Introdução à Análise e Modelagem de Dados Multivariados com o R

Lupércio França Bessegato
Dep. de Estatística/UFJF

Roteiro Geral

1. Fundamentos gráficos em R
2. Visualização de dados multivariados
3. Técnicas de interdependência
   - Componentes principais, análise fatorial, escalonamento multidimensional; análise de agrupamentos
4. Análise de dependência
   - Discriminação e classificação; análise discriminante linear; modelos logit.
5. Referências

Técnicas de Interdependência

Roteiro do Módulo

3. Técnicas de interdependência
   - Componentes principais
   - Análise fatorial
   - Escalonamento multidimensional
   - Análise de agrupamentos

Componentes Principais

Introdução

- Objetivo:
  - √ Explicar a estrutura de variância e covariância de conjunto de variáveis através de algumas combinações lineares das mesmas
  - √ Busca-se:
    - Redução de dados
    - Interpretação

Componentes Principais Exatas

- Algebricamente:
  - √ Combinações lineares particulares das *p* variáveis aleatórias X_1, X_2, ..., X_p.
- Geometricamente:
  - √ Representam a seleção de um novo sistema de coordenadas obtidas por rotação do sistema original
  - √ Os novos eixos representam as direções com maior variabilidade
  - √ Fornecem descrição mais simples e mais parcimoniosa da estrutura de covariâncias

Componentes principais:

- √ São necessárias *p* componentes para reproduzir a variabilidade total do sistema
- √ As componentes são não correlacionadas entre si
  - Ortogonalidade entre as componentes
- √ Variabilidade das p variáveis é aproximada pela variabilidade das k principais componentes
  - Buscam-se situações em que haja quase tanta informação nas k componentes principais quanto nas p variáveis originais

- Análise de componentes principais:
  - √ Não pressupõe normalidade
    - Componentes principais derivadas de populações normais têm interpretações úteis
  - √ Com frequência, revela relações insuspeitadas
    - Pode permitir interpretações que não seriam obtidas preliminarmente
  - √ Em geral, é um passo intermediário para a aplicação de outras técnicas

Componentes Principais Exatas Extraídas da Matriz de Covariâncias

- Sejam o vetor aleatório

$$\mathbf{X}' = [X_1, X_2, \ldots, X_p].$$

com matriz de covariâncias é **Σ**, cujos autovalores são $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p \geq 0$.

- Componentes principais de **Σ**:

$$Y_1, Y_2, ..., Y_p.$$

  - √ Combinações lineares não correlacionadas do vetor aleatório, cujas variâncias são as maiores possíveis

- Definição – Componente principal:
  - √ A j-ésima componente principal da matriz **Σ** é definida como:

$$Y_j = \mathbf{e}_j'\mathbf{X} = e_{j1}X_1 + e_{j2}X_2 + \cdots + e_{jp}X_p.$$

  - √ $\mathbf{e}_j$: autovetor correspondente ao j-ésimo autovalor
- Esperança e variância de Y_j:

$$\mathrm{E}[Y_j] = \mathrm{E}[\mathbf{e}_j'\mathbf{X}] = \mathbf{e}_j'\boldsymbol{\mu} = e_{j1}\mu_1 + e_{j2}\mu_2 + \cdots + e_{jp}\mu_p.$$

$$\mathrm{Var}[Y_j] = \mathrm{Var}[\mathbf{e}_j'\mathbf{X}] = \mathbf{e}_j'\boldsymbol{\Sigma}\mathbf{e}_j = \mathbf{e}_j'\left(\sum_{i=1}^{p}\mathbf{e}_i\mathbf{e}_i'\right)\mathbf{e}_j' = \lambda_j.$$

- Covariância entre duas componentes principais:

$$\mathrm{Cov}[Y_j, Y_k] = 0,\ j \neq k$$

- Comentário:
  - √ Cada autovalor λ_j representa a variância de uma componente principal Y_j.
  - √ Autovalores estão ordenados em ordem decrescente
    - A primeira componente é a de maior variabilidade
    - A p-ésima componente é a de menor variabilidade

- Variâncias total e generalizada de $\mathbf{\Sigma}$:
  - √ Total: $\text{tr}(\mathbf{\Sigma}) = \sum_{i=1}^{p} \sigma_{ii} = \sum_{i=1}^{p} \lambda_i$
  - √ Generalizada de $\mathbf{\Sigma}$: $|\mathbf{\Sigma}| = \prod_{i=1}^{p} \lambda_i$
  - √ Em termos dessas duas medidas globais de variação, os vetores **X** e **Y** são equivalentes

- Proporção da variância total que é explicada pela j-ésima componente principal:

$$\frac{\text{Var}[Y_j]}{\text{Variância total de } \mathbf{X}} = \frac{\lambda_j}{\text{tr}(\mathbf{\Sigma})} = \frac{\lambda_j}{\sum_{i=1}^{p} \lambda_i}$$

  - √ 1ª componente tem a maior proporção de explicação
- Proporção da variância total que é explicada pelas k primeiras componentes principais

$$\frac{\sum_{j=1}^{k} \text{Var}[Y_j]}{\text{Variância total de } \mathbf{X}} = \frac{\sum_{j=1}^{k} \lambda_j}{\text{tr}(\mathbf{\Sigma})} = \frac{\sum_{j=1}^{k} \lambda_j}{\sum_{i=1}^{p} \lambda_i}$$

  - √ Busca-se analisar um conjunto menor de variáveis sem perder muita informação sobre a estrutura de variabilidade original

- Aproximação de $\mathbf{\Sigma}$:
  - √ Analisando as k primeiras componentes principais

$$\mathbf{\Sigma}_{p \times p} \approx \sum_{i=1}^{k} \lambda_i \mathbf{e}_j \mathbf{e}_j'$$

  - √ Cada parcela da soma envolve uma matriz de dimensão pxp correspondente apenas à informação da j-ésima componente principal

Correlação entre Componente Principal e Variável Aleatória

- Os coeficientes de correlação entre a componente principal Y_i de S e a variáve1 X_k é

$$\rho_{Y_i, X_k} = \frac{e_{ik}\sqrt{\lambda_i}}{\sqrt{\sigma_{kk}}}$$

  - √ A magnitude de e_{ik} mede a contribuição da k-ésima variável na i-ésima componente (a despeito das outras variáveis).
    - Não medem a importância de X_k na presença das outras variáveis.
    - Alguns estatísticos recomendam que somente os valores e_{ik} (e não as correlações) sejam consideradas na interpretação dos componentes

Estimação das Componentes Principais – Matriz de Covariâncias

- Em geral, **Σ** é estimada por **S**:

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1p} \\ S_{12} & S_{22} & \dots & S_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ S_{1p} & S_{2p} & \dots & S_{pp} \end{bmatrix}.$$

√ Autovalores de **S**: $\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_p$

√ Autovetores de **S**: $\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \dots, \hat{\mathbf{e}}_p$

- j-ésima componente principal de **S**:

$$\hat{Y}_j = \hat{\mathbf{e}}_j'\mathbf{X} = \hat{e}_{j1}X_1 + \hat{e}_{j2}X_2 + \dots + \hat{e}_{jp}X_p,\ j = 1, 2, \dots, p.$$

- Componentes principais amostrais – Propriedades
  i. Variância: $\text{Var}[\hat{Y}_j] = \hat{\lambda}_j$.
  ii. Covariância entre as componentes: $\text{Cov}(\hat{Y}_j, \hat{Y}_k) = 0,\ j \neq k$
  iii. Variância total estimada explicada pela componente:

$$\frac{\text{Var}[\hat{Y}_j]}{\text{Variância total estimada de } \mathbf{X}} = \frac{\hat{\lambda}_j}{\text{tr}(\mathbf{S})} = \frac{\hat{\lambda}_j}{\sum_{i=1}^{p} \hat{\lambda}_i}$$

  iv. Correlação estimada entre componente e variável:

$$r_{\hat{Y}_j, X_k} = \frac{\hat{e}_{jk}\sqrt{\hat{\lambda}_j}}{\sqrt{S_{kk}}}$$

- Decomposição espectral de **S**:

$$\mathbf{S} = \sum_{j=1}^{p} \hat{\lambda}_j \mathbf{e}_j \mathbf{e}_j'.$$

√ Aproximação de S pelas primeiras k componentes

$$\mathbf{S}_{p\times p} \approx \sum_{i=1}^{k} \hat{\lambda}_i \hat{\mathbf{e}}_j \hat{\mathbf{e}}_j'$$

- Scores das componentes
  √ Valor das componentes para cada elemento amostral
  √ Na prática, o uso das componentes relevantes de dá através dos scores

Componentes Principais de Variáveis Padronizadas

- Padronização do vetor aleatório **X**:

$$\mathbf{Z} = \left(\mathbf{V}^{1/2}\right)^{-1} (\mathbf{X} - \boldsymbol{\mu})$$

√ **V**½: matriz diagonal de desvios-padrão

√ Variável padronizada: $Z_i = \frac{X_i - \mu_i}{\sqrt{\sigma_{ii}}}$

√ Matriz de covariâncias de **Z**:

$$\text{Cov}(\mathbf{Z}) = \left(\mathbf{V}^{1/2}\right)^{-1} \boldsymbol{\Sigma} \left(\mathbf{V}^{1/2}\right)^{-1} = \mathbf{P}$$

√ Componentes principais de **Z**:
  – Obtidas dos autovalores e autovetores de **P**.

- Componente principal das variáveis padronizadas:
  - √ A j-ésima componente principal da matriz **Σ** :
    $$Y_j = \mathbf{e}_j'\mathbf{Z} = \mathbf{e}_j'\left(\mathbf{V}^{1/2}\right)^{-1}(\mathbf{X}-\boldsymbol{\mu}) = e_{j1}Z_1 + e_{j2}Z_2 + \cdots + e_{jp}Z_p.$$
  - √ $\mathbf{e}_j$: autovetor da matriz de correlações **P**.
- Variância total de **P**:
  $$\sum_{j=1}^{p}\mathrm{Var}[Y_j] = \sum_{j=1}^{p}\mathrm{Var}[Z_j] = p$$
  - √ Proporção de variância populacional (padronizada) devido à j-ésima componente
    $$\frac{\mathrm{Var}[Y_j]}{\text{Variância total de } \mathbf{Z}} = \frac{\lambda_j}{\mathrm{tr}(\mathbf{P})} = \frac{\lambda_j}{p},\ k = 1, 2, \ldots, p$$
  - √ Correlação entre Y_j e X_k: $\rho_{Y_j,X_k} = e_{jk}\sqrt{\lambda_j},\ i,k = 1,2,\ldots,p$

22

Comentários

- As componentes principais de **Σ** são diferentes daquelas obtidas de **P**.
  - √ Seus autovalores e autovetores são diferentes
  - √ Um conjunto de componentes principais não é simplesmente uma função do outro conjunto
- A padronização traz consequências
  - √ Variáveis deveriam ser padronizadas se elas são medidas em escalas com amplitudes muito diferentes
    - Ex. Vendas anuais e razão entre lucro/ativos

23

Padronização dos Componentes Principais Amostrais

- Frequentemente são padronizadas:
  - √ Variáveis medidas em diferentes escalas
  - √ Na mesma escala, mas com amplitudes bastante diferentes
- As componentes principais não são invariantes às mudanças na escala

