Bellabeat Case Study

2023-10-02

# **Case Study 2**

## **1.Summary**

Bellabeat is a company that specializes in manufacturing high-tech smart products for women’s health. Their range of smart devices collects data on various aspects such as activity, sleep, stress, and reproductive health. The goal of Bellabeat is to empower women by providing them with insights and knowledge about their own health and habits.

In this case, our focus is on analyzing the fitness data collected by Bellabeat’s smart devices. We aim to identify potential growth opportunities for the company. Specifically, we will be examining the Bellabeat app, which serves as a platform for users to access their health data. Through the app, users can track their activity, sleep patterns, stress levels, menstrual cycle, and mindfulness habits. This valuable data enables users to gain a deeper understanding of their current habits and make informed decisions to improve their overall health and well-being. Additionally, the Bellabeat app seamlessly integrates with other smart wellness products offered by the company.

## **2. Ask**

### **2.1 Business Task**

The objective of this analysis is to identify patterns and trends in consumer usage of non-Bellabeat smart devices, in order to inform and enhance Bellabeat’s marketing strategy and drive growth in the smart health products market for women.

### **2.2 Stakeholders**

**Urška Sršen**: Bellabeat’s cofounder and Chief Creative Officer **Sando Mur**: Mathematician and Bellabeat’s cofounder; key member of the Bellabeat executive team **Bellabeat marketing analytics team**: A team of data analysts responsible for collecting, analyzing, and reporting data that helps guide Bellabeat’s marketing strategy.

## **3. Prepare**

### **3.1 Dataset used:**

The data source used for our case study is FitBit Fitness Tracker Data. This dataset is stored in Kaggle and was made available through Mobius.

### **3.2 Accessibility and privacy of data:**

Verifying the metadata of our dataset we can confirm it is open-source. The owner has dedicated the work to the public domain by waiving all of his or her rights to the work worldwide under copyright law, including all related and neighboring rights, to the extent allowed by law. You can copy, modify, distribute and perform the work, even for commercial purposes, all without asking permission.

### **3.3 Information about our dataset:**

These datasets were generated by respondents to a distributed survey via Amazon Mechanical Turk between 03.12.2016-05.12.2016. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. Variation between output represents use of different types of Fitbit trackers and individual tracking behaviors / preferences.

### **3.4 Data Organization and verification:**

Available to us are 18 CSV documents. Each document represents different quantitative data tracked by Fitbit. The data is considered long since each row is one time point per subject, so each subject will have data in multiple rows.Every user has a unique ID and different rows since data is tracked by day and time.

I mainly used RStudio for cleaning and preparation.

### **3.5 Data Credibility and Integrity:**

Due to the limitation of size (30 users) and not having any demographic information we could encounter a sampling bias. Additionally, Bellabeat’s target demographic is women. We are not sure if the sample includes women, this is a bias especially since women’s health data may differ. Another problem we would encounter is that the dataset is not current and also the time limitation of the survey (2 months long). That is why we will give our case study an operational approach.

## **4. Process**

This analysis is done in Rstudio using R programming language. R is very versatile an has the ability to clean, analyze and visualize the Data.

### **4.1 Installing packages and opening libraries**

We will use the following packages for our analysis:

install.packages("tidyverse")

## 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)

install.packages("ggpubr")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.3'  
## (as 'lib' is unspecified)

install.packages("here")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.3'  
## (as 'lib' is unspecified)

install.packages("lubridate")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.3'  
## (as 'lib' is unspecified)

install.packages("ggrepel")

## Installing package into '/cloud/lib/x86\_64-pc-linux-gnu-library/4.3'  
## (as 'lib' is unspecified)

Opening the libraries:

library(ggpubr)

## Loading required package: ggplot2

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.0

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(here)

## here() starts at /cloud/project

library(skimr)  
library(janitor)

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

library(lubridate)  
library(ggrepel)

### **4.2 Importing datasets**

I will upload the datasets that will help me answer the business task. On my analysis I will focus on the following datasets:

* dailyActivity
* sleepDay
* dailyIntensity
* dailyCalories
* hourlySteps

daily\_activity <- 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\_intensity <- read\_csv("dailyIntensities\_merged.csv")

## Rows: 940 Columns: 10  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): ActivityDay  
## dbl (9): Id, SedentaryMinutes, LightlyActiveMinutes, FairlyActiveMinutes, Ve...  
##   
## 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.

hourly\_steps <- read\_csv("hourlySteps\_merged.csv")

## Rows: 22099 Columns: 3  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): ActivityHour  
## dbl (2): Id, StepTotal  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

daily\_calories <- read\_csv("dailyCalories\_merged.csv")

## Rows: 940 Columns: 3  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): ActivityDay  
## dbl (2): Id, Calories  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

### **4.3 Getting to know the data**

Now we’ll get a little sneak peak at our chosen datasets.

head(daily\_activity)

## # A tibble: 6 × 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>

str(daily\_activity)

## spc\_tbl\_ [940 × 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>

head(daily\_intensity)

## # A tibble: 6 × 10  
## Id ActivityDay SedentaryMinutes LightlyActiveMinutes FairlyActiveMinutes  
## <dbl> <chr> <dbl> <dbl> <dbl>  
## 1 1.50e9 4/12/2016 728 328 13  
## 2 1.50e9 4/13/2016 776 217 19  
## 3 1.50e9 4/14/2016 1218 181 11  
## 4 1.50e9 4/15/2016 726 209 34  
## 5 1.50e9 4/16/2016 773 221 10  
## 6 1.50e9 4/17/2016 539 164 20  
## # ℹ 5 more variables: VeryActiveMinutes <dbl>, SedentaryActiveDistance <dbl>,  
## # LightActiveDistance <dbl>, ModeratelyActiveDistance <dbl>,  
## # VeryActiveDistance <dbl>

str(daily\_intensity)

## spc\_tbl\_ [940 × 10] (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 ...  
## $ ActivityDay : chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...  
## $ SedentaryMinutes : num [1:940] 728 776 1218 726 773 ...  
## $ LightlyActiveMinutes : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...  
## $ FairlyActiveMinutes : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...  
## $ VeryActiveMinutes : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...  
## $ SedentaryActiveDistance : num [1:940] 0 0 0 0 0 0 0 0 0 0 ...  
## $ LightActiveDistance : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...  
## $ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...  
## $ VeryActiveDistance : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. Id = col\_double(),  
## .. ActivityDay = col\_character(),  
## .. SedentaryMinutes = col\_double(),  
## .. LightlyActiveMinutes = col\_double(),  
## .. FairlyActiveMinutes = col\_double(),  
## .. VeryActiveMinutes = col\_double(),  
## .. SedentaryActiveDistance = col\_double(),  
## .. LightActiveDistance = col\_double(),  
## .. ModeratelyActiveDistance = col\_double(),  
## .. VeryActiveDistance = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

head(daily\_sleep)

## # A tibble: 6 × 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

str(daily\_sleep)

## spc\_tbl\_ [413 × 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 12:00:00 AM" "4/16/2016 12:00:00 AM" ...  
## $ 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>

head(hourly\_steps)

