Bellabeat Case Study: Smart Device Usage Analysis

Amartisoaei Robert

2025-12-17

Contents

Introduction

This case study analyzes smart-device usage data to support **Bellabeat**, a wellness technology company that designs health-focused products for women. The goal of this project is to explore how users engage with their activity trackers and identify behavioral patterns that can inform Bellabeat's product development and marketing strategy.

By examining activity, sleep, and heart-rate data from Fitbit users, I aim to uncover meaningful insights about daily habits, wellness routines, and opportunities for Bellabeat to help users improve their overall health.

This analysis follows the standard analytics process: **Ask, Prepare, Process, Analyze, Share, and Act**.

Ask

Business Task

Bellabeat's goal is to grow as efficiently as possible in the global smart device market by offering products and services tailored to customer needs.

To support this goal, I will analyze smart device usage data to understand how consumers use their devices in their daily lives. The insights from this analysis will help the marketing team design targeted strategies for Bellabeat's products and services based on users' behavior patterns.

Key Stakeholders

- **Urška Sršen** – Cofounder and Chief Creative Officer
- **Sando Mur** – Cofounder and key member of the executive team
- **Bellabeat Marketing Analytics Team** – Responsible for data-driven strategic recommendations

Success Metrics

1. Identify key usage patterns and behavioral differences among users.
2. Present clear and accurate visualizations of activity and habits.
3. Provide at least three actionable recommendations for Bellabeat's marketing strategy.

Prepare

Data Source

The dataset used for this analysis is the **Fitbit Fitness Tracker Data**, a public dataset published on Kaggle by user *Mobius*. It contains personal fitness tracker information from **30 consenting Fitbit users**, including daily and minute-level data regarding physical activity, sleep, and heart rate.

Note: This dataset is not collected by Bellabeat directly. It serves as a **proxy** to understand general smart device usage patterns among users similar to Bellabeat's target audience.

Data Structure

The data is split into multiple CSV files. For this analysis, I focus on the following tables, which are linked by the unique `Id` column:

- **Daily Data:** `daily_activity` (Steps, intensity, calories)
- **Sleep Data:** `minute_sleep` (Sleep logs)
- **Weight Data:** `weight_log` (Weight and BMI)
- **Hourly Data:** `hourly_steps`, `hourly_calories`, `hourly_intensities`
- **Minute/Second Data:** `heartrate_seconds`, `minute_steps`, etc.

ROCCC Assessment

To ensure the data is suitable for analysis, I performed a ROCCC assessment:

Criterion	Status	Assessment
Reliable	Medium	Data is machine-generated (Fitbit), so it is accurate, but the sample size (30 users) is small.
Original	Low	Third-party dataset (Kaggle/Mobius), not collected directly by Bellabeat.
Comprehensive	Medium	Includes deep activity/sleep/heart metrics, but lacks demographic data (age, gender).
Current	Low	Data is several years old; usage habits may have evolved.
Cited	High	Properly cited as "Fitbit Fitness Tracker Data" on Kaggle.

