Bellabit Analysis

Emmanuel Stephen Chigozie

2025-01-14

Bellabeat Case Study

Table Of Contents

Introduction

Bellabeat is a wellness-focused technology company specializing in elegantly designed smart devices and apps that empower women with valuable insights into their health and daily habits. Founded in 2013 by Urška Sršen and Sando Mur, Bellabeat integrates its app with smart wellness products to provide users with data on activity, sleep, stress, menstrual cycles, and mindfulness practices, enabling informed health choices. By 2016, the company had expanded its global reach, offering products through its website and online retailers. Bellabeat leverages digital marketing strategies, including Google Search, social media, YouTube, and the Google Display Network, alongside traditional advertising channels like radio and print, to build brand awareness and drive growth.

Ask Phase

Business task: Bellabeat is looking for an effective strategy to use smart devices to boost marketing.

Objective

This project will focus on analyzing data from the Bellabeat app to uncover insights into user habits and preferences. These insights will inform high-level recommendations to optimize Bellabeat's marketing strategy and drive engagement.

Stakeholders

1. Urška Sršen: Bellabeat's cofounder and Chief Creative Officer.
2. Sando Mur: Mathematician and Bellabeat's cofounder; key member of the Bellabeat executive team.

Prepare Phase

About the Data

This project will make use of the FitBit Fitness Tracker Data, which is a public domain dataset made available through Mobius. This Kaggle dataset contains personal fitness tracker data from thirty FitBit users. This means that thirty eligible users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. It includes information about daily activity, steps, and calories that can be used to explore user habits. The dataset does not include personally identifiable information (PII), as it contains aggregated fitness data like activity, heart rate, and sleep patterns. This ensures the privacy of the individuals who provided the data. Furthermore, ethical standards in data analysis have been emphasized by respecting the original intent of the data collection and focusing on aggregated insights rather than individual-level analysis. Lastly, even though the data is anonymized, secure data storage and handling practices have been implemented to prevent unauthorized access or misuse.

The dataset originates from Mobius on Kaggle, a trusted platform, with proper documentation and licensing. Data quality checks are performed to ensure completeness, consistency, and accuracy by identifying missing values, outliers, and logical inconsistencies, and summary statistics are cross-referenced with publicly available benchmarks for validation. Additionally, metadata is reviewed to confirm collection methods, participant details, and alignment with the dataset's stated purpose. Ethical considerations are also addressed by confirming anonymization, proper participant consent, and adherence to privacy standards, ensuring the dataset is reliable and suitable for analysis.

Process Phase

Setting up my environment and loading of the data.

Tools

R for analysis and R studio IDE

Install and Loading Libraries

```r
install.packages("tidyverse")
```

```
## The following package(s) will be installed:
## - tidyverse [2.0.0]
## These packages will be installed into "C:/Users/User/Desktop/Project work/Fitbase/renv/library/R-4.3/
##
## # Installing packages --------------------------------------------------------
## - Installing tidyverse ...                      OK [linked from cache]
## Successfully installed 1 package in 51 milliseconds.
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

Importing and Previewing Data

```r
daily_activity <- read_csv("dailyActivity_merged.csv")
```

```
## Rows: 940 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): ActivityDate
## dbl (14): Id, TotalSteps, TotalDistance, TrackerDistance, LoggedActivitiesDi...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
View(daily_activity)


dailySteps<- read.csv("dailySteps_merged.csv")
View(dailySteps)
sleepDay<- read_csv("sleepDay_merged.csv")
```

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

```r
View(sleepDay)
```

Cleaning the data

checking for empty values

```r
sum(is.na(daily_activity))
```

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

```r
sum(is.na(sleepDay))
```

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

```r
sum(is.na(dailySteps))
```

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

```r
#Checking for empty rows
empty_rows_activity <- daily_activity[rowSums(is.na(daily_activity)) == ncol(daily_activity), ]
empty_rows_steps<- dailySteps[rowSums(is.na(dailySteps))==ncol(dailySteps), ]
empty_rows_sleep<-sleepDay[rowSums(is.na(sleepDay))==ncol(sleepDay), ]

# Check for empty columns
empty_columns_activity <- daily_activity[, colSums(is.na(daily_activity)) == nrow(daily_activity)]
empty_columns_steps<-dailySteps[, colSums(is.na(dailySteps))==nrow(dailySteps)]
empty_columns_sleep<-sleepDay[, colSums(is.na(sleepDay))==nrow(sleepDay)]

#checking for duplicates

duplicates_dailyActivity <- daily_activity[duplicated(daily_activity), ]
cat("Number of duplicate rows in daily_activity:", nrow(duplicates_dailyActivity), "\n")
```

```
## Number of duplicate rows in daily_activity: 0
```

```r
duplicate_steps<- dailySteps[duplicated(dailySteps),]
cat("Number of duplicate rows in dailySteps:",nrow(duplicate_steps), "\n")
```

```
## Number of duplicate rows in dailySteps: 0
```

```r
duplicates_sleep <- sleepDay[duplicated(sleepDay), ]
cat("Number of duplicate rows in sleep:", nrow(duplicates_sleep), "\n")
```

```
## Number of duplicate rows in sleep: 3
```

```r
# Removing Duplicates

sleepDay <- sleepDay[!duplicated(sleepDay), ]

