

ESTATÍSTICA EXPERIMENTAL APLICADA – SOFTWARE R

1. INTRODUÇÃO AO SOFTWARE R

1.1. Instalação do software e formas de trabalho

- Entrar no site: <http://www.r-project.org>
- Selecione a opção **CRAN** (*Comprehensive R Archive Network*)
- Selecione o CRAN do Brasil, (Piracicaba, São Paulo, Curitiba, ...). Sugere-se o mais próximo de sua unidade.
- Selecione o sistema operacional que ira trabalhar (Linux, MacOS X ou **Windows**)
- Selecione a opção **base** - Binaries for base distribution (managed by Duncan Murdoch).
- Selecione a opção **Download R 2.14.0 for Windows** (ou a versão disponível).

Na janela inicial do R (Console), é apresentado o *prompt* (**>**), que é o *prompt* de linhas de comandos do R. Nas linhas de comandos podem se editados comandos do R ou expressões algébricas.

Para executar um comando ou calcular o valor da expressão algébrica da linha de comandos, basta usar a tecla **<ENTER>**.

Para sair do R, use:

> q() ou a opção de menu: **arquivo / sair**

Exemplos:

> 25+35

a) Alguns operadores

- aritméticos:

Adição (+), subtração (-), multiplicação (*), divisão (/), potenciação (^ ou **).

- Lógicos:

igual a (==), menor (<), maior (>), menor ou igual (<=), maior ou igual (>=) e diferente (!=).

b) Algumas funções aritméticas

sqrt(x) - raiz quadrada de x, **log(x,n)** - logaritmo de x na base n, **log(x)** - logaritmo neperiano de x, **log10(x)** - logaritmo decimal de x, **exp(x)** - e^x , **sin(x)** - seno de x (em radianos), **asin(x)** - arco-seno de x, **abs(x)** - modulo(x).

Exemplos:

> sqrt(125)

> log(15,2)

> 10==15

> pi>2

> sin(45)

> exp(1)

> cos(pi)

(pi = 180° em graus ou 3,141593 em radianos)

Para acessar comandos já executados, use as setas direcionais (**↑** e **↓**).

Para limpar a Janela Console use **CTRL + L**.

c) Ambientes de trabalho

O ambiente de trabalho no R pode se reduzir à janela Console ou trabalhar simultaneamente com duas janelas Console + Script (Editor R).

- a) Para abrir um Novo Script use a opção: Arquivo / Novo Script.
- b) Para abrir um Script disponível use a opção: Arquivo / Abrir Script.
- c) **Para navegar** pelas janelas Console e Script ou escolher a forma de visualizá-las simultaneamente, use a opção: **Janelas**.

O Script é um editor de textos do R que possibilita editar os comandos e transferi-los para serem executados na janela Console (CTRL + R) ou com a opção de menu: **Editar / Executar**. A vantagem dessa forma de trabalho é que a janela Script pode ser salva a qualquer momento (Para salvar, a extensão do nome do arquivo deve ser **.R**).

d) Diretórios de trabalho

Observação: As barras utilizadas na descrição de caminhos de diretórios no R são diferentes das utilizadas no Windows. Exemplo: para especificar o caminho na pasta AULAS na raiz do drive C: (No R - C:/AULAS, No Windows C:\AULAS)

Para saber o diretório de trabalho (default), use:

Sintaxe: `> getwd()`

Para alterar o diretório de trabalho, use:

Sintaxe: `> setwd("Caminho")`

Ou, na opção de menu: *Arquivo / Mudar Diretório*

Para criar um diretório pelo R, use:

Sintaxe: `> dir.create("Caminho")`

Para visualizar o conteúdo de um diretório:

- um diretório qualquer:

Sintaxe: `> dir("Nome_do_Diretório")`

Exemplo: `dir("C:/")`

- diretório de trabalho

Sintaxe: `> dir()` ou `> list.files()`

Exemplo:

Veja qual é o diretório de trabalho (default)..... `> getwd()`

Crie uma pasta de trabalho em C:\, com o nome EEAR `> dir.create("C:/EEAR")`

Defina esta pasta como a pasta de trabalho `> setwd("C:/EEAR")`

Visualize o conteúdo do diretório de trabalho `> dir()`

Veja qual é o diretório de trabalho `> getwd()`

1.2. Pacotes no R

Pacotes (*packages*) ou bibliotecas (*library*) são conjuntos de funções, exemplos, e documentações desenvolvidas para determinadas tarefas. No R podem-se encontrar pacotes desenvolvidos pelos responsáveis pelo R ou implementados por usuários.

A finalidade e sintaxe de alguns comandos R, relacionados a “pacotes”, são:

- a) Listar os Pacotes disponíveis no ambiente R (instalados e disponíveis).
Sintaxe: > `(.packages())`
- b) Listar os Pacotes disponíveis em seu computador (instalados e não disponíveis).
Sintaxe: > `.packages(all.available=TRUE)`

Com a instalação inicial, é instalado um conjunto básico de pacotes, ou seja:

Instalados e disponíveis:

[1] "stats" "graphics" "grDevices" "utils" "datasets" "methods" "base"

Instalados e não disponíveis:

[1] "base" "boot" "class" "cluster" "codetools" "compiler" "datasets" "foreign" "graphics"
[10] "grDevices" "grid" "KernSmooth" "lattice" "MASS" "Matrix" "methods" "mgcv" "nlme"
[19] "nnet" "rpart" "spatial" "splines" "stats" "stats4" "survival" "tcltk" "tools"
[28] "utils"

- c) Para disponibilizar um pacote instalado para uso (requerer).
Sintaxe: > `require("nome_do_pacote")`
Exemplo: > `require("MASS")`
 > `require("agricolae")`
 - d) Instalar pacotes (necessita conexão com a internet).
Sintaxe: > `install.packages("nome_do_pacote")`
Com este comando é solicitado o CRAN onde ele será acessado. Sugere-se que use o mais próximo de sua unidade.
Exemplo: > `install.packages("agricolae")`
- O pacote instalado permanece no seu computador até que você o remova. Para usá-lo você precisa disponibilizá-lo (requerê-lo) a cada vez que entrar no R.
- e) Remover (desinstalar) pacotes.
Sintaxe: > `remove.packages("nome_do_pacote")`
 - f) Auxílio sobre um pacote (instalado).
Sintaxe: > `help(package="nome_do_pacote")`
Exemplo: > `help(package="agricolae")`
 - g) Auxílio sobre um comando do R on-line. Necessita conexão com a internet.
Sintaxe: > `help(nome_do_comando)` ou `?nome_do_comando`
Exemplo: > `help(mean)` ou `?mean`
 - h) Auxílio sobre um comando em todos os pacotes instalados.
Sintaxe: > `help.search("nome_do_comando")` ou `??nome_do_comando`
Exemplo: > `help.search("mean")` ou `??mean`

Exercício 1:

- 1) Limpar a janela Console.
- 2) Fechar a janela Script.
- 3) Sair do R
- 4) Abrir o R e visualizar as informações da janela Console e as opções de menu.
- 5) Obter a Referencia Bibliográfica do R.

1.3. Classes de objetos no R

O R trabalha com as seguintes classes (ou tipos) de objetos: variável, vetor, lista, fator, matriz e data.frame. Cada objeto tem: nome, conteúdo e atributo.

- Os **nomes** de objetos são atribuídos pelo usuário e devem iniciar com uma letra (maiúscula ou minúscula), seguido de combinações de letras, números e alguns caracteres (desde que não reservados pelo R).
- O **conteúdo** é o valor do objeto.
- Os **atributos** podem ser numérico, caractere (*label*), complexo ou lógico.

O R faz diferença entre caracteres maiúsculos e minúsculos.

Todo objeto criado fica residente no ambiente R até que feche o software ou que o remova.

Os símbolos “<-”, “->” e “=” são usados para atribuição de valores.

O símbolo “#” é usado para comentários no R.

O ponto e vírgula (;) permite a edição de mais que um comando na mesma linha.

Exemplos:

```
> A <- 25      # atribui a A o valor 25
> A            # exibe na tela o valor de A - equivale a print(A).
> B <- A+30; B  # atribui a B o valor de A somado a 30 e exibe na tela o valor de B.
```

Para mostrar os objetos disponíveis na área de trabalho, use a função ls.

Sintaxe: **ls()**

Para remover um objeto use a função rm.

Sintaxe:

```
> rm(nome_do_objeto1, nome_do_objeto2, ...) # remove os objetos especificados
> rm(list=ls())                             # remove todos os objetos
```

Descrição das classes de objetos

a) Variável

Objeto ao qual é atribuído um valor que pode ter qualquer dos atributos.

Exemplos para cada tipo de atributo:

```
> x <- 10; x    # exemplo com atributo numérico e mostra o conteúdo do objeto x.
> rm(x)         # Remove o objeto x
```

Outros exemplos:

```
> "Maria" -> nome; nome
> y <- 3- 2i; y
> res <- (x==20); res
```

- Para saber o atributo de um objeto use: **mode(nome_do_objeto)**.
- Para verificar se o objeto tem um determinado atributo use:
is.nome_do_atributo(nome_do_objeto)
- Para saber a classe de um objeto use: **class(nome_do_objeto)**. Para alguns tipos de objetos (variável e vetor) o comando informa o atributo.

Exemplos:

```
> mode(x); mode(nome); mode(y); mode(res);
> is.numeric(x); is.complex(x); is.character(nome); is.logical(res)
> class(x); class(nome); class(y); class(res)
```

b) Vetor

Objeto com n valores (de mesmo atributo) que podem ser referenciados por um índice.

Exemplos:

```
> v1 <- c(2, 23, 14)
```

Observações:

- a) A letra *c* representa a função “concatenate” que concatena os valores entre parênteses.
- b) Para referenciar o vetor como um todo use: *nome_do_vetor*.
- c) Para referenciar apenas seus elementos use: *nome_do_vetor[índice_do_elemento]*.

