**First R class: 24/1/2017**

**Data= cardata 2005, folder name: PGA03abhi, Filename: PGA03abhi**

cardata=read.csv(file.choose()) #importing data

head(cardata) # specify first 6 rows of data

head(cardata,20) # specify first 20 rows of dataf

tail(cardata) # specify last 6 rows of data

tail(cardata,20) # specify last 20 rows of data

str(cardata) # display the internal structure of an r object (Here dataframe: cardata)

head(cardata$Price) # display price variable data under cardata dataframe

class(cardata) # outcome to this command will be: [1] "data.frame"

ncol(cardata) # display the total number of coloumns

summary(cardata) # summarizes the Min,Q1, Median, Mean, Q3 and Max data information of the cardata dataframe

summary(cardata$Mileage) # summarizes the Min,Q1, Median, Mean, Q3 and Max data information for only Mileage variable in the cardata dataframe

summary(cardata$Price) # summarizes similar information as above for price varaible

sd(cardata$Price) # return the standard deviation for for all the price variable values in cardata dataframe

hist(cardata$Price) # return a histogram plot for price variable

hist(cardata$Mileage) # return a histogram plot for mileage variable

boxplot(cardata$Price,horizontal = T) # return a horizontal boxplot for price variable

boxplot(cardata$Mileage, horizontal = T) # return a horizontal boxplot for mileage variable

plot(density(cardata$Price)) # return a density curve for price variable

plot(density(cardata$Mileage))

plot(density(log(cardata$Price))) # return a logrithmic transformed density curve for price variable

# this plot will be nearly unskewed with outliers removed to max possibility

# compare this with the regular density plot of price variable

cardata1=cardata[-c(3:6)] # return a dataframe cardata1 by removing column 3,4,5,6 from original dataframe: cardata

# this is done to remove categorical data availble in c3 to c6 here

# this is done to execute next two commands of fing correlation and covariance(these can only be done for numerical data)

cor(cardata1) #find correlation for all the variables in cardata1

cov(cardata1) #find covariance for all the variables in cardata1

plot(cardata$Price,cardata$Mileage) # return a scatter plot for price vs mileage variables

pairs(cardata1) # return an array of scatter plots considering all combinations for variables

table(cardata$Make) # return frequency count for all "Make" categories/variable

prop.table(table(cardata$Make))\*100 # return percentages share for all "Make" categories/variable

table(cardata$Cylinder) # return frequency count for all "cylinder" categories/variable

prop.table(table(cardata$Cylinder))\*100 # return percentages share for all "cylinder" categories/variable

table(cardata$Leather) # return frequency count for all "leather" categories/variable

table(cardata$Sound) # return frequency count for all "sound" categories/variable

barplot(table(cardata$Make)) # return a bar chart for all Make variable data

pie(table(cardata$Make)) # return a bar chart for all Make variable data

table(cardata$Make,cardata$Sound) # return the information on total number of cars among different Make(model or company) with and without Sound variable (Sound system)

barplot(table(cardata$Sound,cardata$Make)) # reutn a stack chart on above information

barplot(table(cardata$Sound,cardata$Make),legend.text = T) # reutn the same stack chart as above with legends defined

barplot(table(cardata$Sound,cardata$Make),legend.text = T,xlab = "Make", ylab = "Sound") # return the same stack chart as above with axes labels as Make and Sound

barplot(table(cardata$Sound,cardata$Make),legend.text = T,beside = T) # reutn the same chart with two individual bars for sound variable for each make type

buickcadillac=subset(cardata,cardata$Make=="Buick" | cardata$Make=="Cadillac") #subsetting

aggregate(cardata$Price~cardata$Sound,FUN = mean) # return the mean prices for cars with and without soundsystem

aggregate(cardata$Price~cardata$Sound,FUN = median) # return the median prices for cars with and without soundsystem

aggregate(cardata$Price~cardata$Make,FUN = mean) # return the mean prices for all cars for different Make(model/company)

makeavg=aggregate(cardata$Price~cardata$Make,FUN = mean) #assigning a user-defined function (here makeavg) for the above mean price command

