BellaBeat Case Study

Khalil M Cherif

2023-12-26

About the Company

Urška Sršen and Sando Mur founded Bellabeat, a high-tech company that manufactures health-focused smart products. Sršen used her background as an artist to develop beautifully designed technology that informs and inspires women around the world. Collecting data on activity, sleep, stress, and reproductive health has allowed Bellabeat to empower women with knowledge about their own health and habits.

The company has invested in traditional advertising media, such as radio, out-of-home billboards, print, and television, but focuses on digital marketing extensively. Bellabeat invests year-round in Google Search, maintaining active Facebook and Instagram pages, and consistently engages consumers on Twitter. Additionally, Bellabeat runs video ads on Youtube and display ads on the Google Display Network to support campaigns around key marketing dates.

Sršen knows that an analysis of Bellabeat's available consumer data would reveal more opportunities for growth. She has asked the marketing analytics team to focus on a Bellabeat product and analyze smart device usage data in order to gain insight into how people are already using their smart devices. Then, using this information, she would like high-level recommendations for how these trends can inform Bellabeat marketing strategy.

What we are asked to do?

Sršen asks to analyze smart device usage data in order to gain insight into how consumers use non-Bellabeat smart devices.

- **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?

By the end of this case study, we will produce a report with the following deliverables:

- **1.** A clear summary of the business task
- **2.** A description of all data sources used
- **3.** Documentation of any cleaning or manipulation of data
- **4.** A summary of our analysis
- **5.** Supporting visualizations and key findings
- **6.** Our top high-level content recommendations based on your analysis

What is the problem you are trying to solve?

The problem we are trying to solve in this case study is to gain insights into how consumers are using Bellabeat's smart devices, with a focus on one specific product. The goal is to understand user behavior, preferences, and patterns related to the smart device usage. By doing so, we aim to provide actionable recommendations for Bellabeat's marketing strategy.

Problem Statement:

- Gain insights into how consumers are currently using a specific Bellabeat product and its associated smart device data.
- Understand user engagement, activity, sleep, stress, and any other relevant metrics provided by the smart devices.
- Identify patterns, trends, and potential areas for improvement or innovation.

Business Decisions Driven by Insights:

- **Product Development and Enhancement:** Identify any shortcomings or pain points in the current product usage to improve overall customer satisfaction.
- **Targeted Marketing Campaigns:** Tailor marketing campaigns based on the identified user segments and preferences. Focus advertising efforts on the most engaging features to maximize the impact of marketing initiatives.
- **User Engagement Strategies:** Develop strategies to increase user engagement with the smart devices and the Bellabeat app. Leverage insights to encourage specific behaviors that contribute to improved health and wellness.
- **Digital Marketing Optimization:** Allocate resources more effectively by focusing on channels and platforms that align with user behavior. Tailor digital marketing strategies to resonate with the identified user segments.

By addressing these aspects, the insights gained from the analysis can drive strategic decisions across various facets of Bellabeat's business, ultimately contributing to improved customer satisfaction, increased market share, and sustained growth in the smart device market.

1. Business Task:

The business task for this case study is to analyze smart device data for a specific Bellabeat product, focusing on understanding how consumers are using their smart devices. The goal is to derive insights that can inform Bellabeat's marketing strategy, drive product improvements, and identify growth opportunities.

2. Key Stakeholders:

Urška Sršen (Chief Creative Officer): As a CCO, she is likely concerned with the creative direction and market positioning of Bellabeat products.

Sando Mur (Co-founder): Being a key member of the executive team, he may be interested in the data-driven insights to inform strategic decisions and product development.

Bellabeat Marketing Analytics Team: This team of data analysts is responsible for collecting, analyzing, and reporting data to guide Bellabeat's marketing strategy. They are key stakeholders in the analysis.

3. Statement of the Business Task:

The business task is to conduct a comprehensive analysis of smart device data associated with a specific Bellabeat product. This analysis aims to uncover insights into consumer behavior, preferences, and usage patterns. The ultimate goal is to provide actionable recommendations that will guide Bellabeat's marketing strategy. The analysis should address questions related to user engagement, feature popularity, potential areas for improvement, and opportunities for growth. The results will be presented to key stakeholders, including the executive team, to inform high-level strategies and tactics for the future success of the product and the company.

Preparing the data:

1. Where is the data located?

All data used for this particular case study was downloaded from Kaggle through this link https://www.kaggle.com/datasets/arashnic/fitbit (CC0: Public Domain, dataset made available through **Mobius**): This Kaggle data set contains personal fitness tracker from thirty fitbit users. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. It includes information about daily activity, steps, and heart rate that can be used to explore users' habits.

2. How is the data organized?

The data is organized by categories as follows: * Daily: Activity, Calories, Intensities, Steps, Sleep

- Hourly: Calories, intensities, Steps
- Minutes: Calories: Narrow, Wide
- Intensities: Narrow, Wide
- METs Narrow
- Sleep
- Steps: Narrow, Wide
- Heart Rate
- Weight

3. Are there issues with bias or credibility in this data?

The data is reliable and has been made available by **Möbius** in his Kaggle account and publicly available through this website https://www.kaggle.com/datasets/arashnic/fitbit.

This dataset was generated by respondents to a distributed survey via *Amazon Mechanical Turk* between 03.12.2016-05.12.2016. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. From the Metadata, we can be assured that we're getting accurate, complete and unbiased data that's been vetted and proven fit for use.

4. Does our data ROCCC?

- **Data Reliable:** Medium. Data collected from third-party and a big portion of it imputed directly by the end users.
- **Data Original:** Medium. BellaBeat, inc owns all rights, title, and interest in the Data collected from a third party.
- **Data Comprehensive:** Medium. Data contains all critical information needed to answer the question asked but needs labeling improvments.
- **Data Current:** Low. The data is said to be updated yearly however, the data made available is outdated and collected from 4/12/2016 to 5/12/2016.
- **Data Cited:** High. Data source is well-documented and made available by **Kaggle**.

Processing the Data

1. Tools chosen:

We specifically chose R programming for this particular project for the following reasons:

- **Data Manipulation and Transformation:** R programming handles large datasets like the ones made available for analysis by BellaBeat. R also is a well known tool for statistical computing and data analysis. It excels at data manipulation and transformation, making it well-suited for complex data cleaning, reshaping, and analysis tasks which fits perfectly with the Bellabeat project scope.
- **Data Visualization:** R has powerful data visualization libraries such as ggplot2, which allows for the creation of highly customized and publication-quality plots and charts.
- **Reproducibility and Automation:** R scripts can be easily documented and shared, facilitating reproducibility. For this project, I am using R as a programming language, making it easier to automate data processing pipelines and analyses.

In summary, we chose R programming for its versatility in statistical analysis, data manipulation, and visualization, especially for tasks related to combining data from users metrics, helping us find trends to better understand users behaviors and how they are currently using their smart devices. Then, using the results, we are planning on presenting recommendations to stakeholders to be implemented by the marketing department.

Steps I have taken to ensure that the data is clean

Ensuring that the data is clean involves several steps to address issues such as missing values, outliers, inconsistencies, and other errors.

- **Understand the Data:** We started by familiarizing ourselves with the structure and content of the datasets. We used functions such as colnames, str, head, print, and summary to get deeper information about the data, like the size of the data frame, the column names, the data type, etc. Performing this task, we quickly noticed that all datasets have the 'Id' field in common. This can be used to merge the datasets.

```
install.packages('tidyverse')
library(tidyverse)
daily_activity <- read.csv("dailyActivity_merged.csv")
daily_sleep <- read.csv("sleepDay_merged.csv")
hourly_step <- read.csv("hourlySteps_merged.csv")
hourly_calories <- read.csv("hourlyCalories_merged.csv")
minute_sleep <- read.csv("minuteSleep_merged.csv")
weight_data <- read.csv("weightLogInfo_merged.csv")
```

Steps taken for checking the data 1. Identify all the columns and their names in the datasets

```
colnames(daily_activity)
colnames(daily_sleep)
colnames(hourly_step)
colnames(minute_sleep)
colnames(hourly_calories)
colnames(weight_data)
```

2. Identify the structure of the datasets

```
str(daily_activity)
str(daily_sleep)
str(hourly_calories)
```

```r
str(minute_sleep)
str(hourly_step)
str(weight_data)
```

3. Display the first few rows of the data frames, matrices, and vectors

```r
head(daily_activity)
head(daily_sleep)
head(hourly_calories)
head(minute_sleep)
head(hourly_step)
head(weight_data)
```

We note from the first analysis that all four datasets have the ‘Id’ field in common which can be used to merge the datasets.

4. Understanding some summary statistics

```r
library(dplyr)
daily_activity %>%
  select(ActivityDate, TotalSteps, TotalDistance,SedentaryMinutes, Calories) %>%
  summary()

daily_sleep %>%
  select(TotalSleepRecords,TotalMinutesAsleep,TotalTimeInBed) %>%
  summary()

hourly_calories %>%
  select(ActivityHour, Calories) %>%
  summary()

minute_sleep %>%
  select(date, value) %>%
  summary()

hourly_step %>%
  select(ActivityHour, StepTotal) %>%
  summary()

weight_data %>%
  select(WeightKg, Fat, BMI) %>%
  summary()
```

From the summaries of each data frame we see clearly that there are no missing values for each dataset.

5. Display the unique participants for each dataset

```r
n_distinct(daily_activity$Id)
n_distinct(daily_sleep$Id)
n_distinct(hourly_calories$Id)
n_distinct(hourly_step$Id)
n_distinct(minute_sleep$Id)
n_distinct(weight_data$Id)
```

- **daily_activity** has 33 unique participants
- **daily_sleep** has 24 unique participants

- **hourly_calories** has 33 unique participants
- **hourly_step** has 33 unique participants
- **minute_sleep** has 24 unique participants
- **weight_data** has 8 unique participants

We have to take in mind that all four datasets have different sizes which can cause a problem merging or joining the datasets for further analysis.

6. Display the number of observations in each dataframe

```
nrow(daily_activity)
nrow(daily_sleep)
nrow(hourly_calories)
nrow(hourly_step)
nrow(minute_sleep)
nrow(weight_data)
```

- **daily_activity** has 940 obs of 15 variables
- **daily_sleep** has 413 obs of 5 variables
- **hourly_calories** has 22099 obs of 3 variables
- **hourly_step** has 22099 obs of 3 variables
- **minute_sleep** has 188521 obs of 4 variables
- **weight_data** has 67 obs of 8 variables

7. Check missing or null values in each dataframe

```
# Find out which columns with missing data if any
missing_values_DA <- sapply(daily_activity, function(x) any(is.na(x))) #daily_activity
missing_values_DS <- sapply(daily_sleep, function(x) any(is.na(x))) #daily_sleep
missing_values_HC <- sapply(hourly_calories, function(x) any(is.na(x))) #hourly_calories
missing_values_HSt <- sapply(hourly_step, function(x) any(is.na(x))) #hourly_step
missing_values_MS <- sapply(minute_sleep, function(x) any(is.na(x))) #minute_sleep
missing_values_weight <- sapply(daily_activity, function(x) any(is.na(x))) #weight
```

*Analyse missing values in the data frame Weight which turns out to be data related to Fat columns with 65 missing values

```
# Print the columns with missing values
missing_values_weight <- sapply(daily_activity, function(x) any(is.na(x))) # find out which columns wit
missing_data_count7 <- sapply(weight_data, function(x) sum(is.na(x))) # count rows with missing data
print(weight_data[is.na(weight_data$Fat) | !is.finite(weight_data$Fat),]) # Print rows with missing dat
```

9. Check for duplicates in the dataframes

```
minute_sleep <- minute_sleep[, !names(minute_sleep) %in% "logId"]
# Check for duplicates in the entire data frame
any_duplicates_DA <- any(duplicated(daily_activity))
any_duplicates_DS <- any(duplicated(daily_sleep))
any_duplicates_HC <- any(duplicated(hourly_calories))
any_duplicates_HSt <- any(duplicated(hourly_step))
any_duplicates_MS <- any(duplicated(minute_sleep))
any_duplicates_W <- any(duplicated(weight_data))

# Print the result
```

```r
if (any_duplicates_DA) {
  cat("There are duplicates in the data frame.\n")
} else {
  cat("There are no duplicates in the data frame.\n")
}

if (any_duplicates_DS) {
  cat("There are duplicates in the data frame.\n")
} else {
  cat("There are no duplicates in the data frame.\n")
}

if (any_duplicates_MS) {
  cat("There are duplicates in the data frame.\n")
} else {
  cat("There are no duplicates in the data frame.\n")
}
```

```r
# Identify and display the duplicated rows in the entire data frame
duplicates_DS <- daily_sleep[duplicated(daily_sleep), ]

# Print the duplicated rows
if (nrow(duplicates_DS) > 0) {
  cat("Duplicated rows:\n")
  print(duplicates_DS)
} else {
  cat("There are no duplicated rows in the data frame.\n")
}

duplicates_MS <- minute_sleep[duplicated(minute_sleep), ]

# Print the duplicated rows
if (nrow(duplicates_MS) > 0) {
  cat("Duplicated rows:\n")
  print(duplicates_MS)
} else {
  cat("There are no duplicated rows in the data frame.\n")
}
```

