Capstone Project

```
library(easypackages)
libraries("tidyverse", "lubridate", "skimr", "DataExplorer", "ggcorrplot")
```

```
options(scipen = 999)
```

Part A: Introduction

The health and fitness wearable tracker industry is booming in recent years. The global fitness tracker market is projected to grow from $36.34 billion in 2020 to $114.36 billion in 2028 at a CAGR of 15.4% in forecast period 2021-2028 (Fortune Business Insights 2021).

In 2020, the North American market size is USD $17.36 billion. During the same period, fitness band has the biggest market share in North American (43%) in terms of device type (Fortune Business Insights 2021).

It is under this context that the Bellabeat executive team would like to understand wearable tracker users behaviour in order to develop a data-driven marketing strategy for its products.

Part B: Data Preparation

This project uses the FitBit Fitness Tracker Data, which includes 18 data files. However, this project will only use 4 of these data files.

```
# Import data files
dailyActivity <- read_csv("Data/dailyActivity_merged.csv", col_types = cols())
sleepDay <- read_csv("Data/sleepDay_merged.csv", col_types = cols())
hourlySteps <- read_csv("Data/hourlySteps_merged.csv", col_types = cols())
hourlyIntensities <- read_csv("Data/hourlyIntensities_merged.csv", col_types = cols())
```

To merge the four files, I need to determine which file I should use as the primary file and then build the rest on it. So here is the data preparation procedure:

- To determine the primary file, I need to see which file has the most cases, and the most useful variables.
- Next, I will join the primary file and the other 3 with left join.
- Once I have the master data set, I will check the data.

```
#    Number of variables in each file
dim(dailyActivity)
```

```
## [1] 940  15
```

```
dim(sleepDay)
```

```
## [1] 413   5
```

```r
dim(hourlySteps)
```

```
## [1] 22099     3
```

```r
dim(hourlyIntensities)
```

```
## [1] 22099     4
```

```r
#   Number of unique cases in each file
n_distinct(dailyActivity$Id)
```

```
## [1] 33
```

```r
n_distinct(sleepDay$Id)
```

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

```r
n_distinct(hourlySteps$Id)
```

```
## [1] 33
```

```r
n_distinct(hourlyIntensities$Id)
```

```
## [1] 33
```

Create a master data file with all needed variables Since dailyActivity has the most relevant variables (15) and unique cases (33), I will use it as the foundation file and build the master from it. The primary keys will be Id and Date (created from field with date info). For the sake of joining files and analyzing data, I need to handle the date variable in each file first. This procedure includes:

- Convert the existing date field from char to date
- Extract the weekdays from the new date field
- Find out the number of week

```r
# Handle the dailyActivity file date field and order the weekdays
dailyActivity$date <- as.Date(dailyActivity$ActivityDate, "%m/%d/%Y")
dailyActivity$days <- weekdays(dailyActivity$date)
dailyActivity$week <- strftime(dailyActivity$date, format = "%V")

dailyActivity$days <- factor(dailyActivity$days,
                             c("Saturday", "Sunday", "Monday", "Tuesday",
                               "Wednesday","Thursday", "Friday"))
```

```r
# Handle the sleepDay file date field
sleepDay$date = as.Date(sleepDay$SleepDay, "%m/%d/%Y %I:%M:%S %p")
```

```
#  Create a new variable "insomenia" in the sleepDay file for analysis.
#  The large the value means that the longer time to fall asleep, which may be
#  caused by many reasons like **stress**. Thus, we use it as an indicator of
#  stress.
```

```
sleepDay <- sleepDay %>%
  mutate (insomnia = TotalTimeInBed - TotalMinutesAsleep)
```

```
# Convert the TotalMinutesAsleep minutes into hours.
sleepDay <- sleepDay %>%
  mutate (TotalHrsAsleep = round(TotalMinutesAsleep / 60))
```

```
# Handle the hourlySteps date field
hourlySteps$date <- as.Date(hourlySteps$ActivityHour, "%m/%d/%Y %I:%M:%S %p")
hourlySteps$hr <- parse_datetime(hourlySteps$ActivityHour, '%m/%d/%Y %I:%M:%S %p')
hourlySteps$hour <- hour(hourlySteps$hr)
```

```
# Handle the hourlyIntensities date field
hourlyIntensities$date <- as.Date(hourlyIntensities$ActivityHour, "%m/%d/%Y %I:%M:%S %p")
hourlyIntensities$hr <- parse_datetime(hourlyIntensities$ActivityHour, '%m/%d/%Y %I:%M:%S %p')
hourlyIntensities$hour <- hour(hourlyIntensities$hr)
```

```
# Since dailyActivity has the most obserations, I use it as the primary data set
# and left join other data sets to it so that most info will be preserved. If I
# use "merge", only cases found in both datasets will be kept, which is not I prefer.

master  <- dailyActivity %>% left_join(sleepDay, by=c("Id","date"))
master  <- master  %>% left_join(hourlySteps, by=c("Id", "date"))
master  <- master  %>% left_join(hourlyIntensities, by=c("Id", "date"))
```

Data Cleaning In terms of **data type**, let's check the structure to find out which one needs to be changed.

```
str(master)
```

```
## tibble [528,387 x 33] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Id                       : num [1:528387] 1503960366 1503960366 1503960366 1503960366 1503960366 .
##  $ ActivityDate             : chr [1:528387] "4/12/2016" "4/12/2016" "4/12/2016" "4/12/2016" ...
##  $ TotalSteps               : num [1:528387] 13162 13162 13162 13162 13162 ...
##  $ TotalDistance            : num [1:528387] 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 ...
##  $ TrackerDistance          : num [1:528387] 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 8.5 ...
##  $ LoggedActivitiesDistance: num [1:528387] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveDistance       : num [1:528387] 1.88 1.88 1.88 1.88 1.88 ...
##  $ ModeratelyActiveDistance: num [1:528387] 0.55 0.55 0.55 0.55 0.55 ...
##  $ LightActiveDistance      : num [1:528387] 6.06 6.06 6.06 6.06 6.06 ...
##  $ SedentaryActiveDistance : num [1:528387] 0 0 0 0 0 0 0 0 0 0 ...
##  $ VeryActiveMinutes        : num [1:528387] 25 25 25 25 25 25 25 25 25 25 ...
##  $ FairlyActiveMinutes      : num [1:528387] 13 13 13 13 13 13 13 13 13 13 ...
##  $ LightlyActiveMinutes     : num [1:528387] 328 328 328 328 328 328 328 328 328 328 ...
##  $ SedentaryMinutes         : num [1:528387] 728 728 728 728 728 728 728 728 728 728 ...
##  $ Calories                 : num [1:528387] 1985 1985 1985 1985 1985 ...
##  $ date                     : Date[1:528387], format: "2016-04-12" "2016-04-12" ...
```

```
##  $ days                    : Factor w/ 7 levels "Saturday","Sunday",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ week                    : chr [1:528387] "15" "15" "15" "15" ...
##  $ SleepDay                : chr [1:528387] "4/12/2016 12:00:00 AM" "4/12/2016 12:00:00 AM" "4/12/20
##  $ TotalSleepRecords       : num [1:528387] 1 1 1 1 1 1 1 1 1 1 ...
##  $ TotalMinutesAsleep      : num [1:528387] 327 327 327 327 327 327 327 327 327 327 ...
##  $ TotalTimeInBed          : num [1:528387] 346 346 346 346 346 346 346 346 346 346 ...
##  $ insomnia                : num [1:528387] 19 19 19 19 19 19 19 19 19 19 ...
##  $ TotalHrsAsleep          : num [1:528387] 5 5 5 5 5 5 5 5 5 5 ...
##  $ ActivityHour.x          : chr [1:528387] "4/12/2016 12:00:00 AM" "4/12/2016 12:00:00 AM" "4/12/20
##  $ StepTotal               : num [1:528387] 373 373 373 373 373 373 373 373 373 373 ...
##  $ hr.x                    : POSIXct[1:528387], format: "2016-04-12 00:00:00" "2016-04-12 00:00:00"
##  $ hour.x                  : int [1:528387] 0 0 0 0 0 0 0 0 0 0 ...
##  $ ActivityHour.y          : chr [1:528387] "4/12/2016 12:00:00 AM" "4/12/2016 1:00:00 AM" "4/12/2016
##  $ TotalIntensity          : num [1:528387] 20 8 7 0 0 0 0 0 13 30 ...
##  $ AverageIntensity        : num [1:528387] 0.333 0.133 0.117 0 0 ...
##  $ hr.y                    : POSIXct[1:528387], format: "2016-04-12 00:00:00" "2016-04-12 01:00:00"
##  $ hour.y                  : int [1:528387] 0 1 2 3 4 5 6 7 8 9 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Id = col_double(),
##   ..   ActivityDate = col_character(),
##   ..   TotalSteps = col_double(),
##   ..   TotalDistance = col_double(),
##   ..   TrackerDistance = col_double(),
##   ..   LoggedActivitiesDistance = col_double(),
##   ..   VeryActiveDistance = col_double(),
##   ..   ModeratelyActiveDistance = col_double(),
##   ..   LightActiveDistance = col_double(),
##   ..   SedentaryActiveDistance = col_double(),
##   ..   VeryActiveMinutes = col_double(),
##   ..   FairlyActiveMinutes = col_double(),
##   ..   LightlyActiveMinutes = col_double(),
##   ..   SedentaryMinutes = col_double(),
##   ..   Calories = col_double()
##   .. )
```

Based on the result, "master" is a data.table, not data.frame. It's better to convert it to **data.frame**.

```
master <- as.data.frame(master)
```

Next, Id is the only variable I need to change from numeric to character.

```
master$Id <- as.character(master$Id)
```

Also, I need to **remove the redundant variables**, which is a result after joining the datasets. In the dailyActivity df, ActivityDate is a character. Since I have created a date field "date" to replace it, and also checked "date" has the same nrow and no NA in this field, it is safe to remove the "dailyActivity".

```
master = master[ , -2]
```

In the master df, the ActivityHour.x is the same as hr.x. Both came from the hourlySteps file. hr.x is a date time data type. Same with ActivityHour.y and hr.y. Therefore, I will **remove the ActivityHour.x and ActivityHour.y**

```
master = master[ -c(24, 28)]
```

I will also **rename** couple variables. hr.x, hr.y, hour.x, and hour.y. ".x" is for steps and ".y" is for intensities.

```
names(master)[names(master)== "hr.x"] <- "stephr"
names(master)[names(master)== "hour.x"] <- "step_24hour"
names(master)[names(master)== "hr.y"] <- "intenhr"
names(master)[names(master)== "hour.y"] <- "inten_24hour"
```

Missing values (NA)
To handle missing values, we need to determine where the missing values come from. Let's see the four individual data sets first.

```
table(is.na(dailyActivity))
```

```
##
## FALSE
## 16920
```

```
table(is.na(sleepDay))
```

```
##
## FALSE
##  3304
```

```
table(is.na(hourlyIntensities))
```

```
##
##  FALSE
## 154693
```

```
table(is.na(hourlySteps))
```

```
##
##  FALSE
## 132594
```

There is no missing value in each of the four data sets. How about in the master dateframe?

```
table(is.na(master))
```

```
##
##    FALSE     TRUE
## 14079228  1772382
```

But in the master dataframe, there are lots of missing values. All of these missing values come from 3 data sets. It is understandable because we used left join to join dailyActivity with the 3 data sets. Since dailyActivity contains more observations than the rest, it is reasonable to see missing values from the 3 data set. So should we keep completed cases only? The answer is **no**.

