AC53013 KMeans Investigation

Robert Meredith

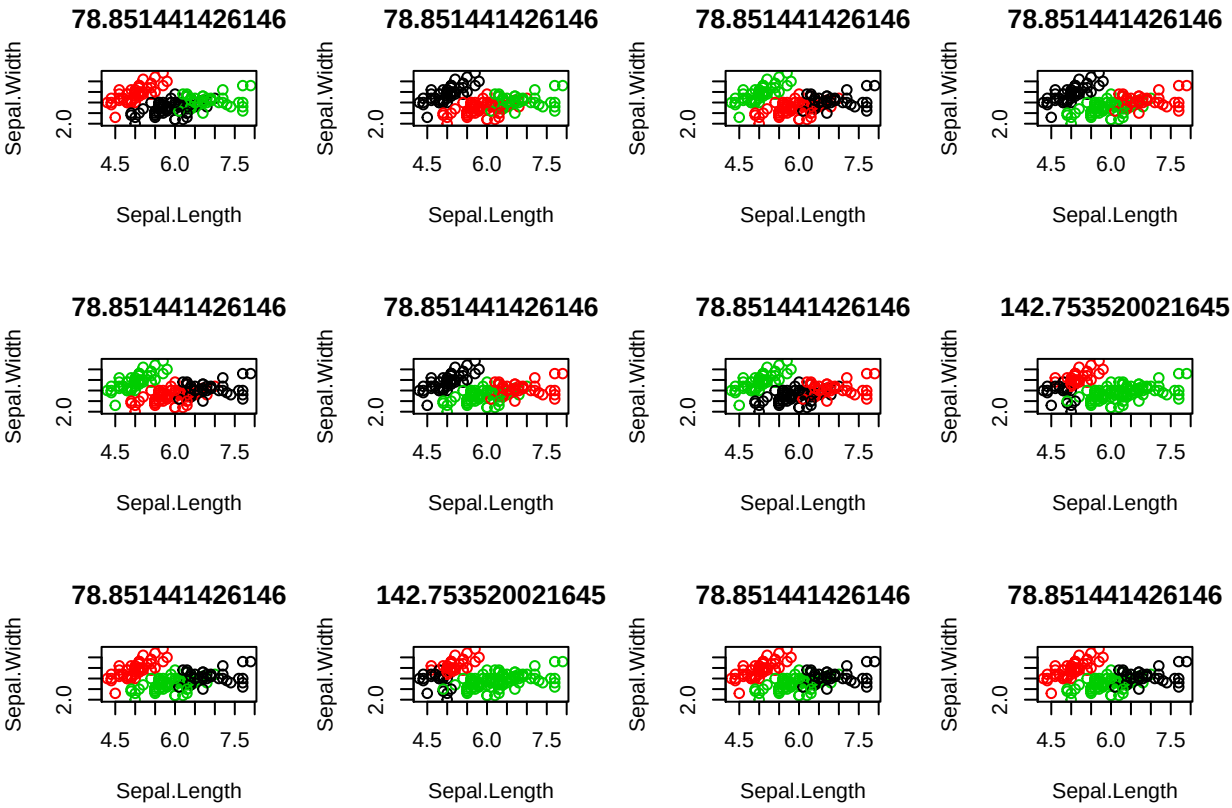
1st March 2020

Introduction

I this assignment I will examine the output of multiple runs of the R K Means algorithm on the Iris dataset and dicuss how Kmeans is working on this data. I will illustrate this through the use of various visualisations.I will assume the reader knows about the KMeans algorithm and not go into choice of number of clusters.

Initial Data Exploration

Using the code provided for this assignment we can see that if we run the loop multiple times that we get different outputs from some of the runs. The clusters are represented by different coloured points on the histogram. Above each histogram is the kmeans "tot.withinss" which is the sum of the "Vector of within-cluster sum of squares, one component per cluster" (Datacamp, 2020). We would like this to be as low as possible as the lower the value the more homogeneity there is in the clusters.



It appears from a visual inspection of this inital exploration that actually there are only two different outputs. The R kmeans algorithm sometimes chooses a different colour for the clusters but there appear to only be two results one of which is more optimum than the other as it has a lower Total Withinss.

To test for this we can compare the cluster output centres and see how many variants there are. However Kmeans does not know which cluster is cluster 1 etc so first we need to compare the cluster centres in any

order using all_equal ignoring row order. Running the code 1000 times shows that we never get a third set of cluster centres as an output.

```
## [1] "First set of centres is"
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     6.850000    3.073684     5.742105    2.071053
## 2     5.006000    3.428000     1.462000    0.246000
## 3     5.901613    2.748387     4.393548    1.433871
```

```
## [1] "Count of first set of centres is:"
```

```
## [1] 804
```

```
## [1] "Second set of centres is"
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     5.175758    3.624242     1.472727   0.2727273
## 2     6.314583    2.895833     4.973958   1.7031250
## 3     4.738095    2.904762     1.790476   0.3523810
```

```
## [1] "Count of second set of centres is:"
```

```
## [1] 196
```

```
## [1] "Count of third set of centres is:"
```

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

A test with each of the algorithms provided with R Kmeans did not show any different results. The optimum result was chosen most of the time with the other result appearing about 20% of the time. The default algorithm in R Kmeans is Hartigan and Wong (1979) however "Lloyd" and "Forgy" are also available. (Datacamp, 2020)

A fairly well known issue with the kmeans algorithm in general is that depending on the starting points chosen the algorithm "is liable to find a local minimum solution instead of a global one, and as such may not find the optimal partition." (**???**). So to investigate this further the starting points of the cluster will be plotted against the outcome. The default in the R Kmeans algorithm is that when the number of cluster are provided, in this case 3, that that number of random rows are taken from the initial dataset as the starting points. (Datacamp, 2020)

Another feature of KMeans is that the initial points might be random by default but given a particular set of starting points the output is not random. We can demonstrate this by running the code say 100,000 time for the same starting points and seeing that the output is the same. The starting points can be provided as a matrix instead of specifying the number of clusters. So we can specify 3 random points in the four dimensional space and see what the results are.

References

Datacamp (2020) *RDocumentation kmeans*. Available at: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/kmeans.