Activity Monitoring

Mari Coonley

January 8, 2017

SUMMARY

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the "quantified self" movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

The data is from a device that collects information at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

The data can be downloaded from the course web site or GitHub repo:

Dataset: Activity monitoring data [52K]

The variables included in this dataset are:

steps: Number of steps taking in a 5-minute interval (missing values are coded as NA)

date: The date on which the measurement was taken in YYYY-MM-DD format

interval: Identifier for the 5-minute interval in which measurement was taken

The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observations in this dataset.

Loading and preprocessing the data

```
activity<-read.csv("./activity.csv")
activity$date<-as.Date(as.character(activity$date, format = "%Y-%m-%d"))
activity$steps<-as.numeric(activity$steps)
summary(activity)
```

```
##      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
```

```
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : num  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 ...
```

What is the mean number of steps taken per day?

1) Calculate the total number of steps taken per day

```
date<-aggregate(steps~date, activity, sum)
str(date)
```

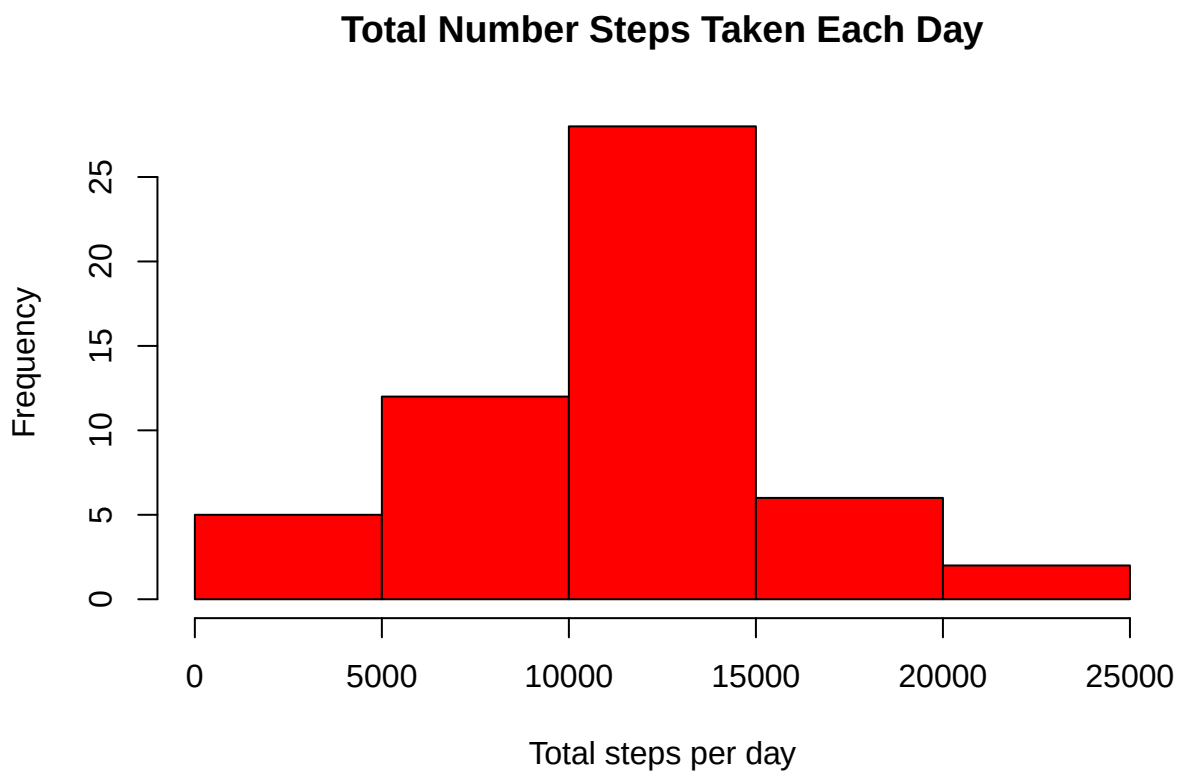
```
## 'data.frame':    53 obs. of  2 variables:
##  $ date : Date, format: "2012-10-02" "2012-10-03" ...
##  $ steps: num  126 11352 12116 13294 15420 ...
```

```
head(date)
```

```
##         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
```

2) If you do not understand the difference between a histogram and a bar plot, research the difference between them. Make a histogram of the total number of steps taken each day.

```
hist(date$steps, xlab = "Total steps per day", main = "Total Number Steps Taken Each Day", col = 'red')
```



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

```
origmean<-mean(date$steps, na.rm = TRUE)
origmean
```

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

```
origmed<-median(date$steps, na.rm= TRUE)
origmed
```

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

What is the average daily activity pattern?

