Reproducible Research 1

Loading and preprocessing the data

```
setwd("C:/Users/jl339/Desktop/datascience/R")

activity_raw <- read.csv("activity.csv")
activity<-activity_raw
head(activity)
```

```
##   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?

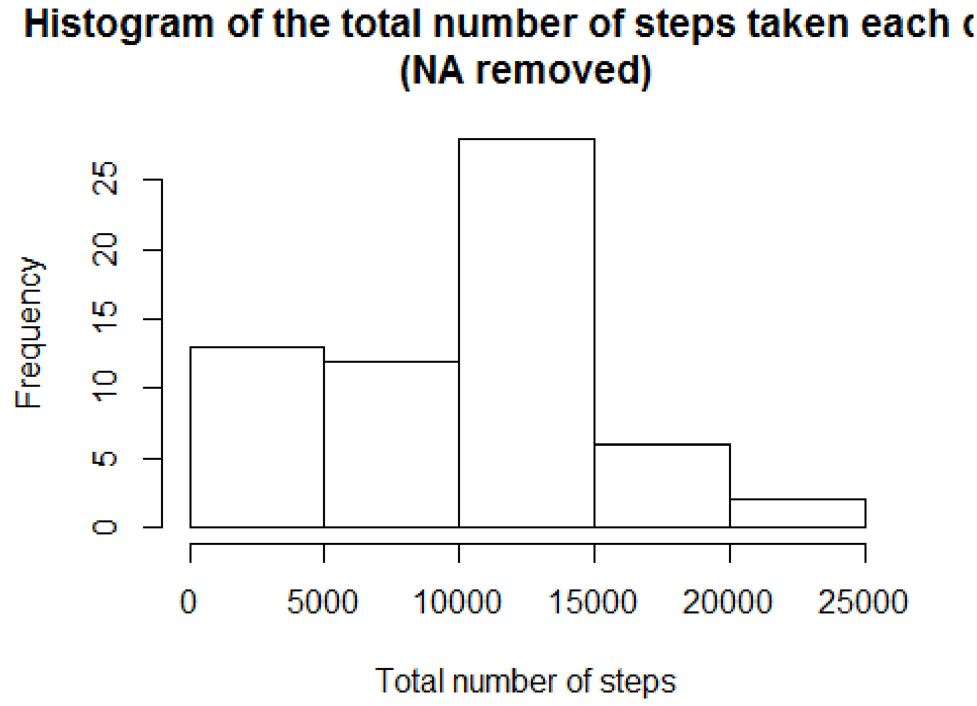
Calculate the total number of steps taken per day

```
stepsperday <- aggregate(activity$steps, by=list(activity$date), FUN=sum,
na.rm=TRUE)
names(stepsperday) <- c("date", "total")
head(stepsperday)
```

```
##         date total
## 1 2012-10-01     0
## 2 2012-10-02   126
## 3 2012-10-03 11352
## 4 2012-10-04 12116
## 5 2012-10-05 13294
## 6 2012-10-06 15420
```

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

```
hist(stepsperday$total,xlab="Total number of steps",main="Histogram of the
total number of steps taken each day\n(NA removed)" )
```



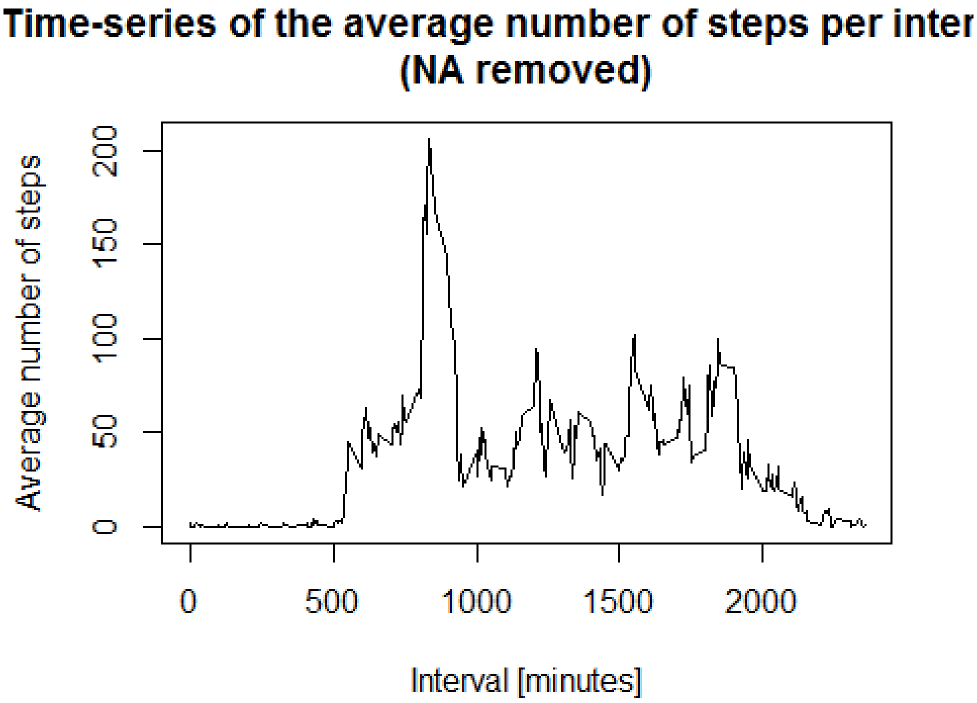
Calculate and report the mean and median of the total number of steps taken per day

