Case Study: Fitbit analysis

Mason Phung

I. Load library & datasets

1. Libraries

```
library(tidyverse) # main function, manipulate data
library(lubridate) # date format
library(gridExtra) # grid.arrange() to print many plots together in a same page
library(ggrepel) # For displaying plot's labels outside of the chart
library(wesanderson) # Wes Anderson color palette for plots
library(scales) # For percent()
```

2. Load datasets

Required information - Daily data of activity and sleep
- Hourly data: steps, intensities, calories
- Weight information

```
raw_daily_activity <- read.csv("datasets/dailyActivity_merged.csv")
raw_daily_sleep <- read.csv("datasets/sleepDay_merged.csv")

raw_hourly_steps <- read.csv("datasets/hourlySteps_merged.csv")
raw_hourly_intensities <- read.csv("datasets/hourlyIntensities_merged.csv")
raw_hourly_calories <- read.csv("datasets/hourlyCalories_merged.csv")

raw_weight_log <- read.csv("datasets/weightLogInfo_merged.csv")
```

II. Data cleaning

The data is not yet cleaned, by observing the data, we will determine the place that we need to clean

1. Merge data

Hourly group has 3 datasets, based on observation, we can see that all of them have the matched information and data collected time, therefore, we will merge all of these 3 hourly datasets into 1 single to make the analysis process more convenient.

```r
# Left join hourly data together by Id and ActivityHour
hourly_activity <- raw_hourly_steps %>%
  left_join(raw_hourly_calories, by = c("Id","ActivityHour")) %>%
  left_join(raw_hourly_intensities, by = c("Id","ActivityHour"))

# Convert the date format of ActivityHour in to mdy_hms
hourly_activity <- hourly_activity %>%
  mutate(ActivityHour = mdy_hms(ActivityHour))

# Seperate ActivityHour into Date and Hour
hourly_activity <- hourly_activity %>%
  separate(
    ActivityHour, into = c("ActivityDate", "Hour"), sep= " "
  )
```

2. Date format cleaning

- In the ***Daily activity*** dataset, we will change the date in to the Month-Day-Year format.
- In the ***Hourly activity***, we will convert all of time format into Month-Day-Year, Hour-Minute-Second.
- In ***Daily sleep***, because the time was collected with both date and hour, we will divide them into 2 different variables for easier analysis.

```
# Convert date format
raw_daily_activity$ActivityDate <- mdy(raw_daily_activity$ActivityDate)

# Separate data and hour
daily_sleep <- raw_daily_sleep %>%
    separate(SleepDay, into = c("ActivityDate", "Hour"), sep= " ") %>%
    mutate(ActivityDate = mdy(ActivityDate)) %>%
    select(-Hour)

# Refine date & time format
hourly_activity <- hourly_activity %>%
  mutate(
    Hour = ifelse(is.na(Hour), "00:00:00",Hour),
    ActivityDate = ymd(ActivityDate)
  )
```

Let's take a brief look at the data after merging and date formatting

```
head(raw_daily_activity)
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366   2016-03-25      11004          7.11            7.11
## 2 1503960366   2016-03-26      17609         11.55           11.55
## 3 1503960366   2016-03-27      12736          8.53            8.53
## 4 1503960366   2016-03-28      13231          8.93            8.93
## 5 1503960366   2016-03-29      12041          7.85            7.85
## 6 1503960366   2016-03-30      10970          7.16            7.16
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               2.57                     0.46
## 2                        0               6.92                     0.73
## 3                        0               4.66                     0.16
## 4                        0               3.19                     0.79
## 5                        0               2.16                     1.09
## 6                        0               2.36                     0.51
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                4.07                       0                33
## 2                3.91                       0                89
## 3                3.71                       0                56
## 4                4.95                       0                39
## 5                4.61                       0                28
## 6                4.29                       0                30
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories
## 1                  12                  205              804     1819
## 2                  17                  274              588     2154
## 3                   5                  268              605     1944
## 4                  20                  224             1080     1932
## 5                  28                  243              763     1886
## 6                  13                  223             1174     1820
```

```r
head(daily_sleep)
```

```
##           Id ActivityDate TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
## 1 1503960366   2016-04-12                 1                327            346
## 2 1503960366   2016-04-13                 2                384            407
## 3 1503960366   2016-04-15                 1                412            442
## 4 1503960366   2016-04-16                 2                340            367
## 5 1503960366   2016-04-17                 1                700            712
## 6 1503960366   2016-04-19                 1                304            320
```

```r
head(hourly_activity)
```

```
##           Id ActivityDate     Hour StepTotal Calories TotalIntensity
## 1 1503960366   2016-04-12 00:00:00       373       81             20
## 2 1503960366   2016-04-12 01:00:00       160       61              8
## 3 1503960366   2016-04-12 02:00:00       151       59              7
## 4 1503960366   2016-04-12 03:00:00         0       47              0
## 5 1503960366   2016-04-12 04:00:00         0       48              0
## 6 1503960366   2016-04-12 05:00:00         0       48              0
##   AverageIntensity
## 1         0.333333
## 2         0.133333
## 3         0.116667
## 4         0.000000
## 5         0.000000
## 6         0.000000
```

3. Check for NAs and duplicates

Take a look at the duplicates

```
# Look for duplicates in raw_daily_activity
sum(duplicated(raw_daily_activity))
```

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

```
# Look for duplicates in daily_sleep
sum(duplicated(daily_sleep))
```

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

```
# Look for duplicates in hourly_activity
sum(duplicated(hourly_activity))
```

```
## [1] 1225
```

Take a look at the duplicates in `daily_sleep` and `hourly_activity` dataset

```
daily_sleep[duplicated(daily_sleep),]
```

```
##             Id ActivityDate TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
## 162 4388161847   2016-05-05                 1                471            495
## 224 4702921684   2016-05-07                 1                520            543
## 381 8378563200   2016-04-25                 1                388            402
```

```
head(hourly_activity[duplicated(hourly_activity),],10)
```

```
##             Id ActivityDate     Hour StepTotal Calories TotalIntensity
## 719 1624580081   2016-04-12 00:00:00        31       55              4
## 720 1624580081   2016-04-12 00:00:00        31       55              4
## 721 1624580081   2016-04-12 00:00:00        31       55              4
## 723 1624580081   2016-04-12 01:00:00         0       51              1
## 724 1624580081   2016-04-12 01:00:00         0       51              1
## 725 1624580081   2016-04-12 01:00:00         0       51              1
## 727 1624580081   2016-04-12 02:00:00         0       50              0
## 728 1624580081   2016-04-12 02:00:00         0       50              0
## 729 1624580081   2016-04-12 02:00:00         0       50              0
## 731 1624580081   2016-04-12 03:00:00         7       51              1
##     AverageIntensity
## 719         0.066667
## 720         0.066667
## 721         0.066667
## 723         0.016667
## 724         0.016667
## 725         0.016667
## 727         0.000000
## 728         0.000000
## 729         0.000000
## 731         0.016667
```

Observation shows that `daily_sleep` duplicates are not actually duplications (each observation shows differences), therefore, we only remove the detected duplication in the `hourly_activity` dataset.

Remove duplications

```
# Choose only distinct observation
hourly_activity <- distinct(hourly_activity)
```

Take a look at the NA

```
any(is.na(raw_daily_activity))
```

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

```
any(is.na(daily_sleep))
```

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

```
any(is.na(hourly_activity))
```

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

There was no NA value left in 3 datasets.

4. Add weekdays into the datasets

In later analysis, we will compared the collected data in each weekday, therefore, adding their names into the datasets is required.

```
daily_activity <- raw_daily_activity %>%
  mutate(weekday = weekdays(ActivityDate)) %>%
  mutate(
    weekday = factor(weekday,
    levels = c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday')
    )
  )

daily_sleep <- daily_sleep %>%
  mutate(weekday = weekdays(ActivityDate)) %>%
  mutate(
    weekday = factor(weekday,
    levels = c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday')
    )
  )

hourly_activity <- hourly_activity %>%
  mutate(weekday = weekdays(ActivityDate)) %>%
  mutate(
    weekday = factor(weekday,
    levels = c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday')
    )
  )
```

```
##           Id ActivityDate TotalSteps TotalDistance TrackerDistance
## 1 1503960366   2016-03-25      11004          7.11            7.11
## 2 1503960366   2016-03-26      17609         11.55           11.55
## 3 1503960366   2016-03-27      12736          8.53            8.53
## 4 1503960366   2016-03-28      13231          8.93            8.93
## 5 1503960366   2016-03-29      12041          7.85            7.85
## 6 1503960366   2016-03-30      10970          7.16            7.16
##   LoggedActivitiesDistance VeryActiveDistance ModeratelyActiveDistance
## 1                        0               2.57                     0.46
## 2                        0               6.92                     0.73
## 3                        0               4.66                     0.16
## 4                        0               3.19                     0.79
## 5                        0               2.16                     1.09
## 6                        0               2.36                     0.51
##   LightActiveDistance SedentaryActiveDistance VeryActiveMinutes
## 1                4.07                       0                33
## 2                3.91                       0                89
## 3                3.71                       0                56
## 4                4.95                       0                39
## 5                4.61                       0                28
## 6                4.29                       0                30
##   FairlyActiveMinutes LightlyActiveMinutes SedentaryMinutes Calories   weekday
## 1                  12                  205              804     1819    Friday
## 2                  17                  274              588     2154  Saturday
## 3                   5                  268              605     1944    Sunday
## 4                  20                  224             1080     1932    Monday
## 5                  28                  243              763     1886   Tuesday
## 6                  13                  223             1174     1820 Wednesday
```

```
##          Id ActivityDate TotalSleepRecords TotalMinutesAsleep TotalTimeInBed
## 1 1503960366   2016-04-12                 1                327            346
## 2 1503960366   2016-04-13                 2                384            407
## 3 1503960366   2016-04-15                 1                412            442
## 4 1503960366   2016-04-16                 2                340            367
## 5 1503960366   2016-04-17                 1                700            712
## 6 1503960366   2016-04-19                 1                304            320
##     weekday
## 1   Tuesday
## 2 Wednesday
## 3    Friday
## 4  Saturday
## 5    Sunday
## 6   Tuesday
```

```
##          Id ActivityDate     Hour StepTotal Calories TotalIntensity
## 1 1503960366   2016-04-12 00:00:00       373       81             20
## 2 1503960366   2016-04-12 01:00:00       160       61              8
## 3 1503960366   2016-04-12 02:00:00       151       59              7
## 4 1503960366   2016-04-12 03:00:00         0       47              0
## 5 1503960366   2016-04-12 04:00:00         0       48              0
## 6 1503960366   2016-04-12 05:00:00         0       48              0
##   AverageIntensity weekday
## 1         0.333333 Tuesday
## 2         0.133333 Tuesday
## 3         0.116667 Tuesday
## 4         0.000000 Tuesday
## 5         0.000000 Tuesday
## 6         0.000000 Tuesday
```

5. Remove unnecessary variables

In `daily_acitivity` dataset, `TrackerDistance` and `LoggedActivitiesDistance` are not needed as well as `AverageIntensity` in `hourly_activity`. Therefore, they will be removed.

```
daily_activity <- daily_activity %>%
  select(
    -c(
      TrackerDistance,
      LoggedActivitiesDistance
    )
  )

hourly_activity <- hourly_activity %>%
  select(
    -c(
      AverageIntensity
    )
  )
```

6. Clean variable name

Convert the names of the variables into a desired format

```r
daily_activity <- daily_activity %>%
  rename(
    "id" = Id,
    "date" = ActivityDate,
    "total_step" = TotalSteps,
    "total_dist" = TotalDistance,
    "very_active_dist" = VeryActiveDistance,
    "moderate_active_dist" = ModeratelyActiveDistance,
    "light_active_dist" = LightActiveDistance,
    "seden_active_dist" = SedentaryActiveDistance,
    "very_active_min" = VeryActiveMinutes,
    "moderate_active_min" = FairlyActiveMinutes,
    "light_active_min" = LightlyActiveMinutes,
    "seden_active_min" = SedentaryMinutes,
    "calories" = Calories
  )

daily_sleep <- daily_sleep %>%
  rename(
    "id" = Id,
    "date" = ActivityDate,
    "sleep_record" = TotalSleepRecords,
    "asleep_min" = TotalMinutesAsleep,
    "in_bed_min" = TotalTimeInBed
  )

hourly_activity <- hourly_activity %>%
  rename(
    "id" = Id,
    "date" = ActivityDate,
    "hour" = Hour,
    "total_step" = StepTotal,
    "calories" = Calories,
    "total_intensity" = TotalIntensity
  )
```

7. Final datasets

```
##           id       date total_step total_dist very_active_dist
## 1 1503960366 2016-03-25      11004       7.11             2.57
## 2 1503960366 2016-03-26      17609      11.55             6.92
## 3 1503960366 2016-03-27      12736       8.53             4.66
## 4 1503960366 2016-03-28      13231       8.93             3.19
## 5 1503960366 2016-03-29      12041       7.85             2.16
## 6 1503960366 2016-03-30      10970       7.16             2.36
##   moderate_active_dist light_active_dist seden_active_dist very_active_min
## 1                 0.46              4.07                 0              33
## 2                 0.73              3.91                 0              89
## 3                 0.16              3.71                 0              56
## 4                 0.79              4.95                 0              39
## 5                 1.09              4.61                 0              28
## 6                 0.51              4.29                 0              30
##   moderate_active_min light_active_min seden_active_min calories   weekday
## 1                  12              205              804     1819    Friday
## 2                  17              274              588     2154  Saturday
## 3                   5              268              605     1944    Sunday
## 4                  20              224             1080     1932    Monday
## 5                  28              243              763     1886   Tuesday
## 6                  13              223             1174     1820 Wednesday
```

```
##           id       date sleep_record asleep_min in_bed_min   weekday
## 1 1503960366 2016-04-12            1        327        346   Tuesday
## 2 1503960366 2016-04-13            2        384        407 Wednesday
## 3 1503960366 2016-04-15            1        412        442    Friday
## 4 1503960366 2016-04-16            2        340        367  Saturday
## 5 1503960366 2016-04-17            1        700        712    Sunday
## 6 1503960366 2016-04-19            1        304        320   Tuesday
```

```
##           id       date     hour total_step calories total_intensity weekday
## 1 1503960366 2016-04-12 00:00:00        373       81              20 Tuesday
## 2 1503960366 2016-04-12 01:00:00        160       61               8 Tuesday
## 3 1503960366 2016-04-12 02:00:00        151       59               7 Tuesday
## 4 1503960366 2016-04-12 03:00:00          0       47               0 Tuesday
## 5 1503960366 2016-04-12 04:00:00          0       48               0 Tuesday
## 6 1503960366 2016-04-12 05:00:00          0       48               0 Tuesday
```

III. Data dictionary

https://www.fitabase.com/media/1930/fitabasedatadictionary102320.pdf

Data header	Description
id	User unique identifier in 10 digits
date	Data value in yyyy/mm/dd format
total_step	Total number of steps taken
total_dist	Total distance traveled
tracker_dist	Total distance tracked with the device
very_active_dist	Distance travelled during very active activity (kilometers)
moderate_active_dist	Distance travelled in moderate active activity (kilometers)
light_active_dist	Distance travelled in light active activity (kilometers)
seden_active_dist	Distance travelled in sedentary active activity (kilometers)
very_active_min	Total time travelled in very active activity (minutes)
moderate_active_min	Total time travelled in moderate active activity (minutes)
light_active_min	Total time travelled in light active activity (minutes)
seden_active_min	Total time travelled in sedentary active activity (minutes)
calories	Total estimated energy expenditure (kil ocalories)
sleep_record	Number of time classified as being “asleep”
asleep_min	Total of minutes classified as being “asleep”
in_bed_min	Total time in bed, including asleep, restless and awake, that occured during a defined sleep record
hour	Hour value in 24hr format
total_intensity	Value calculated by adding all the minute-level intensity values that occured within the hour

IV. Summarize data statistics

1. Number of records in each dataset

```
daily_activity_distinct_id = n_distinct(daily_activity$id)
daily_sleep_distinct_id = n_distinct(daily_sleep$id)
hourly_activity_distinct_id = n_distinct(hourly_activity$id)

daily_activity_distinct_id
```

```
## [1] 35
```

```
daily_sleep_distinct_id
```

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

```
hourly_activity_distinct_id
```

```
## [1] 35
```

There are 35 distinct users in activity dataset, while only 24 users in daily sleep data.

