# Reproducible Research: Peer Assessment 1

By Chuk Yong

## Loading and preprocessing the data

Show any code that is needed to load the data. Process/transform the data (if necessary) into a format suitable for your analysis

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

setwd("~/Desktop/Rstudio/RepData\_PeerAssessment1")  
Data <- read.csv("activity.csv")  
totalSteps <-aggregate(steps~date,data=Data,sum,na.rm=TRUE)  
hist(totalSteps$steps)



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

mean(totalSteps$steps)

## [1] 10766.19

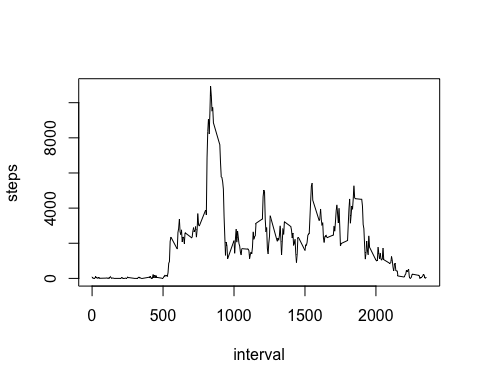
median(totalSteps$steps)

## [1] 10765

## What is the average daily activity pattern?

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)

intervalSteps <- aggregate(steps~interval, Data, sum, na.rm=TRUE)  
plot(steps~interval,data=intervalSteps,type="l")



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

intervalSteps[which.max(intervalSteps$steps),]$interval

## [1] 835

## Imputing missing values

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

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

sum(is.na(Data$steps))

## [1] 2304

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. Ans: I will use the mean for 5-min interval which should be a more accurate prediction of the missing values

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

library(dplyr)

##   
## Attaching package: 'dplyr'

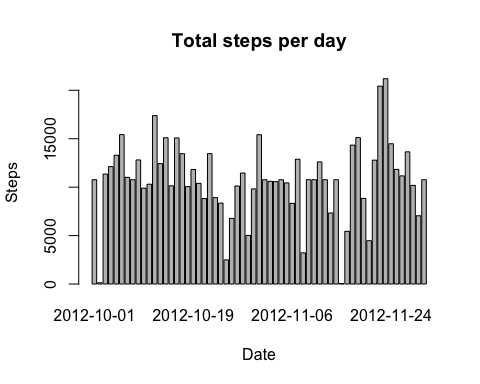
## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

NewData <- Data %>% group\_by(interval) %>%  
 mutate(steps = replace(steps, is.na(steps), mean(steps, na.rm = TRUE)))

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.

totalNewSteps <- aggregate(steps ~ date, data = NewData, sum)  
barplot(totalNewSteps$steps, names.arg = totalNewSteps$date, main= "Total steps per day",xlab = "Date", ylab = "Steps")



Do these values differ from the estimates from the first part of the assignment? Ans: The graph is quite close to the first.

What is the impact of imputing missing data on the estimates of the total daily number of steps?

mean(totalNewSteps$steps)

## [1] 10766.19

median(totalNewSteps$steps)

## [1] 10766.19

Ans: Comparing this to the first set of data, the difference is very small: less than 1.

## 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.

NewDateData <- NewData %>% mutate(DayofWeek = ifelse(weekdays(as.Date(date)) %in% c("Saturday","Sunday"), "Weekend","Weekday"))  
NewDateData$DayofWeek <- as.factor(NewDateData$DayofWeek)

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.

WeekDaySteps <- NewDateData %>% group\_by(DayofWeek, interval) %>% summarise(Totalsteps=sum(steps))  
library(ggplot2)  
ggplot(WeekDaySteps, aes(x=interval, y=Totalsteps)) +   
 geom\_line() +   
 facet\_wrap(~ DayofWeek, nrow=2, ncol=1) +  
 labs(x="Interval", y="Number of steps") +  
 theme\_bw()

