

Bellabeat Marketing Strategy

Ask

Bellabeat a company providing fitness products such as Bellabeat app, Leaf, watch , water bottle and personal guidance wants to access the current trends in the usage of fitness devices. Their main business task to be addressed is developing a marketing strategy for Bellabeat products. This marketing strategy is proposed to be developed on the basis of the current usage of trends provided by users of another fitness device named “Fitbit”.

Questions

Questions to be mainly answered from this analysis are:

1. What are the current trends in smart device usage?
2. How could these trends apply to Bellabeat customers?
3. How could these trends influence Bellabeat Marketing strategy?

Data source

Fitbit user database provided by Mobius.This dataset was generated by respondents to a distributed survey via Amazon Mechanical Turk. This database consists of daily usage data of 33 users for the period April 12, 2016- May 12, 2016. These users consented to the submission of personal tracker data. This data includes minute-level output for physical activity, heart rate, and sleep monitoring. It includes information about daily activity, steps, and heart rate.

Prepare

This data was downloaded from a Kaggle data set provided by Mobius. The data is organized as a set of 18 csv files. The users consented to provide their personal fitness data and it hides individual information. Mainly 5 data sets have been used here. They are:

1. Daily activity of users
2. Hourly intensities of users
3. Hourly steps of users
4. User sleep data
5. Heart rate per seconds of recorded user activity

Process

Using RStudio for cleaning data. This tool is being user here as it can help in cleaning, analyzing and visualizing data. Rstudio is also be used for documentation.

Cleaning data

The following packages were installed in R to read and clean data:

- 1.tidyverse
- 2.tidyr
- 3.dplyr
- 4.lubridate

Steps followed in cleaning:

1. Data checked for any null values.
2. Type of each column in the dataset is checked to make sure it is compatible with analysis and visualization.

Analyze

Data was analyzed using:

1. Histograms
2. Pivot tables
3. Heat maps
4. Column graphs

Share

The analyzed data are simultaneously visualized below. The key observations from each visualization are noted below them.

Combined Cleaning, Analyzing and Visualization Step

The process adopted for cleaning, analyzing and visualizing are explained along with their corresponding code chunks

A set of basic packages were installed and loaded in R

```
library(readr) # reads a csv file
library(tidyverse) # for data cleaning and processing

## — Attaching packages ————— tidyverse 1.3.1 —

## ✓ ggplot2 3.3.6      ✓ dplyr    1.0.9
## ✓ tibble   3.1.7      ✓ stringr  1.4.0
## ✓ tidyr    1.2.0      ✓forcats  0.5.1
## ✓ purrr   0.3.4

## — Conflicts ————— tidyverse_conflicts() —
## ✘ dplyr::filter() masks stats::filter()
## ✘ dplyr::lag()   masks stats::lag()

library(tidyr) # for data cleaning and processing
library(dplyr) # for data processing
library(lubridate) # date and time processing

##
## Attaching package: 'lubridate'

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

```
library(yarrr) # statistical functions

## Loading required package: jpeg

## Loading required package: BayesFactor

## Loading required package: coda

## Loading required package: Matrix

## 
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyR':
## 
##     expand, pack, unpack

## ****
## Welcome to BayesFactor 0.9.12-4.3. If you have questions, please contact Richard M
## oreay (richarddmorey@gmail.com).
##
## Type BFManual() to open the manual.
## *****

## Loading required package: circlize

## =====
## circlize version 0.4.14
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: https://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
## in R. Bioinformatics 2014.
##
## This message can be suppressed by:
## suppressPackageStartupMessages(library(circlize))
## =====

## yarrr v0.1.5. Citation info at citation('yarrr'). Package guide at yarrr.guide()

## Email me at Nathaniel.D.Phillips.is@gmail.com

## 
## Attaching package: 'yarrr'
```

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

```
library(ggplot2)#graphical functions
```

Understanding daily activity and user behavior

The file dailyActivity_merged contains daily data. An initial inspection of data is done.

```
daily_activity<-read_csv("Fitbit_data/dailyActivity_merged.csv") #reading file and assigning it to a dataframe
```

```
## 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.
```

```
head(daily_activity) # show top few rows
```

```
## # A tibble: 6 × 15  
##       Id ActivityDate TotalSteps TotalDistance TrackerDistance LoggedActivitie...  
##   <dbl> <chr>        <dbl>        <dbl>        <dbl>        <dbl>  
## 1 1.50e9 4/12/2016      13162        8.5        8.5        0  
## 2 1.50e9 4/13/2016      10735       6.97       6.97        0  
## 3 1.50e9 4/14/2016      10460       6.74       6.74        0  
## 4 1.50e9 4/15/2016      9762        6.28       6.28        0  
## 5 1.50e9 4/16/2016      12669       8.16       8.16        0  
## 6 1.50e9 4/17/2016      9705        6.48       6.48        0  
## # ... with 9 more variables: VeryActiveDistance <dbl>,  
## #   ModeratelyActiveDistance <dbl>, LightActiveDistance <dbl>,  
## #   SedentaryActiveDistance <dbl>, VeryActiveMinutes <dbl>,  
## #   FairlyActiveMinutes <dbl>, LightlyActiveMinutes <dbl>,  
## #   SedentaryMinutes <dbl>, Calories <dbl>
```

```
glimpse(daily_activity) # gives a glimpse of the data types, number of columns
```

```

## Rows: 940
## Columns: 15
## $ Id <dbl> 1503960366, 1503960366, 1503960366, 150396036...
## $ ActivityDate <chr> "4/12/2016", "4/13/2016", "4/14/2016", "4/15/...
## $ TotalSteps <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 13019...
## $ TotalDistance <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8...
## $ TrackerDistance <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8...
## $ LoggedActivitiesDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ VeryActiveDistance <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.5...
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.3...
## $ LightActiveDistance <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.0...
## $ SedentaryActiveDistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ VeryActiveMinutes <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4...
## $ FairlyActiveMinutes <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21...
## $ LightlyActiveMinutes <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 205, ...
## $ SedentaryMinutes <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775, 818...
## $ Calories <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203...

```

The glimpse() function shows the type of the data. Here Id (User Id) is considered a “double” and ActivityDate (Date) is considered character. Changing the type of column in dataframe for cleaning.

```

# dropping n/a values
daily_activity%>%
  drop_na()%>%
  str(daily_activity) # gives data types in column

```

```

## #tibble [940 x 15] (S3:tbl_df/tbl/data.frame)
## $ Id : num [1:940] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
## $ ActivityDate : chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
## $ TotalSteps : num [1:940] 13162 10735 10460 9762 12669 ...
## $ TotalDistance : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
## $ TrackerDistance : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
## $ LoggedActivitiesDistance: num [1:940] 0 0 0 0 0 0 0 0 0 ...
## $ VeryActiveDistance : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...
## $ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...
## $ LightActiveDistance : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...
## $ SedentaryActiveDistance : num [1:940] 0 0 0 0 0 0 0 0 0 ...
## $ VeryActiveMinutes : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...
## $ FairlyActiveMinutes : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...
## $ LightlyActiveMinutes : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...
## $ SedentaryMinutes : num [1:940] 728 776 1218 726 773 ...
## $ Calories : num [1:940] 1985 1797 1776 1745 1863 ...

```

Dropping unnecessary columns

```

daily_activity<-daily_activity%>%
  dplyr:::select(-TrackerDistance,-LoggedActivitiesDistance,-SedentaryActiveDistance)
str(daily_activity)

```

```

## # tibble [940 × 12] (S3: tbl_df/tbl/data.frame)
##   $ Id                      : num [1:940] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##   ...
##   $ ActivityDate            : chr [1:940] "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##   $ TotalSteps               : num [1:940] 13162 10735 10460 9762 12669 ...
##   $ TotalDistance             : num [1:940] 8.5 6.97 6.74 6.28 8.16 ...
##   $ VeryActiveDistance       : num [1:940] 1.88 1.57 2.44 2.14 2.71 ...
##   $ ModeratelyActiveDistance: num [1:940] 0.55 0.69 0.4 1.26 0.41 ...
##   $ LightActiveDistance      : num [1:940] 6.06 4.71 3.91 2.83 5.04 ...
##   $ VeryActiveMinutes         : num [1:940] 25 21 30 29 36 38 42 50 28 19 ...
##   $ FairlyActiveMinutes       : num [1:940] 13 19 11 34 10 20 16 31 12 8 ...
##   $ LightlyActiveMinutes      : num [1:940] 328 217 181 209 221 164 233 264 205 211 ...
##   ...
##   $ SedentaryMinutes          : num [1:940] 728 776 1218 726 773 ...
##   $ Calories                  : num [1:940] 1985 1797 1776 1745 1863 ...

```

Creating a new dataframe with required column type

```

Day_week<-transform(daily_activity,ActivityDate=as.Date(as.character(daily_activity[[2]]),"`%m/%d/%y`"))
Day_week<-transform(Day_week,Id=as.character(as.double(Day_week[[1]])))
str(Day_week)

```

```

## 'data.frame':    940 obs. of  12 variables:
##   $ Id                      : chr  "1503960366" "1503960366" "1503960366" "1503960366" ...
##   ...
##   $ ActivityDate            : Date, format: "2020-04-12" "2020-04-13" ...
##   $ TotalSteps               : num  13162 10735 10460 9762 12669 ...
##   $ TotalDistance             : num  8.5 6.97 6.74 6.28 8.16 ...
##   $ VeryActiveDistance       : num  1.88 1.57 2.44 2.14 2.71 ...
##   $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##   $ LightActiveDistance      : num  6.06 4.71 3.91 2.83 5.04 ...
##   $ VeryActiveMinutes         : num  25 21 30 29 36 38 42 50 28 19 ...
##   $ FairlyActiveMinutes       : num  13 19 11 34 10 20 16 31 12 8 ...
##   $ LightlyActiveMinutes      : num  328 217 181 209 221 164 233 264 205 211 ...
##   $ SedentaryMinutes          : num  728 776 1218 726 773 ...
##   $ Calories                  : num  1985 1797 1776 1745 1863 ...

```

Changing column names to maintaining naming consistency

```

colnames(Day_week)[colnames(Day_week) == "FairlyActiveMinutes"] <- "ModeratelyActiveMinutes"
glimpse(Day_week)

```

```

## Rows: 940
## Columns: 12
## $ Id <chr> "1503960366", "1503960366", "1503960366", "15...
## $ ActivityDate <date> 2020-04-12, 2020-04-13, 2020-04-14, 2020-04...
## $ TotalSteps <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 13019...
## $ TotalDistance <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8...
## $ VeryActiveDistance <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.5...
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.3...
## $ LightActiveDistance <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.0...
## $ VeryActiveMinutes <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4...
## $ ModeratelyActiveMinutes <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21...
## $ LightlyActiveMinutes <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 205, ...
## $ SedentaryMinutes <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775, 818...
## $ Calories <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203...

```

Converting date to day of week

```

Day_week<-transform(Day_week,day_of_week=weekdays(Day_week$ActivityDate))
glimpse(Day_week)

```

```

## Rows: 940
## Columns: 13
## $ Id <chr> "1503960366", "1503960366", "1503960366", "15...
## $ ActivityDate <date> 2020-04-12, 2020-04-13, 2020-04-14, 2020-04...
## $ TotalSteps <dbl> 13162, 10735, 10460, 9762, 12669, 9705, 13019...
## $ TotalDistance <dbl> 8.50, 6.97, 6.74, 6.28, 8.16, 6.48, 8.59, 9.8...
## $ VeryActiveDistance <dbl> 1.88, 1.57, 2.44, 2.14, 2.71, 3.19, 3.25, 3.5...
## $ ModeratelyActiveDistance <dbl> 0.55, 0.69, 0.40, 1.26, 0.41, 0.78, 0.64, 1.3...
## $ LightActiveDistance <dbl> 6.06, 4.71, 3.91, 2.83, 5.04, 2.51, 4.71, 5.0...
## $ VeryActiveMinutes <dbl> 25, 21, 30, 29, 36, 38, 42, 50, 28, 19, 66, 4...
## $ ModeratelyActiveMinutes <dbl> 13, 19, 11, 34, 10, 20, 16, 31, 12, 8, 27, 21...
## $ LightlyActiveMinutes <dbl> 328, 217, 181, 209, 221, 164, 233, 264, 205, ...
## $ SedentaryMinutes <dbl> 728, 776, 1218, 726, 773, 539, 1149, 775, 818...
## $ Calories <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203...
## $ day_of_week <chr> "Sunday", "Monday", "Tuesday", "Wednesday", ...

```

Ordering day of week to get the right visualization

```

Day_week$rev_ord.x <- factor(Day_week$day_of_week, ordered=TRUE, levels = c("Sunday",
"Saturday", "Friday", "Thursday", "Wednesday", "Tuesday", "Monday"))

```

Normalizing the activity in terms of weeks to understand Total steps taken by user in a day

```

Day_week$week_no<-as.numeric(format(Day_week$ActivityDate,"%W"))
Day_week$norm_week<-1+(Day_week$week_no-min(Day_week$week_no))

```

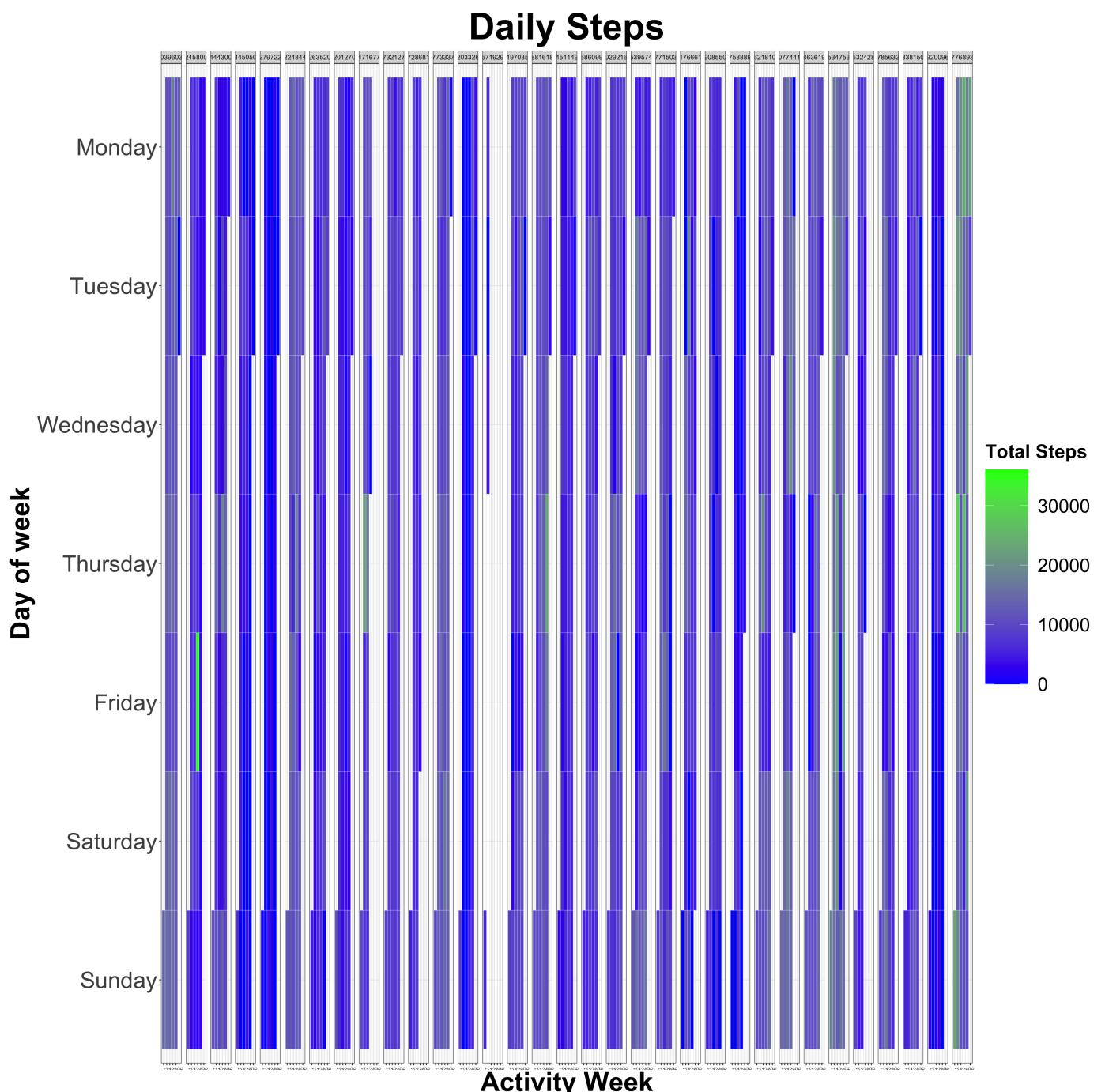
Steps taken by user and their characteristics in terms of days

Generating a heat map representing total steps by each user in a day

```

ggplot(Day_week, aes(x=norm_week, y=rev_ord.x, fill = TotalSteps)) +
  geom_tile(colour = "white") + facet_grid(~Id) + scale_fill_gradient(low="blue", high="green") + xlab("Activity Week") + ylab("Day of week") + ggtitle("Daily Steps") +
  labs(fill = "Total Steps")+
  theme_bw() + theme(axis.text.x = element_text(angle=90,size=7), axis.text.y = element_text(angle=0,size=30),axis.title=element_text(size=35,face="bold"), title = element_text(size=40,face="bold"), legend.text = element_text(size=25), legend.key.size = unit(2, 'cm'),legend.key.height = unit(2, 'cm'), legend.key.width = unit(2, 'cm'),legend.title = element_text(size=25))+
  scale_x_continuous(breaks=round(seq(min(Day_week$norm_week),max(Day_week$norm_week),by=1),1))+theme(plot.title = element_text(hjust = 0.5))

```



Observations:

1. Users very likely to move more or have more steps on a Thursday. 2. Only 1-2 users consistently moved above 30000 steps a day.
3. About ~15 users moved around 10000 steps a day.
4. Most users appear to move significant steps on Sunday, Monday and Thursday.

Breakdown of daily steps are further studied by day of week.

Here, the minimum, maximum and average steps are calculated for each day based on all on all the data provided by the user on that day.

The steps taken on each day are classified to count number of datasets with zero steps on each day, steps less than 10000, steps between 10000 and 20000.

```
# classifying steps and data in terms of days of week
steps_data<-Day_week%>%
  group_by(Days_of_week=rev_ord.x)%>%
  summarise(min_steps=min(TotalSteps),max_steps=max(TotalSteps),avg_steps=mean(TotalSteps),
  steps_lessthan_10k=sum(TotalSteps<10000), steps_morethan_20k=sum(TotalSteps>20000),
  steps_zero=sum(TotalSteps==0), steps_between_10_20k=(sum(TotalSteps>10000) - sum(TotalSteps>20000)))
head(steps_data)
```

```
## # A tibble: 6 × 8
##   Days_of_week min_steps max_steps avg_steps steps_lessthan_10k steps_morethan_...
##   <ord>          <dbl>     <dbl>      <dbl>           <int>           <int>
## 1 Sunday         0        23186    8125.            92              1
## 2 Saturday       0        20500    7781.            80              2
## 3 Friday         0        36019    6933.            88              3
## 4 Thursday       0        29326    8153.            82              6
## 5 Wednesday      0        21727    7448.            93              3
## 6 Tuesday        0        21129    7406.            101             2
## # ... with 2 more variables: steps_zero <int>, steps_between_10_20k <int>
```

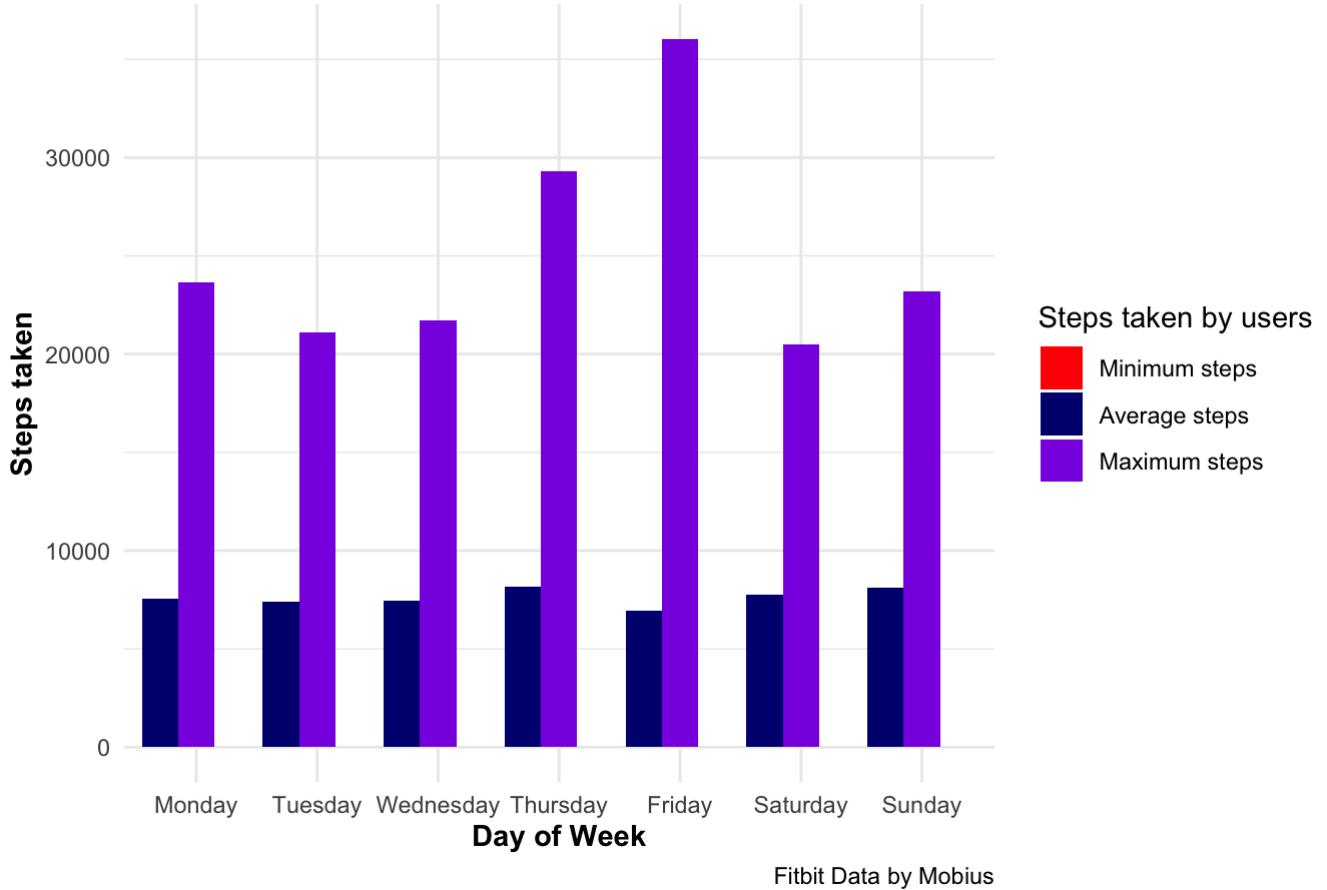
Evaluating minimum, maximum and average steps