2. Statistical summaries

A. Daily activity

```
daily_activity %>%
  mutate(id = as.factor((id))) %>%
  summary()
```

```
##           id             date                total_step     total_dist
##  4020332650:  63   Min.   :2016-03-12   Min.   :    0   Min.   : 0.000
##  1503960366:  50   1st Qu.:2016-04-09   1st Qu.: 3146   1st Qu.: 2.170
##  1624580081:  50   Median :2016-04-19   Median : 6999   Median : 4.950
##  4445114986:  46   Mean   :2016-04-19   Mean   : 7281   Mean   : 5.219
##  4702921684:  46   3rd Qu.:2016-04-30   3rd Qu.:10544   3rd Qu.: 7.500
##  6962181067:  45   Max.   :2016-05-12   Max.   :36019   Max.   :28.030
##  (Other)   :1097
##  very_active_dist moderate_active_dist light_active_dist seden_active_dist
##  Min.   : 0.000   Min.   :0.0000       Min.   : 0.000    Min.   :0.000000
##  1st Qu.: 0.000   1st Qu.:0.0000       1st Qu.: 1.610    1st Qu.:0.000000
##  Median : 0.100   Median :0.2000       Median : 3.240    Median :0.000000
##  Mean   : 1.397   Mean   :0.5385       Mean   : 3.193    Mean   :0.001704
##  3rd Qu.: 1.830   3rd Qu.:0.7700       3rd Qu.: 4.690    3rd Qu.:0.000000
##  Max.   :21.920   Max.   :6.4800       Max.   :12.510    Max.   :0.110000
##
##  very_active_min  moderate_active_min light_active_min seden_active_min
##  Min.   :  0.00   Min.   :  0.0       Min.   :  0.0    Min.   :   0.0
##  1st Qu.:  0.00   1st Qu.:  0.0       1st Qu.:111.0    1st Qu.: 729.0
##  Median :  2.00   Median :  6.0       Median :195.0    Median :1057.0
##  Mean   : 19.68   Mean   : 13.4       Mean   :185.4    Mean   : 992.5
##  3rd Qu.: 30.00   3rd Qu.: 18.0       3rd Qu.:262.0    3rd Qu.:1244.0
##  Max.   :210.00   Max.   :660.0       Max.   :720.0    Max.   :1440.0
##
##     calories          weekday
##  Min.   :   0   Monday   :188
##  1st Qu.:1799   Tuesday  :225
##  Median :2114   Wednesday:198
##  Mean   :2266   Thursday :195
##  3rd Qu.:2770   Friday   :199
##  Max.   :4900   Saturday :199
##                 Sunday   :193
```

Insights
- Users took 7281 steps or 5,2 km daily in average. Which are considered to be low active for an average adult.
- Light activity was recorded in most of the time & distance travelled (285 mins and 3.2km average)

B. Daily sleep

```
daily_sleep %>%
  mutate(id = as.factor((id))) %>%
  summary()
```

```
##           id            date                sleep_record     asleep_min
##  8378563200: 32   Min.   :2016-04-12   Min.   :1.000   Min.   : 58.0
##  5553957443: 31   1st Qu.:2016-04-19   1st Qu.:1.000   1st Qu.:361.0
##  6962181067: 31   Median :2016-04-27   Median :1.000   Median :433.0
##  2026352035: 28   Mean   :2016-04-26   Mean   :1.119   Mean   :419.5
##  3977333714: 28   3rd Qu.:2016-05-04   3rd Qu.:1.000   3rd Qu.:490.0
##  4445114986: 28   Max.   :2016-05-12   Max.   :3.000   Max.   :796.0
##  (Other)   :235
##    in_bed_min         weekday
##  Min.   : 61.0   Monday   :47
##  1st Qu.:403.0   Tuesday  :65
##  Median :463.0   Wednesday:66
##  Mean   :458.6   Thursday :65
##  3rd Qu.:526.0   Friday   :57
##  Max.   :961.0   Saturday :58
##                  Sunday   :55
```

Insights
- User sleeps 520 mins daily ~ 7 hours a day
- Max sleeping time recored to be 796 mins ~ 13 hours in a single day

C. Hourly activity

```
hourly_activity %>%
  mutate(id = as.factor((id))) %>%
  summary()
```

```
##           id              date                hour             total_step     
##  1624580081: 1480   Min.   :2016-03-12   Length:46008       Min.   :    0.0  
##  1927972279: 1480   1st Qu.:2016-03-26   Class :character   1st Qu.:    0.0  
##  2022484408: 1480   Median :2016-04-10   Mode  :character   Median :   21.0  
##  2026352035: 1480   Mean   :2016-04-10                      Mean   :  302.9  
##  4558609924: 1480   3rd Qu.:2016-04-25                      3rd Qu.:  323.0  
##  2320127002: 1479   Max.   :2016-05-12                      Max.   :10565.0  
##  (Other)   :37129                                                            
##     calories      total_intensity       weekday    
##  Min.   : 42.00   Min.   :  0.00   Monday   :6581  
##  1st Qu.: 62.00   1st Qu.:  0.00   Tuesday  :6756  
##  Median : 80.00   Median :  2.00   Wednesday:6691  
##  Mean   : 95.82   Mean   : 11.42   Thursday :6427  
##  3rd Qu.:106.00   3rd Qu.: 15.00   Friday   :6134  
##  Max.   :948.00   Max.   :180.00   Saturday :6760  
##                                    Sunday   :6659  
```

Insights
- The significant difference between the max value of `total_step`, `calories` and `total_intensity` suggests that there are possibly a group of users that are far more active compared to the average users

V. Exploratory Descriptive Analysis (EDA) - by each dataset

1. Daily activity

A. Determine using frequency

The data possess users usage data of a period of 62 days. We will divide the users into different categories based on their device total using day:

- 0-12 days: Rarely
- 13-30 days: Sometimes
- 31-46 days: Often
- 47-61 days: Usually
- 62 days: Always

Note: The frequency table is based on Reverso dictionary

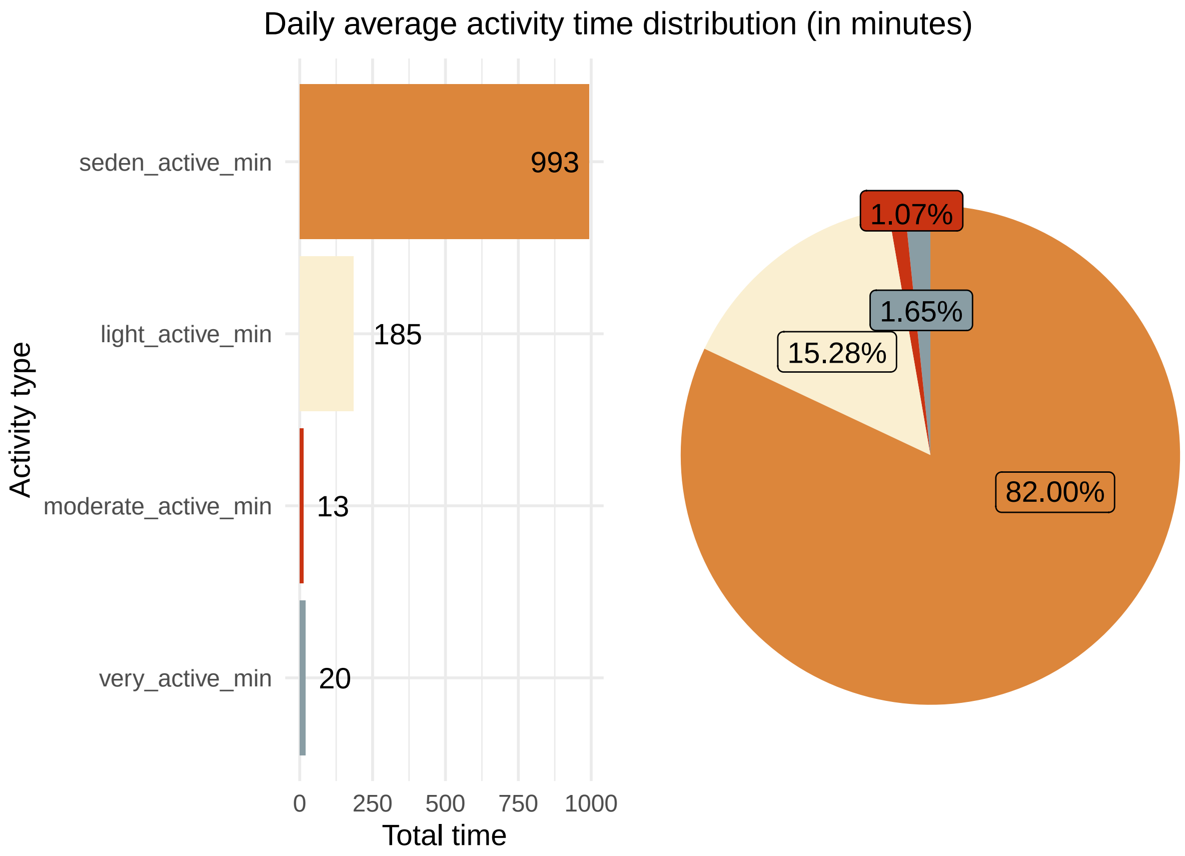
```r
# Determine total using day and usage frequency of each ID
freq_count <- daily_activity %>%
  group_by(id)%>%
  summarize(
    total_using_day = n_distinct(date)
  )%>%
  mutate(
    usage_frequency = case_when(
      0 < total_using_day & total_using_day < 13 ~ "Rarely",
      13 <= total_using_day & total_using_day < 31 ~ "Sometimes",
      31 <= total_using_day & total_using_day  < 47 ~ "Often",
      47 <= total_using_day & total_using_day < 62 ~ "Usually",
      total_using_day == 62 ~ "Always"
    ),
    usage_frequency = factor(
      usage_frequency,
      levels = c(
        "Rarely",
        "Sometimes",
        "Often",
        "Usually",
        "Always"
      )
    )
  )

# Determine the number of user in each usage frequency
freq_count %>%
  group_by(usage_frequency)%>%
  count()
```

```
## # A tibble: 5 x 2
## # Groups:   usage_frequency [5]
##   usage_frequency     n
##   <fct>           <int>
## 1 Rarely              2
## 2 Sometimes           2
## 3 Often              28
## 4 Usually             2
## 5 Always              1
```

Using frequency - Rarely use users (2 people, 1-30% of 62 days)
- Sometimes use users (2 people, 30-49% of 62 days)
- Often use users (28 people, 50-79% of 62 days)
- Usually use users (2 people, 80-99% of 62 days)
- Always use users (1 people, 100% of 62 days)

B. Daily average activity time distribution

```
a1 <- daily_activity %>%
  summarize(
    very_active_min = mean(very_active_min),
    moderate_active_min = mean(moderate_active_min),
    light_active_min = mean(light_active_min),
    seden_active_min = mean(seden_active_min)
  )%>%
  summarize(
    very_active_min = mean(very_active_min),
    moderate_active_min = mean(moderate_active_min),
    light_active_min = mean(light_active_min),
    seden_active_min = mean(seden_active_min)
  )%>%
  pivot_longer(
    cols = everything(),
    names_to = "activity_type",
    values_to = "total_time"
  )%>%
  mutate(
    activity_type = factor(
      activity_type,
      levels = c("very_active_min", "moderate_active_min",
                 "light_active_min", "seden_active_min")
    ),
    total_time = round(total_time, 0)
  )%>%
  ggplot(
    aes(
      x = total_time,
      y = activity_type,
      fill = activity_type
    )
  )+
  geom_bar(
    stat = "identity",
    show.legend = FALSE
  )+
  geom_text(
    aes(
      label = total_time
    ),
    position = position_stack(),
    hjust = c(-0.4,-0.4,-0.4,1.2),
    show.legend = FALSE
  )+
  scale_fill_manual(
    values = wes_palette(
      name = "Royal1",
      n = 4
    )
  )+
  labs(
    x = "Total time",
    y = "Activity type",
    fill = "Total time"
  )+
  theme_minimal()

a2 <- daily_activity %>%
  summarize(
    very_active_min = mean(very_active_min),
    moderate_active_min = mean(moderate_active_min),
    light_active_min = mean(light_active_min),
    seden_active_min = mean(seden_active_min)
  )%>%
  summarize(
    very_active_min = mean(very_active_min),
    moderate_active_min = mean(moderate_active_min),
    light_active_min = mean(light_active_min),
    seden_active_min = mean(seden_active_min)
  )%>%
  pivot_longer(
    cols = everything(),
    names_to = "activity_type",
    values_to = "total_time"
  )%>%
  mutate(
    activity_type = factor(
      activity_type,
      levels = c("very_active_min", "moderate_active_min",
                  "light_active_min", "seden_active_min")
    ),
    total_time = round(total_time, 0)
  )%>%
  ggplot(
    aes(
      x = "",
      y = total_time/1211,
      fill = activity_type
    )
  )+
  geom_bar(
    stat = "identity",
    width = 1,
    show.legend = FALSE
  )+
  coord_polar(
    "y",
    start = 0
  )+
  geom_label(
    aes(
      label = percent(total_time/1211)
      ),
    position = position_stack(vjust = 0.3),
    vjust = c(0,-2.5,0,1.5),
    color = "black",
    show.legend = FALSE,
    size = 4
  )+
  scale_fill_manual(
    values = wes_palette(
    n = 4,
    name = "Royal1"
    )
  )+
  labs(
    x = "",
    y = "",
    fill = "Total time"
  )+
  theme_void()

grid.arrange(
  a1,a2,
  nrow = 1,
  top = "Daily average activity time distribution (in minutes)"
)
```



- On average, 82% of the time was spent in sedentary while people only active in 18% daily.
- When in active, only 20 minutes of the day are used for very active activities, 13 for moderate activities while light active activities take 185 mins (3 hours 5 mins).

