



# Introdução ao uso do em Ciências da Saúde

## 4 - Análise de dados univariada

Fúlvio Borges Nedel

Departamento de Saúde Pública – SPB

Centro de Ciências da Saúde – CCS

Universidade Federal de Santa Catarina – UFSC

*Grups de Recerca d'Amèrica i Àfrica Llatines – GRAAL*

<http://graal.uab.cat>

25 de outubro de 2017



Introdução ao  
R  
4. Análise  
univariada 2/31

Fúlvio Nedel  
SPB/UFSC

Introdução  
Objetivos da  
análise  
Análise  
univariada  
Variáveis  
numéricas  
Variáveis  
categóricas

- 1 **Introdução**
  - **Objetivos da análise**
- 2 **Análise univariada**
  - **Variáveis numéricas**
  - **Variáveis categóricas**



## Objetivos da análise

- 1 Descrever o Índice de Massa Corporal (IMC) e analisar fatores associados à sua média.
- 2 Descrever a frequência de categorias do estado nutricional e analisar fatores possivelmente associados à obesidade:
  - 1 sexo
  - 2 idade
  - 3 condição socioeconômica
  - 4 participação em grupos de promoção da saúde



Introdução ao  
R  
4. Análise  
univariada 4/31

Fúlvio Nedel  
SPB/UFSC

Introdução  
Objetivos da  
análise

Análise  
univariada

Variáveis  
numéricas  
Variáveis  
categóricas

- 1 Introdução
  - Objetivos da análise
- 2 Análise univariada
  - Variáveis numéricas
  - Variáveis categóricas



- Toda análise deve começar com uma descrição univariada
  - completa e
  - adequada ao tipo de variável (contínua, categórica, ...)
- Eventualmente queremos calcular intervalos de confiança para médias e proporções
- Crie uma nova linha comentada em seu arquivo de sintaxe. Algo como:

```
# Análise dos dados  
# -----  
### Análise univariada  
...
```



- Inicie com uma sessão vazia e carregue o arquivo de dados
- Verá que ele agora tem dois objetos,
  - o banco de dados com as variáveis selecionadas antes da transformação e
  - o banco com as variáveis transformadas, para a análise.

```
rm(list=ls())  
load('cursoR.RData')  
ls()  
[1] "cursoR" "cursoR2"  
class(cursoR)  
[1] "data.frame"  
class(cursoR2)  
[1] "data.frame"
```

# Descrição – análise univariada

Uma primeira olhada no banco



```
library(Hmisc)
describe(cursoR2[1:3])
cursoR2[1:3]
```

**A função** `describe{Hmisc}`

**Oferece um resumo um pouco mais detalhado e de muito melhor leitura que summary**

3 Variables 300 Observations

idade

n	missing	distinct	Info	Mean	Gmd	.05	.10
300	0	60	0.999	64.57	14.49	43	49
.25	.50	.75	.90	.95			
56	65	74	80	85			

lowest : 23 25 26 29 34, highest: 89 91 93 94 95

sexo : u8. Sexo:

n	missing	distinct
300	0	2

Value	Feminino	Masculino
Frequency	198	102
Proportion	0.66	0.34

imc

n	missing	distinct	Info	Mean	Gmd	.05	.10
299	1	252	1	27.77	5.554	21.15	22.10
.25	.50	.75	.90	.95			
24.22	27.18	30.48	34.76	37.21			

lowest : 17.30104 17.90123 18.35937 18.48977 18.66201  
highest: 40.62500 40.81633 41.20708 42.09964 46.13610