It is because the sample size of the original data set is small (33 cases). If we only keep the completed cases, the sample size will be shrink even further. Therefore, I prefer to keep all cases. But ignore or remove NA in certain analytical topics.

Outliers?
We noticed that there are some **outliers** in the data set, e.g. TotalSteps, TotalDistance, VeryActiveDistance, etc. Will it be an issue? Let's identify them first based on the maximum value.

```
m <- master %>% summarise(m1 = max(TotalSteps), m2 = max(TotalDistance),
                          m3 = max(VeryActiveDistance))

master %>% select(Id, TotalSteps) %>% group_by (Id) %>%
  filter(TotalSteps == m$m1) %>% summarise(n_distinct(n()))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 1 x 2
##   Id         `n_distinct(n())`
##   <chr>                  <int>
## 1 1624580081                 1
```

```
master %>% select(Id, TotalDistance) %>% group_by (Id) %>%
  filter(TotalDistance >= m$m2) %>% summarise(n_distinct(n()))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 1 x 2
##   Id         `n_distinct(n())`
##   <chr>                  <int>
## 1 1624580081                 1
```

```
master %>% select(Id, VeryActiveDistance) %>% group_by (Id) %>%
  filter(VeryActiveDistance >= m$m3 ) %>% summarise(n_distinct(n()))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 1 x 2
##   Id         `n_distinct(n())`
##   <chr>                  <int>
## 1 1624580081                 1
```

The results show that these extreme cases are the same person. It is reasonable to think that these values are not resulted from typo. Instead, they are the actual records of the same person. Therefore, we will not remove these extreme cases but will be cautious when analyzing the data.

An unsolved issue - Data Consistency
The variable of "distance" does not come with a measurement unit (kilometres / miles), which will create a data consistency issue. The Fitbit device lets users to set the measurement unit. Without adding the measurement unit in the field, the same value could have different meanings in different observations. Since kilometres are longer than miles, the issue will affect the interpretation substantially.

After checking these aspects, the master data set is ready to use.

Part C: Data Exploration Analysis - Sleep Well & Works Out

According to the National Sleep Foundation, It's normal to take **10 to 20 minutes** to fall asleep once a person climbs into bed.
In **our sample**, it will take **39 minutes to fall asleep on average** (i.e. minutes from go to bed to fall asleep). The median is **25.5 minutes**, which is close to the suggested normal minutes.

The Foundation also suggests the **Recommended Hours of Sleep is 7-9 hours** for adults (aged 18-60). In **our sample**, respondents have **around 7 hours of sleep**.

According to the 2018 Physical Activity Guidelines for Americans, 2nd edition, from the US CDC, adults should have **150 minutes physical activities per week**. We will see how active of our sample is later.

```
df_DEA = master %>% select(Id, date, week, insomnia, TotalHrsAsleep, SedentaryMinutes,
                           Calories, TotalSteps, TotalDistance) %>%
  mutate(SedentaryHrs = SedentaryMinutes/60)

df_DEA = df_DEA [ !duplicated(df_DEA), ]
df_DEA <- df_DEA[complete.cases(df_DEA), ]
n_distinct(df_DEA$Id)
```

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

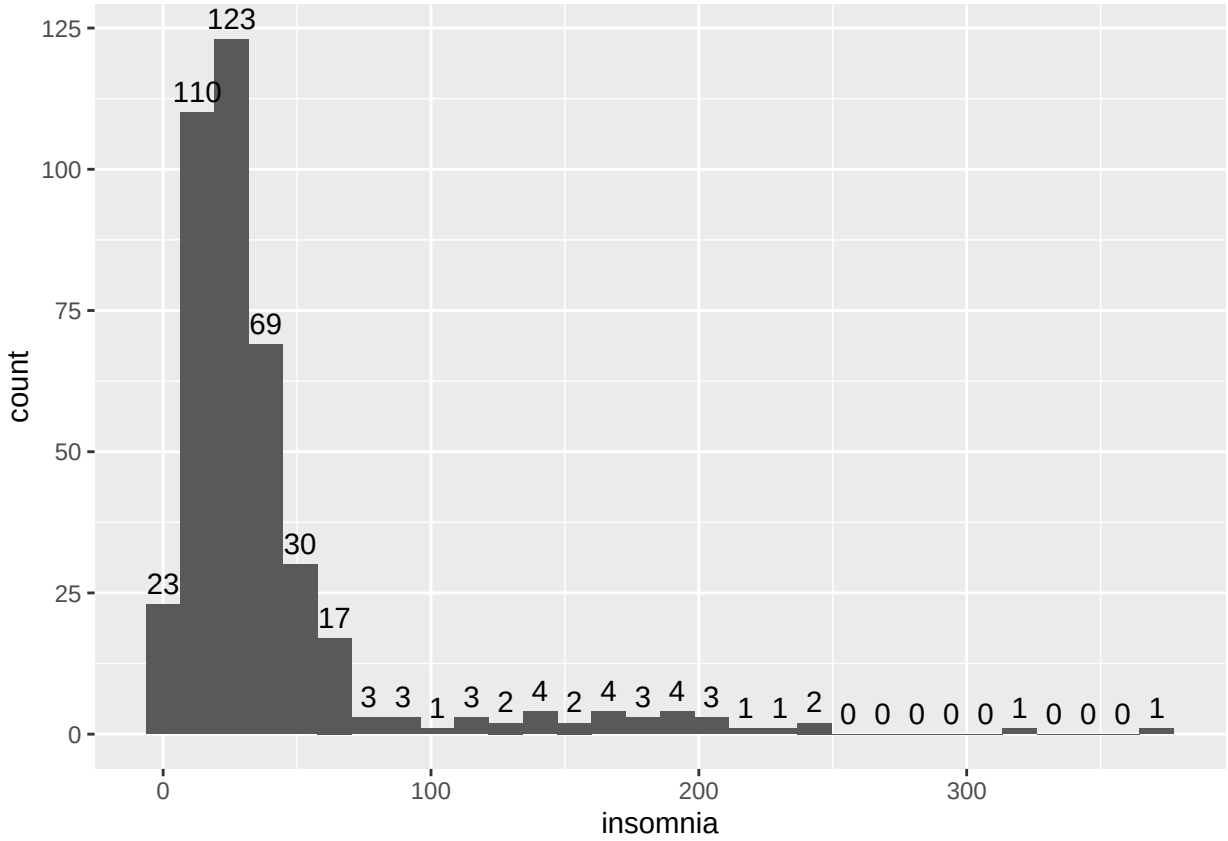
```
summary(df_DEA)
```

```
##       Id                 date                week             insomnia
##  Length:410         Min.   :2016-04-12   Length:410         Min.   :  0.00
##  Class :character   1st Qu.:2016-04-19   Class :character   1st Qu.: 17.00
##  Mode  :character   Median :2016-04-27   Mode  :character   Median : 25.50
##                     Mean   :2016-04-26                      Mean   : 39.31
##                     3rd Qu.:2016-05-04                      3rd Qu.: 40.00
##                     Max.   :2016-05-12                      Max.   :371.00
##  TotalHrsAsleep  SedentaryMinutes    Calories      TotalSteps
##  Min.   : 1.00   Min.   :   0.0   Min.   : 257   Min.   :   17
##  1st Qu.: 6.00   1st Qu.: 631.2   1st Qu.:1841   1st Qu.: 5189
##  Median : 7.00   Median : 717.0   Median :2207   Median : 8913
##  Mean   : 6.99   Mean   : 712.1   Mean   :2389   Mean   : 8515
##  3rd Qu.: 8.00   3rd Qu.: 782.8   3rd Qu.:2920   3rd Qu.:11370
##  Max.   :13.00   Max.   :1265.0   Max.   :4900   Max.   :22770
##  TotalDistance     SedentaryHrs
##  Min.   : 0.010   Min.   : 0.00
##  1st Qu.: 3.592   1st Qu.:10.52
##  Median : 6.270   Median :11.95
##  Mean   : 6.012   Mean   :11.87
##  3rd Qu.: 8.005   3rd Qu.:13.05
##  Max.   :17.540   Max.   :21.08
```

Minutes to Fall Asleep (Suffer from insomnia)

```
ggplot(df_DEA, aes(x=insomnia))+ geom_histogram() +
  stat_bin(aes(y=..count.., label=..count..), geom="text", vjust=-.5)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



How many cases suffered from insomnia every week?

