Bellabeat Case Study - Fitbit Data Analysis

Tim Foltz

2025-08-11

Contents

Executive Summary

Business Task. Analyze public Fitbit tracker data to find usage trends and translate them into marketing opportunities for Bellabeat products.

Key Findings. - Steps and calories are strongly related; most high-calorie days pair with moderate–high steps and ~6–9 hours of sleep. - Light activity dominates time budgets; very-active minutes are scarce. - Weekends and weekdays have similar odds of hitting 10k steps; weekends show wider spread in sleep hours. - Hourly patterns peak mid-day and early evening.

Recommendations. - Run a cross-week 10k Steps Challenge with streak badges and conversion tracking. - Trigger gentle-day routines when sleep < 7 hours (lighter goals, reminders). - Promote intensity-mix micro-goals (e.g., 2×10 minutes fairly active + 40 minutes lightly active).

Ask

- What trends exist in smart device usage?
- How do those trends apply to Bellabeat customers?
- How can they influence Bellabeat's marketing strategy?

Prepare

Data. Fitbit Fitness Tracker Dataset (Mobius/Kaggle, CC0).
Periods. 3.12.16–4.11.16 and 4.12.16–5.12.16.
Notes. Small convenience sample; missingness and non-wear present.

```
library(tidyverse)
library(lubridate)
library(fs)
library(janitor)
library(ggplot2)
library(scales)
library(viridis)
```

```
folder1 <- "data/mturkfitbit_export_3.12.16-4.11.16/Fitabase_Data_3.12.16-4.11.16"
folder2 <- "data/mturkfitbit_export_4.12.16-5.12.16/Fitabase_Data_4.12.16-5.12.16"

file_paths1 <- fs::dir_ls(folder1, glob = "*.csv")
file_paths2 <- fs::dir_ls(folder2, glob = "*.csv")

file_contents_1 <- setNames(lapply(file_paths1, readr::read_csv, show_col_types = FALSE), basename(file_
file_contents_2 <- setNames(lapply(file_paths2, readr::read_csv, show_col_types = FALSE), basename(file_

all_files <- union(names(file_contents_1), names(file_contents_2))
data <- lapply(all_files, function(fname) {
  file1 <- file_contents_1[[fname]]
  file2 <- file_contents_2[[fname]]
  if (!is.null(file1) && !is.null(file2)) bind_rows(file1, file2) else if (!is.null(file1)) file1 else
})
names(data) <- all_files
```

Process

```r
daily_activity <- data[["dailyActivity_merged.csv"]] %>%
  janitor::clean_names() %>%
  mutate(
    activity_date = as.Date(activity_date, format = "%m/%d/%Y"),
    weekday = wday(activity_date, label = TRUE, abbr = FALSE, week_start = 1),
    total_active_minutes = very_active_minutes + fairly_active_minutes + lightly_active_minutes
  ) %>%
  distinct()

sleep_data <- data[["sleepDay_merged.csv"]] %>%
  janitor::clean_names() %>%
  mutate(sleep_day = as.Date(sleep_day, format = "%m/%d/%Y")) %>%
  distinct(id, sleep_day, .keep_all = TRUE)

activity_sleep <- daily_activity %>%
  left_join(sleep_data, by = c("id", "activity_date" = "sleep_day"))
```

```r
daily_activity <- daily_activity %>%
  filter(!(total_steps == 0 & calories > 2500))
activity_sleep <- activity_sleep %>%
  filter(!(total_steps == 0 & calories > 2500))
```

