1708-351-451-13a-ClusterAnalysis-2.Rmd

Roger H. French, JiQi Liu

20 November, 2018

Contents

13.1.2.1 Cluster Analysis: An example of unsupervised learning

13.1.2.1.1 Using k-means with public datasets

In what follows, we are going to learn more about partition clustering

- with k-means while exploring a dataset from the cluster.datasets package.

This package contains datasets that were published in the book,

- Clustering algorithms, by Hartigan (1975), with examples of analyses.

So let's start by installing this dataset on your machine, and loading it.

```
#### if(!require("cluster.datasets")) install.packages("cluster.datasets")
library(cluster.datasets)
```

Understanding the data

We will first focus on

- getting to know the data,
- scaling the data to a common metric,
- and cluster interpretability.

Our first exploration will concern the crime rates

- among different US cities in 1970.

The dataset all.us.city.crime.1970 affords such investigation:

```
data(all.us.city.crime.1970)
crime = all.us.city.crime.1970
```

Let's investigate the attributes in the dataset:

```
ncol(crime)
```

```
## [1] 10
```

```
names(crime)
```

```
##  [1] "city"             "population"       "white.change"
##  [4] "black.population" "murder"           "rape"
##  [7] "robbery"          "assault"          "burglary"
## [10] "car.theft"
```

```r
summary(crime)
```

```
##      city             population     white.change     black.population
##  Length:24          Min.   : 1268   Min.   :-39.400   Min.   :  39.0  
##  Class :character   1st Qu.: 1416   1st Qu.:-20.875   1st Qu.: 117.5  
##  Mode  :character   Median : 2024   Median :-13.450   Median : 302.0  
##                     Mean   : 2932   Mean   : -8.304   Mean   : 452.8  
##                     3rd Qu.: 2923   3rd Qu.:  6.750   3rd Qu.: 585.5  
##                     Max.   :11529   Max.   : 50.800   Max.   :2080.0  
##      murder            rape          robbery         assault     
##  Min.   : 2.600   Min.   : 5.70   Min.   : 53.0   Min.   : 63.0  
##  1st Qu.: 4.400   1st Qu.:16.40   1st Qu.:142.8   1st Qu.:106.8  
##  Median : 9.350   Median :20.20   Median :243.0   Median :157.0  
##  Mean   : 9.188   Mean   :23.18   Mean   :277.9   Mean   :187.8  
##  3rd Qu.:13.525   3rd Qu.:28.10   3rd Qu.:351.8   3rd Qu.:232.2  
##  Max.   :18.400   Max.   :50.00   Max.   :665.0   Max.   :421.0  
##     burglary      car.theft     
##  Min.   : 499   Min.   : 348.0  
##  1st Qu.: 854   1st Qu.: 523.2  
##  Median :1333   Median : 684.0  
##  Mean   :1313   Mean   : 679.6  
##  3rd Qu.:1660   3rd Qu.: 795.8  
##  Max.   :2164   Max.   :1208.0
```

There are 10 attributes.

A look at the R manual page allows us to understand what these variables are about.

```r
?all.us.city.crime.1970
```

Most of them are pretty obvious considering their name, and we will not comment further here.

Looking at the descriptive statistics,

- one can notice that there was a
  - quite important number of crimes in the 24 cities
  - for which data is available in this dataset:
- summing over murder, rape, robbery, assault, burglary, and car.theft,
  - around 2,500 crimes took place per 100,000 residents,
- which means that about 2.5 percent of the population
  - was the victim of a crime that year
- (considering that one person could only be a victim of one crime).

It might be interesting to know if

- cities differ in relation to the crimes that are committed.

