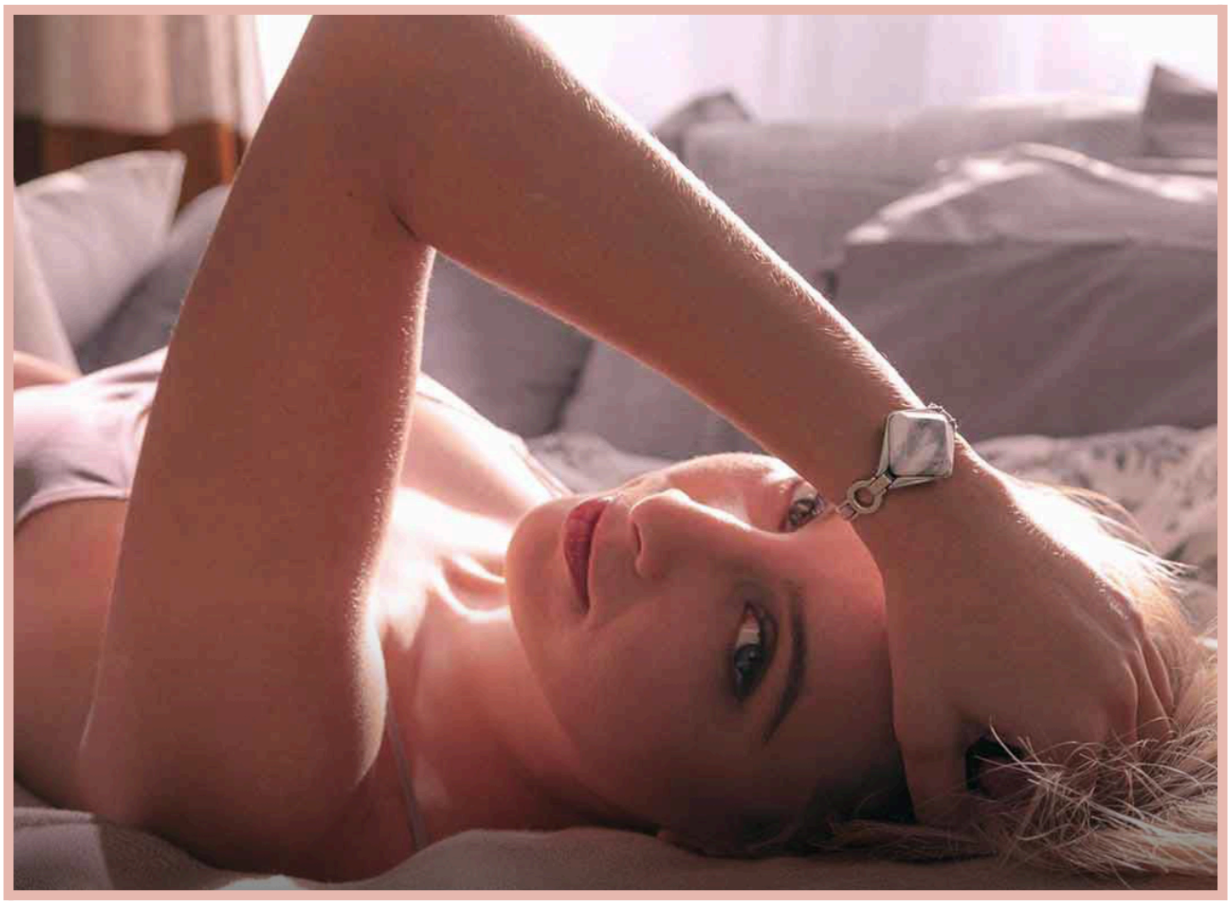
How Can a Wellness Technology Company Play It Smart?

bellabeat



Prepared by

Ryanne

Table of Contents

Introduction

Bellabeat, a high-tech manufacturer of health-focused smart products for women, develops beautifully designed technology that informs and inspires women around the world. Collecting data on activity, sleep, stress, and reproductive health has allowed Bellabeat to empower women with knowledge about their own health and habits.

Since it was founded in 2013, Bellabeat has grown rapidly and quickly positioned itself as a tech-driven wellness company for women. As a junior data analyst in Bellabeat marketing analytics team, I joined this team six months ago and have been busy learning about Bellabeat's mission and business goals and helping the company to achieve them.

Our products comprise of:

- **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:** classic wellness tracker can be worn as a bracelet, necklace, or a clip that connects to the Bellabeat app to track activity, sleep, and stress.
- **Time:** 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:** water bottle that tracks daily water intake using smart technology to ensure that you are appropriately hydrated throughout the day. It connects to the Bellabeat app to track your hydration levels.
- **Bellabeat membership:** Bellabeat also offers a subscription-based membership program for users that gives users 24/7 access to fully personalized guidance on nutrition, activity, sleep, health and beauty, and mindfulness based on their lifestyle and goals.

Urška Sršen, Cofounder and Chief Creative Officer of Bellabeat, believes that analyzing smart device fitness data could help unlock new growth opportunities for the company.

Sršen asks to analyze smart device usage data in order to gain insight into how consumers use non-Bellabeat smart devices. She then wants to select one Bellabeat product to apply these insights into a presentation.

I will follow the steps of the data analysis process: Ask, Prepare, Process, Analyze, Share, and Act, then apply the insight gained from the data analysis into actionable strategies. I'm confident that it will contribute to the growth and success of Bellabeat in the competitive smart devices and fitness market.

1. Ask

The purpose of this step is to define the problem or question to answer with data. We will consider understanding the stakeholder expectations by setting the business tasks, identifying key stakeholders, formulating key questions, and establishing deliverables.

1.1. Business Tasks

The aim of the project is to analyze smart device usage data in order to gain insight into how consumers use non-Bellabeat smart devices to help guide marketing strategy for the company.

1.2. Key Stakeholders

- Urška Sršen: Bellabeat's co founder and Chief Creative Officer
- Sando Mur: Mathematician and Bellabeat co founder; 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.

1.3. Key Questions

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

1.4. Deliverables

- A clear summary in the business task
- A description of data sources used
- Documentation of any cleaning or manipulation of data
- A summary of the analysis
- High-level content recommendations based on the analysis

2. Prepare

The purpose of this step is to decide what data we need to collect in order to answer the questions and how to organize it so that it is useful.

2.1. Data Used

The data used is Fitbit Fitness Tracker from Kaggle generated from a survey via Amazon Mechanical Turk between 12 March to 12 May 2016.

It contains personal fitness tracker information of 30 (thirty) Fitbit users who consented to the submission of personal tracker data, including daily activity, steps and heart rate. The data is stored in 18 CSV files.

2.2. Data Integrity

The data was collected in 2016 and might not be relevant anymore due to a change in the user's activity, sleeping and health habits in the past years. The sample size of 30 FitBit users is not representative of the entire population of female Fitbit users. The data is not original because it's collected in a third party survey.

2.3. Is data ROCCC ?

A good data source is reliable, original, comprehensive, current, and cited (ROCCC). Here is the data credibility check using ROCCC Method:

- **Not reliable - T**he sample size is too small (it only has 30 respondents), so it might not reflect the overall population
- **Not original** - The dataset is third party data
- **Comprehensive** - The parameters match with Bellabeat products, but we miss some information regarding age, gender, etc.
- **Not current** - The data was collected in 2016
- **Cited** - These datasets are considered generated from a survey via Amazon Mechanical Turk, so the resource is considered properly cited.

3. Process

The purpose of this step is to clean up the data to get rid of any possible errors, inaccuracies, or inconsistencies. In this phase, I leverage both Google Sheets and the R Programming Language to process the data. This approach allows me to use the Google Sheets for initial data exploration and cleaning, and then transition to the statistical and analytical capabilities of R for more advanced analysis.

3.1. Google Sheets

I will start with basic data exploration and cleaning in Google Sheets for quick insights. Google Sheets will be used to process small datasets, specifically those less than 1 MB.

The following activities will be undertaken to process the data using Google Sheets:

- Apply Filters to ensure there are no negative values.
- Convert the 'Id' field to the text data type.
- Utilize a pivot table to examine the summary of the data. There are some notes:

  We are unable to track the daily activity of users every day. For example in file dailyActivity_merged, the activity of a user with ID 4057192912 only recorded from April 12 until April 15, 2016.

  This dataset has some limitations, including:
  - Outdated Data: The data was collected in 2016, so the market trends might have significantly changed and the findings may not be relevant to the current market.
  - Narrow Time Frame: The data only covers the period between 12 March to 12 May 2016, which is a short duration for establishing marketing strategies based on this data.
  - Small Sample Size: It only includes a maximum of 33 participants users.
  - No Demographic Info: The data lacks any demographic information about the users, such as age, height, gender, etc.

- Lack of Specific Information: There is no specific information about the device type or activity type

3.2. R Programming Language

R programming language will be used to process the large data (more than 1 MB) and for advanced analysis. The following activities will be undertaken to process the data using R:

3.2.1. Installing the package

Installing and loading common packages and libraries

- The 'tidyverse' is a collection of R packages that share a common philosophy and are designed to work well together. It includes a set of packages for data manipulation, visualization, and analysis.

```
install.packages('tidyverse')
library(tidyverse)
```

- The 'dplyr' is a package for data manipulation that provides a set of functions for filtering, selecting, grouping, summarizing, and joining data.

```
> install.packages("dplyr")
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.0'
```

- The 'ggplot2' is a powerful and flexible package for creating static, interactive, and multi-layered data visualization.

```
install.packages("ggplot2")
library(ggplot2)
```

- The 'janitor' package is designed to clean and organize messy data sets such as clean_names() function.

```
install.packages("janitor")
library(janitor)
```

3.2.2. Upload CSV files to R

Here are 18 files that upload on the R project:

dailyActivity_merged.csv	111 KB
dailyCalories_merged.csv	25 KB
dailyIntensities_merged.csv	71 KB
dailySteps_merged.csv	25 KB
heartrate_seconds_merged.csv	89,6 MB
hourlyCalories_merged.csv	801 KB
hourlyIntensities_merged.csv	899 KB
hourlySteps_merged.csv	797 KB
minuteCalories...rrow_merged.csv	66,4 MB
minuteCaloriesWide_merged.csv	23 MB
minuteIntensitie...row_merged.csv	46,4 MB
minuteIntensitie...ide_merged.csv	3,3 MB
minuteMETsNarrow_merged.csv	47,7 MB
minuteSleep_merged.csv	8,8 MB
minuteStepsNarrow_merged.csv	46,5 MB
minuteStepsWide_merged.csv	3,5 MB
sleepDay_merged.csv	18 KB
weightLogInfo_merged.csv	7 KB

3.2.3. Loading CSV files

Create a data frame and read in the CSV files from the dataset

```
> daily_activity <- read.csv("dailyActivity_merged.csv")
> sleep_day <- read.csv("sleepDay_merged.csv")
> daily_calories <- read.csv("dailyCalories_merged.csv")
> daily_intensities <- read.csv("dailyIntensities_merged.csv")
> daily_steps <- read.csv("dailySteps_merged.csv")
> heartrate_seconds <- read.csv("heartrate_seconds_merged.csv")
> hourly_calories <- read.csv("hourlyCalories_merged.csv")
> hourly_intensities <- read.csv("hourlyIntensities_merged.csv")
> hourly_steps <- read.csv("hourlySteps_merged.csv")
> minute_calories_narrow <- read.csv("minuteCaloriesNarrow_merged.csv")
> minute_calories_wide <- read.csv("minuteCaloriesWide_merged.csv")
> minute_intensities_narrow <- read.csv("minuteIntensitiesNarrow_merged.csv")
> minute_intensities_wide<- read.csv("minuteIntensitiesWide_merged.csv")
> minute_mets_narrow<- read.csv("minuteMETsNarrow.csv")
> minute_mets_narrow<- read.csv("minuteMETsNarrow_merged.csv")
> minute_sleep<- read.csv("minuteSleep_merged.csv")
> minute_steps_narrow<- read.csv("minuteStepsNarrow_merged.csv")
> minute_steps_wide<- read.csv("minuteStepsWide_merged.csv")
> weight_log_info<- read.csv("weightLogInfo_merged.csv")
```

3.2.4. View the data

Take a look at the data and identify all the columns in the dataset.

- Examine the data using 2 examples: ‘daily_activity’ and ‘sleep_day’,by using the ‘head()’ function.

