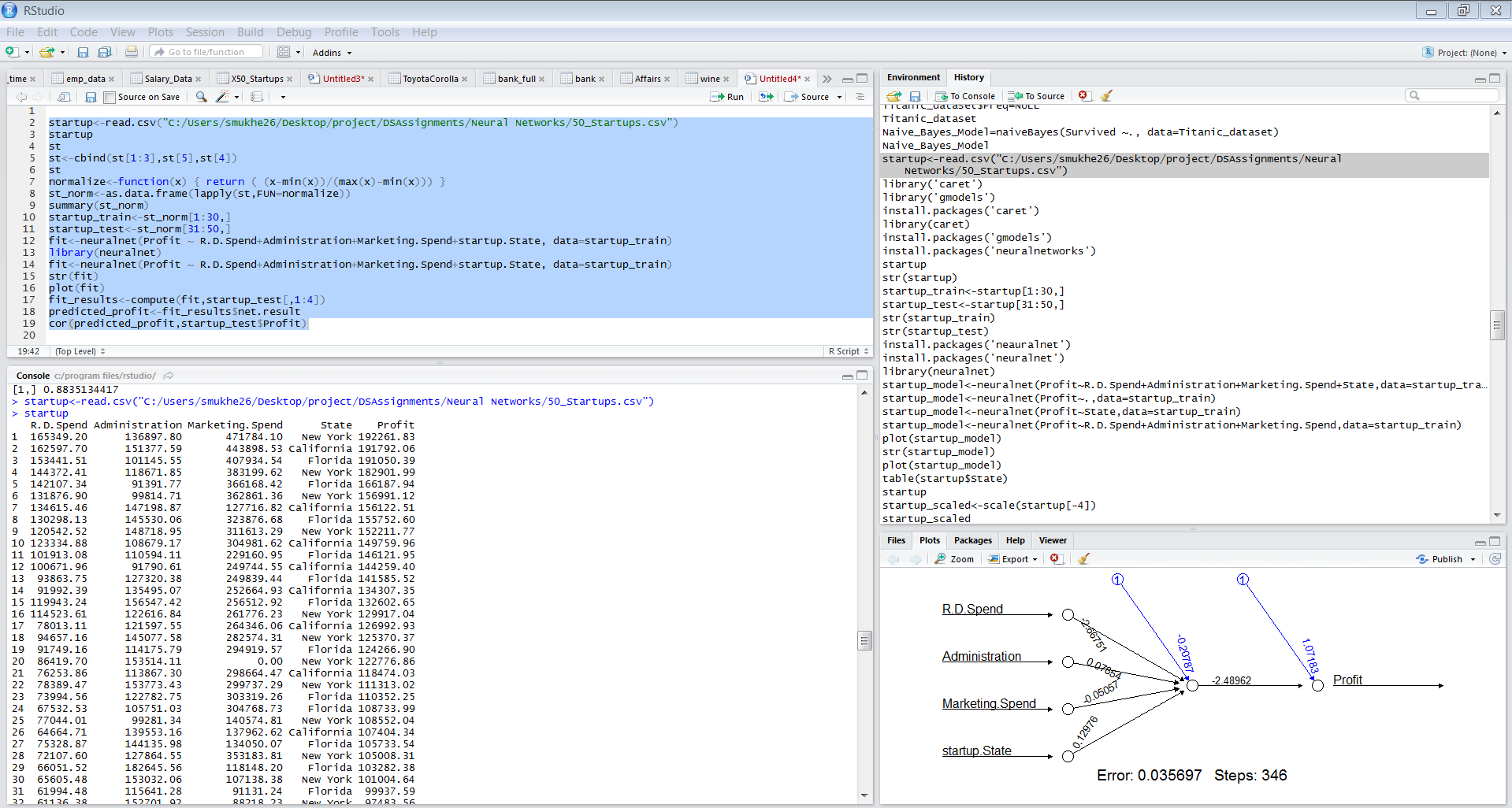
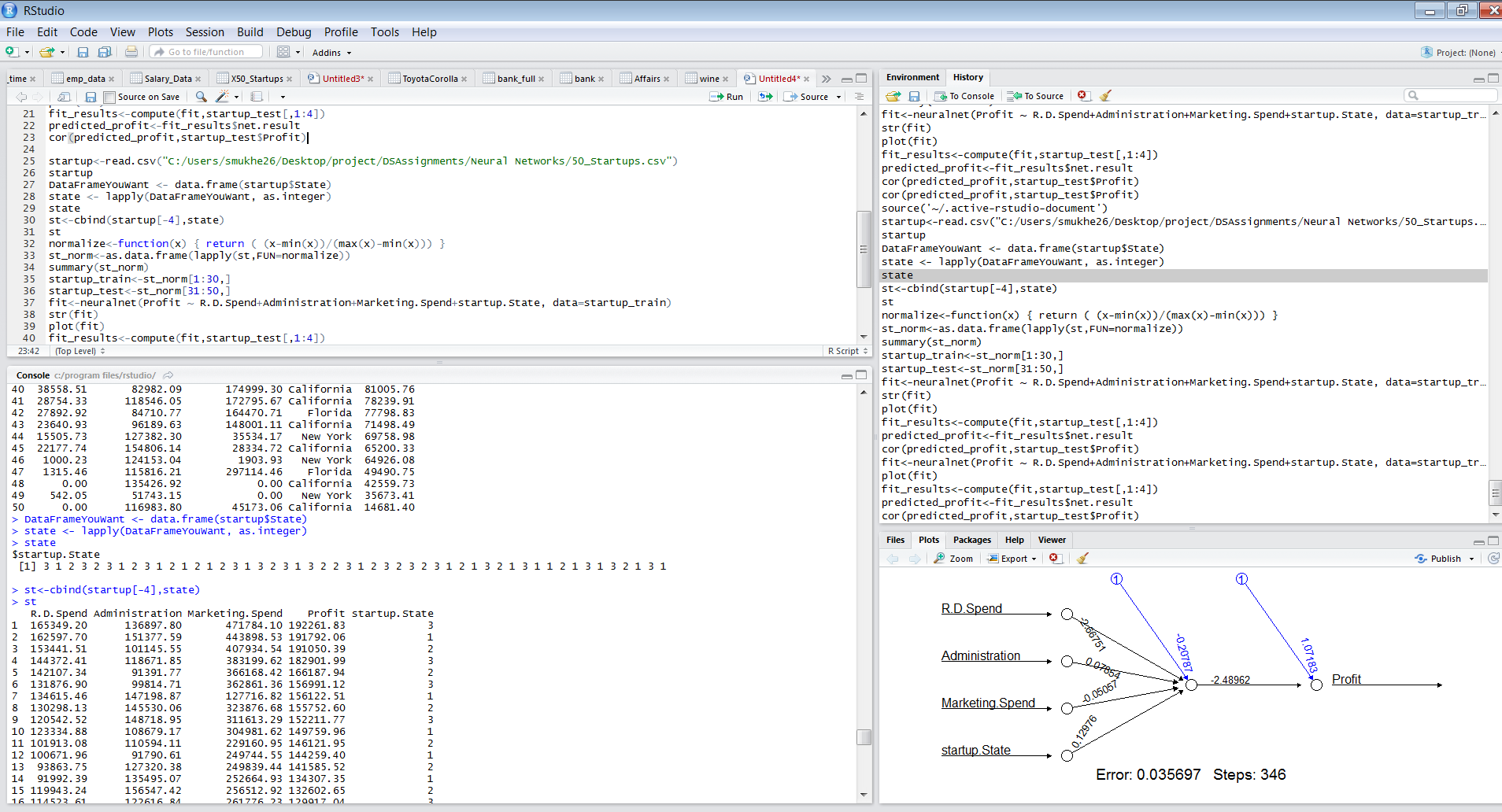
Neural Network Assignment

