# Reproducible Research: Project 1

*Mrugank Akarte*

*9 January 2016*

## Synopsis

In this analysis "Activity" data from a personal activity monitoring device was used.This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

## Loading data set

Reading the data from activity.csv file.

```
data_set<-read.csv("activity.csv")
head(data_set)
```

```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

## Total number of steps per day

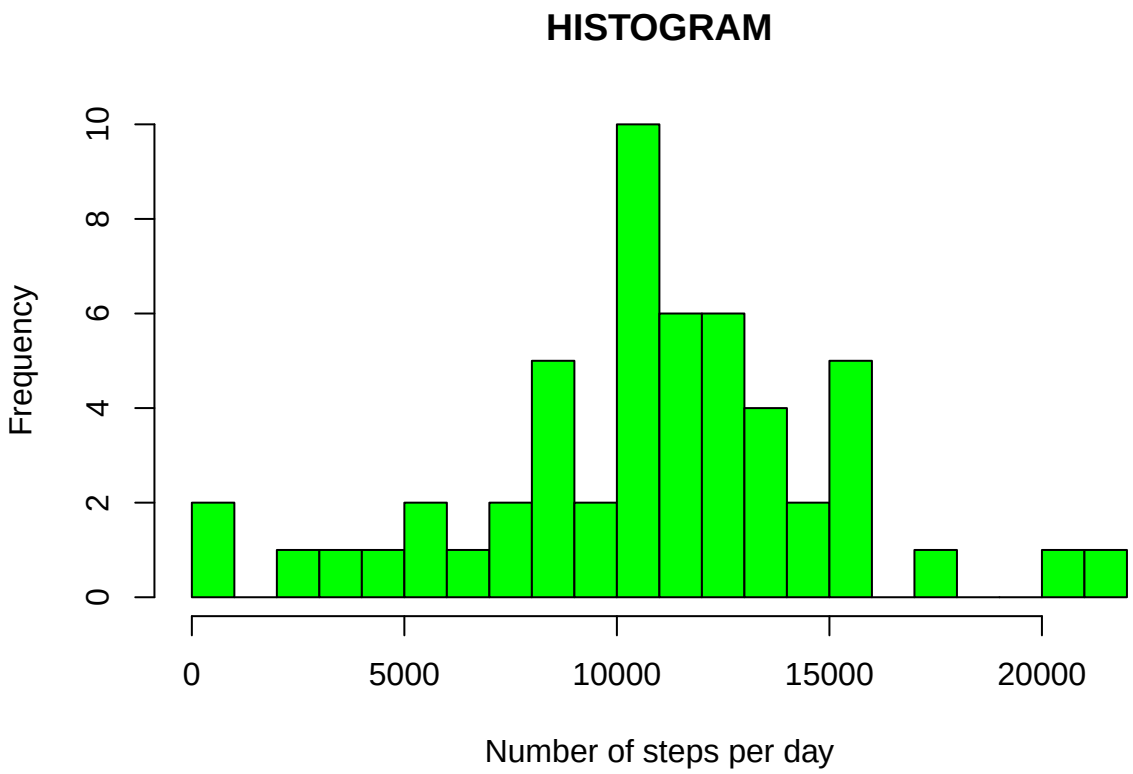
Total number of steps per day are calculated only for those days which are free from NA's.

```
steps_per_day<-aggregate(steps~date, data_set,sum)
head(steps_per_day)
```

```
##         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
```

## Histogram

```
hist(steps_per_day$steps,breaks = 25, col = "green",main = "HISTOGRAM", xlab = "Number of steps per day
```




From the above histogram we can roughly say that maximum number of steps taken per day are in range of 10000-12000.

### Calculating mean and median

```
steps_mean<-mean(steps_per_day$steps)
steps_median<-median(steps_per_day$steps)
steps_mean
```

