Case Study: Fitbit analysis

Mason Phung

```
library(tidyverse) # main function
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate) # data format
library(gridExtra) # grid.arrange() to print many plots together in a same page
```

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

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(ggrepel) # For displaying plot's labels outside of the chart
library(wesanderson) # Wes Anderson color palette for plots
library(scales) # For percent()
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

Load datasets

Clean data

The data is not yet cleaned, we will observe the datasets, determine problems 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.

```
hourly_activity <- raw_hourly_steps %>%
  left_join(raw_hourly_calories, by = c("Id","ActivityHour")) %>%
  left_join(raw_hourly_intensities, by = c("Id","ActivityHour"))

hourly_activity <- hourly_activity %>%
  mutate(ActivityHour = mdy_hms(ActivityHour))

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

3. 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)

hourly_activity <- hourly_activity %>%
  mutate(
    Hour = ifelse(is.na(Hour), "00:00:00",Hour),
    ActivityDate = ymd(ActivityDate)
  )
```

4. Check for NAs and duplicates

Take a look at the duplicates

```
sum(duplicated(raw_daily_activity))
```

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

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

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

```
sum(duplicated(hourly_activity))
```

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

```
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's duplicates are incorrect, therefore, we only remove the detected duplication in the hourly_activity dataset.

Remove duplications

```
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.

5. 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 important.

```
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')
    )
  )
```

6. Remove unnecessary variables

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

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

6. Clean variable name

```
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
  )
```

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 (kilocalories)
sleep_record	Number of minutes 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

/newpage # Summarize data statistics

```
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
```

```
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.

Analysis - data summary stats

Take a look at the means of the variable

```
options(scipen = 999)

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

```
##   total_step total_dist very_active_dist moderate_active_dist light_active_dist
## 1   7280.898   5.219434         1.397416             0.538461          3.193407
##   seden_active_dist very_active_min moderate_active_min light_active_min
## 1       0.001703651        19.67931            13.40301         185.3729
##   seden_active_min calories
## 1         992.5426 2266.266
```

- The average step taken per day is 7638 steps and the calories burned per day on average is 2304 kcal, which are considered to be high for an adult. These number suggest that the audience can be office workers who care for walk/exercise after work.

- The data also shows a high amount of distance and time spent for **light activity**, which can be a point for R&D dept to focus on and develop future features to fit this tendency of usage.

Analysis - Usage time and using habits

1. Daily activity

Determine using frequency

```
daily_activity <- daily_activity %>%
  group_by(id) %>%
  mutate(
    total_using_day = sum(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"
      )
    )
```

```
)
```

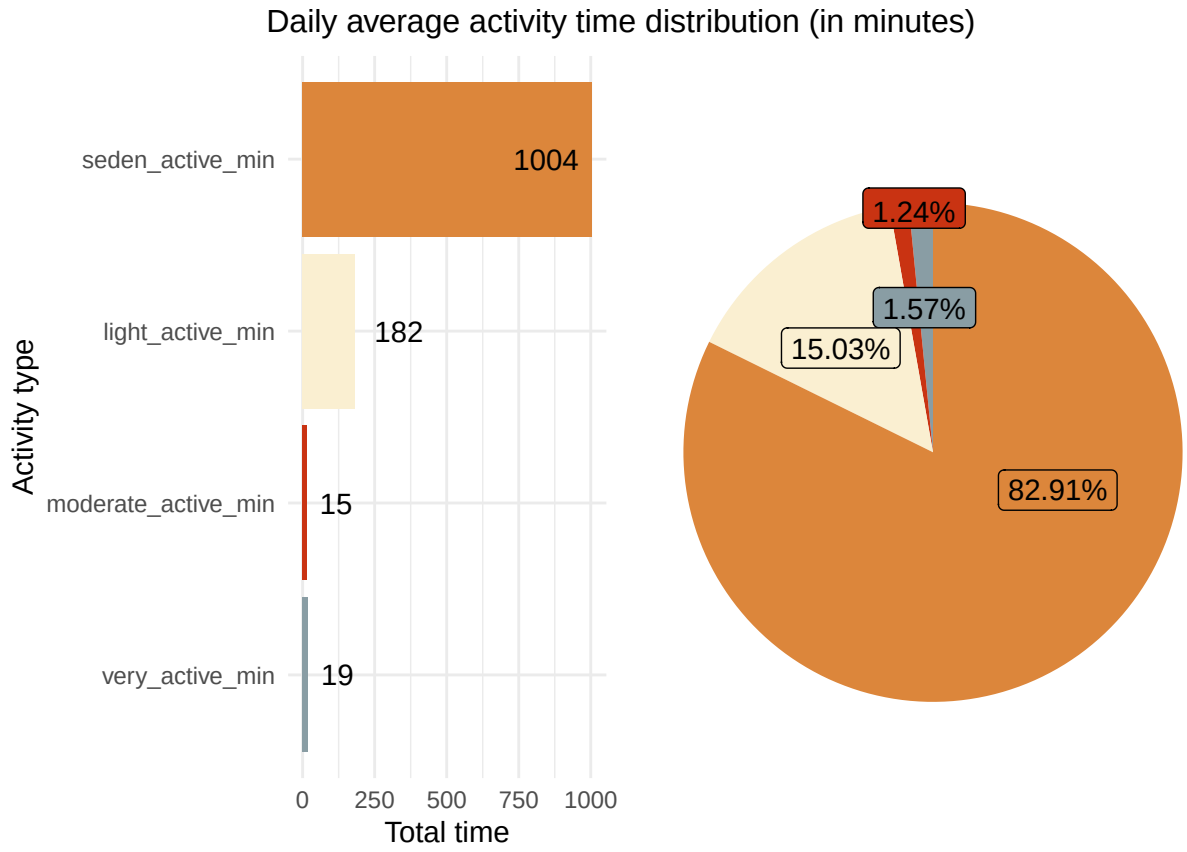
Using frequency - Rarely use users (2 people, 1-30% of 62 days)
- Sometimes use users (2 people, 30-49% of 62 days)
- Often use users (28 people, 50-79% of 62 days)
- Usually use users (2 people, 80-99% of 62 days)
- Always use users (1 people, 100% of 62 days)

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, 82.9% of the time was spent in sedentary while people only active in 17.1% daily.
- When in active, only 19 minutes of the day are used for very active activities, 15 for moderate activities while light active activities take 182 mins (3 hours 2 mins).

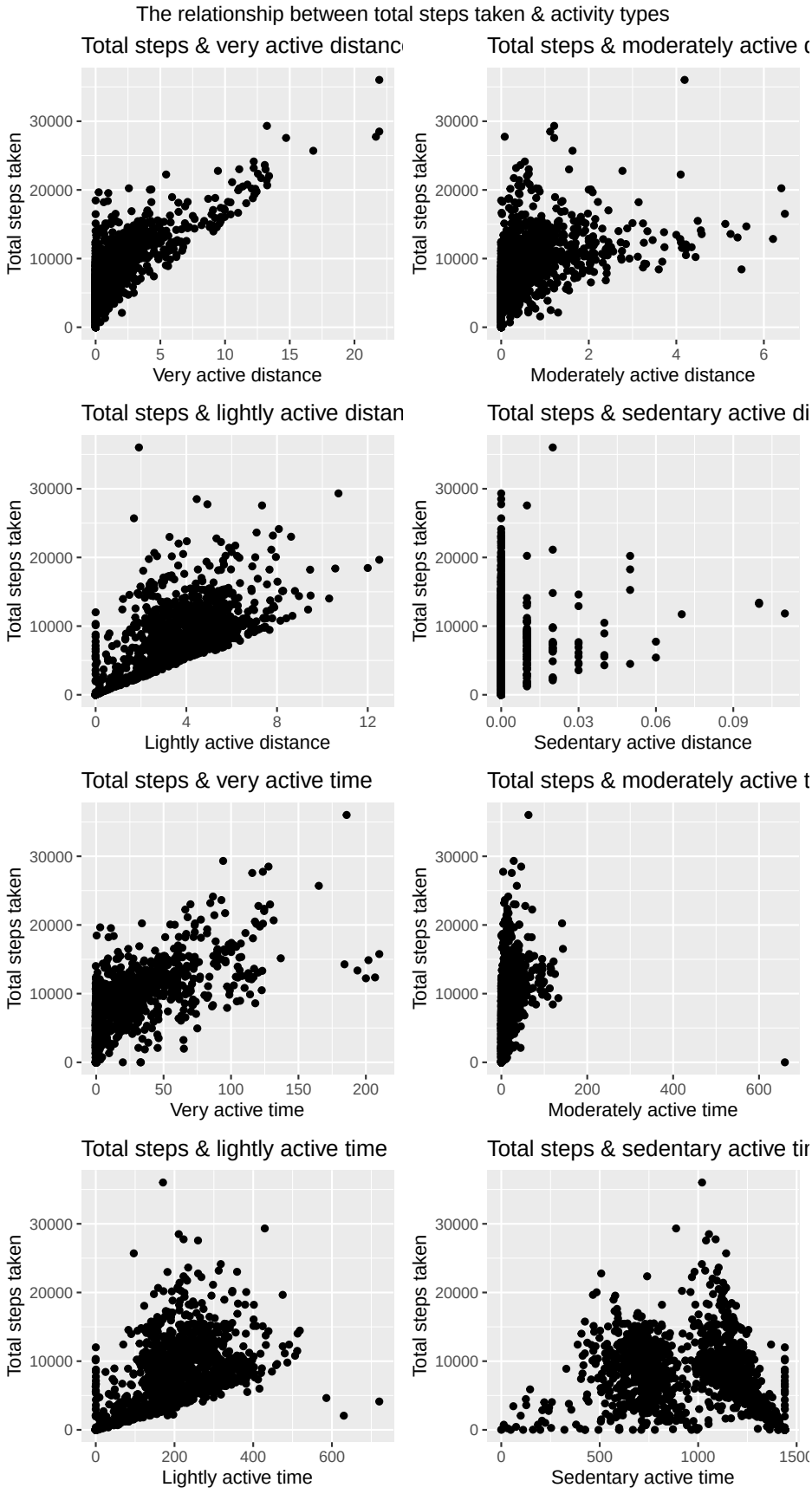
The tracker's main feature is to measure the ***total steps taken***, therefore we can assume that their is always a strong positive relationship between ***total steps taken*** and the ***total distance travelled/ total calories burned***.

