Universidade de Brasília

DEPARTAMENTO DE ESTATÍSTICA

17 julho 2023

Lista 9 - Análise de Agrupamentos

Prof. Dr. George von Borries

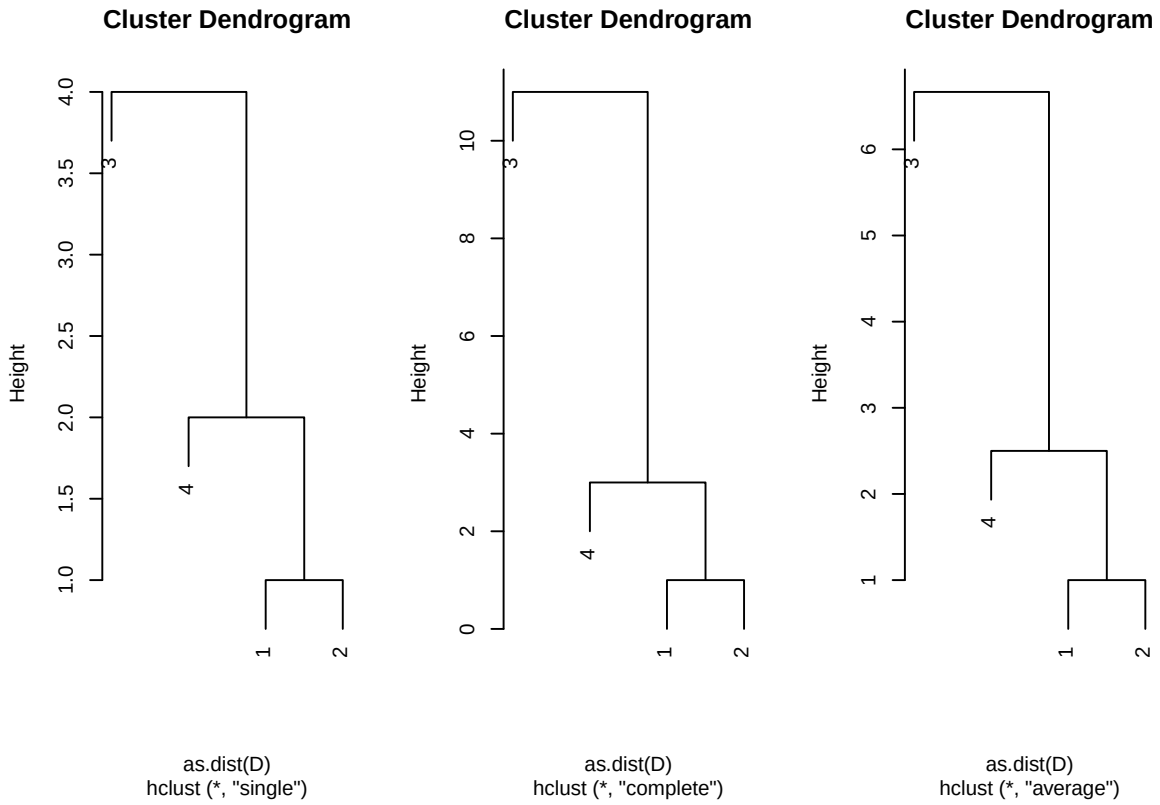
Análise Multivariada 1

Aluno: Bruno Gondim Toledo | Matrícula: 15/0167636

76. Johnson e Wichern - Exercício 12.3.

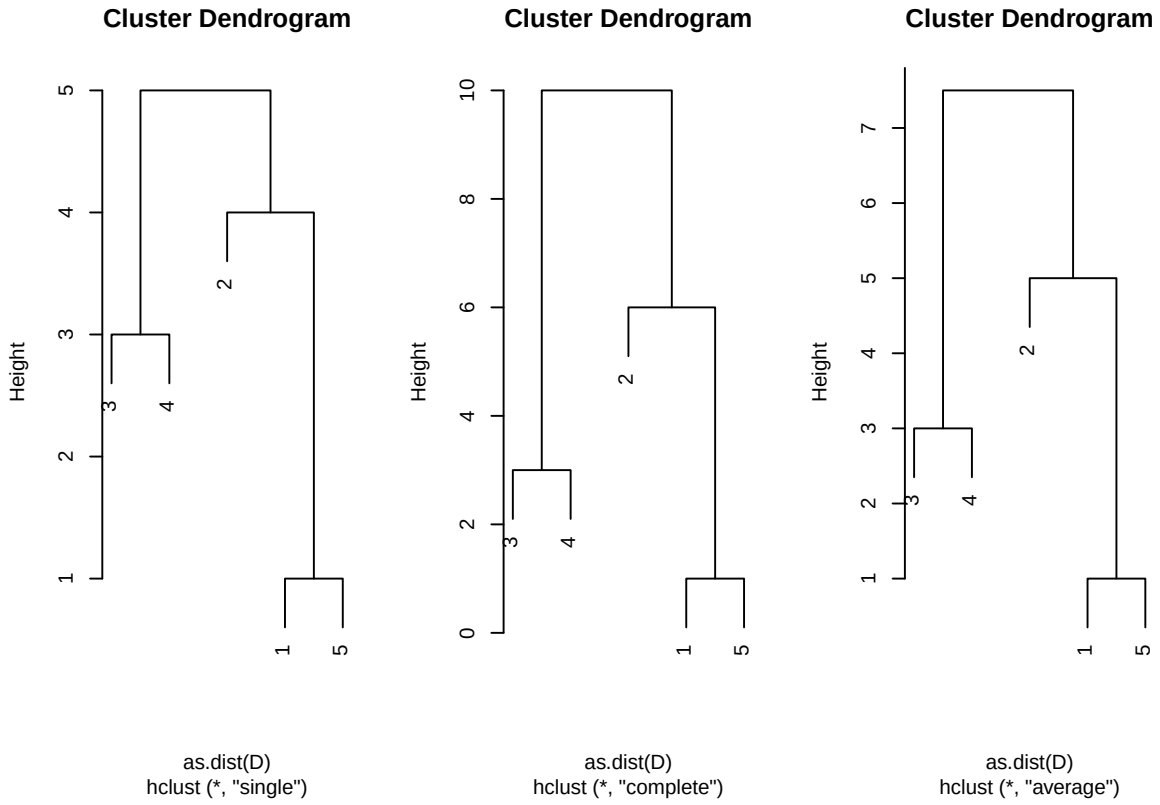
77. Johnson e Wichern - Exercício 12.5.

a), b) e c):



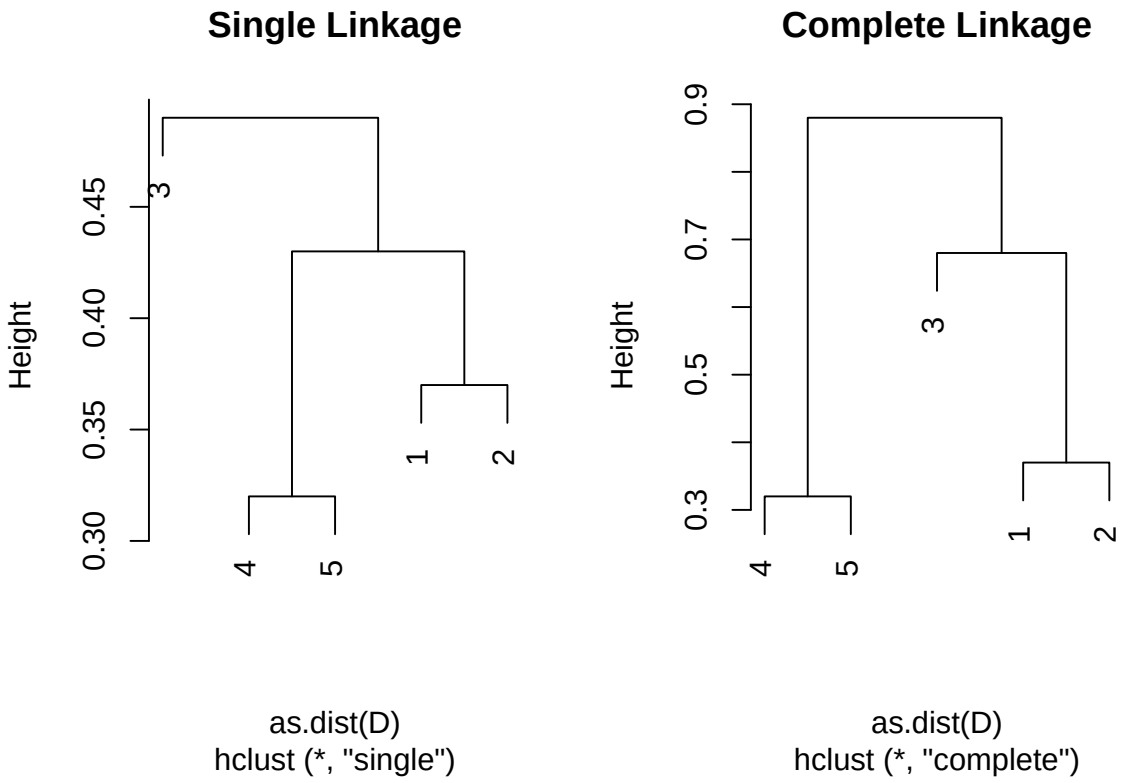
Analisando os dendogramas, percebemos que tanto as abordagens simples, média e completa, agregaram os valores da exata mesma maneira.

78. Johnson e Wichern - Exercício 12.6.



Analisando os dendogramas, percebemos que tanto as abordagens simples, média e completa, agregaram os valores da exata mesma maneira.

