### Load the Data

1.) Load and preprocess the data

setwd("C:\\Users\\bef\\Desktop\\repdata\_data\_activity")  
activity<-read.delim("activity.csv",sep=",",head=T,skip=0,as.is=T)  
summary(activity)

## steps date interval   
## Min. : 0.00 Length:17568 Min. : 0.0   
## 1st Qu.: 0.00 Class :character 1st Qu.: 588.8   
## Median : 0.00 Mode :character Median :1177.5   
## Mean : 37.38 Mean :1177.5   
## 3rd Qu.: 12.00 3rd Qu.:1766.2   
## Max. :806.00 Max. :2355.0   
## NA's :2304

head(activity)

## steps date interval  
## 1 NA 2012-10-01 0  
## 2 NA 2012-10-01 5  
## 3 NA 2012-10-01 10  
## 4 NA 2012-10-01 15  
## 5 NA 2012-10-01 20  
## 6 NA 2012-10-01 25

2.) Process/transform the data (if necessary) into a format suitable for your analysis

activity$date <- as.Date(activity$date, "%Y-%m-%d")  
daily\_steps <- aggregate(steps ~ date, data = activity, sum, na.rm = TRUE)

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

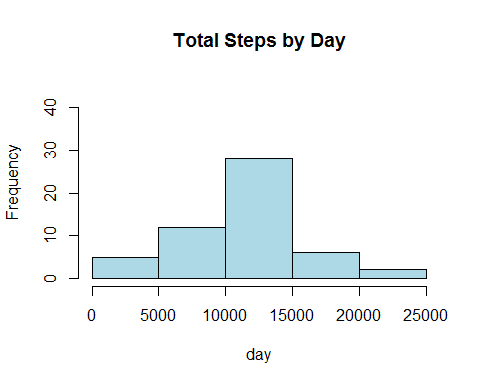
1.) Calculate the total number of steps taken per day

sum(daily\_steps$steps,na.rm=T)

## [1] 570608

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

hist(daily\_steps$steps,xlab="day",col="light blue",main="Total Steps by Day",ylim=c(0,45))



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

round(mean(daily\_steps$steps,na.rm=T),2)

## [1] 10766.19

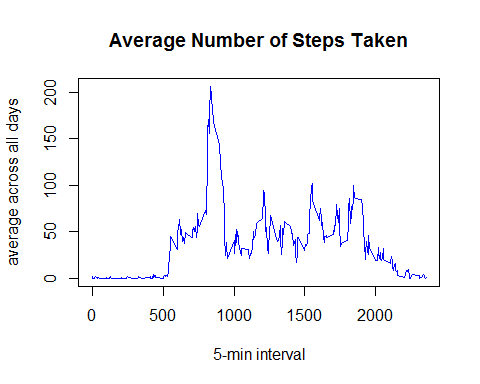
round(median(daily\_steps$steps,na.rm=T),2)

## [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)

time\_series<-tapply(activity$steps, activity$interval, mean, na.rm = TRUE)  
  
plot(row.names(time\_series),time\_series,type="l",col="blue",ylab="average across all days"  
,xlab = "5-min interval",main="Average Number of Steps Taken")



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

max(time\_series)

## [1] 206.1698

which(time\_series==max(time\_series))

## 835   
## 104

### 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)

length(which(is.na(activity$steps)==TRUE))

## [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.

NA\_time\_series<-activity$interval[which(is.na(activity$steps)==TRUE)]  
head(row.names(time\_series))

## [1] "0" "5" "10" "15" "20" "25"

mat1<-matrix(NA, ncol=1, nrow=nrow(activity))  
  
for(i in 1: length(NA\_time\_series))  
 {  
 index<-which(row.names(time\_series)==NA\_time\_series[i])  
 mat1[which(is.na(activity$steps)==TRUE)[i]]<-time\_series[index]  
 }   
   
 head(activity)

## steps date interval  
## 1 NA 2012-10-01 0  
## 2 NA 2012-10-01 5  
## 3 NA 2012-10-01 10  
## 4 NA 2012-10-01 15  
## 5 NA 2012-10-01 20  
## 6 NA 2012-10-01 25

head(mat1[1:nrow(activity)])

## [1] 1.7169811 0.3396226 0.1320755 0.1509434 0.0754717 2.0943396

new\_act<-cbind(mat1,activity)  
   
   
 for(i in 1: nrow(new\_act))  
 {  
 if(is.na(new\_act[i,2])==TRUE){  
 new\_act[i,2]<-new\_act[i,1]  
 } else{  
 new\_act[i,2]<-new\_act[i,2]  
 }  
 }  
  
head(new\_act)