Correlation: Relationship between total steps taken and active type

```
# Total steps vs very active distance
g1 <- daily_activity %>%
  ggplot(
    aes(
      x = very_active_dist,
      y = total_step,
    )
  )+
  geom_jitter(
  )+
  labs(
    x = "Very active distance",
    y = "Total steps taken",
    title = "Total steps & very active distance"
  )


# Total steps vs moderate active distance
g2 <- daily_activity %>%
  ggplot(
    aes(
      x = moderate_active_dist,
      y = total_step,
    )
  )+
  geom_jitter(
  )+
  labs(
    x = "Moderately active distance",
    y = "Total steps taken",
    title = "Total steps & moderately active distance"
  )

# Total steps vs light active distance
g3 <- daily_activity %>%
  ggplot(
    aes(
      x = light_active_dist,
      y = total_step,
    )
  )+
  geom_jitter(
  )+
  labs(
    x = "Lightly active distance",
    y = "Total steps taken",
    title = "Total steps & lightly active distance"
  )

# Total steps vs sedentary active distance
g4 <- daily_activity %>%
  ggplot(
```

```
    aes(
      x = seden_active_dist,
      y = total_step,
    )
  )+
  geom_jitter(
  )+
  labs(
    x = "Sedentary active distance",
    y = "Total steps taken",
    title = "Total steps & sedentary active distance"
  )


# Total steps vs very active time
g5 <- daily_activity %>%
  ggplot(
    aes(
      x = very_active_min,
      y = total_step,
    )
  )+
  geom_jitter(
  )+
  labs(
    x = "Very active time",
    y = "Total steps taken",
    title = "Total steps & very active time"
  )

# Total steps vs moderate active time
g6 <- daily_activity %>%
  ggplot(
    aes(
      x = moderate_active_min,
      y = total_step,
    )
  )+
  geom_jitter(
  )+
  labs(
    x = "Moderately active time",
    y = "Total steps taken",
    title = "Total steps & moderately active time"
  )

# Total steps vs light active time
g7 <- daily_activity %>%
  ggplot(
    aes(
      x = light_active_min,
      y = total_step,
    )
```

```
  )+
  geom_jitter(
  )+
  labs(
    x = "Lightly active time",
    y = "Total steps taken",
    title = "Total steps & lightly active time"
  )


# Total steps vs sedentary active time
g8 <- daily_activity %>%
  ggplot(
    aes(
      x = seden_active_min,
      y = total_step,
    )
  )+
  geom_jitter(
  )+
  labs(
    x = "Sedentary active time",
    y = "Total steps taken",
    title = "Total steps & sedentary active time"
  )

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
Total steps & very active distanc
Total steps & moderately active d
Total steps & lightly active distan
Total steps & sedentary active di
Total steps & very active time
Total steps & moderately active t
Total steps & lightly active time
Total steps & sedentary active tir
Total steps taken
Very active distance
Moderately active distance
Lightly active distance
Sedentary active distance
Very active time
Moderately active time
Lightly active time
Sedentary active time

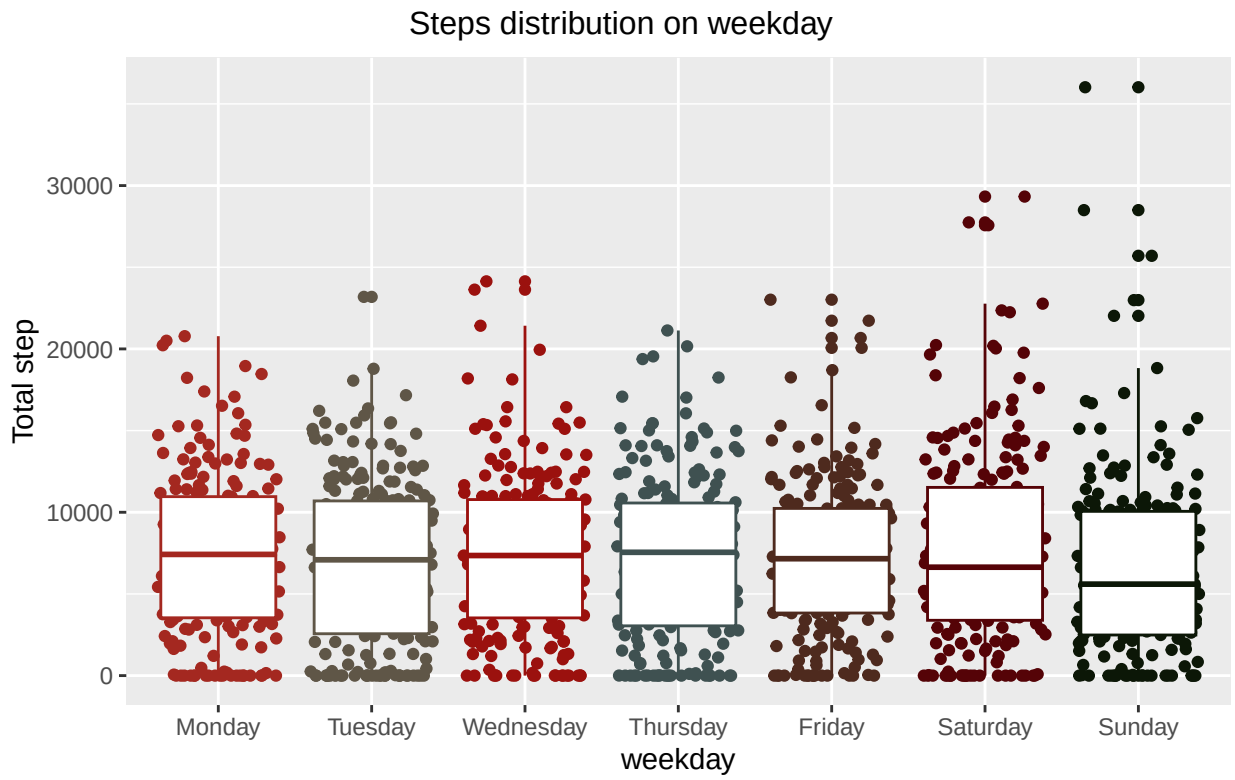
From the plot, we can clearly see that there is:
- A strong relationship between total steps taken and light active distance/time
- A insignificant relationshop between total steps taken and moderate/very active distance/time

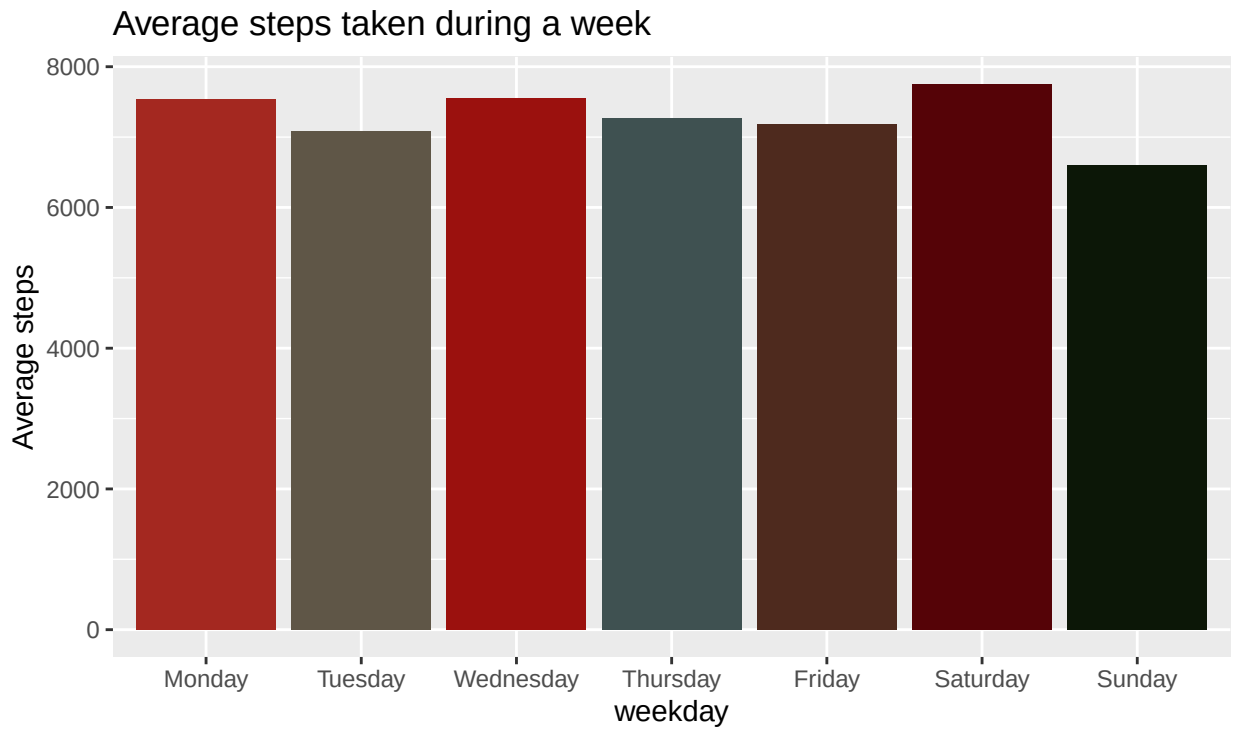
The plots show that most of the customers spend time walk lightly everyday and most of their steps are taken in low intensity. This information suggests that the customers mainly are normal people/workers. Moreover, the relationship between total steps taken and moderate/very active distance proves that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

Steps by weekday

