

Untitled

ShivaniR

11/3/2020

PROBLEM STATEMENT

From the given 'Iris' dataset, predict the optimum number of clusters and represent it visually.

Assumption: To not take species column to form clusters and then use it to check our model performance

```
library(ggplot2)
```

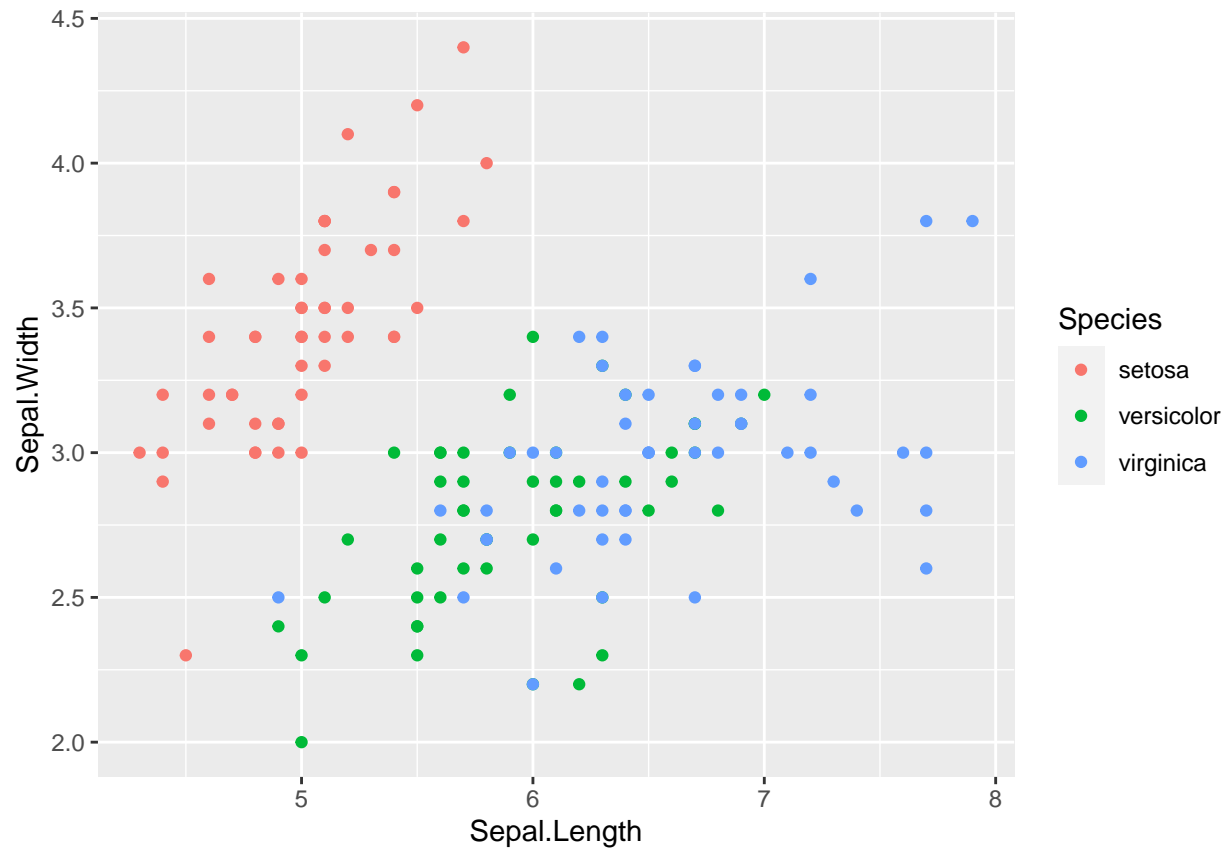
DATA

```
df <- iris  
head(iris)
```

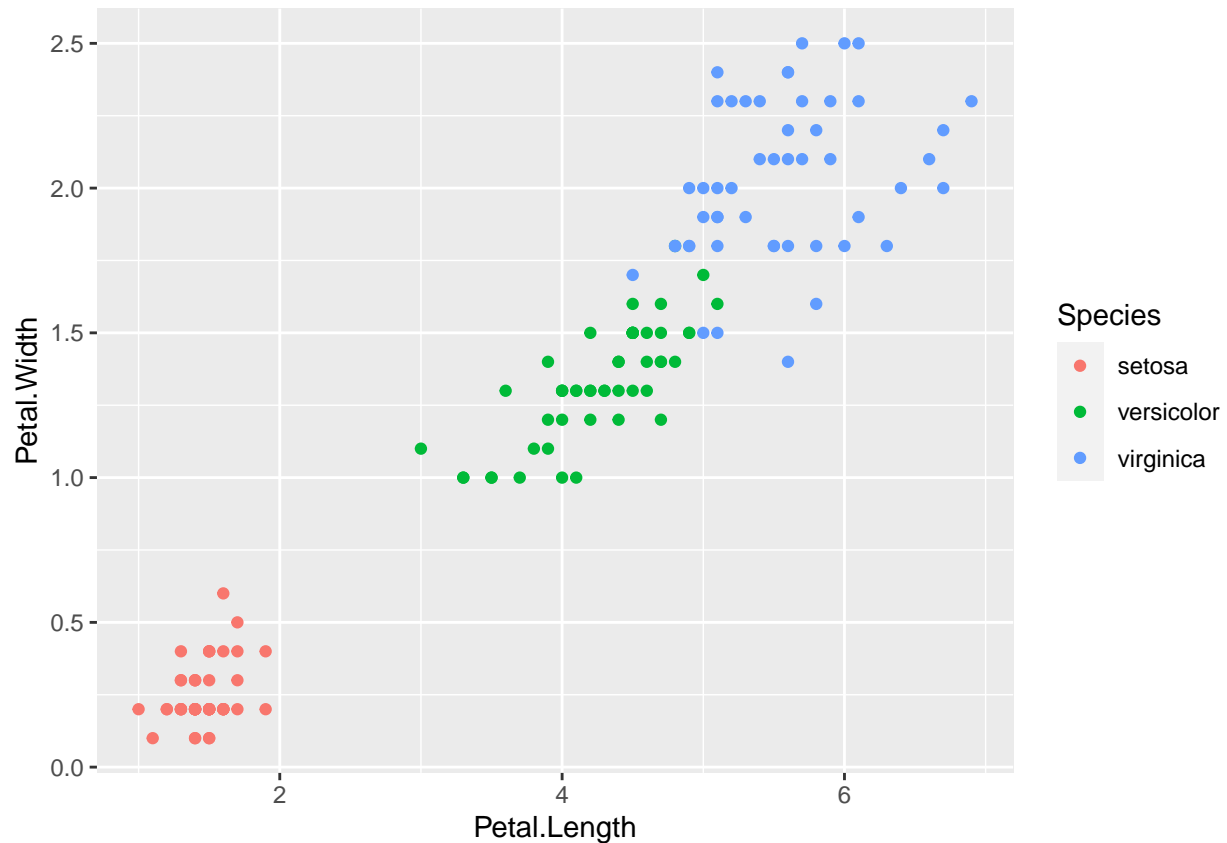
##	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
## 1	5.1	3.5	1.4	0.2	setosa
## 2	4.9	3.0	1.4	0.2	setosa
## 3	4.7	3.2	1.3	0.2	setosa
## 4	4.6	3.1	1.5	0.2	setosa
## 5	5.0	3.6	1.4	0.2	setosa
## 6	5.4	3.9	1.7	0.4	setosa

EXPLORATORY DATA ANALYSIS

```
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, col = Species)) + geom_point()
```



```
ggplot(iris,aes(x = Petal.Length, y = Petal.Width, col= Species)) + geom_point()
```



When Versicolor and Virginica Species is taken into consideration, noise is observed. It is also observed that the parameters, Petal.length and Petal.width separate out Setosa species with very high confidence, hence making it easy to be clustered.

