EST171 - Aprendizado de Máquina
Departamento de Estatística
Universidade Federal de Minas Gerais

Lista 4

Henrique Aparecido Laureano Matheus de Vasconcellos Barroso

Novembro de 2016

Sumário

Exercício I

Seu objetivo é usar as técnicas de redução de dimensionalidade (clustering) e de regras de associação para entender melhor um banco de dados que contém textos com resenhas sobre aplicativos da App Store do Android. Para isso, use a função `load` para carregar o banco `dadosReviewGoogle.RData`. Este banco contém dois objetos, `textos`, que contém as diferentes resenhas sobre os aplicativos , e `notas`, que contém as respectivas notas atribuidas pelos usuários que escreveram essas resenhas. Seu objetivo não é o de predição de notas, mas apenas o de melhor entendimento dos reviews.

```
# <code r> ========================================================================= #
path <- "C:/Users/henri/Dropbox/Scripts/aprendizado de maquina/list 4/"

dados1 <- load(paste0(path, "dadosReviewGoogle.RData"))

# write.csv2(as.data.frame(textos), "dados1.csv", row.names = FALSE)

texto <- read.csv2(paste0(path, "dados1.csv"), encoding = "UTF-8")

dados1 <- cbind(texto, nota)

dados1 <- subset(dados1, !is.na(textos) & !is.na(nota))

dados1[ , 1] <- as.character(dados1[ , 1])

dados1[ , 1] <- gsub(x = dados1[ , 1], pattern = "!\t", replacement = "! ")

dados1[ , 1] <- gsub(x = dados1[ , 1], pattern = "\t", replacement = ". ")
# </code r> ======================================================================== #
```

a) Mostre 5 resenhas do banco juntamente com suas respectivas notas.

Notas	Resenha
1	Péssimo, um lixo! Comprei a nova versão por R$2,00, mas me arrependi pois não roda! Desinstalei e baixei novamente e nada! Só o que faz é reiniciar o serviço de telefonia! NÃO CAIAM NESSA!!!
1	Bugs. Nao consigo jogar é uma bosta gastei meu dinhero para nada
1	Nao satisfeito. O app nao roda fica tudo em branco resolvam este problema que parece geral todos reclamam me admira a appstore manter isto ainda para venda é lastimável

Notas	Resenha
5	Galaxy S3. Muito bom o app não tenho oque reclamar...
1	Fix the issues. O jogo é excelente mas esse problema que todos relatam precisa ser resolvido. Acontece a mesma coisa comigo, fica impossível jogar. Não carrega, trava..

Para os itens que seguem, você pode trabalhar com um subconjunto dos dados originais.

```
# <code r> ============================================================== #
sd1 <- dados1[sample(1:nrow(dados1), 1000), ]
# </code r> ============================================================ #
```

b) Use o código fornecido para converter os textos em uma matriz documento-termo binária (isto é, cada entrada da matriz indica se um termo está presente ou não no respectivo texto.

```
# <code r> ============================================================== #
library(tm)

corp <- VCorpus(
  VectorSource(sd1[ , 1]), readerControl = list(language = "portuguese"))

corp <- tm_map(corp, removeWords, stopwords("portuguese"))

library(SnowballC)

corp <- tm_map(corp, stemDocument)

corp <- tm_map(corp, stripWhitespace)

dtm <- DocumentTermMatrix(corp, control = list(
  tolower = TRUE, removeNumbers = TRUE, removePunctuation = TRUE))

dtm <- as.matrix(dtm)
# </code r> ============================================================ #
```

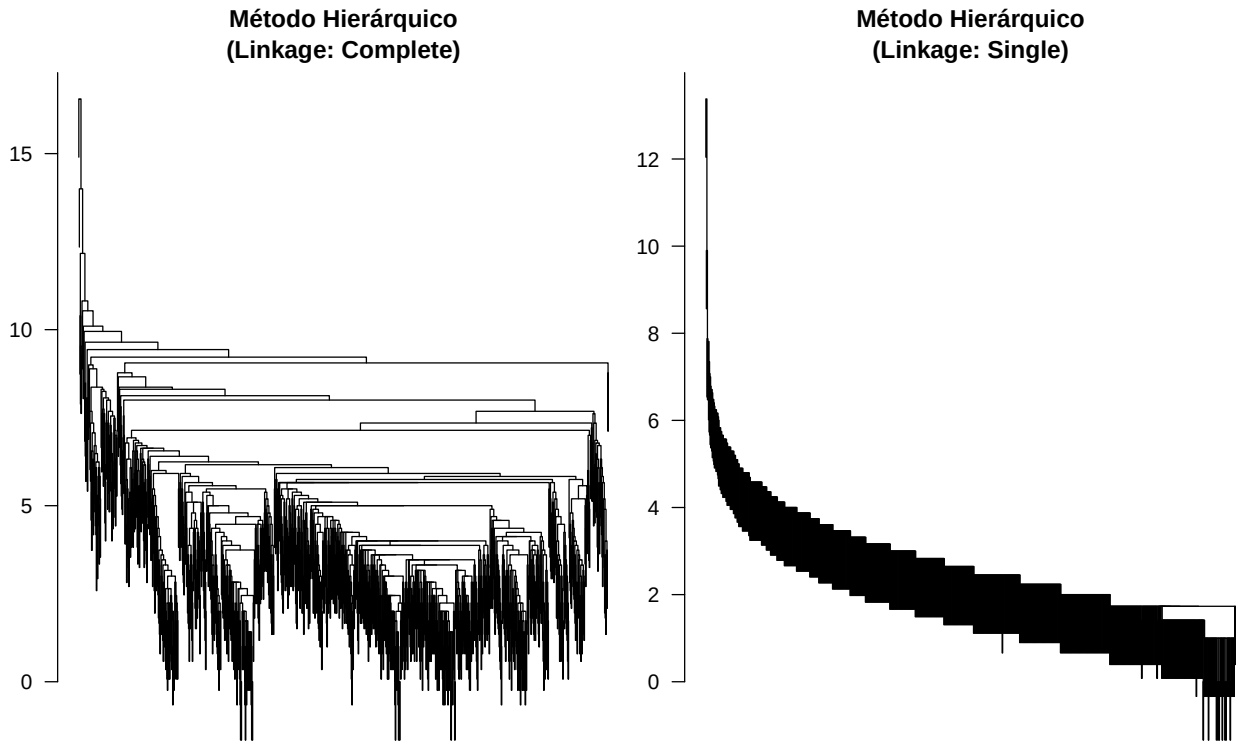
c) Use duas técnicas de clustering para criar agrupamentos dos diferentes textos (não use as notas pra isso). Para a técnica que foi fornecida, faça também duas variações. Interprete os grupos obtidos por cada um dos métodos. Eles concordam entre si?

```r
# <code r> ========================================================================== #
dtm.dist <- dist(dtm)
# </code r> ========================================================================== #
```

Métodos Hierárquicos:

- Complete: Dissimilaridade máxima entre clusters
- Single: Dissimilaridade mínima entre clusters

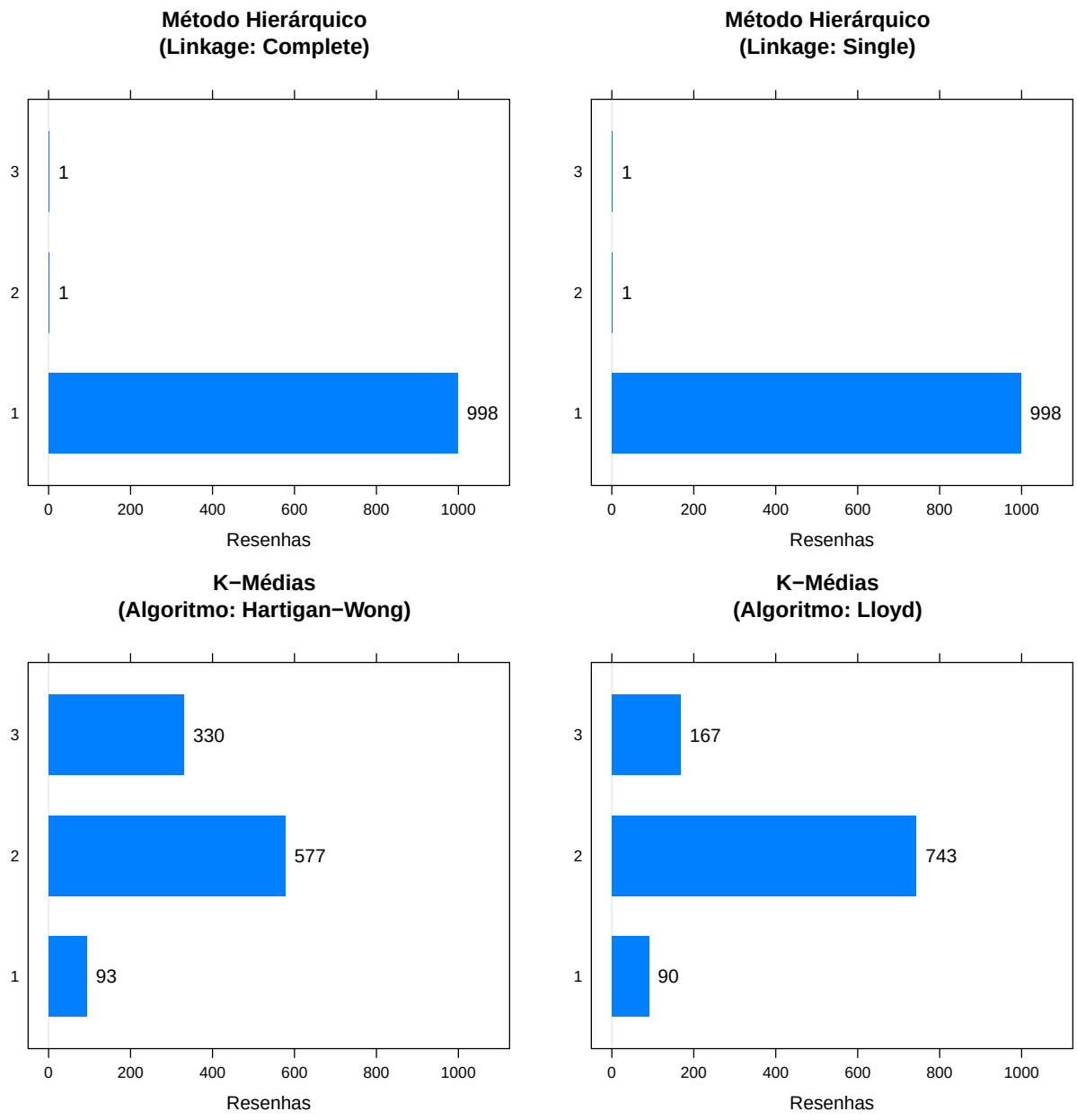
```r
# <code r> ========================================================================== #
clus.comp <- hclust(dtm.dist, method = "complete")

clus.sing <- hclust(dtm.dist, method = "single")

par(mfrow = c(1, 2), mar = c(1, 2, 3, 0) + .1)

plot(clus.comp, main = "Método Hierárquico\n(Linkage: Complete)"
     , labels = FALSE, ylab = NULL, xlab = NA, sub = NA, las = 1)

plot(clus.sing, main = "Método Hierárquico\n(Linkage: Single)"
     , labels = FALSE, ylab = NULL, xlab = NA, sub = NA, las = 1)
# </code r> ========================================================================== #
```



