O software R como instrumento de ensino em Estatística Básica

Gastão Coelho Gomes, gastao@im.ufrj.br
João Ismael Damasceno Pinheiro jismael@im.ufrj.br
Sonia Baptista da Cunha, sonia@im.ufrj.br
Santiago Ramírez Carvajal sramirez@oi.com.br

http://www.r-project.org

"Estatística Básica: A Arte de Trabalhar com Dados",
Ed. Campus-Elsevier. Rio de Janeiro, (2008).
Pinheiro J. I. D.; Cunha, S.; Ramirez, S. C.; e Gomes, C. G.

Porque do minicurso

- A Estatística é uma ferramenta importante para se obter informação de uma massa de dados.
- O R é um pacote que oferece várias funções, já implementadas, dos mais variados métodos estatísticos. Além disso é, também, um ambiente de programação onde se pode usar o que de bom ele já contem para se desenvolver novas implementações.
- Ambos, a Pesquisa Operacional e o processo de desenvolvimento de novos aplicativos em Estatística, podem se beneficiar dessa interação.
- O que propomos é discutir as aplicações no R dos métodos básicos de análise estatística.

Assuntos abordados no minicurso

1) Cap 1: Análise Exploratória de Dados:

No que se refere a **medidas univariadas** examinaremos estatísticas de tendência central, localidade e dispersão (no R: mean, median, var, fivenum, summary e quantile); gráficos de distribuições (no R: barplot, pie, hist, stem e boxplot). Quanto às **medidas bivariadas** examinaremos a interdependência através da covariância, correlação gráfico de dispersão e tabelas de contigências (no R: var, cor, plot, table). Será também feita uma introdução à regressão linear e ao **método de mínimos quadrados** (no R lsfit e ls.print)

2) Cap 2-a Simulação do conceito freqüentista de probabilidade:

Método de Monte Carlo: Através de exemplos de jogos "calcularemos" **probabilidades via simulação**, examinando a estabilidade da aproximação.

3) Cap 2-b: Variáveis Aleatórias:

Examinaremos no R os modelos probabilísticos mais comuns de variáveis aleatórias **discretas**: Binomial, Hipergeométrica, Poisson; e variáveis aleatórias **contínuas**: exponencial, uniforme, Normal e suas derivadas t-Student, Qui-quadrada e F. No R veremos o efeito da primeira letra a ser usada nos comandos relativos aos modelos probabilísticos (p-probability, d-density, q-quantile e r-random).

4) Cap 2-c: Simulação e o Teorema Central do Limite:

Através de simulação será estudado o Teorema Central do Limite: **O efeito do tamanho amostral** e da **população de onde a amostra é extraída** na aproximação da distribuição da média amostral de x pela distribuição Normal.

5) Cap 3-a: Intervalo de confiança:

Serão feitas simulações para o entendimento do conceito de intervalo de confiança, através da geração, por simulação, de várias amostras e o posterior exame dos intervalos de confiança construídos a partir de cada uma dessas amostras.

6) Cap 3-b: Testes de Hipóteses:

Serão recordados os principais componentes dos testes de hipóteses, erros tipos I e II com as correspondentes probabilidades, **p-valor**. Estudaremos o **teste t de Student**, tanto pareado com não pareado, para **comparação de duas populações**, teste quiquadrado para **independência**, e **análise de variância** (no R t.test, chisq.test, aov)

Trabalhando no R

Usaremos aqui três tipos de variáveis:

constantes ou vetores

São os tipos de armazenamento mais básico de uma variável. Se desejarmos que numa variável x esteja a altura (em cm) de 10 indivíduos, faremos:

```
> x = c(172,167,189,157,163,  156,201,186,179,152)
```

Observe que o sinal “>” é um *prompt* do R; o comando “c()”, *combina* uma seqüência de valores numa variável, que aqui foi chamada de “x”; o comando “=” é de atribuição.

Experimente os comandos: > y= 1:10; > x*2; > x+2; >x+y; >x*y; >z=x+y; ; > ?c

matrizes,

São geralmente bancos de dados, com n linhas (as observações) e p colunas (as variáveis). Todas devem ser da mesma característica, geralmente numéricas. Se desejarmos que numa variável “ap” esteja na primeira coluna a altura (em cm) e na segunda o peso (em kg) de 10 indivíduos:

```
> ap = matrix(c(172,167,189,157,163, 156,201,186,179,152,
                68,63,89,90,75, 63,95,120,80,60), 10,2) # peso e altura
```

Observe que o comando “matrix” arruma os dados de um vetor numa matriz o *default* é entrar com o vetor por colunas; os parâmetros “**10, 2**” indicam, respectivamente, o número de **linhas e colunas**; o comando # indica que o que vem depois, na mesma linha, é interpretado como uma observação e não é considerado.

Experimente os comandos: >?matrix; > pa[1,2]; > pa[1,]; > pa[,1];

Trabalhando no R

data frame

São usados para armazenamento de bancos de dados, com n linhas (as observações) e p colunas (as variáveis). Podem não ser da mesma característica, misturando, alfanuméricos com numéricos e fatores. Este comando seria útil, por exemplo, para ler um banco de dados gerado no Excel,
No R um data frame seria lido pelo comando *read.table*. Vamos ler a tabela 1.2, pag 7 do livro [1], para tanto foi gerado um arquivo no Excel de nome tab1_2.tex.

Apresentamos aqui a 3 linhas iniciais dos dados de um total de 45, a primeira linha (apresentada aqui em duas) corresponde aos nomes das variáveis.
ID, CATEG, IDADE, PESO, ALTURA, IMC, Classe_IMC, CINTURA, ...
ID1,A,61,58.2,154.0,24.5,normal,87,109,0.80,MR
ID2,S,69,63.0,152.0,27.3,sobrepeso,89,104,0.86,GR
ID3,S,61,70.1,158.0,28.1,sobrepeso,106,123,0.86,GR

Para armazenarmos os dados no objeto tab1.2, usaremos o comando:

- tab1.2=read.table("f:\\SBPO2010R\\tab1_2.txt", header = T, sep = ",")

Observe que header = T serve para indicar que existe uma linha com os nomes das variáveis (T significa True) e sep indica o separador, no caso vírgula.

Experimente os comandos:
>?read.table; tab1.2[,2],
> attach(tab1.2); CATEG
> tab1.2[,3]; IDADE

Cap. 1: Analise Exploratória de Dados (AED)

Análise Exploratória é um conjunto de técnicas de tratamento de dados que, sem implicar em uma fundamentação matemática mais rigorosa, nos ajuda a tomar um primeiro contato com a informação disponível.

Em um levantamento de dados, a respeito de um determinado assunto, eles costumam ser representados em uma tabela de dados. Em uma tabela de dados cada linha corresponde a uma observação e cada coluna corresponde a uma variável.

As variáveis podem ser:

Qualitativa nominal ou categórica - seus valores possíveis são diferentes categorias não ordenadas.

Qualitativa ordinal - seus valores possíveis são diferentes categorias ordenadas.

Quantitativa discreta - seus valores possíveis são resultados de um processo de contagem.

Quantitativa contínua - seus valores possíveis podem ser expressos através de números reais.

Para descrever o comportamento de uma variável é comum apresentar os valores que ela assume organizados sob a forma de tabelas de frequência e gráficos. Os gráficos mais comuns para representarem variáveis qualitativas são os gráficos de barras e os gráficos de setores.

Usar, para uma variável x que deve ser agrupada, os comandos:***: barplot(table(x)); pie(table(x)).*** *Os principais argumentos desses comandos são:*

Cap1–AED: barplot, pie

barplot(x, beside=F, horiz=F, xlab=, xlim=, col= , space= ,...)

Onde:

x: um vetor de quantidades positivas. Os valores em 'x' representam a proporção, obrigatório

***beside**: se as barras serão de lado ou empilhadas, essa é uma variável do tipo sucesso(T,true) ou fracasso (F, false) o default é F. Como exemplo olhar o apêndice Figura 2.2.*

***xlab**: corresponde ao título da variável x (não obrigatório.), o mesmo para ylab.*

***xlim**: dois valores que correspondem aos limites no gráfico da variável x. (ylim).*

***space** - quantidade de espaço à esquerda antes de cada barra. Se matrix podem ser 2 valores, o primeiro barras do mesmo grupo e o segundo entre grupos.*

***col**: vetor informando as cores das barras. Ver apêndice*

pie(x, labels = names(x), edges = 200, col=NULL...)

Onde:

***x**: um vetor de quantidades positivas. Os valores em 'x' representam as proporções,*

***labels**: um vetor de caracteres fornecendo nomes para os setores. (não obrigatório.)*

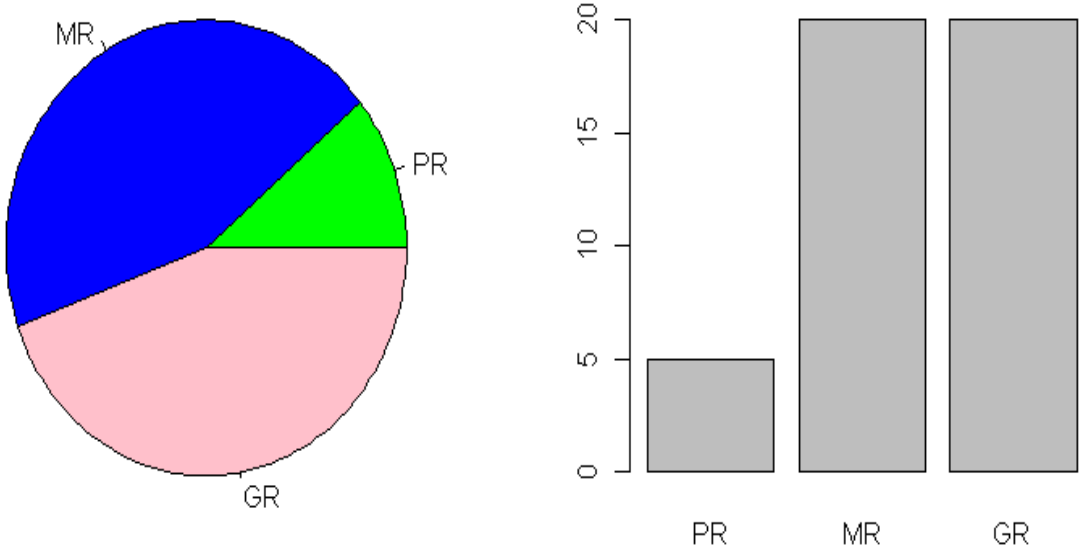
***edges**: um inteiro. A linha do círculo é aproximada por um polígono com este número de.lados.*

***col**: vetor informando as cores das barras*

Cap1-AED – ex: table, names, par, pie, barplot

Exemplo:

```
RCQ=c(.80,.86,.86,.90,.82,.95,.92,.83,.83,.89,.81,.84,.78,.81,.89,
    .87,.74,.80,.91,.86,.85,.84,.85,.74,.76,.83,.80,.78,.85,.87,
    .68,.83,.87,.87,.87,.89,.87,.88,.88,.89,.78,.77,.78,.89,.84)        #digitar RCQ
rcq=rep(2,45); rcq[RCQ < .78]=1; rcq[RCQ > .85]=3                        #codificar
rcq.t= table(rcq)                                                        #tabular
names(rcq.t)=c("PR","MR","GR")                                  #nomear as categorias
par(mfrow=c(1,2))                              # matrix de graficos (1 linha e 2 colunas)
pie(rcq.t, radius=1.2, col=c("green","blue","pink"))             #graf. de setor
barplot  (rcq.t)                                                 #graf. de barras
```

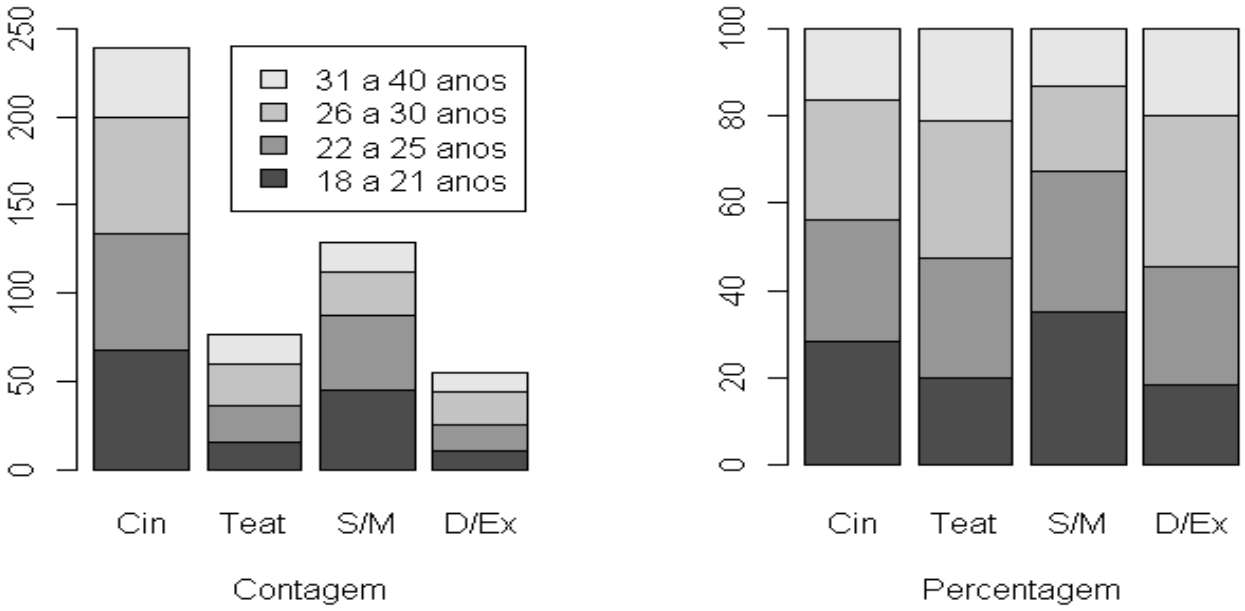


Cap1-AED – ex: barplot (beside=F)

Pag 48 – Figura 2.2

```
mat= matrix(c(68,15,45,10,  66,21,42,15,   66,24,25,19,   39,16,17,11),4,4,byrow=T)
rownames(mat)=c("18 a 21 anos", "22 a 25 anos", "26 a 30 anos", "31 a 40 anos")
colnames(mat)=c("Cin","Teat","S/M","D/Ex")
mat1=prop.table(mat, 2)

par(mfrow=c(1,2), mai=c(.1,.1,.1,.1), mar=c(5, 4, 2, 2) )
barplot(mat,beside=F, ylim=c(0,250),    legend = c("18 a 21 anos", "22 a 25 anos",
      "26 a 30 anos", "31 a 40 anos"),  xlab="Contagem")
#
barplot(mat1, beside=F, xlab="Percentagem")
```

