# Unlocking Growth: A Smart Device Usage Analysis for Bellabeat

badhusha sathar

2025-04-26

# Bellabeat Case Study – Ask Phase

# 1. Business Task Statement

Bellabeat is a wellness technology company specializing in smart devices designed for women’s health. The company aims to expand its presence in the global smart device market by analyzing **smart device usage trends**. By understanding consumer behavior, Bellabeat can refine its **marketing strategy** and align its products with industry trends.

## Objectives

- **Analyze smart device fitness data** to identify key trends.
- **Apply insights** to one Bellabeat product (e.g., Leaf wellness tracker, Time watch, or Spring hydration bottle).
- **Develop high-level marketing recommendations** based on findings.
- **Present analysis** to Bellabeat’s executive team for strategic decision-making.

# 2. Key Stakeholders

- **Urška Sršen** – Cofounder & Chief Creative Officer, responsible for product development & branding.
- **Sando Mur** – Cofounder & key executive team member, providing analytical expertise.
- **Bellabeat Marketing Analytics Team** – Responsible for data insights shaping marketing strategies.
- **Bellabeat Customers** – Women using Bellabeat products, influencing demand and engagement.

# 3. Expected Deliverables

- **A clear statement of the business task** to guide analysis.
- **Identification of key stakeholders** who will use the findings for decision-making.
- **Strategic alignment** of data-driven insights with Bellabeat’s business goals.

# 4. Next Steps

1. **Prepare**: Identify and assess relevant datasets.
2. **Process**: Clean, filter, and analyze data for meaningful insights.
3. **Analyze**: Identify trends and patterns affecting smart device usage.
4. **Share**: Communicate findings using effective visualizations.
5. **Act**: Develop recommendations to enhance Bellabeat’s marketing approach.

# Prepare Phase

# Business Task

Bellabeat aims to analyze smart device usage data to gain insights into consumer habits. These insights will be used to refine marketing strategies and enhance engagement with potential customers.

# Data Sources Used

- **FitBit Fitness Tracker Data** (Public Domain, Kaggle): Contains minute-level physical activity, heart rate, and sleep monitoring data from 30 users.

# Data Organization & Storage

- Data is stored in CSV format and organized in **wide format** for analysis.
- Data integrity and credibility are verified using **ROCCC** (Reliable, Original, Comprehensive, Current, Cited).
- Privacy concerns are reviewed, ensuring compliance with licensing and security guidelines.

# Data Fetching

```
# Load necessary libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
# importing data
dailyActivity_merged <- read.csv("C:/Users/badhu/Documents/project/product/Fitabase Data 4.1
2.16-5.12.16/dailyActivity_merged.csv")
hourlyCalories_merged<-read.csv("C:/Users/badhu/Documents/project/product/Fitabase Data 4.12.
16-5.12.16/hourlyCalories_merged.csv")
hourlyIntensities_merged<-read.csv("C:/Users/badhu/Documents/project/product/Fitabase Data 4.
12.16-5.12.16/hourlyIntensities_merged.csv")
hourlySteps_merged<- read.csv("C:/Users/badhu/Documents/project/product/Fitabase Data 4.12.16
-5.12.16/hourlySteps_merged.csv")
sleepDay_merged<-read.csv("C:/Users/badhu/Documents/project/product/Fitabase Data 4.12.16-5.1
2.16/sleepDay_merged.csv")
minuteMETsNarrow_merged<-read.csv("C:/Users/badhu/Documents/project/product/Fitabase Data 4.1
2.16-5.12.16/minuteMETsNarrow_merged.csv")

# Count the number of rows
total_rows <- nrow(dailyActivity_merged)

# Print result
print(total_rows)
```

```
## [1] 940
```

# Process Phase

This report outlines the data processing phase for analyzing smart device usage trends to support Bellabeat's marketing strategy. The primary goal is to clean and transform the dataset for meaningful analysis.