```
steps_data_summary<- steps_data %>%
  dplyr::select(-steps_lessthan_10k,-steps_zero,-steps_morethan_20k,-steps_between_10_20k)%>%
  gather(Total, Value,-Days_of_week)
steps_data_summary$Days_of_week<-ordered(steps_data_summary$Days_of_week, levels=c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))
```

Plot for steps

```
ggplot(data=steps_data_summary,aes(x=Days_of_week,y=Value,fill=Total))+
  geom_col(position="dodge")+
  theme_minimal()+
  labs(title="Steps summary on the basis of days",x="Day of Week",y="Steps taken",caption="Fitbit Data by Mobius")+
  guides(fill=guide_legend(title="Steps taken by users"))+
  scale_fill_manual(values=c("red", "navyblue", "blueviolet"), name="Steps taken",breaks=c("min_steps", "avg_steps", "max_steps"),labels=c("Minimum steps", "Average steps", "Maximum steps"))+
  theme(plot.title = element_text(hjust = 0.5, face = "bold"), axis.title=element_text(face="bold"))
```

Steps summary on the basis of days



Observations:

1. Minimum steps moved by users is zero.
2. Users move an average of 7000-8000 steps a day.
3. Maximum steps moved by user is 36019.

Understanding user steps trends on different days of week

```
# generating data frame for step ranges on different days of week
steps_data_plot<- steps_data%>%
  dplyr::select(-min_steps,-max_steps,-avg_steps)%>%
  gather(Total, Value,-Days_of_week)

steps_data_plot$Days_of_week<-ordered(steps_data_plot$Days_of_week,      levels=c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))
steps_data_plot$Total<-ordered(steps_data_plot$Total,      levels=c("steps_zero", "steps_lessthan_10k", "steps_between_10_20k", "steps_morethan_20k"))
```

Plot for steps moved

```

ggplot(data=steps_data_plot,aes(x=Days_of_week,y=value,fill=Total))+  

  geom_col(position="dodge") +  

  theme_minimal() +  

  labs(title="Steps moved on the basis of days",x="Day of Week",y="Frequency",caption  

  ="Fitbit Data by Mobius") +  

  guides(fill=guide_legend(title="Steps taken by users")) +  

  scale_fill_manual(values=c("red", "navyblue", "blueviolet", "darkgreen"), name="Ste  

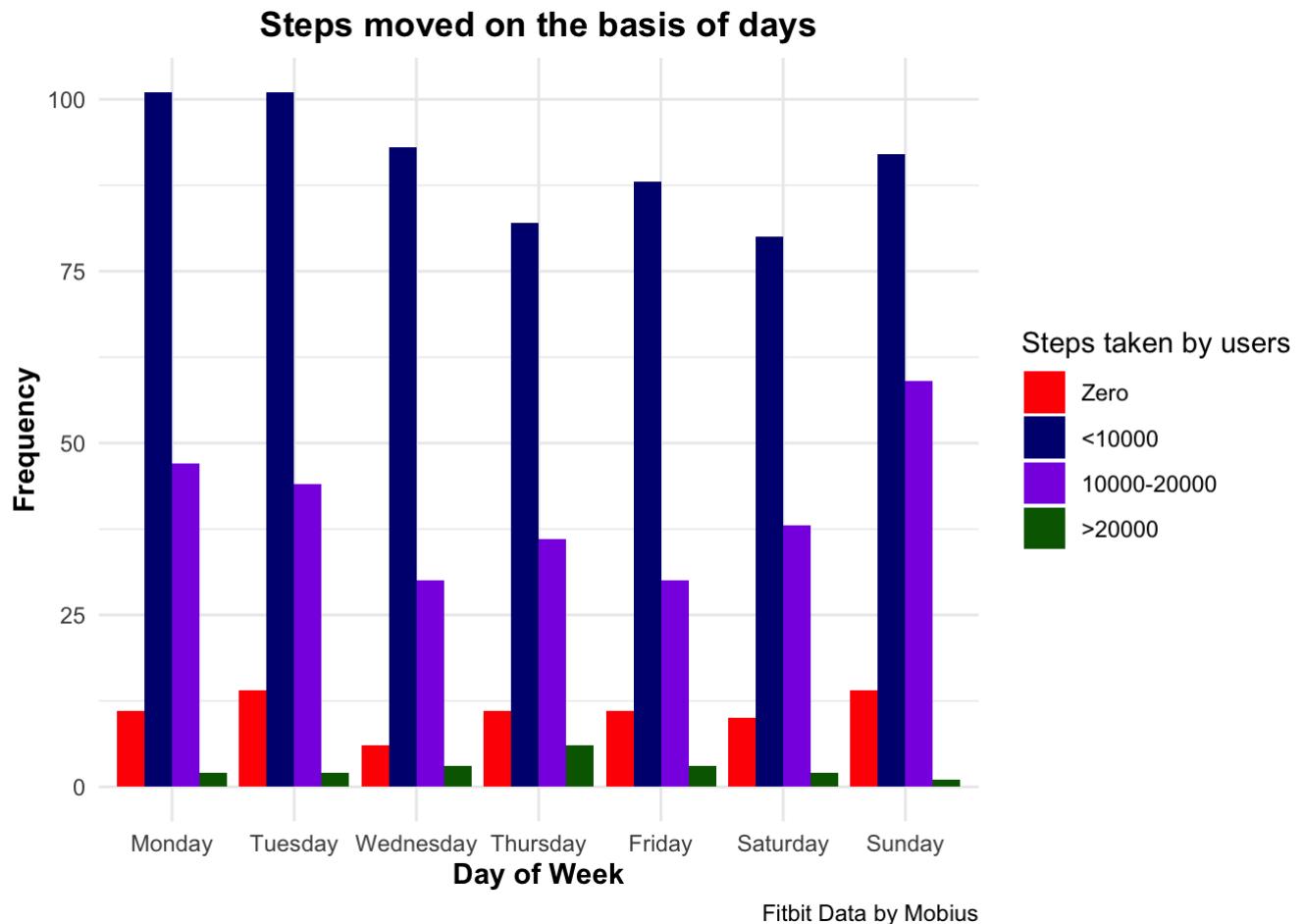
  ps taken",breaks=c("steps_zero", "steps_lessthan_10k", "steps_between_10_20k", "ste  

  ps_morethan_20k"),labels=c("Zero", "<10000", "10000-20000",">20000")) +  

  theme(plot.title = element_text(hjust = 0.5, face = "bold"), axis.title=element_tex  

t(face="bold"))

```



Observations:

1. Users exhibit a general tendency to move less than 10000 steps daily.
2. High likelihood of moving below 10000 steps is on Mondays and Tuesdays.
3. Among users moving 10000-20000 daily, they are very likely to move more on a Sunday.
4. Among users moving above 20000 steps, the likelihood of this movement is high on a Thursday.
5. Among users that move less, likelihood to not move (zero steps) is on Sunday or Tuesday.

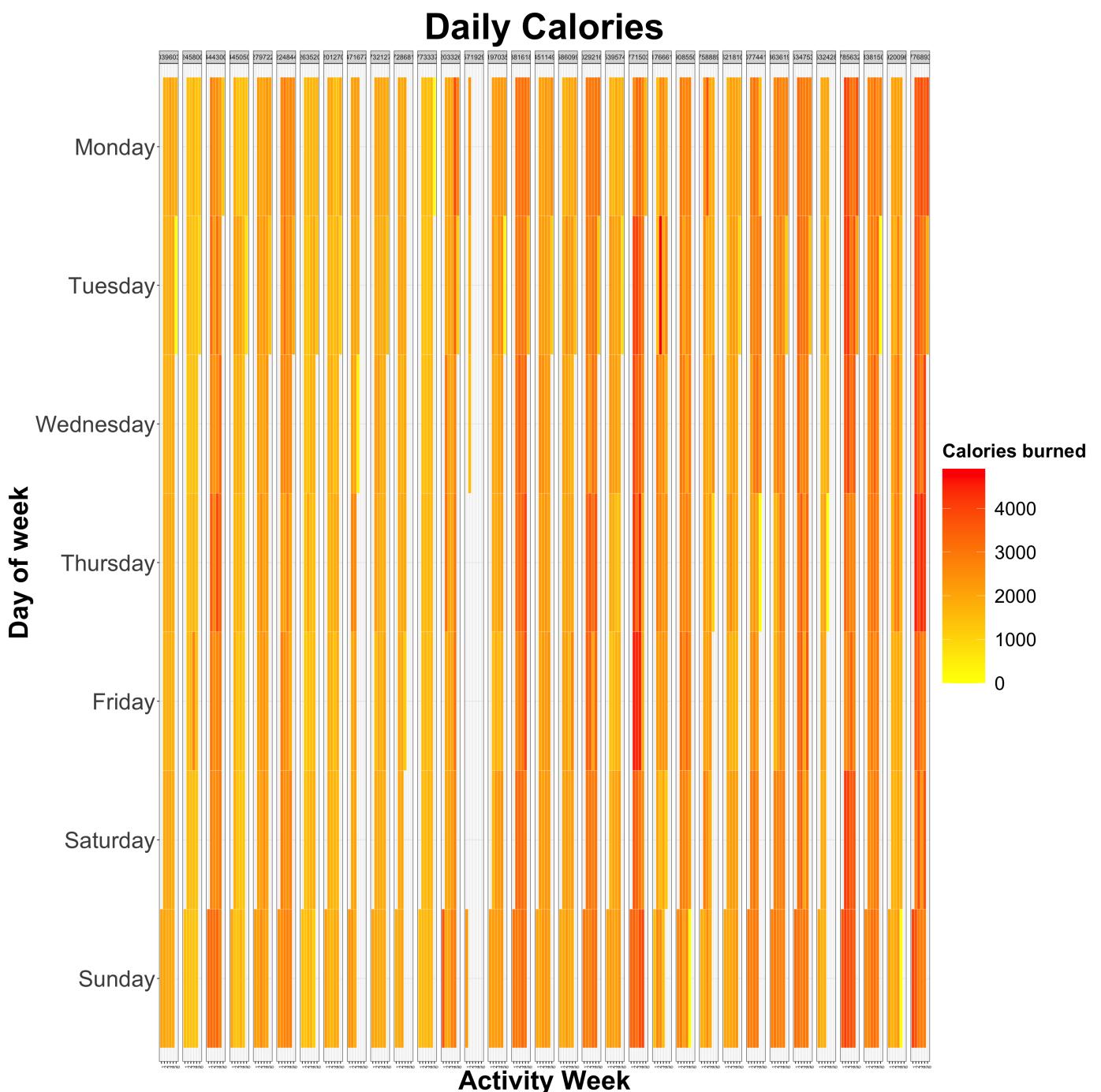
Trends in calories burned

Generating heat map representing daily calories burned

```

ggplot(Day_week, aes(x=norm_week, y=rev_ord.x, fill = Calories)) +
  geom_tile(colour = "white") + facet_grid(~Id) + scale_fill_gradient(low="yellow", h
igh="red") + xlab("Activity Week") + ylab("Day of week") + ggtitle("Daily Calories")
+ labs(fill = "Calories burned")+
  theme_bw() + theme(axis.text.x = element_text(angle=90,size=7), axis.text.y = element
_text(angle=0,size=30),axis.title=element_text(size=35,face="bold"), title = element_text(size=40,face="bold"), legend.text = element_text
(size=25), legend.key.size = unit(2, 'cm'),legend.key.height = unit(2, 'cm'), legend.
key.width = unit(2, 'cm'),legend.title = element_text(size=25))+
  scale_x_continuous(breaks=round(seq(min(Day_week$norm_week),max(Day_week$norm_wee
k),by=1),1))+theme(plot.title = element_text(hjust = 0.5))

```



Observations:

1. About ~5 users burned calories intensively suggesting very few users follow intensive workout regime.
2. High calories burned among users is mainly on a Sunday or Monday.

Segregating calorie ranges in terms of days of week

```

calories_data<-Day_week%>%
  group_by(Days_of_week=rev_ord.x)%>%
  summarise(calories_zero=sum(Calories==0),calories_lessthan1k=sum(Calories<1000), calories_between_1_3k=(sum(Calories>1000) - sum(Calories>3000)), calories_morethan_3k=sum(Calories>3000))%>%
  gather(Total, Value,-Days_of_week)
calories_data$Days_of_week<-ordered(calories_data$Days_of_week, levels=c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))
calories_data$Total<-ordered(calories_data$Total, levels=c("calories_zero", "calories_lessthan1k", "calories_between_1_3k", "calories_morethan_3k"))

head(calories_data)

```

```

## # A tibble: 6 × 3
##   Days_of_week Total      Value
##   <ord>        <ord>     <int>
## 1 Sunday       calories_zero     1
## 2 Saturday     calories_zero     0
## 3 Friday       calories_zero     0
## 4 Thursday     calories_zero     1
## 5 Wednesday    calories_zero     0
## 6 Tuesday      calories_zero     2

```

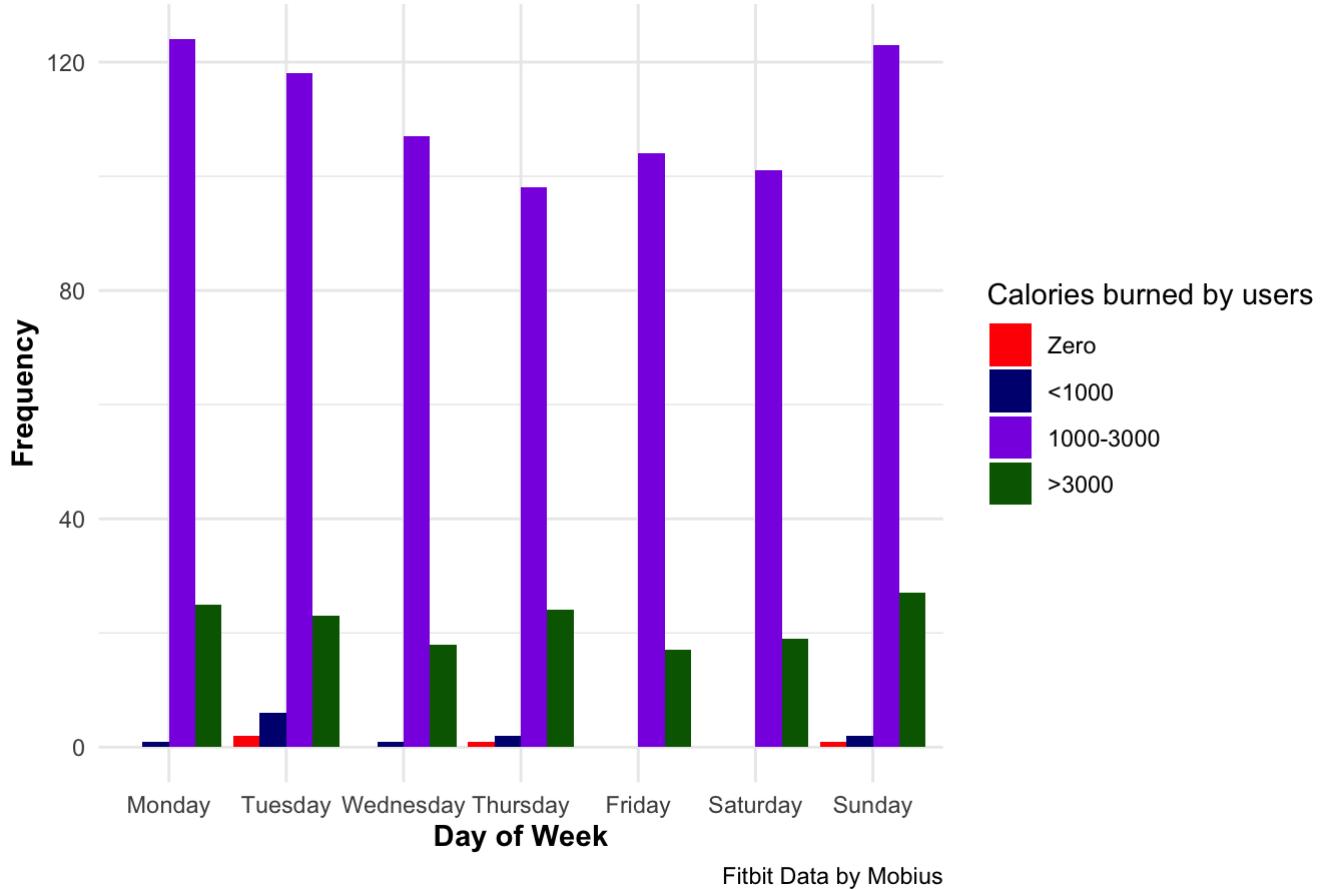
Plot for calories

```

ggplot(data=calories_data,aes(x=Days_of_week,y=Value,fill=Total))+
  geom_col(position="dodge")+
  theme_minimal()+
  labs(title="Steps moved on the basis of days",x="Day of Week",y="Frequency",caption =
  ="Fitbit Data by Mobius")+
  guides(fill=guide_legend(title="Calories burned by users"))+
  scale_fill_manual(values=c("red", "navyblue", "blueviolet", "darkgreen"), name="Calories burned",breaks=c("calories_zero", "calories_lessthan1k", "calories_between_1_3k", "calories_morethan_3k"),labels=c("Zero", "<1000", "1000-3000",">3000"))+
  theme(plot.title = element_text(hjust = 0.5, face = "bold"), axis.title=element_text(face="bold"))

```

Steps moved on the basis of days



Observations:

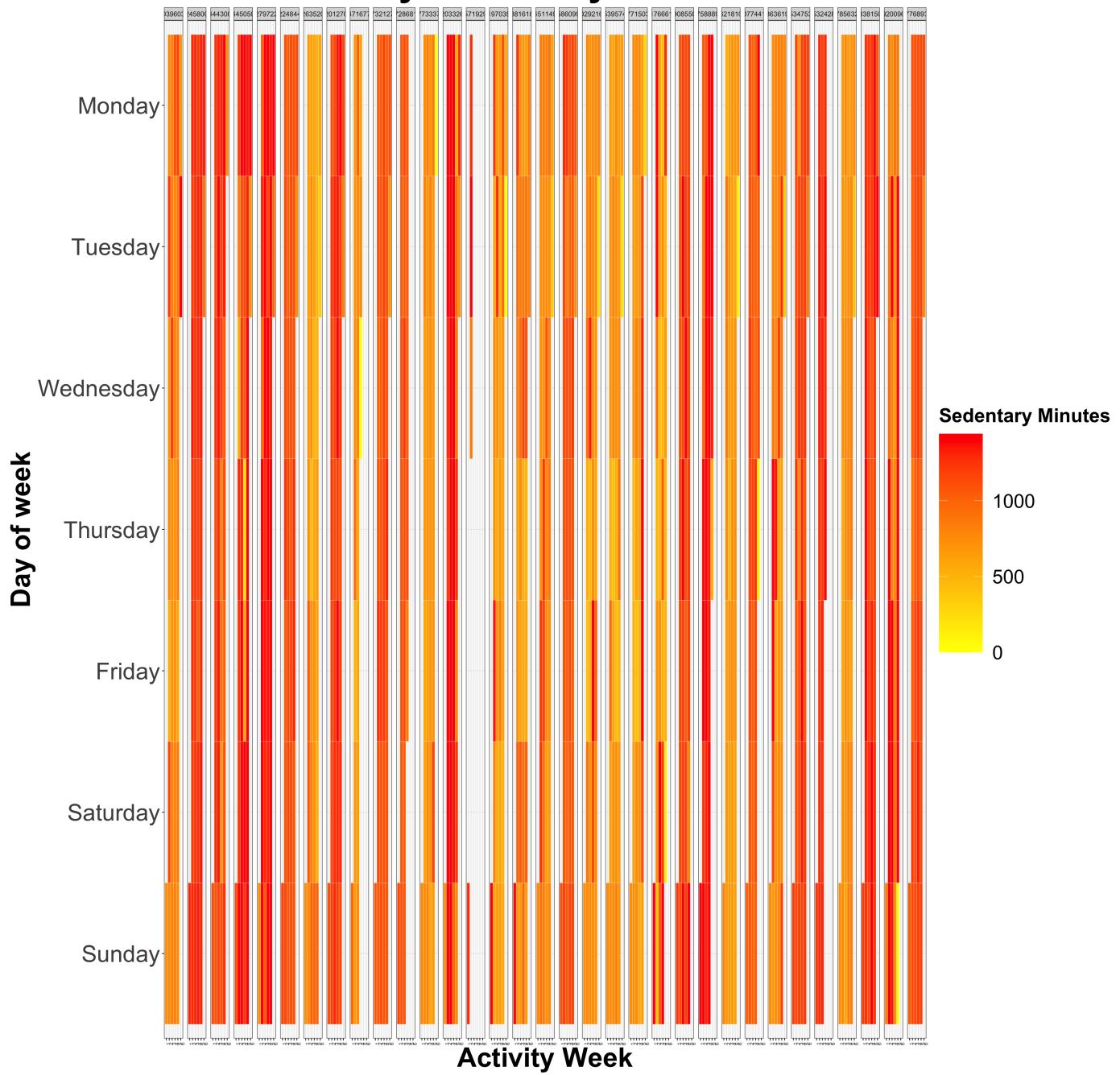
1. Among all users, 1000-3000 calories were commonly burned in a day. It was observed to be high on Monday and Sunday.
2. Few users showed calories burned above 3000 in a day. The most was observed on Monday, Thursday and Sunday.

Trends in daily sedentary minutes

Heat map representation for daily Sedentary Minutes