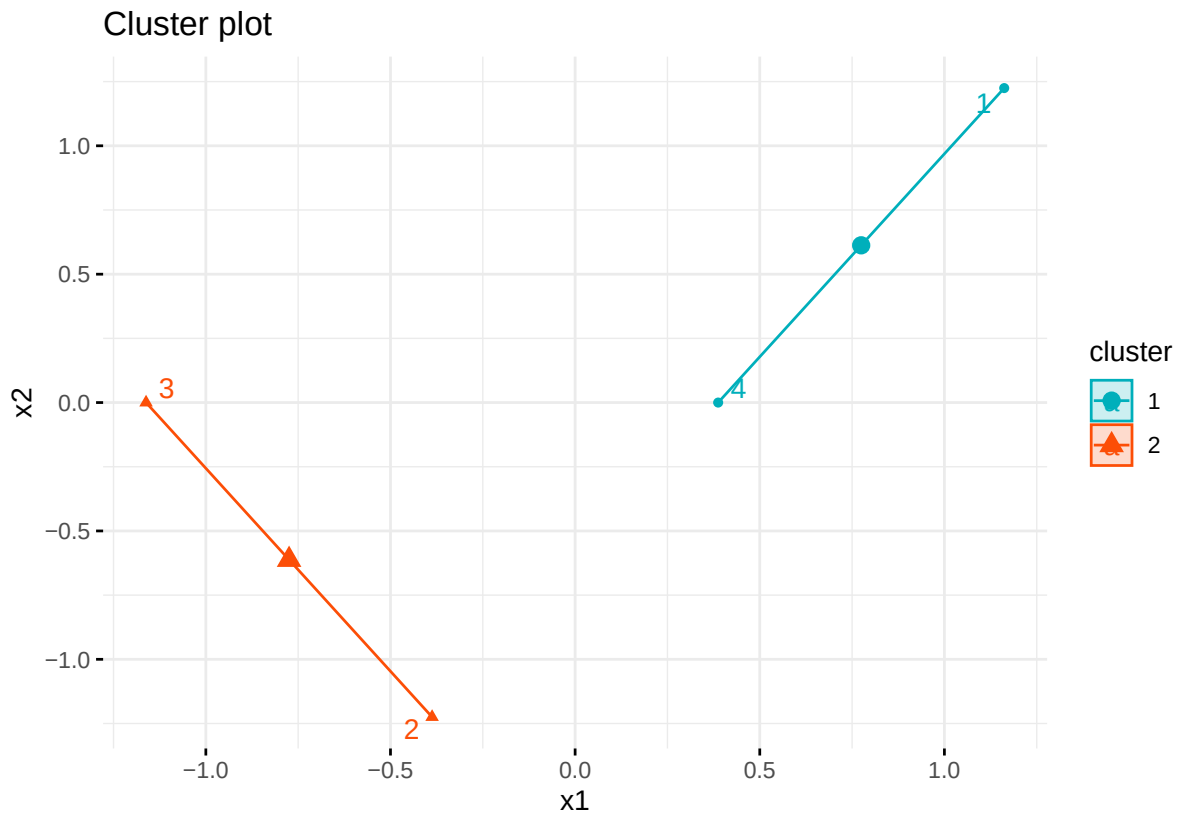
79. Johnson e Wichern - Exercício 12.7.



Analisando os dendogramas, percebemos que tanto as abordagens simples e completa agregaram os valores (1,2) e (4,5) no mesmo grupo, mas diferiram quanto a agregação do valor (3); no caso da agregação simples, o valor (3) foi caracterizado como um grupo robustamente separado dos dois demais grupos, enquanto na agregação completa, o *cluster* do valor (3) foi colocado como mais próximo do *cluster* dos valores (1,2), e esses mais distantes do *cluster* dos valores (4,5).

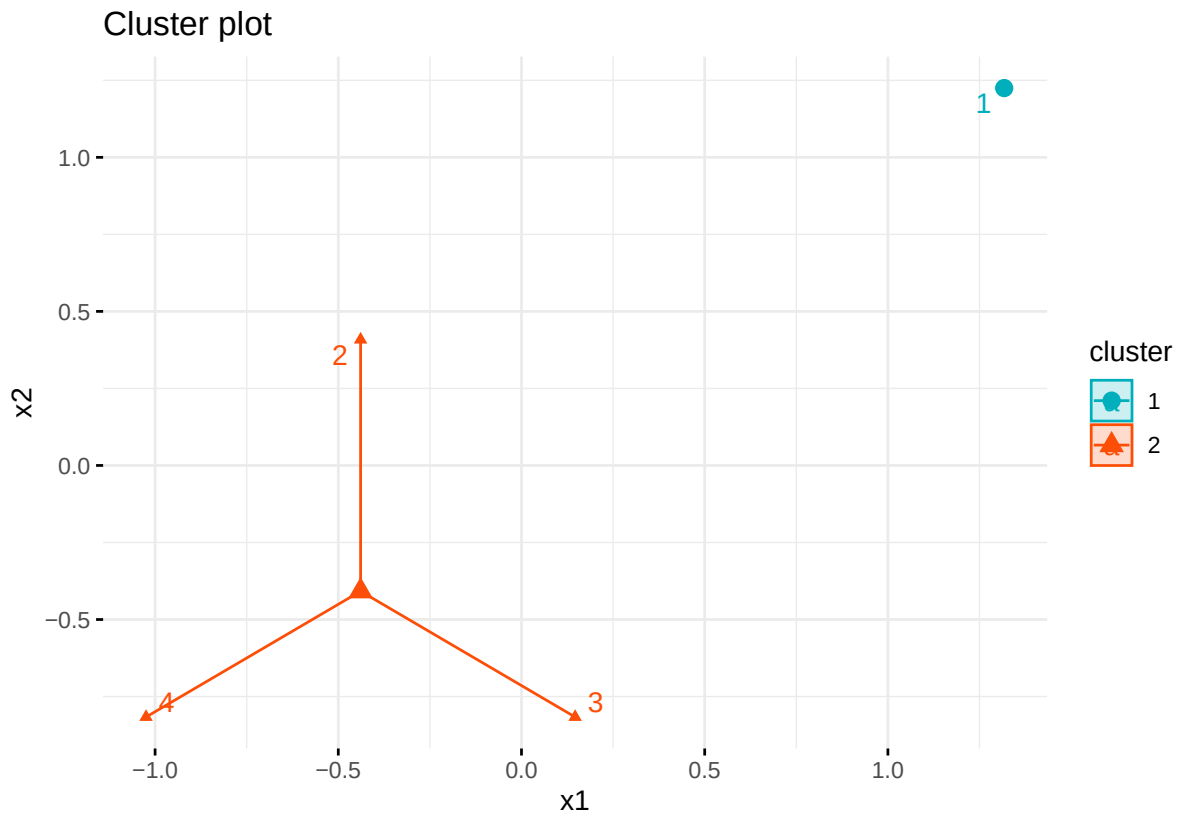
80. Johnson e Wichern - Exercício 12.11.

```
## Too few points to calculate an ellipse
## Too few points to calculate an ellipse
```



81. Johnson e Wichern - Exercício 12.12.

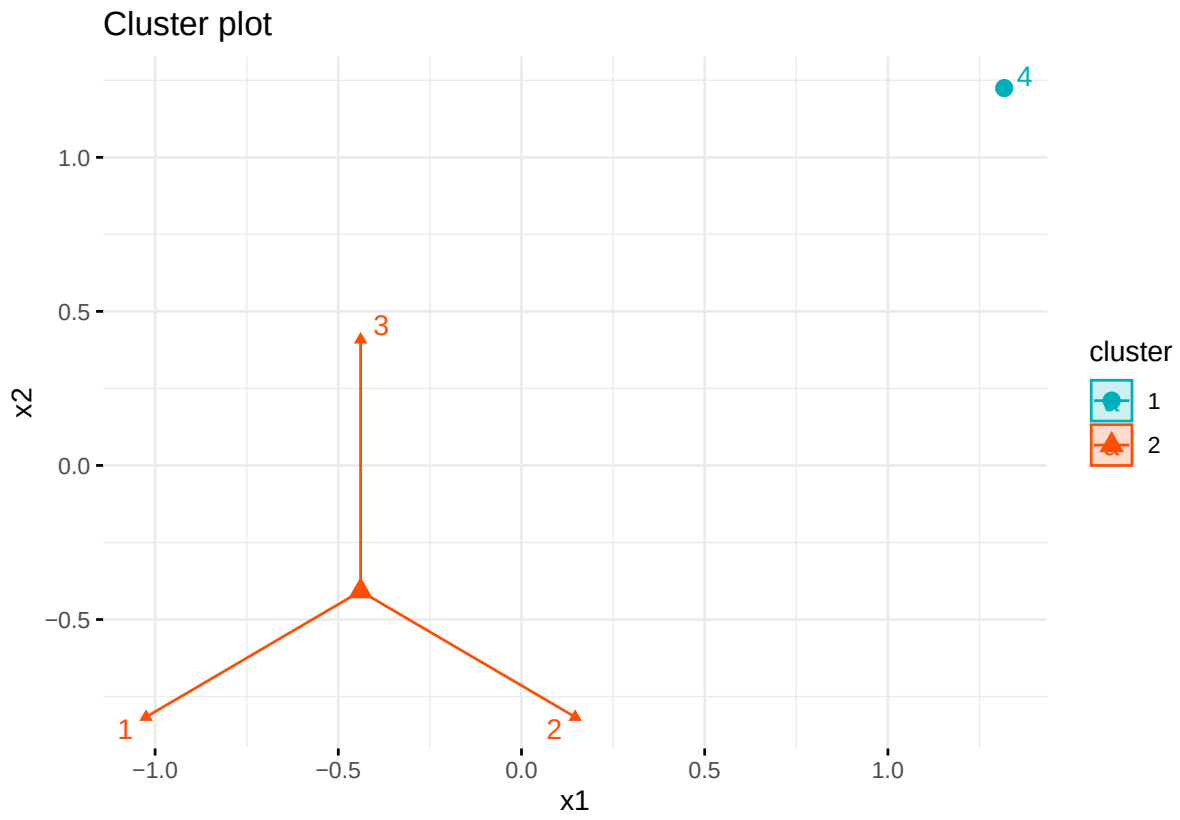
```
## Too few points to calculate an ellipse
## Too few points to calculate an ellipse
```



Conforme elucidado pelo prof. George, o algoritmo *k-means*, após decidir os centros dos grupos (neste caso, ele partiu do que eu defini manualmente inicialmente), itera os pontos afim de encontrar os centros e agrupar de tal modo que minimize a variabilidade dentro; e maximize a variabilidade entre os clusters. No caso, este ponto "ótimo" é o mesmo que o calculado no Exercício 12.11 do livro; portanto independente deu alterar os centros iniciais, o processo iterativo sempre vai retornar para este valor. Isto é verdade pelo número baixo de pontos e número alto de iterações permitidas. Conjuntos com muitos pontos e número de iterações reduzido por vezes irão produzir resultados aglomerativos diferentes.