```
## [1] 10766.19
```

```
steps_median
```

```
## [1] 10765
```

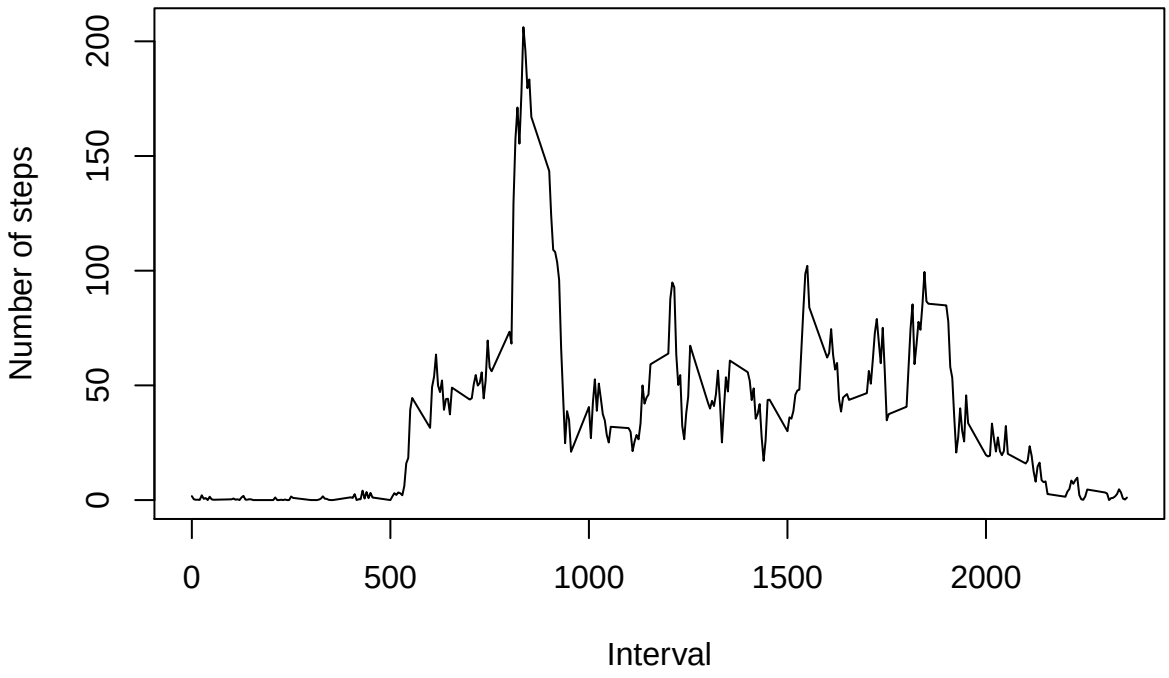
### Time series plot of the average number of steps taken

```
steps_interval<-aggregate(x=list(steps=data_set$steps), by=list(interval=data_set$interval), mean, na.r
head(steps_interval)
```

```
##   interval     steps
```

```
## 1        0 1.7169811
## 2        5 0.3396226
## 3       10 0.1320755
## 4       15 0.1509434
## 5       20 0.0754717
## 6       25 2.0943396
```

```
plot(steps_interval$interval, steps_interval$steps, type = "l",frame.plot = T, xlab = "Interval",ylab =
```




### The 5-minute interval that, on average, contains the maximum number of steps

```
max_steps<-steps_interval[which.max(steps_interval$steps),]
max_steps
```

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

### Code to describe and show a strategy for inputing missing data