## # A tibble: 6 × 3  
## Id ActivityHour StepTotal  
## <dbl> <chr> <dbl>  
## 1 1503960366 4/12/2016 12:00:00 AM 373  
## 2 1503960366 4/12/2016 1:00:00 AM 160  
## 3 1503960366 4/12/2016 2:00:00 AM 151  
## 4 1503960366 4/12/2016 3:00:00 AM 0  
## 5 1503960366 4/12/2016 4:00:00 AM 0  
## 6 1503960366 4/12/2016 5:00:00 AM 0

str(hourly\_steps)

## spc\_tbl\_ [22,099 × 3] (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 AM" "4/12/2016 3:00:00 AM" ...  
## $ StepTotal : num [1:22099] 373 160 151 0 0 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. Id = col\_double(),  
## .. ActivityHour = col\_character(),  
## .. StepTotal = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

head(daily\_calories)

## # A tibble: 6 × 3  
## Id ActivityDay Calories  
## <dbl> <chr> <dbl>  
## 1 1503960366 4/12/2016 1985  
## 2 1503960366 4/13/2016 1797  
## 3 1503960366 4/14/2016 1776  
## 4 1503960366 4/15/2016 1745  
## 5 1503960366 4/16/2016 1863  
## 6 1503960366 4/17/2016 1728

str(daily\_calories)

## spc\_tbl\_ [940 × 3] (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 ...  
## $ ActivityDay: chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...  
## $ Calories : num [1:940] 1985 1797 1776 1745 1863 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. Id = col\_double(),  
## .. ActivityDay = col\_character(),  
## .. Calories = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

### **4.4 Cleaning and formatting the data**

First I checked the number of unique users on every dataset:

n\_unique(daily\_activity$Id)

## [1] 33

n\_unique(daily\_intensity$Id)

## [1] 33

n\_unique(daily\_sleep$Id)

## [1] 24

n\_unique(hourly\_steps$Id)

## [1] 33

n\_unique(daily\_calories$Id)

## [1] 33

Then I find out the number of duplicates in each dataset:

sum(duplicated(daily\_activity))

## [1] 0

sum(duplicated(daily\_intensity))

## [1] 0

sum(duplicated(daily\_sleep))

## [1] 3

sum(duplicated(hourly\_steps))

## [1] 0

sum(duplicated(daily\_calories))

## [1] 0

Only the daily\_sleep dataset contains duplicates. The next code chunk removes the duplicates.

daily\_sleep <- daily\_sleep %>%  
 distinct() %>%  
 drop\_na()  
  
sum(duplicated(daily\_sleep))

## [1] 0

Now, I format the column names across all datasets to lowercase.

clean\_names(daily\_activity)

## # A tibble: 940 × 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>

daily\_activity<- rename\_with(daily\_activity, tolower)  
clean\_names(daily\_sleep)

## # A tibble: 410 × 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 400 more rows

daily\_sleep <- rename\_with(daily\_sleep, tolower)  
clean\_names(daily\_intensity)

## # A tibble: 940 × 10  
## id activity\_day sedentary\_minutes lightly\_active\_minutes  
## <dbl> <chr> <dbl> <dbl>  
## 1 1503960366 4/12/2016 728 328  
## 2 1503960366 4/13/2016 776 217  
## 3 1503960366 4/14/2016 1218 181  
## 4 1503960366 4/15/2016 726 209  
## 5 1503960366 4/16/2016 773 221  
## 6 1503960366 4/17/2016 539 164  
## 7 1503960366 4/18/2016 1149 233  
## 8 1503960366 4/19/2016 775 264  
## 9 1503960366 4/20/2016 818 205  
## 10 1503960366 4/21/2016 838 211  
## # ℹ 930 more rows  
## # ℹ 6 more variables: fairly\_active\_minutes <dbl>, very\_active\_minutes <dbl>,  
## # sedentary\_active\_distance <dbl>, light\_active\_distance <dbl>,  
## # moderately\_active\_distance <dbl>, very\_active\_distance <dbl>

daily\_intensity<- rename\_with(daily\_intensity, tolower)  
clean\_names(daily\_calories)

## # A tibble: 940 × 3  
## id activity\_day calories  
## <dbl> <chr> <dbl>  
## 1 1503960366 4/12/2016 1985  
## 2 1503960366 4/13/2016 1797  
## 3 1503960366 4/14/2016 1776  
## 4 1503960366 4/15/2016 1745  
## 5 1503960366 4/16/2016 1863  
## 6 1503960366 4/17/2016 1728  
## 7 1503960366 4/18/2016 1921  
## 8 1503960366 4/19/2016 2035  
## 9 1503960366 4/20/2016 1786  
## 10 1503960366 4/21/2016 1775  
## # ℹ 930 more rows

daily\_calories <- rename\_with(daily\_calories, tolower)  
clean\_names(hourly\_steps)

## # A tibble: 22,099 × 3  
## id activity\_hour step\_total  
## <dbl> <chr> <dbl>  
## 1 1503960366 4/12/2016 12:00:00 AM 373  
## 2 1503960366 4/12/2016 1:00:00 AM 160  
## 3 1503960366 4/12/2016 2:00:00 AM 151  
## 4 1503960366 4/12/2016 3:00:00 AM 0  
## 5 1503960366 4/12/2016 4:00:00 AM 0  
## 6 1503960366 4/12/2016 5:00:00 AM 0  
## 7 1503960366 4/12/2016 6:00:00 AM 0  
## 8 1503960366 4/12/2016 7:00:00 AM 0  
## 9 1503960366 4/12/2016 8:00:00 AM 250  
## 10 1503960366 4/12/2016 9:00:00 AM 1864  
## # ℹ 22,089 more rows

hourly\_steps <- rename\_with(hourly\_steps, tolower)

Formatting Time/Date Cloumns across all datasets for consistency.

daily\_activity <- daily\_activity %>%  
 rename(date = activitydate) %>%  
 mutate(date = as\_date(date, format = "%m/%d/%Y"))  
  
daily\_sleep <- daily\_sleep %>%  
 rename(date = sleepday) %>%  
 mutate(date = as\_date(date,format ="%m/%d/%Y %I:%M:%S %p" , tz=Sys.timezone()))

## Warning: There was 1 warning in `mutate()`.  
## ℹ In argument: `date = as\_date(date, format = "%m/%d/%Y %I:%M:%S %p", tz =  
## Sys.timezone())`.  
## Caused by warning:  
## ! `tz` argument is ignored by `as\_date()`

daily\_intensity <- daily\_intensity %>%  
 rename(date = activityday) %>%  
 mutate(date = as\_date(date, format = "%m/%d/%Y"))  
  
daily\_calories <- daily\_calories %>%  
 rename(date = activityday) %>%  
 mutate(date = as\_date(date,format ="%m/%d/%Y" , tz=Sys.timezone()))

## Warning: There was 1 warning in `mutate()`.  
## ℹ In argument: `date = as\_date(date, format = "%m/%d/%Y", tz =  
## Sys.timezone())`.  
## Caused by warning:  
## ! `tz` argument is ignored by `as\_date()`

hourly\_steps<- hourly\_steps %>%   
 rename(date\_time = activityhour) %>%   
 mutate(date\_time = as.POSIXct(date\_time,format ="%m/%d/%Y %I:%M:%S %p" , tz=Sys.timezone()))  
  
head(daily\_activity)

