

# Lista de Exercícios - Continuação de PCA e Análise fatorial

Renato Assunção - DCC, UFMG

2018

1. Uma outra maneira importante de apresentar (e entender) os componentes principais está no Result 8.1 de `Johnson_Wichern_Book.pdf` (o livro que recomendei para esta parte do curso), na página 432. Seu trabalho nest exercício é provar o resultado 8.1 APENAS para o primeiro componente principal. A prova está no próprio livro. Você deverá também provar o resultado (2-51) da página 80 APENAS para o primeiro autovetor. A prova também está no livro.

Mais claramente, prove os seguintes resultados:

- (a) Resultado (2-51) da página 80: Seja  $\mathbf{B}$  uma matriz definida positiva  $p \times p$  com autovalores  $\lambda_1 \geq \lambda_2 \geq \dots \lambda_p > 0$  e associados autovetores  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p$  de comprimento (ou norma) 1. Então

$$\max_{\mathbf{x} \neq \mathbf{0}} \frac{\mathbf{x}'\mathbf{B}\mathbf{x}}{\mathbf{x}'\mathbf{x}} = \lambda_1$$

e este máximo é atingido quando  $\mathbf{x} = \mathbf{v}_1$ .

- (b) Resultado 8.1 da página 432: Seja  $\Sigma$  a matriz de covariância do veor aleatório  $\mathbf{bsX} = (X_1, \dots, X_p)'$  com autovalores  $\lambda_1 \geq \lambda_2 \geq \dots \lambda_p > 0$  e associados autovetores  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p$  de comprimento (ou norma) 1. Então a combinação linear  $Y = l_1 X_1 + \dots + l_p X_p = \mathbf{l}'\mathbf{X}$  com comprimento  $\|\mathbf{l}\| = 1$  e que maximiza  $V(Y)$  é obtida ao tomarmos  $\mathbf{l}$  igual ao primeiro autovetor. Neste caso,  $Y = \mathbf{v}_1'\mathbf{X}$  e a variância desta variável atinge  $V(Y) = \lambda_1$ .

Você deverá MEMORIZAR estas duas provas. Este exercício será repetido na parte sem consulta da segunda prova de FECD.

2. Faça o exercício 8.11 de `Johnson_Wichern_Book.pdf`. Para responder a questão, leia o terceiro parágrafo da página 439. Os dados podem ser obtidos no site do livro <http://esminfo.prenhall.com/math/johnsonwichern/data.html>. Existe também em outros sites na web.
3. Mostre que a matriz de covariância

$$\Sigma = \begin{bmatrix} 1.24 & 0.48 & 0.16 \\ 0.48 & 0.86 & 0.12 \\ 0.16 & 0.12 & 0.14 \end{bmatrix}$$

do vetor aleatório  $\mathbf{X} = (X_1, X_2, X_3)^t$  pode ser gerado pelo modelo de análise fatorial com um único fator dado por

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 1.0 \\ 0.0 \\ -1.0 \end{bmatrix} + \begin{bmatrix} 0.8 \\ 0.6 \\ 0.2 \end{bmatrix} F + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}$$

Obtenha a matrix  $\Psi$  com as variâncias específicas dos termos  $\varepsilon_k$  implicadas pelo modelo.

4. Neste exercício, você vai trabalhar com os dados de uma análise química de vinhos. Você vai ler uma matriz com 178 amostras de diferentes vinhos. Haverá uma linha para cada vinho. A primeira coluna indica o cultivar do vinho (entenda como o tipo de uva usado na fabricação do vinho)

tal como Sauvignon Blanc, Cabernet ou Chardonnay (rotulados como 1, 2 ou 3). As 13 colunas seguintes contêm as concentrações de 13 diferentes compostos químicos na amostra.

O objetivo é diferenciar entre os 3 tipos de vinho com base na sua composição química representada pelo vetor 13-dimensional  $\mathbf{X}$ . Você precisa criar uma regra para prever o tipo de vinho (a primeira coluna) a partir das 13 variáveis da composição química. Vamos verificar que, ao invés de usarmos todas as 13 variáveis, poderemos nos basear em dois índices-resumo, os dois primeiros PCAs. Eles resumem quase toda a informação (ou variabilidade) simultânea das 13 variáveis.

