Bellabeat Analysis

Daniele Iacobone

2023-12-05

Bellabeat Case Study

Scenario

Report made for the Marketing Team at Bellabeat, a high-tech manufacturer of health-focused products for women. Bellabeat is a successful small company, but they have the potential to become a larger player in the global smart device market. Urška Sršen, cofounder and Chief Creative Officer of Bellabeat, believes that analyzing smart device fitness data could help unlock new growth opportunities for the company. I have been asked to focus on one of Bellabeat's products and analyze smart device data to gain insight into how consumers are using their smart devices. The insights you discover will then help guide marketing strategy for the company. I will present my analysis to the Bellabeat executive team along with high-level recommendations for Bellabeat's marketing strategy.

Ask

The purpose is to analyze smart device usage data in order to gain insight into how consumers use non-Bellabeat smart devices. Then, select one Bellabeat product to apply these insights to in my presentation. 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?

Prepare

I will use public data that explores smart device users' daily habits. FitBit Fitness Tracker Data (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. The datasets are stored in CSV format, containing data about daily activity, daily calories, hearth rate and many other informations.

Analyze

Preliminary Analysis

Loading Libraries

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(skimr)
```

Loading CSV Files

```
daily_activity <- read_csv("Fitbase_datasets/dailyActivity_merged.csv")
```

```
## Rows: 940 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): ActivityDate
## dbl (14): Id, TotalSteps, TotalDistance, TrackerDistance, LoggedActivitiesDi...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
sleep_day <- read_csv("Fitbase_datasets/sleepDay_merged.csv")
```

```
## Rows: 413 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): SleepDay
## dbl (4): Id, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Display Summary Statistics

```r
head(daily_activity, 5) # Top values
```

```
## # A tibble: 5 x 15
##            Id ActivityDate TotalSteps TotalDistance TrackerDistance
##         <dbl> <chr>             <dbl>         <dbl>           <dbl>
## 1 1503960366 4/12/2016         13162          8.5             8.5
## 2 1503960366 4/13/2016         10735          6.97            6.97
## 3 1503960366 4/14/2016         10460          6.74            6.74
## 4 1503960366 4/15/2016          9762          6.28            6.28
## 5 1503960366 4/16/2016         12669          8.16            8.16
## # i 10 more variables: LoggedActivitiesDistance <dbl>,
## #   VeryActiveDistance <dbl>, ModeratelyActiveDistance <dbl>,
## #   LightActiveDistance <dbl>, SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <dbl>, FairlyActiveMinutes <dbl>,
## #   LightlyActiveMinutes <dbl>, SedentaryMinutes <dbl>, Calories <dbl>
```

```r
tail(daily_activity, 5) # Bottom values
```

```
## # A tibble: 5 x 15
##            Id ActivityDate TotalSteps TotalDistance TrackerDistance
##         <dbl> <chr>             <dbl>         <dbl>           <dbl>
## 1 8877689391 5/8/2016          10686          8.11            8.11
## 2 8877689391 5/9/2016          20226         18.2            18.2
## 3 8877689391 5/10/2016         10733          8.15            8.15
## 4 8877689391 5/11/2016         21420         19.6            19.6
## 5 8877689391 5/12/2016          8064          6.12            6.12
## # i 10 more variables: LoggedActivitiesDistance <dbl>,
## #   VeryActiveDistance <dbl>, ModeratelyActiveDistance <dbl>,
## #   LightActiveDistance <dbl>, SedentaryActiveDistance <dbl>,
## #   VeryActiveMinutes <dbl>, FairlyActiveMinutes <dbl>,
## #   LightlyActiveMinutes <dbl>, SedentaryMinutes <dbl>, Calories <dbl>
```

```r
daily_activity %>%
  select(TotalSteps,
         TotalDistance,
SedentaryMinutes) %>%
  summary()
```

```
##    TotalSteps    TotalDistance    SedentaryMinutes
##  Min.   :    0   Min.   : 0.000   Min.   :   0.0
##  1st Qu.: 3790   1st Qu.: 2.620   1st Qu.: 729.8
##  Median : 7406   Median : 5.245   Median :1057.5
##  Mean   : 7638   Mean   : 5.490   Mean   : 991.2
##  3rd Qu.:10727   3rd Qu.: 7.713   3rd Qu.:1229.5
##  Max.   :36019   Max.   :28.030   Max.   :1440.0
```

```r
sum(is.na(daily_activity)) # There are not NA values
```

```
## [1] 0
```

```r
n_distinct(daily_activity$Id) # Only 33 unique IDs
```

```
## [1] 33
```

This dataset refers only to 33 unique IDs.

```r
head(sleep_day, 5)
```

```
## # A tibble: 5 x 5
##           Id SleepDay        TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##        <dbl> <chr>                       <dbl>              <dbl>          <dbl>
## 1 1503960366 4/12/2016 12:0~                 1                327            346
## 2 1503960366 4/13/2016 12:0~                 2                384            407
## 3 1503960366 4/15/2016 12:0~                 1                412            442
## 4 1503960366 4/16/2016 12:0~                 2                340            367
## 5 1503960366 4/17/2016 12:0~                 1                700            712
```

```r
tail(sleep_day, 5)
```

```
## # A tibble: 5 x 5
##           Id SleepDay        TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##        <dbl> <chr>                       <dbl>              <dbl>          <dbl>
## 1 8792009665 4/30/2016 12:0~                 1                343            360
## 2 8792009665 5/1/2016 12:00~                 1                503            527
## 3 8792009665 5/2/2016 12:00~                 1                415            423
## 4 8792009665 5/3/2016 12:00~                 1                516            545
## 5 8792009665 5/4/2016 12:00~                 1                439            463
```

```r
sleep_day %>%
  select(TotalSleepRecords,
                   TotalMinutesAsleep,
                   TotalTimeInBed) %>%
  summary()
```

```
##  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.000     Min.   : 58.0      Min.   : 61.0
##  1st Qu.:1.000     1st Qu.:361.0      1st Qu.:403.0
##  Median :1.000     Median :433.0      Median :463.0
##  Mean   :1.119     Mean   :419.5      Mean   :458.6
##  3rd Qu.:1.000     3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :3.000     Max.   :796.0      Max.   :961.0
```

```r
sum(is.na(sleep_day)) # There are no NAs
```

```
## [1] 0
```

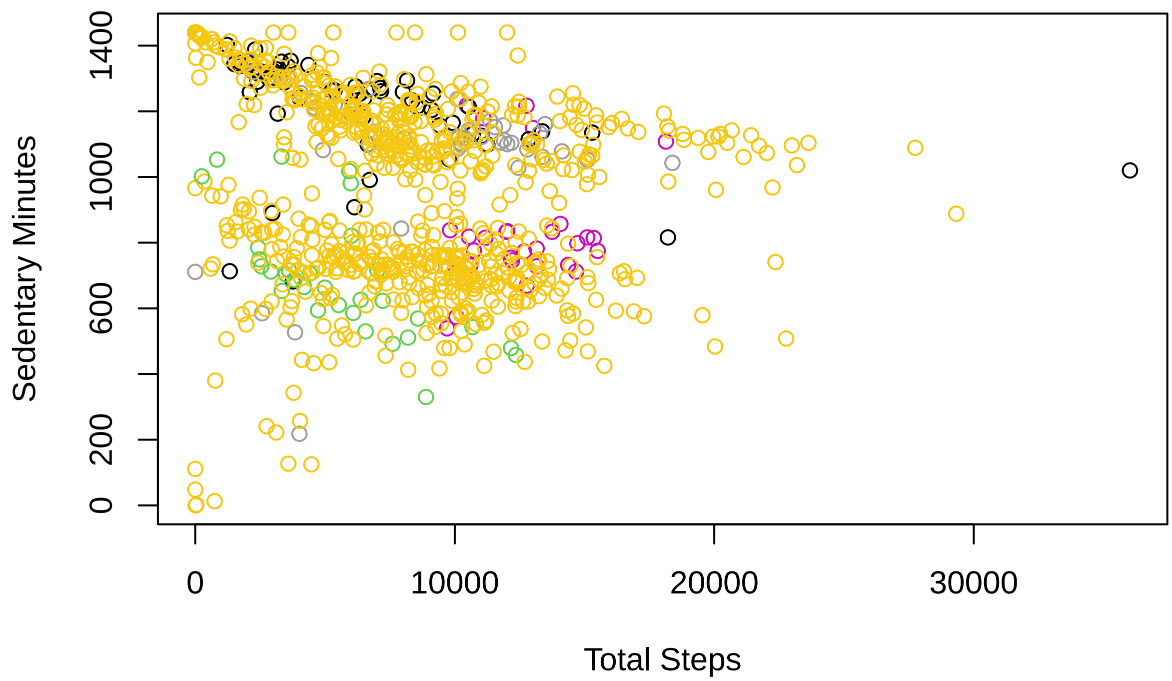
```r
n_distinct(sleep_day$Id)# Only 24 unique IDs
```

```
## [1] 24
```

Quick Visualisations

What's the relationship between steps taken in a day and sedentary minutes?

