BellaBeat_Analysis

Praveen Choragudi

2022-10-24

Table of Contents

Introduction

Bellabeat is a high-tech producer of goods with an emphasis on tracking and enhancing women's health. They are a prosperous small business with the potential to dominate the global market for smart devices. As a junior data analyst for Bellabeat's marketing analyst team, I will be examining some fitness data from smart devices that could open up new business options for the organisation.

Characters and products

Characters

- 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. You joined this team six months ago and have been busy learning about Bellabeat' 's mission and business goals — as well as how you, as a junior data analyst, can help Bellabeat achieve them.

Products

- Bellabeat app: The Bellabeat app provides users with health data related to their activity, sleep, stress, menstrual cycle, and mindfulness habits. This data can help users better understand their current habits and make healthy decisions. The Bellabeat app connects to their line of smart wellness products.
- Leaf: Bellabeat's classic wellness tracker can be worn as a bracelet, necklace, or clip. The Leaf tracker connects to the Bellabeat app to track activity, sleep, and stress.
- Time: This wellness watch combines the timeless look of a classic timepiece with smart technology to track user activity, sleep, and stress. The Time watch connects to the Bellabeat app to provide you with insights into your daily wellness.
- Spring: This is a water bottle that tracks daily water intake using smart technology to ensure that you are appropriately hydrated throughout the day. The Spring bottle connects to the Bellabeat app to track your hydration levels.

Installing and loading common packages and libraries We install and load packages along the way as we may discover we need different packages after we start our analysis. If you already have some of these packages installed and loaded, you can skip those ones - or you can choose to run those specific lines of code anyway. It may take a few moments to run.

Ask phase

Questions guiding the analysis

- What are some trends in the smart device usage?
- How could these trends apply to bellabeat customers
- How could these trends help influence Bellabeat marketing strategy?

Business Task

Analyze fitness data from smart devices to spot patterns in consumer usage, then make suggestions on how these patterns can influence Bellabeat's marketing strategy.

Prepare phase

This dataset was created by participants in a distributed survey who used Amazon Mechanical Turk between December 3, 2016, and December 5, 2016. Thirty eligible Fitbit users agreed to submit their personal tracker data, which included minute-level output for heart rate, sleep, and physical activity monitoring.

Reports can be analysed individually using the export session ID (column A) or timestamp (column B). The difference in output reflects the use of various Fitbit tracker models and personal monitoring habits and preferences.

Data Organization

There are 18 CSV files in the structured dataset. It is set up in a long data format, which means that none of the values in the first column repeat.

Data Credibility/Limitations

Since the statistics were obtained from a third party, it is difficult to confirm their accuracy. Additionally, important participant demographics were left out, making it impossible to determine whether or not the data is representative of the population. Only current Fitbit users are represented in the data, which could lead to sample bias. Due to only 33 people reporting their data, the sample size is extremely tiny. Since the previous update was made two years ago, the data is no longer accurate. I will however continue to examine the dataset to find trends in the users' everyday usage.

Data liscensing/privacy

Our dataset's metadata includes information about licencing and privacy. Open-source data with a CCO:public domain licence. The general public has access to it and can use and reuse it.

```
# Using the install.packages() function to install packages

install.packages("tidyverse")        # collection of R packages designed for data science
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("dplyr")            # for transforming data sets
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("janitor")          # for examining and cleaning dirty data
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("lubridate")        # for date & time formats
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("ggpubr")           # for creating and customizing ggplot2
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("waffle")           # for waffle charts
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("scales")           # scaling used by ggplots
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
install.packages("RColorBrewer")     # for color palette
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
# Using the library () function to load packages

library(tidyverse)          # collection of R packages designed for data science
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)              # for transforming data sets
library(janitor)            # for examining and cleaning dirty data
```

