**Course Code: ITA0447**

**Course Title: STATISTICS WITH R PROGRAMMING FOR NLP**

**LAB DAY : 04**

1. Create below data frame

exam\_data = data.frame(

name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'),

score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),

attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),

qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')

)

a. Write a R program to extract 3rd and 5th rows with 1st and 3rd columns from a given data frame

b. Write a R program to add a new column named country in a given data frame

Country<-c("USA","USA","USA","USA","UK","USA","USA","India","USA","USA")

c. Write a R program to add new row(s) to an existing data frame

new\_exam\_data = data.frame(name = c('Robert', 'Sophia'),score = c(10.5, 9), attempts = c(1, 3),qualify = c('yes', 'no'))

d. Write a R program to sort a given data frame by name and score

e. Write a R program to save the information of a data frame in a file and display the information of the file.

CODE:

exam\_data = data.frame(

name = c('Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'),

score = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19),

attempts = c(1, 3, 2, 3, 2, 3, 1, 1, 2, 1),

qualify = c('yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes')

)

#subdiv1

exam\_data[c(3, 5), c(1, 3)]

#subdiv2

Country <- c("USA", "USA", "USA", "USA", "UK", "USA", "USA", "India", "USA", "USA",)

exam\_data$country <- Country

#sub\_div\_3

new\_exam\_data <- data.frame(name = c('Robert', 'Sophia'), score = c(10.5, 9), attempts = c(1, 3), qualify = c('yes', 'no'))

exam\_data <- rbind(exam\_data, new\_exam\_data)

#subdiv4

exam\_data <- exam\_data[order(exam\_data$name, exam\_data$score), ]

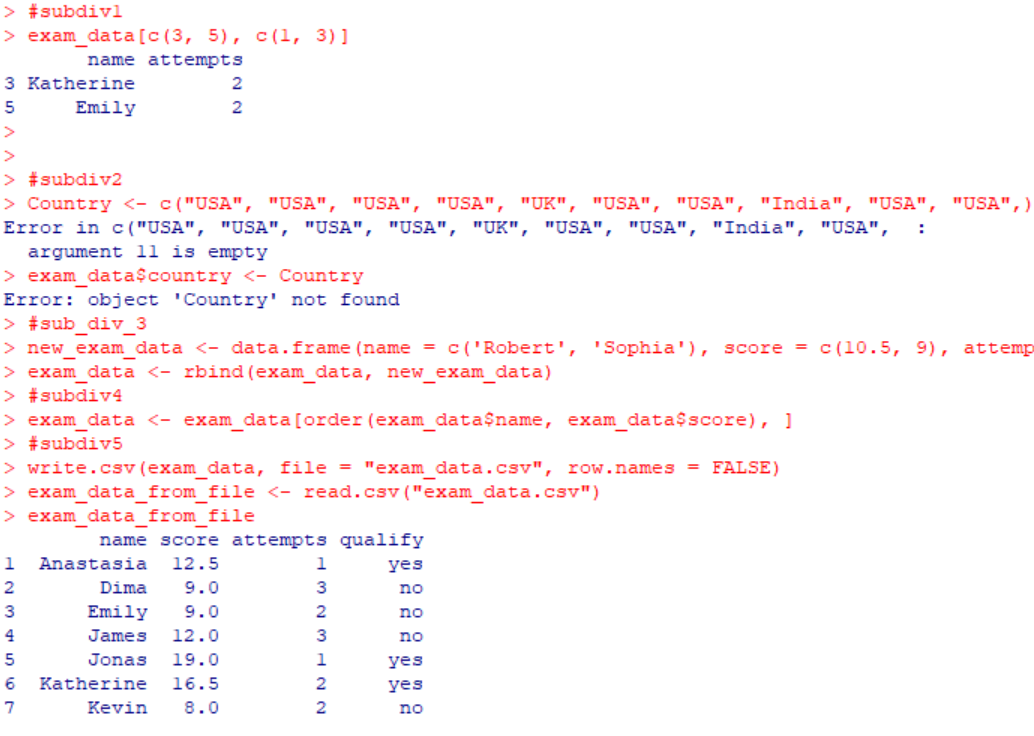
#subdiv5

write.csv(exam\_data, file = "exam\_data.csv", row.names = FALSE)

exam\_data\_from\_file <- read.csv("exam\_data.csv")

exam\_data\_from\_file

OUTPUT:



1. Write a R program to call the (built-in) dataset air quality. Check whether it is a data frame or not? Order the entire data frame by the first and second column. remove the variables 'Solar.R' and 'Wind' and display the data frame.

