# Reproducible Research: Peer Assessment 1

* load all the libraries

is.installed <- function(mypkg) is.element(mypkg, installed.packages()[,1])   
if (is.installed('dplyr') == 'FALSE') {install.packages("dplyr")} else{library(dplyr)}

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

##   
## 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

if (is.installed('ggthemes') == 'FALSE') {install.packages("ggthemes")} else{library(ggthemes)}

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

## Loading required package: ggplot2

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

if (is.installed('scales') == 'FALSE') {install.packages("scales")} else{library(scales)}  
if (is.installed('RColorBrewer') == 'FALSE')   
 {install.packages("RColorBrewer")} else{library(RColorBrewer)}  
if (is.installed('lubridate') == 'FALSE') {install.packages("lubridate")} else{library(lubridate)}  
if (is.installed('ggplot2') == 'FALSE') {install.packages("ggplot2")} else{library(ggplot2)}  
if (is.installed('plyr') == 'FALSE') {install.packages("plyr")} else{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 object is masked from 'package:lubridate':  
##   
## here  
##   
## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

if (is.installed('knitr') == 'FALSE') {install.packages("knitr")} else{library(knitr)}

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

if (is.installed('lattice') == 'FALSE') {install.packages("lattice")} else{library(lattice)}  
if (is.installed('RCurl') == 'FALSE') {install.packages("RCurl")} else{library(RCurl)}

## Loading required package: bitops

* Load the data (i.e. read.csv())
* Process/transform the data (if necessary) into a format suitable for your analysis
* Set working directory

curdir <-getwd()  
file.url<-'http://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip'  
download.file(file.url,destfile=paste(curdir,'/repdata%2Fdata%2Factivity.zip',sep=""))  
unzip(paste(curdir,'/repdata%2Fdata%2Factivity.zip',sep=""),exdir=paste(curdir,sep=""),  
 overwrite=TRUE)

* Read the CSV

data <- read.csv(paste(curdir,'/activity.csv',sep=""))

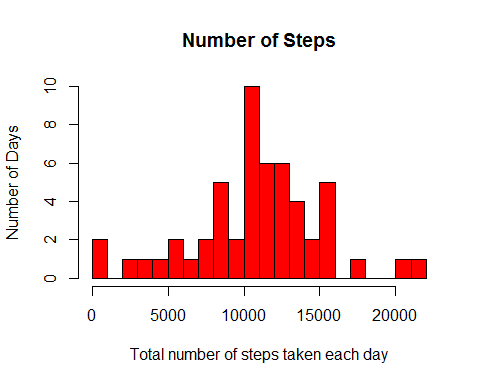
* Ignore missing value

dataClean <- subset(data, is.na(data$steps) == F)  
totalPerDay <- ddply(dataClean, .(date), summarise, steps=sum(steps))

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

* If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day
* Plot / Make a histogram of the total number of steps taken each day

hist(totalPerDay$steps , breaks = 20, main="Number of Steps",   
 xlab="Total number of steps taken each day", ylab = "Number of Days",col="red")



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

mean(totalPerDay$steps)

## [1] 10766.19

* Median

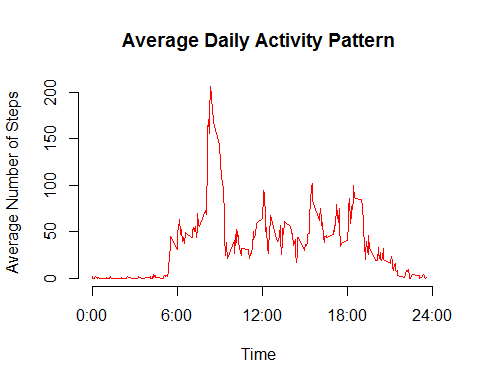
median(totalPerDay$steps)

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

averagePerInterval <- ddply(dataClean, .(interval), summarise, steps=mean(steps))  
plot(averagePerInterval$interval, averagePerInterval$steps,axes = F,   
 type="l", col="red", xlab="Time", ylab="Average Number of Steps",  
 main="Average Daily Activity Pattern")  
axis(1,at=c(0,600,1200,1800,2400), label = c("0:00","6:00","12:00","18:00","24:00"))  
axis(2)



maxSteps <- averagePerInterval[which.max(averagePerInterval$steps),] # 8.35 + 5-minute = (8.35-8.40)

## Imputing missing values

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

missingvalCount <- sum(is.na(data$steps)) # 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.
* Fill NA with the average value for each 5 minutes interval
* Create a new dataset that is equal to the original dataset but with the missing data filled in.

