# Reproducible Research :Assignment 1

## Introduction

This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals throughout 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.

### We check if the source file is loaded onto the wd of the reader, if not, we download it and unzip the file:

```
if(!file.exists("activity.csv")) {
        tempfile <- tempfile()
        download.file("http://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip",destf
ile = tempfile)
        unzip(tempfile)
        unlink(tempfile)
}
```

## Load data

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

## Reading the data

Getting idea about how the data looks.

```
str(activity)
```

```
## '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 ...
```

```
head(activity)
```

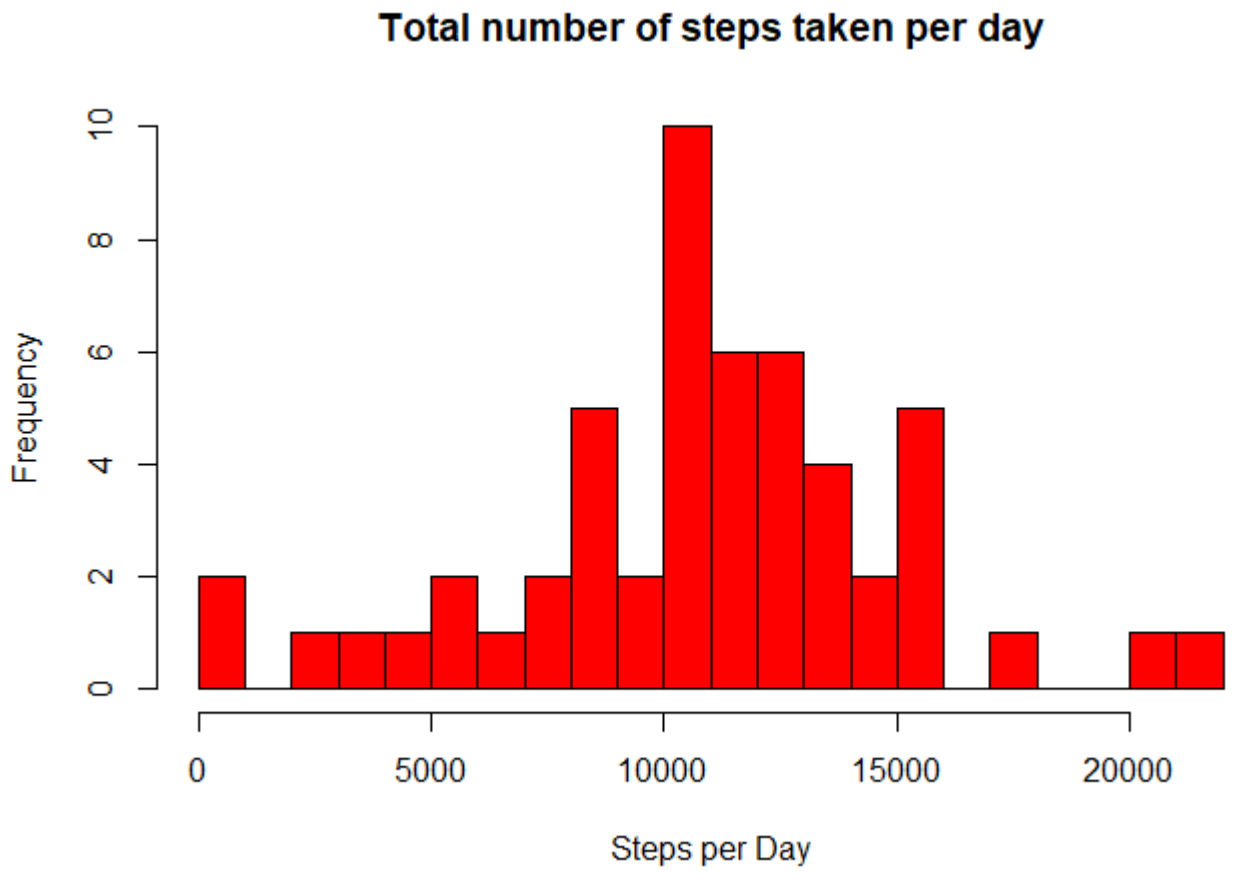
```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

# Total number of steps taken per day

```
activity_steps_day <- aggregate(steps ~ date, data = activity, FUN = sum, na.rm = TRUE)
```

# Histogram of the total number of steps taken each day.

