Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

The dataset is taken from the coursera course web site:

- Dataset: Activity monitoring data [52K]

The variables included in this dataset are:

- **steps**: Number of steps taking in a 5-minute interval (missing values are coded as NA)
- **date**: The date on which the measurement was taken in YYYY-MM-DD format
- **interval**: Identifier for the 5-minute interval in which measurement was taken

1. Load the data (i.e. read.csv())

- It is loaded with `read.csv()` function. The separator is the default ‘,’ and the first row contains the header:

```
df <- read.csv("activity.csv")
dim(df)
```

```
## [1] 17568     3
```

```
names(df)
```

```
## [1] "steps"    "date"     "interval"
```

```
str(df)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

As we can see, there are 17568 rows and 3 columns in the dataset.

2. Process/transform the data (if necessary) into a format suitable for your analysis

- sort them by date & interval

```
df <- df[with(df, order(date, interval)), ]
head(df)
```

```
##   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 mean total number of steps taken per day?

- aggregate the data.frame into two columns: date, steps(sum)
- calculate the mean total number of steps per day.

```
total.steps <- aggregate(steps ~ date, df, sum)
head(total.steps)
```

```
##         date steps
## 1 2012-10-02   126
## 2 2012-10-03 11352
## 3 2012-10-04 12116
## 4 2012-10-05 13294
## 5 2012-10-06 15420
## 6 2012-10-07 11015
```

```
mu <- mean(total.steps$steps)
print(mu)
```

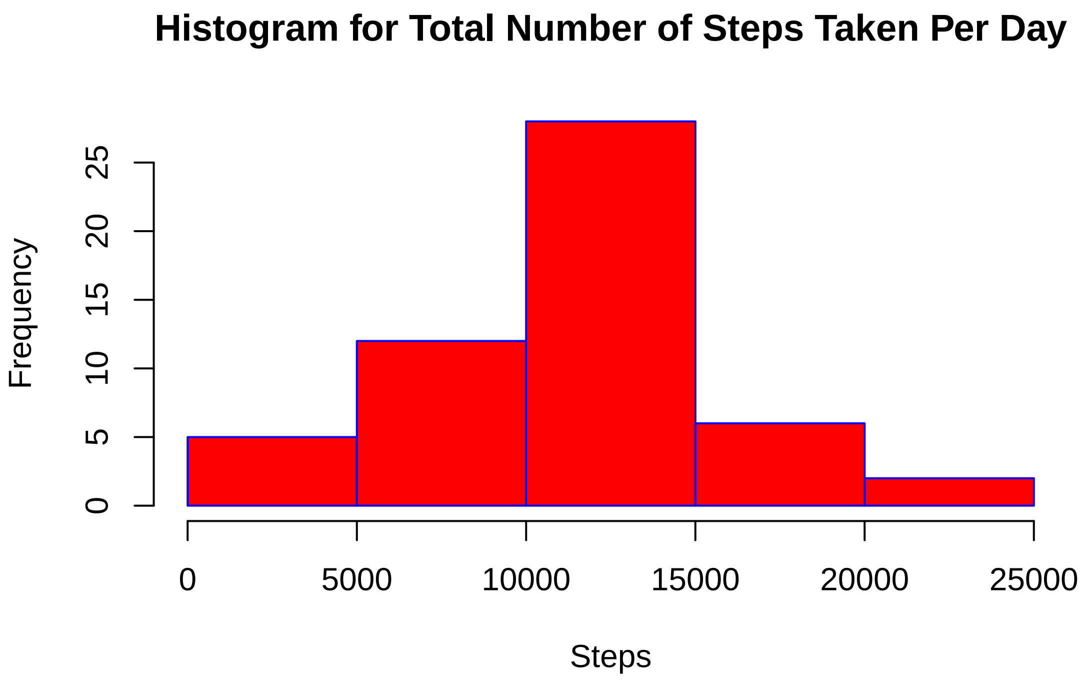
```
## [1] 10766.19
```

> The mean total number of steps taken per day is 10,766 steps.

Make a histogram of the total number of steps taken each day

The following is the code and the histogram:

```
steps <-total.steps$steps
hist(steps,
     main="Histogram for Total Number of Steps Taken Per Day",
     xlab="Steps",
     border="blue",
     col="red",
     breaks=5
)
```



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

```
mean.step.per.day <- mean(total.steps$steps, na.rm = TRUE)
mean.step.per.day
```

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

```
median.step.per.day <- median(total.steps$steps, na.rm = TRUE)
median.step.per.day
```

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

- The mean total number of steps taken per day is 10,766 steps.
- The median total number of steps taken per day is 10,765 steps.