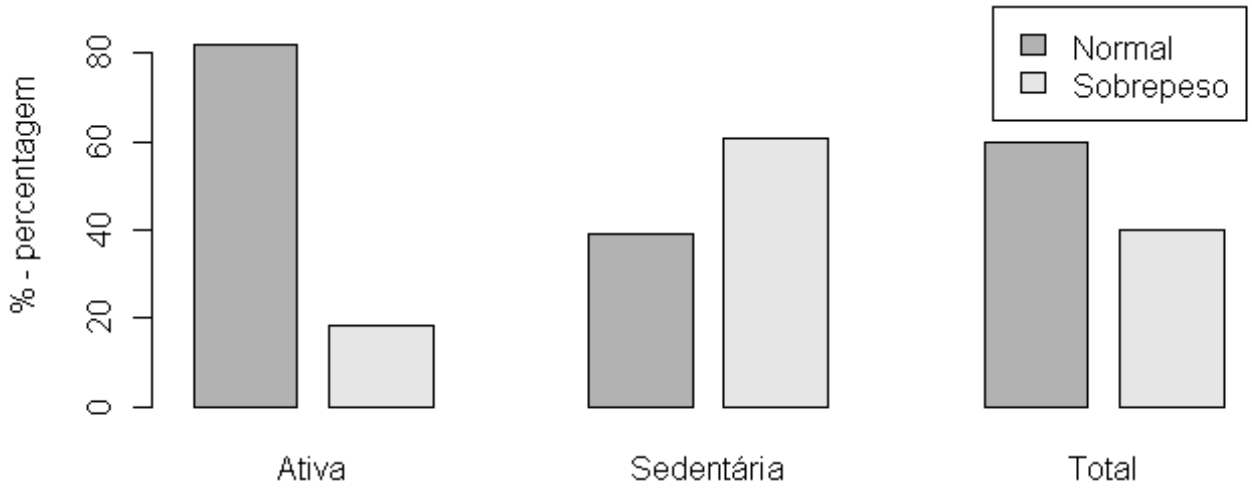


Cap1-AED – ex: barplot (beside=T)

Pag 44 – Figura 2.1

```
mat=matrix(c(81.82,39.13,60,18.18,60.87,40),3,2)
rownames(mat)=c("Ativa","Sedentária","Total")
colnames(mat)=c("Normal","Sobrepeso")

barplot(t(mat), beside = TRUE, space=c(.3,1.5),
    col=gray(c(.7,.9)),
    legend = c("Normal","Sobrepeso"), ylim = c(0, 95), ylab="% - percentagem")
```



Cap1–AED: stem, cut, table

Para as variáveis quantitativas, os mais usados são os Histogramas e os Diagramas Ramo-folhas, cujos comandos são **>hist(x) > stem(x).** Existe também um comando chamado **>cut**, que classifica uma variável numérica. *Os principais argumentos do comando hist são:*

hist(x, breaks= , freq =NULL, right=T, col=NULL, main=, xlim=range(breaks), ylim=NULL, xlab=xname, ylab,...) *Onde:*

x: a variável numérica a ser discretizada, (argumento obrigatório)

breaks: vetor com os limites das classes

freq: variável lógica, se T (True) corresponde à contagem de cada classe; se F (False) equivale a densidade de probabilidade, a área total sob a curva (retângulos) teria soma 1.

right: variável lógica, se T as classes são fechadas à direira; se F são fechadas à esquerda.

col: vetor de cores, pode ser uma única.

main: título principal

xlab e ylab: rótulos dos eixos x e y respectivamente

xlim e ylim: Dois valores limites para o gráfico de cada uma das variáveis

cut(x, breaks, right = T, ...) *Onde:*

x: a variável numérica a ser discretizada, (argumento obrigatório)

breaks: vetor com os limites das classes

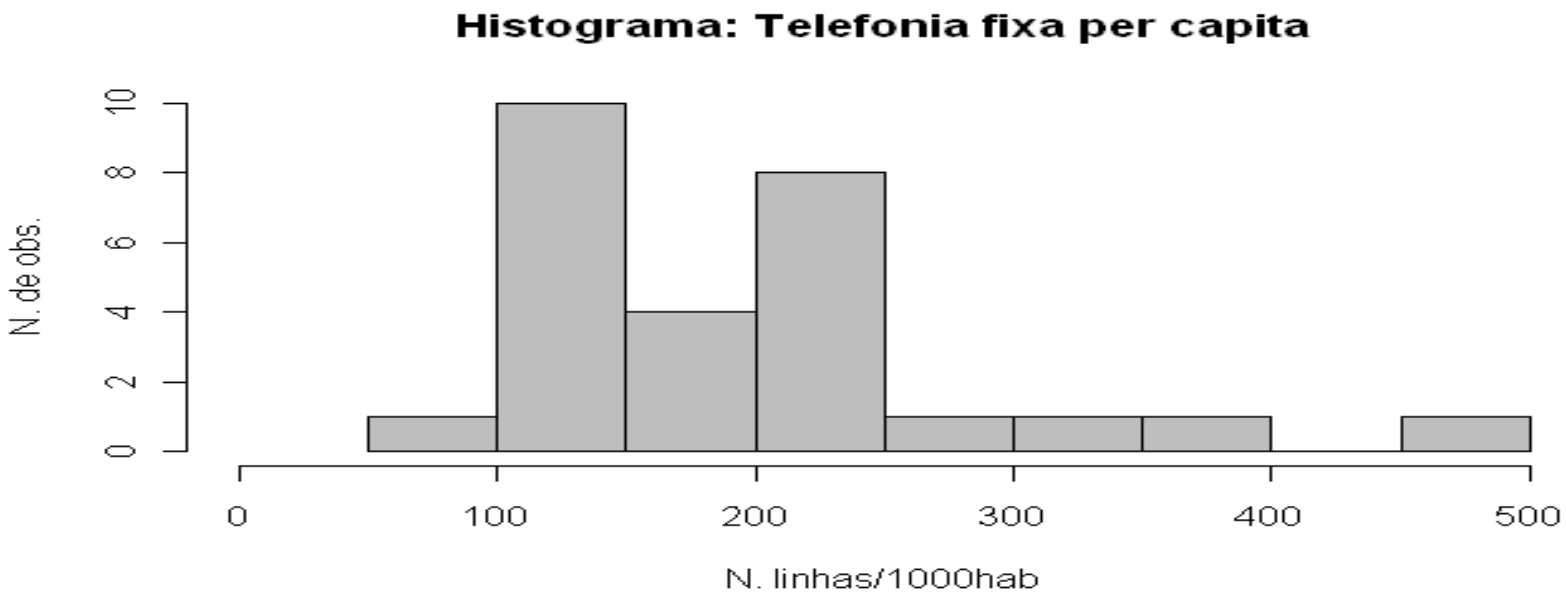
right: variável lógica, se T as classes são fechadas à direira; se F à esquerda

stem(x, …)

Cap1-AED: hist

Exemploda pag 15 – Figura 1.8:

```
> nt=c(183.8,125.4,193.3,162,142.3,140.6,456.8,228.7,231.4,
       86.1,199.6,235.3,218.6,128,125.4,244.2,147.8,118.2,
       347.5,150.1,236.9,214.6,214.1,257.3,362.8,140.7,113.8) #digitação de nt
> hist(nt,   breaks=c(50,100,150,200,250,300,350,400,450,500),  right=T,
         main="Histograma: Telefonia fixa per capita",
         xlab="N. linhas/1000hab",         ylab="N. de obs.",
         xlim=c(0,500), ylim=c(0,10), col="grey")      #histograma da variavel >
```



Cap1: AED – ex: stem, table, cut

***Exemplo**da pag 15 – Figura 1.8:*

```
> nt=c(183.8,125.4,193.3,162,142.3,140.6,456.8,228.7,231.4,
      86.1,199.6,235.3,218.6,128,125.4,244.2,147.8,118.2,
      347.5,150.1,236.9,214.6,214.1,257.3,362.8,140.7,113.8) #digitação de nt

> stem(trunc(nt/10))       ### o ramo a centena e a folha as dezenas

      0 | 8
      1 | 112224444
      1 | 5689
      2 | 011123334
      2 | 5
      3 | 4
      3 | 6
      4 |
      4 | 5


> table(cut(nt, breaks=c(50,100,150,200,250,300,350,400,450,500), right=F))
[50,100) [100,150) [150,200) [200,250) [250,300) [300,350) [350,400) [400,450) [450,500)
       1         9         5         8         1         1         1         0         1
```

Cap1-AED: Medidas (estatísticas)

Para uma dada variável quantitativa, uma medida de centralidade é um "valor típico" em torno do qual se situam os valores daquela variável.As medidas de centralidade mais conhecidas são: a média aritmética e a mediana. Usar os comando: *mean(x); median(y). Por exemplo:*

> mean(nt)
[1] 200.1852
> median(nt)
[1] 193

Uma medida de localização é o quantil. A função apropriada do R para obter os quantis de um vetor numérico x é a função > quantile(x).Se desejarmos determinar os três quartis, usaríamos o comando: ***quantile(x,c(0.25,0.5,0.75))***

Se desejarmos o quinto, o décimo e o nonagésimo percentis, usaríamos o comando:

> ***quantile(x,c(.05,0.10,0.90))***

O comando ***quantile(x,p)*** *retorna o quantil de ordem p das observações de x, podendo p ser um vetor. Por exemplo:*

> quantile(nt, c(.20, .50, .95))
20% 50% 95%
130.6 193.0 358.2

Uma medida de dispersão para uma variável quantitativa é um indicador do grau de espalhamento dos valores da amostra em torno da medida de centralidade. As medidas de dispersão mais conhecidas são: a variância, o desvio-padrão e a distância interquartil=diferença entre o terceiro e o primeiro quartis.

> var(nt)
[1] 7131.464
> sd(nt)
[1] 84.448

Cap1-AED: IEQ, fivenum, boxplot

> q=fivenum(nt); q[4]-q[2] # em q estão os 5 núm. Subtraímos o Q3 do Q1
[1] 92

Os cinco valores, x(1), Q1, Q2, Q3, x(n), mínimo, os três quartis e o máximo, são importantes para se ter uma boa idéia da assimetria dos dados. Esse valores podem ser obtidos pelo comando **fivenum(x). O *summary(x)*** acrescenta também a média ao resultado.

Por exemplo:
> fivenum(nt)
[1] 86 141 193 233 457
> summary(nt)
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   86.0   141.0   193.0   200.2   233.0   457.0
```

O *Box Plot* ou Desenho Esquemático é um gráfico que se costuma utilizar para sintetizar em uma mesma figura várias informações relativas à distribuição de uma determinada variável quantitativa. Nele também são representadas as observações discrepantes.
Observações discrepantes ou outliers são observações cujos valores estão muito afastados dos demais (para mais ou para menos). Essas observações podem afetar de forma substancial o resultado das análises estatísticas. O comando para usar-lo é: **boxplot(x).**

Por exemplo: ver fig 1.25, pag. 28
```
> nt=c(183.8,125.4,193.3,162,142.3,140.6,456.8,228.7,231.4,
       86.1,199.6,235.3,218.6,128,125.4,244.2,147.8,118.2,
       347.5,150.1,236.9,214.6,214.1,257.3,362.8,140.7,113.8) #digitação de nt
> boxplot(nt, ylim=c(50,500),xlab="N. linhas/1000hab")
```

Cap1-AED: Relação entre duas variáveis Qualitativas

Quando se deseja investigar a relação entre duas variáveis qualitativas, o caminho natural é montar uma tabela de contingência. Construir uma tabela de contingência consiste em colocar nas linhas os valores possíveis de uma variável e nas colunas os valores possíveis cruzamento.O comando para fazer a tabela seria: >table(x,y),

Por exemplo:

```
> tab1.2=read.table("f:\\SBPO2010R\\tab1_2.txt", header = T, sep = ",")
> attach(tab1.2)

        > table(CATEG, Classe_IMC)
              Classe_IMC
        CATEG   normal sobrepeso
            A     18         4
            S      9        14
```

Para analisar a relação entre 2 variáveis através de uma tabela de contingência, um procedimento muito útil é calcular os percentuais em relação aos totais das linhas e também os percentuais em relação aos totais das colunas. Os comandos seriam prop.tab(x,1), para linha e prop.tab(x,2) para coluna. Por exemplo usando a tabela 2.5, página 46:

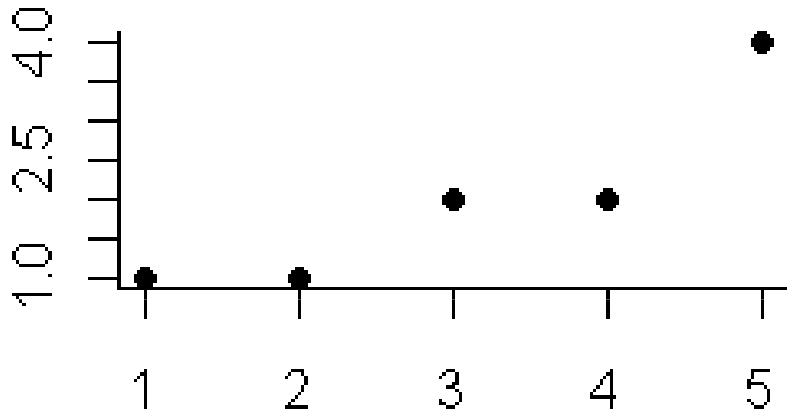
```
> mat= matrix(c(68,15,45,10,  66,21,42,15,   66,24,25,19,   39,16,17,11),4,4,byrow=T)
> rownames(mat)=c("18 a 21 anos","22 a 25 anos","26 a 30 anos","31 a 40 anos")
> colnames(mat)=c("Cin","Teat","S/M","D/Ex")
> mat1=prop.table(mat, 1)          #% por linha    ###tab 2.7
> mat2=prop.table(mat, 2)          #% por coluna ###tab 2.8
```

Cap1-AED: Relação entre duas variáveis Quantitativas

Quando se deseja investigar a relação entre duas variáveis quantitativas o mais adequado é começar pela construção de um Diagrama de Dispersão. Construir um diagrama de dispersão para 2 variáveis quantitativas X e Y consiste em localizar pares de valores observados (xi , yi) como pontos em um sistema de eixos coordenados. O camando seria plot(x,y).

Por exemplo:

>x=c(1,2,3,4,5); y=c(1,1,2,2,4); plot(x,y)



Um indicador do grau de interdependência linear para 2 variáveis quantitativas X e Y é o coeficiente de correlação rxy, que pode assumir qualquer valor real entre -1 e 1. O coeficiente de correlação entre X e Y é calculado por uma das duas expressões matemáticas (equivalentes) a seguir:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\{\sum_{i=1}^{n}(x_i - \bar{x})^2 \cdot \sum (y_i - \bar{y})^2\}^{1/2}} = \frac{\sum_{i=1}^{n} x_i y_i - n \cdot \bar{x} \cdot \bar{y}}{\{(\sum_{i=1}^{n} x_i^2 - n \cdot \bar{x}^2)(\sum_{i=1}^{n} y_i^2 - n \cdot \bar{y}^2)\}^{1/2}}$$

O comando seria: cor(x,y). Por exemplo:

>x=c(1,2,3,4,5); y=c(1,1,2,2,4); cor(x,y)

[1] 0.9036961

Cap1-AED: Relação entre duas variáveis Quantitativas

- **Quando se verifica através do coeficiente de correlação (ou pelo aspecto visual do Diagrama de Dispersão) que existe uma forte relação linear entre 2 variáveis X e Y, pode ser de interesse calcular a equação da reta que representa esta relação entre as 2 variáveis: y = a + b.x. A equação y = a + b.x considera que y é a variável dependente (ou variável resposta) e que x é a variável independente (ou variável preditora) a ser usada para explicar o comportamento da variável y. A equação da reta pode ser usada para se antever qual seria o valor y_0 da variável resposta y correspondente a um determinado valor x_0 da variável preditora x.**
- **As fórmulas que nos permitem calcular os valores de *a* e *b* a partir dos dados são:**

$$b = \frac{\sum_{i=1}^{n} x_i y_i - \frac{\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n}}{\sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}} \qquad \text{e} \qquad a = \bar{y} - b \cdot \bar{x}$$

O coeficiente *b* mede a inclinação da reta de Regressão. Então, ao passarmos de um ponto a outro sobre a reta, *b* mede a relação entre as variações de y e de x. O coeficiente *a* mede o valor de y quando x é igual a zero, ou seja, é o intercepto da reta de Regressão.

O comando para calcular os coeficientes a e b, seria: > ls.print(lsfit(x,y))

