Bellabeat case study 2

Business task

The business task in this case study is to analyze trends going on in the smart device and fitness/wellness app industry to precisely guide Bellabeat's marketing strategy.

The stakeholders for this study would be Urška Sršen and Sando Mur, and the current stockholders or investors in the Bellabeat company.

The main questions to be asked with this analysis are:

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?

Insights obtained from this analysis can indicate what kind of problems or necessities could be directly targeted with bellabeat products, the magnitude of each one of these individual demands, trends in the market over long periods of time, and a deeper more precise description of the target demographic.

Business decisions like the advertisement campaign, the magnitude and proportion of the production of the different products and sales predictions can all be guided with the conclusions of this analysis, and further unexpected insights that could possibly affect other different business decisions may be found.

Description of all data sources used

For this analysis the ""FitBit Fitness Tracker Data" Kaggle data set was used, it 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.

This dataset was generated by respondents to a distributed survey via Amazon Mechanical Turk between 03.12.2016 and 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.

This dataset falls under the copyright license CC0: Public Domain, dataset made available through the Kaggle user Möbius, sourced from the site zenodo.com at:

https://zenodo.org/records/53894#.X9oeh3Uzaao (https://zenodo.org/records/53894#.X9oeh3Uzaao) DOI: 10.5281/zenodo.53894

Beyond this level, the provenance of this data is unverifiable as this is the only information it has been published with. Despite it being dated as 8 years old, it provides valuable information to modern day fitness app usage and habits regarding their users.

It also has a sample size of n = 30, with tables ranging from the sizes of 33 to 8, which is rather small considering it's meant to represent the entire market of fitness and wellness apps.

The datasets are sorted in the long format; with the same IDs taking up multiple rows for different entries and registering in a different value each day.

loading packages

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.3     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.4     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.0
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
errors
```

```
library(lubridate)
library(dplyr)
library(ggplot2)
library(tidyr)
library(janitor)
```

```
## Warning: package 'janitor' was built under R version 4.3.2
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

Loading data

```
activity <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/dailyActivity_merged.csv")
calories <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/dailyCalories_merged.csv")
intensities <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/dailyIntensities_merged.csv")
steps <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/dailySteps_merged.csv")
heartRate <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/heartrate_seconds_merged.csv")
hourlyCalories <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/hourlyCalories_merged.csv")
hourlyIntensities <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/hourlyIntensities_merged.csv")
hourlySteps <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/hourlySteps_merged.csv")
minuteCalories <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/minuteCaloriesNarrow_merged.csv")
minuteIntensities <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/minuteIntensitiesNarrow_merged.csv")
minuteMETS <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/minuteMETsNarrow_merged.csv")
minuteSleep <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/minuteSleep_merged.csv")
minuteSteps <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/minuteStepsNarrow_merged.csv")
dailySleep <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/sleepDay_merged.csv")
weightLog <- read.csv("D:/Usuarios/Casa/Downloads/Case study/archive/Fitabase Data 4.12.16-5.12.16/weightLogInfo_merged.csv")
```

Cleaning and manipulation of data

All 15 tables were originally imported using the read.csv function; and the “date” format was changed and split into 2 variables, date and time.

```
#converting timestamps to date time format and split to date and time

# intensities
hourlyIntensities$ActivityHour=as.POSIXct(hourlyIntensities$ActivityHour, format="%m/%d/%Y %I:%M:%
S %p", tz=Sys.timezone())
hourlyIntensities$time <- format(hourlyIntensities$ActivityHour, format = "%H:%M:%S")
hourlyIntensities$date <- format(hourlyIntensities$ActivityHour, format = "%m/%d/%y")
# hourlySteps
hourlySteps$ActivityHour=as.POSIXct(hourlyIntensities$ActivityHour, format="%m/%d/%Y %I:%M:%S %p",
tz=Sys.timezone())
hourlySteps$time <- format(hourlyIntensities$ActivityHour, format = "%H:%M:%S")
hourlySteps$date <- format(hourlyIntensities$ActivityHour, format = "%m/%d/%y")
# calories
hourlyCalories$ActivityHour=as.POSIXct(hourlyCalories$ActivityHour, format="%m/%d/%Y %I:%M:%S %p",
tz=Sys.timezone())
hourlyCalories$time <- format(hourlyCalories$ActivityHour, format = "%H:%M:%S")
hourlyCalories$date <- format(hourlyCalories$ActivityHour, format = "%m/%d/%y")
# activity
activity$ActivityDate=as.POSIXct(activity$ActivityDate, format="%m/%d/%Y", tz=Sys.timezone())
activity$date <- format(activity$ActivityDate, format = "%m/%d/%y")
# sleep
dailySleep$SleepDay=as.POSIXct(dailySleep$SleepDay, format="%m/%d/%Y %I:%M:%S %p", tz=Sys.timezone
())
dailySleep$date <- format(dailySleep$SleepDay, format = "%m/%d/%y")
```

Eliminating duplicates

```
#checking for duplicates
sum(duplicated(activity))
```

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

```
sum(duplicated(dailySleep))
```

```
## [1] 3
```

```
sum(duplicated(hourlySteps))
```

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

```
#removing duplicates for daily sleep
dailySleep <- dailySleep %>%
  distinct() %>%
  drop_na()

#verifying that the duplicates have been removed
sum(duplicated(dailySleep))
```

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

Analysis summary

Using the distinct function, we can see how many individual users are in each table.

```
#using the disctinct function to tell how many individual IDs or users there are in each dataset.
n_distinct(activity$Id)
```

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

```
n_distinct(calories$Id)
```

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

```
n_distinct(intensities$Id)
```

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

```
n_distinct(dailySleep$Id)
```

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

```
n_distinct(weightLog$Id)
```

```
## [1] 8
```

```
n_distinct(steps$Id)
```