## # A tibble: 6 × 15  
## id date totalsteps totaldistance trackerdistance  
## <dbl> <date> <dbl> <dbl> <dbl>  
## 1 1503960366 2016-04-12 13162 8.5 8.5   
## 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  
## # 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 × 5  
## id date totalsleeprecords totalminutesasleep totaltimeinbed  
## <dbl> <date> <dbl> <dbl> <dbl>  
## 1 1503960366 2016-04-12 1 327 346  
## 2 1503960366 2016-04-13 2 384 407  
## 3 1503960366 2016-04-15 1 412 442  
## 4 1503960366 2016-04-16 2 340 367  
## 5 1503960366 2016-04-17 1 700 712  
## 6 1503960366 2016-04-19 1 304 320

head(daily\_intensity)

## # A tibble: 6 × 10  
## id date sedentaryminutes lightlyactiveminutes fairlyactiveminutes  
## <dbl> <date> <dbl> <dbl> <dbl>  
## 1 1.50e9 2016-04-12 728 328 13  
## 2 1.50e9 2016-04-13 776 217 19  
## 3 1.50e9 2016-04-14 1218 181 11  
## 4 1.50e9 2016-04-15 726 209 34  
## 5 1.50e9 2016-04-16 773 221 10  
## 6 1.50e9 2016-04-17 539 164 20  
## # ℹ 5 more variables: veryactiveminutes <dbl>, sedentaryactivedistance <dbl>,  
## # lightactivedistance <dbl>, moderatelyactivedistance <dbl>,  
## # veryactivedistance <dbl>

head(daily\_calories)

## # A tibble: 6 × 3  
## id date calories  
## <dbl> <date> <dbl>  
## 1 1503960366 2016-04-12 1985  
## 2 1503960366 2016-04-13 1797  
## 3 1503960366 2016-04-14 1776  
## 4 1503960366 2016-04-15 1745  
## 5 1503960366 2016-04-16 1863  
## 6 1503960366 2016-04-17 1728

head(hourly\_steps)

## # A tibble: 6 × 3  
## id date\_time steptotal  
## <dbl> <dttm> <dbl>  
## 1 1503960366 2016-04-12 00:00:00 373  
## 2 1503960366 2016-04-12 01:00:00 160  
## 3 1503960366 2016-04-12 02:00:00 151  
## 4 1503960366 2016-04-12 03:00:00 0  
## 5 1503960366 2016-04-12 04:00:00 0  
## 6 1503960366 2016-04-12 05:00:00 0

### **4.5 Summarizing and Merging Datasets**

First, I get the summary of each dataset to further understand the data.

daily\_activity %>%   
 select(totalsteps,  
 totaldistance,  
 sedentaryminutes,   
 calories) %>%  
 summary()

## totalsteps totaldistance sedentaryminutes calories   
## Min. : 0 Min. : 0.000 Min. : 0.0 Min. : 0   
## 1st Qu.: 3790 1st Qu.: 2.620 1st Qu.: 729.8 1st Qu.:1828   
## Median : 7406 Median : 5.245 Median :1057.5 Median :2134   
## Mean : 7638 Mean : 5.490 Mean : 991.2 Mean :2304   
## 3rd Qu.:10727 3rd Qu.: 7.713 3rd Qu.:1229.5 3rd Qu.:2793   
## Max. :36019 Max. :28.030 Max. :1440.0 Max. :4900

daily\_activity %>%  
 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.:127.0   
## Median : 4.00 Median : 6.00 Median :199.0   
## Mean : 21.16 Mean : 13.56 Mean :192.8   
## 3rd Qu.: 32.00 3rd Qu.: 19.00 3rd Qu.:264.0   
## Max. :210.00 Max. :143.00 Max. :518.0

daily\_calories %>%  
 select(calories) %>%  
 summary()

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

daily\_sleep %>%  
 select(totalsleeprecords, totalminutesasleep, totaltimeinbed) %>%  
 summary()

## totalsleeprecords totalminutesasleep totaltimeinbed   
## Min. :1.00 Min. : 58.0 Min. : 61.0   
## 1st Qu.:1.00 1st Qu.:361.0 1st Qu.:403.8   
## Median :1.00 Median :432.5 Median :463.0   
## Mean :1.12 Mean :419.2 Mean :458.5   
## 3rd Qu.:1.00 3rd Qu.:490.0 3rd Qu.:526.0   
## Max. :3.00 Max. :796.0 Max. :961.0

daily\_intensity %>%  
 select(sedentaryminutes, lightlyactiveminutes, fairlyactiveminutes, veryactiveminutes) %>%  
 summary()

## sedentaryminutes lightlyactiveminutes fairlyactiveminutes veryactiveminutes  
## Min. : 0.0 Min. : 0.0 Min. : 0.00 Min. : 0.00   
## 1st Qu.: 729.8 1st Qu.:127.0 1st Qu.: 0.00 1st Qu.: 0.00   
## Median :1057.5 Median :199.0 Median : 6.00 Median : 4.00   
## Mean : 991.2 Mean :192.8 Mean : 13.56 Mean : 21.16   
## 3rd Qu.:1229.5 3rd Qu.:264.0 3rd Qu.: 19.00 3rd Qu.: 32.00   
## Max. :1440.0 Max. :518.0 Max. :143.00 Max. :210.00

**Some interesting findings from this summary include:**

* The average sedentary time is 991 minutes or 16 hours, which is a **high amount** and should be **reduced.**
* The majority of participants are lightly active, indicating that they **may not be engaging in enough physical activity.**
* On average, participants sleep once for 7 hours, suggesting that they are getting a decent amount of sleep.
* The average total steps per day are 7638, which is slightly below the recommended level for health benefits according to CDC research. The CDC found that taking 8,000 steps per day was associated with a 51% lower risk of all-cause mortality, while taking 12,000 steps per day was associated with a 65% lower risk compared to taking only 4,000 steps.

Next, I merge some datasets to analyze for insights:

daily\_activity\_sleep <- merge(daily\_activity, daily\_sleep, by=c ("id", "date"))  
glimpse(daily\_activity\_sleep)

## Rows: 410  
## Columns: 18  
## $ id <dbl> 1503960366, 1503960366, 1503960366, 150396036…  
## $ date <date> 2016-04-12, 2016-04-13, 2016-04-15, 2016-04-…  
## $ totalsteps <dbl> 13162, 10735, 9762, 12669, 9705, 15506, 10544…  
## $ totaldistance <dbl> 8.50, 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6.3…  
## $ trackerdistance <dbl> 8.50, 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6.3…  
## $ loggedactivitiesdistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ veryactivedistance <dbl> 1.88, 1.57, 2.14, 2.71, 3.19, 3.53, 1.96, 1.3…  
## $ moderatelyactivedistance <dbl> 0.55, 0.69, 1.26, 0.41, 0.78, 1.32, 0.48, 0.3…  
## $ lightactivedistance <dbl> 6.06, 4.71, 2.83, 5.04, 2.51, 5.03, 4.24, 4.6…  
## $ sedentaryactivedistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ veryactiveminutes <dbl> 25, 21, 29, 36, 38, 50, 28, 19, 41, 39, 73, 3…  
## $ fairlyactiveminutes <dbl> 13, 19, 34, 10, 20, 31, 12, 8, 21, 5, 14, 23,…  
## $ lightlyactiveminutes <dbl> 328, 217, 209, 221, 164, 264, 205, 211, 262, …  
## $ sedentaryminutes <dbl> 728, 776, 726, 773, 539, 775, 818, 838, 732, …  
## $ calories <dbl> 1985, 1797, 1745, 1863, 1728, 2035, 1786, 177…  
## $ totalsleeprecords <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ totalminutesasleep <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361, …  
## $ totaltimeinbed <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384, …

daily\_average <- daily\_activity\_sleep %>%  
 group\_by(id) %>%  
 summarise (mean\_daily\_steps = mean(totalsteps), mean\_daily\_calories = mean(calories), mean\_daily\_sleep = mean(totalminutesasleep))  
  
head(daily\_average)