```
hist(activity_steps_day$steps, xlab = "Steps per Day",breaks=25, main = "Total number of steps t
aken per day", col = "red")
```




# Mean and median of the total number of steps taken per day

```
mean_steps <- mean(activity_steps_day$steps)
median_steps <- median(activity_steps_day$steps)

mean_steps <- format(mean_steps,digits=1)
median_steps <- format(median_steps,digits=1)

mean_steps
```

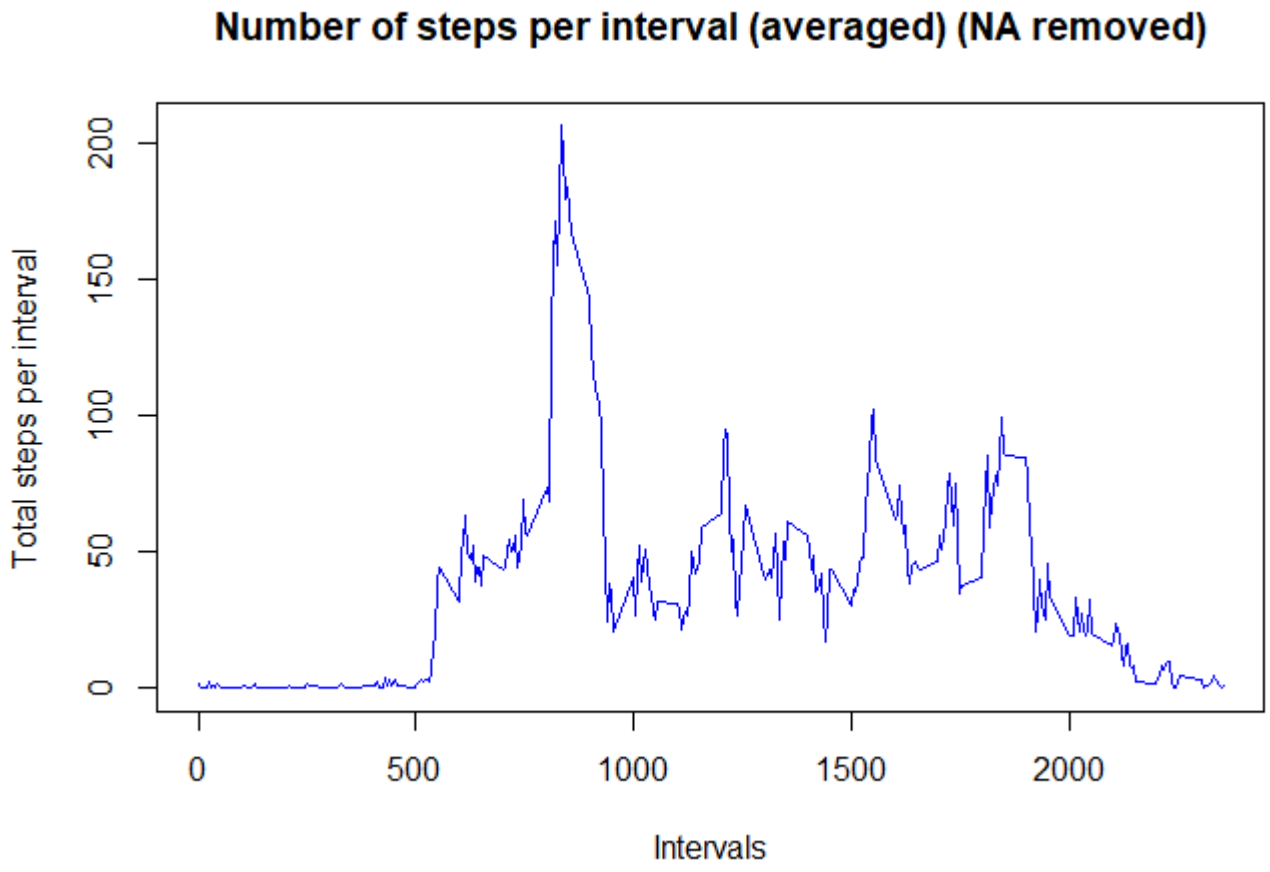
```
## [1] "10766"
```

```
median_steps
```

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

# Time series plot

```
activity_steps_mean <- aggregate(steps ~ interval, data = activity, FUN = mean, na.rm = TRUE)
plot(activity_steps_mean$interval, activity_steps_mean$steps, type = "l", col = "blue", xlab =
"Intervals", ylab = "Total steps per interval", main = "Number of steps per interval (averaged)
 (NA removed)")
```




# Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
max_steps <-max(activity_steps_mean$steps)
max_interval <- activity_steps_mean$interval[which(activity_steps_mean$steps == max_steps)]
max_interval
```

```
## [1] 835
```

```
##The interval 835 has, on average, the highest count of steps, with 206 steps
```

