LUCIANE FERREIRA ALCOFORADO
DOCENTE ASSOCIADA DO DEPARTAMENTO DE ESTATÍSTICA DA UFF

UTILIZANDO A LINGUAGEM R

CONCEITOS, MANIPULAÇÃO, VISUALIZAÇÃO, MODELAGEM E ELABORAÇÃO DE RELATÓRIOS

ALTA BOOKS
EDITORA
Rio de Janeiro, 2021

SUMÁRIO

APRESENTAÇÃO

Que o R é uma ferramenta computacional incrível e apaixonante para análise de dados nenhum especialista no assunto duvida. Investir no seu aprendizado é ingressar numa comunidade ativa e colaborativa que mantém o software R gratuito e atualizado, com inovações incorporadas em cada melhoria ou produção de novos pacotes, compondo atualmente uma biblioteca com quase 20 mil pacotes que permitem análises e manipulações de dados muito mais dinâmicas. Diante de tantos recursos que a linguagem tem a oferecer, você deve estar se perguntando por onde devo começar?

O livro conduzirá o leitor a um caminho de estudos sobre os potenciais usos da linguagem R e seus pacotes, as versões dessa linguagem não têm grandes alterações, portanto o conceito aqui exposto é aplicável a novas versões. Abrange desde a leitura, manipulação, visualização, modelagem e elaboração de um relatório utilizando um pacote denominado *R Markdown* que permite a integração de todas as tarefas que um pesquisador que utilizará análise de dados deverá realizar para produzir seu relatório final, ou seja, num único ambiente é possível realizar a leitura dos dados, escrever o texto, utilizar as ferramentas de análise estatística e produzir um relatório final.

A autora desta obra vem acompanhando a evolução dessa linguagem computacional há mais de dez anos e agora traz ao leitor esse conhecimento acumulado, economizando horas de busca e conduzindo-o para o uso efetivo das ferramentas aplicadas às suas análises de dados, desde a leitura de uma planilha, passando pela limpeza e seus ajustes — tarefa sempre necessária e relativamente fácil de realizar no R — até a produção de relatório contendo análises, modelagem, tabelas com as estatísticas descritivas e gráficos, passos esses fundamentais para o desenvolvimento de uma pesquisa.

O objetivo desta obra é auxiliar o estudante a utilizar os principais pacotes do imenso universo de quase 20 mil possibilidades, tornar o trabalho de pesquisa empolgante à medida que vai conhecendo as ferramentas disponíveis e mergulhando no mundo do R. É possível dizer que o R é uma linguagem de programação com forte vocação para tratamento e análise de dados, sua primeira versão oficial (versão 0.99) data do ano 2000 e em 2019 circula a versão de terceira geração (versão 3.6), a qual iremos utilizar nesta obra. Grande parte dos códigos aqui tratados funcionam em versões mais antigas, especialmente os códigos do pacote básico. As atualizações da linguagem procuram manter uma estabilidade nas funções, melhorando e corrigindo alguns erros que são detectados pelos usuários e colaboradores. Trata-se de uma linguagem dinâmica que conta com uma intensa produção de pacotes que são desenvolvidos diariamente por uma rede de colaboradores do mundo todo. A cada nova versão a linguagem vai ganhando maturidade e, nesse sentido, recomenda-se realizar atualizações periódicas sem que haja prejuízo aos *scripts* já construídos, isto é, os *scripts* contidos neste livro poderão ser aplicados em versões futuras da linguagem. Esse dinamismo de produção de novos pacotes possibilita ao analista acesso a ferramentas cada vez mais poderosas e funcionais, mas que por outro lado exige um processo de atualização constante, especialmente através de obras como esta que conduzirão você nesse processo.

Assim, este livro é útil para quem já conhece o R e quer se atualizar, já que muita coisa mudou nos últimos anos, como também serve para quem está começando a usar e não sabe por onde começar. Em ambos os casos, será desenvolvido por meio desta obra uma integração entre os pacotes disponíveis de uma forma que revolucionará a maneira como o pesquisador desenvolverá suas análises, agora de forma integrada, conjugando texto e códigos do R sem necessidade de abrir vários arquivos ao mesmo tempo naquele cansativo processo de copiar e colar.

