Bellabeat Leaf Case Study — Methods & Validation Notebook

Olena Scott

January 10, 2026

Table of Contents

Objective and Leaf framing

Bellabeat Leaf is a passive wearable designed to support wellness through consistent daily habits rather than performance tracking. This analysis uses wearable activity and sleep-related data to identify movement rhythms, consistency patterns, and feature adoption signals that can inform product messaging and engagement strategies.

Primary questions:

How do daily movement and sedentary behavior vary across weekdays vs weekends, and do patterns replicate across time windows?

What are the typical hour-of-day activity rhythms, and are there consistent “dip” periods where gentle nudges could be effective?

How consistent are users across a month-long observation window, and what is the adoption rate of optional behaviors (sleep logging, weight logging)?

Data sources and observation-window logic

This project uses two Fitbit datasets provided in separate folders:

Window A: 2016-03-12 to 2016-04-11

Window B: 2016-04-12 to 2016-05-12

These are treated as two separate observation windows, not as a continuous 60-day panel. Data was cleaned and modeled in BigQuery into analysis marts.

Key design choices (already implemented in SQL):

A dataset_window field identifies the source window.

window_day_index (1–31) provides a consistent day-within-window scale.

Records were bounded to day index 1–31 to remove overlap at the window boundary.

Header rows ingested as data were dropped (e.g., Id/date parsing failures).

Daily/hourly marts were built for Tableau to prevent join duplication and ensure consistent metrics.

Data loading

```
users<- read_csv("user_summary.csv")
daily <- read_csv("daily_leaf.csv")
hourly <- read_csv("hourly_leaf.csv")
win <- read_csv("window_comparisson.csv")

glimpse(daily)
```

```
## Rows: 1,373
## Columns: 24
## $ Id                          <dbl> 4057192912, 4020332650,
4057192912, 4020332…
## $ log_date                    <date> 2016-03-12, 2016-03-12, 2016-
03-13, 2016-0…
## $ total_steps                 <dbl> 0, 5543, 0, 3226, 3023, 8433, 0,
5906, 1248…
## $ total_distance              <dbl> 0.00, 3.97, 0.00, 2.31, 2.17,
6.23, 0.00, 4…
## $ tracker_distance            <dbl> 0.00, 3.97, 0.00, 2.31, 2.17,
6.23, 0.00, 4…
## $ very_active_distance        <dbl> 0.00, 0.00, 0.00, 0.00, 0.00,
2.45, 0.00, 0…
## $ moderately_active_distance <dbl> 0.00, 0.00, 0.00, 0.00, 0.00,
0.33, 0.00, 0…
## $ light_active_distance       <dbl> 0.00, 3.96, 0.00, 2.28, 2.14,
3.44, 0.00, 4…
## $ sedentary_active_distance  <dbl> 0.00, 0.01, 0.00, 0.00, 0.00,
0.00, 0.00, 0…
## $ very_active_minutes         <dbl> 0, 0, 0, 0, 0, 30, 0, 0, 25, 0,
0, 0, 0, 0,…
## $ fairly_active_minutes       <dbl> 0, 0, 0, 0, 0, 7, 0, 10, 14, 28,
0, 0, 0, 0…
## $ lightly_active_minutes      <dbl> 0, 254, 0, 136, 145, 135, 0,
215, 309, 15, …
## $ sedentary_minutes           <dbl> 1440, 757, 1440, 771, 1005,
1268, 1440, 874…
## $ calories                    <dbl> 1777, 2990, 1777, 2480, 2570,
2453, 1776, 3…
## $ dataset_window              <chr> "2016-03-12_to_2016-04-11",
"2016-03-12_to_…
## $ is_weekend                  <lgl> TRUE, TRUE, TRUE, TRUE, FALSE,
```

```
FALSE, FALSE…
## $ week_day                  <dbl> 7, 7, 1, 1, 2, 2, 3, 3, 4, 4, 5,
5, 6, 6, 7…
## $ window_day_index          <dbl> 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6,
6, 7, 7, 8…
## $ minutes_asleep            <dbl> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA,…
## $ time_in_bed               <dbl> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA,…
## $ sleep_efficiency          <dbl> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA,…
## $ has_sleep_log_for_day     <lgl> FALSE, FALSE, FALSE, FALSE,
FALSE, FALSE, F…
## $ has_weight_log_for_day    <lgl> FALSE, FALSE, FALSE, FALSE,
FALSE, FALSE, F…
## $ weight_kg                 <dbl> NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA,…
```

```
glimpse(hourly)
```

```
## Rows: 46,008
## Columns: 11
## $ Id                <dbl> 1503960366, 5577150313, 1644430081,
5577150313, 6962…
## $ log_date          <date> 2016-03-12, 2016-03-12, 2016-03-12,
2016-03-12, 201…
## $ log_time          <time> 22:00:00, 13:00:00, 20:00:00, 11:00:00,
12:00:00, 1…
## $ total_steps       <dbl> 1132, 1934, 2339, 1041, 3216, 1764, 868,
2947, 945, …
## $ average_intensity <dbl> 0.683333, 1.683333, 1.250000, 1.650000,
1.716667, 0.…
## $ total_intensity   <dbl> 41, 101, 75, 99, 103, 49, 42, 129, 40,
39, 65, 83, 3…
## $ calories          <dbl> 123, 388, 284, 348, 234, 134, 175, 433,
206, 183, 30…
## $ dataset_window    <chr> "2016-03-12_to_2016-04-11", "2016-03-
12_to_2016-04-1…
## $ is_weekend        <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE,
TRUE, TRUE…
## $ week_day          <dbl> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
7, 7, 7, 7…
## $ window_day_index  <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1…
```

```
glimpse(users)
```

```
## Rows: 68
## Columns: 13
## $ Id                    <dbl> 4057192912, 4388161847, 8583815059,
6775888955, …
```

```
## $ dataset_window        <chr> "2016-04-12_to_2016-05-12", "2016-03-
12_to_2016-…
## $ days_with_activity    <dbl> 4, 8, 8, 9, 9, 10, 10, 10, 10, 10,
11, 11, 11, 1…
## $ avg_steps             <dbl> 3838.000, 0.000, 3045.500, 5559.000,
1336.889, 9…
## $ avg_distance          <dbl> 2.862500, 0.000000, 2.377500,
3.985556, 1.074444…
## $ avg_calories          <dbl> 1973.750, 1805.250, 2391.250,
2724.778, 1763.111…
## $ step_variability      <dbl> 2691.666, 0.000, 1665.129, 5231.218,
3235.173, 5…
## $ avg_sedentary_minutes <dbl> 1217.2500, 1384.2500, 1261.7500,
1017.8889, 1261…
## $ active_days_pct       <dbl> 0.1290323, 0.2580645, 0.2580645,
0.2903226, 0.29…
## $ has_sleep_log         <lgl> FALSE, FALSE, FALSE, FALSE, FALSE,
FALSE, FALSE,…
## $ has_weight_log        <lgl> FALSE, FALSE, FALSE, FALSE, FALSE,
FALSE, FALSE,…
## $ sleep_rows            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, …
## $ weight_rows           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, …
```

```
glimpse(win)
```

```
## Rows: 14
## Columns: 15
## $ dataset_window            <chr> "2016-03-12_to_2016-04-11",
"2016-03-12_to_…
## $ week_day                  <dbl> 1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4,
5, 6, 7
## $ avg_total_steps           <dbl> 6058.014, 7118.588, 6459.388,
7510.708, 684…
## $ avg_total_distance        <dbl> 4.322778, 5.054559, 4.592449,
5.371458, 4.8…
## $ avg_tracker_distance      <dbl> 4.322778, 5.023529, 4.400612,
5.346042, 4.8…
## $ avg_very_active_distance  <dbl> 1.161528, 1.283382, 0.937551,
1.252917, 1.2…
## $ avg_moderately_active     <dbl> 0.4138889, 0.4133824, 0.4204082,
0.6320833,…
## $ avg_light_active          <dbl> 2.568056, 3.240882, 3.121020,
3.384167, 2.8…
## $ avg_sedentary_active      <dbl> 0.001250000, 0.001176471,
0.001632653, 0.00…
## $ avg_very_active_minutes   <dbl> 15.34722, 18.95588, 15.32653,
16.97917, 17.…
## $ avg_fairly_active_minutes <dbl> 9.208333, 10.397059, 12.224490,
14.520833, …
```

```
## $ avg_lightly_active_minutes <dbl> 163.5694, 171.6471, 186.6531,
190.1250, 167…
## $ avg_sedentary_minutes      <dbl> 1015.6528, 1032.8529, 1027.2449,
1011.0625,…
## $ avg_calories               <dbl> 2167.597, 2252.868, 2285.735,
2377.458, 229…
## $ total_days                 <dbl> 72, 68, 49, 48, 48, 73, 75, 121,
120, 152, …
```

QA and coverage checks

This section documents data health checks and coverage to ensure downstream insights are trustworthy.

Row counts and unique users (by window)

```
daily %>% count(dataset_window, name = "daily_rows")
```

```
## # A tibble: 2 × 2
##   dataset_window           daily_rows
##   <chr>                         <int>
## 1 2016-03-12_to_2016-04-11        433
## 2 2016-04-12_to_2016-05-12        940
```

```
hourly %>% count(dataset_window, name = "hourly_rows")
```

```
## # A tibble: 2 × 2
##   dataset_window           hourly_rows
##   <chr>                          <int>
## 1 2016-03-12_to_2016-04-11       23909
## 2 2016-04-12_to_2016-05-12       22099
```

```
users %>% count(dataset_window, name = "user_rows")
```

```
## # A tibble: 2 × 2
##   dataset_window           user_rows
##   <chr>                        <int>
## 1 2016-03-12_to_2016-04-11        35
## 2 2016-04-12_to_2016-05-12        33
```

```
daily %>% distinct(Id, dataset_window) %>% count(dataset_window, name
= "n_users_daily")
```

```
## # A tibble: 2 × 2
##   dataset_window           n_users_daily
##   <chr>                            <int>
## 1 2016-03-12_to_2016-04-11            35
## 2 2016-04-12_to_2016-05-12            33
```

```
hourly %>% distinct(Id, dataset_window) %>% count(dataset_window, name
= "n_users_hourly")
```

```
## # A tibble: 2 × 2
##   dataset_window           n_users_hourly
##   <chr>                             <int>
## 1 2016-03-12_to_2016-04-11             34
## 2 2016-04-12_to_2016-05-12             33
```

```
users %>% count(dataset_window, name = "n_users_summary")
```

```
## # A tibble: 2 × 2
##   dataset_window           n_users_summary
##   <chr>                              <int>
## 1 2016-03-12_to_2016-04-11              35
## 2 2016-04-12_to_2016-05-12              33
```

Interpretation notes: The row counts and unique user counts confirm that the data has been correctly consolidated across the two observation windows and that each mart reflects the expected grain.

The number of unique users is consistent across daily and summary marts, confirming that the daily activity table defines the core population for analysis.

At the hourly level, row counts are substantially higher than daily counts, as expected given the finer temporal granularity. The consistency in unique user counts between the hourly and daily marts confirms that no unintended user duplication was introduced during the union or parsing process.

The user summary mart contains one record per user per window, validating its role as a stable aggregation layer for consistency and adoption analysis.

Overall, these checks indicate that:

Both observation windows are present and clearly separated.

Each mart conforms to its intended grain (daily, hourly, or user-window).

No unexpected loss or inflation of users occurred during data cleaning and modeling.

