Untitled

Srikanth

July 4, 2016

## Reproducible Research Assignment 1

This is Project report for **Reproducible Research course** in Coursera.

# Introduction

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 the data and the necessary pacakages for data analysis

library(knitr)  
opts\_chunk$set(echo= TRUE)  
data<-read.csv("activity.csv", header = TRUE)  
head(data)

## 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

summary(data)

## 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

data$date <- as.Date(data$date)  
library(dplyr)

##   
## Attaching package: 'dplyr'

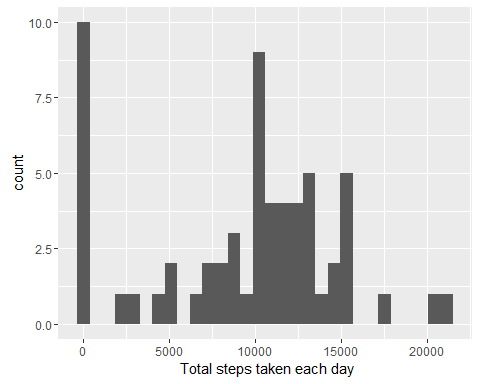
## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

# Total, mean and median on the number of steps taken per day

library(ggplot2)  
tsteps<-tapply(data$steps, data$date, FUN= sum, na.rm= TRUE)  
qplot(tsteps, xlab=" Total steps taken each day")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



mean(tsteps, na.rm=TRUE)

## [1] 9354.23

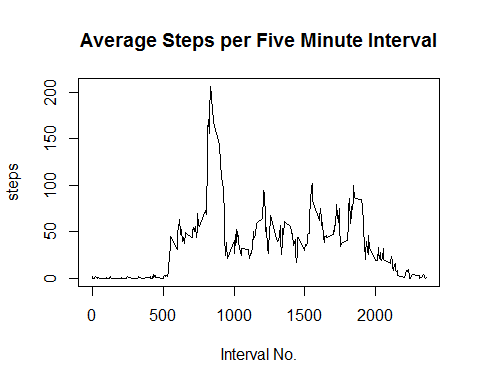
median(tsteps, na.rm=TRUE)

## [1] 10395

# Time series plot of the average number of steps taken

To find the average daily activy pattern we aggregated the data on steps by the interval and took the mean. This was then plotted into a graph where the pattern is easily discernable. The maximum value, while easy to find on the graph, is found numerically and displayed.

aggdata<- aggregate(steps~interval, data=data, mean, na.rm=TRUE)  
plot(aggdata$interval, aggdata$steps, type="l", main="Average Steps per Five Minute Interval",  
 xlab="Interval No.", ylab="steps")



maxi<-max(aggdata$steps)

# Imputing missing values

There are a lot of missing values in the data

nrow(data[is.na(data),])

## [1] 2304

one of the strategy is to fill the missing values with the average number of steps in the same 5 minute interval

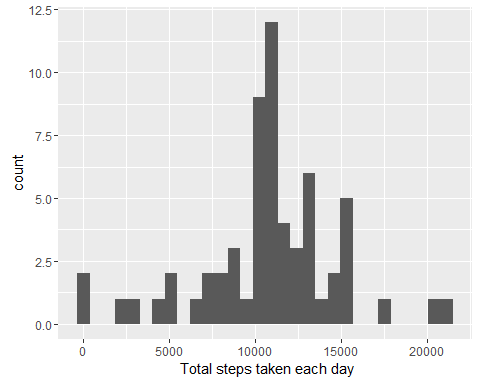
data\_full <- data  
nas <- is.na(data\_full$steps)  
avg\_interval <- tapply(data\_full$steps, data\_full$interval, mean, na.rm=TRUE, simplify=TRUE)  
data\_full$steps[nas] <- avg\_interval[as.character(data\_full$interval[nas])]  
sum(is.na(data\_full$steps))

## [1] 0

With the new data set after filling the missing values, the sum, mean and median is recalculated.

library(ggplot2)  
newtsteps<-tapply(data\_full$steps, data\_full$date, FUN= sum, na.rm=TRUE)  
qplot(newtsteps, xlab=" Total steps taken each day")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



mean(newtsteps, na.rm=TRUE)

## [1] 10766.19

median(newtsteps, na.rm=TRUE)

## [1] 10766.19

# Difference in activity patterns between weekdays and weekends

First step is to tag each day as a weekday or a weekend.

data\_full$day <- weekdays(as.Date(data\_full$date))  
data\_full$dateTime<- as.POSIXct(data\_full$date, format="%Y-%m-%d")  
data\_full$daycategory <- ifelse(data\_full$day %in% c("Saturday", "Sunday"), "Weekend", "Weekday")

Next step is to plot two charts for a weekday and a weekend.

library(plyr)

## -------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

library(lattice)  
aggdata2 <- ddply(data\_full, .(interval, daycategory), summarize, Avg = mean(steps))  
##Plot data in a panel plot  
xyplot(Avg~interval|daycategory, data=aggdata2, type="l", layout = c(1,2),  
 main="Average Steps per Interval Based on Type of Day",   
 ylab="Average Number of Steps", xlab="Interval")

