Bellabeat Capstone Case Study

Regan Thistle

Abstract

This case study explores patterns in activity, sleep, and energy expenditure using Fitbit smart-device data to guide marketing strategies for Bellabeat, a wellness company. Using R for cleaning, analysis, and visualization, I examined relationships between steps, calories, and sleep, as well as temporal patterns by weekday and hour. Results suggest that higher activity correlates strongly with calorie burn, 7–8 hours of sleep align with improved next-day activity, and evenings/weekends present opportunities for targeted engagement. These findings inform data-driven recommendations such as sleep reminders, evening workout nudges, and segmented user messaging to improve product adoption and retention.

Ask

Business Task: Analyze consumer smart-device usage patterns (activity, sleep, intensity) to recommend data-driven marketing actions for Bellabeat.

Key Questions - How do activity and sleep relate to energy expenditure (calories)? - What patterns emerge by weekday and time of day? - Which user segments (by steps) behave differently, and how can Bellabeat engage them?

Success Criteria - Clear, evidence-based recommendations. - Concise visuals supporting the insights.

Prepare

Data Source: Fitbit Fitness Tracker Data (public sample).

Files used: - `dailyActivity_merged.csv` - `sleepDay_merged.csv` - `hourlySteps_merged.csv`

Limitations - Small, non-representative sample (~30 users) - Older data; suitable for behavioral patterns, not market sizing

Process

Load packages and data with flexible column detection.

```r
library(tidyverse)
library(lubridate)
library(glue)
library(rlang)  # for .data pronoun used in dynamic column selection
```

```r
# Helper: safely get a column name (camelCase or snake_case)
get_col <- function(df, options) {
  match <- intersect(names(df), options)
  if (length(match) == 0) stop(glue::glue("None of {options} found in: {paste(names(df), collap
  match[1]
}
```

```r
# Point to your data directory (change via params if needed)
DATA_DIR <- params$data_dir

# ---- Daily Activity ----
act <- read_csv(file.path(DATA_DIR, "dailyActivity_merged.csv"))
act_date_col <- get_col(act, c("ActivityDate", "activity_date"))
act <- act %>%
  mutate(activity_date = lubridate::mdy(.data[[act_date_col]]))

# ---- Sleep ----
sleep <- read_csv(file.path(DATA_DIR, "sleepDay_merged.csv"))
sleep_date_col <- get_col(sleep, c("SleepDay", "sleep_day"))
sleep <- sleep %>%
  mutate(sleep_day = lubridate::mdy_hms(.data[[sleep_date_col]]),
         activity_date = as.Date(sleep_day))

# ---- Hourly Steps ----
hrly_steps <- read_csv(file.path(DATA_DIR, "hourlySteps_merged.csv"))
hr_col <- get_col(hrly_steps, c("ActivityHour", "activity_hour"))
hrly_steps <- hrly_steps %>%
  mutate(activity_hour = lubridate::mdy_hms(.data[[hr_col]]),
         date = as.Date(activity_hour),
         hour = lubridate::hour(activity_hour))
```

```r
# --- Standardize critical columns dynamically ---
# Activity columns
id_col_act      <- get_col(act, c("Id", "id"))
steps_col       <- get_col(act, c("TotalSteps", "total_steps"))
cal_col         <- get_col(act, c("Calories", "calories"))
very_col        <- get_col(act, c("VeryActiveMinutes", "very_active_minutes"))
fairly_col      <- get_col(act, c("FairlyActiveMinutes", "fairly_active_minutes"))
lightly_col     <- get_col(act, c("LightlyActiveMinutes", "lightly_active_minutes"))
sedentary_col   <- get_col(act, c("SedentaryMinutes", "sedentary_minutes"))

act_std <- act %>%
  transmute(
    id = .data[[id_col_act]],
    activity_date,
    total_steps = .data[[steps_col]],
    calories = .data[[cal_col]],
    very_active_minutes   = .data[[very_col]],
    fairly_active_minutes = .data[[fairly_col]],
    lightly_active_minutes= .data[[lightly_col]],
    sedentary_minutes     = .data[[sedentary_col]]
  )

# Sleep columns
id_col_sleep   <- get_col(sleep, c("Id", "id"))
sleep_min_col  <- get_col(sleep, c("TotalMinutesAsleep", "total_minutes_asleep"))

sleep_std <- sleep %>%
  group_by(id = .data[[id_col_sleep]], activity_date) %>%
  summarise(total_sleep = sum(.data[[sleep_min_col]], na.rm = TRUE), .groups = "drop")

# Join daily activity + sleep
df <- act_std %>% left_join(sleep_std, by = c("id", "activity_date"))

# Hourly steps columns (for heatmap)
id_col_hr  <- get_col(hrly_steps, c("Id", "id"))
steps_hr   <- get_col(hrly_steps, c("StepTotal", "Steps", "step_total", "steps"))

hrly_steps_std <- hrly_steps %>%
  transmute(
    id = .data[[id_col_hr]],
    activity_hour,
    date,
    hour,
    steps = .data[[steps_hr]]
  )

# Basic quality checks
list(
  act_cols   = names(act_std),
  sleep_cols = names(sleep_std),
  hourly_cols= names(hrly_steps_std)
)
```

```
## $act_cols
## [1] "id"                     "activity_date"          "total_steps"
## [4] "calories"               "very_active_minutes"    "fairly_active_minutes"
## [7] "lightly_active_minutes" "sedentary_minutes"
##
## $sleep_cols
## [1] "id"            "activity_date" "total_sleep"
##
## $hourly_cols
## [1] "id"            "activity_hour" "date"          "hour"
## [5] "steps"
```

