Aula 13_MCA - Probabilidade e Distribuições. Amostragem Aleatória. Cálculo de Probabilidade e Combinatória. Distribuições Discretas e Contínuas.Estatística Descritiva e Gráficos. Resumo Estatístico para um Único Grupo. Representação Gráfica de Distribuições. Histogramas. Q-Q plot. Boxplot. Resumo Estatístico por Grupos. Gráfico para Dados Agrupados.Testes para uma ou duas Amostras. Teste t para uma Amostra. Teste t para duas Amostras.

Prof. Weligton Gomes

2023-06-05

Probabilidade e Distribuições

Amostragem Aleatória

```
sample(1:40, 5)
```

```
## [1] 30  4 24 11  2
```

```
sample(c ("H", "T"), 10, replace = T) #H = Heads e T = Tails
```

```
##  [1] "T" "T" "T" "T" "T" "H" "H" "H" "H" "T"
```

```
sample(c("succ", "fail"), 10, replace=T, prob=c(0.9, 0.1))
```

```
##  [1] "succ" "succ" "succ" "succ" "succ" "succ" "succ" "succ" "succ" "succ"
```

Cálculos de probabilidade e combinatória

```
1/prod(40:36)
```

```
## [1] 1.266449e-08
```

```
prod(5:1)/prod(40:36)
```

```
## [1] 1.519738e-06
```

```
1/choose(40,5)
```

```
## [1] 1.519738e-06
```

```
# Mega-Sena
1/choose(60,6)
```

```
## [1] 1.997449e-08
```

Distribuição estatística:

Densidade ou probabilidade de ponto

dnorm - A função dnorm retorna o valor da função de densidade de probabilidade para os parâmetros dados de distribuição normal para x, μ e σ, $P(X = x)$;

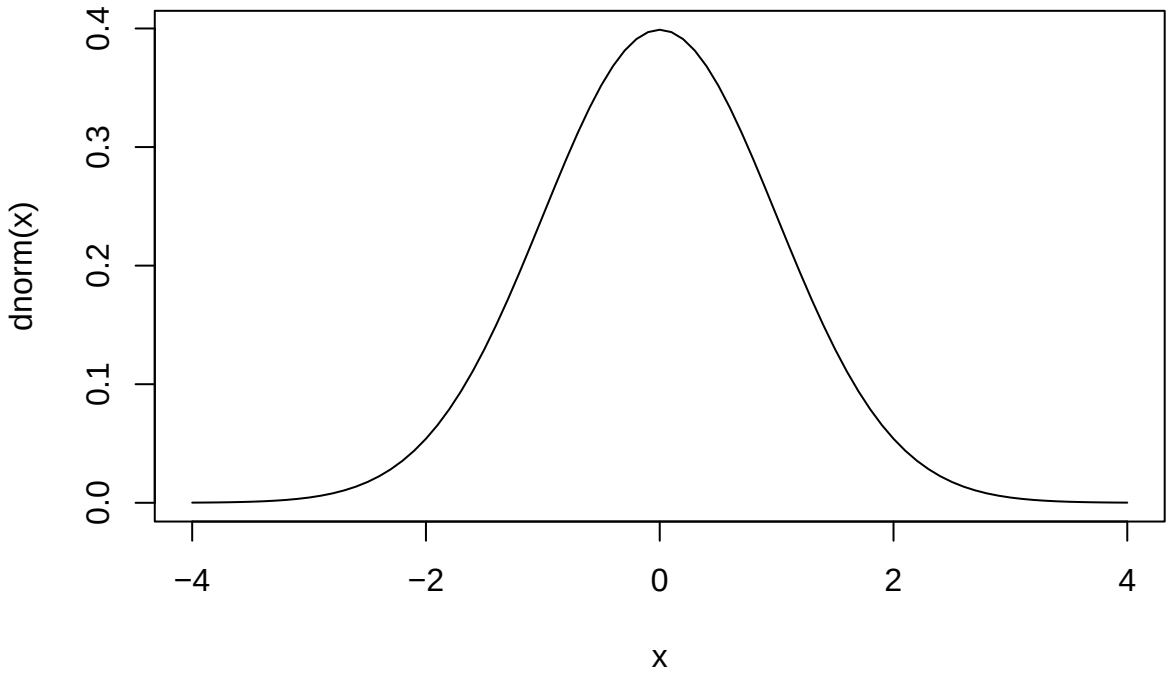
pnorm dá a função de distribuição acumulada (cdf), ou seja, a função pnorm retorna a integral de $-\infty$ para q da fdp da distribuição normal, $P(X \leq x)$;

qnorm - a função qnorm é simplesmente o inverso da cdf, que você também pode pensar como o inverso de pnorm! Qual é o z-score do 50º quantil da distribuição normal? $qnorm(0.5) = 0$,

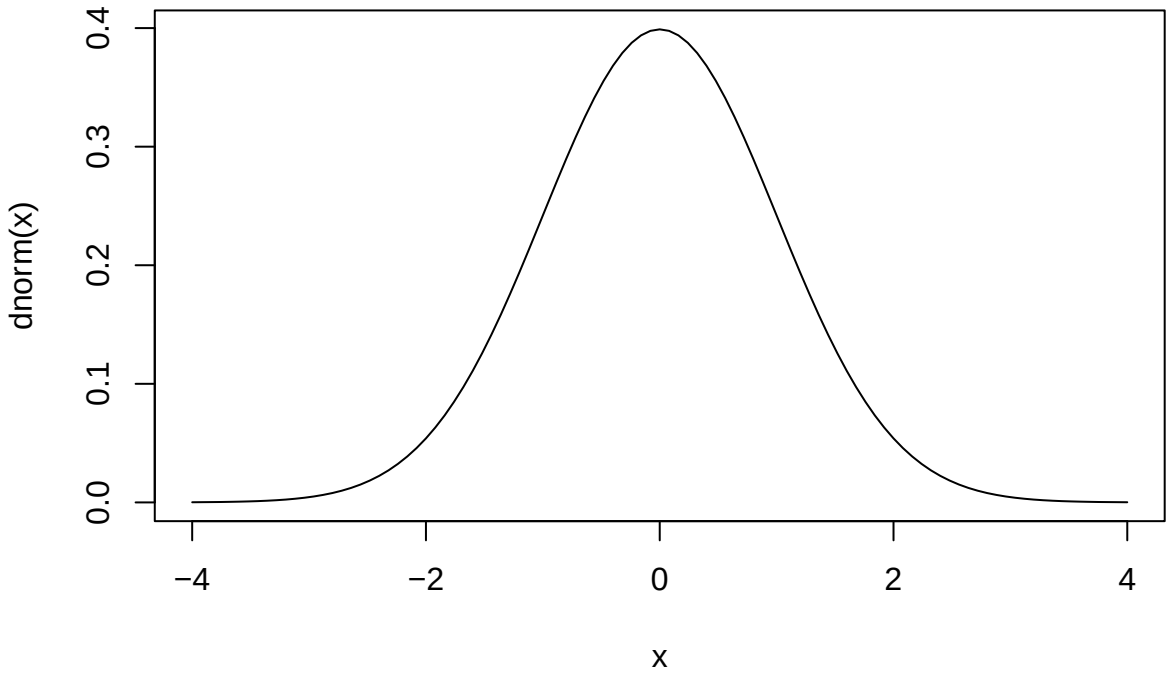
Nas estatísticas, um escore-z (ou escore padrão) de uma observação é o número de desvios padrão acima ou abaixo da média da população.

