

R Base

Folha de Consultas

Obtendo Ajuda

Acessando arquivos de ajuda

?mean
Ver o *help* de uma função.
help.search('weighted mean')
Pesquisar o *help* por meio duma palavra ou frase.
help(package = 'dplyr')
Encontrar o *help* de um pacote.

Mais sobre um objeto

str(iris)
Obter um resumo da estrutura de um objeto.
class(iris)
Saber a classe de um objeto.

Usando Pacotes

install.packages('dplyr')
Baixar e instalar um pacote do CRAN.
library(dplyr)
Carregar o pacote para dentro da sessão, disponibilizando todas as suas funções.
dplyr::select
Usar uma função específica de um pacote.
data(iris)
Carregar um conjunto de dados interno para dentro da sessão.

Diretório de trabalho

getwd()
Encontrar o diretório de trabalho atual (onde estão as entradas e aonde vão as saídas).
setwd('C://local/arquivo')
Mudar o diretório de trabalho atual.
Você pode usar o RStudio para escolher seu diretório de trabalho.

Vetores

Criando vetores

c(2, 4, 6)	2 4 6	Juntar elementos num vetor
2:6	2 3 4 5 6	Uma sequência de números inteiros
seq(2, 3, by=0.5)	2.0 2.5 3.0	Uma sequência customizada
rep(1:2, times=3)	1 2 1 2 1 2	Repetir um vetor
rep(1:2, each=3)	1 1 1 2 2 2	Repetir elementos de um vetor

Funções Vetoriais

sort(x) Retornar x ordenado.	rev(x) Retornar x revertido.
table(x) Contagens de valores.	unique(x) Ver valores únicos.

Selecionando Elementos de um Vetor

Por Posição

x[4]	O quarto elemento.
x[-4]	Todos exceto o quarto.
x[2:4]	Elementos de dois a quatro.
x[-(2:4)]	Todos os elementos exceto de dois a quatro.
x[c(1, 5)]	Elementos um e cinco.

Por Valor

x[x == 10]	Elementos que são iguais a 10.
x[x < 0]	Todos os elementos menores que zero.
x[x %in% c(1, 2, 5)]	Elementos que pertencem ao conjunto {1, 2, 5}.

Vetores com nomes

x['cacau']	Elemento de nome 'cacau'.
-------------------	---------------------------

Programação

Repetição “para”

```
for (variavel in sequencia){  
  Fazer algo  
}
```

Exemplo

```
for (i in 1:4){  
  j <- i + 10  
  print(j)  
}
```

Expressões condicionais

```
if (condicao){  
  Fazer algo  
} else {  
  Fazer algo diferente  
}
```

Exemplo

```
if (i > 3){  
  print('Sim')  
} else {  
  print('Nao')  
}
```

Repetição “enquanto”

```
while (condicao){  
  Fazer algo  
}
```

Exemplo

```
while (i < 5){  
  print(i)  
  i <- i + 1  
}
```

Funções

```
nome_da_funcao <- function(var){  
  Fazer algo  
  return(variavel_nova)  
}
```

Exemplo

```
quadrado <- function(x){  
  aoquad <- x*x  
  return(quadrado)  
}
```

Lendo e Escrevendo Dados

Ver também o pacote **readr**.

Entrada	Saída	Descrição
df <- read.table('arquivo.txt')	write.table(df, 'arquivo.txt')	Ler e escrever um arquivo de texto delimitado.
df <- read.csv('arquivo.csv')	write.csv(df, 'arquivo.csv')	Ler e escrever um arquivo de valores separados por vírgula. Este é um caso especial de read.table/write.table.
load('arquivo.Rdata')	save(df, file = 'file.Rdata')	Ler e escrever um arquivo de dados R, um tipo de arquivo especial para R.

Condições	a == b	São iguais	a > b	maior que	a >= b	maior que ou igual a	is.na(a)	É faltante
	a != b	Não iguais	a < b	menor que	a <= b	menor que ou igual a	is.null(a)	É nulo

Tipos

Convertendo entre tipos corriqueiros de dados em R. Pode-se sempre ir de um valor mais alto na tabela a um valor mais baixo.

<code>as.logical</code>	TRUE, FALSE, TRUE	Valores booleanos (VERDADEIRO ou FALSO).
<code>as.numeric</code>	1, 0, 1	Números inteiros ou de ponto flutuante.
<code>as.character</code>	'1', '0', '1'	Cadeias de caracteres. Geralmente preferíveis a fatores.
<code>as.factor</code>	'1', '0', '1', levels: '1', '0'	Cadeias de caracteres com níveis pré-estabelecidos. Necessário para alguns modelos estatísticos.

