ESTATÍSTICA BÁSICA COM USO DO SOFTWARE R

©Adilson dos Anjos
Departamento de Estatística
- UFPR -

Curitiba, 14 de março de 2014.

MÓDULO 2: Estatística descritiva

Objetivo do Módulo

Ao final desse módulo o aluno deverá ser capaz de reconhecer os diferentes tipos de variáveis, utilizar métodos de estatística descritiva adequados para explorar conjuntos de dados, construir uma tabela de frequências, criar e manipular gráficos para entender o comportamento dos dados. Utilizar o recurso de criação de funções.

2.1 Tipos de variáveis

As característica de uma população ou amostra são denominadas variáveis. Por exemplo, no questionário sobre dados biométricos, as respostas fornecidas são características da amostra de participantes do curso, como altura , peso, estado civil. Essas variáveis possuem naturezas diferentes. Altura e peso são variáveis numéricas e estado civil é uma variável não numérica.

As variáveis numéricas são denominadas de variáveis quantitativas enquanto que as variáveis não numéricas são denominadas qualitativas.

Assim tem-se,

1. Variável qualitativa é aquela que não assume valores numéricos. Ela apenas representa algum atributo ou qualidade. Por exemplo, cor, marca de carro, sexo etc.
2. Variável quantitativa é aquela que pode ser medida numericamente. Por exemplo: altura, peso, número de filhos etc.

Em particular, uma variável quantitativa pode ser classificada como uma *variável discreta* ou uma *variável contínua*.

Uma *variável discreta* é aquela cujos valores são, de maneira geral, contagens. Por exemplo: o número de peças com defeito em um lote, o número de pessoas em uma família, etc. Observe que nesse caso, não existem valores intermediários na contagem. Os valores são inteiros, finitos e enumeráveis.

Uma *variável contínua* é aquela que pode assumir qualquer valor dentro de um intervalo. Em geral, são provenientes de um processo de mensuração. Por exemplo: altura, peso etc.

No **R**, quando uma variável é quantitativa (*numeric*), o comando `summary()` retorna algumas estatísticas sobre o vetor de dados. Já, quando o vetor representa uma variável qualitativa (*integer*), o comando `summary()` retorna os "tipos" encontrados e a freqüência de cada tipo.

```
> x<-c(23,45,78,98,56,6.3,4,105,587,31)
> summary(x)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    4.0    25.0    50.5   103.3    93.0   587.0

> q<-c("a","a","a","b","b","c")
> summary(q)

   Length     Class      Mode
        6 character character
```

Veja que tipo de objeto foi criado:

```
> class(q)

[1] "character"
```

Como esperado, o objeto `q` é do tipo "character", porque foram inseridas letras.

Agora, convertendo o objeto q para `factor` tem-se:

```
> summary(factor(q))

a b c
3 2 1
```

Agora, são mostrados os níveis (letras) e a quantidade de observações de cada um.

2.2 Medidas de tendência central

Uma medida de tendência central ou posição informa a posição de um valor em relação a outros valores na amostra ou população. Existem várias medidas de posição que podem ser utilizadas de acordo com a necessidade e características das informações. Nesse curso serão estudadas a média, mediana, quartis e percentis.

A **Média** ou média aritmética é definida por

$$\text{Média} = \frac{\text{Soma de todos os valores}}{\text{Número de valores somados}}.$$

Em geral, utiliza-se o símbolo $\bar{x}$ para denotar a média amostral e μ para denotar a média populacional. Assim como, utiliza-se n para representar o número de observações em uma amostra e N para o número de observações em uma população.

Utilizando uma simbologia matemática, pode-se utilizar a **notação de somatório**, para definir a média amostral

$$\bar{x} = \frac{\sum\limits_{i=1}^{n} x_i}{n};$$

Para entender a notação de somatório, observe os dados da tabela a seguir:

i	1	2	3	4
y_i	2	5	7	4

A soma de todos os valores é denotada por:

$\sum\limits_{i=1}^{4} y_i = 2 + 5 + 7 + 4 = 18.$

Ou ainda, a soma dos três primeiros:

$\sum\limits_{i=1}^{3} y_i = 2 + 5 + 7 = 14$

Para esses dados a média amostral é

$\bar{x} = \frac{\sum_{i=1}^{4} x_i}{4} = \frac{2+5+7+4}{4} = \frac{18}{4} = 4,5.$

No **R** a média de uma amostra é obtida por meio da função `mean()`

```
> x<-c(2, 5, 7 ,4)
> mean(x)
```

```
[1] 4.5
```

A média é uma medida de posição ou tendência central, que é muito afetada por *outliers*.

Outliers são observações que não correspondem aos valores esperados de uma população ou amostra. Em geral, são valores que estão muito acima ou muito abaixo da maioria dos valores observados.

Existem critérios estatísticos que podem ser utilizados para definir se uma observação é, ou não, um outlier. No entanto, a palavra final é sempre do pesquisador que conhece o fenômeno, e pode informar se a informação obtida é realmente aceitável.

Exemplo: Considere o seguinte conjunto de dados: dados=(6, 9, 9, 6, 70). Observe que há uma onservação que possui um valor muito distante das outras observações. Esse valor, a princípio, pode ser considerado um outlier. Veja como esse valor afeta a média:

Média com outlier

```
> dados<-c(6,9,9,6,70)
> mean(dados)
```

```
[1] 20
```

Média sem outlier

```
> mean(dados[-5]) # sem a quinta observação
```

```
[1] 7.5
```

Pense nisso: Se uma pessoa estiver com os pés dentro de um forno com temperatura de 50 graus Celsius e a cabeça dentro de um freezer com temperatura de zero graus na média estará em uma temperatura agradável!

Por esse motivo é importante que uma medida de posição esteja acompanhada de uma medida de dispersão.

2.3 Mediana

Mediana é o valor na posição $\frac{n+1}{2}$ em um conjunto de dados ordenados.

Exemplo: Considere o seguinte conjunto de n=12 observações: 13, 23, 36, 50, 97, 210, 234, 249, 257, 275, 385, 506.

A mediana é o valor na posição $\frac{n+1}{2} = \frac{12+1}{2} = 6,5$

O valor pode ser obtido pela interpolação dos valores que estão na posição seis e sete do conjunto ordenado de observações.

Mediana $= x_6 + 0.5(x_7 - x_6) = 210 + 0.5(234 - 210) = 222$

No **R** a mediana pode ser obtida com a função `median()`

```
> dados<-c(13, 23, 36, 50, 97, 210, 234, 249, 257, 275, 385, 506)
> median(dados)

[1] 222
```

2.4 Quartis

São medidas que dividem os dados ordenados em quatro partes iguais: primeiro quartil (Q1), segundo quartil (Q2) e terceiro quartil (Q3).

Pode-se dizer que 25% dos valores estão abaixo de Q1 e 75% dos valores estão acima de Q1. A diferençao entre Q3 e Q1 é chamada de *amplitude interquartílica* . O segundo quartil é exatamente igual à mediana.

Existem vários algoritmos que podem ser utilizados para obter os valores dos quartis. No **R** existem 9 tipos programados. Nesse curso será considerado o algoritmo do tipo 4, que faz uma interpolação das observações da amostra.

O primeiro quartil é obtido por $Q_1 = x_{\frac{n}{4}}$ e $Q_2 = x_{\frac{3n}{4}}$.

2.5 Exemplo

Considere as seguintes observações: 16, 38, 18, 20, 20, 18, 22, 34, 7, 58, 31 e 19. No **R**, esses dados podem ser inseridos da seguinte maneira:

```
> x<-c(16, 38, 18, 20, 20, 18, 22, 34, 7 ,58, 31, 19)
```

Os dados podem ser ordenados da seguinte forma::

```
> sort(x)

 [1]  7 16 18 18 19 20 20 22 31 34 38 58
```

O primeiro quartil (Q_1) é obtido da seguinte forma, considerando uma amostra de tamanho $n = 12$

$Q_1 = x_{\frac{12}{4}} = x_3$, portanto, a observação na posição 3 da amostra ordenada é o valor do primeiro quartil.

$Q_3 = x_{\frac{3*12}{4}} = x_9$, portanto, a observação na posição 9 da amostra ordenada é o valor do terceiro quartil.

