Reproducible Research: Peer Assessment 1

Zakia Sultana

March 3, 2016

Loading necessary libraries

```
library(ggplot2)
library(lubridate)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Loading and preprocessing the data

```
activity <- read.csv(file= "activity.csv", head=TRUE, sep=",")
str(activity) # View data structure
```

```
## '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 ...
```

```
activity$date <- ymd(activity$date)

summary(activity) # View data Summary
```

```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8
##  Median :  0.00   Median :2012-10-31   Median :1177.5
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0
##  NA's   :2304
```

Question one : what is mean total number of steps taken per day?

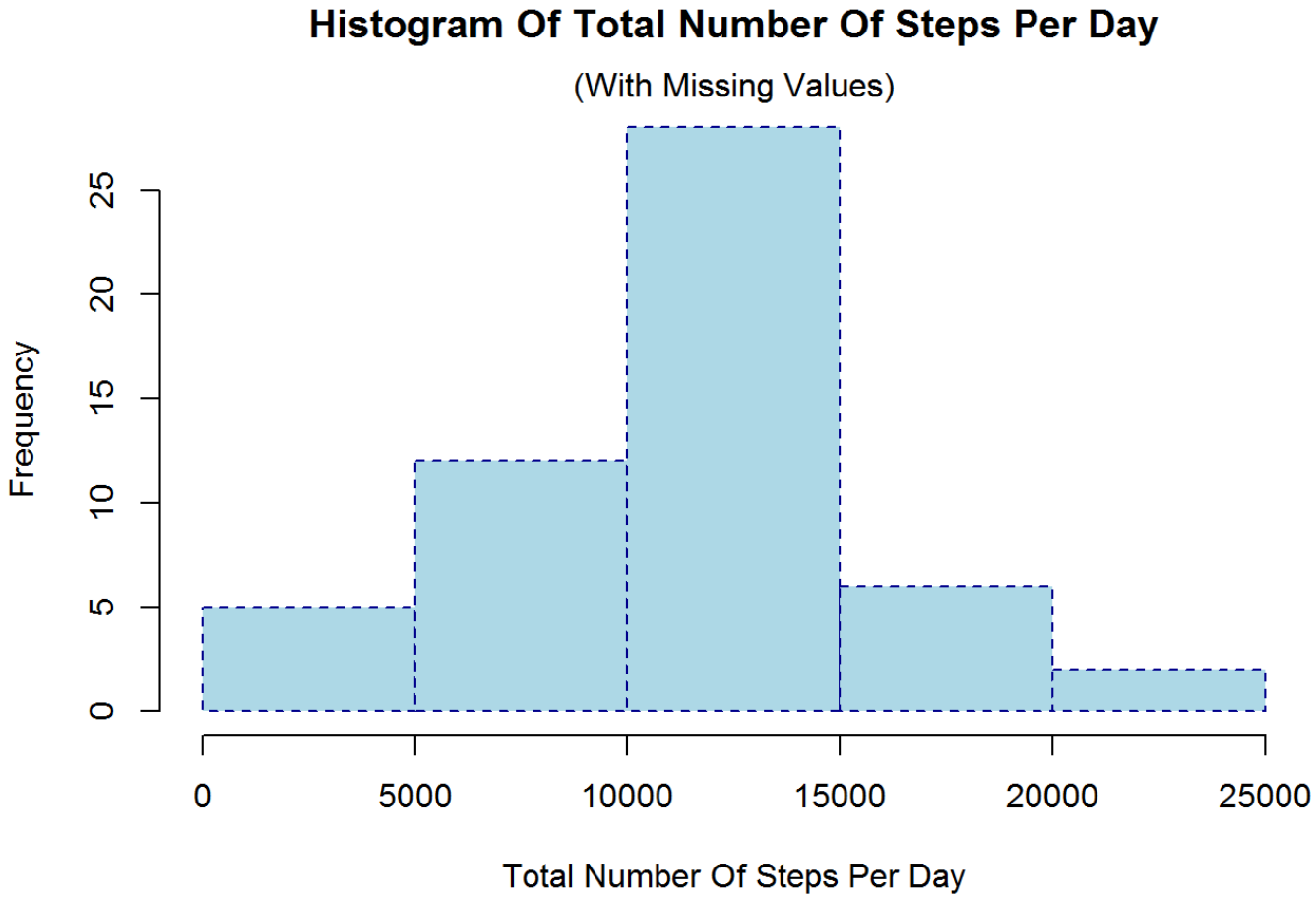
For this part of the work, we ignore the missing values in the dataset.

Step 1) Calculating the total number of steps taken per day:

```
totalstepsperday=aggregate(steps~date,activity,sum) # sum total steps over each day
head(totalstepsperday)
```

```
##         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
```

```
hist(totalstepsperday$steps,  # Frequency(histogram) of total steps per day
     col="lightblue",
     border="blue4",
     lty=2,
     main="Histogram Of Total Number Of Steps Per Day",
     xlab="Total Number Of Steps Per Day")
mtext("(With Missing Values)")
```




```
meantotsteps=mean(totalstepsperday$steps) # Average total steps per day
mediantotsteps=median(totalstepsperday$steps)# Median total steps per day
```

Mean Total Number Of Steps Per Day (with missing values) : 10766.19

Median Total Number Of Steps Per Day (with missing values) : 10765

Question two: what is the average daily activity pattern?

Generate New Dataframe to track average steps per interval :

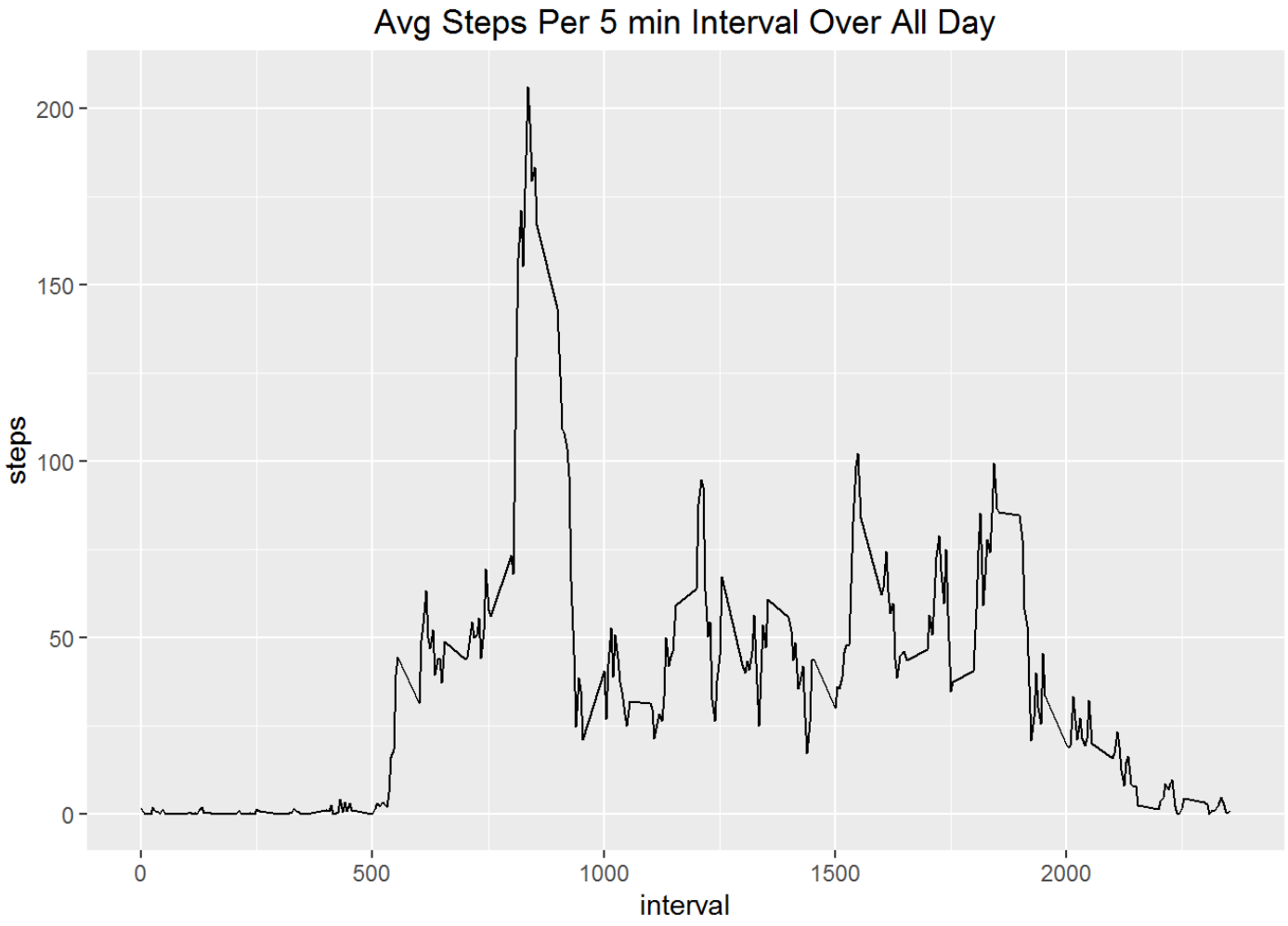
```
avgstepsperinterval=aggregate(steps~interval,activity,mean) # avg step per interval
str(avgstepsperinterval);summary(avgstepsperinterval)
```