[illegible]

```
## K-means clustering with 3 clusters of sizes 38, 62, 50
##
## Cluster means:
##      Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1      6.850000     3.073684     5.742105     2.071053
## 2      5.901613     2.748387     4.393548     1.433871
## 3      5.006000     3.428000     1.462000     0.246000
##
## Clustering vector:
##      [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
##      [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##      [75] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1
##     [112] 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1
##     [149] 1 2
```

```
##
## Within cluster sum of squares by cluster:
## [1] 23.87947 39.82097 15.15100
## (between_SS / total_SS = 88.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

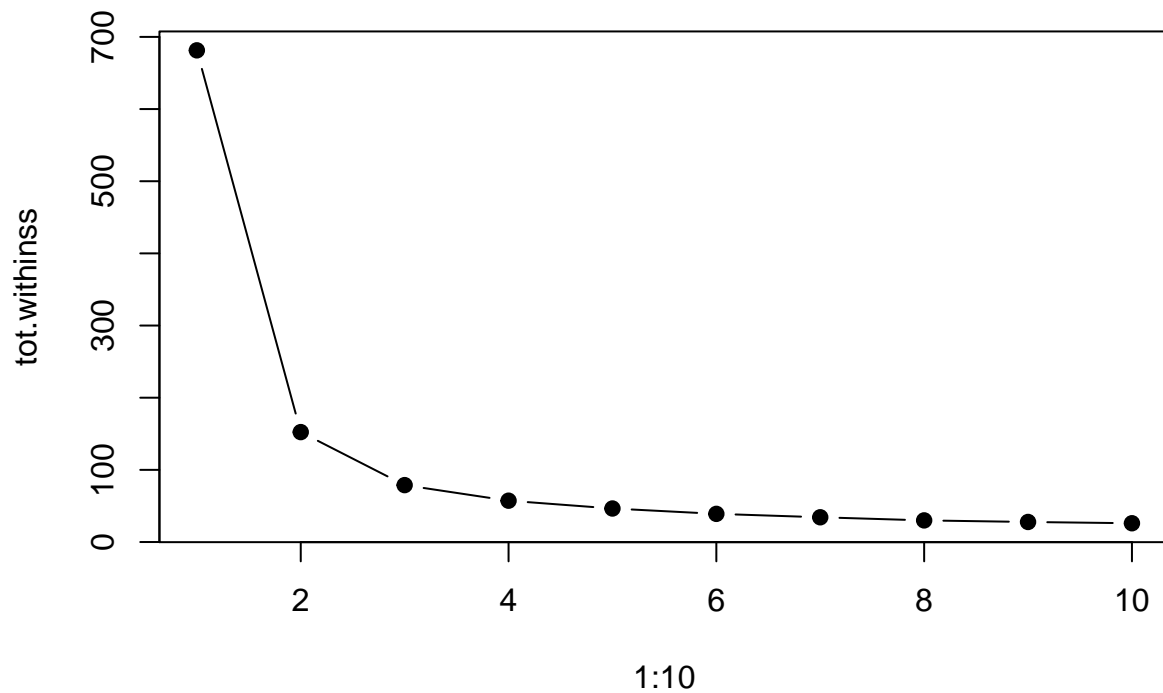
```
table(iriscluster$cluster, df$Species)
```

```
##
##      setosa versicolor virginica
## 1      0          2          36
## 2      0         48          14
## 3     50          0           0
```

```
library(cluster)
```

Optimum number of clusters (Elbow method)

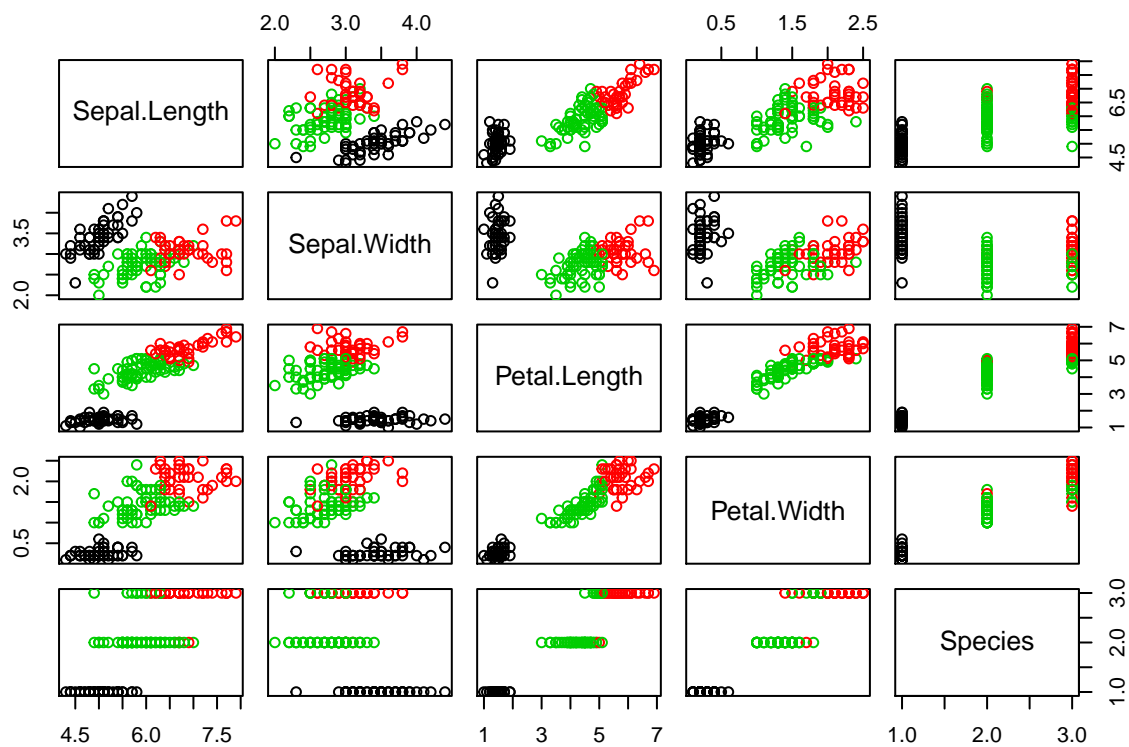
```
tot.withinss <- vector(mode="character", length=10)
for (i in 1:10){
  irisCluster <- kmeans(df[,1:4], center=i, nstart=20)
  tot.withinss[i] <- irisCluster$tot.withinss
}
plot(1:10, tot.withinss, type="b", pch=19)
```



