EP2

Lucas da Silva Kairoff

7/11/2021

```
#################################################################
## AO PREENCHER ESSE CABEÇALHO COM O MEU NOME E O MEU NÚMERO USP,
## DECLARO QUE SOU O ÚNICO AUTOR E RESPONSÁVEL POR ESSE PROGRAMA.
## TODAS AS PARTES ORIGINAIS DESSE EXERCÍCIO PROGRAMA (EP) FORAM
## DESENVOLVIDAS E IMPLEMENTADAS POR MIM SEGUINDO AS INSTRUÇÕES
## DESSE EP E QUE PORTANTO NÃO CONSTITUEM DESONESTIDADE
## ACADÊMICA OU PLÁGIO.
## DECLARO TAMBÉM QUE SOU RESPONSÁVEL POR TODAS AS CÓPIAS
## DESSE PROGRAMA E QUE EU NÃO DISTRIBUI OU FACILITEI A
## SUA DISTRIBUIÇÃO. ESTOU CIENTE QUE OS CASOS DE PLÁGIO E
## DESONESTIDADE ACADÊMICA SERÃO TRATADOS SEGUNDO OS CRITÉRIOS
## DIVULGADOS NA PÁGINA DA DISCIPLINA.
## ENTENDO QUE EPS SEM ASSINATURA NÃO SERÃO CORRIGIDOS E,
## AINDA ASSIM, PODERÃO SER PUNIDOS POR DESONESTIDADE ACADÊMICA.

## Nome : Lucas da Silva Kairoff
## NUSP : 11373518
## Turma: 22
## Prof.: Roberto Hirata Jr.

## Referências:
### Slides do professor
#################################################################
```

Exercício 1

Leitura do arquivo utilizando a função read.table e especificando o nome de cada coluna.

```
df <- read.table('ml-100k/u.data',
                 col.names = c('user_id',
                               'item_id',
                               'rating',
                               'timestamp'))
```

Exercício 2

Criação e preenchimento da matriz de acordo com as colunas `user_id` e `item_id` dos dados. O preenchimento foi feito utilizando o conceito de índices de dataframes e de matrizes.

```
userid <- df$user_id
itemid <- df$item_id
rating <- df$rating
```

```
matriz <- matrix(nrow = max(userid), ncol = max(itemid))

for(index in 1:nrow(df)){
  matriz[df[index,1], df[index,2]] <- df[index,3]
}
```

Exercício 3

Função `contaLinha` utilizando da função `which`. Essa função retorna índices vetorizados seguindo uma certa condição. No caso desse exercício, retorna os índices de cada linha da matriz em que os valores não são NA, ou seja, que foram avaliados.

```
contaLinha <- function(m){
  q_rating_user <- c()
  for (i in 1:nrow(m)){
    q_rating_user <- c(q_rating_user, length(which(!is.na(m[i, ]))))
  }
  q_rating_user
}
```

Exercício 4

Segue a mesma lógica da função anterior, mas com as colunas.

```
contaColuna <- function(m){
  q_rating_movie <- c()
  for (i in 1:ncol(m)){
    q_rating_movie <- c(q_rating_movie, length(which(!is.na(m[, i]))))
  }
  q_rating_movie
}
```

Exercício 5

Nesse exercício, para não utilizar de funções estatística prontas, como a função `mean`, foi feita uma função que calcula a média já desconsiderando os valores NA e depois foi aplicada na função `mediaColuna`. Ambas foram feitas com conceitos vistos no EP1 e nos itens anteriores.

```
# Criação de uma função que calcula a média sem NA
mean_without_na <- function(vector){
  sum_elements <- 0
  len_new_vector <- 0

  for(element in vector){
    if (!is.na(element)){
      len_new_vector <- len_new_vector + 1    # tamanho do novo vetor sem NA
      sum_elements <- sum_elements + element
    }
  }
  mean_vector <- sum_elements / len_new_vector
  mean_vector
}
```

```r
## Função  que calcula média da coluna de uma matriz
mediaColuna <- function(m){
  mean_m <- c()
  for (index in 1:ncol(m)){
    mean_m <- c(mean_m, mean_without_na(m[, index]))
  }
  mean_m
}
```