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

We can also take a look at the summary from some of these tables.

```
#summary of some of the tables

#activity
activity %>%
  select() %>%
  summary()
```

```
## < table of extent 0 x 0 >
```

```
# calories
hourlyCalories %>%
  select(Calories) %>%
  summary()
```

```
##     Calories
##  Min.   : 42.00
##  1st Qu.: 63.00
##  Median : 83.00
##  Mean   : 97.39
##  3rd Qu.:108.00
##  Max.   :948.00
```

```
# sleep
dailySleep %>%
  select(TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed) %>%
  summary()
```

```
##  TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
##  Min.   :1.00      Min.   : 58.0      Min.   : 61.0
##  1st Qu.:1.00      1st Qu.:361.0      1st Qu.:403.8
##  Median :1.00      Median :432.5      Median :463.0
##  Mean   :1.12      Mean   :419.2      Mean   :458.5
##  3rd Qu.:1.00      3rd Qu.:490.0      3rd Qu.:526.0
##  Max.   :3.00      Max.   :796.0      Max.   :961.0
```

```
# weight
weightLog %>%
  select(WeightKg, BMI) %>%
  summary()
```

```
##     WeightKg           BMI
##  Min.   : 52.60   Min.   :21.45
##  1st Qu.: 61.40   1st Qu.:23.96
##  Median : 62.50   Median :24.39
##  Mean   : 72.04   Mean   :25.19
##  3rd Qu.: 85.05   3rd Qu.:25.56
##  Max.   :133.50   Max.   :47.54
```

```
# steps
steps %>%
  select(Id, StepTotal) %>%
  summary()
```

```
##        Id              StepTotal
##  Min.   :1.504e+09   Min.   :    0
##  1st Qu.:2.320e+09   1st Qu.: 3790
##  Median :4.445e+09   Median : 7406
##  Mean   :4.855e+09   Mean   : 7638
##  3rd Qu.:6.962e+09   3rd Qu.:10727
##  Max.   :8.878e+09   Max.   :36019
```

Now, I'm going to merge a couple tables together to compare different variables.

```
#merging the tables sleep and activity by the variables "ID" and "Date"
merged_data <- merge(dailySleep, activity, by=c('Id', 'date'))

#making an table of the average daily steps, calories and sleep
daily_average <- merged_data %>%
  group_by(Id) %>%
  summarise (mean_daily_steps = mean(TotalSteps), mean_daily_calories = mean(Calories), mean_daily
_sleep = mean(TotalMinutesAsleep))
```

According to a 2004 study conducted in Arizona State University, the classification for daily activity in adults are as follows:

<5000 steps/day: sedentary

5000-7499 steps/day: low active

7500-9999 steps/day: somewhat active

10000 < 12499 steps/day: active

12500< steps/day: highly active

Tudor-Locke C, Bassett DR Jr. How many steps/day are enough? Preliminary pedometer indices for public health. Sports Med. 2004;34(1):1-8. doi: 10.2165/00007256-200434010-00001. PMID: 14715035. https://pubmed.ncbi.nlm.nih.gov/14715035/ (https://pubmed.ncbi.nlm.nih.gov/14715035/)

According to this information, I can calculate the average daily steps of each user, categorize them on this classifcation and look at the distribution of user types according to daily activity.

```
#introducing a new variable named user type based on daily steps
user_type <- daily_average %>%
  mutate(user_type = case_when(
    mean_daily_steps < 5000 ~ "sedentary",
    mean_daily_steps >= 5000 & mean_daily_steps < 7500 ~ "low active",
    mean_daily_steps >= 7500 & mean_daily_steps < 10000 ~ "somewhat active",
    mean_daily_steps >= 10000 & mean_daily_steps < 12500 ~ "active",
    mean_daily_steps >= 12500 ~ "highly active"
  ))
```

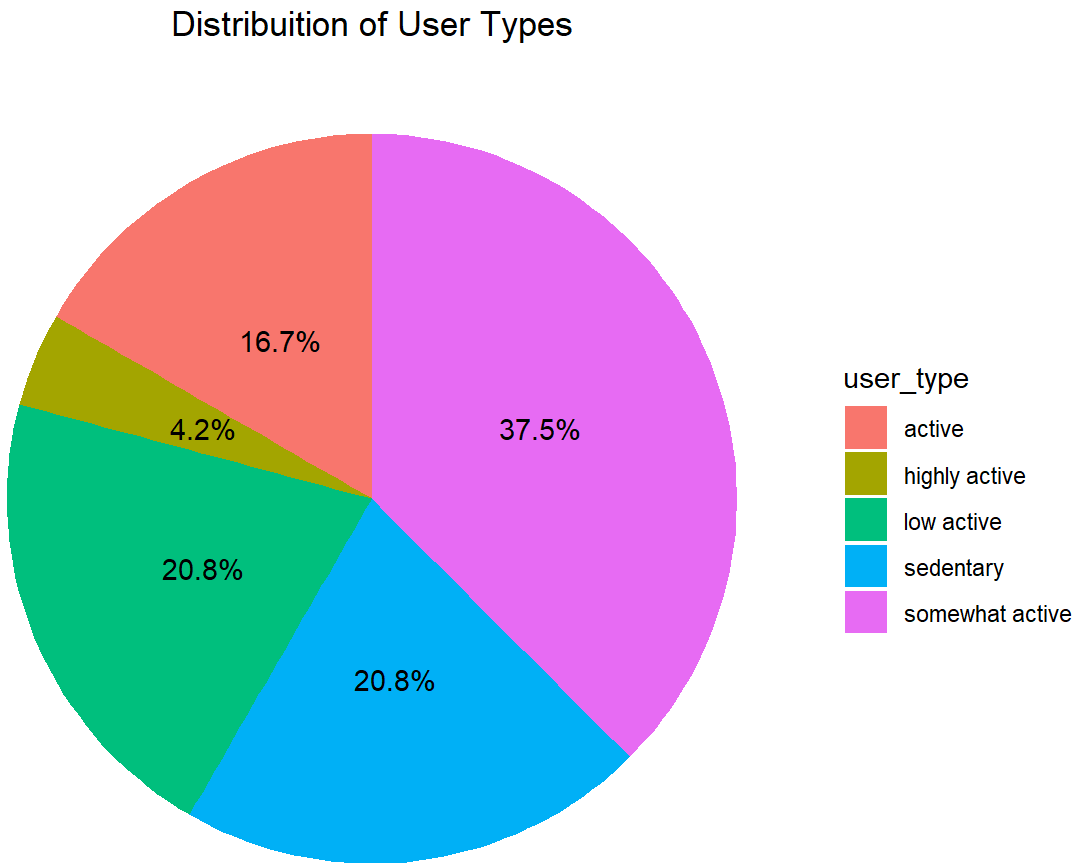
Unfortunately, as there aren't logs for each days on sleep, I couldn't take a good look at the patterns that surround every individual day of the week; but it's an interesting insight to try to achieve for a future analyisis with richer data.