1) 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).

```
time<-aggregate(steps~interval, activity, mean)
str(time)
```

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

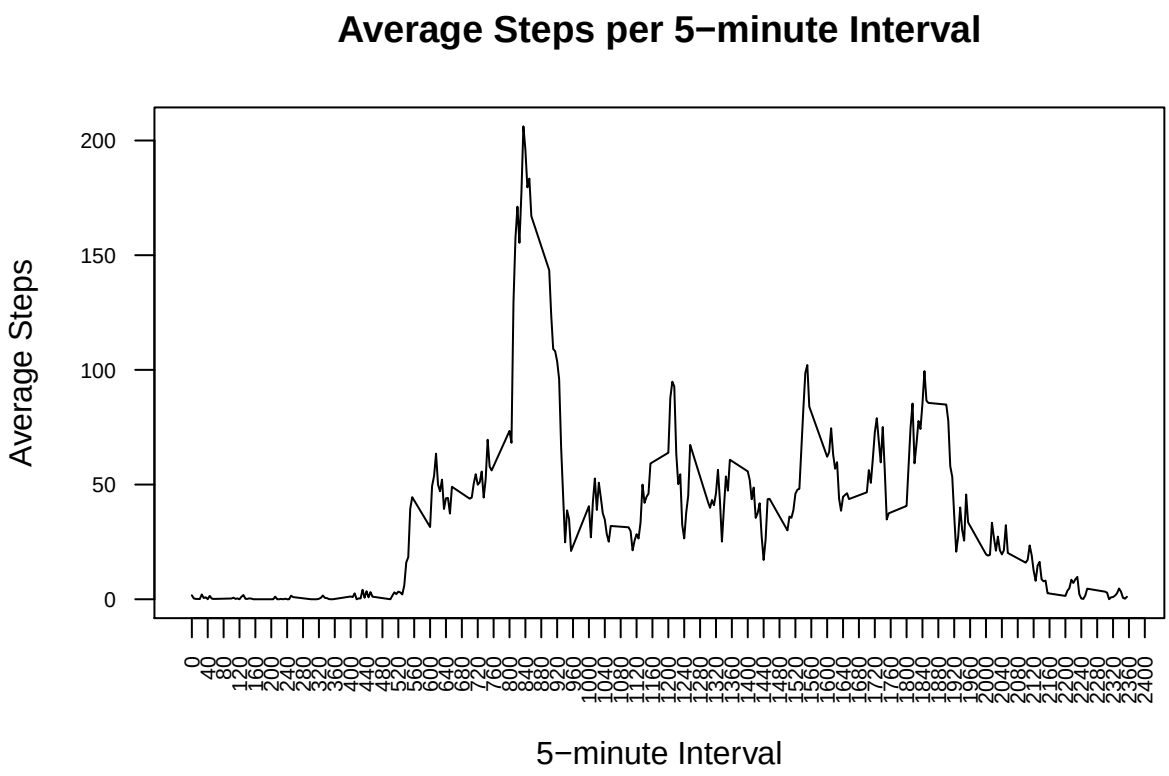
```
summary(time)
```

```
##     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
```

```
head(time)
```

```
##   interval     steps
## 1        0 1.7169811
## 2        5 0.3396226
## 3       10 0.1320755
## 4       15 0.1509434
## 5       20 0.0754717
## 6       25 2.0943396
```

```
with(time, plot(steps~interval, type = "l", main ="Average Steps per 5-minute Interval", xlab = "5-minu
```