Exercício 6

Aplicação da função `mediaColuna`.

```r
mediaFilmes <- mediaColuna(matriz)
```

Exercício 7

Leitura do arquivo com o nome dos filmes e os *id's*. Foi feita uma máscara com valores booleanos da condição de que os filmes tenham uma média maior que 4.3 e aplicada ao vetor com o nome dos filmes. Isso é possível visto que os índices são iguais.

```r
u_item <- read.csv('ml-100k/u.item', sep = '|', header = FALSE,
                   col.names = c('movie_id', 'movie_title', 3:24))

data <- u_item[, c('movie_id', 'movie_title')]

mascara <- mediaFilmes >= 4.3
all_movies <- data$movie_title
nomeFilmes <- all_movies[mascara]
```

Exercício 8

```r
contaUsers <- contaLinha(matriz)
```

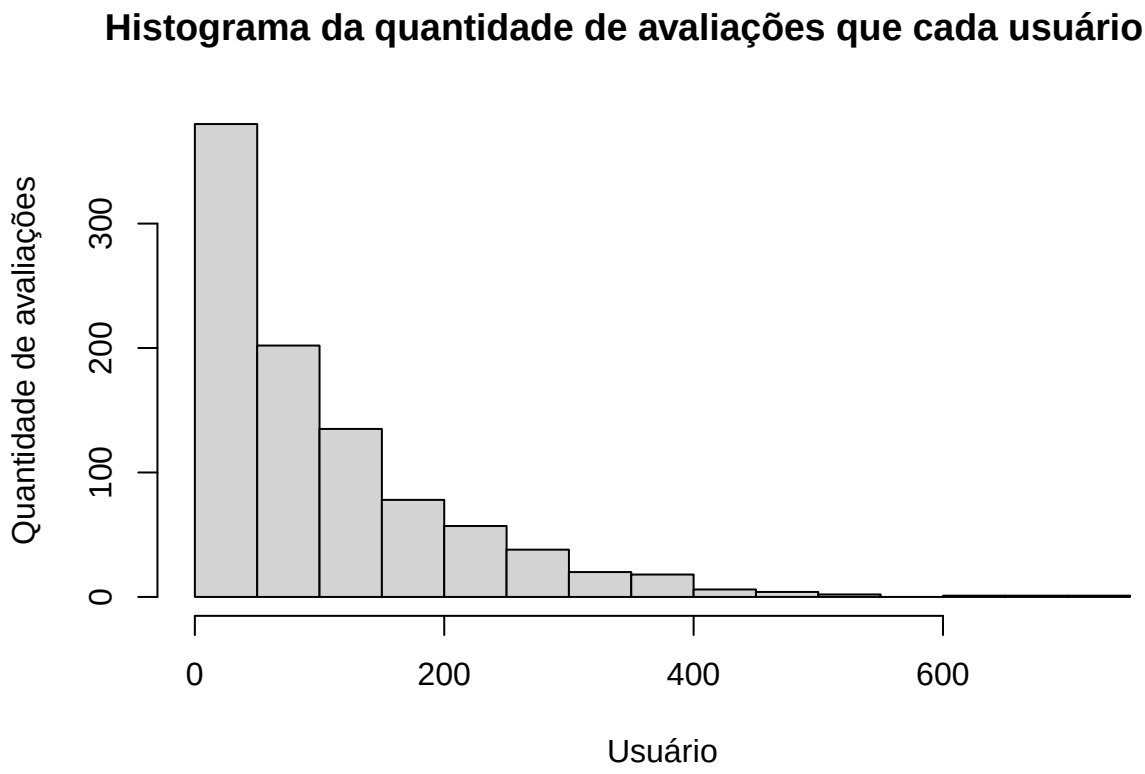
Exercício 9

```r
contaFilmes <- contaColuna(matriz)
```

