**################################# ANN#####################################**

**#Objective: To prepare a model for strength of concrete data using Neural Networks**

**#Data : concrete.csv**

**###########################################################################**

install.packages("neuralnet")

library(neuralnet) ## Artifical Neural Network

library(psych) ## Scatter plot matrix

install.packages("lattice")

require("lattice")#Graphical exploration

**##Step1 : Data Exploration**

concrete <- read.csv(file.choose()) #concrete.csv

View(concrete)

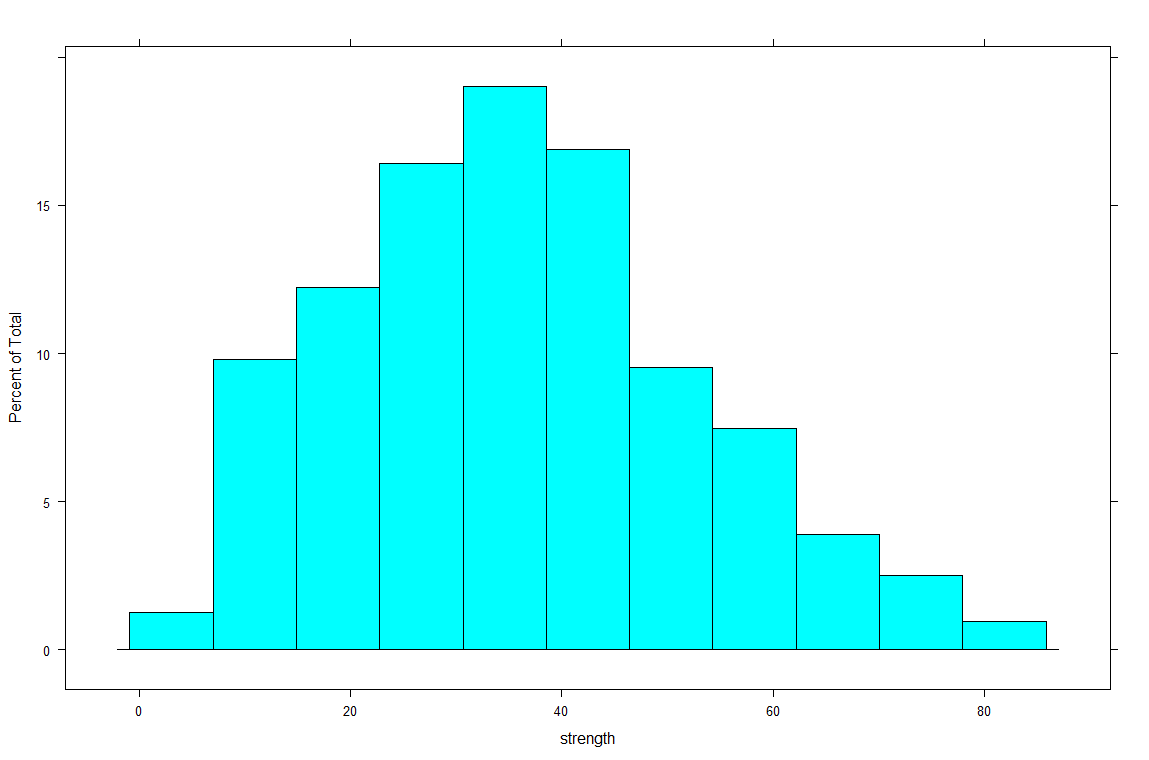
# dataset contains 1030 observations with 9 features.

