Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

We first load the data (assumed to be in a folder named data) and introduce 2 vectors steps.day which represents the toyal number of steps each day and steps.mean.5 which represents the mean over the 61 days period of the steps taken every 5 minutes. We also notice that the third columns express time of the day in the format "%H%M".

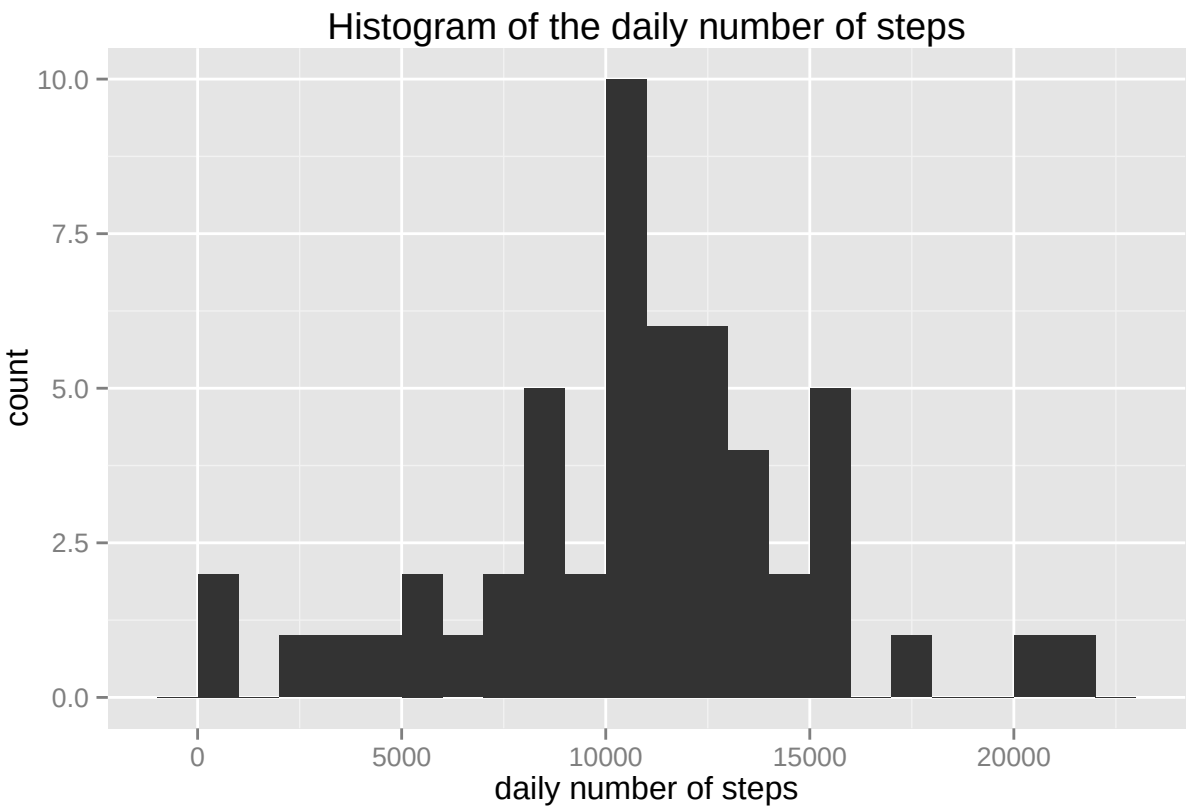
```
library(lattice)
library(ggplot2)
activity=read.csv("./data/activity.csv")
days=levels(activity[,2])
steps.day=tapply(activity[,1],activity[,2],sum)
steps.mean.5=tapply(activity[,1],activity[,3],mean,na.rm=TRUE)
l=length(which(is.na(steps.day)))#we miss the data for 8 days
```

We also notice that all the data of 8 days are completely missing.

What is mean total number of steps taken per day?

We first draw an histogram of the number of steps taken each day

```
qplot(steps.day, geom="histogram",xlab="daily number of steps" ,main="Histogram of the daily number of s
```



We compute the mean by removing missing data.