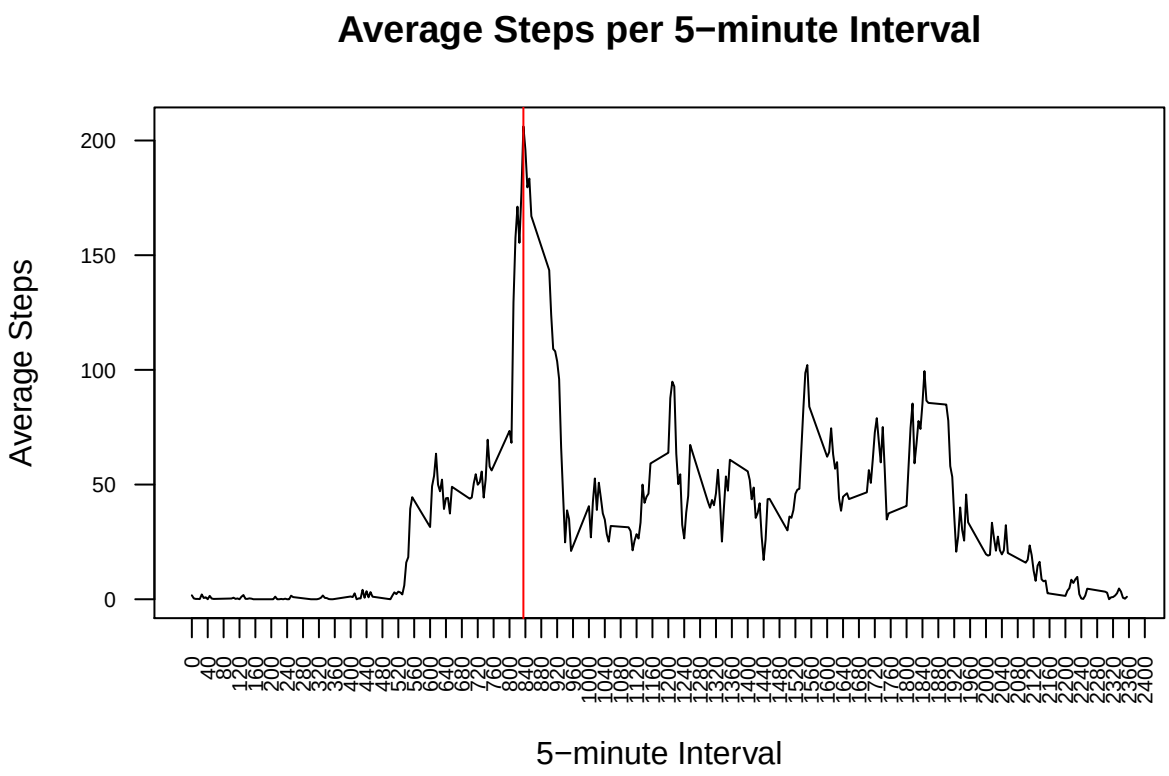


2) Which 5-minute interval, on average across all the days in dataset, contains the maximum number of steps?

```
maxintv<-time [ which(time$steps == max(time$steps)),]
maxintv
```

```
##     interval    steps
## 104      835 206.1698
```

```
with(time, plot(steps~interval, type = "l", main ="Average Steps per 5-minute Interval", xlab = "5-minu
abline(v=835, col="red")
```



Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

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

Running the ***summary ()*** function on the original data set *activity* has already given us this information. Note the last item in the steps column is NA's. There are 2304 rows with NA's, which is equal to 8 days.

```
summary(activity)
```

```
##      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
```

Reviewing the *date* dataset, used to originally calculate the total number steps per day, there are only 53 observations. Recall, the ***aggregate()*** function ignores NAs and the original dataset is composed of 61 days,

the difference in the two sets of total dates verifies that 8 entire days of data is missing, or 2304 rows of observations.

```
str(date)
```

```
## 'data.frame':    53 obs. of  2 variables:
##  $ date : Date, format: "2012-10-02" "2012-10-03" ...
##  $ steps: num  126 11352 12116 13294 15420 ...
```

2) 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.

```
newActivity<-data.frame("steps"=numeric(17568), "interval"= numeric(17568))
for (i in 1:288) {
    newActivity$steps [i] = round(mean(activity$steps[(i+288):(i+575)]), digits = 0)
   }

for (i in 289:17568) {
  if (activity$steps [i] %in% NA) {
    newActivity$steps [i] = round(mean(newActivity$steps[(i-288):(i-1)]), digits=0)
   } else {
   newActivity$steps [i] = activity$steps [i]
   }
}
```

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

```
newActivity$interval<-activity$interval
newActivity$date<-activity$date
summary(newActivity)
```