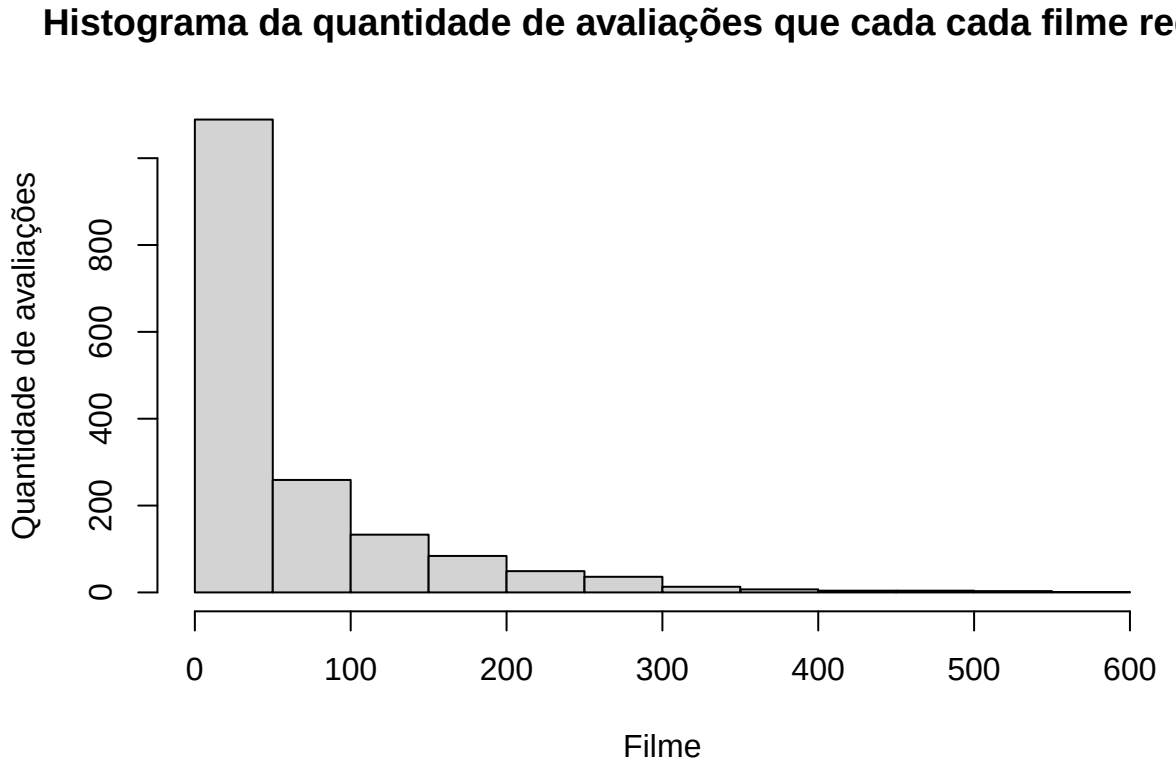
Exercício 10

Visualização gráfica.