Environment Setup and Data Import

I utilize the **tidyverse** ecosystem for data cleaning, analysis, and visualization. In this step, I load the necessary libraries, define file paths, and import the CSV files into data frames. I also perform immediate checks to verify that user IDs overlap across the main tables, ensuring they can be joined later.

```r
# 1. Load required libraries
library(tidyverse)    # Data wrangling and plotting
library(janitor)      # Cleaning column names
library(ggplot2)      # Visualizations
library(lubridate)    # Date and time manipulation
library(dplyr)        # Data manipulation
library(skimr)        # Quick data summaries
library(naniar)       # Missing value checks


# 2. Set Working Directory & Path
```

```r
data_path <- "E:/Docs/Coursera/Case Study 2_ How-can-a-wellness-technology-company-play-it-smart/R_Proj

# 3. Import Data Frames
# Daily Level
daily_activity     <- read_csv(file.path(data_path, "daily_activity.csv"))
minute_sleep       <- read_csv(file.path(data_path, "minute_sleep.csv"))
weight_log         <- read_csv(file.path(data_path, "weight_log.csv"))

# Hourly Level
hourly_calories    <- read_csv(file.path(data_path, "hourly_calories.csv"))
hourly_intensities <- read_csv(file.path(data_path, "hourly_intensities.csv"))
hourly_steps       <- read_csv(file.path(data_path, "hourly_steps.csv"))

# Minute Level
minute_calories    <- read_csv(file.path(data_path, "minute_calories_narrow.csv"))
minute_intensities <- read_csv(file.path(data_path, "minute_intensities_narrow.csv"))
minute_mets        <- read_csv(file.path(data_path, "minute_METs_narrow.csv"))
minute_steps       <- read_csv(file.path(data_path, "minute_steps_narrow.csv"))
heartrate_seconds  <- read_csv(file.path(data_path, "heartrate_seconds.csv"))

# 4. Initial Data Checks
# Preview structure of key datasets
glimpse(daily_activity)
```

```
## Rows: 457
## Columns: 15
## $ Id                      <dbl> 1503960366, 1503960366, 1503960366, 150396036~
## $ ActivityDate            <chr> "3/25/2016", "3/26/2016", "3/27/2016", "3/28/~
## $ TotalSteps              <dbl> 11004, 17609, 12736, 13231, 12041, 10970, 122~
## $ TotalDistance           <dbl> 7.11, 11.55, 8.53, 8.93, 7.85, 7.16, 7.86, 7.~
## $ TrackerDistance         <dbl> 7.11, 11.55, 8.53, 8.93, 7.85, 7.16, 7.86, 7.~
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ VeryActiveDistance      <dbl> 2.57, 6.92, 4.66, 3.19, 2.16, 2.36, 2.29, 3.3~
## $ ModeratelyActiveDistance <dbl> 0.46, 0.73, 0.16, 0.79, 1.09, 0.51, 0.49, 0.8~
## $ LightActiveDistance     <dbl> 4.07, 3.91, 3.71, 4.95, 4.61, 4.29, 5.04, 3.6~
## $ SedentaryActiveDistance <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.0~
## $ VeryActiveMinutes       <dbl> 33, 89, 56, 39, 28, 30, 33, 47, 40, 15, 43, 3~
## $ FairlyActiveMinutes     <dbl> 12, 17, 5, 20, 28, 13, 12, 21, 11, 30, 18, 18~
## $ LightlyActiveMinutes    <dbl> 205, 274, 268, 224, 243, 223, 239, 200, 244, ~
## $ SedentaryMinutes        <dbl> 804, 588, 605, 1080, 763, 1174, 820, 866, 636~
## $ Calories                <dbl> 1819, 2154, 1944, 1932, 1886, 1820, 1889, 186~
```

```r
glimpse(minute_sleep)
```

```
## Rows: 198,559
## Columns: 4
## $ Id    <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 1503960366, 1503~
## $ date  <chr> "3/13/2016 2:39:30 AM", "3/13/2016 2:40:30 AM", "3/13/2016 2:41:~
## $ value <dbl> 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ logId <dbl> 11114919637, 11114919637, 11114919637, 11114919637, 11114919637,~
```

```r
glimpse(weight_log)
```

```
## Rows: 33
## Columns: 8
## $ Id            <dbl> 1503960366, 1927972279, 2347167796, 2873212765, 2873212~
## $ Date          <chr> "4/5/2016 11:59:59 PM", "4/10/2016 6:33:26 PM", "4/3/20~
## $ WeightKg      <dbl> 53.3, 129.6, 63.4, 56.7, 57.2, 88.4, 92.4, 69.4, 99.7, ~
## $ WeightPounds  <dbl> 117.5064, 285.7191, 139.7731, 125.0021, 126.1044, 194.8~
## $ Fat           <dbl> 22, NA, 10, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ BMI           <dbl> 22.97, 46.17, 24.77, 21.45, 21.65, 25.03, 35.01, 27.14,~
## $ IsManualReport <lgl> TRUE, FALSE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, ~
## $ LogId         <dbl> 1.459901e+12, 1.460313e+12, 1.459728e+12, 1.459987e+12,~
```

```r
glimpse(heartrate_seconds)
```

```
## Rows: 1,154,681
## Columns: 3
## $ Id    <dbl> 2022484408, 2022484408, 2022484408, 2022484408, 2022484408, 2022~
## $ Time  <chr> "4/1/2016 7:54:00 AM", "4/1/2016 7:54:05 AM", "4/1/2016 7:54:10 ~
## $ Value <dbl> 93, 91, 96, 98, 100, 101, 104, 105, 102, 106, 109, 112, 111, 109~
```

```r
# 5. Verify ID Consistency
# Check if users exist across different tables (Essential for joining)
intersect(daily_activity$Id, minute_sleep$Id)
```

```
##  [1] 1503960366 1644430081 1844505072 1927972279 2022484408 2026352035
##  [7] 2347167796 3977333714 4020332650 4319703577 4445114986 4558609924
## [13] 4702921684 5553957443 5577150313 6117666160 6775888955 6962181067
## [19] 7007744171 7086361926 8053475328 8378563200 8792009665
```

```r
intersect(daily_activity$Id, weight_log$Id)
```

```
##  [1] 1503960366 1927972279 2347167796 2873212765 2891001357 4445114986
##  [7] 4558609924 4702921684 6962181067 8253242879 8877689391
```

```r
intersect(daily_activity$Id, heartrate_seconds$Id)
```

```
##  [1] 2022484408 2026352035 2347167796 4020332650 4558609924 5553957443
##  [7] 5577150313 6117666160 6391747486 6775888955 6962181067 7007744171
## [13] 8792009665 8877689391
```

Process

In this phase, I clean and transform the data to ensure it is accurate, consistent, and ready for analysis.

1. Standardization and De-duplication

First, I standardize all column names to `snake_case` using the `janitor` package. This ensures consistency across all tables. Immediately after, I remove any duplicate rows to prevent data inflation.

```
# --- 1. Standardize Column Names ---
daily_activity      <- daily_activity %>% clean_names()
minute_sleep        <- minute_sleep %>% clean_names()
weight_log          <- weight_log %>% clean_names()
hourly_steps        <- hourly_steps %>% clean_names()
hourly_calories     <- hourly_calories %>% clean_names()
hourly_intensities  <- hourly_intensities %>% clean_names()
minute_steps        <- minute_steps %>% clean_names()
minute_calories     <- minute_calories %>% clean_names()
minute_intensities  <- minute_intensities %>% clean_names()
minute_mets         <- minute_mets %>% clean_names()
heartrate_seconds   <- heartrate_seconds %>% clean_names()

# --- 2. Remove Duplicates ---
# Identifying duplicates before removal (optional check)
sum(duplicated(minute_sleep))
```

```
## [1] 525
```

```
sum(duplicated(daily_activity))
```

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

```
# Removing duplicates (specifically found in minute_sleep)
minute_sleep <- minute_sleep %>% distinct()
daily_activity <- daily_activity %>% distinct()
```

2. Date and Time Formatting

Now that the structure is clean, I convert date and time columns from `character` string format to proper `Date` or `Datetime` objects using the `lubridate` library. This allows for time-based analysis later.

```
# --- Daily Data ---
daily_activity <- daily_activity %>%
  mutate(activity_date = mdy(activity_date))

# --- Hourly Data ---
hourly_calories <- hourly_calories %>%
  mutate(activity_hour = mdy_hms(activity_hour))

hourly_intensities <- hourly_intensities %>%
  mutate(activity_hour = mdy_hms(activity_hour))

hourly_steps <- hourly_steps %>%
  mutate(activity_hour = mdy_hms(activity_hour))
```

```r
# --- Minute-Level Data ---
# Batch processing minute-level datasets
minute_calories    <- minute_calories %>% mutate(activity_minute = mdy_hms(activity_minute))
minute_intensities <- minute_intensities %>% mutate(activity_minute = mdy_hms(activity_minute))
minute_mets        <- minute_mets %>% mutate(activity_minute = mdy_hms(activity_minute))
minute_steps       <- minute_steps %>% mutate(activity_minute = mdy_hms(activity_minute))

# --- Sleep & Weight Data ---
minute_sleep <- minute_sleep %>%
  mutate(date = mdy_hms(date))

weight_log <- weight_log %>%
  mutate(date = mdy_hms(date))

# --- Heart Rate Data ---
heartrate_seconds <- heartrate_seconds %>%
  mutate(time = mdy_hms(time))
```