K-Médias:

- Algoritmo de Hartigan-Wong
- Algoritmo de Lloyd

```
# <code r> ========================================================================= #
km.hw <- kmeans(dtm, 3, algorithm = "Hartigan-Wong")

km.l <- kmeans(dtm, 3, algorithm = "Lloyd", iter.max = 15)
# </code r> ========================================================================= #


## Número de resenhas em cada cluster
# <code r> ========================================================================= #
library(latticeExtra)

bar <- function(y, main){
  barchart(as.factor(y)
           , xlim = c(-50, 1125)
           , main = main
           , col = "#0080ff"
           , border = "transparent"
           , xlab = "Resenhas"
           , panel = function(...){
             panel.barchart(...)
             args <- list(...)
             panel.text(args$x, args$y, args$x, pos = 4)})}

print(bar(cutree(clus.comp, 3)
          , main = "Método Hierárquico\n(Linkage: Complete)")
  , position = c(0, .5, .5, 1)
  , more = TRUE)

print(bar(cutree(clus.sing, 3)
          , main = "Método Hierárquico\n(Linkage: Single)")
      , position = c(.5, .5, 1, 1)
      , more = TRUE)

print(bar(km.hw$cluster
          , main = "K-Médias\n(Algoritmo: Hartigan-Wong)")
      , position = c(0, 0, .5, .5)
      , more = TRUE)

print(bar(km.l$cluster
          , main = "K-Médias\n(Algoritmo: Lloyd)")
      , position = c(.5, 0, 1, .5))
# </code r> ========================================================================= #
```



Para utilizar a técnica K-Médias é necessário especificar o número de clusters. Na utilização do Método Hierárquico vimos que com mais de três clusters são formados apenas clusters com uma única resenha.

Independente do método e da variação foi obtido um cluster com um número maciço de resenhas, contudo, no K-Médias, para ambos algoritmos, os demais dois clusters tem um número bem considerável de resenhas.

d) Mostre as 5 regras de associação encontradas (não use as notas pra isso) usando o algoritmo *a priori* com maior suporte, as 5 com maior confiança e as 5 com maior lift. Interprete o valor do suporte, lift e confiança de uma regra de sua escolha. Mostre ao menos 3 maneiras distintas essas regras visualmente.

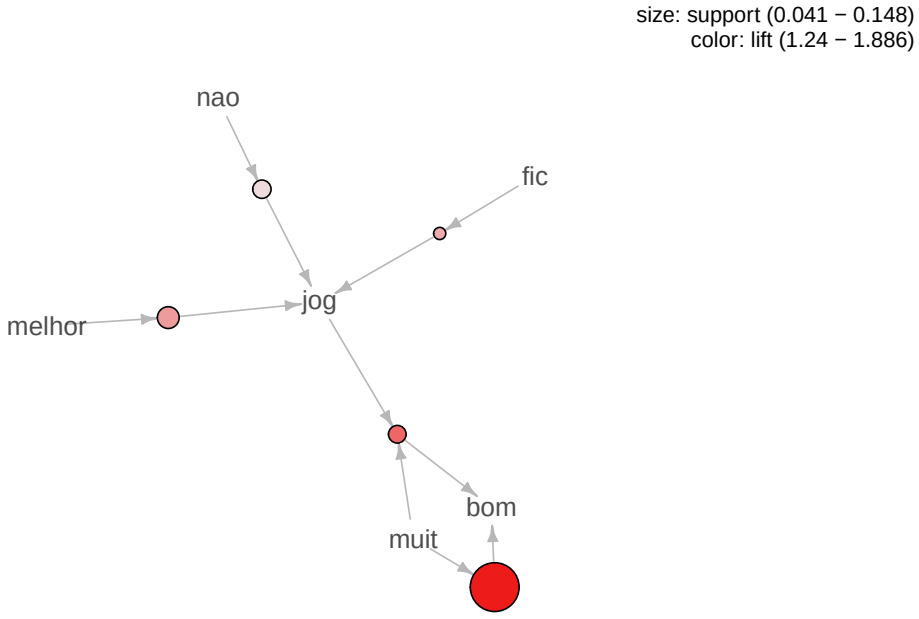
```
# <code r> ============================================================== #
library(arulesViz)

rules <- apriori(dtm, parameter = list(
  support = .005, confidence = .5, maxlen = 3), control = list(
    verbose = FALSE))
# </code r> ============================================================== #
```

Suporte

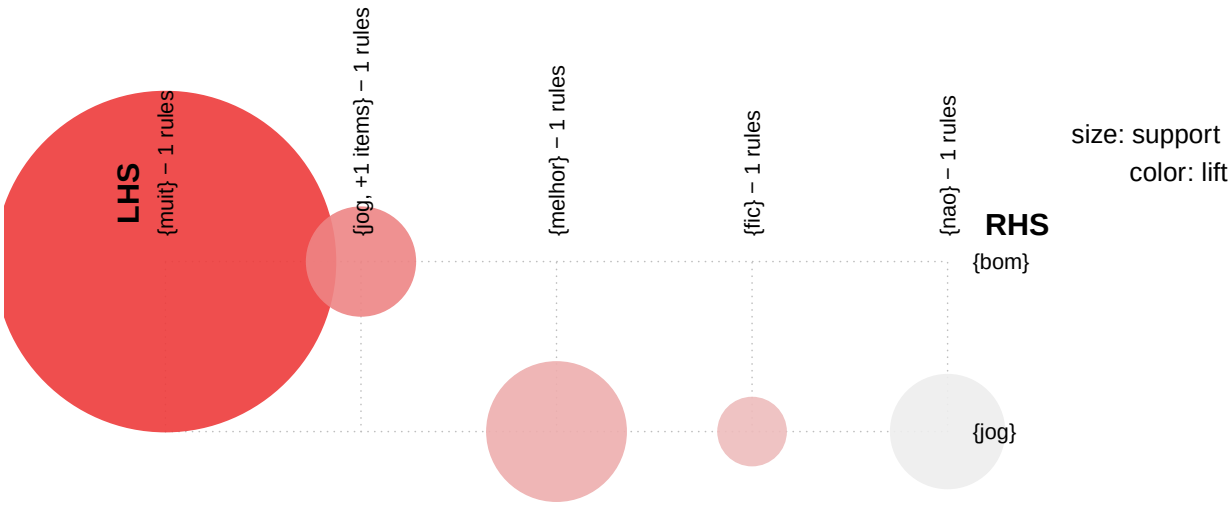
```
# <code r> ============================================================== #
inspect(sort(rules, by = "support", decreasing = TRUE)[1:5])
# </code r> ============================================================== #
```

```
    lhs           rhs    support confidence lift
[1] {muit}     => {bom} 0.148   0.6883721  1.885951
[2] {melhor}   => {jog} 0.069   0.6509434  1.546184
[3] {nao}      => {jog} 0.059   0.5221239  1.240199
[4] {jog,muit} => {bom} 0.057   0.6195652  1.697439
[5] {fic}      => {jog} 0.041   0.6212121  1.475563
```

