Wine Clustering

Amine Agrane & Lydia Khelfane

10/02/2021

K-means clustering on the wine dataset

Objective

The objective of this report is to describe the implementation of a k-Means algorithm and discuss the design decisions. We test our k-Means algorithm on the Wine dataset and discuss the performance of it. We are using clustering for classifying the colour.

Reading the data #Reading data

```
data_wine <- as.data.frame( read.csv(file = './data/winequality.csv', sep=',', stringsAsFactors=F))
```

```
# It show the first few rows
head(data_wine)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.3             0.65        0.00            1.2     0.065
## 2           7.8             0.88        0.00            2.6     0.098
## 3           7.8             0.76        0.04            2.3     0.092
## 4          11.2             0.28        0.56            1.9     0.075
## 5           7.4             0.70        0.00            1.9     0.076
## 6           7.4             0.66        0.00            1.8     0.075
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  15                   21  0.9946 3.39      0.47    10.0
## 2                  25                   67  0.9968 3.20      0.68     9.8
## 3                  15                   54  0.9970 3.26      0.65     9.8
## 4                  17                   60  0.9980 3.16      0.58     9.8
## 5                  11                   34  0.9978 3.51      0.56     9.4
## 6                  13                   40  0.9978 3.51      0.56     9.4
##   quality color
## 1       7   red
## 2       5   red
## 3       5   red
## 4       6   red
## 5       5   red
## 6       5   red
```

Prepare the data:

Transform the color into numeric values After we uploaded the data wine, we drop the quality label and we digitize the color data, The new values of the color feature will bz :

- 1 for red wine.

- 0 for white wine.

```
# %% [code]
colSums(is.na(data_wine))
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##                    0                    0                    0
##       residual.sugar            chlorides  free.sulfur.dioxide
##                    0                    0                    0
## total.sulfur.dioxide              density                   pH
##                    0                    0                    0
##            sulphates              alcohol              quality
##                    0                    0                    0
##                color
##                    0
```

```
# %% [code]
Quality<-data_wine$quality
data_wine$quality<-NULL

# %% [code]
data_wine$color<- ifelse(data_wine$color == "red",1,0)
```

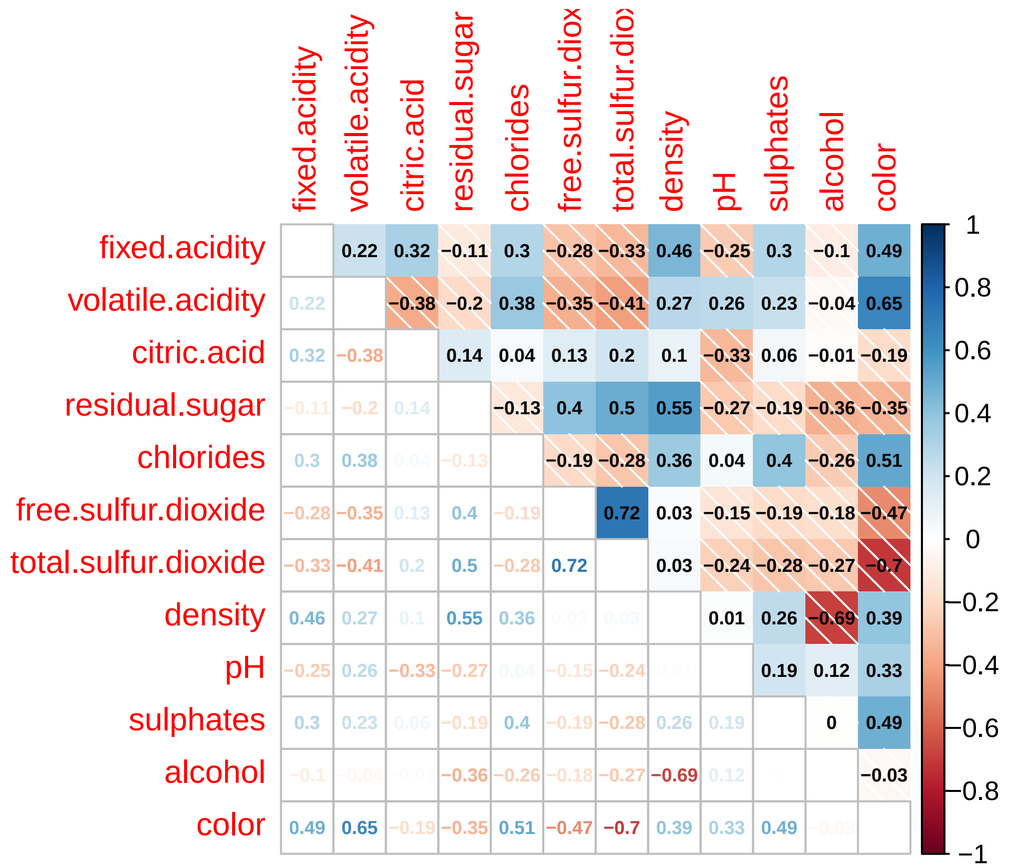
Scale the dataset We also use the `scale()` function to scanle our dataset, `scale`, will calculate the mean and standard deviation of the entire vector/column, then recompute each elements of a vector by subtracting the mean and dividing by the sd. (If you use scale(x, scale=FALSE), it will only subtract the mean but not divide by the std deviation.)

```
# scale the dataset
scaled_dataset_wine <- scale(data_wine)
scaled_dataset_wine <- as.data.frame(scaled_dataset_wine)

# merge the data set with the new numeric value for color
scaled_dataset_wine_for_predict <- cbind(scaled_dataset_wine, color = data_wine$color)
scaled_dataset_wine_for_predict <- as.data.frame(scaled_dataset_wine_for_predict)



head(data_wine)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.3             0.65        0.00            1.2     0.065
## 2           7.8             0.88        0.00            2.6     0.098
## 3           7.8             0.76        0.04            2.3     0.092
## 4          11.2             0.28        0.56            1.9     0.075
## 5           7.4             0.70        0.00            1.9     0.076
## 6           7.4             0.66        0.00            1.8     0.075
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol color
## 1                  15                   21  0.9946 3.39      0.47    10.0     1
## 2                  25                   67  0.9968 3.20      0.68     9.8     1
## 3                  15                   54  0.9970 3.26      0.65     9.8     1
## 4                  17                   60  0.9980 3.16      0.58     9.8     1
## 5                  11                   34  0.9978 3.51      0.56     9.4     1
## 6                  13                   40  0.9978 3.51      0.56     9.4     1
```