We will manually explore several clustering solutions:

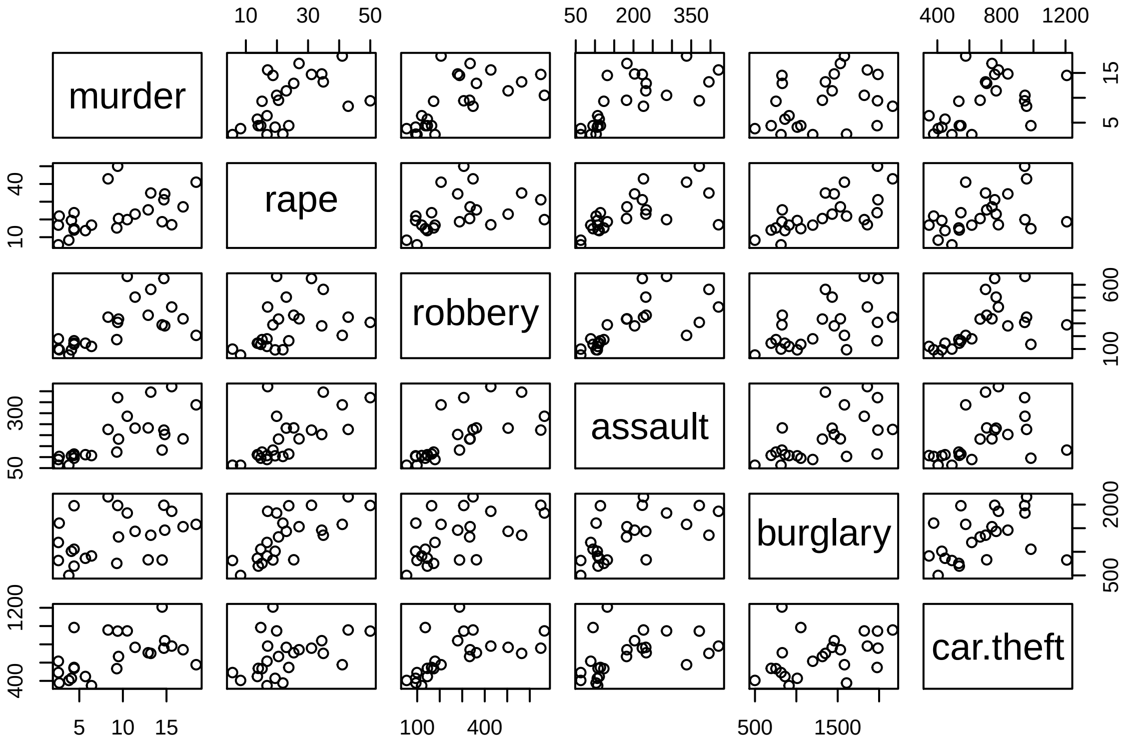
We will only consider here dimensions related to crime,

- which is attributes 5 to 10.

Before we run kmeans(),

- let's have a look at the relationship between the attributes.

```r
plot(crime[5:10])
```



As you can see on your screen,

- there is visibly a strong positive association
  - between the rate of some crimes (such as burglary and rape),
  - and a weaker for others (such as murder and burglary).

Overall it seems that the more of one crime type is committed,

- the more the others are as well.

We can confirm this intuition looking at the correlation matrix (rounded to 3 decimals).

```r
round(cor(crime[5:10]),3)
```

```
##           murder  rape robbery assault burglary car.theft
## murder     1.000 0.526   0.638   0.709    0.353     0.495
## rape       0.526 1.000   0.414   0.667    0.694     0.410
## robbery    0.638 0.414   1.000   0.699    0.551     0.559
## assault    0.709 0.667   0.699   1.000    0.596     0.428
## burglary   0.353 0.694   0.551   0.596    1.000     0.382
## car.theft  0.495 0.410   0.559   0.428    0.382     1.000
```

Yet, the relatively modest values of some correlations

- permits us to imagine a specialization of crime in some cities.

Lets run k-means on this dataset

We will run kmeans() on this dataset

- with an increasing number of clusters
  - (from 2 to 5),
- and will examine to solutions
  - visually and concurrently.

We will have a detailed look at the output

- of only the first and last clustering models, at the end.

We will let the reader modify the code

- with regards to the number of clusters.

We could have implemented a loop to do this,

- but we think it is more interesting
  - if you have a look at each solution individually at your pace.