#Checking for incorrect data types
str(daily_activity)
```

```
## spc_tbl_ [940 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id                      : num [1:940] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ TotalSteps              : num [1:940] 13162 10735 10460 9762 12669 ...
##  $ TotalDistance           : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : num [1:940] 728 776 1218 726 773 ...
##  $ Calories                : num [1:940] 1985 1797 1776 1745 1863 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDate = col_character(),
##   ..   TotalSteps = col_double(),
##   ..   TotalDistance = col_double(),
##   ..   TrackerDistance = col_double(),
##   ..   LoggedActivitiesDistance = col_double(),
##   ..   VeryActiveDistance = col_double(),
##   ..   ModeratelyActiveDistance = col_double(),
##   ..   LightActiveDistance = col_double(),
##   ..   SedentaryActiveDistance = col_double(),
##   ..   VeryActiveMinutes = col_double(),
##   ..   FairlyActiveMinutes = col_double(),
##   ..   LightlyActiveMinutes = col_double(),
##   ..   SedentaryMinutes = col_double(),
##   ..   Calories = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(dailySteps)
```

```
## 'data.frame':    940 obs. of  3 variables:
##  $ Id         : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDay: chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ StepTotal  : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
```

```
str(sleepDay)
```

```
## tibble [410 x 5] (S3: tbl_df/tbl/data.frame)
##  $ Id                : num [1:410] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : chr [1:410] "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:
##  $ TotalSleepRecords : num [1:410] 1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: num [1:410] 327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : num [1:410] 346 407 442 367 712 320 377 364 384 449 ...
```

```r
#Converting character columns to dates
daily_activity$ActivityDate <- as.Date(daily_activity$ActivityDate, format = "%Y/%m/%d")
dailySteps$ActivityDay<-as.Date(dailySteps$ActivityDay,format="%Y/%m/%d")
sleepDay$SleepDay<-as.Date(sleepDay$SleepDay,format="%Y/%m/%d")

#checking datatypes
str(daily_activity)
```

```
## spc_tbl_ [940 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id                      : num [1:940] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : Date[1:940], format: "0004-12-20" NA ...
##  $ TotalSteps              : num [1:940] 13162 10735 10460 9762 12669 ...
##  $ TotalDistance           : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num [1:940] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : num [1:940] 728 776 1218 726 773 ...
##  $ Calories                : num [1:940] 1985 1797 1776 1745 1863 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDate = col_character(),
##   ..   TotalSteps = col_double(),
##   ..   TotalDistance = col_double(),
##   ..   TrackerDistance = col_double(),
##   ..   LoggedActivitiesDistance = col_double(),
##   ..   VeryActiveDistance = col_double(),
##   ..   ModeratelyActiveDistance = col_double(),
##   ..   LightActiveDistance = col_double(),
##   ..   SedentaryActiveDistance = col_double(),
##   ..   VeryActiveMinutes = col_double(),
##   ..   FairlyActiveMinutes = col_double(),
##   ..   LightlyActiveMinutes = col_double(),
##   ..   SedentaryMinutes = col_double(),
##   ..   Calories = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
str(dailySteps)
```

```
## 'data.frame':    940 obs. of  3 variables:
##  $ Id         : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDay: Date, format: "0004-12-20" NA ...
##  $ StepTotal  : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
```

```r
str(sleepDay)
```

```
## tibble [410 x 5] (S3: tbl_df/tbl/data.frame)
##  $ Id                : num [1:410] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : Date[1:410], format: "0004-12-20" NA ...
##  $ TotalSleepRecords : num [1:410] 1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: num [1:410] 327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : num [1:410] 346 407 442 367 712 320 377 364 384 449 ...
```

```r
#Loading necessary packages for cleaning column names
install.packages("janitor")
```

```
## The following package(s) will be installed:
## - janitor [2.2.1]
## These packages will be installed into "C:/Users/User/Desktop/Project work/Fitbase/renv/library/R-4.3
##
## # Installing packages --------------------------------------------------------
## - Installing janitor ...                        OK [linked from cache]
## Successfully installed 1 package in 35 milliseconds.
```

```r
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

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

```r
library(dplyr)


#Cleaning column names
daily_activity<-daily_activity%>%clean_names()
dailySteps<-dailySteps%>%clean_names()
sleepDay<-sleepDay%>%clean_names()

#Viewing updated column names
colnames(daily_activity)
```

```
##  [1] "id"                         "activity_date"
##  [3] "total_steps"                "total_distance"
##  [5] "tracker_distance"           "logged_activities_distance"
##  [7] "very_active_distance"       "moderately_active_distance"
##  [9] "light_active_distance"      "sedentary_active_distance"
## [11] "very_active_minutes"        "fairly_active_minutes"
## [13] "lightly_active_minutes"     "sedentary_minutes"
## [15] "calories"
```

```r
colnames(dailySteps)
```

```
## [1] "id"           "activity_day" "step_total"
```

```r
colnames(sleepDay)
```

```
## [1] "id"                   "sleep_day"            "total_sleep_records"
## [4] "total_minutes_asleep" "total_time_in_bed"
```

Analysis and Share Phase

With the data cleaned and prepared, the next step is to integrate the three datasets into a single unified table.We merge the datasets to create a unified table that combines the key information from daily_activity, daily_steps, and daily_sleep. This integration enables a comprehensive analysis by linking activity, step count, and sleep data for each user on a specific date. By merging these datasets, we can identify patterns, correlations, and trends across multiple aspects of users' behavior, which would not be possible if the data remained fragmented.

