


## Visualização básica de dados tabulares com R (/pt/licoes/visualizacao-basica-dados-tabulares-r)

Diana Santos  (<https://orcid.org/0000-0002-3108-7706>)

Nesta lição são apresentadas algumas formas básicas de interagir com o R, detalhando o conceito de folha de registo (em inglês, “dataframe”), sua inspeção, manipulação e modificação. Também é dada uma introdução a visualizações simples, como os gráficos de barras e de caixa.

 Avaliada por pares (<https://github.com/programminghistorian/ph-submissions/issues/624>)


 CC-BY 4.0 (<https://creativecommons.org/licenses/by/4.0/deed.en>)

 Apoie o PH (</pt/apoie-nos#doacoes>)

### editado por

- Eric Brasil  (<https://orcid.org/0000-0001-5067-8475>)

### revisto por

- Jimmy Medeiros  (<https://orcid.org/0000-0002-8280-3338>)
- Tarssio Barreto

publicado


| 2025-04-02

modificado

| 2025-03-28

dificuldade

| Baixo

 <https://doi.org/10.46430/phpt0053>

## Conteúdos

- [Requisitos](#)
- [Objetivos da lição](#)
- [Folha de registo](#)
  - [Criação de raiz](#)
    - [Adição de colunas](#)
    - [Adição de linhas](#)
  - [Leitura de fora do R](#)
  - [Processamento de colunas de uma folha de registo](#)
  - [Tipos de colunas](#)
- [Gráficos de barras](#)
- [Gráficos de caixa](#)
- [Mais operações sobre folhas de registo](#)
  - [Obter subconjuntos](#)
  - [Juntar mais do que uma folha de registo numa só](#)
  - [Guardar folhas de registo](#)
- [Valores que faltam](#)
- [Observações finais](#)
- [Notas de fim](#)

## Requisitos

Nesta lição consideramos que o leitor já possui algum conhecimento da linguagem R. Se ainda não completou a lição [Noções básicas de R com dados tabulares \(/pt/licoes/nocoes-basicas-R-dados-tabulares\)](#), recomendamos que o faça primeiro.

## Objetivos da lição

Esta lição pretende apresentar a forma como dados tabulares podem ser visualizados em R, explicando o conceito de folha de registo (“dataframe” em inglês) e os tipos de visualização que podem ser obtidos a partir dela.

## Folha de registo

Quem não teve, ao longo da sua vida, de preencher uma folha de registo numa aula ou para entrar num edifício, com informação como o seu nome, telefone, correio eletrónico e, por vezes, com a assinatura e/ou outras informações (data de entrada, hora de entrada, hora de saída, etc.)?

Se generalizarmos este conceito verificamos que, para cada ocorrência (neste caso, para cada pessoa que entra no edifício), preenche-se um conjunto de dados de vários tipos, onde cada coluna tem o mesmo tipo de informação (nome, número de telefone, data, etc.).

Esta estrutura de dados (se usarmos o vocabulário das linguagens de programação) é muito útil para juntar vários tipos de informação sobre uma mesma entidade (linha). Em R chamamos-lhe “dataframe”, que aqui traduzimos por “folha de registo”.

Uma folha de registo é representada por uma tabela. Cada coluna tem o mesmo tipo de dados. Mas as colunas podem ter informação diferente entre si.

Existem muitos dados em R que são organizados em folhas de registo. Por esse motivo, existem várias funções que se aplicam a folhas de registo de forma a facilitar

a sua manipulação.

## Criação de raiz

Para efeitos pedagógicos, vamos criar algumas folhas de registo de raiz, mas devemos referir que, na maioria dos casos, estes dados vêm de fora e são lidos para o R através das suas funções de entrada/saída, em particular `read.table()`, de que falaremos mais à frente.

```
escritores <-  
data.frame(id=c("JulDin", "CamCBra", "MacAss", "CoeNet"),  
           nome=c("Júlio Dinis", "Camilo Castelo Branco", "Machado de  
Assis", "Coelho Neto"), nascimento=c(1839, 1825, 1839, 1864),  
           morte=c(1871, 1890, 1908, 1934), nacionalidade=c("PT", "PT", "BR", "BR"))
```

O comando `data.frame()` cria uma folha de registo. Por sua vez, `c()` — concatenar — cria um vetor com os argumentos.

Para visualizar a folha de registo criada no R basta escrever o seu nome (será mostrada se não for grande demais). Na Figura 1 temos o resultado:

```
> escritores  
   id      nome nascimento morte nacionalidade  
1 JulDin  Júlio Dinis    1839  1871          PT  
2 CamCBra Camilo Castelo Branco 1825  1890          PT  
3 MacAss   Machado de Assis    1839  1908          BR  
4 CoeNet   Coelho Neto      1864  1934          BR  
> |
```

Figura 1. Resultado da inspeção da folha de registo criada.

Nesta é possível identificar cada coluna pelo nome. Por exemplo, a coluna que indica a data de nascimento é identificada como `escritores$nascimento`.

Também é possível obter um resumo da folha de registo:

```
summary(escritores)  
> summary(escritores)  
   id      nome      nascimento      morte  
Length:4      Length:4      Min.   :1825      Min.   :1871  
Class :character Class :character 1st Qu.:1836      1st Qu.:1885  
Mode  :character Mode  :character Median :1839      Median :1899  
Mean  :1842      Mean   :1901  
3rd Qu.:1845      3rd Qu.:1914  
Max.   :1864      Max.   :1934  
  
nacionalidade  
Length:4  
Class :character  
Mode  :character
```

Figura 2. Resultado do comando `summary`.

