

R version 4.4.2 (2024-10-31 ucrt) -- "Pile of Leaves"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

R é um software livre e vem sem GARANTIA ALGUMA.
Você pode redistribuí-lo sob certas circunstâncias.
Digite 'license()' ou 'licence()' para detalhes de distribuição.

R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e
'citation()' para saber como citar o R ou pacotes do R em publicações.

Digite 'demo()' para demonstrações, 'help()' para o sistema on-line de ajuda,
ou 'help.start()' para abrir o sistema de ajuda em HTML no seu navegador.
Digite 'q()' para sair do R.

[Área de trabalho anterior carregada]

```
> # >>>> Conceitos iniciais de R <<<<<
>
> # >>>> Escrevendo principais conceitos/comandos <<<<
>
> # Atribuição de valores
> a=10
> a
[1] 10
> a<-20
> a
[1] 20
>
> # Limpando a memória
> rm(list=ls())
> # Teste
> a
Erro: objeto 'a' não encontrado
>
> # Operações básicas
> 10-20
[1] -10
> 10+20
[1] 30
> 10/20
[1] 0.5
> 0.5*10
[1] 5
> x<-10
> x^2+2*x-500
[1] -380
>
> # Especificando funções
> # Definindo
> quad<-function(x,a,b,c){a*x^2+b*x+c}
> # Avaliando
> quad(30,1,3,10)
[1] 1000
>
> # Vetores
> x=c(1,"a",0.5,"nome",5,6)
> x
[1] "1"      "a"      "0.5"    "nome"   "5"      "6"
>
> # Matrizes
> y=matrix(c(1,4,3,4,3,6,7,1,9),nr=3)
> y
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    4    3    1
[3,]    3    6    9
> det(y)
[1] -6
> # Inversão
> solve(y)
     [,1] [,2] [,3]
[1,]  1.0  0.0  0.0
[2,]  0.0  1.0  0.0
[3,]  0.0  0.0  1.0
```

```

[1,] -3.5    -1    2.833333
[2,]  5.5     2   -4.500000
[3,] -2.5    -1    2.166667
>
> # Gráfico da função
> x=c(1,2,3,4,5,6,7,8,9,10)
> y=quad(x,1,3,10)
> plot(x,y)
> save.image("D:\\vipgu\\Documents\\GitHub\\portifolio-R-DataScience\\1.GraficoDaFuncao")
>
> # Sequências
> 1:100
 [1]  1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18
[19] 19   20   21   22   23   24   25   26   27   28   29   30   31   32   33   34   35   36
[37] 37   38   39   40   41   42   43   44   45   46   47   48   49   50   51   52   53   54
[55] 55   56   57   58   59   60   61   62   63   64   65   66   67   68   69   70   71   72
[73] 73   74   75   76   77   78   79   80   81   82   83   84   85   86   87   88   89   90
[91] 91   92   93   94   95   96   97   98   99  100
> -10:10
 [1] -10   -9   -8   -7   -6   -5   -4   -3   -2   -1    0    1    2    3    4    5    6    7    8
[20]  9   10
> seq(-2,9,by=0.1)
 [1] -2.0 -1.9 -1.8 -1.7 -1.6 -1.5 -1.4 -1.3 -1.2 -1.1 -1.0 -0.9 -0.8 -0.7 -0.6
[16] -0.5 -0.4 -0.3 -0.2 -0.1  0.0  0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9
[31]  1.0  1.1  1.2  1.3  1.4  1.5  1.6  1.7  1.8  1.9  2.0  2.1  2.2  2.3  2.4
[46]  2.5  2.6  2.7  2.8  2.9  3.0  3.1  3.2  3.3  3.4  3.5  3.6  3.7  3.8  3.9
[61]  4.0  4.1  4.2  4.3  4.4  4.5  4.6  4.7  4.8  4.9  5.0  5.1  5.2  5.3  5.4
[76]  5.5  5.6  5.7  5.8  5.9  6.0  6.1  6.2  6.3  6.4  6.5  6.6  6.7  6.8  6.9