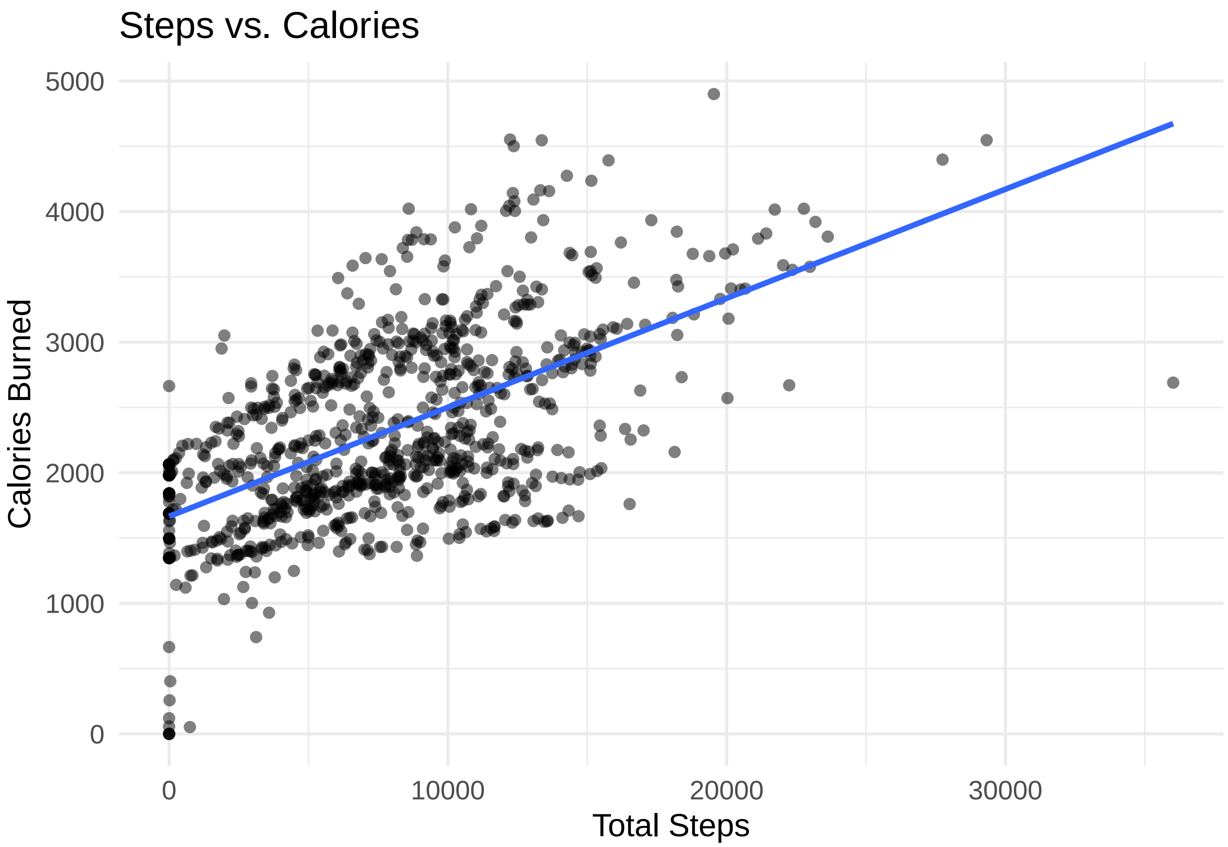
```
# Data Health Summary
n_users <- dplyr::n_distinct(df$id)
date_range <- range(df$activity_date, na.rm = TRUE)
rows_df <- nrow(df)
na_sleep <- sum(is.na(df$total_sleep))
sleep_cov <- mean(!is.na(df$total_sleep))


list(
users = n_users,
date_range = paste(as.character(date_range), collapse = " to "),
rows = rows_df,
sleep_rows_with_data_pct = round(100 * sleep_cov, 1)
)
```

```
## $users
## [1] 33
##
## $date_range
## [1] "2016-04-12 to 2016-05-12"
##
## $rows
## [1] 940
##
## $sleep_rows_with_data_pct
## [1] 43.6
```

