RR-Project1

Wednesday, March 11, 2015

This report contains the analysis of the activity monitoring data specified for this assignment.

First Load the data:

```
steps=read.csv("activity.csv")
summary(steps)
```

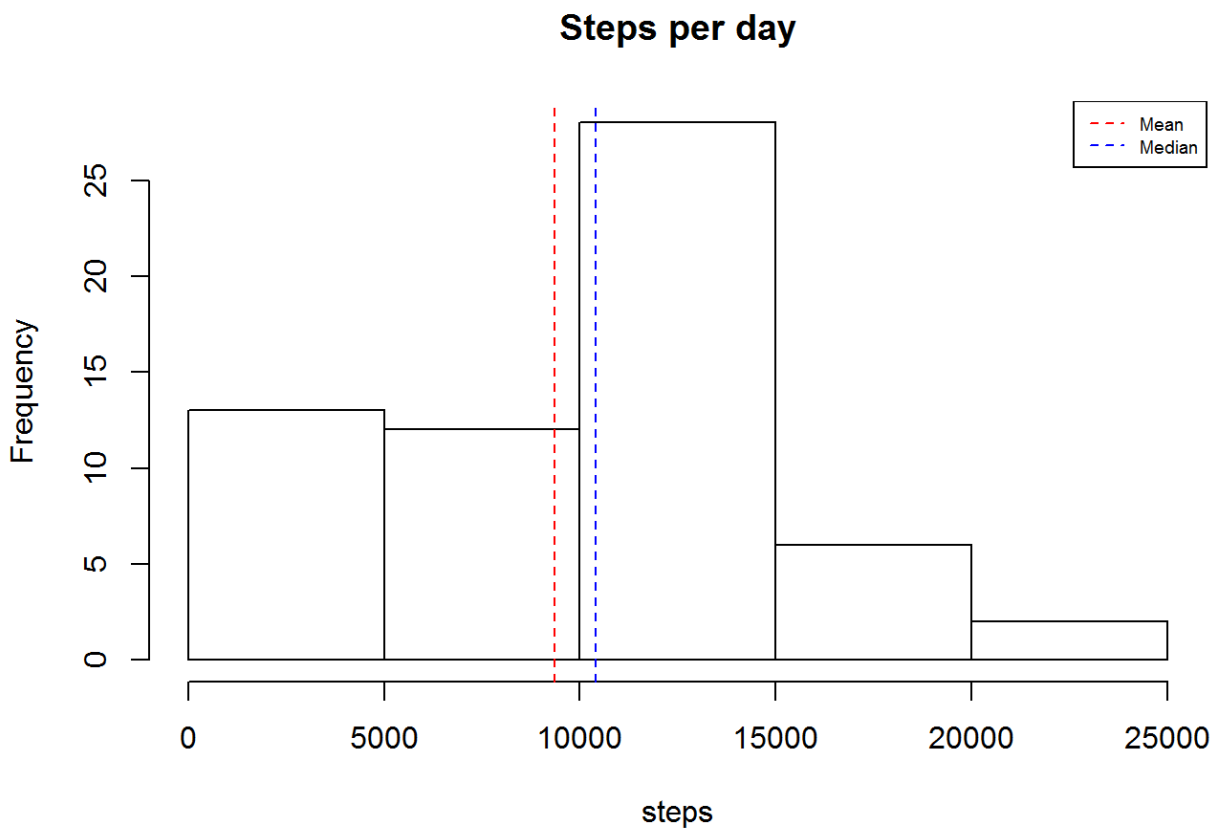
```
##      steps                date          interval     
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0  
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8  
##  Median :  0.00   2012-10-03:  288   Median :1177.5  
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5  
##  3rd Qu.: 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
```

It can be seen that the “steps” column contains some NAs. As instructed we will be imputing the NA’s later on in the assignment and ignoring them for the time-being.

Analysis of Total Steps Per Day

Calculating Total Nr. of Steps taken per day

```
#steps_pd=rowsum(steps$steps,steps$date,na.rm=T)
#hist(steps_pd)
#meanSteps = mean(steps_pd)
#medianSteps = median(steps_pd)
#abline(v=meanSteps, col="red", lty=2)
#abline(v=medianSteps, col="blue", lty=2)
#legend("topright", legend=c("Mean", "Median"), lty=c(2,2), col=c("red","blu
e"), cex=0.6)
myHist <- function (df){
  steps_pd=rowsum(df$steps,df$date,na.rm=T)
  hist(steps_pd, main="Steps per day", xlab="steps")
  meanSteps = mean(steps_pd)
  medianSteps = median(steps_pd)
  abline(v=meanSteps, col="red", lty=2)
  abline(v=medianSteps, col="blue", lty=2)
  legend("topright", legend=c("Mean", "Median"), lty=c(2,2), col=c("red","blu
e"), cex=0.6)
  c(meanSteps,medianSteps)
}

summ_steps = myHist(steps)
```