What is the average daily activity pattern?

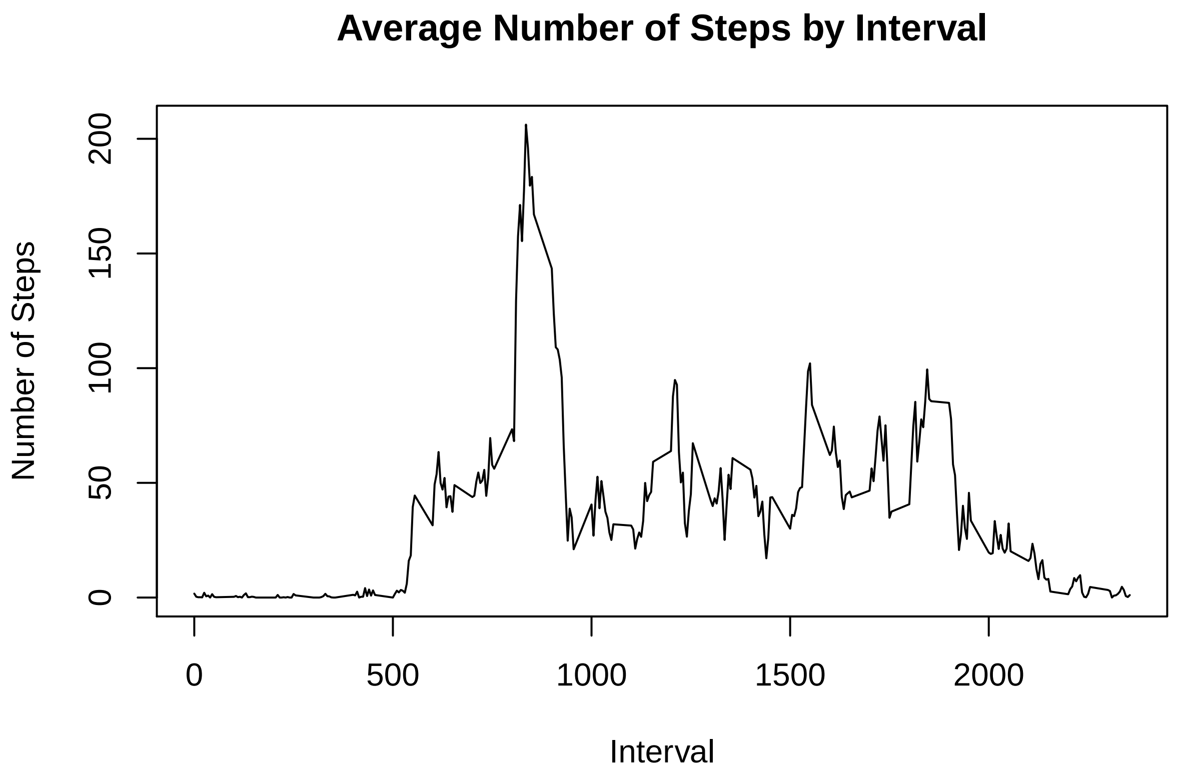
1. Make a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

- let's look at the "total number of steps taken" at each interval across all days

```
steps.by.interval <- aggregate(steps ~ interval, df, mean)
head(steps.by.interval)
```

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

```
plot(
    steps.by.interval$interval
    ,steps.by.interval$steps
    ,type="l"
    ,xlab="Interval"
    ,ylab="Number of Steps"
    ,main="Average Number of Steps by Interval")
```



2. Which interval contains the maximum number of steps?

```
max.interval <- steps.by.interval[which.max(steps.by.interval$steps), "interval"]
```

- The interval that contains the maximum number of steps is 835 (with a total of 206 steps taken.)

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)

```
n <- sum(!complete.cases(df))
```

- There are 2304 rows that have missing values.

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.

