**Practical 1**

**Aim: Data collection, Data curation and management for Unstructured data (NoSQL) with Couch DB**

install.Packages("sofa")

library("sofa")

x<-Cushion$new()

x$ping()

db\_create(x,dbname = 'ds')

db\_list(x)

doc1<-'{"rollno":"01","name":"ABC","GRADE":"A"}'

doc\_create(x,doc1,dbname = "ds",docid = "a\_1")

doc2<-'{"rollno":"02","name":"PQR","GRADE":"A"}'

doc\_create(x,doc2,dbname = "ds",docid = "a\_2")

doc3<-'{"rollno":"03","name":"xyz","GRADE":"B","REMARK":"PASS"}'

doc\_create(x,doc3,dbname = "ds",docid = "a\_3")

db\_changes(x,"ds")

db\_query(x,dbname = "ds",

selector = list('\_id'=list('$gt'=NULL)))$docs

db\_query(x,dbname = "ds",selector = list(GRADE="A"))$docs

db\_query(x,dbname="ds",selector=list(REMARK="PASS"))$docs

db\_query(x,dbname = "ds",selector = list(rollno=list('$gt'='02')),fields=c("name","GRADE"))$docs

library("jsonlite")

res<-db\_query(x,dbname = "ds",selector = list('\_id'=list('$gt'=NULL)),fields=c("name","rollno","GRADE","REMARK"),as="json")

fromJSON(res)$docs

doc\_delete(x,dbname = "ds",docid = "a\_2")

doc\_get(x,dbname = "ds",docid = "a\_2")

doc2<-'{"name":"Sfood","biryani":"TEST","note":"yummy","note2":"yay"}'

doc\_update(x,dbname = "ds",doc=doc2,docid="a\_3",rev = "3-b1fb56db955b142c6efd3b3c52fe9e1b")

doc3<-'{"rollno":"01",

"name":"UZMA",

"GRADE":"A"}'

doc\_update(x,dbname = "ds",doc=doc3,docid = "a\_1",rev = "1-be7c98bddf8ea7c46f4f401ff387593d")

**Practical 3**

**Aim: Practical of Principal Component Analysis(PCA).**

data\_iris <- iris[1:4]

Cov\_data <- cov(data\_iris )

Eigen\_data <- eigen(Cov\_data)

PCA\_data <- princomp(data\_iris ,cor="False")

Eigen\_data$values

PCA\_data$sdev^2

PCA\_data$loadings[,1:4]

Eigen\_data$vectors

summary(PCA\_data)

biplot (PCA\_data)

screeplot(PCA\_data, type="lines")

model2 = PCA\_data$loadings[,1]

model2\_scores <- as.matrix(data\_iris) %\*% model2

library(class)

install.packages("e1071")

library(e1071)

mod1<-naiveBayes(iris[,1:4], iris[,5])

mod2<-naiveBayes(model2\_scores, iris[,5])

table(predict(mod1, iris[,1:4]), iris[,5])

table(predict(mod2, model2\_scores), iris[,5])

**Practical4:**

**Exp4 Practical of Clustering.**

library(ggplot2)

scatter <- ggplot(data=iris, aes(x = Sepal.Length, y = Sepal.Width))

scatter + geom\_point(aes(color=Species, shape=Species)) + theme\_bw()+ xlab("Sepal Length") + ylab("Sepal Width") + ggtitle("Sepal Length-Width")

ggplot(data=iris, aes(Sepal.Length, fill = Species))+ theme\_bw()+ geom\_density(alpha=0.25)+ labs(x = "Sepal.Length", title="Species vs Sepal Length"))

vol <- ggplot(data=iris, aes(x = Sepal.Length))

vol + stat\_density(aes(ymax = ..density.., ymin = -..density..,fill = Species, color = Species),geom = "ribbon", position = "identity") + facet\_grid(. ~ Species) + coord\_flip() + theme\_bw()+labs(x = "Sepal Length", title="Species vs Sepal Length")

vol <- ggplot(data=iris, aes(x = Sepal.Width))

vol + stat\_density(aes(ymax = ..density.., ymin = -..density..,fill = Species,color = Species), geom = "ribbon", position = "identity") + facet\_grid(. ~ Species) + coord\_flip() + theme\_bw()+labs(x = "Sepal Width", title="Species vs Sepal Width")

irisData <- iris[,1:4]

totalwSS<-c()

for (i in 1:15)

