Alguns comandos no R. Aula 1 minicurso

10 de outubro de 2013

I congreso Internacional de Estadística. Trujillo-Perú. 2013

Resumo

Muitos programas de computador auxiliam as análises de dados de pesquisas e estas análises recaem em pacotes estatísticos que no mercado são pagos ou alugados com licenças que expiram ou se renovam periodicamente. No entanto a plataforma R, supera estes impasses, proporcionando um programa livre (o seu código fonte esta disponível para alteração pelo usuário) e gratuito (sem custo monetário) usando a linguagem de programação S, do SPLUS. Este programa é confiável pela qualidade dos seus resultados, e pela estabilidade em ambientes como Linux e Windows. Atualmente o R disponibiliza mais de 1000 pacotes para serem instaladas na sua biblioteca, porem cada área utiliza funções específicas, por exemplo, em econometria, podemos esta interessados inicialmente nos pacotes: "lmtest", "quantreg", "KernSmooth", "car", "strucchange", "AER", além do pacote padrão "stats". Para séries temporais, podemos acrescentar os pacotes: "tseries", "forecast", "fracdiff", "TSA ", "vars", todos estes pacotes podem ser facilmente baixados (download) via internet, além disso os seus tutoriais estão disponíveis no formato htlm, pdf, como alternativa use a ajuda (help) do R.

1 Iniciando o R

Para instalar o R, temos que estar conectados a internet, e baixar diretamente da pagina http://www.r-project.org/. Se estamos usando a plataforma Windows baixar o R para Windows, atualmente a versão é 3.0.1, disponível desde 25/09/2013.

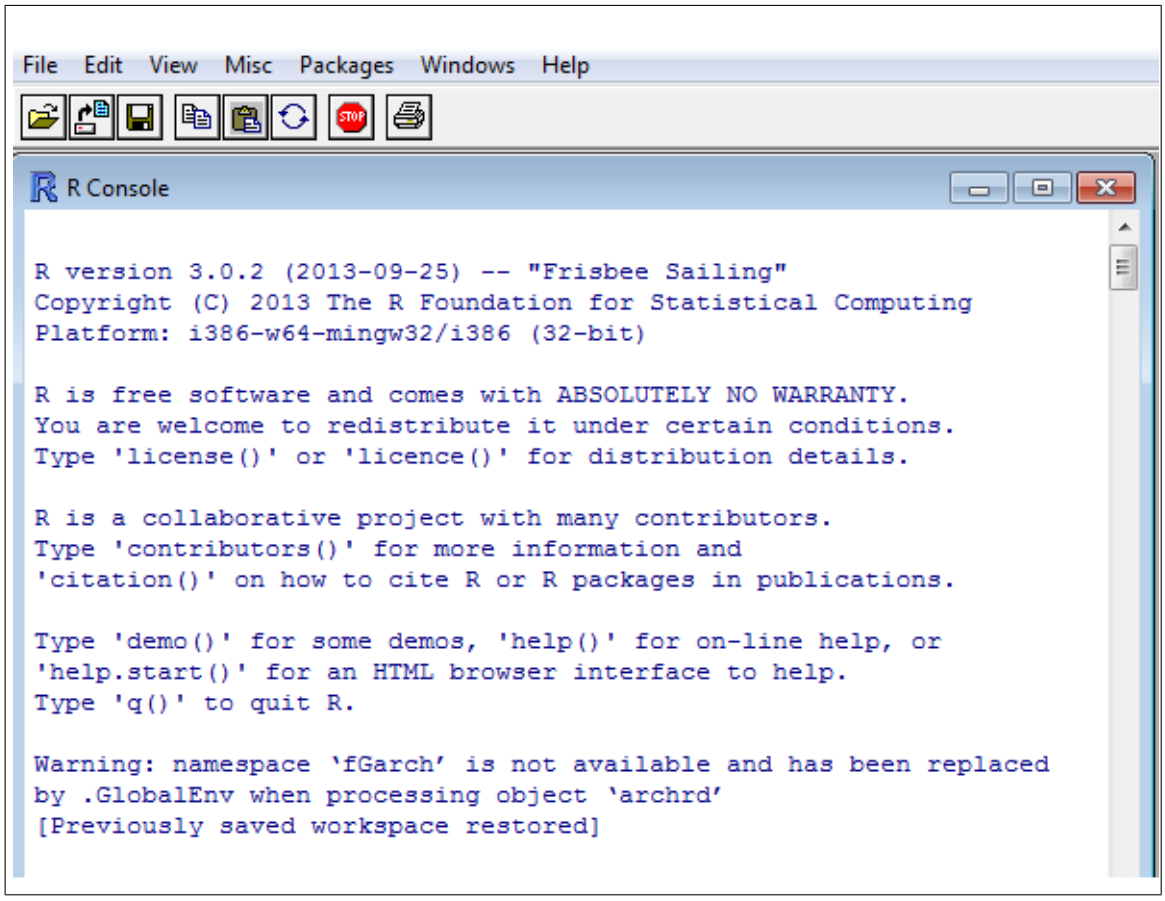

Figura 1. Visualização na inicialização do R

Quando instalamos o R uma lista de pacotes padrão já estão na biblioteca, os quais são visualizados com a função search():

```
> search()
[1] ".GlobalEnv"        "package:stats"     "package:graphics"
[4] "package:grDevices" "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"         "package:base"
```

Para instalar pacotes do R uma lista completa esta disponível com o comando:

```
install.packages()
```

1.1 Dados permanentes e removendo objetos

Durante uma sessão no R, objetos são criados e guardados por nomes, para verificar estes objetos usamos o comando ou função

```
objects( ) # ou >ls( )
```

```
[1] "a1"               "a2"             "aa"          "acei"
[5] "ads"             "aju"           "aju1"        "ajuste"
[9] "ajuste1"      ajuste2"       "ajustehw"           "ara"
```

Para remover objetos antigos ou sem uso utilizamos o comando rm():

```
rm(a1, a2, aa, acei, ads, aju)
```

Para detectar os pacotes na biblioteca local

```
library()
```

Para instalar o pacote **urca** disponível na biblioteca:

```
install.packages("urca")
```

Um pacote instalado tem que ser chamado no R para realizar a tarefa. Para chamar o pacote **urca** já instalado.

```
library(urca)
```

Para verificar os pacotes na biblioteca externa.

```
update.packages()
```

Para ver a lista de pacotes instalados na biblioteca local:

```
installed.packages()
```

1.2 Manipulação de números e vetores

Para declarar números, exemplo $v = \begin{bmatrix}10\end{bmatrix}$, realizamos o seguinte comando:

```
v=10
v
[1] 10
```

Alertamos que no R, o símbolo $<-$ é equivalente ao símbolo $=$
Considere o seguinte vetor com os números 1, 3, 3.4, 6, 99, 3, com o nome $q = \begin{bmatrix}1 & 3 & 3.4 & 6 & 99 & 3\end{bmatrix}$, usamos:

```
q=c(1,3,3.4,6,99,3)
```

Comandos de operações algebraicas são usados com os sinais convencionais, sendo: + para soma; - para a diferença; * para multiplicação e / para divisão.
Assim para inverter q, podemos usar:

```
 1/q
[1] 1.0000000 0.3333333 0.29411765 0.1666667 0.0101010 0.3333333
```

No R podemos utilizar comando lógicos como verdadeiro ou falso para os valores de q maiores do que 3, assim:

```
q>3
[1] FALSE FALSE TRUE TRUE TRUE FALSE
```

para restringir valores de q maiores do que 3:

```
q[q>3]
[1] 3.4 6 99
```

Podemos converter números em caracteres ou dígitos assim:

```
z2=c(0:9) # equivalente a z2=0:9
digit=as.character(z2)
digit
[1] "0" "1" "2" "3" "4" "5" "6" "7" "8" "9"
```

Para retornar dígitos em números com o seguinte comando:

```
d=as.integer(digit)
d
 [1] 0 1 2 3 4 5 6 7 8 9
```

Para preparar uma variável para ingressar seus valores por exemplo

```
e=numeric()
e[3]=17
```

Onde o terceiro elemento é 17 e os dois primeiros ainda não definidos:

```
e
[1] NA NA 17
```

Podemos ainda declarar os outros elementos de e.