Utilizando-se a função `quantile()`, obtém-se os valores dos quartis de x. No **R**, o algoritmo tipo 4 é baseado na interpolação de dados e pode ser utilizado da seguinte maneira:

```
> quantile(x,type=4)

  0%  25%  50%  75% 100%
   7   18   20   31   58
```

A função `summary()`, quando aplicada sobre um conjunto de dados no **R**, fornece algumas estatísticas sobre os dados.

```
> summary(x)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   7.00   18.00   20.00   25.08   31.75   58.00
```

Nesta saída, Min.=valor mínimo, 1st Qu.=primeiro quartil, Median=mediana Mean= média 3rd Qu.=terceiro quartil Max.=valor máximo.

Tanto a função `summary()` quanto a função `quantile()` não necessitam que os dados sejam ordenados.

A função `fivenum()` fornece informações semelhantes:

```
> fivenum(x)

[1]  7.0 18.0 20.0 32.5 58.0
```

Cuidado Observe que as funções `summary()` e `fivenum()` utilizam algoritmos diferentes do especificado no exemplo apresentado.

2.6 Percentis

São medidas que dividem os dados ordenados em 100 partes iguais. Em uma amostra, são possíveis de serem calculados 99 percentis.

O *k-ésimo* percentil, denotado por P_k, é o valor na posição, de forma que k% das medidas são menores que a posição P_k, ou seja, (100-k)% das observações são maiores que P_k. O *k-ésimo* percentil é determinado por:

$P_k =$ é o valor do $kn/100-$ésimo termo no conjunto de dados ordenados, onde k é o percentil e n é o tamanho da amostra.

2.7 Exemplo

Considere o conjunto de dados da amostra x do exemplo 2.5. O percentil 62 (P_{62}) é dado por

$$P_{62} = \frac{kn}{100} = \frac{62 \times 12}{100} = 7,44$$

Nesse caso, o percentil 62 está entre os números nas posições 7 e 8. Pode-se obter esse percentil, interpolando-se as observações nestas posições:

$$\begin{aligned} P_{62} &= x_7 + 0,44(x_8 - x_7) \\ P_{62} &= 20 + 0.44(22 - 21) = 20,88 \end{aligned}$$

No **R**, função `quantile()` pode ser utilizada para obtenção do percentil $P62$,

```
>  quantile(x,.62,type=4)

  62%
20.88
```

Nesse caso, pode-se concluir que 62% dos dados estão abaixo de 20,88 e que 38% estão acima desse valor.

2.8 Medidas de Dispersão

Amplitude é a medida mais simples de dispersão e é definida como sendo a diferença entre o maior e o menor valor entre os dados observados.

A amplitude é bastante influenciada por outliers. Por isso, não é uma boa medida para dados que contenham outliers. Ainda, essa é uma medida de dispersão que utiliza somente 2 observações, independente do tamanho da amostra.

No **R** pode-se utilizar os seguintes comandos:

```
>  x<-c(3,8,12,4,1,15,15)
>  range(x)  #  amplitude

[1]  1 15

>  diff(range(x)) #diferença entre o maior e o menor valor

[1] 14
```

2.9 Valores extremos

Os valores extremos, mínimo e máximo, podem ser obtidos no **R** da seguinte maneira:

```
> min(x)  # valor mínimo de x

[1] 1

> max(x)  # valor máximo de x

[1] 15
```

2.10 Variância e desvio padrão

O desvio padrão fornece uma medida de dispersão das observações ao redor da média. Um desvio padrão pequeno indica que os dados possuem uma amplitude pequena ao redor da média. Já, um desvio padrão grande, indica que os dados possuem uma amplitude grande ao redor da média.

O desvio padrão é fornecido sempre na mesma escala da variável resposta e é obtido pela raiz quadrada da variância.

A variância de uma amostra, denotada por s^2, tem a seguinte expressão:

$$s^2 = \frac{\sum_{i=1}^{n}(x-\bar{x})^2}{n-1}.$$

No caso de uma população:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x-\mu)^2}{N},$$

em que a quantidade $(x-\bar{x})$ é conhecida como desvio de x em relação à média. Observe que $\sum_{i=1}^{n}(x-\bar{x})$ é sempre zero, por isso, aplica-se o quadrado.

Uma maneira alternativa de calcular a variância amostral é:

$$s^2 = \frac{\sum x^2 - \frac{(\sum x)^2}{n}}{n-1}$$

e a variância populacional:

$$\sigma^2 = \frac{\sum x^2 - \frac{(\sum x)^2}{n}}{N}$$

Logo, o desvio padrão pode ser obtido por $s = \sqrt{s^2}$ ou $\sqrt{\sigma^2}$.

Exemplo: Considere os dados das amostras $x_1 = (0, 2, 3, 4, 6)$ e $x_2 = (1, 2, 3, 4, 5)$.

Para x_1 tem-se que $s^2 = 5$ e $s = 2,24$ e, para x_2, $s^2 = 2,5$ e $s = 1,58$. Portanto, os valores da amostra x_1 são mais dispersos do que os da amostra x_2.

No **R** utilizam-se os seguintes comandos:

```
> x1<-c(0,2,3,4,6)
```

```
> x2<-c(1,2,3,4,5)
> var(x1);var(x2) # para obter a variância

[1] 5

[1] 2.5

> sd(x1);sd(x2)   # para obter o desvio padrão

[1] 2.236068

[1] 1.581139
```

2.10.1 Coeficiente de variação (CV)

O coeficiente de variação expressa o desvio padrão como percentual da média.

$$CV = \frac{s}{\bar{x}}100.$$

O CV fornece uma idéia de precisão experimental: quanto menor o CV, menor a variabilidade e melhor a precisão experimental. Por outro lado, quanto maior o CV, maior será a variabilidade experimental e pior será a precisão experimental.

O CV de variação é extremamente afetado pela escala da variável resposta. Por esse motivo ele é, em geral, apenas um bom indicador para comparar variáveis semelhantes.

No **R** pode-se utilizar:

```
> x1<-c(0,2,3,4,6)
> x2<-c(1,2,3,4,5)
> CV.x1<-sd(x1)/mean(x1)*100
> CV.x1

[1] 74.5356

> CV.x2<-sd(x2)/mean(x2)*100
> CV.x2
```

```
[1] 52.70463
```

Nesse exemplo, os valores da variável $x1$ são mais dispersos do que os da variável $x2$.

2.11 Organização de dados em Tabelas e Gráficos

As variáveis, tanto qualitativas quanto quantitativas, podem ser resumidas em tabelas e gráficos. Para cada variável, existem maneiras mais adequadas de representação dos dados. Veremos algumas nesse curso.

Para variáveis qualitativas, em geral são utilizadas tabelas de frequências para representar as frequências de cada categoria. Para as mesmas variáveis, podem ser utilizados gráficos como o gráfico de barras e de setores (pizza).

Para variáveis quantitativas, podem-se ser utilizadas tabelas de frequências para representar a ocorrência de valores em classes pré-estabelecidas. Também podem ser utilizados gráficos como o histograma ou ramo e folhas.

2.12 Tabela de Frequências

Uma tabela de frequências fornece informações sobre a frequência de categorias ou classes em um conjunto de dados.

Exemplo de uma tabela de frequências (Tabela 1):

Tabela 1: Distribuição de frequências de pessoas que tiveram infecção alimentar, em uma amostra de 140 pessoas, no restaurante da empresa.

Categoria	Frequência
Nenhuma	110
Leve	12
Moderada	10
Severa	8
Total	140

Em um tabela, os dados podem ser apresentados tanto na forma de frequência absoluta, quanto na forma de frequência relativa ou em percentagem.

A frequência relativa é obtida por:

$$\text{Frequência relativa de uma categoria} = \frac{\text{Frequência da categoria}}{\text{Soma de todas as Frequências}}$$

A percentagem é simplesmente a Frequência relativa $\times$ 100.

Tabela 2: Frequência relativa e percentagem de pessoas que tiveram infecção alimentar, em uma amostra de 140 pessoas, no restaurante da empresa.