3. Data Integrity and Cleaning

Finally, I handle missing values and remove unrealistic data points (e.g., zero calorie days) that could skew the results.

```r
# --- 1. Filter Invalid Data ---
# Remove days where calories are 0 (likely non-wear days)
daily_activity <- daily_activity %>%
  filter(calories > 0)

# Remove days with 0 steps but >0 very active minutes (impossible data)
daily_activity <- daily_activity %>%
  filter(!(total_steps == 0 & very_active_minutes > 0))

# --- 2. Handle Weight Log Missing Values ---
# We only keep rows where body fat is recorded if specifically analyzing fat, but for general weight tr
# Here, we remove rows with missing fat data as per the original plan.
weight_log <- weight_log %>%
  filter(!is.na(fat))

# --- 3. Final Integrity Check ---
# Verify number of unique users remaining after cleaning
n_distinct(daily_activity$id)
```

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

```r
n_distinct(minute_sleep$id)
```

```
## [1] 23
```

```r
n_distinct(weight_log$id)
```

```
## [1] 2
```

```
n_distinct(heartrate_seconds$id)
```

[1] 14

Summary of the Process Phase

- Standardized: All column names converted to snake_case.
- Cleaned: Duplicate records removed.
- Formatted: Date strings converted to Date/DateTime objects.
- Filtered: Removed unrealistic entries (0 calories) and handled missing values.

Analyze

In this phase, I explore user behavior regarding activity, calories, sleep, and heart rate to uncover patterns relevant to Bellabeat's marketing strategy.
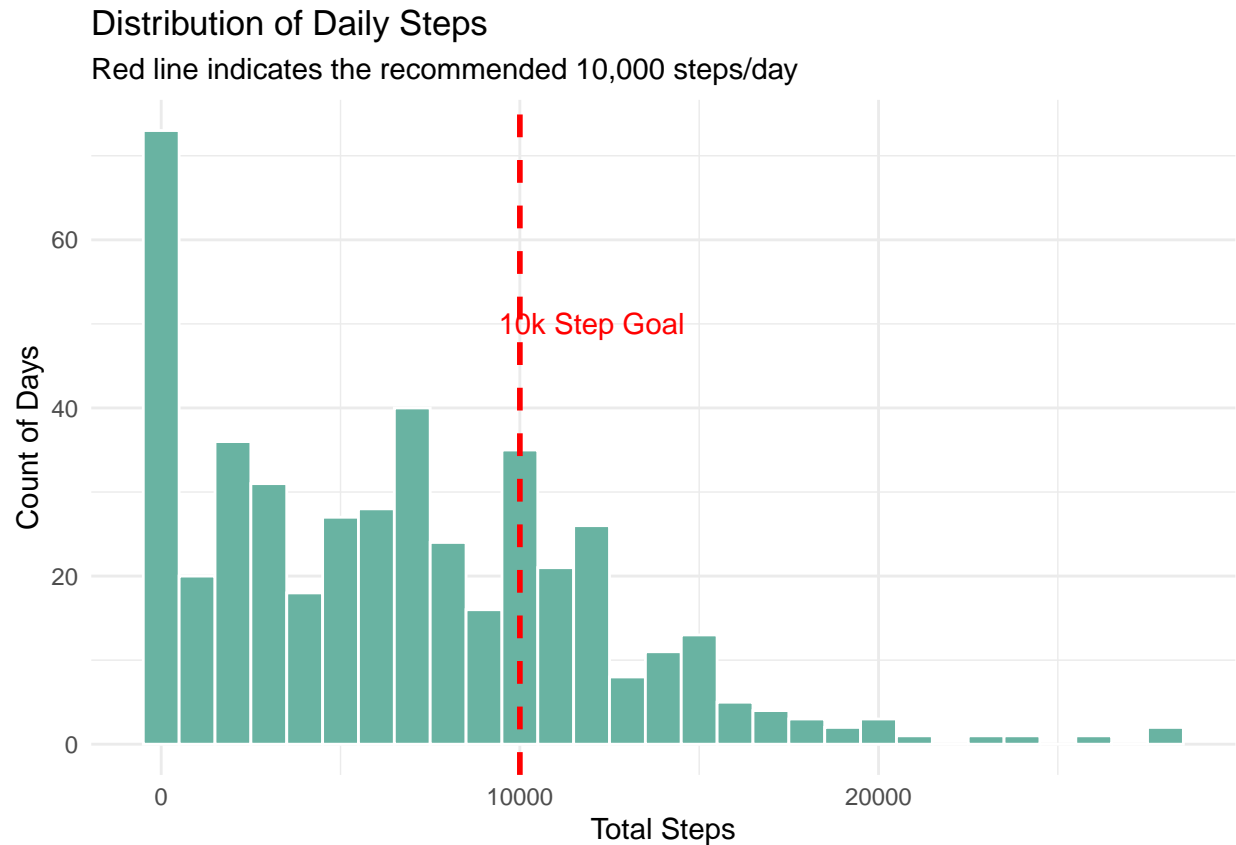
1. Overview of Daily Activity

First, I examine the distribution of daily steps to understand the baseline activity level of the user base.

```
# Summary statistics
daily_activity %>%
  summarise(
    mean_steps = mean(total_steps),
    median_steps= median(total_steps),
    max_steps = max(total_steps)
  )
```

```
## # A tibble: 1 x 3
##   mean_steps median_steps max_steps
##        <dbl>        <dbl>     <dbl>
## 1      6648.         6209     28497
```

```
# Histogram with 10k Step Goal Marker
ggplot(daily_activity, aes(x = total_steps)) +
  geom_histogram(binwidth = 1000, fill = "#69b3a2", color = "white") +
  geom_vline(aes(xintercept = 10000), color = "red", linetype = "dashed", size = 1) +
  annotate("text", x = 12000, y = 50, label = "10k Step Goal", color = "red") +
  theme_minimal() +
  labs(
    title = "Distribution of Daily Steps",
    subtitle = "Red line indicates the recommended 10,000 steps/day",
    x = "Total Steps",
    y = "Count of Days"
  )
```

Distribution of Daily Steps
Red line indicates the recommended 10,000 steps/day

Insights:

The distribution is right-skewed. While some users achieve high activity, a significant portion falls below the 10,000-step recommendation, indicating a potential market for motivation-based features.