The tracker's main feature is to measure the ***total steps taken***, therefore we can assume that their is always a strong positive relationship between ***total steps taken*** and the ***total distance travelled/ total calories burned***.

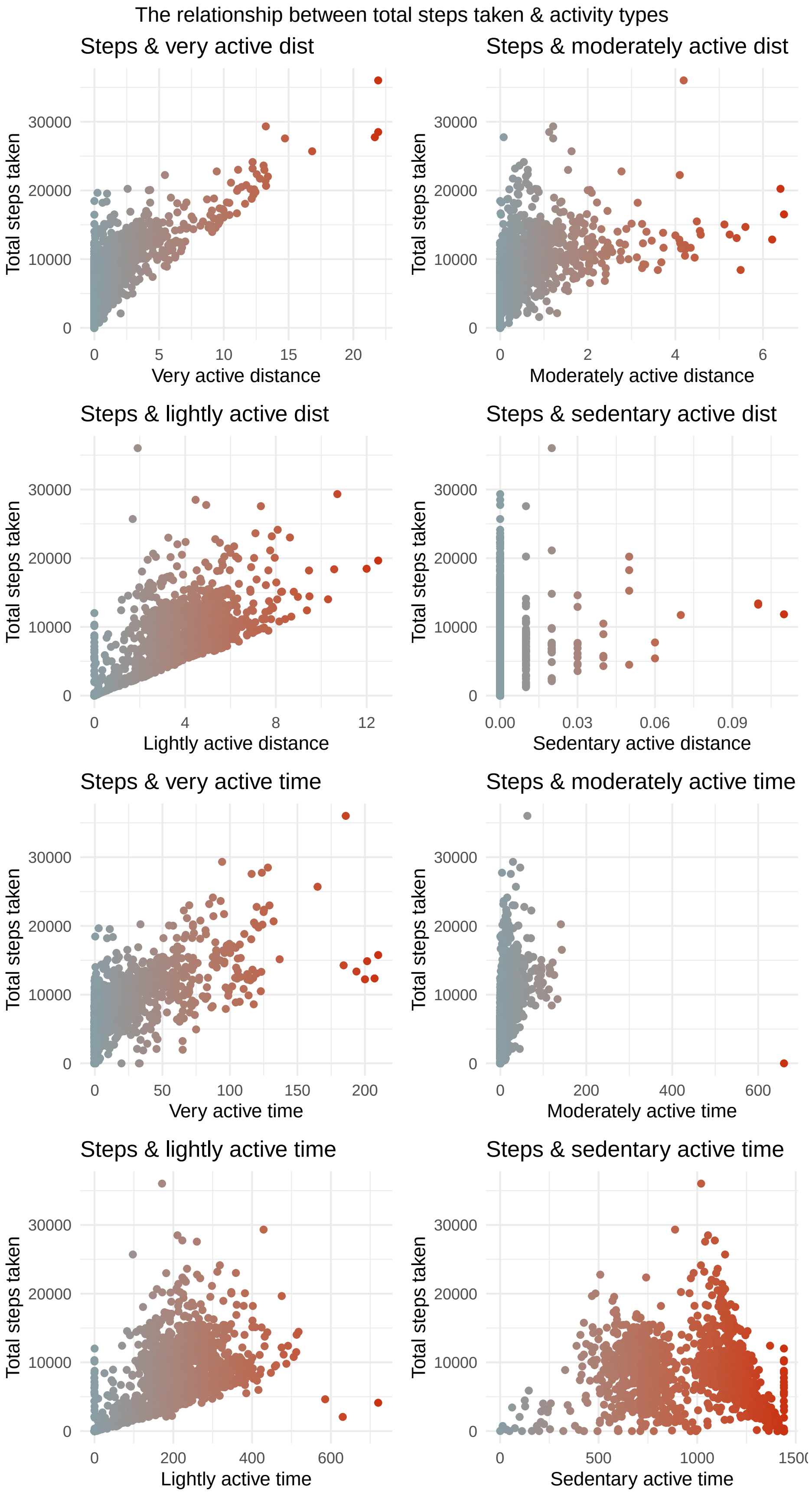
C. Correlation: Relationship between total steps taken and active type

```r
# Total steps vs very active distance
g1 <- daily_activity %>%
  ggplot(
    aes(
      x = very_active_dist,
      y = total_step,
      color = very_active_dist
    )
  )+
  geom_jitter(
  )+
  scale_color_gradientn(
    colors = wes_palette(
      name = "Royal1",
      n = 2
    )
  )+
  labs(
    x = "Very active distance",
    y = "Total steps taken",
    title = "Steps & very active dist"
  )+
  theme_minimal()+
  theme(
    legend.position = 'none'
  )


# Total steps vs moderate active distance
g2 <- daily_activity %>%
  ggplot(
    aes(
      x = moderate_active_dist,
      y = total_step,
      color = moderate_active_dist
    )
  )+
  geom_jitter(
  )+
  scale_color_gradientn(
    colors = wes_palette(
      name = "Royal1",
      n = 2
    )
  )+
  labs(
    x = "Moderately active distance",
    y = "Total steps taken",
    title = "Steps & moderately active dist"
  )+
  theme_minimal()+
  theme(
    legend.position = 'none'
  )

# Total steps vs light active distance
g3 <- daily_activity %>%
  ggplot(
    aes(
      x = light_active_dist,
      y = total_step,
      color = light_active_dist
    )
  )+
  geom_jitter(
  )+
  scale_color_gradientn(
    colors = rev(wes_palette(
      name = "Moonrise1",
      n = 3
    ))
  )+
  scale_color_gradientn(
    colors = wes_palette(
      name = "Royal1",
      n = 2
    )
  )+
  labs(
    x = "Lightly active distance",
    y = "Total steps taken",
    title = "Steps & lightly active dist"
  )+
  theme_minimal()+
  theme(
    legend.position = 'none'
  )
```

```
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.
```

```
# Total steps vs sedentary active distance
g4 <- daily_activity %>%
  ggplot(
    aes(
      x = seden_active_dist,
      y = total_step,
      color = seden_active_dist
    )
  )+
  geom_jitter(
  )+
  scale_color_gradientn(
    colors = wes_palette(
      name = "Royal1",
      n = 2
    )
  )+
  labs(
    x = "Sedentary active distance",
    y = "Total steps taken",
    title = "Steps & sedentary active dist"
  )+
  theme_minimal()+
  theme(
    legend.position = 'none'
  )


# Total steps vs very active time
g5 <- daily_activity %>%
  ggplot(
    aes(
      x = very_active_min,
      y = total_step,
      color = very_active_min
    )
  )+
  geom_jitter(
  )+
  scale_color_gradientn(
    colors = wes_palette(
      name = "Royal1",
      n = 2
    )
  )+
  labs(
    x = "Very active time",
    y = "Total steps taken",
    title = "Steps & very active time"
  )+
  theme_minimal()+
  theme(
    legend.position = 'none'
  )

# Total steps vs moderate active time
g6 <- daily_activity %>%
  ggplot(
    aes(
      x = moderate_active_min,
      y = total_step,
      color = moderate_active_min
    )
  )+
  geom_jitter(
  )+
  scale_color_gradientn(
    colors = wes_palette(
```

```r
      name = "Royal1",
      n = 2
    )
  )+
  labs(
    x = "Moderately active time",
    y = "Total steps taken",
    title = "Steps & moderately active time"
  )+
  theme_minimal()+
  theme(
    legend.position = 'none'
  )

# Total steps vs light active time
g7 <- daily_activity %>%
  ggplot(
    aes(
      x = light_active_min,
      y = total_step,
      color = light_active_min
    )
  )+
  geom_jitter(
  )+
  scale_color_gradientn(
    colors = wes_palette(
      name = "Royal1",
      n = 2
    )
  )+
  labs(
    x = "Lightly active time",
    y = "Total steps taken",
    title = "Steps & lightly active time"
  )+
  theme_minimal()+
  theme(
    legend.position = 'none'
  )


# Total steps vs sedentary active time
g8 <- daily_activity %>%
  ggplot(
    aes(
      x = seden_active_min,
      y = total_step,
      color = seden_active_min
    )
  )+
  geom_jitter(
  )+
  scale_color_gradientn(
    colors = wes_palette(
      name = "Royal1",
      n = 2
    )
  )+
  labs(
    x = "Sedentary active time",
    y = "Total steps taken",
    title = "Steps & sedentary active time"
  )+
  theme_minimal()+
  theme(
    legend.position = 'none'
  )

grid.arrange(
  g1,g2,g3,g4,
  g5,g6,g7,g8,
  nrow = 4,
  ncol = 2,
  top = "The relationship between total steps taken & activity types")
```

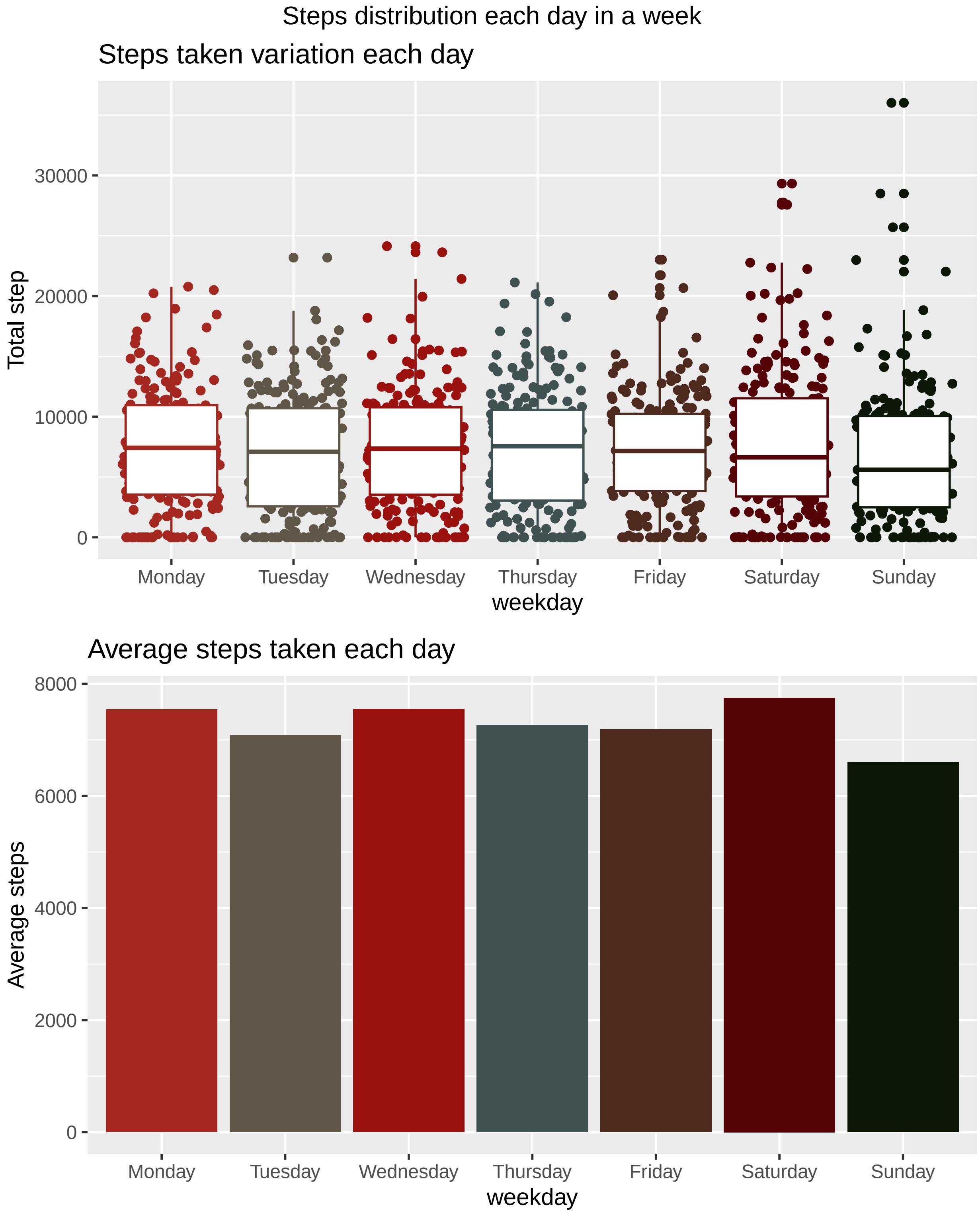

The relationship between total steps taken & activity types
Steps & very active dist
Total steps taken
30000
20000
10000
0
0
5
10
15
20
Very active distance
Steps & moderately active dist
Total steps taken
30000
20000
10000
0
0
2
4
6
Moderately active distance
Steps & lightly active dist
Total steps taken
30000
20000
10000
0
0
4
8
12
Lightly active distance
Steps & sedentary active dist
Total steps taken
30000
20000
10000
0
0.00
0.03
0.06
0.09
Sedentary active distance
Steps & very active time
Total steps taken
30000
20000
10000
0
0
50
100
150
200
Very active time
Steps & moderately active time
Total steps taken
30000
20000
10000
0
0
200
400
600
Moderately active time
Steps & lightly active time
Total steps taken
30000
20000
10000
0
0
200
400
600
Lightly active time
Steps & sedentary active time
Total steps taken
30000
20000
10000
0
0
500
1000
150
Sedentary active time

From the plot, we can clearly see that there is:
- A strong relationship between total steps taken and light active distance/time
- A insignificant relationshop between total steps taken and moderate/very active distance/time

The plots show that most of the customers spend time walk lightly everyday and most of their steps are taken in low intensity. This information suggests that the customers mainly are normal people/workers. Moreover, the relationship between total steps taken and moderate/very active distance proves that they may still take daily walk or other moving exercises.

D. Steps by weekday

```r
h1 <- daily_activity %>%
  group_by(weekday) %>%
  ggplot(
    aes(
      x = weekday,
      y = total_step,
      color = weekday
    )
  )+
  geom_jitter(
    show.legend = FALSE
  )+
  geom_boxplot(
    show.legend = FALSE
  )+
  labs(
    title = "Steps taken variation each day",
    y = "Total step"
  )+
  scale_color_manual(
    values = wes_palette(
      name = "BottleRocket1"
    )
  )

h2 <- daily_activity %>%
  group_by(weekday)%>%
  summarize(
    avg_step = mean(total_step)
  )%>%
  ggplot(
    aes(
      x = weekday,
      y = avg_step,
      fill = weekday
    )
  )+
  geom_bar(
    stat = "identity",
    show.legend = FALSE
  )+
  labs(
    title = "Average steps taken each day",
    y = "Average steps"
  )+
  scale_fill_manual(
    values = wes_palette(
      name = "BottleRocket1"
    )
  )

grid.arrange(
  h1, h2,
  nrow = 2,
  top = "Steps distribution each day in a week"
)
```



Aside from working days, when people are active overally, it is noticable that:
- A significant larger amount of steps on Saturday: Possibly due to users usually spend more time outside, which may leads to more steps taken
- A large drop on steps taken on Sunday: Could be a day off in the week when people spend most of the time rest/indoor.
- People are likely to take more steps in the weekend (there is a considerable amount of people having more than 20k steps).