#now "makeavg" can be used further in the code to return same output as of the above command

aggregate(cardata$Price~cardata$Make+cardata$Leather,FUN = mean) #return mean prices for Individual Car Make categories, with and without Leather(leather finishing)

aggregate(cardata$Price~cardata$Make+cardata$Leather+cardata$Cruise,FUN = mean) #return mean prices for Individual Car Make categories, with and without Leather(leather finishing) and cruise control

aggregate(cardata$Price~cardata$Type+cardata$Mileage,FUN = mean) #return mean prices for Individual Car Type categories, with different Mileage values

aggregate(cbind(cardata$Price,cardata$Mileage)~cardata$Type,FUN = mean) # return mean price and mean mileage for individual car Type

# the mean values will be displayed under column v1 and v2

aggregate(cbind(Price=cardata$Price,Mile=cardata$Mileage)~cardata$Type, FUN=mean) # same as above just the v1 and v2 columns will be displayed as Price and Mile

**25 Jan: Data: Cardata 2005, Foldername: PGAabhi2, filename: PGAabhi2**

**Basic command, T test, Anova, Chisquare Test, data manipulation**

**25 Jan: Data: Cardata 2005, Foldername: PGAabhi2, filename: PGAabhi2**

**Basic command, T test, Anova, Chisquare Test, data manipulation**

cardata=read.csv(file.choose())

head(cardata)

tail(cardata)

str(cardata)

class(cardata) # gives the class of data

ncol(cardata)

nrow(cardata)

colnames(cardata) # gives column names

rownames(cardata)

summary(cardata)

summary(cardata$Price)

summary(cardata$Mileage)

summary(cardata$Make)

sd(cardata$Price)

sd(cardata$Mileage)

describe(cardata) #all info on central tendency, dispersion and shape need psych package

hist(cardata$Price)

hist(cardata$Mileage)

boxplot(cardata$Price,horizontal = T)

boxplot(cardata$Mileage, horizontal = T)

plot(density(cardata$Price))

plot(density(cardata$Mileage))

plot(density(log(cardata$Price))) # logrithimic transformation

str(cardata)

cardata1=cardata[-c(3:6)] # column deletion

cov(cardata1)

cor(cardata1)

cor.plot(cardata1)

pairs(cardata)

table(cardata$Make)

table(cardata$Type)

table(cardata$Make,cardata$Type)

table(cardata$Make,cardata$Sound)

table(cardata$Make,cardata$Leather)

table(cardata$Make,cardata$Sound,cardata$Leather)

round(prop.table(table(cardata$Make)),digits = 4)\*100

round(prop.table(table(cardata$Model)),digits = 4)\*100

pie(round(prop.table(table(cardata$Model)),digits = 4)\*100)

aggregate(cardata$Price~cardata$Sound,FUN = mean)

aggregate(cardata$Price~cardata$Sound,FUN = sd)

aggregate(cbind(Price=cardata$Price,Mile=cardata$Mileage)~cardata$Sound+cardata$Leather+cardata$Make,FUN = mean)

aggregate(cardata$Price~cardata$Sound,FUN = mean) # mean prices for cvars with and without sound system

t.test(cardata$Price~cardata$Sound)

#Test the Hypothesis that the cruise control Has Significant Impact on Mileage of Car:

str(cardata)

aggregate(cardata$Mileage~cardata$Cruise,FUN = mean)

t.test(cardata$Mileage~cardata$Cruise)

aggregate(cardata$Price~cardata$Leather,FUN = mean)

t.test(cardata$Price~cardata$Leather)

aggregate(cardata$Price~cardata$Make,FUN = mean) #Anova

summary(aov(cardata$Price~cardata$Make))

aggregate(cardata$Mileage~cardata$Type, FUN = mean)

summary(aov(cardata$Mileage~cardata$Type))

# chi-square test:

# Thus a relationship exist between the Make and Type variables in the data

table(cardata$Make,cardata$Type)

chisq.test(table(cardata$Make,cardata$Type)) # table is always an input command for chi sq test

# Thus a relationship exist between the Type and Cruise variables in the data

table(cardata$Type,cardata$Cruise)

chisq.test(table(cardata$Type,cardata$Cruise))

#chisquare test of proportion:

#Test weather cars with and without sound systems are in equal proportions

table(cardata$Sound)

chisq.test(table(cardata$Sound)) # by default this command check for equal probabilities

chisq.test(table(cardata$Sound), p=c(0.3,0.7)) # chsq test for given probabilities since P value=.196, which is >0.05 so accept null i.e. the cars with and without sound systems are not in equal proportions

#Data manipulation (creating new variables based on existing variables)

summary(cardata$Price)

cardata$PriceType=ifelse(cardata$Price>=35000,"High Price",ifelse(cardata$Price>=15000,"Medium Price","Low Price"))

summary(cardata$PriceType)

table(cardata$PriceType)

table(cardata$Price,cardata$PriceType)

summary(cardata$Mileage)

cardata$usetype=ifelse(cardata$Mileage>=30000,"Very old",ifelse(cardata$Mileage>=20000,"Old","Recent"))

table(cardata$usetype)

#chisquare test on this:

table(cardata$PriceType,cardata$usetype)

chisq.test(table(cardata$PriceType,cardata$usetype))

cor(cardata$Price,cardata$Mileage)

str(cardata)

cardata$Cylinder=as.factor(cardata$Cylinder) # as.factor converts numerical into character

str(cardata)

cardata$Liter=as.factor(cardata$Liter)

cardata$Doors=as.factor(cardata$Doors)

cardata$Cruise=as.factor(cardata$Cruise)

cardata$Sound=as.factor(cardata$Sound)

cardata$Leather=as.factor(cardata$Leather)

str(cardata)

**29th Jan multiple linear regression**

usedcarprice=read.csv(file.choose())

head(usedcarprice)

tail(usedcarprice)

str(usedcarprice)

summary(usedcarprice)

summary(usedcarprice$Price)

class(usedcarprice)

nrow(usedcarprice)

ncol(usedcarprice)

hist(usedcarprice$Price)

plot(hist(log(usedcarprice$Price)))

boxplot(usedcarprice$Price,horizontal = T)

plot(density(usedcarprice$Price))

plot(density(log(usedcarprice$Price)))

table(usedcarprice$FuelType)

table(usedcarprice$AutoType)

str(usedcarprice)

aggregate(usedcarprice$Price~usedcarprice$AutoType,FUN = mean)

pairs(usedcarprice)

t.test(usedcarprice$Price~usedcarprice$Automatic)

aggregate(usedcarprice$Price~usedcarprice$FuelType, FUN = mean)

summary(aov(usedcarprice$Price~usedcarprice$FuelType))

summary(usedcarprice)

usedcarprice$ageclass=ifelse(usedcarprice$Age>=70, "very old",ifelse(usedcarprice$Age>=55,"old",ifelse(usedcarprice$Age>=44,"recent","new")))

summary(usedcarprice$ageclass)

table(usedcarprice$ageclass)

table(usedcarprice$ageclass,usedcarprice$FuelType)

chisq.test(table(usedcarprice$ageclass,usedcarprice$FuelType))

# Simple Multiple Linear Regression example

str(usedcarprice)

usedcarprice1=usedcarprice[-c(1,7,8)]

usedcarnumeric=Filter(is.numeric,usedcarprice1) # Filter command will select only the type of variables(categorical, numeric orinterger using is.\_\_)

str(usedcarnumeric)

cov(usedcarnumeric)

pairs(usedcarnumeric)

usedcarreg1=lm(Price~.,data=usedcarprice1)

summary(usedcarreg1)

colnames(usedcarprice1)

usedcarreg2=lm(Price~Age+KM+FuelType+HP+CC+Doors+Weight+AutoType+MetColorType+ageclass,data=usedcarprice1)

summary(usedcarreg2)

anova(usedcarreg1,usedcarreg2)

plot(usedcarreg2)

usedcarreg2$residuals # errors

usedcarreg2$fitted.values # Predicted value

sqrt(mean(usedcarreg2$residuals^2)) #RMSE

#step selection of variables in reg model

usedcarstep=step(usedcarreg1)

summary(usedcarstep)

write.csv(usedcarstep,"usedcarreg2.csv")

cars=mtcars

write.csv(mtcars,"mtcars.csv")

getwd() # will return theworking file directory location

**30/1/2018: folder name: abhi03 file name: abhi04**

**(linear regression and step regression)**

cars=mtcars

str(cars)

write.csv(mtcars,"mtcars.csv")

cars1=cars[-c(8,9)]

str(cars1)

carsreg1=lm(mpg~.,data=cars)

summary(carsreg1)

colnames(cars)

carsreg2=lm(mpg~disp+hp+wt+am,data=cars)

summary(carsreg2)

carsstep=step(carsreg1) #in step regression model must consist on models with all variables

summary(carsstep)

str(cardata)