```
>x=c(1,2,3,4,5);  y=c(1,1,2,2,4);
>reg=lsfit(x,y)
>ls.print(reg)
```

Cap. 2-a: Simulação do conceito freqüentista

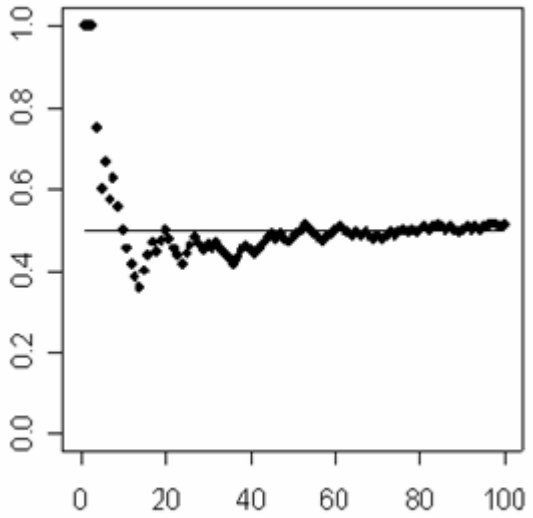
Conceito Freqüentista de Probabilidade: Suponha que o experimento foi repetido n vezes, sempre sob as mesmas condições, e que o evento A **ocorreu m** vezes entre essas **n realizações** do experimento. Então a fração m/n é uma boa aproximação para a probabilidade de A, se o número n de repetições for bastante grande.

Simbolicamente, **P (A) ≅ m/n.**

1	2	3	3	3	4	4	5	5	5
5	5	5	5	6	7	8	8	9	10
10	10	10	10	11	12	13	13	13	14
14	15	15	15	15	15	16	17	18	18
18	19	20	21	22	22	23	23	23	24
25	26	27	27	27	27	27	28	29	30
31	31	31	31	32	32	33	33	33	34
34	35	36	36	37	38	38	39	39	40
41	41	42	43	43	43	44	44	44	45
46	46	47	47	48	49	50	50	50	51

Exemplo: Simulando 100 lançamentos de uma moeda

No R, foram simulados 100 lançamentos de uma moeda equilibrada, isto é, onde as chances de cara e de coroa são iguais. Depois de cada lançamento, foi observado o número acumulado de caras obtidas até esse momento e foi calculada a proporção de caras correspondente. Na tabela a seguir estão apresentados os valores correspondentes ao número acumulado de caras ao longo do processo. Por exemplo, para a jogada de número 29 o número acumulado de caras é 13 e a fração de caras é 13/29. O gráfico abaixo mostra a evolução dessa fração à medida que foram feitos os 100 lançamentos da moeda.



Os comandos no R para a elaboração do gráfico:

```
>x=1:100; y=cumsum(sample(0:1,100,rep=T))
>plot(x,y/1:100, ylim=c(0,1), xlim=c(0,100), pch=16)
>segments(1,0.5,100,0.5)
```

Cap. 2-b: Variáveis Aleatórias (v.a.)

Uma variável aleatória (v.a.) é uma função que associa cada elemento de um espaço amostral a um número real. As variáveis aleatórias podem ser do tipo:

Discreto: se os seus valores pertencem a um conjunto enumerável de números reais (usualmente valores inteiros).

Contínuo: se os seus valores pertencem a um intervalo de números reais.

O modelo probabilístico de uma variável aleatória X estabelece o padrão de comportamento de sua distribuição de probabilidade.

A função de probabilidade p de uma v. a. discreta X é definida por

$\mathbf{p(x)=P[X=x]}$

A função de distribuição acumulada F de uma v. a. X é definida por

$\mathbf{F(x) = P[X \leq x]}$.

Se X é uma v.a discreta que assume os valores $x_1, x_2, x_3, \ldots, x_N$, então :

- *A média ou esperança de X é*
  $E(X) = x_1 . P(X=x_1) + x_2 . P(X=x_2) + x_3 . P(X=x_3) + \ldots + x_N . P(X=x_N)$
- *A Variância de X é calculada por |*
  $Var(X)=(x_1-E(X))^2.P(X=x_1)+(x_2-E(X))^2.P(X=x_2)+\ldots+(x_N-E(X))^2. P(X=x_N)$
- *O desvio padrão de X é igual à raiz quadrada não negativa da sua variância, DP(X)=* $\sqrt{\mathrm{Var(X)}}$

Cap. 2-b - v.a e o R

O trabalho no R com uma v.a. X está baseado em 4 procedimentos:

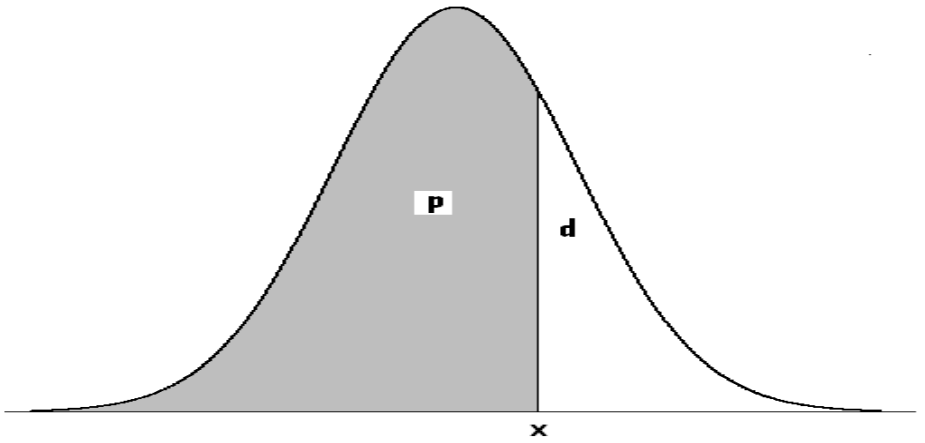
p	probability	– Gera a probabilidade de um valor de **x**;
q	quantile	– Gera o valor x de uma dada probabilidade acumulada, **p**;
d	density	– Gera o valor da função densidade num valor x da variável. Observar que quando a variável é discreta este valor é a probabilidade de x, quando a variável é contínua o resultado é a altura da função densidade de probabilidade;
r	random	– Gera **n** valores do modelo probabilístico em questão.

As distribuições que estudaremos estão listadas a seguir, depois de cada uma delas, entre parênteses está o nome no R.

Entre as discretas: Binomial (**binom**), Hipergeométrica (**hyper**), Poisson (**pos**), Geométrica (**geom**), Binomial negativa- Pascal (**nbinom**);

Entre as contínuas: Uniforme (**unif**), Exponencial (**exp**), Normal (**norm**), t-student (**t**), quiquadrado (**chisq**), F (**f**).

A interligação dos três primeiros procedimentos, **p**,**q** e **d** será ilustrada pela distribuição Normal através do gráfico abaixo:



Cap. 2-b - v.a: pnorm e qnorm

Seja a relação **p = P(X<x)** que aparece na tabela da distribuição Normal, Quiquadrada, t-student e F. Para um dado valor de p acha-se um valor de x, a procura **direta**, que para a Normal no R corresponderia a **pnorm(x, μ, σ)**. Neste caso devo informar o x e os dois parâmetros da distribuição Normal μ e σ.

Já a procura **inversa** seria para um dado valor de x achar um valor de p, que para a Normal no R corresponderia a **qnorm(p,μ,σ)**. Neste outro caso devo informar o valor de p desejado e também, os dois parâmetros da distribuição Normal, μ e σ.

Exemplo: Seja X a v.a. que corresponde ao peso (em kg) de pessoas de uma certa população com média μ=70 Kg e desvio padrão σ=8 Kg, assim X~ N(μ=70, σ2=82).
Se desejarmos:

a) P(X<80) usaremos no R o comando **pnorm(80, 70, 8)**, isto é x=80 é o primeiro parâmetro, enquanto 70 e 8 são parâmetro específicos da distribuição Normal.

b) Admita que o peso limite para ser classificado como obeso é o valor que corresponde a 10% dos mais pesados do população. Achar este peso limite.

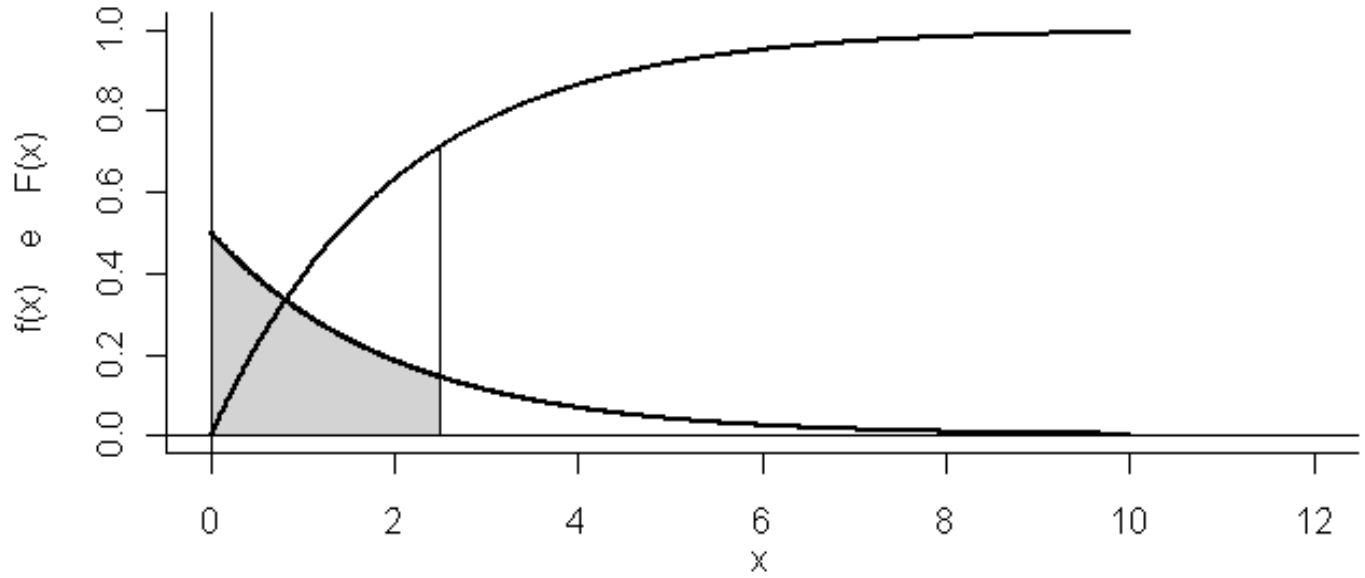
O que se pede é a função inversa. Dado um valor de p=0,90 achar um valor de x que deixa 90% abaixo dele. No R seria **qnorm(0.9, 70, 8)** , isto é, o valor da probabilidade p=0.9 é o primeiro parâmetro, enquanto 70 e 8 são parâmetros específicos da distribuição Normal.

Se usarmos a Normal padrão z=(x-μ)/σ, no caso do item a, o comando seria **pnorm((80-70)/8)** ou pnorm(1.25). Repare que neste caso não foi necessário passar os parâmetros específicos da Normal pois μ=0 e σ =1 coresponde ao *default.* Observação: Sempre que usarmos um p o primeiro parâmetro é um x e os outros são específicos da distribuição em questão.
sempre que usarmos um q o primeiro parâmetro é um p e os outros são específicos.
Sempre que usarmos um d o primeiro parâmetro é um x.
Quando usarmos uma distribuição discreta o d corresponde á probabilidade no ponto x.

Cap. 2-b - v.a: dexp, pexp, points, segments

Pag 118 – Figura 4.12

