

Bellabeat Case Study: Smart Device Usage Analysis

Google Data Analytics Capstone Project

Amartisoaei Robert

2026-01-26

Contents

Introduction	2
Ask	2
Prepare	2
Data Source	2
Data Structure	3
ROCCC Assessment	3
Environment Setup and Data Import	3
Process	6
1. Daily Level Data & Feature Selection	6
2. Granular Timestamp Parsing & Normalization	7
3. Physiological Data Aggregation	7
4. Verification & Validation	8
Analyze	8
1. Overview of Daily Activity	8
2. Activity Intensity: Sedentary vs Active	9
3. Calorie Burn vs Step Count	10
4. Weekly Activity Habits	11
5. Sleep Analysis	12
6. Sleep vs Activity	13
User Segmentation	14
Share	15
Key Findings & Insights	15

Act	16
Recommendations	16
Limitations	17
Future Work	17

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 **SQL** for data cleaning, and **tidyverse** ecosystem for 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.

```
# 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(nanar)         # Missing value checks

# Define a custom theme for consistency
```

```

theme_bellabeat <- function() {
  theme_minimal() +
  theme(
    text = element_text(family = "sans", color = "#333333"),
    plot.title = element_text(face = "bold", size = 16, margin = margin(b = 10)),
    plot.subtitle = element_text(size = 12, color = "#666666", margin = margin(b = 10)),
    panel.grid.minor = element_blank(),
    panel.grid.major.x = element_blank(),
    legend.position = "top",
    axis.text = element_text(size = 10),
    axis.title = element_text(face = "bold")
  )
}

# 2. Set Working Directory & Path
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")) %>% clean_names()
minute_sleep   <- read_csv(file.path(data_path, "minute_sleep.csv")) %>% clean_names()
weight_log     <- read_csv(file.path(data_path, "weight_logs.csv")) %>% clean_names()

# Hourly Data
hourly_steps      <- read_csv(file.path(data_path, "hourly_steps.csv")) %>% clean_names()
hourly_calories   <- read_csv(file.path(data_path, "hourly_calories.csv")) %>% clean_names()
hourly_intensities <- read_csv(file.path(data_path, "hourly_intensities.csv")) %>% clean_names()

# Minute Level
minute_steps      <- read_csv(file.path(data_path, "minute_steps.csv")) %>% clean_names()
minute_calories   <- read_csv(file.path(data_path, "minute_calories.csv")) %>% clean_names()
minute_intensities <- read_csv(file.path(data_path, "minute_intensities.csv")) %>% clean_names()
minute_mets        <- read_csv(file.path(data_path, "minute_MET.csv")) %>% clean_names()

# Heartrate Data
heartrate_seconds <- read_csv(file.path(data_path, "heartrate_seconds.csv")) %>% clean_names()

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

## Rows: 450
## Columns: 13
## $ id                      <dbl> 1644430081, 1644430081, 2022484408, 2022484~
## $ activity_date            <date> 2016-04-06, 2016-04-02, 2016-04-10, 2016-0~
## $ total_steps               <dbl> 8046, 20237, 10005, 15392, 0, 10226, 8426, ~
## $ total_distance             <dbl> 5.85, 14.71, 7.02, 10.93, 0.00, 6.84, 6.16, ~
## $ very_active_distance       <dbl> 1.29, 2.57, 0.08, 3.43, 0.00, 0.00, 0.13, 0~
## $ moderately_active_distance <dbl> 2.12, 6.40, 1.05, 1.75, 0.00, 4.44, 5.49, 2~
## $ light_active_distance      <dbl> 2.45, 5.73, 5.90, 5.75, 0.00, 2.40, 0.54, 6~
## $ sedentary_active_distance   <dbl> 0.00, 0.01, 0.00, 0.00, 0.00, 0.00, 0.00, 0~
## $ very_active_minutes         <dbl> 18, 34, 1, 47, 0, 0, 2, 3, 8, 40, 25, 25, 4~
## $ fairly_active_minutes       <dbl> 41, 141, 23, 42, 660, 101, 120, 57, 56, 26, ~

