

# Distribuições de probabilidade, hipótese nula e valor p

*Paulo S. P Silveira (paulo.silveira@fm.usp.br)*

*Koichi Sameshima (koichi.sameshima@fm.usp.br)*

*José O. Siqueira (jose.siqueira@fm.usp.br)*

## Contents

<b>Objetivos desta aula</b>	<b>2</b>
Binomial . . . . .	4
Poisson . . . . .	4
Normal . . . . .	4
<b>Continuando a deformação...</b>	<b>4</b>
<b>Exemplo hipotético</b>	<b>5</b>
Gráfico (contagens) . . . . .	5
No estilo <i>histogram-like (vertical lines)</i> . . . . .	6
Para manter as bolinhas . . . . .	7
Colorir . . . . .	8
Cores amigáveis para daltônicos: friendlydemo.R . . . . .	9
Distribuição de probabilidades . . . . .	10
nomes das colunas . . . . .	11
criando coluna com as porcentagens . . . . .	12
Características de uma distribuição de probabilidades . . . . .	12
Gráfico em porcentagem . . . . .	13
Corrigindo os eixos . . . . .	14
Density plots . . . . .	15
Dando nome ao gráfico . . . . .	16
Juntando tudo no Rscript gestantes.R . . . . .	17
<b>Novamente, a incerteza</b>	<b>20</b>
Moeda é um exemplo em saúde? . . . . .	20
<b>Distribuição binomial</b>	<b>21</b>
Distribuição binomial: 1 jogada . . . . .	21
Distribuição binomial: 2 jogadas . . . . .	22
Distribuição binomial: 3 jogadas . . . . .	23
Distribuição binomial: 5 jogadas . . . . .	24
Distribuição binomial: 15 jogadas . . . . .	26
(alterando a escala) . . . . .	28
Distribuição binomial: 15 jogadas, moeda desbalanceada . . . . .	28
Cauda = 1 . . . . .	29
Cauda = 2 . . . . .	32
Cauda = 3 . . . . .	33
Cauda = 4 . . . . .	34
Cauda = 5 . . . . .	35
Cauda = 6 . . . . .	36
Voltando para cauda = 4 . . . . .	37

Simulação 1 com Goodcoin.R . . . . .	38
Binomial adaptada a um tratamento . . . . .	39
Hipótese nula e alternativa . . . . .	43
Experimento único . . . . .	44
valor-p . . . . .	44
alfa ( $\alpha$ ) . . . . .	44
Simulação 2 com Goodcoin.R . . . . .	46
Tomada de decisão: $\alpha$ e $\beta$ . . . . .	47
poder ( $1 - \beta$ ) . . . . .	47
e o que acontece na prática? . . . . .	48
Conclusão: ESTE ESTUDO É INCONCLUSIVO. . . . .	48
O que fazer para reduzir $\beta$ ? . . . . .	48
Estratégia 1: aumentar o valor de $\alpha$ . . . . .	49
Estratégia 2: tornar as distribuições mais estreitas . . . . .	50
Estratégia 3: ser capaz de detectar, somente, maiores efeitos . . . . .	52
<b>Distribuição de Poisson</b>	<b>53</b>
exemplo . . . . .	54
<b>Distribuição normal</b>	<b>57</b>
aparência . . . . .	57
simetria . . . . .	57
áreas sob a curva . . . . .	58
$\pm 1dp$ . . . . .	58
$\pm 2dp$ . . . . .	58
$\pm 3dp$ . . . . .	59
variando média e desvio-padrão . . . . .	59
Distribuição normal padronizada . . . . .	65
Criando distribuições normais em R . . . . .	67
<b>TCL e EPM</b>	<b>71</b>
Amostragem ( <i>sampling</i> ) . . . . .	73
distribuição das médias amostrais e EPM . . . . .	76
EPM na simulação com 3000 amostras . . . . .	78
Reamostragem ( <i>bootstrapping</i> ), saindo da fantasia . . . . .	78

## Objetivos desta aula

Ao final desta aula o aluno deve ser capaz de:

- definir distribuição de probabilidades;
- definir e aplicar a distribuição Binomial;
- definir hipóteses estatísticas (hipótese nula e alternativa);
- diferenciar erros do tipo I e II (alfa e beta);
- interpretar o valor-p;

### **Binomial ( $p[\text{sucesso}] = 0.3$ )**

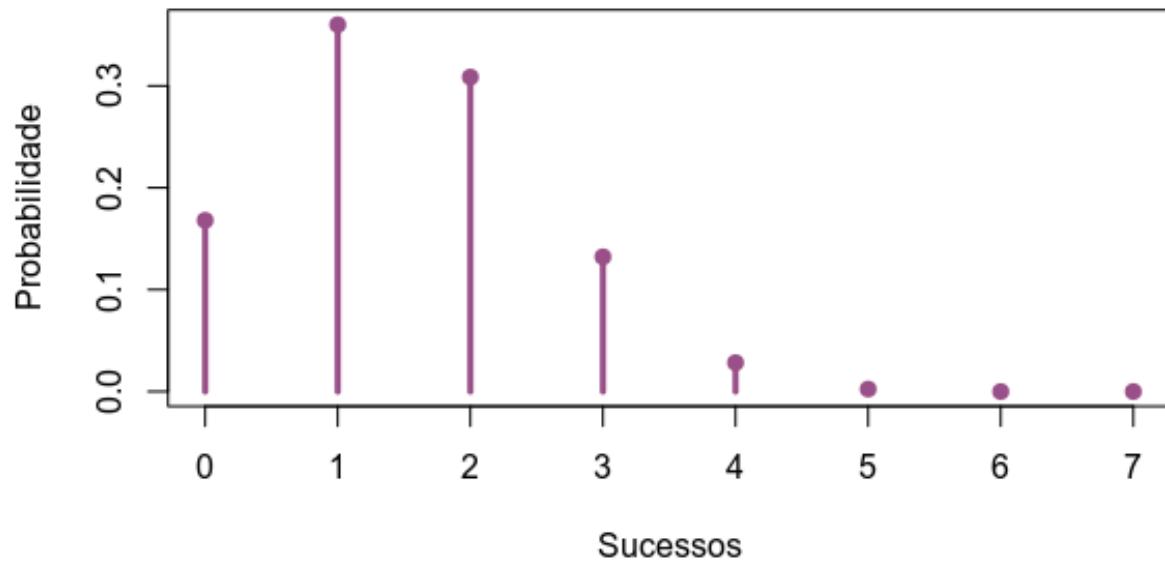


Figure 1:

### **Poisson ( $\lambda=2$ )**

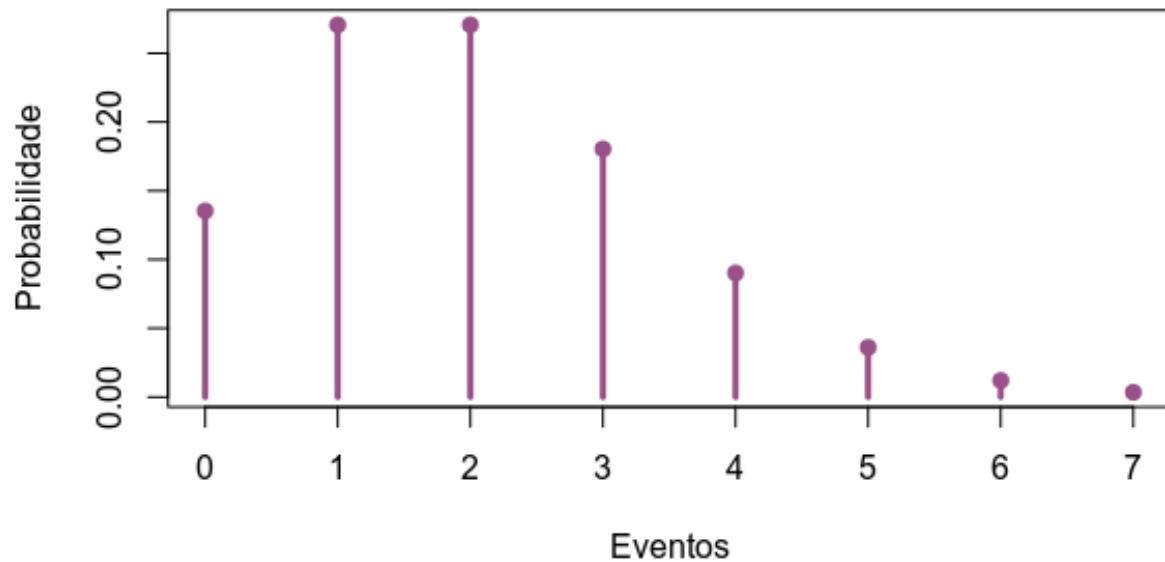


Figure 2:

**Normal (100, 20)**

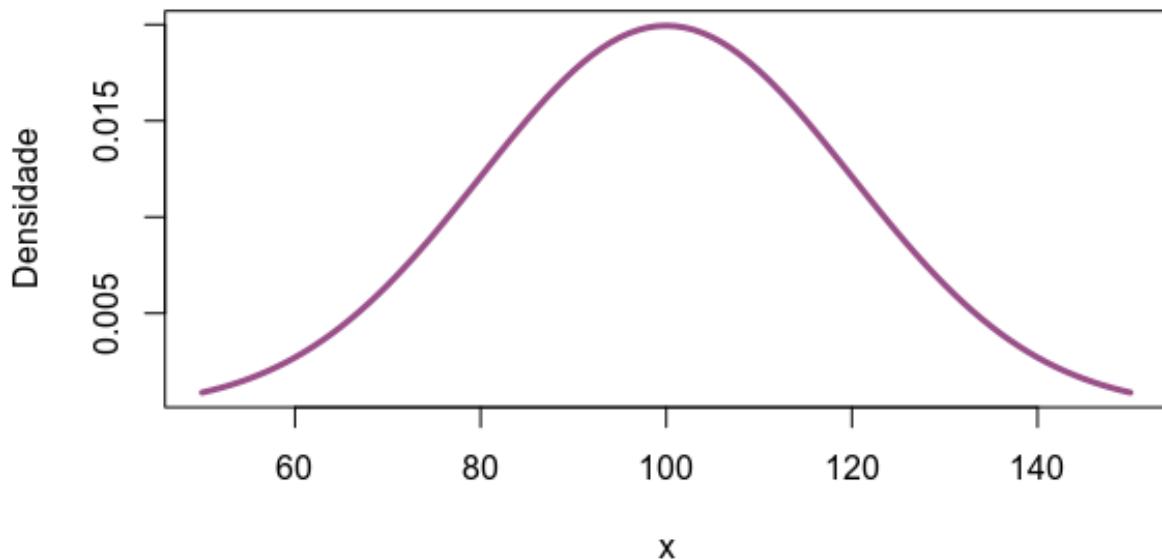


Figure 3:

Binomial

Poisson

Normal

Continuando a deformação...



## Exemplo hipotético

Suponha que foi feito um levantamento sobre o número de drogas em uso por gestantes .

```
DrgGrv <- read.table("Drogas_Gravidez.txt", header=TRUE, sep="\t")
print(DrgGrv)
```

```
##   Drogas Pacientes
## 1      0    1425
## 2      1    1351
## 3      2     793
## 4      3     348
## 5      4     156
## 6      5      58
## 7      6      28
## 8      7      15
## 9      8       6
## 10     9       3
## 11    10       1
## 12    11       0
## 13    12       1
```

O arquivo *Drogas\_Gravidez.txt* tem duas colunas:

- a primeira linha tem os nomes das variáveis (header=TRUE),
- as colunas estão separadas por *tab* (sep="\t")

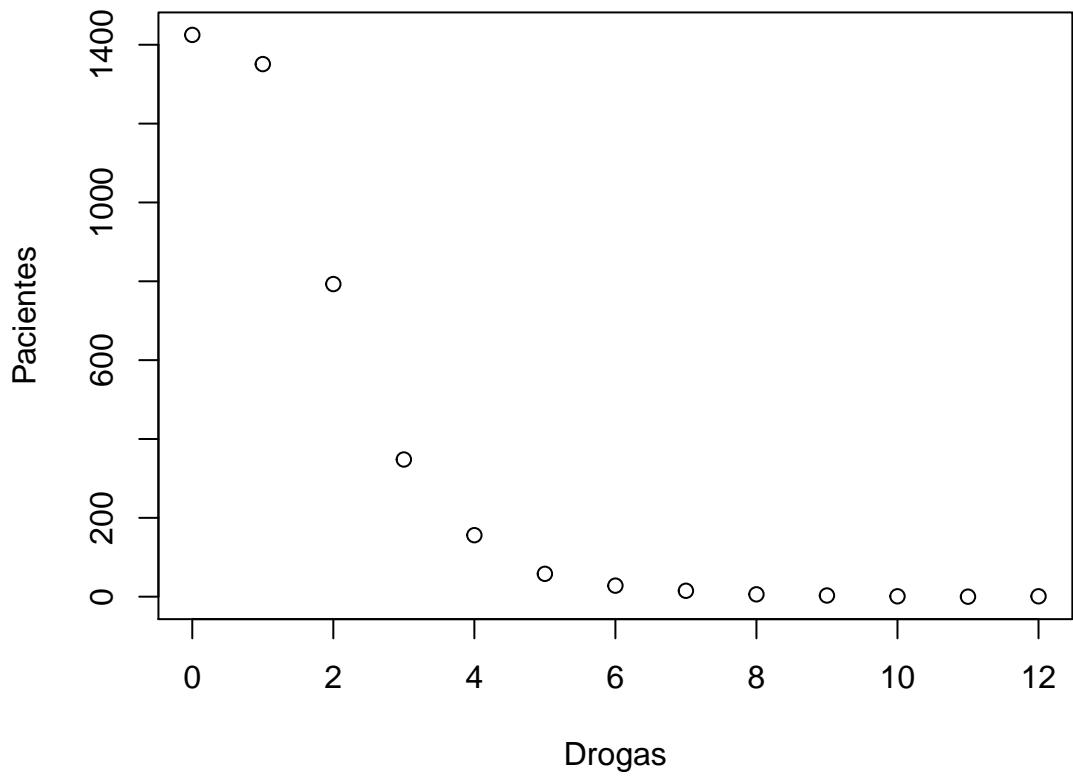
A variável **DrgGrv** é um *data frame*:

```
is.data.frame(DrgGrv)
```

```
## [1] TRUE
```

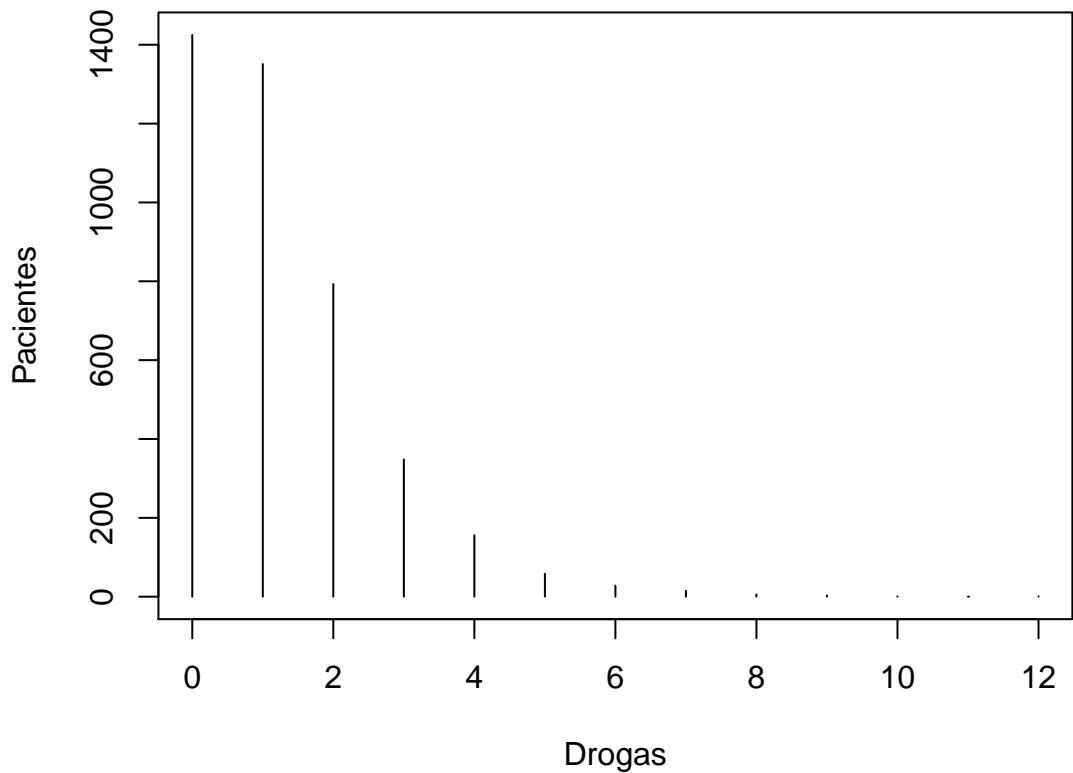
## Gráfico (contagens)

```
plot (DrgGrv)
```



No estilo *histogram-like (vertical lines)*

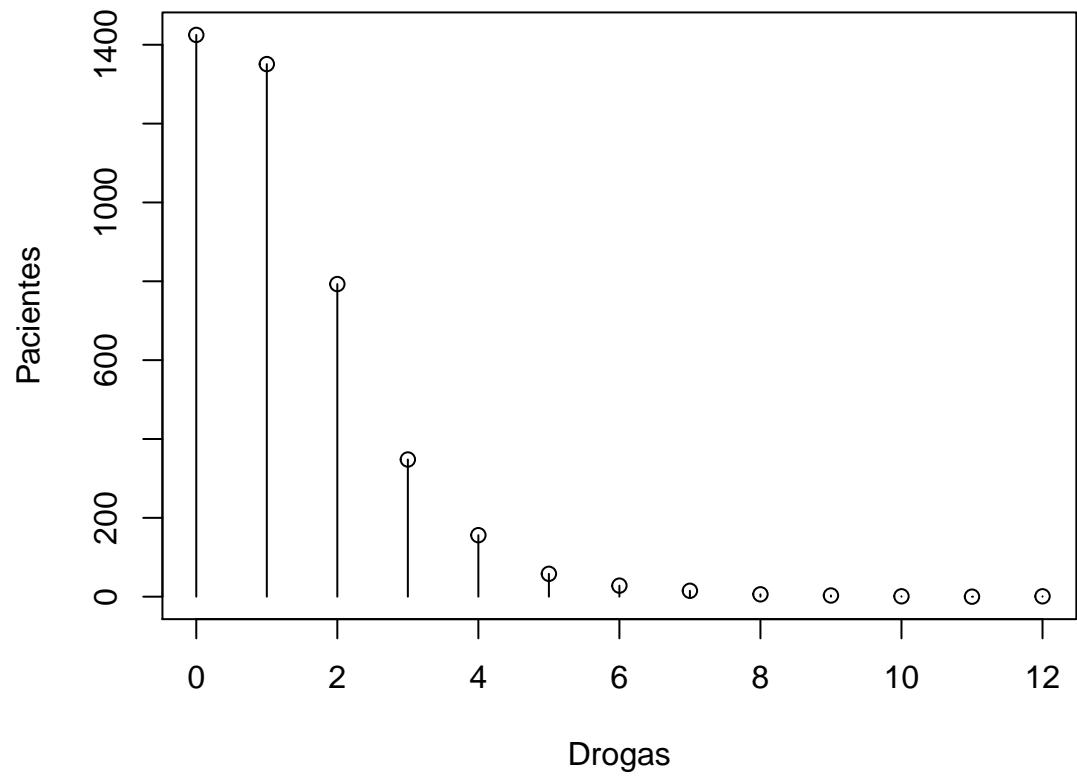
```
plot (DrgGrv,type="h")
```



Como foi que eu descobri isto? Quick-R: Line charts

Para manter as bolinhas

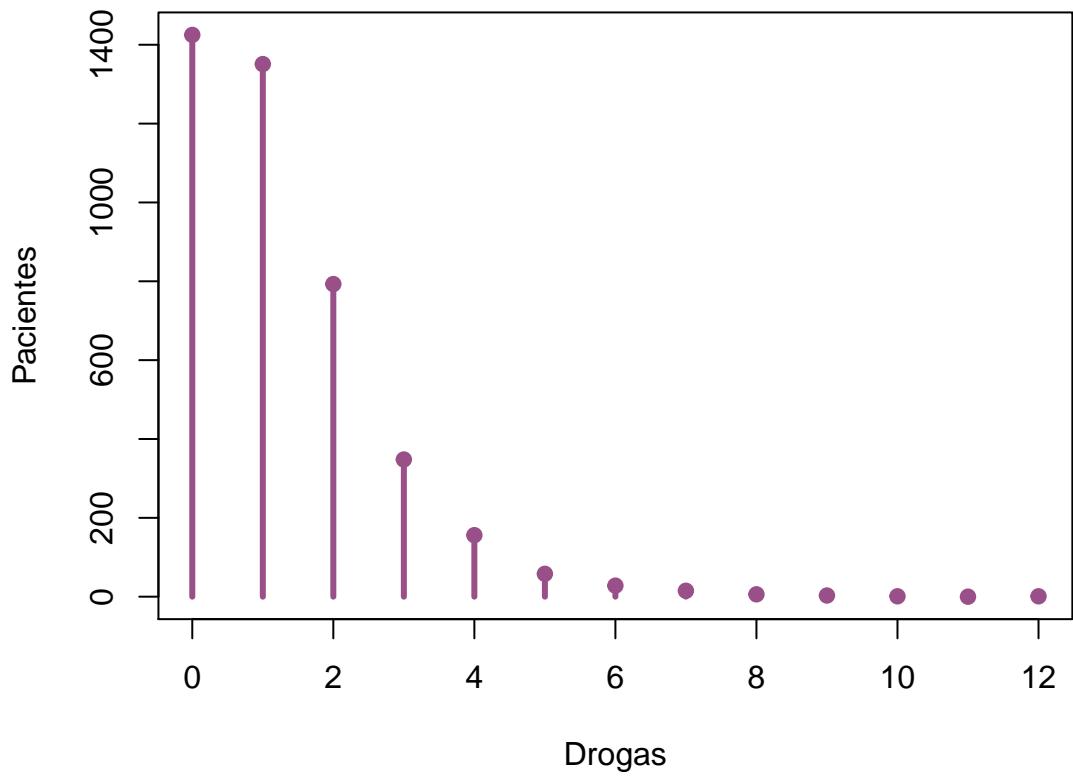
```
plot (DrgGrv,type="h")
points(DrgGrv)
```



### Colorir

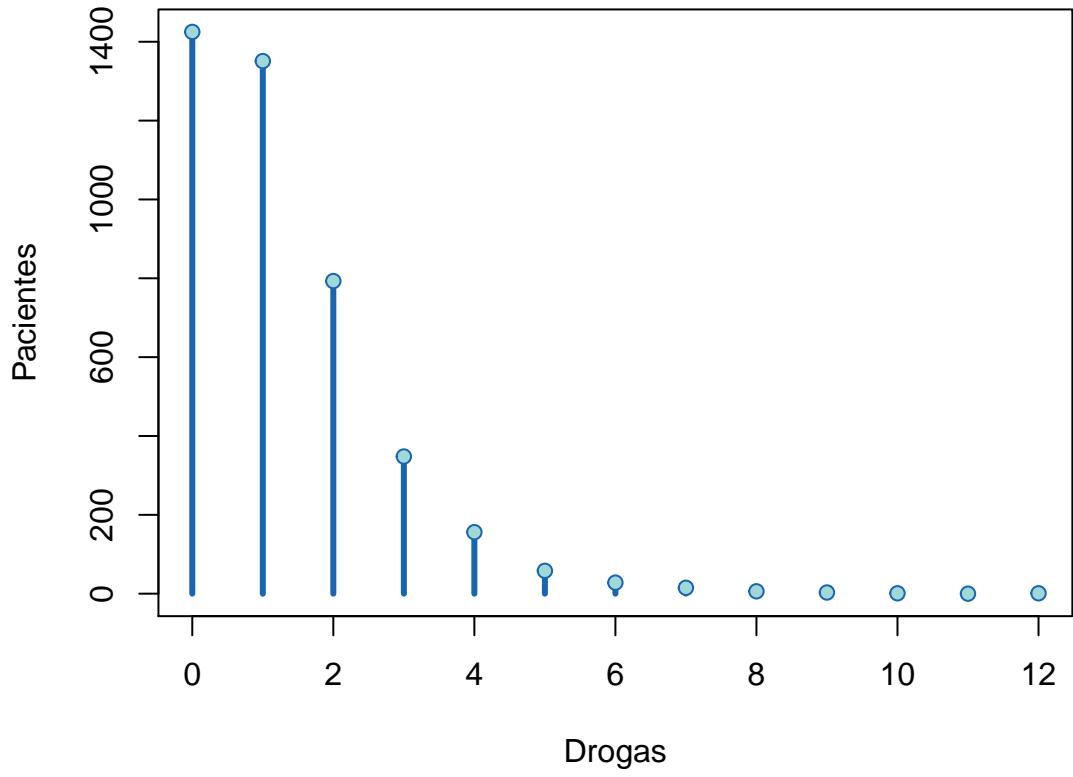
Encontrei mais em Quick-R: Graphical Parameters

```
plot (DrgGrv,type="h", col="#994F88", lwd=3)
points(DrgGrv, pch=21, col="#994F88", bg="#994F88")
```



Cores amigáveis para daltônicos:

```
source("friendlycolor.R")
plot (DrgGrv,type="h", col=friendlycolor(8), lwd=3)
points(DrgGrv, pch=21, col=friendlycolor(8), bg=friendlycolor(12))
```



### friendlydemo.R

Exibindo todas as cores da função friendlycolor():

```
# friendlydemo.R
# exibe as 46 cores disponíveis em friendlycolor.R
source("friendlycolor.R")
indice <- 1 # primeiro indice
# plota um gráfico vazio
plot(NA, xlim=c(0,9), ylim=c(0,7),
     xlab=NA, ylab=NA, axes = FALSE)
for (x in 1:8)
{
  for (y in 1:6)
  {
    # pula se acabaram as cores
    if (indice > 46) {next}
    points ( x, 7-y,
              pch=21, cex=5,
              col="black",
              bg=friendlycolor(indice)
    )
  }
}
```

```

cortexto <- "black"

if (
  (indice >= 25 & indice <= 27)
  |
  (indice >= 31 & indice <= 36)
)
{
  cortexto <- "white"
}
text(x, 7-y, indice, col=cortexto)
# incrementa (proxima cor)
indice <- indice+1
}
}

```



## Distribuição de probabilidades

```

DrgGrv <- read.table("Drogas_Gravidez.txt", header=TRUE, sep="\t")
print(DrgGrv)

##      Drogas Pacientes

```

```

## 1      0    1425
## 2      1    1351
## 3      2     793
## 4      3     348
## 5      4     156
## 6      5      58
## 7      6      28
## 8      7      15
## 9      8       6
## 10     9       3
## 11    10      1
## 12    11      0
## 13    12      1

```

nomes das colunas

```

names(DrgGrv)
## [1] "Drogas"      "Pacientes"

```

criando coluna com as porcentagens

```

DrgGrv$Porcentagem <- round(DrgGrv$Pacientes/sum(DrgGrv$Pacientes)*100,2)
names(DrgGrv)

```

```

## [1] "Drogas"      "Pacientes"    "Porcentagem"
print(DrgGrv)

```

	Drogas	Pacientes	Porcentagem
## 1	0	1425	34.05
## 2	1	1351	32.28
## 3	2	793	18.95
## 4	3	348	8.32
## 5	4	156	3.73
## 6	5	58	1.39
## 7	6	28	0.67
## 8	7	15	0.36
## 9	8	6	0.14
## 10	9	3	0.07
## 11	10	1	0.02
## 12	11	0	0.00
## 13	12	1	0.02

Características de uma distribuição de probabilidades

- a coluna “Porcentagem” soma 1 ou 100%

```

sum(DrgGrv$Porcentagem)

```

```

## [1] 100


- todos os valores são positivos

```

```

DrgGrv$Porcentagem < 0

## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12] FALSE FALSE
sum(DrgGrv$Porcentagem < 0)

## [1] 0
• todos os valores ficam entre 0 e 1 (ou entre 0% e 100%)
DrgGrv$Porcentagem >= 0 & DrgGrv$Porcentagem <= 100

## [1] TRUE TRUE
sum (DrgGrv$Porcentagem >= 0 & DrgGrv$Porcentagem <= 100)

## [1] 13