2. Daily sleep

A. Take a look at the means of the variable

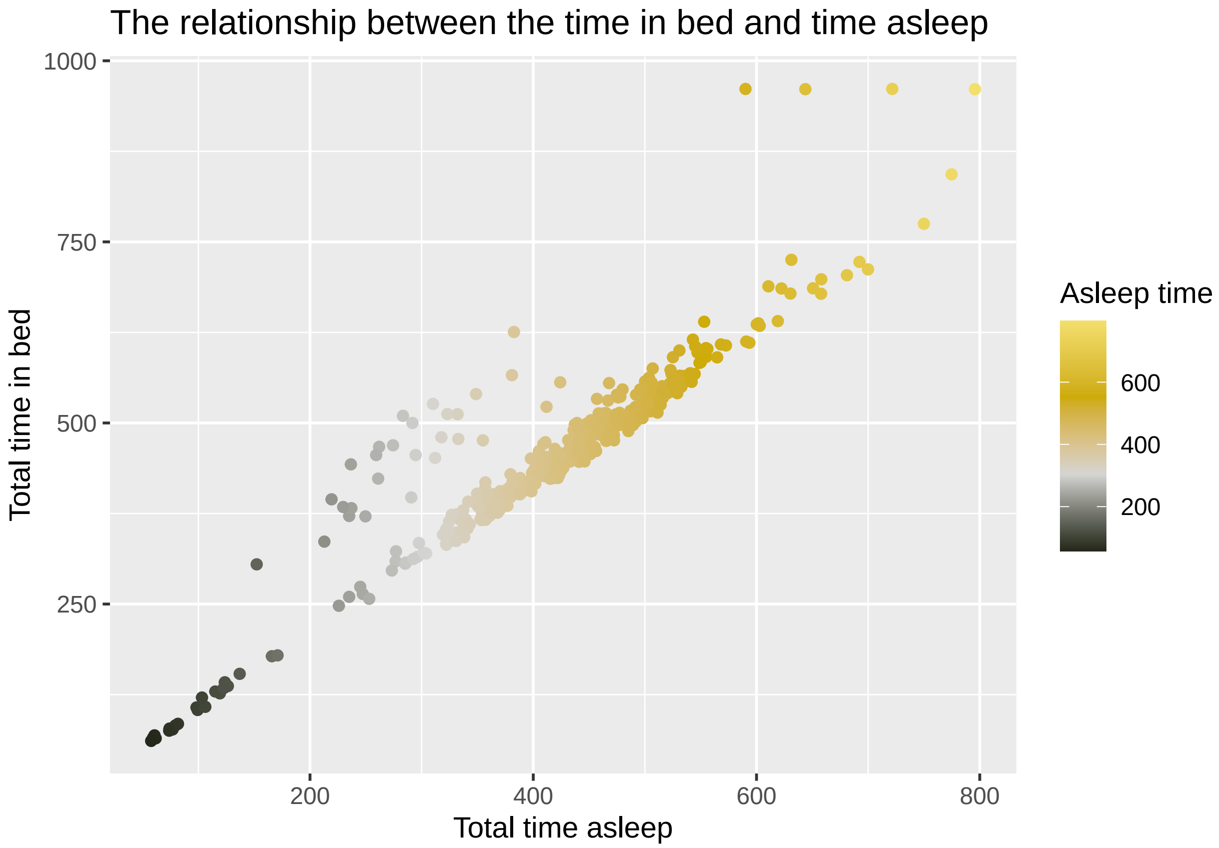
```
options(scipen = 999)

daily_sleep %>%
  ungroup()%>%
  select(-c(id,weekday, date, sleep_record))%>%
  summarize_all(mean)
```

```
##   asleep_min in_bed_min
## 1   419.4673   458.6392
```

Average time asleep for a day is 420 minutes ~ 7 hours while, average time in bed of the participants is 459 minutes ~ 7 hour 39 mins. This means that aside from sleep, people spend another 39 minutes on average in bed.

B. Correlation: Relationship between total minute asleep vs total time in bed

```
daily_sleep %>%
  ggplot(
    aes(
      x = asleep_min,
      y = in_bed_min,
      color = asleep_min
    )
  )+
  geom_jitter(
  )+
  scale_color_gradientn(
    colors = rev(wes_palette(
      name = "Moonrise1",
      type = "continuous"
    ))
  )+
  labs(
    x = "Total time asleep",
    y = "Total time in bed",
    color = "Asleep time",
    title = "The relationship between the time in bed and time asleep"
  )
```



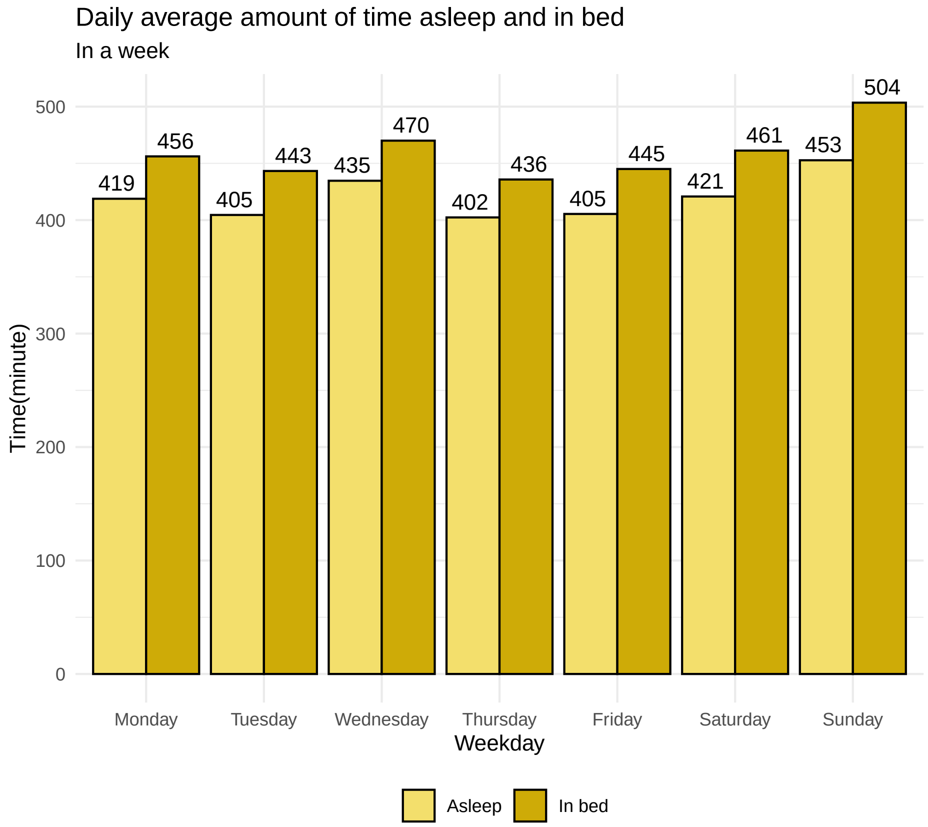
From the graph, we can see the relationship between total time asleep and time in bed, this shows that participants are likely to..sleep when they are in bed (and not do other activities).

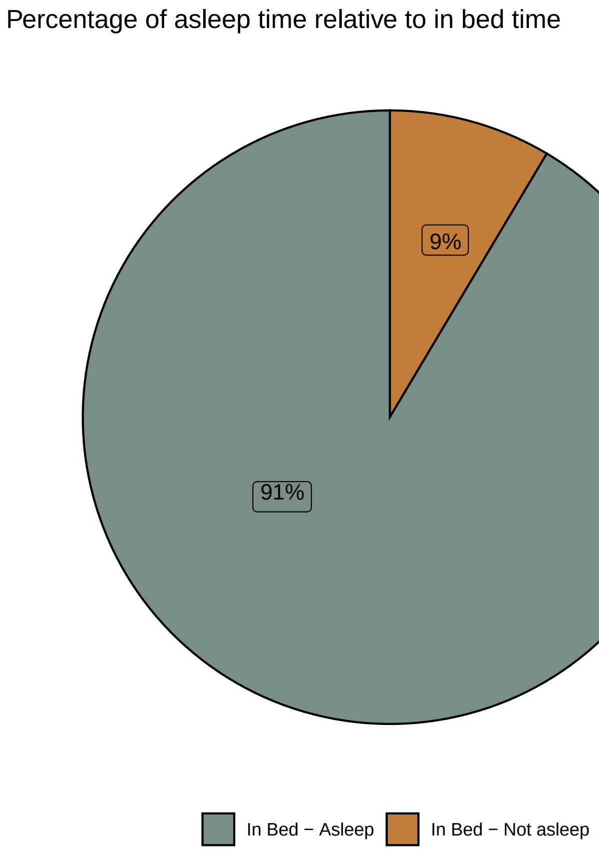
C. Average amount of time asleep and in bed in a week

```
f1 <- daily_sleep %>%
  group_by(
    weekday
  )%>%
  summarize(
    asleep_min = mean(asleep_min),
    in_bed_min = mean(in_bed_min)
  )%>%
  pivot_longer(
    !weekday,
    names_to = "label",
    values_to = "time"
  )%>%
  ggplot(
    aes(
      x = weekday,
      y = time,
      fill = label
    )
  )+
  geom_bar(
    color = "black",
    stat = "identity",
    position = position_dodge()
  )+
  geom_text(
    aes(
      label = round(time, 0)
    ),
    color ="black",
    position = position_dodge(
      width = 1
    ),
    vjust = -0.5
  )+
  scale_fill_manual(
    values = wes_palette(
      name = "Moonrise1",
      n = 2
    ),
    labels = c('Asleep','In bed')
  )+
  labs(
    title = "Daily average amount of time asleep and in bed",
    subtitle = "In a week",
    x = "Weekday",
    y = "Time(minute)",
    fill = ''

  )+
  theme_minimal()+
  theme(
```

```r
    legend.position = 'bottom'
  )


daily_sleep_summary <- daily_sleep %>%
  summarize(
    asleep_min = mean(asleep_min),
    in_bed_min = mean(in_bed_min)
  )%>%
  mutate(
    non_asleep = in_bed_min - asleep_min
  )


f2 <- data.frame(
  label = c("In Bed - Not asleep", "In Bed - Asleep"),
  value = c(daily_sleep_summary$non_asleep, daily_sleep_summary$asleep_min)
  )%>%
  ggplot(
    aes(
      x = "",
      y = value,
      fill = label
    )
  )+
  geom_bar(
    color = "black",
    stat = "identity",
    width = 1
  )+
  coord_polar(
    theta = "y",
    start = 0
  )+
  labs(
    title = "Percentage of asleep time relative to in bed time",
    fill = ""
  )+
  scale_fill_manual(
    values = wes_palette(
      name = "Moonrise2"
    )
  )+
  geom_label(
    aes(
      label = percent(value / sum(value)),
      y = value
    ),
    hjust = c(1,1),
    vjust = c(-1,7.5),
    color = "black",
    size = 4,
    show.legend = FALSE
  )+
  theme_void()+
  theme(
    legend.position = 'bottom'
  )


grid.arrange(
  f1,f2,
  nrow = 2,
  top = "Average sleep time distribution"
)
```

Average sleep time distribution





Observing the plot, it's noticable that:
- The average sleep time daily is always more than 400 minutes (6 hours 40 mins).
- People always spend an extra of 30-50 mins in bed without sleep.
- On Sunday, the participants' sleeping time is longest with an average of 503 mins ~ 8.4 hourse in bed and 452 mins ~ 7 hours 32 mins sleep.

3. Hourly activity: Average distribution of steps, calories, intensity

Average step distribution using bar graph and heatmap

```
d1 <- hourly_activity %>%
  group_by(hour)%>%
  summarize(total_step = mean(total_step)) %>%
  ggplot()+
  geom_col(
    mapping=aes(
      x = hour,
      y = total_step,
      fill = total_step
    )
  )+
  labs(
    title = "Average steps distribution",
    subtitle = "In a day",
    x = "Time",
    y = "Steps",
    fill = "Total steps"
  )+
  scale_fill_gradientn(
    colors = wes_palette(
      name = "Zissou1",
      n = 5
    )
  )+
  theme_classic(
  )+
  theme(axis.text.x = element_text(angle = 90))

d2 <- hourly_activity %>%
  group_by(weekday,hour) %>%
  summarize(total_step = mean(total_step)) %>%
  ggplot(
    mapping = aes(
      x = weekday,
      y = hour
    )
  )+
  geom_tile(
    aes(fill= total_step)
  )+
  scale_fill_gradientn(
    colors = wes_palette(
      name = "Zissou1",
      n = 5
    )
  )+
  labs(
    title = '',
    subtitle = "In a week",
    x = "Weekday",
    y = "Time",
    fill = "Total steps"
  )+
  theme_classic(
  )+
  theme(axis.text.x = element_text(angle = 90))
```

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

Average calories distribution using bar graph and heatmap

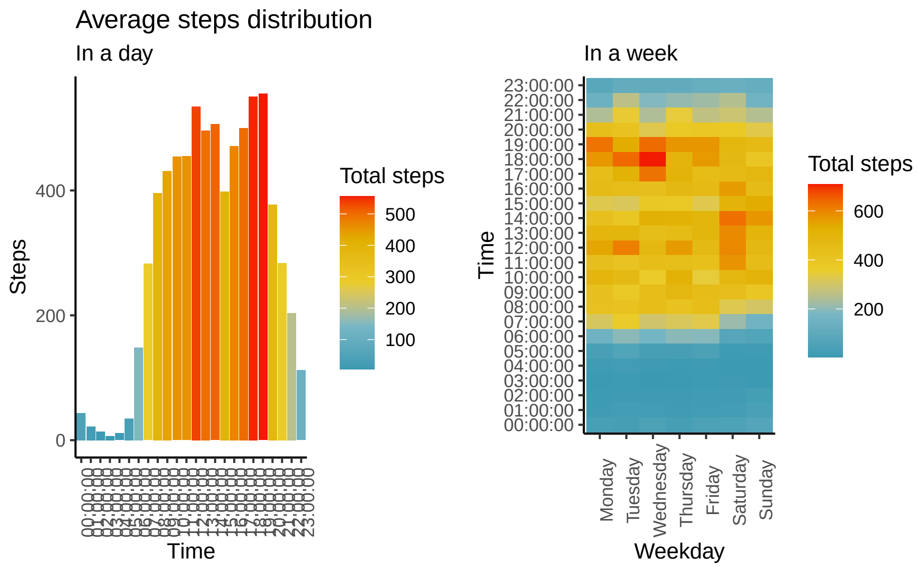
```
d3 <- hourly_activity %>%
  group_by(hour)%>%
  summarize(calories = mean(calories)) %>%
  ggplot()+
  geom_col(
    mapping=aes(
      x = hour,
      y = calories,
      fill = calories
    )
  )+
  labs(
    title = "Average calories distribution",
    subtitle = "In a day",
    x = "Time",
    y = "Calories",
    fill = "Calories"
  )+
  scale_fill_gradientn(
    colors = rev(wes_palette(
      name = "Moonrise1",
      n = 3
    ))
  )+
  theme_classic(
  )+
  theme(axis.text.x = element_text(angle = 90))

d4 <- hourly_activity %>%
  group_by(weekday,hour) %>%
  summarize(calories = mean(calories)) %>%
  ggplot(
    mapping = aes(
      x = weekday,
      y = hour
    )
  )+
  geom_tile(
    aes(fill= calories)
  )+
  scale_fill_gradientn(
    colors = rev(wes_palette(
      name = "Moonrise1",
      n = 3
    ))
  )+
  labs(
    title = '',
    subtitle = "In a week",
    x = "Weekday",
    y = "Time",
    fill = "Calories"
  )+
  theme_classic(
  )+
  theme(axis.text.x = element_text(angle = 90))
```

