Assignment 6

Xin Gao

11/14/2020

Q1. Loading and preprocessing the data

```
setwd("C:/Users/xgao/Desktop/R/Assignment 6")
act<-read.csv("activity.csv")
str(act)
```

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

Q2. What is mean total number of steps taken per day?

Histogram of the total number of steps taken each day:

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

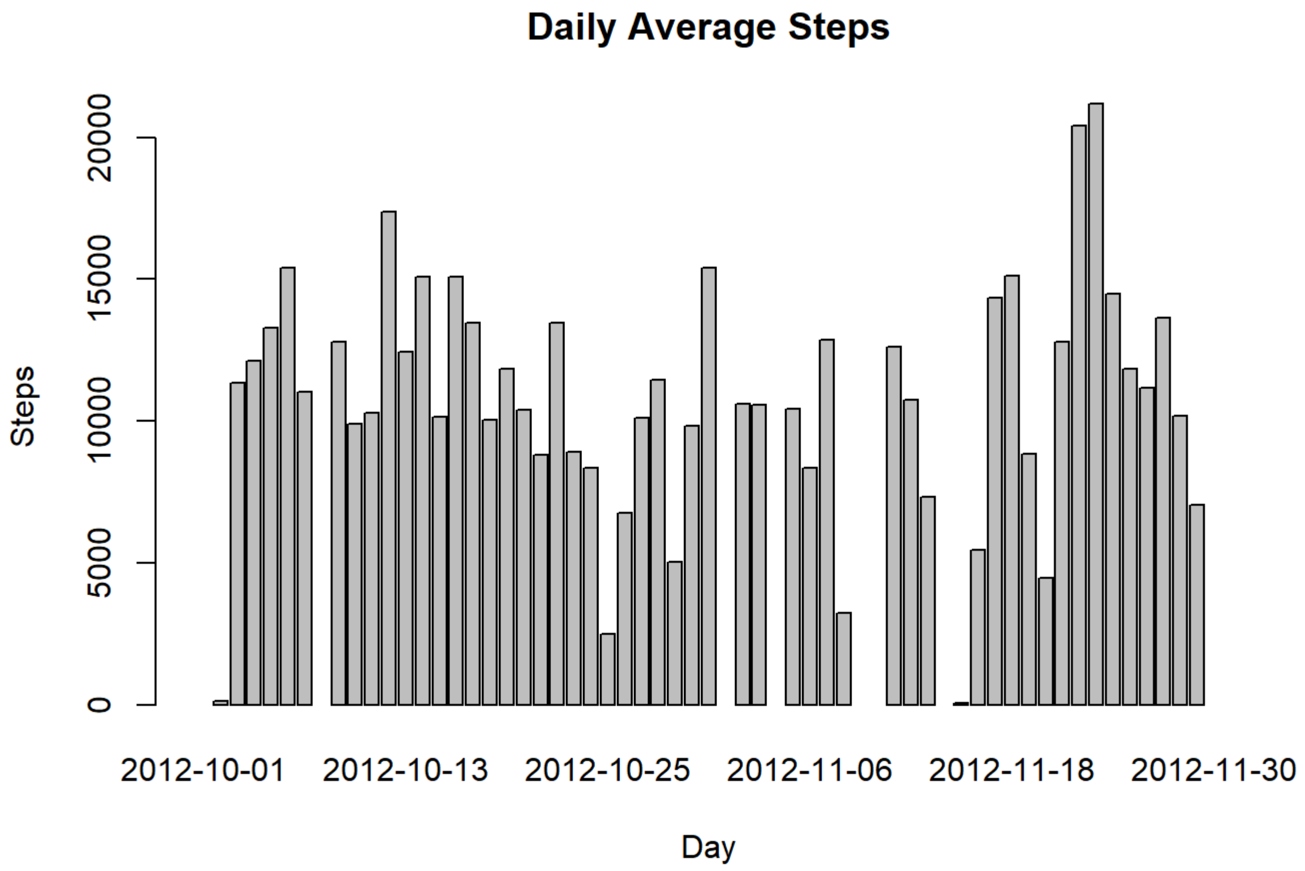
```
##
## Attaching package: 'dplyr'
```

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

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

```
act$Date<-as.Date(act$date, "%Y-%m-%d")
daily<-with(act,tapply(steps,Date,sum))

barplot(daily, main = "Daily Average Steps",
        xlab = "Day",
        ylab = "Steps")
```