```
ggplot(Day_week, aes(x=norm_week, y=rev_ord.x, fill = SedentaryMinutes)) +
  geom_tile(colour = "white") + facet_grid(~Id) + scale_fill_gradient(low="yellow", h
igh="red") + xlab("Activity Week") + ylab("Day of week") + ggtitle("Daily Sedentary
Minutes") + labs(fill = "Sedentary Minutes")+
  theme_bw() + theme(axis.text.x = element_text(angle=90,size=7), axis.text.y = element
_text(angle=0,size=30),axis.text=element_text(size=35),axis.title=element_text(size=3
5,face="bold"), title = element_text(size=40,face="bold"), legend.text = element_text
(size=25), legend.key.size = unit(2, 'cm'),legend.key.height = unit(2, 'cm'), legend.
key.width = unit(2, 'cm'),legend.title = element_text(size=25))+
  scale_x_continuous(breaks=round(seq(min(Day_week$norm_week),max(Day_week$norm_wee
k),by=1),1))+
  theme(plot.title = element_text(hjust = 0.5))
```

Daily Sedentary Minutes



Observations:

1. Users are more Sedentary on Sundays.
2. Users are least Sedentary on Tuesdays and Fridays.

Understanding day wise data of sedentary duration

```

sedentary_data<-Day_week%>%
  group_by(Days_of_week=rev_ord.x)%>%
  summarise(sedentary_zero=sum(SedentaryMinutes==0), sedentary_lessthan500=sum(SedentaryMinutes<1000), sedentary_between_500_1k=(sum(SedentaryMinutes>500) - sum(SedentaryMinutes>1000)), sedentary_morethan_1k=sum(SedentaryMinutes>1000))%>%
  gather(Total, Value,-Days_of_week)
sedentary_data$Days_of_week<-ordered(sedentary_data$Days_of_week, levels=c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))
sedentary_data$Total<-ordered(sedentary_data$Total, levels=c("sedentary_zero", "sedentary_lessthan500", "sedentary_between_500_1k", "sedentary_morethan_1k"))

head(sedentary_data)

```

```

## # A tibble: 6 × 3
##   Days_of_week Total      Value
##   <ord>        <ord>     <int>
## 1 Sunday       sedentary_zero     0
## 2 Saturday     sedentary_zero     0
## 3 Friday       sedentary_zero     0
## 4 Thursday     sedentary_zero     0
## 5 Wednesday    sedentary_zero     0
## 6 Tuesday      sedentary_zero     1

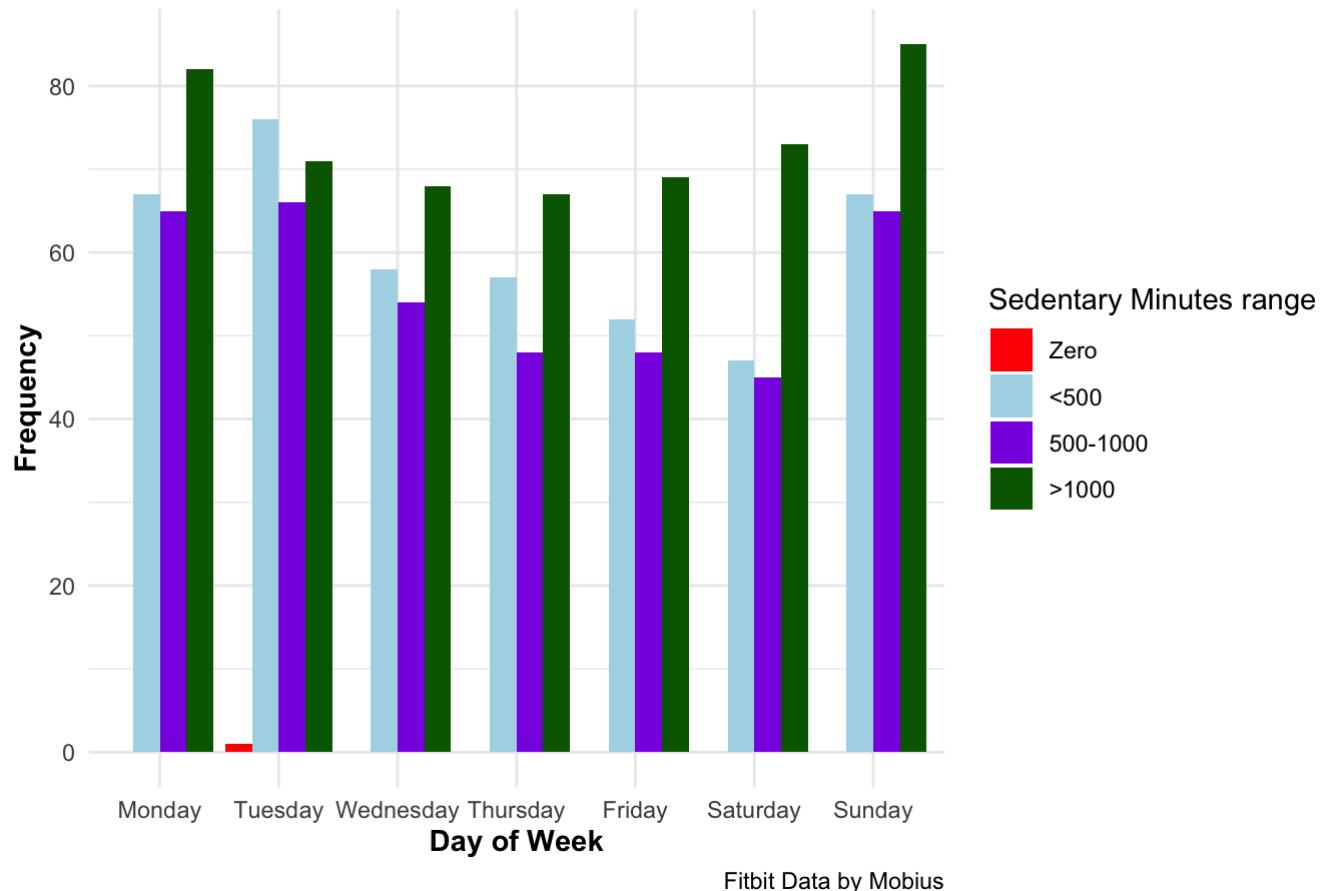
```

```

ggplot(data=sedentary_data,aes(x=Days_of_week,y=Value,fill=Total))+ 
  geom_col(position="dodge")+
  theme_minimal()+
  labs(title="Sedentary minutes on the basis of days",x="Day of Week",y="Frequency",caption="Fitbit Data by Mobius")+
  guides(fill=guide_legend(title="Sedentary Minutes range"))+
  scale_fill_manual(values=c("red", "lightblue", "blueviolet", "darkgreen"), name="Sedentary minutes",breaks=c("sedentary_zero", "sedentary_lessthan500", "sedentary_between_500_1k", "sedentary_morethan_1k"),labels=c("Zero", "<500", "500-1000", ">1000"))+
  theme(plot.title = element_text(hjust = 0.5, face = "bold"), axis.title=element_text(face="bold"))

```

Sedentary minutes on the basis of days



Observations:

1. Many users are sedentary for more than 1000 minutes a week.
2. Most sedentary behaviour is observed on a Sunday.
3. Least sedentary behaviour is observed on Thursday.

Trends in daily intensities of activity of each user

Understanding daily Intensities of each user in terms of lightly, moderately and very active minutes

```
#creating data frame to classify intensities
Active_Intensities<-Day_week%>%
  gather(Total,Value,-TotalSteps,-TotalDistance,-SedentaryMinutes,-ActivityDate,-Id,-
  day_of_week,-rev_ord.x,-week_no,-norm_week,-LightActiveDistance,-ModeratelyActiveDistance,
  -LightActiveDistance,-VeryActiveDistance,-Calories)
```

```

ggplot(data=Active_Intensities,aes(x=ActivityDate,y=Value,fill=Total))+  

  geom_col(position="dodge") +  

  facet_wrap(~Id) +  

  theme_minimal() +  

  theme(axis.text.x = element_text(angle=90,size=10), axis.text.y = element_text(angle=0,size=15),axis.title=element_text(size=35,face="bold"), title = element_text(size=40,face="bold"), legend.text = element_text(size=15), legend.title = element_text(size=20)) +  

  labs(title="Trends in Daily Active Minutes",x="Activity Date",y="Daily Active Minutes") +  

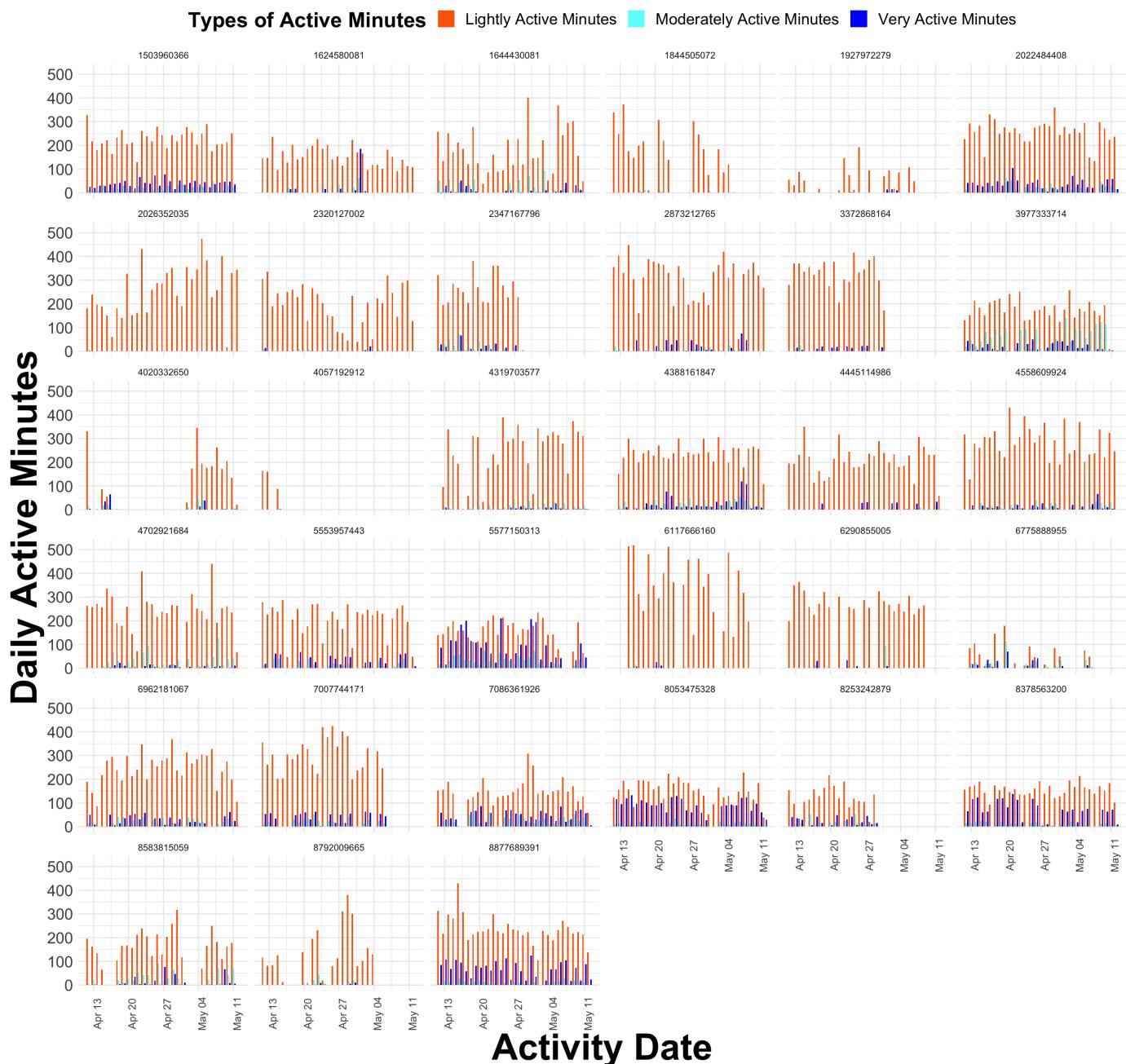
  guides(fill=guide_legend(title="Types of Active Minutes")) +  

  scale_fill_manual(values=c("#FF6600", "#66FFFF", "#0000FF"), name="Types of Active Minutes",breaks=c("LightlyActiveMinutes", "ModeratelyActiveMinutes", "VeryActiveMinutes"),labels=c("Lightly Active Minutes", "Moderately Active Minutes", "Very Active Minutes")) +  

  theme(plot.title = element_text(hjust = 0.5), legend.position ="top" )

```

Trends in Daily Active Minutes



Observations:

1. Almost all users were lightly active suggesting they keep moving throughout the day.
2. About 12 users were consistently very active throughout the duration of the study .
3. About 14 users exhibit consistent moderate daily activity.

Sumarized daily activity

Summarizing daily data of each user using pivot tables

Creating a pivot table to summarize activity of each user Deleting irrelevant information from Day_week dataframe

Keeping relevant column from the existing dataframes and analyzing the results from pivot table

```
daily_activity1<-Day_week%>%
  dplyr::select(-day_of_week,-rev_ord.x,-week_no,-norm_week)
str(daily_activity1)
```

```
## 'data.frame':    940 obs. of  12 variables:
##   $ Id                  : chr  "1503960366" "1503960366" "1503960366" "1503960366"
##   $ ...                 :
##   $ ActivityDate        : Date, format: "2020-04-12" "2020-04-13" ...
##   $ TotalSteps           : num  13162 10735 10460 9762 12669 ...
##   $ TotalDistance         : num  8.5 6.97 6.74 6.28 8.16 ...
##   $ VeryActiveDistance   : num  1.88 1.57 2.44 2.14 2.71 ...
##   $ ModeratelyActiveDistance: num  0.55 0.69 0.4 1.26 0.41 ...
##   $ LightActiveDistance  : num  6.06 4.71 3.91 2.83 5.04 ...
##   $ VeryActiveMinutes    : num  25 21 30 29 36 38 42 50 28 19 ...
##   $ ModeratelyActiveMinutes: num  13 19 11 34 10 20 16 31 12 8 ...
##   $ LightlyActiveMinutes : num  328 217 181 209 221 164 233 264 205 211 ...
##   $ SedentaryMinutes     : num  728 776 1218 726 773 ...
##   $ Calories              : num  1985 1797 1776 1745 1863 ...
```

Pivot table to summarize average activity per day of each user

```
library(lessR) # package to compute pivot tables
```

```
##
## lessR 4.1.8                      feedback: gerbing@pdx.edu
## -----
## > d <- Read("")      Read text, Excel, SPSS, SAS, or R data file
##   d is default data frame, data= in analysis routines optional
##
## Learn about reading, writing, and manipulating data, graphics,
## testing means and proportions, regression, factor analysis,
## customization, and descriptive statistics from pivot tables.
##   Enter: browseVignettes("lessR")
##
## View changes in this or recent versions of lessR.
##   Enter: help(package=lessR) Click: Package NEWS
##   Enter: interact() for access to interactive graphics
##   New function: reshape_long() to move data from wide to long
```

```
##  
## Attaching package: 'lessR'
```

```
## The following objects are masked from 'package:dplyr':  
##  
##     recode, rename
```

```
pivot_daily_activity<-pivot(data=daily_activity1,mean,c(TotalSteps,TotDistance, Ver  
yActiveDistance, ModeratelyActiveDistance,LightActiveDistance, VeryActiveMinutes,Mode  
ratelyActiveMinutes, LightlyActiveMinutes, SedentaryMinutes,Calories),by=Id)
```

```
str(pivot_daily_activity)
```

```
## 'data.frame':    33 obs. of  31 variables:  
##   $ Id                      : Factor w/ 33 levels "1503960366","162458008  
1",...: 1 2 3 4 5 6 7 8 9 10 ...  
##   $ TotalSteps_n             : num  31 31 30 31 31 31 31 31 31 18 31 ...  
##   $ TotalSteps_na            : num  0 0 0 0 0 0 0 0 0 0 0 ...  
##   $ TotalSteps_mean          : num  12117 5744 7283 2580 916 ...  
##   $ TotalDistance_n          : num  31 31 30 31 31 31 31 31 31 18 31 ...  
##   $ TotalDistance_na         : num  0 0 0 0 0 0 0 0 0 0 0 ...  
##   $ TotalDistance_mean        : num  7.81 3.915 5.295 1.706 0.635 ...  
##   $ VeryActiveDistance_n     : num  31 31 30 31 31 31 31 31 31 18 31 ...  
##   $ VeryActiveDistance_na    : num  0 0 0 0 0 0 0 0 0 0 0 ...  
##   $ VeryActiveDistance_mean  : num  2.858 0.939 0.73 0.008 0.096 ...  
##   $ ModeratelyActiveDistance_n: num  31 31 30 31 31 31 31 31 31 18 31 ...  
##   $ ModeratelyActiveDistance_na: num  0 0 0 0 0 0 0 0 0 0 0 ...  
##   $ ModeratelyActiveDistance_mean: num  0.794 0.361 0.951 0.049 0.031 ...  
##   $ LightActiveDistance_n    : num  31 31 30 31 31 31 31 31 31 18 31 ...  
##   $ LightActiveDistance_na   : num  0 0 0 0 0 0 0 0 0 0 0 ...  
##   $ LightActiveDistance_mean : num  4.153 2.607 3.609 1.647 0.507 ...  
##   $ VeryActiveMinutes_n      : num  31 31 30 31 31 31 31 31 31 18 31 ...  
##   $ VeryActiveMinutes_na     : num  0 0 0 0 0 0 0 0 0 0 0 ...  
##   $ VeryActiveMinutes_mean   : num  38.71 8.677 9.567 0.129 1.323 ...  
##   $ ModeratelyActiveMinutes_n: num  31 31 30 31 31 31 31 31 31 18 31 ...  
##   $ ModeratelyActiveMinutes_na: num  0 0 0 0 0 0 0 0 0 0 0 ...  
##   $ ModeratelyActiveMinutes_mean: num  19.161 5.806 21.367 1.29 0.774 ...  
##   $ LightlyActiveMinutes_n   : num  31 31 30 31 31 31 31 31 31 18 31 ...  
##   $ LightlyActiveMinutes_na  : num  0 0 0 0 0 0 0 0 0 0 0 ...  
##   $ LightlyActiveMinutes_mean: num  219.9 153.5 178.5 115.5 38.6 ...  
##   $ SedentaryMinutes_n       : num  31 31 30 31 31 31 31 31 31 18 31 ...  
##   $ SedentaryMinutes_na     : num  0 0 0 0 0 0 0 0 0 0 0 ...  
##   $ SedentaryMinutes_mean   : num  848 1258 1162 1207 1317 ...  
##   $ Calories_n              : num  31 31 30 31 31 31 31 31 31 18 31 ...  
##   $ Calories_na              : num  0 0 0 0 0 0 0 0 0 0 0 ...  
##   $ Calories_mean            : num  1816 1483 2811 1573 2173 ...
```

Number of activity days/data by each user

```
library(epiDisplay)#package for histogram
```

```
## Loading required package: foreign
```

```
## Loading required package: survival
```

```
## Loading required package: MASS
```

```
##  
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
##      select
```

```
## Loading required package: nnet
```

```
##  
## Attaching package: 'epiDisplay'
```

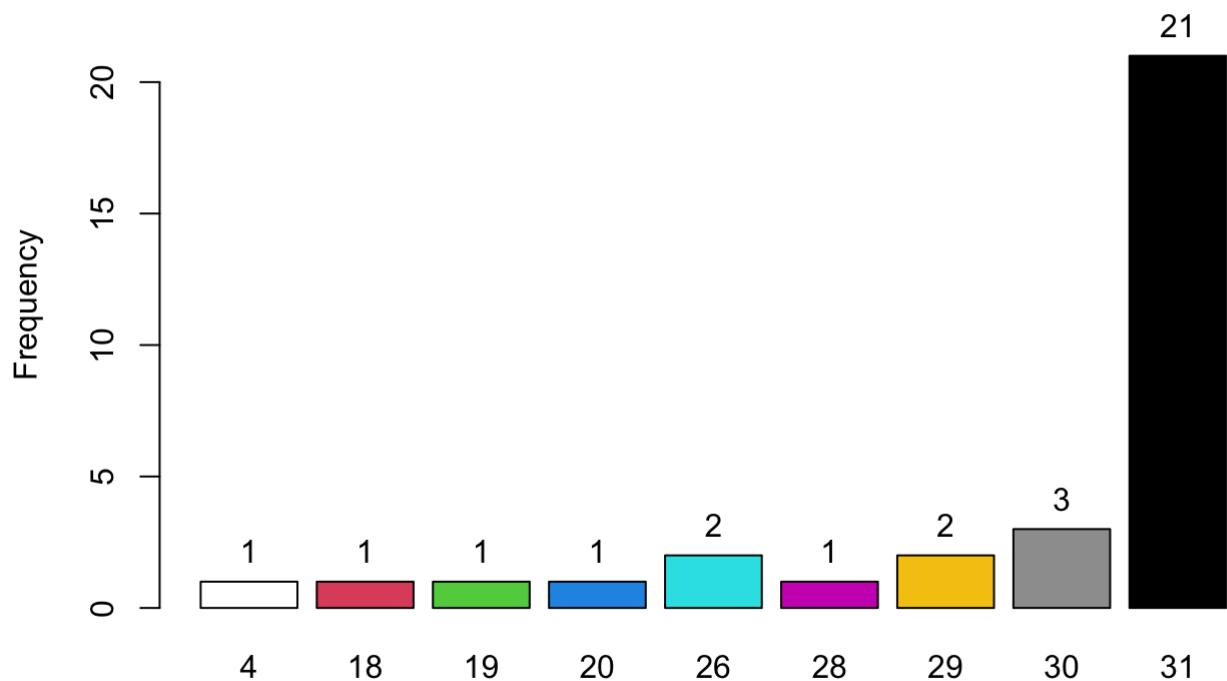
```
## The following object is masked from 'package:ggplot2':
```

