Clustering Wine Dataset

Anushree Shivarudrappa

June 9, 2016

This mini-project is based on the K-Means exercise from ‘R in Action’

Go here for the original blog post and solutions http://www.r-bloggers.com/k-means-clustering-from-r-in-action/

1. Exercise 0: Install these packages if you don’t have them already

install.packages(c(“cluster”, “rattle”,“NbClust”))

```
library(cluster)
library(rattle)
library(NbClust)
```

Now load the data and look at the first few rows

```
data(wine, package="rattle")
head(wine)
```

```
##   Type Alcohol Malic  Ash Alcalinity Magnesium Phenols Flavanoids
## 1    1   14.23  1.71 2.43       15.6       127    2.80       3.06
## 2    1   13.20  1.78 2.14       11.2       100    2.65       2.76
## 3    1   13.16  2.36 2.67       18.6       101    2.80       3.24
## 4    1   14.37  1.95 2.50       16.8       113    3.85       3.49
## 5    1   13.24  2.59 2.87       21.0       118    2.80       2.69
## 6    1   14.20  1.76 2.45       15.2       112    3.27       3.39
##   Nonflavanoids Proanthocyanins Color  Hue Dilution Proline
## 1          0.28            2.29  5.64 1.04     3.92    1065
## 2          0.26            1.28  4.38 1.05     3.40    1050
## 3          0.30            2.81  5.68 1.03     3.17    1185
## 4          0.24            2.18  7.80 0.86     3.45    1480
## 5          0.39            1.82  4.32 1.04     2.93     735
## 6          0.34            1.97  6.75 1.05     2.85    1450
```

2. Exercise 1: Remove the first column from the data and scale it using the scale() function

```
df_wine <- scale(wine[-1])
summary(df_wine)
```

```
##     Alcohol             Malic              Ash
##  Min.   :-2.42739   Min.   :-1.4290   Min.   :-3.66881
##  1st Qu.:-0.78603   1st Qu.:-0.6569   1st Qu.:-0.57051
##  Median : 0.06083   Median :-0.4219   Median :-0.02375
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000  
##  3rd Qu.: 0.83378   3rd Qu.: 0.6679   3rd Qu.: 0.69615  
##  Max.   : 2.25341   Max.   : 3.1004   Max.   : 3.14745  
##    Alcalinity          Magnesium          Phenols        
##  Min.   :-2.663505   Min.   :-2.0824   Min.   :-2.10132  
##  1st Qu.:-0.687199   1st Qu.:-0.8221   1st Qu.:-0.88298  
##  Median : 0.001514   Median :-0.1219   Median : 0.09569  
##  Mean   : 0.000000   Mean   : 0.0000   Mean   : 0.00000  
##  3rd Qu.: 0.600395   3rd Qu.: 0.5082   3rd Qu.: 0.80672  
##  Max.   : 3.145637   Max.   : 4.3591   Max.   : 2.53237  
##    Flavanoids       Nonflavanoids     Proanthocyanins        Color        
##  Min.   :-1.6912   Min.   :-1.8630   Min.   :-2.06321   Min.   :-1.6297  
##  1st Qu.:-0.8252   1st Qu.:-0.7381   1st Qu.:-0.59560   1st Qu.:-0.7929  
##  Median : 0.1059   Median :-0.1756   Median :-0.06272   Median :-0.1588  
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000  
##  3rd Qu.: 0.8467   3rd Qu.: 0.6078   3rd Qu.: 0.62741   3rd Qu.: 0.4926  
##  Max.   : 3.0542   Max.   : 2.3956   Max.   : 3.47527   Max.   : 3.4258  
##       Hue               Dilution          Proline       
##  Min.   :-2.08884   Min.   :-1.8897   Min.   :-1.4890  
##  1st Qu.:-0.76540   1st Qu.:-0.9496   1st Qu.:-0.7824  
##  Median : 0.03303   Median : 0.2371   Median :-0.2331  
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000  
##  3rd Qu.: 0.71116   3rd Qu.: 0.7864   3rd Qu.: 0.7561  
##  Max.   : 3.29241   Max.   : 1.9554   Max.   : 2.9631  
```

Now we'd like to cluster the data using K-Means.
How do we decide how many clusters to use if you don't know that already?
We'll try two methods.

