Reproducible Research: Peer Assessment 1

by Pedro Rebelo
github repo with RMarkdown source code: https://github.com/trashmanp1/RepData_PeerAssessment1

Loading and preprocessing the data

Loading the data is easy, just set the work directory to the directory where data is, unzip the data and load it

```
setwd("/Users/pedrorebelo/RepData_PeerAssessment1")#path to directory where data is
unzip("activity.zip")
dados <- read.csv("activity.csv")
```

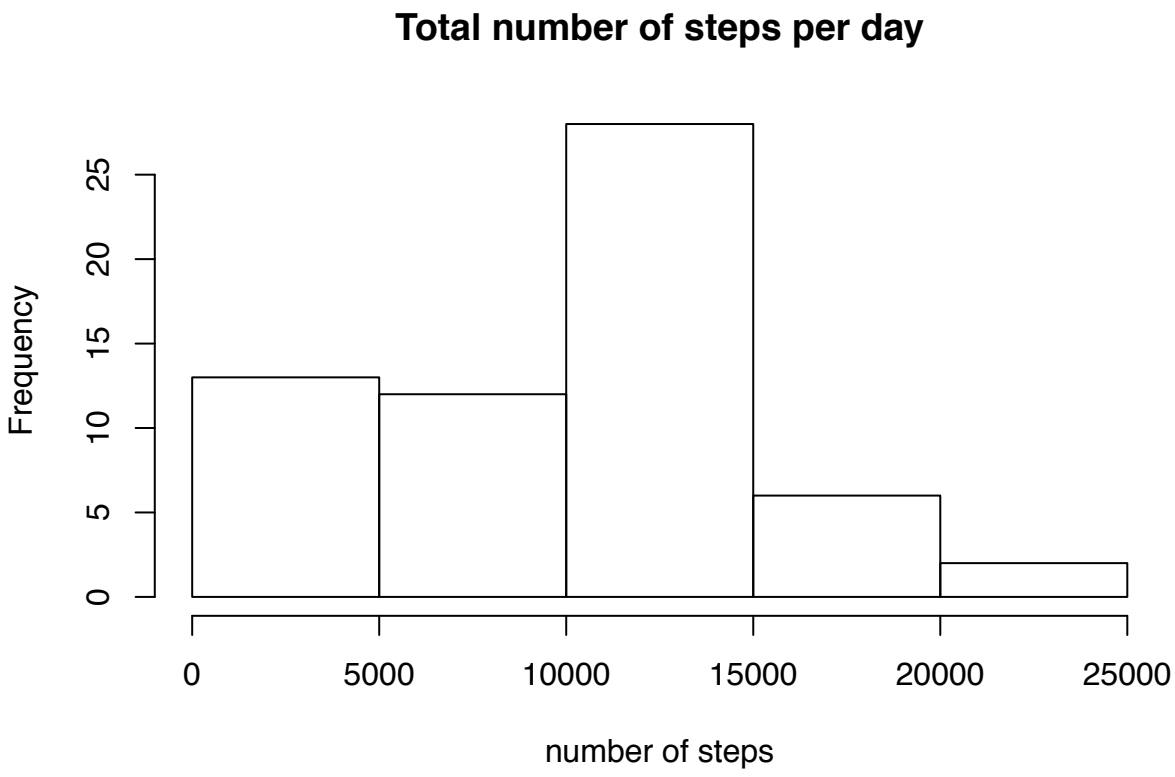
What is mean total number of steps taken per day?

Because in a day there are 288 periods of 5 minutes, we must agregate the result by day, before take the mean and median:

```
passos_por_dia <- aggregate(dados[,"steps"],by=as.list(dados["date"]),FUN=sum,na.rm=TRUE)
names(passos_por_dia) <- c("date","steps")
media<-round(mean(passos_por_dia$steps)) #integer number of steps
mediana<-median(passos_por_dia$steps)
```

The general distribution is given by the hystogram:

```
hist(passos_por_dia$steps,breaks=round(log2(dim(passos_por_dia)[1])+1), #Sturges rule
      xlab="number of steps",
      main="Total number of steps per day")
```