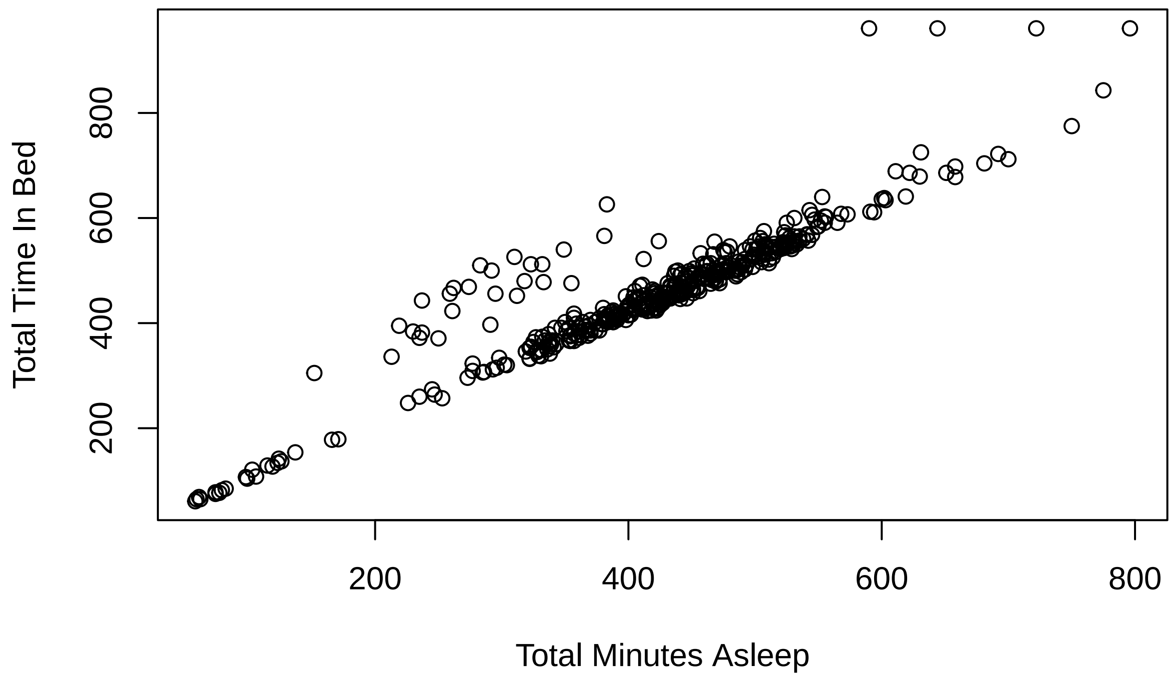
```
plot(daily_activity$TotalSteps, daily_activity$SedentaryMinutes,
     xlab = "Total Steps", ylab= "Sedentary Minutes",
     col = daily_activity$Id)
```



The relationship between steps taken in a day and sedentary minutes is often inversely proportional. In other words, as the number of steps taken increases, sedentary minutes (the time spent sitting or being inactive) generally decreases. If the target audience is already somewhat active, Bellabeat could market the product as a way to measure and optimize the steps they are already taking. Highlight advanced features such as detailed activity analysis, performance tracking, or integration with other fitness metrics.

What's the relationship between minutes asleep and time in bed?

```
plot(sleep_day$TotalMinutesAsleep, sleep_day$TotalTimeInBed,
     xlab = "Total Minutes Asleep", ylab = "Total Time In Bed")
```



The relationship between minutes asleep and time in bed is typically referred to as sleep efficiency. As the graph illustrates, a clearly visible linear relationship exists between the total minutes asleep and the total time in bed.

In Detail Analysis

Join of the Dataframes

Now, I am merging the two datasets using the variable 'Id' as the primary key, which is common to both datasets.

```
full <- daily_activity %>%
  merge(sleep_day, by = c("Id" = "Id"))
n_distinct(full$Id)
```

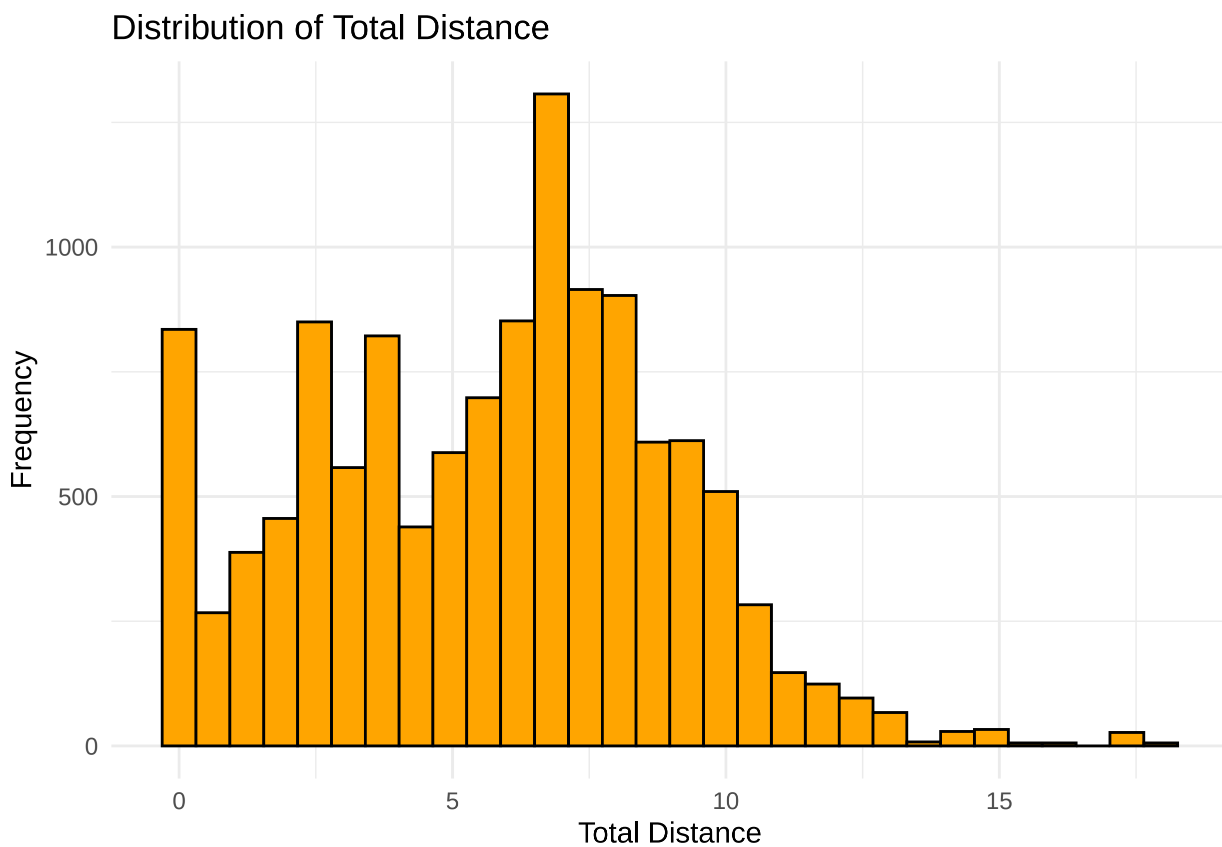
```
## [1] 24
```

The join has been completed, and 24 unique IDs have been identified.