A probabilidade de uma variável aleatória contínua é definida pela área abaixo de uma função positiva,denominada densidade de probabilidade. A densidade em si, não é uma probabilidade, mas uma função matemática que auxilia na atribuição de probabilidades

```
x <- seq(-4,4,0.1)
plot(x,dnorm(x),type="l") # onde "l" -> linha,
```




```
curve(dnorm(x), from=-4, to=4)
```



Probabilidade acumulada, função de distribuição

A função de distribuição cumulativa descreve a probabilidade de "acertar" x ou menos em uma determinada distribuição. As funções no R correspondentes começam com um 'p' (para probabilidade) por convenção. Digamos que é sabido que alguma medida bioquímica em indivíduos saudáveis é bem descrita por uma distribuição normal com uma média de 132 e um desvio padrão de 13. Então, se um paciente tem um valor de 160, há:

```
pnorm(160, mean=132,sd=13)
```

```
## [1] 0.9843739
```

```
1-pnorm(160, mean=132,sd=13)
```

```
## [1] 0.01562612
```

Cerca de 1,5% da população geral, que tem esse valor ou superior.

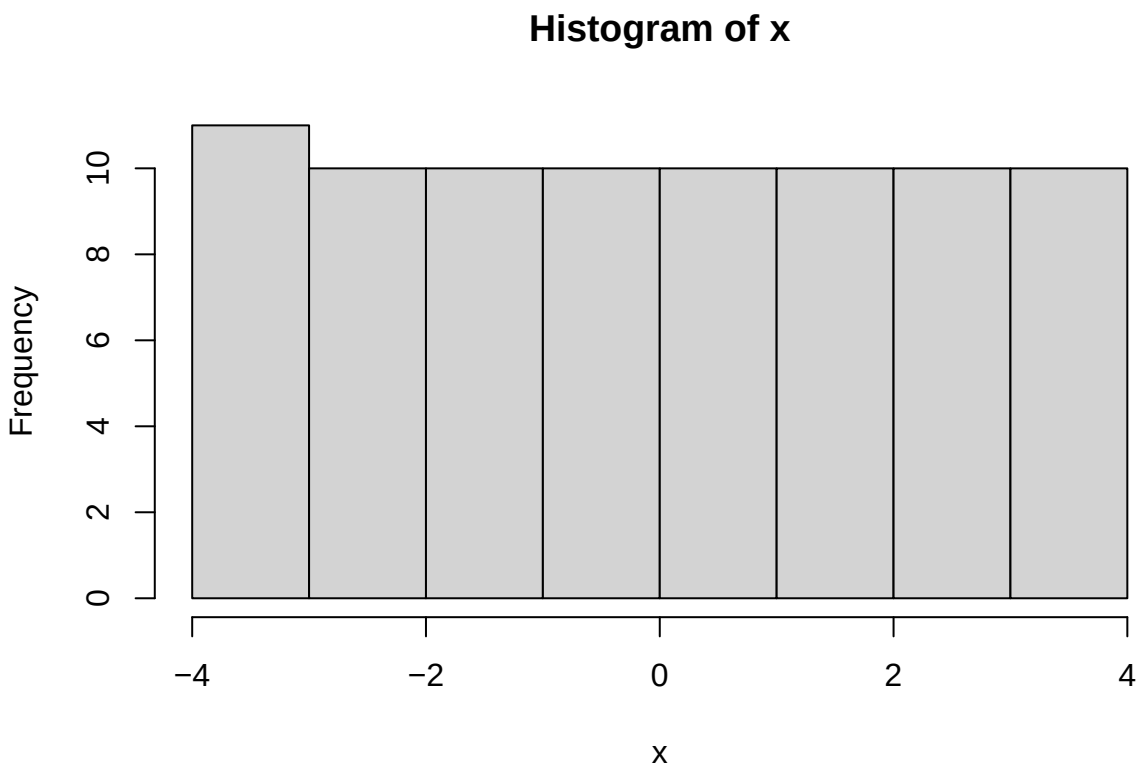
A função pnorm retorna a probabilidade de obter um valor menor do que seu primeiro argumento em uma distribuição normal com a média e o desvio padrão fornecidos.

Estatística Descritiva

mean(x) sd(x) var(x) median(x)

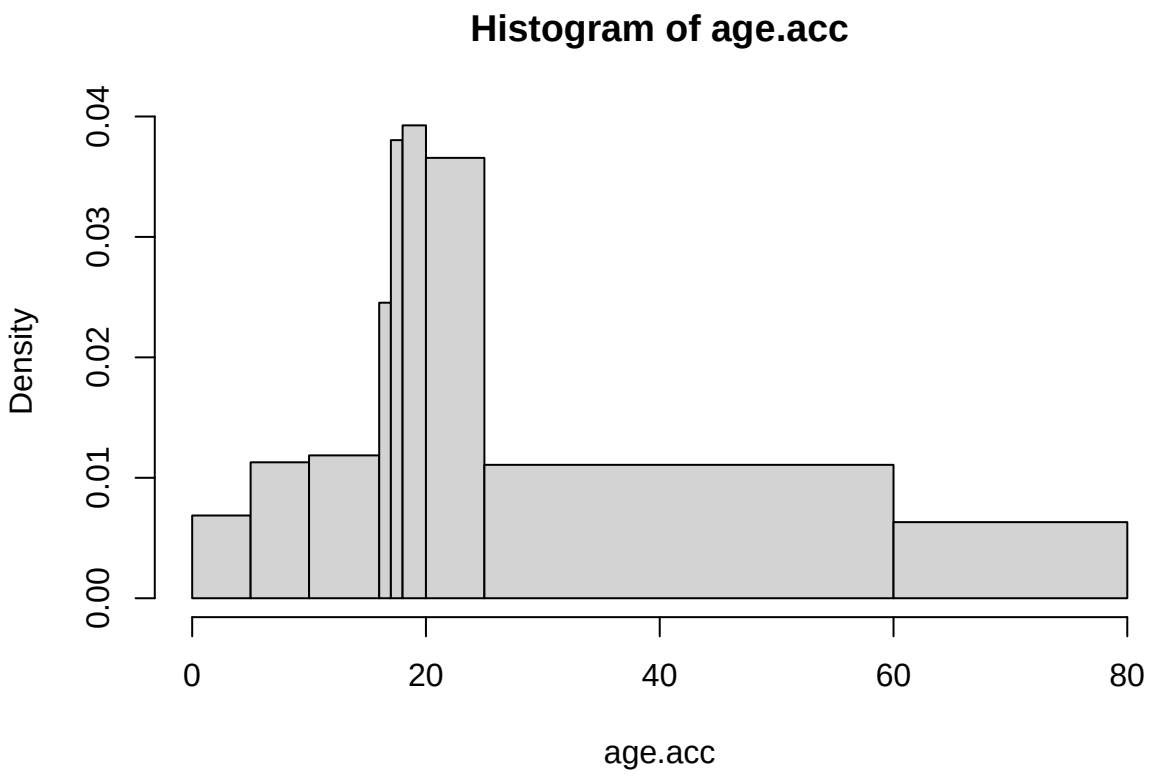
Histograma

```
hist(x)
```



Dividindo as barrar por faixas etárias de $0\check{}4$, $5\check{}9$, $10\check{}15$, $16-17$, $18\check{}19$, $20\check{}24$, $25\check{}59$ e $60\check{}79$ anos de idade. Os dados podem ser inseridos da seguinte forma:

```
mid.age <- c(2.5,7.5,13,16.5,17.5,19,22.5,44.5,70.5)
acc.count <- c(28,46,58,20,31,64,149,316,103)
age.acc <- rep(mid.age,acc.count)

brk <- c(0,5,10,16,17,18,20,25,60,80)
hist(age.acc,breaks=brk)
hist(age.acc,breaks = c(0,5,10,16,17,18,20,25,60,80))
```



