Reproducible Research: Peer Assessment 1

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.

This assignment makes use of data from a personal activity monitoring device. This device collects data 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.

Loading and preprocessing the data

```r
data<-read.csv("./activity.csv")
summary(data)
```

```
##      steps            date              interval     
##  Min.   :  0.00   Length:17568       Min.   :   0.0  
##  1st Qu.:  0.00   Class :character   1st Qu.: 588.8  
##  Median :  0.00   Mode  :character   Median :1177.5  
##  Mean   : 37.38                      Mean   :1177.5  
##  3rd Qu.: 12.00                      3rd Qu.:1766.2  
##  Max.   :806.00                      Max.   :2355.0  
##  NA's   :2304
```

We observe that the variable "steps" contains 2304 NA's. We are going to generate a new data set without the rows that contains NA's.

```r
cleandata<-data[ !is.na(data$steps),]
summary(cleandata)
```

```
##      steps            date              interval     
##  Min.   :  0.00   Length:15264       Min.   :   0.0  
##  1st Qu.:  0.00   Class :character   1st Qu.: 588.8  
##  Median :  0.00   Mode  :character   Median :1177.5  
##  Mean   : 37.38                      Mean   :1177.5  
##  3rd Qu.: 12.00                      3rd Qu.:1766.2  
##  Max.   :806.00                      Max.   :2355.0
```

Observe that the variable "date" its defined as "char", so we are going to convert the class of the variable to "Date"

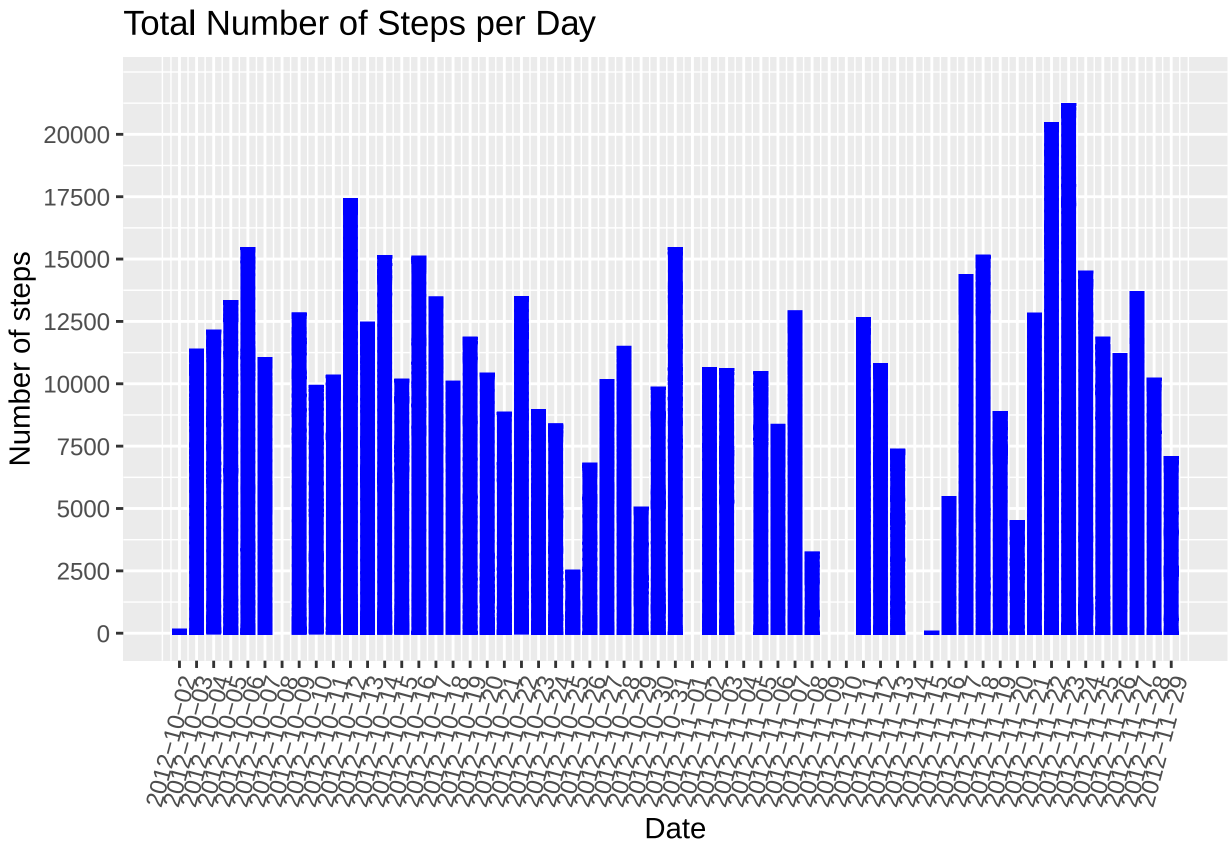
```r
cleandata$date<-as.Date(cleandata$date)
str(cleandata)
```

```
## 'data.frame':    15264 obs. of  3 variables:
##  $ steps   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ date    : Date, format: "2012-10-02" "2012-10-02" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

What is mean total number of steps taken per day?

For answering that, we are going to ignore the missing values in the data set, so we work with the clean data set.