Window day index bounds

```
daily %>% summarise(min=min(window_day_index),
max=max(window_day_index))
```

```
## # A tibble: 1 × 2
##     min   max
##   <dbl> <dbl>
## 1     1    31
```

```
hourly %>% summarise(min=min(window_day_index),
max=max(window_day_index))
```

```
## # A tibble: 1 × 2
##     min   max
##   <dbl> <dbl>
## 1     1    31
```

Adoption / missingness: sleep and weight flags

```
daily %>%
  group_by(dataset_window) %>%
  summarise(
    pct_days_sleep = mean(has_sleep_log_for_day, na.rm=TRUE),
    n_days_sleep = sum(has_sleep_log_for_day, na.rm=TRUE),
    n_days_no_sleep = sum(!has_sleep_log_for_day, na.rm = TRUE),
    pct_days_weight = mean(has_weight_log_for_day, na.rm=TRUE),
    n_days_weight = sum(has_weight_log_for_day, na.rm=TRUE),
    n_days_no_weight = sum(!has_weight_log_for_day, na.rm = TRUE),
      .groups = "drop"
    )
```

```
## # A tibble: 2 × 7
##   dataset_window     pct_days_sleep n_days_sleep n_days_no_sleep
pct_days_weight
##   <chr>                       <dbl>        <int>           <int>
<dbl>
## 1 2016-03-12_to_201…          0                0             433
0.0716
## 2 2016-04-12_to_201…          0.436          410             530
0.0713
## #    2 more variables: n_days_weight <int>, n_days_no_weight <int>
```

```
users %>%
  group_by(dataset_window) %>%
  summarise(
    pct_users_sleep = mean(has_sleep_log, na.rm=TRUE),
    pct_users_weight = mean(has_weight_log, na.rm=TRUE),
    .groups = "drop"
)
```

```
## # A tibble: 2 × 3
##   dataset_window           pct_users_sleep pct_users_weight
##   <chr>                              <dbl>            <dbl>
## 1 2016-03-12_to_2016-04-11           0                0.314
## 2 2016-04-12_to_2016-05-12           0.727            0.242
```

Interpretation notes: Sleep and weight logging show markedly different adoption patterns across the two observation windows.

In the first window (2016-03-12 to 2016-04-11), no valid sleep records are present, indicating that sleep tracking data is not available for this period. Weight logging is observed, but remains limited, with weight records recorded on a small fraction of days and by a minority of users.

In the second window (2016-04-12 to 2016-05-12), sleep logging becomes substantially more prevalent. Over 40% of days include sleep records, and approximately 73% of users log sleep at least once during the period. This suggests a meaningful increase in sleep feature usage and enables sleep-related analysis to be conducted reliably for this window only.

Weight logging remains comparatively sparse in both periods. Fewer than 8% of days include weight records, and fewer than one quarter of users log weight in either window. This indicates that weight tracking is an optional behavior adopted by a small subset of users, rather than a core engagement feature.

Range checks

```
daily %>% summarise(
  neg_steps = sum(total_steps < 0, na.rm=TRUE),
  huge_steps = sum(total_steps > 50000, na.rm=TRUE),
  sedentary_gt_day = sum(sedentary_minutes > 1440, na.rm=TRUE),
  .groups = "drop"
)
```

```
## # A tibble: 1 × 3
##   neg_steps huge_steps sedentary_gt_day
##       <int>      <int>            <int>
## 1         0          0                0
```

Temporal Activity Patterns

Hourly Activity, Steps, and Calories

This section examines intra-day activity patterns, focusing on how user behavior varies by hour of day and weekday. The goal is to identify consistent peaks and troughs in physical activity and energy expenditure.

The analysis complements the Tableau Hourly Rhythm dashboards by validating high-level patterns numerically and ensuring consistency across tools.

```
hourly %>% group_by(week_day, log_time) %>%
  summarise(
    average_steps = mean(total_steps, na.rm=TRUE),
    average_calories = mean(calories, na.rm=TRUE),
    average_intensity = mean(average_intensity, na.rm=TRUE),
    .groups = "drop"
  )
```

```
## # A tibble: 168 × 5
##    week_day log_time average_steps average_calories
average_intensity
```

```
##       <dbl> <time>           <dbl>            <dbl>
<dbl>
##  1        1 00:00            68.4             74.6
0.0533
##  2        1 01:00            39.5             71.6
0.0351
##  3        1 02:00            26.1             69.6
0.0223
##  4        1 03:00             5.43            67.3
0.00848
##  5        1 04:00             5.72            67.2
0.00815
##  6        1 05:00            16.0             68
0.0147
##  7        1 06:00            62.5             72.6
0.0440
##  8        1 07:00           156.              80.6          0.100

##  9        1 08:00           304.              93.5          0.185

## 10        1 09:00           393.             106.           0.249

## #    158 more rows
```

Interpretation notes: Overall, activity levels remain low during early morning hours, increase steadily through mid-morning, and peak during daytime and early evening hours. This pattern is consistent across weekdays, suggesting predictable daily routines among users.

Calories burned and total intensity closely follow step patterns, indicating that step count serves as a reasonable proxy for overall activity intensity.

Weekday vs Weekend Behavior

To understand behavioral differences between workdays and weekends, activity metrics were compared across these two categories.

```
daily %>% group_by(is_weekend) %>%
  summarise(
    average_steps = mean(total_steps, na.rm=TRUE),
    average_calories = mean(calories, na.rm=TRUE),
    average_activity = mean(rowSums(across(c(lightly_active_minutes,
fairly_active_minutes, very_active_minutes)), na.rm=TRUE),
na.rm=TRUE)/60,
    .groups = "drop"
)

## # A tibble: 2 × 4
##   is_weekend average_steps average_calories average_activity
##   <lgl>              <dbl>            <dbl>            <dbl>
```

```
## 1 FALSE                7453.              2302.               3.71
## 2 TRUE                 7188.              2277.               3.64
```

Interpretation notes: Activity levels are slightly higher on weekdays compared to weekends across all measured dimensions. On average, users record approximately 7,453 steps on weekdays, compared to 7,188 steps on weekends, indicating a modest reduction in movement during weekends. A similar pattern is observed for calorie expenditure, with weekday average calories burned (2,302) marginally exceeding weekend averages (2,277).

Overall activity minutes follow the same trend, with users accumulating slightly more active minutes on weekdays (223 minutes) than on weekends (219 minutes). While the differences between weekdays and weekends are not substantial, the consistency across steps, calories, and activity minutes suggests that users maintain a more active routine during the workweek.

These results indicate that daily structure associated with weekdays may encourage slightly higher physical activity, whereas weekends show a small but consistent decline in engagement.

Activity Distribution by Day of Week

This subsection explores how activity varies across individual weekdays, providing a more granular view beyond the weekday/weekend split.

```
daily %>% group_by(week_day) %>%
  arrange(week_day) %>%
  summarise(
    average_steps = mean(total_steps, na.rm=TRUE),
    average_calories = mean(calories, na.rm=TRUE),
    average_activity = mean(rowSums(across(c(lightly_active_minutes,
fairly_active_minutes, very_active_minutes)), na.rm=TRUE),
na.rm=TRUE)/60
  )

## # A tibble: 7 × 4
##   week_day average_steps average_calories average_activity
##      <dbl>         <dbl>            <dbl>            <dbl>
## 1        1         6607.            2227.             3.35
## 2        2         7541.            2298.             3.65
## 3        3         7719.            2339.             3.83
## 4        4         7548.            2321.             3.72
## 5        5         7268.            2224.             3.54
## 6        6         7188.            2325.             3.82
## 7        7         7752.            2326.             3.93
```

Interpretation notes: Daily activity patterns vary meaningfully across individual days of the week and help explain the earlier observation that average weekend activity is slightly lower than weekday activity overall.

Sunday (1) is consistently the least active day, with the lowest average steps (~6,607), calories (~2,227), and activity level. This low level of activity on Sunday substantially pulls down the overall weekend average.

In contrast, Saturday (7) is the most active day of the week, with the highest average steps (~7,752) and the highest overall activity level. This indicates that weekend behavior is not uniform: Saturday and Sunday reflect distinct activity patterns rather than a single “weekend effect.”

Activity increases from Monday into Tuesday and begins to decline by Wednesday, with lower step counts and activity levels continuing through Thursday before a modest rebound on Friday, where calorie expenditure and overall activity remain elevated despite slightly lower step counts.

Taken together, these results clarify why weekends appear less active when averaged as a whole: high activity on Saturday is offset by substantially lower activity on Sunday. Weekday routines, by contrast, produce more consistent activity levels across multiple consecutive days.

These findings suggest that interventions targeting weekend behavior may benefit from differentiating between Saturday and Sunday, rather than treating weekends as a single behavioral category.

Activity Composition and User-Level Engagement

Activity Type Distribution

This section explores how users distribute time across sedentary, lightly active, fairly active, and very active categories.

```
daily %>%
  summarise(
    average_lightly_active_minutes= mean(lightly_active_minutes,
na.rm=TRUE)/60,
    average_fairly_active_minutes = mean(fairly_active_minutes,
na.rm=TRUE)/60,
    average_very_active_minutes = mean(very_active_minutes,
na.rm=TRUE)/60,
    average_sedentary_minutes = mean(sedentary_minutes, na.rm=TRUE)/60
  )

## # A tibble: 1 × 4
##   average_lightly_active_minutes average_fairly_active_…¹
average_very_active_…²
##                            <dbl>                     <dbl>
<dbl>
## 1                           3.13                     0.227
```

```
0.331
## #    abbreviated names: ¹average_fairly_active_minutes,
## #   ²average_very_active_minutes
## #    1 more variable: average_sedentary_minutes <dbl>
```

Interpretation notes: Light activity accounts for the majority of non-sedentary movement, with users averaging approximately 3.1 hours of lightly active time per day, while fairly active time is minimal, averaging just 0.23 hours per day. This indicates that most daily movement occurs at low intensity, with relatively little time spent in sustained moderate activity.

User-Level Activity Variability

This section examines baseline daily activity metrics across all users, including steps, distance, calories burned, and activity. The goal is to understand typical daily behavior and variability across the population.

```
user_variability <- daily %>% group_by(Id, dataset_window) %>%
  summarise(
    average_steps = mean(total_steps, na.rm=TRUE),
    average_distance = mean(total_distance, na.rm=TRUE),
    average_calories = mean(calories, na.rm=TRUE),
    average_activity = mean(rowSums(across(c(lightly_active_minutes,
fairly_active_minutes, very_active_minutes)), na.rm=TRUE),
na.rm=TRUE)/60,
    step_variability = sd(total_steps, na.rm = TRUE),
    n_days = n(),
    .groups = "drop"
)
user_variability <- user_variability %>%
  mutate(
    step_cv = step_variability / average_steps
  )
user_variability
```

```
## # A tibble: 68 × 9
##            Id dataset_window     average_steps average_distance
average_calories
##         <dbl> <chr>                      <dbl>            <dbl>
<dbl>
##  1 1503960366 2016-03-12_to_201…        12275.             8.02
1893.
##  2 1503960366 2016-04-12_to_201…        12117.             7.81
1816.
##  3 1624580081 2016-03-12_to_201…         4093.             2.66
1389.
##  4 1624580081 2016-04-12_to_201…         5744.             3.91
1483.
##  5 1644430081 2016-03-12_to_201…         9275.             6.75
```

```
2916.
##  6 1644430081 2016-04-12_to_201…         7283.            5.30
2811.
##  7 1844505072 2016-03-12_to_201…         3972.            2.63
1727.
##  8 1844505072 2016-04-12_to_201…         2580.            1.71
1573.
##  9 1927972279 2016-03-12_to_201…         2377.            1.65
2373.
## 10 1927972279 2016-04-12_to_201…          916.            0.635
2173.
## #    58 more rows
## #    4 more variables: average_activity <dbl>, step_variability
<dbl>,
## #   n_days <int>, step_cv <dbl>
```

Interpretation notes: User-level summaries reveal substantial heterogeneity in both activity levels and consistency across the population. Average daily steps range widely, from fewer than 1,000 steps per day among the least active users to well over 12,000 steps per day among the most active. Average daily activity time shows a similar spread, varying from less than 1 hour to almost 6 hours per day.