Analyze

Q1: Are steps & active minutes associated with calories?

```
# Scatter with linear trend
p_steps_cal <- ggplot(df, aes(x = total_steps, y = calories)) +
geom_point(alpha = 0.5) +
geom_smooth(method = "lm", se = FALSE) +
labs(title = "Steps vs. Calories", x = "Total Steps", y = "Calories Burned") + theme_minimal(ba
p_steps_cal
```



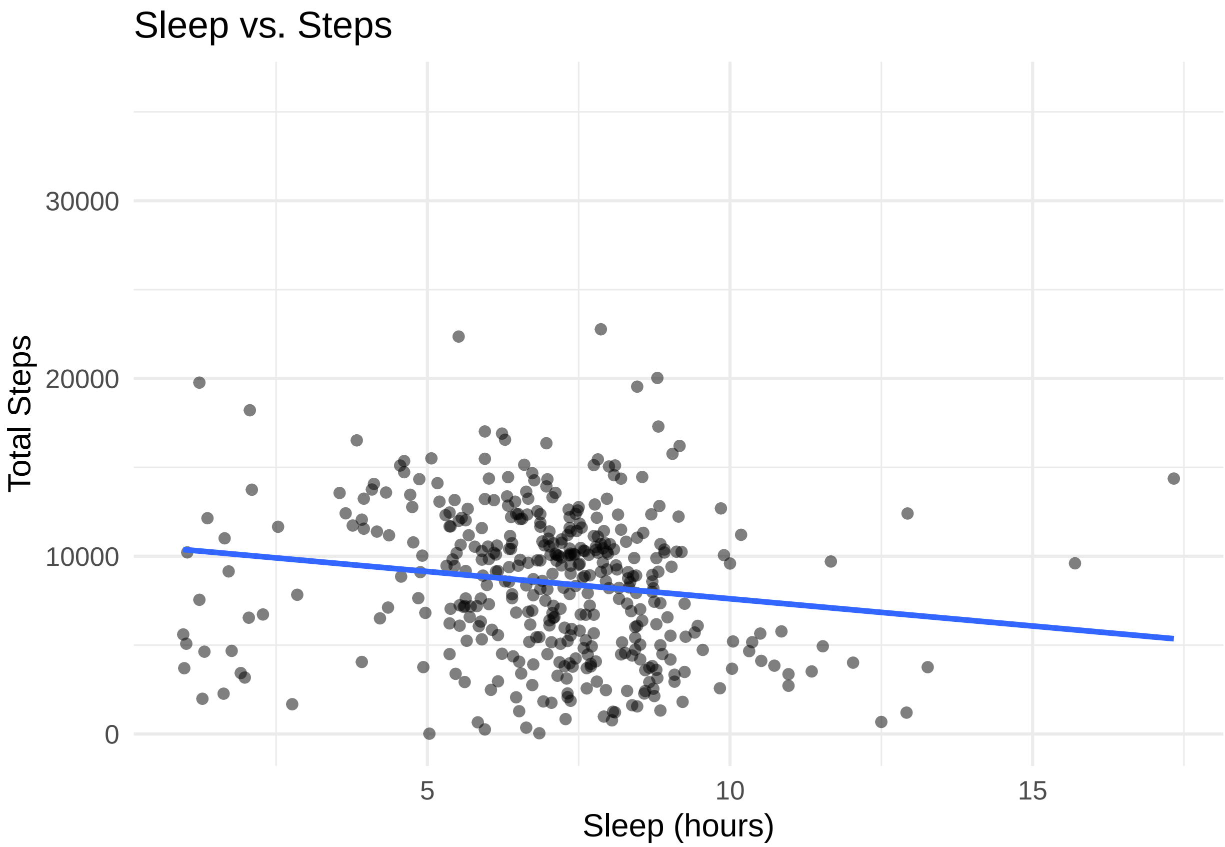
```r
# Simple correlation matrix input
cor_df <- df %>% drop_na(total_steps, calories, total_sleep,
                         very_active_minutes, lightly_active_minutes)
round(cor(select(cor_df, total_steps, calories, total_sleep,
                 very_active_minutes, lightly_active_minutes), use = "pairwise.complete.obs"),
```

```
##                        total_steps calories total_sleep very_active_minutes
## total_steps                  1.000    0.406      -0.155               0.544
## calories                     0.406    1.000       0.010               0.611
## total_sleep                 -0.155    0.010       1.000              -0.073
## very_active_minutes          0.544    0.611      -0.073               1.000
## lightly_active_minutes       0.417    0.114       0.043              -0.205
##                        lightly_active_minutes
## total_steps                             0.417
## calories                                0.114
## total_sleep                             0.043
## very_active_minutes                    -0.205
## lightly_active_minutes                  1.000
```

Takeaway - *Calories rise with steps and very active minutes; lightly active minutes show weaker association with calorie burn.*

Q2: Do users who sleep ~7–8h show higher next-day activity?