```

```
## $ lightly_active_minutes      <dbl> 137, 347, 339, 258, 0, 174, 25, 323, 381, 2~  
## $ sedentary_minutes          <dbl> 1244, 918, 1077, 1093, 780, 685, 1293, 471, ~  
## $ calories                   <dbl> 2925, 4029, 2458, 2733, 4562, 1524, 2360, 3~
```

```
glimpse(minute_sleep)
```

```
## Rows: 198,034  
## Columns: 4  
## $ id              <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 150396036~  
## $ sleep_time     <chr> "2016-03-13 03:35:30 UTC", "2016-03-13 03:50:30 UTC", "20~  
## $ sleep_status   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~  
## $ log_id          <dbl> 11114919637, 11114919637, 11114919637, 11114919637, 11114~
```

```
glimpse(weight_log)
```

```
## Rows: 33  
## Columns: 7  
## $ id              <dbl> 8877689391, 8877689391, 8877689391, 8877689391, 88776893~  
## $ date            <chr> "2016-04-08 06:39:44 UTC", "2016-04-07 06:15:08 UTC", "2~  
## $ weight_kg       <dbl> 85.0, 85.1, 85.4, 85.5, 85.8, 86.0, 86.1, 86.3, 86.6, 12~  
## $ weight_pounds   <dbl> 187.3929, 187.6134, 188.2748, 188.4952, 189.1566, 189.59~  
## $ bmi              <dbl> 25.44, 25.49, 25.56, 25.61, 25.68, 25.76, 25.79, 25.83, ~  
## $ manual_report    <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~  
## $ log_id           <dbl> 1.460098e+12, 1.460010e+12, 1.460189e+12, 1.459493e+12, ~
```

```
glimpse(heartrate_seconds)
```

```
## Rows: 1,154,681  
## Columns: 3  
## $ id              <dbl> 6391747486, 6391747486, 2026352035, 6391747486, 6391747486, ~  
## $ time            <chr> "2016-04-05 21:50:55 UTC", "2016-04-08 05:17:25 UTC", "2016~  
## $ heart_rate      <dbl> 111, 60, 63, 69, 73, 73, 75, 77, 77, 78, 78, 79, 79, 80, 80~
```

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

```
## NULL
```

```
intersect(daily_activity$id, weight_log$id)
```

```
## NULL
```

```
intersect(daily_activity$id, heartrate_seconds$id)
```

```
## NULL
```

```
# Verify User ID overlaps  
n_distinct(daily_activity$id)
```

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

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

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

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

```
## [1] 11
```

Process

Tool Selection: SQL (BigQuery) was utilized for data extraction, cleaning, and strict schema validation due to the dataset's size and the necessity for granular timestamp manipulation. R is reserved for the subsequent Exploratory Data Analysis (EDA) and visualization phases.

Data Integrity & Cleaning Log: The following processing steps were executed to resolve data quality issues identified during the “Prepare” phase.

1. Daily Level Data & Feature Selection

- Logic Applied:

- `daily_activity`: Filtered out records where `Calories = 0` (indicating the device was not worn).
- `weight_log`: Excluded the `Fat` column due to >90% null values (feature selection).
- **Standardization:** Converted CamelCase headers to snake_case.

```
CLEANING SAMPLE: DAILY ACTIVITY & WEIGHT  
-- Daily Activity: Remove non-wear days  
CREATE OR REPLACE TABLE `bellabeat_clean.daily_activity` AS  
SELECT  
    Id AS id,  
    ActivityDate AS activity_date,  
    TotalSteps AS total_steps,  
    Calories AS calories  
  
    -- [Additional columns truncated for brevity]  
FROM `bellabeat_raw.daily_activity`  
WHERE Calories > 0 AND NOT (TotalSteps = 0 AND VeryActiveMinutes > 0);
```

```
-- Weight Log: Drop 'Fat' column due to nulls  
CREATE OR REPLACE TABLE `bellabeat_clean.weight_logs` AS  
SELECT DISTINCT  
    Id AS id,  
    PARSE_TIMESTAMP('%m/%d/%Y %I:%M:%S %p', TRIM(Date)) AS date,
```

```

WeightKg AS weight_kg,
BMI AS BMI,
IsManualReport AS manual_report
FROM `bellabeat_raw.weight_logs`;
```

2. Granular Timestamp Parsing & Normalization

Objective: Standardize time-series data across Hourly and Minute-level datasets.

- Logic Applied:

- **String Parsing:** Applied TRIM() and PARSE_TIMESTAMP() to correct formatting inconsistencies in raw string dates.
- **Deduplication:** Applied DISTINCT across all granular tables (hourly_steps, minute_sleep, minute_METs, etc.) to ensure unique composite keys (User ID + Time).
- **Unit Conversion:** Converted minute_METs by dividing by 10 to retrieve the actual metabolic equivalent.

