title: "Peer Assessment Project 1" output: md_document ---

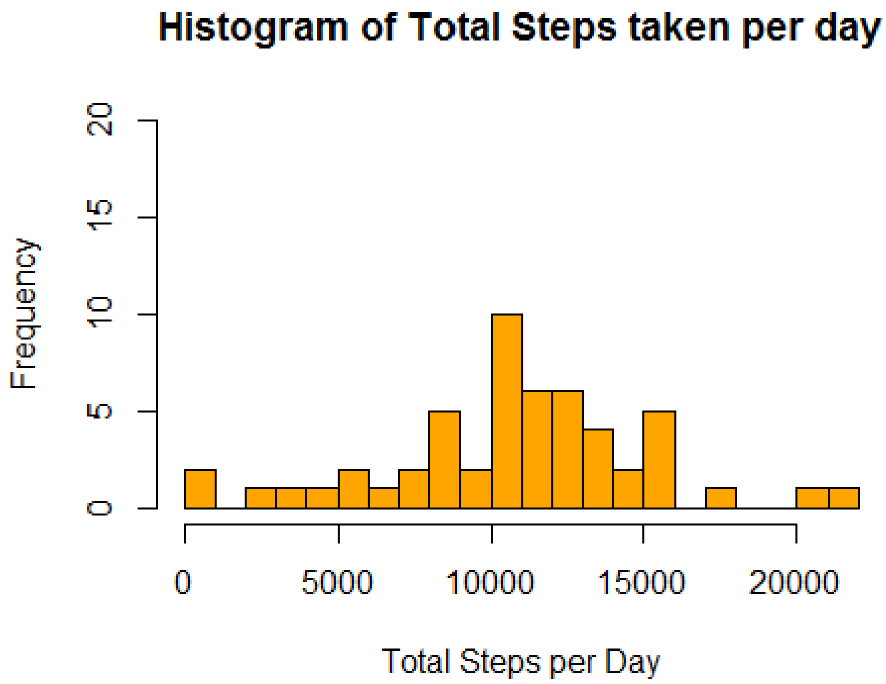
## Loading and preprocessing the data

```
getwd()

## [1] "G:/Data Science Course Materials/Reproducible Research/wk2/New folder
"

setwd("G:/Data Science Course Materials/Reproducible Research/wk2/New Folder"
)
data <- read.csv("activity.csv")
str(data)

## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1
1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...

summary(data)

##      steps                date          interval
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8
##  Median :  0.00   2012-10-03:  288   Median :1177.5
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5
##  3rd Qu.: 12.00   2012-10-05:  288   3rd Qu.:1766.2
##  Max.   :806.00   2012-10-06:  288   Max.   :2355.0
##  NA's   :2304     (Other)   :15840
```

## Mean total number of steps taken

```
# Number of steps taken per day
steps.date <- aggregate(steps ~ date, data = data, FUN = sum)
steps.date

##          date steps
## 1  2012-10-02   126
## 2  2012-10-03 11352
## 3  2012-10-04 12116
## 4  2012-10-05 13294
## 5  2012-10-06 15420
## 6  2012-10-07 11015
## 7  2012-10-09 12811
## 8  2012-10-10  9900
## 9  2012-10-11 10304
## 10 2012-10-12 17382
## 11 2012-10-13 12426
## 12 2012-10-14 15098
## 13 2012-10-15 10139
## 14 2012-10-16 15084
## 15 2012-10-17 13452
## 16 2012-10-18 10056
## 17 2012-10-19 11829
## 18 2012-10-20 10395
## 19 2012-10-21  8821
## 20 2012-10-22 13460
## 21 2012-10-23  8918
## 22 2012-10-24  8355
## 23 2012-10-25  2492
## 24 2012-10-26  6778
## 25 2012-10-27 10119
## 26 2012-10-28 11458
## 27 2012-10-29  5018
## 28 2012-10-30  9819
## 29 2012-10-31 15414
## 30 2012-11-02 10600
## 31 2012-11-03 10571
## 32 2012-11-05 10439
## 33 2012-11-06  8334
## 34 2012-11-07 12883
## 35 2012-11-08  3219
## 36 2012-11-11 12608
## 37 2012-11-12 10765
## 38 2012-11-13  7336
## 39 2012-11-15    41
## 40 2012-11-16  5441
## 41 2012-11-17 14339
## 42 2012-11-18 15110
## 43 2012-11-19  8841
## 44 2012-11-20  4472
## 45 2012-11-21 12787
## 46 2012-11-22 20427
## 47 2012-11-23 21194
## 48 2012-11-24 14478
## 49 2012-11-25 11834
## 50 2012-11-26 11162
## 51 2012-11-27 13646
## 52 2012-11-28 10183
## 53 2012-11-29  7047
```

```
# Histogram of total number of steps taken
hist(steps.date$steps,col="orange",breaks = 20, xlab="Total Steps per Day",
     ylab="Frequency", main="Histogram of Total Steps taken per day", ylim =
c(0, 20))
```



