

This is a Capstone Project for my recently completed Google Data Analytics Professional Certificate. The goal of the project is to apply the six phases of data analysis using real-world data and some of the tools used in the course.

Scenario

I'm working for Bellabeat, a manufacturer of high-tech health and wellness products for women. As a junior analyst on the marketing team, we have been tasked with analyzing data collected from smart devices to gain insight into how consumers use their devices and make marketing recommendations based on those insights.

Let's go through the phases of the analysis process:

Phase 1: Ask

Our task:

Analyze FitBit data to identify trends and make recommendations for marketing Bellabeat products.

Who are our stakeholders?

- Urška Sršen - Bellabeat cofounder and Chief Creative Officer
- Sando Mur - Bellabeat cofounder and key member of Bellabeat executive team

It may be helpful to bear their backgrounds in mind. Sršen is an artist and Mur is a mathematician. This may mean that Sršen will appreciate creativity and Mur will appreciate data-driven ideas. Hopefully, our analysis will allow us to be both creative and data-driven.

The following questions will guide our analysis:

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

We will focus on trends related to physical activity and sleep as these are among the most relevant to women's health issues, as noted in research by Segar et al. [here](#) and Nowakowski et al. [here](#).

Phase 2: Prepare

Our task:

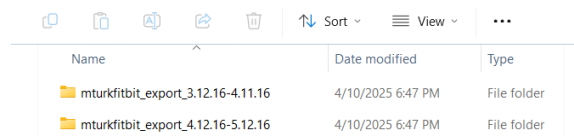
Gather and organize our data for analysis.

Sršen has encouraged us to use “FitBit Fitness Tracker Data” which is available on Kaggle through Mobius. The data is licensed [CC0: Public Domain](#). These datasets were generated by respondents to a distributed survey via Amazon Mechanical Turk between 03.12.2016-05.12.2016.

This data set contains personal fitness tracker from thirty FitBit users. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring.

Data Organization:

The data is organized into 29 CSV documents. We'll begin by opening the documents in Excel to get a better sense of the data organization so we can decide how to proceed. There are multiple folders and CSV files for differing dates and users. We notice that each folder has similar files for different date ranges. This may be due to file limitations or the way the data was collected. After counting the unique user IDs and date ranges, we decide to focus on the files relating the 31 days between 4.12.16-5.12.16 since it has the most robust sleep and activity data and we want to avoid inconsistencies.



Name	Date modified	Type
mturkfitbit_export_3.12.16-4.11.16	4/10/2025 6:47 PM	File folder
mturkfitbit_export_4.12.16-5.12.16	4/10/2025 6:47 PM	File folder

Each CSV file represents different data tracked by Fitbit for the different Fitbit users.

- dailyActivity_merged
- heartrate_seconds_merged
- hourlyCalories_merged
- hourlyIntensities_merged
- hourlySteps_merged
- minuteCaloriesNarrow_merged
- minuteIntensitiesNarrow_merged
- minuteMETsNarrow_merged
- minuteSleep_merged
- minuteStepsNarrow_merged
- weightLogInfo_merged

Opening the files, we see that we are working with “long data” since each subject has many rows of data with each row representing a different date-time entry.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
2	1503960366	5/2/2016	14727	9.71	9.71	0	3.21	0.57	5.92	0	41	15	277	798	2004
3	1503960366	5/3/2016	15103	9.66	9.66	0	3.73	1.05	4.88	0	50	24	254	816	1990
4	1503960366	5/4/2016	11100	7.15	7.15	0	2.46	0.87	3.82	0	36	22	203	1179	1819
5	1503960366	5/5/2016	14070	8.9	8.9	0	2.92	1.08	4.88	0	45	24	250	857	1959
6	1503960366	5/6/2016	12159	8.03	8.03	0	1.97	0.25	5.81	0	24	6	289	754	1896
7	1503960366	5/7/2016	11992	7.71	7.71	0	2.46	2.12	3.13	0	37	46	175	833	1821
8	1503960366	5/8/2016	10060	6.58	6.58	0	3.53	0.32	2.73	0	44	8	203	574	1740
9	1503960366	5/9/2016	12022	7.72	7.72	0	3.45	0.53	3.74	0	46	11	206	835	1819
0	1503960366	5/10/2016	12207	7.77	7.77	0	3.35	1.16	3.26	0	46	31	214	746	1859
1	1503960366	5/11/2016	12770	8.13	8.13	0	2.56	1.01	4.55	0	36	23	251	669	1783
2	1503960366	5/12/2016	0	0	0	0	0	0	0	0	0	0	0	1440	0
3	1624580081	4/12/2016	8163	5.31	5.31	0	0	0	5.31	0	0	0	146	1294	1432
4	1624580081	4/13/2016	7007	4.55	4.55	0	0	0	4.55	0	0	0	148	1292	1411
5	1624580081	4/14/2016	9107	5.92	5.92	0	0	0	5.91	0.01	0	0	236	1204	1572
6	1624580081	4/15/2016	1510	0.98	0.98	0	0	0	0.97	0	0	0	96	1344	1344
7	1624580081	4/16/2016	5370	3.49	3.49	0	0	0	3.49	0	0	0	176	1264	1463
8	1624580081	4/17/2016	6175	4.06	4.06	0	1.03	1.52	1.49	0.01	15	22	127	1276	1554

Note the data limitations:

- Sample size: 33 people is not a representative sample of FitBit or Bellabeat users
- Time: We are looking at data from 31 days in 2016. It would be more informative to see data from the current year and spanning a longer period.
- Lacking personal info: More demographic information about the users would tell us more about whether the data could be biased for age, gender, or location. Adding this information would also be beneficial for marketing and targeting purposes.

Let’s bring our files into RStudio to prepare the data for further work.

Prepare: Step 1- Prepare RStudio

```

type 'q()' to quit R.

> setwd("D:/Bellabeat")
> library(tidyverse)
— Attaching core tidyverse packages — tidyverse 2.0.0 —
✓ dplyr 1.1.4 ✓ readr 2.1.5
✓ forcats 1.0.0 ✓ stringr 1.5.1
✓ ggplot2 3.5.1 ✓ tibble 3.2.1
✓ lubridate 1.9.4 ✓ tidyr 1.3.1
✓ purrr 1.0.4
— Conflicts — tidyverse_conflicts() —
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag() masks stats::lag()
! Use the conflicted package to force all conflicts to become errors
> library(lubridate)
> |

```

Prepare: Step 2- Import datasets

- Daily activity

```

> daily_activity <- read_csv("Fitabase Data 4.12.16-5.12.16/dailyActivity_merged.csv")
Rows: 940 Columns: 15
— Column specification —
Delimiter: ","
chr (1): ActivityDate
dbl (14): Id, TotalSteps, TotalDistance, TrackerDistance, LoggedActivitiesDistance, VeryAct..

