**Salam karo bosdiwalo tumare baap ko**

**For Dark Mode in Rstudio**

**- Tools > Global Options… > Appearance > in the Editor theme select Twilight theme and tap Apply and Ok**

**Practical DS Viva**

---------------------------------------------------------------------------------------- **- What is a Time Series =** A Time Series is a sequence of data points collected or recorded at regular time intervals.

**- What is lattice =** Lattice is an R package for producing trellis graphics, which are visualizations used for data analysis, especially for conditioning plots.

**- What is plotly =** Plotly for R is a library for creating interactive and multi-layered graphs online using R.

**- What is zoo =** Zoo is an R package that provides an infrastructure for regular and irregular time series data.

**- What does dev.off() do =** dev.off() closes the current graphics device in R, useful for resetting the graphical output or concluding a plotting session. **- What is automob =** "Automob" likely refers to a variable or dataset related to automobiles in the context of an R programming task.

**- What's paste in this =** In R, paste() is a function that concatenates strings together, possibly separating them with a specified character. **- What is xts =** Xts is an R package providing data structures and functions to work with time-indexed data, extending the zoo package capabilities. **- What are abline, reg=lm in this =** In this context, abline is used to add a straight line to a plot, and reg=lm specifies that this line should be based on a linear model.

**- What is a boxplot =** A boxplot is a standardized way of displaying the distribution of data based on a five-number summary and outliers. **- What is aov in this =** aov stands for Analysis of Variance, a statistical method used in R to analyze the differences among group means.

**- What are predict, confidence in this =** predict is used to make predictions based on a model, and "confidence" specifies the type of prediction interval as a confidence interval.

**- What is a linear model =** A linear model is a statistical approach to modelling the relationship between a scalar response and one or more explanatory variables.

**- What is Hypothesis testing =** Hypothesis testing is a statistical method that uses sample data to evaluate a hypothesis about a population parameter. **- What is Linear Regression =** Linear Regression is a basic and commonly

used type of predictive analysis which models the relationship between a dependent variable and one or more independent variables by fitting a linear equation.

**- What is a Directional Hypothesis =** A Directional Hypothesis predicts the direction of an expected difference or relationship between variables. **- What is Analysis of Variance =** Analysis of Variance (ANOVA) is a statistical technique that compares mean differences among groups to see if they are statistically significant.

**- What is party in this =** In R, party is a package used for creating and analyzing complex statistical models, such as conditional inference trees. **- How do you create a vector in R =** Use the c() function in R to create a vector, which is a sequence of data elements of the same basic type. **- What does the lm function do in R =** The lm() function fits linear models, allowing for prediction of a dependent variable from one or more independent variables.

**- Explain the data.frame function in R =** The data.frame() function creates data frames, which are tables that store data in rows and columns, allowing for different types of variables.

**- Explain the glm function in R =** The glm() function fits generalized linear models, a framework for modeling a wide variety of data, including binary outcomes, count data, and more, using extensions of linear regression. ---------------------------------------------------------------------------------------- **Installation of R Studio and R Lang**

**- install from the website**

=================================================================== **Practical 1**

=================================================================== 1) Variable Declaration

---------------------------------------------------------------------------------------- name <-"john"

age <- 20

name

---------------------------------------------------------------------------------------- 2) for loop

---------------------------------------------------------------------------------------- for (i in 1:10)print(i)

---------------------------------------------------------------------------------------- 3) Paste

----------------------------------------------------------------------------------------

print("hello")

text <- "hey"

paste("haha...")

var1 <- var2 <- var3 <- "john"

var1

var2

var3

---------------------------------------------------------------------------------------- 4) if statement

---------------------------------------------------------------------------------------- x=5

if(x>10)

{}

{

print(paste(x, "is greater than 10"))

}

---------------------------------------------------------------------------------------- 5) if else statement