SQL Query Used:

```

CLEANING SAMPLE: TIMESTAMP PARSING (HOURLY/MINUTE)
-- Example: Minute METs (Parsing + Unit Conversion)
CREATE OR REPLACE TABLE `bellabeat_clean.minute_MET` AS
SELECT DISTINCT
    Id AS id,
    PARSE_TIMESTAMP('%m/%d/%Y %I:%M:%S %p', TRIM(ActivityMinute)) AS activity_minute,
    SAFE_CAST(METs AS FLOAT64)/10 AS METs
FROM `bellabeat_raw.minute_MET`;

-- Example: Hourly Steps (Trimming whitespace issues)
CREATE OR REPLACE TABLE `bellabeat_clean.hourly_steps` AS
SELECT DISTINCT
    Id AS id,
    PARSE_TIMESTAMP('%m/%d/%Y %I:%M:%S %p', TRIM(ActivityHour)) AS activity_hour,
    StepTotal AS total_steps
FROM `bellabeat_raw.hourly_steps`;
```

3. Physiological Data Aggregation

Objective: Optimize heart rate data for analysis by reducing granularity without losing trend fidelity.

- Logic Applied:

- **Downsampling:** Raw heart rate data (heartrate_seconds) contained second-level noise. Data was aggregated to minute-level averages to align with minute_intensities and minute_steps.

```

CLEANING SAMPLE: AGGREGATION
-- Aggregating second-level heart rate to minute-level
CREATE TABLE `bellabeat_clean.heartrate_min_avg` AS
```

```

SELECT
    id,
    DATETIME_TRUNC(time, MINUTE) AS heart_rate_minute,
    AVG(heart_rate) AS avg_heart_rate
FROM `bellabeat_clean.heartrate_seconds`
GROUP BY id, heart_rate_minute;

```

4. Verification & Validation

Post-processing, the following validation checks were run against all clean tables:

1. **Row Counts:** Confirmed sample size consistency between Raw and Clean tables.
2. **Null Checks:** Verified zero NULLs in primary keys (`id`, `activity_date/time`).
3. **Logical Consistency:** Verified `TotalIntensity` vs `AverageIntensity` mathematical accuracy.
4. **Anomaly Detection:** Confirmed minute-level calories range (Min: 0, Max: >0) and sleep states (1, 2, 3).

Outcome: The data is now stored in the `bellabeat_clean` SQL dataset. It is structurally consistent, free of duplicates, and formatted for direct ingestion into R.

Summary of the Process Phase

- Tools Used: SQL (BigQuery) for cleaning; R (Tidyverse) for analysis.
- Key Actions: Removed duplicates, parsed dates into standard formats, and filtered out non-wear days (Zero Calories).
- Outcome: Clean, standardized datasets ready for exploration.

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.

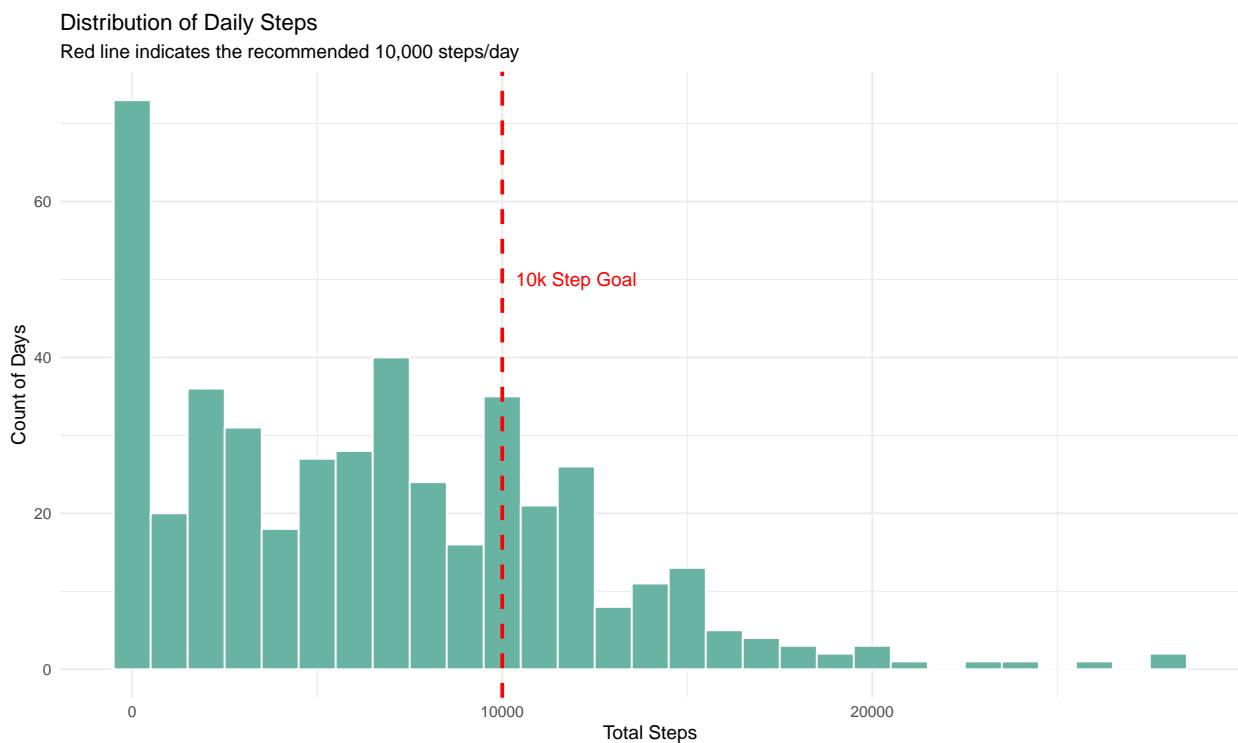
```

# 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"
)
```



