Data Analysis on Iris Flowers

Sridhar

7 January 2018

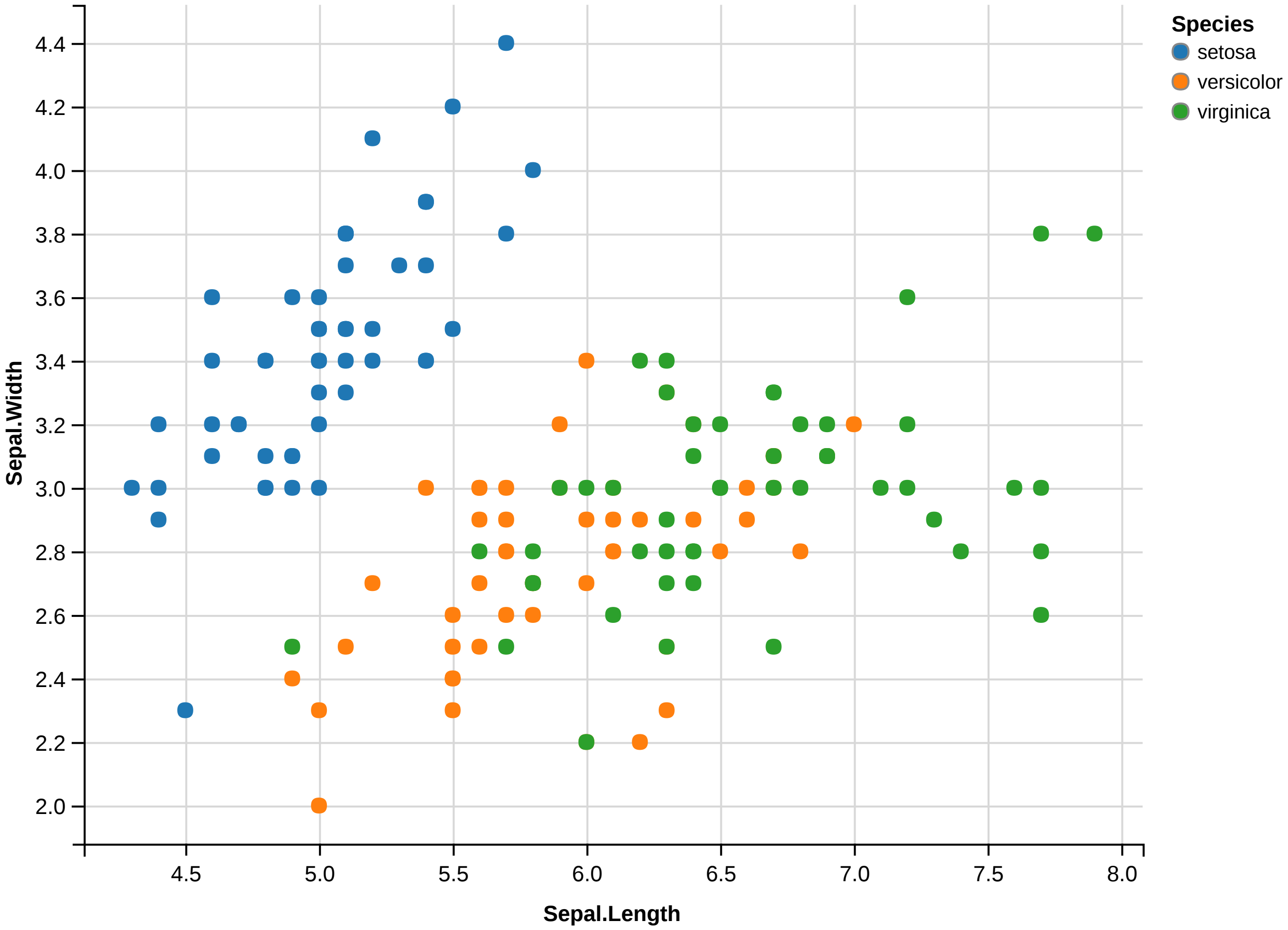
Load the data

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

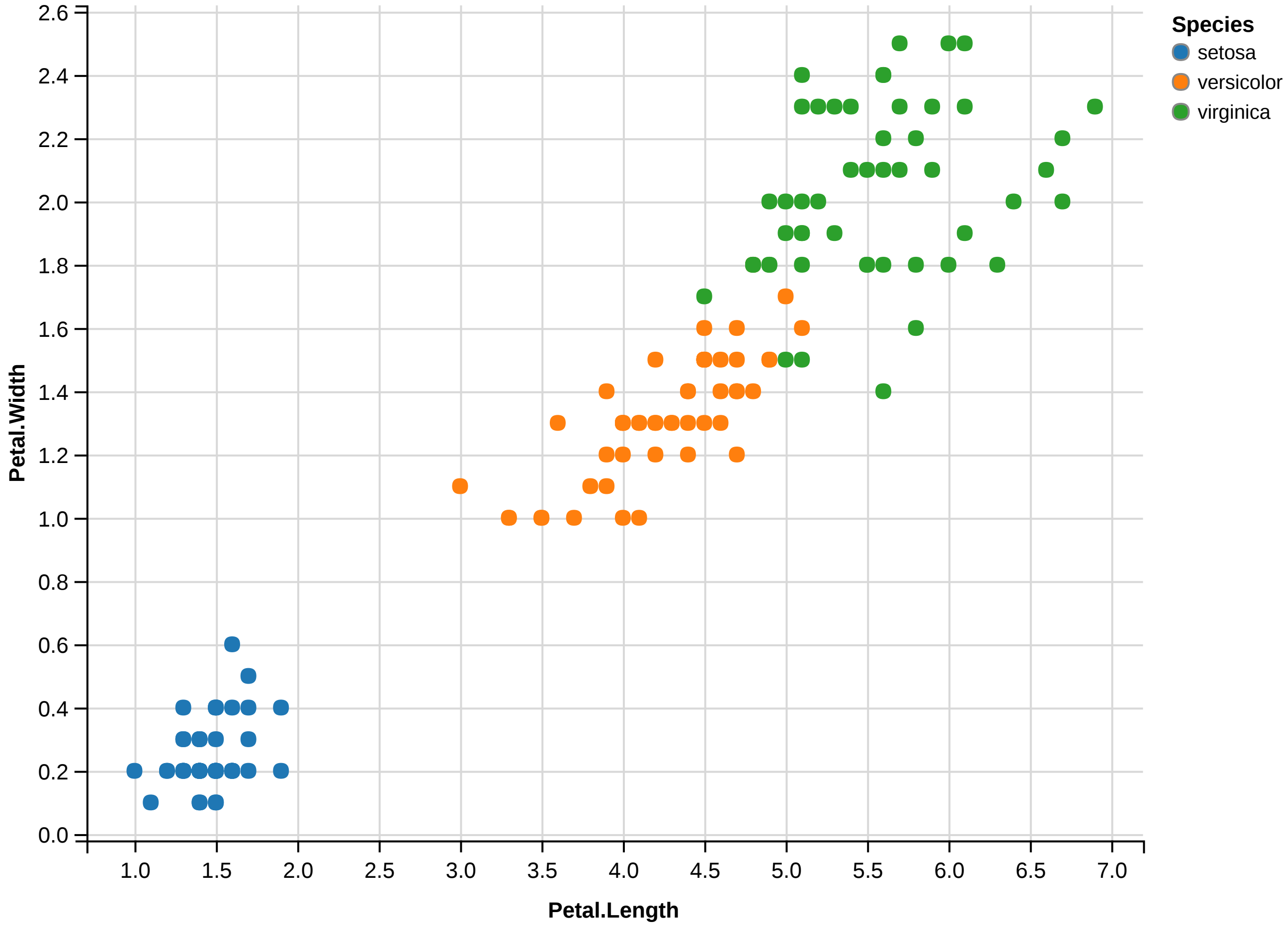
Scatterplot

By using scatterplots,we can find how much the parameters are correlated



The Sepal Length and Sepal width are some what correlated but not that much,we can see that the setosa, is completely separated since they have small sepal length and small sepal width than other species.But the real problem is that the virgincia,versicolor species were mixed apart.Hence we move to the next parameters.

```
iris %>% ggvis(~Petal.Length,~Petal.Width,fill = ~Species) %>% layer_points()
```



Check this,this scatterplot is pretty good,which separates the species and forms a perfect correlation line.