Analyze

KPI Snapshot

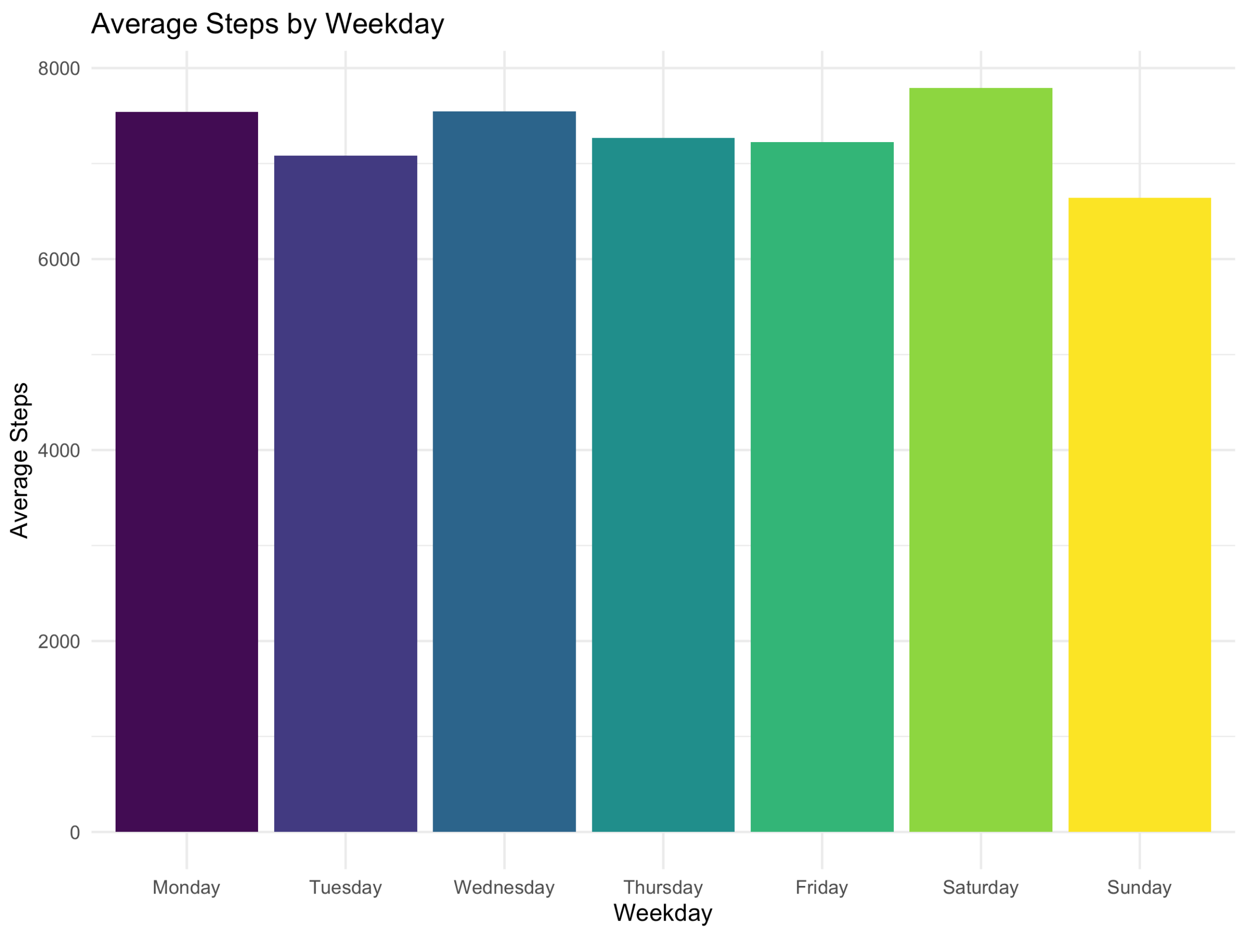
```r
kpis <- activity_sleep %>%
  mutate(
    total_hours_asleep = total_minutes_asleep / 60,
    total_active_hours = total_active_minutes / 60
  ) %>%
  summarise(
    avg_steps = mean(total_steps, na.rm = TRUE),
    median_steps = median(total_steps, na.rm = TRUE),
    avg_active_hours = mean(total_active_hours, na.rm = TRUE),
    avg_calories = mean(calories, na.rm = TRUE),
    avg_sleep_hours = mean(total_hours_asleep, na.rm = TRUE),
    sleep_efficiency = mean(total_minutes_asleep / total_time_in_bed, na.rm = TRUE)
  )
kpis
```

```
## # A tibble: 1 x 6
##   avg_steps median_steps avg_active_hours avg_calories avg_sleep_hours
##       <dbl>        <dbl>            <dbl>        <dbl>           <dbl>
## 1     7297.        7012.             3.64        2264.            7.00
## # i 1 more variable: sleep_efficiency <dbl>
```

Notes. Typical day shows moderate steps, ~6–8 hours sleep; calories track steps plus basal burn.

Average Steps by Weekday

```
avg_steps_weekday <- daily_activity %>%
  group_by(weekday) %>%
  summarise(avg_steps = mean(total_steps, na.rm = TRUE)) %>%
  ungroup()

ggplot(avg_steps_weekday, aes(x = weekday, y = avg_steps, fill = weekday)) +
  geom_col(show.legend = FALSE) +
  labs(title = "Average Steps by Weekday", x = "Weekday", y = "Average Steps") +
  theme_minimal()
```



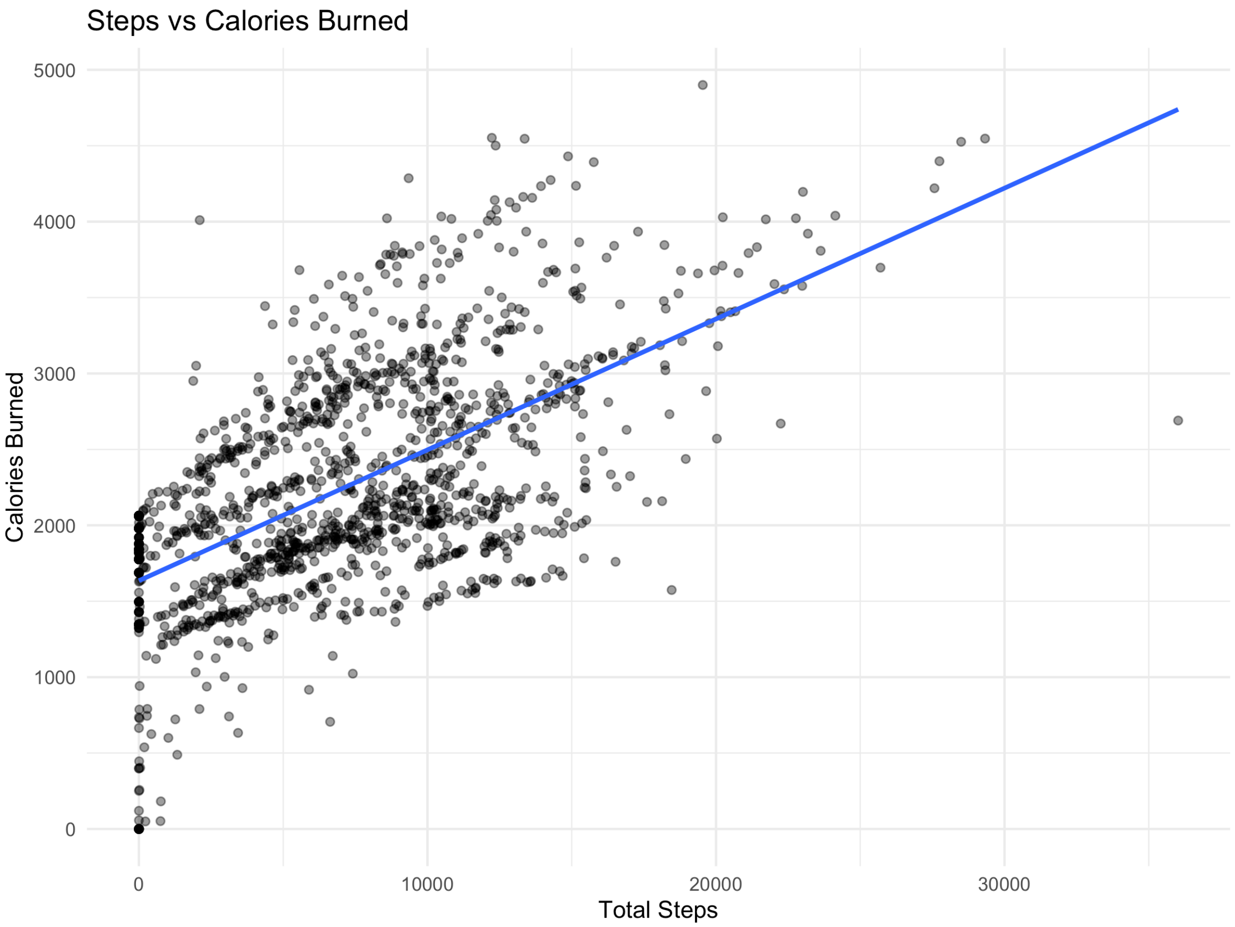
Notes. Steps are steady across the week; slight midweek lift suggests routine-driven activity.