A objetividade do conteúdo, exemplos simples que o leitor poderá facilmente reproduzir, focado no conceito e na funcionalidade, formam

a guia mestre da narrativa desta obra. Há uma criteriosa seleção de pacotes importantes para cada passo da análise de dados que um pesquisador não estatístico deve conhecer para que em pouco tempo tenha autonomia para realizar suas análises de forma independente.

O livro apresenta inicialmente o R tradicional para em seguida apresentar o sistema tidyverse e outros pacotes que permitem manipular dados como, por exemplo, o *data.table*, fornecendo desse modo três formas diferentes de resolver o mesmo problema. Na visualização, apresenta-se rapidamente o modo tradicional para em seguida mostrar o sistema tidyverse procurando sempre mostrar mais de uma forma de fazê-la. Para um iniciante em R conhecer o pacote *DescTools* pode ser uma ferramenta muito versátil para realizar a análise descritiva de forma rápida, acredito que esse seja um grande diferencial em relação ao livro *R para Data Science*. Mostro através de exemplos como interpretar cada resultado obtido. Outro diferencial é a abordagem de testes de hipóteses, sua formulação e como modelar duas variáveis, aplicar o teste e chegar a uma conclusão. O livro procura estabelecer uma trilha de aprendizado que segue a sequência lógica que um pesquisador deve trilhar até a geração do relatório final.

O interesse hoje em aprender o R se espalha por diversas profissões como as de professores, estatísticos, engenheiros, administradores, biólogos, químicos, matemáticos, geógrafos, programadores, contadores, sociólogos, jornalistas, psicólogos, cientistas políticos, médicos, enfermeiros, assistentes sociais e muitos outros.

Em suma, muito mais do que introduzir e mostrar como o R funciona, trago ao leitor a possibilidade de formular e modelar questões de pesquisa, interpretar os indicadores gerados e concluir a análise, gerando finalmente um relatório. Trato nesta obra, portanto, de conduzir o estudante pelas melhores opções que o mundo do R pode proporcionar.

1

Entre para o ambiente do R Studio

OBJETIVO

Neste capítulo você será introduzido ao ambiente R, aprenderá a instalar o software R e o R Studio e a compreender a diferença entre eles. Ao seu final, você estará apto a iniciar seu primeiro arquivo de *script* na linguagem R.

O que é o R?

É uma linguagem idealizada para realizar análise de dados através de um sistema para computação estatística e gráfica, permitindo explorar dados, produzir funções, computar linhas de comando ou utilizar pacotes disponíveis na rede CRAN (Comprehensive R Archive Network). Trata-se de um sistema de licença livre, sem qualquer ônus, e sua disseminação pela comunidade acadêmica permitiu que um grande número de pessoas contribuísse para sua evolução, produzindo um acervo de quase 20 mil pacotes disponíveis hoje e que se encontra em franco crescimento.

Sua popularidade iniciou no ano 2000 quando professores de Estatística da Nova Zelândia tornavam pública a primeira versão do software R. Talvez por coincidência, as iniciais do nome desses professores iniciam pela letra R, Ross Ihaka e Robert Gentleman, uma boa hipótese para o terem nomeado de R.

A linguagem R em processo contínuo de expansão, alcançou em 2015 a sexta posição entre as linguagens computacionais mais populares do mundo, em 2016 subiu para a quinta e a tendência é de que se mantenha sempre no topo do ranking, divulgado anualmente pela revista online IEEE Spectrum.

Como obter o software R?

É muito simples e rápido obter o software R, basta acessar a página do projeto e realizar o download através do endereço https://cran.r--project.org/, selecionando o sistema operacional do seu equipamento, disponível para Linux, Mac e Windows. Nesse link você encontrará a última versão do R. Por exemplo, em abril de 2019 foi disponibilizada a versão 3.6.0. Quando você o instalar pela primeira vez, terá acesso a

versão mais atual, que funcionará perfeitamente até que você faça uma nova atualização.

