# Reproducible Research: Peer Assessment 1

Below Q&A are for Reproducible Research Peer Assessment 1. Please see README.MD file for the requirements.

## Loading and preprocessing the data

First of all as required, we can set all the code chunks and results to be visible in the global setting.

## Warning: package 'knitr' was built under R version 3.2.2

1. Load the data (i.e. read.csv())
2. Process/transform the data (if necessary) into a format suitable for your analysis

RawData <- read.csv("./activity.csv")  
RawData[,2] <- as.Date(RawData[,2])  
head(RawData)

## 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

## What is mean total number of steps taken per day?

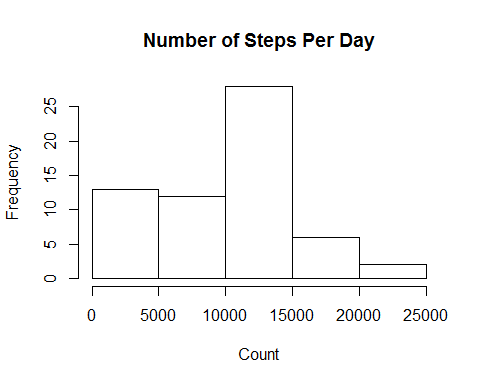
For this part of the assignment, you can ignore the missing values in the dataset.

1. Make a histogram of the total number of steps taken each day

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.2.2

stepsCount <- tapply(RawData$steps,RawData$date,sum,na.rm=TRUE)  
hist(stepsCount,xlab="Count",main="Number of Steps Per Day")



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

meanSteps <- mean(stepsCount)  
meanSteps

## [1] 9354.23

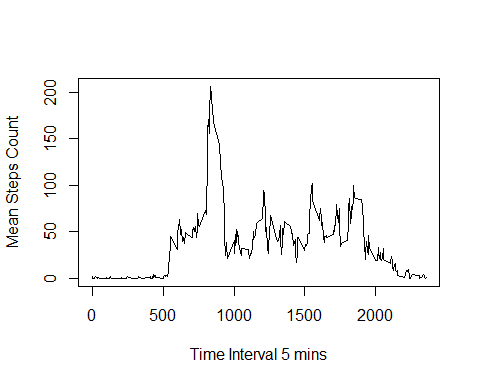
medianSteps <- median(stepsCount)  
medianSteps

## [1] 10395

## 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)

intervalCount <- tapply(RawData$steps,RawData$interval,mean,na.rm=TRUE)  
plot(names(intervalCount),intervalCount,type="l",xlab="Time Interval 5 mins",ylab="Mean Steps Count")



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

intervalMax <- names(intervalCount[match(max(intervalCount),intervalCount)])  
intervalMax

## [1] "835"

## Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as 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)

dataNA <- subset(RawData,is.na(RawData$steps)==TRUE)  
numberNA <- nrow(dataNA)  
numberNA

## [1] 2304

1. 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.

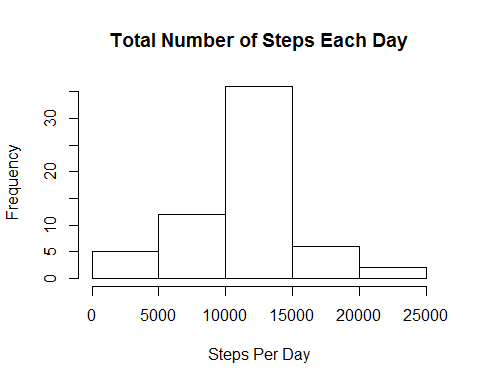
The strategy will consist in assign the NA values the mean of its interval.

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

newData <- RawData  
for (i in 1:nrow(newData)) {  
 int <- newData[i, ]  
 if (is.na(int$steps)) {  
 intv <- newData[newData$interval == int$interval, ]  
 mstp <- mean(intv$steps, na.rm = TRUE)  
 newData[i, 1] = mstp  
 }  
}

1. 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?

totalSteps <- aggregate(steps ~ date,data=newData,FUN= sum)  
colnames(totalSteps) <- c("date","steps")  
hist(totalSteps$steps,xlab="Steps Per Day",main="Total Number of Steps Each Day")



summary(totalSteps$steps,digits=12)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 41.00 9819.00 10766.19 10766.19 12811.00 21194.00

## 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.

newDataWD <- newData  
newDataWD$wd <- rep("weekday", nrow(newDataWD))  
  
for (i in 1:nrow(newDataWD)) {  
 if (weekdays(as.Date(newDataWD[i, "date"])) == "Saturday" | weekdays(as.Date(newDataWD[i,   
 "date"])) == "Sunday") {  
 newDataWD[i, 4] <- "weekend"  
 }  
}  
  
newDataWD$wd <- factor(newDataWD$wd)

1. 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). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

Firstly filter dataset into weekday and weekend

weekDay <- newDataWD[newDataWD$wd == "weekday", ]  
weekEnd <- newDataWD[newDataWD$wd == "weekend", ]

Then aggregate steps by interval

intervalWeekday <- aggregate(steps ~ interval, data = weekDay, FUN = mean)  
colnames(intervalWeekday) <- c("interval", "steps")  
  
intervalWeekend <- aggregate(steps ~ interval, data = weekEnd, FUN = mean)  
colnames(intervalWeekend) <- c("interval", "steps")

Last display the 2 plots.

par(mfrow=c(2,1))  
  
plot(intervalWeekend$interval, intervalWeekend$steps, type = "n", xlab = "interval",   
 ylab = "steps average", main = "Weekend")  
  
lines(intervalWeekend$interval, intervalWeekend$steps, type = "l")  
  
plot(intervalWeekday$interval, intervalWeekday$steps, type = "n", xlab = "interval",   
 ylab = "steps average", main = "Weekday")  
  
lines(intervalWeekday$interval, intervalWeekday$steps, type = "l")