```r
library(ggplot2)
ggplot(cleandata, aes(date, steps)) +
  geom_histogram(stat = "identity", colour = "blue", fill = "blue", width = 0.7)+
  scale_y_continuous(breaks=seq(0,20000,by=2500),limits=(c(0,22000)))+
  scale_x_date(breaks=seq(min(cleandata$date),max(cleandata$date), by="1 day"))+
  theme(axis.text.x = element_text(angle = 75, hjust = 1)) +labs(title = "Total Number of Steps per Day
```



Now we are going to calculate he mean and median total number of steps taken per day

```r
totStepsDay <- aggregate(cleandata$steps, list(Date = cleandata$date), FUN = "sum")$x
#Mean total number of steps taken per day:
mean(totStepsDay)
```

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

```
#Median total number of steps taken per day:
median(totStepsDay)
```

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

What is the average daily activity pattern?

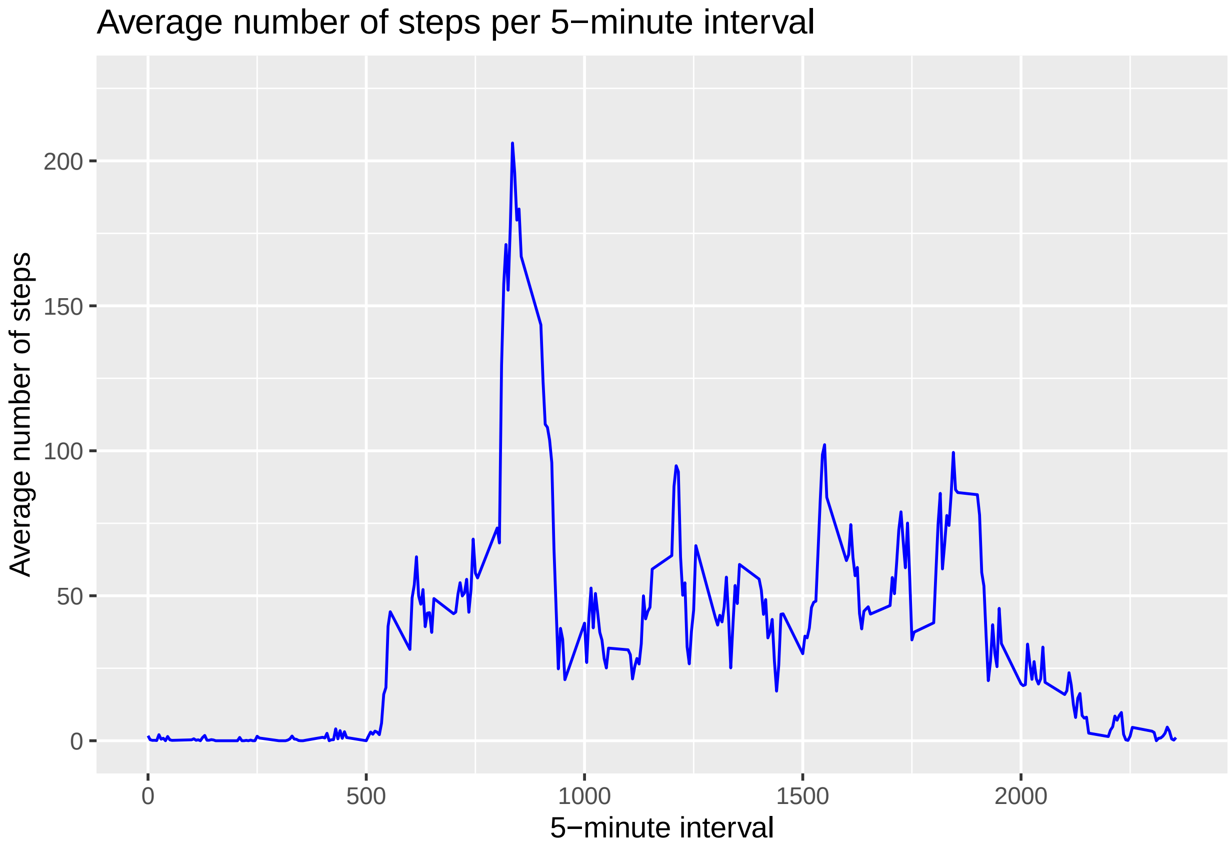
First, we are going to calculate the average steps by each 5-minute interval all across the days

