DATA ANALYSIS
BELLABEAT CASE STUDY WITH R PROGRAM

ABOUT THE COMPANY

Bellabeat is a high-tech company designed to create innovation-focused health products for women. This company, established by Urška Sršen and Sandro Mur, established very rapidly as one of the bright leaders of the wellness industry through very beautifully designed smart devices monitoring sleep, activity, stress, and reproductive health. Bellabeat's focus on the use of technology for the welfare of women puts it in a high place as one of the major players in the global market of smart devices that could lead to growth and innovation in the highly competitive field.

Key Study Roadmap

I. Ask

1. What are some trends in smart device usage?
2. How could these trends apply to Bellabeat customers?
3. How could these trends help influence Bellabeat marketing strategy?

Business Task

To identify potential opportunities for growth and provide strategic recommendations for the enhancement of Bellabeat’s marketing strategy, leveraging current trends in smart device usage.

Key Stakeholders

- Urška Sršen - Bellabeat’s co-founder and Chief Creative Officer
- Sando Mur - Mathematician and Bellabeat’s co-founder
- Bellabeat Market Analytics Team

II. Prepare

The dataset for this project does not require permission from the dataset provider and can be free to use and redistribute. It is, therefore, available under FitBit Fitness Tracker Data (CC0: Public Domain) and is offered through Mobius. This is a Kaggle dataset, personal fitness tracker data from thirty Fitbit users who have given consent to submit their personal tracker data, which includes minute-level output for physical activity, heart rate, and sleep monitoring. The dataset contains daily activity, steps, and heart rate, which one can analyze to see the habits of users. The first thing is to import the CSV file and inspect trends from the recorded day-to-day data.

III. Process

For the case study, I utilized the R programming language. Initially, I installed and loaded the necessary libraries to ensure all required functionalities were available. Following this, I imported the CSV files containing the dataset and used the head() function to preview the data, allowing for a quick initial assessment.

Subsequently, I engaged in data cleaning, which involved identifying and handling missing values, correcting data types, and removing any inconsistencies. This step was crucial to ensure the integrity and accuracy of the dataset.

Next, I performed data manipulation, which included transforming variables, creating new variables, and aggregating data as needed to align with the objectives of the analysis. Finally, I merged the datasets to create a comprehensive dataset suitable for in-depth analysis.

Installing the Libraries

```
> install.packages("tidyverse")
> install.packages("lubridate")
> install.packages("dplyr")
> install.packages("ggplot2")
> install.packages("tidyr")
```

Loading the Libraries

```
> library("tidyverse")
> library("lubridate")
> library("dplyr")
> library("ggplot2")
> library("tidyr")
```

Importing CSV files

```
> daily_activity <- read_csv("D:\\Downloads\\Compressed\\archive\\mturkfitbit_export_4.12.16-
5.12.16\\Fitabase Data 4.12.16-5.12.16\\dailyActivity_merged.csv")

> head(daily_activity)
# A tibble: 6 × 15
          Id ActivityDate TotalSteps TotalDistance TrackerDistance LoggedActivitiesDistance VeryActiveDistance
ModeratelyActiveDista…¹ LightActiveDistance SedentaryActiveDista…² VeryActiveMinutes
FairlyActiveMinutes LightlyActiveMinutes
       <dbl> <chr>            <dbl>         <dbl>           <dbl>                    <dbl>              <dbl>
<dbl>                   <dbl>               <dbl>                  <dbl>
1 1503960366 4/12/2016        13162         8.5             8.5                        0               1.88                  0.550
6.06                    0                   25                     13                 328
2 1503960366 4/13/2016        10735         6.97            6.97                       0               1.57                  0.690
4.71                    0                   21                     19                 217
3 1503960366 4/14/2016        10460         6.74            6.74                       0               2.44                  0.400
3.91                    0                   30                     11                 181
4 1503960366 4/15/2016         9762         6.28            6.28                       0               2.14                  1.26
2.83                    0                   29                     34                 209
5 1503960366 4/16/2016        12669         8.16            8.16                       0               2.71                  0.410
5.04                    0                   36                     10                 221
6 1503960366 4/17/2016         9705         6.48            6.48                       0               3.19                  0.780
2.51                    0                   38                     20                 164
# ℹ abbreviated names: ¹ModeratelyActiveDistance, ²SedentaryActiveDistance
# ℹ 2 more variables: SedentaryMinutes <dbl>, Calories <dbl>

> calories <- read_csv("D:\\Downloads\\Compressed\\archive\\mturkfitbit_export_4.12.16-5.12.16\\Fitabase
Data 4.12.16-5.12.16\\hourlyCalories_merged.csv")

> head(calories)
# A tibble: 6 × 3
          Id ActivityHour          Calories
       <dbl> <chr>                    <dbl>
1 1503960366 4/12/2016 12:00:00 AM       81
2 1503960366 4/12/2016 1:00:00 AM        61
3 1503960366 4/12/2016 2:00:00 AM        59
4 1503960366 4/12/2016 3:00:00 AM        47
5 1503960366 4/12/2016 4:00:00 AM        48
6 1503960366 4/12/2016 5:00:00 AM        48

> intensities <- read_csv("D:\\Downloads\\Compressed\\archive\\mturkfitbit_export_4.12.16-5.12.16\\Fitabase
Data 4.12.16-5.12.16\\hourlyIntensities_merged.csv")

> head(intensities)
```

```
# A tibble: 6 × 4
        Id ActivityHour          TotalIntensity AverageIntensity
     <dbl> <chr>                          <dbl>            <dbl>
1 1503960366 4/12/2016 12:00:00 AM           20            0.333
2 1503960366 4/12/2016 1:00:00 AM             8            0.133
3 1503960366 4/12/2016 2:00:00 AM             7            0.117
4 1503960366 4/12/2016 3:00:00 AM             0            0
5 1503960366 4/12/2016 4:00:00 AM             0            0
6 1503960366 4/12/2016 5:00:00 AM             0            0

> steps <- read_csv("D:\\Downloads\\Compressed\\archive\\mturkfitbit_export_4.12.16-5.12.16\\Fitabase Data
4.12.16-5.12.16\\hourlySteps_merged.csv")

> head(steps)
# A tibble: 6 × 3
        Id ActivityHour          StepTotal
     <dbl> <chr>                     <dbl>
1 1503960366 4/12/2016 12:00:00 AM     373
2 1503960366 4/12/2016 1:00:00 AM      160
3 1503960366 4/12/2016 2:00:00 AM      151
4 1503960366 4/12/2016 3:00:00 AM        0
5 1503960366 4/12/2016 4:00:00 AM        0
6 1503960366 4/12/2016 5:00:00 AM        0

> sleep <- read_csv("D:\\Downloads\\Compressed\\archive\\mturkfitbit_export_4.12.16-5.12.16\\Fitabase Data
4.12.16-5.12.16\\sleepDay_merged.csv")

> head(sleep)
# A tibble: 6 × 5
        Id SleepDay              TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
     <dbl> <chr>                             <dbl>              <dbl>          <dbl>
1 1503960366 4/12/2016 12:00:00 AM               1                327            346
2 1503960366 4/13/2016 12:00:00 AM               2                384            407
3 1503960366 4/15/2016 12:00:00 AM               1                412            442
4 1503960366 4/16/2016 12:00:00 AM               2                340            367
5 1503960366 4/17/2016 12:00:00 AM               1                700            712
6 1503960366 4/19/2016 12:00:00 AM               1                304            320
```