# Dependent Variable: strength

# Independent variables: cement,slag,ash,water,superplastic,coarseagg,fineaagg,age

**#Data Visualization**

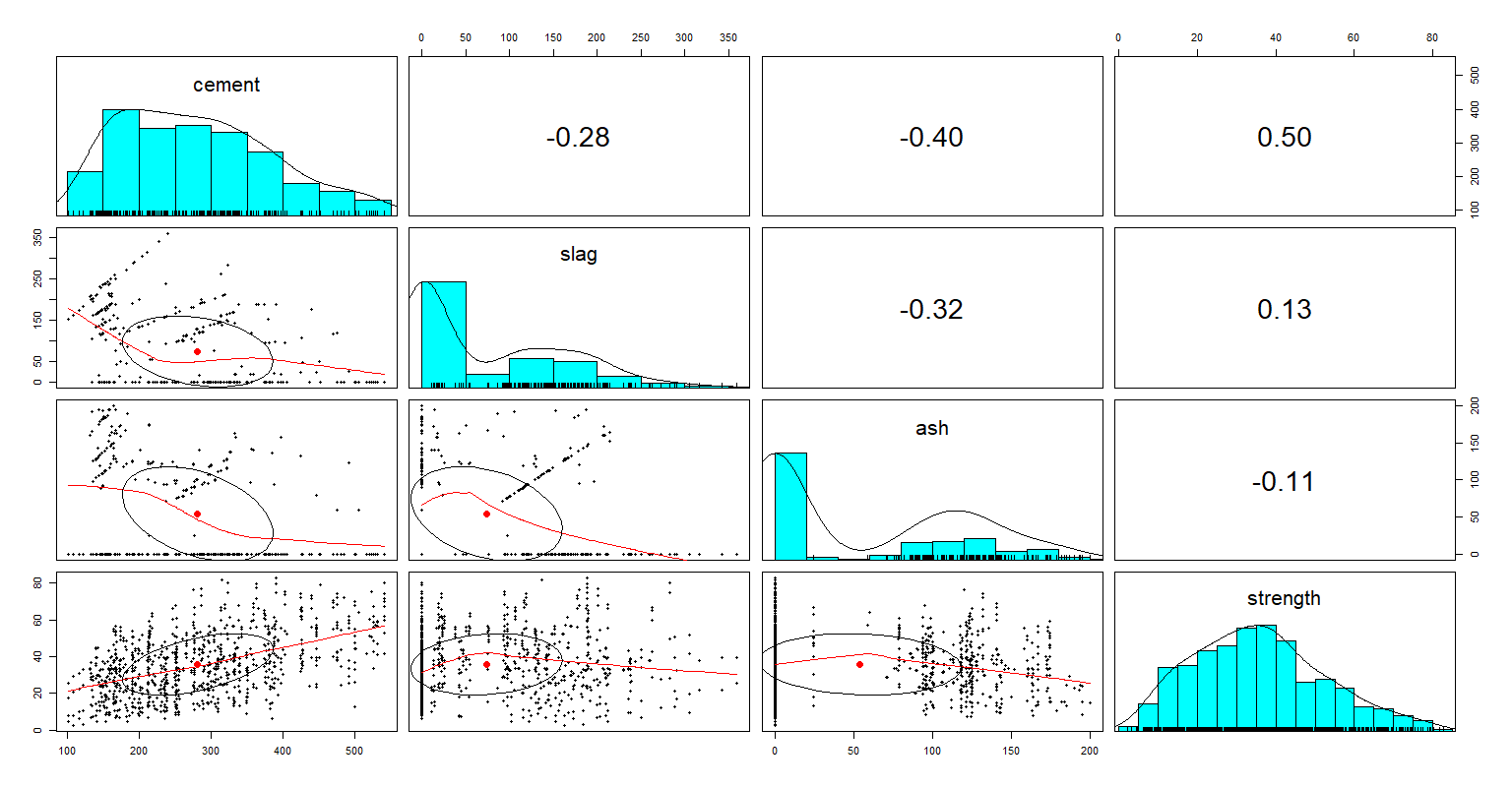
histogram(concrete$strength)



