Bella Beats Case Study

Emma Schenegg

2025-07-23

Introduction

Business Context

Bellabeat is a wellness technology company focused on creating smart products designed specifically for women. Their devices track physical activity, sleep, stress, and reproductive health to promote holistic wellbeing. Bellabeat emphasizes combining stylish design with health insights to support a balanced lifestyle. Their products serve as lifestyle partners, helping women monitor both mental and physical health. The brand is known for female-centric features and empowering users through personalized wellness data. Sršen, co-founder of Bellabeat, has requested an analysis of smart device usage data to gain insights into how consumers—especially women—use non-Bellabeat tracking devices. The objective is to understand key user behaviors, identify trending features, and explore opportunities to improve Bellabeat's product offerings and marketing strategy.

Guiding Problem

The primary problem identifying which trends are most relevant to Bellabeat's target audience. The goal is to reduce internal bias by examining the behaviours of users from competing products and using those findings to: • Improve Bellabeat's existing product features • Tailor marketing campaigns based on real consumer behaviour • Provide stakeholders with evidence-based recommendations for product development Stakeholders • Bellabeat co-founders • The marketing team • Product development and design teams

Datasets

The data used in this analysis comes from the publicly available FitBit Fitness Tracker Data provided by Möbius on Kaggle. It includes anonymized activity, sleep, heart rate, and weight data collected from 33 to 35 users over a period of 31 to 60 days between March and May 2016. Four key datasets were utilized: dailyActivity_merged.csv (tracking total steps, activity intensity levels, sedentary minutes, and calories burned), heartrate_seconds_merged.csv (second-by-second heart rate data in BPM), sleepDay_merged.csv (sleep duration, number of sleep records, and time spent in bed), and weightLogInfo_merged.csv (user weight in kilograms, BMI, and manual/automatic entry type). The activity, heart rate, and weight data were originally split across two separate datasets covering different time periods—12/03/2016 to 11/04/2016 and 12/04/2016 to 12/05/2016—and this analysis will later combine them using the rbind() function to ensure a continuous timeline for analysis. The sleep data is already consolidated and did not require merging.

Business Task Statement

To gain insights into trends in women's behavior with competitor smart tracking devices in order to guide Bellabeat's marketing strategy and product development, helping the brand better connect with existing and potential customers.

Analysis

Install packages

```
options(repos = c(CRAN = "https://cloud.r-project.org"))

install.packages("tidyverse") # (ggplot2, dplyr,tidyr,readr, stringr etc...)
install.packages("lubridate") # Manipulation of date/time
install.packages("corrplot")  # Correlation heat map
```

Load packages

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.5.1
```

```
## Warning: package 'tidyr' was built under R version 4.5.1
```

```
## Warning: package 'dplyr' was built under R version 4.5.1
```

```
## Warning: package 'lubridate' was built under R version 4.5.1
```

```
## ── Attaching core tidyverse packages ──────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.2     ✓ tibble    3.2.1
## ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
## ✓ purrr     1.0.4
## ── Conflicts ──────────────────────────────────── tidyverse_conflicts() ──
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
library(lubridate)
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.5.1
```

```
## corrplot 0.95 loaded
```

Bind and load data sets

The rbind() function was used to combine matching data for Activity, Heart Rate and Weight data sets from two distinct periods — March 12 to April 11, 2016, and April 12 to May 12, 2016 — as they were stored in separate datasets. This allowed for the creation of a continuous dataset covering the full date range for consistent

analysis. As all sleep records were already consolidated in one dataset, no additional merging was necessary using the rbind() function.

```
Activity <- rbind(read.csv("C:/Users/emma3/Downloads/Fitabase Data 4.12.16-5.12.16/dailyActiv
ity_merged2.csv"),read.csv("C:/Users/emma3/Downloads/mturkfitbit_export_3.12.16-4.11.16/daily
Activity_merged.csv"))

Weight <- rbind(read.csv("C:/Users/emma3/Downloads/Fitabase Data 4.12.16-5.12.16/weightLogInf
o_merged.csv"),read.csv("C:/Users/emma3/Downloads/mturkfitbit_export_3.12.16-4.11.16/weightLo
gInfo_merged1.csv"))

Heartrate <- rbind(read.csv("C:/Users/emma3/Downloads/Fitabase Data 4.12.16-5.12.16/heartrate
_seconds_merged.csv"), read.csv("C:/Users/emma3/Downloads/mturkfitbit_export_3.12.16-4.11.16/
heartrate_seconds_merged1.csv"))

Sleep <- read.csv("C:/Users/emma3/Downloads/Fitabase Data 4.12.16-5.12.16/sleepDay_merged.cs
v")
```

Visualise Data

The head() function was used to preview the first few rows of the dataset, allowing for a quick inspection of the column structure and a general understanding of the data.

```
head(Activity)
```

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

```
head(Weight)
```

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

```
head(Heartrate)
```

```
##          Id                 Time Value
## 1 2022484408 4/12/2016 7:21:00 AM    97
## 2 2022484408 4/12/2016 7:21:05 AM   102
## 3 2022484408 4/12/2016 7:21:10 AM   105
## 4 2022484408 4/12/2016 7:21:20 AM   103
## 5 2022484408 4/12/2016 7:21:25 AM   101
## 6 2022484408 4/12/2016 7:22:05 AM    95
```

```
head(Sleep)
```

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

Inspecting Data Formats

The str() function was used to examine the structure of the dataset and verify that the column formats were consistent and appropriate for analysis.

```
str(Activity)
```

```
## 'data.frame':    1397 obs. of  15 variables:
##  $ Id                     : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate           : chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps             : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819

...
##  $ TotalDistance          : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance        : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance     : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance    : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes      : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes    : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes   : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes       : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ Calories               : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

```
str(Weight)
```

```
## 'data.frame':    100 obs. of  8 variables:
##  $ Id            : num  1.50e+09 1.50e+09 1.93e+09 2.87e+09 2.87e+09 ...
##  $ Date          : chr  "5/2/2016 11:59:59 PM" "5/3/2016 11:59:59 PM" "4/13/2016 1:08:52 A
M" "4/21/2016 11:59:59 PM" ...
##  $ WeightKg      : num  52.6 52.6 133.5 56.7 57.3 ...
##  $ WeightPounds  : num  116 116 294 125 126 ...
##  $ Fat           : int  22 NA NA NA NA 25 NA NA NA NA ...
##  $ BMI           : num  22.6 22.6 47.5 21.5 21.7 ...
##  $ IsManualReport: chr  "True" "True" "False" "True" ...
##  $ LogId         : num  1.46e+12 1.46e+12 1.46e+12 1.46e+12 1.46e+12 ...
```

```
str(Heartrate)
```

```
## 'data.frame':    3638339 obs. of  3 variables:
##  $ Id   : num  2.02e+09 2.02e+09 2.02e+09 2.02e+09 2.02e+09 ...
##  $ Time : chr  "4/12/2016 7:21:00 AM" "4/12/2016 7:21:05 AM" "4/12/2016 7:21:10 AM" "4/12/
2016 7:21:20 AM" ...
##  $ Value: int  97 102 105 103 101 95 91 93 94 93 ...
```

```
str(sleep)
```

```
## 'data.frame':    20 obs. of  3 variables:
##  $ extra: num  0.7 -1.6 -0.2 -1.2 -0.1 3.4 3.7 0.8 0 2 ...
##  $ group: Factor w/ 2 levels "1","2": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ID   : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
```

The date and time values in the datasets were initially stored as character strings rather than in proper date-time formats. Time values used the 12-hour clock format with AM/PM indicators instead of the standard 24-hour format (HH:MM:SS). Additionally, dates followed the U.S. format (MM/DD/YYYY), which differs from the UK

format (DD/MM/YYYY). In terms of structure, the activity dataset recorded only the date (without specific times), whereas the sleep dataset included time stamps. However, all time entries in the sleep dataset were set to 00:00:00, offering no variation in time-of-day information.

Adjusting Dates

Dates format

The as.POSIXct() function was used to convert character strings into proper date-time format, enabling accurate time-based analysis.

The Sys.timezone() function was applied to ensure that the timestamps aligned with the system's actual time zone, maintaining consistency in time-related data.

Additionally, time was reformatted to the UK standard (DD/MM/YYYY HH:MM:SS) to improve readability and prevent confusion. The reformatted values were stored in a new column named "FormattedTime".

For Weight, Heartrate and Sleep data sets, the format function was used with the %I:%M:%S %p directive to correctly interpret and format time values stored in the 12-hour clock format with AM/PM indicators, as found in the original datasets. Since Sleep Data set time are all 00:00:00 and don't indicate any specific time where people felt asleep, only the date is relevant and was kept.

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

Heartrate$Time <- as.POSIXct(Heartrate$Time, format="%m/%d/%Y %I:%M:%S %p", tz= Sys.timezone
())
Heartrate$FormattedTime <- format(Heartrate$Time, format = "%d/%m/%Y %H:%M:%S")

Sleep$SleepDay <- as.POSIXct(Sleep$SleepDay, format="%m/%d/%Y %H:%M:%S", tz= Sys.timezone())
Sleep$FormattedTime <- format(Sleep$SleepDay, format = "%d/%m/%Y")
```

Since the Activity dataset includes only dates (without time) and uses the U.S. date format, only the date component was reformatted to the U.K. style for consistency and clarity.

```
Activity$ActivityDate <- as.POSIXct(Activity$ActivityDate, format="%m/%d/%Y", tz= Sys.timezon
e())
Activity$FormattedTime <- format(Activity$ActivityDate, format = "%d/%m/%Y")
```

Remove previous date variable

Since a new date-time variable has been created in the U.K. format, the original date columns are no longer needed and can be safely removed to avoid redundancy.

```
Activity <- Activity %>% select(-ActivityDate)
Weight <- Weight %>%  select(-Date)
Heartrate <- Heartrate %>% select(-Time)
Sleep <- Sleep %>% select(-SleepDay)
```

Distinct values

The n_distinct() function was used to count the number of unique participant IDs in each dataset, providing insight into the sample size and ensuring no duplicate records are counted.

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

```
## [1] 35
```

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

```
## [1] 13
```

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

```
## [1] 15
```

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

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

Mean steps

Evaluate whether, on average, users achieve the widely recommended daily target of 10,000 steps, based on guidelines by Tudor-Locke & Bassett (2004), to assess overall physical activity levels.

```
mean(Activity$TotalSteps, na.rm = TRUE)
```

```
## [1] 7280.898
```

Average steps = 7281, the mean doesn't reach daily steps recommendation

Creating New Variables

Total Active Minutes

Generate a new column that sums Lightly Active, Fairly Active, and Very Active minutes to represent the total daily active minutes, enabling easier analysis of overall activity levels.

```
Activity <- Activity %>%
  mutate(TotalActiveMinutes = LightlyActiveMinutes + FairlyActiveMinutes + VeryActiveMinutes)
```

Awake Sedentary Time

Generate a new column for sedentary time that doesn't take into account sleeping time and where sleeping time higher than sedentary time is not taken into account (After checking Sedentary awake time values some are negative and cannot be accounted as not accurate since total minutes asleep cannot be more than sedentary time).