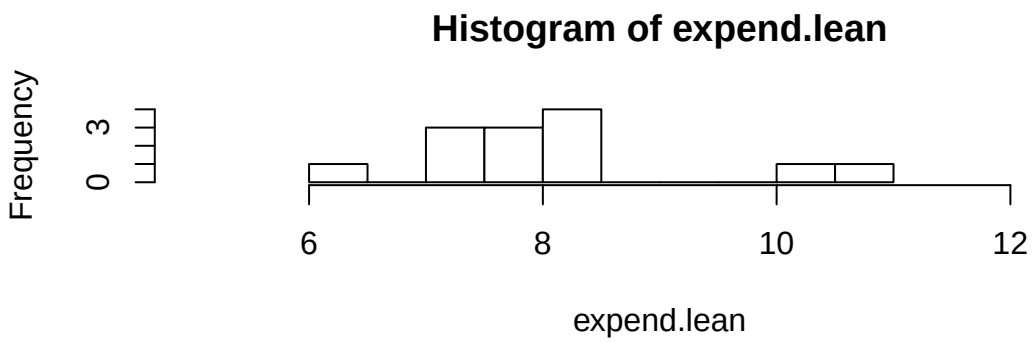
Boxplot

```
library(ISwR)
attach(energy)
```

```
expend.lean <- expend[stature=="lean"]
expend.obese <- expend[stature=="obese"]
```

#Aplicar a função par () no R normal, pois RStudio com erro.

```
par(mfrow=c(2,1))
hist(expend.lean,breaks=10,xlim=c(5,13),ylim=c(0,4),col="white")
hist(expend.obese,breaks=10,xlim=c(5,13),ylim=c(0,4),col="grey")
```




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

boxplot(expend ~ stature)
```

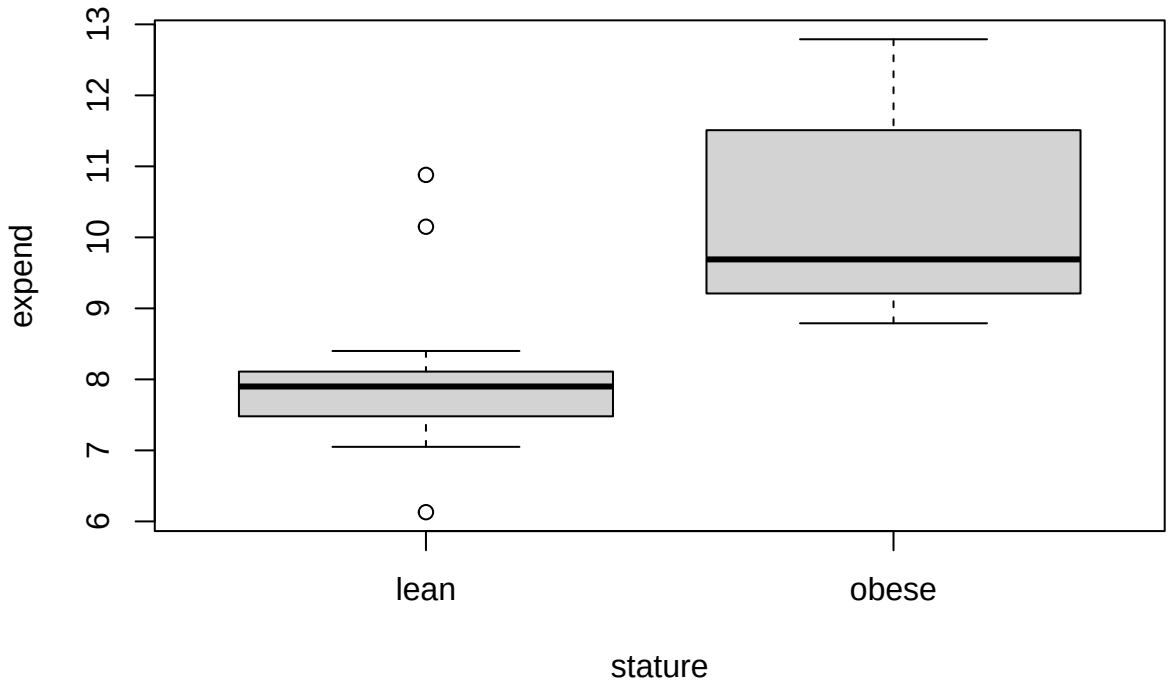
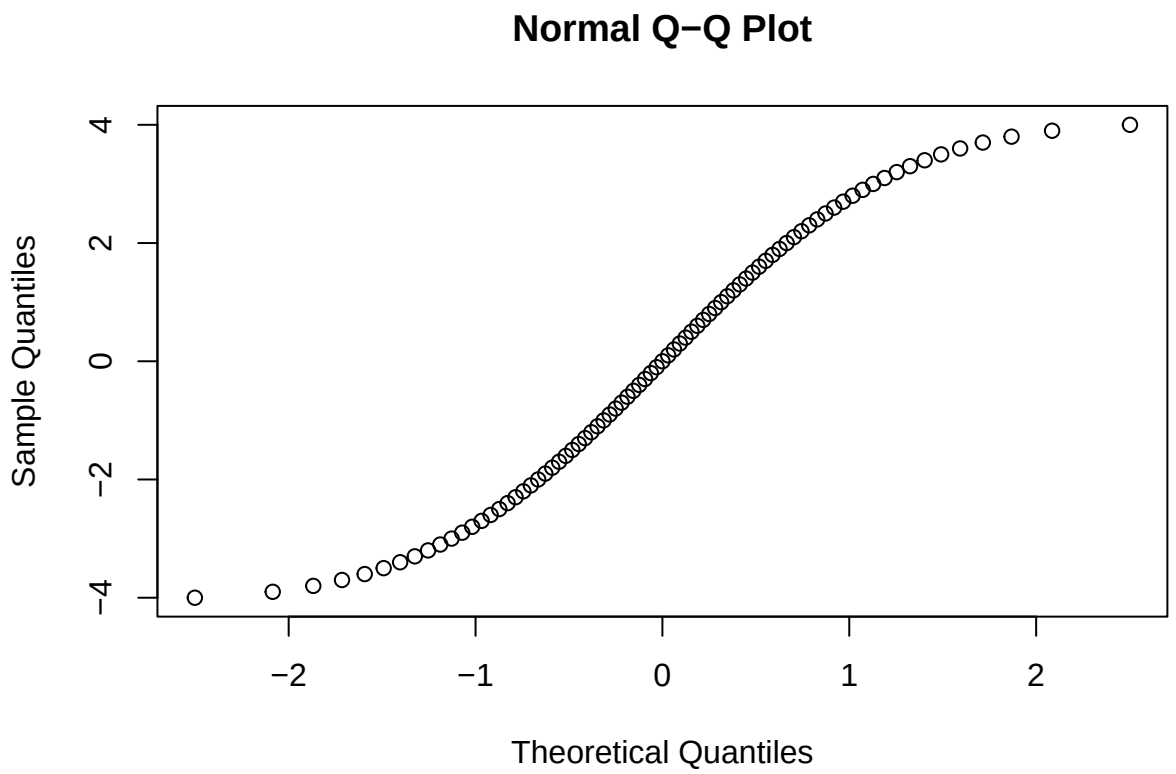


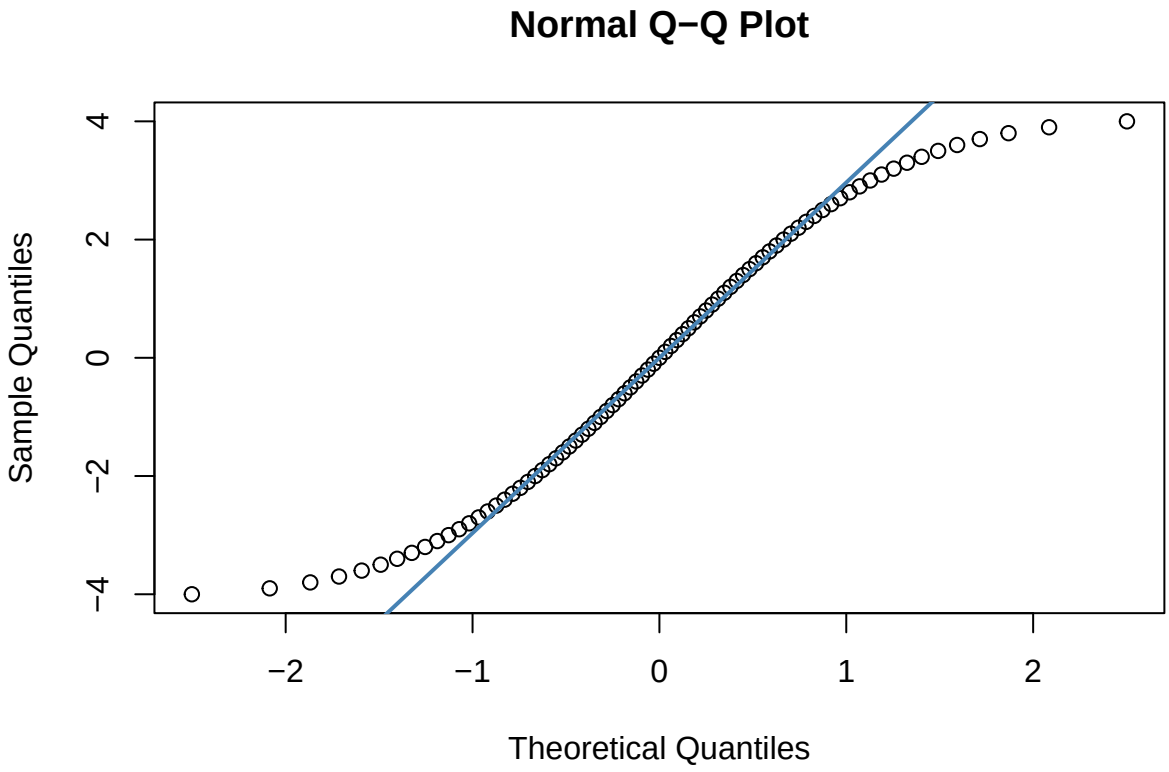
Gráfico Q-Q plot:

Um propósito de calcular a função de distribuição cumulativa empírica (c.d.f.) é ver se os dados podem ser considerados normalmente distribuídos.

```
qqnorm(x)
```




```
qqnorm(x, pch = 1, frame = TRUE) #produz um gráfico QQ  normal da variável
qqline(x, col = "steelblue", lwd = 2) #Linha de referência
```



Resumo Estatístico com o Summary( )

```
attach(juul)
View(juul)
```

age um vetor numérico (anos); menarche é um vetor numérico. Ocorreu menarca (código 1: não, 2: sim)? sex um vetor numérico (1: menino, 2: menina); igf1 um vetor numérico, fator de crescimento semelhante à insulina ($\mu g/l$); tanner um vetor numérico, códigos 1–5: Estágios da puberdade ad modum Tanner. testvol um vetor numérico, volume testicular (ml).

```
summary(igf1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    25.0   202.2   313.5   340.2   462.8   915.0     321
```

```
summary(juul)
```

```
##       age            menarche          sex            igf1
##  Min.   : 0.170   Min.   :1.000   Min.   :1.000   Min.   : 25.0
##  1st Qu.: 9.053   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:202.2
##  Median :12.560   Median :1.000   Median :2.000   Median :313.5
##  Mean   :15.095   Mean   :1.476   Mean   :1.534   Mean   :340.2
##  3rd Qu.:16.855   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:462.8
##  Max.   :83.000   Max.   :2.000   Max.   :2.000   Max.   :915.0
##  NA's   :5        NA's   :635     NA's   :5       NA's   :321
##      tanner        testvol
##  Min.   :1.00   Min.   : 1.000
##  1st Qu.:1.00   1st Qu.: 1.000
##  Median :2.00   Median : 3.000
##  Mean   :2.64   Mean   : 7.896
```

```
##  3rd Qu.:5.00   3rd Qu.:15.000
##  Max.   :5.00   Max.   :30.000
##  NA's   :240    NA's   :859
```

Exemplos com tapply()

A função tapply é utilizada para aplicar um procedimento a diferentes partes dos dados dentro de um array, matriz ou data frame.

Exemplo:

```
attach(juul)
```

```
## The following objects are masked from juul (pos = 3):
##
##     age, igf1, menarche, sex, tanner, testvol
```

```
tapply(igf1, tanner, mean) # Note que a presença de NA não permite o cálculo da média.
```

```
##  1  2  3  4  5
## NA NA NA NA NA
```

```
tapply(igf1, tanner, mean, na.rm=T)
```

```
##        1        2        3        4        5
## 207.4727 352.6714 483.2222 513.0172 465.3344
```

```
aggregate(juul[c("age","igf1")], juul["sex"], mean, na.rm=T)
```

```
##   sex      age     igf1
## 1   1 15.38436 310.8866
## 2   2 14.84363 368.1006
```

```
by(juul, juul["sex"], summary)
```

```
## sex: 1
##       age           menarche        sex         igf1           tanner
##  Min.   : 0.17   Min.   : NA   Min.   :1   Min.   : 29.0   Min.   :1.000
##  1st Qu.: 8.85   1st Qu.: NA   1st Qu.:1   1st Qu.:176.0   1st Qu.:1.000
##  Median :12.38   Median : NA   Median :1   Median :280.0   Median :1.000
##  Mean   :15.38   Mean   :NaN   Mean   :1   Mean   :310.9   Mean   :2.361
##  3rd Qu.:16.77   3rd Qu.: NA   3rd Qu.:1   3rd Qu.:430.2   3rd Qu.:4.000
##  Max.   :83.00   Max.   : NA   Max.   :1   Max.   :915.0   Max.   :5.000
##                  NA's   :621               NA's   :145     NA's   :76
##     testvol
##  Min.   : 1.000
##  1st Qu.: 1.000
##  Median : 3.000
##  Mean   : 7.896
##  3rd Qu.:15.000
##  Max.   :30.000
##  NA's   :141
## ------------------------------------------------------------
## sex: 2
##       age           menarche          sex         igf1           tanner
##  Min.   : 0.25   Min.   :1.000   Min.   :2   Min.   : 25.0   Min.   :1.000
##  1st Qu.: 9.30   1st Qu.:1.000   1st Qu.:2   1st Qu.:233.0   1st Qu.:1.000
##  Median :12.80   Median :1.000   Median :2   Median :352.0   Median :3.000
```

```
##  Mean   :14.84   Mean   :1.476   Mean   :2   Mean   :368.1   Mean   :2.913
##  3rd Qu.:16.93   3rd Qu.:2.000   3rd Qu.:2   3rd Qu.:483.0   3rd Qu.:5.000
##  Max.   :75.12   Max.   :2.000   Max.   :2   Max.   :914.0   Max.   :5.000
##                  NA's   :9                   NA's   :176     NA's   :159
##     testvol
##  Min.   : NA
##  1st Qu.: NA
##  Median : NA
##  Mean   :NaN
##  3rd Qu.: NA
##  Max.   : NA
##  NA's   :713
```

Tabelas

1)