```
## 'data.frame':    288 obs. of  2 variables:
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
##  $ steps   : num  1.717 0.3396 0.1321 0.1509 0.0755 ...
```

```
##     interval          steps
##  Min.   :   0.0   Min.   :  0.000
##  1st Qu.: 588.8   1st Qu.:  2.486
##  Median :1177.5   Median : 34.113
##  Mean   :1177.5   Mean   : 37.383
##  3rd Qu.:1766.2   3rd Qu.: 52.835
##  Max.   :2355.0   Max.   :206.170
```

Time Series Plot - Average Steps per 5 min interval Over all Days

```
par(mar=c(5,6,4,2))
ggplot(avgstepsperinterval, aes(interval, steps)) + geom_line() + ggtitle("Avg Steps Per 5 min Interval Over All Day")
```



Interval with Maximum Average Steps :

```
intmaxsteps=avgstepsperinterval$interval[avgstepsperinterval$steps==max(avgstepsperinterval$steps)]
```

5 min Interval with Maximum Average Steps : 835

Question three: imputing missing values

Number of Missing Values :

```
numissingvalues=nrow(activity[is.na(activity$steps),])
numissingvalues
```

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

Total Number of Missing Values : 2304

Filling Missing Values (use mean of time interval) :

```
activity1=merge(activity,avgstepsperinterval,by="interval") # merge
summary(activity1) # Review data frame & NA totals
```

```
##     interval         steps.x           date               steps.y
##  Min.   :   0.0   Min.   :  0.00   Min.   :2012-10-01   Min.   :  0.000
##  1st Qu.: 588.8   1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.:  2.486
##  Median :1177.5   Median :  0.00   Median :2012-10-31   Median : 34.113
##  Mean   :1177.5   Mean   : 37.38   Mean   :2012-10-31   Mean   : 37.383
##  3rd Qu.:1766.2   3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.: 52.835
##  Max.   :2355.0   Max.   :806.00   Max.   :2012-11-30   Max.   :206.170
##                   NA's   :2304
```

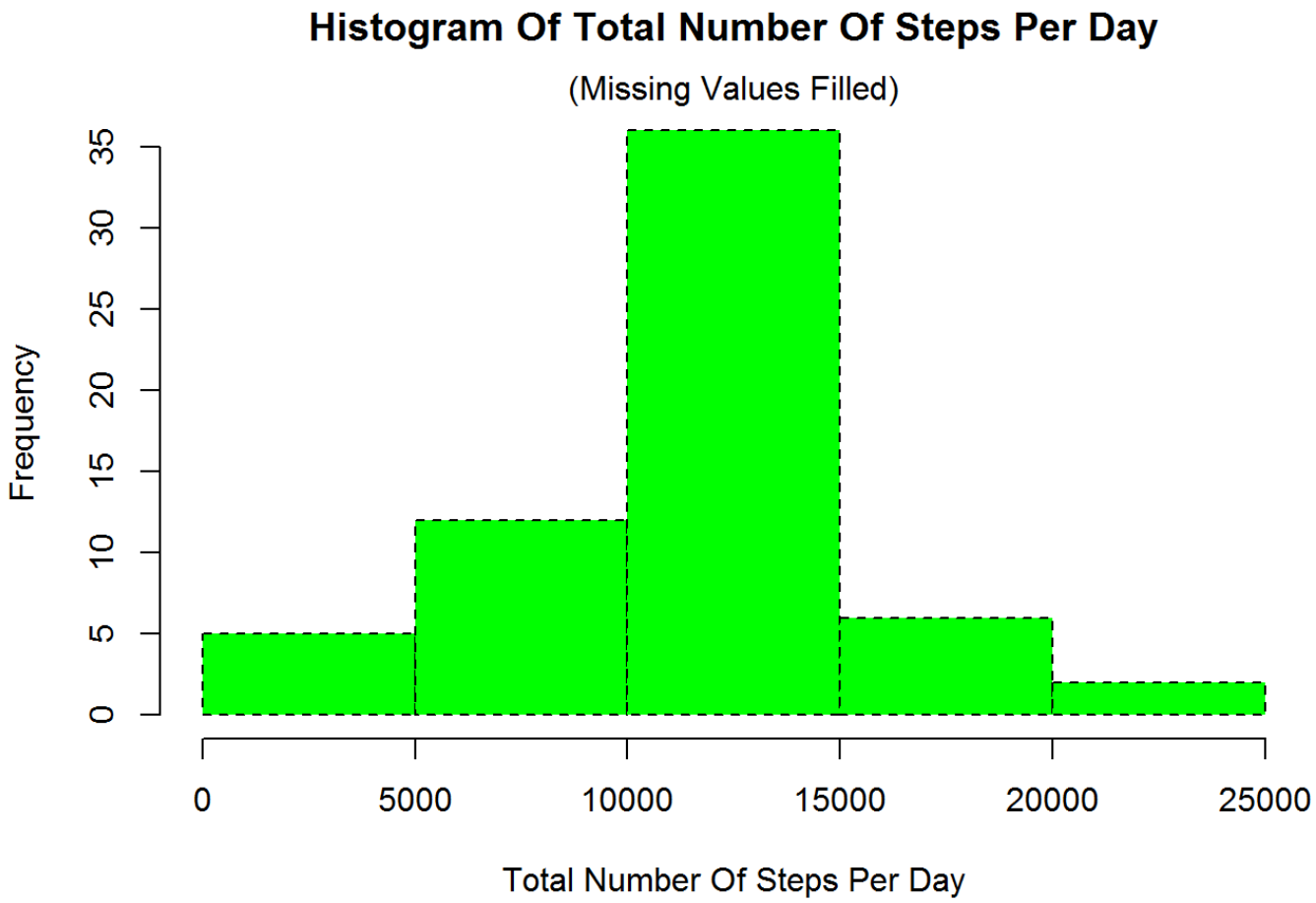
```
# replace NA's with mean of intervals
activity1$steps.x[is.na(activity1$steps.x)]=round(activity1$steps.y[is.na(activity1$step
s.x)],0)
activity1=activity1[,c('interval','steps.x','date')] # Drop the merged column
names(activity1)[2]='steps' # rename steps.x to steps
summary(activity1) # re-check values for NA's
```