```

- Sleep

```

> sleep_day <- read_csv("Fitabase Data 4.12.16-5.12.16/sleepDay_merged.csv")
Rows: 413 Columns: 5
— Column specification —
Delimiter: ","
chr (1): SleepDay
dbl (4): Id, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed

```

- Daily Intensity

```

> daily_intensity <- read_csv("Fitabase Data 4.12.16-5.12.16/dailyIntensities_merged.csv")
Rows: 940 Columns: 10
— Column specification —
Delimiter: ","
chr (1): ActivityDay
dbl (9): Id, SedentaryMinutes, LightlyActiveMinutes, FairlyActiveMinutes, VeryActiveMinutes...

```

- Heartrate

```

> heartrate_seconds <- read_csv("Fitabase Data 4.12.16-5.12.16/hearttrate_seconds_merged.csv")
Rows: 2483658 Columns: 3
— Column specification —
Delimiter: ","
chr (1): Time
dbl (2): Id, Value

```

Prepare: Step 3- Preview to confirm successful import

- daily_activity

```

> head(daily_activity, 3)
# A tibble: 3 × 15
  Id ActivityDate TotalSteps TotalDistance TrackerDistance LoggedActivitiesDistance
  <dbl> <chr> <dbl> <dbl> <dbl> <dbl>
1 1503960366 4/12/2016 13162 8.5 8.5 0
2 1503960366 4/13/2016 10735 6.97 6.97 0
3 1503960366 4/14/2016 10460 6.74 6.74 0

```

- sleep_day

```

> head(sleep_day, 3)
# A tibble: 3 × 5
  Id SleepDay TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
  <dbl> <chr> <dbl> <dbl> <dbl>
1 1503960366 4/12/2016 12:00:00 AM 1 327 346
2 1503960366 4/13/2016 12:00:00 AM 2 384 407
3 1503960366 4/15/2016 12:00:00 AM 1 412 442

```

- daily_intensity

```

> head(daily_intensity, 3)
# A tibble: 3 × 10
  Id ActivityDay SedentaryMinutes LightlyActiveMinutes FairlyActiveMinutes
  <dbl> <chr> <dbl> <dbl> <dbl>
1 1503960366 4/12/2016 728 328 13
2 1503960366 4/13/2016 776 217 19
3 1503960366 4/14/2016 1218 181 11

```

- heartrate_seconds

```
> head(heartrate_seconds, 3)
# A tibble: 3 x 3
      Id Time                               Value
  <dbl> <chr>                               <dbl>
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
```

Prepare: Step 4- Check quality- users, dates, rows

- Unique users

```
> n_distinct(daily_activity$Id)
[1] 33
> n_distinct(sleep_day$Id)
[1] 24
> n_distinct(daily_intensity$Id)
[1] 33
> n_distinct(heartrate_seconds$Id)
[1] 14
```

- Date Ranges

```
> #date ranges
> range(daily_activity$ActivityDate)
[1] "4/12/2016" "5/9/2016"
> range(sleep_day$SleepDay)
[1] "4/12/2016 12:00:00 AM" "5/9/2016 12:00:00 AM"
> range(daily_intensity$ActivityDay)
[1] "4/12/2016" "5/9/2016"
> range(heartrate_seconds$Time)
[1] "4/12/2016 1:00:00 AM" "5/9/2016 9:59:59 PM"
```

- Row count

```
> # Row counts
> nrow(daily_activity)
[1] 940
> nrow(sleep_day)
[1] 413
> nrow(daily_intensity)
[1] 940
> nrow(heartrate_seconds)
[1] 2483658
```

We have confirmed that there are 33 users for the daily activity and intensity datasets and fewer for the sleep and heartrate datasets.

We have confirmed that all the datasets span the same 31 days from April 12 to May 12 of 2016.

We have confirmed that the number of rows is consistent with data from 33 users over a 31 day period.

Prepare: Step 5- Prepare and save files for next phase

```
> write_csv(daily_activity, "daily_activity_prepared.csv")
> write_csv(sleep_day, "sleep_day_prepared.csv")

> write_csv(daily_intensity, "daily_intensity_prepared.csv")
> write_csv(heartrate_seconds, "heartrate_seconds_prepared.csv")
```

Phase 3: Process

Our task:

Clean and transform data to ensure consistency and usability for analysis.

Process Phase: Step 1- Load data and make sure it is all correct for this phase

We'll begin by loading the prepared and saved datasets from the last phase and using `head()` to preview and confirm it's correct.

```
> # Preview data to confirm loading
> head(daily_activity, 3)
# A tibble: 3 x 15
  Id ActivityDate TotalSteps TotalDistance TrackerDistance LoggedActivitiesDistance
  <dbl> <chr>      <dbl>      <dbl>      <dbl>      <dbl>
1 1503960366 4/12/2016      13162      8.5        8.5        0
2 1503960366 4/13/2016      10735      6.97       6.97       0
3 1503960366 4/14/2016      10460      6.74       6.74       0

> head(sleep_day, 3)
# A tibble: 3 x 5
  Id SleepDay      TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
  <dbl> <chr>      <dbl>      <dbl>      <dbl>
1 1503960366 4/12/2016 12:00:00 AM      1          327          346
2 1503960366 4/13/2016 12:00:00 AM      2          384          407
3 1503960366 4/15/2016 12:00:00 AM      1          412          442

> head(daily_intensity, 3)
# A tibble: 3 x 10
  Id ActivityDay SedentaryMinutes LightlyActiveMinutes FairlyActiveMinutes VeryActiveMinutes
  <dbl> <chr>      <dbl>      <dbl>      <dbl>      <dbl>
1 1.50e9 4/12/2016      728        328        13         25
2 1.50e9 4/13/2016      776        217        19         21
3 1.50e9 4/14/2016     1218        181        11         30

> head(heartrate_seconds, 3)
# A tibble: 3 x 3
  Id Time      Value
  <dbl> <chr>      <dbl>
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
```

Process Phase: Step 2- Standardize the date formats

In order to merge and compare the datasets together we need all the date types to be uniform. We notice that `daily_activity` and `daily_intensity` are in MM/DD/YYYY format, which need to be changed to R's YYYY-MM-DD format. The `sleep_day` dataset has date formats that need to be changed to YYYY-MM-DD and timestamps that should be removed. The `heartrate_seconds` dataset has timestamps that are to the level of seconds; we will combine these to daily averages so we can look at this dataset along with the others.

- Let's start with `daily_activity`:

```
25 # Daily activity: Convert ActivityDate from MM/DD/YYYY
26 daily_activity <- daily_activity %>%
27   mutate(ActivityDate = mdy(ActivityDate)) %>%
28   rename(Date = ActivityDate)
```

Let's check that it worked:

```
> head(daily_activity, 3)
# A tibble: 3 x 5
  Id Date       TotalSteps TotalDistance TrackerDistance LoggedActivitiesDistance
  <dbl> <date>     <dbl>         <dbl>         <dbl>         <dbl>
1 1503960366 2016-04-12 13162         8.5           8.5           0
2 1503960366 2016-04-13 10735         6.97          6.97          0
3 1503960366 2016-04-14 10460         6.74          6.74          0
```

Looks good!