## Adição de colunas

Para juntar mais colunas, basta dar-lhes um novo nome e dizer como os valores são calculados. No primeiro exemplo, calculamos quanto tempo um dado autor viveu:

```
escritores$tempoVida<-escritores$morte-escritores$nascimento
```

No segundo exemplo, juntamos o sexo do autor que, neste caso, é sempre masculino, e escolhemos marcá-lo como "masc":

```
escritores$sexo<- "masc"
```

Embora apenas indiquemos um valor (e não um vetor), automaticamente o R repete esse valor tantas vezes quantas a dimensão da coluna.

### Adição de linhas

Também se podem juntar mais linhas, usando a função `rbind()`, que significa "row bind" em inglês (pode ser traduzido como "ligar linhas"), e que recebe como argumentos uma folha de registo e novas linhas ou duas folhas de registo. Mas, neste último caso, temos de atribuir um valor a cada coluna. Se usássemos a função `c()` (concatenar), todos os valores seriam considerados cadeias de caracteres.

```
escritores<-rbind(escritores, c("JorAma", "Jorge Amado", 1912, 2001, "BR", 89, "masc"))
```

Por isso, o melhor é criar uma nova folha de registo para o novo autor e adicioná-la:

```
escritores<-rbind(escritores, data.frame(id="JorAma", nome="Jorge Amado", nascimento=1912, morte=2001, nacionalidade="BR", tempoVida=89, s
```

### Leitura de fora do R

Como referido, a forma mais comum é ler folhas de registo provenientes de fora do R. Nesse caso vêm, em geral, de planilhas ou folhas de cálculo. Ao lê-las é possível indicar:

- Se as colunas têm nome (se tiverem, lemos com a indicação `header=TRUE`).
- Qual o separador (por exemplo, `sep="\t"` se as várias colunas estiverem separadas por tabuladores, `sep=","` se estiverem separadas por vírgulas).
- Qual o separador para casos de números com casas decimais (`dec=","` ou `dec="."`).
- Se houver uma coluna que apenas contenha identificadores (todos diferentes, portanto), também podemos indicá-lo com `rownames=4`, em que o algarismo indica o número da coluna.

Por exemplo,

```
maisescritores<-read.table("fich41.tsv", header=TRUE, sep="\t")
```

leria o que estivesse no ficheiro `fich41.tsv`, interpretando a primeira linha como o nome das colunas e os tabuladores como estando a separar as colunas.

### Processamento de colunas de uma folha de registo

Muitas vezes o que escrevemos na tabela é para ser um código de um conjunto fixo de etiquetas e não uma cadeia de caracteres em português. Nesse caso, devemos indicar ao R que é um fator (`factor`, em R, que designa uma variável categórica em estatística) e não uma palavra.

Na nossa folhinha de registo, BR e PT significam autor brasileiro e português, respetivamente, e queremos considerá-los um fator. O mesmo acontece com o sexo:

```
escritores$nacionalidade<- factor(escritores$nacionalidade)
escritores$sexo<- factor(escritores$sexo)
escritores$id<- factor(escritores$id)
```

Repare como o `summary` se torna muito mais legível:

```
> summary(escritores)
      id      nome      nascimento      morte      nacionalidade
AacAss :1  Length:5      Min.   :1825      Min.   :1871      BR:3
CamCBra:1  Class :character  1st Qu.:1839      1st Qu.:1890      PT:2
CoeNet  :1  Mode  :character  Median :1839      Median :1908
JorAma  :1      Mean  :1856      Mean   :1921
JulDin  :1      3rd Qu.:1864      3rd Qu.:1934
      Max.   :1912      Max.   :2001

      tempoVida  sexo
Min.   :32      masc:5
1st Qu.:65
Median :69
Mean   :65
3rd Qu.:70
Max.   :89
```

Figura 3. Novo resultado de `summary`.

Podemos, claro, adicionar mais valores ao fator:

```
escritores$nacionalidade<-
factor(escritores$nacionalidade, levels=c("PT", "BR", "AN"))
```

E, agora, podemos adicionar escritores angolanos:

```
escritores<- rbind(escritores,
data.frame(id="AgoNet", nome="Agostinho
Neto", nascimento=1922, morte=1979, nacionalidade="AN", tempoVida=57, se
```

## Tipos de colunas

Para além de palavras simples ou texto, uma coluna pode ter valores de um grupo (fatores), valores lógicos (TRUE ou FALSE) e valores numéricos. Estes últimos dividem-se em dois tipos:

- Números inteiros, geralmente correspondentes a contagens.
- Números reais, geralmente correspondentes a medições.

A visualização é especialmente expressiva para valores numéricos, mas os fatores são uma forma útil de organização.

## Gráficos de barras

Estes gráficos representam contagens de um certo número de características. Com a folhinha de registo que temos, a única contagem que faz sentido é a da nacionalidade. O primeiro comando tabula quantos casos temos por nacionalidade e o segundo produz um gráfico de barras:

```
table(escritores$nacionalidade)
barplot(table(escritores$nacionalidade))
```

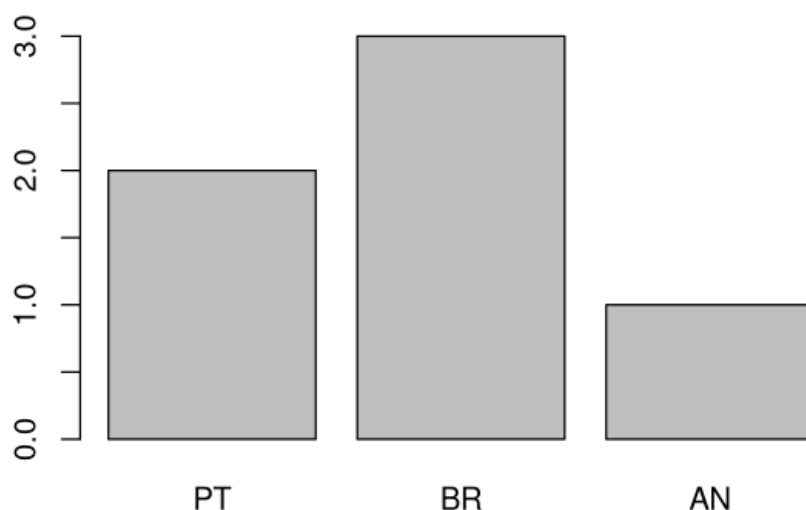


Figura 4. Gráfico de barras com a nacionalidade dos autores.