Total nr. of steps taken per day has **mean of 9354.2295082** steps per day and median of **1.039510^{4}**

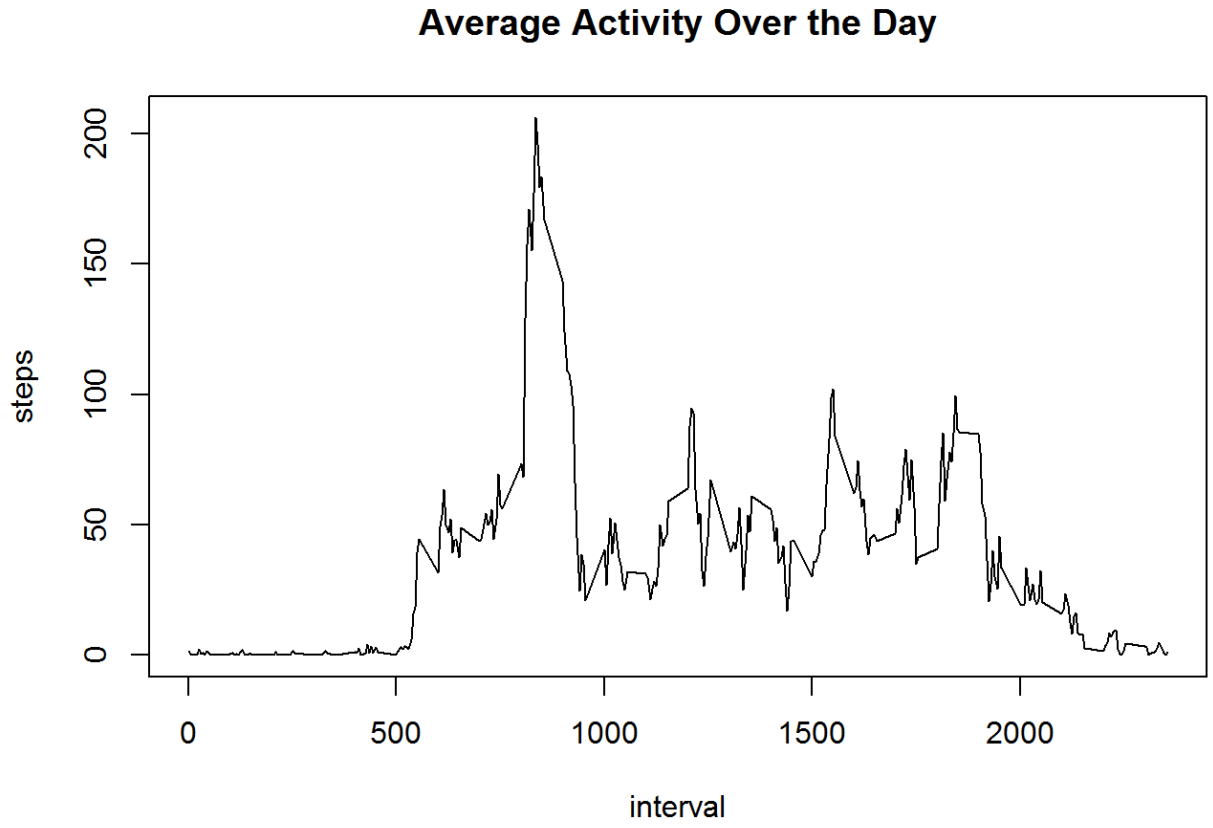
steps per day

Analysing Average Daily Activity Pattern

```
#Following library needed for by function.
library(taRifx)
#create smaller dataframe without the NAs
small_steps=steps[!is.na(steps$steps),]

#Average the NAs over the column interval
steps_avg_int = as.data.frame(by(small_steps$steps, small_steps$interval, mea
n))

#The following did not work. Hence used the by function from taRifx
#steps_avg_int=aggregate(x=c(steps$steps), by=as.list(steps$interval), FUN=mea
n, na.action=na.omit)

#Plot the timeseries of average steps over intervals
```

The following plot show the average step pattern along a series of 5 min intervals in a day

```
plot(steps_avg_int$IDX1,steps_avg_int$value, type="l", xlab="interval", ylab="s
teps", main="Average Activity Over the Day")
```




```
#determine the interval containing the max average steps
max_steps = max(steps_avg_int$value)
max_steps_interval = steps_avg_int[steps_avg_int$value==max_steps,]$IDX1
```

The **835 th interval** contains the maximum nr. of steps in a day.

