**AULA 5 - SEGMENTAÇÃO, ANÁLISE DE CLUSTER E ANÁLISE DISCRIMINANTE**

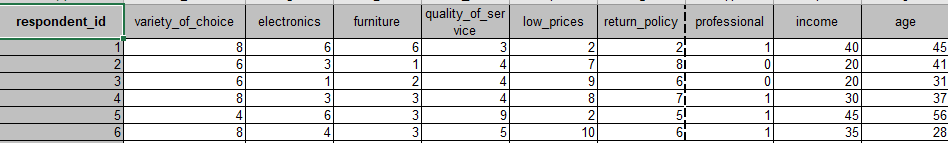
**ANÁLISE DE CLUSTER**

* Análise de Cluster funciona em um grupo de observações;
* Objetivo: Encontrar clusters de observações no espaço n-dimensional de modo que a **similaridade das observações dentro dos clusters seja a mais alta possível** e a **similaridade das observações entre os clusters seja a mais baixa possível.**
* Você pode solicitar qualquer número de clusters
* O objetivo é encontrar um número pequeno de clusters que que podem ser descritos significativamente por suas pontuações médias nas n dimensões
* **Em resumo:** Encontrar perfis de observação
* 2 tipos: **Análise de Cluster Hierárquica:** Determinar o número ideal de clusters
* **Análise de Cluster Não Hierárquica**: Atribuir observações a esses clusters.
* Os dois métodos que vamos utilizar aqui realizam os clusters baseados em distância e tentam encontrar grupos que minimizem a distância entre os membros dentro do grupo, enquanto maximizam a distância dos membros em relação a outros grupos.
* O hclust() (Análise de Cluster Hierárquica) faz isso modelando os dados em uma estrutura de árvore, enquanto o kmeans() (Análise de Cluster Não Hierárquica) usa centróides de grupo (pontos centrais).

**ANÁLISE DISCRIMINANTE**

* Tenta prever uma variável categórica com base em um número de variáveis ​​independentes contínuas ou categóricas
* No nosso caso, usaremos para tentar prever a associação do cluster de uma observação (conforme estabelecido pela análise de cluster) com base em algumas variáveis ​​de segmentação (ou seja, outras informações que temos sobre as observações que não serviram como entrada na análise de cluster).

**BANCO DE DADOS**



respondent\_id - ID do respondente

Notas para atributos da empresa

variety\_of\_choice - Variedade de Escolha

eletronics - eletrônicos

furniture - móveis, mobília  
quality\_of\_service - qualidade de serviço

low\_prices - preços baixos

return\_policy - política de troca

dados do respondente

professional - (0) non-professional & (1) professional

income - renda

age - idade

O que vamos fazer?  
  
1) Formar clusters por notas de atributo - análise de cluster

2) Utilizar dados demográficos para tentar prever em qual cluster um novo cliente estará - análise discriminante

**ANÁLISE DE CLUSTER**

Vamos começar incluindo apenas as variáveis que são relevantes para nossa análise de cluster. Aqui as nossas variáveis de interesse são diferentes atributos de uma loja, como variedade de produtos, qualidade do serviço, etc.

Pacotes Necessários

#Análise de Cluster

library(NbClust)

library(tidyverse)

library(readxl)

#Puxando dados

seg.df <- read\_excel("segmentation\_office.xlsx","SegmentationData")

\*Observação: estamos usando o read\_excel pois estamos puxando uma base em xlsx

Vamos transformar o id do respondente e os códigos dos profissionais em fatores. Isso é necessário para ao realizar a análise para confundirmos dados categóricos com números.

seg.df <- seg.df %>%

mutate(respondent\_id = factor(respondent\_id),

professional = factor(professional, labels = c("non-professional","professional")))