Variability in daily steps also differs markedly across users. Some users exhibit relatively low day-to-day fluctuation, with coefficients of variation below 0.20, indicating stable and consistent activity patterns. In contrast, other users show very high variability, with coefficients of variation exceeding 1.0, meaning that their daily step counts fluctuate as much as or more than their average activity level.

Notably, higher average activity does not necessarily imply greater consistency. Several users with high average step counts still display moderate to high variability, while some lower-activity users exhibit comparatively stable routines. This suggests that activity volume and activity regularity represent distinct behavioral dimensions.

Overall, these results indicate that user engagement with physical activity is best characterized along two axes: how active users are on average, and how consistently they maintain that activity over time. This distinction supports the use of grouped or segmented analyses rather than per-user inspection when evaluating behavioral patterns and designing engagement strategies.

Group-Level Activity Variability

To better understand heterogeneity in user behavior, users were grouped into activity segments based on a combination of average activity level, consistency of daily engagement, and day-to-day variability in steps. This

segmentation allows patterns to be examined at a behavioral level rather than on a per-user basis, providing more actionable insight into engagement styles across the population.

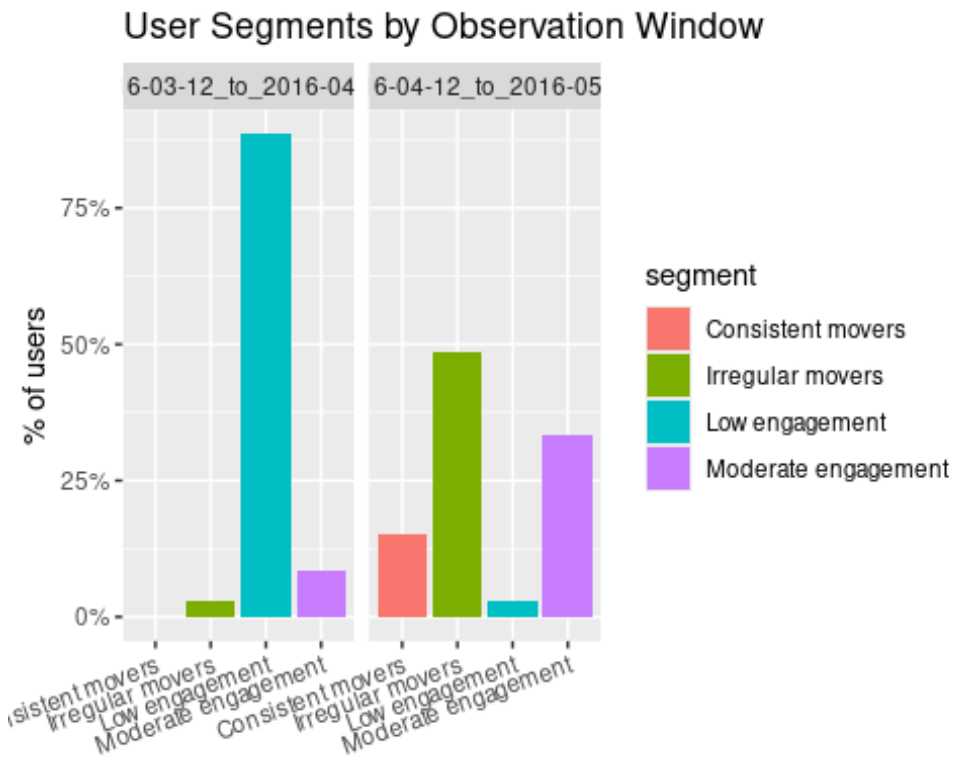
Segments were analyzed separately for each observation window to account for differences in data availability and feature adoption.

```
window_stats <- users %>%
  group_by(dataset_window) %>%
  summarise(
    med_steps = median(avg_steps, na.rm = TRUE),
    med_var = median(step_variability, na.rm = TRUE),
    .groups = "drop"
  )
users_seg_leaf <- users %>%
  left_join(window_stats, by = "dataset_window") %>%
  mutate(
    segment = case_when(
      active_days_pct < 0.50 ~ "Low engagement",
      active_days_pct >= 0.80 & avg_steps >= med_steps &
step_variability <= med_var ~ "Consistent movers",
      active_days_pct >= 0.50 & step_variability > med_var ~
"Irregular movers",
      TRUE ~ "Moderate engagement"
    )
  )
seg_summary_leaf <- users_seg_leaf %>%
  group_by(dataset_window, segment) %>%
  summarise(
    n_users = n(),
    avg_steps = mean(avg_steps, na.rm = TRUE),
    avg_active_days_pct = mean(active_days_pct, na.rm = TRUE),
    avg_variability = mean(step_variability, na.rm = TRUE),
    avg_sedentary = mean(avg_sedentary_minutes, na.rm = TRUE),
    pct_sleep_loggers = mean(has_sleep_log, na.rm = TRUE),
    pct_weight_loggers = mean(has_weight_log, na.rm = TRUE),
    .groups = "drop_last"
  ) %>%
  mutate(pct_users = n_users/sum(n_users)) %>%
  ungroup()%>%
  arrange(dataset_window, desc(pct_users))

seg_summary_leaf
```

```
## # A tibble: 7 × 10
##   dataset_window   segment n_users avg_steps avg_active_days_pct
avg_variability
##   <chr>            <chr>     <int>     <dbl>               <dbl>
<dbl>
## 1 2016-03-12_to_2… Low en…      31     6955.               0.349
```

```
3190.
## 2 2016-03-12_to_2… Modera…       3     6727.                0.860
2461.
## 3 2016-03-12_to_2… Irregu…       1     4093.                0.581
4502.
## 4 2016-04-12_to_2… Irregu…      16     9398.                0.954
4437.
## 5 2016-04-12_to_2… Modera…      11     4019.                0.906
2514.
## 6 2016-04-12_to_2… Consis…       5     9943.                0.994
2573.
## 7 2016-04-12_to_2… Low en…       1     3838                 0.129
2692.
## #    4 more variables: avg_sedentary <dbl>, pct_sleep_loggers
<dbl>,
## #   pct_weight_loggers <dbl>, pct_users <dbl>
```

```r
ggplot(seg_summary_leaf, aes(x = segment, y = pct_users, fill =
segment)) +
  geom_col() +
  facet_wrap(~ dataset_window) +
  scale_y_continuous(labels = percent) +
  labs(title = "User Segments by Observation Window", x = NULL, y = "%
of users") +
  theme(axis.text.x = element_text(angle = 20, hjust = 1))
```



Interpretation notes: In the first observation window (2016-03-12 to 2016-04-11), the user population is dominated by the Low engagement segment, which accounts for the vast majority of users. These users show relatively low consistency, with activity recorded on fewer than half of available days on average. Sleep logging is entirely absent in this window, while weight logging is present for a minority of users across segments. Due to limited variation in segment composition, behavioral differentiation in this window is minimal.

In the second observation window (2016-04-12 to 2016-05-12), user behavior becomes more differentiated. Irregular movers represent the largest segment, characterized by high average step counts and high consistency, but also elevated variability, indicating fluctuating daily routines. Consistent movers, though fewer in number, exhibit the highest overall engagement, with near-daily activity, relatively low variability, and the highest rates of both sleep and weight logging. Moderate engagement users show high consistency but lower overall activity levels, while Low engagement users remain rare in this period.

Across segments, adoption of sleep logging is substantially higher in the second window and is most prevalent among users with higher consistency and activity levels. Weight logging remains limited overall but is more common among the most engaged users, suggesting that deeper feature adoption is associated with sustained and consistent activity behavior.

Relationship between Activity Types and Calories

Activity Time by Type vs Daily Calories Burned

This section examines how daily calorie expenditure relates to different components of daily behavior. The goal is to identify which behavioral signals best align with energy expenditure and which associations are weak or negligible.

```
cor_by_window <- daily %>%
  filter(very_active_minutes > 0) %>%
  group_by(dataset_window) %>%
  summarise(
    cor_very_active_minutes_calories = cor(very_active_minutes,
calories, use = "complete.obs"),
    n_days = n(),
    .groups = "drop"
  )

cor_by_window
```

```
## # A tibble: 2 × 3
##   dataset_window           cor_very_active_minutes_calories n_days
##   <chr>                                               <dbl>  <int>
## 1 2016-03-12_to_2016-04-11                            0.563    211
## 2 2016-04-12_to_2016-05-12                            0.578    531
```

```
lm_by_window <- daily %>%
  group_by(dataset_window) %>%
  do(model = lm(calories ~ very_active_minutes, data = .))

print(summary(lm_by_window$model[[1]]), digits = 4)
```

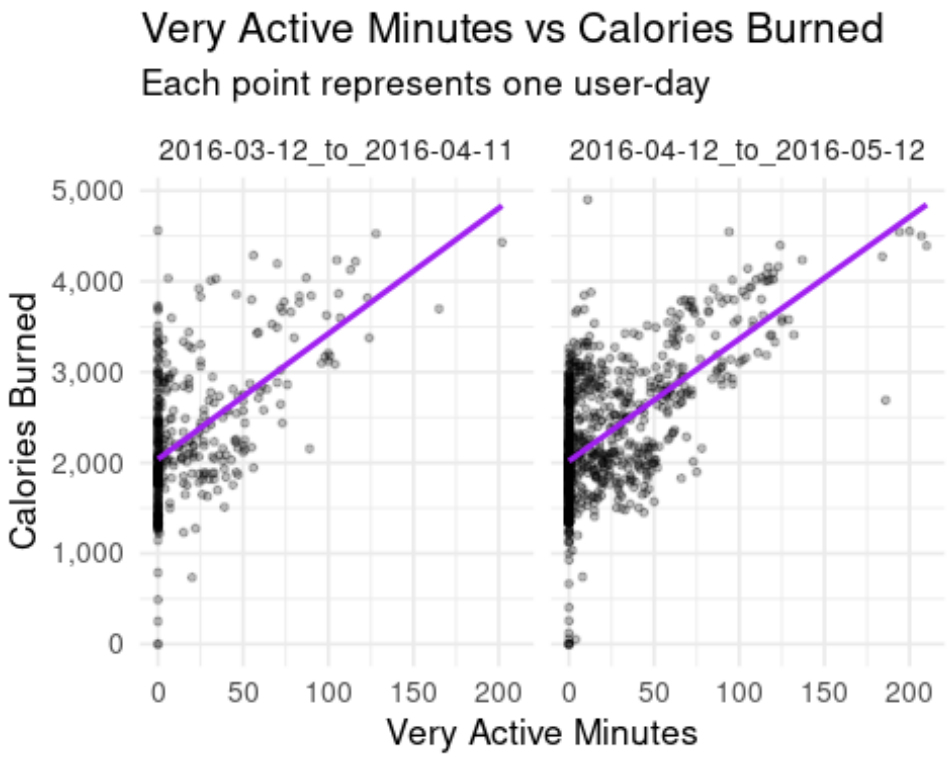
```
##
## Call:
## lm(formula = calories ~ very_active_minutes, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2039.6  -407.4  -110.6   353.4  2522.4
##
## Coefficients:
##                     Estimate Std. Error t value            Pr(>|t|)

## (Intercept)         2039.619     34.588   58.97 <0.0000000000000002
***
## very_active_minutes   13.837      1.021   13.55 <0.0000000000000002
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 621.8 on 431 degrees of freedom
## Multiple R-squared:  0.2989, Adjusted R-squared:  0.2972
## F-statistic: 183.7 on 1 and 431 DF,  p-value: < 0.00000000000000022
```

```
print(summary(lm_by_window$model[[2]]), digits = 4)
```

```
##
## Call:
## lm(formula = calories ~ very_active_minutes, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2020.47  -376.00   -39.85   387.07  2733.27
##
## Coefficients:
##                      Estimate Std. Error t value            Pr(>|
t|)
## (Intercept)         2018.6118    21.9701   91.88
<0.0000000000000002 ***
## very_active_minutes   13.4656     0.5625   23.94
<0.0000000000000002 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 566.1 on 938 degrees of freedom
## Multiple R-squared:  0.3793, Adjusted R-squared:  0.3786
## F-statistic: 573.1 on 1 and 938 DF,  p-value: < 0.00000000000000022
```

```
ggplot(daily, aes(x = very_active_minutes, y = calories)) +
  geom_point(alpha = 0.25, size = 1) +
  geom_smooth(method = "lm", se = FALSE, color = "purple", linewidth =
1) +
  facet_wrap(~ dataset_window) +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  labs(
    title = "Very Active Minutes vs Calories Burned",
    subtitle = "Each point represents one user-day",
    x = "Very Active Minutes",
    y = "Calories Burned"
  ) +
  theme_minimal(base_size = 13)
```




```
cor_by_window <- daily %>%
  filter(lightly_active_minutes > 0) %>%
  group_by(dataset_window) %>%
  summarise(
    cor_lightly_active_minutes_calories = cor(lightly_active_minutes,
```

```
calories, use = "complete.obs"),
    n_days = n(),
    .groups = "drop"
  )

cor_by_window
```

```
## # A tibble: 2 × 3
##   dataset_window          cor_lightly_active_minutes_calories
n_days
##   <chr>                                                 <dbl>
<int>
## 1 2016-03-12_to_2016-04-11                              0.241
366
## 2 2016-04-12_to_2016-05-12                              0.194
856
```

```
lm_by_window <- daily %>%
  group_by(dataset_window) %>%
  do(model = lm(calories ~ lightly_active_minutes, data = .))

print(summary(lm_by_window$model[[1]]), digits = 4)
```