Take a look at the daily_activity data.

```{r}
head(daily_activity)
```

Description: df[,15] [6 × 15]

	Id <dbl>	ActivityDate <chr>	TotalSteps <int>	TotalDistance <dbl>
1	1503960366	4/12/2016	13162	8.50
2	1503960366	4/13/2016	10735	6.97
3	1503960366	4/14/2016	10460	6.74
4	1503960366	4/15/2016	9762	6.28
5	1503960366	4/16/2016	12669	8.16
6	1503960366	4/17/2016	9705	6.48

Take a look at the sleep_day data.

```{r}
head(sleep_day)
```

Description: df[,5] [6 × 5]

	Id <dbl>	SleepDay <chr>	TotalSleepRecords <int>
1	1503960366	4/12/2016 12:00:00 AM	1
2	1503960366	4/13/2016 12:00:00 AM	2
3	1503960366	4/15/2016 12:00:00 AM	1
4	1503960366	4/16/2016 12:00:00 AM	2
5	1503960366	4/17/2016 12:00:00 AM	1
6	1503960366	4/19/2016 12:00:00 AM	1

- Identify all the columns in the data by using ‘colnames()’ function.

Identify all the column in the daily_activity data.

```{r}
colnames(daily_activity)
```

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

Identify all the column in the sleep_day data.

```{r}
colnames(sleep_day)
```

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

3.2.5. Clean the data

After identifying some data, I realized that I need to rename the headers in each data frame using the 'clean_names()' function to ensure they are syntactically valid. For example, they should start with a letter, followed by an underscore, and then include letters/numbers and so on.

```
> daily_activity <- clean_names(daily_activity)
> daily_steps <- clean_names(daily_steps)
> daily_calories <- clean_names(daily_calories)
> daily_intensities <- clean_names(daily_intensities)
> heartrate_seconds <- clean_names(heartrate_seconds)
> hourly_calories <- clean_names(hourly_calories)
> hourly_intensities <- clean_names(hourly_intensities)
> hourly_steps <- clean_names(hourly_steps)
> minute_sleep <- clean_names(minute_sleep)
> sleep_day <- clean_names(sleep_day)
> weight_log_info <- clean_names(weight_log_info)
```

- Checked for observations with missing or null values using the 'sum!complete.cases()' function.

```
> sum(!complete.cases(daily_activity))
[1] 0
> sum(!complete.cases(daily_steps))
[1] 0
> sum(!complete.cases(daily_calories))
[1] 0
> sum(!complete.cases(daily_intensities))
[1] 0
> sum(!complete.cases(heartrate_seconds))
[1] 0
> sum(!complete.cases(hourly_calories))
[1] 0
> sum(!complete.cases(hourly_intensities))
[1] 0
> sum(!complete.cases(hourly_steps))
[1] 0
> sum(!complete.cases(minute_sleep))
[1] 0
> sum(!complete.cases(sleep_day))
[1] 0
> sum(!complete.cases(weight_log_info))
[1] 65
```

The result of 'sum(!complete.cases(weight_log_info))' is 65, indicating that there are 65 rows with missing values in the 'weight_log_info' data frame.

- Use 'glimpse()' functions to get a quick idea of what's in the dataset:

```
> glimpse(weight_log_info)
Rows: 67
Columns: 8
$ id               <dbl> 1503960366, 15039603…
$ date             <chr> "5/2/2016 11:59:59 P…
$ weight_kg        <dbl> 52.6, 52.6, 133.5, 5…
$ weight_pounds    <dbl> 115.9631, 115.9631, …
$ fat              <int> 22, NA, NA, NA, NA, …
$ bmi              <dbl> 22.65, 22.65, 47.54,…
$ is_manual_report <chr> "True", "True", "Fal…
$ log_id           <dbl> 1.462234e+12, 1.4623…
```

- Create a new dataset without the 'fat' column using 'select ()' functions:

```
> weight_log_info_clean <- select(weight_log_info, -fat)
```

- Recheck the new dataset using '!complete.cases()' function:

```
> sum(!complete.cases(weight_log_info_clean))
[1] 0
```

- Check for duplicate observations using the 'duplicated()' function:

```
> sum(duplicated(daily_activity))
[1] 0
> sum(duplicated(daily_steps))
[1] 0
> sum(duplicated(daily_calories))
[1] 0
> sum(duplicated(daily_intensities))
[1] 0
> sum(duplicated(hourly_calories))
[1] 0
> sum(duplicated(hourly_intensities))
[1] 0
> sum(duplicated(hourly_steps))
[1] 0
> sum(!duplicated(sleep_day))
[1] 410
> sum(duplicated(weight_log_info))
[1] 0
```

- Identified three duplicate observations in 'daily_sleep' and removed them using the 'distinct()' function then review the clean data using the 'duplicated()' function:

```
> sleep_day_clean <- sleep_day %>% distinct()
> sum(duplicated(sleep_day_clean))
[1] 0
```

3.2.6. Review the clean data

I will review the clean data by using the 'glimpse()' functions to quickly understand the contents of the following datasets : daily_activities, heartrate_seconds, sleep_day_clean, and weight_log_info_clean dataset.

```
> glimpse(daily_activity)
Rows: 940
Columns: 15
$ id                         <dbl> 15039…
$ activity_date              <chr> "4/12…
$ total_steps                <int> 13162…
$ total_distance             <dbl> 8.50,…
$ tracker_distance           <dbl> 8.50,…
$ logged_activities_distance <dbl> 0, 0,…
$ very_active_distance       <dbl> 1.88,…
$ moderately_active_distance <dbl> 0.55,…
$ light_active_distance      <dbl> 6.06,…
$ sedentary_active_distance  <dbl> 0, 0,…
$ very_active_minutes        <int> 25, 2…
$ fairly_active_minutes      <int> 13, 1…
$ lightly_active_minutes     <int> 328, …
$ sedentary_minutes          <int> 728, …
$ calories                   <int> 1985,…
```

```
> glimpse(heartrate_seconds)
Rows: 2,483,658
Columns: 3
$ id    <dbl> 2022484408, 2022484408, 2022484408, 2022484408, 2022484408, 2022484408, 202…
$ time  <chr> "4/12/2016 7:21:00 AM", "4/12/2016 7:21:05 AM", "4/12/2016 7:21:10 AM", "4/…
$ value <int> 97, 102, 105, 103, 101, 95, 91, 93, 94, 93, 92, 89, 83, 61, 60, 61, 61, 57,…
```

```
> glimpse(sleep_day_clean)
Rows: 410
Columns: 5
$ id                   <dbl> 1503960366, 150396…
$ sleep_day            <chr> "4/12/2016 12:00:0…
$ total_sleep_records  <int> 1, 2, 1, 2, 1, 1, …
$ total_minutes_asleep <int> 327, 384, 412, 340…
$ total_time_in_bed    <int> 346, 407, 442, 367…
```

```
> glimpse(weight_log_info_clean)
Rows: 67
Columns: 7
$ id               <dbl> 1503960366, 1503960366, 1927972279, 2873212765, 2873212765, 4319703…
$ date             <chr> "5/2/2016 11:59:59 PM", "5/3/2016 11:59:59 PM", "4/13/2016 1:08:52 …
$ weight_kg        <dbl> 52.6, 52.6, 133.5, 56.7, 57.3, 72.4, 72.3, 69.7, 70.3, 69.9, 69.2, …
$ weight_pounds    <dbl> 115.9631, 115.9631, 294.3171, 125.0021, 126.3249, 159.6147, 159.394…
$ bmi              <dbl> 22.65, 22.65, 47.54, 21.45, 21.69, 27.45, 27.38, 27.25, 27.46, 27.3…
$ is_manual_report <chr> "True", "True", "False", "True", "True", "True", "True", "True", "T…
$ log_id           <dbl> 1.462234e+12, 1.462320e+12, 1.460510e+12, 1.461283e+12, 1.463098e+1…
```

4. Analyze

The purpose of this step is to think analytically about the data. We will perform calculations, integrate data from multiple resources, create the table to derive insights. The analysis will focus on application usage, correlation, and users' activity to address the identified problems.

4.1. Application Usage Analysis

Analyze application usage by assessing user engagement through the examination of device feature usage, the number of user logins per date, and the average number of users per day.

4.1.1. Device Feature Usage Analysis

The device feature usage analysis will involve examining the number of unique participants in each data frame using the 'n_distinct()' function.

Step 1: Apply the 'n_distinct()' function in each data frame.

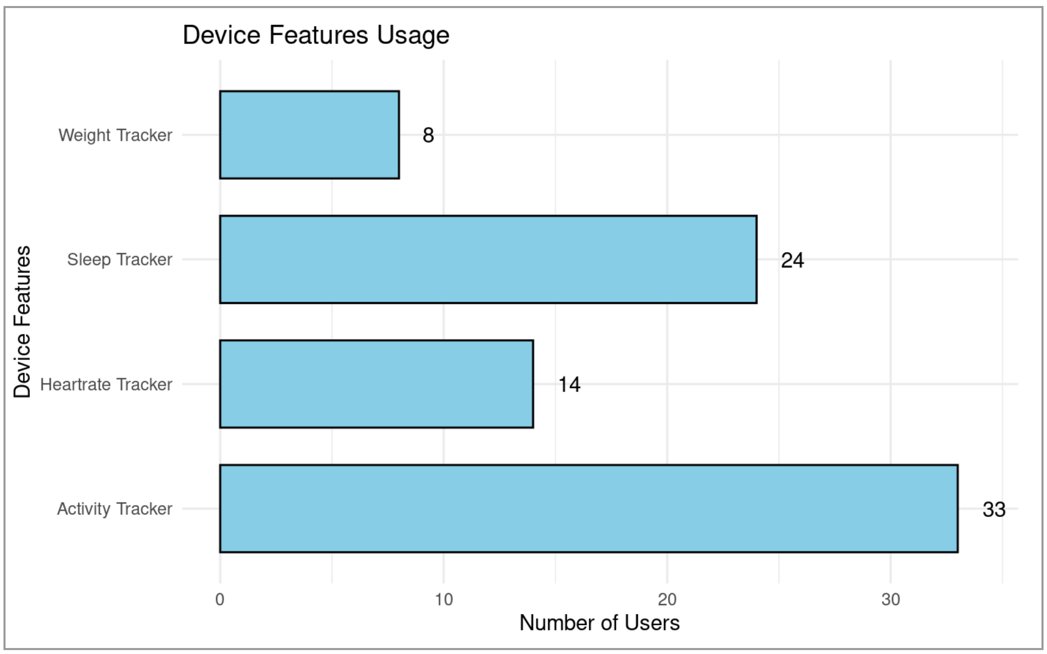
```
> n_distinct(daily_activity$id)
[1] 33
> n_distinct(daily_calories$id)
[1] 33
> n_distinct(daily_intensities$id)
[1] 33
> n_distinct(daily_steps$id)
[1] 33
> n_distinct(heartrate_seconds$id)
[1] 14
> n_distinct(hourly_calories$id)
[1] 33
> n_distinct(hourly_intensities$id)
[1] 33
> n_distinct(hourly_steps$id)
[1] 33
> n_distinct(sleep_day_clean$id)
[1] 24
> n_distinct(weight_log_info_clean$id)
[1] 8
```

Step 2: Create a new data frame to summarize the usage by features.

```
> usage_by_features <- data.frame(
+   device_features = c("Activity Tracker", "Heartrate Tracker", "Sleep Tracker", "Weight Tracker"),
+   no_of_users = c(
+     n_distinct(daily_activity$id),
+     n_distinct(heartrate_seconds$id),
+     n_distinct(sleep_day_clean$id),
+     n_distinct(weight_log_info_clean$id)))
```

Step 3: Create the bar chart for Device Features Usage using the 'ggplot()' and "geom_bar()' function.

```
> ggplot(usage_by_features, aes(x = no_of_users, y = device_features)) +
+   geom_bar(stat = "identity", fill = "skyblue", color = "black", width = 0.7) +
+   geom_text(aes(label = no_of_users), position = position_nudge(x = 1), hjust = -0.1, color = "black") +
+   labs(title = "Device Features Usage",
+        x = "Number of Users",
+        y = "Device Features") +
+   theme_minimal()
```



4.1.2. Number of Users Log per Date Analysis

The number of Users Logs per Date Analysis will compare the trend analysis of how many unique participants log in to track daily activity, sleep data, and weight log information.