```
Activity$FormattedTime <- dmy(Activity$FormattedTime) # parse the datetime columns
Sleep$FormattedTime <- dmy(Sleep$FormattedTime)

Activity %>% count(Id, FormattedTime) %>% filter(n > 1) # check if id/formatted time have uni
que values
```

```
##            Id FormattedTime n
## 1  1503960366    2016-04-12 2
## 2  1624580081    2016-04-12 2
## 3  1844505072    2016-04-12 2
## 4  1927972279    2016-04-12 2
## 5  2022484408    2016-04-12 2
## 6  2026352035    2016-04-12 2
## 7  2320127002    2016-04-12 2
## 8  2347167796    2016-04-12 2
## 9  2873212765    2016-04-12 2
## 10 3977333714    2016-04-12 2
## 11 4020332650    2016-04-12 2
## 12 4057192912    2016-04-12 2
## 13 4445114986    2016-04-12 2
## 14 4558609924    2016-04-12 2
## 15 4702921684    2016-04-12 2
## 16 5553957443    2016-04-12 2
## 17 6962181067    2016-04-12 2
## 18 7007744171    2016-04-12 2
## 19 7086361926    2016-04-12 2
## 20 8053475328    2016-04-12 2
## 21 8253242879    2016-04-12 2
## 22 8378563200    2016-04-12 2
## 23 8792009665    2016-04-12 2
## 24 8877689391    2016-04-12 2
```

```
Sleep %>% count(Id, FormattedTime) %>% filter(n > 1)
```

```
##           Id FormattedTime n
## 1 4388161847    2016-05-05 2
## 2 4702921684    2016-05-07 2
## 3 8378563200    2016-04-25 2
```

```
Sleep_unique <- Sleep %>%
  distinct(Id, FormattedTime, .keep_all = TRUE) # remove duplicates

Activity_unique <-Activity %>%
  distinct(Id, FormattedTime, .keep_all = TRUE)

Activity <- Activity_unique %>%    # join Sleeping time and Sedentary time that belong to two
different data frames
  full_join(
    Sleep_unique[, c("Id", "FormattedTime", "TotalMinutesAsleep")],
    by = c("Id", "FormattedTime") # group new column by Id and date
  ) %>%
  arrange(Id, FormattedTime) %>%
  mutate(
    SedentaryAwakeTime = ifelse(
      SedentaryMinutes > TotalMinutesAsleep,
      SedentaryMinutes - TotalMinutesAsleep, NA)) # only keep values where Sedentary > Sleep,
else NA
# value_if_true, value_if_false (NA)
```

Time In Bed Awake

Generate a new column for the time spent in bed awake by subtracting Sleeping time to Time in bed.

```
Sleep$TimeInBedAwake <- Sleep$TotalTimeInBed - Sleep$TotalMinutesAsleep
```

Means

Now, let's examine the average values (means) for the key variables in our dataset to better understand overall trends.

Sleep

```
Sleep %>%
  select(TimeInBedAwake, TotalMinutesAsleep) %>%
  summary()
```

```
##  TimeInBedAwake    TotalMinutesAsleep
##  Min.   :  0.00   Min.   : 58.0
##  1st Qu.: 17.00   1st Qu.:361.0
##  Median : 25.00   Median :433.0
##  Mean   : 39.17   Mean   :419.5
##  3rd Qu.: 40.00   3rd Qu.:490.0
##  Max.   :371.00   Max.   :796.0
```

Time spent awake in bed ranges widely from 0 to 371 minutes, with an average around 39 minutes. Total minutes asleep vary between 58 and 796, averaging about 420 minutes.

Activity

```
Activity %>%
  select(TotalSteps, VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, TotalActiv
eMinutes, SedentaryAwakeTime) %>%
  summary()
```

```
##    TotalSteps      VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes
## Min.   :    0   Min.   :  0.00   Min.   :  0.0   Min.   :  0.0
## 1st Qu.: 3321   1st Qu.:  0.00   1st Qu.:  0.0   1st Qu.:117.0
## Median : 7142   Median :  2.00   Median :  6.0   Median :196.0
## Mean   : 7377   Mean   : 19.87   Mean   : 13.6   Mean   :188.1
## 3rd Qu.:10645   3rd Qu.: 30.00   3rd Qu.: 18.0   3rd Qu.:263.0
## Max.   :36019   Max.   :210.00   Max.   :660.0   Max.   :720.0
##
## TotalActiveMinutes SedentaryAwakeTime
## Min.   :  0.0   Min.   :   1.0
## 1st Qu.:135.0   1st Qu.: 209.0
## Median :244.0   Median : 311.0
## Mean   :221.6   Mean   : 343.5
## 3rd Qu.:315.0   3rd Qu.: 425.8
## Max.   :720.0   Max.   :1206.0
##                 NA's   :1007
```

Daily total steps range from 0 to 36,019, with an average around 7,377 steps. Very active minutes vary widely (0–210) but average about 20 minutes, while fairly and lightly active minutes average 14 and 188 minutes respectively. Total active minutes average 222, and sedentary daytime minutes average around 344min (5h44m) per day.

Weight

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

```
##       BMI           WeightKg
## Min.   :21.45   Min.   : 52.6
## 1st Qu.:24.00   1st Qu.: 61.5
## Median :24.39   Median : 62.5
## Mean   :25.37   Mean   : 72.5
## 3rd Qu.:25.59   3rd Qu.: 85.3
## Max.   :47.54   Max.   :133.5
```

BMI values range from 21.45 to 47.54, with a mean of about 25.37, while weights range from 52.6 kg to 133.5 kg, averaging 72.5 kg.

Heart rate

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

```
##       Value
## Min.   : 36.0
## 1st Qu.: 64.0
## Median : 74.0
## Mean   : 78.1
## 3rd Qu.: 88.0
## Max.   :203.0
```

Heart rate values span from 36 to 203 bpm, with an average around 78 bpm and a median of 74 bpm.

Daily Activity Breakdown

Activity

```
Activity$weekday <- weekdays(Activity$FormattedTime)

summary_weekday <- Activity %>%
  group_by(weekday) %>%
  summarise(
    avg_steps = mean(TotalSteps, na.rm = TRUE),
    avg_SedentaryAwakeTime = mean(SedentaryAwakeTime, na.rm = TRUE),
    avg_TotalActiveMinutes = mean(TotalActiveMinutes, na.rm = TRUE),
    avg_Calories = mean(Calories, na.rm = TRUE)
  ) %>%
  mutate(weekday = factor(weekday,levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "F
riday", "Saturday", "Sunday"))) # order weekdays Monday to Sunday order rather than alphabeti
cally (mutate).


plot_data <- summary_weekday %>%
  pivot_longer(cols = starts_with("avg_"),
               names_to = "Metric",
               values_to = "Value")

ggplot(plot_data, aes(x = weekday, y = Value, fill = Metric)) +
  geom_col(position = "dodge") +
  facet_wrap(~ Metric, scales = "free_y") +  # separate y-axis scales per metric
  labs(
    title = "Average Activity by Day of the Week",
    x = "Day of the Week",
    y = "Average Value",
    fill = "Metric"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Average Activity by Day of the Week
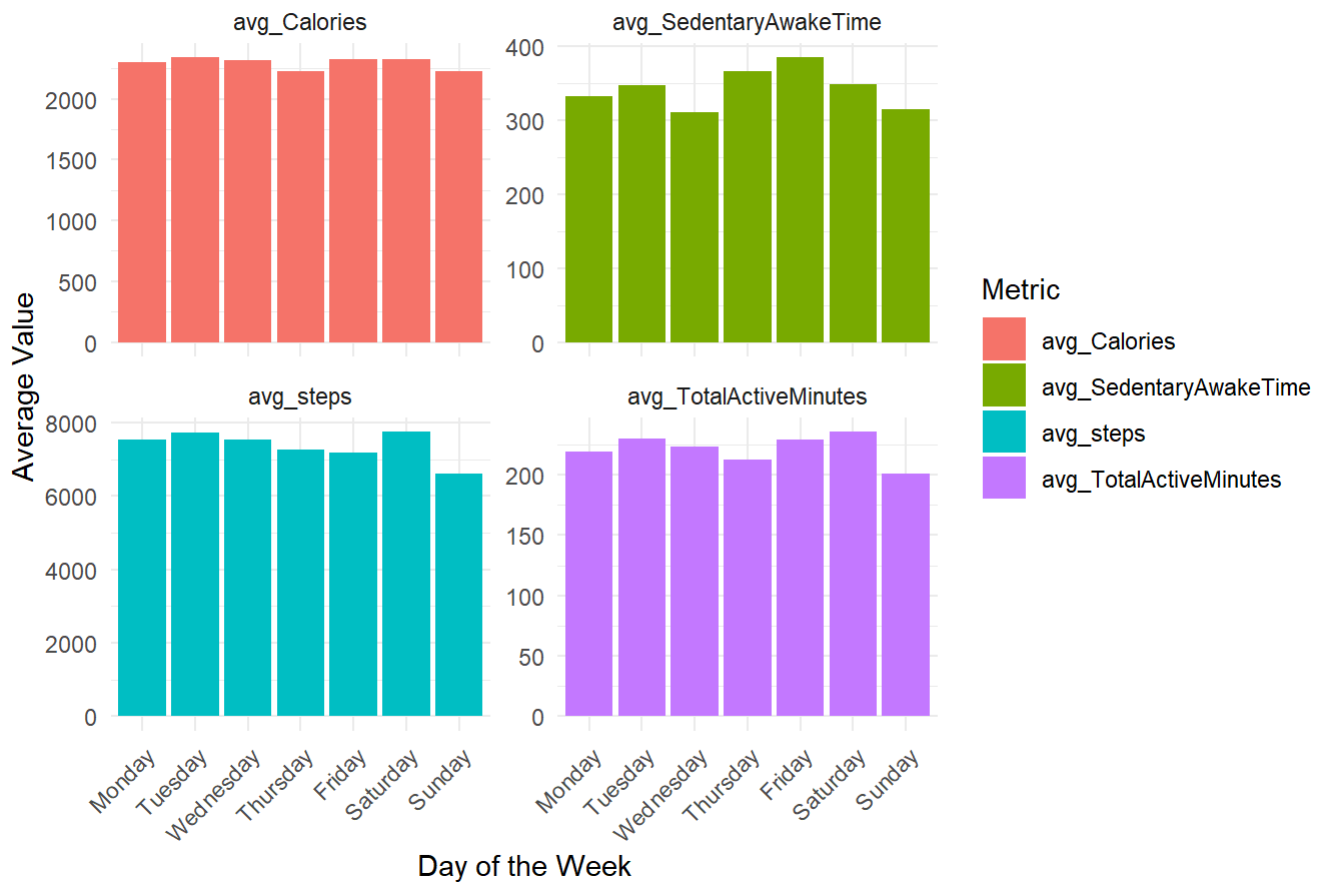
The bar chart show a slightly higher activity and less sedentary behaviour on the weekend but not a big difference between weekend and weekdays.