Para Construir um vetor qualitativo, sendo na ordem 10 homens e 5 mulheres usamos a funçaõ rep():

```
w=c(rep("homem",10), rep("mulher",5))
w
 [1] "homem"  "homem"  "homem"  "homem"  "homem"  "homem"  "homem"
 [8] "homem"  "homem"  "homem"  "mulher" "mulher" "mulher" "mulher"
[15] "mulher"
```

Considere o tamanho do calçado no pé de cada individuo em w:

```
pé=c(40,42,39,41,44,40,42,41,44,40,35,35,37,36,37)
```

Podemos identificar a distribuição de frequências de w e pé da seguinte forma:

```
table(w,pe)
        pe
w        35 36 37 39 40 41 42 44
  homem   0  0  0  1  3  2  2  2
  mulher  2  1  2  0  0  0  0  0
```

Para construir gráficos para as tabelas usamos as funções:

```
plot(table(w,pe), main="Frequências",xlab="sexo",ylab="número")
barplot(table(w,pe), main="Frequências", ylab="casos",xlab="pé")
legend("topleft", legend=c("mulher","homem"), col=c(8,1),lty=c(1,1))
```

Seus respectivo gráficos são:

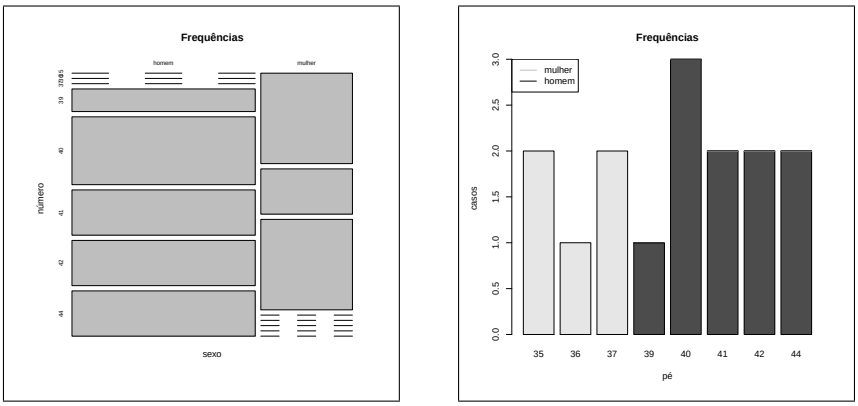

Figura 2. Descrição do calçado por sexo

1.3 Funções matemáticas

Para utilizar as funções matemáticas, usamos os comandos:

```
> history()   # comandos usados previamente
> length(x)   # número de elementos do vetor ou lista x
> log(x)      # log na base e de x
> exp(x)      # antilogaritmo de x
> log(x,n)    # log na base n de x
> log10(x)    # log na base 10 de x
> sqrt(x)     # raiz quadrada de x
> factorial(x)# factorial de x(x!)
> choose(n,x) # combinatoria:  n!/(x!(n-x)!)
> gamma(x)    # x (x-1)!, x inteiro
> floor(x)    # menor inteiro de x
> trunc(x)    # escolha do menor inteiro
> abs(x)      # valor absoluto de x
> cos(x)      # cosseno de x
> sin(x)      # seno de x
> tan(x)      # tangente de x
> acos(x) # inversa cosseno
> asin(x) # inversa seno
> atan(x)     # inversa tangente
> acosh(x)    # inversa cosseno hiperbólica
> asinh(x)    # inversa seno hiperbólica
> atanh(x)    #inversa tangente hiperbólica
> round(x, digits=2)  # aproximação a duas casas decimais
> signif(x, digits=6) # x com notação científica de 6 dígitos
```

1.4 Funções estatísticas

```
> max(x)     # valor máximo de x
> min(x)     # valor mínimo de x
> sum(x)     # soma total dos valores de x
> mean(x)    # média aritmética de x
> median(x)  # mediana de x
> range(x)   # vetor de mínimo e máximo de x
> summary(x) # resumo de estatísticas de locação de x
> var(x)     # variância de x
> cor(x,y)   # correlação de x e y
> sort(x)    # ordena em forma crescente os elementos de x
> rank(x)    # vector de filas em x
> order(x)   # valores originais de x
> quantile(x)# vetor contendo os quartis de x
> cumsum(x)  # vetor contendo a suma total elemento a elemento
> cumprod(x) # vetor contendo o produto elemento a elemento
> cummax(x)  # vetor não decrescente de x, substituindo valores
> cummin(x)  # vetor não crescente de x, substitui os valores
               pelo mínimo
> pmax(x,y,z)# vetor contendo o vetor maior entre x y ou z, e
               substituindo o máximo em cada posição.
> pmin(x,y,z)# vetor contendo o vetor maior entre x y ou z,
               e substituindo o mínimo em cada posição.
> colMeans(x)# calcula a média por colunas na matriz x
> colSums(x) # calcula os totais por coluna na matriz x
> rowMeans(x)# calcula a média por linhas na matriz x
> rowSums(x) # calcula os totais por linhas na matriz x
```

1.5 Cores

O R disponibiliza cores usando a função palette() com 8 cores, outra função é colours(), e que também podem ser identificados por um número já especificado por exemplo:

```
teta=(1:20)*0.92
plot(teta,sin(teta), col="red", pch=16,cex=2) #ou
plot(teta,sin(teta), col=2, pch=16,cex=2)
```

Onde cex, representa o carácter de expansão da bolinha (quanto maior o número maior a bolinha). Para incrementar várias cores no mesmo gráfico podemos realizar o seguinte comando:

```
plot(teta,sin(teta), col=1:20, pch=16,cex=2)
```

Para modificar a bolinha apenas modificamos pch:

```
plot(teta,sin(teta), col=1:20, pch="+",cex=2)
```

Cores podem também ser incrementadas fora do corpo do gráfico

```
plot(teta,sin(teta), main="seno", col.main=4, col=1:20, pch="+",cex=2)
```

Outras alterações de cores podem ser construídas com:

```
col.main    # cor para o título do gráfico
col.axis    # cor para os números dos eixos
col.lab     # cor para os títulos dos eixos
col.sub     # cor para o subtítulo do gráfico
```

A fonte pode ser alterada da seguinte maneira:

```
font.main    # fonte para o título do gráfico
font.axis    # fonte para os números dos eixos
font.lab     # fonte para os títulos dos eixos
font.sub     # fonte para o subtítulo do gráfico
```

Onde: 1 é texto normal, 2 preto, 3 itálico, 4 preto itálico para simbolo fonte. A orientação dos números pode ser alterada com o comando las= ; onde: 1 horizontal a x, 2 perpendicular a x, 3 perpendicular a ambos eixos).

1.6 Fatores ordenados e desordenados

exemplo

```
estados=c("PE","AL","SE","BA","PB","BA","PB","AL","SE","BA","PB",
+ "SE", "BA","SE","BA")
estadosf=factor(estados)
estadosf
 [1] PE AL SE BA PB BA PB AL SE BA PB SE BA SE BA
Levels: AL BA PB PE SE
levels(estadosf)
[1] "AL" "BA" "PB" "PE" "SE"
salarios=c(60,49,59,56,49,48,56,59,60,69,66,58,47,58,68)
```

A função tapply é usada para reordenar vetores, calculando uma característica dos níveis, no exemplo suponha que os salários médio e o desvio padrão por cada trabalhador por dia nos estados seja dado a seguir:

```
mestados=tapply(salarios,estadosf,mean)  #média
mestados
   AL    BA    PB    PE    SE
54.00 57.60 57.00 60.00 58.75
destados=tapply(salarios,estadosf,sd)   #desvio padrão
destados
        AL         BA         PB         PE         SE
 7.0710678 10.5498815  8.5440037         NA  0.9574271
```

Construímos a função do coeficiente de variação de duas formas distintas, e incluímos no comando tapply:

```
cv=function(x) sd(x)*100/mean(x)
cvestados=tapply(salarios,estadosf,cv)
cvestados
       AL        BA        PB        PE        SE
13.094570 18.315767 14.989480        NA  1.629663

cv=function(x) sqrt(var(x))*100/mean(x)
cvestados=tapply(salarios,estadosf,cv)
cvestados
       AL        BA        PB        PE        SE
13.094570 18.315767 14.989480        NA  1.629663
```

1.7 Vetores e Matrizes

Um vetor pode ser considerado como a entrada de uma coleção de dados, sendo estes numéricos ou não. R facilita a manipulação e construção de vetores e matrizes. Considere a construção de 30 números aleatórios Uniformes em duas matrizes de 3 linhas por 5 colunas:

```
z=runif(30)
dim(z)=c(3,5,2)
z
, , 1

          [,1]      [,2]       [,3]       [,4]      [,5]
[1,] 0.9500783 0.9262226 0.05847740 0.34257312 0.3718370
[2,] 0.8591887 0.7057183 0.20592054 0.03569694 0.6709746
[3,] 0.6806468 0.6270572 0.01570158 0.21650180 0.5170189
```

```
, , 2

          [,1]      [,2]       [,3]      [,4]      [,5]
[1,] 0.3539723 0.5918548 0.37104534 0.2148440 0.4169542
[2,] 0.8895178 0.5751237 0.69399721 0.3808024 0.3872492
[3,] 0.8036340 0.7065130 0.03400331 0.1593949 0.2306890
```

1.8 Indexando matrizes

```
x=array(1:20,dim=c(4,5))
x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20

i=array(c(1:3,3:1),dim=c(3,2))
i
     [,1] [,2]
[1,]    1    3
[2,]    2    2
[3,]    3    1
> x[i]
[1] 9 6 3

# x[1,3]=9; x[2,2]=6; x[3,1]=3
```

O índice i indica os valores da matriz x a serem extraídos, temos

```
x[i]=0  # Substituímos x[i] por zero.
x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    0   13   17
[2,]    2    0   10   14   18
[3,]    0    7   11   15   19
[4,]    4    8   12   16   20
```

Construindo uma matriz 3×3 com elementos de 1 ao 9.

```
c=matrix(c(1:9),3,3)
```

```
c
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

1.9 Sequências

Sequencias são construídas, ingressando o primeiro elemento, o ultimo elemento, e o espaçamento.