[91]  7.0  7.1  7.2  7.3  7.4  7.5  7.6  7.7  7.8  7.9  8.0  8.1  8.2  8.3  8.4
[106] 8.5  8.6  8.7  8.8  8.9  9.0
> seq(-2,9,length=200)
 [1] -2.000000000 -1.94472362 -1.88944724 -1.83417085 -1.77889447 -1.72361809
 [7] -1.66834171 -1.61306533 -1.55778894 -1.50251256 -1.44723618 -1.39195980
[13] -1.33668342 -1.28140704 -1.22613065 -1.17085427 -1.11557789 -1.06030151
[19] -1.00502513 -0.94974874 -0.89447236 -0.83919598 -0.78391960 -0.72864322
[25] -0.67336683 -0.61809045 -0.56281407 -0.50753769 -0.45226131 -0.39698492
[31] -0.34170854 -0.28643216 -0.23115578 -0.17587940 -0.12060302 -0.06532663
[37] -0.01005025  0.04522613  0.10050251  0.15577889  0.21105528  0.26633166
[43]  0.32160804  0.37688442  0.43216080  0.48743719  0.54271357  0.59798995
[49]  0.65326633  0.70854271  0.76381910  0.81909548  0.87437186  0.92964824
[55]  0.98492462  1.04020101  1.09547739  1.15075377  1.20603015  1.26130653
[61]  1.31658291  1.37185930  1.42713568  1.48241206  1.53768844  1.59296482
[67]  1.64824121  1.70351759  1.75879397  1.81407035  1.86934673  1.92462312
[73]  1.97989950  2.03517588  2.09045226  2.14572864  2.20100503  2.25628141
[79]  2.31155779  2.36683417  2.42211055  2.47738693  2.53266332  2.58793970
[85]  2.64321608  2.69849246  2.75376884  2.80904523  2.86432161  2.91959799
[91]  2.97487437  3.03015075  3.08542714  3.14070352  3.19597990  3.25125628
[97]  3.30653266  3.36180905  3.41708543  3.47236181  3.52763819  3.58291457
[103] 3.63819095  3.69346734  3.74874372  3.80402010  3.85929648  3.91457286
[109] 3.96984925  4.02512563  4.08040201  4.13567839  4.19095477  4.24623116
[115] 4.30150754  4.35678392  4.41206030  4.46733668  4.52261307  4.57788945
[121] 4.63316583  4.68844221  4.74371859  4.79899497  4.85427136  4.90954774
[127] 4.96482412  5.02010050  5.07537688  5.13065327  5.18592965  5.24120603
[133] 5.29648241  5.35175879  5.40703518  5.46231156  5.51758794  5.57286432
[139] 5.62814070  5.68341709  5.73869347  5.79396985  5.84924623  5.90452261
[145] 5.95979899  6.01507538  6.07035176  6.12562814  6.18090452  6.23618090
[151] 6.29145729  6.34673367  6.40201005  6.45728643  6.51256281  6.56783920
[157] 6.62311558  6.67839196  6.73366834  6.78894472  6.84422111  6.89949749
[163] 6.95477387  7.01005025  7.06532663  7.12060302  7.17587940  7.23115578
[169] 7.28643216  7.34170854  7.39698492  7.45226131  7.50753769  7.56281407
[175] 7.61809045  7.67336683  7.72864322  7.78391960  7.83919598  7.89447236
[181] 7.94974874  8.00502513  8.06030151  8.11557789  8.17085427  8.22613065
[187] 8.28140704  8.33668342  8.39195980  8.44723618  8.50251256  8.55778894
[193] 8.61306533  8.66834171  8.72361809  8.77889447  8.83417085  8.88944724
[199] 8.94472362  9.00000000
>
> # Plot de sequência
> x<-seq(0,100,length=30)
> y=quad(x,1,3,10)
> plot(x[-29],y[-29],type='o',
+ lwd=2,cex=3,xlab="Valores de x",
+ ylab="Valores da função")
>
> # >>>> Comandos de controle de fluxo <<<<