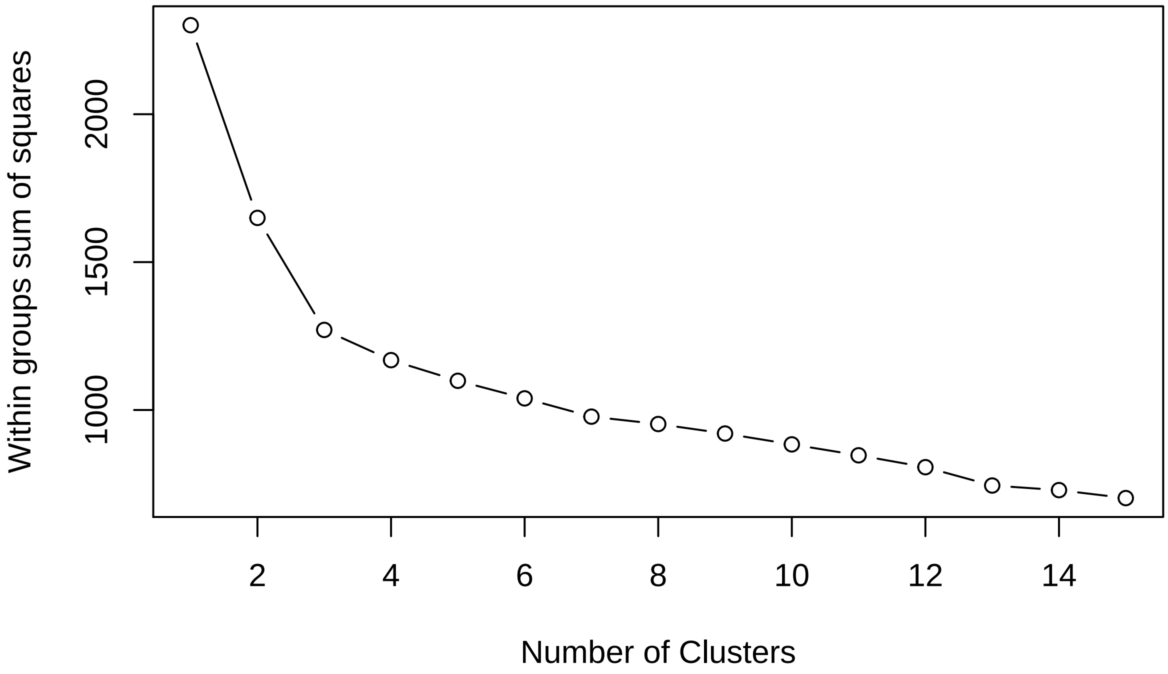
Method 1:

A plot of the total within-groups sums of squares against the number of clusters in a K-means solution can be helpful. A bend in the graph can suggest the appropriate number of clusters.

```
wssplot <- function(data, nc=15, seed=1234){
                  wss <- (nrow(data)-1)*sum(apply(data,2,var))
                      for (i in 2:nc){
                set.seed(seed)
                    wss[i] <- sum(kmeans(data, centers=i)$withinss)}

              plot(1:nc, wss, type="b", xlab="Number of Clusters",
                            ylab="Within groups sum of squares")
       }

wssplot(df_wine)
```



3. Exercise 2:

How many clusters does this method suggest?
Method suggest cluster count as 3 i.e k=3.