Estude o script R abaixo. De propósito, ele tem uma quantidade *mínima* de comentários. Procure identificar o que cada linha está fazendo.

WARNING: o help da função `prcomp` é confuso, misturando PCA e análise fatorial nas explicações.

```
arq = "http://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data"
wine=read.table(arq, sep=",")
```

```
head(wine)
pairs(wine[,2:6])
round(100*cor(wine[,2:14]))
round(apply(wine[,2:14], 2, sd),2)
```

```
wine.pca = prcomp(wine[,2:14], scale. = TRUE)
summary(wine.pca)
wine.pca$sdev
sum((wine.pca$sdev)^2)
screeplot(wine.pca, type="lines")
```

```
# Barplot das variancias acumuladas
barplot(cumsum(wine.pca$sdev^2)/sum(wine.pca$sdev^2))
# os dois primeiros PCA's explicam aprox 60% da variancia total
# os 5 primeiros explicam aprox 80%
```

```
# Os autovetores
dim(wine.pca$rot)
```

```
# O 1o autovetor
wine.pca$rot[,1]
```

```
# O 2o autovetor
wine.pca$rot[,2]
```

```
# Coordenadas dos pontos ao longo do primeiro componente
fscore1 = wine.pca$x[,1]
```

```
# Coordenadas dos pontos ao longo do segundo componente
fscore2 = wine.pca$x[,2]
```

```
# plot dos pontos projetados
plot(fscore1, fscore2, pch="*", col=wine[,1]+8)
```

Seja  $\mathbf{X}_i = (X_{i1}, \dots, X_{i13})$  a *linha*  $i$  da matriz `wine`. Seja  $\mathbf{Z}_i = (Z_{i1}, \dots, Z_{i13})$  a *linha*  $i$  da matriz `wine` PADRONIZADA. Isto é,  $Z_{ij} = (X_{ij} - \bar{x}_j)/s_j$  onde  $\bar{x}_j$  é a média aritmética e  $s_j$  é o desvio-padrão da coluna  $j$  da matriz `wine`. Esta matriz padronizada é obtida com o comando `z = scale(wine[2:14])`:

```
z = scale(wine[2:14]) # colunas de wine padronizadas (desvios padronizados)
round(apply(z, 2, mean), 5) # media das colunas da matriz
round(apply(z, 2, sd), 5) # sd das colunas da matriz
```

Vamos considerar  $\mathbf{Z}_i$  como um *vetor-coluna* 13-dimensional. Ao invés de usarmos o vetor  $\mathbf{Z}_i$ , estamos usando apenas o vetor  $\mathbf{Y}_i$  composto pelos dois índices formados pelos dois primeiros componentes principais:

$$\mathbf{Y}_i = \begin{bmatrix} Y_{i1} \\ Y_{i2} \end{bmatrix} = \begin{bmatrix} \mathbf{v}'_1 \mathbf{Z}_i \\ \mathbf{v}'_2 \mathbf{Z}_i \end{bmatrix}$$

onde  $\mathbf{v}_1$  e  $\mathbf{v}_2$  são os dois primeiros autovetores da matriz de correlação de  $\mathbf{X}$ .

- Preencha os locais com (??) com os valores numéricos corretos (duas casa decimais apenas):

$$\begin{aligned} Y_{i1} &= (??)Z_{i1} + (??)Z_{i2} + (??)Z_{i3} + \dots + (??)Z_{i,13} \\ Y_{i2} &= (??)Z_{i1} + (??)Z_{i2} + (??)Z_{i3} + \dots + (??)Z_{i,13} \end{aligned}$$

- O último gráfico do script R acima é um plot dos pontos  $\mathbf{Y}_i$  dos 178 vinhos. Identifique três regiões do plano  $Y_1, Y_2$  que podem ser usadas para classificar futuras amostras de vinhos em uma das três categorias. Pode apenas esboçar grosseiramente no gráfico a mão livre ou descrever as regiões em palavras.
- Suponha que uma nova amostra de vinho tem sua composição química medida e encontra-se

$$\mathbf{x} = (13.95, 3.65, 2.25, 18.4, 90.18, 1.55, 0.48, 0.5, 1.34, 10.2, 0.71, 1.48, 587.14)$$

Obtenha seu vetor  $\mathbf{z}$ , as suas coordenadas  $(y_1, y_2)$  e prediga o seu tipo. Confira sua resposta no final desta lista.