```
nonas<-sum(is.na(data_set))
nonas
```

```
## [1] 2304
```

```
fill_value <- function(steps, interval) {
  filled <- NA
  if (!is.na(steps))
    filled <- c(steps)
  else
    filled <- (steps_interval[steps_interval$interval==interval, "steps"])
  return(filled)
}
filled_data <- data_set
filled_data$steps <- mapply(fill_value, filled_data$steps, filled_data$interval)
head(filled_data)
```

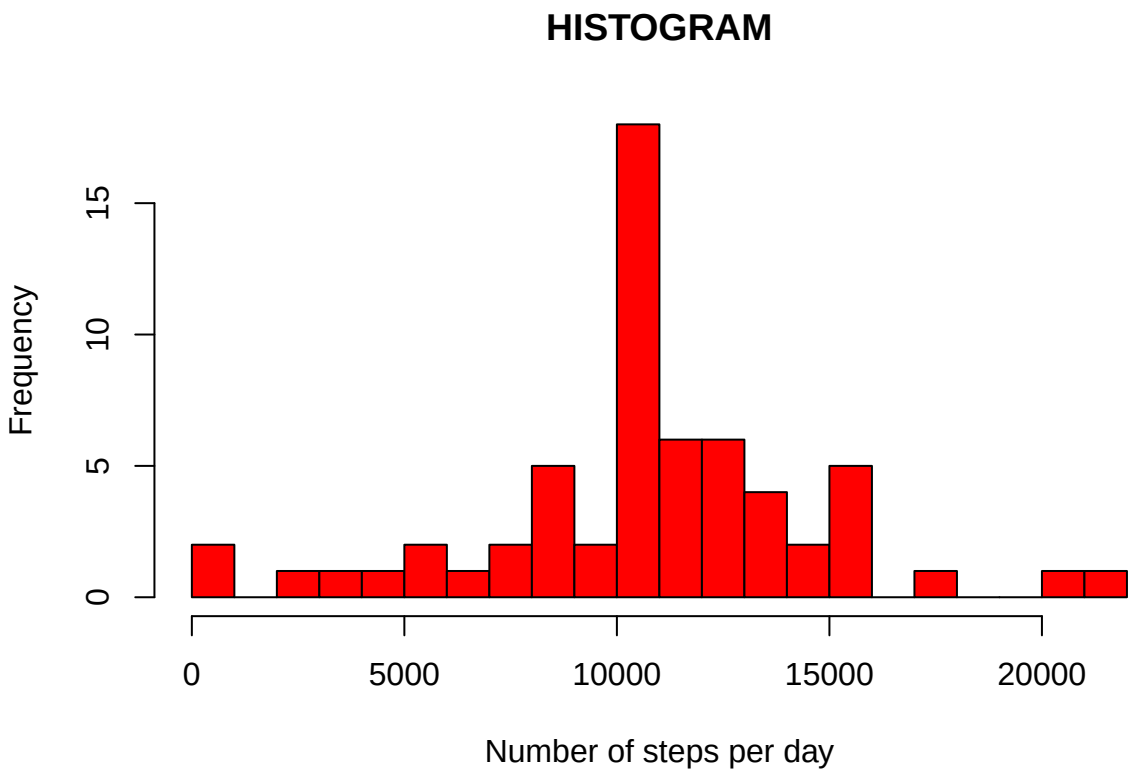
```
##       steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
```

### Histogram of the total number of steps taken each day after missing values are inputed

```
steps_per_day2<-aggregate(steps~date, filled_data, sum)
colnames(steps_per_day2)<-c("date","steps")
head(steps_per_day2)
```

```
##         date    steps
## 1 2012-10-01 10766.19
## 2 2012-10-02   126.00
## 3 2012-10-03 11352.00
## 4 2012-10-04 12116.00
## 5 2012-10-05 13294.00
## 6 2012-10-06 15420.00
```

```
hist(steps_per_day2$steps, breaks = 25, col = "red",main = "HISTOGRAM", xlab = "Number of steps per day
```




We can see change in graph on comparing this graph with the earlier graph of total umber of steps take each day where missing values were ignored.

### New mean and median after missing values are inputed

```
steps_mean2<-mean(steps_per_day2$steps)
steps_mean2
```

```
## [1] 10766.19
```

```
steps_median2<-median(steps_per_day2$steps)

steps_median2
```

```
## [1] 10766.19
```

### Differences in activity patterns between weekdays and weekends

```
filled_data$weekday<-weekdays(as.Date(filled_data$date))
type<-function(date){
  day<-weekdays(date)
```

```
  if(day %in% c("Monday","Tuesday","Wednesday","Thursday","Friday"))
    return("weekday")
  else if(day %in% c("Saturday","Sunday"))
        return("weekend")
  else return("Invalid date!!")

}
filled_data$type<-sapply(as.Date(filled_data$date),FUN = type)

head(filled_data)
```

```
##       steps       date interval weekday    type
## 1 1.7169811 2012-10-01        0  Monday weekday
## 2 0.3396226 2012-10-01        5  Monday weekday
## 3 0.1320755 2012-10-01       10  Monday weekday
## 4 0.1509434 2012-10-01       15  Monday weekday
## 5 0.0754717 2012-10-01       20  Monday weekday
## 6 2.0943396 2012-10-01       25  Monday weekday
```

```
averages<-aggregate(steps~interval+type, filled_data, mean)
library(ggplot2)
ggplot(averages,aes(x=interval,y=steps))+geom_line()+facet_grid(type~.)+ylab("Number of steps")
```