Mas vamos buscar folhas de registo muito mais ricas para demonstrar as potencialidades de visualização. Por exemplo, vejamos uma lista de obras literárias em português, usada no artigo Santos *et al.*<sup>1</sup>, com informação sobre o autor, data de publicação, escola literária e contagens de vários atributos sintáticos e semânticos.

```
periodizacao<- read.table("https://www.linguateca.pt/Diana/
Univ0slo/cursor/dadosPeriodLit.tsv",header=TRUE)
```

O primeiro argumento do comando `read.table()` indica onde se encontra o ficheiro que se pretende ler. Pode estar guardado localmente ([dadosPeriodLit.tsv](#) (`// assets/visualizacao-basica-dados-tabulares-r/dadosPeriodLit.tsv`)) ou estar acessível através de um URL.<sup>2</sup>

Através do comando `names()` podemos verificar os nomes das colunas. Já através de `str()` ou `summary()` podemos verificar o tipo de informação que cada coluna tem:

```
names(periodizacao)
str(periodizacao)
summary(periodizacao)
```

Por agora, basta caracterizar o autor, o sexo, o género literário e a escola literária como fatores. `escola2` inclui uma versão simplificada da escola literária, com os nomes em inglês:

```
periodizacao$autor<- factor(periodizacao$autor)
periodizacao$sexo<- factor(periodizacao$sexo)
periodizacao$genero<- factor(periodizacao$genero)
periodizacao$escola<- factor(periodizacao$escola)
```

Podemos, assim, identificar as escolas literárias presentes no material (removemos

da figura os casos em que a escola é desconhecida, marcados como desc ):

```
periodSemDesc<-subset(periodizacao, escola2!="desc")
periodSemDesc$escola2<-factor(periodSemDesc$escola2)
barplot(table(periodSemDesc$escola2))
```

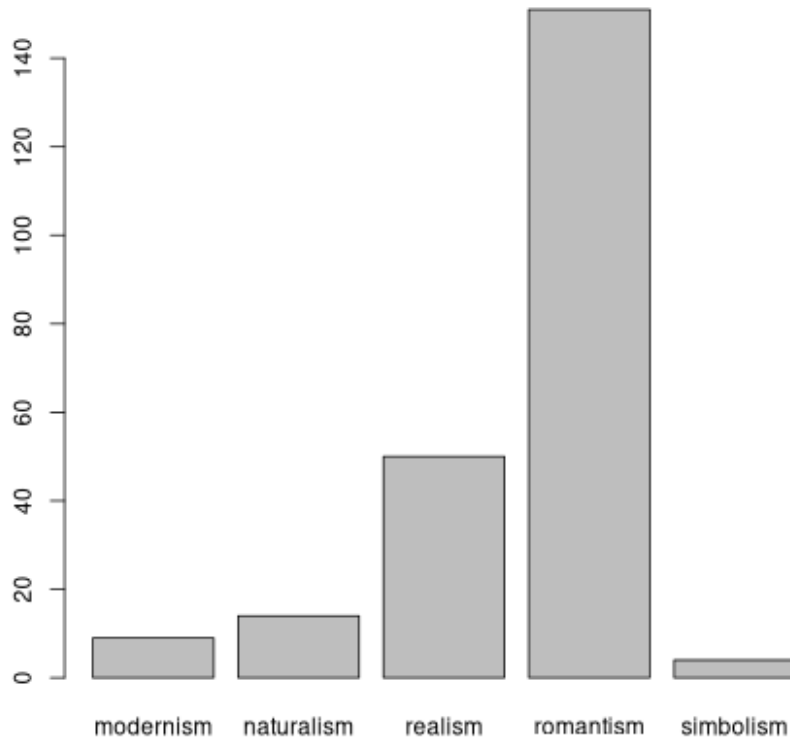


Figura 5. Gráfico de barras com as escolas literárias (em inglês).

Criamos uma nova folha de registo, chamada `periodSemDesc`, a partir da folha de registo original, contendo apenas os casos em que se conhece a escola literária. A indicação `escola2!="desc"` significa que o valor da coluna `escola2` deve ser diferente de `desc`. Se, pelo contrário, quiséssemos um valor igual, usaríamos o sinal `==` em vez de `!=`.