```
mean(stepsperday$total)

## [1] 9354.23

median(stepsperday$total)

## [1] 10395
```

What is the average daily activity pattern?

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)

```
Fivemintavg<-aggregate(steps~interval,data=activity, FUN=mean,na.rm=TRUE)
plot(Fivemintavg,type="l",xlab="Interval [minutes]",
     ylab="Average number of steps",
     main="Time-series of the average number of steps per intervals\n(NA 
removed)")
```



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

```
maxpos<-which(Fivemintavg$steps==max(Fivemintavg$steps))
maxinterval<-Fivemintavg[maxpos,1]
maxinterval

## [1] 835
```

Imputing missing values

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

```
sum(is.na(activity$steps))

## [1] 2304
```

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.

```
napos<-which(is.na(activity$steps))

na.value<-rep(mean(activity$steps, na.rm=TRUE))
```

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

```r
activity[napos,"steps"]<-na.value

head(activity_raw)
```

```
##   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
```

```r
head(activity)
```

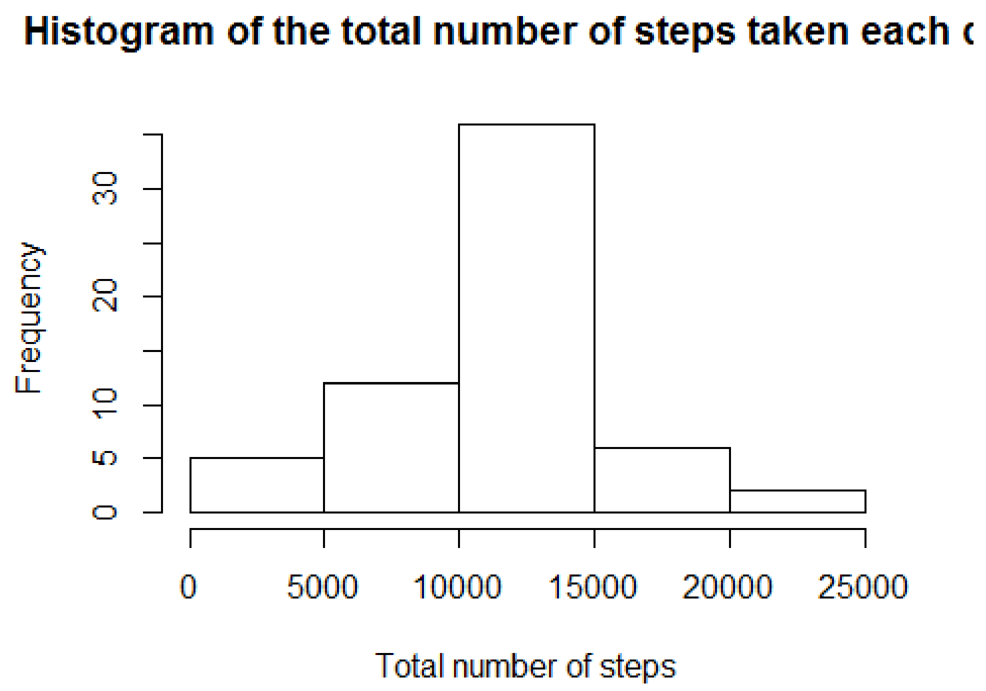
```
##     steps       date interval
## 1 37.3826 2012-10-01        0
## 2 37.3826 2012-10-01        5
## 3 37.3826 2012-10-01       10
## 4 37.3826 2012-10-01       15
## 5 37.3826 2012-10-01       20
## 6 37.3826 2012-10-01       25
```

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?