Separando as variáveis de interesse para a análise de cluster

cluster.data <- seg.df %>%

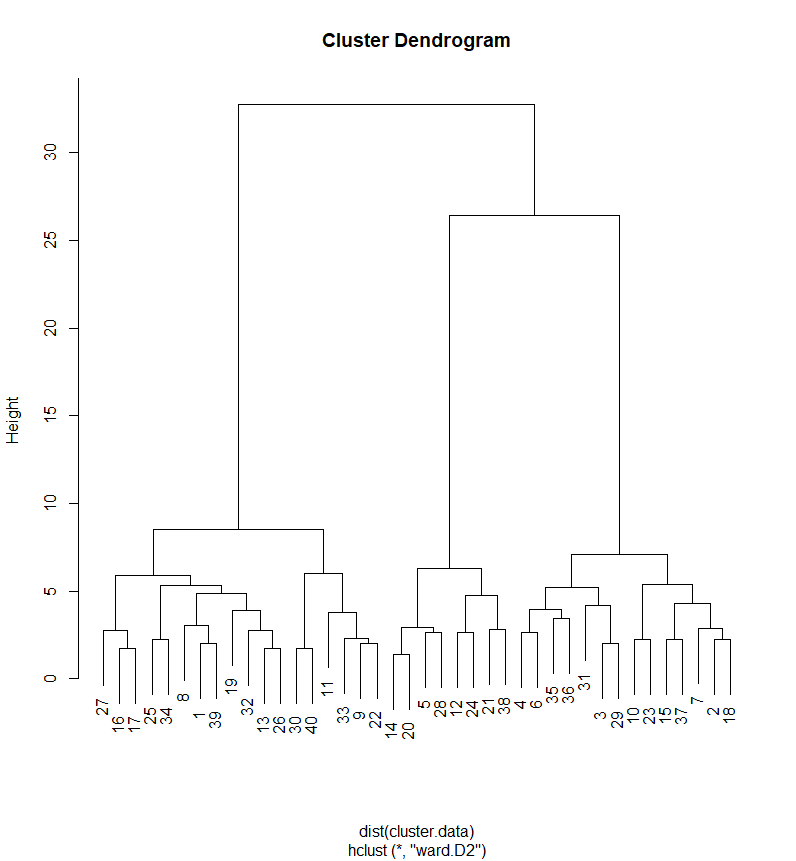
select(variety\_of\_choice, electronics, furniture, quality\_of\_service, low\_prices, return\_policy)

Posteriormente vamos fazer uma análise de cluster hierárquica. Primeiro, vamos criar uma matriz de dissimilaridade e em seguida aplicar um método de agrupamento para criar os clusters

#formando cluster hierárquico através do método de ward

hierarchical.clustering <- hclust(dist(cluster.data), method = "ward.D2")

plot(hierarchical.clustering)



**O que o código faz?**

1. **dist(cluster.data):** calcula a matriz de dissimilaridade (distância) entre as observações do conjunto cluster.data. Dist () indica o cálculo da distância. Aqui ele utiliza a distância euclidiana das observações.
2. **hclust(..., method = "ward.D2"):** O resultado da matriz de dissimilaridade é então usado como entrada para a função hclust(), que executa o agrupamento hierárquico. O método ward.D2 é um critério de aglomeração que tenta minimizar a variância dentro de cada cluster à medida que os clusters são unidos. Esse método é popular por produzir clusters mais compactos e homogêneos, o que geralmente gera melhores resultados em termos de coesão dentro dos clusters.

A partir deste dendograma, parece que podemos dividir as observações em dois, três ou seis grupos de observações. Podemos visualizar um grupo de cluster para calcular o número ótimo de clusters para a nossa análise.

Vamos realizar um teste formal, a regra de parada de Duda-Hart, para ver quantos clusters devemos reter.

#Quantos clusters?

duda <- NbClust(cluster.data, distance = "euclidean", method = "ward.D2", max.nc = 6)

\*\*\* : The Hubert index is a graphical method of determining the number of clusters.

In the plot of Hubert index, we seek a significant knee that corresponds to a

significant increase of the value of the measure i.e the significant peak in Hubert

index second differences plot.

\*\*\* : The D index is a graphical method of determining the number of clusters.