Se tivéssemos executado simplesmente os comandos

```
periodizacao$escola2<-factor(periodizacao$escola2)
barplot(table(periodizacao[periodizacao$escola2!
="desc",]$escola2))
```

o gráfico de barras apresentaria uma barra nula para `desc`, como vemos na Figura 6:

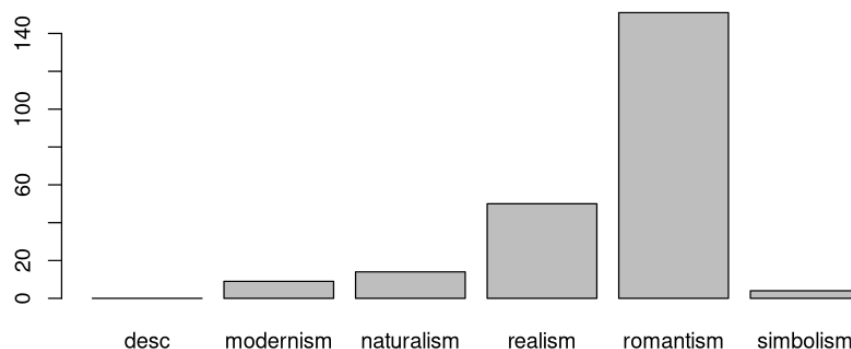


Figura 6. Gráfico de barras com as escolas literárias (em inglês), sem termos removido o valor desc do fator escola.

A indicação `periodizacao[periodizacao$escola2!="desc",]` significa todas as linhas da folha de registo `periodizacao` cuja coluna `escola2` não tenha o valor `desc` e todas as colunas. (Uma folha de registo tem sempre linhas e colunas e podemos seleccioná-las independentemente. Quando não colocamos nada, como depois da vírgula, significa que seleccionamos todas.)

## Gráficos de caixa

Os gráficos de caixa representam um conjunto de números, mostrando a sua mediana e a forma como estão distribuídos. A caixa propriamente dita representa 50% dos dados: a linha de baixo representa o valor dos 25% e a de cima o dos 75%, designados por primeiro quartil e terceiro quartil. A diferença entre Q3 e Q1 é chamada diferença entre quartis ("interquartile range" em inglês, geralmente abreviado por IQR). Os traços horizontais, também chamados bigodes ("whiskers" em inglês), são calculados da seguinte forma:

- Bigode inferior: é o máximo de valor mínimo e de  $Q1 - 1.5 * IQR$ .
- Bigode superior: é o mínimo do valor máximo e de  $Q3 + 1.5 * IQR$ .

Quando há casos fora dos limites descritos pelos bigodes, chamamos-lhes valores discrepantes ("outliers" em inglês). Estes são marcados como pontos discretos.

Veja-se esta figura, retirada do tutorial de Yi, *A complete guide to boxplots* (<https://www.atlassian.com/data/charts/box-plot-complete-guide>) (em inglês):<sup>3</sup>

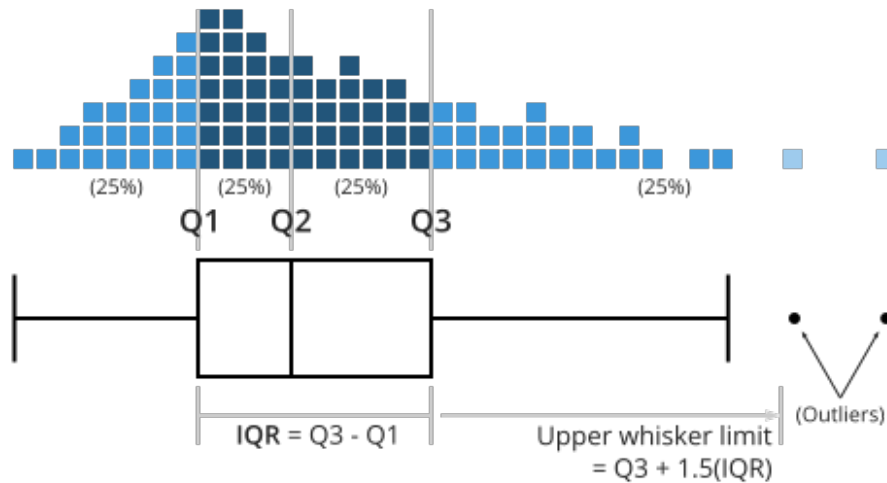


Figura 7. Explicação de um gráfico de caixa, retirada de Yi.

Os gráficos de caixa são úteis, sobretudo, para comparar vários conjuntos de dados. Vejamos, no nosso caso, a diferença no uso de cor por escola literária:

```
boxplot(periodizacao$cor~periodizacao$escola2)
```

O til ( ~ ) é como se designa “por” em R e espera que a indicação à direita seja um fator. À esquerda, temos valores numéricos para fazer os variados gráficos de caixa, um por cada valor do fator.

### Número de palavras de cor por escola literária

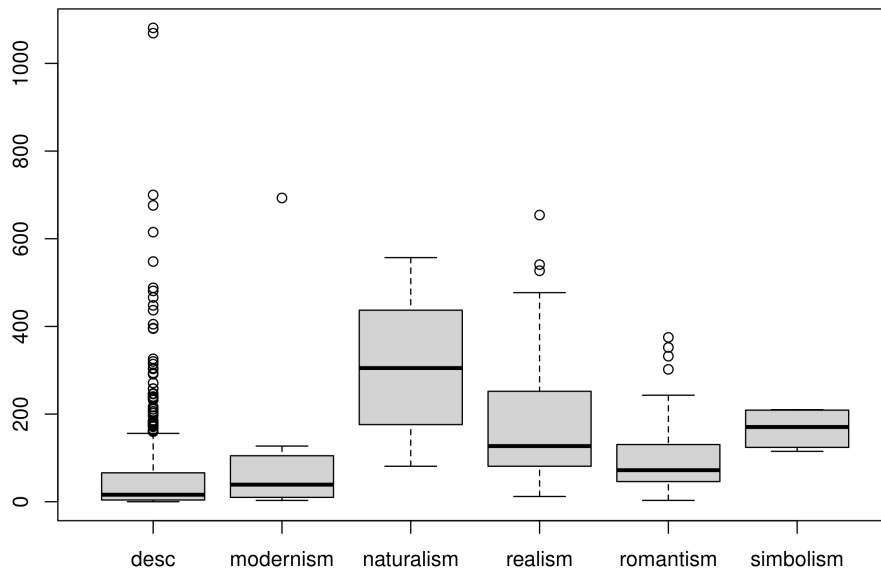


Figura 8. Gráfico de caixa da presença de palavras de cor por escola literária (em inglês).