- For Sleep, we will need an extra line of code to remove the timestamp:

```
# Sleep: Extract date from SleepDay timestamp (MM/DD/YYYY HH:MM:SS AM/PM)
sleep_day <- sleep_day %>%
  mutate(SleepDay = mdy_hms(SleepDay)) %>%
  mutate(Date = as.Date(SleepDay)) %>%
  select(-SleepDay) #Remove timestamp column
```

Check it:

```
> head(sleep_day, 3)
# A tibble: 3 x 5
  Id TotalSleepRecords TotalMinutesAsleep TotalTimeInBed Date
  <dbl> <dbl>         <dbl>         <dbl> <date>
1 1503960366 1 327 346 2016-04-12
2 1503960366 2 384 407 2016-04-13
3 1503960366 1 412 442 2016-04-15
```

The date is in the right format and the timestamp is gone.

- We changed *daily_intensity* was transformed in the same way as the activity dataset. I'll just show the preview with the standardized date format here.

```
> head(daily_intensity, 3)
# A tibble: 3 x 10
  Id Date       SedentaryMinutes LightlyActiveMinutes FairlyActiveMinutes VeryActiveMinutes
  <dbl> <date>     <dbl>         <dbl>         <dbl>         <dbl>
1 1.50e9 2016-04-12 728 328 13 25
2 1.50e9 2016-04-13 776 217 19 21
3 1.50e9 2016-04-14 1218 181 11 30
```

- The heartrate data was recorded the level of seconds. We created a new dataset for daily heartrate data. Then standardized the date formats without timestamps. The last part of the code aggregates the per-second data into daily metrics of avg, max, and min.

```
# Heart rate: Convert date format and turn seconds-level data into daily averages
heartrate_daily <- heartrate_seconds %>% #new daily heartrate dataset to match others
  mutate(Time = mdy_hms(Time)) %>%
  mutate(Date = as.Date(Time)) %>%
  group_by(Id, Date) %>%
  summarise(AvgHeartRate = mean(Value, na.rm = TRUE),
            MinHeartRate = min(Value, na.rm = TRUE),
            MaxHeartRate = max(Value, na.rm = TRUE)) %>% #aggregate seconds into daily metric
  ungroup()
```

Preview to make sure it worked as expected:

```
> head(heartrate_daily, 3)
# A tibble: 3 x 5
  Id Date       AvgHeartRate MinHeartRate MaxHeartRate
  <dbl> <date>     <dbl>         <dbl>         <dbl>
1 2022484408 2016-04-12 75.8 52 134
2 2022484408 2016-04-13 80.3 51 156
3 2022484408 2016-04-14 72.6 50 127
```

We've turned close to 2.5 million rows of heartrate data into 334 rows of daily metrics.

Summary of Process Step 2: We standardized the date columns across all four datasets to a YYYY-MM-DD format to ensure accurate joining and analysis. For the heartrate data, we also aggregated the seconds-level records into daily metrics. The previews confirm that we were successful.

Process Phase: Step 3- Check for and remove duplicates

Before moving on to our analysis, let's make sure none of our datasets have duplicate rows. Duplicates have the potential to inflate or skew our metrics, so checking always is an important practice to ensure data quality.

```
8 # Step 3: Check and remove duplicates
9 # Daily activity
0 daily_activity %>% duplicated() %>% sum() #display count of duplicates
1 daily_activity <- daily_activity %>% distinct() #remove any duplicates
2
3 # Sleep
4 sleep_day %>% duplicated() %>% sum()
5 sleep_day <- sleep_day %>% distinct()
6
7 # Daily intensity
8 daily_intensity %>% duplicated() %>% sum()
9 daily_intensity <- daily_intensity %>% distinct()
0
1 # Heart rate (daily)
2 heartrate_daily %>% duplicated() %>% sum()
3 heartrate_daily <- heartrate_daily %>% distinct()
```

Looking at the console outputs, we can see if any of our datasets had duplicates that were removed:

```
> # Daily activity
> daily_activity %>% duplicated() %>% sum()
[1] 0

> # Sleep
> sleep_day %>% duplicated() %>% sum()
[1] 3

> # Daily intensity
> daily_intensity %>% duplicated() %>% sum()
[1] 0

> # Heart rate (daily)
> heartrate_daily %>% duplicated() %>% sum()
[1] 0
```

Notice that the *sleep_day* dataset had three duplicate entries.

Process Phase: Step 4- Check for missing values

The next step will be to check for missing values or incomplete data. This is important because missing data may be for a reason that is relevant to our analysis. If there are missing values, we will need to decide how to handle them before merging our datasets.

We can do this by using the `is.na` function to check for blanks in each column and the `colSums` function to count how many were found:

```
6 # Step 4: Check for missing values
7 colSums(is.na(daily_activity)) # Check for NA
8 colSums(is.na(sleep_day))
9 colSums(is.na(daily_intensity))
0 colSums(is.na(heartrate_daily))
```


Look at the console outputs to see if missing values were found:

```
> colSums(is.na(sleep_day))
      Id TotalSleepRecords TotalMinutesAsleep TotalTimeInBed      Date 
      0              0              0              0              0
```

This shows the output for the *sleep_day* dataset. The other three datasets also showed zero NA entries.

No missing values were found within any of the datasets' columns. However, since the *sleep_day* (24 users, 410 rows) and *heartrate_daily* (24 users, 334 rows) datasets have fewer users and rows, we can expect to see NA entries after we merge the datasets.

While 73% of users (24/33) tracked sleep and heart rate at least once, only ~44% and ~42% of daily records include sleep and heart rate data, respectively, indicating inconsistent usage.

Process Phase: Step 5- Merge the datasets

In this phase we will combine the four datasets into one set using Id and Date to link them. We will start with the *daily_activity* dataset as our foundation since it has the most complete data. We will then join the Sleep data to it, followed by the Intensity data and then the Heartrate Data.

Since not all users and days have Sleep or Heartrate data, using *left_join()* from *dplyr* will let that data be added only where records exist. Expect NA where there is no sleep or heartrate data.

```
# Step 5: Merge datasets by Id and Date
# Use left_join to keep all daily activity rows
merged_data <- daily_activity %>%
  left_join(sleep_day, by = c("Id", "Date")) %>%
  left_join(daily_intensity, by = c("Id", "Date")) %>%
  left_join(heartrate_daily, by = c("Id", "Date"))
```

We'll also add a column for day of the week. It might be interesting to note if are trends specific to days of the week or comparing weekdays to weekends.

```
mutate(DayOfWeek = weekdays(Date)) #Add day of week column
```

Let's preview to see that it worked. In the preview we expect to see that there are columns from all our datasets together with NA in the sleep and heartrate columns where there is data missing.

	Id	Date	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistance	VeryActiveDistance.x	ModerateDistance
1	1503960366	2016-04-12	13162	8.50	8.50	0	1.88	
2	1503960366	2016-04-13	10735	6.97	6.97	0	1.57	
3	1503960366	2016-04-14	10460	6.74	6.74	0	2.44	
4	1503960366	2016-04-15	9762	6.28	6.28	0	2.14	
5	1503960366	2016-04-16	12669	8.16	8.16	0	2.71	

Our *merged_data* dataset now combines all the key metrics from the different datasets with an added DayOfWeek column to study weekly trends. The missing data from the sleep and

heartrate columns show real-world gaps which will be important to note as we go through our analysis.