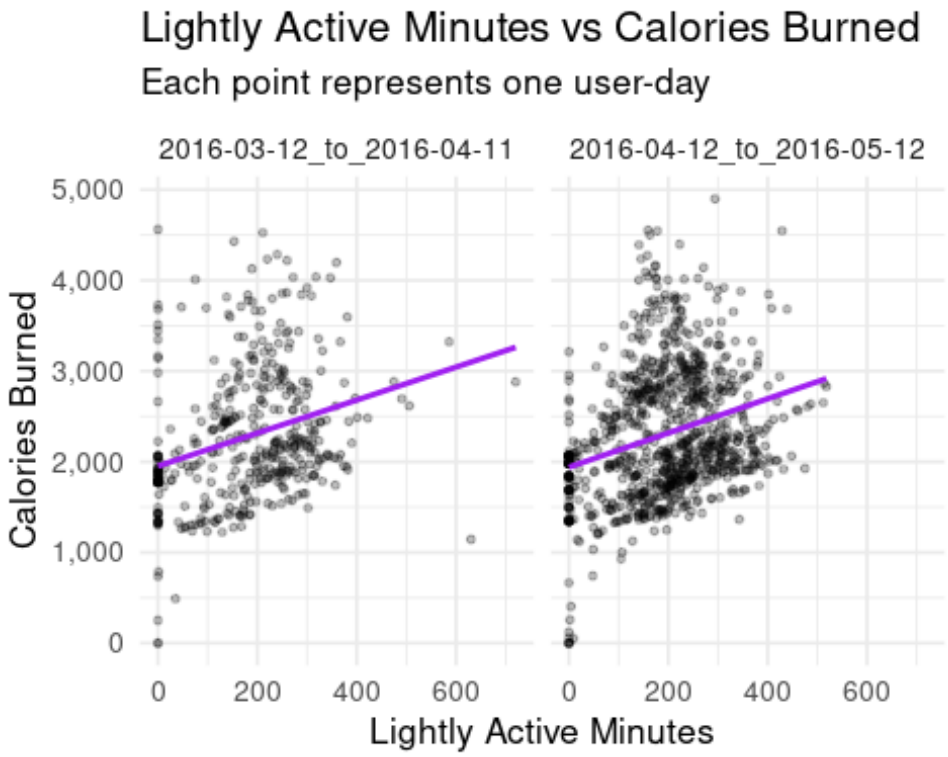
```
##
## Call:
## lm(formula = calories ~ lightly_active_minutes, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1954.6  -518.3  -162.7   348.2  2610.0
##
## Coefficients:
##                          Estimate Std. Error t value
Pr(>|t|)
## (Intercept)            1952.0123    60.7617  32.126 <
0.0000000000000002 ***
## lightly_active_minutes    1.8200     0.2828   6.435
0.000000000328 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 709.3 on 431 degrees of freedom
## Multiple R-squared:  0.08767,    Adjusted R-squared:  0.08555
## F-statistic: 41.41 on 1 and 431 DF,  p-value: 0.0000000003283
```

```
print(summary(lm_by_window$model[[2]]), digits = 4)
```

```
##
## Call:
## lm(formula = calories ~ lightly_active_minutes, data = .)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1940.0  -544.4  -141.2   457.7  2405.5
##
## Coefficients:
##                          Estimate Std. Error t value
Pr(>|t|)
## (Intercept)            1939.9514    45.5870  42.555
<0.0000000000000002 ***
## lightly_active_minutes    1.8861     0.2058   9.166
<0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 688.4 on 938 degrees of freedom
## Multiple R-squared:  0.08221,    Adjusted R-squared:  0.08123
## F-statistic: 84.02 on 1 and 938 DF,  p-value: < 0.00000000000000022
```

```
ggplot(daily, aes(x = lightly_active_minutes, y = calories)) +
  geom_point(alpha = 0.25, size = 1) +
  geom_smooth(method = "lm", se = FALSE, color = "purple", linewidth =
1) +
  facet_wrap(~ dataset_window) +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  labs(
    title = "Lightly Active Minutes vs Calories Burned",
    subtitle = "Each point represents one user-day",
    x = "Lightly Active Minutes",
    y = "Calories Burned"
  ) +
  theme_minimal(base_size = 13)
```




```r
cor_by_window <- daily %>%
  filter(fairly_active_minutes > 0) %>%
  group_by(dataset_window) %>%
  summarise(
    cor_fairy_active_minutes_calories = cor(fairly_active_minutes,
calories, use = "complete.obs"),
    n_days = n(),
    .groups = "drop"
  )

cor_by_window
```

```
## # A tibble: 2 × 3
##   dataset_window          cor_fairy_active_minutes_calories n_days
##   <chr>                                               <dbl>  <int>
## 1 2016-03-12_to_2016-04-11                            0.278    225
## 2 2016-04-12_to_2016-05-12                            0.0686   556
```

```r
lm_by_window <- daily %>%
  group_by(dataset_window) %>%
  do(model = lm(calories ~ fairly_active_minutes, data = .))

print(summary(lm_by_window$model[[1]]), digits = 4)
```

```
##
## Call:
## lm(formula = calories ~ fairly_active_minutes, data = .)
```

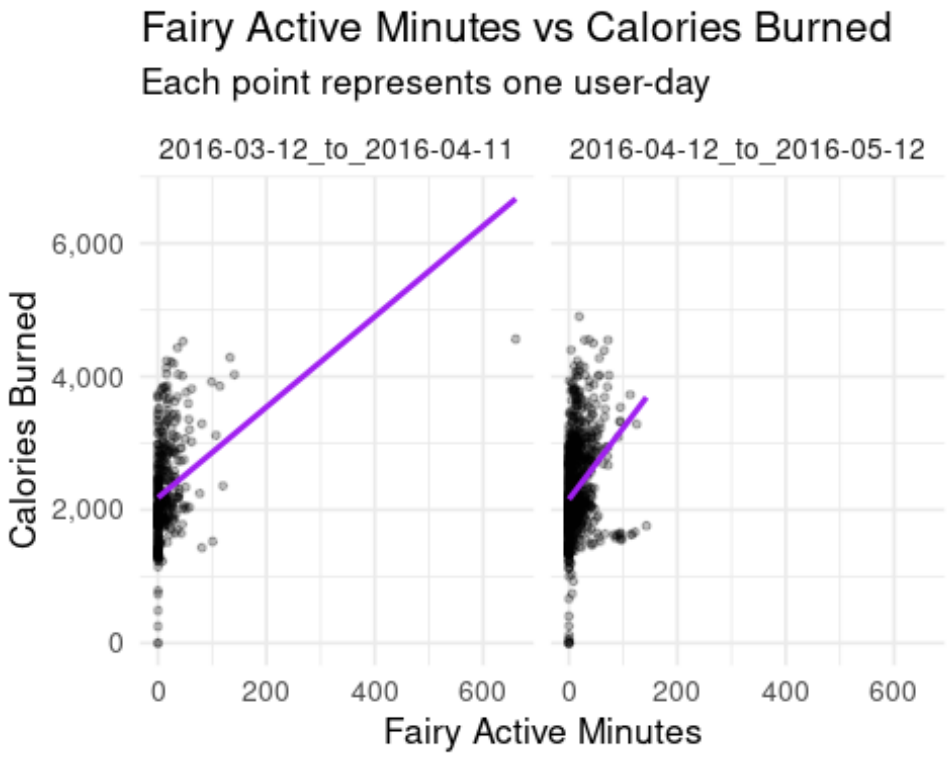
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2182.7  -406.7  -144.2   435.8  2030.7
##
## Coefficients:
##                       Estimate Std. Error t value            Pr(>|
t|)
## (Intercept)           2182.690     35.781  61.002 <
0.0000000000000002 ***
## fairly_active_minutes    6.796      0.906   7.501
0.000000000000364 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 698.4 on 431 degrees of freedom
## Multiple R-squared:  0.1155, Adjusted R-squared:  0.1134
## F-statistic: 56.27 on 1 and 431 DF,  p-value: 0.0000000000003641
```

```
print(summary(lm_by_window$model[[2]]), digits = 4)
```

```
##
## Call:
## lm(formula = calories ~ fairly_active_minutes, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2158.55  -434.52   -97.05   471.74  2538.27
##
## Coefficients:
##                       Estimate Std. Error t value            Pr(>|
t|)
## (Intercept)            2158.55      27.05  79.813
<0.0000000000000002 ***
## fairly_active_minutes    10.69       1.12   9.548
<0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 686 on 938 degrees of freedom
## Multiple R-squared:  0.08858,    Adjusted R-squared:  0.08761
## F-statistic: 91.16 on 1 and 938 DF,  p-value: < 0.00000000000000022
```

```
ggplot(daily, aes(x = fairly_active_minutes, y = calories)) +
  geom_point(alpha = 0.25, size = 1) +
  geom_smooth(method = "lm", se = FALSE, color = "purple", linewidth =
1) +
  facet_wrap(~ dataset_window) +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  labs(
```

```
    title = "Fairy Active Minutes vs Calories Burned",
    subtitle = "Each point represents one user-day",
    x = "Fairy Active Minutes",
    y = "Calories Burned"
  ) +
  theme_minimal(base_size = 13)
```




```
cor_by_window <- daily %>%
  filter(sedentary_minutes > 0) %>%
  group_by(dataset_window) %>%
  summarise(
    cor_sedentary_minutes_calories = cor(sedentary_minutes, calories,
use = "complete.obs"),
    n_days = n(),
    .groups = "drop"
  )

cor_by_window
```

```
## # A tibble: 2 × 3
##   dataset_window          cor_sedentary_minutes_calories n_days
##   <chr>                                            <dbl>  <int>
## 1 2016-03-12_to_2016-04-11                        -0.168    433
## 2 2016-04-12_to_2016-05-12                        -0.118    939
```