Sleep

```r
Sleep$weekday <- weekdays(Sleep$FormattedTime)

summary_sleep <- Sleep %>%
  group_by(weekday) %>%
  summarise(
    avg_timeAsleep = mean(TotalMinutesAsleep, na.rm = TRUE),
    avg_timeInBed = mean(TotalTimeInBed, na.rm = TRUE),
  ) %>%
  mutate(weekday = factor(weekday,levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "F
riday", "Saturday", "Sunday")))

plot_sleep <- summary_sleep %>%
  pivot_longer(cols = starts_with("avg_"),
               names_to = "Metric",
               values_to = "Value")

ggplot(plot_sleep, aes(x = weekday, y = Value, fill = Metric)) +
  geom_col(position = "dodge",
           color = "black",
           size = 0.1
  ) +
  labs(
    title = "Average Time in Bed vs Time Asleep by Weekday",
    x = "Day of the Week",
    y = "Time (minutes)",
    fill = "Metric"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```
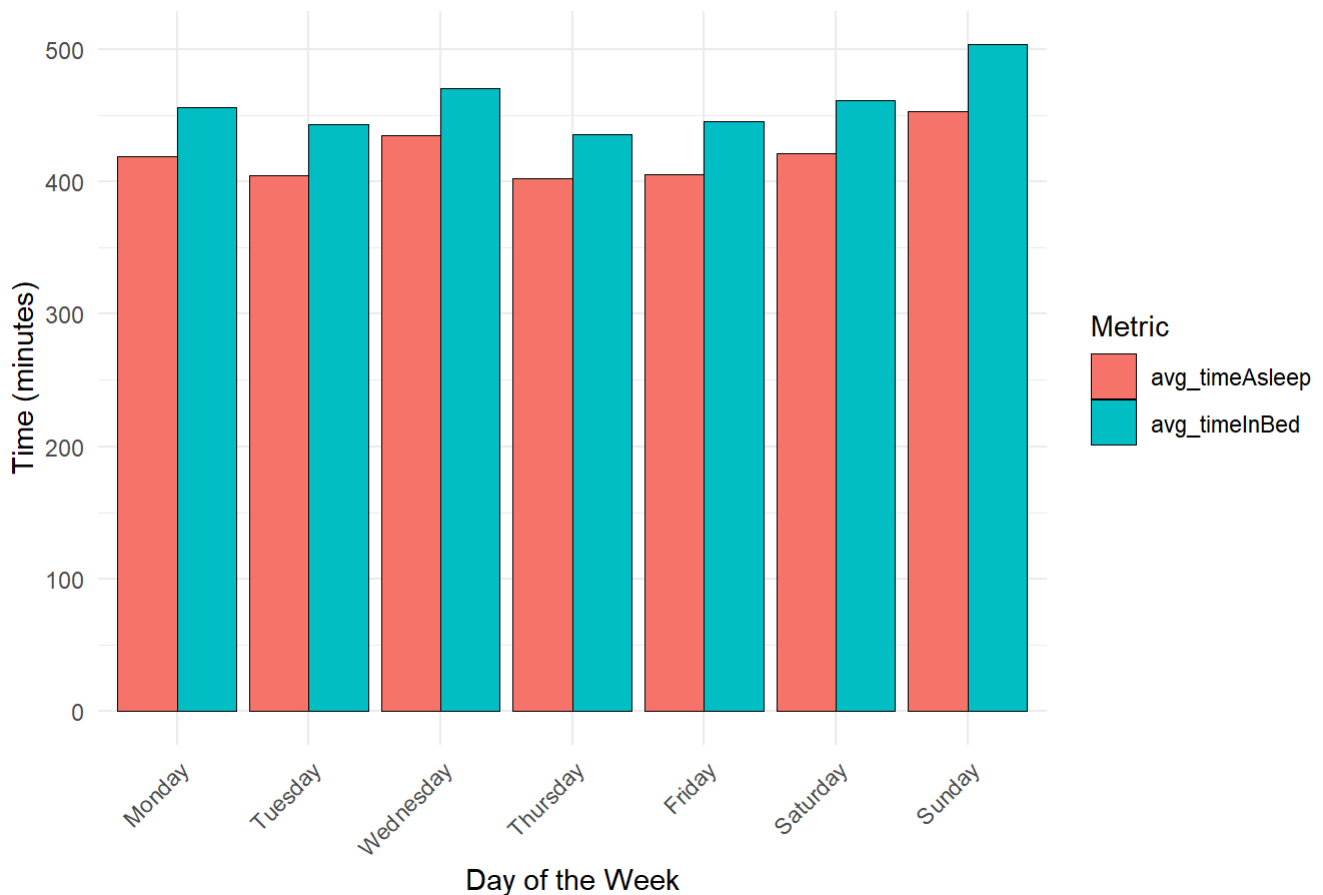
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

Average Time in Bed vs Time Asleep by Weekday

Sleep patterns indicate increased average sleep duration and Time in Bed without sleeping on weekends mainly on Sunday and a midweek peak on Wednesday, suggesting variability tied to work and rest cycles.

Heart Rate

```r
Heartrate$FormattedTime <- lubridate::dmy_hms(Heartrate$FormattedTime)

Heartrate$weekday <- weekdays(Heartrate$FormattedTime)

summary_heartrate <- Heartrate %>%
  group_by(weekday) %>%
  summarise(
    avg_bpm = mean(Value, na.rm = TRUE),
  ) %>%
  mutate(weekday = factor(weekday,levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")))

plot_heartrate <- summary_heartrate %>%
  pivot_longer(cols = starts_with("avg_"),
               names_to = "Metric",
               values_to = "Value")

ggplot(plot_heartrate, aes(x = weekday, y = Value, fill = Metric)) +
  geom_line(size = 1.2)+
  geom_point(size = 2)
```
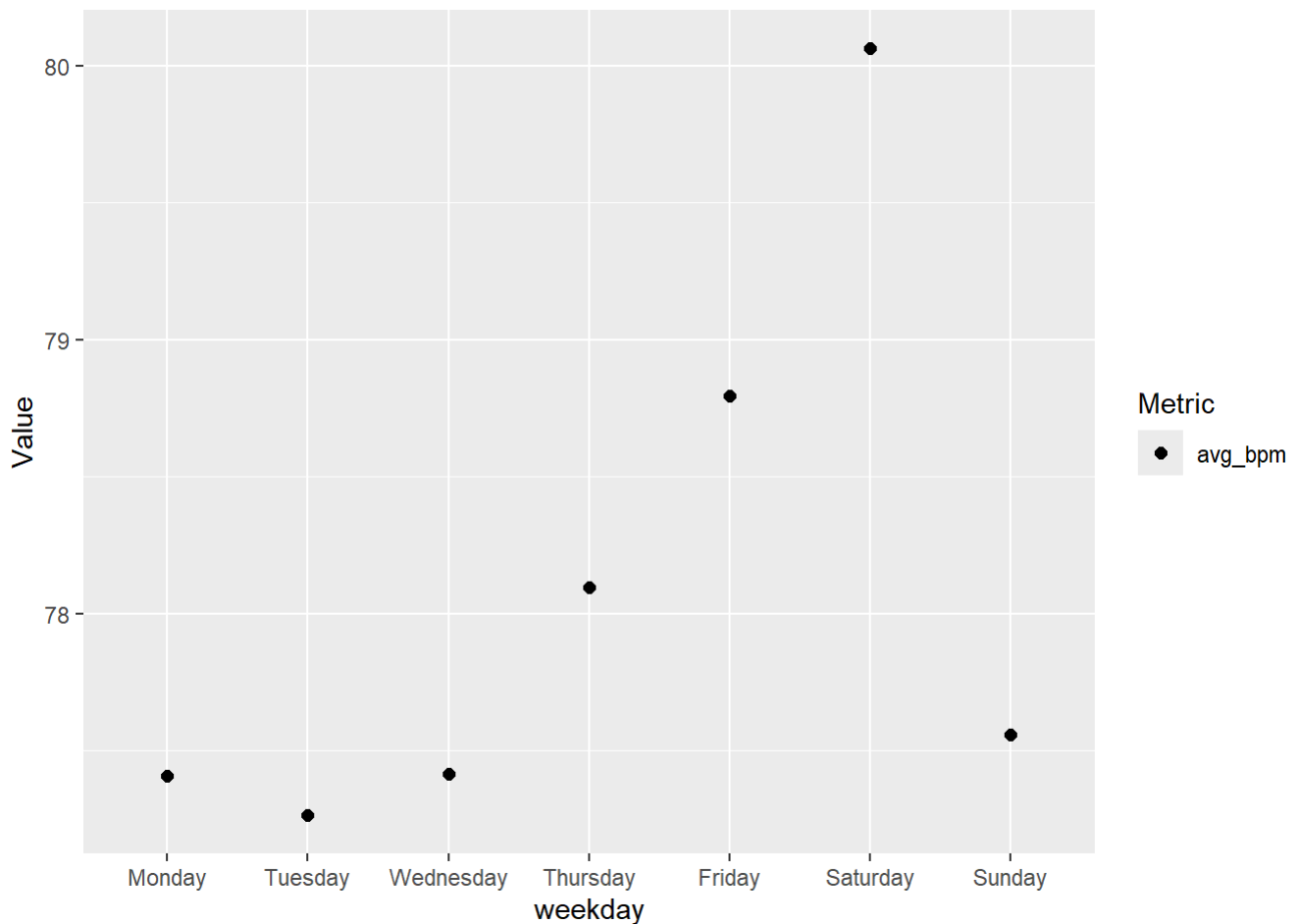
```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```

```
labs(
  title = "Average Heart rate by Weekday",
  x = "Day of the Week",
  y = "Heartrate (BPM)",
  fill = "Metric"
) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## NULL
```

```
# BPM is very slightly higher on Saturday and Lower on Monday, Tuesday and Wednesday.
```

Average heart rate shows a slight increase on Saturday, while Monday, Tuesday, Wednesday, and Sunday exhibit comparatively lower values. However the differences betweens days are light as average BPM is at its lowest on Tuesday (77.3/bpm) and at its highest on Saturday (80.5/bpm).

