MAT iris

Sofian Hamiti

Introduction

The Iris dataset contains 150 instances, corresponding to three equally-frequent species of iris plant (Iris setosa, Iris versicolour, and Iris virginica).

This project has been built in 2 parts, using R:

- This presentation document
- A ShinyApp to retrieve the 10 most similar observations to the inputted vector

The Data

Loading the data:

```
library("ggplot2")
library("gridExtra")
source("cleandata.R")

#Load the dataset from the website and assign names to columns
iris <- read.csv(url("http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"))
colnames(iris) <- c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width", "Species")
```

Simple data quality assessment

```
#Number of NA values in the dataset
sum(is.na(iris))
```

```
## [1] 0
```

```
#Show the duplicate values
subset(iris, duplicated(iris))
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width        Species
## 34           4.9         3.1          1.5         0.1    Iris-setosa
## 37           4.9         3.1          1.5         0.1    Iris-setosa
## 142          5.8         2.7          5.1         1.9 Iris-virginica
```

```
#Call the cleanData function in order to remove NAs, Duplicates, and Negative values.
iris <- cleanData(iris)
```

Analysis of the Tidy dataset

```
#Number of observations and the attributes' data types
str(iris)
```

```
## 'data.frame':    146 obs. of  5 variables:
##  $ Sepal.Length: num  4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 5.4 ...
##  $ Sepal.Width : num  3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 3.7 ...
##  $ Petal.Length: num  1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 0.2 ...
##  $ Species     : Factor w/ 3 levels "Iris-setosa",..: 1 1 1 1 1 1 1 1 1 ...
```

```
#Distribution of species
table(iris$Species)
```

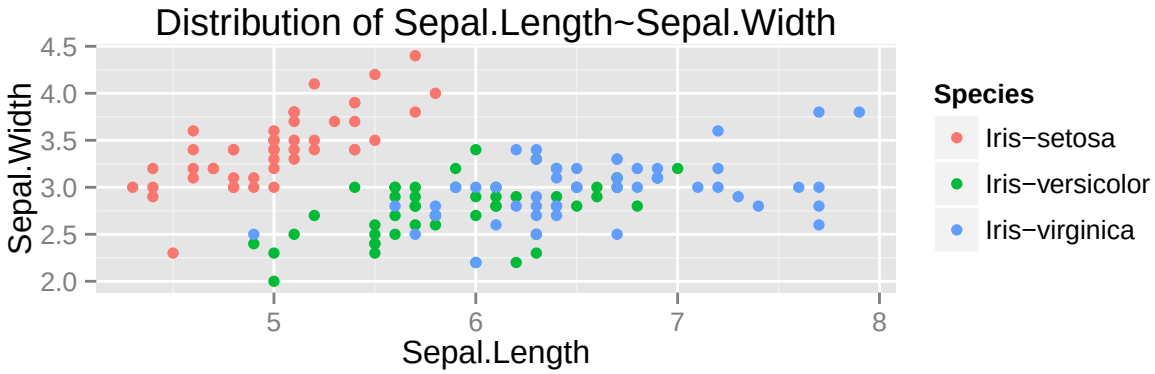
```
##
##     Iris-setosa Iris-versicolor  Iris-virginica
##              47              50              49
```

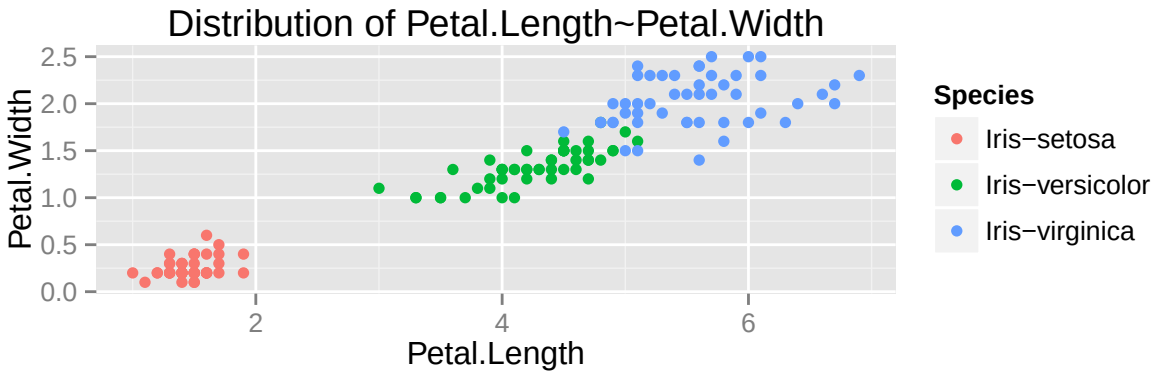
```
#Summary statistics of the flowers' specs from which
#I have defined the extreme values to input in the shinyapp.
summary(iris[,-5])
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.400   Median :1.300
##  Mean   :5.862   Mean   :3.053   Mean   :3.797   Mean   :1.216
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

Exploratory analysis

```
#As we can see in the charts, in future developments, it may be interesting to
#cluster the observations using their Petal specs.
```





The ShinyApp

The app has been built using R and has the following functionalities:

- Reads in the data, accessed from: https://archive.ics.uci.edu/ml/datasets/Iris;
- Clean the data using the cleanData function.
- Takes as arguments, attributes defining a new species of Iris plant with the use of sliders. Sliders helped me to protect the app from unrealistic inputs.
- Returns the ten most similar data points in the existing Iris data based on the euclidean distance from the inputted vector.
- Displays the results in a table.

The code is available on my github repository: https://github.com/SofianHamiti/MAT_iris

The ShinyApp: https://sofianhamiti.shinyapps.io/MAT_iris

Future Possible Improvements

- Improve the accuracy of the calculation. Cosine Similarity may be a good start of research.
- Visualize the output in a plot to have a more intuitive vision of the distance calculated.