```
b=seq(10,14, by=.5)
b
[1] 10.0 10.5 11.0 11.5 12.0 12.5 13.0 13.5 14.0
a=seq(1,9)
ab=a%*%b # multiplicando vetores
ab
     [,1]
[1,]  570
ab=outer(a,b,"*")
ab
      [,1] [,2] [,3]  [,4] [,5]  [,6] [,7]  [,8] [,9]
 [1,]   10 10.5   11  11.5   12  12.5   13  13.5   14
 [2,]   20 21.0   22  23.0   24  25.0   26  27.0   28
 [3,]   30 31.5   33  34.5   36  37.5   39  40.5   42
 [4,]   40 42.0   44  46.0   48  50.0   52  54.0   56
 [5,]   50 52.5   55  57.5   60  62.5   65  67.5   70
 [6,]   60 63.0   66  69.0   72  75.0   78  81.0   84
 [7,]   70 73.5   77  80.5   84  87.5   91  94.5   98
 [8,]   80 84.0   88  92.0   96 100.0  104 108.0  112
 [9,]   90 94.5   99 103.5  108 112.5  117 121.5  126

ab=a%*%t(b)
ab
      [,1] [,2] [,3]  [,4] [,5]  [,6] [,7]  [,8] [,9]
 [1,]   10 10.5   11  11.5   12  12.5   13  13.5   14
 [2,]   20 21.0   22  23.0   24  25.0   26  27.0   28
 [3,]   30 31.5   33  34.5   36  37.5   39  40.5   42
 [4,]   40 42.0   44  46.0   48  50.0   52  54.0   56
 [5,]   50 52.5   55  57.5   60  62.5   65  67.5   70
```

```
 [6,]   60 63.0   66  69.0   72  75.0   78  81.0   84
 [7,]   70 73.5   77  80.5   84  87.5   91  94.5   98
 [8,]   80 84.0   88  92.0   96 100.0  104 108.0  112
 [9,]   90 94.5   99 103.5  108 112.5  117 121.5  126
```

```
d=outer(0:9,0:9)
d
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0    0    0    0    0    0    0    0    0     0
 [2,]    0    1    2    3    4    5    6    7    8     9
 [3,]    0    2    4    6    8   10   12   14   16    18
 [4,]    0    3    6    9   12   15   18   21   24    27
 [5,]    0    4    8   12   16   20   24   28   32    36
 [6,]    0    5   10   15   20   25   30   35   40    45
 [7,]    0    6   12   18   24   30   36   42   48    54
 [8,]    0    7   14   21   28   35   42   49   56    63
 [9,]    0    8   16   24   32   40   48   56   64    72
[10,]    0    9   18   27   36   45   54   63   72    81
```

1.10 Multiplicando matrizes

Para a construção de matrizes o preenchimentos das células ou casas é feita coluna a coluna, como padrão, exemplo:

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

podemos construir esta mesma matriz preenchendo as casas linha por linha, da seguinte forma:

```
at=matrix(c(1:9),3,3,byrow=T)
```

```
at
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

Uma alternativa seria usar a função aperm():

```
b= aperm(a,c(2,1))
b
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9

a*b              # multiplicação elemento a elemento
     [,1] [,2] [,3]
[1,]    1    8   21
[2,]    8   25   48
[3,]   21   48   81

a%*%b            # multiplicação das matrizes a e b
     [,1] [,2] [,3]
[1,]   66   78   90
[2,]   78   93  108
[3,]   90  108  126

crossprod(a,b)  # produto da transposta de a por b

     [,1] [,2] [,3]
[1,]   30   36   42
[2,]   66   81   96
[3,]  102  126  150

t(a)%*%b
     [,1] [,2] [,3]
[1,]   30   36   42
[2,]   66   81   96
[3,]  102  126  150
```

1.11 Inversão de matrizes

```
c=matrix(c(3,2,4,5,7,2,1,5,9),3,3)
d=matrix(c(2,2,3,5,8,2,8,5,8),3,3)
c
```

```
     [,1] [,2] [,3]
[1,]    3    5    1
[2,]    2    7    5
[3,]    4    2    9
> solve(c)
            [,1]        [,2]        [,3]
[1,]  0.36551724 -0.29655172  0.12413793
[2,]  0.01379310  0.15862069 -0.08965517
[3,] -0.16551724  0.09655172  0.07586207

solve(c,d)  # inverte a matriz c e multiplica por d:
           [,1]        [,2]       [,3]
[1,] 0.51034483 -0.29655172  2.4344828
[2,] 0.07586207  1.15862069  0.1862069
[3,] 0.08965517  0.09655172 -0.2344828

> solve(c)%*%d # forma alternativa:
           [,1]        [,2]       [,3]
[1,] 0.51034483 -0.29655172  2.4344828
[2,] 0.07586207  1.15862069  0.1862069
[3,] 0.08965517  0.09655172 -0.2344828
```

1.12 Comandos cbind() e rbind()

É possível formar novas matrizes com as já existentes, usando a função **cbind()** e **rbind()**, a primeira construí matrizes em colunas a segunda em linhas.

```
cbind(c,d)
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    3    5    1    2    5    8
[2,]    2    7    5    2    8    5
[3,]    4    2    9    3    2    8

rbind(c,d)
     [,1] [,2] [,3]
[1,]    3    5    1
[2,]    2    7    5
[3,]    4    2    9
[4,]    2    5    8
```

```
[5,]    2    8    5
[6,]    3    2    8

cbind(1,c)
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    1
[2,]    1    2    7    5
[3,]    1    4    2    9

rbind(1,d)
     [,1] [,2] [,3]
[1,]    1    1    1
[2,]    2    5    8
[3,]    2    8    5
[4,]    3    2    8
```

1.13 Autovalores e autovetores

```
ev=eigen(c)  # calcula autovalores e autovetores.
ev
$values
[1] 13.265694+0.000000i  2.867153+1.646172i  2.867153-1.646172i

$vectors
             [,1]                   [,2]                   [,3]
[1,] 0.3799264+0i  0.7884141+0.0000000i  0.7884141+0.0000000i
[2,] 0.6480231+0i  0.0745376+0.3051027i  0.0745376-0.3051027i
[3,] 0.6600924+0i -0.4774265-0.2276482i -0.4774265+0.2276482

det(c)
[1] 145
det(d)
[1] -57
```

1.14 Data frame ou construtor de dados

Suponha um conjunto de dados com três variáveis: sexo (1=homem, 0=mulher), peso em quilogramas e salário em reais por dia de 12 indivíduos. Para calcular a média, o resumo de estatísticas descritivas e um gráfico (plot)

realizamos os seguintes comandos.

```
ind=data.frame(sex=c(1,0,0,1,0,0,0,1,1,1,1,0),
+ peso=c(67,60,62,69,54,59,57,67,60,65,59,55),
+ salario=c(59,40,45,65,55,45,48,67,65,68,62,45))
apply(ind,2,mean)
     sex     peso  salario
 0.50000 61.16667 55.33333
summary(ind)
library(fields)
bplot(ind$peso,by=ind$sex, xlab="homem=1  mulher=0",
+ ylab="peso", col=2)
boxplot(peso~sexo) #forma alternativa
```

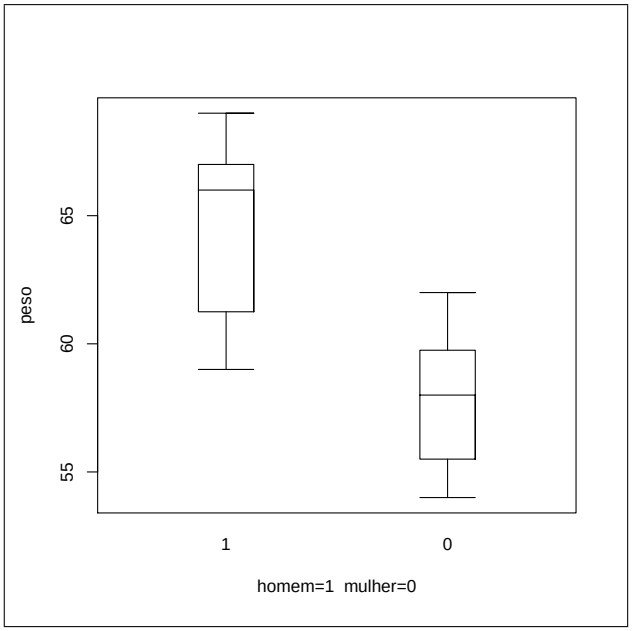

Figura 3. Relação do peso (kg) com o sexo

1.15 Funções attach() e detach()

Usamos estas funções para disponibilizar: attach(); ou remover: detach(), as variáveis internas de um conjunto de dados, pois os comandos:

```
ind$peso
ind$sex
```

Nem sempre é conveniente e de fácil manipulação. Veja no comando a seguir:

```
attach(ind)
peso
 [1] 67 60 62 69 54 59 57 67 60 65 59 55
salário
[1] 59 40 45 65 55 45 48 67 65 68 62 45
sex
 [1] 1 0 0 1 0 0 0 1 1 1 1 0

detach(ind)
salário
Erro: objeto 'salário' não encontrado
```

A função attach e detach, pode ser usada no somente em arquivo de dados dos pacotes do R, se não também para listas e data frame

1.16 Função read.table()

Um amplo conjunto de arquivos com diferentes extensões pode ser lido por esta função.