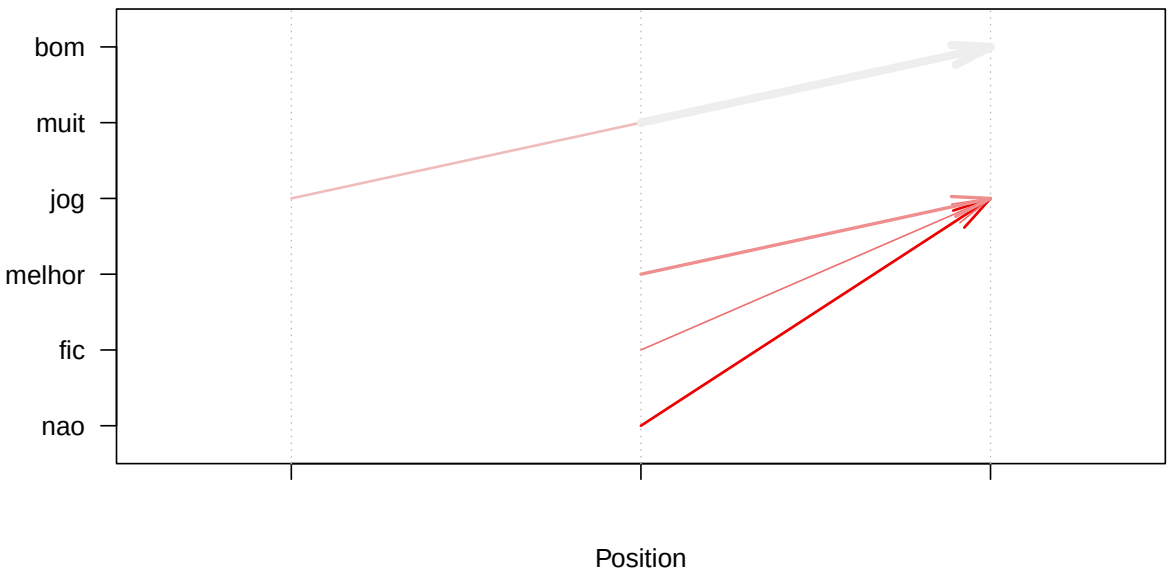
```
# <code r> ============================================================== #
plot(sort(rules, by = "support", decreasing = TRUE)[1:5], method = "graph"
     , control = list(main = NULL, alpha = 1))
# </code r> ============================================================== #
```



```
# <code r> ============================================================== #
plot(sort(rules, by = "support", decreasing = TRUE)[1:5], method = "grouped"
     , control = list(main = NULL))
# </code r> ============================================================== #
```




```
# <code r> ============================================================== #
plot(sort(rules, by = "support", decreasing = TRUE)[1:5], method = "paracoord"
     , control = list(main = NULL))
# </code r> ============================================================== #
```

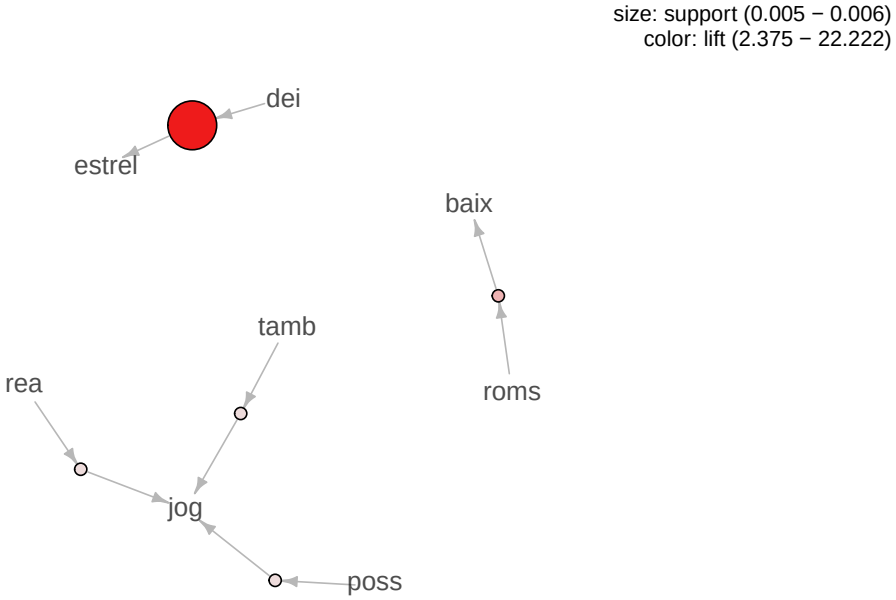


Confiança

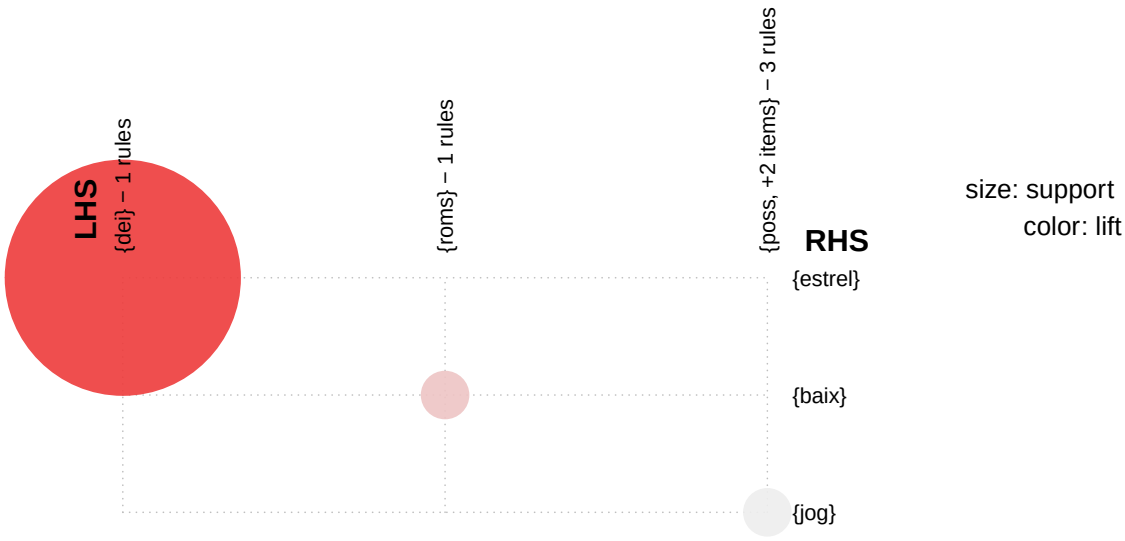
```
# <code r> ================================================================== #
inspect(sort(rules, by = "confidence", decreasing = TRUE)[1:5])
# </code r> ================================================================= #
```

```
    lhs       rhs      support confidence lift
[1] {rea}  => {jog}    0.005   1           2.375297
[2] {poss} => {jog}    0.005   1           2.375297
[3] {roms} => {baix}   0.005   1           8.333333
[4] {tamb} => {jog}    0.005   1           2.375297
[5] {dei}  => {estrel} 0.006   1          22.222222
```

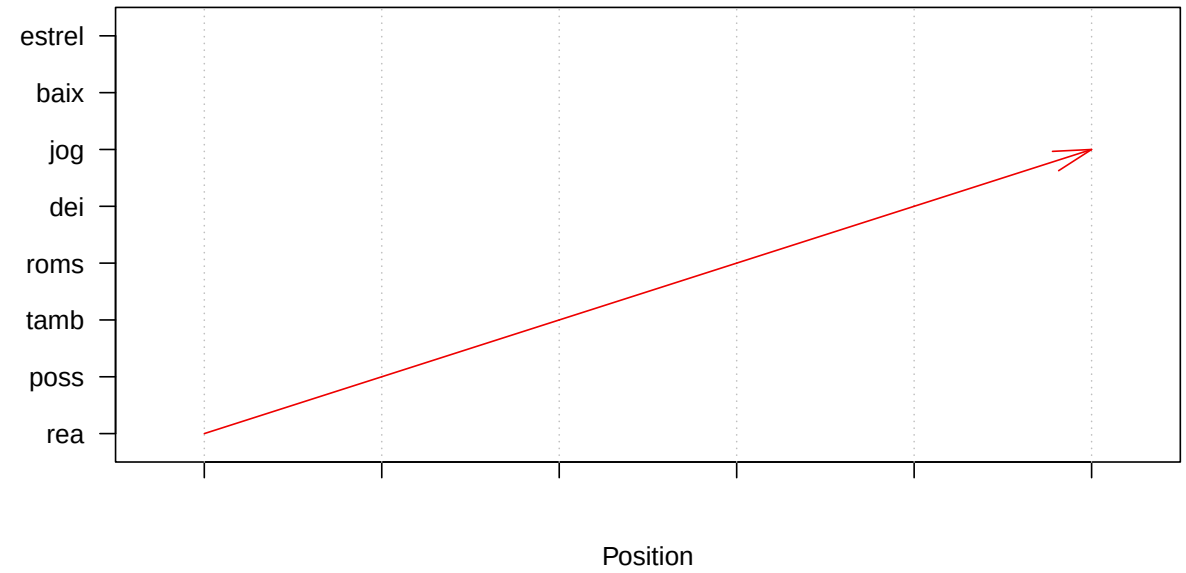
```
# <code r> ================================================================== #
plot(sort(rules, by = "confidence", decreasing = TRUE)[1:5]
     , method = "graph"
     , control = list(main = NULL, alpha = 1))
# </code r> ================================================================= #
```




```
# <code r> ================================================================== #
plot(sort(rules, by = "confidence", decreasing = TRUE)[1:5]
     , method = "grouped"
     , control = list(main = NULL))
# </code r> ================================================================= #
```




```
# <code r> ============================================================== #
plot(sort(rules, by = "confidence", decreasing = TRUE)[1:5]
     , method = "paracoord"
     , control = list(main = NULL))
# </code r> ============================================================== #
```