```
##
```

```
##      alpha
```

```
tab1(pivot_daily_activity$TotalSteps_n, cum.percent = TRUE, main = 'Number of daily activity data provided by users', xlab='Number of days of activity')
```

Number of daily activity data provided by users



```
## pivot_daily_activity$TotalSteps_n :  
##      Frequency Percent Cum. percent  
## 4           1     3.0      3.0  
## 18          1     3.0      6.1  
## 19          1     3.0      9.1  
## 20          1     3.0     12.1  
## 26          2     6.1     18.2  
## 28          1     3.0     21.2  
## 29          2     6.1     27.3  
## 30          3     9.1     36.4  
## 31         21    63.6    100.0  
##   Total      33   100.0   100.0
```

Observations:

1. About 29 users consistently used the Fitbit device for over 26 days.
2. About 21 users used the Fitbit device for 31 days.
3. 4 users used the device for less than 26 days.

Summarized representation of daily activity variables

```

pivot_daily_activity_refined <- pivot_daily_activity%>%
  dplyr::select(-TotalSteps_n,-TotalSteps_na,-TotalDistance_n,-TotalDistance_na,-Very
ActiveDistance_n,-VeryActiveDistance_na,-ModeratelyActiveDistance_n,-ModeratelyActive
Distance_na,-LightActiveDistance_n,-LightActiveDistance_na,-VeryActiveMinutes_n,-Very
ActiveMinutes_na,-ModeratelyActiveMinutes_n,-ModeratelyActiveMinutes_na,-LightlyActiv
eMinutes_n,-LightlyActiveMinutes_na,-SedentaryMinutes_n,-SedentaryMinutes_na,-Calorie
s_n,-Calories_na)

str(pivot_daily_activity_refined)

```

```

## 'data.frame':    33 obs. of  11 variables:
##   $ Id                      : Factor w/ 33 levels "1503960366","162458008
1",...: 1 2 3 4 5 6 7 8 9 10 ...
##   $ TotalSteps_mean          : num  12117 5744 7283 2580 916 ...
##   $ TotalDistance_mean       : num  7.81 3.915 5.295 1.706 0.635 ...
##   $ VeryActiveDistance_mean : num  2.858 0.939 0.73 0.008 0.096 ...
##   $ ModeratelyActiveDistance_mean: num  0.794 0.361 0.951 0.049 0.031 ...
##   $ LightActiveDistance_mean : num  4.153 2.607 3.609 1.647 0.507 ...
##   $ VeryActiveMinutes_mean   : num  38.71 8.677 9.567 0.129 1.323 ...
##   $ ModeratelyActiveMinutes_mean: num  19.161 5.806 21.367 1.29 0.774 ...
##   $ LightlyActiveMinutes_mean : num  219.9 153.5 178.5 115.5 38.6 ...
##   $ SedentaryMinutes_mean    : num  848 1258 1162 1207 1317 ...
##   $ Calories_mean            : num  1816 1483 2811 1573 2173 ...

```

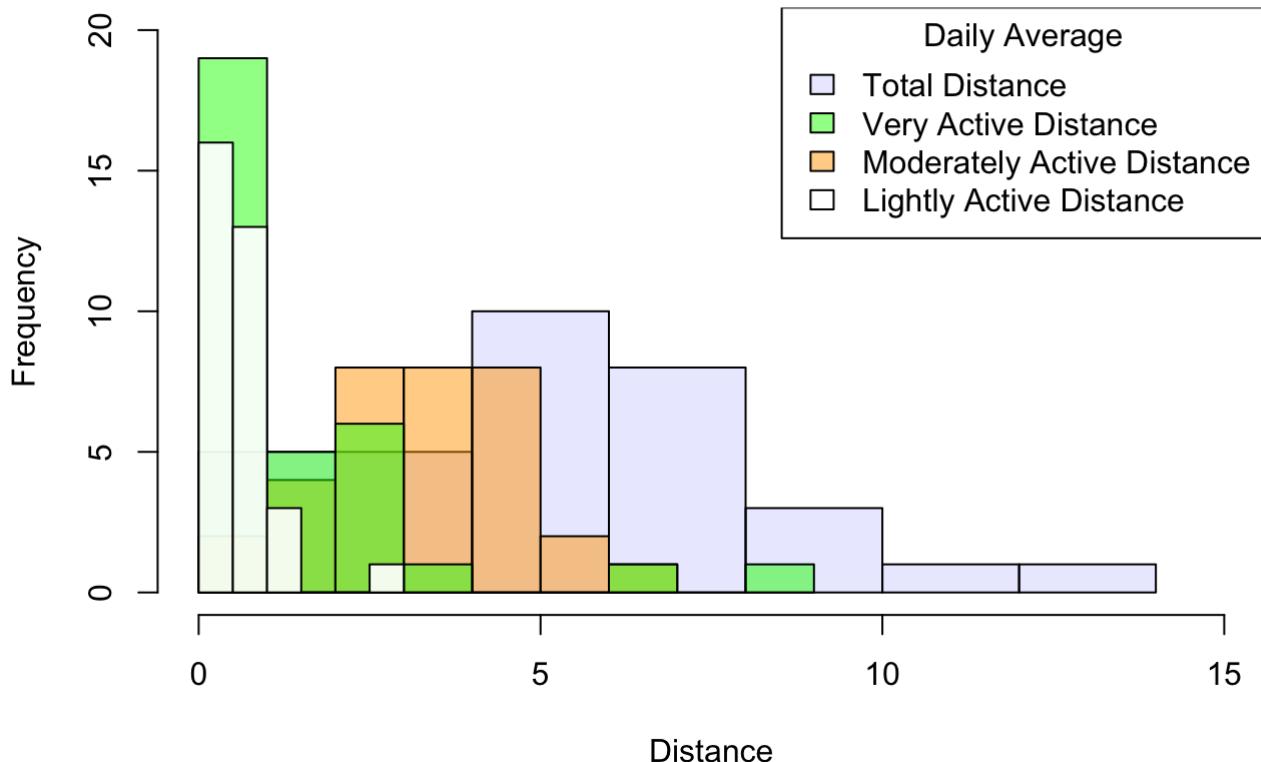
Generating histogram to represent average distance moved daily

```

hist(pivot_daily_activity_refined$TotalDistance_mean, ylim=c(0,20), xlim=c(0,15),col=
yarr::transparent("blue", trans.val = .9), main='Average daily distance', xlab='Dist
ance')
hist(pivot_daily_activity_refined$LightActiveDistance_mean, col=yarr::transparent("o
range", trans.val = 0.5), add=TRUE)
hist(pivot_daily_activity_refined$VeryActiveDistance_mean, col=yarr::transparent("gr
een", trans.val = .5), add=TRUE)
hist(pivot_daily_activity_refined$ModeratelyActiveDistance_mean, col=yarr::transpar
ent("white", trans.val = .1), add=TRUE)
legend('topright', c('Total Distance', 'Very Active Distance', 'Moderately Active Dis
tance', 'Lightly Active Distance'), fill=c(yarr::transparent("blue", trans.val = .9
), yarr::transparent("green", trans.val = .5), yarr::transparent("orange", trans.va
l = 0.5), yarr::transparent("white", trans.val = .1)), title="Daily Average")

```

Average daily distance



Observations:

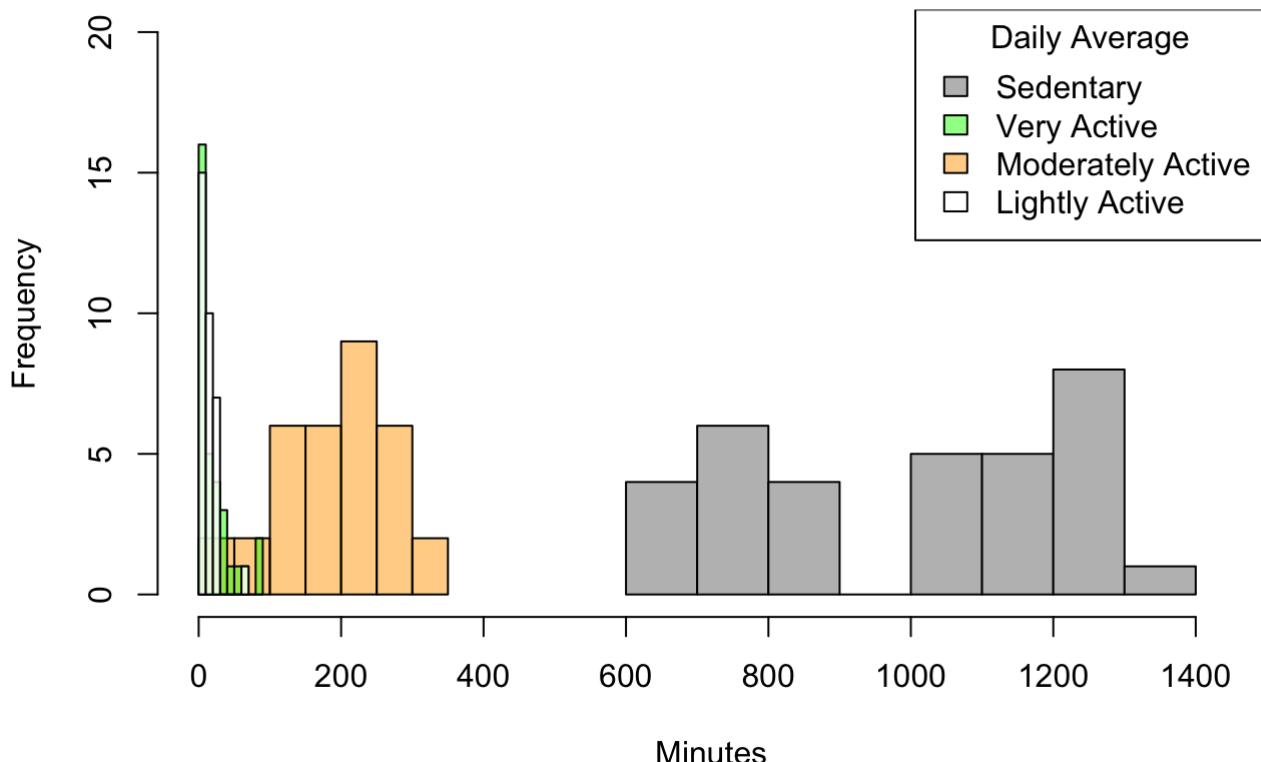
Unit of distance was not provided in the data. It could most likely be in miles.

1. About 10 users moved an average daily distance of 5 units.
2. About 19 users moved a very actively for an average distance of 1 units daily.
3. About 24 users moved moderately for an average distance of 4 units daily.
4. Moderate activity was common among users.

Average daily activity minutes by user

```
hist(pivot_daily_activity_refined$SedentaryMinutes_mean, ylim=c(0,20), xlim=c(0,1440), col=yarrr::transparent("grey", trans.val = 0), main='Average daily activity in minutes', xlab='Minutes')
hist(pivot_daily_activity_refined$LightlyActiveMinutes_mean, col=yarrr::transparent("orange", trans.val = 0.5), add=TRUE)
hist(pivot_daily_activity_refined$VeryActiveMinutes_mean, col=yarrr::transparent("green", trans.val = .5), add=TRUE)
hist(pivot_daily_activity_refined$ModeratelyActiveMinutes_mean, col=yarrr::transparent("white", trans.val = .2), add=TRUE)
legend('topright', c('Sedentary', 'Very Active', 'Moderately Active', 'Lightly Active'), fill=c(yarrr::transparent("grey", trans.val = 0), yarrr::transparent("green", trans.val = .5), yarrr::transparent("orange", trans.val = 0.5), yarrr::transparent("white", trans.val = .2)), title="Daily Average")
```

Average daily activity in minutes



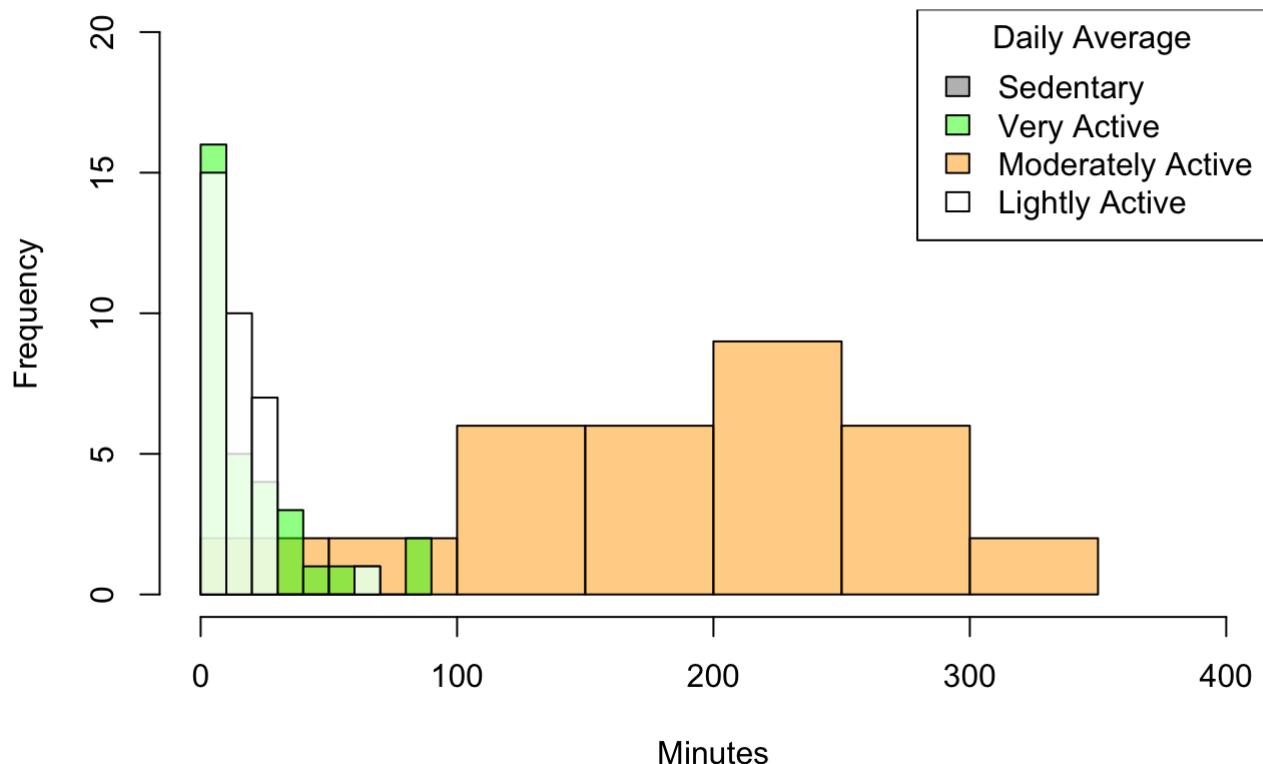
Observations:

1. Users exhibit an average of 600-1400 sedentary minutes daily.
2. Moderate activity is prominent among users.
3. User activity showing long sedentary minutes suggests that they wore the fitness device to work and sleep.

Setting the scale of graph to exclude sedentary minutes

```
hist(pivot_daily_activity_refined$SedentaryMinutes_mean, ylim=c(0,20), xlim=c(0,400),
col=yarrr::transparent("grey", trans.val = 0), main='Average daily activity in minutes',
xlab='Minutes')
hist(pivot_daily_activity_refined$LightlyActiveMinutes_mean,   col=yarrr::transparent(
"orange", trans.val = 0.5), add=TRUE)
hist(pivot_daily_activity_refined$VeryActiveMinutes_mean,   col=yarrr::transparent("green",
trans.val = .5), add=TRUE)
hist(pivot_daily_activity_refined$ModeratelyActiveMinutes_mean,   col=yarrr::transparent(
"white", trans.val = .2), add=TRUE)
legend('topright', c('Sedentary', 'Very Active', 'Moderately Active', 'Lightly Active'),
fill=c(yarrr::transparent("grey", trans.val = 0), yarrr::transparent("green", trans.val =
.5), yarrr::transparent("orange", trans.val = 0.5), yarrr::transparent("white", trans.val =
.2)), title="Daily Average")
```

Average daily activity in minutes



Observations:

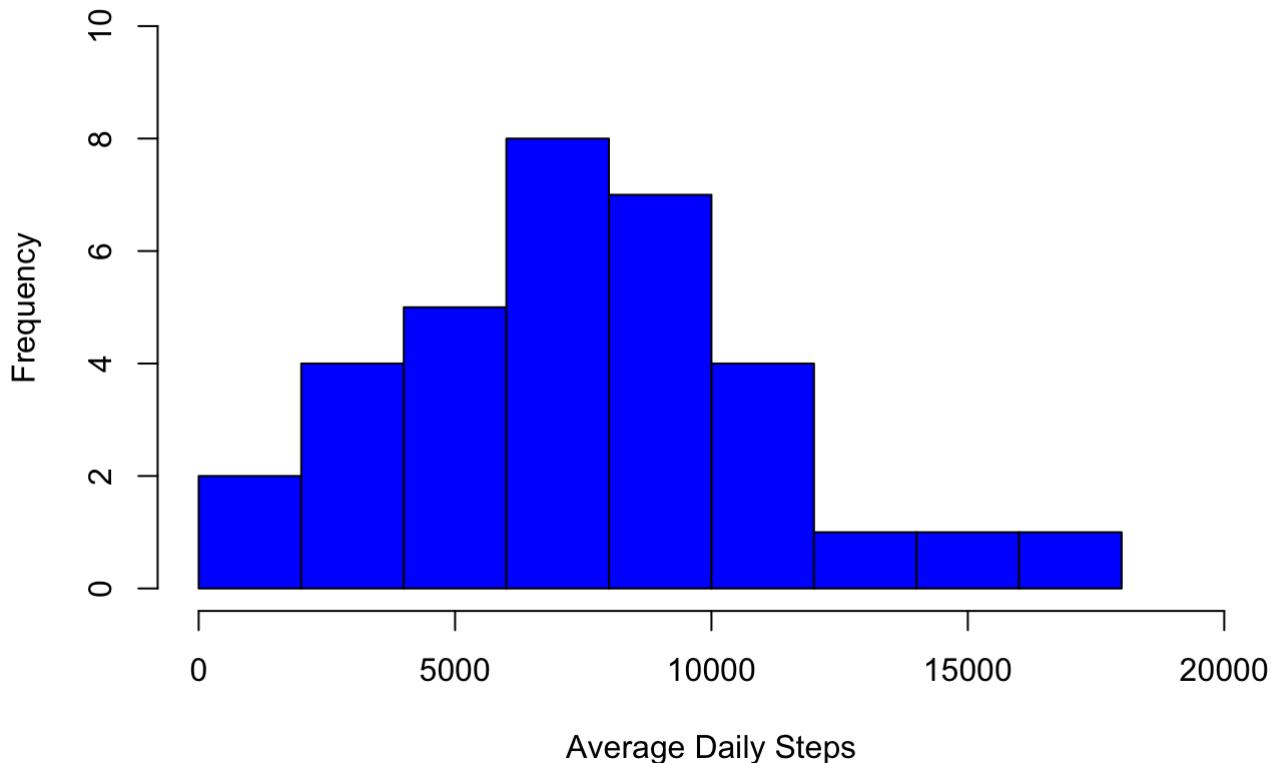
1. On an average users were moderately active for 200 minutes daily.
2. Most users were very active for 10 minutes daily.
3. Moderate activity seemed common among users.

Average steps and calories of users

Histogram of steps per day

```
hist(pivot_daily_activity_refined$TotalSteps_mean, ylim=c(0,10), xlim=c(0,20000), col=yarrr::transparent("blue", trans.val = 0), main=' Average daily steps', xlab='Average Daily Steps')
```

Average daily steps



```
mean(pivot_daily_activity_refined$TotalSteps_mean)
```

```
## [1] 7519.273
```

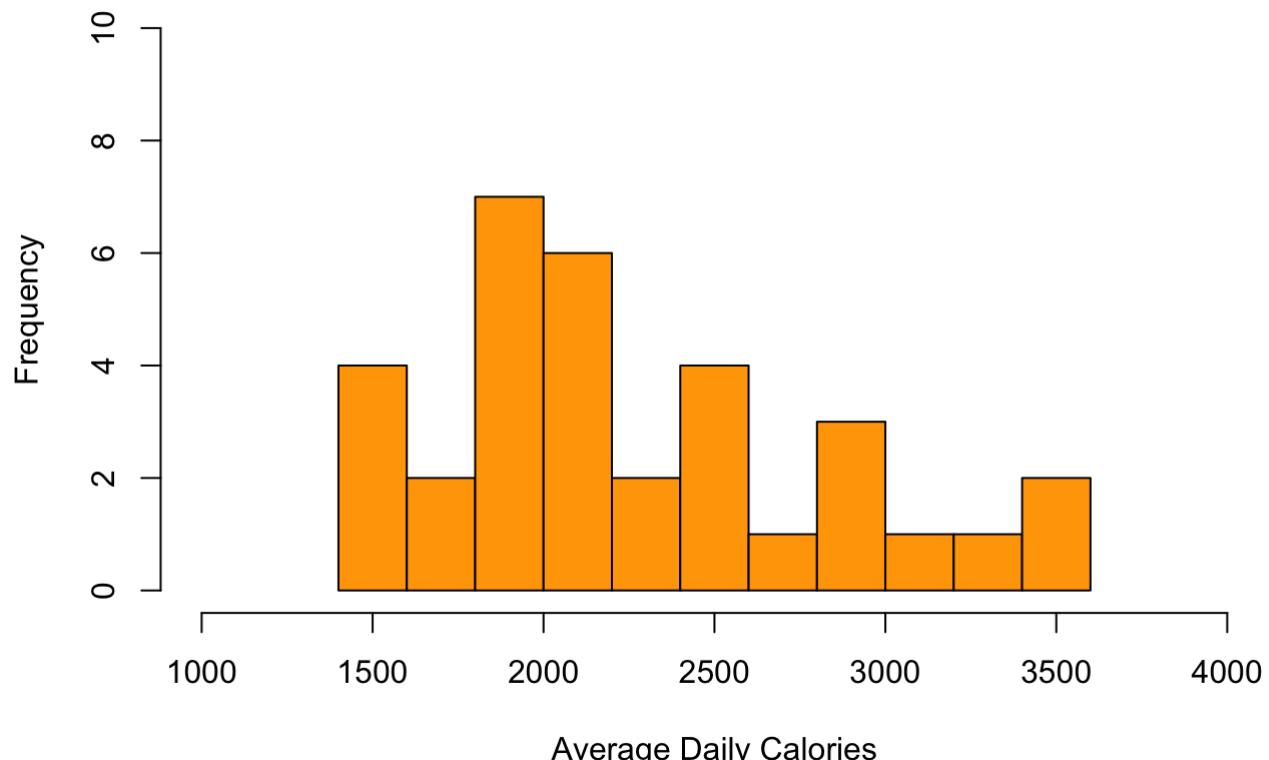
Observations:

1. Average daily steps of all users was ~7519.
2. Users moved an average of 0-18000 steps daily.

Histogram of average daily Calories