# Data Processing Steps

```
# Identify duplicate entries
duplicates <- dailyActivity_merged %>%
  group_by(Id, ActivityDate, TotalSteps) %>%
  summarise(Count = n(), .groups = "drop") %>%
  filter(Count > 1)


# Display the duplicate records
print(duplicates)
```

```
## # A tibble: 0 × 4
## # ℹ 4 variables: Id <dbl>, ActivityDate <chr>, TotalSteps <int>, Count <int>
```

```
# Load necessary library
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
## The following objects are masked from 'package:base':
## 
##     date, intersect, setdiff, union
```

```
# Verify the result
head(sleepDay_merged$SleepDay)
```

```
## [1] "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 12:00:00 AM"
## [4] "4/16/2016 12:00:00 AM" "4/17/2016 12:00:00 AM" "4/19/2016 12:00:00 AM"
```

```
# Load necessary library
library(stringr)

# Remove time portion and convert to date
sleepDay_merged$SleepDay <- str_extract(sleepDay_merged$SleepDay, "^\\d+/\\d+/\\d+")  # Extra
ct only the MM/DD/YYYY part



# Check the result
str(sleepDay_merged$SleepDay)
```

```
##  chr [1:413] "4/12/2016" "4/13/2016" "4/15/2016" "4/16/2016" "4/17/2016" ...
```

```
# Convert SleepDay to Date format (similar to SQL's Convert(date, SleepDay, 101))
sleepDay_merged$SleepDay <- mdy(sleepDay_merged$SleepDay)

# Display updated data structure
str(sleepDay_merged)
```

```
## 'data.frame':    413 obs. of  5 variables:
##  $ Id                : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : Date, format: "2016-04-12" "2016-04-13" ...
##  $ TotalSleepRecords : int  1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: int  327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : int  346 407 442 367 712 320 377 364 384 449 ...
```

```
# Convert ActivityDate from character to Date format
dailyActivity_merged$ActivityDate <- mdy(dailyActivity_merged$ActivityDate)
# Add a new column for the day of the week
dailyActivity_merged <- dailyActivity_merged %>%
  mutate(day_of_week = weekdays(as.Date(ActivityDate, format="%Y-%m-%d")))
# View the updated dataframe
print(head(dailyActivity_merged))
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366   2016-04-12      13162          8.50            8.50
## 2 1503960366   2016-04-13      10735          6.97            6.97
## 3 1503960366   2016-04-14      10460          6.74            6.74
## 4 1503960366   2016-04-15       9762          6.28            6.28
## 5 1503960366   2016-04-16      12669          8.16            8.16
## 6 1503960366   2016-04-17       9705          6.48            6.48
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               1.88                     0.55
## 2                        0               1.57                     0.69
## 3                        0               2.44                     0.40
## 4                        0               2.14                     1.26
## 5                        0               2.71                     0.41
## 6                        0               3.19                     0.78
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                4.71                       0                21
## 3                3.91                       0                30
## 4                2.83                       0                29
## 5                5.04                       0                36
## 6                2.51                       0                38
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  13                  328              728     1985
## 2                  19                  217              776     1797
## 3                  11                  181             1218     1776
## 4                  34                  209              726     1745
## 5                  10                  221              773     1863
## 6                  20                  164              539     1728
##   day_of_week
## 1     Tuesday
## 2   Wednesday
## 3    Thursday
## 4      Friday
## 5    Saturday
## 6      Sunday
```

```
# sleepDay_merged table merged with dailyActivity_merged
dailyActivity_merged <- dailyActivity_merged %>%
full_join(sleepDay_merged, by = c("Id" = "Id", "ActivityDate" = "SleepDay"))



# Verify the change
str(dailyActivity_merged)
```

```
## 'data.frame':    943 obs. of  19 variables:
##  $ Id                      : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDate            : Date, format: "2016-04-12" "2016-04-13" ...
##  $ TotalSteps              : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819
...
##  $ TotalDistance           : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ TrackerDistance         : num  8.5 6.97 6.74 6.28 8.16 ...
##  $ LoggedActivitiesDistance: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance      : num  1.88 1.57 2.44 2.14 2.71 ...
##  $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##  $ LightActiveDistance     : num  6.06 4.71 3.91 2.83 5.04 ...
##  $ SedentaryActiveDistance : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes       : int  25 21 30 29 36 38 42 50 28 19 ...
##  $ FairlyActiveMinutes     : int  13 19 11 34 10 20 16 31 12 8 ...
##  $ LightlyActiveMinutes    : int  328 217 181 209 221 164 233 264 205 211 ...
##  $ SedentaryMinutes        : int  728 776 1218 726 773 539 1149 775 818 838 ...
##  $ Calories                : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
##  $ day_of_week             : chr  "Tuesday" "Wednesday" "Thursday" "Friday" ...
##  $ TotalSleepRecords       : int  1 2 NA 1 2 1 NA 1 1 1 ...
##  $ TotalMinutesAsleep      : int  327 384 NA 412 340 700 NA 304 360 325 ...
##  $ TotalTimeInBed          : int  346 407 NA 442 367 712 NA 320 377 364 ...
```

```
summary(dailyActivity_merged$day_of_week)
```

```
##    Length     Class      Mode 
##       943 character character
```

```
head(dailyActivity_merged)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366   2016-04-12      13162          8.50            8.50
## 2 1503960366   2016-04-13      10735          6.97            6.97
## 3 1503960366   2016-04-14      10460          6.74            6.74
## 4 1503960366   2016-04-15       9762          6.28            6.28
## 5 1503960366   2016-04-16      12669          8.16            8.16
## 6 1503960366   2016-04-17       9705          6.48            6.48
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               1.88                     0.55
## 2                        0               1.57                     0.69
## 3                        0               2.44                     0.40
## 4                        0               2.14                     1.26
## 5                        0               2.71                     0.41
## 6                        0               3.19                     0.78
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                6.06                       0                25
## 2                4.71                       0                21
## 3                3.91                       0                30
## 4                2.83                       0                29
## 5                5.04                       0                36
## 6                2.51                       0                38
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  13                  328              728     1985
## 2                  19                  217              776     1797
## 3                  11                  181             1218     1776
## 4                  34                  209              726     1745
## 5                  10                  221              773     1863
## 6                  20                  164              539     1728
##   day_of_week TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
## 1     Tuesday                 1                327            346
## 2   Wednesday                 2                384            407
## 3    Thursday                NA                 NA             NA
## 4      Friday                 1                412            442
## 5    Saturday                 2                340            367
## 6      Sunday                 1                700            712
```

```
# Convert Date column to proper datetime format
hourlyCalories_merged$ActivityHour <- mdy_hms(hourlyCalories_merged$ActivityHour)

# Extract hour from Date(ActivityHour) and store in a new column 'time_h'
hourlyCalories_merged <- hourlyCalories_merged %>%
  mutate(time_h = hour(ActivityHour))

# Extract only the date part
hourlyCalories_merged <- hourlyCalories_merged %>%
  mutate(ActivityHour = as.Date(ActivityHour))

# Verify changes
str(hourlyCalories_merged)
```

```
## 'data.frame':    22099 obs. of  4 variables:
##  $ Id          : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour: Date, format: "2016-04-12" "2016-04-12" ...
##  $ Calories    : int  81 61 59 47 48 48 48 47 68 141 ...
##  $ time_h      : int  0 1 2 3 4 5 6 7 8 9 ...
```

```
head(hourlyCalories_merged)
```

```
##           Id ActivityHour Calories time_h
## 1 1503960366   2016-04-12       81      0
## 2 1503960366   2016-04-12       61      1
## 3 1503960366   2016-04-12       59      2
## 4 1503960366   2016-04-12       47      3
## 5 1503960366   2016-04-12       48      4
## 6 1503960366   2016-04-12       48      5
```

```
# Convert ActivityHour to proper datetime format
hourlyIntensities_merged <- hourlyIntensities_merged %>%
  mutate(ActivityHour = mdy_hms(ActivityHour))  # Convert to Date-Time format

# Extract hour and store it in a new column `time_h`
hourlyIntensities_merged <- hourlyIntensities_merged %>%
  mutate(time_h = hour(ActivityHour))  # Extract hour

# Extract only the date portion from `ActivityHour`
hourlyIntensities_merged <- hourlyIntensities_merged %>%
  mutate(ActivityHour = as.Date(ActivityHour))  # Keep only the date

# Verify changes
str(hourlyIntensities_merged)
```

```
## 'data.frame':    22099 obs. of  5 variables:
##  $ Id              : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour    : Date, format: "2016-04-12" "2016-04-12" ...
##  $ TotalIntensity  : int  20 8 7 0 0 0 0 0 13 30 ...
##  $ AverageIntensity: num  0.333 0.133 0.117 0 0 ...
##  $ time_h          : int  0 1 2 3 4 5 6 7 8 9 ...
```

```
head(hourlyIntensities_merged)
```

```
##           Id ActivityHour TotalIntensity AverageIntensity time_h
## 1 1503960366   2016-04-12             20         0.333333      0
## 2 1503960366   2016-04-12              8         0.133333      1
## 3 1503960366   2016-04-12              7         0.116667      2
## 4 1503960366   2016-04-12              0         0.000000      3
## 5 1503960366   2016-04-12              0         0.000000      4
## 6 1503960366   2016-04-12              0         0.000000      5
```

```
# Convert ActivityHour
hourlySteps_merged <- hourlySteps_merged %>%
  mutate(ActivityHour = mdy_hms(ActivityHour))  # Convert to Date-Time format

# Extract hour and store it in a new column `time_h`
hourlySteps_merged <- hourlySteps_merged %>%
  mutate(time_h = hour(ActivityHour))  # Extract hour

# Extract only the date portion from `ActivityHour`
hourlySteps_merged <- hourlySteps_merged %>%
  mutate(ActivityHour = as.Date(ActivityHour))  # Keep only the date

# Verify the changes
str(hourlySteps_merged)
```

```
## 'data.frame':    22099 obs. of  4 variables:
##  $ Id          : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour: Date, format: "2016-04-12" "2016-04-12" ...
##  $ StepTotal   : int  373 160 151 0 0 0 0 0 250 1864 ...
##  $ time_h      : int  0 1 2 3 4 5 6 7 8 9 ...
```

```
head(hourlySteps_merged)
```

```
##           Id ActivityHour StepTotal time_h
## 1 1503960366   2016-04-12       373      0
## 2 1503960366   2016-04-12       160      1
## 3 1503960366   2016-04-12       151      2
## 4 1503960366   2016-04-12         0      3
## 5 1503960366   2016-04-12         0      4
## 6 1503960366   2016-04-12         0      5
```

```
# Convert ActivityMinute to Date-Time format
minuteMETsNarrow_merged <- minuteMETsNarrow_merged %>%
  mutate(ActivityMinute = mdy_hms(ActivityMinute))  # Convert to Date-Time format

# Extract time (hour, minute, second) and store in a new column `time_t`
minuteMETsNarrow_merged <- minuteMETsNarrow_merged %>%
  mutate(time_t = format(ActivityMinute, "%H:%M:%S"))  # Extract time as character format

# Extract only the date portion from `ActivityMinute`
minuteMETsNarrow_merged <- minuteMETsNarrow_merged %>%
  mutate(ActivityMinute = as.Date(ActivityMinute))  # Keep only the date

# Verify changes
str(minuteMETsNarrow_merged)
```

```
## 'data.frame':    1325580 obs. of  4 variables:
##  $ Id            : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityMinute: Date, format: "2016-04-12" "2016-04-12" ...
##  $ METs          : int  10 10 10 10 10 12 12 12 12 12 ...
##  $ time_t        : chr  "00:00:00" "00:01:00" "00:02:00" "00:03:00" ...
```

```
head(minuteMETsNarrow_merged)
```

```
##           Id ActivityMinute METs   time_t
## 1 1503960366     2016-04-12   10 00:00:00
## 2 1503960366     2016-04-12   10 00:01:00
## 3 1503960366     2016-04-12   10 00:02:00
## 4 1503960366     2016-04-12   10 00:03:00
## 5 1503960366     2016-04-12   10 00:04:00
## 6 1503960366     2016-04-12   12 00:05:00
```

```
# Merge tables using inner join
hourly_cal_int_step_merge <- hourlyCalories_merged %>%
  inner_join(hourlyIntensities_merged, by = c("Id", "ActivityHour", "time_h")) %>%
  inner_join(hourlySteps_merged, by = c("Id", "ActivityHour", "time_h")) %>%
  select(Id, ActivityHour, time_h, Calories, TotalIntensity, AverageIntensity, StepTotal)

# View structure of merged table
str(hourly_cal_int_step_merge)
```

```
## 'data.frame':    22099 obs. of  7 variables:
##  $ Id              : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityHour    : Date, format: "2016-04-12" "2016-04-12" ...
##  $ time_h          : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Calories        : int  81 61 59 47 48 48 48 47 68 141 ...
##  $ TotalIntensity  : int  20 8 7 0 0 0 0 0 13 30 ...
##  $ AverageIntensity: num  0.333 0.133 0.117 0 0 ...
##  $ StepTotal       : int  373 160 151 0 0 0 0 0 250 1864 ...
```

```
# Save merged table as CSV
write.csv(hourly_cal_int_step_merge, "hourly_cal_int_step_merge.csv", row.names = FALSE)
```

```
# Select required columns
minuteMETsNarrow_selected <- minuteMETsNarrow_merged %>%
  select(Id, ActivityMinute, METs, time_t)

# View the transformed dataset
head(minuteMETsNarrow_selected)
```

```
##           Id ActivityMinute METs   time_t
## 1 1503960366     2016-04-12   10 00:00:00
## 2 1503960366     2016-04-12   10 00:01:00
## 3 1503960366     2016-04-12   10 00:02:00
## 4 1503960366     2016-04-12   10 00:03:00
## 5 1503960366     2016-04-12   10 00:04:00
## 6 1503960366     2016-04-12   12 00:05:00
```

```
#Save the cleaned dataset
#write.csv(minuteMETsNarrow_selected, "minuteMETsNarrow_cleaned.csv", row.names #= FALSE)
```