Startup data:

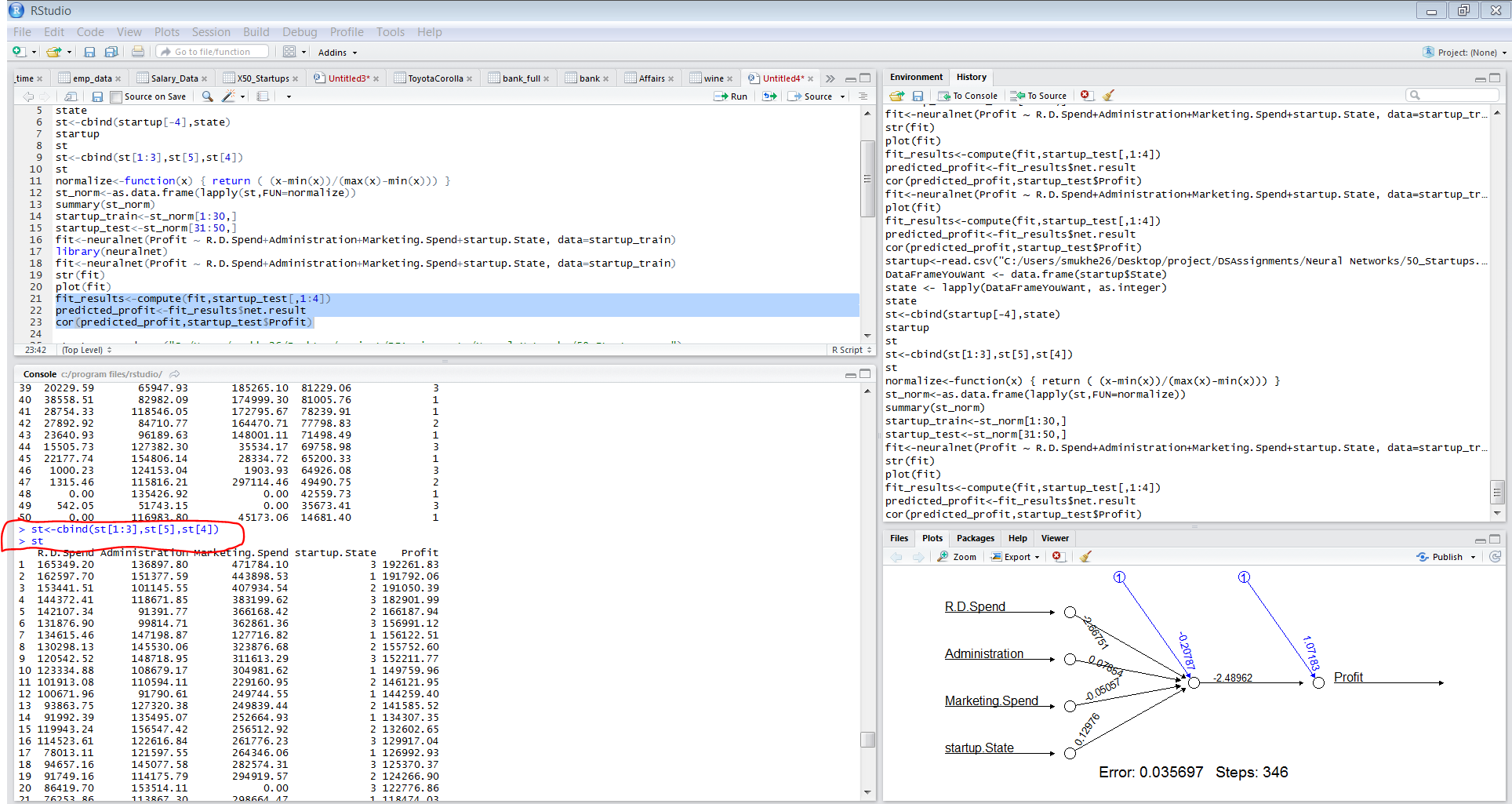
We would work with the startup data and predict the accuracy using neural network. Let us read the entire startup data and try normalizing since we would need scaled data as a feed to neural network



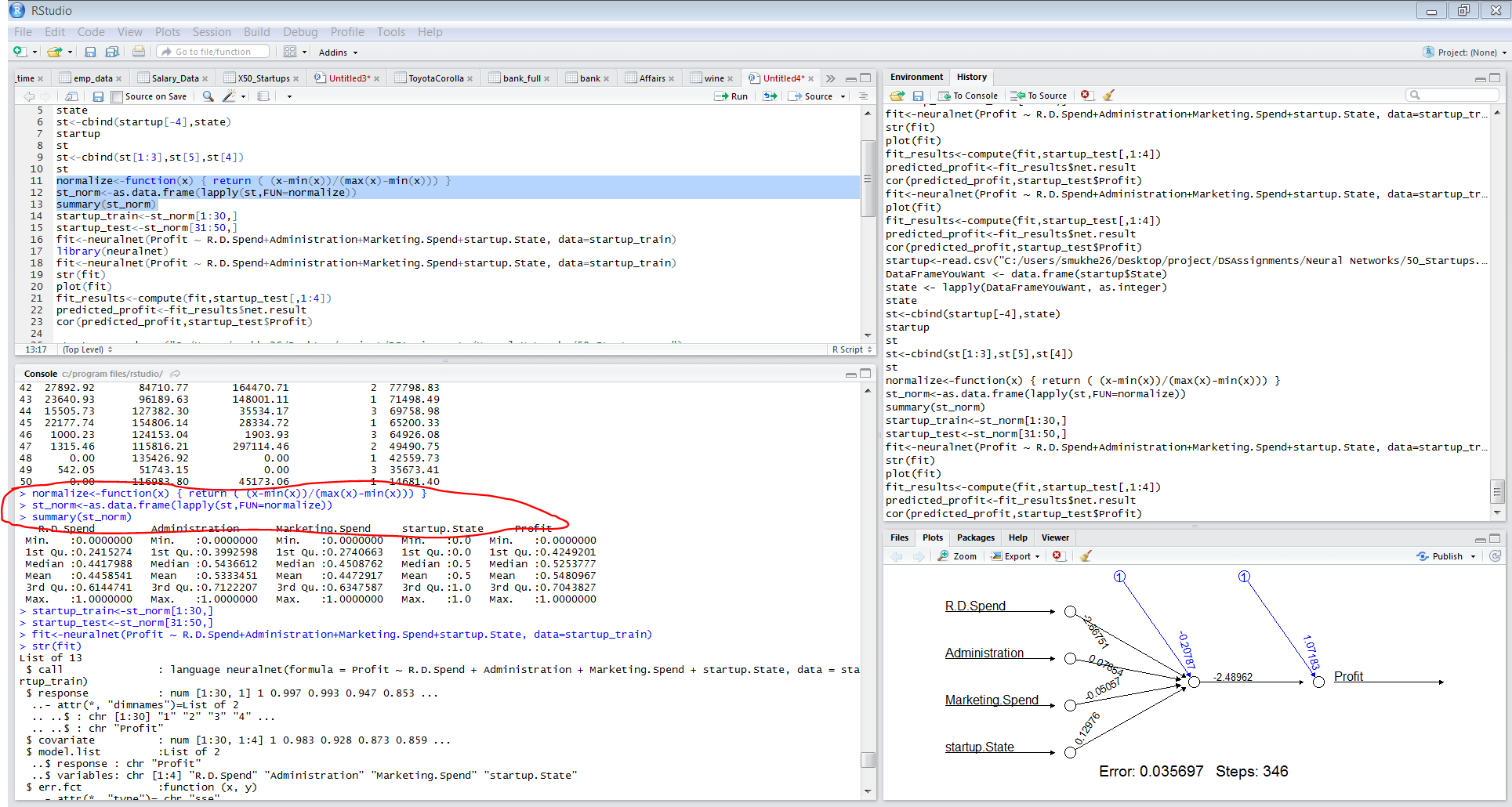
Lets convert the categorical data state to numerical