Exemplos:

```
> v1; v1[2]; v1[c(1,3)]
```

```
> v2 <- c("Trat1", "Trat2", "Trat3", "Trat4"); v2; v2[3]
```

```
> v3 <- rep(5,4); v3 #rep(v,r) repete o valor v, r vezes.
```

```
> v4 <- 1:10; v4 # n:m cria uma sequência de n a m, 1 em 1
```

```
> v5 <- seq(2,14,3); v5 # seq(n,m,p) cria uma sequência com os valores de n a m, variando de p em p.
```

```
> v6 <- c(1,2,3,rep(6,3),seq(2,-14,-3)); v6
```

```
> mode(v1); mode(v2)
```

```
> class(v1); class(v2)
```

c) Lista

Objeto com n valores (que podem ter diferentes atributos) que podem ser referenciados por um índice.

Exemplos:

```
> lista1 <- list(nome="Maria", idade=18); lista1
```

```
> lista2 <- list("Holandeza",60,"parida",2); lista2
```

Observações:

- a) Para referenciar a lista como um todo use: *nome_da_lista*.
- b) Para referenciar apenas seus elementos use: *nome_da_lista\$nome_do_elemento* ou *nome_da_lista[índice_do_elemento]*.

Exemplos:

```
> mode(lista1); class(lista1)
```

```
> mode(lista1$nome); mode(lista1[1]); mode(lista1$idade); mode(lista1[2])
```

Observação: A maioria dos relatórios dos procedimentos estatísticos sai como lista.

d) Fator

É um tipo particular de vetor, onde cada elemento repetido (uma ou mais vezes) é considerado um nível do fator. Muito usado nas análises estatísticas – variáveis classificatórias.

Exemplos:

```
> trat <- as.factor(c(0,0,0,0,10,10,10,10,20,20,20,20)); trat # (as.factor ou factor)
```

```
> mode(trat); class(trat)
```

```
> vt=c("T1", "T1", "T1", "T2", "T2", "T3"); vt
```

```
> fvt=factor(vt); fvt
```

O comando *table* informa o número de repetições dos níveis do fator.

Exemplos:

```
> table(trat)
> table(fvt)
```

Para gerar níveis de um fator pode-se usar a função *gl* (*generate levels*);

Sintaxe: **gl(n,r[, labels= *níveis_do_fator*])**

gera um fator de *n* níveis com *r* repetições, utilizando os níveis especificados em *labels*, ordenados ou não.

```
> fat1 <- gl(5,4); fat1
> fat2 <- gl(3,4,labels=c(0,10,20)); fat2
> fat3 <- gl(3,4,labels=c("C","T1","T2")); fat3
```

e) Matriz

Objeto com $n \times m$ valores (*n* linhas e *m* colunas), com o mesmo atributo, que podem ser referenciados por dois índices.

Exemplos:

```
> c1 <- c(1,1,1,2,2,2); c2 <- c(1:3,1:3); c3=c(12,14,16,25,22,29) ; c1; c2; c3
> m1=cbind(c1,c2,c3); m1 #cbind cria a matriz tendo os vetores como colunas
> m2=rbind(c1,c2,c3); m2 #rbind cria a matriz tendo os vetores como linhas
> m3=matrix(c(1,1,1,2,2,2,1:3,1:3,12,14,16,25,22,29),ncol=3); m3
#ncol especifica o numero de colunas da matriz.
> m4=matrix(c(1,1,1,2,2,2,1:3,1:3,12,14,16,25,22,29),ncol=3, byrow="T"); m4
#byrow especifica os dados serão organizados por linhas.
```

Observações:

- Para referenciar a matriz como um todo use: *nome_da_matriz*.
- Para referenciar a linha *nl* da matriz use: *nome_do_matriz[nl,]* ou a coluna *nc* da matriz *nome_do_matriz[,nc]*.
- Para referenciar o elemento *i, j* da matriz use: *nome_da_matriz[i,j]*.

Exemplos:

```
> m1[,3] #especifica a coluna 3 da matriz m1
> m1[2,] #especifica a linha 2 da matriz m1
> m1[2,3] #especifica o elemento da linha 2 e coluna 3 da matriz m1
```

Algumas operações com matrizes:

```
> A=matrix(c(1,3,2,8,9,11,0,4,3), ncol=3,byrow=T); A
> B=matrix(c(rep(1,3),rep(2,3),rep(3,3)),ncol=3); B
> S=A+B; S # soma de matizes
> D=A-B; D # subtração de matizes
> P=A %*% B; P # produto de matizes
> T=t(A); T # transposta de matiz
> det(A); det(B) # determinante de matizes
> IA=solve(A); IA # inversa de matizes
> IB=solve(B); IB # inversa de matizes
```

f) data.frame

Objeto com $n \times m$ valores (n linhas e m colunas) em que as colunas podem ser de diferentes atributos, organizados na forma de banco de dados, ou seja, as colunas são as variáveis (vetores ou fatores) e as linhas os registros (atributos variados de acordo com a coluna).

Exemplos: Um experimento com 2 tratamentos e 5 repetições, obteve-se os Pesos:

Tratamento	Repetição				
	1	2	3	4	5
1	25,5	28,4	24,1	27,5	26,3
2	31,6	30,5	29,3	31,1	29,4

Neste caso, os dados teriam que ser organizados com três colunas: Tratamento, Repetição e Pesos, sendo Tratamento e Repetição como fatores e Peso como vetor. Os dados precisam ser organizados da forma:

tr	rp	peso
1	1	25,5
1	2	28,4
1	3	24,1
1	4	27,5
1	5	26,3
2	1	31,6
2	2	30,5
2	3	29,3
2	4	31,1
2	5	29,4

Criando o data.frame.

```
> tr <- gl(2,5)
> rp <- c(1:5,1:5)
> peso <- c(25.5, 28.4, 24.1, 27.5, 26.3, 31.6, 30.5, 29.3, 31.1, 29.4)
> df <- data.frame(tr,rp,peso); df
> mode(df)
> class(df)
```

1.4. Formas de entrada de dados

Para exemplificar a entrada de dados, considere os dados de um experimento onde foram tomados os comprimentos (y) para 3 tratamentos (T1, T2 e T3) e 5 repetições (1, 2, 3, 4, 5). Os resultados são apresentados na Tabela 1:

Trat	Rep				
	1	2	3	4	5
T1	35,0	19,0	31,0	15,0	30,0
T2	40,0	35,0	46,0	41,0	33,0
T3	39,0	27,0	20,0	29,0	45,0

a) Criando os dados no programa.

Edita-se as colunas do data.frame e então cria o data.frame.

Exemplo:

```
> trat <- c(rep("T1",5), rep("T2",5), rep("T3",5)); trat
> rep <- c(1:5,1:5,1:5); rep
> y <- c(35, 19, 31, 15, 30, 40, 35, 46, 41, 33, 39, 27, 20, 29, 45); y
> df1 <- data.frame(cbind(trat,rep,y)); df1
```

Verificação de classes e atributos:

```
> mode(df1); class(df1)
> mode(df1$trat); class(df1$trat); mode(df1$rep); class(df1$rep); mode(df1$y); class(df1$y)
```

Valores perdidos:

Supondo que a terceira repetição do tratamento 1 tenha sido perdida:

```
> yc <- c(35, 19, NA, 15, 30, 40, 35, 46, 41, 33, 39, 27, 20, 29, 45); yc
> mode(yc); mean(yc); mean(yc,na.rm=T)
# na.rm=T informa para executar a função desconsiderando os valores perdidos.
```

b) Importando arquivo txt (ASCII)

Sintaxe (Script: Imp_txt.R)

```
nome <- read.table("arquivo_txt", [header=T], [dec=","])
```

onde:

nome	nome do data.frame
arquivo_txt	nome do arquivo txt, com caminho completo.
[header=T]	Usar se os nomes das colunas (variáveis) do data.frame serão os cabeçalhos das colunas no arquivo .txt. Caso contrário o R assume os nomes V1, V2, (Variable 1, Variable 2, ...)
[dec=","]	Usar se o separador de decimais for a vírgula (,). Não usar se forem pontos.

Exemplo:

b1) crie um arquivo Excel e um arquivo ASC com os dados da Tabela 1. Salve na pasta C:\EEAR, com os nomes **A_exemplo.xls** e **A_Exemplo.txt**, respectivamente (O arquivo **A_exemplo.xls** encontra-se disponível no site).

b2) Importe o arquivo.

```
> dftxt <- read.table("C:/EEAR/a_exemplo.txt", header=T, dec=",")
> dftxt
```

b3) Visualize a classe e atributo do arquivo criado

```
> class(dftxt); mode(dftxt)
> class(dftxt[1]); mode(dftxt[1]); class(dftxt[2]); mode(dftxt[2])
```

c) Importando arquivo Excel

Para importar arquivos do Excel é necessário que o pacote RODBC esteja instalado e carregado (requerido).

Sintaxe (Script: Imp.xls.R):

```
install.packages("RODBC")
require(RODBC)
```

```
con=odbcConnectExcel[2]("arquivo_excel")
nome=sqlQuery(con,"select * from\"planilha$\" ")
[odbcClose(con)]
```

onde:

con é o nome da conexão com o arquivo, feita pelo *odbcConnectExcel*
[2] Usar se for o Excel 2007. Se não usar considera versos anteriores.
arquivo_excel nome do arquivo excel, com caminho completo
nome nome do data.frame
planilha nome da planilha a ser importada
[odbcClose(con)] Usar para fechar a conexão – sem o que, não é possível trabalhar na planilha.

Exemplo:

- c1) Crie um arquivo excel com os dados da Tabela 1 e salve-o no caminho: C:\EEAR, com o nome do arquivo **A_EXEMPLO.xls**, e nome da planilha **p1** (disponível no site).
- c2) Importe o arquivo usando:


```
> install.packages("RODBC")
> require(RODBC)
> con_ex=odbcConnectExcel("C:/EEAR/a_exemplo.xls")
> dfxls=sqlQuery(con_ex,"select * from\"p1$\" ")
> odbcClose(con_ex)
```
- c3) verificar a classe e atributos do arquivo criado e das colunas do data.frame


```
> class(dfxls); mode(dfxls); class(dfxls$tr); mode(dfxls$rp)
> class(dfxls$rp); mode(dfxls$rp); class(dfxls$y); mode(dfxls$y)
```

Comando attach

Este comando é usado para separar as variáveis de um data.frame.