Correlations

Let’s check the numerical correlations of the parameters

```
print(cor(iris$Sepal.Length,iris$Sepal.Width))
```

```
## [1] -0.1175698
```

```
print(cor(iris$Petal.Length,iris$Petal.Width))
```

```
## [1] 0.9628654
```

Correlation matrix

For each property the correlations are identified for different species i.e, sentosa,versicolor,virginica

```
type <- levels(iris$Species)
print(type[1])
```

```
## [1] "setosa"
```

```
cor(iris[iris$Species==type[1],1:4])
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000   0.7425467    0.2671758   0.2780984
## Sepal.Width     0.7425467   1.0000000    0.1777000   0.2327520
## Petal.Length    0.2671758   0.1777000    1.0000000   0.3316300
## Petal.Width     0.2780984   0.2327520    0.3316300   1.0000000
```

```
print(type[2])
```

```
## [1] "versicolor"
```

```
cor(iris[iris$Species==type[3],1:4])
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000   0.4572278    0.8642247   0.2811077
## Sepal.Width     0.4572278   1.0000000    0.4010446   0.5377280
## Petal.Length    0.8642247   0.4010446    1.0000000   0.3221082
## Petal.Width     0.2811077   0.5377280    0.3221082   1.0000000
```

```
print(type[3])
```

```
## [1] "virginica"
```

```
cor(iris[iris$Species==type[3],1:4])
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000   0.4572278    0.8642247   0.2811077
## Sepal.Width     0.4572278   1.0000000    0.4010446   0.5377280
## Petal.Length    0.8642247   0.4010446    1.0000000   0.3221082
## Petal.Width     0.2811077   0.5377280    0.3221082   1.0000000
```

Knowing the data

```
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Structure of the data

```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1
 ...
```

Tabulations

```
table(iris$Species)
```

```
## 
##     setosa versicolor  virginica 
##         50         50         50
```

```
round(prop.table(table(iris$Species)) * 100, digits = 1)
```

```
## 
##     setosa versicolor  virginica 
##       33.3       33.3       33.3
```

```
summary(iris)
```

```
##   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 :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  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  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
## 
```

```
summary(iris[c("Petal.Width","Sepal.Width")])
```

```
##   Petal.Width     Sepal.Width   
##  Min.   :0.100   Min.   :2.000  
##  1st Qu.:0.300   1st Qu.:2.800  
##  Median :1.300   Median :3.000  
##  Mean   :1.199   Mean   :3.057  
##  3rd Qu.:1.800   3rd Qu.:3.300  
##  Max.   :2.500   Max.   :4.400
```

Normalization

The normalization/feature scaling is not necessary but still,it improves the accuracy of this classification system.Here normalization process makes all the columns to be in the range of 0 to 1.

```
library(class)
normalize <- function(x) {
num <- x - min(x)
denom <- max(x) - min(x)
return (num/denom)
}


iris_norm <- as.data.frame(lapply(iris[1:4], normalize))


summary(iris_norm)
```

```
##   Sepal.Length     Sepal.Width      Petal.Length     Petal.Width     
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000  
##  1st Qu.:0.2222   1st Qu.:0.3333   1st Qu.:0.1017   1st Qu.:0.08333  
##  Median :0.4167   Median :0.4167   Median :0.5678   Median :0.50000  
##  Mean   :0.4287   Mean   :0.4406   Mean   :0.4675   Mean   :0.45806  
##  3rd Qu.:0.5833   3rd Qu.:0.5417   3rd Qu.:0.6949   3rd Qu.:0.70833  
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
```

Training and Testing sets

The dataset is divided into two parts 1) Training set : To train the classifier,it contains 2/3 of the dataset. 2) Testing set : To test the classifier,it contains 1/3 of the dataset.

So for the division purpose we need random rows,that’s why we are using seed() method.