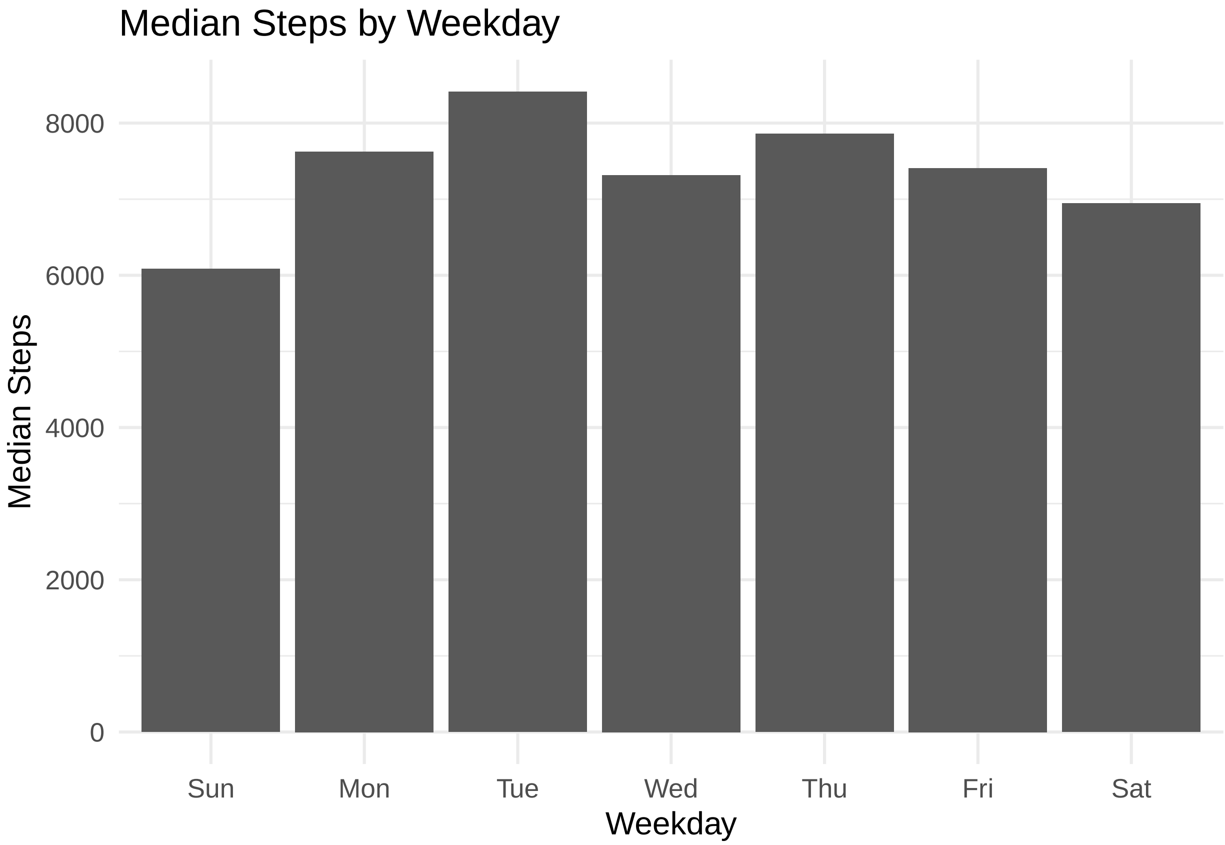
```r
# Convert sleep minutes to hours for readability
df <- df %>% mutate(sleep_hours = total_sleep/60)

p_sleep_steps <- ggplot(df, aes(x = sleep_hours, y = total_steps)) +
geom_point(alpha = 0.5) +
geom_smooth(method = "lm", se = FALSE) +
labs(title = "Sleep vs. Steps", x = "Sleep (hours)", y = "Total Steps") + theme_minimal(base_si
p_sleep_steps
```