O data-frame **arqxls** tem as colunas tr, rp e y

Para separá-las pode-se usar uma das opções:

a) `tr <- arqxls[,1]; rp <- arqxls[,2]; y <- arqxls[,3]; tr; rp; y`

ou

b) `attach(arqxls)`

`tr; class(tr); rp; class(rp); y; class(y)`

1.5. Alguns comandos para manipulação de arquivos.**a) Classificação de um data.frame por uma de suas variáveis.****Sintaxe:**

```
df[order(var,[decreasing="T"]),]
```

onde: **df** – nome do data.frame

var – variável pela qual será feita a ordenação

[decreasing="T"] – se quiser ordem decrescente (*default* – ordem crescente)

Exemplo: Classificar o data.frame df1, criado anteriormente, em ordem decrescente de peso.

```
> df1c <- df1[order(df1$peso,decreasing="T"),];df1c
```

b) Uso de controle de loops (FOR)

Usado para executar uma sequencia de comando controlado por uma variável contadora.

Sintaxe: `for (i in li:ls) {comandos}`

Onde: `i`=variável contadora;

`li`=limite inferior da variação;

`ls`=limite superior da variação;

`comandos`=comandos a serem executados para essa variação da variável `i`.

Exemplo: Criar um vetor `z` sendo $z=2*i$, para `i` variando de 1 a 10.

```
> z <- vector()
```

```
> for (i in 1:10){z[i] <- 2*i}; z
```

c) Comando condicional (IF...THEN...ELSE)

Usado para atribuir valor dependendo de condições.

Sintaxe: `if(condição) valor_se_verdadeira else valor_se_falsa`

Exemplo1: Crie uma variável `y` cujo valor será 100 se $x < 15$, e 200 se $x \geq 15$.

```
> x <- 10;
```

```
> if(x<15) y=100 else y=200; y
```

Exemplo2: Considere o data.frame `df1` criado anteriormente. Crie uma nova coluna “abate” sendo que: Se $\text{peso} \geq 30$ `abate=Sim`, caso contrário `abate=Não`.

```
> df1
```

```
> for(i in 1:length(peso)) {  
  if(df1$peso[i]>=30) df1$abate[i]="Sim" else df1$abate[i]="Não"}  
> df1
```

d) Renomear colunas de um data.frame

Sintaxe: `names(df)[c(c1,c2, ... ,cp)] <- c("NN1","NN2", ... ,"NNp")`

Onde: `df`=data.frame;

`c1, c2, ... , cp`=índices das colunas a serem renomeadas;

`NN1, NN2, ... , NNp`=novos nomes das colunas;

Exemplo: Considere o data.frame `df1`, criado anteriormente, e renomear as colunas `rep` e `pesos` por `rp` e `y`, respectivamente.

```
> df1
```

```
> names(df1)[c(2,3)] <- c("rp","y"); df1
```

e) Concatenar data.frames por linhas (os data.frames devem ter as mesmas colunas).

Sintaxe: `dfr <- rbind(df1,df2, ... ,dfp)`

Onde: `dfr`=data.frame resultante da concatenação;

`df1, df2, ... , dfp` = data.frames que serão concatenados.

f) Concatenar data.frames por colunas

Sintaxe: `dfr <- merge(df1,df2,by.x="Nome",all.x=T)`

Onde: `dfr` - data.frame resultante da concatenação;

`df1, df2` - data.frames que serão concatenados. Os data.frames a serem concatenados devem estar classificados pela variável **Nome**.

Nome - Variável que fará a correspondência dos arquivos a serem concatenados.

Exercício 2: Com os dados do arquivo (**A_Aluno.xls**). Criar um Script no R para:

- criar um data.frame (T1), com os dados da Turma 1 disponíveis no arquivo L2E2.XLS, Planilha Turma1 (importar o arquivo do Excel).
- criar um data.frame (T2), com os dados da Turma 2 disponíveis no arquivo L2E2.XLS, Planilha Turma2 (importar o arquivo do Excel).
- criar um data.frame (Ex), com os dados do Exame disponíveis no arquivo L2E2.XLS, Planilha Exame (importar o arquivo do Excel).
- criar um data.frame (T12), com todos os alunos (Turma 1 e Turma 2).
- criar um data.frame (T12E) incluindo em T1T2 as notas do Exame.
- renomear as colunas de T12E: PRATICA, PROVA, MEDIA e EXAME para: Prat, Prov, Med e Exam.
- calcular Media Final (MF), sendo que $MF = Med$ se $Med \geq 7$ e $MF = (Med + Exam)/2$, caso $Med < 7$.
- criar uma variável Avaliação (AV) sendo $AV = \text{Aprovado}$ se $MF \geq 7$, e $AV = \text{Reprovado}$ se $MF < 7$.
- salvar o Script com o nome Ex2.R.

Solução: Script Ex2.R

```
## Itens a), b) e c)
install.packages("RODBC")
require(RODBC)
con=odbcConnectExcel("C:/EEAR/L2E2.xls")
T1=sqlQuery(con,"select * from\"Turma1$\" ");T1
T2=sqlQuery(con,"select * from\"Turma2$\" ");T2
Ex=sqlQuery(con,"select * from\"Exame$\" ");Ex
odbcClose(con)
## item d)
T12 <- rbind(T1,T2);T12
## item e) - o merge exige que os data.frames estejam ordenados pela coluna referencia
T12c <- T12[order(T12$NOME),];T12c
Exc <- Ex[order(Ex$NOME),];Exc
T12E <- merge(T12c,Exc,by.x="NOME",all.x=T);T12E
## item f)
names(T12E)[2:4]<-c("Prat","Prov","Med");T12E
## item g)
for(i in 1:length(T12E$NOME))
  if(T12E$MEDIA[i]>=7) T12E$MF[i]<-T12E$MEDIA[i] else
    T12E$MF[i]<-(T12E$MEDIA[i]+T12E$EXAME[i])/2
T12E
## item h)
for(i in 1:length(T12E$NOME))
  if(T12E$MF[i]>=7) T12E$AV[i]<-"Aprovado" else
    T12E$AV[i]<-"Reprovado"
T12E
```

2. ESTATÍSTICAS DESCRITIVAS

São estatísticas que trazem informações sobre a posição, a dispersão e a distribuição de um conjunto de valores numéricos.

As principais estatísticas descritivas são: **Medidas de posição, Medidas de dispersão e Medidas separatrizes.**

A maioria dos softwares estatísticos tem um módulo que apresenta um conjunto dessas estatísticas.

a) Medidas de posição

São medidas que indicam um ponto em torno do qual se concentram os dados (a posição dos dados na reta real \mathbf{R}). Sejam $x_1, x_2, x_3, \dots, x_n$, n valores numéricos.

As medidas de posição mais usuais são:

Média aritmética

A média aritmética, indicada por m ou \bar{x} , é definida por:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

Se os valores $x_i, i=1, 2, \dots, k$ ($k < n$) ocorrem com freqüências $f_i, i=1, 2, \dots, k$, respectivamente, então a média aritmética pode ser obtida por:

$$\bar{x} = \frac{\sum_{i=1}^k f_i x_i}{\sum_{i=1}^k f_i}$$

Mediana

A mediana é o valor central do conjunto de valores (se n é ímpar, é o valor central dos valores organizados em ordem crescente ou decrescente, se n for par, ou a média dos dois centrais).

A mediana é tal que 50% dos valores a precedem e 50% a sucedem.

Moda

A moda é o valor que ocorre com maior freqüência. A moda pode não existir ou não ser única.

Média aritmética ponderada

Se aos n valores associam-se fatores de ponderação ou pesos $w_1, w_2, w_3, \dots, w_n$, respectivamente, a média aritmética ponderada é definida como:

$$\bar{x} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}$$

A escolha da medida de posição a ser usada depende das características dos dados e do objetivo do uso. Pode-se estar interessado no tamanho médio de uma espécie animal.

Situações que apresentam valores discrepantes fazem com que a média seja prejudicada, então a mediana pode ser mais apropriada.

Algumas propriedades das medidas de posição:

- a.1) A média é calculada a partir de todas as observações.
- a.2) A mediana não é prejudicada pela presença valores extremos, como é a média.
- a.3) Somando-se, subtraindo-se ou multiplicando-se os valores a uma constante, a média ficará somada, subtraída ou multiplicada pela constante.
- a.4) A soma dos desvios em relação à média é nula, ou seja, $\sum (x_i - \bar{x}) = 0$

A propriedade (a.3) é de extrema importância quando trabalhamos com unidades diferentes dos dados (kg, g, km, m etc.).

b) Medidas de dispersão

Como o próprio nome diz, são medidas que indicam quão dispersos encontram-se os valores.

As medidas de dispersão mais usuais são:

Amplitude total

É a diferença entre o maior e o menor valor (Amplitude total = $x_{\text{máx.}} - x_{\text{mín.}}$).

Soma de quadrados

É a soma dos quadrados dos desvios em relação à média aritmética.

$$SQ = \sum (x_i - \bar{x})^2$$

Variância

É a média das somas dos quadrados dos desvios em relação à média aritmética, geralmente denotada por S^2 .

$$S^2 = \frac{\sum (x_i - \bar{x})^2}{n}$$

Desvio Padrão

É a raiz quadrada da variância, geralmente denotada por s ($s = \sqrt{s^2}$).

Coefficiente de variação

É uma medida de dispersão relativa. É representado por CV e definido como:

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

Representa a porcentagem de variação por unidade de média.

Algumas propriedades das medidas de dispersão:

- b.1) A variância não tem a mesma unidade dos dados, o que ocorre com o desvio padrão.
- b.2) A soma dos quadrados dos desvios em relação à média aritmética (SQ) é a menor soma de desvios ao quadrado, ou seja: $\sum (x_i - \bar{x})^2 \leq \sum (x_i - M)^2$, $\forall M$.
- b.3) Somando-se ou subtraindo-se os valores a uma constante, a **variância** e a **soma de quadrados** não se alteram.
- b.4) Multiplicando-se os valores a uma constante, a **variância** e a **soma de quadrados** ficarão multiplicadas pela constante ao quadrado.

c) Medidas separatrizes

São medidas que representam as posições dos valores do conjunto, dividindo-o em partes iguais e podem ser: quartis, decis etc.

Os quartis dividem o conjunto de valores em quatro partes iguais, como mostra a Tabela a seguir:

Tabela 1 - Descrição dos quartis.

Estatística	Notação	Interpretação
1º quartil	q1	25% dos dados são valores o precedem
2º quartil	q2 = Mediana	50% dos dados são valores o precedem
3º quartil	q3	75% dos dados são valores o precedem

A diferença (q3-q1) é denominada intervalo interquartílico.