Introdução ao  
R  
4. Análise  
univariada 7/31

Fúlvio Nedel  
SPB/UFSC

Introdução  
Objetivos da  
análise

Análise  
univariada

Variáveis  
numéricas  
Variáveis  
categóricas



```
describe(cursoR2[4:6])
```

```
cursoR2[4:6]
```

```
3 Variables      300 Observations
```

```
-----  
imccat : Estado nutricional
```

```
      n missing distinct
```

```
299      1      3
```

```
Value      normal sobrepeso obesidade
```

```
Frequency      99      116      84
```

```
Proportion 0.331      0.388      0.281
```

```
-----  
obeso
```

```
      n missing distinct
```

```
299      1      2
```

```
Value      sim      não
```

```
Frequency      84      215
```

```
Proportion 0.281 0.719
```

```
-----  
abep2 : Classificação ABEP agrupada
```

```
      n missing distinct
```

```
282      18      3
```

```
Value      A/B      C      D/E
```

```
Frequency      73      178      31
```

```
Proportion 0.259 0.631 0.110
```





```
describe(cursoR2[7:8])
```

```
cursoR2[7:8]
```

```
  2 Variables      300 Observations
```

```
-----  
grupo : Participa em grupo de hipertensos ou diabéticos
```

```
      n missing distinct  
298      2         2
```

```
Value      Não   Sim  
Frequency  285   13  
Proportion 0.956 0.044
```

```
-----  
id
```

```
      n missing distinct  
300      0         300
```

```
lowest : 1    10  100 101 102, highest: 95  96  97  98  99  
-----
```



```
describe(cursoR2[7:8])
```

```
cursoR2[7:8]
```

```
  2 Variables      300 Observations
-----
grupo : Participa em grupo de hipertensos ou diabéticos
      n missing distinct
      298      2        2
```

```
Value      Não      Sim
Frequency   285     13
Proportion 0.956 0.044
```

```
-----
id
  n missing distinct
  300      0        300
```

```
lowest : 1   10  100 101 102, highest: 95  96  97  98  99
-----
```

```
class(cursoR2$id)
```

```
[1] "character"
```

```
describe(as.numeric(cursoR2$id))
```

```
as.numeric(cursoR2$id)
```

```
  n missing distinct      Info      Mean      Gmd      .05
  300      0        300        1    150.5    100.3    15.95
  .10     .25     .50     .75     .90     .95
 30.90   75.75   150.50  225.25  270.10  285.05
```



- há vários pacotes, além do Hmisc, para facilitar a análise e apresentação de dados epidemiológicos: `Epi`, `epitools`, `epibasix`, `gmodels`...
- instale e carregue o pacote `epiDisplay`

```
install.packages("epiDisplay", dep=T)
```

```
library(epiDisplay)
```

```
Loading required package: foreign
```

```
Loading required package: MASS
```

```
Loading required package: nnet
```

```
Attaching package: 'epiDisplay'
```

```
The following object is masked from 'package:ggplot2':  
  alpha
```

```
The following object is masked from 'package:lattice':  
  dotplot
```

## Medidas de

- tendência central (média, mediana, moda)
- posição (quantis)
- dispersão (amplitude, desvio-padrão, coeficiente de variação, intervalo interquartilico)
- forma (assimetria e curtose)
- já vimos algumas funções (`mean`, `sd`)

## Gráficos

- histograma
- boxplot
- densidade
- polígono de frequência acumulada
- de barras, para variáveis discretas
- de pontos, ramo-e-folhas. . .



```
attach(cursorR2)
```

Média

```
mean(imc)
```

```
[1] NA
```

```
summary(imc)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
17.30	24.22	27.18	27.77	30.48	46.14	1



## Média

```
mean(imc, na.rm = T)  
[1] 27.76622
```

## Mediana

```
median(imc, na.rm = T)  
[1] 27.1809
```

## Amplitude

```
range(imc, na.rm = T)  
[1] 17.30104 46.13610
```

## Desvio-padrão e coeficiente de variação

```
sd(imc, na.rm = T)  
[1] 4.971806  
round(sd(imc, na.rm = T)/mean(imc, na.rm = T)*100, 2)  
[1] 17.91
```

## Assimetria

```
e1071::skewness(imc, na.rm = TRUE)
[1] 0.6374804
```

## Curtose

```
e1071::kurtosis(imc, na.rm = T)
[1] 0.2573438
```

## Quantiles

```
quantile(imc, na.rm = T)
      0%      25%      50%      75%     100%
17.30104 24.21875 27.18090 30.47595 46.13610
quantile(imc, p = c(.025, .975), na.rm = T)
      2.5%     97.5%
19.79113 38.60288
```

## Intervalo de confiança para a média

