# Reproducible Research Week 2 Assignment

Loading necessary packages and downloading and reading in the necessary data:

library(knitr)  
library(tidyr)

## Warning: package 'tidyr' was built under R version 3.5.1

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.5.1

##   
## 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

path <- "t:/pd/rscripts/activity.csv"  
#download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip", path)  
activity <- read.csv(path)

Question: What is the mean total number of steps taken per day?

First we will remove the NAs from the data set, then group by date and summarise to calculate the total / mean / median steps per day:

activityNoNA <- activity[!is.na(activity$steps),]  
  
activityNoNAsteps <- activityNoNA%>%  
 group\_by(date)%>%  
 summarise(totalsteps = sum(steps), meansteps = mean(steps), mediansteps = median(steps))

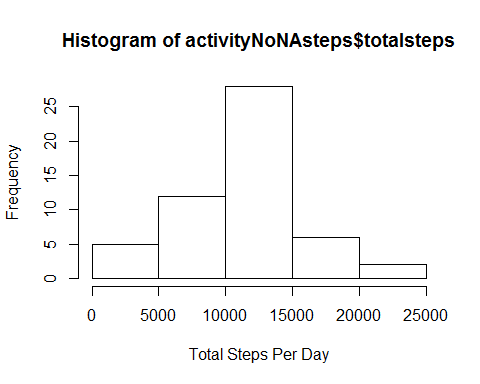
## Warning: package 'bindrcpp' was built under R version 3.5.1

activityNoNAsteps[,c("date", "totalsteps")]

## # A tibble: 53 x 2  
## date totalsteps  
## <fct> <int>  
## 1 10/10/2012 9900  
## 2 10/11/2012 10304  
## 3 10/12/2012 17382  
## 4 10/13/2012 12426  
## 5 10/14/2012 15098  
## 6 10/15/2012 10139  
## 7 10/16/2012 15084  
## 8 10/17/2012 13452  
## 9 10/18/2012 10056  
## 10 10/19/2012 11829  
## # ... with 43 more rows

Creating a histogram of the total number of steps taken per day:

hist(activityNoNAsteps$totalsteps, xlab = "Total Steps Per Day")



Calculating and reporting the mean and median of the total number of stapes taken per day:

Meanstepsperday <- mean(activityNoNAsteps$totalsteps)  
Medianstepsperday <- median(activityNoNAsteps$totalsteps)  
Meanstepsperday

## [1] 10766.19

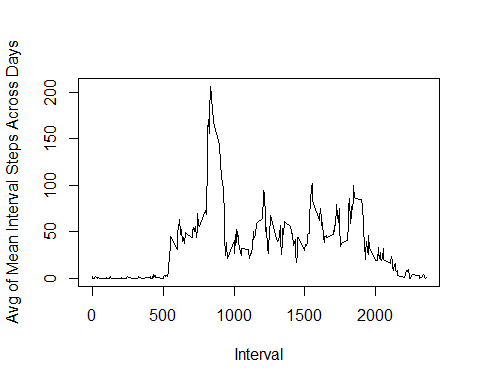
Medianstepsperday

## [1] 10765

Question: What is the average daily pattern?

First I will group by the interval and summarize by mean steps per day, then create a time series plot of the average steps per interval across days:

activityNoNAintervals <- activityNoNA%>%  
 group\_by(interval)%>%  
 summarise(meanintervalsteps = mean(steps))  
  
with(activityNoNAintervals, plot(interval, meanintervalsteps, type = "l", xlab = "Interval", ylab = "Avg of Mean Interval Steps Across Days"))



Calculating and reporting the interval which contains the max number of steps:

Maxinterval <- which.max(activityNoNAintervals$meanintervalsteps)  
activityNoNAintervals[Maxinterval, ]

## # A tibble: 1 x 2  
## interval meanintervalsteps  
## <int> <dbl>  
## 1 835 206.

Question: Imputing missing values

Subsetting by only rows with missing data then calculating and reporting the number of missing values:

activityNAsteps <- activity[is.na(activity$steps),]  
nrow(activityNAsteps)

