Bringing Fitness tracker Analysis and Classification to action

A Case Study for identifying useful features responsible for healthy life

ViveK Nathani (Trent ID:- 0698871)

11/07/2021

Contents

1. Background

As we are currently in a pandemic where health matters the most and monitoring health on daily basis is very difficult without technology. **Today the fitness trackers and wearables, in general, are becoming more and more mainstream, the validity of these devices in their ability to improve health and fitness.** The main phases of wearable technologies are characterized into data collection, processing, and delivery of information, services, and resources for end users.

Simply wearing a fitness tracker won't guarantee a improvement in personal health. Daily monitoring with daily/monthly goal planning and having a proper diet will help achieve the results easily.

Through this case study, we are going to analyze the Fitbit trackers dataset provided by Mobius. **Fitbit device is a high-tech design and health-focused smart device.** The prime focus behind this case study is to take the raw data as an input and produce insightful analytics useful for creating awareness of health amongst the people. For more information and background about Fitbit Devices

For getting our own fitbit tracker device's data and draw analysis on it, you can refer this article.

1.1. Data Source overview

FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius): This Kaggle data set contains personal fitness tracker from thirty fitbit users. Thirty eligible Fitbit users consented to the submission of personal tracker data, including minute-level output for physical activity, heart rate, and sleep monitoring. It includes information about daily activity, steps, and heart rate that can be used to explore users' habits. It's a good database segmented in several tables with different aspects of the data of the device with lots of details about the user behavior.

1.2. Objectives of this case study

Analyze smart device usage data in order to gain insight into how consumers use their Fitbit devices that is searching patterns in how well they use the features in their device. The analysis obtained through this task can be later used for running campaign to drive awareness about health and importance of tracking device in daily life. **This analysis will not only be helpful for people's health monitoring but also for helping the marketing team of Fitbit to plan their marketing strategies.**

Through analysis, I hope to evaluate the next following detailed questions:

- Q1.) How many features are being used by each user?
- Q2.) What are some trends in smart device usage?
- Q3.) Is there any relationship between any kind of feature?
- Q4.) List the features that can help track health better?
- Q5.) How could these trends help influence Fitbit's marketing strategy?

1.3. Approach followed in this case study

I have chosen **exploratory approach** for this analysis by asking questions while looking for trends, patterns, and relationships among variables using regression analysis in the data. The aspect I wanted to explore is users attitude towards the fitness tracker. Such questions to raise as how and how often do they use the product? or how do they respond to the product used?. The rationale behind the use of fitness trackers is that understanding users' routine in using a trackable device would be a good start.

2. Data Description

The whole database consists of 18 data-table out which I will be focusing on few data-tables for this case study and disregarding others as of now. My focus generally lies on analyzing daily patterns and not hourly or minute-level patterns.

2.1. Datasets I will be using

For detecting high-level usage trends, the most interesting data for me is all the **daily activity** and the **sleep data** as they will probably show some interesting patterns but I'll have to merge some tables together to do my analysis. **The data chosen will consist of steps count, calories burnt, hours slept, hours awake, distance covered, weights logged and heart-rate.**

I won't be diving deep into the weight and heart-rate features for this analysis. Will just use those for some basic operations and analysis because the data in those is quite inconsistent and contains more null values than data points.

2.2. Importing the data files from Fitbit dataset

2.2.1. Installation and loading R packages

```
library(tidyverse)          # reading csv
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(flextable)          # for knit the tables to word document
```

```
##
## Attaching package: 'flextable'
```

```
## The following object is masked from 'package:purrr':
##
##     compose
```

```
library(officer)            # for formatting tables
library(ggplot2)              # for plotting graphs
library(gridExtra)          # for combining plots
```

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

```
library(lubridate)          # for mdy()
```

```
##
## Attaching package: 'lubridate'

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

```
library(VennDiagram)          # for venn.diagram()
```

```
## Loading required package: grid

## Loading required package: futile.logger

##
## Attaching package: 'VennDiagram'

## The following object is masked from 'package:flextable':
##
##     rotate
```

```
library(janitor)              # for clean_names()
```

```
##
## Attaching package: 'janitor'

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

```
library(corrplot)           # for plotting cor matrix
```

```
## corrplot 0.89 loaded
```

```
library(stats)              # for cor()
library(ggpubr)             # for ggdonutplot()
```

```
##
## Attaching package: 'ggpubr'

## The following object is masked from 'package:VennDiagram':
##
##     rotate

## The following objects are masked from 'package:flextable':
##
##     border, font, rotate
```

```
library(openair)            # for calender plotting
library(plotly)             # for plot_ly()
```

```
##
## Attaching package: 'plotly'

## The following objects are masked from 'package:flextable':
##
##     highlight, style

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

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

2.2.2. Importing the data

Files from the Fitbit database have been selected and loaded using the below code.

```
path <- "~/Trent University/Data Analytics with R/Assignment - 02/"
daily_activity <- read_csv(paste(path, "dailyActivity_merged.csv", sep =""))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Id = col_double(),
##   ActivityDate = col_character(),
##   TotalSteps = col_double(),
##   TotalDistance = col_double(),
##   TrackerDistance = col_double(),
##   LoggedActivitiesDistance = col_double(),
##   VeryActiveDistance = col_double(),
##   ModeratelyActiveDistance = col_double(),
##   LightActiveDistance = col_double(),
##   SedentaryActiveDistance = col_double(),
##   VeryActiveMinutes = col_double(),
##   FairlyActiveMinutes = col_double(),
##   LightlyActiveMinutes = col_double(),
##   SedentaryMinutes = col_double(),
##   Calories = col_double()
## )
```

```
daily_sleep <- read_csv(paste(path, "sleepDay_merged.csv", sep =""))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Id = col_double(),
##   SleepDay = col_character(),
##   TotalSleepRecords = col_double(),
##   TotalMinutesAsleep = col_double(),
##   TotalTimeInBed = col_double()
## )
```

```
heart_rate <- read_csv(paste(path, "heartrate_seconds_merged.csv", sep =""))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Id = col_double(),
##   Time = col_character(),
##   Value = col_double()
## )
```

```
weight_log <- read_csv(paste(path, "weightLogInfo_merged.csv", sep =""))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Id = col_double(),
##   Date = col_character(),
##   WeightKg = col_double(),
##   WeightPounds = col_double(),
##   Fat = col_double(),
##   BMI = col_double(),
##   IsManualReport = col_logical(),
##   LogId = col_double()
## )
```

2.2.3. Viewing the data

Id	ActivityDate	TotalSteps	TotalDistance	Calories
1,503,960,366	4/12/2016	13,162	8.50	1,985
1,503,960,366	4/13/2016	10,735	6.97	1,797
1,503,960,366	4/14/2016	10,460	6.74	1,776

2.2.3.1. Daily Activity Table The **daily_activity** table contains 940 observation for 33 users with 15 variables. This table represent the activities like **total_distance, total_steps, date_of_activity, calories_burnt** on daily basis.

Columns	Description
Id	User ID
ActivityDate	Date on which task was performed
TotalSteps	Total Steps taken
TotalDistance	Total Distance travelled
TrackerDistance	Total Distance measured
LoggedActivitiesDistance	Difference in distance travelled and measured
VeryActiveDistance	Distance travelled when active
ModeratelyActiveDistance	Distance travelled when moderately active
LightActiveDistance	Distance travelled when lightly active
SedentaryActiveDistance	Distance travelled when sedentarily active
VeryActiveMinutes	Minutes user was Very active
FairlyActiveMinutes	Minutes user was Fairly active
LightlyActiveMinutes	Minutes user was Lightly active
SedentaryMinutes	Minutes user was Sedentarily active
Calories	Total Calories burnt

Id	SleepDay	TotalMinutesAsleep	TotalTimeInBed
1,503,960,366	4/12/2016 12:00:00 AM	327	346
1,503,960,366	4/13/2016 12:00:00 AM	384	407
1,503,960,366	4/15/2016 12:00:00 AM	412	442

2.2.3.2. Daily Sleep Table Looking at the table structure above, we can see that **daily_sleep** table contains 413 observation for 24 users with 5 variables. This table represent the sleep activities specifically and measures the **Number of hours slept and total time in bed** on daily basis.

Columns	Description
Id	User ID
SleepDay	Date on which task was performed
TotalSleepRecords	Total record of sleeps for that user
TotalMinutesAsleep	Total Minutes asleep
TotalTimeInBed	Total Minutes in bed without sleep

Id	Time	Value
2,022,484,408	4/12/2016 7:21:00 AM	97
2,022,484,408	4/12/2016 7:21:05 AM	102

Id	Time	Value
2,022,484,408	4/12/2016 7:21:10 AM	105

2.2.3.3. Heart Rate Table Looking at the table structure above, we can see that **heart_rate** table contains 2483658 observation for 14 users with 3 variables. This table represent the heart beat related information specifically measured at every 5 second interval.

Columns	Description
Id	User ID
Time	Date and Time on which task was performed
Value	Heartbeat count

Id	Date	WeightKg	BMI
1,503,960,366	5/2/2016 11:59:59 PM	52.6	22.65
1,503,960,366	5/3/2016 11:59:59 PM	52.6	22.65
1,927,972,279	4/13/2016 1:08:52 AM	133.5	47.54

2.2.3.4. Weight Log Table Looking at the table structure above, we can see that **weight_log** table contains 67 observation for 8 users with 8 variables. This table represent the weight recorded by the user on specific day in **kG, Pounds unit**. Variables like **BMI and FAT** are also being logged.

Columns	Description
Id	User ID
Date	Date and Time on which task was performed
WeightKg	Weight in KG
WeightPounds	Weight in Pounds
Fat	FAT
BMI	BMI Value
IsManualReport	If the count is reported manually or automatically
LogId	Log ID

2.3. Data Cleaning and Preparation

Steps Included for preparing the dataframes used for analysis

- Cleaning Variable names (using Janitor's clean_names function)
- Transforming the date from %m-%d-%Y format to %Y-%m-%d (using lubricates mdy function)
- Removing redundant variables after transforming and cleaning the data.