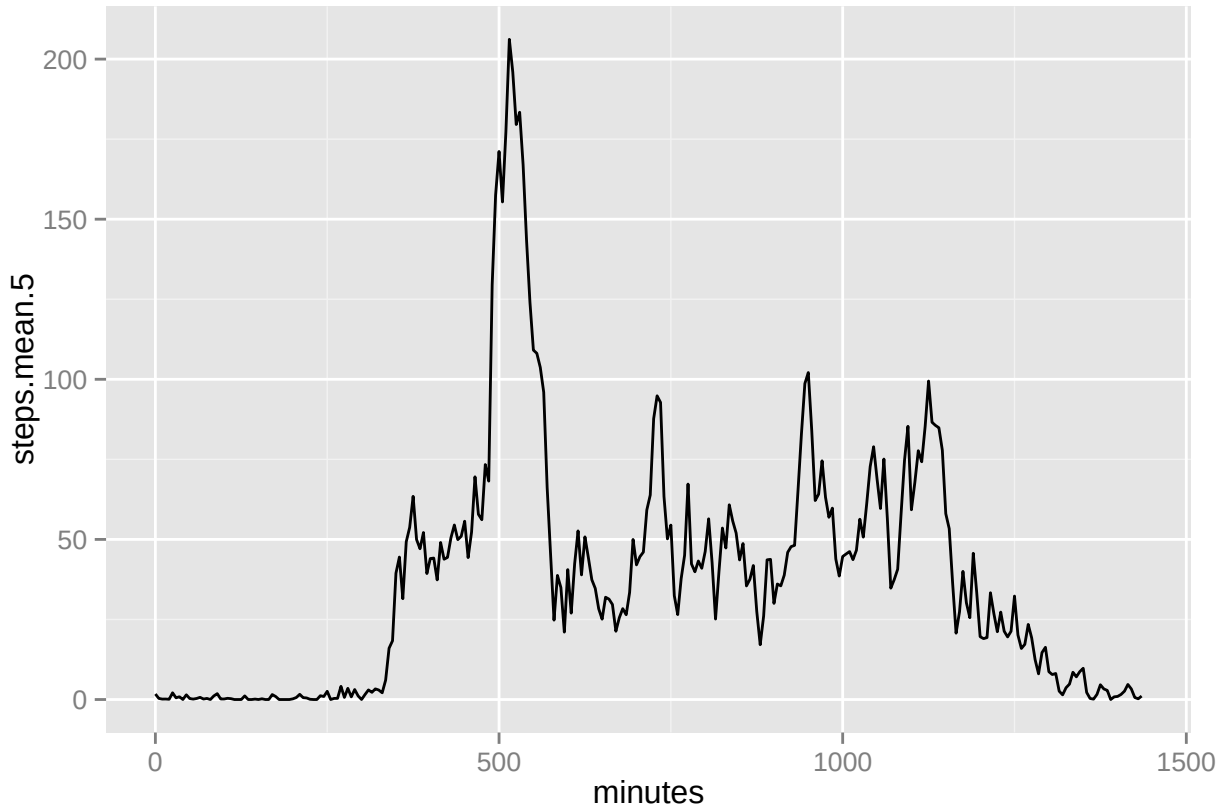
```
m1=mean(steps.day,na.rm=TRUE)
med1=median(steps.day,na.rm=TRUE)
```

The mean number of steps each day is then 1.0766 × 104 and the median 10765.

What is the average daily activity pattern?

We now make a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
minutes=seq(0,1435,5)
data.frame(steps.mean.5,minutes)->ts
ggplot(ts,aes(minutes,steps.mean.5))->g
g+geom_line()
```



And find the 5-minute interval which on average contains the maximum number of steps.