```
hist(pivot_daily_activity_refined$Calories_mean, ylim=c(0,10), xlim=c(1000,4000), col=yarrr::transparent("orange", trans.val = 0), main='Average daily calories', xlab='Average Daily Calories')
```

Average daily calories



```
mean(pivot_daily_activity_refined$Calories_mean)
```

```
## [1] 2282.444
```

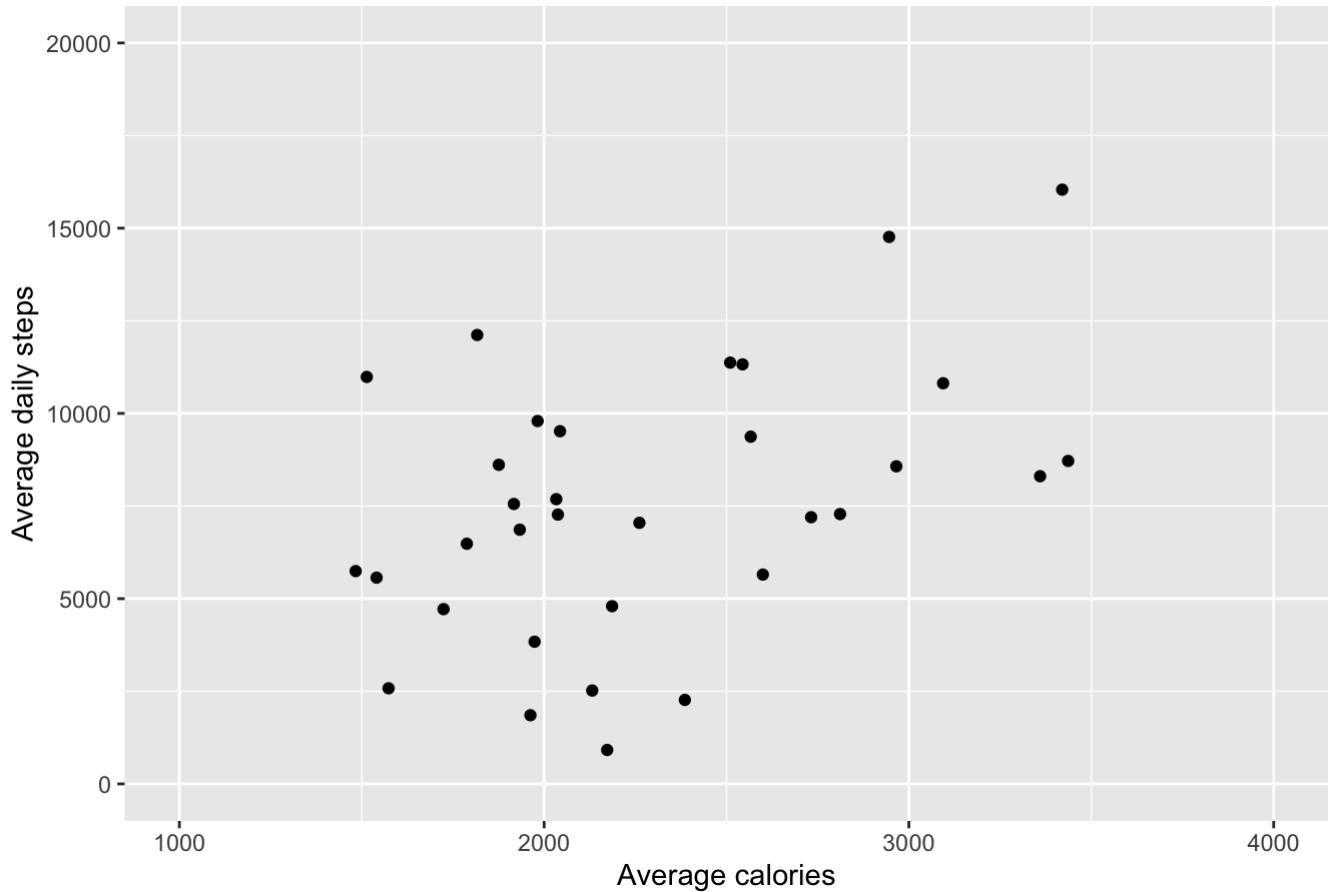
Observations:

1. Users burned an average of 2282 calories daily.
2. Average daily calories each user was in the range 1500-3500.

Relation between average total steps and average calories

```
ggplot(data=pivot_daily_activity_refined)+  
  geom_point(mapping=aes(x=Calories_mean, y=TotalSteps_mean))+  
  xlab("Average calories") + ylab("Average daily steps") + ggtitle("Relation between  
the average daily calories and average daily steps")+xlim(1000,4000)+ylim(0,20000)
```

Relation between the average daily calories and average daily steps



Observations:

1. Average calories burned daily and average daily steps do not have a linear relationship suggesting there are other factors influencing the calories burned daily.

Correlation between different daily variables

```
library(ggcorrplot) #package to plot correlation plot

cor_plot<-dplyr:::select_if(pivot_daily_activity_refined,is.numeric)

correlation_average_daily_activity<-cor(cor_plot, use="complete.obs")
round(correlation_average_daily_activity,2)
```

	TotalSteps_mean	TotalDistance_mean
## TotalSteps_mean	1.00	0.98
## TotalDistance_mean	0.98	1.00
## VeryActiveDistance_mean	0.78	0.82
## ModeratelyActiveDistance_mean	0.49	0.45
## LightActiveDistance_mean	0.72	0.72
## VeryActiveMinutes_mean	0.70	0.73
## ModeratelyActiveMinutes_mean	0.48	0.44
## LightlyActiveMinutes_mean	0.51	0.45
## SedentaryMinutes_mean	-0.39	-0.32
## Calories_mean	0.44	0.55
##	VeryActiveDistance_mean	
## TotalSteps_mean		0.78
## TotalDistance_mean		0.82
## VeryActiveDistance_mean		1.00
## ModeratelyActiveDistance_mean		0.16
## LightActiveDistance_mean		0.25
## VeryActiveMinutes_mean		0.89
## ModeratelyActiveMinutes_mean		0.22
## LightlyActiveMinutes_mean		0.03
## SedentaryMinutes_mean		-0.04
## Calories_mean		0.51
##	ModeratelyActiveDistance_mean	
## TotalSteps_mean		0.49
## TotalDistance_mean		0.45
## VeryActiveDistance_mean		0.16
## ModeratelyActiveDistance_mean		1.00
## LightActiveDistance_mean		0.26
## VeryActiveMinutes_mean		0.18
## ModeratelyActiveMinutes_mean		0.96
## LightlyActiveMinutes_mean		0.12
## SedentaryMinutes_mean		-0.40
## Calories_mean		0.06
##	LightActiveDistance_mean	VeryActiveMinutes_mean
## TotalSteps_mean	0.72	0.70
## TotalDistance_mean	0.72	0.73
## VeryActiveDistance_mean	0.25	0.89
## ModeratelyActiveDistance_mean	0.26	0.18
## LightActiveDistance_mean	1.00	0.21
## VeryActiveMinutes_mean	0.21	1.00
## ModeratelyActiveMinutes_mean	0.19	0.31
## LightlyActiveMinutes_mean	0.83	0.00
## SedentaryMinutes_mean	-0.45	-0.20
## Calories_mean	0.35	0.63
##	ModeratelyActiveMinutes_mean	
## TotalSteps_mean	0.48	
## TotalDistance_mean	0.44	
## VeryActiveDistance_mean	0.22	
## ModeratelyActiveDistance_mean	0.96	
## LightActiveDistance_mean	0.19	
## VeryActiveMinutes_mean	0.31	
## ModeratelyActiveMinutes_mean	1.00	
## LightlyActiveMinutes_mean	0.03	
## SedentaryMinutes_mean	-0.39	
## Calories_mean	0.16	

```

##                                     LightlyActiveMinutes_mean SedentaryMinutes_mean
## TotalSteps_mean                           0.51                  -0.39
## TotalDistance_mean                        0.45                  -0.32
## VeryActiveDistance_mean                  0.03                  -0.04
## ModeratelyActiveDistance_mean           0.12                  -0.40
## LightActiveDistance_mean                 0.83                  -0.45
## VeryActiveMinutes_mean                  0.00                  -0.20
## ModeratelyActiveMinutes_mean            0.03                  -0.39
## LightlyActiveMinutes_mean                1.00                  -0.44
## SedentaryMinutes_mean                  -0.44                  1.00
## Calories_mean                           0.00                  -0.08
##                                     Calories_mean
## TotalSteps_mean                          0.44
## TotalDistance_mean                       0.55
## VeryActiveDistance_mean                 0.51
## ModeratelyActiveDistance_mean          0.06
## LightActiveDistance_mean                0.35
## VeryActiveMinutes_mean                  0.63
## ModeratelyActiveMinutes_mean           0.16
## LightlyActiveMinutes_mean               0.00
## SedentaryMinutes_mean                  -0.08
## Calories_mean                           1.00

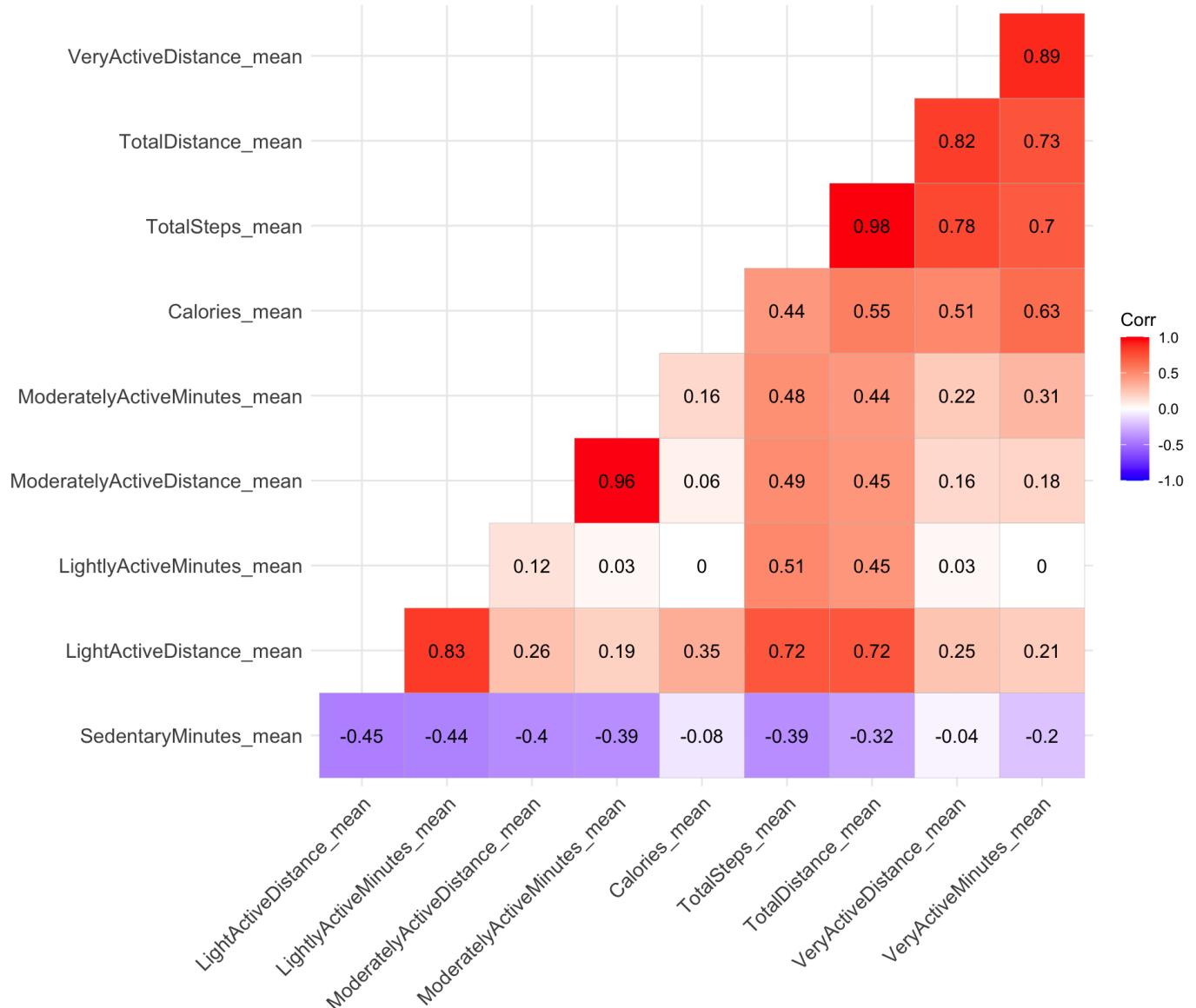
```

```

ggcorrplot(correlation_average_daily_activity,
            hc.order = TRUE,
            type = "lower",
            lab = TRUE) +
  labs(title = "Correlation plot") +
  theme(plot.title = element_text(face="bold", hjust=0.5, size = 20), axis.title = element_text(face = "bold"))

```

Correlation plot



Observations:

This plot establishes a linear relationship between:

1. Total distance and total steps
2. Lightly active distance and lightly active minutes
3. Moderately active distance and moderately active minutes
4. Very active distance and very active minutes

This correlational plot is further represented by relational scatter plots between all variables and density plots

```
library(GGally) #package for scatter plot matrices
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

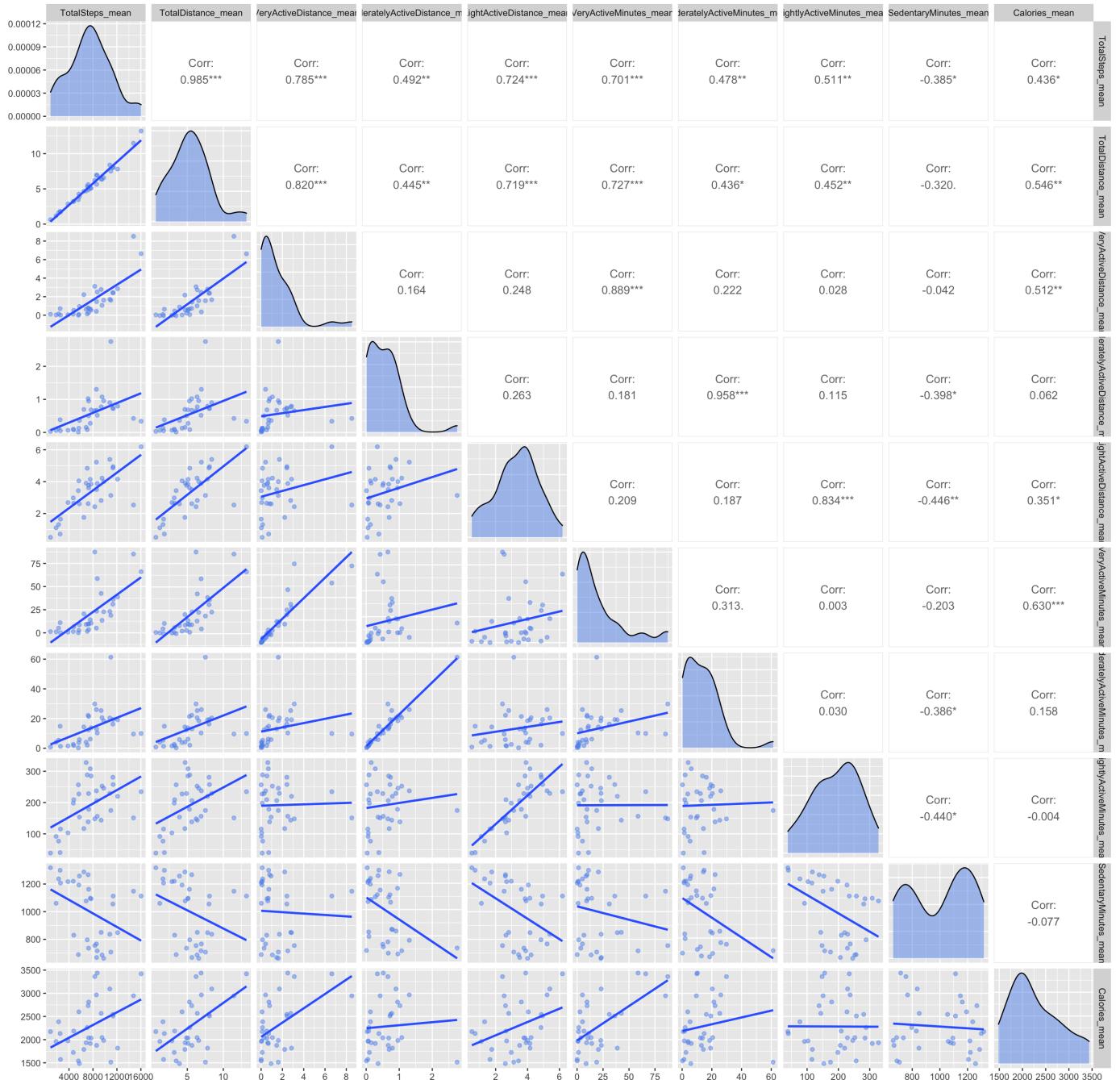
```
# custom function for density plot
my_density <- function(data, mapping, ...){
  ggplot(data = data, mapping = mapping) +
    geom_density(alpha = 0.5,
                 fill = "cornflowerblue", ...)
}

# custom function for scatterplot
my_scatter <- function(data, mapping, ...){
  ggplot(data = data, mapping = mapping) +
    geom_point(alpha = 0.5,
               color = "cornflowerblue") +
    geom_smooth(method=lm,
                se=FALSE, ...)
}

# create scatterplot matrix
ggpairs(cor_plot,
        lower=list(continuous = my_scatter),
        diag = list(continuous = my_density)) +
  labs(title = "Correlation and Density Plots Average Daily Activity Variables") + theme(plot.title = element_text(face="bold", hjust=0.5, size=20))
```

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


Correlation and Density Plots Average Daily Activity Variables



`theme_bw()`

```

## List of 93
## $ line                      :List of 6
## ..$ colour      : chr "black"
## ..$ size        : num 0.5
## ..$ linetype    : num 1
## ..$ lineend     : chr "butt"
## ..$ arrow       : logi FALSE
## ..$ inherit.blank: logi TRUE
## ... attr(*, "class")= chr [1:2] "element_line" "element"
## $ rect                      :List of 5
## ..$ fill        : chr "white"
## ..$ colour     : chr "black"
## ..$ size        : num 0.5
## ..$ linetype    : num 1
## ..$ inherit.blank: logi TRUE
## ... attr(*, "class")= chr [1:2] "element_rect" "element"
## $ text                      :List of 11
## ..$ family     : chr ""
## ..$ face       : chr "plain"
## ..$ colour     : chr "black"
## ..$ size        : num 11
## ..$ hjust      : num 0.5
## ..$ vjust      : num 0.5
## ..$ angle       : num 0
## ..$ lineheight  : num 0.9
## ..$ margin      : 'margin' num [1:4] 0points 0points 0points 0points
## ... ...- attr(*, "unit")= int 8
## ..$ debug      : logi FALSE
## ..$ inherit.blank: logi TRUE
## ... attr(*, "class")= chr [1:2] "element_text" "element"
## $ title                     : NULL
## $ aspect.ratio             : NULL
## $ axis.title               : NULL
## $ axis.title.x              :List of 11
## ..$ family     : NULL
## ..$ face       : NULL
## ..$ colour     : NULL
## ..$ size        : NULL
## ..$ hjust      : NULL
## ..$ vjust      : num 1
## ..$ angle       : NULL
## ..$ lineheight  : NULL
## ..$ margin      : 'margin' num [1:4] 2.75points 0points 0points 0points
## ... ...- attr(*, "unit")= int 8
## ..$ debug      : NULL
## ..$ inherit.blank: logi TRUE
## ... attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.title.x.top          :List of 11
## ..$ family     : NULL
## ..$ face       : NULL
## ..$ colour     : NULL
## ..$ size        : NULL
## ..$ hjust      : NULL
## ..$ vjust      : num 0
## ..$ angle       : NULL