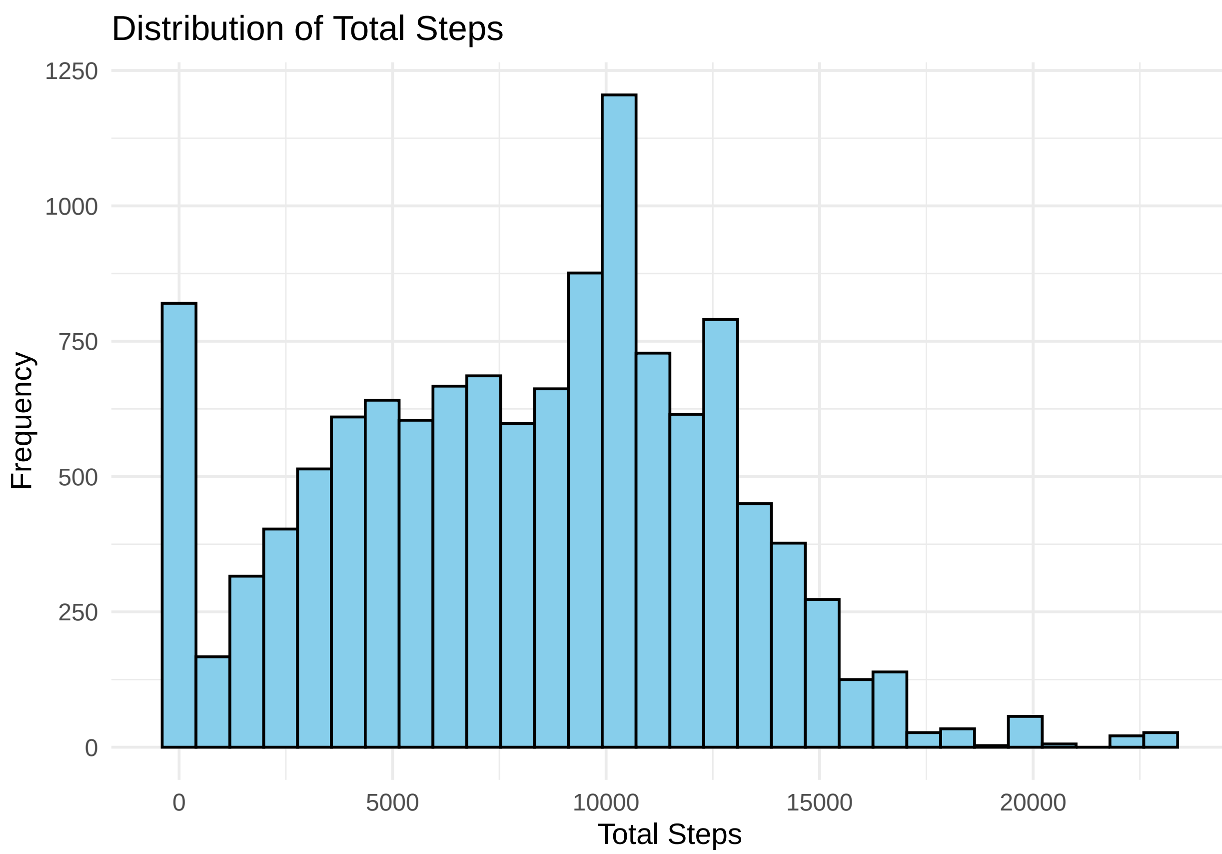
Visualizations for the overall distribution of steps and total distance

```r
library(ggplot2)

full %>%
  ggplot(aes(x = TotalDistance)) +
  geom_histogram(fill = "orange", color = "black", bins = 30) +
  labs(title = "Distribution of Total Distance",
       x = "Total Distance",
       y = "Frequency") +
  theme_minimal()
```




```r
full %>%
  ggplot(aes(x = TotalSteps)) +
  geom_histogram(fill = "skyblue", color = "black", bins = 30) +
  labs(title = "Distribution of Total Steps",
       x = "Total Steps",
       y = "Frequency") +
  theme_minimal()
```



The bar chart for total distance reveals that the majority of measurements were recorded with distances ranging between 5 and 8 km, notably with over 1250 measurements at approximately 7 km. Although there are numerous readings for distances below 5 km, they do not exceed 1000 measurements. On the other hand, measurements beyond 10 km are relatively scarce.

The chart for total steps indicates that the highest number of measurements, around 1200, corresponds to 10,000 steps. The readings show a generally increasing trend up to 10,000 steps, followed by a decreasing trend in measurements.

The two charts are correlated, displaying similar characteristics overall.

Total Steps per Weekday

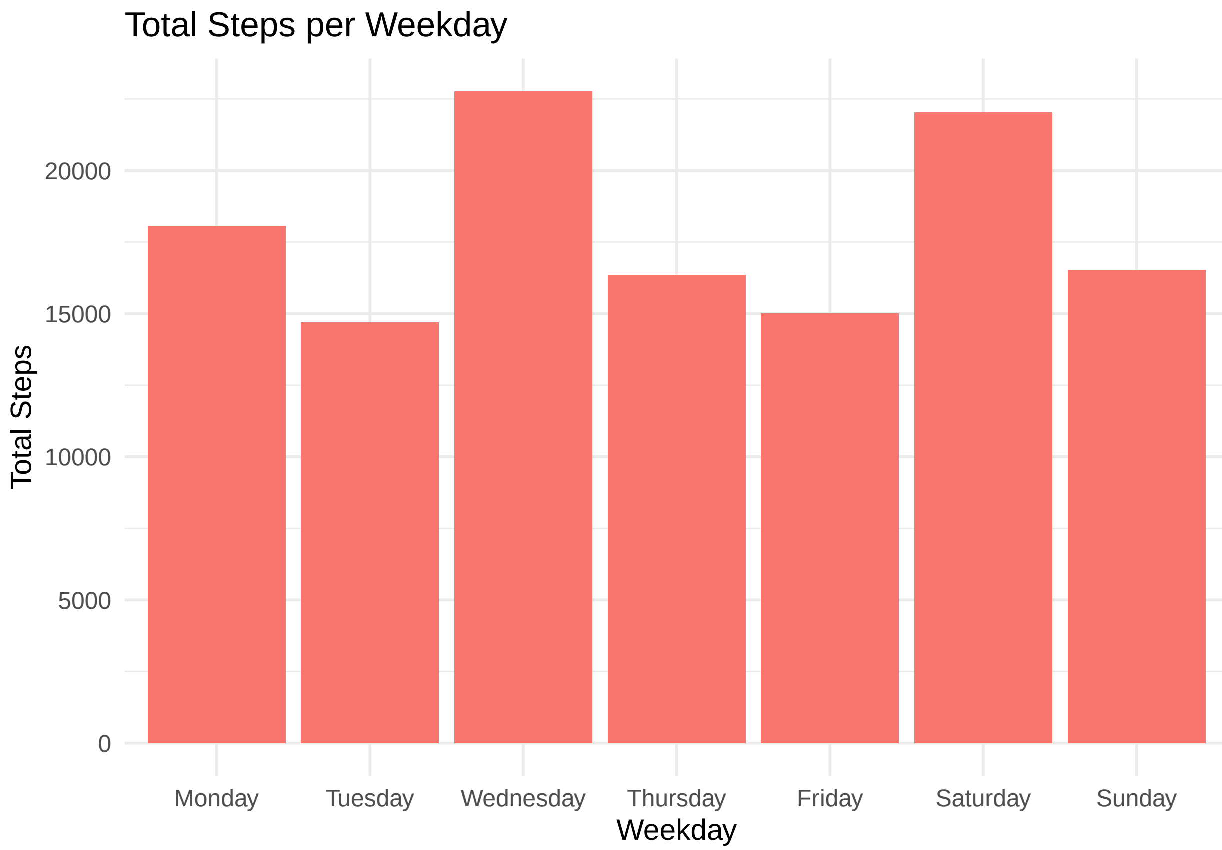
```
library(lubridate)
full <- full %>%
  mutate(ActivityDate = as.Date(ActivityDate),
         Weekday = weekdays(ActivityDate))

full$Weekday <- factor(full$Weekday, levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday",

full <- full[!is.na(full$Weekday), ]

ggplot(full, aes(x = Weekday, y = TotalSteps, fill = "red" )) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Weekday", y = "Total Steps") +
```

```r
  ggtitle("Total Steps per Weekday") +
  theme_minimal() +
  theme(legend.position = "none", legend.key = element_blank())
```