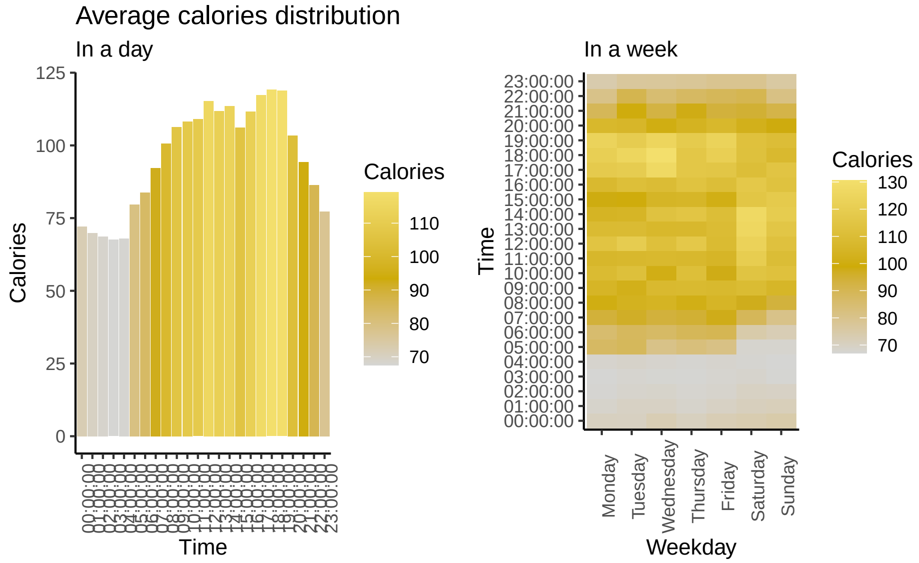
```
## `summarise()` has grouped output by 'weekday'. You can override using the
## `.groups` argument.
```

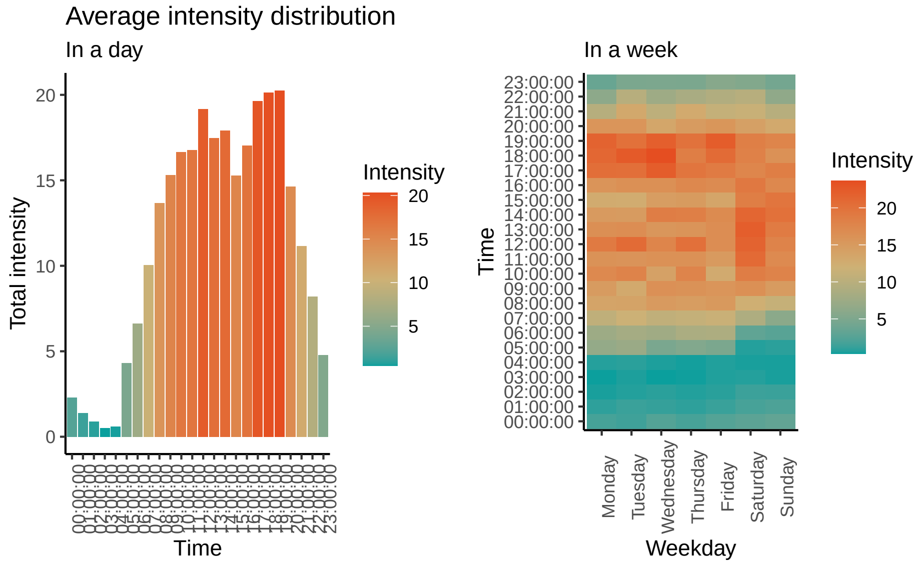
Average intensity distribution using bar graph and heatmap

```
d5 <- hourly_activity %>%
  group_by(hour)%>%
  summarize(total_intensity = mean(total_intensity)) %>%
  ggplot()+
  geom_col(
    mapping=aes(
      x = hour,
      y = total_intensity,
      fill = total_intensity
    )
  )+
  labs(
    title = "Average intensity distribution",
    subtitle = "In a day",
    x = "Time",
    y = "Total intensity",
    fill = "Intensity"
  )+
  scale_fill_gradientn(
    colors = wes_palette(
      name = "AsteroidCity1",
      n = 3
    )
  )+
  theme_classic(
  )+
  theme(axis.text.x = element_text(angle = 90))

d6 <- hourly_activity %>%
  group_by(weekday,hour) %>%
  summarize(total_intensity = mean(total_intensity)) %>%
  ggplot(
    mapping = aes(
      x = weekday,
      y = hour
    )
  )+
  geom_tile(
    aes(fill= total_intensity)
  )+
  scale_fill_gradientn(
    colors = wes_palette(
      name = "AsteroidCity1",
      n = 3
    )
  )+
  labs(
    title = '',
    subtitle = "In a week",
    x = "Weekday",
    y = "Time",
    fill = "Intensity"
  )+
  theme_classic(
  )+
  theme(axis.text.x = element_text(angle = 90))
```

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


Average steps distribution
In a day
Steps
0
200
400
Time
00:00:00
01:00:00
02:00:00
03:00:00
04:00:00
05:00:00
06:00:00
07:00:00
08:00:00
09:00:00
10:00:00
11:00:00
12:00:00
13:00:00
14:00:00
15:00:00
16:00:00
17:00:00
18:00:00
19:00:00
20:00:00
21:00:00
22:00:00
23:00:00
Total steps
500
400
300
200
100
In a week
Time
Weekday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Total steps
600
400
200


Average calories distribution
In a day
Calories
0
25
50
75
100
125
Time
Calories
110
100
90
80
70
In a week
Time
Weekday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Calories
130
120
110
100
90
80
70


Average intensity distribution
In a day
Total intensity
0
5
10
15
20
Time
Intensity
20
15
10
5
In a week
Time
Weekday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Intensity
20
15
10
5

- Participants are active from 7:00 to 21:00 daily, with 2 intensive points at from 12:00 to 14:00 and 17:00 to 19:00.
- On Saturday and Sunday, there is a trend of move less at the evening.
- These 2 time periods are all meal time (while the latter is the getting off work time, workouts and also people may move more to prepare for their dinner).
- At the weekend, people are usually start their days later but move less in the evening and still having the same rest time at the end of the day.
- The heat maps suggest the active pattern of normal office workers

VI. EDA - by dividing users into groups

Adding user segmentation by steps and using frequency

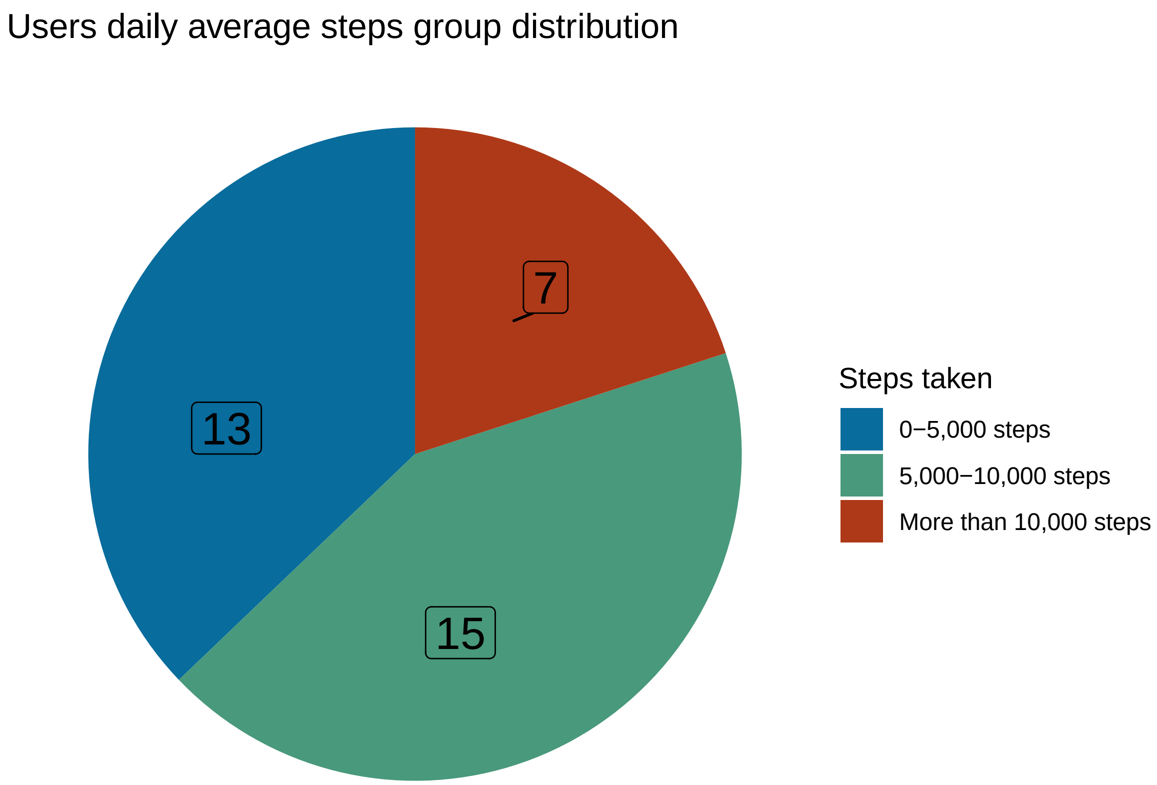
We will divide the users into 3 groups:
- 1st group: Daily average steps taken less than 5000
- 2nd group: Daily average steps taken from 5000 - 10000
- 3rd group: Daily average steps taken more than 10000

```
daily_activity <- daily_activity %>%
  group_by(id)%>%
  mutate(avg_step = mean(total_step))%>%
  mutate(
    group = case_when(
      0 <= avg_step & avg_step < 5000 ~ "1",
      5000 <= avg_step & avg_step < 10000 ~ "2",
      avg_step > 10000 ~ "3",
    )
  )%>%
  mutate(
    group = factor(group, levels = c("1","2","3")
    )
  )%>%
  select(-avg_step)
```

Visualization of segmentation

```
step_count <- daily_activity %>%
  group_by(id) %>%
  summarize(avg_step = mean(total_step)) %>%
  mutate(
    group = case_when(
      0 <= avg_step & avg_step < 5000 ~ "0-5,000 steps",
      5000 <= avg_step & avg_step < 10000 ~ "5,000-10,000 steps",
      avg_step > 10000 ~ "More than 10,000 steps",
    )
  )%>%
  mutate(
    group = factor(
      group,
      levels = c("0-5,000 steps","5,000-10,000 steps","More than 10,000 steps")
    )
  )

step_count %>%
  group_by(
    group
  )%>%
  summarize(
    count = n()
  )%>%
  ggplot(
    aes(
      x = "",
```

```r
    y = count/35,
    fill = group
  )
)+
geom_bar(
  stat = "identity",
  width = 1
)+
coord_polar(
  "y"
)+
scale_fill_manual(
  values = c('#086c9c','#49997c','#ae3918')
)+
geom_label_repel(
  aes(
    label = count,
  ),
  position = position_stack(vjust = 0.5),
  size = 6,
  show.legend = FALSE
)+
labs(
  title = "Users daily average steps group distribution",
  x = "",
  y = "",
  fill = "Steps taken"
)+
theme_void()
```

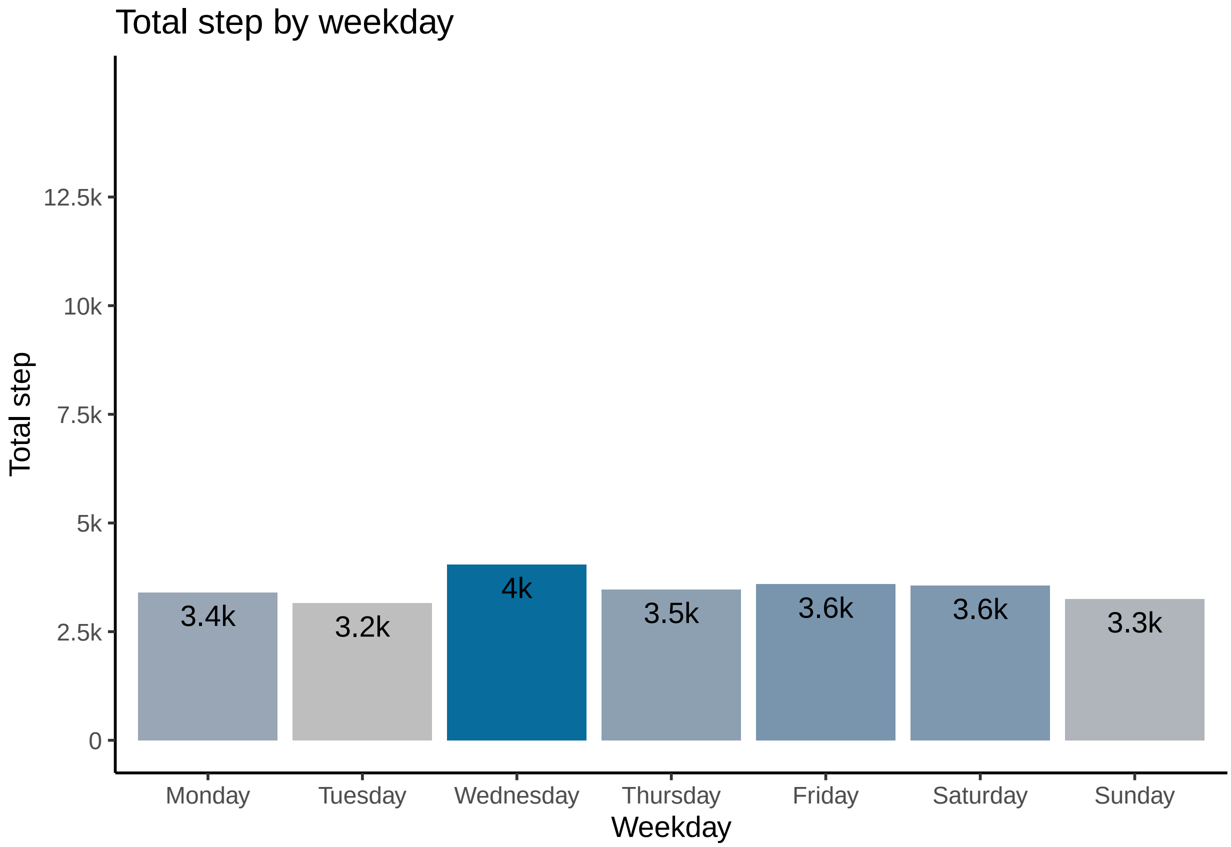


Based on the visualization, we can observe that there is only 7 people (23,3%) possess an average daily step of more than 10,000 while 28 people have less than 10,000 steps a day.