{

clusterIRIS <- kmeans(irisData, centers=i)

totalwSS[i]<-clusterIRIS$tot.withinss

}

plot(x=1:15,y=totalwSS,type="b",xlab="Number of Clusters",

ylab="Within groups sum-of-squares")

install.packages("NbClust")

library(NbClust)

par(mar = c(2,2,2,2))

nb <- NbClust(irisData, method = "kmeans")

hist(nb$Best.nc[1,], breaks = 15, main="Histogram for Number of Clusters")

library(cluster)

cl <- kmeans(iris[,-5], 2)

dis <- dist(iris[,-5])^2

sil = silhouette (cl$cluster, dis)

plot(sil, main = "Clustering Data with Silhoutte plot using 2 Clusters", col = c("cyan", "blue"))

library(cluster)

cl <- kmeans(iris[,-5], 8)

dis <- dist(iris[,-5])^2

sil = silhouette (cl$cluster, dis)

plot(sil, main = "Clustering Data with Silhoutte plot using 8 Clusters", col = c("cyan", "blue", "orange", "yellow", "red", "gray", "green", "maroon"))

install.packages("factoextra")

library(factoextra)

install.packages("clustertend")

library(clustertend)

genx<-function(x){

runif(length(x), min(x), (max(x)))

}

random\_df <- apply(iris[,-5], 2, genx)

install.packages("factoextra")

library(factoextra)

install.packages("clustertend")

library(clustertend)

genx<-function(x){

runif(length(x), min(x), (max(x)))

}

random\_df <- apply(iris[,-5], 2, genx)graph = FALSE)

res$hopkins\_stat

**Practical 5 Time Series:**

Aim: Time series forcasting

data("AirPassengers")

class(AirPassengers)

start(AirPassengers)

end(AirPassengers)

frequency(AirPassengers)

summary(AirPassengers)

plot(AirPassengers)

abline(reg=lm(AirPassengers~time(AirPassengers)))

cycle(AirPassengers)

plot(aggregate(AirPassengers,FUN=mean))

boxplot(AirPassengers~cycle(AirPassengers))

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)

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

**#Practical No. 6**

#Practical of Simple/Multiple Linear Regression

**Simple:**

lsfit(iris$Petal.Length, iris$Petal.Width)$coefficients

plot(iris$Petal.Length, iris$Petal.Width, pch=21, bg=c("red","green3","blue")[unclass(iris$

Species)], main="Iris Data", xlab="Petal length", ylab="Petal width")

abline(lsfit(iris$Petal.Length, iris$Petal.Width)$coefficients, col="black")

#create smple regression model

lm(Petal.Width ~ Petal.Length, data=iris)$coefficients

plot(iris$Petal.Length, iris$Petal.Width, pch=21, bg=c("red","green3","blue")[unclass(iris$

Species)], main="Iris Data", xlab="Petal length", ylab="Petal width")

abline(lm(Petal.Width ~ Petal.Length, data=iris)$coefficients, col="black")

summary(lm(Petal.Width ~ Petal.Length, data=iris))

plot(iris$Sepal.Width, iris$Sepal.Length, pch=21, bg=c("red","green3","blue")[unclass(iris

$Species)], main="Iris Data", xlab="Sepal Width", ylab="Sepal Length")

abline(lm(Sepal.Length ~ Sepal.Width, data=iris)$coefficients, col="black")

summary(lm(Sepal.Length ~ Sepal.Width, data=iris))

**Multiple:**

#What happens if we divide the data up by species, and run three separate linear regressions?

plot(iris$Sepal.Width, iris$Sepal.Length, pch=21, bg=c("red","green3","blue")[unclass(iris

$Species)], main="Iris Data", xlab="Sepal length", ylab="Sepal length")

abline(lm(Sepal.Length ~ Sepal.Width, data=iris)$coefficients, col="black")

abline(lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="setosa"),])$coefficients, col="red")

abline(lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="versicolor"),])$coefficients, col="green3")

abline(lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="virginica"),])$coefficients, col="blue")