Lets check correlation between various columns of Wine Quality data

```r
dataset_wine_numcols <- scaled_dataset_wine[, sapply(scaled_dataset_wine, is.numeric)]

corrplot.mixed(
  cor(dataset_wine_numcols),
  upper = "shade",
  lower = "number",
  tl.pos = "lt",
  addCoef.col = "black",
  number.cex = .6
)
```



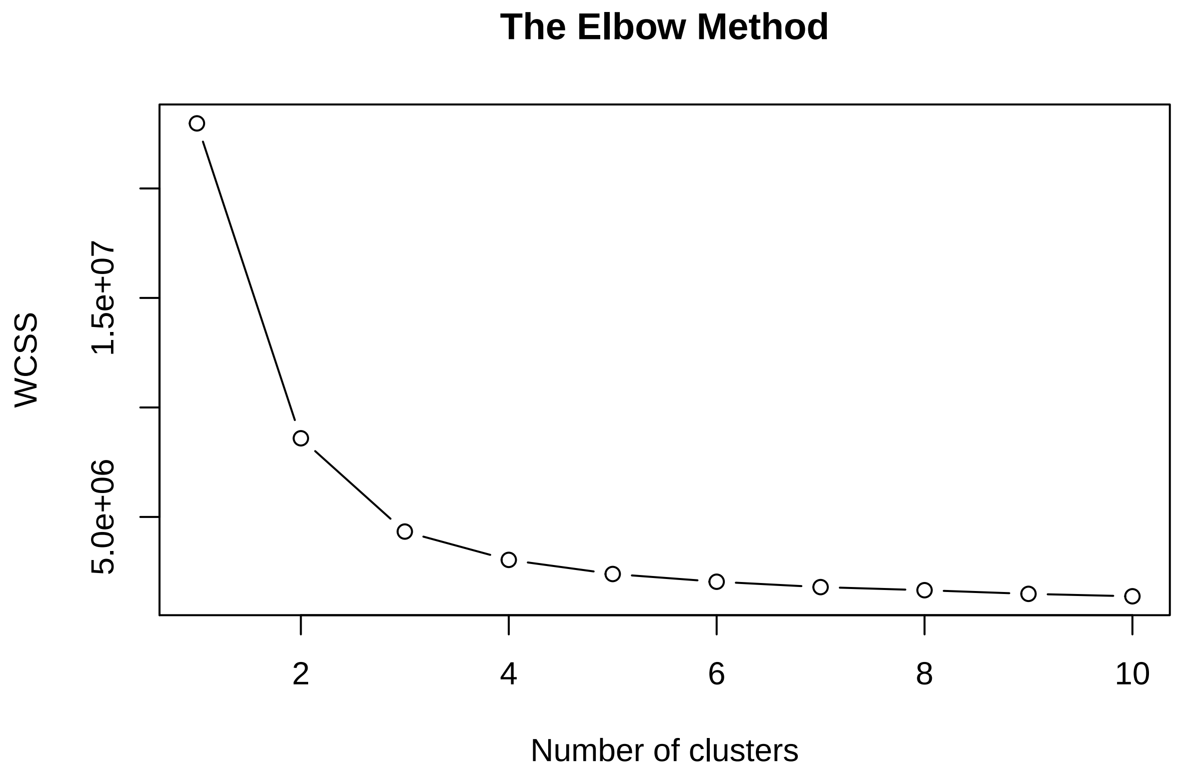
Clustering

Determining optimal number of clusters: We need to determine the number of clusters for which the model is not overfitting but clusters the data as per the actual distribution using the Elbow Method. In elbow method, percentage of variance is explained as a function of the number of clusters plotted.

Elbow Method In cluster analysis, the elbow method is a heuristic used in determining the number of clusters in a data set. The method consists of plotting the explained variation as a function of the number of clusters, and picking the elbow of the curve as the number of clusters to use. This method allows us to identify the number of clusters to use to segment the wine data.