Outras separatrizes usuais são:

Decis: Dividem o conjunto de dados em dez partes iguais.

Percentis: Dividem o conjunto de dados em cem partes iguais.

No R, algumas dessas medidas podem ser obtidas por:

mean(x) – média, **var(x)** - variância, **sd(x)** – desvio padrão, **quantile(x,0.25)** – q1, **quantile(x,0.50)** – q2, **quantile(x,0.75)** – q3, **quantile(x)** – mínimo q1 q2 e q3 e máximo, **IQR(x)** – intervalo interquartílico, **min(x)** - mínimo, **max(x)** – máximo.

Têm-se ainda outras estatísticas como:

skewness(x) – coeficiente de assimetria, **kurtosis(x)** – coeficiente de curtose, **length(x)** – tamanho da amostra, **sum(x)** - somatória, **sum(x^2)** – somatória dos elementos ao quadrado, **sum(x-m)** – soma dos desvios em relação à média **m**.

Exemplo:

Considere que foram observados pesos de 10 animais, apresentados a seguir:

72,5 69,0 75,0 70,8 71,2 73,0 70,0 67,1 71,0 72,0

a) Criar um vetor Y com esses valores;

```
> Y <- c(72.5, 69, 75, 70.8, 71.2, 73, 70, 67.1, 71, 72)
```

b) Obter a média, variância, q1, q2, q3, mínimo e máximo.

```
> mean(Y); var(Y); quantile(Y)
```

c) Obter o tamanho da amostra.

```
> length(Y)
```

d) Obter os coeficientes de assimetria e curtose

```
> skewness(Y); kurtosis(Y) # observe que as funções não são encontradas nos pacotes instalados.
```

d.1) Pedir auxílio de onde encontrar a função skewness

```
> ?? skewness # Verá que ele esta no pacote agricolae
```

d.2) Instalar o pacote agricolae

```
> install.packages("agricolae")
```

d.3) Requerer o pacote agricolae

```
> require("agricolae")
```

d.4) Obtenha as estatísticas desejadas

```
> skewness(Y); kurtosis(Y)
```

Obter as somas dos quadrados dos desvios em relação à média

```
> med <- mean(Y); sqdm <- sum((Y-med)^2); med; sqdm
```

3. FUNÇÕES NO R - APLICAÇÃO EM ESTATÍSTICAS DESCRITIVAS

O R permite uso de funções já implementadas no software ou funções definidas pelo usuário.

Para usar uma função já implementada, basta chamá-la pelo nome, especificando os valores dos argumentos.

Uma função implementada é a função *summary*, que retorna seis estatísticas (mínimo, máximo, média, q1, q2, q3), muito usadas nos testes de diagnósticos.

Exemplo:

```
> x=c(4,7,9,8,6,7,9,5,8,10)
```

```
> summary(x)
```

Para criar uma função no R use a sintaxe:

```
nome_da_função = function(lista_de_argumentos) {  
  comandos_da_função  
  return(resultado)  
}
```

Onde: **nome_da_função** – nome da função.

lista_de_argumentos – lista dos argumentos separados por vírgulas.

comandos-da-função – comandos R necessários para definir a função. Se a função retorna mais que um valor pode-se colocá-los em uma lista, um data.frame etc.

resultado – nome do objeto que contém o resultado da função (pode ser um valor, uma lista ou um data.frame).

Exemplo:

Sabendo-se que o erro padrão da média é dado por: $epm = dp / \sqrt{n}$, onde *dp*=desvio padrão e *n*=tamanho da amostra, crie uma função para obter o *epm*.

```
> epm=function(ag)  
  {epm=sd(ag)/sqrt(length(ag))  
  return(epm) }
```

Observação: **ag** é um nome genérico do objeto (argumento).

Para usar a função basta colocar seu nome, seguido da variável para a qual deseja aplicar a função (no caso *x*). No caso: **> epm(x)**

Obter as estatísticas descritivas definidas na função *EstDesc* - disponível no site, para *x*.

- Abrir a janela Script e copiar a função nesta janela;
- executar os comandos da função;
- aplicar para o vetor *x*. **> est_desc(x)**.

Script da função para Estatísticas Descritivas: (S_EstDesc.R)

```
require(agricolae)  
est_desc = function(v) {  
  m=mean(v); n=length(v); soma=sum(v); sqnc=sum(v^2); sqc=sum(v-m)^2; var=var(v);  
  dp=sd(v); epm=dp/sqrt(n); ca=skewness(v); ck=kurtosis(v); q1=quantile(v,0.25);  
  q2=quantile(v,0.50); q3=quantile(v,0.75); iiq=IQR(v); min=min(v); max=max(v);  
  amp=max-min; res<-data.frame(media=m, n=n, soma=soma, sqnc=sqnc, sqc=sqc,  
  var=var, dp=dp, epm=epm, assim.=ca, kurt.=ck, q1=q1, q2=q2, q3=q3, iiq=iiq,  
  min=min, max=max, amp=amp)  
  return(res) }
```

4. GRÁFICOS DE DIAGNÓSTICOS – APLICAÇÕES NO R

Os gráficos mais comuns na análise exploratória de dados (diagnósticos) são – **histograma, boxplot, ramos e folhas e normalidade**.

Histograma – gráfico das frequências por intervalos de classes.

Box-plot – gráfico que inclui o máximo, mínimo e os quartis q1, q2 e q3 (também denominado gráfico dos cinco números).

Ramos e folhas – é um particular histograma, onde as classes são os ramos e os elementos dentro das classes são as folhas.

Normalidade – gráfico contendo os valores observados e a reta esperada, caso os dados tenham distribuição normal (quanto mais próximos da reta estiverem os pontos, melhor a aproximação à normal).

No R os gráficos: histograma, boxplot e de normalidade são feitos em **janelas gráficas**, o ramos e folhas é feito na janela **Console** do R.

Para salvar a janela gráfica, use o botão contrário do *mouse* e escolha entre copiar para a área de transferência ou salvar no formato desejado para o arquivo.

Exemplo 1:

Considere os valores para as variáveis Y1 e Y2:

y1	42,75	39,25	45,25	41,05	41,45	43,25	40,25	37,35	41,25	42,25	43,25	38,25
y2	42,25	39,35	45,25	41,15	41,25	43,15	40,15	39,15	41,35	42,15	43,25	58,45

- Digite os dados no Excel, planilha dados, e salve na pasta de trabalho com o nome A_GD.xls. Salve como arquivo ASC com o nome A_GD.txt (**obs.:** Encontra-se no site).
- Importe os dados (A_GD.txt) no R.
- Faça os gráficos de diagnósticos: Histograma, boxplot, ramos e folhas e de normalidade, para as variáveis Y1 e Y2.

Solução:

```
> arqgd <- read.table("C:/EEAR/A_GD.txt", header=T, dec=",")
> arqgd; mode(arqgd)
> y1 <- arqgd[,1]; y1
> y2 <- arqgd[,2]; y2
> hist(y1)
> boxplot(y1)
> stem(y1)
> qqnorm(y1); qqline(y1)

> hist(y2)
> boxplot(y2)
> stem(y2)
> qqnorm(y2); qqline(y2)
```

Observe que nos comando gráficos do R, os termos *line*, *curve* ou *points* não geram um novo gráfico, apenas incluem linha, curva ou pontos no gráfico em edição.

Como padrão, o R apresenta um gráfico por janela gráfica. Para criar um ambiente matricial para apresentação de gráficos, usa-se a função:

par(mfrow=c(nl,nc))

onde: **par** - Set or Query Graphical Parameters

mfrow – define as dimensões do ambiente matricial com **nl** linhas e **nc** colunas.

Exemplo 2:

Gere 50 observações com distribuição normal, média 10 e variância 5, e faça os gráficos de diagnósticos: Histograma, boxplot e de normalidade. Os gráficos devem ser colocados em uma janela gráfica com 1 linhas e 3 colunas.

A função para gerar n valores com distribuição normal com média m e desvio padrão dp , é definida como:

rnorm(n,m,dp)

onde: **n** é o número de observações

m a média e

dp o desvio padrão.

Solução:

```
> y <- rnorm(50,10,sqrt(5)); y
```

```
> par(mfrow=c(1,3))
```

```
> hist(y); boxplot(y); qqnorm(y); qqline(y)
```

5. TESTES DE HIPÓTESES

5.1 Introdução

Hipótese estatística: pode ser definida como uma afirmação sobre a distribuição de uma variável aleatória (no geral sobre seus parâmetros).

Exemplos: Em uma população com média μ e variância σ^2 , possíveis hipóteses seriam $H:\mu=0$; $H:\mu>50$; $H:\mu\neq 0$; $H:\sigma^2=100$; $H:\sigma^2<10$.

A hipótese estatística pode ser simples ou composta:

- **Simple:** se a hipótese especifica completamente a distribuição ($H:\mu=0$, $H:\sigma^2=100$).
- **Composta:** se a hipótese não especifica completamente a distribuição ($H:\mu>50$, $H:\sigma^2<10$).

Teste de hipóteses: Como o próprio nome diz, são critérios estatísticos que permitem rejeitar ou não hipóteses testadas, com determinado grau de confiança, baseados em valores amostrais.

Os testes de hipóteses, no geral, apresentam duas hipóteses:

- **Hipótese nula** (ou da nulidade), geralmente representada por H_0 , que é a hipótese natural colocada à prova.
- **Hipótese alternativa**, geralmente representada por H_1 ou H_A , que é a hipótese alternativa à hipótese colocada à prova.

Os testes de hipóteses devem seguir os passos:

Passo 1. Estabelecer as hipóteses (H_0 e H_1).

Passo 2. Obter uma estatística, com distribuição conhecida, que fique completamente definida sob H_0 .

Passo 3. Estabelecer os critérios do teste.

Todo teste estatístico apresenta dois tipos de erro:

Erro tipo I: Erro que se comete ao rejeitar H_0 , dado que ela é verdadeira, geralmente representado por α , e denominado nível de significância do teste.

Erro tipo II: Erro que se comete ao não rejeitar H_0 , dado que ela é falsa.

O critério mais comum em testes de hipóteses é fixar o erro Tipo I (nível de significância do teste).