```
x=seq(0,10,0.01)
plot(x,dexp(x, 1/2), type="l", xlim=c(0,12), ylim=c(0,1), bty="l", ylab="f(x)  e  F(x)")
for(i in seq(0, 2.5, 0.01)) segments(i, 0, i, dexp(i,1/2), col="lightgrey")
abline(v=0, h=0)
points(x,dexp(x, 1/2), type="l", lwd=2, bty="l")
points(x,pexp(x, 1/2), lwd=2, type="l")
segments(2.5,0, 2.5,pexp(2.5,1/2))
```



Cap. 2-b - v.a. disponíveis no R

rbinom(n, size, prob) binomial
rpois(n, lambda) Poisson
rgeom(n, prob) geometrica
rhyper(nn, m, n, k) hipergeometrica
rnbinom(n, size, prob) binomial negativa
runif(n, min=0, max=1) uniforme
rexp(n, rate=1) exponential
rnorm(n, mean=0, sd=1) Gaussiana (normal)
rt(n, df) 'Student' (*t*)
rf(n, df1, df2) Fisher–Snedecor (*F*)
rchisq(n, df) Quiquadrada
rgamma(n, shape, scale=1) gamma
rbeta(n, shape1, shape2) beta
rlnorm(n, meanlog=0, sdlog=1) lognormal
rcauchy(n, location=0, scale=1) Cauchy
rweibull(n, shape, scale=1) Weibull
rwilcox(nn, m, n), Wilcoxon's rank sum statistics
rsignrank(nn, n) Wilcoxon's signed rank statistics
rlogis(n, location=0, scale=1) logistic

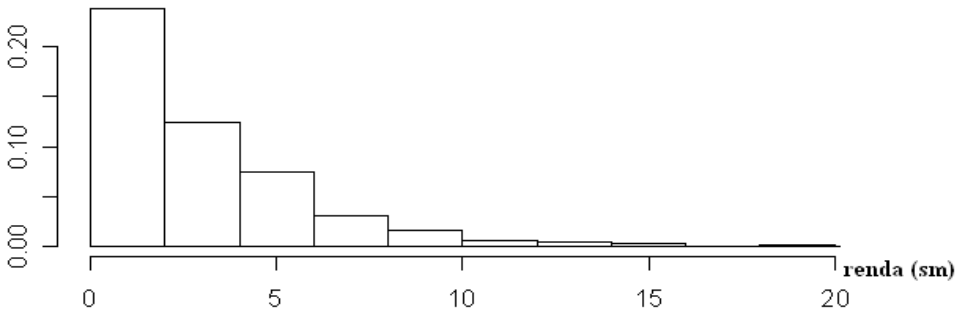
Todas essas distribuições , apresentadas por rnome (nome da variável aleatória), como vimos, trocando a primeira letra e mantendo os parâmetros específicos de cada distribuição (denotados por ...), podem ser usadas substituindo a letra "**r**" por:

d valor da densidade de probabilidade no ponto x	dnome (x, ...)
p probabilidade acumulada no ponto x	pnome (x, ...)
q quantil correspondente a probabilidade acumulada p	dnome (p,...)

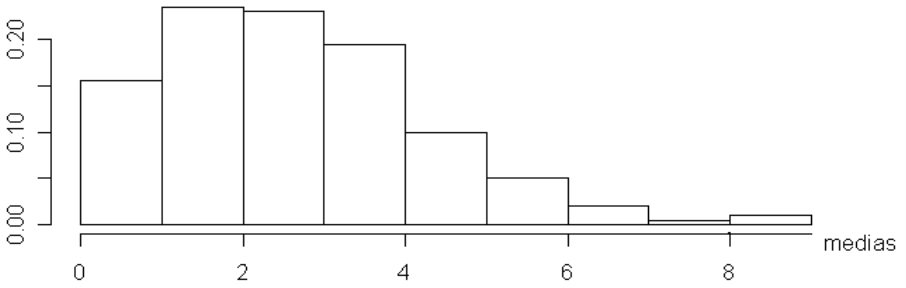
Cap. 2-c: O Teorema Central do Limite (TCL)

O Teorema Central do Limite (abreviadamente, TCL) diz respeito ao comportamento da média amostral à medida que o tamanho n da amostra cresce indefinidamente.

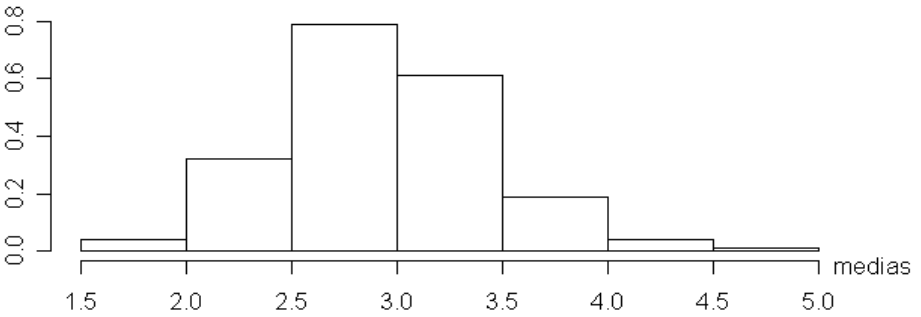
Exemplo 3.1 – A distribuição de renda e o TCL
É um fato conhecido que a distribuição da renda pessoal dos habitantes de um país é usualmente muito desigual, ou seja, muitos ganham pouco e poucos ganham muito. Se forem sorteados 200 habitantes desse país e, com base nas suas rendas mensais construirmos um histograma, ele terá o aspecto.



Agora, se forem sorteadas 200 amostras, cada uma delas contendo 2 habitantes desse país, e se forem calculadas as 200 respectivas médias amostrais, a partir delas obteremos o histograma a seguir:



Agora, cada uma das 200 amostras sorteadas contendo 30 habitantes desse país, e se forem calculadas as 200 médias amostrais, o histograma seria :



Cap. 2-c – TCL: Exemplo

Como pode ser observado, no caso de n = 2 o histograma se aproxima mais de uma curva Normal do que no caso de n = 1. E no caso de n = 30, a semelhança do histograma com uma curva Normal é ainda maior.

O Teorema Central do Limite afirma que, independentemente de qual seja a distribuição original dos Xi's, a distribuição de probabilidade de $\overline{X}_n$ e a distribuição Normal com média μ e variância σ^2/n se aproximam cada vez mais uma da outra, à medida que n cresce.

Portanto, mesmo que a distribuição de probabilidade dos Xi's seja desconhecida, o Teorema Central do Limite garante a possibilidade de usarmos o modelo Normal para calcular, ainda que de forma aproximada, probabilidades relativas à média amostral , desde que n seja suficientemente grande.

Exemplo 6.2: Simulando o efeito do TCL

Para ilustrar o funcionamento do Teorema Central do Limite, vamos exibir agora um exemplo em que a distribuição original a partir da qual os dados são gerados é uma exponencial, modelo este que dá origem a uma função densidade bastante assimétrica (ao contrário do que ocorre ; com a curva Normal). A densidade de uma exponencial com parâmetro λ é dada pela expressão:

$$f(x) = \lambda e^{-\lambda x}, \quad x \geq 0$$

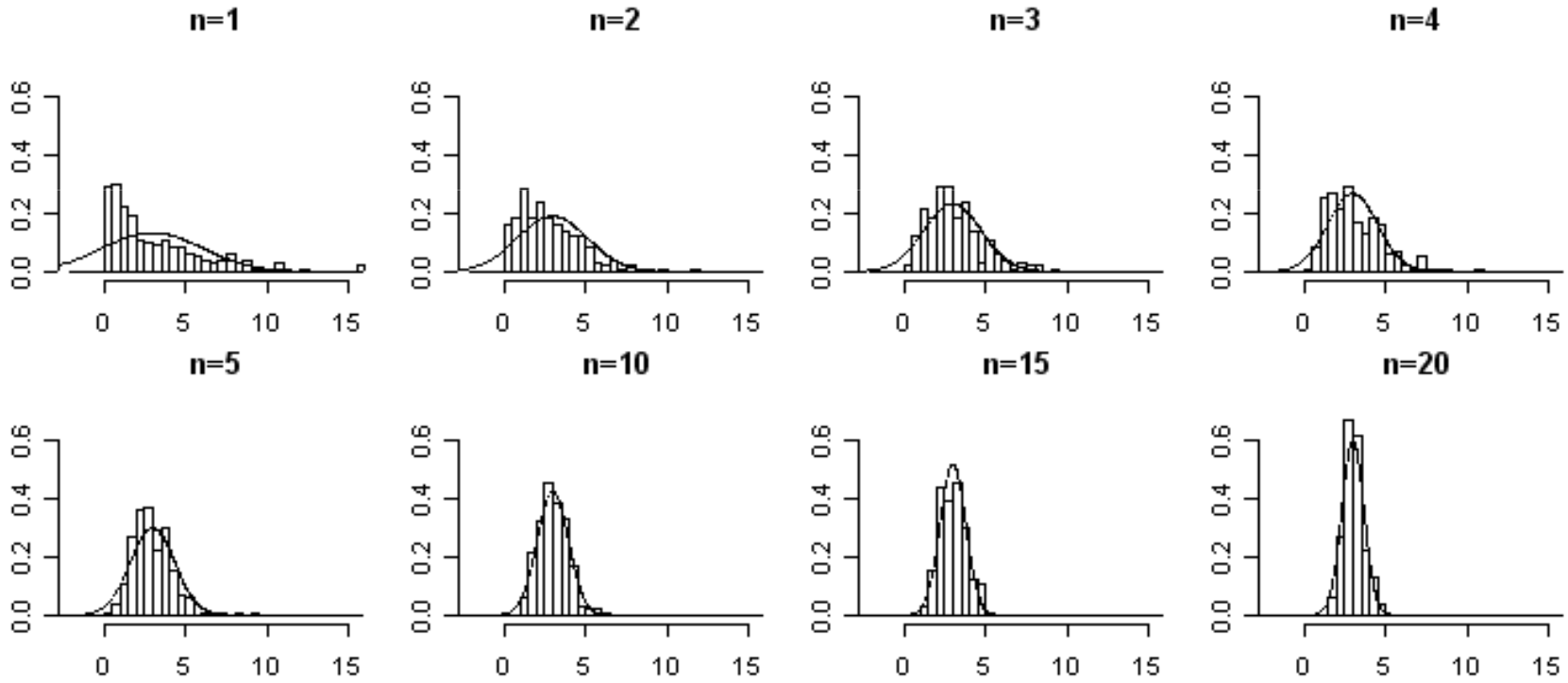
No R, **rexp(n,λ)** simula n valores

Cap. 2-c – TCL: Exemplo

A densidade de uma exponencial com parâmetro λ é dada pela expressão: $f(x) = \lambda e^{-\lambda x}, \quad x \geq 0$

Gerando dados por simulação a partir de uma exponencial com λ = 1/3, para cada um dos seguintes tamanhos n de amostra: 1, 2, 3, 4, 5, 10, 15 e 20,

1. Obtivemos 200 valores da média amostral ;
2. Utilizamos esses 200 valores para construir um histograma;
3. Traçamos no mesmo gráfico uma curva da densidade Normal com $E(\overline{X}_n)=3$ e $DP(\overline{X}_n)=3/\sqrt{n}$



Os 8 histogramas nos mostram que, à medida que o tamanho n da amostra cresce, a forma do histograma se aproxima cada vez mais de uma curva Normal.

Cap. 2-c – TCL: Códigos no R para elaboração da figura com simulações - Exponencial

```
tcl.exp=function(n, N=200, titulo=" ", yl=c(0, .4)) {      ## início da função – tcl.exp
        medias=numeric(N)
        for (i in 1:N)  medias[i]= mean(rexp(n,1/3))
                        hist(medias, xlim=c(-1,10), ylim=yl, freq=F, main=titulo)
        x=seq(-1,10, .02)
        points(x, dnorm(x, 3, 3*sqrt(1/n) ), type="l", lwd=3)
}                                                          ## fim da função