Weight

```r
Weight$FormattedTime <- lubridate::dmy_hms(Weight$FormattedTime)

Weight$weekday <- weekdays(Weight$FormattedTime)

summary_weight <- Weight%>%
  group_by(weekday) %>%
  summarise(
    avg_weight = mean(WeightKg, na.rm = TRUE),
    avg_BMI = mean(BMI, na.rm = TRUE),
  ) %>%
  mutate(weekday = factor(weekday,levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "F
riday", "Saturday", "Sunday")))

plot_weight <- summary_weight %>%
  pivot_longer(cols = starts_with("avg_"),
               names_to = "Metric",
               values_to = "Value")

ggplot(plot_weight, aes(x = weekday, y = Value, fill = Metric)) +
  geom_col() +
  facet_wrap(~ Metric, scales = "free_y") +
  labs(
    title = "Average Weight by Day of the Week",
    x = "Day of the Week",
    y = "Weight (Kg)",
    fill = "Metric"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```
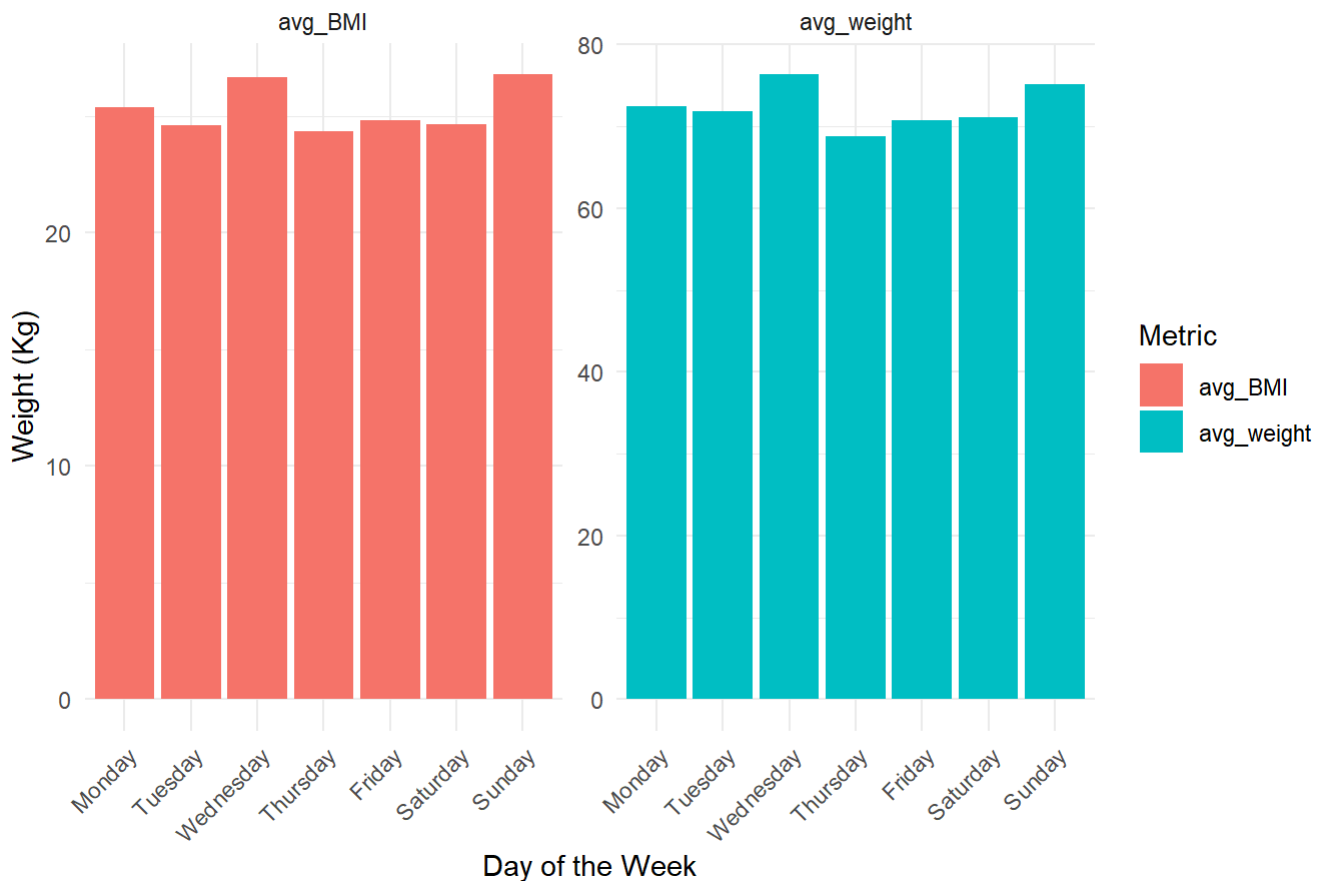
Average Weight by Day of the Week

Weight and BMI higher on Wednesday and Sunday.

Correlations

Firstly, to enable correlation analysis across all data frames, a common column named "Date" was created in each dataset. This column contains only the date portion (day/month/year) without any time information, ensuring consistency since some data frames include hours, minutes, and seconds while others do not. This standardization allows for accurate merging and comparison based solely on the date.

```
Activity$Date <- as.Date(Activity$FormattedTime, format = "%d/%m/%Y")
Sleep$Date <- as.Date (Sleep$FormattedTime, format = "%d/%m/%Y")
Heartrate$Date <- as.Date(Heartrate$FormattedTime, format = "%d/%m/%Y %H:%M:%S")
Weight$Date <- as.Date(Weight$FormattedTime, format = "%d/%m/%Y %H:%M:%S")
```

The Heart Rate dataset contains a large number of records since measurements were taken every 5 minutes throughout each day. To simplify the data and align it with other datasets that have only one record per day, the next step is to aggregate the Heart Rate data by calculating the daily average. This reduces multiple entries per day to a single representative value, facilitating easier comparison and analysis across datasets.

```
Heartrate_daily <- Heartrate %>%
  group_by(Id, Date) %>%
  summarize(AvgHeartRate = mean(Value, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'Id'. You can override using the `.groups`
## argument.
```

Multi correlation matrix

A new data frame called Allmerged is created by merging the four main datasets using the common keys Id and Date, consolidating all relevant information into a single comprehensive table for analysis.

```
Allmerged <- list(Sleep, Weight, Activity, Heartrate_daily) %>%
  reduce(full_join, by = c("Id", "Date"))
```

Before running the correlation matrix, unnecessary variables were removed to simplify the dataset and improve clarity.

```
Allmerged <- Allmerged %>% select(-Id, -Fat, -TotalDistance, -TotalSleepRecords, -TotalMinute
sAsleep.y, -WeightPounds, -TotalTimeInBed, - LogId, -LoggedActivitiesDistance, -TrackerDistan
ce, -VeryActiveDistance, -ModeratelyActiveDistance, -LightActiveDistance, -SedentaryActiveDis
tance)
```

Only the numeric columns were selected from the dataset to prevent errors during analysis.

```
numeric_data <- Allmerged %>% select(where(is.numeric))
```

To prevent errors, each numeric column was checked for a standard deviation of zero (indicating all values are identical), and the results were stored as a logical vector.

```
zero_var_cols <- sapply(numeric_data, function(x) sd(x, na.rm = TRUE) == 0)
```

Only the columns in numeric_data that do not have zero variance were retained for further analysis.

```
numeric_data_filtered <- numeric_data[, !zero_var_cols]
```

A correlation matrix was created using all relevant numeric variables to provide a clearer understanding of the relationships between them.

```
cor_matrix <- cor(numeric_data, use = "pairwise.complete.obs")
print(cor_matrix)
```

```
##                      TotalMinutesAsleep.x TimeInBedAwake    WeightKg
## TotalMinutesAsleep.x          1.000000000   -0.001334242  0.03572671
## TimeInBedAwake               -0.001334242    1.000000000  0.10182853
## WeightKg                      0.035726715    0.101828533  1.00000000
## BMI                          -0.064531995    0.073961387  0.75204413
## TotalSteps                   -0.187220464    0.026720704  0.14066142
## VeryActiveMinutes            -0.091108385   -0.077774997  0.30771486
## FairlyActiveMinutes          -0.244267455    0.318151203 -0.31501245
## LightlyActiveMinutes          0.033251329   -0.147616598 -0.26323294
## SedentaryMinutes             -0.599613188   -0.165831732  0.50157699
## Calories                     -0.027930646   -0.288504744  0.58564920
## TotalActiveMinutes           -0.063660691   -0.092633259 -0.16042018
## SedentaryAwakeTime           -0.887624315   -0.027468526  0.30527963
## AvgHeartRate                 -0.258957552   -0.239026113  0.21448447
##                              BMI TotalSteps VeryActiveMinutes
## TotalMinutesAsleep.x -0.06453199 -0.1872205       -0.09110839
## TimeInBedAwake        0.07396139  0.0267207       -0.07777500
## WeightKg              0.75204413  0.1406614        0.30771486
## BMI                   1.00000000 -0.2475269       -0.05806727
## TotalSteps           -0.24752692  1.0000000        0.67729121
## VeryActiveMinutes    -0.05806727  0.6772912        1.00000000
## FairlyActiveMinutes  -0.23645806  0.3543171        0.22941847
## LightlyActiveMinutes -0.37426708  0.5945910        0.09687130
## SedentaryMinutes      0.25401247 -0.3564955       -0.18339667
## Calories              0.09891451  0.5828115        0.59486362
## TotalActiveMinutes   -0.38932775  0.7702760        0.38377977
## SedentaryAwakeTime    0.35637940 -0.1503423       -0.03634877
## AvgHeartRate          0.08401093  0.2332076        0.11035109
##                      FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes
## TotalMinutesAsleep.x        -0.244267455           0.03325133       -0.5996132
## TimeInBedAwake               0.318151203          -0.14761660       -0.1658317
## WeightKg                    -0.315012447          -0.26323294        0.5015770
## BMI                         -0.236458055          -0.37426708        0.2540125
## TotalSteps                   0.354317083           0.59459095       -0.3564955
## VeryActiveMinutes            0.229418469           0.09687130       -0.1833967
## FairlyActiveMinutes          1.000000000           0.11209072       -0.2054211
## LightlyActiveMinutes         0.112090722           1.00000000       -0.4827047
## SedentaryMinutes            -0.205421091          -0.48270471        1.0000000
## Calories                     0.304282496           0.29223589       -0.1304318
## TotalActiveMinutes           0.364838432           0.93407700       -0.5165498
## SedentaryAwakeTime          -0.003916876          -0.30394856        0.9262972
## AvgHeartRate                 0.138547592           0.18800374        0.4320997
##                         Calories TotalActiveMinutes SedentaryAwakeTime
## TotalMinutesAsleep.x -0.02793065        -0.06366069       -0.887624315
## TimeInBedAwake       -0.28850474        -0.09263326       -0.027468526
## WeightKg              0.58564920        -0.16042018        0.305279630
## BMI                   0.09891451        -0.38932775        0.356379397
## TotalSteps            0.58281154         0.77027597       -0.150342287
## VeryActiveMinutes     0.59486362         0.38377977       -0.036348769
## FairlyActiveMinutes   0.30428250         0.36483843       -0.003916876
## LightlyActiveMinutes  0.29223589         0.93407700       -0.303948561
## SedentaryMinutes     -0.13043182        -0.51654982        0.926297202
## Calories              1.00000000         0.47120072       -0.081361771
## TotalActiveMinutes    0.47120072         1.00000000       -0.299763606
## SedentaryAwakeTime   -0.08136177        -0.29976361        1.000000000
```

```
## AvgHeartRate              0.18903271          0.22900473          0.194388992
##                         AvgHeartRate
## TotalMinutesAsleep.x    -0.25895755
## TimeInBedAwake          -0.23902611
## WeightKg                 0.21448447
## BMI                      0.08401093
## TotalSteps               0.23320757
## VeryActiveMinutes        0.11035109
## FairlyActiveMinutes      0.13854759
## LightlyActiveMinutes     0.18800374
## SedentaryMinutes         0.43209974
## Calories                 0.18903271
## TotalActiveMinutes       0.22900473
## SedentaryAwakeTime       0.19438899
## AvgHeartRate             1.00000000
```

A plot of the correlation matrix was generated to visually highlight the strength and direction of relationships between the numeric variables for easier interpretation.