```
# Preparing daily_activity table:
daily_activity <- daily_activity %>%
  clean_names() %>%
  mutate(activity_date = mdy(activity_date), day_week = weekdays(activity_date)) %>%
  rename(date = activity_date) %>%
  select(-c(5:10)) # drop redundant columns

daily_activity <- daily_activity[, c(1,2,10,3,9,4:8)]


# Preparing daily_sleep table
daily_sleep <- daily_sleep %>%
  clean_names() %>%
  separate(col = sleep_day, c("date", "sleep_time"), sep = " ") %>%
  mutate(date = mdy(date),
         day_week = weekdays(date),
         total_mins_awake = total_time_in_bed - total_minutes_asleep) %>%
  select(-"sleep_time")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 413 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
daily_sleep <- daily_sleep[,c(1,2,6,3,4,5,7)]


# Preparing weight_log data
weight_log <- weight_log %>%
  clean_names() %>%
  mutate(date = mdy_hms(date)) %>%
  separate(col = date, into = c("date", "datetime"), sep = " ") %>%
  mutate(date = ymd(date))
```

2.4. Data LIMITATION

This analysis can be enhanced more if the data was complete, currently the data is **quite outdated**(data is of 2016, current year 2021). We have grown a lot in every technology and also enhanced the accuracy and data analytics. So with this dataset, **our analysis are restricted by various parameters** like data is only of 1 month and 33 users **(quite small)**. We are also **not aware about user's basic information like gender, age location, etc**.

3. Methodology

As mentioned earlier, I will be using exploratory approach to find the answers to my questions. Methodology will include, **Data Exploration**, **Data Merging**, **Data Bucketing**, **Correlations and Relationship Analysis**.

3.1. Features Data Exploration

Looking at the dataframes, we can see that each dataframe consists of user's ID column. This column can be used to find the relation between features used by that user and also can be used to merge 2 or more dataframes for faster processing.

3.1.1. Interaction of users with features of the product.

```
step_ids <- unique(daily_activity$id, incomparables = FALSE)
sleep_ids <- unique(daily_sleep$id, incomparables = FALSE)
heartrate_ids <- unique(heart_rate$Id, incomparables = FALSE)
weight_ids <- unique(weight_log$id, incomparables = FALSE)

columns_to_plot = list(step_ids, sleep_ids, heartrate_ids, weight_ids)
names = c("Steps counting", "Sleep monitoring", "Heart monitoring", "Weight tracking")

# Ploting the venn diagram

futile.logger::flog.threshold(futile.logger::ERROR, name = "VennDiagramLogger")
venn_diagram <- venn.diagram(x = columns_to_plot,
  category.names = names,
  filename = NULL,
  lwd = 2,
  fill = c("skyblue", "pink1", "mediumorchid", "orange"),
  cex = 1, fontface = "bold", fontfamily = "sans",
  cat.cex = .7, cat.fontface = "bold",
  cat.default.pos = "outer", cat.fontfamily = "sans")

grid.draw(venn_diagram)
```

This 4-set Venn Diagram would represent how 33 users interact with 4 features. We can also find number of users interacting with single feature, two features or multiple features. This Venn diagram helps us in answering our 1st question. Interpretation of this Venn diagram is present in Results section with the diagram.

We have looked into how the users interact with features, now let's dive a little deep and find out how do they interact with those features in day to day life.

3.2. Data Merging

Merging the daily activity with sleep data, so that we can find if any of the daily parameters affect our sleep. We would also be interested in finding the correlation between those parameters and the relation. We are merging the data of only 24 customers who have used both of the above features.

```
daily_activity_sleep <- merge(x=daily_activity, y=daily_sleep,
                                by = c("id", "date", "day_week"))
# shape(413 * 14)

# Removing the duplicated values from the merged dataframe

daily_activity_sleep <- daily_activity_sleep[!duplicated(daily_activity_sleep), ]
# shape(410 * 14)
```

We have merged the dataframes as per our convenience for finding the answers to our questions. Let's look at the merged dataframe, there are total of 14 columns, we would be representing only few below, to show the data has been merged.

id	date	day_week	total_steps	calories	total_minutes_asleep	total_time_in_bed
1,503,960,366	2016-04-12	Tuesday	13,162	1,985	327	346
1,503,960,366	2016-04-13	Wednesday	10,735	1,797	384	407
1,503,960,366	2016-04-15	Friday	9,762	1,745	412	442
1,503,960,366	2016-04-16	Saturday	12,669	1,863	340	367

3.3. Data Bucketing

In this section we are going to categories users into different buckets based on different parameters. The parameter can be **"Distance traveled", "Usage Rate", "Steps taken", "Calories burnt", "No. of hours slept", etc..**

3.3.1. Categorizing the user based on their Usage Rate

Usage Rate is defined as number of days user was active given the total number of days in observation. **Usage rate = Active days / Total day observed**

```
# Building the table of daily usage data grouped by id
usage_daily_id <- daily_activity %>%
    group_by(id) %>%
    summarise(active_days = sum(total_steps != 0),
             total_days = sum(total_steps >= 0),
             usage_rate = active_days / total_days)


# Categorizing the customers into different buckets based on their usage_rate
plot_usage_daily_id <- usage_daily_id%>%
    select(4) %>%
    mutate( user_type = case_when(
        usage_rate == 1 ~ "Committed Users",
        usage_rate < 1 & usage_rate >= 0.70 ~ "Casual Customer",
        usage_rate < 0.70 ~ "Dormant Customer" ))


# Grouping the number of users based on user_type
plot_daily_user_type_usage <- plot_usage_daily_id %>%
    group_by(user_type) %>%
    summarise(count = n(),average_usage = mean(usage_rate))


# plotting user type
plot_daily_user_type_usage %>%
     plot_ly(labels = ~ user_type,
             values = ~ count,
             textinfo = 'value+percent',
             marker = list(colors = c("#ABDDDE","#F8AFA8"),
                           line = list(color = '#000000',
                                       width = 0.75)
             )
     ) %>% add_pie(hole = 0.6) %>%
     layout(title = "User Types", showlegend = TRUE)
```

In the above code, at first we are trying to segregate customers in different groups based on their usage_rate. **According to the definition the customer who has used our product daily are considered as "Committed Customer", the customer whose usage rate goes below 70% are considered as "Inactive or Dormant Customers of Fitbit".** We are using donught chart to represent the number of customers in each such group.

3.3.2. Analyzing Daily trends of active users (usage_rate)

We are creating a calendarplot, which helps us in determining the usage rate trends for each day through out the experiment.