Após o download, você deve instalar o programa, abrir o software para então visualizar a janela com alguns avisos, dicas e uma linha com sinal ">" indicando o local onde devem ser inseridos os comandos (figura 1). Se quiser ver o procedimento de instalação em detalhes, assista ao vídeo http://bit.ly/installReRStudio.

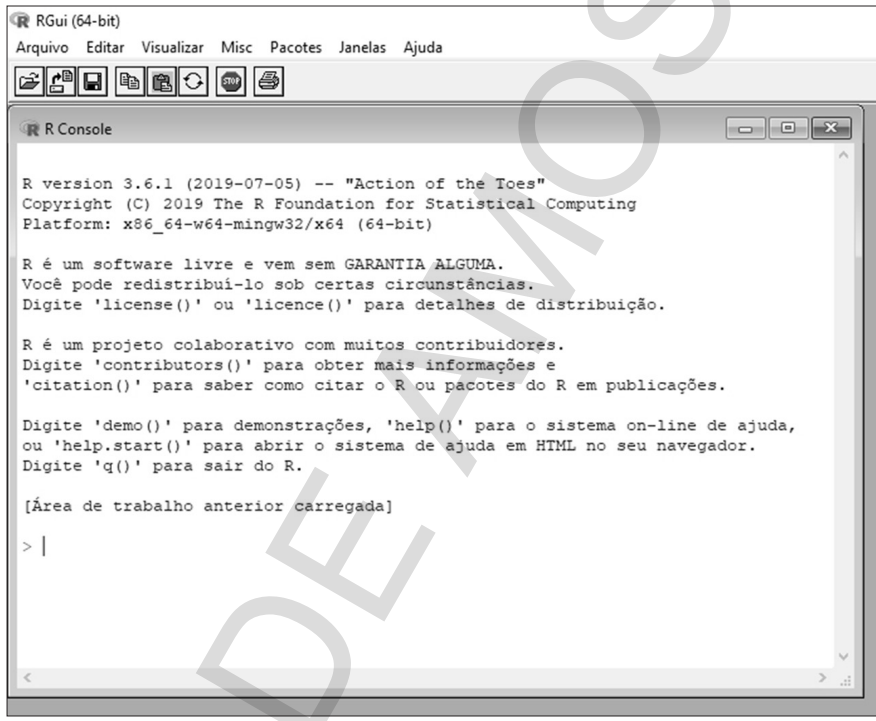


Figura 1: janela do R

A atualização do R não é automática, recomenda-se sua atualização uma vez por ano, e o procedimento é o mesmo da instalação inicial. Tenha a certeza de que o R básico já tem uma estabilidade e funcionará adequadamente nas versões dos últimos anos. A atualização é necessária principalmente para que os pacotes funcionem adequadamente,

eles são constantemente atualizados e evoluem com o passar do tempo, então cultive o hábito de atualizar tudo uma ou duas vezes por ano.

Para saber sobre os pacotes disponíveis no CRAN consulte https://cran.r-project.org/web/packages/index.html. É possível ver a lista por data de publicação ou por nome do pacote.

O que é o R Studio?

O R Studio não é o R, e sim um ambiente de desenvolvimento integrado do R, portanto ele contém o R e o acesso a todos os pacotes disponíveis no CRAN. Atualmente é o melhor ambiente para desenvolvimento de pesquisas e relatórios com análise de dados em que se faça uso do R.

Nele você pode escrever um pequeno relatório, produzir textos para blogs, para apresentações e muito mais. E é isso que iremos desenvolver junto com você neste livro. Então, mãos a obra!

Instale primeiro o R e em seguida o R Studio. Para instalar o R Studio, você deve fazer o *download* através do endereço http://www.rstudio.com/products/rstudio/download/, selecionando o sistema operacional do seu equipamento: disponível para Mac, Windows, Ubuntu, Fedora.

Se tiver dúvidas sobre a instalação assista ao vídeo do canal no Youtube do grupo Estatística é com R da UFF em https://www.youtube.com/watch? v=8LnZNC4hxdQ.