As the bar chart depicts, the number of total steps relative to the day of the week does not follow a consistent trend. The day with the highest number of steps is Wednesday, with approximately 22,500 steps, and another peak is observed on Saturday with a similar figure. The remaining days of the week typically see an average of no fewer than 15,000 steps.

Analysis with other Datasets

Loading other datasets

```r
daily_calories <- read_csv("Fitbase_datasets/dailyCalories_merged.csv")
```

```
## Rows: 940 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityDay
## dbl (2): Id, Calories
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
daily_intensities <- read_csv("Fitbase_datasets/dailyIntensities_merged.csv")
```

```
## Rows: 940 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityDay
## dbl (9): Id, SedentaryMinutes, LightlyActiveMinutes, FairlyActiveMinutes, Ve...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
daily_steps <- read_csv("Fitbase_datasets/dailySteps_merged.csv")
```

```
## Rows: 940 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): ActivityDay
## dbl (2): Id, StepTotal
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
weightloginfo <- read_csv("Fitbase_datasets/weightLogInfo_merged.csv")
```

```
## Rows: 67 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Date
## dbl (6): Id, WeightKg, WeightPounds, Fat, BMI, LogId
## lgl (1): IsManualReport
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Datasets Merging

```r
merge1 <- merge(daily_activity, daily_calories,
                by = c("Id", "Calories"))
merge2 <- merge(daily_intensities, daily_intensities,
                by = c("Id", "ActivityDay",
                       "SedentaryMinutes",
                  "LightlyActiveMinutes",
                  "FairlyActiveMinutes",
                  "VeryActiveMinutes",
                  "SedentaryActiveDistance",
                  "LightActiveDistance",
                  "ModeratelyActiveDistance",
                  "VeryActiveDistance"))

merge_daily <- merge(merge1, merge2, by = c("Id","ActivityDay","SedentaryMinutes", "LightlyActiveMinute
```

```
select(-ActivityDay) %>% rename(Date = ActivityDate)

daily_data <- merge(merge_daily, sleep_day, by = "Id",all=TRUE) %>% drop_na() %>% select(-SleepDay, -Tra

daily_data <- merge(daily_data, weightloginfo, by = "Id")
```

Statistic Analysis and NAs

```
summary(daily_data)
```

```
##        Id             SedentaryMinutes LightlyActiveMinutes FairlyActiveMinutes
##  Min.   :1.504e+09   Min.   :   0.0   Min.   :  0.0        Min.   : 0.00
##  1st Qu.:6.962e+09   1st Qu.: 637.0   1st Qu.:199.0        1st Qu.: 4.00
##  Median :6.962e+09   Median : 683.0   Median :240.0        Median :15.00
##  Mean   :6.409e+09   Mean   : 701.1   Mean   :237.7        Mean   :18.03
##  3rd Qu.:6.962e+09   3rd Qu.: 731.0   3rd Qu.:289.0        3rd Qu.:33.00
##  Max.   :6.962e+09   Max.   :1440.0   Max.   :432.0        Max.   :74.00
##
##  VeryActiveMinutes SedentaryActiveDistance LightActiveDistance
##  Min.   :  0.00    Min.   :0.000000        Min.   :0.00
##  1st Qu.:  0.00    1st Qu.:0.000000        1st Qu.:2.87
##  Median : 18.00    Median :0.000000        Median :3.90
##  Mean   : 23.36    Mean   :0.005705        Mean   :3.88
##  3rd Qu.: 38.00    3rd Qu.:0.000000        3rd Qu.:4.88
##  Max.   :210.00    Max.   :0.110000        Max.   :7.70
##
##  ModeratelyActiveDistance VeryActiveDistance    Calories       Date.x
##  Min.   :0.0000           Min.   :0.000      Min.   :   0   Length:34234
##  1st Qu.:0.1900           1st Qu.:0.000      1st Qu.:1850   Class :character
##  Median :0.6800           Median :1.200      Median :2039   Mode  :character
##  Mean   :0.8986           Mean   :1.588      Mean   :2011
##  3rd Qu.:1.6800           3rd Qu.:2.840      3rd Qu.:2173
##  Max.   :2.3900           Max.   :7.650      Max.   :4552
##
##    TotalSteps    TotalDistance    LoggedActivitiesDistance TotalSleepRecords
##  Min.   :    0   Min.   : 0.00   Min.   :0.0000           Min.   :1.000
##  1st Qu.: 5908   1st Qu.: 3.91   1st Qu.:0.0000           1st Qu.:1.000
##  Median :10320   Median : 6.82   Median :0.0000           Median :1.000
##  Mean   : 9507   Mean   : 6.39   Mean   :0.2726           Mean   :1.099
##  3rd Qu.:12109   3rd Qu.: 8.35   3rd Qu.:0.0000           3rd Qu.:1.000
##  Max.   :20031   Max.   :13.24   Max.   :4.0817           Max.   :3.000
##
##  TotalMinutesAsleep TotalTimeInBed    Date.y             WeightKg
##  Min.   : 59.0      Min.   : 65    Length:34234       Min.   : 52.60
##  1st Qu.:411.0      1st Qu.:424    Class :character   1st Qu.: 61.20
##  Median :442.0      Median :457    Mode  :character   Median : 61.50
##  Mean   :437.2      Mean   :456                       Mean   : 63.88
##  3rd Qu.:476.0      3rd Qu.:497                       3rd Qu.: 62.00
##  Max.   :750.0      Max.   :775                       Max.   :133.50
##
##   WeightPounds        Fat             BMI         IsManualReport
```

```
##  Min.   :116.0   Min.   :22.00   Min.   :22.65   Mode :logical
##  1st Qu.:134.9   1st Qu.:22.00   1st Qu.:23.89   FALSE:1467
##  Median :135.6   Median :25.00   Median :24.00   TRUE :32767
##  Mean   :140.8   Mean   :23.53   Mean   :24.73
##  3rd Qu.:136.7   3rd Qu.:25.00   3rd Qu.:24.21
##  Max.   :294.3   Max.   :25.00   Max.   :47.54
##                  NA's   :32653
##      LogId
##  Min.   :1.461e+12
##  1st Qu.:1.461e+12
##  Median :1.462e+12
##  Mean   :1.462e+12
##  3rd Qu.:1.462e+12
##  Max.   :1.463e+12
##
```

