

Reproducible Research : Peer Assessment 1

```
library(lattice)
```

Loading and Preprocessing the Data

Load csv data **activity.csv** and convert dates to **R date class**

```
activityData <- read.csv("activity.csv")
activityData$date <- as.Date(activityData$date, format = "%Y-%m-%d")
head(activityData)
```

```
##  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
```

What is the mean total number of steps taken per day?

1. Split data by date and calculate steps in each one.
2. Plot the histogram.
3. Calculate mean and median.

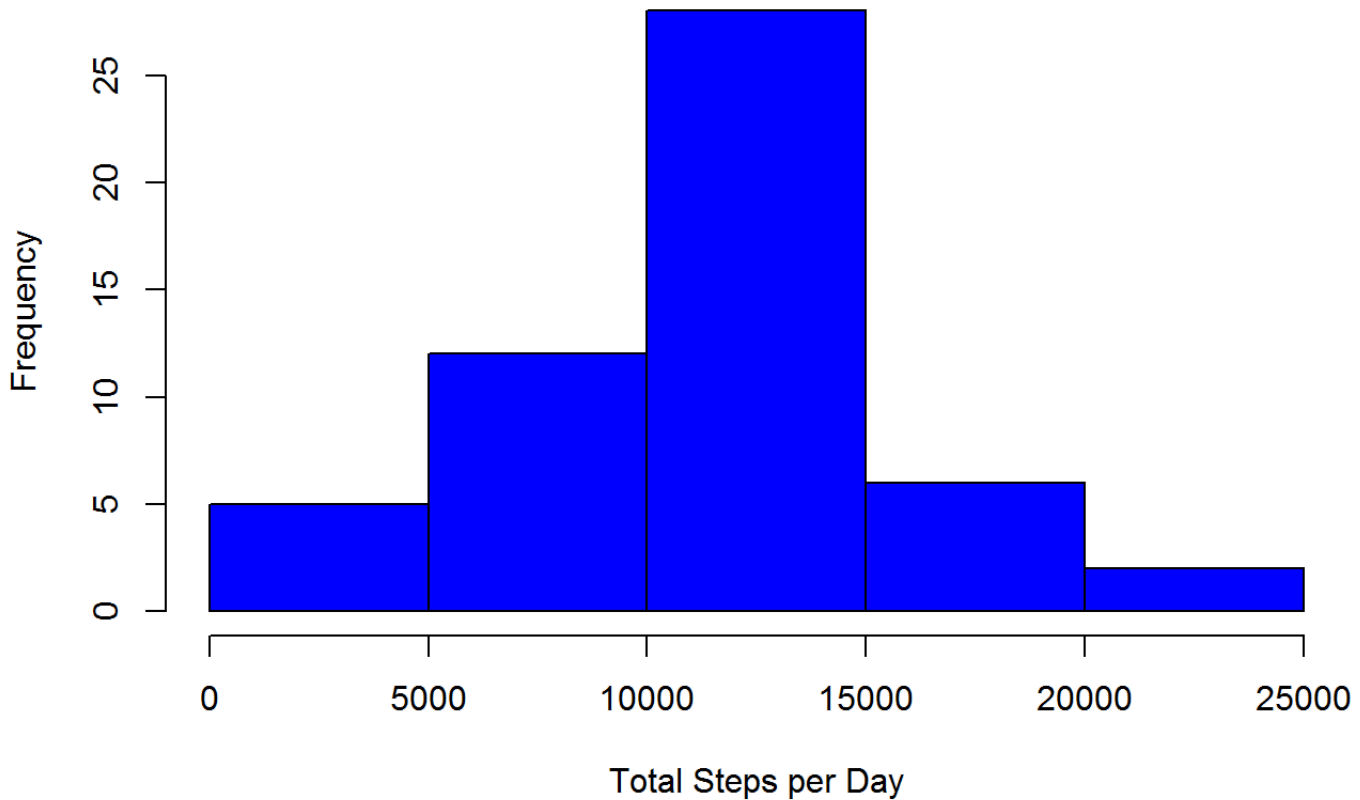
Split the data

```
totalStepsByDay <- tapply(activityData$steps, activityData$date, sum)
```

Plot the histogram..

```
hist(totalStepsByDay, col="blue", xlab="Total Steps per Day",
     ylab="Frequency", main="Histogram of Total Steps taken per day")
```

Histogram of Total Steps taken per day



Calculate the mean of the total number of steps taken per day

```
mean(totalStepsByDay, na.rm = TRUE)
```

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

Calculate the median of the total number of steps taken per day

```
median(totalStepsByDay, na.rm = TRUE)
```

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

What is the average daily activity pattern?