O resultado mostra-nos que o naturalismo tem mais cor. Já o romantismo parece ter menos palavras de cor do que o realismo. Não nos interessa, aqui, efetuar uma análise literária, mas apenas ilustrar o uso dos gráficos de caixa e a sua interpretação.

De facto, para poder comparar devidamente um grande conjunto de obras de tamanho variável, deveríamos ter calculado a percentagem de palavras de cor e não o número de palavras de cor:

```
boxplot(periodizacao$cor/  
periodizacao$tamanho~periodizacao$escola2)
```

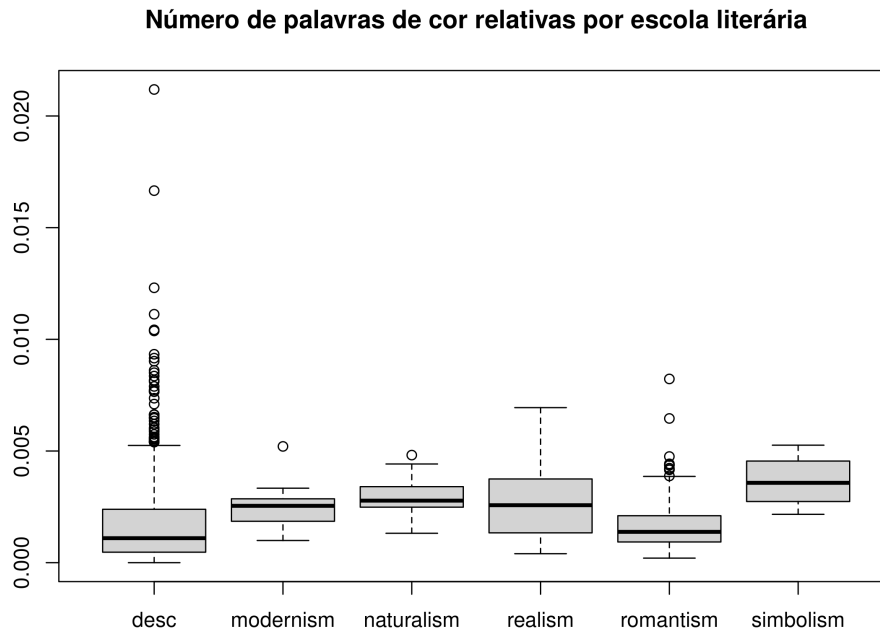


Figura 9. Gráfico de caixa da presença relativa de palavras de cor por escola literária (em inglês).

## Mais operações sobre folhas de registo

### Obter subconjuntos

Vejamos, agora, mais potencialidades do uso e criação de folhas de registo, através da função `subset()`, que permite escolher um subconjunto de colunas e de linhas e criar uma nova folha de registo, à qual aplicaremos mais visualizações:

```
algunsAutores<-subset(periodizacao, (autor=="JulDin" |  
autor=="EcaQue" | autor=="MacAss" | autor=="CoeNet") &  
genero=="Prosa: romance", c(1:147))
```

Como só temos quatro autores, faz sentido dizer que os outros valores do fator devem ser ignorados, o que se faz com o seguinte comando:

```
algunsAutores$autor<-algunsAutores$autor[drop=TRUE]
```

Podemos verificar quantas obras temos por autor:

```
barplot(table(algunsAutores$autor))
```

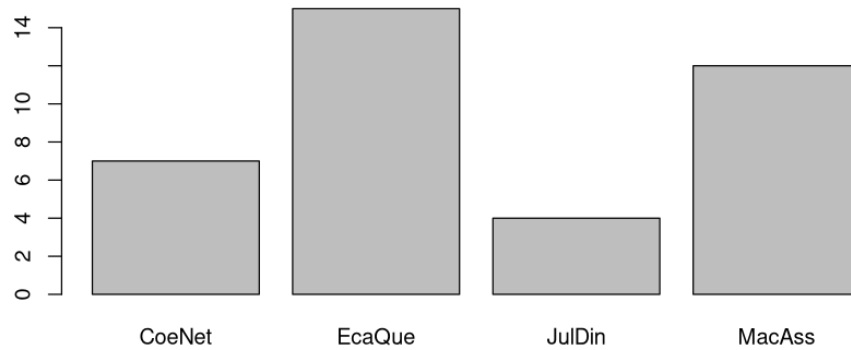


Figura 10. Gráfico de barras com o número de obras por autor.