# Analysis

Bellabeat, a wellness technology company, seeks to leverage smart device usage trends to enhance its marketing strategies. This analysis aims to uncover patterns in smart device usage and their implications for Bellabeat’s business decisions.

```
# Convert date columns to Date type for proper merging
minuteMETsNarrow_selected$ActivityMinute <- as.Date(minuteMETsNarrow_selected$ActivityMinute,
format="%Y-%m-%d")
dailyActivity_merged$ActivityDate <- as.Date(dailyActivity_merged$ActivityDate, format="%Y-%m
-%d")



# Aggregate METs per user per day
METs_summary <- minuteMETsNarrow_selected %>%
  group_by(Id, ActivityMinute) %>%
  summarise(sum_mets = sum(METs, na.rm = TRUE), .groups = "keep")  # Keeps grouping Info
names(METs_summary)
```

```
## [1] "Id"             "ActivityMinute" "sum_mets"
```

```
names(dailyActivity_merged)
```

```
##  [1] "Id"                       "ActivityDate"
##  [3] "TotalSteps"               "TotalDistance"
##  [5] "TrackerDistance"          "LoggedActivitiesDistance"
##  [7] "VeryActiveDistance"       "ModeratelyActiveDistance"
##  [9] "LightActiveDistance"      "SedentaryActiveDistance"
## [11] "VeryActiveMinutes"        "FairlyActiveMinutes"
## [13] "LightlyActiveMinutes"     "SedentaryMinutes"
## [15] "Calories"                 "day_of_week"
## [17] "TotalSleepRecords"        "TotalMinutesAsleep"
## [19] "TotalTimeInBed"
```

```
# Merge with daily activity data to bring in calories burned
final_data <- METs_summary %>%
  inner_join(dailyActivity_merged, by = c("Id" = "Id", "ActivityMinute" = "ActivityDate")) %
>%
  select(Id, ActivityMinute, sum_mets, Calories) %>%
  arrange(ActivityMinute)
head(final_data)
```

```
## # A tibble: 6 × 4
## # Groups:   Id, ActivityMinute [6]
##           Id ActivityMinute sum_mets Calories
##        <dbl> <date>            <int>    <int>
## 1 1503960366 2016-04-12        25241     1985
## 2 1624580081 2016-04-12        17234     1432
## 3 1644430081 2016-04-12        22768     3199
## 4 1844505072 2016-04-12        21704     2030
## 5 1927972279 2016-04-12        15599     2220
## 6 2022484408 2016-04-12        23035     2390
```

```
# Summarizing activity metrics per user
activities_summary <- dailyActivity_merged %>%
  group_by(Id) %>%
  summarise(
    total_steps = sum(TotalSteps, na.rm = TRUE),
    total_Vactive_mins = sum(VeryActiveMinutes, na.rm = TRUE),
    total_Factive_mins = sum(FairlyActiveMinutes, na.rm = TRUE),
    total_Lactive_mins = sum(LightlyActiveMinutes, na.rm = TRUE),
    total_calories = sum(Calories, na.rm = TRUE)
  )

# View the result
print(activities_summary)
```

```
## # A tibble: 33 × 6
##          Id total_steps total_Vactive_mins total_Factive_mins total_Lactive_mins
##       <dbl>       <int>              <int>              <int>              <int>
##  1   1.50e9      375619               1200                594               6818
##  2   1.62e9      178061                269                180               4758
##  3   1.64e9      218489                287                641               5354
##  4   1.84e9       79982                  4                 40               3579
##  5   1.93e9       28400                 41                 24               1196
##  6   2.02e9      352490               1125                600               7981
##  7   2.03e9      172573                  3                  8               7956
##  8   2.32e9      146223                 42                 80               6144
##  9   2.35e9      171354                243                370               4545
## 10   2.87e9      234229                437                190               9548
## # ℹ 23 more rows
## # ℹ 1 more variable: total_calories <int>
```

Result: Strong correlation between activity intense time and calories burned

```
# Calculate average sleep time per user
sleep_summary <- sleepDay_merged %>%
  group_by(Id) %>%
  summarise(
    avg_sleep_time_h = mean(TotalMinutesAsleep, na.rm = TRUE) / 60,
    avg_time_bed_h = mean(TotalTimeInBed, na.rm = TRUE) / 60,
    wasted_bed_time_m = mean(TotalTimeInBed - TotalMinutesAsleep, na.rm = TRUE)
  )

# View the result
print(sleep_summary)
```

```
## # A tibble: 24 × 4
##            Id avg_sleep_time_h avg_time_bed_h wasted_bed_time_m
##         <dbl>            <dbl>          <dbl>             <dbl>
##  1 1503960366             6.00           6.39              22.9
##  2 1644430081             4.9            5.77              52  
##  3 1844505072            10.9           16.0              309  
##  4 1927972279             6.95           7.30              20.8
##  5 2026352035             8.44           8.96              31.5
##  6 2320127002             1.02           1.15               8  
##  7 2347167796             7.45           8.19              44.5
##  8 3977333714             4.89           7.69             168. 
##  9 4020332650             5.82           6.33              30.4
## 10 4319703577             7.94           8.37              25.3
## # ℹ 14 more rows
```

```
#Sleep and calories comparison
# Perform the join and aggregation
sleep_calories_summary <- dailyActivity_merged %>%
  inner_join(sleepDay_merged, by = c("Id" = "Id", "ActivityDate" = "SleepDay"))
```

```
## Warning in inner_join(., sleepDay_merged, by = c(Id = "Id", ActivityDate = "SleepDay")): D
etected an unexpected many-to-many relationship between `x` and `y`.
## ℹ Row 436 of `x` matches multiple rows in `y`.
## ℹ Row 161 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
# Remove duplicate columns (.x or .y suffix)

  sleep_calories_summary <- sleep_calories_summary %>%
  select(-c(TotalSleepRecords.y, TotalMinutesAsleep.y,
 TotalTimeInBed.y))
sleep_calories_summary <-sleep_calories_summary %>%
  group_by(Id) %>%
  summarise(
    total_sleep_m = sum(TotalMinutesAsleep.x, na.rm = TRUE),
    total_time_inbed_m = sum(TotalTimeInBed.x, na.rm = TRUE),
    calories = sum(Calories, na.rm = TRUE)
  )

# View the result
print(sleep_calories_summary)
```

```
## # A tibble: 24 × 4
##            Id total_sleep_m total_time_inbed_m calories
##         <dbl>         <int>              <int>    <int>
##  1 1503960366          9007               9580    46807
##  2 1644430081          1176               1384    11911
##  3 1844505072          1956               2883     5029
##  4 1927972279          2085               2189    11581
##  5 2026352035         14173              15054    43142
##  6 2320127002            61                 69     1804
##  7 2347167796          6702               7370    29570
##  8 3977333714          8222              12912    43691
##  9 4020332650          2795               3038    25560
## 10 4319703577         12393              13051    52642
## # ℹ 14 more rows
```

```
# Summarizing daily activity metrics by day of the week
daily_sum_analysis <- dailyActivity_merged %>%
  group_by(day_of_week) %>%
  summarise(
    total_steps = sum(TotalSteps, na.rm = TRUE),
    total_dist = sum(TotalDistance, na.rm = TRUE),
    total_calories = sum(Calories, na.rm = TRUE)
  )


# View the result
print(daily_sum_analysis)
```

```
## # A tibble: 7 × 4
##   day_of_week total_steps total_dist total_calories
##   <chr>             <int>      <dbl>          <int>
## 1 Friday           938477       669.         293805
## 2 Monday           946109       676.         282910
## 3 Saturday        1025339       738.         295699
## 4 Sunday           838921       608.         273823
## 5 Thursday        1098261       788.         326236
## 6 Tuesday         1235001       886.         358114
## 7 Wednesday       1133906       823.         345393
```