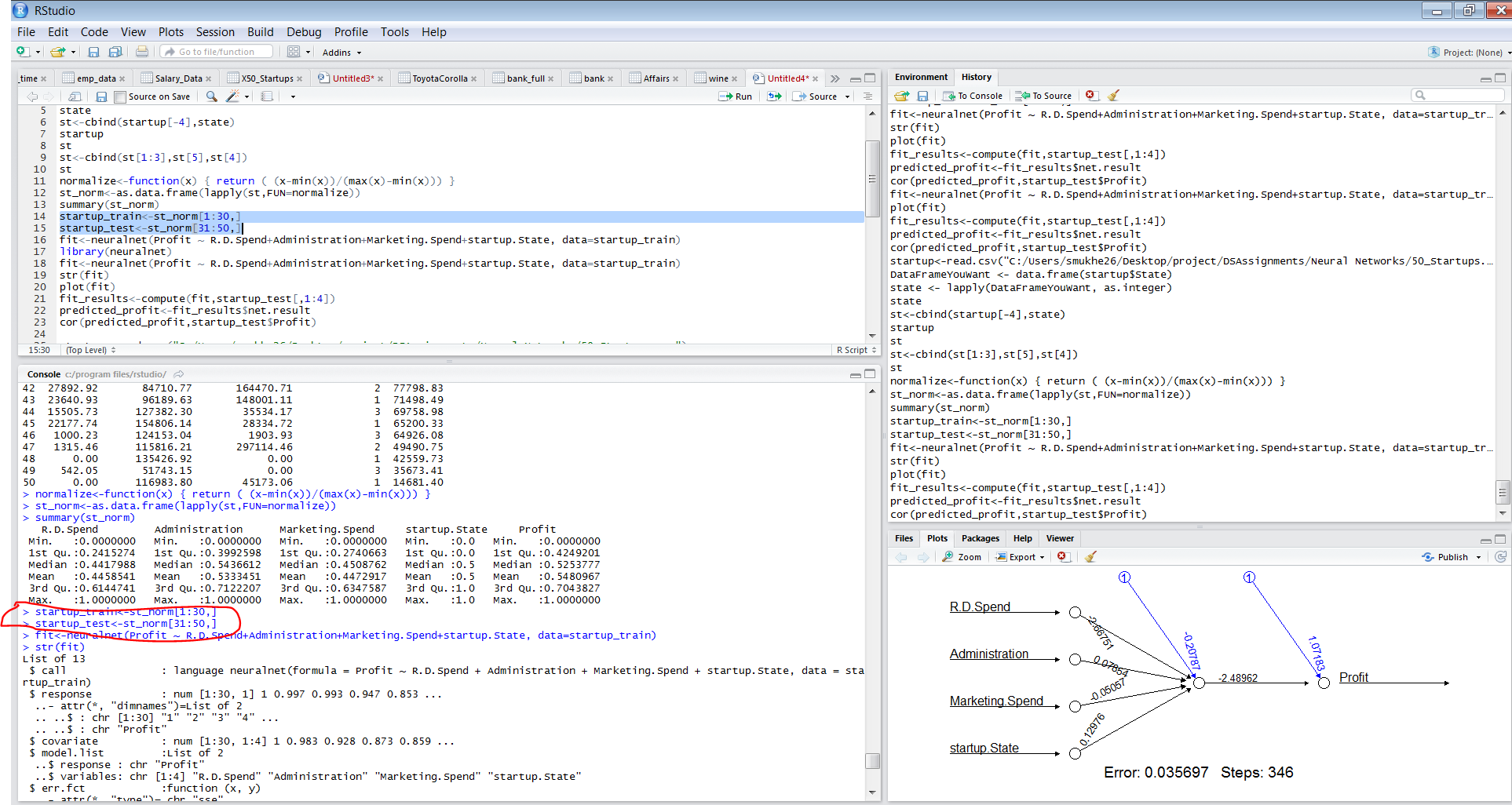
Lets arrange the dataset in such a way that the outcome value would be the last column