2. Activity Intensity: Sedentary vs Active

To understand how users spend their day, I analyze the breakdown of activity intensity.
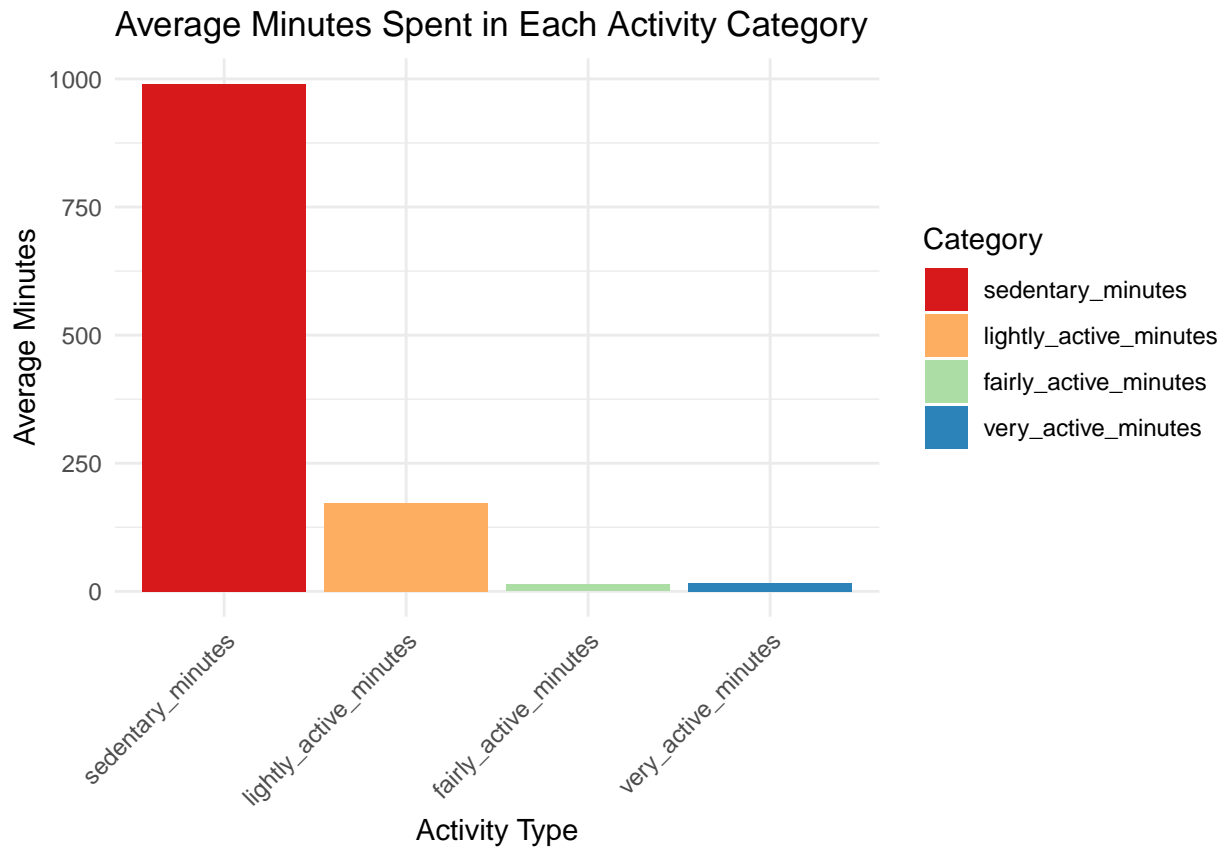
```r
# Reshape data for plotting
activity_long <- daily_activity %>%
  pivot_longer(
    cols = c(very_active_minutes, fairly_active_minutes, lightly_active_minutes, sedentary_minutes),
    names_to = "activity_type",
    values_to = "minutes"
  ) %>%
  mutate(activity_type = factor(activity_type, levels = c("sedentary_minutes", "lightly_active_minutes"

# Bar chart
ggplot(activity_long, aes(x = activity_type, y = minutes, fill = activity_type)) +
  geom_bar(stat = "summary", fun = "mean") +
  scale_fill_brewer(palette = "Spectral") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    title = "Average Minutes Spent in Each Activity Category",
```

```
    x = "Activity Type",
    y = "Average Minutes",
    fill = "Category"
  )
```

Average Minutes Spent in Each Activity Category

Insights:

Users are predominantly sedentary. This highlights a massive opportunity for Bellabeat to introduce "movement breaks" or inactivity alerts.