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

```
library(lubridate)          # for date & time formats
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
```

```
##
##     date, intersect, setdiff, union
```

```
library(ggpubr)             # for creating and customizing ggplot2
library(waffle)             # for waffle charts
library(scales)             # scaling used by ggplots
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(RColorBrewer)       # for color palette

# Reading csv files

activity_daily <- read_csv(file= "/cloud/project/bellabeat/Fitabase Data 4.12.16-5.12.16/dailyActivity_
```

```
## 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.
```

```
calories_daily <- read_csv(file= "/cloud/project/bellabeat/Fitabase Data 4.12.16-5.12.16/dailyCalories_
```

```
## 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.
```

```
intensities_daily <- read_csv(file= "/cloud/project/bellabeat/Fitabase Data 4.12.16-5.12.16/dailyIntens
```

```
## 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.
```

```
steps_daily <- read_csv(file= "/cloud/project/bellabeat/Fitabase Data 4.12.16-5.12.16/dailySteps_merged
```

```
## Rows: 940 Columns: 3
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityDay
## 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.
```

```
steps_hourly <- read_csv(file= "/cloud/project/bellabeat/Fitabase Data 4.12.16-5.12.16/hourlySteps_merge
```

```
## 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.
```

```
sleep_daily <- read_csv(file= "/cloud/project/bellabeat/Fitabase Data 4.12.16-5.12.16/sleepDay_merged.cs
```

```
## Rows: 413 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): SleepDay
## dbl (4): Id, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
weight <- read_csv(file= "/cloud/project/bellabeat/Fitabase Data 4.12.16-5.12.16/weightLogInfo_merged.cs
```

```
## 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.
```

```
# Using the head() function to get a snapshot of each data set.

head(activity_daily)
```

```
## # A tibble: 6 x 15
##       Id Activ~1 Total~2 Total~3 Track~4 Logge~5 VeryA~6 Moder~7 Light~8 Seden~9
##    <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1.50e9 4/12/2~   13162    8.5     8.5        0    1.88   0.550    6.06       0
## 2 1.50e9 4/13/2~   10735    6.97    6.97       0    1.57   0.690    4.71       0
## 3 1.50e9 4/14/2~   10460    6.74    6.74       0    2.44   0.400    3.91       0
## 4 1.50e9 4/15/2~    9762    6.28    6.28       0    2.14   1.26     2.83       0
## 5 1.50e9 4/16/2~   12669    8.16    8.16       0    2.71   0.410    5.04       0
## 6 1.50e9 4/17/2~    9705    6.48    6.48       0    3.19   0.780    2.51       0
## # ... with 5 more variables: VeryActiveMinutes <dbl>,
## #   FairlyActiveMinutes <dbl>, LightlyActiveMinutes <dbl>,
```

```
## #   SedentaryMinutes <dbl>, Calories <dbl>, and abbreviated variable names
## #   1: ActivityDate, 2: TotalSteps, 3: TotalDistance, 4: TrackerDistance,
## #   5: LoggedActivitiesDistance, 6: VeryActiveDistance,
## #   7: ModeratelyActiveDistance, 8: LightActiveDistance,
## #   9: SedentaryActiveDistance
```

```
head(calories_daily)
```

```
## # A tibble: 6 x 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
```

```
head(intensities_daily)
```

```
## # A tibble: 6 x 10
##       Id Activ~1 Seden~2 Light~3 Fairl~4 VeryA~5 Seden~6 Light~7 Moder~8 VeryA~9
##    <dbl> <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1.50e9 4/12/2~     728     328      13      25       0    6.06   0.550    1.88
## 2 1.50e9 4/13/2~     776     217      19      21       0    4.71   0.690    1.57
## 3 1.50e9 4/14/2~    1218     181      11      30       0    3.91   0.400    2.44
## 4 1.50e9 4/15/2~     726     209      34      29       0    2.83   1.26     2.14
## 5 1.50e9 4/16/2~     773     221      10      36       0    5.04   0.410    2.71
## 6 1.50e9 4/17/2~     539     164      20      38       0    2.51   0.780    3.19
## # ... with abbreviated variable names 1: ActivityDay, 2: SedentaryMinutes,
## #   3: LightlyActiveMinutes, 4: FairlyActiveMinutes, 5: VeryActiveMinutes,
## #   6: SedentaryActiveDistance, 7: LightActiveDistance,
## #   8: ModeratelyActiveDistance, 9: VeryActiveDistance
```

```
head(steps_daily)
```

```
## # A tibble: 6 x 3
##           Id ActivityDay StepTotal
##        <dbl> <chr>           <dbl>
## 1 1503960366 4/12/2016       13162
## 2 1503960366 4/13/2016       10735
## 3 1503960366 4/14/2016       10460
## 4 1503960366 4/15/2016        9762
## 5 1503960366 4/16/2016       12669
## 6 1503960366 4/17/2016        9705
```

```
head(steps_hourly)
```

```
## # A tibble: 6 x 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
```

```
head(sleep_daily)
```

```
## # A tibble: 6 x 5
##           Id SleepDay              TotalSleepRecords TotalMinutesAsleep TotalT~1
##        <dbl> <chr>                             <dbl>              <dbl>    <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM                1                327      346
## 2 1503960366 4/13/2016 12:00:00 AM                2                384      407
## 3 1503960366 4/15/2016 12:00:00 AM                1                412      442
## 4 1503960366 4/16/2016 12:00:00 AM                2                340      367
## 5 1503960366 4/17/2016 12:00:00 AM                1                700      712
## 6 1503960366 4/19/2016 12:00:00 AM                1                304      320
## # ... with abbreviated variable name 1: TotalTimeInBed
```

```
head(weight)
```

```
## # A tibble: 6 x 8
##           Id Date                  WeightKg Weight~1   Fat   BMI IsMan~2   LogId
##        <dbl> <chr>                    <dbl>    <dbl> <dbl> <dbl> <lgl>     <dbl>
## 1 1503960366 5/2/2016 11:59:59 PM      52.6     116.    22  22.6 TRUE    1.46e12
## 2 1503960366 5/3/2016 11:59:59 PM      52.6     116.    NA  22.6 TRUE    1.46e12
## 3 1927972279 4/13/2016 1:08:52 AM     134.      294.    NA  47.5 FALSE   1.46e12
## 4 2873212765 4/21/2016 11:59:59 PM     56.7     125.    NA  21.5 TRUE    1.46e12
## 5 2873212765 5/12/2016 11:59:59 PM     57.3     126.    NA  21.7 TRUE    1.46e12
## 6 4319703577 4/17/2016 11:59:59 PM     72.4     160.    25  27.5 TRUE    1.46e12
## # ... with abbreviated variable names 1: WeightPounds, 2: IsManualReport
```

```
# Using the str() function to preview  the structure of each data set.

str(activity_daily)
```

```
## 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>
```

```
str(calories_daily)
```

```
## spec_tbl_df [940 x 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>
```

```
str(intensities_daily)
```

```
## spec_tbl_df [940 x 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>
```

```
str(steps_daily)
```

```
## spec_tbl_df [940 x 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" ...
##  $ StepTotal  : num [1:940] 13162 10735 10460 9762 12669 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDay = col_character(),
##   ..   StepTotal = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(steps_hourly)
```

```
## spec_tbl_df [22,099 x 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"
##  $ StepTotal   : num [1:22099] 373 160 151 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityHour = col_character(),
##   ..   StepTotal = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(sleep_daily)
```

```
## 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>
```

```
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>
```

```
# To verify that the datasets calories_daily, intensities_daily, and steps_daily are subsets found in a

all(calories_daily %in% activity_daily)
```

```
## [1] TRUE
```

```
all(intensities_daily %in% activity_daily)
```

```
## [1] TRUE
```

```
all(steps_daily %in% activity_daily)
```

```
## [1] TRUE
```

```
# Using the rm() function to remove the data sets that are subsets of the activity_daily data set.

rm(calories_daily,intensities_daily,steps_daily)

# Changing the the column names

colnames(activity_daily) <- c("id","date","steps","distance","tracker_distance","logged_distance","acti
                              "light_distance","sedentary_distance","active_min","fair_min","light_min"
                              "sedentary_min","calories")

colnames(sleep_daily) <- c("id","date","sleep_records","minutes_asleep","time_in_bed")

colnames(steps_hourly) <- c("id","date","steps")
```