Passo 4. Calcular o valor da estatística, item (2), para os valores da amostra.

Passo 5. Aplicar o critério do teste.

5.2 Exemplo de Aplicação

Para exemplificar, apresentemos esses passos em uma situação prática:

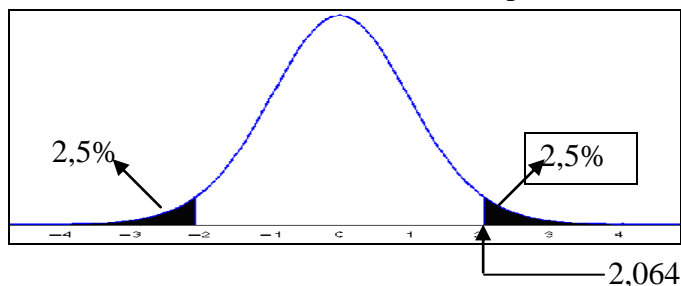
Exemplo: A quantidade de calorias de um produto (v.a. X) é tal que $X \sim N(\mu, \sigma^2)$. Para a indústria, $\mu=30$, mas para os concorrentes $\mu \neq 30$. Para avaliar o produto foi tirada uma amostra de tamanho 25, cujos valores são apresentados a seguir:

30,05	29,38	28,45	31,22	31,07	34,44	34,50	34,48	31,75	30,59
31,92	31,76	30,25	33,28	33,40	31,46	31,43	32,92	29,91	33,63
27,98	33,07	31,01	29,85	29,70					

Passo 1. Hipóteses a serem testadas: $H_0: \mu=30$ e $H_1: \mu \neq 30$.

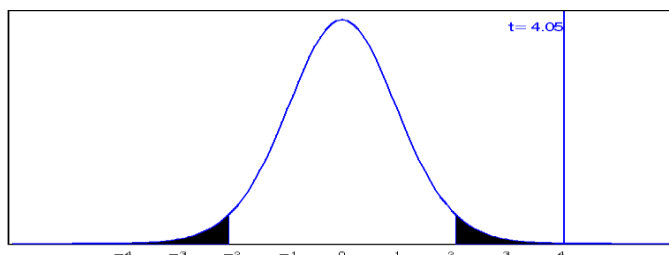
Passo 2. Sabe-se que $T = \frac{\bar{X} - \mu}{S/\sqrt{n}} \sim t(n-1)$, ou seja, T tem distribuição t de Student com $n-1$ graus de liberdade.

Passo 3. Fixando-se $\alpha=0,05$ (5%), tem-se pela tabela da distribuição t (ANEXO 1):



Região Crítica (Região de rejeição de H_0) = $(-\infty, -2,064) \cup (2,064, +\infty)$.

Passo 4. $T_c = \frac{31,5 - 30}{\sqrt{3,43/25}} = 4,050$



Passo 5. Aplicar o critério do teste.

Como T_c pertence à região crítica do teste, rejeita-se H_0 em favor de H_1 , ao nível de 5% de probabilidade.

Conclusão: Ao nível de 5% de probabilidade rejeita-se a hipótese da média de calorias ser 30 (H_0) em favor da hipótese da média de calorias ser diferente de 30 (H_1).

Os resultados dos testes geralmente apresentam o p-valor (*p-value*) dos testes, que é a área limitada pelo valor da estatística calculada. Se o p-valor for menor que o nível de significância do teste, o teste é significativo e rejeita-se H_0 em favor de H_1 a esses nível de significância.

6. TESTES DE NORMALIDADE – APLICAÇÕES NO R

Testes de Normalidade são testes apropriados para testar a hipótese dos valores apresentarem uma distribuição normal (H_0) contra a hipótese de não apresentarem.

Os testes mais usuais são:

Shapiro-Wilk – Teste de fácil aplicação sem o uso de software (manualmente). Possui a restrição de ser muito sensível à falta de simetria. São permitidos valores perdidos e é recomendado para situações em que o número de valores variam de 3 a 5000.

Kolmogorov-Smirnov – Teste usado para grandes conjuntos de dados. São permitidos valores perdidos e recomendado para situações com mais de 1500 valores.

Cramer-von Mises – É um teste não factível sem o uso de softwares, e é um dos testes mais usados na literatura. São permitidos valores perdidos e não tem restrições sobre o número de valores.

Anderson-Darlin – Apresenta resultados muito parecidos com os do **Cramer-von Mises**. São permitidos valores perdidos e não tem restrições sobre o número de valores.

No R, esses testes são disponíveis em alguns pacotes específicos. Os pacotes e as sintaxes são apresentadas a seguir:

Teste de Shapiro-Wilk - faz parte do pacote **stats** (parte da instalação básica), não precisa instalá-lo ou requerê-lo.

Sintaxe: `shapiro.test(vetor)`

Kolmogorov-Smirnov - faz parte do pacote **nortest**, (instalá-lo e requerê-lo).

Sintaxe: `lillie.test(vetor)`

Cramer-von Mises - faz parte do pacote **nortest**, (instalá-lo e requerê-lo).

Sintaxe: `cvm.test(vetor)`

Anderson-Darlin - faz parte do pacote **nortest**, (instalá-lo e requerê-lo).

Sintaxe: `ad.test(vetor)`

Apresenta-se a seguir um Script para testes de Normalidade.

Script para testes de Normalidade (S_TestNorm.R)

```
#####
##### Script para testes de normalidade #####
##### Escolha um dos testes #####
#Shapiro-Wilk ou Kolmogorov-Smirnov ou Cramer-von Mises ou Anderson-Darlin

# Teste de Shapiro-Wilk
shapiro.test(rs)

### Teste de Kolmogorov-Smirnov
#install.packages("nortest")
#require(nortest)
lillie.test(rs)

### Teste Cramer-von Mises
#install.packages("nortest")
#require(nortest)
cvm.test(rs)

#### Teste de Anderson-Darlin
#install.packages("nortest")
#require(nortest)
ad.test(rs)
```

Exemplo:

Para exemplificar os diagnósticos e testes de normalidade, considere os dados apresentados a seguir - pesos y1 e y2 referentes a duas espécies animais (E1 e E2).

y1			y2		
72,50	70,00	69,00	72,00	69,90	68,90
69,00	67,10	69,90	69,10	37,00	70,00
75,00	71,00	70,10	75,00	71,10	70,00
70,80	72,00	71,00	70,90	71,90	71,10
71,20	73,00	74,00	71,00	73,00	74,00
73,00	68,00	72,00	72,90	68,20	72,00

Exercício: Resolver os itens a) a f) usando a janela Script.

- Editar uma planilha no Excel e salvá-la com o nome A_THN.xls.
- Obter as medidas de **posição**: média e mediana; de **dispersão**: variância, desvio padrão e amplitude e **separatrizes** q1, q2, q3 e intervalo interquartilico (para y1 e y2).
- Usar a função *EstDesc* e obtenha todas as estatísticas descritivas (para y1 e y2).
- Fazer os gráficos de diagnósticos: histograma, boxplot, e de normalidade para as variáveis y1 e y2 (mesma janela). Os gráficos devem ser colocados em um documento do Word.
- Fazer os testes de normalidade: Shapiro-Wilk, Kolmogorov-Smirnov, Cramer-von Mises e Anderson-Darlin, para as variáveis y1 e y2 ($\alpha=5\%$).
- Retirar o outlier, de y2, e refazer o gráfico boxplot e o teste Cramer-von Mises.
 - Usando o Edit
 - Editando a planilha do Excel
- Salva o Script com o nome ExTN.R

Solução:

```
## Item a)
install.packages("RODBC")
require(RODBC)
con=odbcConnectExcel("C:/EEAR/A_THN.xls")
edtn=sqlQuery(con,"select * from\"Plan1$\" "); edtn
##odbcClose(con)

## Item b)
attach(edtn);y1;y2
media=mean(y1); media
mediana=quantile(y1,0.50); mediana
variância=var(y1); variância
desv_padr=sd(y1); desv_padr
q123=quantile(y1); q123

## Item c)
#Carregar na função est_desc, selecionar os comandos e executar
est_desc(y1)
est_desc(y1)

## Item d)
par(mfrow=c(1,2))
hist(y1); hist(y2)
boxplot(y1); boxplot(y2)
qqnorm(y1); qqline(y1); qqnorm(y2); qqline(y2)

## Item e)
shapiro.test(y1); shapiro.test(y2)
# os próximos testes precisam do pacote nortest
install.packages("nortest"); require(nortest)
lillie.test(y1); lillie.test(y2)
cvm.test(y1); cvm.test(y2)
ad.test(y1); ad.test(y2)

## Item f)
## f.1) - Editando no R
y2                # observe que o outlier é a observação número 8
y2c <- y2; y2c[8] <- NA; y2c
boxplot(y2c)
cvm.test(y2c)
## f.2) - Editando no Excel
odbcClose(con)    # se a conexão não foi fechada
edtn<-edit(edtn)   # edita-se o(s) valor(es) - substitua os valores a serem
                  retirados por NA e fecha a planilha

boxplot(edtn[,2])
cvm.test(edtn[,2])

## Item g)
Salvar o Script com o nome ExTN.R
```

7. TESTES PARA MÉDIAS, VARIÂNCIAS E PROPORÇÕES

Os testes de hipóteses usuais para médias, variâncias e proporções são classificados em:

- Teste para a média (uma amostra) – variância conhecida
- Teste para a média (uma amostra) – variância desconhecida
- Teste para comparação de duas médias (duas amostras – dados pareados ou não e com variâncias homogêneas ou não)
- Teste para a variância (uma amostra)
- Teste para comparação de duas variâncias (duas amostras)
- Teste para proporção (uma amostra)
- Teste para comparar duas proporções (duas amostras)

No R, os comandos utilizados com as respectivas sintaxes e os pacotes onde se encontram são apresentados a seguir:

a) Teste para média – variância conhecida (uma amostra)

O teste **z** para a média, quando a variância é conhecida faz parte do pacote **TeachingDemos**. (precisa instalá-lo e requerê-lo).