25

Padronização dos Componentes Principais Amostrais

- Frequentemente são padronizadas:
  - √ Variáveis medidas em diferentes escalas
  - √ Na mesma escala, mas com amplitudes bastante diferentes
- As componentes principais não são invariantes às mudanças na escala

26

Análise de Componentes Principais – Matriz de Correlações

- As componentes principais obtidas a partir da matriz de covariâncias são influenciadas pelas variáveis de maior variância
  - √ A padronização das variáveis ameniza esse problema
- Análise de componentes principais de variáveis padronizadas é equivalente a obter as componentes principais através da matriz de correlações

Importante

- √ Um valor pequeno incomum para o último autovalor da matriz de covariâncias (ou correlação) amostral pode indicar uma dependência linear não detectada no conjunto de dados
- √ Valores grande de autovalores (e correspondentes autovetores são importantes em uma análise
- √ Autovalores próximos de zero não devem ser ignorados
  - Autovetores associados podem apontar dependências lineares no conjunto de dados

  (problemas computacionais ou de interpretação)

Gráfico dos Componentes Principais

- Podem:
  - √ revelar observações suspeitas
  - √ fornecer verificações da hipótese de normalidade

- São combinações das variáveis originais:
  - √ Se as observações provém de população normal multivariada, é razoável esperar que as componentes sejam aproximadamente normais
  - √ Se forem usadas como entrada em análises adicionais
    - Verificar se as 1ª.s componentes são aproximadamente normais
- As últimas componentes principais podem ajudar a apontar observações suspeitas

Resumo

- Procedimento auxiliar na verificação de normalidade
  - √ Construir diagrama de dispersão para os pares dos primeiros componentes principais
  - √ Construir Q-Q plots para os valores amostrais gerados por cada componente principal
- Identificação de observações suspeitas:
  - √ Construir diagramas de dispersão e Q-Q plots para as últimas componentes principais.

Exemplo

- Pesquisa de percepção de marcas:
  - √ Avaliação de características relacionadas à marca
  - √ Pergunta:
    - Quão [atributo] é a [marca]?
  - √ Variáveis:
    - Atributos: *perform*, *leader*, *latest*, *fun*, *serious*, *bargain*, *value*, *trendy*, *rebuy*
      - Níveis : 1 (menos) a 10 (mais)
    - brand:
      - Níveis: *a* a *j*
  - √ Respondentes*: 100*
  - √ Dados*: BD_multivariada.xls/brand*

Características das marcas – Perguntas:

Atributo	Exemplo de pergunta
perform	Marca tem um forte desempenho?
leader	Marca é líder no mercado?
latest	Marca tem os produtos mais recentes?
fun	Marca é divertida?
serious	Marca é séria?
bargain	Produtos da marca são uma pechincha
value	Produtos da marca possuem um bom valor?
trendy	Marca está na moda?
rebuy	Eu compraria a marca novamente?

- Fonte: Chapman, C.; Feit, E. M. *R for marketing research and analytics*, Springer, 2015

- Conjunto de dados:

```
> brand.ratings <- read.csv("rintro-chapter8.csv")
> head(brand.ratings)

 perform leader latest fun serious bargain value trendy rebuy brand
1       2      4      8   8       2       9     7      4     6     a
2       1      1      4   7       1       1     1      2     2     a
3       2      3      5   9       2       9     5      1     6     a
4       1      6     10   8       3       4     5      2     1     a
5       1      1      5   8       1       9     9      1     1     a
6       2      8      9   5       3       8     7      1     2     a
```

- √ Estrutura:

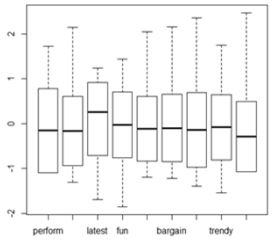
```
> str(brand.ratings)

'data.frame':   1000 obs. of  10 variables:
 $ perform: int  2 1 2 1 1 2 1 2 2 3 ...
 $ leader : int  4 1 3 6 1 8 1 1 1 1 ...
 $ latest : int  8 4 5 10 5 9 5 7 8 9 ...
 $ fun    : int  8 7 9 8 8 5 7 5 10 8 ...
 $ serious: int  2 1 2 3 1 3 1 2 1 1 ...
 $ bargain: int  9 1 9 4 9 8 5 8 7 3 ...
 $ value  : int  7 1 5 5 9 7 1 7 7 3 ...
 $ trendy : int  4 2 1 2 1 1 1 7 5 4 ...
 $ rebuy  : int  6 2 6 1 1 2 1 1 1 1 ...
 $ brand  : Factor w/ 10 levels "a","b","c","d",..: 1 1 1 1 1 1 1 1 1 1 ...
```

- Resumo dos dados:

```
> summary(brand.ratings)

 perform          leader           latest            fun        
 Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000  
 1st Qu.: 1.000   1st Qu.: 2.000   1st Qu.: 4.000   1st Qu.: 4.000  
 Median : 4.000   Median : 4.000   Median : 7.000   Median : 6.000  
 Mean   : 4.488   Mean   : 4.417   Mean   : 6.195   Mean   : 6.068  
 3rd Qu.: 7.000   3rd Qu.: 6.000   3rd Qu.: 9.000   3rd Qu.: 8.000  
 Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000  

    serious          bargain          value           trendy      
 Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.00  
 1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 3.00  
 Median : 4.000   Median : 4.000   Median : 4.000   Median : 5.00  
 Mean   : 4.323   Mean   : 4.259   Mean   : 4.337   Mean   : 5.22  
 3rd Qu.: 6.000   3rd Qu.: 6.000   3rd Qu.: 6.000   3rd Qu.: 7.00  
 Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.00  

     rebuy           brand    
 Min.   : 1.000   a      :100  
 1st Qu.: 1.000   b      :100  
 Median : 3.000   c      :100  
 Mean   : 3.727   d      :100  
 3rd Qu.: 5.000   e      :100  
 Max.   :10.000   f      :100  
                  (Other):400  
```


- Padronização dos dados:
  - √ Melhora a comparabilidade entre indivíduos

```
> brand.sc <- brand.ratings
> brand.sc[, 1:9] <- scale(brand.ratings[, 1:9])
> summary(brand.sc)

  perform            leader            latest              fun         
 Min.   :-1.0888   Min.   :-1.3100   Min.   :-1.6878   Min.   :-1.84677  
 1st Qu.:-1.0888   1st Qu.:-0.9266   1st Qu.:-0.7131   1st Qu.:-0.75358  
 Median :-0.1523   Median :-0.1599   Median : 0.2615   Median :-0.02478  
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000  
 3rd Qu.: 0.7842   3rd Qu.: 0.6069   3rd Qu.: 0.9113   3rd Qu.: 0.70402  
 Max.   : 1.7206   Max.   : 2.1404   Max.   : 1.2362   Max.   : 1.43281  
```

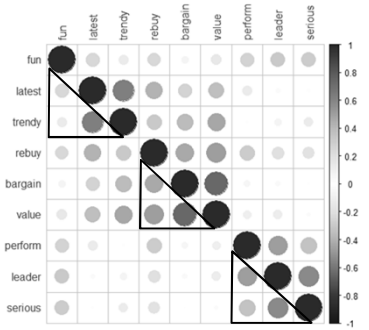


- Matriz de correlação dos dados originais:
  - √ Há grupos de variáveis mais fortemente correlacionadas?

```
> cor(brand.ratings[,1:9], use = "complete.obs")

              perform      leader      latest         fun      serious
perform  1.000000000  0.50020206 -0.122445813 -0.2563323  0.359172206
leader   0.500202058  1.00000000  0.026890447 -0.2903576  0.571215126
latest  -0.122445813  0.02689045  1.000000000  0.2451545  0.009951527
fun     -0.256332316 -0.29035764  0.245154457  1.0000000 -0.281097443
serious  0.359172206  0.57121513  0.009951527 -0.2810974  1.000000000
bargain  0.057129372  0.03309405 -0.254419016 -0.0665528 -0.002655590
value    0.101946104  0.11831017 -0.342713717 -0.1452185  0.023756556
trendy   0.008733494  0.06651244  0.627627667  0.1279736  0.121009377
rebuy    0.306658801  0.20870036 -0.397180225 -0.2371607  0.180702720
             bargain       value      trendy      rebuy
perform  0.05712937  0.10194610  0.008733494  0.3066588
leader   0.03309405  0.11831017  0.066512436  0.2087004
latest  -0.25441902 -0.34271372  0.627627667 -0.3971802
fun     -0.06655280 -0.14521849  0.127973639 -0.2371607
serious -0.00265559  0.02375656  0.121009377  0.1807027
bargain  1.00000000  0.73962672 -0.350533746  0.4673811
value    0.73962672  1.00000000 -0.434534536  0.5059617
trendy  -0.35053375 -0.43453454  1.000000000 -0.2982462
rebuy    0.46738109  0.50596166 -0.298246195  1.0000000
```


- *Correlation plot*:
  - √ Auxilia visualização das correlações

```
> library(corrplot)
> corrplot(cor(brand.sc[, 1:9]), order = "hclust")
```



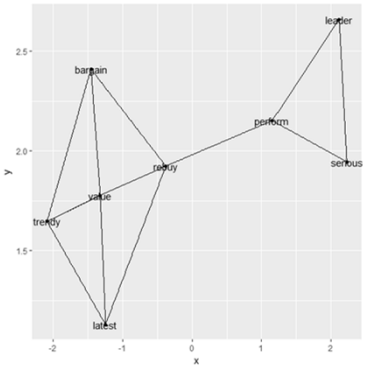
- √ Dados aparentam se agrupar em três grupos:
  - *fun*/*latest*/*trendy*
  - *rebuy*/*bargain*/*value*
  - *perform*/*leader*/*serious*

- *Correlation network plots*:
  - √ Preparação e criação dos objetos:

```
> library(tidyverse)
> library(corrr)
> library(igraph)
> library(ggraph)
> tidy_cors <- brand.sc[,1:9] %>%
+   correlate() %>%
+   stretch()
> tidy_cors
> graph_cors <- tidy_cors %>%
+   filter(abs(r) > .3) %>%
+   graph_from_data_frame(directed = FALSE)
> graph_cors
```


- *Correlation network plot*:

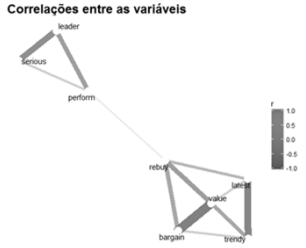
```
> ggraph(graph_cors) +
+   geom_edge_link() +
+   geom_node_point() +
+   geom_node_text(aes(label = name))
```



√ Variáveis aparentam se agrupar em dois grupos:

- *Correlation network plot*:

```
> ggraph(graph_cors) +
+   geom_edge_link(aes(edge_alpha = abs(r), edge_width = abs(r), color =
r)) +
+   guides(edge_alpha = "none", edge_width = "none") +
+   scale_edge_colour_gradientn(limits = c(-1, 1), colors =
c("firebrick2", "dodgerblue2")) +
+   geom_node_point(color = "white", size = 5) +
+   geom_node_text(aes(label = name), repel = TRUE) +
+   theme_graph() +   labs(title = "Correlações entre as variáveis")
```