- We will replace missing values `NA` with the mean steps for that interval (across all days).

```
df.merged <- merge(x=df, y=steps.by.interval[ , c("interval", "steps")], by = "interval")
df.merged <- df.merged[with(df.merged, order(date, interval)), ]
head(df.merged)
```

```
##     interval steps.x       date   steps.y
## 1          0      NA 2012-10-01 1.7169811
## 63         5      NA 2012-10-01 0.3396226
## 128       10      NA 2012-10-01 0.1320755
## 205       15      NA 2012-10-01 0.1509434
## 264       20      NA 2012-10-01 0.0754717
## 327       25      NA 2012-10-01 2.0943396
```

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

```
df.imputed <-
    transform(
        df,
        steps =
            ifelse(
                is.na(df$steps),
                steps.by.interval$steps[
                    match(df$interval, steps.by.interval$interval)
                ],
                df$steps
            )
    )

df.imputed <- df.imputed[with(df.imputed, order(date, interval)), ]
head(df.imputed)
```

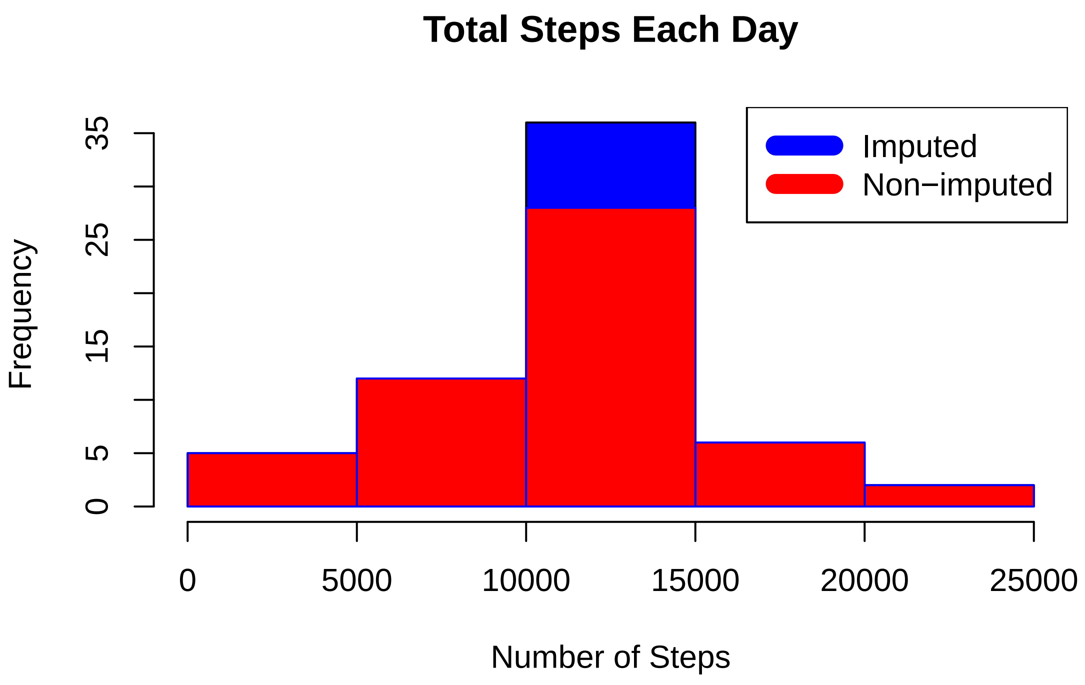
```
##       steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
```

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. 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?

```
total.steps.imputed <- aggregate(steps ~ date, df.imputed, sum)
hist(
    total.steps.imputed$steps,
    main = "Total Steps Each Day",
    col="blue",
    xlab="Number of Steps",
    breaks=5
)

hist(
    total.steps$steps,
    col="red",
    border="blue",
    xlab="Number of Steps",
    breaks=5,
    add=TRUE
)

legend("topright", c("Imputed", "Non-imputed"), col=c("blue", "red"), lwd=10)
```



```
mean.step.per.day.imputed <- mean(total.steps.imputed$steps)
mean.step.per.day
```

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

```r
median.step.per.day.imputed <- median(total.steps.imputed$steps)
median.step.per.day
```

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

- There are not much differences in the mean and median comparing with pre-imputed estimates:
  - before impute, the mean was 10,766.19, after imputed, the mean is 10,766.19.
  - before impute, the median was 10,765, after imputed, the median is 10,766.19.
  - Since we imputed the missing values with the average steps of the interval. The results did not make any impact to the original estimate which was simply to remove the missing vallues.