```

```
>
> # If / else
>
> x=3
> if(x<=5){
+ print("x é menor ou igual a cinco")
+ } else{print("x é maior que cinco")}
[1] "x é menor ou igual a cinco"
>
> loga<-function(x){log(x-100)}
> x=200
> if(x>100){loga(x)} else{
+ print("o ponto não pertence ao domínio")}
[1] 4.60517
>
>
> # for
>
> soma=0
> for(i in 1:50){
+ soma=soma+i
+ }
> soma
[1] 1275
>
> soma=1
> for(i in 1:10){
+ soma=soma+1/(2^i)
+ }
> soma
[1] 1.999023
>
> # while
> soma=1
> i=0
> while(2-soma>0.01){
+ i=i+1
+ soma=soma+1/(2^i)
+ }
> i
[1] 7
> soma
[1] 1.992188
>
> # >>>> Análise exploratória <<<<
>
> # Inserir um conjunto de dados
> x<-c(0,0,0,0,0,0,3,0,0,0,0,0,1,0,0,0)
> x
[1] 0 0 0 0 0 3 0 0 0 0 1 0 0 0
>
> # Mediana
> median(x)
[1] 0
>
> # Média
> mean(x)
[1] 0.2857143
>
> # Variância
> var(x)
[1] 0.6813187
>
> # Desvio padrão
> sd(x)
[1] 0.8254203
> sqrt(var(x))
[1] 0.8254203
>
> # Resumo dos dados
> summary(x)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.0000  0.0000  0.0000  0.2857  0.0000  3.0000
>
```

```

> # valor em certa posição de um vetor (no caso, 6ª)
> x[6]
[1] 3
>
> # Soma dos valores em um vetor
> sum(x)
[1] 4
>
> # Ordenar um vetor
> sort(x)
[1] 0 0 0 0 0 0 0 0 0 0 0 0 0 1 3
>
> # tamanho do vetor
> length(x)
[1] 14
>
> # Função média
> media<-function(x) {
+ sum(x)/length(x) }
> media(x)
[1] 0.2857143
>
> # Programa para o cálculo de uma média amostral
> x=c(3,4,5,2,3,4,2,1)
> soma=sum(x)
> n=length(x)
> media=soma/n
> media
[1] 3
>
> # >>>> Gráficos <<<<
>
> # Carregamento dos dados
> data(iris)
> # Ver conjunto de dados: primeiras linhas
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4          0.2  setosa
2          4.9         3.0          1.4          0.2  setosa
3          4.7         3.2          1.3          0.2  setosa
4          4.6         3.1          1.5          0.2  setosa
5          5.0         3.6          1.4          0.2  setosa
6          5.4         3.9          1.7          0.4  setosa
> # Dimensão do conjunto de dados
> dim(iris)
[1] 150    5
> # Atribuindo (x) a primeira coluna de iris
> x<-iris[,1]
> x
 [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4 5.1
[19] 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5 4.9 5.0
[37] 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0 6.4 6.9 5.5
[55] 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8 6.2 5.6 5.9 6.1
[73] 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4 6.0 6.7 6.3 5.6 5.5
[91] 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8 7.1 6.3 6.5 7.6 4.9 7.3
[109] 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7 6.0 6.9 5.6 7.7 6.3 6.7 7.2
[127] 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8
[145] 6.7 6.7 6.3 6.5 6.2 5.9
> # Atribuindo (y) a segunda coluna de iris
> y<-iris[,2]
> y
 [1] 3.5 3.0 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 3.7 3.4 3.0 3.0 4.0 4.4 3.9 3.5
[19] 3.8 3.8 3.4 3.7 3.6 3.3 3.4 3.0 3.4 3.5 3.4 3.2 3.1 3.4 4.1 4.2 3.1 3.2
[37] 3.5 3.6 3.0 3.4 3.5 2.3 3.2 3.5 3.8 3.0 3.8 3.2 3.7 3.3 3.2 3.2 3.1 2.3
[55] 2.8 2.8 3.3 2.4 2.9 2.7 2.0 3.0 2.2 2.9 2.9 3.1 3.0 2.7 2.2 2.5 3.2 2.8
[73] 2.5 2.8 2.9 3.0 2.8 3.0 2.9 2.6 2.4 2.4 2.7 2.7 3.0 3.4 3.1 2.3 3.0 2.5
[91] 2.6 3.0 2.6 2.3 2.7 3.0 2.9 2.9 2.5 2.8 3.3 2.7 3.0 2.9 3.0 3.0 2.5 2.9
[109] 2.5 3.6 3.2 2.7 3.0 2.5 2.8 3.2 3.0 3.8 2.6 2.2 3.2 2.8 2.8 2.7 3.3 3.2
[127] 2.8 3.0 2.8 3.0 2.8 3.8 2.8 2.8 2.6 3.0 3.4 3.1 3.0 3.1 3.1 3.1 2.7 3.2
[145] 3.3 3.0 2.5 3.0 3.4 3.0
>
> # Histograma para x
> hist(x, nclass=12)
>