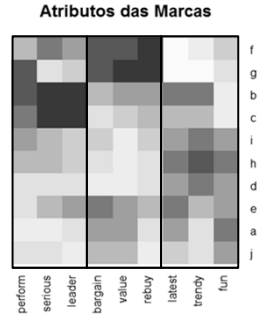


- Qual a média da marca em cada atributo?

```
> brand.mean <- aggregate(. ~brand, data = brand.sc, mean)
> rownames(brand.mean) <- brand.mean[, 1] # use brand for the row names
> brand.mean <- brand.mean[, -1] # remove brand name column
> brand.mean

      perform     leader     latest        fun     serious     bargain
a -0.88591874 -0.5279035  0.4109732  0.6566458 -0.91894067  0.21409609
b  0.93087022  1.0707584  0.7261069 -0.9722147  1.18314061  0.04161938
c  0.64992347  1.1627677 -0.1023372 -0.8446753  1.22273461 -0.60704302
d -0.67989112 -0.5930767  0.3524948  0.1865719 -0.69217505 -0.88075605
e -0.56439079  0.1928362  0.4564564  0.2958914  0.04211361  0.55155051
f -0.05868665  0.2695106 -1.2621589 -0.2179102  0.58923066  0.87400696
g  0.91838369 -0.1675336 -1.2849005 -0.5167168 -0.53379906  0.89650392
h -0.01498383 -0.2978802  0.5019396  0.7149495 -0.14145855 -0.73827529
i  0.33463879 -0.3208825  0.3557436  0.4124989 -0.14865746 -0.25459062
j -0.62994504 -0.7885965 -0.1543180  0.2849595 -0.60218870 -0.09711188
       value      trendy       rebuy
a  0.18469264 -0.52514473 -0.59616642
b  0.15133957  0.74030819  0.23697320
c -0.44067747  0.02552787 -0.13243776
d -0.93263529  0.73666135 -0.49398892
e  0.41816415  0.13857986  0.03654811
f  1.02268859 -0.81324496  1.35699580
g  1.25616009 -1.27639344  1.36092571
```

- *Heat map*:
  - √ Pontos coloridos pela intensidade

```
> library(gplots)
> library(RColorBrewer)
> heatmap.2(as.matrix(brand.mean), col = brewer.pal(9, "GnBu"), trace =
"none",
+ key = FALSE, dend = "none",
+ main = "\n\n\n\n\nAtributos das Marcas")
```



  - √ Ordenação para enfatizar similaridades e padrões
  - √ Há grupos e relações de atributos e marcas:
    - *rebuy*/*bargain*/*value*

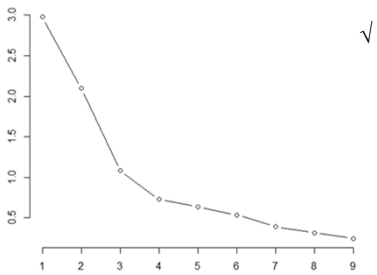
    (Marca com valor alto em um, tende a ter valor alto no outro)

- Componentes principais:
  - √ Reduzir a complexidade dos dados
    - Retenção e análise de apenas um subconjunto das componentes que expliquem grande parte da variabilidade dos dados

```
> > brand.pc <- prcomp(brand.sc[, 1:9])
> summary(brand.pc)

Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7
Standard deviation     1.726 1.4479 1.0389 0.8528 0.79846 0.73133 0.62458
Proportion of Variance 0.331 0.2329 0.1199 0.0808 0.07084 0.05943 0.04334
Cumulative Proportion  0.331 0.5640 0.6839 0.7647 0.83554 0.89497 0.93831
                           PC8     PC9
Standard deviation     0.55861 0.49310
Proportion of Variance 0.03467 0.02702
Cumulative Proportion  0.97298 1.00000
```

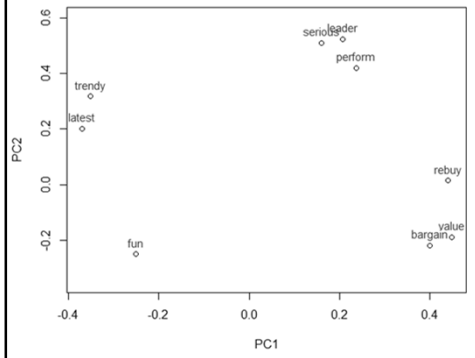

- √ *Scree plot*:

```
> plot(brand.pc, type = "l")
```



  - √ As 2 ou 3 primeiras componentes explicam a maior parte da variabilidade dos dados

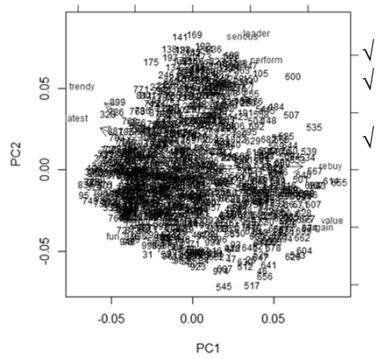
- *Plot* dos coeficientes das duas primeiras componentes:



Regiões:
- √ Liderança:
  - *serious*, *leader* e *perform*
- √ Valor:
  - *rebuy*, *value* e *bargain*
- √ Tendência:
  - *trendy* e *latest*
- √ Isolado:
  - *fun*

- *Biplot* das duas primeiras componentes:
  - √ Auxilia visualização das correlações

```
> biplot(brand.pc, cex = 0.75, expand = 1, arrow.len = 0.15)
```



- √ 4 regiões
- √ Plot muito denso
  - − todos os respondentes
- √ Solução:
  - − Executar ACP usando avaliações agregadas por marca

- Médias dos atributos por marca:

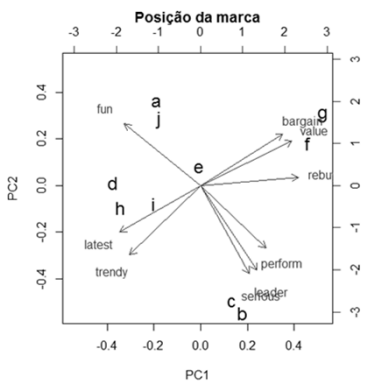
```
> brand.mean <- aggregate(. ~brand, data = brand.sc, mean)
> rownames(brand.mean) <- brand.mean[, 1] # use brand for the row names
> brand.mean <- brand.mean[, -1] # remove brand name column
> brand.mean

      perform     leader     latest        fun     serious     bargain
a -0.88591874 -0.5279035  0.4109732  0.6566458 -0.91894067  0.21409609
b  0.93087022  1.0707584  0.7261069 -0.9722147  1.18314061  0.04161938
c  0.64992347  1.1627677 -0.1023372 -0.8446753  1.22273461 -0.60704302
d -0.67989112 -0.5930767  0.3524948  0.1865719 -0.69217505 -0.88075605
e -0.56439079  0.1928362  0.4564564  0.2958914  0.04211361  0.55155051
f -0.05868665  0.2695106 -1.2621589 -0.2179102  0.58923066  0.87400696
g  0.91838369 -0.1675336 -1.2849005 -0.5167168 -0.53379906  0.89650392
h -0.01498383 -0.2978802  0.5019396  0.7149495 -0.14145855 -0.73827529
i  0.33463879 -0.3208825  0.3557436  0.4124989 -0.14865746 -0.25459062
j -0.62994504 -0.7885965 -0.1543180  0.2849595 -0.60218870 -0.09711188
       value      trendy       rebuy
a  0.18469264 -0.52514473 -0.59616642
b  0.15133957  0.74030819  0.23697320
c -0.44067747  0.02552787 -0.13243776
d -0.93263529  0.73666135 -0.49398892
e  0.41816415  0.13857986  0.03654811
f  1.02268859 -0.81324496  1.35699580
g  1.25616009 -1.27639344  1.36092571
h -0.78254646  0.86430070 -0.60402622
i -0.80339213  0.59078782 -0.20317603
j -0.07379367 -0.48138267 -0.96164748
```


- ACP dos dados agregados por média:
  - √ Realizada nova padronização:
    - – Médias agregadas têm escala um pouco diferente que os dados padronizados

```
> brand.mu.pc <- prcomp(brand.mean, scale = TRUE)
> summary(brand.mu.pc)

Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7
Standard deviation     2.1345 1.7349 0.7690 0.61498 0.50983 0.36662 0.21506
Proportion of Variance 0.5062 0.3345 0.0657 0.04202 0.02888 0.01493 0.00514
Cumulative Proportion  0.5062 0.8407 0.9064 0.94842 0.97730 0.99223 0.99737
                           PC8     PC9
Standard deviation     0.14588 0.04867
Proportion of Variance 0.00236 0.00026
Cumulative Proportion  0.99974 1.00000
```

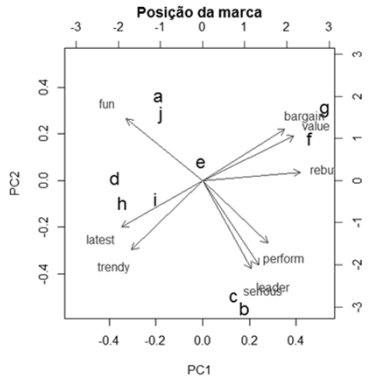
  - √ Primeiras duas componentes com 84% da variabilidade total das avaliações de média

- Mapa de percepção para médias agregadas



- √ Rotação diferente dos atributos
  - − Posição espacial é arbitrária
- √ Mesmo agrupamento global de atributos e estrutura de associações
- √ Posição das variáveis nas componentes consistente com ACP com todas as observações
  - − Pode-se prosseguir com a interpretação do gráfico

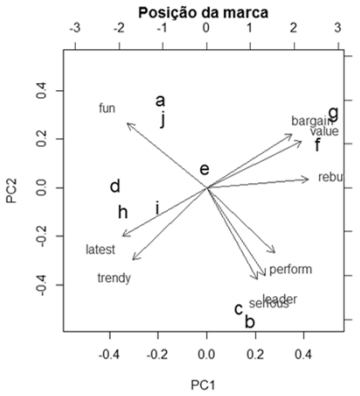
Grupos de atributos



√ *serious* e *leader* estão próximos
- Posição espacial é arbitrária

√ *fun* está distante das outras variáveis e em posição oposta aos atributos de liderança (*serious* e *leader*)

Grupos de marcas



√ Marcas **f** e **g**:
- Fortes em *value*

√ Marcas **a** e **j**:
- Relativamente fortes em *fun*

√ Marca **e**:
- Aparenta não estar bem diferenciada em qualquer das dimensões

√ Pode ser bom ou mau:
- Busca atender muitos consumidores (marca segura)
- Não tem forte percepção de diferenciação
  - Movimentar a marca em alguma direção do mapa

Diferenças entre a marca **c** e **e**:

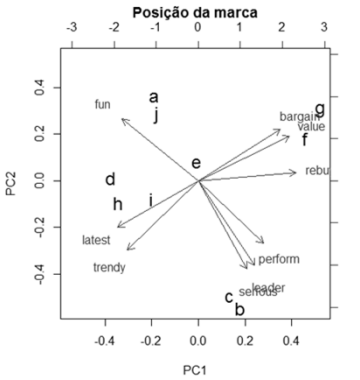
```
> brand.mean["c",] - brand.mean["e",]

   perform    leader     latest       fun  serious   bargain      value
c 1.214314 0.9699315 -0.5587936 -1.140567 1.180621 -1.158594 -0.8588416
     trendy      rebuy
c -0.113052 -0.1689859
```

√ **e** é mais forte que **c** em *value* e *fun*

√ **c** é mais forte que **e** em *perform* e *serious*
- Aspectos do produto ou da mensagem para **e** reforçar

Outra opção:

√ Não seguir outra marca, mas obter espaço diferenciado



√ Gap entre grupo **b** e **c** e **f** e **g**
- Área *value-leader* ou similar

√ Como se posicionar nessa nova área?