```
colSums(is.na(daily_data)) # No NAs
```

```
##                       Id         SedentaryMinutes     LightlyActiveMinutes
##                        0                        0                        0
##      FairlyActiveMinutes        VeryActiveMinutes  SedentaryActiveDistance
##                        0                        0                        0
##      LightActiveDistance ModeratelyActiveDistance       VeryActiveDistance
##                        0                        0                        0
##                 Calories                   Date.x               TotalSteps
##                        0                        0                        0
##            TotalDistance LoggedActivitiesDistance        TotalSleepRecords
##                        0                        0                        0
##       TotalMinutesAsleep             TotalTimeInBed                   Date.y
##                        0                        0                        0
##                 WeightKg             WeightPounds                      Fat
##                        0                        0                    32653
##                      BMI           IsManualReport                    LogId
##                        0                        0                        0
```

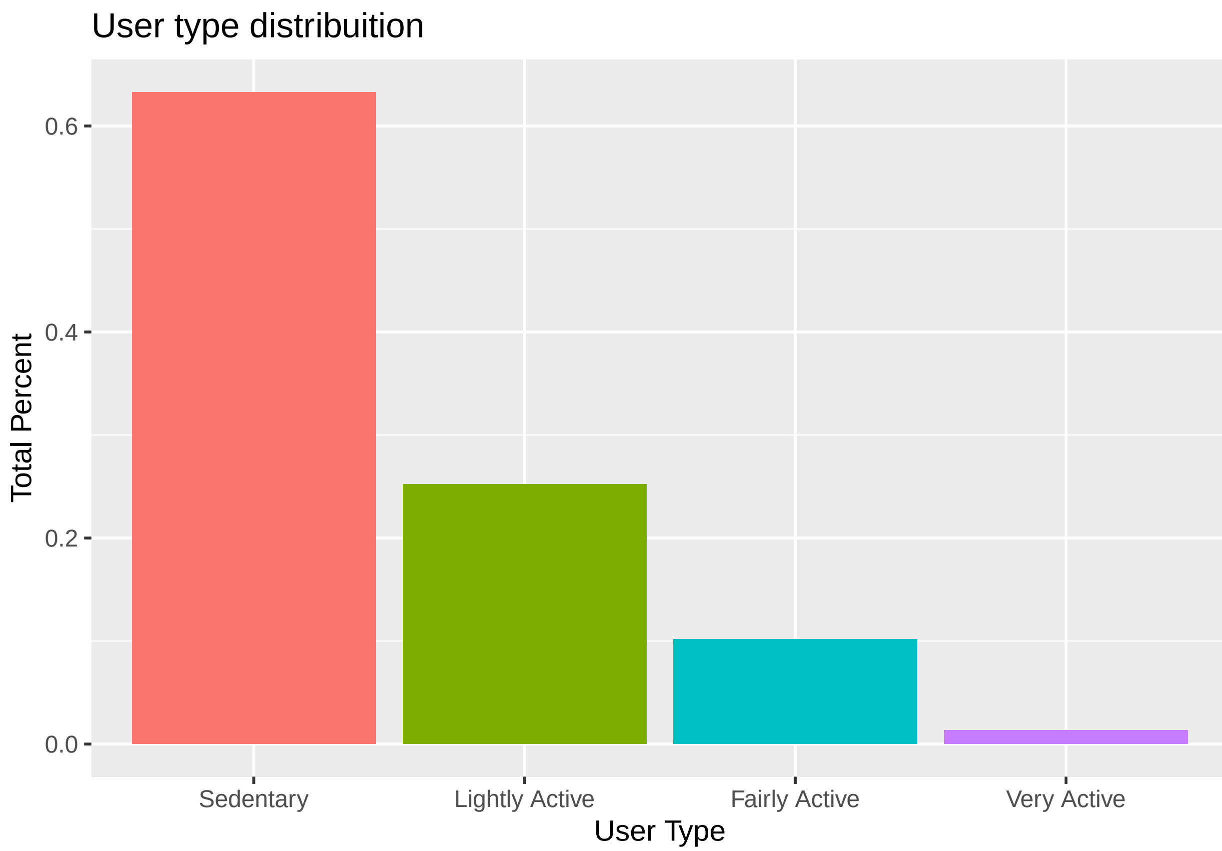
Grouping

In order to better visualize the data I will group the user into four groups for which of their activity types they have more minutes. This is useful to better visualise the patterns.