```
##     interval          steps            date
##  Min.   :   0.0   Min.   :  0.00   Min.   :2012-10-01
##  1st Qu.: 588.8   1st Qu.:  0.00   1st Qu.:2012-10-16
##  Median :1177.5   Median :  0.00   Median :2012-10-31
##  Mean   :1177.5   Mean   : 37.38   Mean   :2012-10-31
##  3rd Qu.:1766.2   3rd Qu.: 27.00   3rd Qu.:2012-11-15
##  Max.   :2355.0   Max.   :806.00   Max.   :2012-11-30
```

** Note absence of NA's in the second summary above

Histogram, mean and median of total steps taken per day for New Dataframe equal to activity, but with missing values filled :

```
newtotalstepsperday=aggregate(steps~date,activity1,sum) # sum total steps over each day
hist(newtotalstepsperday$steps,col="green",lty=2,main="Histogram Of Total Number Of Steps
Per Day",
     xlab="Total Number Of Steps Per Day")
mtext("(Missing Values Filled)")
```



Average and Median total steps per day

```
newmeantotsteps=mean(newtotalstepsperday$steps)
newmediantotsteps=median(newtotalstepsperday$steps)
```

New Mean Total Number Of Steps Per Day (Missing values filled) : 10765.64

New Median Total Number Of Steps Per Day (Missing values filled) : 10762

Impact Of Adding Missing Values :

```
old=c("mean"=meantotsteps,"median"=mediantotsteps)
new=c("mean"=newmeantotsteps,"median"=newmediantotsteps)
oldnew=data.frame(old,new)
oldnew$diff=(new-old)/old*100
oldnew
```