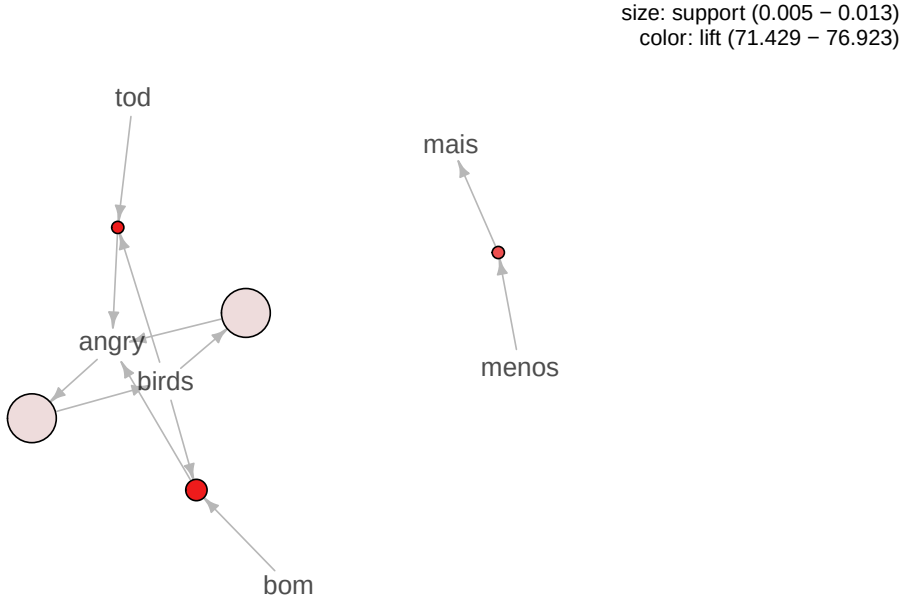


Lift

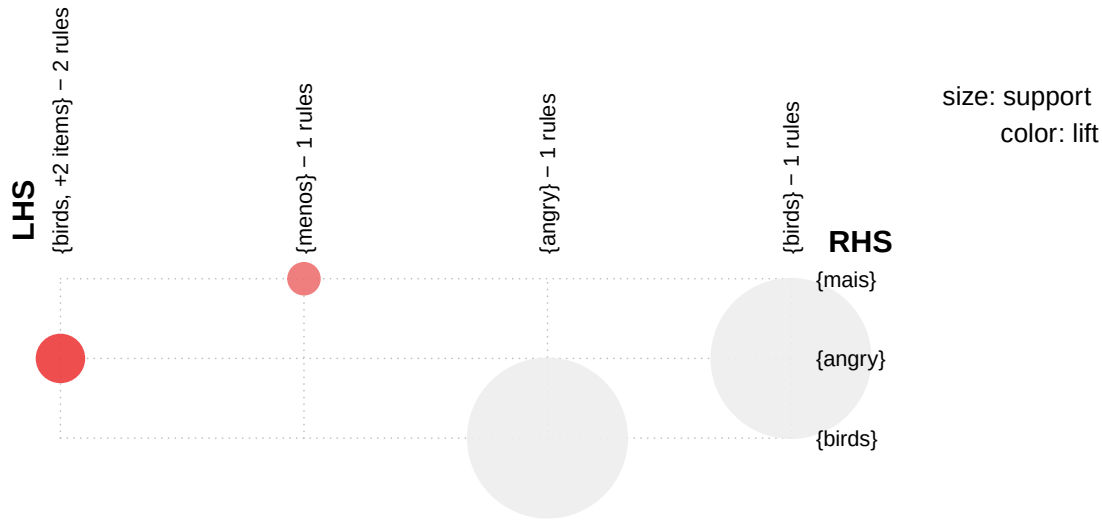
```
# <code r> ========================================================================== #
inspect(sort(rules, by = "lift", decreasing = TRUE)[1:5])
# </code r> ========================================================================= #
```

```
    lhs             rhs     support confidence lift
[1] {birds,tod} => {angry} 0.005   1.0000000  76.92308
[2] {birds,bom} => {angry} 0.007   1.0000000  76.92308
[3] {menos}     => {mais}  0.005   0.8333333  75.75758
[4] {angry}     => {birds} 0.013   1.0000000  71.42857
[5] {birds}     => {angry} 0.013   0.9285714  71.42857
```

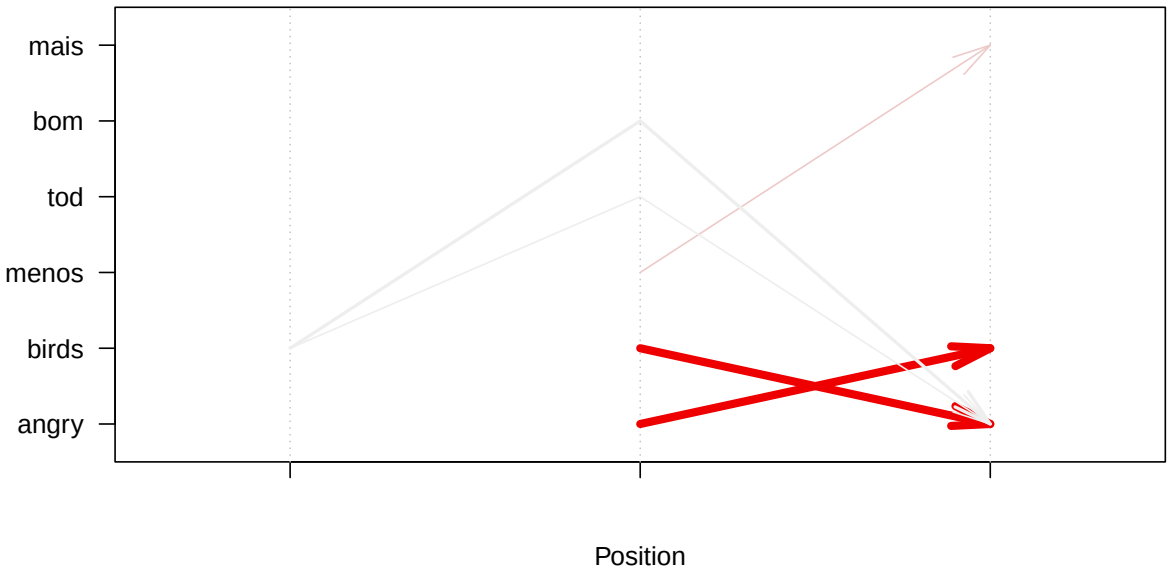
```
# <code r> ========================================================================== #
plot(sort(rules, by = "lift", decreasing = TRUE)[1:5], method = "graph"
     , control = list(main = NULL, alpha = 1))
# </code r> ========================================================================= #
```




```
# <code r> ========================================================================== #
plot(sort(rules, by = "lift", decreasing = TRUE)[1:5], method = "grouped"
     , control = list(main = NULL))
# </code r> ========================================================================= #
```




```
# <code r> ============================================================================ #
plot(sort(rules, by = "lift", decreasing = TRUE)[1:5], method = "paracoord"
     , control = list(main = NULL))
# </code r> ============================================================================ #
```



Regra escolhida: angry => birds

Suporte de 0.013, i.e., probabilidade de 0.013 de que as palavras angry e birds aparecam.

Confiança de 1, i.e., todos os usuários que em suas resenhas utilizaram a palavra angry também usaram a palavra birds.

Lift de 0.714, i.e., dado que o usuário utilizou a palavra angry em sua resenha, a probabilidade dele utilizar a palavra birds aumenta 0.714.

e) Implemente componentes principais para esses dados (não use as notas pra isso). Mostre quais são as 5 variáveis que recebem os maiores coeficientes (cargas) no primeiro componente. Mostre também as 5 variáveis que recebem os menores coeficientes (cargas) no primeiro componente. É possível interpretar essas palavras? Faça o mesmo com o segundo componente. Faça um diagrama de dispersão dos dois primeiros componentes principais. Use uma cor para cada ponto de acordo com a nota atribuida. Há uma relação entre os componentes encontrados e as notas atribuidas? Você consegue encontrar outliers com base nesses gráficos? Mostre ao menos três textos outliers. Repita o procedimento usando os três primeiros componentes, isto é, usando um gráfico em 3d.

```
# <code r> ============================================================== #
pca <- prcomp(dtm)
# </code r> ============================================================= #


## Maiores cargas do primeiro componente
# <code r> ============================================================== #
sort(pca$rotation[ , 1], decreasing = TRUE)[1:5]
# </code r> ============================================================= #

       jog     melhor        nao       baix        fic
0.96340337 0.07838049 0.06380642 0.05858186 0.05602059


## Menores cargas do primeiro componente
# <code r> ============================================================== #
sort(pca$rotation[ , 1])[1:5]
# </code r> ============================================================= #

        muit        demor    recomendo        angry       baixar
-0.010736105 -0.008374956 -0.007447166 -0.006668486 -0.006481232
```

No primeiro componente vemos palavras que são de se esperar que aparecam com grande frequência nas resenhas, mas nenhuma interpretação em relação a utilização dessas palavras é possível.

```r
## Maiores cargas do segundo componente
# <code r> ============================================================== #
sort(pca$rotation[ , 2], decreasing = TRUE)[1:5]
# </code r> ============================================================= #
```

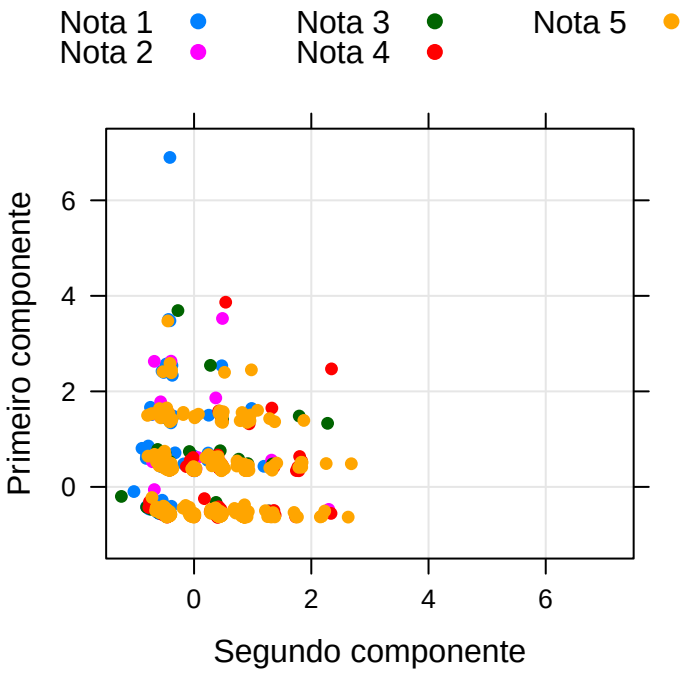
```
       bom       muit        jog       pass      recom
0.87326616 0.42717574 0.03000869 0.02856446 0.02765368
```

```r
## Menores cargas do segundo componente
# <code r> ============================================================== #
sort(pca$rotation[ , 2])[1:5]
# </code r> ============================================================= #
```

```
        nao         fic        baix       legal     atualiz
-0.14399184 -0.03662807 -0.03153271 -0.03018729 -0.02336789
```