```
ci(imc)
      n      mean      sd      se lower95ci upper95ci
299 27.76622 4.971806 0.287527 27.20038 28.33206
```



## Moda

- A moda é uma característica tanto de variáveis numéricas como categóricas
- É o valor mais frequente, ou, em variáveis contínuas, o de maior densidade de frequência.
- Não há entre as funções básicas do R, uma que calcule a moda.





## Moda

- A moda é uma característica tanto de variáveis numéricas como categóricas
- É o valor mais frequente, ou, em variáveis contínuas, o de maior densidade de frequência.
- Não há entre as funções básicas do R, uma que calcule a moda.
- Podemos construí-la:

```
moda <- function(x) {  
  d = density(x, na.rm = T)  
  i <- which.max(d$y)  
  d$x[i]  
}
```

```
moda(imc)
```

```
[1] 25.40029
```

Introdução ao  
R

4. Análise  
univariada  
15/31

Fúlvio Nedel  
SPB/UFSC

Introdução

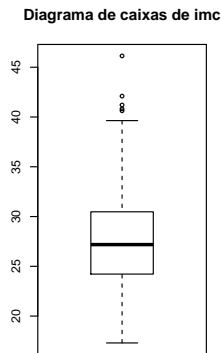
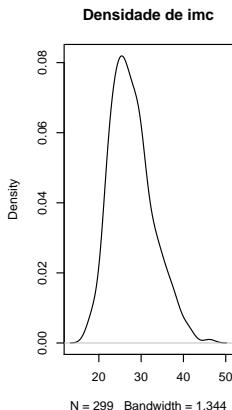
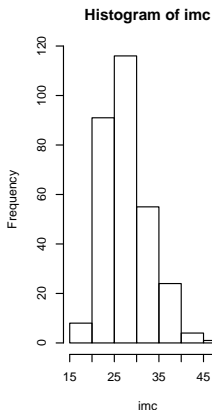
Objetivos da  
análise

Análise  
univariada

Variáveis  
numéricas

Variáveis  
categóricas

```
par(mfrow = c(1,3))  
hist(imc)  
plot(density(imc, na.rm = T), main='Densidade de imc')  
boxplot(imc, main = 'Diagrama de caixas de imc')
```





- frequência absoluta e relativa
- frequências acumuladas
- gráficos de barras
- intervalos de confiança para proporções
  - distribuição binomial
  - distribuição multinomial
- já vimos algumas funções (`table`, `cumsum`)



## Estado nutricional

Com as funções básicas

### *Frequência absoluta*

```
table(imccat)
```

```
imccat
```

normal	sobrepeso	obesidade
99	116	84

### *Acrescentar o total*

```
addmargins(table(imccat))
```

```
imccat
```

normal	sobrepeso	obesidade	Sum
99	116	84	299

### *Frequência acumulada*

```
cumsum(table(imccat))
```

normal	sobrepeso	obesidade
99	215	299



## Estado nutricional

Com as funções básicas (cont.)

### *Frequência relativa*

```
prop.table(table(imccat))  
  
imccat  
    normal sobrepeso obesidade  
0.3311037 0.3879599 0.2809365
```

### *Porcentagem, arredondada para um decimal*

```
round(prop.table(table(imccat))*100, 1)  
  
imccat  
    normal sobrepeso obesidade  
    33.1      38.8      28.1
```

### *Porcentagem acumulada*

```
cumsum(round(prop.table(table(imccat))*100, 1))  
  