```
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(
    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 during a week",
    y = "Average steps"
  )+
  scale_fill_manual(
    values = wes_palette(
      name = "BottleRocket1"
    )
  )

grid.arrange(
  h1, h2,
```

```
  nrow = 2,
  top = "Steps distribution on weekday"
)
```





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

Take a look at the means of the variable

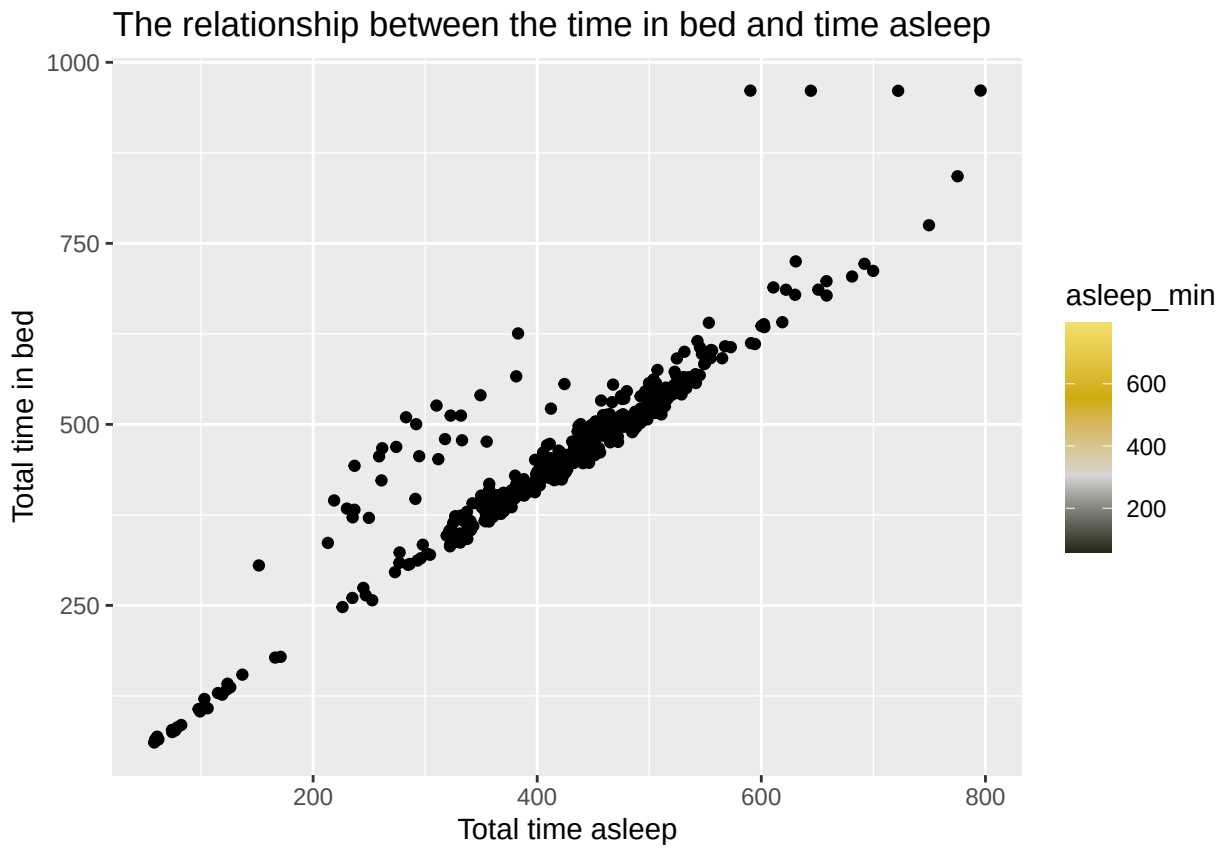
```
options(scipen = 999)

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.

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

```
daily_sleep %>%
  ggplot(
    aes(
      x = asleep_min,
      y = in_bed_min,
      fill = asleep_min
    )
  )+
  geom_jitter(
  )+
  scale_fill_gradientn(
    colours = rev(wes_palette(
      name = "Moonrise1",
      type = "continuous"
    ))
  )+
  labs(
    x = "Total time asleep",
    y = "Total time in bed",
    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.

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(),
    show.legend = FALSE
  )+
  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
    )
  )+
  labs(
    title = "Daily average amount of time asleep and in bed",
    subtitle = "In a week",
    x = "Weekday",
    y = "Time(minute)"
  )+
  theme_minimal()


daily_sleep_summary <- daily_sleep %>%
```

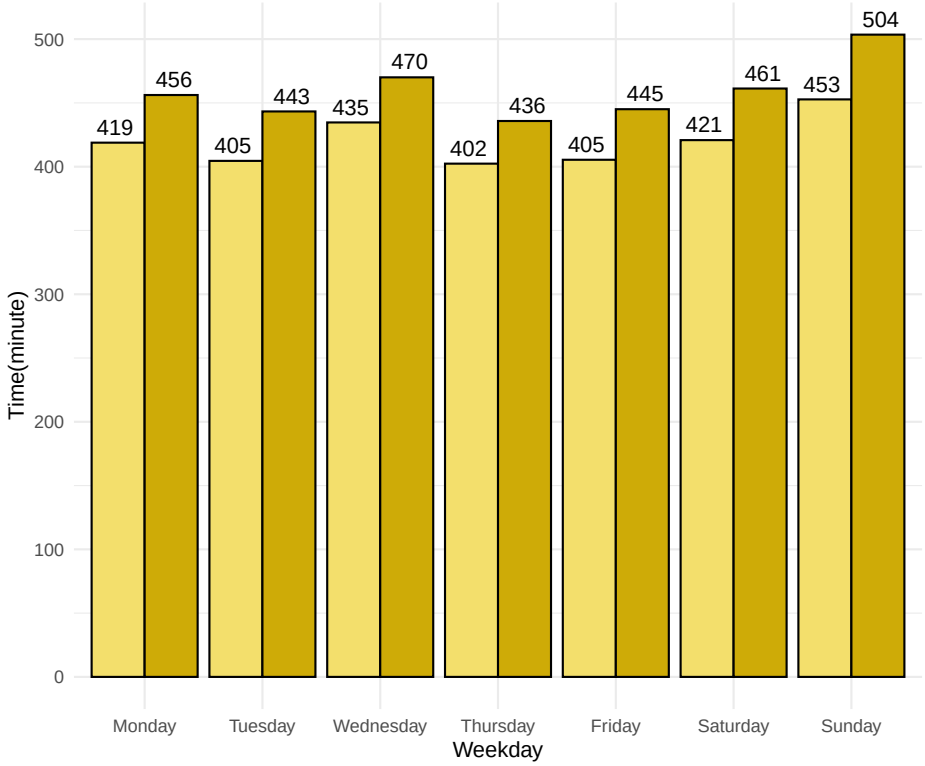
```
  summarize(
    asleep_min = mean(asleep_min),
    in_bed_min = mean(in_bed_min)
  )%>%
  mutate(
    not_asleep = in_bed_min - asleep_min
  )


f2 <- data.frame(
  label = c("In Bed", "Asleep"),
  value = c(daily_sleep_summary$in_bed_min, daily_sleep_summary$not_asleep)
  )%>%
  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 = "Moonrise1"
    )
  )+
  geom_label(
    aes(
      label = percent(value / sum(value)),
      y = value
    ),
    hjust = c(-0.5,2.4),
    vjust = c(7.5,0),
    color = "black",
    size = 4
  )+
  theme_void()


grid.arrange(
  f1,f2,
  nrow = 2,
```

```
  top = "Average sleep time distribution"
)
```