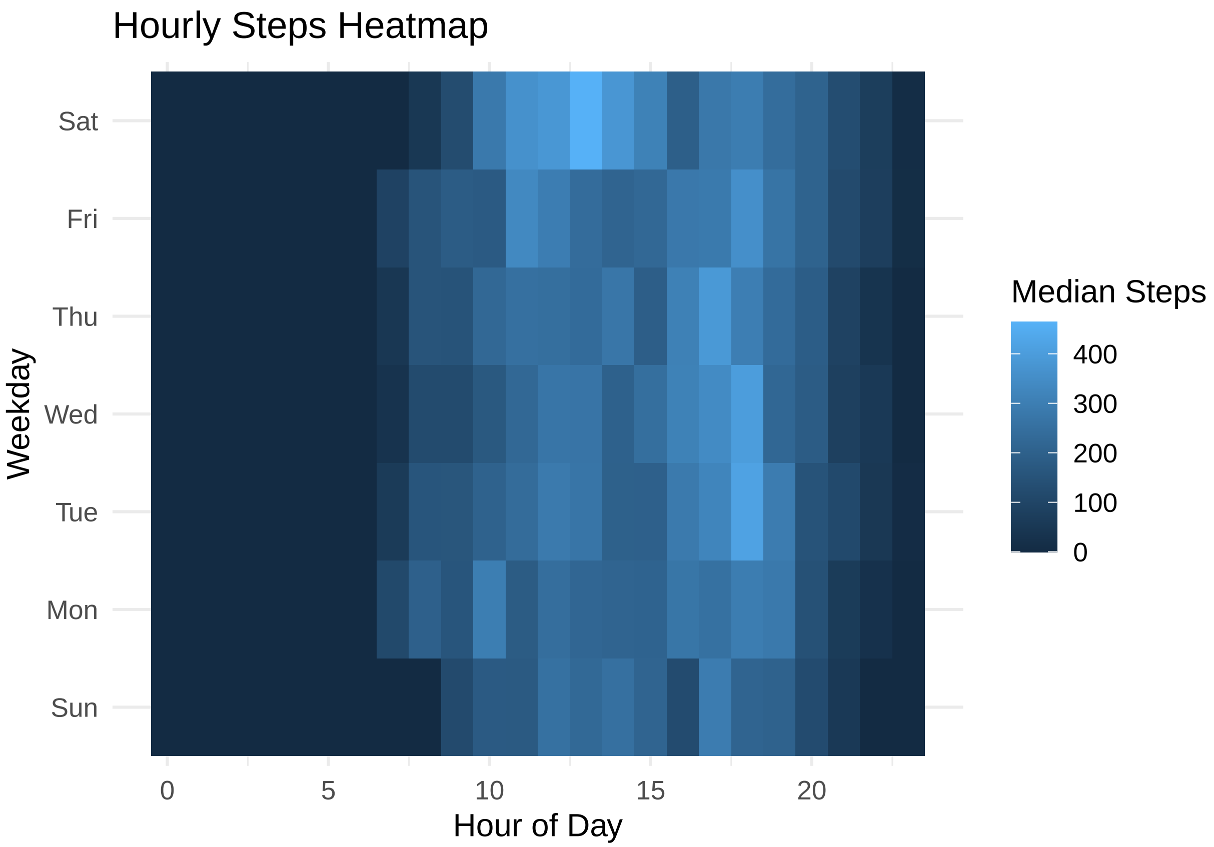
Takeaway: - *Moderate positive relationship up to ~8 hours; very low or very high sleep often corresponds to fewer steps.*

Q3: What weekday & hourly patterns stand out?

```r
weekday_df <- df %>%
  mutate(weekday = wday(activity_date, label = TRUE)) %>%
  group_by(weekday) %>%
  summarise(median_steps = median(total_steps, na.rm = TRUE), .groups = "drop")

p_weekday <- ggplot(weekday_df, aes(weekday, median_steps)) +
geom_col() +
labs(title = "Median Steps by Weekday", x = "Weekday", y = "Median Steps") + theme_minimal(base
p_weekday
```