Após abrir o R Studio, você será levado a um ambiente que se divide em três ou quatro janelas (figura 2): na janela inferior esquerda você visualizará o console do R, após abrir seu primeiro arquivo verá na parte superior esquerda os arquivos que produzirão seus relatórios ou simplesmente o *script* de comandos; na janela direita há diversas informações sobre os objetos, o histórico de comandos, os pacotes disponíveis e auxílio para instalação, ajuda e muito mais!

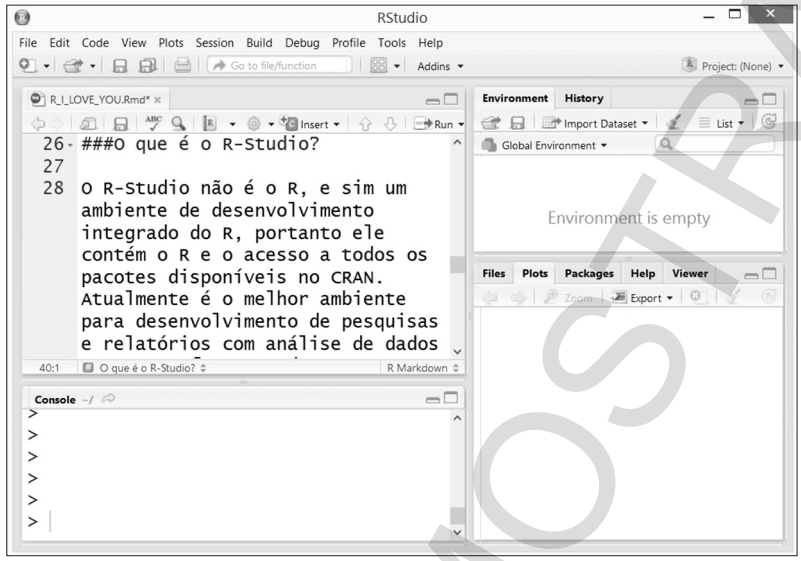


Figura 2: janela do R Studio

Pronto! Agora você já pode começar abrindo seu primeiro *script*. Basta acessar a aba *File — New File — R script*. Veja figura 3:

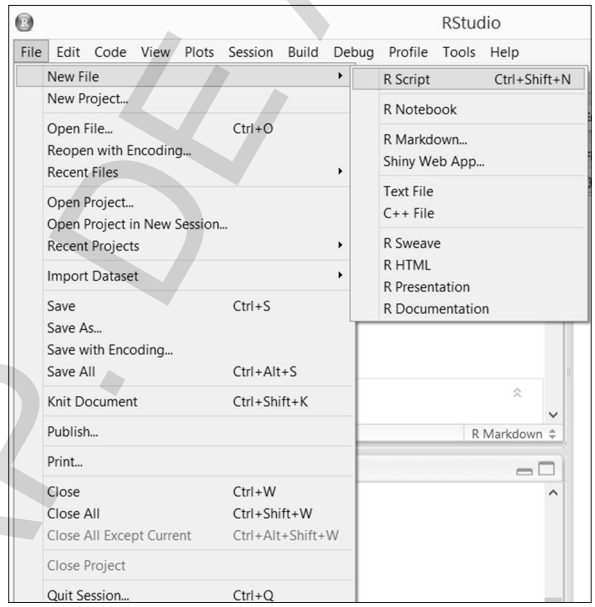


Figura 3: R Script

Após clicar em R Script, você abrirá seu primeiro arquivo de *script*. Nas linhas numeradas você inserirá os comandos da linguagem, conforme pode ser visto na figura 4. No próximo capítulo você aprenderá a executar esses comandos e realizar muitos outros!