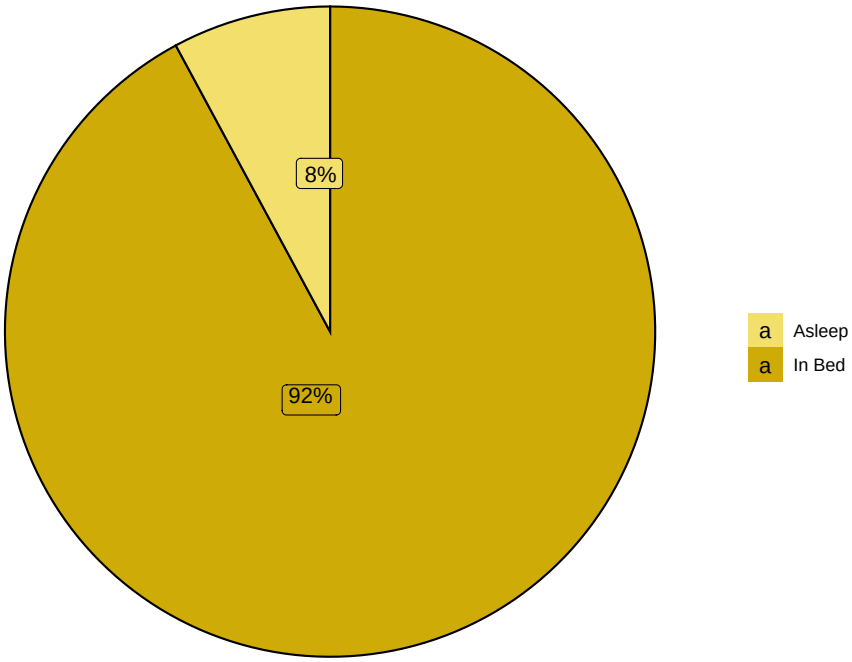
Average sleep time distribution

Daily average amount of time asleep and in bed

In a week



Percentage of asleep time relative to in bed time



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 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(
    subtitle = "In a day",
    x = "Time",
    y = "Steps",
    fill = "Calories"
  )+
  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(
    subtitle = "In a week",
    x = "Weekday",
    y = "Time",
    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.
```

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(
    subtitle = "In a day",
    x = "Time",
    y = "Calories",
    fill = "Calories"
  )+
  scale_fill_gradientn(
    colors = wes_palette(
      name = "Zissou1",
      n = 5
    )
  )+
  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 = wes_palette(
      name = "Zissou1",
      n = 5
```

```
    )
  )+
  labs(
    subtitle = "In a week",
    x = "Weekday",
    y = "Time",
    fill = "Calories"
  )+
  theme_classic(
  )+
  theme(axis.text.x = element_text(angle = 90))
```

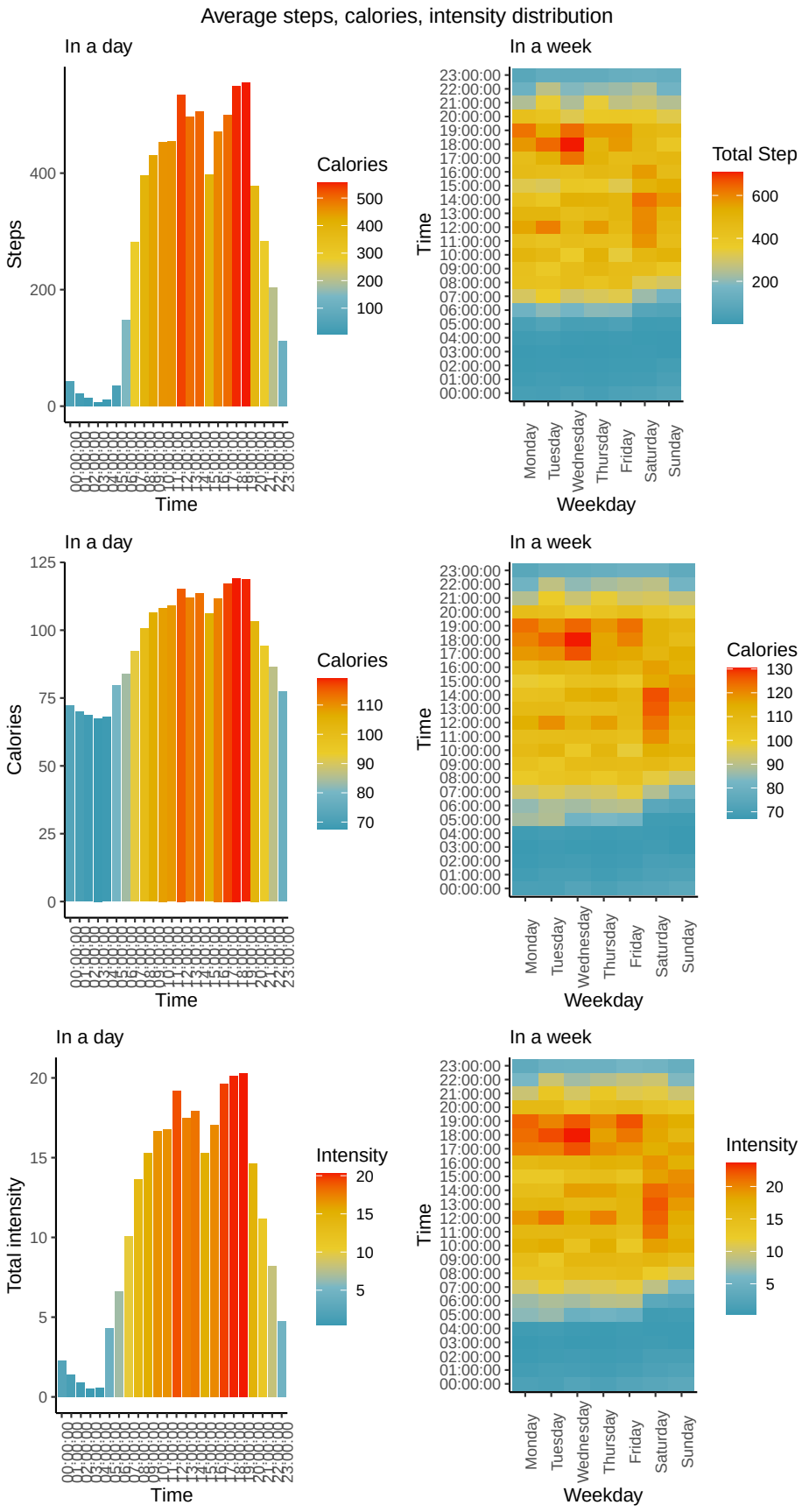
```
## `summarise()` has grouped output by 'weekday'. You can override using the
## `.groups` argument.
```

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(
    subtitle = "In a day",
    x = "Time",
    y = "Total intensity",
    fill = "Intensity"
  )+
  scale_fill_gradientn(
    colors = wes_palette(
      name = "Zissou1",
      n = 5
    )
  )+
  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 = "Zissou1",
      n = 5
    )
  )+
  labs(
    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, calories, intensity distribution
In a day
In a week
Steps
Time
Calories
Total Step
Weekday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Calories
Intensity
Total intensity