## # A tibble: 6 × 4  
## id mean\_daily\_steps mean\_daily\_calories mean\_daily\_sleep  
## <dbl> <dbl> <dbl> <dbl>  
## 1 1503960366 12406. 1872. 360.  
## 2 1644430081 7968. 2978. 294   
## 3 1844505072 3477 1676. 652   
## 4 1927972279 1490 2316. 417   
## 5 2026352035 5619. 1541. 506.  
## 6 2320127002 5079 1804 61

daily\_intensity\_sleep <- merge(daily\_intensity, daily\_sleep, by=c ("id", "date"))  
glimpse(daily\_intensity\_sleep)

## Rows: 410  
## Columns: 13  
## $ id <dbl> 1503960366, 1503960366, 1503960366, 150396036…  
## $ date <date> 2016-04-12, 2016-04-13, 2016-04-15, 2016-04-…  
## $ sedentaryminutes <dbl> 728, 776, 726, 773, 539, 775, 818, 838, 732, …  
## $ lightlyactiveminutes <dbl> 328, 217, 209, 221, 164, 264, 205, 211, 262, …  
## $ fairlyactiveminutes <dbl> 13, 19, 34, 10, 20, 31, 12, 8, 21, 5, 14, 23,…  
## $ veryactiveminutes <dbl> 25, 21, 29, 36, 38, 50, 28, 19, 41, 39, 73, 3…  
## $ sedentaryactivedistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ lightactivedistance <dbl> 6.06, 4.71, 2.83, 5.04, 2.51, 5.03, 4.24, 4.6…  
## $ moderatelyactivedistance <dbl> 0.55, 0.69, 1.26, 0.41, 0.78, 1.32, 0.48, 0.3…  
## $ veryactivedistance <dbl> 1.88, 1.57, 2.14, 2.71, 3.19, 3.53, 1.96, 1.3…  
## $ totalsleeprecords <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …  
## $ totalminutesasleep <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361, …  
## $ totaltimeinbed <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384, …

avg\_intensity\_sleep <- daily\_intensity\_sleep %>%  
 mutate(avg\_intensity = lightlyactiveminutes+fairlyactiveminutes+veryactiveminutes)  
head(avg\_intensity\_sleep)

## id date sedentaryminutes lightlyactiveminutes  
## 1 1503960366 2016-04-12 728 328  
## 2 1503960366 2016-04-13 776 217  
## 3 1503960366 2016-04-15 726 209  
## 4 1503960366 2016-04-16 773 221  
## 5 1503960366 2016-04-17 539 164  
## 6 1503960366 2016-04-19 775 264  
## fairlyactiveminutes veryactiveminutes sedentaryactivedistance  
## 1 13 25 0  
## 2 19 21 0  
## 3 34 29 0  
## 4 10 36 0  
## 5 20 38 0  
## 6 31 50 0  
## lightactivedistance moderatelyactivedistance veryactivedistance  
## 1 6.06 0.55 1.88  
## 2 4.71 0.69 1.57  
## 3 2.83 1.26 2.14  
## 4 5.04 0.41 2.71  
## 5 2.51 0.78 3.19  
## 6 5.03 1.32 3.53  
## totalsleeprecords totalminutesasleep totaltimeinbed avg\_intensity  
## 1 1 327 346 366  
## 2 2 384 407 257  
## 3 1 412 442 272  
## 4 2 340 367 267  
## 5 1 700 712 222  
## 6 1 304 320 345

## **5. Analyze and Share**

### **5.1 User type by activity level**

According to <https://www.medicinenet.com/how_many_steps_a_day_is_considered_active/article.htm> users can be divided according to the number of steps they take every day. Users are divided as follows:

1. Sedentary: Less than 5,000 steps daily
2. Low active: About 5,000 to 7,499 steps daily
3. Somewhat active: About 7,500 to 9,999 steps daily
4. Active: More than 10,000 steps daily
5. Highly active: More than 12,500 steps daily

I sort the users accordingly:

user\_type <- daily\_average %>%  
 mutate(user\_type = case\_when(  
 mean\_daily\_steps < 5000 ~ "sedentary",  
 mean\_daily\_steps >= 5000 & mean\_daily\_steps < 7499 ~ "low active",   
 mean\_daily\_steps >= 7500 & mean\_daily\_steps < 9999 ~ "somewhat active",   
 mean\_daily\_steps >= 10000 & mean\_daily\_steps < 12500 ~ "active",  
 mean\_daily\_steps >= 12500 ~ "highly active"  
 ))  
  
head(user\_type)

## # A tibble: 6 × 5  
## id mean\_daily\_steps mean\_daily\_calories mean\_daily\_sleep user\_type   
## <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 1503960366 12406. 1872. 360. active   
## 2 1644430081 7968. 2978. 294 somewhat act…  
## 3 1844505072 3477 1676. 652 sedentary   
## 4 1927972279 1490 2316. 417 sedentary   
## 5 2026352035 5619. 1541. 506. low active   
## 6 2320127002 5079 1804 61 low active

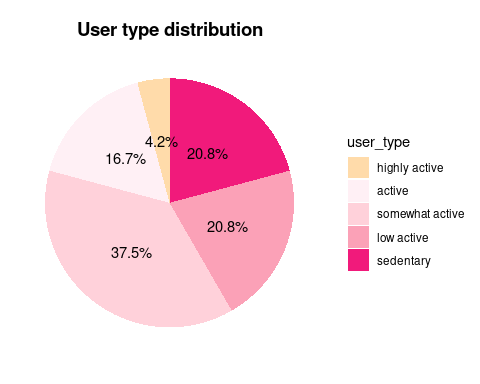
By percentage:

user\_type\_percent <- user\_type %>%  
 group\_by(user\_type) %>%  
 summarise(total = n()) %>%  
 mutate(totals = sum(total)) %>%  
 group\_by(user\_type) %>%  
 summarise(total\_percent = total / totals) %>%  
 mutate(labels = scales::percent(total\_percent))  
  
user\_type\_percent$user\_type <- factor(user\_type\_percent$user\_type , levels = c("highly active", "active", "somewhat active", "low active", "sedentary"))  
  
head(user\_type\_percent)

## # A tibble: 5 × 3  
## user\_type total\_percent labels  
## <fct> <dbl> <chr>   
## 1 active 0.167 16.7%   
## 2 highly active 0.0417 4.2%   
## 3 low active 0.208 20.8%   
## 4 sedentary 0.208 20.8%   
## 5 somewhat active 0.375 37.5%

Next, a pie chart is used to visualize the results.

user\_type\_percent %>%  
 ggplot(aes(x="",y=total\_percent, fill=user\_type)) +  
 geom\_bar(stat = "identity", width = 1)+  
 coord\_polar("y", start=0)+  
 theme\_minimal()+  
 theme(axis.title.x= element\_blank(),  
 axis.title.y = element\_blank(),  
 panel.border = element\_blank(),   
 panel.grid = element\_blank(),   
 axis.ticks = element\_blank(),  
 axis.text.x = element\_blank(),  
 plot.title = element\_text(hjust = 0.5, size=14, face = "bold")) +  
 scale\_fill\_manual(values = c("#FFDBAA","#FFF0F5", "#ffd1da", "#fba1b7", "#F11A7B")) +  
 geom\_text(aes(label = labels),  
 position = position\_stack(vjust = 0.5))+  
 labs(title="User type distribution")



### **5.2 Correlations**

#### **5.2.1 Correlation between daily intensity and time in bed**

I am interested to find out if there is a correlation between the average daily intensity of the user and the total time they spend in bed. Hence, I will perform a correlation test.