Categoria	Frequência	Freq. Relativa	Percentual
Nenhum	110	0,7857	78,57
Leve	12	0,0857	8,57
Moderada	10	0,0714	7,14
Severa	8	0,0571	5,71
Total	140	1,00	100,00

No **R**, a função `prop.table()` gera os percentuais para uma tabela:

```
> x<-c(110,12,10,8)
> prop.table(x)

[1] 0.78571429 0.08571429 0.07142857 0.05714286

> prop.table(x)*100

[1] 78.571429  8.571429  7.142857  5.714286
```

Considere os dados do arquivo *cats* do pacote MASS:

```
> require(MASS)
> data(cats)
> attach(cats)
> summary(cats)

 Sex         Bwt              Hwt
 F:47   Min.   :2.000   Min.   : 6.30
 M:97   1st Qu.:2.300   1st Qu.: 8.95
        Median :2.700   Median :10.10
        Mean   :2.724   Mean   :10.63
        3rd Qu.:3.025   3rd Qu.:12.12
        Max.   :3.900   Max.   :20.50
```

Neste conjunto de dados, as colunas Bwt e Hwt representam o peso do corpo e do coração, respectivamente, de gatos do sexo Masculino (M) e feminino (F).

A função `tapply()` (*t* de table) pode ser utilizada para obtenção de estatísticas por grupos:

```
>  tapply(Bwt,Sex,mean) # média por grupos (sexo)
```

```
       F        M
2.359574 2.900000
```

Nesse exemplo, utiliza-se a variável resposta *Bwt*, agrupa-se por *Sex* e estima-se a média de cada grupo.

Tabelas de frequências podem ser obtidas fazendo-se,

```
>  table(Sex) #ocorrências por sexo
```

```
Sex
 F  M
47 97
```

Suponha uma nova variável, em que, se o peso do coração for maior ou igual a 9,5, o gato está apto e, caso contrário, estará inapto.

```
>  aptidao<-ifelse(Hwt>=9.5,"apto","inapto")
>  table(Sex,aptidao) #ocorrências de aptidão por sexo
```

```
   aptidao
Sex apto inapto
  F   23     24
  M   72     25
```

Observe que foi criada uma nova variável: aptidao.

A função `ifelse()` funciona da seguinte maneira: no exemplo, se `Hwt` é maior do que 9.5, o **R** insere a palavra *apto*, caso contrário, insere a palavra `inapto`. Essas informações são inseridas em uma nova variável chamada de *aptidao*.

Procure mais informações sobre a função `ifelse()`. Encontre exemplos, entenda seu funcionamento!

Pode-se criar uma tabela com essas novas informações:

```
> sex.t<-table(Sex,aptidao) #ocorrências de aptidão por sexo
> sex.t
```

```
   aptidao
Sex apto inapto
  F   23     24
  M   72     25
```

Para obtenção de uma soma marginal, faça:

```
> margin.table(sex.t,1)
```

```
Sex
 F  M
47 97
```

```
> margin.table(sex.t,2)
```

```
aptidao
  apto inapto
    95     49
```

Para obter frequências relativas, utilize:

```
> prop.table(sex.t,1)
```

```
   aptidao
Sex      apto    inapto
  F 0.4893617 0.5106383
  M 0.7422680 0.2577320
```

```
> prop.table(sex.t,2)
```

```
   aptidao
Sex      apto    inapto
  F 0.2421053 0.4897959
  M 0.7578947 0.5102041
```

Se for de interesse obter valores em percentual, multiplique por 100.

Para obter as proporções em função do total geral, basta utilizar:

```
> sex.t/sum(sex.t)

   aptidao
Sex      apto    inapto
  F 0.1597222 0.1666667
  M 0.5000000 0.1736111

> round(sex.t/sum(sex.t),2)#números com duas casas decimais

   aptidao
Sex apto inapto
  F 0.16   0.17
  M 0.50   0.17
```

A função `round()` é utilizada para gerenciar o número de casas decimais.

2.12.1 Classes de frequência

Quando a variável em estudo é uma variável quantitativa, chama-se distribuição de frequências o agrupamento de dados, formando classes com as respectivas frequências de cada classe e organizadas em uma tabela ou gráfico.

A construção de uma tabela de distribuição de frequências depende basicamente do número de classes. O número de classes pode variar em função de arbitrariedade mas, existe uma regra conhecida como Regra de Sturges, $c = 1 + 3,3 \log n$, onde c é o número de classes e n é o número de observações. No **R** essa opção é o padrão do software;

As classes representam os intervalos numéricos em que a variável quantitativa foi classificada. A Largura da classe é em geral determinada por

$$\frac{\text{max(x) - min(x)}}{\text{num. de classes (c)}}$$

.

Não é comum, mas a largura das classes podem ter tamanhos diferentes.

Exemplo

Considere o seguinte conjunto de 25 observações:

```
> d<-c(31,13,12,22,27,33,17,26,16,22,18,13,16,23,20,18,22,15,26,12,
+ 20,21,23,27,30)
```

Um `summary()` desse objeto indica que o menor valor é 12 e o maior valor é 33.

```
> summary(d)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  12.00   16.00   21.00   20.92   26.00   33.00
```

Assim, pode-se escolher (arbitrariamente), que a primeira classe inicie em 10 e a última classe termine em 35. Ainda, pode-se definir o número de classes. Nesse caso definiu-se como 5. Com o uso da função `seq()` pode-se gerar os intervalos de classe.

```
>  brk<-seq(10,35,5);brk  # define os intervalos de classe
```

```
[1] 10 15 20 25 30 35
```

```
>  classes<-c("10-14","15-19","20-24","25-29","30-35") # nomes das classes
```

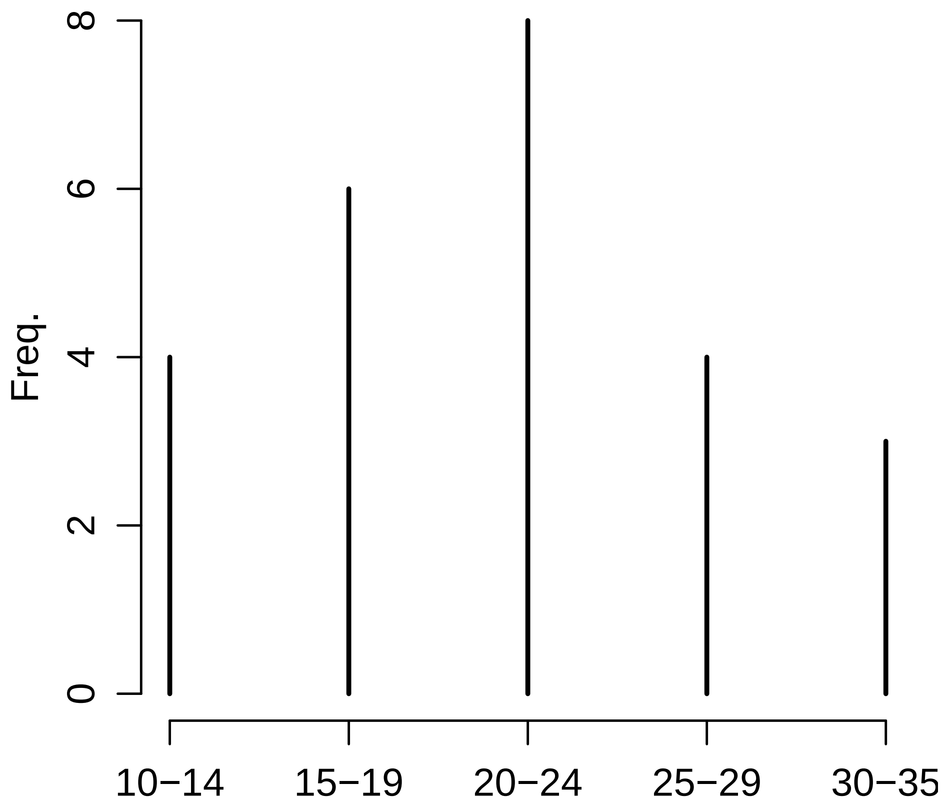
No **R**, uma tabela de frequência pode ser construída com o comando `table()`.

```
>  table(cut(d,breaks=brk,right=FALSE,labels=classes))
```

```
10-14 15-19 20-24 25-29 30-35
    4     6     8     4     3
```

Figura 1: Gráfico de uma tabela de frequências.

```
> plot(table(cut(d,breaks=brk,right=FALSE,labels=classes)),ylab="Freq.")
```



Veja o *help* da função `cut` para saber o que ela faz.

