R- workshop- day2 – program codes

Sample code is in histogram-example.r

```
rm(list=ls())
# Create data for the graph.
v <-  c(9,13,21,8,36,22,12,41,31,33,19)
# Create the histogram.
hist(v,xlab = "Weight",main="Histogram",col = "yellow",border = "blue")
```

Sample code is in boxplot-example.r

```
rm(list=ls())
## Read the csv file
car<-read.csv("car_data.csv")
# Exploratory data analysis
View(car)
dim(car)
head(car)
tail(car)
car1<-car[,-1]
View(car1)
head(car1)
#summary statistics for int and numeric data
summary(car1$mpg)
summary(car1$cyl)
mpg<-car1$mpg
cyl<-car1$cyl
# Plot the chart.
boxplot(mpg ~ cyl, data=mtcars, xlab="Number of Cylinders", ylab="Miles Per Gallon",
        main="Mileage Data",  col=c("green","yellow","purple"))
boxplot(mpg,main="Mileage Data")
boxplot(cyl, main="cyl")
```

Sample code is in barplot-example.r

```
rm(list=ls())
# Create the data for the chart.
H <- c(7,12,28,3,41)
# Plot the bar chart.
barplot(H, main="bar chart")
# Create the data for the chart.
H <- c(7,12,28,3,41)
M <- c("Mar","Apr","May","Jun","Jul")
# Plot the bar chart.
barplot(H,names.arg=M,xlab="Month",ylab="Revenue",col="blue", main="Revenue
        chart",border="red")
##Group Bar Chart and Stacked Bar Chart
# Create the input vectors.
colors <- c("green","orange","brown")
months <- c("Mar","Apr","May","Jun","Jul")
regions <- c("East","West","North")
# Create the matrix of the values.
Values <- matrix(c(2,9,3,11,9,4,8,7,3,12,5,2,8,10,11),nrow=3,ncol=5,byrow=TRUE)
Values
# Create the bar chart.
barplot(Values,main="total
revenue",names.arg=months,xlab="month",ylab="revenue",col=colors)
# Add the legend to the chart.
legend("topleft", regions, cex=0.25, fill=colors)
```

Sample code is in linechart-example.r

```
rm(list=ls())
# Create the data for the chart.
v <- c(7,12,28,13,41)
# Plot the line chart.
plot(v,type = "l")
```

"p" for points, "o" for both 'overplotted', "l" for lines,"b" for both

Plot the line chart.

plot(v,type = "o", col = "red", xlab = "Month", ylab = "Rain fall", main = "Rain fall chart")

Create the data for the chart.

v <- c(5,10,28,13,41)

t <- c(14,7,6,19,3)

Plot the line chart.

plot(v,type = "o",col = "red", xlab = "Month", ylab = "Rain fall", main = "Rain fall chart")

lines(t, type = "o", col = "blue")

More than one line can be drawn on the same chart by using the lines() function.

##After the first line is plotted, the lines() function can use an additional vector as input to ##draw the second line in the chart,

Example 2

##Rainfall chart for 7 days of 2 months

jan<-c(157,160,140,146,155,162,149)

feb<-c(148,159,142,139,162,148,155)

plot(jan,xlab="days",ylab="Rainfall",col="red",type="o",

    main="Rainfall chart for 7 days of 2 months")

lines(feb,col="blue",type="o")

Sample code is in slr-example1.r

rm(list=ls())

Values of height

x <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)

Values of weight.

y <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)

dat<-data.frame(x,y)

dat

Apply the lm() function.

relation <- lm(y~x, data = dat)

```
print(relation)
summary(relation)
##Predict the weight of new persons
# Find weight of a person with height 170.
a <- data.frame(x=170)
result <-  predict(relation,a)
print(result)
##Visualize the Regression Graphically
# Plot the chart.
plot(y,x,col = "blue",main = "Height & Weight Regression",
   abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "Weight in Kg",ylab = "Height in cm")
```