Step 1: Create a new data frame for converting date format using the 'mutate()' function.

```r
data_activity <- daily_activity %>%
  mutate(activity_date = as.Date(mdy(activity_date)))
data_sleep <- sleep_day_clean %>%
  mutate(sleep_day = as.Date(mdy_hms(sleep_day)))
data_weight <- weight_log_info_clean %>%
  mutate(date = as.Date(mdy_hms(date)))
```

Step 2: Count the number of user logs per app using the 'group_by()' function to group the data by date and the 'summarise()' function to calculate the total number of users. This step also involves adding a column to indicate the type of activity using the 'mutate()' function.

```r
# Count the number of unique users per activity date
trend_data_activity<- data_activity %>%
  group_by(activity_date) %>%
  summarise(num_users = n_distinct(id)) %>%
  mutate(activity_type = "Activity") # Adding a column to indicate the type of activity


# Count the number of unique users per sleep date
trend_data_sleep <- data_sleep %>%
  group_by(sleep_day) %>%
  summarise(num_users = n_distinct(id)) %>%
  mutate(activity.type = "Sleep")# Adding a column to indicate the type of activity

# Count the number of unique users per weight date
trend_data_weight <- data_weight %>%
  group_by(date) %>%
  summarise(num_users = n_distinct(id)) %>%
  mutate(activity.type = "Weight")# Adding a column to indicate the type of activity
```

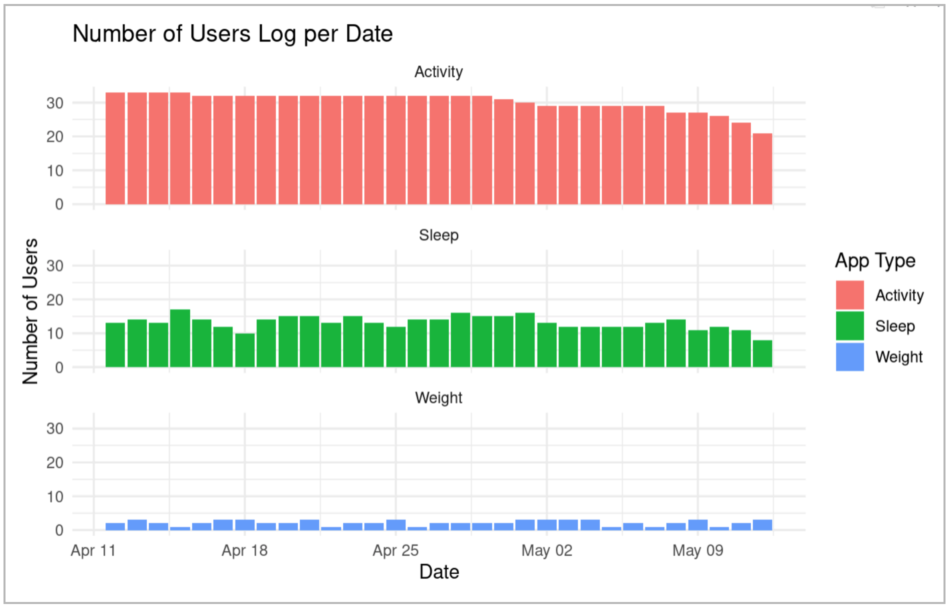
Step 3: Ensure the column names are consistent using the 'colnames()' function & combine the data frame using 'rbind()' function.

```r
# Ensure the column names are consistent
colnames(trend_data_activity) <- c("date", "num_users", "activity_type")
colnames(trend_data_sleep) <- c("date", "num_users", "activity_type")
colnames(trend_data_weight) <- c("date", "num_users", "activity_type")

# Combine the data frame
trend_combined_date <- rbind(trend_data_activity, trend_data_sleep, trend_data_weight)
```

Step 4: Plot the trend analysis using the 'ggplot()' and 'geom_bar()' functions. Utilize the 'facet_wrap()' function for additional insights, considering the number of users per date.

```r
ggplot(trend_combined_date, aes(x = date, y = num_users, fill = activity_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Number of Users Log per Date",
       x = "Date",
       y = "Number of Users",
       fill = "App Type") +
  facet_wrap(~activity_type, scales = "fixed", ncol = 1) +
  theme_minimal()
```



4.1.3. Average Number of Users Utilizing the Activity Tracker by Day Analysis

Building on the findings of the device feature usage analysis (in point 4.1.1), it is noteworthy that the activity tracker is the most utilized among various activities. Hence, it is intriguing to analyze the average number of users utilizing the activity tracker by day, examining this from Sunday to Monday within the time frame from 12 April to 12 May 2016.

Step 1: Create a new data frame from 'daily_activity' to count the number of users on each date using the 'group_by()' and 'summarise()' functions.

```
#Step 1 :  Create a new data frame from 'daily_activity' to count the number of users
on each date using the 'group_by()' &'summarise()' functions,
# Converting the date format using the 'mdy()' function,
# Adding a day column to a new data frame using the 'wday()' function.
# Convert the 'day' numbers in the new data frame to the corresponding day names from
Sunday to Monday using the 'mutate()' and 'recode()' functions.
# Sort the day names in the new data frame from Sunday to Saturday using the
'factor()' function.
usage_by_day <- daily_activity %>% # create a new data frame #
  group_by(activity_date) %>%
  summarise(no_of_users = n_distinct(id)) %>%
  mutate(activity_date = mdy(activity_date),#Converting date format#
         day = factor(wday(activity_date, label = TRUE, abbr = FALSE),
                      levels = c("Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday")))# Adding day col to usage_by_day using wday &
Converting the "day" numbers into the names of the days of the week
```

Step 2: Create a new data frame to summarize the average number of users per day using the 'group_by()' and 'summarise()' functions.

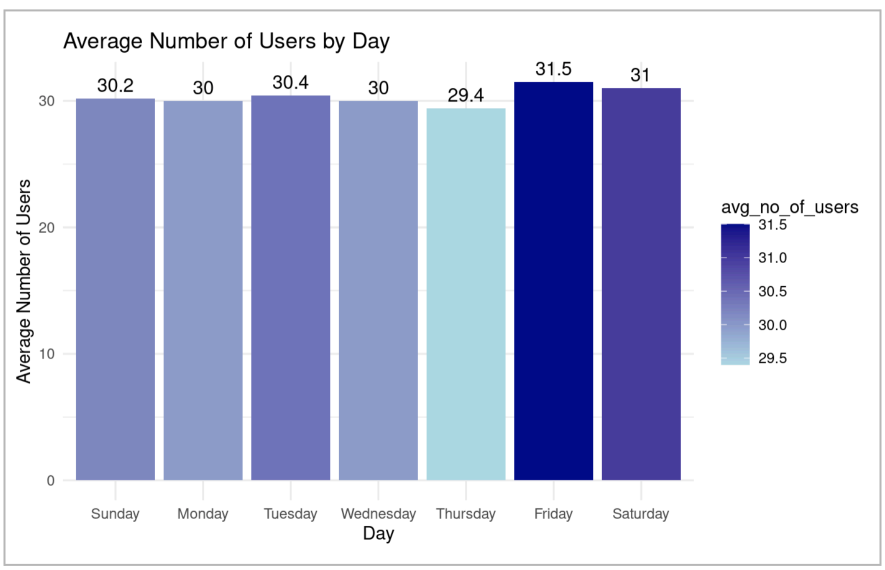
```
usage_by_day_summary <- usage_by_day %>%
  group_by(day) %>%
  summarise(avg_no_of_users = round(mean(no_of_users),1))
```

Step 3: View the summarized data.

	day	avg_no_of_users
1	Sunday	30.2
2	Monday	30.0
3	Tuesday	30.4
4	Wednesday	30.0
5	Thursday	29.4
6	Friday	31.5
7	Saturday	31.0

Step 4: Create the bar chart for Average Number of Users by Day using the 'ggplot()' and 'geom_bar()' functions.

```
ggplot(usage_by_day_summary, aes(x = day, y = avg_no_of_users, fill = avg_no_of_users)) +
  geom_bar(stat = "identity") +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  geom_text(aes(label = round(avg_no_of_users, 2)),
            position = position_dodge(width = 0.9),  # Adjust width as needed
            size = 4,
            vjust = -0.5) +
  labs(title = "Average Number of Users by Day",
       x = "Day",
       y = "Average Number of Users") +
  theme_minimal()
```



4.2. Correlation Analysis

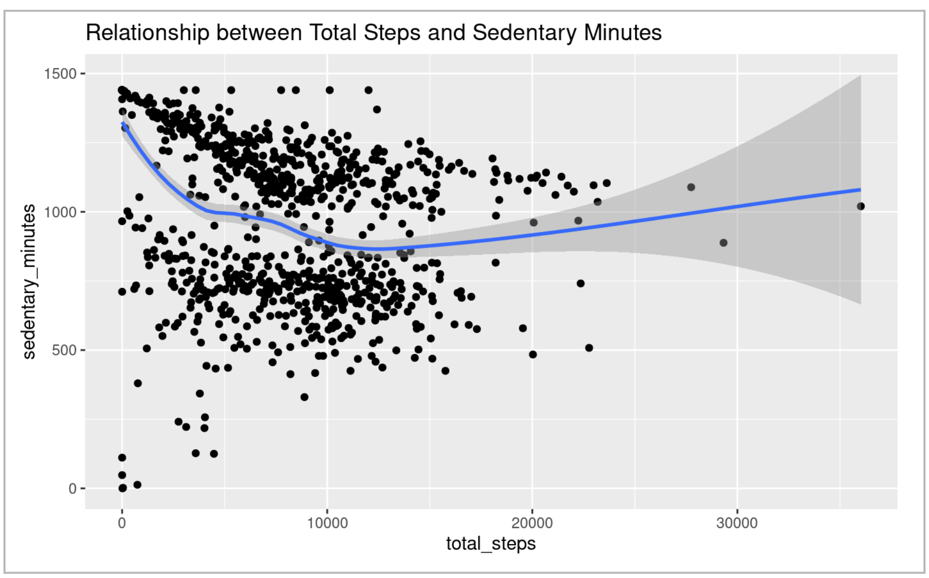
Correlation analysis is a statistical method used to assess the strength and direction of a linear relationship between two variables. The result of a correlation analysis is a correlation coefficient, which ranges from -1 to 1. The sign (positive or negative) indicates the direction of the relationship, the magnitude (closer to 1) reflects the strength of the relationship.

This analysis involves examining the relationship between total steps and sedentary minutes, total minutes asleep and total time in bed, total steps and calories, and total active minutes and calories.

4.2.1. Relationship between total steps and sedentary minutes

This analysis examines the correlation between the total number of steps taken and the total duration of sedentary (inactive) minutes using the 'geom_point()' and 'geom_smooth()' functions.

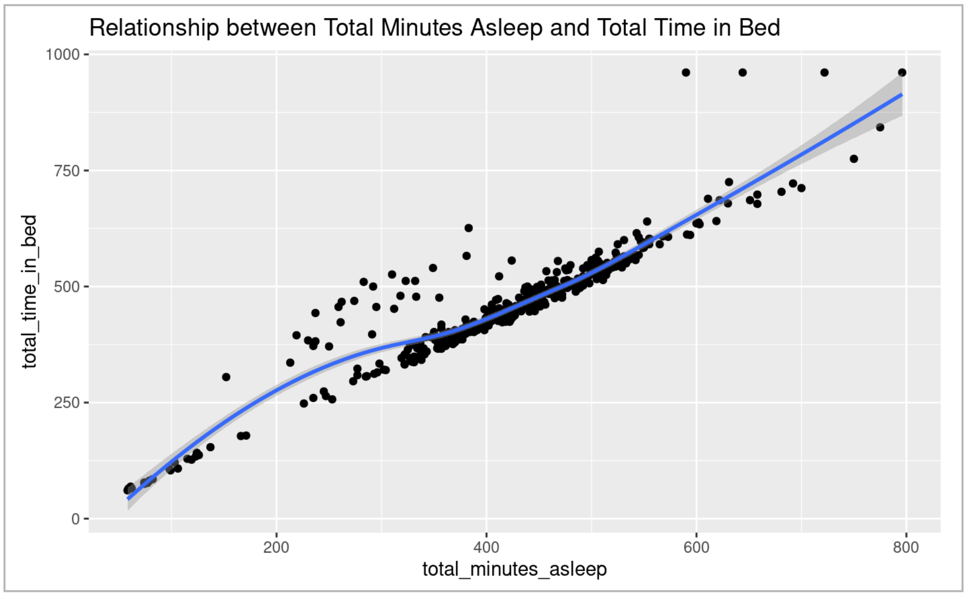
```
ggplot(data=daily_activity, aes(x=total_steps, y=sedentary_minutes)) +
  geom_point() +
  geom_smooth() +
  labs(title = "Relationship between Total Steps and Sedentary Minutes")
```



4.2.2. Relationship between total minutes asleep and total time in bed

This analysis examines the correlation between the total minutes participants spent asleep and the total duration of time they spent in bed using the ‘geom_point()’ and ‘geom_smooth()’ functions.