        graphics.off()
par(mfrow=c(3,3), mai=c(.3,.4,.1,.1))
tcl.exp(1,titulo="n=1")
tcl.exp(2,titulo="n=2")
tcl.exp(3,titulo="n=3")
tcl.exp(4,titulo="n=4")
tcl.exp(5,titulo="n=5")
tcl.exp(6,titulo="n=6")
tcl.exp(10,titulo="n=10",yl=c(0,.6))
tcl.exp(15,titulo="n=15",yl=c(0,.6))
tcl.exp(20,titulo="n=20",yl=c(0,.6))
```

Cap. 2-c – TCL: Códigos no R para elaboração da figura com simulações - Exponencial

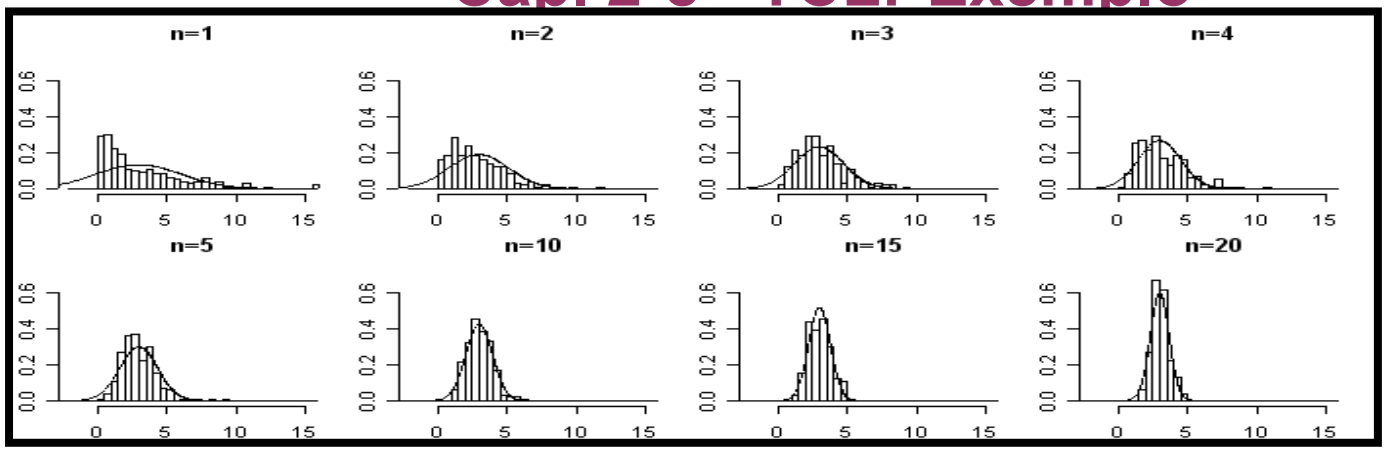
Uma pergunta natural neste ponto seria:

"Quão grande deve ser n para que possamos usar a aproximação fornecida pelo TCL com um nível de precisão aceitável?"

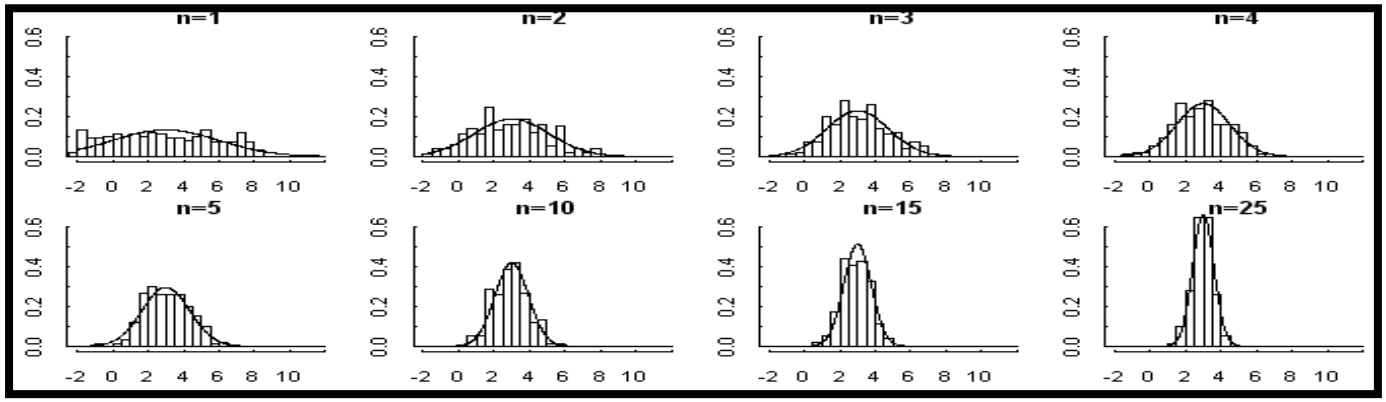
A rapidez com que essa convergência se dá depende de quão distante está a forma da distribuição original das Xi's de uma curva Normal. Em outras palavras, se a distribuição das Xi's já não for muito diferente de uma Normal, com um n não muito grande consegue-se uma boa aproximação. Caso contrário, somente para n bem grande (usualmente, n ≥ 30) a aproximação da distribuição de por uma Normal funcionaria adequadamente.

No exemplo a seguir vamos apresentar esse fenômeno, a saber, a convergência da distribuição de $\overline{X}_n$ para uma Normal à medida que n cresce, gerando por simulação os dados originais a partir de diferentes modelos probabilísticos. Em todos os casos, a distribuição original é bem diferente da Normal, E(X)=3 e DP(X)=3. No que se refere à Simulação, foi seguida a mesma seqüência de passos do exemplo anterior.

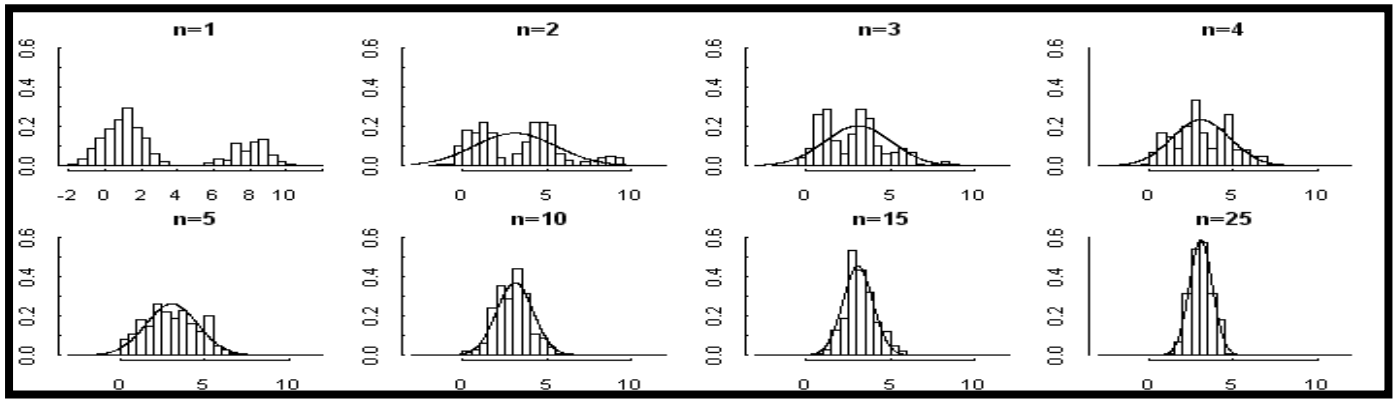
Cap. 2-c – TCL: Exemplo



Exponencial



Uniforme



Mistura de Normais

Cap. 2-c – TCL: Exemplo

Como se pode observar:

1. No caso da distribuição uniforme (A), o histograma de já se aproxima bastante de uma Normal quando n é da ordem de 4.

2. Já no caso da distribuição Exponencial (B) e da mistura de normais (C), modelos esses que se afastam muito mais de um "comportamento gaussiano", a aproximação pela Normal só se mostra mais adequada a partir de n em torno de 10.

3. No caso do modelo em (C), à medida que n cresce, tudo se passa como se houvesse a "erupção de um vulcão dentro do vale".

Cap. 2-c – TCL: Códigos no R para elaboração da figura com as simulações - Uniforme

```
tcl.unif=function(n,N=100,titulo=" ", yl=c(0, .4)) {
medias=numeric(N)
  for (i in 1:N) medias[i]= mean(runif(n, 3-3*sqrt(3), 3+3*sqrt(3)))
  hist(medias, xlim=c(-6,10), ylim=yl, freq=F, main=titulo)
  x=seq(-6,10, .02)
  points(x, dnorm(x, 3, 3*sqrt(1/n) ), type="l", lwd=3)
####medias
}
graphics.off()
par(mfrow=c(3,3), mai=c(.3,.4,.1,.1))
tcl.unif(1,titulo="n=1",yl=c(0,.6))
tcl.unif(2,titulo="n=2",yl=c(0,.6))
tcl.unif(3,titulo="n=3",yl=c(0,.6))
tcl.unif(4,titulo="n=4",yl=c(0,.6))
tcl.unif(5,titulo="n=5",yl=c(0,.6))
tcl.unif(6,titulo="n=6",yl=c(0,.6))
tcl.unif(10,titulo="n=10",yl=c(0,.6))
tcl.unif(15,titulo="n=15",yl=c(0,.6))
tcl.unif(20,titulo="n=20",yl=c(0,.6))
```

Cap. 2-c – TCL: Códigos no R para elaboração da figura com as simulações - Mistura de Normais

```
X2=c(rnorm(350,1,1), rnorm(150,8,1))  ##X2 é a População de onde retiramos as
                                                    ## amostras
br=seq(-2, 12, .5)
tcl.2modas=function(n,N=200,titulo=" ", yl=c(0, .6)) {
        medias=numeric(N)
        for (i in 1:N)  medias[i]= mean(sample(X2,n,rep=T))
        hist(medias,breaks=br, xlim=c(-2,12), tcl=-0,1, ylim=yl, ##xarp=c(-3,12,16),
                        tck=0.05, lab=c(5,5,15), freq=F, main=titulo)
        x=seq(-3,12, .02)
        points(x, dnorm(x, med, dp/sqrt(n) ), type="l", lwd=2)
}
par(mfrow=c(2,4), mai=c(.3,.0,.1,.1), mar=c(2, 2, 2, 1))
hist(X2, freq=F, breaks=seq(-3,12,.5), bty="o", xlim=c(-2,12),##xarp=c(-3,12,16),
        tck=0.05, lab=c(5,5,15), ylim=c(0,.6), main="POPULAÇÃO",  lwd=2)
tcl.2modas(2,titulo="n=2")
tcl.2modas(3,titulo="n=3")
tcl.2modas(4,titulo="n=4")
tcl.2modas(5,titulo="n=5")
tcl.2modas(10,titulo="n=10")
tcl.2modas(15,titulo="n=15")
tcl.2modas(25,titulo="n=25")
```

CAP 3-a) Intervalo de Confiança

Seja um estimador pontual do parâmetro θ, $\hat{\theta}$ é uma variável aleatória, que varia de amostra para amostra. Por isso, há uma certa dose de incerteza inerente a esse processo de estimação. Nosso objetivo agora é obter, com base nos dados amostrais (da única amostra observada), um intervalo ao qual o valor correto do parâmetro deve ter grande chance de pertencer.

- Detalhando um pouco mais: No processo de estimação por intervalo de um parâmetro θ, devemos determinar um intervalo que contenha o verdadeiro valor do parâmetro com probabilidade 1-α, onde α é um pequeno valor pré-fixado. Este intervalo é construído em geral em torno do estimador pontual $\hat{\theta}$, considerando uma margem de erro d, de forma a que, uma vez fixada a probabilidade 1-α, calculemos d tal que $P(\hat{\theta} - d \leq \theta \leq \hat{\theta} + d) = 1-\alpha$. Chama-se intervalo de confiança para θ, ao nível $1 - \alpha$ ao intervalo $[\hat{\theta} - d ; \hat{\theta} + d]$.
- Ou seja, o estimador pontual é o centro do Intervalo de Confiança e o erro absoluto d associado a $\hat{\theta}$ define a amplitude desse intervalo. O que é, de fato, variável aleatória são os extremos do intervalo. O parâmetro tem valor desconhecido, não aleatório (fixo, a ser estimado).

Exemplo: **Intervalo de Confiança para a média populacional, com o desvio padrão conhecido.**

O parâmetro a estimar é a média populacional μ. A estimação será baseada em $X_1, X_2, ..., X_n$ uma amostra aleatória com $E(X_i) = \mu$ e $var(X_i) = \sigma^2$, para todo i = 1,2,..., n. Queremos que seja válida a expressão: $P\left[\left|\overline{X} - \mu\right| \leq d\right] = 1 - \alpha$, o que equivale a : $P[-d \leq \overline{x} - \mu \leq d] = 1-\alpha$.

CAP 3-a) Intervalo de Confiança

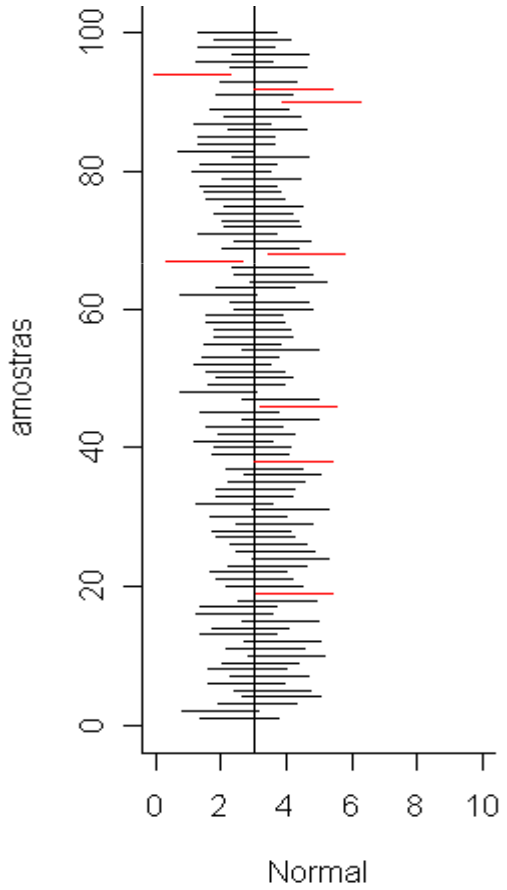
- Lembrando que para n suficientemente grande, pelo Teorema Central do Limite, a média amostral segue uma distribuição que se aproxima da Normal(μ; σ^2/n) (ou é exatamente a Normal(μ; σ^2/n), no caso em que a distribuição comum das v.a. X_i's já é Normal), então :

$P[-d \le \bar{X} - \mu \le d] = 1-\alpha \Rightarrow 1-\alpha = P\left[|Z| \le \frac{d}{\sigma/\sqrt{n}}\right]$ Logo, $d = z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}$.

- Faremos uma simulação para a construção de Intervalos de Confiança com 100 amostras de dois tipos de população: Normal e Exponencial. A lista de comandos do R que usamos é:

```
IC.N = function (N, n, mu, sigma = 3, conf) {
   plot(0, 0, type="n", xlim=c(0,10), ylim=c(0,N), bty="l",
        xlab="Normal", ylab="amostras")
   abline(v=mu)
   z0 = qnorm(1-((1-conf)/2))
   sigma.xbarra = sigma/sqrt(n)
   for (i in 1:N) {
      x = rnorm(n, mu, sigma)
       media = mean(x)
      li = media - z0 * sigma.xbarra
      ls = media + z0 * sigma.xbarra
      plotx = c(li,ls)
      ploty = c(i,i)

      if (li > mu |  ls < mu) lines(plotx,ploty, col="red")
      else lines(plotx,ploty)
   } }
```

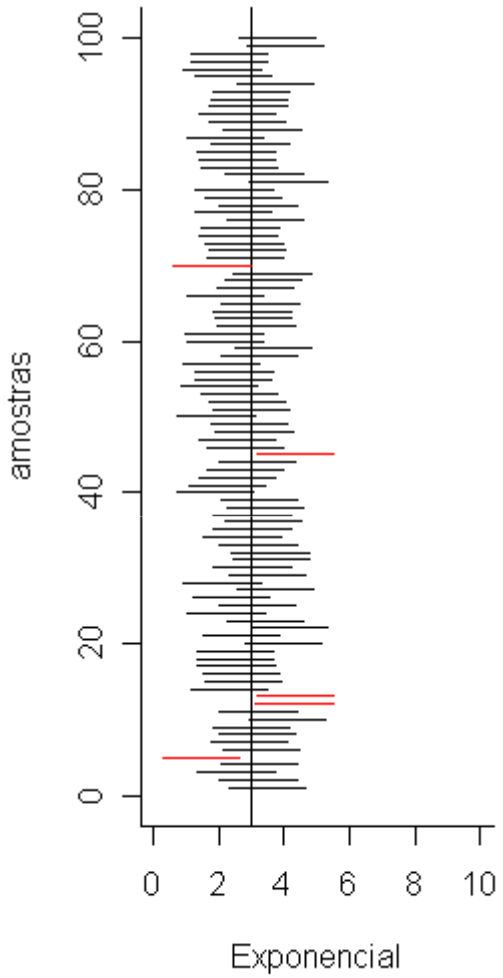


> IC.N(100, 25, 3, 3, .95)

CAP 3-a) Intervalo de Confiança

```
IC.exp = function (N, n, lambda, conf) {
    mu=1/lambda; sigma=1/lambda
    plot(0, 0, type="n", xlim=c(0,10), ylim=c(0,N), bty="l",
        xlab="Exponencial", ylab="amostras")
    abline(v= mu)
    z0 = qnorm(1-((1-conf)/2))
    sigma.xbarra = sigma/sqrt(n)
    for (i in 1:N) {
        x = rexp(n,lambda)
         media = mean(x)
        li = media - z0 * sigma.xbarra
        ls = media + z0 * sigma.xbarra
        plotx = c(li,ls)
        ploty = c(i,i)

        if (li > mu |  ls < mu) lines(plotx, ploty, col="red")
        else lines(plotx, ploty)
    }
}
```