Merging the daily_activity and daily_steps datasets

First, the daily_activity and daily_steps datasets are merged using the id and their respective date columns (activity_date and activity_day).

```r
activity_steps_merged <- merge(daily_activity, dailySteps,
                               by.x = c("id", "activity_date"),
                               by.y = c("id", "activity_day"),
                               all = TRUE)

#Merging the merged datasets above to daily_sleep dataset
merged_data<-merge(activity_steps_merged,sleepDay,
                   by.x=c("id","activity_date"),
                   by.y=c("id","sleep_day"),
                   all=TRUE)

head(merged_data)
```

```
##           id activity_date total_steps total_distance tracker_distance
## 1 1503960366    0004-12-20       13162           8.50             8.50
## 2 1503960366    0005-01-20       10602           6.81             6.81
## 3 1503960366    0005-02-20       14727           9.71             9.71
## 4 1503960366    0005-03-20       15103           9.66             9.66
## 5 1503960366    0005-04-20       11100           7.15             7.15
## 6 1503960366    0005-05-20       14070           8.90             8.90
##   logged_activities_distance very_active_distance moderately_active_distance
## 1                          0                 1.88                       0.55
## 2                          0                 2.29                       1.60
## 3                          0                 3.21                       0.57
## 4                          0                 3.73                       1.05
## 5                          0                 2.46                       0.87
## 6                          0                 2.92                       1.08
##   light_active_distance sedentary_active_distance very_active_minutes
## 1                  6.06                         0                  25
## 2                  2.92                         0                  33
## 3                  5.92                         0                  41
## 4                  4.88                         0                  50
## 5                  3.82                         0                  36
## 6                  4.88                         0                  45
##   fairly_active_minutes lightly_active_minutes sedentary_minutes calories
## 1                    13                    328               728     1985
## 2                    35                    246               730     1820
## 3                    15                    277               798     2004
## 4                    24                    254               816     1990
## 5                    22                    203              1179     1819
## 6                    24                    250               857     1959
##   step_total total_sleep_records total_minutes_asleep total_time_in_bed
## 1      13162                   1                  327               346
## 2      10602                   1                  369               396
## 3      14727                   1                  277               309
## 4      15103                   1                  273               296
## 5      11100                  NA                   NA                NA
## 6      14070                   1                  247               264
```

Daily Step Analysis

Here, we categorize users based on the number of steps taken. We also calculate the average, minimum, and maximum daily steps for all users. Step categories include: Very Active: 10,000 steps and above. Fairly Active: Between 7,500 and 9,999 steps. Lightly Active: Between 5,000 and 7,499 steps. Sedentary: Less than 5,000 steps.

```r
# Average, Minimum, and Maximum Daily Steps
summary(merged_data$total_steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    5002    8911    8315   11207   36019
```

```r
# Categorize users based on step count
step_categories <- merged_data %>%
  mutate(StepCategory = case_when(
    total_steps >= 10000 ~ "Very Active",
    total_steps >= 7500 & total_steps < 10000 ~ "Fairly Active",
    total_steps >= 5000 & total_steps < 7500 ~ "Lightly Active",
    total_steps < 5000 ~ "Sedentary",
    TRUE ~ NA_character_ # Handle missing values
  ))

# Check distribution of Step Categories
table(step_categories$StepCategory)
```

```
## 
##  Fairly Active Lightly Active      Sedentary    Very Active 
##          14723          14549          20836          33654
```

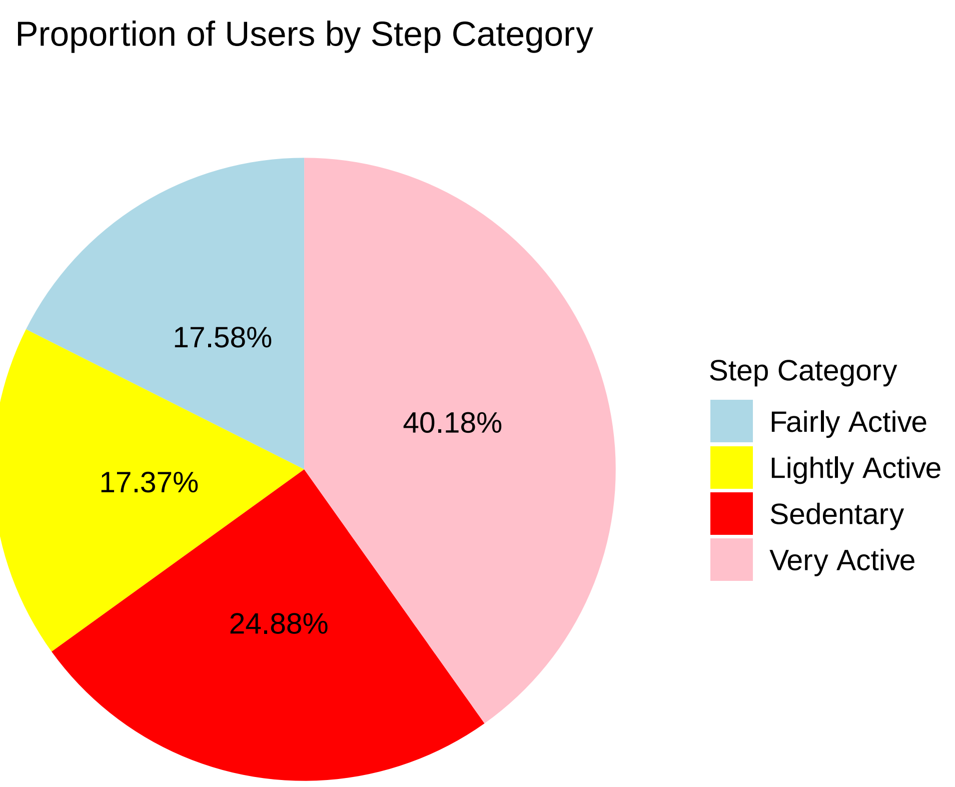
```r
# Calculate proportions of users in each step category
step_category_prop <- step_categories %>%
  group_by(StepCategory) %>%
  summarise(Count = n()) %>%
  mutate(Proportion = round((Count / sum(Count)) * 100, 2))

print(step_category_prop)
```

```
## # A tibble: 4 x 3
##   StepCategory   Count Proportion
##   <chr>          <int>      <dbl>
## 1 Fairly Active  14723       17.6
## 2 Lightly Active 14549       17.4
## 3 Sedentary      20836       24.9
## 4 Very Active    33654       40.2
```