3. Calorie Burn vs Step Count

I analyze the correlation between steps taken and calories burned to confirm the efficacy of step-based goals.
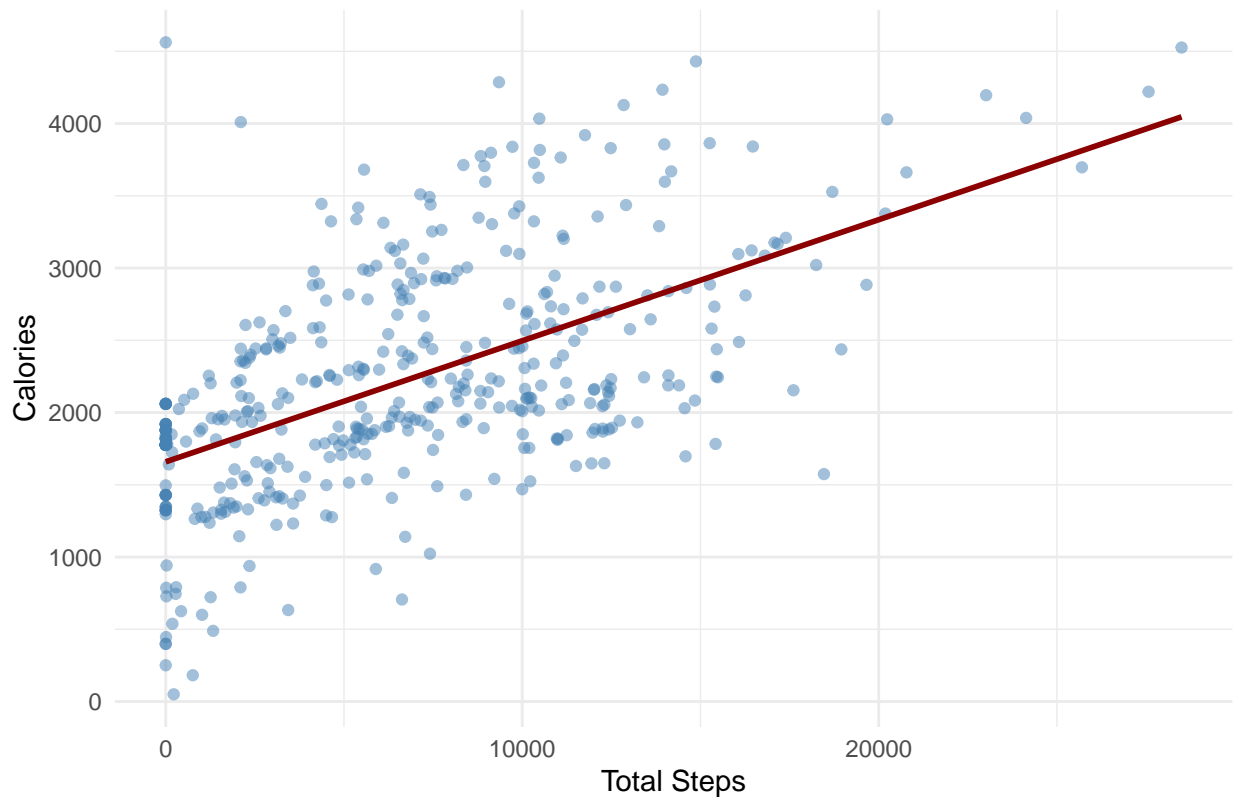
```
ggplot(daily_activity, aes(x = total_steps, y = calories)) +
  geom_point(alpha = 0.5, color = "#4682B4") +
  geom_smooth(method = "lm", color = "darkred", se = FALSE) +
  theme_minimal() +
  labs(
    title = "Relationship: Daily Steps vs Calories Burned",
    x = "Total Steps",
    y = "Calories"
  )
```

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

Relationship: Daily Steps vs Calories Burned

Insights:

As expected, there is a strong positive correlation. This confirms that step-counting is a reliable proxy for calorie expenditure in marketing messaging.

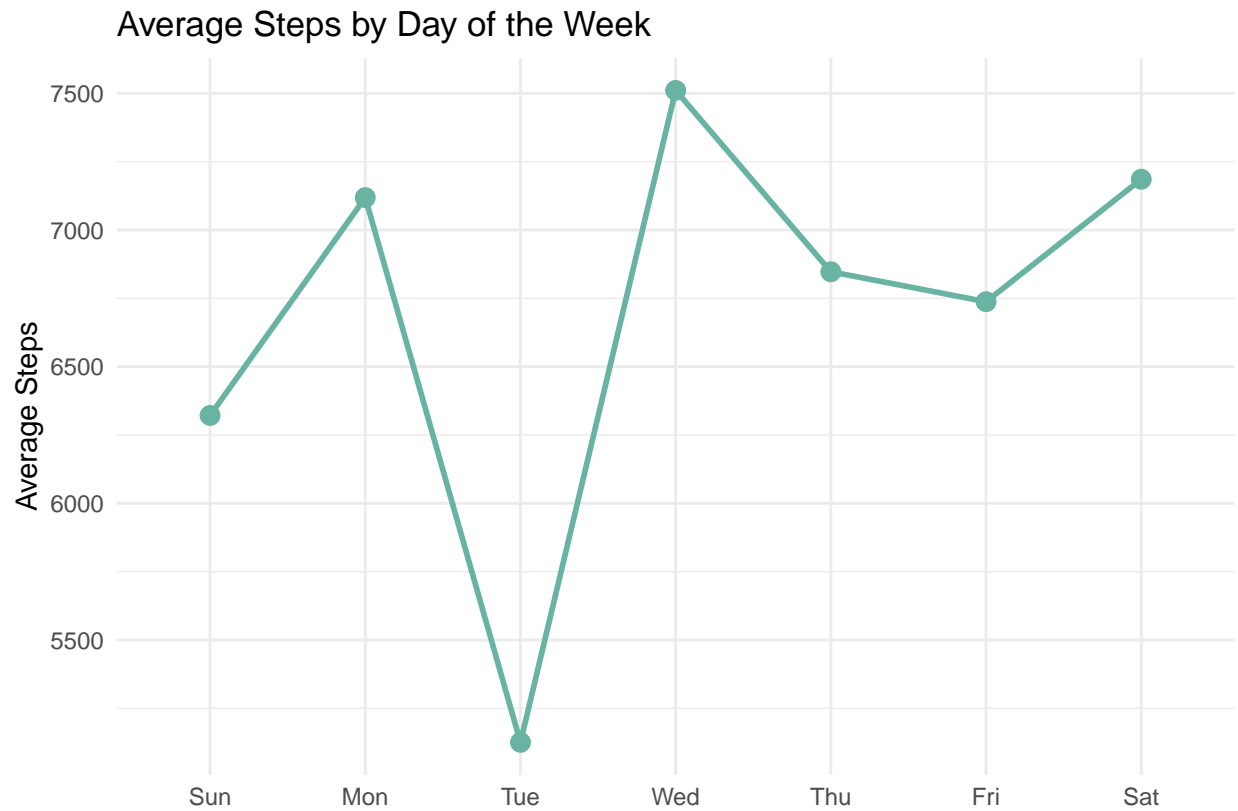## 4. Weekly Activity Habits

Do users slack off on weekends? I visualize activity trends across the week.

```r
# 1. Create weekday column and order it
daily_activity <- daily_activity %>%
  mutate(weekday = wday(activity_date, label = TRUE, abbr = TRUE))

# 2. Aggregate
weekday_steps <- daily_activity %>%
  group_by(weekday) %>%
  summarise(avg_steps = mean(total_steps))

# 3. Plot
ggplot(weekday_steps, aes(x = weekday, y = avg_steps, group = 1)) +
  geom_line(color = "#69b3a2", size = 1) +
  geom_point(color = "#69b3a2", size = 3) +
  theme_minimal() +
  labs(
    title = "Average Steps by Day of the Week",
    y = "Average Steps",
```

```
  x = ""
)
```