Mas o mais interessante será comparar estes quatro autores analisando, por exemplo, a frequência relativa de emoções, no uso de nomes próprios ou na frequência de orações no conjuntivo/subjuntivo:

```

boxplot(algunsAutores$emocoes/
algunsAutores$tamanho~algunsAutores$autor,xlab="",ylab="",
main="Frequência relativa de uso de palavras de emoção em
romances por autor")
boxplot(algunsAutores$propios/
algunsAutores$tamanho~algunsAutores$autor,xlab="",ylab="",
main="Frequência relativa de uso de nomes próprios em romances
por autor")
boxplot(algunsAutores$conjuntivo/
algunsAutores$oracoes~algunsAutores$autor,xlab="",ylab="",
main="Frequência de orações no conjuntivo em romances por
autor")

```

**Frequência relativa de uso de palavras de emoção em romances por autor**

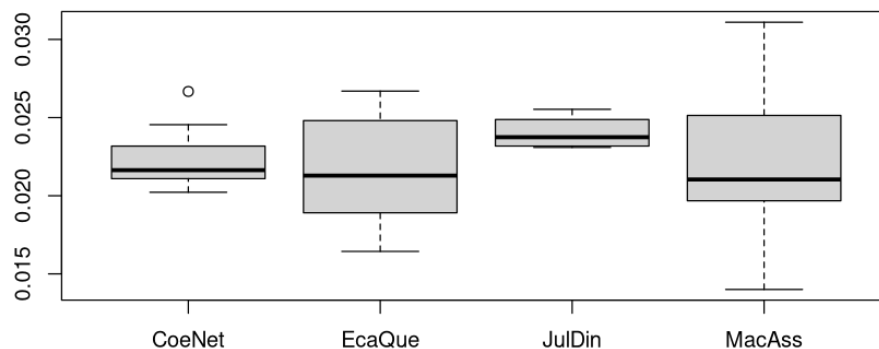


Figura 11. Gráfico de caixa da presença relativa de palavras de emoção por escola literária (em inglês).

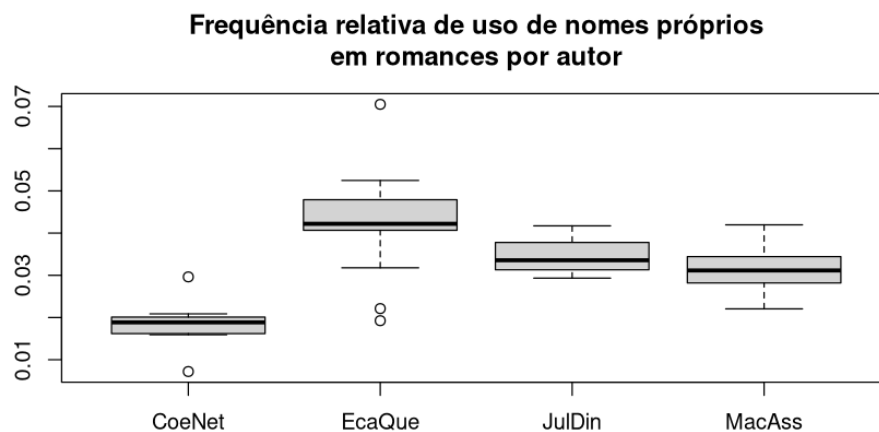


Figura 12. Gráfico de caixa da presença relativa de nomes próprios por escola literária (em inglês).

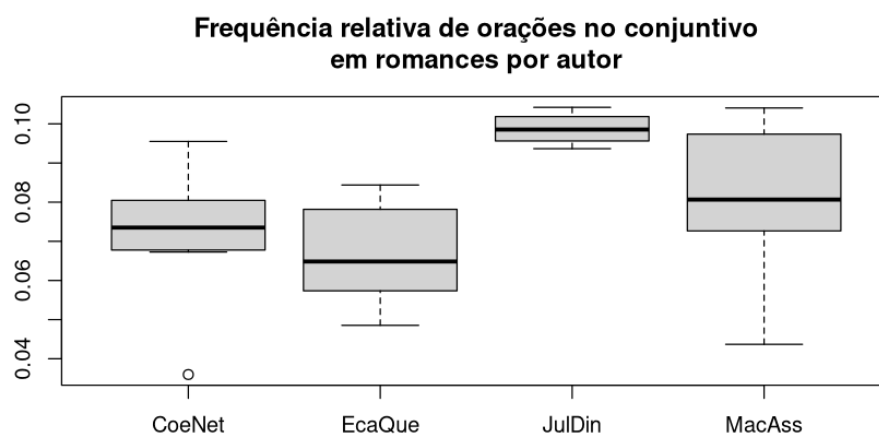


Figura 13. Gráfico de caixa da presença relativa de orações no conjuntivo por escola literária (em inglês).

Juntar mais do que uma folha de registo numa só🔗

Finalmente, para mostrar ainda mais potencialidades do uso das folhas de registo e da forma como a informação pode ser bem distribuída em folhas de registo diferentes, vamos criar uma nova que contenha toda a informação de duas folhas de registo que já usámos: `algunsAutores` e `escritores`. A cada obra queremos associar, para além do nome do autor, informação nova que temos sobre ele, nomeadamente, a variante, o tempo de vida e o sexo. Para isso usamos o comando `merge()` :

```
maisInfo<-merge(algunsAutores,escritores,by.x=c("autor",
"sexo"),by.y=c("id","sexo"))
```

Este comando é muito importante — correspondente ao “join” das bases de dados — porque permite estruturar o conhecimento em ficheiros (tabelas) diferentes, mas juntá-lo quando queremos usar toda a informação. Na Figura 14 vemos as primeiras linhas da folha de registo `maisInfo` :

```

> head(maisInfo[1:8])
  autor sexo   Obra id tamanho      genero aspetu compl
1 CoeNet masc A_conquista 360  31999 Prosa:romance    57  176
2 CoeNet masc   Turbilhão 370  88164 Prosa:romance   328  595
3 CoeNet masc     Esfinge 364  48377 Prosa:romance    67  188
4 CoeNet masc  Rei_negro 368  67053 Prosa:romance    96  158
5 CoeNet masc   Miragem 366  76151 Prosa:romance   210  456
6 CoeNet masc     Mano 365  38318 Prosa:romance    92  308
> |

```

Figura 14. O que o R mostra se pedirmos as primeiras linhas da folha de registo MaisInfo.