Strategy for Imputing Missing Values

```
#Num oF NAs
na_steps = nrow(steps[is.na(steps$steps),])
```

The total nr. of missing values in the dataset are **2304**

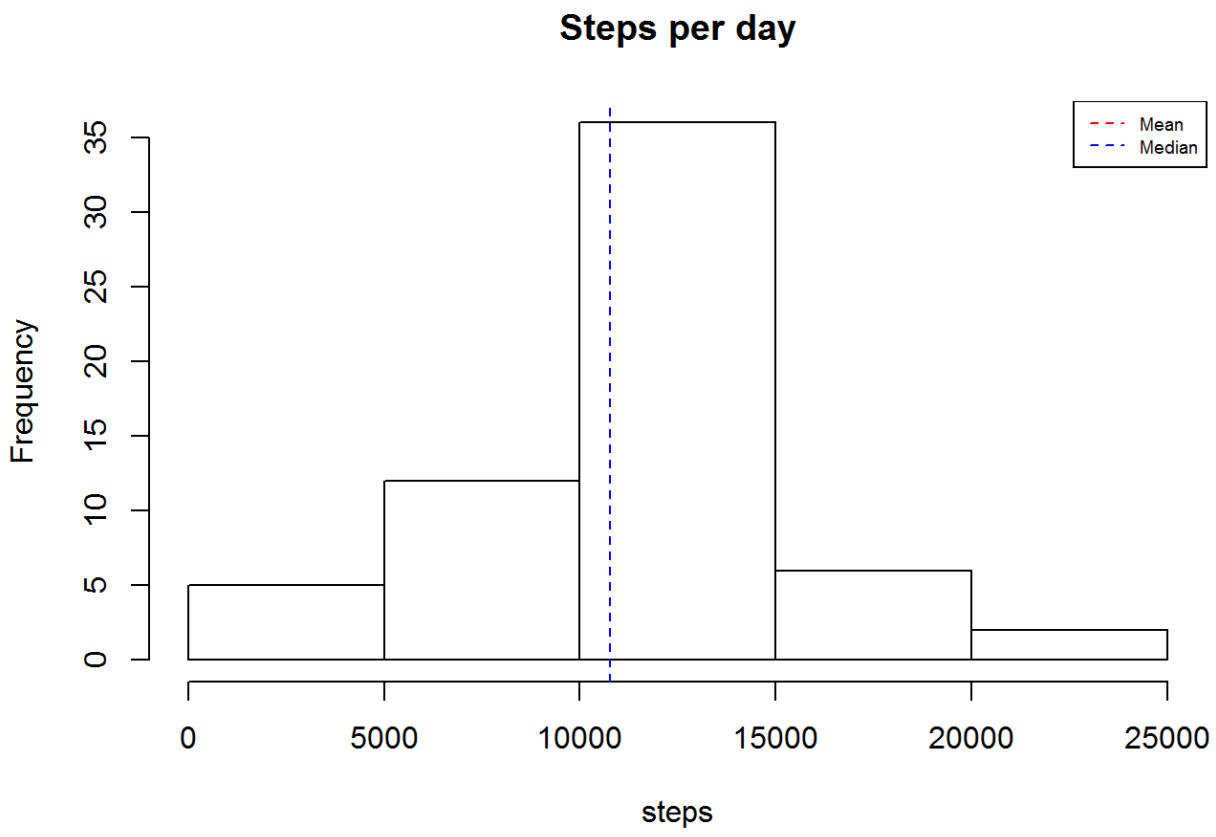
We will fill in all of the missing values in the dataset with the mean for that 5-minute interval. We will not use the day mean because the days having NAs do not have any data for that day.

Filling in the missing values:

```
nasteps=is.na(steps$steps)
n=length(nasteps)
for  (i in 1:n)
{
  if (nasteps[i])
  {
    #Get the row corresponding to the interval of the step with missing value
    intmeanrow = steps_avg_int[steps_avg_int$IDX1 == steps[i,]$interval,]
    #Doublechecking that there is an interval average available.
    if (nrow(intmeanrow) != 0 ) {steps[i,]$steps = intmeanrow$value}
  }
}
```

Replotting the histogram

```
new_summ_steps=myHist(steps)
```



After imputing the values the mean now shifts to 1.076618910^{4} from 9354.2295082 and the median shifts to 1.076618910^{4} from 1.039510^{4}