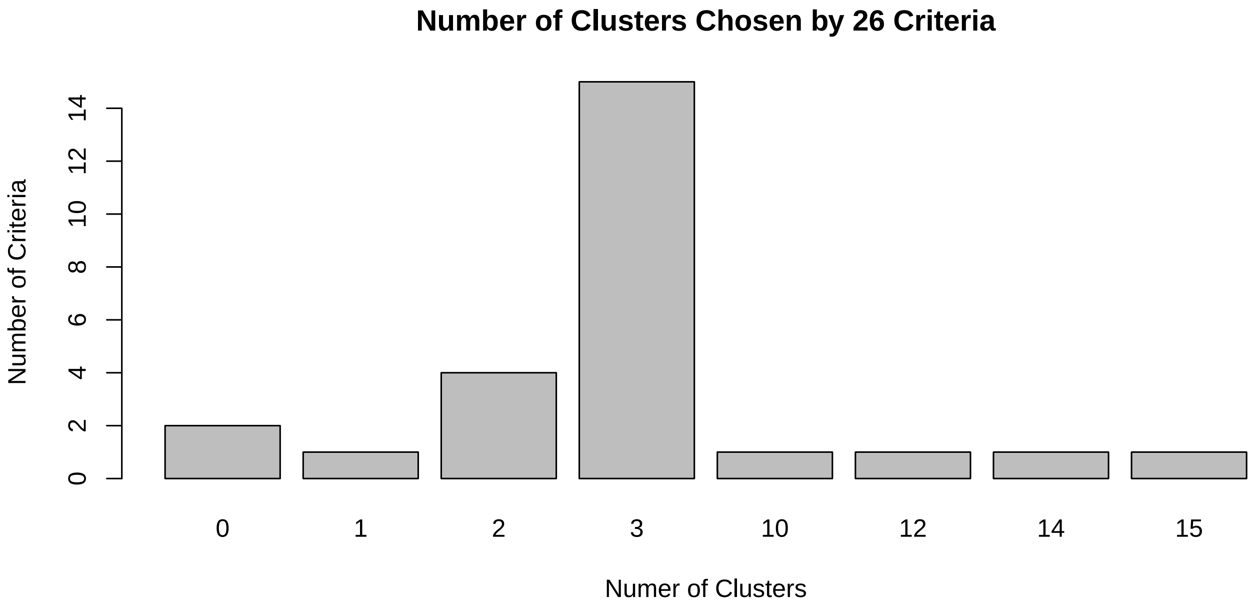
Why does this method work? What's the intuition behind it?
Sum of square error (SSE) plot is used to determine appropriate k value. Plot indicates that there is distinct drop when moving from 1 to 3 clusters. After 3 we can observe decrease in drop off, this suggest a 3-cluster as solution. If we look at the original data it also contains 3 classes.

Method 2:

Use the NbClust library, which runs many experiments and gives a distribution of potential number of clusters.

```
library(NbClust)
set.seed(1234)
nc <- NbClust(df_wine, min.nc=2, max.nc=15, method="kmeans")
```

```
barplot(table(nc$Best.n[1,]),
              xlab="Numer of Clusters", ylab="Number of Criteria",
                    main="Number of Clusters Chosen by 26 Criteria")
```



4. Exercise 3: How many clusters does this method suggest?

By Looking into graph we can say best number of cluster is 3 i.e k=3

5. Exercise 4: Once you've picked the number of clusters, run k-means using this number of clusters. Output the result of calling kmeans() into a variable fit.km

```
fit.km <- kmeans( df_wine, 3 )
str(fit.km)
```

```
## List of 9
##  $ cluster     : int [1:178] 3 3 3 3 3 3 3 3 3 3 ...
##  $ centers     : num [1:3, 1:13] 0.164 -0.923 0.833 0.869 -0.393 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "1" "2" "3"
##   .. ..$ : chr [1:13] "Alcohol" "Malic" "Ash" "Alcalinity" ...
##  $ totss       : num 2301
##  $ withinss    : num [1:3] 326 559 386
##  $ tot.withinss: num 1271
##  $ betweenss   : num 1030
##  $ size        : int [1:3] 51 65 62
##  $ iter        : int 3
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

Now we want to evaluate how well this clustering does.

6. Exercise 5:

Using the table() function, show how the clusters in fit.km$clusterscomparestotheactualwinetypesinwine$Type. Would you consider this a good clustering?