```

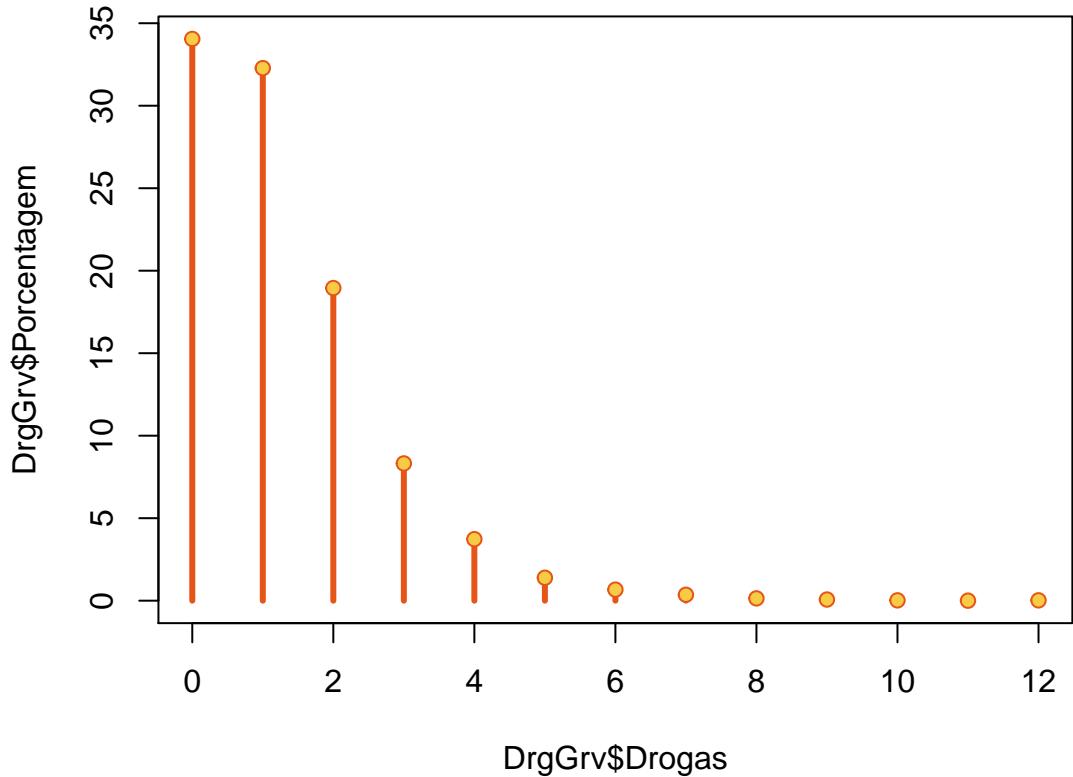
## Gráfico em porcentagem

Como *DrgGrv* agora tem três colunas, precisamos indicar quais colunas quero usar como *x* (**DrgGrv\$Drogas**) e *y* (**DrgGrv\$Porcentagem**)

```

source("friendlycolor.R")
plot (DrgGrv$Drogas, DrgGrv$Porcentagem,
      type="h", col=friendlycolor(20), lwd=3)
points(DrgGrv$Drogas, DrgGrv$Porcentagem,
       pch=21,
       col=friendlycolor(20), bg=friendlycolor(23))

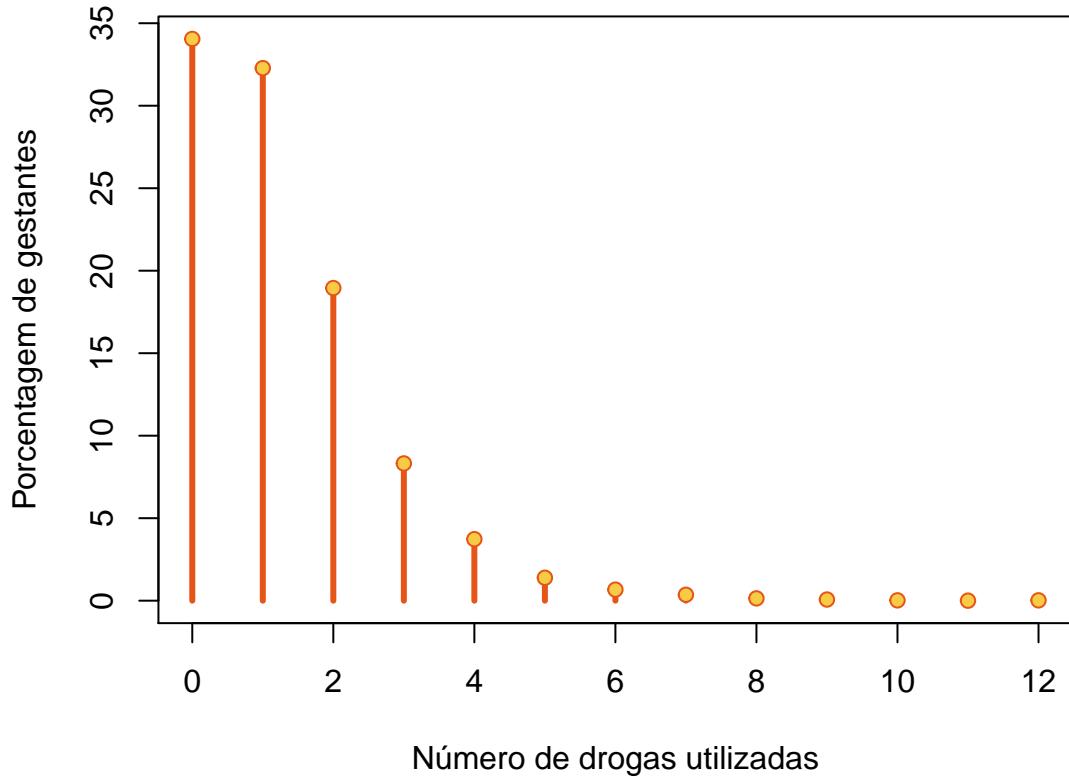
```



### Corrigindo os eixos

... consequentemente, os nomes das variáveis aparecem nos eixos. Isto é corrigido assim:

```
source("friendlycolor.R")
plot (DrgGrv$Drogas, DrgGrv$Porcentagem,
      type="h", col=friendlycolor(20), lwd=3,
      xlab="Número de drogas utilizadas",
      ylab="Porcentagem de gestantes")
points(DrgGrv$Drogas, DrgGrv$Porcentagem,
       pch=21,
       col=friendlycolor(20), bg=friendlycolor(23))
```



### Density plots

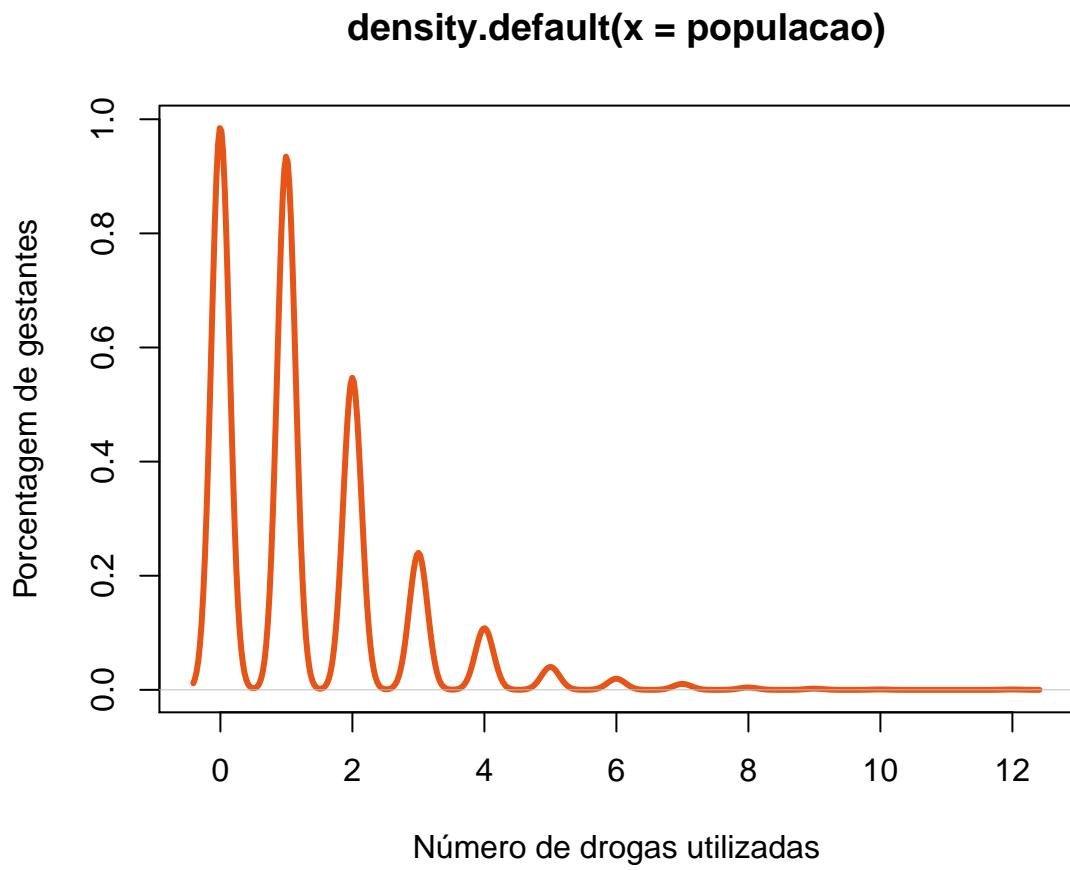
O número de drogas utilizadas é uma variável quantitativa discreta.

Apenas para demonstrar (não é a forma apropriada), caso a variável fosse quantitativa contínua, poderíamos usar um *density plot*

Primeiro criamos uma variável para conter a densidade de probabilidades

```
# desdobra a quantidade de pacientes
populacao <- c() # cria um vetor vazio
for (r in 1:nrow(DrgGrv))
{
  # acumula no vetor o número de drogas
  populacao <- c(populacao, rep(DrgGrv$Drogas, times=DrgGrv$Pacientes))
}
# usa a função R que transforma em densidade de probabilidade
densidade <- density(populacao)

source("friendlycolor.R")
plot (densidade,
       col=friendlycolor(20), lwd=3,
       xlab="Número de drogas utilizadas",
       ylab="Porcentagem de gestantes")
```

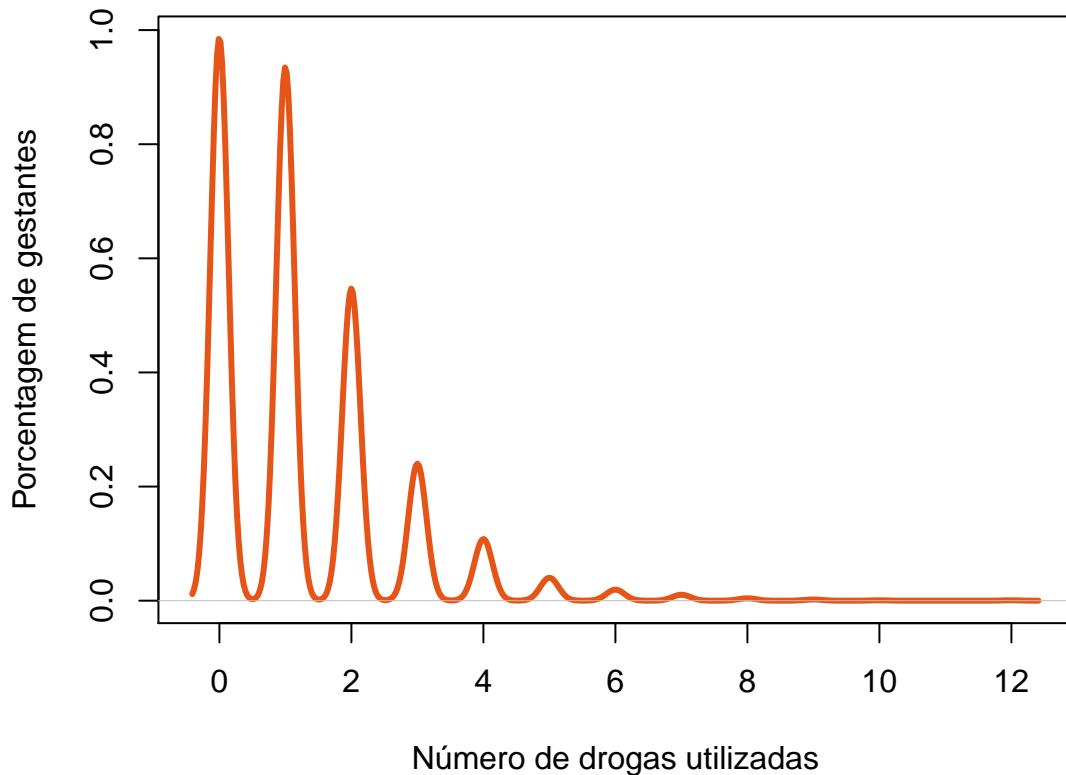


### Dando nome ao gráfico

A função `plot()` tem muitos parâmetros; aqui usamos o parâmetro `main`.

```
source("friendlycolor.R")
# note o título em duas linhas usando-se \\n
plot (densidade,
      main = "Distribuição do uso de\\ndrogas em gestantes",
      col=friendlycolor(20), lwd=3,
      xlab="Número de drogas utilizadas",
      ylab="Porcentagem de gestantes")
```

## Distribuição do uso de drogas em gestantes



### Juntando tudo no Rscript gestantes.R

Tudo o que foi feito até agora pode ser colocado em um único *Rscript*

```
# gestantes.R
#   le os dados e cria os graficos

source("friendlycolor.R")

# le os dados
DrgGrv <- read.table("Drogas_Gravidez.txt",
                      header=TRUE, sep="\t")

# cria coluna para a porcentagem
DrgGrv$Porcentagem <- round(DrgGrv$Pacientes/
                               sum(DrgGrv$Pacientes)*100,2)

# exibe a tabela de dados
cat("Utilização de drogas em gestantes\n")
print(DrgGrv)

# exibe o grafico no estilo 'histograma'
```

```

# (variavel quantitativa discreta)
plot (DrgGrv$Drogas, DrgGrv$Porcentagem,
      main = "Distribuição do uso de\ndrogas em gestantes",
      type="h", col=friendlycolor(20), lwd=3,
      xlab="Número de drogas utilizadas",
      ylab="Porcentagem de gestantes")
points(DrgGrv$Drogas, DrgGrv$Porcentagem,
       pch=21,
       col=friendlycolor(20), bg=friendlycolor(23))

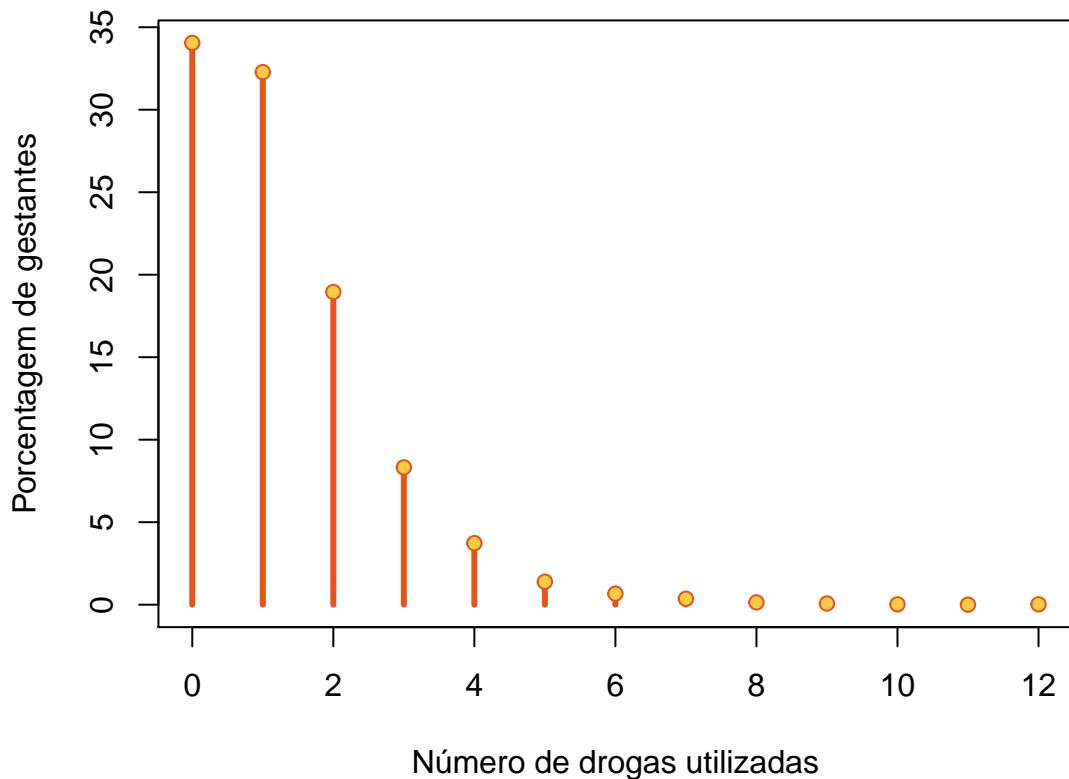
# tratando a variavel como quantitativa continua
# (apenas por exemplo; nao eh apropriado fazer isto)
# cria uma populacao com as quantidades
# de drogas utilizadas pelas pacientes
populacao <- c() # cria um vetor vazio
for (r in 1:nrow(DrgGrv))
{
  # acumula no vetor o numero de drogas
  populacao <- c(populacao,
                  rep(DrgGrv$Drogas, times=DrgGrv$Pacientes))
}
# usa a funcao R que transforma
# em densidade de probabilidade
densidade <- density(populacao)
# exibe o grafico no estilo 'density plot'
plot (densidade,
      main = "Distribuição do uso de\ndrogas em gestantes",
      col=friendlycolor(20), lwd=3,
      xlab="Número de drogas utilizadas",
      ylab="Porcentagem de gestantes")

## Utilização de drogas em gestantes

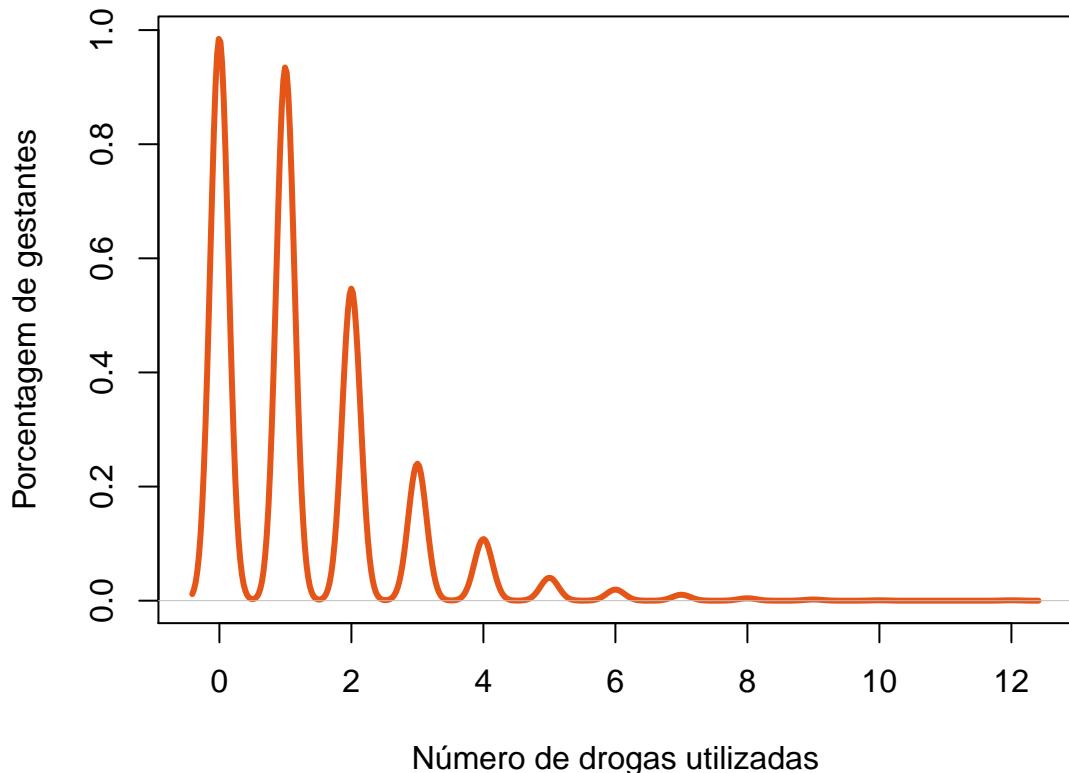
##    Drogas Pacientes Porcentagem
## 1        0     1425      34.05
## 2        1     1351      32.28
## 3        2      793      18.95
## 4        3      348       8.32
## 5        4      156       3.73
## 6        5       58       1.39
## 7        6       28       0.67
## 8        7       15       0.36
## 9        8       6       0.14
## 10       9       3       0.07
## 11      10      1       0.02
## 12      11      0       0.00
## 13      12      1       0.02

```

## Distribuição do uso de drogas em gestantes



## Distribuição do uso de drogas em gestantes



Novamente, a incerteza



Moeda é um exemplo em saúde?

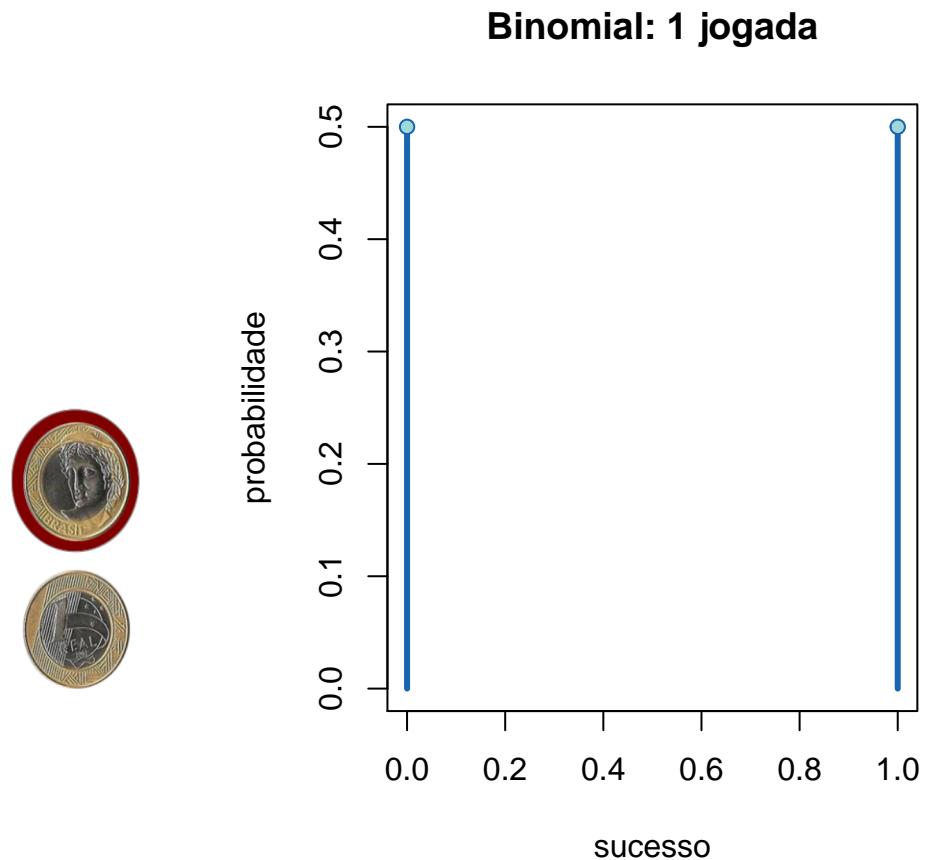


Coroa  
(head)      Cara  
(tail)

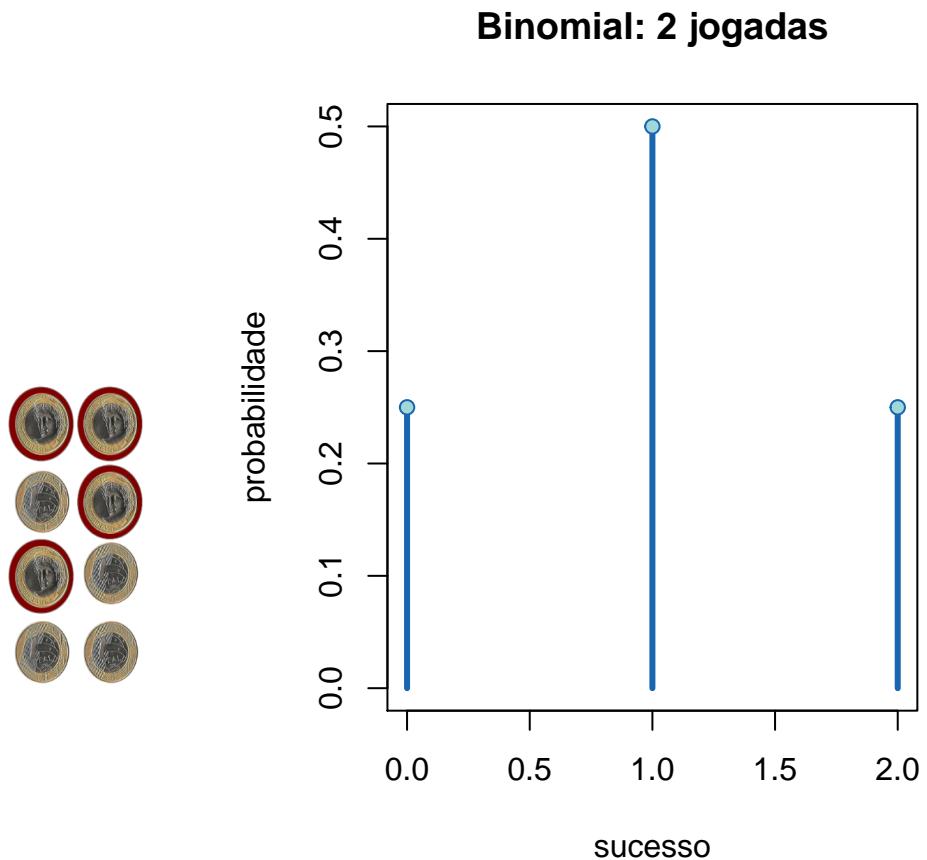
## Distribuição binomial

```
## Registered S3 methods overwritten by 'ggplot2':  
##   method      from  
##   [.quosures    rlang  
##   c.quosures    rlang  
##   print.quosures rlang
```

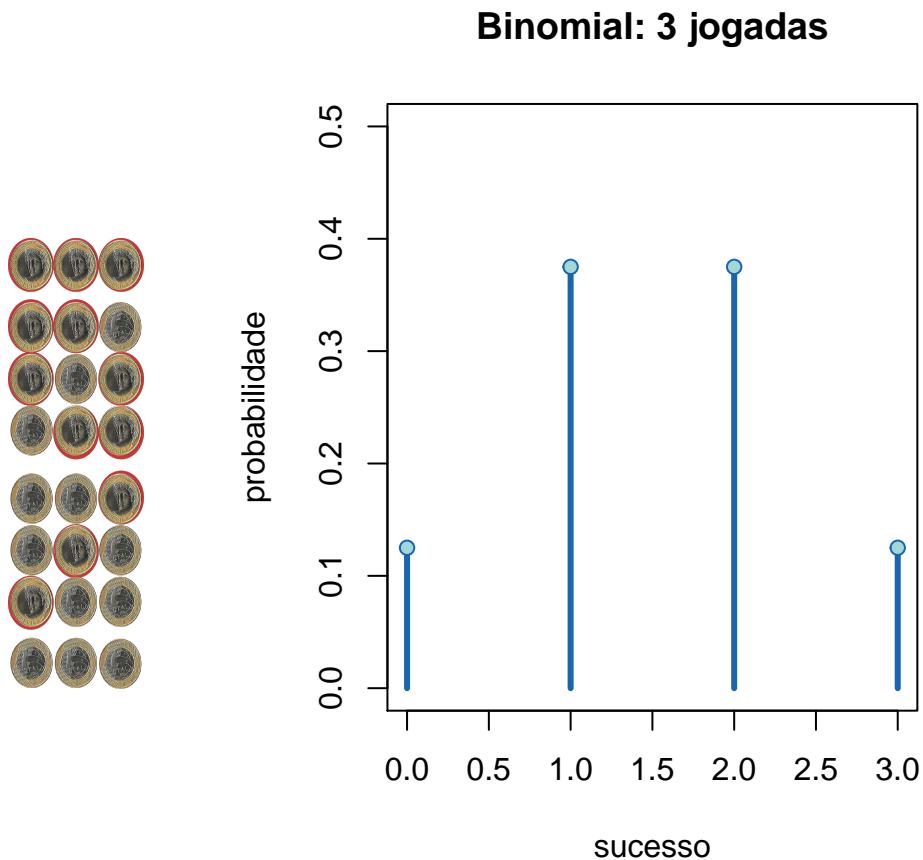
Distribuição binomial: 1 jogada



Distribuição binomial: 2 jogadas



## Distribuição binomial: 3 jogadas



A função R é **dbinom(x, size, prob)**, indicando, respectivamente, quantas jogadas, o total de jogadas e a probabilidade de sucesso de uma jogada.