```
lm_by_window <- daily %>%
  group_by(dataset_window) %>%
```

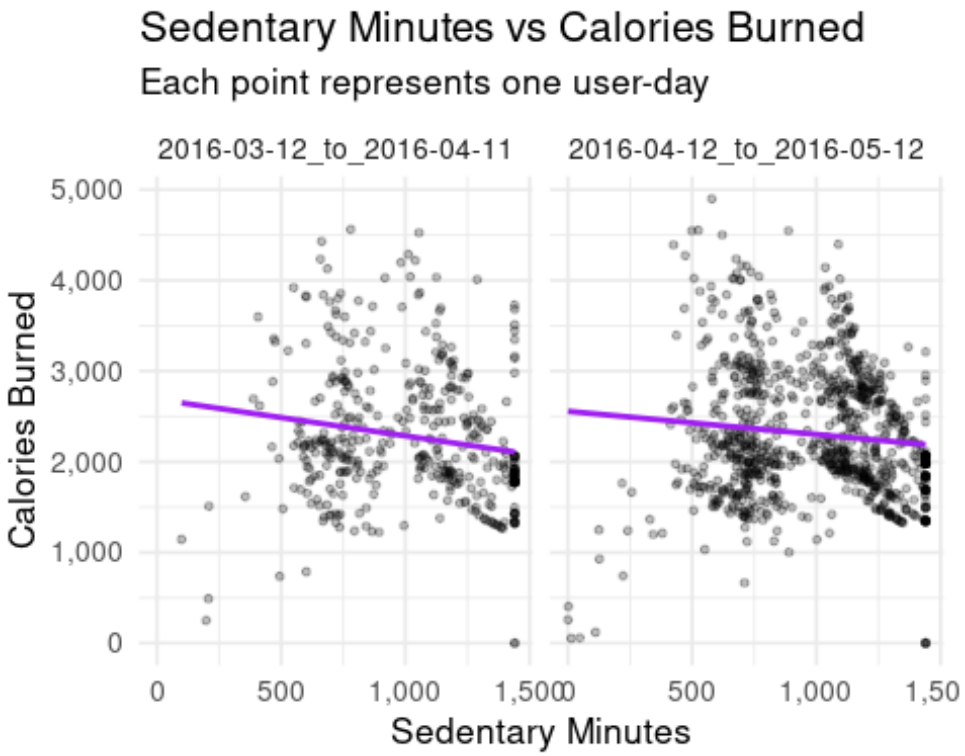
```r
  do(model = lm(calories ~ sedentary_minutes, data = .))

print(summary(lm_by_window$model[[1]]), digits = 4)
```

```
##
## Call:
## lm(formula = calories ~ sedentary_minutes, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2357.6  -397.0  -187.7   423.1  2263.1
##
## Coefficients:
##                    Estimate Std. Error t value             Pr(>|t|)

## (Intercept)       2688.5020   121.6154  22.107 < 0.0000000000000002
***
## sedentary_minutes   -0.4034     0.1138  -3.546             0.000434
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 732 on 431 degrees of freedom
## Multiple R-squared:  0.02834,    Adjusted R-squared:  0.02609
## F-statistic: 12.57 on 1 and 431 DF,  p-value: 0.0004343
```

```r
print(summary(lm_by_window$model[[2]]), digits = 4)
```

```
##
## Call:
## lm(formula = calories ~ sedentary_minutes, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2501.1  -469.9  -150.2   512.5  2491.3
##
## Coefficients:
##                     Estimate Std. Error t value             Pr(>|
t|)
## (Intercept)       2556.37257   80.16884  31.887 <
0.0000000000000002 ***
## sedentary_minutes   -0.25500    0.07739  -3.295
0.00102 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 714.4 on 938 degrees of freedom
## Multiple R-squared:  0.01144,    Adjusted R-squared:  0.01039
## F-statistic: 10.86 on 1 and 938 DF,  p-value: 0.001021
```

```r
ggplot(daily, aes(x = sedentary_minutes, y = calories)) +
  geom_point(alpha = 0.25, size = 1) +
  geom_smooth(method = "lm", se = FALSE, color = "purple", linewidth =
1) +
  facet_wrap(~ dataset_window) +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  labs(
    title = "Sedentary Minutes vs Calories Burned",
    subtitle = "Each point represents one user-day",
    x = "Sedentary Minutes",
    y = "Calories Burned"
  ) +
  theme_minimal(base_size = 13)
```



Interpretation notes: Across both observation windows, calories show the strongest positive association with very active minutes (correlations ~0.56–0.58). In regression models, very active minutes explain a meaningful share of variation in calories ($R^2 \approx 0.30$ in the first window and ≈ 0.38 in the second), with a consistent positive slope (roughly 13–14 additional calories per extra very active minute).

By contrast, calories have only weak-to-moderate relationships with lightly active minutes (correlations ~0.19–0.24) and fairly active minutes (correlations ~0.07–0.28), and these models explain substantially less variance (generally under ~0.12 R^2). Sedentary minutes show a small negative association with calories in both windows (correlations around

−0.12 to −0.17) with very low explanatory power, indicating sedentary time alone is not a strong driver of calorie variation.

Overall, these results suggest that calorie expenditure is most closely tied to higher-intensity activity, while light activity and sedentary time are much weaker predictors.

Sleep and Weight Logging Behavior

Sleep Logging Adoption

This section summarizes sleep logging behavior across time periods, focusing on both the frequency of sleep logs and user participation.

```
users %>%
  filter(sleep_rows > 0) %>%
  group_by(dataset_window, Id) %>%
  summarise(
    pct_sleep_logs_per_period = sleep_rows/31,
    pct_sleep_logs_per_active_days = sleep_rows/days_with_activity,
    .groups = "drop"
  )
```

```
## # A tibble: 24 × 4
##    dataset_window              Id pct_sleep_logs_per_p…¹
pct_sleep_logs_per_a…²
##    <chr>                    <dbl>                   <dbl>
<dbl>
##  1 2016-04-12_to_2016-05-12 1.50e9                  0.806
0.806
##  2 2016-04-12_to_2016-05-12 1.64e9                  0.129
0.133
##  3 2016-04-12_to_2016-05-12 1.84e9                  0.0968
0.0968
##  4 2016-04-12_to_2016-05-12 1.93e9                  0.161
0.161
##  5 2016-04-12_to_2016-05-12 2.03e9                  0.903
0.903
##  6 2016-04-12_to_2016-05-12 2.32e9                  0.0323
0.0323
##  7 2016-04-12_to_2016-05-12 2.35e9                  0.484
0.833
##  8 2016-04-12_to_2016-05-12 3.98e9                  0.903
0.933
##  9 2016-04-12_to_2016-05-12 4.02e9                  0.258
0.258
## 10 2016-04-12_to_2016-05-12 4.32e9                  0.839
0.839
## #    14 more rows
```

```
## #    abbreviated names: ¹pct_sleep_logs_per_period,
## #   ²pct_sleep_logs_per_active_days
```

```
daily %>%
  filter(dataset_window == "2016-04-12_to_2016-05-12") %>%
  group_by(log_date) %>%
  summarise(
    n_users = n(),
    n_sleep_logs = sum(has_sleep_log_for_day, na.rm = TRUE),
    pct_users = n_sleep_logs/sum(n_users),
    .groups = "drop"
  ) %>%
  arrange(log_date)
```

```
## # A tibble: 31 × 4
##    log_date   n_users n_sleep_logs pct_users
##    <date>       <int>        <int>     <dbl>
##  1 2016-04-12      33           13     0.394
##  2 2016-04-13      33           14     0.424
##  3 2016-04-14      33           13     0.394
##  4 2016-04-15      33           17     0.515
##  5 2016-04-16      32           14     0.438
##  6 2016-04-17      32           12     0.375
##  7 2016-04-18      32           10     0.312
##  8 2016-04-19      32           14     0.438
##  9 2016-04-20      32           15     0.469
## 10 2016-04-21      32           15     0.469
## #    21 more rows
```

Interpretation notes: Sleep logging behavior in the second observation window (2016-04-12 to 2016-05-12) shows substantial variation across users, alongside moderate but stable day-level adoption.

At the user level, sleep logging consistency ranges from very low to near-complete coverage. Several users log sleep on fewer than 10–15% of days, while others log sleep on over 90%, and in some cases 100%, of days in the period. For most users, the proportion of sleep logs relative to total days closely matches the proportion relative to active days, indicating that sleep logging tends to occur consistently on days when users are active, rather than sporadically.

At the daily level, sleep logging adoption is relatively stable across the observation window. On most days, approximately 40–50% of active users record sleep data. Adoption fluctuates modestly over time, with occasional peaks above 50%, but no sustained upward or downward trend is observed. Toward the end of the window, the number of active users declines, but the proportion logging sleep remains broadly consistent.

Taken together, these results indicate that sleep tracking is a habitual behavior for a subset of users, rather than a feature that is uniformly

adopted across the population. Sleep logging appears to be driven by individual-level engagement patterns rather than by date-specific or short-term effects, reinforcing the importance of user segmentation when interpreting sleep-related metrics.

Calories vs Time in Bed

To assess whether time in bed quantity alone meaningfully relates to energy output, sleep duration (measured as time spent in bed) is examined next.

```
cor_by_window <- daily %>%
  filter(dataset_window == "2016-04-12_to_2016-05-12") %>%
  summarise(
    cor_in_bed_calories = cor(time_in_bed, calories, use =
"complete.obs"),
    n_days = n(),
    .groups = "drop"
  )

cor_by_window
```

```
## # A tibble: 1 × 2
##   cor_in_bed_calories n_days
##                 <dbl>  <int>
## 1              -0.135    940
```

```
lm_in_bed_cal <- lm(calories ~ time_in_bed, data = daily)
summary(lm_in_bed_cal)
```

```
##
## Call:
## lm(formula = calories ~ time_in_bed, data = daily)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2242.6  -540.7  -174.5   506.4  2584.1
##
## Coefficients:
##              Estimate Std. Error t value             Pr(>|t|)
## (Intercept) 2757.0757   138.8987  19.850 < 0.0000000000000002 ***
## time_in_bed   -0.8022     0.2919  -2.748              0.00626 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 752.4 on 408 degrees of freedom
##   (963 observations deleted due to missingness)
## Multiple R-squared:  0.01817,    Adjusted R-squared:  0.01577
## F-statistic: 7.552 on 1 and 408 DF,  p-value: 0.006262
```

```
ggplot(daily, aes(x = time_in_bed, y = calories)) +
  geom_point(alpha = 0.25, size = 1) +
```

```
  geom_smooth(method = "lm", se = FALSE, color = "purple", linewidth =
1) +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  labs(
    title = "Minutes in Bed vs Calories Burned",
    subtitle = "Each point represents one user-day",
    x = "Total Minutes in Bed",
    y = "Calories Burned"
  ) +
  theme_minimal(base_size = 13)
```

Interpretation notes: Time spent in bed shows a weak negative relationship with calories burned. The correlation between time in bed and daily calories is −0.13, indicating a very small inverse association. The linear model confirms this pattern, with time in bed explaining less than 2% of the variance in calories burned ($R^2 \approx 0.02$).

While the estimated slope is statistically significant, its magnitude is small: each additional minute spent in bed is associated with a decrease of less than 1 calorie, suggesting minimal practical impact. The low explanatory power of the model and substantial residual variation indicate that sleep duration alone is not a meaningful predictor of daily calorie expenditure.

Overall, these results suggest that calories burned are driven primarily by activity-related factors, rather than by sleep quantity, reinforcing the importance of movement intensity and duration over passive time measures.

Weight Logging Behavior

Weight logging behavior was analyzed separately due to its sparse and irregular nature.