Cleaning, Modifying & Creating Data Sets

```
> daily_activity$Date <- as.Date(daily_activity$ActivityDate, format = "%m/%d/%Y")
> daily_activity <- daily_activity %>% mutate(WeekDay=weekdays(Date), WeekNo=format(Date, "%U"))
> daily_activity_v1 <- daily_activity %>% mutate(TotalActiveMinutes =
VeryActiveMinutes+FairlyActiveMinutes+LightlyActiveMinutes)
```

Since the Date column was in character format, I converted it to date format to facilitate easier referencing. Additionally, I added new columns to represent the weekday and week number. Subsequently, I created a new dataset that included an additional column for the total sum of the VeryActive, FairlyActive, and LightlyActive columns.

```
> AvgTimeActive <- daily_activity_v1 %>%
  group_by(WeekDay) %>%
  summarize(
        AvgVeryActiveMinutes = mean(VeryActiveMinutes, na.rm = TRUE),
        AvgFairlyActiveMinutes = mean(FairlyActiveMinutes, na.rm = TRUE),
        AvgLightlyActiveMinutes = mean(LightlyActiveMinutes, na.rm = TRUE),
        AvgTimeActiveMinutes = mean(TotalActiveMinutes, na.rm = TRUE),
        AvgTimeSedentary = mean(SedentaryMinutes, na.rm = TRUE),
        AvgTotalSteps = mean(TotalSteps, na.rm = TRUE),
        AvgCaloriesBurned = mean(Calories, na.rm = TRUE))

> AvgTimeActive$WeekDay <- factor(AvgTimeActive$WeekDay, levels = c("Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))
> AvgTimeActive <- AvgTimeActive[order(AvgTimeActive$WeekDay),]
```

I created a new dataset by calculating the average values of specific columns from the original dataset. Additionally, I arranged the rows to ensure that the first row corresponds to Monday and the last row to Sunday.

```
> AvgPerWeek <- daily_activity_v1 %>%
   group_by(WeekNo) %>%
   summarize(
            AvgTotalSteps = mean(TotalSteps, na.rm = TRUE),
            AvgCaloriesBurned = mean(Calories, na.rm = TRUE))
```

```
> hour_merged <- merge(calories, intensities, by=c("Id", "ActivityHour"))
> hour_merged <- merge(hour_merged, steps, by=c("Id", "ActivityHour"))

> hour_merged$ActivityHour <- mdy_hms(hour_merged$ActivityHour)
> hour_merged$Date <- as.Date(hour_merged$ActivityHour)
> hour_merged$Time <- format(hour_merged$ActivityHour, format="%H:%M:%S")

AvgPerHour <- hour_merged %>%
group_by(Time) %>%
summarize(
            AvgCalories = mean(Calories, na.rm = TRUE),
            AvgIntensity = mean(TotalIntensity, na.rm = TRUE),
            AvgSteps = mean(StepTotal, na.rm = TRUE))
```

In this process, I initially merged the datasets of Calories and Intensities, followed by merging the resulting dataset with the Steps dataset. Subsequently, I corrected the format of the ActivityHour column and derived two new columns containing the Date and Time data. Finally, I created a new dataset by calculating the average values of Calories, Intensities, and Steps, grouping them according to the corresponding time.

```
> sleep$SleepDay <- mdy_hms(sleep$SleepDay)
> sleep$Date <- as.Date(sleep$SleepDay)

> daily_activity_sedentary <- daily_activity_v1 %>% select(Id, Date, TotalActiveMinutes, SedentaryMinutes)

> sleep_activity_sedentary_merged <- sleep %>% left_join(daily_activity_sedentary, by = c("Id", "Date"))
```