8. Transforming the data

We noticed the Date format is different for most of the datasets. Let's make them all similar for future analysis.

```r
library(lubridate)
# Convert 'ActivityDate' to a DateTime object
daily_activity <- read_csv("dailyActivity_merged.csv")
# Create new columns 'Date' and 'Time'
daily_activity <- mutate(daily_activity, ActivityDate = mdy_hms(paste(ActivityDate, "00:00:00")))
daily_activity$Date <- as.Date(daily_activity$ActivityDate)
daily_activity$Time <- format(daily_activity$ActivityDate, "%H:%M:%S")
# Remove the original 'ActivityDate' column
daily_activity$ActivityDate <- NULL
daily_activity$Activity_Date <- NULL
```

```r
# Convert 'SleepDay' to a DateTime object
daily_sleep$SleepDay <- mdy_hms(daily_sleep$SleepDay)
# Create new columns 'Date' and 'Time'
daily_sleep$Date <- as.Date(daily_sleep$SleepDay)
daily_sleep$Time <- format(daily_sleep$SleepDay, "%H:%M:%S")
# Remove the original 'ActivityDate' column
daily_sleep$SleepDay <- NULL

# Convert 'ActivityHour' in hourly_calories to a DateTime object
hourly_calories$ActivityHour <- mdy_hms(hourly_calories$ActivityHour)
hourly_calories$Date <- as.Date(hourly_calories$ActivityHour)
hourly_calories$Time <- format(hourly_calories$ActivityHour, "%H:%M:%S")
hourly_calories$ActivityHour <- NULL

# Convert 'ActivityHour' in hourly_step to a DateTime object
hourly_step$ActivityHour <- mdy_hms(hourly_step$ActivityHour)
hourly_step$Date <- as.Date(hourly_step$ActivityHour)
hourly_step$Time <- format(hourly_step$ActivityHour, "%H:%M:%S")
hourly_step$ActivityHour <- NULL

# Convert 'date' in minute_sleep to a DateTime object
minute_sleep$date <- mdy_hms(minute_sleep$date)
minute_sleep$Date <- as.Date(minute_sleep$date)
minute_sleep$Time <- format(minute_sleep$date, "%H:%M:%S")
minute_sleep$date <- NULL

# Convert 'Date' in weight_data to a DateTime object
weight_data$Date2 <- weight_data$Date
weight_data$Date <- NULL
weight_data$Date2 <- mdy_hms(weight_data$Date2)
weight_data$Date <- as.Date(weight_data$Date2)
weight_data$Time <- format(weight_data$Date2, "%H:%M:%S")
weight_data$Date2 <- NULL
```

verifying all Dates and Times are in the same format

```r
head(daily_activity$Date)
head(daily_activity$Time)

head(daily_sleep$Date)
head(daily_sleep$Time)

head(hourly_calories$Date)
head(hourly_calories$Time)

head(hourly_step$Date)
head(hourly_step$Time)

head(minute_sleep$Date)
head(minute_sleep$Time)

head(weight_data$Date)
head(weight_data$Time)
```

Analyzing users Behavior

Most active time of the day

We merged hourly_steps and hourly_calories to find out when during the day users are most active

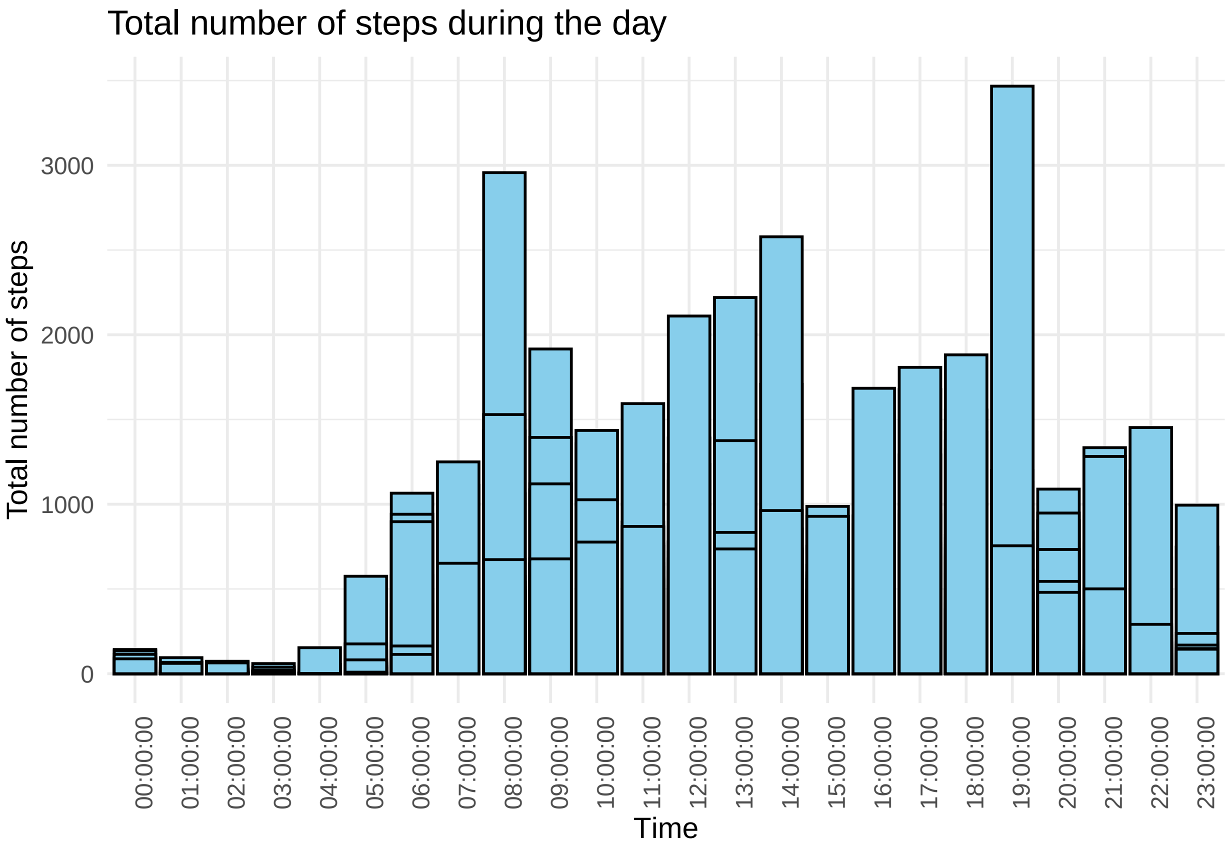
```
steps_calories <- merge(hourly_step, hourly_calories, by = c("Id", "Date", "Time"))
steps_calories_grouped <- steps_calories %>%
  group_by(Time) %>%
  summarise(StepTotal = sum(StepTotal), CalorieTotal = sum(Calories))

steps_calories_grouped2 <- steps_calories %>%
  group_by(Id, Time) %>%
  summarise(StepTotal = mean(StepTotal))
```

Let's Plot the results of the number of steps during the day

```
library(ggplot2)

ggplot(steps_calories_grouped2, aes(x = Time, y = StepTotal)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black", position = "dodge") +
  labs(title = "Total number of steps during the day",
       x = "Time",
       y = "Total number of steps",
       fill = "Id") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90))
```



Behavior 1:

From the analysis of the average number of steps users take during the day, we can observe a peak at 8:00AM, another one at 2:00PM, but the highest average occurs at 7:00PM.

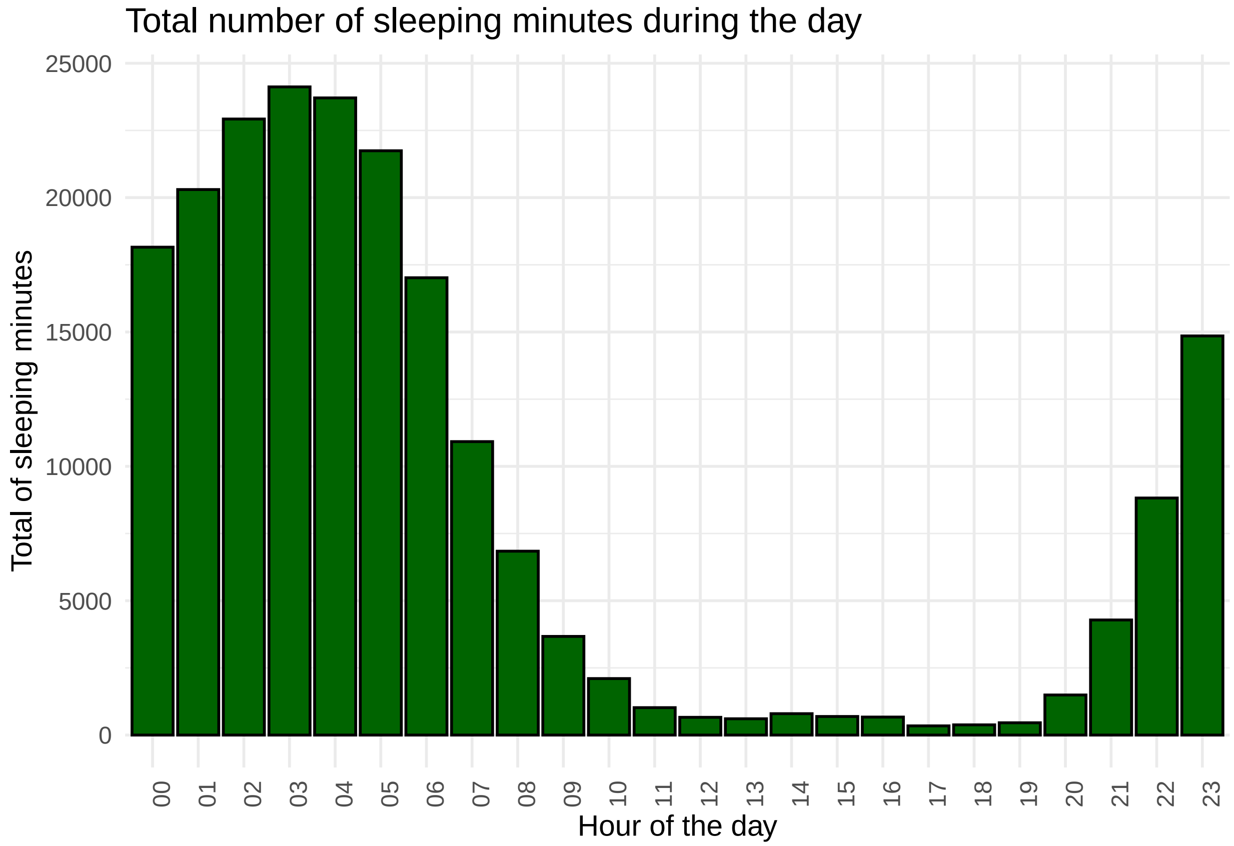
This could be due to the fact that users commute or walk to work between the hours of 8 to 9am in the morning and 6 to 7pm in the evening. Moreover, users walk more or are more active after working hours. Doing errands, going to the gym, going out shopping or for dinner. Lunch time is also a time where number of steps are up.

- Our users are mostly working people who are active going to and leaving work, also during lunch hours.