Group 1: Take less than 5,000 steps on daily average

Average steps by weekday

```r
daily_activity %>%
  filter(
    group == 1
  )%>%
  group_by(
    weekday
  )%>%
  summarize(
    total_step = mean(total_step)
  )%>%
  ggplot(
    aes(
      x = weekday,
      y = total_step,
      fill = total_step
    )
  )+
  geom_bar(
    stat = "identity",
    show.legend = FALSE
  )+
  scale_y_continuous(
    limits = c(0, 15000),
    breaks = c(0, 2500, 5000, 7500, 10000, 12500),
    labels = c(0, "2.5k","5k","7.5k","10k","12.5k")
  )+
  geom_text_repel(
    aes(
      label = paste0(round(total_step/1000,1), "k")
    ),
    vjust = 1.6
  )+
  scale_fill_gradient(
    low = "grey",
    high = "#086c9c"
  )+
  labs(
    x = "Weekday",
    y = "Total step",
    title = "Total step by weekday"
  )+
  theme_classic()
```

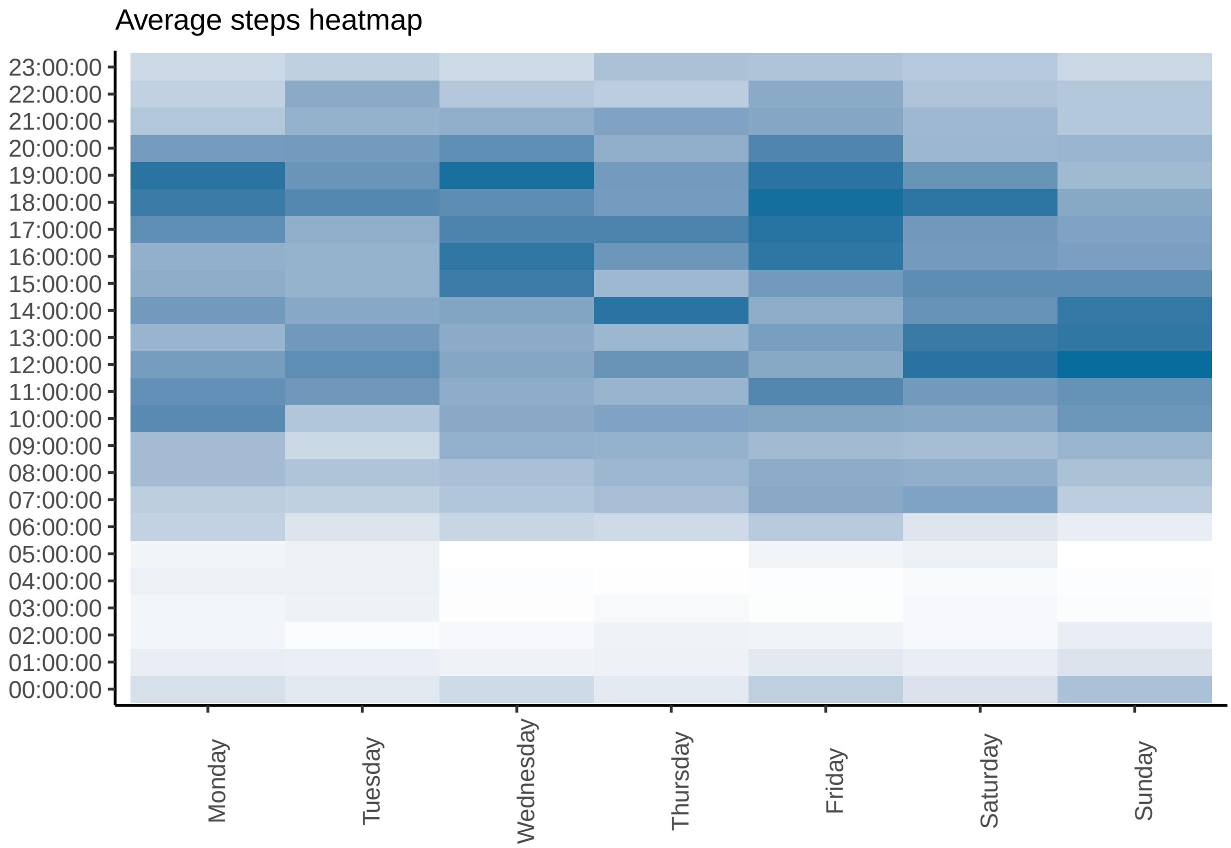


There is no significant difference observed. The average steps taken of users in this group varies from 3.2k to 4k steps a day.

Step heatmap

```
segment <- hourly_activity %>%
  group_by(id, date) %>%
  mutate(
    daily_avg_steps = sum(total_step)
  )

segment %>%
  filter(
    daily_avg_steps < 5000
  )%>%
  group_by(
    weekday,
    hour
  )%>%
  summarize(
    total_step = mean(total_step)
  )%>%
  ggplot(
    mapping = aes(
      x = weekday,
      y = hour
    )
  )+
  geom_tile(
    aes(fill= total_step),
    show.legend = FALSE
  )+
  scale_fill_gradient(
    low = "white",
    high = "#086c9c"
  )+
  labs(
    subtitle = "Average steps heatmap",
    x = NULL,
    y = NULL,
    fill = "Total step"
  )+
  theme_classic(
  )+
  theme(axis.text.x = element_text(angle = 90))
```

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

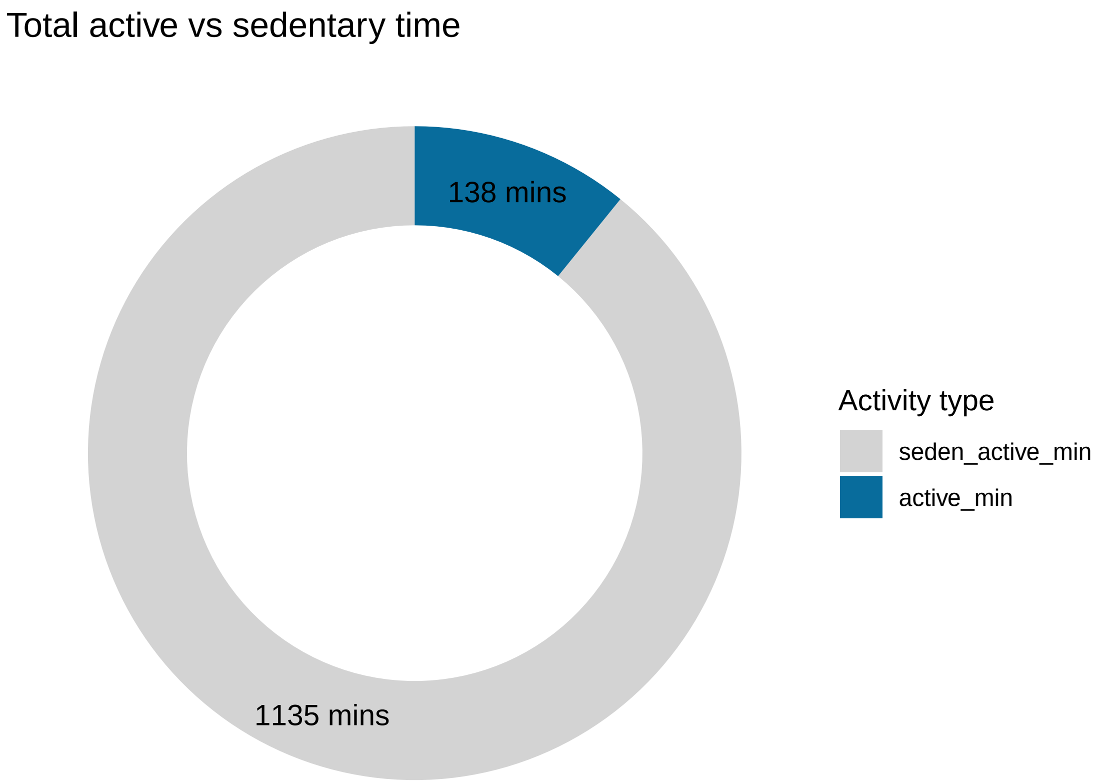


- There are more steps during meal time (around noon, from 18-19h).
- There is a difference between step pattern of weekdays and the weekend. People tend to walk in a shorter time range in the weekend.
- Traces show that users in this group also walk during late night.
- There is no clear step pattern observed. This may suggests that the users did not have a solid active schedule, and the time varies randomly.

Total active vs sedentary time

Note that active time equals to the total of light, moderate and very active time

```r
daily_activity %>%
  filter(
    group == 1
  )%>%
  summarize(
    very_active_min = mean(very_active_min),
    moderate_active_min = mean(moderate_active_min),
    light_active_min = mean(light_active_min),
    seden_active_min = mean(seden_active_min)
  )%>%
  summarize(
    very_active_min = mean(very_active_min),
    moderate_active_min = mean(moderate_active_min),
    light_active_min = mean(light_active_min),
    seden_active_min = mean(seden_active_min)
  )%>%
  mutate(
    active_min = sum(very_active_min, moderate_active_min, light_active_min)
  )%>%
  select(
    -c(very_active_min,moderate_active_min,light_active_min)
  )%>%
  pivot_longer(
    cols = everything(),
    names_to = "activity_type",
    values_to = "total_time"
  )%>%
  mutate(
    activity_type = factor(
      activity_type,
      levels = c("seden_active_min","active_min")
    ),
    total_time = round(total_time, 0)
  )%>%
  ggplot(
    aes(
      x = 3,           # x = 3 = hole size
      y = total_time,
      fill = activity_type
    )
  )+
  geom_bar(
    width = 1,
    stat = "identity"
  )+
  coord_polar(
    theta = "y"
  )+
  xlim(
    c(0.2, 3 + 0.5)    # "3" is hole size
  )+
  geom_text(
    aes(
      label = paste0(total_time, " mins")
    ),
    position = position_stack(vjust = 0.5),
    size = 4,
    show.legend = FALSE
  )+
  scale_fill_manual(
    values =  c("lightgrey","#086c9c")
  )+
  labs(
    x = "",
    y = "",
    fill = "Activity type",
    title = "Total active vs sedentary time"
  )+
  theme_void()
```



Users in group 1 in active for 138 minutes a day ~ 2 hours. Which is a extremely small number compared to sedentary time.

Average total device using days in the whole period

The data possess users usage data of a period of 62 days. We will divide the users into different categories based on their device total using day:
- 0-12 days: Rarely
- 13-30 days: Sometimes
- 31-46 days: Often
- 47-61 days: Usually
- 62 days: Always
Note: The frequency table is based on Reverso dictionary

```r
# Attach total using day count and usage frequency type to each observation
daily_activity <- daily_activity %>%
  group_by(id)%>%
  mutate(
    total_using_day = n_distinct(date)
  )%>%
  mutate(
    usage_frequency = case_when(
      0 < total_using_day & total_using_day < 13 ~ "Rarely",
      13 <= total_using_day & total_using_day < 31 ~ "Sometimes",
      31 <= total_using_day & total_using_day  < 47 ~ "Often",
      47 <= total_using_day & total_using_day < 62 ~ "Usually",
      total_using_day == 62 ~ "Always"
    ),
    usage_frequency = factor(
      usage_frequency,
      levels = c(
        "Rarely",
        "Sometimes",
        "Often",
        "Usually",
        "Always"
      )
    )
  )

daily_activity %>%
  filter(
    group == 1
  )%>%
  summarize(
    total_using_day = mean(total_using_day)
  )%>%
  summarize(
    avg_using_day = mean(total_using_day)
  )
```

```
## # A tibble: 1 x 1
##   avg_using_day
##           <dbl>
## 1          36.2
```

The group has an average total device using day of 36.2, which can be considered as ‘Often’.

Distance travelled

```
daily_activity %>%
  filter(
    group == 1
  )%>%
  summarize(
    very_active_dist = mean(very_active_dist),
    moderate_active_dist = mean(moderate_active_dist),
    light_active_dist = mean(light_active_dist),
    seden_active_dist = mean(seden_active_dist)
  )%>%
  summarize(
    very_active_dist = mean(very_active_dist),
    moderate_active_dist = mean(moderate_active_dist),
    light_active_dist = mean(light_active_dist),
    seden_active_dist = mean(seden_active_dist)
  )%>%
  mutate(
    active_dist = sum(very_active_dist, moderate_active_dist, light_active_dist)
  )%>%
  select(
    -c(very_active_dist,moderate_active_dist,light_active_dist)
  )%>%
  pivot_longer(
    cols = everything(),
    names_to = "activity_type",
    values_to = "total_time"
  )%>%
  mutate(
    activity_type = factor(
      activity_type,
      levels = c("seden_active_dist","active_dist")
    ),
    total_time = round(total_time, 2)
  )
```

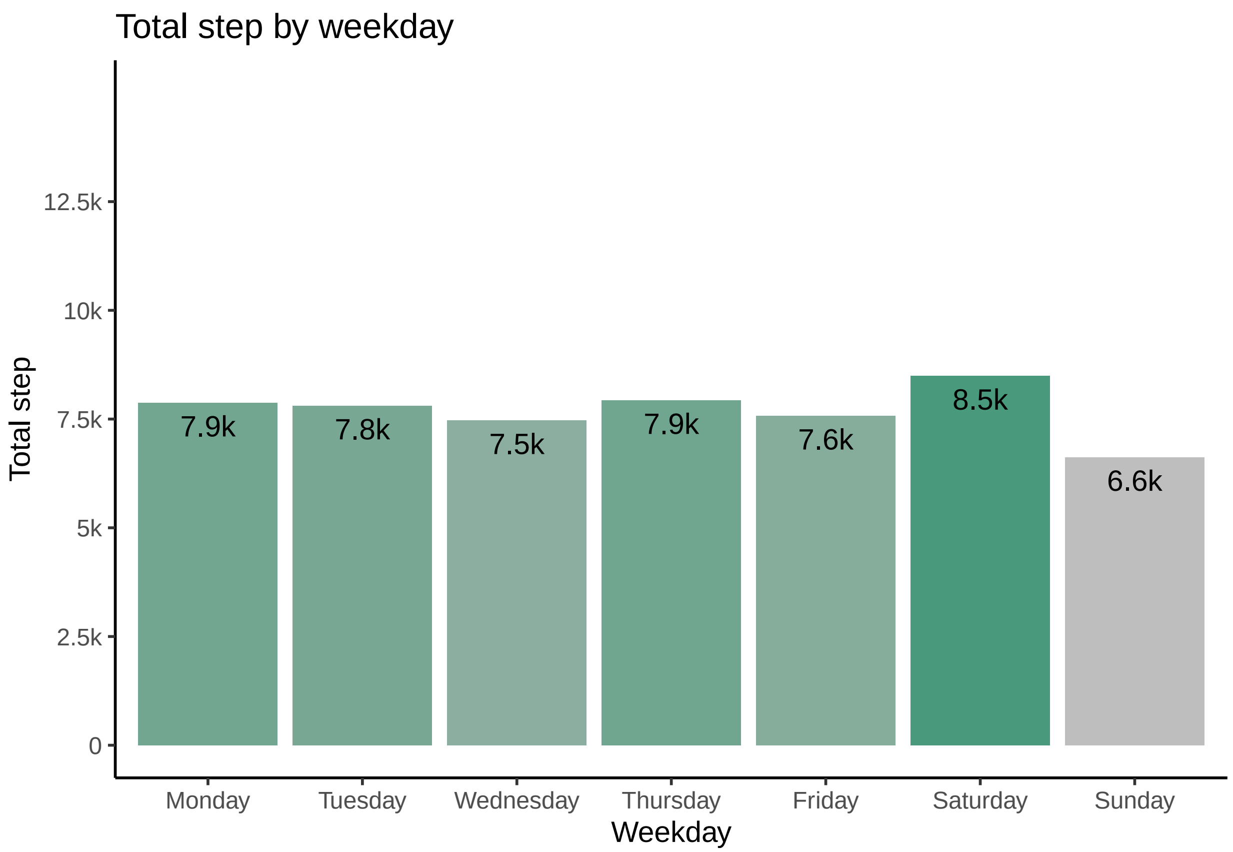
```
## # A tibble: 2 x 2
##   activity_type     total_time
##   <fct>                  <dbl>
## 1 seden_active_dist       0
## 2 active_dist             2.15
```

