Reproducible Research: Peer Assessment 1

## Introduction

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

The raw data is contained in a zip file which is unpacked and read into a variable called **steps**. The raw data is pre-processed to allow easy analysis and address questions about the data:

* the date column is converted into a date object.
* the interval column is converted from an integer to a string representing the start of the five minute interval.

unzip("./activity.zip")  
steps<-read.csv("./activity.csv", header = TRUE)  
steps$date <- as.Date(steps$date)  
# Pad out the interval integer to 4 digits and insert a : between HH and MM  
steps$interval <- formatC(steps$interval, width=4, flag="0")  
steps$interval <- gsub('([0-9]{2})([0-9]{2})', '\\1:\\2',steps$interval)  
head(steps)

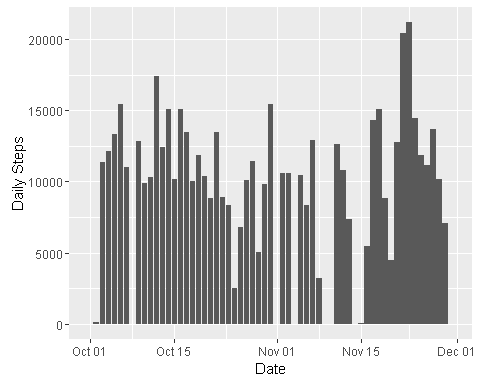
## steps date interval  
## 1 NA 2012-10-01 00:00  
## 2 NA 2012-10-01 00:05  
## 3 NA 2012-10-01 00:10  
## 4 NA 2012-10-01 00:15  
## 5 NA 2012-10-01 00:20  
## 6 NA 2012-10-01 00:25

## Number of Daily steps

The **steps** dataframe is summarised by grouping the time periods for each day and summing the total number of steps. At this stage, the data contains missing values which are ignored for now. This will be addressed later in the paper.

The resulting **daily\_steps** variable can them be plotted as a histogram which gives a sense of how the daily total varies over the two months.

daily\_steps <- steps %>% group\_by(date) %>%  
 summarise(daily\_steps=sum(steps, na.rm=TRUE))  
ggplot(daily\_steps,aes(x=date,y=daily\_steps))+  
 geom\_bar(stat="identity") +  
 ylab("Daily Steps") + xlab("Date")



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

The **daily\_steps** data frame can be used to calculate the mean number of steps per day as follows:

daily\_mean\_steps <- mean(daily\_steps$daily\_steps)  
daily\_median\_steps <- median(daily\_steps$daily\_steps)

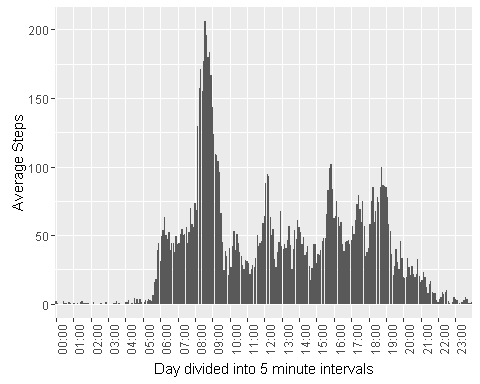
This gives a daily mean of 9354.2295082 steps taken. The daily median is 10395.

## What is the average daily activity pattern?

Each day is divided into five minute intervals, starting at midnight. We first find the mean of each five minute period across all the days in the study for which there are measurements (missing values will be ignored for now and we'll return to them later in the paper).

When graphing the average value in each of the 288 time slots (12 per hour \* 24 hours), we will only create labels on the x axis for each hour to make it easier to read.

daily\_interval <- steps %>% group\_by(interval) %>%  
 summarise(average\_steps=mean(steps, na.rm=TRUE))  
  
# keep labels only ending in 00, ie on the hour  
hour\_intervals <- ifelse(grepl('00$', daily\_interval$interval), daily\_interval$interval, FALSE)  
ggplot(daily\_interval,aes(x=interval,y=average\_steps))+  
 geom\_bar(stat="identity") +  
 ylab("Average Steps") + xlab("Day divided into 5 minute intervals") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1)) +  
 scale\_x\_discrete(breaks=hour\_intervals)



From the graph above, we can see that the average busiest period is between 8am and 9am, and a small calculation can be made to find the precise interval where the maximum occurs. The daily intervals are ordered in reverse order, with the maximum number of steps at the top. The first row of this ordering will therefore give the time interval where the maximum occurs.

interval\_max\_steps <- daily\_interval[order(daily\_interval$average\_steps, decreasing = T)[1],]

This calculation shows that 08:35 is the 5 minute time period with the most steps on average, which also tallies with a visual inspection of the graph.

## Imputing missing values

## Are there differences in activity patterns between weekdays and weekends?