Let's check the sleeping habits of users

```r
library(dplyr)
grouped_min_sleep <- minute_sleep %>%
  mutate(Hour = format(as.POSIXct(Time, format = "%H:%M:%S"), format = "%H")) %>%
  group_by(Hour) %>%
  summarise(TotalValue = sum(value), Count = n())

ggplot(grouped_min_sleep, aes(x = Hour, y = TotalValue)) +
  geom_bar(stat = "identity", fill = "darkgreen", color = "black", position = "dodge") +
  labs(title = "Total number of sleeping minutes during the day",
       x = "Hour of the day",
       y = "Total of sleeping minutes") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90))
```



Behavior #2:

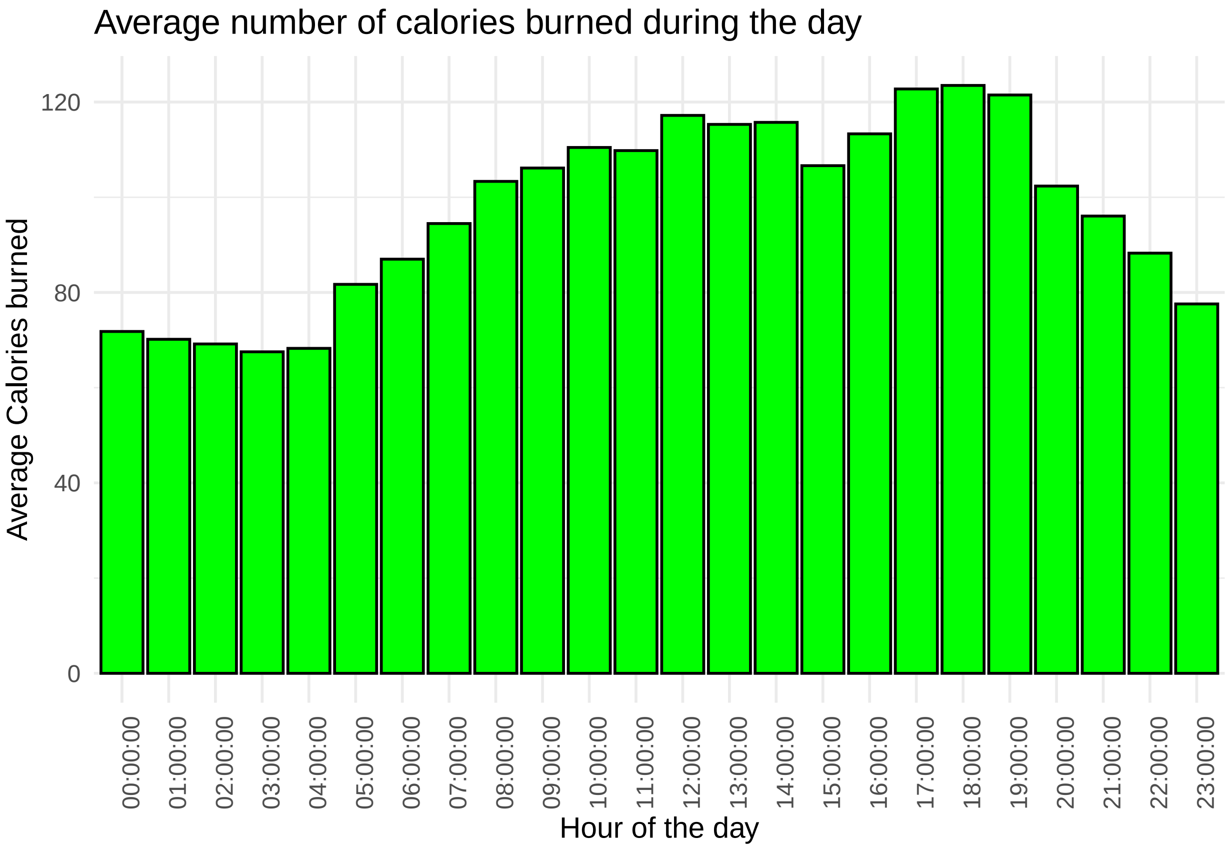
From the analysis of the total sleeping minutes of users during the day, we can observe the lowest are between 8:00AM and 10:00PM, with highest minutes asleep are between the hours of 11:00PM and 7:00AM. This goes confirms our previous conclusion that our users follow the trend and behavior of working adults who go to bed at 11:00PM and wake up at 7:00AM.

- Our users are mostly working people who are awake from 7:00AM to 11:00PM. They mostly go to bed at around 11:00PM

Average calories burned during the day

```r
library(dplyr)
grouped_Cal <- hourly_calories %>%
  group_by(Time) %>%
  summarise(Average_Cal = mean(Calories))

ggplot(grouped_Cal, aes(x = Time, y = Average_Cal)) +
  geom_bar(stat = "identity", fill = "green", color = "black", position = "dodge") +
  labs(title = "Average number of calories burned during the day",
       x = "Hour of the day",
       y = "Average Calories burned") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90))
```



Most active day of the week

We created a new column in the daily_activity dataset called "DayOfWeek". We then did the following to analyze the average active minutes for users during the week. Note that active minutes are the sum of VeryActiveMinutes, FairlyActiveMinutes, and LightlyActiveMinutes.

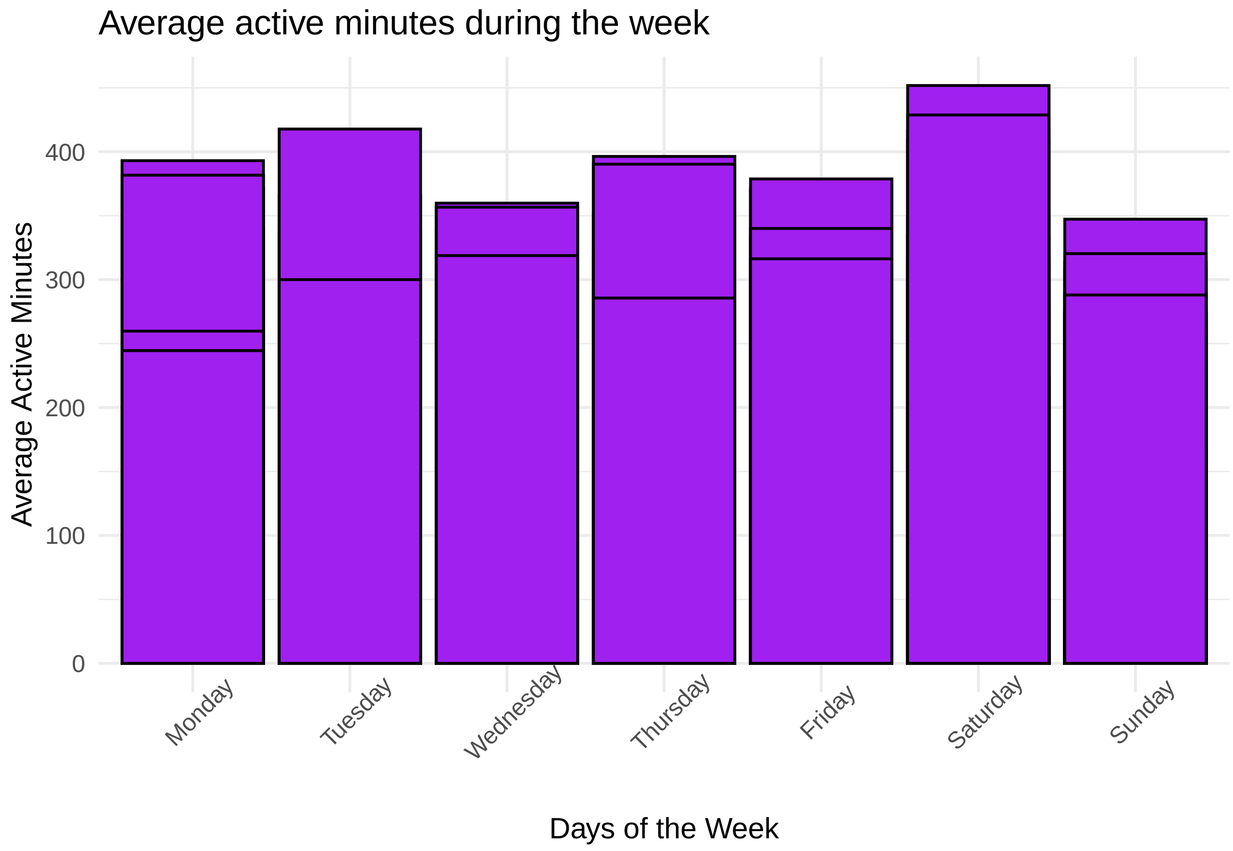
```r
daily_activity$DayOfWeek <- weekdays(as.Date(daily_activity$Date))
daily_activity$ActiveMinutes <- rowSums(daily_activity[, c("VeryActiveMinutes", "FairlyActiveMinutes",
```

We now plot the total activity minutes during the days of the week

```
grouped_ActiveMin <- daily_activity %>%
  group_by(Id, DayOfWeek) %>%
  summarise(Avg_ActiveMin = mean(ActiveMinutes))

grouped_ActiveMin$DayOfWeek <- factor(grouped_ActiveMin$DayOfWeek, levels = c("Monday", "Tuesday", "Wedn

ggplot(grouped_ActiveMin, aes(x = DayOfWeek, y = Avg_ActiveMin)) +
 geom_bar(stat = "identity", fill = "purple", color = "black", position = "dodge") +
  labs(title = "Average active minutes during the week",
       x = "Days of the Week",
       y = "Average Active Minutes") +
       #fill = "Id") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45))
```



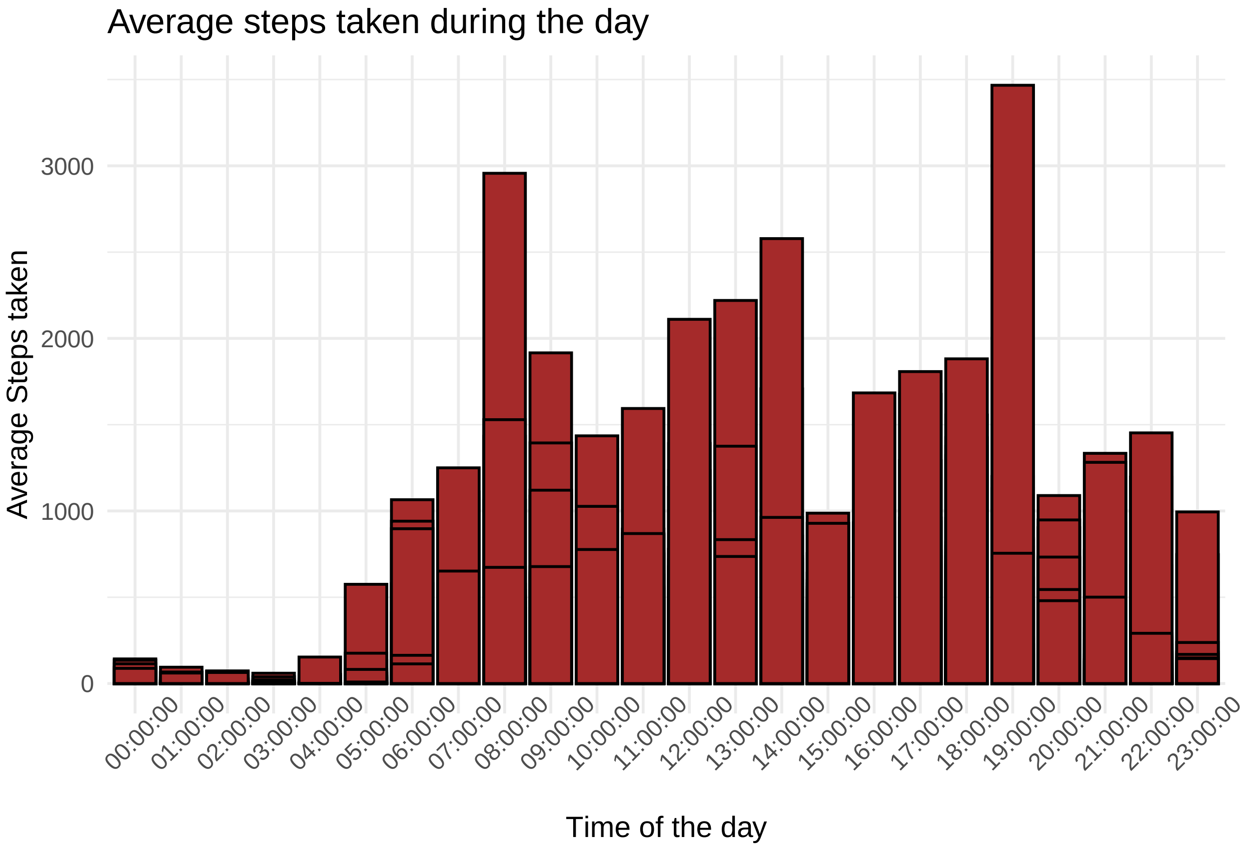
Behavior #3

We can conclude from the bar graph that users are more active during Saturdays. So they are considered weekend warriors as it is expected of most people in general.

Most steps taken during the day

```
grouped_steps <- hourly_step %>%
  group_by(Id, Time) %>%
  summarise(Avg_steps = mean(StepTotal))

ggplot(grouped_steps, aes(x = Time, y = Avg_steps)) +
 geom_bar(stat = "identity", fill = "brown", color = "black", position = "dodge") +
  labs(title = "Average steps taken during the day",
       x = "Time of the day",
       y = "Average Steps taken") +
       #fill = "Id") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45))
```



Behavior #4

From the bar graph, we see that on the average most steps taken by users are at the hours of 7:00AM, 1:00PM and, 6:00PM. Again this strengthen our assumption that users are working adults whom, a good proportion of them, prefer to walk to work and go out walking during lunch breaks. One should also not omit that users might exercise as well during those peak hours.