```
df_DEA_Inso = df_DEA %>% group_by(Id, week) %>% summarise(avginso = mean(insomnia))
```

```
## `summarise()` regrouping output by 'Id' (override with `.groups` argument)
```

Filter by more than 25 minutes

```
df_insomnia = df_DEA %>% filter(insomnia > 25)
n_distinct(df_insomnia$Id)
```

```
## [1] 18
```

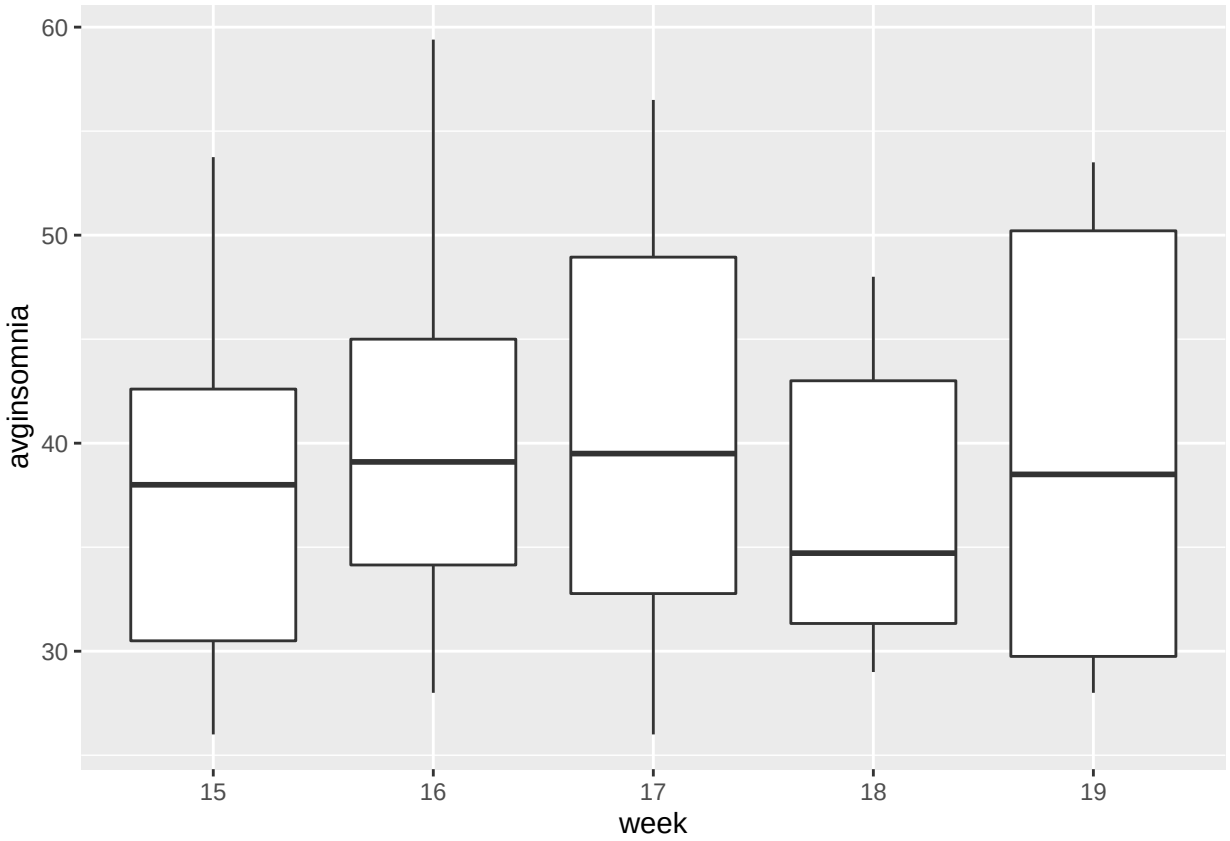
There are 18 cases have insomnia.

```
df_insomnia = df_insomnia %>% group_by(Id, week) %>%
  summarise(avginsomnia = mean(insomnia))
```

```
## `summarise()` regrouping output by 'Id' (override with `.groups` argument)
```

Remove the extreme cases

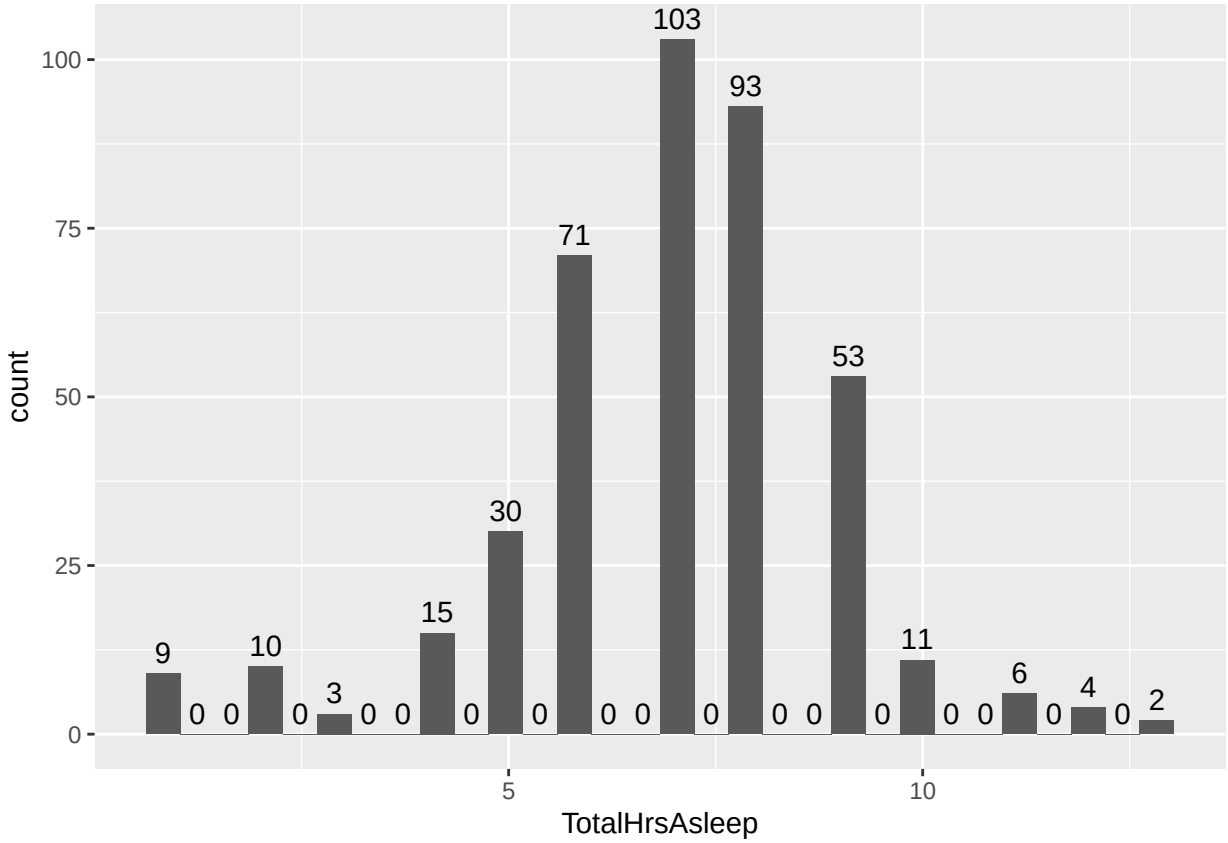
```r
df_insomnia_1 = df_insomnia %>% filter (avginsomnia < 100)

ggplot(data =df_insomnia_1, mapping = aes(x= week, y = avginsomnia))+
  geom_boxplot()
```



Sleep Pattern **Hours of sleep**

```r
ggplot(df_DEA, aes(x=TotalHrsAsleep))+ geom_histogram() +
  stat_bin(aes(y=..count.., label=..count..), geom="text", vjust=-.5)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



From the histogram of total hours of sleep, we see most respondents have **7 to 8 hours of sleep** (103 cases and 93 cases respectively).

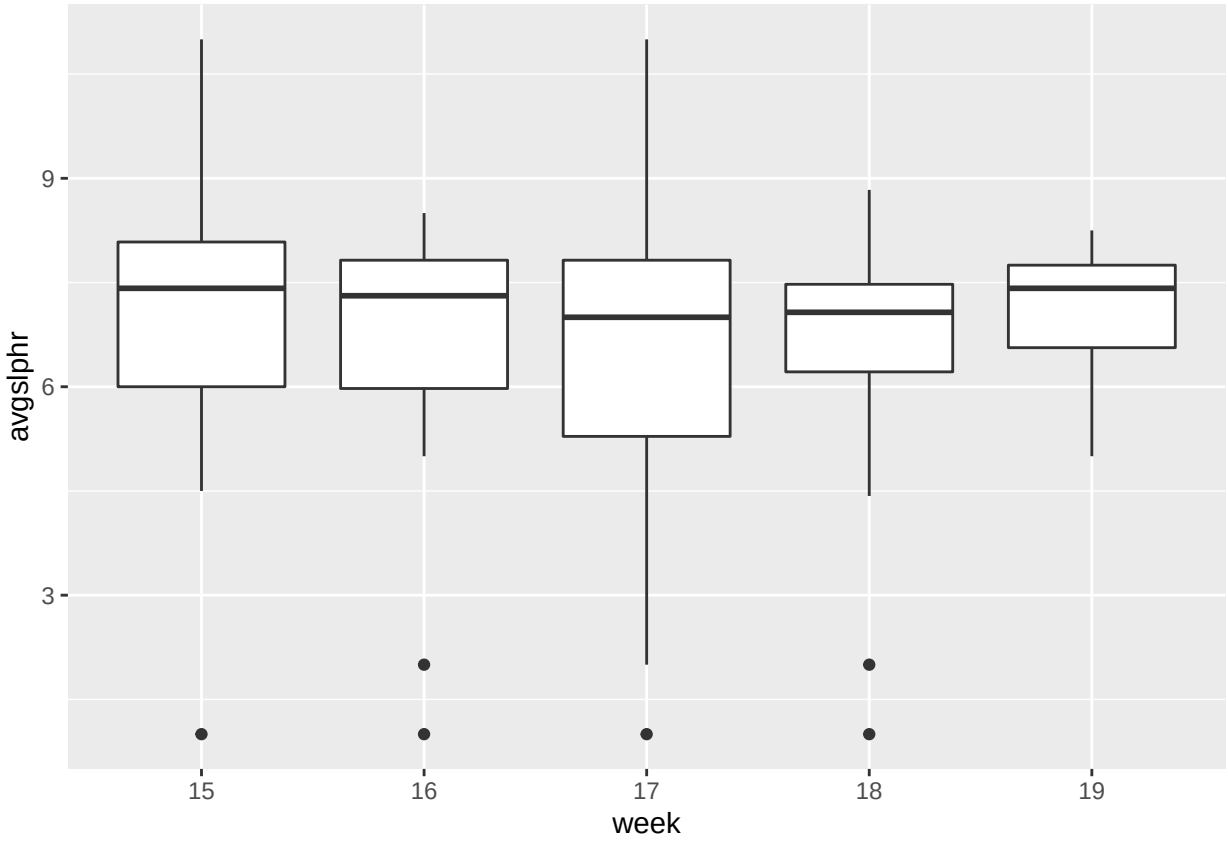
```
df_DEA_SlpHr = df_DEA %>% group_by(Id, week) %>% summarise(avgslphr = mean(TotalHrsAsleep))
```

```
## `summarise()` regrouping output by 'Id' (override with `.groups` argument)
```

```
summary(df_DEA_SlpHr$avgslphr)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   6.000   7.167   6.678   7.830  11.000
```