In the plot of D index, we seek a significant knee (the significant peak in Dindex

second differences plot) that corresponds to a significant increase of the value of

the measure.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Among all indices:

\* 2 proposed 2 as the best number of clusters

\* 18 proposed 3 as the best number of clusters

\* 2 proposed 4 as the best number of clusters

\* 1 proposed 6 as the best number of clusters

\*\*\*\*\* Conclusion \*\*\*\*\*

\* According to the majority rule, the best number of clusters is 3

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

> pseudot2 <- NbClust(cluster.data, distance = "euclidean", method = "ward.D2", max.nc = 6)

\*\*\* : The Hubert index is a graphical method of determining the number of clusters.

In the plot of Hubert index, we seek a significant knee that corresponds to a

significant increase of the value of the measure i.e the significant peak in Hubert

index second differences plot.

\*\*\* : The D index is a graphical method of determining the number of clusters.

In the plot of D index, we seek a significant knee (the significant peak in Dindex

second differences plot) that corresponds to a significant increase of the value of

the measure.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Among all indices:

\* 2 proposed 2 as the best number of clusters

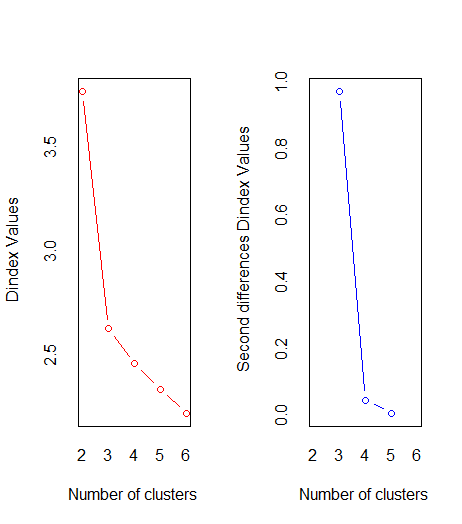
\* 18 proposed 3 as the best number of clusters

\* 2 proposed 4 as the best number of clusters

\* 1 proposed 6 as the best number of clusters

\*\*\*\*\* Conclusion \*\*\*\*\*

\* According to the majority rule, the best number of clusters is 3



Agora vamos fazer uma análise de cluster não hierárquica indicando a quantidade de cluster que vamos reter (que no caso são 3).

# há um elemento de aleatoriedade na análise de clusters

# isso significa que você não obterá sempre o mesmo resultado toda vez que fizer uma análise de clusters

# se você quiser sempre obter o mesmo resultado, precisa fixar o gerador de números aleatórios do R com o comando set.seed

set.seed(1)

# o argumento nstart deve ser incluído e configurado como 25, mas sua explicação está fora do escopo deste tutorial

kmeans.clustering <- kmeans(cluster.data, 3, nstart = 25)

# Fatorizando o indicador de cluster do data frame kmeans.clustering e adicione-o ao data frame seg.df

seg.df <- seg.df %>%

mutate(km.group = factor(kmeans.clustering$cluster, labels=c("cl1","cl2","cl3")))

#Agrupando a média das respostas por cluster

seg.df %>%

group\_by(km.group) %>% # group by cluster (km.group)

summarise(count = n(),

variety = mean(variety\_of\_choice),

electronics = mean(electronics),

furniture = mean(furniture),

service = mean(quality\_of\_service),

prices = mean(low\_prices),

return = mean(return\_policy))

km.group count variety electronics furniture service prices return

*<fct>* *<int>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>* *<dbl>*

1 cl1 14 6.93 2.79 1.43 3.5 8.29 6.29

2 cl2 18 9.11 6.06 5.78 2.39 3.67 3.17

3 cl3 8 5 4.38 1.75 8.5 2.5 4.38

Podemos verificar se há diferença nos clusters com base nos fatores demográficos por exemplo - fazemos uma ANOVA para isso

seg.aov.seg <- aov(income ~ km.group , data=seg.df)

anova(seg.aov.seg)