```
users %>%
  filter(weight_rows > 0) %>%
  group_by(dataset_window, Id) %>%
  summarise(
    pct_weight_logs_per_period = weight_rows/31,
    pct_weight_logs_per_active_days = weight_rows/days_with_activity,
    .groups = "drop"
  )
```

```
## # A tibble: 19 × 4
##    dataset_window              Id pct_weight_logs_per_…¹
pct_weight_logs_per_…²
##    <chr>                    <dbl>                   <dbl>
<dbl>
##  1 2016-03-12_to_2016-04-11 1.50e9                  0.0323
0.0556
##  2 2016-03-12_to_2016-04-11 1.93e9                  0.0323
0.0909
##  3 2016-03-12_to_2016-04-11 2.35e9                  0.0323
0.0714
##  4 2016-03-12_to_2016-04-11 2.87e9                  0.0645
0.182
##  5 2016-03-12_to_2016-04-11 2.89e9                  0.0323
0.125
##  6 2016-03-12_to_2016-04-11 4.45e9                  0.0323
0.0714
##  7 2016-03-12_to_2016-04-11 4.56e9                  0.0323
0.0909
##  8 2016-03-12_to_2016-04-11 4.70e9                  0.0323
0.0714
##  9 2016-03-12_to_2016-04-11 6.96e9                  0.419
1
## 10 2016-03-12_to_2016-04-11 8.25e9                  0.0323
0.0909
## 11 2016-03-12_to_2016-04-11 8.88e9                  0.258
0.727
## 12 2016-04-12_to_2016-05-12 1.50e9                  0.0645
0.0645
## 13 2016-04-12_to_2016-05-12 1.93e9                  0.0323
0.0323
## 14 2016-04-12_to_2016-05-12 2.87e9                  0.0645
```

```
0.0645
## 15 2016-04-12_to_2016-05-12 4.32e9                0.0645
0.0645
## 16 2016-04-12_to_2016-05-12 4.56e9                0.161
0.161
## 17 2016-04-12_to_2016-05-12 5.58e9                0.0323
0.0333
## 18 2016-04-12_to_2016-05-12 6.96e9                0.968
0.968
## 19 2016-04-12_to_2016-05-12 8.88e9                0.774
0.774
## #    abbreviated names: ¹pct_weight_logs_per_period,
## #   ²pct_weight_logs_per_active_days
```

```
daily %>%
  group_by(dataset_window, log_date) %>%
  summarise(
    n_users = n(),
    n_weight_logs = sum(has_weight_log_for_day, na.rm = TRUE),
    pct_users = n_weight_logs/sum(n_users),
    .groups = "drop"
  ) %>%
  arrange(log_date)
```

```
## # A tibble: 62 × 5
##    dataset_window           log_date   n_users n_weight_logs
pct_users
##    <chr>                    <date>       <int>         <int>
<dbl>
##  1 2016-03-12_to_2016-04-11 2016-03-12       2             0
0
##  2 2016-03-12_to_2016-04-11 2016-03-13       2             0
0
##  3 2016-03-12_to_2016-04-11 2016-03-14       2             0
0
##  4 2016-03-12_to_2016-04-11 2016-03-15       2             0
0
##  5 2016-03-12_to_2016-04-11 2016-03-16       2             0
0
##  6 2016-03-12_to_2016-04-11 2016-03-17       2             0
0
##  7 2016-03-12_to_2016-04-11 2016-03-18       2             0
0
##  8 2016-03-12_to_2016-04-11 2016-03-19       2             0
0
##  9 2016-03-12_to_2016-04-11 2016-03-20       2             0
0
## 10 2016-03-12_to_2016-04-11 2016-03-21       2             0
0
## #    52 more rows
```

Interpretation notes: Weight logging is sparse and highly concentrated among a small subset of users across both observation windows.

At the user level, most users log weight on only a very small fraction of days, often below 5–10%. A few users stand out as consistent weight loggers, recording weight on a majority of days and, in some cases, on nearly all active days. This results in a highly skewed distribution, where a small number of users account for a disproportionate share of weight records. For users who do log weight, the proportion of weight logs relative to total days is generally similar to the proportion relative to active days, indicating that weight logging tends to occur on days when users are otherwise engaged.

At the daily level, weight logging adoption is extremely low. On most days, fewer than 10% of active users record a weight measurement, and many days show no weight logs at all, particularly in the earlier part of the first observation window. While the second window exhibits slightly more regular logging, daily adoption remains consistently low, rarely exceeding 10–15% of users even on peak days.

Taken together, these results indicate that weight tracking is a non-core feature for the majority of users and functions as a niche or intentional behavior rather than a habitual daily practice. As a result, weight-related metrics are best interpreted as reflective of a highly engaged subset of users rather than representative of overall population behavior.

Relationships Between Activity, Calories, and Weight

Activity vs Calories

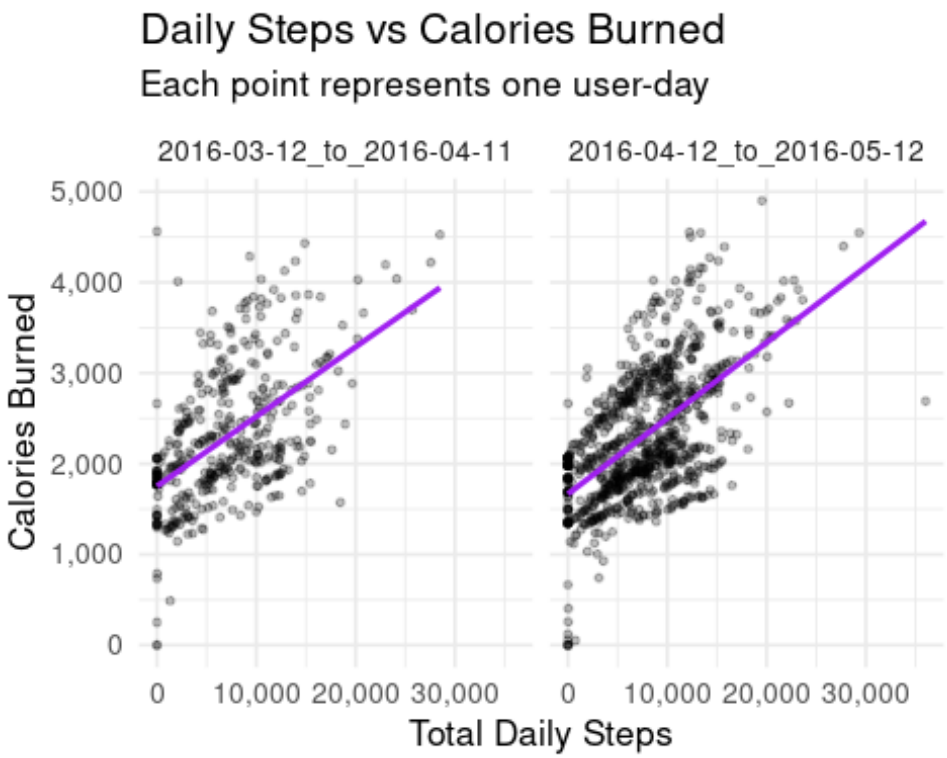
This section explores the relationship between physical activity metrics and average daily calorie expenditure.

```
cor_by_window <- daily %>%
  group_by(dataset_window) %>%
  summarise(
    cor_steps_calories = cor(total_steps, calories, use =
"complete.obs"),
    n_days = n(),
    .groups = "drop"
  )

cor_by_window
```

```
## # A tibble: 2 × 3
##   dataset_window          cor_steps_calories n_days
##   <chr>                                <dbl>  <int>
```

```
## 1 2016-03-12_to_2016-04-11              0.558    433
## 2 2016-04-12_to_2016-05-12              0.592    940

lm_by_window <- daily %>%
  group_by(dataset_window) %>%
  do(model = lm(calories ~ total_steps, data = .))

print(summary(lm_by_window$model[[1]]), digits = 4)

##
## Call:
## lm(formula = calories ~ total_steps, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1752.60  -428.60    21.12   338.33  2809.40
##
## Coefficients:
##                Estimate  Std. Error t value            Pr(>|t|)
## (Intercept) 1752.603619   47.719090   36.73 <0.0000000000000002 ***
## total_steps    0.076795    0.005494   13.98 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 616 on 431 degrees of freedom
## Multiple R-squared:  0.3119, Adjusted R-squared:  0.3103
## F-statistic: 195.4 on 1 and 431 DF,  p-value: < 0.00000000000000022

print(summary(lm_by_window$model[[2]]), digits = 4)

##
## Call:
## lm(formula = calories ~ total_steps, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1983.81  -373.52   -10.63   431.50  1864.81
##
## Coefficients:
##                Estimate  Std. Error t value            Pr(>|t|)
## (Intercept) 1665.742677   34.099664   48.85 <0.0000000000000002 ***
## total_steps    0.083513    0.003716   22.47 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 579.3 on 938 degrees of freedom
## Multiple R-squared:   0.35,  Adjusted R-squared:  0.3493
## F-statistic:   505 on 1 and 938 DF,  p-value: < 0.00000000000000022

ggplot(daily, aes(x = total_steps, y = calories)) +
  geom_point(alpha = 0.25, size = 1) +
```

```r
  geom_smooth(method = "lm", se = FALSE, color = "purple", linewidth =
1) +
  facet_wrap(~ dataset_window) +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  labs(
    title = "Daily Steps vs Calories Burned",
    subtitle = "Each point represents one user-day",
    x = "Total Daily Steps",
    y = "Calories Burned"
  ) +
  theme_minimal(base_size = 13)
```



Interpretation notes: Daily step count shows a moderate positive association with calories burned in both observation windows. The correlation between steps and calories is 0.56 in the first window (2016-03-12 to 2016-04-11) and increases slightly to 0.59 in the second window (2016-04-12 to 2016-05-12), indicating a consistent relationship across time.

Linear regression results reinforce this pattern. In the first window, daily steps explain approximately 31% of the variance in calories burned (R^2 = 0.31), while in the second window, explanatory power increases to 35% (R^2 = 0.35). In both models, the relationship between steps and calories is statistically significant ($p < 0.001$).

The estimated slope coefficients suggest that each additional step is associated with an increase of approximately 0.08 calories, highlighting that step count is a meaningful—but incomplete—proxy for daily energy expenditure. The substantial unexplained variance indicates that factors beyond step volume, such as activity intensity and individual metabolic differences, also play an important role in determining total calories burned.

Weight vs Activity and Calories

```
cor_by_window <- daily %>%
  filter(weight_kg>40) %>%
  group_by(dataset_window) %>%
  summarise(
    cor_weight_calories = cor(weight_kg, calories, use =
"complete.obs"),
    n_days = n(),
    .groups = "drop"
  )

cor_by_window
```

```
## # A tibble: 2 × 3
##   dataset_window           cor_weight_calories n_days
##   <chr>                                  <dbl>  <int>
## 1 2016-03-12_to_2016-04-11               0.430     31
## 2 2016-04-12_to_2016-05-12               0.664     67
```

```
lm_by_window <- daily %>%
  group_by(dataset_window) %>%
  do(model = lm(calories ~ weight_kg, data = .))

print(summary(lm_by_window$model[[1]]), digits = 4)
```

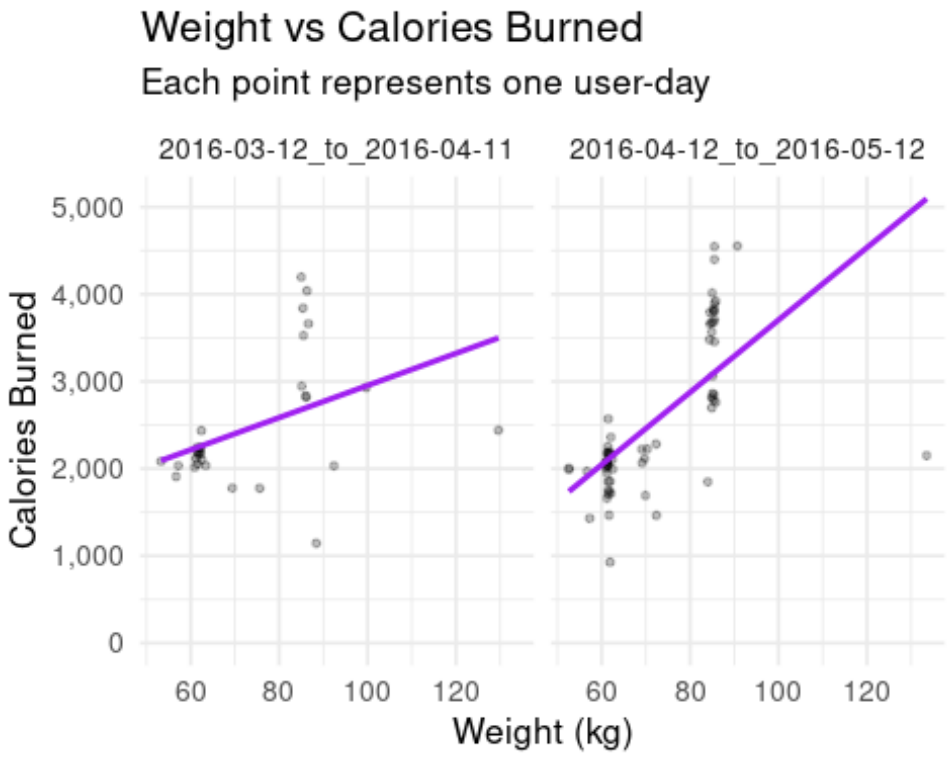
```
##
## Call:
## lm(formula = calories ~ weight_kg, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1595.73  -204.18   -59.89   130.56  1518.90
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1111.331    540.054   2.058   0.0487 *
## weight_kg     18.421      7.178   2.566   0.0157 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 660.7 on 29 degrees of freedom
##   (402 observations deleted due to missingness)
```