From the above figure, we plot for each different values of `number_of_cluster` the sum of squares for number of clusters have been plotted from 1 to 10, we have chosen 2 as the number of clusters as the value of within groups sum of squares does not change significantly after 2.

```r
wcss <- vector()

for (i in 1:10) {

  wcss[i] = sum(kmeans(data_wine, i)$withinss)

}

plot(
  1:10,
  wcss,
  type = 'b',
  main = paste('The Elbow Method'),
  xlab = 'Number of clusters',
  ylab = 'WCSS'
)
```



Application of the K-Means Clustering Now that we apply the Elbow method, we apply a K-Means Clustering algorithm with 2 centroids. K-Means Clustering is an unsupervised learning algorithm that tries to cluster data based on their similarity. Unsupervised learning means that there is no outcome to be predicted, and the algorithm just tries to find patterns in the data. We are using clustering for classifying the colour. The general scenario where you would use clustering is when you want to learn more about your dataset. We want to see if we can cluster the wine element into two distinct cluster named 'white' and 'red' and see if thouse two types of wine have distinct features values that distinguish them. The K-Means Clustering algorithm iterates through two steps:

- Reassign data points to the cluster whose centroid is closest.
- Calculate new centroid of each cluster.

```r
Clusters <- kmeans(data_wine, centers =2 , nstart = 25)
Clusters
```

```
## K-means clustering with 2 clusters of sizes 2808, 3689
##
## Cluster means:
##   fixed.acidity volatile.acidity citric.acid residual.sugar  chlorides
## 1      7.623184        0.4086200   0.2908725       3.076175 0.06580591
## 2      6.904812        0.2871659   0.3397642       7.244809 0.04859257
##   free.sulfur.dioxide total.sulfur.dioxide   density       pH sulphates
## 1            18.40011             63.25855 0.9945725 3.254840 0.5723825
## 2            39.75590            155.69246 0.9947903 3.190808 0.4999485
##    alcohol      color
## 1 10.79743 0.53917379
## 2 10.25932 0.02304147
##
## Clustering vector:
##    [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1
##   [38] 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
##   [75] 1 1 1 1 1 2 1 1 1 1 1 1 2 1 2 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1
##  [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1
##  [149] 1 1 1 1 1 1 2 2 2 2 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [186] 1 1 1 2 2 2 1 2 1 1 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 2 1 1 1 2 1 1
##  [223] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1
##  [260] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [297] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
##  [334] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [371] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1
##  [408] 1 1 1 1 1 1 1 2 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [445] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [482] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1
##  [519] 1 1 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [556] 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2
##  [593] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [630] 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [667] 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1
##  [704] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [741] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1
##  [778] 1 1 1 1 1 1 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [815] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [852] 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [889] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [926] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [963] 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1000] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1037] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1074] 1 1 1 1 1 1 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1111] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1148] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1185] 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1222] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1259] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1296] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1333] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1370] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1
## [1407] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1444] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1481] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1518] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1555] 1 1 1 1 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1592] 1 1 1 1 1 1 1 1 2 2 1 2 2 1 2 2 2 2 1 1 1 2 2 2 1 1 2 2 1 1 2 2 2 2 2 2 2
## [1629] 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 1 2 2
## [1666] 2 2 1 1 2 2 2 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2
## [1703] 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 1
## [1740] 1 2 1 1 1 2 2 1 1 2 2 2 2 1 1 2 2 2 2 2 2 2 1 2 2 2 2 1 2 1 2 1 1 2 2 2 1
## [1777] 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2
## [1814] 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 1 1 2 2
## [1851] 2 2 1 2 2 1 1 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2
## [1888] 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 1 2 2 2 2
## [1925] 2 2 2 2 1 1 2 1 2 1 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
## [1962] 2 2 1 2 2 2 2 1 2 2 2 2 1 1 2 1 2 2 1 2 2 2 2 1 2 2 2 2 2 1 2 1 2 2 1 1 2
## [1999] 1 1 2 1 2 2 2 1 2 2 1 2 2 1 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 1 1
## [2036] 2 2 2 1 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2 1
## [2073] 2 1 2 1 2 2 2 2 1 2 2 2 