- 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

Analysis - by dividing users into groups

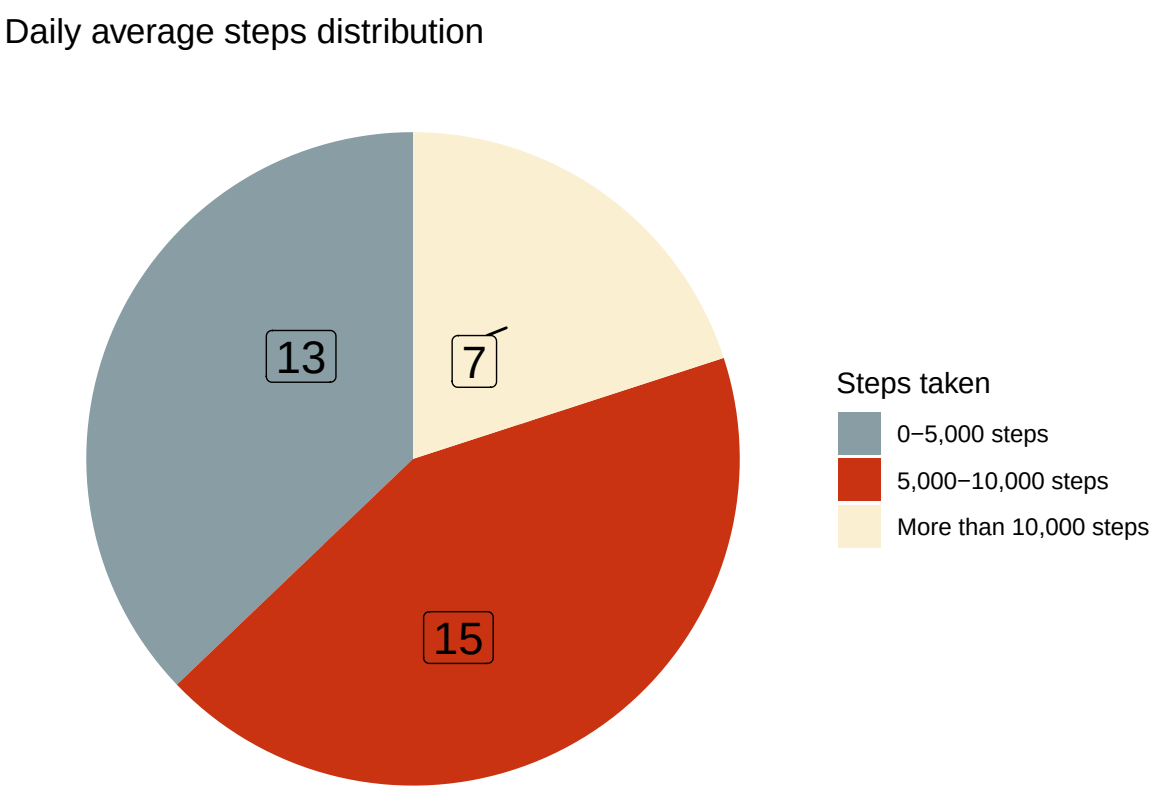
Adding user segmentation by steps and using frequency

```
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 = "",
      y = count/35,
      fill = group
    )
  )+
  geom_bar(
    stat = "identity",
```

```
  width = 1
)+
coord_polar(
  "y"
)+
scale_fill_manual(
  values = wes_palette(
  n = 3,
  name = "Royal1"
  )
)+
geom_label_repel(
  aes(
    label = count,
  ),
  position = position_stack(vjust = 0.5),
  size = 6,
  show.legend = FALSE
)+
labs(
  title = "Daily average steps distribution",
  x = "",
  y = "",
  fill = "Steps taken"
)+
theme_void()
```

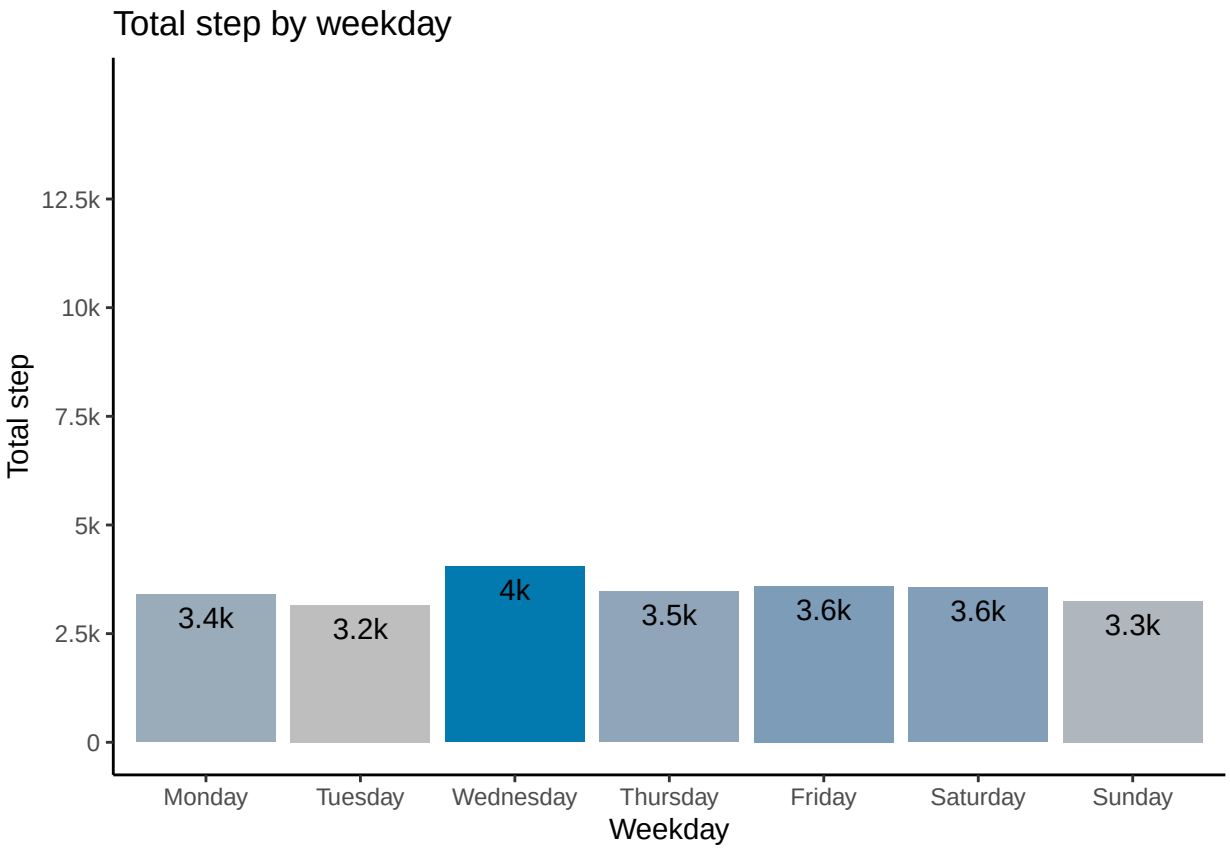


Based on the visualization, we can observe that there is only 7 people (23,3%) possess an average daily step of more than 10,000 while 28 people have less than 10,000 steps a day.

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

Average steps by weekday

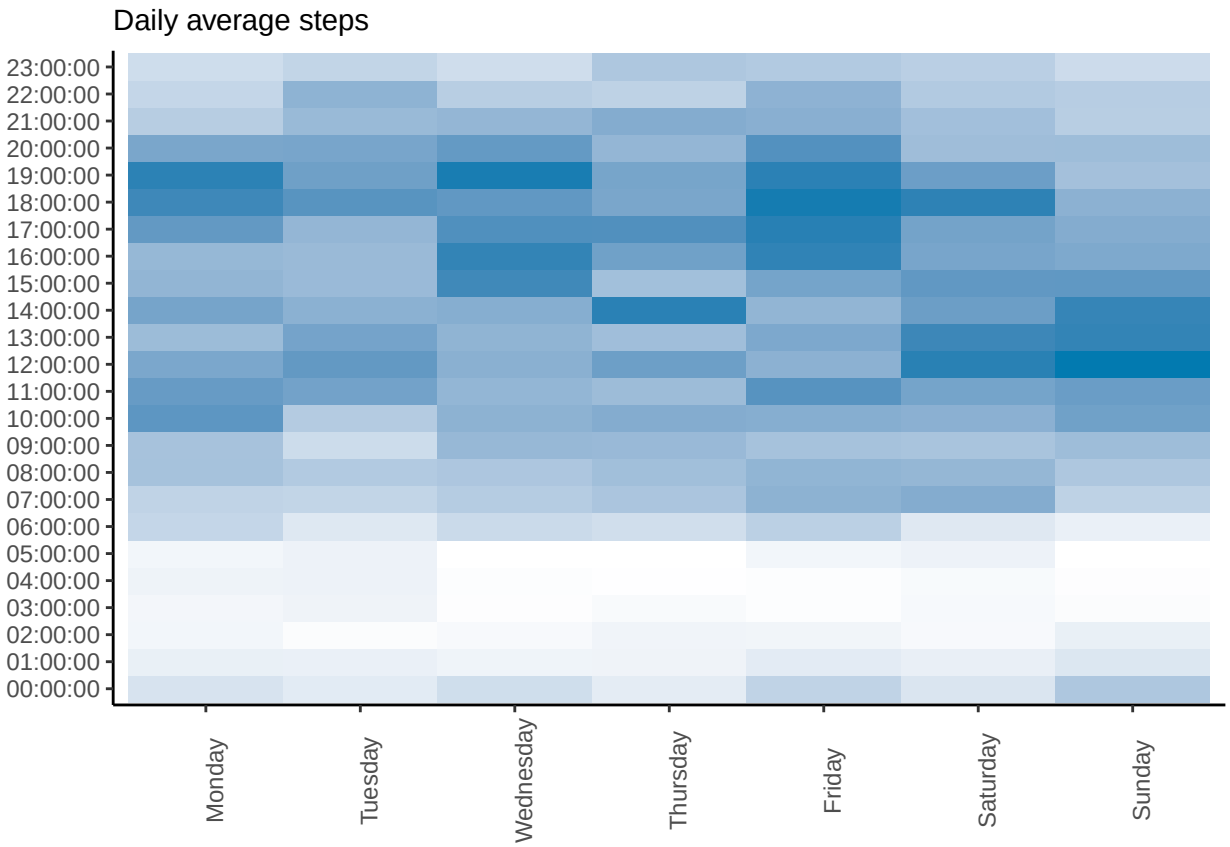
```
daily_activity %>%
  filter(
    group == 1
  )%>%
  group_by(
    weekday
  )%>%
  summarize(
    total_step = mean(total_step)
  )%>%
  ggplot(
    aes(
      x = weekday,
      y = total_step,
      fill = total_step
    )
  )+
  geom_bar(
    stat = "identity",
    show.legend = FALSE
  )+
  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 = "#027ab0"
  )+
  labs(
    x = "Weekday",
    y = "Total step",
    title = "Total step by weekday"
  )+
  theme_classic()
```


