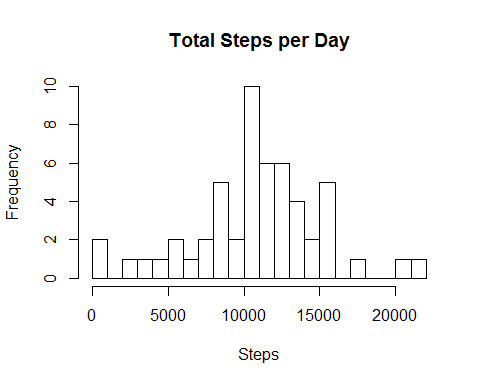
Loading and preprocessing the data The data is in an unzipped file named "activity.csv". If it has not already been loaded, it is loaded into memory.

if (!exists("rawdata")){  
 rawdata <- read.csv("activity.csv")  
}

The data is then aggregated by date and a histogram produced displaying the data. This histogram is devided into 20 buckets, which was arbitrarily chosen.

data <- aggregate(steps ~ date, data=rawdata, sum, na.rm = TRUE)  
hist(data$steps, breaks=20, main="Total Steps per Day", xlab="Steps", ylab="Frequency")

 By looking at the histogram, we can make at fair estimate of what the median number of steps in a day will be. However, while looking at a histogram is a good way to get a quick feel for a data set, estimating visually is not always accurate enough. mean total number of steps taken per day Calculating the mean and median of the aggregated steps per day is straightforward.

rawsteps\_mean <- mean(rawdata$steps, na.rm=TRUE)  
rawsteps\_median <- median(rawdata$steps, na.rm=TRUE)  
print(paste("The mean steps per day is: ", rawsteps\_mean))

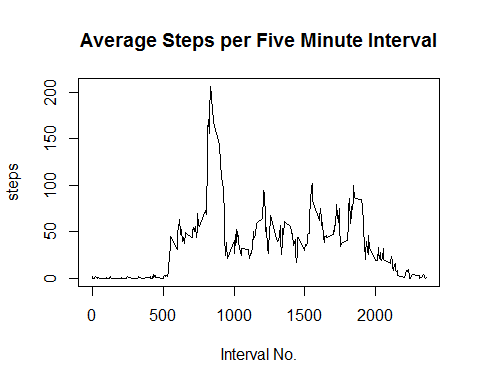
## [1] "The mean steps per day is: 37.3825995807128"

print(paste("The median steps per day is: ", rawsteps\_median))

## [1] "The median steps per day is: 0"

the average daily activity pattern To find the average daily activy pattern we aggregated the data on steps by the interval and took the mean. This was then plotted into a graph where the pattern is easily discernable. The maximum value, while easy to find on the graph, is found numerically and displayed.

stepsdata <- aggregate(steps ~ interval, data=rawdata, mean, na.rm=TRUE)  
plot(stepsdata$interval, stepsdata$steps, type="l", main="Average Steps per Five Minute Interval",  
 xlab="Interval No.", ylab="steps")



maxsteps <- max(stepsdata$steps)  
print(paste("The maximum number of steps in a five minute interval was: ", maxsteps))

## [1] "The maximum number of steps in a five minute interval was: 206.169811320755"

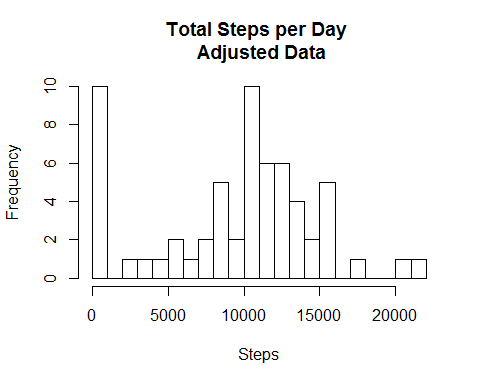
Imputing missing values There are numerous NAs in the initial data set.

missingdata <- sum(is.na(rawdata$steps))  
print(paste("There are", missingdata, "missing data points."))

## [1] "There are 2304 missing data points."

I chose to replace any NA value with the median value in the data set. This is not a terribly sophisticated method of replacing NA values but is sufficient to this task. The mean and median of this corrected data set are then calculated and displayed.

betterdata <- rawdata  
betterdata$steps[is.na(betterdata$steps)] <- median(rawdata$steps, na.rm=TRUE)  
betterdataday <- aggregate(steps ~ date, data=betterdata, sum, na.rm=TRUE)  
hist(betterdataday$steps, breaks=20, main="Total Steps per Day \n Adjusted Data",  
 xlab="Steps", ylab="Frequency")



bsteps\_mean <- mean(betterdata$steps)  
bsteps\_median <- median(betterdata$steps)  
print(paste("The mean is: ", bsteps\_mean))

## [1] "The mean is: 32.4799635701275"

print(paste("The median is: ", bsteps\_median))

## [1] "The median is: 0"

differences in activity patterns between weekdays and weekends In order to quickly determine if there are activity differences between weekends and weekdays we simply plot two graphs, one with weekday data and one with weekend data. By plotting one above the other our x-axes are identical enabling us to "eyeball" (a time-honored scientific method) the graphs to note any differences.

betterdata$date <- as.Date(betterdata$date)  
betterdata$dayname <- weekdays(betterdata$date)  
betterdata$weekend <- as.factor(ifelse(betterdata$dayname == "Saturday" |  
 betterdata$dayname == "Sunday", "weekend", "weekday"))  
library(lattice)

## Warning: package 'lattice' was built under R version 3.2.5

plotdata <- aggregate(steps ~ interval + weekend, betterdata, mean)  
xyplot(steps ~ interval | factor(weekend), data=plotdata, aspect=1/3, type="l")

