# Reproducible Research/Course Project 1

*Zaher Haydar*

*June 23, 2017*

## Loading and preprocessing the data

1. First, we'll download, unzip and load the data into R using read.csv, then store in data.frame **activity**

```
fileUrl <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
if (!file.exists("repdata%2Fdata%2Factivity.zip")) download.file(fileUrl, destfile="./repdata%2F
data%2Factivity.zip")
if (!exists("activity")) activity <- read.csv("activity.csv")
```

2. Next, we'll make sure necessary libraries are loaded. No extra data pre-processing nor transformation wil be done at this stage.

```
require(ggplot2)
require(dplyr)
require(lattice)
```

## Looking at number of steps per day: Total, Mean, Median, etc.

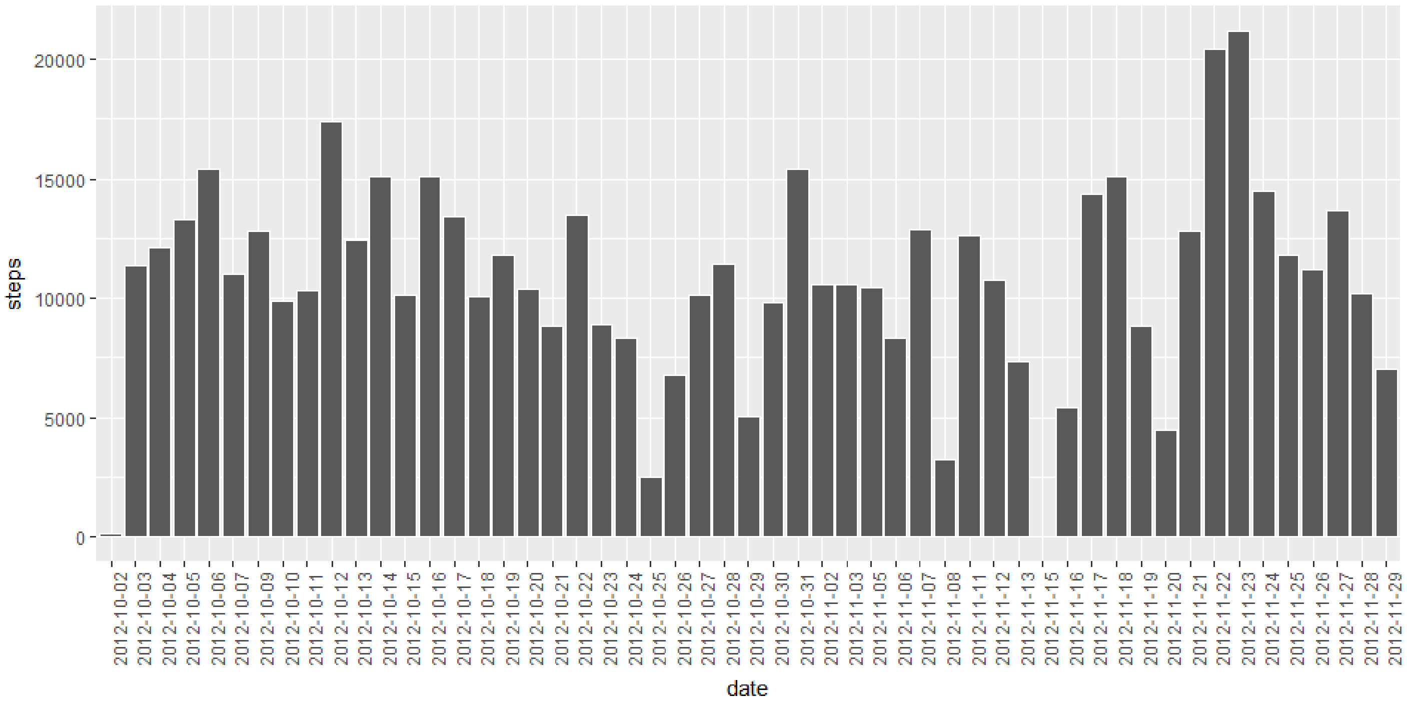
For now, we'll ignore the missing values in the dataset.