User group 1 has an average active distance of 2.15km

Group 2: Take from 5,000 - 10,000 steps on daily average

Average steps by weekday

```r
daily_activity %>%
  filter(
    group == 2
  )%>%
  group_by(
    weekday
  )%>%
  summarize(
    total_step = mean(total_step)
  )%>%
  ggplot(
    aes(
      x = weekday,
      y = total_step,
      fill = total_step
    )
  )+
  geom_bar(
    stat = "identity",
    show.legend = FALSE
  )+
  scale_y_continuous(
    limits = c(0, 15000),
    breaks = c(0, 2500, 5000, 7500, 10000, 12500),
    labels = c(0, "2.5k","5k","7.5k","10k","12.5k")
  )+
  geom_text_repel(
    aes(
      label = paste0(round(total_step/1000,1), "k")
    ),
    vjust = 1.6
  )+
  scale_fill_gradient(
    low = "grey",
    high = "#49997c"
  )+
  labs(
    x = "Weekday",
    y = "Total step",
    title = "Total step by weekday"
  )+
  theme_classic()
```

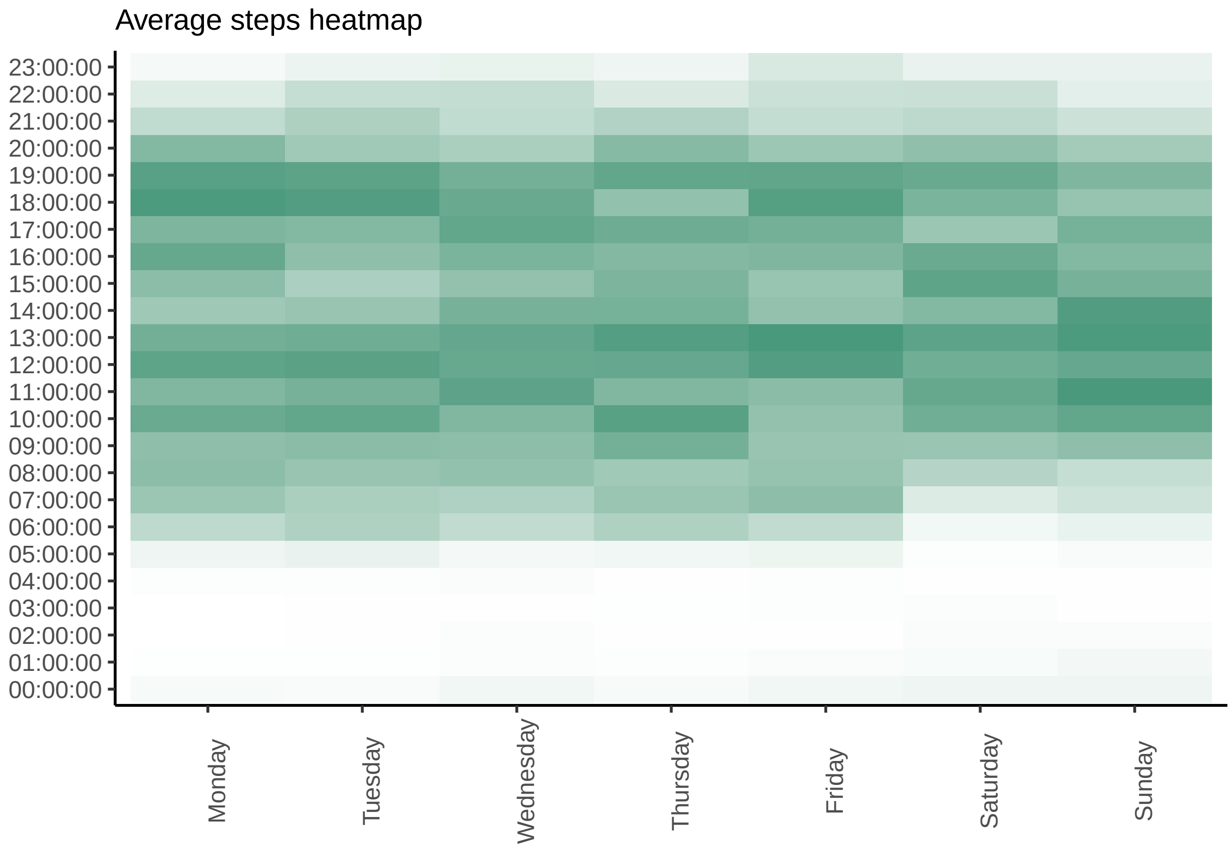


- Group 2 users has an average steps from 7.5k-8.5k in weekdays and Saturday. There is a considerable decrease on Sunday.
- Compared to group 1, this is a quite big difference (around 3-4k/day)

Step heatmap

```r
segment %>%
  filter(
    daily_avg_steps >= 5000,
    daily_avg_steps < 10000
  )%>%
  group_by(
    weekday,
    hour
  )%>%
  summarize(
    total_step = mean(total_step)
  )%>%
  ggplot(
    mapping = aes(
      x = weekday,
      y = hour
    )
  )+
  geom_tile(
    aes(fill= total_step),
    show.legend = FALSE
  )+
  scale_fill_gradient(
    low = "white",
    high = "#49997c"
  )+
  labs(
    subtitle = "Average steps heatmap",
    x = NULL,
    y = NULL,
    fill = "Total step"
  )+
  theme_classic(
  )+
  theme(axis.text.x = element_text(angle = 90))
```

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



- The heatmap seems to have a clearer step pattern. Users tend to starts their walk from 6:00 daily and rest before midnight.

- During weekdays, it visible that users walk more during meal time (12-13h and 18-19h).

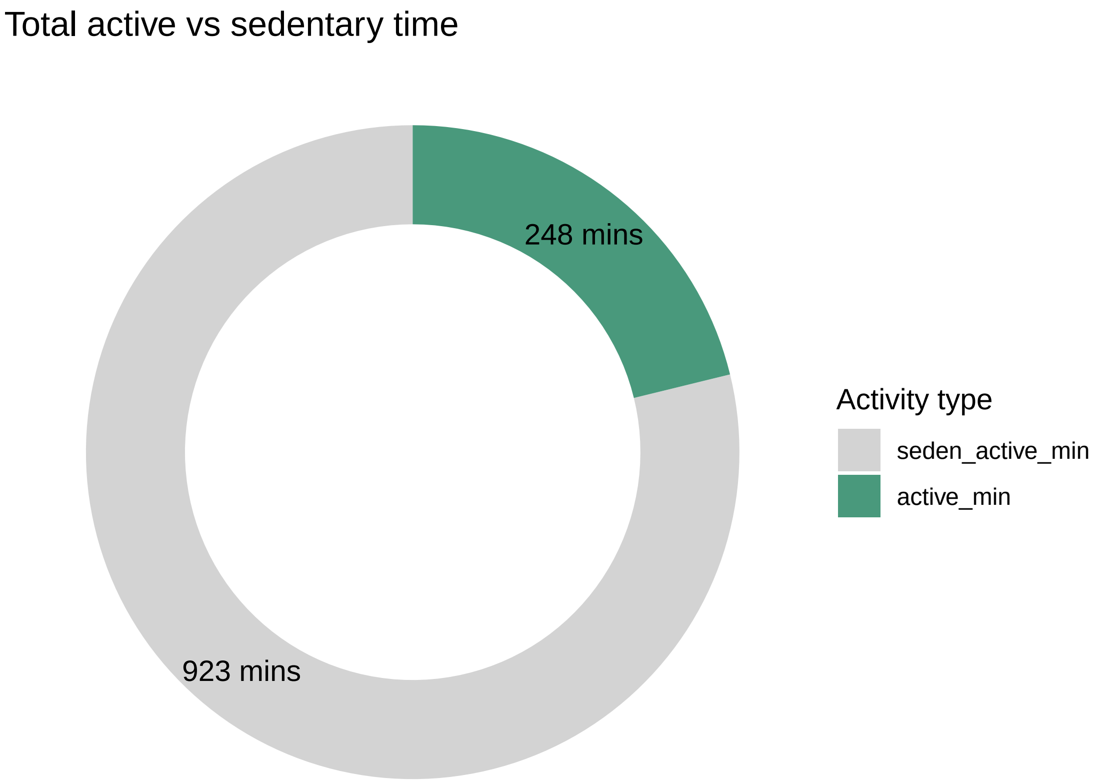
- In the weekend, the starting time, however, is later at around 8:00.

Total active vs sedentary time

Note that active time equals to the total of light, moderate and very active time

```
daily_activity %>%
  filter(
    group == 2
  )%>%
  summarize(
    very_active_min = mean(very_active_min),
    moderate_active_min = mean(moderate_active_min),
    light_active_min = mean(light_active_min),
    seden_active_min = mean(seden_active_min)
  )%>%
  summarize(
    very_active_min = mean(very_active_min),
    moderate_active_min = mean(moderate_active_min),
    light_active_min = mean(light_active_min),
    seden_active_min = mean(seden_active_min)
  )%>%
  mutate(
    active_min = sum(very_active_min, moderate_active_min, light_active_min)
  )%>%
  select(
    -c(very_active_min,moderate_active_min,light_active_min)
  )%>%
  pivot_longer(
    cols = everything(),
    names_to = "activity_type",
    values_to = "total_time"
  )%>%
  mutate(
    activity_type = factor(
      activity_type,
      levels = c("seden_active_min","active_min")
    ),
    total_time = round(total_time, 0)
  )%>%
  ggplot(
    aes(
      x = 3,           # x = 3 = hole size
      y = total_time,
      fill = activity_type
    )
  )+
  geom_bar(
    width = 1,
    stat = "identity"
  )+
  coord_polar(
    theta = "y"
  )+
  xlim(
    c(0.2, 3 + 0.5)    # "3" is hole size
  )+
```

```r
geom_text(
  aes(
    label = paste0(total_time, " mins")
  ),
  position = position_stack(vjust = 0.5),
  size = 4,
  show.legend = FALSE
)+
scale_fill_manual(
  values =  c("lightgrey","#49997c")
)+
labs(
  x = "",
  y = "",
  fill = "Activity type",
  title = "Total active vs sedentary time"
)+
theme_void()
```



- Users were active for 248 mins ~ 4.13 hours a day. This is almost doubled group 1 users.

Average total device using days in the whole period

```r
daily_activity %>%
  filter(
    group == 2
  )%>%
  summarize(
    total_using_day = mean(total_using_day)
  )%>%
  summarize(
    avg_using_day = mean(total_using_day)
  )
```

```
## # A tibble: 1 x 1
##   avg_using_day
##           <dbl>
## 1          40.4
```

The group has an average total device using day of 40.4, which can be considered as ‘Often’.

Distance travelled

```r
daily_activity %>%
  filter(
    group == 2
  )%>%
  summarize(
    very_active_dist = mean(very_active_dist),
    moderate_active_dist = mean(moderate_active_dist),
    light_active_dist = mean(light_active_dist),
    seden_active_dist = mean(seden_active_dist)
  )%>%
  summarize(
    very_active_dist = mean(very_active_dist),
    moderate_active_dist = mean(moderate_active_dist),
    light_active_dist = mean(light_active_dist),
    seden_active_dist = mean(seden_active_dist)
  )%>%
  mutate(
    active_dist = sum(very_active_dist, moderate_active_dist, light_active_dist)
  )%>%
  select(
    -c(very_active_dist,moderate_active_dist,light_active_dist)
  )%>%
  pivot_longer(
    cols = everything(),
    names_to = "activity_type",
    values_to = "total_time"
  )%>%
  mutate(
    activity_type = factor(
      activity_type,
      levels = c("seden_active_dist","active_dist")
    ),
    total_time = round(total_time, 2)
  )
```

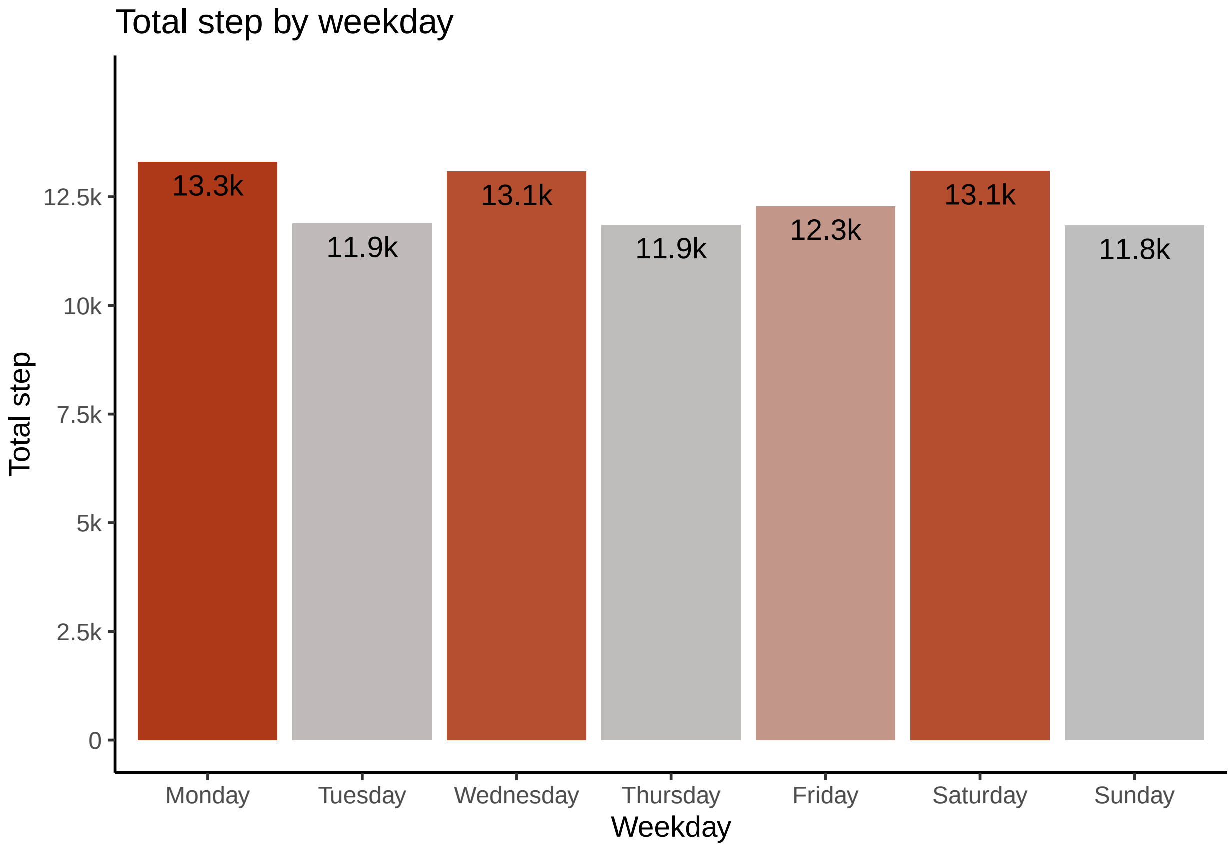
```
## # A tibble: 2 x 2
##   activity_type     total_time
##   <fct>                  <dbl>
## 1 seden_active_dist       0
## 2 active_dist             5.42
```

User group 2 has an average active distance of 5.42km. Notice that this is *2.52* times more than the distance of user group 1 (2.15km).

