Métodos Quantitativos

Prof. Dr. A. L. Korzenowski

Aula 04: Análise de Clusters

Análise de Clusters ou *Cluster Analysis* tem por objetivo determinar padrões de variáveis que categorizem grupos, isto é, que identifiquem grupos de indivíduos.

Quanto ao posicionamento, assim como nos modelos de análise fatorial, a análise de clusters prevê no modelo apenas entradas (variáveis independentes) métricas.

Difere-se da análise discriminante (ou regressão Logística, por ex.) uma vez que nestas, eu sei a qual grupo os elementos da amostra pertencem.

Na análise de cluster o agrupamento dos elementos da amostra em conglomerados se dá pelas similaridades nas respostas das k variáveis independentes utilizadas na análise.

A expressão *Cluster analysis* foi utilizada pela primeira vez por Tryon, em 1939.

Características

- Detecta grupos homogêneos nos dados
- Grupamento de indivíduos ou objetos em grupos desconhecidos
- Não faz distinção entre variáveis dependentes e independentes

Observações

- A análise de aglomerados pode ser caracterizada como uma análise descritiva, não teórica e não inferencial.
- É utilizada principalmente com uma técnica exploratória.

Suposições

- Amostra deve ser representativa da população
- Multicolinearidade mínima
  - Devo evitar que duas ou mais variáveis repliquem o mesmo comportamento entre os grupos
  - Analisar correlações ($> 0{,}8$ pode representar um multicolinearidade)
- Amostra deve estar livre de *outliers*: Consulta visual via diagrama de perfis ou box-plots
- Razão (núm casos / núm clusters) deve ser razoável

Entre os principais **Critérios de medição da distância** estão:

- Distância Euclidiana: $D(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- Distância Euclidiana ao quadrado
- Distância de Manhattan: $D(x, y) = \sum_{i=1}^{n} |x_i - y_i|$
- entre outras...