Process phase

In this phase, I'll make sure my data is accurate by cleaning it up, which includes date formatting, looking for duplicates, ensuring that column names are consistent, looking for missing data, etc. My data will be prepared and perfectly suited to the business task thanks to this.

I'll be utilising the programming language R throughout for all of my data cleaning, analysis, and visualisation because I enjoyed studying it and I want to put it to use and show forth my skills.

```
# Cleaning the the column names

clean_names(activity_daily)
```

```
## # A tibble: 940 x 15
##           id date  steps dista~1 track~2 logge~3 activ~4 moder~5 light~6 seden~7
##        <dbl> <chr> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1    1.50e9 4/12~ 13162    8.5     8.5        0    1.88   0.550    6.06       0
##  2    1.50e9 4/13~ 10735    6.97    6.97       0    1.57   0.690    4.71       0
##  3    1.50e9 4/14~ 10460    6.74    6.74       0    2.44   0.400    3.91       0
```

```
##  4    1.50e9 4/15~  9762    6.28    6.28       0    2.14   1.26     2.83       0
##  5    1.50e9 4/16~ 12669    8.16    8.16       0    2.71   0.410    5.04       0
##  6    1.50e9 4/17~  9705    6.48    6.48       0    3.19   0.780    2.51       0
##  7    1.50e9 4/18~ 13019    8.59    8.59       0    3.25   0.640    4.71       0
##  8    1.50e9 4/19~ 15506    9.88    9.88       0    3.53   1.32     5.03       0
##  9    1.50e9 4/20~ 10544    6.68    6.68       0    1.96   0.480    4.24       0
## 10    1.50e9 4/21~  9819    6.34    6.34       0    1.34   0.350    4.65       0
## # ... with 930 more rows, 5 more variables: active_min <dbl>, fair_min <dbl>,
## #   light_min <dbl>, sedentary_min <dbl>, calories <dbl>, and abbreviated
## #   variable names 1: distance, 2: tracker_distance, 3: logged_distance,
## #   4: active_distance, 5: moderate_distance, 6: light_distance,
## #   7: sedentary_distance
```

```
clean_names(sleep_daily)
```

```
## # A tibble: 413 x 5
##            id date                   sleep_records minutes_asleep time_in_bed
##         <dbl> <chr>                          <dbl>          <dbl>       <dbl>
##  1 1503960366 4/12/2016 12:00:00 AM             1            327         346
##  2 1503960366 4/13/2016 12:00:00 AM             2            384         407
##  3 1503960366 4/15/2016 12:00:00 AM             1            412         442
##  4 1503960366 4/16/2016 12:00:00 AM             2            340         367
##  5 1503960366 4/17/2016 12:00:00 AM             1            700         712
##  6 1503960366 4/19/2016 12:00:00 AM             1            304         320
##  7 1503960366 4/20/2016 12:00:00 AM             1            360         377
##  8 1503960366 4/21/2016 12:00:00 AM             1            325         364
##  9 1503960366 4/23/2016 12:00:00 AM             1            361         384
## 10 1503960366 4/24/2016 12:00:00 AM             1            430         449
## # ... with 403 more rows
```

```
clean_names(steps_hourly)
```

```
## # A tibble: 22,099 x 3
##            id date                   steps
##         <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
## # ... with 22,089 more rows
```

```
# Using the colnames() to view the names of all the columns found in each data set

colnames(activity_daily)
```

```
##  [1] "id"                 "date"               "steps"
##  [4] "distance"           "tracker_distance"   "logged_distance"
##  [7] "active_distance"    "moderate_distance"  "light_distance"
## [10] "sedentary_distance" "active_min"         "fair_min"
## [13] "light_min"          "sedentary_min"      "calories"
```

```
colnames(sleep_daily)
```

```
## [1] "id"             "date"           "sleep_records"  "minutes_asleep"
## [5] "time_in_bed"
```

```
colnames(steps_hourly)
```

```
## [1] "id"    "date"  "steps"
```

```
# Using the sum() function to view the number of duplicates in each data set

sum(duplicated(activity_daily))
```

