peer\_evaluation\_1

Victor

January 13, 2017

## Peer Evaluation project 1

This R markdown document contains the code and solutions for the peer evaluation project 1 of The Reproducible research course.

## 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

We are going to want to download the data from the source and save a zip file in our working directory. The code below creates the directory, downloads and reads in the data to be analyzed.

### SETTING THE WORK DIRECTORY  
# wkdir = "~\\"  
# subdir = "Github\\Reproducible-Research"  
# create directory if it doesn't exist yet  
# dir.create(file.path(wkdir, subdir))  
# setwd(file.path(wkdir, subdir))  
# getwd()  
#zip file download to director  
# url = "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"  
# download.file(url, destfile = "peer\_review\_data.zip")  
#unzip file  
# unzip("peer\_review\_data.zip")  
#read in the data  
activity = read.csv("activity.csv")  
activity$date = as.Date(activity$date)

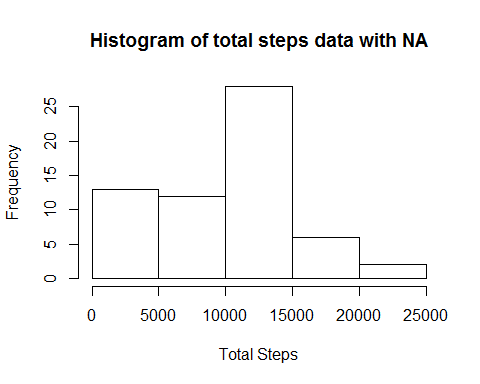
## What is mean total number of steps taken per day?

We are using the aggregate function to calculate the total number of steps per days

#q1  
#1  
steps\_tot = with(activity,aggregate(steps, by =list(date),sum,na.rm = TRUE))  
colnames(steps\_tot) = c("Date","Total.Steps")

We are generating the histogram of the total number of steps taken each day

#2  
hist(steps\_tot$Total.Steps, xlab = "Total Steps", main = "Histogram of total steps data with NA")



we see that the mean number of steps is: 9354.2295082

#3  
mean(steps\_tot$Total.Steps)

## [1] 9354.23

we see that the median number of steps is: 10395

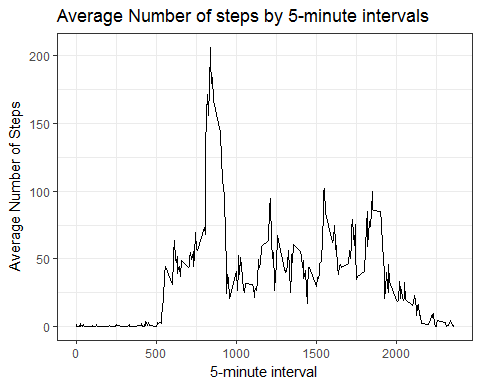
#4  
median(steps\_tot$Total.Steps)

## [1] 10395

## What is the average daily activity pattern

We are going to generate a time series plot to view the average number of steps taken for each 5-minute intervals (x-axis) averaged across all the days (y-axis). We are using the aggregate function to calculate the average number of steps by 5-minute intervals

#q2  
steps\_intervals = with(activity,aggregate(steps, by =list(interval),mean,na.rm= TRUE))  
colnames(steps\_intervals) = c("Interval","Average.Number.Steps")  
  
ggplot(steps\_intervals,aes(x=Interval,y=Average.Number.Steps))+  
 geom\_line()+  
 ggtitle("Average Number of steps by 5-minute intervals")+  
 ylab("Average Number of Steps")+  
 xlab("5-minute interval")+  
 theme\_bw()



The maximum number of steps happens on the: 1705

steps\_intervals[max(steps\_intervals$Average.Number.Steps),]

## Interval Average.Number.Steps  
## 206 1705 56.30189

## Imputing missing values

We see in the activity data there are some days/intervals that have missing values. The total number of missing values is 2304

#q1  
length(activity[is.na(activity$steps),"steps"])