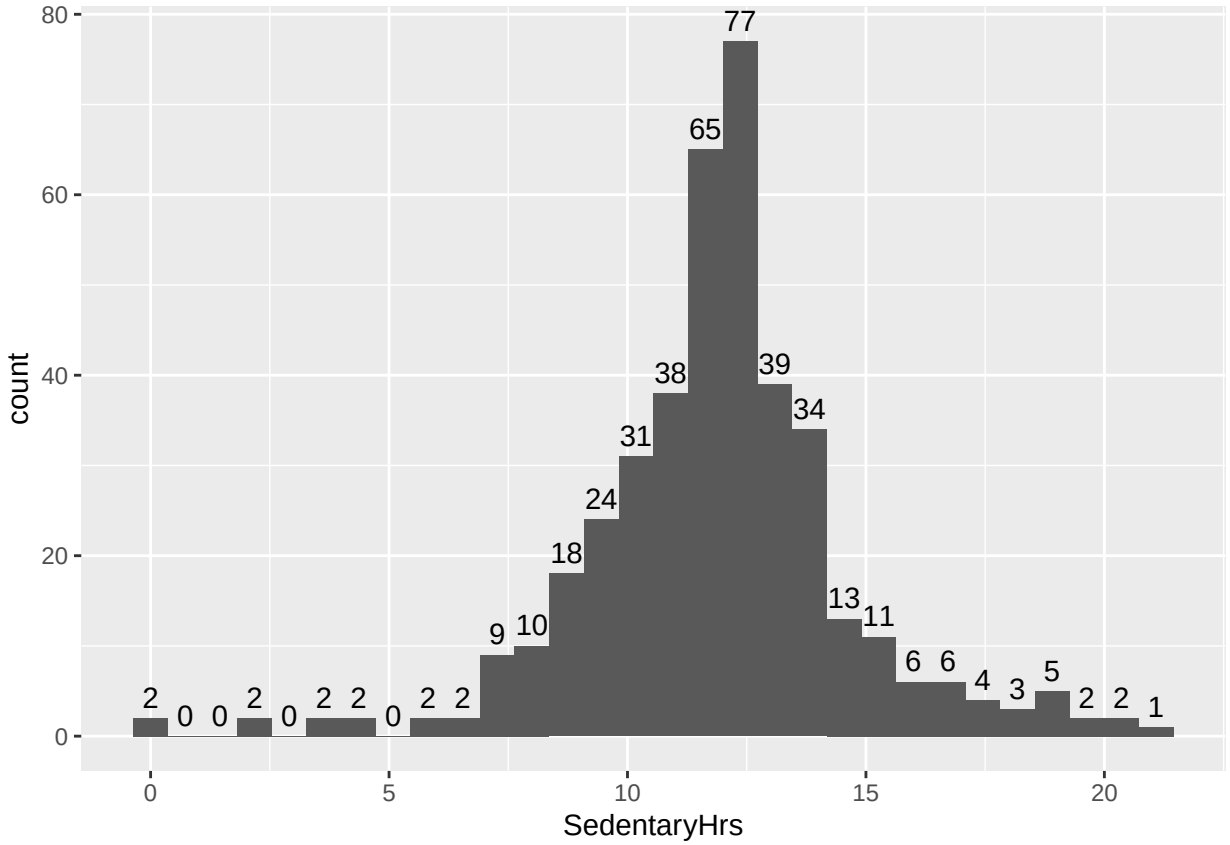
```
ggplot(data =df_DEA_SlpHr, mapping = aes(x= week, y = avgslphr))+ geom_boxplot()
```



Exercise at all? SedentaryMinutes - Sitting or inactive

```
ggplot(df_DEA, aes(x=SedentaryHrs))+ geom_histogram() +
  stat_bin(aes(y=..count.., label=..count..), geom="text", vjust=-.5)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Most respondents have **12 non-active / sitting hours** on average.

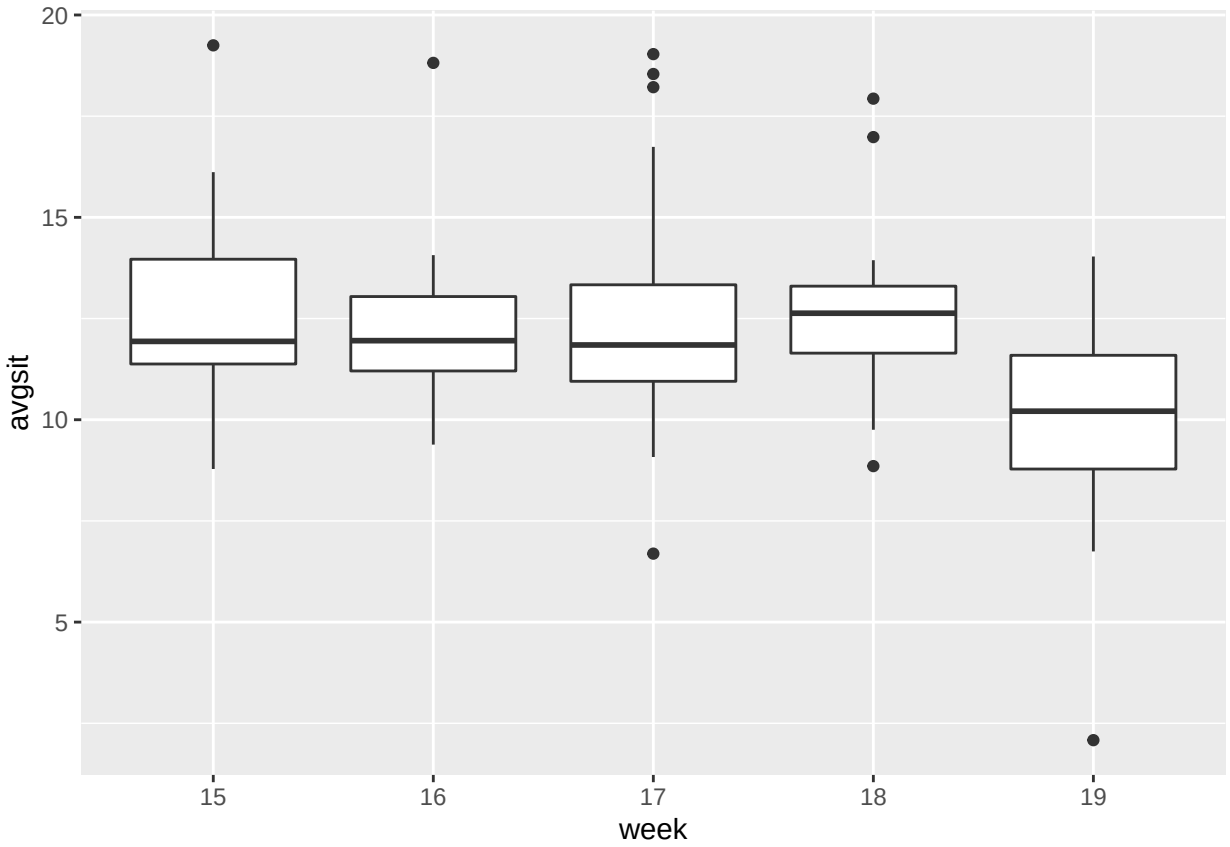
```
df_DEA_Sit = df_DEA %>% group_by(Id, week) %>% summarise(avgsit = mean(SedentaryHrs))
```

```
## `summarise()` regrouping output by 'Id' (override with `.groups` argument)
```

```
summary(df_DEA_Sit$avgsit)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.083  10.907  11.862  12.160  13.067  19.250
```