Mean and median number of steps taken each day

```
mean(daily, na.rm=TRUE)
```

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

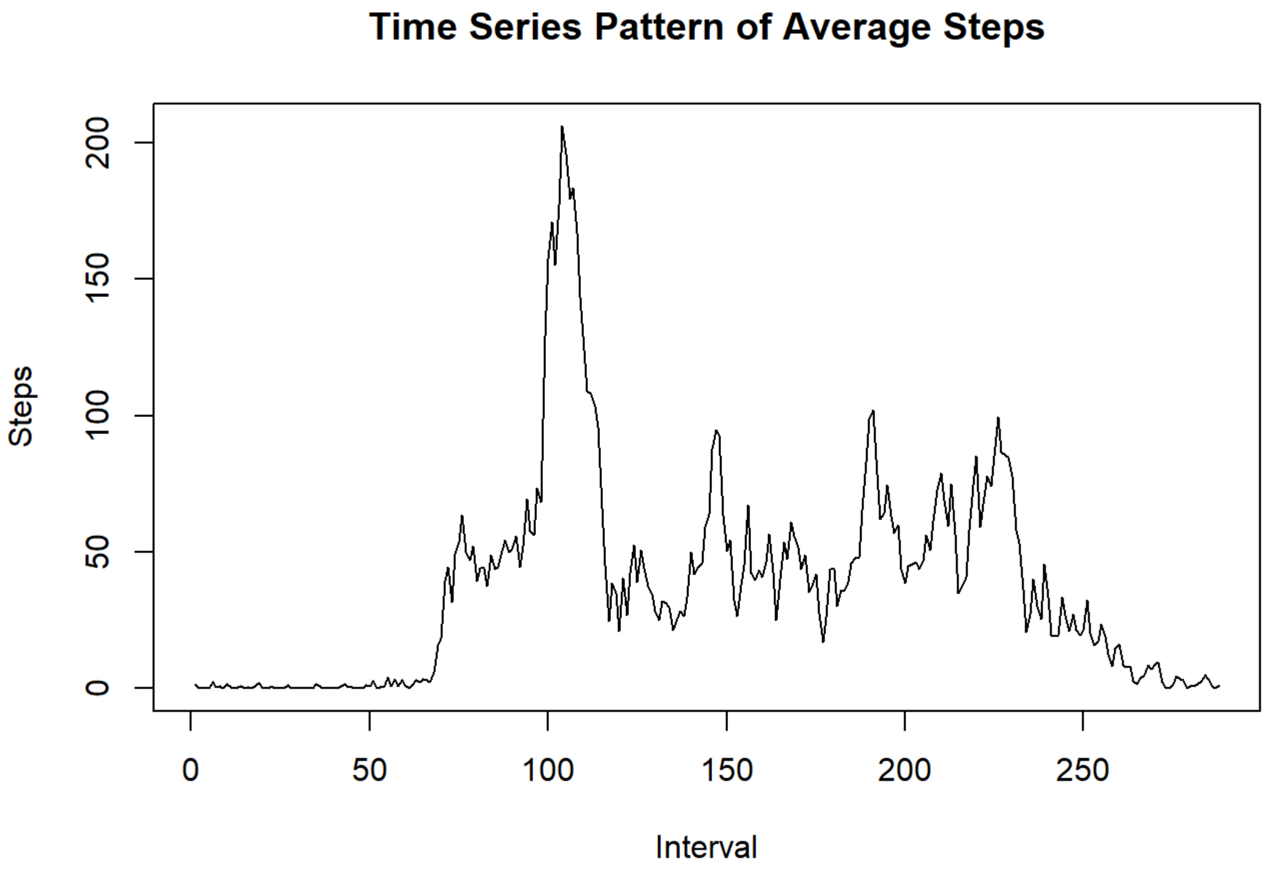
```
median(daily, na.rm=TRUE)
```

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

Q3. What is the average daily activity pattern?

Time series plot of the daily activity pattern:

```
pattern<-with(act,tapply(steps,interval, mean, na.rm=TRUE))
plot(pattern, type = "l", main="Time Series Pattern of Average
Steps",xlab="Interval",ylab="Steps")
```



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

```
df<-as.data.frame(pattern)
max<-max(pattern)
hi<-df[which(df$pattern==max),]
hi
```

```
##      835
## 206.1698
```

Q4. Imputing missing values

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

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

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

2. Devise a strategy for filling in all of the missing values in the dataset – I decide to replace missing values with the mean.

