**Practical 1 : Practical of Data collection, Data curation and management for Unstructured data(NoSQL) --> JSON**

# Give the input file name to the function.

#Data frames are tabular data objects.

# Create the data frame.

BMI <- data.frame(

gender = c("Male", "Male","Female"),

height = c(152, 171.5, 165),

weight = c(81,93, 78),

Age = c(42,38,26)

)

print(BMI)

# Load the package required to read JSON files.

library("rjson")

result <- fromJSON(file = "marks.json")

#result <- fromJSON(file = file.choose())

# Print the result.

print(result)

jsondata <- as.data.frame(result)

print(jsondata)

#structure of data frame

str(jsondata)

#statistical summary and nature of data

summary(jsondata)

#extract few columns

result <- data.frame(jsondata$ATTD ,jsondata$DEPT )

print(result)

# row selection

jsondata[1:5,]

jsondata[6:10,]

jsondata[c(1,3,5),]

jsondata[c(1,3,5,6:9),]

jsondata[c(-1,-3,-5),]

# column selection

jsondata[1:5,1:2]

jsondata[6:10,3:4]

jsondata[c(1,3,5),c(1,3)]

jsondata[c(1,3,5,6:9),c(1,3,4)]

jsondata[c(-1,-3,-5),c(-2)]

#subset selection

res <-subset(jsondata, DEPT=="CS")

print(res)

tempdata <- subset(jsondata, DEPT=="CS" & PERCENT>50)

subset(tempdata, select = c(-DEPT))

subset(tempdata, select = c(PERCENT,ROLL))

tempdata <- subset(jsondata, !(DEPT=="CS") & PERCENT>50)

print(tempdata)

mean(jsondata$ATTD)

mean(jsondata$PERCENT)

aggregate(PERCENT~DEPT, data=jsondata, FUN=mean)

aggregate(PERCENT~DEPT+SEM, data=jsondata, FUN=mean)

aggregate(cbind(PERCENT,ATTD)~DEPT+SEM, data=jsondata, FUN=mean)

fr <-table(jsondata$DEPT, jsondata$SEM)

prop.table(fr)

boxplot(jsondata$ATTD)

boxplot(jsondata$PERCENT)

boxplot(ATTD~DEPT, data=jsondata)

boxplot(ATTD~DEPT, data=jsondata, col=c("red","green", "blue"))

hist(jsondata$ATTD)

sampledata <- data.frame(

gender = c("Male", "Male","Female", "Female", "Male", "Male"),

loc = c("Mumbai", "Delhi","Pune", "Pune", "Delhi", "Pune"),

empname = c("a","b","c","d","e","f")

)

sampledata

sampledata[order(sampledata$loc),]

library(dplyr)

jsondata[ order(jsondata$DEPT, desc(jsondata$SEM)), ]

**#to check whether any columns have NA values**

**result2 <- fromJSON(file = "marks.json")**

**result2**

**mydata <- as.data.frame(result2)**

**mydata[2,2] <- NA**

**mydata[6,3] <- NA**

**colSums(is.na(mydata))**

**colSums(is.na(mydata))**

**nmiss<-sum(is.na(mydata$ATTD))**

**nmiss**

**Pract 2 : MongoDB**

**Pract 8 : Hypothesis Testing**

**Step 1**

To check whether the data follows normal distribution

"test for normal destribution"

data1<-read.csv("…….csv",sep=",",header = T)

shapiro.test(data1$colname)

# p value should be greater than 0.05

#"one sample t test"

#One sample t test is used to test the hypothesis about a single population mean.

apple<-read.csv(file.choose(),sep=",",header = T)

summary(apple)

t.test(apple$C1, alternative="greater", mu=97)

#"independent t test "

#The independent-samples t-test compares the means of two independent groups

#h0 = same, h1 = not same

time<-read.csv(file.choose(),sep=",",header = T)

summary(time)

t.test(videodata$TotalCOld,videodata$TotalDDEMO, alternative="two.sided", var.equal=TRUE)

#"paired t test " Dependent samples

time1<-read.csv(file.choose(),sep=",",header = T)

t.test(time1$time\_before,time1$time\_after,alternative = "greater",paired = T)

"t test for correlation " # h0 - no correlation, h1 - there is correlation

cor<-read.csv(file.choose(),sep=",",header = T)

summary(cor)

cor.test(cor$aptitude,cor$job\_prof,alternative = "two.sided",method="pearson")

"t test for variance "

var<-read.csv(file.choose(),sep=",",header = T)

summary(var)

var.test(var$time\_g1,var$time\_g2,alternative = "two.sided")

apple<-read.csv(file.choose(),sep=",",header = T)

summary(apple)

t.test(apple$C1, alternative="greater", mu=100)