Visualizations

Distribuition of user types based on daily activities

```
#plotting user type pie chart (based on daily steps
ggplot(data = user_type, aes(x = "", fill = user_type)) +
  geom_bar(width = 1, stat = "count") +
  coord_polar(theta = "y") +
  geom_text(aes(label = scales::percent(..count.. / sum(..count..))), stat = "count", position = p
osition_stack(vjust = 0.5)) +
  labs(title = "Distribuition of User Types") +
  theme_void()+
  theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## ℹ Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



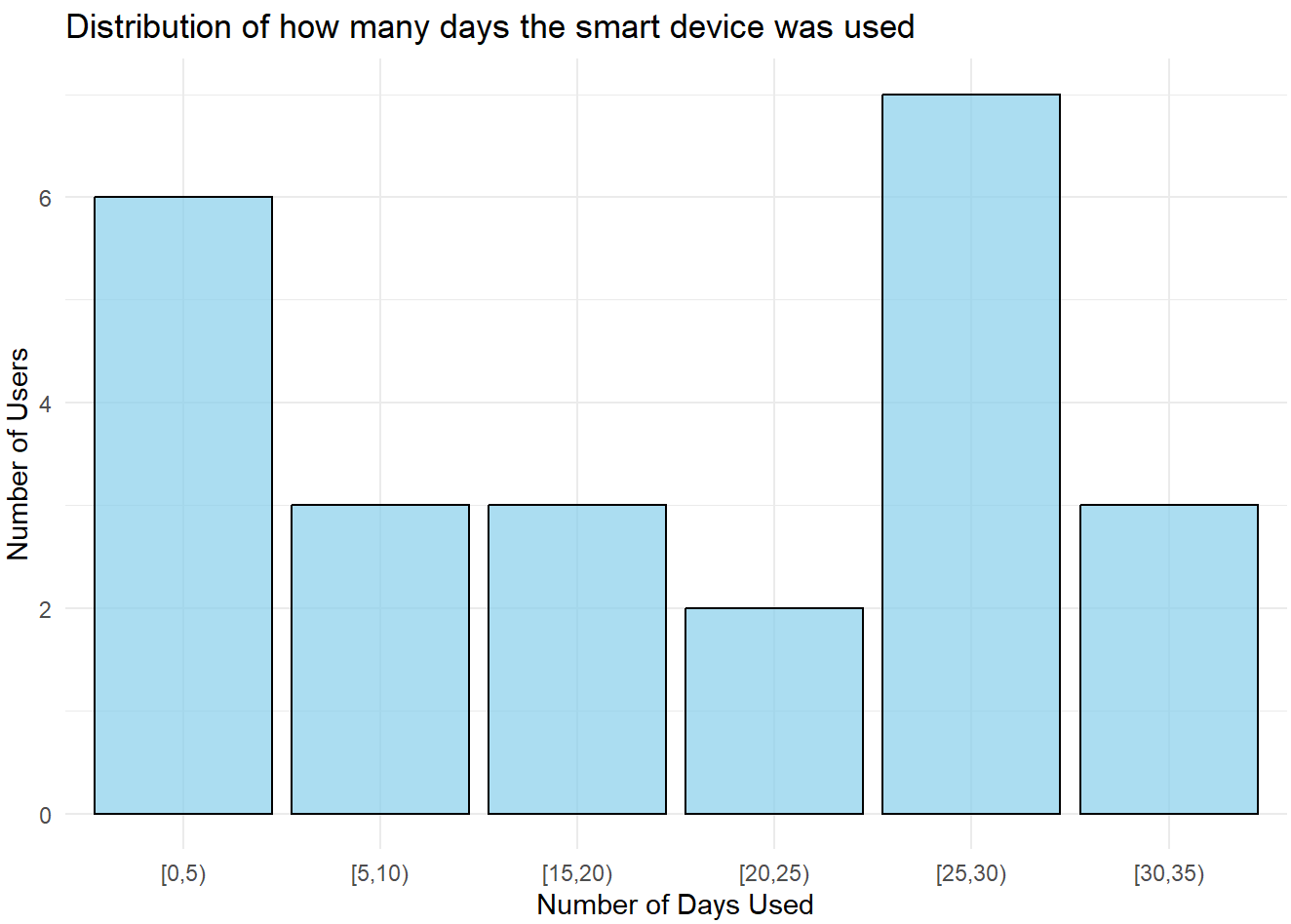
In this table we can see the majority of the userbase falls under the “somewhat active” category, or 7,500-9,999 steps/day, which is lower than the recommended 10,000 daily steps for the majority of the population; along with “sedentary” and “low active” Only 20.9% of the userbase surpass this number.

Distribuition of user types based on daily app usage

