Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

```
#unzip("activity.zip")
df <- read.csv("activity.csv")
library(stringr)
df$interval <- str_pad(df$interval, 4, pad="0")
#df$interval <- strptime(df$interval, "%H%M")
```

What is mean total number of steps taken per day?

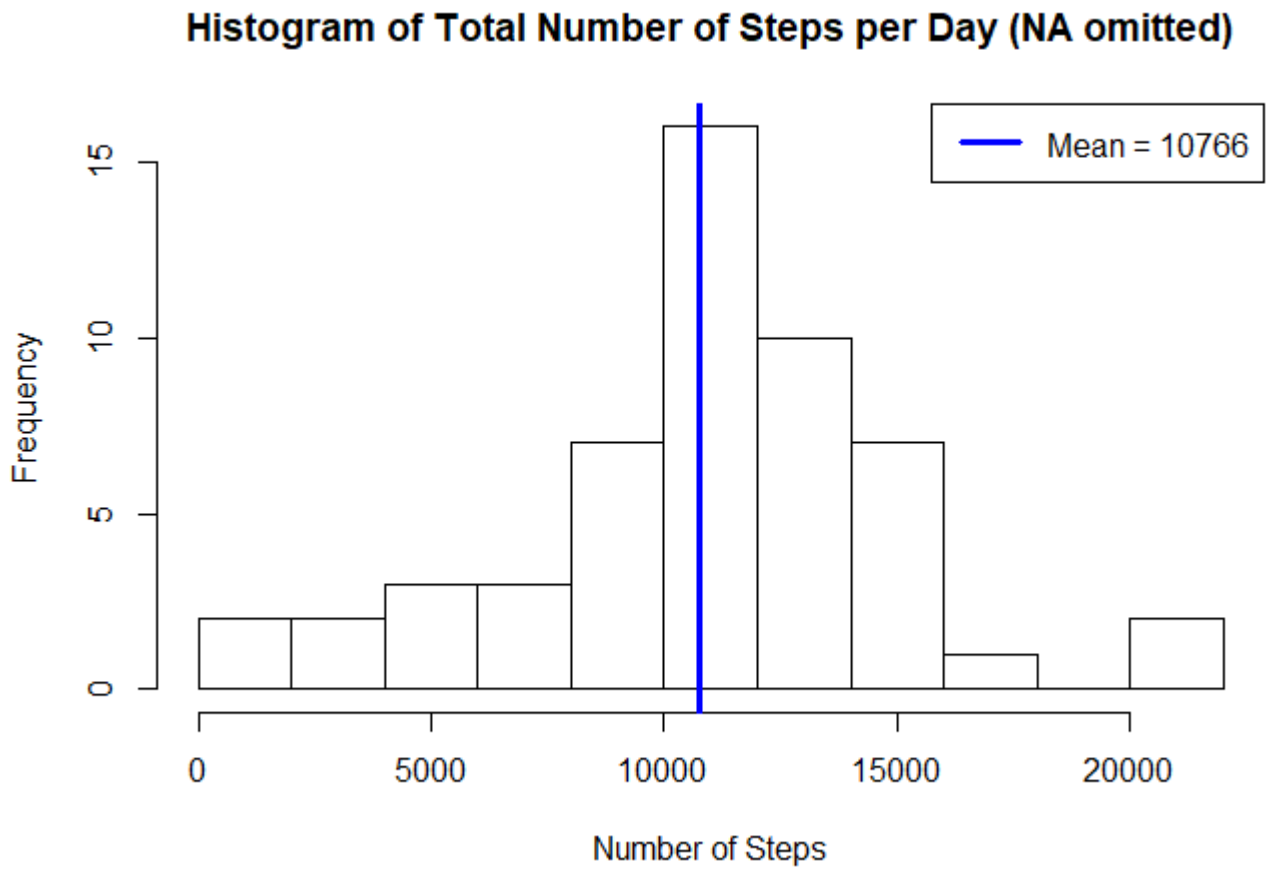
We first calculate the total number of steps per day.

```
dailyTotals <- aggregate(steps ~ date, data=df, FUN=sum, na.action = na.omit)
```

Using these calculations, we can see that the mean of the total number of steps per day is **10766** and the medium is **10765**.