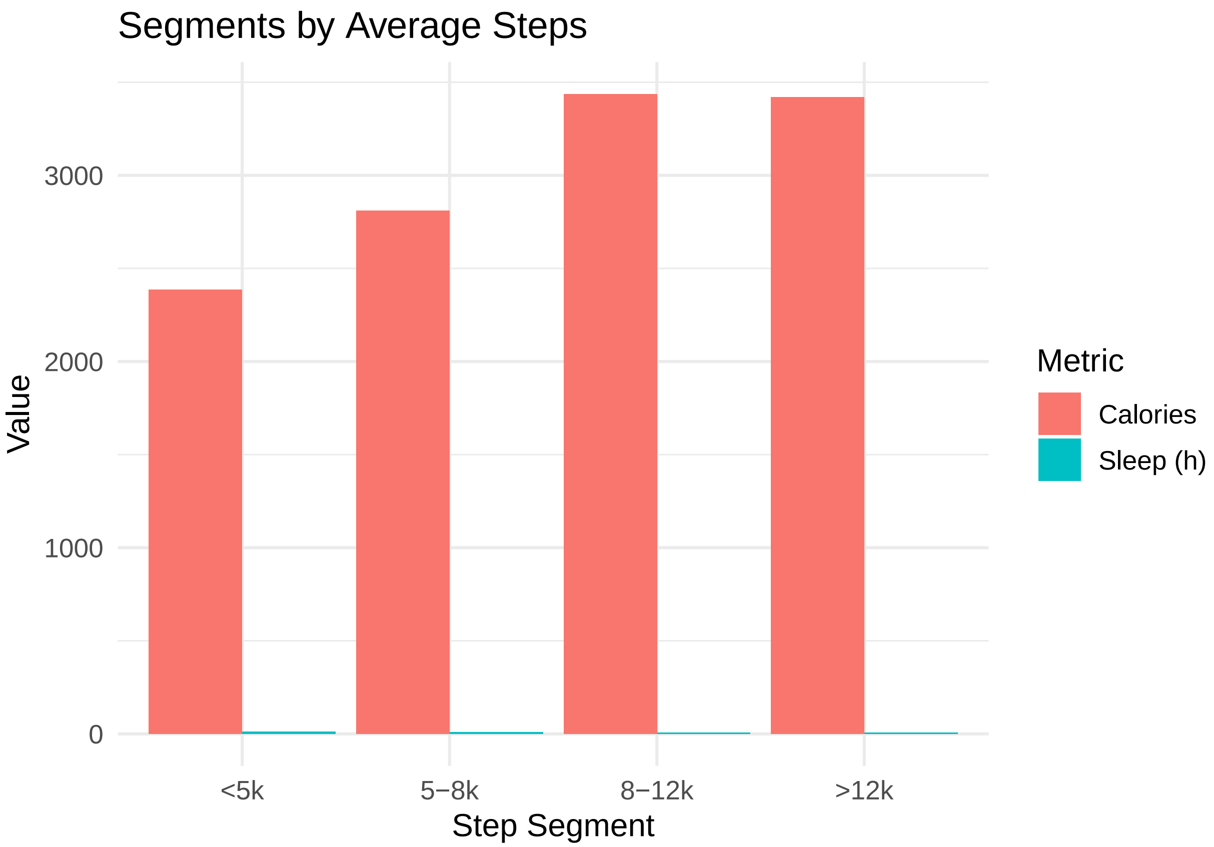
```
# Build hourly pattern table (median steps by hour x weekday)
hr_heat <- hrly_steps_std %>%
  mutate(weekday = wday(date, label = TRUE)) %>%
  group_by(weekday, hour) %>%
  summarise(median_steps = median(steps, na.rm = TRUE), .groups = "drop")

# Heatmap
p_heat <- ggplot(hr_heat, aes(x = hour, y = weekday, fill = median_steps)) +
geom_tile() +
scale_fill_continuous(name = "Median Steps") +
labs(title = "Hourly Steps Heatmap", x = "Hour of Day", y = "Weekday") + theme_minimal(base_si


p_heat
```



Takeaway: - *Activity peaks around commute/evening hours; weekends show different pattern, for example Saturday peaks earlier in the day.*

Q4: Segment users by steps and compare sleep & calories

```
segments <- df %>% group_by(id) %>%
  summarise(avg_steps = mean(total_steps, na.rm = TRUE),
            avg_sleep = mean(total_sleep, na.rm = TRUE),
            avg_cal = mean(calories, na.rm = TRUE), .groups = "drop") %>%
  mutate(step_segment = cut(avg_steps, c(-Inf,5000,8000,12000,Inf),
                            labels = c("<5k","5-8k","8-12k",">12k")))

seg_plot <- segments %>%
  pivot_longer(cols = c(avg_sleep, avg_cal), names_to = "metric", values_to = "value") %>%
  mutate(value = ifelse(metric == "avg_sleep", value/60, value),  # show sleep in hours
         metric = recode(metric, avg_sleep = "Sleep (h)", avg_cal = "Calories")) %>%
  ggplot(aes(step_segment, value, fill = metric)) +
  geom_col(position = position_dodge()) +
  labs(title = "Segments by Average Steps", x = "Step Segment", y = "Value", fill = "Metric")
seg_plot
```



Takeaway: - *Higher-step segments also show higher calories; mid-step segments may balance sleep and activity.*

Share

- **Insight 1:** Steps and very active minutes are strongly associated with higher calories.
- **Insight 2:** 7–8 hours of sleep aligns with higher activity the next day.
- **Insight 3:** Weekday evenings show an activity window; weekends under perform.
- **Insight 4:** Distinct step-based segments suggest targeted interventions.

