FitBit Case Study

Nandini Singh

2022-07-31

ABOUT THE PROJECT:

Bellabeat is a high-tech manufacturer of health-focused products for women.Urška Sršen, cofounder and Chief Creative Officer of Bellabeat, believes that analyzing smart device fitness data could help unlock new growth opportunities for the company.

Analysis tasks:

In this case study we will identify potential opportunities for growth and recommendations for the improvement of devices based on trends in their usage. The data set used for this analysis is : FitBit Fitness Tracker Data (CC0: Public Domain, dataset made available through Mobius).

Loading required packages:

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✔ ggplot2 3.3.6     ✔ purrr   0.3.4
## ✔ tibble  3.1.7     ✔ dplyr   1.0.9
## ✔ tidyr   1.2.0     ✔ stringr 1.4.0
## ✔ readr   2.1.2     ✔ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

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

```
library(dplyr)
library(ggplot2)
library(tidyr)
```

Importing dataset:

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

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

```
daily_calories <- read.csv("dailyCalories_merged.csv")
daily_steps <- read.csv("dailySteps_merged.csv")
hourly_calories <- read.csv("hourlyCalories_merged.csv")
daily_sleep <- read_csv("sleepDay_merged.csv")
```

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

Now that we have imported our required data files, let's summarize them and take a look at their specifications.

```
head(daily_activity)
```

Id	ActivityDate	TotalSteps	TotalDistance	TrackerDistance	LoggedActivitiesDistance
<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1503960366	4/12/2016	13162	8.50	8.50	0
1503960366	4/13/2016	10735	6.97	6.97	0
1503960366	4/14/2016	10460	6.74	6.74	0
1503960366	4/15/2016	9762	6.28	6.28	0
1503960366	4/16/2016	12669	8.16	8.16	0
1503960366	4/17/2016	9705	6.48	6.48	0

6 rows | 1-6 of 15 columns

```
head(daily_calories)
```

	Id	ActivityDay	Calories
	<dbl>	<chr>	<int>
1	1503960366	4/12/2016	1985
2	1503960366	4/13/2016	1797

	Id	ActivityDay	Calories
	<dbl>	<chr>	<int>
3	1503960366	4/14/2016	1776
4	1503960366	4/15/2016	1745
5	1503960366	4/16/2016	1863
6	1503960366	4/17/2016	1728

6 rows

```
head(daily_steps)
```

	Id	ActivityDay	StepTotal
	<dbl>	<chr>	<int>
1	1503960366	4/12/2016	13162
2	1503960366	4/13/2016	10735
3	1503960366	4/14/2016	10460
4	1503960366	4/15/2016	9762
5	1503960366	4/16/2016	12669
6	1503960366	4/17/2016	9705

6 rows

```
head(hourly_calories)
```

	Id	ActivityHour	Calories
	<dbl>	<chr>	<int>
1	1503960366	4/12/2016 12:00:00 AM	81
2	1503960366	4/12/2016 1:00:00 AM	61
3	1503960366	4/12/2016 2:00:00 AM	59
4	1503960366	4/12/2016 3:00:00 AM	47
5	1503960366	4/12/2016 4:00:00 AM	48
6	1503960366	4/12/2016 5:00:00 AM	48

6 rows

```
head(daily_sleep)
```

Id	SleepDay	TotalSleepRecords	TotalMinutesAsleep	TotalTimeInBed
<dbl>	<chr>	<dbl>	<dbl>	<dbl>
1503960366	4/12/2016 12:00:00 AM	1	327	346
1503960366	4/13/2016 12:00:00 AM	2	384	407

Id	SleepDay	TotalSleepRecords	TotalMinutesAsleep	TotalTimeInBed
<dbl>	<chr>	<dbl>	<dbl>	<dbl>
1503960366	4/15/2016 12:00:00 AM	1	412	442
1503960366	4/16/2016 12:00:00 AM	2	340	367
1503960366	4/17/2016 12:00:00 AM	1	700	712
1503960366	4/19/2016 12:00:00 AM	1	304	320

6 rows

```
str(daily_activity)
```

```
## spec_tbl_df [940 × 15] (S3: spec_tbl_df/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 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 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 ...
##  - attr(*, "spec")=
##   .. 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()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(daily_calories)
```

```
## 'data.frame':    940 obs. of  3 variables:
##  $ Id         : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDay: chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ Calories   : int  1985 1797 1776 1745 1863 1728 1921 2035 1786 1775 ...
```

```
str(daily_steps)
```

```
## 'data.frame':    940 obs. of  3 variables:
##  $ Id         : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ ActivityDay: chr  "4/12/2016" "4/13/2016" "4/14/2016" "4/15/2016" ...
##  $ StepTotal  : int  13162 10735 10460 9762 12669 9705 13019 15506 10544 9819 ...
```

```
str(hourly_calories)
```

```
## 'data.frame':    22099 obs. of  3 variables:
##  $ Id          : num  1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ 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" ...
##  $ Calories    : int  81 61 59 47 48 48 48 47 68 141 ...
```

```
str(daily_sleep)
```

```
## spec_tbl_df [413 × 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id                : num [1:413] 1.5e+09 1.5e+09 1.5e+09 1.5e+09 1.5e+09 ...
##  $ SleepDay          : chr [1:413] "4/12/2016 12:00:00 AM" "4/13/2016 12:00:00 AM" "4/15/2016 1
2:00:00 AM" "4/16/2016 12:00:00 AM" ...
##  $ TotalSleepRecords : num [1:413] 1 2 1 2 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep: num [1:413] 327 384 412 340 700 304 360 325 361 430 ...
##  $ TotalTimeInBed    : num [1:413] 346 407 442 367 712 320 377 364 384 449 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   SleepDay = col_character(),
##   ..   TotalSleepRecords = col_double(),
##   ..   TotalMinutesAsleep = col_double(),
##   ..   TotalTimeInBed = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Once we have looked at the structure of the data, it's time to clean it and look for any inconsistencies or errors in it.

Firstly, let's find if their id any duplicate data entry:

```
sum(duplicated(daily_activity))
```

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

```
sum(duplicated(daily_calories))
```

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

```
sum(duplicated(daily_steps))
```

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

```
sum(duplicated(hourly_calories))
```

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

```
sum(duplicated(daily_sleep))
```

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

The above result shows that we have 3 duplicate entries in the daily_sleep data set so we need to remove them.

