

UTILIZAÇÃO DO R NO ENSINO DA ESTATÍSTICA BÁSICA I

Autores:

Carolina Valani Cavalcante *

Luciane Ferreira Alcoforado **

Sumário

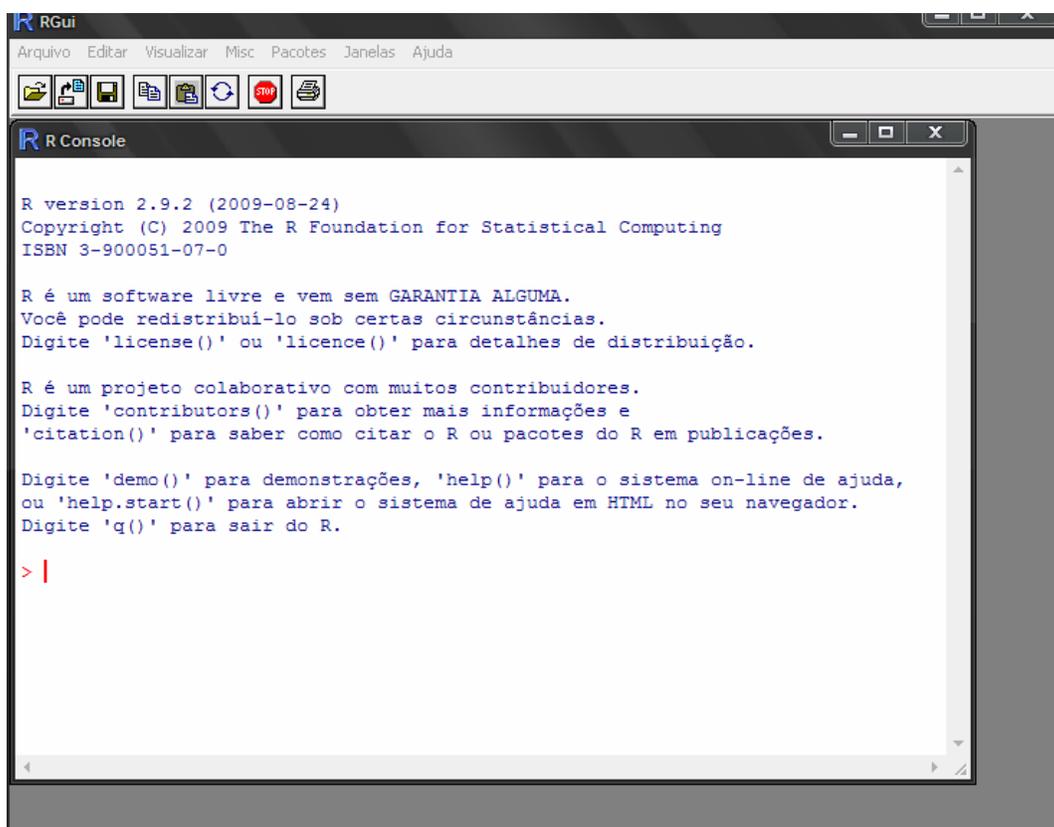
Autores:.....	1
Carolina Valani Cavalcante *.....	1
Luciane Ferreira Alcoforado **	1
1. Introdução.....	5
2. Operações Básicas.....	6
Exemplo 2.1.....	7
3. Armazenando dados	7
Exemplo 3.1.....	8
Exemplo 3.2.....	8
3.1 Vetor	8
Exemplo 3.3.....	9
Exemplo 3.4.....	9
Exemplo 3.6.....	11
3.2 Matriz.....	11
Exemplo 3.7.....	11
Exemplo 3.8.....	12
Exemplo 3.9.....	13
Exemplo 3.10.....	14
Exemplo 3.11.....	14
3.3 Data frame.....	14
Exemplo 3.12.....	15
Exemplo 3.13.....	16

4. Funções mais utilizadas	18
Exemplo 4.1	19
4.1 Usando a função <code>length(x)</code>	19
Exemplo 4.2	20
4.2 Usando as funções <code>min(x)</code> , <code>max(x)</code> e <code>range(x)</code>	20
Exemplo 4.3.....	20
4.3 Usando a função <code>sum(x)</code> e <code>prod(x)</code>	20
Exemplo 4.4.....	20
Exemplo 4.5.....	21
4.4 Usando a função <code>mean(x)</code>	21
Exemplo 4.6.....	21
Exemplo 4.7.....	21
4.5 Usando a função <code>median(x)</code>	21
Exemplo 4.8.....	22
Exemplo 4.9.....	22
4.6 Usando as funções <code>var(x)</code> e <code>sd(x)</code>	22
Exemplo 4.10.....	22
Exemplo 4.11.....	22
4.7 Usando a função <code>quantile(x,p)</code>	23
Exemplo 4.12.....	23
4.8 Usando a função <code>cor(x,y)</code> e <code>cov(x,y)</code>	23
Exemplo 4.13.....	23
4.9 Tabelas	23
4.10 Função <code>summary(x)</code>	24

5.	Gráficos	25
5.1	Dispersão.....	26
	Exemplo 5.1.....	26
	Exemplo5.2	27
5.2	Box-plot.....	28
	Exemplo 5.3.....	28
	Exemplo 5.4.....	29
	Exemplo 5.5.....	30
5.3	Colunas.....	31
	Exemplo 5.6.....	31
	Exemplo 5.7.....	32
5.4	Setores	33
	Exemplo 5.8.....	33
	Exemplo 5.9.....	34
5.5	Histograma.....	35
	Exemplo 5.10.....	35
	Exemplo 5.11.....	36
	Exemplo 5.12.....	37
	Exemplo 5.13.....	38
	Exemplo 5.14.....	39
	Exemplo 5.15.....	40

1. Introdução

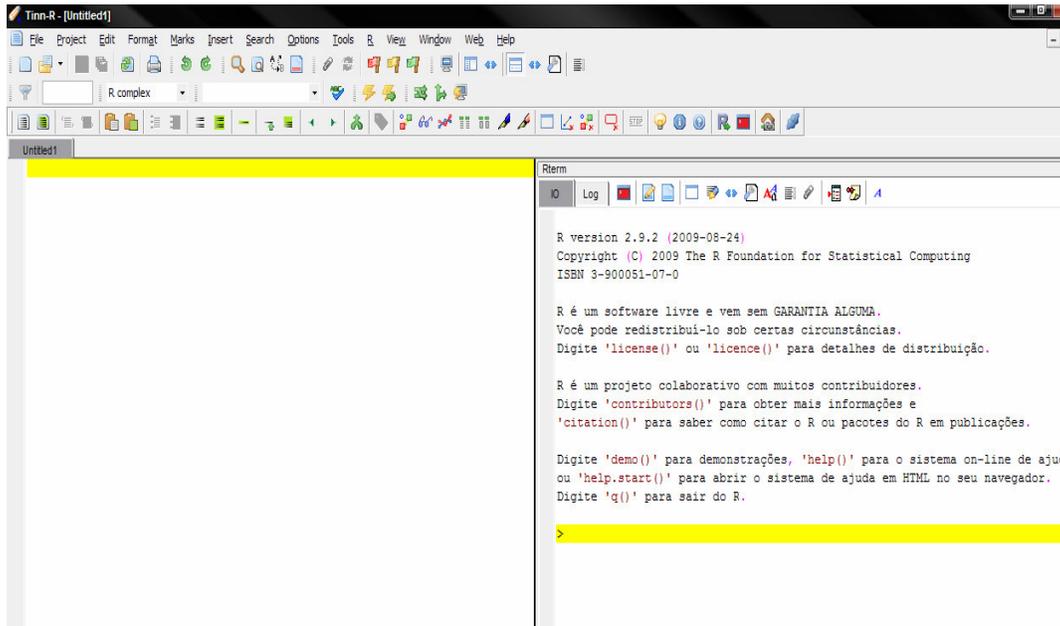
O R é um programa gratuito, de código aberto e livremente distribuído que proporciona análises estatísticas de alta qualidade. Pelo fato de ser usada uma linguagem de programação pode existir certa dificuldade até que se familiarize com os comandos mais comuns. Detalhes sobre o projeto, colaboradores, documentação e diversas outras informações podem ser encontradas na página oficial do projeto em: <http://www.r-project.org>.



Ao abrir o R no seu computador, é assim que o programa será apresentado.

Um outro tipo de R, sendo a única diferença sua forma de ser apresentado, é o Tinn-R. A principal diferença do Tinn-R para o R é que no Tinn-R, a janela de edição fica sempre aberta (lado esquerdo), impossibilitando o uso direto pela janela do R, já que a janela tem apenas a função de mostrar os resultados.

No caso do R, você tem a opção de abrir uma janela de edição, podendo sempre que quiser fazer alterações na própria janela do R. Abaixo está a principal imagem do Tinn-R, onde do lado esquerdo está a tela de edição, e do lado direito a tela do R.



Este material foi desenvolvido durante o projeto de monitoria de Métodos Computacionais I, com o objetivo de auxiliar o aprendizado do conteúdo abordado em Estatística Básica I.

Para que o iniciante no R se torne apto a colocar em prática o que foi apresentado durante as aulas, é preciso uma pequena introdução de como usar o R.

2. Operações Básicas

No R podemos resolver todas as operações básicas vistas frequentemente. São elas:

- Adição/Subtração [+/-]
- Multiplicação [*]
- Divisão [/]
- Potência [** ou ^]
- Raiz quadrada [sqrt()]
- Exponencial [exp(), log()]

obs.: A função log() no R, é equivalente ao log na base e. Caso queria calcular na base 10, basta colocar log10().

