# MODULE 5 WEEK 2

# Loading and preprocessing the data

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

data\_input<-read.table("activity.csv",sep=",",header=TRUE)

Then, for processing the dates as “Date” variable, it needs to be transformed into date format which is done as follows -

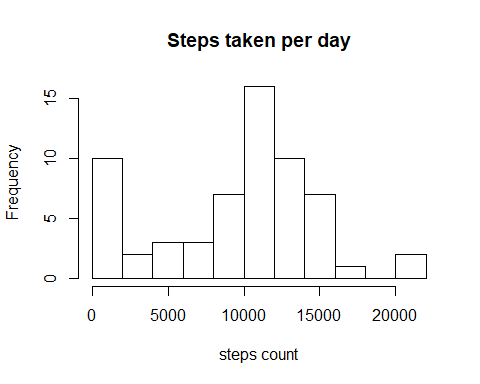
# processing data  
data\_input$date <- as.Date(data\_input$date)

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

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

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

hist(steps\_day, breaks = 8, xlab = "steps count", main = "Steps taken per day")



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

mean\_steps<-mean(steps\_day,na.rm=TRUE)  
median\_steps<-median(steps\_day,na.rm=TRUE)  
mean\_steps

## [1] 9354.23

median\_steps

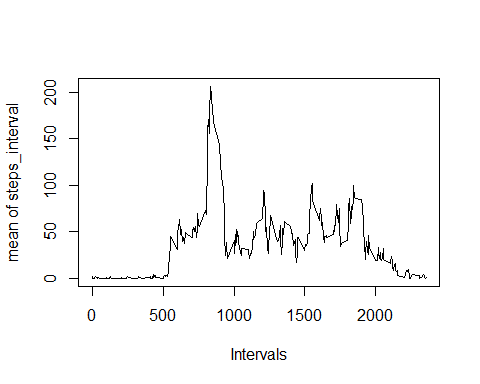
## [1] 10395

## What is the average daily activity pattern?

We want to create a time series plot for the step count in each interval.

* First, interval-wise average step count need to be calculated
* Second, plotting the required figure

steps\_interval<-tapply(data\_input$steps,data\_input$interval,mean,na.rm=TRUE)  
plot(x=names(steps\_interval),y=steps\_interval,type="l",xlab="Intervals",ylab="mean of steps\_interval")



### Next, we need to find which interval contains the maximum number of steps.

max\_steps\_interval<-max(steps\_interval)  
max\_steps\_interval

## [1] 206.1698

# Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as **NA**). The presence of missing days may introduce bias into some calculations or summaries of the data.

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

total\_NA<-sum(is.na(data\_input$steps))  
total\_NA

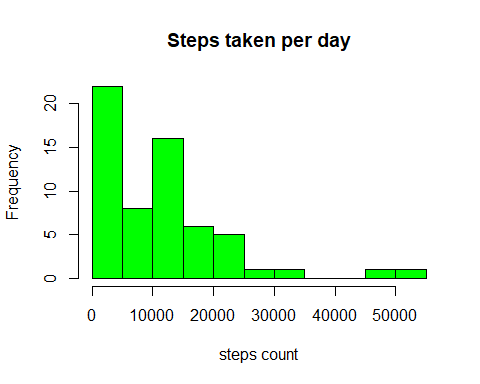
## [1] 2304

## Filling in all of the missing values in the dataset with the mean for that 5-minute interval and Create a new dataset called data\_input\_2 that is equal to the original dataset but with the missing data filled in

df<-data.frame(interval=as.numeric(as.character(names(steps\_interval))),steps\_avg\_interval=steps\_interval)  
data\_input\_2<-merge(data\_input,df,by="interval",all.x=TRUE)  
data\_input\_2[is.na(data\_input\_2$steps),]$steps<-data\_input\_2[is.na(data\_input\_2$steps),]$steps\_avg\_interval

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

steps\_day\_2<-tapply(data\_input\_2$steps,data\_input$date,sum,na.rm=TRUE)  
hist(steps\_day\_2,col="green",breaks = 8, xlab = "steps count", main = "Steps taken per day")



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

mean\_steps\_2<-mean(steps\_day\_2,na.rm=TRUE)  
median\_steps\_2<-median(steps\_day\_2,na.rm=TRUE)  
mean\_steps\_2

## [1] 10766.19

median\_steps\_2

## [1] 10351.62

### We need to find out the difference between the two

print("Difference between means")

## [1] "Difference between means"

mean\_steps\_2 - mean\_steps

## [1] 1411.959

print("Difference between medians")

## [1] "Difference between medians"

median\_steps\_2 - median\_steps

## [1] -43.37736

# 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.

### 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.

data\_input\_2$weekday<-weekdays(as.Date(as.character(data\_input\_2$date)))  
data\_input\_2$weekend<-"weekday"  
data\_input\_2[data\_input\_2$weekday=="sábado" | data\_input\_2$weekday=="domingo",]$weekend<-"weekend"  
data\_input\_2$weekend<-as.factor(data\_input\_2$weekend)

### 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).

results<-with(data\_input\_2, tapply(steps, list(interval, weekend), mean,na.rm=TRUE))  
resultsdf<-as.data.frame(as.table(results))  
resultsdf$Var1<-as.integer(resultsdf$Var1)  
  
library(lattice)  
xyplot(Freq ~ Var1|Var2,data=resultsdf,layout=c(1,2),type="l",xlab="Interval",ylab="Number of steps")