Sintaxe:

```
z.test(v, mu=media, stdev=desvio_padrão, [alternative="two.sided" | "greater" | "less"], [conf.level=1-alfa])
```

onde: **v** = vetor da amostra

média = valor da média a ser testado

desvio_padrão = desvio padrão conhecido

"two.sided" | "greater" | "less" – se a hipóse alternative é que a média é: diferente, maior que ou menor, respectivamente, da média colocada em prova. (*default= two.sided*)

conf.level = Apresenta o intervalo de confiança para a média, ao nível de confiança especificado (*default= 95%*)

b) Teste para média – variância desconhecida (uma amostra)

O teste **t** para a média, quando a variância é desconhecida, faz parte do pacote **stats**, (não precisa instalá-lo ou requerê-lo – faz parte de módulo básico).

Sintaxe:

```
t.test(v, mu=media, [alternative="two.sided" | "greater" | "less"], [conf.level=1-alfa])
```

onde: **v** = vetor da amostra

média = valor da média a ser testado

"two.sided" | "greater" | "less" – se a hipóse alternative é que a média é: diferente, maior que ou menor, respectivamente, da média colocada em prova. (*default= two.sided*)

conf.level = Apresenta o intervalo de confiança para a média ao nível de confiança especificado (*default= 95%*).

c) Teste para comparar duas médias – dados pareados ou não e com variâncias homogêneas ou não.

A resposta pode estar em dois vetores (**v1** e **v2**) ou em um vetor (**v**) - com variável classificatória (**cl**).

Sintaxe:

t.test([v1, v2] | [v~cl], [alternative="two.sided" | "greater" | "less"],
[conf.level=1-alfa], [var.equal=T | F], [paired=T | F])

onde: [v1, v2] se forem dados dois vetores - v1 e v2 são os vetores das amostra, ou [v~cl] se for dado um vetor (v) e a variável classificatória (cl).

"two.sided" | "greater" | "less" – se a hipótese alternativa é que: $v1 \neq v2$, $v1 > v2$ ou $v1 < v2$, respectivamente - *default*= $v1 \neq v2$ (*two.sided*).

conf.level = Apresenta o intervalo de confiança para a média ao nível de confiança especificado (*default*= 95%).

var.equal = se a igualdade das variâncias é verdadeira (T) ou falsa (F) – *default*=F.

paired = se os dados são pareados (T) ou não (F) – *default*=F.

d) Teste para variância (uma amostra)

O teste **sigma** para a variância de uma amostra faz parte do pacote **TeachingDemos**, (precisa instalá-lo e requerê-lo).

Sintaxe:

sigma.test(v, sigma=desv_pad, [alternative="two.sided" | "greater" | "less"],
[conf.level=1-alfa])

onde: v = vetor da amostra

desv_pad = valor do desvio padrão a ser testado

"two.sided" | "greater" | "less" – se a hipótese alternativa é que a média é: diferente, maior que ou menor, respectivamente, da média colocada em prova. (*default*= *two.sided*)

conf.level = Apresenta o intervalo de confiança para a média ao nível de confiança especificado (*default*= 95%).

e) Teste para comparar duas variâncias

A resposta pode estar em dois vetores (v1 e v2) ou em um vetor (v) - com variável classificatória (cl).

Sintaxe:

var.test([v1, v2] | [v~cl], [alternative="two.sided" | "greater" | "less"],
[conf.level=1-alfa])

onde: [v1, v2] se forem dados dois vetores - v1 e v2 são os vetores das amostra, ou [v~cl] se for dado um vetor (v) e a variável classificatória (cl).

"two.sided" | "greater" | "less" – se a hipótese alternativa é que: $\sigma_1^2 \neq \sigma_2^2$, $\sigma_1^2 > \sigma_2^2$ ou $\sigma_1^2 < \sigma_2^2$, respectivamente - *default*= $\sigma_1^2 \neq \sigma_2^2$ (*two.sided*).

conf.level = Apresenta o intervalo de confiança para a média ao nível de confiança especificado (*default*= 95%).

f) Teste para proporções (uma amostra)

O teste para proporção pode usar a distribuição binomial ou a normal aproximada.

Sintaxe:

prop.test | **binom.test**(ns, ta, [p=prop], [alternative="two.sided" | "greater" |
"less"], [conf.level=1-alfa])

onde: **prop.test** – usando a normal aproximada ou **binom.test** – usando a binomial.
ns – número de sucessos (uma amostra) ou o vetor dos números de sucessos (duas ou mais amostras)
ta – tamanho da amostra (uma amostra) ou o vetor dos tamanhos das amostras (duas ou mais amostras)
prop – proporção sendo testada (*default* = 0,5).
"two.sided" | **"greater"** | **"less"** – se a hipótese alternativa é \neq , $>$ ou $<$, respectivamente. (*default* = \neq).
conf.level = Apresenta o intervalo de confiança para a proporção, ao nível de confiança especificado (*default* = 95%).

g) Teste para proporções (duas ou mais amostras)

Sintaxe:

prop.test(c(ns1,ns2,ns3,...), c(ta1,ta2,ta3,...), [**alternative**="two.sided" | "greater" | "less"], [**conf.level**=1-alfa])

onde: **prop.test** – usando a normal aproximada ou **binom.test** – usando a binomial.
ns1, ns2, ns3, ... – números de sucessos das amostras 1, 2, 3, ..., respectivamente.
ta1, ta2, ta3, ... – tamanhos das amostras 1, 2, 3, ..., respectivamente.
"two.sided" | **"greater"** | **"less"** – se a hipótese alternativa é \neq , $>$ ou $<$, respectivamente. (*default* = \neq).
conf.level = Apresenta o intervalo de confiança para a proporção, ao nível de confiança especificado (*default* = 95%).

Apresentam-se a seguir os Scripts para testes de Hipóteses para Médias – casos a), b) e c) - **S_TH_Med.**; testes de Hipóteses para Variâncias – casos d) e e) - **S_TH_Var.**; e para proporções – casos f) e g) - **S_TH_Prop.**

Script para testes de Hipóteses para Médias (S_TH_Medias.R)

```
##### Testes de Hipóteses para Médias (uma ou duas amostras) #####
# 1. Teste para a média (uma amostra) – variância conhecida
#require(TeachingDemos)
z.test(v, mu=media, stdev=desvio_padrao, [alternative="two.sided" | "greater" | "less"], [conf.level=1-alfa])
onde: v = vetor da amostra
      media = valor da média a ser testado
      desvio_padrao = desvio padrão conhecido
      "two.sided" | "greater" | "less" – se a hipótese alternativa é que a média é:
      diferente, maior que ou menor, respectivamente, da média colocada em prova.
      (default= two-sided)
      conf.level = Apresenta o intervalo de confiança para a média, ao nível de
      confiança especificado (default= 95%)

# 2. Teste para a média (uma amostra) – variância desconhecida
#require(stats)
t.test(v, mu=media, [alternative="two.sided" | "greater" | "less"], [conf.level=1-alfa])
onde: v = vetor da amostra
      media = valor da média a ser testado
```


"two.sided" | "greater" | "less" – se a hipótese alternativa é que a média é: diferente, maior que ou menor, respectivamente, da média colocada em prova. (default= two-sided)

conf.level = Apresenta o intervalo de confiança para a média ao nível de confiança especificado (default= 95%).

3. Teste para comparação de duas médias (duas amostras - dados pareados ou não)
t.test([v1, v2] | [v~cl]), [alternative="two.sided" | "greater" | "less"], [conf.level=1-alfa], [var.equal=T | F], [paired=T | F]

onde: [v1, v2] se forem dados dois vetores - v1 e v2 são os vetores das amostra, ou [v~cl] se for dado um vetor (v) e a variável classificatória (cl).

"two.sided" | "greater" | "less" – se a hipótese alternativa é que: $v1 < v2$, $v1 > v2$ ou $v1 < v2$, respectivamente - default= $v1 < v2$ (two-sided).

conf.level = Apresenta o intervalo de confiança para a média ao nível de confiança especificado (default= 95%).

var.equal = se a igualdade das variâncias é verdadeira (T) ou falsa (F) – default=F.

paired = se os dados são pareados (T) ou não (F) – default=F.

Script para testes de Hipóteses para Variâncias (S_TH_Variancias.R)

Testes de Hipóteses para Variâncias (uma ou duas amostras)

1. Teste para a variância (uma amostra)

require(TeachingDemos)

sigma.test(v, sigma=desv_pad, alternative="two.sided" | "greater" | "less"), [conf.level=1-alfa]

onde: v = vetor da amostra

desv_pad = valor do desvio padrão a ser testado

"two.sided" | "greater" | "less" – se a hipótese alternativa é que a média é: diferente, maior que ou menor, respectivamente, da média colocada em prova. (default= two-sided)

conf.level = Apresenta o intervalo de confiança para a média ao nível de confiança especificado (default= 95%).

2. Teste para comparação de duas variâncias (duas amostras)

var.test([v1, v2] | [v~cl]), [alternative="two.sided" | "greater" | "less"], [conf.level=1-alfa]

onde: [v1, v2] se forem dados dois vetores - v1 e v2 são os vetores das amostra, ou [v~cl] se for dado um vetor (v) e a variável classificatória (cl).

"two.sided" | "greater" | "less" – se a hipótese alternativa é que: $v1 < v2$, $v1 > v2$ ou $v1 < v2$, respectivamente - default= $v1 < v2$ (two-sided).

conf.level = Apresenta o intervalo de confiança para a média ao nível de confiança especificado (default= 95%).

Script para testes de Hipóteses para Proporções (S_TH_Proporcoes.R)

Testes de Hipóteses para Proporções (uma ou mais proporções)

1. Teste para uma proporção (uma proporção)

[prop.test | binom.test] (ns, ta, [p=prop], [alternative="two.sided" | "greater" | "less"], [conf.level=1-alfa])

onde: **prop.test** – usando a normal aproximada ou **binom.test** – usando a binomial.

ns – número de sucessos (uma amostra) ou o vetor dos números de sucessos duas ou mais amostras)
ta – tamanho da amostra (uma amostra) ou o vetor dos tamanhos das amostras (duas ou mais amostras)
prop – proporção sendo testada (default = 0,5).
"two.sided" | "greater" | "less" – se a hipótese alternativa é ?, > ou <, respectivamente. (default =two.sided).
conf.level = Apresenta o intervalo de confiança para a proporção, ao nível de confiança especificado (default= 95%).