Steps vs Calories (Outliers Removed)

```
steps_cal_cor <- cor(daily_activity$total_steps, daily_activity$calories, use = "complete.obs")
steps_cal_cor
```

```
## [1] 0.5976419
```

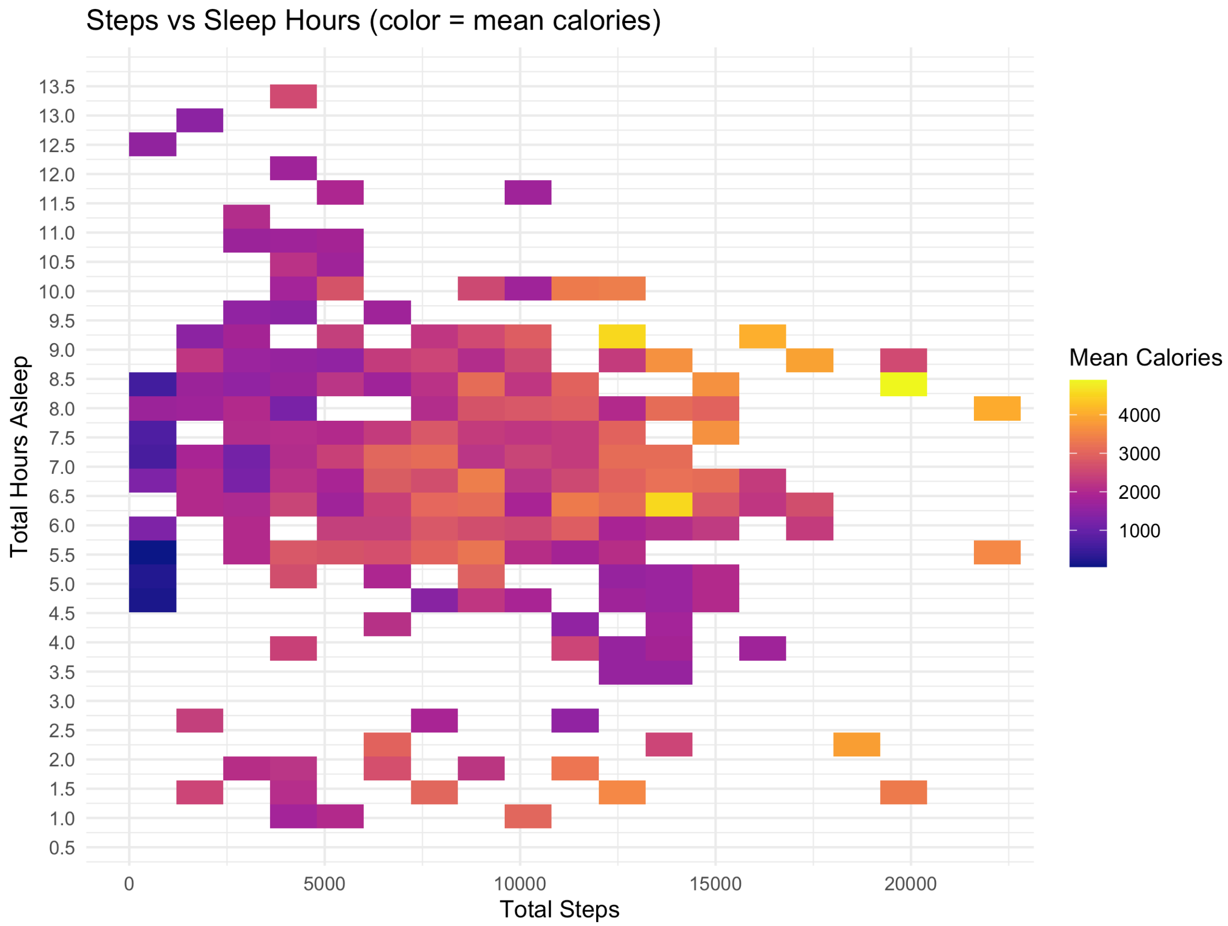
```
ggplot(daily_activity, aes(x = total_steps, y = calories)) +
  geom_point(alpha = 0.45) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Steps vs Calories Burned", x = "Total Steps", y = "Calories Burned") +
  theme_minimal()
```



Notes. Clear positive relationship; removing non-wear anomalies tightens the trend.