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

```
sum(duplicated(sleep_daily))
```

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

```
sum(duplicated(steps_hourly))
```

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

```
# Using the unique function return only unique
sleep_daily <- unique(sleep_daily)

# Using the sum() function to view the number of duplicates in the sleep_daily data set

sum(duplicated(sleep_daily))
```

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

```
# Double checking the data structure

head(activity_daily)
```

```
## # A tibble: 6 x 15
##           id date  steps dista~1 track~2 logge~3 activ~4 moder~5 light~6 seden~7
##        <dbl> <chr> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1503960366 4/12~ 13162    8.5     8.5        0    1.88   0.550    6.06       0
## 2 1503960366 4/13~ 10735    6.97    6.97       0    1.57   0.690    4.71       0
## 3 1503960366 4/14~ 10460    6.74    6.74       0    2.44   0.400    3.91       0
## 4 1503960366 4/15~  9762    6.28    6.28       0    2.14   1.26     2.83       0
## 5 1503960366 4/16~ 12669    8.16    8.16       0    2.71   0.410    5.04       0
## 6 1503960366 4/17~  9705    6.48    6.48       0    3.19   0.780    2.51       0
## # ... with 5 more variables: active_min <dbl>, fair_min <dbl>, light_min <dbl>,
## #   sedentary_min <dbl>, calories <dbl>, and abbreviated variable names
## #   1: distance, 2: tracker_distance, 3: logged_distance, 4: active_distance,
## #   5: moderate_distance, 6: light_distance, 7: sedentary_distance
```

```
head(sleep_daily)
```

```
## # A tibble: 6 x 5
##           id date                   sleep_records minutes_asleep time_in_bed
##        <dbl> <chr>                          <dbl>          <dbl>       <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM             1            327         346
## 2 1503960366 4/13/2016 12:00:00 AM             2            384         407
## 3 1503960366 4/15/2016 12:00:00 AM             1            412         442
## 4 1503960366 4/16/2016 12:00:00 AM             2            340         367
```

```
## 5 1503960366 4/17/2016 12:00:00 AM             1            700         712
## 6 1503960366 4/19/2016 12:00:00 AM             1            304         320
```

```
head(steps_hourly)
```

```
## # A tibble: 6 x 3
##           id date                  steps
##        <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
```

Analysis phase

I will not be using the sleep_day and daily_intensities for analysis because while exploring the datasets, i noticed that the daily_activity table contains the consolidated information from both tables.

Summary statistics:

Analyzing summary statistics from daily_activity and sleep_day tables

```
# Using the mutate() function to change the data type of the date column from chr to date

activity_daily <- activity_daily %>%
  mutate(date= as_date(date, format= "%m/%d/%Y"))

sleep_daily <- sleep_daily %>%
  mutate(date= as.POSIXct(date, format= "%m/%d/%Y  %I:%M:%S %p", tz= Sys.timezone()))

steps_hourly <- steps_hourly %>%
  mutate(date= as.POSIXct(date, format="%m/%d/%Y %I:%M:%S %p", tz= Sys.timezone()))


# Verifying the date column format changes
head(activity_daily)
```

```
## # A tibble: 6 x 15
##           id date       steps distance tracker~1 logge~2 activ~3 moder~4 light~5
##        <dbl> <date>     <dbl>    <dbl>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1503960366 2016-04-12 13162     8.5       8.5         0    1.88   0.550    6.06
## 2 1503960366 2016-04-13 10735     6.97      6.97        0    1.57   0.690    4.71
## 3 1503960366 2016-04-14 10460     6.74      6.74        0    2.44   0.400    3.91
## 4 1503960366 2016-04-15  9762     6.28      6.28        0    2.14   1.26     2.83
## 5 1503960366 2016-04-16 12669     8.16      8.16        0    2.71   0.410    5.04
## 6 1503960366 2016-04-17  9705     6.48      6.48        0    3.19   0.780    2.51
## # ... with 6 more variables: sedentary_distance <dbl>, active_min <dbl>,
## #   fair_min <dbl>, light_min <dbl>, sedentary_min <dbl>, calories <dbl>, and
## #   abbreviated variable names 1: tracker_distance, 2: logged_distance,
## #   3: active_distance, 4: moderate_distance, 5: light_distance
```

```
head(sleep_daily)
```

```
## # A tibble: 6 x 5
##           id date                sleep_records minutes_asleep time_in_bed
```

```
##        <dbl> <dttm>                     <dbl>          <dbl>       <dbl>
## 1 1503960366 2016-04-12 00:00:00            1            327         346
## 2 1503960366 2016-04-13 00:00:00            2            384         407
## 3 1503960366 2016-04-15 00:00:00            1            412         442
## 4 1503960366 2016-04-16 00:00:00            2            340         367
## 5 1503960366 2016-04-17 00:00:00            1            700         712
## 6 1503960366 2016-04-19 00:00:00            1            304         320
```

```
head(steps_hourly)
```

```
## # A tibble: 6 x 3
##           id date                steps
##        <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
```

```
# Using the merge() function to combine the two data sets

activity_merged <- merge(activity_daily, sleep_daily, by= c("id","date"), all.x = TRUE)


# Verifying the merge
head(activity_merged)
```

```
##           id       date steps distance tracker_distance logged_distance
## 1 1503960366 2016-04-12 13162     8.50             8.50               0
## 2 1503960366 2016-04-13 10735     6.97             6.97               0
## 3 1503960366 2016-04-14 10460     6.74             6.74               0
## 4 1503960366 2016-04-15  9762     6.28             6.28               0
## 5 1503960366 2016-04-16 12669     8.16             8.16               0
## 6 1503960366 2016-04-17  9705     6.48             6.48               0
##   active_distance moderate_distance light_distance sedentary_distance
## 1            1.88              0.55           6.06                  0
## 2            1.57              0.69           4.71                  0
## 3            2.44              0.40           3.91                  0
## 4            2.14              1.26           2.83                  0
## 5            2.71              0.41           5.04                  0
## 6            3.19              0.78           2.51                  0
##   active_min fair_min light_min sedentary_min calories sleep_records
## 1         25       13       328           728     1985             1
## 2         21       19       217           776     1797             2
## 3         30       11       181          1218     1776            NA
## 4         29       34       209           726     1745             1
## 5         36       10       221           773     1863             2
## 6         38       20       164           539     1728             1
##   minutes_asleep time_in_bed
## 1            327         346
## 2            384         407
## 3             NA          NA
## 4            412         442
## 5            340         367
```

```
## 6            700         712
```

```
# summary activity_merged data:

summary(activity_merged)
```