# Calculate total number of missing values in the dataset

```
sum(is.na(activity))
```

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

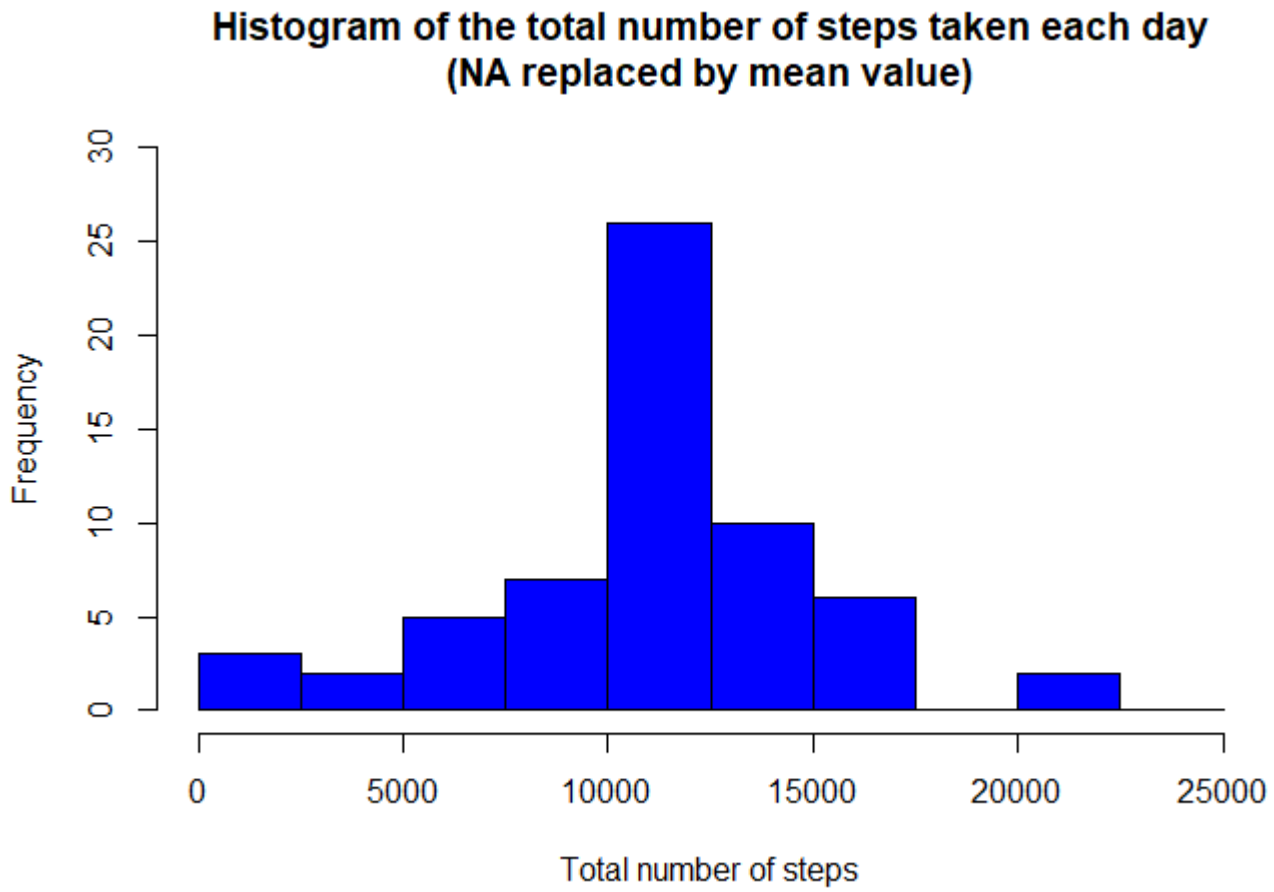
## Let’s take the approach to fill in a missing NA with the average number of steps in the same 5-min interval

## Create a new dataset as the original and use tapply for filling in the missing values with the average number of steps per 5-minute interval

```
data_full <- activity
nas <- is.na(data_full$steps)
avg_interval <- tapply(data_full$steps, data_full$interval, mean, na.rm=TRUE)

data_full$steps[nas] <- avg_interval

sum_data <- aggregate(data_full$steps, by=list(data_full$date), FUN=sum)

names(sum_data) <- c("date", "total")
```

# Compute the histogram of the total number of steps each day

```
hist(sum_data$total,
     breaks=seq(from=0, to=25000, by=2500),
     col="blue",
     xlab="Total number of steps",
     ylim=c(0, 30),
     main="Histogram of the total number of steps taken each day\n(NA replaced by mean value)")
```




# The mean and median are computed like

```
mean(sum_data$total)
```

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

```
median(sum_data$total)
```

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

# Create a new factor variable in the dataset that has NAs been filled with two levels - “weekday” and “weekend” indicating whether a given date is a weekday or weekend day.

```
data_full$date<-as.Date(data_full$date)
data_full$weektype<-ifelse((weekdays(data_full$date) %in% c("Monday","Tuesday","Wednesday","Thursday","Friday")), "weekday","weekend")
```