- Gap *value-leader*:
  - √ Assumindo que o gap reflete aproximadamente a média dessas 4 marcas

```
> colMeans(brand.mean[c("b", "c", "f", "g"), ]) - brand.mean["e", ]

   perform    leader     latest        fun   serious    bargain      value
e 1.174513 0.3910396 -0.9372789 -0.9337707 0.5732131 -0.2502787 0.07921355
      trendy     rebuy
e -0.4695304 0.6690661
```

  - √ Para marca **e** posicionar-se no gap:
    - Poderia focar *performance* e reduzir ênfase em *latest* e *fun*

- Comentário:
  - √ Mapas de percepção podem também ser usados em:
    - Pesquisa de avaliação das marcas
    - Utilizar dados objetivos:
      - Preço, medidas físicas ou combinações de ambos

Análise Fatorial

Análise Fatorial

- Objetivo:
  - √ Descrever as relações de covariância entre muitas variáveis em termos de poucas quantidades aleatórias subjacentes e não observáveis
- Motivação:
  - √ Variáveis de um grupo altamente correlacionadas entre si, mas com pequenas correlações de outros grupos
  - √ É concebível que cada grupo de variáveis represente um fator (ou construto) que seja o responsável pelas correlações observadas

- Análise fatorial:
  - √ Pode ser considerada uma extensão da Análise de Componentes Principais
    - – Ambas são tentativas de aproximar S.
    - – A aproximação baseada em Análise Fatorial é mais elaborada
  - √ Questão principal:
    - – Dados são consistentes com a estrutura prescrita?

- Análise Fatorial Exploratória:
  - √ Busca encontrar os fatores subjacentes às variáveis originais amostradas
  - √ Em geral, efetuada quando não se tem noção clara da quantidade de fatores do modelo e nem do que representam
- Análise Fatorial Confirmatória:
  - √ Tem-se em mãos um modelo fatorial pré-especificado (modelo hipotético) e deseja-se verificar se é aplicável ou consistente com os dados amostrais de que dispõe

Modelo Fatorial Ortogonal via Matriz de Correlações

- Seja o vetor aleatório

$$\mathbf{X}' = [X_1, X_2, \ldots, X_p].$$

com vetor de médias $\boldsymbol{\mu}$, matriz de covariâncias é $\boldsymbol{\Sigma}$, e matriz de correlações **P**.

- Sejam as variáveis originais padronizadas: $Z_i = \frac{X_i - \mu_i}{\sqrt{\sigma_{ii}}}$
  - √ **P** é a matriz de covariâncias do vetor aleatório **Z**, cujos componentes são as variáveis padronizadas

- Modelo Fatorial Ortogonal
  - √ Construído via a matriz de correlação populacional
  - √ Relaciona linearmente as variáveis padronizadas e os m fatores comuns (que são desconhecidos)
  - √ Fatores são variáveis independentes

- Equações do modelo:

$$\begin{aligned} Z_1 &= l_{11}F_1 + l_{12}F_2 + \cdots + l_{1m}F_m + \epsilon_1 \\ Z_2 &= l_{21}F_1 + l_{22}F_2 + \cdots + l_{2m}F_m + \epsilon_2 \\ &\vdots \quad \vdots \qquad \vdots \qquad \vdots \quad \vdots \\ Z_p &= l_{p1}F_1 + l_{p2}F_2 + \cdots + l_{pm}F_m + \epsilon_p \end{aligned}$$

√ Em notação matricial:

$$\mathbf{V}(\mathbf{X} - \boldsymbol{\mu}) = \mathbf{LF} + \boldsymbol{\epsilon}.$$

$$\mathbf{V} = \text{diagonal}[\sigma_1, \sigma_2, \ldots, \sigma_p].$$

$$\mathbf{L}_{p\times m} = \begin{bmatrix} l_{11} & l_{12} & \ldots & l_{1m} \\ l_{21} & l_{22} & \ldots & l_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ l_{p1} & l_{p2} & \ldots & l_{pm} \end{bmatrix}. \quad \mathbf{F}_{m\times 1} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{bmatrix}. \quad \boldsymbol{\epsilon}_{p\times 1} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_p \end{bmatrix}.$$

64

- Modelo fatorial:

$$\mathbf{V}(\mathbf{X} - \boldsymbol{\mu}) = \mathbf{LF} + \boldsymbol{\epsilon}.$$

√ **F**: vetor aleatório contendo m fatores
- Essas variáveis latentes precisam ser identificadas

√ **ε**: vetor dos erros aleatórios
- Erros de medida e variação de Z_i que não é explicada pelos fatores comuns

√ L: matriz de loadings fatoriais
- l_{ij}: representa o grau de relacionamento entre Z_i e F_j.

√ O modelo de análise fatorial assume que as variáveis Z_i estão relacionadas linearmente com os fatores
- Variáveis originais padronizadas são representadas por p+m variáveis não observáveis

65

Modelo de Fatores Ortogonais

- Suposições:
  i. Todos os fatores tem média zero $\mathrm{E}[\mathbf{F}] = \mathbf{0}$.
  ii. Todos os fatores são não correlacionados e tem variância um. $\mathrm{Cov}[\mathbf{F}] = \mathbf{I}_m$.
  iii. Todos os erros tem média igual a zero $\mathrm{E}[\boldsymbol{\epsilon}] = \mathbf{0}$.
  iv. Erros são não correlacionados entre si e não necessariamente tem a mesma variância

$$\mathrm{Cov}[\boldsymbol{\epsilon}] = \text{diagonal}(\psi_1, \psi_2, \ldots, \psi_p). \qquad \begin{aligned} \mathrm{Var}[\epsilon_j] &= \psi_j \\ \mathrm{Cov}(\epsilon_i, \epsilon_j) &= 0, \quad \forall i \neq j. \end{aligned}$$

66

v. Os vetores **e** e **F** são independentes

$$\mathrm{Cov}(\boldsymbol{\epsilon}_{p\times 1}, \mathbf{F}_{m\times 1}) = \mathrm{E}[\boldsymbol{\epsilon}\mathbf{F}'] = \mathbf{0}.$$

√ **F** e **ε** são duas fontes de variação distintas, relacionadas às variáveis padronizadas Z_i, não havendo qualquer relacionamento entre estas fontes de informação.

- Assumido o modelo, **P** pode ser reparametrizada

$$\mathbf{P}_{p\times p} = \mathbf{LL}' + \boldsymbol{\Psi}.$$

√ O objetivo é encontrar as matrizes $\mathbf{L}_{pxm}$ e $\boldsymbol{\psi}_{pxp}$ que possam representar a matriz $\mathbf{P}_{pxp}$.
- Há matrizes de correlação que não podem ser decompostas na forma do modelo

67

- Consequências da decomposição fatorial de **P**:
  - √ Variância de Z_i é decomposta em duas partes:

$$\text{Var}[Z_i] = h_i^2 + \psi_i$$
$$\text{onde } h_i^2 = l_{i1}^2 + l_{i2}^2 + \cdots + l_{im}^2.$$

    - h_i^2: comunalidade
      - variabilidade explicada pelos m fatores que é uma fonte comum de variação de Z_i.
    - ψ_i: variância específica
      - Parte da variabilidade de Zi associada apenas ao erro aleatório

√ Covariâncias entre variáveis e fatores

$$\text{Cov}(Z_i, Z_k) = l_{i1}l_{k1} + l_{i2}l_{k1} + \cdots + l_{im}l_{km},\ i,k = 1,2,\ldots,p, i \neq k.$$
$$\text{Cov}(Z_i, F_j) = \text{Corr}(Z_i, F_j) = l_{ij},\ i = 1,2,\ldots,p \text{ e } i = 1,2,\ldots,m.$$

√ Proporção da variância total explicada pelo fator F_j:

$$\text{Proporção explicada}_{F_j} = \frac{\sum_{i=1}^{p} l_{ij}^2}{p}.$$

Métodos de Estimação de L e ψ

- Escolhe-se o valor de m
- Métodos de estimação das matrizes **L** e **ψ**:
  - √ Método de componentes principais
    - Em geral, utilizado como um análise exploratória dos dados, em termos dos fatores subjacentes
  - √ Método de fatores principais
    - Refinamento do método das componentes principais
  - √ Método da máxima verossimilhança
    - Indicado apenas quando **Z** tem distribuição normal

Método das Componentes Principais

- Matrizes **L** e **ψ** serão estimadas por:

$$\hat{\mathbf{L}} = \left[\sqrt{\hat{\lambda}_1}\hat{\mathbf{e}}_1, \sqrt{\hat{\lambda}_2}\hat{\mathbf{e}}_2, \ldots, \sqrt{\hat{\lambda}_m}\hat{\mathbf{e}}_m\right].$$
$$\hat{\boldsymbol{\psi}} = \text{diagonal}\left(\mathbf{R}_{p\times p} - \hat{\mathbf{L}}_{p\times m}\hat{\mathbf{L}}'_{p\times m}\right).$$

√ Aproximação de R

$$\mathbf{R}_{p\times p} \approx \hat{\mathbf{L}}_{p\times m}\hat{\mathbf{L}}'_{p\times m} + \hat{\boldsymbol{\psi}}.$$

- Matriz residual:

$$\mathbf{MRes} = \mathbf{R}_{p\times p} - \left(\hat{\mathbf{L}}_{p\times m}\hat{\mathbf{L}}'_{p\times m} + \hat{\psi}\right).$$

√ Pode servir como critério de avaliação do modelo
- Seus valores deveriam ser próximos de zero
- Matriz é nula somente quando o valor de m é igual a p

√ Os elementos da diagonal da matriz R são reproduzidos exatamente pela reprodução do modelo
- O mesmo não ocorre para os outros elementos da matriz **R** (covariâncias das variáveis Z_i e Z_j)

- Método das componentes principais na estimação de **LL'** e **ψ**.

$$\text{Proporção explicada}_{F_j} = \frac{\sum_{i=1}^{p} l_{ij}^2}{p}.$$

√ Representa o quanto cada fator consegue captar da variabilidade original das variáveis Z_i.

Método da Máxima Verossimilhança

- Só pode ser utilizado quando a forma da distribuição de probabilidades é conhecida
- Suposição:

√ Vetor aleatório **X** tem distribuição normal p-variada

√ Consequência:
- Vetor das variáveis padronizadas é normal p-variado
- Fatores tem distribuição normal multivariada com vetor de médias zero e matriz de covariâncias $\mathbf{I}_m$
- Erros tem distribuição normal p-variada com vetor de médias zero e matriz de covariâncias **ψ**.

- A função de verossimilhança é expressa como:

$$L(\mathbf{0},\mathbf{P}) = \frac{1}{(2\pi)^{np/2}|\mathbf{LL}'+\boldsymbol{\psi}|^{n/2}} \exp\left\{-\frac{1}{2}\sum_{j=1}^{n}\mathbf{z}_j'\left(\mathbf{LL}'+\boldsymbol{\psi}\right)^{-1}\mathbf{z}_j\right\}.$$

√ A função de verossimilhança depende da matrizes **L** e **ψ**, através da matriz de correlação **P**.