```
# Building the table of daily usage data grouped by date
plot_usage_daily_date <- daily_activity %>%
    group_by(date) %>%
    summarise(active_days = sum(total_steps != 0),
             total_days = sum(total_steps >= 0),
             usage_rate = active_days / total_days)

# plotting usage rate for each day
calendarPlot(plot_usage_daily_date, pollutant = "usage_rate",
            year = 2016,
            month = 4:5,
            annotate = "date",
            main = "Usage Rate by Day")
```

3.3.3. Analyzing Weekly trends of usage_rate

Here we are grouping the daily data by week day. This representation in barchart helps us in understanding the usage rate w.r.t the days in a week.

```
# Building the table of daily usage data grouped by week
plot_usage_weekly <- daily_activity %>%
    group_by(day_week) %>%
    summarise(active_days = sum(total_steps != 0),
             total_days = sum(total_steps >= 0),
             usage_rate = active_days / total_days)

# Building the table of daily usage data grouped by days of a week
plot_usage_daily_weekday <- plot_usage_weekly%>%
    group_by(day_week)%>%
    summarise(average_usage = mean(usage_rate))

# ploting weekday usage in a barchart
weeek_day_usage <- plot_ly(plot_usage_daily_weekday, x = ~ day_week, y = ~ average_usage,
                type = 'bar',
                text = ~ round(average_usage,2), textposition = 'auto',
                marker = list(color = 'rgb(158,202,225)',
                              line = list(color = 'rgb(8,48,107)', width = 1.5)))
```

```
weeek_day_usage <- weeek_day_usage %>% layout(title = "Usage Rate by Weekdays")

weeek_day_usage
```

3.3.4. Categorizing data on various parameters

To make the analysis simpler, the following variables have been categorized:

- Sleep (by hours): < 6h (Insufficient Sleep), 6h - 8h (Sufficient Sleep), > 8h(Over Sleep)
- Calories (by number of calories burnt): < 1500, 1500 - 2500, > 2500
- Steps (by number of steps): < 5000, 5000 - 10000, > 10000
- Distance (by Kilometers): < 5km, 5km - 10 km, > 10km
- Total Inactivity in bed (before sleeping or/and after waking up): < 100 mins, 100 - 200 mins, > 200 mins

```
# Bucketing the users based on Total steps taken by them
daily_activity_sleep <- daily_activity_sleep %>%
  mutate(step_categories = case_when(
    total_steps > 5000 & total_steps <= 10000 ~ "5k - 10k",
    total_steps > 10000 ~ "> 10k",
    TRUE ~ "< 5k"
  )
)

# Bucketing the users based on Total distance traveled by them
daily_activity_sleep <- daily_activity_sleep %>%
 mutate(distance_categories = case_when(
    total_distance > 5 & total_distance <= 10 ~ "5km - 10km",
    total_distance > 10 ~ "> 10km",
    TRUE ~ "< 5km"
  )
)

# Bucketing the users based on Total calories burnt by them
daily_activity_sleep <- daily_activity_sleep %>%
  mutate(calorie_categories = case_when(
    calories > 1500 & calories <= 2500 ~ "1.5k - 2.5k",
    calories > 2500 ~ "> 2.5k",
    TRUE ~ "< 1.5k"
  )
)

# Bucketing the users based on Total hours of sleep taken by them
daily_activity_sleep <- daily_activity_sleep %>%
   mutate(sleep_categories = case_when(
    total_minutes_asleep > 360 & total_minutes_asleep <= 480 ~ "Sufficient Sleep",
    total_minutes_asleep > 480 ~ "Over Sleep",
    TRUE ~ "Insufficient Sleep"
  )
)

# Bucketing the users based on Total hours of inactiveness in bed
daily_activity_sleep <- daily_activity_sleep %>%
  mutate(sleep_inactive_categories = case_when(
    total_mins_awake > 200 ~ "> 200 mins",
    total_mins_awake >= 100 & total_mins_awake <= 200 ~ "100 - 200 mins",
    TRUE ~ "< 100 mins"
  )
)
```

3.4. Relationships Visualization

We are grouping the data by users and aggregating the features data value, so as to analyze and draw relationships.

```
# Creating a data frame for all user's cumulative data of features
average_activity_sleep <- daily_activity_sleep %>%
    group_by(id)%>%
    summarise(average_sleep = mean(total_minutes_asleep),
              average_step = mean(total_steps),
              average_calories = mean(calories),
              average_distance = mean(total_distance),
              average_inactive_in_bed = mean(total_mins_awake),
              active_days = sum(total_steps != 0)) %>%
    mutate(sleep_category = case_when(
        average_sleep > 360 & average_sleep <= 480 ~ "Sufficient Sleep",
        average_sleep > 480 ~ "Over Sleep",
        TRUE ~ "Insufficient Sleep"
    )) %>%
    mutate(step_category = case_when(
        average_step > 5000 & average_step <= 10000 ~ "5k - 10k",
        average_step > 10000 ~ "> 10k",
        TRUE ~ "< 5k"
    )) %>%
    mutate(distance_category = case_when(
        average_distance > 5 & average_distance <= 10 ~ "5km - 10km",
        average_distance > 10 ~ "> 10km",
        TRUE ~ "< 5km"
    )) %>%
    mutate(calorie_category = case_when(
        average_calories > 1500 & average_calories <= 2500 ~ "1.5k - 2.5k",
        average_calories > 2500 ~ "> 2.5k",
        TRUE ~ "< 1.5k"
    )) %>%
    mutate(sleep_inactive_categories = case_when(
      average_inactive_in_bed > 200 ~ "> 200 mins",
      average_inactive_in_bed >= 100 & average_inactive_in_bed <= 200 ~ "100 - 200 mins",
      TRUE ~ "< 100 mins"
  )) %>%
   mutate(user_type = case_when(
        active_days > 26 ~ "Committed Users",
        active_days <= 26 & active_days >= 15 ~ "Casual Customer",
        active_days < 15 ~ "Dormant Customer"
  ))
```

How many users get proper sleep The below code helps us find the number of users in each sleep category.

```
average_activity_sleep1 <- average_activity_sleep %>%
  group_by(sleep_category)%>%
    summarise(count = n())

# Pie chart to represent the number of users in each Sleep Category
plot_ly(average_activity_sleep1, labels = ~sleep_category, values = ~ count,
          type = "pie", domain = list(x = c(0, 0.5), y = c(0, 1)))

# Finding the reason why are the users not getting proper sleep
temp1 <- ggplot(data=average_activity_sleep, aes(x = sleep_category,
                                                  fill = sleep_inactive_categories))+
          geom_bar()+
          labs(x = "Level of Rest", fill = "Level of Inactiveness While in bed")+
          geom_text(stat = "count", aes(label = after_stat(count)),
          position = position_stack(vjust = .5))
```

3.4.1. Comparing 2 plots for understanding the gap of inactiveness and sleep

The below code helps us find the Category of sleepers that have highest level in-activeness in bed.

```
temp2 <- ggplot(data = average_activity_sleep, aes(x=step_category,
                                                    fill=sleep_category)) +
          geom_bar(stat = "count")+
          labs(title="Average Steps compared to Average Sleep Time")+
          geom_text(stat = "count", aes(label = after_stat(count)),
          position = position_stack(vjust = .5))

grid.arrange(temp1,temp2, nrow = 2)
```

3.4.2. Correlation of all the features

We are plotting the Correlations to find which features has a strong impact on the other and which features has least impact. Please refer to results section for detailed explanation for the same.

```
# Finding the correlation between parameters
cor.table = cor(x=daily_activity_sleep[, c(4:7, 11:14)], method = "pearson")
corrplot(cor.table, method = "shade", shade.col = NA, tl.col = "black",
          tl.srt = 90, addCoef.col = "black", cl.pos = "n",,number.cex=0.75)
```

3.4.3. Correlation between Steps and calories burnt

We are plotting the boxplpt to detect outliers in categorized user with respect to steps taken and calories burnt. Also with respect to the bucket what is the average of calories burnt by the users in that bucket.