Result:Daily Sum Analysis - No trends/patterns found

```
# Summarizing daily activity metrics by day of the week
daily_avg_analysis <- dailyActivity_merged %>%
  group_by(day_of_week) %>%
  summarise(
    avg_steps = mean(TotalSteps, na.rm = TRUE),
    avg_dist = mean(TotalDistance, na.rm = TRUE),
    avg_calories = mean(Calories, na.rm = TRUE)
  )

# View the result
print(daily_avg_analysis)
```

```
## # A tibble: 7 × 4
##   day_of_week avg_steps avg_dist avg_calories
##   <chr>           <dbl>    <dbl>        <dbl>
## 1 Friday          7448.     5.31        2332.
## 2 Monday          7819.     5.59        2338.
## 3 Saturday        8203.     5.90        2366.
## 4 Sunday          6933.     5.03        2263
## 5 Thursday        7421.     5.33        2204.
## 6 Tuesday         8125.     5.83        2356.
## 7 Wednesday       7559.     5.49        2303.
```

Result:- No trends/patterns found

```
#TIME EXPENDITURE PER DAY
#head(dailyActivity_merged)
# Summarizing time expenditure per user
time_expenditure <- dailyActivity_merged %>%
  filter(!is.na(TotalTimeInBed)) %>%  # Exclude rows where total_mins_bed is NULL
  group_by(Id) %>%
  summarise(
    sedentary_mins = sum(SedentaryMinutes, na.rm = TRUE),
    lightly_mins = sum(LightlyActiveMinutes, na.rm = TRUE),
    fairlyactive_mins = sum(FairlyActiveMinutes, na.rm = TRUE),
    veryactive_mins = sum(VeryActiveMinutes, na.rm = TRUE)
  )

# View the result
print(time_expenditure)
```

```
## # A tibble: 24 × 5
##            Id sedentary_mins lightly_mins fairlyactive_mins veryactive_mins
##         <dbl>          <int>        <int>             <int>           <int>
##  1 1503960366          18982         5828               507             948
##  2 1644430081           3682          965                78              10
##  3 1844505072           1330          435                 7               0
##  4 1927972279           4886          425                 0               0
##  5 2026352035          18311         7182                 8               3
##  6 2320127002           1129          242                 0               0
##  7 2347167796           9426         3688               242             138
##  8 3977333714          20054         5083              1716             555
##  9 4020332650           6735         1748               124             120
## 10 4319703577          16710         6352               320              68
## # ℹ 14 more rows
```

# visualization of Data

```
# Load required libraries
library(ggplot2)
library(dplyr)
library(plotly)  # For tooltips
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
# Scatter plot for LightlyActiveMinutes vs Calories
p1 <- ggplot(dailyActivity_merged, aes(x = LightlyActiveMinutes, y = Calories)) +
  geom_point(aes(text = paste("Lightly Active Minutes:", LightlyActiveMinutes, "<br>Calorie
s:", Calories)),
             color = "blue", alpha = 0.6) +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Lightly Active Minutes vs Calories Burned",
       x = "Lightly Active Minutes",
       y = "Calories Burned") +
  theme_minimal()
```

```
## Warning in geom_point(aes(text = paste("Lightly Active Minutes:",
## LightlyActiveMinutes, : Ignoring unknown aesthetics: text
```

```
# Scatter plot for FairlyActiveMinutes vs Calories
p2 <- ggplot(dailyActivity_merged, aes(x = FairlyActiveMinutes, y = Calories)) +
  geom_point(aes(text = paste("Fairly Active Minutes:", FairlyActiveMinutes, "<br>Calories:",
Calories)),
             color = "green", alpha = 0.6) +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Fairly Active Minutes vs Calories Burned",
       x = "Fairly Active Minutes",
       y = "Calories Burned") +
  theme_minimal()
```

```
## Warning in geom_point(aes(text = paste("Fairly Active Minutes:",
## FairlyActiveMinutes, : Ignoring unknown aesthetics: text
```

```
# Scatter plot for VeryActiveMinutes vs Calories
p3 <- ggplot(dailyActivity_merged, aes(x = VeryActiveMinutes, y = Calories)) +
  geom_point(aes(text = paste("Very Active Minutes:", VeryActiveMinutes, "<br>Calories:", Cal
ories)),
             color = "purple", alpha = 0.6) +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Very Active Minutes vs Calories Burned",
       x = "Very Active Minutes",
       y = "Calories Burned") +
  theme_minimal()
```

```
## Warning in geom_point(aes(text = paste("Very Active Minutes:",
## VeryActiveMinutes, : Ignoring unknown aesthetics: text
```

```
# Convert ggplot objects to interactive plots with tooltips
p1_interactive <- ggplotly(p1)
```

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

```
p2_interactive <- ggplotly(p2)
```