√ As estimativas de máxima verossimilhança de $\hat{\mathbf{L}}$ e $\hat{\boldsymbol{\psi}}$ são as matrizes **L** e **ψ** que maximizam a função de verossimilhança.

√ Maximização é feita por métodos numéricos

√ Método mais sofisticado que os métodos de componentes e fatores principais
- Produz estimativas mais precisas

- Cuidados:
  - √ Está fundamentado na suposição de normalidade multivariada dos vetores **Z**, **F** e **ε**.
    - Apenas a normalidade do vetor **Z** pode ser investigada a priori a partir dos dados amostrais
    - Fatores e erros são variáveis aleatórias não observáveis

- Valor de m:
  - √ Método de máxima verossimilhança
    - Mudança de valor de m altera as estimativas dos loadings
  - √ Método de componentes principais
    - Aumento no valor de m não altera os loadings para os fatores obtidos anteriormente
  - √ Quando os dados provêm de distribuição normal multivariada
    - Usar método de componentes principais como análise exploratória dos fatores e estimação do valor provável de m
    - Posteriormente, qualidade da solução inicial poderá ser melhorada pelo uso do método de máxima verossimilhança

- Dados omissos:
  - √ São considerados apenas os elementos amostrais com observações completas

    (Análise de componentes principais e análise fatorial)
  - √ Caso haja muitas observações com dados omissos em algumas variáveis, deve-se avaliar até que ponto as análises são válidas.

Rotação dos Fatores

- A matriz de covariância Σ é reproduzida pelos loadings fatoriais obtidos por transformação ortogonal, da mesma maneira que os loadings iniciais.
  - √ Matriz de covariâncias estimada

$$\hat{\mathbf{L}}\hat{\mathbf{L}}' + \hat{\mathbf{\Psi}} = \hat{\mathbf{L}}\mathbf{T}\mathbf{T}'\hat{\mathbf{L}}' + \hat{\mathbf{\Psi}} = \hat{\mathbf{L}}^*\hat{\mathbf{L}}^{*\prime} + \hat{\mathbf{\Psi}}$$

  - √ TT' = T'T = I
  - √ $\hat{L}^*$: matriz de loadings rotacionados
  - √ A matriz de resíduos permanece a mesma ($\hat{h}_i^2$ e $\hat{\Psi}_i$)

$$\mathbf{S}_n - \hat{\mathbf{L}}\hat{\mathbf{L}}' - \hat{\mathbf{\Psi}} = \mathbf{S}_n - \hat{\mathbf{L}}^*\hat{\mathbf{L}}^{*\prime} - \hat{\mathbf{\Psi}}$$

  - √ Do ponto de vista estatístico é irrelevante obter $\hat{L}$ ou $\hat{L}^*$

- Comentários:
  - √ Com a rotação, busca-se uma estrutura mais simples
    - loadings originais podem não ter fácil interpretação
  - √ Ideal: encontrar um padrão de loadings tais que cada variável carregue-se fortemente em um único fator
    (com loadings moderados nos outros fatores)
  - √ Nem sempre é possível obter esta estrutura mais simples

Critérios de Rotação

- Ideal:
  - √ Transformação que fizesse os loadings de cada $\mathbf{Z}_i$ ter valor grande em apenas um dos fatores e valores pequenos (ou moderados) nos outros
    - Para facilitar a interpretação dos fatores
- Alguns critérios para encontrar matriz ortogonal:
  - √ Varimax
  - √ Quartimax
  - √ Orthomax

- Qualidade de ajuste
  - √ A rotação não acrescenta nenhuma melhoria em relação ao ajuste original
    - Matriz residual original não é alterada pela transformação ortogonal
    - Valores estimados de comunalidade e variâncias específicas permanecem inalterados
- Interpretação:
  - √ Novos fatores podem ser de mais fácil interpretação
- Quando a solução sem rotação já é de boa qualidade, não se recomenda rotação
  - √ Solução rotacionada pode ser pior

- Critério Varimax:
  - √ É um dos mais utilizados
  - √ Em geral, produz soluções mais simples
- Critério Quartimax
  - √ Tem tendência de gerar fatores, onde todas as variáveis têm loadings elevados
- Critério Orthomax
  - √ É uma média ponderada dos dois outros métodos

Matriz de Resíduos

- A observação da matriz de resíduos:
  - √ Muitas vezes, pode indicar quando o número de fatores está superdimensionado
  - √ Ex.:
    - Se *m* não for muito pequeno e a matriz de resíduos estiver próxima de zero, recomenda-se testar outras soluções para *m* menores que o valor já especificado

- Importante:
  - √ Análise fatorial deve ser utilizada apenas se utilizada em situações em que as variáveis originais são correlacionadas
  - √ Consequência:
    - Evitar soluções com *m* elevado tal que determinados fatores fiquem relacionados com uma única variável original
  - √ Em situações em que aparecem fatores relacionados a uma única variável Z_i é recomendável retirar a variável Z_i e reestimar o modelo de análise fatorial

Exemplo

- Pesquisa de percepção de marcas:
  - √ Avaliação de características relacionadas à marca
  - √ Pergunta:
    - Quão [atributo] é a [marca]?
  - √ Variáveis:
    - Atributos: *perform*, *leader*, *latest*, *fun*, *serious*, *bargain*, *value*, *trendy*, *rebuy*
      - Níveis : 1 (menos) a 10 (mais)
    - brand:
      - Níveis: *a* a *j*
  - √ Respondentes*: 100*
  - √ Dados*: BD_multivariada.xls/brand*

- Características das marcas – Perguntas:

Atributo	Exemplo de pergunta
perform	Marca tem um forte desempenho?
leader	Marca é líder no mercado?
latest	Marca tem os produtos mais recentes?
fun	Marca é divertida?
serious	Marca é séria?
bargain	Produtos da marca são uma pechincha
value	Produtos da marca possuem um bom valor?
trendy	Marca está na moda?
rebuy	Eu compraria a marca novamente?

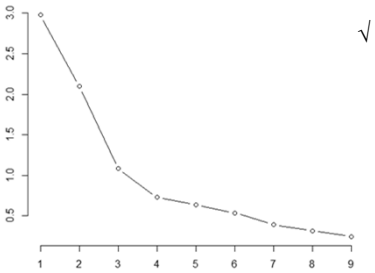
- Fonte: Chapman, C.; Feit, E. M. *R for marketing research and analytics*, Springer, 2015

Determinação da Quantidade de Fatores

- *Scree plot*
- Reter fatores associados a autovalores maiores que 1
  - √ Quantidade de variância que pode ser atribuída a uma única variável
  - √ Fator que captura variância menor que a de uma variável é considerado desprezível

√ *Scree plot*:

```
> plot(brand.pc, type = "l")
```



√ As 2 ou 3 primeiras componentes explicam a maior parte da variabilidade dos dados

- Testes para determinação de m:

```
> # scree tests
> library(nFactors)
> nScree(brand.sc[, 1:9])

 noc naf nparallel nkaiser
1   3   2        3       3
```

√ Aplicando 4 métodos, 3 sugerem que os dados têm 3 fatores

- Autovalores:

```
> # autovalores
> eigen(cor(brand.sc[, 1:9]))$values

[1] 2.9792956 2.0965517 1.0792549 0.7272110 0.6375459 0.5348432 0.3901044
[8] 0.3120464 0.2431469
```

√ Os 3 primeiros autovalores são maiores que 1.

- Escolha final:
  - √ Depende da utilidade da análise
- Verificar algumas soluções com 2 e 3 fatores

Solução com 2 fatores:

```
> # Solução com 2 fatores
> factanal(brand.sc[, 1:9], factors = 2) # perform maximum-likelihood AF
>    # default: varimax

Uniquenesses:
perform  leader  latest     fun serious bargain   value  trendy   rebuy
  0.635   0.332   0.796   0.835   0.527   0.354   0.225   0.708   0.585

Loadings:
         Factor1 Factor2
perform           0.600
leader            0.818
latest  -0.451
fun     -0.137  -0.382
serious           0.686
bargain  0.803
value    0.873   0.117
trendy  -0.534
rebuy    0.569   0.303

                Factor1 Factor2
SS loadings       2.245   1.759
Proportion Var    0.249   0.195
Cumulative Var    0.249   0.445
```

√ Fator 1: Valor
- Loadings fortes em *bargain* e *value*.

√ Fator 2: Liderança
- Cargas fatoriais fortes em *perform*, *leader* e *serious*.

Não parece ser uma má solução

Solução com 3 fatores:

```
> # Solução com 3 fatores
> factanal(brand.sc[, 1:9], factors = 3)

Uniquenesses:
perform  leader  latest     fun serious bargain   value  trendy   rebuy
  0.624   0.327   0.005   0.794   0.530   0.302   0.202   0.524   0.575

Loadings:
         Factor1 Factor2 Factor3
perform           0.607
leader            0.810   0.106
latest  -0.163            0.981
fun              -0.398   0.205
serious           0.682
bargain  0.826           -0.122
value    0.867           -0.198
trendy  -0.356            0.586
rebuy    0.499   0.296  -0.298

                Factor1 Factor2 Fac
SS loadings       1.853   1.752   1.510
Proportion Var    0.206   0.195   0
Cumulative Var    0.206   0.401   0
```

√ Fator 1: Valor
- Cargas fortes em *bargain* e *value*.

√ Fator 2: Liderança no mercado
- Cargas fatoriais fortes em *perform*, *leader* e *serious*.

√ Fator 3: Atualidade
- Cargas fatoriais fortes em *latest* e *trendy*.

Fator adicionado é interpretável

• Comparação dos modelos:

```
> # Solução com 2 fatores

Loadings:
        Factor1 Factor2
perform           0.600
leader            0.818
latest  -0.451
fun     -0.137  -0.382
serious           0.686
bargain  0.803
value    0.873   0.117
trendy  -0.534
rebuy    0.569   0.303
```

```
> # Solução com 3 fatores

Loadings:
        Factor1 Factor2 Factor3
perform           0.607
leader            0.810   0.106
latest  -0.163            0.981
fun              -0.398   0.205
serious           0.682
bargain  0.826           -0.122
value    0.867           -0.198
trendy  -0.356            0.586
rebuy    0.499   0.296  -0.298
```

√ Modelo com 3 fatores:
- Acrescenta na compreensão dos dados conceito claramente interpretável
- Está consistente com sugestões:
  - (*scree plot*, autovalores , *scree tests*, mapas de percepção)
- Aparenta ser superior ao de 2 fatores porque os fatores são melhor interpretáveis

Rotação

- Objetivo:
  - √ Obter novas cargas fatoriais com a mesma proporção de variabilidade
- Tipos:
  - √ Ortogonal:
    - Construtos são independentes
  - √ Oblíqua:
    - Construtos podem estar correlacionados
- Questão:
  - √ Você deseja permitir que os fatores estejam correlacionados ou não

Rotação Oblíqua

- Permitir correlação entre fatores relaciona-se mais com nosso conceito da estrutura latente subjacente e menos com os dados
- Os eixos dimensionais não são perpendiculares, mas assimétricos pelas correlações entre os fatores

- No exemplo:
  - √ Podemos julgar que os construtos valor e liderança estejam correlacionados
  - √ O líder pode colocar um preço especial e, portanto podemos esperar que esses dois construtos sejam correlacionados negativamente
    (ao invés de independentes)

- Rotação Oblimin (oblíqua):

```
> library(GPArotation)
> (brand.fa.ob <- factanal(brand.sc[, 1:9], factors = 3, rotation = "oblimin"))

Loadings:
        Factor1 Factor2 Factor3
perform           0.601
leader            0.816
latest                    1.009
fun              -0.381   0.229
serious           0.689
bargain  0.859
value    0.880
trendy  -0.267   0.128   0.538
rebuy    0.448   0.255  -0.226

Factor Correlations:
        Factor1 Factor2 Factor3
Factor1  1.0000 -0.0388  0.0368
Factor2 -0.3884  1.0000 -0.1091
Factor3  0.0368 -0.1091  1.0000
```

- √ Não há mudança substancial na interpretação dos fatores
  - Loadings ligeiramente diferentes

Resultados apresentam matriz de correlações

- Varimax e Oblimin – Diferenças:

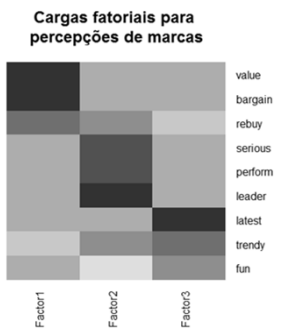
```
> # Rotação Varimax

Loadings:
        Factor1 Factor2 Factor3
perform           0.607
leader            0.810   0.106
latest  -0.163            0.981
fun              -0.398   0.205
serious           0.682
bargain  0.826           -0.122
value    0.867           -0.198
trendy  -0.356            0.586
rebuy    0.499   0.296  -0.298
```

```
> # Rotação Oblimin

Loadings:
        Factor1 Factor2 Factor3
perform           0.601
leader            0.816
latest                    1.009
fun              -0.381   0.229
serious           0.689
bargain  0.859
value    0.880
trendy  -0.267   0.128   0.538
rebuy    0.448   0.255  -0.226
```

- √ Mostra separação distinta dos atributos entre os fatores
- √ F1 é correlacionado com F2 (r = –0,39)
- √ Decisão entre as rotações:
  - Basear-se no conhecimento e domínio interpretativo, em vez da estatística

√ Mapa de calor dos loadings:

```
> library(gplots)
> library(RColorBrewer)
> heatmap.2(brand.fa.ob$loadings, col = brewer.pal(9, "Greens"),
+           trace = "none", key = FALSE, dend = "none",
+           Colv = FALSE, cexCol = 1.2,
+           main = "\n\n\n\nCargas fatoriais para \npercepções de marcas")
```



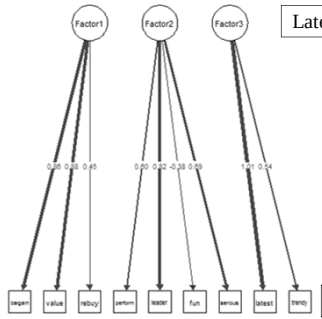
√ Separação clara das atributos nos 3 fatores

√ *Rebuy*:

- Carrega em F1 (value) e F2(leader)
- Consumidores recompram ou pelo valor da marca ou por ela ter liderança

√ *Path diagram*:

```
> library(semPlot)
> semPaths(brand.fa.ob, what = "est", residuals = FALSE,
+    cut = 0.3, posCol = c("white", "darkgreen"),
+    negCol = c("white", "red"), edge.label.cex = 0.75, nCharNodes = 7)
```



Latentes

√ Loading +: green
√ Loading –: red

√ Ao invés de usar as 9 variáveis observadas, os dados poderiam ser representados com os 3 fatores latentes subjacentes

Variáveis observáveis

• Scores dos fatores para as marcas:

√ Estimativa da variável latente para cada observação

```
> # Bartlett scores
> brand.fa.ob <- factanal(brand.sc[, 1:9], factors = 3, rotation = "oblimin",
+                         scores = "Bartlett")
> brand.scores <- data.frame(brand.fa.ob$scores)    # get the factor scores
> brand.scores$brand <- brand.sc$brand              # get the matching brands
> head(brand.scores)

     Factor1    Factor2    Factor3 brand
1  1.6521364 -0.6886749  0.5256104     a
2 -1.4005333 -1.6681901 -0.6764121     a
```

√ Útil em modelos como os de regressão porque pode-se reduzir sua complexidade (número de dimensões)

√ Permite visualizar os dados em um espaço com quantidade menor de dimensões

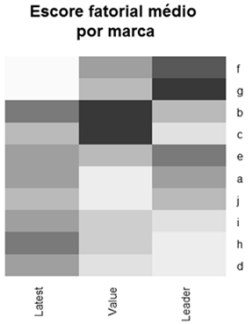
• Uso dos escores para Determinar a posição das marcas nos construtos:

```
> # Determinação da psoição da marca nos fatores
> brand.fa.mean <- aggregate(. ~ brand, data = brand.scores, mean)
> rownames(brand.fa.mean) <- brand.fa.mean[, 1]            # brand names
> brand.fa.mean <- brand.fa.mean[, -1]
> names(brand.fa.mean) <- c("Leader", "Value", "Latest")    # factor names
> brand.fa.mean

 Leader       Value      Latest
a  0.23158792 -1.06993703  0.39326652
b  0.09686823  1.51913070  0.72391174
c -0.58937138  1.45069457 -0.07690784
...
```

√ Média de cada marca por construto

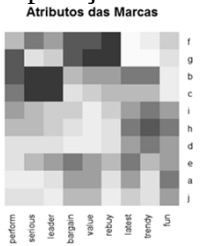
√ Mapa de calor das médias das marcas:

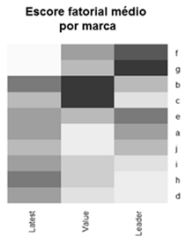
```
> library(gplots)
> library(RColorBrewer)
> heatmap.2(brand.fa.ob$loadings, col = brewer.pal(9, "Greens"),
+           trace = "none", key = FALSE, dend = "none",
+           Colv = FALSE, cexCol = 1.2,
+           main = "\n\n\n\nCargas fatoriais para \npercepções de marcas")
```



√ Média de cada marca por construto

- Comparação:





√ Mapa com scores fatoriais é mais simples que a matriz completa das percepções

√ As similaridades entre as marcas são evidenciadas novamente

– **f**-**g**, **b**-**c**, ...

Usos da Análise Fatorial

- Examinar a estrutura subjacente e as relações das variáveis
- Reduzir a complexidade dos dados em construtos mais simples e melhor interpretáveis

Análise de Agrupamentos

Statistical Learning

- Aprender a partir dos dados por meio de ajuste de modelos estatísticos
- Algumas metodologias de statistical learning:
  - √ Análise de agrupamentos (Clustering)
  - √ Classificação

- Aprendizado supervisionado:
  - √ Modelo é construído com observações cujo resultado (variável dependente) é conhecido
  - √ Objetivo: predizer a resposta a partir de variáveis independentes
  - √ Classificação
- Aprendizado não supervisionado:
  - √ Os grupos dos resultados não são conhecidos, mas tenta-se descobri-los a partir da estrutura dos dados
  - √ Clustering

Classificação e Agrupamento

- Classificar:
  - √ Número de grupos é conhecido e o objetivo é alocar novas observações a um desses grupos
- Agrupar:
  - √ Não há suposições sobre o número de grupos ou sobre a estrutura dos grupos
    - – Técnica mais primitiva

Exemplo Motivador

- Encontrar, avaliar e predizer segmentos de consumidores
- Segmentação de mercado;
  - √ Encontrar grupos de consumidores que diferem com relação ao interesse no produto, participação no mercado ou resposta aos esforços de marketing

Análise de Agrupamentos

- Procurar por uma estrutura subjacente de agrupamento dos dados
  - √ É uma importante técnica exploratória
- Objetivo básico:
  - √ Descobrir agrupamentos naturais dos itens (ou variáveis)
- Em geral, somos capazes de agrupar visualmente objetos em gráficos

- São necessários:
  - √ Medidas de similaridade (ou distância)
  - √ Desenvolvimento de escala quantitativa para medir associação (similaridade) entre os dados
  - √ Algoritmos para ordenar objetos em grupos
- Importante:
  - √ Não há método ou algoritmo que sirva para qualquer situação

- Objetivo da aplicação:
  - √ Encontrar uma solução dentre muitas outras, que represente diferenças reais nos dados e que informe e influencie nas decisões
- Importante:
  - √ Métodos estatísticos são apenas uma parte da solução.

Exemplo

- Segmentação de mercado para produto
  - √ Venda por subscrição
  - √ Amostra com 300 consumidores
- Objetivo:
  - √ Descobrir os segmentos de mercado a partir dos dados (aprendizado não supervisionados) e classificar novos membros a partir dos casos conhecidos (aprendizado supervisionado)

√ Variáveis:

- age: idade, em anos.
- gender: sexo. (“Female”, “Male”)
- income: renda anual, em US$.
- kids: quantidade de filhos em casa, em unidade.
- ownHome: proprietário residência. (“ownNo”, “ownYes”)
- subscribe: Subscrição produto. (“subNo”, “subYes”)
- Segment: segmento de mercado. (“Moving up”, “Suburb mix”, “Travelers”, “Urban hip”).

√ Dados*: rintro-chapter5.csv*

√ Fonte: Chapman, C.; Feit, E. M. *R for marketing research and analytics*, Springer, 2015

Métodos de Agrupamento

- Baseados em distâncias
  - √ Encontrar grupos que minimizem a distância entre membros do grupos e maximizem a distância dos membros dos grupos
  - √ Tipos:
    - Técnicas de agrupamento hierárquicas
    - Técnicas de agrupamento não hierárquicas

Baseados em modelos:

- √ Enxergam os dados como uma mistura de grupos provenientes de diferentes populações (parâmetros desconhecidos)
- √ Tentam modelar de maneira que a variância observada possa ser melhor representada por um pequeno número de grupos com características específicas distintas (médias e desvio padrão)
- √ Exemplo:
  - Mistura de normais
  - Modelo de classe latente com variáveis categóricas

Estágios do Agrupamento

1. Encontrar uma solução de agrupamento
2. Avaliar a solução

Passos

1. Transformar os dados para um particular método de agrupamento:
   - √ Alguns métodos exigem todos os dados quantitativos (*k-means*) ou todos os dados categóricos (*poLCA*)
2. Cálculo da matriz de distância

   (se necessário)
   - √ Alguns métodos exigem uma matriz de similaridade pré-calculada (hierárquicos)

3. Aplicação do método de agrupamento
   - √ Alguns métodos exigem especificar número de grupos desejado (*k-means*)
   - √ Salvar a solução
4. Para alguns métodos, analisar ainda mais o modelo para obter uma solução com k grupos
5. Examinar a solução no modelo com relação à estrutura subjacente e verificar se responde sua necessidade (prática)

- O fato de um algoritmo propor um modelo de agrupamento não significa que ele será de utilidade

Exemplo – Segmentação

- Dados brutos:

```
> seg.raw <- read.csv("rintro-chapter5.csv")
> seg.df <- seg.raw[, -7] # remove the known segment assignments
> summary(seg.df)

      age       gender          income           kids      ownHome
 Min.   :19.26   Female:157   Min.   : -5183   Min.   :0.00   ownNo :159
 1st Qu.:33.01   Male  :143   1st Qu.: 39656   1st Qu.:0.00   ownYes:141
 Median :39.49                Median : 52014   Median :1.00
 Mean   :41.20                Mean   : 50937   Mean   :1.27
 3rd Qu.:47.90                3rd Qu.: 61403   3rd Qu.:2.00
 Max.   :80.49                Max.   :114278   Max.   :7.00
  subscribe
 subNo :260
 subYes: 40
```

- Função para inspeção rápida das diferenças entre os grupos:

```
> # Função para inspeção rápida dos dados
> seg.sum <- function(data, groups){
+ aggregate(data, list(groups), function(x)mean(as.numeric(x)))
+ }
```

  - √ Converte todos os dados do grupo em quantidades numéricas e calcula média
  - √ Poderia ter sido usado a mediana (mais robusta)
  - √ Pode mostrar se há algo interessante (ou desinteressante) ocorrendo na solução

125

- Resultado no conjunto de dados:
  - √ Considerando classificação prévia

```
> seg.sum(seg.df, seg.raw$Segment)
     Group.1      age gender   income     kids  ownHome subscribe
1  Moving up 36.33114   1.30 53090.97 1.914286 1.328571     1.200
2 Suburb mix 39.92815   1.52 55033.82 1.920000 1.480000     1.060
3  Travelers 57.87088   1.50 62213.94 0.000000 1.750000     1.125
4  Urban hip 23.88459   1.60 21681.93 1.100000 1.200000     1.200
```

  - √ Médias de variáveis binárias mostra a proporção dos níveis das variáveis
  - √ Há diferenças óbvias nas médias dos grupos
  - √ Há sugestão de alguma estrutura de agrupamento interessante?