Imputing the data **increases** the steps per day

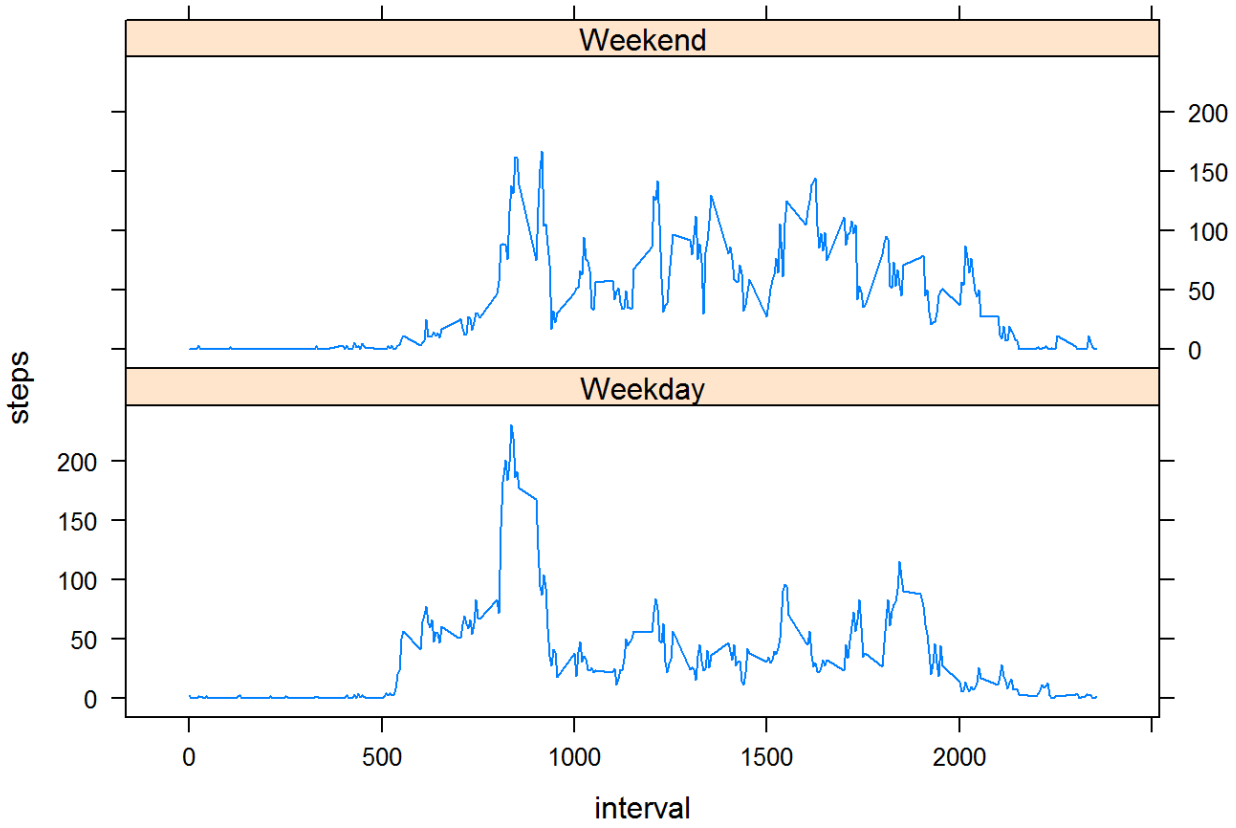
Differences in Activity Pattern between weekdays and weekends

Augmenting the data with factors Weekend / Weekday and computing the average steps per interval for weekdays and weekends.

```
weekdays=weekdays(as.Date(steps$date,"%Y-%m-%d"))
daytype=c()
n=length(weekdays)
for  (i in 1:n)
{
  daytype= c(daytype, if ((weekdays[i]=="Saturday") | (weekdays[i]=="Sunda
y") ) "Weekend" else "Weekday")
}
new_steps=data.frame(steps,daytype)

#Separating the data by weekdays and weekends
weekends=(new_steps$daytype == "Weekend")
weekday_data=new_steps[!weekends,]
weekend_data=new_steps[weekends,]

#Computing the average steps per interval averaged over days for weekdays and w
eekends
steps_avg_int_wd = as.data.frame(by(weekday_data$steps, weekday_data$interval,
mean))
steps_avg_int_we = as.data.frame(by(weekend_data$steps, weekend_data$interval,
mean))

new_data_wd = data.frame(int=steps_avg_int_wd[,1],steps=steps_avg_int_wd[,2],ty
pe="Weekday")
new_data_we = data.frame(int=steps_avg_int_we[,1],steps=steps_avg_int_we[,2],ty
pe="Weekend")

#combining the data
new_data_tot = rbind(new_data_wd,new_data_we)
```

The following plot shows the activity over intervals for weekdays and weekends separately.

```
library(lattice)
xyplot(steps ~ int | type, data=new_data_tot, type="l", layout=c(1,2), xlab="in
terval")
```



It is seen that in a certain morning interval the weekday activity is more than the weekend, but during most of the remainder part of the the weekend activity is more than the weekday activity.