**Pract 9 : Anova testing(normal distribution of data)**

# h0 = two population variances are equal, h1 = not equal

ftest<-read.csv(file.choose(),sep=",",header=T)

var.test(ftest$time\_g1,ftest$time\_g2,alternative = "two.sided")

"one way anova" same as t test, when k=2, h0 = same, h1 = not same

data1<-read.csv(file.choose(),sep = ",",header = T)

names(data1)

summary(data1)

head(data1)

anv<-aov(formula = satindex~dept,data=data1)

summary(anv)

#"two way anova" h0 = same, h1 = not same

#Two Way Anova is used when there are 2 factors under study.

data2<-read.csv(file.choose(),sep=",",header = T)

names(data2)

summary(data2)

anv1<-aov(formula = satindex~ dept+exp+dept\*exp,data = data2)

summary(anv1)

**Pract 6 - Simple/Multiple Regression**

# Employee Data - Summary

empdata<-read.csv("empdata.csv",sep=",",header = T)

summary(empdata)

names(empdata)

# Relation among predictors

pairs(~index+written+language+tech+gk,data = empdata)

#Build linear model

#empmodel<-lm(index~.,data = empdata) #empid also gets included

empmodel<-lm(index~written+language+tech+gk,data = empdata)

#global testing to check whether

#at least one variable has significant impact

#Check p value, it should be less than 0.05

#R squared values should be closer to 100%

#significant variables are marked with \*

summary(empmodel)

plot(empmodel)

#no relation

empmodeltemp<-lm(empid~written+language+tech+gk,data = empdata)

summary(empmodeltemp)

#to get the values of response variable

empdata$pred<-fitted(empmodel)

head(empdata)

#to get the values of resuduals

empdata$res<-residuals(empmodel)

head(empdata)

"to apply the model on new data..."

newempdata<-read.csv("newempdata.csv",header=T)

newempdata$pred<-predict(empmodel,newempdata)

newempdata

"Shapiro test of residual, expected p value > 0.05"

shapiro.test(empdata$res)

"to check the multicolinearity all VIFs should be less than 5"

library(car)

vif(empmodel)

#take any two columns, add some random number and include them in data

empdata2<-read.csv("testempdata.csv",sep=",",header = T)

empmodel3<-lm(index~.,data = empdata2)

summary(empmodel3)

vif(empmodel3)

"plot must be random indicates no heteroscedasticity"

plot(empdata$pred,empdata$res,col="red")

"QQ Plot"

qqnorm(empdata$res,col="blue")

qqline(empdata$res,col="blue")

"detecting heteroscedasticity using ncvtest"

"p value should be greater than 0.05 to indicate homoscedasticity"

library(car)

ncvTest(empmodel3,~written+language+tech+ gk)

#finding influential observations

library(car)

influ<-influence.measures(empmodel)

influ

#influencePlot(empmodel,id.method="identify",main="Influence Plot",sub="Circle size is proportial to Cook's Distance")

influencePlot(empmodel, main="Influence Plot",sub="Circle size is proportial to Cook's Distance")

#remove influential observation

empdatatemp <-empdata[-(33),]

empmodeltemp<-lm(index~written+language+tech+gk,data = empdatatemp)

influ<-influence.measures(empmodeltemp)

influ

influencePlot(empmodeltemp, main="Influence Plot",sub="Circle size is proportial to Cook's Distance")

#hold out validation in R

empdata

library(caret)

emphold<-createDataPartition(empdata$index,p=0.8,list=FALSE)

head(emphold)

dim(emphold)

traindata <- empdata[emphold,]

testdata <-empdata[-emphold,]

traindata

testdata

empmodel<-lm(index~written+language+tech+gk,data = traindata)

traindata$res<-residuals(empmodel)

head(traindata)

RMSEtrain<-sqrt(mean(traindata$res\*\*2))

RMSEtrain

testdata$pred<-predict(empmodel,testdata)

testdata$res<-(testdata$index-testdata$pred)

RMSEtest<-sqrt(mean(testdata$res\*\*2))

RMSEtest

#there should not be much difference in RMSEtrain & RMSEtest

# k fold cross validation

library(caret)

kfolds<-trainControl(method="cv",number=4)

model<- train(index~written+language+tech+gk,data=empdata,method="lm", trControl=kfolds)

model

# repeated k fold cross validation

library(caret)

kfolds<-trainControl(method="cv",number=4, repeats=5)

model<- train(index~written+language+tech+gk,data=empdata,method="lm", trControl=kfolds)

model

#LOOCV

kfolds<-trainControl(method="LOOCV")

model<- train(index~written+language+tech+gk,data=empdata,method="lm", trControl=kfolds)

model

# stepwise regression

null<-lm(index~1,data=empdata)

summary(null)

full<-lm(index~written+language+tech+gk,data=empdata)

summary(full)

step(null,scope=list(lower=null,upper=full),direction="forward")

step(full ,scope=list(lower=null,upper=full),direction="backward")

step(full ,scope=list(lower=null,upper=full),direction="both")