Process Phase: Step 6- Validate merged data

Our new set of *merged_data* looks good as we preview it, but let's make sure that nothing got messed up in the merge. We will check the number of rows, unique users, and dates.

```
# Step 6: Validate merged data
nrow(merged_data) # expect 940
n_distinct(merged_data$Id) # expect 33 users
range(merged_data$Date) # expect 2016-04-12 to 2016-05-12
```

Check our results in the console:

```
> nrow(merged_data) # expect 940
[1] 940
> n_distinct(merged_data$Id) # expect 33 users
[1] 33
> range(merged_data$Date) # expect 2016-04-12 to 2016-05-12
[1] "2016-04-12" "2016-05-12"
```

Our row count, user count, and date range all match expectations. Our data validation confirms a successful merge of the datasets. Our dataset can be saved and used for analysis.

Process Phase: Step 7- Save the cleaned dataset

Let's save our cleaned, merged, and validated dataset to complete the Process phase and make sure all our steps are preserved for the analysis.

```
# Step 7: Save cleaned dataset
write_csv(merged_data, "merged_data_cleaned.csv")
```

Phase 4: Analyze

Our task:

Explore relationships and trends in the data.

Analyze Phase: Step 1- Building R Markdown Report

For the analyze phase, we decided to create an R Markdown report ('analyze_data.Rmd') to analyze 'merged_data_cleaned.csv'. We did a little internet research and figured out how to

create a .css file to reference so we can use Bellabeat theme colors for the headers in our report.

We added steps to the report for loading data, summarizing metrics, and visualizing steps vs. sleep.

Each step used headers, explanatory text, and code chunks.

Analyze Phase: Step 2- Summarize key metrics

Using the *mean* function we created a table of summary statistics for the key measures from our merged data.

```
31- ## Step 2: Summarize Key Metrics
32- To understand typical behavior, we calculate average steps, sleep hours, calories and heart
33- rate as well as the percentage of records with sleep and heart rate information.
34- 
35- ```{r summary-stats}
36- summary_stats <- merged_data %>%
37-   summarise(
38-     AvgSteps = mean(TotalSteps, na.rm = TRUE),
39-     AvgSleepHours = mean(TotalMinutesAsleep / 60, na.rm = TRUE),
40-     AvgCalories = mean(Calories, na.rm = TRUE),
41-     AvgHeartRate = mean(AvgHeartRate, na.rm = TRUE),
42-     PctSleepLogged = mean(!is.na(TotalMinutesAsleep)) * 100,
43-     PctHeartRateLogged = mean(!is.na(AvgHeartRate)) * 100
44-   )
45- summary_stats
46- ```
```

- R Markdown section for Step 2, with header, text, and code chunk

Step 2: Summarize Key Metrics

To understand typical behavior, we calculate average steps, sleep hours, calories and heart rate as well as the percentage of records with sleep and heart rate information.

```
summary_stats <- merged_data %>%
  summarise(
    AvgSteps = mean(TotalSteps, na.rm = TRUE),
    AvgSleepHours = mean(TotalMinutesAsleep / 60, na.rm = TRUE),
    AvgCalories = mean(Calories, na.rm = TRUE),
    AvgHeartRate = mean(AvgHeartRate, na.rm = TRUE),
    PctSleepLogged = mean(!is.na(TotalMinutesAsleep)) * 100,
    PctHeartRateLogged = mean(!is.na(AvgHeartRate)) * 100
  )
summary_stats
```

```
## # A tibble: 1 x 6
##   AvgSteps AvgSleepHours AvgCalories AvgHeartRate PctSleepLogged
##   <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
## 1  7638.         6.99         2304.         78.6         43.6
## #> 1 more variable: PctHeartRateLogged <dbl>
```

- HTML report with header and summary statistics table

Analyze Phase: Step 3- Visualize Steps vs. Sleep

When we merged our data, we noted that fewer users were using sleep and heart rate tracking. Exploring the relationships between these metrics and more popular ones may offer insight into how they could be marketed better to users.

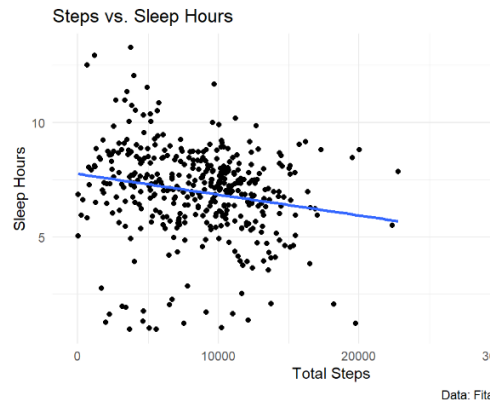
We'll begin by looking at the relationship between steps and sleep. To explore if there is a relationship, we plotted daily steps against sleep hours using a scatterplot with a trendline.

Step 3: Visualize Steps vs. Sleep

To explore whether activity affects sleep, we'll create a scatter plot of daily steps against sleep hours. A strong correlation or lack of correlation could suggest whether these features should be marketed together or separately.

```
## [r steps-vs-sleep, fig.width=6, fig.height=4]
#Visualize steps vs sleep
ggplot(data = merged_data, aes(x = TotalSteps, y = TotalMinutesAsleep / 60)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Steps vs. Sleep Hours",
       x = "Total Steps",
       y = "Sleep Hours",
       caption = "Data: Fitabase 4.12.16-5.12.16") +
  theme_minimal()
```

- R Markdown code for plotting steps vs. sleep hours



- Scatterplot showing no strong correlation between steps and sleep hours

The analysis shows no strong correlation. This suggests that activity levels may not influence sleep. This might be interesting if considering whether activity tracking and sleep tracking should be marketed as related or distinct features.

Analyze Phase: Step 4- Weekly activity patterns

To gain a better understanding of how fitness trackers are being used, the next question to explore is which days of the week were users the most or least active.

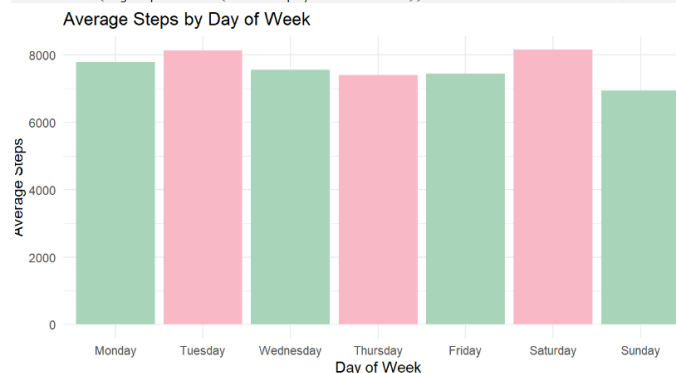
By grouping average steps by days of the week and plotting the results in a bar chart we found that users were most active on Tuesdays and Saturdays, but were least active on Sundays.