    normal sobrepeso obesidade  
    33.1      71.9     100.0
```



Introdução ao  
R

4. Análise  
univariada  
19/31

Fúlvio Nedel  
SPB/UFSC

Introdução  
Objetivos da  
análise

Análise  
univariada

Variáveis  
numéricas

Variáveis  
categóricas

**Mas...**

- um comando pra cada coisa?!



19/03/2019

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

14h30

Introdução ao  
R

4. Análise  
univariada  
19/31

Fúlvio Nedel  
SPB/UFSC

Introdução  
Objetivos da  
análise

Análise  
univariada

Variáveis  
numéricas

Variáveis  
categóricas

**Mas...**

- um comando pra cada coisa?!
- até pras proporções?!!



**Mas...**

- um comando pra cada coisa?!
- até pras proporções?!!
- até pro total da tabela?!!!





## Mas...

- um comando pra cada coisa?!
- até pras proporções?!!
- até pro total da tabela?!!!



## Mas...

- um comando pra cada coisa?!
- até pras proporções?!!
- até pro total da tabela?!!!

## Bom...

- o que pode parecer limitação



## Mas...

- um comando pra cada coisa?!
- até pras proporções?!!
- até pro total da tabela?!!!

## Bom...

- o que pode parecer limitação
- **representa versatilidade**



## Mas...

- um comando pra cada coisa?!
- até pras proporções?!!
- até pro total da tabela?!!!

## Bom...

- o que pode parecer limitação
- representa versatilidade
- o usuário pode construir sua tabela



## Mas...

- um comando pra cada coisa?!
- até pras proporções?!!
- até pro total da tabela?!!!

## Bom...

- o que pode parecer limitação
- representa versatilidade
- o usuário pode construir sua tabela
- ou construir funções com o modelo de tabela que deseja

Introdução ao  
R

4. Análise  
univariada  
19/31

Fúlvio Nedel  
SPB/UFSC

Introdução  
Objetivos da  
análise

Análise  
univariada

Variáveis  
numéricas

Variáveis  
categóricas

**Mas...**

- um comando pra cada coisa?!
- até pras proporções?!!
- até pro total da tabela?!!!

**Bom...**

- o que pode parecer limitação
- representa versatilidade
- o usuário pode construir sua tabela
- ou construir funções com o modelo de tabela que deseje
- **ou usar funções de outros pacotes**



## Estado nutricional

Com as funções básicas (cont.)

```
x <- table(imccat)
px <- round(prop.table(table(imccat))*100, 1)
cumx <- cumsum(x)
cumpx <- cumsum(px)
miolo <- cbind(Freq=x, '%'=px, Freq.acum=cumx, '%acum'=cumpx)
Total <- c(sum(x), sum(px), sum(x), sum(px))
tab.imccat <- rbind(miolo, Total)
```

tab.imccat				
	Freq	%	Freq.acum	%acum
normal	99	33.1	99	33.1
sobrepeso	116	38.8	215	71.9
obesidade	84	28.1	299	100.0
Total	299	100.0	299	100.0

tab.imccat[,c(1,3,2,4)]				
	Freq	Freq.acum	%	%acum
normal	99	99	33.1	33.1
sobrepeso	116	215	38.8	71.9
obesidade	84	299	28.1	100.0
Total	299	299	100.0	100.0

`t(tab.imccat[,1:2])` # note os decimais nas frequências absolutas :(

	normal	sobrepeso	obesidade	Total
Freq	99.0	116.0	84.0	299
%	33.1	38.8	28.1	100

## Introdução ao R

### 4. Análise univariada

21/31

Fúlvio Nedel  
SPB/UFSC

## Introdução

### Objetivos da análise

### Análise univariada

### Variáveis numéricas

### Variáveis categóricas

```
tabua <- function(x, ...){  
  tab <- table(x)  
  ptab <- round(prop.table(tab)*100, 1)  
  cumtab <- cumsum(tab)  
  cumptab <- cumsum(ptab)  
  miolo <- cbind(Freq=tab, '%'=ptab, Freq.acum=cumtab, '%acum'=cumptab)  
  Total <- c(sum(tab), sum(ptab), sum(tab), sum(ptab))  
  tabela <- rbind(miolo, Total)  
  return(tabela)  
}
```

```
tabua(imccat)
```

	Freq	% Freq	Freq.acum	%acum
normal	99	33.1	99	33.1
sobrepeso	116	38.8	215	71.9
obesidade	84	28.1	299	100.0
Total	299	100.0	299	100.0





Já vimos a função `describe`, de `Hmisc`

```
describe(imccat)
```

imccat	: Estado nutricional		
	n	missing	distinct
	299	1	3

Value	normal	sobrepeso	obesidade
Frequency	99	116	84
Proportion	0.331	0.388	0.281

A função `tab1`, do pacote `epiDisplay` (antigo `epicalc`) apresenta a contagem de 'missings' (que `table` não inclui por padrão), as proporções e a frequência acumulada (que `describe` não apresenta), além de incluir um gráfico com os resultados da tabela.