Abaixo segue um exemplo de cada uma das operações, utilizando o R:

Exemplo 2.1

```

> 5+4      #adição      > sqrt(121) #raiz
[1] 9
> 6-2      #subtração   > exp(0)   #exponencial
[1] 4
> 7*3      #multiplicação > log(1)   #log na base e
[1] 21
> 81/9     #divisão     > log10(1) #log na base 10
[1] 9
> 2^2      #potência
[1] 4

```

3. Armazenando dados

Existem quatro tipos básicos de dados no R, os numéricos, os de caracteres, os lógicos e os complexos. Os mais utilizados são os numéricos, que são dados compostos por números, e os de caracteres, que são dados compostos por letras ou palavras.

Quando os dados são armazenados, eles são chamados de objeto. Para armazenar um objeto, basta utilizar o símbolo “<-” ou “=”.

Supondo que um objeto tenha valor 4, ao digitarmos seu nome, é exibido seu valor. Segue o exemplo:

Exemplo 3.1

```
> x<-4      #O valor 4 é armazenado no objeto x
> x         #Exibe o valor do objeto
[1] 4
```

Algumas regras devem ser seguidas na hora de dar nome a um objeto. O nome do objeto precisa começar com uma letra, não pode ser um número, não pode conter símbolos referentes a funções ou nome de funções e a “seta” deve ser estar sempre apontada para o nome do objeto.

Abaixo segue alguns exemplos não válidos:

Exemplo 3.2

```
24e<-4     #O nome do objeto começa com números
12<-2      #O nome do objeto é um número
e*2<-      #O nome do objeto contém o símbolo da multiplicação
x->5       #A seta está para o lado errado
```

Os objetos podem ter várias formas, inclusive funções podem ser armazenadas como objeto. As formas mais utilizadas para armazenar dados são os vetores, matrizes e data-frames.

3.1 Vetor

O vetor é a forma mais simples de ter os dados armazenados. Pode ser um vetor numérico ou de caractere. A função `c()` é utilizada na criação do vetor. Sempre que armazenar um vetor de caractere é necessário colocar as aspas em palavra do vetor.

Exemplo 3.3

```
> x<-c(1,2,3,4)           #Vetor numérico de nome x

> x                       #Exibe o valor do vetor armazenado no objeto x
[1] 1 2 3 4

> w<-c("A","B","A","A")  #Vetor de caracteres de nome w

> w                       #Exibe o valor do vetor armazenado no objeto w
[1] "A" "B" "A" "A"
```

Supondo que queremos fazer algumas análises de uma seqüência de números, ou apenas de números pares, ou apenas de números primos, ou até mesmo de números repetidos. Para isso temos duas importantes funções a função `seq()` e a função `rep()`. A função `seq()` lista a seqüência de números que quiser, no intervalo que quiser. Já a função `rep()` lista números repetidos, quantos números quiser com quantas repetições quiser.

Exemplo 3.4

```
> seq(1,100,1) #Sequência de números de 1 até 100, com intervalo de 1 número
[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

> seq(1,100,2) #Sequência de números de 1 até 100, com intervalo de 2 números
[1] 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
[26] 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99

> seq(1,100,10) #Sequência de números de 1 até 100, com intervalo de 10
[1] 1 11 21 31 41 51 61 71 81 91

> seq(1,20,1) #Sequência crescente de 1 até 20
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

> seq(20,1,-1) #Sequência decrescente de 20 até 1
[1] 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
```

```

> rep(1,10) #Número 1 repetido 10 vezes
[1] 1 1 1 1 1 1 1 1 1 1

> rep(1:5,2) #Sequência de 1 até 5, repetida 2 vezes
[1] 1 2 3 4 5 1 2 3 4 5

> rep(c(1,4),3) #Números 1 e 4 repetidos 3 vezes, alternadamente
[1] 1 4 1 4 1 4

> c(rep(1,3),rep(4,3)) #Números 1 e 4 repetidos 3 vezes, sem alternar
[1] 1 1 1 4 4 4

```

Outra ferramenta que pode ser muito útil ao trabalhar com vetores é a de seleção de elementos dentro de um vetor. Para isso são utilizados os colchetes e indicados as posições ou os próprios elementos.

Exemplo 3.5

```

> x<-c(45,23,89,43)

> x[1] #Seleciona o primeiro elemento do vetor x
[1] 45

> x[4] #Seleciona o quarto elemento do vetor x
[1] 43

> x[x==23] #Seleciona o elemento que tem valor 3 dentro do vetor x
[1] 23

> x[c(1,2,3)] #Seleciona os elementos que estão na primeira, segunda e terceira posição no vetor x
[1] 45 23 89

> x[-c(2,3)] #Seleciona todos os elementos, menos o da segunda e terceira posição
[1] 45 43

> x[x==max(x)] #Seleciona o maior elemento do vetor x
[1] 89

```

Podemos aplicar as operações básicas nos vetores também. Ao somar, subtrair, dividir ou multiplicar dois vetores (numéricos) observamos que cada “coordenada” de um vetor é somada, subtraída, dividida ou multiplicada pela “coordenada” correspondente do outro vetor. Por isso soma e multiplicação de vetores não pode ser confundida com somatório e produtório de vetores. Veja o exemplo 1.

Exemplo 3.6

```
> x+y      #Soma dos vetores x,y
[1] 7 9 6 3 0 8 11

> x+y=(1+6,2+7,3+3,4+(-1),-5+5,6+2,7+4)

> sum(x,y) #Somatório dos vetores x,y
[1] 44

> sum(x,y)=(1+2+3+4+(-5)+6+7+6+7+3+(-1)+5+2+4)
```

3.2 Matriz

Matriz é um conjunto de vetores numéricos. Podemos criar uma matriz a partir de um ou mais vetores, e também podemos transformar uma matriz em vetor. Lembrando que o vetor tem que ser numérico para ser transformado em matriz.

Vamos mostrar alguns comandos para se criar uma matriz, como por exemplo, o comando `matrix()` que cria uma matriz a partir de uma seqüência de números, sendo esta seqüência atribuída às colunas da matriz a ser formada (este é o padrão do comando `matrix()`, preencher a matriz na seguinte ordem: da primeira coluna em direção à última coluna). Veja o exemplo:

Exemplo 3.7

Vamos criar uma matriz a partir da seqüência de números de 1 a 9, com 3 linhas e 3 colunas. Temos duas opções:

```
> matrix(c(1:9),nrow=3)
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

> matrix(c(1:9),ncol=3)
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

`matrix(c(1:9),nrow=3)` ou `matrix(c(1:9),ncol=3)` #Matriz formada pelos números de 1 até 9, com 3 linhas.

Observe que ambos os comandos resultaram na criação da mesma matriz, a seqüência foi alocada da primeira coluna em direção a terceira na ordem de formação dos números.

Entretanto se quiséssemos que a matriz fosse preenchida na ordem da primeira linha para a última, deveríamos informar no comando `matrix()` que o argumento `byrow=T` (pois `byrow=F` é o padrão e pode ser omitido no comando, conforme vemos acima). Veja como ficaria:

Exemplo 3.8

```
> matrix(c(1:9),ncol=3,byrow=T)
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9

> matrix(c(1:9),nrow=3,byrow=T)
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

`matrix(c(1:9),ncol=3,byrow=T)` ou `matrix(c(1:9),nrow=3,byrow=T)`
#Matriz formada pelos números de 1 até 9, com 3 linhas, e distribuição dos números por linha (Primeiro número na primeira linha e na primeira coluna, segundo número do vetor na primeira linha e na segunda coluna e assim por diante.)

Podemos também criar matrizes a partir de vetores pré definidos, veja o exemplo 3.8, em que temos um vetor `x` de comprimento 9 e a partir dele criamos uma matriz 3x3.

Exemplo 3.9

```
> x<-c(1,2,3,4,-5,6,7,8)

> x
[1] 1 2 3 4 -5 6 7 8

> matrix(x,byrow=T,nrow=3)      #Matriz formada pelo vetor x, por linha
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4   -5    6
[3,]    7    8    1

> matrix(x,byrow=F,nrow=3)     #Matriz formada pelo vetor x, por coluna
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2   -5    8
[3,]    3    6    1
```

`matrix(x,byrow=T,nrow=2)` #Matriz formada pelo vetor x, com 2 linhas, e distribuição dos números por linha (Primeiro número do vetor na primeira linha e na primeira coluna, segundo número do vetor na primeira linha e na segunda coluna e assim por diante.)

`matrix(x,byrow=F,nrow=2)` #Matriz formada pelo vetor x, com 2 linhas, e distribuição dos números por coluna (Primeiro número do vetor na primeira linha e na primeira coluna, segundo número do vetor na segunda linha e na primeira coluna e assim por diante.)

Outro comando que também pode ser usado para criação de matrizes é o comando `cbind` ou `rbind`, o primeiro cria matrizes por colunas e o segundo por linhas. Veja o exemplo 3.9

Exemplo 3.10

```
> x<-c(1,2,3,4,5,6)

> y<-c(7,8,9,10,11,12)

> x
[1] 1 2 3 4 5 6

> y
[1] 7 8 9 10 11 12

> e<-cbind(x,y) #Matriz formada pelos vetores x e y

> e
      x y
[1,] 1 7
[2,] 2 8
[3,] 3 9
[4,] 4 10
[5,] 5 11
[6,] 6 12

> class(e) #Com esse comando podemos confirmar que é matriz
[1] "matrix"
```

Cada elemento da matriz é armazenado no objeto matriz associado com seu número de linha e de coluna. Dessa forma podemos obter parte de uma matriz através de comandos bem simples, como pode ser visto no exemplo 3.10

Exemplo 3.11

Suponha que desejamos o conteúdo da primeira coluna ou da primeira linha da matriz, utilizamos o seguinte comando:

```
> e[1,]      #Retirar linha 1, para outras linhas é só mudar o número
x y
1 7

> e[,1]      #Retirar coluna 1, para outras colunas só mudar o número
[1] 1 2 3 4 5 6
```

3.3 Data frame

Data frame é muito parecido com a matriz, a diferença é que enquanto na matriz só podem haver números, no data frame podem ter colunas de

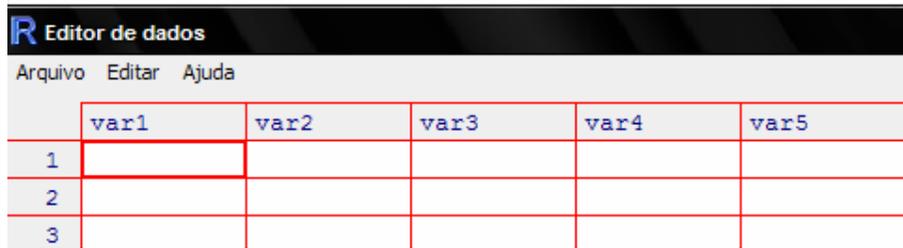
caracteres além das colunas numéricas. Uma das melhores formas de armazenar dados, onde cada coluna é uma variável e cada linha é o indivíduo, ou pessoa, ou unidade.