O resultado da tabela de frequência pode ser visualizado por um gráfico. Basta simplesmente pedir um `plot()` da tabela:

Da mesma forma, uma tabela com frequências relativas e percentagens pode ser construída para esses dados. Tente fazer!!

2.13 Gráficos

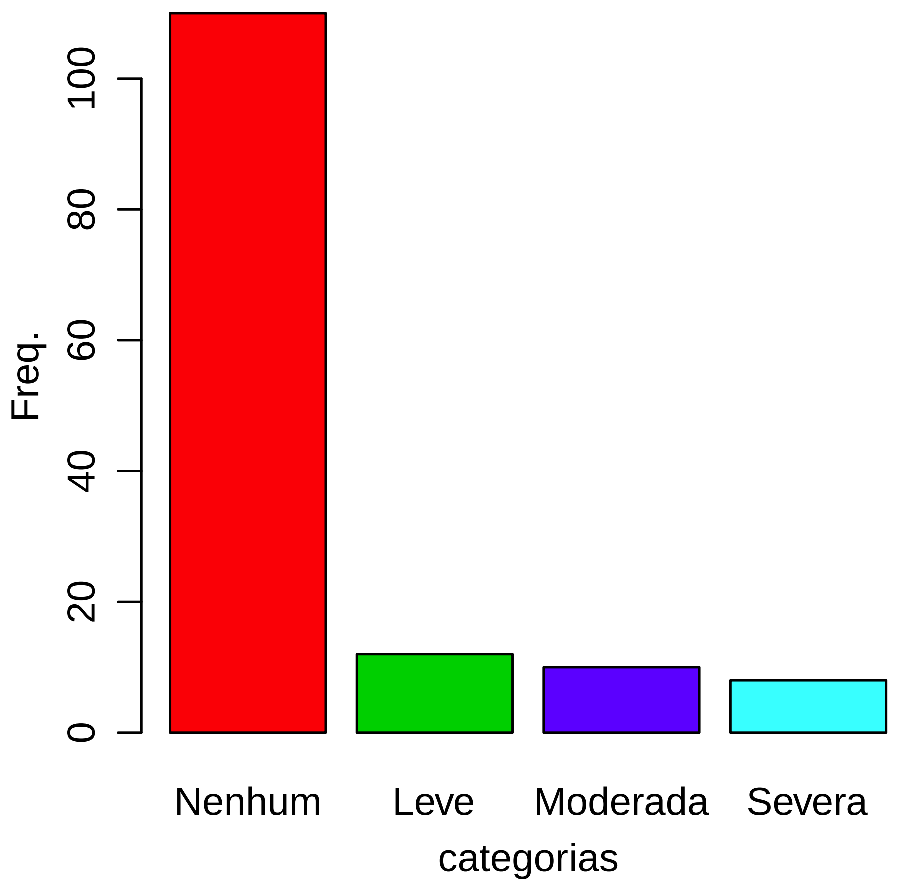
Variáveis qualitativas podem ser representadas por gráficos, tais como o de barras (Figura 2) e o de setores (pizza).

2.13.1 Gráfico de Barras

Para obter um gráfico de barras no **R** utilize o seguinte procedimento:.

Figura 2: Gráfico de barras com frequências absolutas.

```
> x<-c(110,12,10,8)
> barplot(x,ylab="Freq.",xlab="categorias",
+ names=c("Nenhum","Leve","Moderada","Severa"),col=2:5)
```



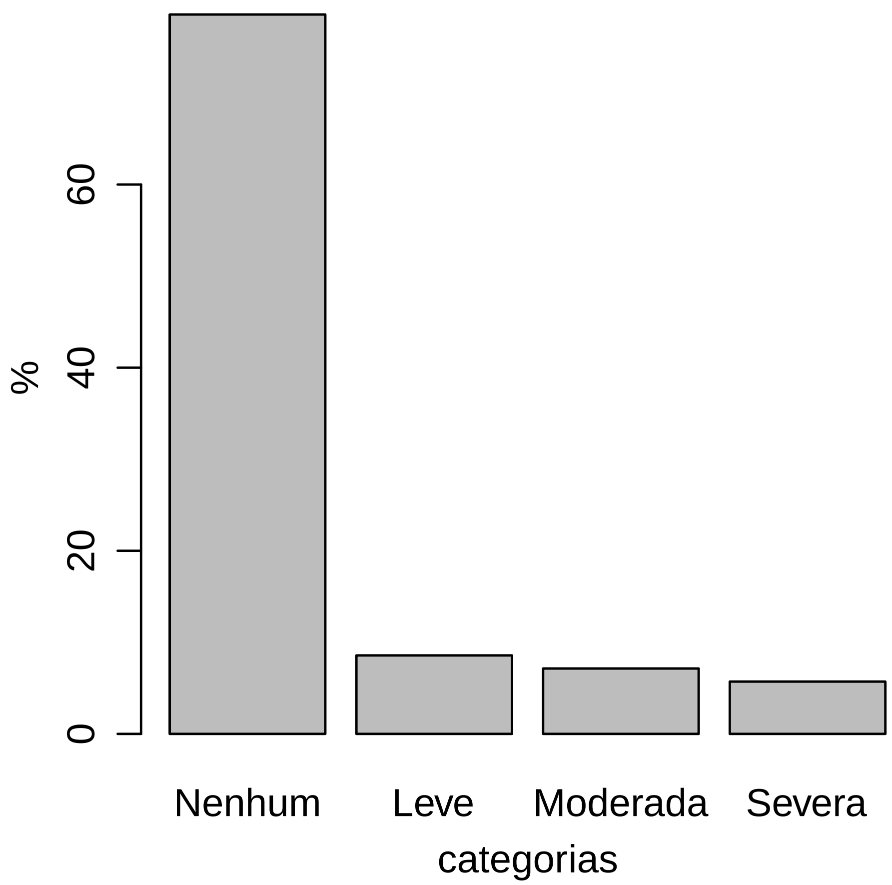
O gráfico de barras pode ser apresentado, utilizando-se as frequências relativas ou percentagens, quando de interesse (Figura 3).

2.13.2 Gráfico de Setores (pizza)

Um gráfico de setores é um círculo dividido em partes que representam as frequências relativas ou percentagens de cada classe ou categoria. . Para obtê-lo, use:

Figura 3: Gráfico de barras com frequências relativas.

```
> xp<-prop.table(x)*100
> barplot(xp,ylab="%",xlab="categorias",
+ names=c("Nenhum","Leve","Moderada","Severa"))
```



2.13.3 Histograma

Histograma é um gráfico que representa a distribuição de frequência absoluta, relativa ou percentual. Observe que as barras estão juntas. Isso ocorre porque um histograma é utilizado para representar uma variável quantitativa contínua.

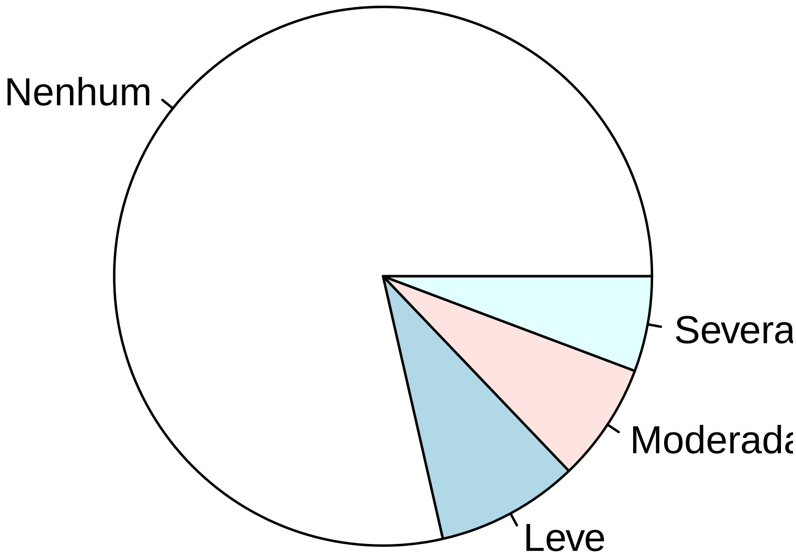
Experimente refazer esse último gráfico sem o argumento `breaks`.