Analysis of Variance Table

Response: income

Df Sum Sq Mean Sq F value Pr(>F)

km.group 2 2314.3 1157.14 5.0159 0.01182 \*

Residuals 37 8535.7 230.69

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Vamos seguir isso com o teste HSD de Tukey para ver exatamente quais médias diferem umas das outras:

TukeyHSD(aov(income ~ km.group, data=seg.df),

+ "km.group")

Tukey multiple comparisons of means

95% family-wise confidence level

Fit: aov(formula = income ~ km.group, data = seg.df)

$km.group

diff lwr upr p adj

cl2-cl1 16.1904762 2.976063 29.40489 0.0133291

cl3-cl1 15.3571429 -1.078076 31.79236 0.0710698

cl3-cl2 -0.8333333 -16.590517 14.92385 0.9908529

ANÁLISE DISCRIMINANTE

Na vida real, geralmente não sabemos o que os potenciais compradores consideram importante, mas temos uma ideia de, por exemplo, sua renda, idade e status profissional.

Podemos testar o quão bem podemos prever se um indivíduo pode fazer parte de um cluster (perfil de classificações de importância) com base nas características dos respondentes (renda, idade, profissional), que também são chamadas de variáveis de segmentação.

A fórmula preditiva poderia então ser usada para prever a pertença ao cluster de novos potenciais compradores.

Vamos medir a média da renda, idade e profissão. Em profissão, transformamos em numérico e calculamos a proporção que está na segunda opção (trabalhando).

seg.df %>%

group\_by(km.group) %>% # Grupo de clientes por cluster.

summarize(income = mean(income),

age = mean(age),

professional = mean(as.numeric(professional)-1))

# A tibble: 3 × 4

km.group income age professional

*<fct>* *<dbl>* *<dbl>* *<dbl>*

1 cl1 32.1 30.9 0.5

2 cl2 48.3 44.2 0.333

3 cl3 47.5 49 0.75

Vemos que os clusters 3 e 2 são um pouco semelhantes em termos de renda e idade, mas diferem na medida em que consistem em profissionais. O cluster 3 difere dos clusters 1 e 2 por ser mais jovem e menos rico.

Vemos, por exemplo, que para as 14 observações no cluster 1, o LDA corretamente prevê que 12 estão no cluster 1, mas prevê erroneamente que 2 estão no cluster 2 e 0 estão no cluster 3.

lda.cluster3 <- lda(km.group ~ income + age + professional, data=seg.df, CV=TRUE) # CV = TRUE garante que podemos armazenar a previsão do LDA na etapa seguinte

seg.df <- seg.df %>%

+ mutate(class = factor(lda.cluster3$class, labels = c("lda1","lda2","lda3"))) # Salve a previsão do LDA como um fator. (As previsões são armazenadas em lda.cluster3$class)

ct <- table(seg.df$km.group, seg.df$class) # how many observations in each cluster were correctly predicted to be in that cluster by LDA?

ct <- table(seg.df$km.group, seg.df$class) # quantas observações em cada cluster foram corretamente previstas como estando naquele cluster pela LDA?

ct

lda1 lda2 lda3

cl1 12 2 0

cl2 3 12 3

cl3 2 3 3

prop.table(ct)

lda1 lda2 lda3

cl1 0.300 0.050 0.000

cl2 0.075 0.300 0.075

cl3 0.050 0.075 0.075

sum(diag(prop.table(ct))) # proporção corretamente predita

[1] 0.675

Digamos que queremos prever a afiliação ao cluster de novas pessoas para as quais só temos renda, idade e status profissional, mas não a afiliação ao cluster. Rodamos uma LDA para decobrir quantas dimensões discriminantes nós temos

lda.cluster3.formula <- lda(km.group ~ income + age + professional, data=seg.df, CV=FALSE) # CV = FALSE ensures that we view the formula that we can use for prediction

> lda.cluster3.formula

Call:

lda(km.group ~ income + age + professional, data = seg.df, CV = FALSE)