Group 3: Take more than 10,000 steps on daily average

Average steps by weekday

```r
daily_activity %>%
  filter(
    group == 3
  )%>%
  group_by(
    weekday
  )%>%
  summarize(
    total_step = mean(total_step)
  )%>%
  ggplot(
    aes(
      x = weekday,
      y = total_step,
      fill = total_step
    )
  )+
  geom_bar(
    stat = "identity",
    show.legend = FALSE
  )+
  scale_y_continuous(
    limits = c(0, 15000),
    breaks = c(0, 2500, 5000, 7500, 10000, 12500),
    labels = c(0, "2.5k","5k","7.5k","10k","12.5k")
  )+
  geom_text_repel(
    aes(
      label = paste0(round(total_step/1000,1), "k")
    ),
    vjust = 1.6
  )+
  scale_fill_gradient(
    low = "grey",
    high = "#ae3918"
  )+
  labs(
    x = "Weekday",
    y = "Total step",
    title = "Total step by weekday"
  )+
  theme_classic()
```

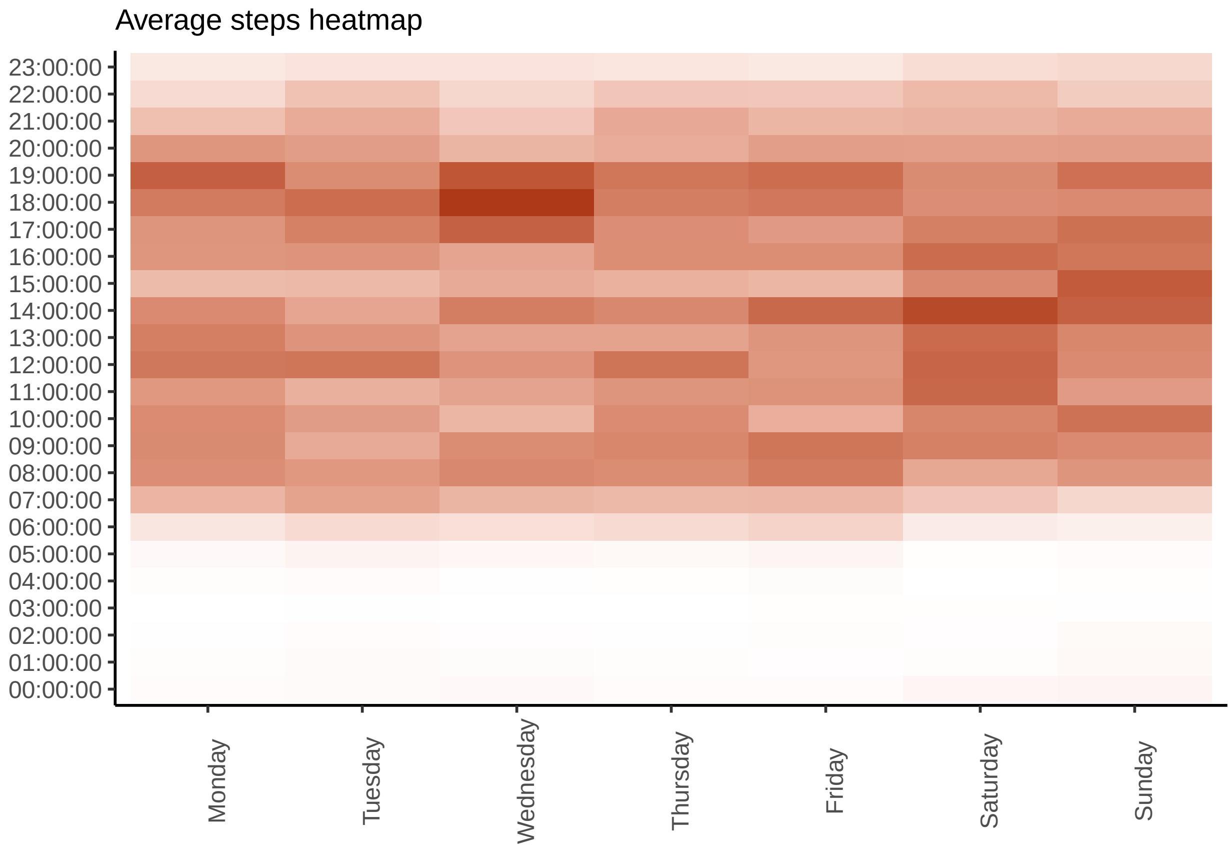


- Group 3 users has an extremely high average daily steps taken of least to be 11.8k and highest to be 13.3k.
- The stats is about 1.5 times bigger than group 2 and triple times bigger than group 1.

Step heatmap

```r
segment %>%
  filter(
    daily_avg_steps >= 10000
  )%>%
  group_by(weekday,hour) %>%
  summarize(total_step = mean(total_step))%>%
  ggplot(
    mapping = aes(
      x = weekday,
      y = hour
    )
  )+
  geom_tile(
    aes(fill= total_step),
    show.legend = FALSE
  )+
  scale_fill_gradient(
    low = "white",
    high = "#ae3918"
  )+
  labs(
    subtitle = "Average steps heatmap",
    x = NULL,
    y = NULL,
    fill = "Total step"
  )+
  theme_classic(
  )+
  theme(axis.text.x = element_text(angle = 90))
```

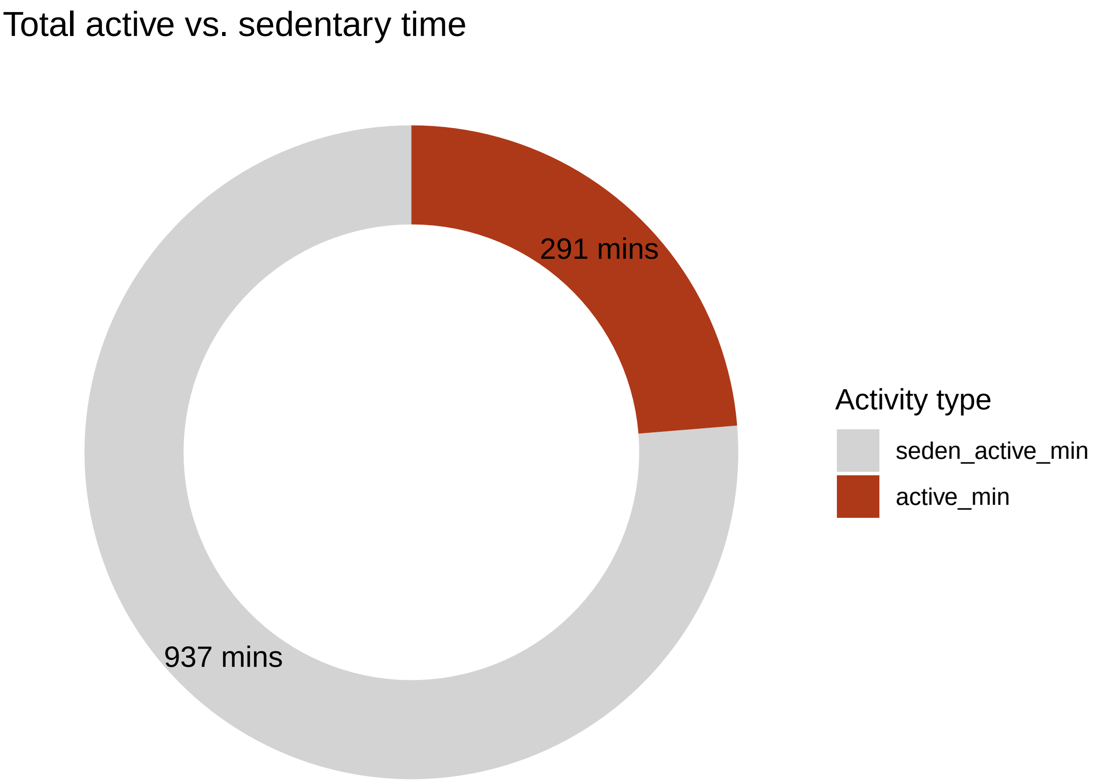
```
## `summarise()` has grouped output by 'weekday'. You can override using the
## `.groups` argument.
```



- A clear heat pattern from 6h(weekdays) and 7h(weekend) can be observed.
- Users tend to walk a lot on Saturday morning and from 18-19h weekdays.

Total active vs. sedentary time

```
daily_activity %>%
  filter(
    group == 3
  )%>%
  summarize(
    very_active_min = mean(very_active_min),
    moderate_active_min = mean(moderate_active_min),
    light_active_min = mean(light_active_min),
    seden_active_min = mean(seden_active_min)
  )%>%
  summarize(
    very_active_min = mean(very_active_min),
    moderate_active_min = mean(moderate_active_min),
    light_active_min = mean(light_active_min),
    seden_active_min = mean(seden_active_min)
  )%>%
  mutate(
    active_min = sum(very_active_min, moderate_active_min, light_active_min)
  )%>%
  select(
    -c(very_active_min,moderate_active_min,light_active_min)
  )%>%
  pivot_longer(
    cols = everything(),
    names_to = "activity_type",
    values_to = "total_time"
  )%>%
  mutate(
    activity_type = factor(
      activity_type,
      levels = c("seden_active_min","active_min")
    ),
    total_time = round(total_time, 0)
  )%>%
  ggplot(
    aes(
      x = 3,            # x = 3 = hole size
      y = total_time,
      fill = activity_type
    )
  )+
  geom_bar(
    width = 1,
    stat = "identity"
  )+
  coord_polar(
    theta = "y"
  )+
  xlim(
    c(0.2, 3 + 0.5)     # "3" is hole size
  )+
  geom_text(
    aes(
      label = paste0(total_time, " mins")
    ),
    position = position_stack(vjust = 0.5),
    size = 4,
    show.legend = FALSE
  )+
  scale_fill_manual(
    values =  c("lightgrey","#ae3918")
  )+
  labs(
    x = "",
    y = "",
    fill = "Activity type",
    title = "Total active vs. sedentary time"
  )+
  theme_void()
```



- Users stay active for an average of 291 mins a day ~ 4.85 hours daily.
- This is not significantly longer compared to group 2.

Average total device using days in the whole period

```
daily_activity %>%
  filter(
    group == 3
  )%>%
  summarize(
    total_using_day = mean(total_using_day)
  )%>%
  summarize(
    avg_using_day = mean(total_using_day)
  )
```

```
## # A tibble: 1 x 1
##   avg_using_day
##           <dbl>
## 1          42.3
```

The group has an average total device using days of 42.3, which can be considered as ‘Often’ (similar to the other 3 groups).

Distance travelled

```
daily_activity %>%
  filter(
    group == 3
  )%>%
  summarize(
    very_active_dist = mean(very_active_dist),
    moderate_active_dist = mean(moderate_active_dist),
    light_active_dist = mean(light_active_dist),
    seden_active_dist = mean(seden_active_dist)
  )%>%
  summarize(
    very_active_dist = mean(very_active_dist),
    moderate_active_dist = mean(moderate_active_dist),
    light_active_dist = mean(light_active_dist),
    seden_active_dist = mean(seden_active_dist)
  )%>%
  mutate(
    active_dist = sum(very_active_dist, moderate_active_dist, light_active_dist)
  )%>%
  select(
    -c(very_active_dist,moderate_active_dist,light_active_dist)
  )%>%
  pivot_longer(
    cols = everything(),
    names_to = "activity_type",
    values_to = "total_time"
  )%>%
  mutate(
    activity_type = factor(
      activity_type,
      levels = c("seden_active_dist","active_dist")
    ),
    total_time = round(total_time, 2)
  )
```

```
## # A tibble: 2 x 2
##   activity_type     total_time
##   <fct>                  <dbl>
## 1 seden_active_dist       0
## 2 active_dist             9.06
```

User group 3 has an average active distance of 9.06km. This is a significantly larger number compared to group 1 and group 2.

VII. Conclusion

1. Users - Device usage frequency: By tracking the device total using days in the periods, most users are considered as ‘Often’ users. - 28 people (80%) are ‘Often’ users - Even based on different steps taking user groups, the average device total using days is still in the ‘Often’ frequency range.

- User grouping: Most users take less than 10,000 steps daily (from around 4k to 7k steps daily). This shows that most users are normal adults who don’t have intesive working/training schedule.
  - Group 1: Users were active in an disordered schedule. Have least walking steps, travel distance and active time. Adults who are not very active and are not exercising frequently.

  - Group 2: Users were active in a more ordered schedule. Have a decent number of steps daily at around. Adults but tends to workout frequently and more active compared to G1.

  - Group 3: Most active users. Possesses extremely intensive activity stats. Adults who always walk/exercising.

2. Daily active time distribution - People spend most of their time (82%) sedentary, 18% active.
- In active time, most of the time are for light activities (84% of active time). - High intensity % calories burnt observed during 17-19h on weekdays, this suggests that users usually workout during this time period. Similar insights can be found on Saturday morning.
- Significantly high steps count are found on weekends and on meal time.

4. Distribution in a week - Similar activities in weekdays
- Usually higher activities on Saturday
- Less active on Sunday

5. Sleep - People sleep more on weekends.
- 9% of the time in bed are not for sleep, this suggest that they may do other activities such as watching TV, using mobile phones, reading, . . .

VIII. Suggestions

• Provide new tracking modes for different types of activities.
• High amount of time not wearing the tracker shows that users may not feel comfortable with the design and consider.
• Create different user profiles, features for different user groups from less active to very active users.
• Aside from sedentary activities, create new feature for light activities as it takes 84% of all the average activities in a day.
• Create a guide or an auto-bot to as an assistant to help reminding, scheduling and reporting.
• Provide health reports.
• Day-end report: people usually spend some time to relax, restless before sleep, so we can provide a report of their health status during the day.
• Week-end report on Sunday when people usually spend their time to rest, be less active and may want to take a look of what had happened through out the week.
• Encourage users to be more active by adding:
• Reward system: game or an experience count system with level-up mechanism.
• Notifications: regular reminds customers to encourage them to be more active.