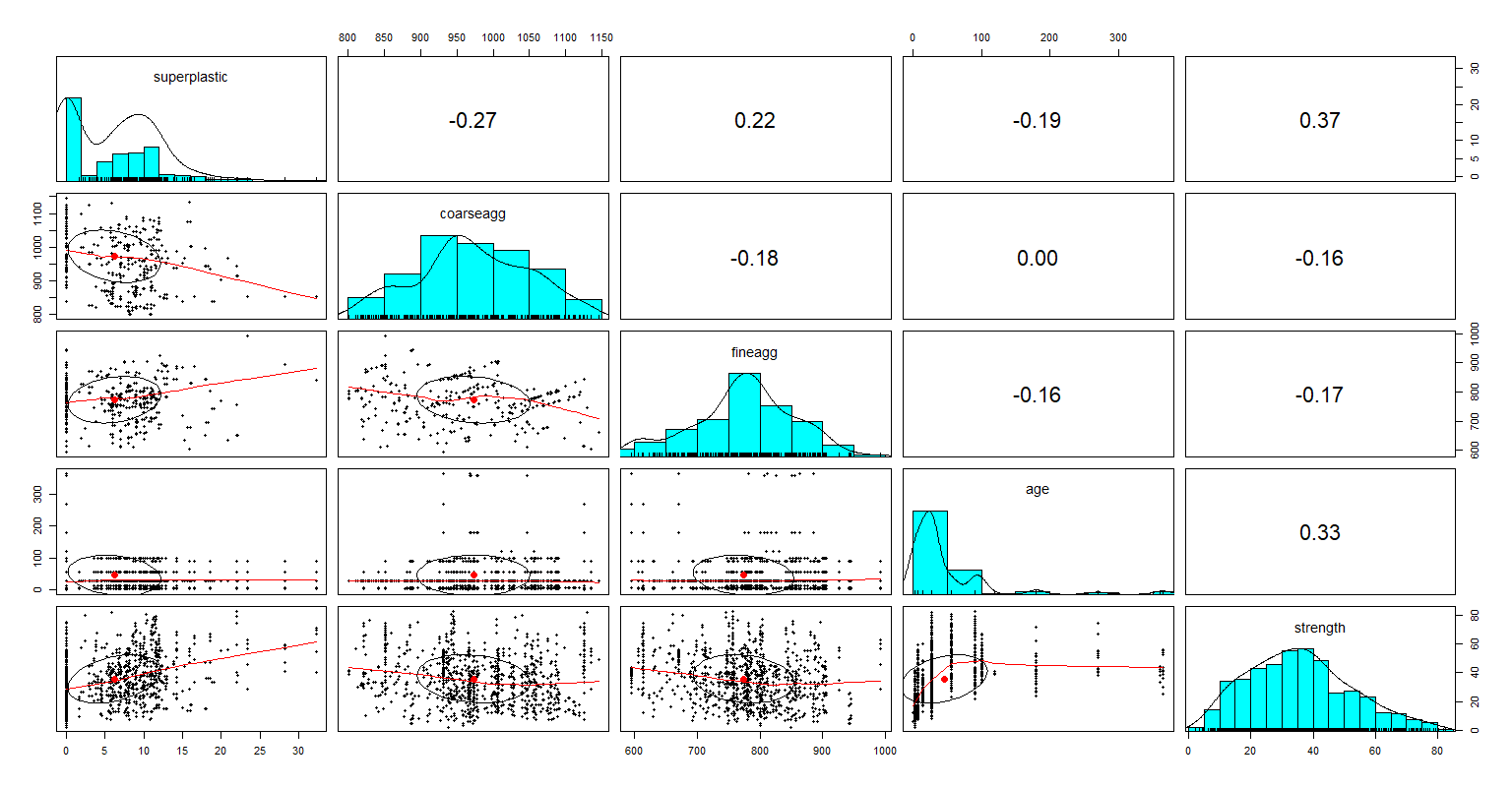
#Distribution is positively skewed.Most of the observations have strength close to mean of 35.5 approx.

#scatter plot

pairs.panels(concrete[c("cement","slag","ash","strength")])



pairs.panels(concrete[c("superplastic","coarseagg","fineagg","age","strength")])



summary(concrete)

#cement slag ash water superplastic coarseagg

#Min. :102.0 Min. : 0.0 Min. : 0.00 Min. :121.8 Min. : 0.000 Min. : 801.0

#1st Qu.:192.4 1st Qu.: 0.0 1st Qu.: 0.00 1st Qu.:164.9 1st Qu.: 0.000 1st Qu.: 932.0

#Median :272.9 Median : 22.0 Median : 0.00 Median :185.0 Median : 6.400 Median : 968.0

#Mean :281.2 Mean : 73.9 Mean : 54.19 Mean :181.6 Mean : 6.205 Mean : 972.9

#3rd Qu.:350.0 3rd Qu.:142.9 3rd Qu.:118.30 3rd Qu.:192.0 3rd Qu.:10.200 3rd Qu.:1029.4

#Max. :540.0 Max. :359.4 Max. :200.10 Max. :247.0 Max. :32.200 Max. :1145.0

#fineagg age strength

#Min. :594.0 Min. : 1.00 Min. : 2.33

#1st Qu.:731.0 1st Qu.: 7.00 1st Qu.:23.71

#Median :779.5 Median : 28.00 Median :34.45

#Mean :773.6 Mean : 45.66 Mean :35.82

#3rd Qu.:824.0 3rd Qu.: 56.00 3rd Qu.:46.13

#Max. :992.6 Max. :365.00 Max. :82.60

#From scatterplot, most of features and dependent variables are not normally distributed.so normalization

#is to be done using customize normalize() function.