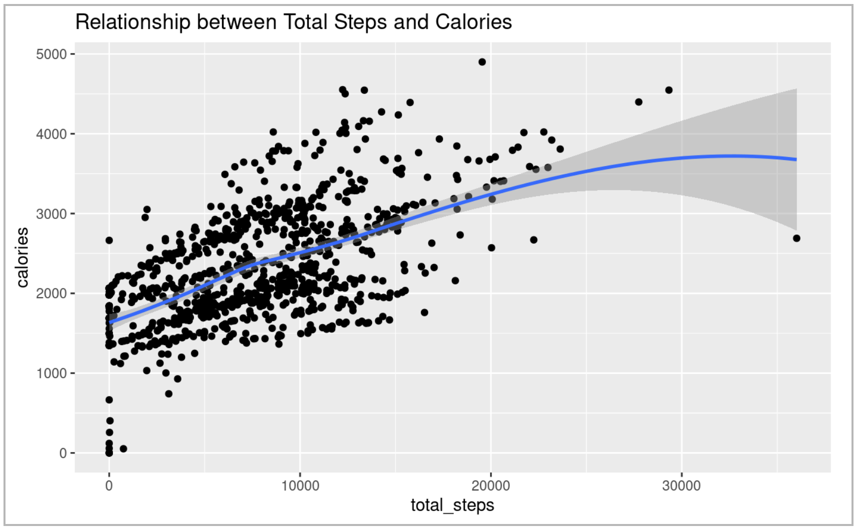
```
ggplot(data=sleep_day_clean, aes(x=total_minutes_asleep, y=total_time_in_bed)) +
  geom_point() +
  geom_smooth() +
  labs(title = "Relationship between Total Minutes Asleep and Total Time in Bed")
```



4.2.3. Relationship between total steps and calories

This analysis examines the correlation between the total number of steps taken and the total calories burned using the ‘geom_point()’ and ‘geom_smooth()’ functions.

```
ggplot(data=daily_activity, aes(x=total_steps, y=calories)) +
  geom_point() +
  geom_smooth() +
  labs(title = "Relationship between Total Steps and Calories")
```

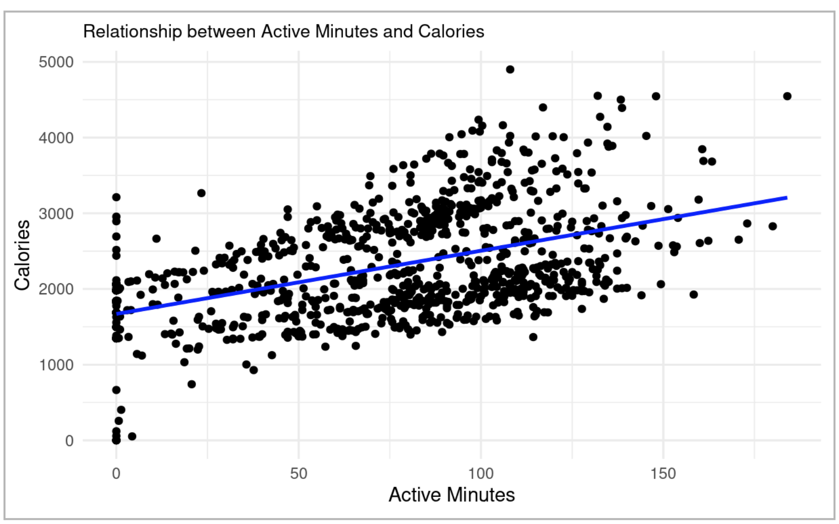


4.2.4. Relationship between active minutes and calories

This analysis examines the correlation between the total active minutes (from light to very active minutes) and the total calories burned using the 'geom_point()' and 'geom_smooth()' functions.

```
data_activity$active_minutes <- rowMeans(select(data_activity,
c("very_active_minutes", "fairly_active_minutes", "lightly_active_minutes")))

ggplot(data_activity, aes(x = active_minutes, y = calories)) +
  geom_point() +  # Scatter plot
  geom_smooth(method = "lm", se = FALSE, color = "blue") +  # Linear regression line
  labs(x = "Active Minutes", y = "Calories", title = "Relationship between Active
Minutes and Calories") +
  theme_minimal() + theme(
    plot.title = element_text(size = 10)  # Adjust the title text size here
  )
```



4.3. Users' Activity Analysis

Users' activity analysis refers to the examination of data related to participants' physical activities, weight, and sleep patterns. This analysis involves active minute analysis, summary statistics, average activity level by day, average total steps by day, and average hourly sleep from Sunday to Saturday.

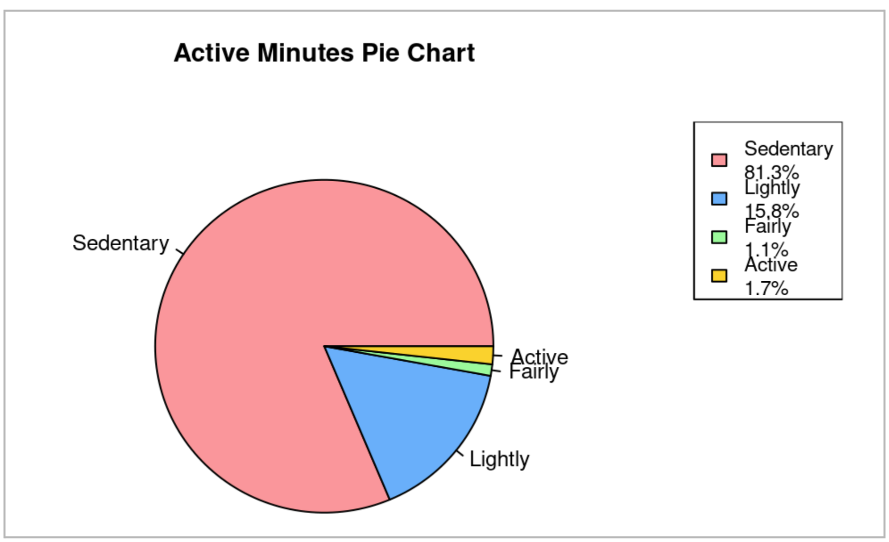
4.3.1. Active Minutes Analysis

The active minute analysis represents the duration of time spent on activities ranging from sedentary to very active, presented in a pie chart using the 'pie()' function. This analysis helps assess how much time a user spends on activities that contribute to their overall exercise and well-being.

```
sedentary <- sum (daily_activity$sedentary_minutes)
active <- sum (daily_activity$very_active_minutes)
fairly <- sum (daily_activity$fairly_active_minutes)
lightly <- sum (daily_activity$lightly_active_minutes)
x <- c(sedentary,lightly,fairly,active)
labels <- c("Sedentary", "Lightly", "Fairly", "Active")
piepercent <- round(100*x/sum(x),1)
label_percent <- paste0(labels, "\n", piepercent, "%")
custom_colors <- c("#FF9999", "#66B2FF", "#99FF99", "#FFD700")


# Plotting the pie chart
pie(x, labels = labels, main = "Active Minutes Pie Chart", col = custom_colors, , cex
= 0.75, cex.main = 0.9, cex.lab = 0.8, cex.axis = 0.8)

# Adding legend
legend("topright", legend = label_percent, cex = 1.0, fill = custom_colors )
```



4.3.2. Summary statistics for Daily Activities, Sleep, and Weight

The summary statistics analysis involves checking a quick summary statistic related to daily activities (such as total steps, total distance, activity minutes, calories burned), sleep (total sleep records, total minutes asleep, total time in bed), and weight log information (weight, bmi) using the 'select()' and 'summary()' functions. This analysis will help us provide an overview of the users' daily activities, sleep, and weight, making it easier to identify trends or irregularities.

For the daily activity dataframe:

````
```{r}
daily_activity %>%
  select(total_steps,
         total_distance,
         very_active_minutes,
         fairly_active_minutes,
         lightly_active_minutes,
         sedentary_minutes,
         calories) %>%
  summary()
```
````

```
  total_steps    total_distance   very_active_minutes fairly_active_minutes
 Min.   :    0   Min.   : 0.000   Min.   :  0.00      Min.   :  0.00
 1st Qu.: 3790   1st Qu.: 2.620   1st Qu.:  0.00      1st Qu.:  0.00
 Median : 7406   Median : 5.245   Median :  4.00      Median :  6.00
 Mean   : 7638   Mean   : 5.490   Mean   : 21.16      Mean   : 13.56
 3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.: 32.00      3rd Qu.: 19.00
 Max.   :36019   Max.   :28.030   Max.   :210.00      Max.   :143.00
 lightly_active_minutes sedentary_minutes    calories
 Min.   :  0.0          Min.   :   0.0    Min.   :   0
 1st Qu.:127.0          1st Qu.: 729.8    1st Qu.:1828
 Median :199.0          Median :1057.5    Median :2134
 Mean   :192.8          Mean   : 991.2    Mean   :2304
 3rd Qu.:264.0          3rd Qu.:1229.5    3rd Qu.:2793
 Max.   :518.0          Max.   :1440.0    Max.   :4900
```

```
For the sleep dataframe:
```{r}
sleep_day_clean %>%
  select(total_minutes_asleep,
  total_time_in_bed) %>%
  summary()
```

 total_minutes_asleep total_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
```

```
For the weight dataframe:
```{r}
weight_log_info_clean %>%
  select(weight_kg, weight_pounds, bmi) %>%
  summary()
```

   weight_kg   weight_pounds   bmi
 Min.   : 52.60   Min.   :116.0   Min.   :21.45
 1st Qu.: 61.40   1st Qu.:135.4   1st Qu.:23.96
 Median : 62.50   Median :137.8   Median :24.39
 Mean   : 72.04   Mean   :158.8   Mean   :25.19
 3rd Qu.: 85.05   3rd Qu.:187.5   3rd Qu.:25.56
 Max.   :133.50   Max.   :294.3   Max.   :47.54
```

4.3.3. Average Activity Level By Day

This analysis calculates the average intensity of physical activity on different days of the week. Identifying patterns in a users' activity level throughout the week can be useful for understanding lifestyle habits and shaping exercises routines.

Step 1: Create a new dataframe to summarize the minutes of activity by date using the 'group_by()' and 'summarise()' functions, convert the date format, and sort the day.

```r
# CREATE A NEW DATAFRAME :
# Summarize the minutes of activity by date using the 'group_by()' and 'summarise()'
functions.
# Convert the date format using the 'mdy()' function.
# Add a 'day' column using the 'wday()' function.
# Convert the 'day' numbers into the name of the days of the week using the
'mutate()' and 'recode()' functions.
# Sort the day from Monday to Sunday using the 'factor()' function.
activity_level <- daily_activity %>%
  group_by(activity_date) %>%
  summarise(active = n_distinct(very_active_minutes),
            fairly = n_distinct(fairly_active_minutes),
            lightly = n_distinct(lightly_active_minutes),
            sedentary = n_distinct(sedentary_minutes)) %>%
  mutate(activity_date = mdy(activity_date),
         day = factor(wday(activity_date, label = TRUE, abbr = FALSE),
                      levels = c("Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday")))
```

Step 2: Summarize the activity level by day using the 'group_by()' and 'summarise()' functions.

```r
activity_level_summary <- activity_level %>%
  group_by(day) %>%
  summarise(active = round(mean(active),1),
            fairly = round(mean(fairly),1),
            lightly = round(mean(lightly),1))
```

Step 3: View the summary.

	day	active	fairly	lightly	sedentary
1	Sunday	14.5	14.0	27.8	27.5
2	Monday	17.8	16.5	26.8	28.0
3	Tuesday	18.0	16.4	27.2	27.8
4	Wednesday	16.8	16.4	27.4	28.0
5	Thursday	15.6	15.4	27.0	27.8
6	Friday	16.0	15.0	29.2	30.2
7	Saturday	16.8	16.2	28.0	29.5

Step 4: Reshape the dataframe to long format.

```r
activity_level_long <- activity_level %>%
  pivot_longer(cols = c(active, fairly, lightly, sedentary),
               names_to = "activity",
               values_to = "minutes")
```