```
data_by_usertype <- daily_data %>%
reframe(user_type = factor(case_when(
    SedentaryMinutes > mean(SedentaryMinutes) & LightlyActiveMinutes < mean(LightlyActiveMinutes) & Fai
    SedentaryMinutes < mean(SedentaryMinutes) & LightlyActiveMinutes > mean(LightlyActiveMinutes) & Fai
    SedentaryMinutes < mean(SedentaryMinutes) & LightlyActiveMinutes < mean(LightlyActiveMinutes) & Fai
    SedentaryMinutes < mean(SedentaryMinutes) & LightlyActiveMinutes < mean(LightlyActiveMinutes) & Fai
),levels=c("Sedentary", "Lightly Active", "Fairly Active", "Very Active")), Calories, TotalSteps, Total
  drop_na()
```

User Type Distribuition

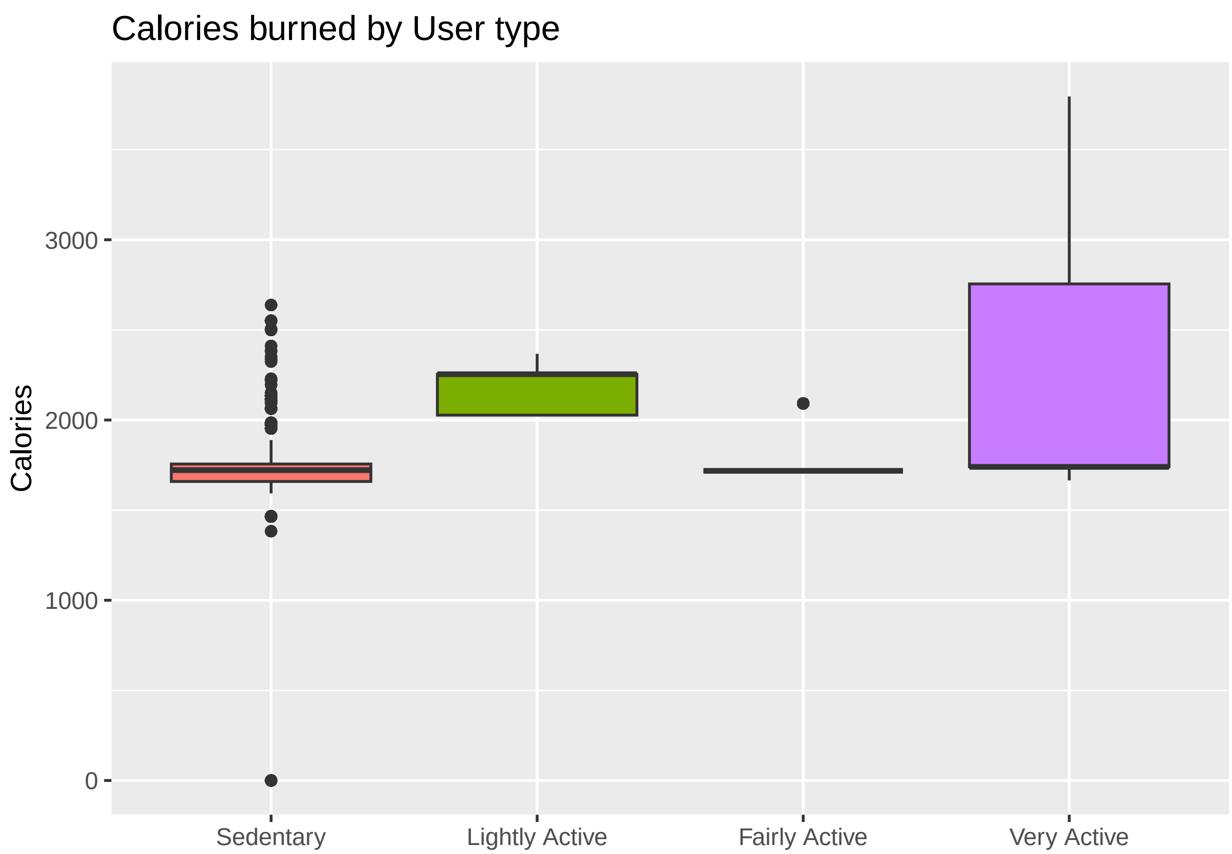
```
data_by_usertype %>%
group_by(user_type) %>%
reframe(total = n()) %>%
mutate(totals = sum(total)) %>%
group_by(user_type) %>%
reframe(total_percent = total / totals) %>%
ggplot(aes(user_type,y=total_percent, fill=user_type)) +
    geom_col() +
  labs(title = "User type distribuition", x = "User Type", y = "Total Percent") +
  theme(legend.position="none")
```



The bar chart reveals that approximately 65% of user types are classified as sedentary, around 25% as lightly active, and 10% as fairly active. The remaining category, very active, represents a relatively low percentage.

Calories burned by User Type

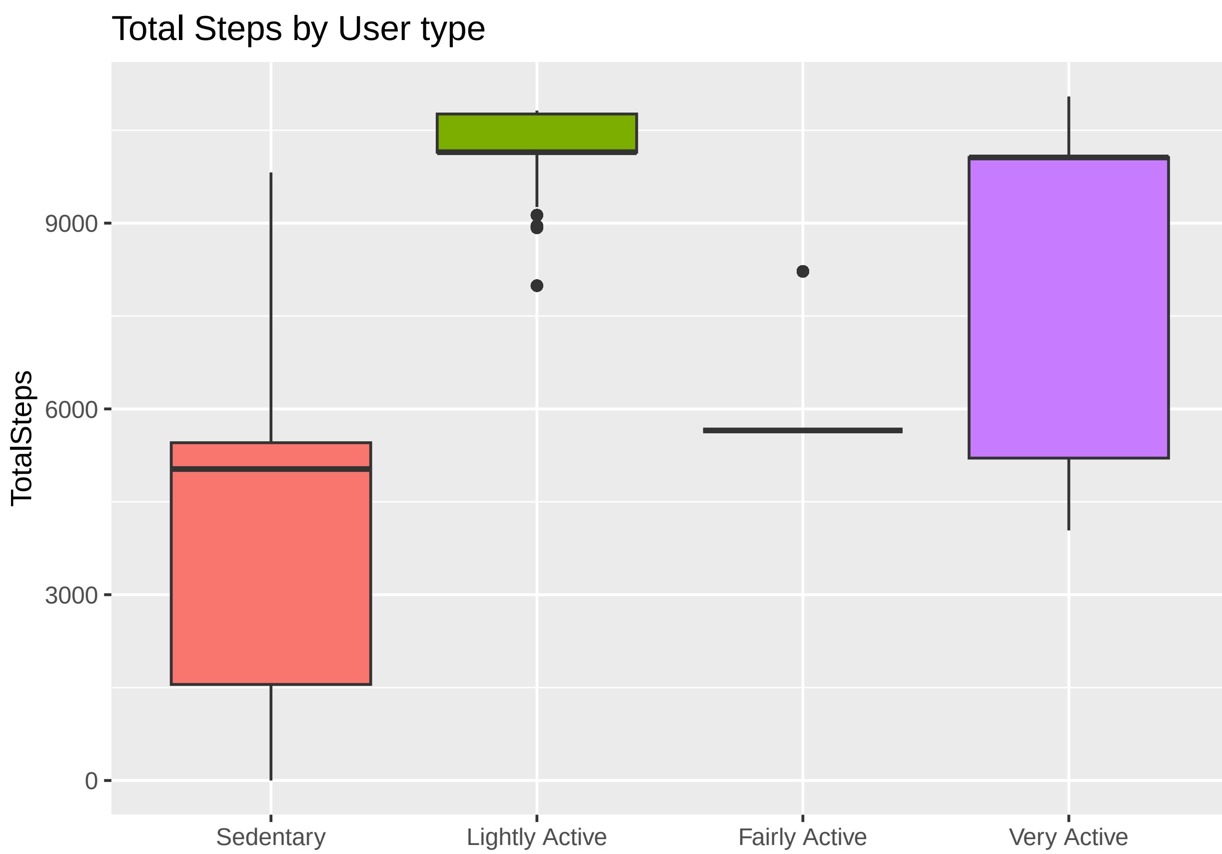
```
ggplot(data_by_usertype, aes(user_type, Calories, fill=user_type)) +
    geom_boxplot() +
    theme(legend.position="none") +
    labs(title = "Calories burned by User type", x = NULL) +
    theme(legend.position="none")
```



As expected, the box plot illustrates that the 'very active' group is the one consuming the highest number of calories, with a range between 1750 and 2750 calories. However, further investigation is needed to understand why the box plot for the 'fairly active' group is not clearly visible, despite representing 10% of the dataset.

Total Steps by User Type

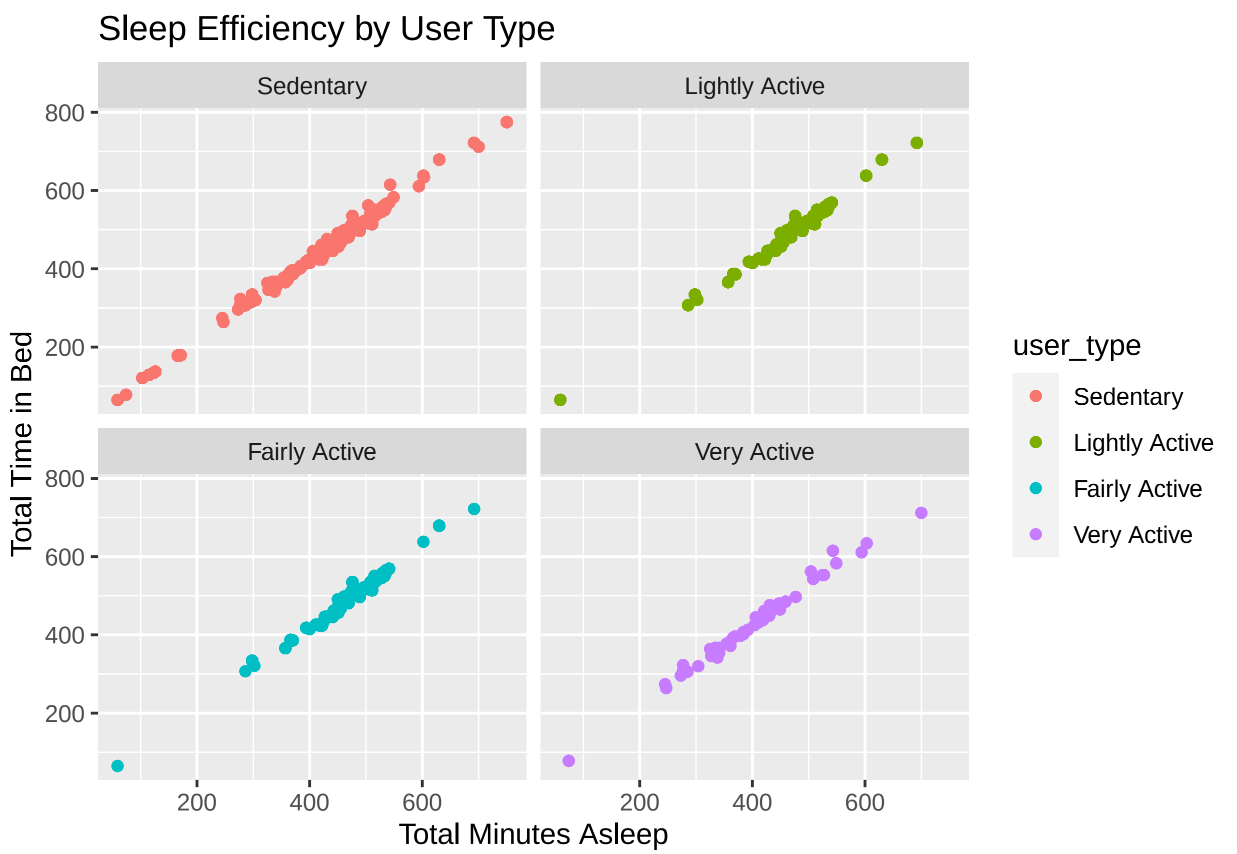
```
ggplot(data_by_usertype, aes(user_type, TotalSteps, fill=user_type)) +
    geom_boxplot() +
    theme(legend.position="none") +
    labs(title = "Total Steps by User type", x = NULL) +
    theme(legend.position="none")
```



The box plot reveals that, as expected based on their categorization, sedentary individuals have a lower range of total steps, ranging from approximately 1500 to 5500. On the other hand, the 'very active' group covers a total steps range from around 5500 to approximately 10,000.

Sleep Efficiency by User Type