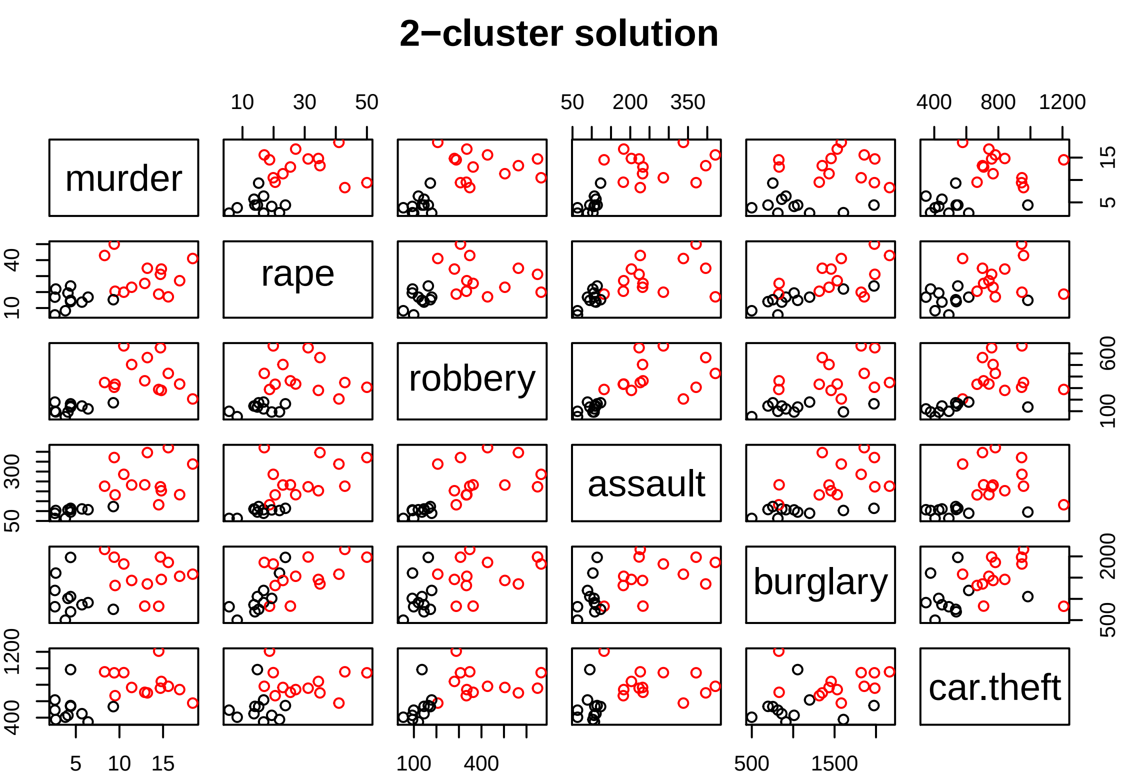
In all our models, we will ask k-means to repeat the procedures 25 times

- (using argument nstart)
- in order to be sure to have a good clustering solution.