1. Split data by intervals.
2. Calculate average of steps.
3. Make a time series plot of the interval (x-axis) and the average number of steps taken (y-axis)
4. Find which 5-minute interval contains the max # of steps.

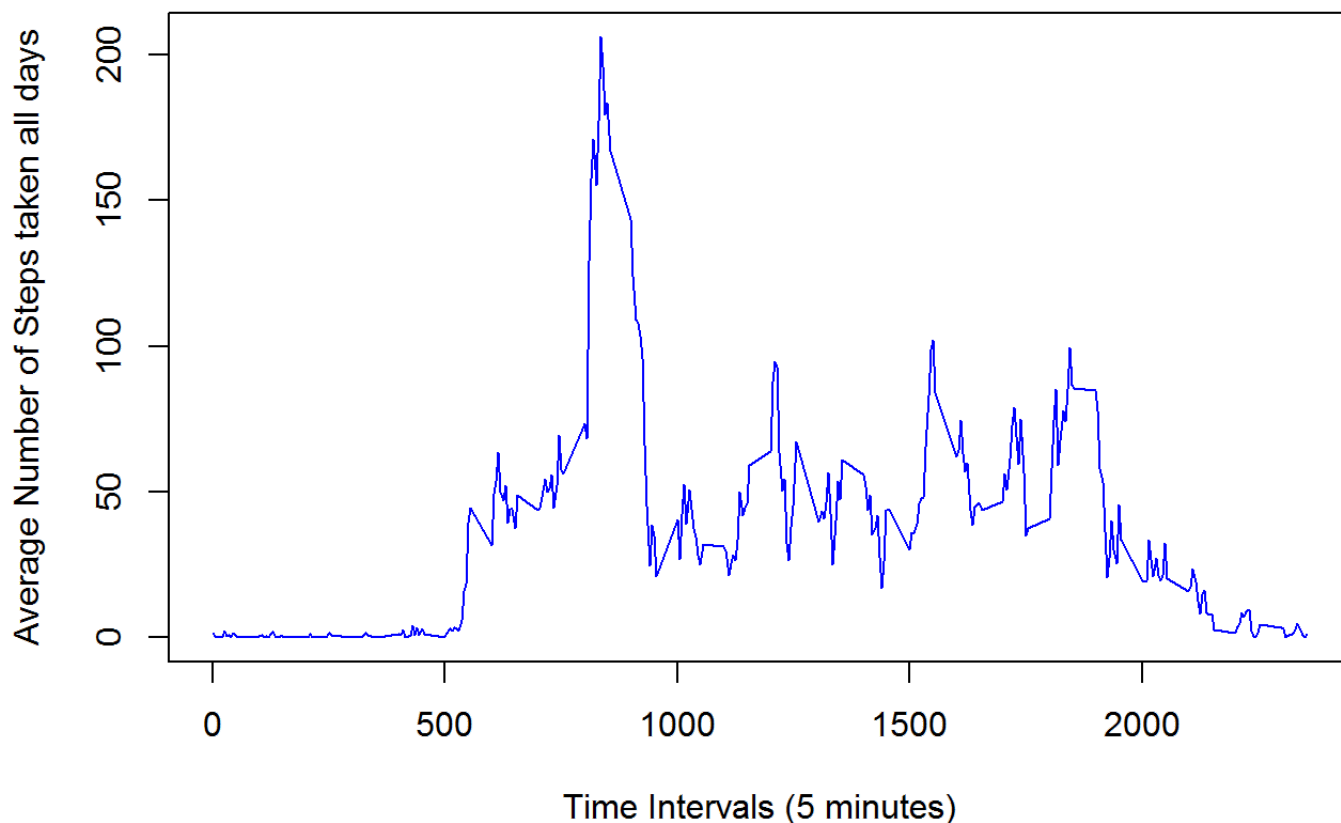
calculate mean steps by interval

```
meanSteps <- tapply(activityData$steps, activityData$interval, mean, na.rm = TRUE)
```

Time series plot of the interval (x-axis) and the average number of steps taken (y-axis)

```
plot(row.names(meanSteps),meanSteps, type = "l", xlab="Time Intervals (5 minutes)", ylab="Average Number of Steps taken all days", main = "Average Steps taken 5 minute Intervals" , col = "blue")
```

Average Steps taken 5 minute Intervals



Find the interval that contains max average number of steps

```
MaxIntervalNum <- which.max(meanSteps)
names(MaxIntervalNum)
```

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

Imputing missing values

1. Calculate and report the total number of missing values in the dataset.
2. Devise a strategy for filling in all of the missing values in the dataset.
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.
4. Make a histogram of the total number of steps taken each day and Calculate the mean and median total number of steps taken per day.