```
ggplot(data =df_DEA_Sit, mapping = aes(x=week, y = avgsit))+ geom_boxplot()
```



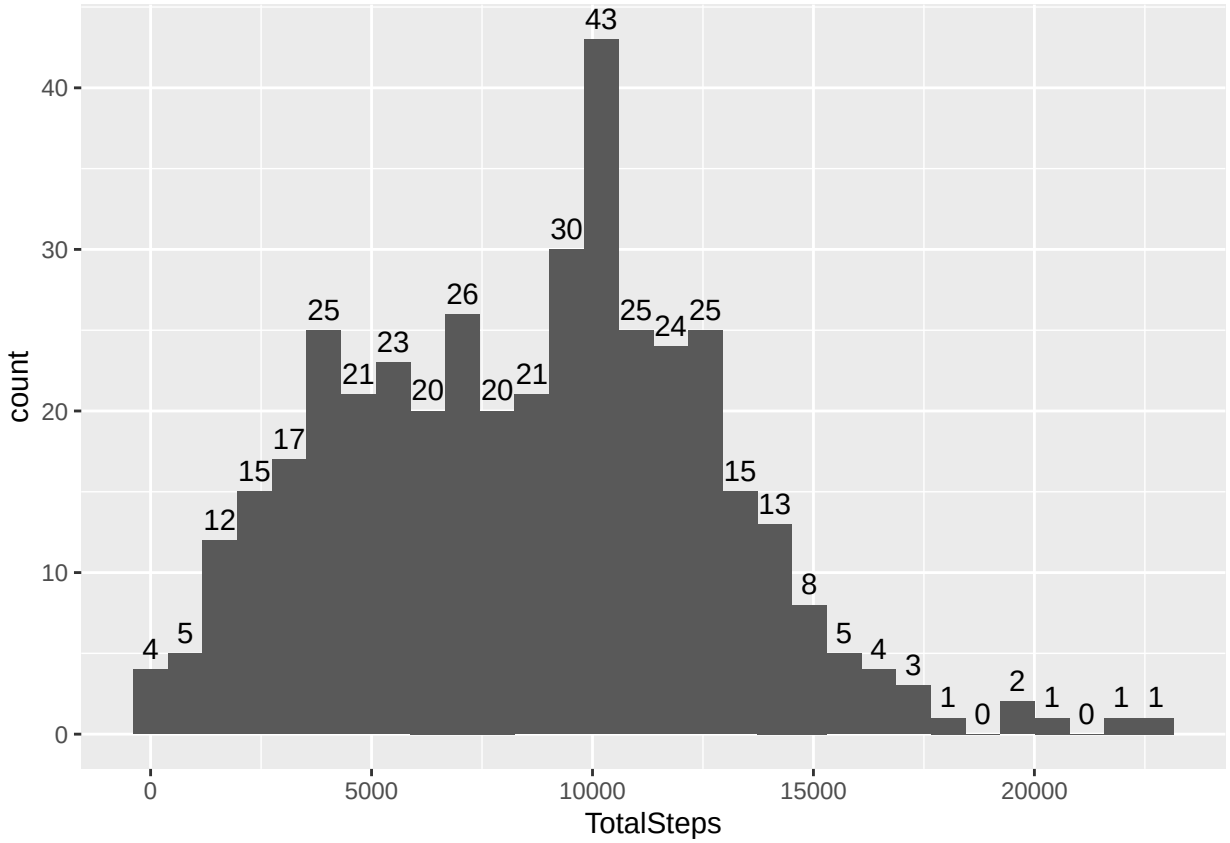
In short, half of the observations fell asleep within 26 minutes which is slightly longer than the normal 20 minutes. The majority of the respondents had seven or more hours of sleep.
That said, some respondents still experienced difficult of falling asleep despite exercise often. Within the five weeks observation, there were 66 observations have experienced insomnia.

Exercise - Steps

CDC recommends most adults aim for **10,000 steps per day.** For most people, this is the equivalent of about 8 kilometers, or 5 miles.
In **our sample**, it is slightly **more than 8000 steps** on average.

```
ggplot(df_DEA, aes(x=TotalSteps))+ geom_histogram() +
  stat_bin(aes(y=..count.., label=..count..), geom="text", vjust=-.5)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



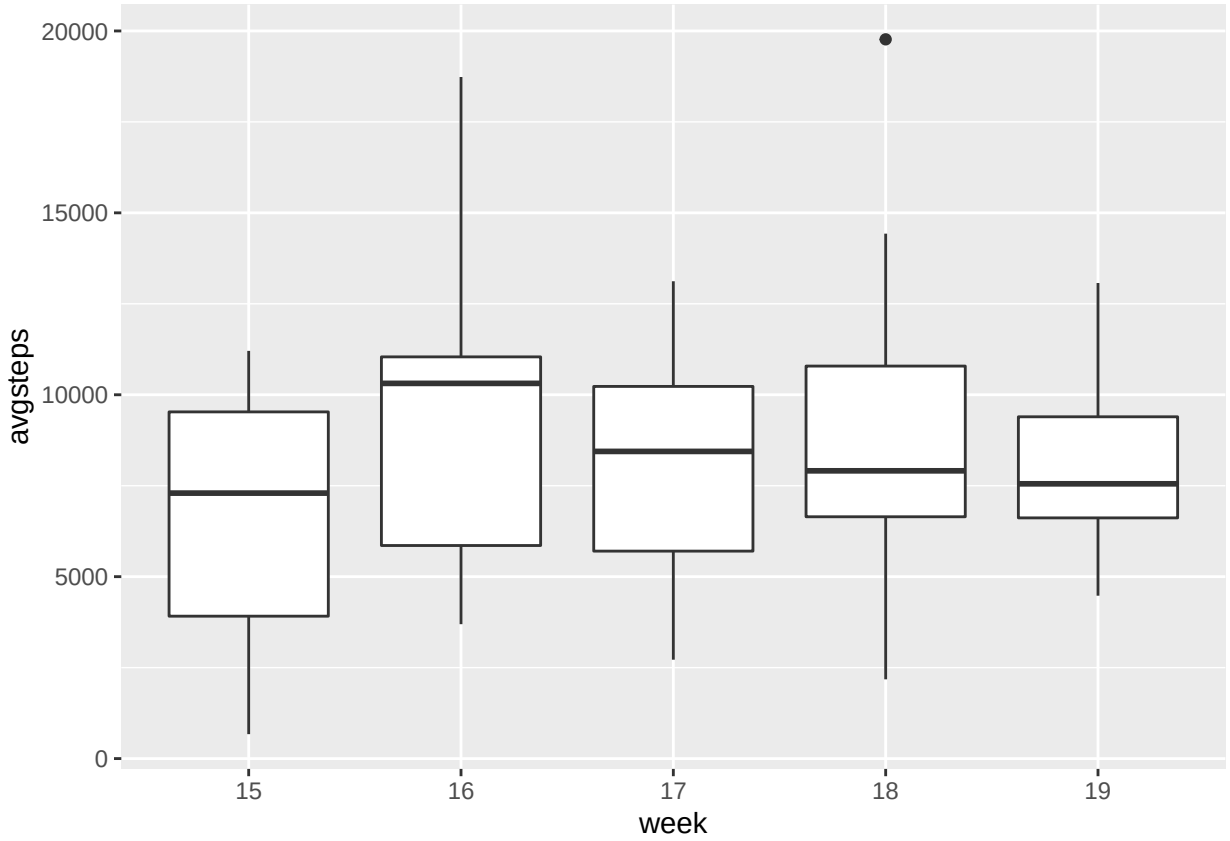

```
df_DEA_Steps = df_DEA %>% group_by(Id, week) %>% summarise(avgsteps = mean(TotalSteps))
```

```
## `summarise()` regrouping output by 'Id' (override with `.groups` argument)
```

```
summary(df_DEA_Steps$avgsteps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   671.3  5559.5  8356.9  8244.8 10365.0 19769.0
```

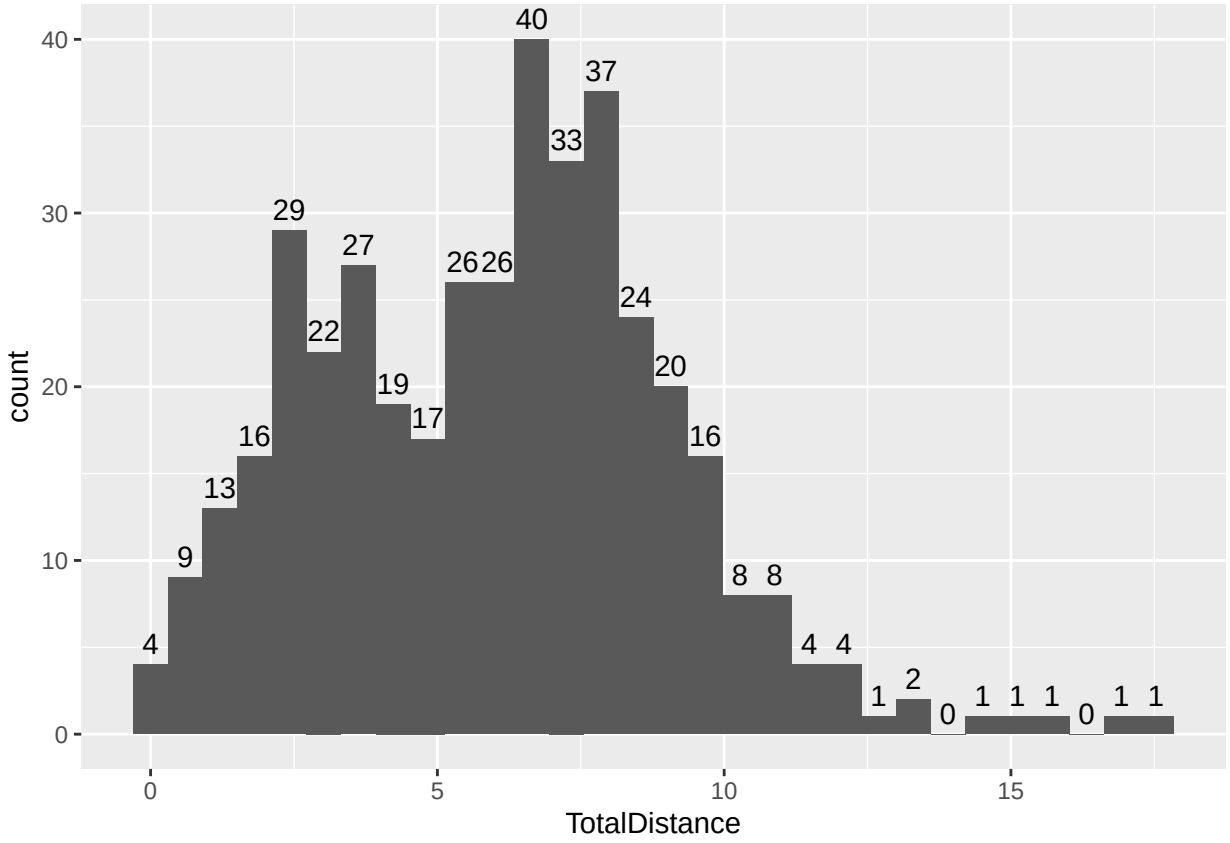
```
ggplot(data =df_DEA_Steps, mapping = aes(x=week, y = avgsteps))+ geom_boxplot()
```