```
stat_box_data <- function(y, upper_limit = max(daily_activity_sleep$calories) * 1.15)
{
    return(
        data.frame(
            y = 0.95 * upper_limit,
            label = paste('count =', length(y), '\n',
                          'mean =', round(mean(y), 1), '\n')
        )
    )
}

# Box plot for steps taken and calories burnt facet_wrap/fill by step_categories
ggplot(data = daily_activity_sleep,
    aes(x=step_categories, y=calories, fill=step_categories)) +
  geom_boxplot()+
  stat_summary(fun.data = stat_box_data, geom = "text",  hjust = 0.5, vjust = 0.7)
```

3.4.4. Correlation between calories burnt and weight logged by Steps taken

We are plotting the relationship between weight in kg of a person with respect to that, how much does the person burns calories or how much steps does he/she takes. We are here looking for a pattern which could help in making a justification that says "Irrespective weight people are concerned about their fitness".

```
# Merging activity and sleep dataframe with weight dataframe
daily_activity_sleep_weight <- merge(daily_activity_sleep, weight_log,
                                        by=c("id","date"))

# Plotting the Line graph with pearson coefficient
ggplot(data=daily_activity_sleep_weight, aes(weight_kg, calories)) +
    geom_point(color = 'purple') +
    geom_smooth(method = 'lm', formula = y ~ x, se = TRUE, color = 'tomato') +
    stat_cor(data = daily_activity_sleep_weight,method = "pearson", label.x = 30, label.y=4600)+
    labs(title='Total weight vs. Total calories burnt w.r.t step categories',
          x = 'Total weight', y = 'Total calories burnt') +
    facet_wrap(~step_categories)
```

3.4.5. Correlation between number of minutes slept and calories burnt with respect to Distance covered

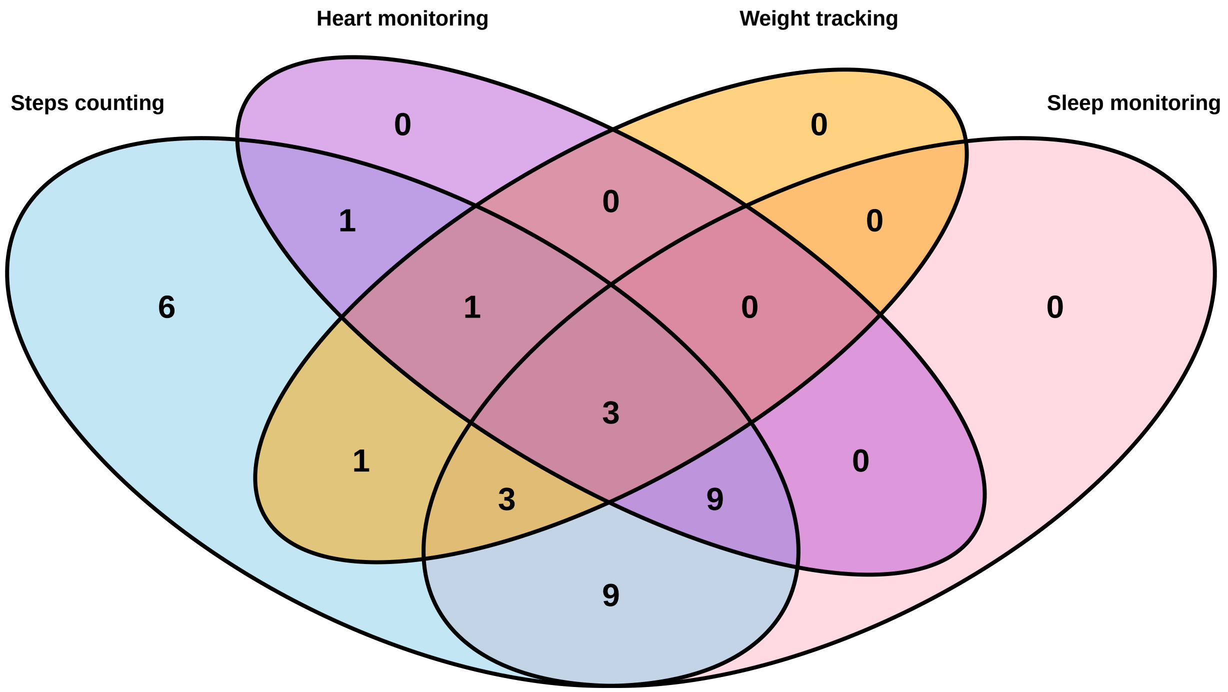
In the below plot, we are trying to see if there is any kind of relationship that says "More the distance covered, more the calories burnt and hence more the number of hours slept". In general and real life, people who do physical work more tend to sleep for 7 to 8 hours a day which is sufficient for their body to rest. So, we are looking for that pattern in our data.

```
# Plotting the Scatterplot with pearson coefficient
ggplot(data = daily_activity_sleep, aes(x=total_minutes_asleep, y=calories)) +
    geom_point(size = 1.5)+ geom_smooth(method = 'loess', formula= y ~ x)+
    stat_cor(data = daily_activity_sleep,method = "pearson", label.x = 10, label.y=5200)+
    labs(x="Number of hours slept" , y="Calories burned")+
    ggtitle("Correlation of Sleep and Calories burnt based on Distance travelled")+
    facet_wrap(c(~ distance_categories))
```

4. Results

4.1. How did the users interacted with the features of the product?