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

```
average<-mean(act$steps, na.rm=TRUE)
```

```
average
```

```
## [1] 37.3826
```

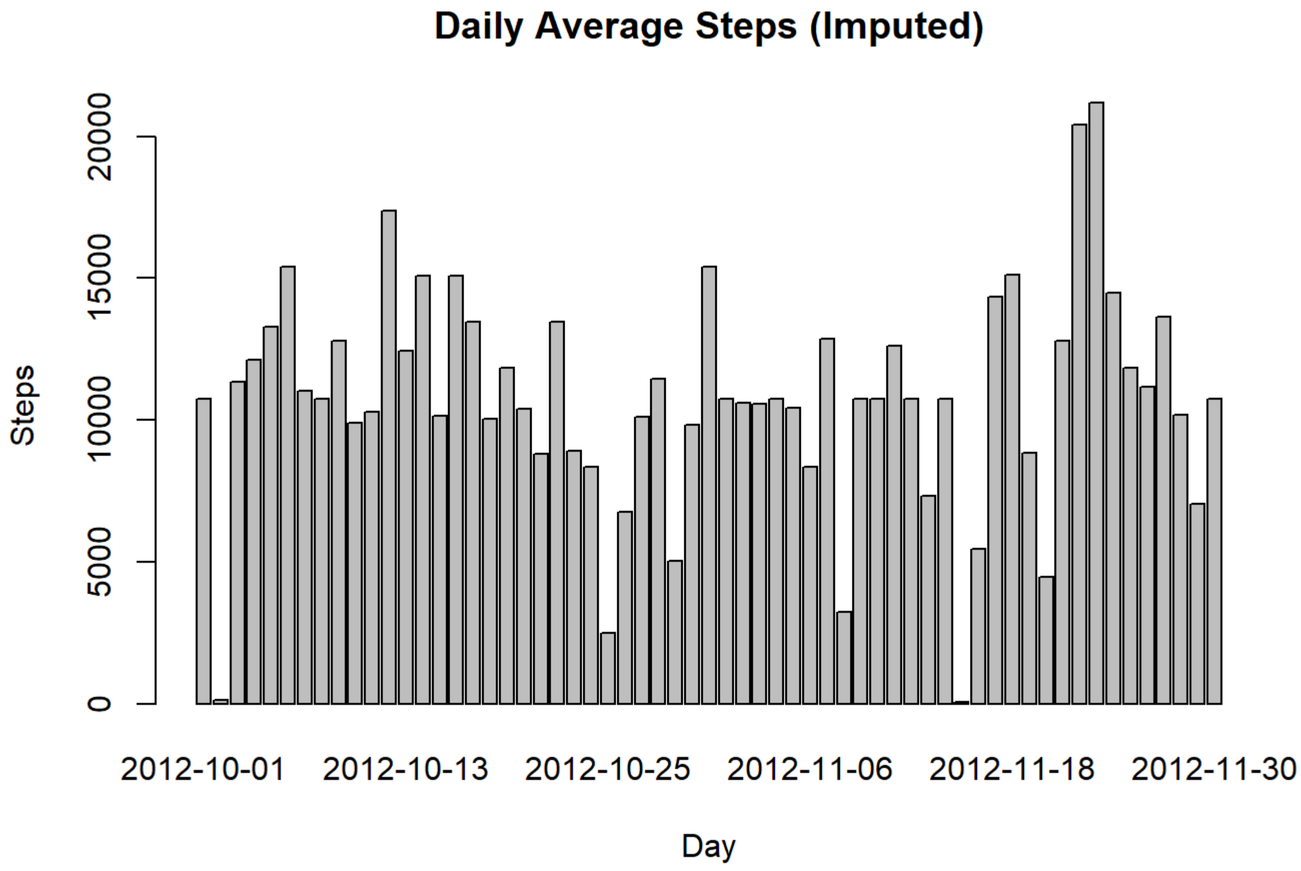
```
act$steps[is.na(act$steps)] <- average
head(act)
```

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

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

```
daily2<-with(act,tapply(steps,Date,sum))

barplot(daily2, main = "Daily Average Steps (Imputed)",
        xlab = "Day",
        ylab = "Steps")
```



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?

```
mean(daily2, na.rm=TRUE)
```

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

```
median(daily2, na.rm=TRUE)
```

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

We can see the mean is the same as the estimates from the first part, because I imputed missing values using the mean. The median is slightly different. The impact of imputing missing data is very small.