```
fitK = kmeans(df[,1:4], 3)
str(fitK)
```

```
## List of 9
## $ cluster      : int [1:150] 1 1 1 1 1 1 1 1 1 1 ...
## $ centers      : num [1:3, 1:4] 5.01 6.85 5.9 3.43 3.07 ...
## ..- attr(*, "dimnames")=List of 2
## .. ..$ : chr [1:3] "1" "2" "3"
## .. ..$ : chr [1:4] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"
## $ totss       : num 681
## $ withinss    : num [1:3] 15.2 23.9 39.8
## $ tot.withinss: num 78.9
## $ betweenss   : num 603
## $ size        : int [1:3] 50 38 62
## $ iter        : int 3
## $ ifault      : int 0
## - attr(*, "class")= chr "kmeans"
```

```
plot(iris,col = fitK$cluster)
```



```
table(Predicted=fitK$cluster,Actual =iris$Species)
```

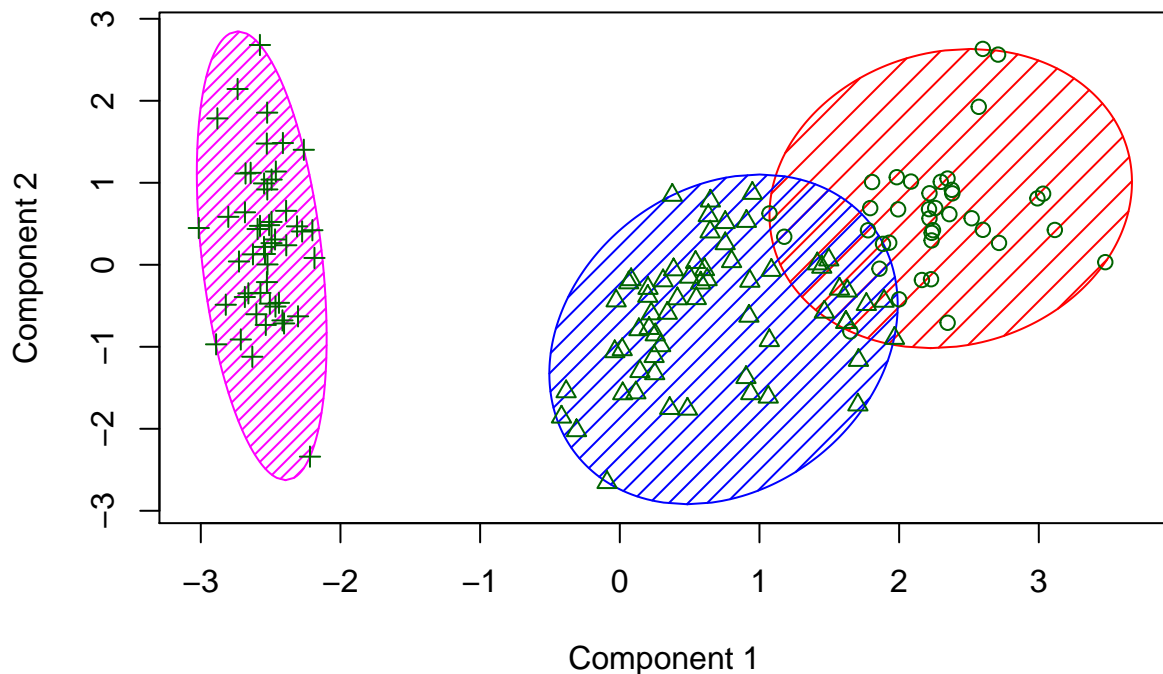
```
##           Actual
## Predicted setosa versicolor virginica
##           1      50           0         0
##           2       0           2        36
##           3       0          48        14
```

Therefore ,the optimal number of clusters is 3.

CLUSTER PLOT

```
clusplot(iris, iriscluster$cluster, color=T, shade=T, labels=0, lines=0)
```

CLUSPLOT(iris)



These two components explain 95.02 % of the point variability.

```
tot.withinss <- vector(mode="character", length=10)
for (i in 1:10){
  iriscluster <- kmeans(df[,1:4], center =i, nstart=20)
  tot.withinss[i] <- iriscluster$tot.withinss
}
```

Therefore, 3 clusters are formed with varying sepal length and sepal width.

The setosa cluster is perfectly explained, however virginica and versicolor have a little noise between their clusters, exact no of centres is obtained using Elbow method above.