```
## NULL
```



4.1.1. Interpretation

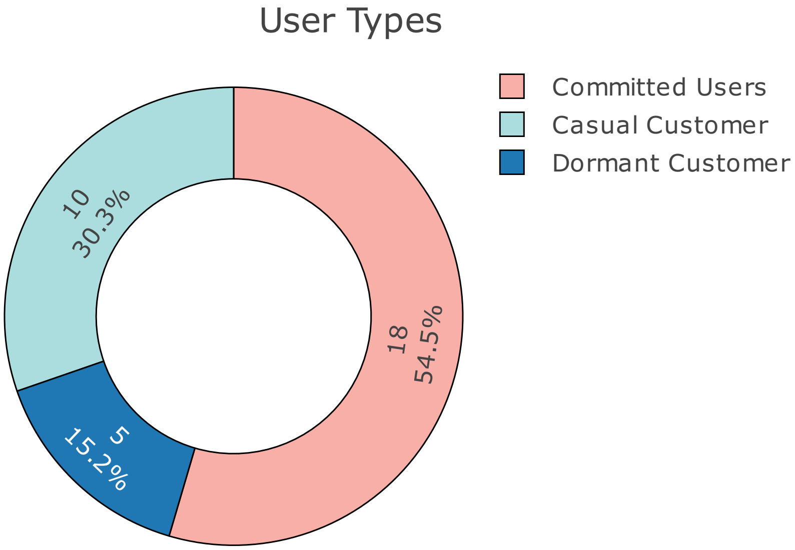
- 100% (33 Ids) have STEPS count records (combine with or without other features)
- 73% (24 Ids) have STEPS count and SLEEP tracking records
- 42% (14 Ids) have STEPS count and HEARTBEATS monitoring records
- 24% (8 Ids) have STEPS count and WEIGHT tracking records

Users who have used all 4 features - 9% (3 Ids) have all four featured records of STEPS - SLEEP - HEARTBEATS - WEIGHT

Users who have used 3 features - 9% (3 ids) used 3 features of STEPS - SLEEP - WEIGHT - 1 id used trio-feature STEPS - HEARTBEATS - WEIGHT

Based on the venn diagram, we could conclude Step is the core function that includes data of all users and Sleep monitoring is the 2nd most used feature. So, for rest of our analysis, we are going to merge Steps and Sleep dataframes based on date and user ID.

4.2. Classifying the type of user based on the usage rate



4.2.1. Interpretation

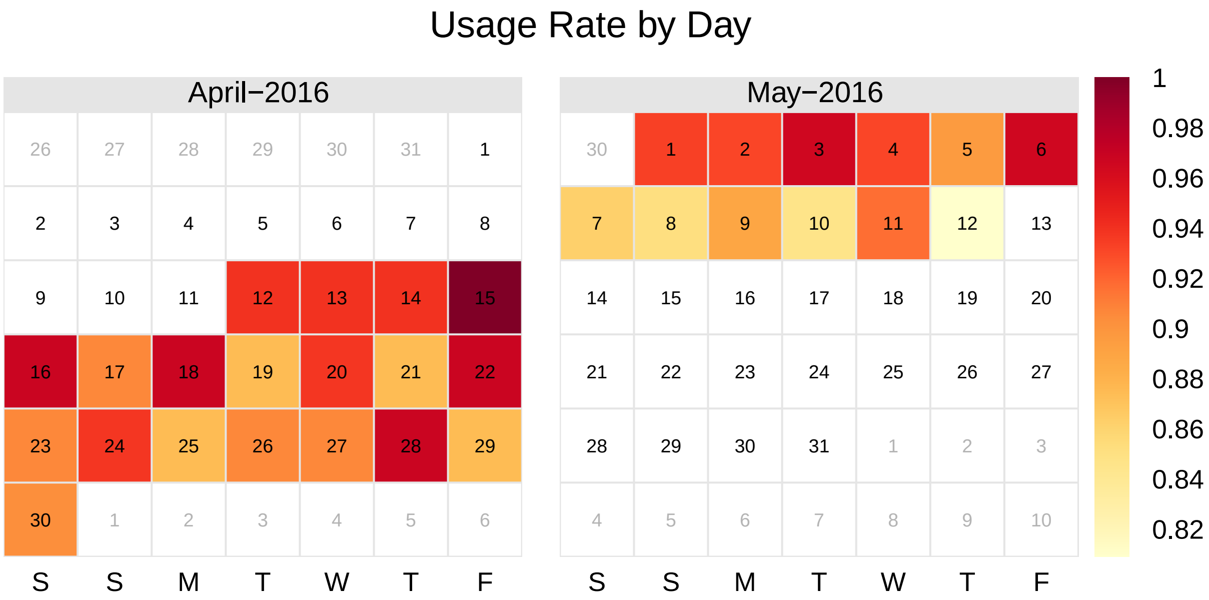
Looking at the chart above, we can say that we have approx **55% (i.e. 18 users) users who have used our product daily (31 days)** through out the experiment. We have **15% of users who can be considered as Dormant/Inactive because they used the product less than 14 days** in a period of 31 days.

NOTE: This data is very useful for marketing and research team, so that they can talk and figure out what challenges the user is facing while using the product and based on the feedback they can improve their product. Secondly, the users who have used their product daily, what changes they have noticed in their health, productivity, etc... should be gathered and used as a marketing tactics to create awareness about health and increase their sales.

4.3. How likely does the user wears the fitbit devices?

4.3.1. Day to Day basis

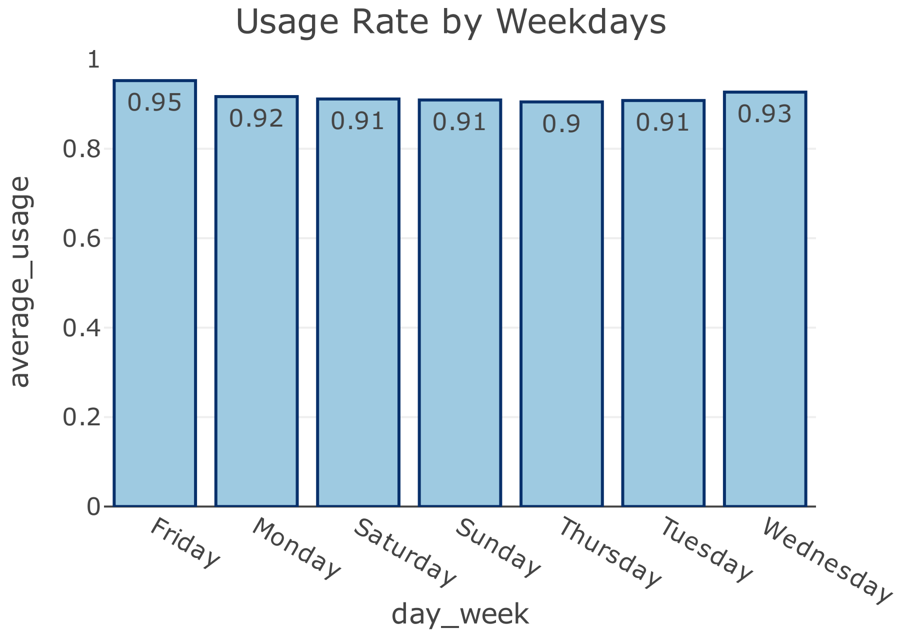
We are plotting a calendarPlot that helps us in identifying, on which dates the usage rate was the highest, if there is any kind of pattern that is followed or not.



Daily Interpretation

Based on the calendar heatmap illustration, **the usage rate varies from 80% to 100%**, and we can figure out that, **usage rate in the first 10 days of experiment was higher compared to usage rate in the last 10 days**. This could be considered as a negative sign for the marketing team to advertise their product because out of 33 users, 15 users (i.e. almost 45%) did not use their product daily, they were either casual user or dormant user.

4.3.2. Weekly basis



Weekly Interpretation

From the above barchart, it is clear that, Friday, Wednesday and Monday are the days with Higher Usage rate, with **friday being the highest amongst them**. Also we can say that weekly the usage rate ranges from 90% to 95%.

4.4 Bucketing the user based on different criteria to gain better insights

We have categorized users based on various categories, let's have a look at the dataframe after the user's are bucketed in different categories.

The new dataframe consists of below columns.

```
colnames(daily_activity_sleep)
```

```
##  [1] "id"                      "date"
##  [3] "day_week"                "total_steps"
##  [5] "calories"                "total_distance"
##  [7] "very_active_minutes"     "fairly_active_minutes"
##  [9] "lightly_active_minutes"  "sedentary_minutes"
## [11] "total_sleep_records"     "total_minutes_asleep"
## [13] "total_time_in_bed"       "total_mins_awake"
## [15] "step_categories"            "distance_categories"
## [17] "calories_categories"        "sleep_categories"
## [19] "sleep_inactive_categories"
```

Now we are creating a average bucket for each user over the time period of experiment.

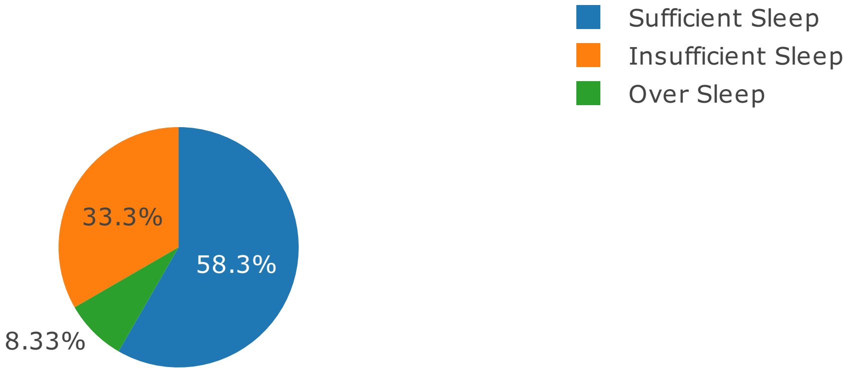
Once we have created the average bucketing, we get a dataframe with shape (24 X 13), 24 representing number of unique users in the observation and 13 representing the columns we are going to use for further analysis.

The bucketed dataframe consists of below columns.