Sample code is in slr-example2.r

```
rm(list=ls())
library("MASS")
data("cars")
head(cars)
str(cars)
summary(cars)
table(is.na(cars))
##There are no null values or missing data in the cars
##dataset. The is.na checks for nulls
##and returns TRUE if nulls exists and FALSE if no nulls.
plot(cars, col='blue', pch=20, cex=2, main="Relationship between Speed and Stopping Distance for
        50 Cars", xlab="Speed in mph", ylab="Stopping Distance in feet")
cor(cars$speed, cars$dist)
## correlation coefficient can take values between -1 to +1 .
##A value closer to +1 suggests strong correlation and a value
##closer to -1 suggests weaker correlation.
##The correlation value is 0.81 which shows that roughly that there is somewhat a strong positive
##correlation between speed and the distance required to stop
mod1 = lm(dist ~ speed, data = cars)
```

```
print(mod1)
summary(mod1)
plot(cars, col='blue', pch=20, cex=2, main="Relationship between Speed and Stopping Distance for
      50 Cars", xlab="Speed in mph", ylab="Stopping Distance in feet")
abline (mod1,col ="red")
##Overall, the relationship between Speed and the distance required to stop the car is positively
##correlated.  There is a linear correlation between the 2 variables.
```

Sample code is in read_csv_apply.r

```
rm(list=ls())
car<-read.csv("car_data.csv")
# Exploratory data analysis
View(car)
dim(car)
head(car)
tail(car)
car1<-car[,-1]
head(car1)
#structure of each attribute.
str(car1)
dim(car1)
#summary statistics for int and numeric data
summary(car1$mpg)
summary(car1$disp)
summary(car1$drat)
summary(car1)
boxplot(car1$mpg,main="mpg")
boxplot(car1$disp,main="disp")
#draw the box plot of numeric quantitative data
#and find the outliers.
par(mfrow=c(2,2))
boxplot(car1$mpg,main="mpg")
```

```
boxplot(car1$disp,main="disp")
boxplot(car1$hp,main="hp")
boxplot(car1$wt,main="weight")
#par(mfrow=c(1,1))
# another method for measuring the central tendency.
#histogram of numeric data
par(mfrow=c(2,2))
hist(car1$mpg,main = "MPG")
hist(car1$cyl,main="cyl")# categorical data, so it is meaningless
hist(car1$disp,main="disp")
hist(car1$wt,main="weight")
## use of apply() function
p1<-apply(car1,2,mean)
print(p1)
p2<-apply(car1,2,median)
print(p2)
p3<-apply(car1,2,sd)
print(p3)
p4<-apply(car1,2,min)
print(p4)
p5<-apply(car1,2,max)
print(p5)
#table()
f_cyl<-table(car1$cyl)
print(f_cyl)
f_gear<-table(car1$gear)
print(f_gear)
f<-table(car1$cyl,car1$gear)
print(f)
f_carb<-table(car1$carb)
print(f_carb)
```

```
par(mfrow=c(2,2))
barplot(f_cyl,col = rainbow(3),main="number of cylinders", xlab="num_cyl", ylab="Count")
barplot(f_gear,col=c("yellow","black","red"))
barplot(f_carb,col=c("yellow","black","red"))
barplot(f,col=c("yellow","black","red"))
# mpg as responce variable
library("corrplot")
y<-car1$mpg
x1<-car1$disp
x2<-car1$hp
m<-cor(car1)
corrplot(m,method="ellipse")
m1<-cor(x1,y)
m2<-cor(x2,y)
m3<-cor(car1[c("mpg","disp","hp")])
m3
corrplot(m3,method = "ellipse") # draw the scatter plot
plot(x1,y,main = "mpg~disp",xlab = "disp",ylab = "mpg")
# simple linear regression with y as mpg and x as disp
L1<-lm(y~x1)
summary(L1)
plot(x2,y,main = "hp~disp",xlab = "disp",ylab = "hp")
L2<-lm(y~x2)
abline(L2,col="red")
abline(L1,col="red")
coefficients(L1)
summary(L1)
pairs(car1)
l3<-lm(y~x1+x2)
summary(l3)
```

Sample code is in logistic-example1.r

```
rm(list=ls())
#Install required packages
install.packages('caret')
#Import required library
library(caret)
## read the data using url
df <- read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
write.csv(df,"output-df.csv") ## create a csv file
df1<-read.csv("output-df.csv")
dim(df1)
head(df1)
 str(df1)
head(df1)
## check for null values
sum(is.na(df1))
summary(df1)
rank_count <- as.factor(df1$rank)
rank_count
admit_count<-as.factor(df1$admit)
admit_count
logit <- glm(admit ~ gre+gpa+rank,data=df1,family="binomial")
summary(logit)
##A student have a profile with 790 in GRE, 3.8 GPA and he studied from a rank-1 college.
##Now you want to predict the ##chances of that boy getting admit in future.
anova(logit)
x <- data.frame(gre=790,gpa=3.8,rank=1)
p<- predict(logit,x)
p
#there is 85% chance that this guy will get the admit.
predicted <- predict(logit, df1, type="response")
```

```
predicted
p1<-round(predicted, digits = 2)
print(p1)
#converting probabilities into 1 and 0
pre <- ifelse(p1>=0.5, 1, 0)
print(pre)
## Confusion matrix is created with table() fun. and create the matrix format from this
cm<-as.matrix(table(df1$admit , pre))
head(cm)
View(cm)
```