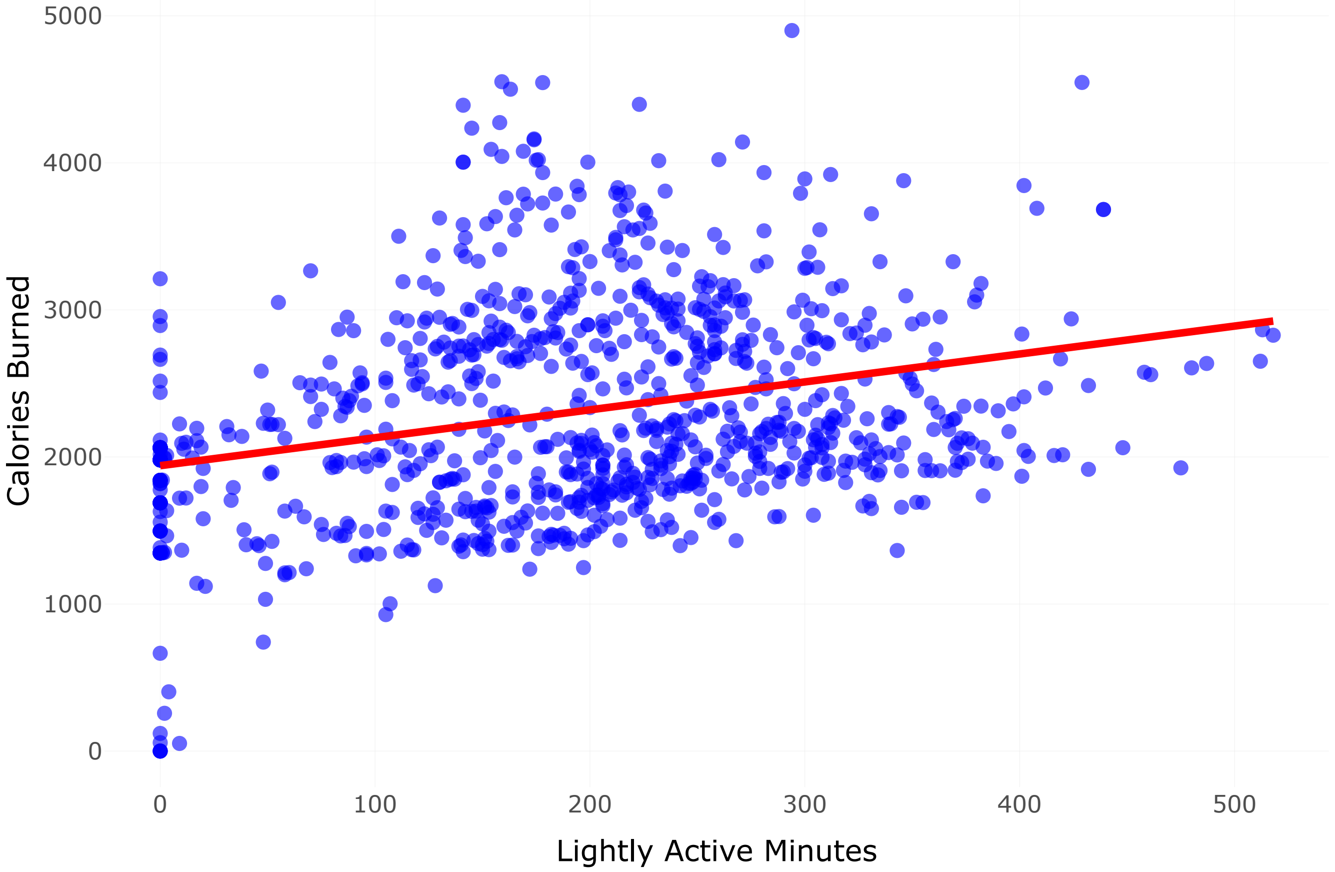
```
## `geom_smooth()` using formula = 'y ~ x'
```

```
p3_interactive <- ggplotly(p3)
```

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

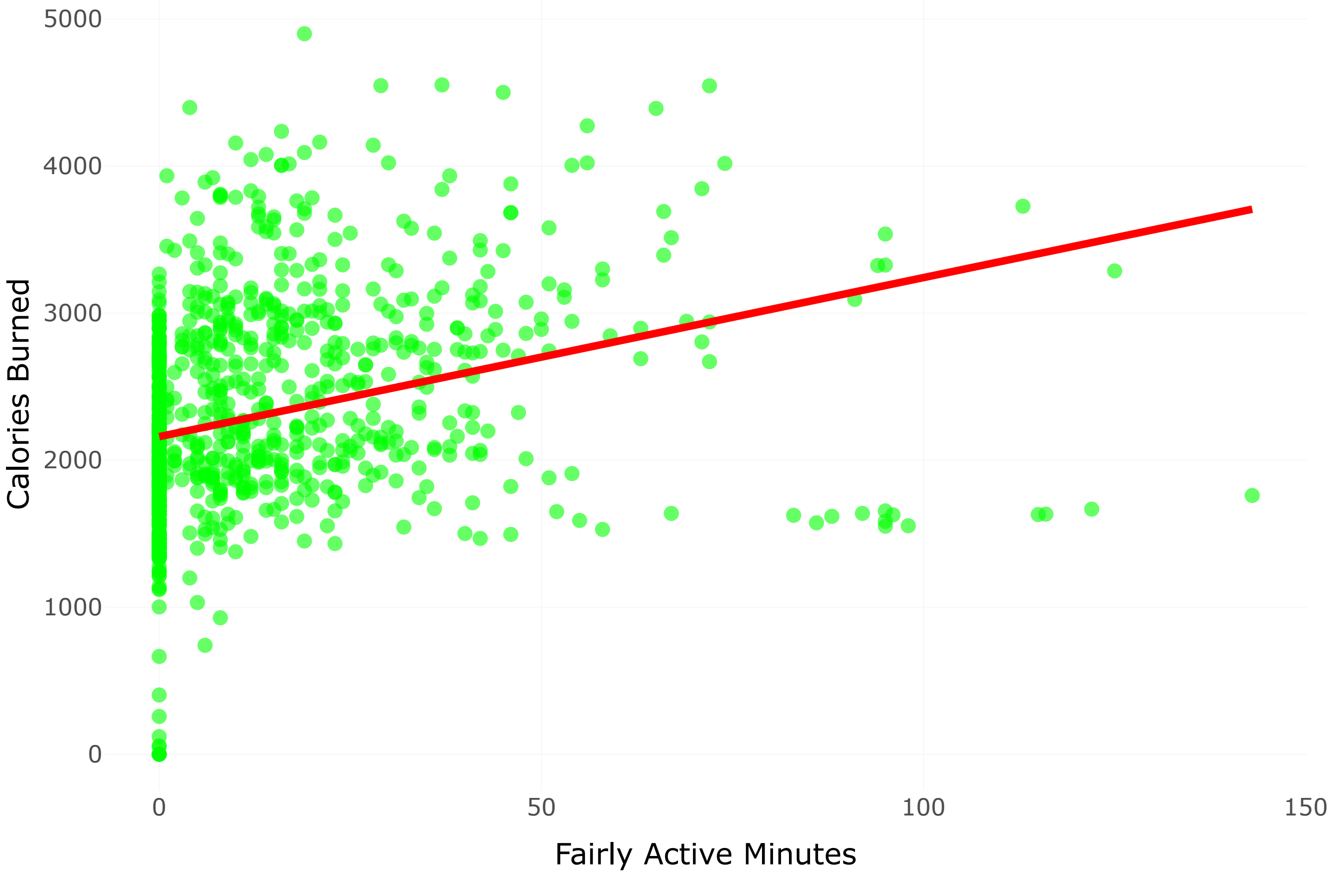
```
# Display interactive graphs
p1_interactive
```