```
> IC.exp(100, 25, 1/3, .95)
```

CAP 3-b: Teste de Hipóteses (TH)

O objetivo de um teste de hipótese é avaliar a validade de uma afirmação sobre determinada característica da população, usando para isso os dados de uma amostra. Essa característica é representada pela v.a. contínua X, cujo comportamento probabilístico é expresso pela função de densidade f, com parâmetro Θ, que tem valor desconhecido.

Em um teste existem duas hipóteses envolvidas : H0, denominada hipótese nula e H1, denominada hipótese alternativa. O procedimento de teste de hipótese consiste em estabelecer um critério de decisão que leve a Aceitar ou Rejeitar H0, com base nos valores amostrais.
A Estatística de teste é uma função da amostra aleatória utilizada para definir o critério de decisão. Estabelecer o critério de decisão consiste em dividir o conjunto dos valores possíveis da estatística de teste em duas partes denominadas Região de Aceitação, **A** e Região de Rejeição, **R** , da hipótese nula.

Em um teste de hipótese há dois tipos possíveis de erro de decisão :

Erro I - Rejeitar H0, quando H0 é verdadeira

Erro II - Aceitar H0, quando H0 é falsa.

As probabilidades de ocorrência dos erros são α = P [Erro I] e β = P [Erro II].

A probabilidade de erro I, α, é o nível de significância do teste, cujo valor é arbitrado pelo pesquisador, deve ser pequena, pois corresponde a probabilidade de um erro (0,05 ou 0,01).

O nível crítico ou p-valor do teste é o menor valor de α para o qual, ainda rejeitaríamos H0 de acordo com os dados observados.

CAP 3-b) – TH: teste t

t.test(x, y = NULL, alternative = c("two.sided", "less", "greater"), mu = 0,

paired = FALSE, var.equal = FALSE, conf.level = 0.95, ...)

Procedimentos de teste de hipóteses, com nível de significância α e amostras de tamanho n:

	Hipóteses	Região de Rejeição de H_0
σ desconhecido	$H_0: \mu = \mu_0$ $H_1: \mu \neq \mu_0$	$\bar{X} < \mu_0 - \frac{S}{\sqrt{n}} t_{n-1,1-\alpha/2}$ ou $\bar{X} > \mu_0 + \frac{S}{\sqrt{n}} t_{n-1,1-\alpha/2}$
	$H_0: \mu \leq \mu_0$ $H_1: \mu > \mu_0$	$\bar{X} > \mu_0 + \frac{S}{\sqrt{n}} t_{n-1,1-\alpha}$
	$H_0: \mu \geq \mu_0$ $H_1: \mu < \mu_0$	$\bar{X} < \mu_0 + \frac{S}{\sqrt{n}} t_{n-1,\alpha}$

Uma Amostra

Categoria do teste	Condições	Hipóteses	Estatística de Teste	Região de Rejeição de H_0
Não pareado	$\sigma_X = \sigma_Y = \sigma$ $X \sim N(\mu_X; \sigma_X^2)$ e $Y \sim N(\mu_Y; \sigma_Y^2)$	$H_0: \mu_X - \mu_Y = 0$ $H_1: \mu_X - \mu_Y \neq 0$	$T = \frac{\bar{X} - \bar{Y}}{\sqrt{S_p^2\left(\frac{1}{m} + \frac{1}{n}\right)}}$, onde $S_p^2 = \frac{(m-1)S_X^2 + (n-1)S_Y^2}{m+n-2}$ e m e n são os tamanhos amostrais	$\lvert T_{obs} \rvert > t_{1-\frac{\alpha}{2}}$
Pareado	$D_i = X_i - Y_i$ $D \sim$ Normal (μ_D, σ_D)	$H_0: \mu_D = 0$ $H_1: \mu_D \neq 0$	$T = \frac{\bar{D}}{S_D/\sqrt{n}}$, n é o tamanho da amostra	$\lvert T_{obs} \rvert > t_{1-\frac{\alpha}{2}}$

Duas Amostras

Obs.: Os testes acima são bilaterais.

CAP 3–b) TH: exemplo de teste t – amostras independentes

```
#pag.233- teste t amostras independente - comparação log(salário)
#         entre os grupos de comércio e de serviço
Log.Sal=c(1.289,1.569,1.250,1.344,1.456,1.636,1.573,1.713,0.906,0.903,
          0.977,1.220,1.103,1.069,1.287,1.410,1.496,1.311,1.337,1.366,
          1.227,1.191,1.459,1.280,1.152,1.740,1.649,1.765,2.410,1.701,
          1.538,1.924,1.925,1.721,1.549,1.891,1.534,1.638,1.207,1.682,
          1.206,1.423,2.010,1.431,1.265,1.570)
Sal=exp(Log.Sal)
setor= c(rep("C",23), rep("S",23))

t.test(Sal[setor=="C"],  Sal[setor=="S"], var.equal=T)
```

```
#       Two Sample t-test
data:  Sal[setor == "C"] and Sal[setor == "S"]
t = -3.6822, df = 44, p-value = 0.0006289
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.3079005 -0.6751838
sample estimates:
mean of x           mean of y
         3.786010  5.277552
```

CAP 3–b) TH: exemplo de teste t – amostra pareada

```
#pag.237- teste t pareado
P1=c(6.3,1.5,5.9,6.4,5.5,5.4,5.4,8.0,5.9,8.0,6.5,2.0,3.6,6.0,9.8,6.8,5.3,
    8.7,6.5,6.4,7.7,8.5,5.3,6.9,8.0,8.2,7.1,8.4,6.0,5.5,7.2,6.4,5.5,6.4)
P2=c(3.6,3.8,3.0,6.0,4.3,4.6,6.4,5.5,6.0,4.3,4.3,5.2,3.4,2.8,8.3,7.1,5.5,
    8.2,3.8,5.5,6.7,6.7,4.4,3.4,5.9,6.0,5.9,6.8,5.0,6.2,5.4,4.7,3.6,5.2)

t.test(P1, P2, alt="greater", paired = T)
```

```
        Paired t-test
data:  P1 and P2
t = 4.4176, df = 33, p-value = 5.072e-05
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 0.716695      Inf
sample estimates:
mean of the differences
          1.161765
```

CAP 3–b) TH: exemplo de teste Quiquadrado, chisq.test

```
#pag.233- teste Quiquadrado -
tcont=matrix(c(68,35,85,25,15,30,258,    74,61,7,61,2,20,225), 7,2)
chisq.test(tcont)
```

```
        Pearson's Chi-squared test
data:  tcont
X-squared = 98.6424, df = 6, p-value < 2.2e-16
```

CAP 3–b) TH: exemplo de teste Quiquadrado, chisq.test

aov(formula, data = NULL, ...)

```
##########################################
#pag.244- ANOVA - Comparação de três rações para suinos
A=c(44,49,43,51,44,75,42,51,34,30,53,42,45,36,30,
   32,21,33,42,10,40,39,52,46,29,42,47,45,39,59)
B=c(34,36,40,54,59,53,44,54,32,68,69,54,41,46,47,
   65,66,45,57,39)
C=c(57,40,40,36,45,66,39,50,25,21,29,27,28,39,42,
   21,30,41,43,29,42,44,58,28,49)
aumento.P=c(A,B,C)
racao=c(rep("A",30),rep("B",20), rep("C",25))
```

summary.aov(aov(aumento.P ~ racao))

```
            Df Sum Sq Mean Sq F value   Pr(>F)
racao        2 1538.5   769.3  5.6136 0.005425 **
Residuals   72 9866.6   137.0
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Warning message:
In model.matrix.default(mt, mf, contrasts) :
  variable 'racao' converted to a factor
```

Apêndices

A-1) Apresentaremos aqui algumas figuras feitas no R na elaboração do livro.

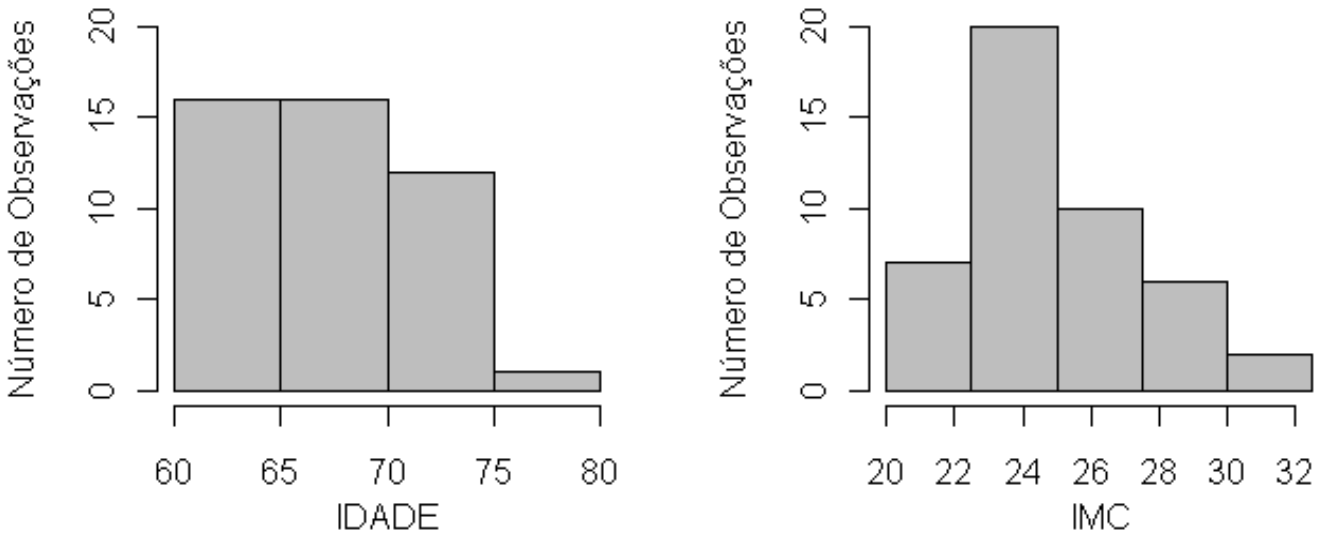
Achamos que o exame desses códigos acompanhado dos resultados pode ser um bom aprendizado, um exercício, ou talvez uma recordação do material aqui exposto.

A-2) Resumo de comandos

a) Criação de dados
b) Informação de uma Variável
c) Seleção de dados e manipulação
d) Estatísticas e operações matemáticas
e) Corte e extração de dados
f) Operação com Matrizes
g) Gráficos (Plotting)
h) Teste de hipóteses
i) Programming
j) Commandos auxiliaries em Gráficos
k) Comando par (Graphical parameters)
l) Input and output

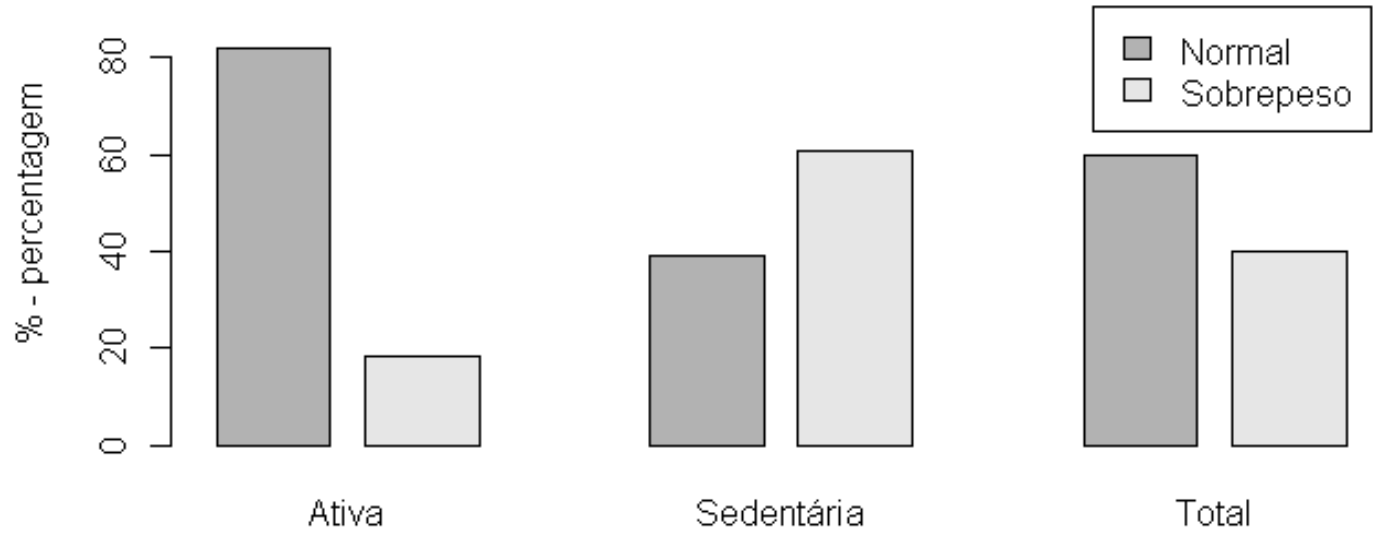
A1) – Pag 14 – Figura 1.5

```
IDADE=c(61,69,61,71,63,71,72,68,66,69,72,67,63,66,63,63,60,67,71,63,60,69,64,63,66,
        71,64,70,63,66,64,69,69,64,63,72,73,68,71,72,69,68,68,73,79)
IMC= c(24.5,27.3,28.1,30.1,25.4,30.1,28.0,23.4,26.8,22.8,25.5,22.8,23.5,23.2,20.3,22.6,23.9,
       24.3,27.1,22.7,23.7,25.8,21.3,24.3,24.3,24.8,21.9,23.4,21.6,21.4,22.1,22.7,22.7,21.1,
       26.8,27.8,27.5,26.7,28.6,25.3,23.9,25.8,24.7,28.4,23.5)
par(mfrow=c(1,2))
hist(IDADE, breaks=c(60,65,70,75,80), ylim=c(0,20), ylab="Número de Observações",
       main= " ", col="grey", right = F)
hist(IMC,   breaks=c(20.0,22.5,25.0,27.5,30.0,32.5),
      ylab="Número de Observações",main= " ", col="grey", right = F )
```



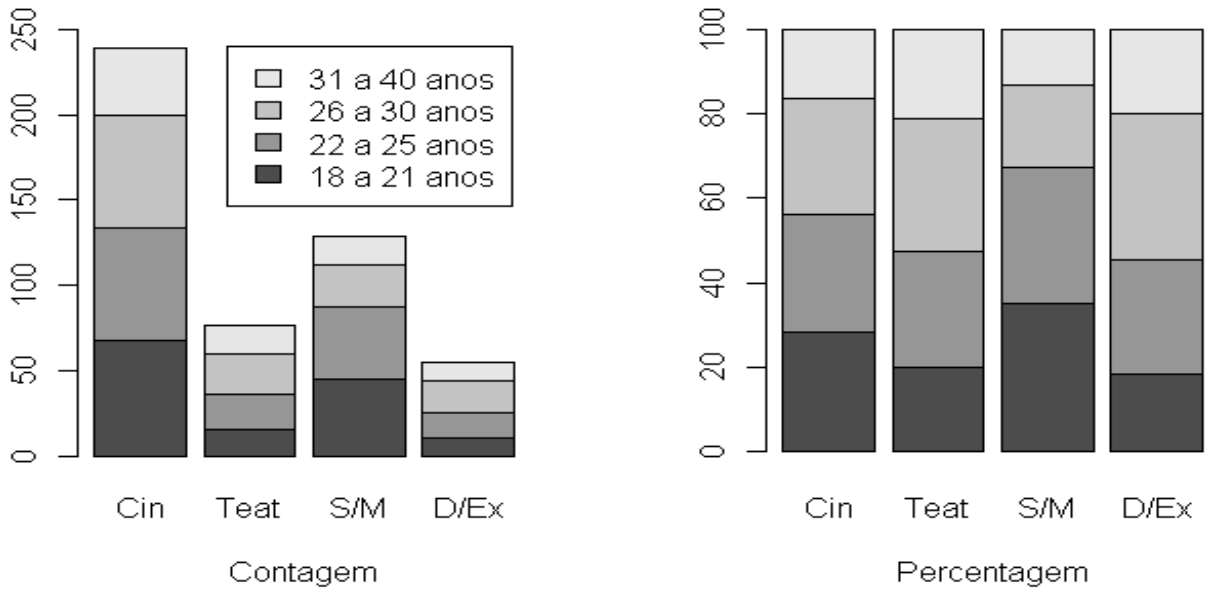
A1) – Pag 44 – Figura 2.1