Total step by weekday
Total step
12.5k
10k
7.5k
5k
2.5k
0
3.4k
3.2k
4k
3.5k
3.6k
3.6k
3.3k
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Weekday

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 = "#027ab0"
  )+
  labs(
    subtitle = "Daily average steps",
    x = NULL,
    y = NULL,
    fill = "Total step"
  )+
  theme_classic(
  )+
  theme(axis.text.x = element_text(angle = 90))
```

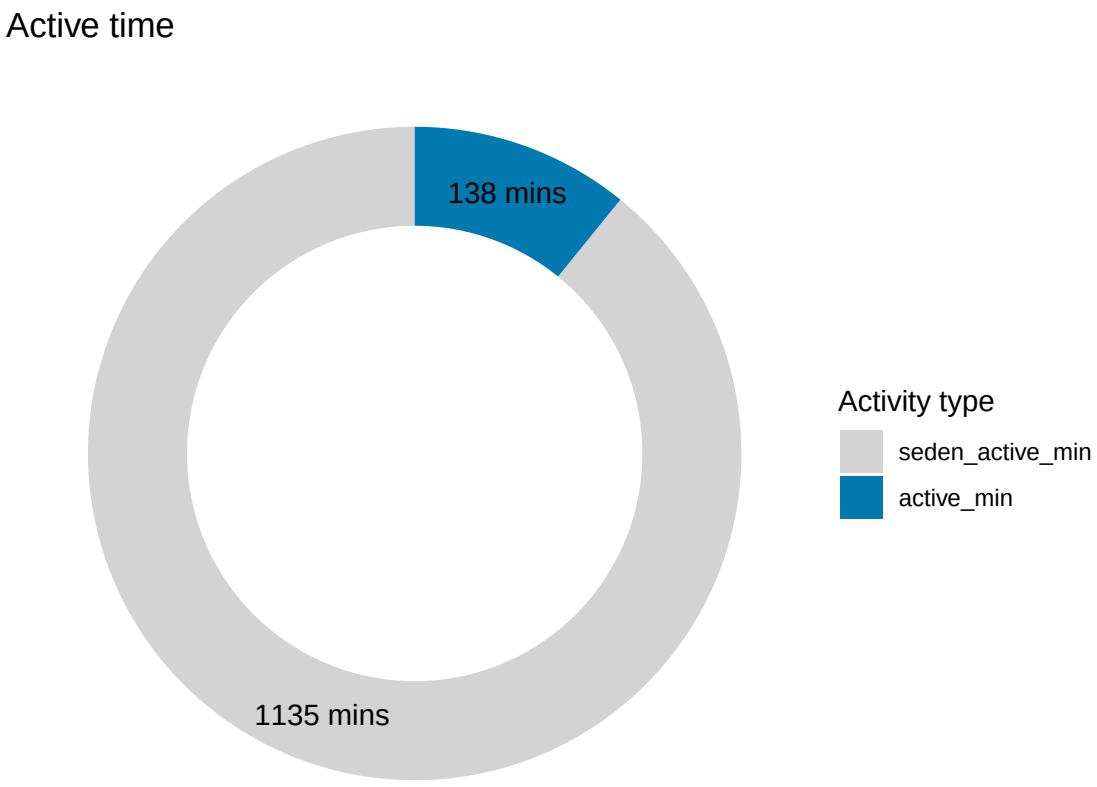
```
## `summarise()` has grouped output by 'weekday'. You can override using the
## `.groups` argument.
```


Daily average steps
23:00:00
22:00:00
21:00:00
20:00:00
19:00:00
18:00:00
17:00:00
16:00:00
15:00:00
14:00:00
13:00:00
12:00:00
11:00:00
10:00:00
09:00:00
08:00:00
07:00:00
06:00:00
05:00:00
04:00:00
03:00:00
02:00:00
01:00:00
00:00:00
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

Activity type

```
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","#027ab0")
)+
labs(
  x = "",
  y = "",
  fill = "Activity type",
  title = "Active time"
)+
theme_void()
```



Using frequency

```
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
```

Distance travelled

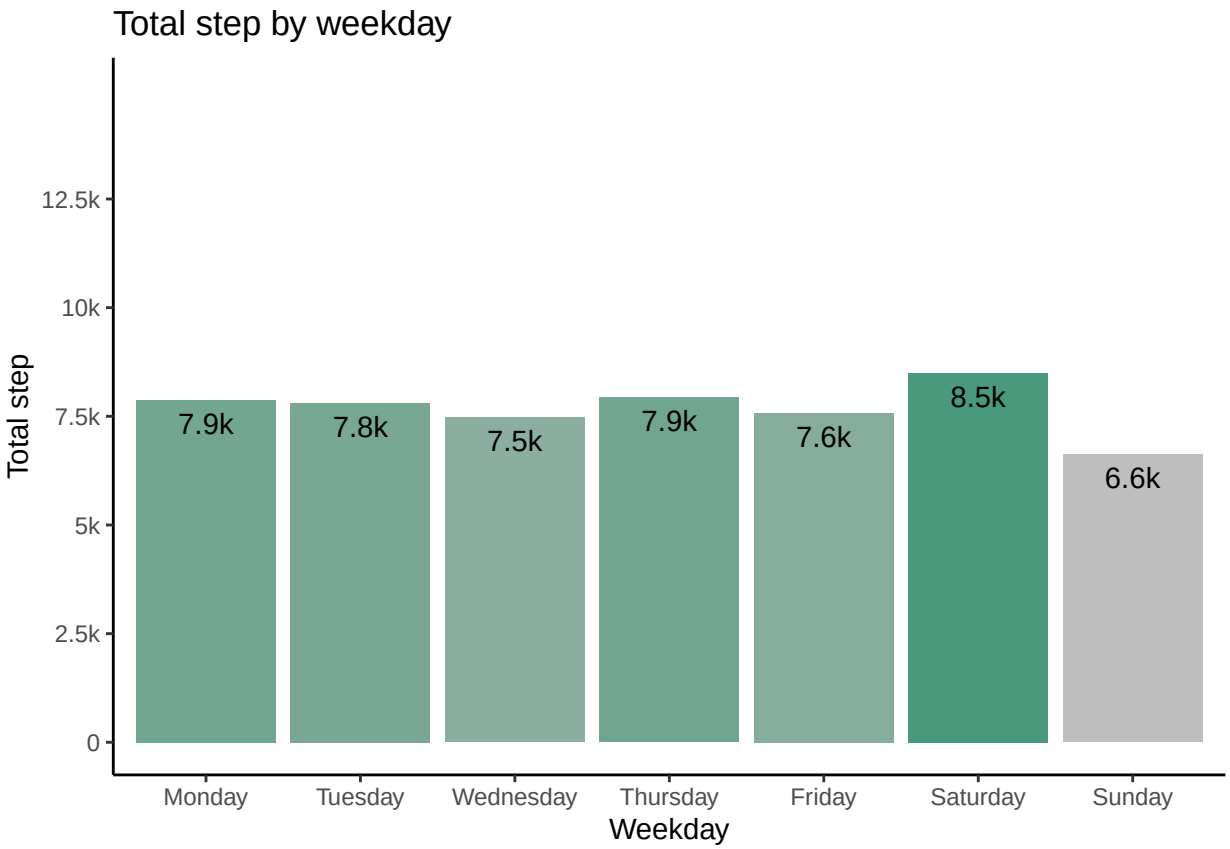
```
daily_activity %>%
  filter(
    group == 1
  )%>%
  summarize(
    very_active_dist = mean(very_active_dist),
    moderate_active_dist = mean(moderate_active_dist),
    light_active_dist = mean(light_active_dist),
    seden_active_dist = mean(seden_active_dist)
  )%>%
  summarize(
    very_active_dist = 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
```

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

Average steps by weekday

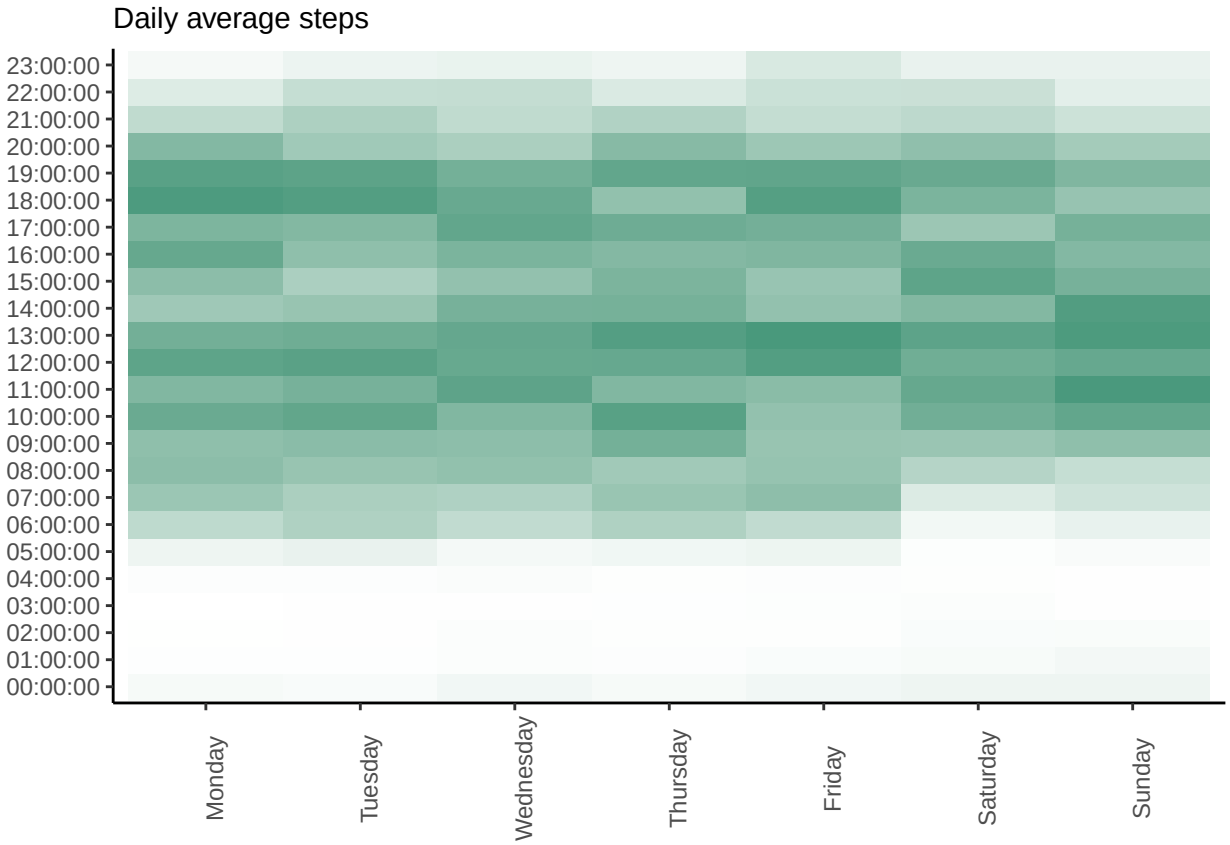
```
daily_activity %>%
  filter(
    group == 2
  )%>%
  group_by(
    weekday
  )%>%
  summarize(
    total_step = mean(total_step)
  )%>%
  ggplot(
    aes(
      x = weekday,
      y = total_step,
      fill = total_step
    )
  )+
  geom_bar(
    stat = "identity",
    show.legend = FALSE
  )+
  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()
```


Total step by weekday
Total step
12.5k
10k
7.5k
5k
2.5k
0
7.9k
7.8k
7.5k
7.9k
7.6k
8.5k
6.6k
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Weekday

Step heatmap

