Reproducible Research: Peer Assessment 1

## Loading and preprocessing the data

Attach packages and set working directory

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

library(lubridate)

## Warning: package 'lubridate' was built under R version 3.1.2

library(knitr)

## Warning: package 'knitr' was built under R version 3.1.2

library(reshape2)  
library(ggplot2)  
library(lattice)

## Warning: package 'lattice' was built under R version 3.1.2

setwd('~/Cousera/Reproducible Research/RepData\_PeerAssessment1')

Unzip and read the data

unzip("activity.zip")  
data <- read.csv("activity.csv", stringsAsFactors=FALSE, na.strings="NA")

Calculate summary statistics (the total, mean and median number of steps per day) using the summarise function and then melt into a long data frame.

stats <- melt(as.data.frame(summarise(group\_by(data, date),  
 Total=sum(steps, na.rm=TRUE),   
 Mean=mean(steps, na.rm=TRUE),   
 Median=median(steps, na.rm=TRUE))),  
 measure.vars=c("Total", "Mean","Median"),  
 variable.name="Statistic",  
 value.name="steps")

Also, create day, month and year variables to be used in the histogram and graphic for the report. Finally, create a datetime variable formated as POSIXct for the time series plot.

stats$day <- day(as.POSIXlt(stats$date))  
stats$month <- month(as.POSIXlt(stats$date), label=TRUE, abbr=FALSE)  
stats$year <- year(as.POSIXlt(stats$date))   
  
data$datetime <- as.POSIXct(data$date, format="%Y-%m-%d") + minutes(data$interval)

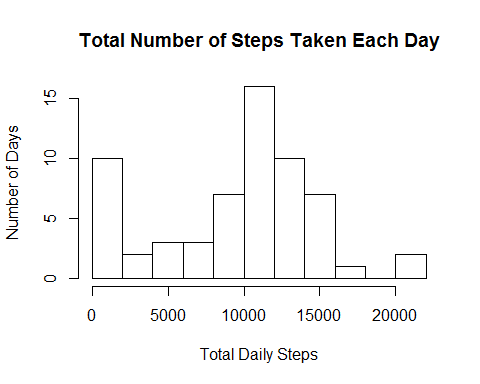
## What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset.

1. Make a histogram of the total number of steps taken each day

Keep the total rows from the stats data frame

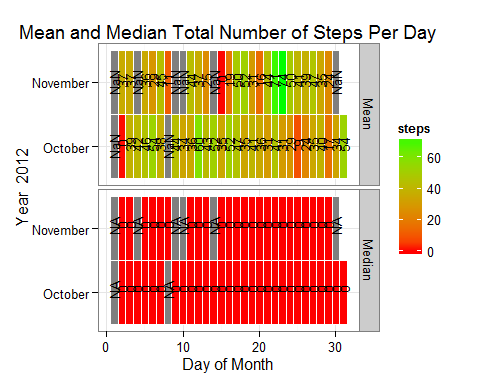
hist <- hist(stats$steps[stats$Statistic=='Total'], breaks=12,  
 xlab="Total Daily Steps",  
 ylab="Number of Days",  
 main="Total Number of Steps Taken Each Day")



1. Calculate and report the mean and median total number of steps taken per day

Keep the mean and median rows from the stats data frame and then plot the data by day and month

plot <- filter(stats, Statistic=="Mean" | Statistic=="Median")  
  
ggplot(plot, aes(day, month, fill=steps)) +   
 geom\_tile(colour = "white") +   
 scale\_fill\_gradient(low="red", high="green") +  
 geom\_text(aes(day, month, label=format(steps, digits=0), size=0.5, angle=90)) +  
 guides(size=FALSE) +  
 facet\_grid(Statistic ~ .) +   
 labs(x="Day of Month", y=paste("Year ", stats$year),   
 title="Mean and Median Total Number of Steps Per Day") +  
 theme\_bw()

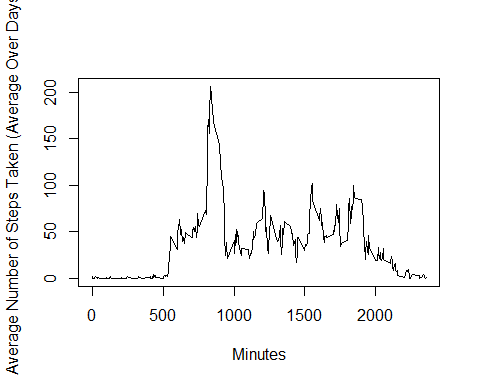


## What is the average daily activity pattern?

1. 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)

Calculate the average number of steps taken, averaged across all days and store in data frame.

time <- as.data.frame(summarise(group\_by(data, interval), Mean=mean(steps, na.rm=TRUE)))  
  
plot(time$interval, time$Mean, type="l", xlab="Minutes",  
 ylab="Average Number of Steps Taken (Average Over Days)")



1. Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

max <- filter(time, Mean==max(time$Mean))

*The* ***835*** *minute interval has an average of* ***206*** *steps, the maximum average number of steps.*

## Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)
2. 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.
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.
4. 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?