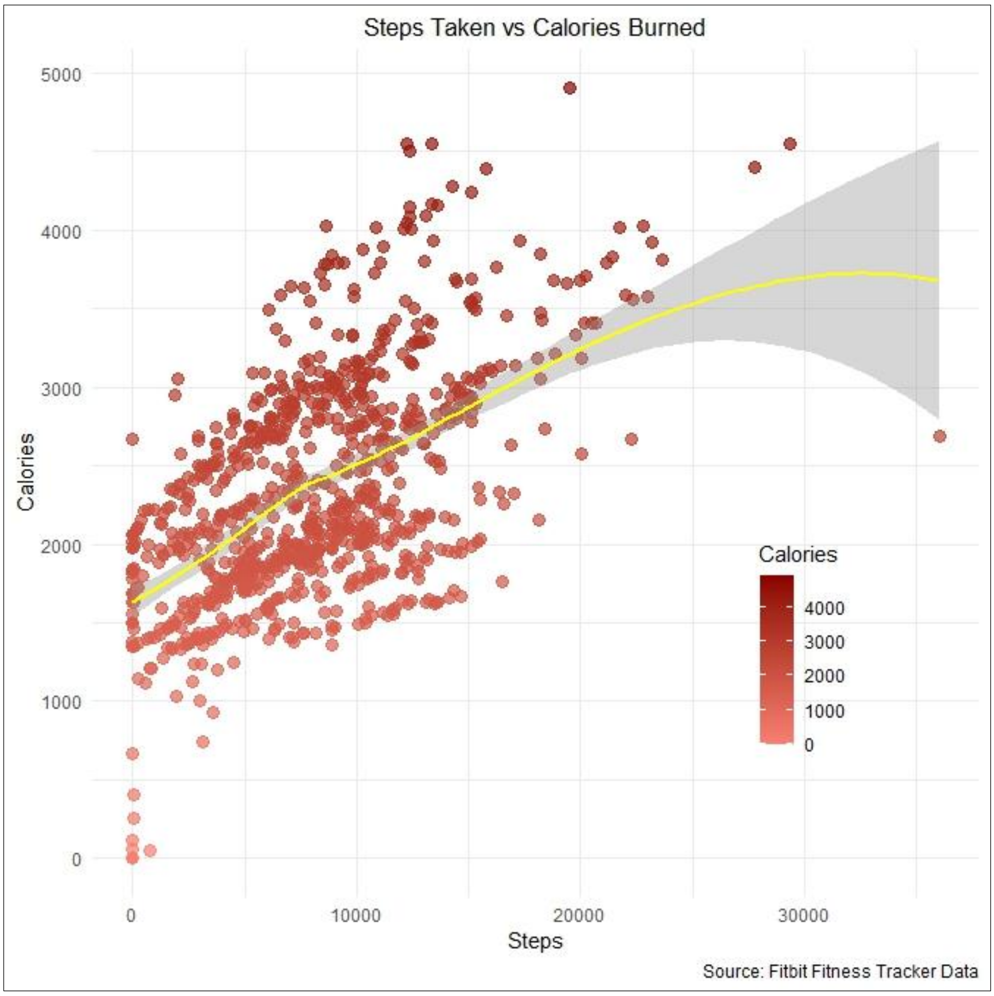
I created a new dataset by extracting four specific columns (Id, Date, TotalActiveMinutes, SedentaryMInutes) from the original dataset. Subsequently, I merged this dataset with the Sleep dataset using a left join, as the number of rows differed between the two datasets.

IV. Analyze

In this section of the study, I will extract relevant data and generate visualizations to identify and analyze trends. This process will involve selecting key metrics, creating charts and graphs, and interpreting the results to gain insights into the underlying patterns and relationships within the dataset.

STEPS & CALORIES

```
daily_activity_v1 %>%
  ggplot(aes(x = TotalSteps, y = Calories, color = Calories)) +
  geom_point(size = 3, alpha = 0.7) +
  geom_smooth(method = "loess", formula = y ~ x, color = "yellow", se = TRUE) +
  scale_color_gradient(low = "salmon", high = "red4") +
  theme_minimal() +
  theme(
    legend.position = c(0.8, 0.3),
    legend.spacing.y = unit(1, "mm"),
    plot.title = element_text(hjust = 0.5)) +
  labs(
    title = "Steps Taken vs Calories Burned",
    x = "Steps",
    y = "Calories",
    caption = "Source: Fitbit Fitness Tracker Data")
```



The graph illustrates a positive correlation between the number of **steps taken** and the **calories burned**. This indicates that as the number of steps increases, the amount of calories burned also rises. This relationship suggests that higher physical activity, as measured by the number of steps, leads to greater energy expenditure.

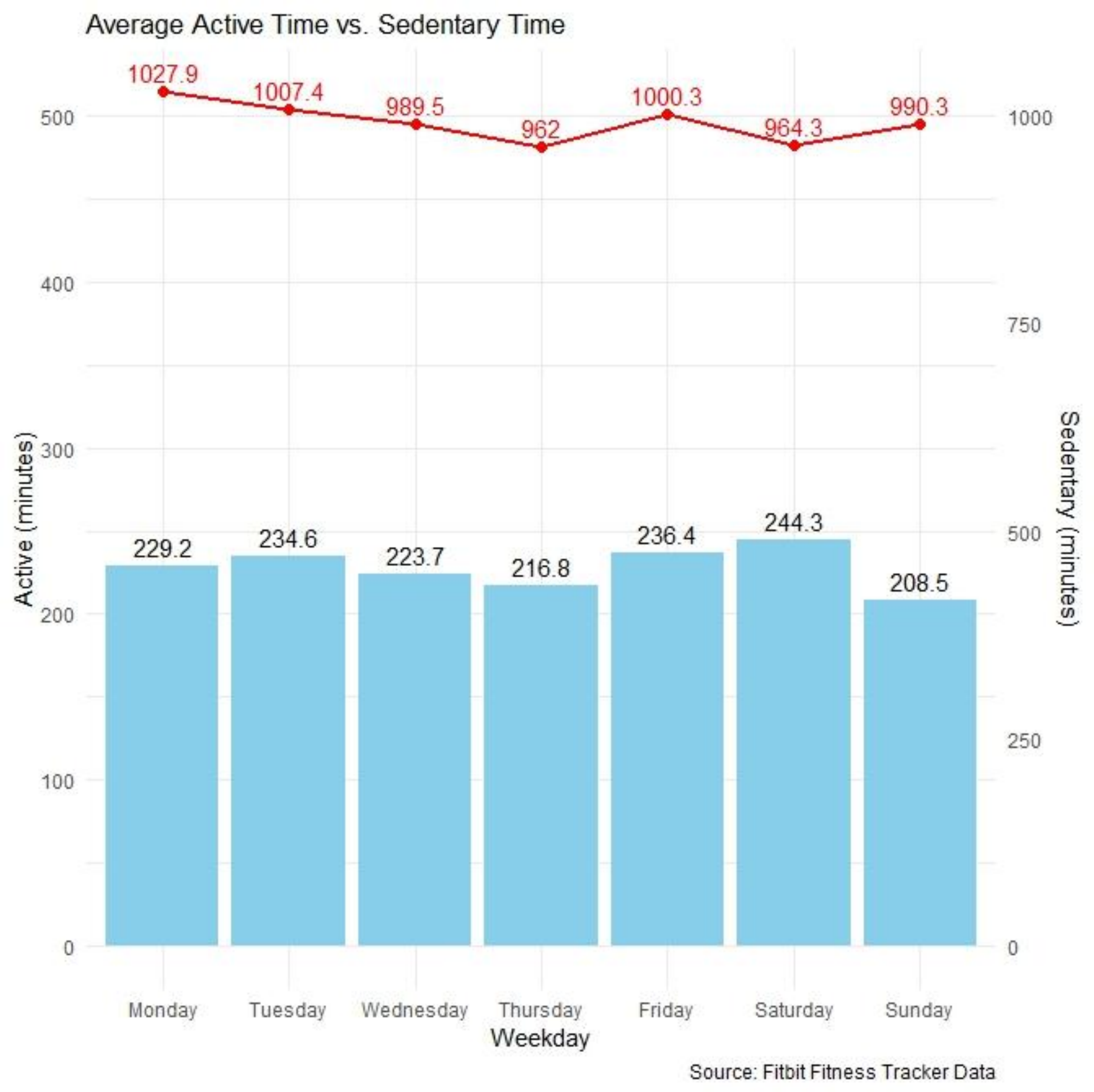
ACTIVE & SEDENTARY TIME

```
> daily_activity_v1 %>% select(TotalSteps, TotalDistance, VeryActiveMinutes, FairlyActiveMinutes, LightlyActiveMinutes, TotalActiveMinutes, SedentaryMinutes, Calories) %>% summary()

  TotalSteps    TotalDistance    VeryActiveMinutes FairlyActiveMinutes LightlyActiveMinutes TotalActiveMinutes
 Min.   :    0   Min.   : 0.000   Min.   :  0.00   Min.   :  0.00      Min.   :  0.0        Min.   :  0.0
 1st Qu.: 3790   1st Qu.: 2.620   1st Qu.:  0.00   1st Qu.:  0.00      1st Qu.:127.0        1st Qu.:146.8
 Median : 7406   Median : 5.245   Median :  4.00   Median :  6.00      Median :199.0        Median :247.0
 Mean   : 7638   Mean   : 5.490   Mean   : 21.16   Mean   : 13.56      Mean   :192.8        Mean   :227.5
 3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.: 32.00   3rd Qu.: 19.00      3rd Qu.:264.0        3rd Qu.:317.2
 Max.   :36019   Max.   :28.030   Max.   :210.00   Max.   :143.00      Max.   :518.0        Max.   :552.0
 SedentaryMinutes    Calories
 Min.   :   0.0   Min.   :   0
 1st Qu.: 729.8   1st Qu.:1828
 Median :1057.5   Median :2134
 Mean   : 991.2   Mean   :2304
 3rd Qu.:1229.5   3rd Qu.:2793
 Max.   :1440.0   Max.   :4900
```

```
AvgTimeActive %>%
  ggplot(aes(x = WeekDay)) +
  geom_bar(aes(y = AvgTimeActiveMinutes), stat = "identity", fill = "skyblue") +
  geom_text(aes(y = AvgTimeActiveMinutes, label = round(AvgTimeActiveMinutes, 1)), vjust = -0.5, color = "black") +
```