CODE:

data(airquality)

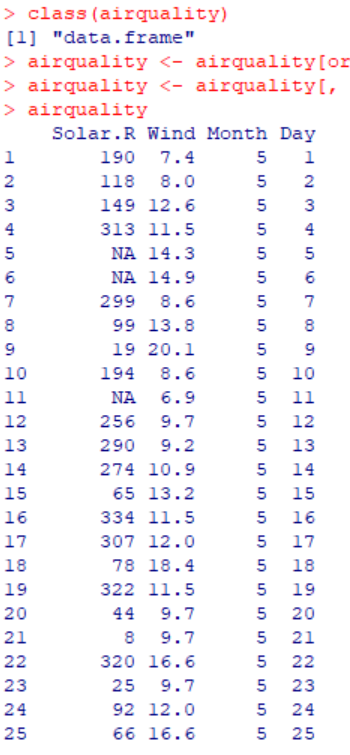
class(airquality)

airquality <- airquality[order(airquality$Month, airquality$Day),]

airquality <- airquality[, -c(1, 4)]

airquality

OUTPUT:



1. Write a R program to create a factor corresponding to height of women data set , which inbuild in R, contains height and weights for a sample of women.

CODE:

data(women)

height <- women$height

height\_factor <- cut(height, breaks = c(50, 55, 60, 65, 70, 75))

levels(height\_factor)

OUTPUT:



1. Write a R program to extract the five of the levels of factor created from a random sample from the LETTERS (Part of the base R distribution.)

CODE:

set.seed(123)

letters\_sample <- sample(LETTERS, 20, replace=TRUE)

letters\_factor <- factor(letters\_sample)

five\_levels <- levels(letters\_factor)[1:5]

five\_levels

OUTPUT:



1. **Iris** dataset is a very famous dataset in almost all data mining, machine learning courses, and it has been an R build-in dataset. The dataset consists of 50 samples from each of three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor). Four features(variables) were measured from each sample, they are the **length** and the **width** of sepal and petal, in centimetres. Perform the following EDA steps .

(i)Find dimension, Structure, Summary statistics, Standard Deviation of all features.

(ii)Find mean and standard deviation of features groped by three species of Iris flowers (Iris setosa, Iris virginica and Iris versicolor)

(iii)Find quantile value of sepal width and length

(iV)create new data frame named iris1 which have a new column name **Sepal.Length.Cate** that categorizes “Sepal.Length” by quantile

(V) Average value of numerical varialbes by two categorical variables: Species and Sepal.Length.Cate:

(vi) Average mean value of numerical varialbes by Species and Sepal.Length.Cate

(vii)Create Pivot Table based on Species and Sepal.Length.Cate.

CODE:

data(iris)

#subdiv1

dim(iris)

str(iris)

summary(iris)

apply(iris[, 1:4], 2, sd)

#subdiv2

aggregate(iris[, 1:4], by = list(Species = iris$Species), FUN = mean)

aggregate(iris[, 1:4], by = list(Species = iris$Species), FUN = sd)

#subdiv3

quantile(iris$Sepal.Length)

quantile(iris$Sepal.Width)

#subdiv4

iris1 <- iris

iris1$Sepal.Length.Cate <- cut(iris1$Sepal.Length, breaks = quantile(iris1$Sepal.Length), include.lowest = TRUE)

head(iris1)

#subdiv5

aggregate(iris1[, 1:4], by = list(Species = iris1$Species, Sepal.Length.Cate = iris1$Sepal.Length.Cate), FUN = mean)

#subdiv6

library(dplyr)

iris1 %>%

group\_by(Species, Sepal.Length.Cate) %>%

summarise(across(everything(), mean))

#subdiv7

library(tidyr)