1.16.1 Arquivos com extensão txt

Considere as seguintes variáveis $x1$: idade dos funcionários, $x2$: sexo (1 homem; 0 mulher), $x3$: anos de experiência na função, $x4$: estado civil, $x5$: número de filhos, $x6$: nota de $0-10$ de um teste de avaliação de conhecimento na sua área, $x7$: teste psicotécnico de 0 a 50, $x8$: satisfação com a vida pessoal. estes dados em bloco de notas com extensão txt, obtidos de Mingoti, 2005. Ver dados na integra no apêndice.

```
funcao=read.table("func.txt",header=T)
funcao
```

```
    grupo idade sexo ecivil nfilhos aexper tpsico tconhec satisf
1       0    24    1      1       0      2    7.7    36.0      1
2       0    29    1      1       0      4    7.9    36.0      1
3       0    38    1      0       1      6    8.6    40.8      0
.............................................................
107     0    24    1      1       0      2    7.3    36.0      0
108     0    27    1      0       0      3    7.4    36.0      1
109     1    42    1      0       2     14    9.9    44.0      1
```

Considere os dados de consumo de água de uma residência durante o período de 12/05 a 04/11. Com as seguintes variáveis: valor em reais, consumo em m^3, dias de consumo, e construindo o imóvel (1:sim, 2:não). Dados próprios do autor. Ver dados na integra no apêndice.

```
ca=read.table("consumoagua2.txt",header=T) #disponível em meus documentos
ca
   dado valor consumo diasconsumo Construindo
1     1 14.76      11          30           0
2     2 13.47      10          31           0
.........................................
64   64 34.39      14          31           1
65   65 22.21      11          32           0
```

Observação: use o comando choose(), para escolher um conjunto de dados de um diretório especifico no computador, podemos usar:

```
data<-read.table(file.choose(),header=T)
```

1.17 Script com extensão .R

Exemplo com extensão R, lida por read.table com dados do próprio R

```
exemplo=read.table(file.choose(),header=T) #dadosr.R em aula 3
exemplo
  dado valor
1    1  11.5
2    2  11.7
3    3  11.8
4    4  11.9
5    5  12.7
```

1.18 Função scan()

A função scan também pode incluir elementos no R:

```
scan()
1: 12
2: 13
3: 14
```

```
4: 15
5:
Read 4 items
[1] 12 13 14 15
```

Use a função scan(), para ingressar dados diretamente na plataforma R.

1.19 Função sample

Usa-se para escolher aleatoriamente uma amostra de um conjunto de observações ou uma base de dados. Como exemplo considere uma amostra de 8 observações dos primeiros 20, do banco de dados de consumo de água (ca)

```
set.seed(10)
ca20=ca[sample(1:20,8),]
ca20
   dado valor consumo diasconsumo Construindo
11   11 31.53      16          31           1
6     6 13.47      10          32           0
8     8 13.47      10          30           0
12   12 37.55      18          32           1
2     2 13.47      10          31           0
4     4 52.75      22          33           1
15   15 37.55      18          32           0
14   14 16.48      11          30           0
```

1.20 Rcmdr

Para importar dados do SPSS, STATISTICA, MINITAB, EXCEL, TXT, etc, podemos usar o pacote livre **Rcmdr**, já disponível na biblioteca.

```
library(Rcmdr)
```

este pacote trabalha via janelas para maior comodidade.

1.21 Dados disponíveis no R

R possuí mais de 100 conjunto de dados alocados no pacote **dataset** que podem ser chamados diretamente (sem extensão) com a função data.

```
data()
```

No R pode-se acessar a conjunto de dados de outros pacotes disponíveis na sua biblioteca. Se por acaso esta instalado o pacote **lmtest**, o conjunto de dados ativo neste pacote pode ser chamado da seguinte maneira:

```
data(package="lmtest")
```

1.22 Modificando e editando dados

Considere o conjunto de dados anterior. Para modificar este conjunto será disponibilizado uma tela de editor de dados a seguir:

```
xnovo=edit(wages,package="lmtest")
```

Para editar um conjunto de dados novo, podemos usar a função **edit** da seguinte forma:

```
xn=edit(data.frame())
```

1.23 Exportando dados do R para o Excel

Para exportar uma coluna de dados, por exemplo o valor pago no consumo de água, executamos no R

```
writeClipboard(as.character(valor))
```

Na tela do Excel, usar o comando colar. Suponha que esta interessado em exportar o conjunto de dados **ca**, consumo de água numa residência, disponíveis no R, para transferir este conjunto de dados, use o seguinte comando

```
write.table(ca,"clipboard",sep="\t",col.names=NA)
```

Na tela do Excel faça Ctrl+V ou um click em colar.

1.24 Funções: apply, tapply, sapply e lapply

A função apply, calcula medidas estatísticas por linhas ou por colunas, e a função sapply calcula apenas por coluna. A diferença da função apply da função tapply é que esta última calcula medidas estatísticas por estratos. Considere os dados de consumo de água de uma residência:

```
apply(ca,2,mean)
       dado       valor     consumo diasconsumo Construindo
 33.0000000  25.5690769  12.6153846  30.7076923   0.2615385
```

```
tapply(valor,Construindo,mean)
       0        1
21.15271 38.03882
sapply(ca,mean)
       dado       valor     consumo diasconsumo Construindo
 33.0000000  25.5690769  12.6153846  30.7076923   0.2615385
lapply(ca,mean) # verificar
```

1.25 Distribuições de probabilidade

R proporciona uma serie de funções para variáveis aleatórias estatísticas conhecidas, dentro destas funções temos distribuição, aleatorização, probabilidade e quantidade.

Distribuição	**nome no R**	**argumentos adicionais.**
beta	beta	shape1,shap2,npc
binomial	binom	size,prob
cauchy	cauchy	location,scale
chi-quadrado	chisq	df,npc
exponencial	exp	rate
F	f	df1,df2,npc
gamma	gamma	shape,scale
geometric	geom	prob
hypergeometric	hyper	m,n,k
log-normal	lnorm	meanlog,sdlog
logistic	logis	location,scale
negativa binomial	nbinom	size,prob
normal	norm	mean,sd
poisson	pois	lambda
t-student	t	df,ncp
uniforme	unif	min,max
weibull	weibull	shape,scale
wilcoxon	wilcoxon	m,n

Nomes iniciando com "d"calculara a densidade da função, "p"para calcular a probabilidade ou função acumulada, "q"para o valor quantílico e "r"para simulação de números aleatórios.
Exemplos:

```
rpois(10,5) # 10 números aleatórios da dist. poisson com parâmetro 5
 [1] 3 2 3 1 8 6 1 3 2 3
```

```
rnorm(5)    # 5 números aleatórios da normal padrão
[1]  0.04659011 -0.05019082 -1.28753167  1.15411245 -1.37573809
qt(0.95,50) # valor quantilico da t, com 50 gl
[1] 1.675905
qchisq(0.95,1) # valor da qui-quadrado
# com 95% de confiança e 1 gl
[1] 3.841459
dpois(4,5) # densidade da dist poisson, x=4, parâmetro=5.
[1] 0.1754674
ppois(4,5) # P(x<=4), parâmetro=5.
[1] 0.4404933
```

1.26 Algoritmos de programação no R

R é uma linguagem de programação orientada a objetos, que além de analisar dados, têm sua própria programação, aqui usaremos esta programação para analises de estatísticas e econometria. Para maiores detalhes podemos pesquisar em Venables and Ripley (2000 e 2002). Você pode estender a funcionalidade do R, criando novas funções ou programando por exemplo:

```
mediana.media.razao <- function(x)
{
return(median(x)/mean(x))
}
set.seed(20)
z = rexp(10000)
mediana.media.razao(z)
[1] 0.6925193
```

Construindo a mediana

```
mediana=function(x) {
odd.even=length(x)%%2
if (odd.even == 0) (sort(x)[length(x)/2]+sort(x)[1+ length(x)/2])/2
else sort(x)[ceiling(length(x)/2)]
}
```

Construindo a média geométrica

```
mediageometrica = function (x) exp(mean(log(x)))
```

Construindo a média harmonica

```
mediaharmonica=function (x) 1/mean(1/x)
```

Um algoritmo que envolva as medidas de tendência central

```
medidascentrais = function(y, medida) {
switch(medida,
média = mean(y),
mgeometrica = exp(mean(log(y))),
mharmonica = 1/mean(1/y),
mediana = median(y),
stop("Medida não inclusa"))
}
medidascentrais(y,"média")
```

Gerando uma distribuição uniforme para o lançamento de um dado

```
ndado=function(x){ return(round(runif(x)*6+0.5))}
hist(ndado(100000)
```

Com o comando **sample**, podemos realizar este mesmo experimento da seguinte maneira:

```
gerando=sample(1:6, 100000, replace=TRUE)
hist(gerando)
```

Considere o seguinte objeto c

```
c=c(1,2.2,7,15,3,6.5)
```

Para determinar que classe de objeto é o vetor c, podemos identificar usando os seguintes comandos:

```
is.character(c)
[1] FALSE
is.numeric(c)
[1] TRUE
```