```
#calculate how many days each user uses the app
daily_use <- merged_data %>%
  group_by(Id) %>%
  summarize(days_used=sum(n()))

#dividing  daily use into brackets  of 5 days
daily_use <- daily_use %>%
  mutate(days_range = cut(days_used, breaks = seq(0, max(days_used) + 5, by = 5), right = FALSE))

# Plot the daily use
ggplot(data = daily_use, aes(x = days_range)) +
  geom_bar(fill = 'skyblue', color = 'black', alpha = 0.7) +
  labs(title = "Distribution of how many days the smart device was used",
       x = "Number of Days Used",
       y = "Number of Users") +
  theme_minimal()
```

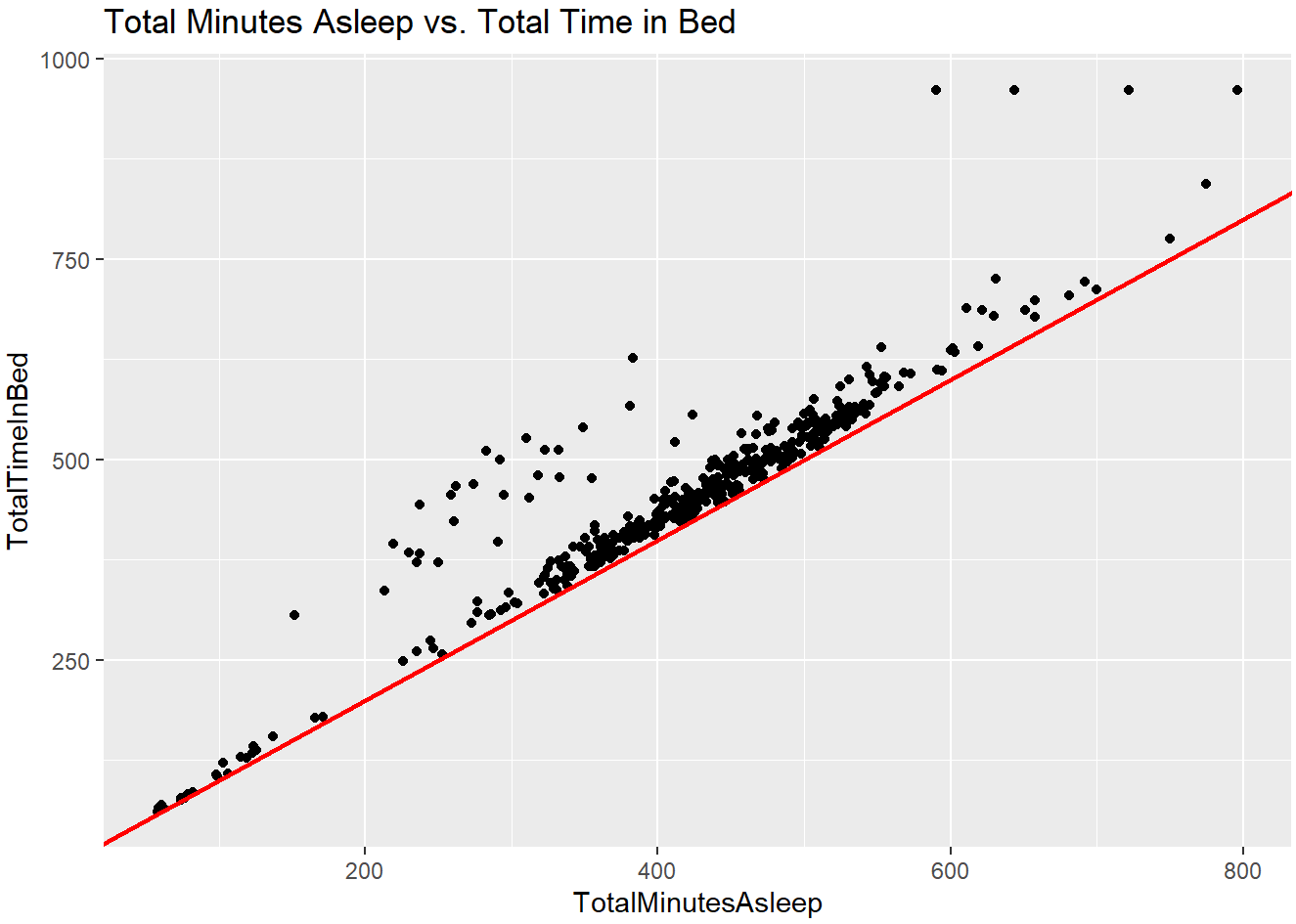


There isn't a visible bias that can be seen here but you can observe there is a great concentration towards the extremes; users tend to either rarely use the app or use it most days of the month.

Daily sleep vs minutes on bed

```
#plotting minutes in bed and minutes asleep with a vertical line of a slope of 1
ggplot(data = dailySleep, aes(x = TotalMinutesAsleep, y = TotalTimeInBed)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, linetype = "solid", color = "red", size = 1) +
  labs(title = "Total Minutes Asleep vs. Total Time in Bed")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

