# Reproducible Research: Peer Assessment 1

library(lattice)

## Loading and cleaning the data

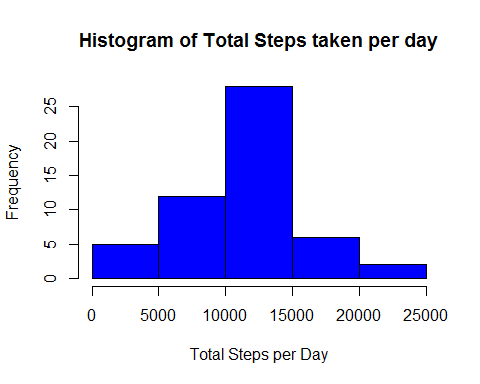
activity\_data <- read.csv("activity.csv")  
activity\_data$date <- as.Date(activity\_data$date,"%Y-%m-%d")

## Mean total number of steps taken per day

total\_steps\_per\_day <- tapply(activity\_data$steps, activity\_data$date,sum)

## Histogram of total number of steps per day

hist(total\_steps\_per\_day,col="blue",xlab="Total Steps per Day",   
 ylab="Frequency", main="Histogram of Total Steps taken per day")

 Compute Mean total steps taken per day

mean(total\_steps\_per\_day,na.rm=TRUE)

## [1] 10766.19

Compute Median total steps taken per day

median(total\_steps\_per\_day,na.rm=TRUE)

## [1] 10765

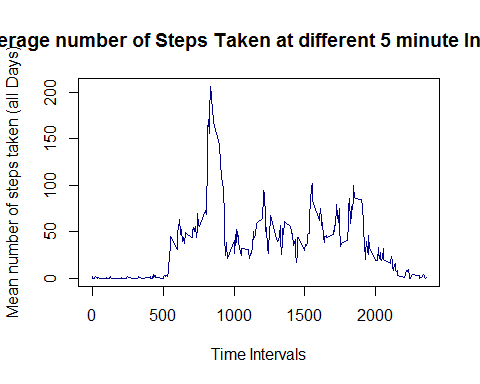
## Average daily activity pattern

Compute mean of steps over all days by time interval

mean\_steps\_by\_interval <- tapply(activity\_data$steps,activity\_data$interval,  
 mean,na.rm=TRUE)

Timeseries plot of of the 5-minute interval and the average number of steps taken, averaged across all days

plot(row.names(mean\_steps\_by\_interval),mean\_steps\_by\_interval,type="l",  
 xlab="Time Intervals",   
 ylab="Mean number of steps taken (all Days)",   
 main="Average number of Steps Taken at different 5 minute Intervals",  
 col="darkblue")

 Time interval that contains maximum average number of steps over all days

interval\_num <- which.max(mean\_steps\_by\_interval)  
interval\_max\_steps <- names(interval\_num)  
interval\_max\_steps

## [1] "835"

The r interval\_max\_steps minute or \*\* 104th \*\* 5 minute interval contains the maximum number of steps on average across all the days

## Inputing missing values

Compute the number of NA values in the activity dataset

num\_na\_values <- sum(is.na(activity\_data))  
num\_na\_values #Print number of NA values

## [1] 2304

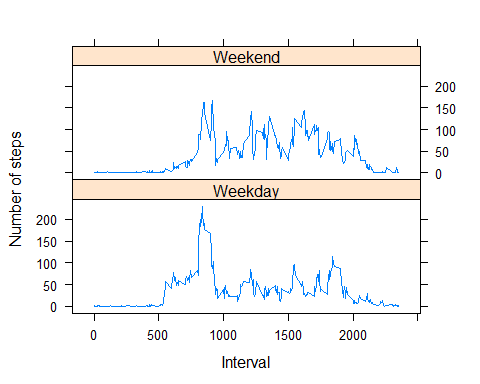
Fill in missing values using the **average interval value across all days**

na\_indices <- which(is.na(activity\_data))  
imputed\_values <- mean\_steps\_by\_interval[as.character(activity\_data[na\_indices,3])]  
names(imputed\_values) <- na\_indices  
for (i in na\_indices) {  
 activity\_data$steps[i] = imputed\_values[as.character(i)]  
}  
sum(is.na(activity\_data)) # The number of NAs after imptation should be 0

## [1] 0

## Finding differences in activity patterns between weekdays and weekends

days <- weekdays(activity\_data$date)  
activity\_data$day\_type <- ifelse(days == "Saturday" | days == "Sunday",   
 "Weekend", "Weekday")  
mean\_steps\_by\_interval <- aggregate(activity\_data$steps,  
 by=list(activity\_data$interval,  
 activity\_data$day\_type),mean)  
names(mean\_steps\_by\_interval) <- c("interval","day\_type","steps")  
xyplot(steps~interval | day\_type, mean\_steps\_by\_interval,type="l",  
 layout=c(1,2),xlab="Interval",ylab = "Number of steps")

 Mean, median, max and min of the steps across all intervals and days by Weekdays/Weekends

tapply(mean\_steps\_by\_interval$steps,mean\_steps\_by\_interval$day\_type,  
 function (x) { c(MIN=min(x),MEAN=mean(x),  
 MEDIAN=median(x),MAX=max(x))})

## $Weekday  
## MIN MEAN MEDIAN MAX   
## 0.00000 35.61058 25.80314 230.37820   
##   
## $Weekend  
## MIN MEAN MEDIAN MAX   
## 0.00000 42.36640 32.33962 166.63915