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.

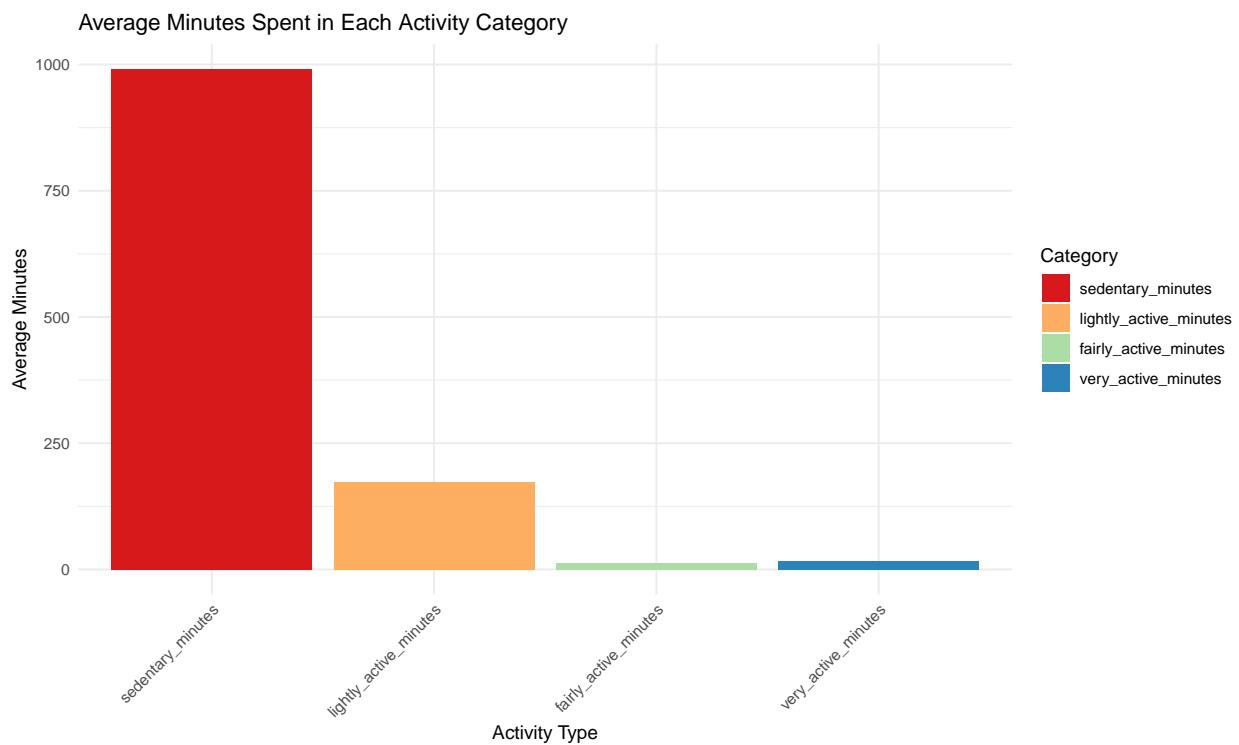
```
# Reshape data for plotting
activity_long <- daily_activity %>%
  pivot_longer(
    cols = c(very_active_minutes, fairly_active_minutes, lightly_active_minutes, sedentary_minutes),
    names_to = "activity_type",
    values_to = "minutes"
```

```

) %>%
  mutate(activity_type = factor(activity_type, levels = c("sedentary_minutes", "lightly_active_minutes",
  "fairly_active_minutes", "very_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"
)

