# Reproducible Research Assignment 1

## Loading and Processing the data

if (!file.exists('activity.csv')) {  
 unzip('activity.zip',overwrite=TRUE)  
}  
activity <- read.csv("activity.csv", header = T, sep = ",")

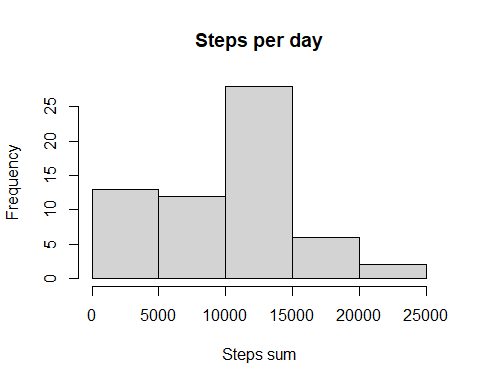
echo=TRUE

## what is the mean total number of steps taken per day?

su <- tapply(activity$steps, activity$date, sum, na.rm=T)  
echo=TRUE

# Histogram of total number of steps taken each day

hist(su, xlab = "Steps sum", main = "Steps per day")



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

mean\_su <- round(mean(su))  
median\_su <- round(median(su))  
print(c("Mean",mean\_su))

## [1] "Mean" "9354"

print(c("Median",median\_su))

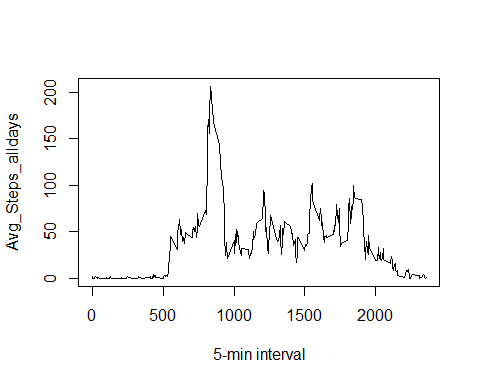
## [1] "Median" "10395"

echo=TRUE

## What is the average daily activity pattern?

Here we generate a time series plot: the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

Avg\_Steps\_alldays <- tapply(activity$steps, activity$interval, mean, na.rm=T)  
plot(Avg\_Steps\_alldays ~ unique(activity$interval), type="l", xlab = "5-min interval")



5-minute interval, on average across all the days in the dataset, which contains the maximum number of steps

Avg\_Steps\_alldays[which.max(Avg\_Steps\_alldays)]

## 835   
## 206.1698

echo=TRUE

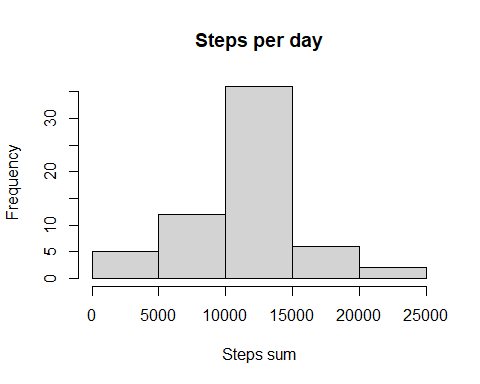
## Inputting missing values

New dataset that is equal to the original dataset but with the missing data filled

activity2 <- activity  
for (i in 1:nrow(activity)){  
 if(is.na(activity$steps[i])){  
 activity2$steps[i]<- Avg\_Steps\_alldays[[as.character(activity[i, "interval"])]]  
 echo=TRUE  
 }  
}

Histogram of the total number of steps taken each day

su2 <- tapply(activity2$steps, activity2$date, sum, na.rm=T)  
hist(su2, xlab = "Steps sum", main = "Steps per day")



echo=TRUE

New mean and median total number of steps taken per day

mean\_su2 <- round(mean(su2))  
median\_su2 <- round(median(su2))  
print(c("New Mean",mean\_su2))

## [1] "New Mean" "10766"

print(c("New Median",median\_su2))

## [1] "New Median" "10766"

echo=TRUE

Impact of imputing missing data on the estimates of the total daily number of steps:

1. About 14% of new values are set, all of them with the same value of the all mean, so the result is that we drive both the mean and median to values closer to the old mean
2. This drove both the median and the mean to the same value.

## 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.

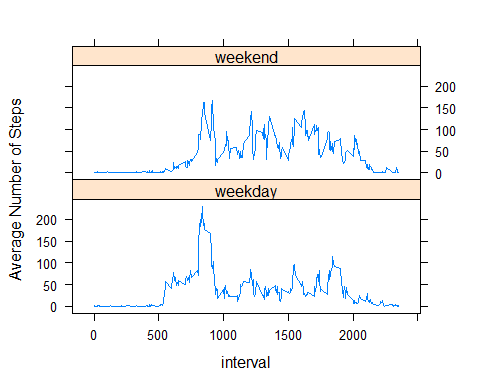
Sys.setenv(LANGUAGE = "en")  
Sys.setlocale("LC\_TIME", "English")

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

activity2$weekday <- c("weekday")  
activity2[weekdays(as.Date(activity2[, 2])) %in% c("Saturday", "Sunday", "saturday", "sunday"), ][4] <- c("weekend")  
activity2$weekday <- factor(activity2$weekday)  
echo=TRUE

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)

activity2\_weekend <- subset(activity2, activity2$weekday == "weekend")  
activity2\_weekday <- subset(activity2, activity2$weekday == "weekday")  
mean\_activity2\_weekday <- tapply(activity2\_weekday$steps, activity2\_weekday$interval, mean)  
mean\_activity2\_weekend <- tapply(activity2\_weekend$steps, activity2\_weekend$interval, mean)  
library(lattice)  
df\_weekday <- data.frame(interval = unique(activity2\_weekday$interval), avg = as.numeric(mean\_activity2\_weekday), day = rep("weekday", length(mean\_activity2\_weekday)))  
df\_weekend <- data.frame(interval = unique(activity2\_weekend$interval), avg = as.numeric(mean\_activity2\_weekend), day = rep("weekend", length(mean\_activity2\_weekend)))  
df\_final <- rbind(df\_weekday, df\_weekend)  
xyplot(avg ~ interval | day, data = df\_final, layout = c(1, 2), type = "l", ylab = "Average Number of Steps")



echo=TRUE