Reproducible Research: Project 1

lnma

4/27/2020

Load and preprocess data

1. Unzip data to csv file

```
url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
destfile <- "step_data.zip"
download.file(url, destfile)
unzip(destfile)
```

2. Load the data

```
activity <- read.csv("activity.csv", sep = ",")
```

3. Add a column of day that the first day is 2012-10-01

```
activity$date<-as.Date(as.character(activity$date))
activity$day<-activity$date-activity$date[1]+1
```

4. Have a look at the data

```
str(activity)
```

```
## 'data.frame':    17568 obs. of  4 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
##  $ day     : 'difftime' num  1 1 1 1 ...
##   ..- attr(*, "units")= chr "days"
```

```
head(activity)
```

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

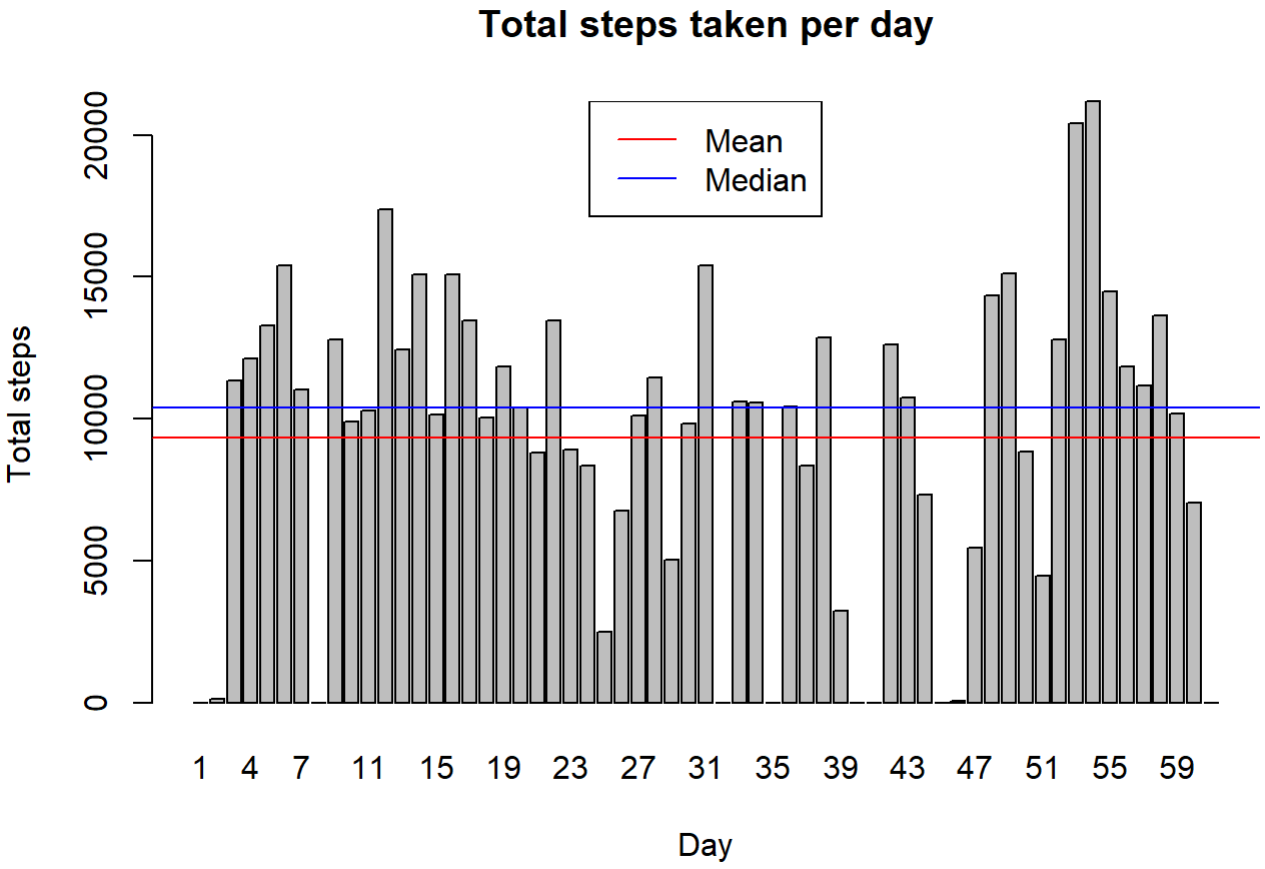
What is mean total number of steps taken per day?