#2. Teste para comparar duas ou mais proporções (duas ou mais proporções)
[prop.test] (c(ns1,ns2,ns3,...), c(ta1,ta2,ta3,...), [alternative="two.sided" | "greater" | "less"], [conf.level=1-alfa])
 onde: **prop.test** – usando a normal aproximada ou **binom.test** – usando a binomial.
ns1, ns2, ns3, ... – números de sucessos das amostras 1, 2, 3, ..., respectivamente.
ta1, ta2, ta3, – tamanhos das amostras 1, 2, 3, ..., respectivamente.
"two.sided" | "greater" | "less" – se a hipótese alternativa é ?, > ou <, respectivamente. (default = ?).
conf.level = Apresenta o intervalo de confiança para a proporção, ao nível de confiança especificado (default= 95%).

Exemplos:

Para exemplificar estes testes de hipóteses serão resolvidos os exercícios da lista a seguir, cujos dados estão disponíveis no arquivo Excel (A_Thmvp.xls), planilhas Ex1 a Ex5.

LISTA DE EXERCÍCIOS – TESTES DE HIPÓTESES

1. Seja X uma v.a. correspondente ao peso do animal. Sabe-se que $X \sim N(\mu, \sigma^2)$ com $\mu=30g$ e $\sigma^2=40$. A média do peso pode variar se as condições de cultivo não são adequadas. Para verificar se as condições estão adequadas, obteve-se uma amostra (n=25), cujos valores foram (disponíveis no arquivo A_THMVP.xls, planilha Ex1):

30,26	39,62	21,29	31,35	26,12	29,83	26,31	39,97	28,01	37,54
25,28	33,51	39,57	41,42	23,69	40,53	25,88	23,11	34,38	35,15
40,16	29,51	26,09	24,96	26,47					

Fazer um teste de hipóteses para testar $H_0: \mu=30$ vs $H_1: \mu \neq 30$ ($\alpha=0,05$).
(Teste para a média – variância conhecida).

Resolução:

Hipótese testadas: $H_0: \mu=30$ vs $H_1: \mu \neq 30$, **Nível de significância:** $\alpha=0,05$.

```
> require(RODBC)
> con=odbcConnectExcel("C:/EEAR/ethmvp.xls")
> Ex1=sqlQuery(con,"select * from\"Ex1$\" "); Ex1
> install.packages("TeachingDemos")
> require(TeachingDemos)
> z.test(Ex1$Pesos, mu=30, stdev=sqrt(40), alternative="t")
```

Conclusão:

2. Uma máquina para colocar vacinas nos frascos enche-os com uma distribuição X tal que $X \sim N(500\text{ml}, \sigma^2)$. Cada dia toma-se uma amostra de 16 frascos para avaliar se a máquina está regulada. Em um dia obteve-se a amostra (disponíveis no arquivo thmvp.xls, planilha Ex2):

509,42	476,79	507,27	458,10	473,30	501,41	515,07	518,07
463,76	493,95	521,62	503,41	480,71	495,32	481,55	472,32

Observe que a média amostral é 492ml.

- 2.a) Fazer um teste de hipóteses para testar se a média difere ou não de 500ml, com $\alpha=0,01$. (**Teste para a média, variância desconhecida**).

Resolução:

Hipótese testadas: $H_0: \mu = 500$ vs $H_1: \mu \neq 500$, **Nível de significância:** $\alpha=0,01$.

> Ex2=sqlQuery(con,"select * from\"Ex2\$\" "); Ex2
> t.test(Ex2\$Y, mu=500, alternative="t", conf.level=.99)

Conclusão:

- 2.b) Fazer um teste de hipóteses para testar se a variância difere ou não de 400ml² ($\alpha=0,10$). (**Teste para a variância**)

Resolução:

Hipótese testadas: $H_0: \sigma^2 = 400$ vs $H_1: \sigma^2 \neq 400$, **Nível de significância:** $\alpha=0,01$.

> sigma.test(Ex2\$Y, sigma=sqrt(400), alternative="t", conf.level=0.90)

Conclusão:

3. A quantidade de calorias X de um produto segue uma distribuição $X \sim N(\mu, \sigma^2)$. O fabricante afirma que a quantidade de calorias é 30 cal. Um concorrente questiona e diz que é mais que 30. Toma-se uma amostra de tamanho 25 e os valores observados (disponíveis no arquivo thmvp.xls, planilha Ex3) foram:

36,25	30,69	32,68	30,43	31,24	32,73	30,06	32,71	30,44
31,48	28,56	29,29	31,28	33,3	32,32	30,63	31,13	32,72
30,63	31,37	35,46	25,15	30,55	36,57	29,87		

Observe que a média amostral é 31,5ml. Fazer um teste de hipóteses para testar se o fabricante ou o concorrente tinha razão ($\alpha=0,10$). (**Teste para a média, variância desconhecida**).

Resolução:

Hipótese testadas: $H_0: \mu = 30$ vs $H_1: \mu > 30$, **Nível de significância:** $\alpha=0,10$.

> Ex3=sqlQuery(con,"select * from\"Ex3\$\" "); Ex3
> t.test(Ex3\$Y, mu=30, alternative="g", conf.level=.90)

Conclusão:

4. Em um experimento de adubação de pastagens foram comparados dois tipos de adubação (química e orgânica). Seja X a altura da planta com $X_q \sim N(\mu_q, \sigma^2)$ e $X_o \sim N(\mu_o, \sigma^2)$. Observaram-se 12 parcelas de X_1 e 10 parcelas de X_2 . Os resultados (disponíveis no arquivo thmvp.xls, planilha Ex4) são apresentados a seguir:

X1	58,0	67,1	56,4	63,5	65,9	60,0	62,6	59,6	70,0	67,9	69,9	74,1
X2	74,9	82,7	71,8	70,0	71,8	78,0	69,6	75,8	62,8	72,4		

- 4.a) Fazer um teste de hipóteses para testar a hipótese de homocedasticidade ($\alpha=0,05$). (**Teste para duas variâncias**).

Resolução:

Hipótese testadas: $H_0: \sigma_1^2 = \sigma_2^2$ vs $H_1: \sigma_1^2 \neq \sigma_2^2$, **Nível de significância:** $\alpha=0,05$.

> Ex4vv=sqlQuery(con,"select * from \"Ex4vv\$\" "); Ex4vv

> var.test(Ex4vv\$Y1, Ex4vv\$Y2)

ou

> Ex4cv=sqlQuery(con,"select * from \"Ex4cv\$\" "); Ex4cv

> var.test(Ex4cv\$Prod~ Ex4cv\$Trat)

Conclusão:

- 4.b) Fazer um teste de hipóteses para testar as hipóteses: $H_0: \mu_q = \mu_o$ vs $H_1: \mu_q \neq \mu_o$ ($\alpha=0,05$). (**Teste para duas médias**).

Hipótese testadas: $H_0: \mu_q = \mu_o$ vs $H_1: \mu_q \neq \mu_o$, **Nível de significância:** $\alpha=0,05$.

> t.test(Ex4vv\$Y1, Ex4vv\$Y2, var.equal=T)

ou

> t.test(Ex4cv\$Prod~ Ex4cv\$Trat, var.equal=T)

Conclusão:

5. Avalio-se o diâmetro da pata de 25 animais. Pata com o procedimento cirúrgico - CPC e pata sem o procedimento cirúrgico SPC. Seja X o diâmetro da pata com $X_{cpc} \sim N(\mu_{cpc}, \sigma^2)$ e $X_{spc} \sim N(\mu_{spc}, \sigma^2)$. Os resultados (disponíveis no arquivo thmvp.xls, planilha Ex5) são apresentados a seguir.

SPC	14,8	15,1	14,7	14,7	15,1	16,2	14,5	14,5	14,1	14,1	15,1	15,1	15,5
	14,7	14,8	14,4	14,6	15,2	14,8	14,8	15,0	15,8	14,7	14,1	15,8	
CPC	14,4	14,9	15,3	14,7	14,5	14,8	14,9	15,4	15,3	15,4	14,9	14,8	15,1
	15,6	15,0	15,7	14,4	14,9	14,9	15,2	15,7	15,0	15,0	15,3	15,4	

Fazer um teste de hipóteses para testar as hipóteses: $H_0: \mu_{cpc} = \mu_{spc}$ vs $H_1: \mu_{cpc} \neq \mu_{spc}$ ($\alpha=0,05$). **Teste para comparar médias, dados pareados**.

Resolução:

Hipótese testadas: $H_0: \mu_1 = \mu_2$ vs $H_1: \mu_1 \neq \mu_2$, **Nível de significância:** $\alpha=0,05$.

> Ex5=sqlQuery(con,"select * from \"Ex5\$\" "); Ex5

> t.test(Ex5\$SPC, Ex5\$CPC, alternative="l", paired=T)

Conclusão:

6. Um produtor afirma que 60% dos peixes pesam mais que 1kg. O comprador quis testar a hipótese do vendedor e tomou uma amostra de 200 peixes, dos quais 104 pesaram mais de 1kg. Fazer um teste de hipóteses para testar as hipóteses: $H_0: p > 60\%$ vs $H_1: p < 60\%$ ($\alpha = 0,05$). **Teste para proporção (uma amostra)**

Resolução:

Hipótese testadas: $H_0: p > 60\%$ vs $H_1: p < 60\%$, **Nível de significância:** $\alpha = 0,05$.

> prop.test(104,200,p=0.6, alternative="l", conf.level=0.90)

ou

> binom.test(104,200,p=0.6, alternative="l", conf.level=0.90)

Conclusão:

7. Para testar se a proporção de novilhas de uma fazenda F1 é a mesma que na fazenda F2, o agricultor tomou uma amostra de tamanho 100 na F1 - obteve 25 novilhas, e uma amostra de tamanho 120 na F2 – obteve 45 novilhas. Fazer um teste de hipóteses para testar as hipóteses: $H_0: p_1 = p_2$ vs $H_1: p_1 \neq p_2$ ($\alpha = 0,05$). **Teste para proporções (duas amostras)**

Resolução: Usando a normal aproximada).