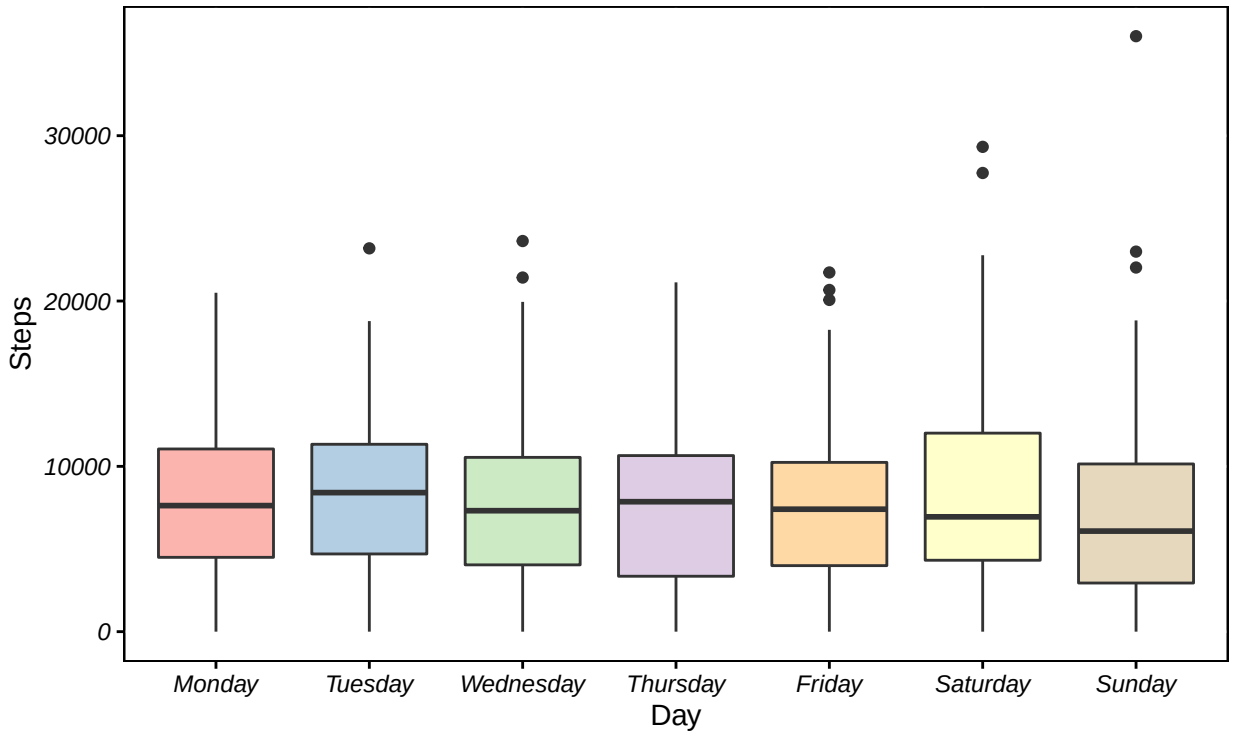
```
##        id                 date                steps          distance
##  Min.   :1.504e+09   Min.   :2016-04-12   Min.   :    0   Min.   : 0.000
##  1st Qu.:2.320e+09   1st Qu.:2016-04-19   1st Qu.: 3790   1st Qu.: 2.620
##  Median :4.445e+09   Median :2016-04-26   Median : 7406   Median : 5.245
##  Mean   :4.855e+09   Mean   :2016-04-26   Mean   : 7638   Mean   : 5.490
##  3rd Qu.:6.962e+09   3rd Qu.:2016-05-04   3rd Qu.:10727   3rd Qu.: 7.713
##  Max.   :8.878e+09   Max.   :2016-05-12   Max.   :36019   Max.   :28.030
##
##  tracker_distance logged_distance  active_distance  moderate_distance
##  Min.   : 0.000   Min.   :0.0000   Min.   : 0.000   Min.   :0.0000
##  1st Qu.: 2.620   1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.:0.0000
##  Median : 5.245   Median :0.0000   Median : 0.210   Median :0.2400
##  Mean   : 5.475   Mean   :0.1082   Mean   : 1.503   Mean   :0.5675
##  3rd Qu.: 7.710   3rd Qu.:0.0000   3rd Qu.: 2.053   3rd Qu.:0.8000
##  Max.   :28.030   Max.   :4.9421   Max.   :21.920   Max.   :6.4800
##
##  light_distance   sedentary_distance   active_min        fair_min
##  Min.   : 0.000   Min.   :0.000000   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 1.945   1st Qu.:0.000000   1st Qu.:  0.00   1st Qu.:  0.00
##  Median : 3.365   Median :0.000000   Median :  4.00   Median :  6.00
##  Mean   : 3.341   Mean   :0.001606   Mean   : 21.16   Mean   : 13.56
##  3rd Qu.: 4.782   3rd Qu.:0.000000   3rd Qu.: 32.00   3rd Qu.: 19.00
##  Max.   :10.710   Max.   :0.110000   Max.   :210.00   Max.   :143.00
##
##    light_min     sedentary_min       calories    sleep_records
##  Min.   :  0.0   Min.   :   0.0   Min.   :   0   Min.   :1.000
##  1st Qu.:127.0   1st Qu.: 729.8   1st Qu.:1828   1st Qu.:1.000
##  Median :199.0   Median :1057.5   Median :2134   Median :1.000
##  Mean   :192.8   Mean   : 991.2   Mean   :2304   Mean   :1.119
##  3rd Qu.:264.0   3rd Qu.:1229.5   3rd Qu.:2793   3rd Qu.:1.000
##  Max.   :518.0   Max.   :1440.0   Max.   :4900   Max.   :3.000
##                                                  NA's   :530
##  minutes_asleep   time_in_bed
##  Min.   : 58.0   Min.   : 61.0
##  1st Qu.:361.0   1st Qu.:403.8
##  Median :432.5   Median :463.0
##  Mean   :419.2   Mean   :458.5
##  3rd Qu.:490.0   3rd Qu.:526.0
##  Max.   :796.0   Max.   :961.0
##  NA's   :530     NA's   :530
```