Exemplo 3.12

```
> idade<-c(25,29,35,33,31,36)
> status<-c("solteiro","casado","casado","solteiro","casado","casado")
> nfil<-c(1,1,2,1,1,3)
> idade
[1] 25 29 35 33 31 36
> status
[1] "solteiro" "casado" "casado" "solteiro" "casado" "casado"
> nfil
[1] 1 1 2 1 1 3
> data.frame(idade,status,nfil)
  idade status nfil
1    25 solteiro   1
2    29  casado   1
3    35  casado   2
4    33 solteiro   1
5    31  casado   1
6    36  casado   3
```

Uma maneira diferente de criar um data frame é utilizando a função `edit(data.frame())`. Ao usar essa função, uma janela, parecida com uma planilha do Excel, é aberta para digitar as informações do banco de dados. Para colocar os nomes das variáveis, basta clicar em cima do “var1”, que uma janela é aberta para você escrever o nome da variável de dizer se é numérica ou de caractere.

```
edit(data.frame())
```



The screenshot shows the R 'Editor de dados' window. At the top, there is a menu bar with 'Arquivo', 'Editar', and 'Ajuda'. Below the menu bar is a table with 5 columns labeled 'var1', 'var2', 'var3', 'var4', and 'var5'. The rows are numbered 1, 2, and 3. The first row has a red border around the 'var1' cell.

Há ainda um outro caminho para ter o banco de dados no R. A função `read.csv2(file="local do arquivo/nome do arquivo.csv")` pode ler arquivos com a extensão `.csv` que estiverem salvos no computador. Então, caso tenha digitado o banco de dados no Excel, basta salvar com a extensão `.csv` que ele poderá ser lido no R. E para você utilizar os dados que foram lidos, basta usar a função `attach(nome do objeto)`, dessa forma poderá usar todas as outras funções sem maiores problemas.

Exemplo 3.13

```
> dados<-read.csv2(file="C:/Documents and Settings/personal/Meus documentos/Faculdade/Monitoria MCI/R/acidentes.csv")
```

```
> dados
  acidente veiculo gravidade feridos idade
1         1   Carro  Pequena         0    35
2         2   Moto   Elevada         3    22
3         3   Carro  Moderada         2    30
4         4   Moto   Moderada         1    28
5         5   Carro  Elevada         2    20
6         6   Carro  Pequena         1    25
7         7   Moto   Moderada         2    35
8         8   Moto   Elevada         3    22
9         9   Carro  Pequena         0    32
10        10   Moto   Moderada         1    27
11        11   Carro  Elevada         2    20
12        12   Carro  Pequena         1    24
13        13   Moto   Moderada         2    28
14        14   Carro  Moderada         2    34
15        15   Carro  Elevada         3    29
16        16   Moto   Pequena         0    33
17        17   Moto   Moderada         2    25
18        18   Carro  Elevada         3    23
19        19   Carro  Moderada         1    30
20        20   Moto   Elevada         1    21
```

Agora que foi feita uma introdução do R, já é possível utiliza-lo para por em prática o que foi estudado em sala de aula.

4. Funções mais utilizadas

Ao estudar Estatística Básica I são utilizados muitos cálculos para observar diferentes medidas. As principais medidas estudadas e calculadas são as medidas de centralidade, medidas de dispersão, medidas de posição e medidas de assimetria. Usando o R é possível realizar esses cálculos, e muitos outros, de forma prática e rápida, facilitando a observação dos resultados do banco de dados.

Abaixo segue o nome das principais funções utilizadas, e quando elas são utilizadas.

Função	Utilização
Sum(x)	Somatório dos elementos de x
prod(x)	Produtório dos elementos de x
Max(x)	Elemento máximo em x
Min(x)	Elemento mínimo em x
range(x)	Elemento máximo e mínimo em x
length(x)	Número total de elementos em x
mean(x)	Média de x
median(x)	Mediana de x
Var(x)	Variância de x
sd(x)	Desvio padrão de x
cor(x,y)	Correlação entre x e y
quantile(x,p)	Quantil
cov(x,y)	Covariância de x e y

Para exemplificar cada uma das funções usaremos um banco de dados que mostra o sexo, peso e altura de uma determinada turma de alunos. Para isso usaremos a função “read.csv2(file=...)”.

Exemplo 4.1

```
> dados=read.csv2(file="C:/Documents and Settings/personal/Meus documentos/Faculdade/Monitoria MCI/R/turma.csv")  
  
> dados  
  Sexo Peso Altura  
1    f   70   160  
2    f   71   157  
3    f   69   160  
4    f   70   160  
5    m   70   158  
6    m   72   163  
7    m   68   159  
8    f   69   163  
9    m   68   159  
10   m   69   163  
11   f   71   160  
12   f   69   160  
13   f   69   161  
14   m   69   158  
15   f   69   159  
16   f   68   160  
17   m   70   162  
18   f   72   161  
19   f   73   162  
20   m   70   156  
  
> attach(dados)
```

Agora que os dados já estão armazenados podemos começar as análises descritivas.

4.1 Usando a função length(x)

Usamos essa função para calcular o tamanho da amostra. Utilizada no banco de dados, podemos descobrir o número de variáveis.

Exemplo 4.2

```
> length(dados)      #Número de variáveis observadas
[1] 3

> length(Peso)       #Número de pesos observados, que é o total da amostra
[1] 20
```

4.2 Usando as funções min(x), max(x) e range(x)

Usamos a função min(x) para achar o menor valor da variável quantitativa, a função max(x) para achar o maior valor da variável quantitativa e range(x) para achar, ao mesmo tempo, o menor e o maior valor da variável quantitativa observada.

Exemplo 4.3

```
> min(Peso)          #Menor peso observado
[1] 68

> max(Peso)          #Maior peso observado
[1] 73

> range(Peso)        #Menor e maior valores observados
[1] 68 73
```

4.3 Usando a função sum(x) e prod(x)

Essas funções, basicamente, somam e multiplicam, respectivamente, os valores da variável observada. Não esquecer que sum(x) e prod(x) é diferente de soma de vetores de produto de vetores, como visto anteriormente.

Exemplo 4.4

```
> sum(Peso)          #Soma de todos os valores da variável peso
[1] 1396

> prod(Peso)         #Produto de todos os valores da variável peso
[1] 7.506978e+36
```

Podemos usar a função `sum(x)` para compor a média (que sabemos que é o somatório dos valores dividido pelo tamanho da amostra).

Exemplo 4.5

```
> sum(Peso)/length(Peso)      #Média
[1] 69.8

> mean(Peso)                  #Média
[1] 69.8
```

4.4 Usando a função `mean(x)`

Usamos a função `mean(x)` para achar a média da variável observada.

Exemplo 4.6

```
> mean(Peso)                  #Média do peso
[1] 69.8

> mean(Altura)                #Média da altura
[1] 160.05
```

Também podemos encontrar a média somente para alguns valores na variável. Nesse caso temos uma forma de especificar nossa escolha.

Exemplo 4.7

Média do peso dos alunos de sexo feminino.

```
> mean(Peso[Sexo=="f"])      #Média do peso dos alunos do sexo feminino
[1] 70
```

4.5 Usando a função `median(x)`

Usamos a função `median(x)` para achar a mediana da variável observada.

Exemplo 4.8

```
> median(Peso)      #Mediana do peso
[1] 69.5

> median(Altura)    #Mediana da altura
[1] 160
```

Também podemos encontrar a mediana somente para alguns valores na variável. Nesse caso temos uma forma de especificar nossa escolha.

Exemplo 4.9

Mediana do peso dos alunos do sexo masculino.

```
> median(Peso[Sexo=="m"])  #Mediana do peso dos alunos do sexo masculino
[1] 69.5
```

4.6 Usando as funções var(x) e sd(x)

Usamos as funções var(x) e sd(x) para achar, respectivamente, a variância e o desvio padrão da variável observada.

Exemplo 4.10

```
> var(Peso)          #Variância do peso
[1] 1.957895

> sd(Altura)         #Desvio padrão da altura
[1] 1.959457
```

Também podemos encontrar a variância e o desvio padrão somente para alguns valores na variável. Nesse caso temos uma forma de especificar nossa escolha.

Exemplo 4.11

Variância do peso dos alunos do sexo feminino e desvio padrão da altura dos alunos do sexo masculino.

```

> var(Peso[Sexo=="f"])      #Variância do peso do alunos do sexo feminino
[1] 2.181818

> sd(Altura[Sexo=="m"])    #Desvio padrão da altura dos alunos do sexo masculino
[1] 2.604940

```

4.7 Usando a função `quantile(x,p)`

Usamos a função `quantile(x,p)` para calcular percentil.