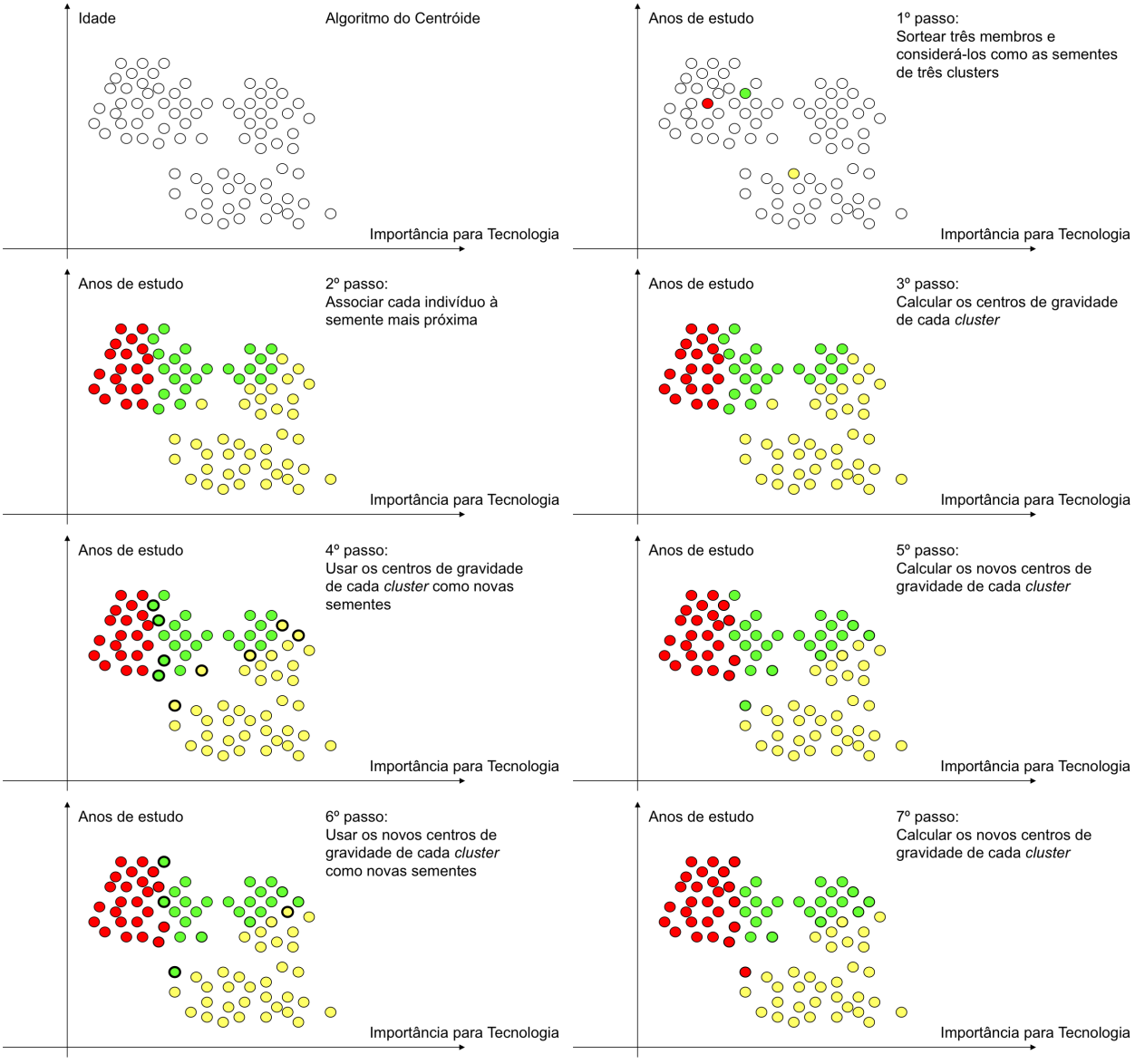
O R tem uma vasta variedade de funções para Análise de Clusters. Na aula de hoje nós vamos discutir uma das principais abordagens: Partitioning ou K-menas ou K-médias. O algoritmo k-means é um algoritmo que utiliza o cálculo de kcentróides para o agrupamento dos k clusters. Embora não haja uma solução ótima para o problema de determinar o número de clusters a serem extraídos, uma abordagem promissora é discutida na aula de hoje.