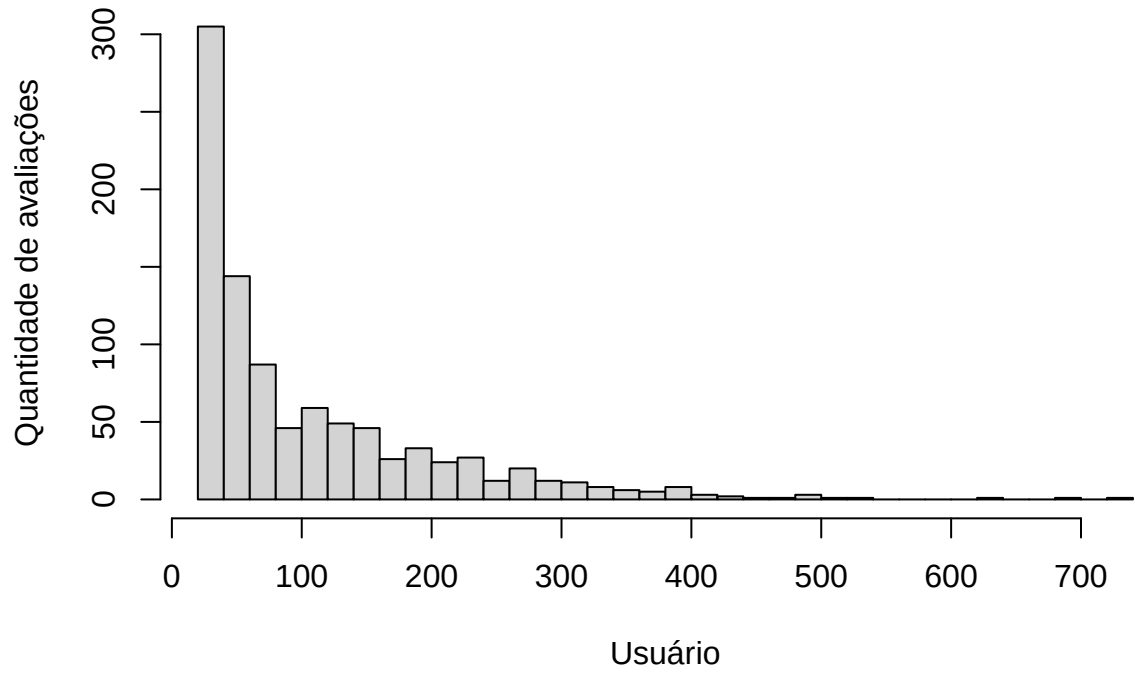
```r
hist(contaUsers,
     main = 'Histograma da quantidade de avaliações que cada usuário deu',
     xlab = 'Usuário',
     ylab = 'Quantidade de avaliações')
```




```
hist(contaFilmes,
    main = 'Histograma da quantidade de avaliações que cada cada filme recebeu',
    xlab = 'Filme',
    ylab = 'Quantidade de avaliações')
```

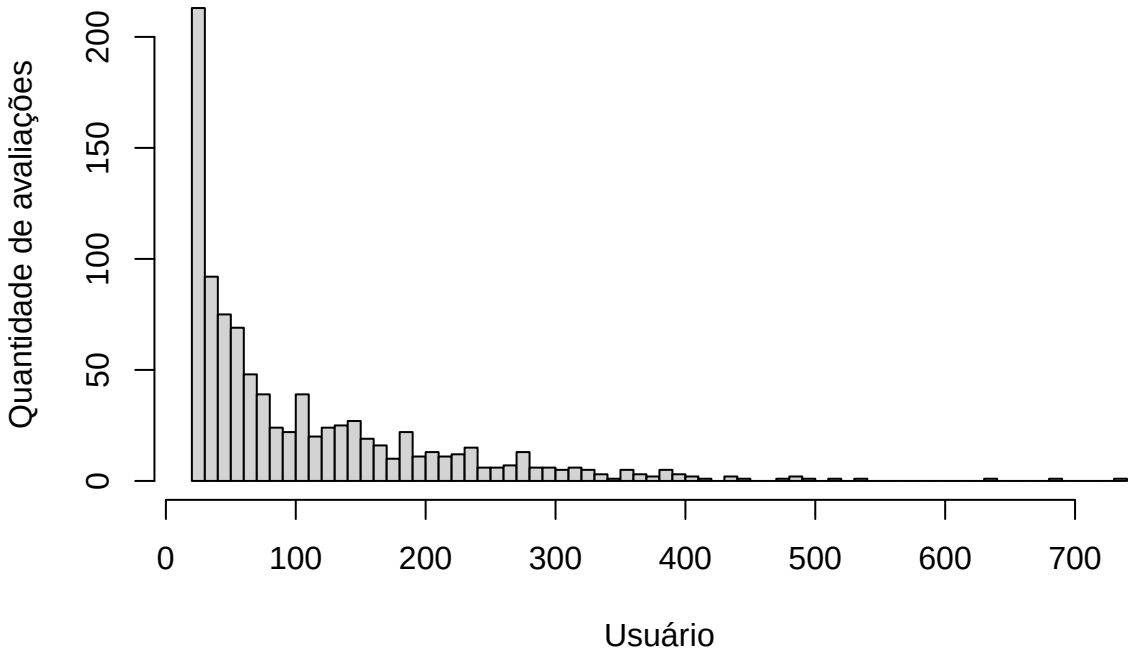


```r
## Experimentando outros resultados
hist(contaUsers, breaks = 50,
     main = 'Histograma da quantidade de avaliações que cada usuário deu (breaks = 50)',
     xlab = 'Usuário',
     ylab = 'Quantidade de avaliações')
```