Exemplo 4.12

```

> quantile(Peso,0.7)      #Percentil 70
70%
70

```

4.8 Usando a função `cor(x,y)` e `cov(x,y)`

Usamos essas funções quando queremos observar a relação entre duas variáveis quantitativas.

Exemplo 4.13

```

> cor(Peso,Altura)      #Correlação entre peso e altura
[1] 0.2149977

> cov(Peso,Altura)     #Covariância entre peso e altura
[1] 0.5894737

```

4.9 Tabelas

Utilizamos as tabelas para observar de forma mais rápida o comportamento de certa variável. Deve-se ter atenção para a diferença entre as tabelas de variáveis qualitativas e quantitativas discretas para as quantitativas contínuas.

Para as variáveis qualitativas e quantitativas discretas basta usar a função `table(x)`.

```
> table(Sexo)           #Tabela para variáveis qualitativas e quantitativas discretas
Sexo
  f  m
12  8
```

Podemos fazer também a tabela de frequência relativa, nesse caso usamos `prop.table(x)`.

```
> prop.table(table(Sexo))  #Tabela de frequência relativa
Sexo
  f  m
0.6 0.4
```

Para fazer tabela de frequências para variáveis quantitativas contínuas é um pouco mais complicado. Devemos observar seu menor valor, seu maior valor e a amplitude de cada intervalo. Para isso usamos a função `range(x)` que já foi utilizada anteriormente, e usamos a regra da raiz do tamanho da amostra, no caso é raiz de `length(x)`. O valor encontrado deve ser arredondado, e a amplitude será o valor encontrado mais 1.

```
> range(Altura)           #Maior e menor alturas observadas
[1] 156 163

> sqrt(length(Altura))+1  #Amplitude do intervalo
[1] 5.472136

> table(cut(Altura, seq(155,164,1=5)))  #Tabela de frequência para variáveis quantitativas contínuas

(155,157] (157,160] (160,162] (162,164]
      2         5         8         5
```

4.10 Função `summary(x)`

Usada para variáveis quantitativas. Essa função dá um resumo de algumas funções que foram aprendidas anteriormente: `min(x)`, `max(x)`, `median(x)`, `mean(x)`, `quantile(x,0.25)` e `quantile(x,0.75)`.

```

> summary(Peso)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
68.00  69.00   69.50   69.80   70.25   73.00

> summary(Altura)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
156.0  159.0   160.0   160.1   161.2   163.0

```

5. Gráficos

Ao digitar a função para criação de gráfico no R, ao invés de aparecer na própria janela o resultado (como acontece com as outras funções), uma nova janela é aberta, automaticamente, contendo o gráfico. Como nos gráficos feitos no Excel, também podemos colocar títulos, diferentes cores e legendas nos gráficos do R. Abaixo teremos um exemplo de cada tipo de gráfico, primeiro da forma mais simples, e depois com título e cores modificadas.

Para fazer um gráfico com título, mudar o estilo e cor dos pontos, basta adicionar algumas funções dentro da própria função do gráfico. Abaixo seguem algumas funções para esse tipo de formatação.

```
main="Título do gráfico" # Coloca um título no gráfico
```

```
pch=número # Muda o tipo do ponto no gráfico de dispersão
```

```
col="nome da cor" # Coloca cor no gráfico
```

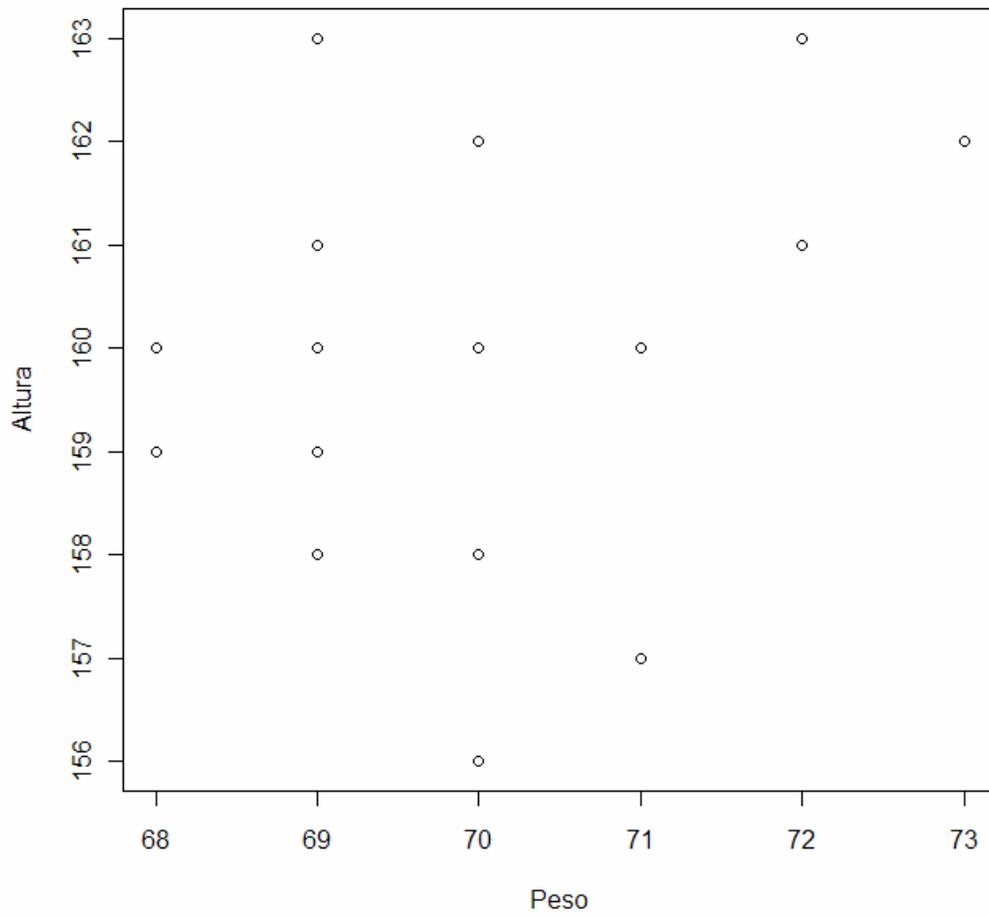
```
abline(lm(Variável1~Variável2)) # Reta de regressão linear
```

5.1 Dispersão

Para realizar o gráfico de dispersão no R, basta utilizarmos a função `plot(x,y)`, onde `x` e `y` são os vetores onde estão armazenados os valores das variáveis quantitativas.

Exemplo 5.1

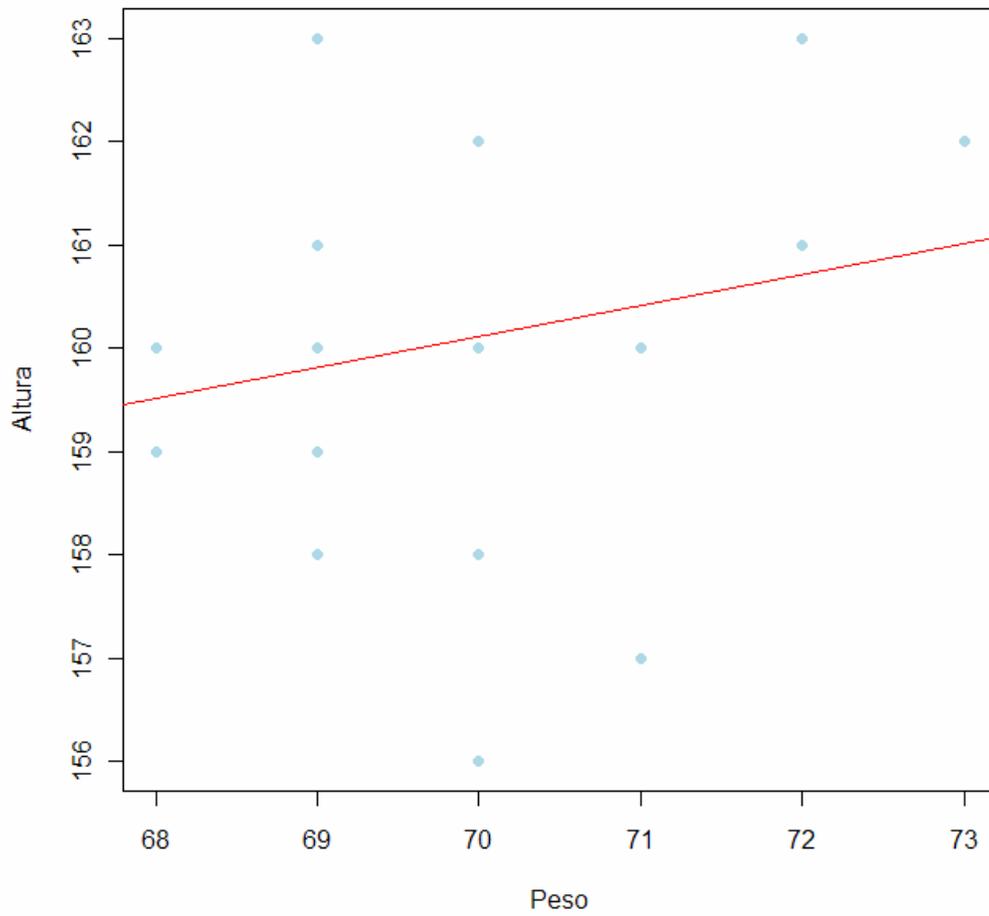
```
> plot(Peso,Altura) #Gráfico de dispersão relacionando peso e altura
```



Exemplo5.2

```
plot(Peso,Altura,main="Gráfico de dispersão entre Peso e Altura",pch=16,col="lightblue")  
abline(lm(Altura~Peso),col="red")
```