```
table(fit.km$cluster)
```

```
##
##  1  2  3
## 51 65 62
```

```
table(wine$Type)
```

```
##
##  1  2  3
## 59 71 48
```

```
table_clust_wine <- table(wine$Type, fit.km$cluster)
table_clust_wine
```

```
##
##      1  2  3
##   1  0  0 59
##   2  3 65  3
##   3 48  0  0
```

Its a confusion matrix. To find "fit.km" is a good cluster or not, using an adjusted Rank index provided by the flexclust package.

```
library(flexclust)
randIndex(table_clust_wine)
```

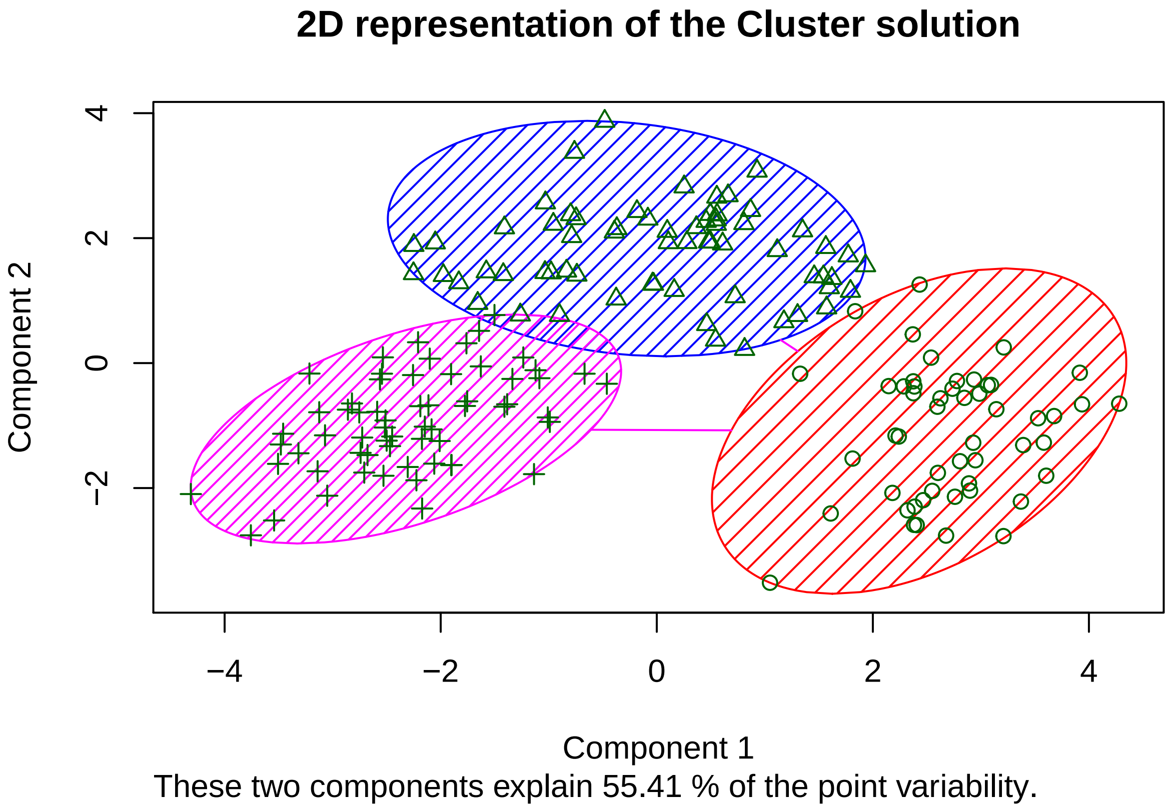
```
##      ARI
## 0.897495
```

The adjusted Rand index provides a measure of the agreement between two partitions, adjusted for chance. It ranges from -1 (no agreement) to 1 (perfect agreement). Agreement between the wine varietal type and the cluster solution is 0.9. So its a good clustering

7. Exercise 6:

Visualize these clusters using function clusplot() from the cluster library Would you consider this a good clustering?

```
clusplot( df_wine, fit.km$cluster, color = T, shade = T,
          main='2D representation of the Cluster solution')
```


2D representation of the Cluster solution
Component 2
4
2
0
−2
−4
−2
0
2
4
Component 1
These two components explain 55.41 % of the point variability.