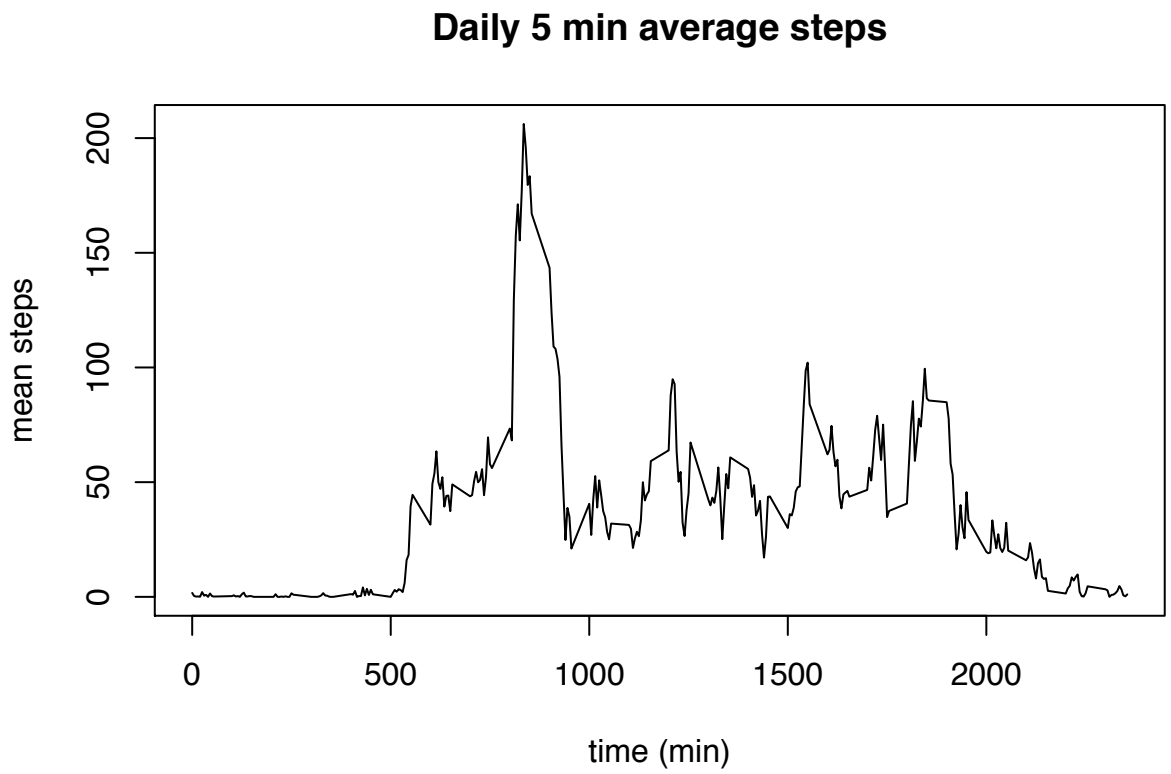


so, the mean is 9354steps per day, and the median is 10395 steps per day.

What is the average daily activity pattern?

We can get the number of steps for the 5 minute intervals averaged over all days

```
passos_por_5_min <- aggregate(dados[,"steps"],by = as.list(dados["interval"]),
                              FUN = mean, na.rm=TRUE)
plot(passos_por_5_min, type = "l",
      xlab="time (min)",
      ylab="mean steps",
      main="Daily 5 min average steps")
```



The 5-minute interval, on average across all the days in the dataset that contains the

```
maximo<-passos_por_5_min$interval[which.max(passos_por_5_min$x)]
```

maximum number of steps begin at 835 min.

Imputing missing values

```
n_falhas<-sum(is.na(dados$steps))
tamanho<-length(dados$steps)
percentagem<-n_falhas/tamanho*100
```

The total sum of missing values is 2304 that corresponds 13.1148% of the total data points.