Step 5: Reshape the data for easier plotting using the 'ggplot()' and 'geom_bar()' functions. Utilize the 'facet_wrap()' function for additional insights, considering the activity.

```
ggplot(activity_level_long, aes(x = day, y = minutes, fill = activity)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Activity Level by Day",
       x = "Day",
       y = "Average Minutes",
       fill = "Activity") +
  scale_fill_manual(values = c("active" = "#000000", "fairly" = "#666666", "lightly" = "#999999",
sedentary = "#cccccc")) +
  facet_wrap(~ activity, scales = "fixed") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```

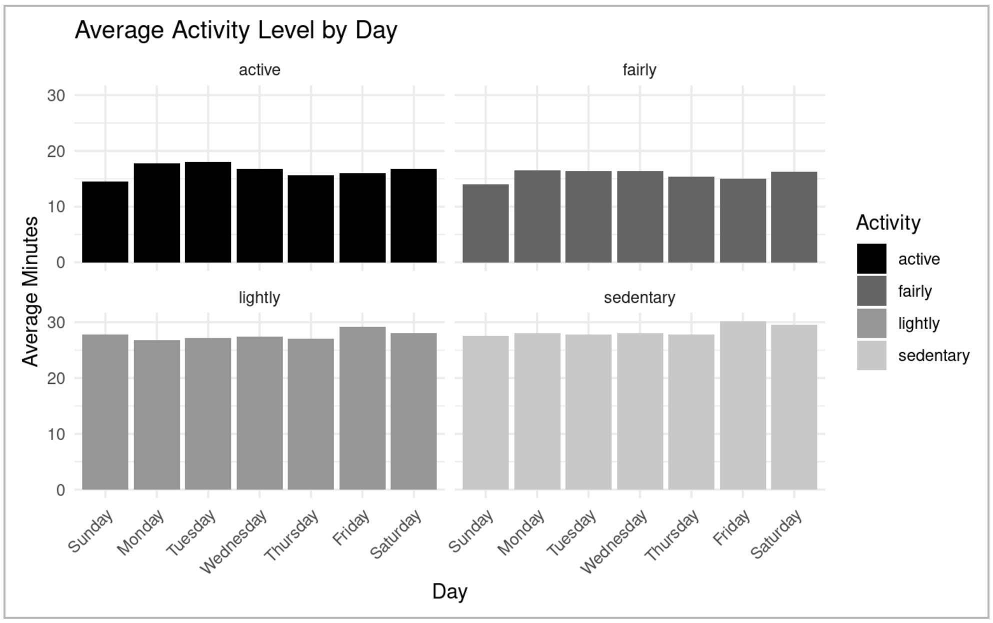


Table Summary of the Day with the Highest Average Minutes

Activity Level	Day with the Highest Average Minutes
Active	Tuesday
Fairly	Monday
Lightly	Friday
Sedentary	Friday

4.3.4. Average Total Steps By Day

The average total steps by day analysis calculates the average number of steps taken by a user on each day. This analysis aids in understanding the users' overall activity level and identifies variations in step count.

Step 1: Use the 'mutate()' function to convert the date format, add the day of the week using the 'wday()' function, and then summarize the total steps average by day using the 'group_by()' and 'summarise()' functions.

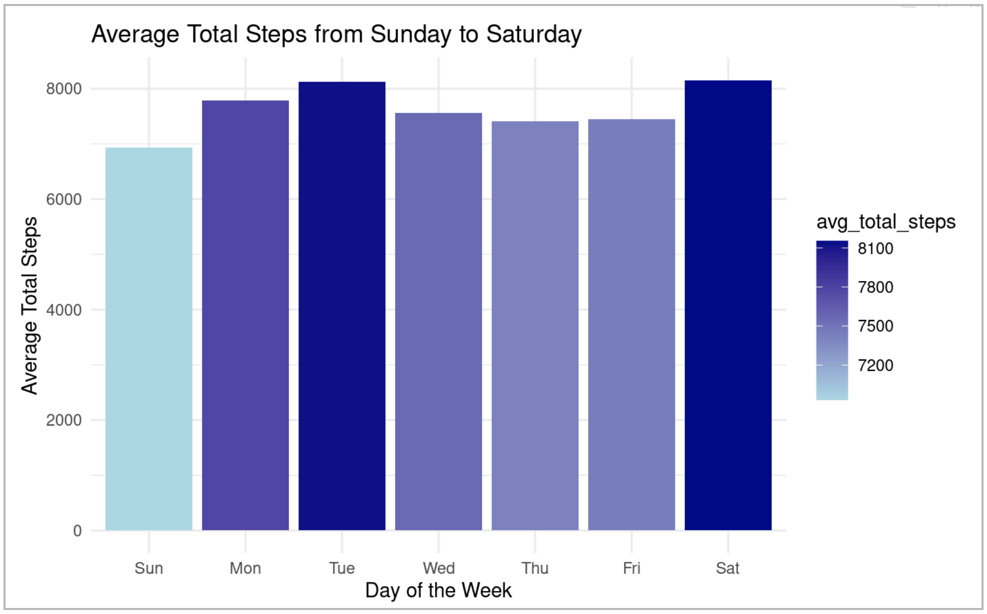
```
daily_steps_avg <- daily_steps %>%
  mutate(activity_day = mdy(activity_day),  # Convert to date-time
         day_of_week = wday(activity_day, label = TRUE)) %>%  # Extract day of the week
  group_by(day_of_week) %>%
  summarise(avg_total_steps = round(mean(step_total, na.rm = TRUE), 0))
```

Step 2: View the data.

	day_of_week	avg_total_steps
1	Sun	6933
2	Mon	7781
3	Tue	8125
4	Wed	7559
5	Thu	7406
6	Fri	7448
7	Sat	8153

Step 3: Create the bar chart using the 'ggplot()' and 'geom_col()' functions.

```
ggplot(daily_steps_avg, aes(x = day_of_week, y = avg_total_steps, fill = avg_total_steps)) +
  geom_col() +
  scale_fill_gradient(low = "lightblue", high ="darkblue") +
  labs(title = "Average Total Steps from Sunday to Saturday",
       x = "Day of the Week",
       y = "Average Total Steps") +
  theme_minimal()
```



4.3.5. Average Hourly Sleep from Sunday to Saturday

This analysis breaks down the average duration of sleep by hour for each day of the week to identify sleep patterns and variation in users' sleep duration throughout the week.

Step 1: Use the 'mutate()' function to convert the date format, add the day of the week using the 'wday()' function, and then summarize the average hourly sleep by day using the 'group_by()' and 'summarise()' functions.

```
sleep_day_trend <- sleep_day_clean %>%
  mutate(sleep_day = mdy_hms(sleep_day),  # Convert to date-time
         day_of_week = wday(sleep_day, label = TRUE),  # Extract day of the week
         hourly_sleep = round(total_minutes_asleep / 60,1)) %>%   # Calculate average hourly sleep
           group_by(day_of_week) %>%
  summarize(avg_hourly_sleep = round(mean(hourly_sleep, na.rm = TRUE),2))
```

Step 2: Reorder days of the week using the 'factor()' function.

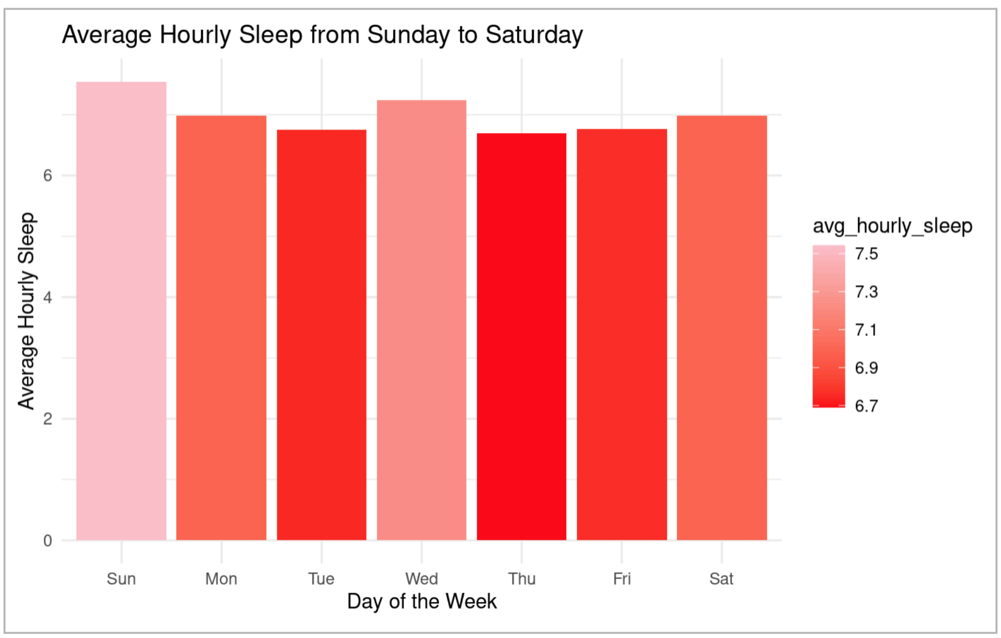
```
sleep_day_trend$day_of_week <- factor(sleep_day_trend$day_of_week, levels = c("Sun", "Mon", "Tue",
"Wed", "Thu", "Fri", "Sat"))
```

Step 3: View the Summary.

	day_of_week	avg_hourly_sleep
1	Sun	7.54
2	Mon	6.99
3	Tue	6.75
4	Wed	7.24
5	Thu	6.69
6	Fri	6.76
7	Sat	6.99

Step 4: Create the bar chart using the ‘ggplot()’ and ‘geom_col()’ function.