```
daily_sleep <- daily_sleep %>%
  distinct() %>%
  drop_na()
```

Let's verify that the duplicates are removed.

```
sum(duplicated(daily_sleep))
```

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

The above result shows that we have no duplicate entries so now we can move to cleaning and renaming the column headers. For this, we need to load packages skimr and janitor.

```
install.packages("skimr",repos = "http://cran.us.r-project.org")
```

```
## package 'skimr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\NANDINI\AppData\Local\Temp\Rtmp0sryeL\downloaded_packages
```

```
library(skimr)
install.packages("janitor",repos = "http://cran.us.r-project.org")
```

```
## package 'janitor' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\NANDINI\AppData\Local\Temp\Rtmp0sryeL\downloaded_packages
```

```
library(janitor)
```

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

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

Now we use the following to set a standard for column header names so that it is easier for us to merge the datasets later in the analysis.

```
clean_names(daily_activity)
```

id	activity_date	total_steps	total_distance	tracker_distance	logged_activities_dista
<dbl>	<chr>	<dbl>	<dbl>	<dbl>	<
1503960366	4/12/2016	13162	8.50	8.50	0.000
1503960366	4/13/2016	10735	6.97	6.97	0.000
1503960366	4/14/2016	10460	6.74	6.74	0.000
1503960366	4/15/2016	9762	6.28	6.28	0.000
1503960366	4/16/2016	12669	8.16	8.16	0.000
1503960366	4/17/2016	9705	6.48	6.48	0.000
1503960366	4/18/2016	13019	8.59	8.59	0.000
1503960366	4/19/2016	15506	9.88	9.88	0.000
1503960366	4/20/2016	10544	6.68	6.68	0.000
1503960366	4/21/2016	9819	6.34	6.34	0.000

1-10 of 940 rows | 1-6 of 15 columns

Previous **1** 2 3 4 5 6 … 94 Next

```
daily_activity <- rename_with(daily_activity, tolower)
clean_names(daily_calories)
```

id	activity_day	calories
<dbl>	<chr>	<int>
1503960366	4/12/2016	1985
1503960366	4/13/2016	1797
1503960366	4/14/2016	1776
1503960366	4/15/2016	1745
1503960366	4/16/2016	1863
1503960366	4/17/2016	1728
1503960366	4/18/2016	1921
1503960366	4/19/2016	2035

id	activity_day	calories
<dbl>	<chr>	<int>
1503960366	4/20/2016	1786
1503960366	4/21/2016	1775

1-10 of 940 rows Previous **1** 2 3 4 5 6 … 94 Next

```
daily_calories <- rename_with(daily_calories, tolower)
clean_names(daily_steps)
```

id	activity_day	step_total
<dbl>	<chr>	<int>
1503960366	4/12/2016	13162
1503960366	4/13/2016	10735
1503960366	4/14/2016	10460
1503960366	4/15/2016	9762
1503960366	4/16/2016	12669
1503960366	4/17/2016	9705
1503960366	4/18/2016	13019
1503960366	4/19/2016	15506
1503960366	4/20/2016	10544
1503960366	4/21/2016	9819

1-10 of 940 rows Previous **1** 2 3 4 5 6 … 94 Next

```
daily_steps <- rename_with(daily_steps, tolower)
clean_names(hourly_calories)
```

id	activity_hour	calories
<dbl>	<chr>	<int>
1503960366	4/12/2016 12:00:00 AM	81
1503960366	4/12/2016 1:00:00 AM	61
1503960366	4/12/2016 2:00:00 AM	59
1503960366	4/12/2016 3:00:00 AM	47
1503960366	4/12/2016 4:00:00 AM	48
1503960366	4/12/2016 5:00:00 AM	48
1503960366	4/12/2016 6:00:00 AM	48
1503960366	4/12/2016 7:00:00 AM	47
1503960366	4/12/2016 8:00:00 AM	68

id	activity_hour	calories
<dbl>	<chr>	<int>
1503960366	4/12/2016 9:00:00 AM	141

1-10 of 10,000 rows Previous **1** 2 3 4 5 6 … 1000 Next

```
hourly_calories <- rename_with(hourly_calories, tolower)
clean_names(daily_sleep)
```

id	sleep_day	total_sleep_records	total_minutes_asleep	total_time_in_bed
<dbl>	<chr>	<dbl>	<dbl>	<dbl>
1503960366	4/12/2016 12:00:00 AM	1	327	346
1503960366	4/13/2016 12:00:00 AM	2	384	407
1503960366	4/15/2016 12:00:00 AM	1	412	442
1503960366	4/16/2016 12:00:00 AM	2	340	367
1503960366	4/17/2016 12:00:00 AM	1	700	712
1503960366	4/19/2016 12:00:00 AM	1	304	320
1503960366	4/20/2016 12:00:00 AM	1	360	377
1503960366	4/21/2016 12:00:00 AM	1	325	364
1503960366	4/23/2016 12:00:00 AM	1	361	384
1503960366	4/24/2016 12:00:00 AM	1	430	449

1-10 of 410 rows Previous **1** 2 3 4 5 6 … 41 Next

```
daily_sleep <- rename_with(daily_sleep, tolower)
```

In this analysis we'll be looking at various attributes in a single data so we need to make sure the dates are consistent.

```
daily_activity <- daily_activity %>%
  rename(date = activitydate) %>%
  mutate(date = as_date(date, format = "%m/%d/%Y"))
daily_calories <- daily_calories %>%
  rename(date = activityday) %>%
  mutate(date = as_date(date, format = "%m/%d/%Y"))
daily_steps <- daily_steps %>%
  rename(date = activityday) %>%
  mutate(date = as_date(date, format = "%m/%d/%Y"))
daily_sleep <- daily_sleep %>%
  rename(date = sleepday) %>%
  mutate(date = as_date(date,format ="%m/%d/%Y %I:%M:%S %p"))
```

Let's take a look at the cleaned datasets.