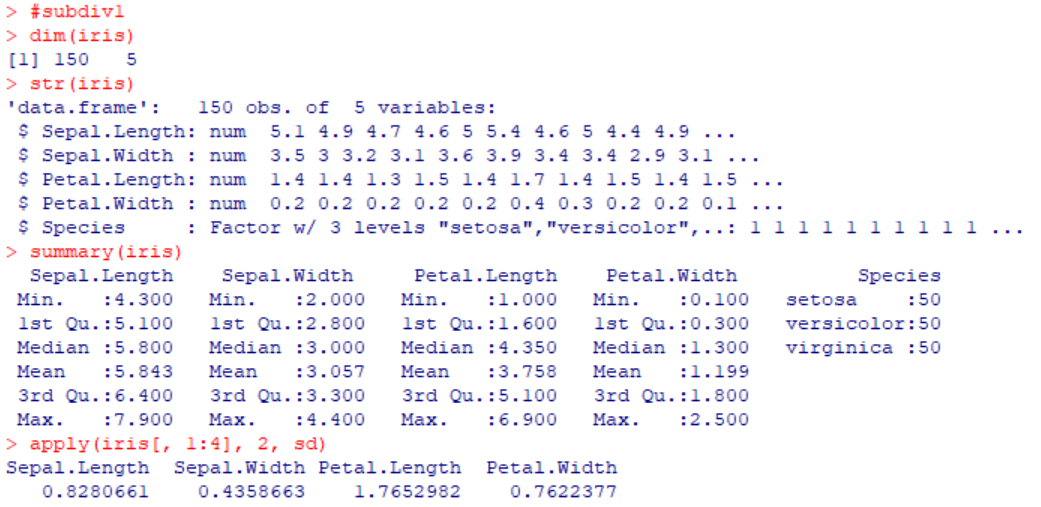
pivot\_table <- pivot\_wider(iris1, names\_from = Sepal.Length.Cate, values\_from = c(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width))