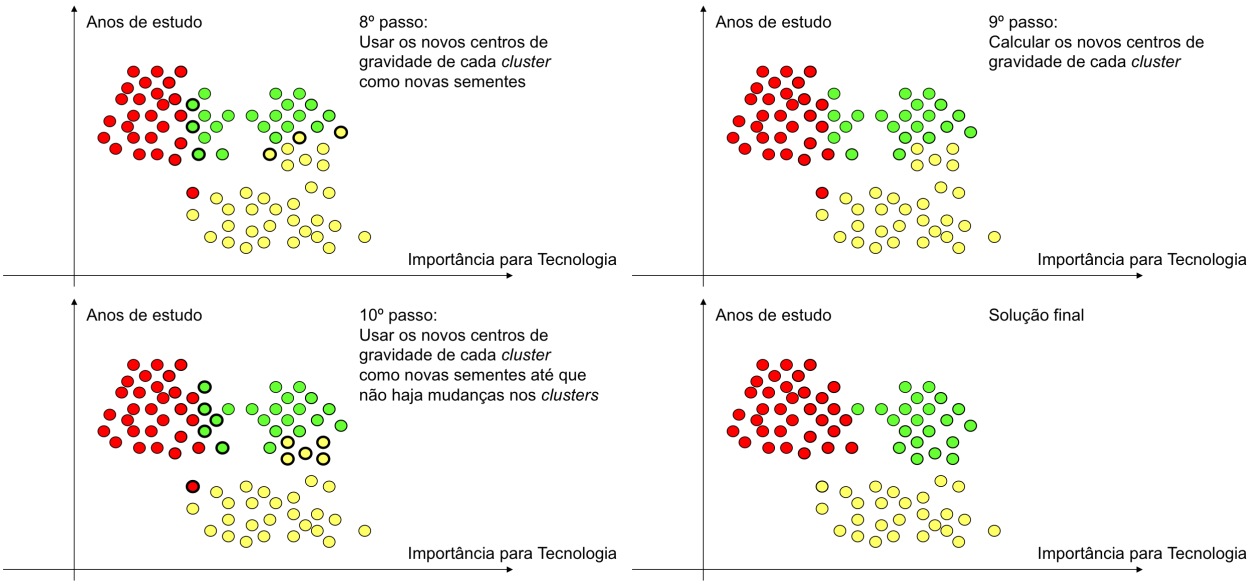
K-means

O cluster K-means é o método de particionamento mais popular. Requer que o analista especifique o número de clusters a serem extraídos. Uma plotagem da soma dos quadrados dentro dos grupos pelo número de clusters extraídos pode ajudar a determinar o número apropriado de clusters.

Note que a aplicação do método traz algumas exigências como o fato de lidar apenas com variáveis quantitativas (preferêncialmente contínuas), não haver dados faltantes (ver a função **na.omit**) e os dados serem inicialmente padronizados (ver a função **scale**).

A sequencia de imagens a seguir exemplifica o funcionamento do método para onde deseja-se agrupar os respondentes pelo número de anos de estudo e importância para a tecnologia.



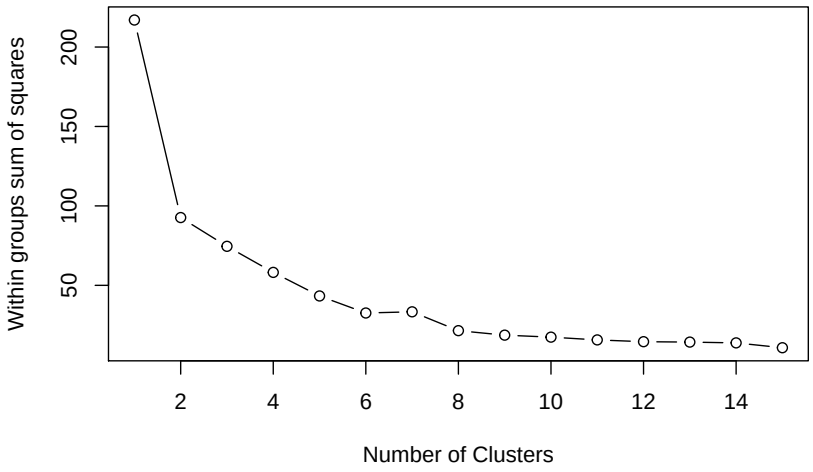


Considere inicialmente a base de dados **mtcars**. Vamos isolar as variáveis quantitativas de interesse, excluir os casos com dados faltantes e padronizar as variáveis. Na sequência, vamos construir um gráfico para escolher o número de agrupamentos.

```
require(car)
attach(mtcars)
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
cluster<-na.omit(mtcars)
cluster<-scale(cluster)
.cluster <- cluster[,1:7] # Apenas variáveis quantitativas
# Determine number of clusters
wss <- (nrow(.cluster)-1)*sum(apply(.cluster,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(.cluster, centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
ylab="Within groups sum of squares")
```



A análise do gráfico é similar a análise do scree-plot, que veremos mais detalhadamente na aula de Análise fatorial. Por enquanto, basta saber que devemos escolher o ponto de corte que indica o número de clusters quando as inclinações do gráfico tendem a ser iguais a zero.