Funções Matemáticas

<code>log(x)</code>	Logaritmo natural.	<code>sum(x)</code>	Soma.
<code>exp(x)</code>	Função exponencial.	<code>mean(x)</code>	Média.
<code>max(x)</code>	Maior elemento.	<code>median(x)</code>	Mediana.
<code>min(x)</code>	Menor elemento.	<code>quantile(x)</code>	Quantis.
<code>round(x, n)</code>	Arredondar para n casas decimais.	<code>rank(x)</code>	Postos dos elementos.
<code>signif(x, n)</code>	Arredondar para n algarismos significativos.	<code>var(x)</code>	A variância.
<code>cor(x, y)</code>	Correlação.	<code>sd(x)</code>	O desvio padrão.

Atribuições de variáveis

```
> a <- 'animal'
> a
[1] 'animal'
```

O Ambiente

<code>ls()</code>	Listar todas as variáveis no ambiente.
<code>rm(x)</code>	Remover x do ambiente.
<code>rm(list = ls())</code>	Remover todas as variáveis do ambiente.

Você pode usar o painel *environment* no Rstudio para ver as variáveis que estão em seu ambiente.

Matrizes

```
m <- matrix(x, nrow = 3, ncol = 3)
Criar uma matriz a partir de x.
```



`m[2,]` - Selecionar uma linha



`m[, 1]` - Selecionar uma coluna



`m[2, 3]` - Selecionar um elemento

`t(m)`

Transpor

`m %*% n`

Multiplicação de Matrizes

`solve(m, n)`

Encontrar x em: $m \cdot x = n$

Listas

```
l <- list(x = 1:5, y = c('a', 'b'))
```

Uma lista é uma coleção de elementos que podem ser de diferentes tipos.

`l[[2]]`

Segundo elemento de l.

`l[1]`

Nova lista somente com o primeiro elemento.

`l$x`

Elemento chamado x.

`l['y']`

Nova lista somente com o elemento chamado y.

Ver também o pacote **dplyr**.

Data Frames

```
df <- data.frame(x = 1:3, y = c('a', 'b', 'c'))
Um caso especial de lista onde todos elementos têm o mesmo tamanho.
```

x	y
1	a
2	b
3	c

Subagrupar matrizes

`df[, 2]`



`df[2,]`



`df[2, 2]`



Subagrupar listas

`df$x`



`df[[2]]`



Entendendo um "data frame"

`View(df)`

Ver o data frame inteiro.

`head(df)`

Ver as 6 primeiras linhas.

`nrow(df)`

Número de linhas.

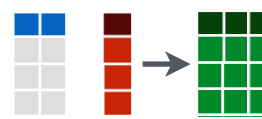
`ncol(df)`

Número de colunas.

`dim(df)`

Número de linhas e colunas.

cbind - Concatenar colunas.



rbind - Concatenar linhas.



Strings

Ver também o pacote **stringr**.

`paste(x, y, sep = ' ')` Juntar múltiplos vetores.

`paste(x, collapse = ' ')` Juntar elementos de um vetor.

`grep(pattern, x)` Encontrar expressões regulares em x.

`gsub(pattern, replace, x)` Substituir ocorrência em x por um *string*.

`toupper(x)` Converter para caixa alta.

`tolower(x)` Converter para caixa baixa.

`nchar(x)` Número de caracteres em um *string*.

Fatores

`factor(x)`

Tornar vetor em fator. Can set the levels of the factor and the order.

`cut(x, breaks = 4)`

Tornar um vetor numérico em um fator 'cortando'-o em seções.

Estatística

`lm(y ~ x, data=df)`
Modelo linear.

`glm(y ~ x, data=df)`
Modelo linear generalizado.

`summary`

Obter informações mais detalhadas de um modelo.

`t.test(x, y)`
Fazer um teste t para diferença entre médias.

`pairwise.t.test`
Fazer um teste t para dados pareados.

`prop.test`
Testar a diferença de duas proporções.

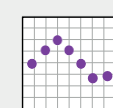
`aov`
Análise de variância.

Distribuições

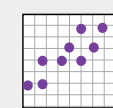
	Variável aleatória	Função densidade	Distribuição acumulada	Quantil
Normal	<code>rnorm</code>	<code>dnorm</code>	<code>pnorm</code>	<code>qnorm</code>
Poisson	<code>rpois</code>	<code>dpois</code>	<code>ppois</code>	<code>qpois</code>
Binomial	<code>rbinom</code>	<code>dbinom</code>	<code>pbinom</code>	<code>qbinom</code>
Uniforme	<code>runif</code>	<code>dunif</code>	<code>punif</code>	<code>qunif</code>

Gráficos

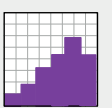
Ver também o pacote **ggplot2**.



`plot(x)`
Valores de x ordenados.



`plot(x, y)`
Valores de x contra y.



`hist(x)`
Histograma de x.

Datas

Ver o pacote **lubridate**.