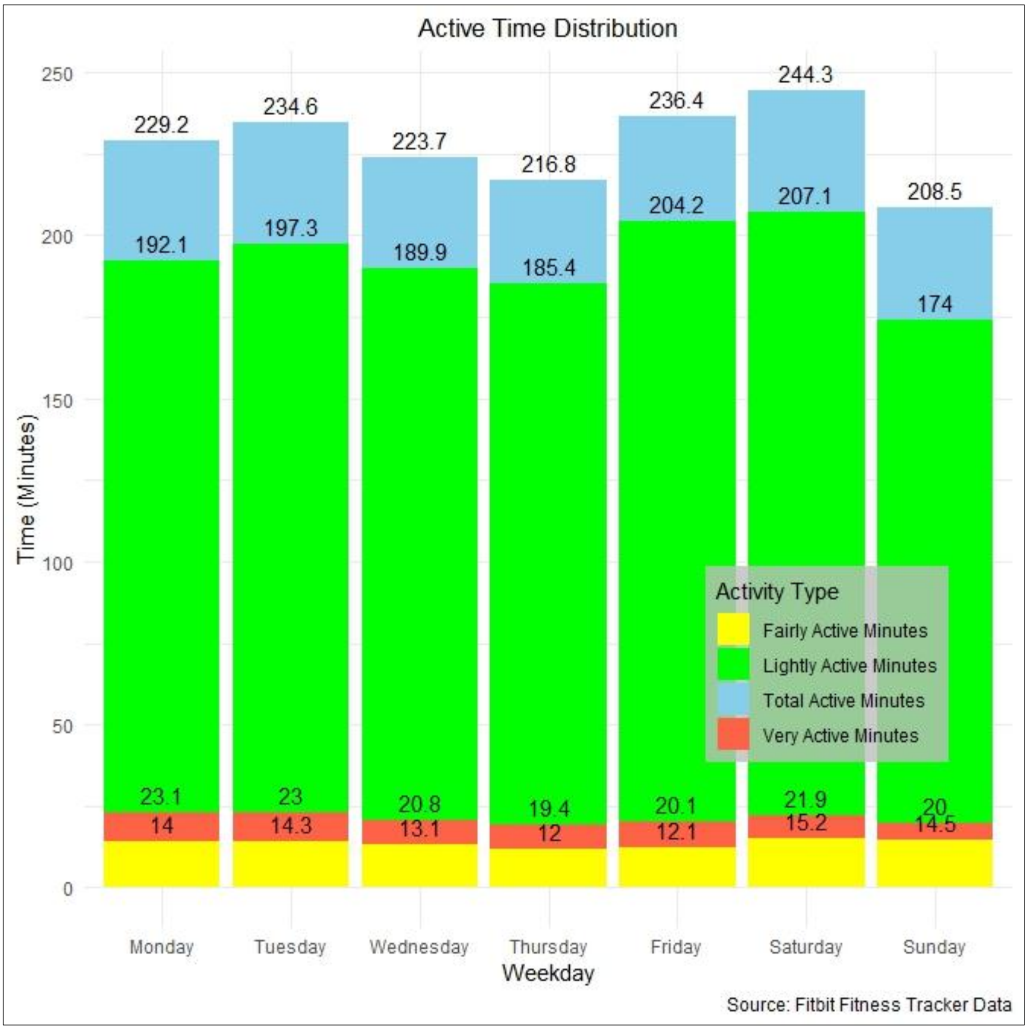
```
  geom_line(aes(y = AvgTimeSedentary * .5, group = 1), color = "red", size = 1) +
  geom_point(aes(y = AvgTimeSedentary * .5), color = "red", size = 2) +
  geom_text(aes(y = AvgTimeSedentary * .5, label = round(AvgTimeSedentary, 1)), vjust = -0.5, color = "red") +
  theme_minimal() +
  theme(legend.position = c(0.8, 0.3), legend.spacing.y = unit(1, "mm")) +
  labs(
    title = "Average Active Time vs. Sedentary Time",
    x = "Weekday",
    caption = "Source: Fitbit Fitness Tracker Data") +
  scale_y_continuous(
    name = "Active (minutes)",
    sec.axis = sec_axis(~.*2, name = "Sedentary (minutes)"))
```




```
AvgTimeActive %>%
  ggplot(aes(x = WeekDay)) +
  geom_bar(aes(y = AvgTimeActiveMinutes, fill = "Total Active Minutes"), stat = "identity", position = "dodge") +
  geom_bar(aes(y = AvgLightlyActiveMinutes, fill = "Lightly Active Minutes"), stat = "identity", position = "dodge")
+
  geom_bar(aes(y = AvgVeryActiveMinutes, fill = "Very Active Minutes"), stat = "identity", position = "dodge") +
  geom_bar(aes(y = AvgFairlyActiveMinutes, fill = "Fairly Active Minutes"), stat = "identity", position = "dodge")
+
```

```
  geom_text(aes(y = AvgTimeActiveMinutes, label = round(AvgTimeActiveMinutes, 1)), vjust = -0.5, position = position_dodge(width = 0.9), color = "black") +
  geom_text(aes(y = AvgLightlyActiveMinutes, label = round(AvgLightlyActiveMinutes, 1)), vjust = -0.5, position = position_dodge(width = 0.9), color = "black") +
  geom_text(aes(y = AvgVeryActiveMinutes, label = round(AvgVeryActiveMinutes, 1)), vjust = -0.5, position = position_dodge(width = 0.9), color = "black") +
  geom_text(aes(y = AvgFairlyActiveMinutes, label = round(AvgFairlyActiveMinutes, 1)), vjust = -0.5, position = position_dodge(width = 0.9), color = "black") +
  scale_fill_manual(
    values = c(
      "Total Active Minutes" = "skyblue",
      "Very Active Minutes" = "tomato",
      "Fairly Active Minutes" = "yellow",
      "Lightly Active Minutes" = "green")) +
  theme_minimal() +
  theme(
    legend.position = c(0.8, 0.3),
    legend.spacing.y = unit(1, "mm"),
    plot.title = element_text(hjust = 0.5),
    legend.background = element_rect(fill = alpha("gray", 0.8), color = NA)) +
  labs(
    title = "Active Time Distribution",
    x = "Weekday",
    y = "Time (Minutes)",
    fill = "Activity Type",
    caption = "Source: Fitbit Fitness Tracker Data")
```