```
ggplot(sleep_day_trend, aes(x = day_of_week, y = avg_hourly_sleep, fill = avg_hourly_sleep)) +
  geom_col() +
  scale_fill_gradient(low = "red", high = "pink") +
  labs(title = "Average Hourly Sleep from Sunday to Saturday",
       x = "Day of the Week",
       y = "Average Hourly Sleep") +
  theme_minimal()
```



5. Share

The purpose of this step is to communicate findings and insights to stakeholders. Visualizations and summaries will be utilized to effectively communicate the result of the analysis. Here are some findings and insights derived from the application usage analysis, correlation analysis, and users' activity analysis.

5.1. Application Usage Analysis

The analysis was conducted by analyzing user engagement through the examination of device feature usage, the number of user logins per date, and the average number of users per day, and aims to gain insights into how users interact with the application. Examining these aspects can identify popular features, peak usage times, and user engagement levels.

5.1.1. Device Feature Usage Analysis Findings:

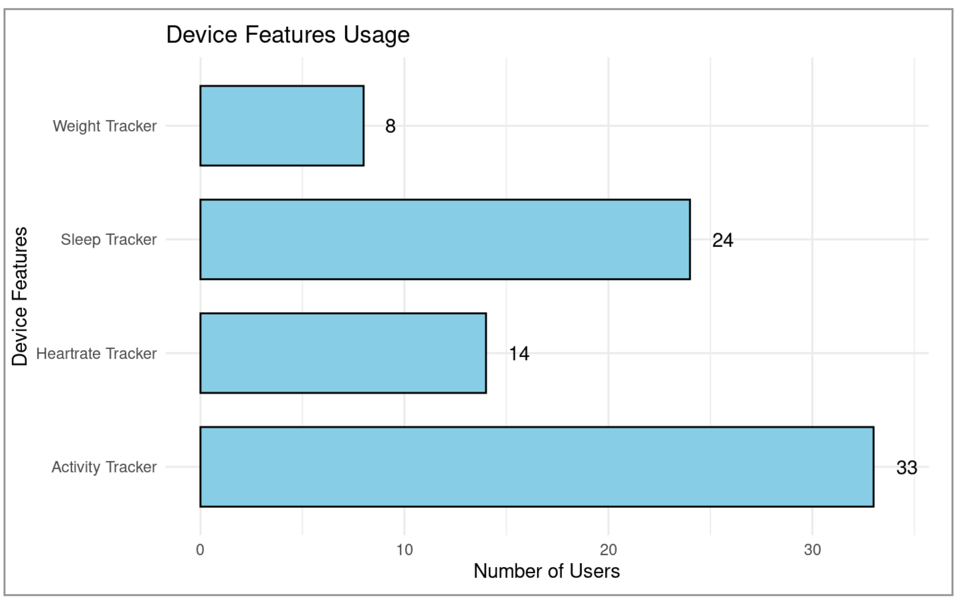


Figure 5.1.1. Device Features Usage

Based on the 'Device Features Usage' chart in figure 5.1.1, the Activity Tracker feature of smart device (which monitors activity levels, steps, and calories) appears to be more frequently used than the Sleep Tracker (which monitors asleep and awake time in bed), Heart Rate Tracker, or Weight Tracker features.

This finding suggests that users might be more interested in tracking their activity patterns and progress toward fitness goals than in monitoring their sleep, heart rate, or weight patterns.

5.1.2. Number of Users Log per Date Analysis Findings

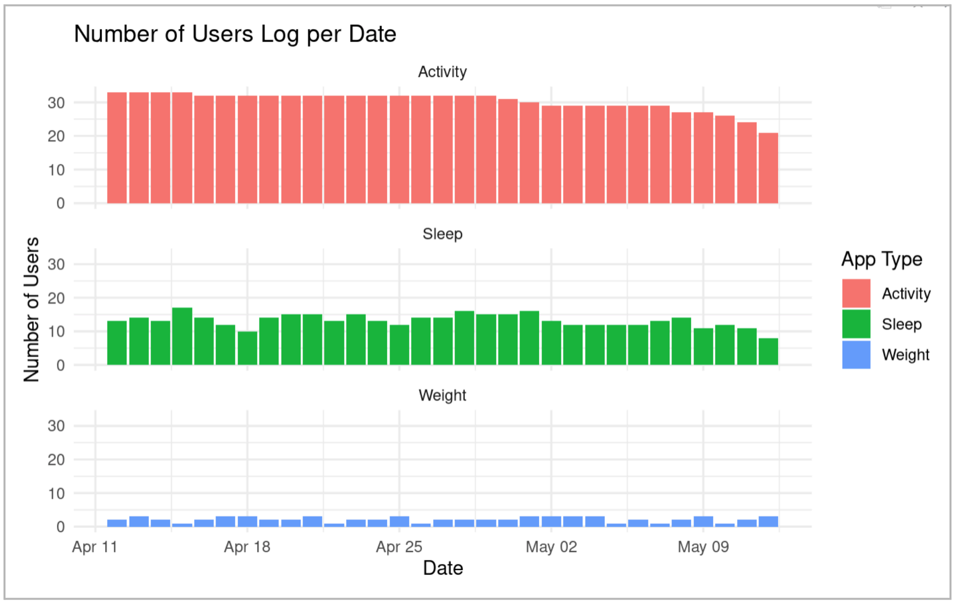


Figure 5.1.2. Number of Users Log per Date

As depicted in Figure 5.1.2, ‘Number of Users Log per Date’, it is observed that, overall, users are consistently and frequently tracking their activity data. Sleep monitoring, while popular, shows less consistently. Notably, weight tracking appears almost insignificant in comparison to other types of activity monitoring. These findings suggest a higher user interest in physical activity tracking.

5.1.3. Usage rate by day of the week for Activity Tracker's Findings

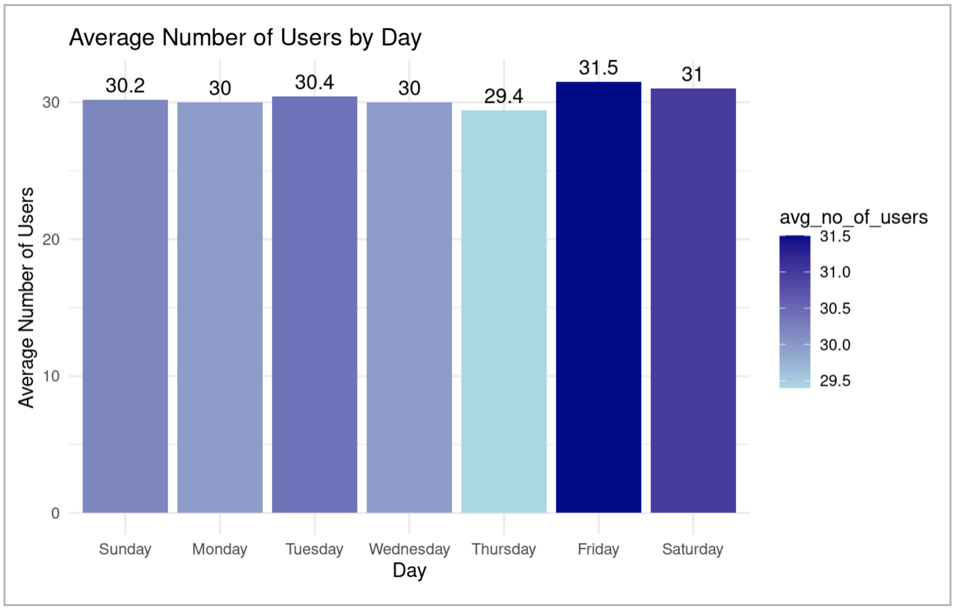


Figure 5.1.3. Average Number of Users by Day

Based on the 'Average Number of Users by Day' chart in figure 5.1.3, the smart device's Activity Tracker appears to be consistently used every day of the week. However, there is a slight variation in the usage pattern, with the device being least used on Thursdays and most used on Fridays.

This trend suggests that users might be less motivated to use the device on Thursdays, possibly because they have already been actively tracking their health for most of the week.

Conversely, users may exhibit increased motivation to use the device on Fridays. This pattern could be attributed to the fact that Fridays are often considered the last day of the workweek, and users may be interested in reviewing their progress throughout the week.

5.2. Correlation Analysis

Correlation analysis involves examining the relationship between total steps and sedentary minutes, total minutes asleep and total time in bed, total steps and calories, as well as total active minutes and calories. The sign (positive or negative) indicates the direction of the relationship, the magnitude (closer to 1) reflects the strength of the relationship.

5.2.1. Relationship between Total Steps and Sedentary Minutes Findings

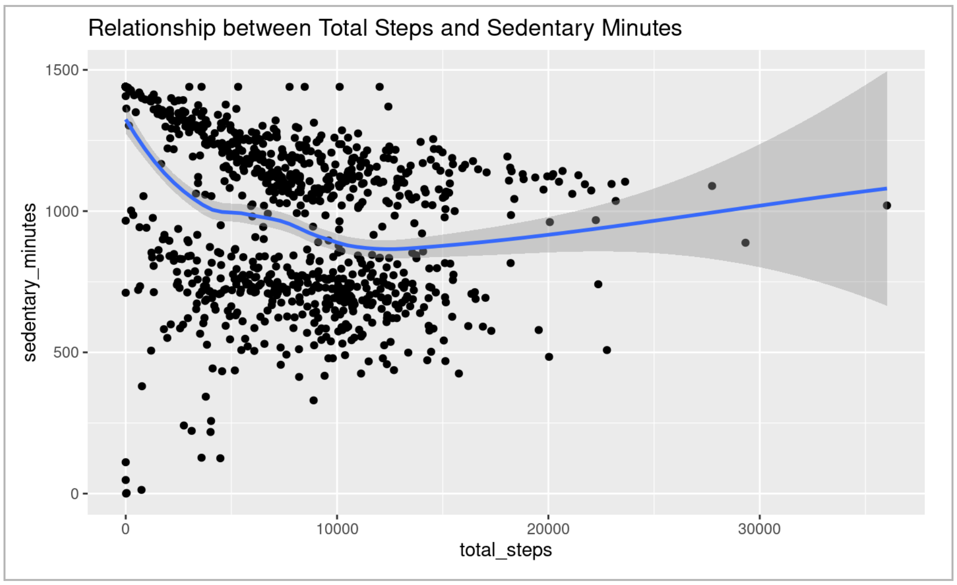


Figure 5.2.1. Relationship between Total Steps and Sedentary Minutes

Figure 5.2.1 illustrates a negative relationship between total steps and sedentary minutes for 0-10.000 steps. Sedentary minutes decrease with increasing steps, indicating that the more daily steps a user takes, the less time they spend inactive. Beyond 10,000 steps, however, the chart indicates a positive relationship. This change could be attributed to participants getting tired and taking a rest after an extended period of walking.

5.2.2. Relationship between Total Minutes Asleep and Total Time in Bed Findings

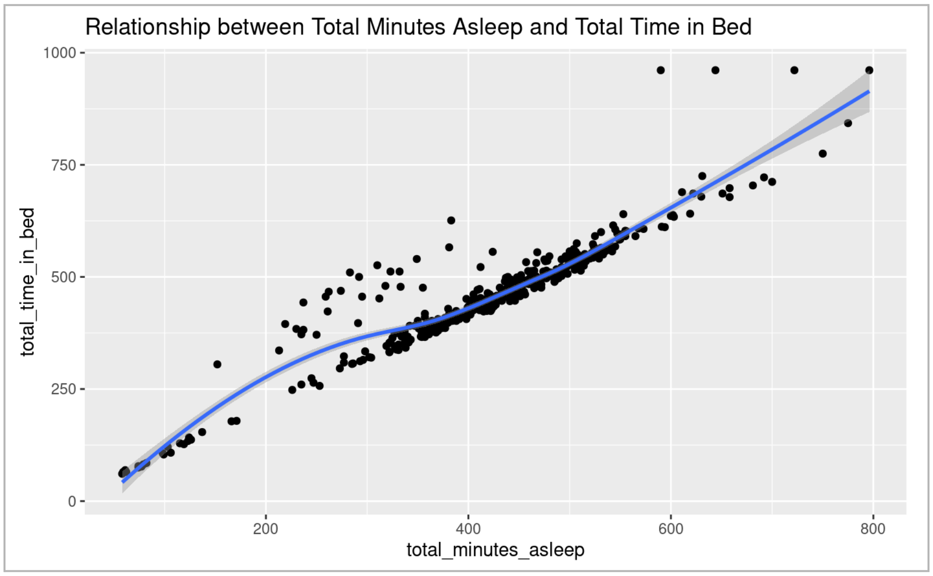


Figure 5.2.2. Relationship between Total Minutes Asleep and Total Time in Bed

Figure 5.2.2 shows a positive correlation between time spent asleep and time spent in bed. This can be used to encourage users to get into bed earlier in order to get the recommended amount of sleep.

5.2.3. Relationship between Total Steps and Calories Burns Findings

The relationship between total steps and calories burned is depicted in Figure 5.2.3, revealing a positive correlation. This finding suggests that encouraging users to set a daily step target can help them achieve their goal of burning calories.

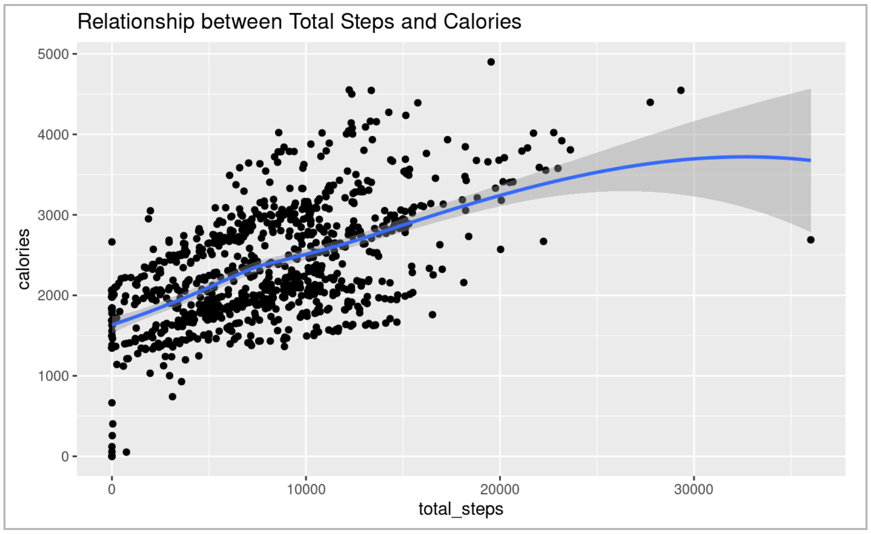


Figure 5.2.3. Relationship between Total Steps and Calories

5.2.4. Relationship between Active Minutes and Calories Findings

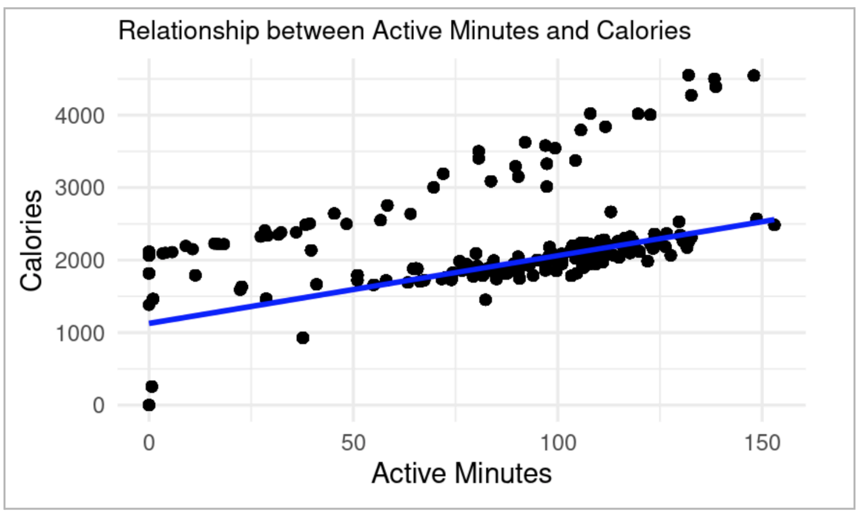


Figure 5.2.4. Relationship between Active Minutes and Calories

The relationship between active minutes and calories is illustrated in Figure 5.2.4 indicating a positive correlation. This implies that as the number of active minutes increases, there is a corresponding increase in the calories burned.

5.3. Users' Activity Analysis

The Users' Activity Analysis, achieved through active minutes analysis, summary statistics, average activity level by day, average total steps by day, and average hourly sleep from Sunday to Saturday, aims to gain comprehensive insights into participants' physical activities, weight, and sleep patterns. The purpose is to provide a detailed understanding of various aspects related to user activity.

5.3.1. Active Minutes Analysis Findings

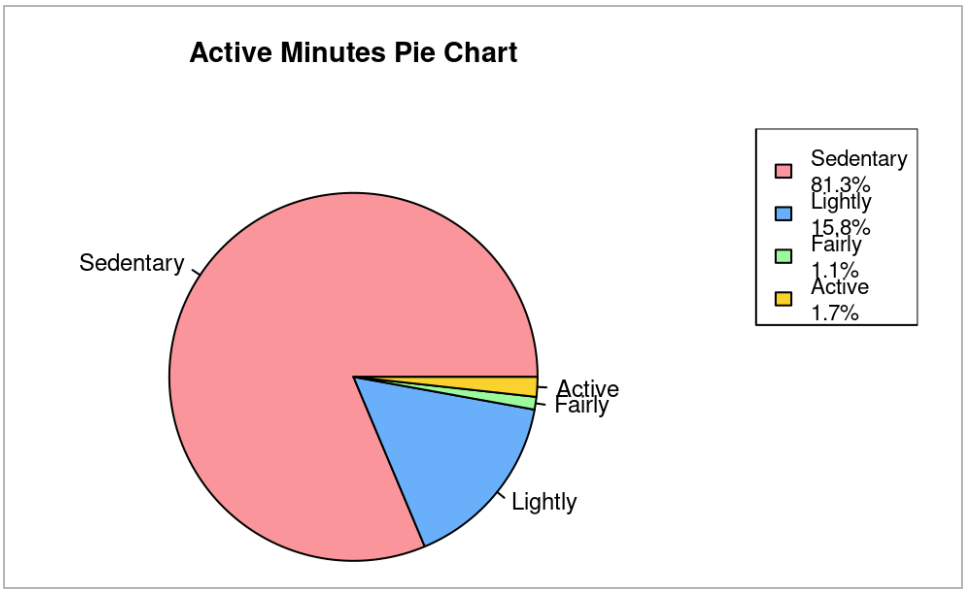


Figure 5.3.1. Active Minutes Pie Chart

As seen in Figure 5.3.1, the 'Active Minutes Pie Chart', it appears that the participants spend 81.3% of their time in sedentary minutes. This indicates that people tend to be inactive for long periods. The percentage of active minutes consists of very active minutes (1.7%) and fairly active minutes (1.1%). These active minutes are significantly lower compared to other activity categories.

5.3.2. Summary statistics for Daily Activities, Sleep, and Weight Findings