```
set.seed(1234)
ind <- sample(2, nrow(iris), replace=TRUE, prob=c(0.67, 0.33))
ind
```

```
##   [1] 1 1 1 1 2 1 1 1 1 1 2 1 1 2 1 2 1 1 1 1 1 1 1 1 1 2 1 2 2 1 1 1 1 1 1
##  [36] 2 1 1 2 2 1 1 1 1 1 1 2 1 1 2 1 1 2 1 1 1 1 2 1 2 2 1 1 1 1 2 1 1 1 1
##  [71] 1 2 1 2 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 2 1 1 1 1 1 1 1 2 1 1 1 1 1
## [106] 1 1 1 1 1 2 1 2 1 1 2 2 1 1 2 2 2 2 2 1 1 1 1 1 1 2 1 1 1 2 1 2 1 1 2
## [141] 1 2 1 1 1 1 2 1 2 1
```

```
iris.training <- iris[ind==1, 1:4]

head(iris.training)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1          5.1         3.5          1.4         0.2
## 2          4.9         3.0          1.4         0.2
## 3          4.7         3.2          1.3         0.2
## 4          4.6         3.1          1.5         0.2
## 6          5.4         3.9          1.7         0.4
## 7          4.6         3.4          1.4         0.3
```

```
iris.test <- iris[ind==2, 1:4]

head(iris.test)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width
## 5           5.0         3.6          1.4         0.2
## 11          5.4         3.7          1.5         0.2
## 14          4.3         3.0          1.1         0.1
## 16          5.7         4.4          1.5         0.4
## 26          5.0         3.0          1.6         0.2
## 28          5.2         3.5          1.5         0.2
```

Here the data is being separated!with the above found random possibilities.

```
iris.trainLabels <- iris[ind==1,5]

print(iris.trainLabels)
```

```
##   [1] setosa     setosa     setosa     setosa     setosa     setosa
##   [7] setosa     setosa     setosa     setosa     setosa     setosa
##  [13] setosa     setosa     setosa     setosa     setosa     setosa
##  [19] setosa     setosa     setosa     setosa     setosa     setosa
##  [25] setosa     setosa     setosa     setosa     setosa     setosa
##  [31] setosa     setosa     setosa     setosa     setosa     setosa
##  [37] setosa     setosa     versicolor versicolor versicolor versicolor
##  [43] versicolor versicolor versicolor versicolor versicolor versicolor
##  [49] versicolor versicolor versicolor versicolor versicolor versicolor
##  [55] versicolor versicolor versicolor versicolor versicolor versicolor
##  [61] versicolor versicolor versicolor versicolor versicolor versicolor
##  [67] versicolor versicolor versicolor versicolor versicolor versicolor
##  [73] versicolor versicolor versicolor versicolor virginica  virginica
##  [79] virginica  virginica  virginica  virginica  virginica  virginica
##  [85] virginica  virginica  virginica  virginica  virginica  virginica
##  [91] virginica  virginica  virginica  virginica  virginica  virginica
##  [97] virginica  virginica  virginica  virginica  virginica  virginica
## [103] virginica  virginica  virginica  virginica  virginica  virginica
## [109] virginica  virginica
## Levels: setosa versicolor virginica
```

```
iris.testLabels <- iris[ind==2, 5]

print(iris.testLabels)
```

```
##  [1] setosa     setosa     setosa     setosa     setosa     setosa    
##  [7] setosa     setosa     setosa     setosa     setosa     setosa    
## [13] versicolor versicolor versicolor versicolor versicolor versicolor
## [19] versicolor versicolor versicolor versicolor versicolor versicolor
## [25] virginica  virginica  virginica  virginica  virginica  virginica 
## [31] virginica  virginica  virginica  virginica  virginica  virginica 
## [37] virginica  virginica  virginica  virginica 
## Levels: setosa versicolor virginica
```

Classification

Here the k-Nearest Neighbour Classification is applied,with the training set and the testing set and the species were predicted.The knn() method does a good job by predicting the species based on the training set and they were tested by the testing set.

