Reproducible Research: Peer Assessment 1

BP

Wednesday, July 08, 2015

Pre-requisites:

1. Unzip file activity.zip
2. Assumed activity.csv and PA1_template.Rmd files are in the working directory
3. ggplot2 package is installed in R studio

Loading and processing the data

```
data <- read.csv("activity.csv", colClasses = c("integer", "Date", "factor"))
noNA <- na.omit(data)
rownames(noNA) <- 1:nrow(noNA)
head(noNA)
```

```
##   steps       date interval
## 1     0 2012-10-02        0
## 2     0 2012-10-02        5
## 3     0 2012-10-02       10
## 4     0 2012-10-02       15
## 5     0 2012-10-02       20
## 6     0 2012-10-02       25
```

```
dim(noNA)
```

```
## [1] 15264     3
```

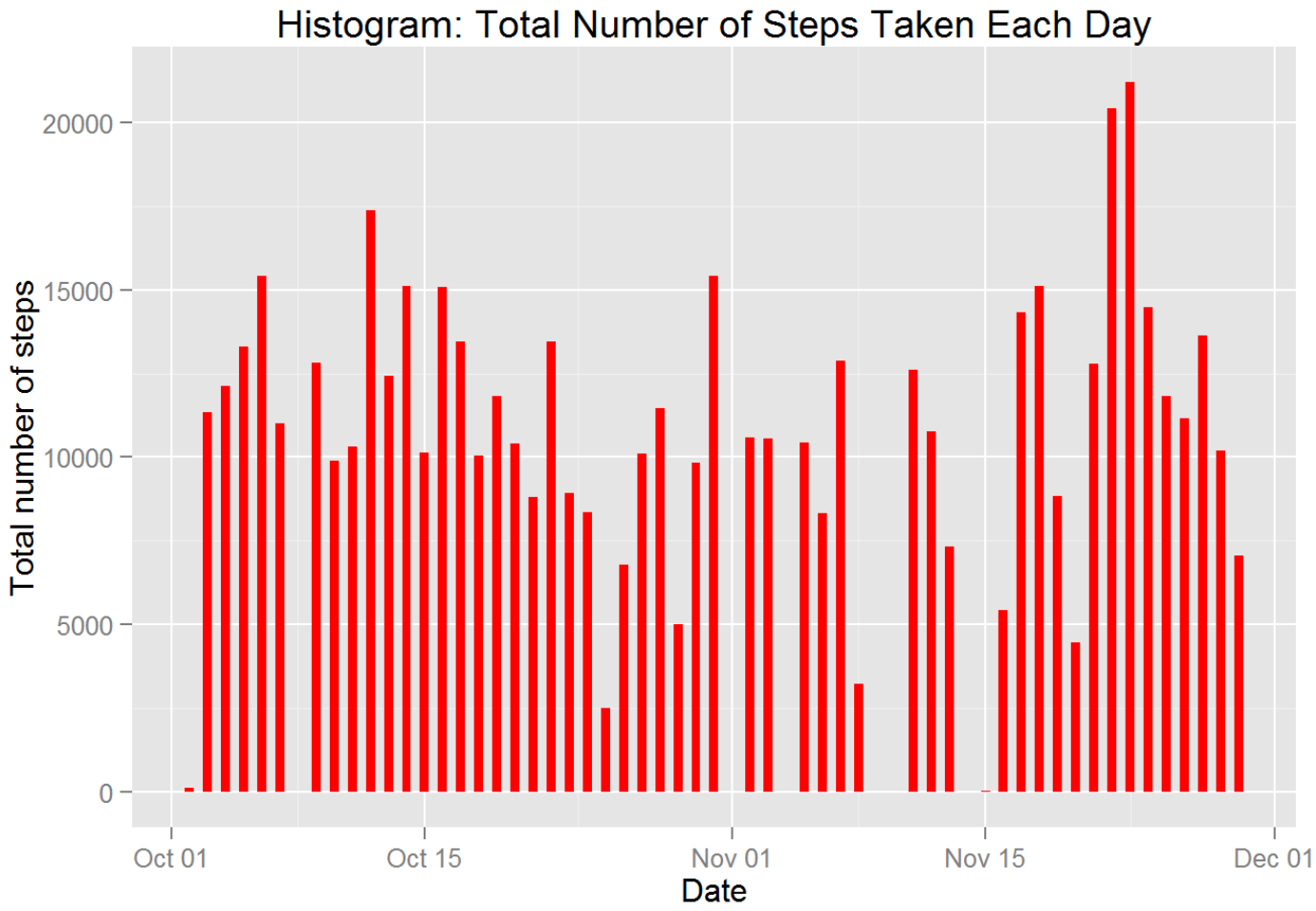
```
library(ggplot2)
```

What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset.

1. Calculate the total number of steps taken per day
2. Make a histogram of the total number of steps taken each day

```
ggplot(noNA, aes(date, steps)) + geom_bar(stat = "identity", fill = "red", width = 0.5) +
labs(title = "Histogram: Total Number of Steps Taken Each Day", x = "Date", y = "Total nu
mber of steps")
```



3. Calculate and report the mean and median total number of steps taken per day

Mean total number of steps taken per day:

```
totalSteps <- aggregate(noNA$steps, list(Date = noNA$date), FUN = "sum")$x
mean(totalSteps)
```