A este objeto c, podemos a signar nome a cada valor definido da seguinte forma:

```
names(c)=c("a","b","c","d","e","f")
c
   a    b    c    d    e    f
 1.0  2.2  7.0 15.0  3.0  6.5
```

Podemos associar uma comparação logica com o vetor c, da seguinte forma:

```
c>3
    a     b     c     d     e     f
FALSE FALSE  TRUE  TRUE FALSE  TRUE
```

Quando estamos diante de uma matriz, podemos definir os nomes das colunas com **colnames()**, e os nomes das linhas com **rownames()**, veja o exemplo:

```
d=matrix(c(2,2,3,5,8,2,8,5,8),3,3)
colnames(d)=c("c1","c2","c3")
rownames(d)=c("1°","2°","3°")
d
   c1 c2 c3
1°  2  5  8
2°  2  8  5
3°  3  2  8
```

A matriz d é entendida como um conjunto de dados que podem ser associados as linhas e as colunas. Para determinar se há associação entre linhas e colunas o teste utilizado pode se o qui-quadrado com a função **chisq.test()**

1.27 Laços no R

Em algumas situações variáveis contínuas podem ser transformadas em variáveis ordinais ou dummy, para realizar estas operações o R proporciona vários laços como else if ou ifelse:
Exemplo, para construir variáveis binárias, considere o consumo de água, onde consumo menor ou igual a 10 m^3 é zero e acima de 10 m^3 é 1. O valor de consumo sendo zero se o valor é menor ou igual a 20 reais, e 1 caso contrario.

```
consumoc=numeric(length(consumo))
for(i in 1:length(consumo)){if (consumo[i]<=10) consumoc[i]=0
else consumoc[i]=1}

valorc=numeric(length(valor))
for(i in 1:length(valor)){if (valor[i]<=20) valorc[i]=0 else valorc[i]=1}
```

Nota: Podemos construir os laços considerando a condição como texto, exemplo: consumoc[i]=”[0 − 10]”. Comando alternativo de construção:

```
consumoc2=ifelse(consumo<=10,"[0-10]","Acima de 10")
valorc2=ifelse(valor<=20,"[0-20]","Acima de 20")
```

limitação deste laço é que utilizado apenas para duas condições. Para laços com mais de duas condições podemos usar os seguintes comandos:

```
valorc3=numeric(length(valor))
for(i in 1:length(valor)){if (valor[i]<20) valorc3[i]="[0-20)"
else if (valor[i]<30)
valorc3[i]="[20-30)" else valorc3[i]="[30 a +"}
table(valorc3)
```

1.28 A função cut()

Alguma funções usam laços para criar novas variáveis categóricas, no exemplo de valor pago em reais pelo consumo de água, podemos construir categorias por quartis de $.25, .50, .75, 1$, da seguinte maneira:

```
q=cut(valor,quantile(valor), include.lowest=T)
table(q)
q
[13.5,16.5] (16.5,22.2] (22.2,32.7] (32.7,69.3]
         17          16          16          16
```

Outras formas podem ser usando intervalos a critério pessoal, no exemplo a seguir, consideramos com valor mínimo 13 e valor máximo 70, com amplitude do intervalo de 19, e com classes fechadas no mínimo e aberto no máximo.

```
valr=cut(valm,seq(13,70,19),right=F,include.upper=T)
table(valr)
valr
[13,32) [32,51) [51,70)
     48      15       2
```

1.29 Criando funções. Teste que compara duas médias

Como exemplo implementamos a estatística de comparação de duas amostras independentes.

```
testeduas <- function(y1, y2) {
+ n1 <- length(y1); n2 <- length(y2)
+ yb1 <- mean(y1); yb2 <- mean(y2)
+ s1 <- var(y1); s2 <- var(y2)
+ s <- ((n1-1)*s1 + (n2-1)*s2)/(n1+n2-2)
+ tst <- (yb1 - yb2)/sqrt(s*(1/n1 + 1/n2))
+ tst
+ }
x=runif(20)
y=rnorm(20)
testeduas(x,y)
[1] 2.874636
```

Uma forma alternativa é usando a função **t.test()**

```
t.test(x,y)

        Welch Two Sample t-test

data:  x and y
t = 2.8746, df = 21.784, p-value = 0.008858
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1846283 1.1429380
sample estimates:
 mean of x  mean of y
 0.5539270 -0.1098562
```

1.30 Coeficiente de regressão

Implementando o calculo da tangente (inclinação) do coeficiente no modelo de regressão linear simples:

```
mq1<- function(X, y) {
+ X <- qr(X)
+ qr.coef(X, y)
+ }
mq1(x,y)
[1] -0.08755885
```

1.31 Teste Qui- Quadrado

Útil para associar linhas e colunas, considere os dados em d, com as seguintes hipóteses:
H0: não há associação entre linhas e colunas.
Ha: há associação entre linhas e colunas.

```
chisq.test(d)

        Pearson's Chi-squared test

data:  d
X-squared = 4.6497, df = 4, p-value = 0.3252

Warning message:
In chisq.test(d) : Aproximação Qui-quadrado pode estar incorreta
```

Para um nível $\alpha = 0,05$ de significância, podemos afirmar que não há evidencias para afirmar que a associação entre linhas e colunas (linhas e colunas são independentes). Um outro exemplo do uso da distribuição Qui-quadrado é verificar se um conjunto de dados provem de uma determinada distribuição teórica:

```
chisq.test(c)

        Chi-squared test for given probabilities

data:  c
X-squared = 22.549, df = 5, p-value = 0.0004116
```

Pelo p-value, podemos afirmar que a $0,05$ de significância não há evidencias para afirmar que a distribuição dos números em c sejam uniformes. Para implementar este teste de associação no exemplo de consumo de água, considerando a classificação mediante laços (ver laços no R) entre consumoc e valorc. temos:

```
table(consumoc,valorc)
        valorc
consumoc  0  1
       0 24  0
```

```
       1  3 38
chisq.test(consumoc,valorc)$expected
chisq.test(consumoc,valorc)$observed
chisq.test(consumoc,valorc)

        Pearson's Chi-squared test with Yates' continuity correction

data:  consumoc and valorc
X-squared = 49.8015, df = 1, p-value = 1.701e-12
barplot(table(consumoc,valorc), beside=T)
```

Concluimos que há associação entre consumo de agua e o valor pago em reais.

1.32 Teste t - student

O teste mais usado pelos estatísticos, é útil para comparar médias de duas amostras tanto independente como emparelhadas.
Exemplo, considere as exportações mensais (FOB-US) do Brasil durante os anos de 2009 − 2010. Após digitação das observações, temos:

```
exp2009
 [1]  9781.920  9586.406 11809.225 12321.617 11984.585 14467.785 14141.930
 [8] 13840.850 13863.222 14081.686 12652.892 14462.624
exp2010
 [1] 11305.07 12197.24 15727.50 15161.21 17702.50 17093.91 17672.92 19236.25
 [9] 18832.79 18380.42 17687.33 20918.14
```

H0: A médias anuais das exportações durante 2009 e 2010 é a mesma.
Há: C.C.

```
t.test(exp2009,exp2010,paired = FALSE)

        Welch Two Sample t-test

data:  exp2009 and exp2010
t = -4.2813, df = 18.202, p-value = 0.0004394
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -6075.666 -2077.757
sample estimates:
mean of x mean of y
```

```
 12749.56  16826.27
```

Conclusão: não há evidencias para aceitar que a média das exportações de 2009 e 2010 é a mesma.

Quando um conjunto de dados é dado em formato de matriz (data.frame), podemos realizar o teste t de uma coluna em função da outra, por exemplo no consumo de água, queremos testar a hipótese nula H0: o valor de consumo de água é o mesmo construindo ou não versus a hipótese alternativa Ha: Caso contrario.

```
attach(ca)
t.test(valor~Construindo)

        Welch Two Sample t-test

data:  valor by Construindo
t = -5.8383, df = 21.731, p-value = 7.465e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -22.88867 -10.88356
sample estimates:
mean in group 0 mean in group 1
       21.15271        38.03882
```

Concluímos que não há evidencias para aceitar que o valor de consumo de água é o mesmo quando estamos construindo ($p - valor < 0.05$)

Exemplos adicionais podem ser comparando entre as variáveis classificadas (ver seção de laços) da seguinte forma:

```
t.test(valor~consumoc)
t.test(consumo~valorc)
boxplot(consumo~valorc2, main="Consumo e valor pago em Reais")
boxplot(consumo~valorc3, main="Consumo e valor pago em Reais")
```

1.33 Teste F

Para determinar se há a mesma variabilidade no valor de consumo, quando estamos ou não construindo, utilizamos o teste F. A hipótese nula é, H0: O valor de consumo de água tem a mesma variabilidade quando construímos que quando não construímos. No R, temos:

```
var.test(valor~Construindo)
        F test to compare two variances
data:  valor by Construindo
F = 0.4834, num df = 47, denom df = 16, p-value = 0.05478
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.194815 1.014916
sample estimates:
ratio of variances
         0.4834048
```

Concluímos que não há evidencias para rejeitar a hipótese nula, por tanto a variabilidade no valor de consumo é o mesmo.
O correspondente gráfico (Box-plot) é apresentado a seguir:

```
boxplot(valor~Construindo, col=8, names=c("não construção", "construção"))
```