```

```
## ..$ lineheight : NULL
## ..$ margin      : 'margin' num [1:4] 0points 0points 2.75points 0points
## ... -- attr(*, "unit")= int 8
## ..$ debug       : NULL
## ..$ inherit.blank: logi TRUE
## ... attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.title.x.bottom      : NULL
## $ axis.title.y              :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : NULL
##   ..$ size        : NULL
##   ..$ hjust       : NULL
##   ..$ vjust       : num 1
##   ..$ angle       : num 90
##   ..$ lineheight  : NULL
##   ..$ margin      : 'margin' num [1:4] 0points 2.75points 0points 0points
##   ... -- attr(*, "unit")= int 8
##   ..$ debug       : NULL
##   ..$ inherit.blank: logi TRUE
##   ... attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.title.y.left         : NULL
## $ axis.title.y.right        :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : NULL
##   ..$ size        : NULL
##   ..$ hjust       : NULL
##   ..$ vjust       : num 0
##   ..$ angle       : num -90
##   ..$ lineheight  : NULL
##   ..$ margin      : 'margin' num [1:4] 0points 0points 0points 2.75points
##   ... -- attr(*, "unit")= int 8
##   ..$ debug       : NULL
##   ..$ inherit.blank: logi TRUE
##   ... attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text                 :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : chr "grey30"
##   ..$ size        : 'rel' num 0.8
##   ..$ hjust       : NULL
##   ..$ vjust       : NULL
##   ..$ angle       : NULL
##   ..$ lineheight  : NULL
##   ..$ margin      : NULL
##   ..$ debug       : NULL
##   ..$ inherit.blank: logi TRUE
##   ... attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.x                :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : NULL
##   ..$ size        : NULL
##   ..$ hjust       : NULL
##   ..$ vjust       : num 1
```

```
## ..$ angle      : NULL
## ..$ lineheight : NULL
## ..$ margin     : 'margin' num [1:4] 2.2points 0points 0points 0points
## ... -- attr(*, "unit")= int 8
## ..$ debug      : NULL
## ..$ inherit.blank: logi TRUE
## ... attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.x.top      :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : NULL
##   ..$ size        : NULL
##   ..$ hjust       : NULL
##   ..$ vjust       : num 0
##   ..$ angle       : NULL
##   ..$ lineheight  : NULL
##   ..$ margin      : 'margin' num [1:4] 0points 0points 2.2points 0points
##   ... -- attr(*, "unit")= int 8
##   ..$ debug       : NULL
##   ..$ inherit.blank: logi TRUE
##   ... attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.x.bottom    : NULL
## $ axis.text.y      :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : NULL
##   ..$ size        : NULL
##   ..$ hjust       : num 1
##   ..$ vjust       : NULL
##   ..$ angle       : NULL
##   ..$ lineheight  : NULL
##   ..$ margin      : 'margin' num [1:4] 0points 2.2points 0points 0points
##   ... -- attr(*, "unit")= int 8
##   ..$ debug       : NULL
##   ..$ inherit.blank: logi TRUE
##   ... attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.y.left     : NULL
## $ axis.text.y.right    :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : NULL
##   ..$ size        : NULL
##   ..$ hjust       : num 0
##   ..$ vjust       : NULL
##   ..$ angle       : NULL
##   ..$ lineheight  : NULL
##   ..$ margin      : 'margin' num [1:4] 0points 0points 0points 2.2points
##   ... -- attr(*, "unit")= int 8
##   ..$ debug       : NULL
##   ..$ inherit.blank: logi TRUE
##   ... attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.ticks      :List of 6
##   ..$ colour      : chr "grey20"
##   ..$ size        : NULL
##   ..$ linetype    : NULL
##   ..$ lineend     : NULL
```

```
## ..$ arrow      : logi FALSE
## ..$ inherit.blank: logi TRUE
## ... attr(*, "class")= chr [1:2] "element_line" "element"
## $ axis.ticks.x          : NULL
## $ axis.ticks.x.top      : NULL
## $ axis.ticks.x.bottom   : NULL
## $ axis.ticks.y          : NULL
## $ axis.ticks.y.left     : NULL
## $ axis.ticks.y.right    : NULL
## $ axis.ticks.length     : 'simpleUnit' num 2.75points
## ...- attr(*, "unit")= int 8
## $ axis.ticks.length.x   : NULL
## $ axis.ticks.length.x.top: NULL
## $ axis.ticks.length.x.bottom: NULL
## $ axis.ticks.length.y   : NULL
## $ axis.ticks.length.y.left: NULL
## $ axis.ticks.length.y.right: NULL
## $ axis.line       : list()
## ...- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ axis.line.x        : NULL
## $ axis.line.x.top    : NULL
## $ axis.line.x.bottom : NULL
## $ axis.line.y        : NULL
## $ axis.line.y.left   : NULL
## $ axis.line.y.right  : NULL
## $ legend.background  :List of 5
##   ..$ fill      : NULL
##   ..$ colour    : logi NA
##   ..$ size      : NULL
##   ..$ linetype  : NULL
##   ..$ inherit.blank: logi TRUE
## ...- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ legend.margin     : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.
5points
## ...- attr(*, "unit")= int 8
## $ legend.spacing     : 'simpleUnit' num 11points
## ...- attr(*, "unit")= int 8
## $ legend.spacing.x   : NULL
## $ legend.spacing.y   : NULL
## $ legend.key        :List of 5
##   ..$ fill      : chr "white"
##   ..$ colour    : logi NA
##   ..$ size      : NULL
##   ..$ linetype  : NULL
##   ..$ inherit.blank: logi TRUE
## ...- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ legend.key.size    : 'simpleUnit' num 1.2lines
## ...- attr(*, "unit")= int 3
## $ legend.key.height  : NULL
## $ legend.key.width   : NULL
## $ legend.text       :List of 11
##   ..$ family    : NULL
##   ..$ face      : NULL
##   ..$ colour    : NULL
##   ..$ size      : 'rel' num 0.8
##   ..$ hjust     : NULL
```

```
## ..$ vjust      : NULL
## ..$ angle      : NULL
## ..$ lineheight : NULL
## ..$ margin     : NULL
## ..$ debug      : NULL
## ..$ inherit.blank: logi TRUE
## ...- attr(*, "class")= chr [1:2] "element_text" "element"
## $ legend.text.align      : NULL
## $ legend.title           :List of 11
## ..$ family      : NULL
## ..$ face        : NULL
## ..$ colour      : NULL
## ..$ size        : NULL
## ..$ hjust       : num 0
## ..$ vjust       : NULL
## ..$ angle       : NULL
## ..$ lineheight  : NULL
## ..$ margin      : NULL
## ..$ debug       : NULL
## ..$ inherit.blank: logi TRUE
## ...- attr(*, "class")= chr [1:2] "element_text" "element"
## $ legend.title.align      : NULL
## $ legend.position        : chr "right"
## $ legend.direction       : NULL
## $ legend.justification   : chr "center"
## $ legend.box             : NULL
## $ legend.box.just        : NULL
## $ legend.box.margin      : 'margin' num [1:4] 0cm 0cm 0cm 0cm
## ...- attr(*, "unit")= int 1
## $ legend.box.background   : list()
## ...- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ legend.box.spacing     : 'simpleUnit' num 11points
## ...- attr(*, "unit")= int 8
## $ panel.background       :List of 5
## ..$ fill        : chr "white"
## ..$ colour      : logi NA
## ..$ size        : NULL
## ..$ linetype    : NULL
## ..$ inherit.blank: logi TRUE
## ...- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ panel.border        :List of 5
## ..$ fill        : logi NA
## ..$ colour      : chr "grey20"
## ..$ size        : NULL
## ..$ linetype    : NULL
## ..$ inherit.blank: logi TRUE
## ...- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ panel.spacing       : 'simpleUnit' num 5.5points
## ...- attr(*, "unit")= int 8
## $ panel.spacing.x     : NULL
## $ panel.spacing.y     : NULL
## $ panel.grid          :List of 6
## ..$ colour      : chr "grey92"
## ..$ size        : NULL
## ..$ linetype    : NULL
## ..$ lineend     : NULL
```

```
## ..$ arrow      : logi FALSE
## ..$ inherit.blank: logi TRUE
## ... attr(*, "class")= chr [1:2] "element_line" "element"
## $ panel.grid.major      : NULL
## $ panel.grid.minor      :List of 6
##   ..$ colour      : NULL
##   ..$ size        : 'rel' num 0.5
##   ..$ linetype    : NULL
##   ..$ lineend     : NULL
##   ..$ arrow       : logi FALSE
##   ..$ inherit.blank: logi TRUE
## ... attr(*, "class")= chr [1:2] "element_line" "element"
## $ panel.grid.major.x    : NULL
## $ panel.grid.major.y    : NULL
## $ panel.grid.minor.x    : NULL
## $ panel.grid.minor.y    : NULL
## $ panel.ontop      : logi FALSE
## $ plot.background      :List of 5
##   ..$ fill        : NULL
##   ..$ colour      : chr "white"
##   ..$ size        : NULL
##   ..$ linetype    : NULL
##   ..$ inherit.blank: logi TRUE
## ... attr(*, "class")= chr [1:2] "element_rect" "element"
## $ plot.title      :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : NULL
##   ..$ size        : 'rel' num 1.2
##   ..$ hjust       : num 0
##   ..$ vjust       : num 1
##   ..$ angle       : NULL
##   ..$ lineheight  : NULL
##   ..$ margin      : 'margin' num [1:4] 0points 0points 5.5points 0points
## ... ...- attr(*, "unit")= int 8
##   ..$ debug       : NULL
##   ..$ inherit.blank: logi TRUE
## ... attr(*, "class")= chr [1:2] "element_text" "element"
## $ plot.title.position : chr "panel"
## $ plot.subtitle      :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
##   ..$ colour      : NULL
##   ..$ size        : NULL
##   ..$ hjust       : num 0
##   ..$ vjust       : num 1
##   ..$ angle       : NULL
##   ..$ lineheight  : NULL
##   ..$ margin      : 'margin' num [1:4] 0points 0points 5.5points 0points
## ... ...- attr(*, "unit")= int 8
##   ..$ debug       : NULL
##   ..$ inherit.blank: logi TRUE
## ... attr(*, "class")= chr [1:2] "element_text" "element"
## $ plot.caption      :List of 11
##   ..$ family      : NULL
##   ..$ face        : NULL
```

```
## ..$ colour      : NULL
## ..$ size        : 'rel' num 0.8
## ..$ hjust       : num 1
## ..$ vjust       : num 1
## ..$ angle       : NULL
## ..$ lineheight  : NULL
## ..$ margin       : 'margin' num [1:4] 5.5points 0points 0points 0points
## ... -- attr(*, "unit")= int 8
## ..$ debug       : NULL
## ..$ inherit.blank: logi TRUE
## ... -- attr(*, "class")= chr [1:2] "element_text" "element"
## $ plot.caption.position   : chr "panel"
## $ plot.tag            :List of 11
## ..$ family       : NULL
## ..$ face         : NULL
## ..$ colour       : NULL
## ..$ size         : 'rel' num 1.2
## ..$ hjust        : num 0.5
## ..$ vjust        : num 0.5
## ..$ angle        : NULL
## ..$ lineheight   : NULL
## ..$ margin        : NULL
## ..$ debug         : NULL
## ..$ inherit.blank: logi TRUE
## ... -- attr(*, "class")= chr [1:2] "element_text" "element"
## $ plot.tag.position    : chr "topleft"
## $ plot.margin        : 'margin' num [1:4] 5.5points 5.5points 5.5points 5.
5points
## ... -- attr(*, "unit")= int 8
## $ strip.background   :List of 5
## ..$ fill          : chr "grey85"
## ..$ colour        : chr "grey20"
## ..$ size          : NULL
## ..$ linetype      : NULL
## ..$ inherit.blank: logi TRUE
## ... -- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ strip.background.x  : NULL
## $ strip.background.y  : NULL
## $ strip.placement   : chr "inside"
## $ strip.text        :List of 11
## ..$ family       : NULL
## ..$ face         : NULL
## ..$ colour       : chr "grey10"
## ..$ size         : 'rel' num 0.8
## ..$ hjust        : NULL
## ..$ vjust        : NULL
## ..$ angle        : NULL
## ..$ lineheight   : NULL
## ..$ margin       : 'margin' num [1:4] 4.4points 4.4points 4.4points 4.4points
## ... -- attr(*, "unit")= int 8
## ..$ debug       : NULL
## ..$ inherit.blank: logi TRUE
## ... -- attr(*, "class")= chr [1:2] "element_text" "element"
## $ strip.text.x      : NULL
## $ strip.text.y      :List of 11
## ..$ family       : NULL
```

```

## ..$ face      : NULL
## ..$ colour    : NULL
## ..$ size      : NULL
## ..$ hjust     : NULL
## ..$ vjust     : NULL
## ..$ angle     : num -90
## ..$ lineheight: NULL
## ..$ margin    : NULL
## ..$ debug     : NULL
## ..$ inherit.blank: logi TRUE
## ... attr(*, "class")= chr [1:2] "element_text" "element"
## $ strip.switch.pad.grid   : 'simpleUnit' num 2.75points
## ... attr(*, "unit")= int 8
## $ strip.switch.pad.wrap   : 'simpleUnit' num 2.75points
## ... attr(*, "unit")= int 8
## $ strip.text.y.left       :List of 11
## ..$ family     : NULL
## ..$ face       : NULL
## ..$ colour     : NULL
## ..$ size       : NULL
## ..$ hjust      : NULL
## ..$ vjust      : NULL
## ..$ angle      : num 90
## ..$ lineheight: NULL
## ..$ margin     : NULL
## ..$ debug      : NULL
## ..$ inherit.blank: logi TRUE
## ... attr(*, "class")= chr [1:2] "element_text" "element"
## - attr(*, "class")= chr [1:2] "theme" "gg"
## - attr(*, "complete")= logi TRUE
## - attr(*, "validate")= logi TRUE

```

Observations:

1. This graph verifies observations from the previous correlation graph.
2. The density curves verifies observations from the histograms represented above.

Hourly activity trends

Trends in hourly intensities

Hourly intensities heat map very plotted and analyzed to know the probable time that a user is most likely to be active

```
hourly_intensities<-read_csv("Fitbit_data/hourlyIntensities_merged.csv")
```

```

## Rows: 22099 Columns: 4
## — Column specification ——————
## Delimiter: ","
## chr (1): ActivityHour
## dbl (3): Id, TotalIntensity, AverageIntensity
##
## 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.

```

```
glimpse(hourly_intensities)
```

```
## Rows: 22,099
## Columns: 4
## $ Id              <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 15039...
## $ ActivityHour    <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM", "4/1...
## $ TotalIntensity   <dbl> 20, 8, 7, 0, 0, 0, 0, 13, 30, 29, 12, 11, 6, 36, 5...
## $ AverageIntensity <dbl> 0.3333333, 0.1333333, 0.1166667, 0.0000000, 0.0000000, 0.0...
```

```
#cleaning file and converting data types
time_intensities<-hourly_intensities%>%
  drop_na()%>%
  transform(Date_time=mdy_hms(as.character(hourly_intensities$ActivityHour)))
```

```
str(time_intensities)
```

```
## 'data.frame': 22099 obs. of 5 variables:
## $ Id             : num 1503960366 1503960366 1503960366 1503960366 1503960366 ...
## $ ActivityHour   : chr "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00 AM" "4/12/2016 3:00:00 AM" ...
## $ TotalIntensity : num 20 8 7 0 0 0 0 13 30 ...
## $ AverageIntensity: num 0.333 0.133 0.117 0 0 ...
## $ Date_time      : POSIXct, format: "2016-04-12 00:00:00" "2016-04-12 01:00:00" ...
...
```

```
time_intensities$date<-as.Date(time_intensities$Date_time)
```

```
time_intensities$time<-format(as.POSIXct(time_intensities$Date_time),format="%H:%M:%S")
```

```
str(time_intensities)
```

```
## 'data.frame': 22099 obs. of 7 variables:
## $ Id             : num 1503960366 1503960366 1503960366 1503960366 1503960366 ...
## $ ActivityHour   : chr "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00 AM" "4/12/2016 3:00:00 AM" ...
## $ TotalIntensity : num 20 8 7 0 0 0 0 13 30 ...
## $ AverageIntensity: num 0.333 0.133 0.117 0 0 ...
## $ Date_time      : POSIXct, format: "2016-04-12 00:00:00" "2016-04-12 01:00:00" ...
## $ date           : Date, format: "2016-04-12" "2016-04-12" ...
## $ time           : chr "00:00:00" "01:00:00" "02:00:00" "03:00:00" ...
```

```
glimpse(time_intensities)
```

```

## Rows: 22,099
## Columns: 7
## $ Id              <dbl> 1503960366, 1503960366, 1503960366, 1503960366, 15039...
## $ ActivityHour    <chr> "4/12/2016 12:00:00 AM", "4/12/2016 1:00:00 AM", "4/1...
## $ TotalIntensity   <dbl> 20, 8, 7, 0, 0, 0, 0, 13, 30, 29, 12, 11, 6, 36, 5...
## $ AverageIntensity <dbl> 0.3333333, 0.1333333, 0.1166667, 0.0000000, 0.0000000, 0.0...
## $ Date_time        <dttm> 2016-04-12 00:00:00, 2016-04-12 01:00:00, 2016-04-12...
## $ date             <date> 2016-04-12, 2016-04-12, 2016-04-12, 2016-04-12, 2016...
## $ time              <chr> "00:00:00", "01:00:00", "02:00:00", "03:00:00", "04:0...

```

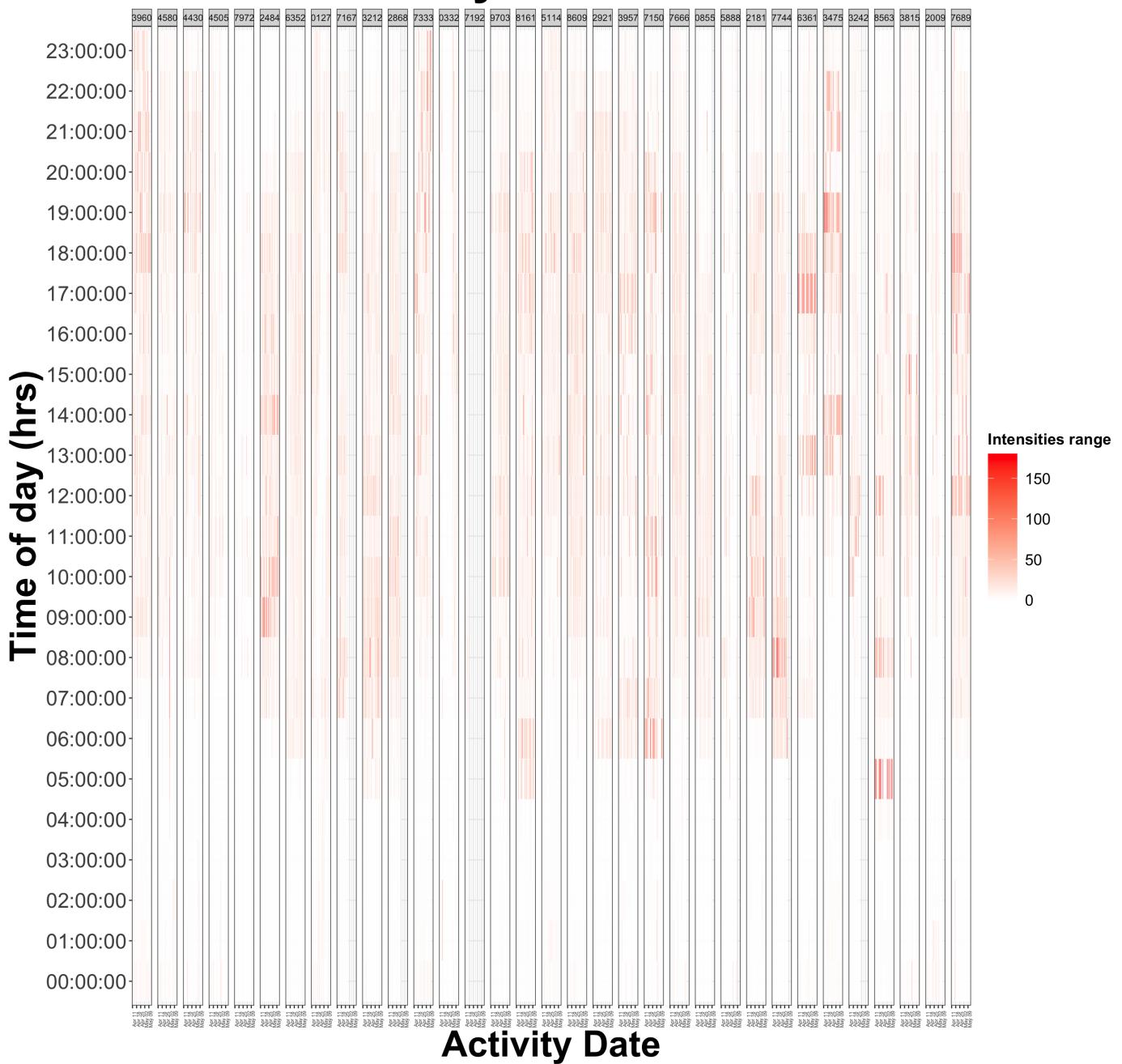
Representing hourly intensities with a heat map

```

ggplot(time_intensities, aes(x=date, y=time, fill = TotalIntensity)) +
  geom_tile(colour = "white") + facet_grid(~Id) + scale_fill_gradient(low="white", hi
gh="red") + xlab("Activity Date") + ylab("Time of day (hrs)") + ggtitle("Hourly Inte
nsities") + labs(fill = "Intensities range")+
  theme_bw()+
  theme(plot.title = element_text(hjust = 0.5),axis.text.x = element_text(angle=90,si
ze=5), axis.text.y = element_text(angle=0,size=20),axis.title=element_text(size=35),ax
is.title=element_text(size=35,face="bold"), title = element_text(size=40,face="bold"
), legend.text = element_text(size=15), legend.key.size = unit(1, 'cm'),legend.key.he
ight = unit(1, 'cm'), legend.key.width = unit(1, 'cm'),legend.title = element_text(si
ze=15))

```

Hourly Intensities



Observations:

1. Each user has a different movement and exercise regime, so hourly intensities heat map does not project a general trend.
2. Reading of intensities being close to zero from ~00:00-07:00 hrs suggests users use fitbit even while sleeping.