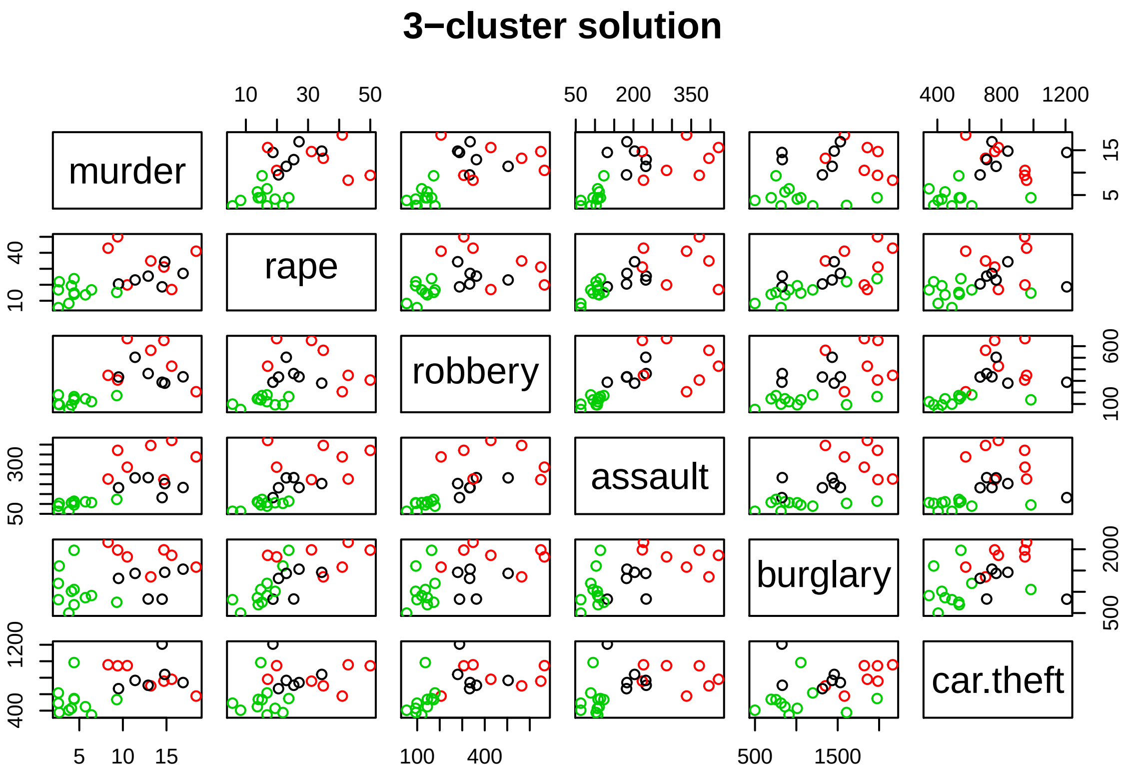
We will start by standardizing our data,

- in order to avoid one attribute
  - that is more important than the others
- in computing the distances.

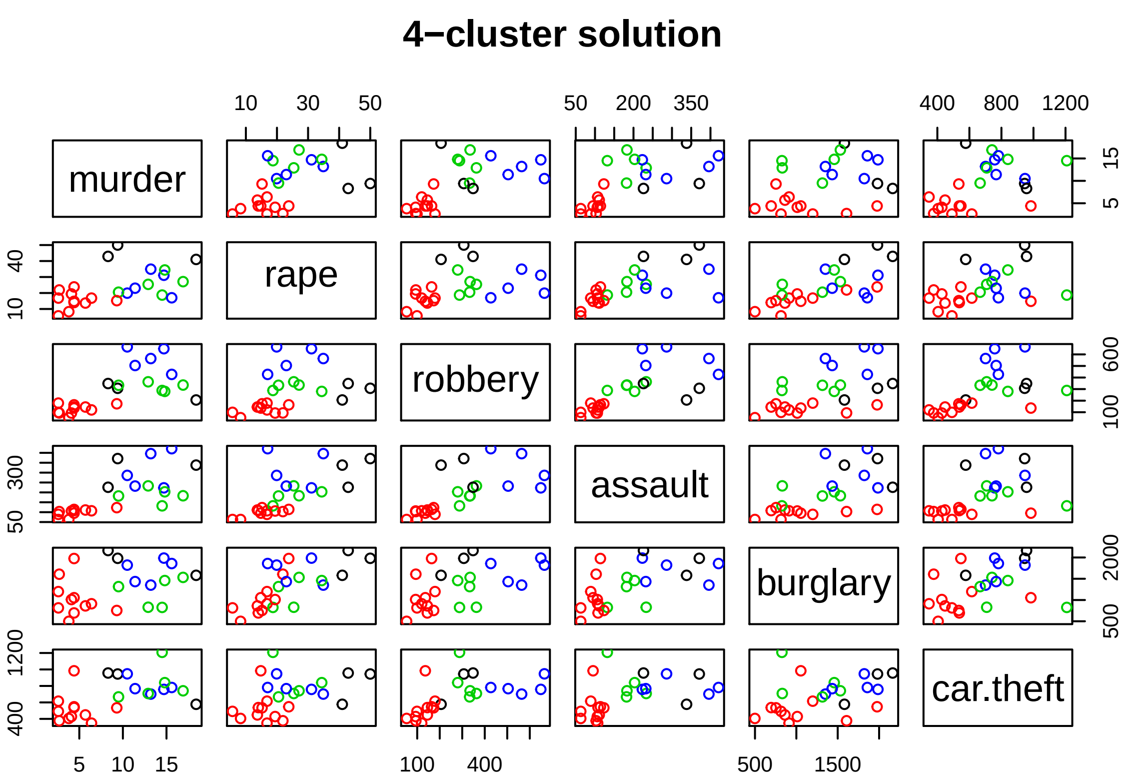
```
crime.scale = data.frame(scale(crime[5:10]))
set.seed(234)
TwoClusters = kmeans(crime.scale, 2, nstart = 25)
plot(crime[5:10],col = as.factor(TwoClusters$cluster), main = "2-cluster solution")
```