Steps × Sleep Hours → Mean Calories (Heatmap)

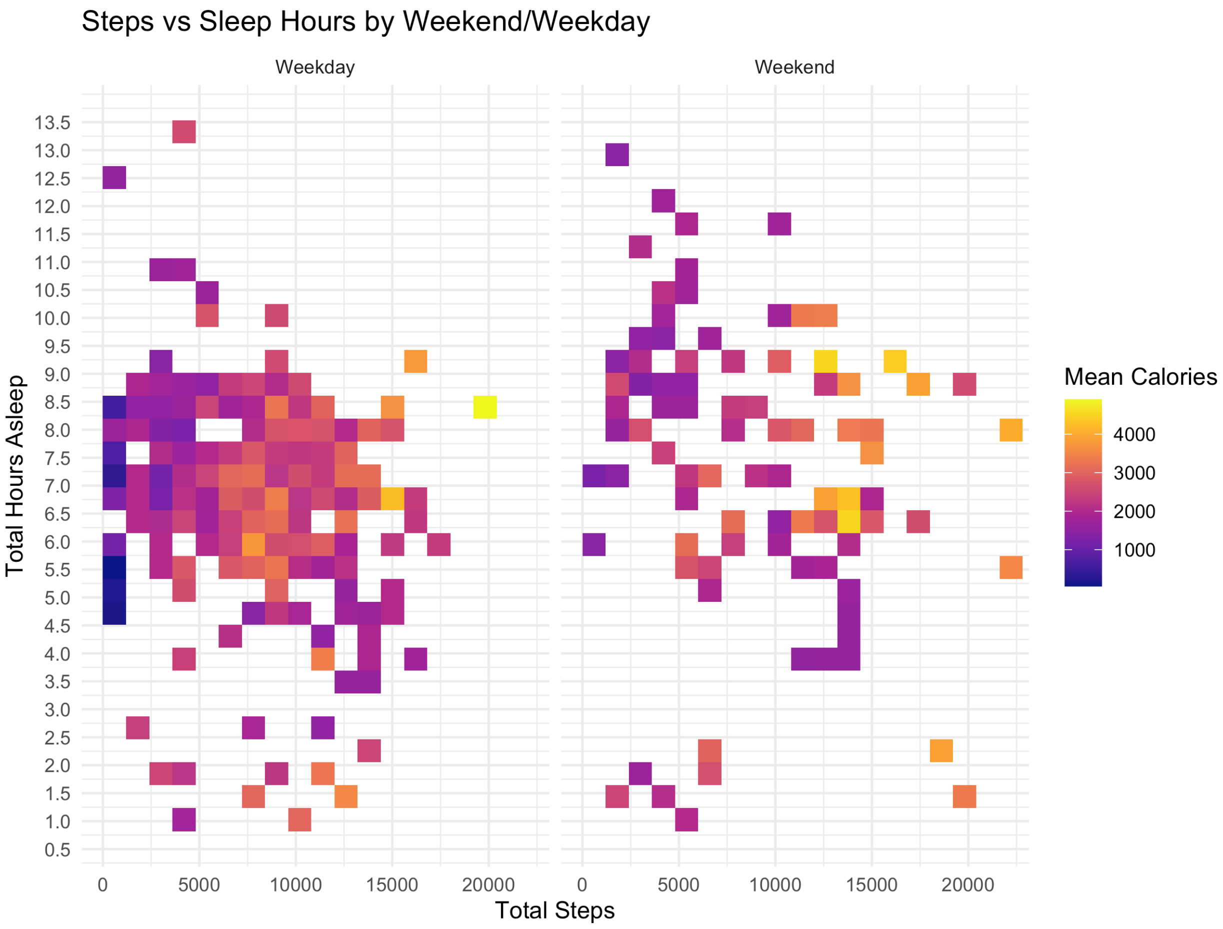
```
activity_sleep2 <- activity_sleep %>%
  mutate(
    total_hours_asleep = total_minutes_asleep / 60,
    is_weekend = factor(if_else(weekday %in% c("Saturday","Sunday"), "Weekend", "Weekday"),
                       levels = c("Weekday","Weekend"))
  )

y_max <- max(activity_sleep2$total_hours_asleep, na.rm = TRUE)
y_lim <- ceiling(y_max * 2) / 2
x_max <- quantile(activity_sleep2$total_steps, 0.99, na.rm = TRUE)

ggplot(activity_sleep2, aes(x = total_steps, y = total_hours_asleep)) +
  stat_summary_2d(aes(z = calories), fun = mean, bins = 30) +
  scale_fill_viridis(name = "Mean Calories", option = "C") +
  coord_cartesian(xlim = c(0, x_max)) +
  scale_y_continuous(breaks = seq(0, y_lim, 0.5)) +
  labs(title = "Steps vs Sleep Hours (color = mean calories)", x = "Total Steps", y = "Total Hours Asle
  theme_minimal()
```



Notes. High calories concentrate with moderate–high steps and ~6–9 hours of sleep; very short sleep correlates with lower calories at similar steps.

Steps × Sleep Hours by Weekend/Weekday (Heatmap)