```
colnames(average_activity_sleep)
```

```
##  [1] "id"                         "average_sleep"
##  [3] "average_step"               "average_calories"
##  [5] "average_distance"           "average_inactive_in_bed"
##  [7] "active_days"                "sleep_category"
##  [9] "step_category"              "distance_category"
## [11] "calorie_category"           "sleep_inactive_categories"
## [13] "user_type"
```

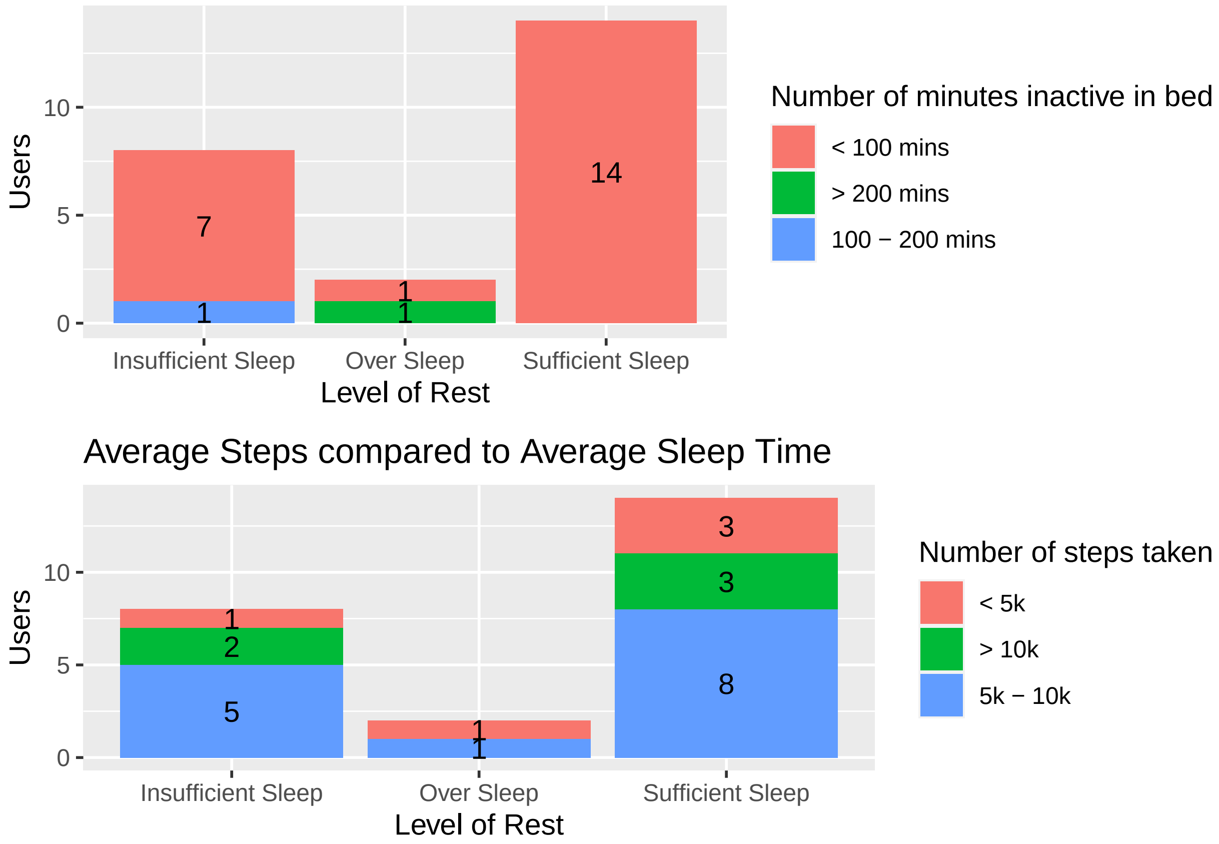
4.5. Are the users getting proper amount of sleep or are they just lying on the bed?



As observed from the pie chart we can see that **approx 58% of the users get proper sleep** and **approx 33% of users don't get proper sleep.**

So, let's look deep into what behavior hinders those 33% user from getting proper sleep

- Is the factor of fitness freakness responsible for that users who goes out for walking?
- Is the factor the sleep inactiveness responsible for that user who spend more time in bed doing nothing?

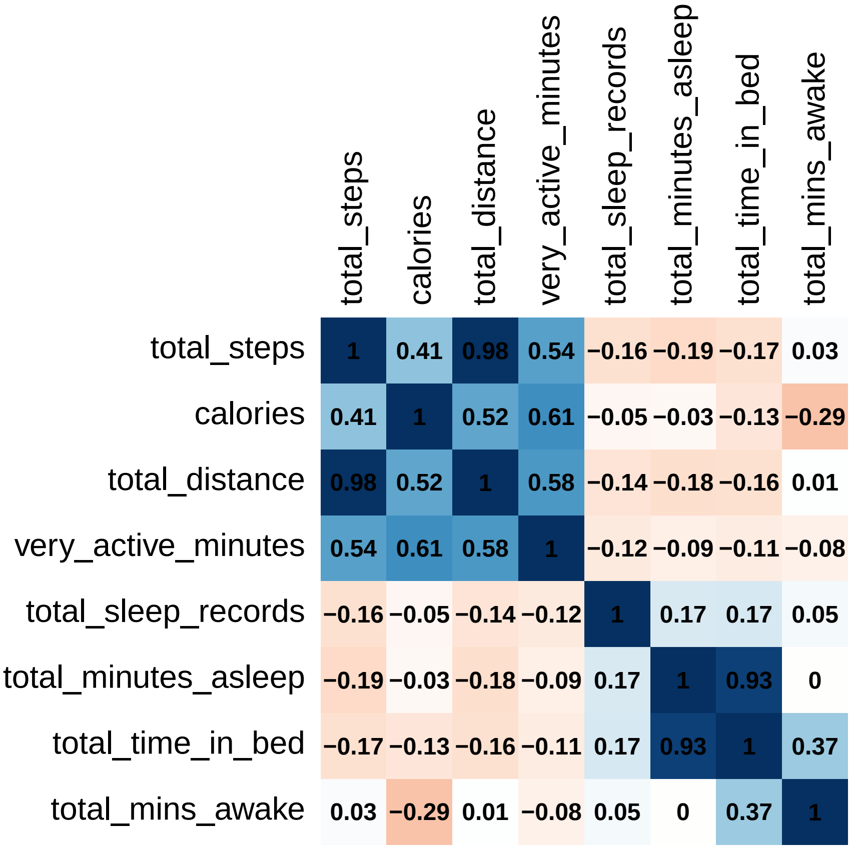


4.5.1. Finding the factor behind Insufficient Sleep

From this grid plot, we can see that there are 8 users who aren't getting sufficient sleep. Out of which 1 user is not fitness freak as he/she takes steps less than 5000, and when we compare the same with respect to time spent on bed doing nothing, there are 7 users who spend approx 2 hours extra in bed daily doing nothing and 1 user who spends approx 3 hours in bed doing nothing.

From this observation we can say that out of 33 users, 8 users who have Insufficient sleep tend to **lie in bed and doing nothing or maybe using phone**.

4.6. Feature Correlation Matrix



	total_steps	calories	total_distance	very_active_minutes	total_sleep_records	total_minutes_asleep	total_time_in_bed	total_mins_awake
total_steps	1	0.41	0.98	0.54	−0.16	−0.19	−0.17	0.03
calories	0.41	1	0.52	0.61	−0.05	−0.03	−0.13	−0.29
total_distance	0.98	0.52	1	0.58	−0.14	−0.18	−0.16	0.01
very_active_minutes	0.54	0.61	0.58	1	−0.12	−0.09	−0.11	−0.08
total_sleep_records	−0.16	−0.05	−0.14	−0.12	1	0.17	0.17	0.05
total_minutes_asleep	−0.19	−0.03	−0.18	−0.09	0.17	1	0.93	0
total_time_in_bed	−0.17	−0.13	−0.16	−0.11	0.17	0.93	1	0.37
total_mins_awake	0.03	−0.29	0.01	−0.08	0.05	0	0.37	1

4.6.1. Interpretation of Correlation Matrix

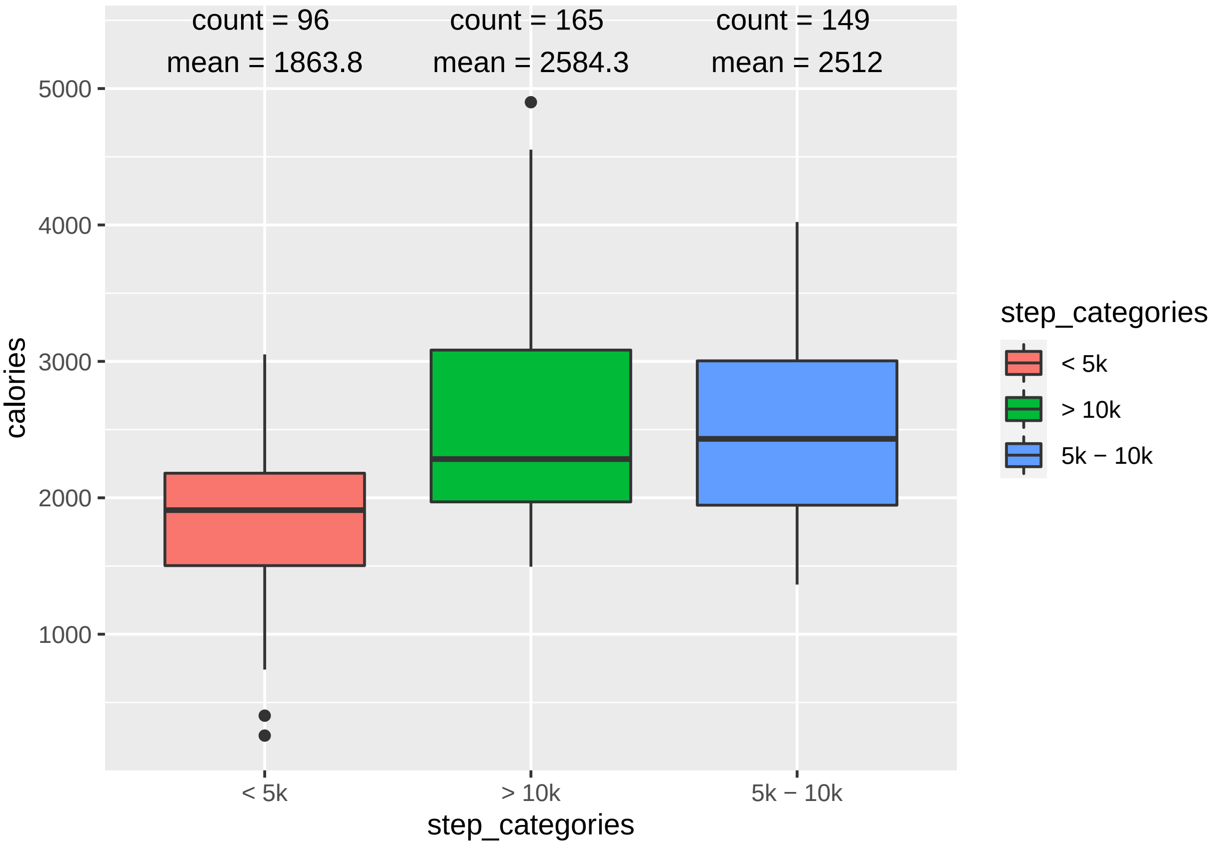
Looking at the correlation Matrix, the **blue color's shades represent positive relationship** and **pink color's shades represent negative relationship**. White color being neutral and yet that won't be reflecting any effective relationship.

- 0.98 being the highest correlation co-efficient between **Total Distance traveled** and **Total Steps taken**
- 0.93 being the 2nd highest correlation co-efficient between **Total Time in bed** and **Total Time asleep**
- A good relationship is defined between **Total Distance traveled(0.58)**, **Total Calories burnt (0.61)**, **Total Steps taken(0.54)** and **Very Active Minutes**

  NOTE:- The active minutes represent that users are doing some physical exercise, thus we get a strong relationship with all physical factors
- Correlation co-efficient for **Calories** and **Total Distance traveled** is **0.52**, whereas for **Calories** and **Total Steps** taken is **0.41**.

4.7. Relationship among different parameters and buckets

4.7.1. Relationship between Steps and calories burnt

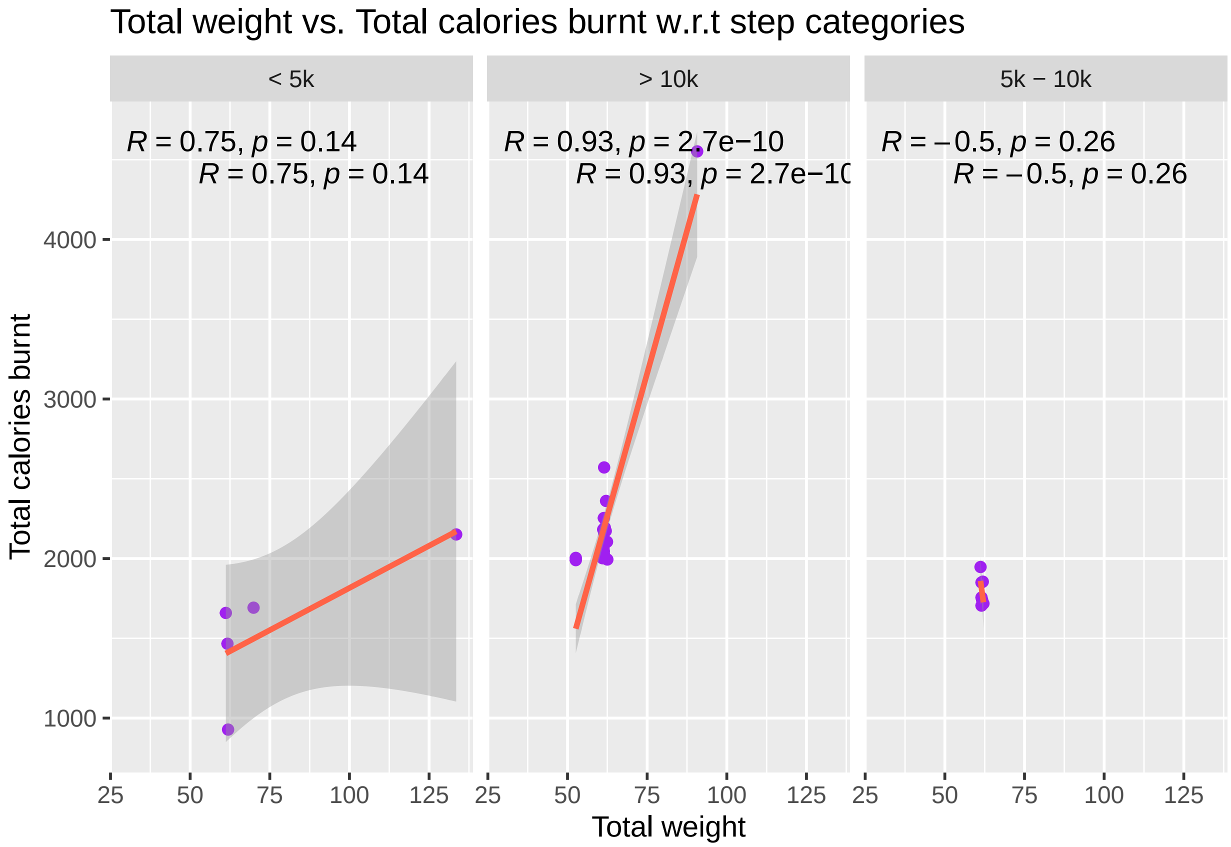


Interpretation of Boxplot (Steps Vs Calories)