```
avgSteps <- aggregate(cleandata$steps, list(interval = cleandata$interval), FUN = "mean")
names(avgSteps)[2] <- "meanSteps"
head(avgSteps)
```

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

Now, with this information, we can 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)

```
ggplot(avgSteps,aes(x=interval,y=meanSteps)) +
  geom_line(colour="blue") +
  coord_cartesian(ylim=c(0,225)) +
  labs(title = "Average number of steps per 5-minute interval", x = "5-minute interval", y = "Average nu
```



And we can calculate the interval with the maxium number of steps…

```
avgSteps[avgSteps$meanSteps == max(avgSteps$meanSteps), 1]
```

```
## [1] 835
```

Imputing missing values

We are going to calculate the total number of missing values in the dataset and stored the rows with NAs in a new data set

```
dataNA<-data[is.na(data$steps), ]
nrow(dataNA)
```

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

Now, we are going to fill the NAs with the mean steps of the corresponent 5-minute interval calculated previously

```
for (i in 1:length(dataNA$steps)){
  interval5m<-dataNA$interval[i]
  #searching mean steps of the 5-minute interval
  dataNA$steps[i]<-avgSteps[avgSteps$interval==interval5m ,2]
}
```

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

Finally, we extract the rows without NAs from the original data set, and we can join the two data sets

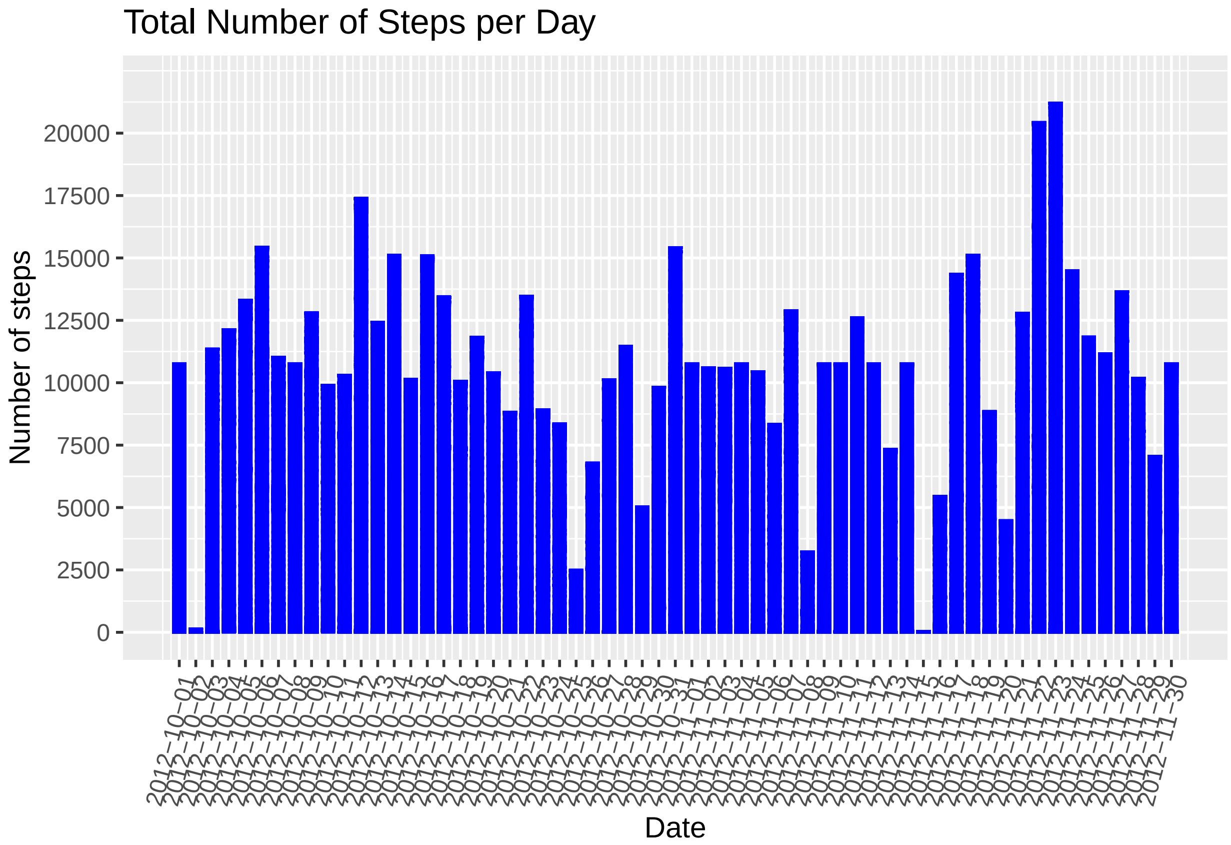
```
dataNONA<-data[!is.na(data$steps), ]
newData<-rbind(dataNA,dataNONA)
summary(newData)
```