Para uma moeda balanceada (prob=0.5), a probabilidade de 0 sucesso ( $x=0$ ) em 3 jogadas ( $size=3$ ) é:

```
dbinom(0, 3, 0.5)
```

```
## [1] 0.125
```

1 sucesso em 3 jogadas:

```
dbinom(1, 3, 0.5)
```

```
## [1] 0.375
```

2 sucesso em 3 jogadas:

```
dbinom(2, 3, 0.5)
```

```
## [1] 0.375
```

3 sucesso em 3 jogadas:

```
dbinom(3,3,0.5)  
## [1] 0.125
```

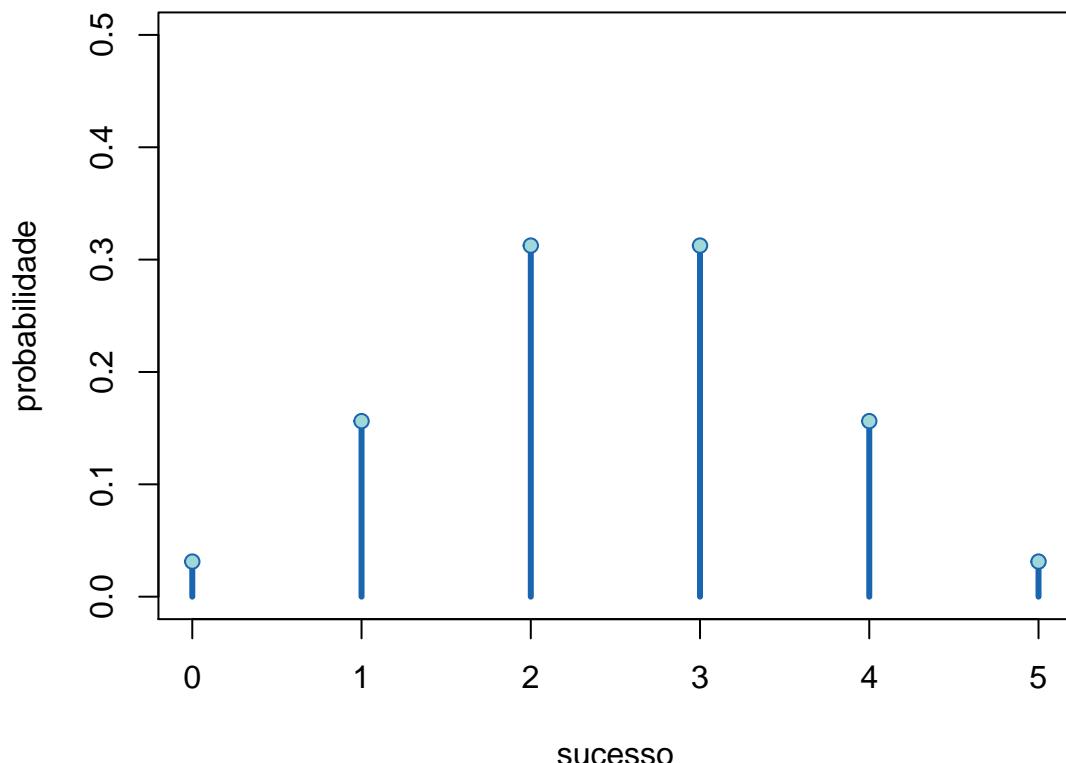
---

## Distribuição binomial: 5 jogadas

Os gráficos foram produzidos utilizando o seguinte código:

```
source("friendlycolor.R")  
jogadas <- 5  
sucesso <- 0:jogadas  
probabilidade <- dbinom(sucesso,jogadas,0.5)  
plot(sucesso, probabilidade,  
      main = paste("Binomial: ",  
                  jogadas, " jogadas", sep=""),  
      ylim = c(0,0.5),  
      type="h",  
      col=friendlycolor(8), lwd=3)  
points(sucesso,probabilidade, pch=21,  
       col=friendlycolor(8),  
       bg=friendlycolor(12))
```

## Binomial: 5 jogadas



É possível ver todos os valores em uma tabela:

```
jogadas <- 5
sucesso <- 0:jogadas
probabilidade <- dbinom(sucesso,jogadas,0.5)
cat ("Sucesso\tProbabilidade\n")
for (i in 1:(jogadas+1))
{
  cat (sucesso[i],"\t",probabilidade[i],"\n")
}

## Sucesso  Probabilidade
## 0      0.03125
## 1      0.15625
## 2      0.3125
## 3      0.3125
## 4      0.15625
## 5      0.03125
```

... ou, mais facilmente ainda, criando um *data frame*:

```
jogadas <- 5
sucesso <- 0:jogadas
probabilidade <- dbinom(sucesso,jogadas,0.5)
binomial <- data.frame(sucesso,probabilidade)
print(binomial)

##   sucesso probabilidade
## 1      0      0.03125
## 2      1      0.15625
## 3      2      0.31250
## 4      3      0.31250
## 5      4      0.15625
## 6      5      0.03125
```

... as colunas do *data frame* podem ser renomeadas:

```
names(binomial) <- c("Sucesso","FR")
print(binomial)
```

```
##   Sucesso     FR
## 1      0 0.03125
## 2      1 0.15625
## 3      2 0.31250
## 4      3 0.31250
## 5      4 0.15625
## 6      5 0.03125
```

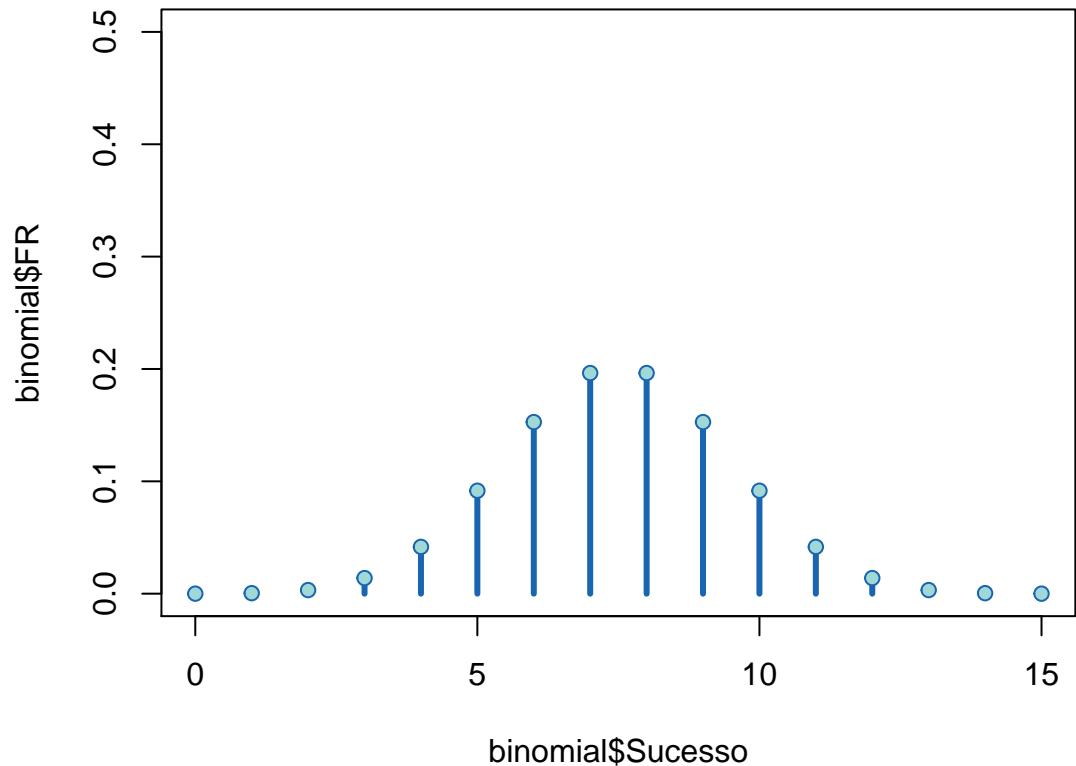
---

## Distribuição binomial: 15 jogadas

Este código foi modificado para usar um *data frame*.

```
source("friendlycolor.R")
jogadas <- 15
sucesso <- 0:jogadas
probabilidade <- dbinom(sucesso,jogadas,0.5)
binomial <- data.frame(sucesso,probabilidade)
names(binomial) <- c("Sucesso","FR")
plot(binomial$Sucesso,
      binomial$FR,
      main = paste("Binomial: ",
                   jogadas, " jogadas", sep=""),
      ylim = c(0,0.5),
      type="h",
      col=friendlycolor(8), lwd=3)
points(binomial$Sucesso,
       binomial$FR,
       pch=21,
       col=friendlycolor(8),
       bg=friendlycolor(12))
```

## Binomial: 15 jogadas



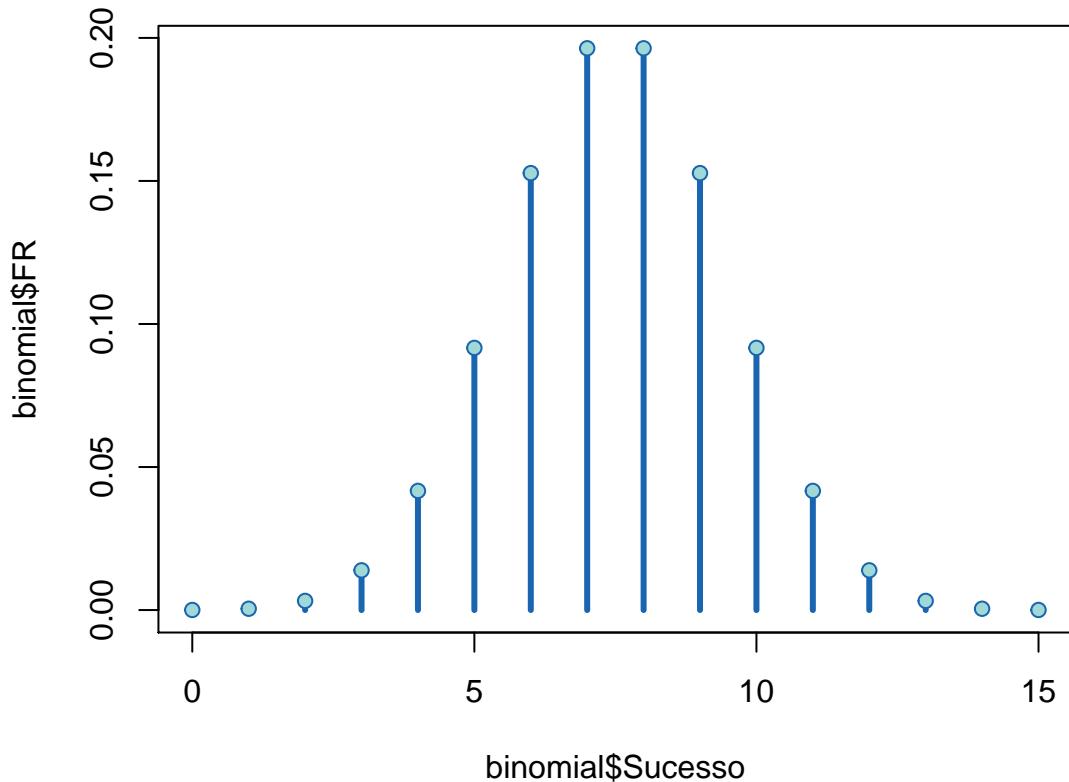
Observe que a soma de todas as colunas é igual a 1. Então, quanto mais jogadas, maior a dispersão e menor a altura das distribuições.



(alterando a escala)

```
plot(binomial$Sucesso,
      binomial$FR,
      main = paste("Binomial: ",jogadas, " jogadas", sep=""),
      ylim = c(0,max(binomial$FR)),
      type="h",
      col=friendlycolor(8), lwd=3)
points(binomial$Sucesso,
       binomial$FR,
       pch=21,
       col=friendlycolor(8),
       bg=friendlycolor(12))
```

Binomial: 15 jogadas



Distribuição binomial: 15 jogadas, moeda desbalanceada

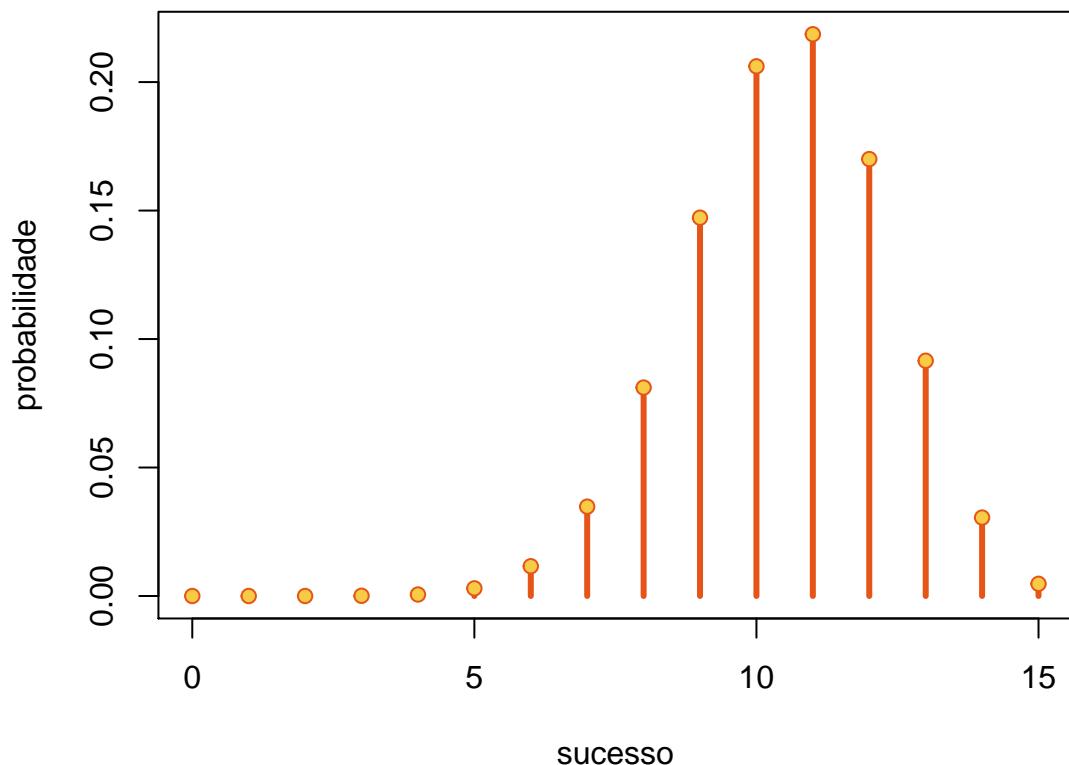
```
source("friendlycolor.R")
p.sucesso <- 0.7 # *** probabilidade de sucesso ***
jogadas <- 15
sucesso <- 0:jogadas
probabilidade <- dbinom(sucesso,jogadas,p.sucesso)
```

```

plot(sucesso, probabilidade,
      main = paste("Binomial: ",
                   jogadas, " jogadas, ",
                   "P[s] = ", p.sucesso,
                   sep=""),
      ylim = c(0,max(probabilidade)),
      type="h",
      col=friendlycolor(20), lwd=3)
points(sucesso,probabilidade, pch=21,
       col=friendlycolor(20),
       bg=friendlycolor(23))

```

**Binomial: 15 jogadas,  $P[s] = 0.7$**



Cauda = 1

```

source("friendlycolor.R")
p.sucesso <- 0.5 # *** probabilidade de sucesso ***
jogadas <- 15
sucesso <- 0:jogadas
probabilidade <- dbinom(sucesso,jogadas,p.sucesso)
# criando um data frame
binomial <- data.frame(sucesso,probabilidade)
names(binomial) <- c("Sucesso", "FR")

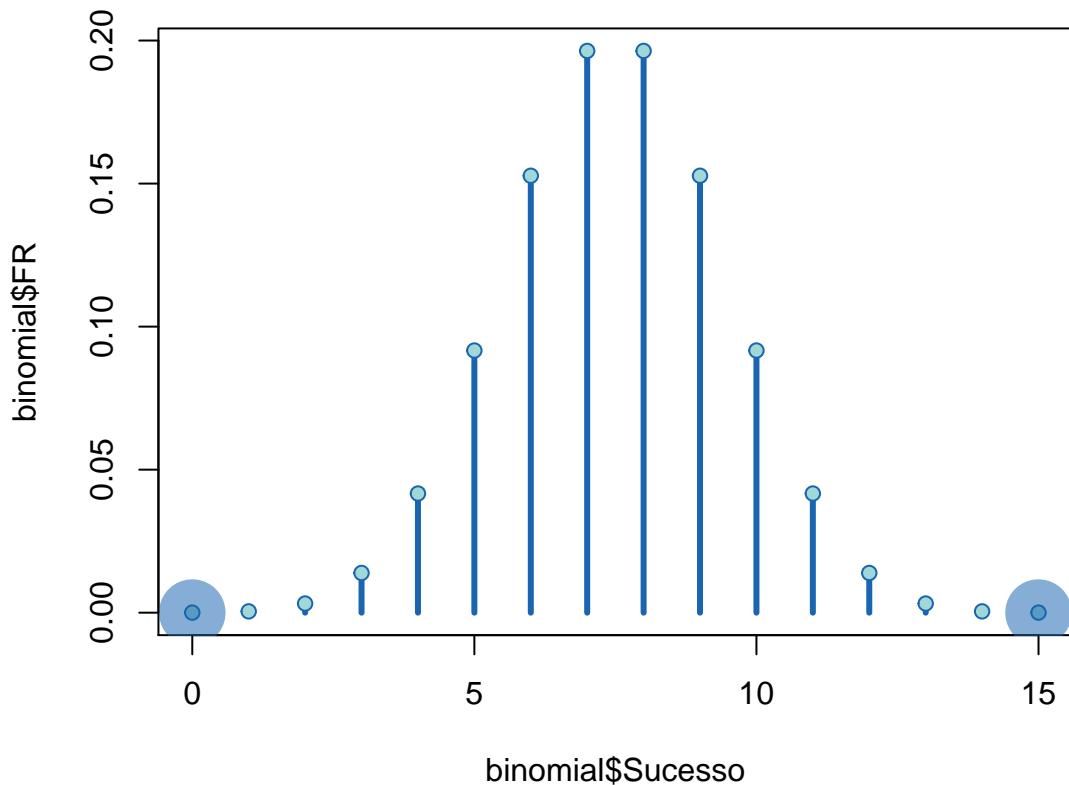
```

```

# grafico
cauda <- 1 # a partir de onde hachurar
# largura da hachura
hachura <- 500/jogadas
if (hachura < 10) {hachura <- 10}
if (hachura > 35) {hachura <- 35}
plot(binomial$Sucesso,
      binomial$FR, # *** usando o dataframe
      main = paste("Binomial: ",
                  jogadas, " jogadas, ",
                  "P[sucesso] = ", p.sucesso,
                  sep=""),
      ylim = c(0,max(probabilidade)),
      type="h",
      col=friendlycolor(8), lwd=3)
points(sucesso,probabilidade, pch=21,
       col=friendlycolor(8),
       bg=friendlycolor(12))
# cauda esquerda
lines (binomial$Sucesso[binomial$Sucesso<cauda],
       binomial$FR[binomial$Sucesso<cauda],
       col= paste(friendlycolor(8),"88",sep=""),
       lwd=hachura, type="h")
# cauda direita
lines (binomial$Sucesso[binomial$Sucesso>15-cauda],
       binomial$FR[binomial$Sucesso>15-cauda],
       col= paste(friendlycolor(8),"88",sep=""),
       lwd=hachura, type="h")

```

## Binomial: 15 jogadas, P[sucesso] = 0.5



```
total <- sum(dbinom(c(0,15), jogadas, p.sucesso))
cat("Total = ", total, sep="")
```

## Total = 6.103516e-05

$P[s \leq 0] + P[s \geq 15] \approx 6/10000$



Note que o *data frame* facilitou a construção do gráfico e hachura de suas caudas. No entanto, os eixos dos gráficos receberam o nome das variáveis (o que não é conveniente). Dois parâmetros de *plot()* resolvem o problema:

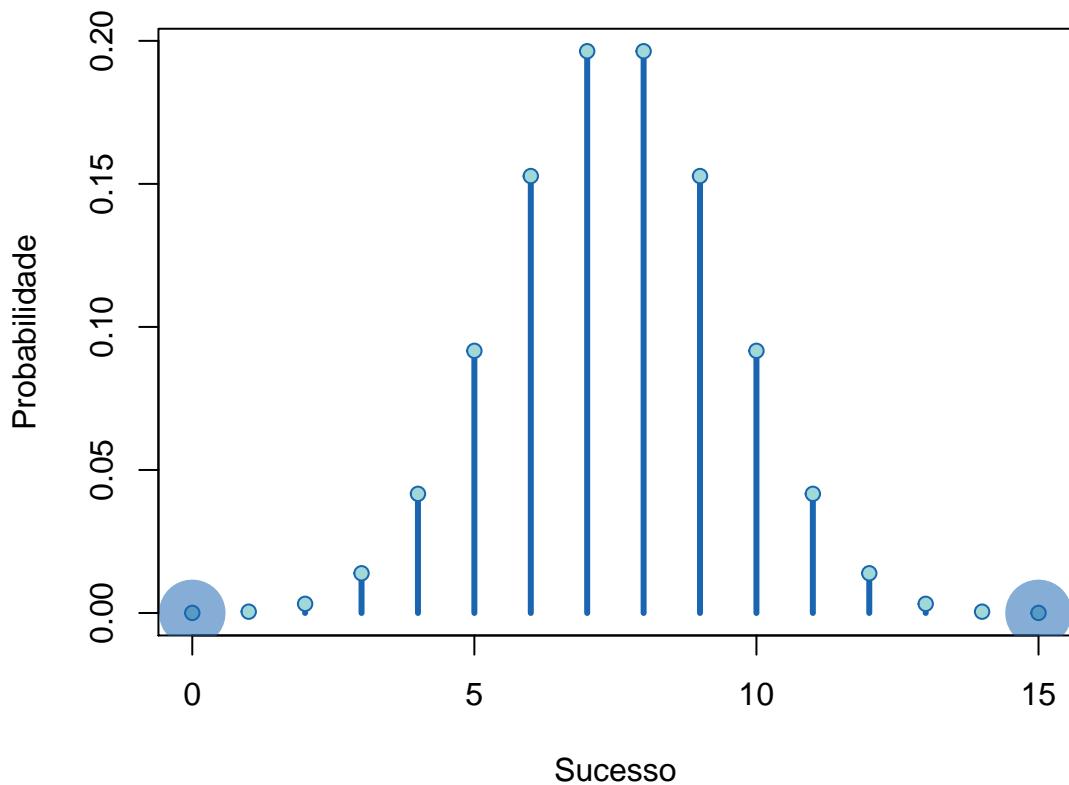
```
plot(binomial$Sucesso,
      binomial$FR,
      main = paste("Binomial: ",
                  jogadas, " jogadas, ",
                  "P[sucesso] = ", p.sucesso,
                  sep=""),
      xlab = "Sucesso", # label do eixo x
      ylab = "Probabilidade", # label do eixo y
      ylim = c(0,max(probabilidade)),
```

```

type="h",
col=friendlycolor(8), lwd=3)
points(sucesso,probabilidade, pch=21,
       col=friendlycolor(8),
       bg=friendlycolor(12))
# cauda esquerda
lines (binomial$Sucesso[binomial$Sucesso<cauda] ,
       binomial$FR[binomial$Sucesso<cauda] ,
       col=paste(friendlycolor(8),"88",sep=""),
       lwd=hachura, type="h")
# cauda direita
lines (binomial$Sucesso[binomial$Sucesso>15-cauda] ,
       binomial$FR[binomial$Sucesso>15-cauda] ,
       col=paste(friendlycolor(8),"88",sep=""),
       lwd=hachura, type="h")

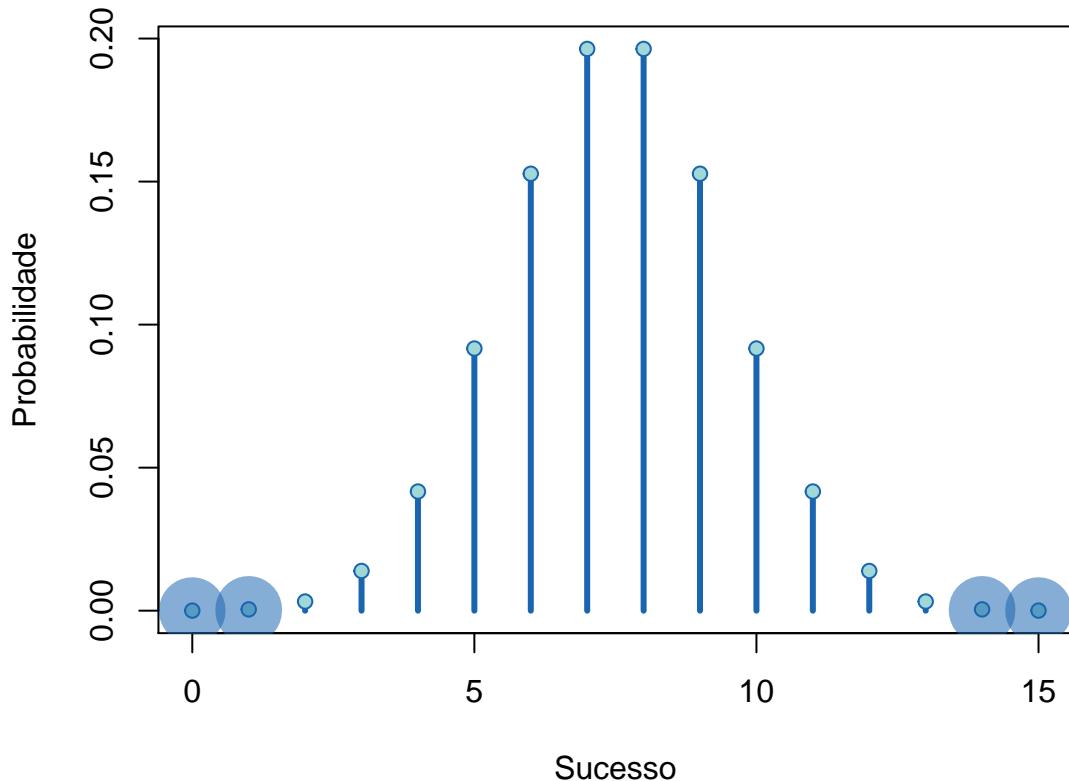
```

**Binomial: 15 jogadas,  $P[\text{sucesso}] = 0.5$**



Cauda = 2

### Binomial: 15 jogadas, $P[\text{sucesso}] = 0.5$



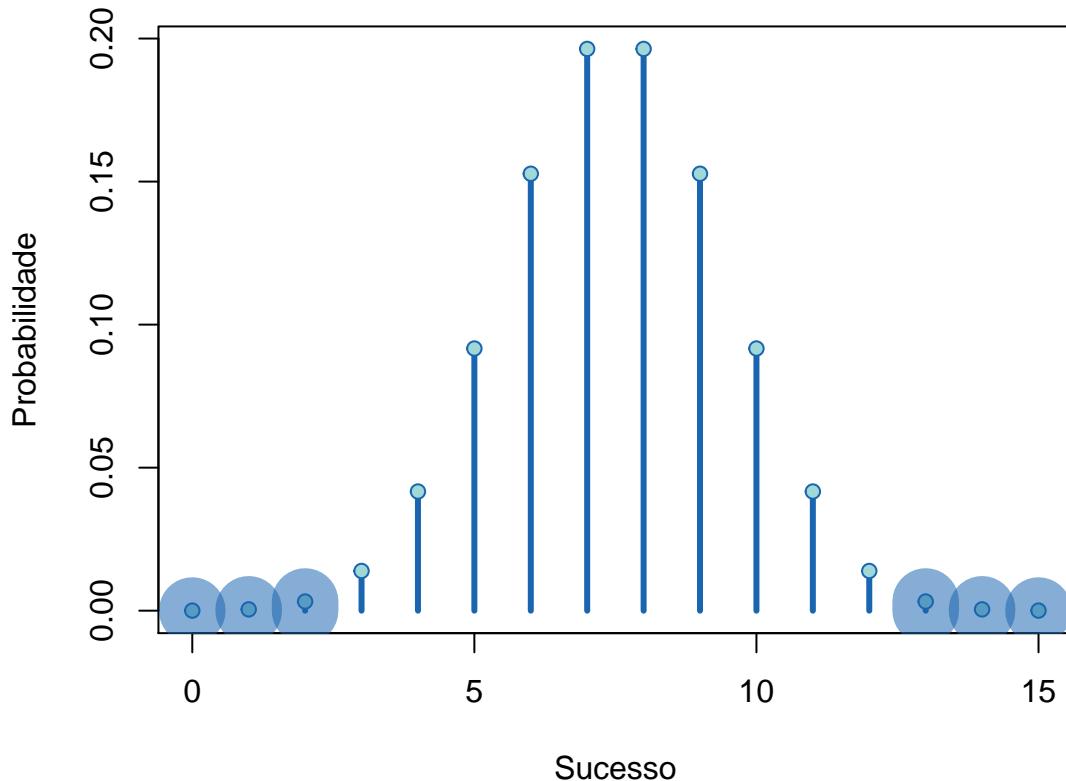
```
total <- sum(dbinom(c(0:1,14:15),jogadas,p.sucesso))
cat("Total = ", total, sep="")
```