Por *default*, o **R** utiliza a frequência absoluta para construir o histograma. Se tiver interesse em representar as frequências relativas, utilize a opção `freq=FALSE` nos argumentos da função `hist()`.

Observe que no eixo Y, onde antes aparecia "Frequency" agora aparece o texto

Figura 4: Gráfico de Setores.

```
> names(x)<-c("Nenhum","Leve","Moderada","Severa")
> xp<-prop.table(x)*100
> pie(xp,labels=names(x))
```



"Density"!

Em um histograma, pode-se inserir a linha que representa a densidade dos dados utilizando-se a função `lines()` junto com a função `density()`, da seguinte maneira:

A função `rug()` insere no histograma "riscos" indicando a frequência de observações em cada classe.

2.13.4 Ramo e folhas

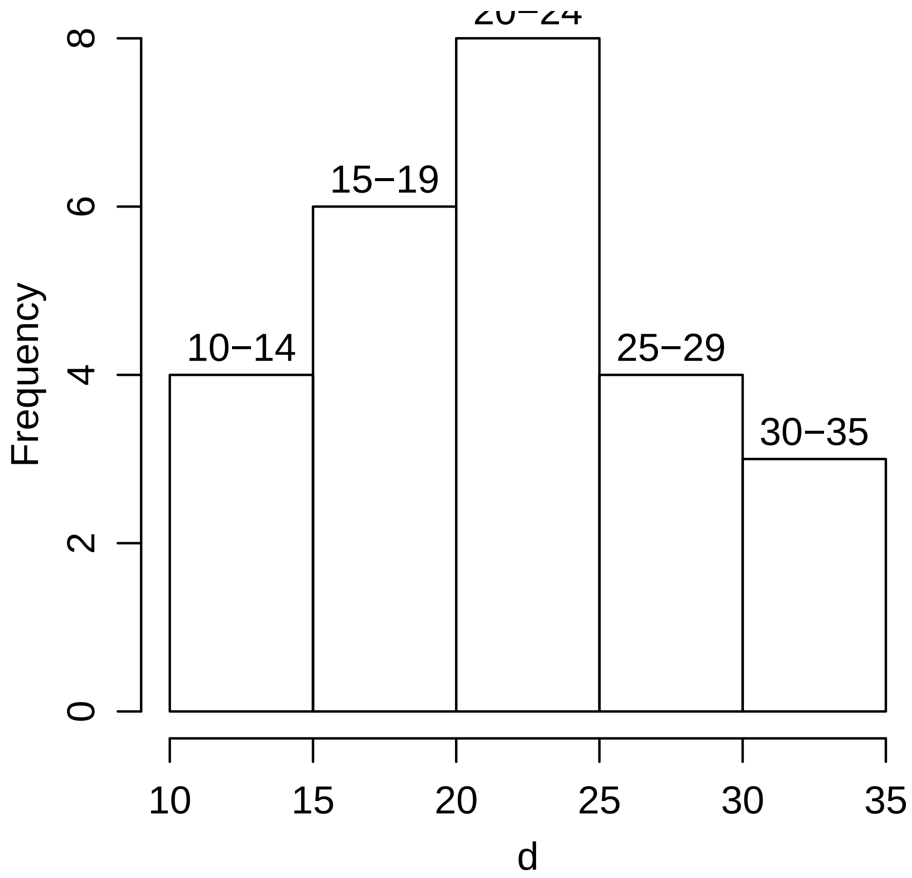
O gráfico de Ramo e folhas é útil para representar o comportamento de variáveis. Além de indicar a forma da distribuição ele mostra a frequência de cada observação.

Figura 5: Histograma com definição de classes.

```
> brk<-seq(10,35,5);brk  # define os intervalos de classe

[1] 10 15 20 25 30 35

> d<-c(31,13,12,22,27,33,17,26,16,22,18,13,16,23,20,18,22,15,26,12,
+ 20,21,23,27,30)
> classes<-c("10-14","15-19","20-24","25-29","30-35") # nomes das classes
> hist(d,breaks=brk,right=F,labels=classes,main="")
```



Esse gráfico é construído colocando-se em uma coluna (Ramo), por exemplo, os números interios de uma variável e em outra coluna os números decimais (folhas):

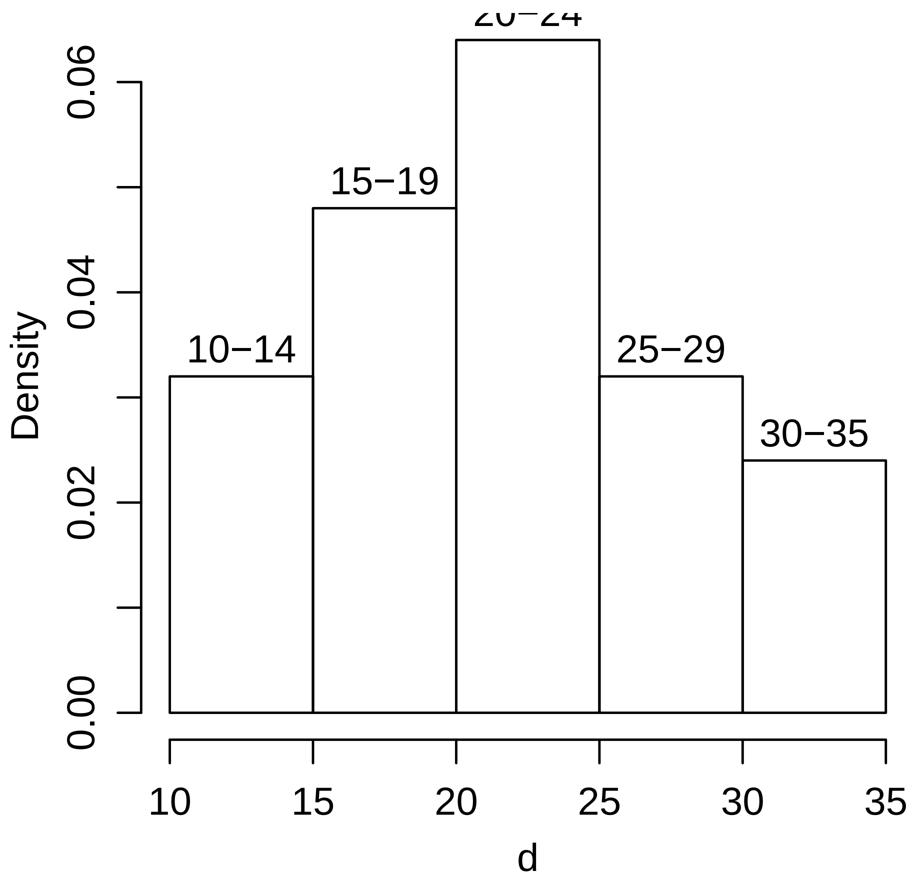
Por exemplo para a variável d, um ramo e folhas é construído da seguinte maneira:

```
> stem(d)

  The decimal point is 1 digit(s) to the right of the |
```

Figura 6: Histograma com frequências relativas.

```
> hist(d,breaks=brk,freq=FALSE,right=F,labels=classes,main="")
```




```
1 | 2233
1 | 566788
2 | 00122233
2 | 6677
3 | 013
```