```
##      steps            date              interval
##  Min.   :  0.00   Length:17568       Min.   :   0.0
##  1st Qu.:  0.00   Class :character   1st Qu.: 588.8
##  Median :  0.00   Mode  :character   Median :1177.5
##  Mean   : 37.38                      Mean   :1177.5
##  3rd Qu.: 27.00                      3rd Qu.:1766.2
##  Max.   :806.00                      Max.   :2355.0
```

Finally we are going to make a histogram of the total number of steps taken each day

```
newData$date<-as.Date(newData$date)
ggplot(newData, aes(date, steps)) +
  geom_histogram(stat = "identity", colour = "blue", fill = "blue", width = 0.7) +
  scale_colour_manual("",values=c("green","red")) +
  scale_y_continuous(breaks=seq(0,20000,by=2500),limits=(c(0,22000)))+
  scale_x_date(breaks=seq(min(newData$date), max(newData$date), by="1 day"))+
  theme(axis.text.x = element_text(angle = 75, hjust = 1)) +
  labs(title = "Total Number of Steps per Day", x = "Date", y = "Number of steps")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



Now we can calculate the mean and median total number of steps taken per day.

```r
totNewStepsDay <- aggregate(newData$steps, list(Date = newData$date), FUN = "sum")$x
#Mean total number of steps taken per day:
mean(totNewStepsDay)
```

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

```r
#Median total number of steps taken per day:
median(totNewStepsDay)
```

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

Do these values differ from the estimates from the first part of the assignment?

```r
#Difference of Means
mean(totStepsDay)-mean(totNewStepsDay)
```

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

```r
#Difference of Medians
median(totStepsDay)-median(totNewStepsDay)
```

```
## [1] -1.188679
```

In conclusion, there is not differences between the new values and the first ones, so there is no impact of inputting missing data on the estimates of the total daily number of steps.

Are there differences in activity patterns between weekdays and weekends?