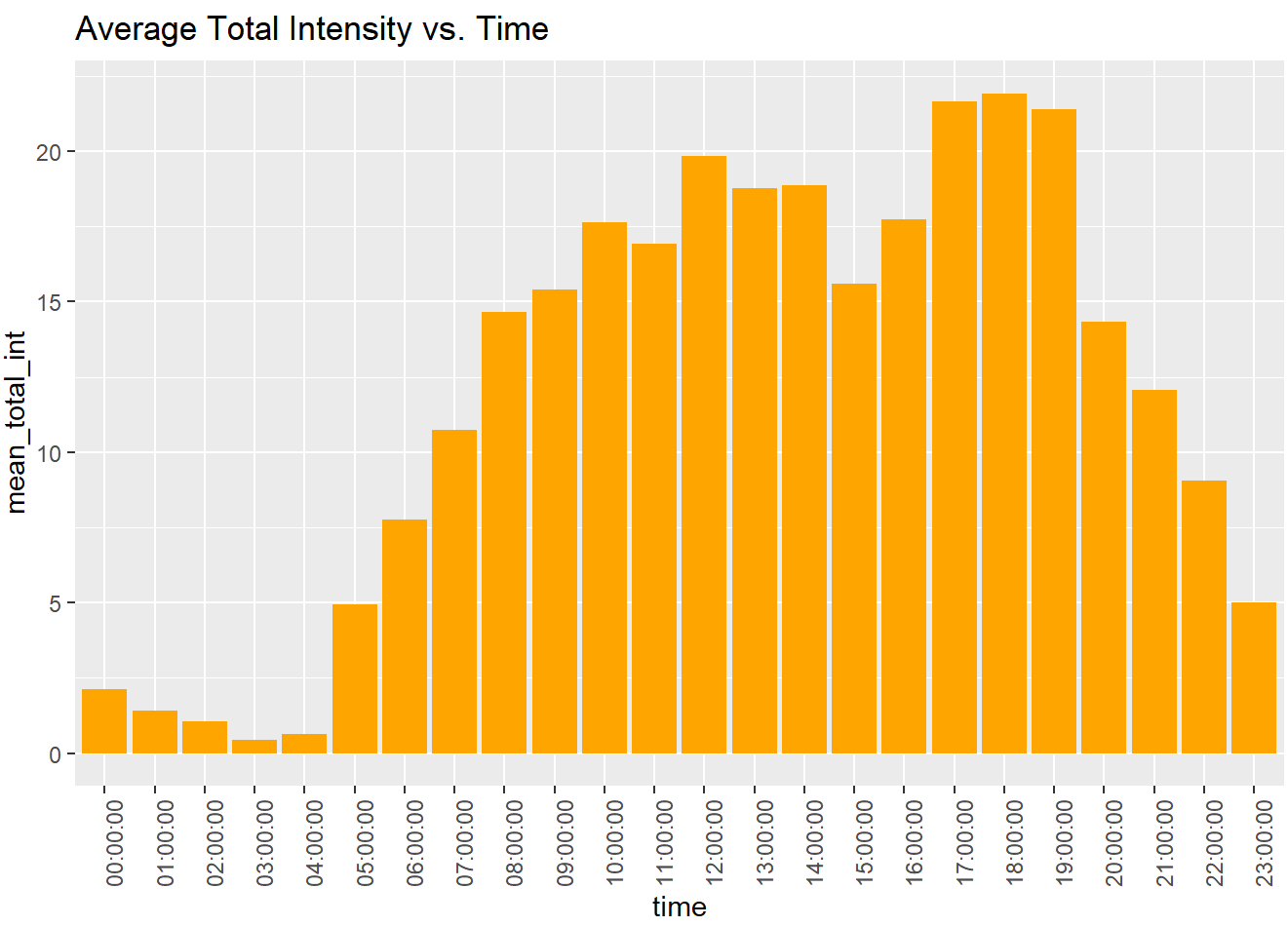


In this graph, the red line represents a function with a slope of 1. This shows how it is clearly impossible to have more minutes asleep as time in bed, as you can't sleep if you're not in bed. Now, digging deeper, you can seee most people fall asleep a couple minutes after laying down, but some spend up to hours in bed before falling asleep.

```
#plotting intensities vs time
int_new <- hourlyIntensities %>%
  group_by(time) %>%
  drop_na() %>%
  summarise(mean_total_int = mean(TotalIntensity))

ggplot(data=int_new, aes(x=time, y=mean_total_int)) + geom_histogram(stat = "identity", fill='oran
ge') +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(title="Average Total Intensity vs. Time")
```

```
## Warning in geom_histogram(stat = "identity", fill = "orange"): Ignoring unknown
## parameters: `binwidth`, `bins`, and `pad`
```