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.

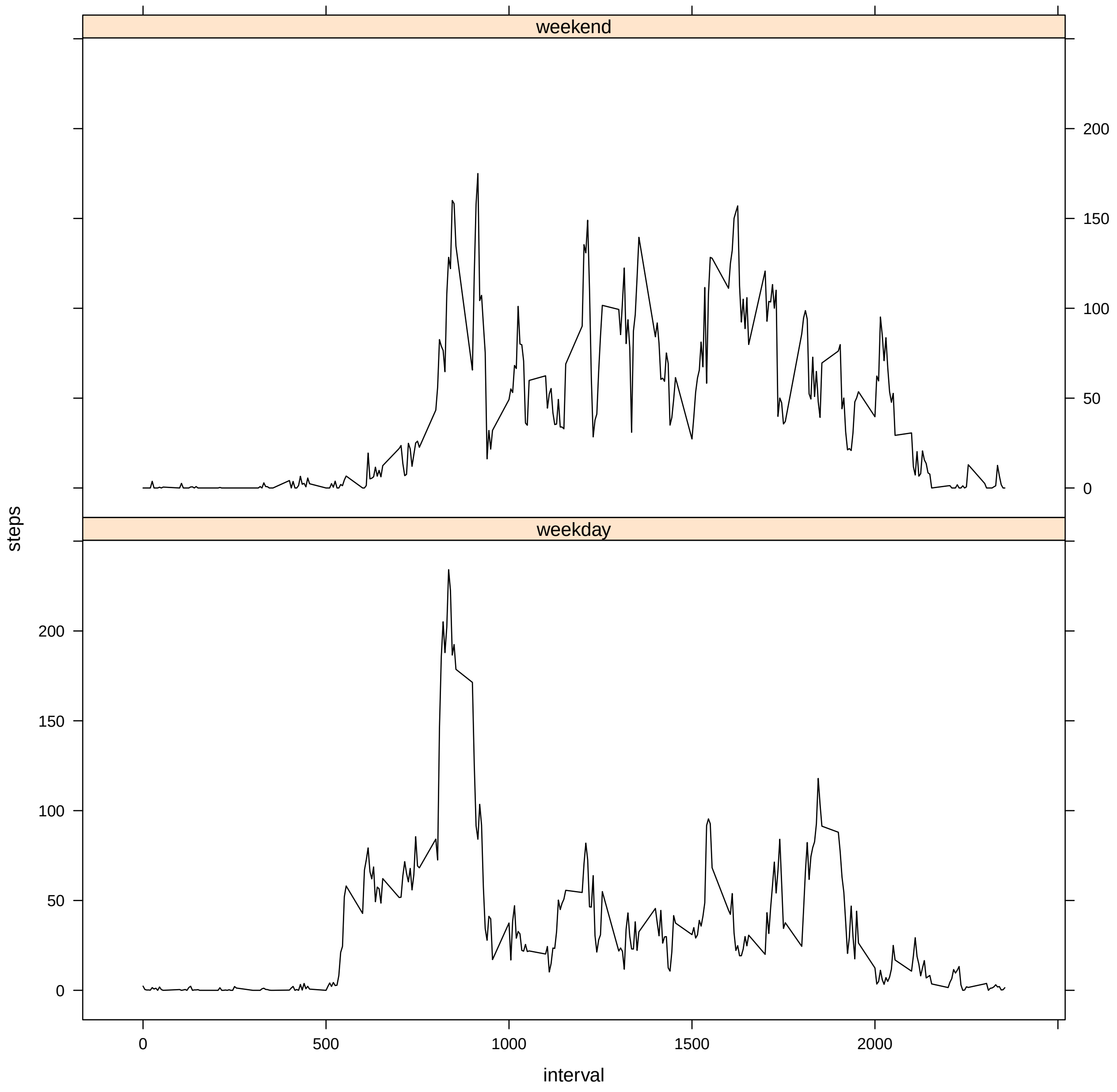
```r
df$date <- as.Date(as.character(df$date), "%Y-%m-%d")
df$weekend <- factor(
                (weekdays(df$date) %in% c('Saturday', 'Sunday')),
                labels=c('weekday', 'weekend')
              )
head(df)
```

```
##   steps       date interval weekend
## 1    NA 2012-10-01        0 weekday
## 2    NA 2012-10-01        5 weekday
## 3    NA 2012-10-01       10 weekday
## 4    NA 2012-10-01       15 weekday
## 5    NA 2012-10-01       20 weekday
## 6    NA 2012-10-01       25 weekday
```

```r
steps.by.interval.week <- aggregate(steps ~ interval + weekend, df, mean)
```