```
corrplot(cor_matrix,
         method = "color",        # use colored tiles instead of numbers or circles
         type = "upper",          # show only the upper triangle (to avoid repetition)
         tl.cex = 0.8,            # text label size for variable names
         number.cex = 0.5,        # size of the numbers (correlation values)
         addCoef.col = "black",   # color of coefficients
         tl.srt    = 55,         # rotate labels 55 degrees
         tl.col = "darkred")      # color labels
```
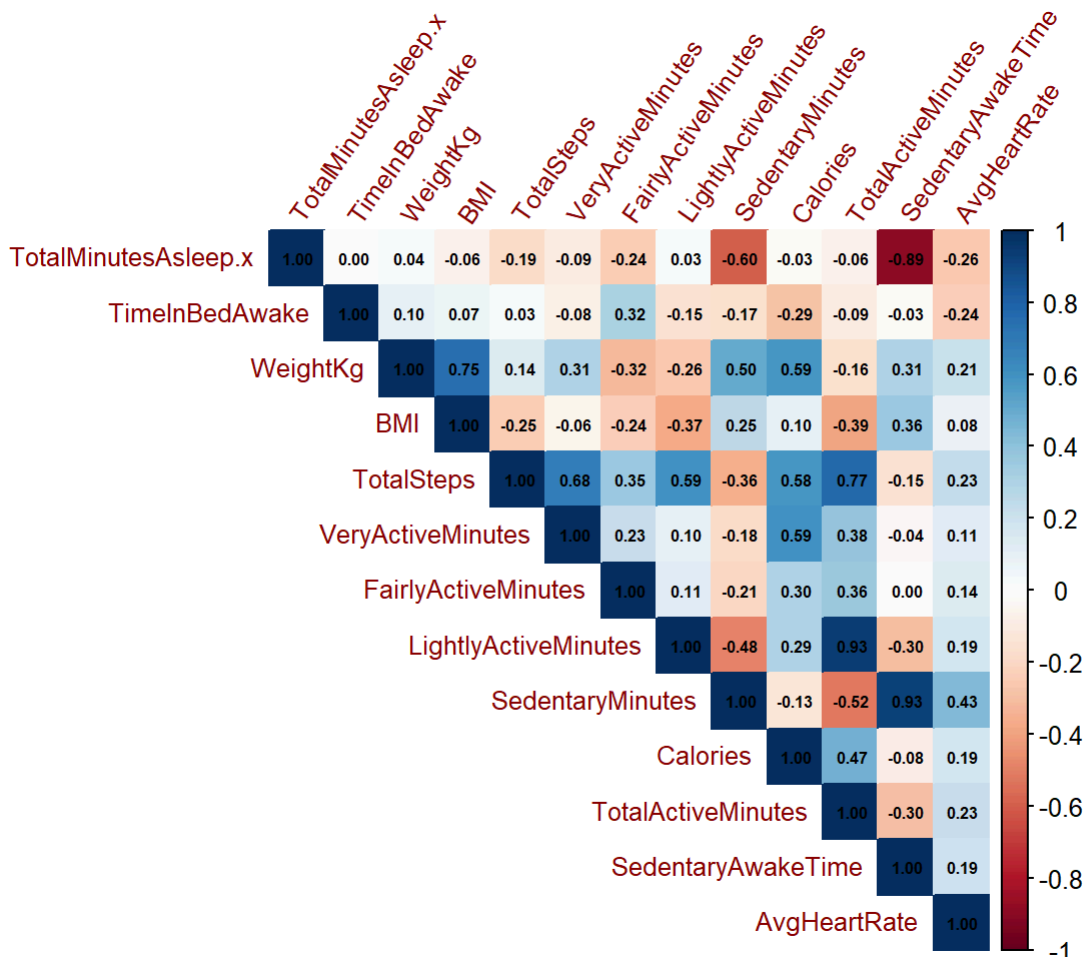
The correlation matrix reveals several interesting relationships between variables. To gain a deeper understanding, individual correlation tests and corresponding plots will be created for the most notable correlations highlighted in the matrix.

1- Sedentary time correlate with total sleeping time (without accounting for sedentary time spend sleeping).

```
cor.test(Activity$SedentaryAwakeTime, Activity$TotalMinutesAsleep, use = "complete.obs")
```

```
##
##  Pearson's product-moment correlation
##
## data:  Activity$SedentaryAwakeTime and Activity$TotalMinutesAsleep
## t = -36.97, df = 364, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9084037 -0.8649250
## sample estimates:
##        cor
## -0.8886447
```

```
# r(342) = -0.89, p < .001 (Very strong negative correlation/Very significant).

plot(Activity$SedentaryAwakeTime, Activity$TotalMinutesAsleep,
     main = "Sedentary Minutes vs Total Sleep",
     xlab = "Sedentary Minutes (awake)",
     ylab = "Total Minutes Asleep",
     pch = 19, col = "darkblue")
abline(lm(TotalMinutesAsleep ~ SedentaryAwakeTime, data = Activity), col = "red", lwd = 2) #
adding a trend line (linear regression)
```

Sedentary Minutes vs Total Sleep
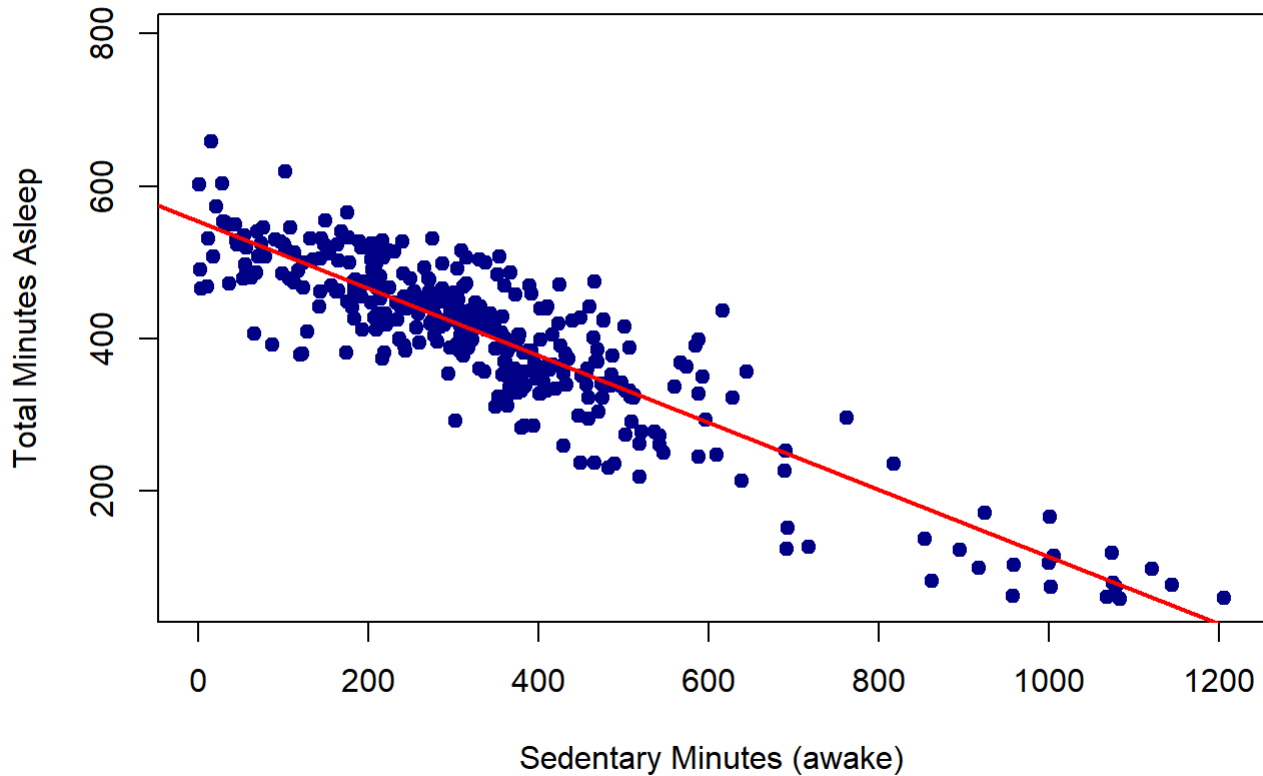
A very strong negative correlation was found between sedentary minutes and sleep duration, indicating that individuals who are more sedentary are likely to sleep less.

2- Heart rate correlate with sleeping time

```
Heartrate_daily <- Heartrate %>%
  group_by(Id, Date) %>%
  summarize(AvgHeartRate = mean(Value, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'Id'. You can override using the `.groups`
## argument.
```

```
# Aggregate the Heart Rate data by computing the daily average, in order to reduce multiple v
alues per day and match them with the corresponding single daily Sleep record.

HeartbeatSleep <- merge(Heartrate_daily, Activity, by= c("Id", "Date"))

cor.test(HeartbeatSleep$AvgHeartRate, HeartbeatSleep$TotalMinutesAsleep)
```

```
## 
##  Pearson's product-moment correlation
## 
## data:  HeartbeatSleep$AvgHeartRate and HeartbeatSleep$TotalMinutesAsleep
## t = -3.5283, df = 179, p-value = 0.0005314
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3864854 -0.1133612
## sample estimates:
##        cor
## -0.2550026
```

```
# r(179)= -0.26, p<.001 (Weak negative correlation, very significant).