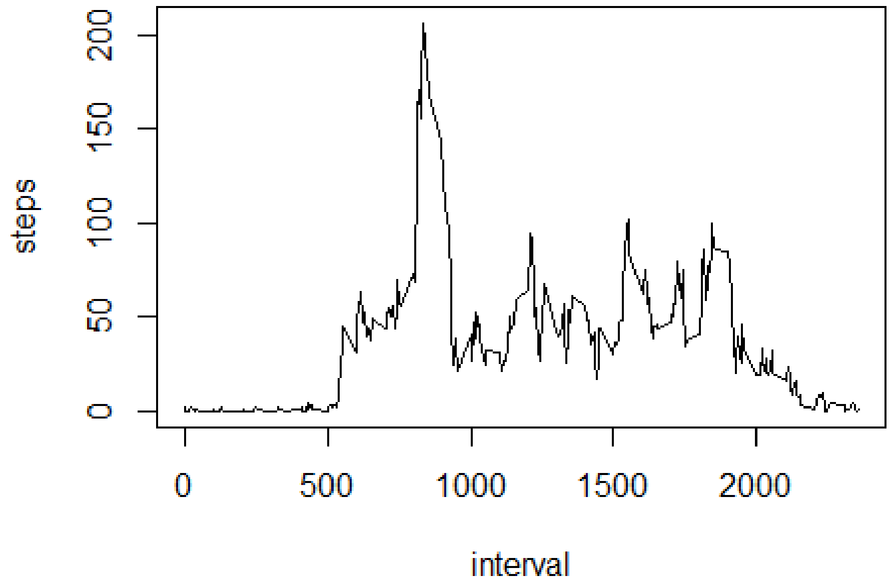
```
# Calculate and report the mean and median total number of steps taken per da
y
mean(steps.date$steps, na.rm = TRUE)

## [1] 10766.19

median(steps.date$steps, na.rm = TRUE)

## [1] 10765
```

Mean and median of the total number of steps taken per day are very close (10766 and 10765 steps, respectively).

## Average daily activity pattern

A time series plot (type = "l") of the 5-minute interval (x-axis) and the average number of steps taken averaged across all days (y-axis).

```
steps.interval <- aggregate(steps ~ interval, data = data, FUN = mean)
plot(steps.interval, type = "l")
```



```
#Which 5-minute interval, on average across all the days in the dataset, cont
ains the maximum number of steps?
steps.interval[which.max(steps.interval$steps), ]

##     interval    steps
## 104      835 206.1698
```

Interval "835" contains on average the maximum number of steps (206 steps).

## Imputing missing values

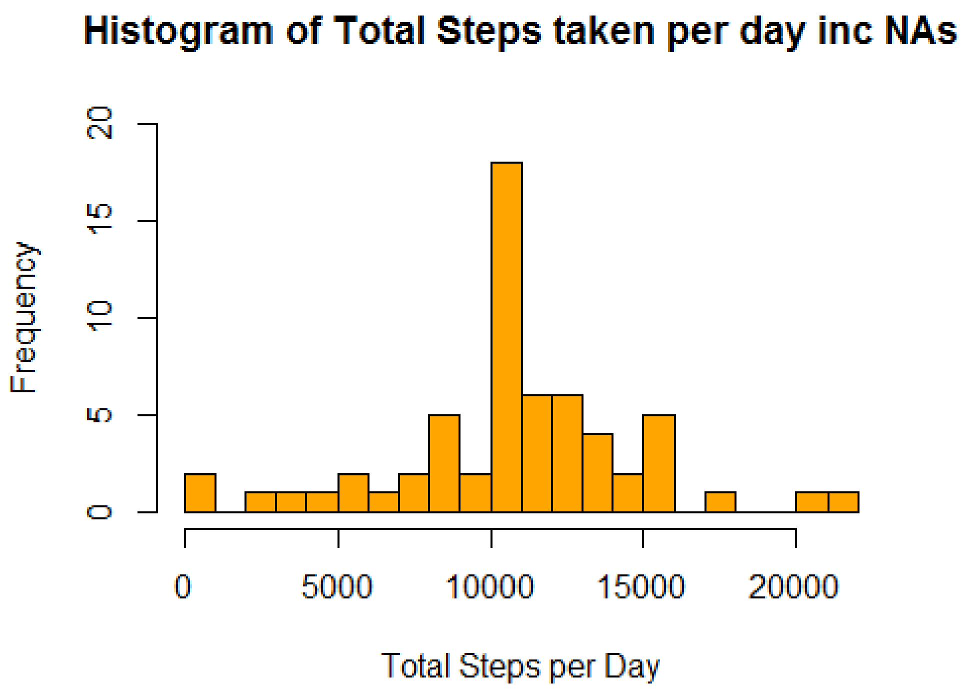
Total number of missing values in the dataset amounts to 2304 (that is almost 13 % of total observations).

```
# Total number of missing values in the dataset
sum(is.na(data))

## [1] 2304

# Strategy to impute missing values - all missing values are filled in with m
ean value for that 5-minute interval
# Replace each missing value with the mean value of its 5-minute interval
fill.value <- function(steps, interval) {
  filled <- NA
  if (!is.na(steps))
    filled <- c(steps) else filled <- (steps.interval[steps.interval$interval
== interval, "steps"])
```

```
    return(filled)
}
filled.data <- data
filled.data$steps <- mapply(fill.value, filled.data$steps, filled.data$interv
al)
```

