PRATEEK MOHANTY

20BCE1482

LAB-5

Data Visualization

CSE3020

Q1

CODE

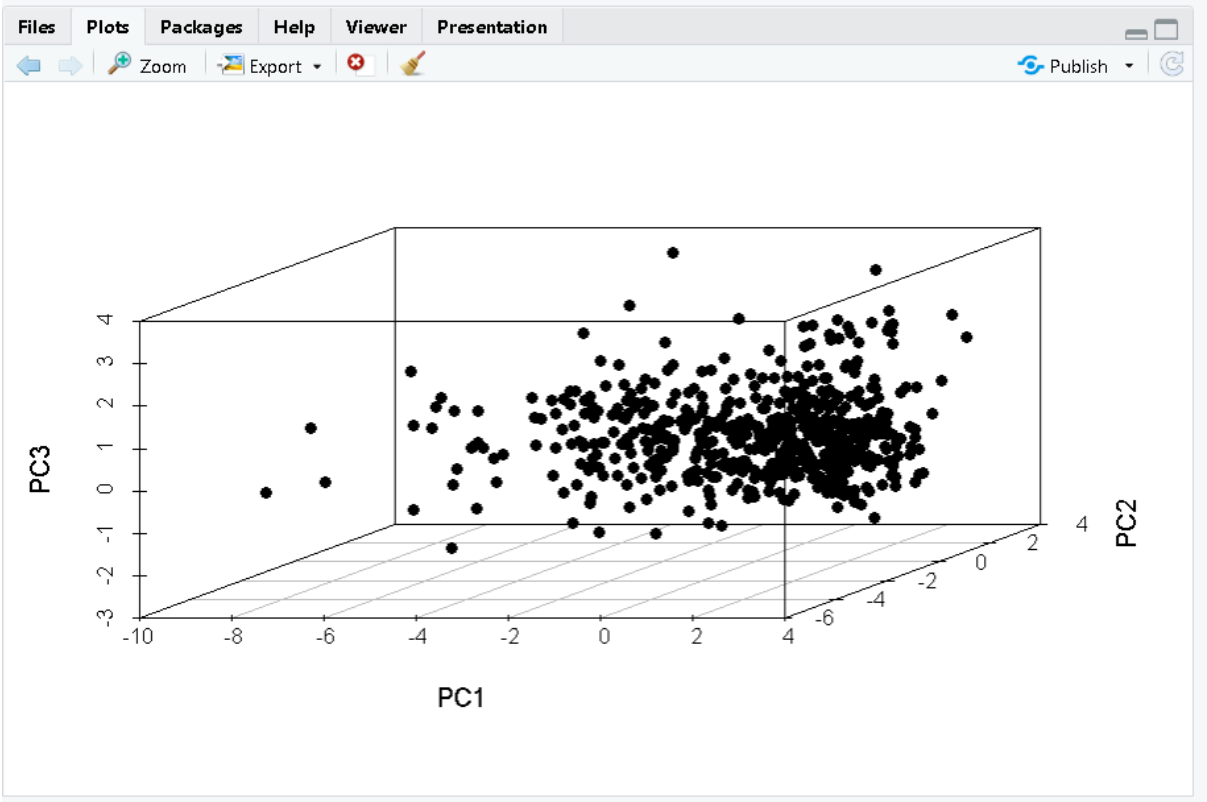
```
#Q1

#i)

library("dplyr")

data <- read.csv('data.csv')

data

# Load the dataset from the CSV file

data <- read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data", header = FALSE)

# Extract the feature variables and the target variable

feature_vars <- data[, 2:11]

target_var <- data[, 1]

# Run PCA on the feature variables

pca_result <- prcomp(select_if(feature_vars,is.numeric), scale = TRUE)
```