#The coefficients doing separate per species regressions of Sepal.Length ~ Sepal.Width are:

lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="setosa"),])$coefficients

lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="versicolor"),])$coefficients

lm(Sepal.Length ~ Sepal.Width, data=iris[which(iris$Species=="virginica"),])$coefficients

lm(Sepal.Length ~ Sepal.Width:Species + Species - 1, data=iris)$coefficients

#Using the summary command on the linear model object gives:

summary(lm(Sepal.Length ~ Sepal.Width:Species + Species - 1, data=iris))

#Simplify with AIC (Akaike Information Criterion)

summary(step(lm(Sepal.Length ~ Sepal.Width \* Species, data=iris)))

#I just introduced a model of the form Sepal.Length ~ Sepal.Width:Species + Species - 1,

#which gave identical coefficients to those found doing species specific regressions:

lm(Sepal.Length ~ Sepal.Width:Species + Species - 1, data=iris)$coefficients

lm(Sepal.Length ~ Sepal.Width:Species + Species, data=iris)$coefficients

**Practical 7**

**Aim: Practical of Logistics Regression.**

library(datasets)

ir\_data<- iris

head(ir\_data)

str(ir\_data)

levels(ir\_data$Species)

sum(is.na(ir\_data))

ir\_data<-ir\_data[1:100,]

set.seed(100)

samp<-sample(1:100,80)

ir\_test<-ir\_data[samp,]

ir\_ctrl<-ir\_data[-samp,]

install.packages("ggplot2")

library(ggplot2)

install.packages("GGally")

library(GGally)

ggpairs(ir\_test)

y<-ir\_test$Species; x<-ir\_test$Sepal.Length

glfit<-glm(y~x, family = 'binomial')

summary(glfit)

newdata<- data.frame(x=ir\_ctrl$Sepal.Length)

predicted\_val<-predict(glfit, newdata, type="response")

prediction<-data.frame(ir\_ctrl$Sepal.Length, ir\_ctrl$Species,predicted\_val)

prediction

qplot(prediction[,1], round(prediction[,3]), col=prediction[,2], xlab = 'Sepal Length', ylab = 'Prediction using Logistic Reg.')

**Practical 8**

**Aim: practical of Hypothesis**

x= c(6.2, 6.6, 7.1, 7.4, 7.6, 7.9, 8, 8.3, 8.4, 8.5, 8.6, 8.8, 8.8, 9.1, 9.2, 9.4, 9.4, 9.7, 9.9, 10.2, 10.4, 10.8,11.3, 11.9)

t.test(x-9,alternative="two.sided",conf.level=0.95)

x=c(418,421,421,422,425,427,431,434,437,439,446,447,448,453,454,463,465)

y=c(429,430,430,431,36,437,440,441,445,446,447)

test2<-t.test(x,y,alternative="two.sided",mu=0,var.equal=F,conf.level=0.95)

test2

**Practical 9**

**Aim: Practical of Analysis of Variance.**

y1 = c(18.2, 20.1, 17.6, 16.8, 18.8, 19.7, 19.1)

y2 = c(17.4, 18.7, 19.1, 16.4, 15.9, 18.4, 17.7)

y3 = c(15.2, 18.8, 17.7, 16.5, 15.9, 17.1, 16.7)

y = c(y1, y2, y3)

n = rep(7, 3)

n

group = rep(1:3, n)

group

tmp = tapply(y, group, stem)

stem(y)

tmpfn = function(x) c(sum = sum(x), mean = mean(x), var = var(x),n = length(x))

tapply(y, group, tmpfn)

tmpfn(y)

data = data.frame(y = y, group = factor(group))

fit = lm(y ~ group, data)

anova(fit)

df = anova(fit)[, "Df"]

names(df) = c("trt", "err")

df

alpha = c(0.05, 0.01)

qf(alpha, df["trt"], df["err"], lower.tail = FALSE)

anova(fit)["Residuals", "Sum Sq"]

anova(fit)["Residuals", "Sum Sq"]/qchisq(c(0.025, 0.975), 18,lower.tail = FALSE)