```
##             old      new         diff
## mean   10766.19 10765.64 -0.005102409
## median 10765.00 10762.00 -0.027868091
```

“diff” column in the above dataframe indicates the % difference in the mean and median values from the earlier estimates with missing values and current estimates with missing values filled in. There is a very marginal, practically negligible difference between the earlier and current estimates. Question four: are there differences in activity patterns between weekdays and weekends? — add a day column for weekdays/weekends and weekday function to identify weekends

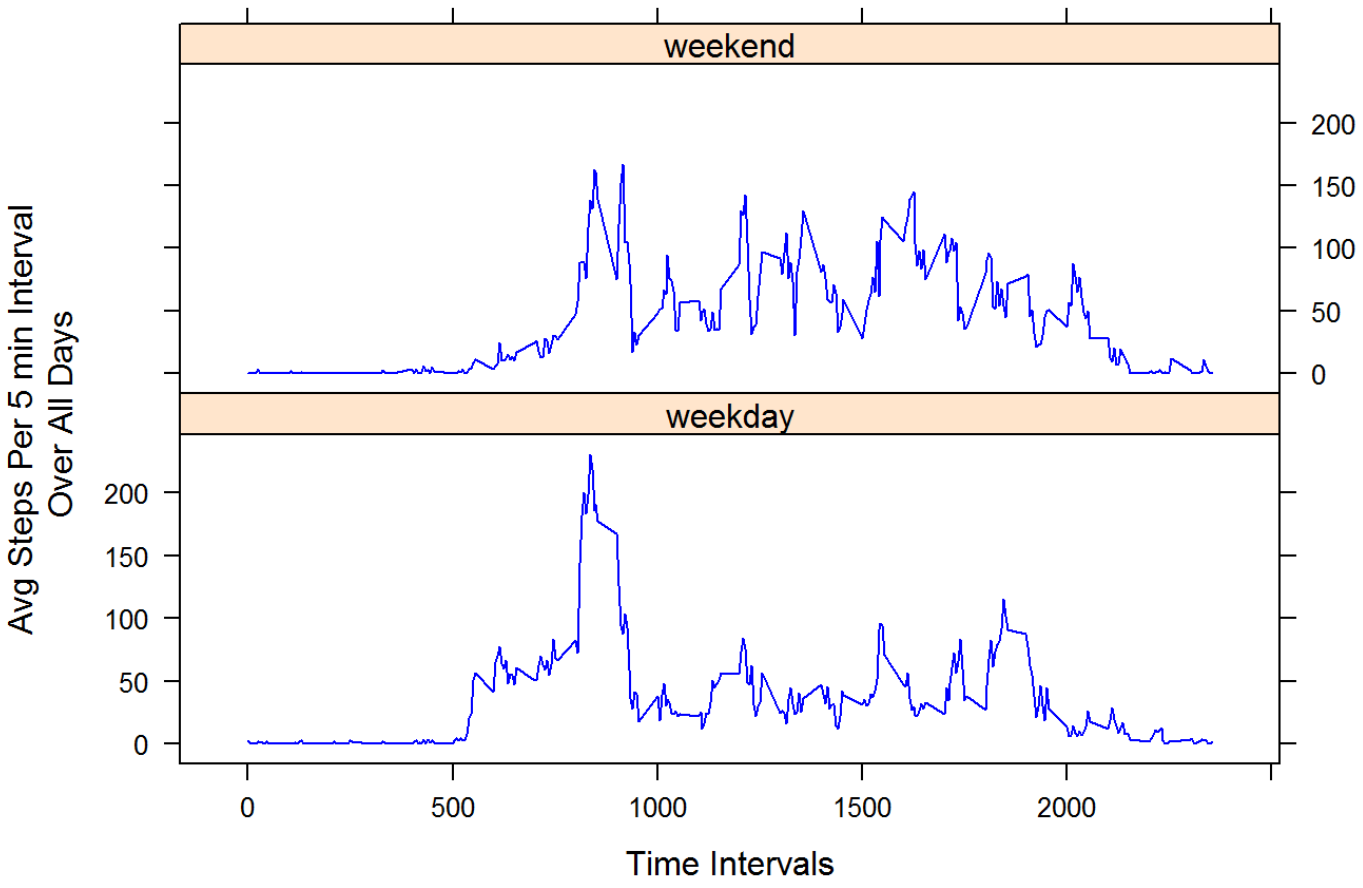
```
activity1$day[weekdays(as.Date(activity1$date))%in%c("Sunday","Saturday")]="weekend"
activity1$day[is.na(activity1$day)]="weekday" # All other days are weekdays
activity1$day=as.factor(activity1$day) # convert this to factor variable
str(activity1)
```

```
## 'data.frame':    17568 obs. of  4 variables:
##  $ interval: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ steps   : num  2 0 0 0 0 0 0 0 0 0 ...
##  $ date    : POSIXct, format: "2012-10-01" "2012-11-23" ...
##  $ day     : Factor w/ 2 levels "weekday","weekend": 1 1 2 1 2 1 2 1 1 2 ...
```

Find average steps per interval by type of day

```
average=aggregate(steps~interval+day,activity1,mean)
```

```
library(lattice)
xyplot(steps~interval|day,
       average,
       type="l",
       main="Avg Steps Per 5 min Interval Over All Days",
       xlab="Time Intervals",
       ylab="Avg Steps Per 5 min Interval \nOver All Days",
       col="blue",
       layout=c(1,2))
```


Avg Steps Per 5 min Interval Over All Days
weekend
weekday
Avg Steps Per 5 min Interval Over All Days
0
50
100
150
200
0
500
1000
1500
2000
Time Intervals