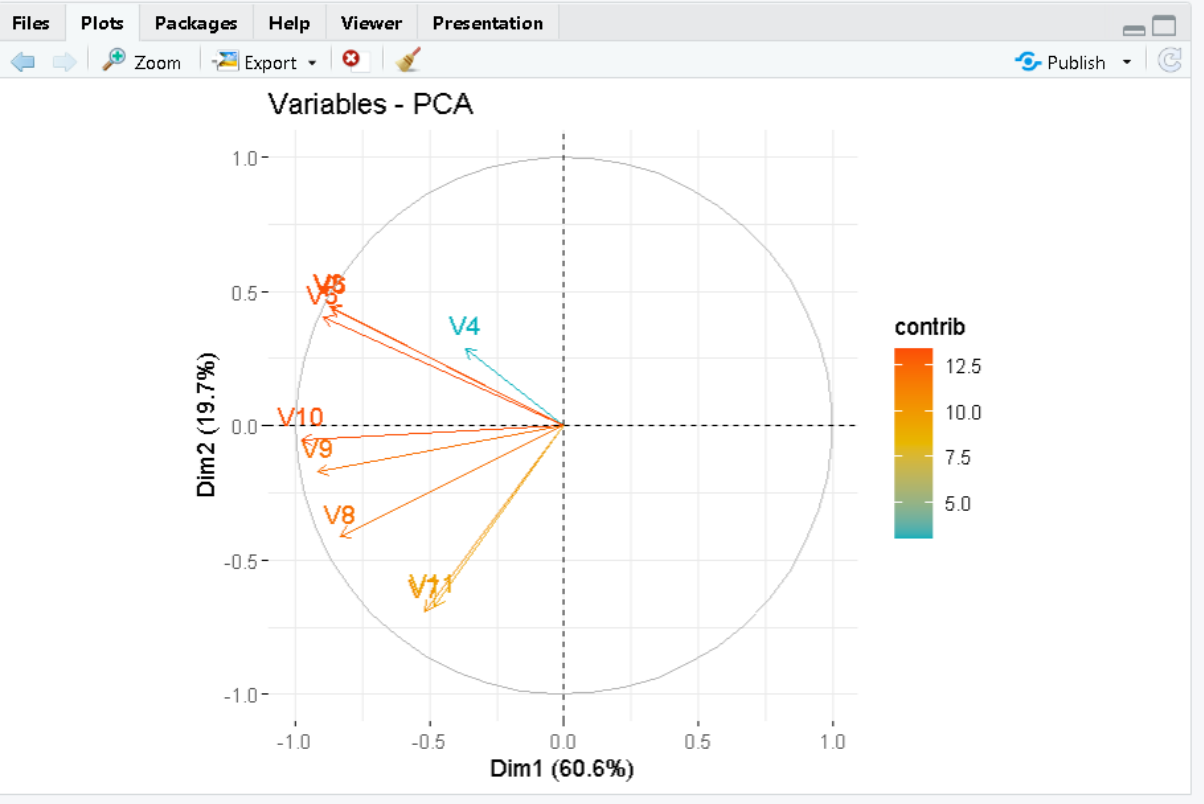
```
pca_result
# Plot the PCA results
library(scatterplot3d)
scatterplot3d(pca_result$x[,1],pca_result$x[,2],pca_result$x[,3],
              xlab="PC1",ylab="PC2",zlab="PC3",pch=19)
library(factoextra)
fviz_pca_var(pca_result, col.var = "contrib", gradient.cols =
c("#00AFBB", "#E7B800", "#FC4E07"))  # ii)
# Load the required libraries
library(caret)
library(tidyverse)
# Load the data
wdbc <- read.csv('data.csv')
for (x in wdbc$diagnosis)
{
  if(x=='M')wdbc$diagnosis[k]=0
  else wdbc$diagnosis[k]=1;
  k=k+1;
}
wdbc$diagnosis<-as.numeric(wdbc$diagnosis)
wdbc<-na.omit(wdbc)
sum(is.na(wdbc))
# Check the number of rows in the dataset
nrow(wdbc)
```

```
# Split the data into training and testing sets
set.seed(123)
split_index <- createDataPartition(wdbc$diagnosis, p = 0.7, list = FALSE)
train_data <- wdbc[split_index, ]
test_data <- wdbc[-split_index, ]
# Check the number of rows in the training and testing sets
nrow(train_data)
nrow(test_data)
# Check the number of rows in the diagnosis variable
nrow(train_data$diagnosis)
# Perform PCA on the training data
pca_res <- prcomp(train_data[, -1], center = TRUE, scale. = TRUE)
# Construct a model using the first 6 principal components
model1 <- train(diagnosis ~ pca_res$x[, 1:6], data = train_data, method = "glm")
# Predict the diagnosis using the model and the test data
predictions1 <- predict(model1, newdata = test_data)
# Evaluate the model performance
confusionMatrix(predictions1, test_data$diagnosis)
# Construct a model using the original 30 variables
model2 <- train(diagnosis ~ ., data = train_data, method = "glm")
# Predict the diagnosis using the model and the test data
predictions2 <- predict(model2, newdata = test_data)
```

```
# Evaluate the model performance
confusionMatrix(predictions2, test_data$diagnosis)
# Compare the performance of the two models
cmp <- resamples(list(model1, model2))
summary(cmp)
```