Compute the number of missing values:

```
totalNaVals <- sum(is.na(activityData))
totalNaVals
```

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

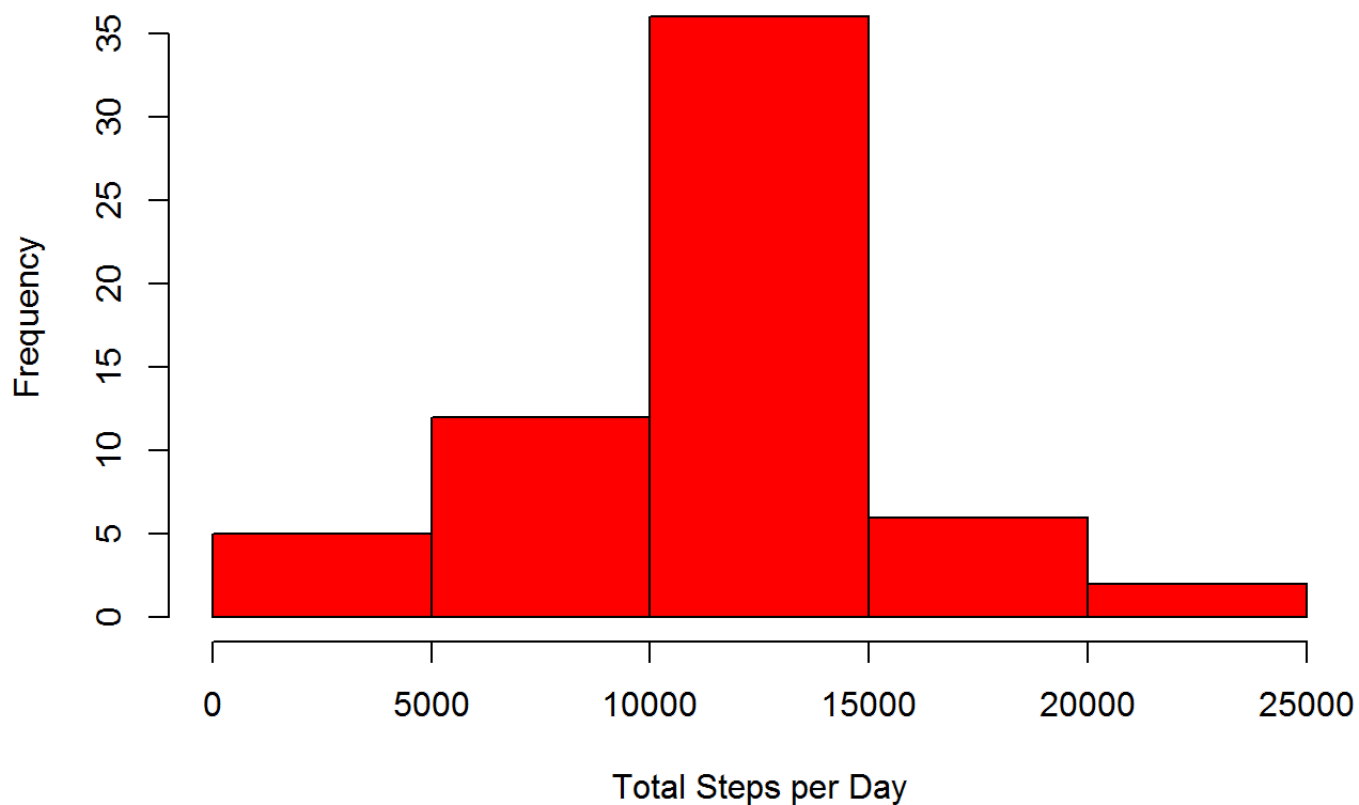
Fill in missing values using the **average interval value across all days**

```
naIndices <- which(is.na(activityData))
imputedValues <- meanSteps[as.character(activityData[naIndices,3])]
names(imputedValues) <- naIndices
for (i in naIndices) {
  activityData$steps[i] = imputedValues[as.character(i)]
}
sum(is.na(activityData))
```

```
## [1] 0
```

```
totalSteps <- tapply(activityData$steps, activityData$date, sum)
hist(totalSteps, col="red", xlab="Total Steps per Day",
      ylab="Frequency", main="Histogram of Total Steps taken per day")
```