- O que podemos fazer com isto? Imagine uma medida de qualidade de cada um dos vinhos. Ela poderia ser obtida tomando-se a nota média que um conjunto de experts ou apreciadores de vinhos poderia dar a um dos vinhos. A seguir, podemos tentar, no espaço reduzido das duas componentes, entender quais os valores das componentes levam a um bom vinho. Diferentes combinações das 13 variáveis químicas podem levar a uma mesma nota alta.
5. O arquivo `beer.txt` é um dataset da página de Karl Wuensch, East Carolina University. Uma nova cervejaria está interessada em conhecer o comportamento de escolha do consumidor de cerveja artesanal. Um grupo de 231 consumidores avaliou a importância de sete qualidades ao decidir se deve ou não comprar uma cerveja. Para cada qualidade, foi dada uma nota numa escala de 0 a 100 para sua importância. As sete qualidades ou variáveis são as seguintes:
- COST: baixo custo por volume (300ml de cerveja)
  - SIZE: grande tamanho da garrafa (volume)
  - ALCOHOL: alto percentual de álcool da cerveja
  - REPUTATION: boa reputação da marca
  - COLOR: a cor da cerveja
  - AROMA: agradável aroma da cerveja
  - TASTE: gosto saboroso da cerveja

A variável SES é uma categoria de status socioeconômico (valores maiores significam status mais elevados). A variável grupo não é explicada, não sei do que se trata. Ignore-a durante o exercício.

O script abaixo executa o seguinte: Leia os dados numa matriz. Use `summary(beer)` (ou olhe os dados na tela) para verificar que existem 11 NAs na variável `AROMA`. Obtenha a matriz de covariância  $S$  das 7 variáveis de qualidade e verifique que os seus desvios-padrão não são muito distintos. Obtenha a matriz de correlação  $R$ .

```

arq = "http://www.unt.edu/rss/class/mike/data/beer.txt"
beer <- read.table(arq, header=TRUE)
# beer = as.matrix(read.table("beer.txt", header=T))

head(beer)
summary(beer)

S = var(beer[,1:7], na.rm=T)
S
sqrt(diag(S)) # sd's not very different

R = cor(beer[,1:7], use ="complete.obs")

round(100*R)

```

Você deve gastar um tempo olhando a matriz de correlação  $R$ , a menos que ela seja muito grande. Você está planejando usar PCA ou FA para capturar a essência das correlações nesta matriz. Observe que há muitas correlações grandes e médias em  $R$ . Todas as variáveis tem algumas correlações grandes, com a exceção de `reputation` que é moderadamente (e negativamente) correlacionada com todo o resto. É óbvio que existe uma estrutura de correlação entre as variáveis.

O pacote `corrplot` permite visualizar a matriz de correlação  $R$  de um jeito muito legal. Veja em <http://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html>. Instale e carregue este pacote. Faça um gráfico na forma da matriz  $R$  em que cada célula possui uma elipse representando grau de correlação entre as duas variáveis. Quanto mais achatada e parecida com uma linha reta, mais correlacionadas são as duas variáveis. Se a elipse for parecida com um círculo, é sinal de que a correlação é próxima de zero. Neste caso, a imagem estará quase transparente. Correlações positivas são azuis, negativas são vermelhas.

```

library(corrplot)
corrplot(R, method = "ellipse")
# plotando as elipses e os valores das correlacoes
corrplot.mixed(R, upper = "ellipse")
# rearranjando as linhas e colunas para agrupar variaveis com correlacoes parecidas
corrplot.mixed(R, order = "AOE", upper = "ellipse", cl.align = "r")

```

Parece haver dois grupos de variáveis, um formado por `COST`, `ALCOHOL`, `SIZE` e outro formado por `COLOR`, `AROMA`, `TASTE`. Elas são bem positivamente correlacionadas dentro de cada grupo e, ao mesmo tempo, pouco correlacionadas com as variáveis do outro grupo. Uma variável, `REPUTATION`, forma um grupo à parte, sendo fracamente e negativamente correlacionada com todas as outras seis.