O mesmo pode ser dito no segundo componente.

```r
## Diagrama de dispersão dos dois primeiros componentes principais
# <code r> ============================================================== #
xyplot(pca$x[ , 1] ~ pca$x[ , 2], groups = sd1$nota
       , xlim = c(-1.5, 7.5), ylim = c(-1.5, 7.5), type = c("p", "g"), pch = 16
       , xlab = "Segundo componente", ylab = "Primeiro componente"
       , key = list(space = "top", text = list(paste("Nota", 1:5))
                    , points = list(pch = 16, col = trellis.par.get(
                      "superpose.symbol")$col[1:5]), columns = 3))
# </code r> ============================================================= #
```



Nenhuma relação é observada entre as notas e os componentes, mas alguns poucos outliers podem ser vistos.

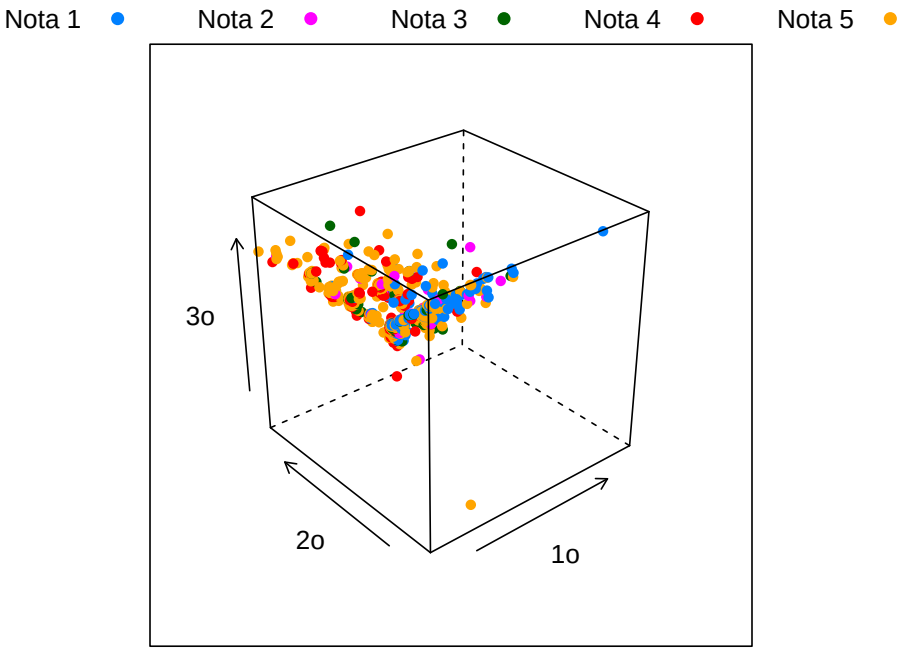
```
## Outliers
# <code r> ============================================================================ #
pca$x[ , 1][pca$x[ , 1] > 3.65]
# </code r> ============================================================================ #
```

```
     215      837      848
3.866555 3.691804 6.897367
```

Número	Nota	Primeiro componente	Segundo componente
215	4	3.8665553	0.5395614
837	3	3.6918041	−0.2755661
848	1	6.8973669	−0.4120806

Número	Resenha
215	Melhorar inteligência artificial!!!! É o jogo que eu mais jogo no celular dentre muitos que tenho! Gosto realmente MUITO dele! Gráficos habilidade, jogabilidade, etc.. Muito legal! Mas a inteligência artificial tem que melhorar! Eu sei que não dá pra ser 100% mas o computador comprar do lixo e descartar a mesma carta eh falha dos desenvolvedores... Além disso ele poderia completar o jogo que acabei de fazer ao invés de montar outro com a sequência certa para o que seria a canastra do jogo que fiz... Enfim, são opiniões de um fã do jogo! Grato
837	De que adianta jogar e não salvar?. O emulador é ótimo, consigo jogar Super Mario normal nele, mas não adianta nada você jogar pelo app gratuito, pra depois ter que pagar, pra continuar de onde parou. Pelo menos isso, acho que poderia ter na versão grátis. Jogar um monte de fases e ter que pagar depois é sacanagem.
848	Problemas. Comprei o jogo mas sempre dá problema, ele baixa o arquivo do jogo e diz que o download foi completo, clico ok para começar o jogo e aparece aquele vídeo de introdução do jogo e após isso o jogo fecha, abro novamente aí o jogo diz que falta baixar alguns mb's do jogo ainda mas quando termina esse novo download o mesmo problema acontece. Já tentei mais de 10 vezes e nada. Meu dispositivo é um Acer Iconia Tab A500

```r
## Diagrama de dispersão dos três primeiros componentes principais
# <code r> ============================================================== #
cloud(pca$x[ , 3] ~ pca$x[ , 1] + pca$x[ , 2], groups = sd1$nota, pch = 16
      , xlab = "1o", ylab = "2o", zlab = "3o"
      ,  key = list(space = "top", text = list(paste("Nota", 1:5))
                    , points = list(pch = 16, col = trellis.par.get(
                      "superpose.symbol")$col[1:5]), columns = 5))
# </code r> ============================================================== #
```



Nenhuma relação é observada entre as notas e os componentes, mas alguns poucos outliers podem ser vistos.

```r
## Outliers
# <code r> ============================================================== #
pca$x[ , 3][pca$x[ , 3] < -1.65]
# </code r> ============================================================== #
```

```
       287        562        633
-10.637222  -1.656143  -2.374130
```

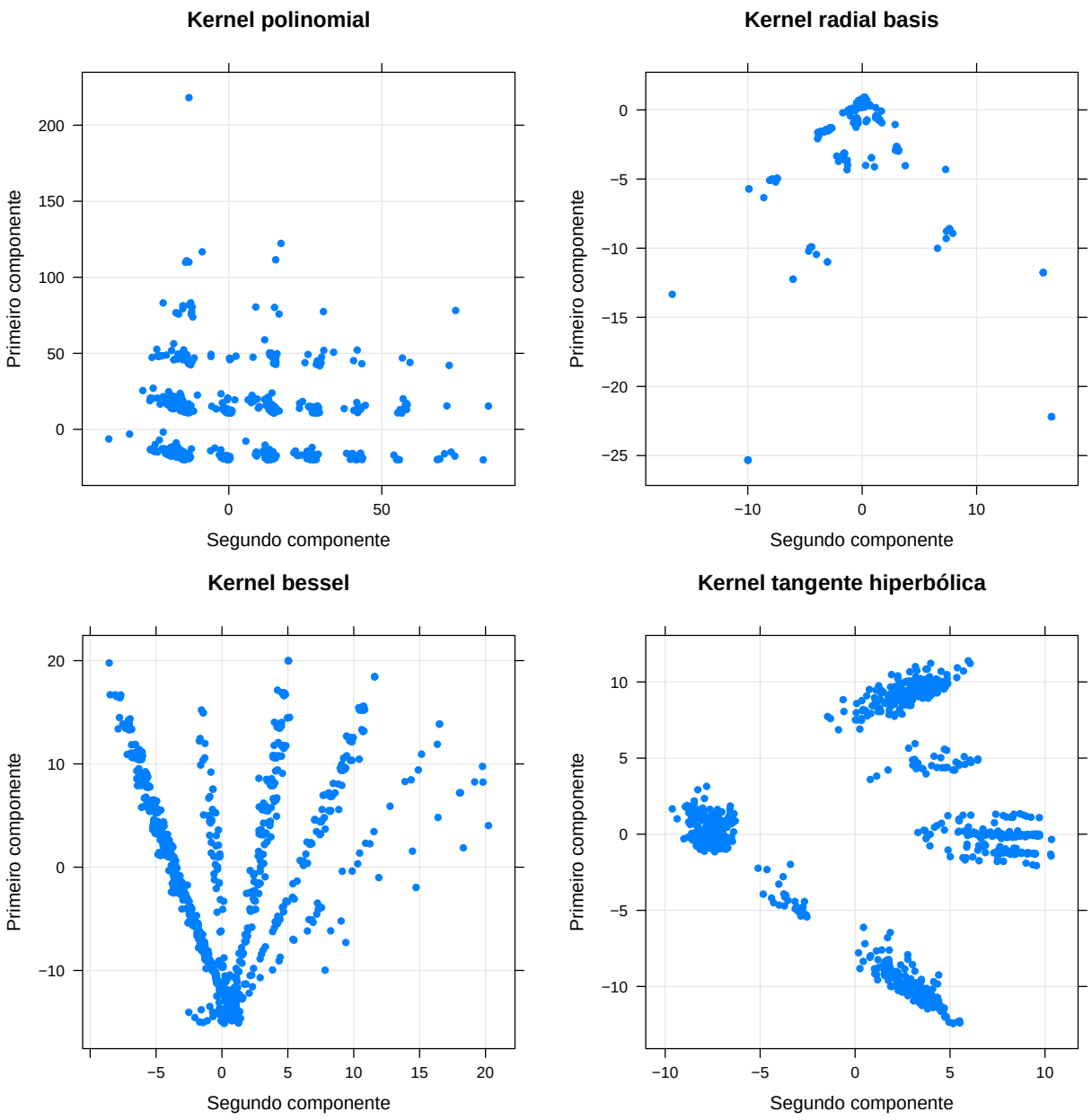
Número	Nota	Primeiro componente	Segundo componente	Terceiro componente
287	5	1.4976688	−0.7937088	−10.6372219
562	4	−0.0562927	−0.6778608	−1.6561426
633	2	−0.5795473	−0.5392238	−2.3741299