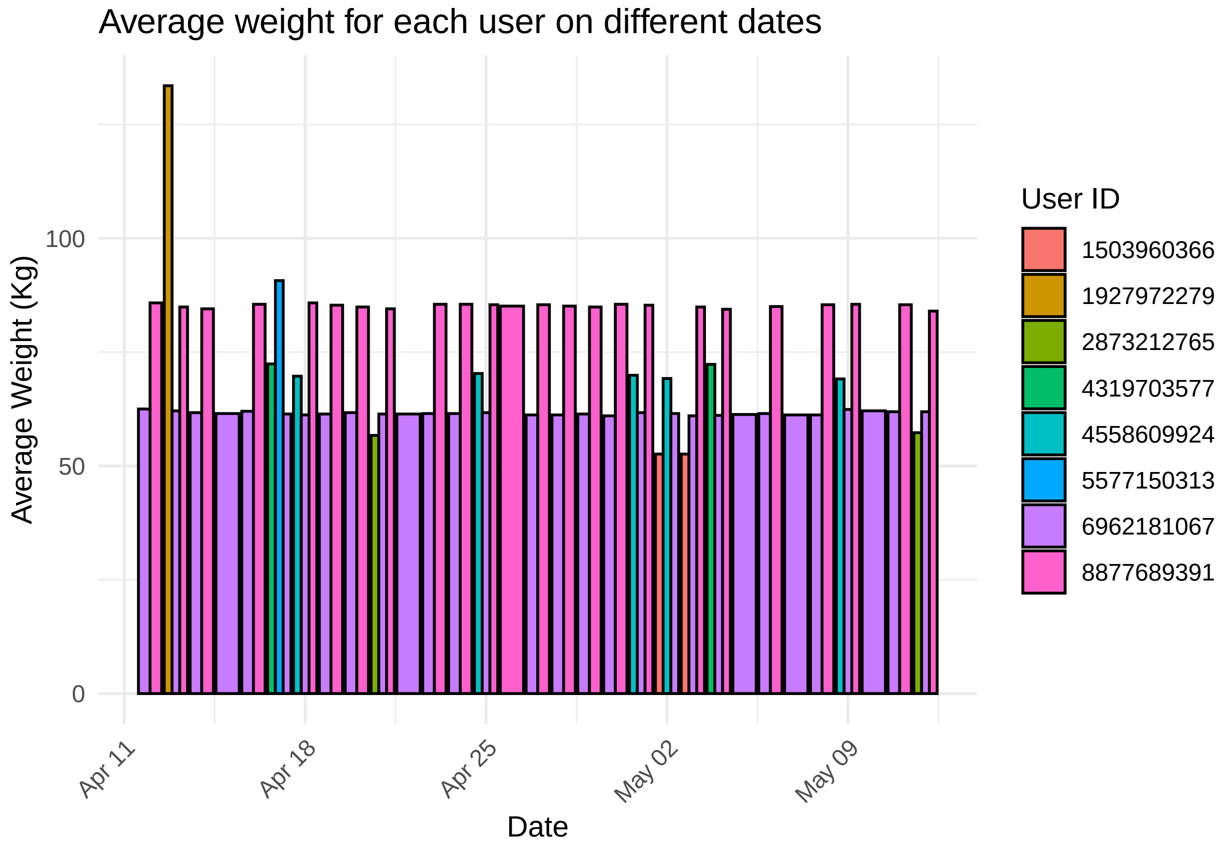
Weight loss journey

To analyze this users behavior and understand if they are in a journey to losing weight, we decided to analyze the weight changes of users during the available data period which is from the 03.12.2016 to 05.12.2016.

```
library(ggplot2)
library(dplyr)

grouped_weight <- weight_data %>%
  group_by(Id, Date) %>%
  summarise(AvgWeightKg = mean(WeightKg))

# Plotting
ggplot(grouped_weight, aes(x = Date, y = AvgWeightKg, fill = as.factor(Id))) +
  geom_bar(stat = "identity", position = "dodge", color = "black") +
  labs(title = "Average weight for each user on different dates",
       x = "Date",
       y = "Average Weight (Kg)",
       fill = "User ID") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



From the graph we see that the selected users have stable weight and there are no signs of weight loss behavior. However, the data is limited and historical data, especially for weight changes, is necessary in order to better understand users weight behavior. Thus more data is needed to be collected to draw a better conclusion. This analysis is thus inconclusive.

Analyzing users Preferences

Logs entry preference

Let's study manual or automatic logs for engaged users.

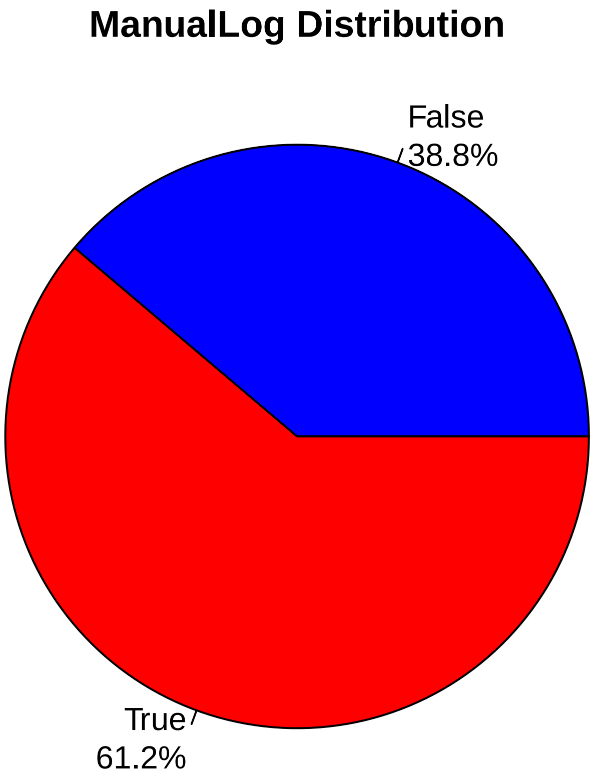
```
library(scales)
manual_log_counts <- table(weight_data$IsManualReport)
percentages <- manual_log_counts / sum(manual_log_counts) * 100
print(manual_log_counts)
```

```
##
## False  True
##    26    41
```

```
labels <- paste0(names(manual_log_counts), "\n", sprintf("%.1f%%", percentages))
```

```r
# Set the overall size of the pie chart
par(mar = c(1, 1, 1, 1))

# Plot the pie chart with labels inside
pie(manual_log_counts, labels = labels, col = c("blue", "red"), main = "ManualLog Distribution", cex = 
```



Preference #1:

We can see from this pie chart that 61% of engaged users do enter the data about their weight manually while 39% don't. It is interesting to note that users prefer to enter their data and are conscious about taking full advantage of their smart devices. However this portion of engaged users is very insignificant because from the 33 unique users we are studying only 8 entered the weight logs. From those 8 only 5 have entered the logs more than 2 times during the whole month. To get conclusive analysis users need to enter their data on regular basis thus more work is needed to raise users awarenesss regarding this particular feature,

Users prefer to wear their devices continuously?

In order to understand this users preference, we decided to analyze the non activity periods registered by the smart devices or what it is here called "Sedentary Minutes". While sedentary means not moving, if we omit the minutes users are asleep the rest of the minutes per day should be should be active minutes, very, fairly, or lightly active. So the total minutes per day or 1440 minutes minus the total sleeping minutes minus the total active minutes should give us the total of sedentary minutes. If we compare those values to the sedentary minutes the devices are recording we should have a better understanding if the users are actually not moving or not wearing their devices at all.

```r
#compare non activity minutes to sedentary minutes
library(dplyr)
combined_activity_sleep <- merge(daily_sleep, daily_activity, by = c("Id", "Date"), all = TRUE)

combined_activity_sleep <- combined_activity_sleep %>%
  mutate(Difference = 1440 - ActiveMinutes - TotalTimeInBed)
```

```
# Remove rows where TotalMinutesAsleep is NA or ActiveMinutes is 0
combined_activity_sleep <- combined_activity_sleep %>%
  filter(!is.na(TotalTimeInBed) & ActiveMinutes != 0)

# Check if the calculated difference matches SedentaryMinutes
combined_activity_sleep <- combined_activity_sleep %>%
  mutate(Diff_hour = round(as.numeric(SedentaryMinutes)/60 - as.numeric(Difference)/60 , 1))

combined_activity_sleep$Match <- combined_activity_sleep$Difference == combined_activity_sleep$Sedentar

# Display the relevant columns
selected_columns <- c("Id", "Date", "ActiveMinutes", "TotalTimeInBed", "SedentaryMinutes", "Difference"
result <- combined_activity_sleep[, selected_columns]
```

Let's Plot the result and see the proportion of users who wear their devices compare to the ones who don't wear it continuously.

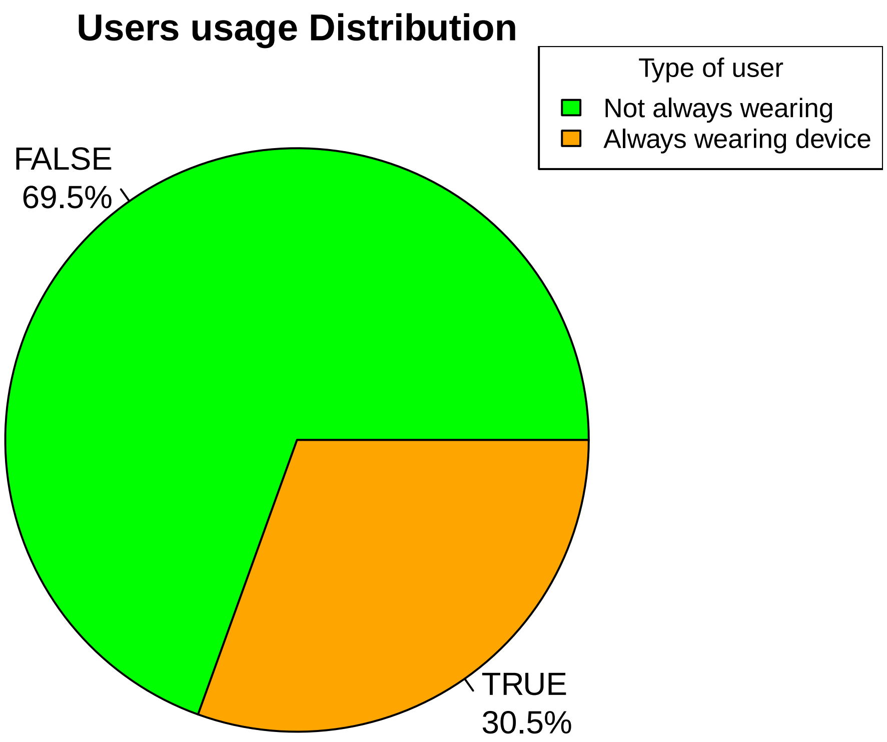
```
library(scales)
SedUsers_counts <- table(result$Match)

# Calculate percentages manually
percentages <- prop.table(SedUsers_counts) * 100

# Create labels with percentages
labels <- paste0(names(SedUsers_counts), "\n", sprintf("%.1f%%", percentages))

par(mar = c(1, 1, 1, 1))
# Plot the pie chart
pie(SedUsers_counts, labels = labels, col = c("green", "orange"), main = "Users usage Distribution", ce

