https://lh3.googleusercontent.com/IBpaDiJXgkknRuspFuREqaK1-sjLqcB9u5jWxjTIt0zTchlfLEmwSGSvVCkkyx5zr6YjJO1pk_E1Ysy_cU1c_vtQK_yQOW2-sKIF1Vd2LTAlwOPHwvWvsE1_NLLH1-VtmFMGp2H79-sKA-U8OQ

**TIJUANA INSTITUTE OF TECHNOLOGY**

**ACADEMIC**

**DEPARTMENT OF SYSTEMS AND COMPUTATION COMPUTER**

**SYSTEMS ENGINEERING**

**SEMESTER**

FEBRUARY- JULY 2022

**SUBJECT**

BDD-1703SC9C Data Mining

**Activity**

Practice 1

**Teacher**

MC JOSE CHRISTIAN ROMERO HERNANDEZ

**Students**

17212327 -Alderete Delgado Angel Esteban

17210659 - Villegas Carmona Damaris

Tijuana, BC May 17, 2022

Here we start with the loading of the csv file that will be used for the data analysis of this practice, once loaded we proceed to convert the states to categorical data in this case numbers,

then we divide the dataframe in two with a random seed so that the data is distributed randomly

getwd()

setwd("/home/chris/Documents/itt/Enero\_Junio\_2020/Mineria\_de\_datos/DataMining/Mach ineLearning/MultipleLinearRegression")

getwd()

# Importing the dataset

dataset <- read.csv('50\_Startups.csv')

# Encoding categorical data

dataset$State = factor(dataset$State,

levels = c('New York', 'California', 'Florida'), labels = c(1,2,3))

dataset

# Splitting the dataset into the Training set and Test set # Install.packages('caTools')

library(caTools) set.seed(123)

split <- sample.split(dataset$Profit, SplitRatio = 0.8) training\_set <- subset(dataset, split == TRUE)

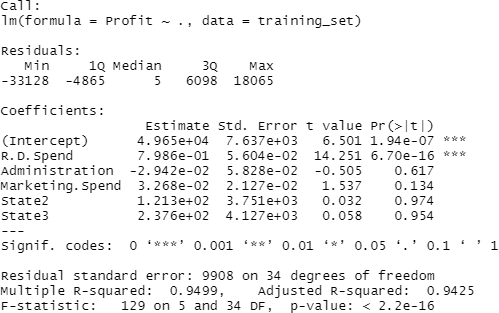
test\_set <- subset(dataset, split == FALSE)

# Fitting Multiple Linear Regression to the Training set

#regressor = lm(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend + State)

regressor = lm(formula = Profit ~ .,

data = training\_set ) summary(regressor)

Finally, the results of the model that through regression

Here they show us the predictions that each of the fields of the dataframe would have



# Prediction the Test set results

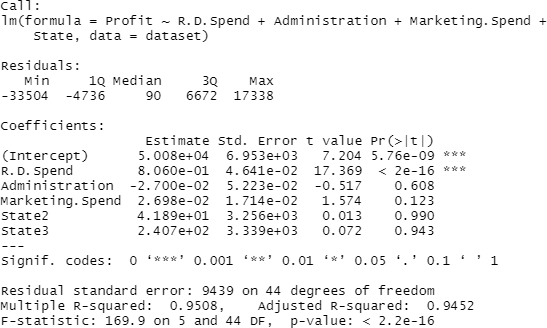
y\_pred = predict(regressor, newdata = test\_set) y\_pred

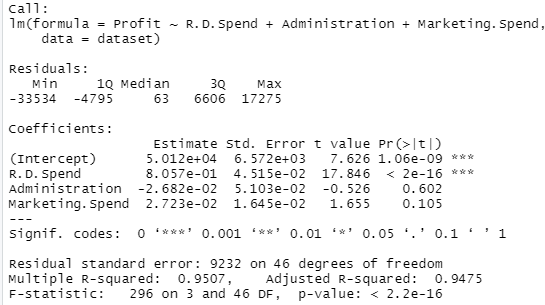
In this section we start to optimize the dataframe for the use of backward elimanation by reducing the dataframe to just a few key dataframe fields that will be used for this analysis.

# Assigment: visualize the siple liner regression model with R.D.Spend # Building the optimal model using Backward Elimination

regressor = lm(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend + State,

data = dataset ) summary(regressor)





regressor = lm(formula = Profit ~ R.D.Spend + Administration + Marketing.Spend, data = dataset )

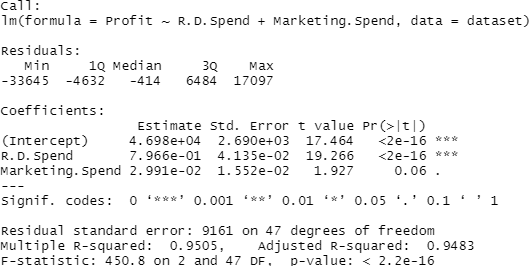
summary(regressor)

regressor = lm(formula = Profit ~ R.D.Spend + Marketing.Spend, data = dataset )

summary(regressor)

regressor = lm(formula = Profit ~ R.D.Spend + Marketing.Spend, data = dataset )

summary(regressor)





y\_pred = predict(regressor, newdata = test\_set) y\_pred

Backwardelimination

Once the dataframe has been reduced, we proceed to use the backwardelimination function. we create the function to be performed in the dataframe that we reduce

# Homework analise the follow atomation backwardElimination function backwardElimination <- function(x, sl) {

numVars = length(x)

for (i in c(1:numVars)){

regressor = lm(formula = Profit ~ ., data = x)

maxVar = max(coef(summary(regressor))[c(2:numVars), "Pr(>|t|)"]) if (maxVar > sl){

j = which(coef(summary(regressor))[c(2:numVars), "Pr(>|t|)"] == maxVar) x = x[, -j]

}

numVars = numVars - 1

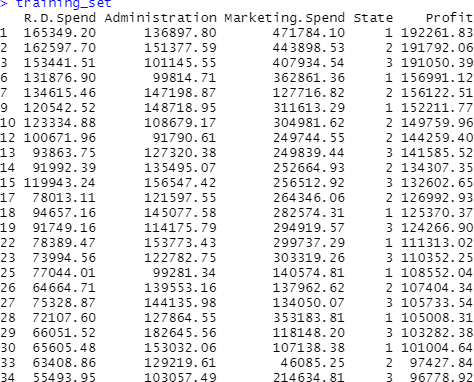
}

return(summary(regressor))

}

SL = 0.05

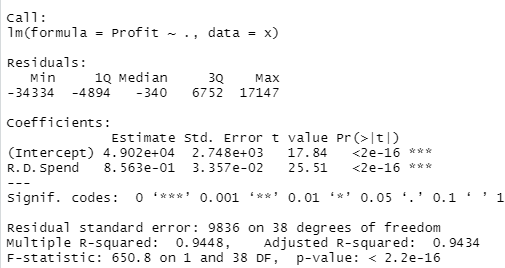
#dataset = dataset[, c(1,2,3,4,5)] training\_set



backwardelimination results

Here we are shown the results as the previous analysis but with the function of the analysis of

backwardeliminatio and it will show us a series of statistical results such as the mean, the median, the quartile, the standard error among others that it shows me



backwardElimination(training\_set, SL)