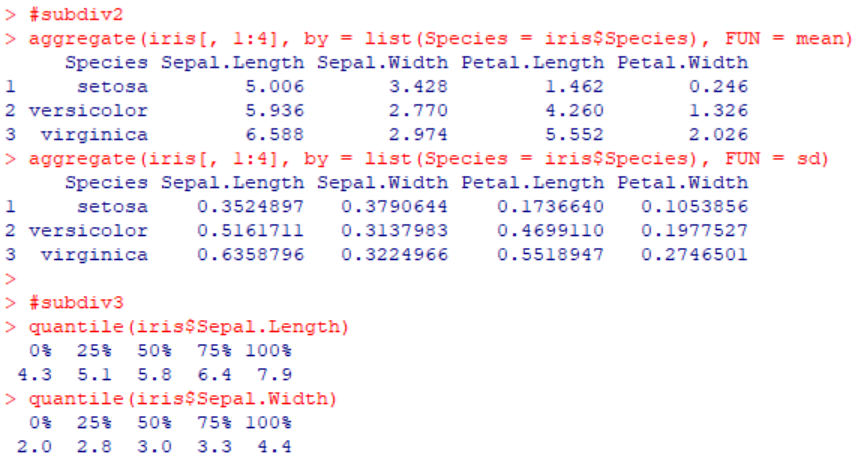
pivot\_table %>%

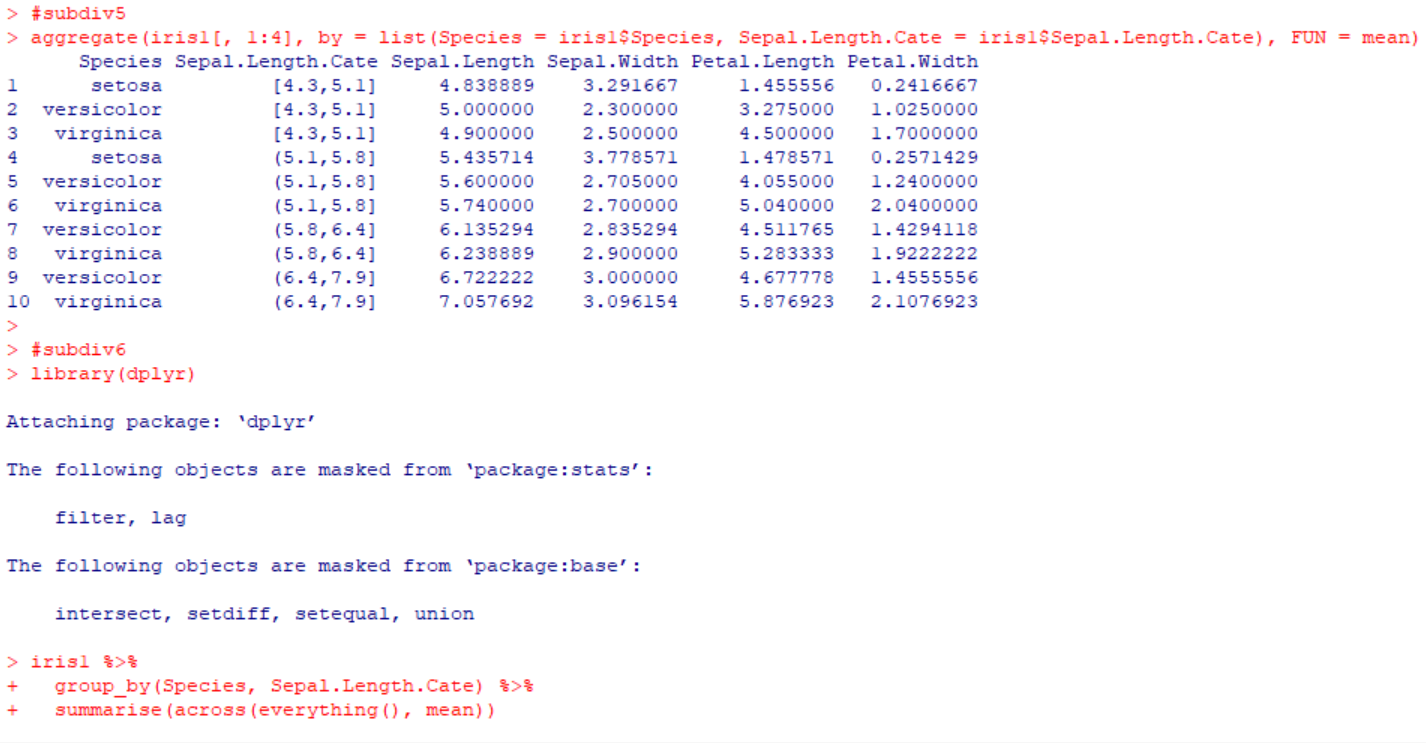
group\_by(Species) %>%

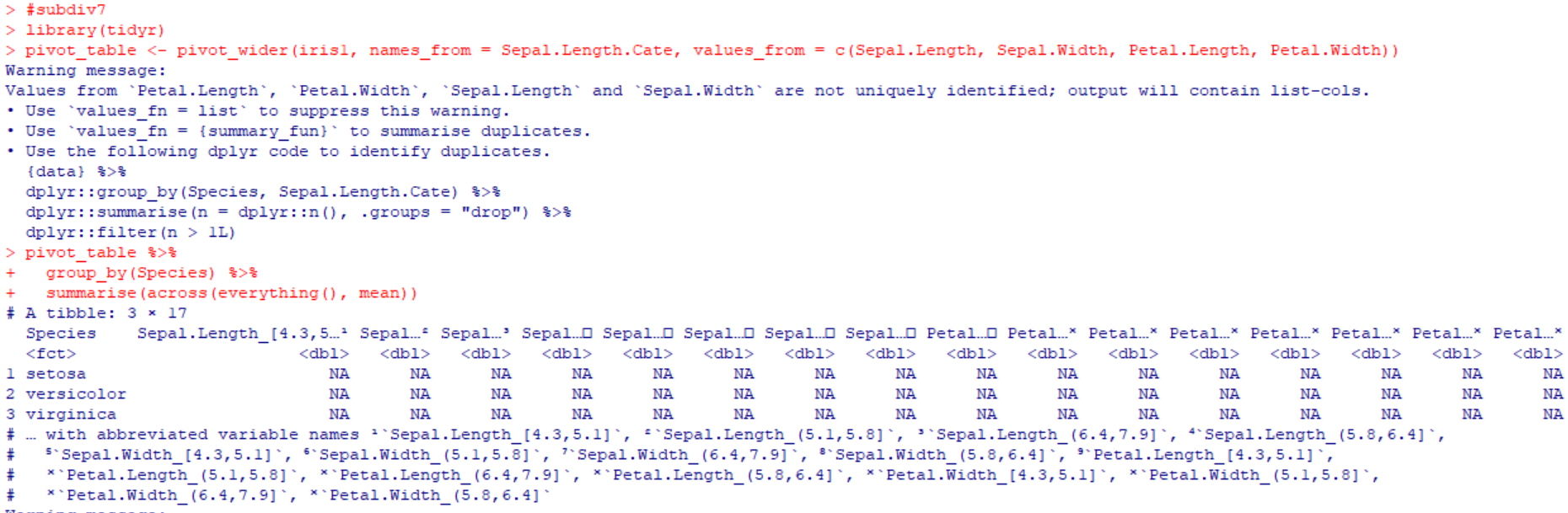
summarise(across(everything(), mean))