126

Algoritmos de Agrupamento

- Raramente podemos examinar todas as possibilidades de agrupamentos
  - √ Há algoritmos de agrupamento que não têm de verificar todas as configurações

127

Métodos de Agrupamentos Hierárquicos

- Agrupa as observações de acordo com sua similaridade
- Agrupamento inicia com cada observação em seu próprio cluster
- Agrupam sucessivamente observações ou grupos mais próximos, um por um
  - √ (Método aglomerativo)

128

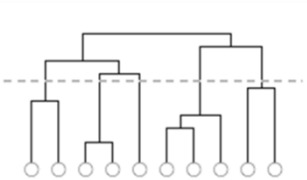
√ Em geral, são usadas em análises exploratórias dos dados com o objetivo de:
- identificar possíveis agrupamentos
- estimar o valor provável do número de grupos k

- Técnicas Não-Hierárquicas:
  √ É necessário que o valor do número de grupos já esteja pré-especificado pelo pesquisador

Propriedade de Hierarquia

- Em cada estágio do algoritmo, cada novo grupo formado é um agrupamento de grupos formados nos estágios anteriores
  √ Se 2 elementos aparecem juntos em algum estágio do processo, eles aparecerão juntos em todos os estágios subsequentes
- Uma vez unidos, estes elementos não poderão ser separados

Dendograma (ou Dendrograma)



√ Representa a árvore (ou história) do agrupamento
- Escala Vertical: nível de similaridade (ou dissimilaridade)
- Eixo Horizontal: elementos amostrais na ordem relacionada à história do agrupamento

Exemplo – Segmentação

- Distâncias Euclidianas:
  - √ Definidas apenas para variáveis quantitativas

```
> # matriz de distâncias
> d <- dist(seg.df[, c("age", "income", "kids")])
> as.matrix(d)[1:5, 1:5]# Matriz de distância 5 primeiras observações

          1         2         3         4         5
1     0.000 13936.531  5313.626 31559.178 29870.205
2 13936.531     0.000  8622.906 45495.698 43806.727
3  5313.626  8622.906     0.000 36872.800 35183.828
4 31559.178 45495.698 36872.800     0.000  1688.977
5 29870.205 43806.727 35183.828  1688.977     0.000
```

  - √ Matriz de distâncias sem os fatores
    - (gender, ownHome, subscribe)
  - √ Fatores não são irrelevantes
    - Trabalhar com métrica de discrepância adequada para dados mistos

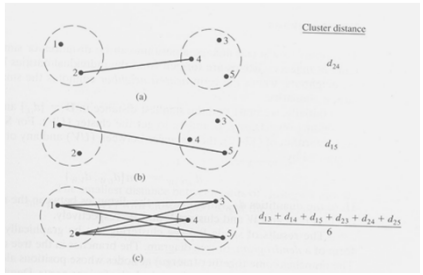
Matriz de dissimilaridades – todas variáveis

```
> # matriz de dissimilaridades - todas variáveis
> library(cluster)
> seg.dist <- daisy(seg.df)# metric = "gower", default na presença de fator
> as.matrix(seg.dist)[1:5, 1:5]

          1         2         3         4         5
1     0.000 13936.531  5313.626 31559.178 29870.205
2 13936.531     0.000  8622.906 45495.698 43806.727
3  5313.626  8622.906     0.000 36872.800 35183.828
4 31559.178 45495.698 36872.800     0.000  1688.977
5 29870.205 43806.727 35183.828  1688.977     0.000
```

- √ Métrica:
  - Distância de Gower (escalonada entre 0 e 1)

Métodos de Agrupamentos

- Medida de similaridade (ou distância) entre 2 conglomerados



Método do Centróide:

- √ Distância entre dois grupos é definida como sendo a distância entre os vetores de médias (centróides)
  - cada membro do par pertence a grupos diferentes

$C_1 = \{X_1, X_2\}$ e $C_2 = \{X_3, X_4, X_5\}$

$$d(C_1, C_2) = \left[(\bar{\mathbf{C}}_1 - \bar{\mathbf{C}}_2)'(\bar{\mathbf{C}}_1 - \bar{\mathbf{C}}_2)\right]^{1/2}$$

Distância Euclidiana entre os dois grupos

$$\bar{\mathbf{C}}_1 = \frac{1}{2}(X_1 + X_2)$$

$$\bar{\mathbf{C}}_2 = \frac{1}{3}(X_3 + X_4 + X_5)$$

Método de *Ward*

- Objetivo do procedimento:
  - √ Minimizar a perda de informação ao juntar 2 grupos
- Partição desejada:
  - √ A que produz os grupos mais heterogêneos entre si, com elementos homogêneos dentro de cada grupo
- Fundamento do método:
  - √ Em cada passo do agrupamento há mudança de variação entre os grupos e dentro dos grupos
  - √ Procedimento também denominado de mínima variância

137

- Métodos anteriores:
  - √ quando se passa de $(n - k)$ para $(n - k - 1)$ grupos o nível de fusão aumenta (nível de similaridade decresce) e a qualidade da partição decresce.
  - √ Variação entre grupos diminui e a variação dentro dos grupos a

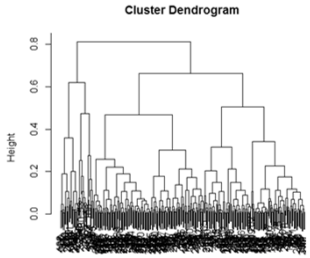
138

Exemplo – Segmentação

- Agrupamento hierárquico–ligação completa

```
> # Agrupamento hierárquico - ligação completa
> seg.hc <- hclust(seg.dist, method = "complete")
> plot(seg.hc)
```

√ Dendrograma – Representação do agrupamento

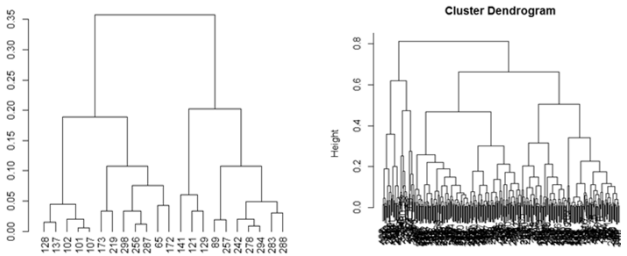


√ Altura representa dissimilaridade entre os elementos agrupados.
√ Leitura difícil

139

- Dendrograma – Zoom à esquerda

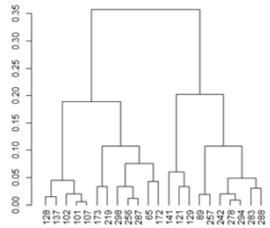
```
> # Zoom à esquerda
> plot(cut(as.dendrogram(seg.hc), h=0.5)$lower[[1]])
```



√ Mostra como cada consumidor foi sendo agrupado sucessivamente
  – ID: row.number

140

- Similaridade de observações:



√ Observações 101 e 107:
  − Agrupadas com pouca altura
√ Observações 278 e 294:
  − Agrupadas com pouca altura
√ Observações 173 e 141:
  − Agrupadas nível mais alto (relativamente discrepantes)

```
> seg.df[c(101, 107),]
         age gender   income kids ownHome subscribe
101 24.73796   Male 18457.85    1   ownNo     subYes
107 23.19013   Male 17510.28    1   ownNo     subYes
```

Similares em todas as variáveis

```
> seg.df[c(278, 294),]
         age gender   income kids ownHome subscribe
278 36.23860 Female 46540.88    1   ownNo     subYes
294 35.79961 Female 52352.69    1   ownNo     subYes
```

```
> seg.df[c(173, 141),]
         age gender   income kids ownHome subscribe
173 64.70641   Male 45517.15    0   ownNo     subYes
141 25.17703 Female 20125.80    2   ownNo     subYes
```

Diferem bastante nas 1ª.s variáveis

- Qualidade do ajuste:
  √ Método *cophenetic correlation* (CPCC)
    – Medida da fidelidade do dendrograma em preservar as distâncias pareadas dos dados originais
  √ Cálculo:

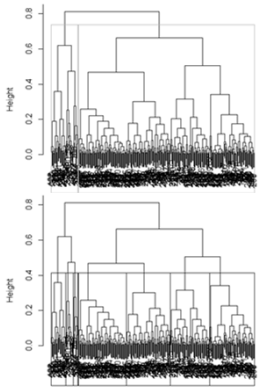
```
> # Qualidade do ajuste
> cor(cophenetic(seg.hc), seg.dist)
[1] 0.7682436
```

    – CPCC > 0,7
      ▪ Indica um ajuste relativamente forte, significando que a árvore hierárquica representa bem a distância entre consumidores

- Grupos no dendrograma
  √ Partição conforme altura:
    – h = 0,7

```
> # partição conforme altura
> plot(seg.hc)
> rect.hclust(seg.hc, h=0.7, border="green")
```

    – k = 2 grupos



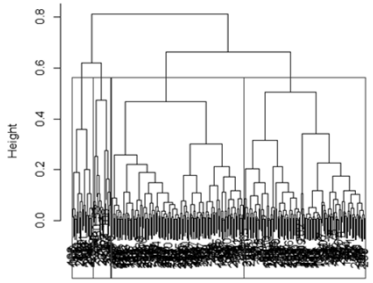
    – h = 0,4

```
> plot(seg.hc)
> rect.hclust(seg.hc, h=0.4, border="blue")
```

    – k = 7 grupos

- Grupos no dendrograma
  √ Partição em número desejado de clusters:
    – k = 4

```
# partição em qte de clusters
plot(seg.hc)
rect.hclust(seg.hc, k = 4, border = "red")
```



Alocação por grupos:

√ Quantidade de itens alocados em cada grupo:

```
> # Quantidade de objetos por grupo
> seg.hc.segment <- cutree(seg.hc, k = 4)
> table(seg.hc.segment)
seg.hc.segment
  1   2   3   4
124 136  18  22
```

– Grupos 1 e 2 dominam a atribuição dos grupos

146

√ Inspeção das variáveis por grupo:

```
> # Inspeção das variáveis por grupo
> seg.sum(seg.df, seg.hc.segment)
  Group.1      age   gender   income     kids  ownHome subscribe
1       1 40.78456 2.000000 49454.08 1.314516 1.467742         1
2       2 42.03492 1.000000 53759.62 1.235294 1.477941         1
3       3 44.31194 1.388889 52628.42 1.388889 2.000000         2
4       4 35.82935 1.545455 40456.14 1.136364 1.000000         2
```

– 1 e 2 se diferenciam de 3 e 4 pela variável *subscribe*

– Entre os que não subscrevem (*subscribe* = 1)

- Grupo 1: todos homens (*gender* = 2)
- Grupo 2: todas mulheres (*gender* = 1)

– Entre os que subscrevem (*subscribe* = 2)

- Grupo 3: todos possuem casa (*ownHome* = 2)
- Grupo 2: todos não possuem casa (*ownHome* = 1)

– Classificação pode não ser interessante do ponto de vista prático

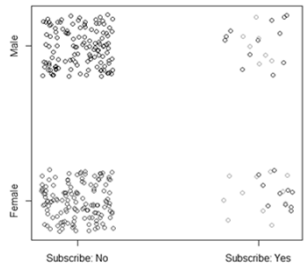
- Focar quais segmentos entre quem não subscreve?