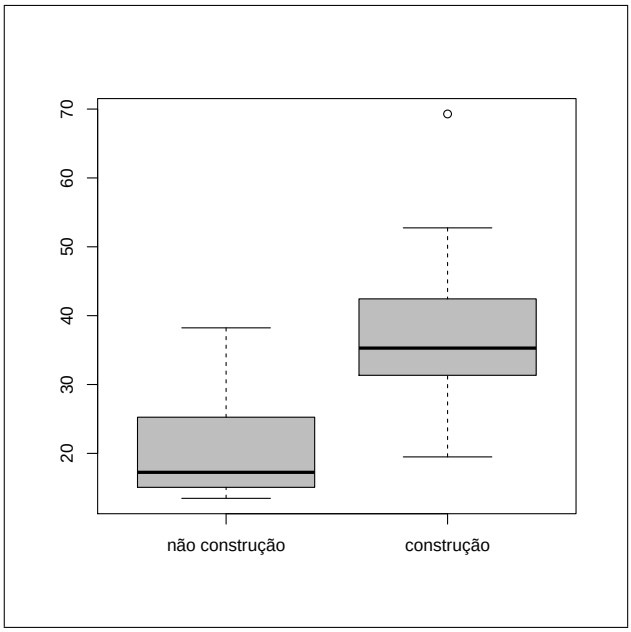

Figura 4. Valor pago em reais no consumo de água

Em alguns casos há necessidade de identificar se a variabilidade das exportações é a mesma comparando 2009 e 2010.
H0: As exportações de 2009 e 2010 tem a mesma variabilidade (dados mensais coletados do site http://www.ipeadata.gov).
Há: C.C.

```
var.test(exp2009,exp2010)
        F test to compare two variances
data:  exp2009 and exp2010
F = 0.3728, num df = 11, denom df = 11, p-value = 0.1166
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1073346 1.2951624
sample estimates:
ratio of variances
         0.3728482
```

Conclusão: Não houve mudanças na variabilidade das exportações de 2009 e 2010.

1.34 Gráficos

Considere os dados mensais das exportações do Brasil (FOB) durante 2009 e 2010, para realizar um gráfico de box-plot, fazemos:

```
boxplot(exp2009,exp2010, main="exportações do Brasil",
names=c(2009,2010))
```

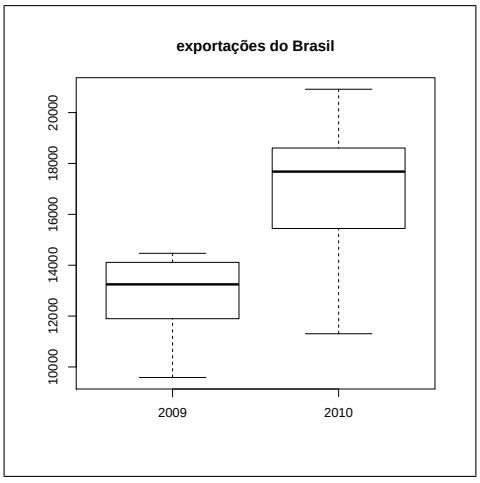

Figura 5. Exportações FOB em dólares.

Para o gráfico da função acumulativa das exportações:

```
plot(ecdf(exp2010), do.point=FALSE, verticals=TRUE,
lty=3, xlim=c(10000,23000))
lines(ecdf(exp2009), do.point=FALSE, verticals=TRUE,
col=2)
```

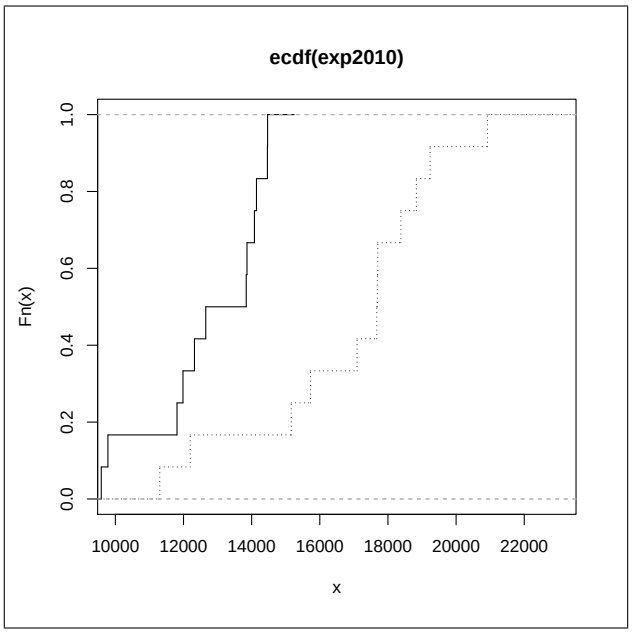

Figura 6. Função acumulada das exportações 2009-2010

Considere as exportações do Brasil, um histograma e Q-Q plot será:

```
hist(exp, main="Exportações FOB-US", col=8)
qqnorm(exp, xlim=c(-5,5),ylim=c(-1000,21000))
qqline(exp)
```

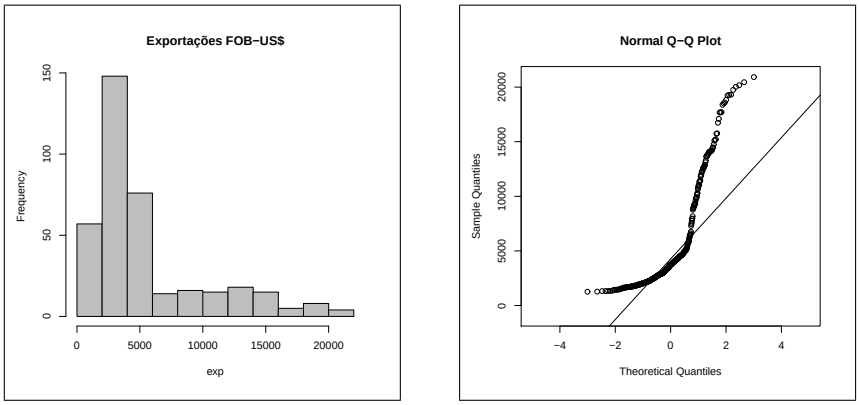

Figura 7. Gráficos das Exportações no Brasil. 2001-2010

Para ativar alguns gráficos, pacotes são necessários, exemplo:

```
library(MASS)
```

```
par(mfrow=c(2,1))
z = rnorm(500000)
hist(z, prob=T)
curve(dnorm(x),add=T)
hist.scott(z, prob=T)
```

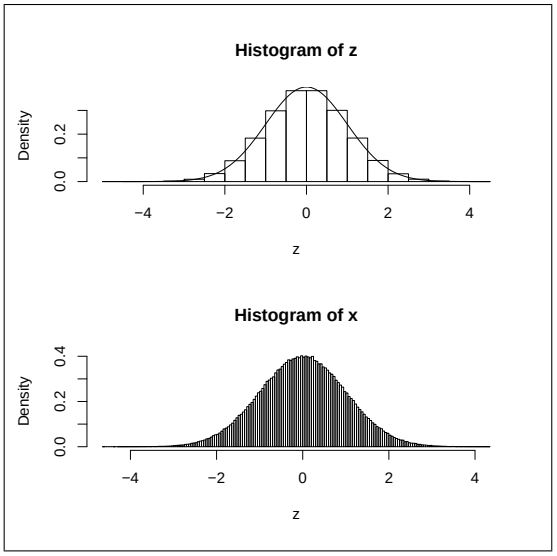

Figura 8. Histograma de números aleatórios normais

```
boxcox(dist~speed, data=cars) #transformação Box-Cox
```

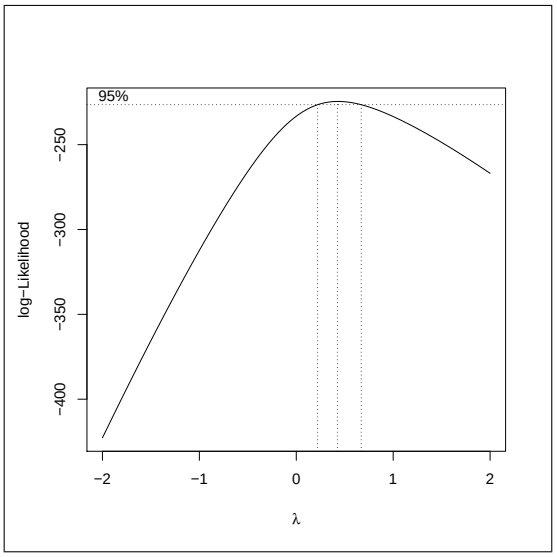

Figura 9. Valor ótimo na transformação Box-Cox

Para ter um alcance maior sobre a capacidade gráfica do R, podemos usar as seguintes funções:

```
demo(graphics)
demo(image)
demo(persp)
demo(recursion)
demo(plotmath)
demo(package = .packages(all.available = TRUE))
```