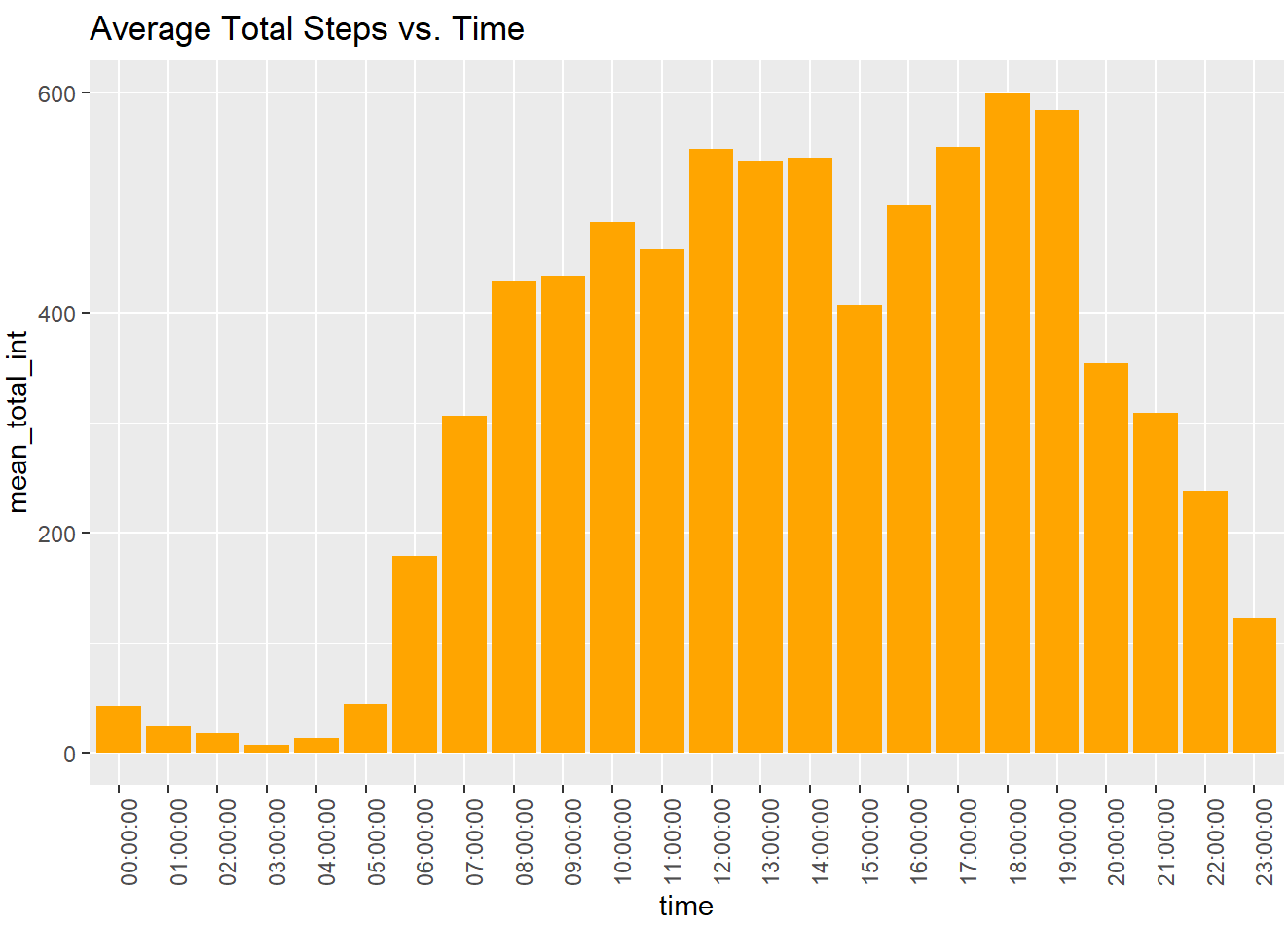
Most active minutes occur between 17:00 and 19:00, or 5:00 PM and 7:00 PM, when people are leaving from work or just done with it. After 7:00 Pm there is a strong decline in physical activity.

from 10:00 to 14:00 or 10:00 AM to 2:00 Pm there is not as high of a peak but a more consistent high intensity period of time.

```
#plotting time of the day and hourlysteps
int_new <- hourlySteps %>%
  group_by(time) %>%
  drop_na() %>%
  summarise(mean_total_int = mean(StepTotal))

ggplot(data=int_new, aes(x=time, y=mean_total_int)) + geom_histogram(stat = "identity", fill='oran
ge') +
  theme(axis.text.x = element_text(angle = 90)) +
  labs(title="Average Total Steps vs. Time")
```

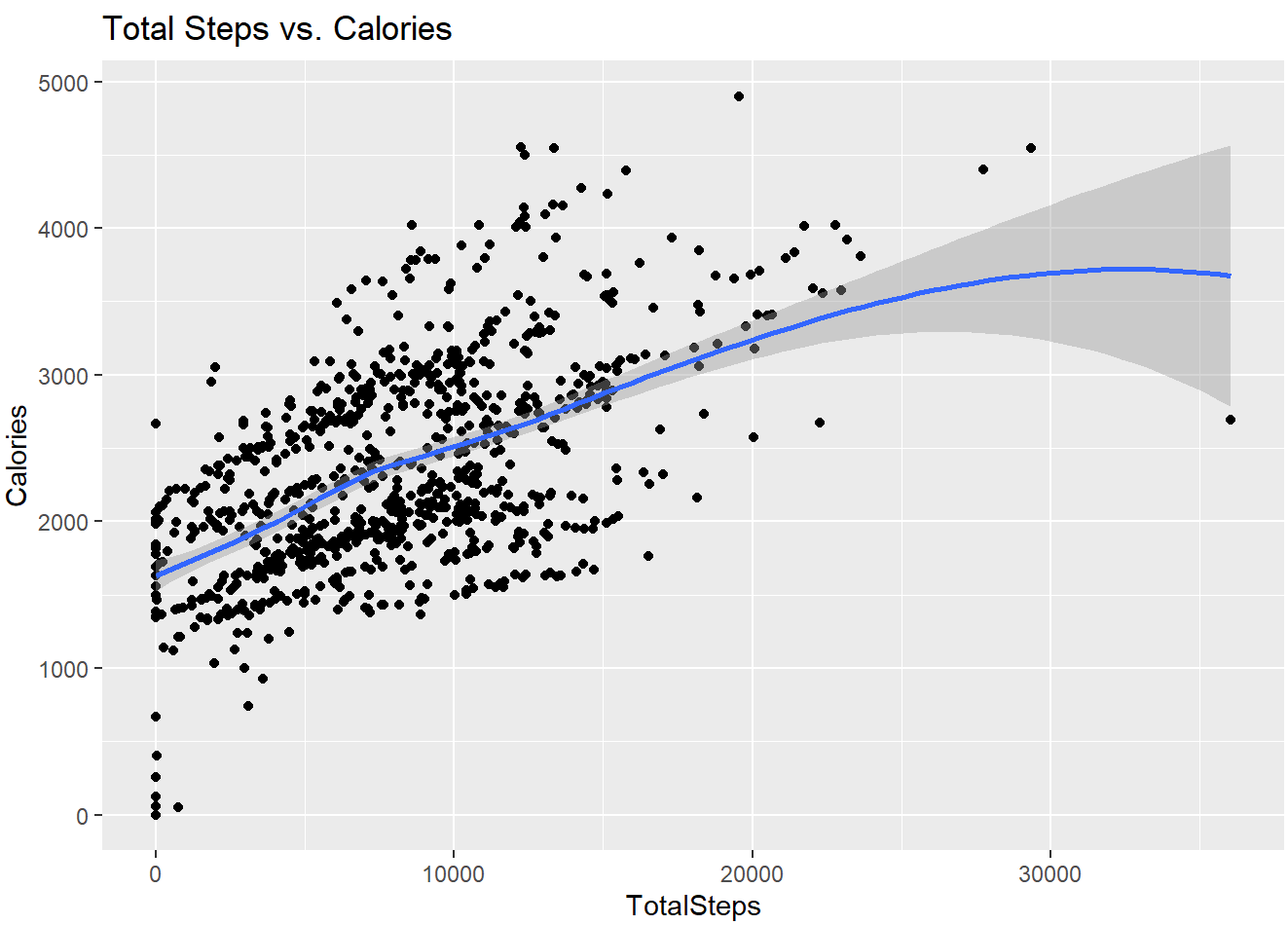
```
## Warning in geom_histogram(stat = "identity", fill = "orange"): Ignoring unknown
## parameters: `binwidth`, `bins`, and `pad`
```