## Total = 0.0009765625

$P[s \leq 1] + P[s \geq 14] \approx 1/1000$

Cauda = 3

**Binomial: 15 jogadas,  $P[\text{sucesso}] = 0.5$**



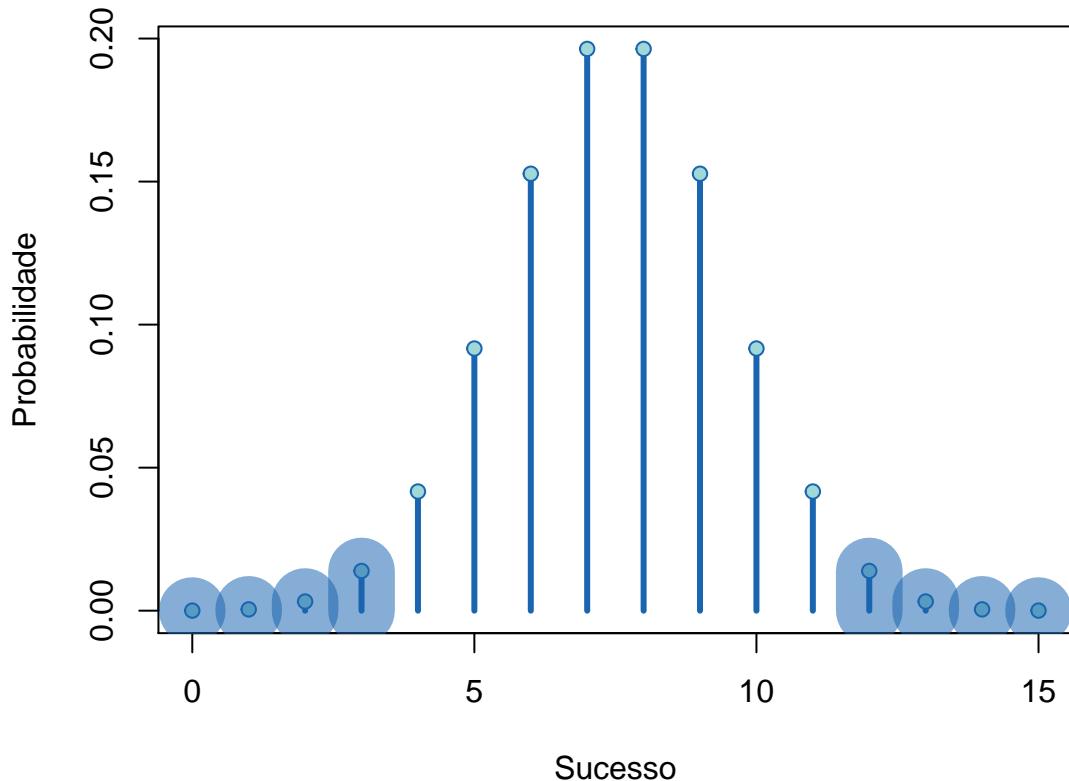
```
total <- sum(dbinom(c(0:2,13:15),jogadas,p.sucesso))
cat("Total = ", total, sep="")
```

## Total = 0.007385254

$P[s \leq 2] + P[s \geq 13] \approx 0.7\%$

Cauda = 4

### Binomial: 15 jogadas, $P[\text{sucesso}] = 0.5$

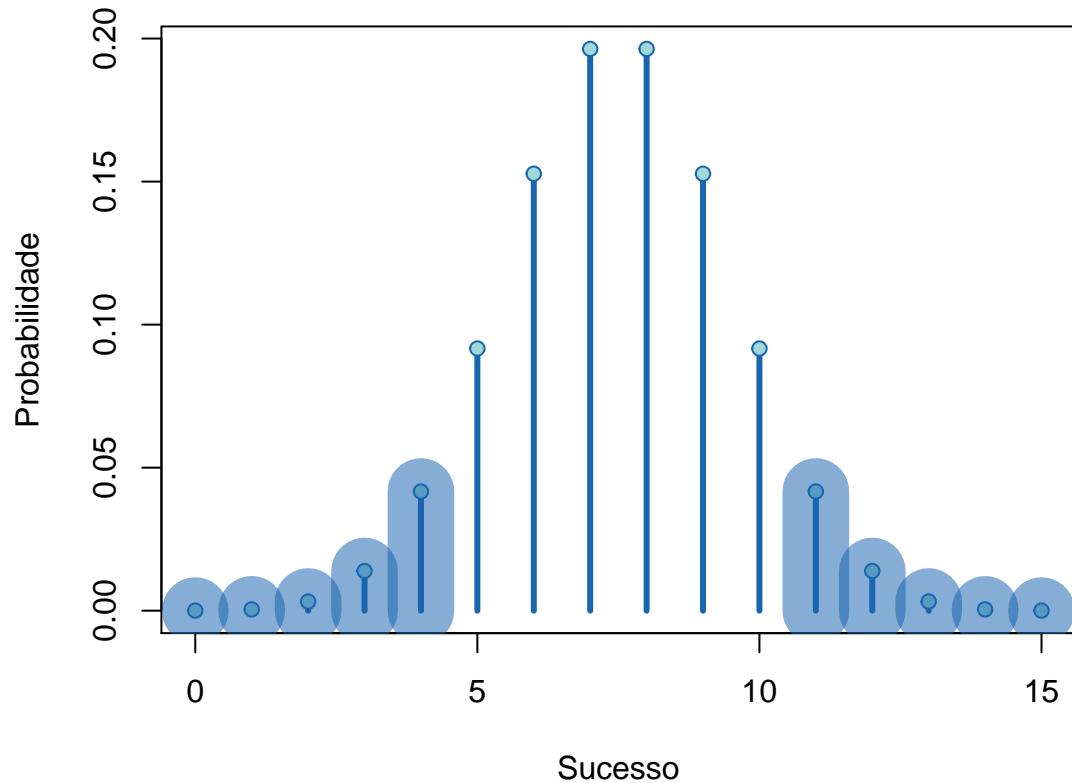


```
total <- sum(dbinom(c(0:3,12:15),jogadas,p.sucesso))
cat("Total = ", total, sep="")
```

```
## Total = 0.03515625
P[s ≤ 3] + P[s ≥ 12] ≈ 3.52%
```

Cauda = 5

**Binomial: 15 jogadas,  $P[\text{sucesso}] = 0.5$**



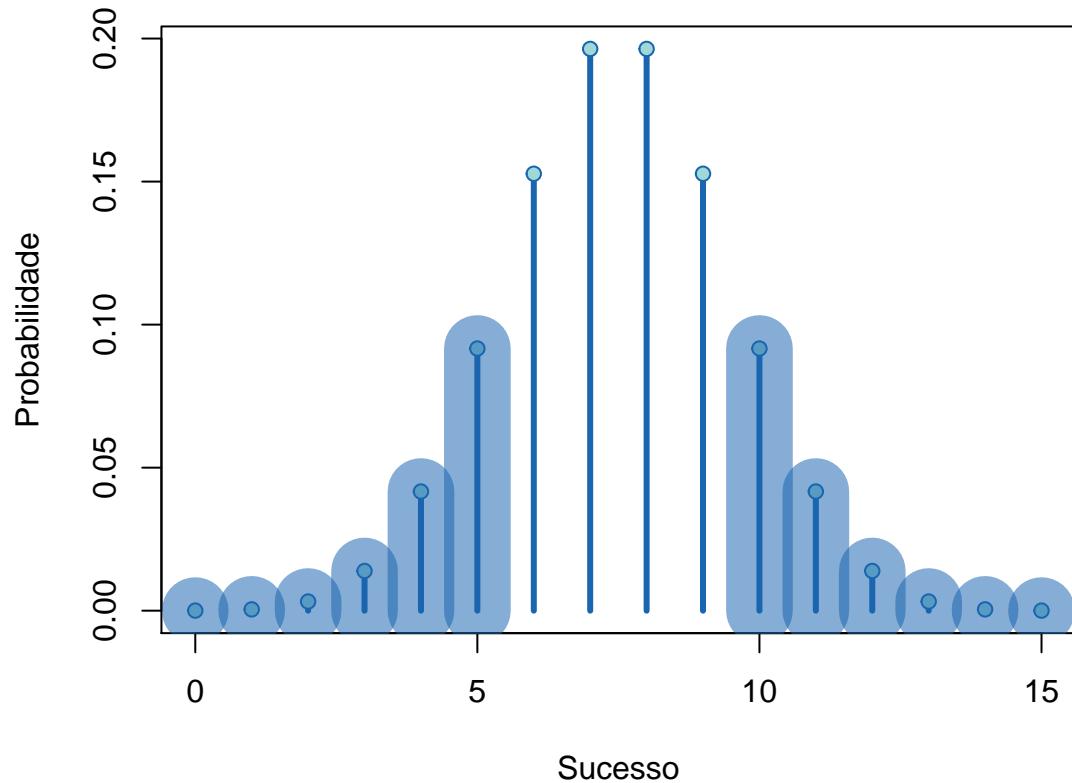
```
total <- sum(dbinom(c(0:4,11:15),jogadas,p.sucesso))
cat("Total = ", total, sep="")
```

## Total = 0.1184692

$P[s \leq 4] + P[s \geq 11] \approx 11.85\%$

Cauda = 6

## Binomial: 15 jogadas, $P[\text{sucesso}] = 0.5$



```
total <- sum(dbinom(c(0:5,10:15),jogadas,p.sucesso))
cat("Total = ", total, sep="")
```

## Total = 0.3017578

$P[s \leq 5] + P[s \geq 10] \approx 30.18\%$

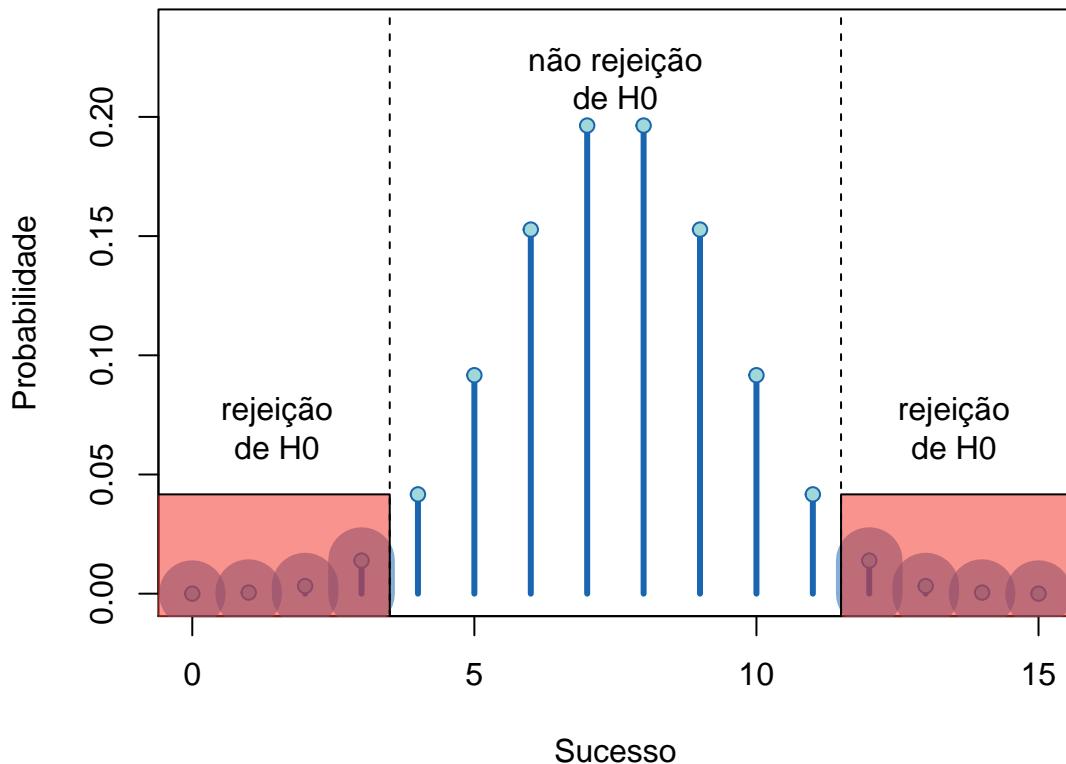
Voltando para cauda = 4

$P[s \leq 3] + P[s \geq 12] \approx 3.52\%$

$H_0 : P[\text{sucesso}] = 0.5$

$H_1 : P[\text{sucesso}] \neq 0.5$

## Binomial: 15 jogadas, $P[\text{sucesso}] = 0.5$



$\alpha \dots$  probabilidade do erro do tipo I

(rejeitar  $H_0$  incorretamente)

### Simulação 1 com Goodcoin.R

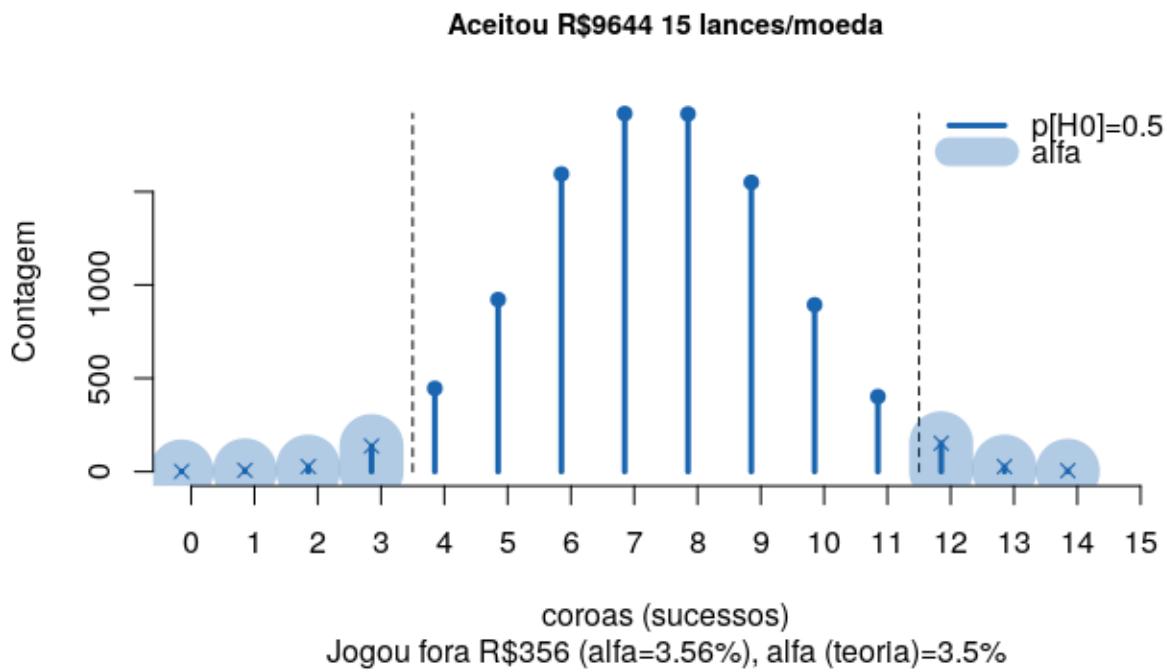
Dado um valor a receber em moedas de R\$1.00, metade da quantia eh oferecida em moedas com um balanceamento de referencia, e metade em moedas falsas, conhecidas por balanceamento distinto. Seu desafio eh distinguir os dois conjuntos atraves de experimentos. Numero de moedas (inteiro, default=10000): **10000**

Para testar se a moeda eh verdadeira, joga-se cara ou coroa certo numero de vezes cada moeda (um experimento). Numero de lancamentos por experimento (numero inteiro, default=15): **15**

Qual a proporcao maxima de moedas verdadeiras que voce aceita perder, i.e. alfa = probabilidade do erro do tipo I ou de falso-positivo). (numero entre 0 e 1). alfa (default=0.05): **0.05**

As moedas verdadeiras tem balanceamento de referencia ( $H_0$ ). (caso queira moedas balanceadas, escolha o valor igual a 0.5) Qual a probabilidade de sortear coroa para uma moeda verdadeira? (número entre 0 e 1).  $P[\text{coroa}|H_0]$  (default=0.5): **0.5**

```
source("Goodcoin.R")
```



### Binomial adaptada a um tratamento

Certo método educacional consegue ensinar higiene pessoal a 50% dos pacientes com autismo.

Com o novo método proposto pelo Instituto Ayres Soares, no entanto, entre 15 crianças acompanhadas, 10 (66.66%) conseguiram aprender a cuidar de sua higiene.

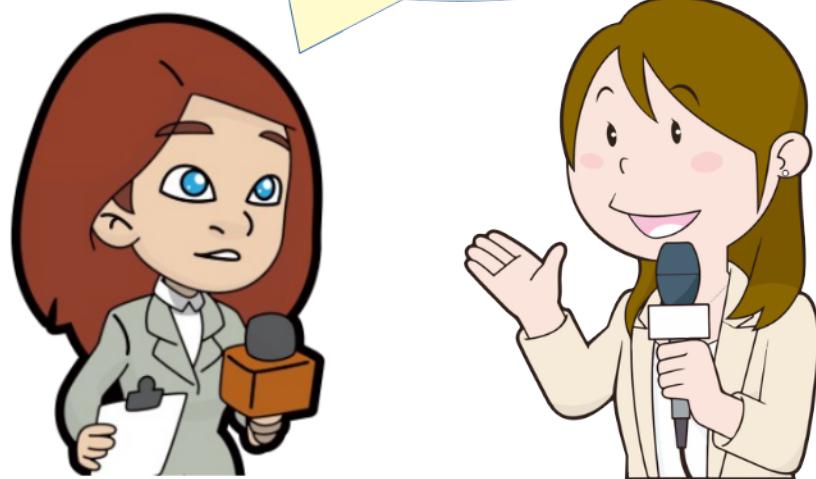
Se os métodos fossem iguais, somente metade das crianças (7 ou 8) deveriam aprender.

O novo método é melhor?

No Instituto Ayres Soares,  
estão ensinando as crianças autistas com sucesso!  
Antes, só 50% das crianças com autismo conseguiam  
aprender cuidados de higiene pessoal, mas agora os pesquisadores  
conseguiram ensinar 10 das 15 crianças estudadas, 67%,  
que é uma **taxa** de 134% de sucesso em comparação  
com o método tradicional, **provando** que este  
método novo funciona mesmo!



A melhora foi de 50% para 67%,  
um ganho de apenas 17%.  
Você não **acha** que é **muito pouco**  
para **confirmar** que  
este método é melhor  
que o antigo?





São crianças autistas,  
então qualquer  
aumento é **significativo**,  
você não acha?



Não sei não:  
50% seriam 7 ou 8 em 15.  
Conseguiram 10, o que são 2 a mais.  
Duas em 15 dá 13.3%.  
Eu acho pouco...



## Hipótese nula e alternativa

$$H_0 : \mu_{novo} = 0.5$$

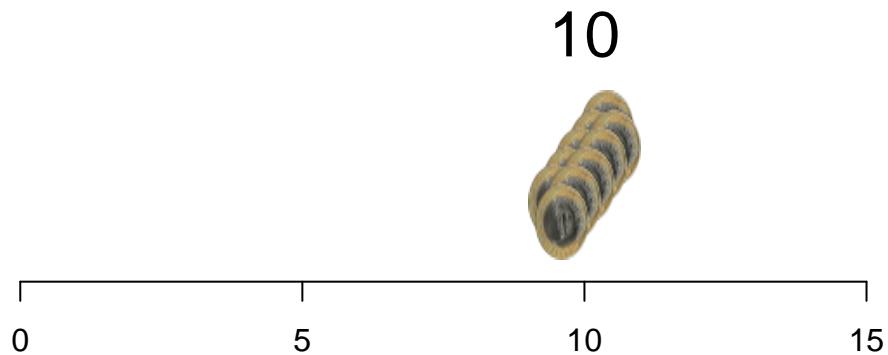
$$H_1 : \mu_{novo} > 0.5$$

$$\alpha = 0.05 = 5\%$$

(este teste é unicaudal, tem direção)

## Experimento único

Resultado: 10 crianças em 15 aprenderam com o novo método.



### valor-p

Probabilidade de observar a melhora de 10 crianças em 15 testadas, sob  $H_0$  (i.e., assumindo-se que o novo tratamento tem o mesmo efeito que o tratamento antigo).

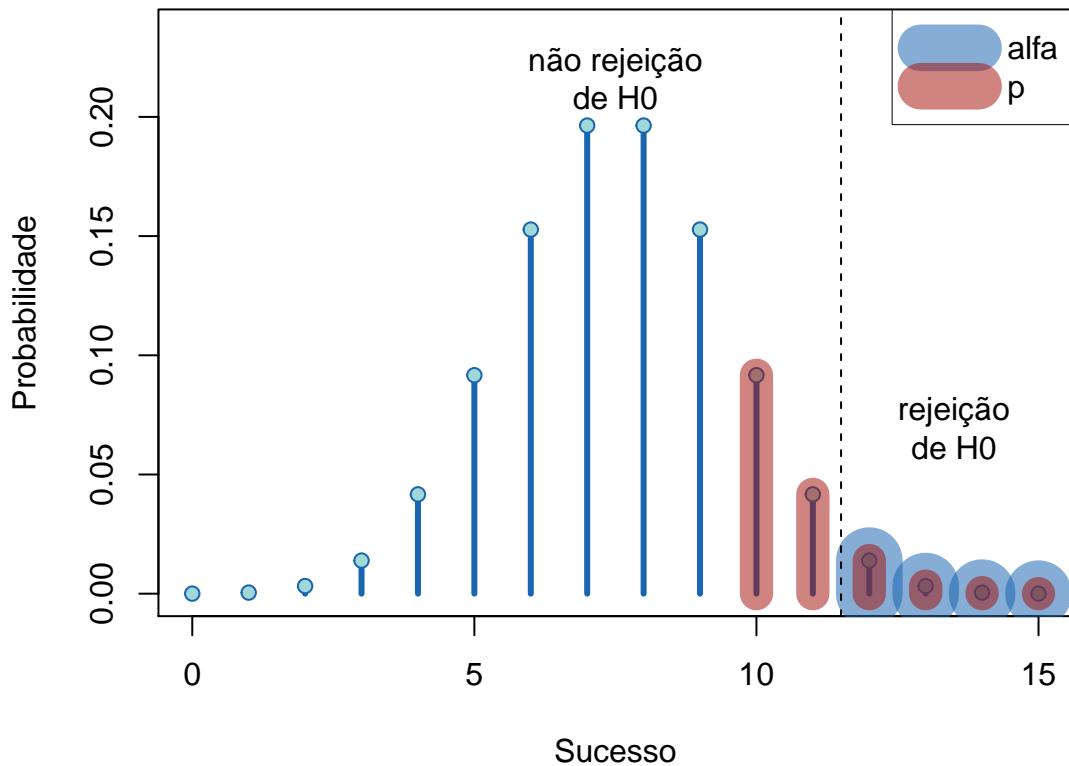
```
p <- sum(binomial$FR[binomial$Sucesso>=10])
cat("p = ",p,"\\n")
## p = 0.1508789
P[s ≥ 10] ≈ 15.09%
```

### alfa ( $\alpha$ )

Probabilidade do erro do tipo I: critério escolhido pelo pesquisador.

```
alfa <- sum(binomial$FR[binomial$Sucesso>=12])
cat("alfa = ",alfa,"\\n")
## alfa = 0.01757812
P[s ≤ 3] + P[s ≥ 12] ≈ 1.78%
```

## Binomial: 15 jogadas, $P[\text{sucesso}] = 0.5$



Decisão: não se rejeita  $H_0$

Portanto, não há evidência de que o novo tratamento seja superior ao tratamento tradicional, tomado como referência.



## Simulação 2 com Goodcoin.R

```
source("Goodcoin.R")
```

Dado um valor a receber em moedas de R\$1.00, metade da quantia eh oferecida em moedas com um balanceamento de referencia, e metade em moedas falsas, conhecidas por balanceamento distinto. Seu desafio eh distinguir os dois conjuntos atraves de experimentos. Numero de moedas (inteiro, default=10000): **20000**

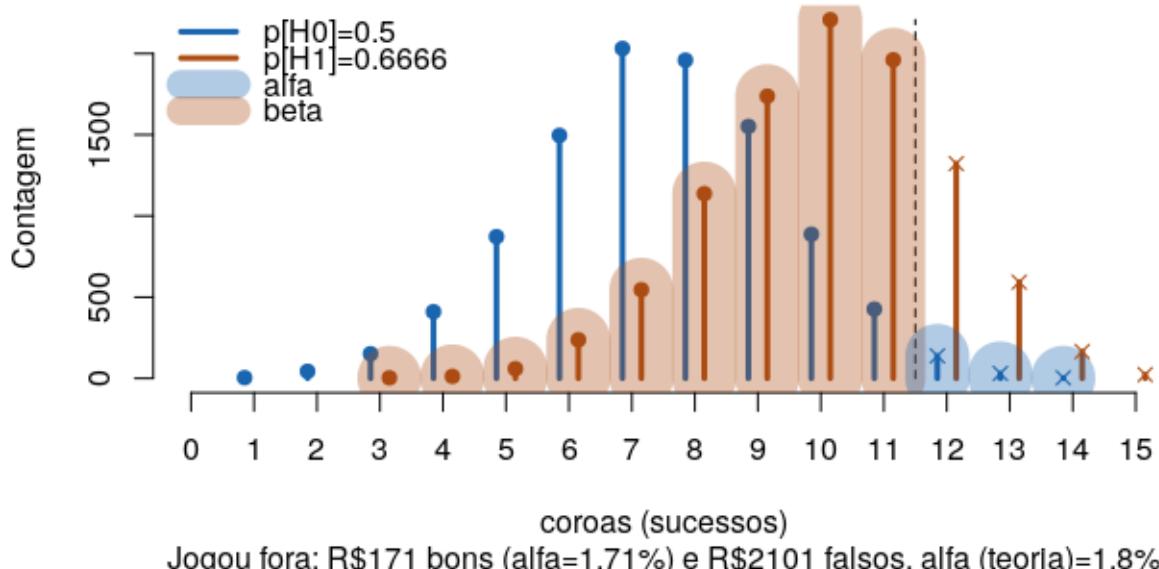
Para testar se a moeda eh verdadeira, joga-se cara ou coroa certo numero de vezes cada moeda (um experimento). Numero de lancamentos por experimento (numero inteiro, default=15): **15**

Qual a proporcao maxima de moedas verdadeiras que voce aceita perder, i.e. alfa = probabilidade do erro do tipo I ou de falso-positivo). (numero entre 0 e 1). alfa (default=0.05): **0.05**

As moedas verdadeiras tem balanceamento de referencia (H0). (caso queira moedas balanceadas, escolha o valor igual a 0.5) Qual a probabilidade de sortear coroa para uma moeda verdadeira? (número entre 0 e 1). P[coroa|H0] (default=0.5): **0.5**

As moedas falsas tem outro balanceamento. (para simular, forneça uma probabilidade diferente de 0.5 ou deixe em branco para simular somente a moeda verdadeira) Qual a probabilidade de sortear coroa para uma moeda falsa? (número entre 0 e 1). P[coroa|H1] (default=0.5): **0.6666**

**Aceitou R\$9829 bons e R\$7899 falsos (beta=78.99%), 15 lances/moeda**



Jogou fora: R\$171 bons (alfa=1.71%) e R\$2101 falsos, alfa (teoria)=1.8%

$\beta$  ... probabilidade do erro do tipo II

(não rejeitar  $H_0$  incorretamente)

**Tomada de decisão:**  $\alpha$  e  $\beta$

$H_0$  verdadeira

$H_0$  falsa

não rejeita  $H_0$

ok

$\beta$

rejeita  $H_0$

$\alpha$

ok

**poder** ( $1 - \beta$ )

Probabilidade de rejeitar  $H_0$  corretamente:

O efeito **não existe**

O efeito **existe**

**sem** evidência de efeito

não rejeitou  $H_0$ , corretamente

$\beta$

**com** evidênciade efeito

$\alpha$

rejeitou  $H_0$ ,corretamente  $poder = 1 - \beta$

**e o que acontece na prática?**

**Não sabemos** se o efeito existe

não rejeitou  $H_0$ ,**sem** evidênciade efeito, então...