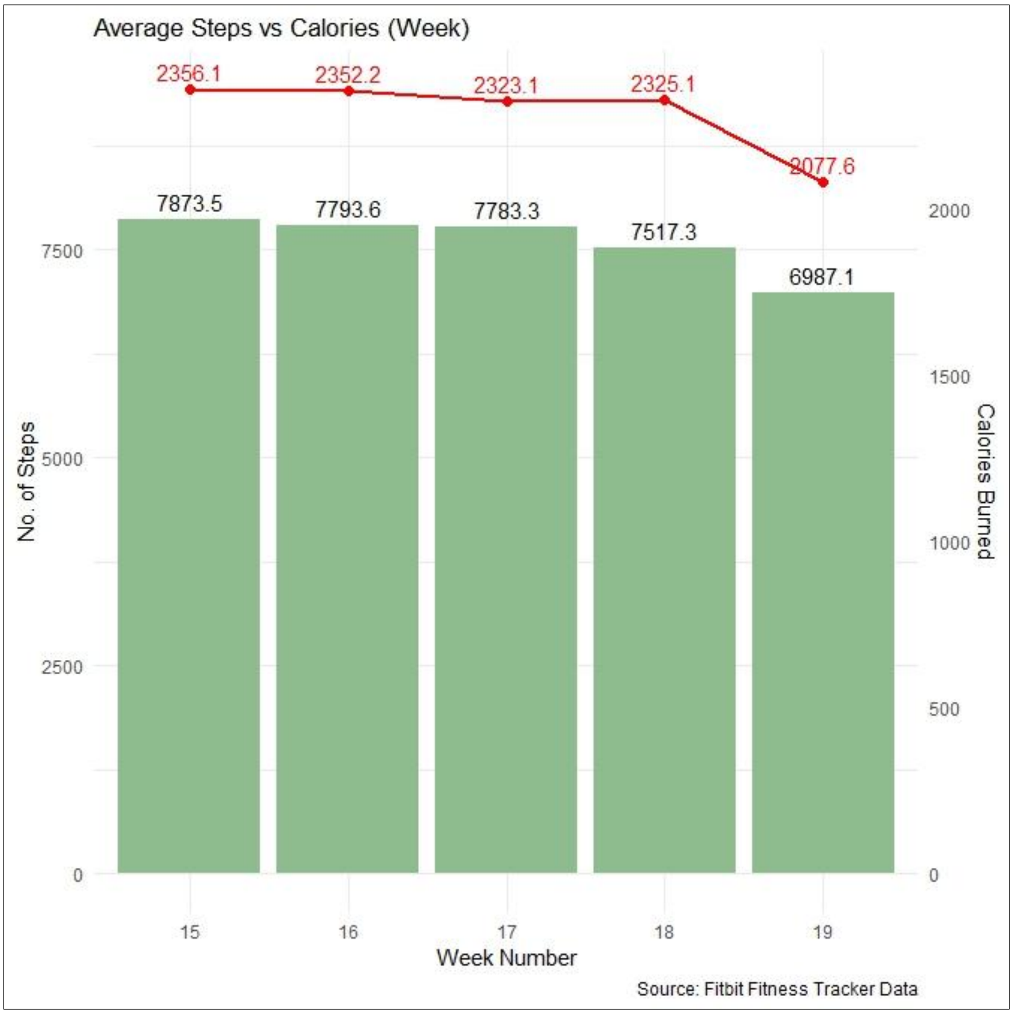
The visualized data, grouped by weekday, reveals that **sedentary time** is highest on Mondays, averaging 1,027 minutes. In contrast, **active time** peaks at 244 minutes on Saturdays, followed by 236 minutes on Fridays, with Sunday being the least active day of the week.



Although the average **Total Active Time** is 227 minutes, the distribution indicates that **Light Active** minutes constitute the largest portion, averaging 192 minutes. This activity peaks on Saturdays at 207 minutes, followed by Fridays at 204 minutes. Additionally, the data shows that the combined average time for **Very Active** and **Fairly Active** minutes is 34 minutes, with distributions of 21 minutes and 13 minutes, respectively.

