**LINEAR REGRESSION**

regressiondata = read.csv(file.choose(), header=TRUE)

regressiondata

model = lm (TMAX~RAIN, data = regressiondata)

summary(model)

model$residuals

mean(model$residuals^2)

sqrt(mean(model$residuals^2))

plot(model)

**MULTIPLE LINEAR REGRESSION**

regressiondata = read.csv(file.choose(), header=TRUE)

regressiondata

multi.fit =lm (RAIN ~ PRCP + TMAX + TMIN , data = regressiondata)

summary(multi.fit)

multi.fit$residuals

mean(multi.fit$residuals^2)

sqrt(mean(multi.fit$residuals^2))

plot(multi.fit)

**LOGISTIC REGRESSION**

input <- RD[,c("DATE","PRCP","TMAX","TMIN")]

print(head(input))

input <- RD[,c("DATE","PRCP","TMAX","TMIN")]

DATE.data = glm(formula = DATE ~ PRCP + TMAX + TMIN, data = input, family = binomial)

print(summary(DATE.data))

DATE.data$residuals

mean(DATE.data$residuals^2)

sqrt(mean(DATE.data$residuals^2))

**SUPPORT VECTOR MACHINE**

library(e1071)

plot(RD)

plot(RD$DATE, RD$PRCP, col=RD$RAIN)

plot(RD$TMAX, RD$TMIN, col=RD$RAIN)

RD

s<-sample(150,100)

col<- c("TMAX", "TMIN", "RAIN")

RD\_train <- RD[s,col]

RD\_test <- RD[-s,col]

svmfit <- svm(RAIN ~., data = RD\_train, kernal = "linear", cost = .1, scale = FALSE)

print(svmfit)

plot(svmfit, RD\_train[,col])

tuned <- tune(svm, RAIN~., data = RD\_train, kernel = "linear", ranges=

list(cost=c(0.001,0.01,.1,.1,10,100)))

summary(tuned)

p<-predict(svmfit, RD\_test[,col], type="class")

plot(p)

table(p,RD\_test[,3])

mean(p== RD\_test[,3])

svmfit$residuals

mean(svmfit$residuals^2)

sqrt(mean(svmfit$residuals^2))

chisq.test(svmfit$coefs^2)

**NAIVE BAYES**

library(naivebayes)

library(dbplyr)

library(ggplot2)

library(psych)

data <- read.csv(file.choose(), header = T)

str(data)

xtabs(~TMAX+RAIN, data = data)

data$RAIN <- as.factor(data$RAIN)

data$TMAX <- as.factor(data$TMAX)

#Data Partition

set.seed(1234)

ind <- sample(2,nrow(data), replace = T, prob = c(0.8,0.2))

train <- data[ind == 1,]

test <- data[ind == 2,]

#Naive Bayes Model

model <- naive\_bayes(RAIN ~., data = train)

model

plot(model)

#Predict

p <- predict(model, train, type = 'prob')

head(cbind(p, train))

model$prior

mean(model$prior^2)

sqrt(mean(model$prior^2))

chisq.test(model$prior^2)