... o efeito não existe**OU** ... o efeito existe e a probabilidade de **decisão errada é  $\beta$**  (se o efeito existe -> erro do tipo II)

rejeitou  $H_0$ ,**com** evidênciade efeito, então...

... o efeito não existe e a probabilidade de **decisão errada é  $\alpha$**  (se o efeito não existe -> erro do tipo I)  
**OU** ... o efeito existe e a probabilidade de **decisão correta é  $1 - \beta$** , i.e., o poder do teste (de declarar a diferença acertadamente).

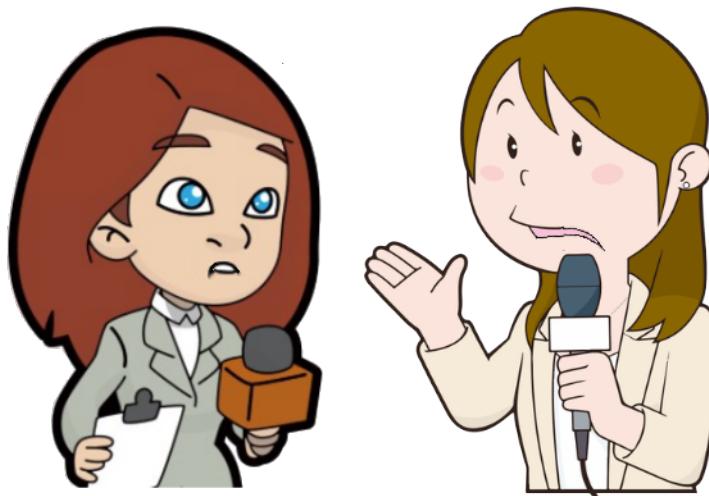
Então, neste exemplo, como **não rejeitamos  $H_0$** , declarar que os dois tratamentos são iguais (i.e., aceitar  $H_0$ ) é a decisão incorreta com probabilidade de ...

```
p.sucesso <- 0.6666  
jogadas <- 15  
beta <- sum(dbinom(0:11,jogadas,p.sucesso))  
cat("beta = ",beta,"\\n")
```

```
## beta = 0.7909156
```

**Conclusão: ESTE ESTUDO É INCONCLUSIVO.**

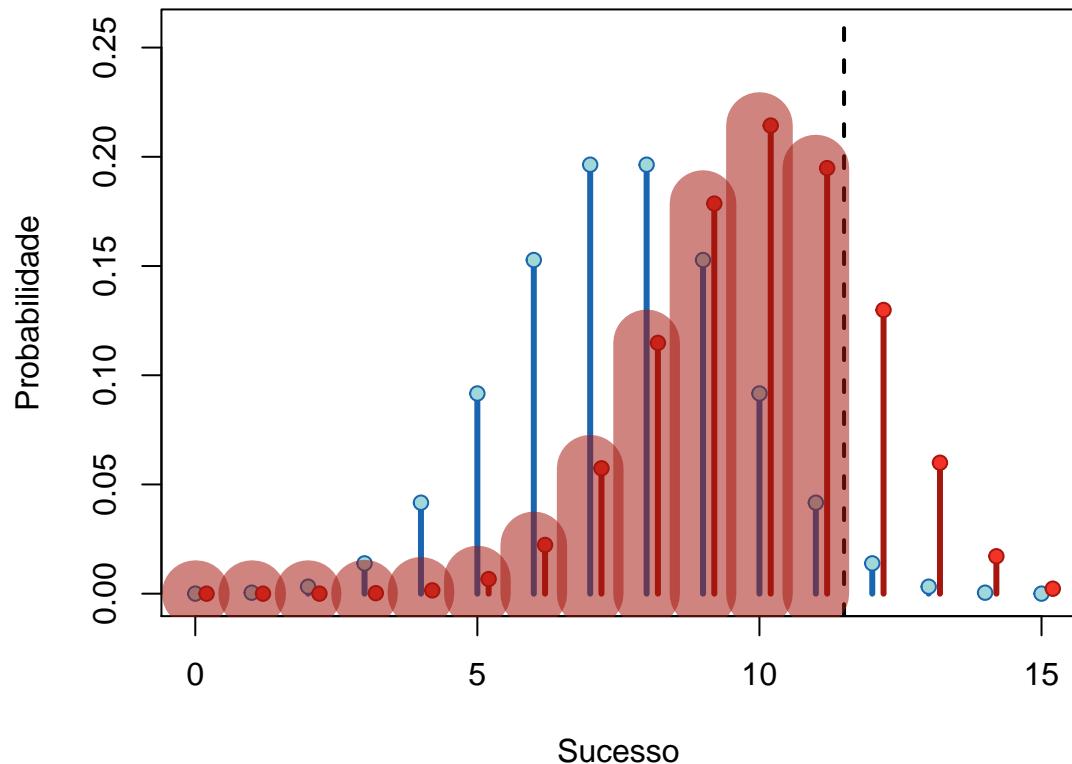
Não temos evidência para dizer que os dois tratamentos são diferentes, e menos ainda podemos afirmar que os dois são iguais.



**O que fazer para reduzir  $\beta$ ?**

A literatura costuma usar o nível de significância  $\alpha = 0.05 = 5\%$  e poder entre 80% e 90% (de  $\beta = 0.2 = 20\%$  a  $\beta = 0.1 = 10\%$ ).

## Binomial: 15 jogadas

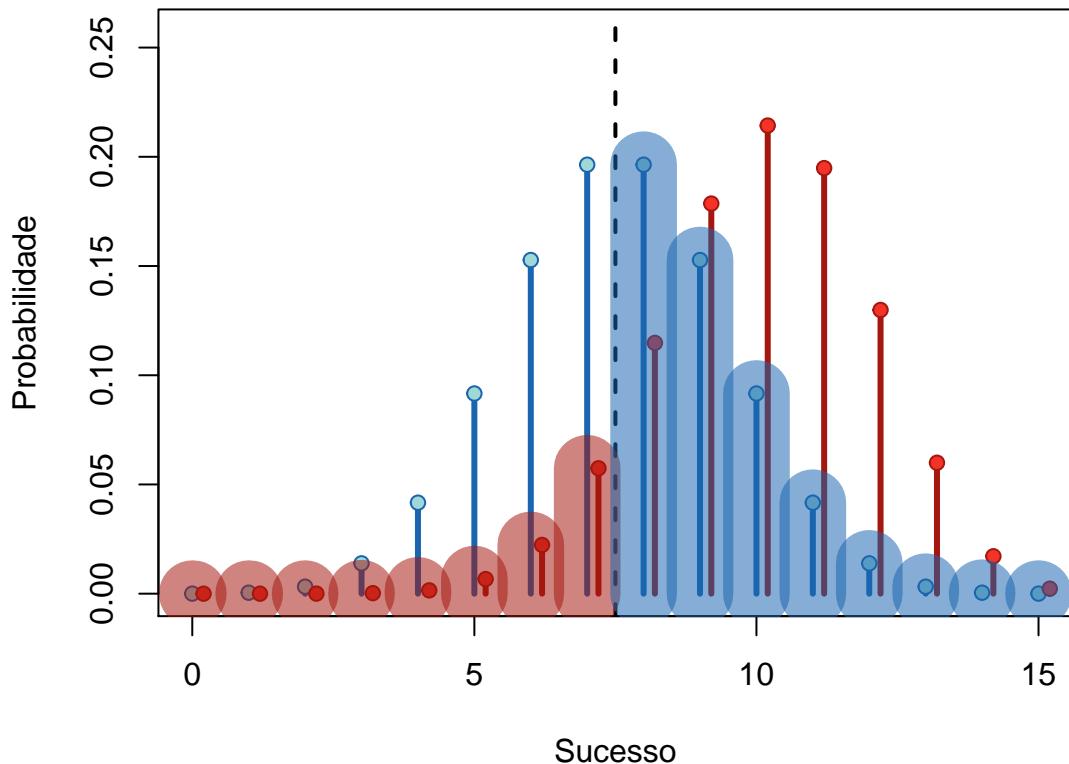


Estratégia 1: aumentar o valor de  $\alpha$

```
source("Goodcoin.R")
```

Mesmo com  $alfa = 0.6 \dots$

## Binomial: 15 jogadas



```
## cutoff = 7
cat("beta = ", round(beta*100, 2), "%\n")

## beta = 8.82 %
cat("poder = ", round((1-beta)*100, 2), "%\n")

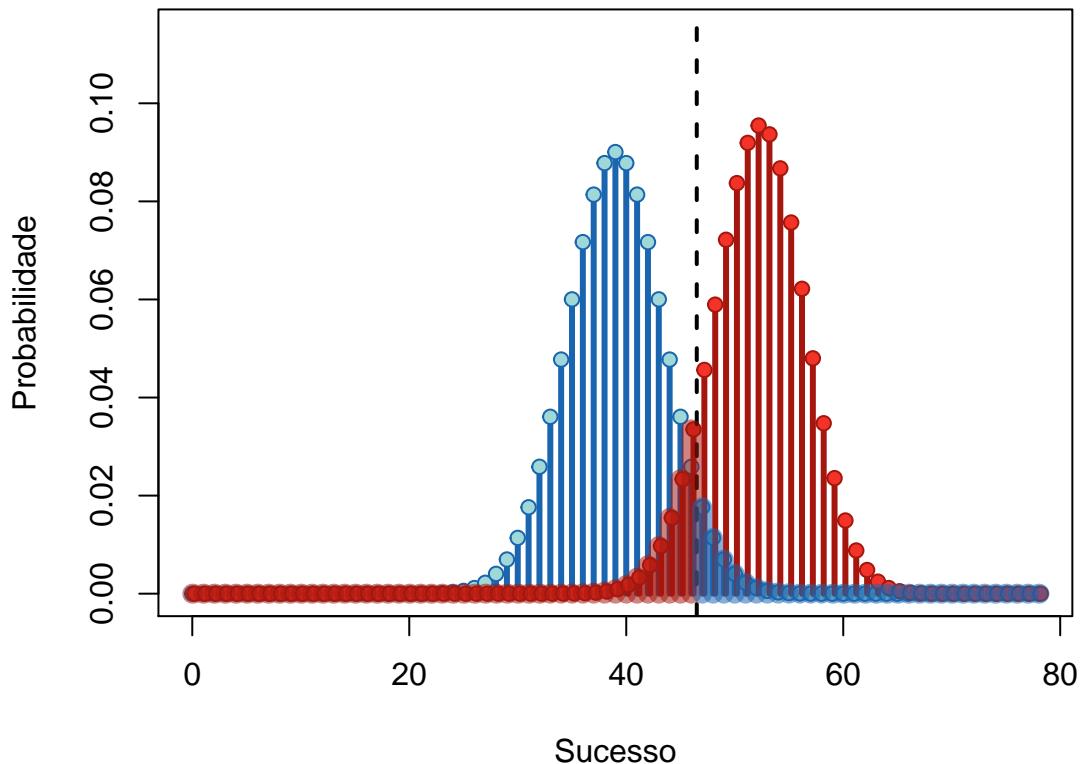
## poder = 91.18 %
```

Estratégia 2: tornar as distribuições mais estreitas

```
source("Goodcoin.R")
```

Com 78 crianças e  $\alpha = 0.05 \dots$

## Binomial: 78 jogadas



```

## cutoff = 46
cat("beta = ", round(beta*100, 2), "%\n")

## beta = 9.46 %
cat("poder = ", round((1-beta)*100, 2), "%\n")

## poder = 90.54 %

```

Aplicando-se o novo método aplicado a 78 crianças:

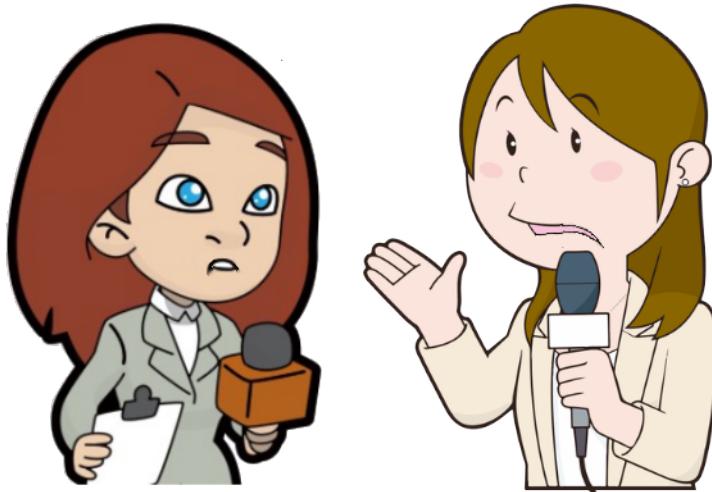
- se 47 ou mais crianças aprenderem higiene pessoal, rejeitamos  $H_0$  e podemos afirmar que o novo tratamento é melhor que o tradicional com 90% de probabilidade de estarmos corretos (no entanto, pode ser que os tratamentos sejam iguais e estejamos errados, com  $\alpha = 5\%$  de probabilidade).
- se somente até 46 crianças aprenderem higiene pessoal, não podemos rejeitar  $H_0$ . No entanto, como escolhemos poder de 90% **antes do iniciarmos o estudo**, podemos aceitar  $H_0$  e afirmar que os dois tratamentos são iguais porque a igualdade não deve ser decorrente de insuficiência amostral (no entanto, podemos estarmos enganados e, na verdade, o novo tratamento ser melhor, com  $\beta = 10\%$  de probabilidade).

Notou que

$$\frac{47}{78} = 0.6025641 \approx 60\%$$

e antes, quando foi inconclusivo,

$$\frac{10}{15} = 0.6666667 \approx 67\% ?$$



### Estratégia 3: ser capaz de detectar, somente, maiores efeitos

```
source("Goodcoin.R")
```

Caso eu não tenha como avaliar mais do que 15 crianças...

```
jogadas <- 15
sucesso <- 0:jogadas
p.sucesso <- 0.88 # *** um novo tratamento, com efeito maior
probabilidade <- dbinom(sucesso,jogadas,p.sucesso)
binomial <- data.frame(sucesso,probabilidade)
names(binomial) <- c("Sucesso","FR")
binomial$FA <- NA
binomial$FAdec <- NA
for(b in 1:(jogadas+1))
{
  binomial$FA[b] <- sum(binomial$FR[binomial$Sucesso<=b-1])
  binomial$FAdec[b] <- sum(binomial$FR[binomial$Sucesso>=b-1])
}
print(binomial)
```

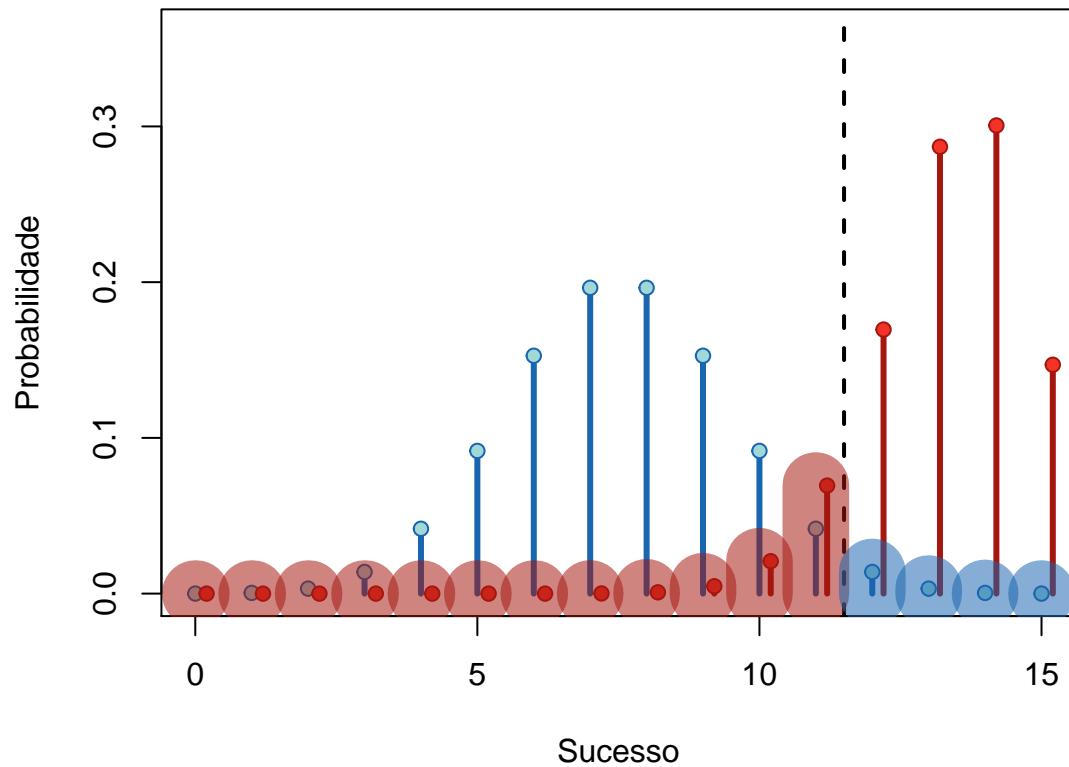
	Sucesso	FR	FA	FAdec
## 1	0	1.540702e-14	1.540702e-14	1.0000000
## 2	1	1.694772e-12	1.710179e-12	1.0000000
## 3	2	8.699832e-11	8.870849e-11	1.0000000
## 4	3	2.764613e-09	2.853322e-09	1.0000000
## 5	4	6.082149e-08	6.367481e-08	1.0000000
## 6	5	9.812534e-07	1.044928e-06	0.9999999
## 7	6	1.199310e-05	1.303802e-05	0.9999990
## 8	7	1.130778e-04	1.261158e-04	0.9999870
## 9	8	8.292370e-04	9.553528e-04	0.9998739
## 10	9	4.729722e-03	5.685075e-03	0.9990446
## 11	10	2.081078e-02	2.649585e-02	0.9943149
## 12	11	6.936925e-02	9.586511e-02	0.9735041
## 13	12	1.695693e-01	2.654344e-01	0.9041349

```

## 14      13 2.869634e-01 5.523978e-01 0.7345656
## 15      14 3.006283e-01 8.530261e-01 0.4476022
## 16      15 1.469739e-01 1.000000e+00 0.1469739

```

## Binomial: 15 jogadas



```

## cutoff = 11
cat("beta = ",round(beta*100,2),"%\n")

## beta = 9.59 %
cat("poder = ",round((1-beta)*100,2),"%\n")

## poder = 90.41 %

```

Com apenas 15 crianças disponíveis, o estudo de um novo método poderá ser conclusivo se o método for capaz de ensinar com sucesso 88% ou mais das crianças (i.e., tenha efeito maior que os 2/3 do exemplo anterior); neste caso poderá ser considerado igual ao tradicional no caso de até 11 crianças ou melhor que o tradicional se 12 ou mais crianças aprenderem higiene pessoal.

## Distribuição de Poisson

Esta é uma distribuição de probabilidades para uma variável quantitativa discreta (contagem de eventos) quando:

- não sabemos o número máximo de ocorrências;

- os eventos são raros;
- as ocorrências dos eventos são independentes;
- a probabilidade de ocorrência de um evento em um certo intervalo é a mesma para todos os demais intervalos de tempo;
- a probabilidade de ocorrência dos eventos é proporcional ao tamanho do intervalo;
- em uma porção infinitesimal do intervalo, a probabilidade de mais de uma ocorrência do evento é desprezível.

## exemplo

Certo estudo demonstrou que a distribuição mensal de acidentes de trabalho entre moradores de determinada comunidade, entre 1977 e 1987, obedecia uma distribuição de Poisson com média 33 ocorrências por ano.

Qual a probabilidade estimada de que num certo mês do ano sejam observados 3 acidentes de trabalho?

A distribuição de Poisson tem um único parâmetro,  $\lambda$ , que é a taxa de ocorrência dos eventos.

Neste exemplo, a taxa mensal é  $\frac{33}{12} = 2.75$

A função R é **dpois(x, lambda)** indicando, respectivamente, quantas ocorrências e a taxa das ocorrências.

No caso de ocorrerem 2.75 (lambda=2.75) eventos para o tempo e o tamanho da população considerada, a probabilidade de não ocorrer evento algum ( $x=0$ ) é:

```
dpois(x=0, lambda=2.75)
```

```
## [1] 0.06392786
```

de 1 ocorrência:

```
dpois(1, 2.75)
```

```
## [1] 0.1758016
```

de 2 ocorrências:

```
dpois(2, 2.75)
```

```
## [1] 0.2417272
```

etc.

Diferentemente da distribuição binomial, a de Poisson pode ser computada ao infinito.

Neste exemplo, a ocorrência de exatamente 3 eventos é:

```
dpois(x=3, lambda=2.75)
```

```
## [1] 0.2215833
```

A probabilidade de ocorrer até 3 eventos é a soma das probabilidades de ocorrência de nenhum, 1, 2 ou 3 eventos:

```
sum(dpois(0:3, lambda=2.75))
```

```
## [1] 0.70304
```

Como não há limite superior, caso quiséssemos saber a probabilidade de 4 ou mais eventos, precisamos usar o complemento:

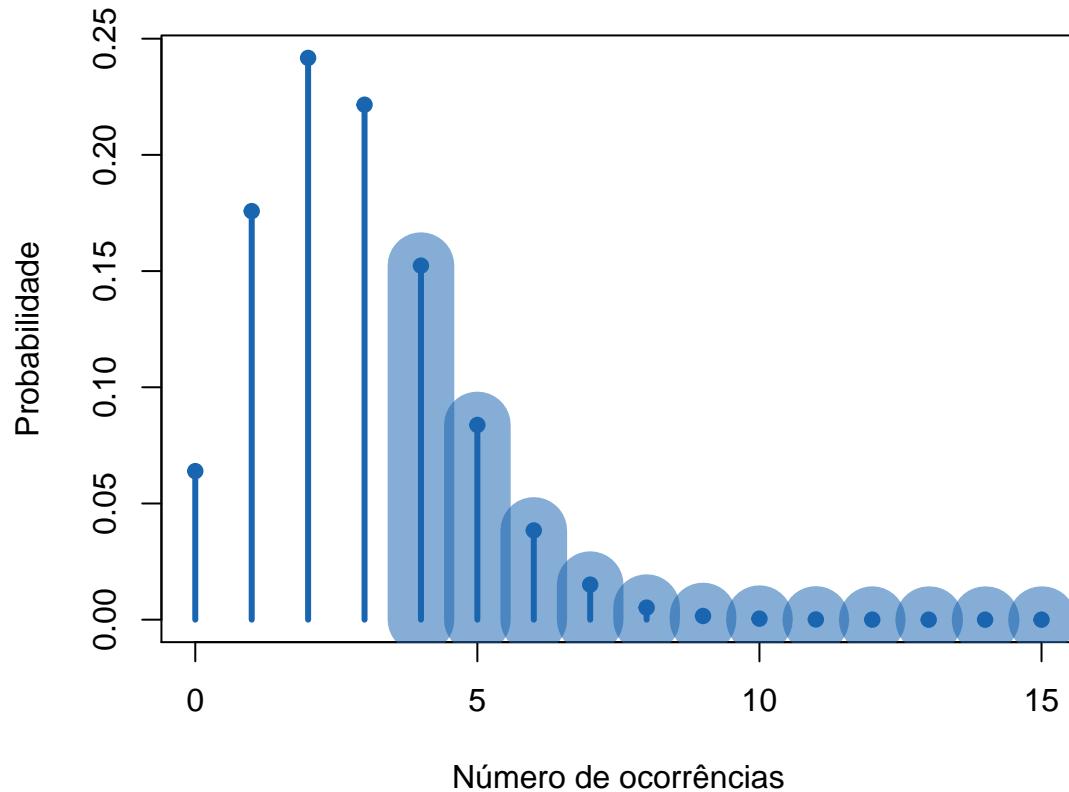
```
1-sum(dpois(0:3, lambda=2.75))
```

```
## [1] 0.29696
```

O seguinte código mostra a aparência desta distribuição:

```
# Poisson_com_caudas.R
# altere o que deseja
lambda <- 2.75
max_plotar <- 15 # valor maximo para plotar (Poisson vai a + infinito)
cauda <- 3 # a partir de onde hachurar e calcular p da area
cor = "#1965B0" # RGB
# grafico e calculo
hachura <- 500/max_plotar
if (hachura < 10) {hachura <- 10}
if (hachura > 35) {hachura <- 35}
cor_transparencia <- paste(cor,"88",sep="")
eventos <- 0:max_plotar
probs <- dpois(eventos,lambda)
# data frame com a distribuição de Poisson
poisson <- data.frame(eventos,probs)
names(poisson) <- c("Eventos", "Probabilidade")
plot (poisson$Eventos, poisson$Probabilidade,
      main="Distribuição de Poisson",
      xlab="Número de ocorrências",
      ylab="Probabilidade",
      lwd=3, col=cor, type = "h")
points(poisson$Eventos, poisson$Probabilidade,
       col=cor, bg=cor, pch=21)
# cauda direita
lines (poisson$Eventos[poisson$Eventos>cauda] ,
       poisson$Probabilidade[poisson$Eventos>cauda] ,
       col= paste(cor,"88",sep=""),
       lwd=hachura, type="h")
```

## Distribuição de Poisson



```
print(poisson)

##      Eventos Probabilidade
## 1          0 6.392786e-02
## 2          1 1.758016e-01
## 3          2 2.417272e-01
## 4          3 2.215833e-01
## 5          4 1.523385e-01
## 6          5 8.378618e-02
## 7          6 3.840200e-02
## 8          7 1.508650e-02
## 9          8 5.185984e-03
## 10         9 1.584606e-03
## 11        10 4.357667e-04
## 12        11 1.089417e-04
## 13        12 2.496580e-05
## 14        13 5.281228e-06
## 15        14 1.037384e-06
## 16        15 1.901871e-07

prob_dir <- sum(poisson$Probabilidade[poisson$Eventos<=cauda])
cat("\nP[eventos <= ",cauda,"] =",prob_dir)
```

##

```

## P[eventos <= 3] = 0.70304
cat("\nP[eventos > cauda,] =", 1-prob_dir)

##
## P[eventos > 3] = 0.29696
cat("\n")

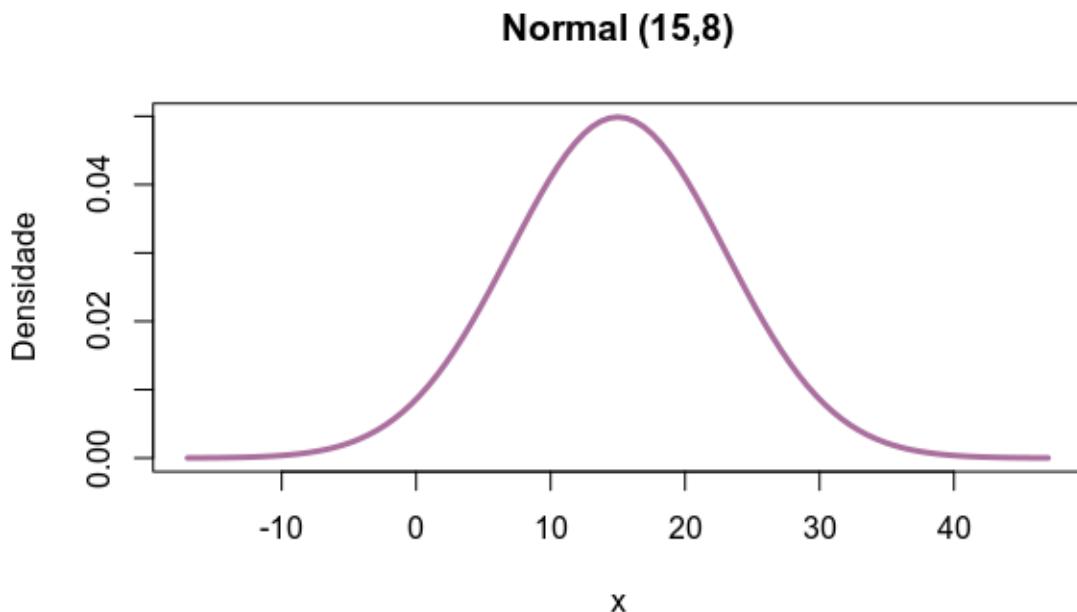
```

## Distribuição normal

A distribuição normal tem dois parâmetros, média ( $\mu$ ) e desvio-padrão ( $\sigma$ ).