82. Johnson e Wichern - Exercício 12.13.

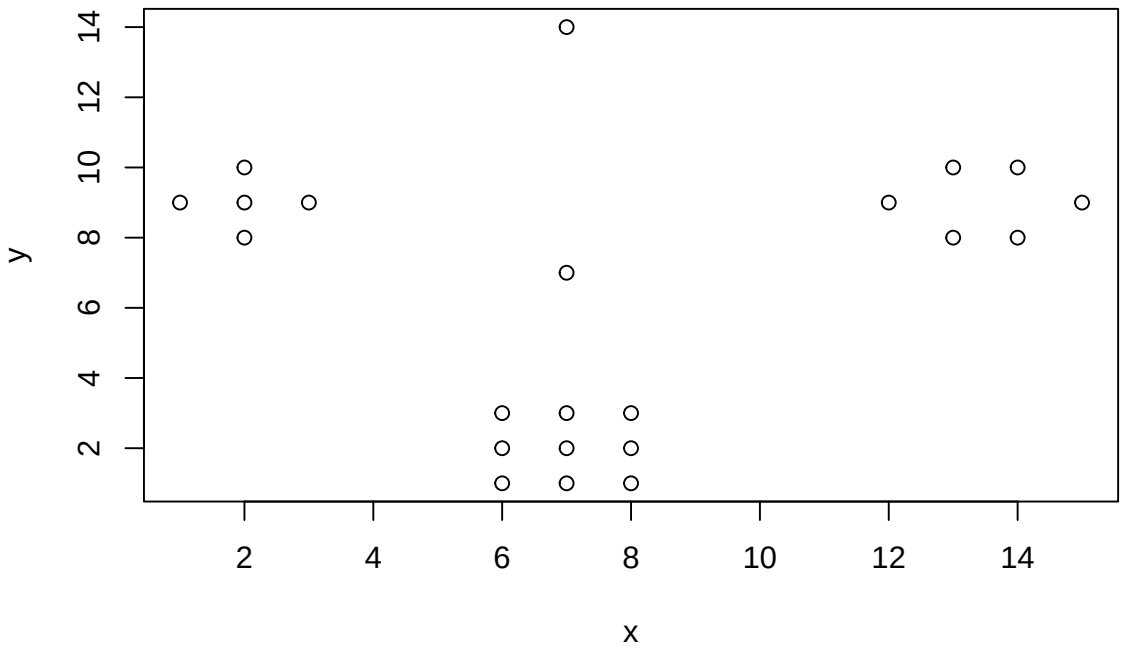
```
## Too few points to calculate an ellipse
## Too few points to calculate an ellipse
```



Os agrupamentos foram idênticos, mas inverteu os clusters: Antes o cluster 1 continha os pontos “ACD” e o cluster 2; o ponto “B”. Agora, o cluster 1 contém o ponto “B” e o cluster 2 contém os pontos “ACD”. O motivo é análogo ao descrito no item anterior.

83.

a)



Pela análise do gráfico, aparetam haver entre 3 a 5 grupos: sendo 3 grupos sólidos agrupados, e 2 *outliers* dispersos que provavelmente otimizariam formando um grupo para cada, ou ainda podem ser talvez agregados a algum dos 3 grupos mais robustos, porém aumentando assim sua dispersão.

b)

Irei apresentar os valores na forma corrida para caber melhor no documento, mas é bom observar que a forma "natural" destes valores são matrizes triangulares inferiores. Favor verificar o código para exibi-los estruturados.

Distâncias euclidianas:

1.4142136, 1, 1.4142136, 2, 7.8102497, 11, 12.0415946, 12.0415946, 13.0384048, 13.0384048, 14, 6.3245553, 7.8102497, 8.4852814, 9.2195445, 8.6023253, 9.2195445, 9.8994949, 9.4339811, 10, 10.6301458, 1, 2, 1.4142136, 6.4031242, 10.0498756, 11, 11.1803399, 12, 12.1655251, 13.0384048, 5.8309519, 8.0622577, 8.6023253, 9.2195445, 8.9442719, 9.4339811, 10, 9.8488578, 10.2956301, 10.8166538, 1, 1, 7.0710678, 10, 11.045361, 11.045361, 12.0415946, 12.0415946, 13, 5.3851648, 7.2111026, 7.8102497, 8.4852814, 8.0622577, 8.6023253, 9.2195445, 8.9442719, 9.4339811, 10, 1.4142136, 7.8102497, 10.0498756, 11.1803399, 11, 12.1655251, 12, 13.0384048, 5.0990195, 6.4031242, 7.0710678, 7.8102497, 7.2111026, 7.8102497, 8.4852814, 8.0622577, 8.6023253, 9.2195445, 6.4031242, 9, 10.0498756, 10.0498756, 11.045361, 11.045361, 12, 4.472136, 6.7082039, 7.2111026, 7.8102497, 7.6157731, 8.0622577, 8.6023253, 8.5440037, 8.9442719, 9.4339811, 7.0710678, 7.2111026, 8.4852814, 8.0622577, 9.2195445, 9.4339811, 7, 11.045361, 11, 11.045361, 12.0415946, 12, 12.0415946, 13.0384048, 13, 13.0384048, 1.4142136, 1.4142136, 2.236068, 2.236068, 3, 5.3851648, 8.4852814, 7.8102497, 7.2111026, 9.2195445, 8.6023253, 8.0622577, 10, 9.4339811, 8.9442719, 2, 1, 2.236068, 2.236068, 6.7082039, 9.8994949, 9.2195445, 8.6023253, 10.6301458, 10, 9.4339811, 11.4017543, 10.8166538, 10.2956301, 2.236068, 1, 2.236068, 6.0827625, 8.6023253, 7.8102497, 7.0710678, 9.2195445, 8.4852814, 7.8102497, 9.8994949, 9.2195445, 8.6023253, 2, 1.4142136, 7.6157731, 10.6301458, 9.8994949, 9.2195445, 11.3137085, 10.6301458, 10, 12.0415946, 11.4017543, 10.8166538, 1.4142136, 7.0710678,

9.4339811, 8.6023253, 7.8102497, 10, 9.2195445, 8.4852814, 10.6301458, 9.8994949, 9.2195445, 8.2462113, 10.8166538, 10, 9.2195445, 11.4017543, 10.6301458, 9.8994949, 12.0415946, 11.3137085, 10.6301458, 4.1231056, 4, 4.1231056, 5.0990195, 5, 5.0990195, 6.0827625, 6, 6.0827625, 1, 2, 1, 1.4142136, 2.236068, 2, 2.236068, 2.8284271, 1, 1.4142136, 1, 1.4142136, 2.236068, 2, 2.236068, 2.236068, 1.4142136, 1, 2.8284271, 2.236068, 2, 1, 2, 1, 1.4142136, 2.236068, 1, 1.4142136, 1, 1.4142136, 2.236068, 1.4142136, 1, 1, 2, 1

Distâncias 'Manhattan':

2, 1, 2, 2, 11, 11, 13, 13, 14, 14, 14, 8, 11, 12, 13, 12, 13, 14, 13, 14, 15, 1, 2, 2, 9, 11, 11, 13, 12, 14, 14, 8, 11, 12, 13, 12, 13, 14, 13, 14, 15, 1, 1, 10, 10, 12, 12, 13, 13, 13, 7, 10, 11, 12, 11, 12, 13, 12, 13, 14, 2, 11, 11, 13, 11, 14, 12, 14, 6, 9, 10, 11, 10, 11, 12, 11, 12, 13, 9, 9, 11, 11, 12, 12, 12, 6, 9, 10, 11, 10, 11, 12, 11, 12, 13, 10, 10, 12, 11, 13, 13, 7, 12, 11, 12, 13, 12, 13, 14, 13, 14, 2, 2, 3, 3, 3, 7, 12, 11, 10, 13, 12, 11, 14, 13, 12, 2, 1, 3, 3, 9, 14, 13, 12, 15, 14, 13, 16, 15, 14, 3, 1, 3, 7, 12, 11, 10, 13, 12, 11, 14, 13, 12, 2, 2, 10, 15, 14, 13, 16, 15, 14, 17, 16, 15, 2, 8, 13, 12, 11, 14, 13, 12, 15, 14, 13, 10, 15, 14, 13, 16, 15, 14, 17, 16, 15, 5, 4, 5, 6, 5, 6, 7, 6, 7, 1, 2, 1, 2, 3, 2, 3, 4, 1, 2, 1, 2, 3, 2, 3, 3, 2, 1, 4, 3, 2, 1, 2, 1, 2, 3, 1, 2, 1, 2, 3, 2, 1, 1, 2, 1

Distâncias de Mahalanobis:

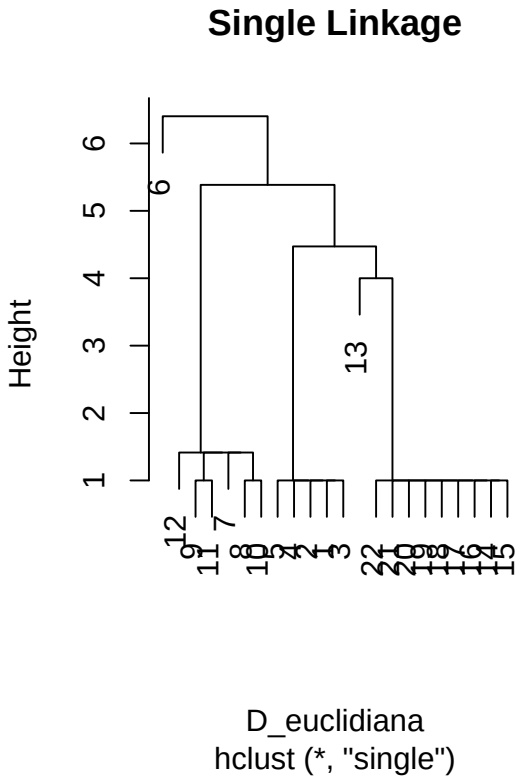
0.1089331, 0.0554805, 0.1355485, 0.2219218, 3.2670729, 6.7131345, 7.8962527, 8.2156376, 9.2699563, 9.6159566, 10.8741683, 2.4240299, 4.1896137, 4.8797448, 5.6808369, 5.1240362, 5.8274751, 6.6418749, 6.1919795, 6.9087261, 7.7364335, 0.0667603, 0.2670413, 0.1355485, 2.1890222, 5.7478824, 6.7131345, 7.2729452, 7.9891849, 8.575611, 9.6159566, 2.1874696, 4.5315583, 5.1240362, 5.8274751, 5.5861938, 6.1919795, 6.9087261, 6.7743499, 7.3934433, 8.1234976, 0.0667603, 0.0554805, 2.7233265, 5.548045, 6.6335101, 6.9262795, 7.8962527, 8.2156376, 9.3761961, 1.7871295, 3.6104434, 4.1896137, 4.8797448, 4.5315583, 5.1240362, 5.8274751, 5.5861938, 6.1919795, 6.9087261, 0.1089331, 3.3911514, 5.4817283, 6.6874063, 6.7131345, 7.9368412, 7.9891849, 9.2699563, 1.5203101, 2.8228491, 3.3887117, 4.0655352, 3.6104434, 4.1896137, 4.8797448, 4.5315583, 5.1240362, 5.8274751, 2.290541, 4.4939165, 5.4817283, 5.7478824, 6.6335101, 6.9262795, 7.9891849, 1.2611901, 3.142234, 3.6104434, 4.1896137, 4.0500412, 4.5315583, 5.1240362, 5.091369, 5.5861938, 6.1919795, 3.3887117, 3.3848461, 4.8797448, 4.1593228, 5.6808369, 5.7520648, 3.2712554, 7.9870936, 8.0779979, 8.2798631, 9.5092731, 9.6134851, 9.828658, 11.1649733, 11.282493, 11.5109736, 0.1089331, 0.1355485, 0.2620667, 0.3152975, 0.4993241, 1.5209755, 3.9215902, 3.3911514, 2.9716736, 4.709628, 4.192497, 3.7863269, 5.6311865, 5.1273632, 4.7345007, 0.2670413, 0.0554805, 0.3491371, 0.3152975, 2.3586004, 5.3377199, 4.709628, 4.192497, 6.2459707, 5.6311865, 5.1273632, 7.2877421, 6.6862656, 6.19575, 0.2959063, 0.0554805, 0.2620667, 1.9842103, 3.9217803, 3.2670729, 2.7233265, 4.5629898, 3.9215902, 3.3911514, 5.3377199, 4.709628, 4.192497, 0.2670413, 0.1355485, 3.0399231, 6.0767728, 5.3377199, 4.709628, 6.9717158, 6.2459707, 5.6311865, 8.0001795, 7.2877421, 6.6862656, 0.1089331, 2.6921485, 4.6874485, 3.9217803, 3.2670729, 5.3153503, 4.5629898, 3.9215902, 6.0767728, 5.3377199, 4.709628, 3.6048668, 6.1786718, 5.3153503, 4.5629898, 6.9267865, 6.0767728, 5.3377199, 7.8084219, 6.9717158, 6.2459707, 1.0704147, 1.068165, 1.1768763, 1.6579498, 1.6690078, 1.7910268, 2.3790055, 2.4033713, 2.538698, 0.0554805, 0.2219218, 0.0667603, 0.1355485, 0.3152975, 0.2670413, 0.3491371, 0.5421939, 0.0554805, 0.1089331, 0.0667603, 0.1355485, 0.2959063, 0.2670413, 0.3491371, 0.2620667, 0.1089331, 0.0667603, 0.4357322, 0.2959063, 0.2670413, 0.0554805, 0.2219218, 0.0667603, 0.1355485, 0.3152975, 0.0554805, 0.1089331, 0.0667603, 0.1355485, 0.2620667, 0.1089331, 0.0667603, 0.0554805, 0.2219218, 0.0554805

