

# Exercícios R básico

## Problema 1.

Como seria possível ler a matriz identidade de dimensão 2? Sugestão: Use o comando array

### Solução

```
mat1 <- array(c(1,0,0,1), dim = c(2,2)); mat1
```

```
##      [,1] [,2]  
## [1,]    1    0  
## [2,]    0    1
```

## Problema 2.

Usando o comando data.frame(), entre com a matriz de dados peso=(60,70,80) e altura=(160,150,170).

### Solução

```
altura <- c(160,150,170)  
peso <- c(60,70,80)  
nome <- c("Ari", "Carla", "Vitor")  
df1 <- data.frame(altura = altura, peso = peso, row.names = nome); df1
```

```
##      altura peso  
## Ari      160   60  
## Carla    150   70  
## Vitor    170   80
```

## Problema 3.

Dada uma lista com a nota de cinco alunos, por exemplo, {3,7,9,6,7} encontre a média, o desvio padrão e ordene as notas. Sugestão: usar as funções mean, sd, sort no vetor definido.

### Solução

```
nota <- c(3, 7, 9, 6, 7)  
mean(nota); sd(nota); sort(nota)
```

```
## [1] 6.4  
## [1] 2.19089  
## [1] 3 6 7 7 9
```

## Problema 4.

Dada uma lista com a nota de 10 alunos de duas turmas, por exemplo, {3,7,9,6,7,6,8,9,4,9} e {3,6,9,6,7,7,8,8,4,8}. Encontre um algoritmo que calcule quantos alunos possuem a mesma nota.

### Solução

Uma possível solução seria

```
nota_turma1 <- c(3,7,9,6,7,9,8,9,4,5) # vetor com as notas da primeira turma
nota_turma2 <- c(3,6,9,6,7,7,8,8,4,9) # vetor com as notas da segunda turma
nota_turma <- c(nota_turma1, nota_turma2)
contador <- 0
# queremos primeiro extrair quais notas estão repetidas
notas <- sort(unique(nota_turma)) # valores sem repetição ordenados
for (i in 1:length(notas)) {
  for (j in 1:length(nota_turma)) {
    if (notas[i] == nota_turma[j]) contador <- contador + 1
  }
  cat(sprintf("Alunos com nota %d: %d, \n", notas[i], contador))
  contador <- 0
}
```

```
## Alunos com nota 3: 2,
## Alunos com nota 4: 2,
## Alunos com nota 5: 1,
## Alunos com nota 6: 3,
## Alunos com nota 7: 4,
## Alunos com nota 8: 3,
## Alunos com nota 9: 5,
```

## Problema 5.

Fazer um algoritmo para calcular e imprimir as 10 primeiras potências de 3

### Solução

```
for (i in 1:10) {
  cat(sprintf("%da potência de 3 = %.0f\n", i, 3^i))
}
```

```
## 1a potência de 3 = 3
## 2a potência de 3 = 9
## 3a potência de 3 = 27
## 4a potência de 3 = 81
## 5a potência de 3 = 243
## 6a potência de 3 = 729
## 7a potência de 3 = 2187
## 8a potência de 3 = 6561
## 9a potência de 3 = 19683
## 10a potência de 3 = 59049
```

## Problema 6.

Usar o comando for para calcular a soma dos 10 primeiros termos de

$$e^x = \sum_{i=0}^{\infty} \frac{x^i}{i!}$$

com  $x = 2$

### Solução

```
x <- 2
resp <- 1 # fatorial de 0 é 1
fat <- 1
for (i in 1:10) {
  fat <- fat*i
  resp <- resp + x^i/fat
}
resp
```

```
## [1] 7.388995
```

## Problema 7.

Escreva um laço que calcula o fatorial de um inteiro n. Sugestão: usar o while.

### Solução

```
n <- scan("", what = integer())
k <- 1
fatorial <- 1 # fatorial de 0 é 1
while(k < n){
  k <- k + 1
  fatorial <- fatorial*k
}
fatorial
```

## Problema 8.

Um número é, por definição, primo se ele não tem divisores, exceto 1 e ele próprio. Prepara um algoritmo para ler um número e verificar se este é primo ou não.