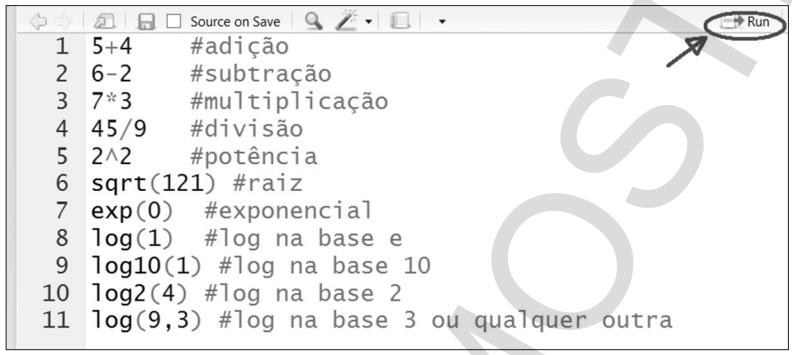


Figura 4: R Script com comandos

EXERCÍCIOS DE FIXAÇÃO PARA O APRENDIZADO

Responda as perguntas a seguir:

|1| O que é o R?

|2| Quais sistemas operacionais podem rodar o R?

|3| Qual o comando para solicitar ajuda no R?

|4| Qual o comando para citar o R?

|5| O que é a R Foundation?

|6| O que é o R Journal?

|7| O que é o R Studio?

|8| Como citar um pacote do R?

2

Aprenda o essencial do pacote base

OBJETIVO

Neste capítulo você será introduzido aos comandos do pacote básico do R e aprenderá a realizar desde operações básicas até a criação de tabela de dados e sua manipulação. Ao final, você estará apto a criar *scripts* para manipulação de vetores, matrizes e tabela de dados.

Quando você instala o R, alguns pacotes já se tornam disponíveis automaticamente, como por exemplo: *stats, graphics, grDevices, utils, datasets, methods e base*. Iremos explorar alguns comandos do pacote base. Tudo que você precisa fazer neste momento é copiar os comandos a seguir no seu arquivo de *script* e apertar o botão *Run* com o cursor posicionado na primeira linha. O resultado será mostrado na parte inferior (Console). Se tiver dúvida sobre como criar um arquivo *script*, veja a figura 3 no capítulo anterior.

Operações matemáticas

```
5+4 #adição

## [1] 9

6-2 #subtração

## [1] 4

7*3 #multiplicação

## [1] 21

45/9 #divisão

## [1] 5

2^2 #potência

## [1] 4

sqrt(121) #raiz

## [1] 11

exp(0) #exponencial

## [1] 1

log(1) #log na base e

## [1] 0

log10(1) #log na base 10

## [1] 0

log2(4) #log na base 2

## [1] 2

log(9,3) #log na base 3 ou
qualquer outra

## [1] 2
```

Vetores

Para criar um objeto tipo vetor usamos a função ***c( )***. Para usar uma função do R basta saber o nome e os argumentos que devem ser colocados entre parênteses, separados por vírgula. No caso do vetor, os argumentos são a sequência de números ou caracteres:

```
#cria um vetor com 3 números
c(7, 4, 1)

## [1] 7 4 1

#cria um vetor com 3 nomes (vetor de caractere)
c("sete", "quatro", "um")

## [1] "sete" "quatro" "um"
```

O comando *c(sete, quatro, um)* não funciona e dá retorno de erro, pois faltam as aspas nos nomes para que fique correto.

Em todos os vetores aparece [1] antes dos elementos do vetor — algo é comum no R — para indicar a posição do objeto que vem logo em seguida. Se tivermos um vetor com muitos valores é um bom recurso para ter ideia da posição dos elementos:

```
#cria uma sequência de números inteiros entre 5 e 35

5:35

## [1] 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
## [24] 28 29 30 31 32 33 34 35
```

Nesse exemplo fica mais claro a indicação das posições entre colchetes. Na segunda linha, o próximo valor aparece na posição 24.

Os vetores devem ser armazenados em um objeto com um nome apropriado:

```
#nota de 3 avaliações do aluno 1
#armazenada no objeto Nota.aluno1
Nota.aluno1 = c(8, 8.6, 8.8)

#nota de 3 avaliações do aluno 2
#armazenada no objeto Nota.aluno2
Nota.aluno2 = c(7.3, 6.7, 7)

#consultando o conteúdo do objeto Nota.aluno1
Nota.aluno1

## [1] 8.0 8.6 8.8



#consultando o conteúdo do objeto Nota.aluno2
Nota.aluno2

## [1] 7.3 6.7 7.0
```

Atente-se que *nota.aluno1* é diferente de *Nota.aluno1*, pois inicia com *n minúsculo* e portanto não existe para o R uma vez que criamos um objeto com nome iniciando por *N maiúsculo*. As letras maiúsculas e minúsculas produzem objetos distintos para a linguagem R e isso pode ser um fator de erro para quem está iniciando.