STEPS & CALORIES WEEK TREND

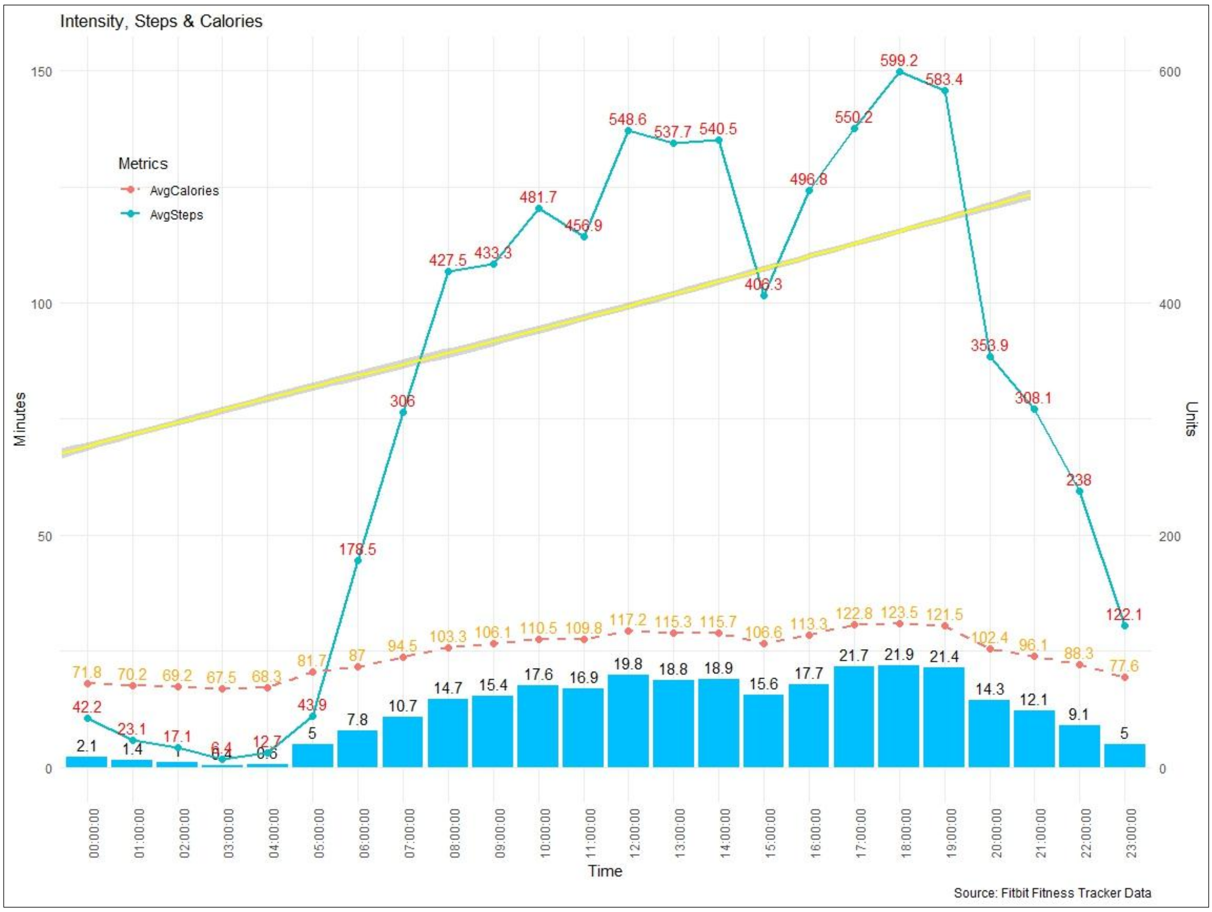
```
AvgPerWeek %>%
  ggplot(aes(x = WeekNo)) +
  geom_bar(aes(y = AvgTotalSteps), stat = "identity", fill = "darkseagreen") +
  geom_text(aes(y = AvgTotalSteps, label = round(AvgTotalSteps, 1)), vjust = -0.5, color = "black") +
  geom_line(aes(y = AvgCaloriesBurned * 4, group = 1), color = "red", size = 1) +
  geom_point(aes(y = AvgCaloriesBurned * 4), color = "red", size = 2) +
  geom_text(aes(y = AvgCaloriesBurned * 4, label = round(AvgCaloriesBurned, 1)), vjust = -0.5, color = "red") +
  theme_minimal() +
  theme(legend.position = c(0.8, 0.3), legend.spacing.y = unit(1, "mm")) +
  labs(
    title = "Average Steps vs Calories (Week)",
    x = "Week Number",
    caption = "Source: Fitbit Fitness Tracker Data") +
  scale_y_continuous(
    name = "No. of Steps",
    sec.axis = sec_axis(~./4, name = "Calories Burned"))
```



The data indicates a **downward trend** in the average **number of steps** taken **over time**. Notably, there is a **significant decline** during **weeks 18 and 19**. This pronounced dip suggests a period of reduced physical activity, which may warrant further investigation to understand the underlying causes.