Gere uma nova matriz eliminando as poucas linhas em que existem NAs. A seguir, obtenha os autovetores e autovalores da matriz de covariância  $S$  com a função `eigen`. Vamos trabalhar com matriz de covariância porque os desvios-padrão das setes variáveis são parecidos.

```

newbeer = na.omit(beer)
S = cov(newbeer[,1:7])
fit = eigen(S) # usa o algoritmo QR em cima da matriz S
# autovalores
fit$values
# autovetores
fit$vectors

```

Na análise acima, tivemos de gerar a matriz de covariância e, a seguir, passá-la à função `eigen`. A função `eigen` não enxerga mais os dados originais, somente a matriz  $R$ . Outra maneira é fornecer diretamente a matriz  $X$  de dados  $n \times p$  e pedir que os componentes principais da matriz de covariância induzida (ou da matriz de correlação) seja calculada. A função `prcomp` faz isto através da decomposição SVD de  $X$ .

```
pca.beer = prcomp(newbeer[,1:7])

# Se quiser obter PCA da matriz de correlação, use
# pca.beer = prcomp(newbeer[,1:7], scale. = TRUE)

# Os 7 autovetores
pca.beer$rot

# Os 7 autovalores
(pca.beer$sdev)^2

# verifique que os autovetores acima são os mesmos daqueles retornados por eigen.
# verifique que os autovetores têm norma euclidiana = 1.
# Por exemplo, o 1o PCA:
sum(pca.beer$rot[,1]^2)

# Grafico scree com os 7 autovalores (ou variancias de cada PCA)
plot(pca.beer)

# Barplot das variancias acumuladas indicando a escolha de 2 PCAs
barplot(cumsum(pca.beer$sdev^2))

# Resumo
summary(pca.beer)

# Note que o quadrado da linha Standard deviation acima é igual aos autovalores
# obtidos com fit$values

# Vamos usar apenas os dois 1os PCs para representar R com dois fatores
# Carga do Fator = sqrt(LAMBDA) * EIGENVECTOR

cargafat1 = pca.beer$sdev[1] * pca.beer$rot[,1]
cargafat2 = pca.beer$sdev[2] * pca.beer$rot[,2]

# matriz de cargas
L = cbind(cargafat1, cargafat2)

rownames(L) = rownames(R)[1:7]

round(L, 2)

plot(L, type="n", xlim=c(-40, 20), ylim=c(-10, 25))
text(L, rownames(L))
abline(h=0)
abline(v=0)
```

A interpretação dos resultados obtidos não é simples. Com os eixos rotacionados conseguiremos um resultado bem mais interpretável. Não existe uma rotina nativa em R para obter a rotação ótima

dos fatores no caso da estimação pelo método de componentes principais. Em R, a rotação ótima está implementada apenas para a estimação das cargas  $L$  por meio do método de máxima verossimilhança, um método que veremos em breve. Para o caso do método de componentes principais, o último gráfico mostra que uma rotação horária de aproximadamente  $90^\circ + 15^\circ$  ou  $\pi/2 + 15(\pi/180)$  deve colocar a maioria dos pontos em apenas um dos dois eixos ortogonais:

```
# Fazendo manualmente uma rotacao horaria de pi/2+15*pi/180
phi = pi/2 + 15*(pi/180)
T = matrix(c(cos(phi), -sin(phi), sin(phi), cos(phi)), ncol=2, byrow=T)

Lstar = L %*% T # usando a multiplicacao por linha da matriz L

plot(Lstar, type="n", xlim=c(-20, 30), ylim=c(-15, 35))
text(Lstar, rownames(L))
abline(h=0); abline(v=0)

round(Lstar,2)
```

A interpretação dos fatores é bem mais simples agora. O primeiro fator tem cargas positivas e grandes em **COLOR**, **AROMA**, **TASTE** e uma carga negativa moderada em **REPUTATION**. Alguém com uma nota (ou escore) muito elevado neste fator é alguém que preza e diferencia qualidades ligadas ao paladar da cerveja e também seus aspectos estéticos. Este componente poderia ser chamado de *Degustador*.