Número	Resenha
287	O melhor jogo estilo gta para Android. Vale a pena ser comprado, basicamente ele usa o mesmo conceito que vemos no GTA, só que voce não pega o carro de ninguem, pode explorar o mapa aberto mas usa apenas seus proprios bat vehiculos e tem algumas características de RPG, como o aumento de nível e desbloqueio de habilidades e armas. A gameloft caprichou no jogo, voce poder planar e cair batendo no chão desequilibrando os inimigos, é ótimo e super emocionante, ou pular de um lugar mais algo e cair batendo e deixar um inimigo inconsciente, com direito a camera em slow motion, enfim ficou até bem feito, mas possui algumas coisinhas que deixaram ele chato, as lutas são muito bonitas de se ver, mas não exigem nada do jogador, apesar das missões principais serem até legais, elas não são suficiente para voce aumentar o nivel, logo voce deve recorrer às side quests que são sempre a mesma coisa, massante e repetitiva, e ainda, o batrangue ficou muito poderoso, voce pode somente com ele m...
562	Antes era melhor.... Cara, eu quando compre o meu lg optimus l5ii dual, baixei esse jogo, era fooooda, viciei ate, ai foi atualizando e fico uma merda, fica travando o jogo, ja teve vez do meu boneco ficar ativando o jetpack sem eu clicar na tela, e na hr q começa da um leg triste, ajeita pf cara
633	JOGO QUASE PERFEITO. JOGO BACANA AS ARMADURAS ESTÃO IGUAIS AO FILME. TALVEZ PODE SER UNS DOS MELHORES PARA PODER BAIXAR O GRÁFICO É BEM REALISTA O JOGADOR PODE SE SENTIR O PRÓPRIO HERÓI JONY STARK SALVANDO O MUNDO. JOGO QUASE PERFEITO PODEM BAIXAR PENA QUE DEIXARAM DE SALVAR AS COSTAS DOS USUÁRIOS EM NUVEM PORQUE TEM RANK MUNDIAL TALVEZ A MAIORIA DAS PESSOAS DEVEM TROCAR DE APARELHO EM ALGUM MOMENTO DE SUA VIDA. VOCÊS DEVERIAM SALVAR AS CONTAS EM NUVEM PARA PODER GARANTIR O RANK DE TODOS USUÁRIOS. Rodo perfeito no meu OptimusG

f) Implemente kernel PCA para esses dados, e trabalhe com ao menos duas variações dela. Plote novamente o gráfico de dispersão para essas novas técnicas. Eles são muito diferentes entre si? E com relação a componentes principais? Repita o procedimento usando os três primeiros componentes, isto é, usando um

gráfico em 3d.

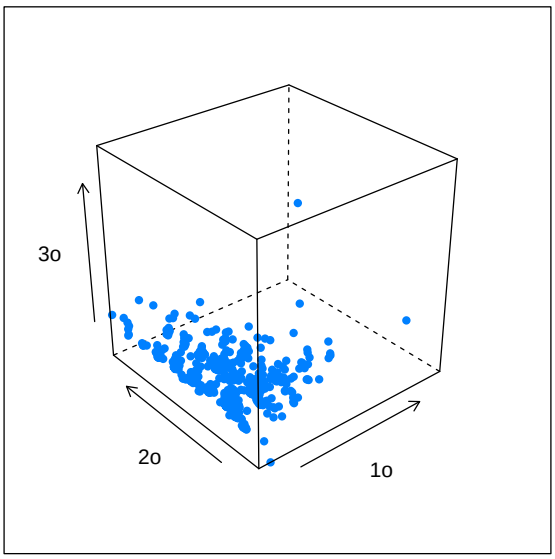
```
# <code r> ========================================================== #
library(kernlab)

## Kernel polinomial
pca2 <- kpca(dtm, kernel = "polydot",kpar = list(degree = 1), features = 3)

## Kernel radial basis, "gaussiano"
pca3 <- kpca(dtm, kernel = "rbfdot", kpar = list(sigma = 1), features = 3)

## Kenrel bessel
pca4 <- kpca(dtm, kernel = "besseldot", kpar = list(degree = 1), features = 3)

## Kernel tangente hiperbólica
pca5 <- kpca(dtm, kernel = "tanhdot", kpar = list(scale = 1), features = 3)
# </code r> ========================================================== #


## Diagramas de dispersão dos dois primeiros componentes principais
# <code r> ========================================================== #
kpc2 <- function(f, main){
  xyplot(f, main = main
         , type = c("p", "g")
         , pch = 16
         , xlab = "Segundo componente"
         , ylab = "Primeiro componente")}

print(kpc2(pca2@rotated[ , 1] ~ pca2@rotated[ , 2]
           , "Kernel polinomial")
      , position = c(0, .5, .5, 1)
      , more = TRUE)

print(kpc2(pca3@rotated[ , 1] ~ pca3@rotated[ , 2]
           , "Kernel radial basis")
      , position = c(.5, .5, 1, 1)
      , more = TRUE)

print(kpc2(pca4@rotated[ , 1] ~ pca4@rotated[ , 2]
           , "Kernel bessel")
      , position = c(0, 0, .5, .5)
      , more = TRUE)

print(kpc2(pca5@rotated[ , 1] ~ pca5@rotated[ , 2]
           , "Kernel tangente hiperbólica")
      , position = c(.5, 0, 1, .5))
# </code r> ========================================================== #
```



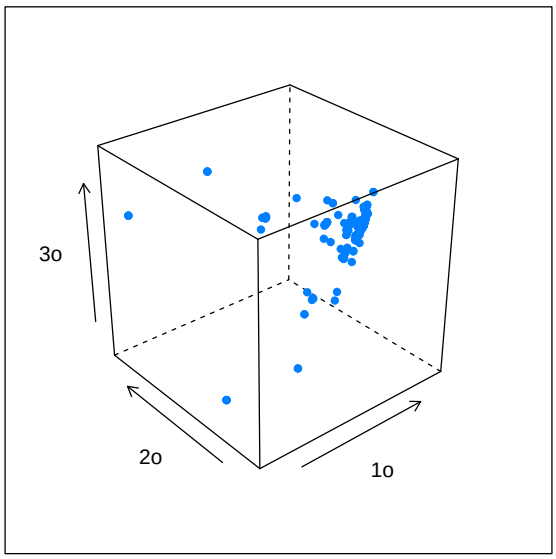

```
## Diagramas de dispersão dos três primeiros componentes principais
# <code r> ========================================================================= #
kpc3 <- function(f, main){
  cloud(f, main = main, pch = 16, xlab = "1o", ylab = "2o", zlab = "3o")}

print(kpc3(pca2@rotated[ , 3] ~ pca2@rotated[ , 1] + pca2@rotated[ , 2]
           , "Kernel polinomial")
      , position = c(0, .5, .5, 1)
      , more = TRUE)

print(kpc3(pca3@rotated[ , 3] ~ pca3@rotated[ , 1] + pca3@rotated[ , 2]
           , "Kernel radial basis")
```

```
       , position = c(.5, .5, 1, 1)
       , more = TRUE)

print(kpc3(pca4@rotated[ , 3] ~ pca4@rotated[ , 1] + pca4@rotated[ , 2]
           , "Kernel bessel")
       , position = c(0, 0, .5, .5)
       , more = TRUE)

print(kpc3(pca5@rotated[ , 3] ~ pca5@rotated[ , 1] + pca5@rotated[ , 2]
           , "Kernel tangente hiperbólica")
       , position = c(.5, 0, 1, .5))
# </code r> ========================================================================== #
```

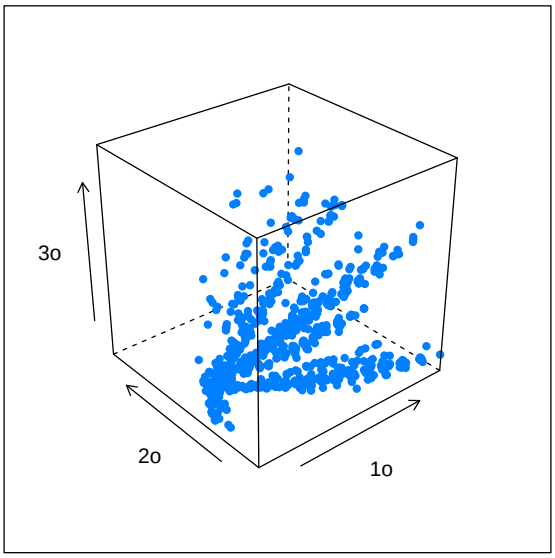
Kernel polinomial



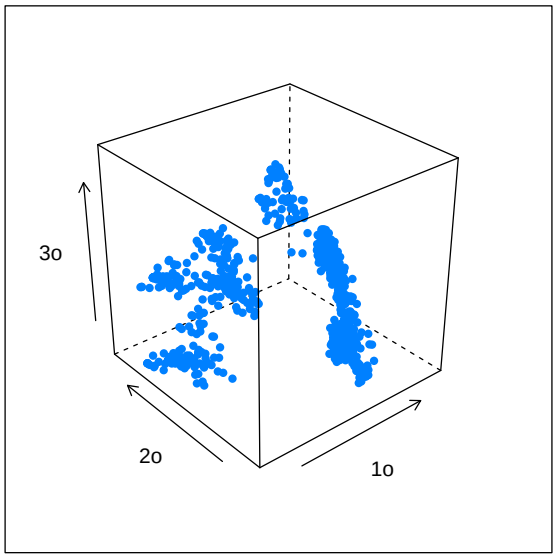
Kernel radial basis



Kernel bessel



Kernel tangente hiperbólica



De uma variação, kernel, para outra os gráficos de dispersão são muito diferentes entre si. Nos kernels bessel e tangente hiperbólica observamos uma maior semelhança nos valores dos componentes principais.