2 2 1 2 2 1 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2110] 2 2 2 1 2 2 2 2 1 1 2 2 1 1 1 2 1 2 1 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2
## [2147] 1 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 1
## [2184] 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 1 2 1 2 2 2 2 1 2 2 2 2
## [2221] 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2
## [2258] 2 2 1 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2295] 2 1 2 2 2 2 2 1 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 2
## [2332] 1 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2
## [2369] 2 2 2 2 2 2 1 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 1 2 2 2 2 2
## [2406] 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 1 2 1 2 2 2 1 1 2 2 1 1 2 1 2 2 2 2 2 2 2
## [2443] 1 2 2 2 1 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 1 1 2 2 1 2 1
## [2480] 2 2 2 2 2 1 2 1 2 1 2 2 2 2 2 1 2 1 1 2 2 2 2 2 2 1 1 2 2 2 2 2 2 1 1 1 2
## [2517] 2 1 1 2 2 2 2 2 1 1 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 1 2 2 1 2 2 1 1 2 2 1 2
## [2554] 2 2 2 1 1 2 2 1 2 2 2 1 2 2 1 1 1 1 2 1 2 1 2 2 2 1 1 2 1 1 2 2 2 2 2 2 2
## [2591] 1 2 1 2 2 1 2 2 2 1 2 2 2 2 1 2 1 2 2 1 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1
## [2628] 2 2 2 2 2 2 2 2 1 1 1 1 2 1 1 2 1 1 1 1 1 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2
## [2665] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2
## [2702] 2 2 2 2 2 1 1 2 1 1 2 1 1 1 1 2 1 2 1 2 1 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2
## [2739] 1 1 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 1 2 1 2 1 2 2 2 2 1 2 2
## [2776] 2 2 2 1 2 2 1 2 2 2 2 1 2 1 2 2 1 2 2 2 2 2 1 1 1 1 2 1 1 2 2 2 2 2 1 2 2
## [2813] 1 1 2 1 2 2 1 2 2 2 2 1 1 1 1 1 1 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2
## [2850] 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [2887] 2 1 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 1 2 1 2 2 2
## [2924] 2 2 2 2 1 1 1 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2
## [2961] 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2 1 2 2 2 1 1 1 2 1 2 2
## [2998] 2 1 2 2 2 2 1 2 1 1 2 2 1 1 2 2 2 1 2 2 2 1 2 1 1 2 2 2 2 2 1 1 2 1 1 1 2
## [3035] 1 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2
## [3072] 1 2 1 2 2 1 2 2 2 2 2 1 2 2 1 2 2 2 2 1 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2
## [3109] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 1 1 2 1 1 1 1 1 2
## [3146] 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 1 2 2 1 2 2 2 2 2 2 1 2 1 2 2 2 2 1 2
## [3183] 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 1 2 2 1 1 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2
## [3220] 1 2 2 2 2 2 2 2 1 1 2 1 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 1
## [3257] 2 2 2 2 2 2 2 2 2 1 2 1 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3294] 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 1 2 2 2 1
## [3331] 2 2 2 2 2 1 2 1 1 2 2 2 2 2 2 1 2 1 2 2 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 2 2
## [3368] 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2
## [3405] 2 2 2 2 2 2 2 1 1 1 2 2 2 1 2 2 1 1 2 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 1 2 1
## [3442] 2 2 1 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2
## [3479] 1 2 1 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 1 2 2 2 2 1 2 2 1 2 2 2 1
## [3516] 2 1 2 2 2 2 2 1 1 1 1 2 2 2 2 2 1 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3553] 2 2 2 1 1 2 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1
## [3590] 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 2 2 2 1 1 2 1 2 2 2 2 2 2 2 2 1
## [3627] 2 2 2 2 2 1 2 2 1 2 2 1 1 2 1 1 1 2 1 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 1 2 2
## [3664] 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2
## [3701] 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3738] 1 1 2 2 2 1 2 2 1 2 1 1 1 2 1 1 2 2 1 1 1 1 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2
## [3775] 2 2 2 2 1 2 2 2 2 1 1 1 2 1 2 2 2 2 2 1 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [3812] 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1
## [3849] 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 1 2 2 2 1 2 2 1 1 2 2 2 2 2
## [3886] 2 2 2 1 1 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2
## [3923] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 1 2 2 1 2 2 2 2 2 1 1 2 2 1 2 2
## [3960] 2 1 2 2 2 2 2 2 2 2 1 2 1 1 2 2 2 1 2 2 2 2 1 1 1 2 2 2 1 1 2 2 2 2 2 2 2
## [3997] 1 1 1 1 1 2 2 2 2 1 1 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [4034] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 1 2 2
## [4071] 2 1 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2
## [4108] 2 2 2 2 1 2 2 1 2 2 1 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 1 2 1 2 2 1 2 2 2 1 2
## [4145] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 1 2 2 2 1 2 2 2 2 1 1 2 2 2 2 2 1 2 2 2
## [4182] 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 1 1 2 2 2 1 2 2 1 2 1 2 2 2 2
## [4219] 2 2 2 2 2 1 2 2 2 2 2 1 1 2 2 2 1 2 2 1 1 1 2 1 2 2 2 1 2 2 2 2 2 1 2 2 2
## [4256] 2 2 2 1 2 2 2 1 1 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2
## [4293] 2 1 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 2 1 2 2 2 1 2 2 1
## [4330] 2 2 2 1 2 2 2 1 2 1 2 2 2 1 1 1 2 2 2 2 2 2 2 1 1 2 2 1 1 2 2 2 2 2 2 2 1
## [4367] 2 2 2 2 2 2 1 2 2 2 2 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 1 1 2 2 2 2 2 1 1
## [4404] 1 2 2 2 2 2 1 2 1 2 1 1 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 1 1
## [4441] 1 1 1 1 1 1 1 2 2 2 1 2 1 1 2 2 1 2 2 2 1 1 1 1 2 2 2 2 1 2 1 2 1 2 1 2 2
## [4478] 2 1 1 1 2 1 2 1 1 1 1 2 2 2 2 2 1 2 2 2 1 2 1 1 2 1 2 2 2 1 1 1 2 2 1 2 1
## [4515] 1 2 2 1 2 1 2 2 2 2 2 1 2 2 2 2 1 2 2 1 1 1 2 1 1 2 1 2 1 2 2 1 1 2 2 1 1
## [4552] 1 1 1 2 1 1 1 1 2 2 1 2 