```
tab1(imccat)
tab1(imccat, cum.percent = T)
tab1(imccat, cum.percent = T, missing = F)
```



## Introdução ao R

### 4. Análise univariada 23/31

Fúlvio Nedel  
SPB/UFSC

#### Introdução

#### Objetivos da análise

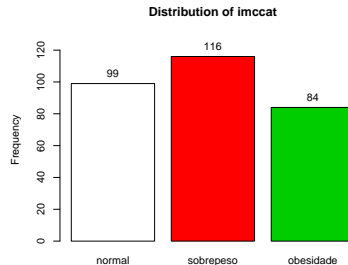
#### Análise univariada

#### Variáveis numéricas

#### Variáveis categóricas

```
tab1(imccat, missing = F)
```

imccat :			
	Frequency	%(NA+)	%(NA-)
normal	99	33.0	33.1
sobrepeso	116	38.7	38.8
obesidade	84	28.0	28.1
NA's	1	0.3	0.0
Total	300	100.0	100.0



O argumento `missing=FALSE` faz com que não sejam incluídos no gráfico.

## Introdução ao R

### 4. Análise univariada 24/31

Fúlvio Nedel  
SPB/UFSC

## Introdução

### Objetivos da análise

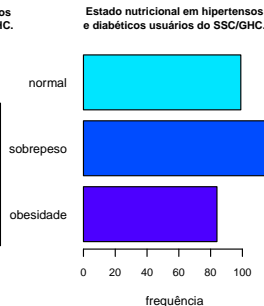
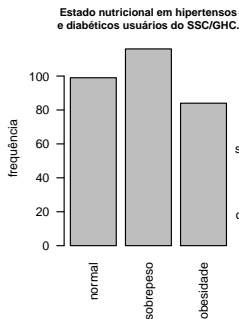
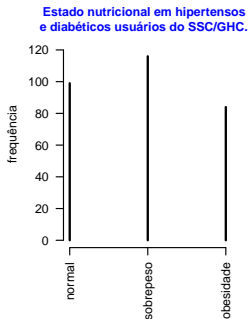
### Análise univariada

### Variáveis numéricas

### Variáveis categóricas

```

titulo <- "Estado nutricional em hipertensos
e diabéticos usuários do SSC/GHC."
par(mfrow=c(1,3))
plot(table(imccat), main=titulo, cex.main=1, adj=.6, las=2,
     ylab = 'frequência', xlab = '', col.main=4)
barplot(table(imccat), main=titulo, cex.main=.9, las=2,
         ylab = 'frequência')
par(mar=c(5,3,4,2))
barplot(rev(table(imccat)), main=titulo, cex.main=.9, cex.axis = .9,
        xlab = 'frequência', horiz = T, las=1, col=topo.colors(5))
    