The boxplot shows the **direct correlation between the number of steps taken and the calories burnt**, where greater the steps, more the calories burnt.

- The average calories burnt by a person taking less than 5k steps is around 1850 calories a day. The average calories burnt by a person taking between 5k and 1k steps is around 2500 calories a day. The average calories burnt by a person taking more than 10k steps is around 2600 calories a day.
- **We have around 96 observation in Steps less than 5k, 165 observations in steps greater than 10k and about 149 observations in steps between 5k to 10k steps.**
- There is an overlap between calories burnt by people walking between 5000 - 10000 steps and people walking more than 10000 steps a day. One possible reason for this is the intensity of walking, which has not been considered for this analysis.

4.7.2. Relationship between calories burnt and weight logged by Steps taken



Interpretation of Line graph (Calories Vs Weight given Steps taken)

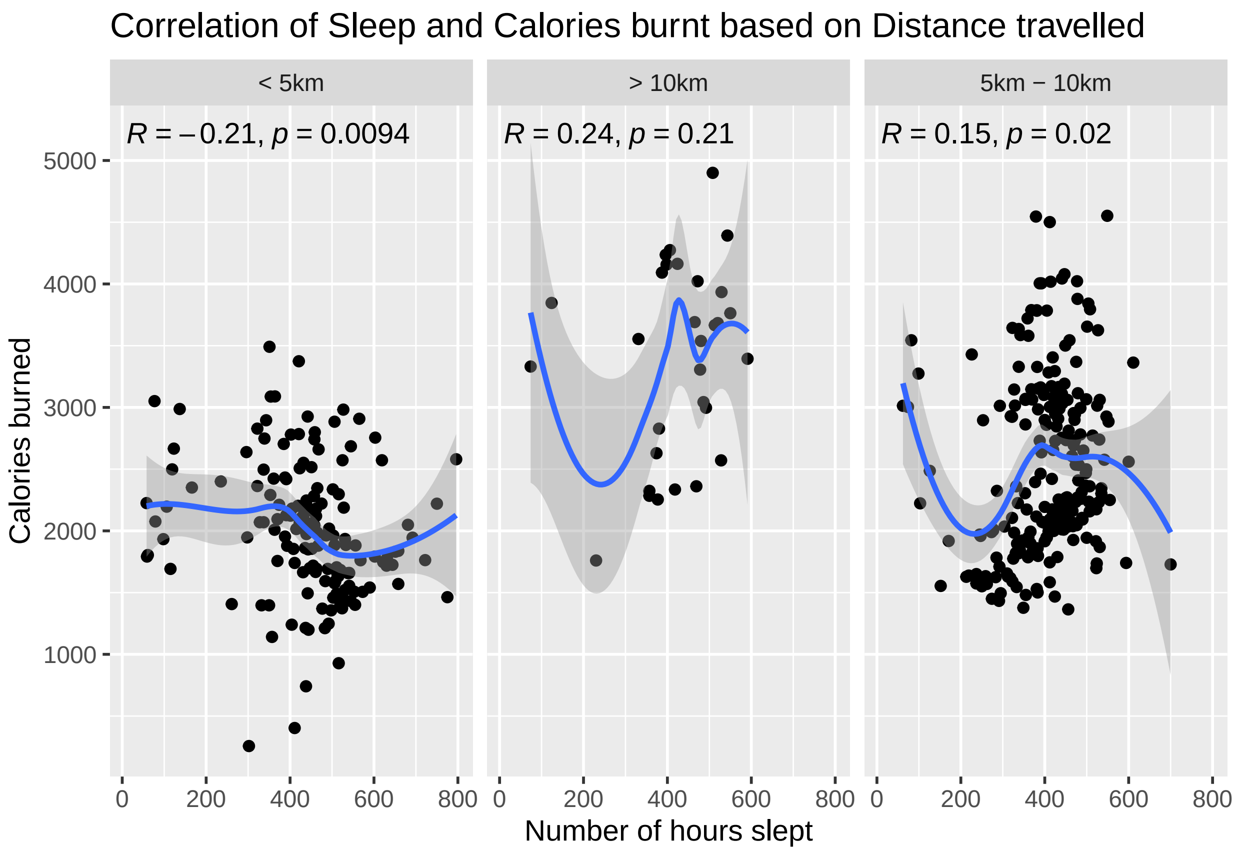
The line graph shows **positive and direct relationship between calories burnt and weight given the number of steps taken are either greater than 10k or less than 5k**.

- Looking at the chart for steps less than 5k, we observe that there is user with weight greater than 120KGs (approx ~ 135Kgs) takes less than 5k steps and burns 2200 calories a day.
- Looking at the chart for steps greater than 10k, we observe that all the users in this category are having weight less than 60Kgs and they burn around 2000 to 2600 calories in a day. Also, we can see a user with weight approx 82Kgs burns calories more than 4500 calories a day.
- For the chart with steps in between 5k to 10k, we can see all the users are clustered around 60Kgs weight and with less than 2000 calories a day.

Key Finding

- Out of all 3 categories, we get the strong association and positive relationship between this two variables when total steps taken is greater than 10k because we get **R(co-efficient) as 0.93** and **p-value as 2.7e-10(i.e. it is less than 0.05)**. Thus we can conclude that, **Calories burnt are dependent on Weight of the person if he/she takes approx 10k steps a day**.

4.7.3. Relationship between Hours slept and Calories burnt by Distance traveled



Interpretation of Scatterplot (Sleep Vs Calories given the Distance traveled)

The Scatterplot shows **no/weak relationship between calories burnt and number of hours slept**, this is justified because of the curvy lines in the graph and not the linear graph.

- In each category of distance, we see multiple entries of users for each day but the value of r is less than 0.7 this is why we say that the relationship here is either weak or could be considered as no relationship.
- In the category of 5km - 10 kms the r value is small representing weak relationship but the value of p is 0.02 (i.e. is less than 0.05), it signifies a good correlation and this creates a confusion. Why is it so can be explained as below..

In our case, the effect size (the pearson's r) is the strength of a relationship between two variables, which is relatively small, but the significant p value refers to the fact that, given our sample size, the error measurement associated with these variables is small enough so we are getting the correlation as reliable.

5. Conclusion

To summarize, let's go through the results one by one to extract the piece of information that we were looking through out the case study.