---------------------------------------------------------------------------------------- x <- 5

if (x > 10) {

print(paste(x, "is greater than 10"))

} else if (x < 10) {

print(paste(x, "is less than 10"))

} else {

print("x is equal to 10")

}

---------------------------------------------------------------------------------------- 6) Array

---------------------------------------------------------------------------------------- automob <- c("car", "bike", "truck")

automob

--------------------------------------------

num <- 1:10

num

--------------------------------------------

length(num)

--------------------------------------------

num <- c(13,2,34,55,3)

sort(num)

--------------------------------------------

num[3]

=================================================================== **Practical 2**

=================================================================== 1) Time Series

- ggplot2

- lattice

- plotly

- zoo

- xts

---------------------------------------------------------------------------------------- rainfall <- c(242,4242,53252,262,464,343,34,3444,556,363,385,388) rainfall.timeseries <- ts(rainfall, start=c(2012,1), frequency = 12) print(rainfall.timeseries)

---------------------------------------------------------------------------------------- 2) Plotting a Graph

---------------------------------------------------------------------------------------- plot(rainfall.timeseries)

png(file="rainfall.png")

plot(rainfall.timeseries)

dev.off()

---------------------------------------------------------------------------------------- 3) rain.png

---------------------------------------------------------------------------------------- rainfall <- c(242,4242,53252,262,464,463,34,444)

rainfall.timeseries <- ts(rainfall, start=c(2012,1), frequency = 15) print(rainfall.timeseries)

--------------------------------------------

plot(rainfall.timeseries)

png(file="rainfall.png")

plot(rainfall.timeseries)

dev.off()

=================================================================== **Practical 3**

===================================================================

1) Dataset

---------------------------------------------------------------------------------------- data("AirPassengers")

class(AirPassengers)

--------------------------------------------

start(AirPassengers)

--------------------------------------------

end(AirPassengers)

--------------------------------------------

frequency(AirPassengers)

--------------------------------------------

summary(AirPassengers)

--------------------------------------------

plot(AirPassengers)

--------------------------------------------

print(AirPassengers)

--------------------------------------------

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

--------------------------------------------

boxplot(AirPassengers~cycle(AirPassengers))

---------------------------------------------------------------------------------------- 2) Cars

---------------------------------------------------------------------------------------- print("cars")

--------------------------------------------

data("cars")

class(cars)

--------------------------------------------

frequency(cars)

--------------------------------------------

summary(cars)

--------------------------------------------

plot(cars)

---------------------------------------------------------------------------------------- reg <- lm(speed~dist,data=cars)

plot(reg)

<return> - tap enter after every return

<return> - tap enter after every return

<return> - tap enter after every return

<return> - tap enter after every return

=================================================================== **Practical 4**

=================================================================== 1) Analysis Variance

---------------------------------------------------------------------------------------- data("warpbreaks")

head(warpbreaks)

--------------------------------------------

summary(warpbreaks)

--------------------------------------------

Model\_1<- aov(breaks~wool + tension,data = warpbreaks)

summary(Model\_1)

--------------------------------------------

plot(Model\_1)

<return> - tap enter after every return

<return> - tap enter after every return

<return> - tap enter after every return

<return> - tap enter after every return

---------------------------------------------------------------------------------------- 2) Model 2

---------------------------------------------------------------------------------------- Model\_2<- aov(breaks~wool + tension + wool:tension,data = warpbreaks) summary(Model\_2)

--------------------------------------------

plot(Model\_2)

<return> - tap enter after every return

<return> - tap enter after every return

<return> - tap enter after every return

<return> - tap enter after every return

---------------------------------------------------------------------------------------- 3) Plant

---------------------------------------------------------------------------------------- data("PlantGrowth")

summary(PlantGrowth)

--------------------------------------------

levels(PlantGrowth$group)

--------------------------------------------

weight = PlantGrowth$weight

group = PlantGrowth$group

mean(weight)