plot(HeartbeatSleep$AvgHeartRate, HeartbeatSleep$TotalMinutesAsleep,
     main = "Heart Rate vs Total Sleep",
     xlab = "Heart Rate",
     ylab = "Total Minutes Asleep",
     pch = 19, col = "darkblue")
abline(lm(TotalMinutesAsleep ~ AvgHeartRate, data = HeartbeatSleep), col = "red", lwd = 2) #
adding a trend line (linear regression)
```

Heart Rate vs Total Sleep
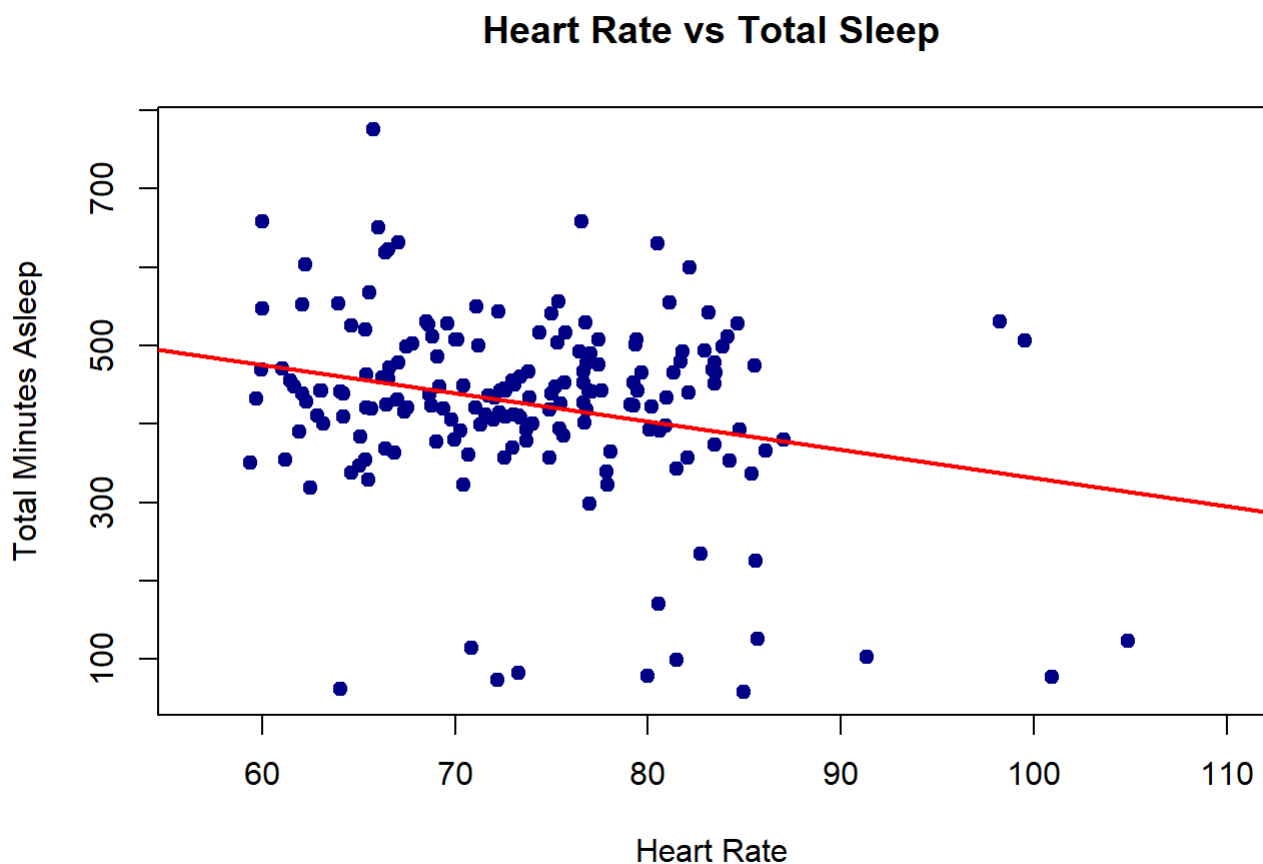
There is a negative correlation between sleep duration and heart rate, suggesting that more sleep is associated with lower resting heart rates.

3- Heart Rate correlates with Sedentary time

```
HeartrateActivity <- merge(Heartrate_daily, Activity, by= c("Id", "Date")) # using awake sede
ntary time rather than sedentary time (including sleeping time) as Sleep naturally lowers hea
rt rate (due to parasympathetic activity)

cor.test(HeartrateActivity$AvgHeartRate, HeartrateActivity$SedentaryAwakeTime)
```

```
##
##  Pearson's product-moment correlation
##
## data:  HeartrateActivity$AvgHeartRate and HeartrateActivity$SedentaryAwakeTime
## t = 2.5124, df = 156, p-value = 0.01301
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.0423680 0.3427891
## sample estimates:
##       cor
## 0.1972037
```

```
# r(156)= 0.20, p<.05 (Weak positive correlation, statistically significant)

ggplot(HeartrateActivity, aes(x = SedentaryAwakeTime, y = AvgHeartRate)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Avg Heart Rate vs Sedentary Awake Time",
       x = "Sedentary Awake Time (mins)",
       y = "Average Heart Rate") +
  theme_minimal()
```
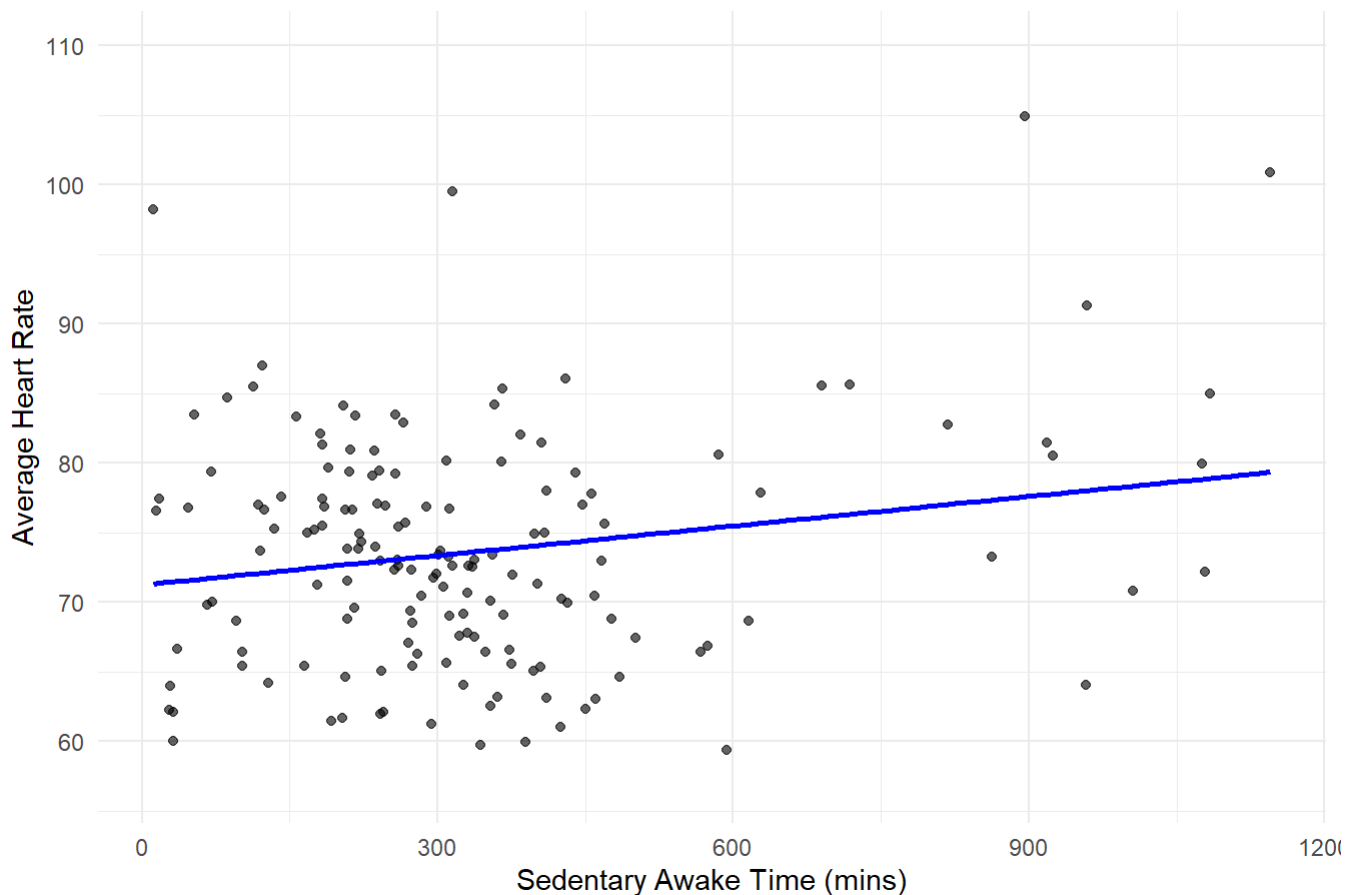
```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 311 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 311 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

Avg Heart Rate vs Sedentary Awake Time

A slight positive correlation was observed between sedentary time and average heart rate, suggesting that individuals who are more sedentary may tend to have higher resting heart rates.

4- Total Active Minutes and Steps

```
cor.test(Activity$TotalActiveMinutes, Activity$TotalSteps)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Activity$TotalActiveMinutes and Activity$TotalSteps
## t = 44.682, df = 1371, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7475305 0.7906786
## sample estimates:
##       cor
## 0.7699834
```

```
# r(1371)= 0.77, p<.001 (Very strong positive correlation, very significant)
```

```
ggplot(Activity, aes(x= TotalActiveMinutes, y= Activity$TotalSteps))+
  geom_point(alpha= 0.6)+
  geom_smooth(method = "lm", se= FALSE, Color="blue")+
  labs(title= "Total Active Time vs Steps",
       x= "Total Daily Active Time (min)",
       y= "Daily Steps")+
  theme_minimal()
```

```
## Warning in geom_smooth(method = "lm", se = FALSE, Color = "blue"): Ignoring
## unknown parameters: `Color`
```

```
## Warning: Use of `Activity$TotalSteps` is discouraged.
## i Use `TotalSteps` instead.
## Use of `Activity$TotalSteps` is discouraged.
## i Use `TotalSteps` instead.
```