Then we have to normalize the entire dataset



Then we divide the entire dataset into training and test data

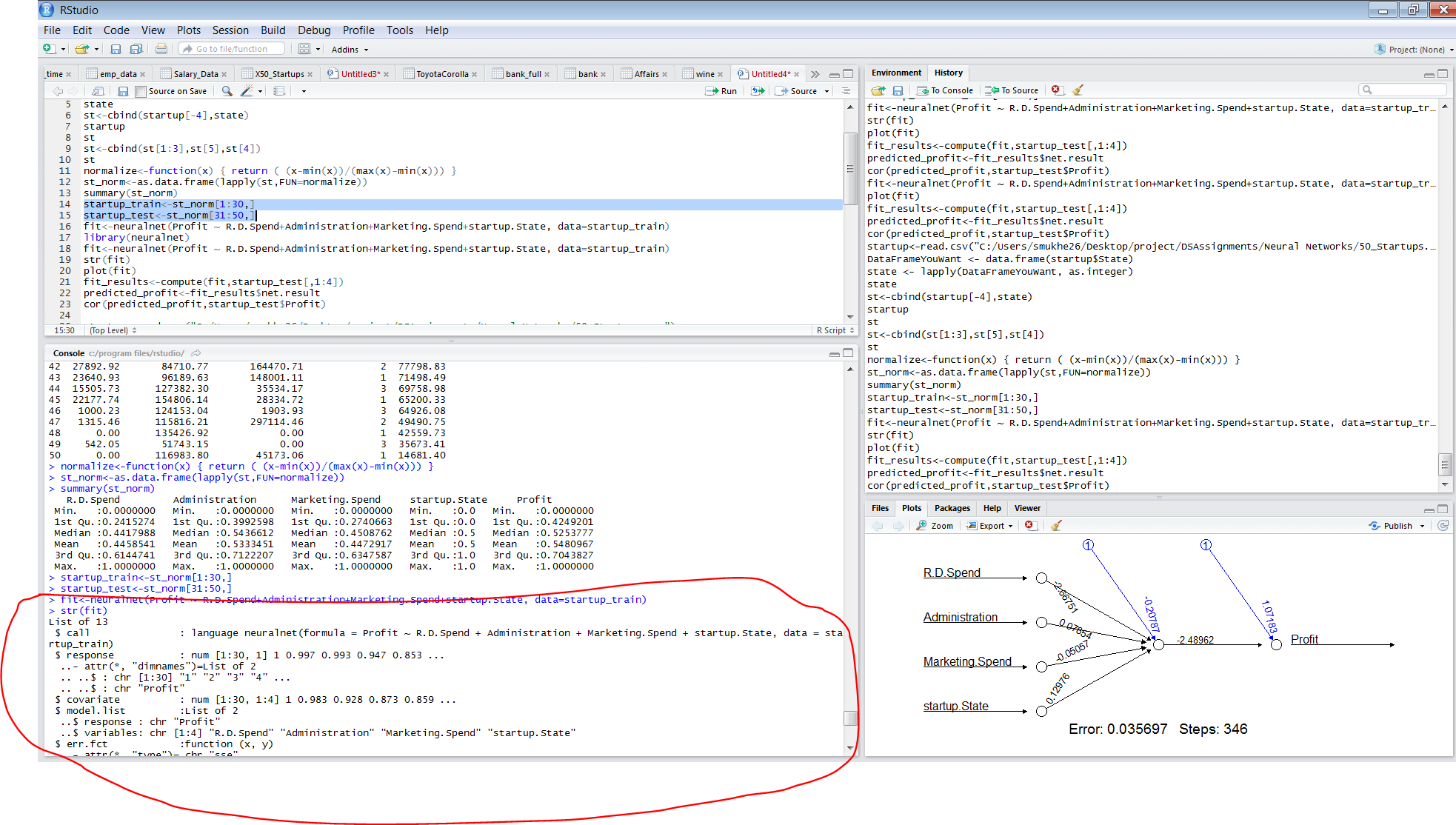


We have to install the package

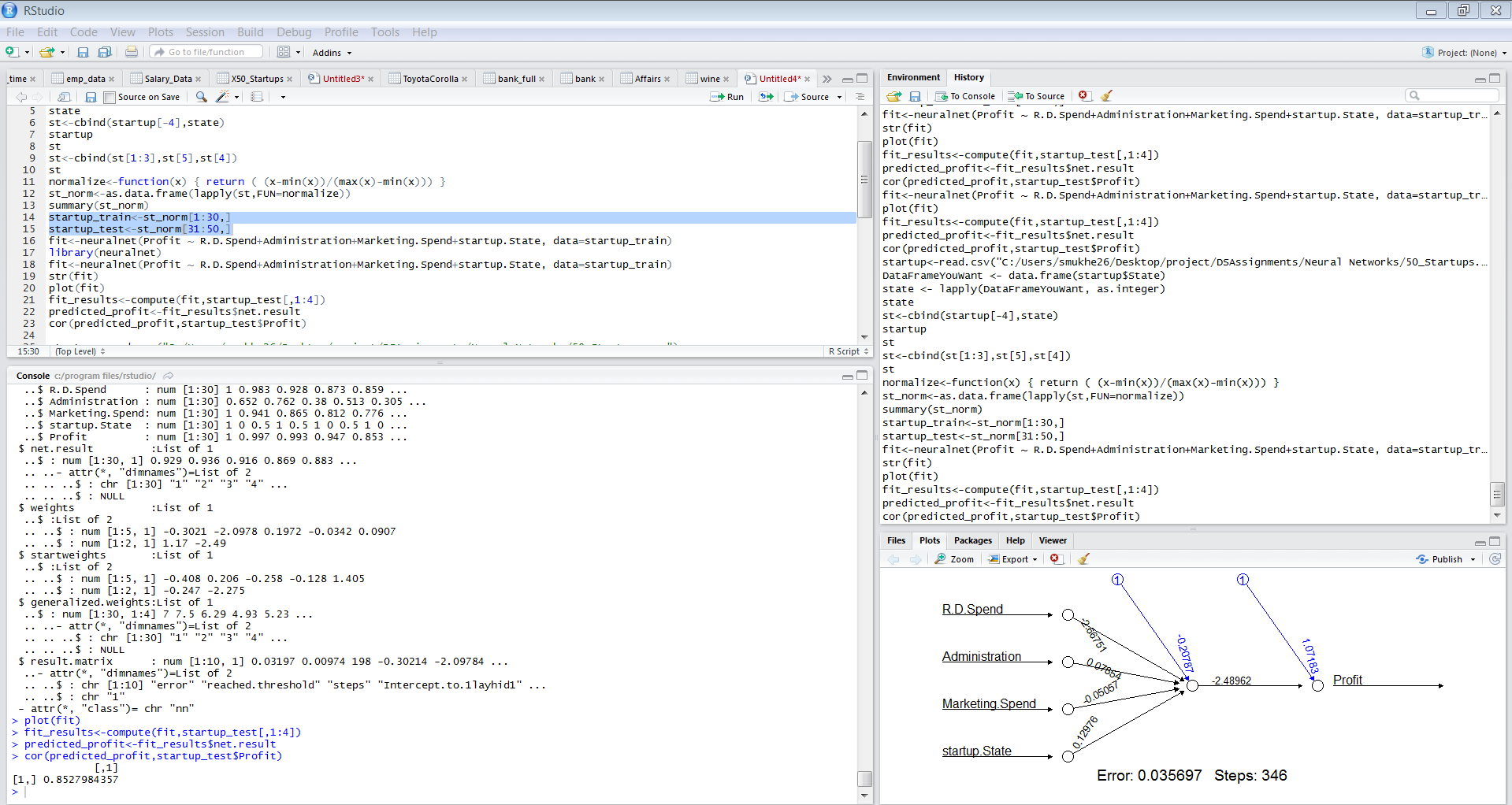
Install.packages(‘neuralnet’)

