JonPerezCP1

library(knitr)

## Warning: package 'knitr' was built under R version 3.2.3

library(lattice)  
data=read.csv("~/Documents/Coursera/Reproducible Research/Course Project 1/activity.csv")

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

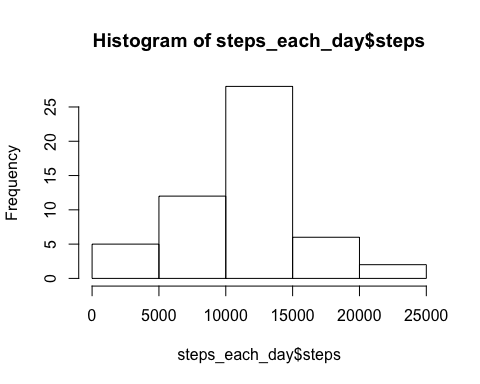
## 1.Calculate the total number of steps taken per day

steps\_day<-sum(data$steps,na.rm=TRUE)  
steps\_day

## [1] 570608

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

steps\_each\_day<-aggregate(steps~date, data=data, FUN=sum, na.rm=TRUE)  
hist(steps\_each\_day$steps)



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

steps\_each\_day\_mean <- mean(steps\_each\_day$steps)  
steps\_each\_day\_median <- median(steps\_each\_day$steps)  
steps\_each\_day\_mean

## [1] 10766.19

steps\_each\_day\_median

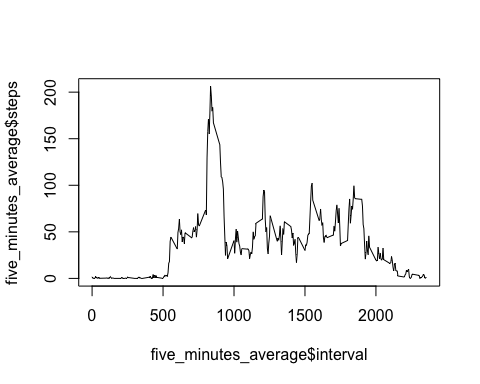
## [1] 10765

# 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)

five\_minutes\_average <- aggregate(steps~interval, data=data, FUN=mean, na.rm=TRUE)  
plot(x = five\_minutes\_average$interval, y = five\_minutes\_average$steps, type = "l")



## 2. Which 5-minute interval, on average across all the days in the dataset, contains the

## maximum number of steps?

max\_steps <- max(five\_minutes\_average$steps)  
 for (i in 1:288)   
 {  
 if (five\_minutes\_average$steps[i] == max\_steps)  
 five\_minute\_interval\_at\_max\_steps <- five\_minutes\_average$interval[i]  
 }  
five\_minute\_interval\_at\_max\_steps

## [1] 835

# Imputing missing values

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

total\_na <- 0  
for (i in 1:17568)  
 {  
 if(is.na(data$steps[i]))   
 total\_na <- total\_na+1   
 }  
total\_na

## [1] 2304

## 2. Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated.

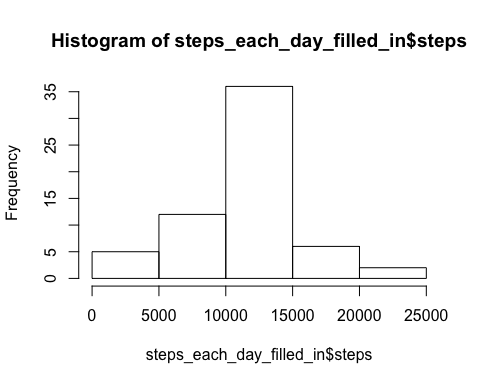
## For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.

## 3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

activity\_filled\_in <- data  
for (i in 1:17568) # loop to find the na  
{  
 if(is.na(activity\_filled\_in$steps[i])) # if steps is na store the pointer   
 {   
 five\_minute\_pointer <- activity\_filled\_in$interval[i] #store the value of pointer to find the mean on five minute interval  
 for (j in 1:288) # loop to find the value of pointer on the data frame of five minute interval  
 {  
 if (five\_minutes\_average$interval[j] == five\_minute\_pointer) # finding the value of mean of five minute interval data frame  
 activity\_filled\_in$steps[i] <- five\_minutes\_average$steps[j] # replacing the na by the mean in that fime minute interval   
   
 }  
 }  
}

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

steps\_each\_day\_filled\_in <- aggregate(steps~date, data=activity\_filled\_in, FUN=sum, na.rm=TRUE)  
hist(steps\_each\_day\_filled\_in$steps)



## 4. Calculate the mean and median and explain the imoact of imputing missing data on the estimates

## of the total daily number of steps

steps\_each\_day\_mean\_filled\_in <- mean(steps\_each\_day\_filled\_in$steps)  
steps\_each\_day\_median\_filled\_in <- median(steps\_each\_day\_filled\_in$steps)

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

## 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.

library(lubridate)

## Warning: package 'lubridate' was built under R version 3.2.3

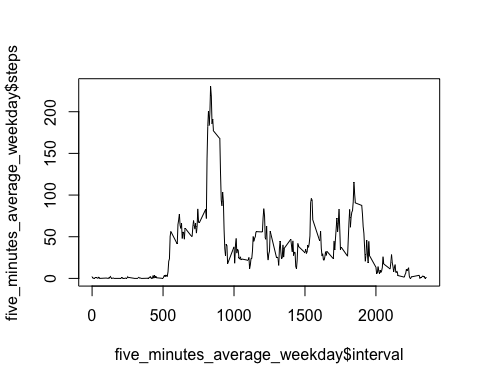
week <- wday(activity\_filled\_in$date)  
week\_day <- week  
for (i in 1:17568) # loop to find the na  
{  
 if(week[i] == 1)  
 week\_day[i] <- 'weekend'  
 if(week[i] == 2)  
 week\_day[i] <- 'weekday'  
 if(week[i] == 3)  
 week\_day[i] <- 'weekday'  
 if(week[i] == 4)  
 week\_day[i] <- 'weekday'  
 if(week[i] == 5)  
 week\_day[i] <- 'weekday'  
 if(week[i] == 6)  
 week\_day[i] <- 'weekday'  
 if(week[i] == 7)  
 week\_day[i] <- 'weekend'  
}

### Creating a new factor variable in the dataset "activity\_filled\_in"

activity\_filled\_in$weekday <-week\_day

## 2. 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).

weekday <- grep("weekday",activity\_filled\_in$weekday)  
weekday\_frame <- activity\_filled\_in[weekday,]  
weekend\_frame <- activity\_filled\_in[-weekday,]  
  
five\_minutes\_average\_weekday <- aggregate(steps~interval, data=weekday\_frame, FUN=mean, na.rm=TRUE)  
five\_minutes\_average\_weekend <- aggregate(steps~interval, data=weekend\_frame, FUN=mean, na.rm=TRUE)  
  
plot(x = five\_minutes\_average\_weekday$interval, y = five\_minutes\_average\_weekday$steps, type = "l")



plot(x = five\_minutes\_average\_weekend$interval, y = five\_minutes\_average\_weekend$steps, type = "l")