1.35 Tabelas

Algumas tabelas estatísticas podem ser construídas com a função *table()*.

```
estado=c("PB","PE","AL","SE","BA","MG","ES","RJ","PB","PE","AL",
"SE","BA","MG","ES","RJ","PB","PE","AL","SE","BA","MG","ES","RJ",
"PB","PE","AL","SE","BA","MG","ES","RJ")

estadof=factor(estado)
estadof
[1] PB PE AL SE BA MG ES RJ PB PE AL SE BA MG ES RJ PB PE AL SE BA
[22] MG ES RJ PB PE AL SE BA MG ES RJ
Levels: AL BA ES MG PB PE RJ SE
set.seed(10)
salarios=c(round(runif(32)*100))
salarios
[1] 51 31 43 69  9 23 27 27 62 43 65 57 11 60 36 43  5 26 40 84 86 62 78 36 41
[26] 71 84 24 77 36 54  9
salariom=tapply(salarios,estadof,mean)
salariom
   AL    BA    ES    MG    PB    PE    RJ    SE
58.00 45.75 48.75 45.25 39.75 42.75 28.75 58.50
salariof=factor(cut(salarios,breaks=0+15*(0:7)))
salariof
 [1] (45,60] (30,45] (30,45] (60,75] (0,15]  (15,30] (15,30] (15,30] (60,75]
[10] (30,45] (60,75] (45,60] (0,15]  (45,60] (30,45] (30,45] (0,15]  (15,30]
[19] (30,45] (75,90] (75,90] (60,75] (75,90] (30,45] (30,45] (60,75] (75,90]
[28] (15,30] (75,90] (30,45] (45,60] (0,15]
Levels: (0,15] (15,30] (30,45] (45,60] (60,75] (75,90]
table(salariof,estadof)
         estadof
```

```
salariof  AL BA ES MG PB PE RJ SE
  (0,15]   0  2  0  0  1  0  1  0
  (15,30]  0  0  1  1  0  1  1  1
  (30,45]  2  0  1  1  1  2  2  0
  (45,60]  0  0  1  1  1  0  0  1
  (60,75]  1  0  0  1  1  1  0  1
  (75,90]  1  2  1  0  0  0  0  1

table(salariof,estadof)
          estadof
salariof   AL BA ES MG PB PE RJ SE
   (0,15]  1  1   0  0  1  0  1  2
  (15,30]  0  0   1  1  0  0  1  1
  (30,45]  0  0   2  0  0  1  0  0
  (45,60]  2  0   1  0  0  1  0  0
  (60,75]  0  1   0  2  3  0  1  0
  (75,90]  0  1   0  1  0  2  0  1
 (90,105]  0  1   0  0  0  0  1  0
```

1.36 Comandos adicionais

Em situações onde a ausência de informação ou dados faltantes(missing), as funções do R precisam declarar esta ausência com o comando na.rm=T, indicando que a informação apresenta dados faltantes. Exemplo

```
df=c(2,NA,5,8,NA) # O último elemento esta ausente
mean(f)        #Comando padrão para calcular a média
[1] NA
mean(f,na.rm=T) #Comando declarando a ausência de elementos
[1] 5
```

2 R e Econometria

Dados econômicos, precisam de uma atenção matemática para dar fundamento a suas teorias. Atualmente pelo avanço computacional não há barreiras para serem aplicadas, e torna-se até divertida se definimos a plataforma de trabalho computacional. A econometria requer de uma abordagem diferenciada, pois faz uso de da teoria econômica, economia matemática, estatística econômica, estatística matemática e inferência estatística. Esta

disciplina aproxima a Estatística da Economia, onde a teoria econômica formula as hipóteses. Por exemplo: uma redução de preços de uma mercadoria devera aumentar a quantidade demandada dessa mercadoria, considerando uma relação inversa entre preço e quantidade demandada, ou a despesa de consumo per capita depende da renda per capita, considerando uma relação direta, ou seja: a medida que a renda aumenta a despesa tende a aumentar. Outro exemplo onde consideremos a área microeconômica: consumo de água de uma residência e o valor pago, é natural imaginar que há uma relação direta entre consumo e valor pago pelo consumo. Porem precisamos saber qual e quanto é esta força de relação, por tanto o econometrista fornece esta informação usando a inferência estatística e a estatística econômica. As estimativas de esta relação fica por conta de estatística matemática. Para completar e cerrar este circulo de disciplinas abordadas na Econometria precisamos utilizar uma ferramenta de cálculo computacional, que brinde estabilidade, precisão e rapidez nos seus resultados, para superar tudo isto utilizaremos o ambiente R. A econometria surge como uma disciplina autônoma em virtude de aglutinar todas estas áreas do conhecimento merecendo um estudo próprio.

2.1 Tipos de Dados

Os tipos de dados são:

1. Dados de corte transversal (cross-section), são dados de uma ou mais variáveis coletadas no mesmo ponto do tempo. Por exemplo, os censos no Brasil são coletados num instante do tempo. No censo, variáveis demográficas são exploradas como: contagem da população, número de indivíduos em cada família, idade dos integrantes da família ou variáveis econômicas, como: renda, consumo, tipo de moradia (alugada ou própria) etc., estes dados são adquiridas a cada década pelo IBGE.

2. Dados longitudinais ou de séries temporais, são caracterizadas por observações ao longo do tempo. No Brasil um site que disponibiliza dados desta natureza é o IPEADATA.

3. Dados de painel, é uma mistura de dados de corte transversal e longitudinais, ou ao longo do tempo estudamos o comportamento de unidades individuais. Por exemplo, séries mensal do índice de preços, série anuais do produto interno bruto (PIB), ou série anual da balança comercial dos municípios do Brasil, considerando cada município como unidades individuais.

2.2 Metodologia

Os passos da metodologia econométrica tradicional ou clássica se inicia desde a formulação da hipótese até as conclusões, em geral são:

2.2.1 Hipótese

Como referência usaremos a hipótese nula $H0$, a qual nulifica qualquer diferença, e a hipóteses alternativa Ha, aquela que contradiz, total ou parcialmente a hipótese nula. Exemplo:

1. $H0$: não existe uma relação entre consumo de água e valor pago pela mesma.
2. Ha: existe uma relação entre consumo de água e valor pago pela mesma.

A hipótese alternativa pode ser dividida em três possibilidades, no exemplo do consumo de água pode ser dividida em: existe alguma relação, ou a relação é negativa ou a relação é positiva. A primeira denomina-se de hipótese alternativa com teste bicaudal, o segundo de teste unicaudal á esquerda, e finalmente o último com teste unicaudal a direita. Nosso exemplo considere Ha: existe alguma relação (teste bicaudal)

2.2.2 Modelo matemático

Usaremos uma função de consumo simples

$$\hat{Y} = \beta_1 + \beta_2 X \tag{1.1}$$

A variável dependente Y, valor pago pelo consumo é estimado por $\hat{Y}$. A variável independente é X, consumo de água em m^3. β_1, representa o valor pago quando não há consumo de água. Nosso interesse é determinar a tangente ou a variação pelo consumo β_2.

2.2.3 Modelo econométrico do valor pago

O modelo matemático fornece a relação determinística entre valor pago em reais e consumo de água, porem as relações entre as variáveis econômicas são em geral inexatas. Assim ao longo do tempo no podemos esperar que o valor pago e o consumo em m^3 se situem exatamente numa linha reta, isto

por que além do consumo, outras variáveis afetam o valor pago, por exemplo os dias de consumo, o período de reforma da casa, ou a inflação, podendo alterar o valor pago pelo consumo.

2.2.4 Nível de significância

Denominado de α, frequentemente usa-se $\alpha = 0.05$; em situações de risco, usa-se $\alpha = 0.01$. Em testes bicaudais divide-se α em duas partes, e localizam-se nos extremos da distribuição de prova. Para testes unicaudais localiza-se no extremo da distribuição indicada pela hipótese alternativa. Nosso exemplo $\alpha = 0.05$ e por tanto $\alpha/2 = 0.025$.

2.2.5 Obtenção de Dados

Considere 65 observações (mês a mês) do consumo de água em uma residencia e o valor pago em reais, durante o período de 12/2005 a 04/2011. Ver dados no apêndice A. Um plot é apresentado a seguir:

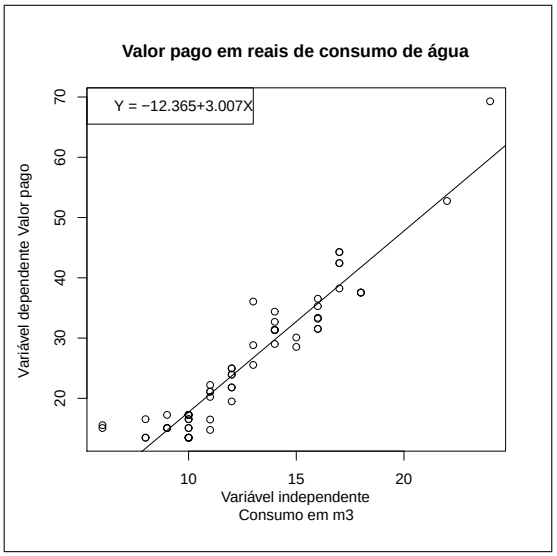

Figura 10. Relação causal entre consumo e valor pago em reais

Podemos observar visualmente que há uma relação positiva entre valor pago e consumo de água, onde a variação pelo consumo ou variação do valor pago (tangente) β_2, cresce positivamente. Precisamos verificar se esta variação β_2 é estatisticamente diferente de zero (significativa).