Average Steps by Day of the Week

Insights: Activity is relatively consistent, though Tuesday shows a suspicious dip (likely a data syncing issue identified in the cleaning phase) rather than user behavior.
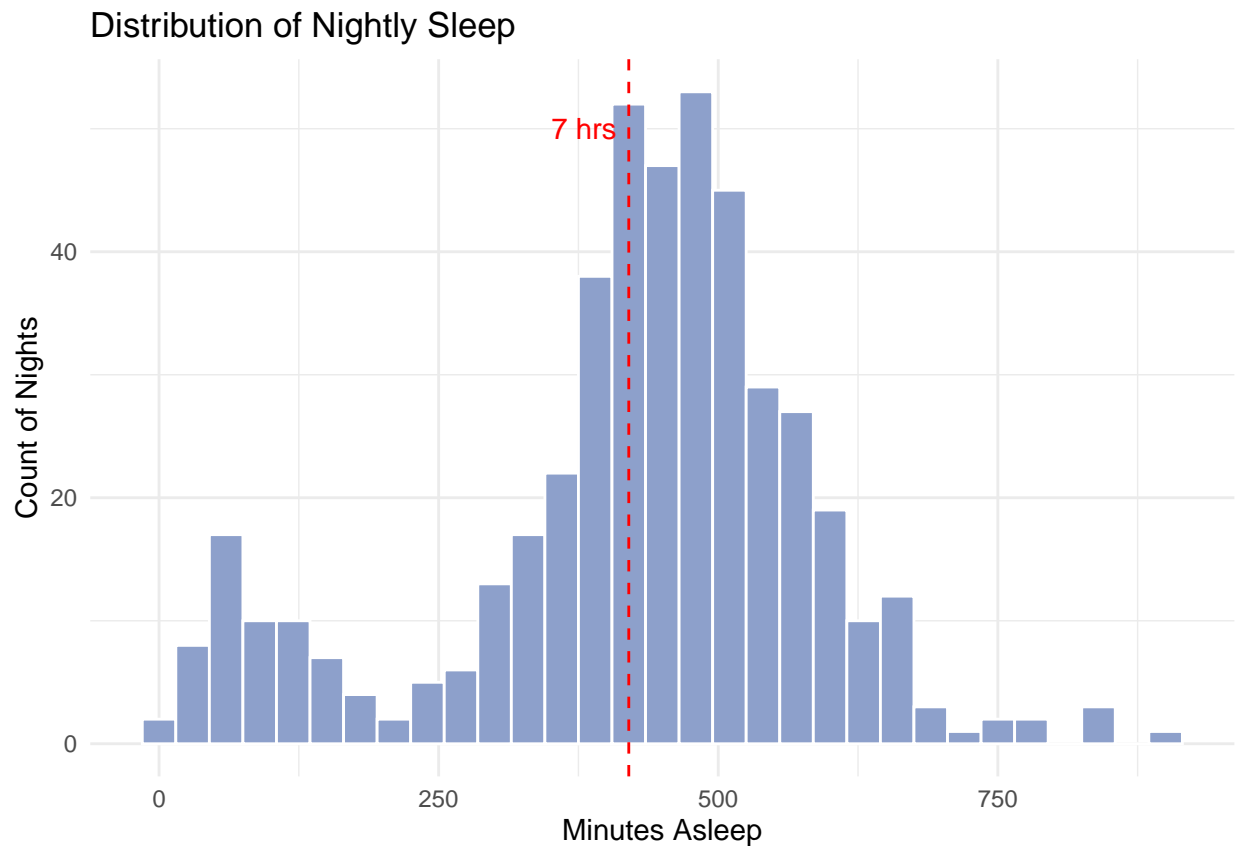
5. Sleep Analysis

I examine sleep duration to see if users are meeting the recommended 7-9 (420-540)

```
# Aggregate minute sleep to daily level
sleep_daily <- minute_sleep %>%
  mutate(sleep_date = as_date(date)) %>%
  group_by(id, sleep_date) %>%
  summarise(minutes_asleep = n(), .groups = "drop")

# Histogram
ggplot(sleep_daily, aes(x = minutes_asleep)) +
  geom_histogram(binwidth = 30, fill = "#8da0cb", color = "white") +
  geom_vline(xintercept = 420, linetype="dashed", color="red") + # 7 hours
  annotate("text", x = 380, y = 50, label = "7 hrs", color = "red") +
  theme_minimal() +
  labs(
    title = "Distribution of Nightly Sleep",
    x = "Minutes Asleep",
```

```
    y = "Count of Nights"
  )
```