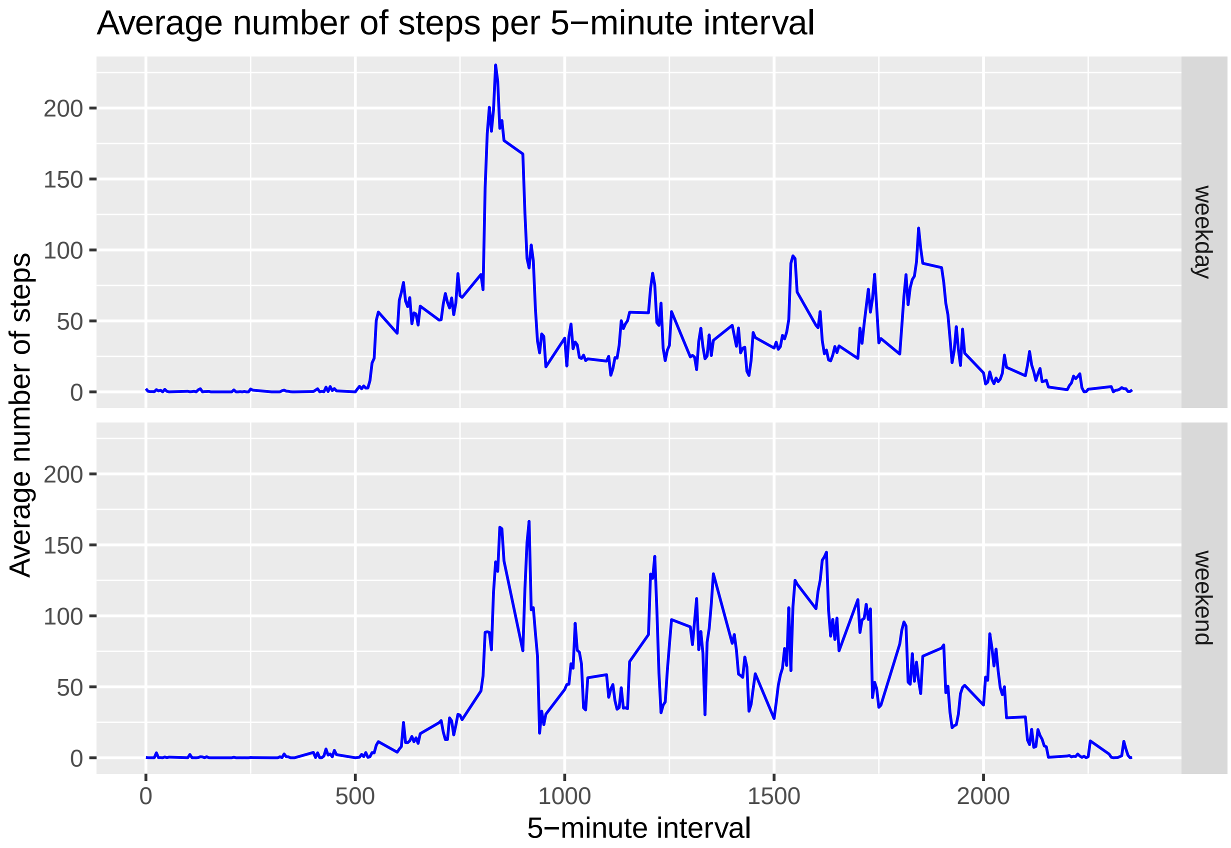
First, we are going to create a new factor variable in the data set with two levels – “weekday” and “weekend” (indicating whether a given date is a weekday or weekend day), and fill it with the correspondent value acording to the value of the variable “date”

```
newData$dayOfWeek<-factor(c("weekday","weekend"))
for (i in 1:length(newData$date)){
  day<-weekdays(newData$date[i])
  if ( day=="Saturday" | day=="Sunday") { newData$dayOfWeek[i]<-"weekend" }
  else { newData$dayOfWeek[i]<-"weekday" }
}
```

Finally, we are going to 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).

```
avgStepsWeekDays<- aggregate(newData$steps, list(interval = newData$interval, dayOfWeek=newData$dayOfWe
names(avgStepsWeekDays)[3] <- "meanSteps"

ggplot(avgStepsWeekDays,aes(x=interval,y=meanSteps)) +
  geom_line(colour="blue") +
  facet_grid(rows=vars(dayOfWeek))+
  coord_cartesian(ylim=c(0,225)) +
  labs(title = "Average number of steps per 5-minute interval", x = "5-minute interval", y = "Average nu
```


Average number of steps per 5−minute interval
Average number of steps
weekday
weekend
5−minute interval