**##Step2: Data preprocessing and preparation**

#custom normalization function

normalize <- function(x){

return((x-min(x))/(max(x)-min(x)))

}

# apply normalization to entire data frame

concrete\_norm <- as.data.frame(lapply(concrete, normalize))

View(concrete\_norm)

**#Datapreparation: splitting dataset into training and testing with 75% and 25% proportion resp.**

# create training and test data

concrete\_train <- concrete\_norm[1:773, ]

concrete\_test <- concrete\_norm[774:1030, ]

attach(concrete)

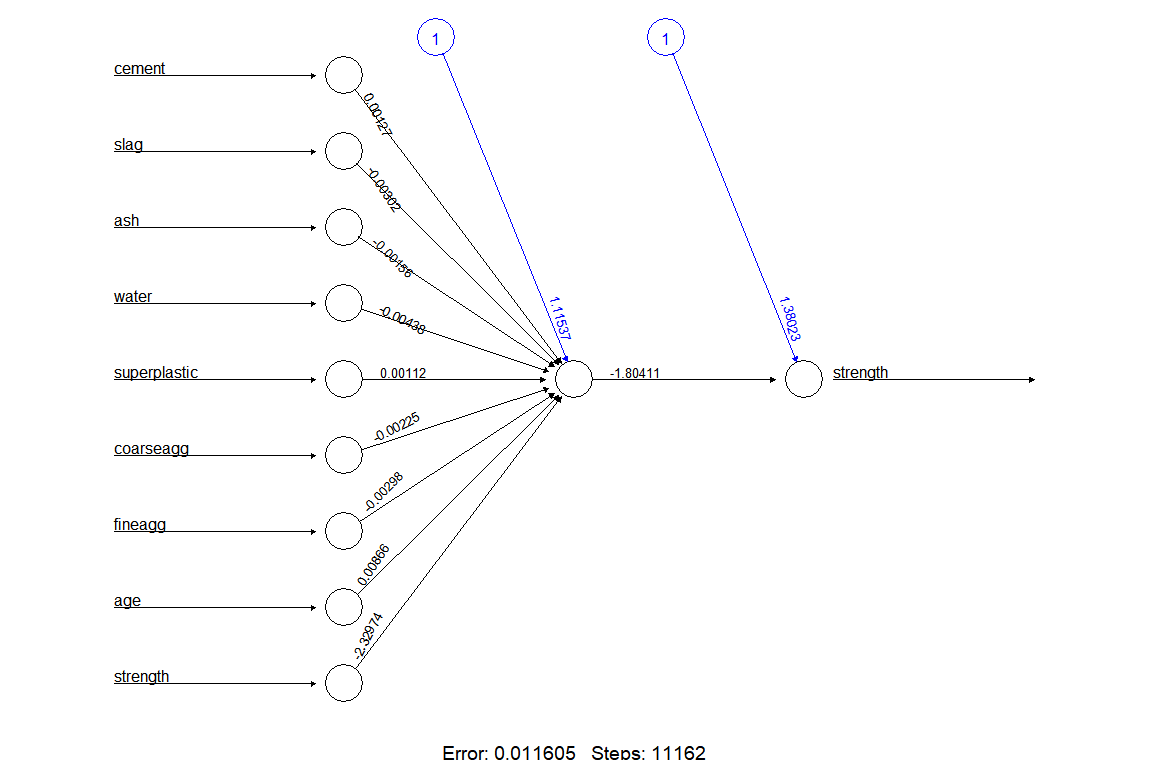
**##Step 3: Model Training**

# Feedforward neuron network with only a one hidden neuron is built on training data.

concrete\_model <- neuralnet(formula = strength ~ cement + slag + ash + water + superplastic + coarseagg + fineagg + age + strength, data = concrete\_train)