1. Let's calculate the total number of steps taken per day

```
stepsPerDay <- aggregate(steps ~ date, activity, sum)
```

1. Then, let's plot the total number of steps taken each day as a histogram chart

```
ggplot(stepsPerDay, aes(x=date, y=steps)) + geom_bar(stat="identity", colour="white") + theme(ax
is.text.x = element_text(angle = 90))
```



3. Next, let’s look at the mean and median values of the total number of steps taken per day. We’ll look at the comprehensive output from R **summary** function

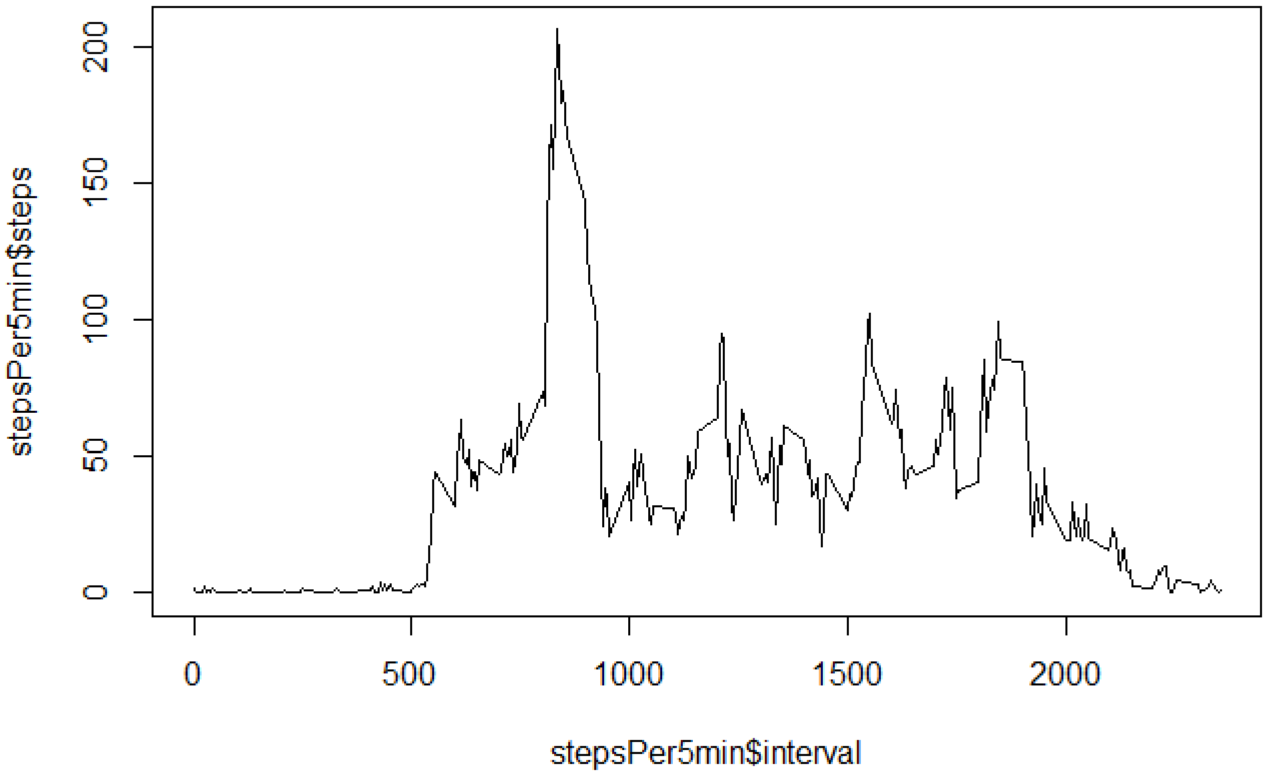
```
summary(stepsPerDay)
```

```
##          date        steps      
##  2012-10-02: 1   Min.   :   41  
##  2012-10-03: 1   1st Qu.: 8841  
##  2012-10-04: 1   Median :10765  
##  2012-10-05: 1   Mean   :10766  
##  2012-10-06: 1   3rd Qu.:13294  
##  2012-10-07: 1   Max.   :21194  
##  (Other)   :47
```

# Analyzing average daily activity pattern

1. We want to look at the average number of steps taken, averaged across all days per each 5-minute interval. We’ll use a time series plot to show that view

```
stepsPer5min <- aggregate(steps ~ interval, activity, mean)
plot(stepsPer5min$interval, stepsPer5min$steps, type="l")
```



2. Specifically, let's single out Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps. R **filter** function will give us the interval number plus the value of the maximum steps per interval

```
filter(stepsPer5min, stepsPer5min$steps==max(stepsPer5min$steps))
```

```
##   interval    steps
## 1      835 206.1698
```

# Imputing missing values

There are a number of days/intervals where there are missing values (NA's). The presence of missing days may introduce bias into some calculations or summaries of the data.