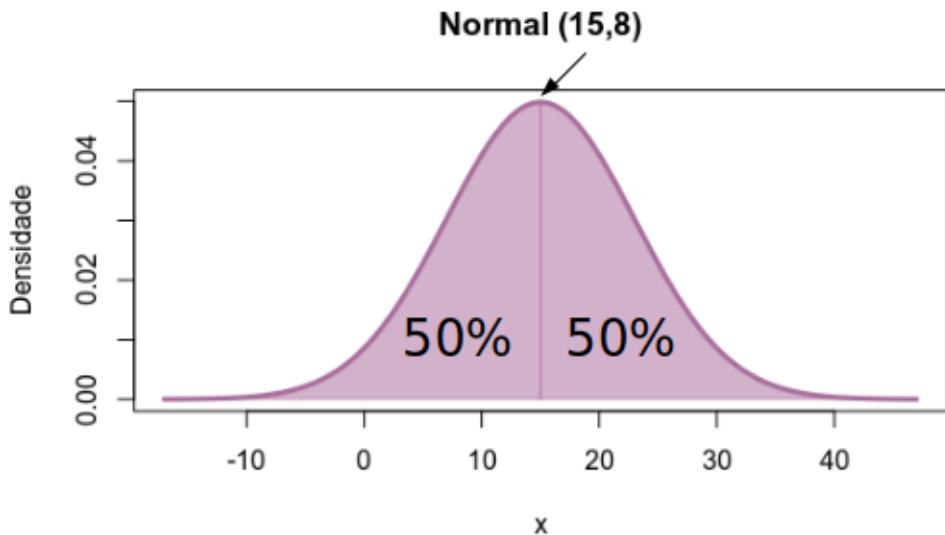
### aparência

Vamos assumir uma distribuição  $N(\mu = 15, \sigma = 8)$ :



### simetria

É uma distribuição simétrica, portanto média, moda e mediana coincidem:

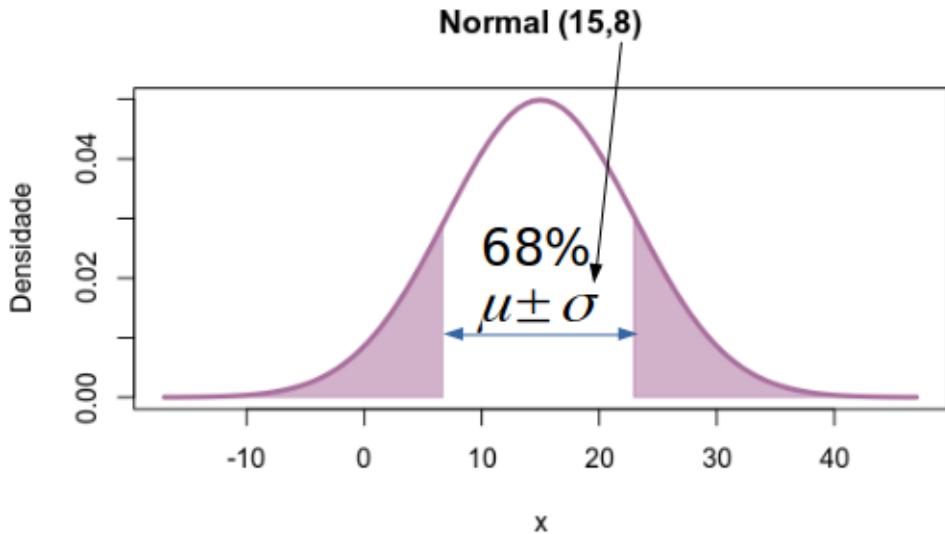


Metade da área sob a curva está à esquerda e metade à direita da média.

### áreas sob a curva

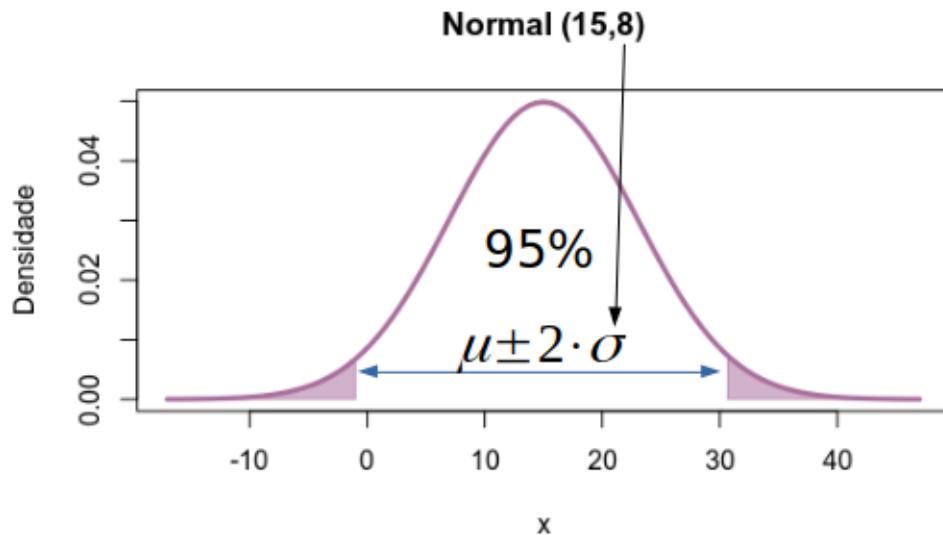
$\pm 1dp$

cerca de 68% da área entre -1 e +1 desvio-padrão:



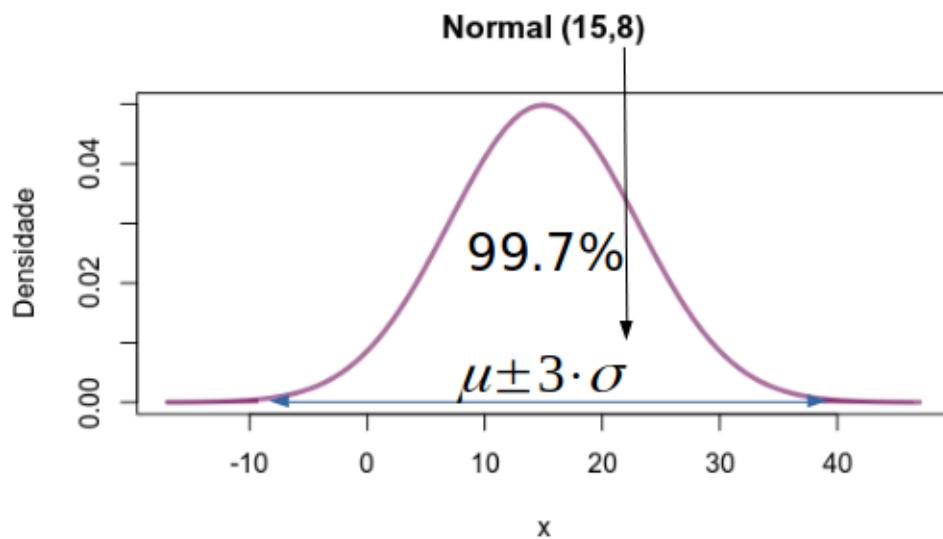
$\pm 2dp$

cerca de 95% da área entre -2 e +2 desvio-padrão:



$\pm 3dp$

cerca de 99.7% da área entre -3 e +3 desvio-padrão:



### variando média e desvio-padrão

Para cada par de parâmetros  $\mu$  e  $\sigma$ , define-se completamente uma distribuição normal.

Admita (não tome estes valores como variáveis da prática médica) que as seguintes variáveis tenham distribuições normais:

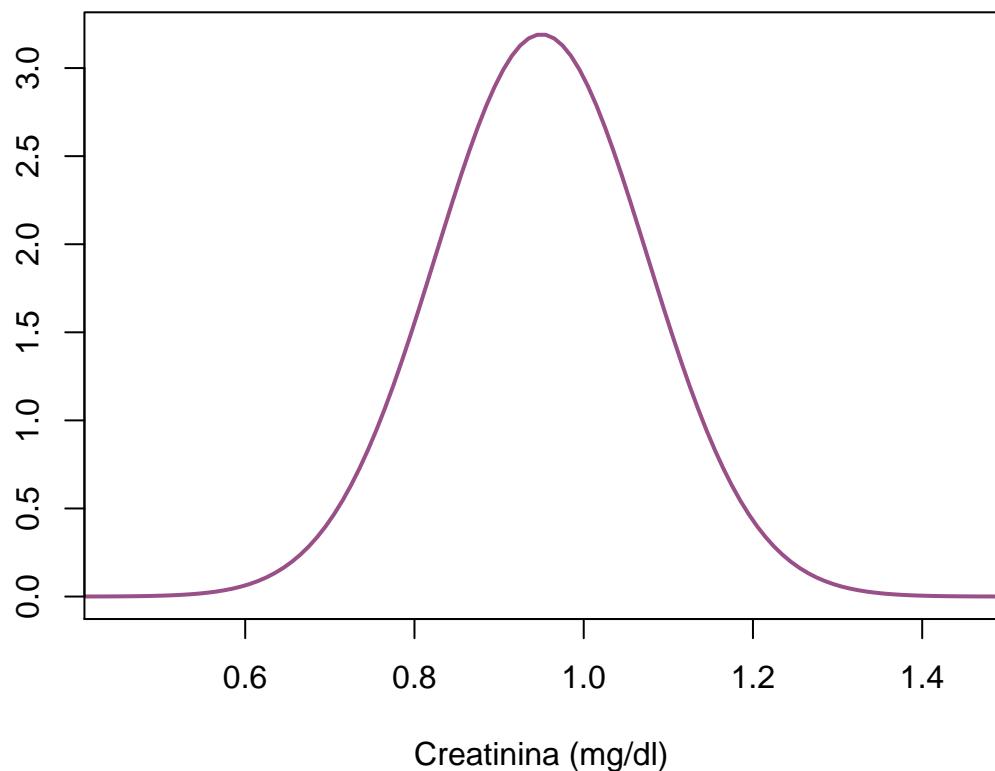
```
# GraficoNormal.R
source ("friendlycolor.R")
variavel <- "Creatinina"
unidade <- "mg/dl"
media <- 0.95
desvpad <- 0.125
x <- seq(from=media-5*desvpad, to=media+5*desvpad, by=0.01)
```

```

y <- dnorm(x, mean=media, sd=desvpad)
xy <- data.frame(x,y)
plot(x,y,
  main=paste("N(",media,",",desvpad,")",sep=""),
  xlab=paste(variavel, " (",unidade,")",sep=""),
  ylab=NA,
  xlim=c(media-4*desvpad,media+4*desvpad),
  col=friendlycolor(2),type="l",lwd=2
)

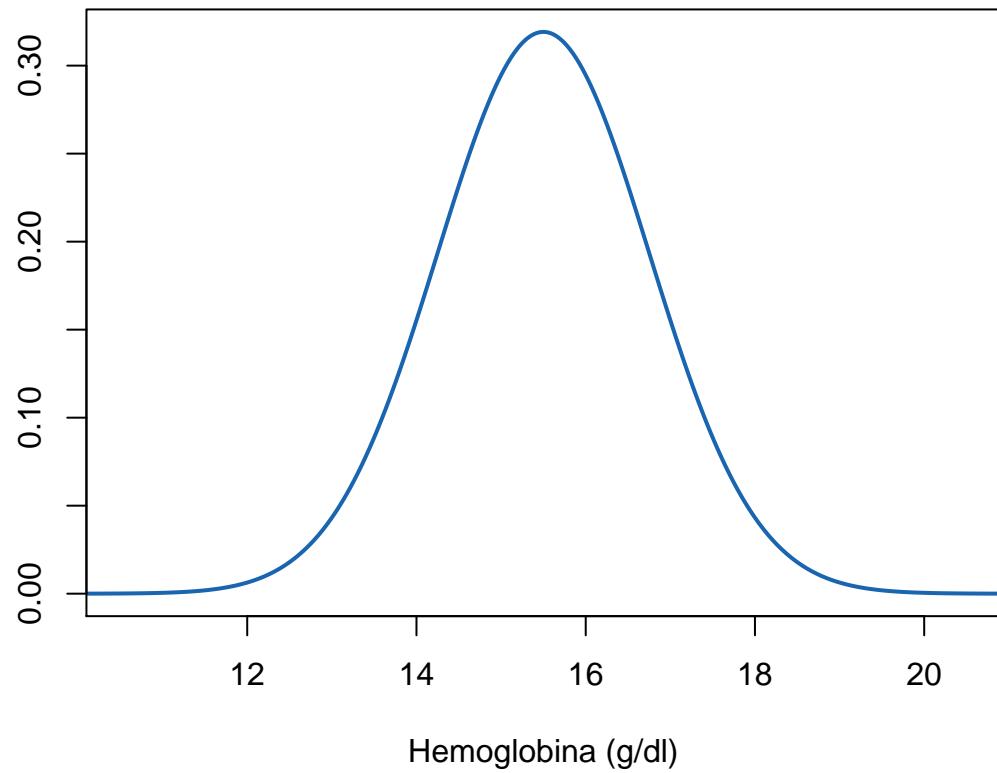
```

$$N(0.95, 0.125)$$

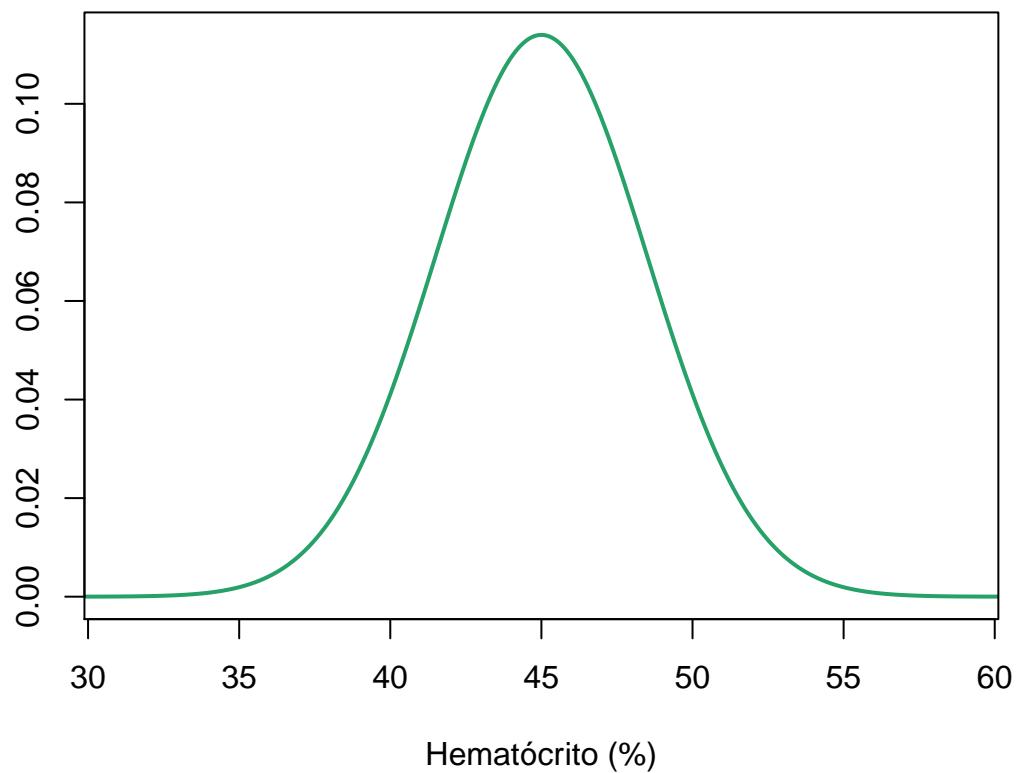


... com o mesmo código R, geramos:

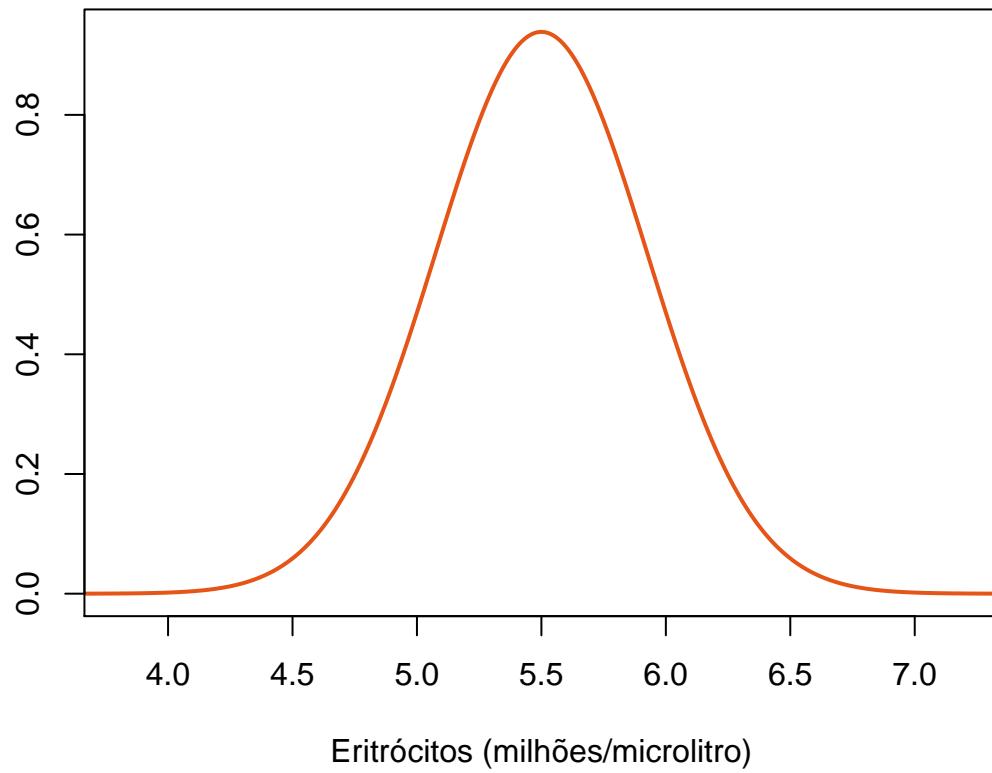
**N(15.5,1.25)**



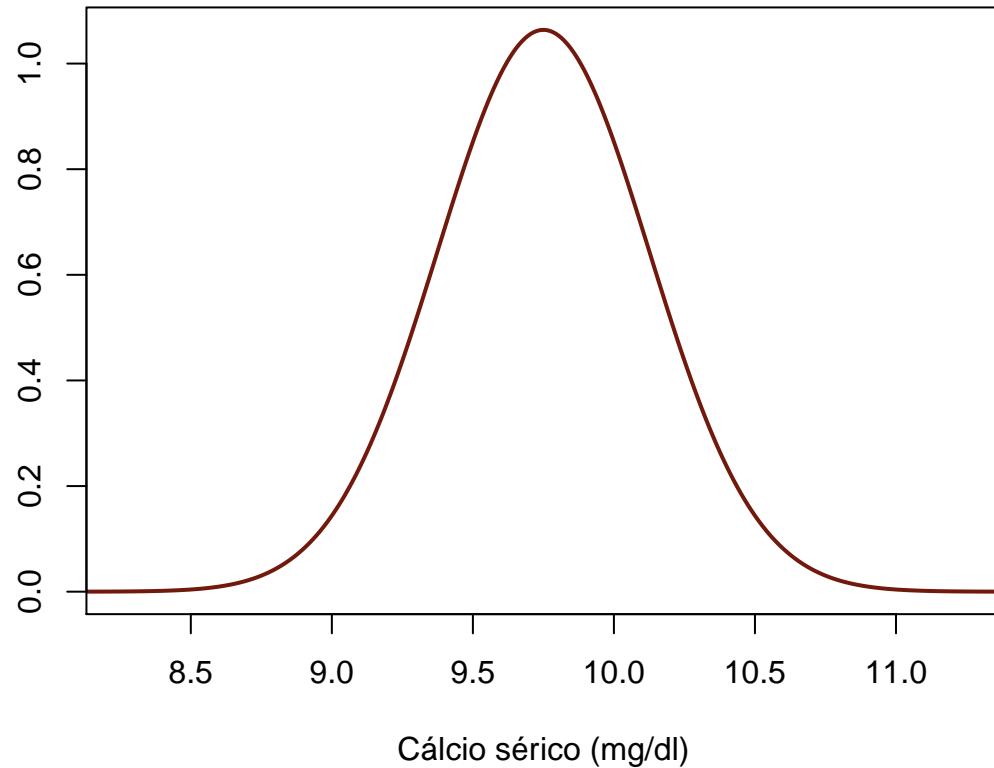
**N(45,3.5)**



$$N(5.5, 0.425)$$

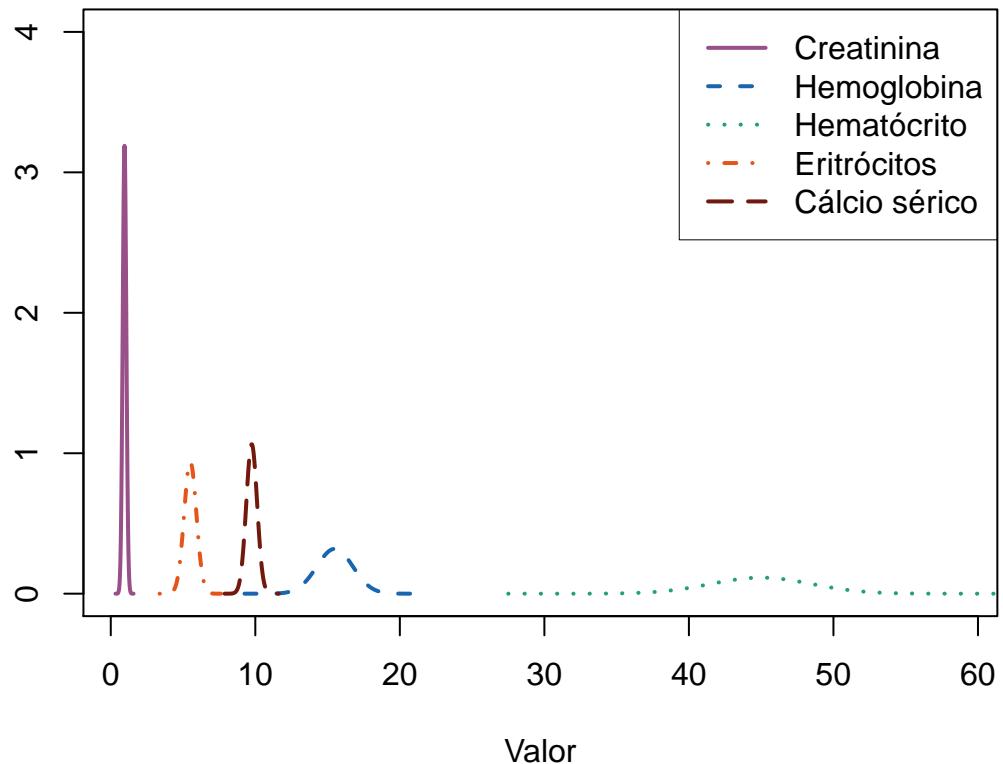


$$N(9.75, 0.375)$$



Parecem todas iguais, mas observe na mesma escala (desconsiderando as unidades de medida):

## Distribuições normais não padronizadas



## Distribuição normal padronizada

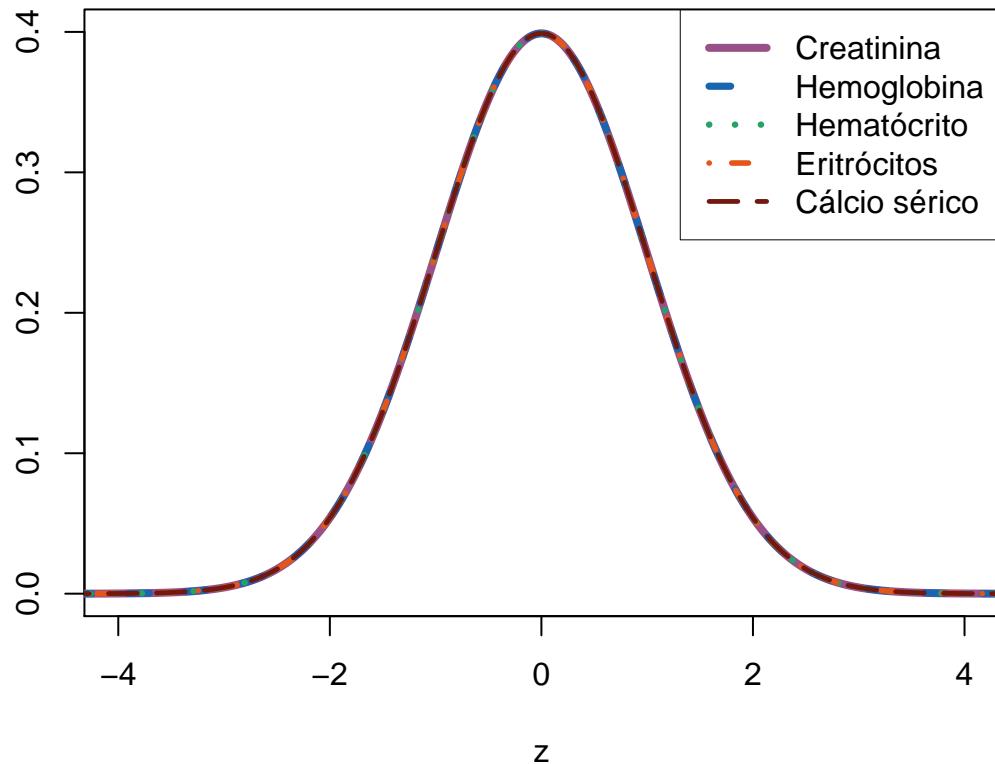
Para padronizar **qualquer** distribuição, basta aplicar a todos os seus valores  $x$ :

$$z = \frac{x - \mu}{\sigma}$$

Subtrair a média faz com que a distribuição fique centrada em 0 e dividir por  $\sigma$  faz com que o desvio-padrão seja igual a 1. A distribuição resultante é dada em escore  $z$ .

Caso normalizemos todas as curvas acima, obteremos:

## Distribuições normais padronizadas



A vantagem é que conhecemos as propriedades de qualquer distribuição normal, mas da normal padronizada memorizamos facilmente seus principais valores.



A função R que (dado o valor  $q$ , devolve a probabilidade) é:

**pnorm(q, mean = 0, sd = 1, lower.tail = TRUE)**

que calcula (por *default*) as probabilidades acumuladas de  $-\infty$  (lower.tail=TRUE) ao valor  $q$  solicitado. Assume, também por *default*, média igual a zero (mean=0) e desvio-padrão (em inglês, *standard deviation*) igual a 1 (sd=1), portanto uma normal padronizada.

Não é necessário converter tudo para a normal padronizada quando quiser encontrar quaisquer as probabilidades.

No exemplo, com média igual a 15 e desvio-padrão igual a 8, encontramos as áreas, respectivamente, para  $\pm 1sd$ :

```
2*(pnorm(15+8, mean=15, sd=8)-0.5)
```

```
## [1] 0.6826895
```

ou, na versão da normal padronizada (que serve para simplificar):

```
2*(pnorm(1)-0.5)
```

```
## [1] 0.6826895
```

a distribuição normal é simétrica, então achamos a probabilidade acumulada de  $-\infty$  até 1 desvio-padrão, subtraímos a metade esquerda da distribuição (encontrando a área entre 0 e 1 desvio-padrão) e, então, multiplicamos por 2.

Similarmente, para  $\pm 2sd$ :

```
2*(pnorm(2)-0.5)
```

```
## [1] 0.9544997
```

e para  $\pm 3sd$ :

```
2*(pnorm(3)-0.5)
```

```
## [1] 0.9973002
```

Note que, entre  $\pm 2dp$ , não temos exatamente 95% da área. Para achar o valor correto, a função R que faz o reverso de **pnorm** (dada a probabilidade, devolve o valor **q**) é:

```
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE)
```

então obtemos (deixando 2.5% em cada cauda):

```
qnorm(0.025)
```

```
## [1] -1.959964
```

e

```
qnorm(0.975)
```

```
## [1] 1.959964
```

Então, 95% da área fica, aproximadamente, no intervalo dado por  $\pm 1.96dp$ .