```
## Multiple R-squared:  0.1851, Adjusted R-squared:  0.157
## F-statistic: 6.585 on 1 and 29 DF,  p-value: 0.01571
```

```
print(summary(lm_by_window$model[[2]]), digits = 4)
```

```
##
## Call:
## lm(formula = calories ~ weight_kg, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2943.04  -291.38   -33.57   385.55  1443.74
##
## Coefficients:
##             Estimate Std. Error t value       Pr(>|t|)
## (Intercept) -442.823    424.435  -1.043          0.301
## weight_kg     41.475      5.786   7.167 0.000000000877 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 654.5 on 65 degrees of freedom
##   (873 observations deleted due to missingness)
## Multiple R-squared:  0.4415, Adjusted R-squared:  0.4329
## F-statistic: 51.37 on 1 and 65 DF,  p-value: 0.000000000877
```

```
ggplot(daily, aes(x = weight_kg, y = calories)) +
  geom_point(alpha = 0.25, size = 1) +
  geom_smooth(method = "lm", se = FALSE, color = "purple", linewidth =
1) +
  facet_wrap(~ dataset_window) +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  labs(
    title = "Weight vs Calories Burned",
    subtitle = "Each point represents one user-day",
    x = "Weight (kg)",
    y = "Calories Burned"
  ) +
  theme_minimal(base_size = 13)
```




```
cor_by_window <- daily %>%
  filter(weight_kg>40) %>%
  group_by(dataset_window) %>%
  summarise(
    cor_weight_steps = cor(weight_kg, total_steps, use =
"complete.obs"),
    n_days = n(),
    .groups = "drop"
  )

cor_by_window
```

```
## # A tibble: 2 × 3
##   dataset_window           cor_weight_steps n_days
##   <chr>                               <dbl>  <int>
## 1 2016-03-12_to_2016-04-11           -0.194     31
## 2 2016-04-12_to_2016-05-12            0.289     67
```

```
lm_by_window <- daily %>%
  group_by(dataset_window) %>%
  do(model = lm(total_steps ~ weight_kg, data = .))

print(summary(lm_by_window$model[[1]]), digits = 4)
```

```
##
## Call:
## lm(formula = total_steps ~ weight_kg, data = .)
```

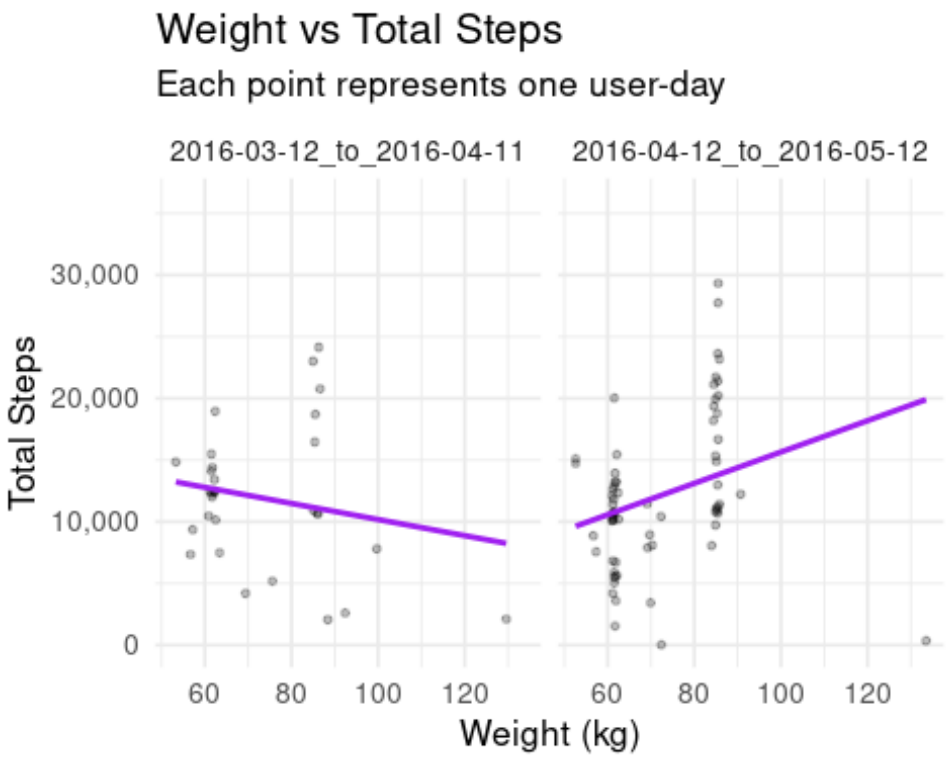
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8873.6 -3048.6  -283.8  1669.6 13062.4
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16703.41    4617.50   3.617  0.00112 **
## weight_kg     -65.24      61.37  -1.063  0.29659
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5649 on 29 degrees of freedom
##   (402 observations deleted due to missingness)
## Multiple R-squared:  0.0375, Adjusted R-squared:  0.004308
## F-statistic:  1.13 on 1 and 29 DF,  p-value: 0.2966
```

```
print(summary(lm_by_window$model[[2]]), digits = 4)
```

```
##
## Call:
## lm(formula = total_steps ~ weight_kg, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19549.7  -3449.8   -406.4   3000.3  15514.4
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2956.46    3827.53   0.772   0.4427
## weight_kg     126.96      52.18   2.433   0.0177 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5902 on 65 degrees of freedom
##   (873 observations deleted due to missingness)
## Multiple R-squared:  0.08347,    Adjusted R-squared:  0.06937
## F-statistic:  5.92 on 1 and 65 DF,  p-value: 0.01773
```

```
ggplot(daily, aes(x = weight_kg, y = total_steps)) +
  geom_point(alpha = 0.25, size = 1) +
  geom_smooth(method = "lm", se = FALSE, color = "purple", linewidth =
1) +
  facet_wrap(~ dataset_window) +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  labs(
    title = "Weight vs Total Steps",
    subtitle = "Each point represents one user-day",
    x = "Weight (kg)",
    y = "Total Steps"
```

```
  ) +
theme_minimal(base_size = 13)
```



Interpretation notes: Weight-related analyses are based on a small subset of days with weight logs (31 days in the first window; 67 days in the second), so results should be interpreted as descriptive rather than representative of the full population.

Across both windows, weight shows a positive association with calories burned, stronger in the second window. Correlations are 0.43 (Window 1) and 0.66 (Window 2). The linear models are consistent with this: weight explains ~19% of calorie variation in the first window ($R^2 \approx 0.19$) and ~44% in the second window ($R^2 \approx 0.44$), with a positive slope in both cases. This likely reflects that higher body mass tends to increase energy expenditure estimates for comparable activity levels.

The relationship between weight and steps is weaker and less stable. In the first window, the correlation is slightly negative (−0.19) and the regression is not statistically meaningful (very low $R^2 \approx 0.04$; $p \approx 0.30$). In the second window, the association turns modestly positive (correlation 0.29) with a small but statistically detectable effect ($R^2 \approx 0.08$). Overall, weight is not a strong predictor of step volume in this sample.

Taken together, these results suggest that weight is more closely tied to estimated calorie expenditure than to daily step counts, but the sparsity and

selectivity of weight logging means conclusions should remain cautious and framed as patterns within a highly engaged subset of users.

Engagement and Retention Trends

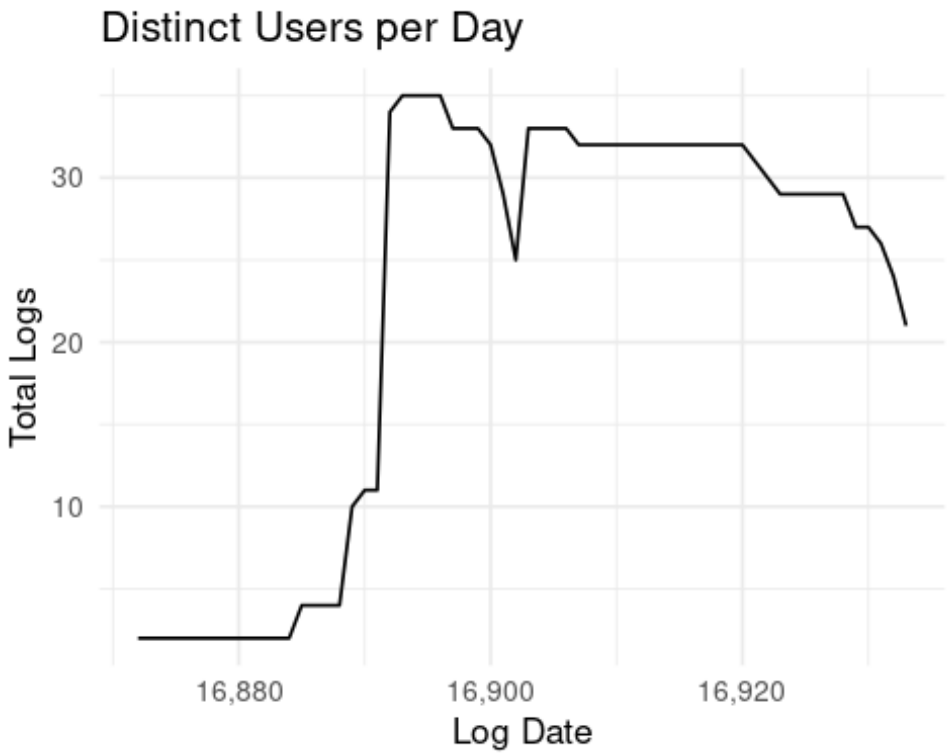
Daily Logging Participation

This section focuses on how many users log data on a given day, providing insight into engagement and retention over time.

```
unique_users <- daily %>%
  group_by(log_date) %>%
  summarize(distinct_users = n_distinct(Id), .groups = "drop")

unique_users
```

```
## # A tibble: 62 × 2
##    log_date   distinct_users
##    <date>              <int>
##  1 2016-03-12              2
##  2 2016-03-13              2
##  3 2016-03-14              2
##  4 2016-03-15              2
##  5 2016-03-16              2
##  6 2016-03-17              2
##  7 2016-03-18              2
##  8 2016-03-19              2
##  9 2016-03-20              2
## 10 2016-03-21              2
## #    52 more rows
```

```
ggplot(unique_users, aes(x = log_date, y = distinct_users)) +
  geom_line() +
  scale_x_continuous(labels = scales::comma) +
  labs(
    title = "Distinct Users per Day",
    x = "Log Date",
    y = "Total Logs"
  ) +
  theme_minimal(base_size = 13)
```



Interpretation notes: User participation increases sharply early in the observation period and stabilizes before gradually declining toward the end. This pattern is consistent with typical engagement decay seen in consumer health tracking applications.