```



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.

```

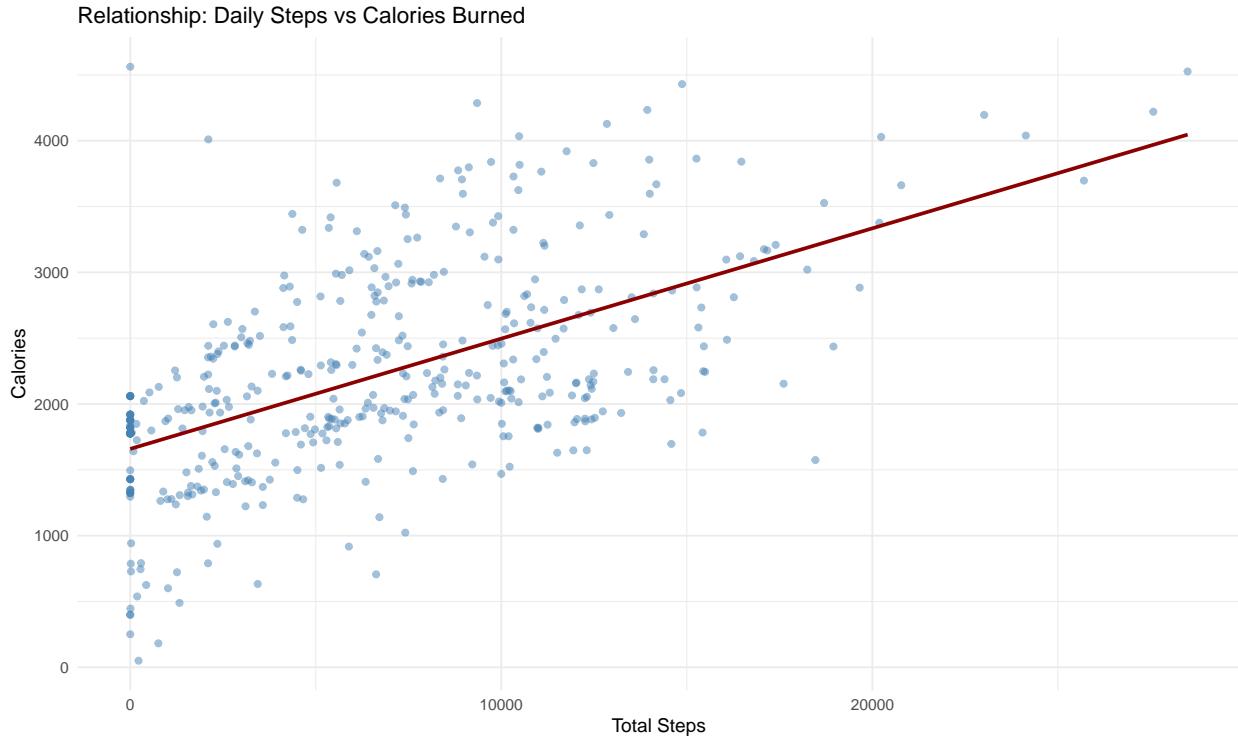
ggplot(daily_activity, aes(x = total_steps, y = calories)) +
  geom_point(alpha = 0.5, color = "#4682B4") +
  geom_smooth(method = "lm", color = "darkred", se = FALSE) +

```

```

theme_minimal() +
labs(
  title = "Relationship: Daily Steps vs Calories Burned",
  x = "Total Steps",
  y = "Calories"
)