1. Let's calculate the total number of missing values (NA's) in the dataset

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

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

2. We want to fill in all of the missing values in the dataset. We'll use the averages across all days to fill the 5-minute intervals where values are missing

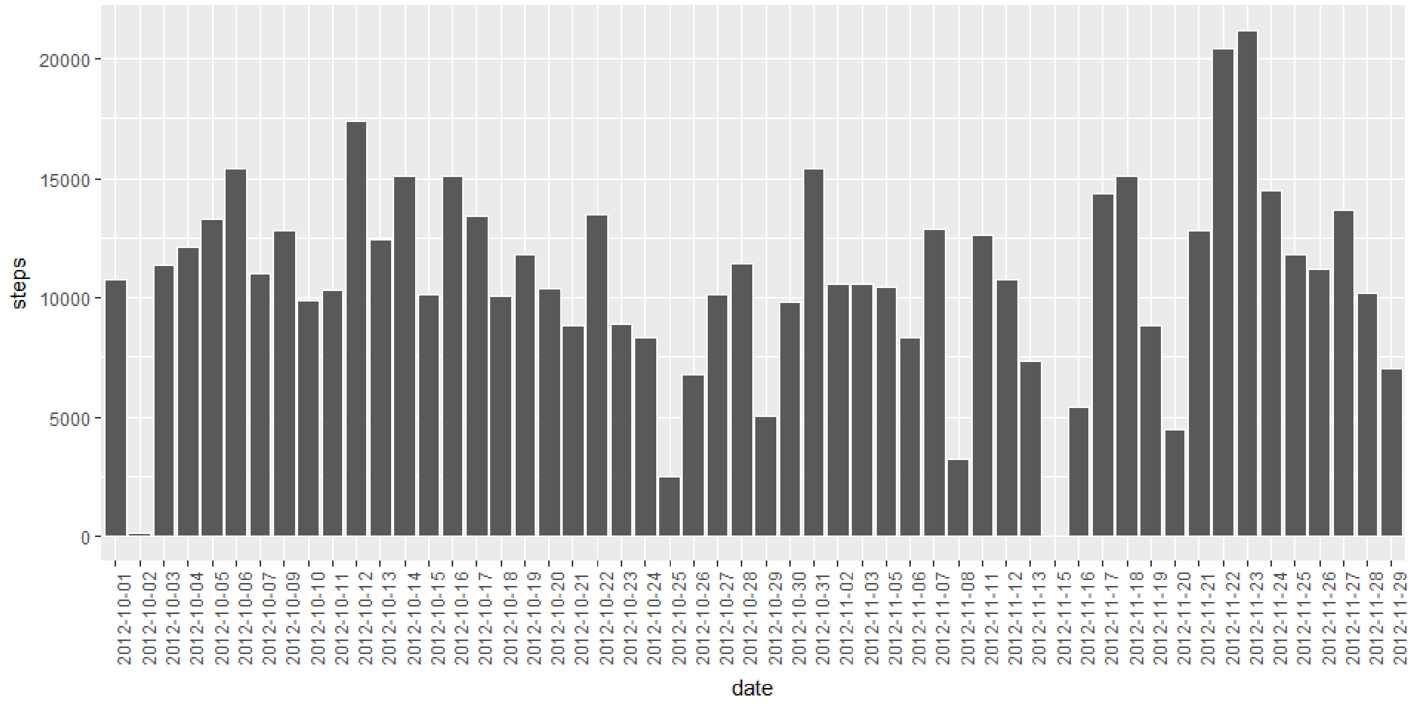
3. A new dataset **activity2** as a copy of the orginal **activit** dataset but with the missing data filled in.

```
activity2 <- activity
i <- 1
for (i in 1:288) {

     if (is.na(activity2[i,1])) activity2[i,1] <- stepsPer5min[i,2]
     i <- i+1
}
```

4. Based on the imputed dataset, we’ll look at the histogram of the total number of steps taken each day as well as the mean and median total number of steps taken per day.

```
stepsPerDay2 <- aggregate(steps ~ date, activity2, sum)
ggplot(stepsPerDay2, aes(x=date, y=steps)) + geom_bar(stat="identity", colour="white") + theme(a
xis.text.x = element_text(angle = 90))
```



```
summary(stepsPerDay2)
```

```
##          date        steps      
##  2012-10-01: 1   Min.   :   41  
##  2012-10-02: 1   1st Qu.: 8860  
##  2012-10-03: 1   Median :10766  
##  2012-10-04: 1   Mean   :10766  
##  2012-10-05: 1   3rd Qu.:13191  
##  2012-10-06: 1   Max.   :21194  
##  (Other)   :48
```

We notice that these values are essentially the same as the ones obtained from the original dataset. This means the impact of imputing missing data on the estimates of the total daily number of steps in this case was *Negligibe*