Step 4: Weekly Activity Patterns

To identify when users are most active, we'll calculate average steps by day of the week. This may offer insight into lifestyle patterns that could be informative for targeted Bellabeat marketing campaigns.

```
## [r weekly-steps, fig.width=7, fig.height=4]
#Calculate average steps by day of week
weekly_steps <- merged_data %>%
  mutate(DayOfWeek = factor(DayOfWeek, levels = c("Monday", "Tuesday", "Wednesday",
"Thursday", "Friday", "Saturday", "Sunday"))) %>%
  group_by(DayOfWeek) %>%
  summarise(AvgSteps = mean(TotalSteps, na.rm = TRUE))
```

-R Markdown code for analyzing weekly activity patterns



- Bar chart showing highest steps on Tuesday and Saturday, lowest on Sunday

As a physiotherapist, I found this particularly interesting since I often try to encourage my clients to incorporate changes in their activity level around their normal lifestyle. With the knowledge gained from this analysis, we can consider encouraging marketing, promotions, or social media activity to align with our users natural tendencies.

Analyze Phase: Step 5- Sleep Patterns by Day of Week

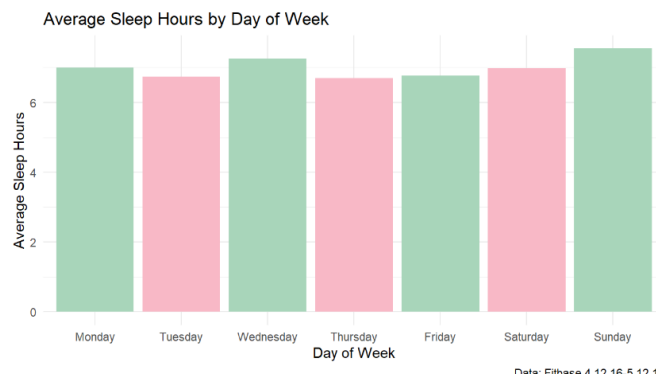
Building on Step 4, we analyzed average sleep hours grouped by day of the week, finding higher average sleep on Sunday (~7.5 hours) and lower average sleep on Tuesday (~6.7 hours).

```
## R Markdown code for analyzing sleep patterns by day of week

# Calculate average sleep hours by day of week
sleep_by_day <- merged_data %>%
  mutate(DayOfWeek = factor(DayOfWeek, levels = c("Monday", "Tuesday", "Wednesday",
    "Thursday", "Friday", "Saturday", "Sunday"))) %>%
  group_by(DayOfWeek) %>%
  summarise(AvgSleepHours = mean(TotalMinutesAsleep / 60, na.rm = TRUE))

# Visualize with a bar chart
ggplot(sleep_by_day, aes(x = DayOfWeek, y = AvgSleepHours, fill = DayOfWeek)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("Monday" = "#A8D5BA", "Tuesday" = "#F8BBD0", "Wednesday" =
    "#A8D5BA", "Thursday" = "#F8BBD0", "Friday" = "#A8D5BA", "Saturday" = "#F8BBD0", "Sunday" =
    "#A8D5BA")) +
  labs(title = "Average Sleep Hours by Day of Week",
    x = "Day of Week",
    y = "Average Sleep Hours",
    caption = "Data: Fitbase 4.12.16-5.12.16") +
  theme_minimal() +
  theme(legend.position = "none")
```

- R Markdown code for analyzing sleep patterns by day of week



- Bar chart showing sleep variations across the week

Bearing in mind what we learned in Step 4 about weekly activity patterns (highest steps on Tuesday and Saturday, lowest on Sunday), we can see how our sleep findings match. Sunday has the highest sleep which aligns with it being a less active day. Tuesday has the lowest sleep which aligns with it being a more active day. This likely aligns with their work or activity schedules. This will be another useful insight when considering how to promote features or uses of Bellabeat products.

Analyze Phase: Step 6- Sedentary vs. Active behavior

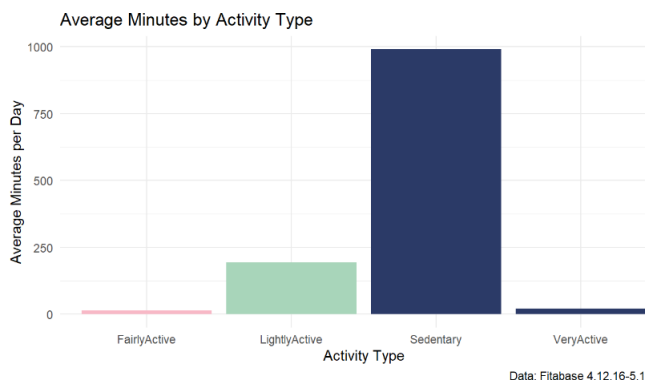
The next comparison was between minutes spent being sedentary and different levels of activity. We found that sedentary activity is the vast majority of most users' days with an average of 991 minutes (over 16.5 hours!) with much less time spent in very active (~21

minutes), fairly active (~13 minutes) and lightly active (~192 minutes) activity. The total minutes across activity types (1217 minutes) do not account for sleep time (~420 minutes), suggesting FitBit may classify some sleep periods as sedentary. This highlights a need for clearer activity categorization in future data collection.

```
## Step 5: Sedentary vs. Active behavior
To understand user activity profiles, we will compare how users balance their activity
between sedentary minutes, lightly active minutes, fairly active minutes, and very active
minutes. This may be informative for strategies to encourage more activity.

```{r sedentary vs active, fig.width=7, fig.height=4}
#Summarize activity minutes
activity_profile <- merged_data %>%
 summarise(
 Sedentary = mean(SedentaryMinutes.x, na.rm=TRUE),
 LightlyActive = mean(LightlyActiveMinutes.x, na.rm = TRUE),
 FairlyActive = mean(FairlyActiveMinutes.x, na.rm = TRUE),
 VeryActive = mean(VeryActiveMinutes.x, na.rm = TRUE)
) %>%
 pivot_longer(everything(), names_to = "ActivityType", values_to = "Minutes")
```

- R Markdown code for comparing time spent at different level of activity



- Bar chart showing high sedentary time compared to active time

This shows that even on days when users are more active (Tuesday and Saturday), they are still spending the majority of their time being sedentary. This will be important to educate to users since research shows important benefits to increasing light activity throughout the day. For example, research by LaCroix et al [here](#) shows that adults who engaged in more than six hours of light activity per day had a 46% lower risk of heart attack.

### Analyze Phase: Step 7- Heart rate and activity correlation

We noted earlier that only ~42% of records include heart rate data, so analyzing benefits and uses of this data may show us opportunities to promote and market heart rate tracking features.

To assess the relationship between activity levels and heart rate, we will plot average heart rate against steps and calories.

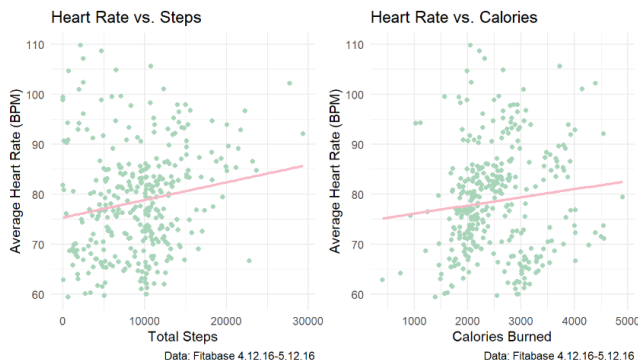
```
Filter non-NA heart rate data
heart_rate_data <- merged_data %>% filter(!is.na(AvgHeartRate))