INTENSITY, STEPS & CALORIES

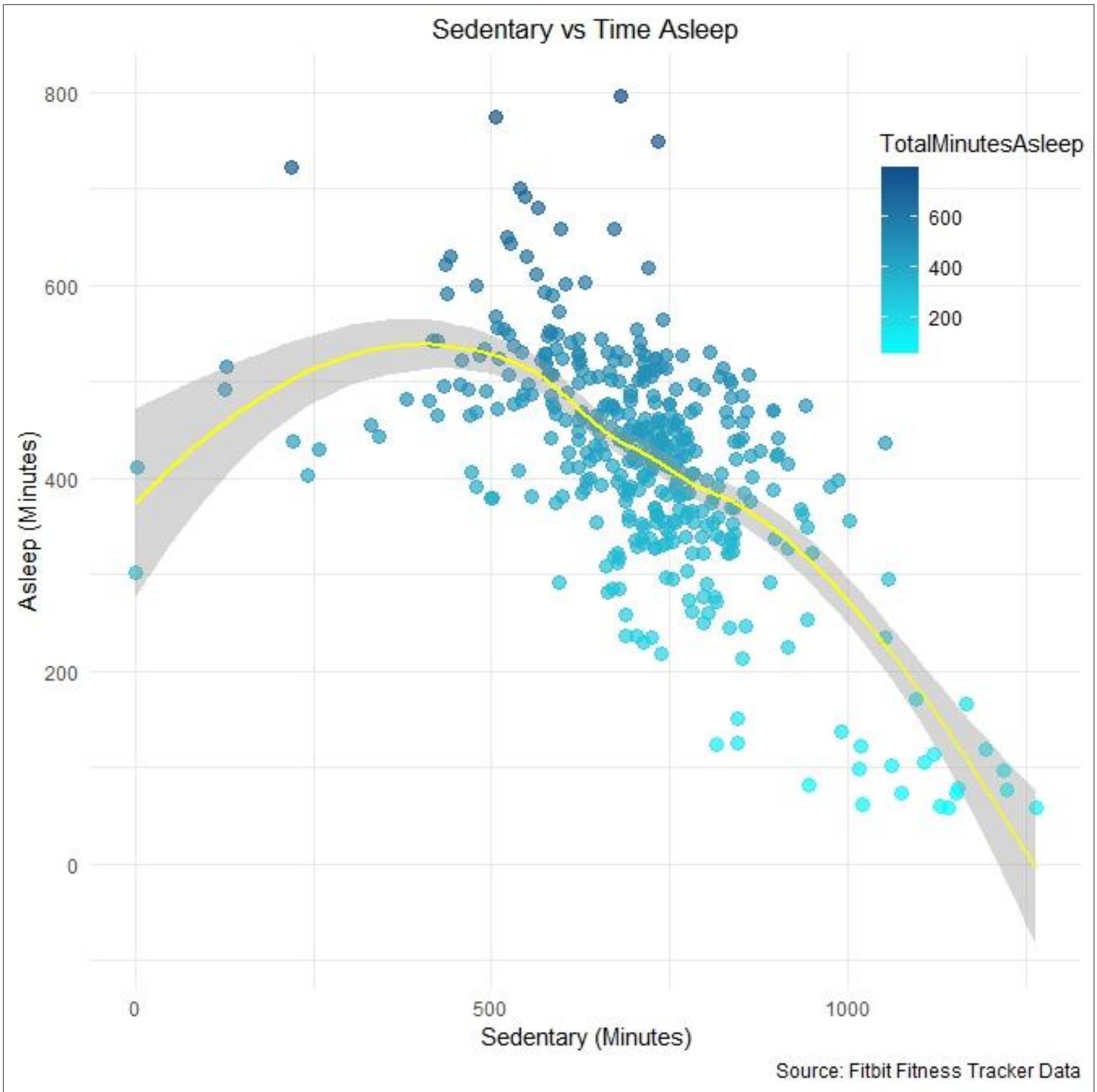
```
AvgPerHour %>%
  ggplot(aes(x = Time)) +
  geom_bar(aes(y = AvgIntensity), stat = "identity", fill = "deepskyblue") +
  geom_text(aes(y = AvgIntensity, label = round(AvgIntensity, 1)), vjust = -0.5, color = "black") +
  geom_line(aes(y = AvgSteps / 4, group = 1, color = "AvgSteps"), size = 1) +
  geom_point(aes(y = AvgSteps / 4, color = "AvgSteps"), size = 2) +
  geom_text(aes(y = AvgSteps / 4, label = round(AvgSteps, 1)), vjust = -0.5, color = "red") +
  geom_line(aes(y = AvgCalories / 4, group = 1, color = "AvgCalories"), linetype = "dashed", size = 1) +
  geom_point(aes(y = AvgCalories / 4, color = "AvgCalories"), size = 2) +
  geom_text(aes(y = AvgCalories / 4, label = round(AvgCalories, 1)), vjust = -0.5, color = "orange") +
  geom_smooth(aes(y = AvgCalories, x= AvgIntensity), method = "loess", formula = y ~ x, color = "yellow", se =
TRUE) +
  theme_minimal() +
  theme(
    legend.position = c(0.1, 0.8),
    axis.text.x = element_text(angle = 90),
    legend.spacing.y = unit(1, "mm")) +
  labs(
    title = "Intensity, Steps & Calories",
    x = "Time",
    y = "Minutes",
    caption = "Source: Fitbit Fitness Tracker Data",
    color = "Metrics") +
  scale_y_continuous(
    sec.axis = sec_axis(~.*4, name = "Units"))
```



The plotted data indicates that participants are **most active** between **17:00 and 19:00**, with an average activity **intensity of 21 minutes**. During this period, participants also take the **greatest number of steps** and **burn the most calories**.