#stepreg on cardata

cardata=read.csv(file.choose())

head(cardata)

tail(cardata)

cardatareg1=lm(Price~.,data=cardata)

summary(cardatareg1)

colnames(cardata)

str(cardata)

cardatastep=step(cardatareg1)

summary(cardatastep)

**New project Logistic regression on telecom churn data**

**Folder name: abhi03, Filename: telcochurn**

telcochurn=read.csv(file.choose())

head(telcochurn)

tail(telcochurn)

str(telcochurn)

telcochurn1=telcochurn1[-c(1,3,4)]

#How many customers have churned

table(telcochurn1$Churn)

# H M customer churnedsubscribed to int plan

table(telcochurn1$Churn,telcochurn1$International.Plan)

#h M costomer sub to Int plan & voicemail plan

table(telcochurn1$International.Plan,telcochurn1$Voice.mail.Plan)

#Is there a rel betwen Acc Length &Numcustomerservice calls

cov(telcochurn1$Account.Length,telcochurn1$Number.Customer.Service.calls)

cor(telcochurn1$Account.Length,telcochurn1$Number.Customer.Service.calls)

plot(telcochurn1$Account.Length,telcochurn1$Number.Customer.Service.calls)

table(telcochurn1$Number.Customer.Service.calls)

table(telcochurn1$Churn,telcochurn1$Number.Customer.Service.calls)

barplot(table(telcochurn1$Churn,telcochurn1$Number.Customer.Service.calls),legend.text = T)

#test the hypothesis the avg account length of customer chrun True & False are equal

aggregate(telcochurn1$Account.Length~telcochurn1$Churn,FUN = mean)

aggregate(telcochurn1$Account.Length~telcochurn1$Churn,FUN = median)

aggregate(telcochurn1$Account.Length~telcochurn1$Churn,FUN = sd)

t.test(telcochurn1$Account.Length~telcochurn1$Churn)

#test the hypothesis the avg Num custservice calls of customer chrun True & False are equal

aggregate(telcochurn1$Number.Customer.Service.calls~telcochurn1$Churn,FUN = mean)

aggregate(telcochurn1$Number.Customer.Service.calls~telcochurn1$Churn,FUN = median)

aggregate(telcochurn1$Number.Customer.Service.calls~telcochurn1$Churn,FUN = sd)

t.test(telcochurn1$Number.Customer.Service.calls~telcochurn1$Churn)

# Is there relationship bet churn and International Plan subs

table(telcochurn1$Churn,telcochurn1$International.Plan)

chisq.test(table(telcochurn1$Churn,telcochurn1$International.Plan))

# logistic regression

telcologit=glm(Churn~.,data=telcochurn1,family = "binomial")

summary(telcologit)

anova(telcologit,test = "Chisq") #analysis of deviance table

telconumeric=Filter(is.numeric,telcochurn1)

cor(telconumeric)

telcologit$fitted.values

head(telcologit$fitted.values,10)

table(Actual=telcochurn1$Churn,Predicted=telcologit$fitted.values>=0.5) # return the confusion matrix

logitaccuracy=(4191+152)/(4191+102+555+152) # logistic reg accuracy

logitaccuracy

logitclass=ifelse(telcologit$fitted.values>=0.5," True."," False.")

head(logitclass)

confusionMatrix(logitclass,telcochurn1$Churn) #in Con. matrix first give predicted term then actual term

**1st Feb**

#Decision Trees on Bank data

bankdata=read.csv(file.choose())

head(bankdata)

tail(bankdata)

str(bankdata)

summary(bankdata)

colSums(is.na(bankdata)) #return the total missing values in all columns

# Test the Hypo.the avg. balance of customers subscribed to term deposit(y) is equal

aggregate(bankdata$balance~bankdata$y,FUN = mean)

aggregate(bankdata$balance~bankdata$y,FUN = median)

t.test(bankdata$balance~bankdata$y)