Since this graph is a very similar replication of the aforementioned, we can deduce most of this physical intensity comes from walking. Either to or from work.

```
#plotting activity and calories consumed
ggplot(data=activity, aes(x=TotalSteps, y=Calories)) +
  geom_point() + geom_smooth() + labs(title="Total Steps vs. Calories")
```

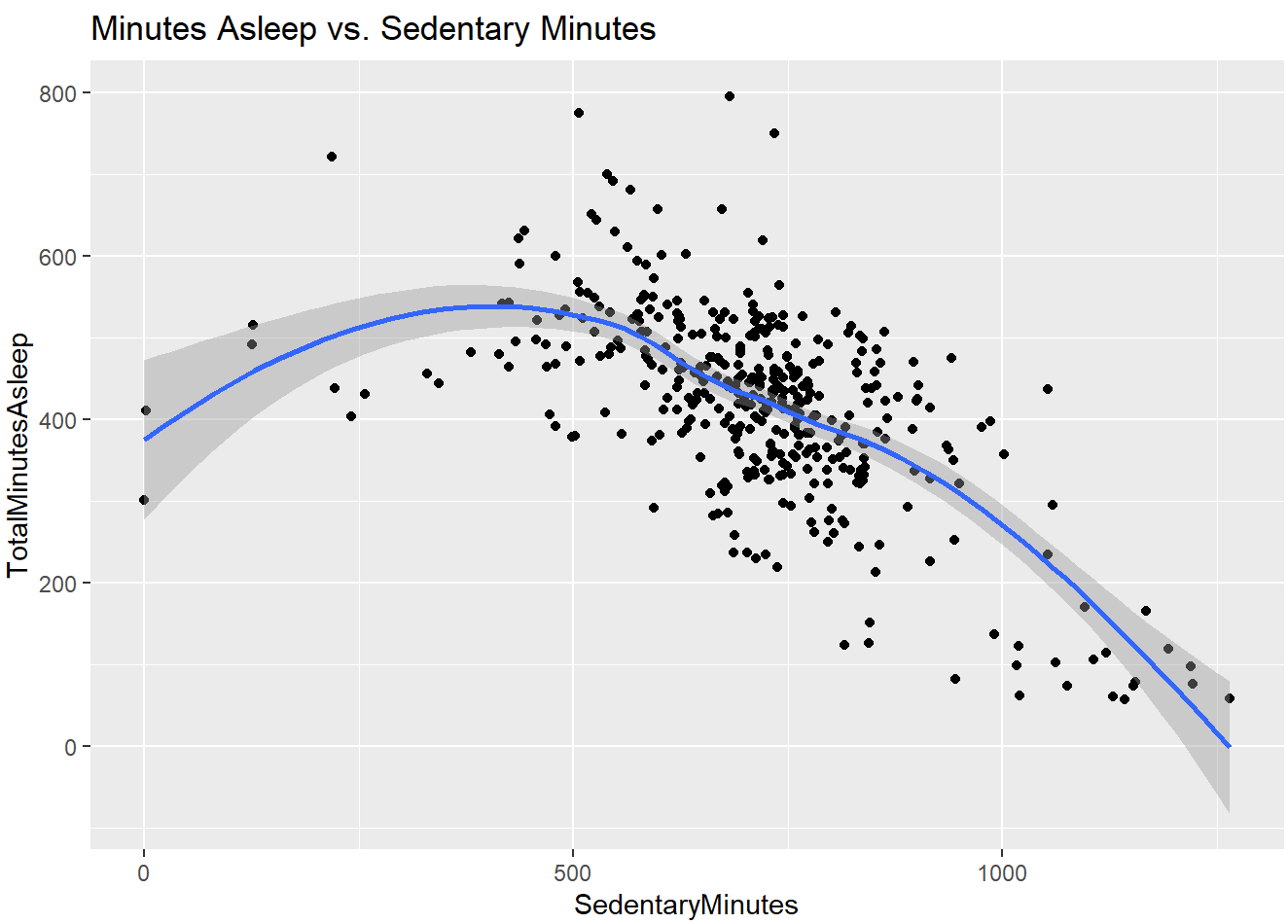
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