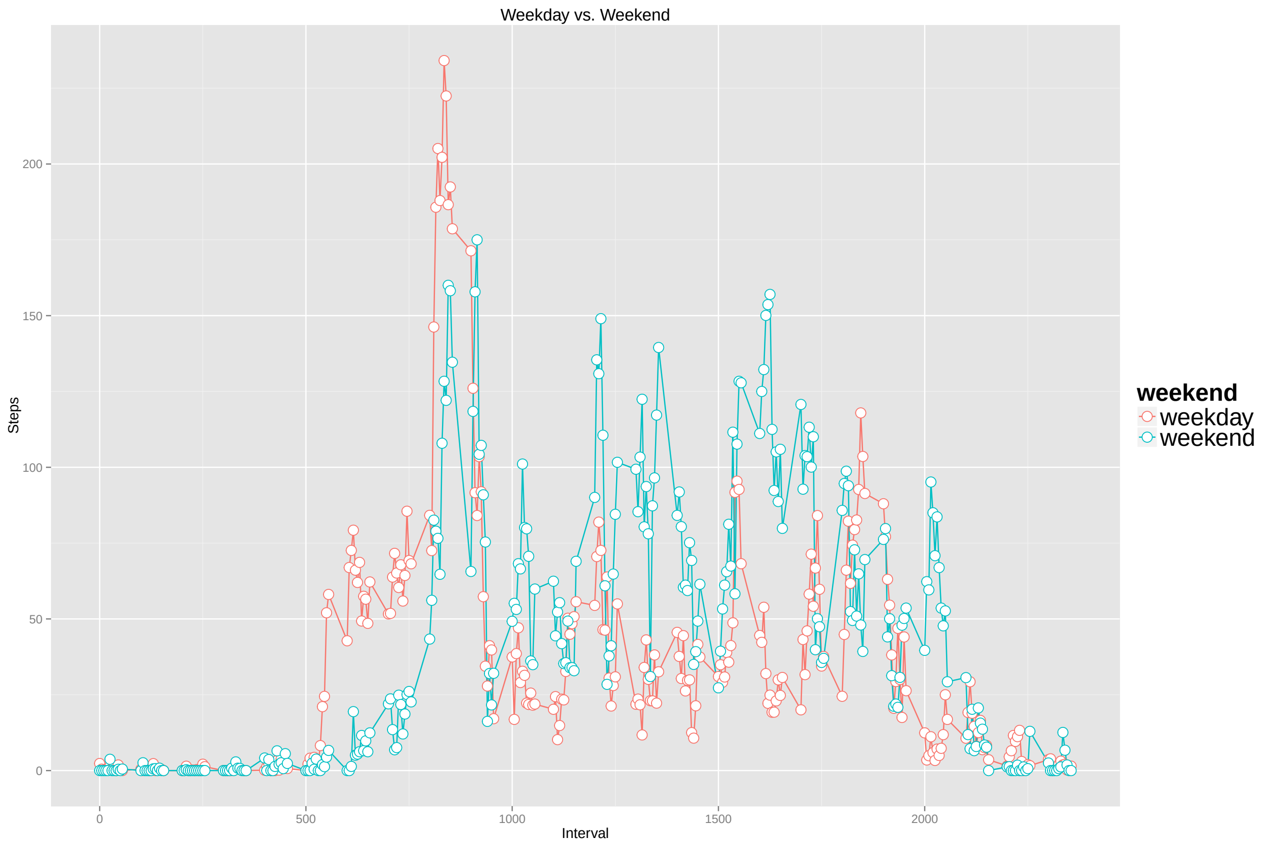
2. Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```r
library("lattice")
xyplot(steps ~ interval | weekend,
    data = steps.by.interval.week,
    type = "l",
    layout=c(1,2),
    col.line = c(rep("black",3), "red"))
```



- There are definitely different activity patterns between weekdays and weekends.

Here is another way to see the patterns:


Weekday vs. Weekend
Steps
200
150
100
50
0
0
500
1000
1500
2000
Interval
weekend
weekday
weekend