Reproducible Research: Peer Assessment 1

Randall Helms

15 November 2016

This report uses activity monitoring data from a personal activity monitoring device, taken at five minute intervals over a two month period in autumn 2012.

Loading and preprocessing the data

Let's start by reading the data in to R and doing some basic analyses to check out the way the data is structured:

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
#read in activity data

file = "activity.csv"

activity <- read.table(file,header = TRUE,sep = ",")

#set each column up as a vector (optional but tidier)

steps <- activity$steps
date <- activity$date
interval <- activity$interval
```

Next, let's do some basic analyses of the data:

```
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
```

```
str(activity)
```

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

```
summary(activity)
```

```
##      steps                date          interval     
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0  
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8  
##  Median :  0.00   2012-10-03:  288   Median :1177.5  
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5
```

```
##  3rd Qu.: 12.00   2012-10-05:  288   3rd Qu.:1766.2
##  Max.   :806.00   2012-10-06:  288   Max.   :2355.0
##  NA's   :2304     (Other)   :15840
```

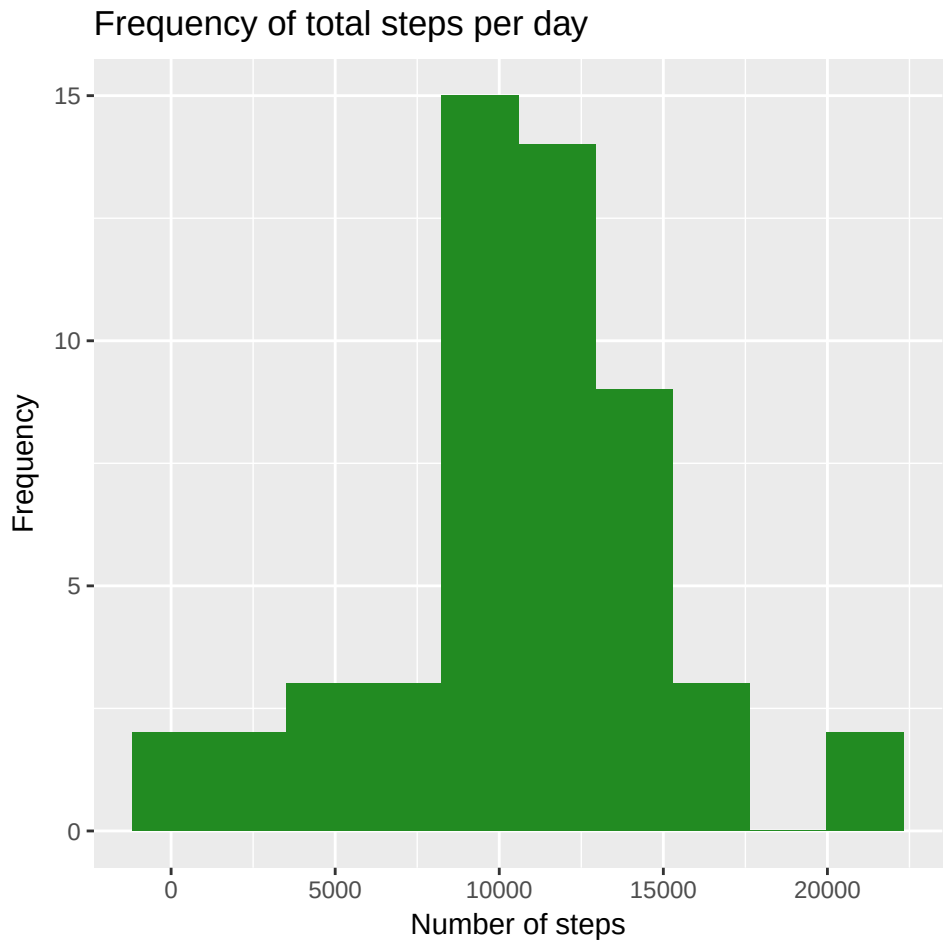
What is mean total number of steps taken per day?

Let's start this process by aggregating the number of steps by day:

```
stepsDay <- aggregate(steps ~ date, data = activity, sum, na.action = na.omit)
```

Now let's use that to create a histogram showing the distribution of steps by day:

```
ggplot(stepsDay,aes(x=steps))+
  geom_histogram(fill='forestgreen',bins=length(stepsDay$steps)/5)+
  ggtitle("Frequency of total steps per day")+
  xlab("Number of steps")+
  ylab("Frequency")
```



Next we can calculate the sum, mean, and median number of steps taken per day:

```
meanDay <- mean(stepsDay$steps)
medianDay <- median(stepsDay$steps)
```

Here are the results:

```
## [1] "The mean number of steps taken per day is: 10766.1886792453"
```

```
## [1] "The median number of steps taken per day is: 10765"
```

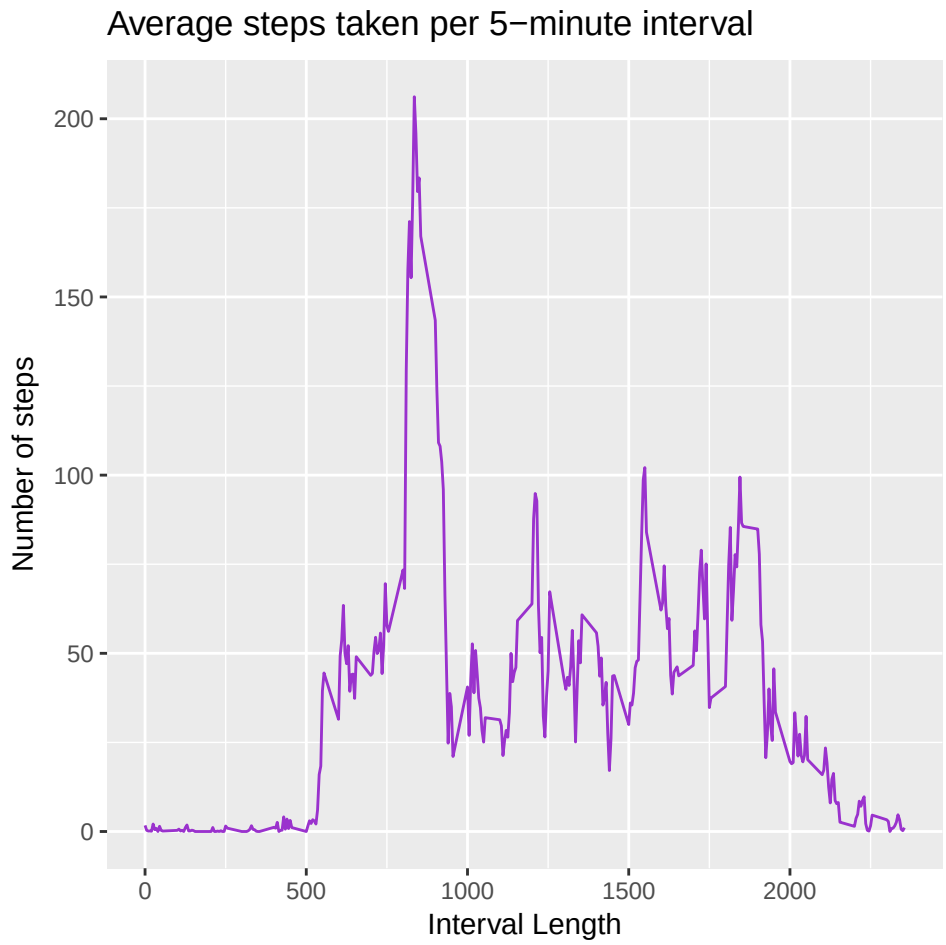
What is the average daily activity pattern?

The next step is to calculate the average daily number of steps taken by each time interval, which can be done with the aggregate function:

```
averageActivity <- aggregate(steps ~ interval,data=activity,mean,na.action = na.omit)
```

Then we can use this new vector to create a line chart:

```
ggplot(averageActivity, aes(x=interval,y=steps))+
  geom_line(color="darkorchid3")+
  ggtitle("Average steps taken per 5-minute interval")+
  xlab("Interval Length")+
  ylab("Number of steps")
```



Wow, pretty neat! But what is the interval with the highest average number of steps?

Let's quickly calculate that:

```
maxRow <- averageActivity[which.max(averageActivity$steps),] #find and print row with most steps
```

Here's the result:

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

The 835 second interval had the highest number of average steps: 206

Imputing missing values

There were a *lot* of missing values in the original data set, but how many exactly?

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

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

Let's try to rectify that by filling in the gaps in the data with an average value for the relevant interval, although first we create a copy of the data frame:

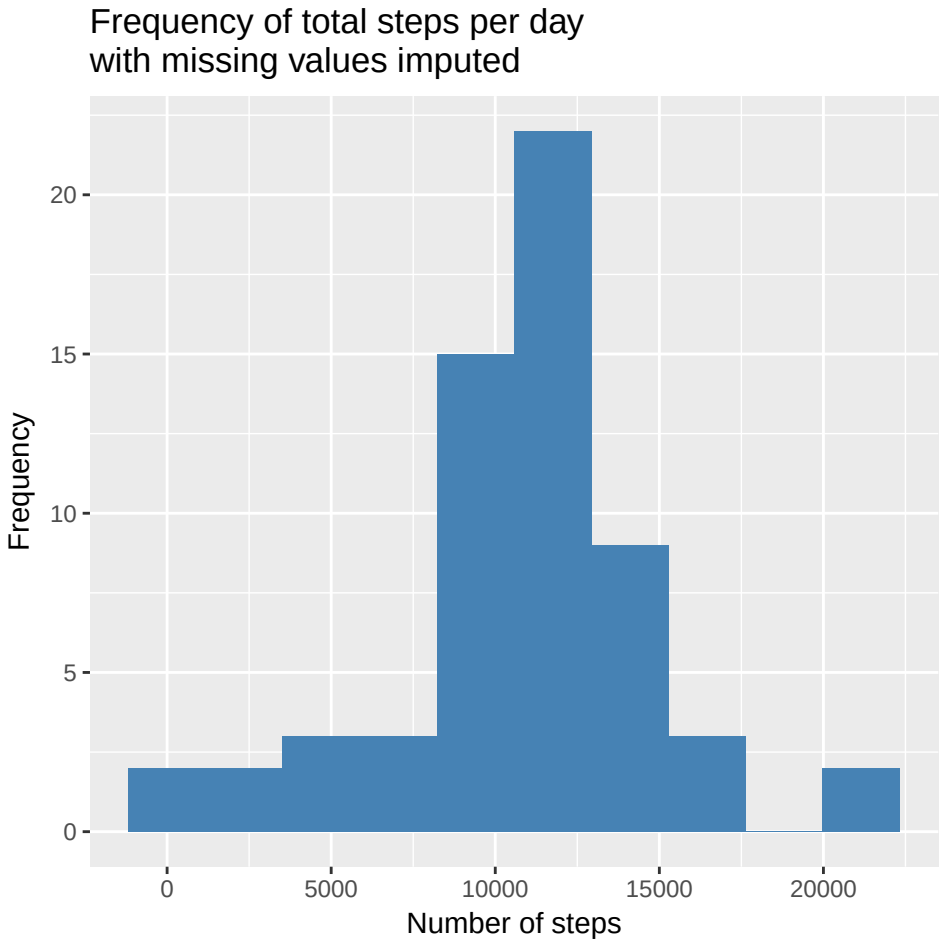
```
activityFull <- activity #create copy of the data frame

naValues <- is.na(activityFull$steps) #create logical vector for catching NA steps

intervalAvg <- tapply(activityFull$steps,activityFull$interval,mean,na.rm=TRUE,simplify = TRUE) #calcul

activityFull$steps[naValues] <- intervalAvg[as.character(activityFull$interval[naValues])] #replace na
```

Now that we have done so, let's re-calculate total steps by day:

```
stepsDayFull <- aggregate(steps ~ date,data=activityFull,sum,na.action = na.omit)
```

And then use that to re-plot the histogram of steps taken per day:

```
ggplot(stepsDayFull,aes(x=steps))+
  geom_histogram(fill='steelblue',bins=length(stepsDay$steps)/5)+
  ggtitle("Frequency of total steps per day \nwith missing values imputed")+
  xlab("Number of steps")+
  ylab("Frequency")
```



We can also recalculate the mean and median values for steps per day:

```
meanDayFull <- mean(stepsDayFull$steps)
medianDayFull <- median(stepsDayFull$steps)
```

And here are the values for them:

```
## [1] "The mean number of steps taken per day (with imputed values included) is: 10766.1886792453"
```

```
## [1] "The median number of steps taken per day (with imputed values included)  is: 10766.1886792453"
```

What are the differences between the data with and without the imputed values?