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

Median total number of steps taken per day:

```
median(totalSteps)
```

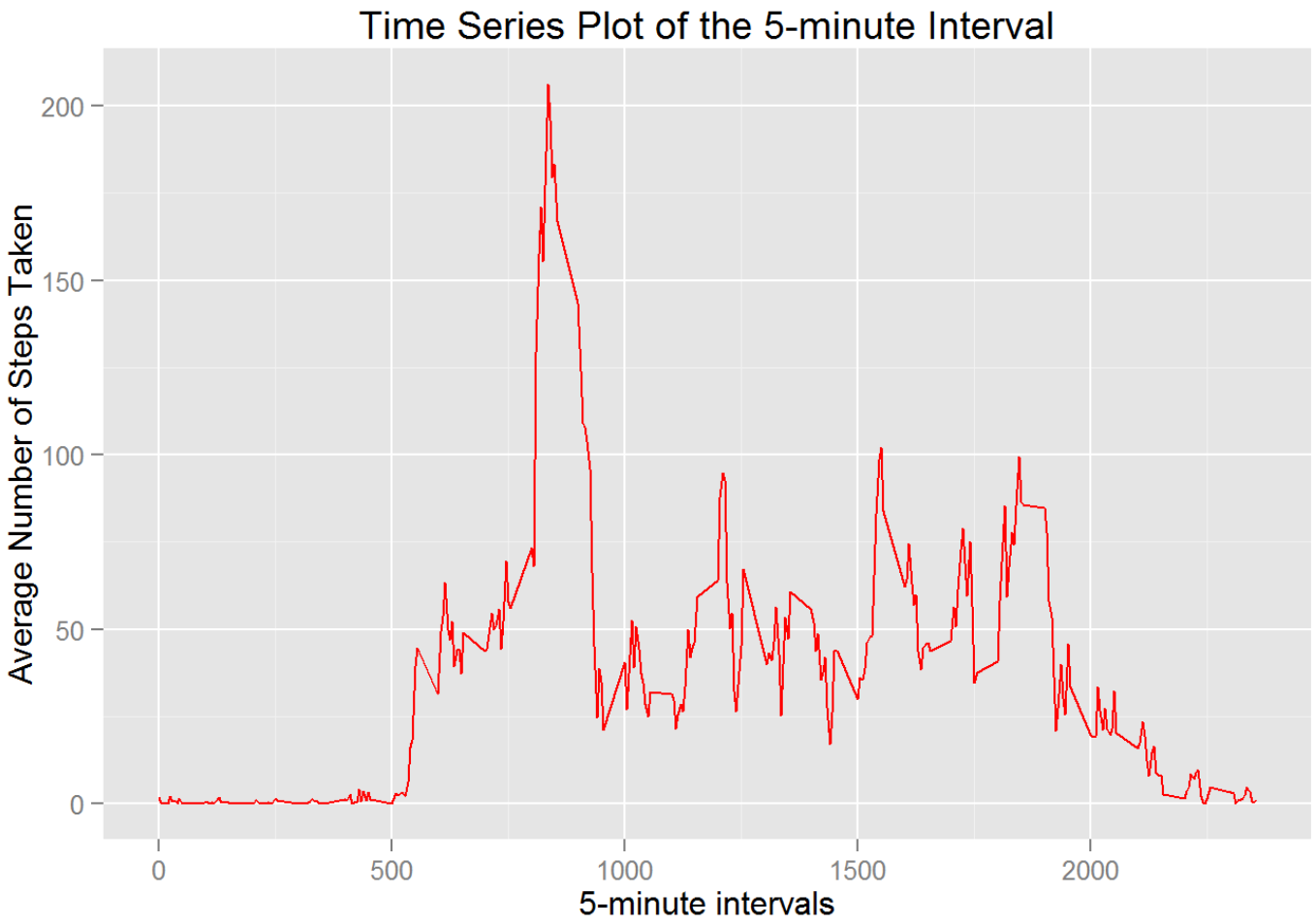
```
## [1] 10765
```

What is the average daily activity pattern?

1. Make a time series plot (i.e. type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
avgSteps <- aggregate(noNA$steps, list(interval = as.numeric(as.character(noNA$interva
l))), FUN = "mean")
names(avgSteps)[2] <- "meanOfSteps"

ggplot(avgSteps, aes(interval, meanOfSteps)) + geom_line(color = "red", size = 0.5) + lab
s(title = "Time Series Plot of the 5-minute Interval", x = "5-minute intervals", y = "Ave
rage Number of Steps Taken")
```



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

```
avgSteps[avgSteps$meanOfSteps == max(avgSteps$meanOfSteps), ]
```

```
##     interval meanOfSteps
## 104      835    206.1698
```

Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
sum(is.na(data))
```

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

2. Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.

Iam going to use the mean for that 5-minute interval to fill each NA value in the steps column.

3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
newData <- data
for (i in 1:nrow(newData)) {
    if (is.na(newData$steps[i])) {
        newData$steps[i] <- avgSteps[which(newData$interval[i] == avgSteps$interval), ]$m
eanOfSteps
    }
}