Uma vez definido o número clusters, o R possui funções para a identfiicação destes. Execute o conjunto de comandos a seguir:

```
# K-Means Cluster Analysis
fit <- kmeans(.cluster, 6) # 6 cluster solution
# get cluster means
aggregate(.cluster,by=list(fit$cluster),FUN=mean)
```

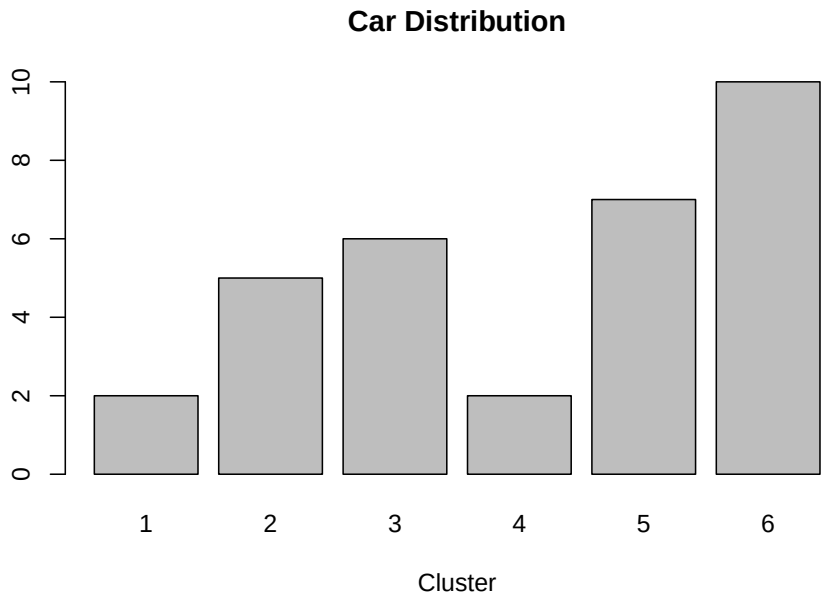
```
##   Group.1         mpg        cyl       disp         hp       drat         wt
## 1       1 -0.77827533  1.0148821  0.7687521  2.2287939  0.5301080  0.1561131
## 2       2  0.25707456 -0.7769098 -0.4244185 -0.7713723 -0.3115188 -0.1239195
## 3       3  1.65523937 -1.2248578 -1.1624447 -1.0382807  1.2252295 -1.3738462
## 4       4 -1.04374966  1.0148821  1.0027387  1.4339030 -0.2367076  0.4984887
## 5       5  0.05370226 -0.4249507 -0.6752799 -0.3829905  0.5461390 -0.3322650
## 6       6 -0.79486748  1.0148821  1.0280738  0.5442086 -1.0203557  0.9879326
##         qsec
## 1 -1.8460295
## 2  1.4915135
## 3  0.3075550
## 4 -1.2444436
## 5 -0.1240147
## 6 -0.2253849
```

```
# append cluster assignment
.cluster <- data.frame(.cluster, fit$cluster)
head(.cluster)
```

```
##                           mpg        cyl        disp         hp       drat
## Mazda RX4           0.1508848 -0.1049878 -0.57061982 -0.5350928  0.5675137
## Mazda RX4 Wag       0.1508848 -0.1049878 -0.57061982 -0.5350928  0.5675137
## Datsun 710          0.4495434 -1.2248578 -0.99018209 -0.7830405  0.4739996
## Hornet 4 Drive      0.2172534 -0.1049878  0.22009369 -0.5350928 -0.9661175
## Hornet Sportabout -0.2307345  1.0148821  1.04308123  0.4129422 -0.8351978
```