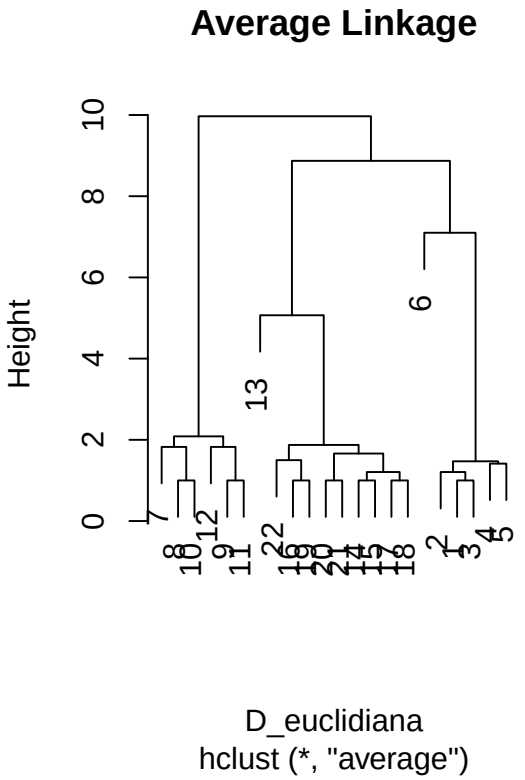
Apesar dos valores serem bem diferentes, isso se dá mais pelo método de cálculo de distância de cada uma das técnicas. A distância Euclidiana trabalha basicamente com a "distância bruta" entre um ponto e outro, literalmente medindo a distância linear. A distância Manhattan trabalha com distância absoluta entre as coordenadas dos pontos. A distância de Mahalanobis busca centralizar os dados, calculando as distâncias levando em consideração a correlação entre as dimensões.

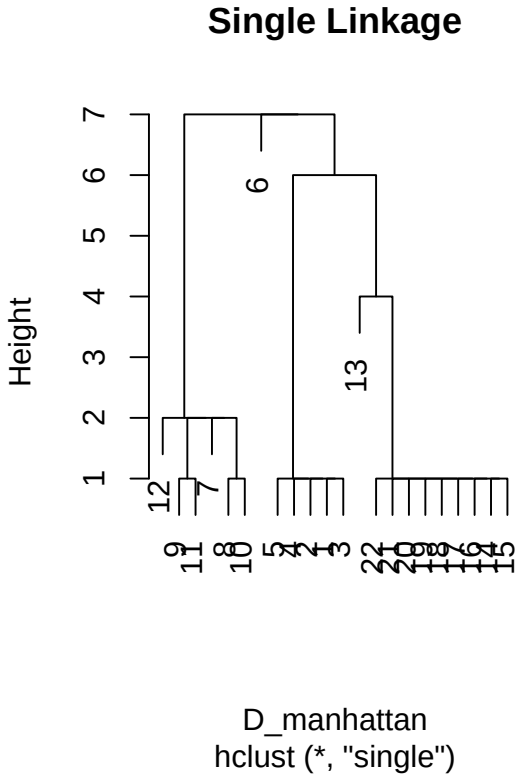
Portanto, apesar de improvável, é possível que mesmo com valores observados absolutamente distoantes, agrupar as variáveis segundo as três distâncias trabalhadas e em todos os casos, retornar os exatos mesmos clusters pros dados.

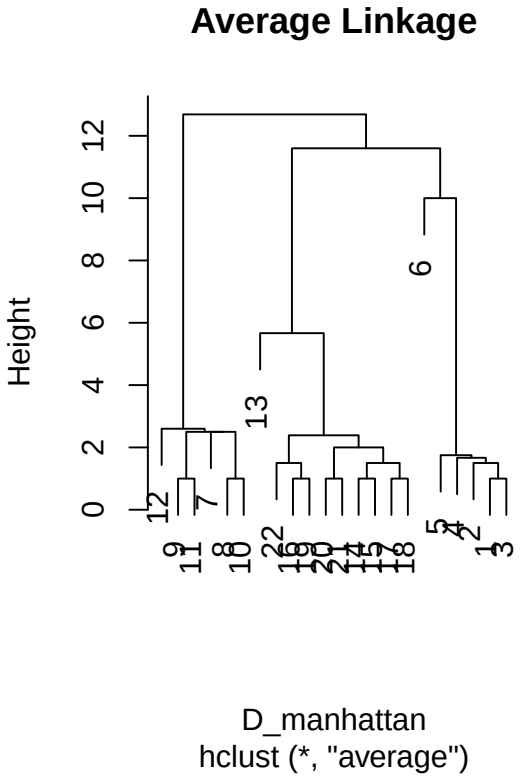
c)

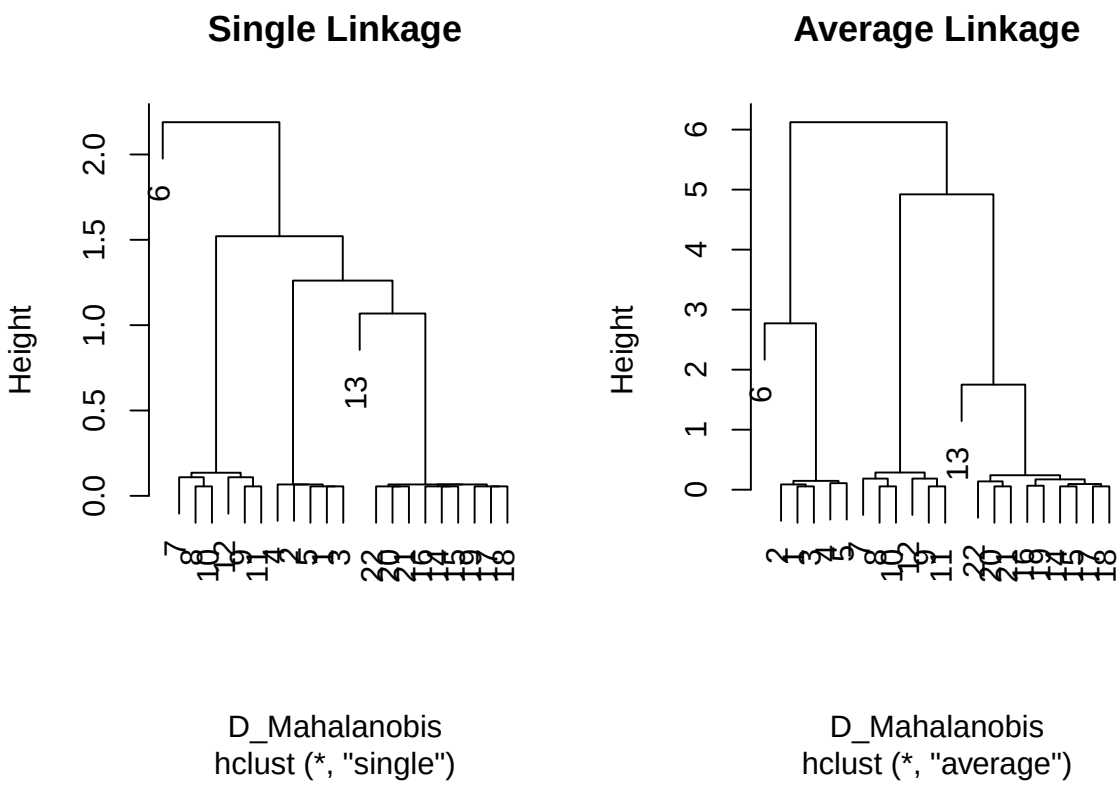
Irei apresentar corridamente três painéis, cada um composto por dois dendogramas (agregação simples e média), referentes respectivamente aos valores de distância Euclidiana, Manhattan e de Mahalanobis.