```
head(daily_activity)
```

id <dbl>	date <date>	totalsteps <dbl>	totaldistance <dbl>	trackerdistance <dbl>	loggedactivitiesdistance <dbl>
1503960366	2016-04-12	13162	8.50	8.50	0
1503960366	2016-04-13	10735	6.97	6.97	0
1503960366	2016-04-14	10460	6.74	6.74	0
1503960366	2016-04-15	9762	6.28	6.28	0
1503960366	2016-04-16	12669	8.16	8.16	0
1503960366	2016-04-17	9705	6.48	6.48	0

6 rows | 1-6 of 15 columns

```
head(daily_calories)
```

	id <dbl>	date <date>	calories <int>
1	1503960366	2016-04-12	1985
2	1503960366	2016-04-13	1797
3	1503960366	2016-04-14	1776
4	1503960366	2016-04-15	1745
5	1503960366	2016-04-16	1863
6	1503960366	2016-04-17	1728

6 rows

```
head(daily_steps)
```

	id <dbl>	date <date>	steptotal <int>
1	1503960366	2016-04-12	13162
2	1503960366	2016-04-13	10735
3	1503960366	2016-04-14	10460
4	1503960366	2016-04-15	9762
5	1503960366	2016-04-16	12669
6	1503960366	2016-04-17	9705

6 rows

```
head(daily_sleep)
```

id <dbl>	date <date>	totalsleeprecords <dbl>	totalminutesasleep <dbl>	totaltimeinbed <dbl>

id	date	totalsleeprecords	totalminutesasleep	totaltimeinbed
<dbl>	<date>	<dbl>	<dbl>	<dbl>
1503960366	2016-04-12	1	327	346
1503960366	2016-04-13	2	384	407
1503960366	2016-04-15	1	412	442
1503960366	2016-04-16	2	340	367
1503960366	2016-04-17	1	700	712
1503960366	2016-04-19	1	304	320

6 rows

Now,for the analysis we merge the datasets: daily_activity, daily_calories and daily_steps.

```
daily_activity_calories <- merge(daily_activity, daily_calories, by=c("id", "date"))
glimpse(daily_activity_calories)
```

```
## Rows: 940
## Columns: 16
## $ id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036…
## $ date                     <date> 2016-04-12, 2016-04-13, 2016-04-14, 2016-04-…
## $ 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…
## $ 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, 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, 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.x               <dbl> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203…
## $ calories.y               <int> 1985, 1797, 1776, 1745, 1863, 1728, 1921, 203…
```

```
daily_activity_sleep <- merge(daily_activity, daily_sleep, by=c ("id", "date"))
glimpse(daily_activity_sleep)
```

```
## Rows: 410
## Columns: 18
## $ id                       <dbl> 1503960366, 1503960366, 1503960366, 150396036…
## $ date                     <date> 2016-04-12, 2016-04-13, 2016-04-15, 2016-04-…
## $ totalsteps               <dbl> 13162, 10735, 9762, 12669, 9705, 15506, 10544…
## $ totaldistance            <dbl> 8.50, 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6.3…
## $ trackerdistance          <dbl> 8.50, 6.97, 6.28, 8.16, 6.48, 9.88, 6.68, 6.3…
## $ loggedactivitiesdistance <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ veryactivedistance       <dbl> 1.88, 1.57, 2.14, 2.71, 3.19, 3.53, 1.96, 1.3…
## $ moderatelyactivedistance <dbl> 0.55, 0.69, 1.26, 0.41, 0.78, 1.32, 0.48, 0.3…
## $ lightactivedistance      <dbl> 6.06, 4.71, 2.83, 5.04, 2.51, 5.03, 4.24, 4.6…
## $ sedentaryactivedistance  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …
## $ veryactiveminutes        <dbl> 25, 21, 29, 36, 38, 50, 28, 19, 41, 39, 73, 3…
## $ fairlyactiveminutes      <dbl> 13, 19, 34, 10, 20, 31, 12, 8, 21, 5, 14, 23,…
## $ lightlyactiveminutes     <dbl> 328, 217, 209, 221, 164, 264, 205, 211, 262, …
## $ sedentaryminutes         <dbl> 728, 776, 726, 773, 539, 775, 818, 838, 732, …
## $ calories                 <dbl> 1985, 1797, 1745, 1863, 1728, 2035, 1786, 177…
## $ totalsleeprecords        <dbl> 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
## $ totalminutesasleep       <dbl> 327, 384, 412, 340, 700, 304, 360, 325, 361, …
## $ totaltimeinbed           <dbl> 346, 407, 442, 367, 712, 320, 377, 364, 384, …
```

For our daily_steps dataset we will convert the date string to date time.

```
daily_steps<- daily_steps %>%
  rename(date_time = date) %>%
  mutate(date_time = as.POSIXct(date_time,format ="%m/%d/%Y %I:%M:%S %p" , tz=Sys.timezone()))

head(daily_steps)
```

	id <dbl>	date_time <dttm>	steptotal <int>
1	1503960366	2016-04-12 05:30:00	13162
2	1503960366	2016-04-13 05:30:00	10735
3	1503960366	2016-04-14 05:30:00	10460
4	1503960366	2016-04-15 05:30:00	9762
5	1503960366	2016-04-16 05:30:00	12669
6	1503960366	2016-04-17 05:30:00	9705

6 rows

Now that we have merged the data sets, it;s time to analyse.

For the first task we'll be looking at the steps walked and minutes slept over the course of a week i.e. for each weekday. In the next block we are calculating the weekdays based on the dates present in the data set columns.

```
weekday_steps_sleep <- daily_activity_sleep %>%
  mutate(weekday = weekdays(date))

weekday_steps_sleep$weekday <-ordered(weekday_steps_sleep$weekday, levels=c("Monday", "Tuesday",
"Wednesday", "Thursday",
"Friday", "Saturday", "Sunday"))
```

We calculate the average steps walked and minutes slept every weekday.