```



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.

```

# 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))

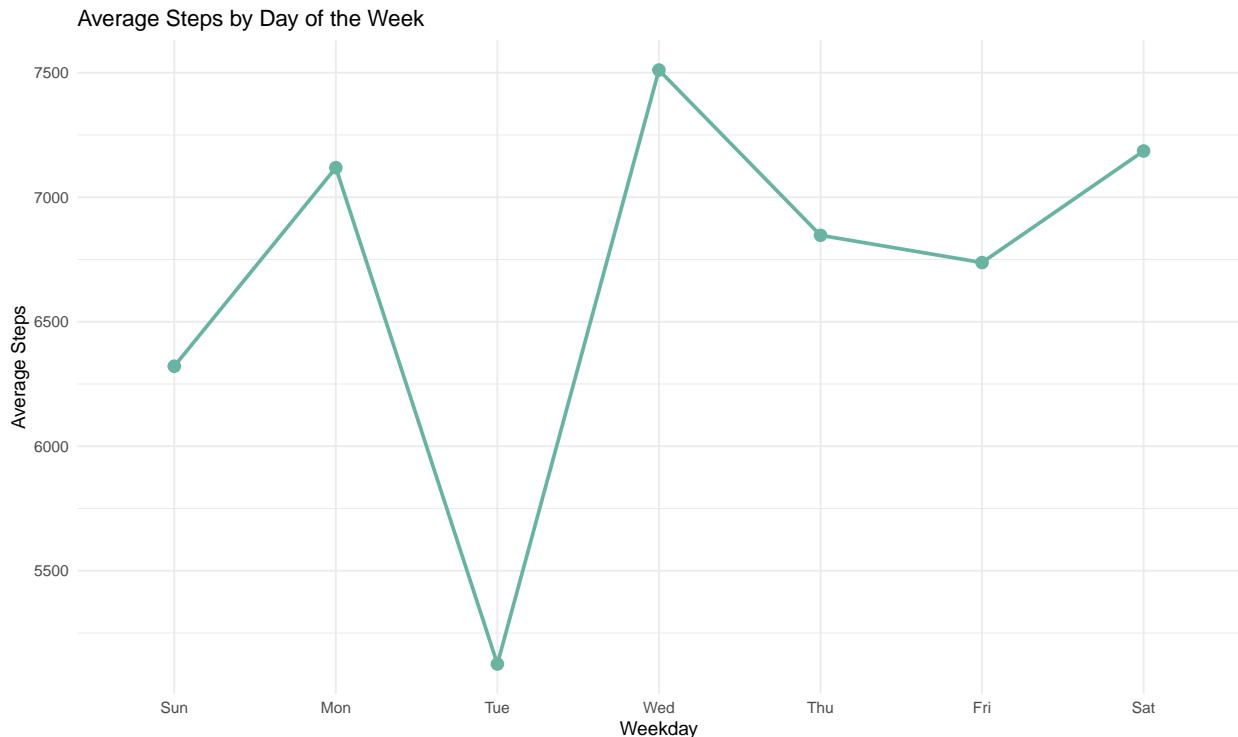
# 3. Plot
ggplot(weekday_steps, aes(x = weekday, y = avg_steps, group = 1)) +
  geom_line(color = "#69b3a2", size = 1) +

```

```

geom_point(color = "#69b3a2", size = 3) +
theme_minimal() +
labs(
  title = "Average Steps by Day of the Week",
  y = "Average Steps",
  x = "Weekday"
)

```



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 mins.)

```

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

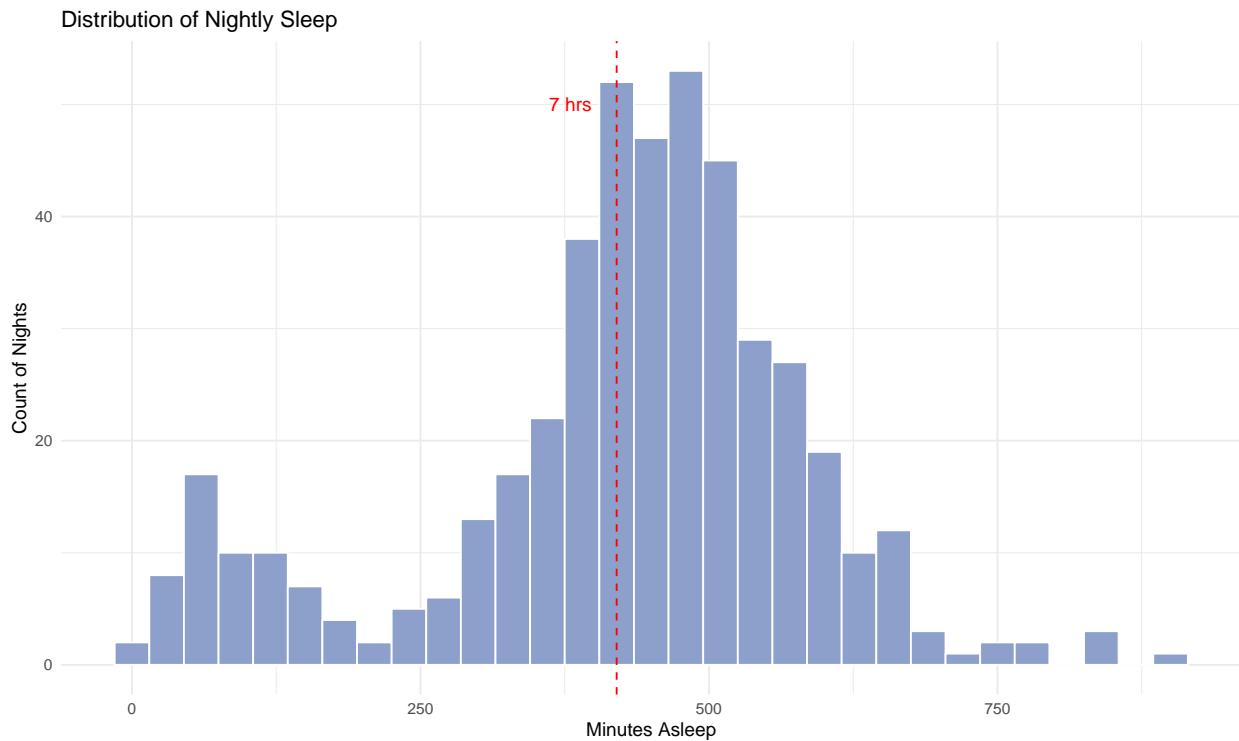
# Histogram
ggplot(sleep_daily, aes(x = minutes_asleep)) +
  geom_histogram(binwidth = 30, fill = "#8da0cb", color = "white") +
  geom_vline(xintercept = 420, linetype="dashed", color="red") + # 7 hours
  annotate("text", x = 380, y = 50, label = "7 hrs", color = "red") +
  theme_minimal() +