# Add legend
legend("topright", legend = c("Not always wearing", "Always wearing device"), fill = c("green", "orange
```



Preference #2:

Users don't prefer to wear their devices at all the time, actually 70% of them don't actually wear their devices continuously.

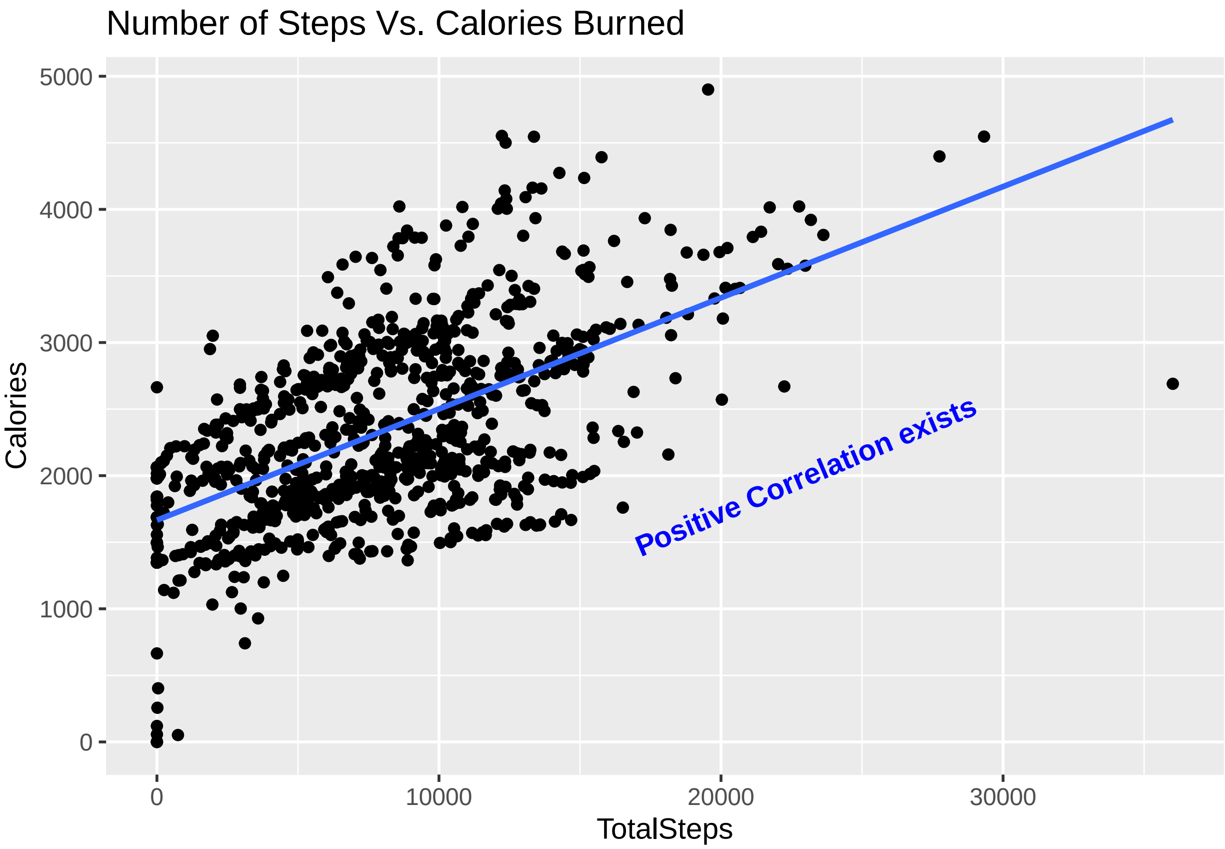
Trends & Patterns:

During this section we will explore common trends and patterns in our current users.

Calorie crusher?

What's the relationship between steps taken in a day and calories? Do more steps means more calories burned? For this analysis we will use the daily activity data set as it contains data about the number of steps taken daily by users and the number of calories burned as well.

```
library(tidyverse)
library(ggplot2)
ggplot(data = daily_activity,
       aes(x = TotalSteps, y = Calories)) +
  geom_point() + geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Number of Steps Vs. Calories Burned") +
  annotate("text", x=23000, y=2000, label="Positive Correlation exists", color="blue", fontface="bold",
```



Trend #1:

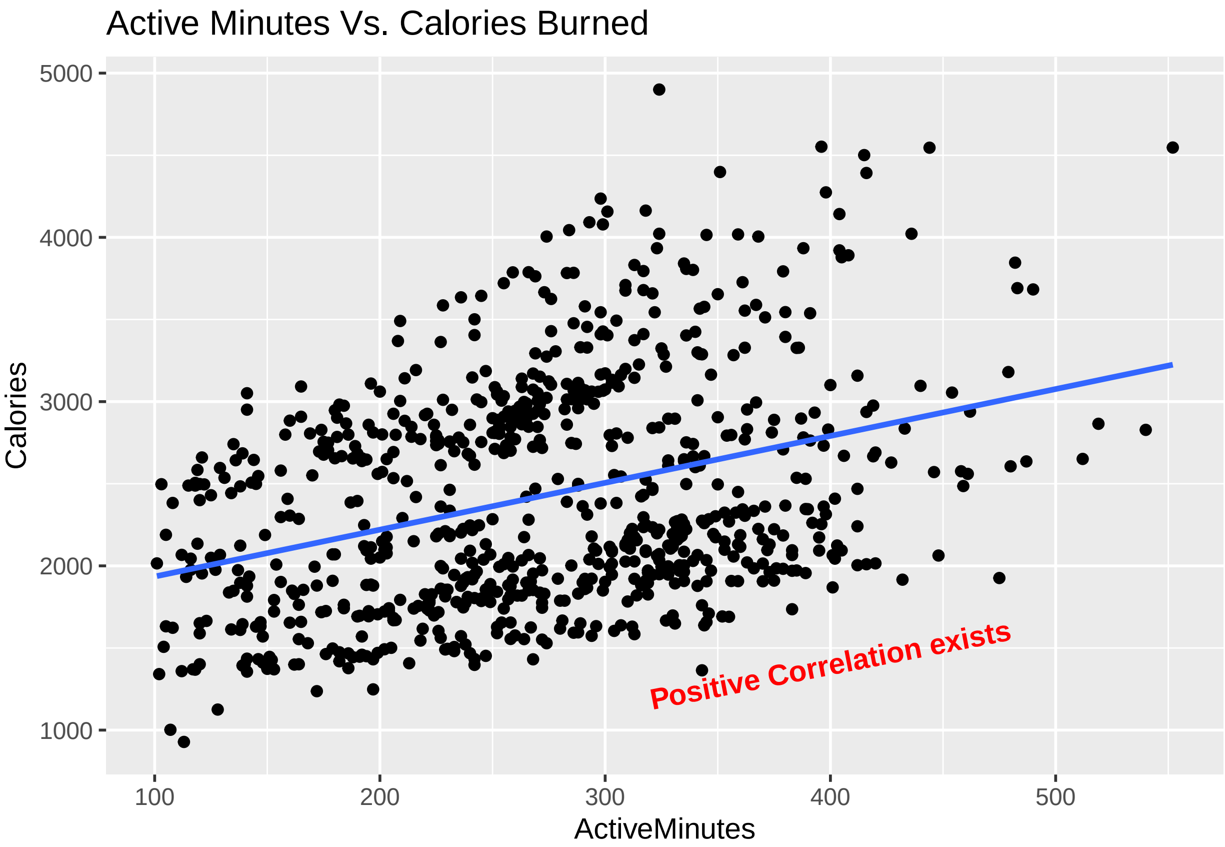
From the scattered plot we can quickly see a positive correlation between the number of steps taken and the number of calories burned. We can conclude that the more steps users take, the more calories they burn. The opposite is correct as well.

More active minutes means more calories?

Let's find out if there is a correlation between active minutes and number of calories burned.

```
#ActiveMinutes was previously created and it is the sum of very active, fairly active, and lightly acti

ggplot(data = daily_activity[daily_activity$ActiveMinutes >= 100 &
                             daily_activity$Calories >= 100,] ,
       aes(x = ActiveMinutes, y = Calories)) +
  geom_point() + geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Active Minutes Vs. Calories Burned") +
  annotate("text", x=400, y=1400, label="Positive Correlation exists", color="red", fontface="bold", si
```



Trend #2:

In fact there is a positive correlation, but not a strong one, between the number of active minutes and the number of calories burned. So the more users spend minutes actively the more calories they burn.

High heart rate intervals mean more calories?

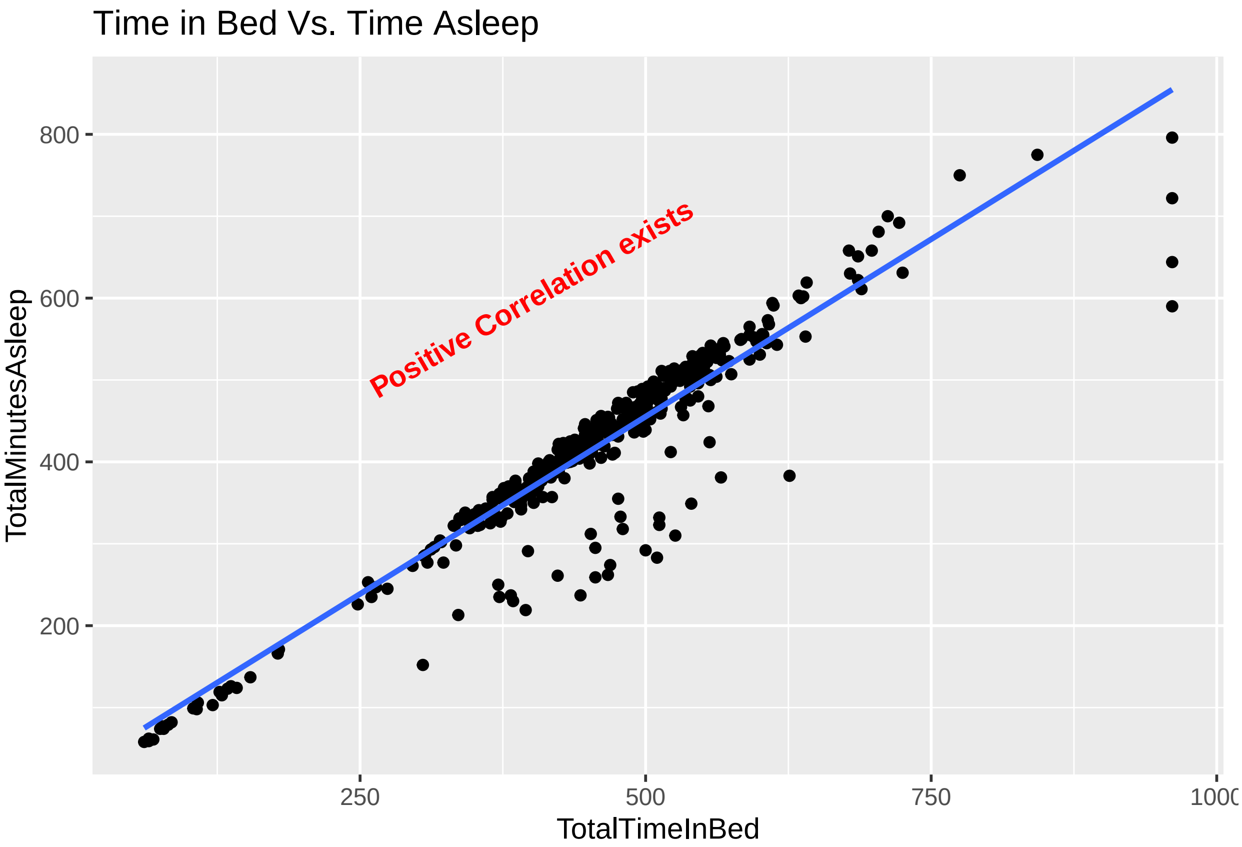
*There is no correlation between the heart rate and number of calories burned.

Sleeping Pattern

More time in bed means more sleeping hours?

Let's see if there is a correlation between time in bed and sleeping time.

```
library(ggplot2)
ggplot(data=daily_sleep, aes(x=TotalTimeInBed, y=TotalMinutesAsleep)) + geom_point() + geom_smooth(meth
  labs(title = "Time in Bed Vs. Time Asleep") +
  annotate("text", x=400, y=600, label="Positive Correlation exists", color="red", fontface="bold", siz
```



Trend #2:

We can rapidly conclude from this strong correlation between Time in Bed and Time Asleep, that the more time users stay in bed the more they sleep. There are few exceptions with some users who stay more time in bed with less time asleep. They could be reading, using their phones or doing other things in bed rather than sleeping.

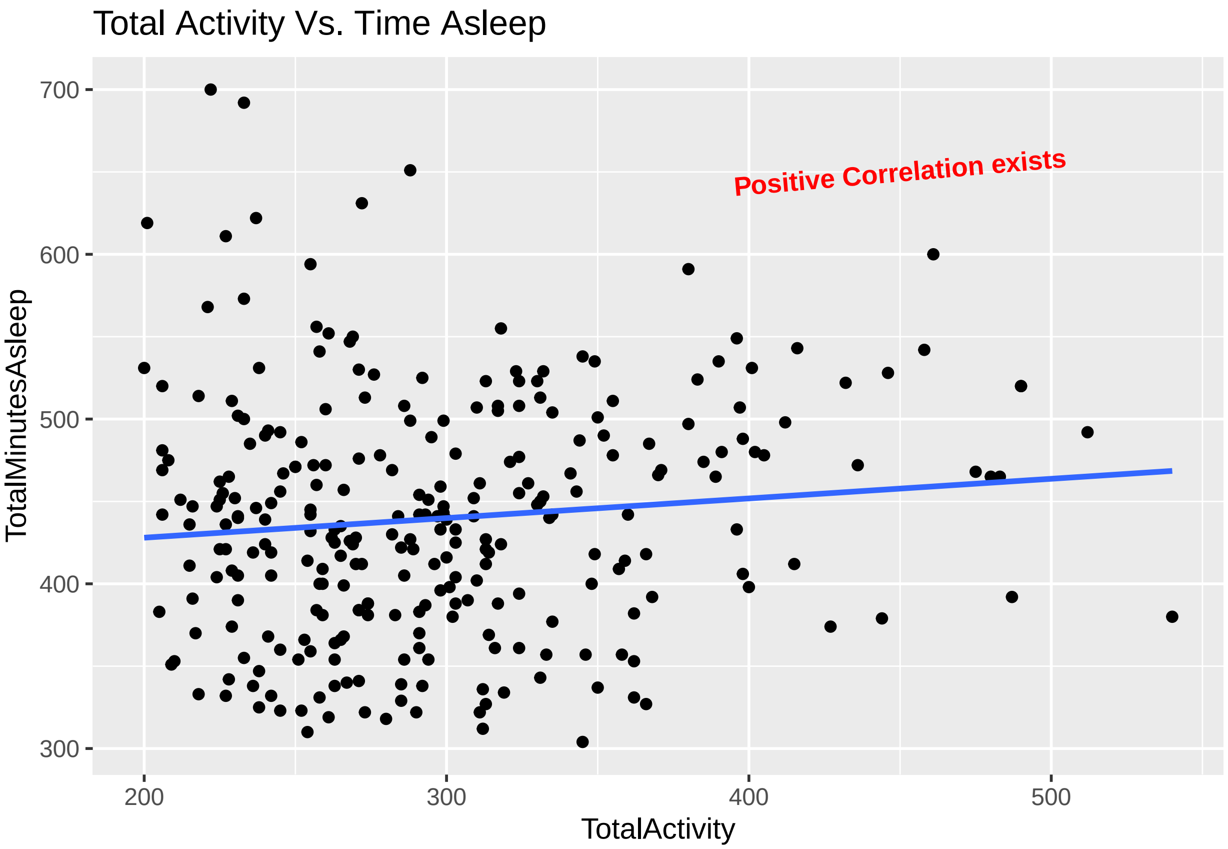
More active minutes means more sleeping?

Let's find out if users who spend more time actively sleep more. For this analysis we will merge both daily activity and daily sleep databases.