O uso deste comando permite-nos, por exemplo, fazer um diagrama de caixa pela variedade do português e não pelos autores. Na Figura 15 observamos o uso das vírgulas:

```

boxplot(maisInfo$virg/
maisInfo$tamanho~maisInfo$nacionalidade,xlab="",ylab="",
main="Frequência relativa de uso de vírgulas em romances por
variante")

```

**Frequência relativa de uso de vírgulas em romances por variante**

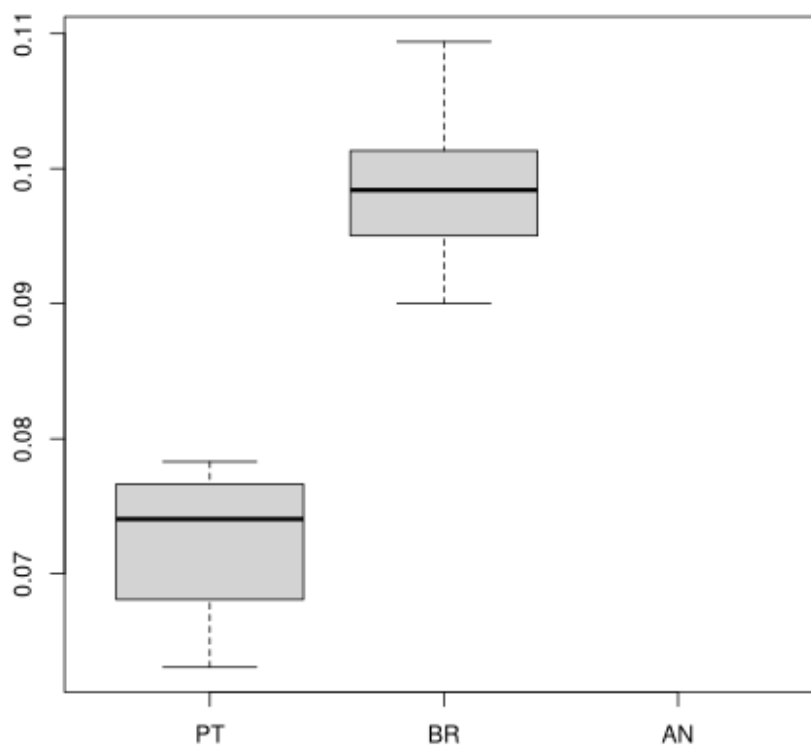


Figura 15. Gráfico de caixa da presença relativa de vírgulas por variante.

### Guardar folhas de registo

Finalmente, assim como é possível ler folhas de registo de fora do R, também é possível guardá-las fora do R, para serem usadas por outros programas ou quando voltarmos a este ambiente (o R). Para isso o comando mais usual é o `write.table()`.

Vamos guardar a folha de registo `maisInfo` num ficheiro chamado `obras4autoresComInfoAutor.txt` (mais propriamente, deveria ser chamado `.tsv`, visto que o separador vai ser um tabulador indicado por `sep="\t"`, mas a extensão `.txt` permite ler diretamente num navegador ("browser", em inglês)):

```
write.table(maisInfo,"obras4autoresComInfoAutor.txt", sep="\t",
quote=FALSE)
```

`quote=FALSE` indica que os valores não serão envolvidos em aspas, o que seria o caso se fosse `TRUE`.

## Valores que faltam

Uma questão real de observações empíricas é que pode haver valores a que não temos acesso. As folhas de registo com grandes quantidades de dados invariavelmente têm esse problema.

Por outro lado, pode haver razões para não haver dados em algumas colunas, mesmo em questões triviais: no exemplo que temos vindo a desenvolver, como preencher a data da morte de um autor ainda vivo?

O R tem o conceito de valor `NA` ("not available" em inglês, significando inexistente em português). Praticamente todas as funções do R têm um comportamento apropriado para estes valores. Além disso, é possível testar e identificar os casos que faltam, através das funções `is.na()` ou `na.exclude()`.

No caso dos diagramas apresentados na presente lição, esses casos são simplesmente excluídos da visualização, como podemos verificar adicionando um autor ainda vivo e pedindo um diagrama de caixa do tempo de vida:

```
escritores<-rbind(escritores,
data.frame(id="Pepet",nome="Pepetela",nascimento=1941,morte=NA,naci
boxplot(escritores$tempoVida~escritores$nacionalidade)
```