head(newData)
```

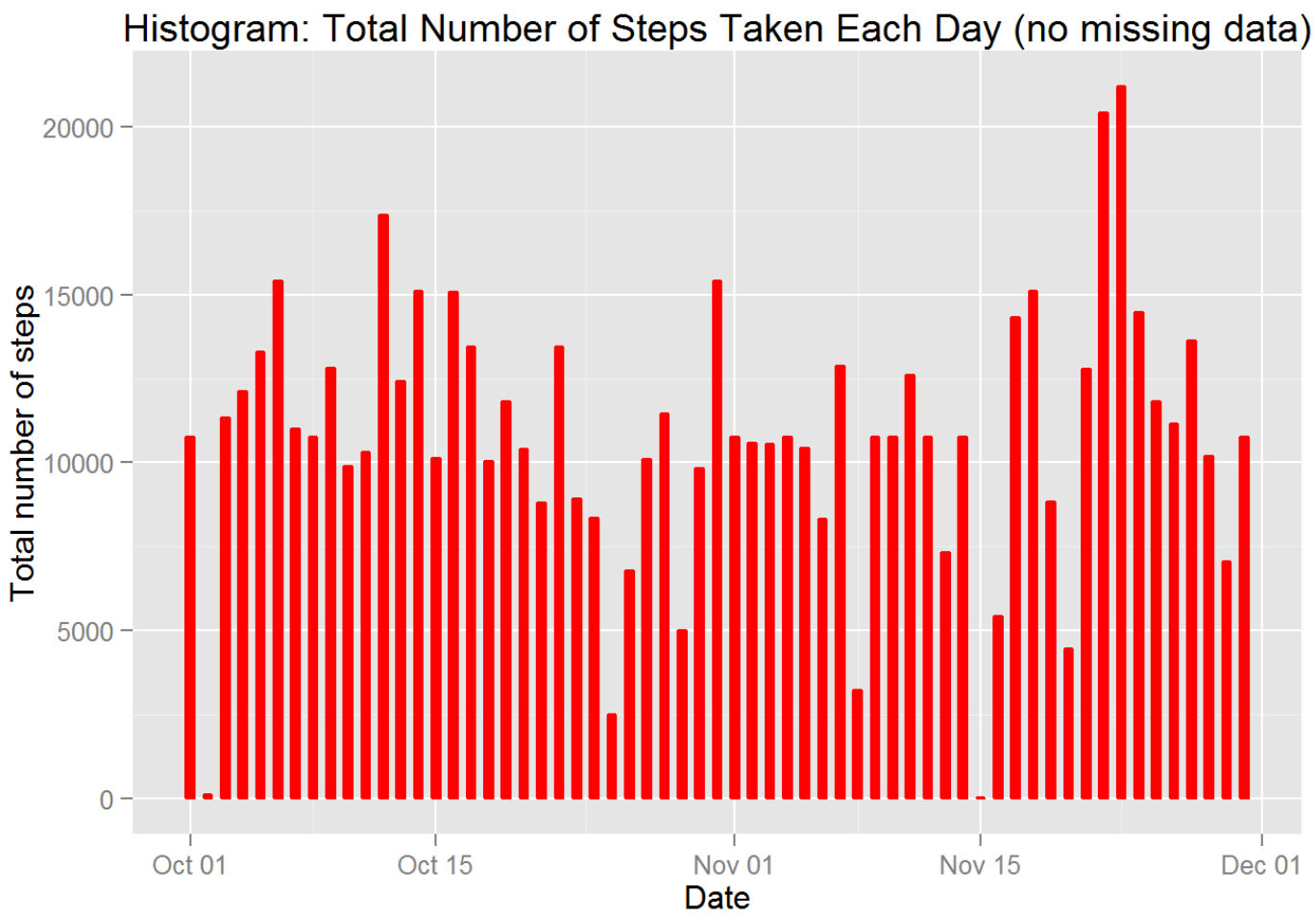
```
##       steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
```

```
sum(is.na(newData))
```

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

4. Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day.

```
ggplot(newData, aes(date, steps)) + geom_bar(stat = "identity",
                                             colour = "red",
                                             fill = "red",
                                             width = 0.5) +  labs(title = "Histogram: Tot
al Number of Steps Taken Each Day (no missing data)", x = "Date", y = "Total number of st
eps")
```



- Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

Mean total number of steps taken per day:

```
newTotalSteps <- aggregate(newData$steps,
                           list(Date = newData$date),
                           FUN = "sum")$x
newMean <- mean(newTotalSteps)
newMean
```

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

Median total number of steps taken per day:

```
newMedian <- median(newTotalSteps)
newMedian
```

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

Compare them with the two before imputing missing data:

```
oldMean <- mean(totalSteps)
oldMedian <- median(totalSteps)
newMean - oldMean
```

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

```
newMedian - oldMedian
```

```
## [1] 1.188679
```

So, after imputing the missing data, the new mean of total steps taken per day is the same as that of the old mean; the new median of total steps taken per day is greater than that of the old median.

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.

```
head(newData)
```

```
##       steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
```

```
newData$weekdays <- factor(format(newData$date, "%A"))
levels(newData$weekdays)
```

```
## [1] "Friday"    "Monday"    "Saturday"  "Sunday"    "Thursday"  "Tuesday"
## [7] "Wednesday"
```