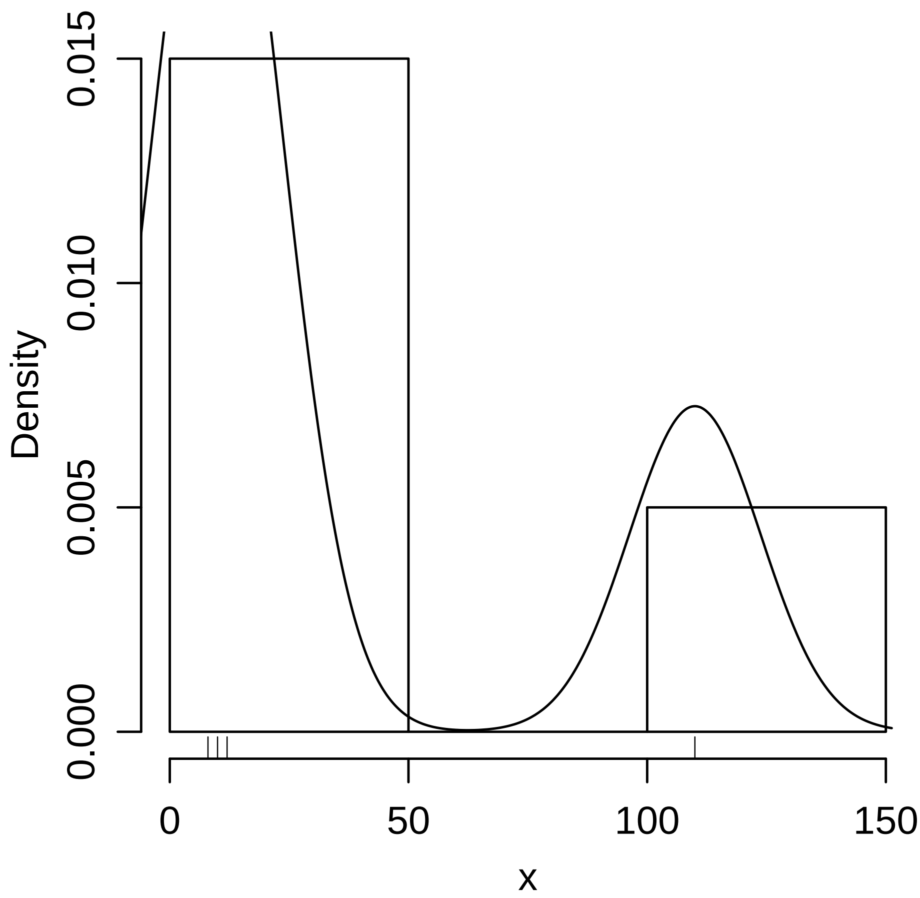
2.13.5 Construindo o Box-plot

O box-plot é um gráfico que mostra a posição central, dispersão e simetria dos dados de uma amostra .

Considere o seguinte conjunto de dados: 17, 22, 23, 27, 29, 32, 38, 42, 46, 52, 60,

Figura 7: Histograma com a linah de densidade.

```
> hist(x,prob=T,main="")
> lines(density(x)) # insere a linha
> rug(x)            # insere uma barra com freq. de pontos
```



92

Para construir o box-plot dessas observações, devemos seguir os seguintes passos:

Passo 1: Calcular a mediana , o primeiro e terceiro quartis, e a amplitude interquartílica.

```
> box<-c(17,22,23,27,29,32,38,42,46,52,60,92)
> quantile(box,type=2)

  0%  25%  50%  75% 100%
  17   25   35   49   92

> AIQ<-quantile(box,.75,type=2)-quantile(box,.25,type=2)
> AIQ
```

```
75%
 24
```

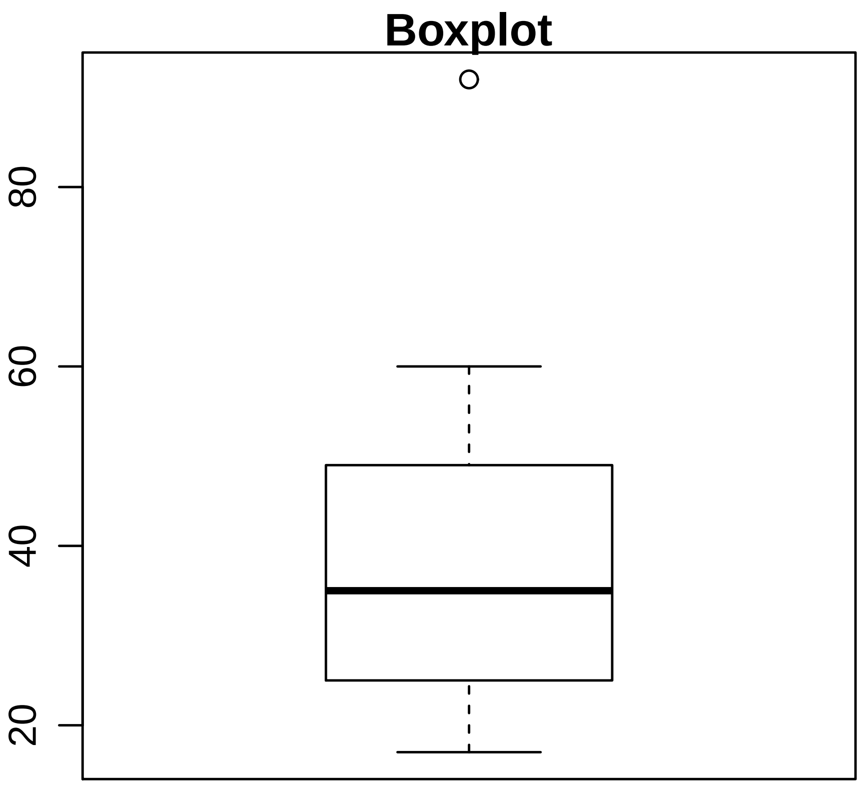
Passo 2: Calcular 1,5 $\times$ AIQ =1,5 $\times$ 24=36 e,
Limite inferior = Q1-36=25-36=-11
Limite superior = Q3+36=49+36=85

Passo 3: Encontrar o menor valor e o maior valor dentro dos limites inferiores e superiores, respectivamente: menor=17 e maior=60;

Passo 4: Construir o gráfico (Figura 8). No **R** utilize:

Figura 8: Boxplot individual.

```
> boxplot(box,main="Boxplot")
```



Tente identificar no gráfico os valores obtidos manualmente. A linha central é a mediana!

Em Portugal, o boxplot é chamado de caixa de bigodes!

Além do gráfico, a função `boxplot()` também retorna as estatísticas obtidas para construção do gráfico. Essas informações podem ser recuperadas com a função `boxplot.stats()`:

```
> boxplot.stats(box)

$stats
[1] 17 25 35 49 60

$n
[1] 12

$conf
[1] 24.05344 45.94656

$out
[1] 92
```

O Box plot também pode ser utilizado para comparar grupos.

Considere os conjuntos anteriores `x` e `box`:

2.14 Explorando graficamente cats

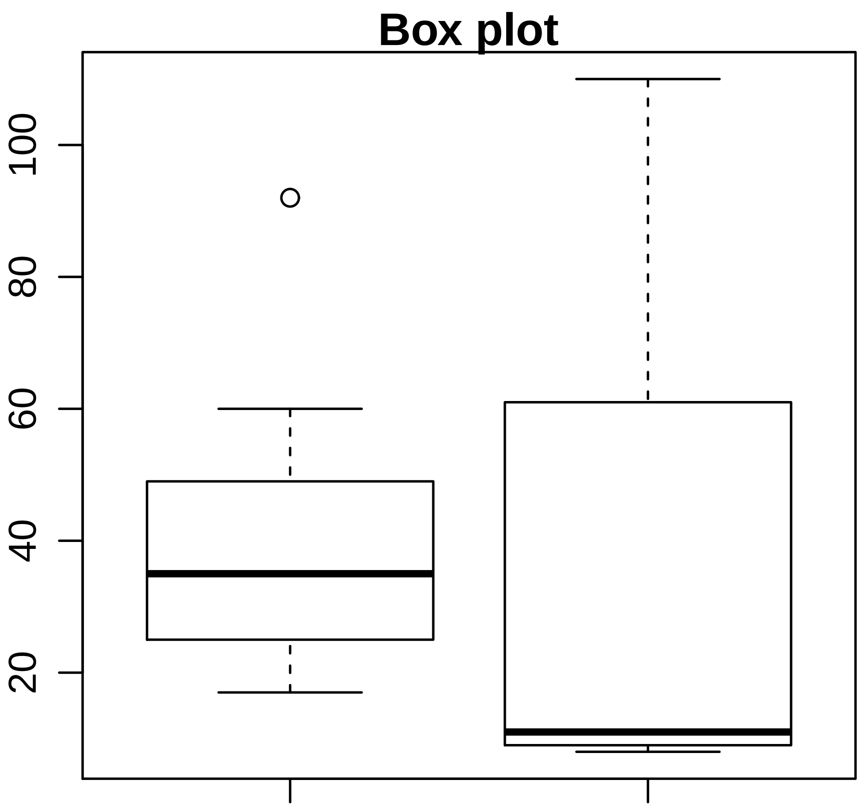
Vamos construir alguns gráficos utilizando os dados sobre gatos (`cats`) disponível no pacote `MASS`.

```
> require(MASS)
> data(cats)
> attach(cats)
> aptidao<-ifelse(Hwt>=9.5,"apto","inapto")
> sex.t<-table(Sex,aptidao) #ocorrências de aptidão por sexo
> sex.t

   aptidao
Sex apto inapto
  F   23     24
  M   72     25
```