# Test hypo the avg of customers with diff. marital status is equal or same

aggregate(bankdata$balance~bankdata$marital,FUN = mean)

summary(aov(bankdata$balance~bankdata$marital))

#Test hypo that there is relaionship bet maritial status and education

table(bankdata$marital,bankdata$education)

chisq.test(table(bankdata$marital,bankdata$education))

#decision tree

bankdatarpart=rpart(y~.,data = bankdata)

summary(bankdatarpart)

bankginipredict=predict(bankdatarpart,type="class") # predict

head(bankginipredict)

table(Predicted=bankginipredict,Actual=bankdata$y)

(38904+1845)/(38904+3444+1018+1845)

bankdatarpart1=rpart(y~.,data = bankdata,parms = list(split="information"))

summary(bankdatarpart1)

rpart.plot(bankdatarpart1)

bankinfopredict=predict(bankdatarpart1,type="class")

head(bankinfopredict)

table(Predicted=bankinfopredict,Actual=bankdata$y)

(38853+1855)/(38853+3434+1069+1855)

#Splitting Data into train and test dataset (Random Sampling)

samplesize=floor(0.80\*nrow(bankdata))

set.seed(123)

trainindex=sample(seq\_len(nrow(bankdata)),size = samplesize)

train=bankdata[trainindex,]

test=bankdata[-trainindex,]

#bias sampling

biastrainsample=bankdata[1:36000,]

biastestsample=bankdata[36001:45211,]

**2nd Feb**

str(train)

trainlogit=glm(y~.,data=train,family = "binomial")

summary(trainlogit)

table(Predicted=trainlogit$fitted.values>0.5,train$y) #gives the confusion matrix to check accuracy

(31146+1524)/(31146+2719+779+1524)

testlogit=predict(trainlogit,test,type="response") #Preciting the Y class

table(Predicted=testlogit>0.5,test$y) #gives the confusion matrix to check accuracy

(7766+346)/(7766+700+231+346)

biastrainsamplelogit=glm(biastrainsample$y~.,data=biastrainsample,family = "binomial")

summary(biastrainsamplelogit)

table(Predicted=biastrainsamplelogit$fitted.values>0.5,biastrainsample$y)

(33118+704)/(33118+1722+456+704)

biastestsamplelogit=predict(biastrainsamplelogit,test,type="response")

**#decisiontree for regression model**

str(usedcars)

usedcarrpart=rpart(Price~.,data=usedcars[-1])

summary(usedcarrpart)

rpart.plot(usedcarrpart)

usedcarrpartpredict=predict(usedcarrpart,type = "vector")

head(usedcarrpartpredict)

sqrt(mean(usedcars$Price-usedcarrpartpredict)^2)

bankrandforest=randomForest(y~.,data=bankdata,ntree=3000,importance=T,do.trace=100)

varImpPlot(bankrandforest) # return the plot of variable importance

**#on telcochurn**

telcorpart=rpart(Churn~.,data=telcochurn1)

summary(telcorpart)

rpart.plot(telcorpart)

rpartpredict=predict(telcorpart,type = "class")

table(Predicted=rpartpredict,Actual=telcochurn1$Churn)

(4233+512)/(4233+195+60+512)

telcorandforest=rfcv(telcochurn1[-18],telcochurn1$Churn,cv.fold = 5)

telcorandforest

telcomrandforest=randomForest(Churn~.,data=telcochurn1,ntree=3000,importance=T,do.trace=100)

print(telcomrandforest)

**5th feb German credit risk data hypothesis and applying for loop, ANN,SVM and Naïve Bayes algorithms**

gcredit=read.csv(file.choose())

head(gcredit)

tail(gcredit)

str(gcredit)

summary(gcredit)

hist(gcredit$Credit.Amount)

boxplot(gcredit$Credit.Amount, horizontal = T)

plot(density(gcredit$Credit.Amount))

plot(density(log(gcredit$Credit.Amount)))

cov(gcredit$Duration.of.Credit..month.,gcredit$Age..years.)