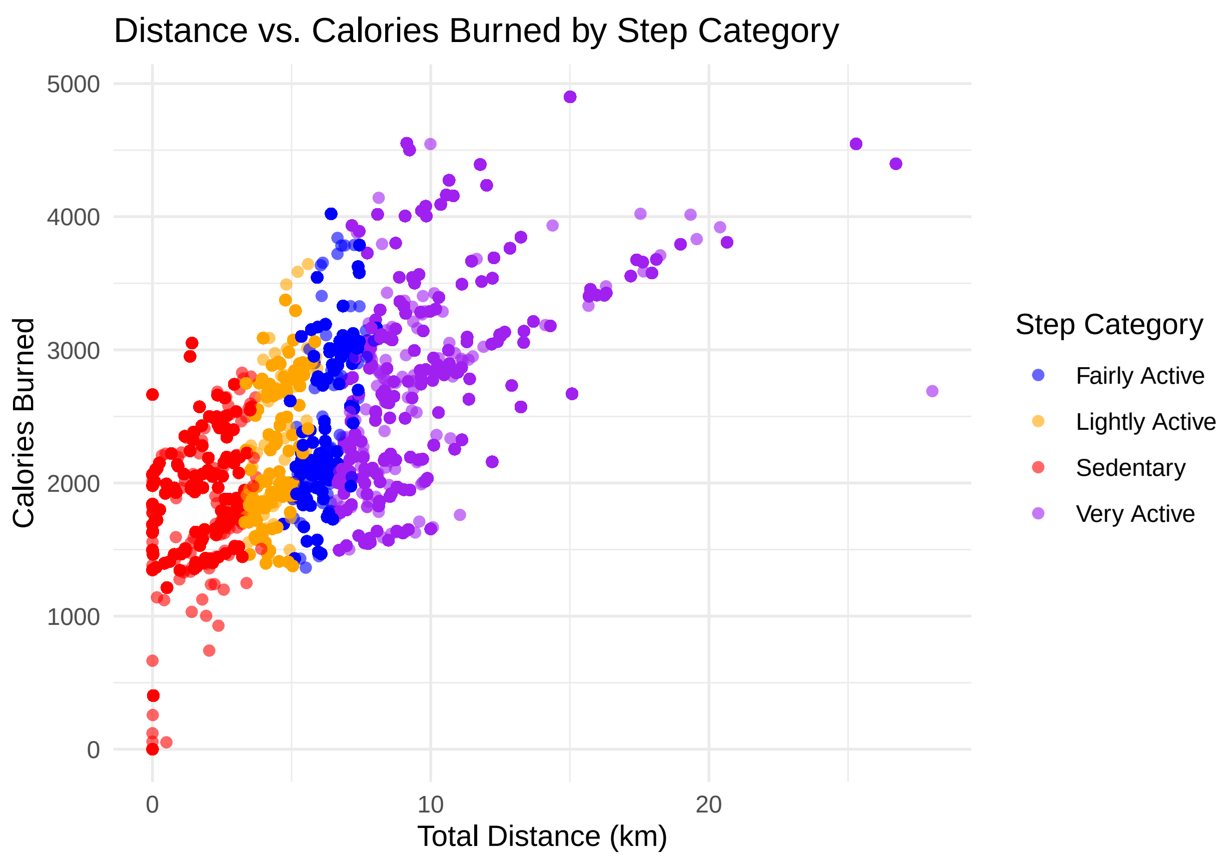
Create the pie chart

```r
ggplot(step_category_prop, aes(x = "", y = Proportion, fill = StepCategory)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  labs(
    title = "Proportion of Users by Step Category",
    fill = "Step Category"
  ) +
  theme_void() +
  scale_fill_manual(values = c("Sedentary" = "red", "Lightly Active" = "yellow", "Fairly Active" = "lig
  geom_text(aes(label = paste0(Proportion, "%")), position = position_stack(vjust = 0.5)) +
  theme(
    plot.title = element_text(hjust = 0.5, margin = margin(b = 10)), # Center title, reduce bottom marg
    legend.position = "right", # Move legend to the right
    legend.text = element_text(size = 11),
    plot.margin = margin(5, 5, 5, 5) # Reduce the plot margins
  )
```



Distance V Calorie Burn

```
ggplot(step_categories, aes(x = total_distance, y = calories, color = StepCategory)) +
  geom_point(alpha = 0.6) +
  labs(
    title = "Distance vs. Calories Burned by Step Category",
    x = "Total Distance (km)",
    y = "Calories Burned",
    color = "Step Category"
  ) +
  theme_minimal() +
  scale_color_manual(values = c("Sedentary" = "red", "Lightly Active" = "orange", "Fairly Active" = "blu
```



Observations

Positive Correlation: There is a clear positive relationship between distance traveled and calories burned. As users cover more distance, they tend to burn more calories.

Step Categories:

a. Sedentary (red): These users generally show the lowest distances traveled and calories burned, clustered toward the bottom-left.
b. Lightly Active (yellow): These points show a moderate increase in both distance and calorie burn compared to sedentary users.
c. Fairly Active (blue): Users in this category display a higher range of distance and calorie burn, indicating increased activity levels.
d. Very Active (purple): These users cover the highest distances and burn the most calories, with points extending farthest along both axes.

Spread Within Categories: Each step category shows some spread in calorie burn for similar distances, likely influenced by factors like body weight, age, and exercise intensity.

Insights: 1. The plot emphasizes the role of physical activity in increasing calorie expenditure. 2. Encouraging users to move from lower activity categories (e.g., Sedentary) to higher ones (e.g., Fairly or Very Active) can lead to substantial improvements in calories burned and overall fitness. 3. Additional factors (e.g., pace, incline) might explain variations in calorie burn for similar distances within a step category.

