, only 24 monitor sleep, and just 8 users monitor their weight. These reveals a possible gap in the market and missed opportunity to fully engage a consumer into utilizing the device as only 18.2 % of participants use these three available functions.
- We have established that average sedentary time is about 16 hours a day, which is 2/3 of a day.
- The most active times of the day are lunch hours and time between 5 pm and 8 pm. This suggests that our participants are likely to have sitting job, office jobs etc. This insight might help to develop daily activity routine suggestions for employed people who have sedentary jobs but generating a reminders and encouraging them complete their streaks and reach steps targets, etc.
- We have established that there is a negative correlation between amount if sedentary minutes and time asleep. This suggests that the more sedentary minutes a person spends a day the less sleep they get.

- We have established a positive relationship between time spent in bed and total time asleep. This insight suggest that users might benefit from a device feature to remind them to go to bed.
- We have established that the more steps users take a day the more calories they burn, which is highly predictable and logical.

Recommendations

Based on this particular data set analysis, I can recommend following for the marketing campaign:

- Focus on a holistic marketing approach, which delivers a comprehensive message about users holistic well being, e.g how sleep quality is connected to daily steps or daily activity, the connection between calories burning and weight, etc.
- Optimize the usability of weight input feature, for example utilizing AI voice recognition technology for the weight input. Make weight input process as easy as possible, including reminder feature to record the weight daily.
- Focus on developing a daily activity program for people with sedentary 9-5 jobs, which contains activity streaks, reminders to take a break and have a walk/ stretch, reminder for water intakes, etc.
- Optimize the device design to provide the maximum comfort during the night to encourage more customers to monitor their sleep.
- Develop a feature for daily challenges such as daily steps count to increase calories burning, improve sleep and overall health.
- Generate weekly/monthly activity report, that focus on the positive outcomes. For example, the report focus on improvements in sleep patterns, the increase of daily steps, or reaching target weight.