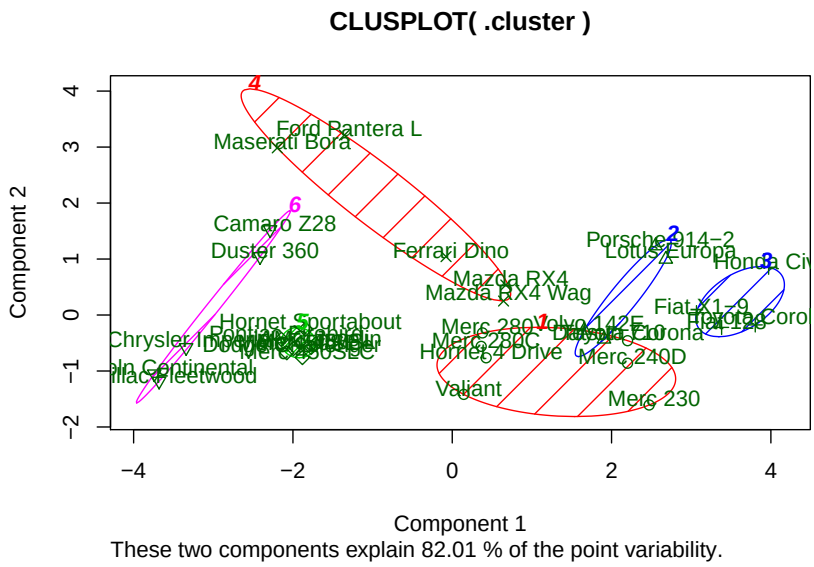
```
## Valiant           -0.3302874 -0.1049878 -0.04616698 -0.6080186 -1.5646078
##                           wt       qsec fit.cluster
## Mazda RX4         -0.610399567 -0.7771651           5
## Mazda RX4 Wag     -0.349785269 -0.4637808           5
## Datsun 710        -0.917004624  0.4260068           5
## Hornet 4 Drive    -0.002299538  0.8904872           2
## Hornet Sportabout  0.227654255 -0.4637808           6
## Valiant            0.248094592  1.3269868           2
```

Um diagrama de colunas pode mostrar a distribuição dos casos entre os clusters. **Execute um código apropriado para construir um gráfico similar ao apresentado a seguir**.

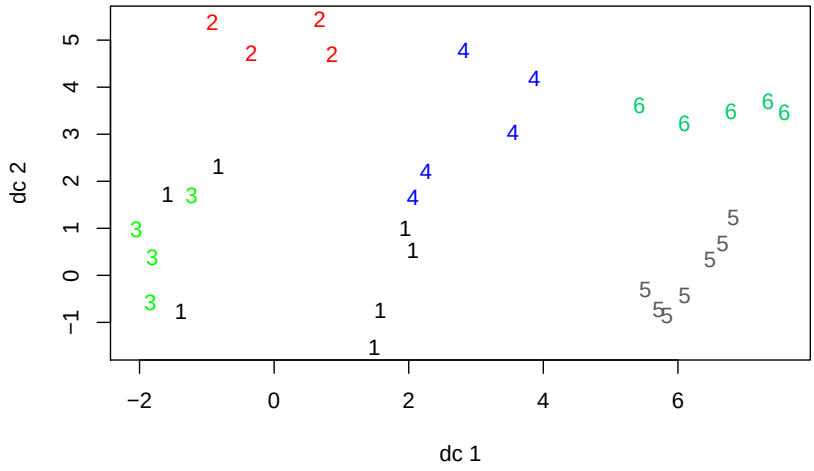


É sempre uma boa idéia analisar visualmente os resultados dos clusters...

```
# K-Means Clustering with 6 clusters
fit <- kmeans(cluster, 6)

# Cluster Plot against 1st 2 principal components

# vary parameters for most readable graph
library(cluster)
clusplot(.cluster, fit$cluster, color=TRUE, shade=TRUE,
   labels=2, lines=0)

# Centroid Plot against 1st 2 discriminant functions
library(fpc)
```




```
plotcluster(.cluster, fit$cluster)
```