### Solução

```
n <- scan("", what = integer())
k <- 2
resp <- 0
while (k<n/2) {
```

```

if (n%%k == 0){
  cat(sprintf("O número %d é divisível por %d\n", n, k))
  resp <- 1
  break()
}
k <- k + 1
}
if(resp == 0) cat(sprintf("O número %d é primo\n", n))

```

## Funções Escritas pelo Usuário

O R permite que o usuário escreva funções de seu interesse. A definição geral de uma função é dada por  
 nome <-função(arg1, arg2, ...) expressão

### Exemplo 1.

A função abaixo calcula o fatorial do número n

```

fat<-function(n){
  fat<-1
  while(n!=1){
    fat<-fat*n
    n<-n-1 }
  fat}

```

### Exemplo 2.

Escreva uma função para calcular

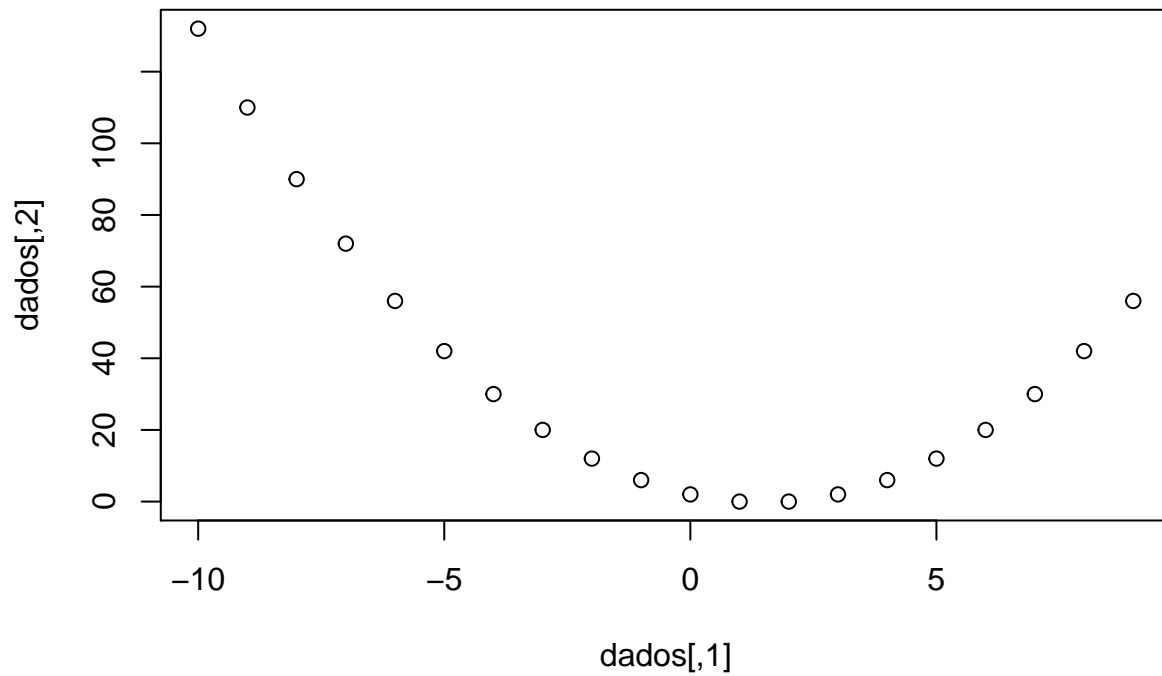
$$f(x) = x^2 - 3x + 2$$

Atribua uma sequência de 20 valores e faça o gráfico desta função.

```

f<-function(x) {x^2-3*x+2}
dados<-matrix(0,20,2)
dados[,1]<- -10:9
for(i in 1:20)
{
  dados[i,2]<-f(dados[i,1])
}
plot(dados)