Share phase

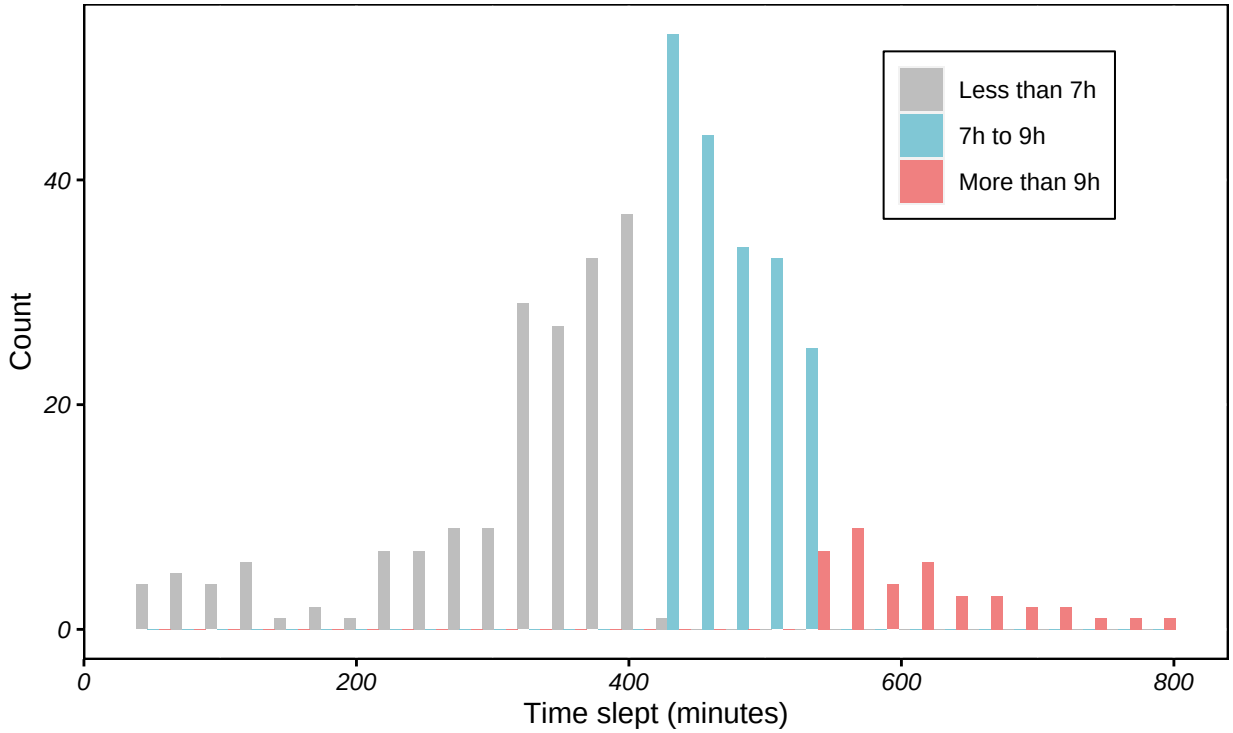
```
# Setting up custom themes for ggplot2

custom_theme <- function() {
  theme(
    panel.border = element_rect(colour = "black",
```

```
                                fill = NA,
                                linetype = 1),
    panel.background = element_rect(fill = "white",
                                    color = 'grey50'),
    panel.grid.minor.y = element_blank(),
    axis.text = element_text(colour = "black",
                             face = "italic",
                             family = "Helvetica"),
    axis.title = element_text(colour = "black",
                              family = "Helvetica"),
    axis.ticks = element_line(colour = "black"),
    plot.title = element_text(size=23,
                              hjust = 0.5,
                              family = "Helvetica"),
    plot.subtitle=element_text(size=16,
                               hjust = 0.5),
    plot.caption = element_text(colour = "black",
                                face = "italic",
                                family = "Helvetica")
  )
}


# Correlation between calories and steps

activity_merged %>%
  group_by(steps, calories) %>%
  ggplot(aes(x = steps, y = calories, color = calories)) +
  geom_point() +
  geom_smooth(formula = y ~ x, method = "lm")+
  custom_theme() +
  theme(legend.position = c(.8, .3),
        legend.spacing.y = unit(1, "mm"),
        panel.border = element_rect(colour = "black", fill=NA),
        legend.background = element_blank(),
        legend.box.background = element_rect(colour = "black")) +
  labs(title = 'Calories burned by total steps taken',
       y = 'Calories',
       x = 'Total Steps',
       caption = 'Data Source: FitBit Fitness Tracker Data')
```


```
# Analyzing user weekly activity and steps

activity_merged %>%
  mutate(weekdays = weekdays(date)) %>%
  select(weekdays, steps) %>%
  mutate(weekdays = factor(weekdays, levels = c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
  drop_na() %>%
  ggplot(aes(weekdays, steps, fill = weekdays)) +
  geom_boxplot() +
  custom_theme() +
  scale_fill_brewer(palette="Pastel1") +
  theme(legend.position="none") +
  labs(
    title = "Weekly user activity",
    x = "Day", line,
    y = "Steps",
    caption = 'Data Source: FitBit Fitness Tracker Data 2016'
  )
```




```
# Establishing user nightly sleep cycle by minutes

activity_merged %>%
  select(minutes_asleep) %>%
  drop_na() %>%
  mutate(sleep_quality = ifelse(minutes_asleep <= 420, 'Less than 7h',
                                ifelse(minutes_asleep <= 540, '7h to 9h',
                                       'More than 9h'))) %>%
  mutate(sleep_quality = factor(sleep_quality,
                                levels = c('Less than 7h','7h to 9h',
                                           'More than 9h'))) %>%
  ggplot(aes(x = minutes_asleep, fill = sleep_quality)) +
  geom_histogram(position = 'dodge', bins = 30) +
  custom_theme() +
  scale_fill_manual(values=c("grey", "#80c7d5", "lightcoral")) +
  theme(legend.position = c(.80, .80),
        legend.title = element_blank(),
        legend.spacing.y = unit(0, "mm"),
        panel.border = element_rect(colour = "black", fill=NA),
        legend.background = element_blank(),
        legend.box.background = element_rect(colour = "black")) +
  labs(
    title = "Sleep distribution",
    x = "Time slept (minutes)",
    y = "Count",
    caption = 'Data Source: FitBit Fitness Tracker Data'
  )
```



Data Source: FitBit Fitness Tracker Data

```
# Separating the  date and time column in steps_hourly data set

steps_hourly <- steps_hourly %>%
  separate(date, into= c("date", "time"), sep = " ") %>%
  mutate(date= ymd (date))

head(steps_hourly)
```