Exercício 2

Baixe o arquivo lista4.R. Ele mostra um código para baixar o banco de dados IncomeESL, que será utilizado neste exercício. Este banco mede diversas covariáveis em indivíduos americanos, como salário, origem e nível de educação. O código fornecido converte este banco para o formato `transactions`, que será usado para implementar as regras de associação vistas em aula. Em particular, o código discretiza as variáveis numéricas.

```
## lista4.R
# <code r> ========================================================================= #
data("IncomeESL")

## Removendo casos com missing
IncomeESL <- IncomeESL[complete.cases(IncomeESL), ]

## Preparando os dados
IncomeESL[["income"]] <- factor((as.numeric(IncomeESL[["income"]]) > 6) + 1
                                , levels = 1:2
                                , labels = c("$0-$40,000", "$40,000+"))

IncomeESL[["age"]] <- factor((as.numeric(IncomeESL[["age"]]) > 3) + 1
                             , levels = 1:2, labels = c("14-34", "35+"))

IncomeESL[["education"]] <- factor(
  (as.numeric(IncomeESL[["education"]]) > 4) + 1
  , levels = 1:2, labels = c("no college graduate", "college graduate"))

IncomeESL[["years in bay area"]] <- factor(
  (as.numeric(IncomeESL[["years in bay area"]]) > 4) + 1
  , levels = 1:2, labels = c("1-9", "10+"))

IncomeESL[["number in household"]] <- factor(
  (as.numeric(IncomeESL[["number in household"]]) > 3) + 1
  , levels = 1:2, labels = c("1", "2+"))

IncomeESL[["number of children"]] <- factor(
  (as.numeric(IncomeESL[["number of children"]]) > 1) + 0
  , levels = 0:1, labels = c("0", "1+"))
```

```
## Criando transactions
Income <- as(IncomeESL, "transactions")
# </code r> ========================================================== #
```

Usando o algoritmo a priori:

Mostre as 10 regras (juntamente com suporte, confiança e lift) com maior lift entre regras com suporte de ao menos 0.001, confiança ao menos 0.8, e tamanho máximo 3.

```
# <code r> ========================================================== #
rules <- apriori(Income, list(support = .001, confidence = .8, maxlen = 3)
                 , control = list(verbose = FALSE))
options(digits = 2)
inspect(sort(rules, by = "lift", decreasing=TRUE)[1:10])
# </code r> ========================================================== #
```

```
     lhs                            rhs                                 support confidence lift
[1]  {marital status=divorced,
      language in home=spanish} => {ethnic classification=hispanic}  0.0042       0.97  7.6
[2]  {occupation=laborer,
      language in home=spanish} => {ethnic classification=hispanic}  0.0128       0.94  7.4
[3]  {occupation=retired,
      language in home=spanish} => {ethnic classification=hispanic}  0.0016       0.92  7.2
[4]  {occupation=unemployed,
      language in home=spanish} => {ethnic classification=hispanic}  0.0031       0.91  7.2
[5]  {number of children=1+,
      language in home=spanish} => {ethnic classification=hispanic}  0.0291       0.90  7.1
[6]  {income=$0-$40,000,
      language in home=spanish} => {ethnic classification=hispanic}  0.0403       0.90  7.1
[7]  {number in household=2+,
      language in home=spanish} => {ethnic classification=hispanic}  0.0278       0.90  7.1
[8]  {type of home=house,
      language in home=spanish} => {ethnic classification=hispanic}  0.0313       0.89  7.0
[9]  {occupation=student,
      language in home=spanish} => {ethnic classification=hispanic}  0.0106       0.89  7.0
[10] {marital status=widowed,
      language in home=spanish} => {ethnic classification=hispanic}  0.0012       0.89  7.0
```

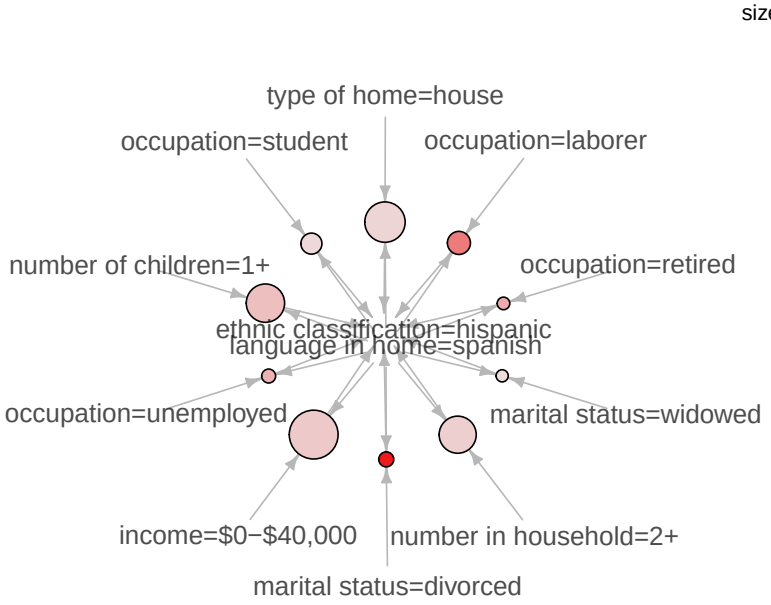
Mostre as 10 regras (juntamente com suporte, confiança e lift) com maior confiança entre regras com suporte de ao menos 0.001, confiança ao menos 0.8, e tamanho máximo 3.

```
# <code r> ========================================================================== #
inspect(sort(rules, by = "confidence", decreasing=TRUE)[1:10])
# </code r> ========================================================================== #
```

```
     lhs                                                  rhs                              support confidence lift
[1]  {marital status=widowed}                          => {dual incomes=not married}       0.0294          1  1.7
[2]  {marital status=divorced}                         => {dual incomes=not married}       0.0979          1  1.7
[3]  {marital status=single}                           => {dual incomes=not married}       0.4091          1  1.7
[4]  {age=14-34,
      ethnic classification=east indian}               => {income=$0-$40,000}              0.0012          1  1.6
[5]  {income=$0-$40,000,
      ethnic classification=east indian}               => {age=14-34}                      0.0012          1  1.7
[6]  {marital status=cohabitation,
      ethnic classification=pacific islander}          => {age=14-34}                      0.0015          1  1.7
[7]  {marital status=cohabitation,
      ethnic classification=pacific islander}          => {language in home=english}       0.0015          1  1.1
[8]  {occupation=laborer,
      ethnic classification=pacific islander}          => {education=no college graduate}  0.0015          1  1.4
[9]  {occupation=clerical/service,
      ethnic classification=pacific islander}          => {language in home=english}       0.0015          1  1.1
[10] {householder status=live with parents/family,
      ethnic classification=pacific islander}          => {education=no college graduate}  0.0041          1  1.4
```

Plote as 10 regras com maior lift entre regras com suporte de ao menos 0.001, confiança ao menos 0.8, e tamanho 3.

```
# <code r> ========================================================================== #
plot(sort(rules, by = "lift", decreasing = TRUE)[1:10], method = "graph"
     , control = list(main = NULL, alpha = 1))
# </code r> ========================================================================== #
```



Mostre todas as regras (juntamente com suporte, confiança e lift) com maior confiança entre regras com suporte de ao menos 0.001, confiança ao menos 0.7, tamanho máximo 3, tamanho mínimo 2 e que tenha 'etnic classification=hispanic' do lado esquerdo da regra.

```
# <code r> ============================================================== #
rules <- apriori(Income, list(
  support = .001, confidence = .7, maxlen = 3, minlen = 2)
  , appearance = list(lhs = "ethnic classification=hispanic", default = "rhs")
  , control = list(verbose = FALSE))
options(digits = 2)
inspect(sort(rules, by = "confidence", decreasing = TRUE))
# </code r> ============================================================== #
```

```
    lhs                                  rhs                                 support confidence lift
[1] {ethnic classification=hispanic} => {education=no college graduate}   0.110       0.86  1.2
[2] {ethnic classification=hispanic} => {income=$0-$40,000}               0.099       0.78  1.3
[3] {ethnic classification=hispanic} => {age=14-34}                       0.090       0.71  1.2
```

Explore os dados você mesmo. Mostre ao menos duas regras de associação que você achou interessante além das já apresentadas na lista. Justifique porque as achou interessantes.

```
# <code r> ============================================================== #
rules <- apriori(Income, list(support = .2, confidence = .8, maxlen = 3)
                 , appearance = list(
                   rhs = "marital status=married", default = "lhs")
                 , control = list(verbose = FALSE))

inspect(sort(rules, by = "lift", decreasing = TRUE))
# </code r> ============================================================== #
```

```
    lhs                           rhs                          support confidence     lift
[1] {dual incomes=yes,
     language in home=english} => {marital status=married} 0.2190227  0.9365672 2.428294
[2] {dual incomes=yes}         => {marital status=married} 0.2370564  0.9357061 2.426061
```

No código acima encontramos as regras de decisão com maior suporte e lift que levam ao estado civil casado. Elas são: Renda dupla e língua inglesa utilizada em casa, e renda dupla. Essas duas regras apresentam uma confiança bem grande, 0.94, ou seja, indivíduos com renda dupla e que falam inglês em casa tem alta probabilidade de serem casados.