Histogram of Total Steps taken per day

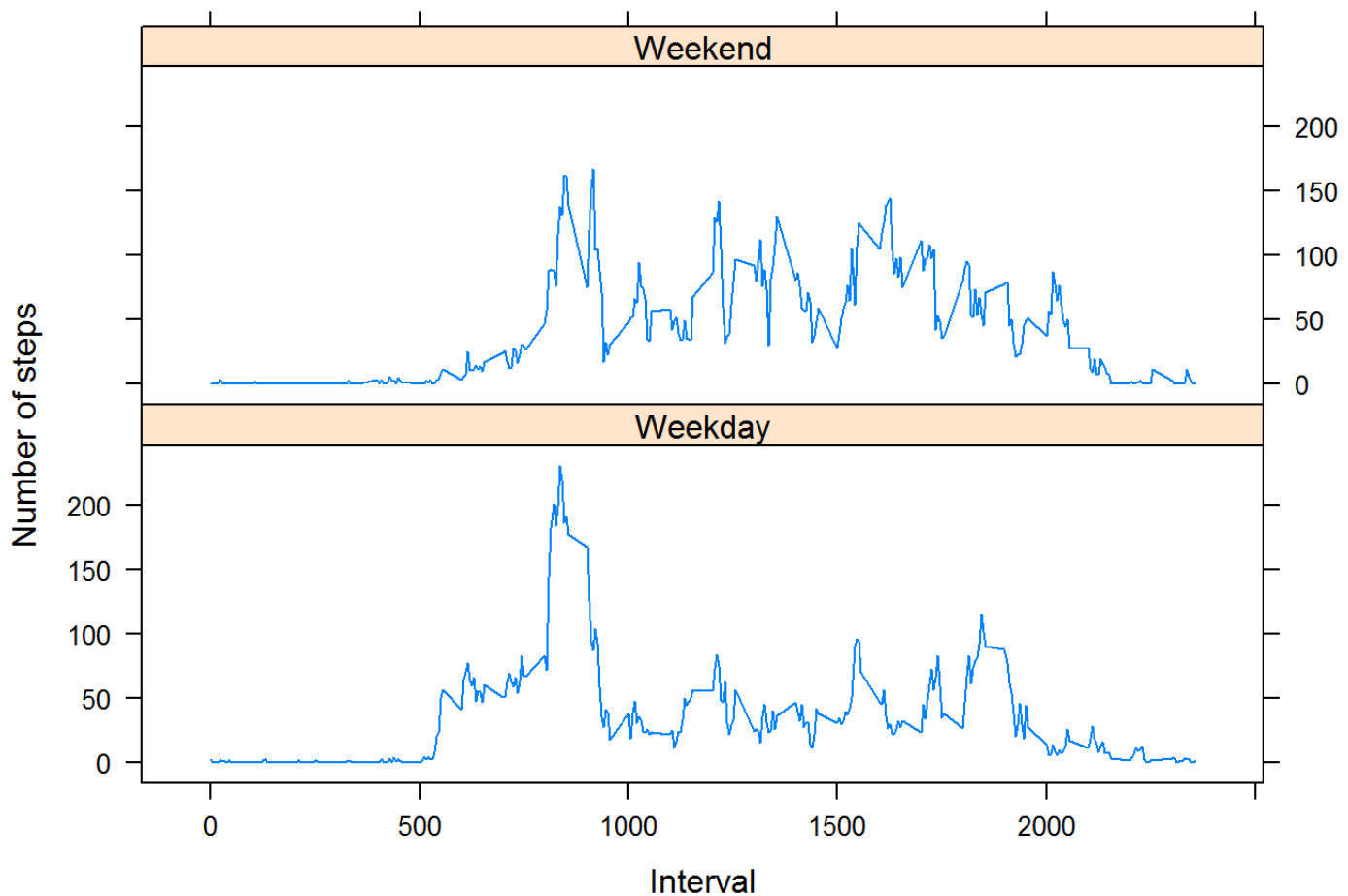


Are there differences in activity patterns between weekdays and weekends?

1. 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
2. Make a panel plot containing a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

```
days <- weekdays(activityData$date)
activityData$day_type <- ifelse(days == "Saturday" | days == "Sunday", "Weekend", "Weekday")
meanSteps <- aggregate(activityData$steps,
                        by=list(activityData$interval, activityData$day_type), mean)
names(meanSteps) <- c("interval", "day_type", "steps")

xyplot(steps~interval | day_type, meanSteps, type="l", layout=c(1,2), xlab="Interval", ylab = "Number of steps")
```



Compute the mean, median, max and min of the steps across all intervals and days by Weekdays and Weekends.

```
tapply(meanSteps$steps,meanSteps$day_type,  
       function (x) { c(MINIMUM=min(x),MEAN=mean(x),  
                        MEDIAN=median(x),MAXIMUM=max(x))})
```

```
## $Weekday
```

```
##   MINIMUM      MEAN    MEDIAN  MAXIMUM  
##  0.00000  35.61058  25.80314 230.37820  
##
```

```
## $Weekend
```

```
##   MINIMUM      MEAN    MEDIAN  MAXIMUM  
##  0.00000  42.36640  32.33962 166.63915
```