Calorie Burn v Active Minute

This comprises the average and total number of calories burned by each user. We correlate calorie burn with active minutes to better understand user participation in physical activities.

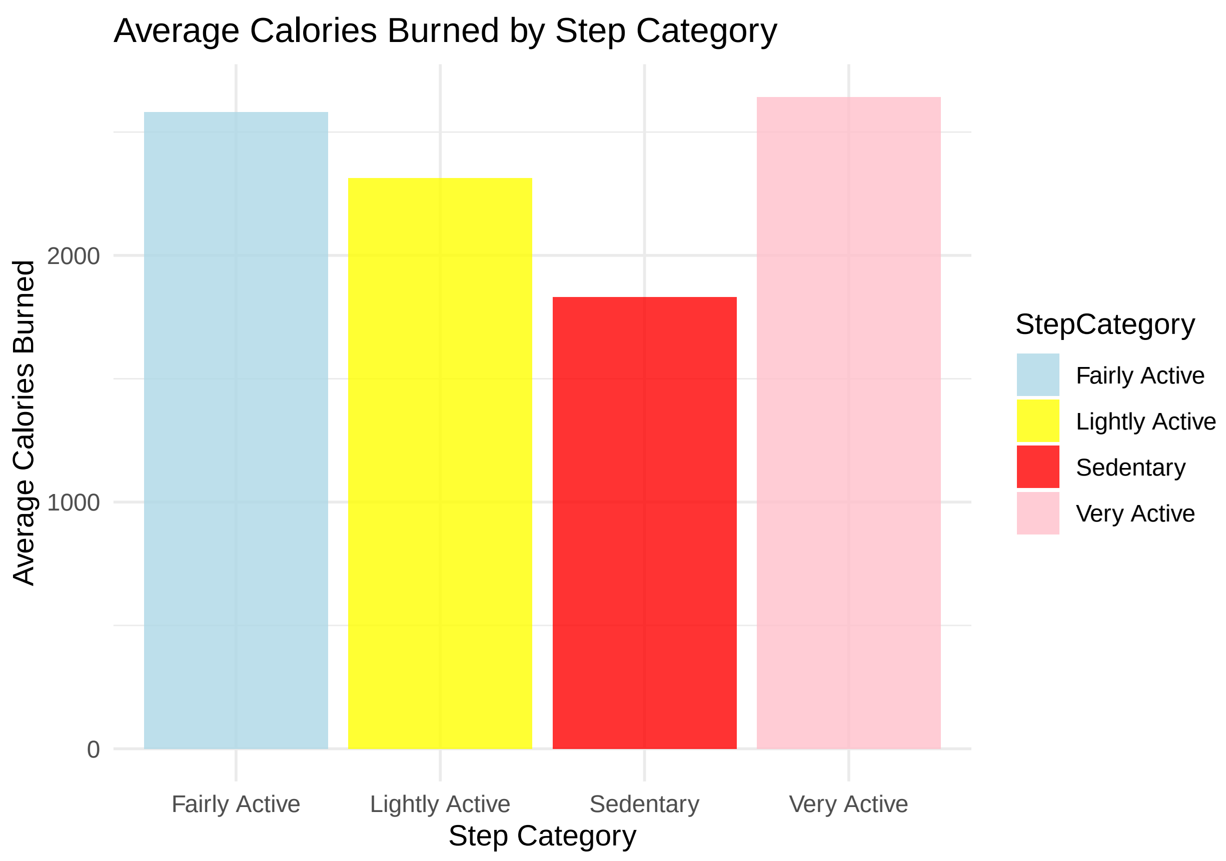
```r
step_categories <- merged_data %>%
  mutate(StepCategory = case_when(
    total_steps >= 10000 ~ "Very Active",
    total_steps >= 7500 ~ "Fairly Active",
    total_steps >= 5000 ~ "Lightly Active",
    TRUE ~ "Sedentary"
  ))
#Calculate Average Calories and Active Minutes per Step Category
summary_stats <-step_categories %>%
  group_by(StepCategory) %>%
  summarize(
    AvgCaloriesBurned = mean(calories, na.rm = TRUE),
    AvgActiveMinutes = mean(very_active_minutes + fairly_active_minutes + lightly_active_minutes, na.rm
  )

print(summary_stats)
```

```
## # A tibble: 4 x 3
##   StepCategory   AvgCaloriesBurned AvgActiveMinutes
##   <chr>                      <dbl>            <dbl>
## 1 Fairly Active              2582.             287.
## 2 Lightly Active             2314.             247.
## 3 Sedentary                  1831.             118.
## 4 Very Active                2643.             313.
```

```r
#Calories Burned vs Active Minutes by Step Category

ggplot(summary_stats, aes(x = StepCategory, y = AvgCaloriesBurned, fill = StepCategory)) +
  geom_bar(stat = "identity", alpha = 0.8) +
  labs(
    title = "Average Calories Burned by Step Category",
    x = "Step Category",
    y = "Average Calories Burned"
  ) +
  scale_fill_manual(values = c("Sedentary" = "red", "Lightly Active" = "yellow", "Fairly Active" = "lig
  theme_minimal()
```



Observations

a. Sedentary Users: Users in this category burned the fewest average calories, which is expected due to low physical activity levels.

b. Lightly and Fairly Active Users: These categories show a moderate increase in average calories burned compared to sedentary users, reflecting increased physical activity.

c. Very Active Users: This category burned the highest average calories, demonstrating a direct correlation between high activity levels and calorie expenditure.