We can also see this in the following histogram:

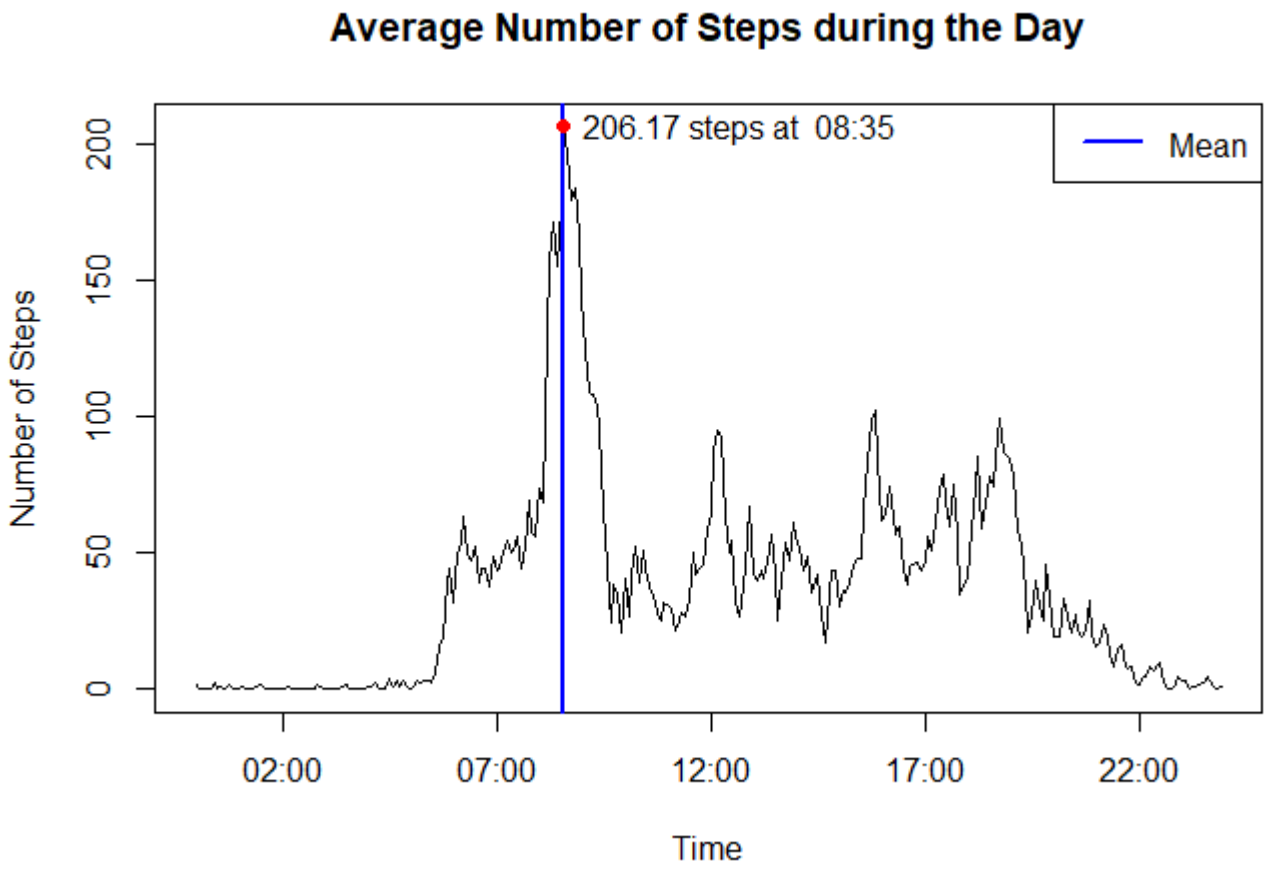
```
hist(dailyTotals$steps, breaks=10, main="Histogram of Total Number of Steps per Day (NA omitte
d)", xlab="Number of Steps")
abline(v=mean(dailyTotals$steps), col="blue", lwd=3)
legend(x = "topright", legend=paste("Mean =", as.integer(mean(dailyTotals$steps))), col="blue",
 lty=1, lwd=3)
```



What is the average daily activity pattern?