```

```

  labs(
    title = "Distribution of Nightly Sleep",
    x = "Minutes Asleep",
    y = "Count of Nights"
  )

```



Insights:

- The distribution of sleep duration is centered slightly **above the 7-hour (420-minute) mark**. This suggests that for the majority of recorded nights, the subjects are achieving a healthy amount of sleep (typically between 7 and 8.5 hours).
- There is a small, distinct cluster of records on the far left (under 200 minutes or ~3 hours). These likely represent naps or brief periods of rest, rather than full nights of sleep, distinct from the main sleeping pattern.

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"))

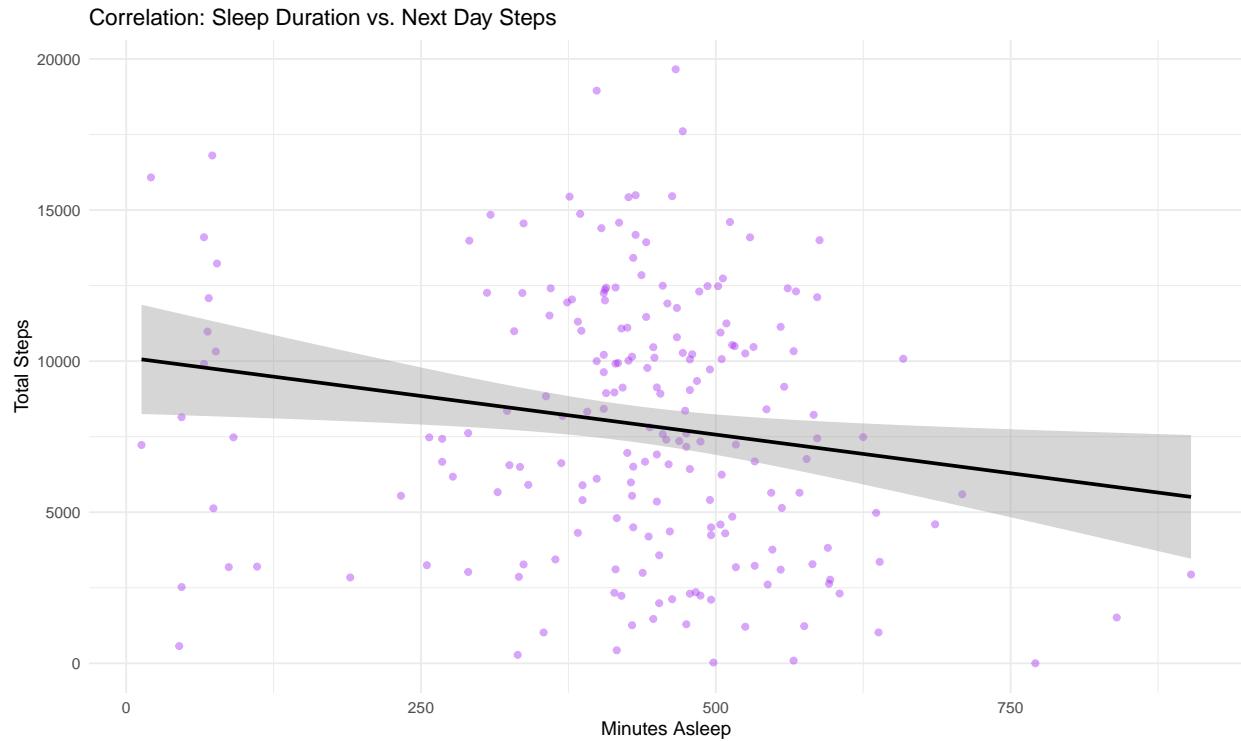
# Scatter plot
ggplot(daily_activity_sleep, aes(x = minutes_asleep, y = total_steps)) +
  geom_point(alpha = 0.4, color = "purple") +
  geom_smooth(method = "lm", color = "black") +

```