Insights: This chart effectively illustrates the relationship between physical activity and energy expenditure. It emphasizes how higher step counts (indicative of more activity) significantly impact the number of calories burned, which can inform recommendations for users aiming to improve fitness or manage weight.

Sleep duration

Here, we analyze the average, minimum, and maximum values for total_minutes_asleep and total_time_in_bed to understand user sleep patterns. Additionally, we assess how many users meet the recommended 7-9 hours of sleep daily by comparing total_time_in_bed vs total_minutes_asleep.

```
# Summary of Sleep Duration
summary(merged_data$total_minutes_asleep)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    58.0   361.0   430.0   418.3   488.0   796.0    2804
```

```
summary(merged_data$total_time_in_bed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##      61     406     461     457     522     961    2804
```

```
# Proportion of users with recommended sleep duration
prop_good_sleep <- mean(merged_data$total_minutes_asleep >= 420 & merged_data$total_minutes_asleep <= 54
cat("Proportion of users with recommended sleep duration:", round(prop_good_sleep * 100, 2), "%\n")
```

```
## Proportion of users with recommended sleep duration: 46.48 %
```

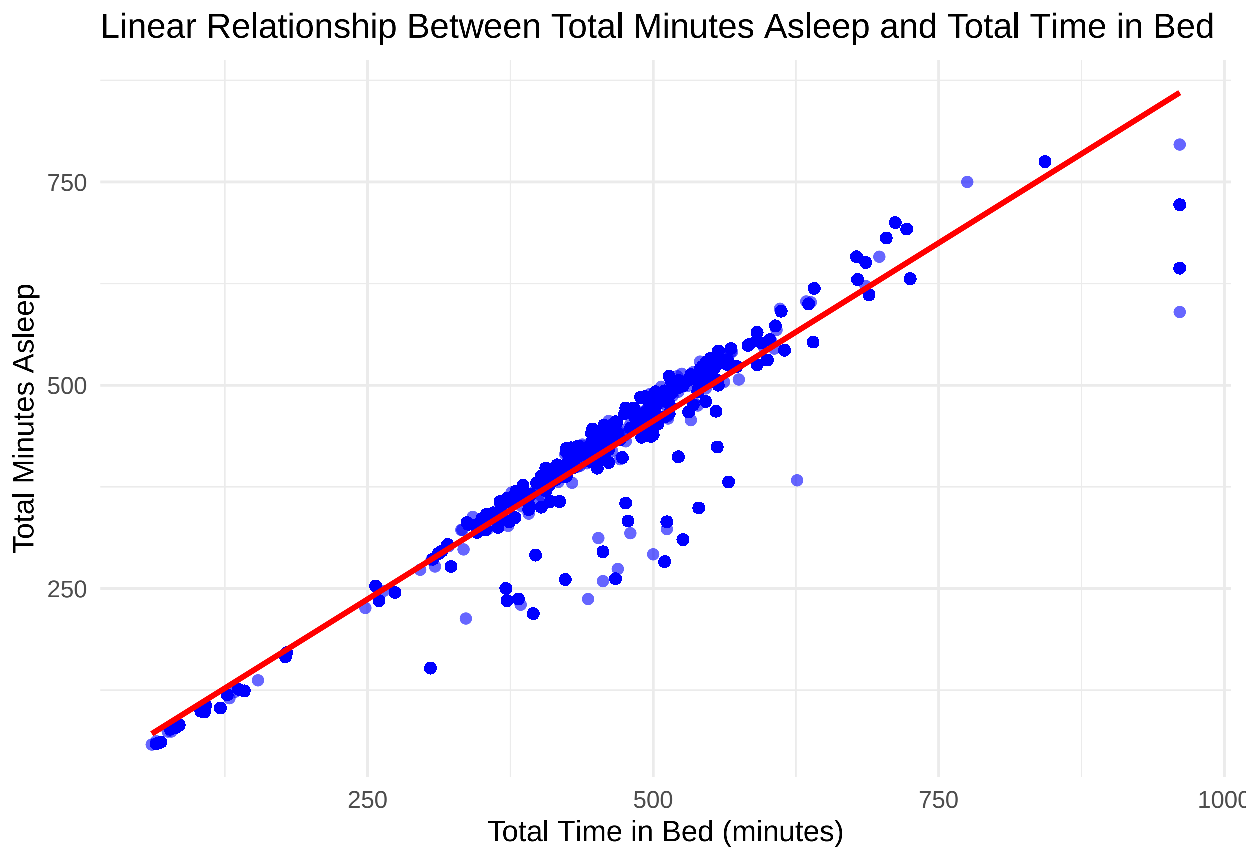
```
#Total Time Asleep vs Total Time in Bed
ggplot(merged_data, aes(x = total_time_in_bed, y = total_minutes_asleep)) +
  geom_point(alpha = 0.6, color = "blue") +
  geom_smooth(method = "lm", se = FALSE, color = "red", size = 1) +
  labs(title = "Linear Relationship Between Total Minutes Asleep and Total Time in Bed",
       x = "Total Time in Bed (minutes)",
       y = "Total Minutes Asleep") +
  theme_minimal()
```

```
## 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.

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

## Warning: Removed 2804 rows containing non-finite outside the scale range
## ('stat_smooth()').

## Warning: Removed 2804 rows containing missing values or values outside the scale range
## ('geom_point()').
```



Observations

1. Strong Positive Correlation: The scatterplot shows a strong positive relationship between the total time spent in bed and the minutes asleep. As the total time in bed increases, the total minutes asleep also increase. Most points are clustered close to the red line, indicating a consistent trend.