OUTPUT





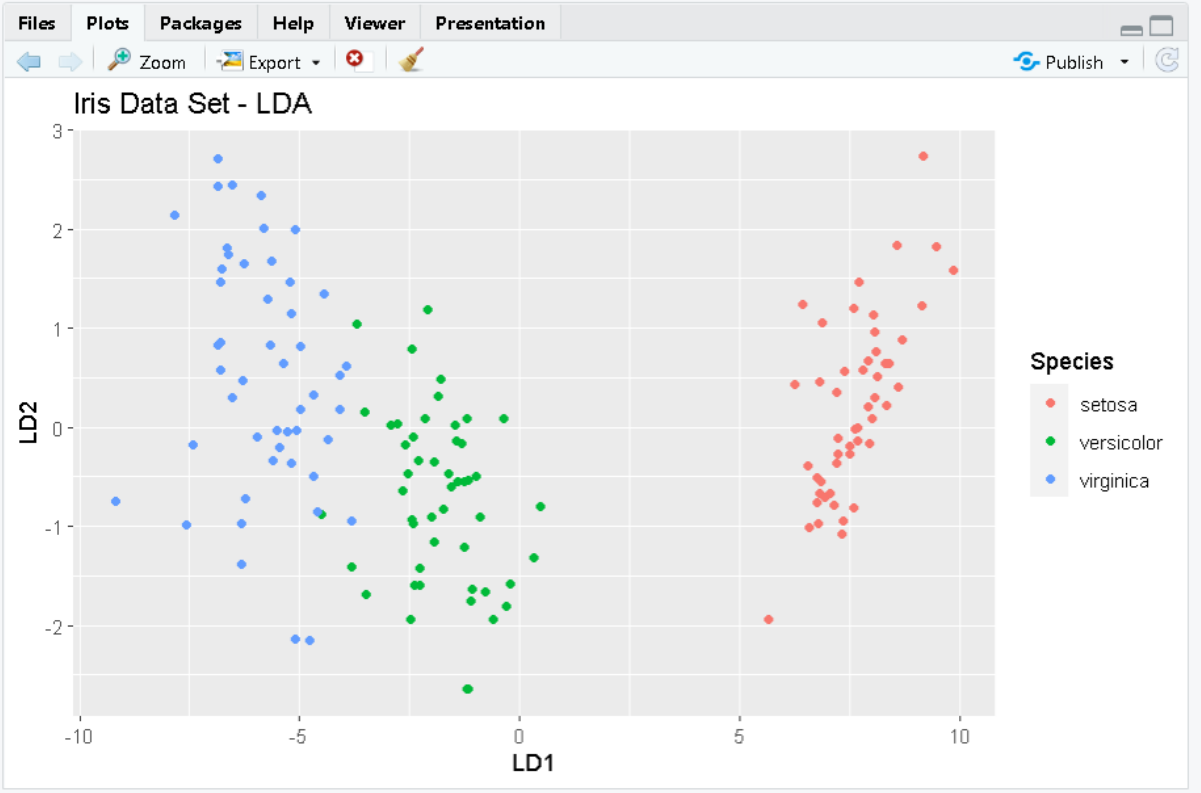
Q2

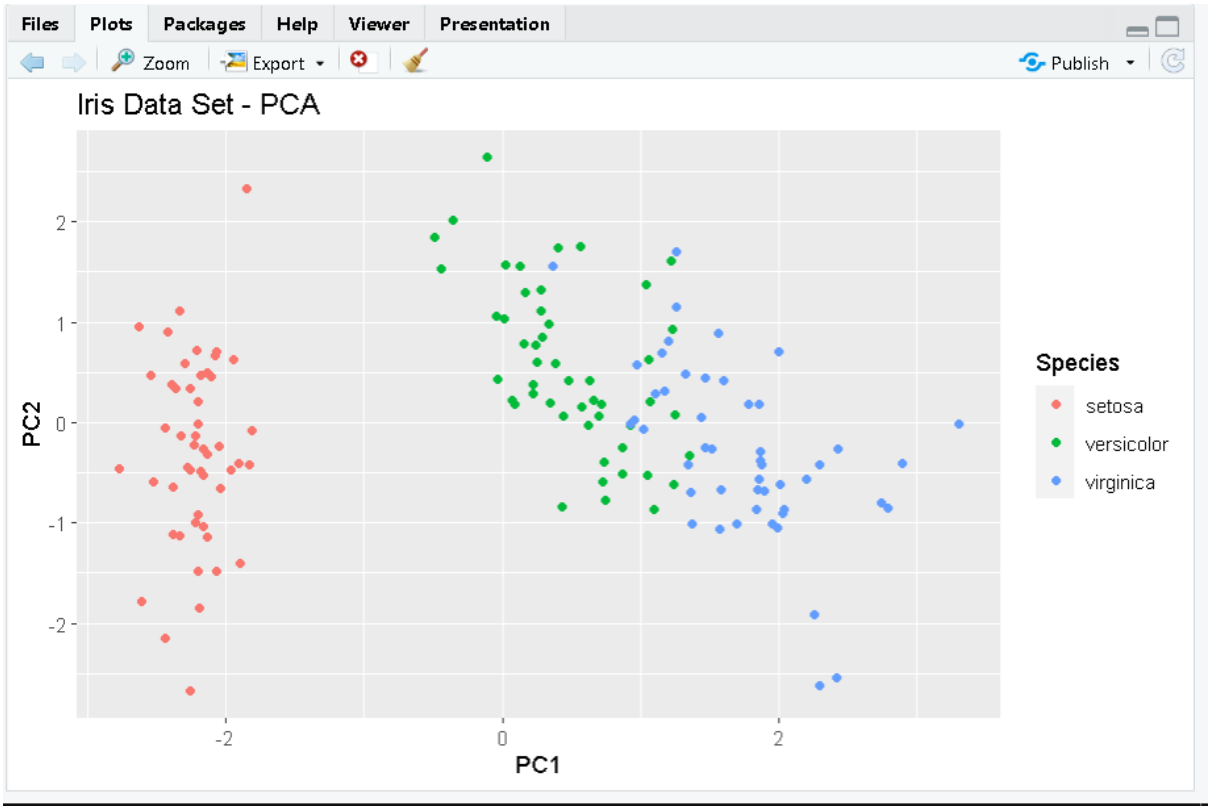
CODE

```
library(MASS)

data(iris)# Perform LDA

iris.lda <- lda(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width, data = iris)# Access the Linear Discriminants

iris$LD1 <- predict(iris.lda)$x[,1]

iris$LD2 <- predict(iris.lda)$x[,2]

ggplot(iris, aes(x = LD1, y = LD2, color = Species)) + geom_point() +ggtitle("Iris Data Set - LDA") +xlab("LD1") + ylab("LD2")

# Perform PCA
```

```
iris.pca <- prcomp(iris[,1:4], scale = TRUE)# Access the Principal Components

iris$PC1 <- iris.pca$x[,1]

iris$PC2 <- iris.pca$x[,2]

library(ggplot2)

ggplot(iris, aes(x = PC1, y = PC2, color = Species)) + geom_point() +ggtitle("Iris Data Set - PCA") +xlab("PC1") + ylab("PC2")
```

OUTPUT




Files
Plots
Packages
Help
Viewer
Presentation
Zoom
Export
Publish
Iris Data Set - PCA
PC2
PC1
2
1
0
-1
-2
-2
0
2
Species
setosa
versicolor
virginica