1. Calculate the total number of steps taken per day

```
totalStep_day<-tapply(activity$steps,activity$day,sum,na.rm=TRUE)
```

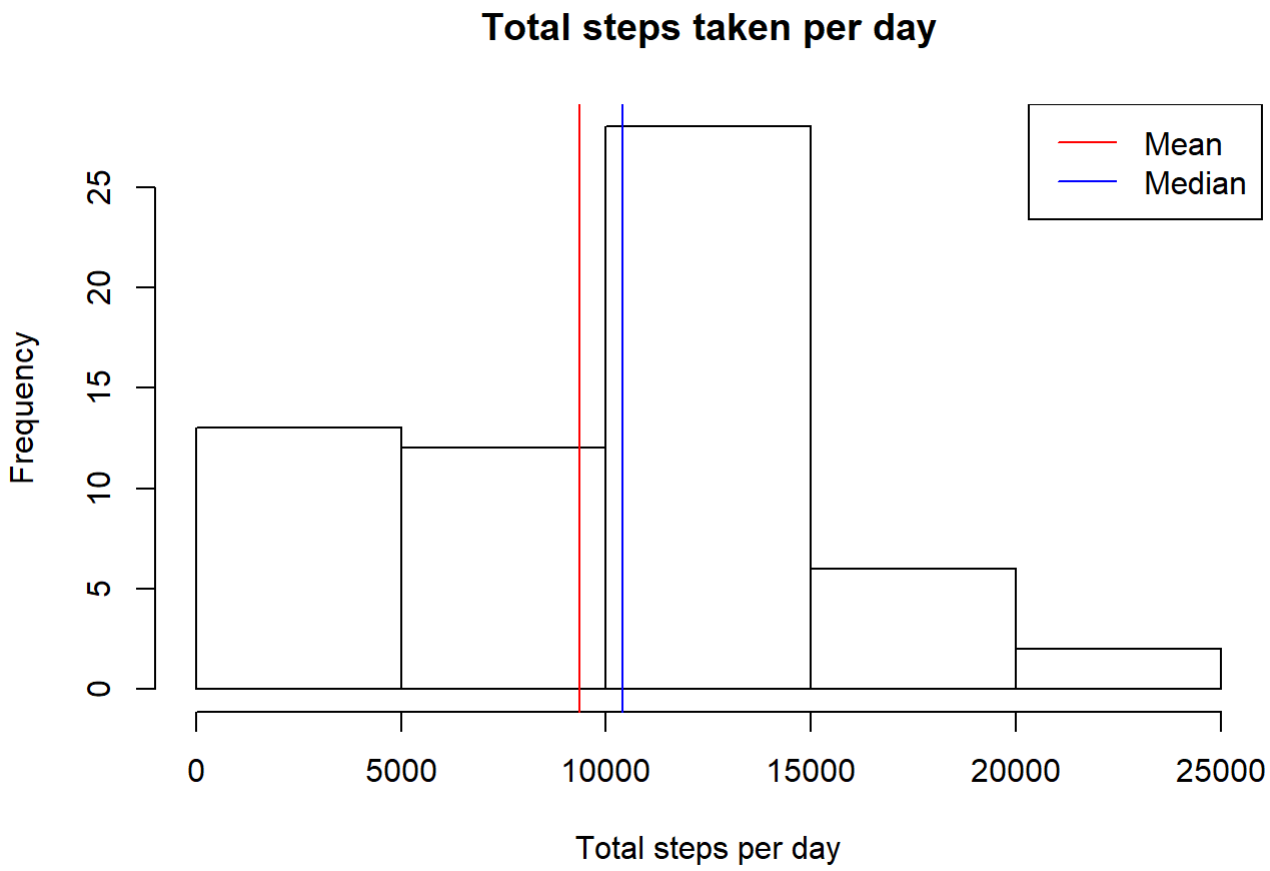
2. Barplot total number of steps taken for each day
   The mean and median values are plotted as red and blue lines respectively

```
barplot(totalStep_day,main='Total steps taken per day',
     xlab = 'Day',ylab='Total steps')
abline(h = mean(totalStep_day), lty = 1, lwd = 1, col = "red")
abline(h = median(totalStep_day), lty = 1, lwd = 1, col = "blue")
legend('top',c('Mean','Median'),col=c('red','blue'),lty = c(1,1),lwd=c(1,1))
```



3. Histogram of total number of steps taken for each day
   The mean and median values are plotted as red and blue lines respectively

```
hist(totalStep_day,main='Total steps taken per day',
        xlab = 'Total steps per day',ylab='Frequency')
#add lines of mean and median in red and blue respectively
abline(v = mean(totalStep_day), lty = 1, lwd = 1, col = "red")
abline(v = median(totalStep_day), lty = 1, lwd = 1, col = "blue")
legend('topright',c('Mean','Median'),col=c('red','blue'),lty = c(1,1),lwd=c(1,1))
```



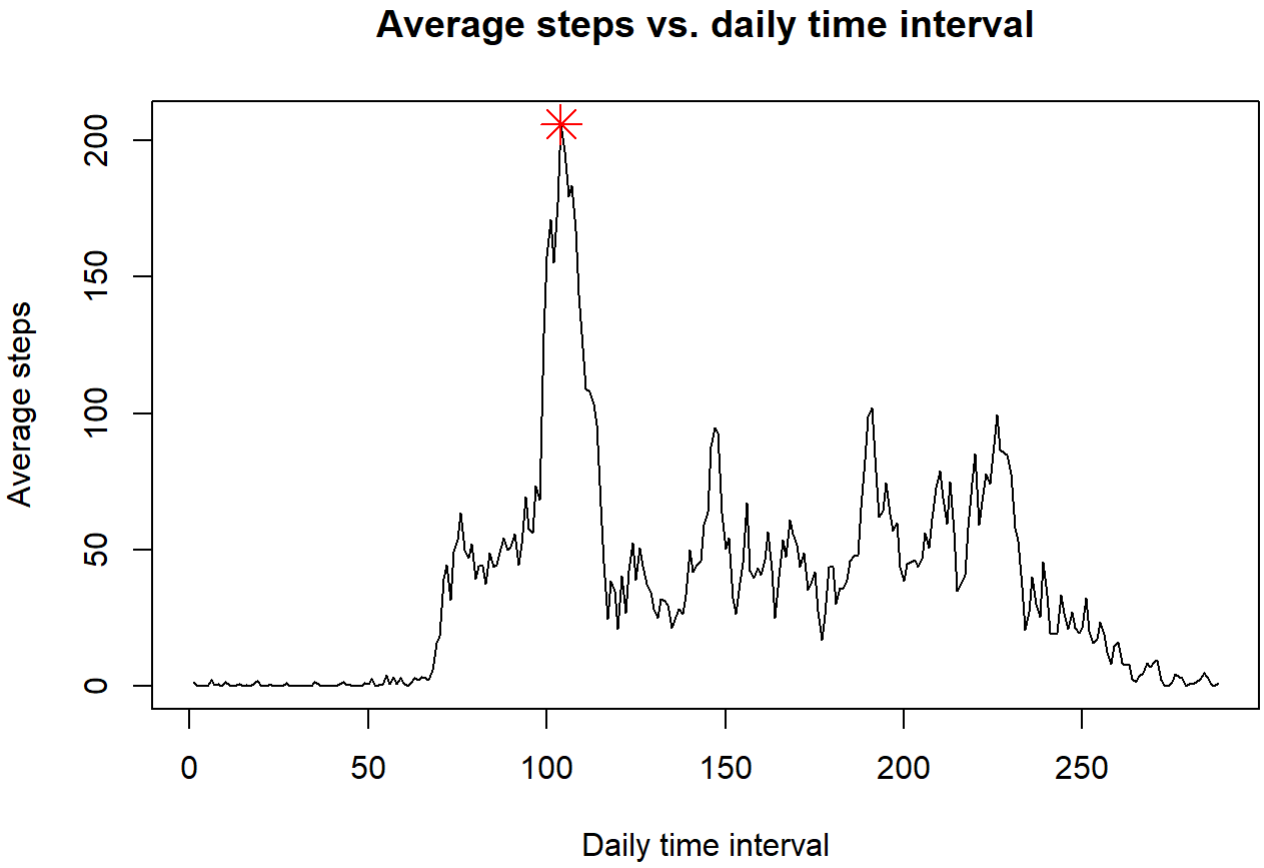
What is the average daily activity pattern?

1. Calculate the average number of steps taken per interval

```
avgStep_interval<-tapply(activity$steps,activity$interval,mean,na.rm=TRUE)
```

2. 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), the maxium point is highlighted

```
plot(avgStep_interval,ty='l',main='Average steps vs. daily time interval',
     xlab = 'Daily time interval',ylab = 'Average steps')
#maximum average number of steps
x_max=as.numeric(which(avgStep_interval==max(avgStep_interval)))
y_max=max(avgStep_interval)
p<-c(round(x_max),round(y_max))
points(t(p),pch=8,col='red',cex=2)
```



3.Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?
Interval (835) contains on average the maximum number of steps (206.1698).

```
x_max=which(avgStep_interval==max(avgStep_interval))
y_max=max(avgStep_interval)
x_max
```

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

```
y_max
```

```
## [1] 206.1698
```

Imputing missing values

1.Calculate and report the total number of missing values in the dataset

```
na_number_steps<-sum(is.na(activity$steps))
na_number_steps
```

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

2. Devise a strategy for filling in all of the missing values in the dataset. The strategy uses the mean mean for that 5-minute interval.

```
activity_day<-split(activity,activity$day)
avgStep_day<-as.numeric(tapply(activity$steps,activity$day,mean,na.rm=TRUE))
activity_NA<-data.frame()
for(i in 1:length(avgStep_day)){
        if (sum(!is.na(activity_day[[i]][,1]))==0){
                activity_day[[i]]$steps<-avgStep_interval
        }else if(sum(is.na(activity_day[[i]]$steps))!=0){
                for (j in 1:length(activity_day[[i]]$steps)){
                        activity_day[[i]][j,1]<-avgStep_interval[j]
                }
        }
        activity_NA<-rbind(activity_NA,activity_day[[i]])
}
```

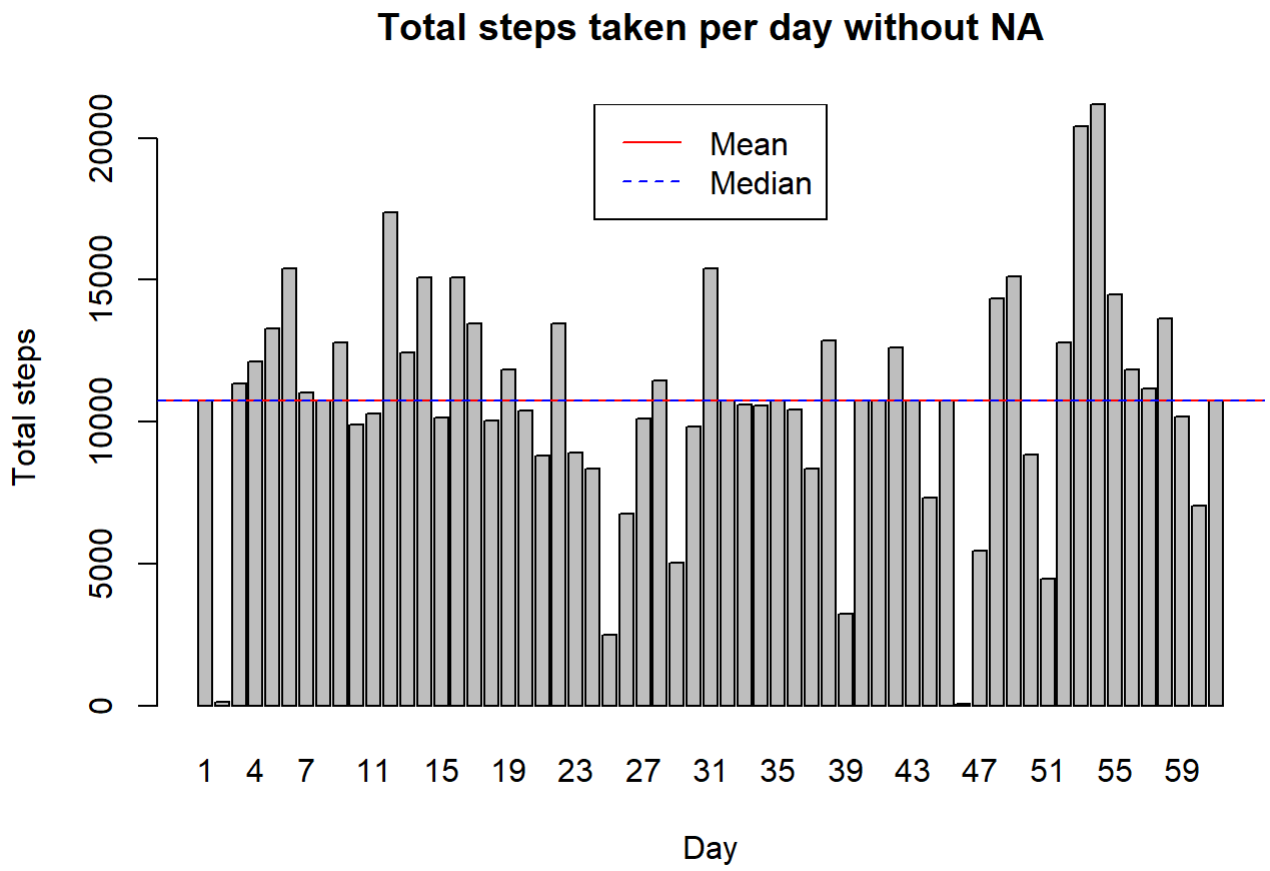
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
head(activity_NA)
```

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

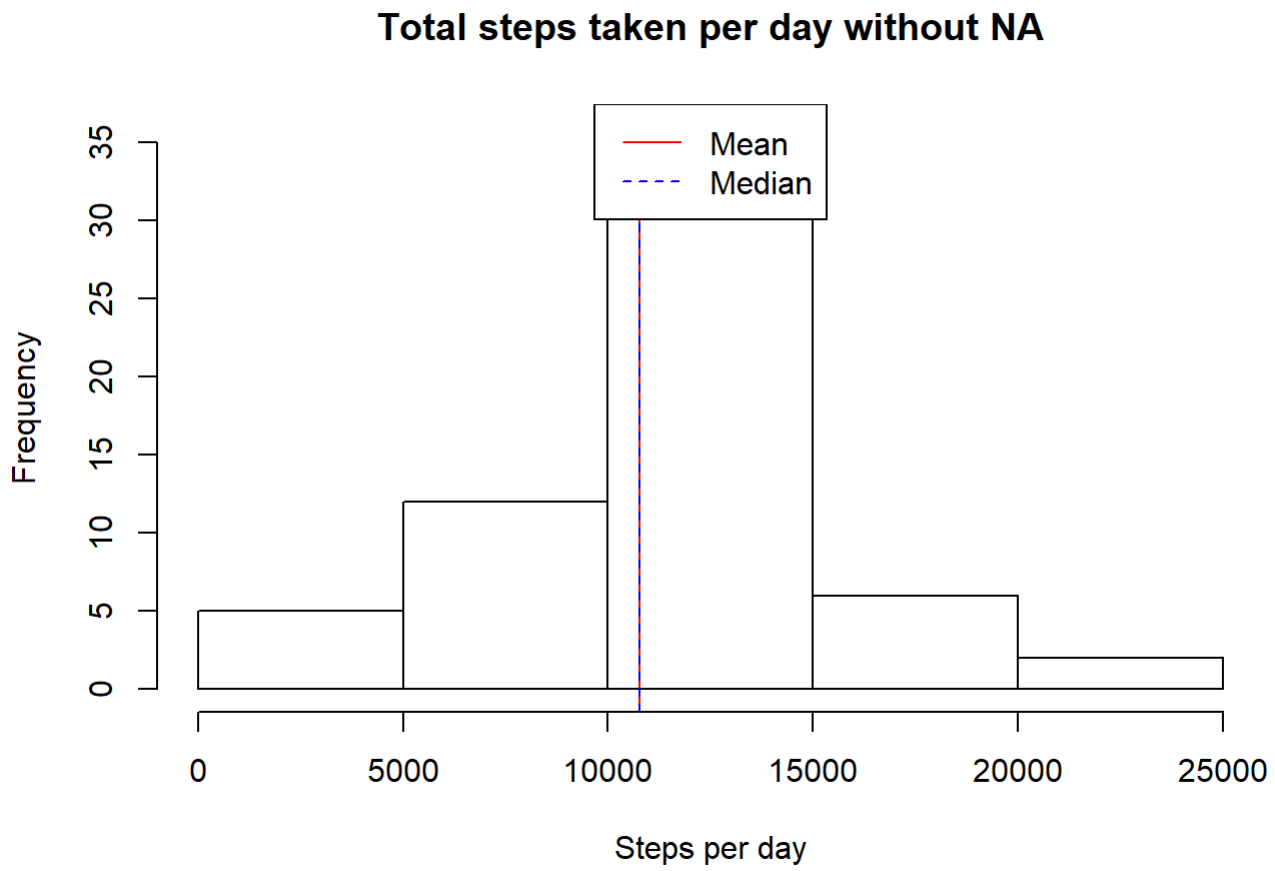
4. Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?
   Barplot total number of steps taken for each day without NAs

```
totalStep_NA<-tapply(activity_NA$steps,activity_NA$day,sum,na.rm=TRUE)
barplot(totalStep_NA,main='Total steps taken per day without NA',
        xlab = 'Day',ylab='Total steps')
abline(h = mean(totalStep_NA), lty = 1, lwd = 1, col = "red")
abline(h = median(totalStep_NA), lty = 2, lwd = 1, col = "blue")
legend('top',c('Mean','Median'),col=c('red','blue'),lty = c(1,2),lwd=c(1,1))
```



Histogram of total number of steps taken for each day without NAs

```
hist(totalStep_NA,main='Total steps taken per day without NA',
        xlab = 'Steps per day',ylab='Frequency')
abline(v = mean(totalStep_NA), lty = 1, lwd = 1, col = "red")
abline(v = median(totalStep_NA), lty = 2, lwd = 1, col = "blue")
legend('top',c('Mean','Median'),col=c('red','blue'),lty = c(1,2),lwd=c(1,1))
```



Imputing missing values, mean of the total number of steps taken per day increased while median decreased. The strategy leads to equal value of mean and median of the total number of steps per day

Are there differences in activity patterns between weekdays and weekends?

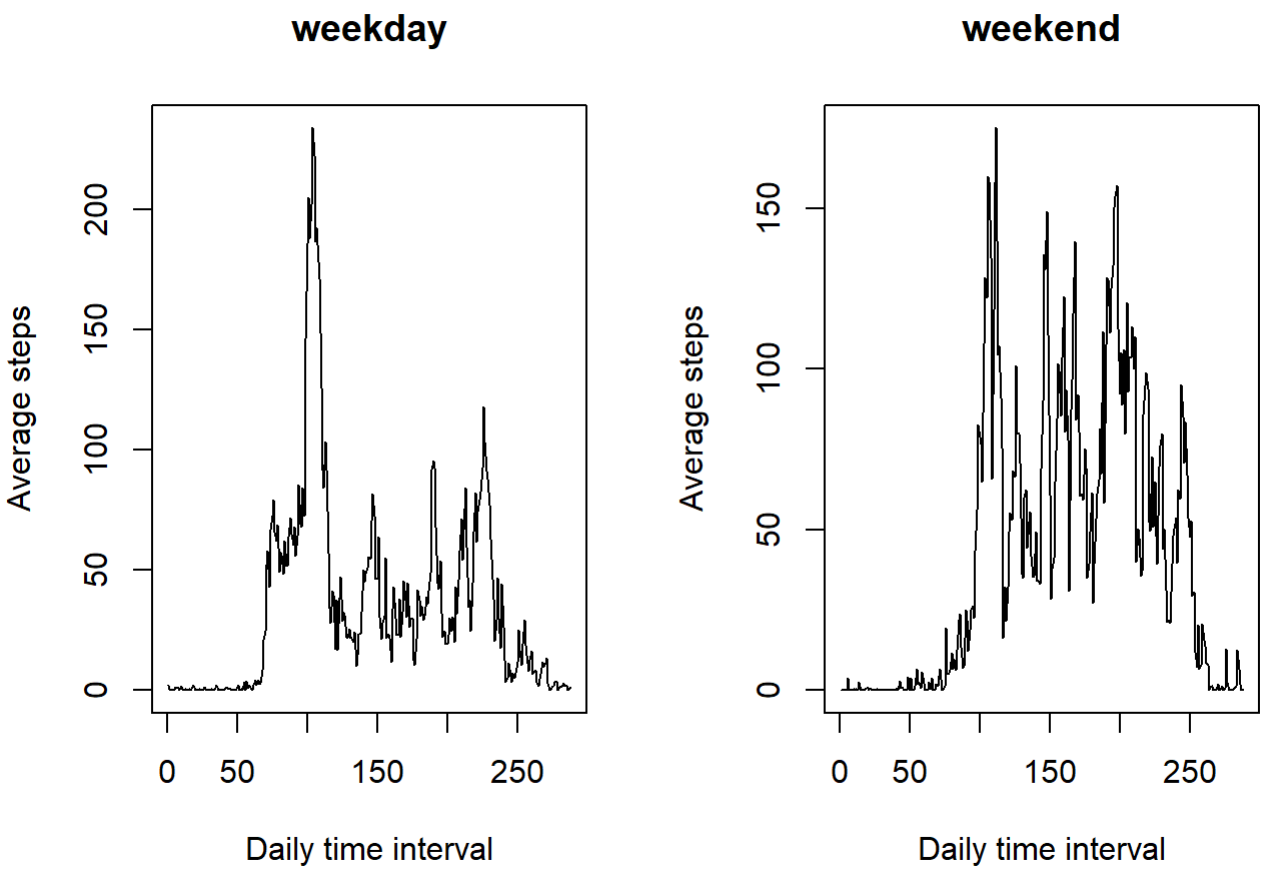
1. Create a new factor variable in the dataset with two levels – “weekday” and “weekend” indicating whether a given date is a weekday or weekend day.

```
activity$week<-weekdays(activity$date)
activity$week<-factor(activity$week,
                      levels = c('Monday','Tuesday','Wednesday','Thursday','Friday','Saturday',
'Sunday'),
                      ordered = TRUE)
activity$weekday[activity$week%in%c('Monday','Tuesday','Wednesday','Thursday','Friday')]<-'weekd
ay'
activity$weekday[activity$week%in%c('Saturday','Sunday')]<-'weekend'

#average number of steps for weekdays and weekends
activity$weekday<-factor(activity$weekday,levels = c('weekday','weekend'))
avgStep_wd<-tapply(activity[activity$weekday=='weekday','steps'],
                         activity[activity$weekday=='weekday','interval'],mean,na.rm=TRUE)
avgStep_we<-tapply(activity[activity$weekday=='weekend','steps'],
                   activity[activity$weekday=='weekend','interval'],mean,na.rm=TRUE)
```

2. Make 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).

```
par(mfrow=c(1,2))
plot(avgStep_wd,ty='l',main='weekday',xlab = 'Daily time interval',ylab = 'Average steps')
plot(avgStep_we,ty='l',main='weekend',xlab = 'Daily time interval',ylab = 'Average steps')
```




```
par(mfrow=c(1,1))
```