Now let's see how the activity looks throughout the day. First we calculate the average number of steps for each 5-minute interval.

```
intervalAverages <- aggregate(steps ~ interval, data = df[,c(1,3)], FUN=mean, na.action = na.omi
t)
#library(stringr)
#intervalAverages$interval <- str_pad(intervalAverages$interval, 4, pad="0")
intervalAverages$intervalFull <- strptime(intervalAverages$interval, "%H%M")
plot(x=intervalAverages$intervalFull, y=intervalAverages$steps, type="l", main="Average Number o
f Steps during the Day", xlab="Time", ylab="Number of Steps")
maxSteps <- max(intervalAverages$steps)
maxTime <- intervalAverages[which(grepl(max(intervalAverages$steps), intervalAverages$steps)),3]
abline(v=as.POSIXct(maxTime), col="blue", lwd=2)
points(x=maxTime, y=maxSteps, pch=19, col="red")
text(x=maxTime, y=maxSteps, paste(round(maxSteps, 2), "steps at ", format(maxTime, "%H:%M")), po
s = 4)
legend(x="topright", legend="Mean", col="blue", lty=1, lwd=2)
```



As we can see, the **08:35** time interval contains the most number of steps, on average, across all the days in the dataset.

Imputing missing values

There are **2304** observations in the dataset with missing values.

Let's replace those missing values with the average steps for their respective interval.

```
df2 <- df

for (i in 1:length(df2$interval)) {
  if (is.na(df2[i,1])) {
    df2[i,1] <- intervalAverages[which(grepl(df2[i,3], intervalAverages$interval)),2]
  }
}
head(df2)
```

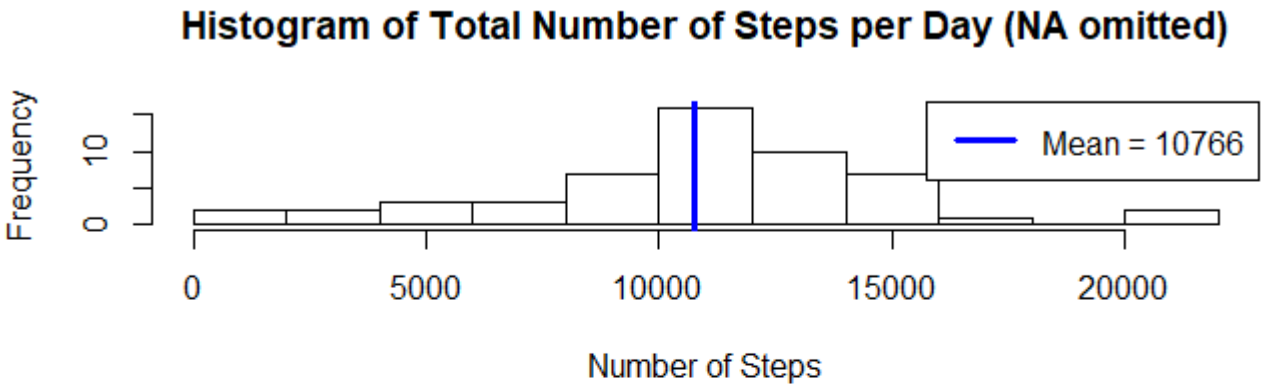
```
##       steps       date interval
## 1 1.7169811 2012-10-01     0000
## 2 0.3396226 2012-10-01     0005
## 3 0.1320755 2012-10-01     0010
## 4 0.1509434 2012-10-01     0015
## 5 0.0754717 2012-10-01     0020
## 6 2.0943396 2012-10-01     0025
```