- The First question that we came across is **how are the users interacting with the features**, The VennDiagram present in section 4.1 illustrates that **Step Monitoring** feature is widely used by all users, whereas **Weight Monitoring** feature is least used because of various reasons, out of which one could be the manual logging of weight to the device.

- The Second question that we came across is **trends analysis or pattern finding in the usage of the device by the user**. The calendarPlot and the barchart in section 4.3 represents the usage rate on daily basis and on weekly basis, we have found that users tend to use product in the first half of the experiment and later on the usage rate went on decreasing. Also, Friday turn out to be having the Highest usage rate of 95%.

- Answer to our Third question is the relationship amongst the variables which can be found in the Correlation Matrix present in section 4.6, through this we found out that **Steps taken and Distance traveled** represents a strongest relationship and dependency on each other with **pearson coefficient being around 0.98**.

When it comes to find the list of features that influence health, we have bucketed users into different categories and analyzed each bucket to get the influential factor for health.

A.) Sleep and factors responsible for Insufficient sleep

The section 4.5 represents that We have approx **33% of Insufficient sleepers** out of which **approx 90% of the users have Insufficient amount of sleep because of their behavior where they spend average of 2.5 hours on bed lying around doing nothing or using phone**. This observation can be found in section . Quality of sleep/Inactiveness on bed does affects our health and mind.

B.) Steps taken and Calories burnt

The average number of steps to be taken for a healthy adult is 10k steps a day. Also, as we have observed a direct relationship between steps taken and calories burnt in the section 4.7. More the number of steps taken, more is the distance traveled and more amount of calories burnt. **Higher the calories burnt, lesser the chances of getting health issues related to heart and also we stay fit.**

C.) Weight and Calories Burnt

People with more weight should definitely burn more calories by taking more steps a day. Higher calories are burnt when people with higher weight take more than 10k steps a day. This is illustrated using the Co-efficient value (r=0.93) and p-value(p=2.77e-10) in section 4.7

- Last question for us to conclude this finding is **How does this analysis be helpful for fitbit to market their product**. We can say that each section has it's own significance and has it's own term for understanding. **This case study will help the team evaluate their customers using various buckets that we created like usage rate, sleep quality, steps taken, calories burnt. The team can find out the "Dormant Customers" and get their feedback to improve their features/product.**

- Using the feedback of the "Loyal Customer", the marketing team can advertise their product, take help from NGOs and run campaign to create awareness of the product and it's benefit on health. **This will not only help them target their monthly/yearly sales but also make citizens aware about the health with such cool and handy health monitoring devices.**

6. References

- About Fitness trackers and wearables
- Background Knowledge of Fitbit Device
- Crowd Sourced Fitbit dataset

- Analyzing Fitbit data: Forensic approach
- How anyone can use their own fitbit data and analyze it