```



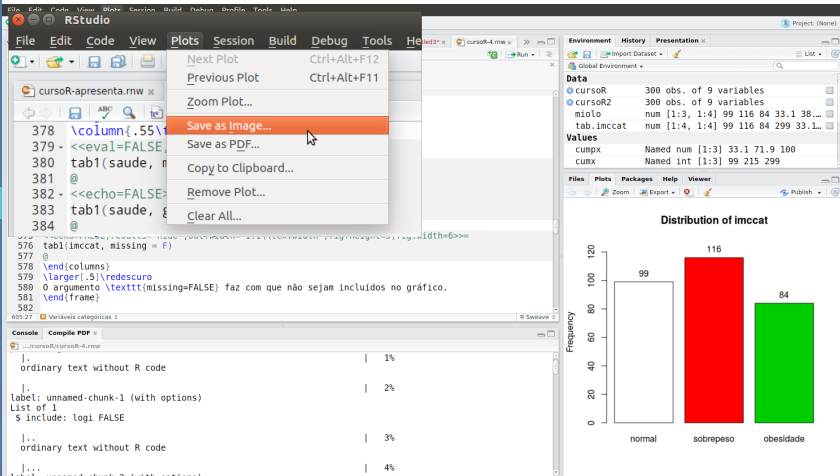
No Rstudio, gráficos podem facilmente ser salvos como figuras, PDF ou copiados para a área de transferência:

The screenshot shows the RStudio interface. The 'Plots' menu is open, and 'Save as PDF...' is highlighted. The background shows a bar chart titled 'Distribution of imccat' with three bars: 'normal' (white), 'sobrepeso' (red), and 'obesidade' (green). The y-axis is labeled 'Frequency' and ranges from 0 to 120. The x-axis labels are 'normal', 'sobrepeso', and 'obesidade'. The values above the bars are 99, 116, and 84 respectively.

**Distribution of imccat**

Categoria	Frequência
normal	99
sobrepeso	116
obesidade	84

No Rstudio, gráficos podem facilmente ser salvos como figuras, PDF ou copiados para a área de transferência:



The screenshot shows the RStudio interface. The 'Plots' menu is open, and 'Save as Image...' is highlighted. The background shows a bar chart titled 'Distribution of imccat' with three bars: 'normal' (99), 'sobrepeso' (116), and 'obesidade' (84). The y-axis is labeled 'Frequency' and ranges from 0 to 120. The x-axis labels are 'normal', 'sobrepeso', and 'obesidade'.

**Distribution of imccat**

Categoria	Frequência
normal	99
sobrepeso	116
obesidade	84

No Rstudio, gráficos podem facilmente ser salvos como figuras, PDF ou copiados para a área de transferência:

The screenshot shows the RStudio interface. The 'Plots' menu is open, displaying the following options:

- Next Plot (Ctrl+Alt+F12)
- Previous Plot (Ctrl+Alt+F11)
- Zoom Plot...
- Save as Image...** (highlighted)
- Save as PDF...
- Copy to Clipboard...
- Remove Plot...
- Clear All...

The background shows a bar chart titled 'imccat' with the following data:

Category	Frequency
normal	99
sobrepeso	116
obesidade	84

The console shows the following output:

```

| ordinary text without R code | 1%
|
| label: unnamed-chunk-1 (with options) | 2%
| List of 1 |
| $ include: logi FALSE |
|
|.. ordinary text without R code | 3%
|... | 4%

```



- Para buscar uma função que os calcule, comece com

```
??"confidence interval"
```

- Temos (no mínimo)
  - `prop.test{stats}` → aproximação normal e
  - `ci{epiDisplay}` → método exato (binomial)

Trabalhar, de momento, com a variável obeso, que é dicotômica

```
table(obeso)
```

```
obeso  
sim não  
84 215
```

```
prop.test(table(obeso))
```

1-sample proportions test with continuity correction

```
data: table(obeso), null probability 0.5  
X-squared = 56.522, df = 1, p-value = 5.558e-14  
alternative hypothesis: true p is not equal to 0.5  
95 percent confidence interval:  
0.2314582 0.3361532  
sample estimates:  
p  
0.2809365
```



Trabalhar, de momento, com a variável obeso, que é dicotômica

```
table(obeso)
```

```
obeso
```

```
sim não
```

```
84 215
```

```
prop.test(table(obeso))
```

```
1-sample proportions test with continuity correction
```

```
data: table(obeso), null probability 0.5
```

```
X-squared = 56.522, df = 1, p-value = 5.558e-14
```

```
alternative hypothesis: true p is not equal to 0.5
```

```
95 percent confidence interval:
```

```
0.2314582 0.3361532
```

```
sample estimates:
```

```
↓ p  
0.2809365
```

Trabalhar, de momento, com a variável obeso, que é dicotômica