Prior probabilities of groups:

cl1 cl2 cl3

0.35 0.45 0.20

Group means:

income age professionalprofessional

cl1 32.14286 30.92857 0.5000000

cl2 48.33333 44.22222 0.3333333

cl3 47.50000 49.00000 0.7500000

Coefficients of linear discriminants:

LD1 LD2

income 0.02718175 -0.04456448

age 0.08017200 0.03838914

professional 0.42492950 2.10385035

Proportion of trace:

LD1 LD2

0.7776 0.2224

Vemos que o LDA reteve duas dimensões discriminantes. A primeira dimensão explica 77,76% da variância em km.group, a segunda dimensão explica 22,24% da variância em km.group. A tabela com coeficientes nos dá a fórmula para cada dimensão: **Pontuação Discriminante 1 = 0,03 × renda + 0,08 × idade + 0,42 × profissional** & **Pontuação Discriminante 2 = -0,04 × renda + 0,04 × idade + 2,1 × profissional**.

Para atribuir novas observações a um determinado cluster, precisamos fazer o seguinte passo a passo:

1. Calcular as pontuações discriminantes médias dos clusters e de cada nova observação (preenchendo os valores (médios) de renda, idade e status profissional de cada cluster ou observação nas funções discriminantes)
2. Depois calcular as distâncias geométricas entre as pontuações discriminantes das novas observações e as pontuações discriminantes médias dos clusters
3. Atribuir cada observação ao cluster que estiver mais próximo no espaço geométrico.

O R faz tudo isso sozinho! ;)

Usamos a fórmula LDA para prever a pertença ao cluster para os respondentes existentes. Para isso, usamos o comando predict e comparamos a pertença predita ao cluster (armazenada no componente class do objeto LDA) com a pertença real ao cluster.

new\_data <- new\_data %>%

mutate(prediction = predict(lda.cluster3.formula, new\_data)$class)

# A tibble: 4 × 4

income age professional prediction

*<dbl>* *<dbl>* *<chr>* *<fct>*

1 65 20 professional cl1

2 65 35 non-professional cl2

3 35 45 non-professional cl2

4 35 60 professional cl3

TAREFA

**Qual a média da renda dos clusters?**

#Média por renda

seg.df %>%

group\_by(km.group) %>%

summarise(mean\_income = mean(income, na.rm = TRUE))

**Qual a média da idade dos clusters?**

seg.df %>%

group\_by(km.group) %>%

summarise(mean\_age = mean(age, na.rm = TRUE))

**O cluster é formado por mais pessoas com ou sem trabalho?**

seg.df %>%

group\_by(km.group, professional) %>%

summarise(count = n()) %>%

arrange(km.group, desc(count))

**Avaliando os clusters com base nas informações que você possui, como você nomearia os três? E porque?**

**A avaliação a respeito dos atributos pode ser explicada com base nos dados demográficos (idade e renda)?**

seg.df %>%

group\_by(km.group) %>% # group by cluster (km.group)

summarise(count = n(),

variety = mean(variety\_of\_choice),

electronics = mean(electronics),

furniture = mean(furniture),

service = mean(quality\_of\_service),

prices = mean(low\_prices),

return = mean(return\_policy))

**A avaliação muda por pessoas empregadas x não empregadas por cluster?**

seg.df %>%

group\_by(km.group, professional) %>% # group by cluster (km.group)

summarise(count = n(),

variety = mean(variety\_of\_choice),

electronics = mean(electronics),

furniture = mean(furniture),

service = mean(quality\_of\_service),

prices = mean(low\_prices),

return = mean(return\_policy))

**#testar se diferenças são significativas**

# Filtrar os dados apenas para o cluster 3 (se quiser testar para outro grupo, basta alterar onde tem 3 pelo cluster que quiser)

cluster3\_data <- seg.df %>%

filter(km.group == "cl3")

t\_test\_result <- t.test(low\_prices ~ professional, data = cluster3\_data)

**# Ver o resultado do teste t**

t\_test\_result

**Você confiaria apenas nesses dados para clusterizar um grupo de clientes? justifique sua resposta.**