cor(gcredit$Duration.of.Credit..month.,gcredit$Age..years.)

table(gcredit$Purpose)

sum(table(gcredit$Purpose))

(234+103+181+280+12+22+50+9+97+12)

**>Check the hypothesis that the mean credit amount is same for all the credibility classes**

aggregate(gcredit$Credit.Amount~gcredit$Creditability,FUN = mean)

aggregate(gcredit$Credit.Amount~gcredit$Creditability,FUN = median)

t.test(gcredit$Credit.Amount~gcredit$Creditability)

> **Check the hypothesis that the mean age of people is same for all the credibility classes**

aggregate(gcredit$Age..years.~gcredit$Creditability,FUN = median)

aggregate(gcredit$Age..years.~gcredit$Creditability,FUN = mean)

t.test(gcredit$Age..years.~gcredit$Creditability)

> **Check the hypothesis that the average credit amount is same for all the Purpose classes**

aggregate(gcredit$Credit.Amount~gcredit$Purpose,FUN = mean)

summary(aov(gcredit$Credit.Amount~gcredit$Purpose))

> **Check the hypothesis that the average credit amount is same for all the Payment status classes**

aggregate(gcredit$Credit.Amount~gcredit$Payment.Status.of.Previous.Credit,FUN = mean)

summary(aov(gcredit$Credit.Amount~gcredit$Payment.Status.of.Previous.Credit))

> **Check the hypothesis that there exist relationship between credibility and account balance**

table(gcredit$Creditability,gcredit$Account.Balance)

chisq.test(table(gcredit$Creditability,gcredit$Account.Balance))

> **Check the hypothesis that there exist relationship between credibility and Marital Status**

table(gcredit$Creditability,gcredit$Sex...Marital.Status)

chisq.test(table(gcredit$Creditability,gcredit$Sex...Marital.Status))

str(gcredit)

factordata=gcredit[-c(3,6,14)]

str(factordata)

#For loop applied to convert factor variables from codded int to factor format

for (i in 1:ncol(factordata)) {

factordata[,i]=as.factor(factordata[,i])

}

str(factordata)

gcredit1=data.frame(factordata,gcredit[c(3,6,14)])

str(gcredit1)

#neural network alg:

creditneural=nnet(Creditability~.,data = gcredit1,size=5)

summary(creditneural)

# confusion matrix

table(Predicted=creditneural$fitted.values>0.5,Actual=gcredit1$Creditability)

#SVM

creditsupportvector=ksvm(Creditability~.,data=gcredit1,cross=3)

print(creditsupportvector)

supportvectorpredict=predict(creditsupportvector,type="response")

table(supportvectorpredict,gcredit1$Creditability)

(161+675)/(161+25+139+675)

#naive bayes

creditnaivebayes=naiveBayes(Creditability~.,data=gcredit1)

print(creditnaivebayes)

naivebayespredict=predict(creditnaivebayes,gcredit1,type="class")

table(naivebayespredict,gcredit1$Creditability)

(163+610)/(163+90+137+610)

**4th March Mckinsey Loan Prediction Problem with mice package for missing value imputation:**

loantrain=read.csv(file.choose(),na.strings=c("","NA"),stringsAsFactors = F)

sort(colSums(is.na(loantrain)),decreasing = T)

loantest=read.csv(file.choose(),na.strings = c("","NA"),stringsAsFactors = F)

sort(colSums(is.na(loantest)),decreasing = T)

loantrain$datatype=1

loantest$datatype=2

combineddata=rbind(loantrain[-21],loantest)

sort(colSums(is.na(combineddata)),decreasing = T)

combined=combineddata[-1]

combcat=Filter(is.character,combined)

combnum=Filter(is.numeric,combined)

sort(colSums(is.na(combcat)),decreasing = T)

catcols=c('Customer\_Existing\_Primary\_Bank\_Code','Primary\_Bank\_Type','Employer\_Code','Employer\_Category1','City\_Code','City\_Category')

for (col in catcols) {

combcat[sapply(combcat[col], function(x) ifelse(is.na(x),names(which.max(table(x))),x))]

}

combcat=combcat[sapply(combcat, function(x) ifelse(is.na(x),names(which.max(table(x))),x))]