```
ggplot(data_by_usertype, aes(x = TotalMinutesAsleep,
                             y = TotalTimeInBed,
                             col = user_type)) +
  geom_point() +
  labs(title = "Sleep Efficiency by User Type",
       x = "Total Minutes Asleep",
       y = "Total Time in Bed") +
  facet_wrap(~ user_type)
```



The chart indicates that the readings for the four groups display sleep efficiency values that are generally similar on average.

Division based on Physical Fitness

```
user_fit <- data_by_usertype %>%
  group_by(user_type, BMI) %>%
  mutate(Forma = if_else(BMI < 18.5, "Underweight", if_else(BMI > 18.5 & BMI < 24.9, "Normal Weight", i
  reframe(Forma, Calories, Date.x)
```

I have categorized based on Body Mass Index (BMI), an index used to assess the proportion of weight to height in a person. BMI is calculated by dividing weight in kilograms by the square of height in meters. The division you provided is as follows:

Underweight: BMI less than 18.5 Normal weight: BMI between 18.5 and 24.9 Overweight: BMI between 25 and 29.9 Obese: BMI greater than 30

```
unique(user_fit$Forma)
```

```
## [1] "Normal Weight" "Overweight"    "Obese"
```

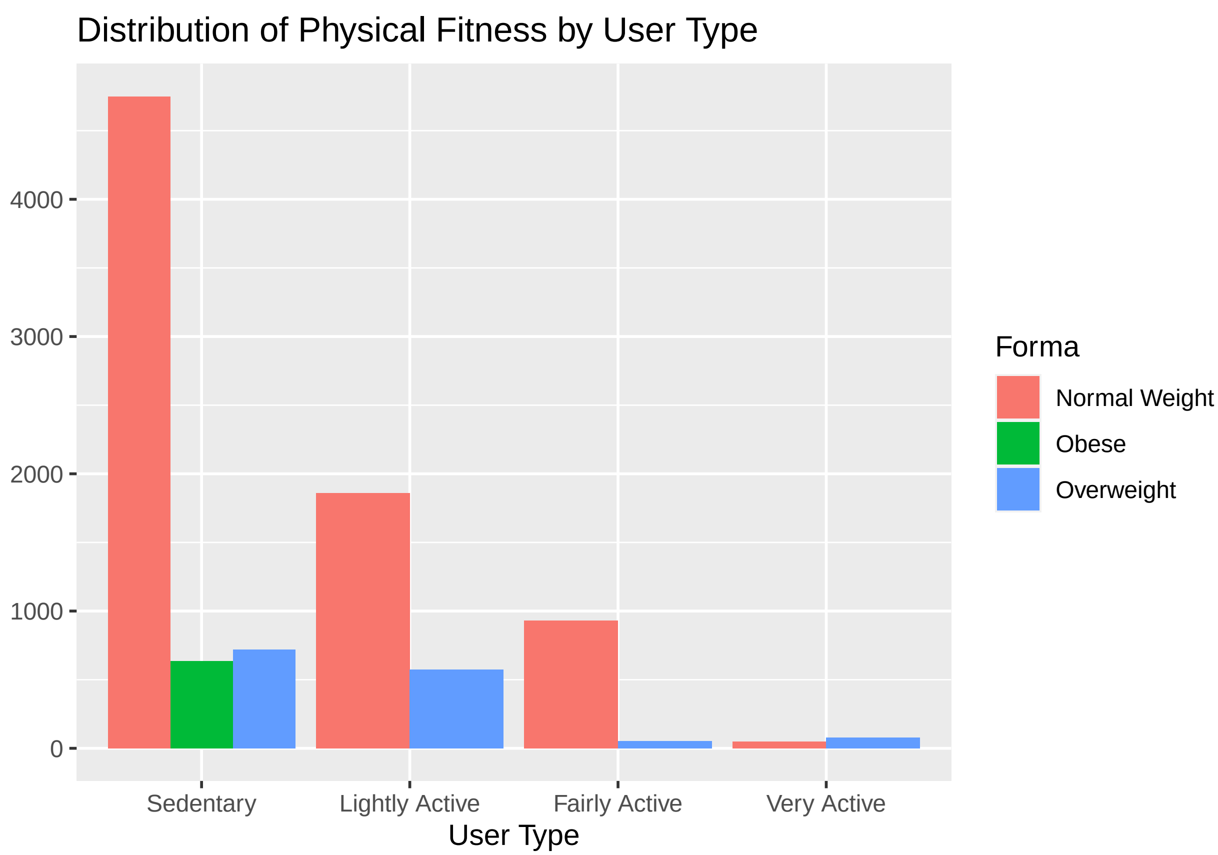
```
sum(is.na(user_fit$Forma))
```

```
## [1] 0
```

Distribution of Physical Fitness by User Type

```
user_fit %>%
  group_by(user_type, Forma) %>%
  summarise(Count = n()) %>%
  drop_na() %>%
  ggplot(aes(x = user_type, y = Count, fill = Forma)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "User Type", y = NULL , title = "Distribution of Physical Fitness by User Type")
```

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

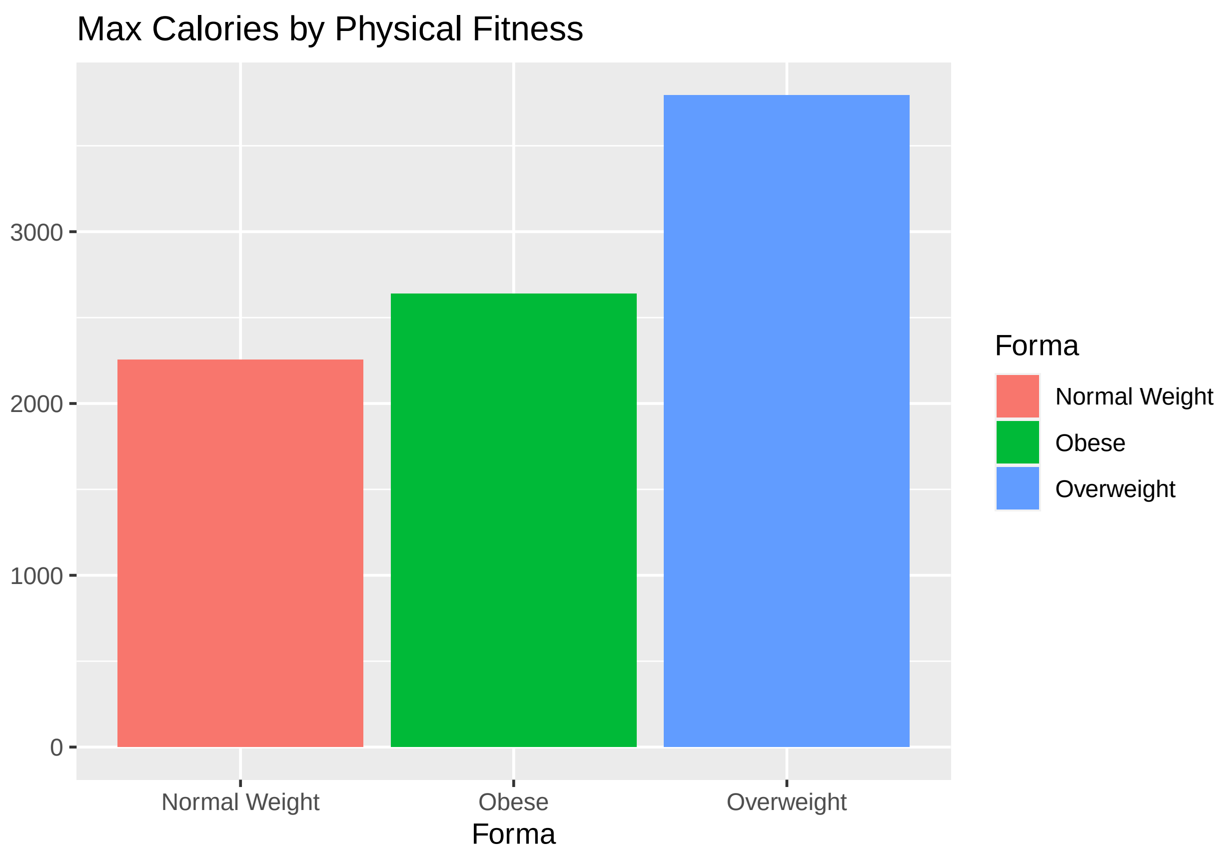


The chart illustrates that the sedentary group is predominantly composed of normal-weight individuals, with fewer than 1/3 being obese and others being overweight. On the other hand, the very active group has some individuals with normal weight and some who are overweight.

Total Calories by Physical Fitness