Exercício 3

Neste exercício você irá explorar alguns sistemas de recomendação para o MovieLens Dataset. Para anto, instale a biblioteca `recommenderlab`, e carregue os dados usando `data(MovieLense)`.

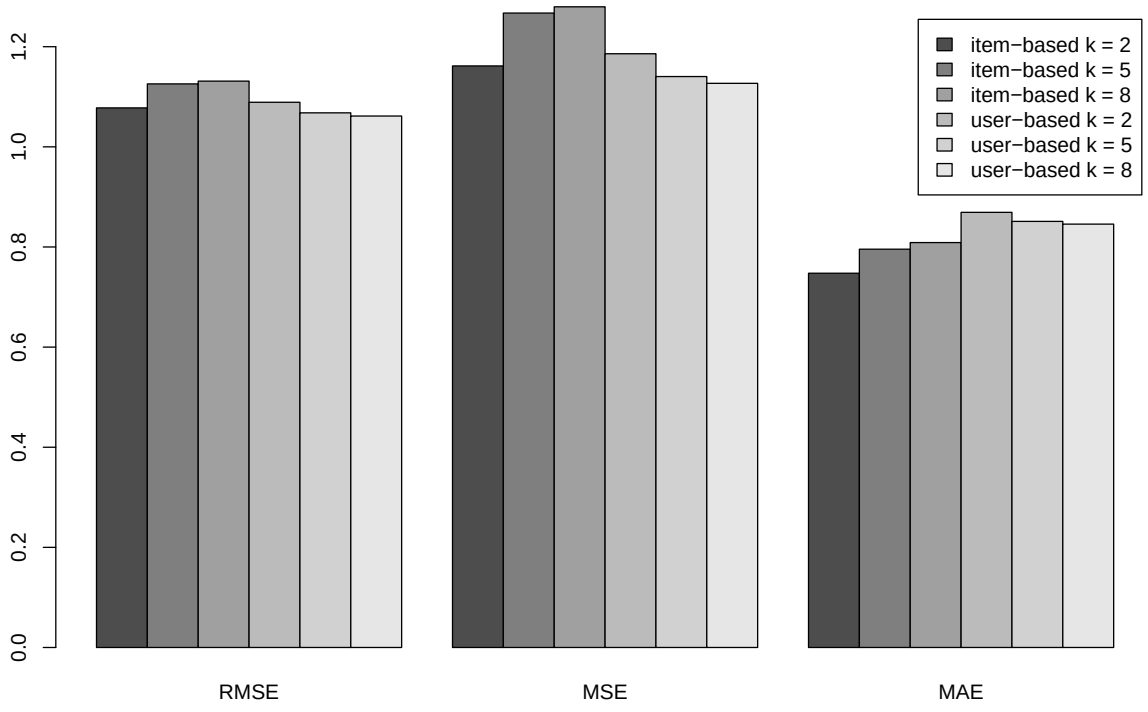
```
# <code r> ============================================================== #
library(recommenderlab)

data("MovieLense")
# </code r> ============================================================= #
```

a) Usando 75% dos dados para treinamento e assumindo que são dadas 12 avaliações por usuário, compare a perfomance dos seguintes métodos com relação a quão boas as predições das notas são:

- **Filtro colaborativo com base nos produtos com $k = 2$**
- **Filtro colaborativo com base nos produtos com $k = 5$**
- **Filtro colaborativo com base nos produtos com $k = 8$**
- **Filtro colaborativo com base nos usuários com $k = 2$**
- **Filtro colaborativo com base nos usuários com $k = 5$**
- **Filtro colaborativo com base nos usuários com $k = 8$**

Você deve estimar o EQM (MSE em inglês), o REQM (RMSE em inglês) e o MAE.

```
# <code r> ============================================================== #
plot(
  evaluate(
    evaluationScheme(
      MovieLense
      , method = "split"
      , train = .75
      , given = 12)
    , list(
      "item-based k = 2" = list(name = "IBCF", param = list(k = 2))
      , "item-based k = 5" = list(name = "IBCF", param = list(k = 5))
      , "item-based k = 8" = list(name = "IBCF", param = list(k = 8))
      , "user-based k = 2" = list(name = "UBCF", param = list(nn = 2))
      , "user-based k = 5" = list(name = "UBCF", param = list(nn = 5))
      , "user-based k = 8" = list(name = "UBCF", param = list(nn = 8)))
    , type = "ratings"
  )
)
# </code r> ============================================================= #
```

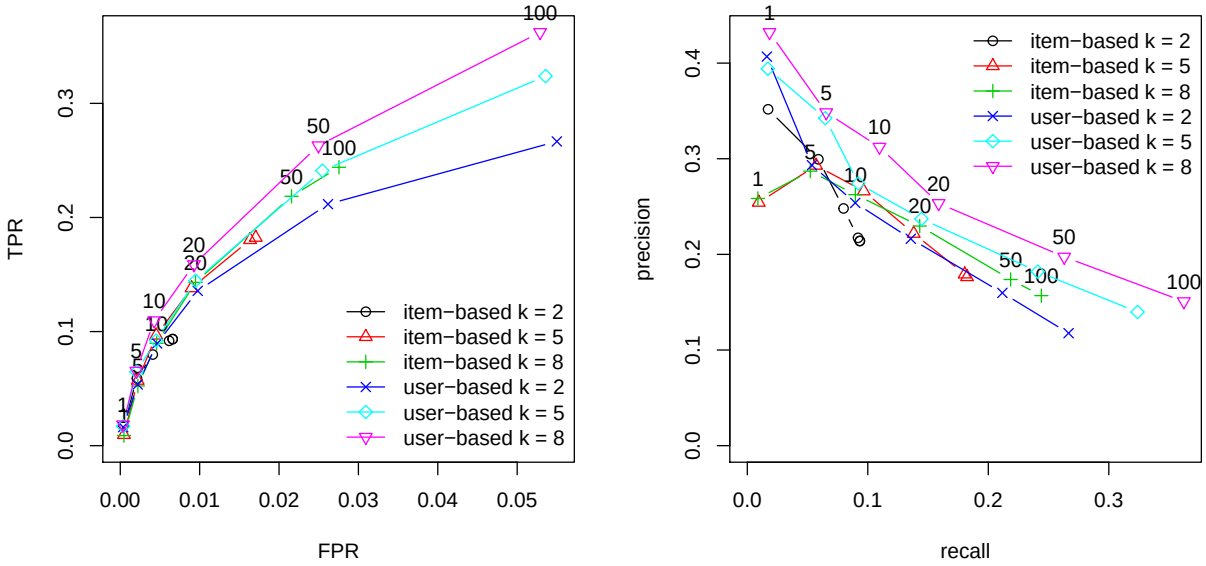


No filtro colaborativo com base nos produtos melhores predições são obtidas com $k = 2$, já no filtro colaborativo com base nos usuários melhores predições são obtidas com $k = 8$, em que k é o número de produtos ou usuários mais parecidos.

b) Compare os mesmos métodos que o descrito no item anterior, mas desta vez usando os métodos de avaliação com base nas N melhores recomendações. Você deve considerar uma avaliação como sendo boa quando sua nota é maior ou igual a 4. Você deve estimar a sensibilidade, 1 - especificidade, precisão e lembrança (recall) para $N = 1, 5, 10, 20, 50$ e 100 recomendações.

```
# <code r> ============================================================== #
results <- evaluate(
  evaluationScheme(
    MovieLense, method = "split", train = .75, given = 12, goodRating = 4)
  , list("item-based k = 2" = list(name = "IBCF", param = list(k = 2))
         , "item-based k = 5" = list(name = "IBCF", param = list(k = 5))
         , "item-based k = 8" = list(name = "IBCF", param = list(k = 8))
         , "user-based k = 2" = list(name = "UBCF", param = list(nn = 2))
         , "user-based k = 5" = list(name = "UBCF", param = list(nn = 5))
         , "user-based k = 8" = list(name = "UBCF", param = list(nn = 8)))
  , n = c(1, 5, 10, 20, 50, 100))
```

```
par(mfrow = c(1, 2), mar = c(5, 4, 2, 2) + .1)
plot(results, annotate = c(3, 6))
plot(results, "prec/rec", annotate = c(3, 6), legend = "topright") ; layout(1)
# </code r> ========================================================================== #
```



No gráfico da esquerda temos 1 - especificidade na abscissa e a sensibilidade na ordenada. No gráfico da direita temos a lembrança na abscissa e a precisão na ordenada.

No filtro coloborativo com base nos produtos melhores resultados foram obtidos com $k = 8$. No filtro colaborativo com base nos usuários melhores resultados foram obtidos também com $k = 8$.

Com relação a 1 - especificidade e sensibilidade, quanto maior o número de recomendações, N, melhores são os resultados. De modo geral, com relação a precisão e a lembrança melhores resultados são obtidos com menores números de recomendações.

c) Alguns dos métodos foi uniformemente melhor que os outros? Justifique.

Quando olhamos para medidas como RMSE, MSE e MAE, ficamos com a impressão de que os filtros coloborativos com base nos usuários são menos influenciados pelo valor de k, obtendo melhores resultados nas métricas RMSE e MSE. Quando olhamos para as medidas de sensibilidade, 1 - espeficidade, lembrança e precisão, os filtros coloborativos com base nos produtos apresentam resultados mais semelhantes, independente de k. Para k's maiores os resultados dos filtros colaborativos com base no usuários se destacam.

Nenhum dos métodos foi uniformemente melhor que os outros.