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 1
## [4589] 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 1 1 1 1 1 1 2 2 1 1 1 2 1 1 2
## [4626] 2 2 2 2 2 2 1 2 2 1 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 1 2 1 2 2 1 2 2 2
## [4663] 2 2 2 2 2 2 1 2 1 2 2 2 1 2 2 1 2 1 2 1 1 1 1 1 2 1 1 1 2 2 2 1 1 1 2 2 2
## [4700] 2 1 2 2 2 2 2 2 2 2 1 1 2 2 2 2 1 1 2 1 2 2 1 1 2 2 2 1 1 2 2 2 2 2 2 2 1
## [4737] 2 2 2 2 1 2 1 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 1 2 2
## [4774] 2 2 1 2 1 1 1 1 2 1 1 2 1 2 2 2 2 2 1 2 1 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 1
## [4811] 2 2 2 1 1 2 1 1 1 1 1 2 2 2 1 2 2 2 2 2 1 1 1 2 2 2 2 2 1 2 2 2 1 1 2 2 2
## [4848] 2 2 2 2 1 1 2 2 2 2 2 2 2 1 2 1 2 2 1 2 2 2 2 1 1 2 2 1 2 2 2 2 2 2 2 2 1
## [4885] 2 2 2 2 2 1 1 2 1 2 2 2 2 2 1 1 1 1 1 2 1 2 2 2 1 2 2 1 1 2 1 1 1 1 2 2 1
## [4922] 1 1 2 2 2 1 2 2 2 2 2 2 1 2 2 2 1 1 2 1 1 2 2 2 2 2 1 1 2 1 1 1 2 2 2 1 1
## [4959] 1 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 1 2 2 2 2
## [4996] 2 2 2 1 2 2 2 1 1 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2
## [5033] 1 1 1 1 2 2 2 1 2 1 1 2 2 2 1 2 2 1 1 2 1 1 1 2 2 2 2 1 2 2 1 2 2 2 2 1 2
## [5070] 2 1 2 1 2 2 1 2 2 1 1 2 1 1 1 2 1 2 1 1 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2
## [5107] 2 2 2 2 1 2 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 1 1 2 1 1 2 2 2
## [5144] 2 2 2 2 2 2 2 2 2 2 1 2 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 1 2
## [5181] 2 1 1 1 1 2 2 2 2 1 2 2 2 1 2 2 2 2 2 1 2 2 2 1 2 2 1 2 2 2 1 2 2 2 1 2 2
## [5218] 2 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 1 1 2 2 2 2 2 2 1 1 2 2 2 2 1 2 1
## [5255] 2 2 2 2 2 2 2 1 2 2 2 2 2 1 1 2 1 1 1 2 2 1 2 1 1 2 2 2 2 2 2 2 1 2 2 2 2
## [5292] 1 2 1 2 2 2 1 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 1
## [5329] 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 1 1 1 1 2 2
## [5366] 2 2 2 2 2 2 2 1 2 1 2 1 1 1 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 1 1 2 2 1 1 2
## [5403] 1 1 1 1 1 1 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 1 2 1 2 1 2 2 2 2 2 2 1 1 2 2
## [5440] 1 2 2 1 1 2 1 2 1 1 1 2 1 2 2 2 1 1 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2
## [5477] 1 2 1 2 2 2 2 2 1 2 1 2 2 1 2 2 1 2 1 2 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 1 1
## [5514] 2 1 2 2 2 1 2 2 1 1 1 1 1 1 2 2 1 1 2 1 1 2 1 2 1 2 2 2 1 2 2 2 2 2 1 2 2
## [5551] 2 1 2 2 1 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 1 1 2
## [5588] 2 1 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 1 2 2 2 2 2 2 1 2 2 2 1 1 1 2 2 2
## [5625] 1 2 2 1 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 1 2 2 2 1 1 1
## [5662] 2 2 1 2 2 2 2 2 2 1 2 1 1 2 2 1 1 1 2 2 2 2 1 1 1 1 1 2 2 1 2 1 1 1 2 1 2
## [5699] 2 1 2 2 1 1 2 2 2 1 2 2 1 1 1 1 1 2 2 2 2 2 2 2 1 1 2 2 2 2 1 2 2 2 2 1 2
## [5736] 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 2 1 2
## [5773] 1 2 2 2 2 2 2 2 1 2 1 1 2 1 2 2 1 1 2 1 1 1 1 1 2 2 1 1 1 2 2 2 1 2 2 1 1
## [5810] 2 2 2 1 2 2 2 2 2 2 1 2 2 1 1 2 2 1 2 2 2 1 1 2 2 2 2 1 2 1 2 2 2 1 2 2 1
## [5847] 2 2 2 1 1 1 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 1
## [5884] 1 1 1 2 1 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 1 2 2 1 2 1 2 2 1 2 2 2 2
## [5921] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 1 1 1 2 2 2 1 2 2 2 2 2 2 2
## [5958] 2 2 2 2 2 2 2 2 2 2 1 2 2 1 2 1 2 2 2 1 2 2 2 2 2 1 2 2 2 1 1 2 2 2 2 2 2
## [5995] 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
## [6032] 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1
## [6069] 1 1 1 1 2 2 2 1 2 2 2 2 2 2 1 2 1 1 1 2 2 2 1 1 2 1 2 1 2 1 1 2 2 2 1 2 2
## [6106] 1 1 2 1 2 1 1 1 2 2 1 1 1 2 2 2 1 2 2 2 2 1 1 2 2 2 2 2 1 2 2 1 2 1 1 1 1
## [6143] 1 1 1 1 1 2 1 1 2 1 1 1 1 2 2 2 1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 1 2 1 1 1 2
## [6180] 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 1 1 2 1 1 1 1 2 2 2 2 1
## [6217] 2 2 2 2 1 2 2 1 2 2 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 1 2 1 1 2 2 1
## [6254] 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 2 2 2 1 1 2 2 2 2 2 2 2 2
## [6291] 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 1 2 1 2 2 1 2 1 1 2 2 1 1 1 2 2 1 2 1 2
## [6328] 2 1 1 2 2 2 1 1 1 2 2 1 2 2 2 2 1 2 2 1 2 2 2 2 1 1 1 1 2 2 2 2 2 2 1 2 2
## [6365] 1 2 2 2 2 2 2 2 1 2 2 2 1 1 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2
## [6402] 2 2 1 1 2 1 1 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 1 2 2 2 1 1 1 1 2 2
## [6439] 1 2 2 1 2 2 1 2 1 2 2 2 2 1 1 1 2 2 2 2 2 2 1 1 1 2 1 2 1 2 1 2 1 2 2 2 1
## [6476] 2 1 1 2 2 2 2 2 2 2 2 1 2 2 1 2 2 1 2 2 1 1
##
## Within cluster sum of squares by cluster:
## [1] 3256316 5335227
##  (between_SS / total_SS =  62.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

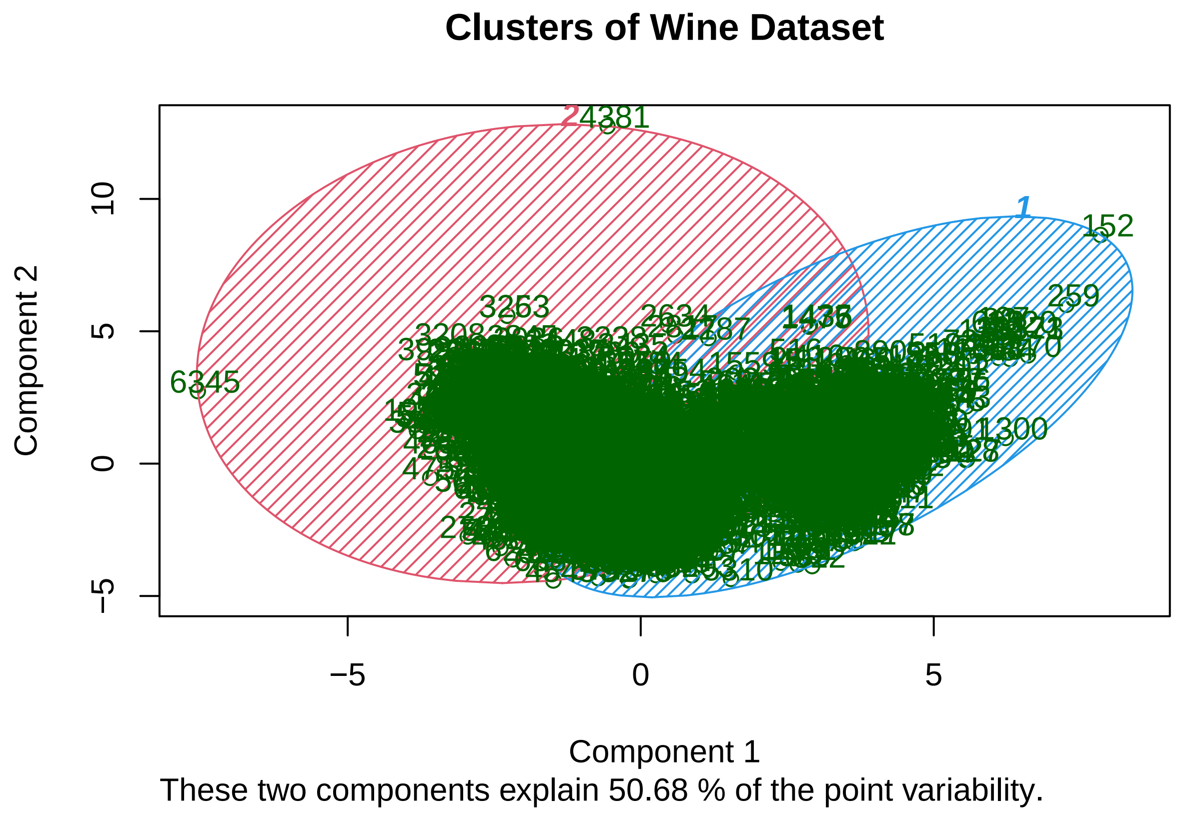
```
data_wine$quality<-Quality
data_wine$cluster<-Clusters$cluster
```

Visualisation of the clusters of Wine quality data Our data set is multidimensional with N=14 (14 different features). We have to find a way to plot a 13 dimensional dataset (The 14ith columns is the cluster class of that specific row and is mapped to color in the figure) into a 2 dimensional figure. To acheive this we use `clusplot`function. The clusplot uses '*PCA* to draw the data. It uses the first two principal components to explain the data on the figure.

Principal components are the (orthogonal) axes that along them the data has the most variability, if your data is 2d then using two principal components can explain the whole variability of the data, thus the reason you see 100% explained. If your data is from a higher dimension but has a lot of correlations you can use a lower dimensional space to explain it.

We can see above the figure thats describes the clustering of our 13 dimensional dataset into two distincts clusters :