Tabela de dados

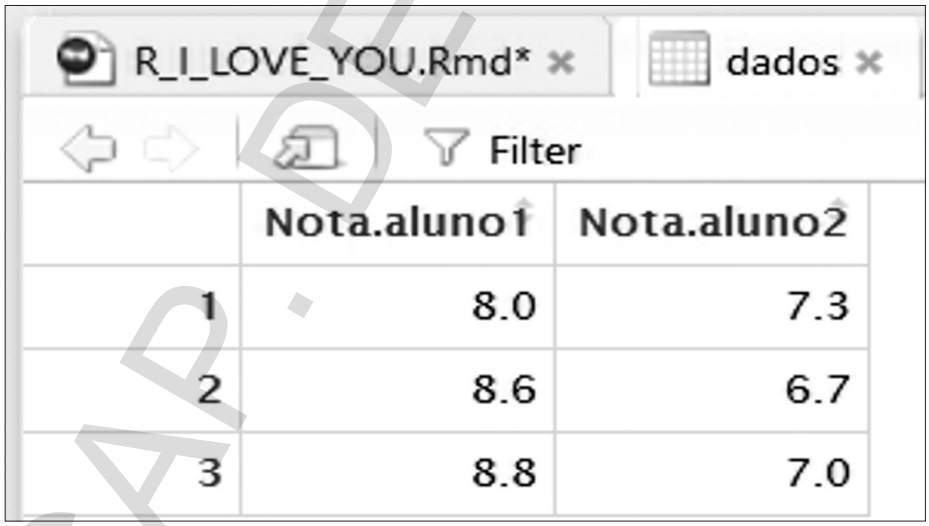
Trata-se de uma tabela com dados organizados por colunas onde cada coluna é um vetor que pode ser numérico ou de caractere.

```
#criamos uma tabela de nome notas contendo

#as notas dos dois alunos em cada coluna
notas = data.frame(Nota.aluno1, Nota.aluno2)

#consultando o conteúdo do objeto
notas

## Nota.aluno1 Nota.aluno2
## 1         8.0         7.3
## 2         8.6         6.7
## 3         8.8         7.0
```

Podemos também visualizar o conteúdo do objeto com o comando:

```
View(notas)
```

Nesse caso, abrirá uma janela com a tabela de dados como a que se apresenta na figura 5:



	Nota.aluno1	Nota.aluno2
1	8.0	7.3
2	8.6	6.7
3	8.8	7.0

Figura 5: visualizando o *dataframe*

Podemos ainda acrescentar uma coluna que identifique a que se refere cada linha:

```
#criamos o vetor Tipo com as referências das linhas
Tipo = c("Prova A", "Prova B", "Prova C")

#acrescenta o vetor Tipo ao objeto notas

notas = data.frame(notas, Tipo)
notas

## Nota.aluno1 Nota.aluno2 Tipo
## 1         8.0         7.3 Prova A
## 2         8.6         6.7 Prova B
## 3         8.8         7.0 Prova C
```

A coluna Tipo agora identifica a que se refere cada nota dos alunos 1 e 2.