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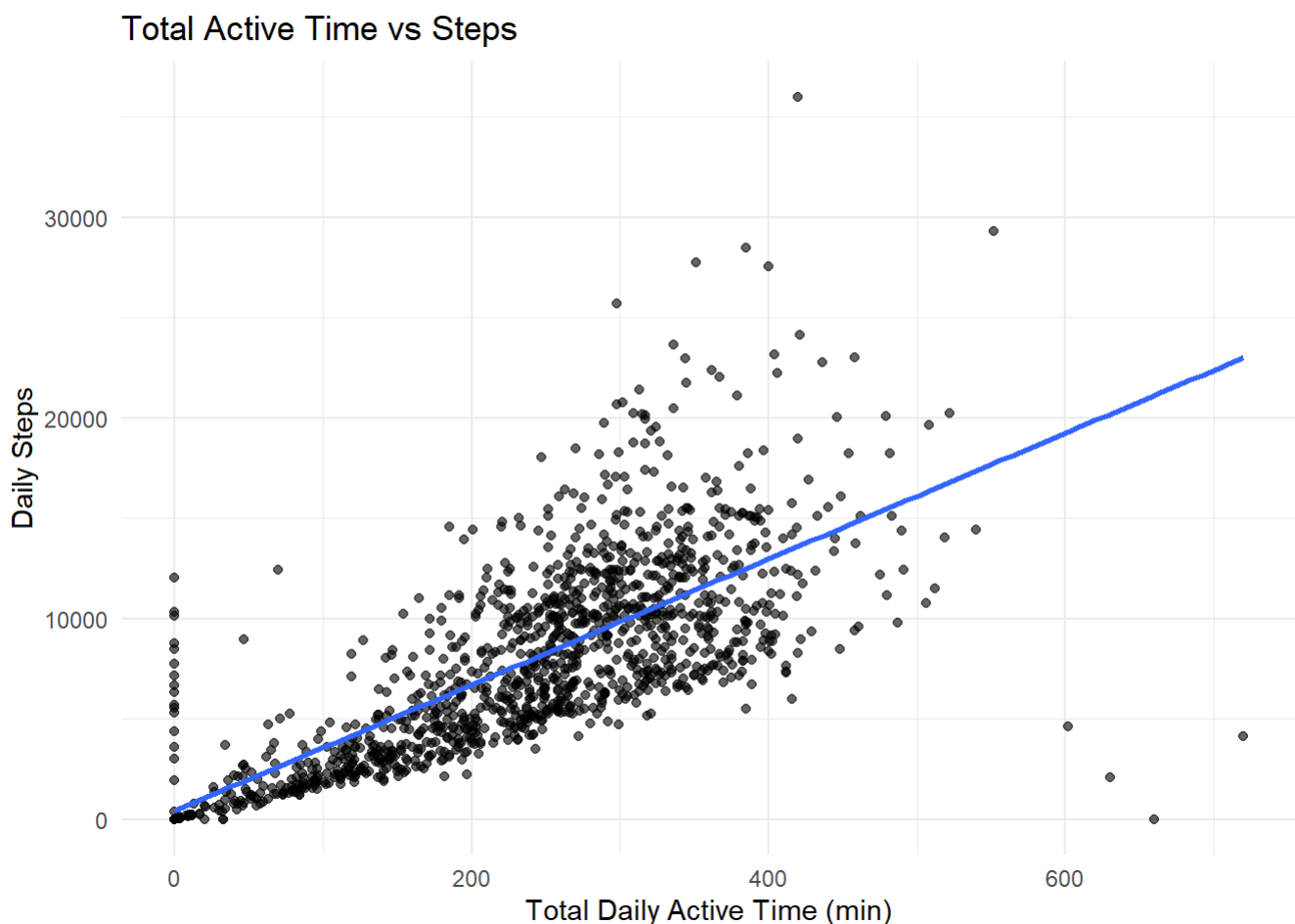

The correlation between Total Active Minutes and Total Steps is very strong (r = 0.77, p < .001), indicating a significant and positive linear relationship. This suggests that individuals who spend more time being active tend to take more steps throughout the day. The strong correlation also implies that a large portion of daily physical activity is likely made up of walking or stepping-based movement, reinforcing the role of step count as a key component of overall activity levels.

5- Calories and Very Active minutes

```
cor.test(Activity$Calories, Activity$VeryActiveMinutes)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Activity$Calories and Activity$VeryActiveMinutes
## t = 27.328, df = 1371, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5584704 0.6270336
## sample estimates:
##       cor
## 0.593829
```

```
# r(1371) =0.59p<.001 (positive correlation, Very significant)

ggplot(Activity, aes(x=Calories, y=VeryActiveMinutes))+
  geom_point(alpha=0.6)+ #60% opaque
  geom_smooth(method = "lm", se= FALSE, Color="blue")+ # linear model, False = hides shaded c
onfidence interval
  labs(title= "Calories vs Very Active Minutes",
      x= "Calories",
      y="Very Active Minutes")+
  theme_minimal()
```

```
## Warning in geom_smooth(method = "lm", se = FALSE, Color = "blue"): Ignoring
## unknown parameters: `Color`
```

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

Calories vs Very Active Minutes
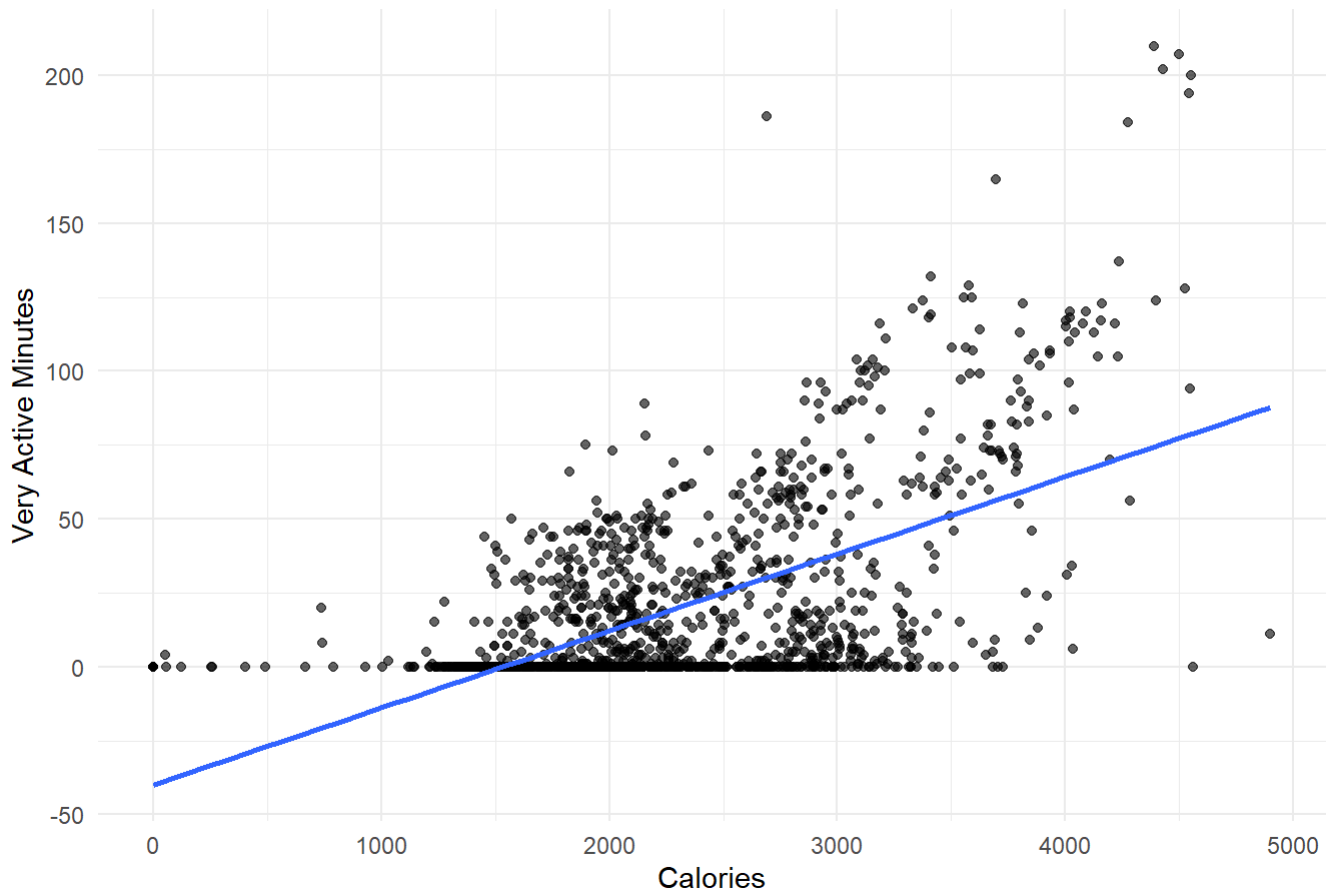
There is a clear positive correlation between Very Active Minutes and Calories Burned, indicating that as the amount of very active time increases, the number of calories burned also tends to rise.

6- Calories and Steps

```
cor.test(Activity$Calories, Activity$TotalSteps)
```
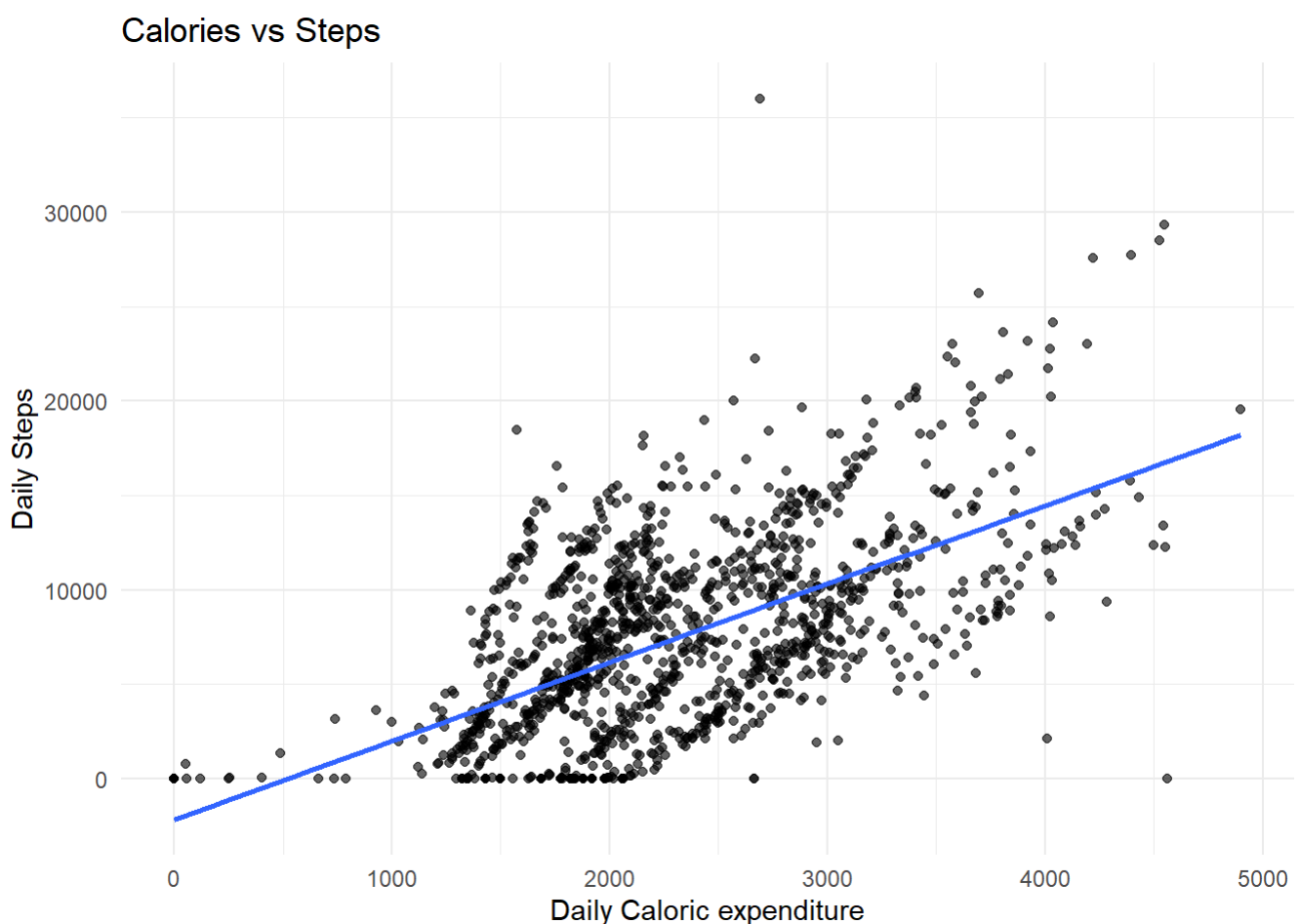
```
##
##  Pearson's product-moment correlation
##
## data:  Activity$Calories and Activity$TotalSteps
## t = 26.368, df = 1371, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5438633 0.6141333
## sample estimates:
##       cor
## 0.5800765
```

```
# r(1371)=0.58, p<.001 (positive correlation, very significant)

ggplot(Activity, aes(x=Calories, y=TotalSteps))+
  geom_point(alpha=0.6)+
  geom_smooth(method = "lm", se= FALSE, Color="blue")+
  labs(title = "Calories vs Steps",
       x="Daily Caloric expenditure",
       y="Daily Steps")+
  theme_minimal()
```

```
## Warning in geom_smooth(method = "lm", se = FALSE, Color = "blue"): Ignoring
## unknown parameters: `Color`
```

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


There is a positive relationship between daily step count and calorie burnt, indicating that individuals who walk more steps tend to expend more calories throughout the day.