2. Outliers: Some points fall below the red line, showing instances where users spent more time in bed but had less sleep. This could be due to restlessness, difficulty falling asleep, or interruptions during sleep. Proportionality:

The linear trend suggests that, on average, a greater time in bed correlates with more time asleep, but it's not a perfect one-to-one relationship. The slope of the line indicates the proportion of time in bed that users typically spend asleep. Insights: Users with less efficient sleep may benefit from interventions like improving sleep hygiene or addressing underlying issues like stress or medical conditions. The relationship could help in estimating expected sleep time based on time spent in bed, useful for monitoring sleep quality.

Correlation Between Activity and Sleep

Analyzing the correlation between activity levels, such as steps and active minutes, and sleep quality, measured by total minutes asleep, helps identify how physical activity impacts rest. This relationship can offer insights into promoting better sleep habits through increased activity, aligning with Bellabeat's goal of improving user wellness.

```r
# Add StepCategory as a temporary column to the dataset
step_categories <- merged_data %>%
  mutate(StepCategory = case_when(
    total_steps >= 10000 ~ "Very Active",
    total_steps >= 7500 ~ "Fairly Active",
    total_steps >= 5000 ~ "Lightly Active",
    TRUE ~ "Sedentary"
  ))

# Calculate correlation between sleep and steps for each step category
# For "Very Active"
cor_very_active_steps <- cor(step_categories$total_minutes_asleep[step_categories$StepCategory == "Very
                             step_categories$total_steps[step_categories$StepCategory == "Very Active"]
                             use = "complete.obs")

# For "Fairly Active"
cor_fairly_active_steps <- cor(step_categories$total_minutes_asleep[step_categories$StepCategory == "Fa
                               step_categories$total_steps[step_categories$StepCategory == "Fairly Acti
                               use = "complete.obs")

# For "Lightly Active"
cor_lightly_active_steps <- cor(step_categories$total_minutes_asleep[step_categories$StepCategory == "L
                                step_categories$total_steps[step_categories$StepCategory == "Lightly Ac
                                use = "complete.obs")

# For "Sedentary"
cor_sedentary_steps <- cor(step_categories$total_minutes_asleep[step_categories$StepCategory == "Sedent
                           step_categories$total_steps[step_categories$StepCategory == "Sedentary"],
                           use = "complete.obs")

# Print the correlations
print(paste("Correlation between sleep and very active steps: ", round(cor_very_active_steps, 2)))
```

```
## [1] "Correlation between sleep and very active steps:  0.02"
```

```r
print(paste("Correlation between sleep and fairly active steps: ", round(cor_fairly_active_steps, 2)))
```

```
## [1] "Correlation between sleep and fairly active steps:  -0.02"
```

```r
print(paste("Correlation between sleep and lightly active steps: ", round(cor_lightly_active_steps, 2))
```

```
## [1] "Correlation between sleep and lightly active steps:  -0.14"
```

```r
print(paste("Correlation between sleep and sedentary steps: ", round(cor_sedentary_steps, 2)))
```

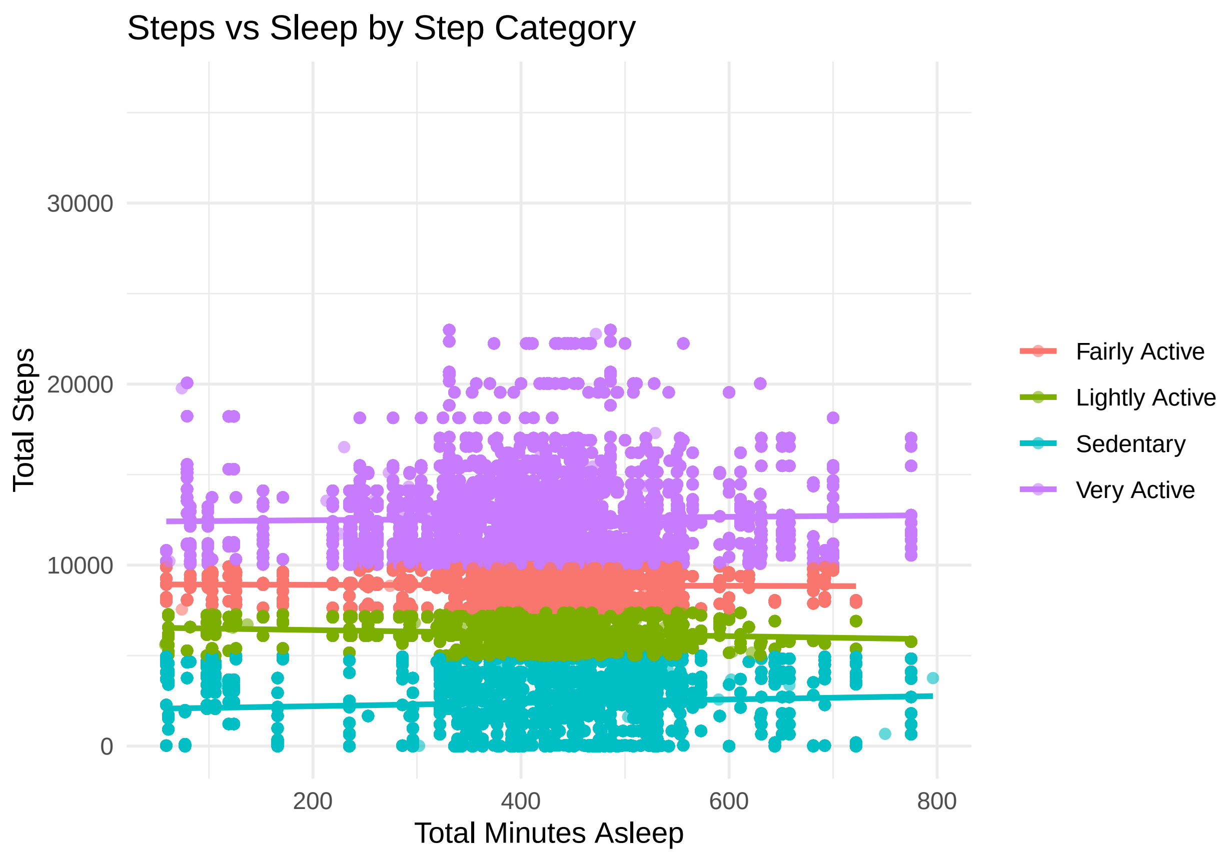
```
## [1] "Correlation between sleep and sedentary steps:  0.08"
```

```r
# Scatter plot for each StepCategory
ggplot(step_categories, aes(x = total_minutes_asleep, y = total_steps, color = StepCategory)) +
  geom_point(alpha = 0.6) +  # Scatter points
  geom_smooth(method = "lm", se = FALSE, aes(group = StepCategory)) +
  labs(title = "Steps vs Sleep by Step Category",
       x = "Total Minutes Asleep",
       y = "Total Steps") +
  theme_minimal() +
  theme(legend.title = element_blank())
```