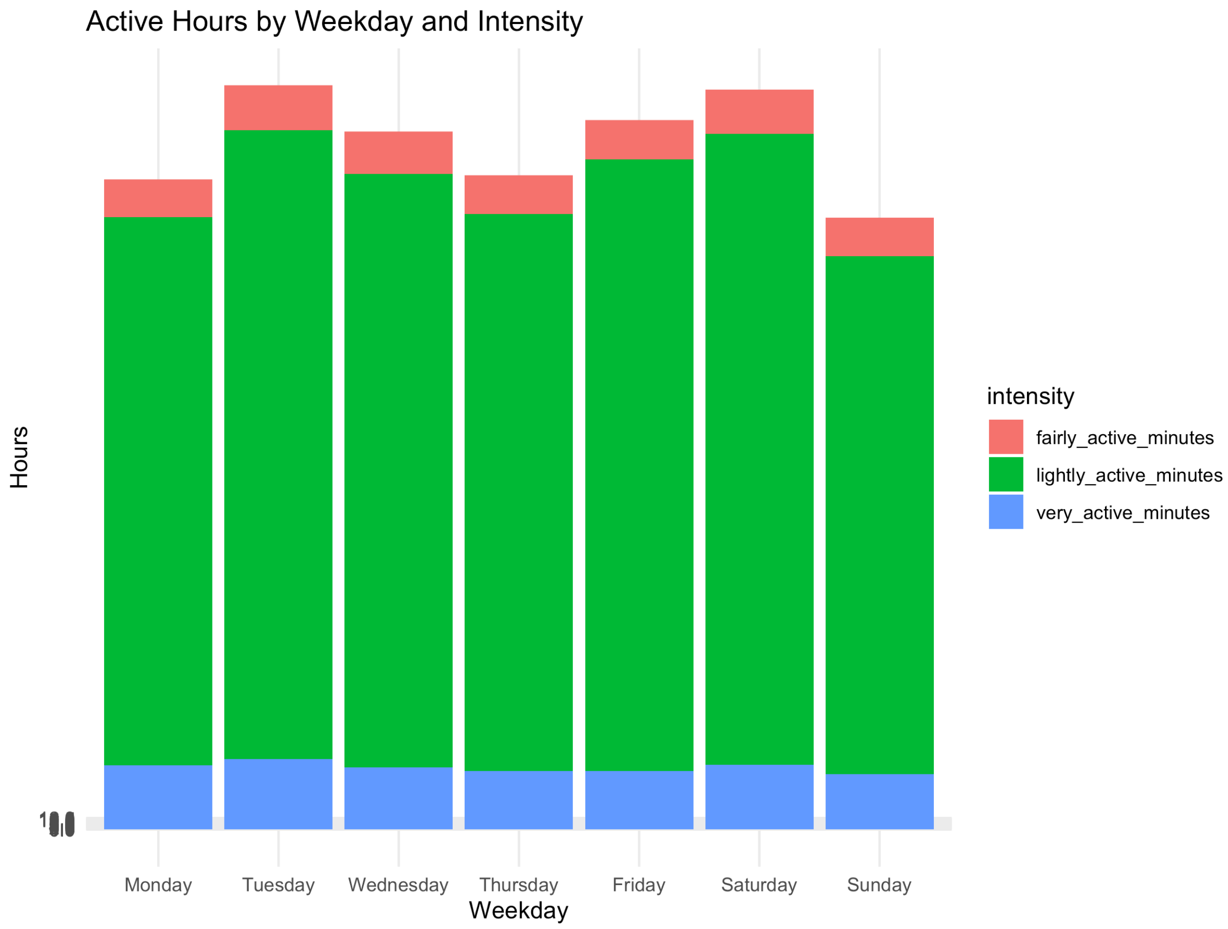
```
ggplot(activity_sleep2, aes(x = total_steps, y = total_hours_asleep)) +
  stat_summary_2d(aes(z = calories), fun = mean, bins = 30) +
  scale_fill_viridis(name = "Mean Calories", option = "C") +
  coord_cartesian(xlim = c(0, x_max)) +
  scale_y_continuous(breaks = seq(0, y_lim, 0.5)) +
  facet_wrap(~ is_weekend) +
  labs(title = "Steps vs Sleep Hours by Weekend/Weekday", x = "Total Steps", y = "Total Hours Asleep") +
  theme_minimal()
```



Notes. Weekend sleep is more variable; weekday patterns are tighter around workday schedules.

Active Hours by Weekday and Intensity

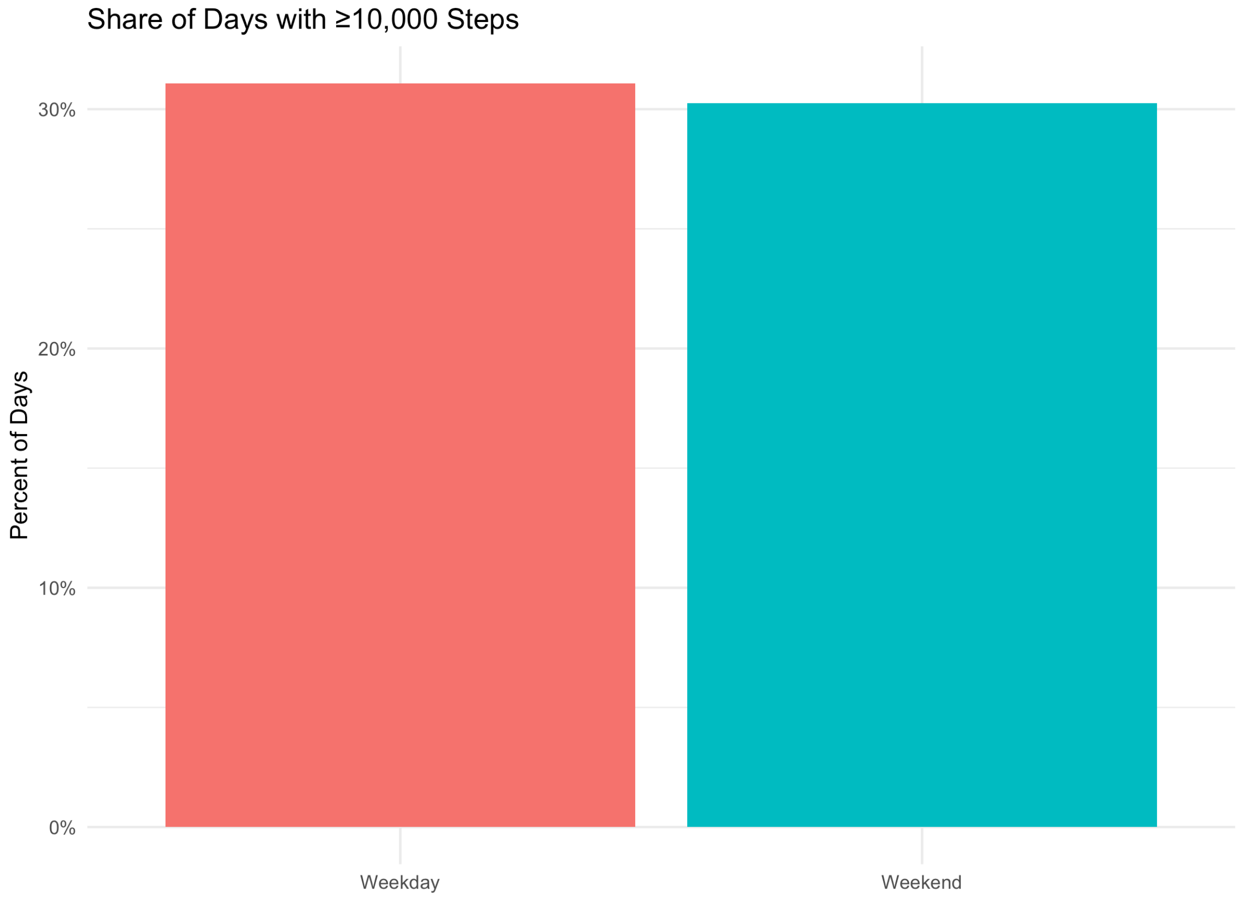
```
intensity_long <- daily_activity %>%
  select(weekday, very_active_minutes, fairly_active_minutes, lightly_active_minutes) %>%
  pivot_longer(
    cols = c(very_active_minutes, fairly_active_minutes, lightly_active_minutes),
    names_to = "intensity",
    values_to = "minutes"
  ) %>%
  mutate(hours = minutes / 60)

y_max <- max(intensity_long$hours, na.rm = TRUE)
y_lim <- ceiling(y_max * 2) / 2

ggplot(intensity_long, aes(x = weekday, y = hours, fill = intensity)) +
  geom_col() +
  scale_y_continuous(breaks = seq(0, y_lim, 0.5)) +
  labs(title = "Active Hours by Weekday and Intensity", x = "Weekday", y = "Hours") +
  theme_minimal()
```