```
mat=matrix(c(81.82,39.13,60,18.18,60.87,40),3,2)
rownames(mat)=c("Ativa","Sedentária","Total")
colnames(mat)=c("Normal","Sobrepeso")
barplot(t(mat), beside = TRUE, space=c(.3,1.5),
    col=gray(c(.7,.9)),
    legend = c("Normal","Sobrepeso"), ylim = c(0, 95), ylab="% - percentagem")
```



A1) – Pag 48 – Figura 2.2

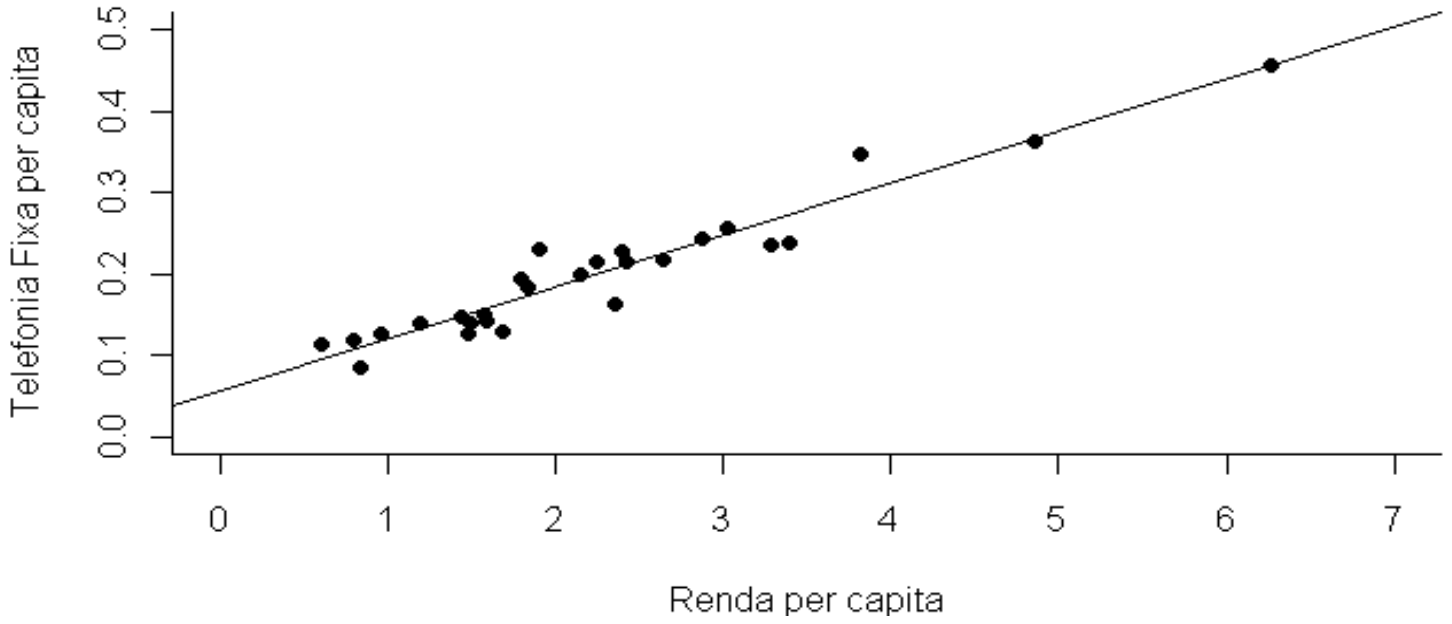
```
mat= matrix(c(68,15,45,10,  66,21,42,15,   66,24,25,19,   39,16,17,11),4,4,byrow=T)
rownames(mat)=c("18 a 21 anos", "22 a 25 anos", "26 a 30 anos", "31 a 40 anos")
colnames(mat)=c("Cin","Teat","S/M","D/Ex")
mat1=mat;  for (i in 1:4) {mat1[,i]<-mat1[,i]*100/sum(mat1[,i]) }
par(mfrow=c(1,2), mai=c(.1,.1,.1,.1), mar=c(5, 4, 2, 2) )
barplot(mat,beside=F, ylim=c(0,250),    legend = c("18 a 21 anos", "22 a 25 anos",
      "26 a 30 anos", "31 a 40 anos"),  xlab="Contagem")###,
barplot(mat1, beside=F, xlab="Percentagem")
```



A1) – Pag 52 – Figura 2.5

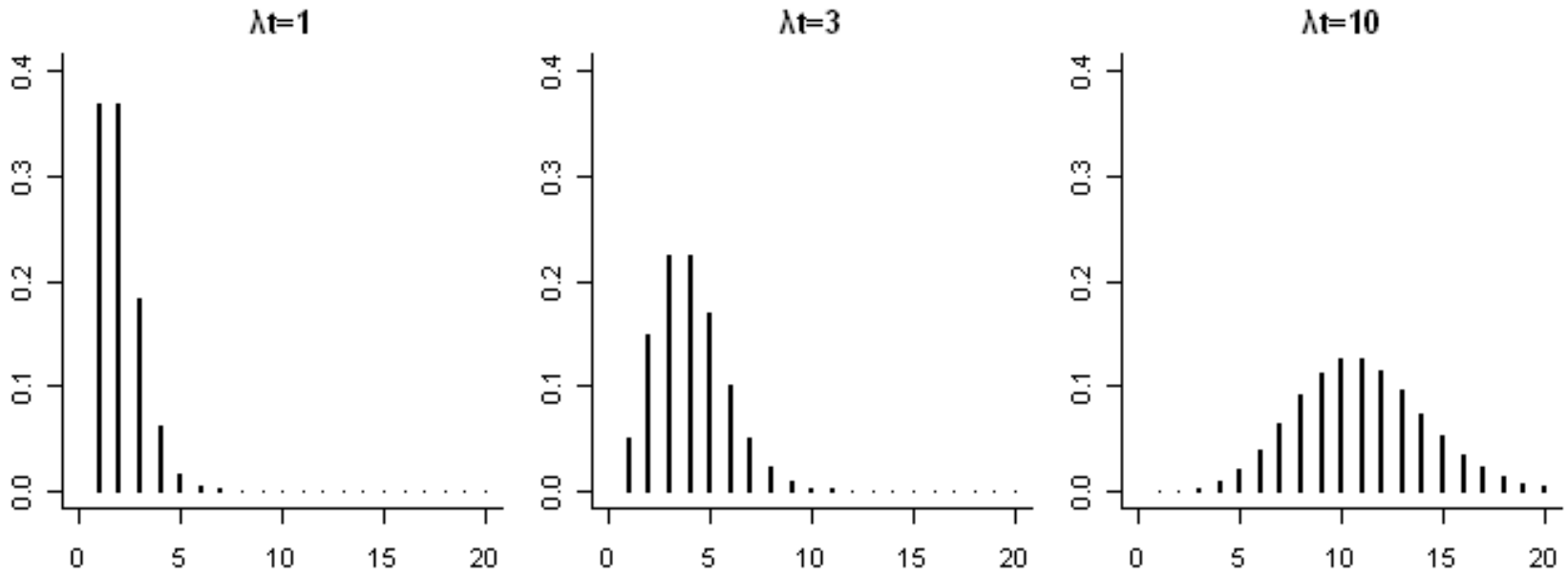
```
X= c(1.841,1.482,1.789,2.35,1.59,1.187,6.259,2.403,1.904,0.837,2.147,3.285,2.647,1.687,
    0.964,2.87,1.437,0.792,3.82,1.575,3.399,2.421,2.25,3.031,4.859,1.498,0.6)
y =c(0.1837,0.1254,0.1933,0.1620,0.1423,0.1406,0.4568,0.2287,0.2314,0.0861,0.1996,0.2353,0.2186,
0.127989918,0.125401821,0.244160823,

0.147764068,0.118189414,0.347453968,0.150133247,0.236855181,0.214618752,0.21409709,0.2572
91227,
     0.362829054,0.140651081,0.113849399)
plot(x, y, xlim=c(0,7), ylim=c(0,.5), pch=16, bty="l", xlab="Renda per capita",
    ylab="Telefonia Fixa per capita")
abline(lsfit(x,y))
```



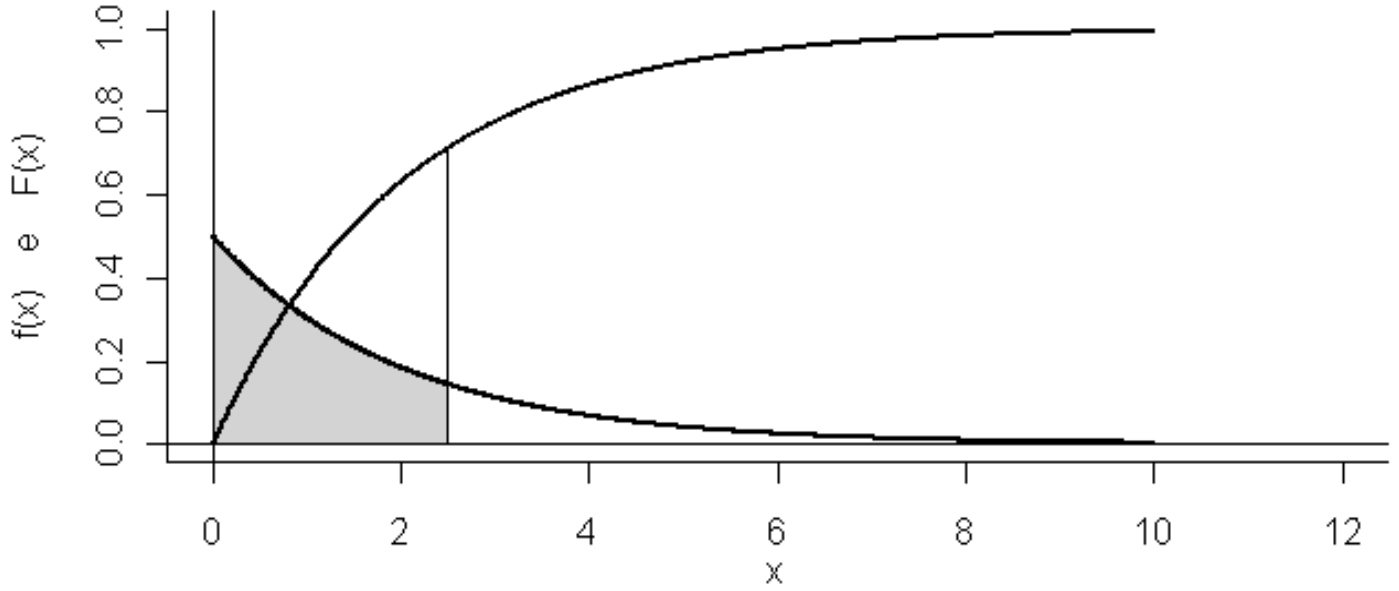
A1) – Pag 109 – Figura 4.5

```
x=0:20
y1=dpois(x,1); y2=dpois(x,3); y3=dpois(x,10)
names(y1)=x; names(y2)=x; names(y3)=x;
par(mfrow=c(1,3))
plot(y1,ylim=c(0,.4), type="h", xlim=c(0,20), lwd=2, bty="l", main="λt=1")
plot(y2,ylim=c(0,.4) , type="h", xlim=c(0,20) , lwd=2, bty="l", main="λt=3")
plot(y3,ylim=c(0,.4), , type="h", xlim=c(0,20) , lwd=2, bty="l", main="λt=10")
```



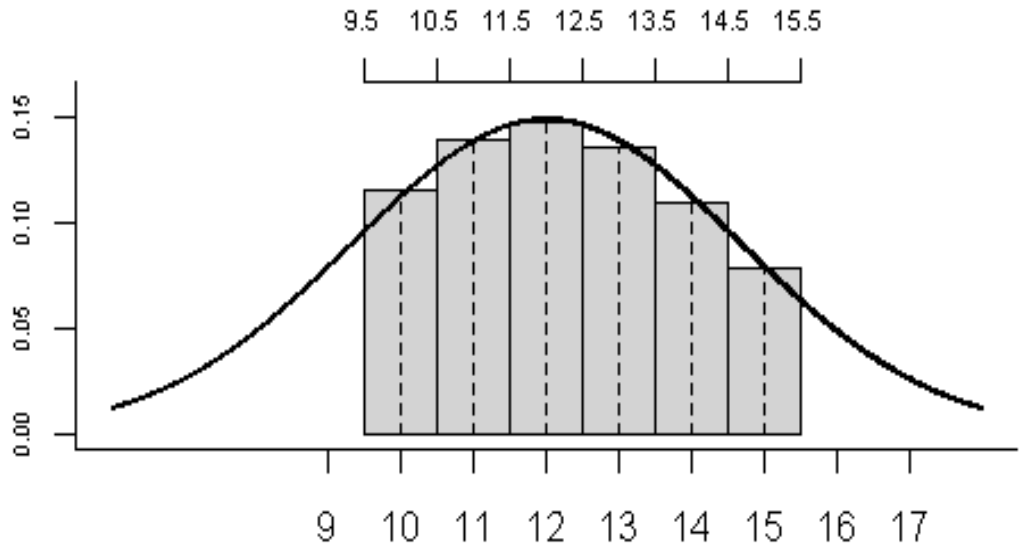
A1) – Pag 118 – Figura 4.12