```
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 = "Daily average steps",
    x = NULL,
    y = NULL,
    fill = "Total step"
  )+
  theme_classic(
  )+
  theme(axis.text.x = element_text(angle = 90))
```

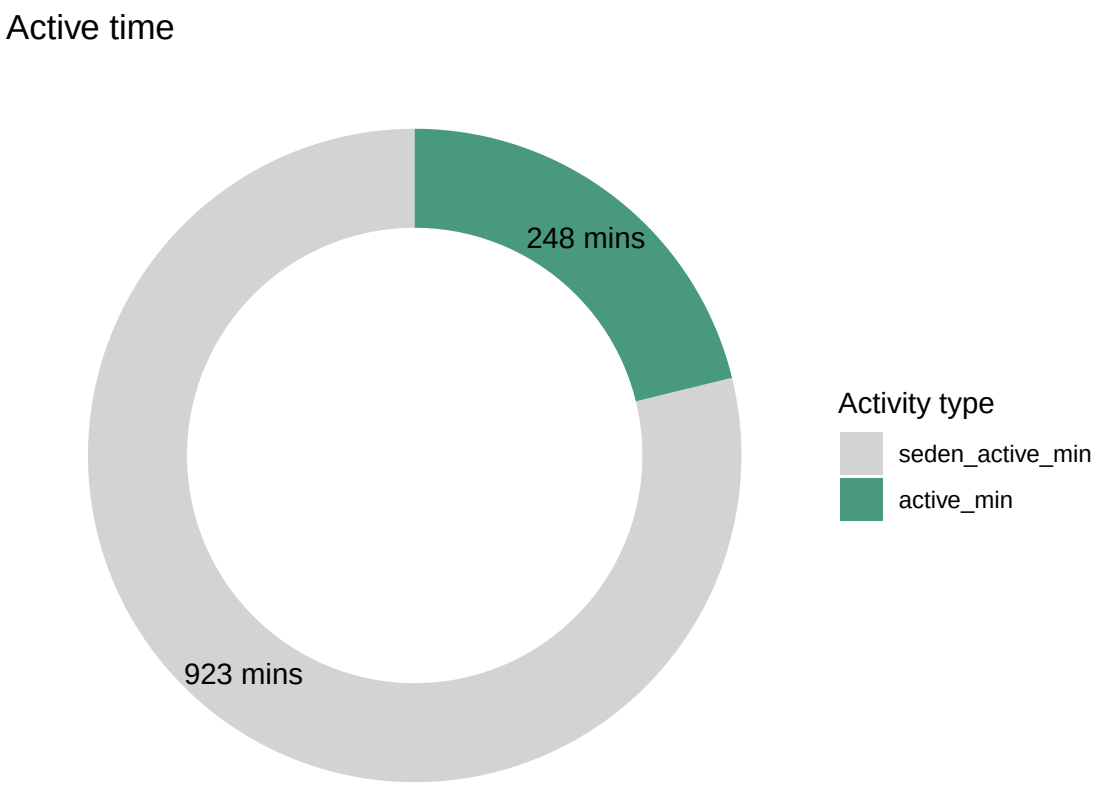
```
## `summarise()` has grouped output by 'weekday'. You can override using the
## `.groups` argument.
```


Daily average steps
23:00:00
22:00:00
21:00:00
20:00:00
19:00:00
18:00:00
17:00:00
16:00:00
15:00:00
14:00:00
13:00:00
12:00:00
11:00:00
10:00:00
09:00:00
08:00:00
07:00:00
06:00:00
05:00:00
04:00:00
03:00:00
02:00:00
01:00:00
00:00:00
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

Activity type

```
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
  )+
  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 = "Active time"
)+
theme_void()
```



Using frequency