Figura 9: Boxplot por grupos.

```
> boxplot(box,x,main="Box plot")
```



Um boxplot para avaliar o comportamento de `Hwt` em função do sexo do gato (Figura 10):

Um gráfico de barras (Figura 11):

O mesmo gráfico com as barras invertidas (Figura 12) :

Inserindo uma legenda (Figura 13):

Um gráfico de setores(Figura 14):

Procure outras opções para construção desses gráficos. Por exemplo, tente alterar cores, inserir texto, títulos etc.

Não esqueça de retirar do caminho de procura o objeto `cats`.

Figura 10: Boxplot para comparar o Hwt para diferentes Sexos de gatos.

```
> boxplot(Hwt~Sex,col=2:3)
```

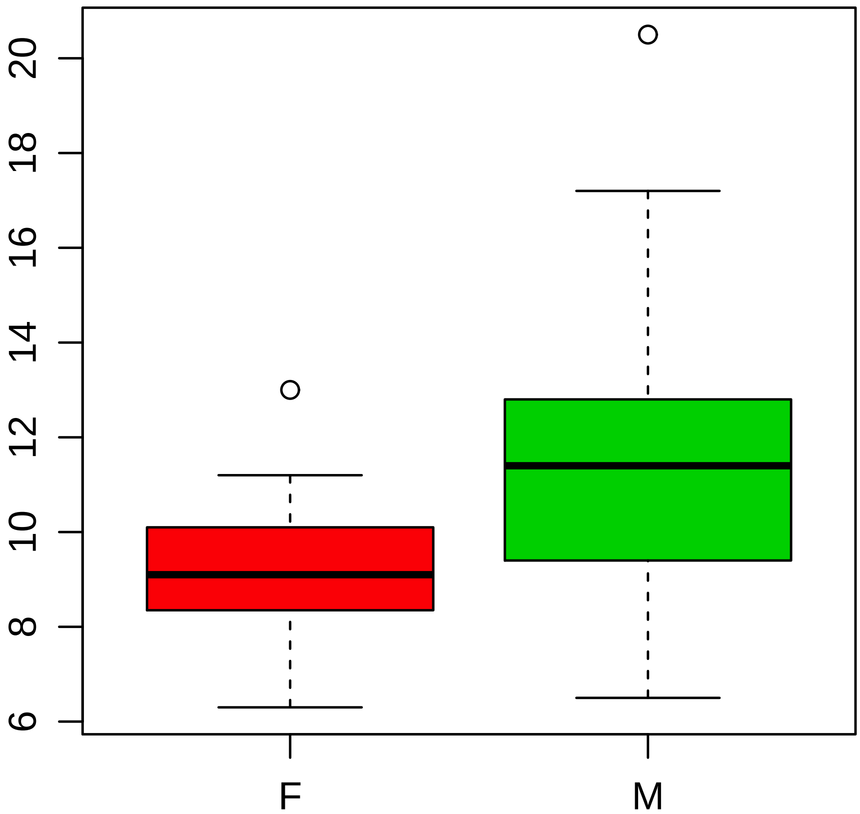


Figura 11: Gráfico de barras normal.

```
> barplot(sex.t, col=4:5)
```

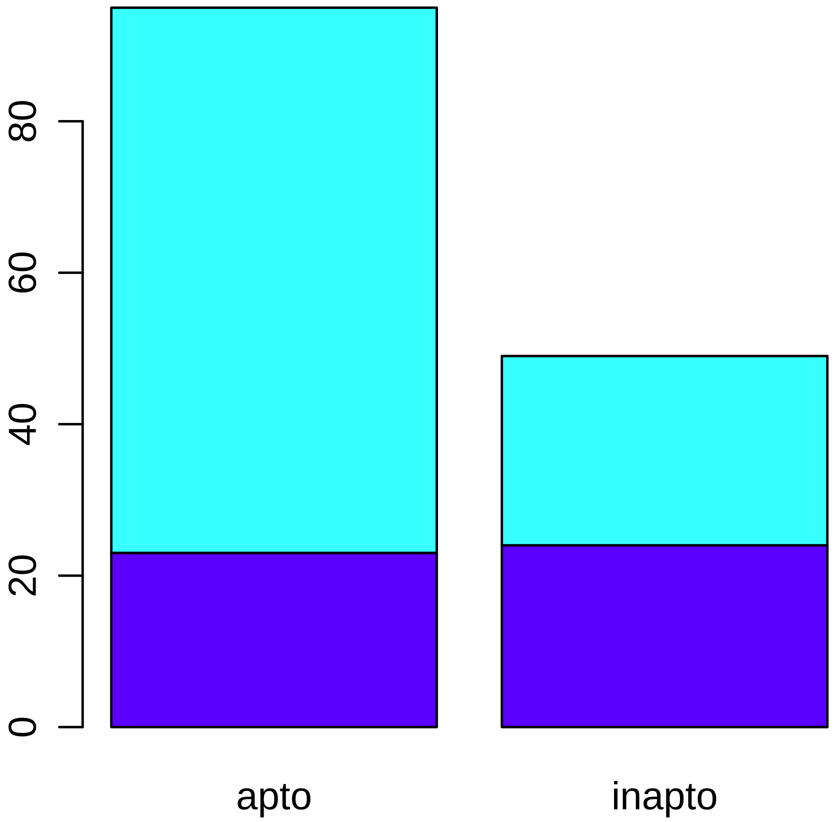


Figura 12: Gráfico de barras invertido.

```
> barplot(t(sex.t),col=5:4) # o "t" inverte os valores da barra
```

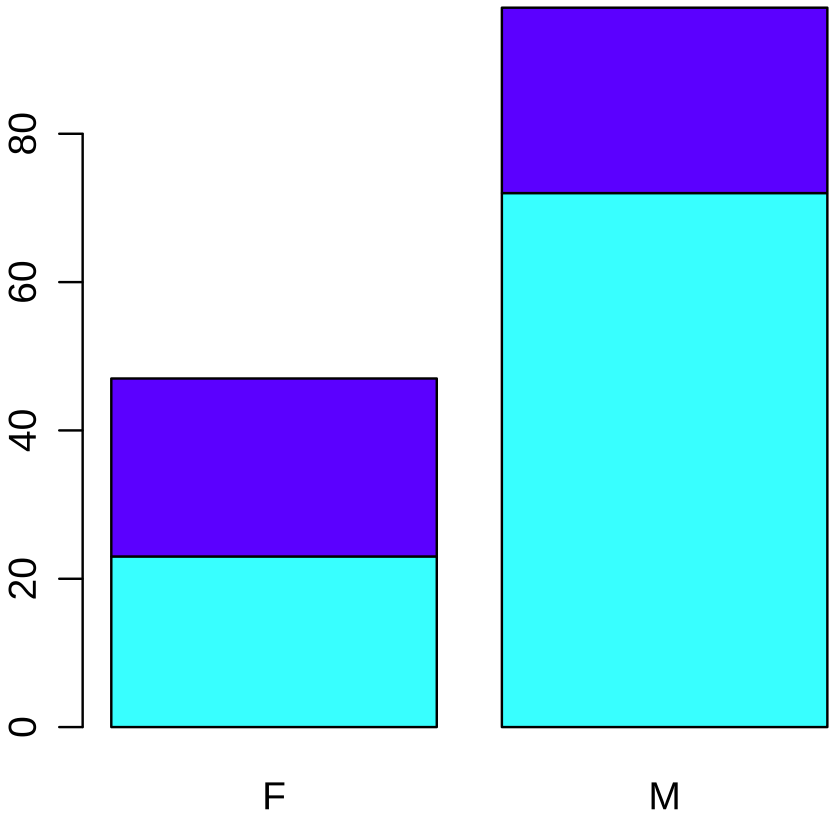


Figura 13: gráfico debarras com legenda.

```
> barplot(sex.t,beside=T,legend.text=rownames(sex.t), col=c("white","gray"))
```