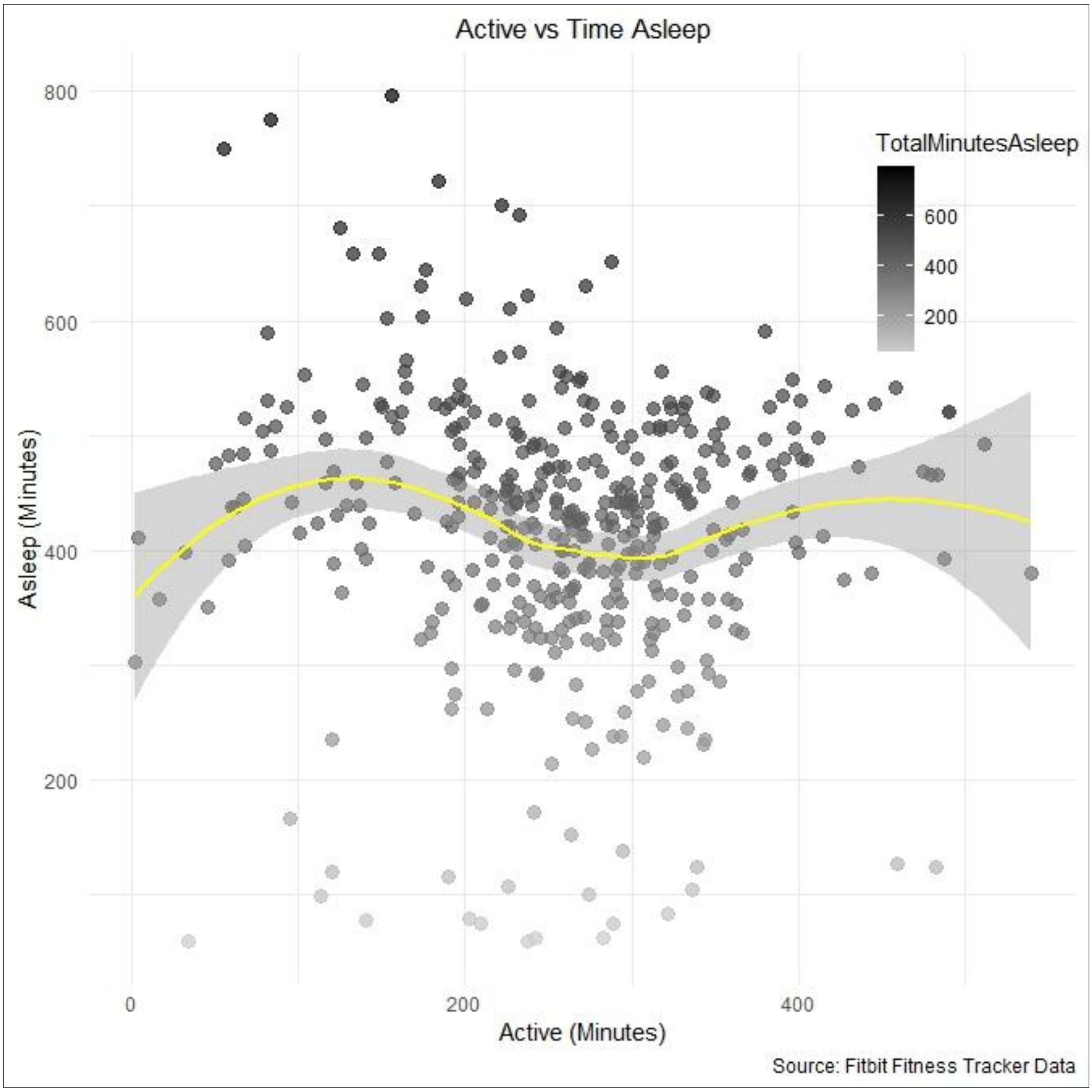
SEDENTARY, ACTIVE & SLEEP TIME

```
sleep_activity_sedentary_merged %>%
  ggplot(aes(x = SedentaryMinutes, y = TotalMinutesAsleep, color = TotalMinutesAsleep)) +
  geom_point(size = 3, alpha = 0.7) +
  geom_smooth(method = "loess", formula = y ~ x, color = "yellow", se = TRUE) +
  scale_color_gradient(low = "cyan", high = "dodgerblue4") +
  theme_minimal() +
  theme(
    legend.position = c(0.9, 0.8),
    legend.spacing.y = unit(1, "mm"),
    plot.title = element_text(hjust = 0.5)) +
  labs(
    title = "Sedentary vs Time Asleep",
    x = "Sedentary (Minutes)",
    y = "Asleep (Minutes)",
    caption = "Source: Fitbit Fitness Tracker Data")
```




```
sleep_activity_sedentary_merged %>%
  ggplot(aes(x = TotalActiveMinutes, y = TotalMinutesAsleep, color = TotalMinutesAsleep)) +
  geom_point(size = 3, alpha = 0.7) +
```

```
  geom_smooth(method = "loess", formula = y ~ x, color = "yellow", se = TRUE) +
  scale_color_gradient(low = "gray80", high = "gray0") +
  theme_minimal() +
  theme(
    legend.position = c(0.9, 0.8),
    legend.spacing.y = unit(1, "mm"),
    plot.title = element_text(hjust = 0.5)) +
  labs(
    title = "Active vs Time Asleep",
    x = "Active (Minutes)",
    y = "Asleep (Minutes)",
    caption = "Source: Fitbit Fitness Tracker Data")
```



The analysis of the two graphs reveals a **negative correlation** between **time asleep** and **sedentary time**. Specifically, as sedentary time increases, the amount of time spent asleep decreases, forming a downward slope. This suggests that higher levels of inactivity are associated with reduced sleep duration.

Conversely, the relationship between **active time** and **sleep duration** shows that the highest average sleep time occurs when active time is around 200 minutes. However, as the average active time approaches 300 minutes, there is a noticeable decline in sleep duration. This indicates that while **moderate levels of activity** are **beneficial** for sleep, **excessive activity** may have a **diminishing return** on sleep quality.

V. Share

In this section of the report, I will present the findings to stakeholders, highlighting any outliers and detailing the methodologies used to create the visualizations and analyze the data. To facilitate a clear understanding of the trends, I recommend incorporating the visuals, such as graphs and charts, into a PowerPoint presentation. This format will enhance the stakeholders' ability to quickly grasp key insights and patterns.

Additionally, I will provide the raw data in CSV format to ensure transparency and allow stakeholders to conduct further analysis if desired. This approach not only promotes a comprehensive understanding of the data but also encourages collaborative evaluation and informed decision-making.

VI. Act, Conclusion & Apply

To promote a fit and healthy lifestyle among Bellabeat users, it is essential to incorporate features that actively engage users in their exercise routines and activities. Bellabeat can draw inspiration from health organizations such as the CDC and WHO, which provide activity guidelines to promote health and wellness. For instance, these organizations recommend at least 150 minutes of moderate exercise per week, or aiming for 10,000 steps a day, and suggest that most healthy adults should get at least seven hours of sleep per night. Based on these guidelines, here are my recommendations:

1. **Creation of Exercise Levels**: Develop exercise programs tailored to different fitness levels, from beginners to advanced users. This could include varying the number of steps or the intensity of activities to suit users who are just starting their fitness journey.

2. **Notification Alerts**: Implement a feature that allows users to set reminders for their exercise routines. Based on the data, Bellabeat can suggest optimal times for exercise, such as 5 PM, to help users prepare for their workout sessions.

3. **Dashboard for Step Count**: Create a dashboard that displays the remaining steps needed to reach the daily goal of 10,000 steps. This visual aid can motivate users to stay active throughout the day.

4. **Hourly Activity Reminders**: For users who are at work or tend to be sedentary, Bellabeat can send hourly notifications to encourage light stretching or brief activities. This can help break up long periods of inactivity and promote overall health.

5. **Promotion of Outdoor and Group Activities**: Encourage users to participate in outdoor activities or group exercises. This can be facilitated through community features within the app, promoting social interaction and motivation.

In conclusion, by integrating these features, Bellabeat can significantly enhance user engagement and promote a healthier lifestyle. The combination of personalized exercise programs, timely reminders, and motivational dashboards can help users achieve their fitness goals more effectively. Additionally, promoting outdoor and group activities can foster a sense of community and support among users, further encouraging a commitment to health and wellness. By aligning with established health guidelines, Bellabeat can ensure that its recommendations are both effective and credible, ultimately leading to better health outcomes for its users.