```
library(tidyverse)
library(dplyr)
combined_data <- merge(daily_sleep, daily_activity, by= c("Id", "Date"))
combined_data <- combined_data %>%
  mutate(TotalActivity = VeryActiveMinutes + LightlyActiveMinutes +
           FairlyActiveMinutes)

result <- combined_data %>%
  group_by(Id, Date) %>%
  filter(TotalActivity >= 200 & TotalMinutesAsleep >= 300) %>%
  summarize(Id, TotalActivity = mean(TotalActivity, na.rm = TRUE),
    TotalMinutesAsleep = mean(TotalMinutesAsleep, na.rm = TRUE), Date
  )

  ggplot(data=result, aes(x=TotalActivity, y=TotalMinutesAsleep)) + geom_point() + geom_smooth(method =
    labs(title = "Total Activity Vs. Time Asleep") +
  annotate("text", x=450, y=650, label="Positive Correlation exists", color="red", fontface="bold", siz
```



Trend #3:

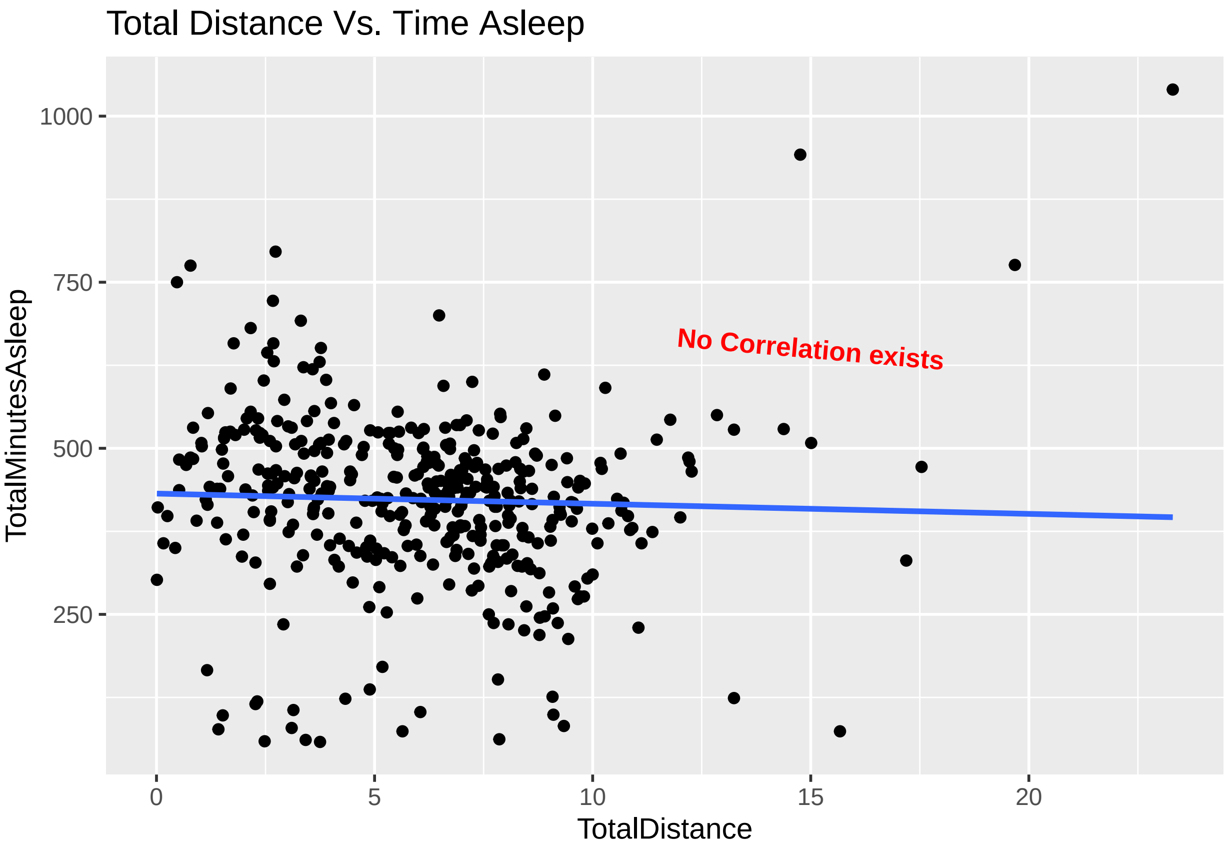
This analysis is not conclusive even though there is a correlation present. So the number of active minutes a day is not necessarily connected with the number of sleeping minutes.

More distance traveled means more sleeping?

Let's find out if users who walk longer distances a day sleep for longer time.

```
result2 <- combined_data %>%
  group_by(Id, Date) %>%
  summarize(
    TotalDistance = sum(TotalDistance, na.rm = TRUE),
    TotalMinutesAsleep = sum(TotalMinutesAsleep, na.rm = TRUE)
  )

  ggplot(data=result2, aes(x=TotalDistance, y=TotalMinutesAsleep)) + geom_point() + geom_smooth(method =
    labs(title = "Total Distance Vs. Time Asleep") +
  annotate("text", x=15, y=650, label="No Correlation exists", color="red", fontface="bold", size=3.5, a
```



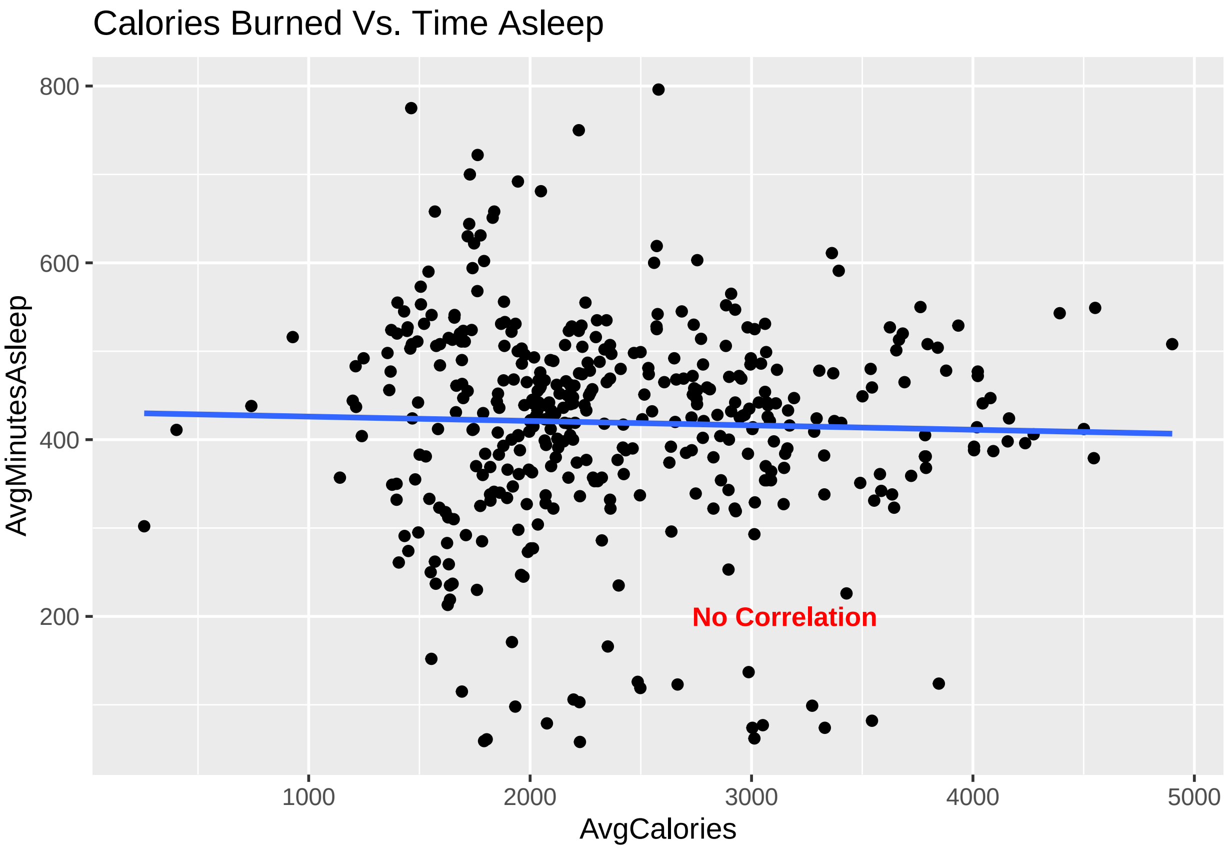
Trend #4:

Similarly we cannot conclude that the longer users travel a day the more minutes they sleep.

More calories burned mean more sleeping?

```
result3 <- combined_data %>%
  group_by(Id, Date) %>%
  summarize(
    AvgCalories = mean(Calories, na.rm = TRUE),
    AvgMinutesAsleep = mean(TotalMinutesAsleep, na.rm = TRUE))