Scatter plot: Heart rate vs. steps
p1 <- ggplot(heart_rate_data, aes(x = TotalSteps, y = AvgHeartRate)) +
 geom_point(color = "#A8D5BA") +
 geom_smooth(method = "lm", se = FALSE, color = "#F8B8C6") +
 labs(title = "Heart Rate vs. Steps",
 x = "Total Steps",
 y = "Average Heart Rate (BPM)",
 caption = "Data: Fitabase 4.12.16-5.12.16") +
 theme_minimal()

Scatter plot: Heart rate vs. calories
p2 <- ggplot(heart_rate_data, aes(x = Calories, y = AvgHeartRate)) +
 geom_point(color = "#A8D5BA") +
 geom_smooth(method = "lm", se = FALSE, color = "#F8B8C6") +
 labs(title = "Heart Rate vs. Calories",
 x = "Calories Burned",
 y = "Average Heart Rate (BPM)",
 caption = "Data: Fitabase 4.12.16-5.12.16") +
 theme_minimal()

Display plots side by side
grid.arrange(p1, p2, ncol = 2)
```

- R Markdown code for analyzing heart rate correlations



- Scatter plots showing correlations between heart rate and steps/calories

The plots indicate a positive but weak relationship, most likely due to low tracking numbers for heart rate. This may suggest an opportunity to promote heart rate tracking features in Bellabeat products.

### Analyze Phase: Step 8- Summary of Key Findings

Steps 1-7 of the Analyze phase uncovered several interesting insights from the FitBit user data which can help inform Bellabeat's marketing strategies.

- Step 1: Loaded and previewed our cleaned and merged dataset (*merged\_data\_cleaned.csv*).
- Step 2: Summarized key metrics, finding average steps at ~7638, sleep at ~7 hours, and heart rate tracking at ~42%.

- Step 3: Plotted steps vs. sleep hours showing no correlation, showing that activity and sleep are independent wellness factors.
- Step 4: Analyzed weekly activity patterns showing the highest step totals on Tuesday and Saturday, and the lowest on Sunday.
- Step 5: Examined weekly sleep patterns, finding higher sleep on Sundays and lower on Tuesday.
- Step 6: Compared sedentary vs. active minutes finding that sedentary time dominates, with ~16.5 hours/day even on active days indicating a need to encourage more movement.
- Step 7: Explored how heart rate correlates with steps and calories, finding a weak positive relationship which along with the low tracking rates of heart rate metrics suggest an opportunity to promote heart rate monitoring features.

## Phase 5: Share

### Our task:

*Communicate insights gained from data analysis*

### Share Phase: Step 1- Key Findings Recap

We'll continue to work in our R Markdown report for this phase, so we can see the natural flow from the analyze phase to the share phase. It will also make it easier to share our entire process and analysis with the rest of our team.

We began by briefly recapping the most actionable findings to prepare to make recommendations.

- Weekly Activity Patterns (Step 4): Users are most active on Tuesdays and Saturdays, with the lowest activity on Sunday.
- Sleep Patterns (Step 5): Users sleep the most on Sunday (7.55 hours) and the least on Tuesday (6.74 hours), with only ~44% tracking sleep.
- Sedentary Behavior (Step 6): Sedentary time dominates at ~16.5 hours/day, even on active days, indicating a need to encourage more movement throughout the day.
- Heart Rate Tracking (Step 7): Only ~42% of users track heart rate, with a weak positive correlation to steps/calories, suggesting that this feature is not used enough.

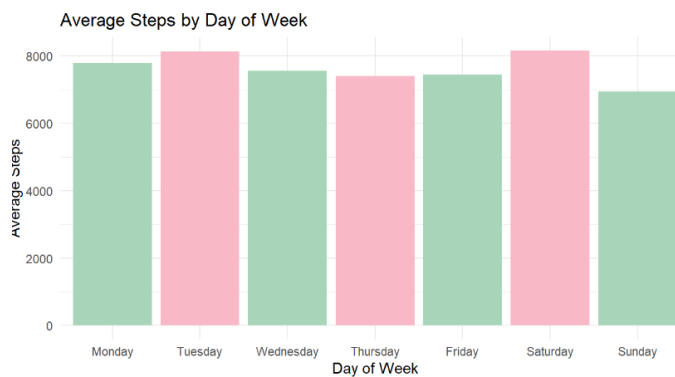


We made note of the importance of recognizing the limitations of our data (from 2016, limited time frame, not necessarily women), explaining that in the Act phase we'll make recommendations for improving our data to accurately reflect Bellabeat's users.

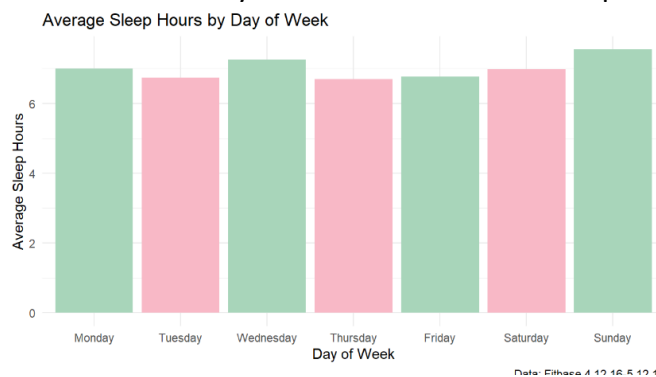
## Share Phase: Step 2- Visualizations and insights

The following visualizations highlight the key trends, telling a story of user engagement and opportunities for Bellabeat:

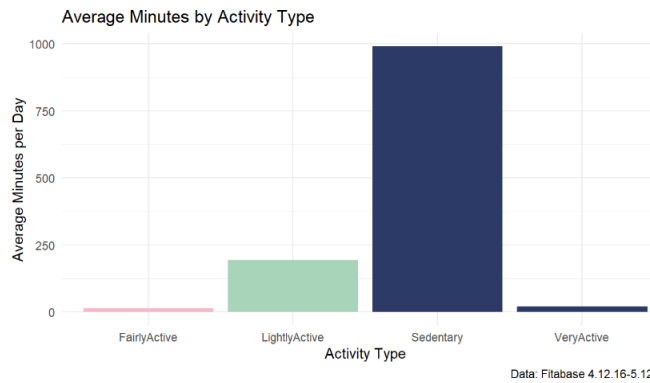
**Weekly Activity Patterns (Step 4):** The chart below shows that users take the most steps on Tuesday and Saturday, with Sunday being the least active day. This suggests lifestyle patterns (e.g., work, family or social activities on weekdays/weekends) that Bellabeat can leverage for targeted engagement.



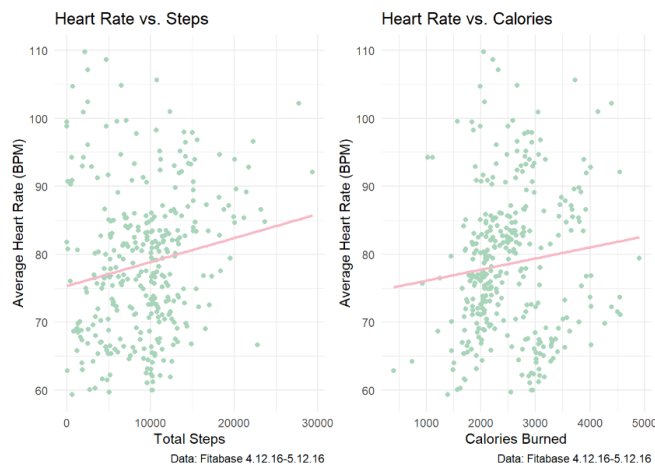
**Sleep Patterns (Step 5):** The next chart shows that users sleep more on Sundays (7.55 hours) and less on Tuesdays (6.74 hours), indicating a need for better rest on active days. It should also be noted that only ~44% of users tracked sleep.



**Sedentary vs. Active Behavior (Step 6):** This bar chart reveals that users spend ~16.5 hours/day sedentary, even on active days, highlighting a critical opportunity to encourage more frequent movement throughout the day.



**Heart Rate and Activity Correlation (Step 7):** Scatter plots show a weak positive correlation between heart rate and steps/calories, with only ~42% tracking heart rate, suggesting that this feature is being underused.