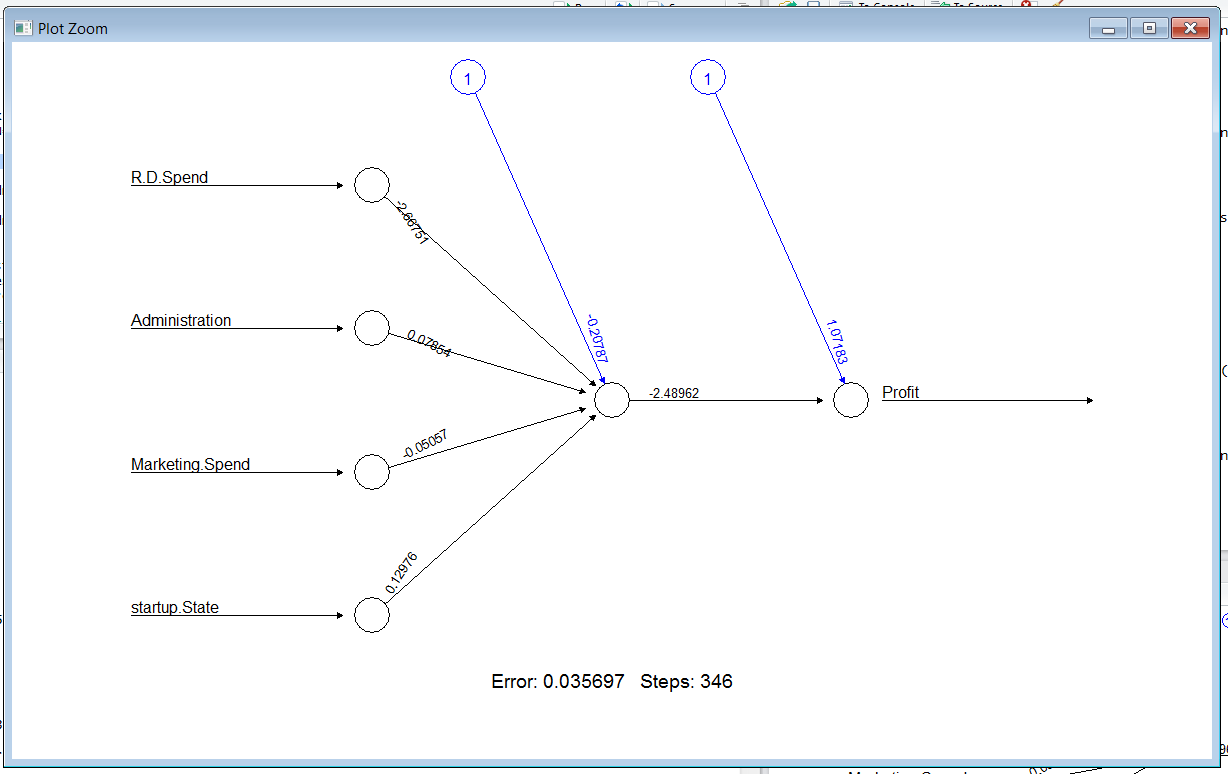
Library(neuralnet)