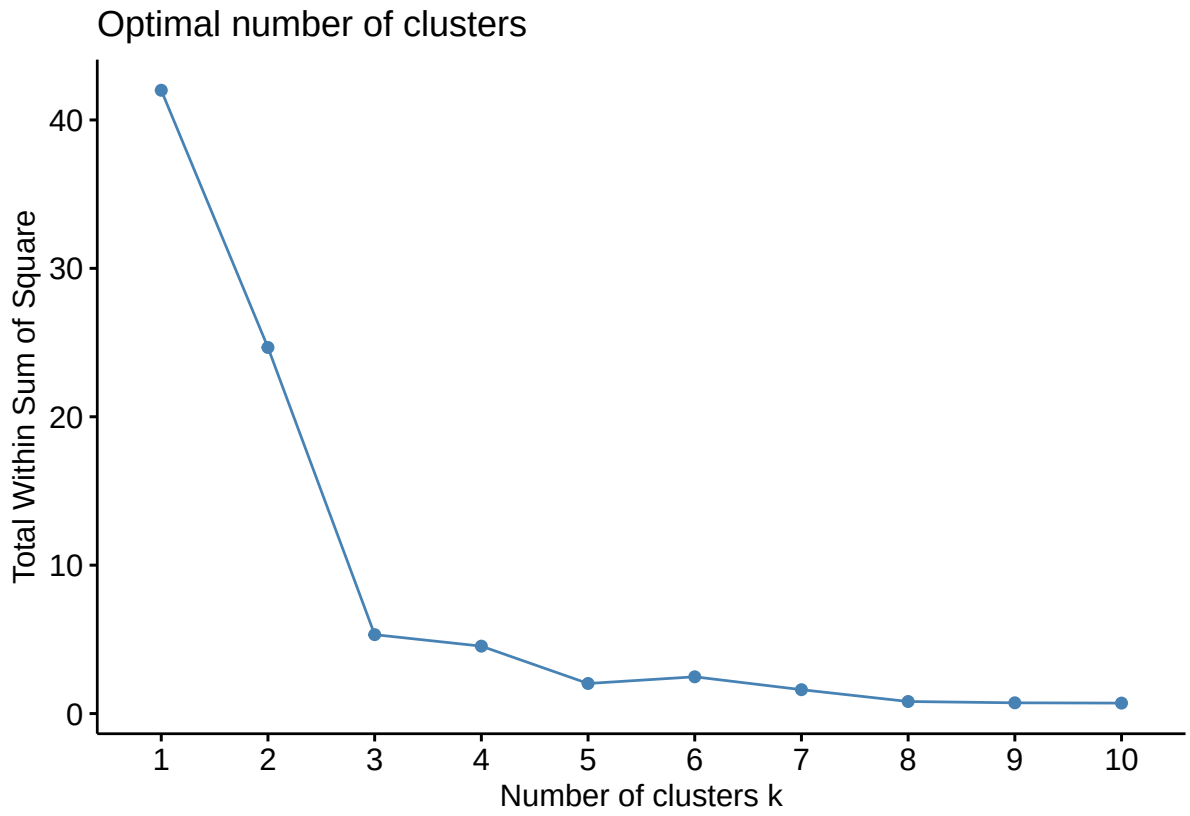




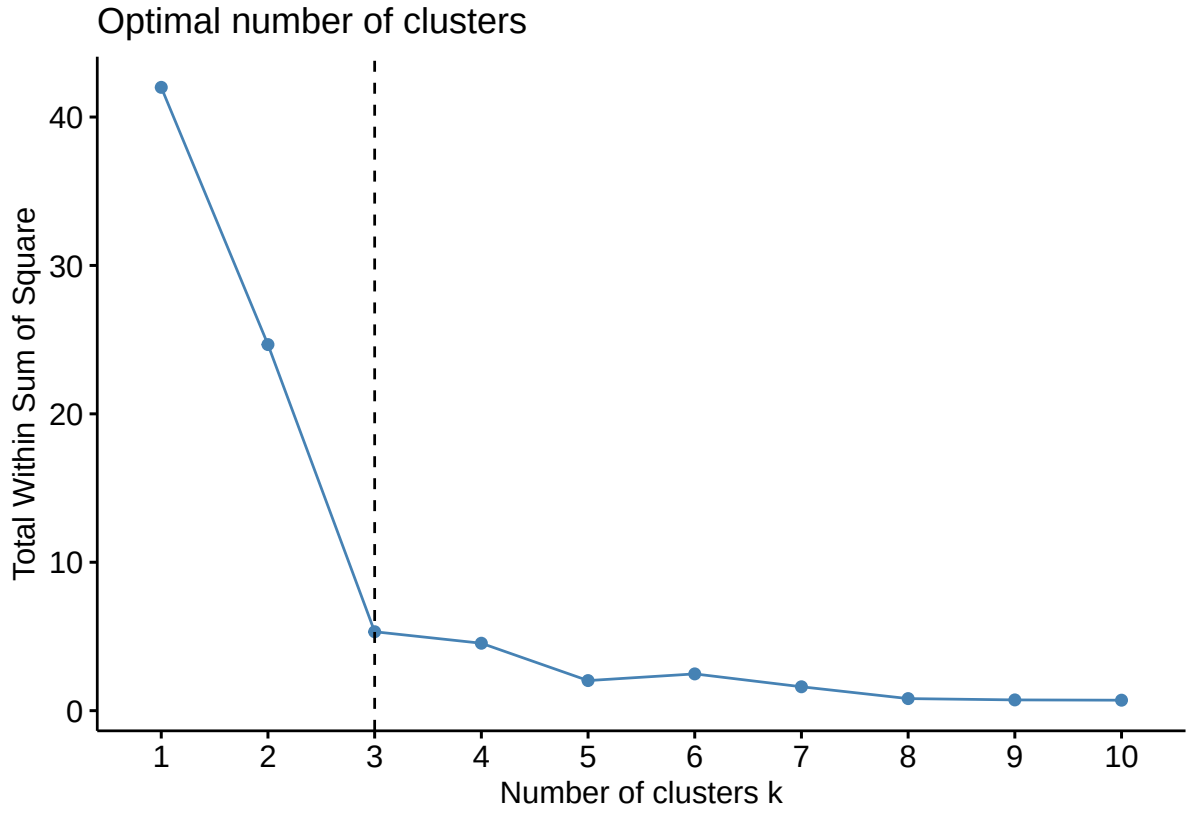
Em todos os dendogramas, foi confirmada a suspeita levantada no item (a); em que haviam 3 grupos aglomerativos bem definidos, e mais 2 grupos formados cada um por apenas um *outlier*. Cada dendograma teve seu formato específico, mas todos foram eficientes em agrupar os dados pelos seus similares.

d)

Primeiro, devemos identificar o número ideal de *clusters*, já que o *k-means* necessita que o usuário entre manualmente com o número de *clusters* que o algoritmo deve separar. Já foi visto anteriormente que o número é 3 ou 5, dependendo da abordagem que queira se fazer quanto aos *outliers*. Porém, irei também seguir a praxe deste algorítmo, que é *plotar* um gráfico que ajuda a determinar o número ideal de *clusters*.



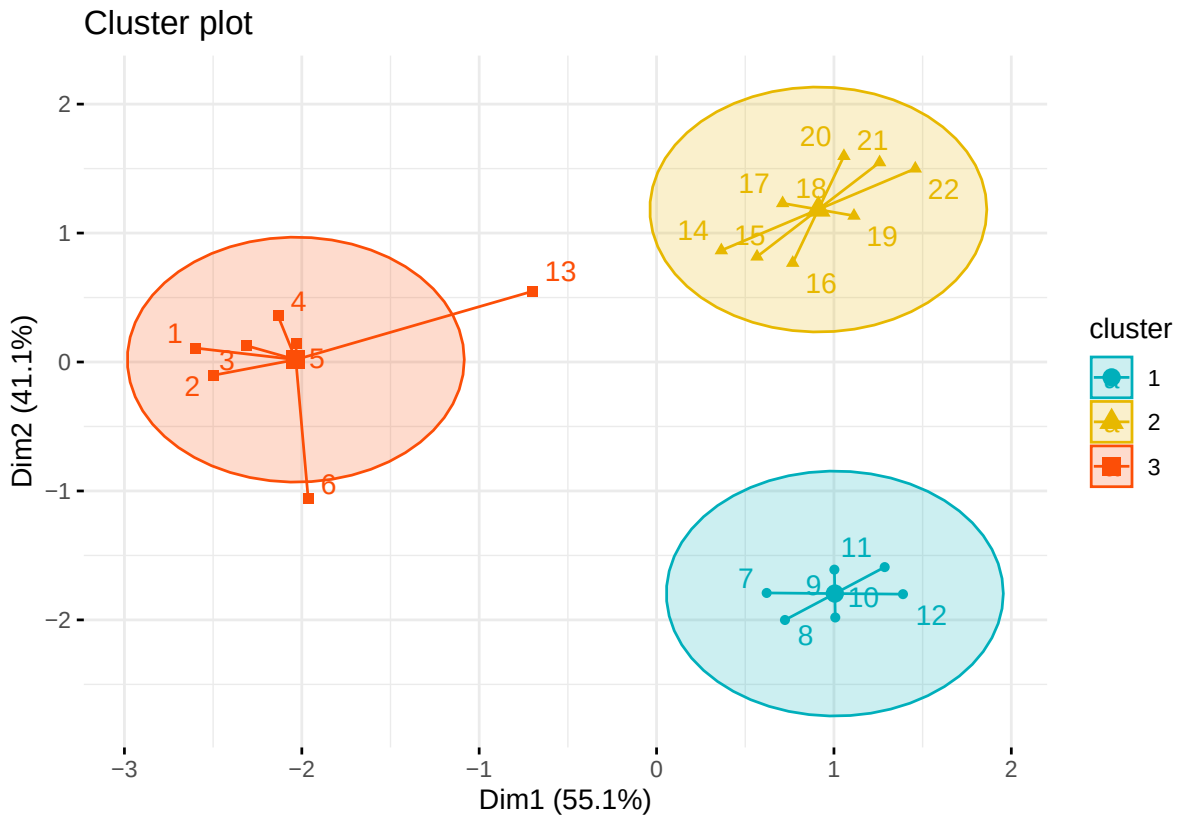
Pelo método de *elbow*, o número ideal são 3 *clusters*...