---

## Criando distribuições normais em R

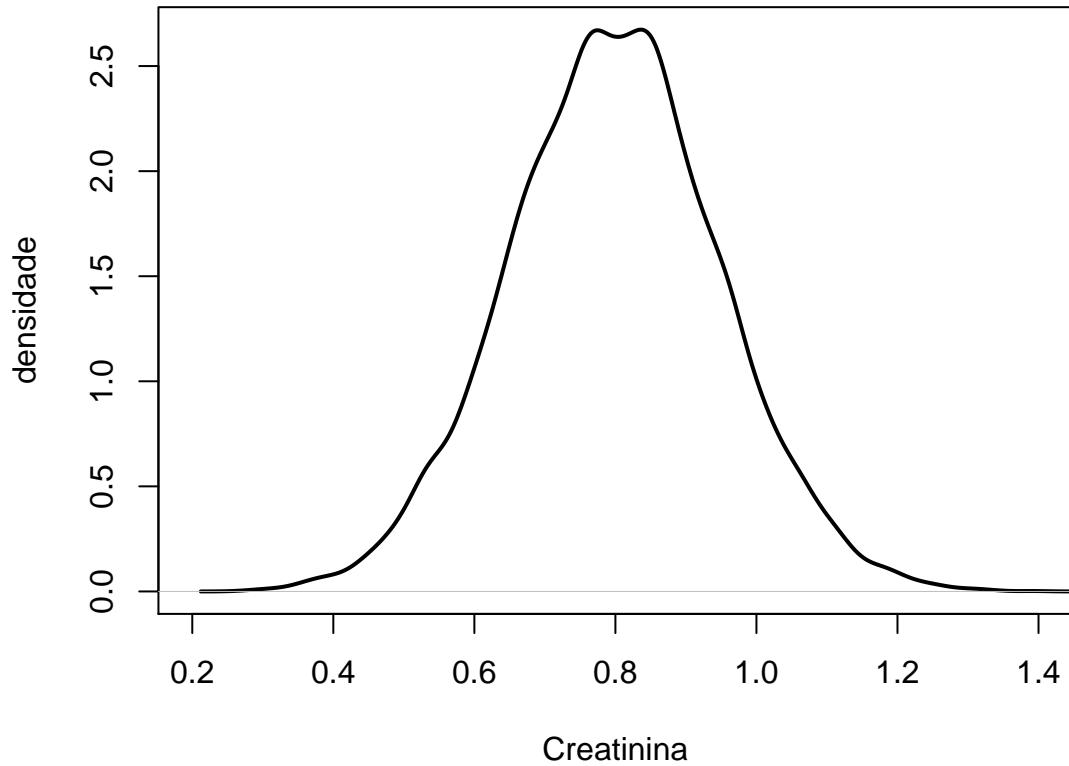
Para mulheres o valor de referência da creatinina sérica é de 0.5 a 1.1 mg/dl. Assumiremos, então, distribuição aproximadamente normal com média de 0.8 mg/dl e desvio-padrão de 0.15 mg/dl.

O seguinte código usa a função **rnorm()** para criar, usando um gerador de números pseudo-aleatórios (**random**), uma população com 10000 indivíduos:

```
# numero de individuos
n <- 10000
# assumindo que a distribuicao eh simetrica%
mu <- 0.8 # (0.5+1.1)/2
# assumindo que deram o intervalo de 95%
dp <- 0.15 # (1.1-0.5)/4
# cria a populacao e exibe o grafico
v <- rnorm(n,mu,dp)
d <- density(v)
plot (d, type="l",
      main="Distribuicao ficticia",
      xlim=c(mu-4*dp,mu+4*dp),
```

```
xlab="Creatinina",
ylab="densidade",
lty=1, lwd=2)
```

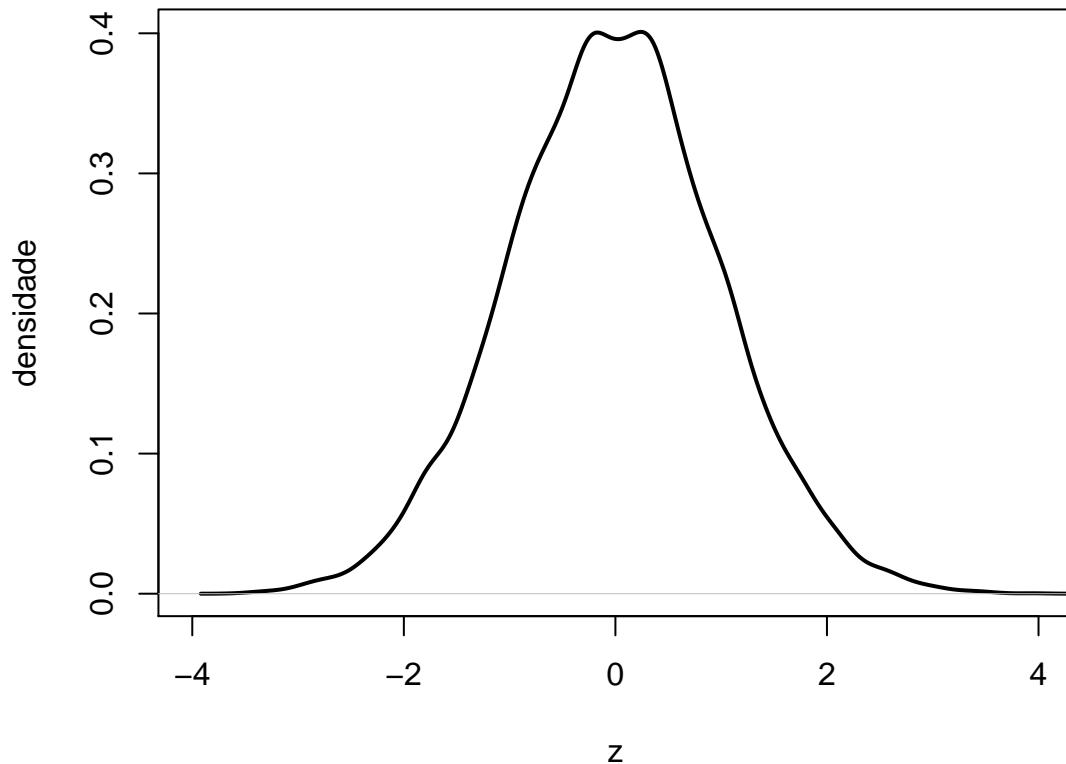
## Distribuição fictícia



e o seguinte código exibe a distribuição padronizada

```
v2 <- (v-mu)/dp
d2<- density(v2)
plot (d2, type="l",
      main="Distribuição fictícia padronizada",
      xlim=c(-4, 4),
      xlab="z",
      ylab="densidade",
      lty=1, lwd=2)
```

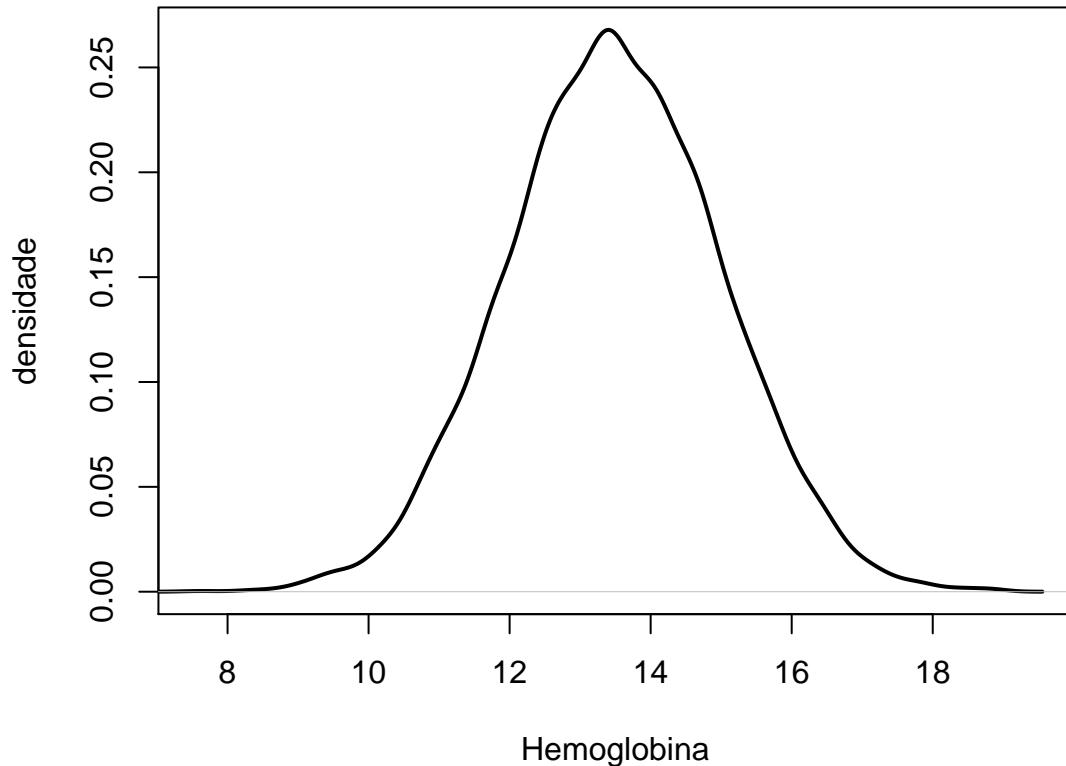
## Distribuição fictícia padronizada



Da mesma forma, assumindo-se que o valor de referência para hemoglobina em mulheres é de  $13.5 \pm 1.5g/dl$ , correspondendo estes valores à média  $\pm 2$  desvio-padrão, e que a distribuição da hemoglobina segue aproximadamente uma distribuição normal, criamos:

```
# numero de individuos
n <- 10000
# media
mu <- 13.5
# desvio-padrao
dp <- 1.5
# cria a populacao e exibe o grafico
v <- rnorm(n,mu,dp)
d <- density(v)
plot (d, type="l",
      main="Distribuicao ficticia",
      xlim=c(mu-4*dp,mu+4*dp),
      xlab="Hemoglobina",
      ylab="densidade",
      lty=1, lwd=2)
```

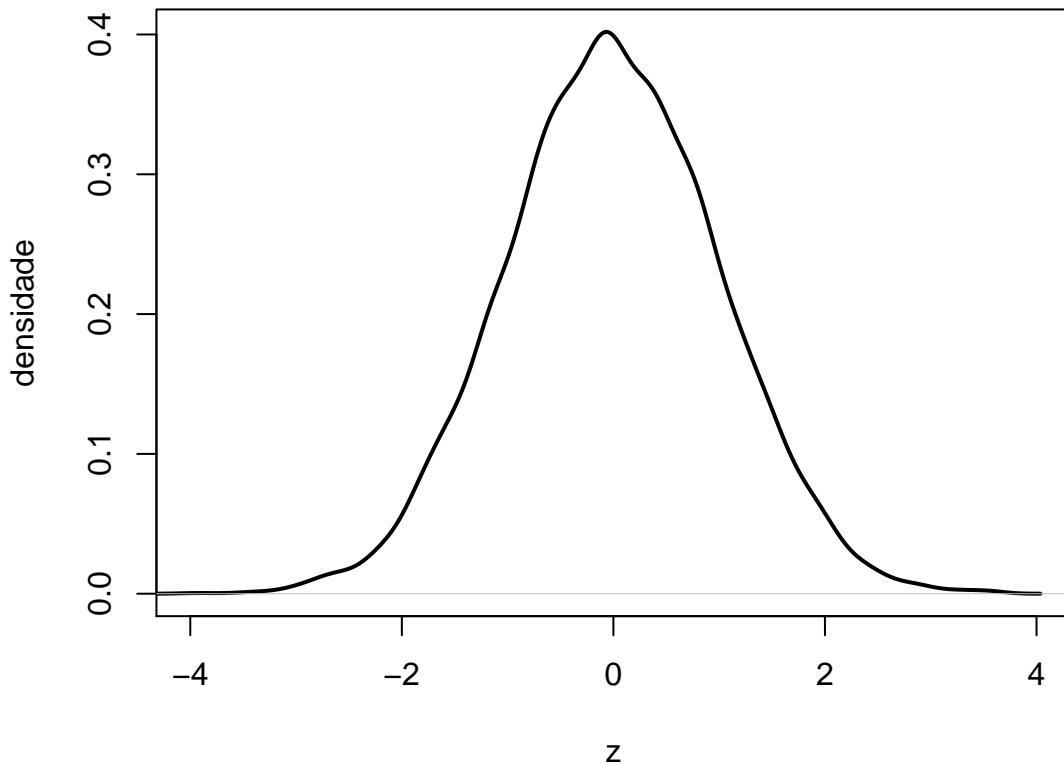
## Distribuicao ficticia



a qual é igualmente padronizada:

```
v2 <- (v-mu)/dp
d2<- density(v2)
plot (d2, type="l",
      main="Distribuicao ficticia padronizada",
      xlim=c(-4, 4),
      xlab="z",
      ylab="densidade",
      lty=1, lwd=2)
```

## Distribuição fictícia padronizada



## TCL e EPM

O Teorema Central do Limite (TCL) é uma das descobertas mais poderosas para as análises estatísticas. Para experimentar com ele, vamos criar uma população com uma variável fictícia que **não** tem distribuição normal:

A íntegra do *Rscript* que será desenvolvido adiante está em *Bootstrapping\_populacaoXamostra.R*

```
# Cria populacao ficticia
source ("friendlycolor.R")

# Uma variavel qualquer
N <- 1000000 # tamanho da populacao
mu1 <- 80 # media de 2/3 da populacao
sigma1 <- 30 # desvio padrao de 2/3 da populacao
mu2 <- 180 # media de 1/3 da populacao
sigma2 <- 50 # desvio padrao de 2/3 da populacao
# set.seed(123)

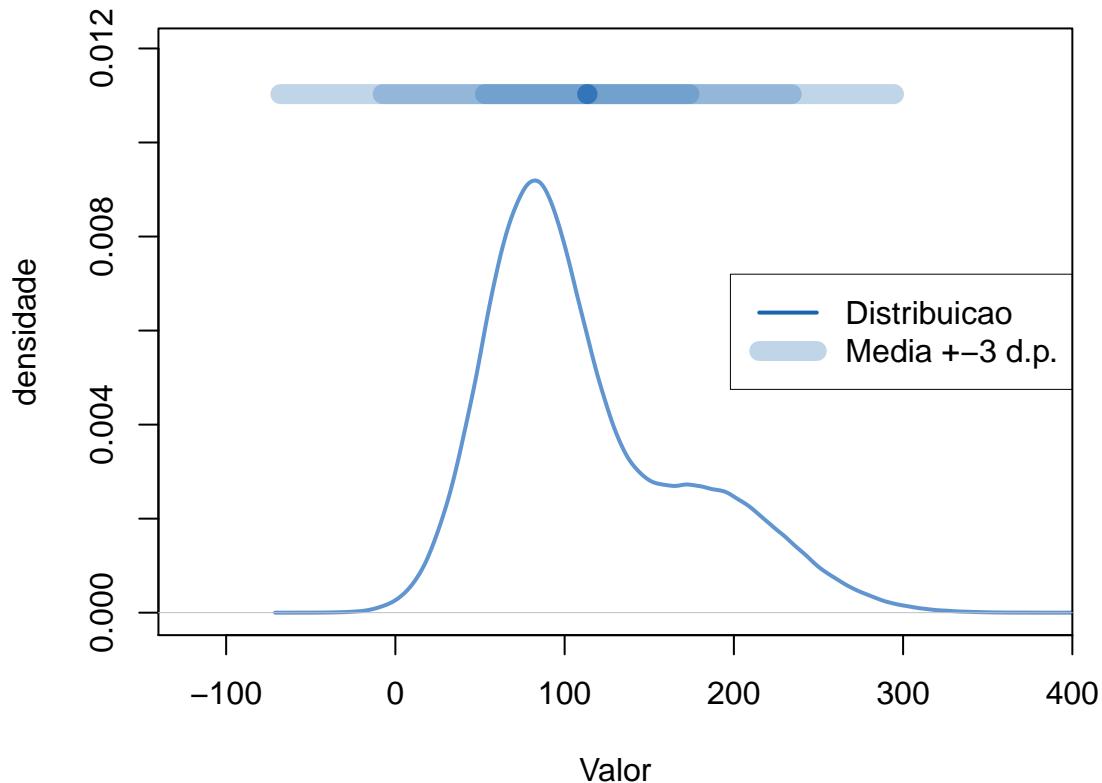
# criando uma populacao ficticia
pop_valores <- round(rnorm(2*N/3, mean=mu1, sd=sigma1),0)
pop_valores <- c(pop_valores,
```

```

        round(rnorm(1*N/3, mean=mu2, sd=sigma2),0))
mean_pop <- mean(pop_valores)
sd_pop <- sd(pop_valores)
# distribuicao dos valores nesta populacao ficticia
dpop_valores <- density(pop_valores)
plot (dpop_valores, main="Populacao ficticia",
      xlab = "Valor", ylab = "densidade",
      xlim = c(min(mu1,mu2)-4*max(sigma1,sigma2),
                max(mu1,mu2)+4*max(sigma1,sigma2)),
      ylim = c(0,max(dpop_valores$y)*1.3),
      col = friendlycolor(10),
      lwd=2, type = "l")
tp <- 44
for (i in -3:3)
{
  lines(c(mean_pop-i*sd_pop, mean_pop),
        c(max(dpop_valores$y)*1.2,max(dpop_valores$y)*1.2),
        lwd=10, lty=1, col = paste(friendlycolor(8),tp,sep=""))
}
legend("right",
       c("Distribuicao", "Media +-3 d.p."),
       col=c(friendlycolor(8),paste(friendlycolor(8),"44",sep="")),
       lwd=c(2,10),
       lty=c(1,1),
       box.lwd=0, bg="transparent")

```

## População fictícia



### Amostragem (*sampling*)

Retiraremos desta população  $B$  amostras com  $n$  indivíduos. As amostras aparecem em laranja.

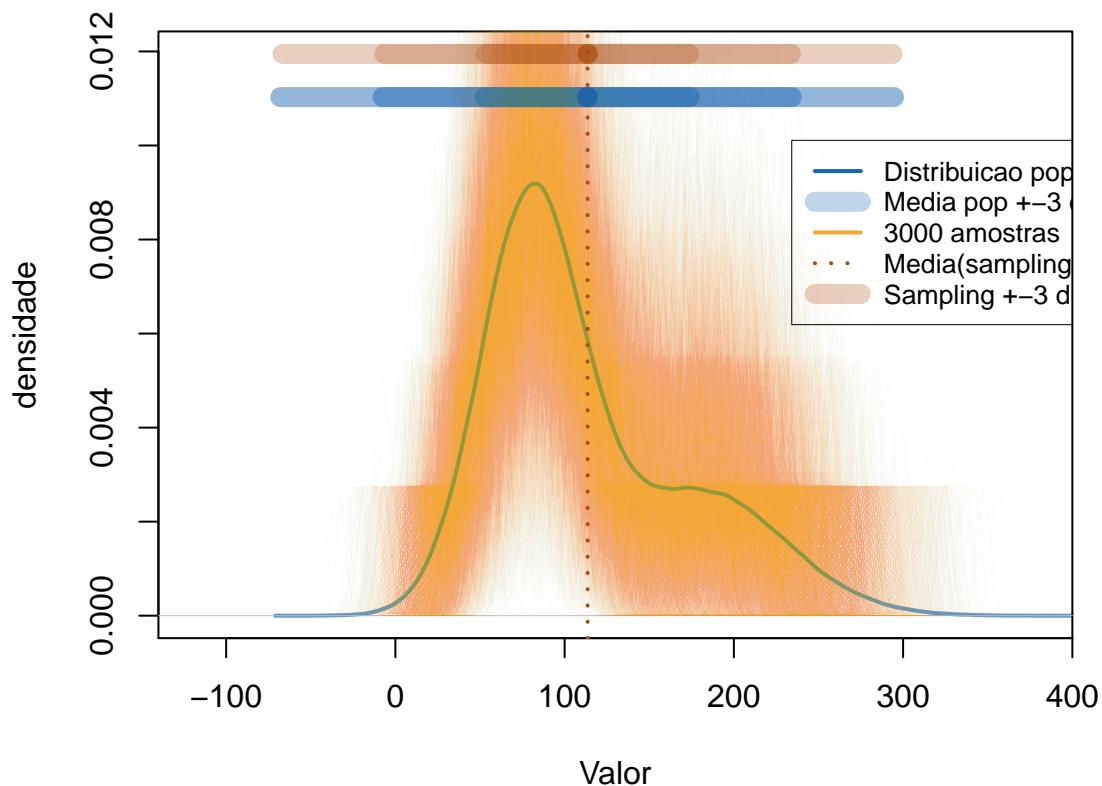
```
B <- 3000
n <- 36
# represesta a populacao
plot (dpop_valores, main=paste("Amostragem (",B," amostras com n = ",n,")",sep=""),
      xlab = "Valor", ylab = "densidade",
      xlim = c(min(mu1,mu2)-4*max(sigma1,sigma2),
                max(mu1,mu2)+4*max(sigma1,sigma2)),
      ylim = c(0,max(dpop_valores$y)*1.3),
      col = friendlycolor(10),
      lwd=2, type = "l")
# plota media e dp populacional
tp <- 44
for (i in -3:3)
{
  lines(c(mean_pop-i*sd_pop, mean_pop),
        c(max(dpop_valores$y)*1.2,max(dpop_valores$y)*1.2),
        lwd=10, lty=1, col = paste(friendlycolor(8),tp,sep=""))
}
```

```

amo_med <- c() # guardando as medias amostras
amo_sd <- c() # guardando os d.p. amostras
for (a in 1:B)
{
  amostra <- sample(pop_valores, n, replace=FALSE)
  amo_med <- c(amo_med,mean(amostra))
  amo_sd <- c(amo_sd,sd(amostra))
  amo_dens <- density(amostra, bw = 4)
  lines(amo_dens, col=paste(friendlycolor(22),"04",sep=""), lwd=0.4)
}
mean_amo <- mean(amo_med)
sd_amo <- mean(amo_sd)
abline(v=mean_amo, lwd=2, lty=3, col=friendlycolor(19))
tp <- 44
for (i in -3:3)
{
  lines(c(mean_pop-i*sd_pop, mean_pop),
        c(max(dpop_valores$y)*1.2,max(dpop_valores$y)*1.2),
        lwd=10, lty=1, col = paste(friendlycolor(8),tp,sep="") )
}
tp <- 44
for (i in -3:3)
{
  lines(c(mean_amo-i*sd_amo, mean_amo),
        c(max(dpop_valores$y)*1.3,max(dpop_valores$y)*1.3),
        lwd=10, lty=1, col = paste(friendlycolor(19),tp,sep="") )
}
legend(x=mean_pop+2*sd_pop, y=max(dpop_valores$y)*1.1,
       c("Distribuicao pop.",
         "Media pop +-3 d.p.",
         paste(B,"amostras"),
         "Media(sampling)",
         "Sampling +-3 d.p."
       ),
       col=c(
         friendlycolor(8),
         paste(friendlycolor(8),"44",sep=""),
         friendlycolor(22),
         friendlycolor(19),
         paste(friendlycolor(19),"44",sep=""))
       ),
       lwd=c(2,10,2,2,10),
       lty=c(1,1,1,3,1),
       box.lwd=0, bg="transparent",
       cex=0.8)

```

## Amostragem (3000 amostras com $n = 36$ )



A linha pontilhada vertical em vermelho corresponde à media das medias amostrais. Na parte alta do grafico a barra horizontal em vermelho mostra a média das médias amostrais e a média dos desvios-padrão amostrais.

Repare que a média das médias amostrais coincide com a média populacional e que a média dos desvios-padrão amostrais coincide com o d.p. populacional.

```
v <- ""
v <- paste(v, "Populacao:\n")
v <- paste(v, "\tmedia populacional:", round(mean_pop,3), "\n")
v <- paste(v, "\td.p. populacional:", round(sd_pop,3), "\n")
v <- paste(v, "\n")
v <- paste(v, "Amostras:", B, "com n =", n, "\n")
v <- paste(v, "\tmedia das medias amostrais:", round(mean_amo,3), "\n")
v <- paste(v, "\tmedia dos d.p. amostrais:", round(sd_amo,3))
cat(v)
```

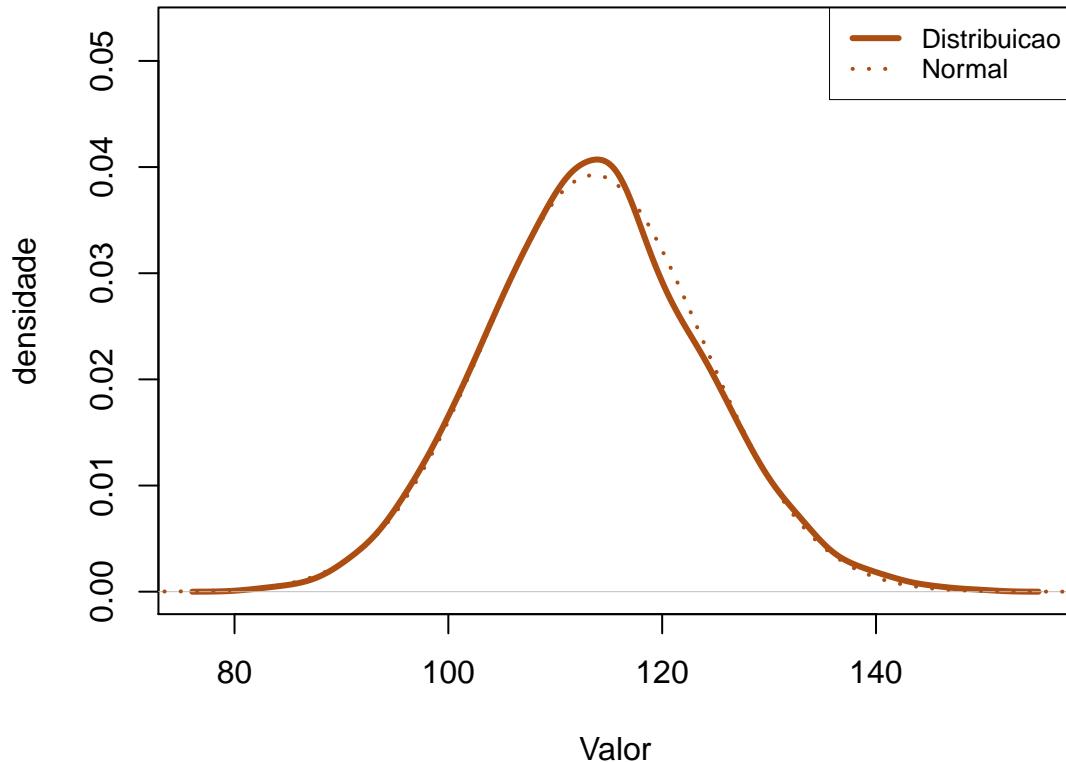
```
## Populacao:
##      media populacional: 113.33
##      d.p. populacional: 60.436
##
## Amostras: 3000 com n = 36
##      media das medias amostrais: 113.581
##      media dos d.p. amostrais: 60.089
```

## distribuição das médias amostrais e EPM

O próximo gráfico mostra a distribuição das médias amostrais.

```
damo_med <- density(amo_med)
plot (damo_med, main=paste("Distribuicao de Medias Amostrais\n(",B," amostras com n = ",n,")",sep=""),
      xlab = "Valor", ylab = "densidade",
      ylim = c(0, max(damo_med$y)*1.3),
      col = friendlycolor(19),
      lwd=3, type = "l")
sd_epmamostral <- sd(amo_med)
minx <- mean_amo-5*sd_epmamostral
maxx <- mean_amo+5*sd_epmamostral
byx <- (maxx-minx)/100
x_normal <- seq(from=minx, to=maxx, by=byx)
y_normal <- dnorm(x_normal, mean=mean_amo, sd=sd_epmamostral)
lines(x_normal,y_normal, lwd=2, lty=3, col = friendlycolor(19))
legend("topright",
       c("Distribuicao",
         "Normal"
       ),
       col=c(
         friendlycolor(19),
         friendlycolor(19)
       ),
       lwd=c(3,2),
       lty=c(1,3),
       box.lwd=0, bg="transparent",
       cex=0.8)
```

## Distribuicao de Medias Amostrais (3000 amostras com n = 36)



O TCL estabelece que a distribuição das médias amostrais, independentemente da forma da distribuição da população original tem distribuição NORMAL com média igual à média da população e com certo desvio-padrão.

A linha pontilhada é uma distribuição normal sobreposta para mostrar que a distribuição simulada (linha sólida) é aproximadamente normal **apesar** da distribuição da variável original não ter este tipo de distribuição.