```r
stepsperday2 <- aggregate(activity$steps, by=list(activity$date), FUN=sum,
na.rm=TRUE)
names(stepsperday2) <- c("date", "total")
head(stepsperday2)
```

```
##         date    total
## 1 2012-10-01 10766.19
## 2 2012-10-02   126.00
## 3 2012-10-03 11352.00
## 4 2012-10-04 12116.00
## 5 2012-10-05 13294.00
## 6 2012-10-06 15420.00
```

```r
hist(stepsperday2$total,xlab="Total number of steps",main="Histogram of the
total number of steps taken each day" )
```



```
mean(stepsperday2$total)

## [1] 10766.19

median(stepsperday2$total)

## [1] 10766.19
```

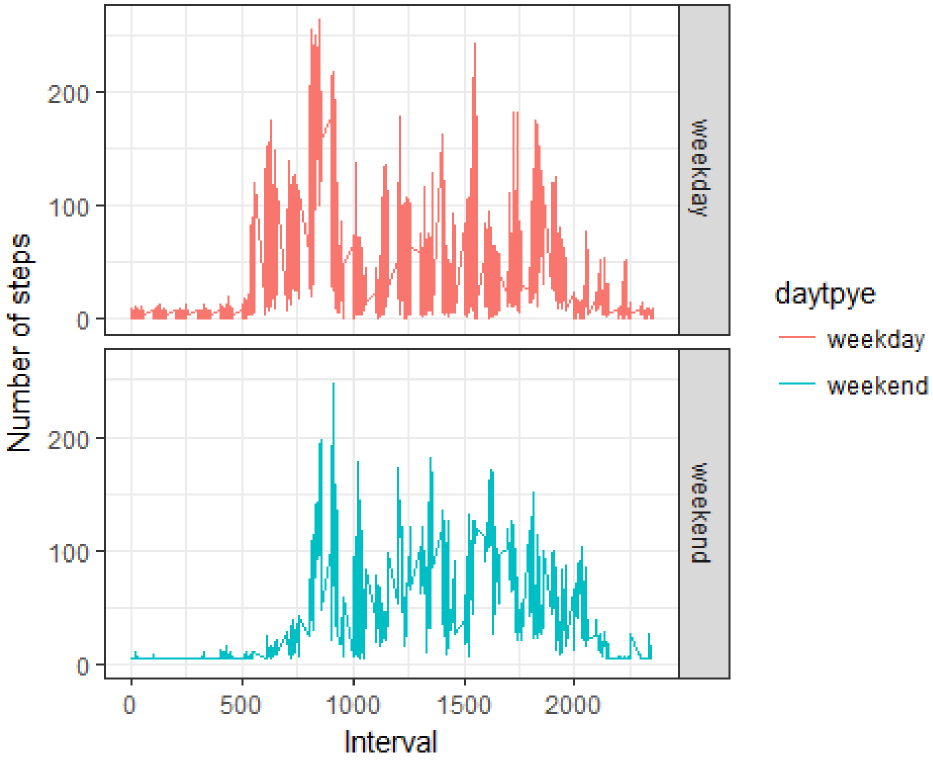
Are there differences in activity patterns between weekdays and weekends?

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.

```
activity$date <- as.POSIXct(activity$date, format="%Y-%m-%d")
newdata<-cbind(activity,weekdays.Date(activity$date))
names(newdata) <- c("steps", "date","interval","weekday")
newdata2 <- cbind(newdata, daytype=ifelse(newdata$weekday == "Saturday" |
newdata$weekday == "Sunday", "weekend", "weekday"))
head(newdata2)

##     steps       date interval weekday daytype
## 1 37.3826 2012-10-01        0  Monday weekday
## 2 37.3826 2012-10-01        5  Monday weekday
## 3 37.3826 2012-10-01       10  Monday weekday
## 4 37.3826 2012-10-01       15  Monday weekday
## 5 37.3826 2012-10-01       20  Monday weekday
## 6 37.3826 2012-10-01       25  Monday weekday
```

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.

```
library(ggplot2)
Fivemintavg2<-
aggregate(newdata2$steps,by=list(newdata2$daytype,newdata2$weekday,
newdata2$interval), mean)
names(Fivemintavg2) <- c("daytpye", "weekday","interval","mean")
ggplot(Fivemintavg2,aes(x=interval,y=mean,color=daytpye,group=daytpye),fil)+
geom_line() +   theme_bw() + guides(fill=FALSE)+ facet_grid(daytpye~.)+
labs(x="Interval", y="Number of steps")
```


Interval
Number of steps
weekday
weekend
daytpye
weekday
weekend