```
levels(newData$weekdays) <- list(weekday = c("Monday", "Tuesday",
                                             "Wednesday",
                                             "Thursday", "Friday"),
                                 weekend = c("Saturday", "Sunday"))
levels(newData$weekdays)
```

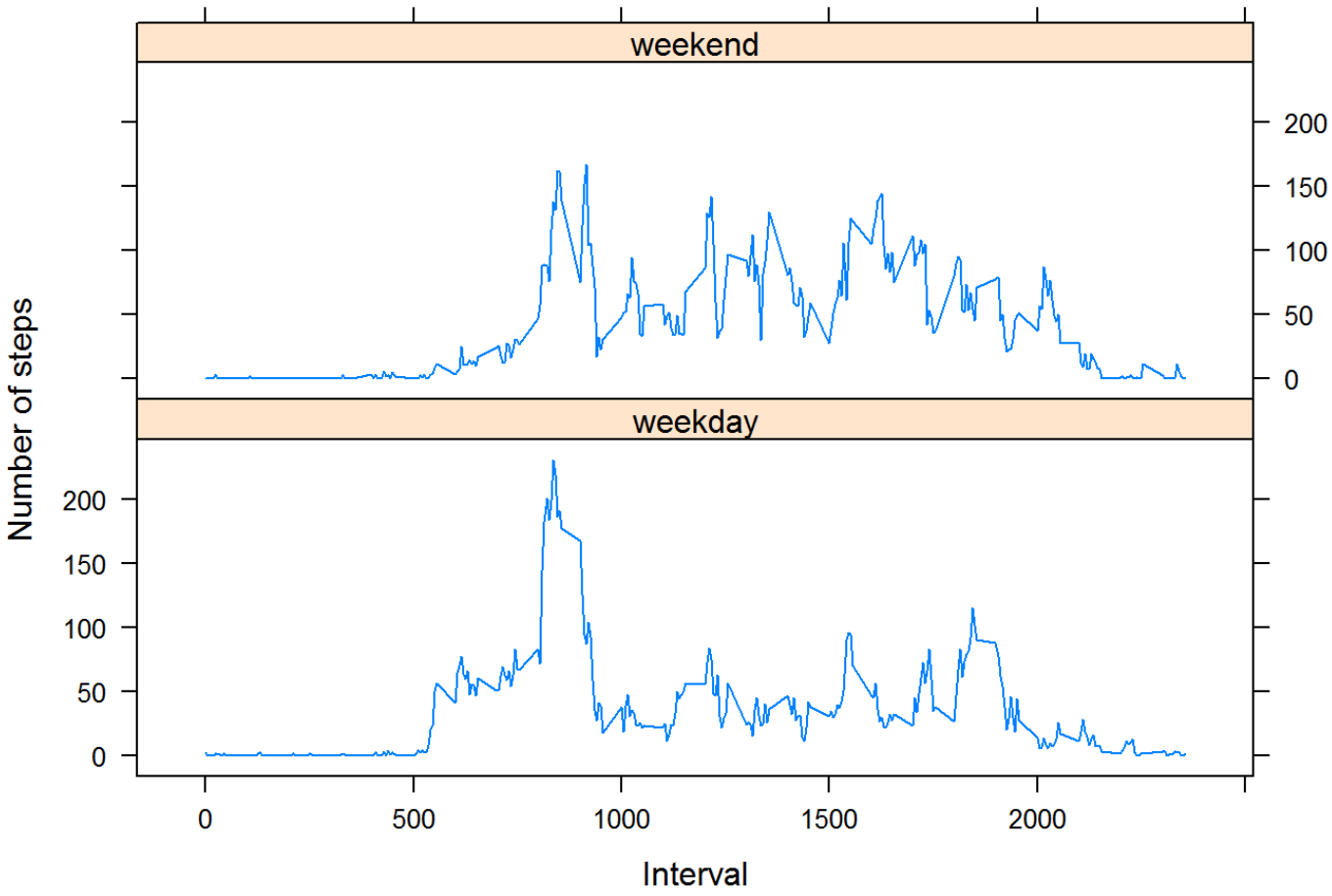
```
## [1] "weekday" "weekend"
```

```
table(newData$weekdays)
```

```
##
## weekday weekend
##   12960    4608
```

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

```
avgSteps <- aggregate(newData$steps,
                      list(interval = as.numeric(as.character(newData$interval)),
                           weekdays = newData$weekdays),
                      FUN = "mean")
names(avgSteps)[3] <- "meanOfSteps"
library(lattice)
xyplot(avgSteps$meanOfSteps ~ avgSteps$interval | avgSteps$weekdays,
       layout = c(1, 2), type = "l",
       xlab = "Interval", ylab = "Number of steps")
```


weekend
weekday
Number of steps
Interval
0
50
100
150
200
0
500
1000
1500
2000