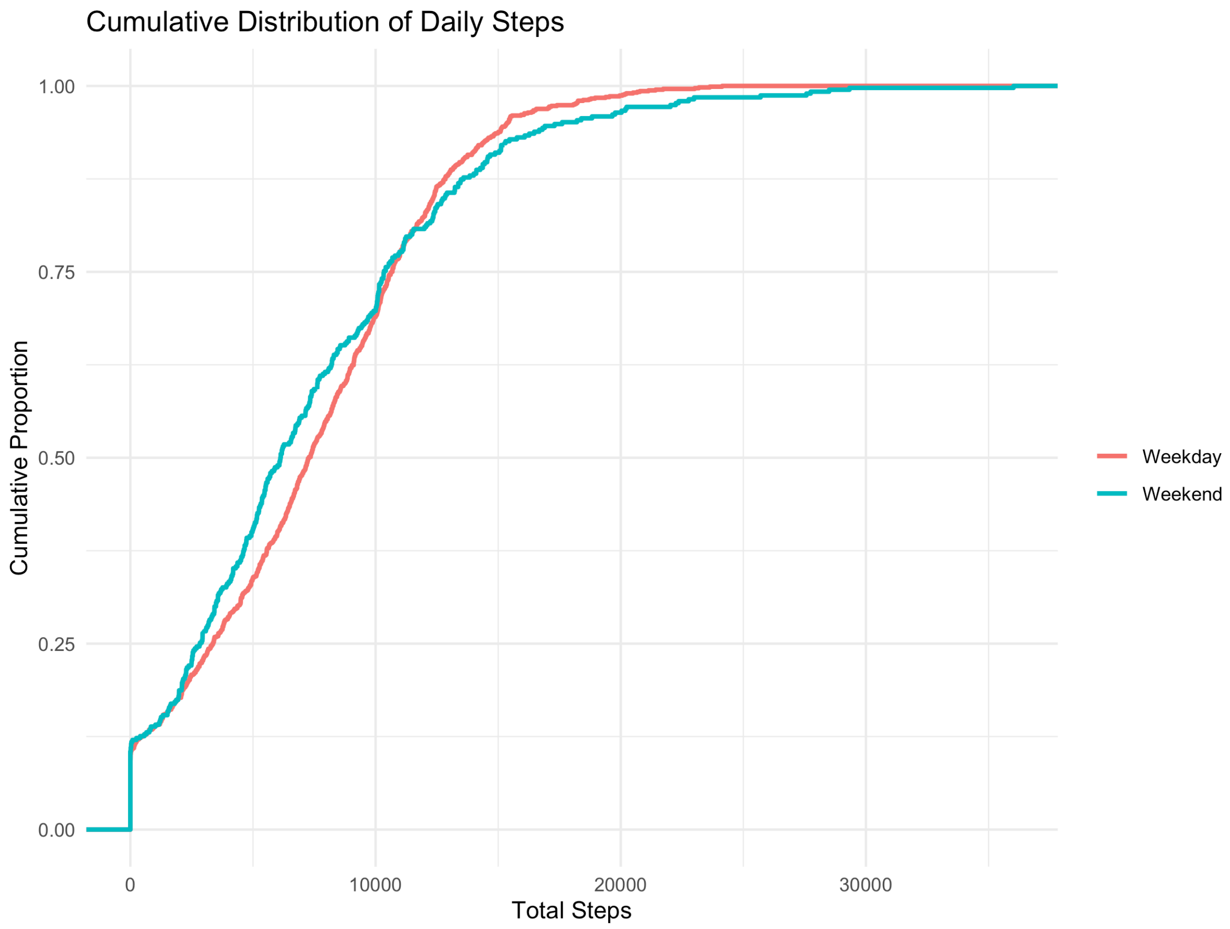
Notes. Light activity dominates; scope for micro-bursts of higher intensity without increasing total time burden.