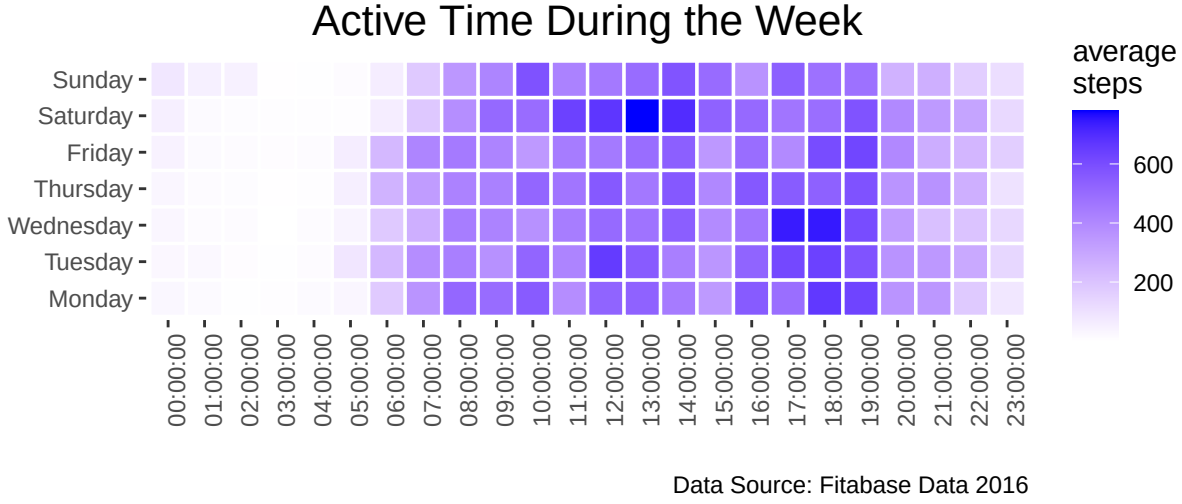
```
## # A tibble: 6 x 4
##           id date       time     steps
##        <dbl> <date>     <chr>    <dbl>
## 1 1503960366 2016-04-12 00:00:00   373
## 2 1503960366 2016-04-12 01:00:00   160
## 3 1503960366 2016-04-12 02:00:00   151
## 4 1503960366 2016-04-12 03:00:00     0
## 5 1503960366 2016-04-12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

```
# Adding a weekday column to the data set

steps_weekday <- (steps_hourly)%>%
  mutate(weekday= weekdays(date))%>%
  group_by (weekday,time) %>%
  summarize(average_steps= mean(steps), .groups = 'drop')

steps_weekday$weekday <- ordered(steps_weekday$weekday,
                                 levels=c("Monday", "Tuesday", "Wednesday","Thursday","Friday", "Saturda
```

```
head(steps_weekday)
```

```
## # A tibble: 6 x 3
##   weekday time     average_steps
##   <ord>   <chr>            <dbl>
## 1 Friday  00:00:00          44.3
## 2 Friday  01:00:00          19.0
## 3 Friday  02:00:00          16.3
## 4 Friday  03:00:00          10.8
## 5 Friday  04:00:00          14.4
## 6 Friday  05:00:00          61.2
```

```
# Creating a heatmap to better visualize the volume and time of activity within a weekday

ggplot(steps_weekday, aes(x= time, y= weekday,
                          fill= average_steps)) +
  theme(axis.text.x= element_text(angle = 90))+
  labs(title= "Active Time During the Week",
       x=" ", y=" ",fill = "average\nsteps",
       caption= 'Data Source: Fitabase Data 2016')+
  scale_fill_gradient(low= "white", high="blue")+
  geom_tile(color= "white",lwd =.6,linetype =1)+
  coord_fixed()+
  theme(plot.title= element_text(hjust= 0.5,vjust= 0.8, size=16),
        panel.background= element_blank())
```



Act Phase

Recommendations:

- Sleep Journal: The Bellabeat app may offer a customised sleep journal for each user in addition to the health information it offers. The user will be able to see trends and alter their behaviour to improve their sleep if they use this notebook to track their sleep for a certain period of time.
- Client segmentation Being in the healthcare industry, Bellabeat Company understands the value of "understanding your consumer." Bellabeat Company may take advantage of the growing need in the healthcare sector for individualised treatment and value-based care. It would be wonderful to include essential consumer demographic information like age and occupation!

- Customized notifications and alarms: Bellabeat can use the app's notifications and alarms to remind users to go to bed or exercise. Additionally, this needs to be tailored.
- Periodic report: Users can receive thorough evaluations of their weekly performance, which will both inspire them and help them identify their strongest points for development.

Conclusion

Bellabeat Firm understands the value of data collecting and analysis in enhancing business choices. Bellabeat Company is a high-tech company with significant potential to become a worldwide smart device market. In this case study, I used fitbit data to learn how Bellabeat customers interact with non-Bellabeat products, spot some usage patterns for smart devices, analyse how Bellabeat customers might be affected by these patterns, and then offer suggestions that could have an impact on Bellabeat's marketing strategy.