Act (Recommendations)

Based on insights, propose actions Bellabeat can implement:

1. **Sleep → Activity Nudges:** Bedtime reminders + smart alarms to target 7–8h sleep (push notifications).
2. **Evening Micro-Workouts:** 10-minute post-dinner step prompts to capitalize on evening window.
3. **Weekend Streaks:** Gamified weekend challenges; shareable badges to lift engagement.
4. **Segmented Messaging:**
    - <5k: Gentle habit-building prompts, low-bar goals (3–5k).
    - 5–8k: Progress badges + social accountability.
    - 8–12k and >12k: Performance tips, accessory upsells.

Next Steps

To build on these findings, Bellabeat could design small experiments to validate the recommendations. For example, an A/B test of evening push notifications encouraging a 10-minute walk could measure the impact on evening step counts. Similarly, piloting a weekend “Streak Challenge” could reveal whether gamification lifts engagement during traditionally lower-activity periods. Finally, integrating sleep reminder features and tracking subsequent activity could help confirm the relationship between sleep duration and next-day performance. These steps would provide stronger evidence for scaling marketing strategies across Bellabeat’s product line.

Appendix

Reproducibility

```r
sessionInfo()
```

```
## R version 4.5.1 (2025-06-13)
## Platform: x86_64-pc-linux-gnu
## Running under: Ubuntu 20.04.6 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3;  LAPACK version 3.9.0
##
## locale:
##  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C           LC_TIME=C.UTF-8
##  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8    LC_MESSAGES=C.UTF-8
##  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
## [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8 LC_IDENTIFICATION=C
##
## time zone: UTC
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] rlang_1.1.6     glue_1.8.0      lubridate_1.9.4 forcats_1.0.0
##  [5] stringr_1.5.1   dplyr_1.1.4     purrr_1.1.0     readr_2.1.5
##  [9] tidyr_1.3.1     tibble_3.3.0    ggplot2_3.5.2   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] Matrix_1.7-3       bit_4.6.0          gtable_0.3.6       crayon_1.5.3
##  [5] compiler_4.5.1     tinytex_0.57       tidyselect_1.2.1   parallel_4.5.1
##  [9] splines_4.5.1      scales_1.4.0       yaml_2.3.10        fastmap_1.2.0
## [13] lattice_0.22-7     R6_2.6.1           labeling_0.4.3     generics_0.1.4
## [17] knitr_1.50         pillar_1.11.0      RColorBrewer_1.1-3 tzdb_0.5.0
## [21] stringi_1.8.7      xfun_0.53          bit64_4.6.0-1      timechange_0.3.0
## [25] cli_3.6.5          mgcv_1.9-3         withr_3.0.2        magrittr_2.0.3
## [29] digest_0.6.37      grid_4.5.1         vroom_1.6.5        rstudioapi_0.17.1
## [33] hms_1.1.3          nlme_3.1-168       lifecycle_1.0.4    vctrs_0.6.5
## [37] evaluate_1.0.4     farver_2.1.2       rmarkdown_2.29     tools_4.5.1
## [41] pkgconfig_2.0.3    htmltools_0.5.8.1
```

Export Cleaned Data & Figures

```
CLEAN_DIR <- params$clean_dir
if (!dir.exists(CLEAN_DIR)) dir.create(CLEAN_DIR, recursive = TRUE)

write_csv(df, file.path(CLEAN_DIR, "daily_merged_clean.csv"))

ggsave("fig_steps_calories.png", p_steps_cal, width = 7, height = 5, dpi = 300)
ggsave("fig_sleep_steps.png", p_sleep_steps, width = 7, height = 5, dpi = 300)
ggsave("fig_weekday_steps.png", p_weekday, width = 7, height = 5, dpi = 300)
ggsave("fig_hourly_heatmap.png", p_heat, width = 7, height = 5, dpi = 300)
ggsave("fig_segments.png", seg_plot, width = 7, height = 5, dpi = 300)
```