Weekend vs Weekday (Goal & Distribution)

```
daily_activity <- daily_activity %>%
  mutate(
    is_weekend = factor(if_else(weekday %in% c("Saturday","Sunday"), "Weekend", "Weekday"),
                        levels = c("Weekday","Weekend")),
    hit_10k = total_steps >= 10000
  )

pct_10k <- daily_activity %>%
  group_by(is_weekend) %>%
  summarise(pct = mean(hit_10k, na.rm = TRUE))

ggplot(pct_10k, aes(x = is_weekend, y = pct, fill = is_weekend)) +
  geom_col(show.legend = FALSE) +
  scale_y_continuous(labels = scales::percent) +
  labs(title = "Share of Days with  10,000 Steps", x = "", y = "Percent of Days") +
  theme_minimal()
```



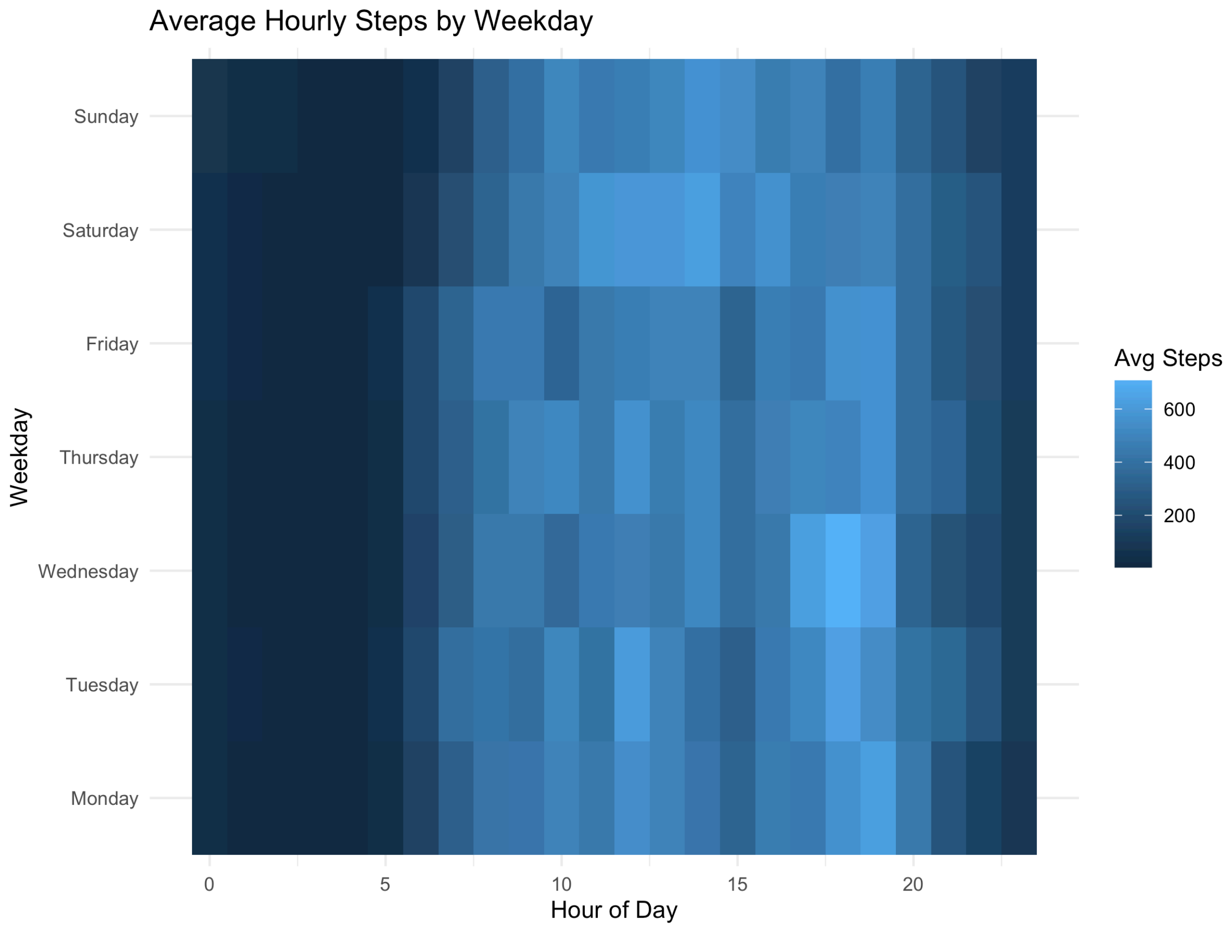
```
ggplot(daily_activity, aes(x = total_steps, color = is_weekend)) +
  stat_ecdf(size = 1) +
  labs(title = "Cumulative Distribution of Daily Steps", x = "Total Steps", y = "Cumulative Proportion"
  theme_minimal()
```



Notes. Read proportions at 10k to quantify weekend vs weekday goal attainment.

Hourly Steps Heatmap

```
show_heatmap <- "hourlySteps_merged.csv" %in% names(data)

if (show_heatmap) {
  hourly_steps <- data[["hourlySteps_merged.csv"]] %>%
    janitor::clean_names() %>%
    mutate(
      activity_hour = mdy_hms(activity_hour),
      date = as.Date(activity_hour),
      hour = lubridate::hour(activity_hour),
      weekday = wday(date, label = TRUE, abbr = FALSE, week_start = 1)
    )

  heat <- hourly_steps %>%
    group_by(weekday, hour) %>%
    summarise(steps = mean(step_total, na.rm = TRUE), .groups = "drop")

  p08 <- ggplot(heat, aes(x = hour, y = weekday, fill = steps)) +
    geom_tile() +
    labs(title = "Average Hourly Steps by Weekday", x = "Hour of Day", y = "Weekday", fill = "Avg Steps
    theme_minimal()

  print(p08)
} else {
  message("hourlySteps_merged.csv not found in data; skipping heatmap.")
}
```



Notes - Step activity peaks during **morning (7–9 AM)** and **early evening (5–8 PM)** on most weekdays. - Weekends show a more diffuse pattern with weaker morning peaks. - Timing nudges around these windows should maximize engagement.