Lightly Active Minutes vs Calories Burned

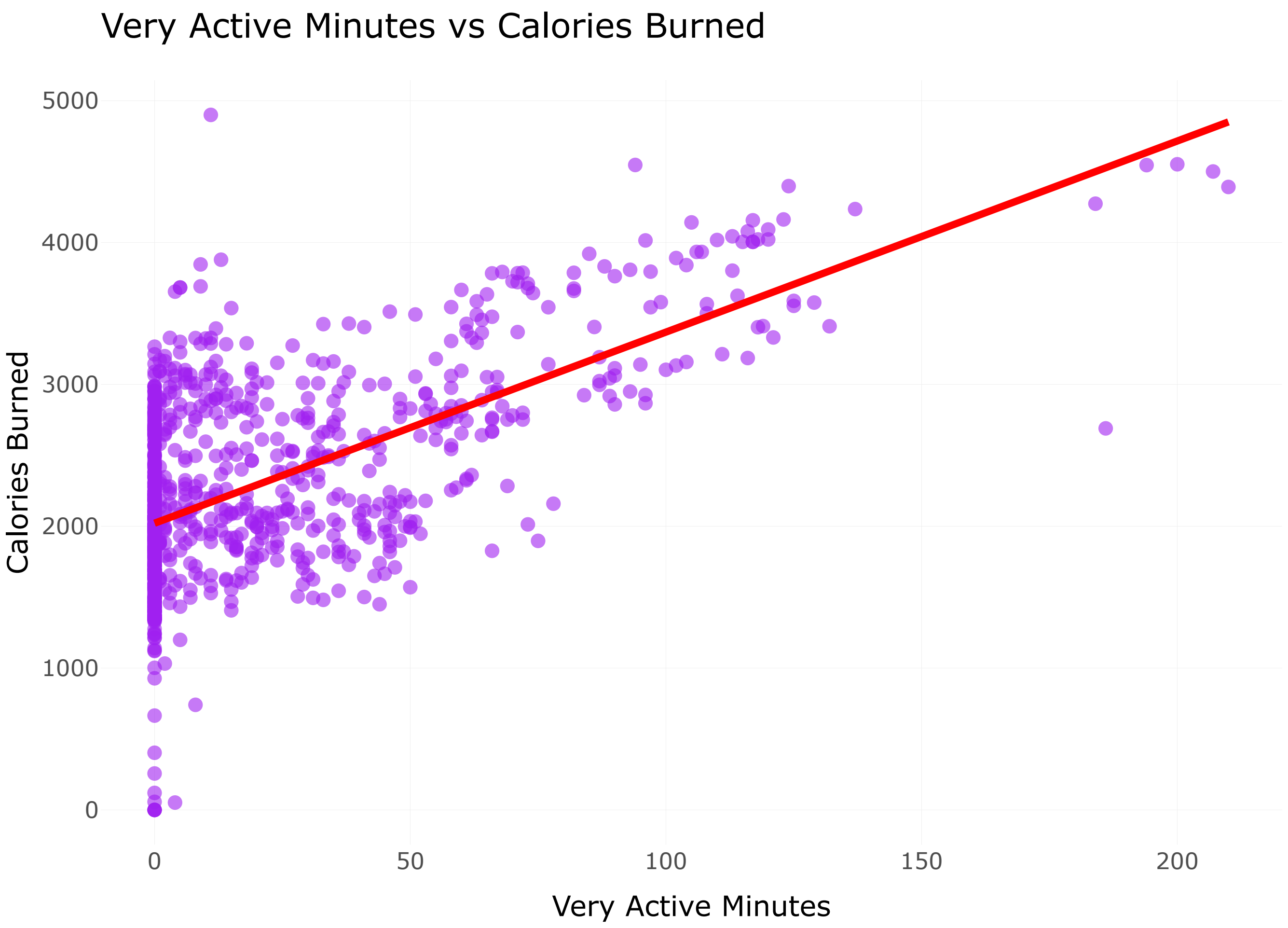


```
p2_interactive
```

Fairly Active Minutes vs Calories Burned



```
p3_interactive
```



## key findings

- There is a strong correlation between Very Active minutes and the amount of calories burned.
- The respective trend lines of the graphs, we can conclude that the higher the intensity and the duration of the activity, the more calories is burned.

# METs and Average Calories Burned

The Metabolic Equivalent of Task (MET) is a measure used to describe the energy expenditure of different activities compared to resting energy consumption.

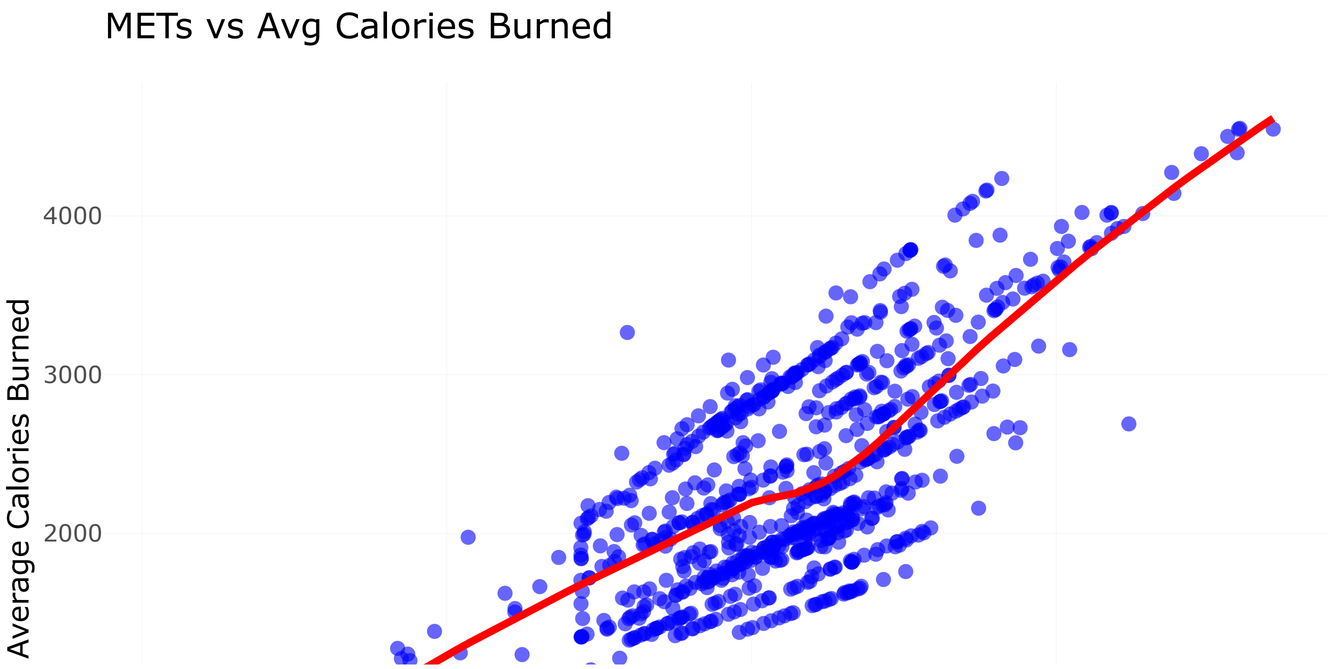
```
# Load necessary libraries
library(ggplot2)
library(dplyr)
library(plotly)

#  dataset is `minuteMETsNarrow_merged`with calories column added
df <- final_data

# Calculate average calories burned per MET level
df_summary <- df %>%
  group_by(sum_mets) %>%
  summarize(avg_calories = mean(Calories, na.rm = TRUE))

# Create a scatter plot with tooltips
p <- ggplot(df_summary, aes(x = sum_mets, y = avg_calories)) +
  geom_point(aes(text = paste("METs:", sum_mets, "<br>Avg Calories:", round(avg_calories,
2))),
             color = "blue", alpha = 0.6) +  # Scatter plot with tooltips
  geom_smooth(method = "loess", color = "red", se = FALSE) +  # Regression trend line
  labs(title = "METs vs Avg Calories Burned",
       x = "Metabolic Equivalent of Task (METs)",
       y = "Average Calories Burned") +
  theme_minimal()
```

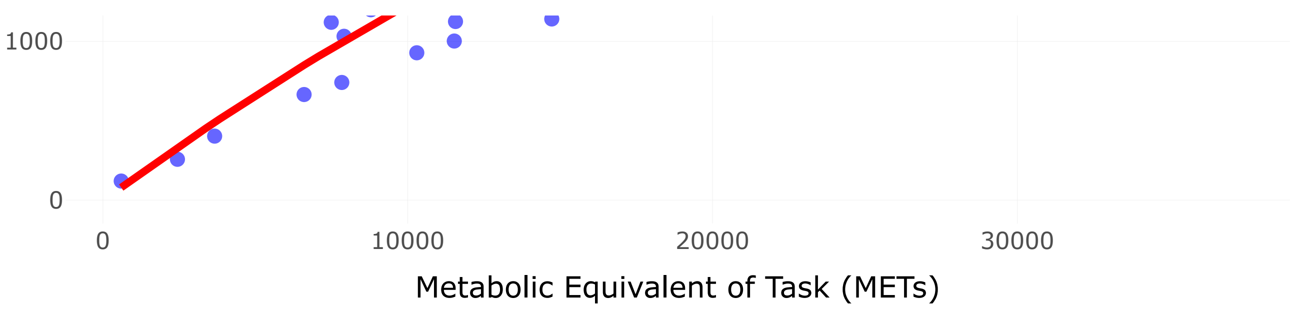
```
## Warning in geom_point(aes(text = paste("METs:", sum_mets, "<br>Avg Calories:",
## : Ignoring unknown aesthetics: text
```

```
# Convert ggplot to interactive plotly object
p_interactive <- ggplotly(p)
```