Validando soluções de cluster

A função **cluster.stats()** no pacote **fpc** fornece um mecanismo para comparar a semelhança de duas soluções de cluster usando uma variedade de critérios de validação (coeficiente gama de Hubert, o índice Dunn e o índice rand corrigido) - Faça uma pesquisa rápida pelo Wikipedia (em inglês) para identificar o que são e como são analisados os resultados destes testes ou busque a ajuda do R para referências.

```
fit.4 <- kmeans(.cluster, 5)
fit.6 <- kmeans(.cluster, 6)

# comparing 2 cluster solutions
library(fpc)
d <- dist(.cluster)
?cluster.stats
cluster.stats(d, fit.6$cluster, fit.4$cluster)
```

```
## $n
## [1] 32
##
## $cluster.number
## [1] 6
##
## $cluster.size
## [1] 6 3 9 5 2 7
##
## $min.cluster.size
## [1] 2
##
## $noisen
## [1] 0
##
## $diameter
## [1] 2.521042 1.063531 3.191473 3.066201 1.743684 2.542386
##
## $average.distance
## [1] 1.6939630 0.7557628 1.6294047 2.0835688 1.7436844 1.5386806
##
## $median.distance
## [1] 1.7281257 0.9080748 1.1323655 2.0925288 1.7436844 1.5502951
##
## $separation
## [1] 1.850641 1.943421 1.943421 1.850641 3.288921 2.265725
##
## $average.toother
## [1] 4.936936 5.025395 4.482797 4.640351 5.706625 3.837261
##
## $separation.matrix
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
## [1,] 0.000000 6.650141 4.918375 1.850641 4.948767 2.265725
## [2,] 6.650141 0.000000 1.943421 5.459398 5.892630 3.971922
## [3,] 4.918375 1.943421 0.000000 3.930084 3.288921 2.539290
## [4,] 1.850641 5.459398 3.930084 0.000000 4.573405 3.126330
## [5,] 4.948767 5.892630 3.288921 4.573405 0.000000 4.884672
## [6,] 2.265725 3.971922 2.539290 3.126330 4.884672 0.000000
##
## $ave.between.matrix
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
## [1,] 0.000000 7.441778 5.854110 3.400871 6.199997 3.420525
## [2,] 7.441778 0.000000 2.600955 6.325449 6.205864 4.805458
## [3,] 5.854110 2.600955 0.000000 5.256948 5.344541 3.314711
## [4,] 3.400871 6.325449 5.256948 0.000000 5.471015 3.950478
## [5,] 6.199997 6.205864 5.344541 5.471015 0.000000 5.703605
## [6,] 3.420525 4.805458 3.314711 3.950478 5.703605 0.000000
##
## $average.between
## [1] 4.604483
##
## $average.within
## [1] 1.617865
##
```

```
## $n.between
## [1] 410
##
## $n.within
## [1] 86
##
## $max.diameter
## [1] 3.191473
##
## $min.separation
## [1] 1.850641
##
## $within.cluster.ss
## [1] 41.74265
##
## $clus.avg.silwidths
##         1         2         3         4         5         6
## 0.4732446 0.7057533 0.3834517 0.3617335 0.6725498 0.4736131
##
## $avg.silwidth
## [1] 0.4649016
##
## $g2
## NULL
##
## $g3
## NULL
##
## $pearsongamma
## [1] 0.6384547
##
## $dunn
## [1] 0.5798704
##
## $dunn2
## [1] 1.248318
##
## $entropy
## [1] 1.688352
##
## $wb.ratio
## [1] 0.3513674
##
## $ch
## [1] 33.07111
##
## $cwidegap
## [1] 1.6687847 0.9080748 2.4579751 2.0491218 1.7436844 1.4764389
##
## $widestgap
## [1] 2.457975
##
## $sindex
## [1] 1.881567
```

```
##
## $corrected.rand
## [1] 0.4974606
##
## $vi
## [1] 0.8172869
```

Atividade

1. Identificar os tipos de testes apresentados pela função **cluster.stats()** e o que significam seus resultados.