```

```

> # Histograma para y
> hist(y)
> # Diagrama de dispersão
> plot(x,y)
>
> # >>> Exemplo <<<
>
> # Dados aleatórios
> x<-c(3,6,7,12,18,20,21,29,35,36,40,48,50,60,72)
>
> mean(x)
[1] 30.46667
> var(x)*(14/15)
[1] 401.9822
> sqrt(var(x)*(14/15))
[1] 20.04949
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3.00   15.00   29.00   30.47   44.00   72.00
>
> # diagrama de caixa
> boxplot(x)
>
> # diagrama ramo-e-folhas
> stem(x, scale=0.5)

The decimal point is 2 digit(s) to the right of the |

 0 | 01112223444
 0 | 5567
>
> # >>>> Tipos de variáveis <<<<
>
> # Números decimais
> x=3.0
> # ponto flutuante
> class(x)
[1] "numeric"
>
> x=as.integer(3)
> # inteiro
> class(x)
[1] "integer"
>
> # Caracteres
> x="João"
>
> # Lógica
> y=3
> comp=(y<2)
> class(comp)
[1] "logical"
>
> # Número complexo
> sqrt(-1)
[1] NaN
Mensagem de aviso:
In sqrt(-1) : NaNs produzidos
> sqrt(-1+0i)
[1] 0+1i
>
> # >>>> Operações <<<<
>
> a=10; b=0.05; c=15.36
>
> # Operações numéricas
>
> a+b
[1] 10.05
> a-b
[1] 9.95
> a^b
[1] 1.122018

```

```
> b/c
[1] 0.003255208
> a*b
[1] 0.5
>
> # Operações relacionais
>
> # verificar se a é maior do que b
> a>b
[1] TRUE
>
> # verificar se a é maior ou igual a b
> a>=b
[1] TRUE
>
> # a é igual (exatamente) a b
> a==b
[1] FALSE
>
> # a ≠ diferente de b
> a!=b
[1] TRUE
>
> d="joao"
> d>a
[1] TRUE
>
> # Operadores lógicos: & == (e) // | == (ou)
> (a>b) & (b<c)
[1] TRUE
> (a>b) | (b>c)
[1] TRUE
> (a>b) | (b>c)
[1] TRUE
> (a>b) | (b>c)
[1] TRUE
> (a>b) | (b>c)
[1] TRUE
> (a>b) | (b>c)
[1] TRUE
>
```