```



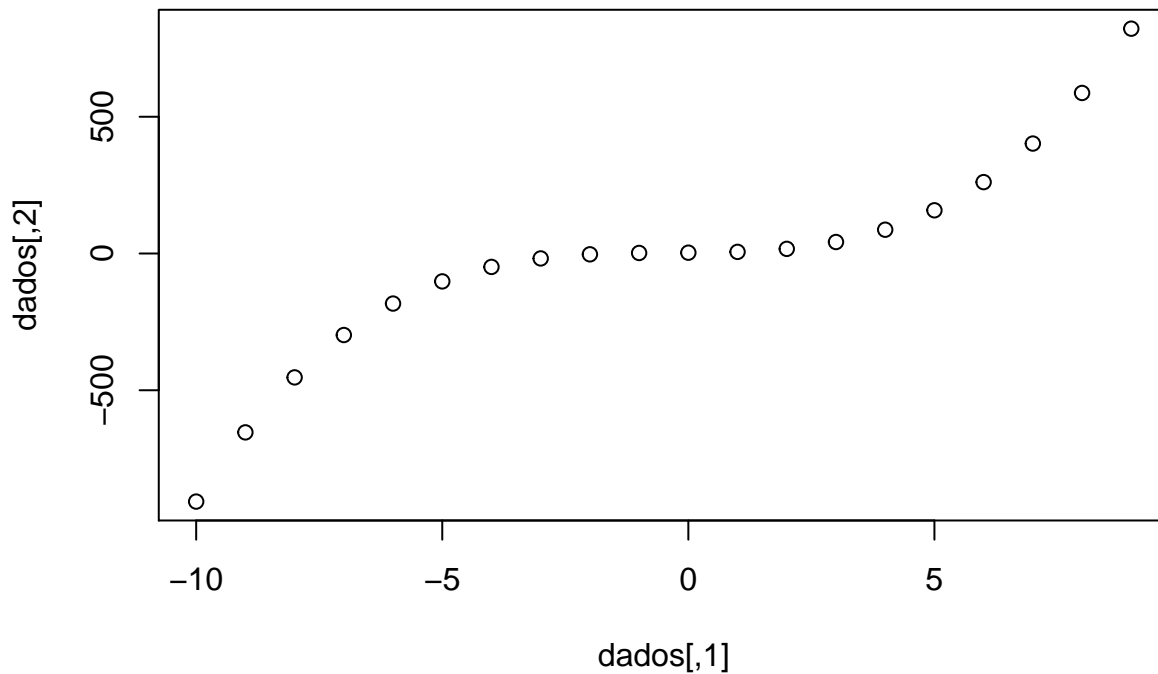
### Problema 9.

Faça o gráfico do exemplo anterior com a função

$$f(x) = x^3 + x^2 + x + 3$$

### Solução

```
f<-function(x) {x^3+x^2+x+3}
dados<-matrix(0,20,2)
dados[,1]<- -10:9
for(i in 1:20)
{
  dados[i,2]<-f(dados[i,1])
}
plot(dados)
```



### Problema 10.

Escreva funções para calcular a média e o desvio padrão de um vetor de valores. Compare os resultados de suas funções com aqueles resultados obtidos pelo R.

### Solução

```
f <- function(v){
  media <- 0
  for (i in 1:length(v)) {
    media <- media + v[i]/length(v)
  }
  variancia <- 0
  for (i in 1:length(v)) {
    variancia <- (variancia + (v[i]-media)^2)/(length(v)-1)
  }
  desvio <- sqrt(variancia)
  cat(sprintf("A média do vetor é %f e seu desvio padrão é %f\n", media, desvio))
}
v <- scan("")
f(v)
```

### Problema 11.

Escreva uma função que calcula o produtório.

## Solução

```
f <- function(n){  
  p <- 1  
  k <- 1  
  while (k<=n) {  
    p <- p * k  
    k <- k + 1  
  }  
  p  
}  
n <- scan("", what = integer())  
cat(sprintf("O produto de %d é %d\n", n, f(n)))
```

## Problema 12.

Escreva uma função que calcula o valor de  $\pi$

$$\pi = \sqrt{\sum_{i=1}^{\infty} \frac{6}{i^2}}$$

## Solução

```
f <- function(n){  
  f <- 0  
  k <- 1  
  while (k <= n) {  
    f <-f + 6/k^2  
    k <- k+1  
  }  
  f  
}  
resp <- sqrt(f(1000)); resp
```

```
## [1] 3.140638
```