# Differences in activity patterns between weekdays and weekends

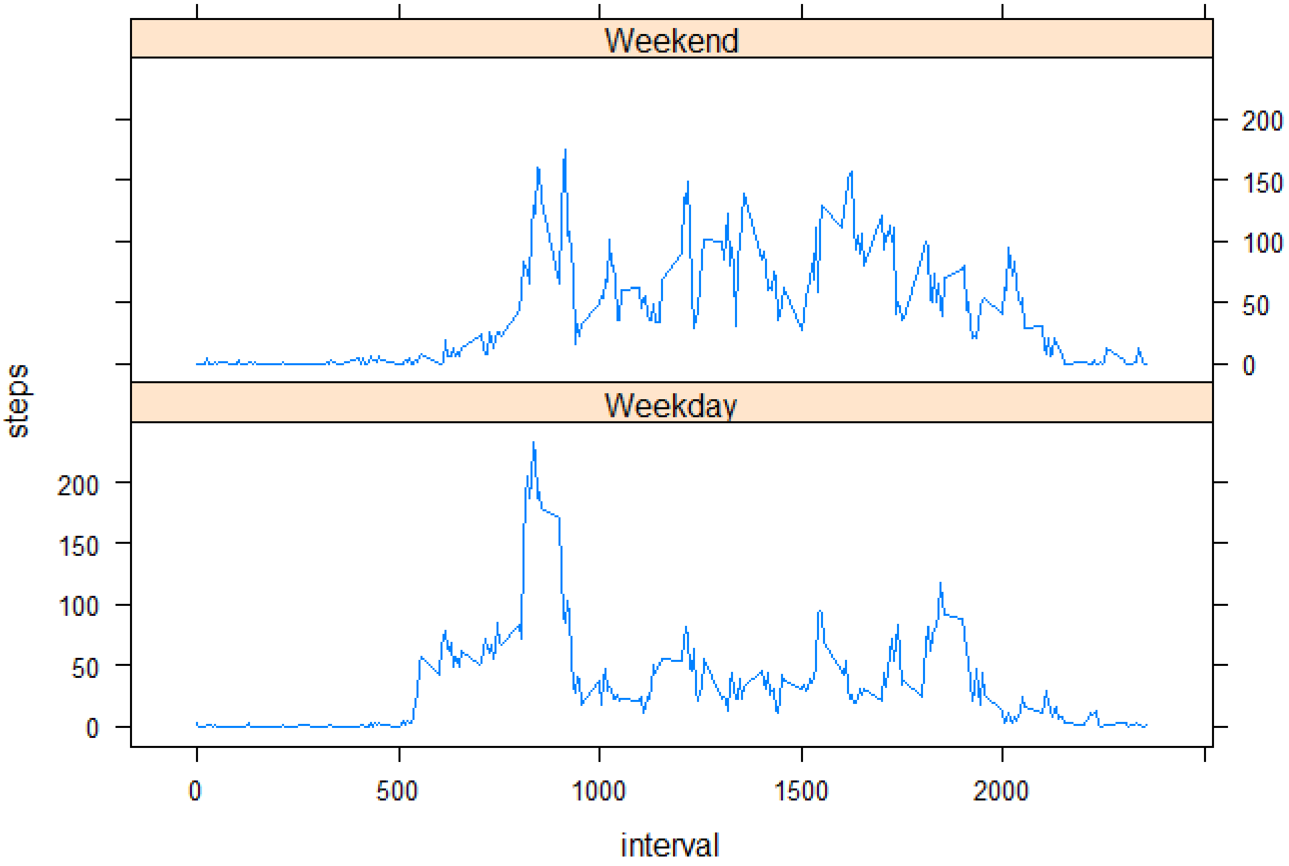
For this part, we’ll work on the dataset with the filled-in missing values.

1. First, we’ll add a new factor variable to **activity2** with two levels - “Weekday” and “Weekend” indicating whether a given date is a weekday or weekend day.

```
activity2$date <- as.Date(as.character(activity2$date))
activity2$day <- ifelse((weekdays(activity2$date) %in% c("Saturday","Sunday")),"Weekend", "Weekd
ay")
activity2$day <- as.factor(activity2$day)
```

2. Finally, we’ll make a panel plot containing a time series plot of the 5-minute intervaland the average number of steps taken, averaged across all weekday days or weekend days.

```
stepsPer5min2 <- aggregate(steps ~ interval+day, activity2, mean)
xyplot(steps ~ interval | day, data = stepsPer5min2, type='l', layout = c(1, 2))
```


Weekend
Weekday
steps
interval