```
weekday_steps_sleep <-weekday_steps_sleep%>%
  group_by(weekday) %>%
  summarize (daily_steps = mean(totalsteps), daily_sleep = mean(totalminutesasleep))

head(weekday_steps_sleep)
```

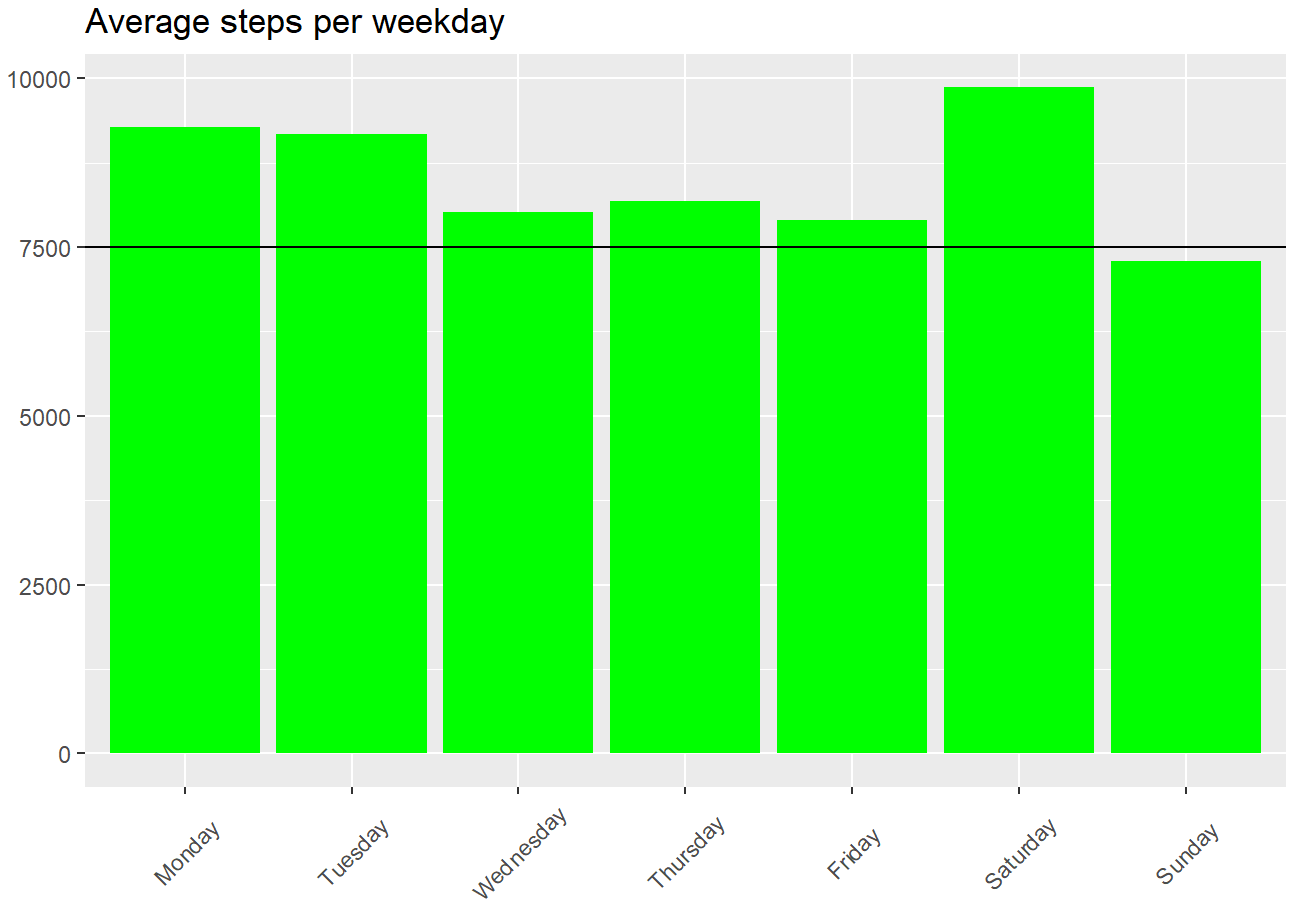
weekday <ord>	daily_steps <dbl>	daily_sleep <dbl>
Monday	9273.217	419.5000
Tuesday	9182.692	404.5385
Wednesday	8022.864	434.6818
Thursday	8183.516	401.2969
Friday	7901.404	405.4211
Saturday	9871.123	419.0702

6 rows

An average active person should take about 7500 steps per day and the recommended minutes of sleep is 480 minutes i.e. 8 hours per day, therefore we use these two as the y-intercepts. So let's plot the steps taken and minutes slept.

1)Average number to steps per weekday:

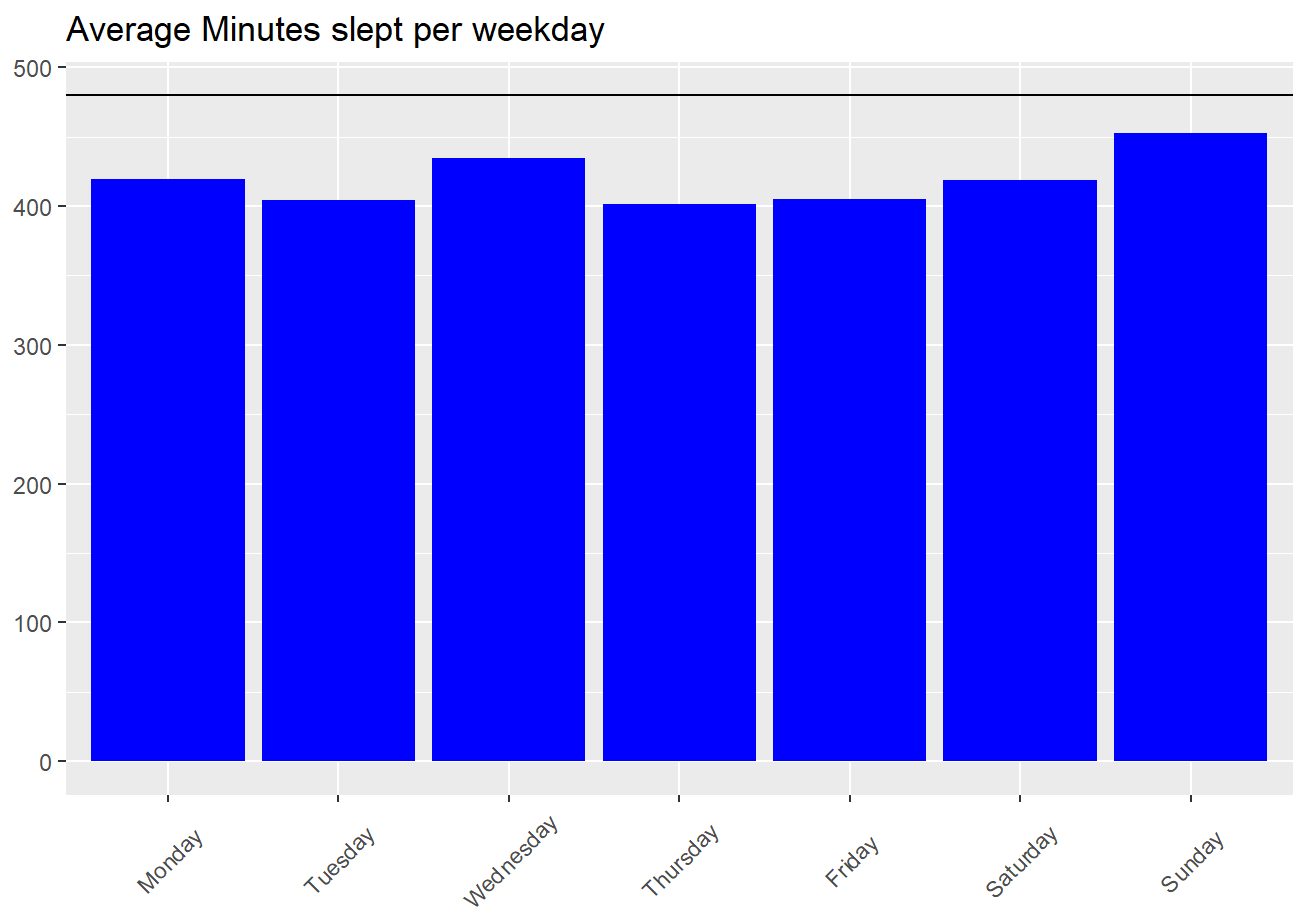
```
ggplot(weekday_steps_sleep) +
      geom_col(aes(weekday, daily_steps), fill = "green") +
      geom_hline(yintercept = 7500) +
      labs(title = "Average steps per weekday", x= "", y = "") +
      theme(axis.text.x = element_text(angle = 45,vjust = 0.5,hjust = 0.5))
```



2)Average

number of minutes slept per weekday:

```
ggplot(weekday_steps_sleep, aes(weekday, daily_sleep)) +
      geom_col(fill = "blue") +
      geom_hline(yintercept = 480) +
      labs(title = "Average Minutes slept per weekday", x= "", y = "") +
      theme(axis.text.x = element_text(angle = 45,vjust = 0.5, hjust = 0.5))
```

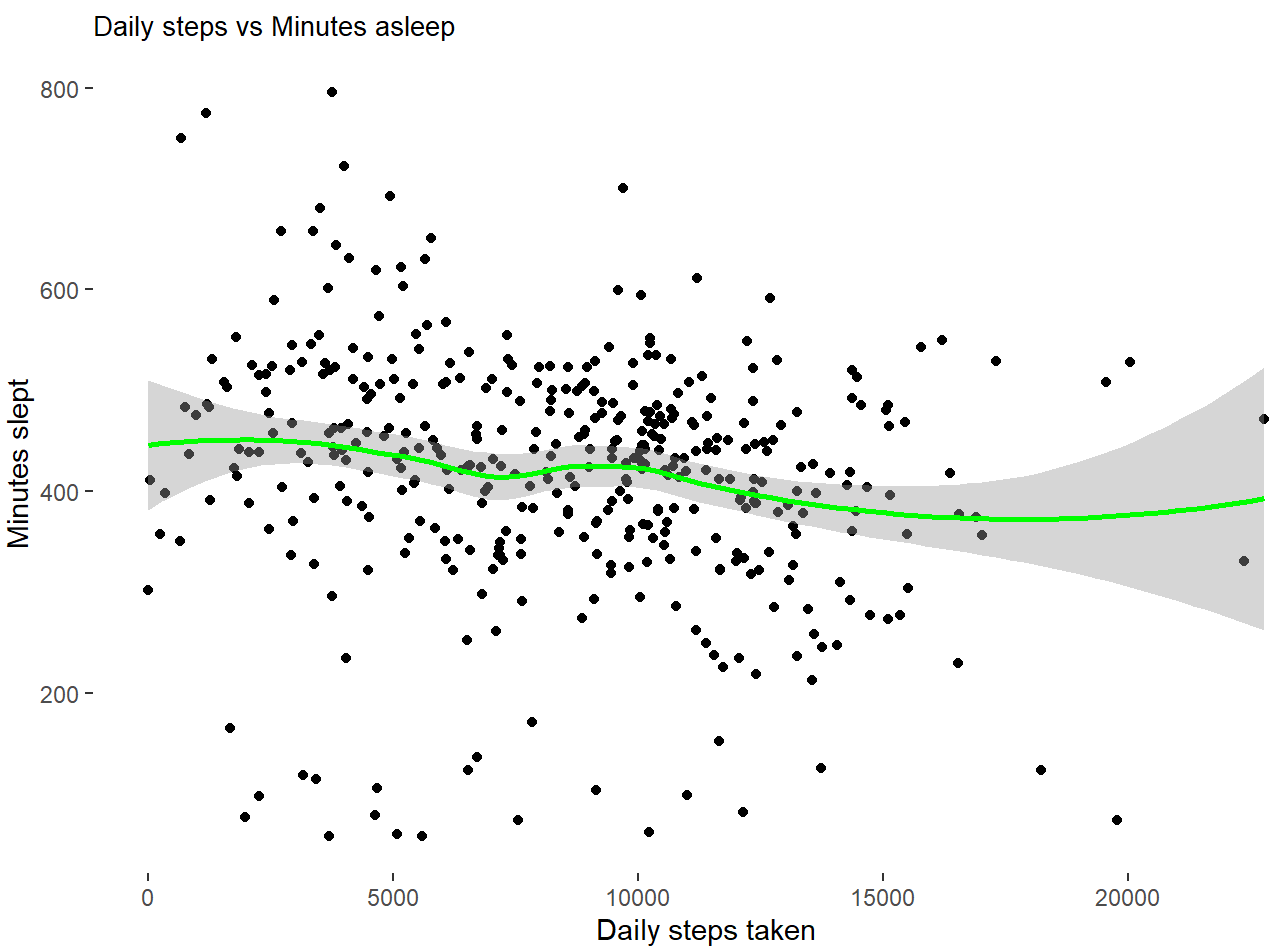


Looking at the above to graphs we conclude that: -People walk more than 7500 steps everyday of the week except for sunday and also that the average number of steps walked is the highest on saturday(close to 10,000 steps). -People are sleeping for less than 480 minutes everyday that is less than the recommended sleep time, also that people are sleeping the most on sunday.

Let's find out if there is any correlation between the number of minutes slept and the total number of steps taken.