And then apply the model on the data



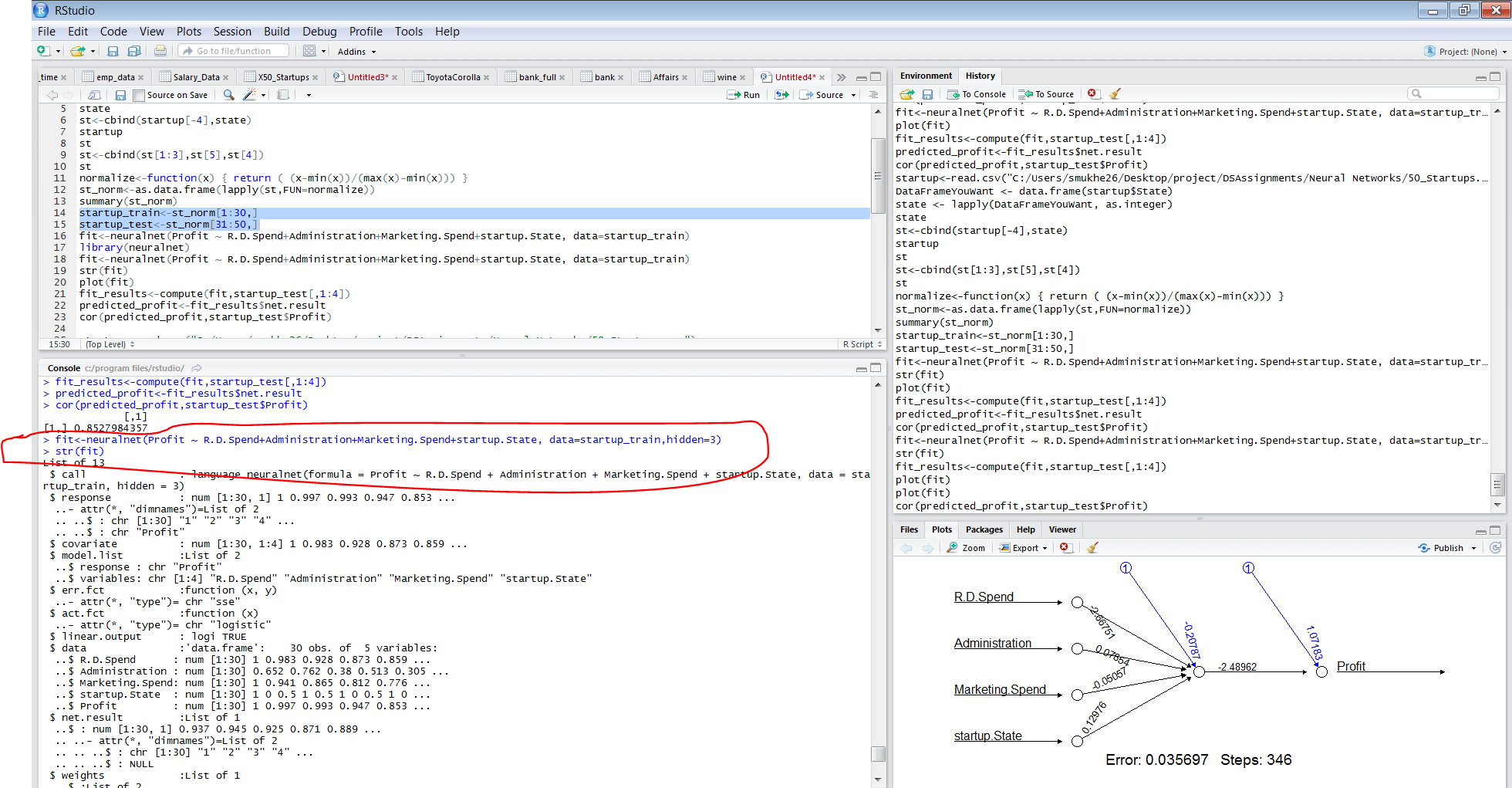
Let’s plot the fit

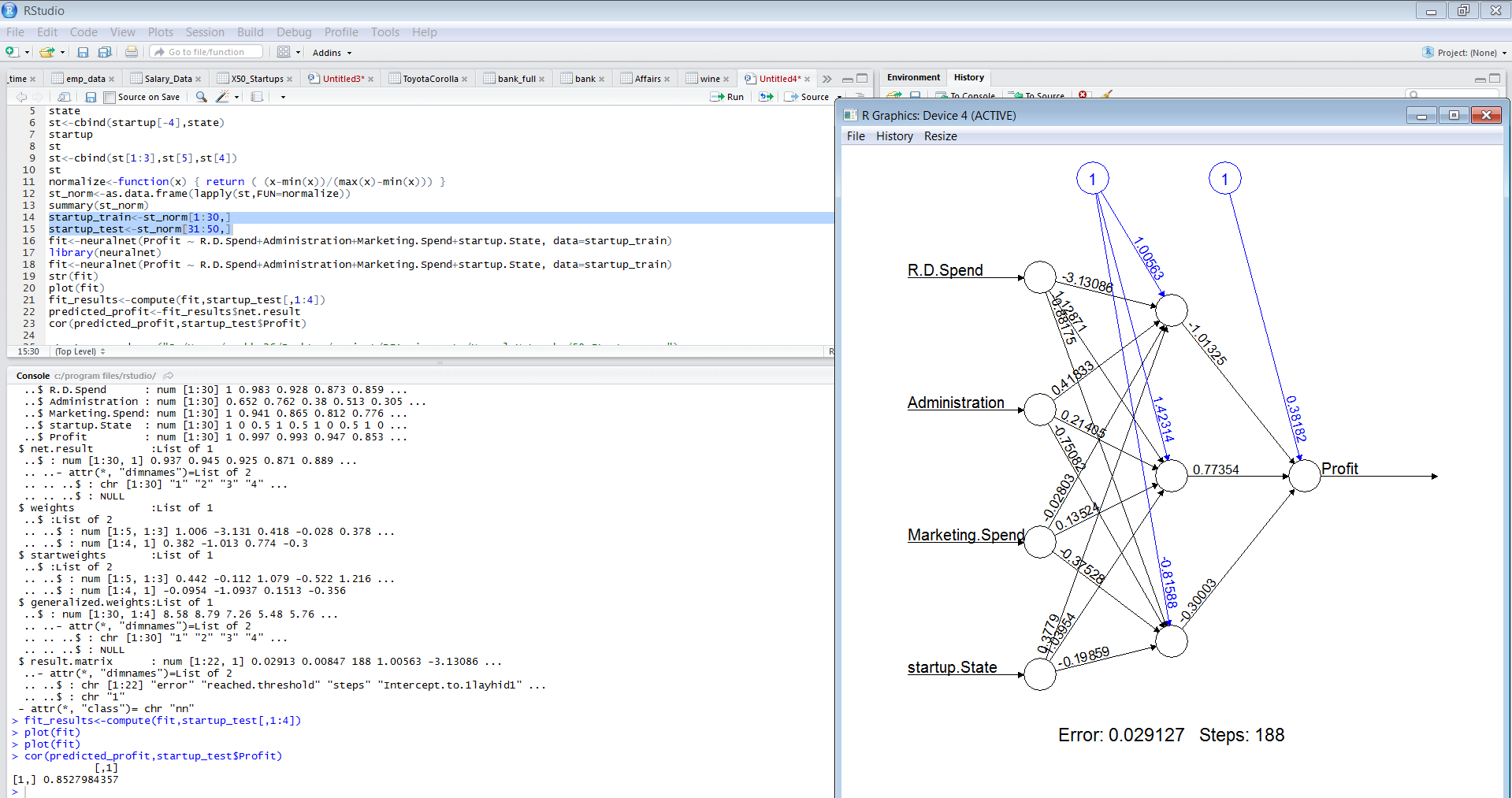


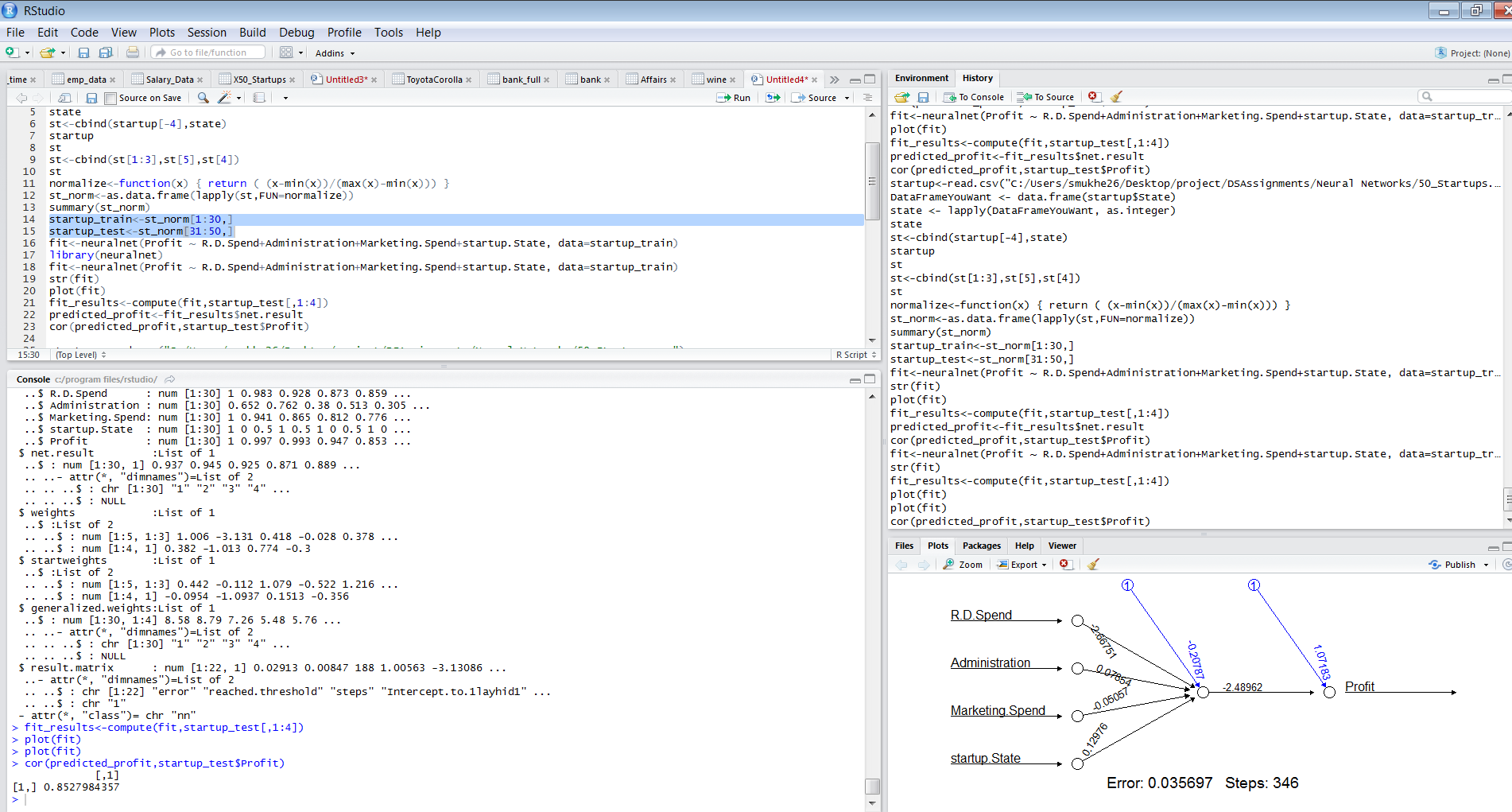


The accuracy being 85 %

Let’s try the parameter hidden=3 and apply the neuralnetworks model again on the dataset