Distribution of Nightly Sleep

Insights: Most nights fall roughly between 4 to 7 hours of sleep. This suggests that many users may not reach the commonly recomended 7-9 hours of sleep, indicating sleep improvement as key recommendation.
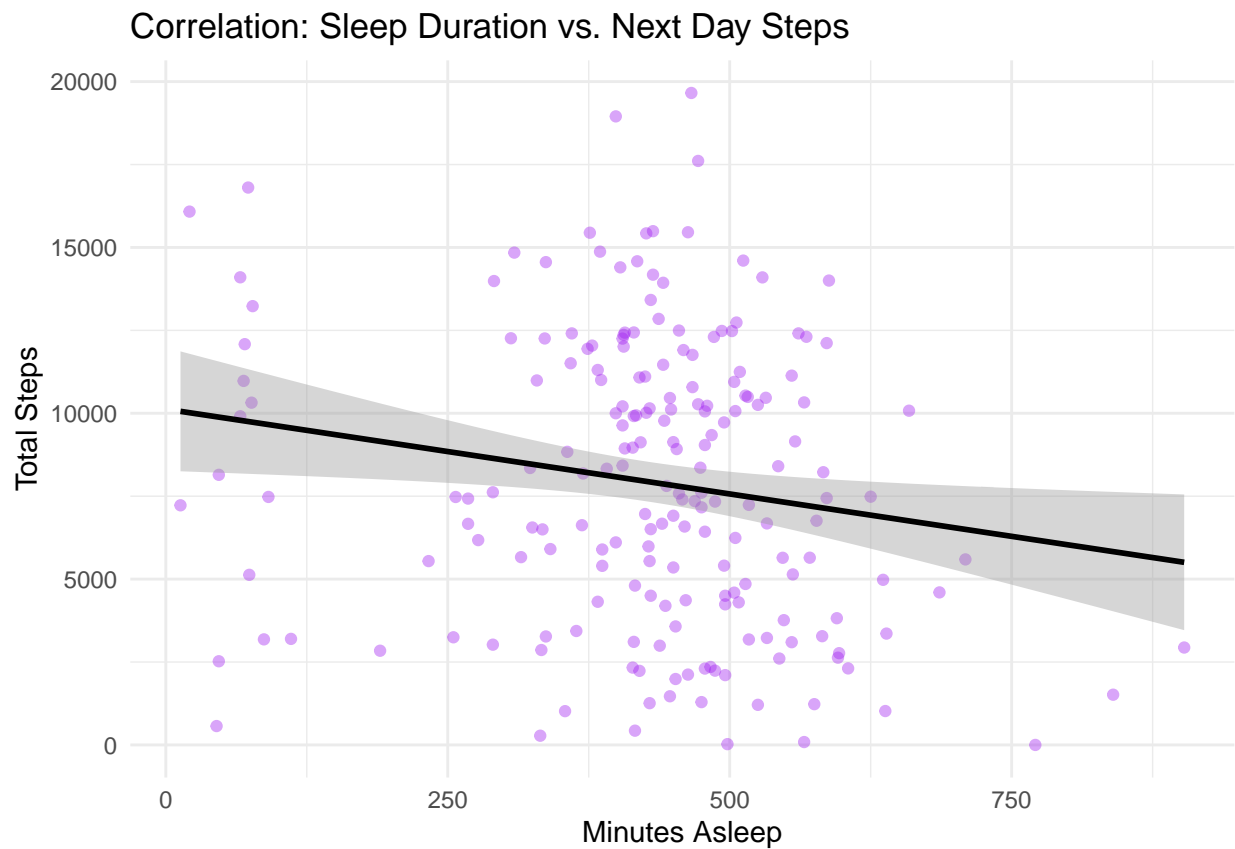
6. Sleep vs Activity

Does sleeping more lead to more activity the next day? I merge the datasets to find out.

```
# Join tables
daily_activity_sleep <- daily_activity %>%
  rename(sleep_date = activity_date) %>%
  inner_join(sleep_daily, by = c("id", "sleep_date"))

# Scatter plot
ggplot(daily_activity_sleep, aes(x = minutes_asleep, y = total_steps)) +
  geom_point(alpha = 0.4, color = "purple") +
  geom_smooth(method = "lm", color = "black") +
  theme_minimal() +
  labs(
    title = "Correlation: Sleep Duration vs. Next Day Steps",
    x = "Minutes Asleep",
    y = "Total Steps"
  )
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

Correlation: Sleep Duration vs. Next Day Steps

Insights: The relationship is weak (flat trend line). This suggests that simply sleeping more doesn't automatically result in more movement; users need separate nudges for both behaviors.

User Sementation

Finally, I categorize days into activity levels to quantify how often users are "active enough."

```
# 1. Categorize
daily_activity <- daily_activity %>%
  mutate(activity_level = case_when(
    total_steps < 5000 ~ "Sedentary",
    total_steps < 10000 ~ "Lightly Active",
    TRUE ~ "Active"
  )) %>%
  mutate(activity_level = factor(activity_level, levels = c("Sedentary", "Lightly Active", "Active")))