```
For the daily activity dataframe:
```{r}
daily_activity %>%
  select(total_steps,
         total_distance,
         very_active_minutes,
         fairly_active_minutes,
         lightly_active_minutes,
         sedentary_minutes,
         calories) %>%
  summary()
```

  total_steps   total_distance   very_active_minutes fairly_active_minutes
 Min.   :   0   Min.   : 0.000   Min.   : 0.00   Min.   : 0.00
 1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 0.00   1st Qu.: 0.00
 Median : 7406   Median : 5.245   Median : 4.00   Median : 6.00
 Mean   : 7638   Mean   : 5.490   Mean   : 21.16   Mean   : 13.56
 3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.: 32.00   3rd Qu.: 19.00
 Max.   :36019   Max.   :28.030   Max.   :210.00   Max.   :143.00
 lightly_active_minutes sedentary_minutes   calories
 Min.   : 0.0   Min.   :   0.0   Min.   :   0
 1st Qu.:127.0   1st Qu.: 729.8   1st Qu.:1828
 Median :199.0   Median :1057.5   Median :2134
 Mean   :192.8   Mean   : 991.2   Mean   :2304
 3rd Qu.:264.0   3rd Qu.:1229.5   3rd Qu.:2793
 Max.   :518.0   Max.   :1440.0   Max.   :4900
```

```
For the sleep dataframe:
```{r}
sleep_day_clean %>%
  select(total_minutes_asleep,
  total_time_in_bed) %>%
  summary()
```

 total_minutes_asleep total_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
```

```
For the weight dataframe:
```{r}
weight_log_info_clean %>%
  select(weight_kg, weight_pounds, bmi) %>%
  summary()
```