# logistic regression, bankloan file contains 600 records

bankloan<-read.csv("bankloan.csv",header=T)

head(bankloan)

str(bankloan)

# Example of changing data formats

# age is an integer, but it should be considerred as factor/category

# we need to do the conversion

bankloan$AGE<-factor(bankloan$AGE)

str(bankloan)

**Pract 7 - Logistic Regression**

#glm - generalized linear model

riskmodel<- glm(DEFAULTER~AGE+EMPLOY+ADDRESS+DEBTINC+CREDDEBT+OTHDEBT,family=binomial,data=bankloan)

summary(riskmodel)

model.matrix(riskmodel)

null<-glm(DEFAULTER ~ 1, family=binomial,data=bankloan)

anova(null,riskmodel, test="Chisq")

#here h0 is rejected, means at least one variable is significant

bankloan$predprob<-round(fitted(riskmodel),2)

head(bankloan)

library(gmodels)

CrossTable(bankloan$DEFAULTER,fitted(riskmodel)>0.5)

head(bankloan)

table (bankloan$DEFAULTER,fitted(riskmodel)>0.5)

library(ROCR)

bankloan$predprob<-fitted(riskmodel)

pred<-prediction(bankloan$predprob,bankloan$DEFAULTER)

perf<-performance(pred,"tpr","fpr")

plot(perf)

abline(0,1)

auc<-performance(pred,"auc")

auc@y.values

# Perfect model if C=1 Worthless if C=0.5

riskmodel<- glm(DEFAULTER~EMPLOY+ADDRESS+DEBTINC+CREDDEBT,family=binomial,data=bankloan)

library(ResourceSelection)

#function hoslem.test requires observed Y and predicted probs

hltest<-hoslem.test(bankloan$DEFAULTER,fitted(riskmodel),g=10)

hltest

confusionMatrix(traindata$predY,traindata$DEFAULTER,positive="1")

bankloan

heads(bankloan)

riskmodel<-glm(DEFAULTER~EMPLOY+ADDRESS+DEBTINC+CREDDEBT,family=binomial,data=bankloan)

summary(riskmodel)

index<-createDataPartition(bankloan$DEFAULTER,p=0.7,list=FALSE)

traindata<-bankloan[index,]

traindata$predprob<-predict(riskmodel,traindata,type='response')

traindata$predY<-ifelse(traindata$predprob>0.30,1,0)

confusionMatrix(traindata$predY,traindata$DEFAULTER,positive="1")

**Pract 4 - clustering**

bankloan<-read.csv("bankloan.csv",header=T)

bankloanca<-subset(bankloan,select=c(-AGE,-SN))

CL<-kmeans(bankloanca,centers = 3, nstart = 3)

CL

CL$centers

head(CL$cluster)

library(tidyverse) # data manipulation

library(cluster) # clustering algorithms

library(factoextra) # clustering algorithms & visualization

fviz\_cluster(CL, data = bankloan)

##

library(NbClust)

nb <- NbClust(bankloan, diss=NULL, distance = "euclidean", min.nc=2, max.nc=15, method = "kmeans",index = "all", alphaBeale = 0.1)

#nb <- NbClust(bankloan, min.nc=2, max.nc=5, method = "kmeans",index = "all")

nb$All.index

nb$All.CriticalValues

nb$Best.nc

CL<-kmeans(bankloanca,centers = 3, nstart = 1)

CL

fviz\_cluster(CL, data = bankloan)

**Pract 3 - PCA**

data("iris")

head(iris)

summary(iris)

library()

"to find principal component"

mypr<-prcomp(iris[,-5],scale=T)

"to understand use of scale"

plot(iris$Sepal.Length,iris$Sepal.Width)

plot(scale(iris$Sepal.Length),scale(iris$Sepal.Width))

mypr

summary(mypr)

plot(mypr,type="l")

biplot(mypr,scale=0)

"extract pc scores"

str(mypr)

mypr$x

iris2<-cbind(iris,mypr$x[,1:2])

head(iris2)

cor(iris[,-5],iris2[,6:7])

install.packages("pls")

library(pls)

names(iris)

pcmodel<-pcr(Sepal.Length~Species+Sepal.Width+Petal.Length+Petal.Width,ncomp=3,data=iris,scale=T)

summary(pcmodel)

iris$pred<-predict(pcmodel,iris,ncomp = 2)

head(iris)