Portanto, executando o *k-means* para 3 *clusters*, iremos obter o seguinte resultado:

```
##   cluster          x          y
```

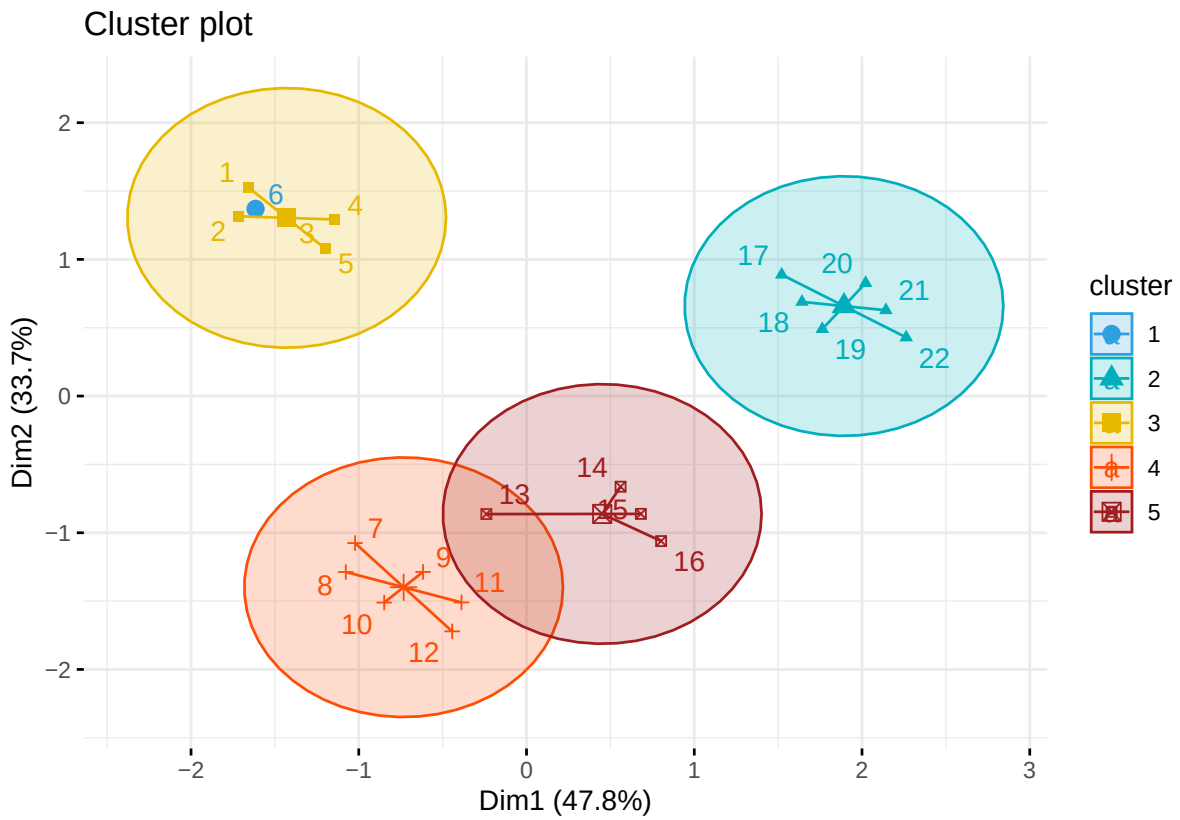
```
## 1       1  1.3728584  0.7004490
## 2       2 -0.1489924 -1.0973700
## 3       3 -0.9851742  0.8105195
```



Aqui, notamos que o *k-means* foi relativamente eficiente em classificar os dois *outliers* em um dos *clusters*, sem muita perda de generalização.

Porém, se quisermos forçar a mão e testar a aglomeração *k-means* com 5 grupos, este será o resultado:

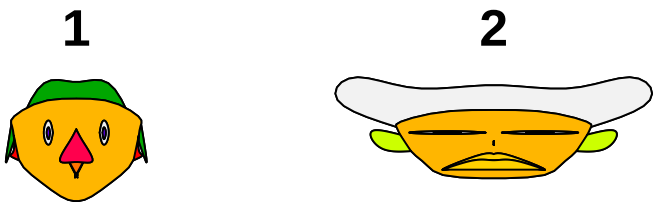
```
##   cluster ponto    x    y cluster
## 1       1   6.0  7.0 14.0    3.00
## 2       2  19.5  7.0  1.5    2.00
## 3       3   3.0  2.0  9.0    3.00
## 4       4   9.5 13.5  9.0    1.00
## 5       5  14.5  7.0  4.0    2.25
```



Em que notamos que o *k-means* não foi nada eficiente em identificar os *outliers* cada um como sendo um grupo robustamente separado dos outros três.

84.

a)

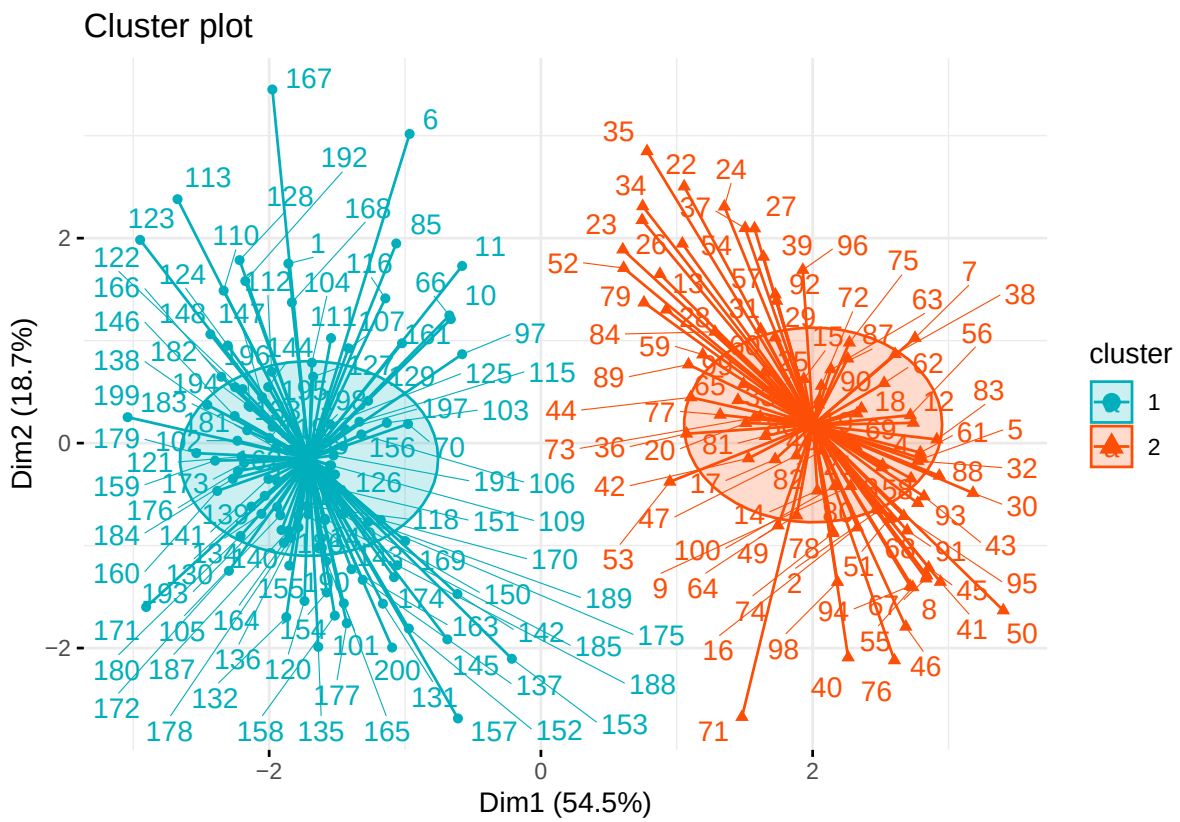


Apesar d'eu particularmente não gostar desse tipo de gráfico, por talvez trazer um ar de ridículo a um trabalho potencialmente sério, é inegável seu valor num exemplo como esse, em que conseguimos identificar de forma simples e didática a diferença entre os dois grupos de notas disponíveis, de forma muito mais visual que vetores numéricos ou gráficos potencialmente de interpretação complexa para o público leigo.

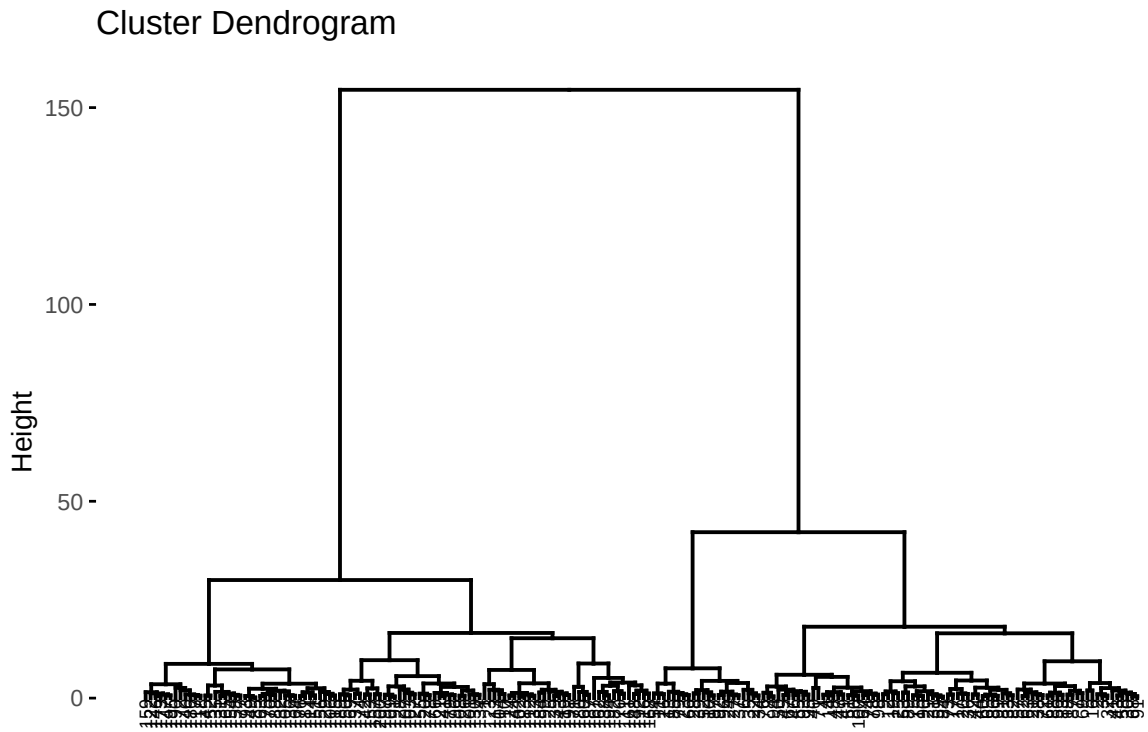
b)

Aqui, irei testar diferentes formas de agrupamento, para avaliar quais métodos performam melhor para este conjunto de dados.