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

```
# Display interactive graph
p_interactive
```





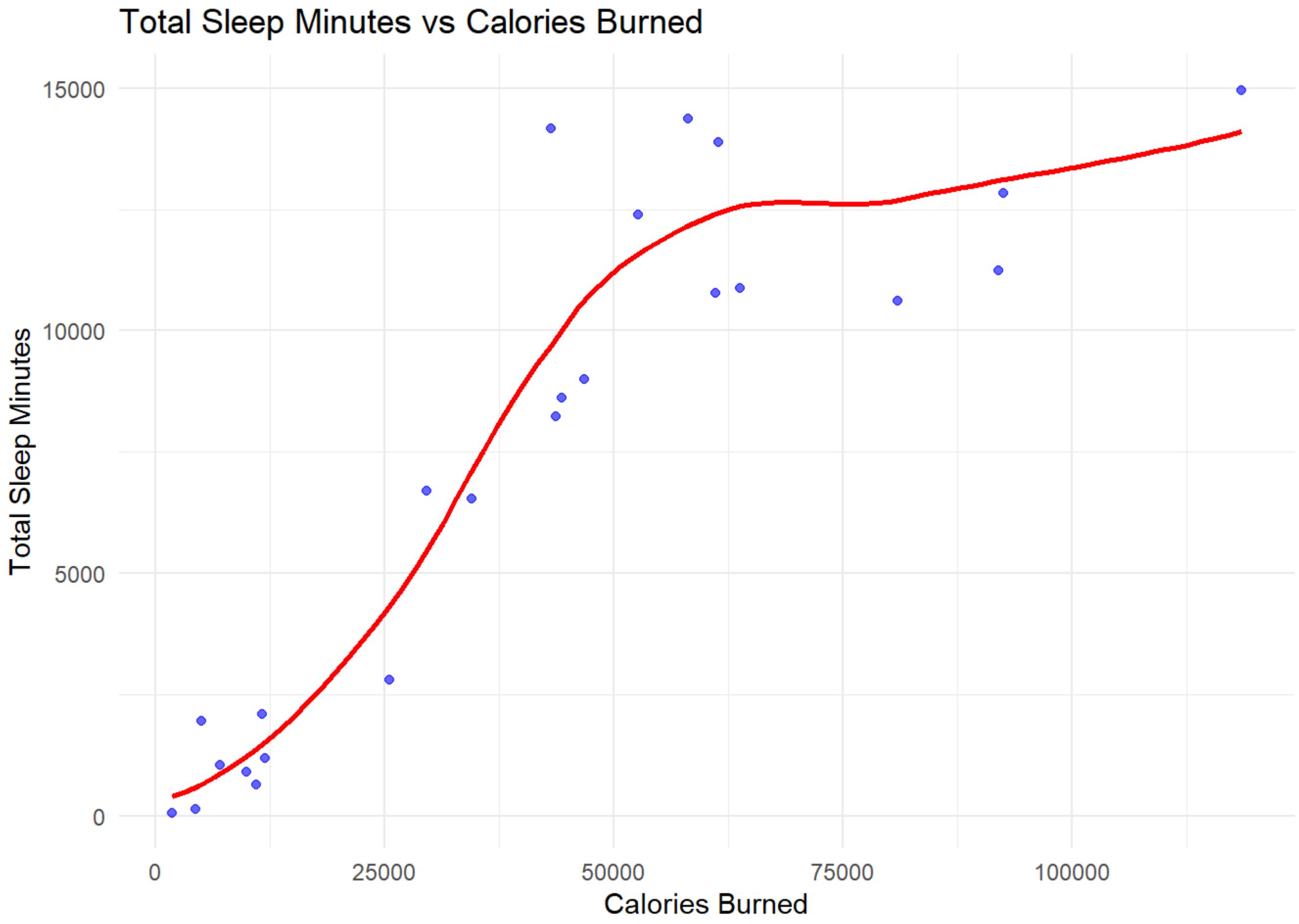
## key findings

- Strong positive corellation between METs and average calories burned.
- The amount of calories burned for every user is highly dependent on their MET values they spend every day.

# Sleep and Calories Comparison

```
# Load required libraries
library(ggplot2)

# Create scatter plot with a smooth trend line
ggplot(sleep_calories_summary, aes(x = calories, y = total_sleep_m)) +
  geom_point(color = "blue", alpha = 0.6) +  # Scatter plot points
  geom_smooth(method = "loess", color = "red", se = FALSE) +  # Smooth trend line
  labs(title = "Total Sleep Minutes vs Calories Burned",
       x = "Calories Burned",
       y = "Total Sleep Minutes") +
  theme_minimal()
```

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

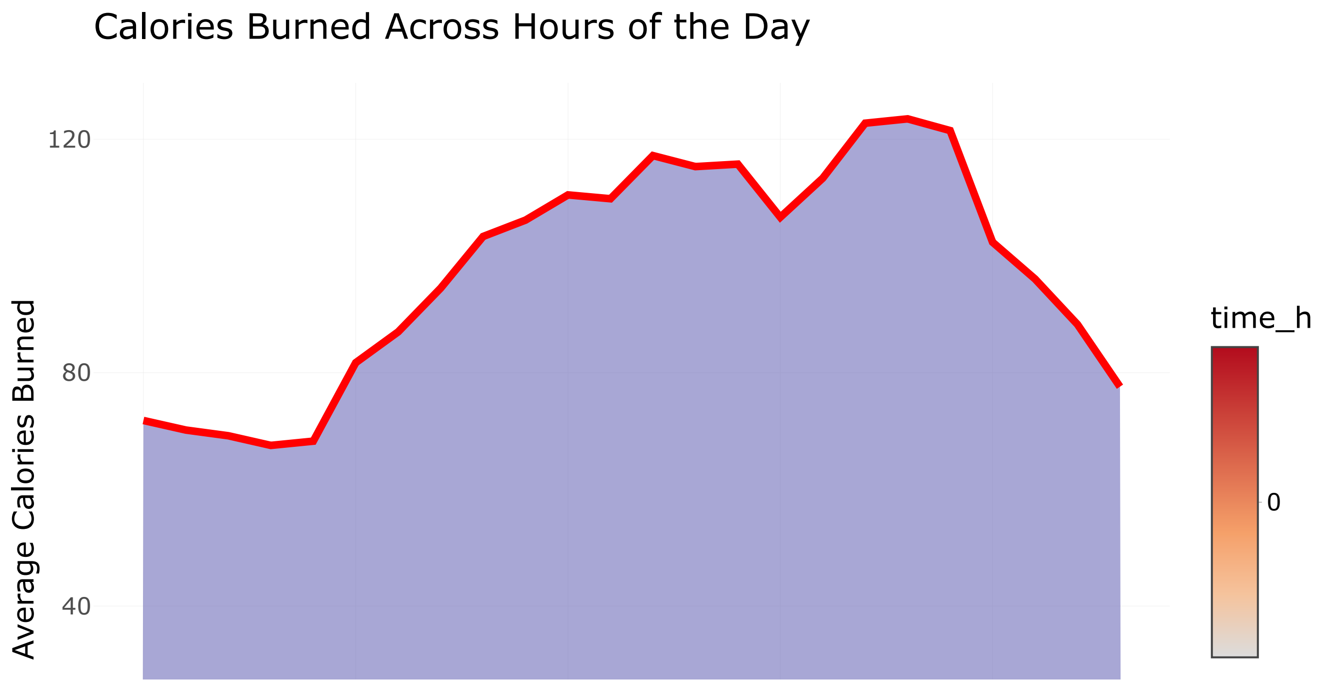


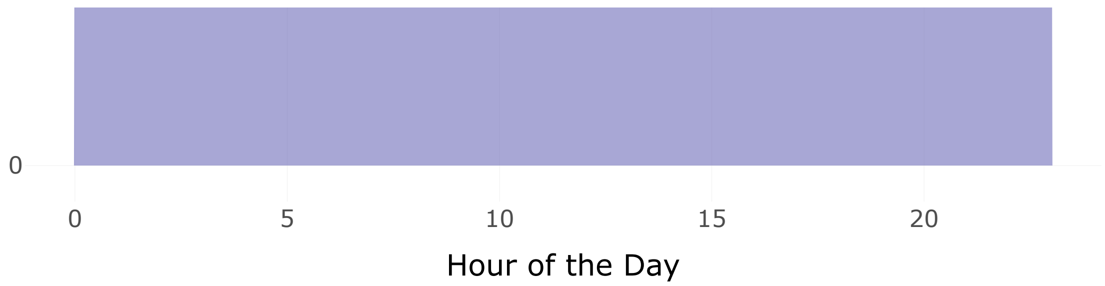
# Key Findings

- Strong positive corellation between amount of sleep and calories burned. Higher duration of sleep is associated with higher amount of calories burned.
- An adequate duration and good quality of sleep constitutes to higher calories burned during the sleeping process.
- However sleeping more than the required range doesn’t seem to burn more calories and in fact causes the opposite to occur, which is burn fewer calories.