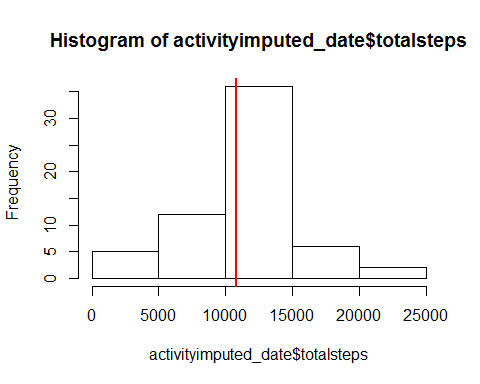
## [1] 2304

I replace the missing values with the average steps for the given interval:

activityimputed <- left\_join(activity, activityNoNAintervals, by='interval') %>%  
 mutate(steps = ifelse(is.na(steps), meanintervalsteps, steps))

Creating a histogram of the total number of steps taken each day for the data frame including imputed data:

activityimputed\_date <- activityimputed%>%  
 group\_by(date)%>%  
 summarise(totalsteps = sum(steps), meansteps = mean(steps), mediansteps = median(steps))  
  
hist(activityimputed\_date$totalsteps)  
abline(v = mean(activityimputed\_date$totalsteps), col = "red", lwd = 2)



Calculating the mean and median of total average steps using imputed data where NA:

mean(activityimputed\_date$totalsteps)

## [1] 10766.19

median(activityimputed\_date$totalsteps)

## [1] 10766.19

Then compare them to the figures computed without imputed values to see if there is any difference:

identical(mean(activityNoNAsteps$totalsteps), mean(activityimputed\_date$totalsteps))

## [1] TRUE

identical(median(activityNoNAsteps$totalsteps), median(activityimputed\_date$totalsteps))

## [1] FALSE

Mean does not change when you impute values however the median does slighty:

median(activityimputed\_date$totalsteps) - median(activityNoNAsteps$totalsteps)

## [1] 1.188679

The median for total steps is 1.18 greater for the imputed data than that with the NAs removed, a meaningless figure.

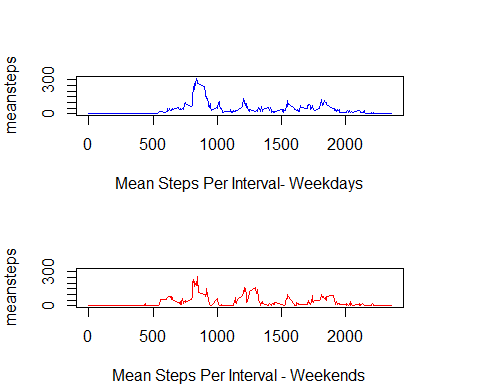
Question: Are there differences in activity patterns between weekdays and weekends?

Creating a new factor for weekends vs weekdays:

activityimputed$date <- as.Date(activityimputed$date)  
  
activityimputed <- activityimputed %>%   
 mutate(weekday = weekdays(date))  
  
weekday\_weekend = c("Monday" = "Weekday", "Tuesday" = "Weekday", "Wednesday" = "Weekday", "Thursday" = "Weekday", "Friday"="Weekday", "Saturday" = "Weekend", "Sunday"="Weekend")   
activityimputed$weekend <- as.factor(weekday\_weekend[activityimputed$weekday])  
  
activityimputed\_byintervalweekend <- activityimputed %>%   
 filter(weekend == "Weekend") %>%   
 group\_by(interval) %>%   
 summarise(totalsteps = sum(steps), meansteps = mean(steps), mediansteps = median(steps))  
  
activityimputed\_byintervalweekday <- activityimputed %>%   
 filter(weekend == "Weekday") %>%   
 group\_by(interval) %>%   
 summarise(totalsteps = sum(steps), meansteps = mean(steps), mediansteps = median(steps))

Plotting the difference in activity for weekend and weekdays:

par(mfrow = c(2, 1), mar = c(4,4,4,4))  
rng <- range(activityimputed\_byintervalweekday$meansteps)  
with(activityimputed\_byintervalweekday, plot(interval, meansteps, type = "l", col = "blue", xlab = "Mean Steps Per Interval- Weekdays", ylim = rng))  
with(activityimputed\_byintervalweekend, plot(interval, meansteps, type = "l", col = "red", xlab = "Mean Steps Per Interval - Weekends", ylim = rng))



and together on a single plot:

par(mfrow = c(1, 1))  
with(activityimputed\_byintervalweekday, plot(interval, meansteps, type = "l", col = "blue", xlab = "Mean Steps Per Interval", ylim = rng))  
with(activityimputed\_byintervalweekend, points(interval, meansteps, type = "l", col = "red", ylim = rng))  
legend("topright", pch = 19, col = c("blue", "red"), legend = c("Weekdays", "Weekends"))