OUTPUT:









1. Randomly Sample the iris dataset such as 80% data for training and 20% for test and create Logistics regression with train data, use species as target and petals width and length as feature variables , Predict the probability of the model using test data, Create Confusion matrix for above test model CODE:

data(iris)

set.seed(123)

trainIndex <- sample(1:nrow(iris), 0.8\*nrow(iris))

trainData <- iris[trainIndex, ]

testData <- iris[-trainIndex, ]

library(nnet)

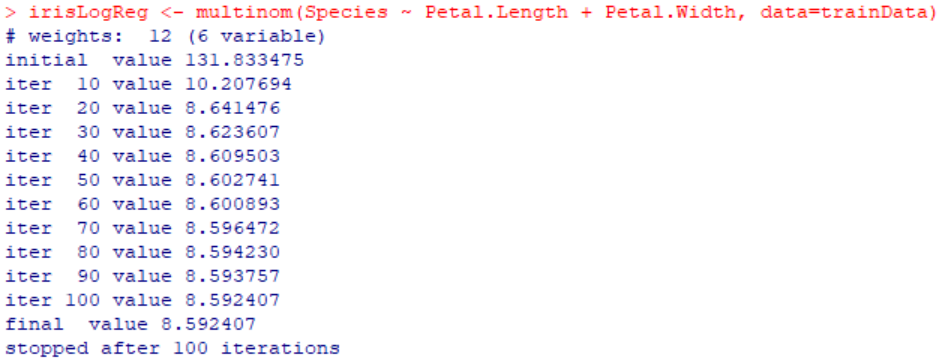
irisLogReg <- multinom(Species ~ Petal.Length + Petal.Width, data=trainData)

testData$predicted <- predict(irisLogReg, newdata=testData, type="prob")

library(caret)

confusionMatrix(data=testData$Species, reference=testData$predicted)

OUTPUT:



1. (i)Write suitable R code to compute the mean, median, mode of the following values c (90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20) (ii) Write R code to find 2nd highest and 3rd Lowest value of above problem.

CODE:

#subdiv1

x <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20)

mean(x)

median(x)

install.packages("DescTools")

library(DescTools)

Mode(x)

#subdiv2

x <- c(90, 50, 70, 80, 70, 60, 20, 30, 80, 90, 20)

sort(x, decreasing = TRUE)[2]

sort(x)[3]

OUTPUT:



1. Explore the air quality dataset. It contains daily air quality measurements from New York during a period of five months: • Ozone: mean ozone concentration (ppb), • Solar.R: solar radiation (Langley), • Wind: average wind speed (mph), • Temp: maximum daily temperature in degrees Fahrenheit, • Month: numeric month (May=5, June=6, and so on), • Day: numeric day of the month (1-31). i. Compute the mean temperature (don’t use build in function)

ii. Extract the first five rows from air quality.

iii. Extract all columns from air quality except Temp and Wind

iv. Which was the coldest day during the period?

v. How many days was the wind speed greater than 17 mph?

CODE:

#subdiv1

data(airquality)

mean\_temp <- sum(airquality$Temp)/length(airquality$Temp)

mean\_temp

#subdiv2

data(airquality)

airquality[1:5,]

#subdiv3

data(airquality)

airquality[, !(names(airquality) %in% c("Temp", "Wind"))]

#subdiv4

data(airquality)

coldest\_day <- airquality[which.min(airquality$Temp),]

coldest\_day

#subdiv5

data(airquality)

n\_days\_windy <- sum(airquality$Wind > 17, na.rm = TRUE)

n\_days\_windy

OUTPUT:



