UNIVERSIDADE FEDERAL DE MINAS GERAIS

INSTITUTO DE CIÊNCIAS EXATAS – ICEx

DEPARTAMENTO DE ESTATÍSTICA

BIOESTATÍSTICA BÁSICA USANDO O AMBIENTE COMPUTACIONAL R

Autores

Aloísio Joaquim Freitas Ribeiro *(coordenador)*
Edson Francisco Ferreira
Ilka Afonso Reis (*colaboradora*)
Lourdes Coral Contreras Montenegro (*colaboradora*)

Esta apostila é parte integrante do material produzido pelo projeto "Modernização do Ensino da Disciplina Introdução à Bioestatística – EST179" sob o Edital PROGRAD/UFMG 002/2009.

Índice

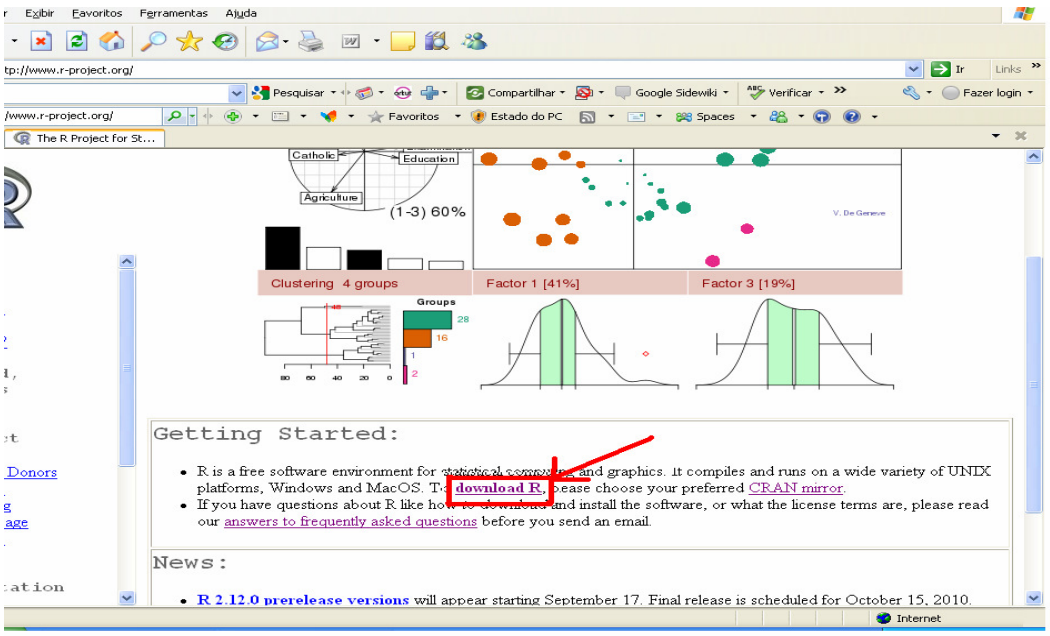
Aula 1 - Introdução ao R

O R é ao mesmo tempo uma linguagem de programação e um ambiente para computação estatística e gráfica. Algumas das suas principais características são: o seu caráter gratuito e a sua disponibilidade para uma gama bastante variada de sistemas operacionais. Apesar do seu caráter gratuito o R é uma ferramenta bastante poderosa com boa capacidade de programação. Ele tem sido utilizado por pesquisadores das mais diversas áreas na análise de dados. O objetivo deste texto é introduzir aos alunos da disciplina Introdução à Bioestatística o uso do R. Esperamos com isto tornar mais interessante o curso de Introdução à Bioestatística, permitindo ao aluno utilizar as técnicas estatísticas aprendidas na disciplina e aprimorar o entendimento dos conceitos estatísticos estudados.

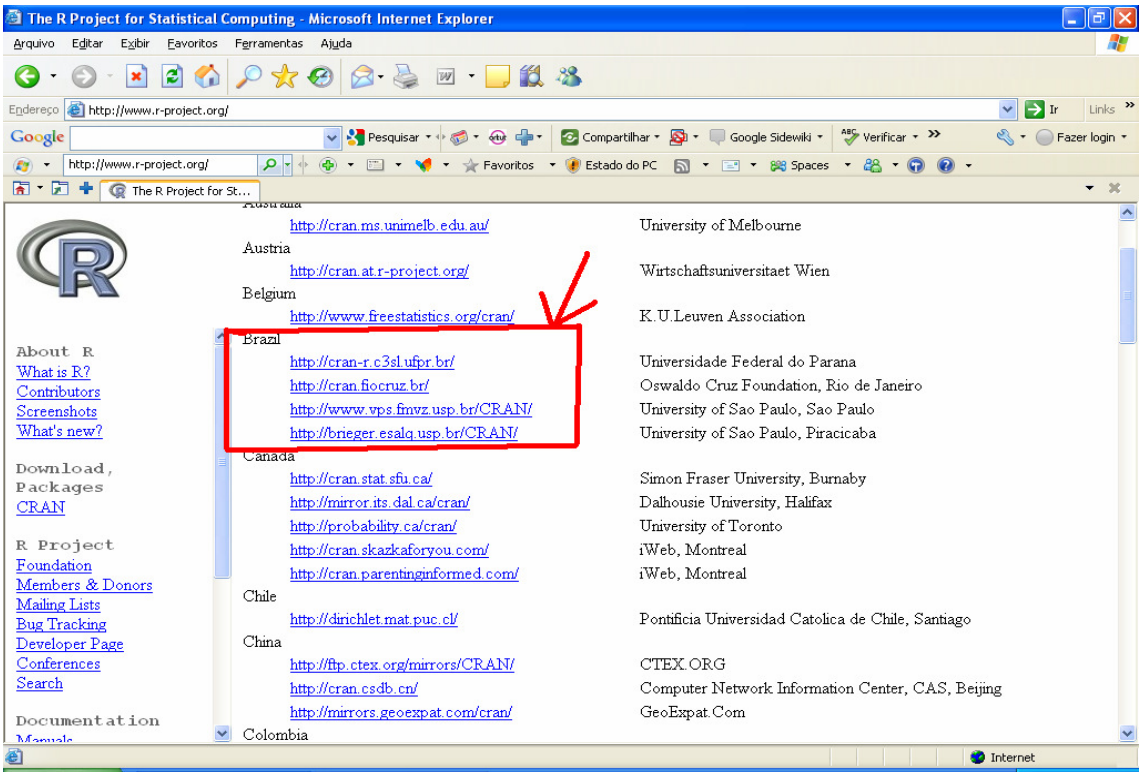
Nesta primeira aula trataremos da instalação e de alguns aspectos gerais do R importantes para a sua utilização.

1.1 - Como Instalar o R?

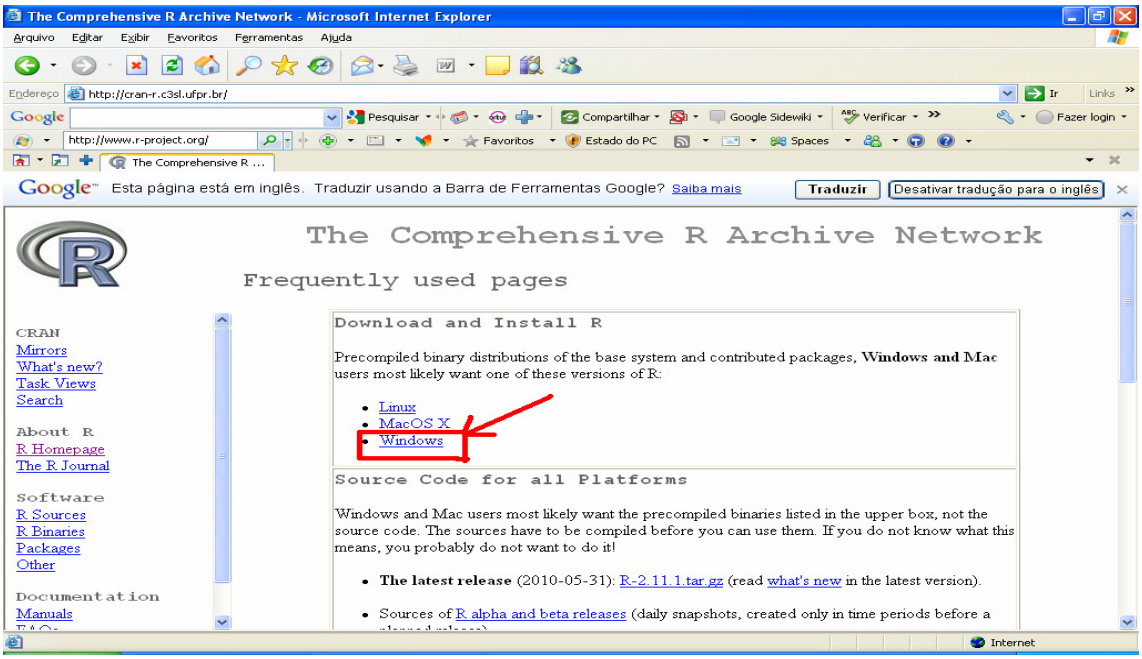
1º passo) Vá ao endereço www.r-project.org da página principal do projeto R e clique em download R, como mostrado na figura seguinte.



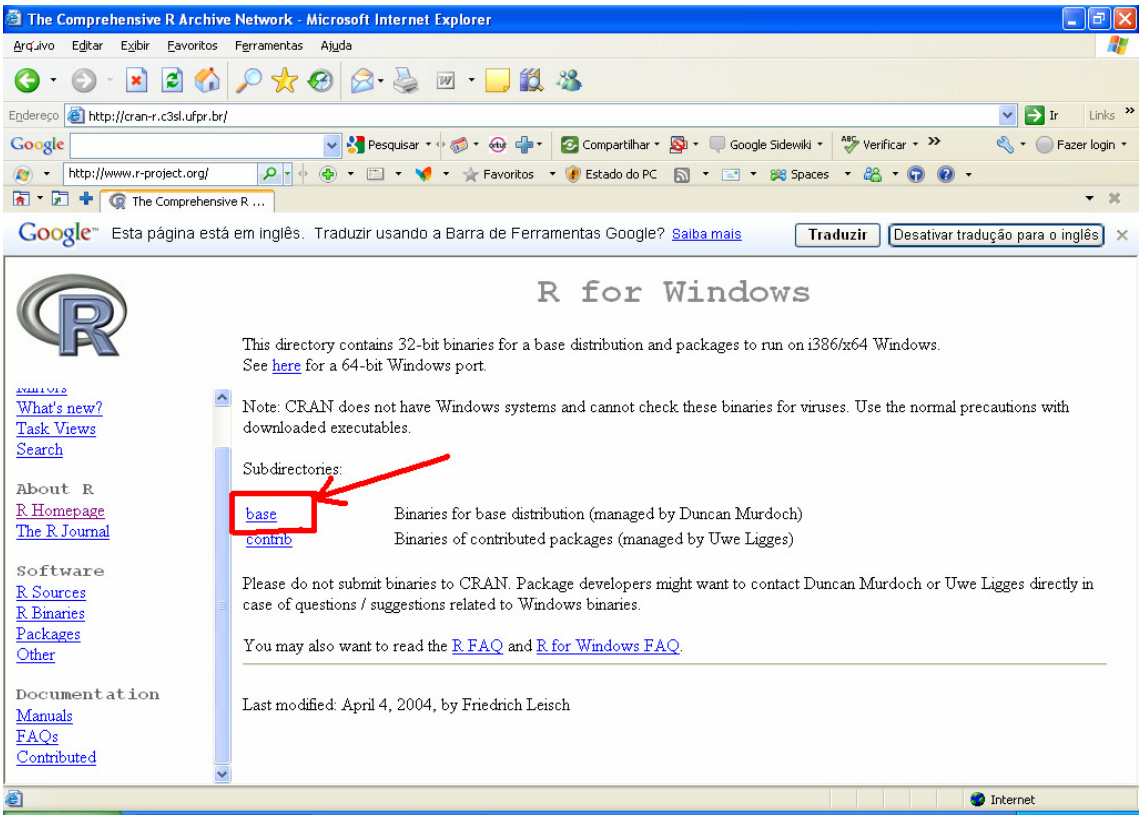
2º passo) Escolha o espelho de sua preferência, no Brasil existem 4.



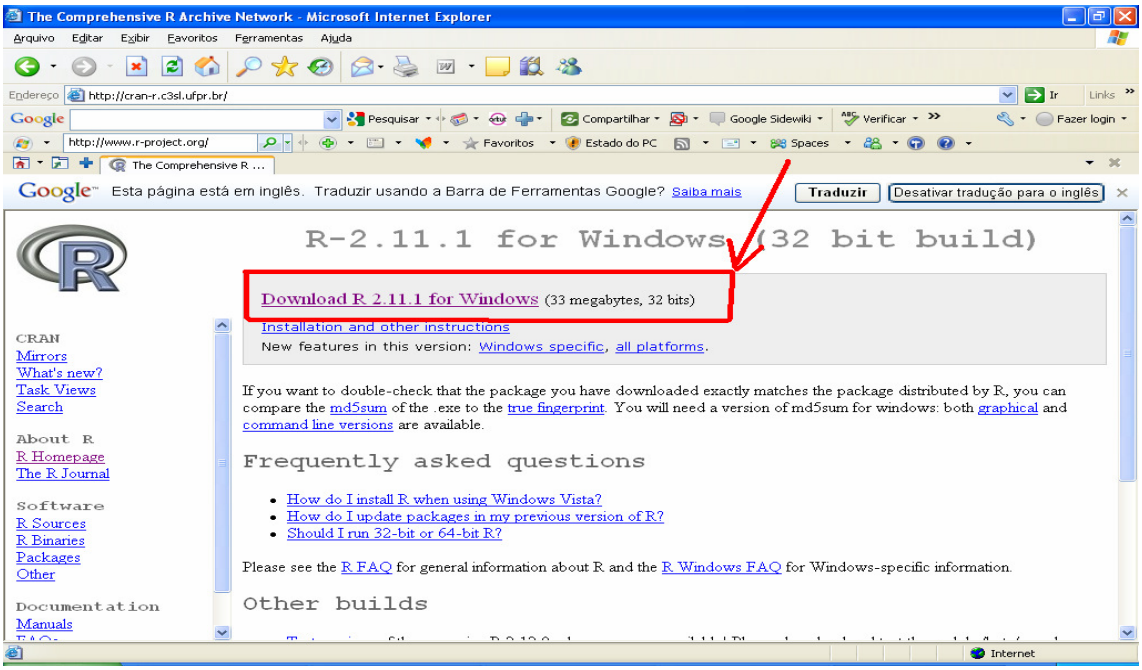
3º passo) Clique em um dos espelhos e abrirá uma nova tela. Se você utiliza plataforma Windows clique em **Windows**, caso contrário clique na plataforma conveniente.



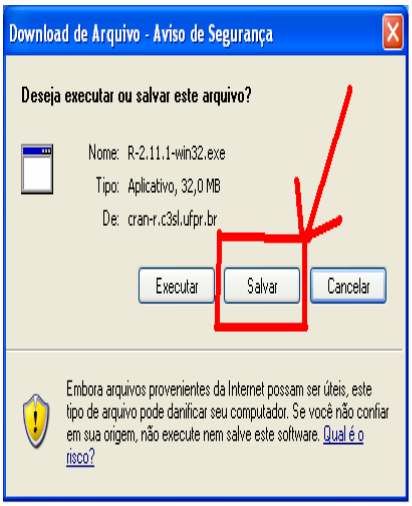
4º passo) Clique em base.

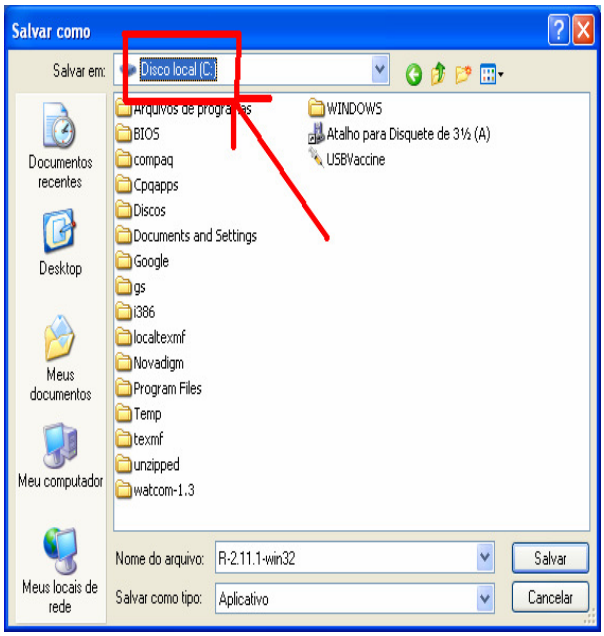


5º passo) Após clicar em base aparecerá a seguinte tela. Clique em Download R 2.11.1 for Windows



6º passo) Na nova janela clique na opção referente a salvar o arquivo e selecione a pasta onde o arquivo será salvo. Depois é só executá-lo.



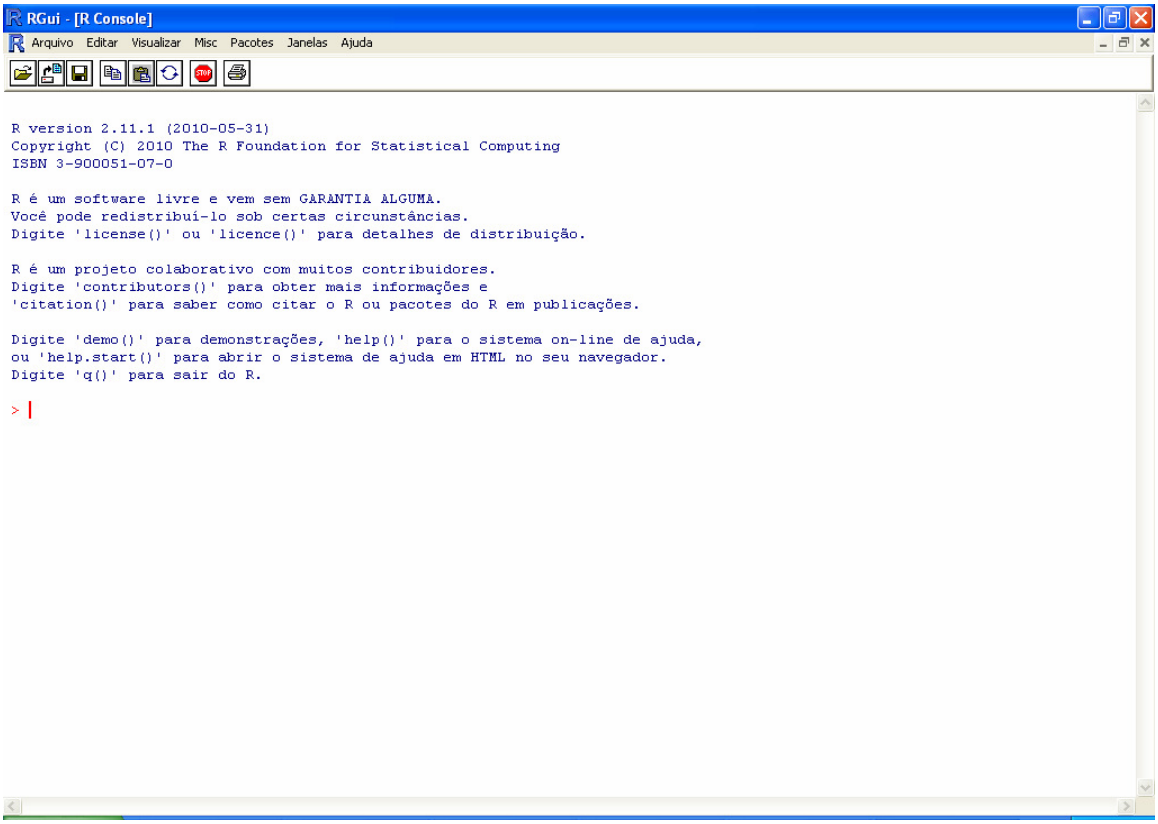


1.2 - Aspectos Gerais do R

1.2.1 - Iniciando o R

O R é uma linguagem interativa, ou seja, permite ao usuário enviar um comando por vez e receber o resultado. Para isso você precisa conhecer e digitar os comandos, pois ele não possui "menus" para clicar. Existem alguns módulos desenvolvidos para o R que permitem ao usuário escolher os comandos através de cliques, mas não trataremos deles neste texto.

Ao instalar o R ele criou um ícone na área de trabalho. Clique no ícone R e o programa será inicializado mostrando a seguinte tela:



O símbolo **>** indica a linha de comando ("prompt") na qual serão digitados os comandos para execução das análises. Os comandos aparecem escritos em vermelho e os seus resultados (as respostas) em azul. Por exemplo, para calcular a raiz quadrada de 16 digite o comando ***sqrt(16)*** na linha de comandos e tecle ENTER.

```
> sqrt (16)
[1] 4
```

Importante: Ao invés de digitar ***sqrt(16)*** na linha de comandos você pode copiar e colar o texto ***sqrt(16)*** (sem o sinal >) em frente ao sinal > desta linha.

Observe que a linha de comando está em vermelho e a linha de resposta em azul. Mais adiante você entenderá o símbolo [1]. Para executar outros comandos você deve proceder desta forma: digitar o comando e teclar ENTER.

Algumas vezes na linha de comando aparece o sinal +. Ele indica que o comando está incompleto e esperando o restante do mesmo. Você deve digitar o restante do comando em frente ao sinal + e teclar ENTER. Por exemplo, veja o que acontece ao executar o ***sqrt(16***

```
> sqrt(16
+ )
[1]  4
```

Caso você não queira completar a ação e sim interrompê-la, tecle em ***STOP*** no menu principal do R.

1.2.2 - Comentários no R

O sinal # (jogo da velha) é utilizado para inserir comentários. Significa que tudo que está depois do jogo da velha antes de dar o comando ENTER é comentário.

Exemplo:

```
> sqrt (16)                  # calcula a raiz quadrada de dezesseis
[1]  4
```

A frase ***"calcula a raiz quadrada de dezesseis"*** é um comentário.

1.2.3 - Uso de Maiúsculas e Minúsculas

Tecnicamente, R é uma linguagem de expressões de sintaxe muito simples, mas faz distinção entre maiúsculas e minúsculas, de modo que os caracteres **A** e **a** são entendidos como sendo símbolos diferentes. A maioria dos comandos é escrita em minúsculas. É recomendável não utilizar acentos e cedilha ao nomear objetos no R.

Exemplo:

```
> SQRT(16)
Erro: não foi possível encontrar a função "SQRT"
```

Não reconhece a função, pois a função correta é ***sqrt(16)*** com letras minúsculas.

1.2.4 - Separador de Casas Decimais

Para separar a parte inteira da parte decimal (separador de decimais) o R utiliza ponto.

Exemplo:

```
> sqrt (21)
[1] 4.582576
```

Entenda o resultado como 4,582576.

1.2.5 - Utilizando os Comandos de Ajuda no R

Durante a utilização do software é possível consultar a sintaxe de algum comando ou obter mais informações sobre determinada função. Para isso o R conta com o comando ***help.*** A sintaxe do comando é a seguinte:

> help (nome da função)

> ? nome da função

As duas sintaxes acima são equivalentes, ou seja, produzem o mesmo resultado. Por exemplo, para saber mais sobre a função ***sqrt.***

```
> help (sqrt)          # Obtendo ajuda sobre a função raiz quadrada
```

Ao executar o exemplo acima, uma interface do menu de ajuda será executada mostrando o tópico da função ***sqrt,*** que é a função matemática para o cálculo de raiz quadrada.

No menu principal, em ***Ajuda*****,** são disponíveis alguns manuais e comandos de ajuda. Para acessá-los clique em ***Ajuda-Funções R*** e escreva a função de interesse seguida de ENTER.

Os arquivos de ajuda do R são geralmente compostos de 9 tópicos.

1) Description – descrição sumária da função.

2) Usage – define como utilizar a função e quais são seus argumentos.

3) Arguments – indica o significado de cada argumento.

4) Details – indica detalhes ao quais se devem estar atendo ao usar a função.

5) Value – indica como é apresentado o resultado da função.

6) Note – notas sobre a função.

7) Authors – lista os autores da função.

8) References – referências bibliográficas sobre a função.

9) See Also – lista funções do R relacionadas.

10) Examples – Exemplos de uso da função.

Veja o arquivo de ajuda sobre a função ***mean.***

```
> help (mean)          # Obtendo ajuda sobre a função média
```

Observe que esta função faz parte do pacote base.

Agora que você já sabe como utilizar os comandos de ajuda, faça bom proveito deles.

Mas o que fazer quando não sabemos qual função do R faz a análise desejada?

Você pode usar o comando ***help.search( )*** ou simplesmente ??( ). Por exemplo, se você quiser informação sobre funções para calcular mediana (“median”)

```
> help.search(“median”)          # é o  mesmo que  >?? median
```

Você também pode buscar ajuda na internet, no site do R, com o comando ***RsiteSearch( )***. Para utilizar esta função você precisa estar conectado à internet. Por exemplo, para buscar ajuda sobre funções para construir tábuas de vida ***(“life table”)***

```
> RSiteSearch(“life table”)
```

1.2.6 - Como Citar o R ou os Pacotes do R em Suas Publicações e Trabalhos?

A função ***citation( )*** indica como citar o R.

```
> citation()

   To cite R in publications use:

     R Development Core Team (2010). R: A language and environment for
     statistical computing. R Foundation for Statistical Computing,
     Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

   A BibTeX entry for LaTeX users is

     @Manual{,
       title = {R: A Language and Environment for Statistical Computing},
       author = {{R Development Core Team}},
       organization = {R Foundation for Statistical Computing},
       address = {Vienna, Austria},
       year = {2010},
       note = {{ISBN} 3-900051-07-0},
       url = {http://www.R-project.org/},
     }

   We have invested a lot of time and effort in creating R, please cite
it when using it for data analysis. See also 'citation("pkgname")' for
   citing R packages.
```

Aula 2 – Objetos do R

O R opera com entidades chamadas de objetos. Objetos podem ser vetores, matrizes, funções ou estruturas mais gerais. Durante uma sessão do R objetos são criados e armazenados por nome.

Por exemplo, vamos criar um objeto de nome raiz no qual vamos armazena a raiz quadrada de 16, para isto faça:

```
> raiz <- sqrt(16)          # lê-se raiz recebe raiz quadrada de 16
> raiz                              # mostra o conteúdo de raiz
[1] 4
```

Ao invés do símbolo **<-** você pode usar o sinal de igualdade.

```
> raiz = sqrt(16)
```

Para ver todos os objetos criados na sua sessão de trabalho, use a função ***objects().***

```
> objects()
[1] "raiz"
```

Caso você queira remover um objeto use o comando ***rm*** (abreviação de remove). Por exemplo, para remover o objeto raiz faça:

```
> rm(raiz)                # remove o objeto raiz
```

Nesta aula abordaremos alguns dos objetos do R. Iniciaremos com os vetores.

2.1 - Vetores

Os vetores são os objetos mais importantes do R. Podem ser formados por números, nomes, elementos lógicos, desde que todos os elementos sejam do mesmo tipo.

2.1.1 -Criando Vetores

Podemos entrar com dados definindo vetores com o comando ***c ( )*** ("c" corresponde a *concatenate*) ou usando funções que criam vetores. Veja e experimente com os seguintes exemplos.

Para criar um vetor com as observações 23,0 21,8 26,1 27 , referentes as idades, em anos, de 4 pessoas, faça:

```
> idade <-  c (23 , 21.8 , 26.1 , 27)        # cria o vetor idade
> idade                    # mostra os elementos do vetor idade
[1] 23.0   21.8   26.1   27.0
```

Suponha que os elementos do vetor acima são as idades de Maria, Pedro, João e Rosa. Para criar um vetor com estes nomes:

```
> nome<-c("Maria","Pedro","João","Rosa")
> nome
[1] "Maria" "Pedro" "João" "Rosa"
```

Ao criar um vetor de nomes (caracteres), os elementos devem estar entre aspas duplas.

2.1.2 - Valores Faltantes no R

Vamos agora construir um vetor com o número de anos de estudo destas 4 pessoas. Sabemos que Maria, Pedro e João possuem respectivamente 10, 12 e 8 anos de estudo, mas esta informação não é conhecida para Rosa. Como fazer neste caso?

O R utiliza o símbolo ***NA*** ("not available") para observações faltantes.

```
> anosestudo<-c(10,12,18,NA)
> anosestudo
[1] 10 12 18 NA
```

2.1.3 - Nomeando os Objetos

Os nomes dos objetos devem começar com letras e podem conter letras, números e pontos. Ao nomear objetos evite o uso de cedilha e acentos e lembre-se também que o R faz a distinção entre letras maiúsculas e minúsculas. O R possui alguns nomes reservados, isto é, nomes que não podem ser utilizados pelo usuário para nomear objetos porque têm significado especial na linguagem R. Um deles é o nome ***NA*** que representa observações faltantes ou não disponíveis. Outros exemplos são: ***FALSE, .Inf, NaN, NULL, TRUE, break ,else, for, function, if, in, next, repeat, while.***

2.1.4 - Operações com Vetores

Vetores podem ser utilizados em operações aritméticas realizadas para cada elemento.

Considerando o vetor idade em anos, vamos obter as idades em meses.

```
> idademes<-idade*12
> idademes
[1] 276.0 261.6 313.2 324.0
```

A simbologia utilizada pelo R para operadores aritméticos elementares é apresentada na tabela seguinte:

Operador Aritmético Elementar	Simbologia
Soma	+
Subtração	-
Divisão	/
Multiplicação	*
Potência	^

Outras funções aritméticas	**Simbologia**	**Outras funções aritméticas**	**Simbologia**
Raiz quadrada	***sqrt( )***	Soma de todos os elementos	***sum( )***
logaritmo	***log( )***	Produto de todos os elementos	***prod( )***
exponencial	***exp( )***	Mínimo	***min( )***
Seno	***sin( )***	Máximo	***max()***
Cosseno	***cos( )***	Comprimento	***length( )***
tangente	***tan( )***	Média dos valores	***mean( )***
		Variância	***var( )***
		Desvio padrão	***sd( )***
		Mediana	***median( )***

Para calcular a distância de cada uma das idades do vetor idade em relação à idade média

```
> distidade<-idade - mean(idade)
> distidade
[1] -1.475  -2.675  1.625  2.525
```

Outras duas funções muito úteis são ***sort*** e ***rank.*** A função ***sort*** ordena os elementos do vetor e a função ***rank*** atribui posições aos elementos do vetor. Experimento estas 2 funções com o vetor idade: (23 , 21.8 , 26.1 , 27).

```
> sort(idade)         # ordena os valores em ordem crescente
[1] 21.8 23.0 26.1 27.0
> rank(idade)         # atribui posições aos elementos
[1] 2 1 3 4
```

Observe que ordenando as idades em ordem crescente, o primeiro valor de idade (23) ocupa a segunda posição no vetor ordenado de forma crescente, 21,8 ocupa a primeira posição e assim por diante.

Caso queira ordenar os elementos do vetor em ordem decrescente,

```
> sort(idade, decreasing = TRUE) # decreasing = FALSE é o padrão.
[1] 27.0 26.1 23.0 21.8
```

2.1.5 - Criando Vetores Formados por Seqüências Regulares

Os comandos ***seq*** e ***rep*** são muito úteis para criar vetores constituídos por seqüências regulares. Vamos ver alguns exemplos:

a) criar um vetor com números de 1 a 15 de nome seq1

```
> seq1<-1:15
> seq1
[1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
```

b) criando a seqüência no sentido inverso

```
> seq2<-15:1
> seq2
[1] 15 14 13 12 11 10  9  8  7  6  5  4  3  2  1
```

c) criando uma seqüência de 1 a 15 com intervalos de tamanho 2

```
> seq3<-seq(from=1, to =15, by=2)
> seq3
[1]  1  3  5  7  9 11 13 15
> # ou simplesmente
> seq3<-seq(1,15,2)
> seq3
[1]  1  3  5  7  9 11 13 15
```

d) criando a seqüência: 2, 2 , 2 , 3 , 3, 3, 4 , 4 , 4 , 5 , 5 , 5 , 6 , 6 , 6

```
> seq4<-rep(c(2,3,4,5,6),each=3) #each=3 indica que cada elemento deve ser repetido 3 vezes
> seq4
 [1] 2 2 2 3 3 3 4 4 4 5 5 5 6 6 6
```

e) criando a seqüência: 2 , 3 , 4 , 2 , 3 , 4 , 2 , 3 , 4 , 2 , 3 , 4 , 2 , 3 , 4

```
>seq5<-rep(c(2,3,4),times=5) #times=5 indica que a sequência 2.3.4 deve ser repetida 5 vezes
> seq5
 [1] 2 3 4 2 3 4 2 3 4 2 3 4 2 3 4
```

f) sequência com 5 elementos "X" e 5 elementos "Y".

```
> seq6<-rep(c("x","y"),each=5)
> seq6
 [1] "x" "x" "x" "x" "x" "y" "y" "y" "y" "y"
```

g) seqüência com os elementos x1, x2, x3, x4, x5, x6, x7, x8, x9 e x10. Para isto vamos usar o comando *paste*.

```
> seq7<-paste(c("X"),1:10, sep="") # sep="" indica que o nome X fica colado ao número
> seq7
[1] "X1"  "X2"  "X3"  "X4"  "X5"  "X6"  "X7"  "X8"  "X9"  "X10"
```

h) seqüência com os elementos aluno 1, aluno 2, aluno 3, aluno 4.

```
> seq8<-paste(c("aluno"),1:4,sep=" ") # sep=" " indica que o nome aluno é separado do número por um espaço em branco
> seq8
[1] "aluno 1" "aluno 2" "aluno 3" "aluno 4"
```

i) seqüência com os elementos aluno_1, aluno_2, aluno_3, aluno_4.

```
> seq9<-paste(c("aluno"),1:4,sep="_") # sep="_" indica que o nome aluno é
separado do número por um traço.
> seq9
[1] "aluno_1" "aluno_2" "aluno_3" "aluno_4"
```

2.1.6 - Vetores Lógicos

No R podemos trabalhar com vetores lógicos. Vetores lógicos são formados pelos elementos ***TRUE, FALSE e NA***. Veja abaixo um exemplo de um vetor lógico.

Considere o vetor idade com os elementos: 23 21,8 26,1 27.

```
> id23<-idade<23     # atribui TRUE se idade <23 e FALSE caso contrário
> id23
[1] FALSE TRUE FALSE FALSE
```

Observe que o primeiro elemento do vetor id23 é ***FALSE***, pois o primeiro elemento do vetor idade é maior ou igual a 23, isto é, não satisfaz a condição lógica idade<23.

Operadores lógicos são muito úteis na manipulação de dados, como veremos adiante. No quadro abaixo é apresentada à simbologia para operadores lógicos usada pelo R.

Operação lógica	**Operador**	**Operação lógica**	**Operador**
menor que	<	maior ou igual a	>=
menor ou igual a	<=	Igual a	= =
maior que	>	Diferente de	!=

Se ***c1*** e ***c2*** são 2 expressões lógicas ***c1 & c2*** é a sua interseção(***"and"***), ***c1|c2*** a sua união (***"or"***) e ***!c1*** é a negação de ***c1***. Veremos alguns exemplos na próxima seção.

2.1.7 - Indexando Vetores, Selecionando e Modificando Conjuntos de Dados

Os elementos de um vetor são indexados por números variando de 1 até o comprimento do vetor. Por exemplo, em um vetor com 4 elementos estes índices variam de 1 a 4. Subconjuntos ou elementos de um vetor podem ser selecionados indicando entre colchetes os elementos a serem selecionados. Esta indicação pode ser feita através de condições lógicas ou especificando os índices dos elementos a serem selecionados. Vamos ver alguns casos, utilizando os dados do exemplo abaixo:

Exemplo: Utilizando o conjunto de dados abaixo, referente ao peso e altura de 15 mulheres, crie os vetores: altura e peso.

Altura (cm)	Peso (kg)
159	62
161	67
148	60
160	61
158	62
164	65
164	66
153	63
157	64
163	69
159	68
156	59
149	70
157	58
162	71

```
> altura<- c(159, 161, 148, 160, 158, 164, 164, 153, 157, 163, 159, 156,
149, 157, 162)
> peso <- c(62, 67, 60, 61, 62, 65, 66, 63, 64, 69, 68, 59, 70,58, 71)
```

1) Selecionando o primeiro elemento do vetor altura.

```
> altura[1]
[1] 159
```

2) Selecionando os 4 primeiros elementos

```
> altura[1:4]
[1] 159 161 148 160
>  altura[c(1:4)]      #  o  vetor  c(1:4)  indica  os  elementos  a  serem
selecionados
[1] 159 161 148 160
```

3) Selecionando os elementos de ordem ímpar:

```
> altura[seq(1, 15, 2)]
[1] 159 148 158 164 157 159 149 162
```

4) Selecionando as alturas das mulheres menores do que 160 cm

```
> altura[altura<160]        # aqui estamos utilizando uma condição lógica
[1] 159 148 158 153 157 159 156 149 157
```

5) Selecionando as alturas das mulheres com peso abaixo de 65 kg

```
> altura[peso<65]
[1] 159 148 160 158 153 157 156 157
```

6) Selecionando as alturas das mulheres com peso abaixo de 65 kg e altura abaixo de 160 cm.

```
> altura[peso<65 & altura <160]       # interseção de 2 condições lógicas
[1] 159 148 160 158 153 157 156 157
```

7) Selecionado as alturas das mulheres que não tenham peso abaixo de 65 kg e altura abaixo de 160 cm

```
> altura[!(peso<65 & altura <160)]
[1] 161 164 164 163 159 149 162
```

8) Selecionando os 14 primeiros elementos do vetor altura. Podemos fazer isto de duas formas: indicando os elementos as serem incluídos ou aqueles a serem excluídos:

```
> altura[1:14]      # entre colchetes está a indicação dos elementos a
serem selecionados
 [1] 159 161 148 160 158 164 164 153 157 163 159 156 149 157
> altura[-15] # o sinal - indica que o décimo quinto elemento do vetor
deve ser excluído
 [1] 159 161 148 160 158 164 164 153 157 163 159 156 149 157
```

Vamos considerar agora um vetor de nomes:

```
> frutas<-rep(c("laranja","banana","limão","jaboticaba"),times=5)
> frutas
[1] "laranja"    "banana"     "limão"      "jaboticaba" "laranja"
[6] "banana"     "limão"      "jaboticaba" "laranja"    "banana"
[11] "limão"      "jaboticaba" "laranja"    "banana"     "limão"
[16] "jaboticaba" "laranja"    "banana"     "limão"      "jaboticaba"
```

9) Selecionando os elementos do vetor correspondentes a frutas cítricas

```
> citricas<-frutas[frutas=="laranja" | frutas=="limão"] # | é o simbolo
para união (“or”)
> citricas
 [1] "laranja"  "limão"  "laranja"  "limão"  "laranja" "limão"  "laranja"
 [8] "limão"   "laranja" "limão"
```

2.1.8 - Modificando e Incluindo Elementos em um Vetor

O vetor altura possui 15 alturas. Suponha que a altura da primeira mulher era 169 ao invés de 159. Como podemos substituir este elemento pelo valor correto?

```
> altura[1]<-169
> altura
 [1] 169 161 148 160 158 164 164 153 157 163 159 156 149 157 162
```

Suponha que temos a informação sobre 2 novas mulheres com alturas iguais a 170 e 175 cm e queremos construir um novo vetor de alturas incluindo estes elementos. Vamos fazer isto de 2 maneiras diferentes.

```
>altura1<-c(altura,170,175) #o vetor altura1 é formado pelo vetor altura
e 2 novas observações.
> altura1
 [1] 169 161 148 160 158 164 164 153 157 163 159 156 149 157 162 170 175
>  altura[c(16,17)]<-c(170,175)  #acrescenta  2  novas  observações  nas
posições 16 e 17
> altura
 [1] 169 161 148 160 158 164 164 153 157 163 159 156 149 157 162 170 175
```

Além dos vetores o R trabalha com outros tipos de objetos: Matrizes: Fatores, listas, data.frames, arranjos e funções. Vamos tratar aqui somente de fatores, matrizes e data.frames. Os interessados em saber mais sobre estes e outros objetos podem consultar os manuais do R e a bibliografia indicada no final da apostila.

2.2 - Fator

Na seção anterior vimos como criar um vetor de caracteres. O vetor frutas guarda a informação da variável frutas. Diferente do vetor de caracteres o objeto ***factor*** possui outros atributos além dos dados.

```
> frutas<-rep(c("laranja","banana","limão","jaboticaba"),times=5)
> frutas
 [1] "laranja"    "banana"     "limão"      "jaboticaba" "laranja"
 [6] "banana"     "limão"      "jaboticaba" "laranja"    "banana"
[11] "limão"      "jaboticaba" "laranja"    "banana"     "limão"
[16] "jaboticaba" "laranja"    "banana"     "limão"      "jaboticaba"
```

Transformando o vetor frutas num fator

```
> frutas <-factor(frutas)
> frutas
 [1] laranja    banana     limão      jaboticaba laranja    banana
 [7] limão      jaboticaba laranja    banana     limão      jaboticaba
[13] laranja    banana     limão      jaboticaba laranja    banana
[19] limão      jaboticaba
Levels: banana jaboticaba laranja limão
```

Observe que além dos dados, o fator frutas possui agora informação sobre os níveis do fator (categorias da variável frutas: banana jaboticaba laranja limão). Suponha agora que queiramos criar um novo fator com as categorias sim para cítricas e não para não cítricas.

```
> citricas<-frutas        # copia o objeto frutas para o objeto citricas
> levels(citricas)        # mostra os objetos de cítricas
 [1] "banana"  "jaboticaba"  "laranja"  "limão"
> levels(citricas)<-c("nao","nao","sim","sim")   #modifica os níveis dos
fatores
> cítricas          # mostra o fator modificado
 [1] sim nao sim nao sim nao sim nao sim nao sim nao sim nao sim nao sim
nao sim
[20] nao
Levels: nao sim
```

Observação: em muitas análises estatísticas as variáveis precisam ser declaradas como fatores.

2.3 - Matriz

Uma matriz é uma coleção de vetores de mesmo comprimento organizados um do lado do outro. No R, todos os elementos de um vetor e também de uma matriz devem ser do mesmo tipo, isto é, devem ser todos numéricos ou devem ser todos caracteres.

Suponha que um teste diagnóstico foi aplicado a 20 pacientes doentes e a 30 não doentes. Dos doentes 18 apresentaram resultado positivo no teste e dos não doentes 26 apresentaram resultados negativos. Estes dados podem ser organizados em uma matriz

	Positivo	Negativo
Doente	18	02
Não Doente	04	26

Para construir uma matriz com as freqüências dadas na tabela, vamos usar a função ***matrix.*** Esta função organiza os elementos de um vetor numa matriz com a dimensão desejada. A dimensão da matriz é especificada informando o número de colunas. Caso você não especifique o contrário, os elementos do vetor são organizados na matriz no sentido das colunas.

Criando um vetor de nome **freq** com as freqüências da tabela. .

```
> freq<-c(18, 02, 04, 26)
```

Criando uma matriz 2 x 2 (2 linhas x 2 colunas), de nome M, com os elementos do vetor freq.

```
> M <-matrix(freq, ncol=2)        # ncol é o número de colunas da matriz
> M
     [,1] [,2]
[1,]   18    4
[2,]    2   26
```

Observe que o R colocou os 2 primeiros elementos do vetor na primeira coluna e os outros 2 na segunda. Para preencher a matriz no sentido das linhas, você deve fazer como segue:

```
> M <-matrix(freq, ncol = 2, byrow = TRUE)
> # byrow = TRUE indica que a matriz deve ser preenchida no sentido das
linhas.
> M
     [,1] [,2]
[1,]   18    2
[2,]    4   26
```

Para atribuir nomes às colunas e linhas da matriz, devemos usar o subcomando ***dimnames***. Fazemos ***dimnames*** igual a uma lista de tamanho 2 contendo 2 vetores, o primeiro com o nome das linhas e o segundo com os nomes das colunas da matriz. Lista é outro tipo de objeto do R. Uma lista é um objeto mais flexível, podendo ser formado por vários outros objetos. No caso acima a lista é formada por 2 vetores.

```
> M<-matrix(freq, ncol = 2, byrow = TRUE, dimnames = list(c("doente","não
doente"), c("positivo","negativo")))
> M
               positivo negativo
doente               18        2
não doente            4        26
```

Os elementos da matriz também são indexados. Para o exemplo, M[i,j] retorna o elemento na linha i e coluna j.

```
> M[1,1]
[1] 18
> M[2,2]
[1] 26
```

Se omitirmos um dos índices i ou j em M[i,j], o que acontece?

```
> M[1,]         #retorna a primeira linha da matriz
positivo negativo
      18        2
> M[,1]           #retorna a primeira coluna da matriz
    doente não doente
        18          4
```

Vamos agora responder algumas perguntas sobre a matriz M.

Qual a dimensão da matriz?

```
> dim(M)
[1] 2 2
```

O primeiro número refere-se ao número de linhas e o segundo número ao número de colunas. Portanto a matriz M tem 2 linhas e 2 colunas.

Quantos elementos têm a matriz?

```
> length(M)
[1] 4
```

Qual a soma dos elementos da matriz?

```
> sum(M)
[1] 50
```

Qual a soma das linhas da matriz, isto é qual o número total de doentes e não doentes?

```
> rowSums(M)
doente não doente
    20       30
```

Qual a soma das colunas da matriz, isto é qual o número total de resultados positivos e negativos?

```
> colSums(M)
positivo negativo
      22       28
```

Qual a proporção de resultados positivos entre os doentes? E de resultados negativos entre os não doentes? Vamos chamar estas proporções de S e E, respectivamente.

```
> S<-M[1,1]/sum(M[1,])
> S
[1] 0.9
> E<-M[2,2]/sum(M[2,])
> E
[1] 0.866666
```

2.4 - Data.Frames

Data.Frames são muito parecidos com matrizes, eles possuem linhas e colunas, portanto tem duas dimensões. Entretanto, diferentemente de matrizes, cada coluna pode armazenar elementos de diferentes tipos. Por exemplo: a primeira coluna pode ser numérica enquanto a segunda pode ser constituída de caracteres.

Data.Frames é a melhor forma de se armazenar dados onde cada linha corresponde a uma unidade, indivíduo, ou pessoa, e cada coluna representa uma medida realizada em cada unidade, isto é, uma variável.

O R possui vários conjuntos de dados, muitos deles organizados sob a forma de data.frames. Para ver os conjuntos de dados disponíveis, digite o comando seguinte:

```
> data()
```

Observe que há um conjunto de dados chamado ***women***, com peso e altura de 15 mulheres. Para ver este conjunto de dados, vamos carregá-lo, fazendo como segue:

```
> data(women)          # carrega o conjunto de dados
> women                # mostra o conjunto de dados
   height weight
1      58    115
2      59    117
3      60    120
4      61    123
5      62    126
6      63    129
7      64    132
8      65    135
9      66    139
10     67    142
11     68    146
12     69    150
13     70    154
14     71    159
15     72    164
```

O objeto ***women*** é um data.frame. Os dados de cada linha são de um mesmo indivíduo. Na primeira coluna temos a variável altura e na segunda a variável peso.

Vamos agora ver como podemos criar data.frame. Considere os dados abaixo relativos ao salário, estado civil e idade de 5 pessoas.

Pessoa	Idade (anos)	Salário (R$)	Estado Civil
1	24	2000	Solteiro
2	30	3000	Casado
3	54	1700	Casado
4	31	500	Solteiro
5	19	550	Casado

Nosso data.frame de nome dados será formado pelas variáveis idade, salário e estado civil.

```
dados<-
data.frame(idade=c(24,30,54,31,19),salário=c(2000,3000,1700,500,550),
estadocivil= c("solteiro","casado","casado","solteiro","casado"))
>dados
  idade salario estadocivil
1    24    2000    solteiro
2    30    3000      casado
3    54    1700      casado
4    31     500    solteiro
5    19     550      casado
```

Na aula 3 veremos outras formas de entrada de dados no R.

Assim com a matriz, os elementos do data.frame são indexados.

```
> dados[1,]           # mostra os dados da primeira linha
  idade salário estadocivil
1    24    2000    solteiro
> dados[,1]          # mostra os dados da primeira coluna
[1] 24 30 54 31 19
```

Para referirmos à variável idade, ao invés de especificar a coluna do data.frame onde esta a variável idade, como no exemplo acima, podemos fazer:

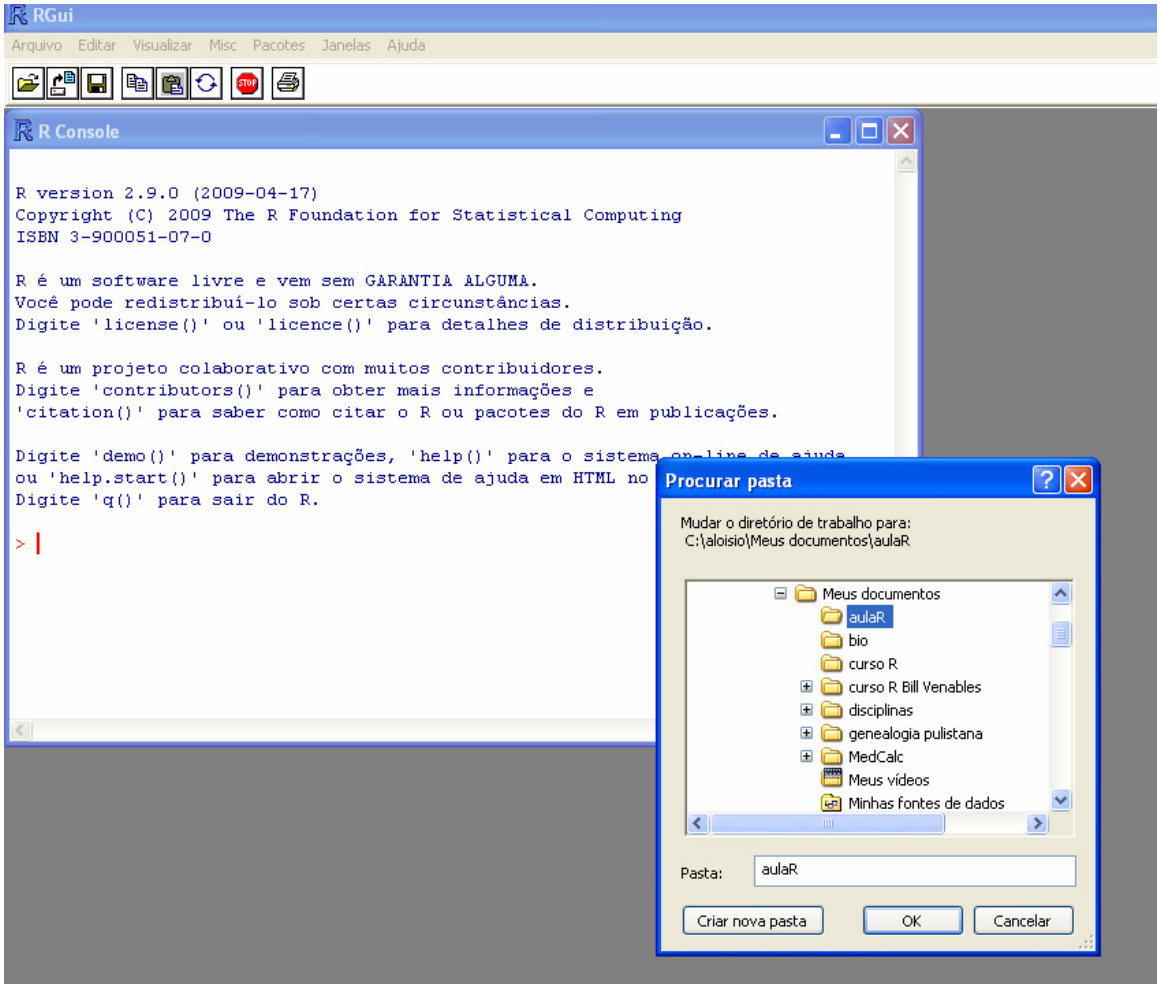
```
> dados$idade
[1] 24 30 54 31 19
> dados$salário
[1] 2000 3000 1700  500  550
> dados$estadocivil
[1] solteiro  casado  casado  solteiro  casado
Levels: casado  solteiro
> #Obtendo a idade média
> mean(dados$idade)
[1] 31.6
```

Aula 3 – Armazenando os Resultados e o Histórico de Comandos de uma Sessão de Trabalho

Agora que você já possui alguma familiaridade com o R, vamos tratar de um aspecto muito importante: como armazenar os objetos criados durante a sessão (área de trabalho), o histórico de comandos executados (histórico) e os resultados da sua sessão de trabalho ("output")?

Utilizando o **Windows Explorer** crie uma pasta (diretório) onde você irá salvar os arquivos com os históricos de comandos executados e os resultados obtidos. Por exemplo, no diretório meus documentos crie a pasta aulaR.

Você pode salvar ou ler arquivos em/de qualquer pasta do seu computador. Para facilitar sua vida você pode escolher o diretório onde irá guardar os arquivos com resultados de suas análises ou de onde irá ler dados de arquivos externos. Para isto, vá a arquivos no menu principal e clique em ***mudar diretório***. Selecione sua pasta de trabalho (aulaR), como indicado abaixo



Para ver como salvar os resultados e o histórico de comandos da sua sessão, considere o exemplo:

```
> # crie um vetor de nome dados com as observações: 17, 20, 35, 60, 80,
20, 59, 50, 43, 30
>dados <- c(17, 20, 35, 60, 80, 20, 59, 50, 43, 30)
>dados
>#  calcule a media destes dados
>mean (dados)
```

Agora antes de encerrar sua sessão de trabalho salve os resultados e o histórico de comandos.

3.1 - Salvando um Arquivo

3.1.1 - Salvando a Área de Trabalho

Vá ao menu principal, clique em arquivo e depois em salvar área de trabalho. Informe o nome do arquivo, neste caso aula1 com extensão RData (aula1.RData). Observe que na linha de comandos aparece a sintaxe do comando executado

```
> save.image("C: :\\ Meus documentos aula1.RData")
```

3.1.2 - Salvando o Histórico de Comandos

Vá ao menu principal, clique em arquivo e depois em salvar histórico. Informe o nome do arquivo, neste caso aula1. Observe que na linha de comandos aparece a sintaxe do comando executado

```
> save.image("C:\\ Meus documentos\\aulaR\\aula1")
```

3.1.3 - Salvando o "Output"

Vá ao menu principal, clique em arquivo e depois em salvar em arquivo. Informe o nome do arquivo, neste caso resultados. Observe que na linha de comandos aparece a sintaxe do comando executado

Uma vez que você salvou os resultados e o histórico de comandos finalize o R clicando em arquivo e depois em sair ou digite ***q( )*** na tela de comandos seguido de ENTER.

Suponha que você pretenda continuar sua sessão de trabalho, utilizando os objetos armazenados no arquivo aula1. Para isto faça como indicado abaixo:

1) Inicialize o R
2) Mude o diretório para a pasta aulaR

3) Carregue o arquivo aula1.RData. Para isto clique em arquivo no menu principal e depois em carregar área de trabalho. Selecione o arquivo aula1.RData. Para ver os objetos presentes em aula1.RData digite ***objects( ).***

```
> objects( )
[1] "dados"
> dados
 [1] 17 20 35 60 80 20 59 50 43 30
```

Como era esperado há apenas um objeto de nome dados, criado na sessão anterior.

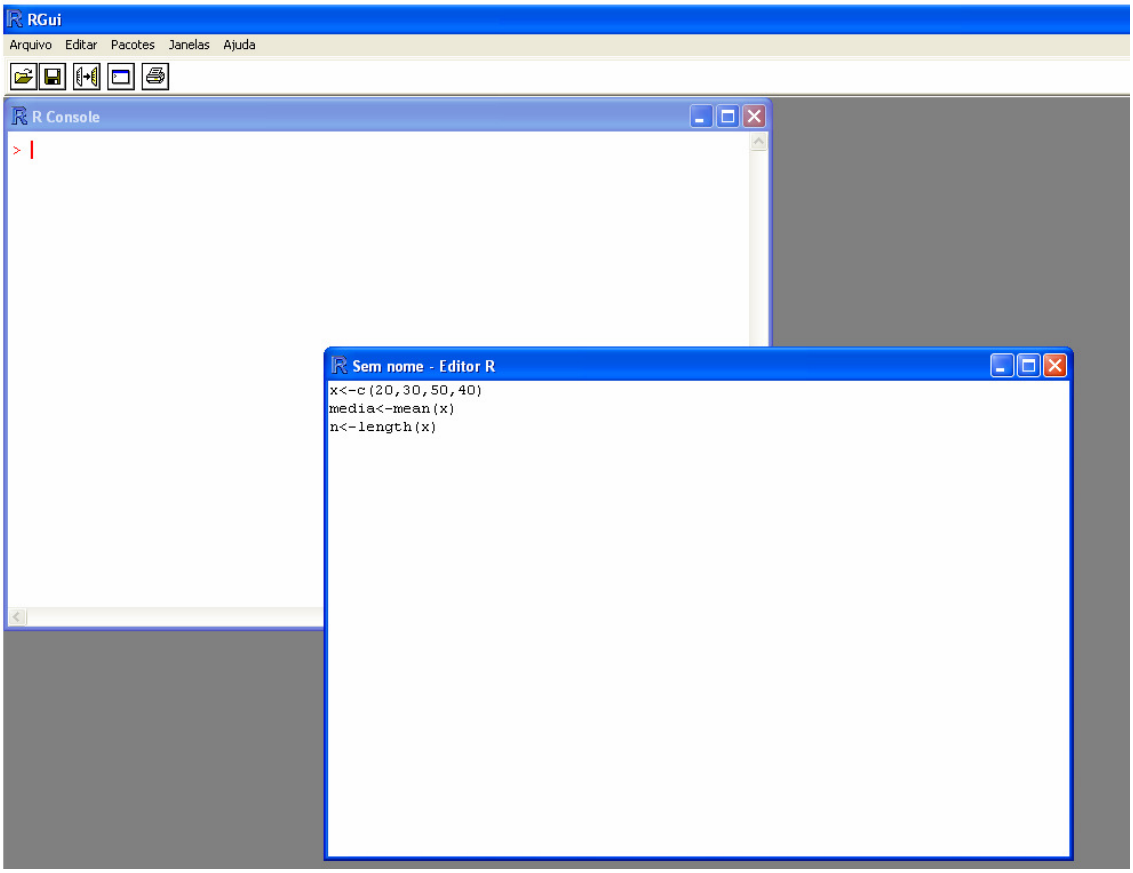
Você também pode carregar o histórico dos comandos da última sessão salvos em aula1. Para isto, clique em arquivo e depois em carregar histórico. Selecione o arquivo aula1. Para ver os comandos executados anteriormente clique no comando ↑ do teclado. Desta forma aparecerão na tela os comandos executados.

Quando você terminar suas análise provavelmente irá utilizar os resultados em um relatório. Use um editor de textos, por exemplo, o WORD, para ler o arquivo resultados.

3.2 - Executando um Script - Arquivo Texto com Comandos

Uma maneira de otimizar o uso do R e poupar tempo é criar uma arquivo texto e depois executá-lo no R. Para isto vamos utilizar o ***editor R,*** o editor de textos do R. O arquivo texto criado no R editor é chamado script. Para criar um script no R clique em arquivo e depois em novo script. O R irá abrir uma janela cujo nome é sem nome – Editor R. Nesta janela digite os comandos abaixo como mostrado na figura seguinte:

```
x<-c(20,30,50,40)
x
media<-mean(x)
media
n<-length(x)
>n
```



Depois, com o cursor ativo na janela Editor R clique em ***salvar como*** informando o nome do arquivo que neste caso será teste. Uma vez salvo o arquivo observe que no alto da janela aparece o nome do arquivo seguido de Editor R. Deste modo você criou um arquivo com os comandos a serem executados. Para executá-los, faça como segue: com o curso ativo na janela Editor R selecione todas as linhas a serem executadas. Por exemplo selecione a primeira linha. Depois clique no botão direito do mouse e depois em executar linha ou seleção (ou use o atalho Ctrl+R). Para executar todas as linhas do arquivo, clique no botão direito do mouse e depois na opção selecionar tudo. Todo o texto será marcado. Torne a clicar no botão direito do mouse de depois em executar linha ou seleção.

Daqui em diante sempre use o script do R. Com ele é fácil refazer análises ou alterar comandos. No script você também pode inserir observações sobre o que foi feito, usando # para indicar a presença de um comentário.

Quando for fechar a janela do script o R perguntará se você deseja salvar o arquivo. Diga que sim, indique o nome e endereço onde o arquivo deverá ser salvo.

Aula 4 - Entrada de dados no R

Há diferentes maneiras de entrada de dados no R. O formato mais adequado depende do tamanho do conjunto de dados, se os dados já existem em outro formato para serem importados ou se serão digitados diretamente no R.

4.1 - Entrando com Dados Diretamento no R – via teclado

Como visto na aula 2, podemos entrar com dados definindo vetores com o comando c( ) ("c" corresponde a *concatenate*). Connsidere os dados abaixo referente ao peso e altura de 15 mulheres. Vamos criar 2 vetores de nomes `peso1` e `altura1` contendo estas observções.

Altura (cm)	Peso (kg)
159	62
161	67
148	60
160	61
158	62
164	65
164	66
153	63
157	64
163	69
159	68
156	59
149	70
157	58
162	71

```
>altura1<- c(159,161,148,160,158,164,164,153,157,163,159,156,149,157,162)
> altura1
 [1] 159 161 148 160 158 164 164 153 157 163 159 156 149 157 162
> peso1 <- c(62,67,60,61,62,65,66,63,64,69,68,59,70,58,71)
> peso1
 [1] 62 67 60 61 62 65 66 63 64 69 68 59 70 58 71
```

Esta forma de entrada de dados é conveniente quando se tem um pequeno número de dados.

4.1.1 - Utilizando o Comando *scan*

Outra maneira de entrar com dados na forma de vetor é utilizar a função ***scan***. Esta função coloca o R em modo *prompt* onde o usuário deve digitar cada dado seguido da tecla ENTER. Para encerrar a entrada de dados basta digitar ENTER duas vezes consecutivas. Veja como isto funciona para os dados de peso e altura de mulheres dados anteriormente.

```
> altura <- scan ( )        # carrega a função scan
1: 159
2: 161
3: 148
4: 160
5: 158
6: 164
7: 164
8: 153
9: 157
10: 163
11: 159
12: 156
13: 149
14: 157
15: 162
16:
Read 15 items
```

Também pode-se criar um vetor com nome peso1 e logo após criar um data.frame de nome dados com as variáveis peso e altura, fazendo:

```
> dados<-data.frame(peso,altura)
> dados
   peso altura
1     62    159
2     67    161
3     60    148
4     61    160
5     62    158
6     65    164
7     66    164
8     63    153
9     64    157
10    69    163
11    68    159
12    59    156
13    70    149
14    58    157
15    71    162
```

Caso o vetor a ser criado seja de caracteres faça o argumento ***what=””.*** Por exemplo, para criar um vetor com os nomes: Ana, Maria e Manuela

```
> nomes<-scan(,what="")
1: Ana
2: Maria
3: Manuela
4:
Read 3 items
> nomes
[1] "Ana"     "Maria"    "Manuela"
```

O formato ***scan*** é mais ágil que o anterior e é conveniente para digitar vetores longos. Esta função pode também ser usada para ler dados de um arquivo ou conexão, aceitando inclusive endereços de URL's (endereços da web).

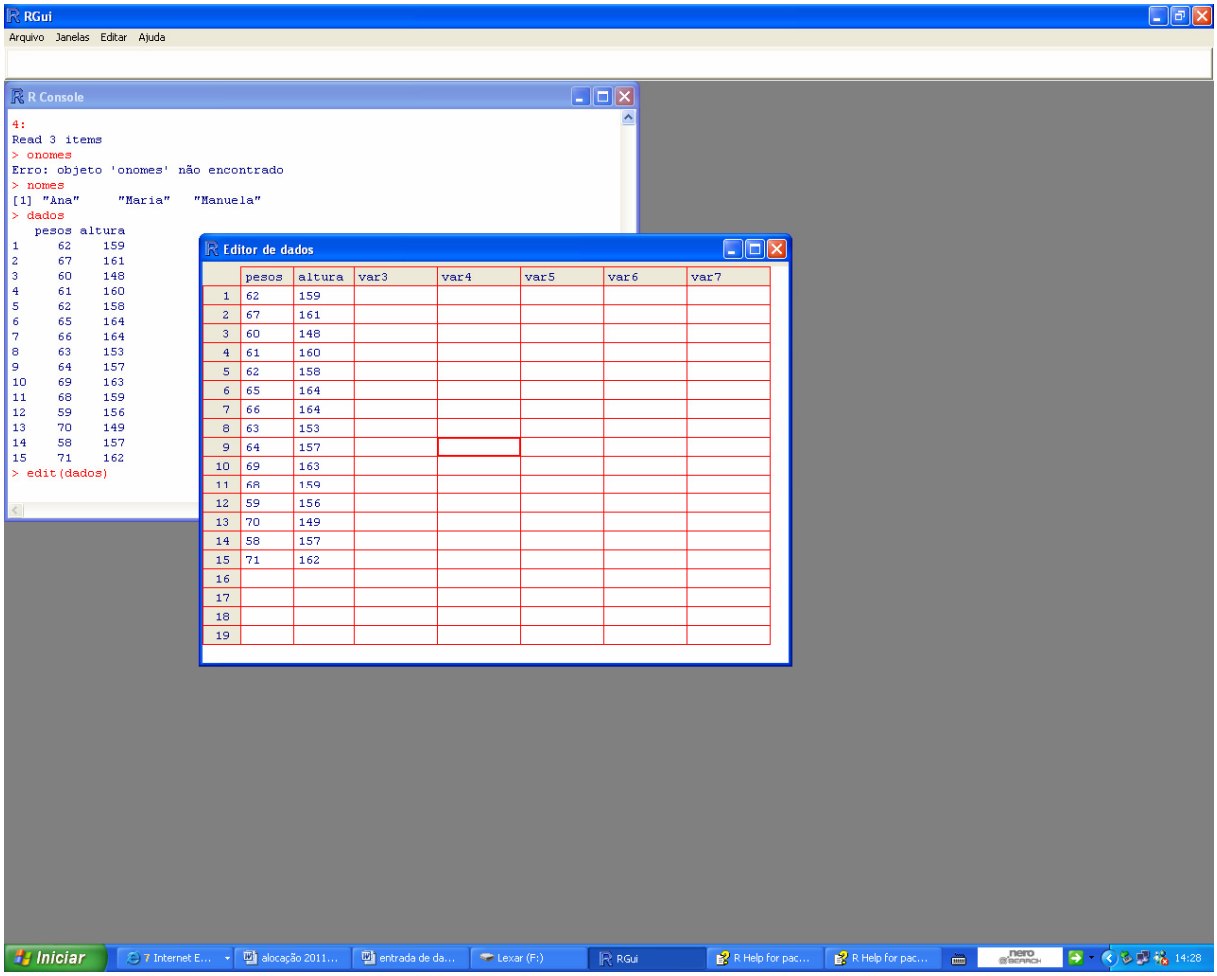
4.1.2 – Criando Data.frames com o Comando *edit*

O comando ***edit*** pode ser usado quando queremos editar ou modificar um objeto já existente. Por exemplo, suponha que queiramos introduzir uma terceira variável, idade, no data.frame dados. As idades são dadas por: 20, 34, 25, 25, 36, 49, 67, 43, 21, 30, 54, 60, 23, 45, 15. Como podemos fazer isto? Usando o comanado ***edit( ).***

Digite o comando abaixo

```
> dados <-edit(dados)
```

Observe que o R abriu uma janela com os dados chamada EDITOR de dados, como mostrado abaixo.



Na terceira coluna desta janela esceva o nome da variável idade. Selecione o tipo de variável: numérica ou character. No nosso caso, numérica. Depois digite as idades nas

células seguintes. Após digitar a última idade, clique no símbolo X no canto superior direito da janela.

Uma vez que você entrou com os dados relativos às idades, seu data.frame. modificado será

```
> dados
   pesos altura idade
1     62    159    20
2     67    161    34
3     60    148    25
4     61    160    25
5     62    158    36
6     65    164    49
7     66    164    67
8     63    153    43
9     64    157    20
10    69    163    31
11    68    159    54
12    59    156    60
13    70    149    23
14    58    157    45
15    71    162    15
```

A função ***edit*** pode também ser utilizada para entrada de dados na forma de um data.frame. Considere os dados contendo nomes e notas de 5 alunos

Nomes	Pedro	Maria	João	José	Francisco
Notas	30	25	17	37	29

Para criar um data.frame de nome notas com os dados acima faça:

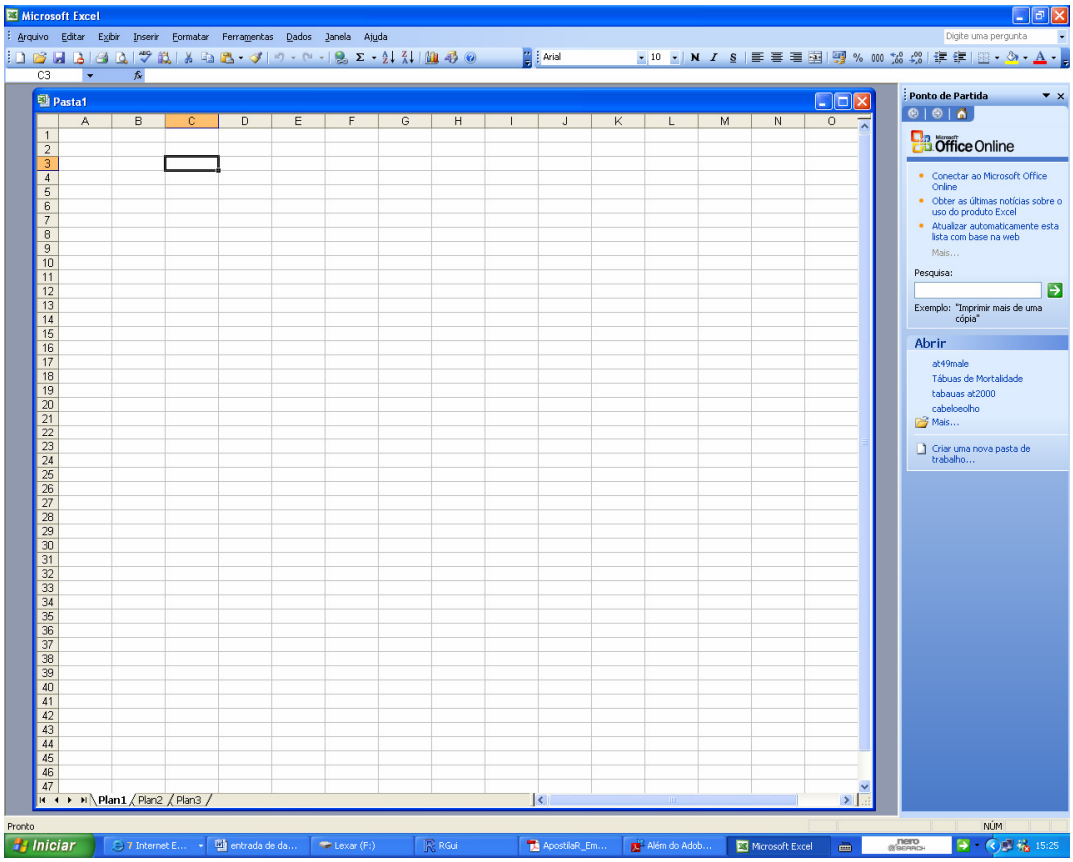
```
> notas<-edit(data.frame())
```

Para entrar com os dados proceda como no exemplo anterior.

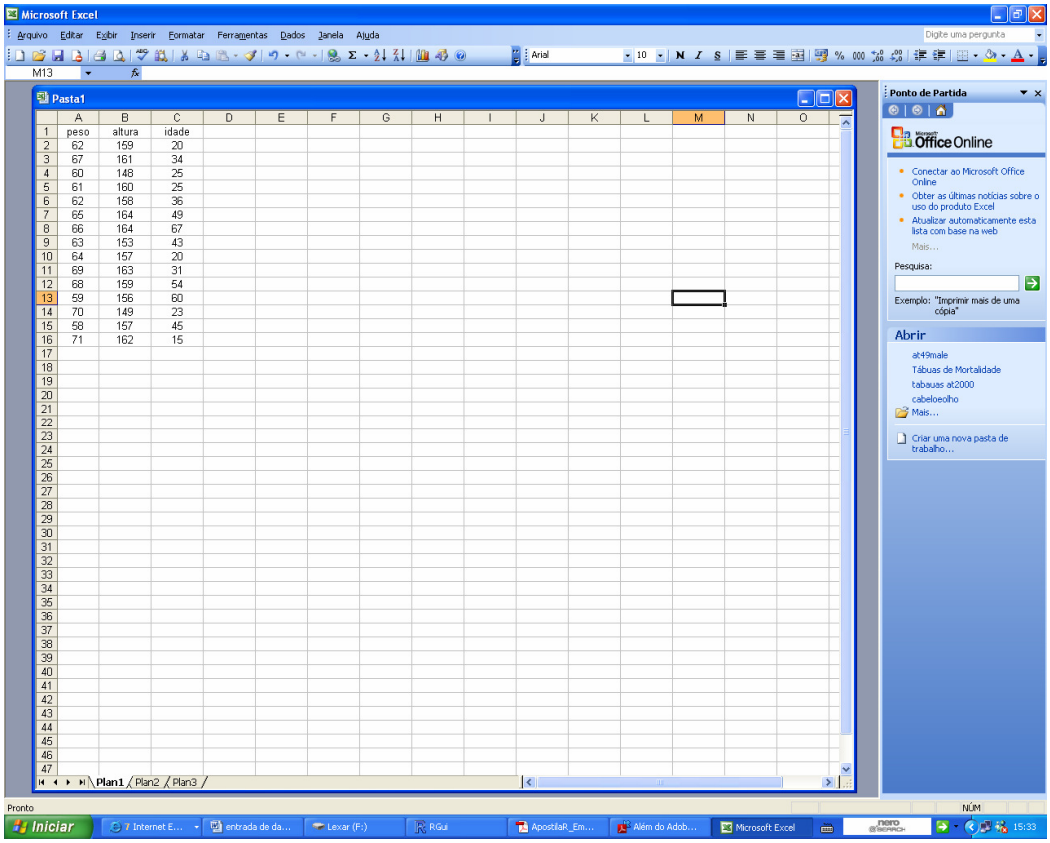
4.2 – Lendo Dados de um Arquivo Texto

Muitas vezes os dados que iremos utilizar já foram digitados e armazenados num arquivo utilizando outro programa. Neste caso, você pode importá-los sem a necessidade de digitá-los novamente. Vamos ver como importar dados externos quando eles estão em formato texto. Para isto vamos considerar duas funções do R: ***read.table( )*** e ***read.csv( ).***

Para mostrar como utilizar estas funções vamos construir um arquivo de dados no EXCEL. Comece abrindo o EXCEL irá aparecer para você a seguinte janela.



Geralmente as colunas da planilha correspondem às variáveis e as linhas às observações. Vamos entrar com os dados relativos ao peso, altura e idade das 15 mulheres, com mostrada na figura abaixo. Na primeira célula das colunas escreva os nomes das variáveis.



Agora que entramos os dados vamos salvá-los em um arquivo. Para isto clique no menu principal em arquivo e depois em salvar como. Em ***salvar em*** selecione o endereço onde você deseja salvar o arquivo. Sugestão: salve na pasta aulaR criada que você criou na aula anterior. Em nome do arquivo escreva **mulheres**, o nome do arquivo a ser criado. Em salvar como tipo selecione a opção ***texto (separado por tabulações)***. Finalmente clique em salvar. Deste modo o arquivo é criado no formato texto.

Repita os passos acima, salvando o arquivo com o nome de **mulherescsv.** Escolha em salvar como tipo ***csv (separado por vírgulas).*** Deste modo o arquivo é criado no formato texto onde as colunas (variáveis) são separadas por ponto e vírgula (;).

Agora que criamos os arquivos mulheres e mulherescsv, vamos ver como fazemos para lê-los no R. Vamos utilizar a função ***read.table*** para ler arquivos texto separado por tabulação e ***read.csv*** para ler arquivos texto separado por vírgulas.

Utilizando a função *read.table* para ler o arquivo mulheres **-** Mude o diretório do R para o diretório onde você salvou os arquivos mulheres e mulherescsv. Para ler o arquivo mulheres execute o comando

```
> data <-read.table("mulheres.txt",dec=",",header=T)
> data
    pesos altura idade
1      62    159    20
2      67    161    34
3      60    148    25
4      61    160    25
5      62    158    36
6      65    164    49
7      66    164    67
8      63    153    43
9      64    157    20
10     69    163    31
11     68    159    54
12     59    156    60
13     70    149    23
14     58    157    45
15     71    162    15
```

Com o comando acima o R lê os dados do arquivo mulheres.txt e armazena no data.frame data. O primeiro argumento da função é o nome do arquivo que deve estar sempre entre aspas. O segundo argumento indica que no arquivo excel a ser lido a separação da parte inteira da parte decimal dos números é feita pelo símbolo de vírgula (***dec=","***), diferente do R que utiliza que utiliza o símbolo de ponto para isto. O terceiro argumento indica que os dados da primeira linha devem ser entendidos como os nomes das variáveis presentes nas colunas (***header =T***). A opção ***header = F*** é opção padrão do R, isto é caso não especifiquemos ***header = T*** o R assumirá que os dados da primeira linha são observações e não nomes das variáveis. Agora vamos ler o arquivo mulherescsv utilizando a função ***read.csv*** **.**

```
> data1 <-read.csv("mulherescsv.csv",dec=",",sep=";",header =T)
> data1
    pesos altura idade
1      62    159    20
2      67    161    34
3      60    148    25
4      61    160    25
5      62    158    36
6      65    164    49
7      66    164    67
8      63    153    43
9      64    157    20
10     69    163    31
11     68    159    54
12     59    156    60
13     70    149    23
14     58    157    45
15     71    162    15
```

Observe que utilizamos um novo argumento na função ***read.csv***, o argumento ***sep***. No arquivo csv as variáveis presentes nas colunas da planilha excel são separadas por ponto e vírgula. Isto é informado ao R ***sep* = “;”** .

Aula 5 - Análise Descritiva e Exploratória de Dados – variáveis qualitativas

Nesta aula vamos utilizar o arquivo *cabeloeolho* (ver página 54 e arquivo em anexo) que possui informações sobre as variáveis: cor dos olhos, cor dos cabelos e sexo para 592 pessoas. O arquivo encontra-se no formato csv. Salve o arquivo no seu diretório de trabalho e depois, leia-o no R como indicado abaixo.

```
> # leitura de dados externos
> cabeloeolho<-read.csv("dados.csv", sep=";",dec =",", header=T)
```

Para ler as primeiras linhas do arquivo, use a função ***head.***

```
> head(cabeloeolho)          # exibe as primeiras linhas do arquivo
   sexo     cabelo  olho
1 masculino  preto  preto
2 masculino  preto  preto
3 masculino  preto  preto
4 masculino  preto  preto
5 masculino  preto  preto
6 masculino  preto  preto
```

Observe que as variáveis: sexo, cor dos cabelos e cor dos olhos foram nomeadas no R como sexo, cabelo e olho.

Utilizando este arquivo, vamos ver como executar as seguintes tarefas no R: construção de tabelas de freqüências absolutas e relativas, construção de diagramas de barras e de setores.

5.1 - Construção de Tabelas de Freqüências

Para obter as tabelas de freqüência vamos usar a função ***table.*** Para obter a tabela de freqüência da variável cor dos cabelos, faça:

```
> t1<- table(cabeloeolho$cabelo)
> t1
cabelo
castanho     loiro    preto    ruivo
      71       127      108      286
```

A tabela acima mostra as freqüências absolutas da variável cabelo, por exemplo, o número 127 indica que há 127 pessoas com cabelos loiros.

Vimos que para fazermos referência a uma variável de um data.frame temos de escrever o nome do data.frame seguido do símbolo **$** e do nome da variável. Podemos fixar o data.frame usando a função ***attach.*** Feito isto, podemos fazer referência à variável apenas pelo seu nome.

Utilizando a função ***attach*** com o data.frame ***cabeloeolho.***

```
>attach(cabeloeolho)
```

Após a execução do comando acima o R busca todas as variáveis no data.frame fixado. Quando você terminar de trabalhar com o data.frame fixado utilize o comando ***detach (nome do dataset)*** para desabilitar o comando ***attach.***

Uma vez fixado o data.set **cabeloeolho**, para construir tabela de freqüências, basta fazer:

```
>t1<- table(cabelo)        # calcula a tabela de freqüências da variável
cor do cabelo e guarda no objeto t1
> t1
cabelo
castanho     loiro    preto    ruivo
      71       127      108      286
```

Para obter a tabela de freqüências relativas, faça:

```
>prop.table(t1)      # exibe uma tabela com as freqüências relativas
(proporções)
cabelo
 castanho     loiro      preto      ruivo
0.1199324   0.2145270   0.1824324   0.4831081
```

A proporção de pessoas com cabelos loiros é 0,2145270 (127/592), ou seja, 21,45% das pessoas possuem cabelos loiros. Na tabela acima as proporções são apresentadas com 7 casas decimais. Caso queira apresentar os resultados com um número menor de casas decimais, por exemplo, duas, use a função ***round*** como segue.

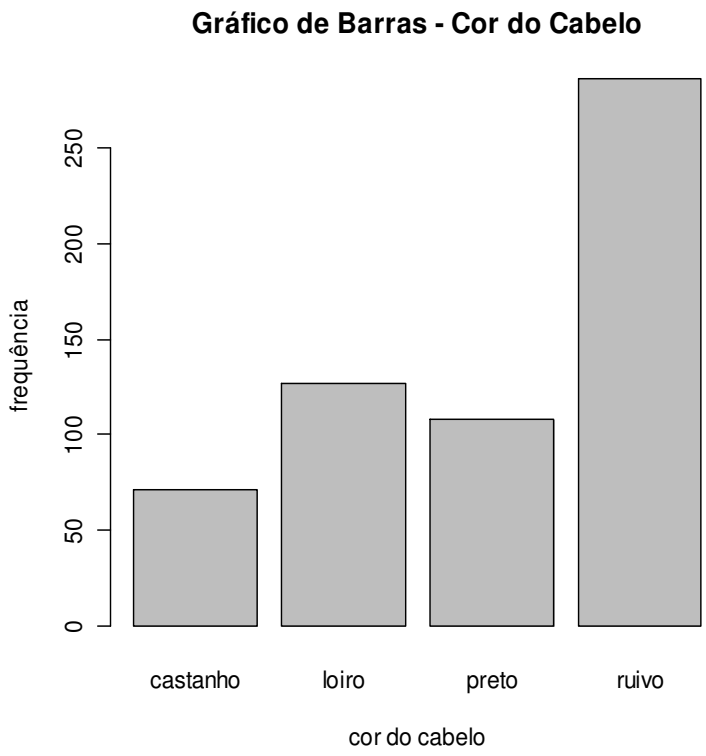
```
>round(prop.table(t1), 2)  # o 2 indica arredondamento com duas casas
decimais
cabelo
castanho    loiro    preto    ruivo
    0.12     0.21     0.18     0.48
```

Observe que as casas decimais nas freqüências relativas foram reduzidas à duas. Caso queira outra quantidade de casas decimais mude o valor 2 para o valor que você desejar.

5.2 - Construção de Diagramas de Barras e de Setores

Para construir o diagrama de barras da variável cor dos cabelos, execute o seguinte comando:

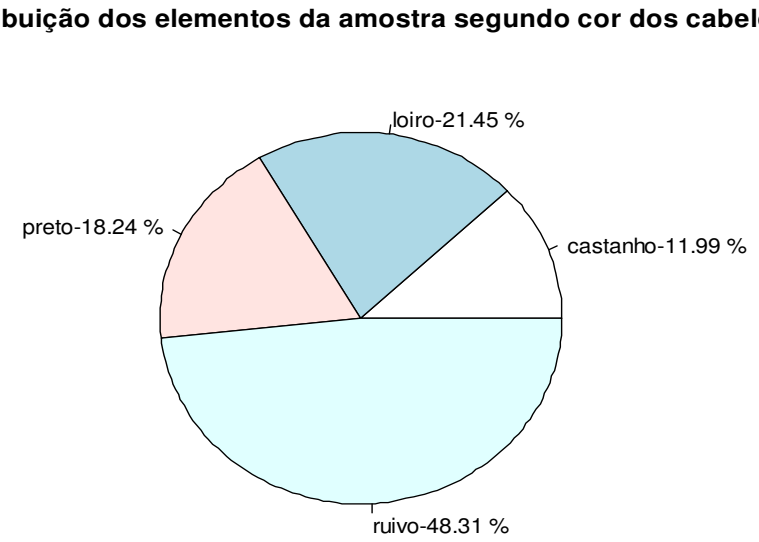
```
>barplot(t1,main="Gráfico de Barras - Cor do Cabelo", ylab= "frequência",
xlab="cor do cabelo")
```



Observe que o argumento do gráfico é a tabela de freqüências absolutas da cor dos cabelos, t1. O argumento ***main*** especifica o título do gráfico e os argumentos ***xlab*** e ***ylab*** os nomes das variáveis nos eixos x e y.

Para construir o gráfico de setores (gráfico de pizza) da variável cor dos olhos, faça como segue.

```
>pie(t1, labels = paste(paste(c("castanho","loiro","preto", "ruivo"),
round(prop.table(t1)*100,2), sep="-"), "%",sep=" "),main="Distribuição
dos elementos da amostra segundo cor dos cabelos")
```



Entendendo o comando acima:

a) Na função ***pie*****,** o argumento t1 é a tabela ou vetor com as freqüências usadas para construir o gráfico. Independente de t1 ser uma tabela de freqüências relativas ou absolutas o gráfico será o mesmo.

b) O argumento ***labels*** é usado para nomear as categorias da variável. Na figura acima especificamos os nomes e também as porcentagens correspondentes. Para isto fizemos uso do comando ***paste*** já visto na aula 2. Para entender o que faz o comando ***paste*** execute-o separadamente, como segue:

```
>paste(c("castanho","loiro","preto","ruivo"),round(prop.table(t1)*100,
2),sep="-")
>paste(paste(c("castanho","loiro","preto","ruivo"),round(prop.table
(t1)*100,2), sep="-"), "%",sep=" ")
```

5.3 – Exercícios

Dados para construção do arquivo cabeloeolho.

Sexo feminino

	cabelo			
olho	castanho	loiro	preto	ruivo
azul	7	64	9	34
castanho	7	5	5	29
preto	16	4	36	66
verde	7	8	2	14

Sexo masculino

	cabelo			
olho	castanho	loiro	preto	ruivo
azul	10	30	11	50
castanho	7	5	10	25
preto	10	3	32	53
verde	7	8	3	15

1) Obtenha a distribuição de freqüências absoluta e relativa para as variáveis: sexo e cor dos olhos. Construa o diagrama de barras a partir da tabela de freqüências relativas.

2) Construa o diagrama de pizza para as variáveis sexo e cor dos olhos.

Aula 6 – Análise Descritiva e Exploratória de Dados – variáveis quantitativas

Nesta aula vamos aprender a utilizar o R para fazer a descrição de um conjunto de dados. Para isto vamos utilizar o conjunto de dados iris, disponível no R. Este famoso conjunto de dados possui medidas da largura e comprimento da sépala e da pétala, em centímetros, para 50 flores de cada uma de 3 espécies de íris: *setosa, versicolor* e *virginica.*

Além deste conjunto de dados, o R disponibiliza outros conjuntos de dados. Para listar estes conjuntos de dados, faça:

```
> data()
```

Para carregar o conjunto de dados ***iris*** execute o seguinte comando:

```
>data(iris)    # entre parênteses informamos o nome do conjunto de dados
>iris           # exibe o conjunto de dados
```

Para ler as primeiras linhas do conjunto de dados, faça:

```
>head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

O conjunto de dados ***iris*** possui 4 variáveis quantitativas e uma variável categórica. A seguir vamos ver como utilizar o R para fazer uma análise descritiva da variável comprimento da sépala (Sepal.Lenght). Começaremos com a representação gráfica da distribuição desta variável. Vários gráficos podem ser utilizados para isto: histograma, diagrama de ramo e folhas, diagrama de pontos e boxplot.

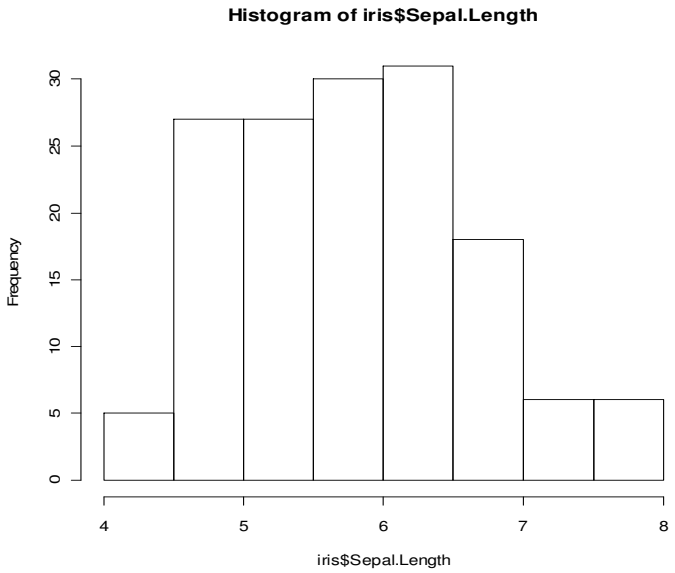
Afixe o arquivo iris utilizando a função *attach.*

```
>attach(iris)
```

6.1 – Histograma

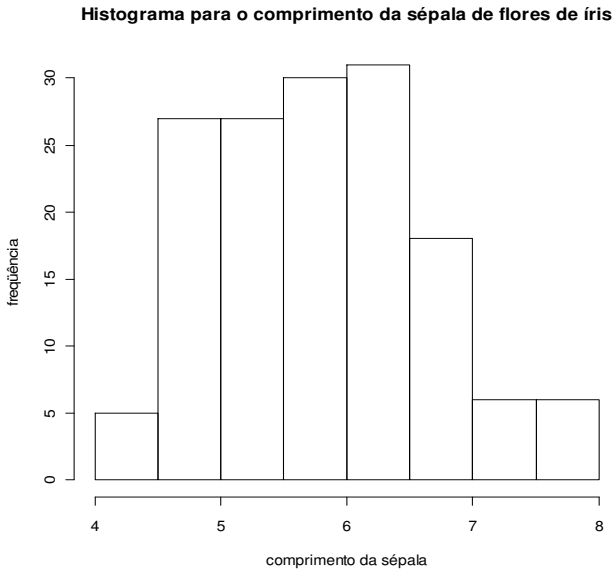
Podemos construir o histograma de freqüências absolutas ou de densidade. Para isto usamos a função ***hist.***

```
>hist(Sepal.Length)
```



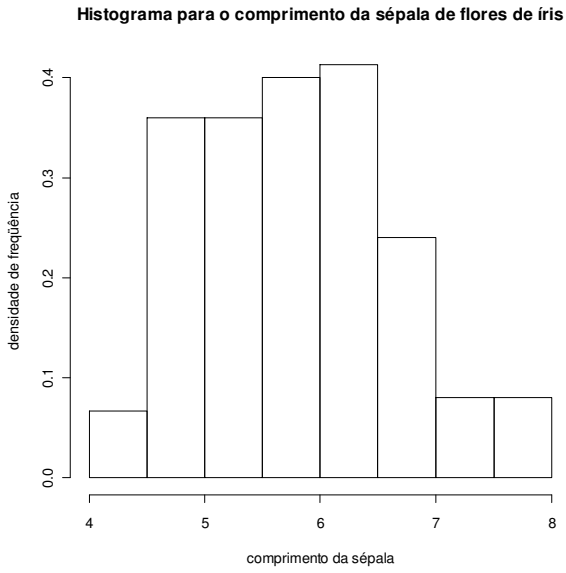
Observe que no eixo y estão as freqüências absolutas. Por exemplo, há 5 plantas com comprimento de sépala entre 4 e 4,5 cm. Podemos modificar o título do gráfico e os nomes das variáveis presentes nos eixos x e y. Para isto, vamos usar os argumentos ***main, xlab*** e ***ylal***. Os nomes devem estar entre aspas duplas.

```
>hist(Sepal.Length, main="Histograma para o comprimento da sépala de flores de íris", xlab="comprimento da sépala", ylab="freqüência")
```



Para construir o histograma de densidades, utilizamos o argumento ***freq***. Se ***freq = T*** (situação padrão), o R produz o histograma de freqüências, se ***freq = F,*** o histograma de densidades.

```
>hist(Sepal.Length, freq=F, main="Histograma para o comprimento da sépala de flores de íris", xlab="comprimento da sépala", ylab="densidade de freqüência")
```



6.2 - Gráfico de Freqüências Acumuladas

Para construir o gráfico de freqüências acumuladas vamos construir uma função, um tipo de objeto muito útil do R. A construção de funções está além dos objetivos deste tutorial. Entretanto vamos explicar rapidamente como construí-las e executá-las. Embora o R já possua uma função para calcular a média dos elementos de um vetor, função ***mean,*** vamos construir uma função para esta tarefa.

Ao utilizarmos à função ***mean*** para calcular a média dos elementos de um vetor x escrevemos ***mean(x).*** O vetor x é o elemento de entrada da função (argumento da função) e a média de x é o elemento de saída da função.

Nossa função para calcular a media receberá o nome de média e terá como elemento de entrada um vetor x de observações. A seguir apresentamos a sintaxe desta função.

```
>media<-function(x)
 {
 + mediax<-sum(x)/length(x)
 + return(mediax)
 }
```

Com o comando media<-function(x) construímos a função de nome media cujo argumento de entrada é o vetor de observações x. Toda função realiza algumas tarefas. Estas tarefas são delimitadas pelo símbolo **{ }**. A primeira tarefa a ser realizada é o cálculo da média de x (mediax<-***sum(x)/length(x)***). Toda função possui um elemento de saída. Neste caso ele é a média de x. O comando ***return(mediax)*** indica que a função deve retornar a média de x.

O primeiro passo para executar uma função é carregá-la. Para isto você pode copiar e colar a sintaxe da função no R. Uma vez feito isto podemos utilizá-la. Considere o vetor de observações x com os valores7, 32, 50, 57, 43 e 12

```
> x<-c(7, 32, 50, 57, 43, 12)
```

Obtenha a média de x fazendo:

```
> media(x)
[1] 33.5
```

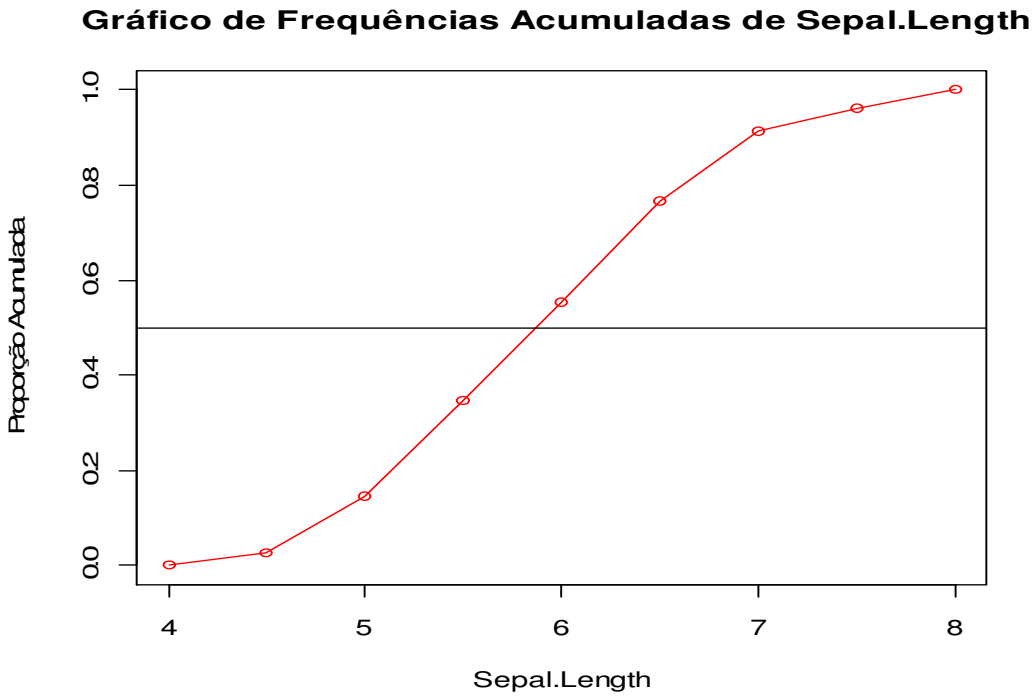
A função para construção do gráfico de freqüências acumuladas, de nome ogiva, é dada a seguir. Não se preocupe em entender toda a sintaxe da função. Utilize-a assim como você utiliza outras funções disponíveis no R. Antes de executar a função carregue-a copiando e colando os comandos no R.

```
ogiva<-function(x,g=1){
  g1<-(as.numeric(g)+1)
  a<-min(g1)
  breaks<-hist(x,plot=F)$breaks
  x.cut = cut(x[g1==a], breaks, right=FALSE)
  x.freq = table(x.cut)
  cumfreq0 = c(0, cumsum(x.freq))/length(x[g1==a])
  plot(breaks, cumfreq0, main=paste("Gráfico de Frequências Acumuladas
de",
  deparse(substitute(x)),""), xlab=paste(deparse(substitute(x))),
  ylab="Proporção Acumulada",col=a,ylim=c(0,1))
  lines(breaks, cumfreq0,col=a)
  abline(h=0.5)

for (i in sort(unique(g1))[-1]) {
   par(new=T)
   x.cut = cut(x[g1==i], breaks, right=FALSE)
   x.freq = table(x.cut)
   cumfreq0 = c(0, cumsum(x.freq))/length(x[g1==i])
   plot(breaks, cumfreq0, col=i,ylim=c(0,1),ylab="",xlab="" )
   lines(breaks, cumfreq0,col=i)
   abline(h=0.5)           }

if (length(g)!=1) {
  legend("bottomright", col=(sort(unique(g1))),lty=1,
         names(table(g)),bty="n",title=deparse(substitute(g)) )  }
                      }   ## Fim da funcao
```

Uma vez carregada à função, você pode utilizá-la assim como você faz com outras funções tais como ***mean, hist***, etc.

Executando a função para a variável comprimento da sépala.

```
> ogiva(Sepal.Length)
```



6.3 - Diagrama de Ramo e Folhas ("stem and leaf")

Para construir o diagrama de ramo e folhas vamos utilizar a função ***stem.***

```
> stem(Sepal.Length)
  The decimal point is 1 digit(s) to the left of the |

  42 | 0
  44 | 0000
  46 | 000000
  48 | 00000000000
  50 | 00000000000000000000
  52 | 00000
  54 | 0000000000000
  56 | 00000000000000
  58 | 0000000000
  60 | 000000000000
  62 | 0000000000000
  64 | 000000000000
  66 | 0000000000
  68 | 0000000
  70 | 00
  72 | 0000
  74 | 0
  76 | 00000
  78 | 0
```

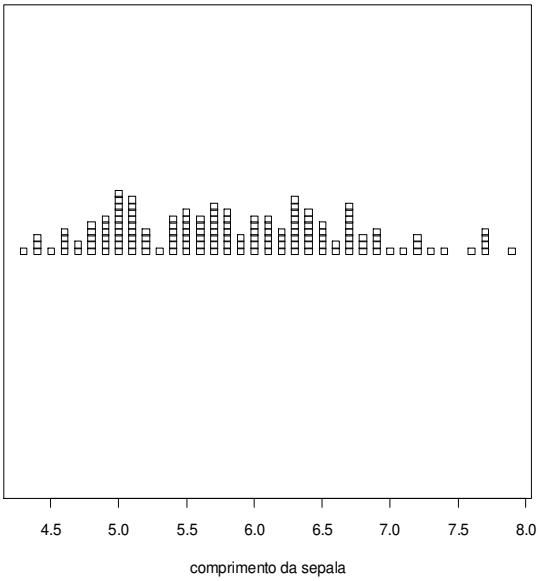
Na primeira linha do diagrama estão registrados os valores de 4,2 a 4,3 cm., na segunda linha de 4,4 a 4,5, e assim por diante. Observe que exite uma observação entre 4,2 e 4,3, representada por 42|0 na primeira linha. Existem quatro observações entre 4,4 e 4,5 representadas por 44|0000. Os dados abaixo, em ordem crescente, ajudam a entender o diagrama acima.

```
> sort(iris$Sepal.Length)
 [1] 4.3 4.4 4.4 4.4 4.5 4.6 4.6 4.6 4.6 4.7 4.7 4.8 4.8 4.8 4.8 4.8 4.9 4.9
 [19] 4.9 4.9 4.9 4.9 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.1 5.1 5.1 5.1
 [37] 5.1 5.1 5.1 5.1 5.1 5.2 5.2 5.2 5.2 5.3 5.4 5.4 5.4 5.4 5.4 5.4 5.5 5.5
 [55] 5.5 5.5 5.5 5.5 5.5 5.6 5.6 5.6 5.6 5.6 5.6 5.7 5.7 5.7 5.7 5.7 5.7 5.7
 [73] 5.7 5.8 5.8 5.8 5.8 5.8 5.8 5.8 5.9 5.9 5.9 6.0 6.0 6.0 6.0 6.0 6.0 6.1
 [91] 6.1 6.1 6.1 6.1 6.1 6.2 6.2 6.2 6.2 6.3 6.3 6.3 6.3 6.3 6.3 6.3 6.3 6.3
[109] 6.4 6.4 6.4 6.4 6.4 6.4 6.4 6.5 6.5 6.5 6.5 6.5 6.6 6.6 6.7 6.7 6.7 6.7
[127] 6.7 6.7 6.7 6.7 6.8 6.8 6.8 6.9 6.9 6.9 6.9 7.0 7.1 7.2 7.2 7.2 7.3 7.4
[145] 7.6 7.7 7.7 7.7 7.7 7.9
```

6.4 - Diagrama de Pontos

Para construir o diagrama de pontos utilizamos a função ***stripchart.***

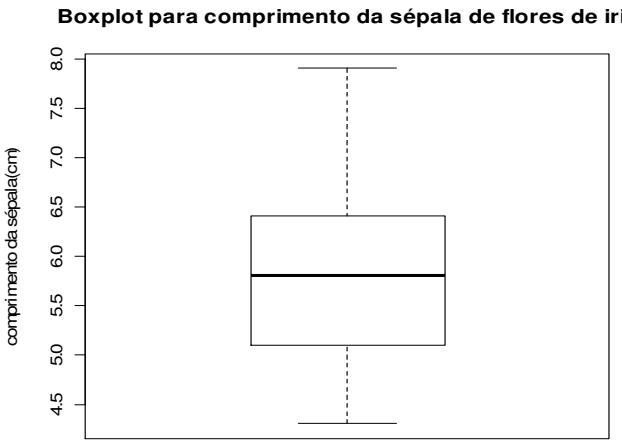
```
> stripchart(Sepal.Length,method="stack", xlab= “comprimento da sépala”)
```



6.5 - Boxplot

Para construir o boxplot vamos usar a função ***boxplot.***

```
> boxplot(Sepal.Length,ylab="comprimento da sépala(cm)", main="Boxplot
para comprimento da sépala de flores de iris")
```



6.6 - Obtendo Estatísticas Descritivas

6.6.1 - Medidas de Posição

Mínimo, Primeiro Quartil, Mediana, Média, Terceiro Quartil e Máximo.

Como já visto, as funções ***min, max, mean, median*** retornam os valores mínimos, máximo , médio e mediano de um conjunto de dados. Por exemplo, para obter a média do comprimento da sépala, faça:

```
> mean(Sepal.Length)
[1] 5.843333
```

Você também pode usar a função ***summary,*** que retorna várias medidas descritivas.

```
> summary(Sepal.Length)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.300   5.100   5.800   5.843   6.400   7.900
```

Onde:

Min – mínimo

1st Qu. – primeiro quartil

Median - mediana

Mean – média

3rd Qu. – terceiro quartil

Max - máximo

A função ***summary*** pode ser aplicada a todo o data.frame ao invés de uma variável específica.

```
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
 setosa    :50
 versicolor:50
 virginica :50
```

Para a variável espécie (species), qualitativa, não faz sentido calcular as medidas calculadas para as outras variáveis, que são todas quantitativas. Neste caso o R retorna as freqüências em cada categoria.

6.6.2 - Medidas de Variação: Média, Desvio Padrão e Coeficiente de Variação

O comando ***summary*** não retorna a variância, nem o desvio padrão. Para obtê-los use as funções ***var*** (de variance) e ***sd*** (de standard deviation)

```
> var(Sepal.Length)
[1] 0.6856935
> sd(Sepal.Length)
[1] 0.8280661
```

Se aplicamos estas função ***var*** ao data.frame íris obtemos o seguintes resultado:

```
> var(iris)
             Sepal.Length Sepal.Width Petal.Length Petal.Width Species
Sepal.Length   0.68569351 -0.04243400    1.2743154   0.5162707      NA
Sepal.Width   -0.04243400  0.18997942   -0.3296564  -0.1216394      NA
Petal.Length   1.27431544 -0.32965638    3.1162779   1.2956094      NA
Petal.Width    0.51627069 -0.12163937    1.2956094   0.5810063      NA
Species                NA          NA           NA          NA      NA
Warning message:
In var(iris) : NAs introduzidos por coerção
```

A matriz apresentada acima é chamada de matriz de variâncias e co-variâncias. Na diagonal da matriz (em negrito) temos as variâncias de cada uma das variáveis. A variância de comprimento da sépala é 0,6857 cm^2, da largura é 0,1899 cm^2. Os elementos fora da diagonal são chamados de co-variâncias. Falaremos sobre eles mais adiante. Observe que para a variável espécie (Species) o R retorna o símbolo NA. Isto ocorre porque esta é uma variável qualitativa. Isto também acontece quando calculamos o desvio padrão.

```
> sd(iris)
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
   0.8280661    0.4358663    1.7652982    0.7622377           NA
Warning message:
In var(as.vector(x), na.rm = na.rm) : NAs introduzidos por coerção
```

Ao invés de indicar como argumento das funções ***var*** e ***sd*** todo o data.frame íris podemos selecionar somente as variáveis que nos interessam.

```
> sd(íris[,1:4]) #seleciona as variáveis nas colunas 1 a 4 e todas as
linhas do data.frame.
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
   0.8280661    0.4358663    1.7652982    0.7622377
```

Para obter o coeficiente de variação da variável comprimento da sépala, faça:

```
> cv<- sd(Sepal.Length)/mean(Sepal.Length)
> cv
[1] 0.1417113
```

Para obter o ***CV*** das variáveis comprimento e largura da sépala e da pétala, faça

```
> cv<- sd(iris[,1:4])/mean(iris[,1:4])
> cv
Sepal.Length  Sepal.Width Petal.Length  Petal.Width
   0.1417113    0.1425642    0.4697441    0.6355511
```

6.6.3 - Obtendo os Quantis da Distribuição: quartis, decis, percentis

Como regra geral podemos utilizar o comando **quantile( )** para os ***quartis, decis e percentis***. Basta, para isso, utilizar um vetor no segundo argumento com as probabilidades correspondentes aos quantis desejados. Por exemplo, para calcular o quantil de ordem 0,25.

```
> quantile(Sepal.Length, probs = 0.25)
25%
5.1
```

Para calcular todos os ***decis,*** faça o argumento ***probs*** igual ao vetor com as probabilidades correspondentes.

```
> quantile(Sepal.Length,probs=seq(0.1,0.9,0.1))
 10%  20%  30%  40%  50%  60%  70%  80%  90%
4.80 5.00 5.27 5.60 5.80 6.10 6.30 6.52 6.90
```

Observe que a mediana é igual a 5,8 cm e o quantil 0,20 é igual a 5 cm.

6.6.4 - Calculando os Escores Padronizados

Qual a distância em desvios padrões de uma flor de íris setosa com comprimento de sépala igual a 5 cm em relação ao comprimento médio da sépala das plantas da mesma espécie?

```
>z<-(5-mean(Sepal.Length[Species=="setosa"]))/sd(Sepal.Length[Species=
="setosa"])
> z
[1] -0.01702177
```

Para obter os escores padronizados para todas as plantas.

```
>z<-(Sepal.Length[Species=="setosa"]-mean(Sepal.Length[Species=="setosa"]))
/sd(Sepal.Length[Species=="setosa"])
> z
 [1]  0.26667447 -0.30071802 -0.86811050 -1.15180675 -0.01702177  1.11776320
 [7] -1.15180675 -0.01702177 -1.71919923 -0.30071802  1.11776320 -0.58441426
[13] -0.58441426 -2.00289548  2.25254817  1.96885193  1.11776320  0.26667447
[19]  1.96885193  0.26667447  1.11776320  0.26667447 -1.15180675  0.26667447
[25] -0.58441426 -0.01702177 -0.01702177  0.55037071  0.55037071 -0.86811050
[31] -0.58441426  1.11776320  0.55037071  1.40145944 -0.30071802 -0.01702177
[37]  1.40145944 -0.30071802 -1.71919923  0.26667447 -0.01702177 -1.43550299
[43] -1.71919923 -0.01702177  0.26667447 -0.58441426  0.26667447 -1.15180675
[49]  0.83406695 -0.01702177
```

Qual a média e o desvio padrão de Z?

```
>mean(z)
[1] -6.761407e-16
>sd(z)
[1] 1
```

A média é – 6,76 x 10^{-16}, isto é praticamente zero. Para apresentar o resultado com 4 casas decimais, faça

```
>round(mean(z),4)
[1] 0
```

6.7 - Comparando as Três Espécies de Íris

6.7.1 - Obtendo Medidas Descritivas por Espécie

Como a espécie influencia no comprimento da sépala de flores de íris? Para responder esta pergunta, vamos fazer uma análise descritiva desta variável para cada uma das espécies, começando com as medidas descritivas.

Obtendo medidas descritivas para a espécie *setosa.*

```
> summary(Sepal.Length[Species=="setosa"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.300   4.800   5.000   5.006   5.200   5.800
```

Obtendo medidas descritivas para a espécie *versicolor*

```
>  summary(Sepal.Length[Species=="versicolor"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.900   5.600   5.900   5.936   6.300   7.000
```

Obtendo medidas descritivas para a espécie *virginica*

```
>  summary(Sepal.Length[Species=="virginica"])
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.900   6.225   6.500   6.588   6.900   7.900
```

Há um modo mais fácil de fazer a análise por espécie. Para isto vamos usar a função ***tapply.*** A função ***tapply*** aplica uma função a uma variável segundo grupos definidos por uma segunda variável. Observe que para o exemplo o primeiro argumento da função é a variável que queremos estudar, o segundo é a variável que define os grupos a serem comparados e o terceiro a função de interesse.

```
> tapply(Sepal.Length,Species,summary)
$setosa
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.300   4.800   5.000   5.006   5.200   5.800

$versicolor
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.900   5.600   5.900   5.936   6.300   7.000
```

```
$virginica
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.900   6.225   6.500   6.588   6.900   7.900
```

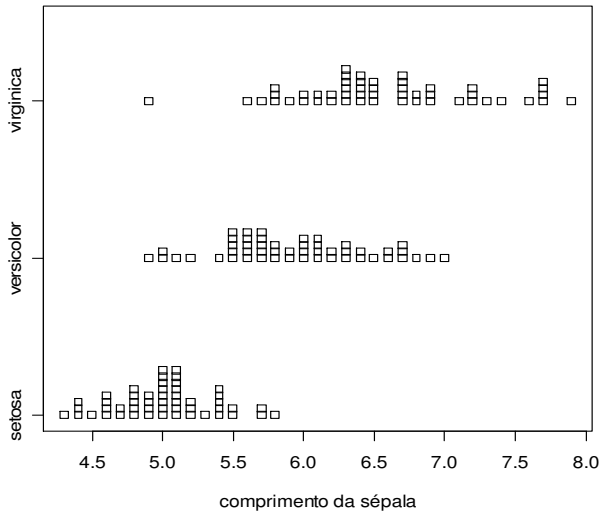
Obtendo os desvios padrões para cada espécie.

```
> tapply(Sepal.Length,Species,sd)
    setosa versicolor  virginica
 0.3524897  0.5161711  0.6358796
```

6.7.2 - Obtendo os Diagramas de Pontos por Espécie

Para obter o digrama de pontos segundo espécie (Species) basta colocar o símbolo ~ depois do nome da variável reposta seguido da variável que define os grupos.

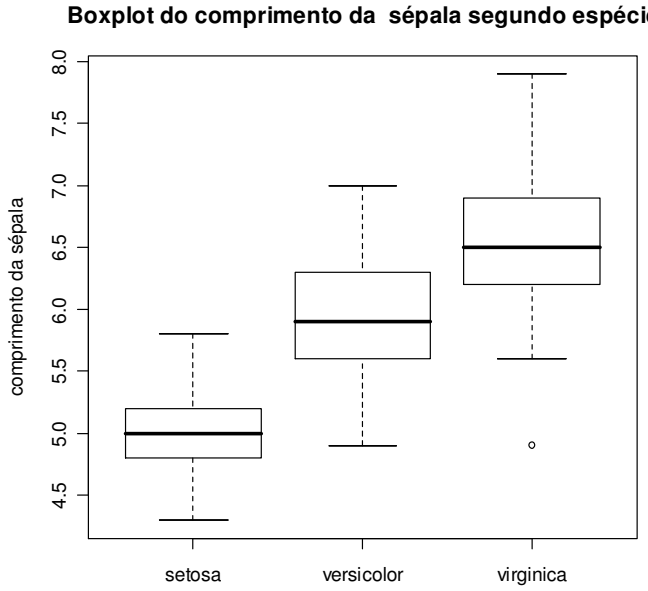
```
>stripchart(Sepal.Length~Species,method="stack",xlab="comprimento da sépala")
```



6.7.3 - Obtendo o Boxplot por Espécie

O boxplot por espécie é obtido de maneira análoga ao diagrama de pontos.

```
>boxplot(Sepal.Length~Species, ylab = "comprimento da sépala",main = "Boxplot do comprimento da  sépala segundo espécie")
```



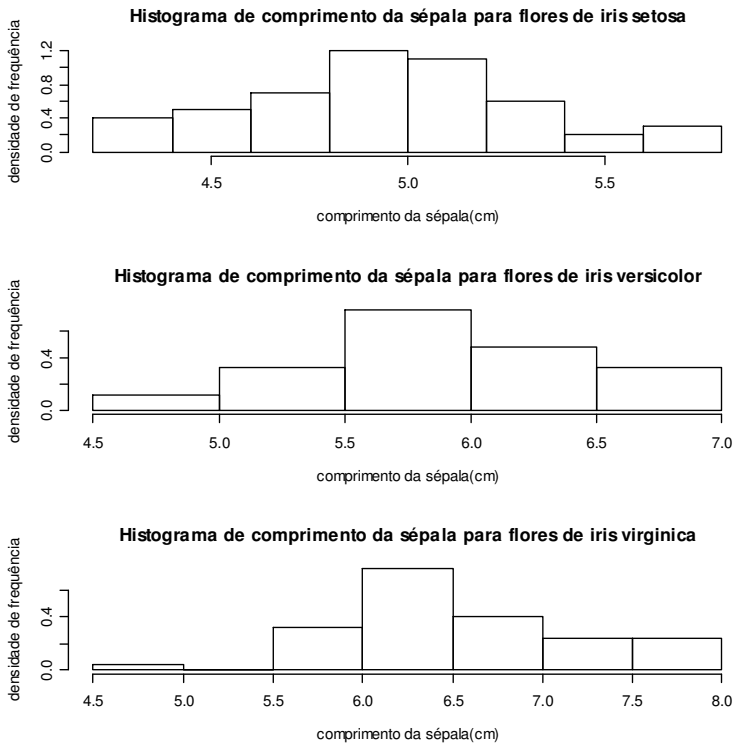
6.7.4 - Obtendo os Histogramas por Espécie

Antes de construir os histogramas para cada espécie, vamos dividir a janela gráfica em 3 partes. Para isto execute o comando:

```
> par(mfrow=c(3,1))    # mfrow=c(3,1) especifica que a janela é dividida em 3 linhas e 1 coluna.
```

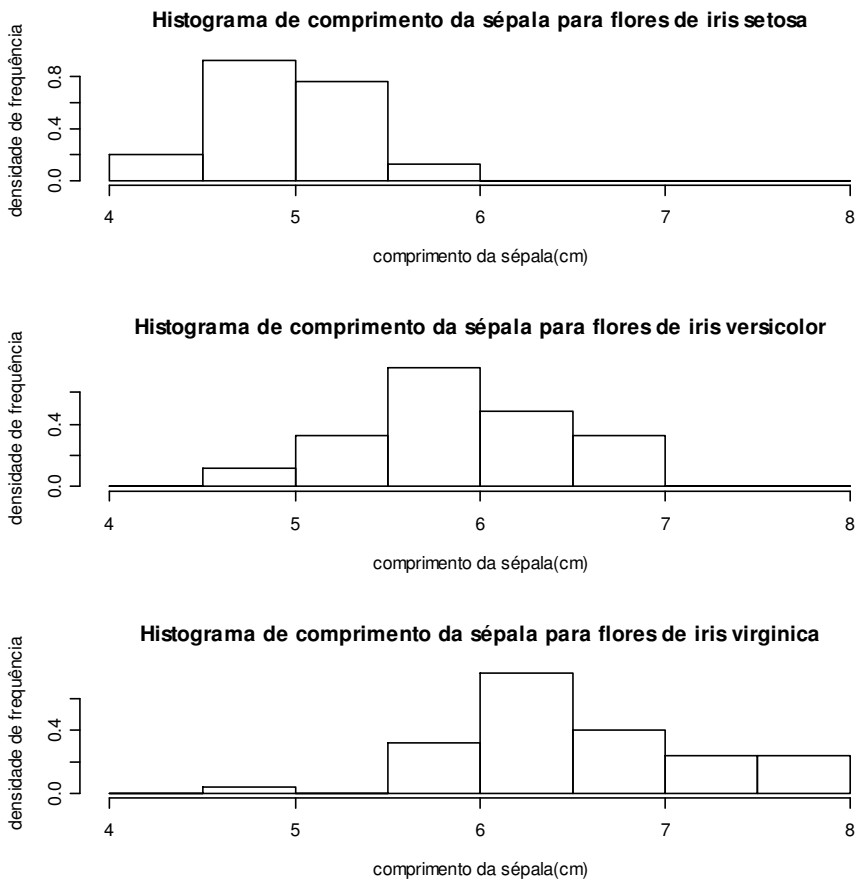
Observe que o R abriu uma janela gráfica. Agora é só especificar os gráficos a serem colocados em cada uma das células da janela.

```
>hist(Sepal.Length[Species=="setosa"],freq=F,main="Histograma de comprimento da sépala para flores de iris setosa", xlab="comprimento da sépala(cm)",ylab="densidade de frequência")

>hist(Sepal.Length[Species=="versicolor"],freq=F,main="Histograma de comprimento da sépala para flores de iris versicolor", xlab="comprimento da sépala(cm)",ylab="densidade de frequência")

>hist(Sepal.Length[Species=="virginica"],freq=F,main="Histograma de comprimento da sépala para flores de iris virginica", xlab="comprimento da sépala(cm)",ylab="densidade de frequência")
```



Observe que os histogramas criados pelo R possuem classes diferentes. Para facilitar a comparação podemos especificar as mesmas classes para os 3 histogramas como o argumento ***breaks*** (limites de classe).

```
>par(mfrow=c(3,1))
>    hist(Sepal.Length[Species=="setosa"],freq=F,breaks=seq(4,8,0.5),main
="Histograma de comprimento da sépala para flores de iris setosa",
xlab="comprimento da sépala(cm)",ylab="densidade de frequência")


>    hist(Sepal.Length[Species=="versicolor"],freq=F,breaks=seq(4,8,0.5),
main="Histograma de comprimento da sépala para flores de iris
versicolor", xlab="comprimento da sépala(cm)", ylab="densidade de
frequência")

>  hist(Sepal.Length[Species=="virginica"],freq=F,  breaks=seq(4,8,0.5),
main="Histograma de comprimento da sépala para flores de iris
virginica",xlab="comprimento da sépala(cm)", ylab = "densidade de
frequência")
```



6.8 - Exercícios:

1) Obtenha para variável comprimento da pétala os gráficos: histograma, boxplot, diagrama de pontos, diagrama de ramo e folhas e gráfico de freqüências acumuladas.

2) Suponha que você deseje classificar uma flor de íris quanto ás espécies *setosa, virginica* ou *versicolor* a partir do comprimento da sépala. Sabendo que seu comprimento de sépala é igual a 5 cm, em qual espécie você a classificaria?

3) Faça uma análise descritiva das variáveis: largura da sépala, largura e comprimento da pétala de flores de íris por espécie. Compare as espécies.

Aula 7 – Descrevendo a Associação Entre Variáveis Categóricas

Nesta aula, vamos utilizar o conjunto de dados ***cabeloeolho,*** usado na aula 3, para:

1) Construção de tabelas de freqüência segundo 2 variáveis (tabelas de classificação cruzada)

2) Construção de tabelas de freqüência relativas onde as proporções são calculadas em relação ao total por linha (ou por coluna).

3) Gráficos comparativos das distribuições de uma das variáveis segundo as categorias de outra variável.

4) Tabelas de freqüência para a situação de 3 variáveis categóricas.

Para isto leia o arquivo ***cabeloeolho*** e afixe-o usando o comando ***attach.***

7.1 - Construção de Tabelas de Freqüência Segundo Duas Variáveis (tabelas de classificação cruzada)

Para obter a tabela de classificação cruzada de cor dos cabelos versus cor dos olhos, faça:

```
> t2<- table(cabelo,olho)
> t2
          olho
cabelo     azul castanho preto verde
  castanho   17       14    26    14
  loiro      94       10     7    16
  preto      20       15    68     5
  ruivo      84       54   119    29
```

Importante: A variável que aparece como primeiro argumento da função ***table*** é sempre colocada na linha da tabela e a segunda variável na coluna.

A saída do R mostra a distribuição conjunta das variáveis cor dos cabelos e cor dos olhos. Observe que no cruzamento da linha cabelo preto com a coluna olho castanho

aparece o valor 15, informando que há 15 pessoas que possuem essas duas características. As freqüências fornecidas são as absolutas. Caso queira as freqüências relativas, faça:

```
> prop.table(t2)
        olho
cabelo            azul    castanho       preto       verde
  castanho 0.028716216  0.023648649  0.043918919  0.023648649
  loiro    0.158783784  0.016891892  0.011824324  0.027027027
  preto    0.033783784  0.025337838  0.114864865  0.008445946
  ruivo    0.141891892  0.091216216  0.201013514  0.048986486
```

Na tabela de freqüências relativas o valor 0,025337838 informa que aproximadamente 2,5% [(15/592) x 100] das pessoas amostradas possuem cabelos pretos e olhos castanhos. Observe que a soma de todas as freqüências relativas é igual a 1.

Obtendo as proporções com 3 casas decimais.

```
> round(prop.table(t2),3)
          olho
cabelo      azul castanho  preto  verde
  castanho 0.029    0.024  0.044  0.024
  loiro    0.159    0.017  0.012  0.027
  preto    0.034    0.025  0.115  0.008
  ruivo    0.142    0.091  0.201  0.049
```

7.2 - Construção de Tabelas de Freqüência com Marginais Fixas

Ao estudar a associação entre duas variáveis categóricas é usual obter para categoria de uma das variáveis a distribuição relativa da outra variável (distribuições condicionais). Por exemplo, podemos obter a distribuição da cor dos olhos para cada cor de cabelo. Na ausência de associação entre cor dos olhos e cor dos cabelos, esperaríamos observar a mesma distribuição de cor dos olhos para cada uma das categorias de cor dos cabelos.

Poderíamos também comparar as distribuições de cor de cabelo para as diferentes cores de olhos. A seguir, mostramos como obter no R estas distribuições para esses dois casos.

Caso 1: Obtendo as distribuições segundo cor dos olhos para cada cor de cabelo (proporções calculadas em relação aos totais das linhas)

```
>t2olho<-prop.table(t2,1) # o 1 indica que as proporções são calculadas
em relação ao total por linha
>t2olho
        olho
cabelo           azul    castanho      preto      verde
  castanho 0.23943662  0.19718310  0.36619718  0.19718310
  loiro    0.74015748  0.07874016  0.05511811  0.12598425
  preto    0.18518519  0.13888889  0.62962963  0.04629630
  ruivo    0.29370629  0.18881119  0.41608392  0.10139860
```

Como as proporções são calculadas em relação ao total por linha, então o número em destaque indica a proporção de pessoas que possuem olhos castanhos entre aquelas com cabelos pretos. Veja que a soma dos valores em cada linha é igual a 1.

Observe que a porcentagem de pessoas com olhos azuis varia muito com a cor do cabelo. Ela é de 18,51% (0,1851 x 100) entre aqueles com cabelos pretos e de 74,01% entre aqueles de cabelos loiros.

A distribuição esperada da cor do olho na ausência de associação é aquela obtida quando não consideramos a cor dos cabelos (distribuição marginal da cor dos olhos).

```
> prop.table(table(olho))

olho
     azul  castanho     preto     verde
0.3631757  0.1570946  0.3716216  0.1081081

> round(prop.table(table(olho)),4)
olho
    azul castanho    preto    verde
  0.3632   0.1571   0.3716   0.1081
```

Na tabela de freqüências relativas o valor 0,1571 informa que aproximadamente 15,71% [(93/592) x 100] das pessoas amostradas possuem olhos castanhos.

Caso 2 : Obtendo as distribuições segundo cor dos cabelos para cada cor de olho (proporções calculadas em relação ao total por coluna)

```
>t2cabelo <- prop.table(t2,2)    # o 2 indica que as proporções são
calculadas em relação ao total por coluna
> t2cabelo
         olho
cabelo           azul   castanho      preto      verde
  castanho 0.07906977  0.15053763  0.11818182  0.21875000
  loiro    0.43720930  0.10752688  0.03181818  0.25000000
  preto    0.09302326  0.16129032  0.30909091  0.07812500
  ruivo    0.39069767  0.58064516  0.54090909  0.45312500
```

O número em destaque indica a proporção de pessoas que possuem cabelos pretos entre aquelas com olhos castanhos. Observe que a distribuição da cor dos cabelos varia com a cor dos olhos, indicando que estas 2 variáveis estão associadas.

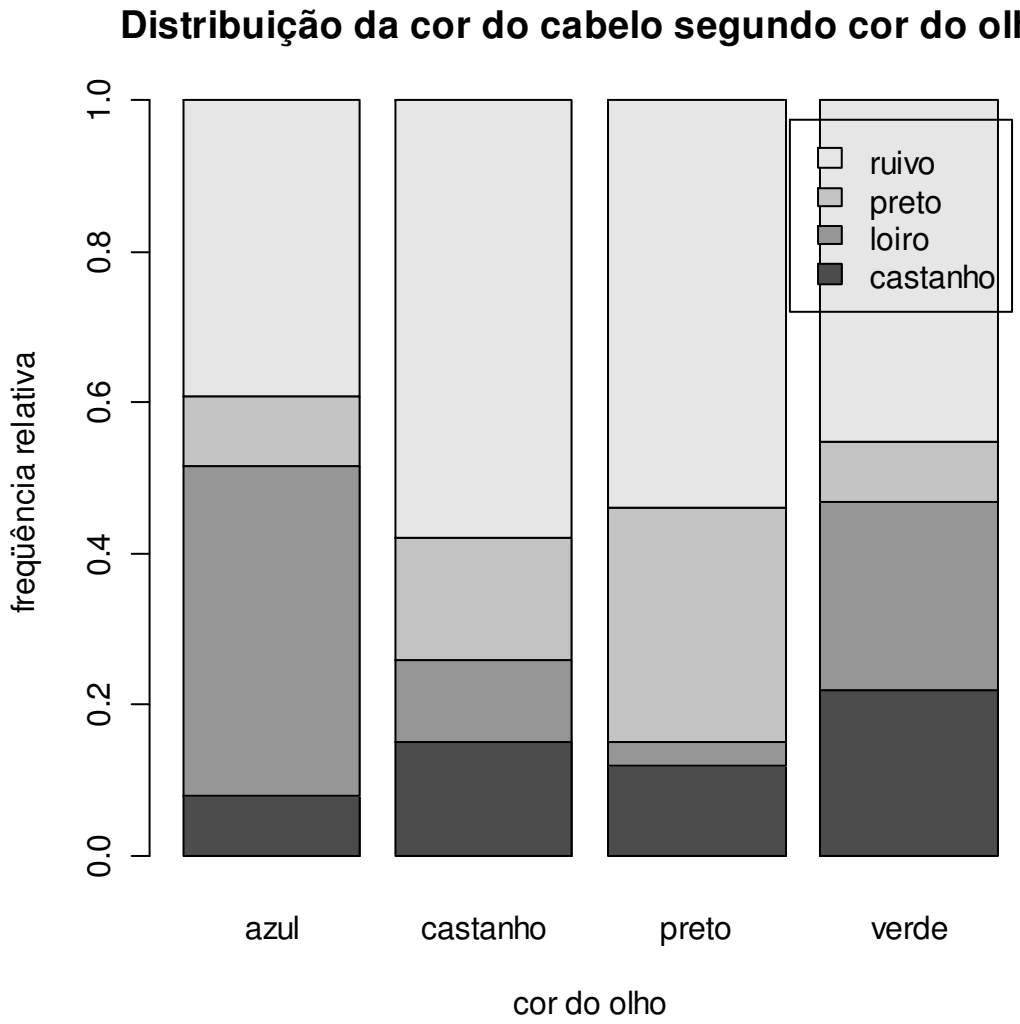
Some as proporções em cada coluna. Qual valor você encontrou?

7.3 - Gráficos Comparativos das Distribuições de uma das Variáveis Segundo as Categorias da Outra Variável

Caso 1: Comparando as categorias de cores de olhos quanto as distribuição de cor dos cabelos .

Vamos representar graficamente as distribuições presentes nas colunas da tabela t2cabelo, reproduzida a seguir. Para isto vamos usar a função *barplot.*

```
          olho
cabelo            azul    castanho      preto      verde
  castanho 0.07906977  0.15053763  0.11818182  0.21875000
  loiro    0.43720930  0.10752688  0.03181818  0.25000000
  preto    0.09302326  0.16129032  0.30909091  0.07812500
  ruivo    0.39069767  0.58064516  0.54090909  0.45312500

> barplot(t2cabelo, main = "Distribuição da cor do cabelo segundo cor do
olho", xlab = "cor do olho", ylab = "freqüência relativa", legend = T)
```



Importante:

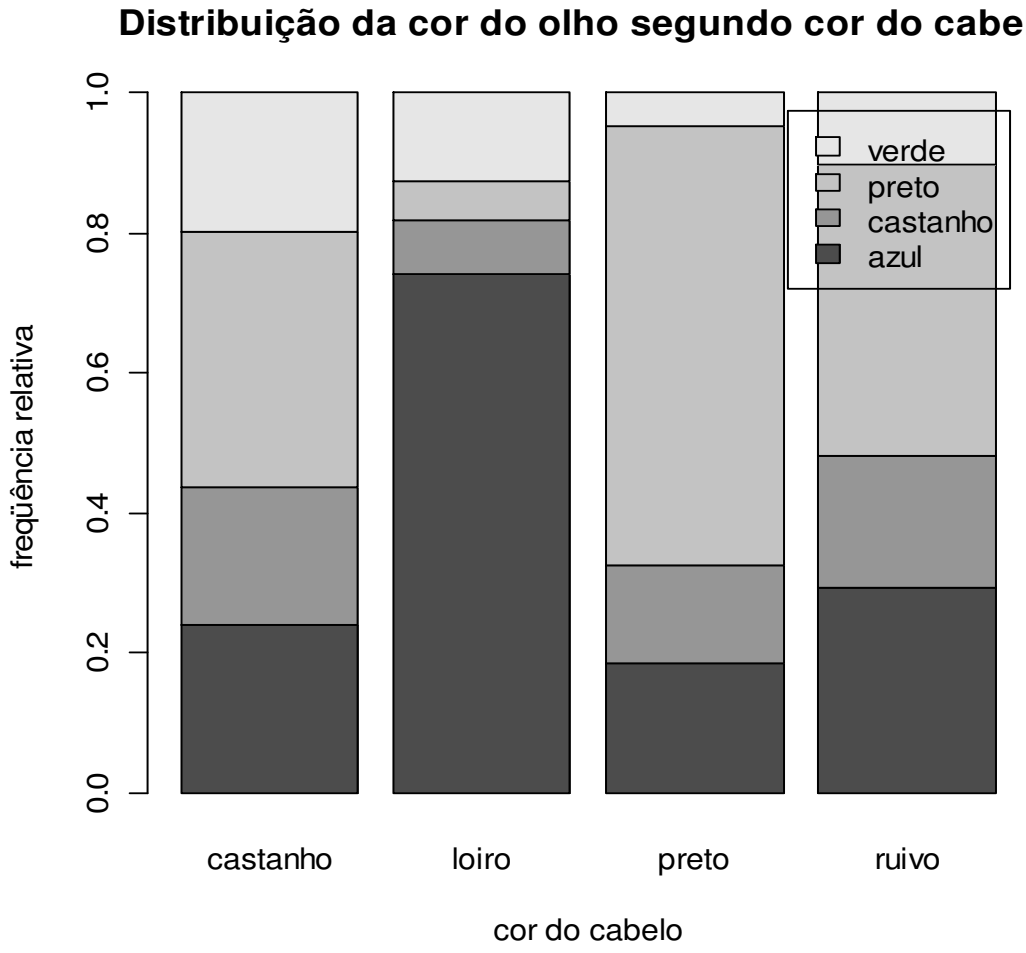
a) se ***legend* = *T*,** a legenda e adicionada ao gráfico. Se ***legend* = *F*,** nenhuma legenda é adicionada.

b) O gráfico é construído a partir das distribuições que se encontram nas colunas. Então, antes de fazer o gráfico construa a tabela de freqüências relativas de modo que as distribuições a serem representadas no gráfico fiquem nas colunas.

Caso 2: Comparando as categorias de cores de cabelos quanto ás distribuição de cor dos olhos

Como indicado acima, vamos construir a tabela de modo que as distribuições que queremos comparar fiquem nas colunas. Para isto vamos inverter a ordem das variáveis cor dos olhos e cor dos cabelos na função ***table.***

```
> t3<-table(olho, cabelo)
> t3olho<-prop.table(t3,2)
> barplot(t3olho,main="Distribuição da cor do olho segundo cor do cabelo", xlab="cor do cabelo", ylab="freqüência relativa", legend=T)
```



7.4 - Obtendo Tabelas de Classificação Cruzada Entre Duas Varáveis Para Cada Categoria de uma Terceira Variável.

Para obter tabelas de classificação quanto a cor dos olhos e cor dos cabelos para cada sexo, faça como segue:

```
> t4<- table(cabelo,olho,sexo)
> t4
, , sexo = feminino

          olho
cabelo     azul castanho preto verde
  castanho    7        7    16     7
  loiro      64        5     4     8
  preto       9        5    36     2
  ruivo      34       29    66    14

, , sexo = masculino

          olho
cabelo     azul castanho preto verde
  castanho   10        7    10     7
  loiro      30        5     3     8
  preto      11       10    32     3
  ruivo      50       25    53    15
```

A primeira tabela corresponde ao sexo feminino e a segunda ao sexo masculino. Observe agora que temos 64 mulheres loiras e de olhos azuis, enquanto para os homens este número é igual a 30.

As freqüências relativas podem ser obtidas em relação ao:

1) total geral de observações igual a 592 pessoas;

2) total de observações em cada categoria de cor de cabelo (primeiro argumento da função ***table)*;**

3) total de observações em cada categoria de cor dos olhos (segundo argumento da função ***table)***;

4) total de observações em cada sexo (terceiro argumento da função ***table)***;

Para o primeiro caso basta fazer,

```
> prop.table(t4)

, , sexo = feminino

          olho
cabelo            azul    castanho       preto       verde
  castanho 0.011824324 0.011824324 0.027027027 0.011824324
  loiro    0.108108108 0.008445946 0.006756757 0.013513514
  preto    0.015202703 0.008445946 0.060810811 0.003378378
  ruivo    0.057432432 0.048986486 0.111486486 0.023648649

, , sexo = masculino

          olho
cabelo            azul    castanho       preto       verde
  castanho 0.016891892 0.011824324 0.016891892 0.011824324
  loiro    0.050675676 0.008445946 0.005067568 0.013513514
  preto    0.018581081 0.016891892 0.054054054 0.005067568
  ruivo    0.084459459 0.042229730 0.089527027 0.025337838
```

Neste caso as proporções são calculadas sobre o total de 592 pessoas. Por exemplo, 10,81% representa o número de mulheres que possuem cabelos loiros e olhos azuis dentre o total geral, para os homens esse valor é 5,07%.

Para os casos 2, 3 e 4, basta especificar a variável em relação á qual queremos calcular as proporções utilizando o argumento *margin*. Se margin = 1, as proporções são calculadas em relação aos totais das categorias da variável que aprece como primeiro argumento na tabela (no exemplo cor do cabelo), margim = 2 e margin = 3 para as variáveis presentes nos 2º e 3º argumentos da tabela. Por exemplo, para obter a distribuição quanto á cor dos olhos e dos cabelos para cada sexo, faça:

```
> prop.table(t4, margin=3)   #margin = 1 especifica que as proporções
devem ser calculadas com  relação aos totais da variável que aparece como
terceiro argumento da função table que originou a tabela t4.
, , sexo = feminino

          olho
cabelo            azul    castanho       preto       verde
  castanho 0.022364217 0.022364217 0.051118211 0.022364217
  loiro    0.204472843 0.015974441 0.012779553 0.025559105
  preto    0.028753994 0.015974441 0.115015974 0.006389776
  ruivo    0.108626198 0.092651757 0.210862620 0.044728435

, , sexo = masculino

          olho
cabelo            azul    castanho       preto       verde
  castanho 0.035842294 0.025089606 0.035842294 0.025089606
  loiro    0.107526882 0.017921147 0.010752688 0.028673835
  preto    0.039426523 0.035842294 0.114695341 0.010752688
  ruivo    0.179211470 0.089605735 0.189964158 0.053763441
```

Como *margin=3* as proporções foram calculadas sobre o terceiro argumento, ou seja, sexo. Então 20,44% é a proporção de mulheres que possuem cabelos loiros e olhos azuis comparada sobre o total (313) de mulheres. Para os homens esse valor é 10,75% calculado sobre o total (279) de homens.

As freqüências relativas podem também ser calculadas fixando as categorias de 2 variáveis. Por exemplo, para obter as distribuições condicionais de cor dos olhos para cada cor de cabelo, por sexo, fazemos:

```
> prop.table(t4, margin=c(1,3)) # as variáveis que aparecem nos
argumentos 1 e 3 estão fixas.
, , sexo = feminino

          olho
cabelo           azul   castanho      preto       verde
  castanho 0.18918919  0.18918919  0.43243243  0.18918919
  loiro    0.79012346  0.06172840  0.04938272  0.09876543
  preto    0.17307692  0.09615385  0.69230769  0.03846154
  ruivo    0.23776224  0.20279720  0.46153846  0.09790210

, , sexo = masculino

          olho
cabelo           azul   castanho      preto      verde
  castanho 0.29411765 0.20588235 0.29411765 0.20588235
  loiro    0.65217391 0.10869565 0.06521739 0.17391304
  preto    0.19642857 0.17857143 0.57142857 0.05357143
  ruivo    0.34965035 0.17482517 0.37062937 0.10489510
```

Observe que foram fixados cabelo e sexo, ou seja, dentre as mulheres que possuem cabelos loiros 79,01% possuem olhos azuis. Para os homens essa proporção é de 65,21%.

7.5 - Exercícios

1) Para pensar: o padrão de associação entre cor de cabelo e cor de olhos é o mesmo para os 2 sexos?

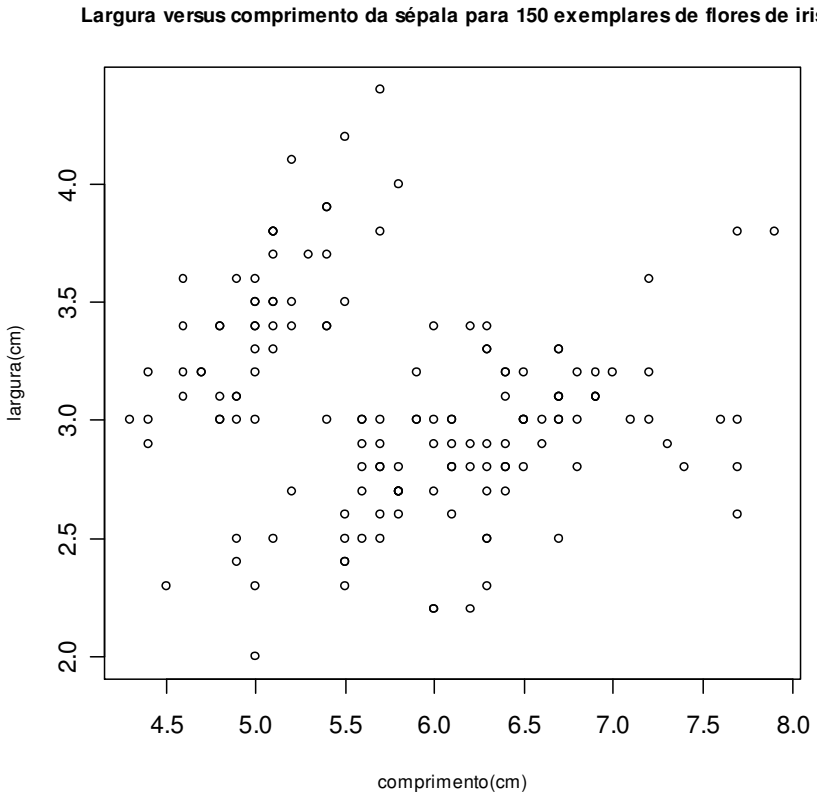
2) Utilizando tabelas de freqüência e gráficos estude a associação entre sexo e cor dos olhos e entre sexo e cor dos cabelos.

Aula 8 - Associação Entre Variáveis Quantitativas

Ao estudar a associação entre 2 variáveis quantitativas é usual construir diagramas de dispersão e calcular medidas de associação entre elas. Nesta aula, vamos ver como executar estas duas tarefas no R. Para isto, vamos utilizar o conjunto de dados iris que contém dados sobre comprimento e largura da sépala para flores de três espécies de íris: *versicolor, setosa* e *virginica.* Para isto carregue e afixe o data.frame iris.

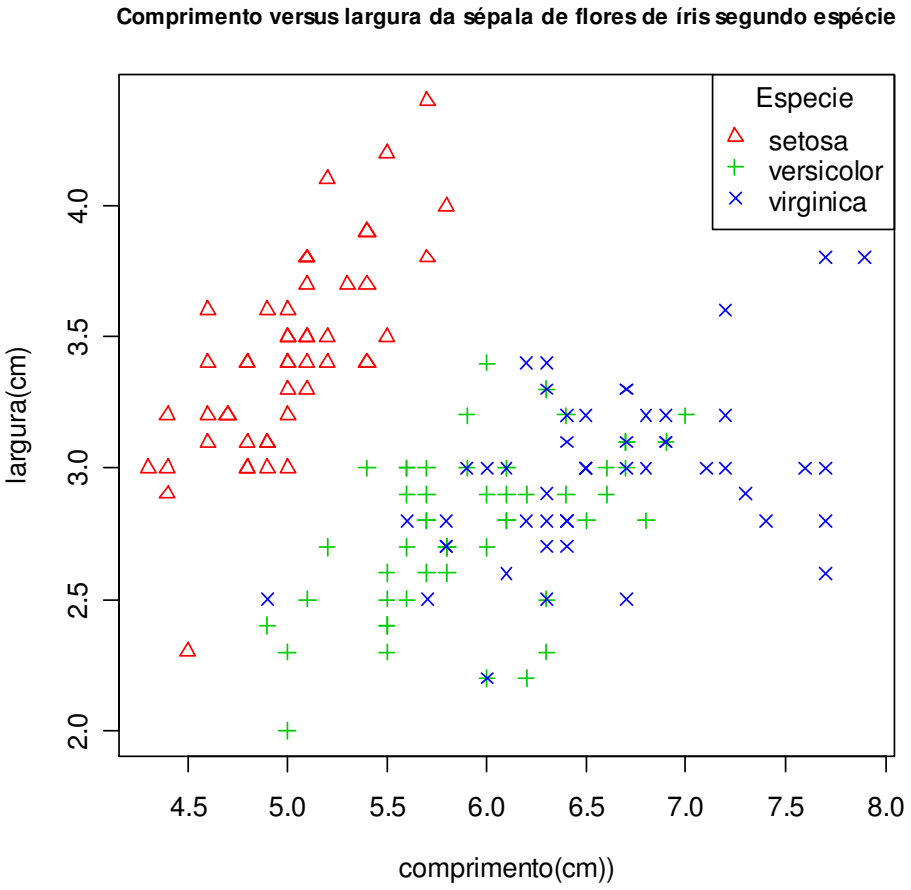
```
> data(iris)
> attach(iris)
```

Vamos iniciar nossa análise construindo o diagrama de dispersão da largura versus comprimento da sépala para os 150 exemplares de flores de íris observados.



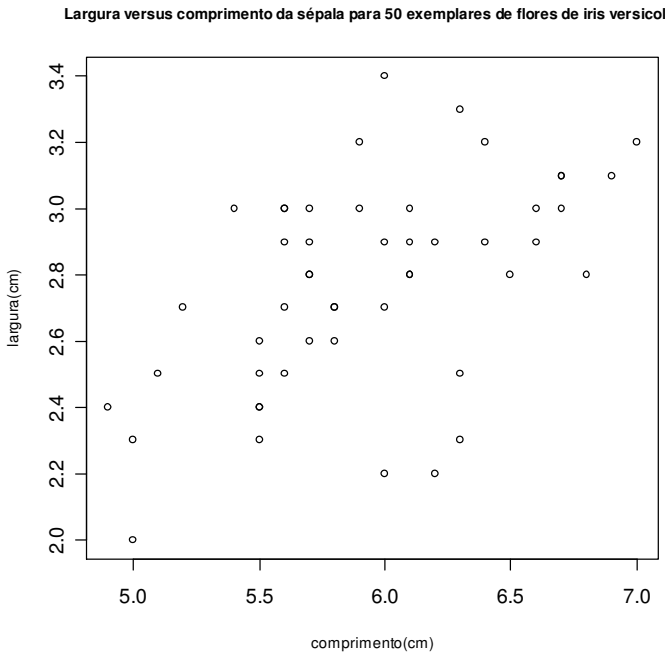
Observe que o gráfico apresenta 2 agrupamentos de pontos. Estes agrupamentos resultam do fato de termos 3 espécies de íris. Para visualizarmos como as espécies influenciam a associação entre comprimento e largura da sépala é interessante usar símbolos e cores diferentes para representá-las.

```
>plot(Sepal.Length,Sepal.Width, col=c(2:4)[Species], pch=c(2:4)[Species],
main= "Comprimento versus largura da sépala de flores de íris segundo
espécie",xlab ="comprimento(cm))", ylab="largura(cm)", cex.main=0.8)

># o argumento col=c(2:4)[Species] indica que cores diferentes devem ser
usadas para identificar as especies. (2 = vermelho, 3 = verde, 4 = azul)

># o argumento pch = c(2:4)[Species] indica que símbolos diferentes devem
ser usadas para identificar as especies. (2 =Δ, 3 = + , 4 = x)

># cex.main=0.8 reduz o tamanho das letras no título do gráfico (main =1
é o padrão)
>legend("topright"   ,legend=c("setosa",   "versicolor",   "virginica"),
pch=c(2:4), col=c(2:4),title= "Especie")
```



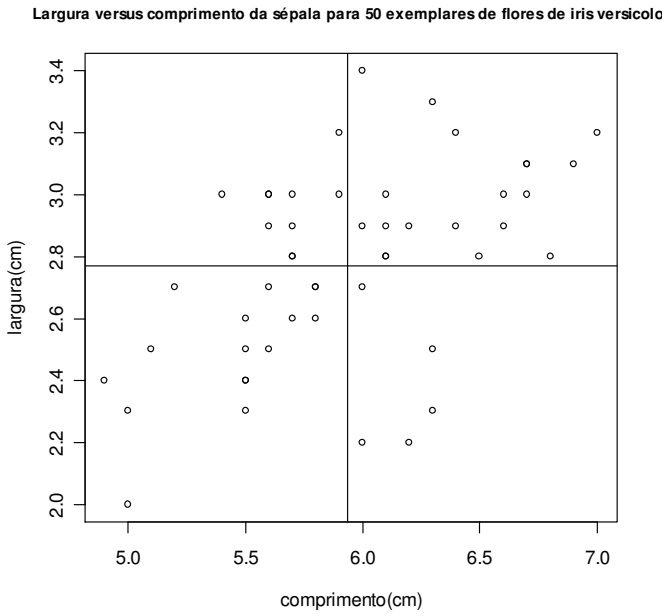
Como o gráfico sugere as três variedades apresentam padrões diferentes de associação entre o comprimento e largura da sépala, Vamos então analisar cada variedade separadamente, começando pela variedade *versicolor*

```
>plot(Sepal.Length[Species=="versicolor"],Sepal.Width[Species=="versicolo
r"], main="Largura versus comprimento da sépala para 50 exemplares de
flores de iris versicolor", cex.main=0.75,
xlab="comprimento(cm)",ylab="largura(cm)",cex.lab=0.75)
```



A inclusão no gráfico de linhas referentes às médias das variáveis pode ajudar na interpretação. Vamos fazer isto utilizando o comando ***abline.***

```
> abline(v=mean(Sepal.Length[Species=="versicolor"]))
# adiciona linha vertical cruzando o eixo x na media do comprimento da
sépala.

> abline(h=mean(Sepal.Width[Species=="versicolor"]))
# adiciona linha horizontal cruzando o eixo y na media da largura da
sépala.
```



O gráfico indica uma associação positiva entre as variáveis: comprimento e largura da sépala. Podemos quantificar esta associação calculando o coeficiente de correlação de Pearson. Porém antes disto vamos calcular a covariância amostral.

```
>cov(Sepal.Length[Species=="versicolor"],Sepal.Width[Species=="versicolor
"])
[1] 0.08518367
```

A correlação amostral é obtida dividindo a co-variância pelo produto dos desvios padrões das variáveis ou usando a função *cor*. No primeiro caso, fazemos:

```
>r<cov(Sepal.Length[Species=="versicolor"],Sepal.Width[Species=="versicol
or"])/(sd(Sepal.Length[Species=="versicolor"])*sd(Sepal.Width[Species=="
versicolor"]))
>r
[1] 0.5259107
```

Utilizando a função *cor,* temos:

```
>cor(Sepal.Length[Species=="versicolor"],Sepal.Width[Species=="versicolor
"])
[1] 0.5259107
```

As funções *cov* e *cor* podem ser aplicadas a mais de duas variáveis. Neste caso o R retorna respectivamente a matriz de variâncias e co-variâncias e a matriz de correlações. Veja a seguir as matrizes obtidas para as variáveis quantitativas do conjunto de dados Iris, espécie versicolor.

Matriz de covariâncias

```
> cov(iris[Species=="versicolor",1:4])# seleciona as linhas do data.frame
cuja variedade é versicolor e as colunas 1 a 4.
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length   0.26643265  0.08518367   0.18289796  0.05577959
Sepal.Width    0.08518367  0.09846939   0.08265306  0.04120408
Petal.Length   0.18289796  0.08265306   0.22081633  0.07310204
Petal.Width    0.05577959  0.04120408   0.07310204  0.03910612
```

Matriz de correlações

```
> cor(iris[Species=="versicolor",1:4])# seleciona as linhas do data.frame
cuja variedade é versicolor e as colunas 1 a 4.
             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000   0.5259107    0.7540490   0.5464611
Sepal.Width     0.5259107   1.0000000    0.5605221   0.6639987
Petal.Length    0.7540490   0.5605221    1.0000000   0.7866681
Petal.Width     0.5464611   0.6639987    0.7866681   1.0000000
```

Observe que a matriz acima é simétrica. *cor(Sepal.Length,Sepal.Width)* = *cor(Sepal.Width,Sepal.Length)*. Observe também que os elementos da diagonal são iguais a 1. A maior correlação é aquela entre comprimento e largura da pétala.

8.1 - Exercícios

1) Obtenha a matriz de variâncias e co-variâncias e a matriz de correlações para as outras espécies de íris.

2) Obtenha para cada uma das espécies os diagramas de dispersão entre comprimento e largura da pétala.

Aula 9 – Aplicação de Probabilidade Condicional – avaliação de testes diagnósticos

Uma aplicação interessante do conceito de probabilidade condicional é a avaliação da qualidade de testes diagnósticos. Geralmente a avaliação de um teste é feita aplicando-o a dois grupos de indivíduos: um grupo de indivíduos doentes e outro de não doentes. Espera-se que os indivíduos doentes (D) apresentem resultados positivos no teste (+) e os não doentes (ND) apresentem resultados negativos (-). Disto resultam 2 medidas de qualidade do teste, denominadas sensibilidade (s) e especificidade (*e*), definidas como as probabilidades condicionais:

s = P(+ | D) e *e* = P(- | ND).

Numa situação de diagnóstico interessa ao examinador conhecer outras duas probabilidades condicionais denominadas valores de predição do teste: valor de predição positivo (VPP) e valor de predição negativo (VPN), definidas como:

VPP = P(D | +) e VPN = P(ND | -).

Conhecidos os valores da sensibilidade, da especificidade, da proporção e da prevalência da doença (p = P(D), os valores de VPP e VPN são obtidos através do Teorema de Bayes, como:

$$VPP = \frac{ps}{ps + (1-p)(1-e)} \qquad VPN = \frac{(1-p)e}{(1-p)e + p(1-s)}$$

Nesta aula, vamos mostrar como obter no R os valores da sensibilidade, especificidade e valores de predição para um exemplo:

Exemplo: Um teste diagnóstico foi proposto para uma doença infecciosa que acomete cavalos adultos. Para avaliar a qualidade do teste, ele foi aplicado a 200 animais doentes e a 500 animais não doentes.

	Resultado do teste		
	Positivo	Negativo	Total
Doentes	150	50	200
Não doentes	20	480	500
Total	170	530	700

1) Cálculo de *s* e *e*

```
> s<-150/200
> e<-480/500
> s
[1] 0.75
> e
[1] 0.96
```

2) Supondo que a prevalência da doença é de 5%, obtenha os valores de predição.

```
> p<- 0.05            # fazendo prevalência igual a 0.05
> VPP<-(p*s)/((p*s)+(1-p)*(1-e))
> VPN<-(1-p)*(1-e)/ ((1-p)*(1-e) +p*(1-s))
> VPP
[1] 0.4966887
> VPN
[1] 0.7524752
```

3) Construa uma função de nome QT com os argumentos: D – número de doentes, ND – número de não doentes, NDP – número de doentes com resultados positivos, NNDN – número de não doentes com resultados negativos e p – prevalência da doença, que retorne os valores da sensibilidade, especificidade e valores de predição.

```
QT<-function(nd,nnd,ndp,nndn,p){
s<-ndp/nd
e<-nndn/nnd
VPP<-(p*s)/((p*s)+(1-p)*(1-e))
VPN<-(1-p)*(1-e)/ ((1-p)*(1-e) +p*(1-s))
resultado<-list(s,e,p,VPP,VPN)
names(resultado)<-c("sensibilidade","especificidade","prevalencia",
"VPP","VPN")
return(resultado)}
```

Executando a função para o exemplo:

```
> QT(200,500,150,480,0.05)
$sensibilidade
[1] 0.75
$especificidade
[1] 0.96
$prevalencia
[1] 0.05
$VPP
[1] 0.4966887
$VPN
[1] 0.7524752
```

9.1 - Exercícios:

1) Para o exemplo acima obtenha os valores de VPP e VPN para valores de prevalência variando de 0,1 a 0,9 em intervalos de tamanho 0,1. Faça diagramas de dispersão de VPP e VPN versus p. Comente**.**

2) Suponha que os valores de sensibilidade e especificidade foram dados. Modifique a função QT, alterando seus argumentos para e, s e p. Utilizando esta função calcule VPP e VPN para as seguintes situações:

a) p = 0,10, s = 0.80 e especificidade variando de 0,1 a 0,9 em intervalo de tamanho 0.1. Como a mudança nos valores da especificidade influencia VPP e VPN.

b) p = 0,10, e = 0.80 e sensibilidade variando de 0,1 a 0,9 em intervalo de tamanho 0.1. Como a mudança nos valores da sensibilidade influencia VPP e VPN.

Aula 10 – Distribuições de Probabilidade Binomial e Poisson

O R possui funções para calcular probabilidades e quantis para vários modelos de probabilidade discretos e contínuos. Nesta aula veremos como fazer isto para os modelos Binomial e Poisson.

10.1 - Distribuição Binomial

Considere uma variável aleatória Binomial com parâmetros n e p. Veremos, através de exemplos, como obter no R as seguintes quantidades:

a) Probabilidades pontuais P(X = x).

b) Probabilidades acumuladas P(X ≤ x).

c) Quantis da distribuição, isto é o menor valor x tal que P(X ≤ x) ≥ p.

Seja X uma variável aleatória Binomial com n = 20 e p = 0.3

a) Qual a probabilidade de X ser igual a 5?

Para calcular esta probabilidade vamos usar a função ***dbino(x,size,prob)***. Os argumentos desta função são: o valor de x e os parâmetros da distribuição: ***size,*** o tamanho da amostra e ***prob,*** a probabilidade de sucesso p.

```
> dbinom(5,size = 20,prob = 0.3)
[1] 0.1788631
```

O resultado será o mesmo se executarmos o comando ***dbinom(5, 20, 0.3).***

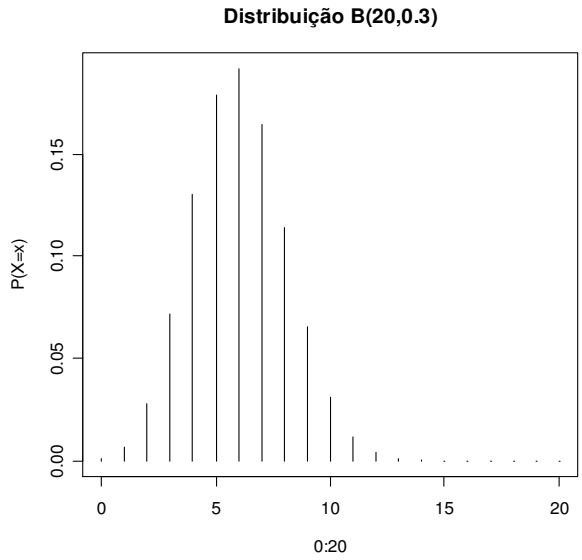
```
> dbinom(5,20,0.3)
[1] 0.1788631
```

A variável X assume os valores inteiros de 0 a 20. Vamos calcular as probabilidades para cada um dos valores de X e guardá-las no objeto pbin.

```
> pbin <-dbinom(0:20,20,0.3)
> round(pbin,4)
[1] 0.0008 0.0068 0.0278 0.0716 0.1304 0.1789 0.1916 0.1643 0.1144 0.0654
[11]0.0308 0.0120 0.0039 0.0010 0.0002 0.0000 0.0000 0.0000 0.0000 0.0000
[21] 0.0000
```

O gráfico das probabilidades P(X = x) versus X é obtido fazendo:

```
> plot(0:20, pbin,type="h",main="Distribuição B(20,0.3)")
```



Na função *plot* argumento *type* especifica o tipo do gráfico. Veja abaixo as opções para este argumento.

* "'p"' for *p*oints,

* "'l"' for *l*ines,

* "'b"' for *b*oth,

* "'c"' for the lines part alone of "'b"',

* "'o"' for both "*o*verplotted",

* "'h"' for "*h*istogram" like (or "high-density")

vertical lines,

* "'s"' for stair *s*teps,

* "'S"' for other *s*teps, see _Details_ below,

* "'n"' for no plotting.

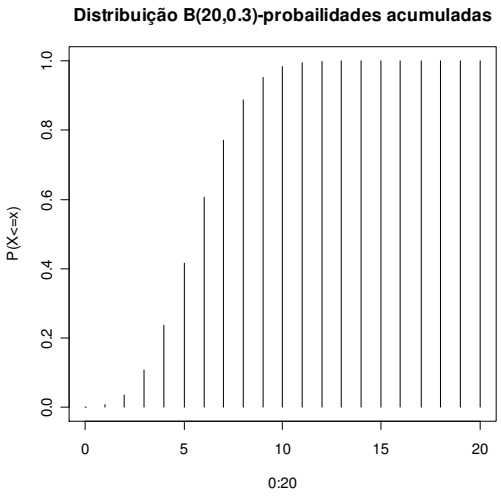
b) Calculando probabilidades acumuladas. Qual a P(X ≤ 5)?

Vamos usar a função ***pbinom***. Os argumentos são os mesmos da função ***dbinom***.

```
> pbinom(5,20,0.3)
[1] 0.4163708
```

Do mesmo modo que construímos o gráfico da distribuição de probabilidade podemos construir o gráfico com as probabilidades acumuladas.

```
> pacbin <-pbinom(0:20,20,0.3)
> pacbin
 [1] 0.0007979227 0.0076372598 0.0354831323 0.1070868045 0.2375077789
 [6] 0.4163708294 0.6080098122 0.7722717974 0.8866685371 0.9520381027
[11] 0.9828551836 0.9948618385 0.9987211204 0.9997389530 0.9999570600
[16] 0.9999944497 0.9999994573 0.9999999623 0.9999999983 1.0000000000
[21] 1.0000000000
> plot(0:20,pacbin, type="h",main="Distribuição B(20,0.3)-probabilidades
acumuladas", ylab="P(X<=x)")
```



c) Qual a probabilidade de X ser maior do que 5?

Abaixo são apresentadas duas maneiras diferentes de obter P(X > 5).

```
> 1-pbinom(5,20,0.3)
[1] 0.5836292

> pbinom(5,20,0.3,lower.tail=F)
[1] 0.5836292
```

O padrão é fazer ***lower.tail = T***, isto é calcular P(X ≤ x). Quando fazemos ***lower.tail = F*****,** o R retorna P(X > x).

d) Encontrando os quantis

Encontre o quantil de ordem 0,75 da distribuição B(20, 0.3). Vamos usar a função qbinom(q,size,prob).O argumento q é a ordem do quantil desejado.

```
> qbinom(0.75,20,0.3)
[1] 7
```

e) Verificando que a média e a variância de um B(20, 0.3) são μ = 6 e σ^2 = 4,2.A média e variância de uma distribuição binomial B(n,p) são $\mu = \sum_{x=0}^{n} xP(X = x) = np$ e $\sigma^2 = \sum_{x=0}^{n} (x-\mu)^2 P(X = x) = np(1-p)$. Para uma B(20,0.3), são 0.6 e 4.2. Verifique que isto é verdadeiro.

```
> x<-0:20                        # cria vetor com possíveis valores de X
> media <-sum(x*pbin)            # calcula o valor esperado da v.a.B(20,0.3)
> media
[1] 6

> var<-sum(((x-media)^2)*pbin)   # calcula a variância da v.a. B(20,0.3)
> var
[1] 4.2
```

10.2 - Distribuição de Poisson

Utilizamos as funções ***dbinom, pbinom*** e ***qbinom*** para obter probabilidades simples, acumuladas e quantis de uma distribuição binomial. Para a distribuição de Poisson vamos utilizar as funções ***dpois, ppois*** **e** ***qpois.*** Ao utilizar estas funções é necessário especificar o parâmetro lambda da distribuição. A seguir, apresentamos alguns exemplos.

Considere uma variável aleatória X com distribuição Poisson com parâmetro $\lambda = 10$.

a) Qual a probabilidade de X ser igual a 3?

```
 > dpois(3,lambda = 10)
[1] 0.007566655

> dpois(3,10)
[1] 0.007566655
```

b) Qual a probabilidade de X ser menor ou igual a 3?

```
> ppois(3,10)
[1] 0.01033605
```

c) Qual a mediana de X?

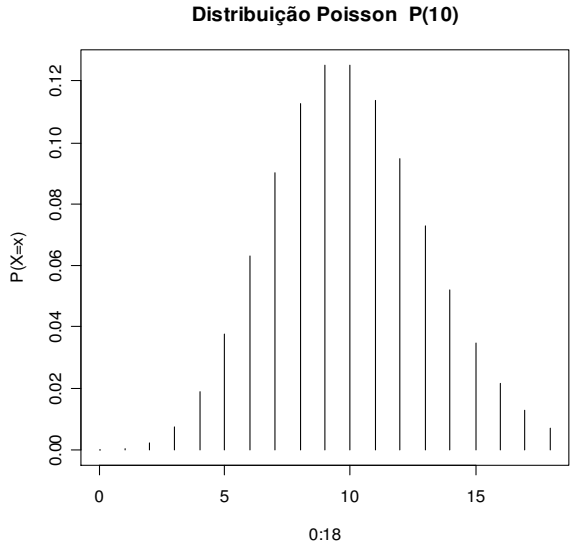
```
> qpois(0.5,10)
[1] 10
```

d) Qual o quantil de ordem 0.99?

```
> qpois(0.99,10)
[1] 18
```

e) Obtenha o gráfico da distribuição de X?

```
> plot(0:18, dpois(0:18,10), main="Distribuição Poisson P(10)", ylab =
"P(X=x)", type="h")
```



10.2.1 – Aproximação da Binomial Pela Poisson

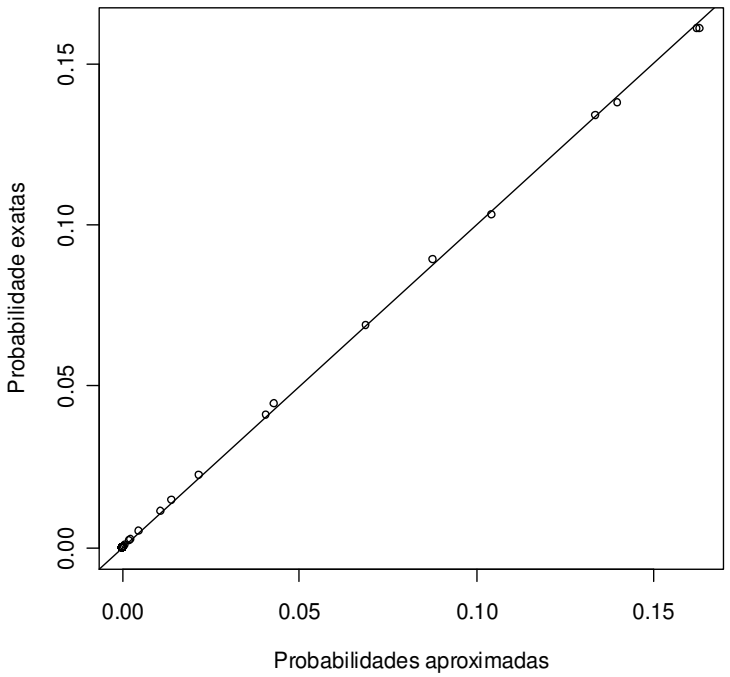
Quando o tamanho da amostra, n, é grande e a probabilidade de sucesso, p, é pequena, a distribuição Binomial pode ser aproximada por uma Poisson com $\lambda = np$.

Para n = 200 e p = 0.03 vamos ver que as probabilidades calculadas utilizando o modelo Binomial (exatas) e aquelas utilizando o modelo Poisson (aproximadas) são muito parecidas.

```
> # Obtendo as probabilidades pelo modelo binomial
> p1<-dbinom(0:200,200,0.03)
> # Obtendo as probabilidades pelo modelo Poisson
> p2<-dpois(0:200,200*0.03)
> p1<-round(p1,4) # arredonda os valores de p1 para 4 casas decimais
> p2<- round(p2,4)
> p1
[1] 0.0023 0.0140 0.0430 0.0879 0.1338 0.1622 0.1631 0.1398 0.1043 0.0688
[11]0.0407 0.0217 0.0106 0.0047 0.0020 0.0007 0.0003 0.0001 0.0000 0.0000
> p2
[1] 0.0025 0.0149 0.0446 0.0892 0.1339 0.1606 0.1606 0.1377 0.1033 0.0688
[11]0.0413 0.0225 0.0113 0.0052 0.0022 0.0009 0.0003 0.0001 0.0000 0.0000
```

A comparação fica mais fácil se fizermos o gráfico de p1 versus p2.

```
>plot(p1,p2,xlab="Probabilidades aproximadas",ylab="Probabilidade exatas")
> abline(a=0, b=1) # adiciona ao gráfico a reta com intercepto igual a zero e inclinação igual a 1.
```



10.3 - Exercícios

1) Para uma B(50, 0.40), calcule as probabilidades simples e acumuladas para x = 20, 25 e 30.

2) Construa os gráficos das distribuições de probabilidade binomial com n = 20 e probabilidades de sucesso iguais a 0.10, 0.30, 0.50, 0.70 e 0.90. Como o valor de p afeta a forma da distribuição? Em qual situação há maior simetria.

3) Para uma B(50, 0.40) encontre os quantis de ordem 25, 50 e 75.

4) Obtenha o gráfico da distribuição acumulada da Poisson com parâmetro igual a 10.

5) Para uma distribuição Poisson com parâmetro igual a 20 obtenha:

a)$P(X = x)$ para x = 15, 20 e 25

b) $P(X \leq x)$ para x = 15, 20 e 25

c) $P(X > x)$ para x = 18 e 30

d) Os quantis de ordem 0.25, 0.50, 0.75 e 0.99

e) Obtenha o gráfico da distribuição de probabilidade. Considere valores de x variando de 0 ao quantil de ordem 99.

Aula 11 – Distribuição Normal

11.1 - Usando as Funções *dnorm, pnorm* e *qnorm*

As funções ***pnorm*** e ***qnorm*** retornam as probabilidades acumuladas e os quantis da distribuição normal. A função ***dnorm*** retorna o valor da função densidade de probabilidade. No R, os parâmetros que definem a função densidade de probabilidade de uma distribuição Normal são a média e o desvio padrão. Portanto ao utilizarmos as funções ***dnorm, pnorm*** e ***qnorm*** é preciso especificá-los. Se não o fizermos, o R assume que eles são respectivamente iguais a 0 e 1 (distribuição normal padrão).

Para uma variável aleatória X com distribuição Normal com média 30 e desvio padrão 5, vamos ver como usar o R para responder as seguintes perguntas:

a) Qual o valor da função densidade quando X é igual a 30?

```
> dnorm(30, mean=20, sd=5)
[1] 0.01079819
```

b) Qual a probabilidade de X ser menor do que 25?

```
> pnorm(25, mean=20, sd=5)
[1] 0.8413447
```

c) Qual a probabilidade de x ser maior do que 32?

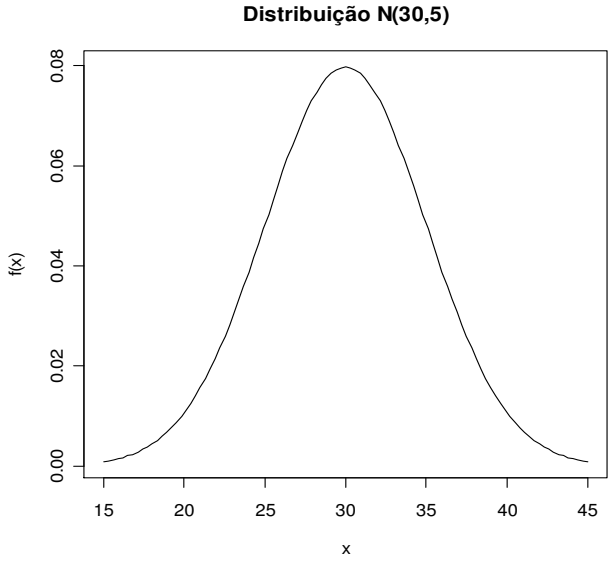
```
> pnorm(32, mean = 20, sd = 5, lower.tail=F)
[1] 0.008197536
```

d) Qual o valor de X que deixa 75% dos valores abaixo dele?

```
> qnorm(0.75, mean=20, sd=5)
[1] 23.37245
```

e) Obtenha os gráficos da função densidade de probabilidade e da função distribuição acumulada.

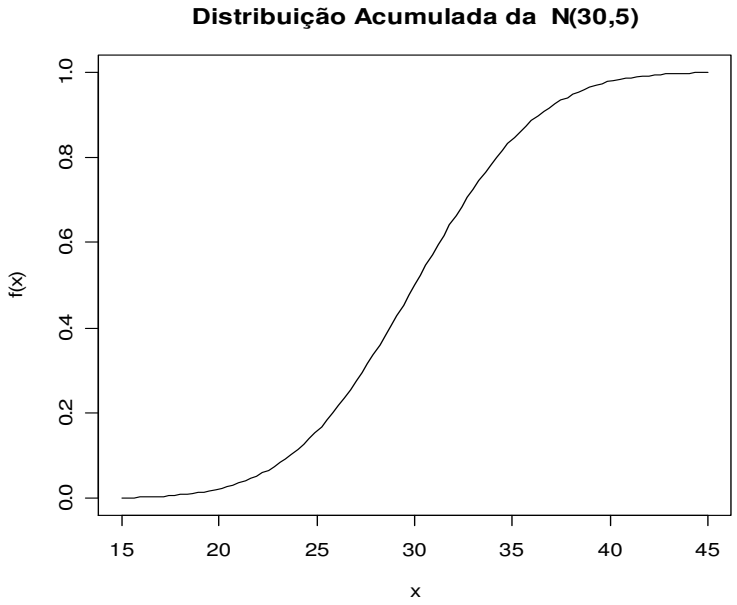
```
> plot(function(x) dnorm(x,30,5), 15, 45, ylab = "f(x)", main =
"Distribuição N(30,5)")
```



Com o comando acima fazemos o gráfico da função densidade de probabilidade para valores de x de 15 a 45.

Para fazermos o gráfico da função distribuição acumulada basta substituir ***dnorm(x,30,5)*** por ***pnorm(x,30,15)***, como mostrado abaixo..

```
> plot(function(x) pnorm(x,30,5),15,45,ylab = "f(x)",main = "Distribuição
Acumulada da N(30,5)")
```



11.2 - Verificando a Suposição de Normalidade

Em muitas análises estatísticas supomos que os dados são uma amostra de uma distribuição Normal. Dois métodos gráficos são úteis para verificar se esta suposição é válida.

11.2.1 - Histograma com Distribuição Normal Ajustada

Constrói-se o histograma de densidades e adiciona a ele a função densidade normal com média e desvio padrão iguais aos observados na amostra.

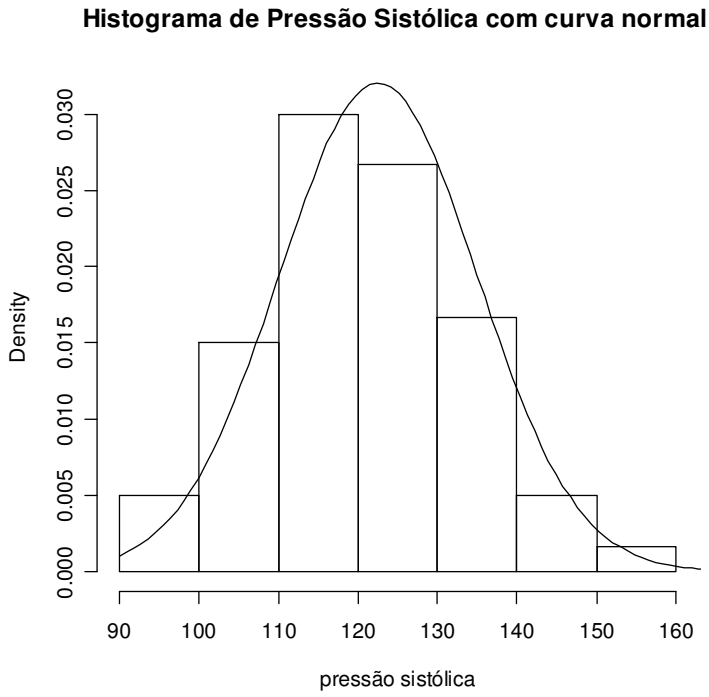
Vamos utilizar o seguinte conjunto de dados, relativos à pressão sistólica de uma amostra de mulheres adultas, para mostrar como obter no R o histograma com a curva normal ajustada.

```
>ps<-c(94,98,100,102,104,108,108,108,110,110,110,110,112,114,114,
116,116,116,116,118,118,118,118,118,118,120,120,120,120,120,122,
122,124,124,124,128,128,128,128,128,128,130,130,130,130,130,132,
132,132,134,136,138,138,140,140,140,142,142,146,152)

>hist(ps,freq=F,ylim=c(0,dnorm(mean(ps),mean(ps),sd(ps))),
main="Histograma de Pressão Sistólica com curva normal ",xlab="pressão
sistólica")
```

O comando acima constrói o histograma de densidades (Observe que ***freq = F***). O argumento ***ylim*** estabelece os limites dos valores do eixo Y. O limite inferior é zero e o superior é dado pelo valor da função densidade calculada quando a pressão sistólica é igual á média amostral.

```
> plot(function(x) dnorm(x, mean(ps), sd(ps)), 90, 180, add=T)
```

O comando acima adiciona ao histograma o gráfico da função densidade de probabilidade Normal. O comando ***add = T*** indica que o segundo gráfico (plot) dever ser adicionado ao gráfico anterior (hist). Fizemos os valores de pressão sistólica varia entre 90 e 180, o seu intervalo de variação no histograma.



11.2.2 - Gráfico dos Quantis ou *qqplot*

A idéia deste gráfico é comparar os valores observados na amostra com os valores esperados caso a distribuição fosse Normal. Os valores observados e esperados são plotados num diagrama de dispersão. Se o modelo é adequado espera-se que os pontos estejam próximos de uma reta.

Passos para construção do qqplot:

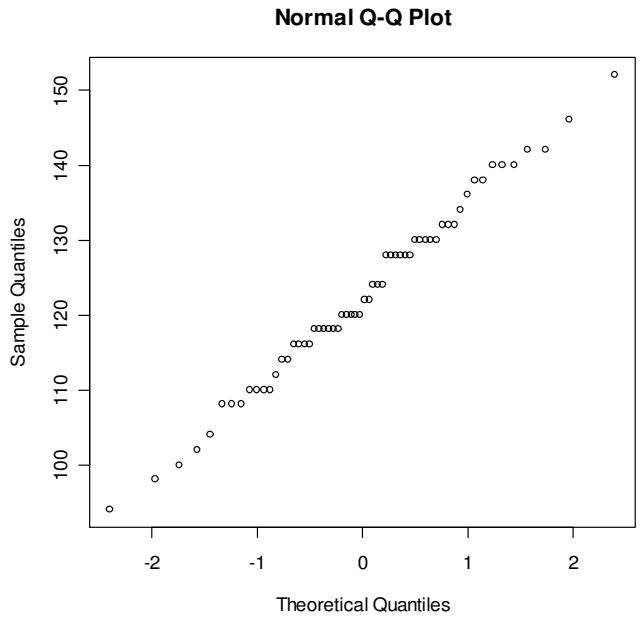
a) Para cada valor observado x, estima-se $P(X \leq x)$ pela proporção amostral de valores menores ou iguais a x. Estas estimativas são denominadas probabilidades empíricas.

b) Assuma que o modelo normal é adequado para descrever os dados e que sua média e desvio padrão são iguais aos observados na amostra. Usando as probabilidades empíricas obtenha os quantis correspondentes da distribuição Normal.

c) Construa um diagrama de dispersão colocando no eixo Y os valores observados e no eixo X os quantis obtidos no item b.

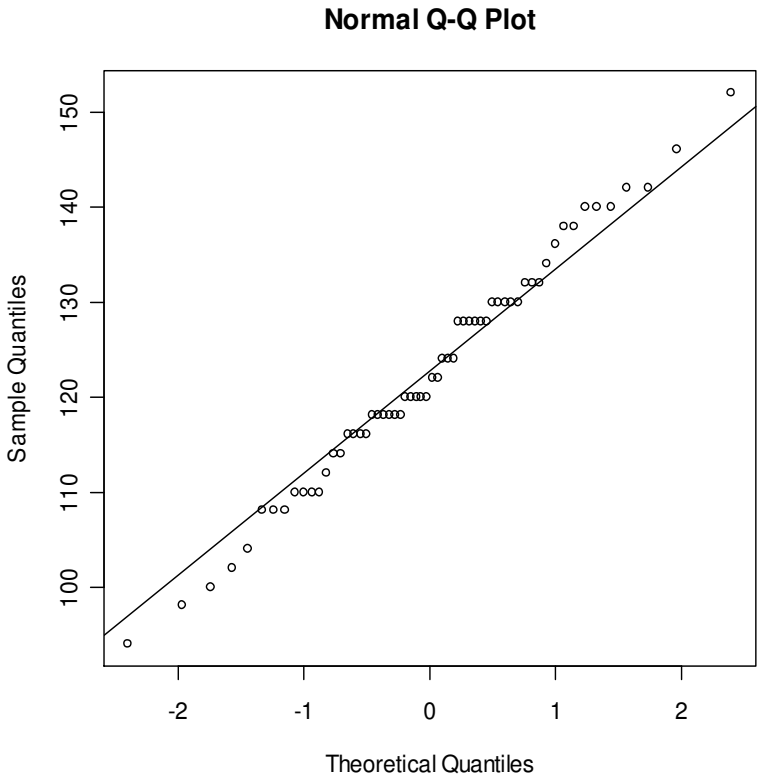
A função ***qqnorm*** do R constrói qqplot para a distribuição Normal. Veja como é fácil utilizá-la.

```
> qqnorm(ps)
```



Para facilitar a análise do gráfico adicione uma reta ao grafico utilizado o comando ***qqline( ).***

```
> qqline(ps)
```



11.3 - Exercícios

1) Para uma distribuição Normal padrão obtenha:

a) P(-3 < X < 3), P(-2 < X < 2) e P(-1 < X < 1)

b) P(X < 0)

c) Os quantis de ordem 0.025, 0.05, 0.10, 0.25 e 0.50.

d) O gráfico da função densidade de probabilidade. Faça x variar de -3.5 até 3.5.

2) Para uma distribuição Normal com média 40 e desvio padrão 8, obtenha:

a) $P(X < x)$ para x = 25, 40 e 50

b) $P(X > x)$ para x = 30 e 45.

c) Os quantis de ordem 0.025, 0.05, 0.25 e 0.50.

d) O gráfico da função distribuição acumulada. Faça os valores de x variarem entre a média – 3 desvios padrão (16) e a média + 3 desvios padrão (64).

3) Considere uma distribuição binomial B(n,p). Quando $np > 5$ e $n(1-p)$ são maiores do que 5. As probabilidades binomiais podem ser aproximadas pelas probabilidades obtidas pelo modelo Normal com média $\mu = np$ e variância $\sigma^2 = np(1-p)$. Seja X a variável aleatória B(n,p) e Y a variável aleatória Normal com média $\mu = np$ e variância $\sigma^2 = np(1-p)$. As aproximações para as probabilidades são obtidas como segue.

a) $P(X = x) = P(x - 0,5 < Y < x + 0,5)$

b) $P(X \leq x) = P(Y < x + 0,5)$

c) $P(X < x) = P(Y < x - 0,5)$

d) $P(X \geq x) = P(Y > x - 0,5)$

e) $P(X > x) = P(Y > x + 0,5)$

Obtenha as probabilidades $P(X = x)$ pelo modelo binomial (probabilidades exatas) e as probabilidades aproximadas pelo modelo Normal para n = 20 e p = 0,4 e valores de x variando de 0 a 20. Compare os valores obtidos pelos 2 modelos.

4) Para os seguintes conjuntos de dados construa o histograma com curva normal ajustada e o qqplot para o modelo normal. O modelo Normal é satisfatório?

Conjunto de dados A

53, 170, 5, 113, 474, 67, 108, 97, 196, 163, 19, 44, 6, 167, 141, 12, 11, 66, 357, 48, 88, 23, 14, 64, 37, 217, 272, 28, 21, 76, 14, 68, 58, 351, 47, 8, 285, 98, 22, 142, 77, 187, 25, 48, 6, 178, 52, 155, 151, 13.

Conjunto de dados B (o ponto é o separador decimal)

21.44, 37.55, 24.93, 31.06, 21.97, 20.57, 11.98, 33.93, 30.03, 21.03, 32.37, 27.87,

35.67, 30.01, 27.25, 34.01, 18.32, 24.78, 35.51, 21.13, 21.38, 48.58, 11.35, 31.33,

29.57, 20.05, 23.09, 25.28, 25.78, 49.42, 18.24, 20.81, 44.87, 33.76, 27.55, 15.60,

43.96, 28.57, 23.82, 24.29, 43.15, 35.10, 38.47, 14.63, 15.77, 24.79, 37.77, 37.42,

22.61, 56.14, 38.62, 25.80, 30.06, 27.32, 25.64, 43.33, 29.56, 29.83, 10.18, 16.06,

35.57, 27.98, 38.73, 28.60, 51.41, 29.00, 17.85, 22.01, 20.69, 40.30, 37.06, 24.43,

29.35, 20.07, 34.06, 38.65, 39.41, 47.64, 42.12, 22.47, 42.51, 11.43, 33.28, 44.79,

10.84, 43.06, 36.71, 20.72, 25.51, 20.11, 20.16, 36.74, 44.82, 33.21, 14.87, 27.68,

30.30, 24.93, 23.65, 26.21

Aula 12 – Geração de Variáveis Aleatórias

Em geral, nas análises estatísticas assumimos um modelo de probabilidade para a variável aleatória de interesse. Utilizando métodos estatísticos podemos verificar se uma distribuição de probabilidade se ajusta bem a um conjunto de dados. Exemplos disto são os gráficos vistos na seção anterior para verificar se a suposição de normalidade é adequada.

Nesta aula vamos tratar de simulação de variáveis aleatórias, isto é dada uma variável aleatória X com certa distribuição de probabilidade, por exemplo, distribuição Normal, como simular uma amostra de X?

Considere uma distribuição N(0,1). Como podemos gerar uma amostra de 10 observações desta distribuição?

Dado um valor da probabilidade acumulada, podemos obter o quantil correspondente. A probabilidade acumulada assume valores no intervalo entre 0 e 1. Então ao acaso escolha um valor entre 0 e 1. A partir deste valor obtenha o quantil da Normal com média 0 e desvio padrão 1. Desta forma você obtém uma observação da distribuição N(0,1).

Para gerar 10 observações, selecione ao acaso 10 valores no intervalo entre 0 e 1 e para cada um deles obtenha o quantil correspondente. Como fazer isto no R?

Para gerar uma sequência aleatória de valores entre 0 e 1 vamos utilizar a função ***runif***.

```
> u <-runif(10, min = 0, max = 1) #gera 10 valores entre 0 e 1
> u
 [1] 0.58391707 0.73240660 0.08024068 0.43739791 0.63956568 0.96185502
 [7] 0.45585754 0.92816454 0.88680702 0.91843008
```

Para cada valor de u obtenha o quantil correspondente da distribuição N(0,1)

```
> x<-qnorm(u,0,1)
> x
 [1]  0.2119246  0.6201078 -1.4034545 -0.1575698  0.3572981  1.7726305
 [7] -0.1108755  1.4622566  1.2097210  1.3945889
```

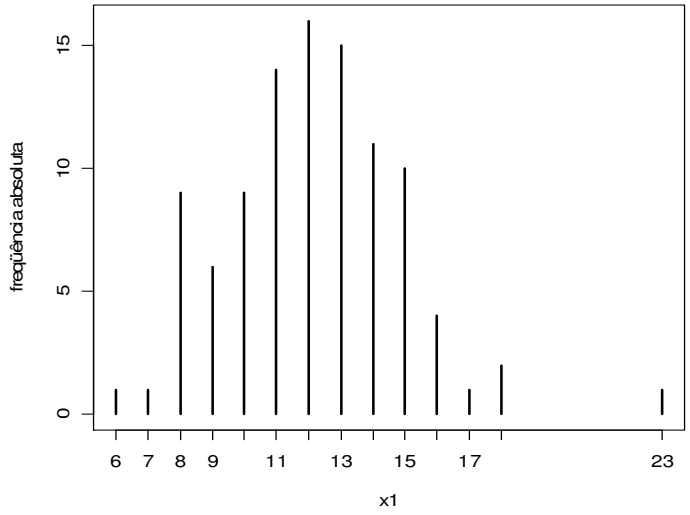
Existe uma maneira mais simples de gerar valores de uma distribuição normal no R: utilizar a função ***rnorm***. Para gerar 10 observações de uma variável aleatória normal com média 0 e desvio padrão 1, faça:

```
> xnorm<-rnorm(10, mean = 0, sd =1)    # gera 10 valores de uma N(0,1) e
armazena em xnorm.
> xnorm
 [1]  1.8590629 -1.5991701 -1.3867766 -1.9763305 -0.8249046 -0.6251096
 [7]  0.8172476 -0.9039651  0.9888936 -0.4423563
```

O R possui funções para gerar observações de várias outras distribuições de probabilidade. Para gerar valores das distribuições binomial e Poisson utiliza-se as funções ***rbinom*** e ***rpois***. A letra r que inicia os nomes das funções é a primeira letra da palavra random, cuja tradução é aleatório. Abaixo seguem alguns exemplos de utilização destas funções:

1) Gerar 100 valores de uma distribuição B(30,0;4)

```
> x1<-rbinom(100,30,0.4)
> x1
  [1]  9 11  9 14 15 14 10 13 16 15 13 14 13 18  8 14  8 15 13 12 13 15 13 16  8
 [26] 13 12 23 14 11 11 12 10  8 12 13 18  9 10 10 12  6 11 13  7 12 13  8 14 16
 [51] 12 10 15 12 15 11  8 11 15  8 11 14 14 12 12 11 11 11 17  8 10  9 15 13 14
 [76] 11 10 13 12 14 15 12 13  9 12 15 10  9 11 11 16 13 12 11 12 10 14 13 12  8

> tx1<-table(x1)
> tx1
x1
 6  7  8  9 10 11 12 13 14 15 16 17 18 23
 1  1  9  6  9 14 16 15 11 10  4  1  2  1

> plot(tx1, ylab="freqüência absoluta") #gráfico da distribuição de
freqüências de x1
```

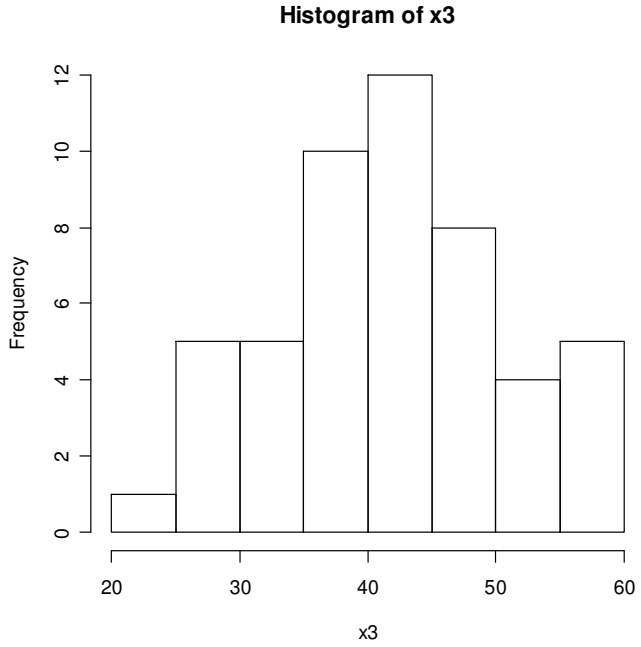


2) Gerar 50 valores de uma distribuição P(5)

```
> x2<-rpois(50,5)
> x2
 [1]  9  6  3  5  4  5  7  6  3  7  5  6  8  4  4  4  6  1  3  2 11 12  5  1  4
[26]  1  3  3  8  7  7  5  2  8  3  3  6  5  5  5  6  3  5  6  6  7 10  6  7  2
> table(x2)
x2
 1  2  3  4  5  6  7  8  9 10 11 12
 3  3  8  5  9  9  6  3  1  1  1  1
```

3) Gerar 50 valores de uma distribuição N(40,8)

```
> x3<-rnorm(50,40,8)
> x3
 [1] 46.03488 50.51355 45.27353 49.74057 47.59037 25.98789 51.01137 36.52534
 [9] 34.37448 47.71775 39.80518 44.21646 30.57270 46.82816 56.18880 45.18143
[17] 39.82137 35.68909 40.13909 46.11032 42.61036 41.24322 55.92718 25.10701
[25] 56.69804 44.32670 52.86171 42.18448 43.71929 36.08772 32.54297 25.94555
[33] 29.49525 37.74751 42.97423 41.00715 34.71615 55.07324 43.10573 30.36278
[41] 55.72475 39.88075 44.50527 42.66683 27.87353 23.88561 51.44608 36.37026
[49] 39.88015 38.18624
> hist(x3) # Histograma de x3
```



12.1 - Exercícios

1) Gere uma amostra de 100 observações de uma distribuição N(50,10) e construa seu histograma.

2) Gere uma amostra de 50 observações de uma distribuição P(3) e obtenha a sua distribuição de freqüência.

Aula 13 – Teorema Central do Limite

O teorema central do limite é um resultado muito importante para a realização de inferências estatísticas. Ele nos diz qual é a distribuição de probabilidade de $\overline{X}$, a média amostral.

13.1 - Teorema Central do Limite

Considere uma população identificada por uma variável aleatória X cuja média $\mu = E(X)$ e cuja variância $\sigma^2 = VAR(X)$ são conhecidas. Considere todas as possíveis amostras de tamanho n retiradas com reposição desta população. Então, à medida que n cresce, a distribuição de $\overline{X}$ aproxima-se de uma distribuição normal com média $\mu = E(\overline{X})$ e $VAR(\overline{X}) = \frac{\sigma^2}{n}$.

Em geral assumimos que para n > 30 a aproximação da distribuição de $\overline{X}$ pela distribuição normal é satisfatória. A velocidade da convergência para a distribuição normal depende da forma da distribuição de X. Ela é mais rápida quando a distribuição de X é simétrica. Quando X tem distribuição normal, a distribuição de $\overline{X}$ é exatamente normal qualquer que seja o tamanho da amostra.

A seguir vamos construir uma função no R para visualizamos o resultado acima quando a variável X tem distribuição Normal. O funcionamento da função é descrito acima para a situação em que a média é 30 e variância é 25 e o tamanho de amostra é 5. Os passos da função são os seguintes:

1) gere uma amostra aleatória de tamanho 5 uma distribuição normal com média 30 e variância 25.
2) calcule para o 5 valores gerados a média amostral e armazene o valor no vetor de nome media
3) repita os passos 1 e 2 um número grande vezes, por exemplo 1.000. Com isto teremos 1.000 valores de $\overline{X}$, o bastante para termos uma idéia da distribuição de $\overline{X}$.

4) obtenha o histograma dos 1.000 valores de $\overline{X}$ e calcule a sua média e variância.

5) repita os passos 1 a 4 para tamanhos de amostras n = 10,15,20,30,50.

Os passos 1 a 4 acima foram implementados na função ***tclnormal,*** cujos argumentos são: ***n,*** o tamanho da amostra, ***nsimul,*** o número de valores de $\overline{X}$ gerados; ***mu,*** a média de X e ***sigma***, o desvio padrão de X.

```
>tclnormal<-function(n,nsimul,mu,sigma)
{
media<-rep(0,nsimul)
for(i in 1:nsimul)
{
x<-rnorm(n,mu,sigma)
media[i] <- mean(x)
}
mmedia<-mean(media)
varmedia<-var(media)
sdmedia<-sd(media)
hist(media,xlim=c(mu-3.5*sigma/sqrt(5),  mu+3.5*sigma/sqrt(5)),  xlab  =
expression(bar(X)),main=paste("Histograma,n=",
deparse(substitute(n)),sep=" "))
resultado<-list(mmedia=mmedia,varmedia=varmedia,sdmedia=sdmedia)
# mmedia é a media das médias amostrais
# varmedia é a variância das médias amostrais
return(resultado)
}
```

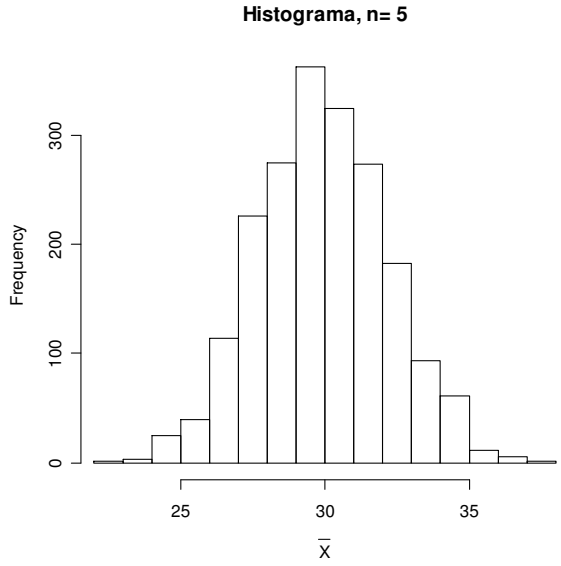
Não se preocupe em entender a sintaxe da função *tclnormal* mostrada acima. Isto vai além dos objetivos deste texto. Utilize-a como você faz com outras funções do R.

13.1.1 - Utilizando a Função *tclnormal*

Carregue a função ***tclnormal*** copiando e colando no R o texto acima, em negrito. Depois execute a função para n = 5, nsimul = 2000, mu = 30 e sigma = 5, como mostrado abaixo. A função retorna a média (mmedia), a variância (varmedia) e desvio padrão (dpmedia) de $\overline{X}$ obtidos através das simulações (isto é a média, a variância e o desvio padrão dos 2.000 valores de $\overline{X}$) e o seu histograma.

```
> tclnormal(5,2000,30,5)
$mmedia
[1] 29.99947
$varmedia
[1] 5.08578
$sdmedia
[1] 2.255167
```

Compare o valor de mmedia e varmedia, obtidos por simulação, com aqueles dados pelo teorema central do limite $\mu = E(\overline{X}) = 30$ e $VAR(\overline{X}) = \frac{\sigma^2}{n} = \frac{25}{5} = 5$.Como esperado, eles são muito próximos. Observe que, como esperado, o histograma sugere uma distribuição Normal para $\overline{X}$.



Vimos que a variância de $\overline{X}$ diminui com o aumento do tamanho da amostra, isto é o valor de $\overline{X}$, a média amostral, aproxima-se cada vez mais da média da população. Para visualizarmos este resultado vamos executar a função *tclnormal* para vários tamanhos de amostras: 10,15,20,30 e 50 e organizar os histogramas numa única janela.

Copie e cole os comandos abaixo no R.

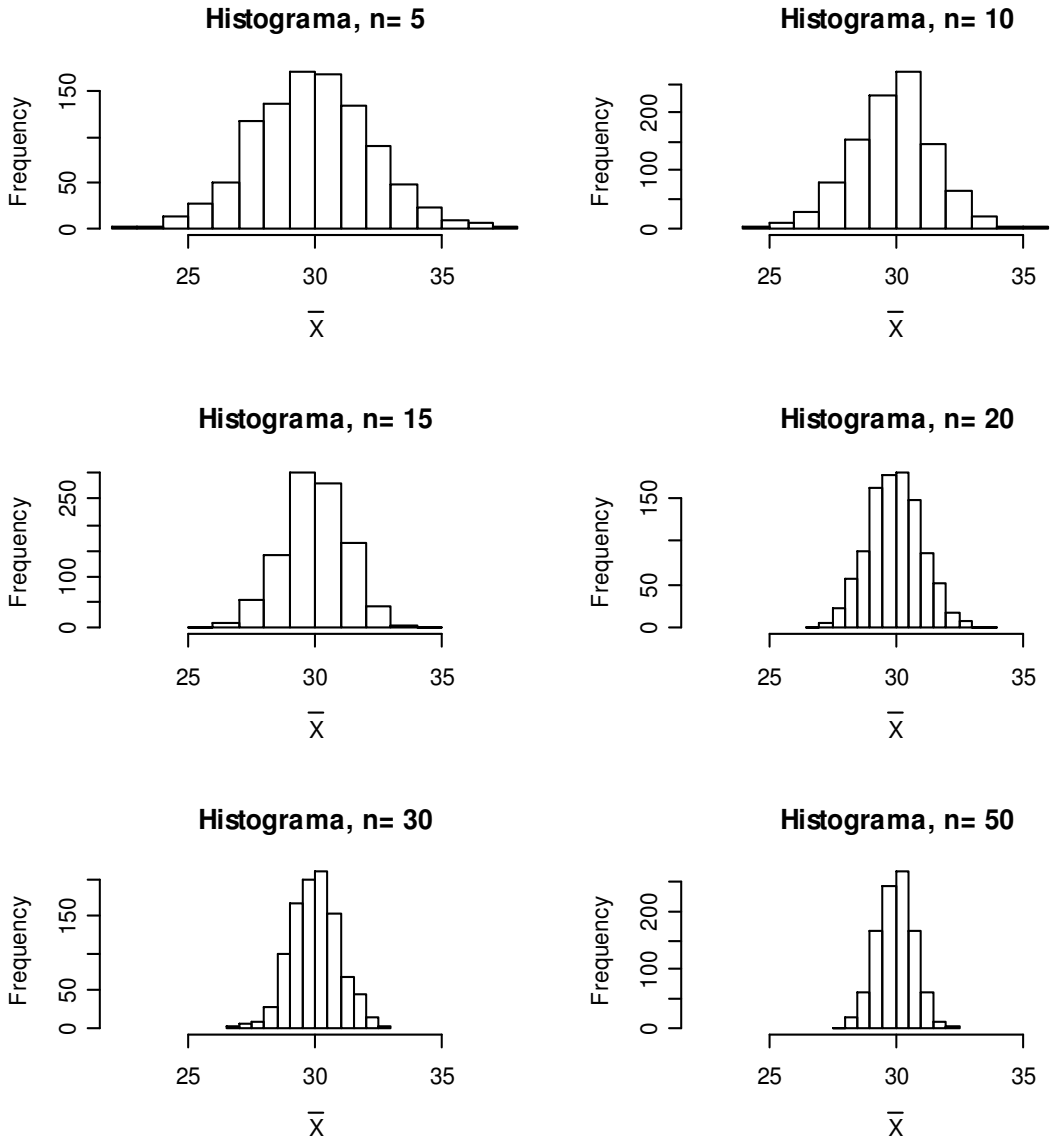
```
>par(mfrow=c(3,2)) #divide a janela em 6 células (2 linhas e 3 colunas)
>tclnormal(5,1000,30,5)
>tclnormal(10,1000,30,5)
>tclnormal(15,1000,30,5)
```

```
>tclnormal(20,1000,30,5)
>tclnormal(30,1000,30,5)
>tclnormal(50,1000,30,5)
```

O R retornará valores parecidos com os apresentados no quadro abaixo:

n	5	10	15	20	30	50
mmedia	29.98506	30.00992	29.99555	29.98595	29.99656	29.98162
varmedia	4.965183	2.610514	1.646883	1.280810	0.8035554	0.5165981
dpmedia	2.228269	1.615708	1.283309	1.131729	0.8964125	0.7187476

Observe que independente do valor de n, o valor de mmdia é próximo de 30 e que a variância diminuir com o valor de n. Este resultado pode ser visualizado nos gráficos seguintes:



13.1.2 - População Poisson

Considere uma população especificada por uma distribuição de Poisson com parâmetro λ . Lembre-se que para a distribuição de poisson E(X) = VAR(X) = λ. Neste caso vamos utilizar a função *tclpoisson,* cuja sintaxe é dada abaixo.

```
>tclpois<-function(n,nsimul,lambda)
{
media<-rep(0,nsimul)
for(i in 1:nsimul)
{
x<-rpois(n,lambda)
media[i] <- mean(x)
}
mmedia<-mean(media)
varmedia<-var(media)
sdmedia<-sd(media)
hist(media,  xlim=c(lambda-3.5*sqrt(lambda/5),lambda+3.5*sqrt(lambda/5)),
xlab=expression(bar(X)),main=paste("Histograma,n=",deparse(substitute(n))
,sep=" "))
resultado<-list(mmedia=mmedia,varmedia=varmedia,sdmedia=sdmedia)
# mmedia é a media das médias amostrais
# varmedia é a variância das médias amostrais
# sdmedia é o desvio padrão das médias amostrais
return(resultado)
}
```

13.1.3 - População Bernoulli – distribuição amostral da proporção

Podemos ver a proporção amostral com sendo a média de uma variável assumindo o valor 1, se ocorre sucesso, e o valor 0, se ocorre fracasso. Portanto, a distribuição amostral da proporção pode ser obtida como caso particular do teorema central do limite: a proporção amostral $\hat{p}$ possui aproximadamente distribuição normal com média p e variância $\frac{p(1-p)}{n}$.

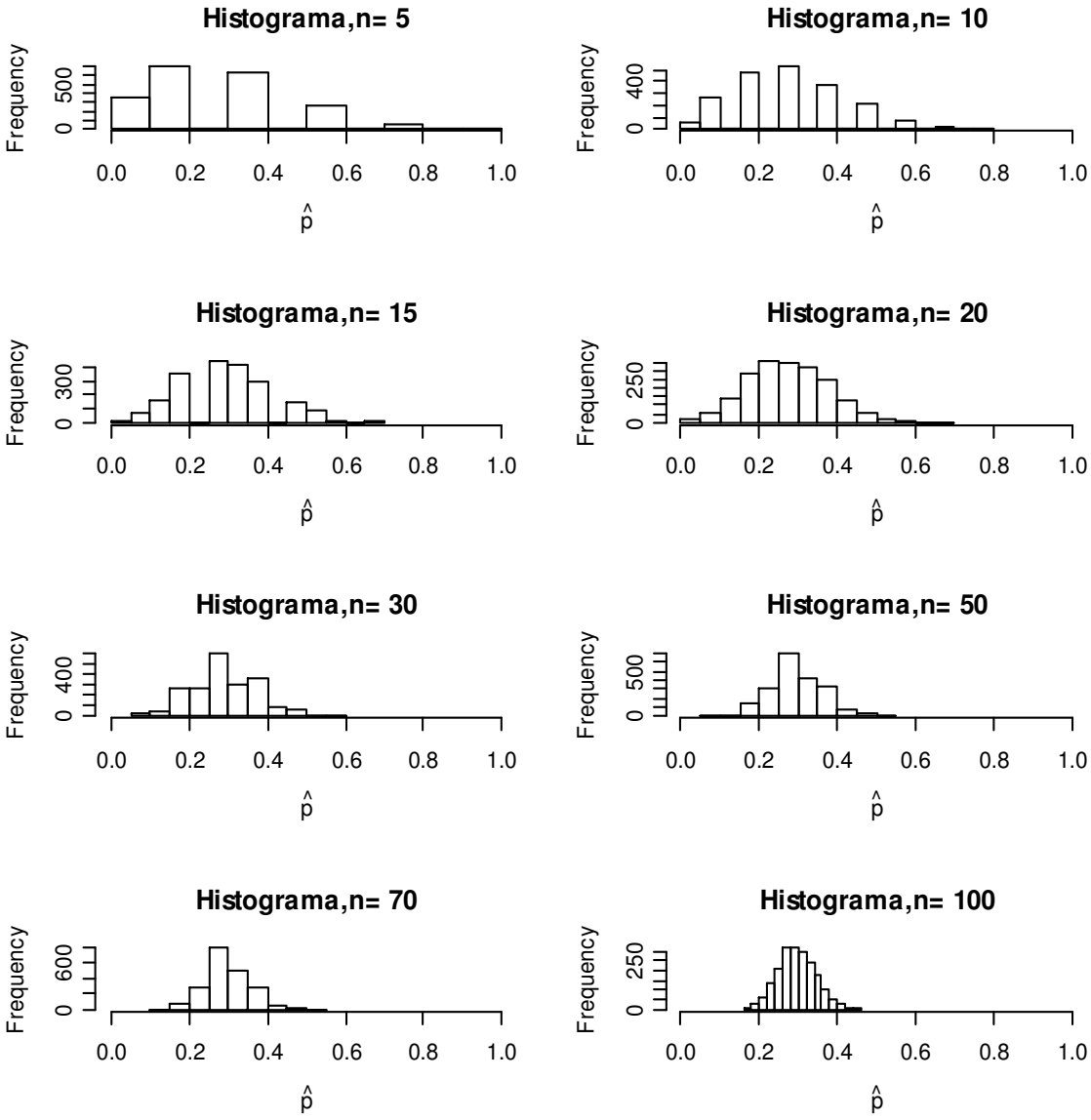
Assim como construímos a função ***tclpois*** e ***tclnormal*** obtemos a função ***tclbern***. Esta função considera uma população especificada por uma variável aleatória X com distribuição de Bernouli com probabilidade de sucesso p, o terceiro argumento da função.

```
tclber<-function(n,nsimul,p)
{
media<-rep(0,nsimul)
for(i in 1:nsimul)
{
x<-rbinom(n,1,p)
media[i] <- mean(x)
}
mmedia<-mean(media)
varmedia<-var(media)
sdmedia<-sd(media)
hist(media,xlim=c(0,1),xlab=expression(hat(p)),main=paste("Histograma,n="
,deparse(substitute(n)),sep=" "))
resultado<-list(mmedia=mmedia,varmedia=varmedia,sdmedia=sdmedia)
# mmedia é a media das médias amostrais
# varmedia é a variância das médias amostrais
# sdmedia é o desvio padrão das médias amostrais
return(resultado)
}
```

Execute os comandos abaixo:

```
>par(mfrow=c(4,2))
>tclber(n = 5,nsimul = 2000, p= 0.3)
>tclber(10,2000,0.3)
>tclber(15,2000,0.3)
>tclber(20,2000,0.3)
>tclber(30,2000,0.3)
tclber(50,2000,0.3)
>tclber(70,2000,0.3)
>tclber(100,2000,0.3)
```

Você deverá observa resultados similares aos mostrados na figura seguinte:



Observe que a distribuição aproxima-se da distribuição normal à medida que n cresce.

13.2 – Exercícios

1) Execute os comandos abaixo e comente os resultados.

```
>par(mfrow=c(3,2))
>tclpois(n = 5,nsimul = 1000, lambda = 10)
>tclpois (10,1000,10)
>tclpois (15,1000,10)
>tclpois (20,1000,10)
>tclpois (30,1000,10)
>tclpois (50,1000,10)
```

2) Em geral, assume-se que a aproximação da distribuição da proporção amostral pela Normal é satisfatória se np > 5 e n(1-p) > 5. Para tamanho de amostra 50 e valores de p iguais a 0.1, 0.2, 0.3, 0.5, 0.7, 0.8 e 0.9 execute a função tclbern1, a função anterior com os títulos dos gráficos modificados. Verifique como o valor de p influencia a aproximação da distribuição da proporção amostral pela distribuição Normal. Em qual situação a distribuição se parece mais com a distribuição Normal?

```
tclber1<-function(n,nsimul,p)
{
media<-rep(0,nsimul)
for(i in 1:nsimul)
{
x<-rbinom(n,1,p)
media[i] <- mean(x)
}
mmedia<-mean(media)
varmedia<-var(media)
sdmedia<-sd(media)
hist(media,xlim=c(0,1),xlab = expression(hat(p)),main=paste("Histograma,
n =50 ,p = ",deparse(substitute(p)),sep=" "))
resultado<-list(mmedia=mmedia,varmedia=varmedia,sdmedia=sdmedia)
# mmedia é a media das médias amostrais
# varmedia é a variância das médias amostrais
# sdmedia é o desvio padrão das médias amostrais
return(resultado)
}
```

Aula 14 – Distribuição *t* de Student

Na aula anterior vimos que quando X tem distribuição Normal com média μ e variância σ^2, a distribuição da média amostral $\overline{X}$ é exatamente normal com média μ e variância σ^2/n. Logo a variável padronizada $Z = \frac{\overline{X} - \mu}{\sigma}$ tem distribuição N(0,1).

Entretanto quando fazemos inferências para a média populacional E(X) = μ, geralmente desconhecemos o valor do desvio padrão populacional σ. Então estimamos σ pelo desvio padrão amostral $s = \sqrt{\frac{\sum_{1=1}^{n}(x_i - \overline{x})^2}{n-1}}$. Se substituirmos σ por s em $Z = \frac{\overline{X} - \mu}{\sigma}$, como fica a distribuição da nova variável $T = \frac{\overline{X} - \mu}{S}$?

A média de T continua sendo zero, pois quando multiplicamos uma variável por uma constante (neste caso σ/S) sua média (neste caso, 0) é multiplicada pela constante. Entretanto o desvio padrão de T é maior do que o desvio padrão de Z, igual a 1.. A variação de Z provém da variação de $\overline{X}$, a média amostral enquanto a variação de T provém da variação de $\overline{X}$ e S. Quanto menor for o tamanho de amostra maior deve ser a incerteza introduzida ao substituir σ por S, pois S deve aproximar-se de σ com o aumento de n. Por esta razão a distribuição de probabilidade de T depende do tamanho da amostra, isto é, para cada tamanho de amostra temos uma distribuição diferente. Ela é conhecida como distribuição t de Student e é indexada por uma quantidade chamada graus de liberdade, igual a n – 1 (tamanho de amostra -1).

Podemos usar o R para calcular probabilidades e quantis para uma distribuição t de Student. Para isto vamos utilizar as funções ***pt*** e ***qt.***

Exemplo: Considere uma variável aleatória com distribuição t com 10 graus de liberdade a) qual a probabilidade de T ser menor ou igual a -2?

```
> pt(-2,10)
[1] 0.03669402
```

b) Qual o valor desta probabilidade para a distribuição N(0,1)?

```
> pnorm(-2,0,1)
[1] 0.02275013
```

Observe que o valor é muito maior para a distribuição T do que para a N(0,1). Isto é esperado, dado a maior incerteza associada a T do que a Z.

c) Qual o quantil de ordem 0.9 da distribuição t de Student?

```
> qt(0.9,10)
[1] 1.372184
```

d) Qual o quantil de ordem 0.9 da distribuição N(0,1)?

```
> qnorm(0.90,0,1)
[1] 1.281552
```

Compare os valores. Porque eles são tão difeentes?

e) Obtenha agora o quantil de ordem 0.025 para distribuição t com graus de liberdade variando de 5 a 100 em intervalos de tamanhos 5.

```
> qt(0.025,seq(5,100,5))
[1] -2.570582 -2.228139 -2.131450 -2.085963 -2.059539 -2.042272 -2.030108
[8] -2.021075 -2.014103 -2.008559 -2.004045 -2.000298 -1.997138 -1.994437
[15] -1.992102 -1.990063 -1.988268 -1.986675 -1.985251 -1.983972
```

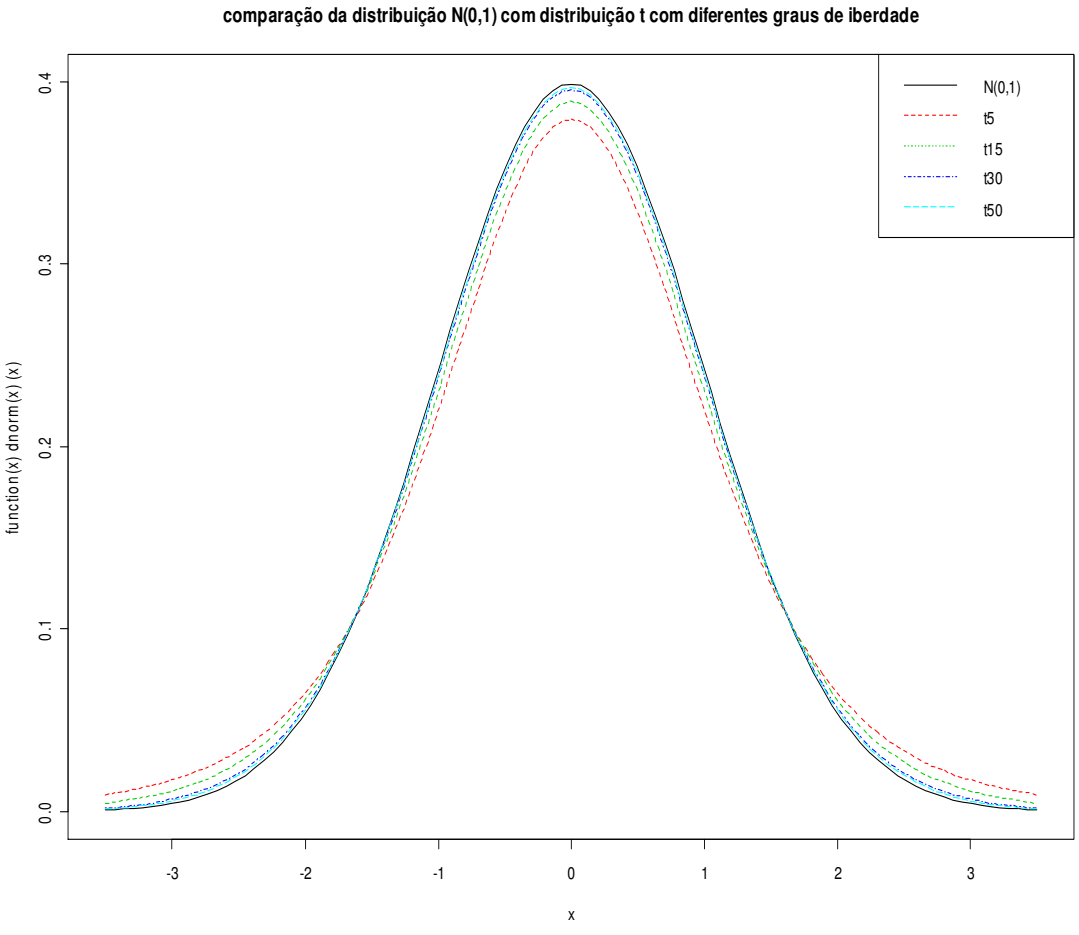
f) Compare os valores acima com o valor do quantil 0.025 da N(0,1).

```
> qnorm(0.025,0,1)
[1] -1.959964
```

Observe que os valores do quantil 0,025 da distribuição t de Student convergem para o quantil 0,025 da N(0,1) com o aumento dos graus de liberdade.

g) Compare a função densidade de probabilidade da N(0,1) com as funções densidade de probabilidade de uma variável aleatória T de Sudent com graus de liberdade iguais a 5, 15, 30 e 50.

```
>plot(function(x)dnorm(x),-3.5,3.5,col=1,lty=1,main="comparação da distribuição N(0,1) com distribuição t com diferentes graus de iberdade")
>plot(function(x)dt(x,5),-3.5,3.5,add=T,col=2, lty=2)
>plot(function(x)dt(x,10),-3.5,3.5, add=T,col=3, lty=2)
>plot(function(x)dt(x,30),-3.5,3.5, add=T,col=4, lty=4)
>plot(function(x)dt(x,50),-3.5,3.5, add=T,col=5, lty=5)
legend("topright",legend = c("N(0,1)","t5","t15","t30","t50"),lty= 1:5, col=1:5)
```



Observe que com o aumento dos graus de liberdade a distribuição T converge para a N(0,1).

Entendendo o comando ***legend*** acima**.**

- “topright” indica que a legenda deve ser posicionada no canto superior direito do gráfico.
- legend = c(“N(0,1)”,”t5”,”t15”,”t30”,”t50”) é o texto da legenda.
- lty = 1:5 indica o tipo de linha utilizado para representar cada curva no gráfico.Por exemplo lty =1 indica linha contínua.
- col = 1:5 indica a cor utilizada para representar cada curva no gráfico.Por exemplo col = 1 para cor preta.

14.1 – Exercícios

1) Para uma variável aleatória T com distribuição t de sudent com 15 graus de liberdade obtenha:

a) P(T <2), P(T > 3)

b) Os quantis de ordem 0.05, 0.10 e 0.25.

Aula 15 – Inferência Para Média e Proporção – caso de uma população

Nesta aula vamos aprender, através de exemplos, como obter no R intervalos de confiança e testes de hipóteses para as seguintes situações**:**

1) Uma média populacional.

2) Uma proporção populacional.

15.1 - - Inferência Para uma Média Populacional

Exemplo: Uma indústria farmacêutica produz comprimidos de um antiácido por dia. Este medicamento é produzido com a especificação de que teor médio de carbonato de sódio em cada comprimido seja igual a 400 mg. O órgão responsável pela fiscalização de medicamentos selecionou ao acaso e analisou uma amostra de 20 comprimidos entre os comprimidos produzidos pela empresa, encontrando os seguintes valores:

403.43 382.84 401.85 396.40 389.87 397.46 402.12 406.28 404.48 411.89

402.09 407.59 391.65 389.81 393.21 409.04 422.56 401.36 420.26 423.46.

a) Ao nível de significância de 5% verifique se há evidências de que o teor médio de carbonato de sódio dos comprimidos produzidos pela empresa é diferente do especificado.

b) Construa também um intervalo de 95% de confiança para o teor médio de carbonato de cálcio dos comprimidos fabricados pela empresa?

Solução:

Vamos realizar o teste t para a média populacional. As hipóteses a serem testadas são:

H0: $\mu = 400$ Ha: $\mu \neq 400$

Onde μ é o teor médio de carbonato de cálcio dos comprimidos fabricados.

Entrando com os dados:

```
teor<-c(403.43,382.84,401.85,396.40,389.87,397.46,402.12,
406.28,404.48,411.89,402.09,407.59,391.65,389.81,393.21,
409.04,422.56,401.36,420.26,423.46)
```

Para realizar o teste vamos utilizar a função ***t.test***. Veja a seguir os argumentos desta função.

`x`	a (non-empty) numeric vector of data values.
`y`	an optional (non-empty) numeric vector of data values.
`alternative`	a character string specifying the alternative hypothesis, must be one of `"two.sided"` (default), `"greater"` or `"less"`. You can specify just the initial letter.
`mu`	a number indicating the true value of the mean (or difference in means if you are performing a two sample test).
`paired`	a logical indicating whether you want a paired t-test.
`var.equal`	a logical variable indicating whether to treat the two variances as being equal. If `TRUE` then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.
`conf.level`	confidence level of the interval.
`formula`	a formula of the form `lhs ~ rhs` where `lhs` is a numeric variable giving the data values and `rhs` a factor with two levels giving the corresponding groups.
`data`	an optional matrix or data frame (or similar: see model.frame) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
`subset`	an optional vector specifying a subset of observations to be used.
`na.action`	a function which indicates what should happen when the data contain `NA`s.

Defaults to `getOption("na.action")`.

`...` further arguments to be passed to or from methods.

Os 2 primeiros argumentos são os dados. No caso de inferência para uma média populacional, quando temos apenas uma amostra, devemos especificar somente o argumento x, o vetor teor no nosso exemplo.

O terceiro argumento, *alternative,* informa o tipo de hipótese alternativa. A opção padrão do R é hipótese alternativa bilateral (alternative="two-sided".) . Veja acima com fazer no caso de alternativas uni-laterais.

O argumento *mu* indica o valor da média populacional sobre a hipótese nula, no nosso caso faça ***mu = 400.*** Quando no especificado o R assume *mu = 0***.**

Os argumentos *paired* e *var.equal* devem ser especificados quando se tratar da comparação de 2 médias populacionais, o que não é o caso.

O argumento *conf.level* indica o coeficiente de confiança a ser utilizado no cálculo do intervalo de confiança. Se não for especificado, o R assume *conf.level = 0.95. (95% de confiança).* Falaremos mais tarde sobre os argumentos *data e formula.*

Aplicando a função ***t.test*** ao nosso exemplo.

```
> t.test(teor, mu=400)

        One Sample t-test

data:  teor
t = 1.1702, df = 19, p-value = 0.2564
alternative hypothesis: true mean is not equal to 400
95 percent confidence interval:
 397.7266 408.0384
sample estimates:
mean of x
 402.8825
```

Como resultados do teste de hipóteses, o R retorna o valor da estatística de teste, os graus de liberdade associados, o P-valor e o intervalo de confiança para a média populacional.

Se seu objetivo é apenas construir o intervalo de confiança, ignore os resultados do teste de hipóteses.

15.2 - Inferência Para uma Proporção Populacional

Exemplo: A prefeitura de uma cidade decidiu fazer uma pesquisa para avaliar a opinião dos moradores quanto à realização de uma obra e decidiu que a obra só será realizada se houver aprovação da mesma por 70% da população consultada. Considerando que uma amostra de 200 moradores corretamente selecionada foi ouvida, dos quais 120 responderam favoráveis, ao nível de significância de 5%, qual deve ser a decisão da prefeitura? Construa também um intervalo de 95% de confiança para a proporção de moradores favoráveis ao projeto.

Para responder as questões acima vamos utilizar a função ***prop.test***. Veja is argumentos desta função no quadro seguinte.

Arguments:
x: a vector of counts of successes or a matrix with 2 columns giving the counts of successes and failures, respectively.
n: a vector of counts of trials; ignored if 'x' is a matrix.
p: a vector of probabilities of success. The length of 'p' must be the same as the number of groups specified by 'x', and its elements must be greater than 0 and less than 1.
alternative: a character string specifying the alternative hypothesis, must be one of "'two.sided"' (default), "'greater"' or"'less"'. You can specify just the initial letter. Only used for testing the null that a single proportion equals a given value, or that two proportions are equal; ignored otherwise.
conf.level: confidence level of the returned confidence interval. Must be a single number between 0 and 1. Only used when testing the null that a single proportion

equals a given value, or that two proportions are equal; ignored otherwise.
correct: a logical indicating whether Yates' continuity correction should be applied.

No argumento x informamos o número observado de sucessos e no argumento n o número de realizações do experimento aleatório. Com o argumento p especificamos o valor da proporção amostral sob a hipótese nula (a opção padrão do R é p = 0.5). Os argumentos *alternative e conf.level* são similares àqueles da função *t.test*. Com o argumento *correct* indicamos se o teste deve ser realizado com ou sem a correção de continuidade de Yates. O objetivo da correção de continuidade é melhorar a qualidade da aproximação da distribuição da estatística de teste pela distribuição Qui-Quadrado. A opção padrão para o argumento *correct=TRUE.*

Teste com alternativa bi-lateral com correção de continuidade

Hipóteses: H0: p = 0.,7 x Ha: p ≠ 0.7

```
> prop.test(x = 120, n = 200, alternative="two.sided", conf.level =
0.95,p=0.7,correct=T)
        1-sample proportions test with continuity correction
data:  120 out of 200, null probability 0.7
X-squared = 9.0536, df = 1, p-value = 0.002622
alternative hypothesis: true p is not equal to 0.7
95 percent confidence interval:
 0.5283160 0.6677775
sample estimates:
  p
0.6
```

Faça uma comparação dos resultados entre os testes de hipóteses e intervalos de confiança realizados com e sem correção de continuidade.

15.2.1 - Outra Forma de Declarar os Dados na Função *prop.test*

Suponha que tivéssemos criado no R um vetor com as repostas de cada morador entrevistado. Para os moradores que foram favoráveis atribuímos reposta = “sim” e “não” caso contrário, como mostrado abaixo:

```
> dados<-c(rep(“sim”,120),rep(“não”,80”)
> dados
  [1] "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim"
 [13] "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim"
 [25] "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim"
 [37] "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim"
 [49] "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim"
 [61] "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim"
 [73] "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim"
 [85] "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim"
 [97] "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim"
[109] "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim" "sim"
[121] "não" "não" "não" "não" "não" "não" "não" "não" "não" "não" "não" "não"
[133] "não" "não" "não" "não" "não" "não" "não" "não" "não" "não" "não" "não"
[145] "não" "não" "não" "não" "não" "não" "não" "não" "não" "não" "não" "não"
[157] "não" "não" "não" "não" "não" "não" "não" "não" "não" "não" "não" "não"
[169] "não" "não" "não" "não" "não" "não" "não" "não" "não" "não" "não" "não"
[181] "não" "não" "não" "não" "não" "não" "não" "não" "não" "não" "não" "não"
[193] "não" "não" "não" "não" "não" "não" "não" "não"
```

Na função ***prop.test***, ao invés de fazermos x igual ao número de sucessos e n igual o número de experimentos aleatórios, podemos fazer x igual à tabela de freqüências da variável resposta de interesse . Neste caso a informação do argumento n é ignorada.

Obtendo a tabela de frequências

```
> t<-table(dados)
> t
dados
não sim
 80  120
>#  realizando o teste de hipóteses
> prop.test(x = t, alternative="two.sided", conf.level = 0.95,p=0.7)
```

O que fazer quando a aproximação Qui-Quadrado para a distribuição da estatística de teste sob a hipótese nula não é satisfatória?

A distribuição da estatística do teste (estatística Qui-Quadrado) sob a hipótese nula é aproximadamente uma distribuição Qui-quadrado com 1 grau de liberdade. Esta aproximação é considerada satisfatória quando os valores esperados de sucessos e fracassos sob a hipótese nula (np_0 e $n(1-p_0)$) forem maiores do que 5. Quando isto não ocorre os resultados obtidos utilizando tal aproximação (função ***prop.test***) podem não ser confiáveis. Nestes casos é recomendável utilizar a função ***binom.test***. O teste realizado por esta função utiliza a distribuição exata ao invés da distribuição aproximada.

```
> binom.test(x = 120, n=200, alternative="two.sided", conf.level = 
0.95,p=0.7)

        Exact binomial test

data:  120 and 200
number of successes = 120, number of trials = 200, p-value = 0.002565
alternative hypothesis: true probability of success is not equal to 0.7
95 percent confidence interval:
 0.5285357 0.6684537
sample estimates:
probability of success
                   0.6
```

Para o exemplo os números esperados de sucessos (respostas sim) e fracassos (respostas não) obtidos quando H0: p = 0,7 é verdadeira, respectivamente 140 (200 x 0,7) e 60 (200 x 0,3), são muito maiores do que 5. Isto garante que a aproximação pela distribuição Qui-Quadrado é satisfatória, o que pode ser visto comparando os resultados dos dois testes.

15.3 - Exercícios

1) Refaça o exemplo da seção 15.1 considerando

a) Ha: $\mu > 400$ e construa um intervalo com 90% de confiança.

b) Ha: $\mu < 400$ e construa um intervalo com 99% de confiança.

2)Suponha que foram ouvidos 20 eleitores, dos quais 16 responderam favoráveis ao projeto. Teste a hipótese alternativa de que $p \neq 0,70$ usando as funções ***prop.test*** (com e sem correção de continuidade) e ***binom.test***. Compare os resultados obtidos com o teste aproximado (***prop.test***) com e sem correção de continuidade com o resultado do teste exato (***binom.test***). Comente.

Aulas 16 - Comparação de Duas Proporções Populacionais

Nesta aula, vamos ver como utilizar o R para comparação de duas proporções populacionais. Duas situações serão consideradas:

1) teste de homogeneidade de duas populações

2) teste de independência entre duas variáveis aleatórias

16.1 - Tese de Homogeneidade de Duas Populações

Exemplo: Dois processos A e B de polimento de lentes intra-oculares foram avaliados. Cada processo foi utilizado em 300 lentes, alocadas aleatoriamente aos processos. Os números observados de lentes livres de defeitos decorrentes dos polimentos A e B foram 253 e 196 respectivamente.

a) Realize o teste de hipóteses adequado para verificar se há evidências de que os processos diferem com relação à proporção de lentes livres de defeitos.

b) Obtenha um intervalo de 95% de confiança para a diferença entre as proporções populacionais de lentes intra-oculares polidas pelos processos A e B livres de defeitos.

As hipóteses a serem testadas são: H0: $p_A = p_B$ x Ha: $p_A \neq p_B$

p_A – proporção de lentes do tipo A livres de defeito

p_B – proporção de lentes do tipo B livres de defeito

Para testá-las vamos utilizar a função ***prop.test.*** No argumento ***x,*** informamos o número de sucessos (lentes livres de defeitos) e, no argumento ***y,*** os tamanhos de amostra. A correção de continuidade é utilizada a menos que o contrário seja especificado.

```
> prop.test(x = c(253,196), n=c(300,300))
2-sample test for equality of proportions with continuity correction
data:  c(253, 196) out of c(300, 300)
X-squared = 27.7526, df = 1, p-value = 1.379e-07
alternative hypothesis: two.sided
95 percent confidence interval:
 0.1189026 0.2610974
sample estimates:
   prop 1    prop 2
0.8433333 0.6533333
```

O R retorna o valor observado da estatística do teste Qui-quadrado (27,75), os graus de liberdade da distribuição da estatística de teste quando H0 é verdadeira (distribuição Qui-quadrado com 1 grau de liberdade) e o p-valor (1.379e-07). O valor-p é muito pequeno (muito menor que o nível de significância $\alpha = 0,05$), indicando que há evidências amostrais de que os processos de polimento diferem-se quanto às proporções de lentes defeituosas. Com 95% de confiança, a diferença entre as proporções de lentes defeituosas produzidas pelos processos A e B está entre 0,11 e 0,26.

Suponha que tivéssemos testado 20 lentes de cada tipo e que tivéssemos observado respectivamente 18 e 14 lentes dos tipos A e B livres de defeitos. Observe os resultados do teste neste caso.

```
> prop.test(x = c(18,14),n=c(20,20))
2-sample test for equality of proportions with continuity correction
data:  c(18, 14) out of c(20, 20)
X-squared = 1.4063, df = 1, p-value = 0.2357
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.09004558  0.49004558
sample estimates:
prop 1 prop 2
   0.9    0.7

Warning message:
In prop.test(x = c(18, 14), n = c(20, 20)) :
Aproximação Qui-quadrado pode estar incorreta
```

Observe que o R apresenta a mensagem (em negrito acima) de que a aproximação pela distribuição Qui-Quadrado pode estar incorreta. Isto acontece porque os valores esperados

de lentes defeituosas quando H0 é verdadeira são menores do que 5 (20 x 0,2 = 4). Nesta situação duas alternativas, podem ser consideradas.

1) fazer o teste exato de Fisher, que pode ser realizado utilizando a função ***fisher.test.***

2) obter o p-valor através de simulações de Monte-Carlo. Use a função ***chisq.test.***

16.1.1 - Teste Exato de Fisher

Para usar a função ***fisher.test,*** os dados devem estar organizados numa tabela ou matriz, com informação do número de sucessos e fracassos.

	Livres de Defeitos	Defeituosas
Processo A	18	02
Processo B	14	06

```
> # criando a matriz de dados no R
> dados<-matrix(c(18,14,2,6), nrow=2, dimnames = list(c("Lente tipo A",
"Lente tipo B"), c("Não defeituosas","defeituosa")))
> dados
             Não defeituosas defeituosa
Lente tipo A              18          2
Lente tipo B              14          6

> # realizando o teste exato de Fishee
> fisher.test(dados)
        Fisher's Exact Test for Count Data
data:  Dados
p-value = 0.2351
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
  0.5553199 43.3242534
sample estimates:
odds ratio
  3.731286
```

As hipóteses testadas pelo teste exato de Fisher e pelo teste Qui-Quadrado de Pearson são as mesmas. Porém no "output' do teste de Fisher, ela é escrita em termos da razão das chances $OR = \frac{\frac{p_A}{(1-p_A)}}{\frac{p_B}{(1-p_B)}}$. Se $p_A = p_B$, então OR = 1. Portanto H0: OR = 1 é equivalente a H0: $p_A = p_B$.

No caso do exemplo, a razão das chances é estimada em 3,73 e com 95% de confiança está entre 0,55 e 43,32.

16.1.2 - Obtendo o Valor P por Simulação de Monte Carlo

As funções ***chisq.test*** e ***prop.test*** realizam o teste Qui-Quadrado de Pearson para comparar 2 proporções. A função ***prop.test*** pode ser utilizada quando a variável resposta possui 2 categorias. Quando não for este o caso, podemos usar a função ***chisq.test.*** Nesta função, há duas opções para o cálculo do P-valor: usar a aproximação Qui-Quadrado para a distribuição da estatística de teste (fazendo o argumento ***simul = F)*** ou obtê-lo por simulações de Monte Carlo (fazendo o argumento ***simul = T).***

A forma de entrada dos dados na função ***chisq.test*** é idêntica àquela da função ***fisher.test***. A seguir, a função **chisq.test** será utilizada para calcular o valor-P do teste Qui-Quadrado de Pearson, considerando a aproximação Qui-Quadrado e as simulações de Monte Carlo.

Obtendo P- valor através da distribuição Qui-Quadrado

```
> chisq.test(Dados)
  Pearson's Chi-squared test with Yates' continuity correction
data:  Dados
X-squared = 1.4062, df = 1, p-value = 0.2357
Warning message:
In chisq.test(Dados) : Aproximação Qui-quadrado pode estar incorreta
```

Obtendo P- valor através de simulação

```
> chisq.test(Dados,simul=T)
  Pearson's Chi-squared test with simulated p-value (based on 2000
        replicates)
data:  Dados
X-squared = 2.5, df = NA, p-value = 0.2289
```

Observe que o valor da estatística de teste obtido quando especificamos ***simul=T*** é diferente do valor obtido quando fazemos ***simul = F.*** Isto acontece, porque na segunda situação, a estatística é calculada com correção de continuidade. Se você especificar ***correct = F,*** os valores obtidos nos dois casos serão idênticos.

```
> chisq.test(Dados,correct=F)
        Pearson's Chi-squared test
data:  Dados
X-squared = 2.5, df = 1, p-value = 0.1138
Warning message:
In chisq.test(Dados, correct = F) :
  Aproximação Qui-quadrado pode estar incorreta
```

16.2 - Teste de Independência Entre Duas Variáveis Qualitativas

Exemplo: Num estudo sobre a efetividade do uso de capacetes de segurança para ciclista na prevenção de lesões na cabeça, a equipe classificou uma amostra de ciclistas que sofreram acidentes durante certo período de tempo quanto ao uso do capacete e a ocorrência de lesão na cabeça. Os resultados observaram foram:

Uso do capacete	Lesão na cabeça	
	Sim	Não
Sim	17	130
Não	218	428

Os dados acima constituem evidências de existência de associação entre uso de capacete e lesão na cabeça? Para responder a esta pergunta, faça o teste de hipóteses adequado ao nível de significância de 5%.

Solução:

As hipóteses a serem testadas são:

H0: $p_1 = p_2$ x Ha: $p_1 \neq p_2$

p_1 = P(lesão = sim| uso do capacete =sim)

p_2 = P(lesão = sim| uso do capacete =não)

Elas também podem ser escritas em função da razão das chances

$$H0: \frac{\frac{p_1}{(1-p_1)}}{\frac{p_2}{(1-p_2)}} = 1 \quad \text{x} \quad Ha: \frac{\frac{p_1}{(1-p_1)}}{\frac{p_2}{(1-p_2)}} \neq 1$$

Como se trata de um teste da igualdade de proporções, podemos utilizar as funções ***prop.test, fisher.test*** **e** ***chisq.test*** para realizar o teste das hipóteses. A seguir são apresentados os resultados da aplicação da função ***prop.test.***

```
> prop.test(x = c(17,218),n=c(147,646))
        2-sample test for equality of proportions with continuity
correction
data:  c(17, 218) out of c(147, 646)
X-squared = 27.2018, df = 1, p-value = 1.833e-07
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.2892529 -0.1543772
sample estimates:
   prop 1    prop 2
0.1156463 0.3374613
```

Criando a matriz de dados para utilização da função ***chisq.test.*** e ***fisher.test***

```
>bicicleta<-matrix(c(17,218,130,428),nrow=2,dimnames=list(capacete=
c("Sim ", "não"), lesão=c("sim","não")))
> bicicleta
        lesão
capacete sim não
    Sim   17 130
    não  218 428

> chisq.test(bicicleta)
        Pearson's Chi-squared test with Yates' continuity correction
data:  bicicleta
X-squared = 27.2018, df = 1, p-value = 1.833e-07

> chisq.test(bicicleta,simul=T)
  Pearson's Chi-squared test with simulated p-value (based on 2000
        replicates)
data:  bicicleta
X-squared = 28.2555, df = NA, p-value = 0.0004998

> fisher.test(bicicleta)
        Fisher's Exact Test for Count Data
data:  bicicleta
p-value = 2.273e-08
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.1416075 0.4413995
sample estimates:
odds ratio
 0.2571032
```

16.3 - Exercícios

1) Uma amostra de 418 moscas foi classificada segundo o tipo de olhos e tipo de pelos, obtendo-se os resultados abaixo:

	Olhos		
Pelos	Normais	Reduzidos	Total
Normais	244	82	326
Reduzidos	80	12	92
Total	324	94	418

Ao nível de significância de 5%, há evidências de associação entre tipo de olho e tipo de pelo?

2) Uma amostra de 100 insetos foi selecionada de uma região úmida e outra amostra de 100 insetos de uma região seca. Um inseticida foi aplicado aos insetos das duas amostras e a sobrevivência dos mesmos foi observada, obtendo-se os resultados abaixo:

	Sobrevivência		
Região	Sim	Não	Total
Seca	90	10	100
Úmida	35	65	100
Total	324	94	200

Ao nível de significância de 5%, pode-se dizer que o efeito do inseticida é diferente entre as regiões?

Aula 17 – Teste Qui-Quadrado Para Variáveis Categóricas

Na aula 16 vimos como usar o R para fazer teste de homogeneidade e independência para tabelas 2 x 2. Nesta aula vamos considerar o caso de tabelas de contingência r x c (r ou c maiores do que 2). Para isto vamos utilizar a função ***chisq.test.*** A função ***prop.test*** pode ser utilizado quando a variável reposta possui apenas 2 categorias de resposta (sucesso e fracasso).

Além dos modelos de homogeneidade e independência, o teste o teste Qui-Quadrado é útil para testar o ajuste de outros modelos. Na última seção desta aula veremos como usar a função ***chisq.test*** em alguns exemplos.

17.1 - Teste de Homogeneidade

Exemplo: Um estudo comparou um hospital comunitário (hospital A) com um hospital universitário (hospital B) quanto à precisão no preenchimento da causa de morte no atestado de óbito. Para isto, uma amostra de 229 atestados do hospital A e outra de 346 atestados do hospital B foram selecionadas e categorizadas segundo a precisão no registro da informação em: registro preciso, registro impreciso e registro incorreto. Os resultados são apresentados na tabela abaixo:

	Qualidade do registro		
Hospital	Correto	Impreciso	Incorreto
A	157	18	54
B	268	44	34

Os resultados do estudo sugerem práticas diferentes no preenchimento da causa da morte nos atestados de óbito pelos dois hospitais? Faça o teste de hipótese adequado ao nível de significância de 5%.

O primeiro passo é construir uma matriz com os dados observados.

```
>qualidade<-matrix(c(157,268,18,44,54,34),nrow=2,dimnames=
 list( Hospital = c("Comunitário","Universitário"), Registro=c("correto",
"impreciso", "incorreto")))
> qualidade
              Registro
Hospital        correto impreciso incorreto
  Comunitário       157        18        54
  Universitário     268        44        34
```

Realizando o teste Qui-Quadrado

```
> chisq.test(qualidade)
      Pearson's Chi-squared test
data:  qualidade
X-squared = 21.5235, df = 2, p-value = 2.120e-05
```

A partir do teste acima (P-valor = 0,0000212) concluímos que há evidências amostrais de que os hospitais comunitário e universitário diferem-se no preenchimento dos atestados de óbito.

A análise da tabela de freqüências relativas, dada a seguir, permite entender como os dois hospitais se diferem no preenchimento da causa de morte no atestado de óbito.

```
> prop.table(qualidade,1)
              Registro
Hospital          correto  impreciso  incorreto
  Comunitário   0.6855895  0.07860262  0.2358079
  Universitário 0.7745665  0.12716763  0.0982659
```

Observe que a proporção de registros corretos e imprecisos é maior no hospital universitário e a proporção de registros incorretos é maior no hospital universitário, sugerindo que a qualidade no indicando que a qualidade no preenchimento é melhor no hospital universitário.

Outra análise interessante é a análise dos resíduos do modelo. Os resíduos são definidos como $r = \frac{O_i - E_i}{\sqrt{E_i}}$ são conhecidos como resíduos de Pearson. Para obter os resíduos, faça como segue:

```
>teste<-chisq.test(qualidade)      #guarda      os      resultados      de
chisq.test(qualidade) em teste
> teste$res                                  # retorna os residuos
                                Registro
      Hospital          correto    impreciso  incorreto
      Comunitário     -0.9424167  -1.346752  3.201502
      Universitário    0.7666951   1.095638  -2.604555
```

Observe que o hospital comunitário apresenta resíduos negativos para as categorias correto e impreciso, indicando que as freqüências observadas de resultados nestas categorias são menores do que aquelas esperadas sob a hipótese nula. Por outro lado, o hospital universitário apresenta resíduo negativo para a categoria incorreto, indicando que ele apresenta freqüência de resultados incorretos abaixo do esperado. A categoria com maior resíduo, em valor absoluto, é a de resultado incorreto. Esta categoria é a que mais contribui para a estatística Qui-Quadrado e consequentemente para a diferença entre as distribuições da qualidade do registro da causa de morte.

Para obter as freqüências observadas e esperadas, proceda como segue:

```
> teste$obs
               Registro
Hospital         correto impreciso incorreto
  Comunitário        157        18         54
  Universitário      268        44         34

> teste$exp
               Registro
Hospital          correto   impreciso   incorreto
  Comunitário    169.2609    24.69217    35.04696
  Universitário 255.7391    37.30783    52.95304
```

17.2 - Teste de Independência

Exemplo: Uma amostra de pacientes portadores de melanoma, um tipo de câncer de pele, foi classificada segundo o local de aparecimento do câncer e o seu tipo. Os dados são dados na tabela abaixo:

	Local		
Tipo	Cabeça	Tronco	Extremidades
freckle	22	2	10
superficial	16	54	115
nodular	19	33	73
indeterminado	11	17	28

Verifique, através de um teste de hipóteses ao nível de significância de 5%, se há evidências de associação entre tipo e localização do câncer.

O primeiro passo é a entrada dos dados no R. No exemplo anterior construímos uma matriz com os dados. Neste exemplo, vamos primeiro criar um ***data.frame*** com as variáveis tipo, local e freqüência. Depois, utilizando a função ***xtabs,*** construiremos uma tabela de freqüência.

```
>dados<-data.frame(freq=c(22,16,19,11,2,54,33,17,10,115,73,28),tipo=
rep(c("frekle","superficial","nodular","indeterminado"),each=3),local=
rep(c("cabeça", "tronco", "extremidade"),times=4))
> dados
   freq          tipo       local
1    22        frekle      cabeça
2    16        frekle       tronco
3    19        frekle extremidade
4    11   superficial      cabeça
5     2   superficial       tronco
6    54   superficial extremidade
7    33       nodular      cabeça
8    17       nodular       tronco
9    10       nodular extremidade
10  115 indeterminado      cabeça
```

```
11    73 indeterminado       tonco
12    28 indeterminado extremidade
```

Usando a função ***xtabs*** para construir a tabela de freqüência.

O comando *xtabs(freq~tipo+local,data=Dados)* especifica que as freqüências em *freq* devem ser organizada numa tabela segundo as categorias de *tipo e local*, utilizando os dados do ***data.frame*** *dados.*

```
> tabeladados<-xtabs(freq~tipo+local,data=Dados)
> tabeladados
              local
tipo           cabeça extremidade tronco
  frekle           22          19     16
  indeterminado   115          28     73
  nodular          33          10     17
  superficial      11          54      2
```

A tabela de freqüências relativas dada abaixo sugere associação entre tipo e local do câncer. Observe que o tumor do tipo "frekle" distribui de maneira quase uniforme entre os locais. Os tumores nodular e indeterminado são mais freqüentes na cabeça e o superficial nas extremidades.

```
> prop.table(tabeladados,1)
                                  local
tipo                cabeça extremidade  tronco
  frekle        0.38596491  0.33333333  0.28070175
  indeterminado 0.53240741  0.12962963  0.33796296
  nodular       0.55000000  0.16666667  0.28333333
  superficial   0.16417910  0.80597015  0.02985075
```

Construindo o teste Qui-Quadrado

```
> testetabeladados<-chisq.test(tabeladados)
> testetabeladados
        Pearson's Chi-squared test
data:  tabeladados
X-squared = 122.9907, df = 6, p-value < 2.2e-16
```

Concluímos do teste acima que os dados constituem evidência de associação entre tipo e local do tumor (P-valor < 2.2e-16).

Obtendo freqüências esperadas e resíduos.

```
> testetabeladados$exp
                local
tipo             cabeça extremidade   tronco
  frekle         25.7925   15.8175   15.39
  indeterminado 97.7400   59.9400   58.32
  nodular        27.1500   16.6500   16.20
  superficial    30.3175   18.5925   18.09

> testetabeladados$res
                              local
tipo              cabeça     extremidade  tronco
  frekle         -0.7467563   0.8002017  0.1554929
  indeterminado   1.7458407  -4.1254995  1.9222829
  nodular         1.1227187  -1.6297257  0.1987616
  superficial    -3.5083606   8.2115732 -3.7830037
```

Observe que os maiores resíduos são observados para o tipo superficial, cuja distribuição de localização é a que mais se difere das demais.

17.- 3 – Exercícios

1) Uma amostra de pacientes psiquiátricos foi classificada segundo o diagnóstico e prescrição de tratamento medicamentoso (sim ou não). Os dados observados são dados na tabela abaixo:

	Prescrição de Medicamento	
Diagnóstico	Não	Sim
Distúrbio de afetivo	2	12
Neurose	19	18
Distúrbio de personalidade	52	47
Esquizofrenia	6	105
Outros diagnósticos	18	0

Verifique se a prescrição de medicamentoso depende do diagnóstico.

2) A tabela abaixo contém a distribuição dos suicídios observados durante um ano no Reino Unido segundo o sexo e o instrumento utilizado. Há evidências de associação entre o sexo e o instrumento utilizado?

	Sexo	
Instrumento utilizado	Feminino	Masculino
Drogas	863	890
Gás	33	209
Arma de fogo	47	356
Forca	730	1568
Salto	149	132
Outros	108	220

Aula 18 - Teste Qui-quadrado Para o Ajuste de Modelos

Exemplo: O cruzamento de dois tipos de plantas pode resultar em três genótipos diferentes: A, B e C. Um modelo genético teórico sugere que estes genótipos devem ocorrer na razão 1:2:1. Noventa plantas foram obtidas do cruzamento das variedades parentais. Destas, 18 apresentaram genótipo do tipo A, 44 do tipo B e 28 do tipo C. Estes dados suportam ou contradizem o modelo genético?

Podemos utilizar o teste Qui-Quadrado para avaliar se o modelo genético pode ser usado para descrever as freqüências de genótipos resultantes do cruzamento dos dois tipos de planta. As hipóteses a serem testadas são:

H0: O modelo genético teórico é adequado para descrever as freqüências de genótipos rutilantes do cruzamento dos dois tipos de plantas

Ha: O modelo genético teórico não é adequado para descrever as freqüências de genótipos rutilantes do cruzamento dos dois tipos de plantas

As freqüências observadas dos genótipos A, B e C são respectivamente 18, 44 e 28. As freqüências esperadas assumindo o modelo genético como adequado são respectivamente 22.5, 50 e 22.5.

Para realizar o teste no R, vamos utilizar a opção ***chisq.test.*** É necessário especificar os valores observados (argumento ***x***) e o modelo de probabilidade estabelecido em H0 (argumento ***p***). Há diferentes maneiras de especificar este modelo. Podemos fazer ***p*** igual à distribuição de probabilidade (p_A = 0,25, p_B= 0,50, p_C = 0,25). Neste caso, a soma dos elementos de ***p*** é igual a 1. Podemos também fazer ***p*** igual às freqüências absolutas esperadas (.22.5, 50 e 22.5). Neste caso, deve-se utilizar a função ***rescale.p=T*** para alterar a escala dos valores em ***p***, de tal modo que somem 1.

```
> # colocando em p as probabilidades esperadas quando H0 é verdadeira.
> chisq.test(x=c(18,44,28),p=c(0.25,0.50,0.25))
> # colocando em p as probabilidades esperadas quando H0 é verdadeira.
> chisq.test(x=c(18,44,28),p=c(22.5,45,22.5), rescale.p=T)
  Chi-squared test for given probabilities
  data:  c(18, 44, 28)
  X-squared = 2.2667, df = 2, p-value = 0.3220
```

Podemos concluir do teste acima que, ao nível de significância de 5%, não há evidências de que o modelo genético teórico seja inadequado para descrever as freqüências de genótipos resultantes dos cruzamentos dos dois tipos de plantas (Valor-P = 0,32220).

A função ***chisq.test*** pode ser utilizada para fazer o teste do ajuste do modelo somente quando o modelo de probabilidades estabelecido em H0 é completamente conhecido. Quando ele for estimado a partir das freqüências observadas, o valor da estatística de teste obtido é correto, mas os graus de liberdade e o valor-p retornados são incorretos. Nesta situação, o número de graus de liberdade apropriado é igual a (*n – 1 – número de parâmetros estimados)*, onde *n* é o número de categorias da variável resposta.

Quando algum dos valores esperados for menor do que 5, a distribuição Qui-Quadrado não é adequada para descrever o comportamento da estatística de teste sob H0. Neste caso, o p-valor pode ser obtido por simulação de Monte Carlo. Neste método, um número grande amostras da distribuição especificada em H0 é gerado e, para cada uma delas, é calculada a estatística de teste. O valor-P é obtido como a freqüência relativa de amostras cujo valor da estatística de teste é maior ou igual ao valor observado.

```
> chisq.test(x=c(18,44,28),p=c(0.25,0.50,0.25), rescale.p=T, simul=T)
        Chi-squared test for given probabilities with simulated
   p-value (based on 2000 replicates)
   data:  c(18, 44, 28)
   X-squared = 2.2667, df = NA, p-value = 0.3378
```

Exemplo: Um pesquisador acredita que, numa determinada população, o número de descendentes deixados por indivíduo pode ser descrito por uma distribuição Poisson com λ=1. A tabela abaixo apresenta as probabilidades calculadas para esta distribuição.

x	0	1	2	3	4	≥5
P(X=x)	0,3679	0,3679	0,1839	0,0613	0,0153	0,0037

Observando uma amostra de 500 pessoas desta população, o pesquisador encontrou os seguintes resultados, dados na tabela seguinte:

Número de filhos	Freqüências observadas
0	170
1	180
2	95
3	35
4	18
≥5	2

O modelo de Poisson é adequado para descrever o número de descendentes deixados pelos indivíduos desta população? Considere nível de significância de 5%.

```
> # obtendo probabilidades esperadas sob H0
> p<-c(dpois(0:4,1), ppois(4,1,lower.tail=F) )
> p
[1]   0.367879441   0.367879441   0.183939721   0.061313240   0.015328310
0.003659847

> freqobs<-c(170,180,95,40,8,5)

> chisq.test(x = freqobs,p = p)

        Chi-squared test for given probabilities

data:  freqobs
X-squared = 9.6252, df = 5, p-value = 0.08658

Warning message:
In chisq.test(x = freqobs, p = p) :
  Aproximação Qui-quadrado pode estar incorreta
```

Como a aproximação Qui-Quadrado pode estar incorreta, vamos obter o P-valor por simulação.

```
> chisq.test(x=freqobs,p=p,simul=T)
Chi-squared test for given probabilities with simulated p-value (based
on 2000 replicates)
data:  freqobs
X-squared = 9.6252, df = NA, p-value = 0.08346
```

Observe que, ao nível de significância de 5%, não rejeitamos H0 (valor-P = 0.08346). Não há evidências suficientes de que o modelo proposto pelos pesquisadores para descrever o número de descendentes seja inadequado. De todo modo, vale a pena observar os resíduos do modelo.

```
> modelo<-chisq.test(x=freqobs,p=p,simulate.p.value=T)
> modelo$res
[1] -0.9755215 -0.2367118  0.3550368  1.7130695  0.1326521  2.3535599
```

Observe que o modelo superestima as freqüências de indivíduos com 0 ou 1 descendentes e subestima as demais freqüências.

Suponha que o pesquisador acredite que o modelo adequado para descrever o número de descendentes é Poisson, mas não tem idéia do valor do parâmetro λ, o número médio de descendentes por indivíduo. Neste caso, vamos estimar λ pelo número médio de descendentes observado na amostra.

$$\lambda = [(170 \times 0) + (180 \times 1) + ... + (5 \times 2)]/500.$$

No R, isto pode ser obtido como segue:

```
> # estimando lambda
> ndesc<- 0:5
> lambda <-sum(freqobs*ndesc)/sum(freqobs)
```

Continuando com o teste Qui-quadrado :

```
> # Probabilidades esperadas sob H0
> p<-c(dpois(0:4,lambda), ppois(4,lambda,lower.tail=F) )
> # realizando o teste Qui-Quadrado - Valor P obtido por aproximação Qui-
quadrado
> chisq.test(x=freqobs,p=p)
        Chi-squared test for given probabilities
data:  freqobs
X-squared = 3.1011, df = 5, p-value = 0.6844
Warning message:
In chisq.test(x = freqobs, p = p) :
  Aproximação Qui-quadrado pode estar incorreta
```

Neste caso, como estimamos o valor de λ, o número de graus de liberdade da distribuição Qui-quadrado, *gl = (6 – 1 - número de parâmetros estimados = 4*), é diferente daquele informado pelo R (*gl = 5*). Entretanto, o p-valor obtido por simulação de Monte Carlo está tecnicamente correto.

```
> # realizando o teste Qui-Quadrado - Valor P obtido por aproximação Qui-
quadrado
> chisq.test(x=freqobs,p=p,simulate.p.value=T)
        Chi-squared test for given probabilities with simulated p-value
(based
        on 2000 replicates)
data:  freqobs
X-squared = 3.1011, df = NA, p-value = 0.6787
```

Para obter o valor P utilizando a distribuição Qui-Quadrado com graus de liberdades adequados (*gl = 6 - 1 -1 = 4*), faça:

```
> valorP<-pchisq(3.1011,4,lower.tail=F)
> valorP
[1] 0.5410514
```

18.1 - Exercícios:

1) Do cruzamento de plantas puras de ervilhas com sementes amarelas lisas com plantas puras de sementes verdes rugosas, obtêm-se ervilhas com sementes amarelas lisas. As plantas da primeira geração produzem por auto-fecundação ervilhas de 4 tipos: amarelas lisas, amarelas rugosas, verdes lisas e verde rugosas. Pela Teoria Mendeliana, as proporções esperadas de sementes destes 4 tipos são respectivamente 9/16, 3/16,3/16 e 1/16. Na tabela abaixo, são apresentados as freqüências observadas dos 4 fenótipos observadas a partir da autofecundação de plantas híbridas**:**

Tipo de Ervilha	Freqüência Observada
Amarelas lisas	315
Verdes lisas	108
Amarelas Rugosas	101
Verdes Rugosas	32

Verifique através de um teste de hipóteses, ao nível de significância de 5%, se a Teoria Mendeliana é adequada para descrever as freqüências de fenótipos resultantes de autofecundação de plantas da primeira geração.

2) Os dados da tabela abaixo representam o número de árvores da espécie *Guapira opposita* observados por metro quadrado em uma área de restinga. A área de restinga foi dividida em 94 quadrantes (áreas menores de mesmo tamanho), e, em cada um deles, contou-se o número de árvores.

X - Número de arvores por quadrante	Número de quadrantes com X árvores
0	6
1	18
2	23
3	19
4	11
5	06
6	5
7	4
8	1
≥ 9	1

Pode-se observar que há 6 quadrantes com 5 árvores e 1 quadrante com mais de 9 árvores.

O objetivo do estudo era avaliar se a distribuição da espécie na região é completamente aleatória, ou seja, a chance de uma árvore ocorrer em qualquer ponto da região é a mesma, independente da existência de qualquer outra árvore na proximidade. Quando a distribuição das árvores é completamente aleatória, a distribuição das plantas por quadrante é uma distribuição de Poisson.

Verifique utilizando um teste de hipóteses ao nível de significância de 5% se a distribuição das árvores de *Guapira opposita* é completamente aleatória na área de restinga. (obtenha o P-valor por simulação de Monte Carlo e também utilizando a aproximação Qui-Quadrado).

Respostas aos Exercícios

Aula 5

1) freqüência absoluta

```
sexo

 feminino masculino

      313       279


olho

    azul castanho    preto    verde

     215       93      220       64
```

Freqüência relativa

```
sexo

 feminino masculino

0.5287162 0.4712838


olho

     azul   castanho     preto      verde

0.3631757   0.1570946   0.3716216   0.1081081
```

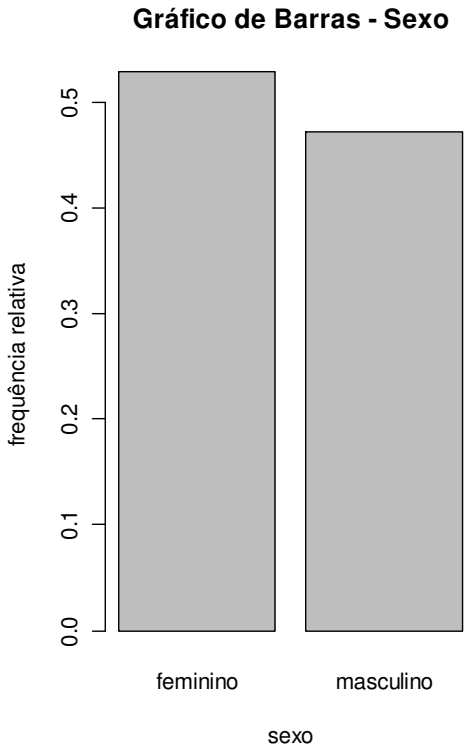


Gráfico de Barras - Sexo
frequência relativa
0.0
0.1
0.2
0.3
0.4
0.5
feminino
masculino
sexo

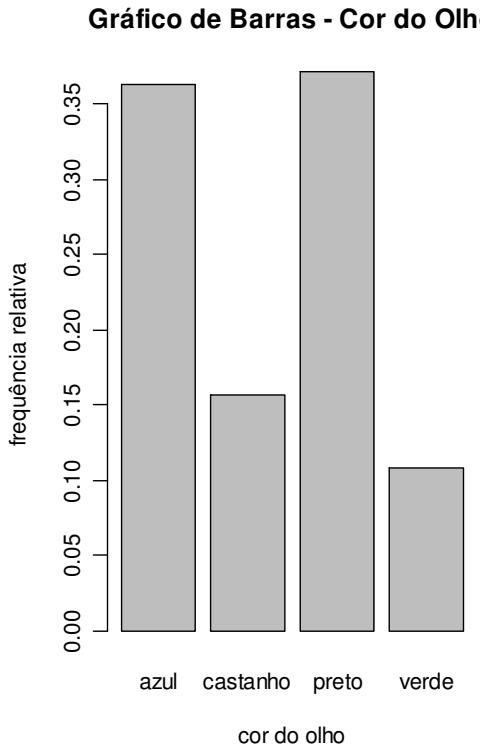
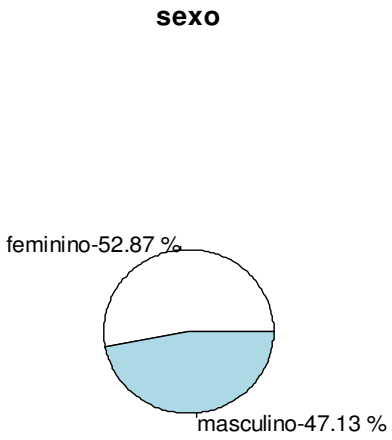
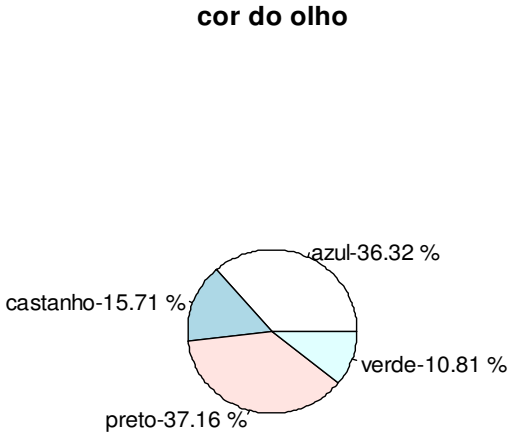


Gráfico de Barras - Cor do Olho
frequência relativa
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
azul
castanho
preto
verde
cor do olho

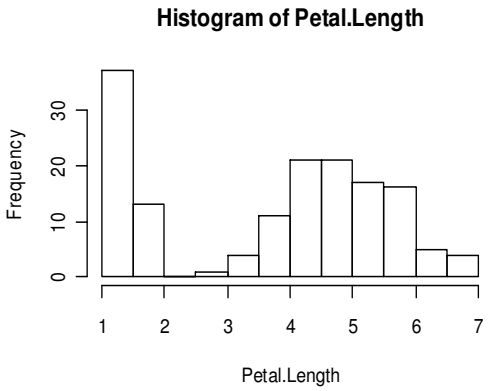
2)

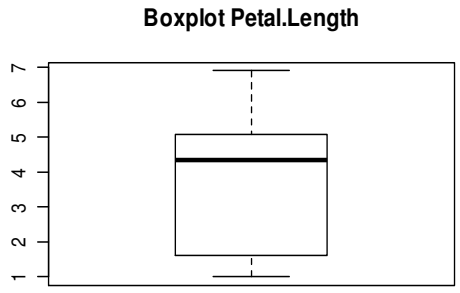

sexo
feminino-52.87 %
masculino-47.13 %

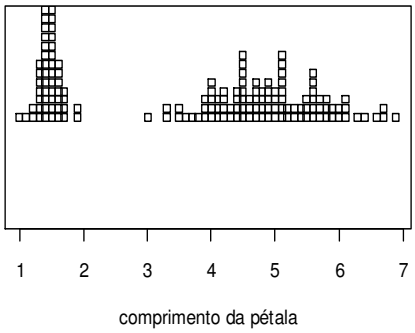

cor do olho
azul-36.32 %
castanho-15.71 %
verde-10.81 %
preto-37.16 %

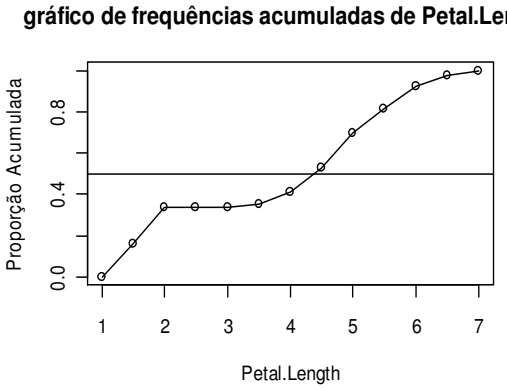
Aula 6

1)









Ramo e folhas

```
The decimal point is at the |

  1 | 01223333333444444444444
  1 | 55555555555556666666777799
  2 |
  2 |
  3 | 033
  3 | 55678999
  4 | 000001112222334444
  4 | 55555555666777778888999999
  5 | 000011111111223344
  5 | 55566666677788899
  6 | 0011134
  6 | 6779
```

2) setosa

3) Parte numérica

Largura da sépala

$setosa

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.300   3.200   3.400   3.428   3.675   4.400
```

$versicolor

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.525   2.800   2.770   3.000   3.400
```

$virginica

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.200   2.800   3.000   2.974   3.175   3.800
```

Variância

```
  setosa    versicolor  virginica
0.14368980  0.09846939  0.10400408
```

Desvio Padrão

```
setosa versicolor  virginica
 0.3790644  0.3137983  0.3224966
```

Comprimento da pétala

$setosa

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.400   1.500   1.462   1.575   1.900
```

$versicolor

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3.00    4.00    4.35    4.26    4.60    5.10
```

$virginica

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  4.500   5.100   5.550   5.552   5.875   6.900
```

Variância

```
  setosa   versicolor   virginica
0.03015918  0.22081633  0.30458776
```

Desvio Padrão

```
 setosa   versicolor  virginica
 0.1736640  0.4699110  0.5518947
```

Largura da pétala

$setosa

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.100   0.200   0.200   0.246   0.300   0.600
```

$versicolor

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.200   1.300   1.326   1.500   1.800
```

$virginica

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.400   1.800   2.000   2.026   2.300   2.500
```

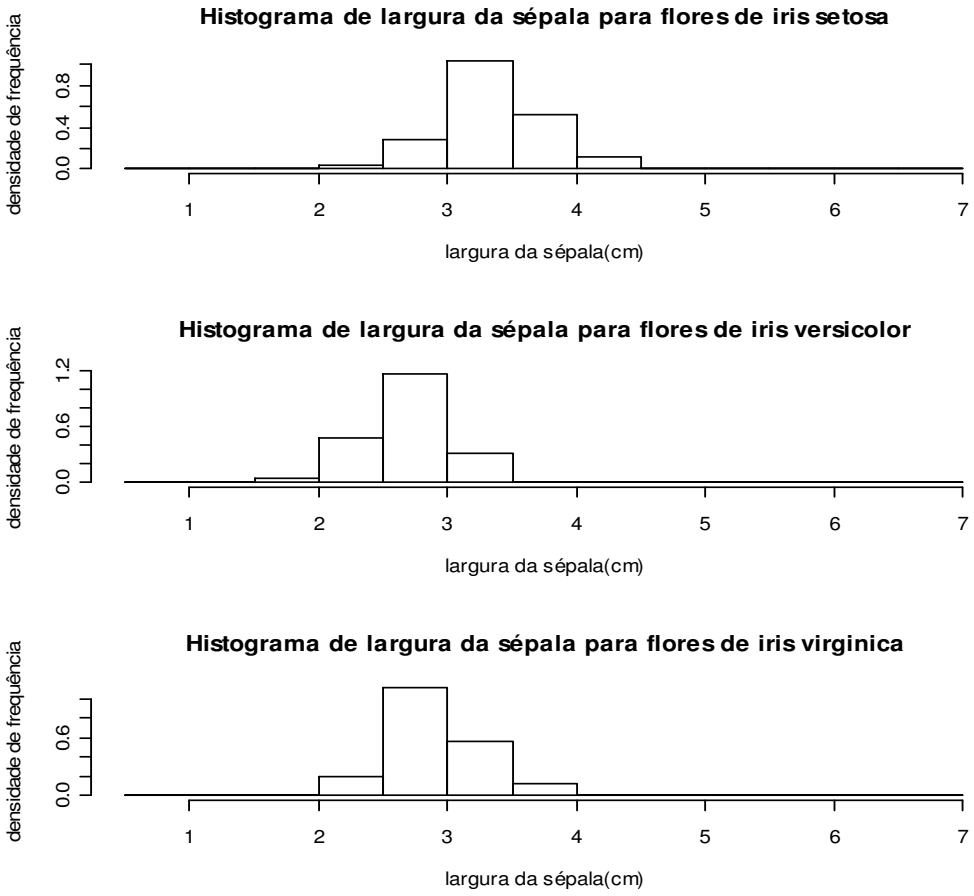
Variância

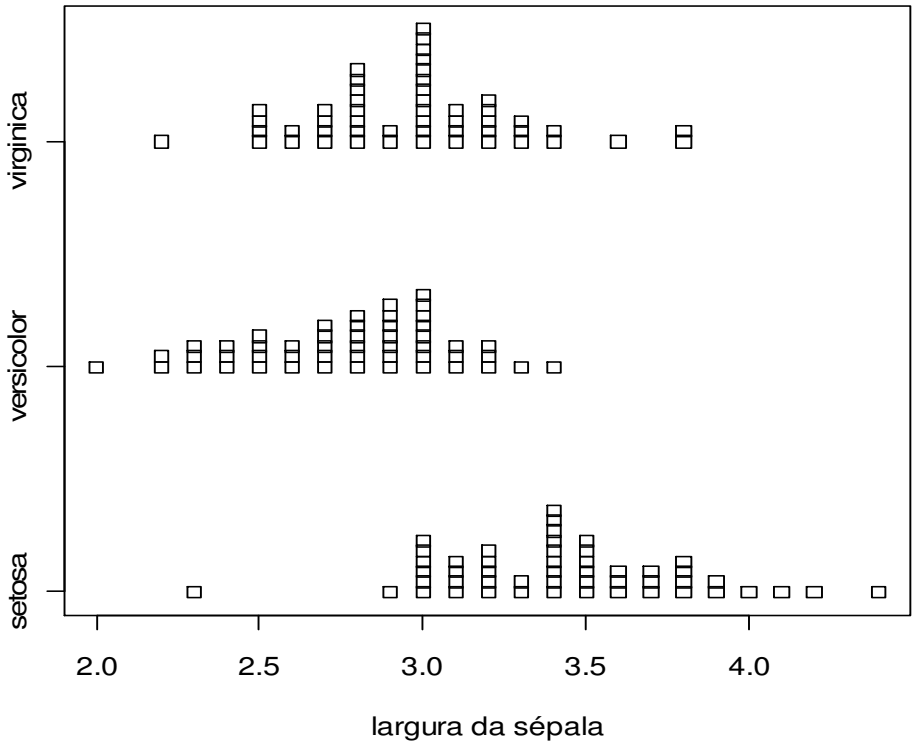
```
 setosa  versicolor  virginica
0.01110612  0.03910612  0.07543265
```

Desvio Padrão

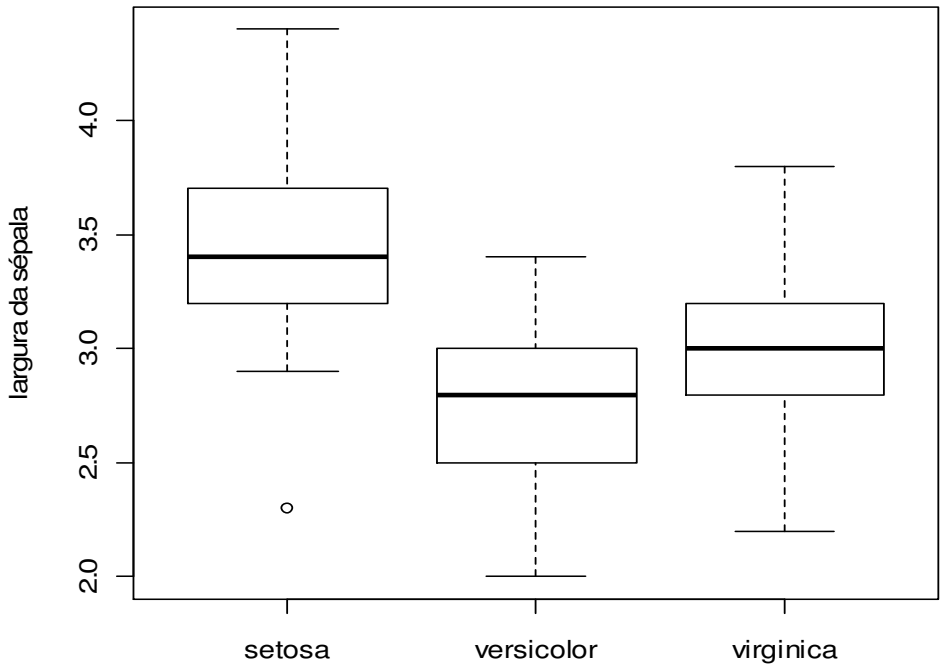
```
  setosa  versicolor  virginica
0.1053856  0.1977527  0.2746501
```

Parte gráfica



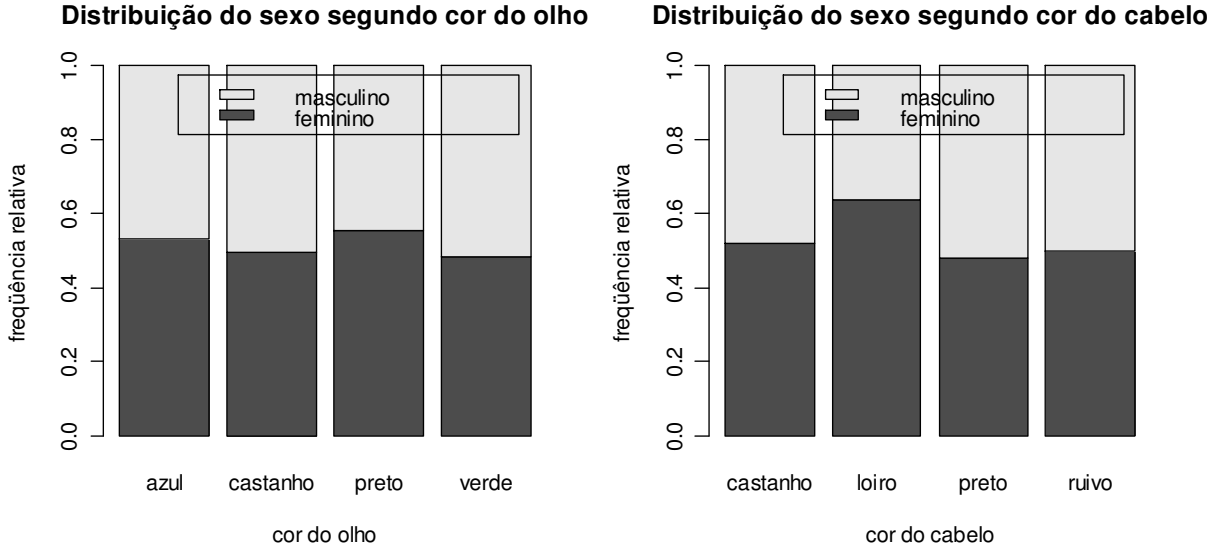


Boxplot da largura da sépala segundo espécie



Aula 7

2)



Aula 8

1)

Covariância - setosa

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	0.12424898	0.09921633	0.016355102	0.010330612
Sepal.Width	0.09921633	0.14368980	0.011697959	0.009297959
Petal.Length	0.01635510	0.01169796	0.030159184	0.006069388
Petal.Width	0.01033061	0.00929796	0.006069388	0.011106122

Correlação - setosa

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	1.0000000	0.7425467	0.2671758	0.2780984
Sepal.Width	0.7425467	1.0000000	0.1777000	0.2327520
Petal.Length	0.2671758	0.1777000	1.0000000	0.3316300
Petal.Width	0.2780984	0.2327520	0.3316300	1.0000000

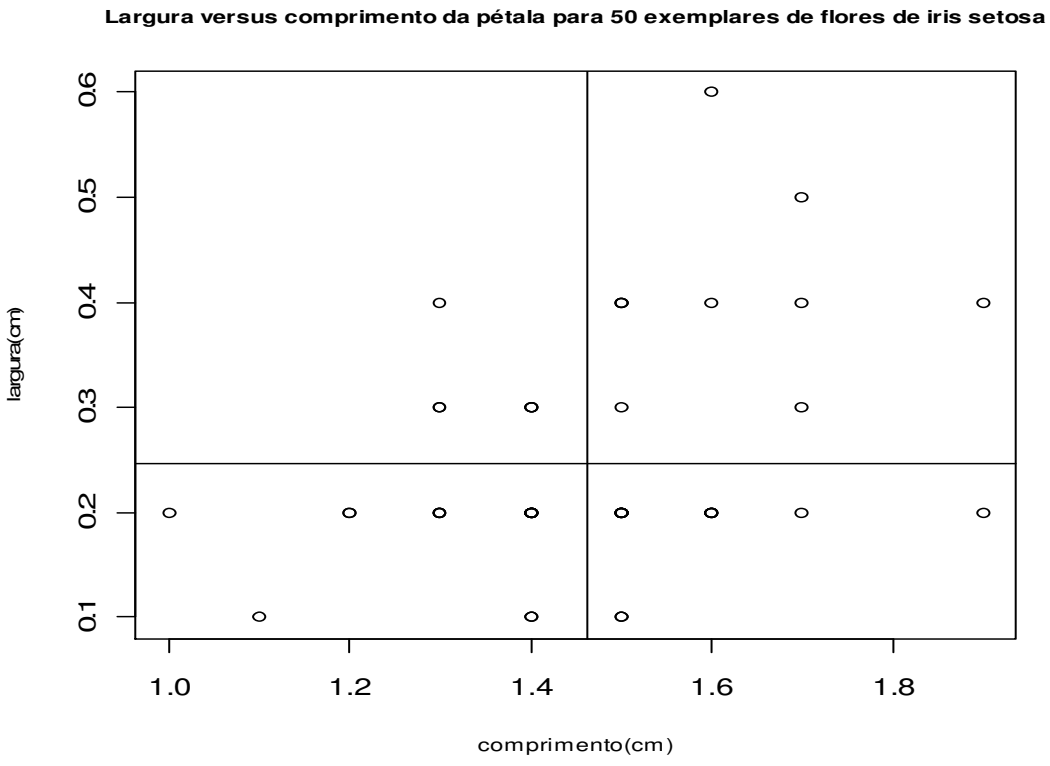
Covariância - viginica

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	0.40434286	0.09376327	0.30328980	0.04909388
Sepal.Width	0.09376327	0.10400408	0.07137959	0.04762857
Petal.Length	0.30328980	0.07137959	0.30458776	0.04882449
Petal.Width	0.04909388	0.04762857	0.04882449	0.07543265

Correlação – virginica

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	1.0000000	0.4572278	0.8642247	0.2811077
Sepal.Width	0.4572278	1.0000000	0.4010446	0.5377280
Petal.Length	0.8642247	0.4010446	1.0000000	0.3221082
Petal.Width	0.2811077	0.5377280	0.3221082	1.0000000

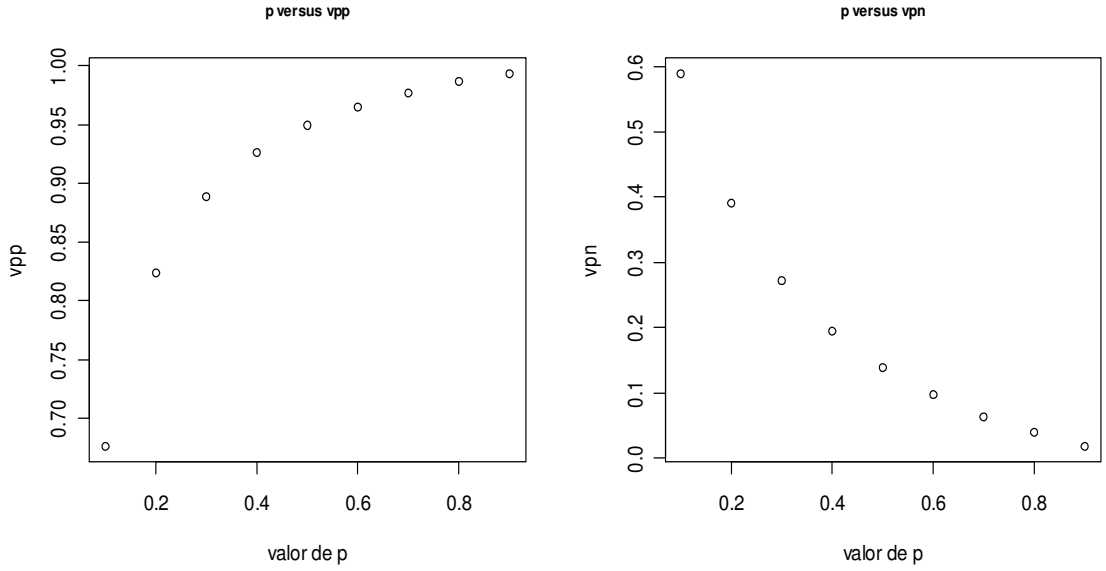
2)



Aula 9

1)

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
VPP	0.6757	0.8242	0.8893	0.9259	0.9494	0.9657	0.9777	0.9868	0.9941
VPN	0.5902	0.3902	0.2718	0.1935	0.1379	0.0964	0.0642	0.0385	0.0175



2)

a)

e	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
VPP	0.0899	0.1000	0.1127	0.1290	0.1509	0.1818	0.2286	0.3077	0.4706
VPN	0.9759	0.9730	0.9692	0.9643	0.9574	0.9474	0.9310	0.9000	0.8182

b)

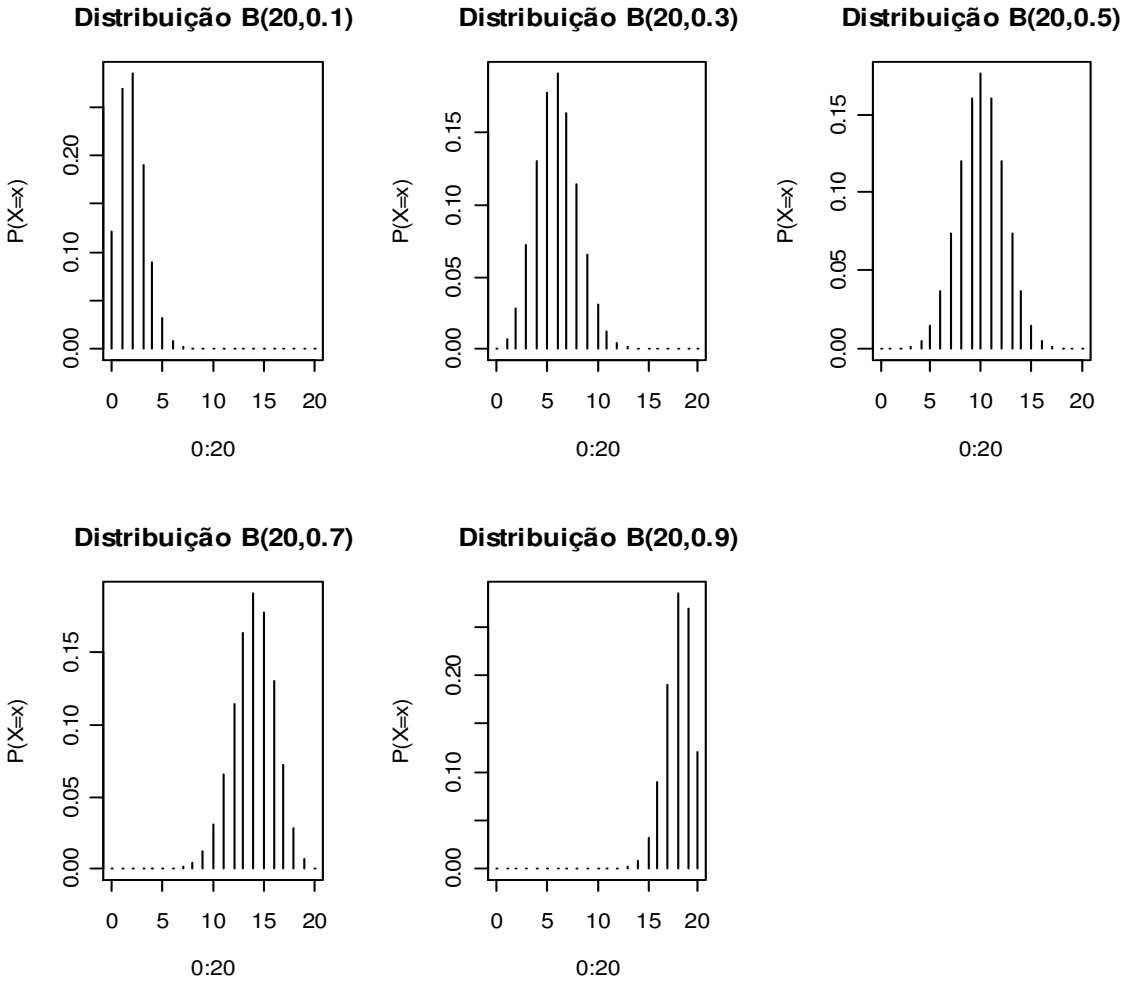
s	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
VPP	0.0526	0.1000	0.1429	0.1818	0.2174	0.2500	0.2800	0.3077	0.3333
VPN	0.6667	0.6923	0.7200	0.7500	0.7826	0.8182	0.8571	0.9000	0.9474

Aula 10

1)

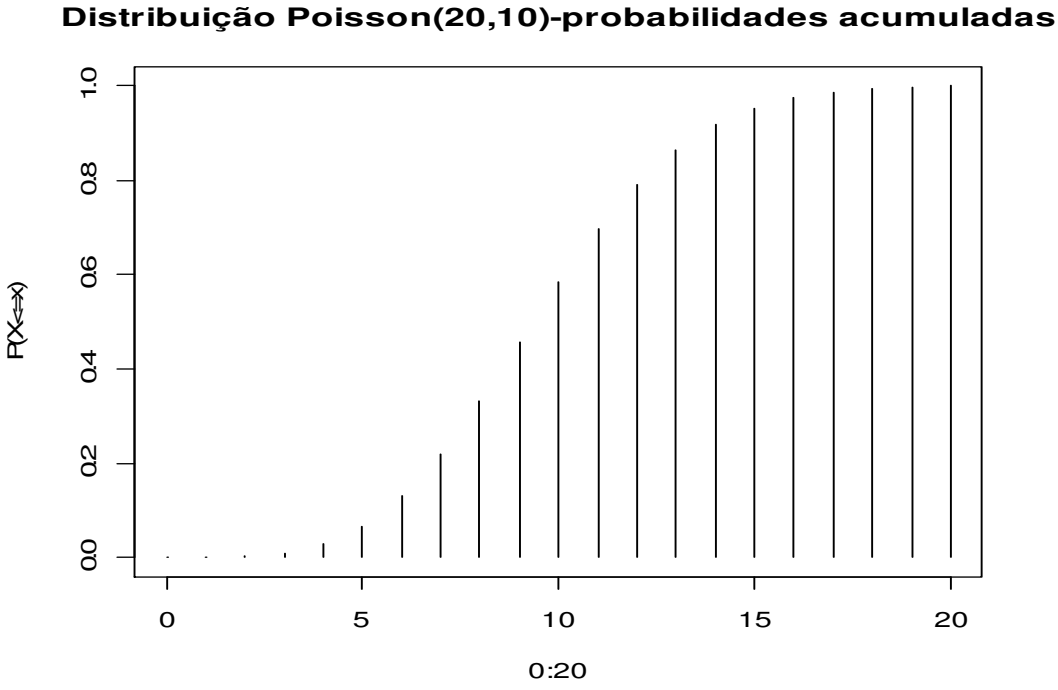
x	20	25	30
simples	0.1146	0.0405	0.0020
acumulada	0.5610	0.9427	0.9986

2)



3) 18, 20 e 22.

4)



5)

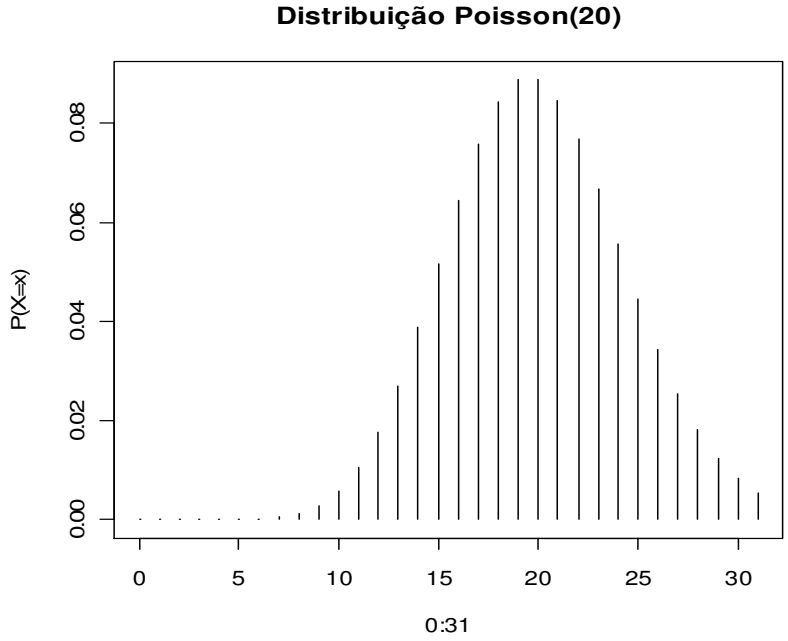
a) 0.05164885, 0.08883532 e 0.04458765

b) 0.1565131, 0.5590926 e 0.8878150

c) 0.61857805 e 0.01347468

d) 17, 20, 23 e 31

e)



Aula 11

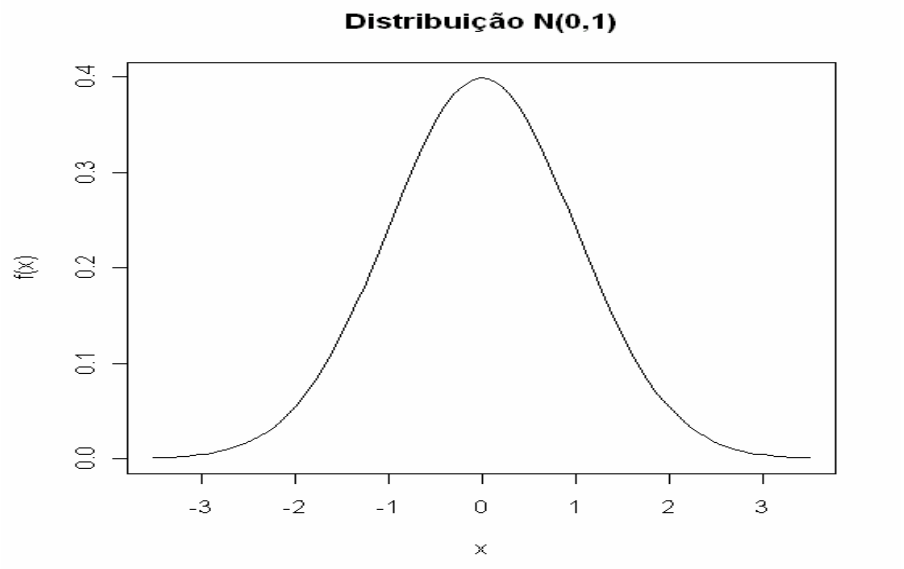
1)

a) 0.9973002, 0.9544997 e 0.6826895

b) 0.5

c) -1.9599640, -1.6448536, -1.2815516, -0.6744898 e 0.0000000

d)



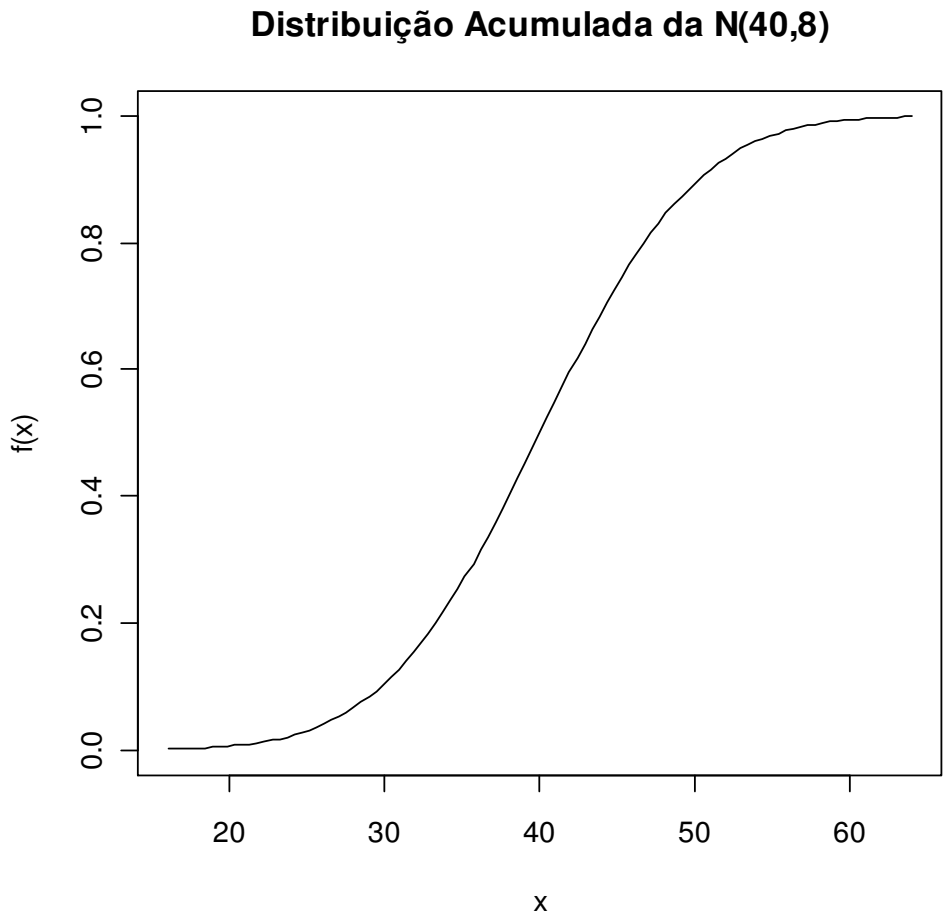
2)

a) 0.03039636, 0.50000000 e 0.89435023

b) 0.8943502, 0.2659855

c) 24.32029, 26.84117, 34.60408 e 40.00000

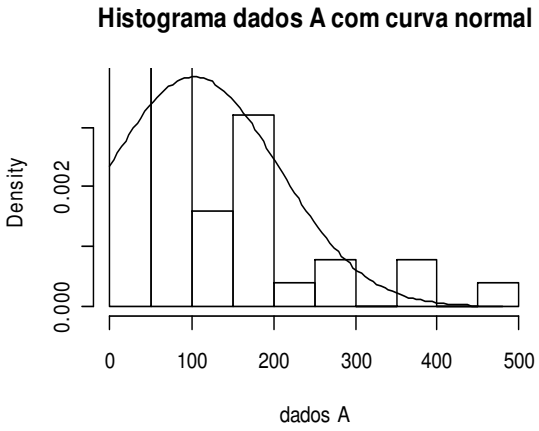
d)

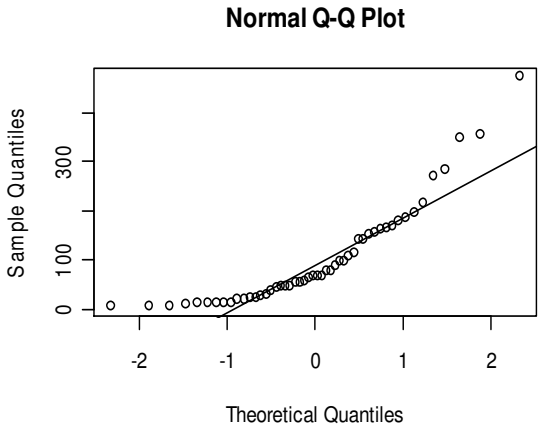


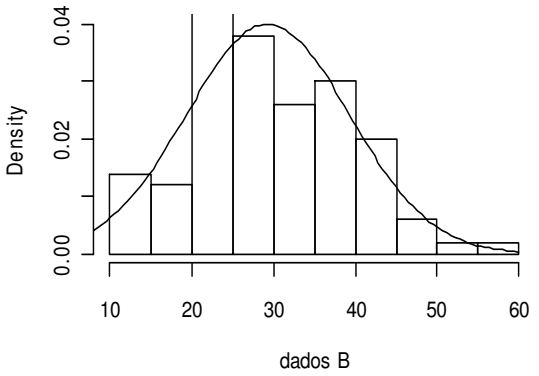
3)

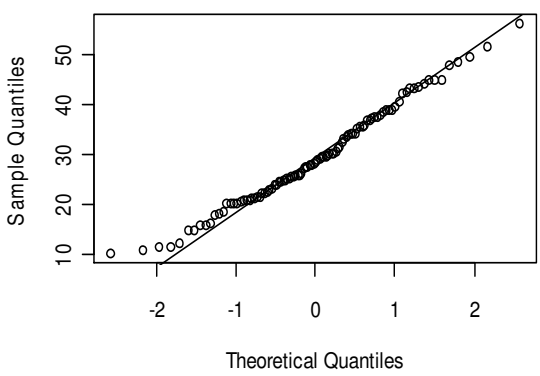
x	prob. exata	prob. aproximada
0	0.00004	0.00026
1	0.00049	0.00120
2	0.00309	0.00453
3	.01235	0.01396
4	0.03499	0.03508
5	0.07465	07184
6	0.12441	0.11986
7	0.16588	0.16296
8	0.17971	0.18052
9	0.15974	0.16296
10	0.11714	0.11986
11	0.07099	0.07184
12	0.03550	0.03508
13	0.01456	0.01396
14	0.00485	0.00453
15	0.00129	0.00120
16	0.00027	0.00026
17	0.00004	0.00005
18	0.00000	0.00001
19	0.00000	0.00000
20	0.00000	0.00000

4)


Histograma dados A com curva normal
Density
0.000
0.002
0
100
200
300
400
500
dados A

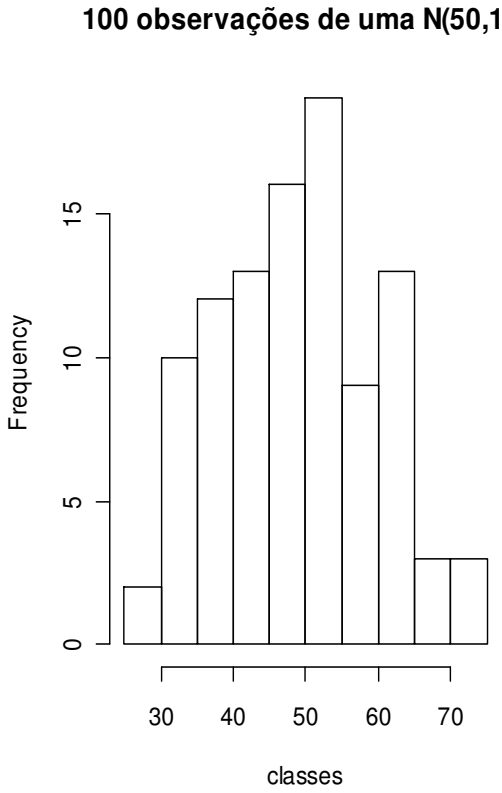

Normal Q-Q Plot
Sample Quantiles
0
100
300
-2
-1
0
1
2
Theoretical Quantiles

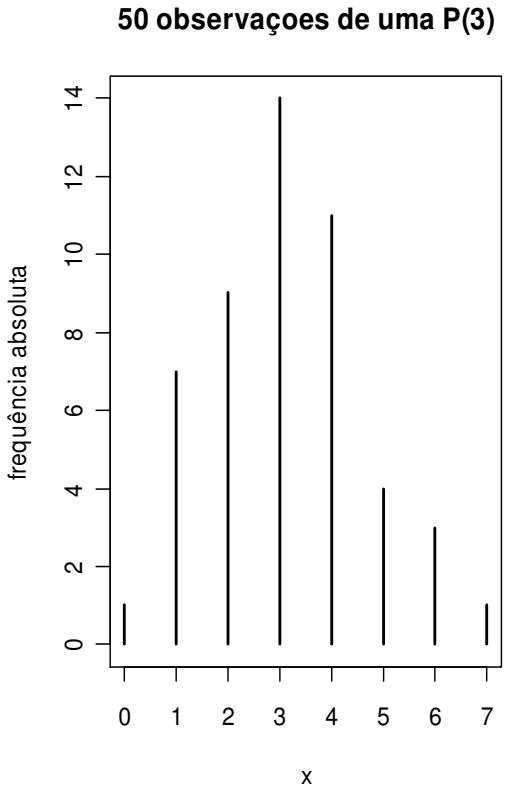

Histograma dados B com curva normal
Density
0.00
0.02
0.04
10
20
30
40
50
60
dados B


Normal Q-Q Plot
Sample Quantiles
10
20
30
40
50
-2
-1
0
1
2
Theoretical Quantiles

Aula 12

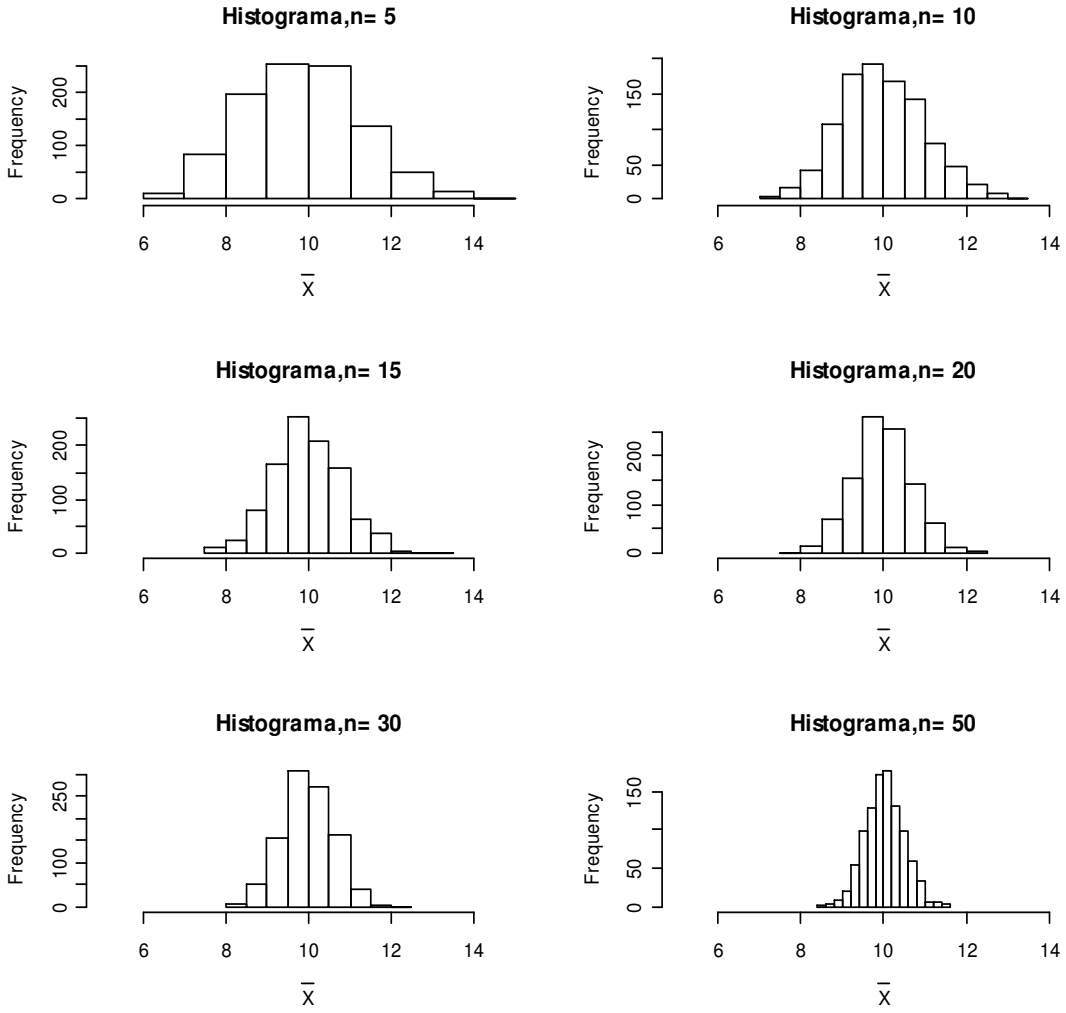
1 e 2)


100 observações de uma N(50,10)
Frequency
0
5
10
15
30
40
50
60
70
classes

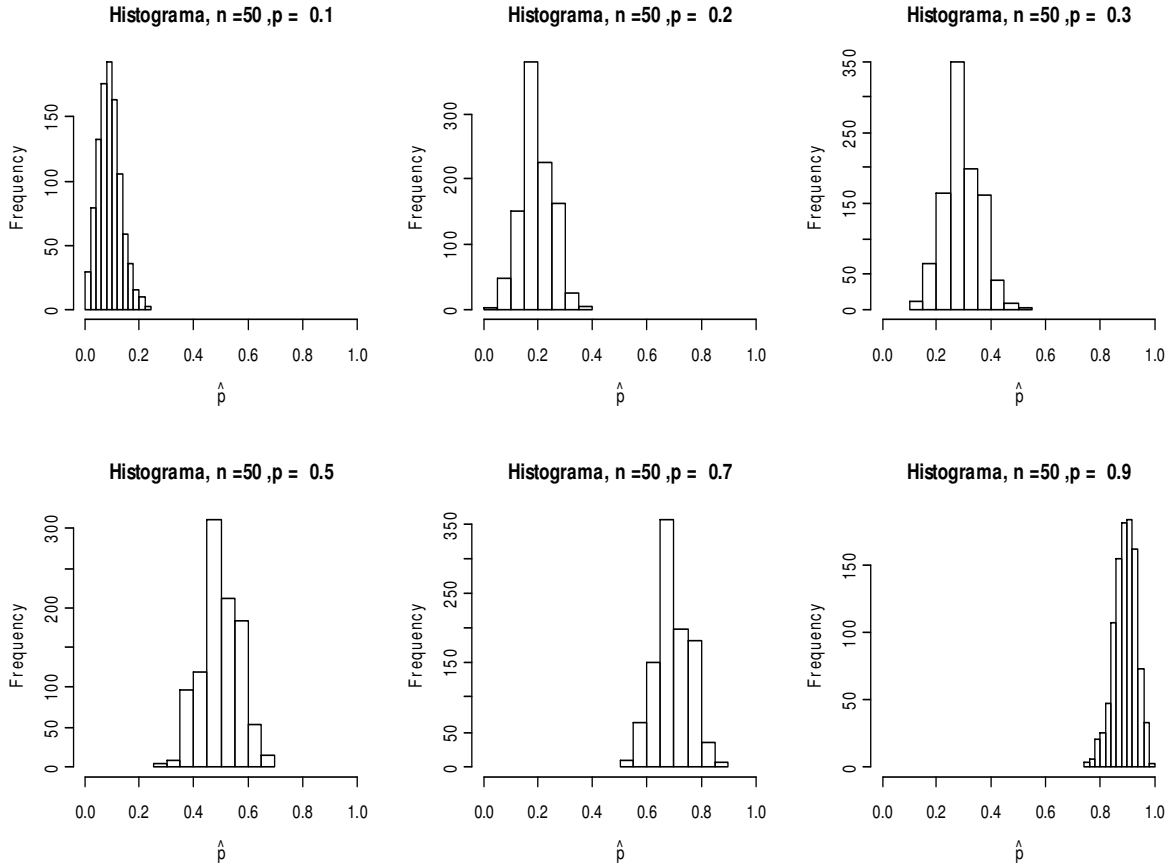

50 observaçoes de uma P(3)
frequência absoluta
0
2
4
6
8
10
12
14
0
1
2
3
4
5
6
7
x

Aula 13

1)


Histograma,n= 5
Frequency
0
100
200
6
8
10
12
14
X
Histograma,n= 10
Frequency
0
50
150
6
8
10
12
14
X
Histograma,n= 15
Frequency
0
100
200
6
8
10
12
14
X
Histograma,n= 20
Frequency
0
100
200
6
8
10
12
14
X
Histograma,n= 30
Frequency
0
100
250
6
8
10
12
14
X
Histograma,n= 50
Frequency
0
50
150
6
8
10
12
14
X

2)



Aula 14

1) a) 0.9680275 b) 0.004486369

b) -1.753050, -1.340606 e -0.691197

Aula 15

1)

a) H0: $\mu = 400$ Ha: $\mu > 400$

```
One Sample t-test

data:  teor
t = 1.1702, df = 19, p-value = 0.1282
alternative hypothesis: true mean is greater than 400
90 percent confidence interval:
 399.6118      Inf
sample estimates:
mean of x
 402.8825
```

Conclusão: não rejeita H0 ao nível de 1%

b) H0: $\mu = 400$ Ha: $\mu < 400$

```
    One Sample t-test

data:  teor
t = 1.1702, df = 19, p-value = 0.8718
alternative hypothesis: true mean is less than 400
99 percent confidence interval:
     -Inf 409.1381
sample estimates:
mean of x
 402.8825
```

Conclusão: não rejeita H0 ao nível de 1%.

2) H0: $p = 0.7$ Ha: $p \neq 0.7$

Teste com correção de continuidade

```
   1-sample proportions test with continuity correction

data:  16 out of 20, null probability 0.5
X-squared = 6.05, df = 1, p-value = 0.01391
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.5573138 0.9338938
sample estimates:
  p
0.8
```

Conclusão: rejeita-se H0 ao nível de 5%.

Teste sem correção de continuidade

```
        1-sample proportions test without continuity correction

data:  16 out of 20, null probability 0.5
X-squared = 7.2, df = 1, p-value = 0.00729
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.5839826 0.9193423
sample estimates:
  p
0.8
```

Conclusão: rejeita-se H0 ao nível de 5%.

Teste exato

```
        Exact binomial test

data:  16 and 20
number of successes = 16, number of trials = 20, p-value = 0.4652
alternative hypothesis: true probability of success is not equal to 0.7
95 percent confidence interval:
 0.563386 0.942666
sample estimates:
probability of success
                   0.8
```

Conclusão: não rejeita H0 ao nível de 5%.

Aula 16

1)

H0: $p_1 = p_2$ x Ha: $p_1 \neq p_2$

Usando a função *prop.test*

```
     2-sample test for equality of proportions with continuity
correction

data:  c(244, 80) out of c(326, 92)
X-squared = 5.3618, df = 1, p-value = 0.02058
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.21146024 -0.03073768
sample estimates:
   prop 1    prop 2
0.7484663 0.8695652
```

Conclusão: rejeita-se H0 ao nível de 5%.

2)

```
  2-sample test for equality of proportions with continuity correction

data:  c(90, 35) out of c(100, 100)
X-squared = 62.208, df = 1, p-value = 3.09e-15
alternative hypothesis: two.sided
95 percent confidence interval:
 0.4295616 0.6704384
sample estimates:
prop 1 prop 2
  0.90   0.35
```

Conclusão: rejeita-se Ho ao nível de 5%.

Aula 17

Aula 18

1)

Ho:Teoria Mendeliana é adequada Há: Teoria Mendeliana não é adequada

```
Chi-squared test for given probabilities

data:  c(315, 108, 101, 32)

X-squared = 0.47, df = 3, p-value = 0.9254
```

Conclusão: não rejeita Ho ao nível de 5%.

2)