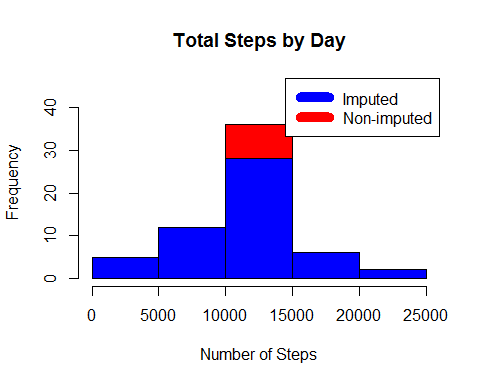
## mat1 steps date interval  
## 1 1.7169811 1.7169811 2012-10-01 0  
## 2 0.3396226 0.3396226 2012-10-01 5  
## 3 0.1320755 0.1320755 2012-10-01 10  
## 4 0.1509434 0.1509434 2012-10-01 15  
## 5 0.0754717 0.0754717 2012-10-01 20  
## 6 2.0943396 2.0943396 2012-10-01 25

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

new\_d.set<-new\_act[,2:4]

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?

new\_d.steps<-aggregate(steps ~ date, data = new\_d.set, sum, na.rm = TRUE)  
hist(new\_d.steps$steps,xlab="Number of Steps",col="red",main="Total Steps by Day",ylim=c(0,45))  
hist(daily\_steps$steps,xlab="Number of Steps",col="blue",main="Total Steps by Day",ylim=c(0,45), add=T)  
legend("topright", c("Imputed", "Non-imputed"), col=c("blue", "red"), lwd=10)



The histogram has a higher peak height. The frequency now maxes out at 35 instead of 30.

round(mean(new\_d.steps$steps),2)

## [1] 10766.19

median(new\_d.steps$steps)

## [1] 10766.19

After replacing the values the mean is the same but the median is slightly different.

### 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.

col.names

new\_d.set[,4]<-weekdays(new\_d.set[,2])  
  
  
  
head(data.frame(new\_d.set,stringsAsFactors=FALSE))

## steps date interval V4  
## 1 1.7169811 2012-10-01 0 Monday  
## 2 0.3396226 2012-10-01 5 Monday  
## 3 0.1320755 2012-10-01 10 Monday  
## 4 0.1509434 2012-10-01 15 Monday  
## 5 0.0754717 2012-10-01 20 Monday  
## 6 2.0943396 2012-10-01 25 Monday

day\_type<-matrix(NA, ncol=1, nrow=length(weekdays(new\_d.set[,2])))  
  
for(i in 1:length(weekdays(new\_d.set[,2])))  
{  
  
if(new\_d.set[,4][i]=="Saturday")  
 {  
 day\_type[i,1]<-"weekend"  
 } else if(new\_d.set[,4][i]=="Sunday")  
 {  
 day\_type[i,1]<-"weekend"  
 } else{  
 day\_type[i,1]<-"weekday"  
 }  
}

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.

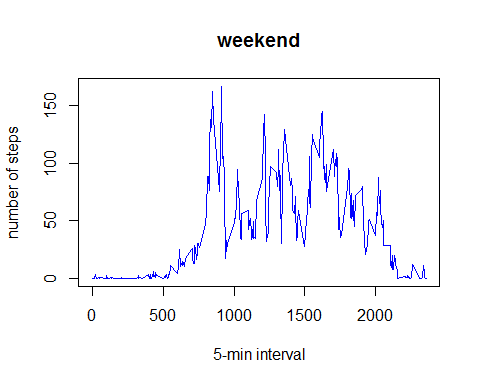
new\_d.set<-cbind(new\_d.set,day\_type)  
colnames(new\_d.set)<-c("steps","date","interval","day","day\_type")

separate weekday and weekend by which function.

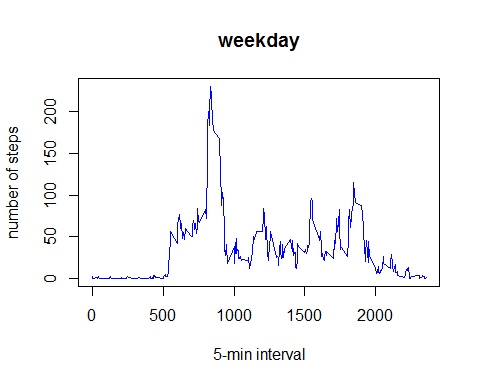
new\_d.set\_we<- new\_d.set[which((new\_d.set)[,5]=="weekend"),]  
new\_d.set\_wd<- new\_d.set[which((new\_d.set)[,5]=="weekday"),]  
  
new\_d.set2\_we<-tapply(new\_d.set\_we$steps, new\_d.set\_we$interval, mean, na.rm = TRUE)  
new\_d.set2\_wd<-tapply(new\_d.set\_wd$steps, new\_d.set\_wd$interval, mean, na.rm = 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). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

plot(row.names(new\_d.set2\_we),new\_d.set2\_we,type="l",col="blue",ylab="number of steps"  
,xlab = "5-min interval",main="weekend")



plot(row.names(new\_d.set2\_wd),new\_d.set2\_wd,type="l",col="blue",ylab="number of steps"  
,xlab = "5-min interval",main="weekday")



There are differences in activity. The weekday has more activity in the morning and the weekend has more activity in the afternoon.