2.2.6 Estatística de prova

Denominada de função pivô ou estatística de teste. No âmbito paramétrico usa-se as estatísticas: t student (amostras pequenas), quando n, é grande é indiferente o uso da t student ou normal. No exemplo para testar qual a inclinação da variação pelo consumo de água, usamos a seguinte estatística:

$$t_0 = \frac{\hat{\beta}_2}{dp(\hat{\beta}_2)} \approx t_{n-2gl} \tag{1.2}$$

onde: t_0 é o valor calculado da estatística t, $\hat{\beta}_2$ é a estimativa do parâmetro (variação pelo consumo) β_2, $dp(\hat{\beta}_2)$ é o desvio padrão da estimativa do parâmetro β_2, o valor t_0 segue uma distribuição t-student com $n-2$ graus de liberdade(gl). Os graus de liberdade é calculado diferenciando o tamanho da amostra com o número de parâmetro do modelo (n-p). O valor quantílico $t_{n-2gl}(0.025)$, corresponde a distribuição t-student com 0.025 de significância com $n-2$ graus de liberdade, e n é o número total de observações. Nosso exemplo $\beta_2 = 3.007$, e $dp(\beta_2) = 0.13$, por tanto $t_0 = 23.13$ é comparado com o valor da tabela da t-student com 63 gl é com $\alpha/2 = 0.025$, sendo este valor igual 1.9983.

2.2.7 Tipos de erro

Ocorre o erro tipo I (α) ao se rejeitar a hipótese de nulidade quando deveria aceitá-la. Por outro lado quando se aceita a hipótese de nulidade quando deveria rejeitá-la comete-se o erro tipo II (β). Nas duas situações tomou-se uma decisão errada. O interesse é a minimização dos erros tipo I e II que na prática só é possível aumentando-se o tamanho da amostra. Por razões diversas o aumento do tamanho da amostra muitas vezes torna-se impraticável. A gravidade do erro tipo I é distinta da do erro tipo II, quando se consegue identificar o erro mais grave a opção é a minimização deste, porem quando a probabilidade de ocorrência de um tipo de erro diminui a do outro aumenta e vice versa. Podemos observar este critério na seguinte tabela:

	Não rejeita	rejeita
H0 verdadeira	Correto $(1-\alpha)$	Erro do tipo I (α)
H0 falsa	Erro do tipo II (β)	Correto $(1-\beta)$

2.2.8 Regra de Decisão

Os pacotes ou plataformas estatísticas apresentam o valor observado da função pivô e o $p - valor$ (Área de probabilidade está relacionada como nível de significância), assim a função pivô apresenta o valor numérico na função de densidade da Estatística, exemplo dentro da densidade da normal. Quando o *pvalor* é menor do que $\alpha = 0.05$ no caso do teste unicaudal (e $\alpha = 0.025$ no caso bicaudal), concluímos que não há evidencia suficiente para aceitar H0. Caso contrario se o $p - valor$ for maior que $\alpha = 0.05$, podemos concluir que não há evidencia suficiente para rejeitar H0. Nosso exemplo: $t_0 = 23.13 > 1.998$, (1.998 é valor da distribuição t com 63 gl e nível de significância bicaudal de 0.05), este valor localiza-se na cauda positiva da distribuição com 63 graus de liberdade($n - 2 = 65 - 2$). Por outro lado o $p - valor < 0.025$, por tanto não há evidencias para aceitar H0, em favor de aceitar que o coeficiente de inclinação (ou tangente) é positiva. No recorrer do livro usaremos códigos para as medidas do $p - valor$, que em geral aparece em todo comando **summary()**. Como consequência a linha que aparece a seguir será excluída das saídas do R:

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1  1
```

será interpretada da seguinte maneira:

- 0 *** significância menor que 0.001
- 0.001 ** significância menor que 0.01
- 0.01 * significância menor que 0.05
- 0.05 . significância menor que 0.1
- 0.1 significância menor que 1

2.2.9 Conclusões

Esta parte do teste de hipóteses é explicada em conjunto: O Estatístico e o Economista, que são os profissionais da área em que as observações foram coletadas, decidem sobre as providencias futuras. No exemplo podemos afirmar que a cada metro cubico a mais de consumo de água será pago em média 3 reais a mais. No momento este valor pago pode parecer alto, porem na falta de este liquido elementar, este valor pode aumentar e por tanto sofrerá mudanças.

2.3 Previsões

Muitas vezes estamos interessados em determinar o que acontece quando uma quantidade de consumo de água em m^3 é utilizada e qual o valor pago em reais. Por exemplo se nosso interesse é saber quanto sera pago por $20m^3$ de consumo de água em uma residência. A conta a realizar é:
Valor pago $= -12.365 + 3.007 * 20 = 47.775$ reais
A interpretação deste valor é: em média o valor pago em reais por $20m^3$ de água consumida é de aproximadamente 47.8 reais

2.3.1 Resíduos

Uma questão fundamental em econometria é identificar o comportamento dos resíduos de um modelo. Considere Y, o valor real pago e $\hat{Y}$ valor pago estimado a partir de um modelo, então este erro é aleatório e definido por:

$$\epsilon_i = Y_i - \hat{Y}_i \tag{1.3}$$

A abordagem mais formal dos erros é feita nos próximos capítulos. Uma questão importante é determinar qual o comportamento da estimativa destes erros aos quais chamamos de resíduos. Para determinar a normalidade dos resíduos realizamos o teste de **Jarque Bera**, com a função **jarque.bera.test()** da biblioteca **tseries**, com o seguinte gráfico:

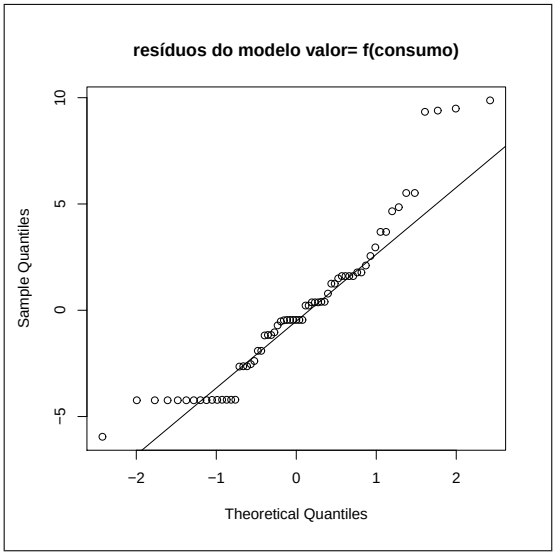

Figura 11. Normalidade dos resíduos

2.4 Exercícios

1. Por que a econometria precisa ser uma disciplina autônoma?
2. Que é econometria?
3. Pesquise dados de corte transversal, corte longitudinal e de painel.
4. Construa uma metodologia econométrica para o consumo familiar em função da renda familiar (linear).
5. No item anterior é possível encontrar uma relação quadrática?
6. Quais são os pressupostos básicos de um modelo de regressão linear simples?
7. Como é chamado o parâmetro de inclinação ou tangente num modelo de regressão linear simples em econometria?
8. Que é o projeto R?
9. R é um programa econométrico?

Referências

[1] Albert, Jim (2009). *Bayesian computation with R.* Editora Springer. Second Edition. New York, USA.

[2] Borde, Arvind (1992). TEX *by example.* A beginner´s guide. Ed. Academic Press Professional. United States of America.

[3] Carneiro, Orlando (1997). *Econometria Básica. Teoria e Aplicações.* Editora Atlas. Segunda Edição. son Paulo.

[4] Conover, W. J. (1999) *Practical nonparametric statistics.* 3.ed. New York.

[5] Cribari Neto, F (2002). *C for Econometricians.* conputational Economics 14. p.135-149.

[6] Dalgaard, Peter(2008). *Introductory Statistics with R.* Editora Springer. Second Edition. New York, USA.

[7] Ehlers, Ricardo.(2007): Análise de séries Temporais. Universidade Federal do Paraná.

[8] Frery, Alejandro; Cribari-Neto, Francisco.(2011).*Elementos de Estatística conputacional Usando Plataformas de Software Livre/Gratuito*. Publicações Matemáticas. Editora IMPA.

[9] Gujarati, Damodar N (2000). *Econometria Básica*. Makron Books. Terceira Edição. son Paulo.

[10] Knuth, D.E. (1981).*The Art of conputer Programming, Third Edition. Volume 2: Seminumerical. Algorithms*, Addison-Wesley Publishing conpany, Massachusetts.

[11] Korgi, Rodrigo de Castro (2003). *El universo LaTeX*. Editora Facultad de Ciencias. Segunda Edição. Bogota.

[12] Krawczyk, H. *How to Predict Congruential Generators*. In:Journal of Algorithms, V. 13, N. 4, 1992.

[13] L'Ecuyer, P (2001). *Software for uniform random number generation: distinguishing the good and the bad*. In Proceedings of the 2001 Winter Simulation Conference.

[14] Maindonald, John; Braun, W John(2010). *Data analysis and graphics using R : an example-based approach*. Cambridge University Press, New York, USA.

[15] Mingoti(2005).*Analise De Dados Atraves De Metodos De Estatistica Multivariada - Uma Abordagem Aplicada*. Editora UFMG. Belo Horizonte.