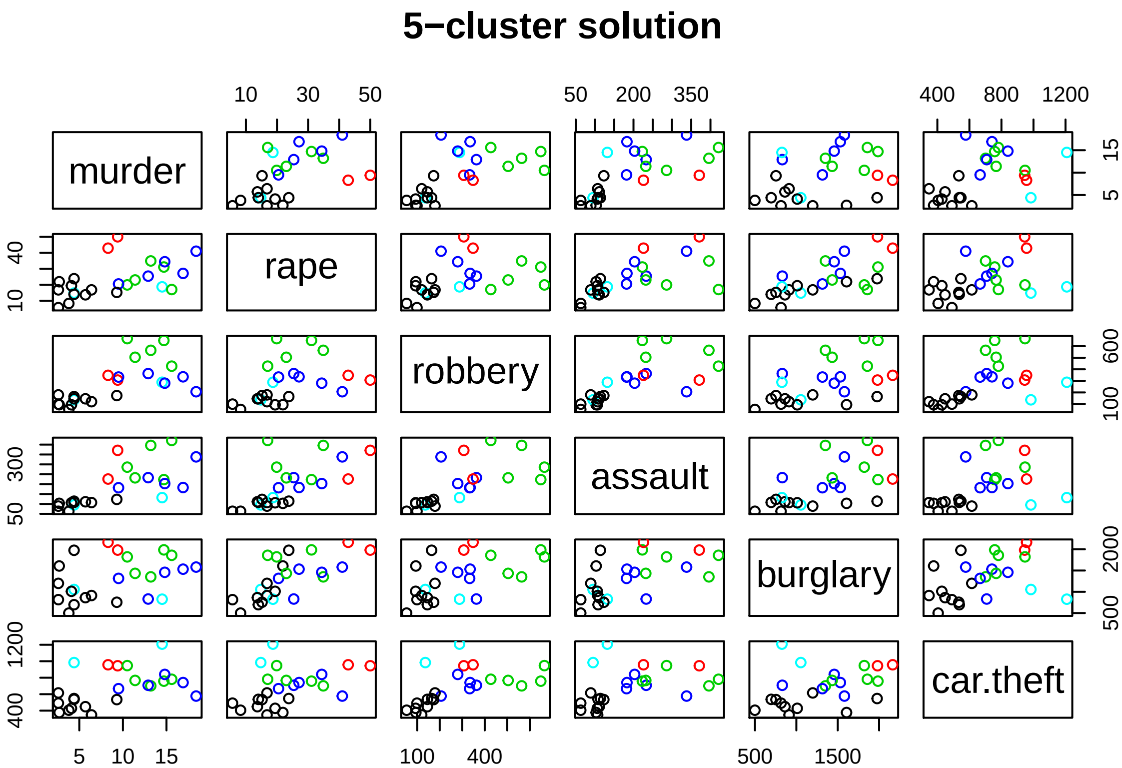

```
ThreeClusters = kmeans(crime.scale, 3, nstart = 25)
plot(crime[5:10],col = as.factor(ThreeClusters$cluster), main = "3-cluster solution")
```




```
FourClusters = kmeans(crime.scale, 4, nstart = 25)
plot(crime[5:10],col = as.factor(FourClusters$cluster), main = "4-cluster solution")
```




```
FiveClusters = kmeans(crime.scale, 5, nstart = 25)
plot(crime[5:10],col = as.factor(FiveClusters$cluster), main = "5-cluster solution")
```



The relationship between

- several types of crimes and
- cluster membership for k=2 to k=5

Lets check the interpretation of the clusters

To our domain knowledge

An important aspect of cluster analysis is the interpretation of the clusters.