```
ggplot(daily_activity_sleep, aes(x=totalsteps, y=totalminutesasleep))+
  geom_jitter() +
  geom_smooth(color = "green") +
  labs(title = "Daily steps vs Minutes asleep", x = "Daily steps taken", y= "Minutes slept") +
   theme(panel.background = element_blank(),
        plot.title = element_text( size=10))
```

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



As per the plot, there is no correlation between the two i.e. the number of steps taken is not affected by the number of minutes slept and vice-versa.

Now for the second task let's find out : Hourly calories throughout the day:

```
hourly_calories <- hourly_calories %>%
  separate(activityhour, c("date", "time"), sep=" ")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 22099 rows [1, 2, 3,
## 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

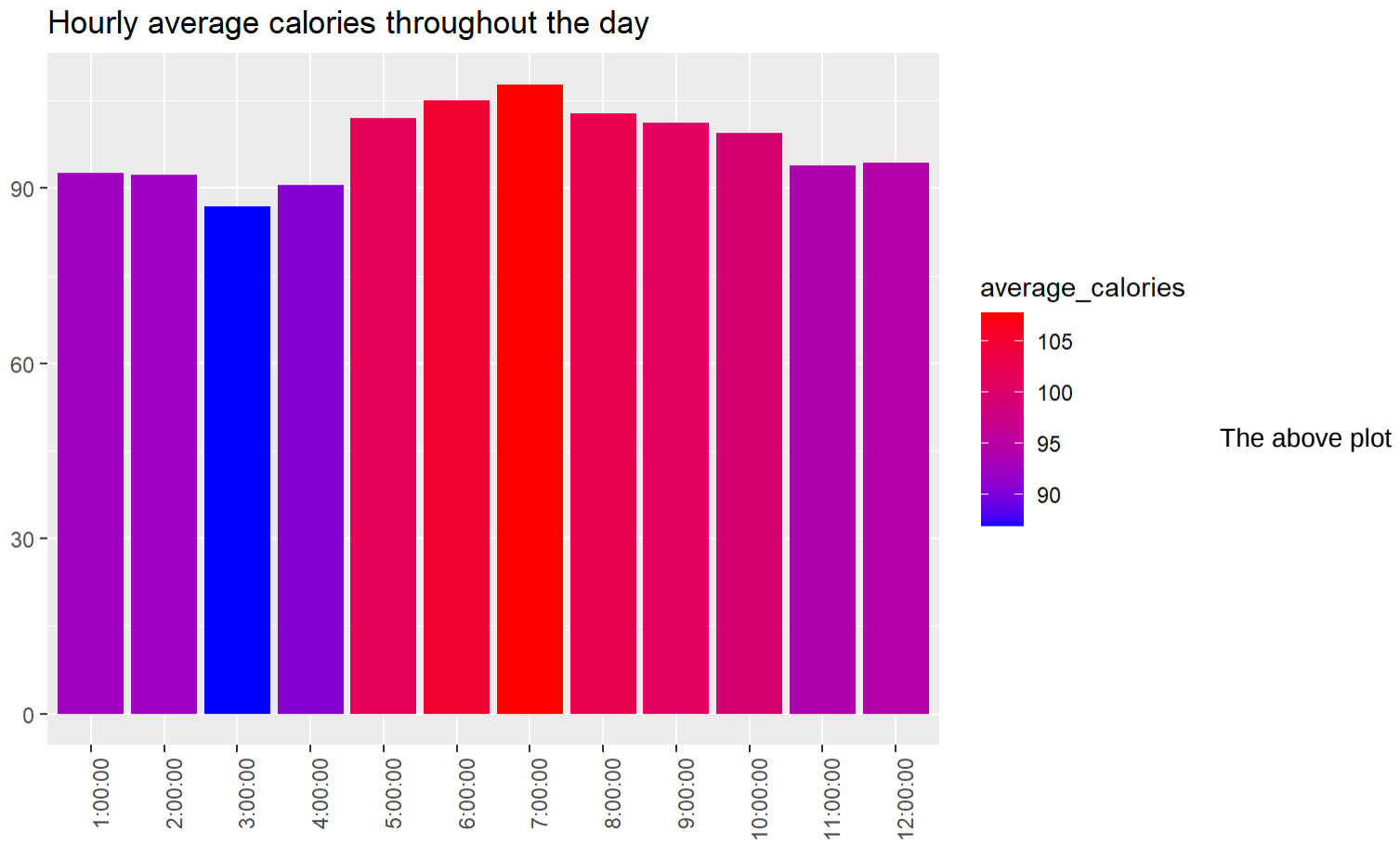
```
head(hourly_calories)
```

	id	date	time	calories
	<dbl>	<chr>	<chr>	<int>
1	1503960366	4/12/2016	12:00:00	81
2	1503960366	4/12/2016	1:00:00	61
3	1503960366	4/12/2016	2:00:00	59
4	1503960366	4/12/2016	3:00:00	47
5	1503960366	4/12/2016	4:00:00	48
6	1503960366	4/12/2016	5:00:00	48

6 rows

Now we plot the graph to see the daily trend in calorie intake :

```
hourly_calories %>%
  group_by(time) %>%
  summarize(average_calories = mean(calories)) %>%
  ggplot() +
  geom_col(mapping = aes(x=time, y = average_calories, fill =average_calories)) +
  labs(title = "Hourly average calories throughout the day", x="", y="") +
  scale_fill_gradient(low = "blue", high = "red")+
  theme(axis.text.x = element_text(angle = 90, hjust="1"))+
   scale_x_discrete(limits = c("1:00:00","2:00:00","3:00:00","4:00:00","5:00:00","6:00:00","7:00:0
0","8:00:00","9:00:00","10:00:00","11:00:00","12:00:00"))
```