Classifying intensity of activity in different ranges based on time of day

```

hourly_intensity_data<-time_intensities%>%
  group_by(hour_of_day=time)%>%
  summarise(hour_zero=sum(TotalIntensity==0),intensity_lessthan50=sum(TotalIntensity<50),
            intensity_between_50_100=(sum(TotalIntensity>50) - sum(TotalIntensity>100)), intensity_morethan_1k=sum(TotalIntensity>100))%>%
  gather(Total, Value,-hour_of_day)
hourly_intensity_data$Total<-ordered(hourly_intensity_data$Total, levels=c("hour_zero", "intensity_lessthan50", "intensity_between_50_100", "intensity_morethan_1k"))

head(hourly_intensity_data)

```

```

## # A tibble: 6 × 3
##   hour_of_day Total     Value
##   <chr>        <ord>    <int>
## 1 00:00:00    hour_zero    616
## 2 01:00:00    hour_zero    701
## 3 02:00:00    hour_zero    730
## 4 03:00:00    hour_zero    792
## 5 04:00:00    hour_zero    788
## 6 05:00:00    hour_zero    729

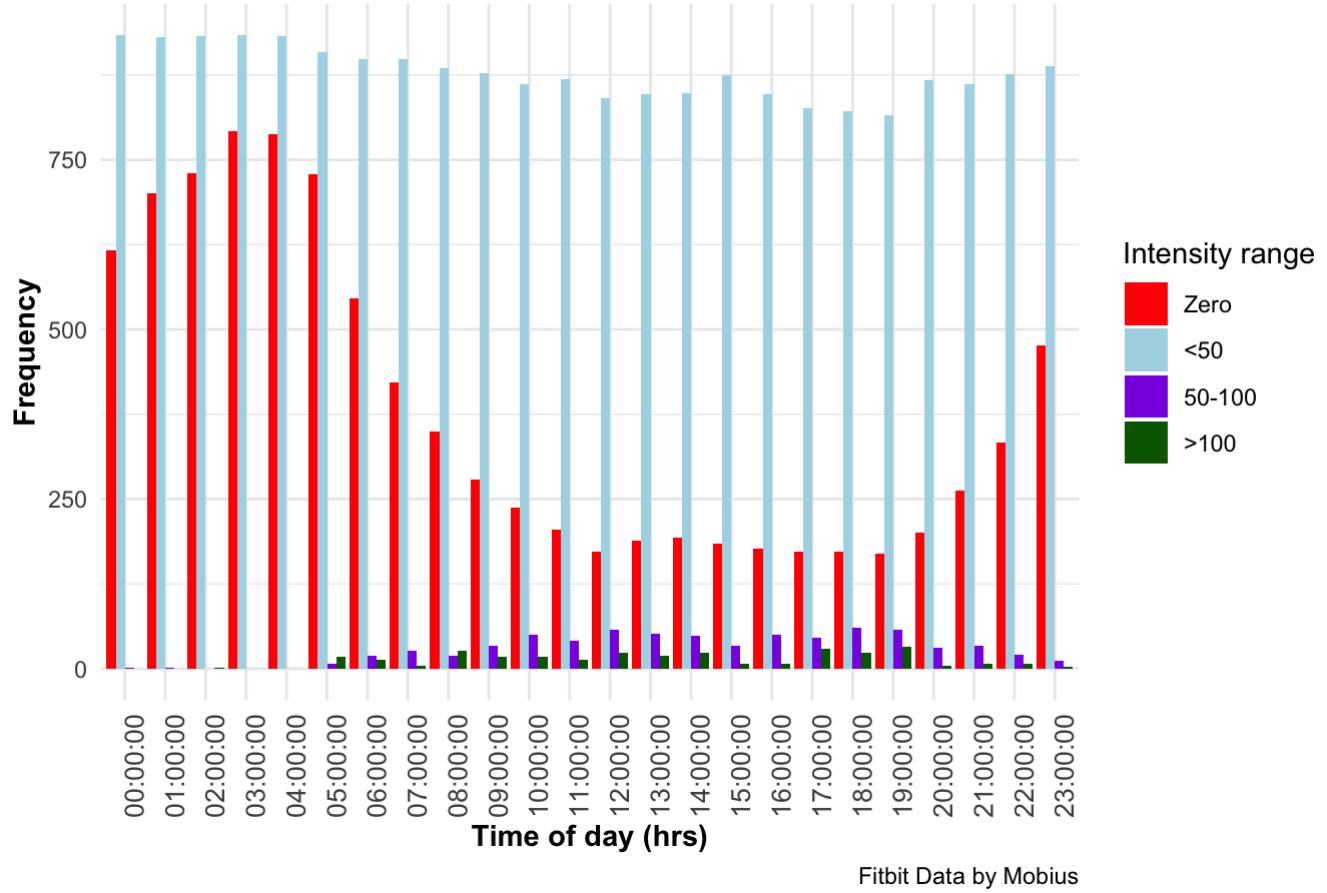
```

```

ggplot(data=hourly_intensity_data,aes(x=hour_of_day,y=value,fill=Total))+ 
  geom_col(position="dodge")+
  theme_minimal()+
  labs(title="Hourly Intensities",x="Time of day (hrs)",y="Frequency",caption="Fitbit Data by Mobius")+
  guides(fill=guide_legend(title="Intensity range"))+
  scale_fill_manual(values=c("red", "lightblue", "blueviolet", "darkgreen"), name="Intensity",breaks=c("hour_zero", "intensity_lessthan50", "intensity_between_50_100", "intensity_morethan_1k"),labels=c("Zero", "<50", "50-100",">100"))+theme(axis.text.x = element_text(angle=90,size=10))+
  theme(plot.title = element_text(hjust = 0.5, face = "bold"), axis.title = element_text(face = "bold"))

```

Hourly Intensities



Observations:

1. Users exhibit low intensities (<50) throughout the day.
2. Zero intensities are significant between 23:00-5:00 hrs.
3. Moderate and high intensities of activity are observed between 5:00-23:00 hrs.
4. Significant moderate and high intensities of activity are observed between 10:00-20:00 hrs.

Trends in hourly steps

Hourly steps data was read from a .csv and was processed for analysis

```
hourlySteps_merged <- read_csv("Fitbit_data/hourlySteps_merged.csv")
```

```
## Rows: 22099 Columns: 3
## — Column specification —
## Delimiter: ","
## chr (1): ActivityHour
## 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.
```

```
drop_na(hourlySteps_merged)
```

```

## # A tibble: 22,099 × 3
##       Id ActivityHour      StepTotal
##   <dbl> <chr>           <dbl>
## 1 1503960366 4/12/2016 12:00:00 AM     373
## 2 1503960366 4/12/2016 1:00:00 AM      160
## 3 1503960366 4/12/2016 2:00:00 AM      151
## 4 1503960366 4/12/2016 3:00:00 AM       0
## 5 1503960366 4/12/2016 4:00:00 AM       0
## 6 1503960366 4/12/2016 5:00:00 AM       0
## 7 1503960366 4/12/2016 6:00:00 AM       0
## 8 1503960366 4/12/2016 7:00:00 AM       0
## 9 1503960366 4/12/2016 8:00:00 AM      250
## 10 1503960366 4/12/2016 9:00:00 AM    1864
## # ... with 22,089 more rows

```

```
head(hourlySteps_merged)
```

```

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

```

```
str(hourlySteps_merged)
```

```

## spec_tbl_df [22,099 × 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
## $ Id          : num [1:22099] 1503960366 1503960366 1503960366 1503960366 1503960
366 ...
## $ ActivityHour: chr [1:22099] "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/1
2/2016 2:00:00 AM" "4/12/2016 3:00:00 AM" ...
## $ StepTotal   : num [1:22099] 373 160 151 0 0 ...
## - attr(*, "spec")=
##   .. cols(
##     ..   Id = col_double(),
##     ..   ActivityHour = col_character(),
##     ..   StepTotal = col_double()
##     .. )
## - attr(*, "problems")=<externalptr>

```

```

# changing type of data sets
hourlySteps<-hourlySteps_merged%>%
  transform(Date_time=mdy_hms(as.character(hourlySteps_merged$ActivityHour)))

```

```
hourlySteps$date<-as.Date(hourlySteps$Date_time)
```

```
hourlySteps$time<-format(as.POSIXct(hourlySteps$Date_time),format="%H:%M:%S")
```

```
str(hourlySteps)
```

```
## 'data.frame': 22099 obs. of 6 variables:  
## $ Id : num 1503960366 1503960366 1503960366 1503960366 1503960366 ...  
## $ ActivityHour: chr "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016 2:00:00 AM" "4/12/2016 3:00:00 AM" ...  
## $ StepTotal : num 373 160 151 0 0 ...  
## $ Date_time : POSIXct, format: "2016-04-12 00:00:00" "2016-04-12 01:00:00" ...  
## $ date : Date, format: "2016-04-12" "2016-04-12" ...  
## $ time : chr "00:00:00" "01:00:00" "02:00:00" "03:00:00" ...
```

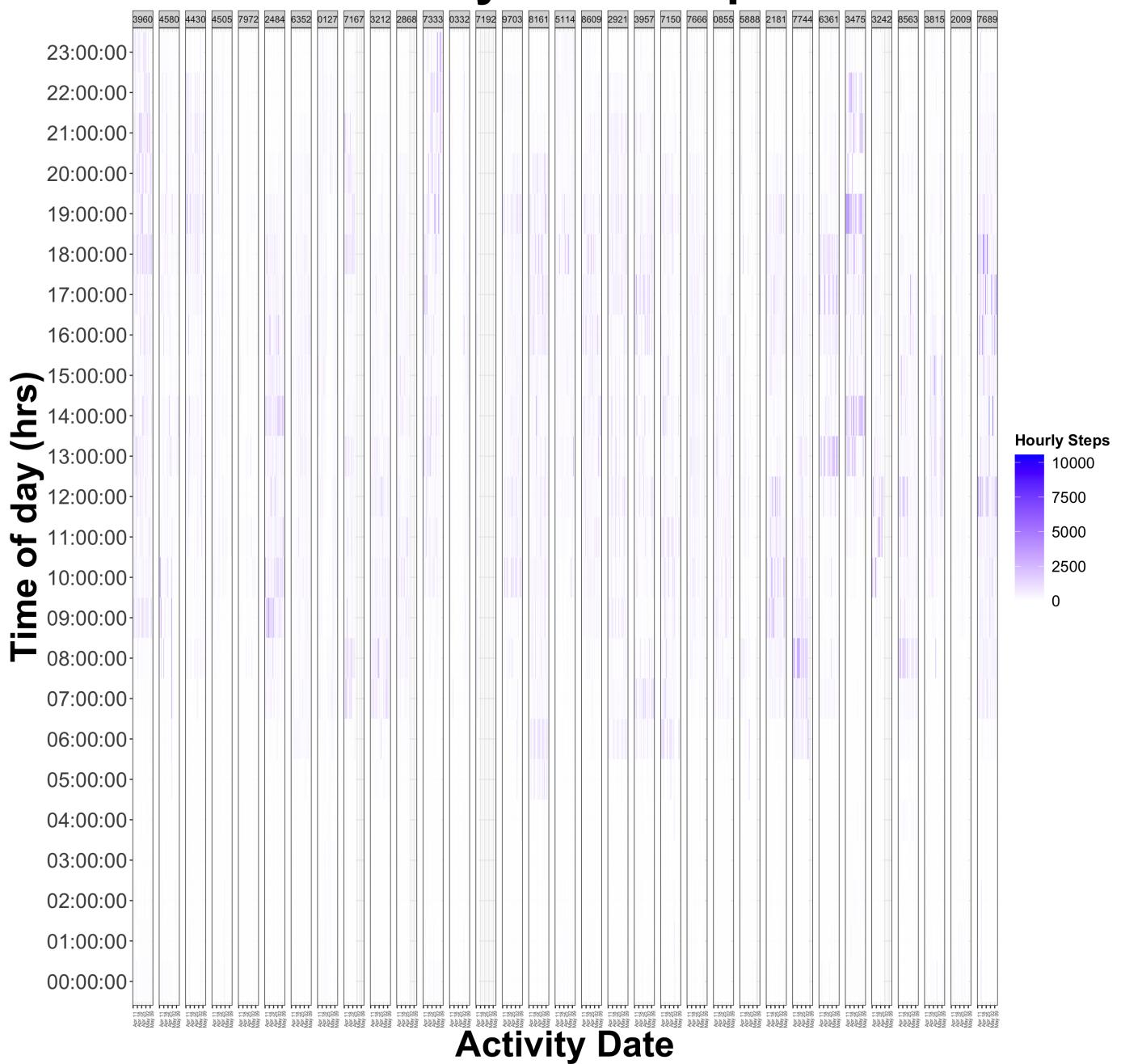
```
max(hourlySteps$StepTotal) #max hourly step
```

```
## [1] 10554
```

Plot for hourly steps

```
ggplot(hourlySteps, aes(x=date, y=time, fill = StepTotal)) +  
  geom_tile(colour = "white") + facet_grid(~Id) + scale_fill_gradient(low="white", high="blue") + xlab("Activity Date") + ylab("Time of day (hrs)") + ggtitle("Hourly Total Steps") + labs(fill = "Hourly Steps") +  
  theme_bw() + theme(plot.title = element_text(hjust = 0.5), axis.text.x = element_text(angle=90, size=5), axis.text.y = element_text(angle=0, size=20), axis.title = element_text(size=35), axis.title = element_text(size=35, face="bold"), title = element_text(size=40, face="bold"), legend.text = element_text(size=15), legend.key.size = unit(1, 'cm'), legend.key.height = unit(1, 'cm'), legend.key.width = unit(1, 'cm'), legend.title = element_text(size=15))
```

Hourly Total Steps



Observations:

Similar to the previous heatmap for hourly intensities, users do not show a common trend in terms of hourly steps.

Classifying hourly steps into ranges based on time of day

```

hourlySteps_data<-hourlySteps%>%
  group_by(hour_of_day=time)%>%
  summarise(steps_zero=sum(StepTotal==0), steps_lessthan2500=sum(StepTotal<2500), steps_between_2500_5000=(sum(StepTotal>2500) - sum(StepTotal>5000)), steps_between_5000_7500=(sum(StepTotal>5000) - sum(StepTotal>7500)), steps_morethan_7500=sum(StepTotal>7500))%>%
  gather(Total, Value,-hour_of_day)
hourlySteps_data$Total<-ordered(hourlySteps_data$Total, levels=c("steps_zero", "steps_lessthan2500", "steps_between_2500_5000", "steps_between_5000_7500", "steps_morethan_7500"))

head(hourlySteps_data)

```

```

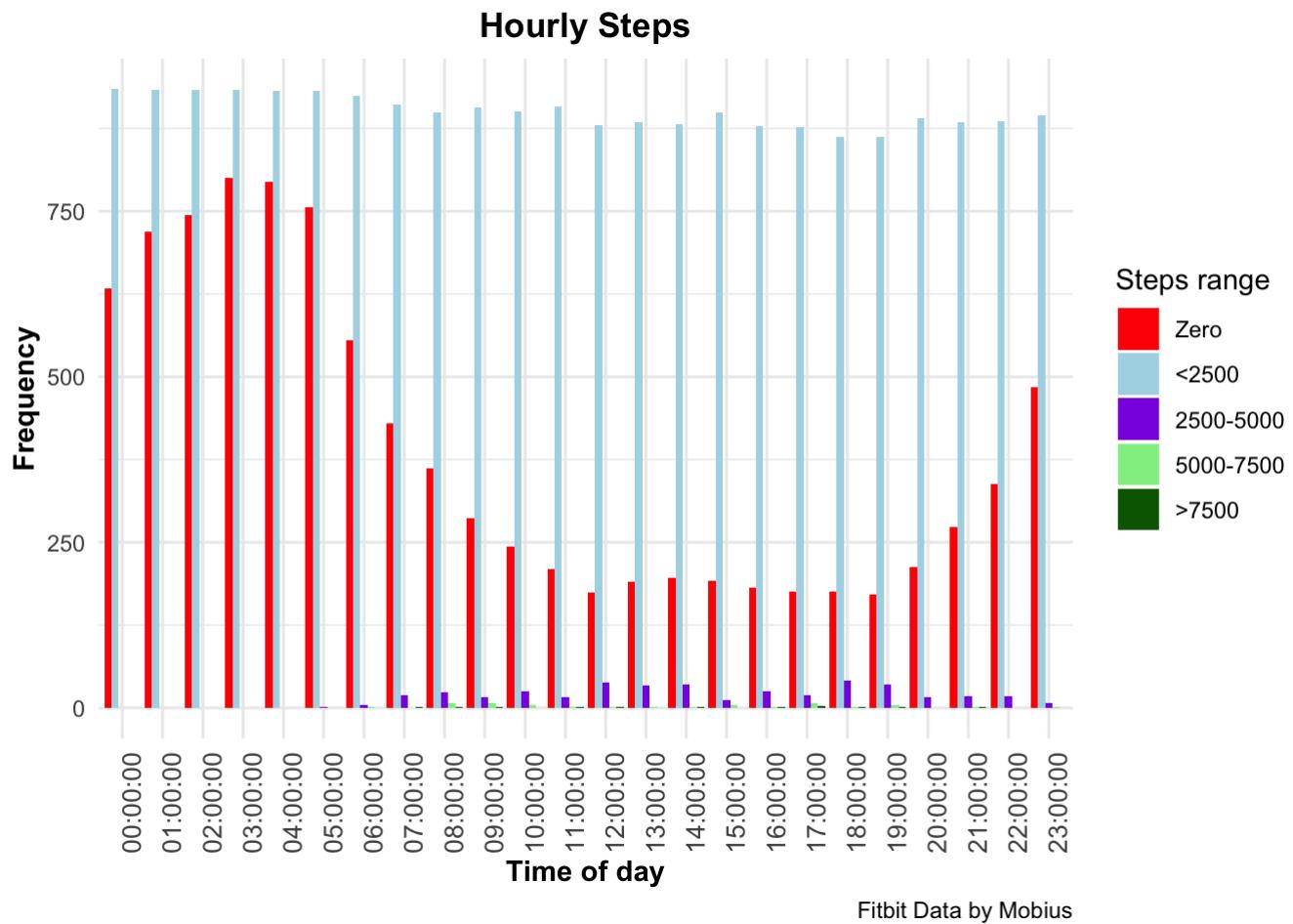
## # A tibble: 6 × 3
##   hour_of_day Total      Value
##   <chr>        <ord>     <int>
## 1 00:00:00    steps_zero    633
## 2 01:00:00    steps_zero    719
## 3 02:00:00    steps_zero    744
## 4 03:00:00    steps_zero    800
## 5 04:00:00    steps_zero    795
## 6 05:00:00    steps_zero    756

```

```

ggplot(data=hourlySteps_data,aes(x=hour_of_day,y=Value,fill=Total))+ 
  geom_col(position="dodge")+
  theme_minimal()+
  labs(title="Hourly Steps",x="Time of day",y="Frequency",caption="Fitbit Data by Mobius")+
  guides(fill=guide_legend(title="Steps range"))+
  scale_fill_manual(values=c("red", "lightblue", "blueviolet", "lightgreen","darkgreen"), name="Steps",breaks=c("steps_zero", "steps_lessthan2500", "steps_between_2500_5000", "steps_between_5000_7500","steps_morethan_7500"),labels=c("Zero", "<2500", "2500-5000","5000-7500",>7500))+theme(axis.text.x = element_text(angle=90,size=10))+
  theme(plot.title = element_text(hjust = 0.5, face = "bold"), axis.title = element_text(face = "bold"))

```



Observations:

1. Users move less between 00:00-06:00 hrs.
 2. Users show significant movement (hourly steps>2500) between 08:00-22:00 hrs.

Sleep activity among users

Reading sleep data provided, cleaning and processing it

```
drop_na(minutesleep_merged)
```

```

## # A tibble: 188,521 × 4
##       Id date           value     logId
##   <dbl> <chr>        <dbl>    <dbl>
## 1 1503960366 4/12/2016 2:47:30 AM     3 11380564589
## 2 1503960366 4/12/2016 2:48:30 AM     2 11380564589
## 3 1503960366 4/12/2016 2:49:30 AM     1 11380564589
## 4 1503960366 4/12/2016 2:50:30 AM     1 11380564589
## 5 1503960366 4/12/2016 2:51:30 AM     1 11380564589
## 6 1503960366 4/12/2016 2:52:30 AM     1 11380564589
## 7 1503960366 4/12/2016 2:53:30 AM     1 11380564589
## 8 1503960366 4/12/2016 2:54:30 AM     2 11380564589
## 9 1503960366 4/12/2016 2:55:30 AM     2 11380564589
## 10 1503960366 4/12/2016 2:56:30 AM    2 11380564589
## # ... with 188,511 more rows

```

```
head(minuteSleep_merged)
```

```

## # A tibble: 6 × 4
##       Id date           value     logId
##   <dbl> <chr>        <dbl>    <dbl>
## 1 1503960366 4/12/2016 2:47:30 AM     3 11380564589
## 2 1503960366 4/12/2016 2:48:30 AM     2 11380564589
## 3 1503960366 4/12/2016 2:49:30 AM     1 11380564589
## 4 1503960366 4/12/2016 2:50:30 AM     1 11380564589
## 5 1503960366 4/12/2016 2:51:30 AM     1 11380564589
## 6 1503960366 4/12/2016 2:52:30 AM     1 11380564589

```

```

#changing type of columns in dataframe
minuteSleep<-minuteSleep_merged%>%
  transform(Date_time=mdy_hms(as.character(minuteSleep_merged$date)))

```

```

minuteSleep$date_activity<-as.Date(minuteSleep$Date_time)

minuteSleep$time<-format(as.POSIXct(minuteSleep$Date_time),format="%H:%M:%S")

minuteSleep$time<- as.POSIXct(minuteSleep$time, format="%H:%M:%S")
minuteSleep@Id<-as.character(minuteSleep@Id)
str(minuteSleep)

```

```

## 'data.frame': 188521 obs. of 7 variables:
## $ Id      : chr "1503960366" "1503960366" "1503960366" "1503960366" ...
## $ date    : chr "4/12/2016 2:47:30 AM" "4/12/2016 2:48:30 AM" "4/12/2016 2:49:30 AM" "4/12/2016 2:50:30 AM" ...
## $ value   : num 3 2 1 1 1 1 2 2 2 ...
## $ logId   : num 11380564589 11380564589 11380564589 11380564589 11380564589 ...
## $ Date_time: POSIXct, format: "2016-04-12 02:47:30" "2016-04-12 02:48:30" ...
## $ date_activity: Date, format: "2016-04-12" "2016-04-12" ...
## $ time    : POSIXct, format: "2022-05-13 02:47:30" "2022-05-13 02:48:30" ...