# 2. Plot
ggplot(daily_activity, aes(x = activity_level, fill = activity_level)) +
  geom_bar() +
  scale_fill_manual(values = c("Sedentary" = "#e76f51", "Lightly Active" = "#e9c46a", "Active" = "#2a9d8
  theme_minimal() +
  labs(
```
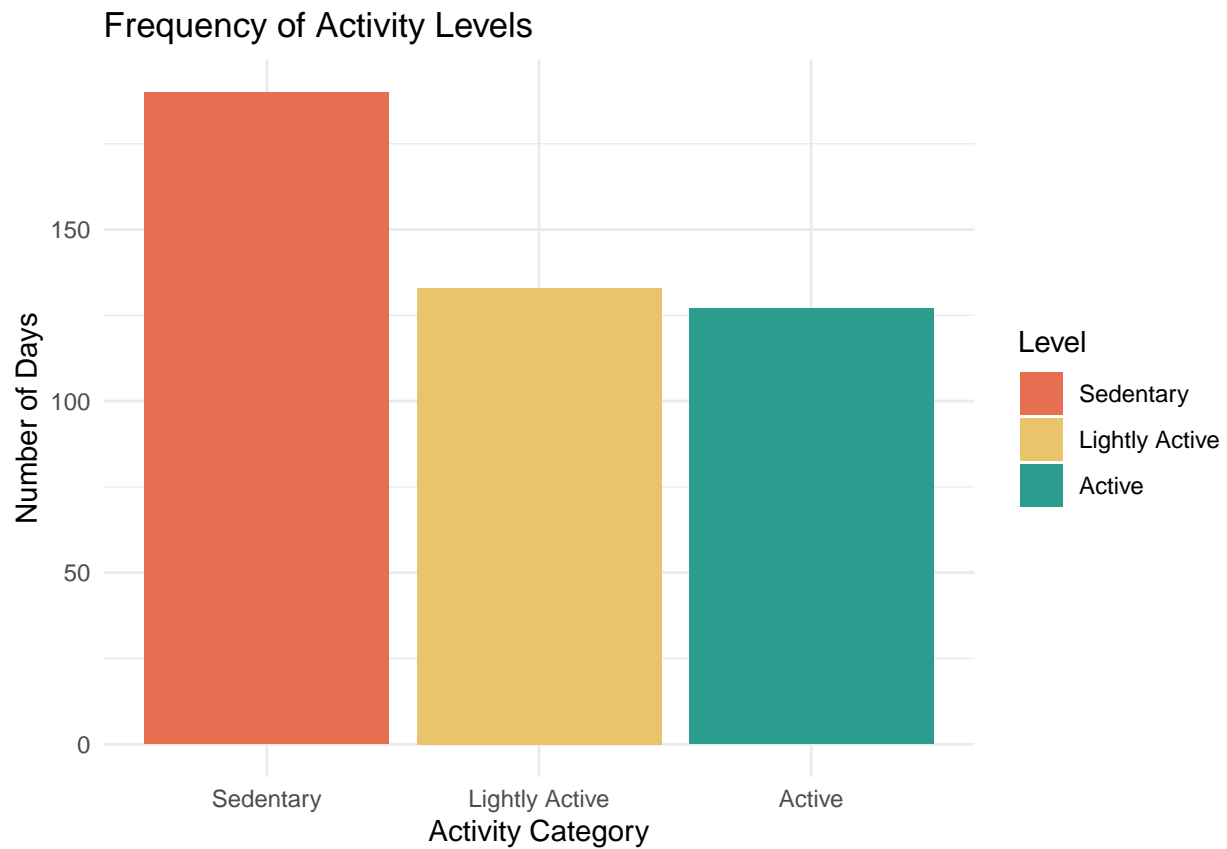
```
  title = "Frequency of Activity Levels",
  x = "Activity Category",
  y = "Number of Days",
  fill = "Level"
)
```

Frequency of Activity Levels

Share

Key Findings & Insights

Based on the analysis of 30 Fitbit users, I have identified six key trends that directly impact Bellabeat's marketing strategy.

1. Sedentary Lifestyles Prevail

- **Observation:** The majority of users do not reach the recommended 10,000 steps per day.
- **Data:** Most days fall into the **"Sedentary"** or **"Lightly Active"** categories.
- **Implication:** There is a significant market opportunity for features that encourage small, incremental movements rather than intense athleticism.

2. Steps Drive Calorie Burn

- **Observation:** There is a strong, positive correlation between daily steps and calories burned.

- **Implication:** Step-based goals are a scientifically valid and easily understood metric for users aiming to manage their weight.

3. **High Sedentary Time**

   - **Observation:** Users spend a disproportionate amount of their waking hours in "Sedentary Minutes," with very limited "Very Active" time.
   - **Implication:** Users may not be aware of how inactive they are during the workday, suggesting a need for "smart nudges."

4. **Sleep Deprivation is Common**

   - **Observation:** A large portion of sleep records show durations between **4 to 7 hours**, falling short of the recommended 7–9 hours.
   - **Implication:** Sleep wellness is a major area for improvement where Bellabeat can offer unique value through coaching.

5. **Sleep Does Not Guarantee Activity**

   - **Observation:** There is no strong correlation between sleep duration and the next day's step count.
   - **Implication:** Rest alone isn't enough to motivate movement. Bellabeat needs to treat Sleep and Activity as two separate pillars that require distinct motivation strategies.

6. **Data Consistency Gaps**

   - **Observation:** We observed unusual dips in activity (specifically on Tuesdays) and zero-calorie days.
   - **Implication:** These are likely data syncing issues or non-wear times. Improvements in device comfort or battery life could reduce these data gaps.

Act

Recommendations

Based on the data findings and Bellabeat's goal of empowering women through wellness, I propose the following actionable strategies:

1. **Product & App Features**

   - **"Sedentary Nudges":** Since users spend significant time inactive, implement "smart alerts" that trigger after 60 minutes of inactivity. These should suggest short, achievable actions (e.g., "Time for a 2-minute stretch").
   - **Daily Wellness Score:** Develop a composite metric that combines **Steps + Sleep + Active Minutes**. This gives users a holistic view of their health rather than focusing solely on weight or calories, aligning with Bellabeat's brand identity.
   - **Sleep Coaching:** Given the prevalence of insufficient sleep (4–7 hours), introduce a "Sleep Hygiene" module. This could include bedtime reminders and visualization of how sleep consistency impacts their "Wellness Score."
   - **Sync Reminders:** To address data gaps (like the Tuesday dip), add push notifications encouraging users to sync their device if no data is detected by mid-day.

2. Marketing Strategy

- **Campaign: "Small Steps, Big Impact":** Use the strong step-calorie correlation to market the idea that users don't need to be athletes to get fit. Focus messaging on hitting the 10k mark through casual walking.
- **Campaign: "Rest to Recharge":** Position Bellabeat devices not just as activity trackers, but as "Rest Managers." Highlight the sleep tracking features to appeal to the demographic that feels overworked and sleep-deprived.

Limitations

To maintain analytical rigor, it is important to acknowledge the limitations of this case study: * **Sample Size:** Only 30 users were analyzed, which may not be statistically representative of the global population. * **Demographics:** The dataset lacks gender and age information. As Bellabeat focuses on women, assuming these users are female proxies introduces risk. * **Data Age:** The dataset is several years old; post-pandemic wearable usage behavior may differ significantly.

Future Work

To refine these recommendations, Bellabeat should consider:

1. **First-Party Data Collection:** Launch an opt-in study with current Bellabeat users to gather demographic-specific data.
2. **Longitudinal Analysis:** Analyze usage over a full year to identify seasonal trends (e.g., do users drop off in winter?).
3. **Qualitative Research:** Conduct user surveys to understand *why* users stop wearing the device on certain days.

Report generated on: 2025-12-17 **Author:** Amartisoaei Robert