# Install required packages

```
library(ggplot2)
library(dplyr)
```

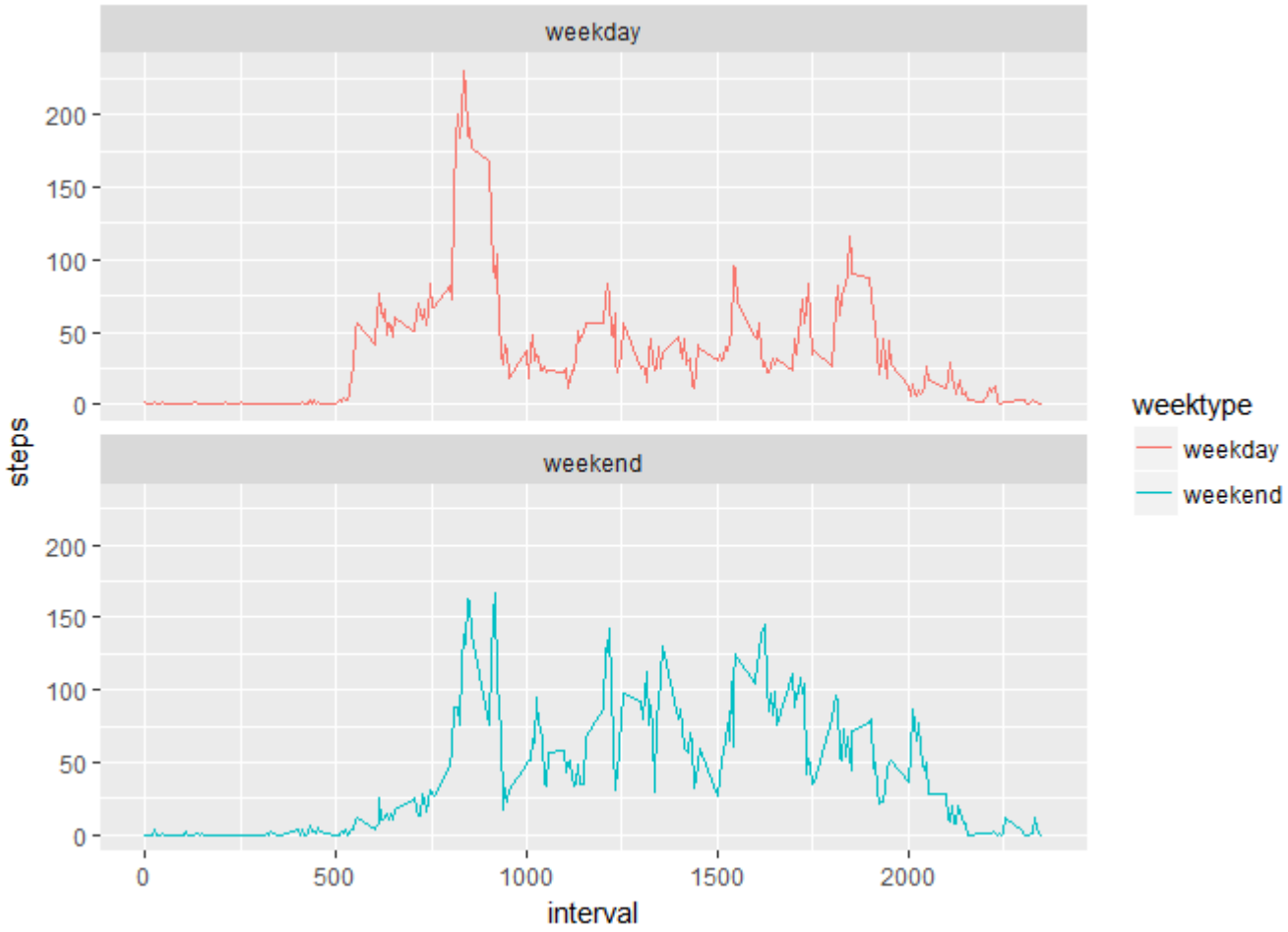
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

# Calculate the average steps in the 5-minute interval and use ggplot for making the time series of the 5-minute interval for weekday and weekend, and compare the average steps:

```
interval_full <- data_full %>%
 group_by(interval, weektype) %>%
 summarise(steps = mean(steps))

##Here is the Panel plot comparing the average number of steps taken per 5-minute interval acros
s weekdays and weekends


s <- ggplot(interval_full, aes(x=interval, y=steps, color = weektype)) +
  geom_line() +
  facet_wrap(~weektype,ncol=1,nrow=2)
print(s)
```




**From the two plots it seems that the test object is more active earlier in the day during weekdays compared to weekends, but more active throughout the weekends compared with weekdays (probably because the oject is working during the weekdays, hence moving less during the day).