If we choose the mean to fill the points without registration, we do not change the overall amean, but we will decrease the variance of the overall data. On the other hand if we use the median it will not only decrease the variance but also change the overall mean. If we use zero for the missing values we increse the variance and decrease the overall mean.Because the median of all 5 min intervall is zero, i chose zero. So i will increse the dispersion of data and decrese the mean.

```
dados_martelados<-dados
dados_martelados$steps[is.na(dados$steps)]<-0
summary(dados$steps)
```

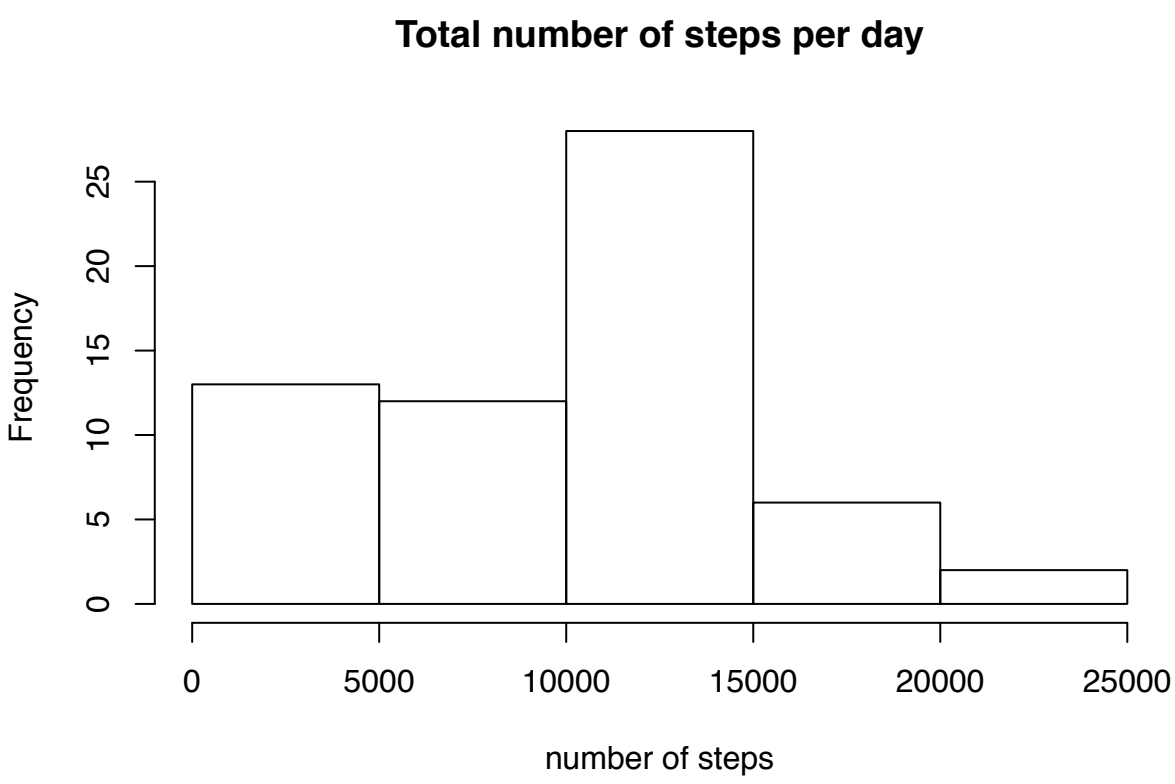
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0     0.0     0.0    37.4    12.0   806.0    2304
```

```
summary(dados_martelados$steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0    32.5     0.0   806.0
```

The histogram of the new data set is:

```
n_passos_por_dia <- aggregate(dados_martelados[,"steps"],by=as.list(dados_martelados["date"]),FUN=sum,na
names(n_passos_por_dia) <- c("date","steps")
n_media<-round(mean(n_passos_por_dia$steps)) #integer number of steps
n_mediana<-median(n_passos_por_dia$steps)
hist(passos_por_dia$steps,breaks=round(log2(dim(n_passos_por_dia)[1])+1), #Sturges rule
      xlab="number of steps",
      main="Total number of steps per day")
```



and the new mean is 9354 steps per day, and the new median is 10395 steps per day. So the results are equal to the previous, because we do not change the sum of the total number of steps by day.

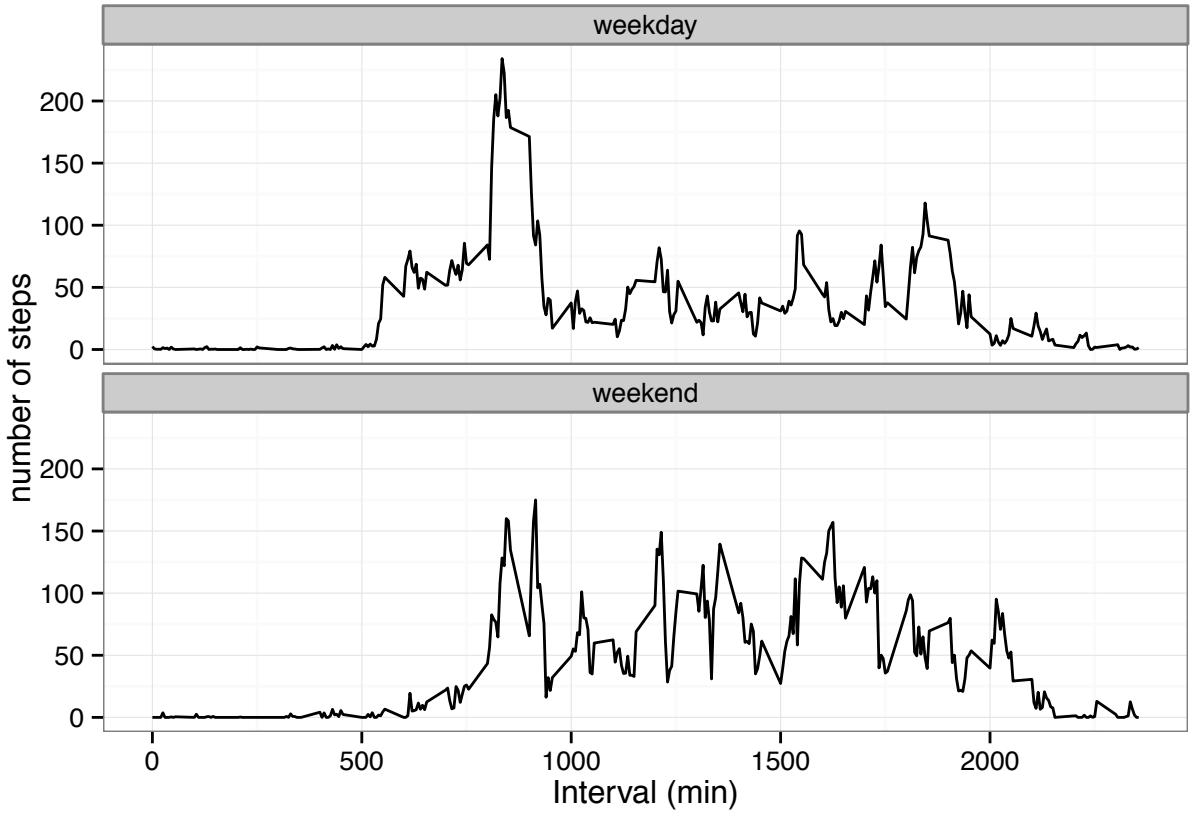
Are there differences in activity patterns between weekdays and weekends?

Based on the next figure, we see that the begining of the day on a weekend have less activity then a weekday but have more activity along the day.

```
#create a new factor from date
wd <- function(date) {
    if (weekdays(as.Date(date)) %in% c("Saturday", "Sunday")) {
        "weekend"
    } else {
        "weekday"
    }
}
dados$wd <- as.factor(sapply(dados$date, wd))

#calculate the 5 min mean by wd
steps_wd<-aggregate(dados[,"steps"],by=as.list(c(dados["wd"],dados["interval"])), FUN=mean,na.rm=TRUE)

#create a ggplot
library(ggplot2)
graph <- ggplot(steps_wd,aes(interval,x))+ theme_bw()
graph <- graph + geom_line() + facet_wrap(~ wd, nrow = 2)
graph <- graph + xlab("Interval (min)")
graph <- graph + ylab("number of steps")
graph
```


weekday
weekend
number of steps
0
50
100
150
200
0
500
1000
1500
2000
Interval (min)