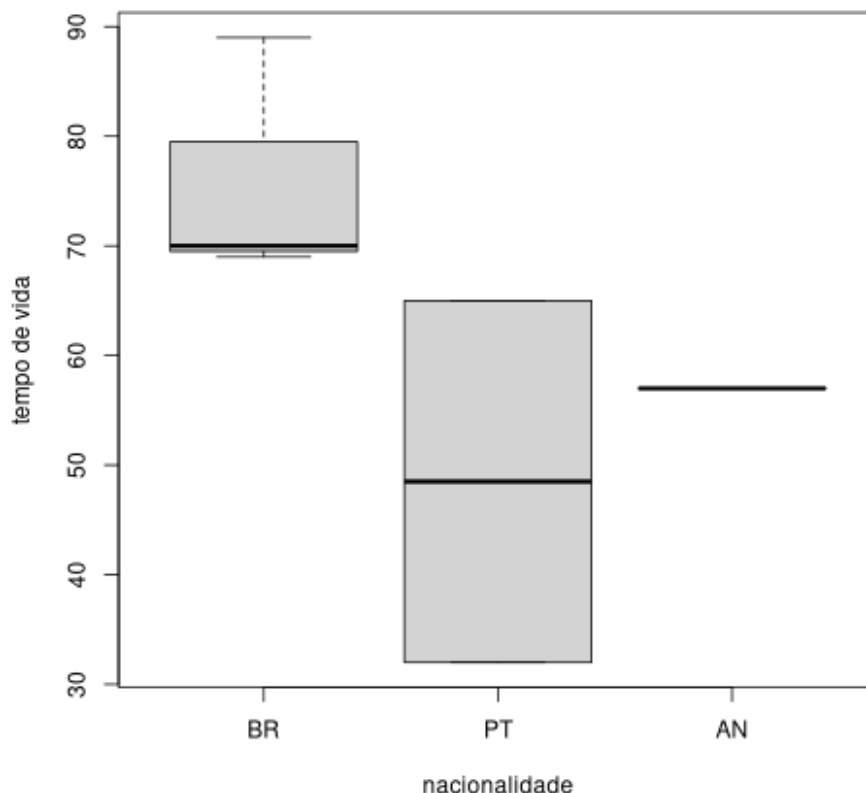


Figura 16. Gráfico de caixa do tempo de vida de alguns escritores por nacionalidade.

Como só existe um autor angolano com tempo de vida diferente de NA, nomeadamente, Agostinho Neto, só é mostrado no gráfico de caixa um ponto, visualizado como uma linha.<sup>4</sup>

## Observações finais

Nesta lição explicámos o conceito e as funcionalidades de uma folha de registo e algumas formas simples de visualizar o seu conteúdo, usando gráficos de barras e gráficos de caixa.

Agora, pode seguir para lições mais complicadas como “Investigar a literatura lusófona através dos tempos usando a Literateca” (a publicar por *Programming Historian* em 2025), em que as folhas de registo vêm do projeto AC/DC.


## Notas de fim

1. Diana Santos, Emanuel Pires, João Marques Lopes, Rebeca Fuão & Cláudia Freitas. “Periodização automática: Estudos linguístico-estatísticos de literatura lusófona” In *Linguamática*, vol. 12, nº 1 (2020): 80-95. ↵
2. Em alguns navegadores este comando pode produzir o seguinte erro: `Error in file(file, "rt") : cannot open the connection to 'https://www.linguateca.pt/...' . Nesse caso, leia o ficheiro para o seu próprio computador fora do R e execute apenas read.table("dadosPeriodLit.tsv", header=TRUE) . ↵`
3. Mike Yi. *A complete guide to box plots*, <https://www.atlassian.com/data/>.

[charts/box-plot-complete-guide](https://www.atlassian.com/data/charts/box-plot-complete-guide) (<https://www.atlassian.com/data/charts/box-plot-complete-guide>) (em inglês, consultado a 3 de maio de 2024). ↵

4. O comportamento pode variar em versões diferentes do R. Se obtiver uma mensagem de erro do `boxplot`, antes de o invocar, execute os comandos seguintes, que retiram explicitamente os NA e transformam as idades em inteiros: `novoEscritores<-na.omit(escritores)`  
`novoEscritores$tempoVida<-as.integer(novoEscritores$tempoVida)`  
`boxplot(novoEscritores$tempoVida~novoEscritores$nacionalidade)`  
↵

## SOBRE O(A) AUTOR(A)

Diana Santos é professora catedrática no Departamento de Literatura, Estudos de Área e Línguas Europeias da Universidade de Oslo, Noruega.  (<https://orcid.org/0000-0002-3108-7706>)

## CITAÇÃO SUGERIDA

Diana Santos, "Visualização básica de dados tabulares com R", *Programming Historian em português* 5 (2025), <https://doi.org/10.46430/phpt0053>.

*The Programming Historian em português* (ISSN: 2753-9296) é publicado com uma licença [CC-BY](https://creativecommons.org/licenses/by/4.0/deed.pt) (<https://creativecommons.org/licenses/by/4.0/deed.pt>).


Este projeto é administrado pelo ProgHist Ltd, com o número de instituição de caridade [1195875](https://register-of-charities.charitycommission.gov.uk/charity-search/-/charity-details/5181272/charity-overview) (<https://register-of-charities.charitycommission.gov.uk/charity-search/-/charity-details/5181272/charity-overview>) e número de companhia [12192946](https://find-and-update.company-information.service.gov.uk/company/12192946) (<https://find-and-update.company-information.service.gov.uk/company/12192946>).


[ISSN 2397-2068 \(inglês\) \(/\)](#)

[ISSN 2517-5769 \(espanhol\) \(/es\)](#)


[ISSN 2631-9462 \(francês\) \(/fr\)](#)


**[ISSN 2753-9296 \(português\) \(/pt\)](#)**

 [Hospedado no GitHub](https://github.com/programminghistorian/jekyll) (<https://github.com/programminghistorian/jekyll>)


 Última atualização em 30 October 2025 (<https://github.com/programminghistorian/jekyll/commits/gh-pages>)

 [Subscrição RSS feed](https://programminghistorian.org/feed.xml) (<https://programminghistorian.org/feed.xml>)

 [Histórico da página](https://github.com/programminghistorian/jekyll/commits/gh-pages/pt/licoes/visualizacao-basica-dados-tabulares-r.md) (<https://github.com/programminghistorian/jekyll/commits/gh-pages/pt/licoes/visualizacao-basica-dados-tabulares-r.md>)

 [Envie a sua sugestão \(/pt/reportar-um-erro\)](#)

[Política de remoção de lições \(/pt/licoes-politica-remocao\)](#)

 [Concordância das traduções \(/translation-concordance\)](#)