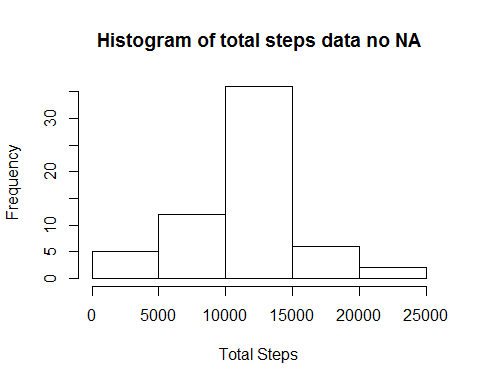
## [1] 2304

We are going to impute these missing values by using the mean for that 5-minute interval. We are doing so to reduce the bias introduces by the presence of missing values

#q3  
steps\_t = with(activity,aggregate(steps, by =list(interval),mean,na.rm= TRUE))  
colnames(steps\_t) = c("interval","steps")  
activity\_NA = activity  
  
for(i in 1:length(activity\_NA[,1])){  
 if(is.na(activity\_NA[i,"steps"])){  
 activity\_NA[i,"steps"] = steps\_t[steps\_t$interval == activity\_NA[i,"interval"],"steps"]  
 }  
}  
steps\_tot\_NA\_imputed = with(activity\_NA,aggregate(steps, by =list(date),sum))  
colnames(steps\_tot\_NA\_imputed) = c("Date","Total.Steps")

We want to look at the new data and evaluate the effect of imputing the NA values. To do so we built a histogram of the total number of steps done per day. we also calculated the mean and median of the new data set. Mean: 10766.19 and Median: 10766.19

hist(steps\_tot\_NA\_imputed$Total.Steps, xlab = "Total Steps", main = "Histogram of total steps data no NA")



mean(steps\_tot\_NA\_imputed$Total.Steps)

## [1] 10766.19

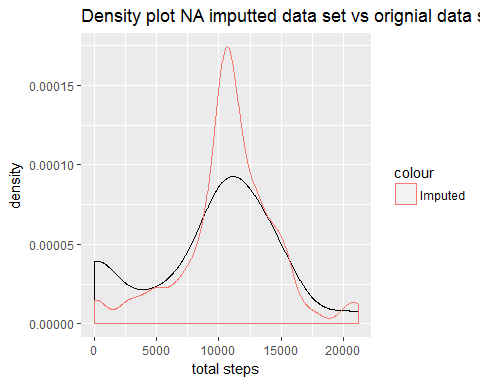
median(steps\_tot\_NA\_imputed$Total.Steps)

## [1] 10766.19

We observe that both the mean and the median change between the two data sets. The median of the original data set: 10395 and the median of the new data set: 10766.19. We see that the ,edian is increased by 371.19. The mean of the original data set: 9354.2295082 and the mean of the new data set: 10766.19. We see that the mean is increased by 1411.96.

We also observe that the histograms have a different distribution. To better visualize the difference we can look at density curves.

stepsperday = merge(steps\_tot\_NA\_imputed,steps\_tot, by = "Date")  
  
ggplot(stepsperday,aes(x=Total.Steps.y))+  
 geom\_density()+  
 geom\_density(aes(x=Total.Steps.x,color="Imputed"))+  
 ggtitle("Density plot NA imputted data set vs orignial data set")+  
 xlab("total steps")



## Difference in activity patterns between weekdays and weekends

We want to see the difference in average steps taken for weekdays versus weekends. We need to create a new column that distinguishes wether the is a week day or a weekend day. We use the function weekdays() to do so.

activity\_NA$weekday = as.factor(ifelse(weekdays(activity\_NA$date) %in% c("Saturday", "Sunday"),"weekend","weekday"))

We visualize a timeseries view of these averages over each 5 min intervals for both weekdays and weekends.

steps\_interval\_weekday = with(activity\_NA,aggregate(steps,by = list(interval,weekday),mean))  
colnames(steps\_interval\_weekday) = c("interval","weekday","steps")  
  
  
ggplot(steps\_interval\_weekday,aes(x=interval,y=steps))+  
 facet\_wrap(~weekday,nrow=2,ncol=1)+  
 geom\_line()+  
 ggtitle("Mean steps by 5min interval and weekday/weekend")+  
 ylab("steps")+  
 xlab("Interval")+  
 theme\_bw()