```
Visualizations and Insights
Our visualizations highlight the key findings identified in the Analyze phase, telling a story about user habits and how they engage with their health tracking devices and where opportunities may lie for Bellabeat:

- **Weekly Activity Patterns (Step 4)**: The chart below shows that users take the most steps on Tuesday and Saturday, with Sunday being the least active day. This suggests lifestyle patterns (e.g., work, family or social activities on weekdays/weekends) that Bellabeat can leverage for targeted engagement.
{r}
steps_plot

- **Sleep Patterns (Step 5)**: The bar chart shows that users sleep more on Sundays (7.55 hours) and less on Tuesdays (6.74 hours), indicating a need for better rest on active days. It should also be noted that only ~44% of users tracked sleep.
{r}
sleep_plot

- **Sedentary vs. Active Behavior (Step 6)**: This bar chart reveals that users spend ~16.5 hours/day sedentary, even on active days, highlighting a critical opportunity to encourage more frequent movement throughout the day.
{r}
sedentary_plot

- **Heart Rate and Activity Correlation (Step 7)**: Scatter plots show a weak positive correlation between heart rate and steps/calories, with only ~42% tracking heart rate, suggesting that this feature is being underused.
{r}
grid.arrange(hr_steps_plot, hr_calories_plot, ncol = 2)
```

-R Markdown code for the Share phase, presenting key findings and visualizations

## Share Phase: Visualizations and Storytelling

In the Share phase, we will present the key findings from the Analyze phase (Steps 1-7). These insights highlight trends in FitBit user data, setting the stage for marketing recommendations for Bellabeat products in the Act phase.

### Key Findings Recap

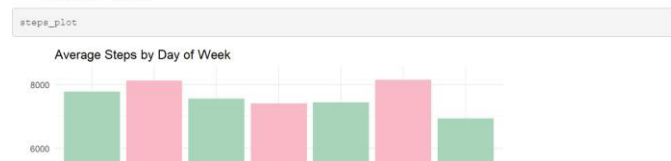
The Analyze phase revealed the following key trends from the FitBit user data:

- **Weekly Activity Patterns (Step 4):** Users are most active on Tuesdays and Saturdays, with the lowest activity on Sunday.
- **Sleep Patterns (Step 5):** Users sleep the most on Sunday (7.55 hours) and the least on Tuesday (6.74 hours), with only ~44% tracking sleep.
- **Sedentary Behavior (Step 6):** Sedentary time dominates at ~16.5 hours/day, even on active days, indicating a need to encourage more movement throughout the day.
- **Heart Rate Tracking (Step 7):** Only ~42% of users track heart rate, with a weak positive correlation between heart rate and steps/calories, suggesting that this feature is not used enough.

### Visualizations and Insights

Our visualizations highlight the key findings identified in the Analyze phase, telling a story about user habits and how they engage with their health tracking devices and where opportunities may lie for Bellabeat:

- **Weekly Activity Patterns (Step 4):** The chart below shows that users take the most steps on Tuesday and Saturday, with Sunday being the least active day. This suggests lifestyle patterns (e.g., work, family or social activities on weekdays/weekends) that Bellabeat can leverage for targeted engagement.



-Screenshot of Share Phase HTML output

## Phase 5: Act

### Our task:

*Present recommendations for Bellabeat products based on our Analyze phase findings*

### Act Phase: Step 1- Marketing recommendations

We made the following marketing recommendations

Based on these findings, we propose the following strategies to enhance marketing of the Bellabeat app and Leaf tracker:

- **Encourage Activity on Low-Step Days (Sunday):** Launch a “Sunday Stroll” campaign, encouraging users to take a short walk with Leaf’s step tracking, based on Sunday being the low activity day as identified in Step 4.
- **Promote Sleep Tracking on High-Activity Days (Tuesday/Saturday):** Use Step 5’s finding of lower sleep on active days like Tuesdays to promote Leaf’s sleep tracking with messages to app users like “Busy Tuesday? Track your sleep with Leaf to ensure recovery!” This encourages better rest habits, which are crucial for health.
- **Boost Heart Rate Tracking (Step 7):** Address the low tracking rate (~42%) with a campaign like “With All Your Heart!” emphasizing Leaf’s heart rate feature on the busiest high-step days (Tuesday/Saturday). This can enhance users’ understanding of their

physiological response to activity and encourage engagement with their Bellabeat products.

- **Host a “Move More Challenge” with Social Sharing (Step 6):** Launch a 30-day challenge to reduce sedentary time (~16.5 hours/day) by encouraging small daily activities (e.g., 5-minute walks hourly). Highlight research (e.g., LaCroix et al., 2025) showing how consistent light activity reduces heart attack risk by 46%. Add social sharing in the Bellabeat app, letting users post progress with #MoveMoreWithLeaf, fostering community and brand visibility.
- **Offer a “Buddy System” for Activity Goals (Step 4):** Add a Bellabeat app feature for users to pair with a friend or another random user for step goals, with in-app cheers or badges when both succeed, encouraging light movement and community especially on low-activity days.
- **Develop a “Wellness Insights Blog Series” (Steps 5 and 7):** Launch a blog series on Bellabeat’s site with topics like “Why Tracking Sleep on Active Days Matters” (noting Tuesday’s 6.74 hours sleep) and “How Heart Rate Tracking Boosts Wellness.” Reach out to users for testimonials to publish with the blogs to drive interest in Leaf’s tracking features.

#### Act Phase: Step 1.5- Additional Recommendation

The FitBit dataset used in this analysis provides valuable insights, but it may not fully reflect Bellabeat’s user base, particularly given Bellabeat’s focus on women’s wellness and the limited time frame and sample size of the FitBit data. To ensure that the success metrics for these marketing strategies are realistic and meaningful, we recommend that Bellabeat collect usage data from its own clients before fully implementing these campaigns. This data should include:

- Average daily steps by day of week, to establish baselines for campaigns like "Sunday Stroll."
- Percentage of users tracking sleep, especially on high-activity days.
- Average daily sedentary minutes, to set a realistic target for the "Move More Challenge."
- Percentage of users tracking heart rate, to tailor the "With All Your Heart!" campaign.

Collecting this data over 3-6 months via the Bellabeat app (complying with protocols for user consent and privacy) will provide a solid foundation for setting achievable targets and measuring success.

## Act Phase: Step 2- Implementation plan

Bellabeat can implement these marketing recommendations with the following phased approach:

1. **Data Collection (Q2 2025):** Collect usage data from Bellabeat users over 3-6 months to establish baselines for steps, sleep tracking, sedentary time, and heart rate tracking, as outlined in the recommendation above.
2. **Launch Campaigns (Q3 2025):** Roll out the “Sunday Stroll,” “With All Your Heart!,” and “Move More Challenge” campaigns, using social media (e.g. Instagram, Facebook, X), Google Search ads, and in-app notifications. Partner with wellness influencers to promote the “Move More Challenge” with #MoveMoreWithLeaf.
3. **Update the Bellabeat App (June 2025):** Add features like sleep tracking notifications on selected days, a buddy system for step goals, and social sharing capabilities, ensuring seamless integration with Leaf’s tracking features.
4. **Content Strategy (July 2025):** Launch the “Wellness Insights Blog Series,” featuring user testimonials and educational content on sleep and heart rate tracking, promoted via email campaigns and social media.

## Act Phase: Step 3- Success metrics

To evaluate the impact of these strategies, Bellabeat can track the following metrics over the first 6 months. Since the FitBit data may not reflect Bellabeat users, we recommend using these as interim engagement metrics while Bellabeat collects its own baseline data:

- **Sunday Activity:** Measure the percentage of users who engage with the “Sunday Stroll” campaign (target: 10% log a walk on Sundays after receiving notifications) and buddy system participation rates (target: 5% of users join within 3 months).
- **Sleep Tracking:** Measure the percentage of users who start tracking sleep on Tuesdays and Saturdays after receiving notifications (target: 10% of users who receive the message begin tracking within a month).
- **Sedentary Time:** Measure participation in the “Move More Challenge” (target: 10% of users join and log at least one activity per day) and social sharing activity (target: 500 #MoveMoreWithLeaf posts within 6 months).
- **Heart Rate Tracking:** Measure the percentage of users who activate heart rate tracking after the “With All Your Heart!” campaign (target: 8% of users who see the campaign start tracking within a month).

- **User Engagement:** Assess social sharing activity (e.g., #MoveMoreWithLeaf posts) and blog engagement metrics (e.g., page views, click-through rates) to gauge community and brand visibility.

Once Bellabeat collects its own data, these metrics can be updated with specific targets (e.g., 10% increase in Sunday steps, 10% increase in heart rate tracking) based on the newly established baselines.

We coded the sections of the Act phase into our R Markdown file and made sure it knits as we expect it to. (So glad we took the time to figure out how make the .css so we could use Bellabeat branded colors for the headers! It looks great!)

## Act Phase: Recommendations and Next Steps

In the Act phase, we present recommendations for Bellabeat based on our Analyze phase findings, followed by an implementation plan and success metrics to ensure these strategies enhance user engagement and wellness. These recommendations align with Bellabeat's mission of empowering women with knowledge about their own health and habits.

### Marketing Recommendations

Based on these findings, we propose the following strategies to enhance marketing of the Bellabeat app and Leaf tracker:

- **Encourage Activity on Low-Step Days (Sunday):** Launch a "Sunday Stroll" campaign, encouraging users to take a short walk with Leaf's step tracking, based on Sunday being the low activity day as identified in Step 4.
- **Promote Sleep Tracking on High-Activity Days (Tuesday/Saturday):** Use Step 5's finding of lower sleep on active days like Tuesdays to promote Leaf's sleep tracking with messages to app users like "Busy Tuesday? Track your sleep with Leaf to ensure recovery!" This encourages better rest habits, which are crucial for health.
- **Boost Heart Rate Tracking (Step 7):** Address the low tracking rate (~42%) with a campaign like "With All Your Heart!" emphasizing Leaf's heart rate feature on the busiest high-step days (Tuesday/Saturday). This can enhance users' understanding of their physiological response to activity and encourage engagement with their Bellabeat products.
- **Host a "Move More Challenge" with Social Sharing (Step 6):** Launch a 30-day challenge to reduce sedentary time (~16.5 hours/day) by encouraging small daily activities (e.g., 5-minute walks hourly). Highlight research (e.g., LaCroix et al., 2025) showing how consistent light activity reduces heart attack risk by 46%. Add social sharing in the Bellabeat app, letting users post progress with #MoveMoreWithLeaf, fostering community and brand visibility.
- **Offer a "Buddy System" for Activity Goals (Step 4):** Add a Bellabeat app feature for users to pair with a friend or another random user for step goals, with in-app cheers or badges when both succeed, encouraging light movement and community especially on low-activity days.
- **Develop a "Wellness Insights Blog Series" (Steps 5 and 7):** Launch a blog series on Bellabeat's site with topics like "Why Tracking Sleep on Active Days Matters" (noting Tuesday's 6.74 hours sleep) and "How Heart Rate Tracking Boosts Wellness." Reach out to users for testimonials to publish with the blogs to drive interest in Leaf's tracking features.

### Additional Recommendation

#### Collect Bellabeat User Data for Accurate Baselines

The FitBit dataset used in this analysis provides valuable insights, but it may not fully reflect Bellabeat's user base, particularly given Bellabeat's focus on women's wellness and the limited time frame and sample size of the FitBit data. To ensure that the success metrics for these marketing strategies are realistic and meaningful, we recommend that Bellabeat collect usage data from its own clients before fully implementing these

As mentioned in the Ask phase, we hope that the recommendations and plans show a balance of creativity for Urška Sršen and data-driven insights for Sando Mur.

**Summary of Act phase:** In the Act phase, we proposed 6 marketing strategies for Bellabeat's Leaf tracker in `analyze\_data.Rmd`, including campaigns like "Sunday Stroll" and "With All Your Heart!," a "Move More Challenge," a buddy system, and a blog series. Recognizing that the FitBit data may not reflect Bellabeat's users, we also recommended collecting Bellabeat-specific usage data to establish accurate baselines. We outlined a phased implementation plan and interim success metrics, which to be updated once Bellabeat-specific baselines are set.