*A Correlation test* is used to evaluate the association between two or more variables.

There are different methods used to perform correlation analysis. I am going to use the Pearson Correlation.

*Pearson correlation (r)*, which measures a linear dependence between two variables (x and y). It’s also known as a parametric correlation test because it depends to the distribution of the data. It can be used only when x and y are from normal distribution. The plot of y = f(x) is named the linear regression curve.

**Correlation formula**

In the formula below, \* x and y are two vectors of length n \* mx and my corresponds to the means of x and y, respectively.

**Pearson correlation formula**

*mx* and *my* are the means of x and y variables.

The p-value (significance level) of the correlation can be determined:

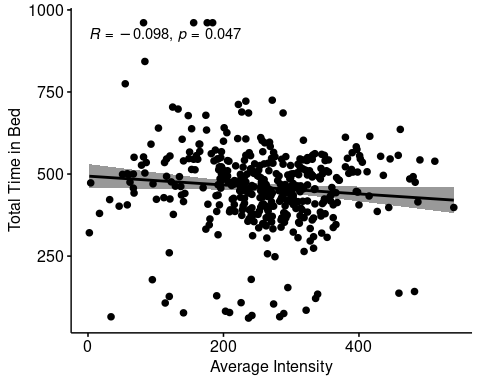
1. by using the correlation coefficient table for the degrees of freedom : df=n−2, where n is the number of observation in x and y variables.
2. or by calculating the t value as follows:

In the case 2) the corresponding p-value is determined using t distribution table for df=n−2

**If the p-value is < 5%, then the correlation between x and y is significant.**

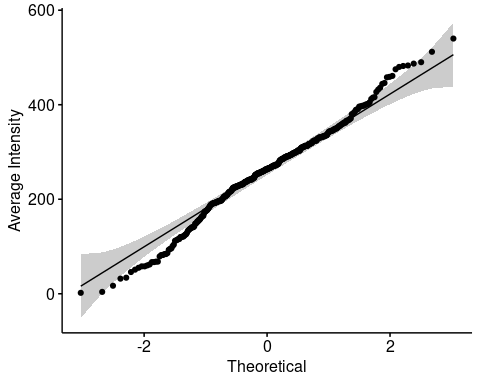
First, I visualize the data using a scatter plot.

ggscatter(avg\_intensity\_sleep, x = "avg\_intensity" , y ="totaltimeinbed",   
 add = "reg.line", conf.int = TRUE,   
 cor.coef = TRUE, cor.method = "pearson",  
 xlab ="Average Intensity" , ylab = "Total Time in Bed")

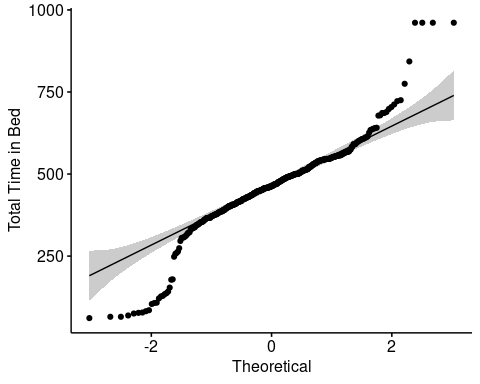


Visual inspection of the data normality using Q-Q plots (quantile-quantile plots). Q-Q plot draws the correlation between a given sample and the normal distribution.

# Average Intensity  
ggqqplot(avg\_intensity\_sleep$avg\_intensity, ylab = "Average Intensity")



# Total Time in Bed  
ggqqplot(avg\_intensity\_sleep$totaltimeinbed, ylab = "Total Time in Bed")



Pearson Correlation Test between Average Intensity and Total Time in Bed:

res <- cor.test(avg\_intensity\_sleep$avg\_intensity, avg\_intensity\_sleep$totaltimeinbed,   
 method = "pearson")  
res

##   
## Pearson's product-moment correlation  
##   
## data: avg\_intensity\_sleep$avg\_intensity and avg\_intensity\_sleep$totaltimeinbed  
## t = -1.9879, df = 408, p-value = 0.04749  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## -0.192958030 -0.001104167  
## sample estimates:  
## cor   
## -0.097941

**In the result above:**

* t is the t-test statistic value (t = -1.9879),
* df is the degrees of freedom (df= 408),
* p-value is the significance level of the t-test (p-value = 0.04749).
* conf.int is the confidence interval of the correlation coefficient at 95% (conf.int = [-0.192958030, -0.001104167]);
* sample estimates is the correlation coefficient (Cor.coeff = -0.097941).

*The p-value of the test is 0.04749, which is less than the significance level alpha = 0.05. We can conclude that Average Intensity and Total Time in Bed are correlated with a correlation coefficient of -0.097941 and p-value of 0.04749.*

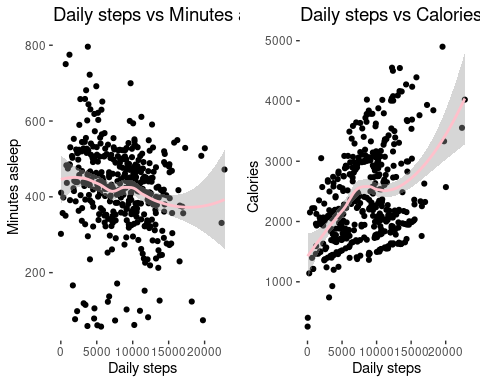
**There is a slight correlation between intensity and total time in bed. The chart shows that the higher the average intensity throughout the day, the less time is spent in bed. However, the correlation is not strong enough to conclude that intensity level is the cause behind time spent in bed. Bellabeat could suggest activities for the users throughout the day to reduce time wasted in bed.**

#### **5.2.2 Further Correlations**

I will now try to identify if there are any correlations between daily steps and sleep as well as calories.

ggarrange(  
 ggplot(daily\_activity\_sleep, aes(x=totalsteps, y=totalminutesasleep))+  
 geom\_jitter() +  
 geom\_smooth(color = "pink") +   
 labs(title = "Daily steps vs Minutes asleep", x = "Daily steps", y= "Minutes asleep") +  
 theme(panel.background = element\_blank(),  
 plot.title = element\_text(size=14)),   
 ggplot(daily\_activity\_sleep, aes(x=totalsteps, y=calories))+  
 geom\_jitter() +  
 geom\_smooth(color = "pink") +   
 labs(title = "Daily steps vs Calories", x = "Daily steps", y= "Calories") +  
 theme(panel.background = element\_blank(),  
 plot.title = element\_text(size=14))  
)

## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'  
## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'



According to our results: **There is no correlation between daily activity levels and sleep. However, there is a positive correlation between the daily steps taken by users and the calories they burn.**

### **5.3 Steps throughout the day**

As I go further into the analysis, I want to know precisely when users are more active throughout the day.