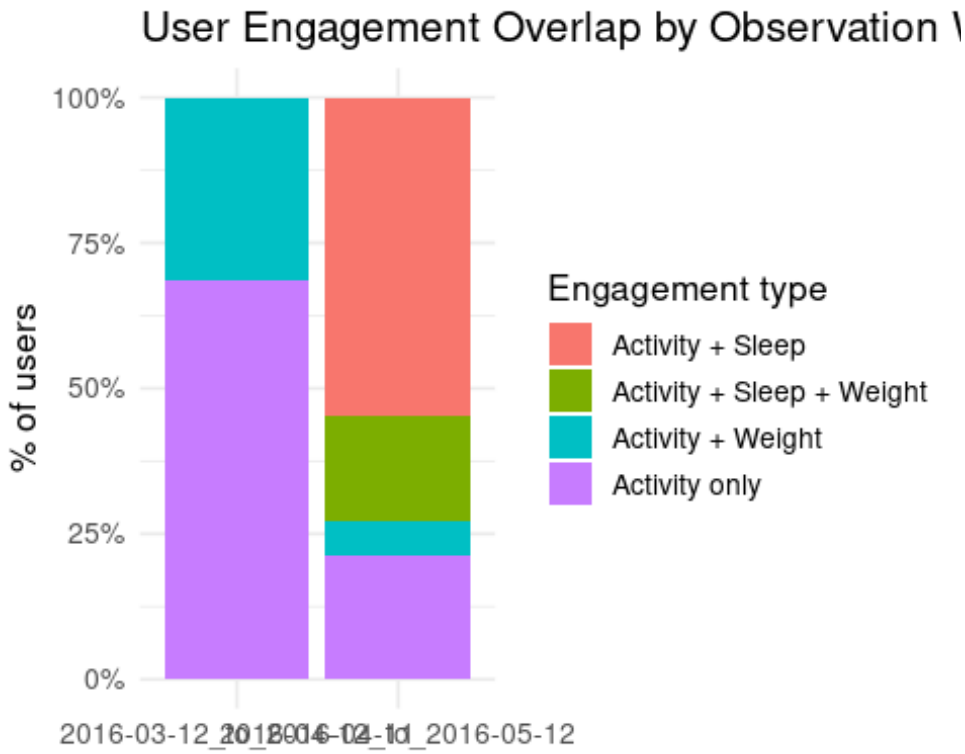
Cross-Feature Engagement

```
user_overlap <- users %>%
  mutate(
    engagement_group = case_when(
      has_sleep_log & has_weight_log ~ "Activity + Sleep + Weight",
      has_sleep_log & !has_weight_log ~ "Activity + Sleep",
      !has_sleep_log & has_weight_log ~ "Activity + Weight",
      TRUE ~ "Activity only"
    )
  )

overlap_summary <- user_overlap %>%
  group_by(dataset_window, engagement_group) %>%
  summarise(
    n_users = n(),
    .groups = "drop"
  ) %>%
  group_by(dataset_window) %>%
  mutate(
    pct_users = n_users / sum(n_users)
  )
```

```
overlap_summary
```

```
## # A tibble: 6 × 4
## # Groups:   dataset_window [2]
##   dataset_window           engagement_group          n_users
pct_users
##   <chr>                    <chr>                       <int>
<dbl>
## 1 2016-03-12_to_2016-04-11 Activity + Weight              11
0.314
## 2 2016-03-12_to_2016-04-11 Activity only                  24
0.686
## 3 2016-04-12_to_2016-05-12 Activity + Sleep               18
0.545
## 4 2016-04-12_to_2016-05-12 Activity + Sleep + Weight       6
0.182
## 5 2016-04-12_to_2016-05-12 Activity + Weight               2
0.0606
## 6 2016-04-12_to_2016-05-12 Activity only                   7
0.212
```

```
ggplot(overlap_summary,
       aes(x = dataset_window, y = pct_users, fill =
engagement_group)) +
  geom_col() +
  scale_y_continuous(labels = scales::percent) +
  labs(
    title = "User Engagement Overlap by Observation Window",
    x = NULL,
    y = "% of users",
    fill = "Engagement type"
  ) +
  theme_minimal(base_size = 13)
```



Interpretation notes: User engagement differs substantially across the two observation windows, both in feature adoption and depth of tracking.

In the first observation window (2016-03-12 to 2016-04-11), engagement is limited almost entirely to activity tracking. Approximately 69% of users log only activity, while the remaining 31% log activity in combination with weight. Sleep tracking is entirely absent during this period, indicating either limited feature usage or incomplete data availability. Weight logging, while present, is restricted to a minority of users and does not represent a core behavior.

In the second observation window (2016-04-12 to 2016-05-12), engagement becomes more diversified. A majority of users (~55%) log both activity and sleep, and an additional 18% log activity, sleep, and weight together. Activity-only users decline to just over 21%, while activity-plus-weight users remain rare (~6%). This shift highlights a clear increase in sleep feature adoption, while weight tracking continues to be used by a small, highly engaged subset.

Overall, activity tracking serves as the foundational behavior across all users, sleep tracking emerges as a meaningful secondary feature for engaged users, and weight tracking remains niche and selective.

Tracker Distance vs Total Distance

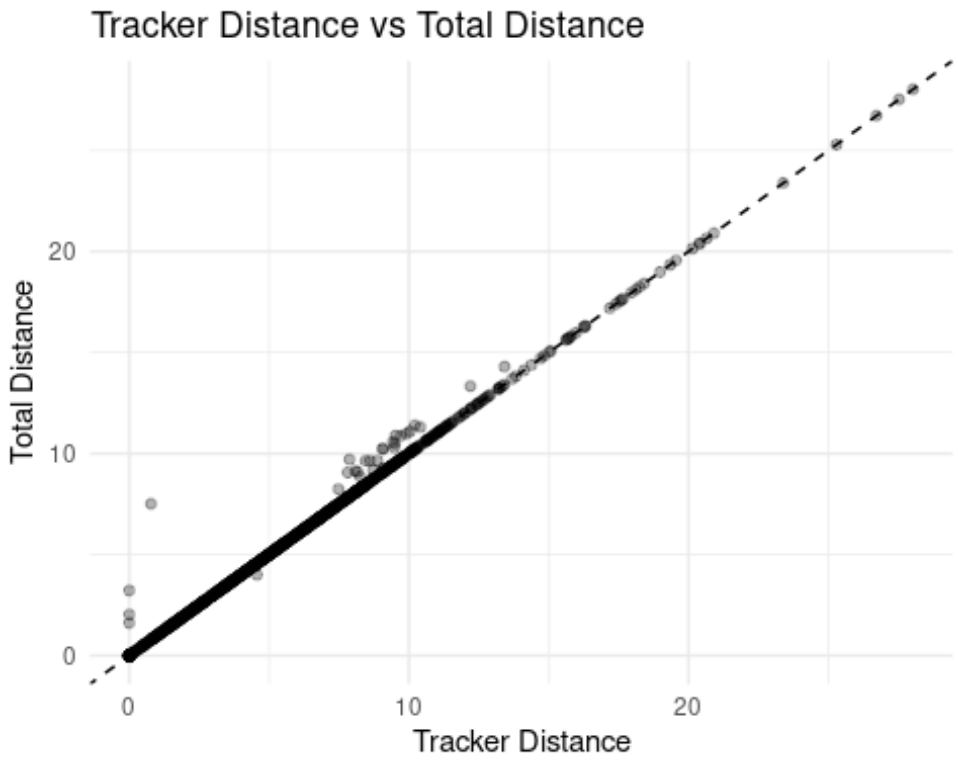
```
cor_dist <- daily %>%
  summarise(
    cor_total_tracker = cor(total_distance, tracker_distance, use =
"complete.obs")
  )

cor_dist
```

```
## # A tibble: 1 × 1
##   cor_total_tracker
##               <dbl>
## 1             0.998
```

```
daily_leaf <- daily %>%
  mutate(
    distance_diff = total_distance - tracker_distance
  )

summary(daily_leaf$distance_diff)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.5700  0.0000  0.0000  0.0271  0.0000  6.7300
```

```
ggplot(daily,
       aes(x = tracker_distance, y = total_distance)) +
  geom_point(alpha = 0.3) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
  labs(
    title = "Tracker Distance vs Total Distance",
    x = "Tracker Distance",
    y = "Total Distance"
  ) +
  theme_minimal()
```



Interpretation notes: Tracker distance and total distance are nearly identical across the dataset. The correlation between the two measures is extremely high (r ≈ 0.998), indicating near-perfect alignment.

The distribution of differences further confirms this finding: for the vast majority of records, the difference between total distance and tracker distance is exactly zero, with a median of 0.0 and minimal variation. Occasional positive deviations exist, but they are infrequent and small relative to overall distance values.

These results suggest that total distance is overwhelmingly derived from tracker-recorded movement, with negligible contribution from non-tracker sources. As a result, tracker distance alone is sufficient for distance-based analyses without material loss of information.

Overall Conclusions and Recommendations

This analysis examined user activity, sleep, and logging behavior across two time periods using both SQL-based data marts, Tableau dashboards, and statistical summaries in R.

Key Conclusions

1. Activity tracking is the core and universal behavior: All users engage with activity tracking, making it the primary and most reliable signal of

engagement. Other features build on top of this foundation rather than replacing it.

2. Sleep tracking shows strong adoption among engaged users: Once available, sleep tracking is adopted by a majority of users and is used consistently by those who adopt it. Sleep logging appears to be a habitual behavior rather than a sporadic one.

3. Weight tracking is a niche, high-intent feature: Weight logging is rare, highly concentrated, and used primarily by a small subset of users. It should not be treated as a baseline behavior or population-wide metric.

4. Calories are driven by activity intensity, not just volume: Higher-intensity activity (very active minutes) explains substantially more variation in calories than steps or light activity alone. Sedentary time and sleep duration show weak or negligible relationships with calorie expenditure.

5. Tracker data is highly reliable and internally consistent: The near-perfect alignment between tracker distance and total distance supports confidence in the device's core movement measurements.

6. Weekday vs. weekend patterns are subtle, with gradual declines beginning midweek and slightly lower activity on weekends rather than abrupt changes.

7. User behavior varies more by engagement segment than by calendar timing, highlighting the importance of user-level segmentation.

Business Recommendations for Bellabeat Leaf

1. Anchor the product experience around activity-first engagement

Treat activity tracking as the default entry point for all users.

Design onboarding and dashboards to emphasize daily movement trends before introducing secondary features.

2. Position sleep tracking as the primary engagement deepener

Promote sleep tracking once users demonstrate consistent activity logging.

Use nudges that connect sleep quality to activity consistency, rather than to calorie outcomes, since sleep duration alone does not meaningfully predict calories.

3. Reframe weight tracking as an optional, goal-driven tool

Avoid positioning weight logging as a daily expectation.

Instead, market it toward users with specific goals (e.g., weight management or long-term progress tracking), acknowledging its intentional and infrequent use.

4. Emphasize activity intensity over step count alone

Highlight very active minutes and intensity-based insights in the Leaf app.

Introduce prompts or feedback that encourage short bursts of higher-intensity movement, which have a clearer relationship with calorie expenditure than additional low-intensity steps.

5. Simplify distance-based reporting using tracker distance

Use tracker distance as the primary distance metric in dashboards and analyses.

This reduces complexity while maintaining accuracy, improving clarity for both users and internal analytics.

6. Segment Users by Engagement Patterns

User behavior varies significantly across engagement groups (low engagement, irregular movers, consistent movers). Treating all users the same obscures meaningful differences.

Adopt engagement-based segmentation in analytics and product design:

- Low engagement: simplify goals, reduce friction
- Irregular movers: emphasize consistency streaks
- Consistent movers: introduce advanced insights (sleep efficiency, intensity optimization)