Exercise - Distance
In terms of distance, it is about **6 km** on average.

```
ggplot(df_DEA, aes(x=TotalDistance))+ geom_histogram() +
  stat_bin(aes(y=..count.., label=..count..), geom="text", vjust=-.5)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



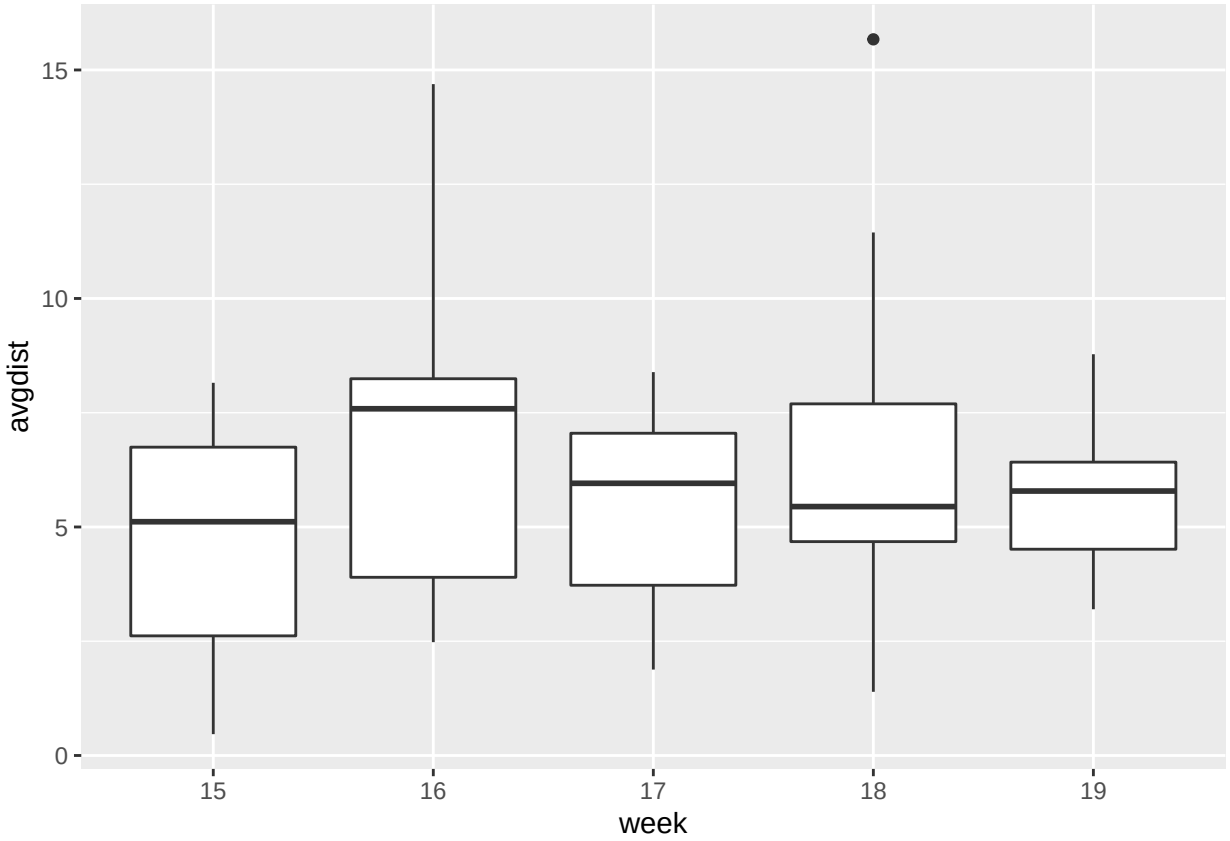

```
df_DEA_Dist = df_DEA %>% group_by(Id, week) %>% summarise(avgdist = mean(TotalDistance))
```

```
## `summarise()` regrouping output by 'Id' (override with `.groups` argument)
```

```
summary(df_DEA_Dist$avgdist)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4667  3.7800  5.9037  5.8330  7.6625 15.6700
```

```
ggplot(data =df_DEA_Dist, mapping = aes(x=week, y = avgdist))+ geom_boxplot()
```



How many **Calories Burned** among these people?
On average, it is slightly more than **2200 calories** been burned in our sample.

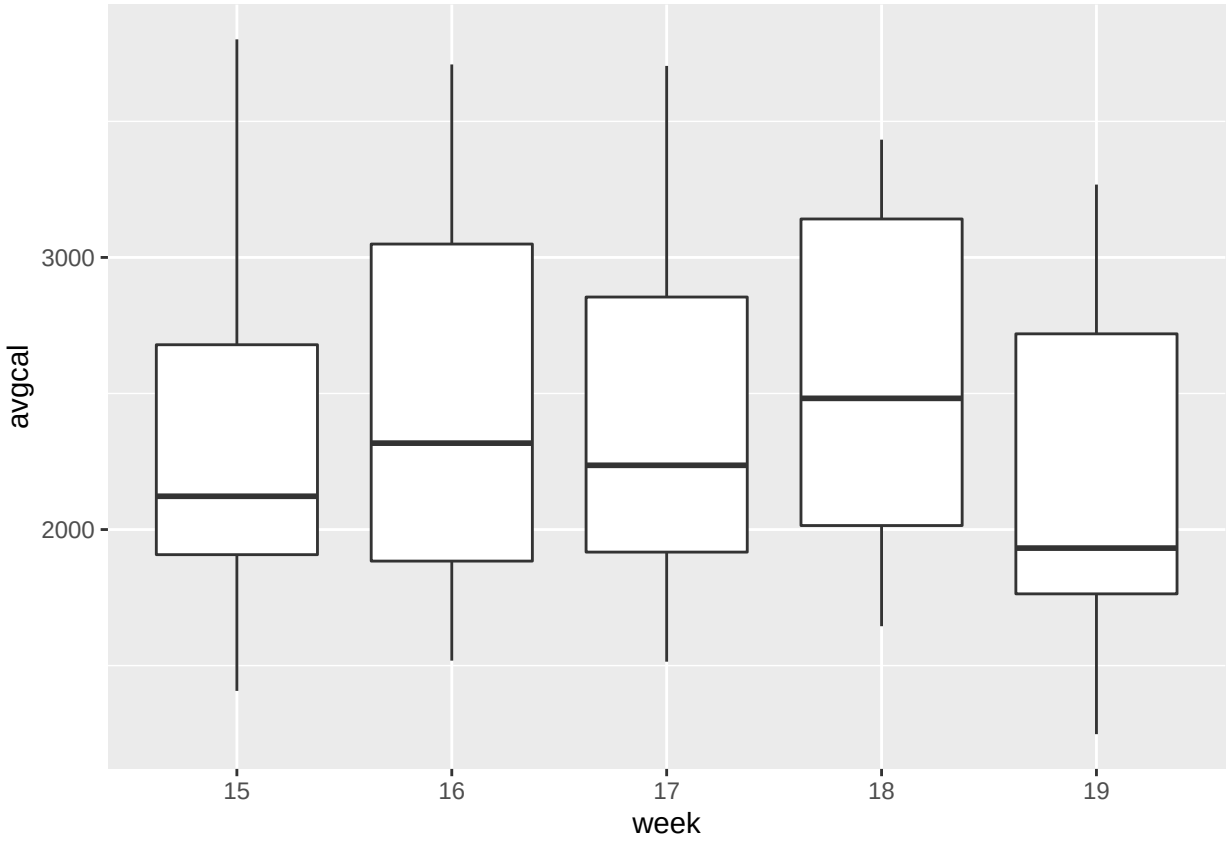
```
df_DEA_Calo = df_DEA %>% group_by(Id, week) %>% summarise(avgcal = mean(Calories))
```

```
## `summarise()` regrouping output by 'Id' (override with `.groups` argument)
```

```
summary(df_DEA_Calo$avgcal)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1248    1869    2211    2384    2920    3802
```

```
ggplot(data =df_DEA_Calo, mapping = aes(x=week, y = avgcal))+ geom_boxplot()
```



Part D: Analysis - Relationships Between Sleep & Exercise