147

Plot dos 4 segmentos:

√ Por *gender* e *subscribe*

```
# Plot dos 4 segmentos
plot(jitter(as.numeric(seg.df$gender)) ~
        jitter(as.numeric(seg.df$subscribe)),
        col = seg.hc.segment, yaxt = "n", xaxt = "n", ylab = "", xlab = "")
axis(1, at = c(1, 2), labels = c("Subscribe: No", "Subscribe: Yes"))
axis(2, at = c(1, 2), labels = levels(seg.df$gender))
```



√ Não subscrevem:

– Partição de 2 segmentos correlacionados com *gender* perfeitamente.

√ Gráfico para inspeção rápida

148

Comentários

√ Técnicas de statistical learning frequentemente adotam o caminho de menor resistência

√ Tornou mais influentes os fatores binários

– (*gender*, *subscribe*, *ownHome*)

√ Devem ser tentados vários métodos para encontrar solução de utilidade

149

Métodos Hierárquicos – Comentários Finais

- Fontes de erros e de variação não são formalmente considerados
  - √ Esses métodos são sensíveis a *outliers* ou pontos de perturbação
- Verificar sensibilidade da configuração dos grupos
  - √ Não permitem a realocação de objetos que possam ter sido agrupados incorretamente nos estágios iniciais

150

- Recomenda-se tentar vários métodos de agrupamento e de atribuição de distâncias (similaridades)
- Empates na matriz de distâncias podem produzir múltiplas soluções ao problema de agrupamento hierárquico
- A maioria dos métodos produz clusters esféricos ou elípticos

151

Técnicas de Agrupamento Não Hierárquicas

- Objetivo:
  - √ Encontrar diretamente uma partição de *n* elementos em *k* grupos
  - √ Requisitos:
    - coesão interna (semelhança interna)
    - isolamento (separação) dos clusters formados
- Busca da "melhor" partição de ordem *k*
  - √ Satisfaz algum critério de qualidade
  - √ Procedimentos computacionais para investigar partições 'quase' ótima

    (inviável a busca exaustiva)

152

- Métodos Não Hierárquicos *vs.* Hierárquicos :
  - √ Especificação prévia do número de cluster

    (ao contrário das técnicas aglomerativas)
  - √ Novos grupos podem ser formados pela divisão (ou junção) de grupos já combinados:
    - Se em um passo do algoritmo, dois elementos tiverem sido colocados em um mesmo grupo, não significa que estarão juntos na partição final
    - Não é mais possível a construção de dendogramas
  - √ Em geral, são do tipo iterativo

153

- √ Tem maior capacidade de analisar grande número de dados
- √ A matriz de distância não tem de ser calculada e os dados básicos não precisam ser armazenados durante a execução do procedimento
- √ Métodos hierárquicos são mais adequados para agrupar itens que variáveis

Métodos Não Hierárquicos – Estrutura

- Iniciam-se:
  1. partição inicial de itens em grupos
  2. conjunto inicial de sementes que formarão o núcleo dos clusters
- Escolha das configurações iniciais pode afetar partição final
  - √ Viés na escolha das sementes iniciais
  - √ Alternativas:
    - Seleção aleatória de sementes
    - Partição aleatória de itens em grupos iniciais

Método das *k*-Médias

- Provavelmente, um dos mais conhecidos e mais utilizados
- Ideia Básica:
  - √ Cada elemento amostral é alocado àquele *cluster* cujo centróide é o mais próximo do elemento

- Tenta encontrar grupos que são mais compactos em termos da soma dos quadrados dos desvios de cada observação a partir do centróide (centro multivariado)
- Trabalha com distância Euclidiana
  - √ Adequado apenas para dados quantitativos ou dados que podem ser razoavelmente transformados em números

Exemplo – Segmentação

- Há mistura de dados quantitativos e binários
  - √ Binários podem ser transformados em numéricos sem alteração de significado
  - √ Agrupamento de dados binários não é ótimo por k-médias

    (será tentado, pois os dados são mistos)

√ Recodificação dos fatores binários:

```
> # Recodificação dos fatores binários
> seg.df.num <- seg.df
> seg.df.num$gender <- ifelse(seg.df$gender == "Male", 0, 1)
> seg.df.num$ownHome <- ifelse(seg.df$ownHome == "ownNo", 0, 1)
> seg.df.num$subscribe <- ifelse(seg.df$subscribe == "subNo", 0, 1)
> summary(seg.df.num)
 age             gender           income            kids
 Min.   :19.26   Min.   :0.0000   Min.   : -5183   Min.   :0.00
 1st Qu.:33.01   1st Qu.:0.0000   1st Qu.: 39656   1st Qu.:0.00
 Median :39.49   Median :1.0000   Median : 52014   Median :1.00
 Mean   :41.20   Mean   :0.5233   Mean   : 50937   Mean   :1.27
 3rd Qu.:47.90   3rd Qu.:1.0000   3rd Qu.: 61403   3rd Qu.:2.00
 Max.   :80.49   Max.   :1.0000   Max.   :114278   Max.   :7.00
    ownHome       subscribe
 Min.   :0.00   Min.   :0.0000
 1st Qu.:0.00   1st Qu.:0.0000
 Median :0.00   Median :0.0000
 Mean   :0.47   Mean   :0.1333
 3rd Qu.:1.00   3rd Qu.:0.0000
 Max.   :1.00   Max.   :1.0000
```

√ Médias de variáveis binárias mostram proporção

√ Agrupamento por k-médias

- Adotado k = 4 grupos

```
> # Método k-means
> set.seed(96743)
> seg.k <- kmeans(seg.df.num, centers = 4)
```

√ Verificação das médias por grupo proposto

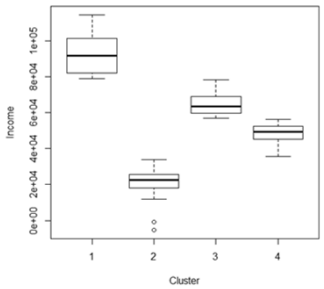
```
> # Verificação das médias por grupo
> seg.sum(seg.df, seg.k$cluster)
 Group.1      age   gender   income      kids  ownHome subscribe
1       1 56.37245 1.428571 92287.07 0.4285714 1.857143  1.142857
2       2 29.58704 1.571429 21631.79 1.0634921 1.301587  1.158730
3       3 44.42051 1.452632 64703.76 1.2947368 1.421053  1.073684
4       4 42.08381 1.454545 48208.86 1.5041322 1.528926  1.165289
```

√ Há algumas diferenças interessantes

- Grupos parecem variar por *age*, *gender*, *kids*, *income* e *ownHome*

Distribuição de renda por grupo proposto:

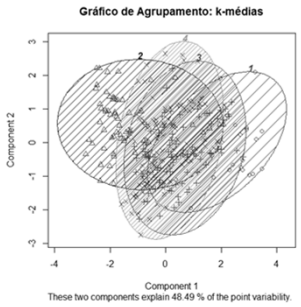
```
> # distribuição de renda por grupo proposto
> boxplot(seg.df.num$income ~seg.k$cluster, ylab = "Income", xlab = "Cluster")
```



√ Diferenças substanciais de renda por segmento

- Visualização dos *clusters*:
  - √ Redução dimensional por cp's ou EMD

```
> library(cluster)
> clusplot(seg.df, seg.k$cluster, color = TRUE, shade = TRUE, labels = 4,
+      lines = 0, main = "Gráfico de Agrupamento: k-médias")
```



  - √ Sobreposição significativa dos grupos 3 e 4
    - √ (nessa redução dimensional)
  - √ Grupos 1 e 2 estão diferenciados moderadamente

- √ Solução A – hierárquico completo

	Group.1	age	gender	income	kids	ownHome	subscribe
1	1	40.78456	2.000000	49454.08	1.314516	1.467742	1
2	2	42.03492	1.000000	53759.62	1.235294	1.477941	1
3	3	44.31194	1.388889	52628.42	1.388889	2.000000	2
4	4	35.82935	1.545455	40456.14	1.136364	1.000000	2

- √ Solução B – k-means

	Group.1	age	gender	income	kids	ownHome	subscribe
1	1	56.37245	1.428571	92287.07	0.4285714	1.857143	1.142857
2	2	29.58704	1.571429	21631.79	1.0634921	1.301587	1.158730
3	3	44.42051	1.452632	64703.76	1.2947368	1.421053	1.073684
4	4	42.08381	1.454545	48208.86	1.5041322	1.528926	1.165289

- √ Solução B mais interessante
  - Grupos claramente diferenciados em variáveis chave
  - Pode-se cruzar membro do grupo com variáveis chave
- √ Estratégia possível:
  - Grupo 1: moderadamente bem diferenciado, com maior renda média
    - Pode ser bom alvo para campanha potencial

Limitação do *k-Means*

- Exige especificar a quantidade de clusters
  - √ Pode ser difícil determinar se uma solução é melhor que outra
  - √ No exemplo da Segmentação:
    - Deveria ser repetida a análise para k=3 e k=5 e determinar qual solução oferece resultado mais eficiente

- Há fortes argumentos para não se fixar o número de *clusters k*
  - √ Mesmo sabendo-se que a população consiste de *k* grupos, dependendo do método de amostragem, pode não aparecer na amostra os dados provenientes de um grupo mais raro
    - Forçar *k* grupos levaria a *clusters* sem sentido
  - √ Em casos em que o algoritmo requer o uso de um valor especificado de *k*, é sempre uma boa idéia executar novamente o algoritmo para diversas escolhas de *k*

Referências

Bibliografia Recomendada

- ALBERT, J.; RIZZO, M. *R by Example*. Springer, 2012.
- CHAPMAN, C.; FEIT, E. M. *R for marketing research and analytics*. Springer, 2015.
- KLEIBER, C.; ZEILEIS, A. *Applied econometrics with R*. Springer, 2008.
- DALGAARD, P. *Introductory statistics with R*. Springer, 2008.