Matrizes

A diferença entre a matriz e a tabela de dados é que no primeiro caso as linhas ou colunas são formadas por vetores exclusivamente numéricos ou exclusivamente de caracteres. Já na tabela de dados, podemos ter os dois tipos misturados, ou seja, podemos ter uma coluna de caracteres e outra numérica.

```
#cria uma matriz por colunas
Mc = cbind(Nota.aluno1, Nota.aluno2)
Mc

## Nota.aluno1 Nota.aluno2
## [1,]         8.0         7.3
## [2,]         8.6         6.7
## [3,]         8.8         7.0

#cria uma matriz por linhas
Ml = rbind(Nota.aluno1, Nota.aluno2)
Ml

## [,1] [,2] [,3]
## Nota.aluno1  8.0  8.6  8.8
## Nota.aluno2  7.3  6.7  7.0
```

Não se pode misturar na matriz elementos de natureza distinta. Por exemplo: as notas são de natureza numérica, enquanto que tipo é de natureza não numérica. Ao misturar esses objetos em uma matriz, o resultado é uma matriz *não numérica*. O correto, então, é armazenar como *data.frame*!

```
#cria uma matriz por colunas misturando vetor numérico com
caractere
M = cbind(Nota.aluno1, Nota.aluno2, Tipo)
M

## Nota.aluno1 Nota.aluno2 Tipo
## [1,] "8"         "7.3"       "Prova A"
## [2,] "8.6"       "6.7"       "Prova B"
## [3,] "8.8"       "7"         "Prova C"

#analisando a estrutura do objeto M
str(M)



##  chr [1:3, 1:3] "8" "8.6" "8.8" "7.3" "6.7" "7" "Prova A"
"Prova B" ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:3] "Nota.aluno1" "Nota.aluno2" "Tipo"
```

Nesse caso, a matriz M foi interpretada como caractere em todas as colunas. Percebemos isso, pois os números aparecem entre aspas.

Acessando valores de posições específicas dos objetos

Para acessar uma posição de um objeto tipo vetor ou matriz, devemos informar a posição desejada entre *colchetes*. No caso de um *dataframe*, adicionalmente é possível informar o nome da coluna desejada, precedida do símbolo $ através do comando ***nomedodataframe$nomedacoluna***.

```
#nota do aluno 1 na posição 2 do vetor
Nota.aluno1[2]

## [1] 8.6

#valor armazenado na posição 2 do objeto Tipo
Tipo[2]

## [1] "Prova B"

#valor da linha 1 e coluna 2 da matriz Mc
Mc[1, 2]

## Nota.aluno2
##         7.3

#valor da linha 2 e coluna 3 da matriz Mc
Ml[2, 3]

## Nota.aluno2
##           7



#todos os valores da coluna 2
notas[ , 2]

## [1] 7.3 6.7 7.0

#todos os valores da linha 2
notas[2, ]

## Nota.aluno1 Nota.aluno2 Tipo
## 2         8.6         6.7 Prova B

#valores do vetor Nota.aluno2 contido no objeto notas
notas$Nota.aluno2

## [1] 7.3 6.7 7.0
```

Funções estatísticas básicas

Todas as funções no R apresentam a seguinte estrutura:
nomedafunção(argumentos separados por vírgula)

Existem muitas funções disponíveis, neste momento iremos listar uma pequena seleção de funções para realizar tarefas simples com números, vetores e matrizes:

- *apply*(*D, i, f*), retorna os valores resultantes da aplicação da função f ao objeto D, considerando a aplicação da função nas linhas i=1 ou nas colunas i=2;
- *c*(*valor1, valor2, …, valor n*), concatena uma sequência de valores seja numérico ou de caractere. Neste último caso os valores devem estar entre aspas;
- *cbind*(*x1, x2,…, xn*), cria uma matriz com n colunas formada pelos vetores *x1*, *x2*, …, *xn*;
- *ceiling*(*x*), retorna o menor inteiro maior ou igual ao valor *x*;
- *cor(x, y)*, calcula o coeficiente de correlação;
- *cumsum*(*x*), *cumprod*(*x*), *cummin*(*x*) ou *cummax*(*x*), retorna um vetor com valores acumulados em soma, produto, mínimo ou máximo sobre os elementos de *x*;
- *data.frame*(*x1, x2, …, xn*), cria um dataframe com os vetores *x1*, *x2*, …, *xn*;
- *det*(*M*), calcula o determinante da matriz quadrada M;
- *dim*(*M*), retorna as dimensões do objeto M;
- *diff*(*x*), retorna um vetor com a diferença entre os valores de *x*;
- *eigen*(*M*), retorna os autovalores e autovetores da matriz quadrada M;
- *floor(x)*, retorna o maior inteiro menor ou igual a *x*;
- *identical(x, y)*, verifica se os vetores são idênticos;
- *intersect(x, y),* realiza a interseção de dois conjuntos;
- *head(D)*, mostra o cabeçalho do objeto D;
- *length(x)*, calcula o comprimento do vetor *x*;
- *mean(x)*, calcula a média do vetor *x*;