Os indivíduos que tiverem seu segundo fator muito positivo terão dado notas altas para os aspectos de **COST**, **ALCOHOL**, **SIZE** e, ao mesmo tempo, dado uma nota moderadamente baixa para **REPUTATION**. Alguém que possui uma nota muito alta neste segundo fator é alguém que gosta de muita cerveja barata e com muito álcool, e não se importa muito com a reputação da cerveja. Este componente poderia ser chamado de *Bebum Barato*.

Para obter uma estimativa das variâncias dos fatores específicos (isto é, da matriz  $\Psi$ ), usamos o código abaixo:

```
matpsi = diag(diag(S - Lstar %*% t(Lstar)))
round(matpsi, 2)

sum( (S - Lstar %*% t(Lstar) - matpsi)^2 )/sum(S^2)
```

O último comando mostra que a matriz residual  $\Sigma - L^*(L^*)' - \Psi$  tem uma soma de suas entradas (ao quadrado) muito pequena em comparação com a soma das entradas na matriz de covariância  $\Sigma$  (apenas 0.005303857 ou 0.5%).

No nosso modelo, os indivíduos recebem escores *independentes* destes dois fatores. Eles não são fatores competidores, um indivíduo pode receber altas doses dos dois fatores. Ele pode gostar de tomar muita cerveja com muito álcool e que seja barata. Isto é, ter um escore alto no fator 2. Ao mesmo tempo, este mesmo indivíduo pode apreciar também as cervejas mais refinadas, mais caras e com mais sabor e aroma. Isto é, ter um escore alto também no fator 1. Em suma, ele pode ser um esteta que adora se embriagar.

Para encontrar uma estimativa dos escores dos dois fatores para cada um dos 220 indivíduos que restaram na matriz **newbeer** após eliminar as 11 linhas com NAs, usamos o procedimento de regressão linear. Lembre-se que o modelo de análise fatorial estabelece que as 7 notas do indivíduo  $i$  é representada por

$$\underset{(7 \times 1)}{\mathbf{X}_i} = \underset{(7 \times 1)}{\boldsymbol{\mu}} + \underset{(7 \times 2)}{\mathbf{L}^*} \underset{(2 \times 1)}{\mathbf{F}_i} + \underset{(7 \times 1)}{\boldsymbol{\epsilon}_i}$$

onde  $\mathbf{F}'_i = (F_{1i}, F_{2i})$  são os escores (ou as doses) que o indivíduo  $i$  possui dos fatores 1 e 2. Como observamos diretamente  $\mathbf{X}_i$  e como estimamos a média populacional  $\boldsymbol{\mu}$  e a matriz de cargas rotacionadas  $\mathbf{L}^*$ , podemos usar mínimos quadrados ou regressão linear para estimar os escores  $F_{1i}$  e  $F_{2i}$ .

Por exemplo, o primeiro indivíduo na matriz `newbeer` tem a sua representação fatorial estimada por

$$\mathbf{X}_1 = \begin{bmatrix} 90 \\ 80 \\ 70 \\ 20 \\ 50 \\ 70 \\ 60 \end{bmatrix} = \hat{\boldsymbol{\mu}} + \hat{\mathbf{L}}^* \hat{\mathbf{F}}_1 + \hat{\boldsymbol{\epsilon}}_1 = \begin{bmatrix} 47.25 \\ 43.50 \\ 46.50 \\ 48.25 \\ 51.00 \\ 44.75 \\ 67.25 \end{bmatrix} + \begin{bmatrix} 0.11 & 31.74 \\ 4.84 & 32.07 \\ 3.09 & 30.19 \\ -12.95 & -10.83 \\ 25.81 & 0.05 \\ 24.79 & -1.37 \\ 22.94 & -2.28 \end{bmatrix} \begin{bmatrix} F_{1i} \\ F_{2i} \end{bmatrix} + \begin{bmatrix} \hat{\epsilon}_{11} \\ \hat{\epsilon}_{21} \\ \hat{\epsilon}_{31} \\ \hat{\epsilon}_{41} \\ \hat{\epsilon}_{51} \\ \hat{\epsilon}_{61} \\ \hat{\epsilon}_{71} \end{bmatrix}$$