```
## [1] "The difference in mean steps taken between data with and without imputed values is 0 steps."
```

```
## [1] "The difference in median steps taken between data with and without imputed values is 1.18867924
```

Are there differences in activity patterns between weekdays and weekends?

Finally, let's check and see if there are any differences in the activity patterns between weekdays and weekends.

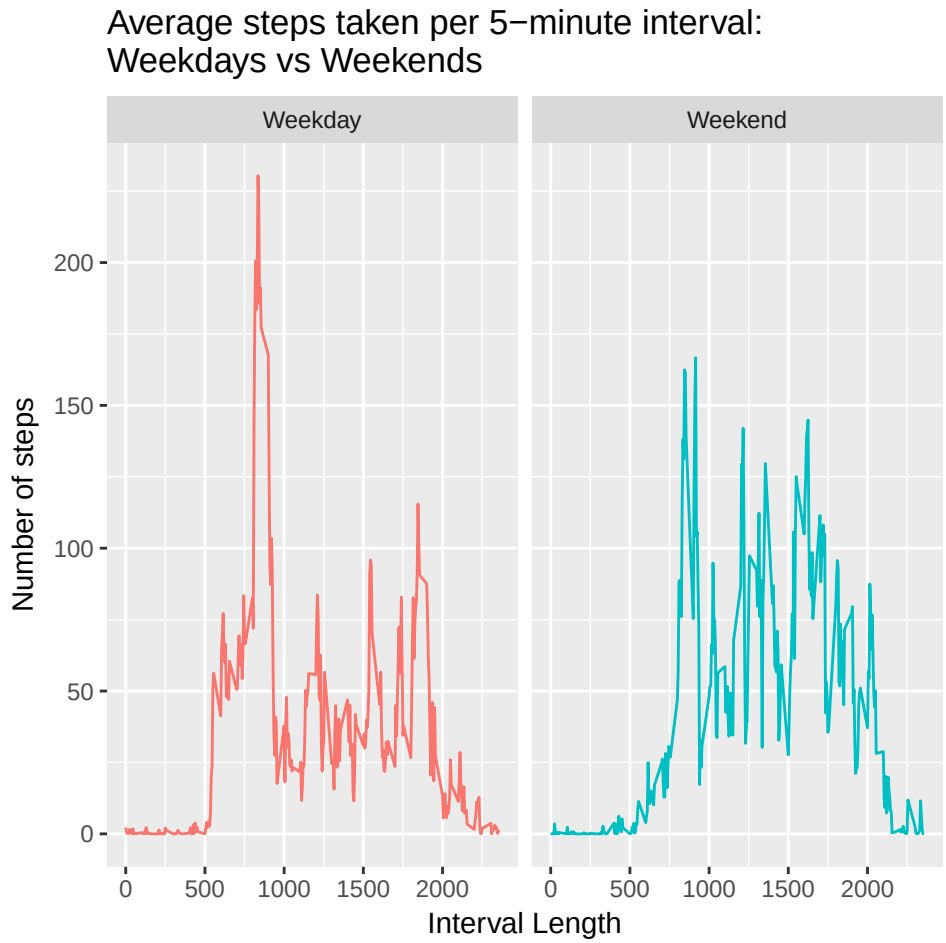
In order to do so, we need to add new columns to the `activityFull` dataframe and then re-aggregate the data for steps by interval and whether it is a weekday or weekend:

```
activityFull$dayname <- weekdays(as.Date(activityFull$date)) #add column for day of the week
```

```
activityFull$daytype <- ifelse(activityFull$dayname %in% c('Saturday','Sunday'),'Weekend','Weekday') #u

afFinal <- aggregate(steps ~ interval + daytype,data=activityFull,mean,na.action = na.omit)
```

Now that our data is ready, let's plot weekends versus weekdays on two separate line charts to make a simple visual comparison:

```
ggplot(afFinal, aes(x=interval,y=steps,color=daytype))+
  geom_line()+
  ggtitle("Average steps taken per 5-minute interval:\nWeekdays vs Weekends")+
  xlab("Interval Length")+
  ylab("Number of steps")+
  facet_grid(~daytype)+
  theme(legend.position="none")
```



And that's it! Thanks for taking the time to read through this report.