7- Time in Bed (Awake) and Fairly Active minutes

```
cor.test(Allmerged$TimeInBedAwake, Allmerged$FairlyActiveMinutes)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Allmerged$TimeInBedAwake and Allmerged$FairlyActiveMinutes
## t = 6.8117, df = 412, p-value = 3.43e-11
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2287887 0.4021966
## sample estimates:
##       cor
## 0.3181512
```

```
# r(412)=0.32, p<.001 (weak positive correlation, very significant)


ggplot(Allmerged, aes(x=TimeInBedAwake, y=FairlyActiveMinutes))+
  geom_point(alpha=0.6)+
  geom_smooth(method="lm", se=FALSE, Color="blue")+
  labs(title="Time in Bed Awake vs Daily Fairly Active Minutes",
       x="Time in Bed Awake (min)",
       y="Daily Fairly Active Minutes")+
  coord_cartesian(xlim = c(0, 200), ylim = c(0, 200)) +  # zoom to focus area without droppin
g data
  theme_minimal()
```
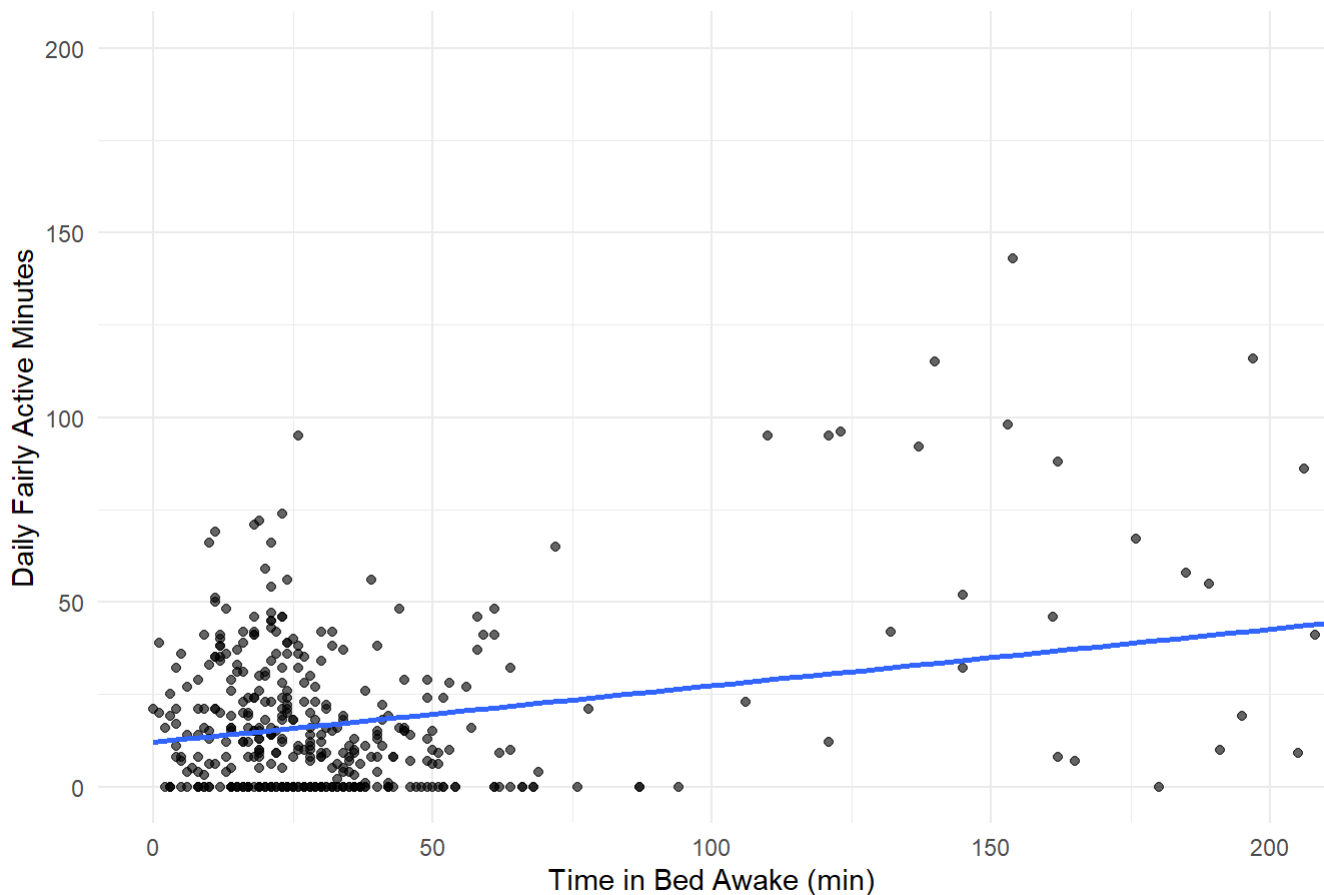
```
## Warning in geom_smooth(method = "lm", se = FALSE, Color = "blue"): Ignoring
## unknown parameters: `Color`
```

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

```
## Warning: Removed 964 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 964 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

Time in Bed Awake vs Daily Fairly Active Minutes

The more time Daily fairly active time is spent the more time in bed is spent before and after sleeping and vice versa. Because correlation does not imply causation, this association highlights a connection but doesn't clarify the direction of influence. Those results could indicate that spending time in bed helps relaxation and increases motivation for the rest of the day resulting in higher fairly active time but it could also indicate that longer fairly active time leads to the need of more time to relax once in bed.

Summary of Analysis

Key findings from the data analysis include:

Activity Trends

The average daily step count was 7,281—well below the recommended 10,000. Users were more active on Saturdays and least active on Fridays and Sundays. Weekends showed slightly more active minutes and lower sedentary behavior.

Sleep Patterns

Users sleep longer on weekends, especially on Sundays. A significant portion of time in bed is spent awake, particularly on weeknights.

Heart Rate Insights

Slightly elevated average heart rate on weekends. Weak negative correlation between heart rate and sleep duration.

Weight and BMI

Slight increase in weight and BMI over weekends, peaking on Sundays and Wednesdays. Correlation Highlights: Strong positive correlation between total active minutes and step count (r = 0.77). Moderate positive correlation between very active minutes and calories burned. Strong negative correlation between sedentary awake time and total sleep (r = -0.89). Slight positive correlation between sedentary awake time and heart rate.

Key Insight Summaries

Less activity correlates with reduced sleep and slightly higher heart rate. Most users do not meet daily activity guidelines. User behaviors differ significantly between weekdays and weekends, affecting sleep, heart rate, and calorie burn.

Based on the analysis, the following strategic recommendations are proposed for Bellabeat:

1. Introduce Personalized Activity Nudges

Insight: Most users do not meet the 10,000-step daily recommendation; activity is lowest on weekdays,especially Fridays. Action: Use app notifications to encourage short bursts of movement during sedentary periods, especially during workdays. Example: "Take 1,000 steps by 3 PM to hit your daily goal!"

2. Emphasize Weekend Wellness Features

Insight: Users sleep more and are slightly more active on weekends. Action: Launch weekend wellness campaigns featuring mindfulness, recovery, hydration tracking, and sleep rituals. "Recharge Sundays":Guided breathing + hydration reminders + sleep prep content.

3. Create Smart Sleep Coaching

Insight: There's a strong negative correlation between sedentary time and sleep; many users spend time in bed awake. Action: Offer in-app sleep coaching to help users improve sleep efficiency (not just duration), including: Pre-bed routines: Light/stretch reminders before sleep. Content to reduce "awake in bed" time (e.g., sleep stories, relaxation sounds)

4. Launch Heart Health Awareness Tools

Insight: Higher sedentary time is weakly correlated with higher heart rate. Action: Introduce heart rate monitoring insights that detect patterns and offer tips (e.g., breathing exercises, movement alerts)."We noticed your heart rate is elevated—want to take a 3-minute breathing break?"

5. Add Adaptive Daily Goals

Insight: Activity and sleep levels vary by day of the week. Action: Use machine learning to adapt daily step or sleep goals based on user trends (e.g., higher weekend activity). Adjusted expectations may improve user satisfaction and consistency.

6. Implement "Lifestyle Snapshot" Dashboards

Insight: Correlation analysis showed meaningful connections between activity, heart rate, sleep, and calories. Action: Provide users with a weekly health summary combining: Steps, Heart rate trends, Sleep efficiency, Calories burned. Example: "This week you were most active on Saturday. Sleep improved on days with less

sedentary time."

7. Gamify Movement & Sleep

Insight: Users fall short on physical activity and may benefit from motivation. Action: Add achievements, badges, and streaks for: Meeting step goals, Improving sleep efficiency, Lowering resting heart rate over time.

Area for future research

1. Audience Segmentation

Further research could explore how different age groups of female users engage with wearable technology. Specifically: Do younger women (ages 20–35) prioritize fitness tracking, step goals, and the visual/aesthetic appeal of devices? Do older women (ages 35+) place more value on features that support wellness, heart health, sleep monitoring, and stress management?

2. Competitive Differentiation

Further exploration could assess whether Bellabeat can create a stronger market position by focusing on holistic wellness rather than traditional fitness tracking alone. Specifically: Do users respond more positively to devices that support both physical and mental wellbeing, such as stress reduction, sleep quality, mindfulness, and emotional balance? Would framing Bellabeat as a lifestyle partner—rather than a step counter—better align with the needs and expectations of its coreaudience?

3. Female-Specific Health Features

Further development could explore how Bellabeat can better support women's unique health needs by integrating features tailored to different life stages. Specifically: Do users benefit from tracking tools related to menstrual cycles, fertility windows, and hormonal fluctuations? Could women entering perimenopause and menopause find value in tools focused on stress management, sleep support, and holistic wellness guidance?

Understanding these preferences could help Bellabeat tailor its product features and marketing messages to better meet the distinct needs of these segments.