Gráfico de dispersão entre Peso e Altura

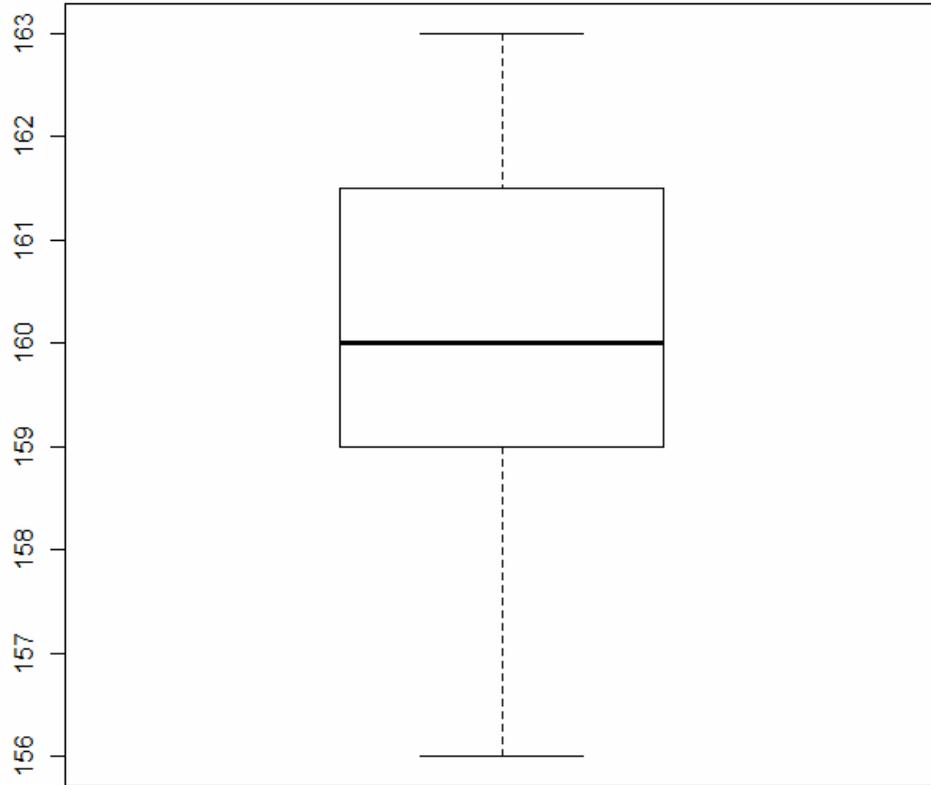


5.2 Box-plot

Para fazer um gráfico de box-plot, basta utilizar a função de mesmo nome, `boxplot(x)`, sendo `x` o nome da variável.

Exemplo 5.3

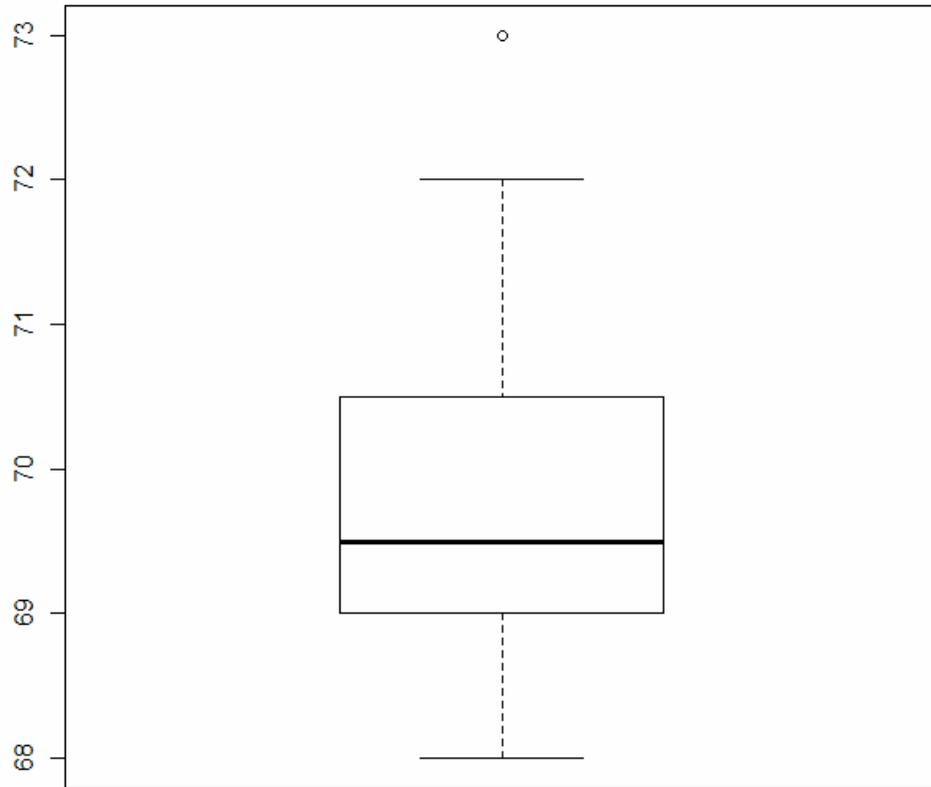
```
> boxplot(Altura) #Box-plot para a variável altura
```



Exemplo 5.4

```
boxplot(Peso, main="Box-plot para Peso")
```

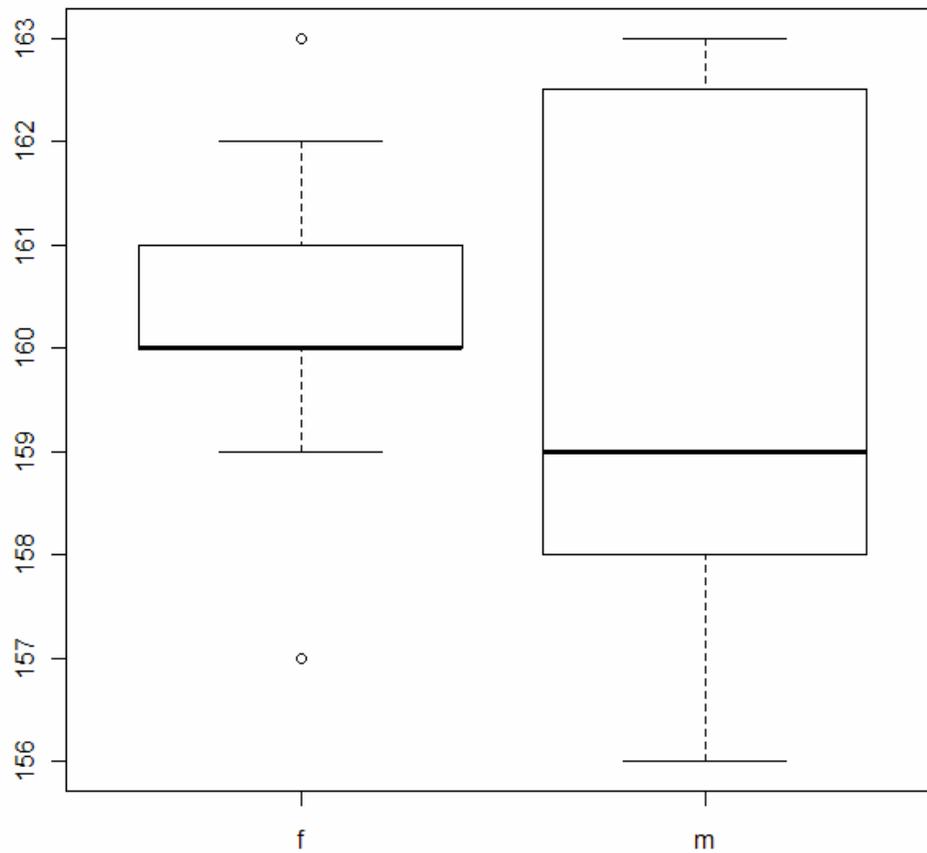
Box-plot para Peso



Exemplo 5.5

```
boxplot(Altura~Sexo,main="Box-plot da Altura de acordo com o sexo")
```