Share (Slide Deck Outline)

1. **Title Slide** — *Bellabeat Case Study – Fitbit Data Analysis*
   Tim Foltz – August 2025
2. **Business Task**
   - Analyze Fitbit usage trends to understand consumer habits

- Apply insights to Bellabeat products to improve marketing strategies
- Deliver actionable recommendations based on data findings

3. **Data Sources**
    - Fitbit Fitness Tracker Dataset (CC0 Public Domain)
    - ~30 participants with activity, sleep, and steps data
    - Two periods: March–May 2016
    - Some gaps in sleep and heart rate tracking
4. **Methodology**
    - Data cleaning, standardization, and merging across date ranges
    - Derived helper columns (weekend/weekday, total active hours)
    - Filtered outliers (e.g., calories > 2500 with 0 steps)
    - Explored patterns in activity, sleep, and calories
5. **Plot 1 – Average Steps by Weekday**
    - Most steps midweek; Monday/Sunday lowest
    - Suggests potential for weekend activity campaigns
6. **Plot 2 – Steps vs Calories Burned**
    - Clear positive relationship
    - Calories > 0 at ~0 steps suggests logged workouts without Fitbit movement data
7. **Plot 3 – Sleep Hours vs Steps (Color/Size by Calories)**
    - High calories with moderate–high steps and ~6–9 hours sleep
    - Very short sleep correlates with lower calories at similar steps
8. **Plot 4 – Weekend vs Weekday % 10k Steps**
    - Weekdays higher very-active minutes; weekends more lightly active minutes
9. **Plot 5 – Active Hours Intensity**
    - Light activity dominates; scope for short bursts of higher intensity without extra time
10. **Plot 6 – Percent of Days with 10k Steps**
    - ~30% of days meet goal; no major weekday/weekend difference
    - Longer weekend sleep may reduce active hours
11. **Plot 7 – Distribution of Daily Steps**

- ECDF shows similar weekend/weekday goal attainment proportions

12. **Plot 8 – Hourly Steps Heatmap**
    - Moderate steps, 6–8 hours sleep, calories aligned with activity and basal burn
    - Sleep efficiency stable; opportunity to raise fairly-active time
13. **Key Findings**
    - Weekends lower average steps
    - Steps strongly correlate with calories
    - More sleep modestly linked to higher next-day activity
    - Activity peaks midday with evening bump
14. **Recommendations**
    - Weekend step challenges
    - Sleep coaching for <7h sleepers
    - Mixed-intensity badge rewards
    - Hydration/activity nudges at peak hours
15. **Act – Next Steps**
    - KPIs: +10% weekend steps, +5% active hours
    - A/B test push notification timing
    - Segment by sleep/activity patterns for tailored messaging
    - Gather seasonal, demographic, and campaign-overlap data

---

Act (Recommendations & Next Steps)

Recommendations. - 10k Steps Challenge (all week). Badges for streaks; spotlight top movers.
- **Sleep-Aware Coaching.** When sleep < 7h, suggest lighter goals and timed walk reminders.
- **Intensity Mix.** Promote 2×10 min fairly-active + 40 min lightly-active as a daily target.

Success Metrics. - Challenge participation rate; % days 10k; average steps delta vs baseline.
- Days with sleep <7h that still meet a scaled goal.
- Increase in fairly-active minutes without reducing total hours.

Experiment Plan. - A/B push timing (morning vs afternoon nudges).
- Targeting by prior-week sleep pattern.
- Content variants (badge vs streak messaging).

Data to Collect Next. - Longer time horizon (seasonality).
- Demographics and segments.

- Device wear-time to flag non-wear days.
- Campaign impression & click logs to tie to behavior change.

Limitations

Convenience sample (2016), potential non-wear and logging gaps, short window limits causal inference.

Reproducibility

```r
set.seed(42)
sessionInfo()
```

```
## R version 4.5.1 (2025-06-13)
## Platform: aarch64-apple-darwin20
## Running under: macOS Tahoe 26.0
## 
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRblas.0.dylib 
## LAPACK: /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRlapack.dylib;  LAPACK ve
## 
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
## 
## time zone: America/Phoenix
## tzcode source: internal
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
##  [1] viridis_0.6.5     viridisLite_0.4.2 scales_1.4.0      janitor_2.2.1    
##  [5] fs_1.6.6          lubridate_1.9.4   forcats_1.0.0     stringr_1.5.1    
##  [9] dplyr_1.1.4       purrr_1.0.4       readr_2.1.5       tidyr_1.3.1      
## [13] tibble_3.3.0      tidyverse_2.0.0   ggplot2_3.5.2    
## 
## loaded via a namespace (and not attached):
##  [1] generics_0.1.4     lattice_0.22-7     stringi_1.8.7      hms_1.1.3         
##  [5] digest_0.6.37      magrittr_2.0.3     evaluate_1.0.4     grid_4.5.1        
##  [9] timechange_0.3.0   RColorBrewer_1.1-3 fastmap_1.2.0      Matrix_1.7-3      
## [13] gridExtra_2.3      mgcv_1.9-3         cli_3.6.5          rlang_1.1.6       
## [17] crayon_1.5.3       splines_4.5.1      bit64_4.6.0-1      withr_3.0.2       
## [21] yaml_2.3.10        tools_4.5.1        parallel_4.5.1     tzdb_0.5.0        
## [25] vctrs_0.6.5        R6_2.6.1           lifecycle_1.0.4    snakecase_0.11.1  
## [29] bit_4.6.0          vroom_1.6.5        pkgconfig_2.0.3    pillar_1.10.2     
## [33] gtable_0.3.6       glue_1.8.0         xfun_0.52          tidyselect_1.2.1  
## [37] rstudioapi_0.17.1  knitr_1.50         farver_2.1.2       nlme_3.1-168
## [41] htmltools_0.5.8.1  rmarkdown_2.29     labeling_0.4.3     compiler_4.5.1
```