```
clusplot(
  data_wine,
 Clusters$cluster,
  lines = 0,
  shade = TRUE,
  color = TRUE,
  labels = 2,
  plotchar = FALSE,
  span = TRUE,
  main = paste('Clusters of Wine Dataset')
)
```



Model Evaluation Now we calculate Hopkin's statistic for given dataset and random dataset. The Hopkins statistic is a way of measuring the cluster tendency of a data set. It belongs to the family of sparse sampling tests. It acts as a statistical hypothesis test where the null hypothesis is that the data is generated by a Poisson point process and are thus uniformly randomly distributed

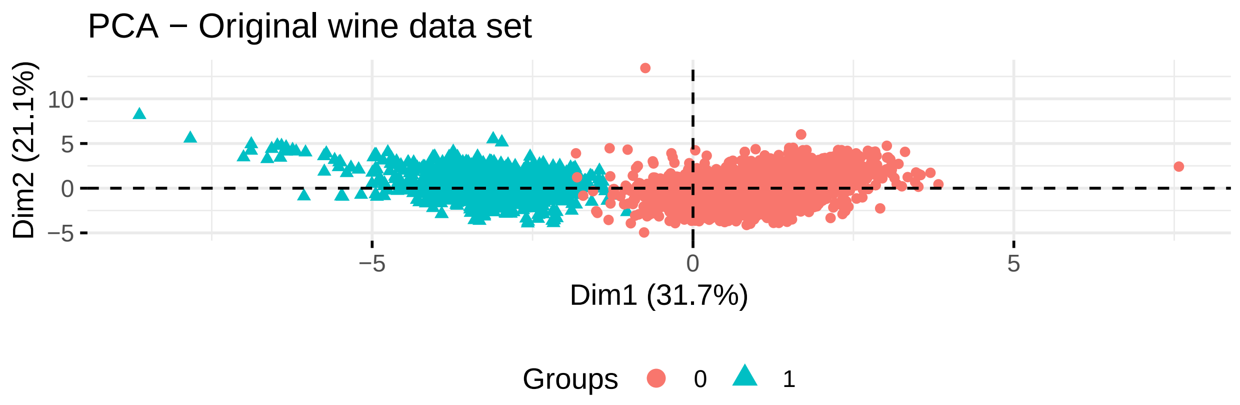
```
# We create a new win dataset which is generated randomly
random_dataset_wine <-  as.data.frame( apply(data_wine, 2, function(x) {
    runif(length(x), min(x), (max(x))) }) )