> prop.test(c(25,45),c(100,120), alternative="t", conf.level=0.95)

Conclusão:

8. TESTE QUI-QUADRADO PARA TABELAS DE CONTINGÊNCIA

Tabelas de Contingência são tabelas de frequências de dupla entrada, ou seja, tabelas de frequências envolvendo duas variáveis classificatórias, uma nas linhas e outra nas colunas. Esses testes utilizados para testar a independência das variáveis da tabela (linhas e colunas). Na prática pode-se ter a tabela de frequências ou todos os registros, para então criar a tabela de frequências. O teste mais usado é o Teste Qui-Quadrado, que tem a pressuposição de que as frequências esperadas sejam ≥ 5 . Se a pressuposição não for satisfeita usa-se o Teste Exato de Fisher (Teste não paramétrico)

No R, os comandos utilizados com as respectivas sintaxes e os pacotes onde se encontram são apresentados a seguir:

a) Teste Qui-quadrado:

Sintaxe:

chisq.test(x[, y])[\$expected]

onde: **x** – se x for a matriz da tabela de contingência; ou **x, y** – se x e y são os registros (fatores)

\$expected – para mostrar a tabela dos valores esperados, caso os fatores da tabela forem independentes.

b) Teste Exato de Fisher

Sintaxe:

fisher.test(x[, y], [workspace=200], [conf.level=1-alfa])

onde: **x** – se x for a matriz da tabela de contingência; ou **x, y** – se x e y são os registros (fatores)
workspace – define uma área de trabalho (desnecessário para tabelas 2x2)
conf.level – para retornar o intervalo de confiança.

Exemplo 1 – Quando se tem a tabela de frequências

Em um experimento para testar se a ocorrência de uma determinada doença depende da espécie animal, obteve-se a tabela de frequências:

Espécies	Presença da doença	
	Sim	Não
E1	25	90
E2	180	230
E3	2	25
E4	4	8

Fazer um teste de hipóteses para testar se a presença da doença independe ou não da espécie.

Resolução:

```
> mx <- matrix(c(25,90,180,230,2,25,4,8), ncol=2, byrow=T); mx
> chisq.test(mx)
> chisq.test(mx)$expected
```

Como a pressuposição para o teste qui-quadrado não foi satisfeita, analisamos os dados pelo Teste Exato de Fisher.

```
> fisher.test(mx)
```

Conclusão:

Exemplo 2 – Quando se tem os registros.

- Visualize o arquivo folic.txt (disponível no site). São dados de um experimento de reprodução animal em que cada folículo era classificado quanto ao Diâmetro (DF = 1 - pequeno, 2 - médio e 3 - grande); Qualidade (QF = 1 - ruim, 2 - regular, 3 - bom) e qualidade do ovócito (QO = 1 - infértil, 2 - fértil).
- Salve-o na pasta EEAR e importe para o R.
- Proceda os testes de hipóteses considerando as variáveis duas-a-duas.

Resolução:

```
> folic <- read.table("C:/EEAR/folic.txt")
> folic
> v1<- as.factor(folic$V1); v1
> v2<- as.factor(folic$V2); v2
> v3<- as.factor(folic$V3); v3

> table(v1,v2)
> chisq.test(v1,v2)$expected
> chisq.test(v1,v2)

> table(v1,v3)
> chisq.test(v1,v3)$expected
> chisq.test(v1,v3)
```

```
> table(v2,v3)
> chisq.test(v2,v3)$expected
> chisq.test(v2,v3)
```

Conclusões:

9. GRÁFICOS - GRÁFICOS DE DISPERSÃO NO R

Alguns comandos para gráficos de dispersão:

• Comando plot

Sintaxe: **plot**(x, y, [**type**="tipo"], [**col**="cor" | "número_da_cor"] [**pch**="símbolo" | número_do_símbolo], [**lwd**=número_espessura_da_linha], [**lty**=efeito_da_linha] [**main**="título_do_gráfico"], [**xlab**="título_do_eixo_x"], [**ylab**="título_do_eixo_y"], [**font** = número] , [**font.main**=número], [**font.lab**=número])

Onde: x e y são os vetores com os valores de (x) - abcissas e de (y) – ordenadas.

tipo= **p** (pontos), **l** (linhas), **b** (ambos – pontos e linhas), **o** (linha segmentada), **s** (escada), **c** (linha quebrada), **h** – histograma, - *default*=p.

cor = **red**, **blue**, **green**, ... (ou 1=preto, 2=vermelho, 3=verde, 4=azul, ...).

pch = "\$", "k", "@",ou 1 (círculo), 2 (triângulo), 3 (+), 4 (x), 5 (losango),

lwd = espessura da linha

lty = efeito da linha – tracejada, pontilhada,

• Comando R para colocar legendas num plot.

legend("posição",**title**="título_legenda",**legend**=levels(tr), [**col**="cor" | "número_da_cor"] [**pch**="símbolo" | número_do_símbolo], [**lwd**=número_espessura_da_linha], [**lty**=efeito_da_linha]

• Comando R para configurar as escalas dos eixos.

Sintaxe: **plot**(**c**(xmin,xmax), **c**(ymin,ymax), "n")

Onde: **xmin** e **xmax** são os limites da escala do eixo x e **ymin** e **ymax** são os limites da escala do eixo y.

n – indica para editar apenas os eixos.

• Comando R para incluir linhas, retas, pontos, segmentos no gráfico.

a) Incluir retas ($y=a+bx$)

Sintaxe: **abline**(**a**=valor_de_a, **b**=valor_de_b, [**type**], [**col**], [**pch**], [**main**], [**xlab**], [**ylab**], [**lwd**], [**lty**])

Onde: **valor_de_a** = valor do parâmetro a (intercepto), **valor_de_b** = valor do parâmetro b (coeficiente angular).

[**type**], [**col**], [**pch**], [**main**], [**xlab**], [**ylab**], [**lwd**], [**lty**] – como definidos anteriormente.

b) Incluir linhas de grade

```
abline(h=seq(0,14,1),v=seq(0,5,0.5),lty=3,col="gray")
```

c) Incluir linhas – horizontal ou vertical

Sintaxe: **abline(v=valor_de_x, h=valor_de_y, [type], [col], [pch], [main], [xlab], [ylab], [lwd], [lty])**

Onde: **valor_de_x** = valor de x - por onde passará a reta vertical, **valor_de_y** = valor de y - por onde passará a reta horizontal, **[type]**, **[col]**, **[pch]**, **[main]**, **[xlab]**, **[ylab]**, **[lwd]**, **[lty]** – como definidos anteriormente.

d) Incluir segmentos de reta

Sintaxe: **segments(x_a, y_a, x_b, y_b)**

Faz um segmento de reta de A(x_a, y_a) a B(x_b, y_b)

e) Incluir pontos

Sintaxe: **points(x,y[,v1,v2])**

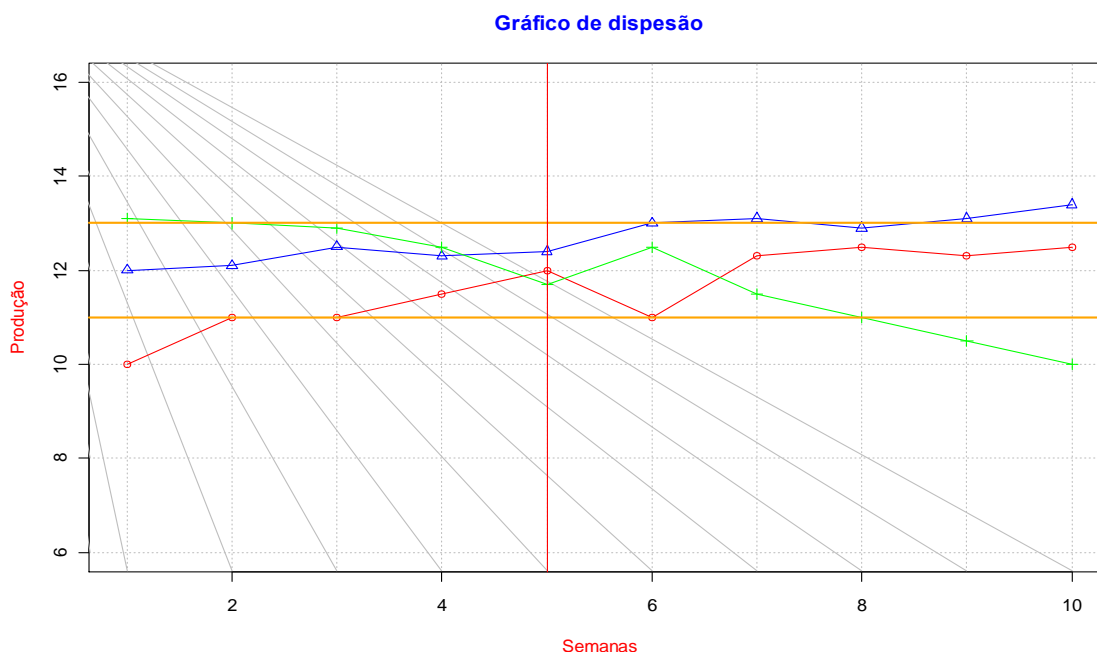
Onde: x,y são as coordenadas do ponto a ser incluído ou [v1,v2] que são os vetores das coordenadas dos pontos.

Exemplo:

Editar um programa R para, a partir dos dados a seguir, que estão disponíveis no site no arquivo A_GRAF.txt.

Tratamento	Semanas									
	1	2	3	4	5	6	7	8	9	10
T1	10	11	11	11.5	12	11	12.3	12.5	12.3	12.5
T2	12	12.1	12.5	12.3	12.4	13	13.1	12.9	13.1	13.4
T3	13.1	13	12.9	12.5	11.7	12.5	11.5	11	10.5	10

Editar o gráfico:



- **Comandos – usando o plot :**

```
> graf <- read.table("C:/EEAR/GRAF.TXT", header=T)
> graf
> x = graf[,1]
> y1 = graf[,2]; y2 = graf[,3]; y3 = graf[,4]
> x; y1; y2; y3
> plot(c(1,10),c(6,16), "n", main="Gráfico de dispesão", col.main="blue",
      xlab="Semanas", ylab="Produção", col.lab="red") # configurando eixos
> abline(h=seq(6,16,2),v=seq(1,10,1),lty=3,col="gray") # linhas de grade
> lines(x,y1, type="o",col="red", pch=1) # linha y1
> lines(x,y2, type="o",col="blue", pch=2) # linha y2
> lines(x,y3, type="o",col="green", pch=3) # linha y3
> abline(h=c(11,13), col="orange", lwd=2) # linha de ref. Horiz
> abline(v=5, col="red", lwd=1) # linha de ref. Vert.
```