First, I will seperate the date and time from the date\_time column:

hourly\_steps <- hourly\_steps %>%  
 separate(date\_time, into = c("date", "time"), sep= " ") %>%  
 mutate(date = ymd(date))

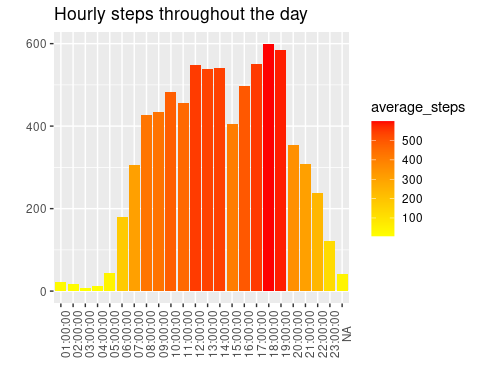
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 934 rows [1, 25, 49, 73,  
## 97, 121, 145, 169, 193, 217, 241, 265, 289, 313, 337, 361, 385, 409, 433, 457,  
## ...].

head(hourly\_steps)

## # A tibble: 6 × 4  
## id date time steptotal  
## <dbl> <date> <chr> <dbl>  
## 1 1503960366 2016-04-12 <NA> 373  
## 2 1503960366 2016-04-12 01:00:00 160  
## 3 1503960366 2016-04-12 02:00:00 151  
## 4 1503960366 2016-04-12 03:00:00 0  
## 5 1503960366 2016-04-12 04:00:00 0  
## 6 1503960366 2016-04-12 05:00:00 0

Now, I will plot the average steps taken by a user throughout the day.

hourly\_steps %>%  
 group\_by(time) %>%  
 summarize(average\_steps = mean(steptotal)) %>%  
 ggplot() +  
 geom\_col(mapping = aes(x=time, y = average\_steps, fill = average\_steps)) +   
 labs(title = "Hourly steps throughout the day", x="", y="") +   
 scale\_fill\_gradient(low = "yellow", high = "red")+  
 theme(axis.text.x = element\_text(angle = 90))



**Users are more active from 8 am to 7 pm, as can be seen. Taking extra steps between 5 and 7 in the evening and over lunch from 12 to 2 in the afternoon.**

### **5.4 Use of smart device**

I am interested in finding out how frequently the users in our sample use their device now that we have observed certain trends in activity, sleep, and calories burned. By doing so, we can plan our marketing approach and determine which features would make the application more useful. Knowing that the date interval is 31 days, we will divide our sample into three categories and determine the proportion of users who use their smart device on a daily basis:

1. high use - users who use their device between 21 and 31 days.
2. moderate use - users who use their device between 10 and 20 days.
3. low use - users who use their device between 1 and 10 days.

daily\_use <- daily\_activity\_sleep %>%  
 group\_by(id) %>%  
 summarize(days\_used=sum(n())) %>%  
 mutate(usage = case\_when(  
 days\_used >= 1 & days\_used <= 10 ~ "low use",  
 days\_used >= 11 & days\_used <= 20 ~ "moderate use",   
 days\_used >= 21 & days\_used <= 31 ~ "high use",   
 ))  
  
head(daily\_use)

## # A tibble: 6 × 3  
## id days\_used usage   
## <dbl> <int> <chr>   
## 1 1503960366 25 high use  
## 2 1644430081 4 low use   
## 3 1844505072 3 low use   
## 4 1927972279 5 low use   
## 5 2026352035 28 high use  
## 6 2320127002 1 low use

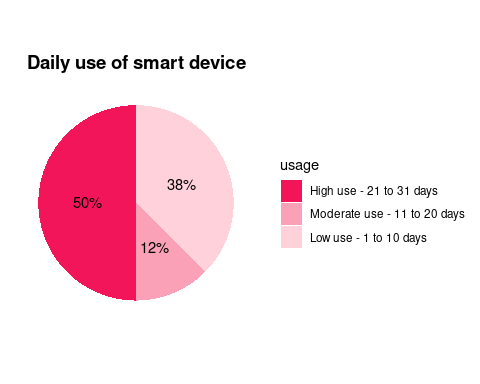
Sorting them by percentage:

daily\_use\_percent <- daily\_use %>%  
 group\_by(usage) %>%  
 summarise(total = n()) %>%  
 mutate(totals = sum(total)) %>%  
 group\_by(usage) %>%  
 summarise(total\_percent = total / totals) %>%  
 mutate(labels = scales::percent(total\_percent))  
  
daily\_use\_percent$usage <- factor(daily\_use\_percent$usage, levels = c("high use", "moderate use", "low use"))  
  
head(daily\_use\_percent)

## # A tibble: 3 × 3  
## usage total\_percent labels  
## <fct> <dbl> <chr>   
## 1 high use 0.5 50%   
## 2 low use 0.375 38%   
## 3 moderate use 0.125 12%

Visualizing the results:

daily\_use\_percent %>%  
 ggplot(aes(x="",y=total\_percent, fill=usage)) +  
 geom\_bar(stat = "identity", width = 1)+  
 coord\_polar("y", start=0)+  
 theme\_minimal()+  
 theme(axis.title.x= element\_blank(),  
 axis.title.y = element\_blank(),  
 panel.border = element\_blank(),   
 panel.grid = element\_blank(),   
 axis.ticks = element\_blank(),  
 axis.text.x = element\_blank(),  
 plot.title = element\_text(hjust = 0.5, size=14, face = "bold")) +  
 geom\_text(aes(label = labels),  
 position = position\_stack(vjust = 0.5))+  
 scale\_fill\_manual(values = c("#F31559","#FBA1B7","#FFD1DA"),  
 labels = c("High use - 21 to 31 days",  
 "Moderate use - 11 to 20 days",  
 "Low use - 1 to 10 days"))+  
 labs(title="Daily use of smart device")



#Merging daily activity and use   
daily\_use\_merged <- merge(daily\_activity, daily\_use, by=c ("id"))  
head(daily\_use\_merged)