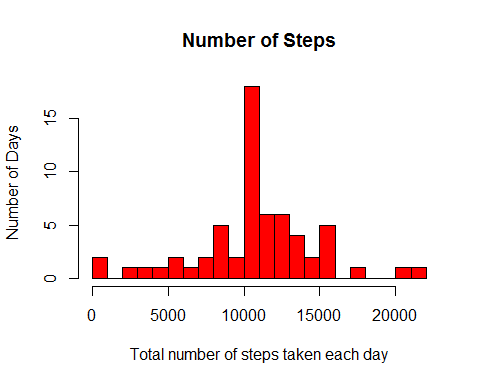
missingValFillin <- data   
for (i in 1:nrow(missingValFillin )){  
 if (is.na(missingValFillin $steps[i])){  
 missingValFillin $steps[i] <- averagePerInterval$steps[which(missingValFillin $interval[i]   
 == averagePerInterval$interval)]}  
}  
  
missingValFillin <- arrange(missingValFillin, interval) # sorting the data by interval  
missingvalCount <- sum(is.na(missingValFillin$steps)) # 0 ; test count the missing value

* 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?
* TotalStepsMissingValueFillin

totalPerDayStepsMissingvalueFillin <- ddply(missingValFillin, .(date),   
 summarise, steps=sum(steps))

* Trying plot the data to investigate

hist(totalPerDayStepsMissingvalueFillin$steps, breaks = 20, main="Number of Steps",   
 xlab="Total number of steps taken each day", ylab = "Number of Days",col="red")



mean(totalPerDayStepsMissingvalueFillin$steps) # 10766.19

## [1] 10766.19

median(totalPerDayStepsMissingvalueFillin$steps) # 10766.19

## [1] 10766.19

abs(mean(totalPerDay$steps)-  
 mean(totalPerDayStepsMissingvalueFillin$steps)) # 0

## [1] 0

abs(median(totalPerDay$steps)- median(totalPerDayStepsMissingvalueFillin$steps))/  
 median(totalPerDay$steps)

## [1] 0.0001104207

#0.0001104207

totalDifference <- sum(totalPerDayStepsMissingvalueFillin$steps)   
- sum(dataClean$steps) # 86129.51

## [1] -570608

totalDifference

## [1] 656737.5

## 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.

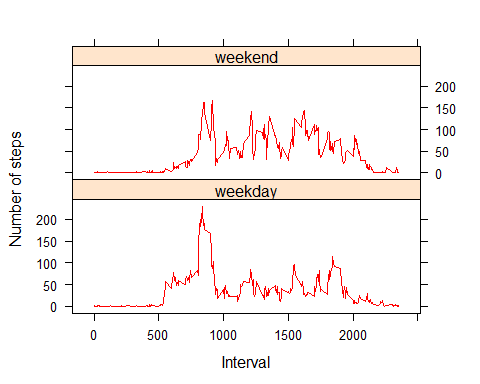
Sys.setlocale("LC\_TIME", "English")

## [1] "English\_United States.1252"

missingValFillin$weekdays <- weekdays(as.Date(missingValFillin$date))  
missingValFillin$weekdays <- ifelse(missingValFillin$weekdays  
%in% c("Saturday", "Sunday"),"weekend", "weekday")  
average <- ddply(missingValFillin, .(interval, weekdays), summarise, steps=mean(steps))

* 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.

xyplot(steps ~ interval | weekdays, data = average, layout = c(1, 2),   
 type="l", xlab = "Interval", ylab = "Number of steps" , col="red")



## End of reporting