  ggplot(data=result3, aes(x=AvgCalories, y=AvgMinutesAsleep)) + geom_point() + geom_smooth(method = "l
    labs(title = "Calories Burned Vs. Time Asleep") +
  annotate("text", x=3150, y=200, label="No Correlation", color="red", fontface="bold", size=3.5, angle=
```



Trend #5:

Similarly we cannot conclude that the more users burn calories a day the more minutes they sleep.

##Healthy lifestyle?

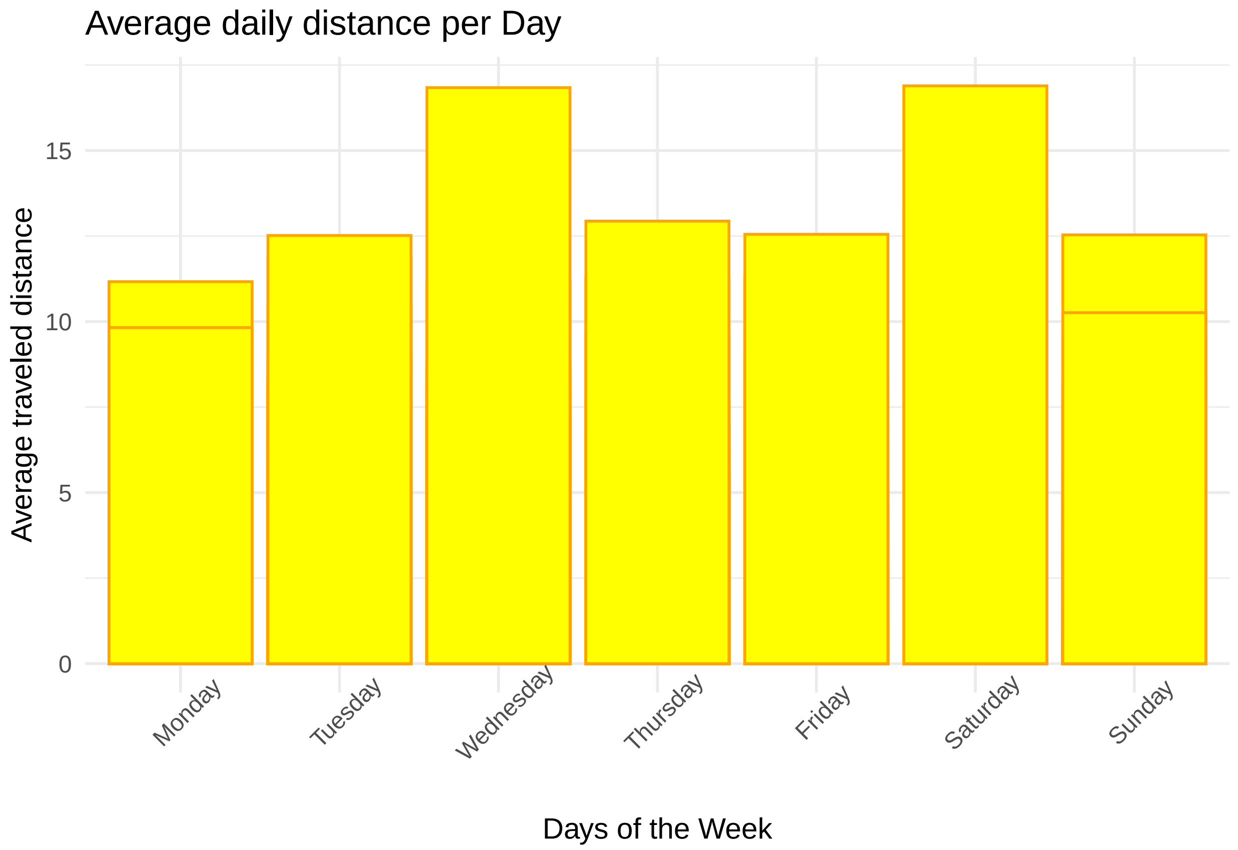
Consistent users mean consistent results?

What is the average distance traveled by users for each day of the week? It is consistent or changes from day to day?

```
library(tidyverse)
library(dplyr)
grouped_distance <- daily_activity %>%
  group_by(Id, DayOfWeek) %>%
  summarise(avg_distance = mean(TotalDistance))

grouped_distance$DayOfWeek <- factor(grouped_distance$DayOfWeek, levels = c("Monday", "Tuesday", "Wedne

ggplot(grouped_distance, aes(x = DayOfWeek, y = avg_distance)) +
 geom_bar(stat = "identity", fill = "yellow", color = "orange", position = "dodge") +
  labs(title = "Average daily distance per Day",
       x = "Days of the Week",
       y = "Average traveled distance" ,
       fill = "Id") +
  geom_smooth(method = "lm", se = TRUE) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45))
```



Trend #6:

We can conclude the existence of a regular walking trend for our users with a consistent average daily distance traveled. We note also two exceptions for both Wednesdays and Saturdays where we see a slight increase in the average distance traveled.

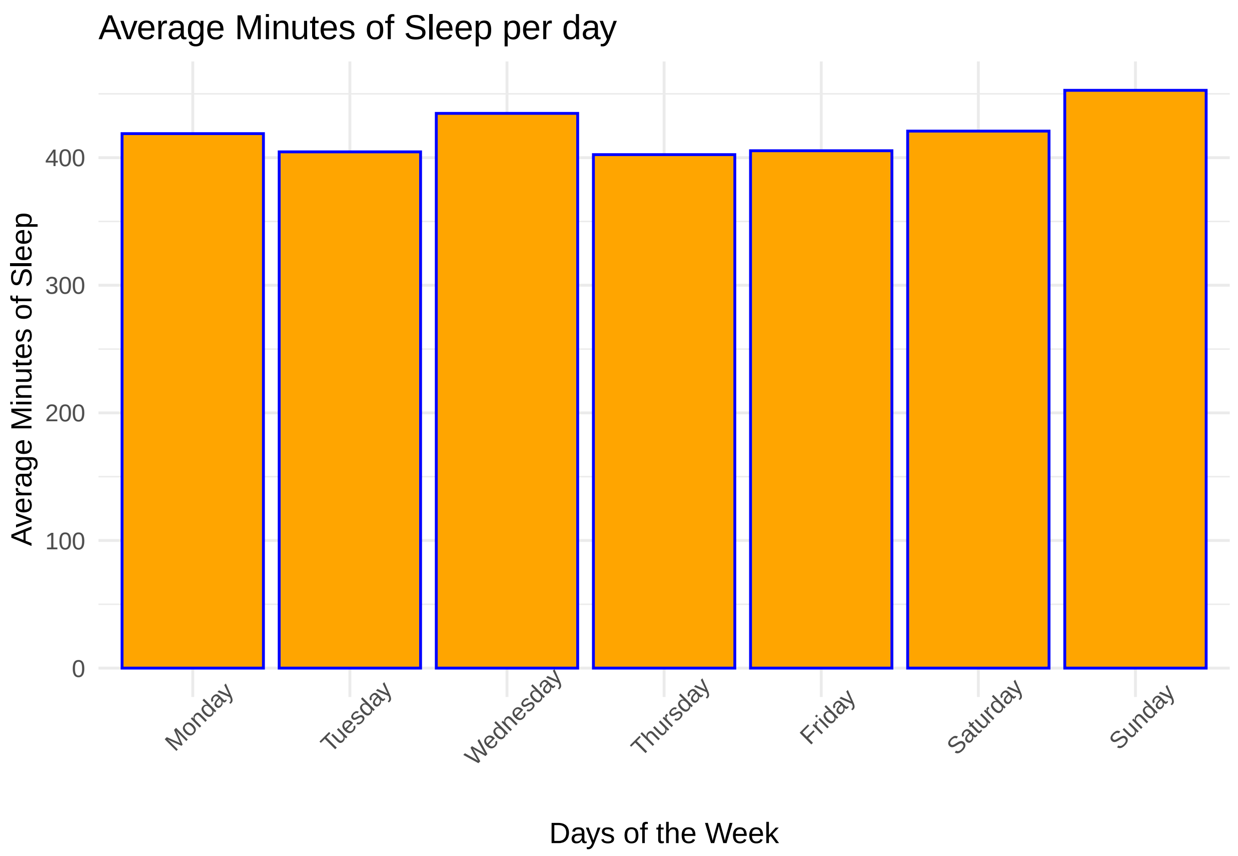
Consistent users mean consistent results?

What is the sleeping pattern for users for each day of the week?

```
combined_data$DayOfWeek <- weekdays(as.Date(combined_data$Date))
combined_data$DayOfWeek <- factor(combined_data$DayOfWeek, levels = c("Monday", "Tuesday", "Wednesday",

grouped_sleep <- combined_data %>%
  group_by(DayOfWeek) %>%
  summarise(avg_sleep = mean(TotalMinutesAsleep))

ggplot(grouped_sleep, aes(x = DayOfWeek, y = avg_sleep)) +
 geom_bar(stat = "identity", fill = "orange", color = "blue", position = "dodge") +
  labs(title = "Average Minutes of Sleep per day",
       x = "Days of the Week",
       y = "Average Minutes of Sleep" ,
       fill = "Id") +
  geom_smooth(method = "lm", se = TRUE) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45))
```



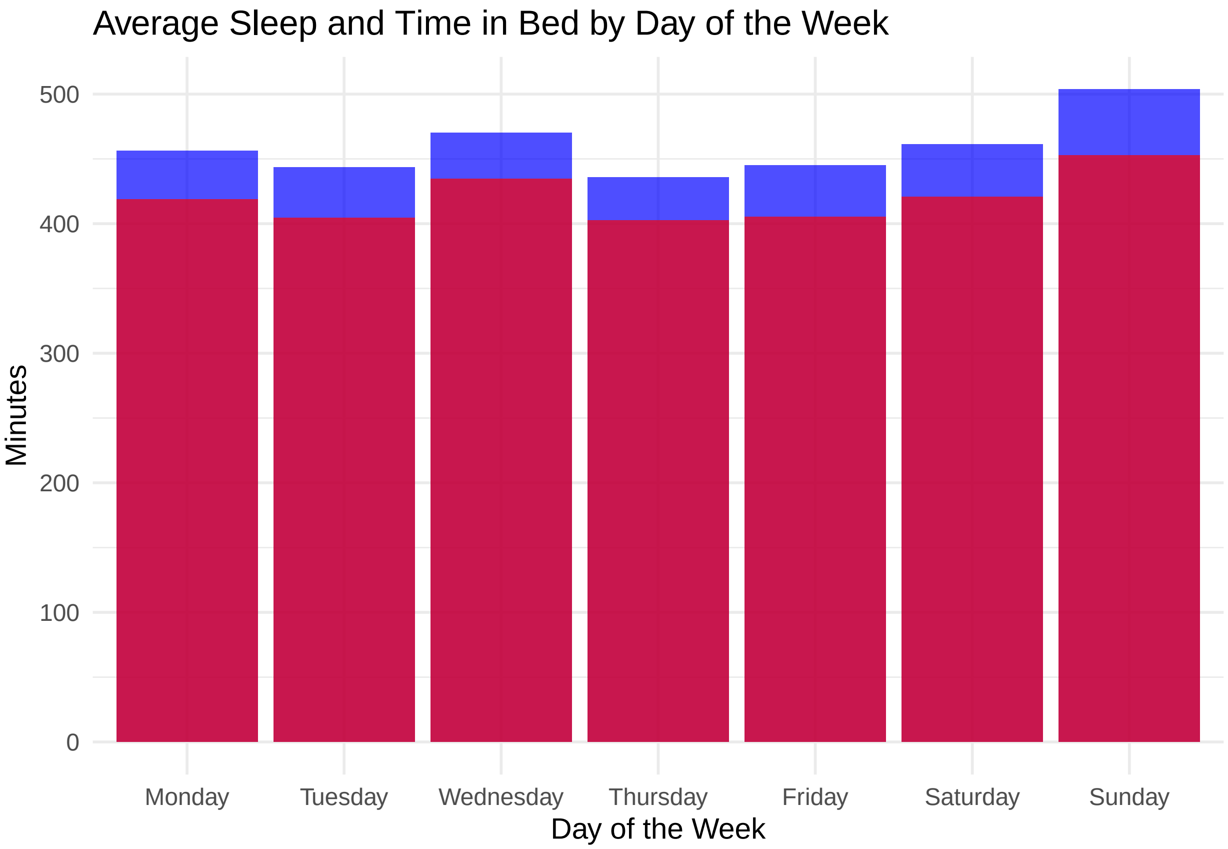
Trend #7:

Users maintain a consistent sleeping habit for the days of the week as they sleep on average 7 hours per day.

```
combined_data$DayOfWeek <- weekdays(as.Date(combined_data$Date))
combined_data$DayOfWeek <- factor(combined_data$DayOfWeek, levels = c("Monday", "Tuesday", "Wednesday",

grouped_sleepBed <- combined_data %>%
  group_by(DayOfWeek) %>%
  summarise(avg_sleep = mean(TotalMinutesAsleep), avg_timeInBed = mean(TotalTimeInBed))