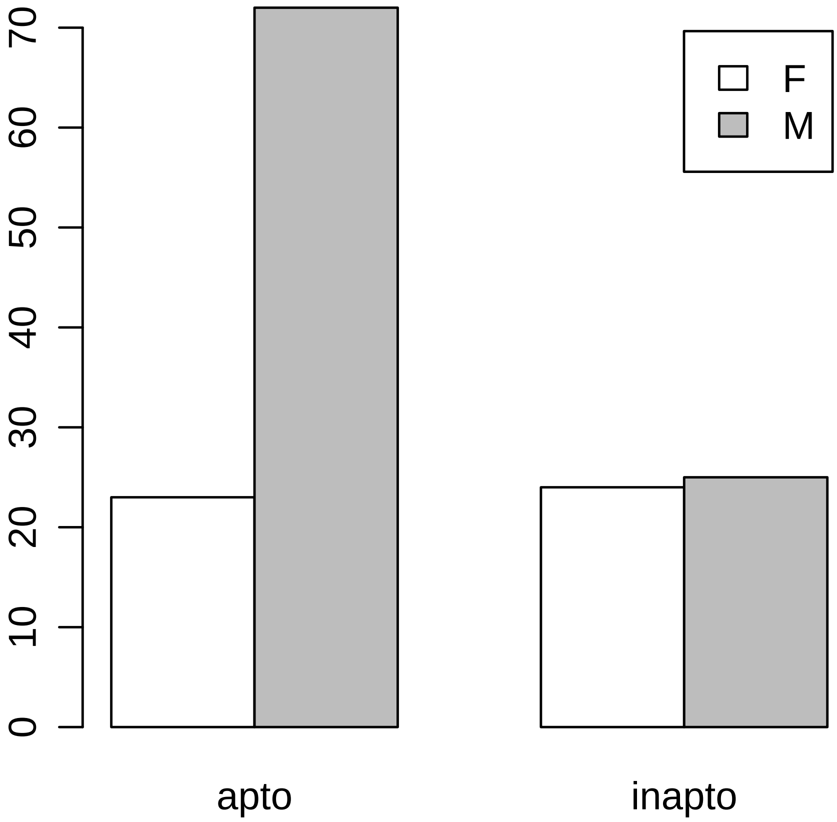
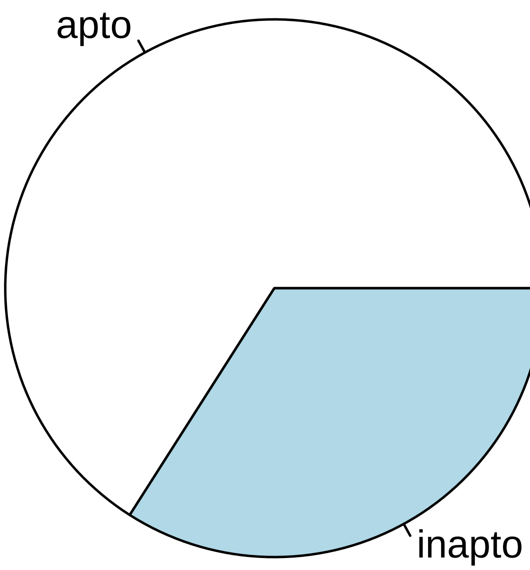


Figura 14: Gráfico de setores.

```
> pie(margin.table(sex.t,2)) # um gráfico de setores
```



```
> detach(cats)
```

3 Como escrever funções

Em algumas situações, algumas tarefas podem ser executadas por funções criadas pelo próprio usuário. No **R** é possível criar essas funções. Por exemplo, pode haver interesse que uma função faça a análise descritiva utilizando estatísticas definidas pelo usuário. A sintaxe de criação é a seguinte:

```
> minha.f<-function(argumentos){expressão}
```

Evite colocar nomes de funções que já existam. Isso pode gerar conflitos no **R**. O nome de uma função não pode começar com números.

3.0.1 Alguns exemplos

Por exemplo um função que forneça algumas estatísticas descritivas univariadas chamada *desc* pode ser criada no **R**. Neste caso serão necessárias algumas linhas de comandos:

```
>  desc<-function(dados)
+ {
+ med<-median(dados)
+ max<-max(dados)
+ min<-min(dados)
+ soma<-sum(dados)
+ print(c(mediana=med,maximo=max,minimo=min,soma=soma))
+ }
```

Execute essas linhas de comando sequencialmente no **R**. Você criará uma função chamada `desc()`.

Nesse exemplo, *dados* é o argumento necessário para que a função seja executada, ou seja, um vetor ou matriz de dados.

Execute a função sobre um objeto, por exemplo:

```
> dados<-c(12,45,87,89,52,41,36)
> desc(dados)
```

```
mediana  maximo  minimo    soma
     45      89      12     362
```

Uma função um pouco mais elaborada:

Suponha que se queira contar o número de NA's (abreviação de Not Available ou, em português, dados não observados) em um vetor de dados:

```
> num.nas<-function(x)sum(is.na(x))
```

`is.na()` é uma função que verifica se uma observação é um `NA`.

```
> meu.vetor<-1:10
> meu.vetor[1:3]<-NA #acrescenta três NA's ao vetor
> num.nas(meu.vetor)
```

```
[1] 3
```

No exemplo a seguir, quando p é Falso, a função conta o número de NA's. Quando p é Verdadeiro, a função calcula a proporção de NA's.

```
> p.nas<-function(x,p)
+   {
+     if(p)
+       return(mean(is.na(x)))
+     else
+       return(sum(is.na(x)))
+   }
```

```
> p.nas(meu.vetor,FALSE)
```

```
[1] 3
```

```
> p.nas(meu.vetor,TRUE)
```

```
[1] 0.3
```

Pode-se definir um valor padrão (*default*) para a função:

```
> p2.nas<-function(x,p=FALSE)
+    {
+      if(p)
+        return(mean(is.na(x)))
+      else
+        return(sum(is.na(x)))
+    }
```

Aplicando-se a nova função, tem-se:

```
> p2.nas(meu.vetor)

[1] 3

> p2.nas(meu.vetor,TRUE)

[1] 0.3

> p2.nas(meu.vetor,T)

[1] 0.3
```

3.1 Exercícios - Módulo 2

Não é necessário entregar esse exercício. Ele serve apenas para você praticar o que aprendeu nesse módulo.

Utilize os dados biométricos dos participantes do curso:

```
> dados<-read.csv("http://www.ufpr.br/~aanjos/ead/dados/biom.csv",h=T,dec=',')[,-1]
```

1. Estime a média, variância e desvio padrão para a variável `Idade`;
2. Estime a média, variância e desvio padrão para a variável `Idade` separando por `Sexo`;
3. Crie uma nova variável para agrupar pessoas com idade acima e abaixo da mediana;
4. Estime a média e o desvio padrão da variável `Peso` das pessoas com idade acima e abaixo da mediana de `Idade`.
5. Construa uma tabela de frequências para as variáveis `peso` e outra para `altura`;
6. Construa um histograma para a variável `Idade`;
7. Construa um gráfico de barras para a variável `Sapato`;
8. Construa um boxplot para a variável Peso;
9. Construa um boxplot para a variável Peso, considerando os grupos com `Idade` abaixo e acima da mediana.
10. Construa um gráfico de setores para as variáveis `Sapato` e `Sexo`.

Experimente utilizar: `summary(dadosbiom)`

Explore outras variáveis!!