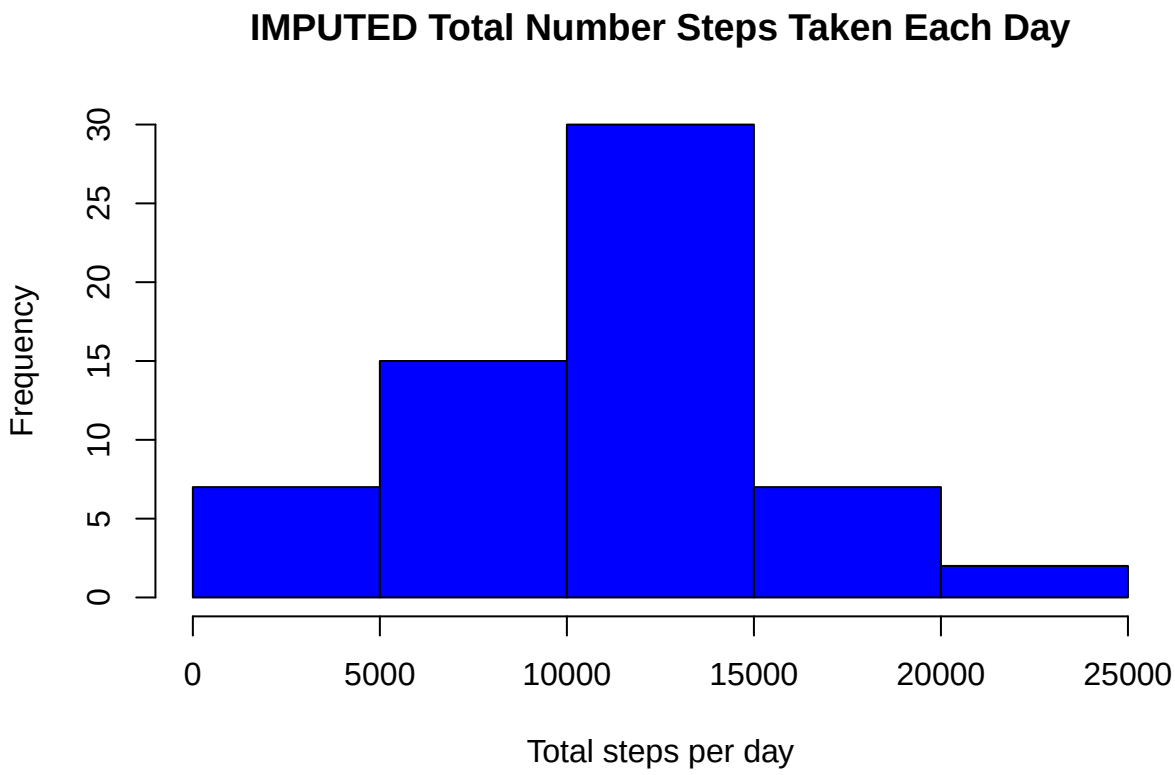
```
##      steps           interval           date
##  Min.   :  0.00   Min.   :   0.0   Min.   :2012-10-01
##  1st Qu.:  0.00   1st Qu.: 588.8   1st Qu.:2012-10-16
##  Median :  0.00   Median :1177.5   Median :2012-10-31
##  Mean   : 36.56   Mean   :1177.5   Mean   :2012-10-31
##  3rd Qu.: 29.00   3rd Qu.:1766.2   3rd Qu.:2012-11-15
##  Max.   :806.00   Max.   :2355.0   Max.   :2012-11-30
```

```
str(newActivity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
```

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?

```
newdate<-aggregate(steps~date, newActivity, sum)
hist(newdate$steps, xlab = "Total steps per day", main = "IMPUTED Total Number Steps Taken Each Day", c
```




```
newmean<-mean(newdate$steps, na.rm = TRUE)
newmean
```

```
## [1] 10528.07
```