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

## Warning: Removed 2804 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 2804 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



Observation

1. General Trend : There doesn't appear to be a strong correlation between the number of steps and total minutes asleep. This suggests that the amount of sleep a user gets may not directly impact their daily step count.

2. Activity Levels:

a. Sedentary users tend to have fewer total steps overall, as expected, and their sleep durations are distributed across a wide range.
b. Lightly Active, Fairly Active, and Very Active users show progressively higher step counts. However, their sleep durations are also spread across various levels, indicating no clear pattern connecting higher activity levels with more or less sleep.

3. Category Differentiation: The colors show distinct step ranges for each category, aligning with how step categories were defined (e.g., Very Active users exceeding 10,000 steps daily).

This chart highlights that while activity levels (steps) are distinct among categories, sleep duration remains relatively independent of activity level. This insight can guide Bellabeat to focus on personalizing marketing strategies for activity-based goals rather than sleep patterns alone.

Act Phase

Our objective is to explore how marketing strategies can be improved using insights from smart devices, particularly those provided by the Bellabeat app. We recommend conducting further research with larger datasets, as our sample size was limited and did not consider variables like geographical location.

Recommendations to optimize Bellabeat's marketing strategy and drive engagement.

1. Sleep Duration

Trend Utilization:
Leverage the popularity of personalized health recommendations by integrating sleep tracking with actionable insights through wearable devices.

App Enhancements:
- Add a feature to detect sleep irregularities and provide tailored suggestions like mindfulness exercises or stress-reduction activities.
- Develop an AI-powered "Sleep Coach" to recommend optimal bedtime routines and improve sleep hygiene.

Marketing Strategy:
Highlight Bellabeat's ability to enhance sleep quality with targeted campaigns that emphasize smart device integration and data-backed solutions.

---

2. Distance vs. Calorie Burn

Trend Utilization:
Capitalize on fitness motivation trends by encouraging users to increase activity levels with engaging goals.

App Enhancements:
- Introduce progress-tracking challenges, such as step milestones or distance goals, with badges and rewards.
- Include personalized calorie-burn goals based on users' fitness levels and activity history.

Marketing Strategy:
Position the Bellabeat app as a dynamic fitness companion offering real-time tracking and motivational features. Use testimonials and success stories in marketing campaigns to inspire users.

---

3. Calories Burned vs. Active Minutes

Trend Utilization:
Tap into the demand for real-time feedback and AI-driven fitness tools to enhance user engagement.

App Enhancements:
- Implement AI-powered predictive analytics for calorie burn, enabling users to plan workouts more effectively.
- Add visual data comparisons showing how increased activity impacts calorie burn over time.

Marketing Strategy:
Create interactive advertisements showcasing the app's ability to analyze and predict calorie burn. Focus on attracting fitness enthusiasts and weight management audiences.

---

4. Activity and Sleep Insights

Trend Utilization:
Appeal to users seeking integrated health insights by offering data-driven personalization.

App Enhancements:
- Introduce comparative dashboards linking sleep quality, activity levels, and stress to overall wellness.
- Provide weekly summaries that highlight correlations between user habits like activity, sleep, and stress.

Marketing Strategy:
Segment users into activity-focused and sleep-focused groups for personalized campaigns. Emphasize the app's holistic approach to health and wellness insights.

---

5. Mindfulness and Stress Management

Trend Utilization:
Tap into the growing demand for mindfulness tools and stress management solutions.

App Enhancements:
- Add guided meditation sessions and breathing exercises triggered by real-time stress detection.
- Integrate mood tracking linked to activity and sleep data to offer personalized wellness tips.

Marketing Strategy:
Promote Bellabeat's integration of stress detection, mindfulness, and health tracking features. Focus campaigns on the app's proactive approach to stress reduction and overall wellness.

---

6. Emerging Trends in Smart Device Usage

Focus Areas:
- **Wearable Compatibility:** Ensure the app is compatible with emerging wearable devices like smart rings and glasses to broaden its user base.
- **Sustainability Focus:** Develop eco-friendly wearable devices to align with Bellabeat's commitment to wellness and environmental responsibility.
- **Social Connectivity:** Add social sharing features for fitness milestones and mindfulness habits to boost user engagement and app visibility through community interactions.

In conclusion, Bellabeat can enhance its app and marketing strategies by leveraging emerging trends in health and wellness technology. Key opportunities include integrating actionable insights for sleep tracking, fitness goals, and stress management, supported by AI-driven analytics and personalized recommendations. Enhancing user engagement through progress tracking, social connectivity, and sustainability-focused initiatives will further align with current consumer demands. By emphasizing its holistic approach to health and wellness, Bellabeat can position itself as a leader in the smart device market, driving user adoption and long-term growth.