```

theme_minimal() +
labs(
  title = "Correlation: Sleep Duration vs. Next Day Steps",
  x = "Minutes Asleep",
  y = "Total Steps"
)

```



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 Segmentation

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")))

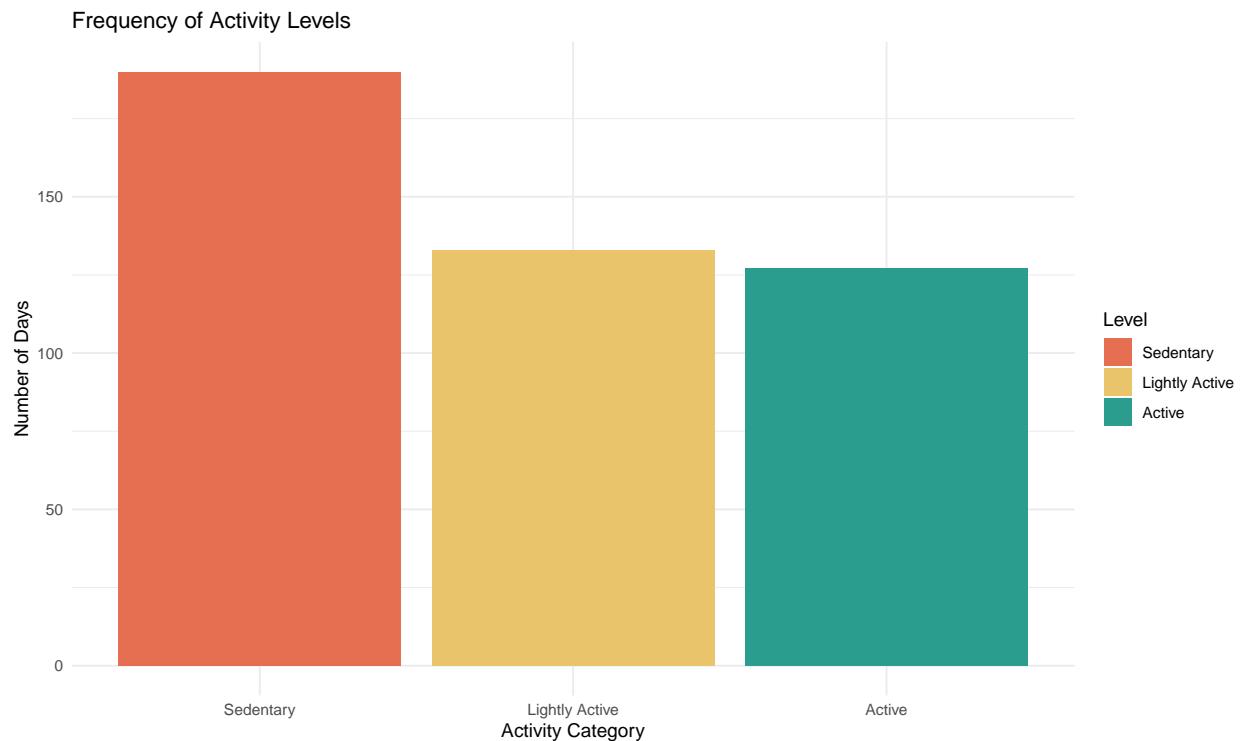
# 2. Plot
ggplot(daily_activity, aes(x = activity_level, fill = activity_level)) +

```

```

geom_bar() +
scale_fill_manual(values = c("Sedentary" = "#e76f51", "Lightly Active" = "#e9c46a", "Active" = "#2a9d2c"),
theme_minimal() +
labs(
  title = "Frequency of Activity Levels",
  x = "Activity Category",
  y = "Number of Days",
  fill = "Level"
)

```



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 less common

- **Observation:** A large portion of sleep records show durations between **7 to 8.5 hours**, keeping it stable.
- **Implication:** Sleep wellness is a major area for consistency where Bellabeat can offer unique value through coaching and engaging messages.

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:** I 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 keeping consistency for sleep (7-8.5 hours), introducing a “Sleep Hygiene” module and engaging notifications. 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: 2026-01-26 **Author:** Amartisoaei Robert