scaled_random_dataset_wine <- as.data.frame( scale(random_dataset_wine))
```

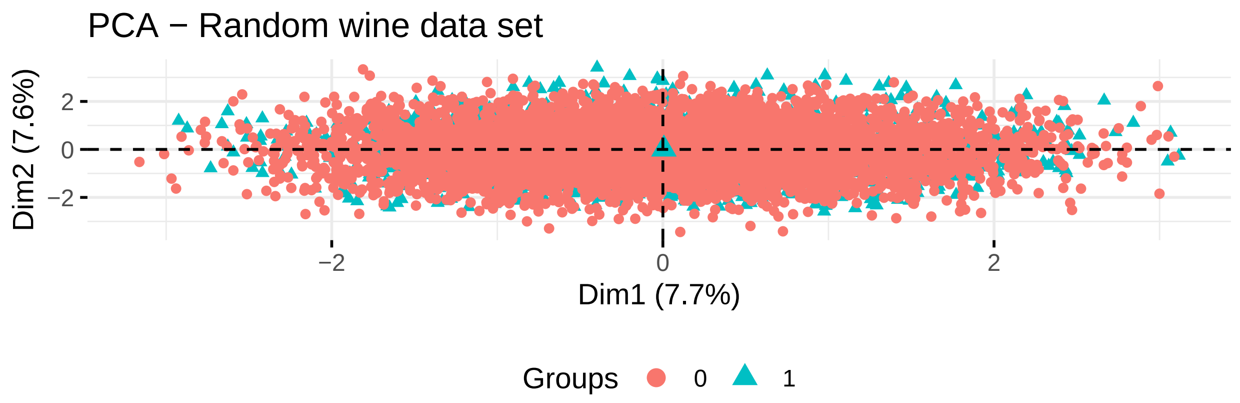
```r
head(scaled_random_dataset_wine)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar   chlorides
## 1 -0.8026733879       -0.3911714   0.1879288     -0.7446946 -0.03314292
## 2 -1.0848709879        1.1180161   0.2472231      0.9188023 -0.49154085
## 3 -0.0001179953       -1.0142763  -0.1104976     -0.1035954 -1.16401043
## 4  0.0890916042       -0.3336504   1.2280384      0.2091890 -0.51002767
## 5  0.9551244127       -0.7050160   1.6516889      1.0138996 -1.13543513
## 6 -0.5985994763       -1.6964842  -0.1350264     -0.7215075 -0.62499142
##   free.sulfur.dioxide total.sulfur.dioxide    density         pH  sulphates
## 1         -0.72264820         -0.415450806 -1.1978239 -0.5666100  1.5634846
## 2          1.50300315         -0.487215059 -0.2043935 -1.3021191  1.2257242
## 3         -0.50730768         -0.831269780 -1.1969927  0.8310095  0.5062353
## 4         -0.07655379         -0.009552158  0.9021747  1.0950135  1.1594807
## 5          1.24203011         -1.670101513  1.6737663  0.3568060 -1.5278126
## 6         -1.62216679          0.993817845 -1.2894723  1.1046026  1.4790508
##      alcohol      color     quality     cluster
## 1 -0.1421040 -1.6648430  0.17338013 -1.67054210
## 2  1.6253228  0.5834692 -1.68395368  0.09027573
## 3  1.1128410  1.2922150 -0.15627106 -0.33628186
## 4 -0.5160173 -1.4657653 -0.20156007  1.20766011
## 5 -0.7604312  0.8931076  0.09921602  0.67058505
## 6  1.3098455  0.4594090  1.14969097  0.82388289
```

In the above figures, we compare the real wine dataset and the random wine data set. It can be seen that the real win data set contains 2 real clusters. However the randomly generated data set doesn't contain any meaningful clusters.