```
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
```

Distance travelled

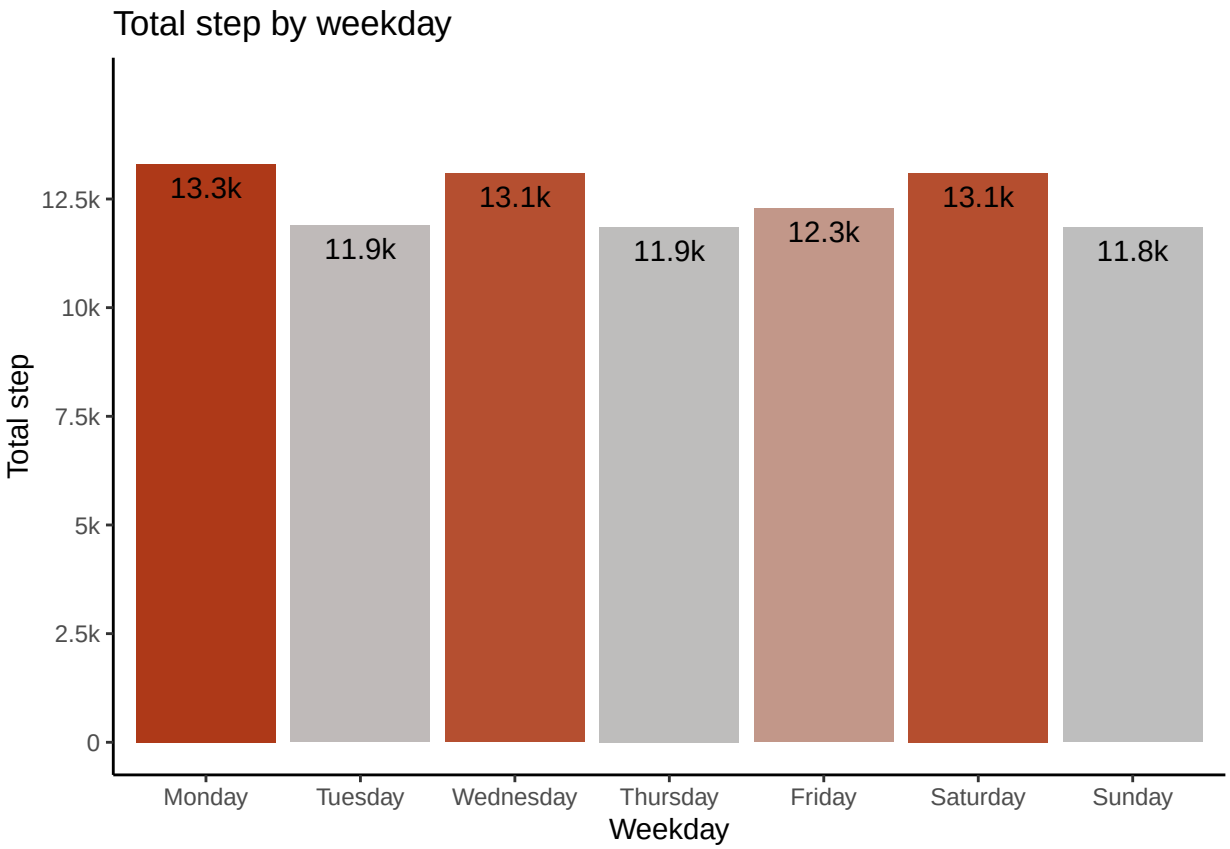
```
daily_activity %>%
  filter(
    group == 2
  )%>%
  summarize(
    very_active_dist = mean(very_active_dist),
    moderate_active_dist = mean(moderate_active_dist),
    light_active_dist = mean(light_active_dist),
    seden_active_dist = mean(seden_active_dist)
  )%>%
  summarize(
    very_active_dist = 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
```

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

Average steps by weekday

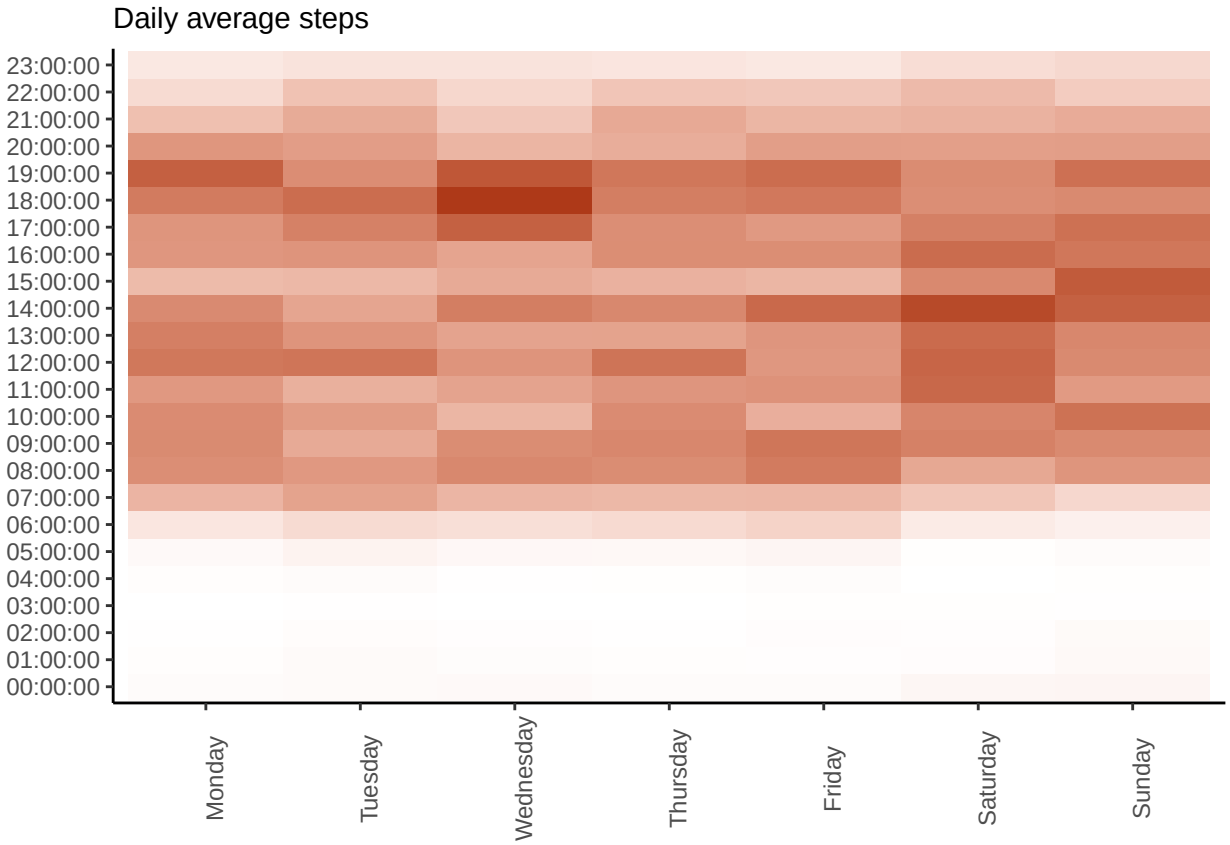
```
daily_activity %>%
  filter(
    group == 3
  )%>%
  group_by(
    weekday
  )%>%
  summarize(
    total_step = mean(total_step)
  )%>%
  ggplot(
    aes(
      x = weekday,
      y = total_step,
      fill = total_step
    )
  )+
  geom_bar(
    stat = "identity",
    show.legend = FALSE
  )+
  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()
```


Total step by weekday
Total step
12.5k
10k
7.5k
5k
2.5k
0
13.3k
11.9k
13.1k
11.9k
12.3k
13.1k
11.8k
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Weekday

Step heatmap

```
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 = "Daily average steps",
    x = NULL,
    y = NULL,
    fill = "Total step"
  )+
  theme_classic(
  )+
  theme(axis.text.x = element_text(angle = 90))
```

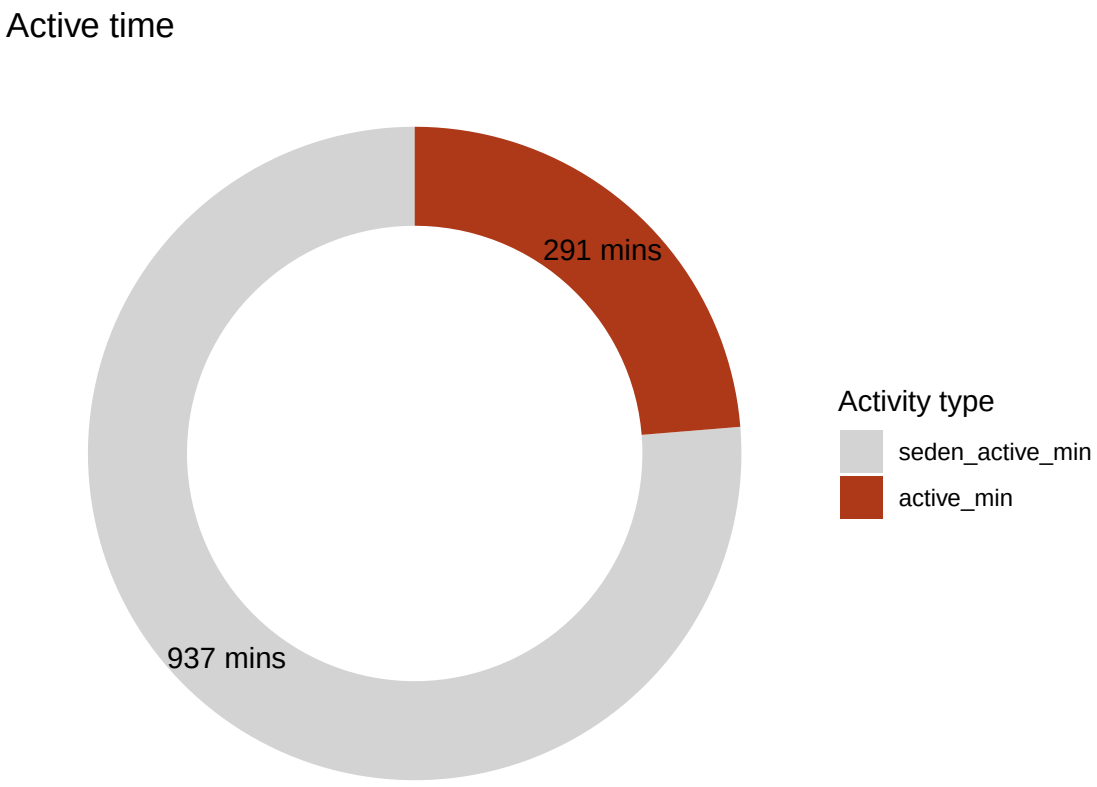
```
## `summarise()` has grouped output by 'weekday'. You can override using the
## `.groups` argument.
```


Daily average steps
23:00:00
22:00:00
21:00:00
20:00:00
19:00:00
18:00:00
17:00:00
16:00:00
15:00:00
14:00:00
13:00:00
12:00:00
11:00:00
10:00:00
09:00:00
08:00:00
07:00:00
06:00:00
05:00:00
04:00:00
03:00:00
02:00:00
01:00:00
00:00:00
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday

Activity type

```
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 = "Active time"
)+
theme_void()
```



Using frequency

```
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
```

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
```