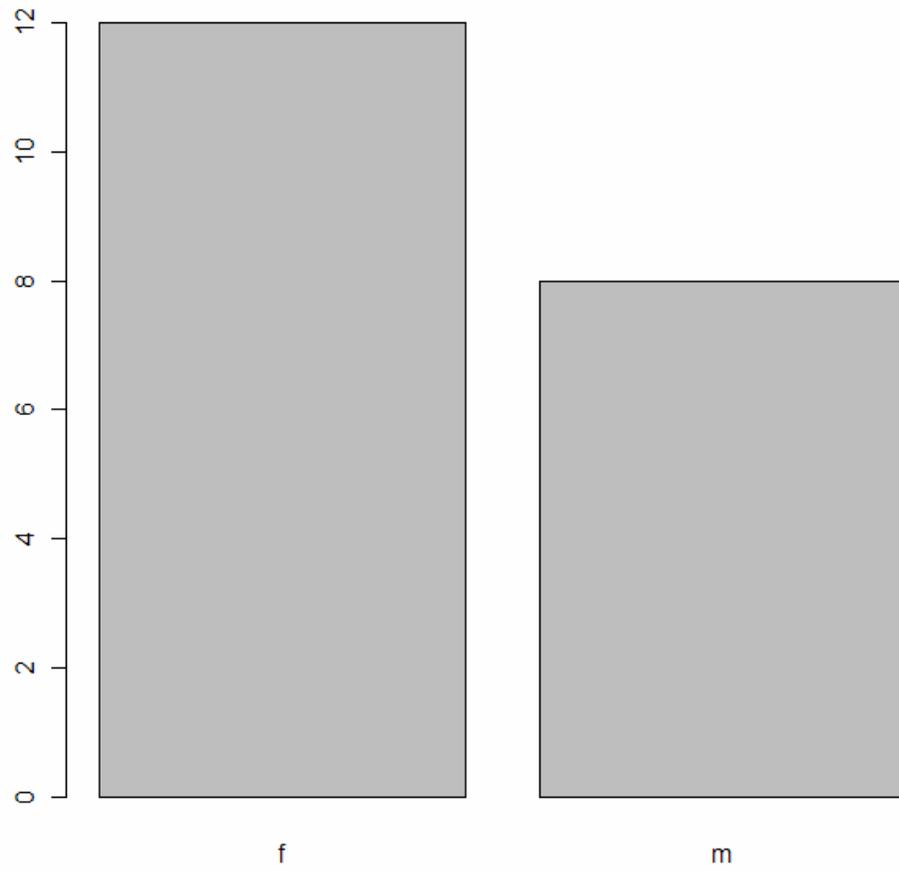
Box-plot da Altura de acordo com o sexo



5.3 Colunas

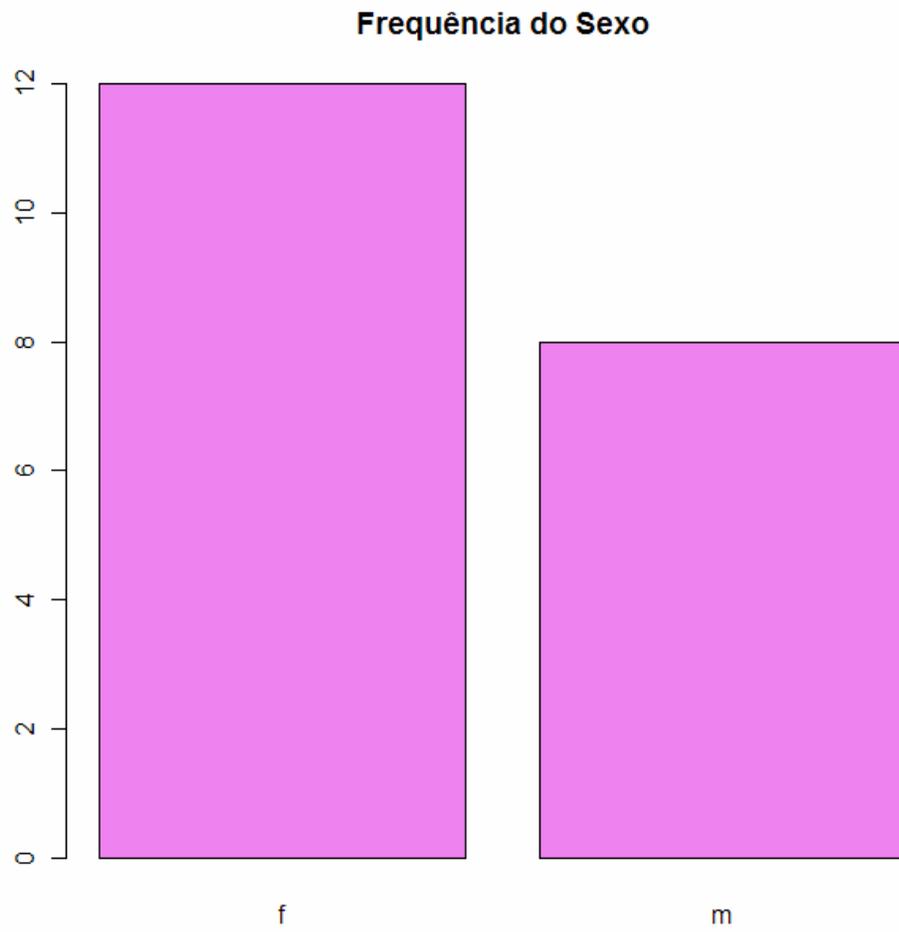
Exemplo 5.6

```
> tablesexo<-table(Sexo)      #Tabela para variáveis qualitativas e quantitativas discretas  
> barplot(tablesexo)        #Gráfico de colunas para a variável peso
```



Exemplo 5.7

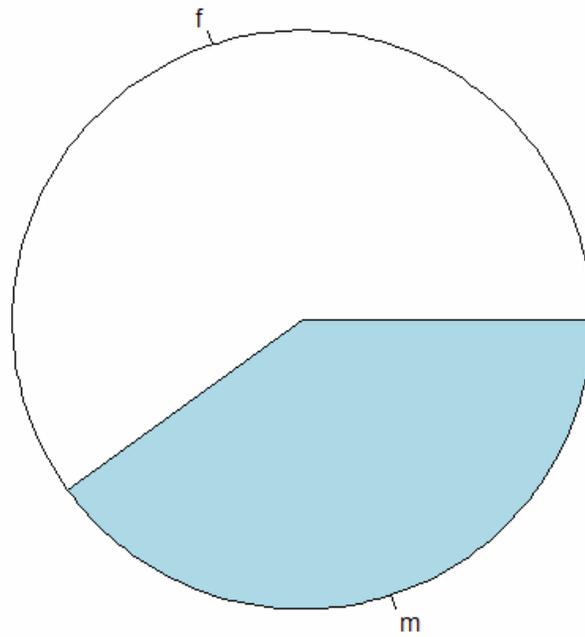
```
barplot(table(Sexo),main="Frequência do Sexo",col="violet")
```



5.4 Setores

Exemplo 5.8

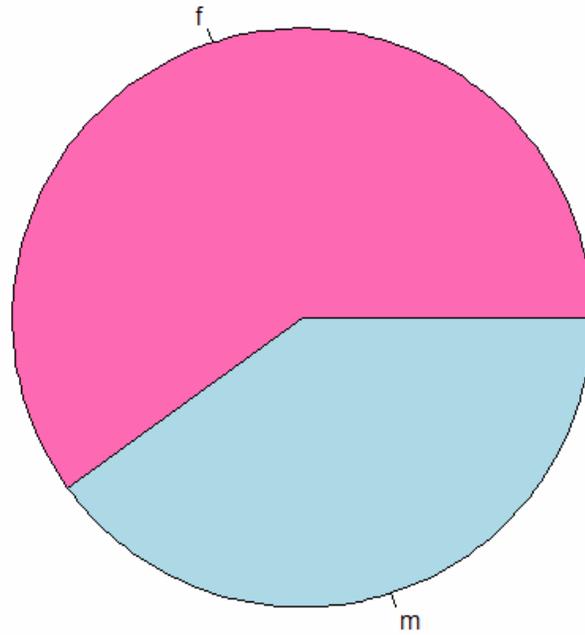
```
pie(table(Sexo)) #Gráfico de setores
```



Exemplo 5.9

```
pie(table(Sexo),main="Gráfico de setores por Sexo",col=c("hotpink","lightblue"))
```