#==

dim(mtcars)

apply(mtcars, 2, mean) #2 means apply on columns

apply(mtcars, 2, var)

carmodel <- prcomp(mtcars, scale=T)

carmodel

summary(carmodel)

names(carmodel)

plot(carmodel,type="l")

biplot(carmodel,scale = 0, cex=0.65)

# music choice

newgendata <- read.csv("studdata2.csv", header = T, sep = ",")

newgenmodel <- prcomp(newgendata, scale=T)

summary(newgenmodel)

biplot(newgenmodel,scale = 0, cex=0.65)

plot(newgenmodel,type="l")

**Pract 10 - Decision Tree**

install.packages("partykit")

install.packages("CHAID",repos = "http://R-Forge.R-project.org",type="source")

library(CHAID)

library(partykit)

mydata<-data.frame(iris)

model2<-ctree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,data=mydata)

plot(model2)

str(iris)

**model3 <-predict(model2,iris) # Prediction**

model2<-chaid(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,data=mydata)

mydata$Sepal.Length=factor(mydata$Sepal.Length)

mydata$Sepal.Width=factor(mydata$Sepal.Width)

mydata$Petal.Length=factor(mydata$Petal.Length)

mydata$Petal.Width=factor(mydata$Petal.Width)

plot(model2)

mtcars

mydata<-data.frame(mtcars)

amodel<-ctree(mpg ~ cyl+disp+hp+wt+gear,data=mydata)

plot(amodel)

mydata$cyl=factor(mydata$cyl)

mydata$disp=factor(mydata$disp)

mydata$hp=factor(mydata$hp)

mydata$wt=factor(mydata$wt)

mydata$gear=factor(mydata$gear)

mydata$mpg = factor(mydata$mpg)

bmodel<-chaid(mpg ~ cyl+disp+hp+wt+gear,data=mydata)

plot(bmodel)

#Classfication Tree to predict class of flower

library("rpart")

png(file="tree5.png")

mydata<-data.frame(iris)

amodel<-rpart(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,data=mydata, method="class")

plot(amodel)

text(amodel)

dev.off()

#Classfication Tree to predict defaulter status

png(file="tree6.png")

mydata<-read.csv("bankloan.csv", header = T, sep = ",")

head(mydata)

amodel<-rpart(DEFAULTER~AGE+EMPLOY+ADDRESS+DEBTINC+CREDDEBT+OTHDEBT,data=mydata, method="class")

plot(amodel)

text(amodel)

dev.off()

#Regression Tree to predict mileage

library("rpart")

png(file="tree3.png")

mydata<-data.frame(mtcars)

amodel<-rpart(mpg ~ cyl+disp+hp+wt+gear,data=mydata, method="anova" )

plot(amodel)

text(amodel)

dev.off()

#regression tree to predict index

library("rpart")

mydata<-read.csv("empdata.csv", header = T, sep = ",")

head(mydata)

amodel<-rpart(index ~ tech+written+language+gk ,data=mydata, method="anova" )

png(file="tree4.png")

plot(amodel)

text(amodel)

dev.off()

**Pract 5 - time series forecasting**

x <- ts (1:20, frequency = 4, start = c(1959,1)) # frequency 4 => Quarterly Data

x <-ts (1:12, frequency = 12, start = c(1959,2)) # freq 12 => Monthly data.

ts (1:20, frequency = 1, start = c(1959), end=c(1962))# Freq 1 - yearly data

plot(x)

#==

library(tseries)

adf.test(AirPassengers) # Augmented Dickey-Fuller Test p-value < 0.05 indicates the TS is stationary

data(AirPassengers) # loads a dataset

class(AirPassengers) # identifies class

#class(mtcars)

start(AirPassengers)

end(AirPassengers)

frequency(AirPassengers)

summary(AirPassengers)

plot(AirPassengers) # plots year VS number of passangers

abline(reg=lm(AirPassengers~time(AirPassengers))) # fits line

cycle(AirPassengers) #This will print the cycle across years.

#This will aggregate the cycles and display a year on year trend

plot(aggregate(AirPassengers,FUN=mean))

#Box plot across months will give us a sense on seasonal effect

boxplot(AirPassengers~cycle(AirPassengers))

#Imp Deductions

#1. The year on year trend clearly shows increase in number of passangers

#2. The variance and the mean value in July and August is much higher than other months

#3. Variation is small otherwise, there is a seasonal effect

#https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/

acf(log(AirPassengers))

acf(diff(log(AirPassengers)))

(fit <- arima(log(AirPassengers), c(0, 1,1),seasonal = list(order = c(0, 1, 1), period = 12)))

pred <- predict(fit, n.ahead = 10\*12)

summary(pred)

plot(pred)

pred

ts.plot(AirPassengers,2.718^pred$pred, log = "y", lty = c(1,3))