```r
scaled_dw_plot <-
  fviz_pca_ind(
    prcomp(scaled_dataset_wine),
    title = "PCA - Original wine data set",
    habillage = data_wine$color,
    geom = "point",
    legend = "bottom"
  )
scaled_rdw_plot <-
  fviz_pca_ind(
    prcomp(scaled_random_dataset_wine),
    title = "PCA - Random wine data set",
    habillage = data_wine$color,
    geom = "point",
    legend = "bottom"
  )


grid.arrange(scaled_dw_plot,
             scaled_rdw_plot,
             nrow = 2,
             ncol = 1)
```

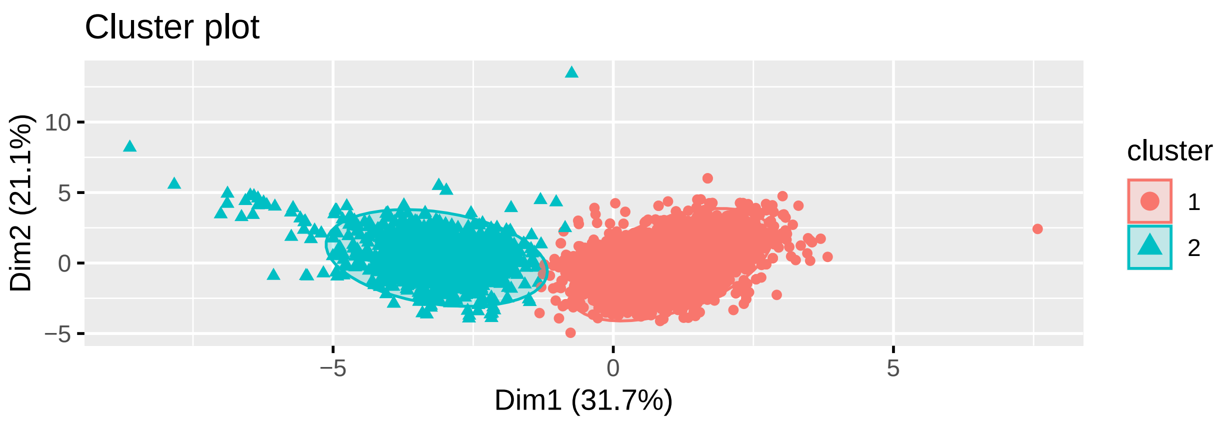


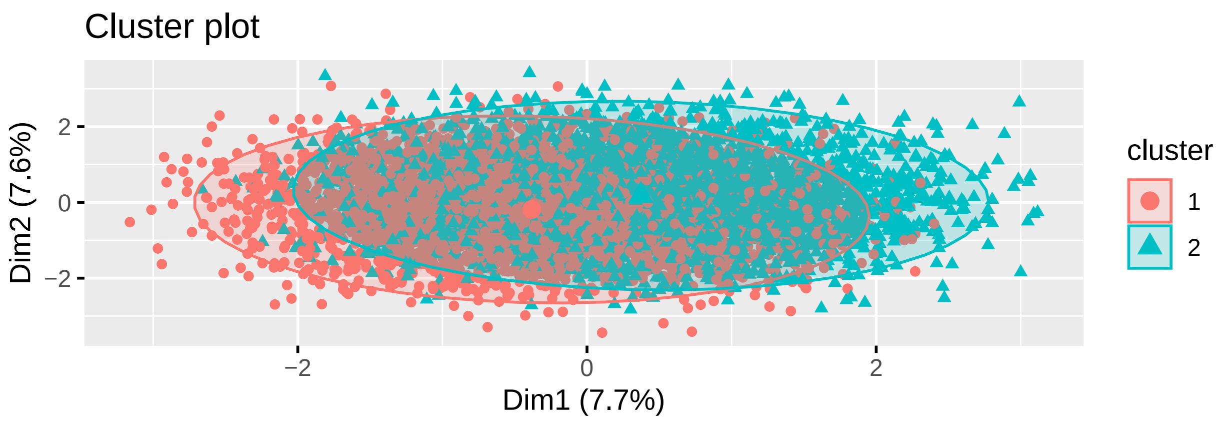


Another wat to compare the clustring results between the real and random dataset is to use the `fviz_cluster`function. The `fviz_cluster` function provides ggplot2-based elegant visualization of partitioning methods including kmeans [stats package]; pam, clara and fanny [cluster package]; dbscan [fpc package]; Mclust [mclust package]; HCPC [FactoMineR]; hkmeans [factoextra]. Observations are represented by points in the plot, using principal components if ncol(data) > 2. An ellipse is drawn around each cluster.

```r
km_dw <- kmeans(scaled_dataset_wine, 2, nstart = 20)
km_dw_aggregate <-
  aggregate(scaled_dataset_wine,
            by = list(km_dw$cluster),
            FUN = mean)

km_dw_plot <-
  fviz_cluster(
    list(data = scaled_dataset_wine, cluster = km_dw$cluster),
    ellipse.type = "norm",
    geom = "point",
    stand = FALSE
  )


km_rdw <- kmeans(scaled_random_dataset_wine, 2, nstart = 20)
km_rdw_aggregate <-
  aggregate(scaled_random_dataset_wine,
            by = list(km_rdw$cluster),
            FUN = mean)
```

```
km_rdw_plot <-
  fviz_cluster(
    list(data = scaled_random_dataset_wine, cluster = km_rdw$cluster),
    ellipse.type = "norm",
    geom = "point",
    stand = FALSE
  )


grid.arrange(km_dw_plot,
             km_rdw_plot,
             nrow = 2,
             ncol = 1)
```


Cluster plot
Dim2 (21.1%)
Dim1 (31.7%)
cluster


Cluster plot
Dim2 (7.6%)
Dim1 (7.7%)
cluster