```
user_fit %>%
  group_by(Forma) %>%
  summarise(MaxCalories = max(Calories)) %>%
```

```r
ggplot(aes(x = Forma, y = MaxCalories, fill = Forma)) +
geom_bar(stat = "identity") +
labs(x = "Forma", y = NULL, title = "Max Calories by Physical Fitness")
```



Surprisingly, the chart reveals that overweight individuals are the ones consuming the most calories, exceeding 3500 calories. Additionally, obese individuals rank second in terms of calorie consumption, averaging around 2500 calories. Meanwhile, normal-weight individuals consume the least, slightly above 2200 calories.

```r
is.Date(user_fit$Date.x)
```

```
## [1] FALSE
```

```r
user_fit$Date.x <- as.Date(user_fit$Date.x, format="%m/%d/%Y")
is.Date(user_fit$Date.x)
```

```
## [1] TRUE
```

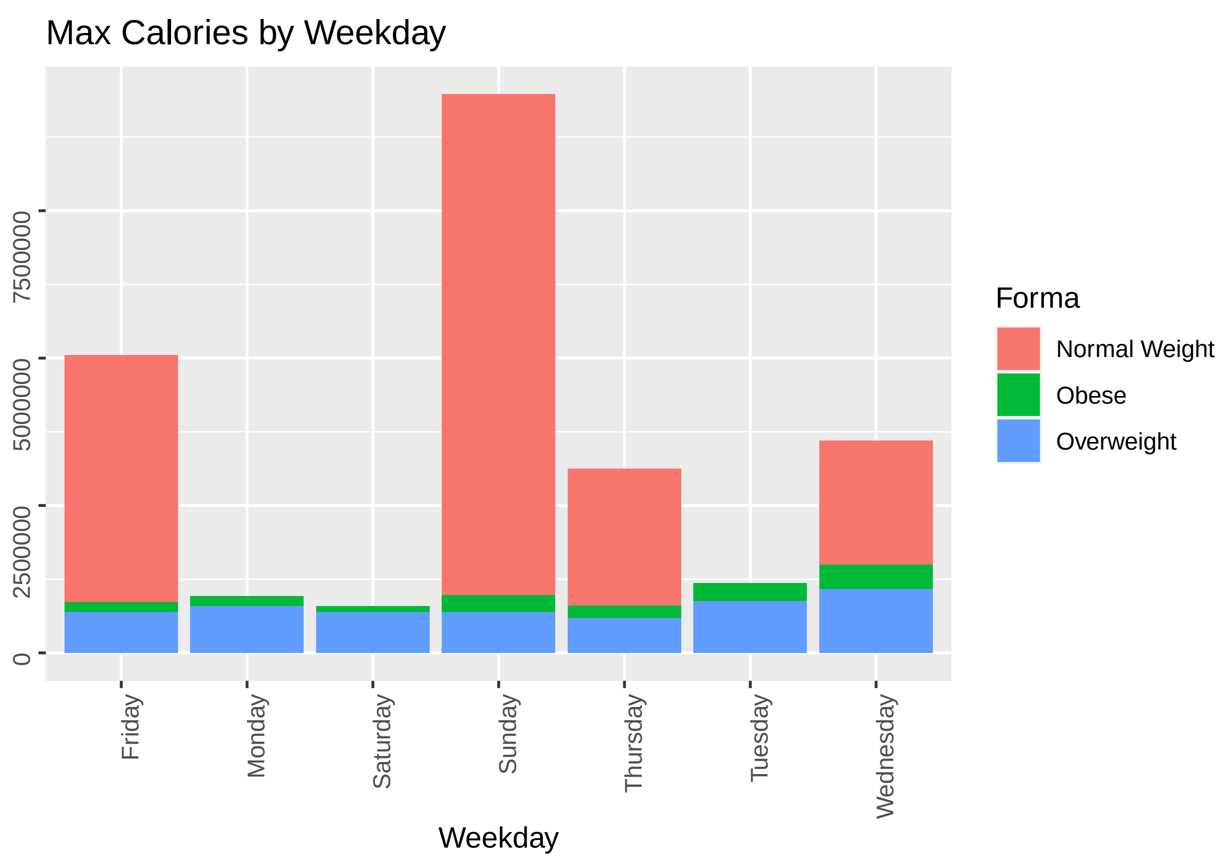
```r
head(user_fit$Date.x)
```

```
## [1] "2016-04-21" "2016-04-21" "2016-04-21" "2016-04-21" "2016-04-21"
## [6] "2016-04-21"
```

```r
user_fit <- user_fit %>%
  mutate("Weekday" = weekdays(Date.x))
```

Max Calories by Weekday

```r
user_fit %>%
  group_by(Forma) %>%
  reframe(MaxCalories = max(Calories, na.rm = TRUE), Weekday) %>%
  ggplot(aes(x = Weekday, y = MaxCalories, fill = Forma)) +
  geom_bar(stat = "identity") +
  labs(x = "Weekday", y = NULL, title = "Max Calories by Weekday") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1), axis.text.y = element_text(angle = 90, hjust
```



The chart highlights that Sunday is the day when normal-weight individuals typically consume more calories. On the other hand, overweight and obese individuals have an average maximum calorie consumption throughout the week, with a slightly higher amount on Wednesday.

Share

General Comment

Based on the recent analyses, it appears that there are interesting patterns in calorie consumption and physical fitness among different user types. Surprisingly, overweight individuals tend to have the highest maximum calorie consumption, while normal-weight individuals consume more on Sundays. Additionally, the sedentary group is mainly composed of normal-weight individuals, while the very active group shows variation with both normal-weight and overweight individuals. These insights provide valuable information for understanding the relationships between lifestyle factors, caloric intake, and physical fitness among the studied groups.

Reccomendations for the Marketing Team

- Utilize the ongoing pattern of sustained involvement to present tailored observations and suggestions. Bellabeat has the opportunity to position its products as supportive companions, delivering practical guidance derived from users' monitored data. This approach enhances the overall user experience, aiding individuals in improving their physical fitness and BMI.
- Enhanced Notification Strategy: Explore the optimization of notification tactics by discerning the peak usage periods of each user. Delivering timely and pertinent notifications aligned with users' peak activity times or when they are most likely to interact with their devices can significantly boost communication effectiveness. This encourages users to continually outperform their previous achievements.
- Community Engagement and Challenges: Encourage community engagement by incorporating features that allow users to share achievements or challenges with others. Creating a sense of community around Bellabeat products can foster a supportive environment and enhance the overall user experience.

Conclusion

In conclusion, this case study has offered valuable insights into the usage of smart devices. The in-depth analysis and findings have illuminated user engagement, providing a comprehensive understanding of the challenges and opportunities within this context.