***k-means*:**



Aglomerativo:

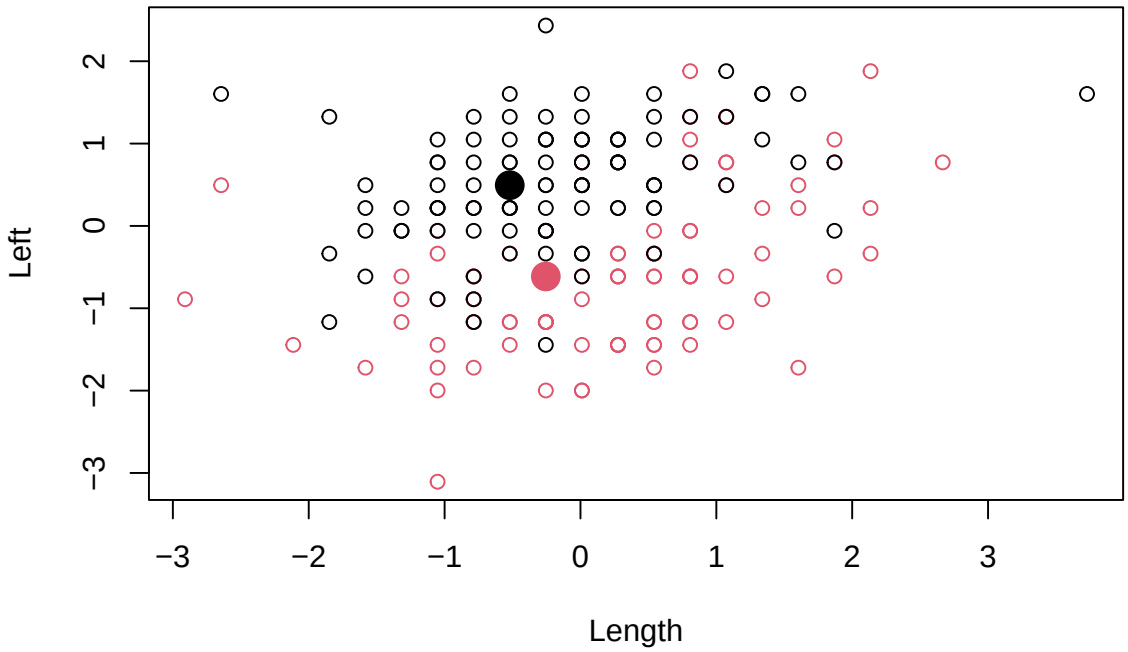


Para um conjunto relativamente grande como esse, é praticamente impossível pela figura verificar onde

está cada valor. Entretanto, ao verificar os dois grupos principais formados pelo dendograma e verificando os valores que foram agregados à eles, notamos que este foi extremamente eficiente em dividir as notas genuínas das falsificadas, com pouquíssimas observações sendo classificadas incorretamente.

Algorítmos não hierárquicos:

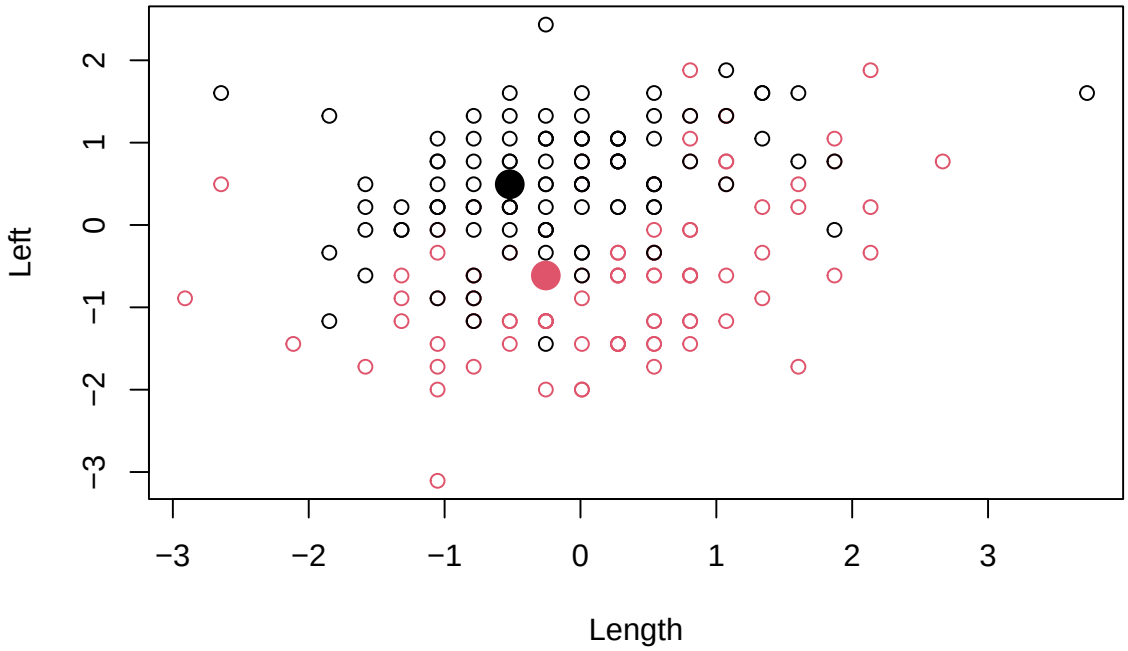
CLARA:



Var1	Freq
1	103
2	97

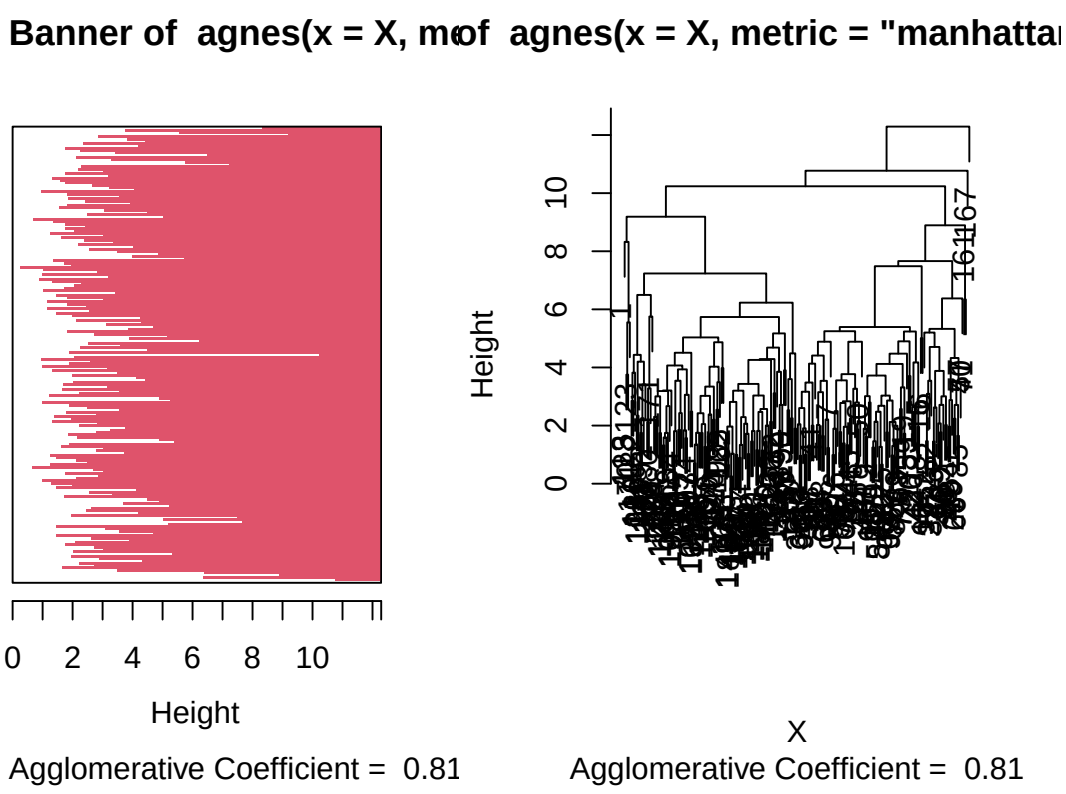
Notamos que CLARA agrupou apenas 3 valores errados, apontando 3 notas genuínas como falsificadas. Além disso, não apontou nenhuma falsificada como genuína.

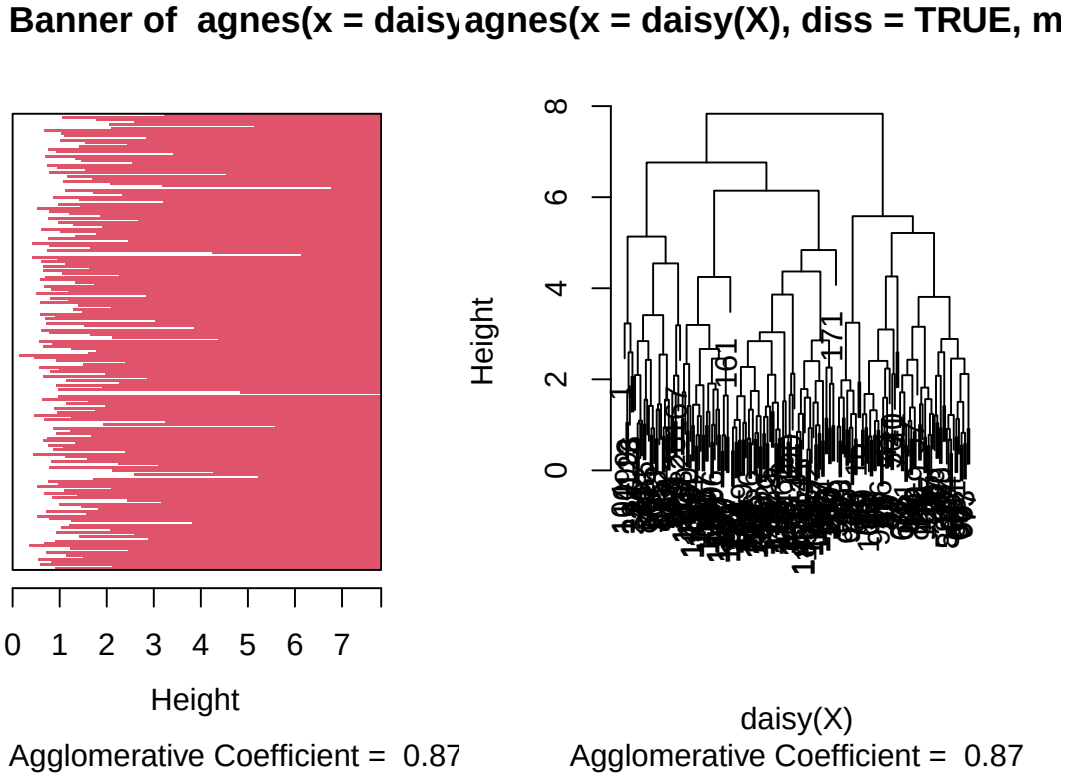
PAM:



Notamos que PAM também agrupou apenas 3 valores errados, também apontando 3 notas genuínas como falsificadas. Também não apontou nenhuma falsificada como genuína. O resultado foi idêntico ao retornado por PAM neste caso.