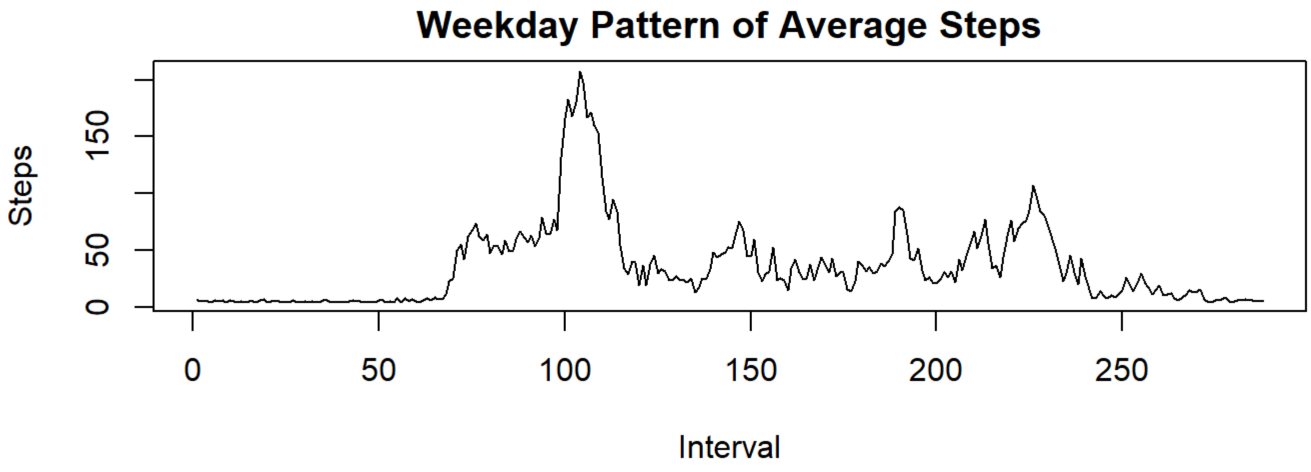
Q6. Are there differences in activity patterns between weekdays and weekends?

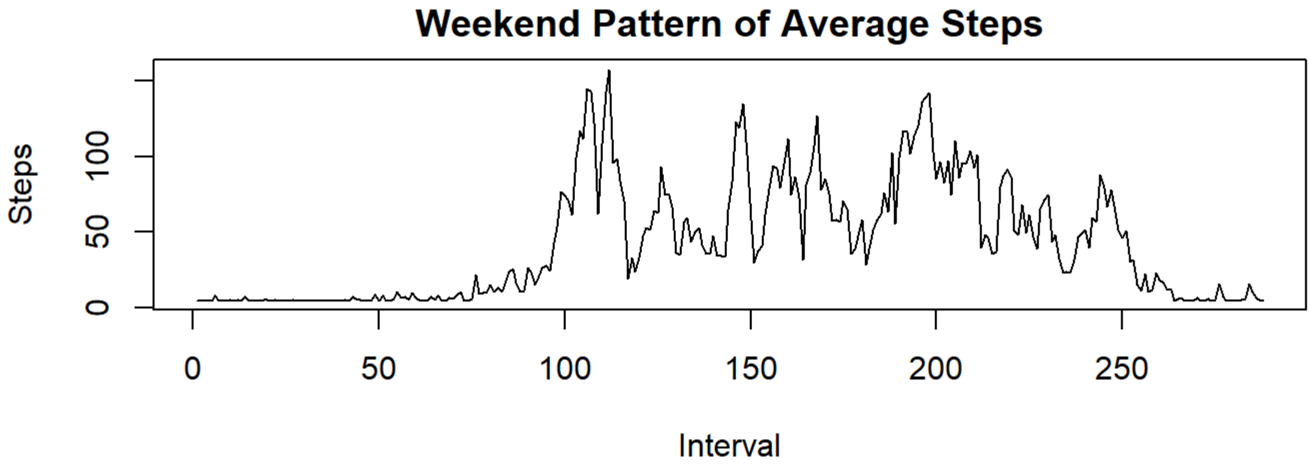
Create a new factor variable in the dataset with two levels – “weekday” and “weekend” indicating whether a given date is a weekday or weekend day.

```
act$week<-weekdays(act$Date)
act$weekend<-ifelse(act$week=="Saturday"|act$week=="Sunday",1,0)
```

Make a panel plot containing a time series plot by weekday and weekend:

```
act0<-subset(act,act$weekend==0)
act1<-subset(act,act$weekend==1)
pattern0<-with(act0,tapply(steps,interval, mean, na.rm=TRUE))
pattern1<-with(act1,tapply(steps,interval, mean, na.rm=TRUE))
par(mfrow=c(2,1), mar=c(4,4,2,1))
plot(pattern0, type = "l", main="Weekday Pattern of Average
Steps",xlab="Interval",ylab="Steps")
plot(pattern1, type = "l", main="Weekend Pattern of Average
Steps",xlab="Interval",ylab="Steps")
```


Weekday Pattern of Average Steps
Steps
0
50
150
0
50
100
150
200
250
Interval


Weekend Pattern of Average Steps
Steps
0
50
100
0
50
100
150
200
250
Interval