```
df_relp = master [ !duplicated(master), ]
df_relp <- df_relp[complete.cases(df_relp), ]
```

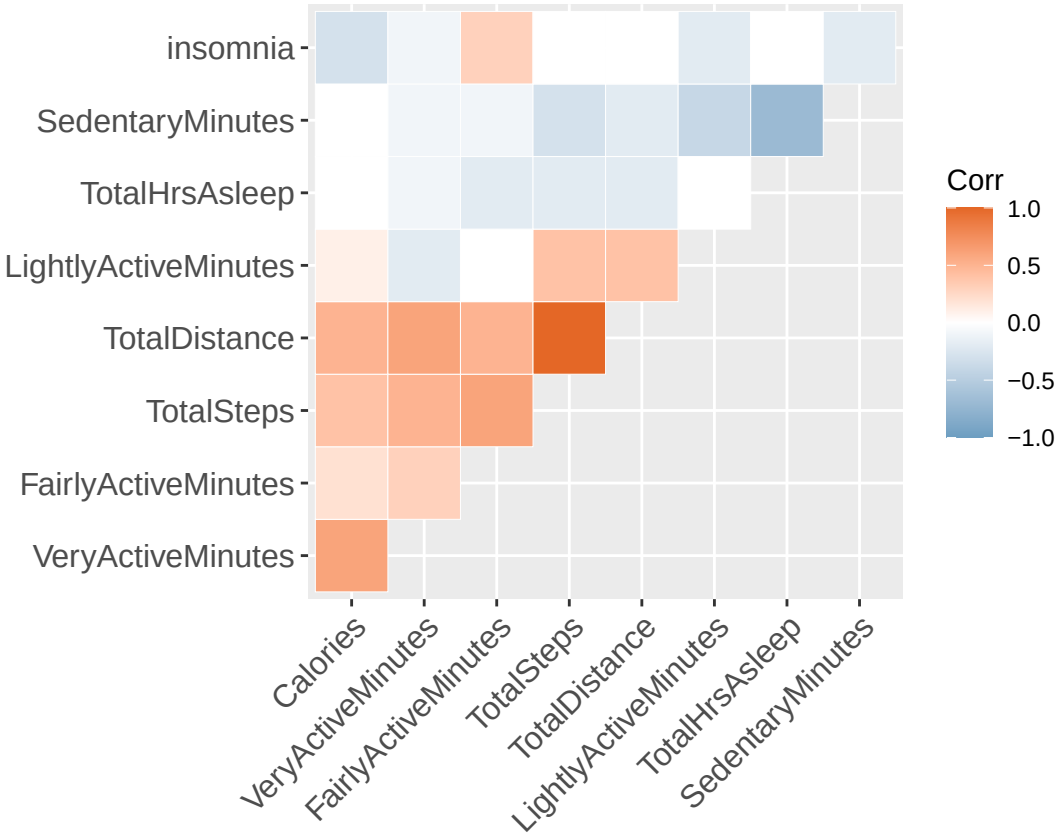
Correlations among relevant variables
The focus of this section is on sleep and exercise. It aims to identiy, if any, relationships between sleep and exercise. Here are the findings:

- The correlation plot shows that Calories and Sedentary Minutes will affect sleeping in a couple ways.
- Calories has a moderate inverse relationship with Insomnia. In other words, the fewer the calories burned, the longer the minutes of falling asleep.
- Meanwhile, Sedentary Minutes is strongly associated with Total Hours Asleep inversely. The longer the inactive time (or sitting time), the fewer the hours of sleep.
- On the other hand, Calories is positvely associated with exercise indicators.
  For instance, Calories is strongly associated with Very Active Minutes and Total Distance.

Accordingly, healthy sleep (i.e. fall asleep faster and sleep more soundly) may be related to the amount of calories burned and inactive time.
In other words, we might need to do more exercises for healthy sleep.

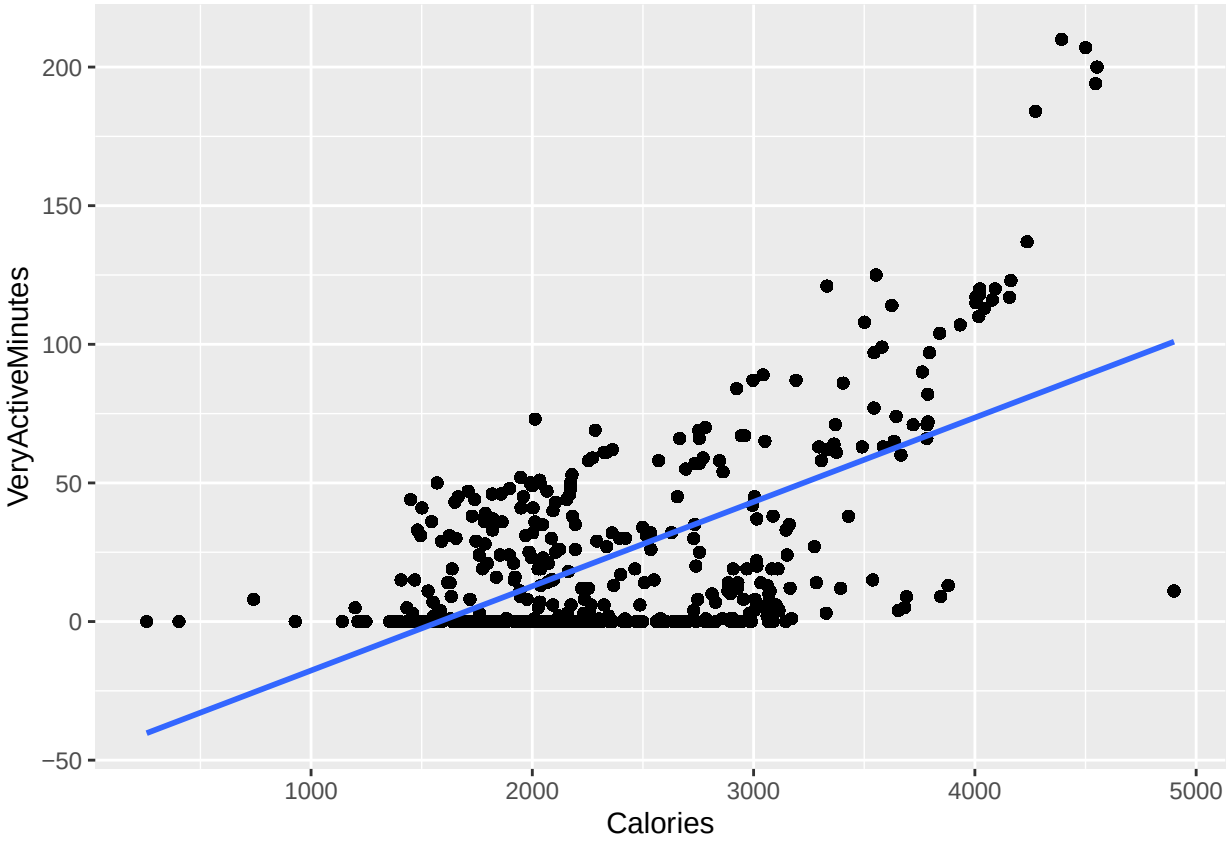
```
df_relp_1 = df_relp %>% select( c( 14, 2, 3, 10, 11, 12, 13, 22, 23))
```

```
corr <- round(cor(df_relp_1), 1)
ggcorrplot(corr, hc.order = TRUE, type = "upper",
           outline.col = "white",
           ggtheme = ggplot2::theme_gray,
           colors = c("#6D9EC1", "white", "#E46726"))
```



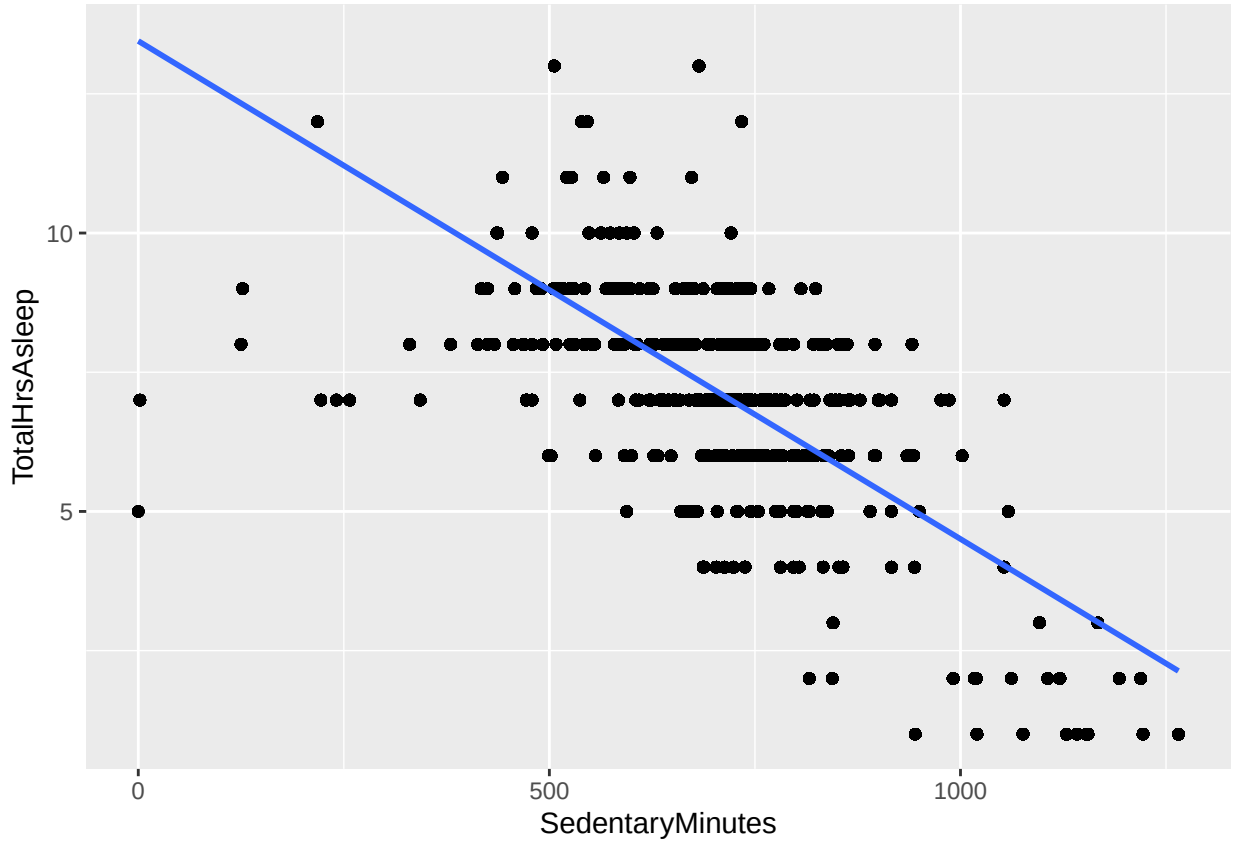

```
ggplot(data=df_relp_1) +
  geom_point(aes(x=Calories, y=VeryActiveMinutes))+
  geom_smooth(method =lm, aes(x=Calories, y=VeryActiveMinutes))
```

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




```
ggplot(data=df_relp_1) +
  geom_point(aes(x=SedentaryMinutes, y=TotalHrsAsleep))+
  geom_smooth(method =lm, aes(x=SedentaryMinutes, y=TotalHrsAsleep))
```

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



Part E: Analysis - Workouts Patterns

Which weekdays exercise more?
Survey respondents were more willing to exercise more on Saturday regardless the distance and intensity. Then, they started to lose the momentum during the weekdays until Saturday. Users spent the least exercising time on Sunday. Maybe users need to rest or do other activities.