- *median*(*x*), calcula a mediana do vetor *x*;
- *min*(*x*) ou *max*(*x*), calcula o mínimo ou o máximo de *x*;
- *ncol*(*M*) ou *nrow*(*M*), retorna o número de colunas ou linhas da matriz M;
- *polyroot*(*x*), encontra as raízes do polinômio de ordem n cujos coeficientes são representados no vetor x em ordem decrescente;
- *prod*(*x*), multiplica os valores de *x*;
- *quantile(x, k)*, calcula o percentil de ordem $0 \leq k \leq 1$ dos valores de *x*;
- *Re*(*x*), retorna a parte real de um vetor *x*;
- *rep(x, k),* cria um vetor repetindo a sequência *x* k vezes;
- *round(x, k)*, arredonda o valor *x* com k casas decimais;
- *sd*(*x*), calcula o desvio-padrão do vetor *x*;
- *seq(i, j, k)*, cria uma sequência de i até j com tamanho de passo k;
- *setdiff(x, y),* retorna um vetor contendo os elementos do conjunto diferença entre *x* e *y*;
- *setequal(x, y)*, verifica se os elementos dos vetores *x* e *y* são iguais, independentemente da frequência em que aparecem no vetor;
- *solve*(*A, b*), resolve A*x* = b, retornando *x*;
- *sort*(*x*), ordena os valores do vetor *x* em ordem crescente;
- *sort*(*x, decreasing* = *T*), ordena os valores de *x* em ordem decrescente;
- *str*(*D*), retorna a estrutura do objeto D;

- *sum(x)*, soma os valores de *x*;
- *union(x, y)*, retorna os elementos da união entre *x* e *y*;
- *var*(*x*) ou *var(x, y)*, calcula a variância do vetor *x* ou a covariância entre *x* e *y*;
- *View(D)*, mostra o dataframe em janela separada.

EXERCÍCIOS DE FIXAÇÃO PARA O APRENDIZADO

Realize este exercício e veja como é simples utilizar as funções que selecionamos anteriormente:

|1| Crie três vetores *x*, *y* e *z* e uma matriz quadrada com esses vetores em colunas, sendo:

$$x = [1,2,2]\, y = [\frac{1}{2}, 1,1]\, z = [\frac{1}{2}, 1,1]$$

$$M = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 2 & 1 & 1 \\ 2 & 1 & 1 \end{pmatrix}$$

|2| Calcule a média e a mediana de *x*.

|3| Calcule a correlação entre *x* e seu vetor de soma acumulada.

|4| Obtenha as dimensões da matriz M.

|5| Obtenha os autovalores e autovetores da matriz *M*.

|6| Obtenha o piso e o teto do valor 8.799.

|7| Arredonde o valor 8.799 para uma casa decimal.

|8| Encontre as raízes do polinômio $x^2 - 9$, retornado as raízes reais.

|9| Retorne as diferenças entre os elementos consecutivos do vetor *x*.

| 10 | Obtenha o vetor de somas acumuladas do vetor *z*, o vetor do produto acumulado do vetor *y* e o vetor do valor máximo acumulado do vetor *x*.

| 11 | Obtenha o desvio-padrão e a variância de *x*.

| 12 | Crie a matriz *H* e obtenha a média dos elementos de cada linha usando a função *apply(H, i=2, mean)*.

$$H = \begin{pmatrix} 0 & 8 & 3 \\ 4 & 1 & 0 \\ 3 & 5 & 1 \end{pmatrix}$$

| 13 | Crie os seguintes vetores:

```
X=c(3, 8, 1, 2.5)
Y=c(8, 0, 2)
```

Obtenha o resultado da união e da intersecção entre *X* e *Y*.