   weight_kg   weight_pounds   bmi
 Min.   : 52.60   Min.   :116.0   Min.   :21.45
 1st Qu.: 61.40   1st Qu.:135.4   1st Qu.:23.96
 Median : 62.50   Median :137.8   Median :24.39
 Mean   : 72.04   Mean   :158.8   Mean   :25.19
 3rd Qu.: 85.05   3rd Qu.:187.5   3rd Qu.:25.56
 Max.   :133.50   Max.   :294.3   Max.   :47.54
```

Figure 5.3.2. Summary Statistics for Daily Activities, Sleep, and Weight

Insight from Figure 5.3.2 reveal compelling data regarding participants' daily activities, sleep patterns, and weight:

- Participants take an average of 7,638 steps per day, covering a distance of 5,490 miles. While this is below the common recommendation of 10,000 steps per day, it serves as a baseline for potential activity improvements.
- Daily activity includes a mean very active time of 21.16 minutes, a fairly active time of 13.56 minutes, and a lightly active time of 192.8 minutes. Sedentary behavior accounts for a mean 991.2 minutes, emphasizing the opportunity for increased physical activity.
- The average calorie burn is 2,304 calories per day, providing insights into the participants' energy expenditure.
- Sleep data indicates an average total sleep duration of 419.2 minutes, equivalent to around 7 hours of sleep per day. Participants spend an average of 458.5 minutes in bed, suggesting it takes around 39 minutes for them to fall asleep. Although the recommended sleep duration for most adults is 7-9 hours per night, individual variations may apply.
- The average weight is 72.04 kg or 158.8 pounds, with a mean BMI of 25.19. According to the World Health Organization, the classification for the average BMI places participants in the overweight range (BMI 25 to 29.9).

This summary prompts further analysis, including an examination of average activity level by day, average total steps from Sunday to Saturday, and hourly sleep average from Sunday to Saturday.

5.3.3. Average Activity Level By Day Findings

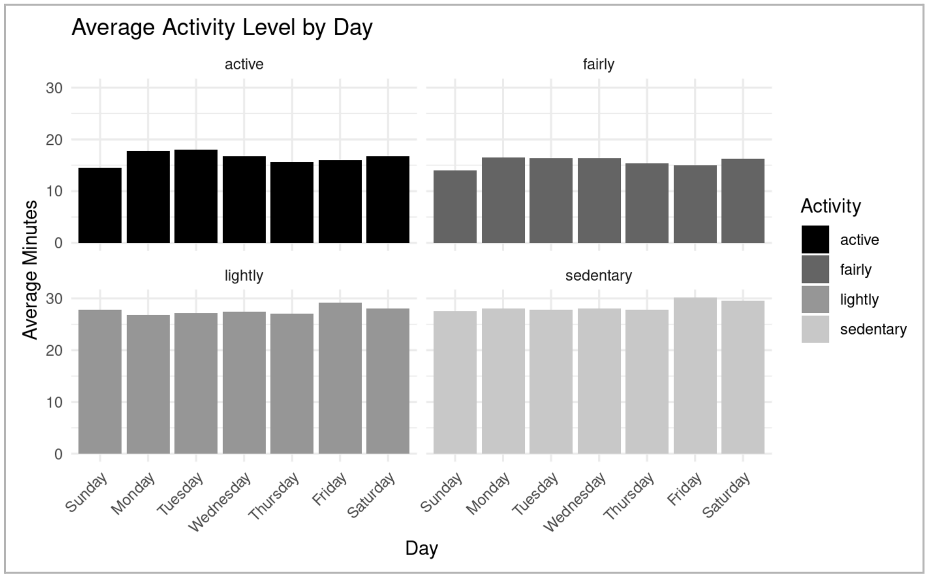


Figure 5.3.3. Average Activity Level By Day

Table Summary of the Day with the Highest Average Minutes

Activity Level	Day with the Highest Average Minutes
Active	Tuesday
Fairly	Monday
Lightly	Friday
Sedentary	Friday

As Illustrated in Figure 5.3.3 and the 'Table Summary of the Day with the Highest Average MInutes', it is evident that the highest average minutes of active activity are consistently recorded on Tuesday, with fairly active activity peaking on Monday. In contrast, both lightly and sedentary activity exhibit their highest averages on Friday. We can speculate that the participants may experience fatigue after a long workday on Fridays, contributing to the observed increase in sedentary minutes during this day.

5.3.4. Average Total Steps from Sunday to Saturday Findings

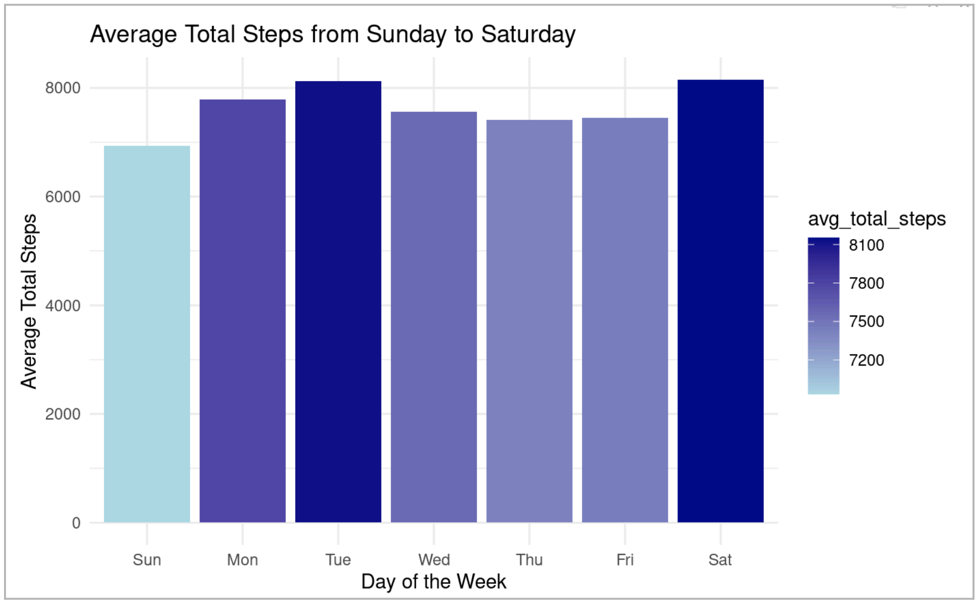


Figure 5.3.4. Average Total Steps from Sunday to Saturday

Figure 5.3.4. Average Total Steps from Sunday to Saturday reveal distinct patterns in average total steps throughout the week:

- Tuesday and Saturday: These days, consistently record higher average total steps, indicating increased physical activity. This suggests that participants are more active during the middle and the end of the week.
- Monday and Wednesday: Both days show above-average steps, reflecting good mid-week engagement in physical activity.
- Sunday, Thursday, and Friday: These days exhibit slightly lower average steps. It is plausible to speculate that the participants may experience fatigue after a long workday on Fridays, and Sundays might be a preferred day to rest before the start of the work week.

5.3.5. Average Hourly Sleep from Sunday to Saturday Findings

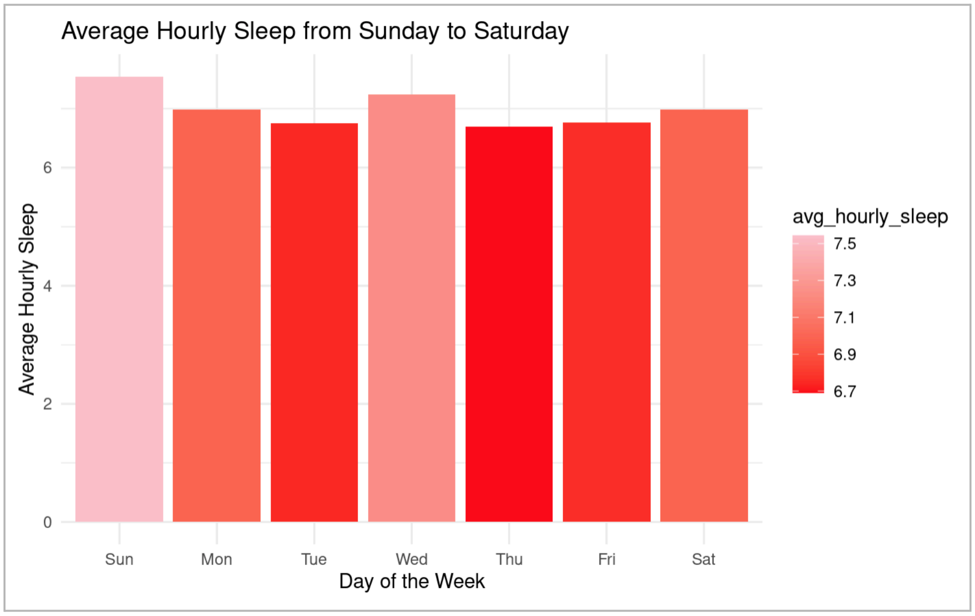


Figure 5.3.5. Average Hourly Sleep from Sunday to Saturday

Figure 5.3.5 illustrates distinct patterns in average hourly sleep throughout the week:

- Sunday and Wednesday: These days, consistently record higher average hourly sleep, suggesting that participants may allocate more time for rest these days.
- Monday, Thursday, and Saturday: These days show a moderate average hourly sleep duration.
- Tuesday and Friday: Both days exhibit slightly lower average hourly sleep.

6. Act

The purpose of this step is to provide the stakeholders with recommendations based on the findings, enabling them to make data-driven decisions. The project aims to analyze smart device usage data to gain insight into how consumers use non-Bellabeat smart devices. This insight will help guide the marketing strategy for the company. Based on the analysis, I will recommend both short-term and long-term marketing strategies for Bellabeat.

6.1. Short-term Marketing Strategies

In pursuit of immediate goals, Bellabeat is implementing short-term marketing strategies that prioritize quick results:

6.1.1. Sleep and Steps Challenge:

Bellabeat's strategic "Sleep and Steps Challenge" aims to enhance user's well-being, leveraging the popularity of the Activity Tracker and Sleep Tracker features. Key elements making it a potent short-term marketing strategy include:

- **Prompt Users Join 'Sleep & Steps Challenge'**: Use personalized notification to prompt users to participate in the 'Sleep and Step Challenge' and explore Bellabeat's app for comprehensive activity and sleep tracking.

- **Clearly Outline the Goals:** Establish clear and achievable goals, such as achieving 10,000 steps daily or ensuring 7 hours of sleep nightly.

- **Implement a Rewards System:** Introduces a rewarding system featuring badges, virtual rewards, and exclusive discounts to boost user motivation and foster a sense of community.

6.1.2. Calorie Burn Awareness Campaign:

The Calorie Burn Awareness Campaign connects the positive relationship between total step and calorie burns. With a daily average step count of 7,638 and an average daily calorie burn of 2,308, promoting the 10,000 Steps Goal becomes pivotal:

- **Promote the 10,000 Steps Goals:** Encourage participants to strive for 10,000 steps daily.

- **Provide Tips for Success:** Share creative ways to increase the daily step count, such as short walks during breaks or using stars. Emphasize the importance of consistency and gradual progression.

- **Challenge Participants:** Introduce weekly challenges like the “Lunchtime Walkathon” to maintain engagement and motivation.

6.1.3. BMI Health Awareness Week:

Designed with a BMI mean of 35.19 in ind, classifying participants in the overweight range with the medium-risk obesity.

- **Personalized Content Delivery:** Provide users with personalized information about BMI and its significance in overall health.

- **Tailored Nutrition Guidance**: Encourage users to adopt and track healthy habits, promoting mindful eating and hydration.

- **Interactive Challenges:** Introduce engaging daily or weekly challenges, motivating users to incorporate physical activity into their routines for a healthier lifestyle.

6.1.4. Optimize Specific Marketing Campaigns on Targeted Days:

Align promotional efforts with user trend, such as:

- **Peak Usage Days (especially Fridays):** Share tips and insights on these days to foster a sense of community, motivation, and active participation.

- **Days with Highest Average Steps (especially Saturday):** Encourage users to celebrate their fitness achievements, share challenges, and build a supportive community.

- **Stress Relief Campaign on Fridays:** Recognize the observed increase in sedentary minutes on Fridays, potentially due to long workdays.

- **Promoting Rest and Recovery on Sundays:** Acknowledge the slightly lower average steps on Sundays as users prepare for the upcoming workweek starting on Monday.

6.1.5. Actively Seeking User Feedback:

Initiate feedback collection through surveys, review and social media interactions. Use this feedback to guide future product updates and enhancements, ensuring that Bellabeat products are in line with user needs and expectations.

6.1.6. A more thorough analysis is required:

Despite providing valuable insights, the data faces limitations, including being outdated, captured within a narrow time frame, derived from a small sample size, lacking demographic information, and offering no specific details. A thorough analysis and understanding of user behavior and habits are imperative before implementing any strategies.

6.2. **Long-term Marketing Strategies**

Long-term marketing strategies are visionary plans aiming for lasting value and sustained competitive advantage. Bellabeat's comprehensive approach includes:

6.2.1. Continuously innovate Bellabeat Products:

Ensure Bellabeat remains at the forefront of technology and wellness trends, consistently innovating products to meet evolving user expectations and maintain a competitive edge.

- **Proactive Inactivity Notification:** Address the findings that 81.3% of participants 81.3% are inactive for extended periods by introducing a proactive feature. Implement a pop-up notification on Bellabeat products, encouraging users to move after being inactive for one hour. This promotes healthier habits and aligns with wellness goals.

- **Design an Attractive and User-friendly Collection, Especially for Leaf and Time:** Emphasize not only the functionality but also the aesthetic appeal of the products. Encourage users to choose Bellabeat products for both their usefulness and visually appealing, stylish design.

6.2.2. User Retentions and Brand Advocacy Programs

To ensure long-term user engagement, implement retention programs and launch a brand advocacy initiative that keeps users connected and excited about the Bellabeat products.

- **Loyalty Rewards and Exclusive Content:** Offer loyalty rewards, exclusive content, or challenges to motivate users to consistently use the smart device and the application. Provide incentives for continued engagement, creating a sense of value and exclusivity of loyal Bellabeat users

- **Launch a Brand Advocacy Program:** Invite satisfied users to become ambassadors for Bellabeat. Incentivize users to share their positive experience, testimonials, and success stories, contributing to positive word-of-mouth marketing.

- **Recognize and Reward Users Participating in Brand Advocacy Programs to Build a Positive Brand Image:** Offer incentives for users participating in the brand advocacy programs. This could include exclusive discounts, early access to new features, or a special recognition within the Bellabeat community. This approach will contribute to building a positive brand image.

7. Summary

I have followed the steps of the data analysis process - **Ask**, **Prepare, Process**, **Analyze**, **Share**, and **Act -** and applied the insight gained to understand consumer usage of non-Bellabeat smart devices. The objective is to assist Bellabeat in improving its marketing strategy by gaining a comprehensive understanding of the preferences and behavior of its users with these smart devices.

7.1. Findings

- **Activity and Sleep Tracker Dominance:** The Activity and Sleep Tracker features emerge as the most frequently used among users, with a consistent focus on tracking activity data. (See Figure 5.1.1 and 5.1.2)

- **Day-wise Activity Trends:** The Activity Tracker is consistently used every day of the week, with a slight increase on Fridays. This aligns with the common notion of the end of the workweek, suggesting that users may be more focused on ensuring they meet their fitness goals before the weekend starts. (See Figure 5.1.3)

- **Relationship and Patterns:**
    - As the total number of steps increases, the number of minutes spent sitting down decreases, especially for steps ranging between 0 to 10,000. This is known as a negative relationship.(See Figure 5.2.1)
    - When there is a strong connection between two or more variables, we call it a positive relationship. Positive relationships exist between total minutes asleep and total time in bed (see Fig. 5.2.2), total step and calorie burns (see Fig. 5.2.3), as well as active minutes and calorie burns (see Fig 5.2.4).

- **Usage Patterns:**
  - Participants spend a significant portion, 81.3% of their time in sedentary minutes (991.2 minutes) and only 1.7% in very active minutes (21.16 minutes).(see Figure 5.3.1 and Figure 5.3.2)
  - The daily average step count is 7,638 and the average daily calorie burn is 2,308. (See Figure 5.3.2)
  - Participants, on average, sleep for 7 hours per day (See Figure 5.3.2)

- **Health Metrics:** The average weight is 72.04 kg or 158.8 pounds, resulting in a mean BMI 35.19, classifying participants in the overweight range with the medium-risk obesity.(See Figure 5.3.2)

- **Day-wise Activity Peaks:** The highest average minutes of active activity are consistently recorded on Tuesday. In contrast, both light and sedentary activity show their highest averages on Fridays. (See Figure 5.3.3)

- **Steps and Sleep Variation:**
  - Tuesday and Saturday consistently record higher average steps, while Sunday, Thursday, and Friday exhibit slightly lower average steps.(See Figure 5.3.4)
  - Sunday and Wednesday consistently record higher average sleep, while Tuesday and Friday exhibit slightly lower average hourly sleep. (See Figure 5.3.5)

- **There is one finding that deserves questioning:**
  While the Activity Tracker sees consistent usage throughout the week, Fridays exhibit a slight increase in usage (see Figure 5.1.3). Curiously, on the same day, both light and sedentary activities reach their highest averages (see Figure 5.3.3).

  A more detailed analysis is required to understand how activity measurements, spanning from active to sedentary, are determined. The elevated Friday usage may be attributed to a diverse user base with varying activity patterns.

7.2. Recommendations

7.2.1. Short-term Marketing Strategies:

- **“Sleep & Steps” Challenge**:
    - Prompt users with a notification to join “Sleep & Steps Challenge” by using Bellabeat's App. Clearly outline the goals - for example, achieve 10,000 steps daily or 7 hours of sleep nightly.
    - Implement a reward system where users earn badges, virtual rewards, or exclusive discounts for completing the challenge.

- **Calorie Burn Awareness Campaign:** Develop a campaign to promote daily 10,000 steps goals that aim to burn calories. Share helpful tips and exciting challenges to assist users in achieving this daily milestone.

- **BMI Health Awareness Week**: Dedicate a short-term campaign to BMI health awareness, providing users with personalized content, nutrition guidance and interactive challenges to address overweight concerns.

- **Optimize specific marketing campaigns on targeted days**, such as:
    - Optimize marketing campaigns or promotions on peak usage days, especially on Friday.
    - Launch promotions or engagement activities on days with the highest average steps, particularly Saturday.
    - Implement a stress relief campaign on Friday, and
    - Promote rest, recovery, and self-care on Sunday and Friday.

- **Actively seeking user feedback**: Initiate feedback collection and act on this feedback to inform future product updates and enhancement.

- **A more thorough analysis is required**: Due to data limitations, a more comprehensive analysis is required. Before implementing strategies, it is crucial to have a complete understanding of users behavior.

7.2.2. Long-term Marketing Strategies:

- **Continuously innovate Bellabeat products**, including:
    - Proactive inactivity notification: Introducing a pop-up notification to encourage movement after an hour of inactivity.
    - Design an attractive and user-friendly collection for Leaf and Time.

- **User retention and brand advocacy programs**, including:
    - Implement retention programs, offering loyalty rewards, exclusive content, or challenges to consistently motivate the smart device and app usage.
    - Launch a brand advocacy program, inviting satisfied users to become Bellabeat ambassadors.
    - Recognize and reward users participating in brand advocacy programs to build a positive brand image.