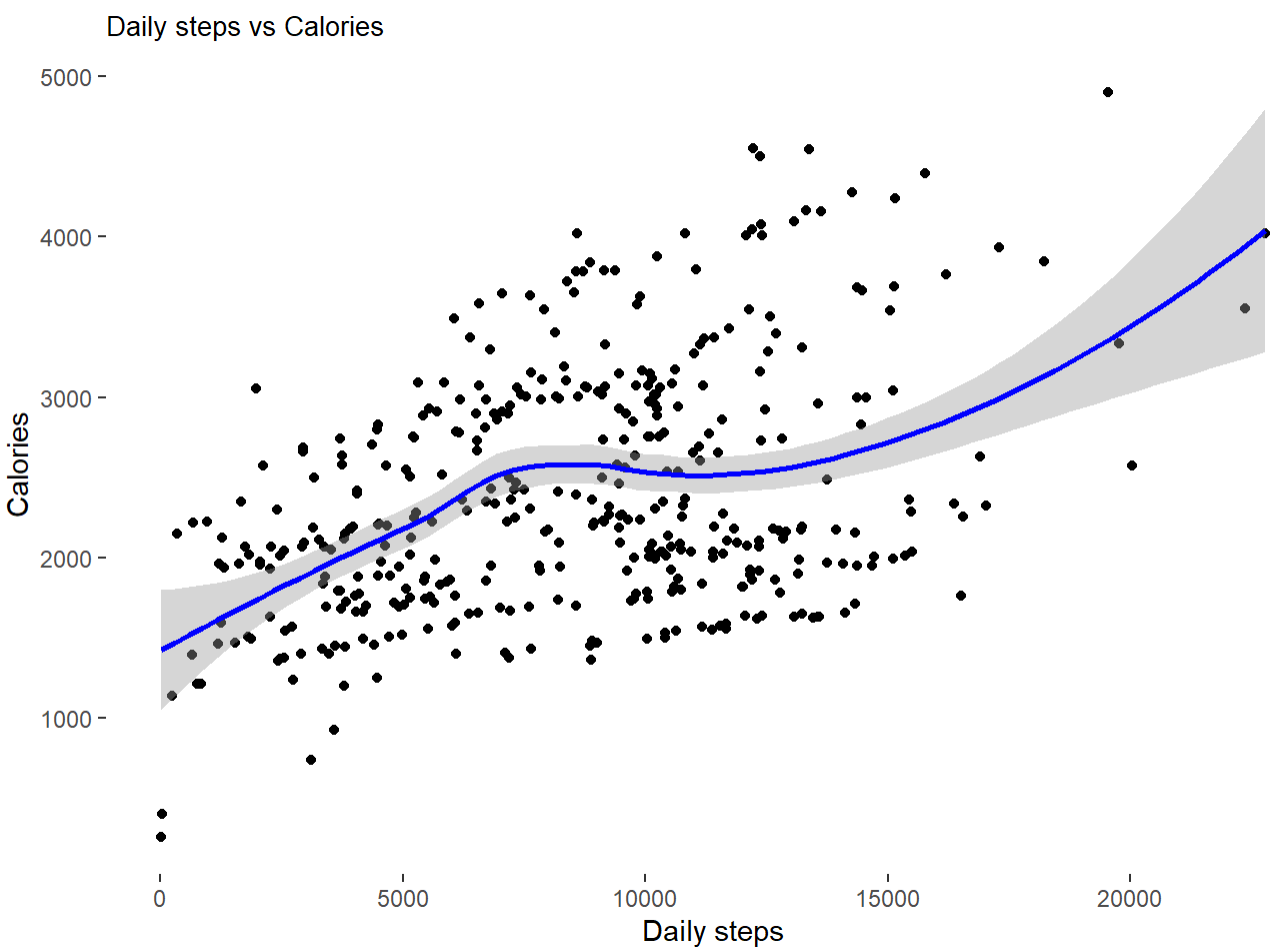


The above plot tells that most of the users are consuming the most calories between 6:00:00 and 8:00:00(peak at 7:00:00).It also shows that users are consuming the least calories around 3:00:00.

Let's figure out if there is any correlation between the number of steps taken and the calories consumed throughout the day.

```
ggplot(daily_activity_sleep, aes(x=totalsteps, y=calories))+
  geom_jitter() +
  geom_smooth(color = "blue") +
  labs(title = "Daily steps vs Calories", x = "Daily steps", y= "Calories") +
   theme(panel.background = element_blank(),
        plot.title = element_text( size=10))
```

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



From the graph above, we conclude that there is a positive correlation between daily steps taken and the calories burned, which makes it very clear that more calorie consumption leads to more steps taken in order to burn them out.

We have seen some trends in activity, calories and steps so let's find out about the use of our devices. For this part of the analysis, we'll categories users as: Low Users:using devices between 1 to 7 days. Moderate Users:using devices between 8 to 20 days. High Users: using devices between 21 to 31 days.

```
device_use <- daily_activity_sleep %>%
  group_by(id) %>%
  summarize(days_used=sum(n())) %>%
  mutate(usage = case_when(
    days_used >= 1 & days_used <= 7 ~ "low users",
    days_used >= 8 & days_used <= 20 ~ "moderate users",
    days_used >= 21 & days_used <= 31 ~ "high users",
  ))

head(device_use)
```

id	days_used	usage
<dbl>	<int>	<chr>
1503960366	25	high users
1644430081	4	low users
1844505072	3	low users
1927972279	5	low users
2026352035	28	high users
2320127002	1	low users

6 rows

converting the above results to percentage data:

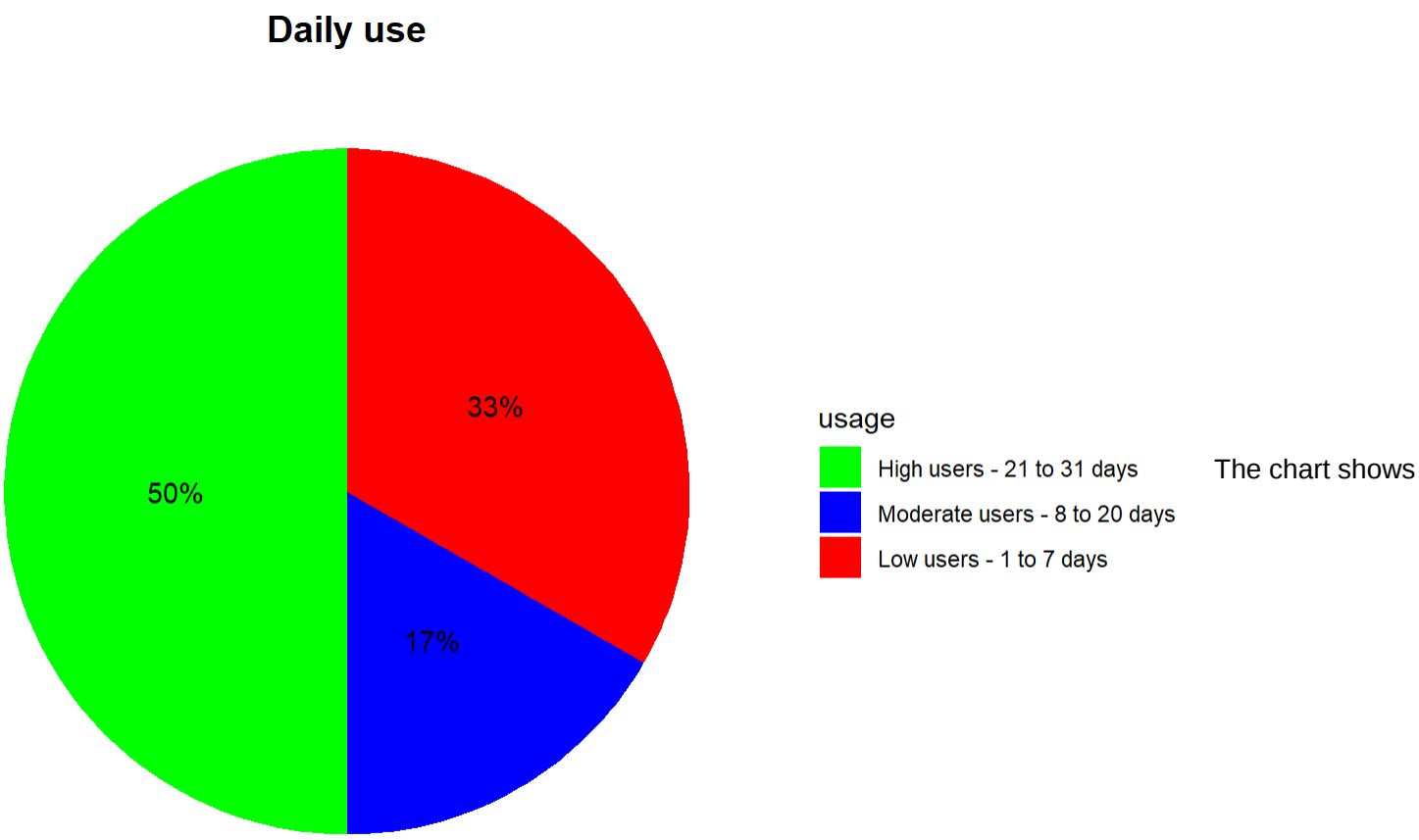
```
percent_use <- device_use %>%
  group_by(usage) %>%
  summarise(total = n()) %>%
  mutate(totals = sum(total)) %>%
  group_by(usage) %>%
  summarise(total_percent = total / totals) %>%
  mutate(labels = scales::percent(total_percent))

percent_use$usage <- factor(percent_use$usage, levels = c("high users", "moderate users", "low use
rs"))

head(percent_use)
```

usage <fct>	total_percent <dbl>	labels <chr>
high users	0.5000000	50%
low users	0.3333333	33%
moderate users	0.1666667	17%

3 rows

We make a piechart to represent this data:

```
percent_use %>%
  ggplot(aes(x="",y=total_percent, fill=usage)) +
  geom_bar(stat = "identity", width = 1)+
  coord_polar("y", start=0)+
  theme_minimal()+
  theme(axis.title.x= element_blank(),
        axis.title.y = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        axis.ticks = element_blank(),
        axis.text.x = element_blank(),
        plot.title = element_text(hjust = 0.5, size=14, face = "bold")) +
  geom_text(aes(label = labels),
            position = position_stack(vjust = 0.5))+
  scale_fill_manual(values = c("#00FF00","#0000FF","#FF0000"),
                    labels = c("High users - 21 to 31 days",
                               "Moderate users - 8 to 20 days",
                               "Low users - 1 to 7 days"))+
  labs(title="Daily use")
```



that the company still has 33% users who are using the devices for less than a weeks time. It also shows that 50% users are using the devices for more than 3 weeks.

CONCLUSION

In this study we looked at trends in activity, daily steps taken, daily number of minutes slept plus daily and hourly calories. We found out:

1)The number of steps taken by a person is more than recommended i.e. 7500 steps daily every weekday except for sunday, this sudden change in number of steps can affect the number of calories burned by the person. We should add reminders to the devices to remind the user that they have to take at least 7500 steps daily, in the end of the week if the person completes 7500 every weekday we can give them an appreciation message for keeping up the good work.

2)One of the concerning findings of this study was that people are not sleeping the required amount of hours i.e. 8 hours, we need to do additions to the devices to remind the users of the importance of good nights sleep and how it can affect our work and mood throughout the day. We can add some type of game in the system which advances levels only when a person slept 8-9 hours, no less no more.

3)Most of the users are consuming the most calories between 6:00:00 and 8:00:00(peak at 7:00:00).It also shows that users are consuming the least calories around 3:00:00. We add a chart to the display showing the user the amount of calorie intake and calories burned, it can be customized so each user can keep a track of their exercises and eating habits. In this section we also found out that there is a positive correlation between the number of steps taken and the calories burned, we should encourage our customers to exercise more on days they consume more calories to take care of their digestion.

4)The company still has 33% users who are using the devices for less than a weeks time. It also shows that 50% users are using the devices for more than 3 weeks and 17% of users use the device for a moderate period of 8 to 20 days. In order to retain our customers we can give them perks of using a device for longer periods we can unlocks some more features of the device and they would be revoked if the person goes back to the previous using habits. We can make the products more fashionable and fancy so that they add a touch of elegance, so people can wear them with any outfits because we know the obsession of people to look put together everywhere they go.