```
which.max(steps.mean.5)
```

```
## 835
## 104
```

```
maxsteps=5*which.max(steps.mean.5)-5
as.vector(maxsteps)
```

```
## [1] 515
```

So the 5 minite interval starts at 8.35 or equivalently from the minute 515 in the day.

Imputing missing values

We first notice that the data of 8 days are missing

```
missing.data=which(is.na(activity),arr.ind=T)[,1]
(missing.day=unique(activity[missing.data,2]))
```

```
## [1] 2012-10-01 2012-10-08 2012-11-01 2012-11-04 2012-11-09 2012-11-10
## [7] 2012-11-14 2012-11-30
## 61 Levels: 2012-10-01 2012-10-02 2012-10-03 2012-10-04 ... 2012-11-30
```

Moreover since

```
length(missing.day)*24*12-length(missing.data)
```

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

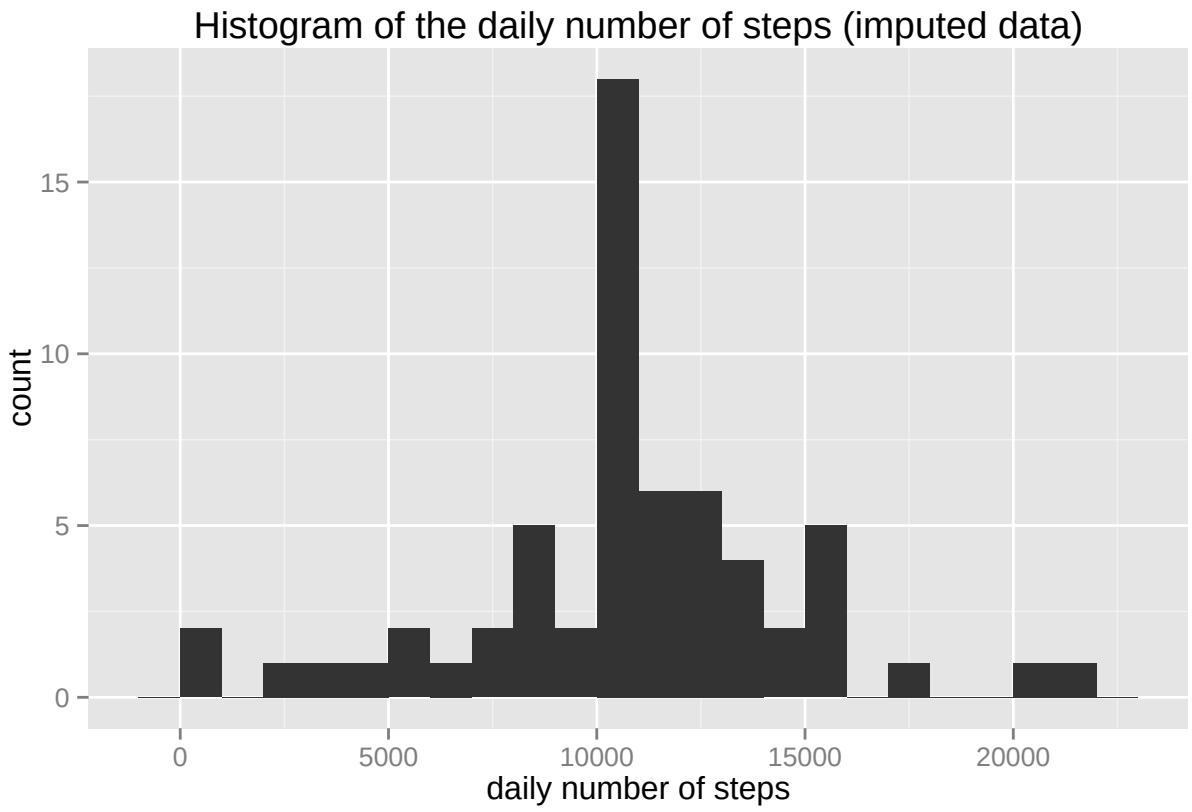
these days are completely missing.

So we decide to replace the missing data with the mean value of the corresponding five minute interval (we have no other information for that day) and costruct a data.frame which we named activity.I. We could also have used the median.

```
activity.I=activity
activity.I[missing.data,1]=steps.mean.5
activity.I[missing.data,1]=steps.mean.5
```

We now make a histogram of the total number of steps taken each day (with the imputed data) and calculate and report the mean and median total number of steps taken per day.