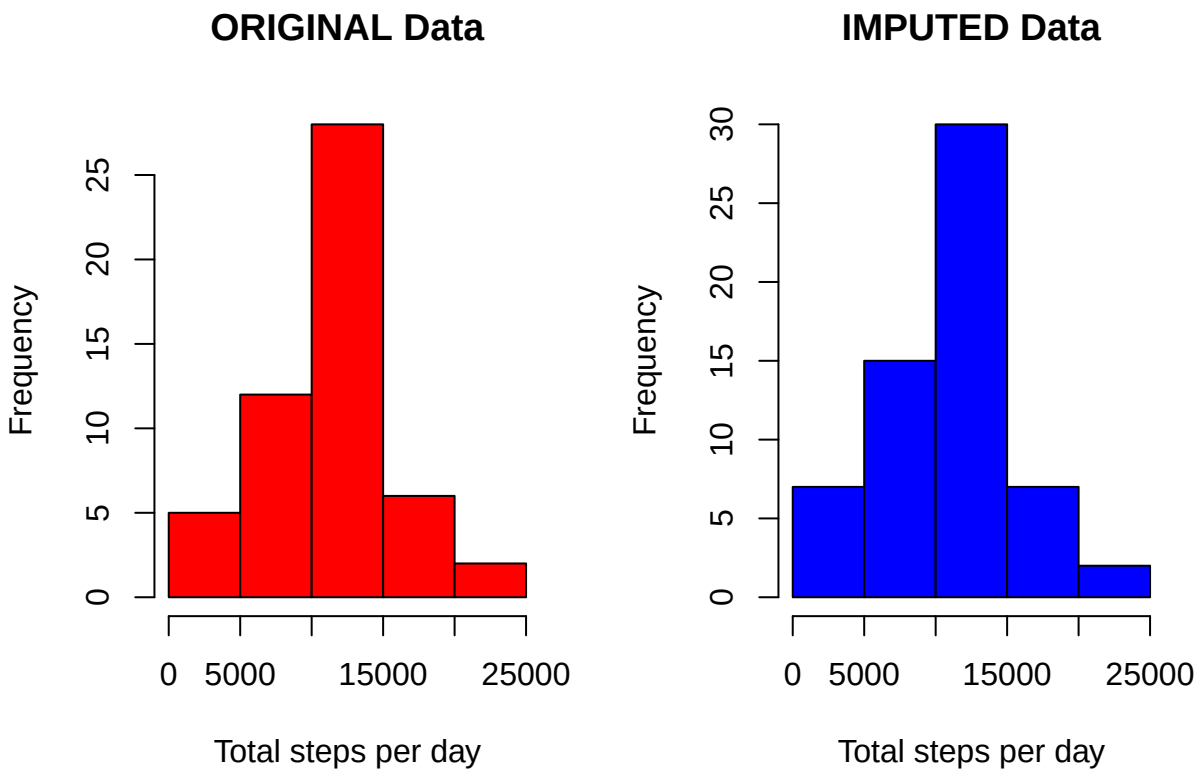
```
newmed<-median(newdate$steps, na.rm= TRUE)
newmed
```

```
## [1] 10600
```

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?

As evident, the histograms are almost identical and there is a difference of less than 240 steps in both mean and median of the original data set and the new data set with the imputed value. In my opinion, the impact of imputing missing data in this data set is minimal at most and question if the difference is reasonable enough to justify the additional time invested in imputing the missing data for this specific data set.

```
par(mfrow = c(1,2))
hist(date$steps, xlab = "Total steps per day", main = "ORIGINAL Data", col = 'red')
hist(newdate$steps, xlab = "Total steps per day", main = "IMPUTED Data", col = 'blue')
```




```
comp<-data.frame("mean"= numeric(2), "median"=numeric(2))
row.names(comp)<- c("original", "imputed")
comp[1,]<-c(origmean,origmed)
comp[2,]<-c(newmean,newmed)
comp
```

```
##              mean median
## original 10766.19  10765
## imputed  10528.07  10600
```

Are there differences in activity patterns between weekdays and weekends?

For this part the **weekdays()** function may be of some help here. Use the dataset with the filled-in missing values for this part.

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.

```
newActivity$wkday<-weekdays(newActivity$date)
newActivity$wkDEnd<-as.factor(ifelse(newActivity$wkday
%in% c("Saturday","Sunday"), "Weekend", "Weekday"))
str(newActivity)
```

```
## 'data.frame':    17568 obs. of  5 variables:
```

```
##  $ steps   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ wkday   : chr  "Monday" "Monday" "Monday" "Monday" ...
##  $ wkDEnd  : Factor w/ 2 levels "Weekday","Weekend": 1 1 1 1 1 1 1 1 1 1 ...
```

2) Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute intervals (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 the simulated data.

```
weekdays<-newActivity[newActivity$wkDEnd %in% c("Weekday"),]
weekend<-newActivity[newActivity$wkDEnd %in% c("Weekend"),]
daytime<-aggregate(steps~interval, weekdays, mean)
endtime<-aggregate(steps~interval, weekend, mean)
par(mfrow = c(2,1),oma=c(3,3,0,0), mar = c(3,3,2,2))
with(daytime, plot(steps~interval, type = "l", main = "Weekdays", cex.main= .75, xaxt = "n", yaxt = "n"
axis(3, xaxp = c(0,2400,30), labels = FALSE, cex.axis = 0.65)
axis(4, cex.axis = 0.65)
with(endtime, plot(steps~interval, type = "l", main = "Weekend",cex.main = .75, ylab = "", xaxp = c(0, 1
mtext(text = "Number of Steps", side = 2, line = 0, outer = TRUE)
mtext(text = "Interval", side = 1, line = 0, outer = TRUE)
```