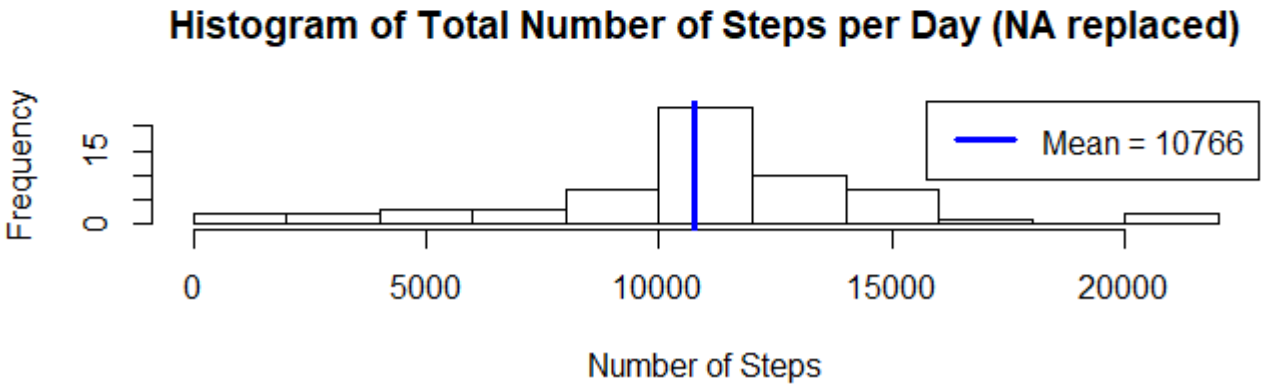
Now with the missing values filled with data, let's see how this new dataset compares to the original dataset.

```
dailyTotals2 <- aggregate(steps ~ date, data=df2, FUN=sum, na.action = na.omit)
par(mfcol=c(2,1))

hist(dailyTotals$steps, breaks=10, main="Histogram of Total Number of Steps per Day (NA omitte
d)", xlab="Number of Steps")
abline(v=mean(dailyTotals$steps), col="blue", lwd=3)
legend(x = "topright", legend=paste("Mean =", as.integer(mean(dailyTotals$steps))), col="blue",
 lty=1, lwd=3)

hist(dailyTotals2$steps, breaks=10, main="Histogram of Total Number of Steps per Day (NA replace
d)", xlab="Number of Steps")
abline(v=mean(dailyTotals2$steps), col="blue", lwd=3)
legend(x = "topright", legend=paste("Mean =", as.integer(mean(dailyTotals2$steps))), col="blue",
 lty=1, lwd=3)
```





We can see that the mean of the total number of steps per day is **10766** and the medium is **10766**. The calculations with replaced missing values are practically the same as the calculatations with missing values omitted.

Are there differences in activity patterns between weekdays and weekends?

As we see from the plots below, there are clear differences in the step activity for weekend vs weekday.

```
# Add new column with day of the week
df2$weekday <- weekdays(as.Date(df2$date))

# Convert day of the week to weekday or weekend
df2$weekday <- lapply(df2$weekday, function(x){
  switch(x,
    "Saturday" = x <- "weekend",
    "Sunday" = x <- "weekend",
    x <- "weekday")
  return(x)
})

# Convert to a factor variable
df2$weekday <- factor(df2$weekday, levels = c("weekday", "weekend"))

#Calculate averages
weekdayIntervalAverages <- aggregate(steps ~ interval + weekday, data = df2[,c(1,3,4)], FUN=mea
n)

#plot
library(lattice)
xyplot(steps~interval | weekday, data=weekdayIntervalAverages, xlim=c(0,2400), ylim=c(-20,250),
 type="l", layout=c(1,2), xlab="Interval", ylab="Number of Steps", main="Average Number of Step
s: Weekday vs Weekend")
```