```
steps.mean.5.I=tapply(activity.I[,1],activity.I[,3],mean,na.rm=TRUE)
steps.day.I=tapply(activity.I[,1],activity.I[,2],sum)
qplot(steps.day.I, geom="histogram",xlab="daily number of steps" ,main="Histogram of the daily number o
```




```
mean(steps.day.I)
```

```
## [1] 10766
```

```
median(steps.day.I)
```

```
## [1] 10766
```

Notice that the change in the histogram is just due to the higher frequence of the class starting at 10000 steps. Mean and median are unchanged, the mean by construction. We notice that imputing missing data this way make them loss some variability.

```
var(steps.day.I)
```

```
## [1] 15795782
```

```
var(steps.day,na.rm=TRUE)
```

```
## [1] 18225902
```

Are there differences in activity patterns between weekdays and weekends?

We first 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.

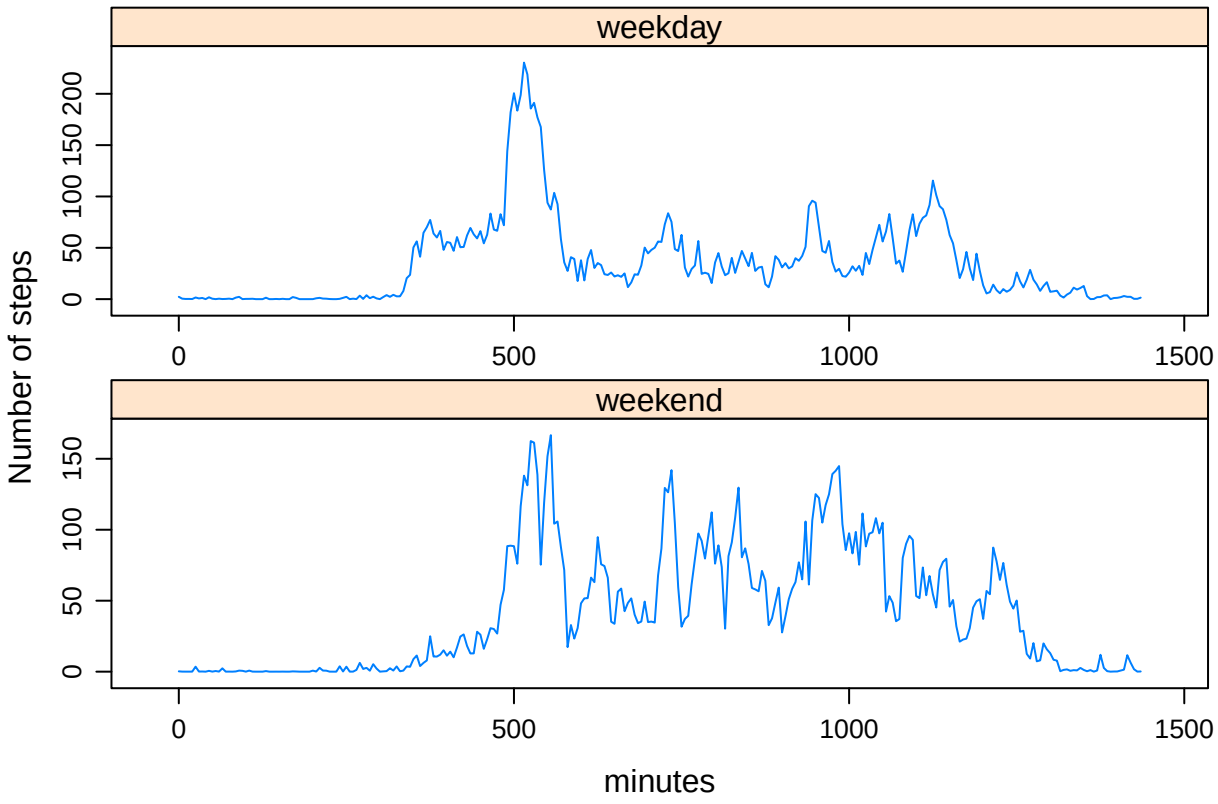
```
weekdays(as.Date(activity.I[,2]))->week_Day
weekends=c("Saturday","Sunday")
is.element(week_Day,weekends)->temp
temp2=which(temp=="TRUE")
temp3=temp
temp3[temp2]="weekend"
temp3[-temp2]="weekday"
table(temp3)
```