Gráfico de setores por Sexo

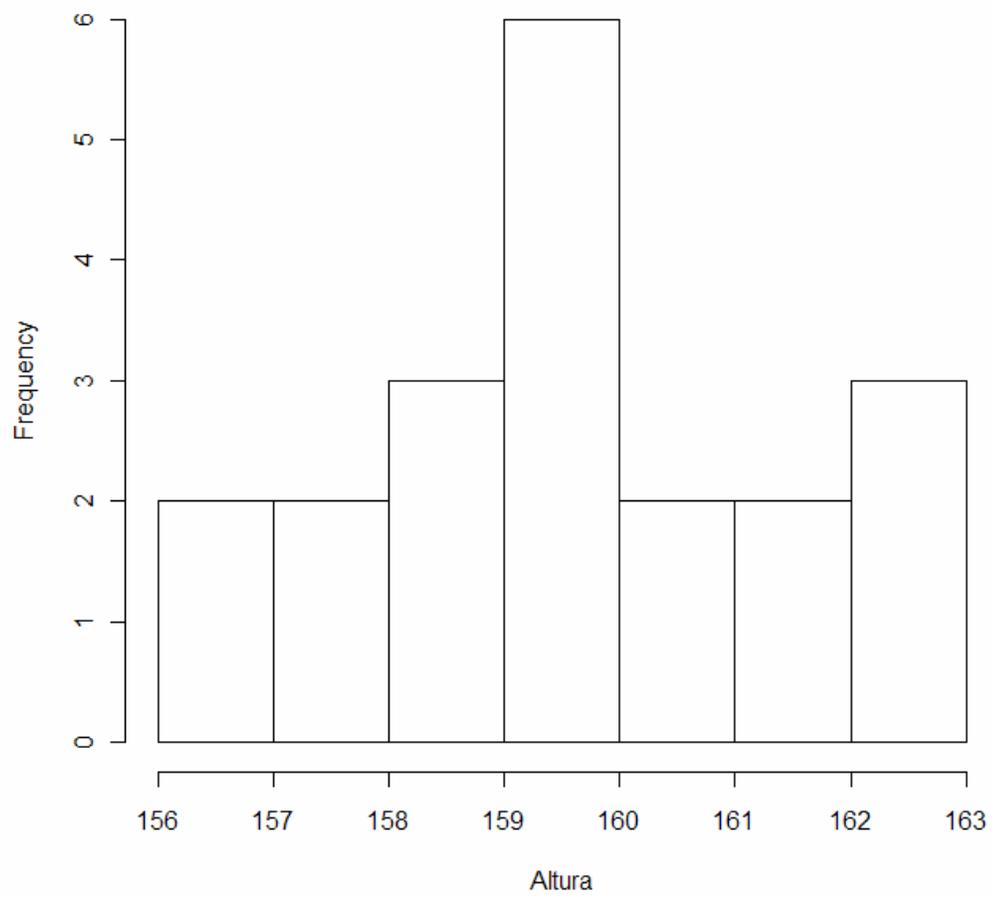


5.5 Histograma

Exemplo 5.10

```
> hist(Altura) #Histograma para a variável altura
```

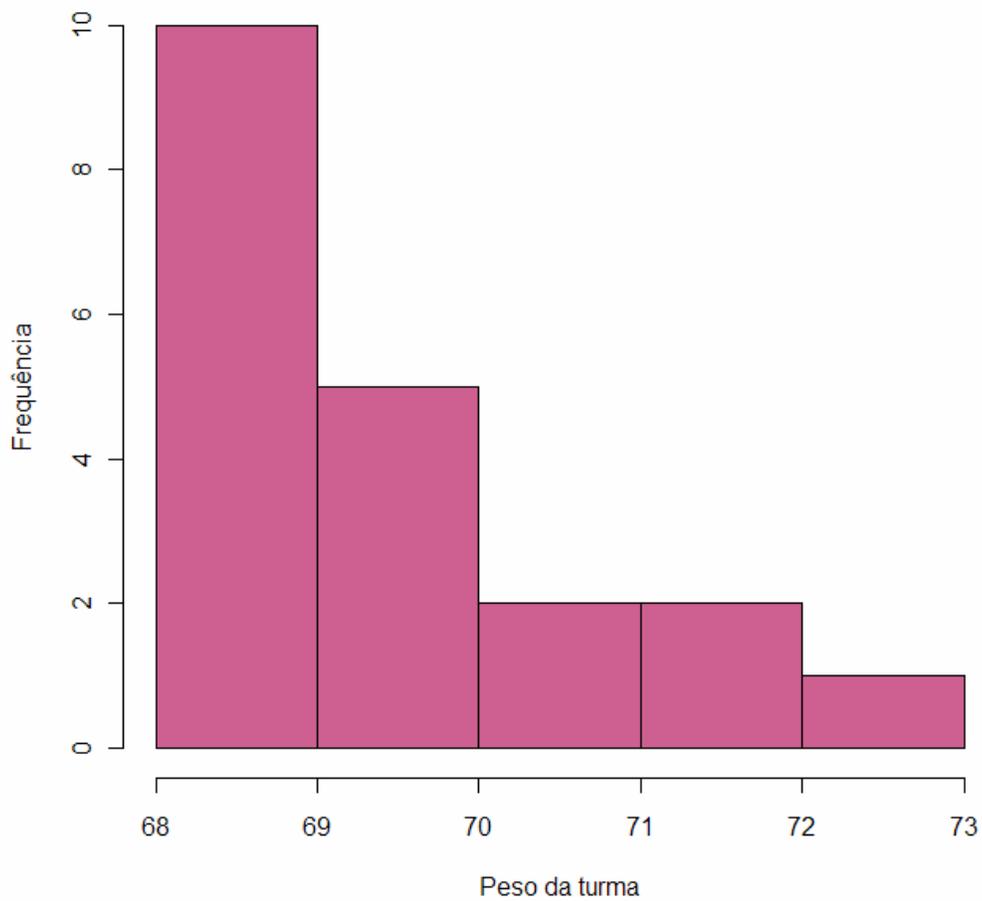
Histogram of Altura



Exemplo 5.11

```
hist(Peso,main="Histograma do Peso",col="hotpink3",xlab="Peso da turma",ylab="Frequência")
```

Histograma do Peso



Uma outra função que pode ser muito interessante para comparar gráficos de diferentes variáveis é a função `par(mfrow=c(x,y))`, onde `x` é o número de linhas e `y` o número de colunas, seguida das funções dos gráficos. Essa função faz com que mais de um gráfico apareça na mesma janela.

Exemplo 5.12

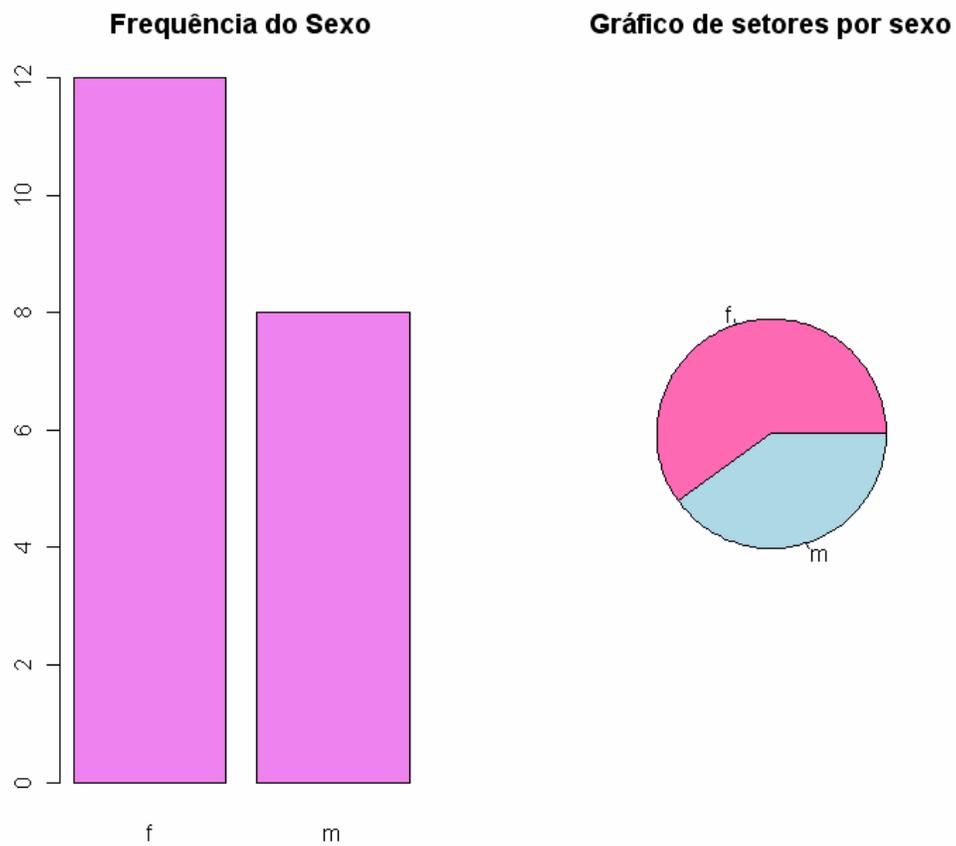
```
attach(dados)

par(mfrow=c(1,2)) #Colocar dois gráficos na mesma janela

table(Sexo)->tablesexo

barplot(tablesexo,main="Frequência do Sexo",col="violet")

pie(table(Sexo),main="Gráfico de setores por sexo",col=c("hotpink","lightblue"))
```



Exemplo 5.13

```

par(mfrow=c(2,2)) #Colocar quatro gráficos na mesma janela

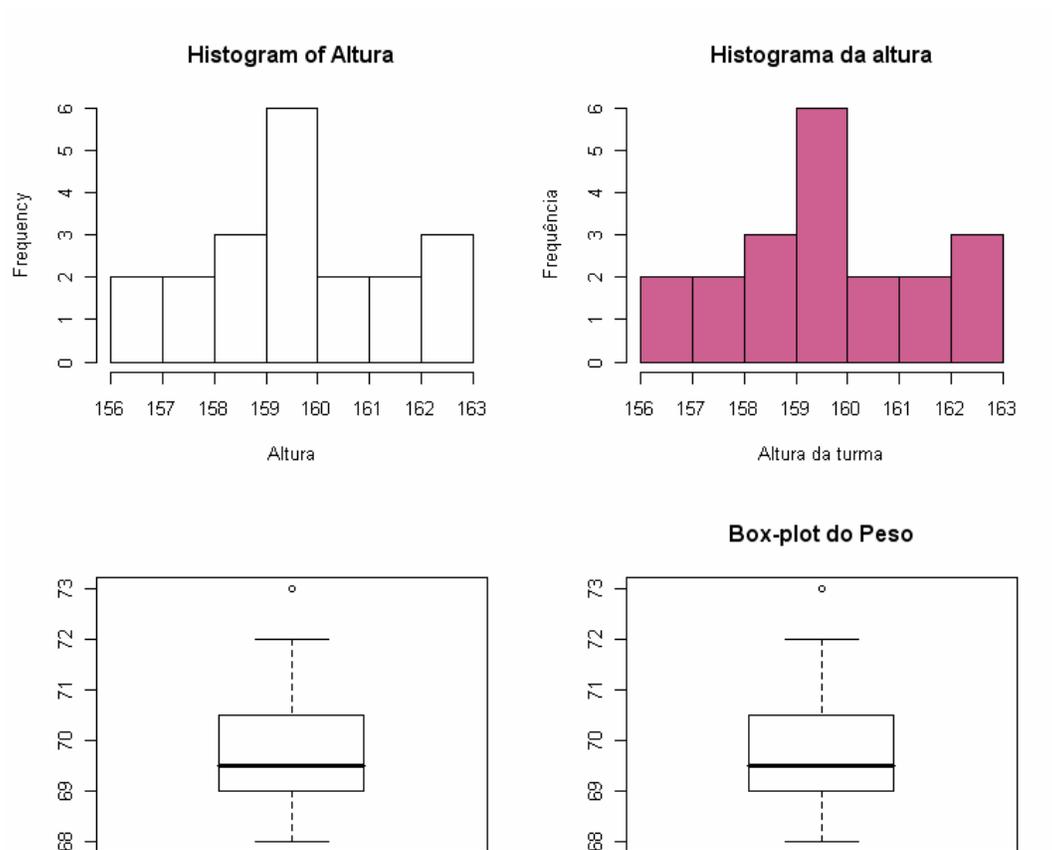
hist(Altura)

hist(Altura,main="Histograma da altura",col="hotpink3",xlab="Altura da turma",ylab="Frequência")

boxplot(Peso)

boxplot(Peso,main="Box-plot do Peso")