- As can be seen in the preceding screenshot,
  - the interpretation of the clusters
  - in the 2-cluster solution is quite straightforward.
- Cities with a low criminality make up the black cluster,
  - whereas the red cluster is composed
  - of cities with higher criminality.

The pattern is more complex in the model with three clusters.

- At first sight, it seems that
  - this cluster is about a low average and high criminality.
- But this is denied by a closer inspection:
  - burglary and car.theft can be high in the green cluster,
  - rape and murder can be low to average, while assault and robbery are low.
- The black cluster seems to be concerned with cities with average crime.
- But looking more closely,
  - murder can be higher in this cluster than in the red one;
  - this is true to a lesser extent for rape and car.theft.
- We could consider this cluster as representing cities
  - with a high murder rate
  - and an average rate of other crimes.
- The red cluster is the most dispersed of the three,

    - yet it is the easiest to interpret.
- Cities in this cluster have average to high values
    - for all the study's dimensions of crime.
- The solutions with four and five clusters
    - are even more difficult to interpret.

It is usually advised to consider

- a number of clusters manageable for interpretation
    - (not hundreds of clusters) and that are meaningful,
    - even if a larger number of clusters explains the data better.

Let's now examine the textual output of R

- for our first (TwoClusters) solution.

```
TwoClusters
```

```
## K-means clustering with 2 clusters of sizes 11, 13
##
## Cluster means:
##       murder       rape    robbery    assault   burglary  car.theft
## 1 -0.9128346 -0.6991864 -0.8438639 -0.8328348 -0.5708682 -0.7166146
## 2  0.7723985  0.5916192  0.7140387  0.7047064  0.4830424  0.6063662
##
## Clustering vector:
##  [1] 1 2 1 1 2 1 2 2 2 2 2 2 1 1 2 2 1 1 1 2 2 1 1 2
##
## Within cluster sum of squares by cluster:
## [1] 18.39421 47.16265
##  (between_SS / total_SS =  52.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

The Cluster means reports the centroids

- for the final iteration of the algorithm,
    - usually when convergence is achieved.
- This information confirms our visual interpretation
    - of the clustering solution,
- one factor has high means on all crime dimensions,
    - whereas the other has low means.
- This section is directly accessible as data by typing:

```
TwoClusters$centers
```

```
##       murder       rape    robbery    assault   burglary  car.theft
## 1 -0.9128346 -0.6991864 -0.8438639 -0.8328348 -0.5708682 -0.7166146
## 2  0.7723985  0.5916192  0.7140387  0.7047064  0.4830424  0.6063662
```

The Clustering vector reports on the membership of the observations

- to each of the clusters for instance,
    - the first observation is part of cluster 2 (low criminality),
    - whereas the last is part of cluster 1 (average to high criminality).
- This section is directly accessible as data by typing:

```
TwoClusters$cluster
```

```
##  [1] 1 2 1 1 2 1 2 2 2 2 2 2 1 1 2 2 1 1 1 2 2 1 1 2
```

The section Within cluster sum of squares by cluster

- reports on the overall squared distance
  - between the data points and their centroid,
  - within each of the clusters.
- We can also see a division between
  - the between sum of squares (BSS)
  - and the total sum of square (TSS).
- The BSS refers to the overall squared difference,
  - for each data point,
  - between the mean of its centroid
  - and the overall mean.
- The TSS refers to the overall squared distance
  - of the data points
  - to the mean of all the means.

We can also see (under Available components)

- that we can examine other values we have not yet seen.
  - totss is the total sum of squares,
  - tot.withinss is the total of the sum of squares within clusters,
  - between ss is the total sum of squares within clusters,
  - size is the number of cases classified in each of the clusters,
  - iter is the number of iterations required for convergence,
  - and ifault signals warnings and problems
  - (with a value of 0 if there is no issue).

13.1.2.1.2 Compare the Between Sum of Squares (BSS) and Total Sum of Squares (TSS) for the clusters

We are now going to plot the differences between

- the value BSS/TSS for each of the clusters.

Basically, this value shows how much of the data

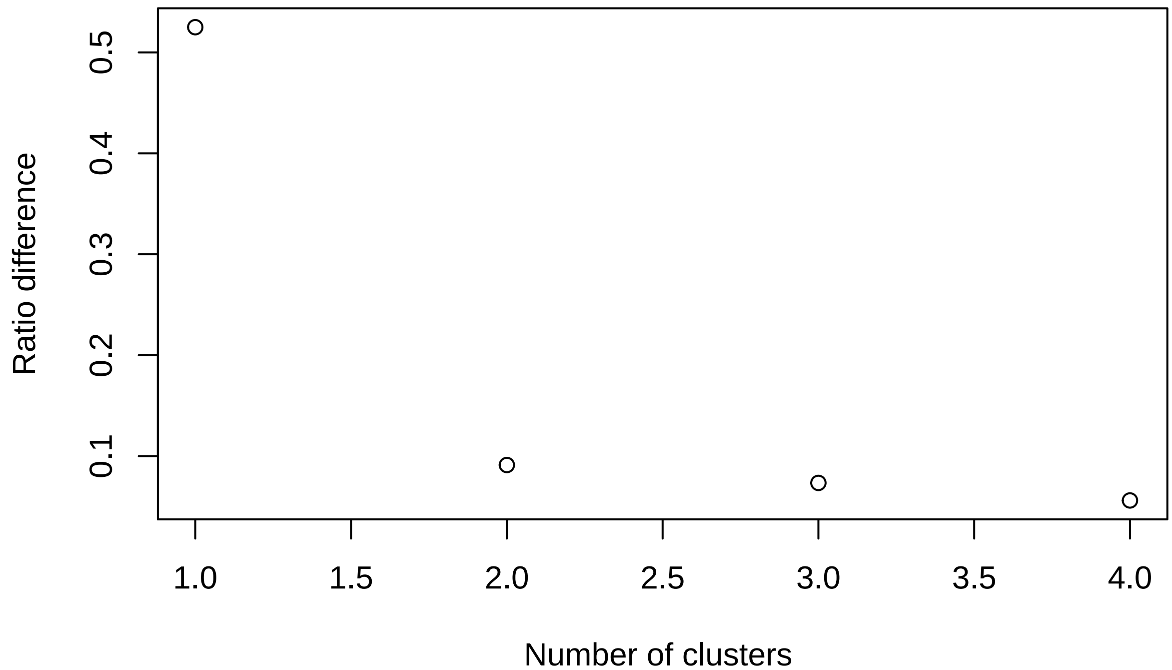
- is explained by the clustering solution,
- as it divides the BSS by the TSS.

It involves computing the ratio differences

- to a vector and using the plot() function.

The first value is the ratio for the TwoClusters model.

```
v = rep(0,4)
v[1] = TwoClusters[[6]]/TwoClusters[[3]]
v[2] = (ThreeClusters[[6]]/ThreeClusters[[3]]) - v[1]
v[3] = (FourClusters[[6]]/FourClusters[[3]]) - sum(v[1:2])
v[4] = (FiveClusters[[6]]/FiveClusters[[3]]) - sum(v[1:3])
plot(v, xlab = "Number of clusters ",
   ylab = "Ratio difference")
```



We can see in the preceding graph that

- the ratio is around .5 in the TwoClusters solution,
- and that it doesn't increase much with more clusters.

The TwoClusters solution should therefore be preferred.

Moreover, we have seen that

- solutions with more than two clusters are difficult to interpret.

A BSS/TSS = 1 is the best possible value,

- yet it will seldom be reached.

13.1.2.2 Cites

- Learning Predictive Analytics with R, Eric Mayor, Packtpub 2015