We still have the same accuracy as 85 %

Code

startup<-read.csv("C:/Users/smukhe26/Desktop/project/DSAssignments/Neural Networks/50\_Startups.csv")

startup

DataFrameYouWant <- data.frame(startup$State)

state <- lapply(DataFrameYouWant, as.integer)

state

st<-cbind(startup[-4],state)

st

normalize<-function(x) { return ( (x-min(x))/(max(x)-min(x))) }

st\_norm<-as.data.frame(lapply(st,FUN=normalize))

summary(st\_norm)

startup\_train<-st\_norm[1:30,]

startup\_test<-st\_norm[31:50,]

fit<-neuralnet(Profit ~ R.D.Spend+Administration+Marketing.Spend+startup.State, data=startup\_train)

str(fit)

plot(fit)

fit\_results<-compute(fit,startup\_test[,1:4])

predicted\_profit<-fit\_results$net.result

cor(predicted\_profit,startup\_test$Profit)

fit<-neuralnet(Profit ~ R.D.Spend+Administration+Marketing.Spend+startup.State, data=startup\_train,hidden=3)

plot(fit)

fit\_results<-compute(fit,startup\_test[,1:4])

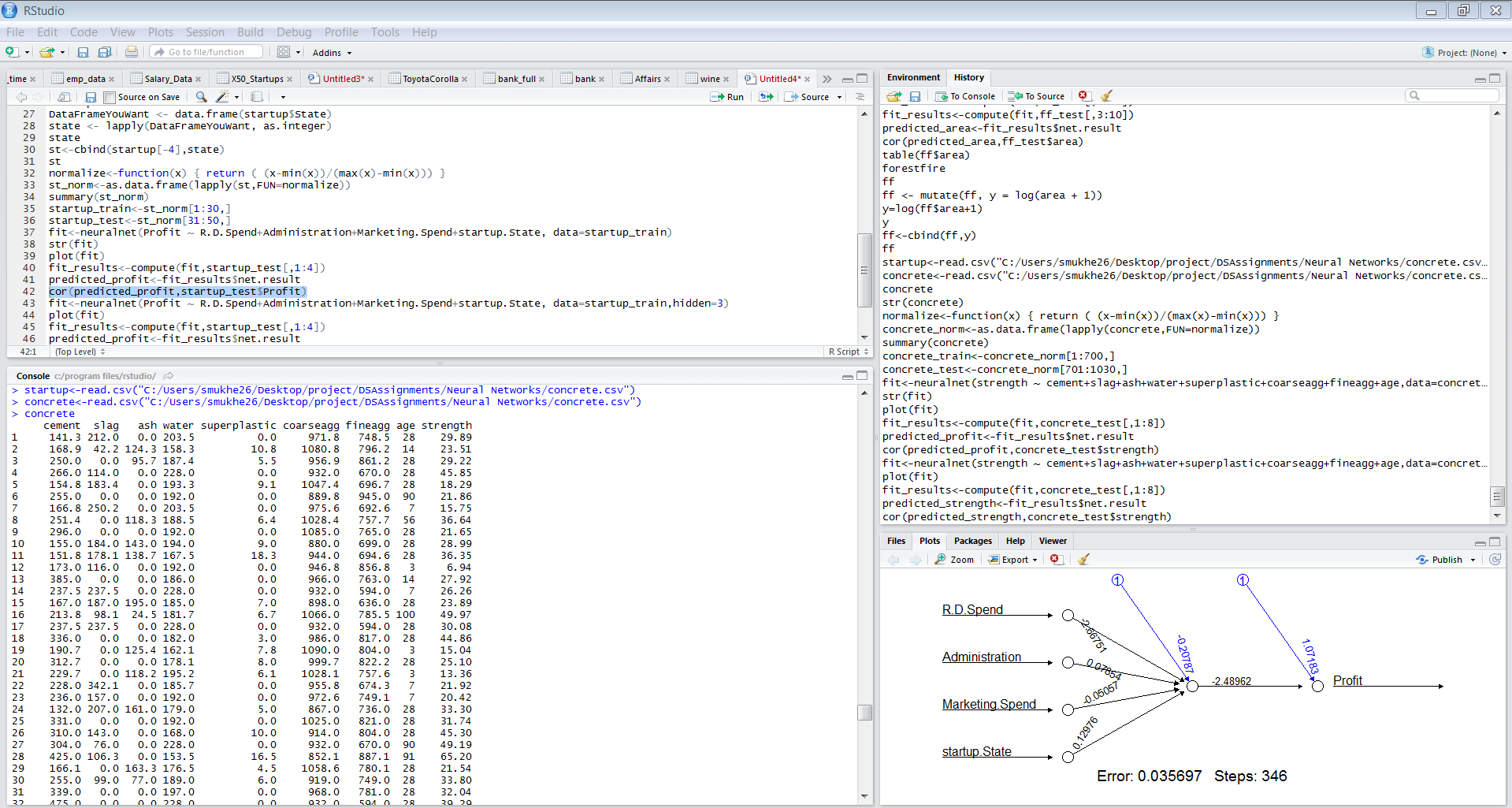
predicted\_profit<-fit\_results$net.result