```
iris_pred <- knn(train = iris.training, test = iris.test, cl = iris.trainLabels, k=3)
iris_pred
```

```
##  [1] setosa     setosa     setosa     setosa     setosa     setosa    
##  [7] setosa     setosa     setosa     setosa     setosa     setosa    
## [13] versicolor versicolor versicolor versicolor versicolor versicolor
## [19] versicolor versicolor versicolor versicolor versicolor versicolor
## [25] virginica  virginica  virginica  virginica  versicolor virginica 
## [31] virginica  virginica  virginica  virginica  virginica  virginica 
## [37] virginica  virginica  virginica  virginica 
## Levels: setosa versicolor virginica
```

Comparison

We need to make sure that our classifier has classified the species correctly,in order to do that we merge the real species name and the predicted name.As a result we find something unsual.

```
irisTestLabels <- data.frame(iris.testLabels)

merge <- data.frame(iris_pred, iris.testLabels)

names(merge) <- c("Predicted Species", "Observed Species")

merge
```

```
##    Predicted Species Observed Species
## 1             setosa           setosa
## 2             setosa           setosa
## 3             setosa           setosa
## 4             setosa           setosa
## 5             setosa           setosa
## 6             setosa           setosa
## 7             setosa           setosa
## 8             setosa           setosa
## 9             setosa           setosa
## 10            setosa           setosa
## 11            setosa           setosa
## 12            setosa           setosa
## 13        versicolor       versicolor
## 14        versicolor       versicolor
## 15        versicolor       versicolor
## 16        versicolor       versicolor
## 17        versicolor       versicolor
## 18        versicolor       versicolor
## 19        versicolor       versicolor
## 20        versicolor       versicolor
## 21        versicolor       versicolor
## 22        versicolor       versicolor
## 23        versicolor       versicolor
## 24        versicolor       versicolor
## 25         virginica        virginica
## 26         virginica        virginica
## 27         virginica        virginica
## 28         virginica        virginica
## 29        versicolor        virginica
## 30         virginica        virginica
## 31         virginica        virginica
## 32         virginica        virginica
## 33         virginica        virginica
## 34         virginica        virginica
## 35         virginica        virginica
## 36         virginica        virginica
## 37         virginica        virginica
## 38         virginica        virginica
## 39         virginica        virginica
## 40         virginica        virginica
```

The classifier did a small mistake i.e, instead of versicolor,it predicted as virginica. This k-NN classification is not 100 % percent accurate.

Proper summary

```
library(gmodels)
CrossTable(x = iris.testLabels, y = iris_pred, prop.chisq=FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  40
##
##
##                 | iris_pred
## iris.testLabels |     setosa | versicolor |  virginica |  Row Total |
## ----------------|------------|------------|------------|------------|
##          setosa |         12 |          0 |          0 |         12 |
##                 |      1.000 |      0.000 |      0.000 |      0.300 |
##                 |      1.000 |      0.000 |      0.000 |            |
##                 |      0.300 |      0.000 |      0.000 |            |
## ----------------|------------|------------|------------|------------|
##      versicolor |          0 |         12 |          0 |         12 |
##                 |      0.000 |      1.000 |      0.000 |      0.300 |
##                 |      0.000 |      0.923 |      0.000 |            |
##                 |      0.000 |      0.300 |      0.000 |            |
## ----------------|------------|------------|------------|------------|
##       virginica |          0 |          1 |         15 |         16 |
##                 |      0.000 |      0.062 |      0.938 |      0.400 |
##                 |      0.000 |      0.077 |      1.000 |            |
##                 |      0.000 |      0.025 |      0.375 |            |
## ----------------|------------|------------|------------|------------|
##    Column Total |         12 |         13 |         15 |         40 |
##                 |      0.300 |      0.325 |      0.375 |            |
## ----------------|------------|------------|------------|------------|
##
##
```