```
table(sex)
```

```
## sex
##   1   2
## 621 713
```

```
table(sex,menarche)
```

```
##    menarche
## sex   1   2
##   1   0   0
##   2 369 335
```

```
table(menarche,tanner)
```

```
##         tanner
## menarche   1   2   3   4   5
##        1 221  43  32  14   2
##        2   1   1   5  26 202
```

```
tanner.sex <- table(tanner,sex)
tanner.sex
```

```
##       sex
## tanner   1   2
##      1 291 224
##      2  55  48
##      3  34  38
##      4  41  40
##      5 124 204
```

```
margin.table(tanner.sex,1) # 1 e 2 é o número do índice que representa a linha (1) ou a coluna (2). É u
```

```
## tanner
##   1   2   3   4   5
## 515 103  72  81 328
```

```
margin.table(tanner.sex,2)
```

```
## sex
```

```
##   1   2
## 545 554
```

```
prop.table(tanner.sex,1) # Frequências relativas em uma tabela são geralmente expressas como proporções
```

```
##       sex
## tanner         1         2
##      1 0.5650485 0.4349515
##      2 0.5339806 0.4660194
##      3 0.4722222 0.5277778
##      4 0.5061728 0.4938272
##      5 0.3780488 0.6219512
```

```
prop.table(tanner.sex,2)
```

```
##       sex
## tanner          1          2
##      1 0.53394495 0.40433213
##      2 0.10091743 0.08664260
##      3 0.06238532 0.06859206
##      4 0.07522936 0.07220217
##      5 0.22752294 0.36823105
```

2)

```
caff.marital <- matrix(c(652,1537,598,242,36,46,38,21,218,327,106,67), nrow=3,byrow=T)
caff.marital
```

```
##      [,1] [,2] [,3] [,4]
## [1,]  652 1537  598  242
## [2,]   36   46   38   21
## [3,]  218  327  106   67
```

```
colnames(caff.marital) <- c("0","1-150","151-300",">300")
rownames(caff.marital) <- c("Married","Prev.married","Single")
caff.marital
```

```
##                0 1-150 151-300 >300
## Married      652  1537     598  242
## Prev.married  36    46      38   21
## Single       218   327     106   67
```

```
names(dimnames(caff.marital)) <- c("marital","consumption")
caff.marital
```

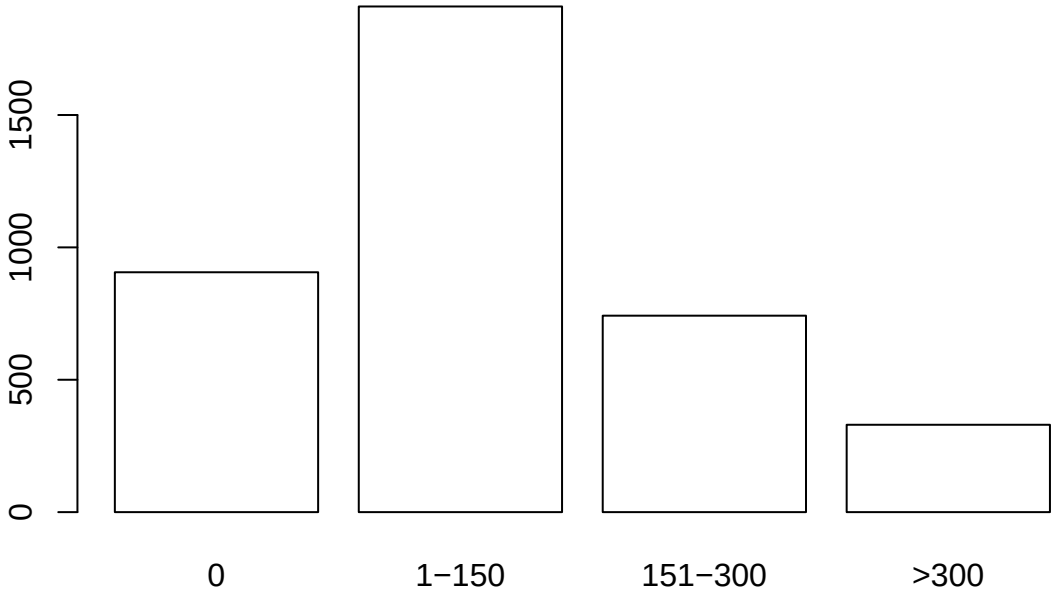
```
##              consumption
## marital            0 1-150 151-300 >300
##   Married        652  1537     598  242
##   Prev.married    36    46      38   21
##   Single         218   327     106   67
```

```
total.caff <- margin.table(caff.marital,2)
total.caff
```

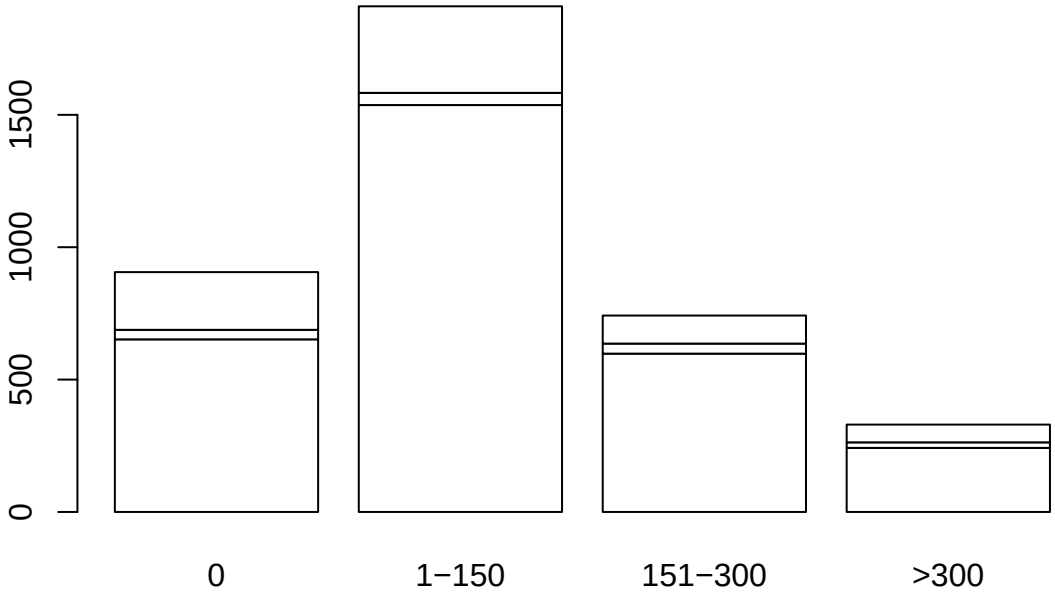
```
## consumption
##       0   1-150 151-300    >300
##     906    1910     742     330
```

Barplot

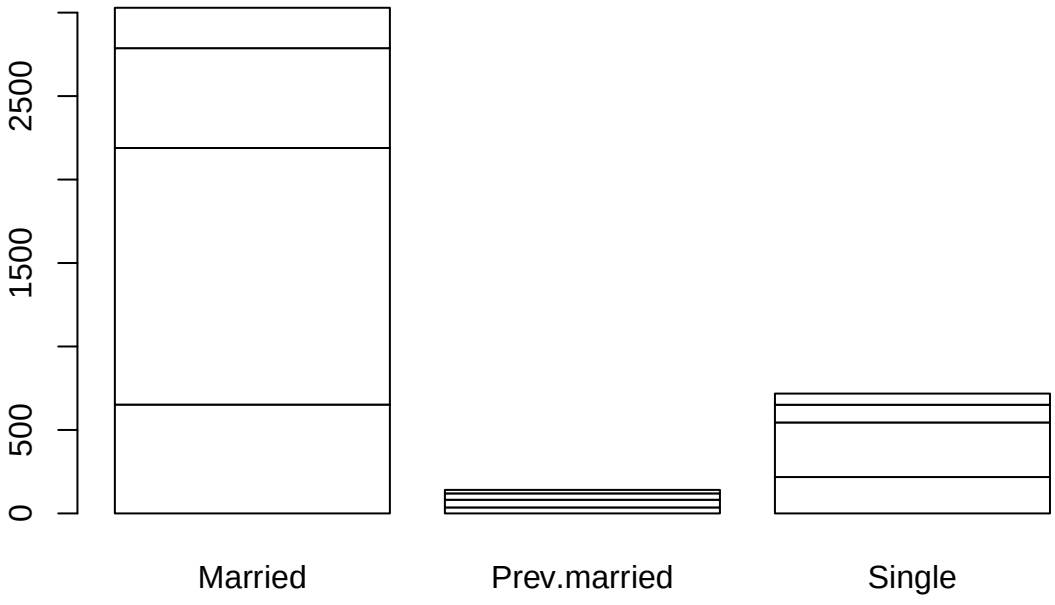
```
barplot(total.caff, col="white")
```



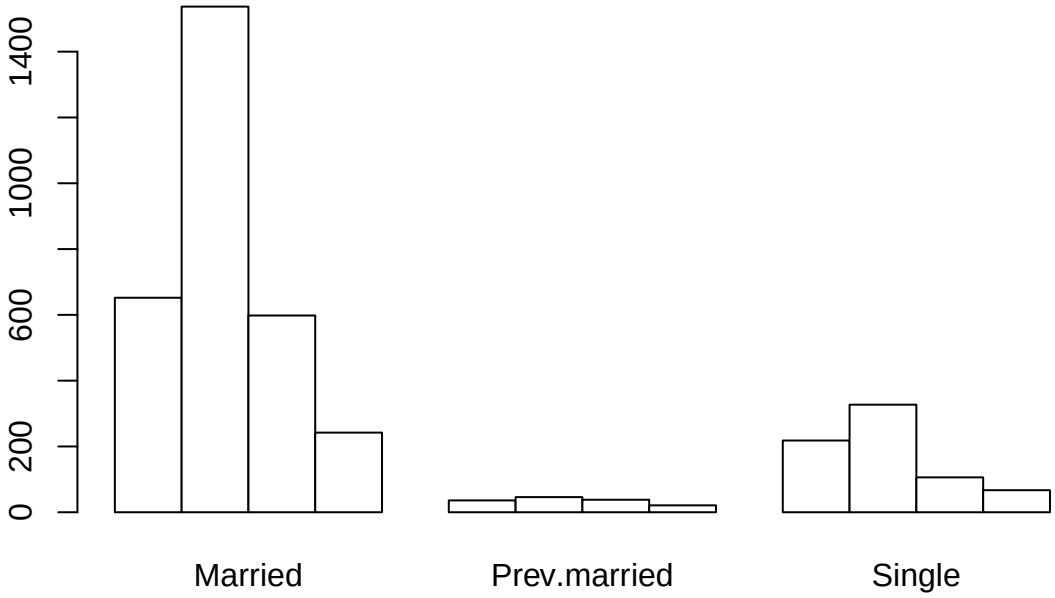

```
barplot(caff.marital, col="white")
```



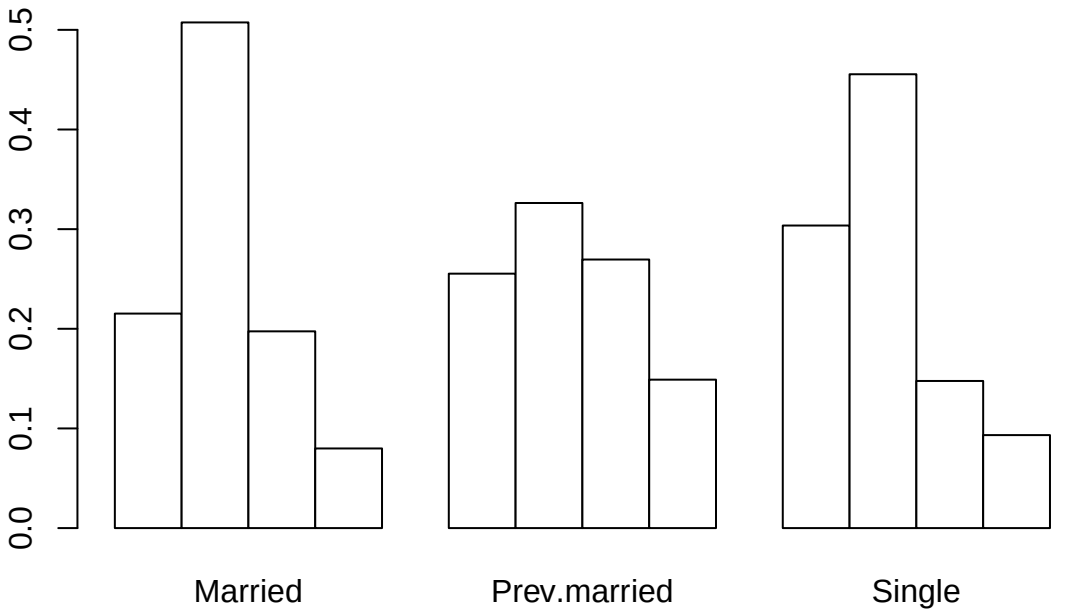

```
barplot(t(caff.marital), col="white")
```




```
barplot(t(caff.marital), col="white", beside=T) #beside = T - Se você quiser colocar as contribuições d
```




```
barplot(prop.table(t(caff.marital),2), col="white", beside=T)
```




```
barplot(prop.table(t(caff.marital),2),beside=T,
        legend.text=colnames(caff.marital),
        col=c("white","grey80","grey50","black"))
```

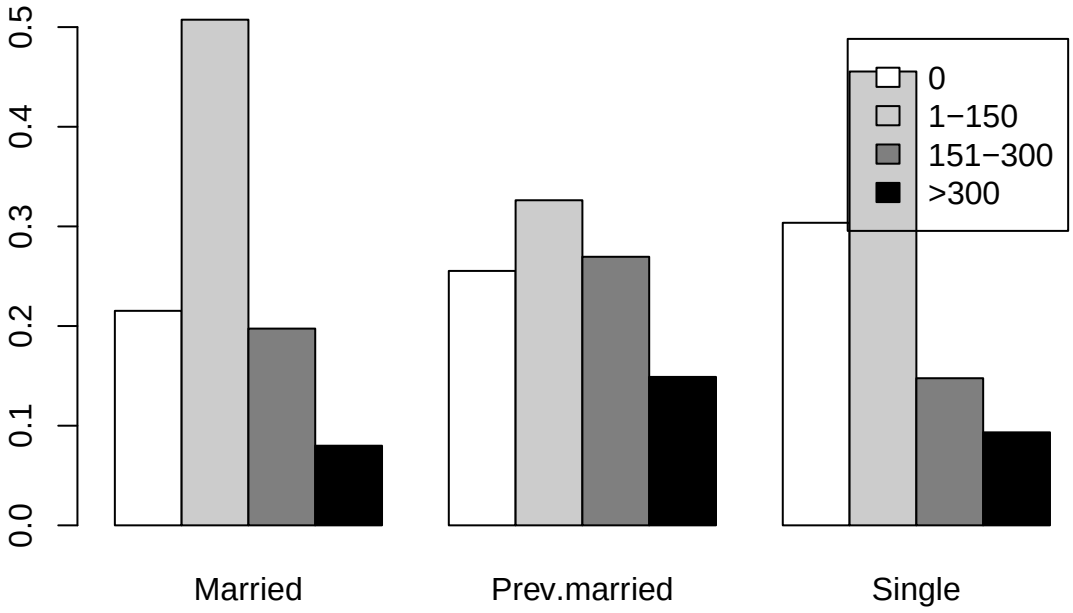
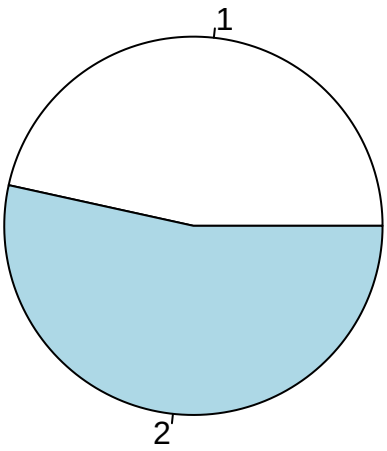


Gráfico de pizza

```
pie(table(sex))
```



Função similar ao table

```
xtabs(~ tanner + sex, data=juul)
```

```
##       sex
## tanner   1   2
##      1 291 224
##      2  55  48
##      3  34  38
##      4  41  40
##      5 124 204
```

```
ftable(coma + diab ~ dgn, data=stroke)
```

```
##     coma  No     Yes
##     diab  No Yes  No Yes
## dgn
## ICH       53   6  19   1
## ID       143  21  23   3
## INF      411  64  23   2
## SAH       38   0   9   0
```

Frequência absoluta e Relativa

Análise Descritiva Univariada: Variáveis Qualitativas

```
sex_prop <- table(sex)
sexo.tbp <- prop.table(sex_prop) # Dá os valores em cada célula divididos pela soma total:

#install.packages("fields")
require(fields)
```

```
## Loading required package: fields
```

```
## Loading required package: spam
```

```
## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
```

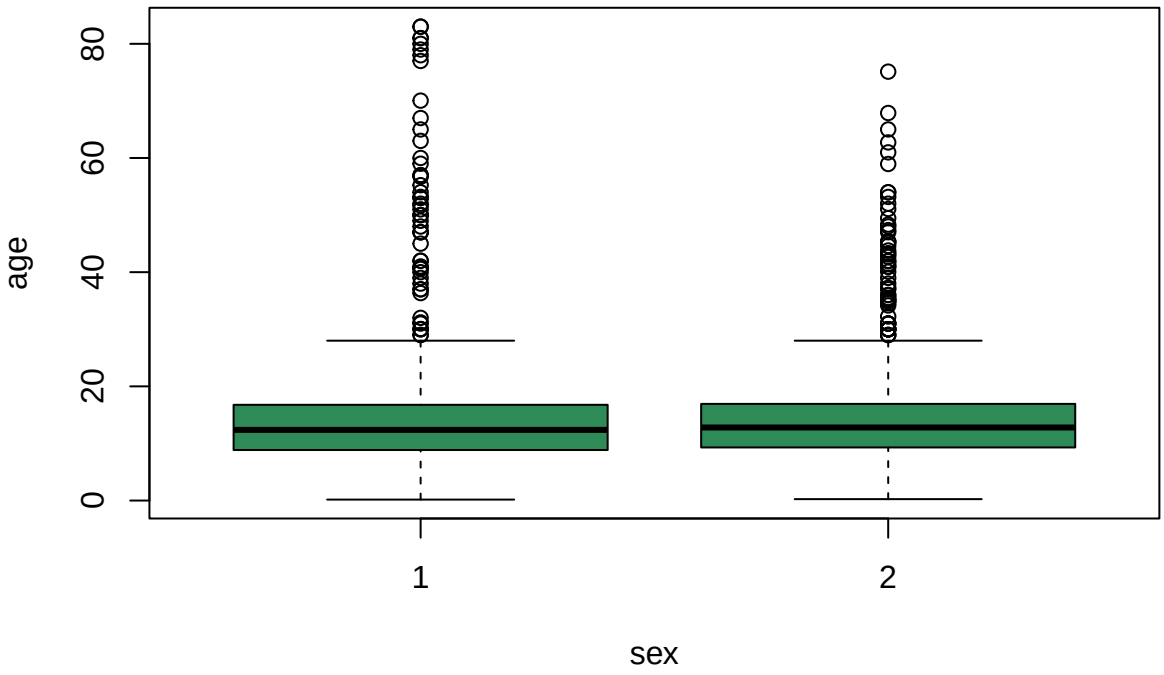
```
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridis

## Loading required package: viridisLite

##
## Try help(fields) to get started.
```

```
tab<- cbind(stats(age), stats(sex))
colnames(tab)<- c("Idade","Sexo")
round(tab,2)
```

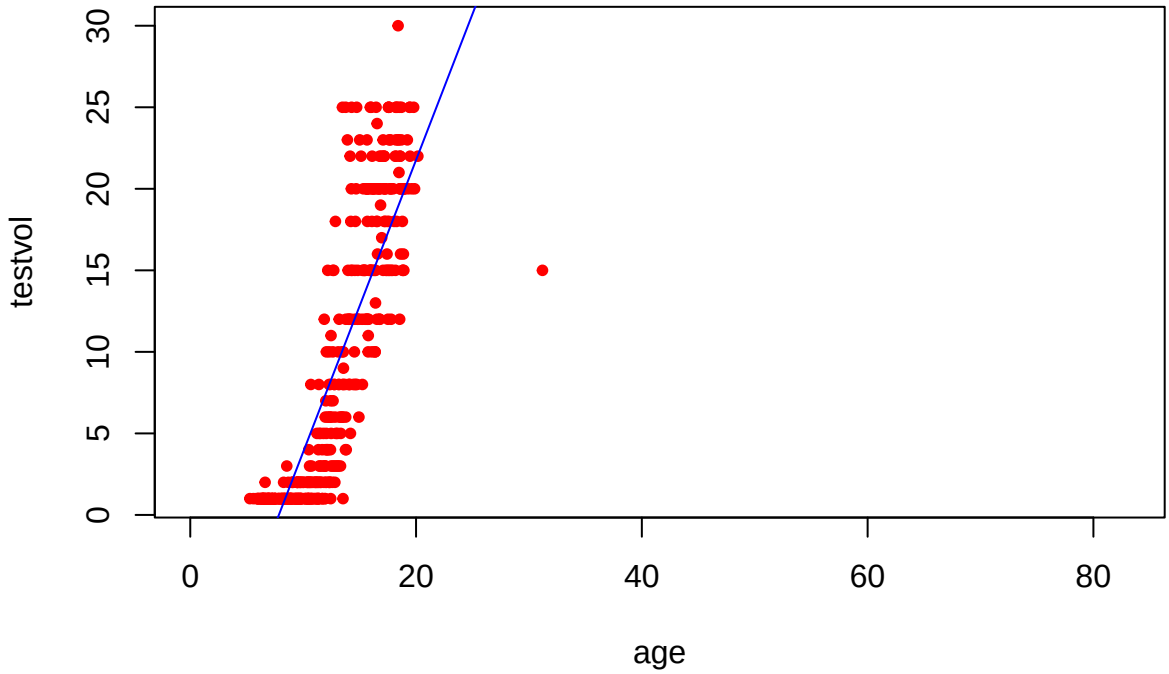
```
##                   Idade    Sexo
## N              1334.00 1334.00
## mean             15.10    1.53
## Std.Dev.         11.25    0.50
## min               0.17    1.00
## Q1                9.05    1.00
## median           12.56    2.00
## Q3               16.86    2.00
## max              83.00    2.00
## missing values    5.00    5.00
```

Análise Descritiva Bivariada: Quantitativa x Qualitativa

```
par(mfrow=c(1,1))
boxplot(age ~ sex, col="seagreen4", xlab="sex", ylab="age")
```




```
plot(age, testvol, pch=20, col="red")
abline(lm(testvol ~ age), col="blue") #Linha de regressão linear
```



Comparação de Variância. Teste t Emparelhado.

```
daily.intake <- c(5260,5470,5640,6180,6390,6515,6805,7515,7515,8230,8770)

length(daily.intake)
```

```
## [1] 11
```

```
mean(daily.intake)
```

```
## [1] 6753.636
```

```
sd(daily.intake)
```

```
## [1] 1142.123
```

```
quantile(daily.intake)
```

```
##   0%  25%  50%  75% 100%
## 5260 5910 6515 7515 8770
```

```
summary(daily.intake)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5260    5910    6515    6754    7515    8770
```

Os testes t são baseados na suposição de que os dados vêm da distribuição normal. No caso de uma amostra, temos, portanto, os dados $x_1, \ldots, x_n$ assumidos como realizações independentes de variáveis aleatórias com distribuição $N(\mu, \sigma^2)$, o que denota a distribuição normal com média μ e variância σ^2, e nós deseja testar a hipótese nula de que $\mu = \mu_0$. Podemos estimar os parâmetros μ e σ pela média empírica $\bar{x}$ e desvio padrão s, embora devamos perceber que nunca poderíamos localizar seus valores com exatidão.

Você pode querer investigar se a ingestão de energia das mulheres se desvia sistematicamente de um valor recomendado de $7725kJ$. Assumindo que os dados vêm de uma distribuição normal, o objetivo é testar se essa distribuição pode ter média $\mu = 7725$. Isso é feito com t.test da seguinte maneira:

```
t.test(daily.intake, mu=7725)
```

```
##
##  One Sample t-test
##
## data:  daily.intake
## t = -2.8208, df = 10, p-value = 0.01814
## alternative hypothesis: true mean is not equal to 7725
## 95 percent confidence interval:
##  5986.348 7520.925
## sample estimates:
## mean of x
##  6753.636
```

Os testes t são bastante robustos contra desvios da distribuição normal, especialmente em amostras maiores, mas às vezes você deseja evitar fazer essa suposição. Para este fim, os métodos livres de distribuição são convenientes.

Teste de Wilcoxon (Teste não paramétrico)

É um método livre de distribuição. O teste de classificação sinalizada de Wilcoxon para uma amostra é uma alternativa não paramétrica ao teste t para uma amostra quando os dados não podem ser considerados como normalmente distribuídos. É usado para determinar se a mediana da amostra é igual a um valor padrão conhecido (ou seja, um valor teórico).

```
wilcox.test(daily.intake, mu=7725, alternative = "two.sided")
```

```
## Warning in wilcox.test.default(daily.intake, mu = 7725, alternative =
## "two.sided"): cannot compute exact p-value with ties
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  daily.intake
## V = 8, p-value = 0.0293
## alternative hypothesis: true location is not equal to 7725
```

A hipótese nula é que as distribuições são as mesmas e, portanto, têm a mesma mediana. A alternativa é bilateral. A hipótese alternativa é que o deslocamento da posição verdadeira não é igual a zero, a distribuição de uma população é deslocada para a esquerda ou para a direita da outra", o que implica diferentes medianas); alternative = "two.sided", "greater" ou "less".

Teste para duas amostras (Dados Agrupados)

O teste t de duas amostras é usado para testar a hipótese de que duas amostras podem vir de distribuições com a mesma média.

```
attach(energy)
```

```
## The following objects are masked from energy (pos = 9):
##
##     expend, stature
```

```
#View(energy)

t.test(expend~stature)
```

```
##
##  Welch Two Sample t-test
##
## data:  expend by stature
## t = -3.8555, df = 15.919, p-value = 0.001411
## alternative hypothesis: true difference in means between group lean and group obese is not equal to 0
## 95 percent confidence interval:
##  -3.459167 -1.004081
## sample estimates:
##  mean in group lean mean in group obese
##            8.066154           10.297778
```

Mesmo que seja possível em R realizar o teste t de duas amostras sem a suposição de que as variâncias são as mesmas, você ainda pode estar interessado em testar essa suposição, e R fornece a função var.test para esse fim, implementando um F teste sobre a razão das variâncias do grupo. É chamado da mesma forma que t.test:

Comparação de variâncias

```
var.test(expend~stature)
```

```
##
##  F test to compare two variances
##
## data:  expend by stature
## F = 0.78445, num df = 12, denom df = 8, p-value = 0.6797
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
```

```
##  0.1867876 2.7547991
## sample estimates:
## ratio of variances
##           0.784446
```

```
t.test(expend~stature, var.equal=TRUE) # No caso de as variâncias entre os grupos serem as mesmas.
```

```
##
##  Two Sample t-test
##
## data:  expend by stature
## t = -3.9456, df = 20, p-value = 0.000799
## alternative hypothesis: true difference in means between group lean and group obese is not equal to 0
## 95 percent confidence interval:
##  -3.411451 -1.051796
## sample estimates:
##  mean in group lean mean in group obese
##            8.066154           10.297778
```

Teste de Wilcoxon para duas amostras:

É aplicado em situações em que se tem um par de amostras independentes e se quer testar se as populações que deram origem a essas amostras podem ser consideradas semelhantes ou não.

Usando o Wilcoxon Signed-Rank Test, podemos decidir se as distribuições de população de dados correspondentes são idênticas, sem supor que seguem a distribuição normal.

Para realizar o teste de Wilcoxon de duas amostras, comparando as médias de duas amostras independentes (x e y), a função R wilcox.test ().

A hipótese nula é que as distribuições são as mesmas e, portanto, têm a mesma mediana.

```
wilcox.test(expend~stature)
```

```
## Warning in wilcox.test.default(x = DATA[[1L]], y = DATA[[2L]], ...): cannot
## compute exact p-value with ties
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  expend by stature
## W = 12, p-value = 0.002122
## alternative hypothesis: true location shift is not equal to 0
```

Note que p-value = 0.002122. Assim, ao nível de significância de 5%, rejeitamos H0 e concluimos que as diferenças entre os grupos são estatisticamente significativas.

Teste t pareado

```
attach(intake)
intake
```

```
##     pre post
## 1  5260 3910
## 2  5470 4220
## 3  5640 3885
## 4  6180 5160
```

```
## 5  6390 5645
## 6  6515 4680
## 7  6805 5265
## 8  7515 5975
## 9  7515 6790
## 10 8230 6900
## 11 8770 7335
```

```
post - pre
```

```
##  [1] -1350 -1250 -1755 -1020  -745 -1835 -1540 -1540  -725 -1330 -1435
```

```
t.test(pre, post, paired=T)
```

```
##
##  Paired t-test
##
## data:  pre and post
## t = 11.941, df = 10, p-value = 3.059e-07
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  1074.072 1566.838
## sample estimates:
## mean difference
##        1320.455
```

O teste de Wilcoxon de pares combinados

```
wilcox.test(pre, post, paired=TRUE)
```

```
## Warning in wilcox.test.default(pre, post, paired = TRUE): cannot compute exact
## p-value with ties
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  pre and post
## V = 66, p-value = 0.00384
## alternative hypothesis: true location shift is not equal to 0
```