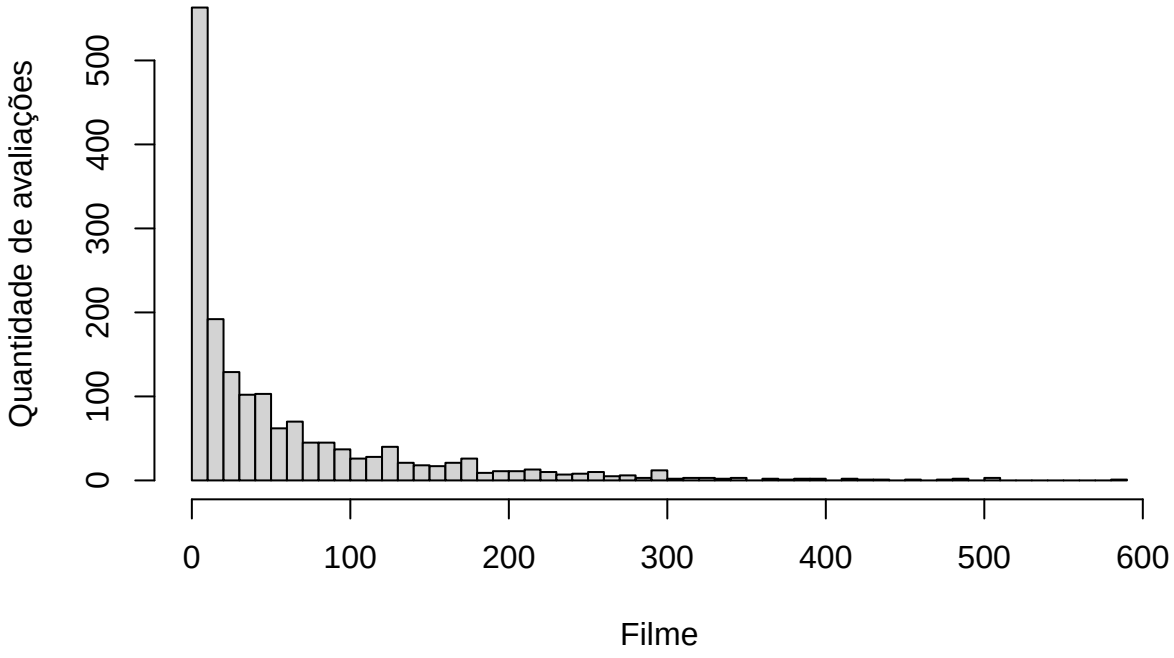

```r
hist(contaUsers, breaks = 100,
     main = 'Histograma da quantidade de avaliações que cada usuário deu (breaks = 100',
     xlab = 'Usuário',
     ylab = 'Quantidade de avaliações')
```

listograma da quantidade de avaliações que cada usuário deu (breaks




```
hist(contaFilmes, breaks = 50,
     main = 'Histograma da quantidade de avaliações que cada cada filme recebeu (breaks = 50)',
     xlab = 'Filme',
     ylab = 'Quantidade de avaliações')
```

ograma da quantidade de avaliações que cada cada filme recebeu (brea



```r
hist(contaFilmes, breaks = 100,
     main = 'Histograma da quantidade de avaliações que cada cada filme recebeu (breaks = 100)',
     xlab = 'Filme',
     ylab = 'Quantidade de avaliações')
```