## id date totalsteps totaldistance trackerdistance  
## 1 1503960366 2016-05-07 11992 7.71 7.71  
## 2 1503960366 2016-05-06 12159 8.03 8.03  
## 3 1503960366 2016-05-01 10602 6.81 6.81  
## 4 1503960366 2016-04-30 14673 9.25 9.25  
## 5 1503960366 2016-04-12 13162 8.50 8.50  
## 6 1503960366 2016-04-13 10735 6.97 6.97  
## loggedactivitiesdistance veryactivedistance moderatelyactivedistance  
## 1 0 2.46 2.12  
## 2 0 1.97 0.25  
## 3 0 2.29 1.60  
## 4 0 3.56 1.42  
## 5 0 1.88 0.55  
## 6 0 1.57 0.69  
## lightactivedistance sedentaryactivedistance veryactiveminutes  
## 1 3.13 0 37  
## 2 5.81 0 24  
## 3 2.92 0 33  
## 4 4.27 0 52  
## 5 6.06 0 25  
## 6 4.71 0 21  
## fairlyactiveminutes lightlyactiveminutes sedentaryminutes calories days\_used  
## 1 46 175 833 1821 25  
## 2 6 289 754 1896 25  
## 3 35 246 730 1820 25  
## 4 34 217 712 1947 25  
## 5 13 328 728 1985 25  
## 6 19 217 776 1797 25  
## usage  
## 1 high use  
## 2 high use  
## 3 high use  
## 4 high use  
## 5 high use  
## 6 high use

#Segregating total activity by percentage  
minutes\_worn <- daily\_use\_merged %>%   
 mutate(total\_minutes\_worn = veryactiveminutes+fairlyactiveminutes+lightlyactiveminutes+sedentaryminutes)%>%  
 mutate (percent\_minutes\_worn = (total\_minutes\_worn/1440)\*100) %>%  
 mutate (worn = case\_when(  
 percent\_minutes\_worn == 100 ~ "All day",  
 percent\_minutes\_worn < 100 & percent\_minutes\_worn >= 50~ "More than half day",   
 percent\_minutes\_worn < 50 & percent\_minutes\_worn > 0 ~ "Less than half day"  
 ))  
  
head(minutes\_worn)

## id date totalsteps totaldistance trackerdistance  
## 1 1503960366 2016-05-07 11992 7.71 7.71  
## 2 1503960366 2016-05-06 12159 8.03 8.03  
## 3 1503960366 2016-05-01 10602 6.81 6.81  
## 4 1503960366 2016-04-30 14673 9.25 9.25  
## 5 1503960366 2016-04-12 13162 8.50 8.50  
## 6 1503960366 2016-04-13 10735 6.97 6.97  
## loggedactivitiesdistance veryactivedistance moderatelyactivedistance  
## 1 0 2.46 2.12  
## 2 0 1.97 0.25  
## 3 0 2.29 1.60  
## 4 0 3.56 1.42  
## 5 0 1.88 0.55  
## 6 0 1.57 0.69  
## lightactivedistance sedentaryactivedistance veryactiveminutes  
## 1 3.13 0 37  
## 2 5.81 0 24  
## 3 2.92 0 33  
## 4 4.27 0 52  
## 5 6.06 0 25  
## 6 4.71 0 21  
## fairlyactiveminutes lightlyactiveminutes sedentaryminutes calories days\_used  
## 1 46 175 833 1821 25  
## 2 6 289 754 1896 25  
## 3 35 246 730 1820 25  
## 4 34 217 712 1947 25  
## 5 13 328 728 1985 25  
## 6 19 217 776 1797 25  
## usage total\_minutes\_worn percent\_minutes\_worn worn  
## 1 high use 1091 75.76389 More than half day  
## 2 high use 1073 74.51389 More than half day  
## 3 high use 1044 72.50000 More than half day  
## 4 high use 1015 70.48611 More than half day  
## 5 high use 1094 75.97222 More than half day  
## 6 high use 1033 71.73611 More than half day

Examining these findings, we may determine that

* 50% of the sample’s users use their smartphone every 21 to 31 days.
* 14% of people use their gadget 11–20 days per week.
* 38% of our group use their device just occasionally.

**To be more specific, we want to know how long consumers spend using their device each day. In order to filter results by daily use of the device as well, we are going to merge daily\_use with daily\_activity in a new constructed data frame.**

We must construct a new data frame that divides the total daily minutes users spent wearing the device into three categories:

* All day - device was worn all day.
* More than half day - device was worn more than half of the day.
* Less than half day - device was worn less than half of the day.

minutes\_worn\_percent<- minutes\_worn%>%  
 group\_by(worn) %>%  
 summarise(total = n()) %>%  
 mutate(totals = sum(total)) %>%  
 group\_by(worn) %>%  
 summarise(total\_percent = total / totals) %>%  
 mutate(labels = scales::percent(total\_percent))  
  
  
minutes\_worn\_highuse <- minutes\_worn%>%  
 filter (usage == "high use")%>%  
 group\_by(worn) %>%  
 summarise(total = n()) %>%  
 mutate(totals = sum(total)) %>%  
 group\_by(worn) %>%  
 summarise(total\_percent = total / totals) %>%  
 mutate(labels = scales::percent(total\_percent))  
  
minutes\_worn\_moduse <- minutes\_worn%>%  
 filter(usage == "moderate use") %>%  
 group\_by(worn) %>%  
 summarise(total = n()) %>%  
 mutate(totals = sum(total)) %>%  
 group\_by(worn) %>%  
 summarise(total\_percent = total / totals) %>%  
 mutate(labels = scales::percent(total\_percent))  
  
minutes\_worn\_lowuse <- minutes\_worn%>%  
 filter (usage == "low use") %>%  
 group\_by(worn) %>%  
 summarise(total = n()) %>%  
 mutate(totals = sum(total)) %>%  
 group\_by(worn) %>%  
 summarise(total\_percent = total / totals) %>%  
 mutate(labels = scales::percent(total\_percent))  
  
#Sorting by percentage of time worn  
minutes\_worn\_highuse$worn <- factor(minutes\_worn\_highuse$worn, levels = c("All day", "More than half day", "Less than half day"))  
minutes\_worn\_percent$worn <- factor(minutes\_worn\_percent$worn, levels = c("All day", "More than half day", "Less than half day"))  
minutes\_worn\_moduse$worn <- factor(minutes\_worn\_moduse$worn, levels = c("All day", "More than half day", "Less than half day"))  
minutes\_worn\_lowuse$worn <- factor(minutes\_worn\_lowuse$worn, levels = c("All day", "More than half day", "Less than half day"))  
  
head(minutes\_worn\_percent)

## # A tibble: 3 × 3  
## worn total\_percent labels  
## <fct> <dbl> <chr>   
## 1 All day 0.365 36%   
## 2 Less than half day 0.0351 4%   
## 3 More than half day 0.600 60%

head(minutes\_worn\_highuse)

## # A tibble: 3 × 3  
## worn total\_percent labels  
## <fct> <dbl> <chr>   
## 1 All day 0.0676 6.8%   
## 2 Less than half day 0.0432 4.3%   
## 3 More than half day 0.889 88.9%

head(minutes\_worn\_moduse)

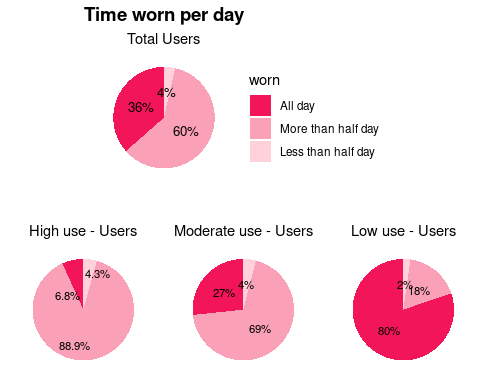
## # A tibble: 3 × 3  
## worn total\_percent labels  
## <fct> <dbl> <chr>   
## 1 All day 0.267 27%   
## 2 Less than half day 0.04 4%   
## 3 More than half day 0.693 69%

head(minutes\_worn\_lowuse)