```
## temp3
## weekday weekend
##   12960    4608
```

```
data.frame(activity.I,temp3)->newdata
```

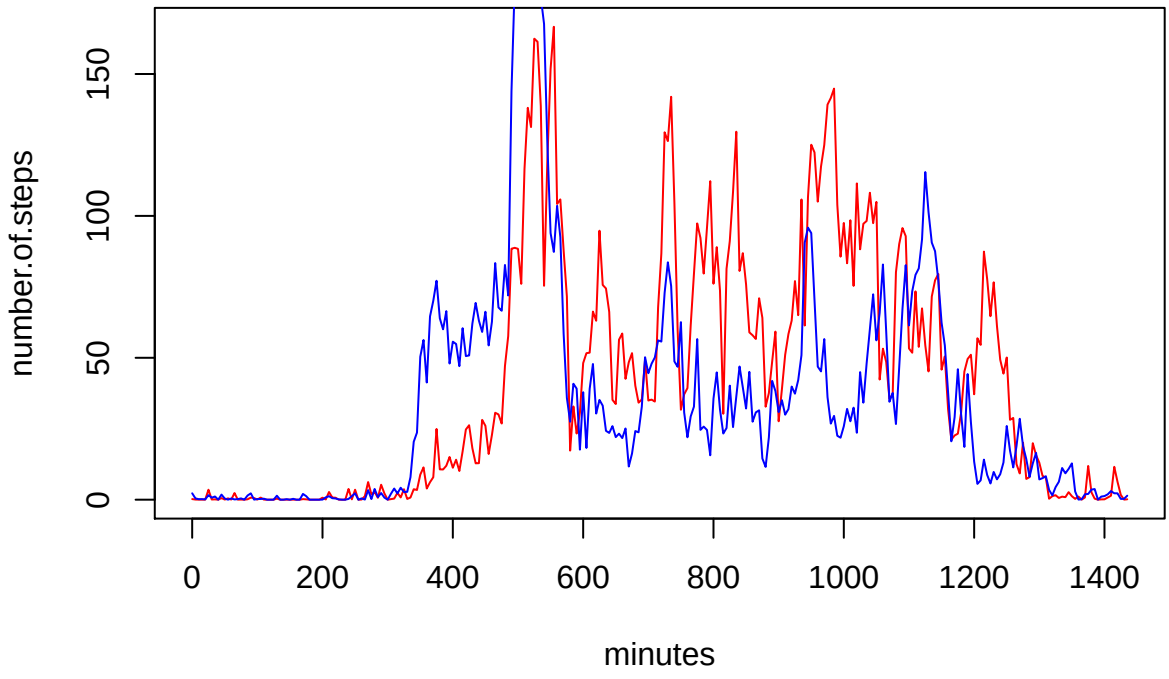
We make now a panel plot containing a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). The plot should look something like the following, which was creating using simulated data:

```
WE=tapply(newdata[temp2,1],newdata[temp2,3],mean,na.rm=TRUE)#c
WD=tapply(newdata[-temp2,1],newdata[-temp2,3],mean,na.rm=TRUE)
data.frame(WE,minutes,"weekend")->sI
names(sI)[1]="number of steps"
names(sI)[3]="day"
data.frame(WD,minutes,"weekday")->sII
names(sII)[1]="number of steps"
names(sII)[3]="day"
x=data.frame(rbind(sI,  sII))
#tidy.name.vector <- make.names(colnames(x), unique=TRUE)
library(lattice)
xyplot(number.of.steps ~ minutes | day,
       data = x, scales = "free", layout = c(1,2),
       auto.key = list(x = .6, y = .7, corner = c(0, 0)),type="l",ylab="Number of steps")
```



We notice a clear difference between weekdays and workdays that could be better seen by plotting both the graphs on the same plot.

```
plot(number.of.steps ~ minutes,data=x[1:nrow(sI),],type="l",col="red")
points(number.of.steps ~ minutes,data=x[-(1:nrow(sI)),],type="l",col="blue")
```



the activity in the weekends for instance. . . starts later!