```

Statistics of Sleep data provided by users

```
Count_sleepdata <- minuteSleep%>%
  count(Id,date_activity)
head(Count_sleepdata)
```

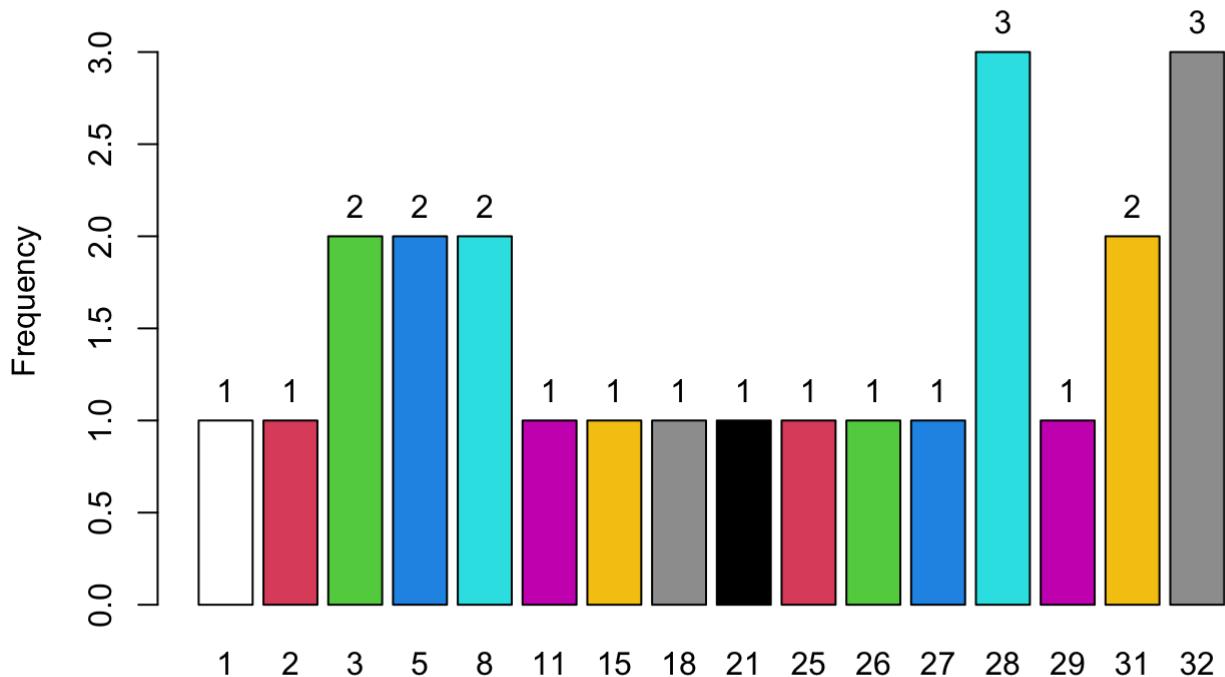
```
##           Id date_activity   n
## 1 1503960366 2016-04-12 346
## 2 1503960366 2016-04-13 407
## 3 1503960366 2016-04-15 442
## 4 1503960366 2016-04-16 400
## 5 1503960366 2016-04-17 679
## 6 1503960366 2016-04-19 320
```

```
SleepCount_per_Id<-Count_sleepdata%>%
  count(Id)
str(SleepCount_per_Id)
```

```
## 'data.frame': 24 obs. of 2 variables:
## $ Id: chr "1503960366" "1644430081" "1844505072" "1927972279" ...
## $ n : int 25 5 5 8 32 1 18 28 11 29 ...
```

```
library(epiDisplay)
tab1(SleepCount_per_Id$n, cum.percent = TRUE, main = 'Daily sleep data provided by users', xlab='Number of days of sleep')
```

Daily sleep data provided by users



```

## SleepCount_per_Id$n :
##          Frequency Percent Cum. percent
## 1              1     4.2      4.2
## 2              1     4.2      8.3
## 3              2     8.3     16.7
## 5              2     8.3     25.0
## 8              2     8.3     33.3
## 11             1     4.2     37.5
## 15             1     4.2     41.7
## 18             1     4.2     45.8
## 21             1     4.2     50.0
## 25             1     4.2     54.2
## 26             1     4.2     58.3
## 27             1     4.2     62.5
## 28             3    12.5     75.0
## 29             1     4.2     79.2
## 31             2     8.3     87.5
## 32             3    12.5    100.0
##   Total        24   100.0    100.0

```

Observations:

1. Total 24 users provided sleep data of which about 12 users provided sleep data for 25 days and above.
2. Reason for lower data could be : (a) User charges device at night or (b) It is cumbersome to use the device while sleeping.

Hourly sleep levels of users

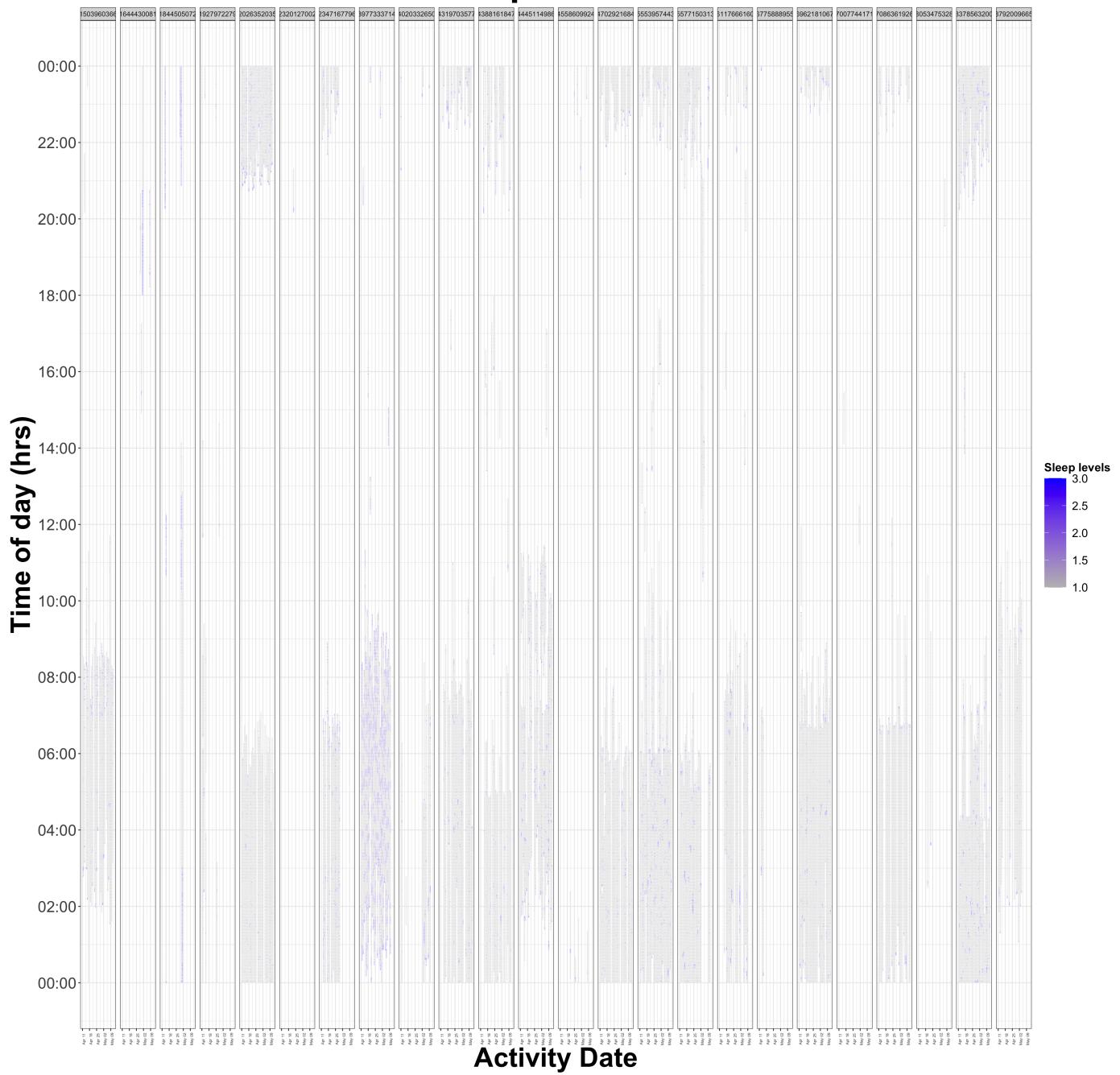
Representing sleep data of users in terms of sleep levels

```

ggplot(minuteSleep, aes(x=date_activity, y=time, fill = value)) +
  geom_tile(colour = "white") + facet_grid(~Id) + scale_fill_gradient(low="gray", high="blue") + xlab("Activity Date") + ylab("Time of day (hrs)") + ggtitle("Sleep Monitor") + labs(fill = "Sleep levels")+
  theme_bw() + theme(plot.title = element_text(hjust = 0.5), axis.text.x = element_text(angle=90, size=5), axis.text.y = element_text(angle=0, size=20), axis.title=element_text(size=35), axis.title=element_text(size=35, face="bold"), title = element_text(size=40, face="bold"), legend.text = element_text(size=15), legend.key.size = unit(1, 'cm'), legend.key.height = unit(1, 'cm'), legend.key.width = unit(1, 'cm'), legend.title = element_text(size=15))+
  scale_y_datetime(date_breaks = "2 hours", date_labels="%H:%M")

```

Sleep Monitor



Observations:

1. Users seem to have different sleep hours and duration.
2. Users consistently sleep between 02:00-06:00 hrs.
3. Unlike while being awake, not all users wear the fitness device to sleep regularly.
4. Only 1 user exhibits consistent level 3 in sleep data.

Heart rate data

Understanding the importance of heart rate data obtained in Fitbit among users

```
#loading relevant file and cleaning
heartrate_seconds_merged <- read_csv("Fitbit_data/heartrate_seconds_merged.csv")
```

```
## Rows: 2483658 Columns: 3
## — Column specification —
## Delimiter: ","
## chr (1): Time
## dbl (2): Id, Value
##
## 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.
```

```
head(heartrate_seconds_merged)
```

```
## # A tibble: 6 × 3
##       Id Time           Value
##   <dbl> <chr>        <dbl>
## 1 2022484408 4/12/2016 7:21:00 AM    97
## 2 2022484408 4/12/2016 7:21:05 AM   102
## 3 2022484408 4/12/2016 7:21:10 AM   105
## 4 2022484408 4/12/2016 7:21:20 AM   103
## 5 2022484408 4/12/2016 7:21:25 AM   101
## 6 2022484408 4/12/2016 7:22:05 AM   95
```

```
#cleaning and changing type of data in columns of dataframe
heartrate_seconds<-heartrate_seconds_merged%>%
  drop_na()%>%
  transform(Date_time=mdy_hms(as.character(heartrate_seconds_merged$Time)))
```

```
heartrate_seconds$date_activity<-as.Date(heartrate_seconds$Date_time)
heartrate_seconds$Id<-as.character(heartrate_seconds$Id)
```

```
Count_heartdata <- heartrate_seconds%>%
  count(Id,date_activity)
head(Count_heartdata)
```

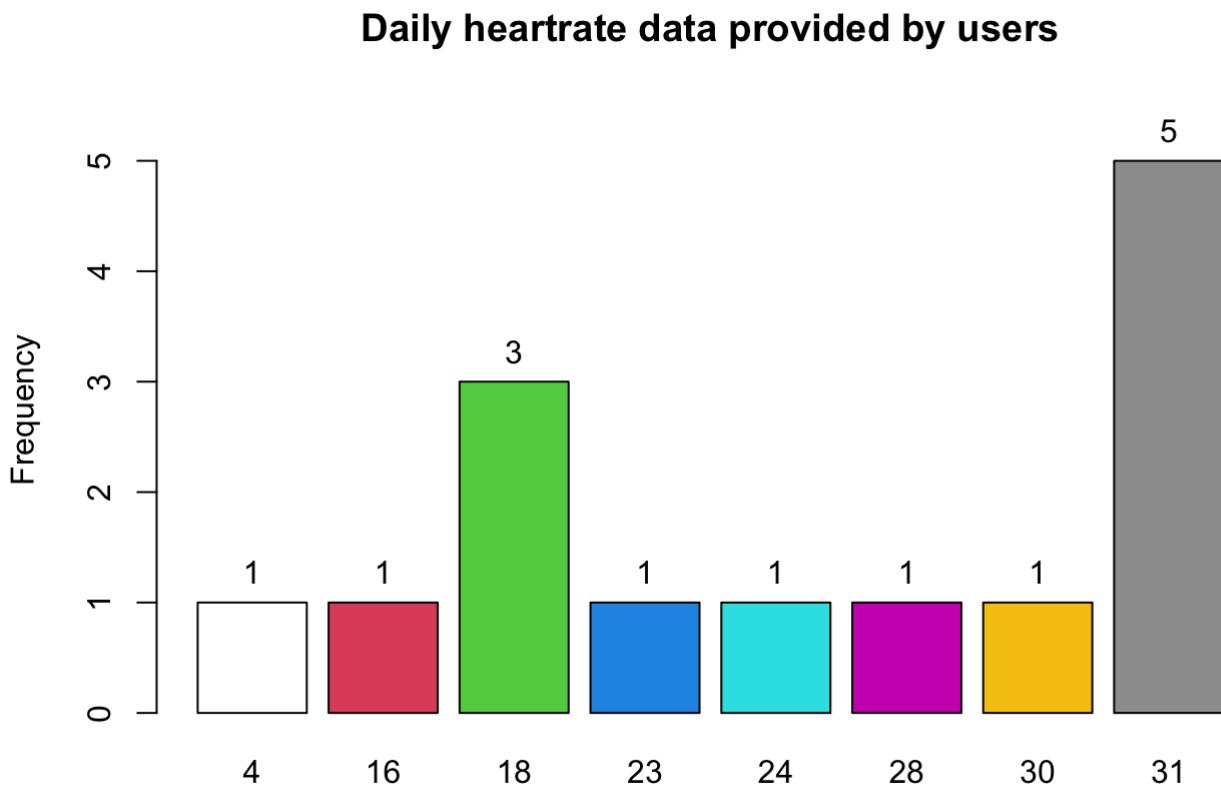
```
##       Id date_activity     n
## 1 2022484408 2016-04-12 4836
## 2 2022484408 2016-04-13 5332
## 3 2022484408 2016-04-14 5560
## 4 2022484408 2016-04-15 5302
## 5 2022484408 2016-04-16 3143
## 6 2022484408 2016-04-17 4948
```

```
Heartdata_per_Id<-Count_heartdata%>%
  count(Id)
str(Heartdata_per_Id)
```

```
## 'data.frame': 14 obs. of 2 variables:
## $ Id: chr "2022484408" "2026352035" "2347167796" "4020332650" ...
## $ n : int 31 4 18 16 30 31 31 28 23 18 ...
```

Statistics of heart rate data provided by users

```
tab1(Heartdata_per_Id$n, cum.percent = TRUE, main = 'Daily heartrate data provided by users')
```



```
## Heartdata_per_Id$n :
##             Frequency Percent Cum. percent
## 4                  1     7.1      7.1
## 16                 1     7.1     14.3
## 18                 3    21.4     35.7
## 23                 1     7.1     42.9
## 24                 1     7.1     50.0
## 28                 1     7.1     57.1
## 30                 1     7.1     64.3
## 31                 5    35.7    100.0
##   Total            14    100.0    100.0
```

Observations:

1. About 14 users provided heart rate data.
2. Among these users only 7 users provided data for above 27 days.
3. Lesser data suggests, users are not as keen in recording/sharing heart rate per second as much as their intensities of activities.

Consolidated Trends and Observations

1. In a day, users move an average of ~7500 steps.
2. Minimum steps moved is zero and maximum steps moved is 36019.

3. User activity recorded a common movement of 0-10000 steps daily. It was highest on Mondays and Tuesdays.
4. Users very less likely to move above 20000 steps daily.
5. User activity exhibited a high movement of 10000-20000 steps on a Sunday.
6. User activity showed significant movement (steps>20000) on a Thursday.
7. Most users burned 1000-3000 calories daily. The trends show high calories burned on Sundays and Mondays.
8. Few user data sets suggest daily calories above 3000. Average daily calories of all users is 2282.
9. User activity exhibited most sedentary behavior on Sundays and least sedentary behavior on Mondays.
10. Almost all users were lightly active daily and many of them showed moderate activity.
11. Among all 33 users, 21 users showed usage for 31 days, 8 users showed activity for 26-30 days and 4 users showed activity for less than 26 days.
12. Users moved an average daily distance of 5 units.
13. Moderate activity was common among all users daily.
14. Average daily calories and steps do not show a linear relation suggesting factors other than steps influence daily calories.
15. Hourly activity trends show that each user have different practices of activity and movement.
16. Users show significant moderate and high intensities between 10 am -8 pm.
17. Users exhibit significant movement between 8 am - 10 pm daily.
18. Less or almost zero intensity/steps between 11 pm-6 am suggests that users wear Fitbit to bed.
19. Not many users have reported sleep activity. This may be due to charging of Fitbit or inconvenience in wearing it during sleep.
20. User sleep activity is consistent between 2-6 am.
21. Only 14 users consented to provide heart rate data. Lesser data suggests, users are not as keen in recording/sharing heart rate per second as much as their intensity of activities.

Shortcomings with Fitbit

The following points can be used to improve development of new line of Bellabeat products.

1. Standard steps per hour in a Fitbit are programmed and the device gives an hourly cue to move. There are times you could be busy in not in a condition to move. General notifications during busy hours could be disturbing.
2. As Fitbit device is worn on the hand, steps are falsely counted by mere hand movement, for example while cooking and performing home chores.
3. User interface is complicated. It cannot be used easily by old aged users.
4. It is designed for unisex use.
5. Fitbit may be an inconvenience to wear during sleep and while performing household chores.

Answers to questions

1. What are the current trends in smart device usage?

Each user is a unique individual hence they do not show identical behavior with respect to daily activities.

Users can be divided into several segments based on their daily activity. They are:

- a. Users with light activity, (b) Users with moderate activity, (c) Users with heavy activity. Among these groups, users with light and moderate activity are the most common.
Users showed tendencies to move more on Sundays, Mondays and Tuesdays. This suggests (a) Users move more on Sunday weekend activities such as going on hike/treks/shopping, (b) High movement on Mondays and Tuesdays can be attributed to user excitement in doing exercises at the start of the week.

Average daily distance moved by users is 5 units.

Users moved an average of 7519 steps and burned 2282 calories daily.

Users exhibit movement between 8 am and 10 pm daily.

Very few users do intensive physical activities or walk more than 20k steps daily (In the current case 2).

Not many users have reported sleep activity. This may be due to charging of Fitbit battery or inconvenience in wearing it during sleep.

Fewer heart rate and sleep data suggests users are mainly keen on recording their daily activity (steps, intensities and calories).

2. How could these trends apply to Bellabeat customers?

Bellabeat is a company aimed towards women centric fitness devices.

The current study suggests that every user has a different activity regime and use of fitness devices. Bellabeat customers are also going to exhibit similar traits. The individuality of each user and Bellabeat app's advices for women based on their individual vitals must be highlighted in the marketing campaign.

3. How could these trends influence Bellabeat Marketing strategy?

Bellabeat marketing strategy must be designed towards encouraging the holistic development of individuals (physical and physiological) by the use of their fitness devices. Bellabeat must promote the women-centric goal it thrives on. Bellabeat must provide the reasons for being fit as a women and statistics on current health challenges among women. Bellabeat's marketing campaign must stress that being fit is the most natural way to overcome these challenges. It should work around making more and more women conscious of their physical and psychological well being. The marketing campaign must highlight aspects like wellness, readiness score and fifth vital sign provided by its products.

The individuality of each user and Bellabeat app's advices for women based on their individual vitals must be highlighted in the marketing campaign.

Bellabeat Leaf products such as pendant or clip is not worn on the hand. Hence, less likely that it interferes during sleep or other daily home chores. This feature should be highlighted towards marketing products like Bellabeat Leaf.

The fact that people move more only on few days (Sunday, Monday and Tuesday) in the current study, should be used by Bellabeat to promote daily exercises and their importance. The analytics provided by Bellabeat's app should be used to promote more activeness among users. This can be done by giving badges.

Bellabeat marketing team should focus on promoting women's well being while stating the use of Bellabeat's products. Bellabeat products must be promoted as to being synonymous to better health and well being.

The marketing team may explain all the features of their products elaborately as slides/and documentation on its website and compare its superiority to existing products.

Act

The observations from the current analyses can help in the marketing strategy of Bellabeat app, Leaf and Time products.

The following are few main suggestions would help in marketing and future proofing Bellabeat products:

1. Individuality of users should be preserved providing users with individual-centric profiles to set their daily activity goals. Bellabeat app's advices for women based on their individual vitals must be highlighted.
2. The marketing team may explain all the features of their products elaborately as slides/and documentation on its website and compare its superiority to existing products.
3. Regular wearing of device must be encouraged by providing daily/weekly health analytics to the user.
4. The Bellabeat app can give badges to user based on their weekly/monthly achievements.
5. Users with 100+ days activity streak or many number of badges can be given discounts on personal trainer services, nutrition guidance or other Bellabeat products such as Spring.
6. Bellabeat device should have a provision for users to record/set sleep or work hours so that the fitness device does not prompt them to move when they are busy. It can also provide a facility to integrate with their daily calendars.
7. Bellabeat products can also have special functions to record household chores as daily activities.
8. User interface must be made friendly for users of all age groups. It should be simplified for users 65 years and above.
9. Bellabeat can come up with a product specifically designed for individuals/women above the age of 65 years.
10. Bellabeat's campaign should work around making more and more women conscious of their physical and psychological well being.
11. Modern lifestyle leads to a lot of complications in female reproductive health. Bellabeat membership service can integrate services provided by psychiatrists and gynecologists. The data and analytics provided by the Bellabeat app can help psychiatrists, nutritionists and gynecologists in providing better lifestyle advises to women.
12. Bellabeat membership service can also team up with fitness trainers to provide specialized videos of exercises, weight training, resistance band workouts, dance and yoga for users of beginner, intermediate and advanced levels.
13. Bellabeat can launch personalized fitness products like weights, resistance bands, exercise balls, workout apparel for users.
14. Currently bracelets straps, pendant, chain, watch have limited design and color variations. Increasing variety pendant design or replaceable pendant/bracelet stones/bracelet straps could make it an accessory for users to pair with variety of their dresses. This would encourage the user to wear Bellabeat devices more than often.