--------------------------------------------

mean(weight[group=="ctrl"])

--------------------------------------------

mean(weight[group=="trt1"])

--------------------------------------------

mean(weight[group=="trt2"])

--------------------------------------------

tapply(weight,group,mean)

--------------------------------------------

tapply(weight,group,length)

--------------------------------------------

aov(weight ~ group)

=================================================================== **Practical 5**

=================================================================== 1) Linear Regression

---------------------------------------------------------------------------------------- height <- c(43,65,6,6,36,56,43,43,64,7,44,75)

weight <- -c(43,5,6,6,36,465,65,7,65,54,67,45)

student <- lm(weight~height)

print(student)

--------------------------------------------

plot(student)

<return> - tap enter after every return

<return> - tap enter after every return

<return> - tap enter after every return

<return> - tap enter after every return

--------------------------------------------

predict(student, data.frame(height = 19), interval = "confidence") --------------------------------------------

df <- datasets::cars

my\_lm <- lm(dist~speed, data=df)

print(my\_lm)

--------------------------------------------

lm(formula = dist ~ speed, data = df)

--------------------------------------------

variable\_speed <- data.frame(speed = c(11,12,432,354,4,56,54,6,56)) linear\_model <- lm(dist~speed, data=df)

predict(linear\_model, newdata= variable\_speed)

--------------------------------------------

predict(linear\_model, newdata = variable\_speed, interval="confidence") =================================================================== **Practical 6**

=================================================================== 1) Hypothesis testing

---------------------------------------------------------------------------------------- x <- rnorm(100)

t.test(x,mu=5)

---------------------------------------------------------------------------------------- 2) Two Sample Testing

---------------------------------------------------------------------------------------- x <- rnorm(100)

y <- rnorm(100)

t.test(x,y)

---------------------------------------------------------------------------------------- 3) Directional Hypothesis

---------------------------------------------------------------------------------------- x <- rnorm(100)

t.test(x, mu = 2, alternative = 'greater')

---------------------------------------------------------------------------------------- 4) Testing

---------------------------------------------------------------------------------------- dataf <- seq(1,20,by=1)

dataf

mean(dataf)

sd(dataf)

a <- t.test(dataf,alternate="two sided",mu = 10,conf.int=0.95) a

=================================================================== **Practical 7**

=================================================================== 1) Decision Tree

---------------------------------------------------------------------------------------- install.packages("party")

--------------------------------------------

library(party)

--------------------------------------------

print(head(readingSkills))

--------------------------------------------

input.dat<-readingSkills[c(1:105),]

output.tree<-ctree(nativeSpeaker ~ age + shoeSize + score,data=input.dat) plot(output.tree)

=================================================================== **Practical 8**

=================================================================== 1) Logistic regression

---------------------------------------------------------------------------------------- input <- mtcars[,c("am","cyl","hp","wt")]

print(head(input))

---------------------------------------------------------------------------------------- input <- mtcars[,c("am","cyl","hp","wt")]

am.data = glm(formula=am~cyl+hp+wt,data = input,family=binomial) print(summary(am.data))

--------------------------------------------

plot(am.data)

<return> - tap enter after every return

<return> - tap enter after every return

<return> - tap enter after every return

<return> - tap enter after every return

=================================================================== **Practical 9**

=================================================================== 1) K means Clustering

---------------------------------------------------------------------------------------- install.packages("ggplot2")

library(ggplot2)

df <- iris

head(iris)

--------------------------------------------

ggplot(df,aes(Petal.Length,Petal.Width))+geom\_point(aes(col=Species),size=4) --------------------------------------------

set.seed(101)

irisCluster <- kmeans(df[,1:4],center=3,nstart = 20)

irisCluster

--------------------------------------------

install.packages("cluster")

--------------------------------------------

clusplot(iris,irisCluster$cluster,color=T,shade=T,label=0,lines=0) - tick the "cluster" from the package