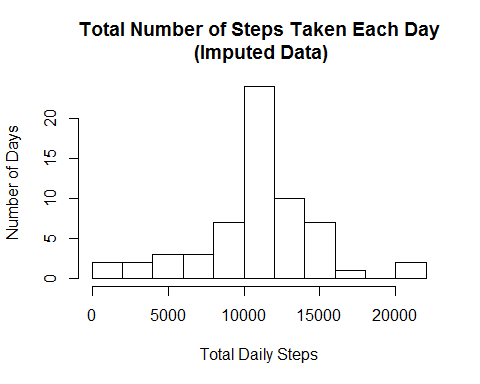
Count Missing values in data

miss <- as.numeric(sum(is.na(data$steps)))

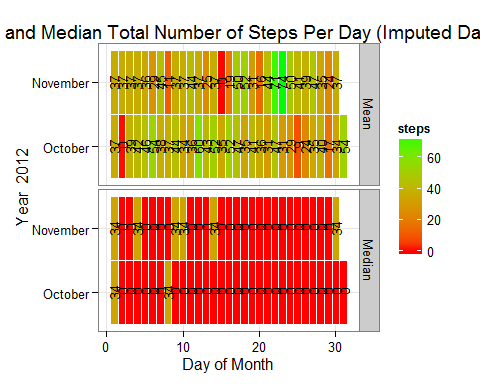
*There are a total of* ***2304*** *missing values in the dataset.*

The 5 minute interval mean will be used to impute missing values in the original dataset. Merge the means by time data with the original data. If the original steps data is missing then replace it with the interval mean. Rerun the above code using the Imputed dataset to answer part 4.

dataimp <- merge(data, time, by="interval", all.x=TRUE)  
dataimp$steps <- ifelse(is.na(dataimp$steps), dataimp$Mean, dataimp$steps)  
  
statsimp <- melt(as.data.frame(summarise(group\_by(dataimp, date),  
 Total=sum(steps),   
 Mean=mean(steps),   
 Median=median(steps))),  
 measure.vars=c("Total", "Mean","Median"),  
 variable.name="Statistic",  
 value.name="steps")  
  
statsimp$day <- day(as.POSIXlt(statsimp$date))  
statsimp$month <- month(as.POSIXlt(statsimp$date), label=TRUE, abbr=FALSE)  
statsimp$year <- year(as.POSIXlt(statsimp$date))   
  
histimp <- hist(statsimp$steps[statsimp$Statistic=='Total'], breaks=12,  
 xlab="Total Daily Steps",  
 ylab="Number of Days",  
 main="Total Number of Steps Taken Each Day\n (Imputed Data)")

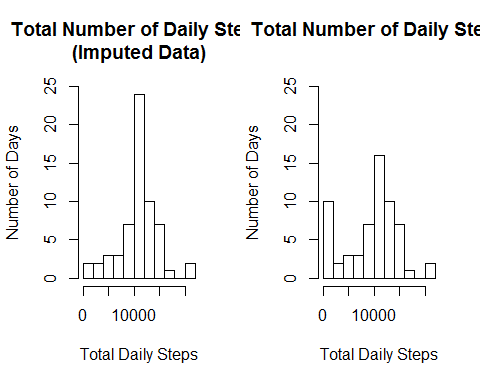


plotimp <- filter(statsimp, Statistic=="Mean" | Statistic=="Median")  
  
ggplot(plotimp, aes(day, month, fill=steps)) +   
 geom\_tile(colour = "white") +   
 scale\_fill\_gradient(low="red", high="green") +  
 geom\_text(aes(day, month, label=format(steps, digits=0), size=0.5, angle=90)) +  
 guides(size=FALSE) +  
 facet\_grid(Statistic ~ .) +   
 labs(x="Day of Month", y=paste("Year ", statsimp$year),   
 title="Mean and Median Total Number of Steps Per Day (Imputed Data)") +  
 theme\_bw()



*The 8 days with NA as the median and mean number of steps now have a median number of steps equal to 34 and a mean number of steps equal to 37. The mean and median number of steps for all of the other days remains the same. Note,the distribution of the total number of steps is no longer bimodal.*

par(mfrow=c(1,2))  
plot(histimp, xlab="Total Daily Steps",   
 ylab="Number of Days", ylim=c(0,25),  
 main="Total Number of Daily Steps\n(Imputed Data)")  
plot(hist, xlab="Total Daily Steps",  
 ylab="Number of Days", ylim=c(0,25),  
 main="Total Number of Daily Steps\n")



par(mfrow=c(1,1))

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

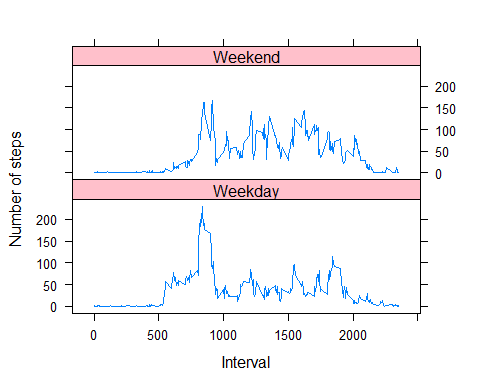
For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

1. 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.

dataimp$wday <- wday(dataimp$date, label=TRUE, abbr=FALSE)  
dataimp$weekend <- as.factor(ifelse(dataimp$wday=="Saturday" | dataimp$wday=="Sunday",  
 "Weekend", "Weekday"))

1. 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).

time2 <- as.data.frame(summarise(group\_by(dataimp, interval, weekend),  
 Mean=mean(steps)))  
  
xyplot(Mean ~ interval | weekend, data=time2,   
 layout=c(1,2), type="l",  
 strip=strip.custom(bg="pink"),  
 xlab="Interval", ylab="Number of steps")



*In general, there is more activity during the middle of the day on the weekends compared to the weekdays.*