```



Exemplo 5.14

```

par(mfrow=c(3,1)) #colocar 3 gráficos na mesma janela

plot(Peso,Altura)

plot(Peso,Altura,main="Gráfico de dispersão entre Peso e Altura",pch=16,col="lightblue")

plot(Peso,Altura,main="Gráfico de dispersão entre Peso e Altura",pch=16,col="lightblue")

abline(lm(Altura~Peso),col="red")

```

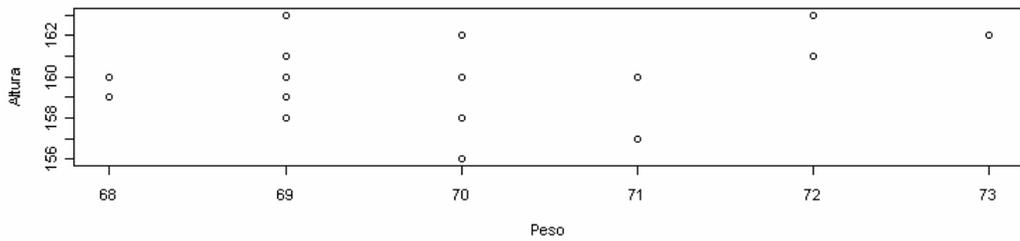


Gráfico de dispersão entre Peso e Altura

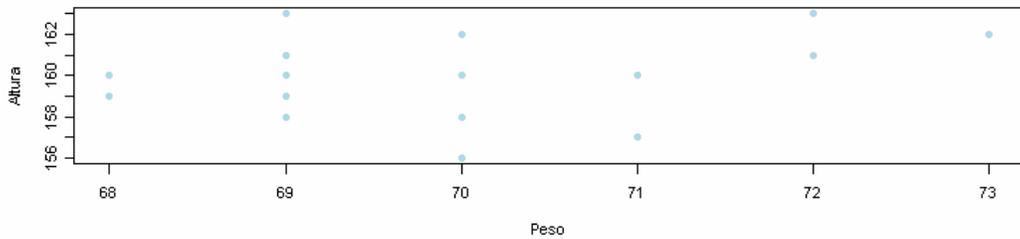
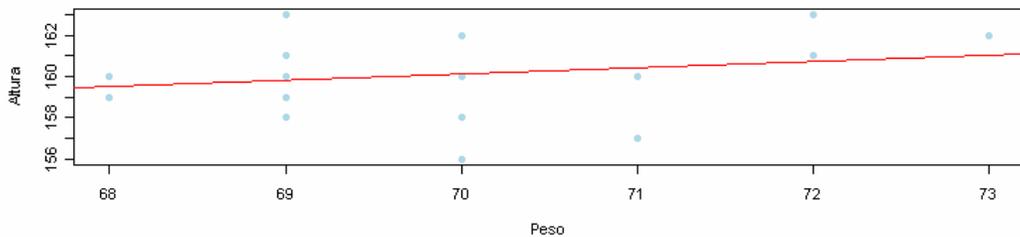


Gráfico de dispersão entre Peso e Altura



Exemplo 5.15

```

par(mfrow=c(1,3)) #colocar 3 gráficos na mesma janela

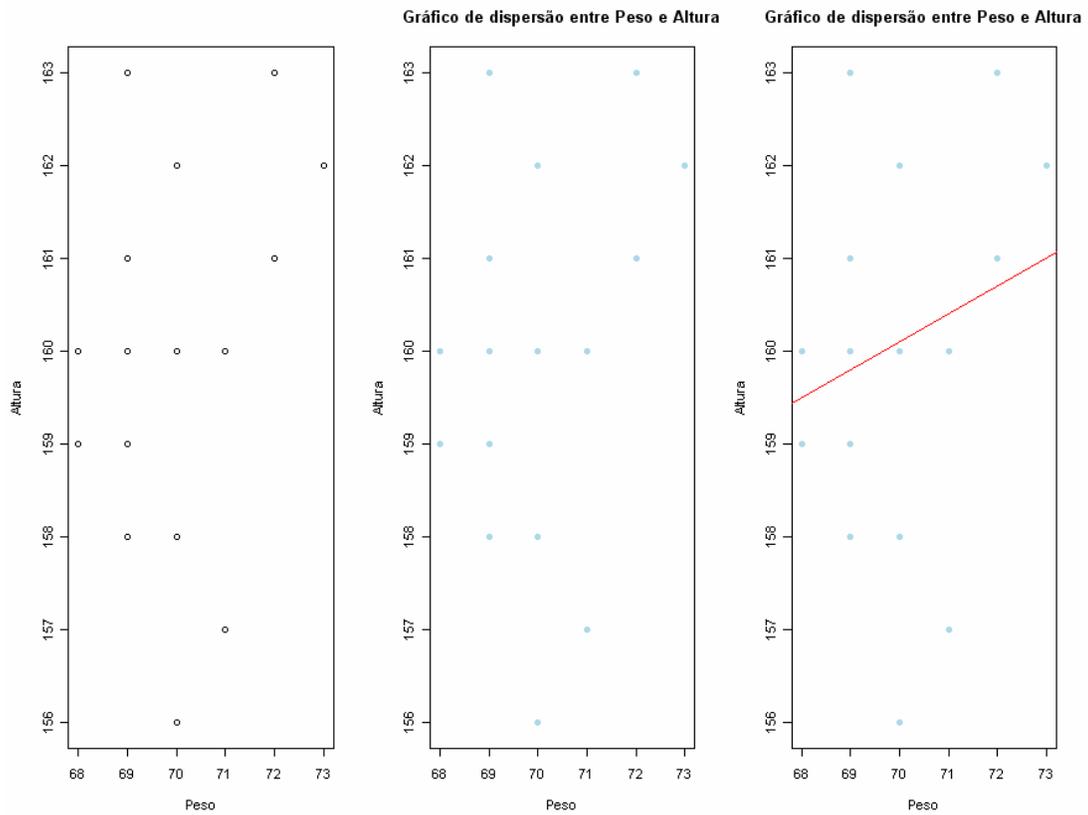
plot(Peso,Altura)

plot(Peso,Altura,main="Gráfico de dispersão entre Peso e Altura",pch=16,col="lightblue")

plot(Peso,Altura,main="Gráfico de dispersão entre Peso e Altura",pch=16,col="lightblue")

abline(lm(Altura~Peso),col="red")

```



Exercícios

Resolva as seguintes operações:

a) $43 - [(24 - 3) + 5]$

b) $\left[\frac{21}{7} \sqrt{25} + 3 \right] - \log_{10}(121)$

c) $\frac{(13^5 + 34)}{\sqrt{23} - 3^{\sqrt{4}}}$

2. Crie uma seqüência de dados de 1 a 30 apenas com números ímpares.
3. Crie uma seqüência decrescente de dados de 102 a 334 apenas com números pares.
4. Crie uma seqüência de dados de 1 a 5, com cada um dos números repetindo 2 vezes.
5. Crie uma seqüência de dados de 1 a 4, repetindo essa seqüência 3 vezes.
6. A partir dos resultados abaixo, retorne o comando que realiza esse resultado:

a) [1] 5 6 1

b) [1] -5 -4 -3 -2 -1 0 1 2 3 4 5

c) [1] 1 1 1 2 2 2 3 3 3

d) 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

Agora, dado os vetores abaixo, encontre os comandos que realize os seguintes resultados:

$x \leftarrow -c(1,3,5,7,9)$ $y \leftarrow -c(2,3,5,11,13)$

e) [1] 2 4 6 8 10

f) [1] 4 6 10 22 26

g) [1] 3 6 10 18 22

7. Suponha que você marcou o tempo que você leva para chegar à faculdade durante 10 dias e esses tempos foram: 17,13,13,11,13,15,16,18,21,26.

Armazene esses valores em um vetor chamado tempo. Para esses valores, calcule o tempo máximo, mínimo e o tempo médio que você levou para chegar na faculdade.

8. Em uma pesquisa feita sobre a população de sapos em 10 regiões diferentes, foram observados os números de sapo, a umidade e a temperatura de cada região. Abaixo seguem os dados:

sapos 7-12-10-16-17-21-13-24-25-31

umid 63-23-29-36-41-74-44-58-61-76

temp 32-30-29-16-24-10-11-25-22-12

- Armazene os dados acima como data frame no R.
- Calcule a média e mediana para cada uma das observações.
- Faça dois gráficos de dispersão, para aparecerem na mesma janela, um comparando o número de sapos com a umidade da região e outro comparando o número de sapos com a temperatura.

9. Durante um ano foi observado o consumo de energia de uma casa e a temperatura, a fim de descobrir se havia alguma relação entre o consumo e a temperatura. Abaixo seguem os dados de cada mês:

Mês = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

Consumo de energia = 87, 45, 92, 138, 99, 41, 59, 118, 132, 50, 43, 52

Temperatura em °C = 20, 19, 26, 35, 23, 15, 26, 29, 32, 21, 17, 23

- Armazene os dados do consumo de energia como “consumo” e da temperatura com “temp”, como um objeto do tipo `data.frame`.
- Calcule a correlação e a covariância entre esses dados.
- Faça um gráfico de dispersão entre as duas variáveis e inclua no gráfico uma linha de tendência, na cor vermelha e a equação da reta.

10. O tempo de vida de algumas baterias, da mesma marca, foi observado a fim de descobrir se o tempo de garantia atual (3 anos) era suficiente. Abaixo seguem os tempos observados:

4.0 2.3 2.6 2.9 3.5 3.6 3.9 3.8 4.3 2.4 3.2 3.3 2.0 2.6 2.7 2.9
3.8 2.9 2.2 3.7 2.1 3.6 3.6 2.3 4.0

- Construa a tabela de frequência e o histograma
- Calcule a média
- Calcule a variância