```
table(obeso)
```

```
obeso
```

```
sim não
```

```
84 215
```

```
prop.test(table(obeso))
```

```
1-sample proportions test with continuity correction
```

```
data: table(obeso), null probability 0.5
```

```
X-squared = 56.522, df = 1, p-value = 5.558e-14
```

```
alternative hypothesis: true p is not equal to 0.5
```

```
95 percent confidence interval:
```

```
0.2314582 0.3361532
```

```
sample estimates:
```

```
↓ p  
0.2809365
```

```
ci(obeso)
```

events	total	probability	se	exact.lower95ci	exact.upper95ci
215	299	0.7190635	0.02599274	0.6644214	0.769295

### Trabalhar, de momento, com a variável obeso, que é dicotômica

```
table(obeso)
```

```
obeso
```

```
sim não  
84 215
```

```
prop.test(table(obeso))
```

```
1-sample proportions test with continuity correction
```

```
data: table(obeso), null probability 0.5
```

```
X-squared = 56.522, df = 1, p-value = 5.558e-14
```

```
alternative hypothesis: true p is not equal to 0.5
```

```
95 percent confidence interval:
```

```
0.2314582 0.3361532
```

```
sample estimates:
```

```
↓ p  
0.2809365
```

```
ci(obeso)
```

events	total	probability	se	exact.lower95ci	exact.upper95ci
215	299	0.7190635	0.02599274	0.6644214	0.769295



## Apresentar os resultados numa tabela bonita ;)

```
x <- ci(obeso)
# x[5:6]
IC95 <- matrix(c(as.numeric(rev(1-x[5:6])), as.numeric(x[5:6])),
               nrow = 2, byrow = T)
colnames(IC95) <- c('IC95.inf', 'IC95.sup')
tab.obeso <- cbind(tabua(obeso)[-3,1:2], round(IC95*100, 2))
```

```
options(OutDec = ',') # marcador decimal como vírgula
tab.obeso
```

	Freq	%	IC95.inf	IC95.sup
sim	84	28,1	23,07	33,56
não	215	71,9	66,44	76,93

## De volta a estado nutricional, que é politômica

```
tabua(imccat)[-3]
```

	Freq	%	%acum
normal	99	33,1	33,1
sobrepeso	116	38,8	71,9
obesidade	84	28,1	100,0
Total	299	100,0	100,0

O que se faz muitas vezes é calcular o IC da probabilidade da categoria contra seu complemento, i.e., todas as outras categorias em conjunto.

```
icbin <- rbind( icex <- ci(imccat=='normal'),
               icmb <- ci(imccat=='sobrepeso'),
               icboa <- ci(imccat=='obesidade') )
```

```
cbind(tabua(imccat)[-4,-c(3:4)], round(icbin[5:6]*100,2))
```

	Freq	%	exact.lower95ci	exact.upper95ci
normal	99	33,1	27,80	38,76
sobrepeso	116	38,8	33,24	44,58
obesidade	84	28,1	23,07	33,56

Entretanto, isso não é muito adequado, porque a distribuição de probabilidades da categoria de uma variável politômica segue uma distribuição multinomial, que é um pouco mais complexa que a binomial.

```
library(MultinomialCI)
round(multinomialCI(table(imccat), .05)*100, 2)

      [,1] [,2]
[1,] 27,09 39,46
[2,] 32,78 45,14
[3,] 22,07 34,44
```

Entretanto, isso não é muito adequado, porque a distribuição de probabilidades da categoria de uma variável politômica segue uma distribuição multinomial, que é um pouco mais complexa que a

**Claro que a tabela pode ser melhorada:**

```
icmulti <- round(multinomialCI(table(imccat), .05)*100, 2)
rownames(icmulti) <- rownames(tab.imccat[-4,])
colnames(icmulti) <- c('IC95.inf', 'IC95.sup')
icmulti
```

	IC95.inf	IC95.sup
normal	27,09	39,46
sobrepeso	32,78	45,14
obesidade	22,07	34,44



### Introdução ao R

### 4. Análise univariada 31/31

Fúlvio Nedel  
SPB/UFSC

	%	IC normal	IC binomial	IC multinomial
normal	33,1	27,86 - 38,8	27,8 - 38,76	27,09 - 39,46
sobrepeso	38,8	33,29 - 44,6	33,24 - 44,58	32,78 - 45,14
obesidade	28,1	23,15 - 33,62	23,07 - 33,56	22,07 - 34,44