## # A tibble: 3 × 3  
## worn total\_percent labels  
## <fct> <dbl> <chr>   
## 1 All day 0.802 80%   
## 2 Less than half day 0.0224 2%   
## 3 More than half day 0.175 18%

Visualizing the results:

ggarrange(  
 ggplot(minutes\_worn\_percent, aes(x="",y=total\_percent, fill=worn)) +  
 geom\_bar(stat = "identity", width = 1)+  
 coord\_polar("y", start=0)+  
 theme\_minimal()+  
 theme(axis.title.x= element\_blank(),  
 axis.title.y = element\_blank(),  
 panel.border = element\_blank(),   
 panel.grid = element\_blank(),   
 axis.ticks = element\_blank(),  
 axis.text.x = element\_blank(),  
 plot.title = element\_text(hjust = 0.5, size=14, face = "bold"),  
 plot.subtitle = element\_text(hjust = 0.5)) +  
 scale\_fill\_manual(values = c("#F31559", "#FBA1B7", "#FFD1DA"))+  
 geom\_text(aes(label = labels),  
 position = position\_stack(vjust = 0.5), size = 3.5)+  
 labs(title="Time worn per day", subtitle = "Total Users"),  
 ggarrange(  
 ggplot(minutes\_worn\_highuse, aes(x="",y=total\_percent, fill=worn)) +  
 geom\_bar(stat = "identity", width = 1)+  
 coord\_polar("y", start=0)+  
 theme\_minimal()+  
 theme(axis.title.x= element\_blank(),  
 axis.title.y = element\_blank(),  
 panel.border = element\_blank(),   
 panel.grid = element\_blank(),   
 axis.ticks = element\_blank(),  
 axis.text.x = element\_blank(),  
 plot.title = element\_text(hjust = 0.5, size=14, face = "bold"),  
 plot.subtitle = element\_text(hjust = 0.5),   
 legend.position = "none")+  
 scale\_fill\_manual(values = c("#F31559", "#FBA1B7", "#FFD1DA"))+  
 geom\_text\_repel(aes(label = labels),  
 position = position\_stack(vjust = 0.5), size = 3)+  
 labs(title="", subtitle = "High use - Users"),   
 ggplot(minutes\_worn\_moduse, aes(x="",y=total\_percent, fill=worn)) +  
 geom\_bar(stat = "identity", width = 1)+  
 coord\_polar("y", start=0)+  
 theme\_minimal()+  
 theme(axis.title.x= element\_blank(),  
 axis.title.y = element\_blank(),  
 panel.border = element\_blank(),   
 panel.grid = element\_blank(),   
 axis.ticks = element\_blank(),  
 axis.text.x = element\_blank(),  
 plot.title = element\_text(hjust = 0.5, size=14, face = "bold"),   
 plot.subtitle = element\_text(hjust = 0.5),  
 legend.position = "none") +  
 scale\_fill\_manual(values = c("#F31559", "#FBA1B7", "#FFD1DA"))+  
 geom\_text(aes(label = labels),  
 position = position\_stack(vjust = 0.5), size = 3)+  
 labs(title="", subtitle = "Moderate use - Users"),   
 ggplot(minutes\_worn\_lowuse, aes(x="",y=total\_percent, fill=worn)) +  
 geom\_bar(stat = "identity", width = 1)+  
 coord\_polar("y", start=0)+  
 theme\_minimal()+  
 theme(axis.title.x= element\_blank(),  
 axis.title.y = element\_blank(),  
 panel.border = element\_blank(),   
 panel.grid = element\_blank(),   
 axis.ticks = element\_blank(),  
 axis.text.x = element\_blank(),  
 plot.title = element\_text(hjust = 0.5, size=14, face = "bold"),   
 plot.subtitle = element\_text(hjust = 0.5),  
 legend.position = "none") +  
 scale\_fill\_manual(values = c("#F31559", "#FBA1B7", "#FFD1DA"))+  
 geom\_text(aes(label = labels),  
 position = position\_stack(vjust = 0.5), size = 3)+  
 labs(title="", subtitle = "Low use - Users"),   
 ncol = 3),   
 nrow = 2)



**According to our graphs, 36% of all users use the gadget continuously throughout the day, 60% do so for more than half of the day, and only 4% do so for less than the full day.**

The following results are obtained by filtering the total number of users taking into account the days on which they used the gadget and checking how long they wore it each day:

* **High users** - Just 6.8% of the users that have used their device between 21 and 31 days wear it all day. 88.9% wear the device more than half day but not all day.
* **Moderate users** are the ones who wear the device less on a daily basis.
* **Low users** wear their device much more than they use it.

## **6.Conclusions**

### **6.1 Target Audience**

It isn’t specified whether the sample data includes women and Bellabeat’s app and campaign are mainly directed towards women. I suggest using a more relevant and accurate dataset to derive insights directly related to women as their health metrics are different from their male counterparts.

### **6.2 Results**

The results from our analysis are as follows:

* The average sedentary time is 991 minutes or 16 hours, which is a **high amount and should be reduced.**
* The majority of participants are lightly active, indicating that they **may not be engaging in enough physical activity.**
* On average, participants sleep once for 7 hours, suggesting that they are getting a **decent amount of sleep.**
* The average total steps per day are **7638**, which is slightly below the recommended level for health benefits according to CDC research. The CDC found that taking **8,000** steps per day was associated with a 51% lower risk of all-cause mortality, while taking 12,000 steps per day was associated with a 65% lower risk compared to taking only 4,000 steps.
* There is no correlation between daily activity levels and sleep. However, there is a **positive correlation between the daily steps taken by users and the calories they burn.**
* Users are more active from 8 am to 7 pm, as can be seen. Taking extra steps between **5 and 7 in the evening** and over lunch from **12 to 2 in the afternoon.**
* **50%** of the sample’s users use their smartphone every **21 to 31 days.**
* **14%** of people use their gadget **11–20 days** per month.
* **38%** of our group use their device just **occasionally.**
* **High users** - Just 6.8% of the users that have used their device between 21 and 31 days wear it all day. 88.9% wear the device more than half day but not all day.
* **Moderate users** are the ones who wear the device less on a daily basis.
* **Low users wear** their device much more than they use it.

### **6.3 Recommendations**

Bellabeat’s mission is to empower women by providing them with the data to discover themselves.

1. **Notifications:**

* The app can send daily notifications to remind users to take their recommended 8,000 steps, specially between 5 and 7 in the evening and over lunch from 12 to 2 in the afternoon.
* It can send notifications to remind low and moderate users of their daily activities as well as send them information on their daily steps, activities and sleep patterns.

1. **Posts and Articles:**

Since Bellabeat’s mission is to empower women by providing them with the relevant knowledge they need to discover and take care of themselves, then the app should have posts and articles about:

* Women’s health
* Calorie consumption and diet suggestions
* Activity suggestions and routines
* The importance of sleep and how to improve it
* Etc..

This way, it can empower them with the knowledge they need to take their health into their own hands.

1. **Reward System:** Bellabeat can create a reward system for women that use the app more frequently and are more consistent with their health by incentivizing them with discounts and rewards. This will encourage women to use the app more while keeping them healthy.