```
x=seq(0,10,0.01)
plot(x,dexp(x, 1/2), type="l", xlim=c(0,12), ylim=c(0,1), bty="l", ylab="f(x)    e
F(x)")
for(i in seq(0, 2.5, 0.01)) segments(i, 0, i, dexp(i,1/2), col="lightgrey")
abline(v=0, h=0)
points(x,dexp(x, 1/2), type="l", lwd=2, bty="l")
points(x,pexp(x, 1/2), lwd=2, type="l")
segments(2.5,0, 2.5,pexp(2.5,1/2))
```



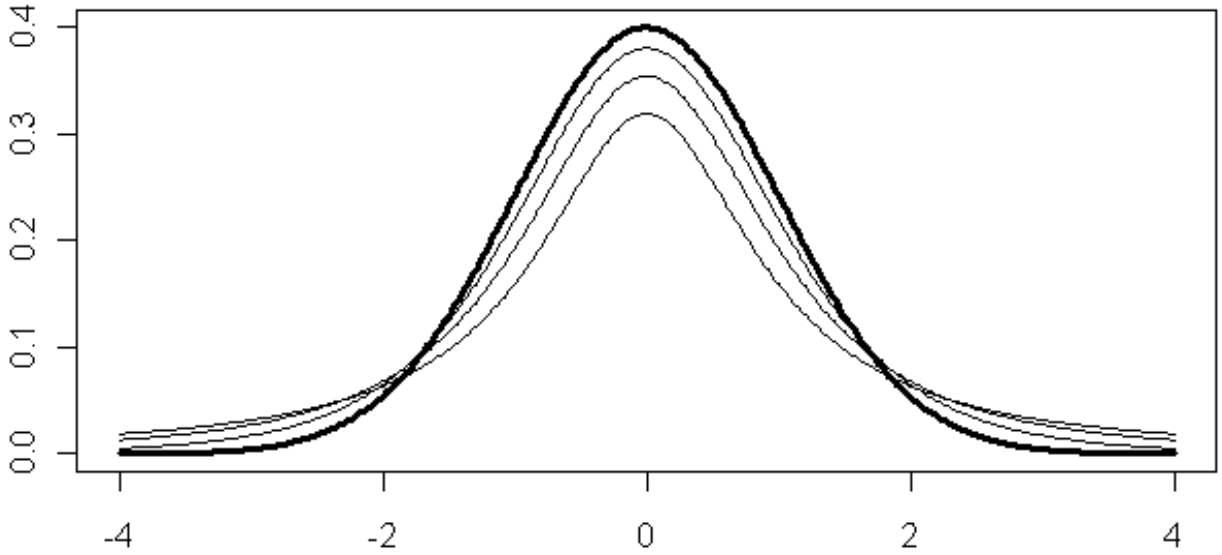
A1) – Pag 163 – Figura 6.8

```
plot(p, xlim=c(6,18), ylim=c(0,.16), type="n", bty="l", xaxt="n", xlab=" ", ylab=" ",
       cex.axis=.6)
for (i in 10:15) {
rect(i-.5,0, i+.5, dbinom(i,30,.4), col="lightgrey")###, lty=2)
segments(i,0,i,dbinom(i,30,.4), lty=2)
            }
x=seq(6,18,.01); lines(x, dnorm(x,12,sqrt(7.2)), lwd=2)
x=seq(6,18,.01); lines(x, dnorm(x,12,sqrt(7.2)), lwd=2)
axis(1, 9:16, cex.axis=.9)
axis(3, seq(9.5,15.5,1), cex.axis=.7)
```



A1) – Pag 198 – Figura 7.8

```
x1=seq(-4,4,.02)
plot(x1,dnorm(x1), type="l", lwd=3, xlab=" ", ylab=" ")
lines(x1, dt(x1,1)); lines(x1, dt(x1,2)); lines(x1, dt(x1,5))
```



A2–Resumo de comandos

a)Criação de dados

c(...)função genérica para combinar argumentos com o formando de um vetor

from:to gera uma sequênce; “:” tem prioridade de operator; pe.:1:3 + 1 is “2,3,4”

seq(from,to,by) gera uma sequência; by= especifica incremento;
length= especifica um comprimento desejado

rep(x,times) replicate x onúmero times de vezes; use each= para repetir cada elemento de x ; rep(c(1,2,3),2) gera 1 2 3 1 2 3; rep(c(1,2),each=2) gera 1 1 2 2

matrix(x,nrow=,ncol=) matrix; elementos de x se reciclam caso x não seja suficientemente grande

rbind(...)combina vetores em linhas num estrutura de matrizes de dados

cbind(...)combina vetores em colunas num estrutura de matrizes de dados

array(x,dim=) matriz com dados x; especificar dimensões como **dim**=c(3,4,2); elementos de x se reciclam caso x não seja suficientemente grande

factor(x,levels=) codifica um vetor x como um fator

gl(n,k,length=n*k,labels=1:n) gerar níveis (fatores), especificando o padrão de seus níveis; k é o número de níveis, e n é o número de repetições

data.frame(...) criar um banco de dados Por exemplo
data.frame(v=1:4,ch=c("g","B","casa","d"),n=5);

list(...)criar uma lista de argumentos; Por exemplo: list(a=c(1,2),b="hi",c=3i);

A2–Resumo de comandos

b) Informação de uma Variável

length(x) número de elementos em x

dim(x) Obter ou definir a dimensão de um objeto; dim (x) = c(3,2)

dimnames(x) Obter ou definir os nomes das dimensões de um objeto

nrow(x) número de linhas;

ncol(x) número de colunas

c) Seleção de dados e manipulação

choose(n, k) calcula a combinação de elementos escolhidos entre n, resultando: n! / [(n-k)! k!]

cut(x,breaks) divide x em intervalos (fatores); breaks é o número de intervalos de corte ou um vetor com os valores específicos

table(x) retorna uma tabela com as quantidades dos diferentes valores de x (tipicamente para variáveis dos tipos inteiros ou fatores)

sample(x, size) retira aleatoriamente com e sem reposição, elementos de tamanho SIZE, do vetor x, a opção replace = TRUE permite a retirada com reposição

prop.table(x,margin=) transforma a tabela como tabela de propoção marginal, margin=1 (com relação as linhas), margin=2 (com relação as colunas)

sort(x) Classifica os elementos de x em ordem crescente, para classificar em ordem decrescente rev(sort(x))

A2–Resumo de comandos

d) Estatísticas e operações matemáticas

mean(x) media dos elementos de x

median(x) mediana dos elementos de x

quantile(x,probs=) quantis de x correspondendo a uma dada probabilidade

var(x) ou **cov(x**) variância dos elementos de x (calculado com n-1),
se x é uma matriz a matriz de covariância é calculada

sd(x) desvio padrâo de of x

cor(x) matriz de correlação de x, se x for uma matriz ou (1 se x é um vector)

cor(x, y) correlação linear entre X e Y, ou matriz de correlação se eles são matrizes

round(x, n) arredonda os elementos de x para n casas decimais

sum(x) soma os elementos de x

prod(x) multilica os elementos de x

max(x) acha o máximo dos elementos de x

min(x) acha o mínimo dos elementos de x

range(x) equivalente a c(min(x),max(x)

cumsum(x) um vetor onde o ésimo elemento é a soma de x[1] até x[i]

cumprod(x) um vetor onde o ésimo elemento é o produto de x[1] até x[i]

cummin(x) um vetor onde o ésimo elemento é o mínimo de x[1] até x[i]

cummax(x) um vetor onde o ésimo elemento é o mínimo de x[1] até x[i]

sin,cos,tan,asin,acos,atan,atan2,log,log10,exp)

log(x, base) calcula o logaritmo de x na base=base

weighted.mean(x, w) media ponderada de x com peso= w

A2–Resumo de comandos

e) Corte e extração de dados

indexação de Vetores

x[n] n-ésimo elemento do vetor

x[-n] todos, menos o n-ésimo elemento

x[1:n] os primeiros n elemento

x[-(1:n)] elementos de n+1 até o final

x[c(4,3,2)] elementos especificados

x[y > 5] todo elementos de onde os valores de y são maiores que 5

x[x > 3 & x < 5] todo elementos entre 3 e 5

x["nome"] elemento denominado "nome"

indexação de Matrizes

x[i,j] elemento na linha i, coluna j

x[i,] linha i

x[,j] coluna j

x[,c(1,3)] colunas 1 and 3

x["nome",] linha nomeada "nome"

indexação de data frames

x[["nome"]] coluna chamada "nome"

x$nome equivalente a coluna chamada "nome"

A2–Resumo de comandos

f) Operação com Matrizes

t(x) transposta da matrix x

diag(x) retira a diagonal da matrix x

%*% multiplicação matricial

solve(a,b) resove a equação: a %*% x = b em relação a x

solve(a) matriz inversa de a

rowSum(x) soma das linhas da matrix x

colSum(x) soma das colunas da matrix x

rowMeans(x) média das linhas da matrix x

colMeans(x) id média das colunas da matrix x

g) Gráficos (Plotting)

plot(x, y) diagrama de disperção: plot dos pares (x,y) num sistema de eixos coordenados

hist(x) histogram dasfrequências of x

barplot(x) gráfico de barras of x; usar horiz=T ara barras horizontal

pie(x) gráfico de setores (pie-chart)

boxplot(x)

qqnorm(x) quantis de x em relação aos valores esperados de uma dist. Normal

A2–Resumo de comandos

parametros dos commando de Gráfico

type="p" especifica o tipo de plot, "p": pontos, "l": linhas, "b": pontos ligados por linhas, "o": idêntico. mas as linhas passam sobre os pontos, "h": linhas verticais, "s": escada (steps), os dados são representados pelas alturas verticais

xlim=, ylim= especifica os limites inferiores e superiores dos eixos, por exemplo, com xlim=c(1, 10) ou xlim=range(x)

xlab=, ylab= nomeia os eixos, o nome deve ser do tipo caracter

main= título principal, deve ser do tipo caracter

sub= sub-título (escrito em fonte menor)

ver em Comando par (Graphical parameters) outros parâmetros que também podem ser usados, como: bty(tipo de caixa), lwd (lagura da linha), lty (tipo de linha), pch(tipo de ponto), cex, xaxs, yaxs, xaxt,yaxt

h) Teste de hipóteses

t.test(),

prop.test(),

chisq.test()

aov(formula) analysis of variance model

anova(fit,...) analysis of variance (or deviance) tables for one or more fitted model objects

... Use o commando > ??"test" para procurar todos os testes disponíveis

A2–Resumo de comandos

i) Programming

function(arglist) expr para definer uma função

return(value)

if(cond) expr

if(cond) cons.expr else alt.expr

for(var in seq) expr

while(cond) expr

repeat expr

break

Usar chaves {} entre commandos, dlimitando o início e o fim de um grupo de comandos

A2–Resumo de comandos

j) Commandos auxiliaries em Gráficos

points(x, y) adiciona pontos (a opção type= pode ser usada)

lines(x, y) adiciona linhas (a opção type= pode ser usada)

text(x, y, labels, ...) adiciona texto na coordenada (x,y); um uso típico é:
plot(x, y, type="n"); text(x, y,names)

segments(x0, y0, x1, y1) desenha linhas do ponto (x0, y0) ao (x1, y1)

arrows(x0, y0, x1, y1, angle= 30, code=2) desenha seta do ponto
(x0, y0) ao (x1, y1)

abline(a,b) desenha uma reta de inclinação b e intercepto a

abline(h=y) desenha uma reta horizontal em y

abline(v=x) desenha uma reta vertical em x

abline(lsfit(x,y)) desenha uma reta da regressão feita em lsfit(x,y)

rect(x1, y1, x2, y2) desenha um retângulo que a esquerda inferior tem coordenadas (x1, y1) e o limite da direita superiores (x2, y2)

polygon(x, y) desenha um polígono que une os pontos com coordenadas X e Y

legend(x, y, legend) Acrescenta a lenda no ponto (x, y) com os símbolos dada pela legend

title()adiciona um título e, opcionalmente, um sub-título

axis(side) acrescenta um eixo na parte inferior (side = 1), à esquerda (2), na parte superior (3), ou à direita (4)

box()desenhar uma caixa em torno do plot

A2–Resumo de comandos

k) Comando par (Graphical parameters)

Todos estes comando podem ser definidos a nível global com o par (...), que especifica os parâmetros, mas também muitos podem ser usados como parametros dos commando de Gráfico .

mfcol vetor da forma c(nr, nc), que reparte a janela gráfica como uma matriz de nc, linhas e nr colunas, os plot´s são, então, elaborado em colunas

mfrow vetor da forma c(nr, nc), que reparte a janela gráfica como uma matriz de nc, linhas e nr colunas, os plot´s são, então, elaborado em linhas(matrix for row)

bty controla o tipo de caixa desenhada ao redor do enredo, valores permitidos são: "o","l", "7", "c", "u" ou "]" ; se bty="n" a caixa não é desenhada

lty controla o tipo de linhas, pode ser um inteiro ou string (1: "solid", 2: "dashed", 3: "dotted", 4: "dotdash", 5: "longdash", 6:"twodash",...), ou uma string de até oito caracteres (entre 0 e 9), que especifica o comprimento, em pontos ou pixels, dos elementos desenhados e os espaços em branco, por exemplo lty=”44” equivale a lty=2.

lwd número que controla a largura das linhas, default 1

pch controla o tipo de símbolo, pode também ser um número inteiro entre 1 e 25

ps um inteiro que controla o tamanho em pontos de textos e símbolos

pty um caracter que especifica o tipo da região, "s": quadrado, "m": máximal

A2–Resumo de comandos

I) Input and output

read.table(file) lê um arquivo em formato de tabela e cria um quadro de dados a partir dele, o separador default = é espaço em branco; header = TRUE ler a primeira linha como um cabeçalho de nomes de coluna; as.is = TRUE para evitar aque um vetore de caracteres seja convertido em factores; uso comment.char = """; para evitar "#" seja interpretado como um comentário, uso skip = n para pular n linhas antes da leitura de dados, consulte a ajuda para as opções de linha de nomeação, o tratamento NA, e outros

read.csv("filename",header=TRUE) idêntico, mas com os padrões estabelecidos para ler arquivos delimitados por vírgulas

read.delim("filename",header=TRUE) idêntico, mas com os padrões estabelecidos para ler arquivos delimitados por tabulações

read.fwf(file,widths,header=FALSE,sep=" ",as.is=FALSE)

ler uma tabela de dados, em formato fix, de largura m para um data.frame '; widths é um vetor inteiro, informando os tamanhos dos campos.

sink(file) saída de todos os comandos para um arquivo, até aparecer um comando sink () que desliga. .

write.table(x, file="",row.names=T,col.names=T,sep=" ") imprime x após a conversão para banco de dados

save(file,...) guarda os objetos especificados (...) no formato XDR

load()carregar o conjunto de dados salvos com o comando save

data(x) carrega um conjunto de dados especificado