Example for k mediod for categorical data

R Markdown

input dataset for catergorical data

```
set.seed(123)
a <- matrix(sample(0:1, 1000*10, replace = TRUE), 1000, 10)
a1 <- matrix(sample(1:5, 1000*3, replace = TRUE), 1000, 5)
data <- cbind(a, a1)
colnames(data) <- c(paste(c("bin"), 1:10, sep = ""),paste(c("cat"), 1:5, sep
= ""))
data[1:10,1:15]
```

```
##       bin1 bin2 bin3 bin4 bin5 bin6 bin7 bin8 bin9 bin10 cat1 cat2 cat3
##  [1,]    0    0    0    0    0    0    0    1    1     1    2    1    2
##  [2,]    1    1    0    1    1    1    0    1    0     1    2    3    3
##  [3,]    0    0    0    0    1    1    0    0    1     0    5    1    5
##  [4,]    1    1    1    1    1    1    1    1    0     1    2    5    5
##  [5,]    1    1    0    0    0    0    0    0    1     1    1    1    2
##  [6,]    0    0    1    1    0    0    1    1    1     0    2    2    4
##  [7,]    1    1    0    1    1    0    0    0    0     0    5    5    3
##  [8,]    1    0    0    0    0    1    1    0    0     0    5    2    5
##  [9,]    1    0    0    1    1    1    1    1    0     1    5    1    4
## [10,]    0    0    0    1    1    0    0    1    1     0    1    1    1
##       cat4 cat5
##  [1,]    2    1
##  [2,]    2    3
##  [3,]    5    1
##  [4,]    2    5
##  [5,]    1    1
##  [6,]    2    2
##  [7,]    5    5
##  [8,]    5    2
##  [9,]    5    1
## [10,]    1    1
```

calculate the dissimilar matrix for the categorical data

```
dis_jaccard=distance(data,method="jaccard")
dis_jaccard[1:20]
```

```
##  [1] 0.4615385 0.4000000 0.5333333 0.3000000 0.3636364 0.5833333 0.5454545
##  [8] 0.4615385 0.3000000 0.3636364 0.4166667 0.3333333 0.4615385 0.4000000
## [15] 0.5384615 0.5384615 0.2000000 0.3000000 0.4545455 0.4166667
```

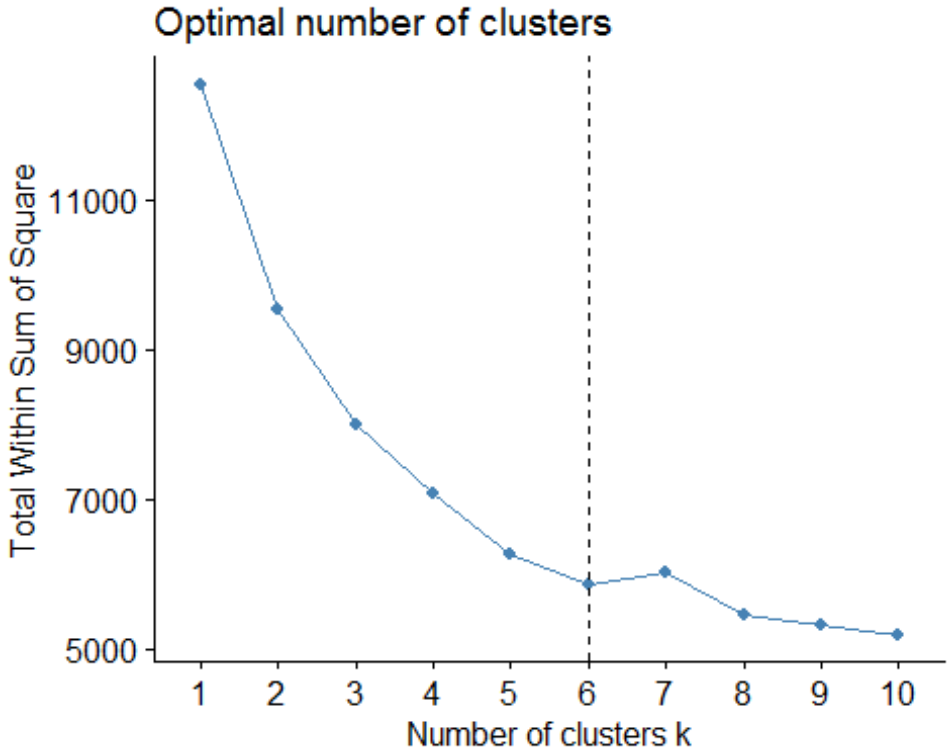
plot.partition clusplot clara.obbject clusGap

Find the best k

Elbow method:

Recall that, the basic idea behind partitioning methods, such as k-means clustering, is to define clusters such that the total intra-cluster variation (known as total within-cluster variation or total within-cluster sum of square) is minimized: $minimize(\sum_{k=1}^{k} W\ (Ck))$

```
fviz_nbclust(data,clara,method="wss") +
  geom_vline(xintercept = 6, linetype = 2)
```



Gap statistic method

The gap statistic compares the total within intracluster variation for different values of k with their expected values under null reference distribution of the data, i.e. a distribution with no obvious clustering.