```
df_I = df_relp %>% group_by(days) %>%
  summarise(avgIntensity = mean(TotalIntensity))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

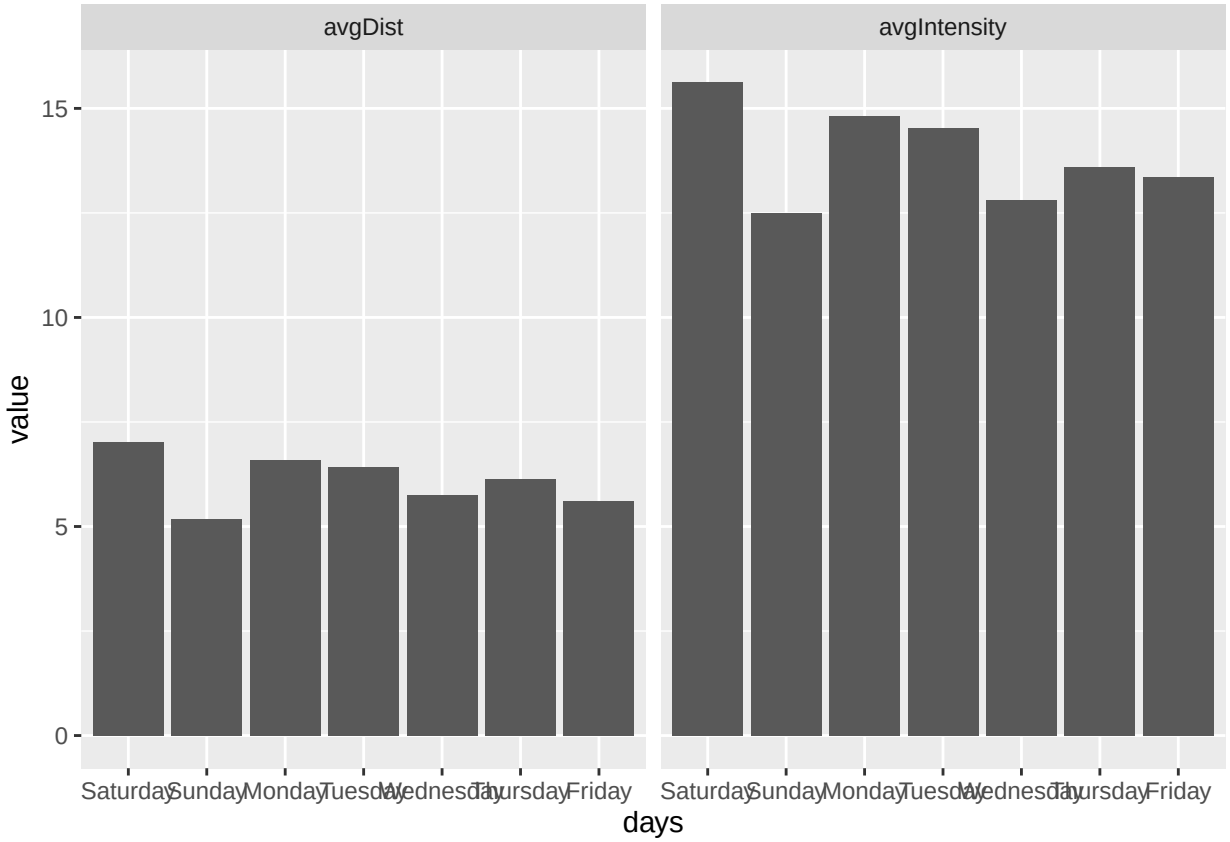
```
df_S = df_relp %>% group_by(days) %>%
  summarise(avgSteps = mean(TotalSteps))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
df_D = df_relp %>% group_by(days) %>%
  summarise(avgDist = mean(TotalDistance))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
df_gpA = cbind(df_I, df_S[!names(df_S) %in%
                            names(df_I)])

df_gpA = cbind(df_gpA, df_D[!names(df_D) %in%
                              names(df_gpA)])

df_gpA <- df_gpA %>% gather(wkout, value,
                            c(avgIntensity, avgDist))

ggplot(df_gpA) +
  geom_col(mapping = aes(x=days, y=value)) +
  facet_wrap(~wkout)
```

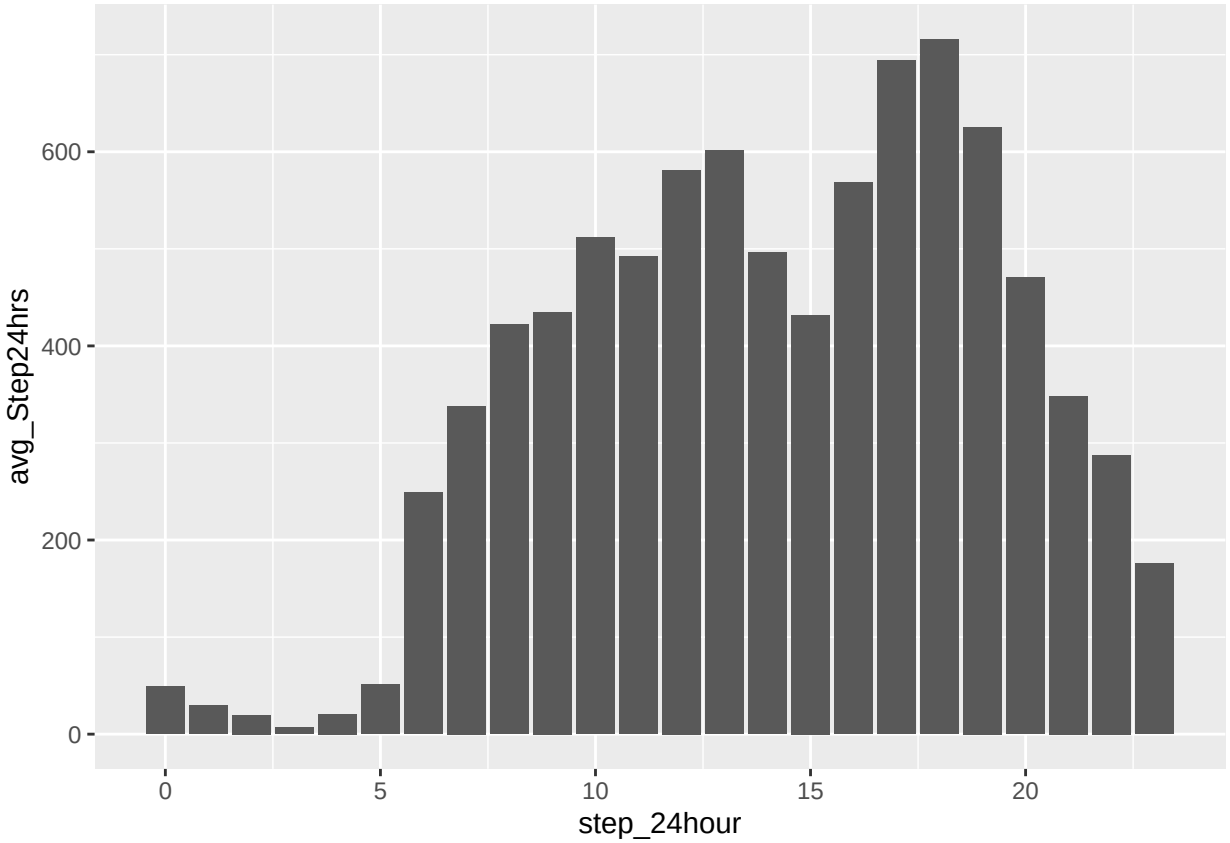


Which hours are likely to exercise?
The most commonly workout period is from 5pm to 7 pm and followed by 12 pm to 1pm. It is very likely that the survey respondents were office workers who did workouts after work and during the lunch break.

```
df_Step_24hrs = df_relp %>% group_by(step_24hour) %>%
  summarise(avg_Step24hrs = mean(StepTotal))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
ggplot(df_Step_24hrs) +
  geom_col(mapping = aes(x=step_24hour, y=avg_Step24hrs))
```



Part F: Proposed Marketing Strategies

What and How to Promote

The survey shows:

- Unable to fall asleep quickly (within 20 minutes) might be a result of stress.
- Healthy sleep (i.e. fall asleep faster and sleep more soundly) may be related to the amount of calories burned and inactive time. In other words, exercise might be a way for healthy sleep.

Implications:
Bellabeat may consider to focus on promoting the benefits of exercise among the public (mental health, reduce stress, help to sleep better) for the following reasons:

- The increase of mental health awareness among the public.
- Both Naomi Osaka and Simone Biles withdrew from competitions because of mental health concern. Therefore, mental health is an important health issue that everybody should take care of.
- The lockdown resulted from pandemic have worsened the situation, sepecially among Gen Z.
- The 2020 Tokyo Olympic and Paralympic Games might motivate people to exercise.

Marketing:

- Work with mental health organizations to promote the importance and benefits of exercise.

- Work with atheletes to promote exercises.
- Adopt the user generated content strategy. Encourage Bellabeat users to create and post their exercise-related videos on Bellabeat Wellness YouTube.
- Promote Bellabeat Leaf in the market because its functions will meet modern life's needs:
  - Alleviate stress: Mindfulness tracking and Stress resistance insights
  - Encourage exercises: Activity tracking and Inactive alerts
  - Monitor sleep pattern: Sleep tracking

Figure 1: Leaf

Who, When and Where to Promote

Two observations regarding exercising patterns:
1. On Saturday, respondents were willing to exercise more, regardless of distance or intensity. As the week passed, respondents started to lose their momentum.

Implications:

- Bellabeat could stress the importance of persistence and consistency.

- Bellabeat's mobile health tracking devices could help people to acheive those objectives effectively.

2. The most common workout period is from 5pm to 7 pm and followed by 12 pm to 1pm. It is very likely that the survey respondents were office workers who did workouts after work and during the lunch break. Implications: Office workers is likely to be the target group for Bellabeat.

Marketing:

- Bellabeat could promote its mobile health tracking devices by launching Bellabeat video ad on the digital screens in the gym during the lunch time and after work hours.

- Bellabeat could organize community and provide health tips, athlete talks, new exercise trends, new products, exercise challenges, etc. to raise loyalty among users.