**#Visualizing Neural Network**

plot(concrete\_model)



**##Step 4: Model Evaluation**

#building the predictor, exclude the dependent variable column

model\_results <- compute(concrete\_model, concrete\_test[1:8])

View(model\_results)

predicted\_strength <- model\_results$net.result

predicted\_strength

# examine the correlation between predicted and actual values

cor(predicted\_strength1, concrete\_test$strength)

#0.81

**##Step 5: Improving model performance**

gc()

## building back propagation neutral network with two hidden layers with 2 neurons each and tanh as

## activation function and another parameters.

concrete\_model2 <- neuralnet(strength ~ cement + slag +

ash + water + superplastic +

coarseagg + fineagg + age,

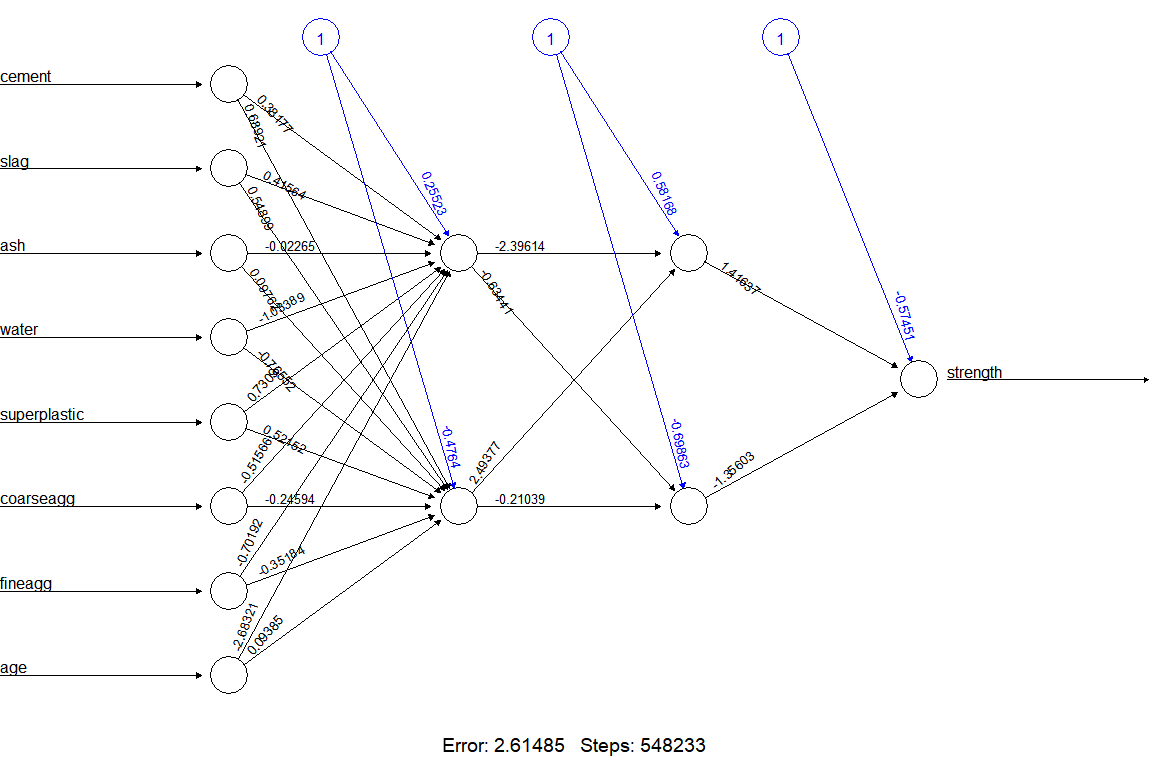
data = concrete\_train, hidden = c(2,2),algorithm = 'backprop',

learningrate = 0.0001,linear.output=F,

stepmax=1e+08,act.fct = 'tanh')

**#Visualizing Neural Network**

plot(concrete\_model2)



#building the new predictor, excluding the dependent variable

model\_results2 <- compute(concrete\_model2, concrete\_test[,-9])

View(model\_results2)

predicted\_strength2 <- model\_results2$net.result

predicted\_strength2

**#Evaluating New Model**

# examine the correlation between predicted and actual values

cor(predicted\_strength2, concrete\_test$strength)

#[,1]

#[1,] 0.9110631