Sample code is in logistic-example2.r

```
rm(list=ls())
# Load the required packages
library(caret)
library("MASS")
install.packages("caret")
# Load the dataset
#Import required library
library(caret)
data("iris")
head(iris)
library(e1071)
library(caTools)
iris <- read.csv("iris.csv")
View(iris)
head(iris)
str(iris)
summary(iris)
plot(sepal_length ~ petal_length,  xlab = "Petal Length (cm)",  ylab = "Sepal Length (cm)",
   pch = c(16, 17, 18)[as.numeric(species)],  # different 'pch' types  main = "Iris Dataset",
```

```
    col = c("red", "green","blue")[as.numeric(species)],  data = iris)
# Split the dataset into training and testing sets
sample<- sample.split(iris$species, SplitRatio = 0.7)
trainData<-subset(iris,sample==TRUE)
testData<-subset(iris,sample==FALSE)
dim(trainData)
table(traiDatan$species)
dim(testData)
table(testData$species)
# Train the logistic regression model
logRegModel <- glm(species ~ ., data = trainData, family = "binomial")
summary(logRegModel)
# Use the trained model to predict the outcomes
predictions <- predict(logRegModel, trainData)
predictions
p1<-round(predictions, digits = 2)
print(p1)
#converting probabilities into 1 and 0
pre <- ifelse(p1>=0.5, 1, 0)
print(pre)
## Confusion matrix is created with table() fun. and create the matrix format from this
cm<-as.matrix(table(trainData$species , pre))
head(cm)
View(cm)
# Test the logistic regression model using teat dat
logRegModel2 <- glm(species ~ ., data = testData, family = "binomial")
# Use the trained model to predict the outcomes
prediction <- predict(logRegModel2, testData)
prediction
p1<-round(prediction, digits = 2)
print(p1)
```

```
#converting probabilities into 1 and 0
pre <- ifelse(p1>=0.5, 1, 0)
print(pre)
## Confusion matrix is created with table() fun. and create the matrix format from this
cm<-as.matrix(table(testData$species , pre))
head(cm)
View(cm)
```

Sample code is in nn-example.r

```
rm(list=ls())
#creating training data set
Sci = c(70,71,72,73,68,69,65,69,80,68)
Mat = c(91,92,93,94,65,69,61,55,91,79)
Eng = c(82,83,84,85,73,66,50,62,95,68)
Pass = c(1,1,1,1,0,0,0,0,1,0)
df=data.frame(Sci,Mat,Eng,Pass)
#installing and loading neuralnet package
install.packages("neuralnet")
library("neuralnet")
#fitting the model
nn=neuralnet(Pass~Sci+Mat+Eng,data=df, hidden=3,act.fct = "logistic", linear.output = FALSE)
#Pass~Sci+Mat+Eng, Pass is label and Sci,Mat,Eng are features.
#df  is dataframe, hidden=3 stands for a single hidden layer with 3 neurons
#act.fct = "logistic" is used for smoothing the result.
#linear.ouput=FALSE is set FALSE for apply act.fct
#Plot the neural network
plot(nn)
#creating testing set
sci = c(80,75,65,68)
mat = c(95,92,69,45)
eng = c(85,83,55,50)
```

```
test=data.frame(sci,mat,eng)
#prediction
predict=compute(nn,test)
predict$net.result
probab<- predict$net.result
p<-round(probab, digits = 2)
print(p)
#converting probabilities into 1 and 0
pre <- ifelse(p>=0.5, 1, 0)
print(pre)
```

Sample code of- Naive Bayes algorithm

```
rm(list=ls())
library(e1071)
library(caTools)
library(caret)
iris <- read.csv("iris.csv")
View(iris)
head(iris)
str(iris)
summary(iris)
#Check for null values
sum(is.na(iris))
table(iris$species)
sample<- sample.split(iris$species, SplitRatio = 0.7)
train<-subset(iris,sample==TRUE)
test<-subset(iris,sample==FALSE)
dim(train)
table(train$species)
dim(test)
```

```
table(test$species)
nb<-naiveBayes(train$species~., data = train)
nb
pred<- predict(nb,test)
cm<-table(pred,test$species)
cm
#Model Evaluation
confusionMatrix(cm)
```