# Popular Time for Activities

```
# Load required libraries
library(ggplot2)
library(dplyr)
library(plotly)

# Assuming your dataset is `hourlyCalories_merged`
df <- hourlyCalories_merged
# Convert DateTime column (if needed)
#df$ActivityHour <- as.POSIXct(df$ActivityHour, format="%m/%d/%Y %I:%M:%S %p")

# Extract hour from ActivityHour
df <- df %>%
 # mutate(hour = format(ActivityHour, "%H")) %>%  # Extract hour as character
  group_by(time_h) %>%
  summarize(avg_calories = mean(Calories, na.rm = TRUE))  # Aggregate calories by hour
#head(hourlyCalories_merged)
# Convert hour to numeric for proper sorting
df$time_h <- as.numeric(df$time_h)

# Create area chart with tooltips
p <-ggplot(df, aes(x = time_h, y = avg_calories)) +
  geom_area(aes(fill = time_h), alpha = 0.6) +  # Color mapped to hour
  geom_line(color = "red", linewidth= 1) +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +  # Adjust gradient fill
  labs(title = "Calories Burned Across Hours of the Day",
       x = "Hour of the Day",
       y = "Average Calories Burned") +
  theme_minimal()

# Convert ggplot to interactive plotly object
p_interactive <- ggplotly(p)

# Display interactive graph with tooltips
p_interactive
```



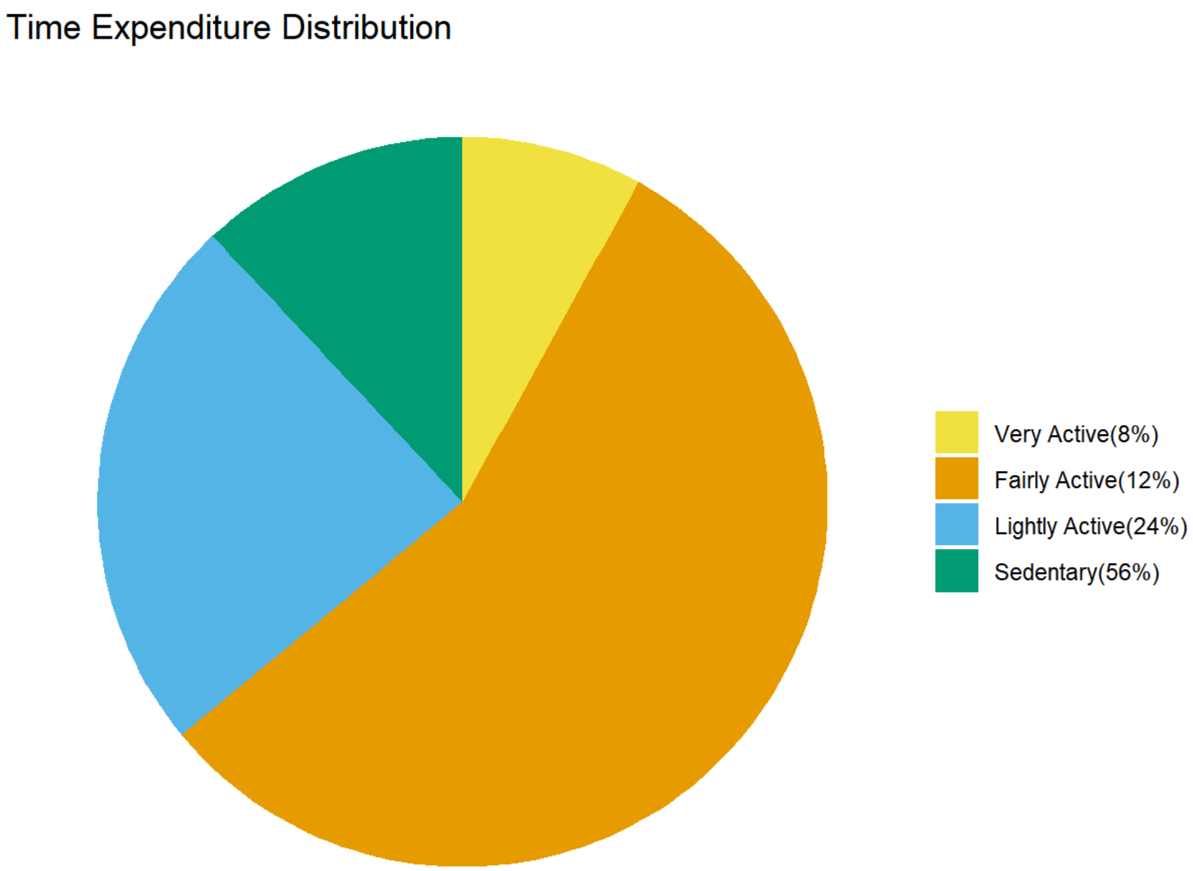


# Key Findings

- From the graph above, we can infer that the most popular time people are active throughout the day is between 5:00 AM - 21:00PM

# Time distribution chart

```
# Load necessary libraries
library(ggplot2)
library(dplyr)

#  time_expenditure table data
time_expenditure <- data.frame(
  category = c("Sedentary", "Lightly Active", "Fairly Active", "Very Active"),
  minutes = c(700, 300, 150, 100)
)

# Calculate percentage
time_expenditure <- time_expenditure %>%
  mutate(percentage = minutes / sum(minutes) * 100,
         label = paste0(  round(percentage, 1), "%"))

# Create a pie chart
ggplot(time_expenditure, aes(x = "", y = minutes, fill = category)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  theme_void() +
  theme(legend.title = element_blank()) +
  labs(title = "Time Expenditure Distribution") +
  scale_fill_manual(values = c("Sedentary" = "#E69F00",
                               "Lightly Active" = "#56B4E9",
                               "Fairly Active" = "#009E73",
                               "Very Active" = "#F0E442"),
                    labels = paste(time_expenditure$category, "(", round(time_expenditure$per
centage, 1), "%)", sep = "")) +
  guides(fill = guide_legend(reverse = TRUE))
```



```
# Save the chart
ggsave("time_expenditure_pie_chart.png", width = 6, height = 6)
```

# CONCLUSION

- Activity Duration & Intensity Affect Calories Burned – Longer and higher-intensity activities lead to more calories burned.
- METs Provide Valuable Insights – MET values help measure activity intensity and calorie expenditure.
- Sleep Patterns Vary – Most users have adequate sleep, but some oversleep or undersleep, affecting health.
- Peak Activity Hours – Users are most active between 5:00 AM - 9:00 PM, indicating ideal times for fitness engagement.

# RECOMMENDATIONS.

- Highlight MET Tracking – Promote MET-based tracking in smart devices to provide users deeper insights into calorie burn.
- Activity Notifications – Implement smart device alerts to encourage movement during peak activity times (5:00 AM - 9:00 PM).
- Improve Sleep Tracking Features – notifications for better sleep habits.
- Gamify Calorie Burn – Launch weekly/daily calorie challenges where users earn points for burning calories, redeemable for product discounts.