ggplot(grouped_sleepBed, aes(x = DayOfWeek)) +
  geom_bar(aes(y = avg_timeInBed), stat = "identity", fill = "blue", alpha = 0.7, position = "dodge") +
  geom_bar(aes(y = avg_sleep), stat = "identity", fill = "red", alpha = 0.7, position = "dodge") +
  labs(title = "Average Sleep and Time in Bed by Day of the Week",
       y = "Minutes",
       x = "Day of the Week") +
  scale_fill_manual(values = c("blue", "red")) +
  theme_minimal()
```



Trend #8:

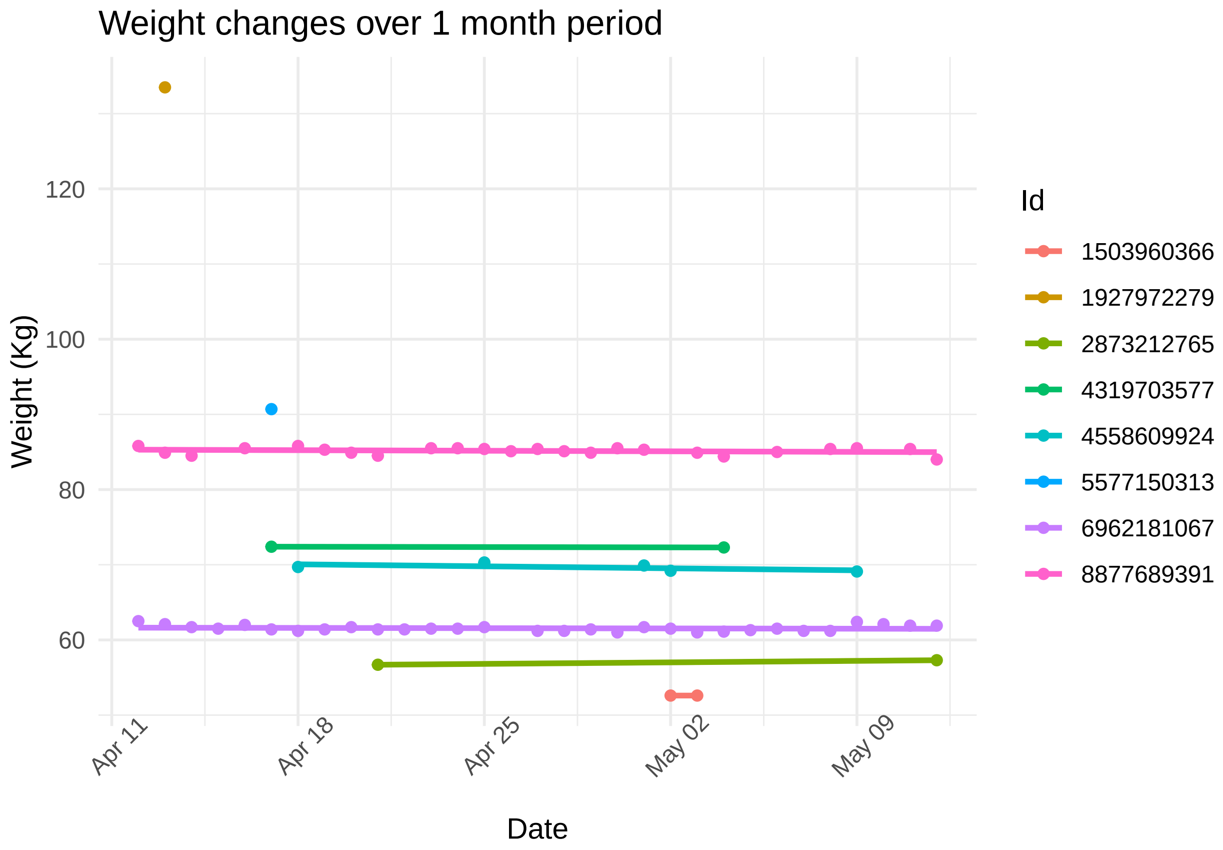
We see a clear consistency in the minutes users stay in bed vs the sleeping minutes. Almost every day users stay in bed the same time before falling asleep. Similarly we see a consistency in the average sleeping time for our users with an average of 420 minutes or 7 hours per day.

Weight loss pattern?

More calories burned mean more weight loss?

```r
 weight_data2 <- weight_data %>%
  group_by(Id) %>%
  summarise(across(everything(), list(~ mean(., na.rm = TRUE)), .names = "Avg_{.col}"))

 library(ggplot2)
ggplot(data = weight_data, aes(x = Date, y = WeightKg, color = as.factor(Id))) +
  geom_point() +
  geom_smooth(aes(group = Id), method = "lm", se = FALSE) +
  labs(title = "Weight changes over 1 month period",
       x = "Date",
       y = "Weight (Kg)",
       color = "Id") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45))
```



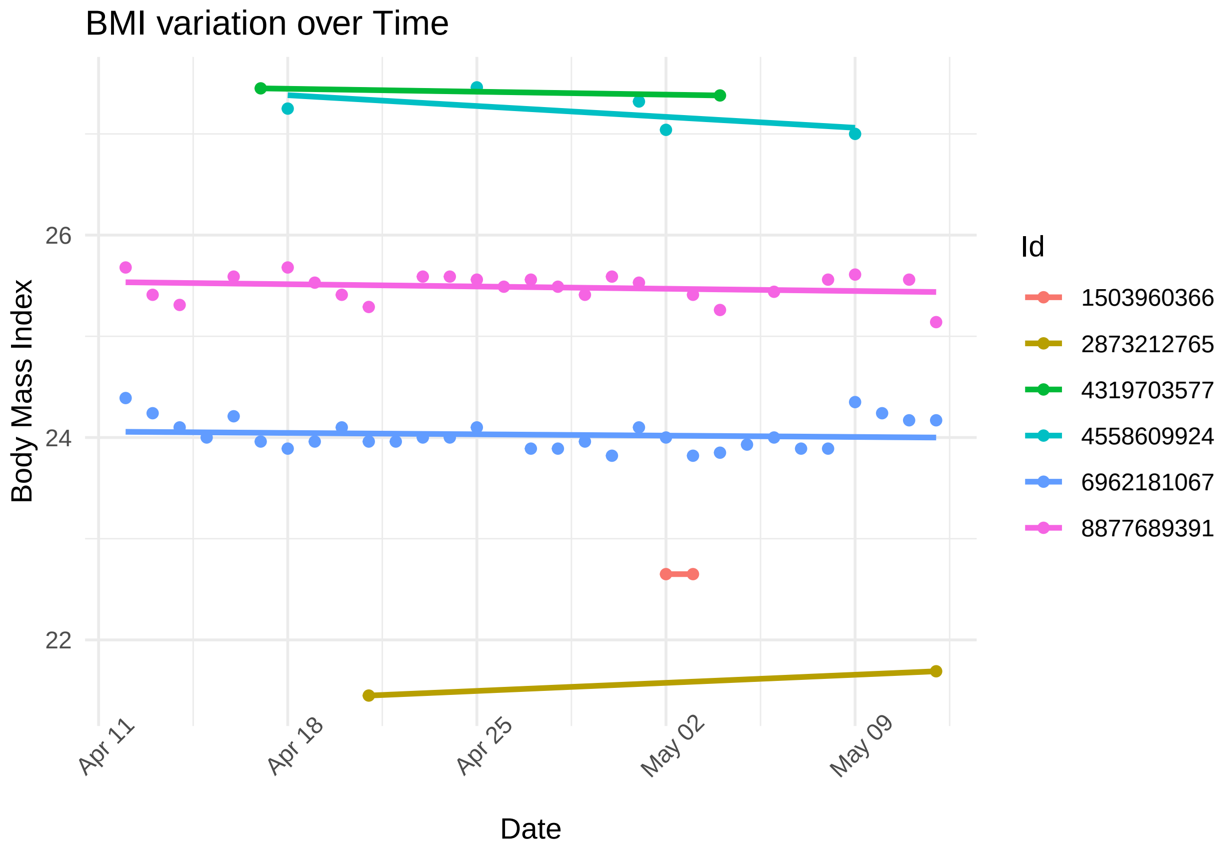
Trend #9:

From the scattered plot, we see that from the 8 users who recorded their weight over the 1 month period, very small variation is recorded in their weights. Again more data is necessary to draw a constructive conclusion and fi out if users are gaining or losing weight over time.

More active minutes mean less BMI?

```r
library(ggplot2)
filtered_data <- weight_data[weight_data$BMI < 28, ]

ggplot(data = filtered_data, aes(x = Date, y = BMI, color = as.factor(Id))) +
  geom_point() +
  geom_smooth(aes(group = Id), method = "lm", se = FALSE) +
  labs(title = "BMI variation over Time",
       x = "Date",
       y = "Body Mass Index",
       color = "Id") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45))
```

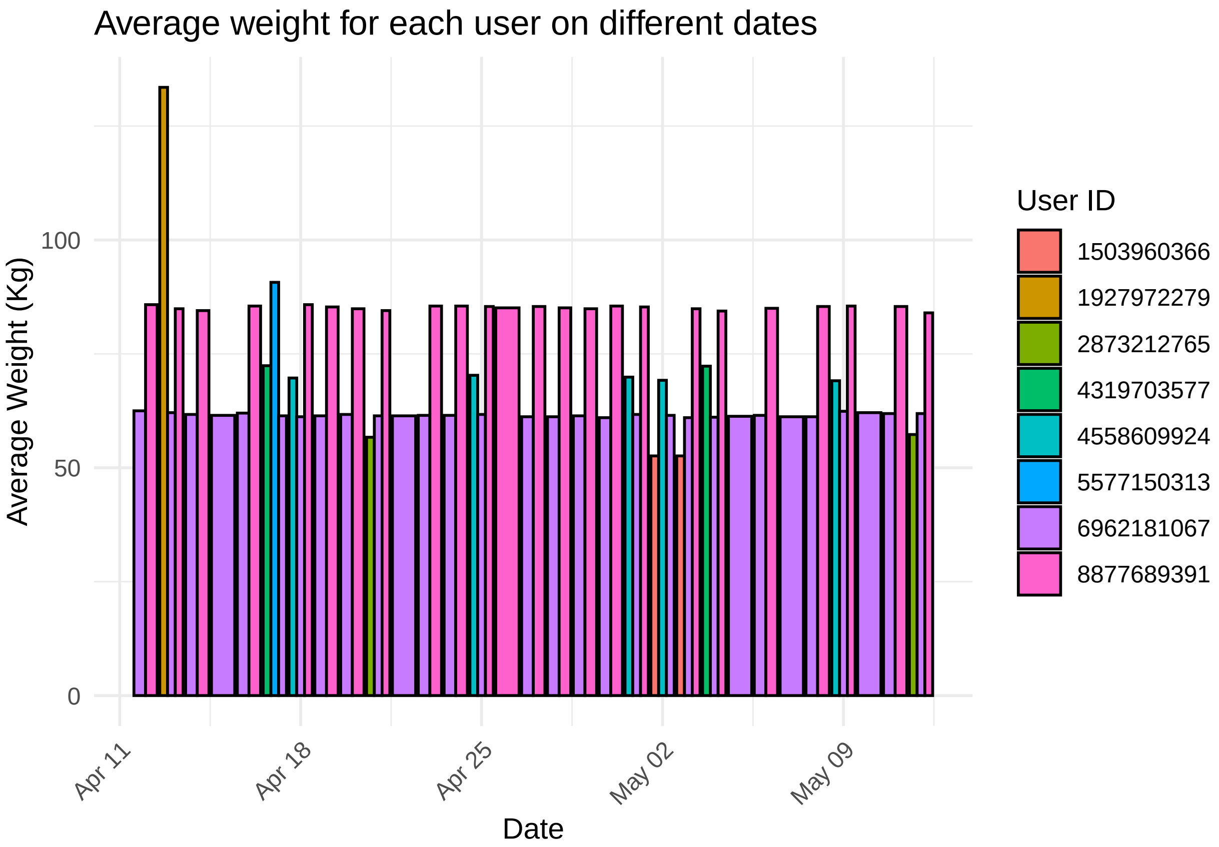


Let's Plot the results

```
grouped_weight <- weight_data %>%
  group_by(Id, Date) %>%
  summarise(AvgWeightKg = mean(WeightKg))  # We used the mean aggregation function to calculate the ave
```

```
## `summarise()` has grouped output by 'Id'. You can override using the `.groups`
## argument.
```

```
ggplot(grouped_weight, aes(x = Date, y = AvgWeightKg, fill = as.factor(Id))) +
  geom_bar(stat = "identity", position = "dodge", color = "black") +
  labs(title = "Average weight for each user on different dates",
       x = "Date",
       y = "Average Weight (Kg)",
       fill = "User ID") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Trend #10:

From the 8 users that registered their weights over the 1 month period, only 5 have consistent registered inputs. All others have only 1 registered weight log.

Summary of Findings:

Users Behavior

- **Early birds:** Rise at 7:00am and go to bed at 10:00pm
- **Weekend Warriors:** Most active on Saturdays.
- **Work Commuters:** Most distance traveled at commuting hours (8:00am, 1:00pm, and 6:00pm)

Users Preferences:

- **Data Conscious:** Users prefer not to enter their data manually.
- **Smart Device Inseparable:** They do not wear their devices continuously.

Users Pattern & Trends:

- **Calories:** More Steps, distance, active minutes = More Calories burned.
- **Healthy Lifestyle:** Consistency in sleeping and activity habits = Maintaining shape (Weight and BMI)

Recommendations:

- **Survey & data collection:** More data is necessary to draw better conclusions. Historical data will give us a better understanding of user trends over time.
- **Data entry Log reward program:** Push users to log their data about their weight. This will make them use the devices more and achieve their goals more efficiently.
- **Notification features:** Add new notification features that will alert users when their average daily sleeping minutes, steps, distances, calories are not met. Move! Time to go to bed! For example. This will enhance users experience with the app and help them in return achieve their goals more rapidly.

THANK YOU!

We hope you'll use these insights and recommendations to guide the marketing department and help unlock new growth opportunities for the company.

Here is the link to read the case study from my personal portfolio:

PORTFOLIO

Link to download the case study presentation:

PRESENTATION