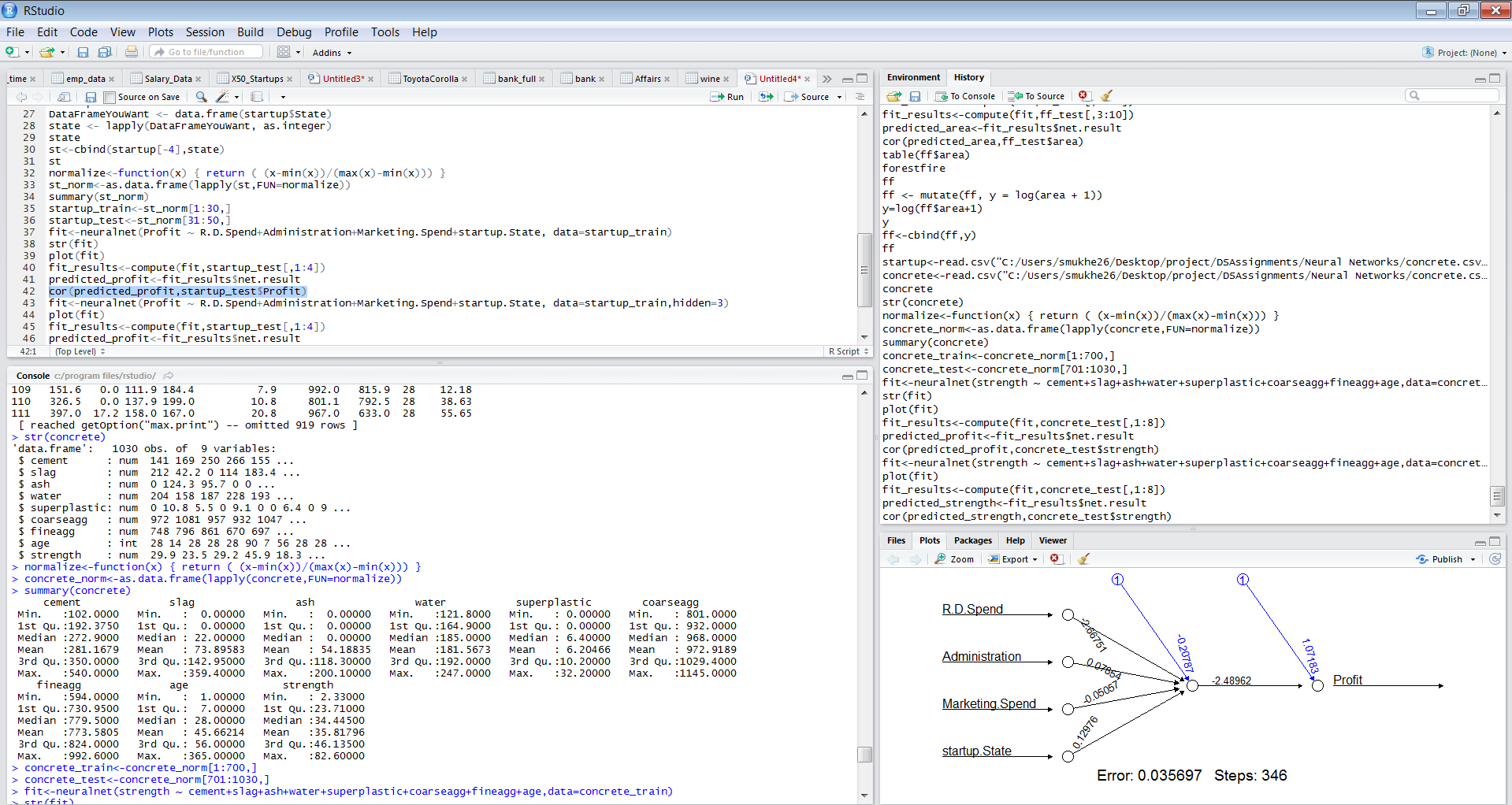
cor(predicted\_profit,startup\_test$Profit)

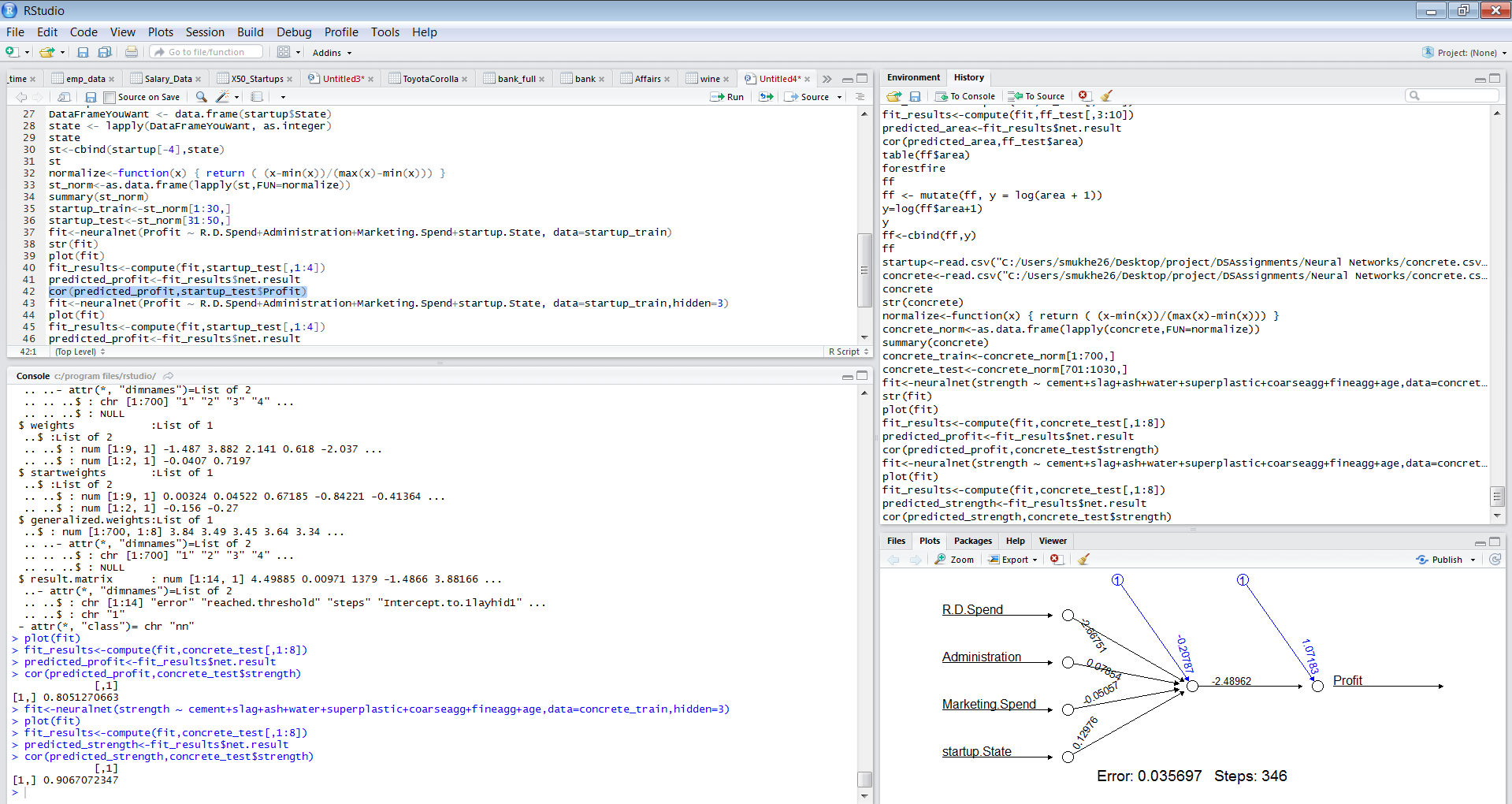
Predicting forest fire area

Concrete strength prediction using neural networks

Let us read the data concrete.csv into a data frame







We get a 90 % accuracy after putting hidden=3

Code

concrete<-read.csv("C:/Users/smukhe26/Desktop/project/DSAssignments/Neural Networks/concrete.csv")

concrete

str(concrete)

normalize<-function(x) { return ( (x-min(x))/(max(x)-min(x))) }

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

summary(concrete)

concrete\_train<-concrete\_norm[1:700,]

concrete\_test<-concrete\_norm[701:1030,]

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

str(fit)

plot(fit)

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

predicted\_profit<-fit\_results$net.result

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

fit<-neuralnet(strength ~ cement+slag+ash+water+superplastic+coarseagg+fineagg+age,data=concrete\_train,hidden=3)

plot(fit)

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

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

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