A not very strong positive correlation can be seen with the total steps in a day and total calories burnt in a day.

```
#plotting total minutes asleep and sedentary minutes
ggplot(data=merged_data, aes(x=SedentaryMinutes, y=TotalMinutesAsleep)) +
  geom_point(color='black') + geom_smooth() +
  labs(title="Minutes Asleep vs. Sedentary Minutes")
```

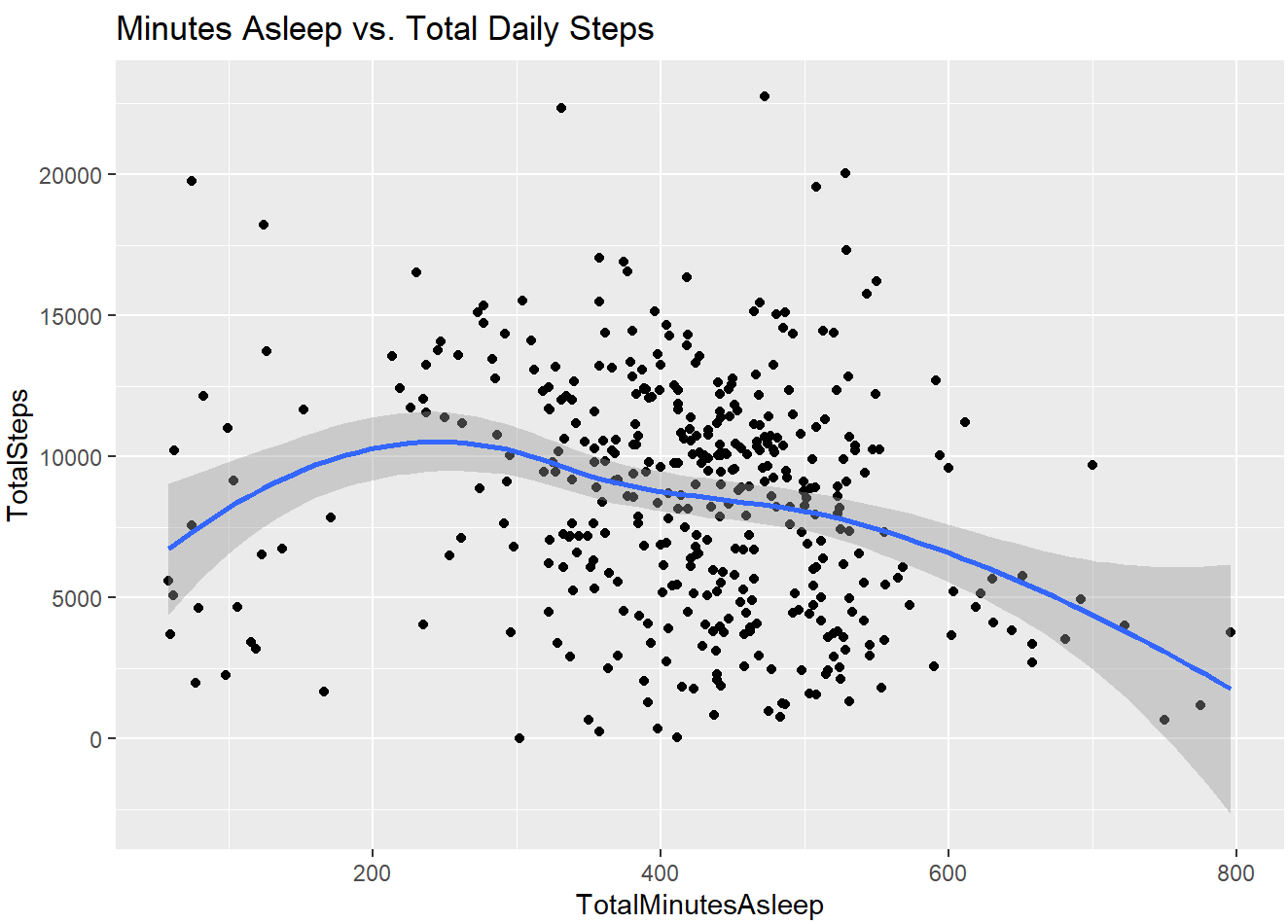
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Again, we can see a steeper but weaker correlation with the total amount of time asleep and the sedentary minutes; the more sedentary someone is, the less sleep they have. It's important to note here not to confuse correlation with causation, as there may be external factors at play that are causational for both of these.

```
#plotting total minutes asleep and steps
ggplot(data=merged_data, aes(x=TotalMinutesAsleep, y=TotalSteps)) +
  geom_point(color='black') + geom_smooth() +
  labs(title="Minutes Asleep vs. Total Daily Steps")
```

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



If make a similar comparison, this time with steps instead of calories consumed in a day, the correlation becomes less steep, and less strong. But it is still there.

Recommendations

It is clear how important total daily steps are for daily calories burnt, daily intensities and even daily sleep. The app should implement some way of having **notifications that recommend its users to try getting a higher number of total daily steps** maybe through getting down one bus stop earlier or just walking to certain closeby places and saving some gas too; especially since **most users are less physically active than they should**.

Given how **steps and phyisical activity seem to contribute to sleep quality**, some kind of **mechanism developed for users with poor sleep** that works through specific recommendations and notifications, could also be thought out, since the app collects enough data to tell which specific users both don't get good enough sleep and good enough physical activity throughout the day; this mechanism could take into account physical activity, daily steps, or time in bed using their phones without sleeping, to pin down the specific reason this poor sleep quality is going on.

In addition, there is a larger portion of the userbase that regularly utilizes the smart device, but there is also a smaller but **significant portion that only uses the device for less than 10 days**, so given how many people are using the smart device for this little time, there may be reason to make some mechanism that ensures this device is being used through notifications or gamefications, having monthly goals for device usage.