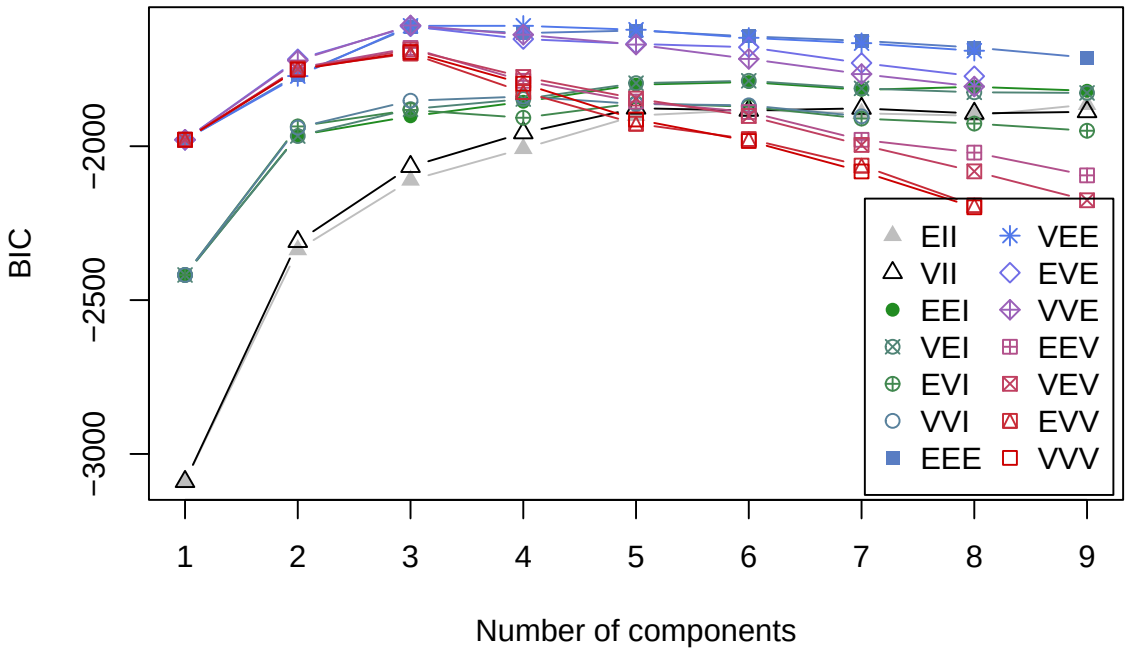
AGNES:





Aqui, testamos tanto AGNES utilizando as distâncias de Manhattan com aglomeração simples no primeiro caso, e usando distâncias euclidianas com aglomeração completa no segundo caso. Em nenhum dos dois AGNES performou tão bem quanto CLARA e PAM para este conjunto de dados.

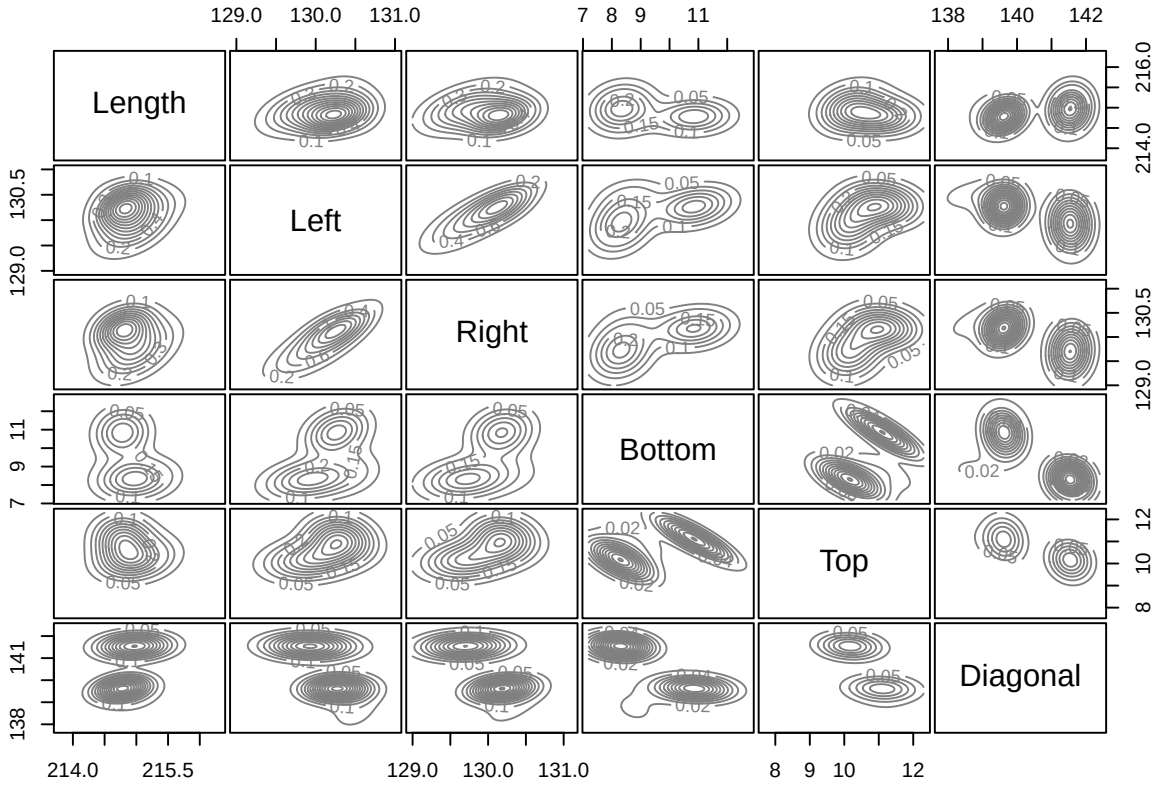
c)




```
## Best BIC values:
##              VVE,3        VEE,4        VEE,3
## BIC      -1607.574 -1608.767736 -1608.793746
## BIC diff     0.000    -1.194096    -1.220106
```

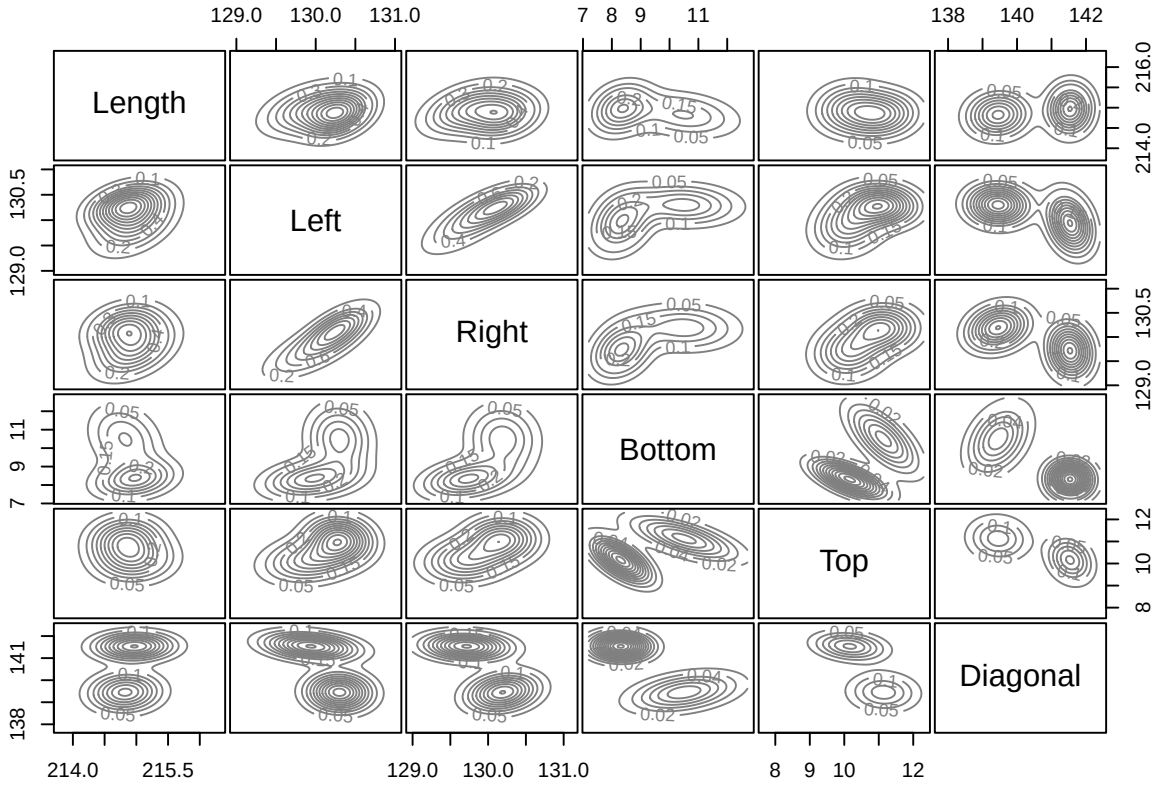
Para o método *mclust*, está indicando que o ideal seriam 3 agrupamentos, com o modelo *VVE*. Como sabemos que são apenas 2 grupos, temos que este método provavelmente não irá funcionar bem.

Seguindo a sugestão da BIC, iremos ajustar com o modelo *VVE* de 3 grupos



gsim	Freq
1	16
2	105
3	79

Percebemos que este foi o modelo que mais errou dos testados até agora. Apenas por fins didáticos, testarei o modelo mais 'complexo' *VVV*, forçando o número de *clusters* como igual à dois.



gsim	Freq
1	108
2	92

Percebemos que aqui, foi dissolvido o grupo 3 que possivelmente continham informações mais de "fronteira" entre os dois grupos mais sólidos, e estas foram diluídas entre os 2 grupos robustos existentes, com um dos grupos "ganhando" 3 itens, enquanto o outro ficando com o restante das 13 observações. Apesar do erro bruto não parecer tão grande, é um pouco decepcionante para um algoritmo tão robusto e pesado um resultado como este. Isso nos leva a suspeitar que as distribuições diferem bastante de uma normal multivariada, apesar deste não ser exatamente um pressuposto rígido deste modelo.

d)

rand_kmeans	rand_clara	rand_pam	APER_mclustVVE3	APER_mclustVVV2	rand_mclustVVE3	rand_mclustVVV2
0.8456292	0.9406018	0.9406018	0.895	0.92	-0.0049758	-0.004616

Aqui notamos que, de todos os algorítmos aqui testados, os que perfomaram melhor foram os (esquecidos e discriminados) CLARA e PAM.