## Preparing plot after computing missing data

```
steps.date <- aggregate(steps ~ date, data = filled.data, FUN = sum)
hist(steps.date$steps, col="orange",breaks = 20, xlab="Total Steps per Day",
     ylab="Frequency", main="Histogram of Total Steps taken per day inc NAs",
ylim =c(0, 20))
```



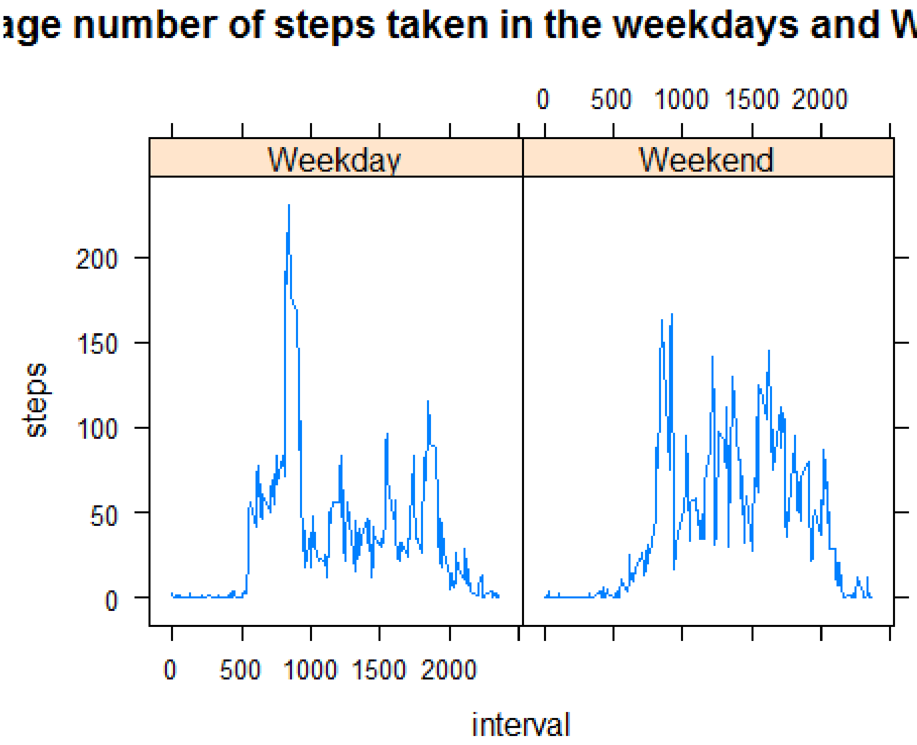
```
#mean and median after computing missing values
mean(steps.date$steps)

## [1] 10766.19

median(steps.date$steps)

## [1] 10766.19
```

Imputing missing values, mean of the total number of steps taken per day remains the same while median marginally and now equals to the mean,compared to estimates from the first part (ingoring missing values).

## Differences in activity patterns between weekdays and weekends

Create a new factor variable in the dataset with two levels -- "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

```
DayGrp <- function(date) {
  if (weekdays(as.Date(date)) %in% c("Saturday", "Sunday")) {
    "Weekend"
  } else {
    "Weekday"
  }
}
filled.data$DayGrp <- as.factor(sapply(filled.data$date, DayGrp))

library(lattice)

steps.type <- aggregate(steps ~ interval + DayGrp, filled.data, mean)
#Plotting the patterns between weekdays and weekend
xyplot(steps ~ interval | DayGrp, data=steps.type, layout=c(2,1), type='l', m
ain = "Average number of steps taken in the weekdays and Weekend")
```


age number of steps taken in the weekdays and Wee
Weekday
Weekend
steps
interval