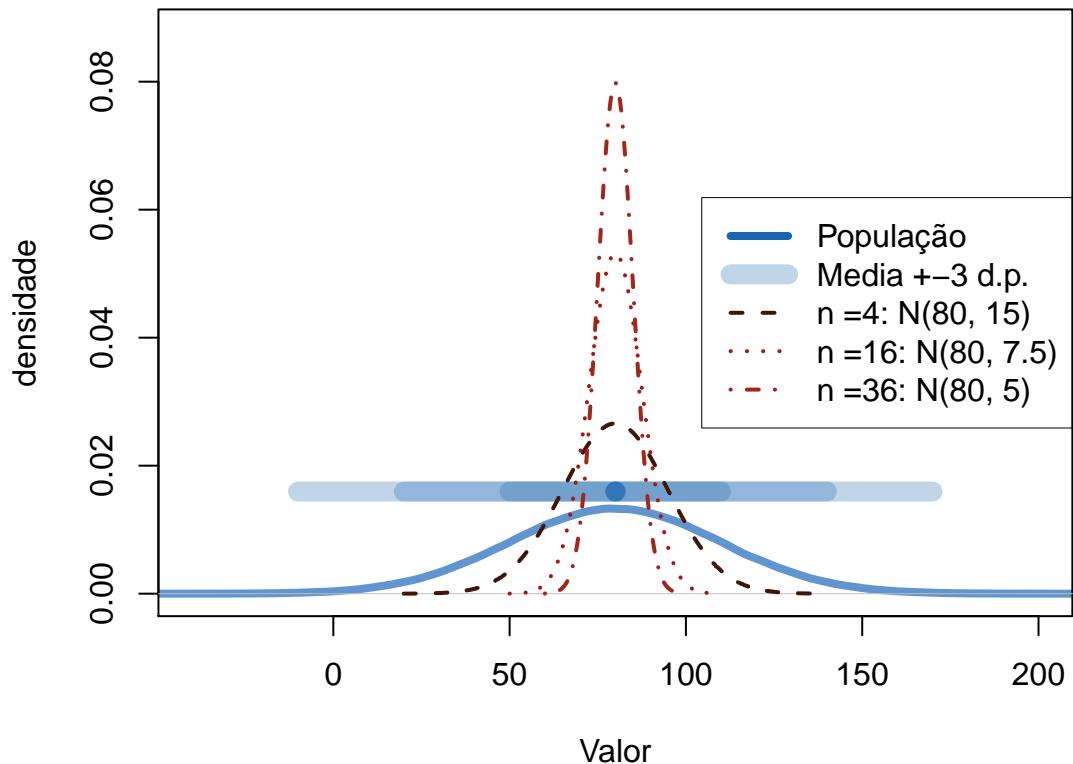
O desvio-padrão das médias amostrais recebe o nome de Erro Padrão das Médias Amostrais (EPM), o qual (em teoria) é estimado pelo desvio-padrão da população dividido pela raiz quadrada do tamanho (n) de cada amostra:

$$EPM = \frac{\sigma}{\sqrt{n}}$$



Você pode executar o *Rscript Populacao\_e\_EPM.R*

## População fictícia e amostras média = 80, d.p. = 30



Observe o comportamento do EPM quando aumenta o tamanho da amostra.

---

### EPM na simulação com 3000 amostras

Como as amostras tiveram tamanho de 36, o EPM é 6 vezes menor que o desvio-padrão populacional (ou o desvio-padrão médio das 3000 amostras)

```
v <- ""
v <- paste(v, "\tmedia dos d.p. amostrais:", round(sd_amo, 3), "\n")
v <- paste(v, "\td.p. das médias amostrais (EPM):", round(sd_epmamostral, 3), "\n")
cat(v)

##      media dos d.p. amostrais: 60.089
##      d.p. das médias amostrais (EPM): 10.157
```

### Reamostragem (*bootstrapping*), saindo da fantasia

Ninguem faz  $B$  amostras de uma população. Na prática somente uma amostra é obtida.

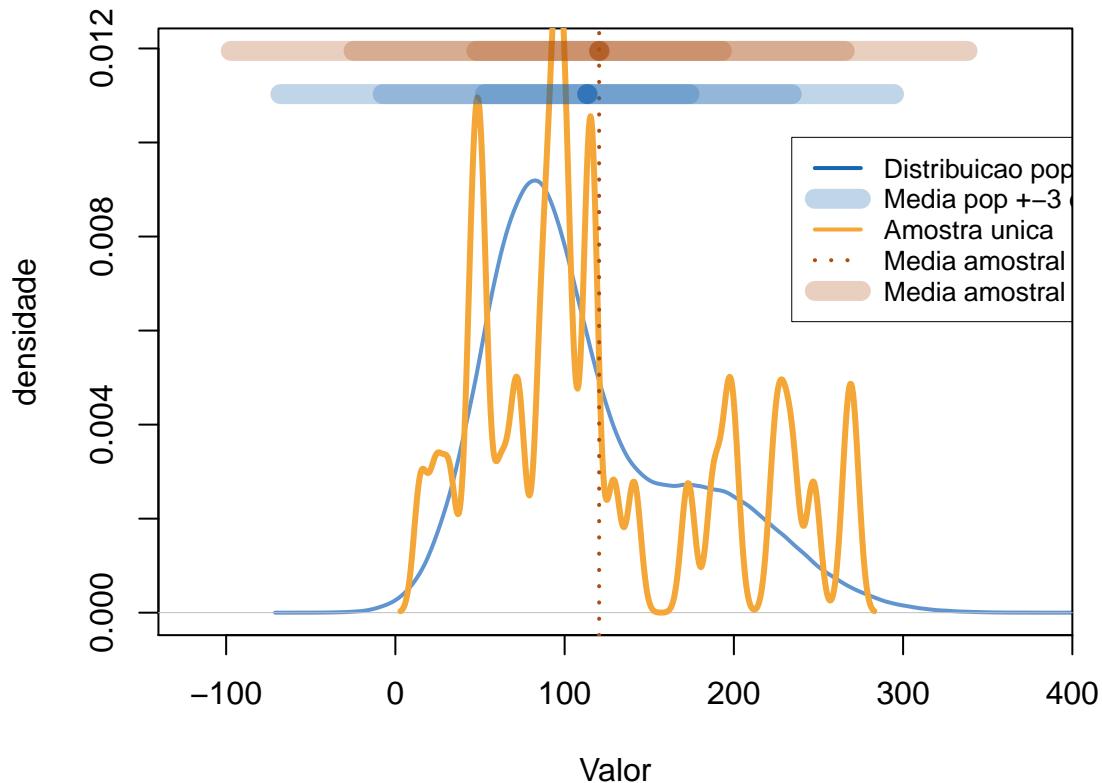
```
# repreSENTA a populacAO
plot (dpop_valores, main=paste("População e amostra unica com n = ",n,sep=""),
      xlab = "Valor", ylab = "densidade",
```

```

xlim = c(min(mu1,mu2)-4*max(sigma1,sigma2),
         max(mu1,mu2)+4*max(sigma1,sigma2)),
ylim = c(0,max(dpop_valores$y)*1.3),
col = friendlycolor(10),
lwd=2, type = "l")
# amostra unica
amostra_unica <- sample(pop_valores, n, replace=FALSE)
mean_amouni <- mean(amostra_unica)
sd_amouni <- sd(amostra_unica)
amouni_dens <- density(amostra_unica, bw = 4)
lines(amouni_dens, col= paste(friendlycolor(22),sep=""), lwd=3)
abline(v=mean_amouni, lwd=2, lty=3, col=friendlycolor(19))
tp <- 44
for (i in -3:3)
{
  lines(c(mean_pop-i*sd_pop, mean_pop),
        c(max(dpop_valores$y)*1.2,max(dpop_valores$y)*1.2),
        lwd=10, lty=1, col = paste(friendlycolor(8),tp,sep="") )
}
tp <- 44
for (i in -3:3)
{
  lines(c(mean_amouni-i*sd_amouni, mean_amouni),
        c(max(dpop_valores$y)*1.3,max(dpop_valores$y)*1.3),
        lwd=10, lty=1, col = paste(friendlycolor(19),tp,sep="") )
}
legend(x=mean_pop+2*sd_pop, y=max(dpop_valores$y)*1.1,
       c("Distribuicao pop.",
         "Media pop +-3 d.p.",
         "Amostra unica",
         "Media amostral",
         "Media amostral +-3 d.p."
       ),
       col=c(
         friendlycolor(8),
         paste(friendlycolor(8),"44",sep=""),
         friendlycolor(22),
         friendlycolor(19),
         paste(friendlycolor(19),"44",sep=""))
       ),
       lwd=c(2,10,2,2,10),
       lty=c(1,1,1,3,1),
       box.lwd=0, bg="transparent",
       cex=0.8)

```

## Populacão e amostra única com $n = 36$



A amostra, se não houver problemas, traz propriedades da população:

```
v <- ""
v <- paste(v, "Populacão:\n")
v <- paste(v, "\tmedia populacional:", round(mean_pop, 3), "\n")
v <- paste(v, "\td.p. populacional:", round(sd_pop, 3), "\n")
v <- paste(v, "\n")
v <- paste(v, "Amostra com n =", n, "\n")
v <- paste(v, "\tmedia amostral:", round(mean_amouni, 3), "\n")
v <- paste(v, "\td.p. amostral:", round(sd_amouni, 3), "\n")
cat(v)

## Populacão:
##      media populacional: 113.33
##      d.p. populacional: 60.436
##
##  Amostra com n = 36
##      media amostral: 120.389
##      d.p. amostral: 72.561
```

Na prática, porém, não temos a população como referência e, assim, não fazemos ideia se temos uma amostra representativa:

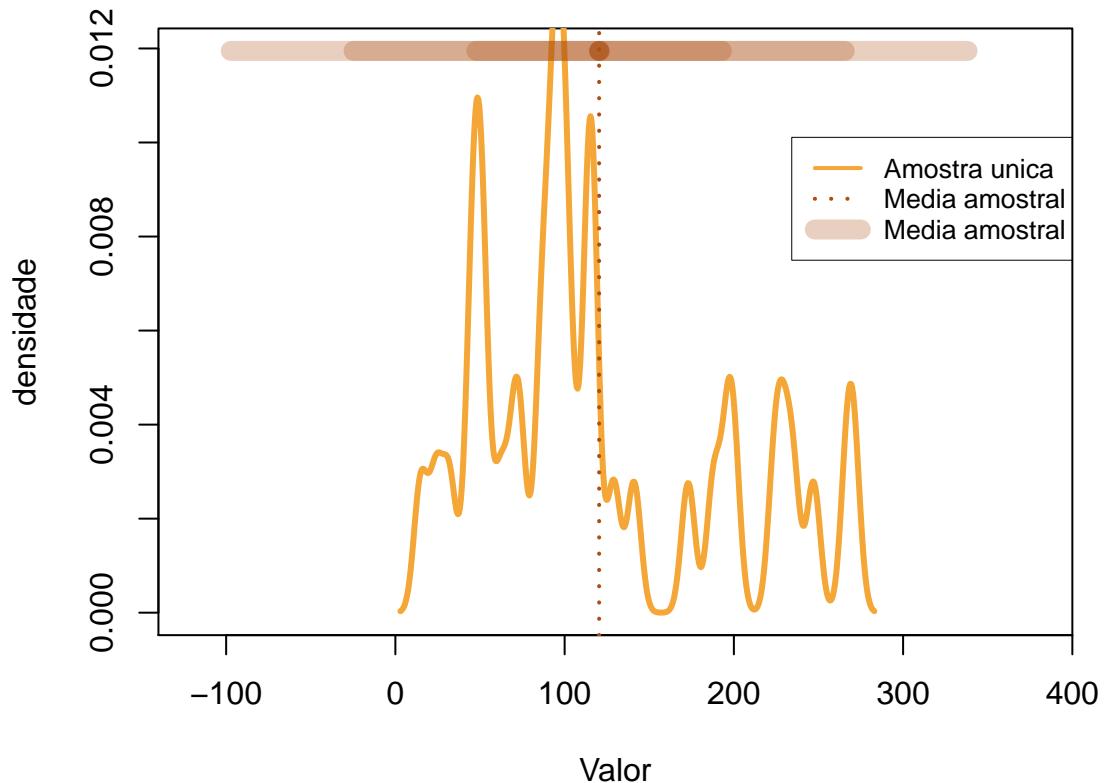
```
# reapresenta sem a população
plot (NA, main=paste("Amostra única com n = ", n, sep=""),
```

```

xlab = "Valor", ylab = "densidade",
xlim = c(min(mu1,mu2)-4*max(sigma1,sigma2),
         max(mu1,mu2)+4*max(sigma1,sigma2)),
ylim = c(0,max(dpop_valores$y)*1.3)
)
# reapresenta a ultima amostra
lines(amouni_dens, col=paste(friendlycolor(22),sep=""), lwd=3)
abline(v=mean_amouni, lwd=2, lty=3, col=friendlycolor(19))
tp <- 44
for (i in -3:3)
{
  lines(c(mean_amouni-i*sd_amouni, mean_amouni),
        c(max(dpop_valores$y)*1.3,max(dpop_valores$y)*1.3),
        lwd=10, lty=1, col = paste(friendlycolor(19),tp,sep="") )
}
legend(x=mean_pop+2*sd_pop, y=max(dpop_valores$y)*1.1,
       c("Amostra unica",
         "Media amostral",
         "Media amostral +-3 d.p."),
       col=c(
         friendlycolor(22),
         friendlycolor(19),
         paste(friendlycolor(19),"44",sep=""))
       ),
       lwd=c(2,2,10),
       lty=c(1,3,1),
       box.lwd=0, bg="transparent",
       cex=0.8)

```

## Amostra unica com $n = 36$



O que podemos fazer é um processo similar ao anterior, fazendo  $B$  reamostragens de  $n$  elementos. Isto parece inútil, pois sempre teremos a própria amostra:

```

amostra <- c(1, 2, 3, 4, 5)
n <- length(amostra)
cat("amostra: ", amostra,
    ": média =", round(mean(amostra),2),
    "e dp =", round(sd(amostra),2),
    "\n")

## amostra: 1 2 3 4 5 : média = 3 e dp = 1.58
# 9 reamostragens sem reposicao
for (r in 1:9)
{
  # *** replace=FALSE por default ***
  reamostra <- sample(amostra,n)
  cat("reamostra",r,": ", reamostra,
      ": média =", round(mean(reamostra),2),
      "e dp =", round(sd(reamostra),2),
      "\n")
}

## reamostra 1 : 5 1 3 4 2 : média = 3 e dp = 1.58
## reamostra 2 : 1 2 4 3 5 : média = 3 e dp = 1.58

```

```

## reamostra 3 : 5 2 3 1 4 : média = 3 e dp = 1.58
## reamostra 4 : 1 2 3 4 5 : média = 3 e dp = 1.58
## reamostra 5 : 4 1 3 2 5 : média = 3 e dp = 1.58
## reamostra 6 : 5 1 3 4 2 : média = 3 e dp = 1.58
## reamostra 7 : 5 2 1 3 4 : média = 3 e dp = 1.58
## reamostra 8 : 5 3 4 2 1 : média = 3 e dp = 1.58
## reamostra 9 : 3 1 2 5 4 : média = 3 e dp = 1.58

```

Então, para obtermos variações de nossa única amostra, fazemos a reamostragem COM REPOSICAO:

```

# 9 reamostragens com reposicao
for (r in 1:9)
{
  # *** replace=TRUE permite a reposicao ***
  reamostra <- sample(amostra,n,replace=TRUE)
  cat("reamostra",r,": ", reamostra,
      ": média =", round(mean(reamostra),2),
      "e dp =", round(sd(reamostra),2),
      "\n")
}

## reamostra 1 : 5 5 4 1 4 : média = 3.8 e dp = 1.64
## reamostra 2 : 2 2 4 2 1 : média = 2.2 e dp = 1.1
## reamostra 3 : 4 4 2 2 3 : média = 3 e dp = 1
## reamostra 4 : 4 5 5 5 4 : média = 4.6 e dp = 0.55
## reamostra 5 : 5 3 2 4 5 : média = 3.8 e dp = 1.3
## reamostra 6 : 1 5 3 4 2 : média = 3 e dp = 1.58
## reamostra 7 : 1 4 1 5 1 : média = 2.4 e dp = 1.95
## reamostra 8 : 5 3 4 3 5 : média = 4 e dp = 1
## reamostra 9 : 2 3 2 4 5 : média = 3.2 e dp = 1.3

```

Voltando à população, podemos fazer as  $B$  amostras com  $n$  elementos com reposição:

```

B <- 3000
n <- 36
# representa a amostra
plot (NA, main=paste("Bootstrapping (",B," reamostras com n = ",n,")",sep=""),
      xlab = "Valor", ylab = "densidade",
      xlim = c(min(mu1,mu2)-4*max(sigma1,sigma2),
              max(mu1,mu2)+4*max(sigma1,sigma2)),
      ylim = c(0,max(dpop_valores$y)*1.3),
      col = friendlycolor(22),
      lwd=2, type = "l")
boot_med <- c() # guardando as medias amostras do bootstrapping
boot_sd <- c() # guardando os d.p. amostras do bootstrapping
for (a in 1:B)
{
  amostra <- sample(amostra_unica, n, replace=TRUE)
  boot_med <- c(boot_med,mean(amostra))
  boot_sd <- c(boot_sd,sd(amostra))
  amo_dens <- density(amostra, bw = 4)
  lines(amo_dens, col=paste(friendlycolor(15),"04",sep=""), lwd=0.4)
}
# replota a amostra unica
lines (amouni_dens, lwd=3, col = friendlycolor(22))
mean_boot <- mean(boot_med)

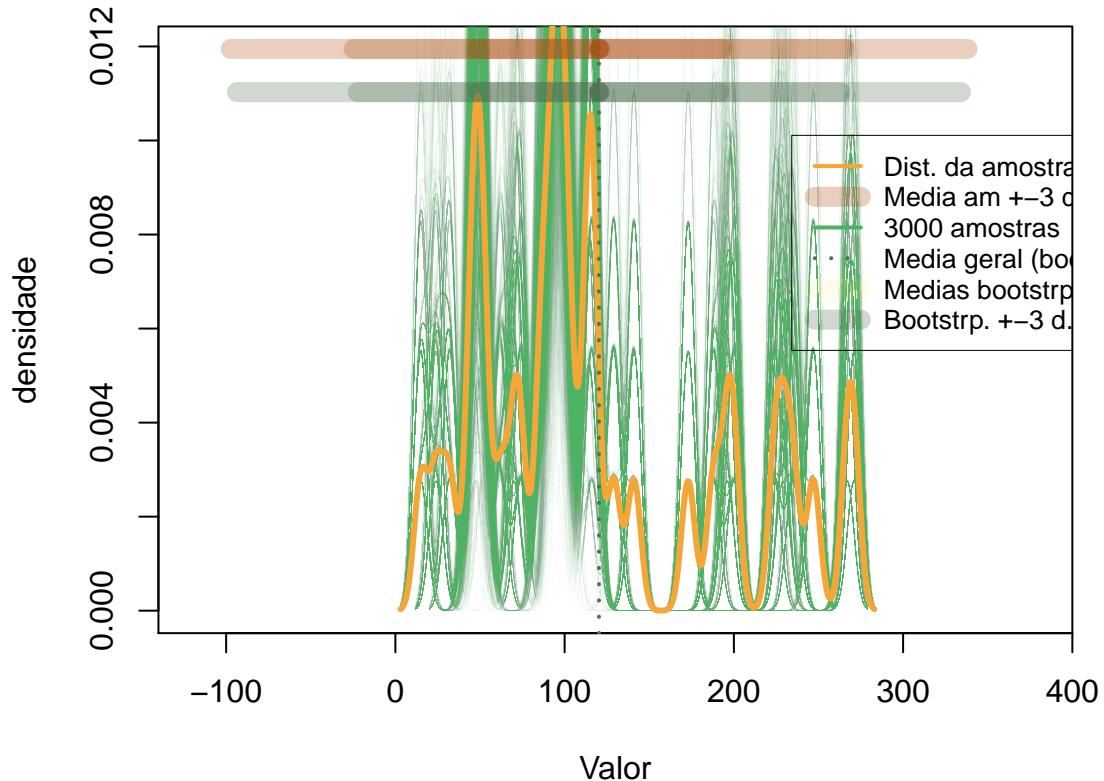
```

```

sd_boot <- mean(boot_sd)
abline(v=mean_boot, lwd=2, lty=3, col=friendlycolor(13))
# replota media e dp da amostra unica
tp <- 44
for (i in -3:3)
{
  lines(c(mean_amouni-i*sd_amouni, mean_amouni),
        c(max(dpop_valores$y)*1.3,max(dpop_valores$y)*1.3),
        lwd=10, lty=1, col = paste(friendlycolor(19),tp,sep=""))
}
# plota media e dp do bootstrapping
tp <- 44
for (i in -3:3)
{
  lines(c(mean_boot-i*sd_boot, mean_boot),
        c(max(dpop_valores$y)*1.2,max(dpop_valores$y)*1.2),
        lwd=10, lty=1, col = paste(friendlycolor(13),tp,sep=""))
}
legend(x=mean_pop+2*sd_pop, y=max(dpop_valores$y)*1.1,
       c("Dist. da amostra",
         "Media am +-3 d.p.",
         paste(B,"amostras"),
         "Media geral (bootstrp.)",
         "Medias bootstrap",
         "Bootstrp. +-3 d.p."
       ),
       col=c(
         friendlycolor(22),
         paste(friendlycolor(19),"44",sep=""),
         friendlycolor(15),
         friendlycolor(13),
         paste(friendlycolor(24),"20",sep=""),
         paste(friendlycolor(13),"44",sep="")
       ),
       lwd=c(2,10,2,2,10,10),
       lty=c(1,1,1,3,1,1),
       box.lwd=0, bg="transparent",
       cex=0.8)

```

## Bootstrapping (3000 reamostras com n = 36)



E verificar que, novamente, a distribuição das médias amostrais é normal:

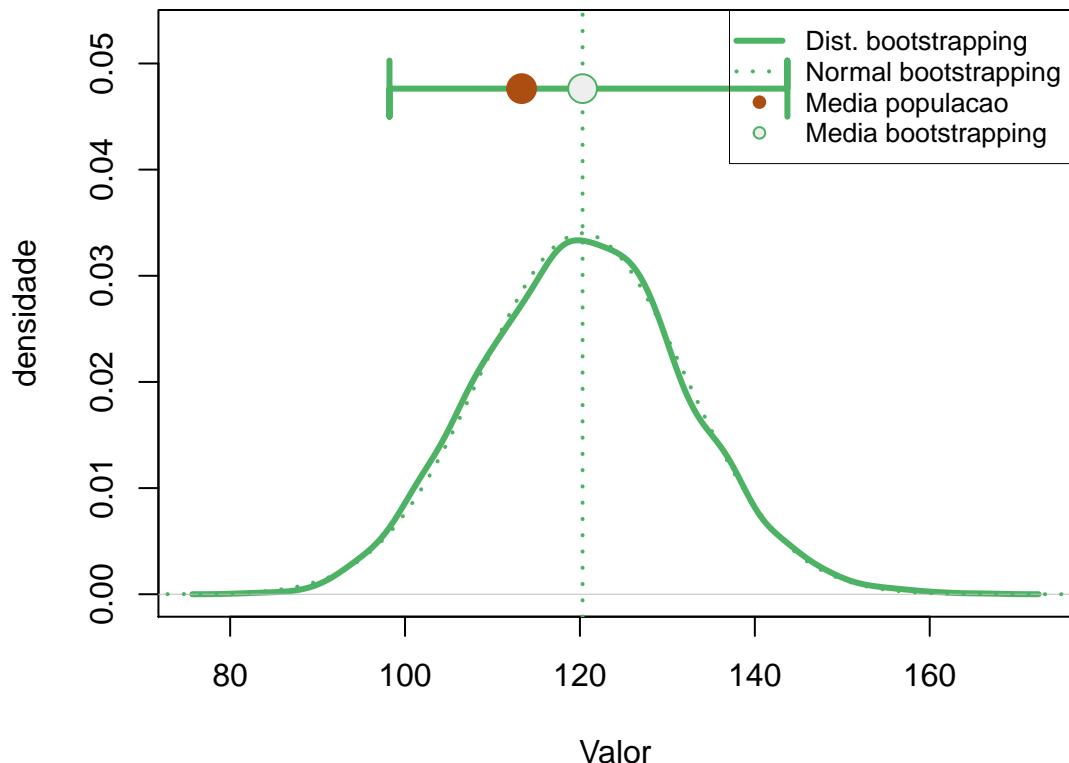
```
# apresenta o EPM obtido por bootstrapping
dboot_med <- density(boot_med)
plot (dboot_med, main=paste("Distribuicao de Medias Amostrais\n",B," amostras com n = ",n,""),sep=""),
      xlab = "Valor", ylab = "densidade",
      ylim = c(0, max(damo_med$y)*1.3),
      col = friendlycolor(15),
      lwd=3, type = "l"
)
abline(v=mean_boot, lwd=2, lty=3, col=friendlycolor(15))
sd_epmboot <- sd(boot_med)
minx <- mean_boot-5*sd_epmboot
maxx <- mean_boot+5*sd_epmboot
byx <- (maxx-minx)/100
x_normal <- seq(from=minx, to=maxx, by=byx)
y_normal <- dnorm(x_normal, mean=mean_boot, sd=sd_epmboot)
lines(x_normal,y_normal, lwd=2, lty=3, col = friendlycolor(15))
icx <- quantile (boot_med, probs=c(0.025,0.975))
icy <- rep(max(damo_med$y)*1.3,3)
icy <- icy * c(0.95,0.9,0.85)
lines(c(icx[1],icx[1],icx[1],icx[2],icx[2],icx[2]),
      c(icy[1],icy[3],icy[2],icy[2],icy[1],icy[3]),
      lwd=3, col = friendlycolor(15))
```

```

points(mean_boot, icy[2], pch=21, bg = friendlycolor(45), col = friendlycolor(15), cex=2)
points(mean_pop, icy[2], pch=21, bg = friendlycolor(19), col = friendlycolor(19), cex=2)
legend("topright",
  c("Dist. bootstrapping",
    "Normal bootstrapping",
    "Media populacao",
    "Media bootstrapping"
  ),
  col=c(
    friendlycolor(15),
    friendlycolor(15),
    friendlycolor(19),
    friendlycolor(15)
  ),
  pt.bg=c(NA,NA,friendlycolor(19),friendlycolor(45)),
  lwd=c(3,2,NA,NA),
  lty=c(1,3,NA,NA),
  pch=c(NA,NA,21,21),
  box.lwd=0, bg="transparent",
  cex=0.8)

```

## Distribuicao de Medias Amostrais (3000 amostras com n = 36)



O valor de EPM, a partir do *bootstrapping*, também é próximo a 1/6 do desvio-padrão populacional.

```

v <- ""
v <- paste(v,"Populacao (que nunca veremos):\n")
v <- paste(v,"\\tmedia populacional:",round(mean_pop,3),"\n")
v <- paste(v,"\\td.p. populacional:",round(sd_pop,3),"\n")
v <- paste(v,"\\n")
v <- paste(v,"Amostra com n =",n,"\n")
v <- paste(v,"\\tmedia amostral:",round(mean_amouni,3),"\n")
v <- paste(v,"\\td.p. amostral:",round(sd_amouni,3),"\n")
v <- paste(v,"Reamostras:",B,"com n =",n,"\n")
v <- paste(v,"\\tmedia das medias amostrais:",round(mean_boot,3),"\n")
v <- paste(v,"\\tmedia dos d.p. amostrais:",round(sd_boot,3),"\n")
v <- paste(v,"EPM por bootstrapping a partir da amostra:\n")
icx <- round(icx,3)
v <- paste(v,"\\tmedia das medias amostrais:",round(mean_boot,3),"[",icx[1],",",icx[2],"]","\n")
v <- paste(v,"\\tEPM (desvio-padrão das medias amostrais):",round(sd_epmboot,3),"\n")
cat(v)

```

```

## Populacao (que nunca veremos):
##     media populacional: 113.33
##     d.p. populacional: 60.436
##
## Amostra com n = 36
##     media amostral: 120.389
##     d.p. amostral: 72.561
## Reamostras: 3000 com n = 36
##     media das medias amostrais: 120.3
##     media dos d.p. amostrais: 71.309
## EPM por bootstrapping a partir da amostra:
##     media das medias amostrais: 120.3 [ 98.221 , 143.724 ]
##     EPM (desvio-padrão das medias amostrais): 11.726

```

Observe, ainda, que a média populacional está dentro do intervalo de confiança de 95% estimado pelo *bootstrapping*.