The reference dataset is generated using Monte Carlo simulations of the sampling process. That is, for each variable (x_i) in the data set we compute its range $[min(x_i), max(x_j)]$ and generate values for the n points uniformly from the interval min to max.

For the observed data and the the reference data, the total intracluster variation is computed using different values of k. The gap statistic for a given k is defined as follow:

$Gap_n(k) = E_n^* log(W_k) - log(W_k)$

Where$E_n^* log(W_k)$ denotes the expectation under a sample of size n from the reference distribution. E???n is defined via bootstrapping (B) by generating B copies of the reference datasets and, by computing the average $log(W_k^*)$.

Note that, the logarithm of the W_k values is used, as they can be quite large.

The gap statistic measures the deviation of the observed W_k value from its expected value under the null hypothesis.

The estimate of the optimal clusters k will be value that maximize $Gapn(k)$ (i.e, that yields the largest gap statistic). This means that the clustering structure is far away from the uniform distribution of points.

The standard deviation (sd_k) of $log(W_k^*)$ is also computed in order to define the standard error (sk) of the simulation as follow:

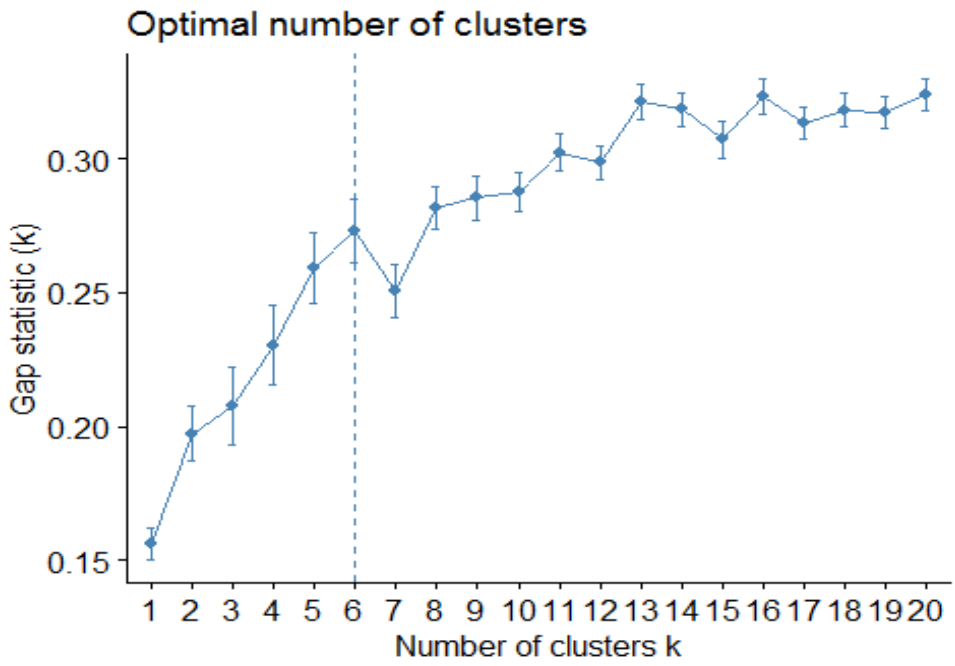
$$s_k = sd_k \times \sqrt{1 + 1/B}$$

Finally, a more robust approach is to choose the optimal number of clusters K as the smallest k such that:

$$Gap(k) > Gap(k+1) - s_{k+1}$$

That is, we choose the smallest value of k such that the gap statistic is within one standard deviation of the gap at k+1.

```
set.seed(123)
gap_stat <- clusGap(x=data, FUN = clara, K.max =20)
# Plot gap statistic
fviz_gap_stat(gap_stat)
```



```r
set.seed(123)
clara=clara(data, k=6, metric = c("jaccard"),samples = 50)
clara
```

```
## Call:     clara(x = data, k = 6, metric = c("jaccard"), samples = 50)
## Medoids:
##      bin1 bin2 bin3 bin4 bin5 bin6 bin7 bin8 bin9 bin10 cat1 cat2 cat3
## [1,]    0    0    1    1    1    1    1    1    1     1    5    4    1
## [2,]    1    1    0    1    0    1    0    1    0     1    2    2    3
## [3,]    0    1    0    1    1    0    0    0    1     0    4    2    2
## [4,]    1    1    1    1    0    0    0    0    0     1    1    2    2
## [5,]    1    0    1    0    0    1    1    0    0     1    4    4    3
## [6,]    1    1    0    0    1    0    1    1    1     0    5    2    3
##      cat4 cat5
## [1,]    5    4
## [2,]    2    2
## [3,]    4    2
## [4,]    1    2
## [5,]    4    4
## [6,]    5    2
## Objective function:   0.2368056
## Clustering vector:    int [1:1000] 2 2 6 2 4 2 6 6 6 4 5 3 5 4 5 5 4 3 ...
## Cluster sizes:            104 246 164 144 239 103
## Best sample:
##  [1]  28  42  52  65  82  97 100 107 125 145 146 153 161 169 176 185 234
## [18] 253 259 280 283 297 308 310 316 330 394 398 406 410 422 484 493 499
## [35] 524 525 528 547 624 636 664 695 742 748 768 802 854 913 917 939 946
## [52] 978
##
## Available components:
##  [1] "sample"     "medoids"    "i.med"      "clustering" "objective"
##  [6] "clusinfo"   "diss"       "call"       "silinfo"    "data"
```

```r
dd <- cbind(data, cluster = clara$clustering)
dd[1:20,]
```

```
##       bin1 bin2 bin3 bin4 bin5 bin6 bin7 bin8 bin9 bin10 cat1 cat2 cat3
##  [1,]    0    0    0    0    0    0    0    1    1     1    2    1    2
##  [2,]    1    1    0    1    1    1    0    1    0     1    2    3    3
##  [3,]    0    0    0    0    1    1    0    0    1     0    5    1    5
##  [4,]    1    1    1    1    1    1    1    1    0     1    2    5    5
##  [5,]    1    1    0    0    0    0    0    0    1     1    1    1    2
##  [6,]    0    0    1    1    0    0    1    1    1     0    2    2    4
##  [7,]    1    1    0    1    1    0    0    0    0     0    5    5    3
##  [8,]    1    0    0    0    0    1    1    0    0     0    5    2    5
##  [9,]    1    0    0    1    1    1    1    1    0     1    5    1    4
## [10,]    0    0    0    1    1    0    0    1    1     0    1    1    1
## [11,]    1    0    1    1    0    0    0    0    1     1    4    4    2
## [12,]    0    0    0    1    1    1    1    1    1     0    4    2    1
## [13,]    1    0    1    1    0    0    1    1    1     1    4    5    3
```

```
## [14,]    1    1    1    0    0    1    1    1    0    1    1    5    2
## [15,]    0    0    1    0    0    1    0    0    1    0    3    3    5
## [16,]    1    1    0    0    1    1    1    0    0    1    3    5    5
## [17,]    0    1    1    1    1    0    1    0    0    1    2    3    2
## [18,]    0    1    0    0    1    0    0    1    1    1    3    5    4
## [19,]    0    0    0    1    0    1    0    1    1    0    2    4    1
## [20,]    1    0    0    0    0    1    1    1    0    0    5    2    4
##       cat4 cat5 cluster
##  [1,]    2    1       2
##  [2,]    2    3       2
##  [3,]    5    1       6
##  [4,]    2    5       2
##  [5,]    1    1       4
##  [6,]    2    2       2
##  [7,]    5    5       6
##  [8,]    5    2       6
##  [9,]    5    1       6
## [10,]    1    1       4
## [11,]    4    4       5
## [12,]    4    2       3
## [13,]    4    5       5
## [14,]    1    5       4
## [15,]    3    3       5
## [16,]    3    5       5
## [17,]    2    3       4
## [18,]    3    5       3
## [19,]    2    4       1
## [20,]    5    2       6
```

```
cluster=as.matrix(clara$clustering)
cbind(table(cluster),clara$clusinfo)
```

```
##       size  max_diss   av_diss isolation
## 1 104  104 0.3846154 0.2340053  1.076923
## 2 246  246 0.3636364 0.2084413  1.454545
## 3 164  164 0.4000000 0.2348017  1.200000
## 4 144  144 0.3333333 0.2099303  1.333333
## 5 239  239 0.4545455 0.2648536  1.363636
## 6 103  103 0.3636364 0.2397637  1.090909
```