A matriz  $\hat{\mathbf{L}}^*$  está na matriz `Lstar` no final do script R e é a mesma para todos os indivíduos. O vetor  $\hat{\boldsymbol{\mu}}$  também é o mesmo para todos os indivíduos e é obtido simplesmente tomando a média aritmética de cada uma das sete qualidades de modo que  $\boldsymbol{\mu}$  é aproximadamente igual ao resultado do comando `mu = apply(newbeer[,1:7], 2, mean)`.

```
> apply(newbeer[,1:7], 2, mean)
  COST    SIZE ALCOHOL REPUTAT    COLOR    AROMA    TASTE
47.25   43.50   46.50   48.25   51.00   44.75   67.25
```

Assim, para o indivíduo  $i$  podemos estimar seus escores  $F_{1i}$  e  $F_{2i}$  pelos valores  $\hat{F}_{1i}$  e  $\hat{F}_{2i}$  que minimizam o comprimento (ao quadrado) da diferença entre  $\mathbf{X}_i$  e  $\hat{\boldsymbol{\mu}} + \hat{\mathbf{L}}^* \mathbf{F}_i$ :

$$\underset{\mathbf{F}_i}{\operatorname{argmin}} \|\mathbf{X}_i - \hat{\boldsymbol{\mu}} - \hat{\mathbf{L}}^* \mathbf{F}_i\|^2$$

Ou seja, para o indivíduo  $i = 1$ , queremos o vetor  $\mathbf{F}_1$  que minimize a norma euclidiana (ao quadrado) do vetor

$$\mathbf{X}_1 - \hat{\boldsymbol{\mu}} - \hat{\mathbf{L}}^* \mathbf{F}_1 = \begin{bmatrix} 90 - 47.25 \\ 80 - 43.50 \\ 70 - 46.50 \\ 20 - 48.25 \\ 50 - 51.00 \\ 70 - 44.75 \\ 60 - 67.25 \end{bmatrix} - \begin{bmatrix} 0.11 & 31.74 \\ 4.84 & 32.07 \\ 3.09 & 30.19 \\ -12.95 & -10.83 \\ 25.81 & 0.05 \\ 24.79 & -1.37 \\ 22.94 & -2.28 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{21} \end{bmatrix}$$

O seguinte código em R faz isto através de loop sobre as linhas da matriz `newbeer`:

```
## Factor scores dos n=220 individuos
factors = matrix(0, nrow=nrow(beer), ncol=2)
mu = apply(newbeer[,1:7], 2, mean)
for(i in 1:nrow(newbeer)){
  y = newbeer[i, 1:7] - mu
  factors[i,] = lm(y ~ 0 + Lstar)$coef
}
```

Podemos visualizar os fatores de cada um dos 220 indivíduos pedindo um plot da matriz `factors`:

```
plot(factors, xlab="fator 1", ylab="fator2")

# mas... onde estao os 220 individuos?
# Varios individuos poduziram o MESMO vator x --> estimamos com os mesmos fatores

plot(jitter(factors, amount=0.05), xlab="fator 1", ylab="fator2")
```

Como vários indivíduos produziram o mesmo vetor  $\mathbf{X}_i$  seus fatores  $\mathbf{F}_i$  também coincidem. Assim, o comando `jitter` foi usado. Ele perturba as coordenadas de cada ponto aleatoriamente com um ruído uniforme entre `-amount` e `+amount`. No novo plot, podemos enxergar todos os indivíduos.

6. Opcional (não precisa entregar): Assista à aula de Andrew Ng sobre Componentes Principais no program de pós-graduação em CS de Stanford. O vídeo está no YouTube em <https://www.youtube.com/watch?v=ey2PE5xi9-A> a partir de 37:20. A primeira parte do vídeo você pode pular. Ela trata de Anaálise Fatorial mas ele estima a matriz de cargas  $\mathbf{L}$  usando máxima verossimilhança e o algoritmo EM, assuntos que ainda vamos estudar (nós estimamos  $\mathbf{L}$  usando os componentes principais). Esta aula de Andrew Ng não é a aula simplificada do coursera, é a aula real que os alunos de Machine Learning em Stanford precisam fazer. Você não tem de entregar nada neste exercício.
7. Opcional (não precisa entregar): Repita a análise do documento `FAinR.pdf`, uma boa análise prática usando a ideia de fatores latentes.