



PUC-RIO

INF1036 - Probabilidade Computacional

Material 5 - Variáveis Aleatórias
Professora - Ana Carolina Letichevsky*
2023.1

*Material Adaptado de Professor Hélio Lopes

Uma **variável aleatória** é uma variável quantitativa, cujo resultado (valor) depende de fatores aleatórios.

As variáveis aleatórias podem ser classificadas em:

- **Discreta:** Tem um número finito ou contável de possíveis resultados a serem listados;
- **Contínua:** Tem um número infinito de possíveis resultados representados por um intervalo na reta numérica;
- **Mista:** Pode tanto assumir valores discretos quanto assumir todos os valores em um determinado intervalo.

Diversas situações reais muitas vezes se aproximam de certas **distribuições estocásticas** definidas por algumas hipóteses, daí a importância de se conhecer e manipular algumas destas distribuições tão presentes em nosso cotidiano.

As distribuições de probabilidade podem ser **contínuas** ou **discretas**, dependendo se elas definem probabilidades para **variáveis contínuas** ou **discretas**.

Com uma **distribuição de probabilidade discreta**, cada valor possível da variável aleatória discreta pode ser associado a uma probabilidade diferente de zero. Deste modo, uma distribuição de probabilidade discreta é, por vezes, apresentada em forma de tabela.

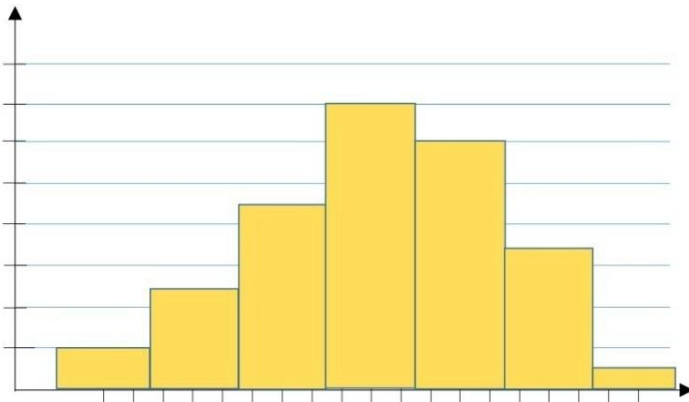
As **probabilidades de variáveis aleatórias contínuas** são definidas como a área sob a curva da sua distribuição. Assim, apenas as faixas de valores podem ter uma probabilidade diferente de zero. A probabilidade de que uma variável aleatória contínua seja igual a algum valor é sempre zero.

Distribuições Discretas e Contínuas

Variável aleatória
discreta



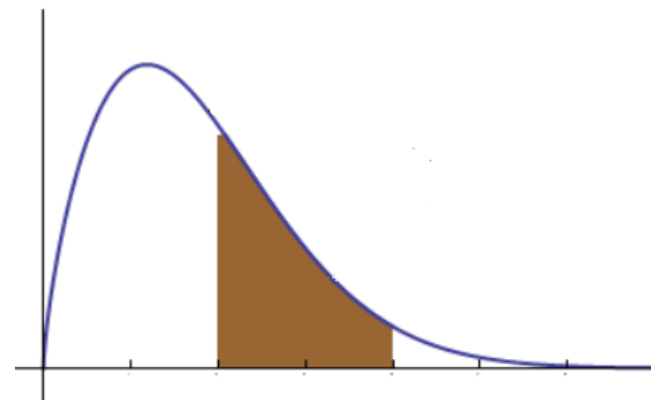
Distribuição de
probabilidade
discreta



Variável aleatória
contínua



Distribuição de
probabilidade
contínua



Distribuições Discretas

Exemplo 1) Uma urna possui 10 bolas sendo 3 com o número 1 e 7 com o número 2.

a) Definimos a variável X como o número de bolas obtidas em uma extração aleatória.

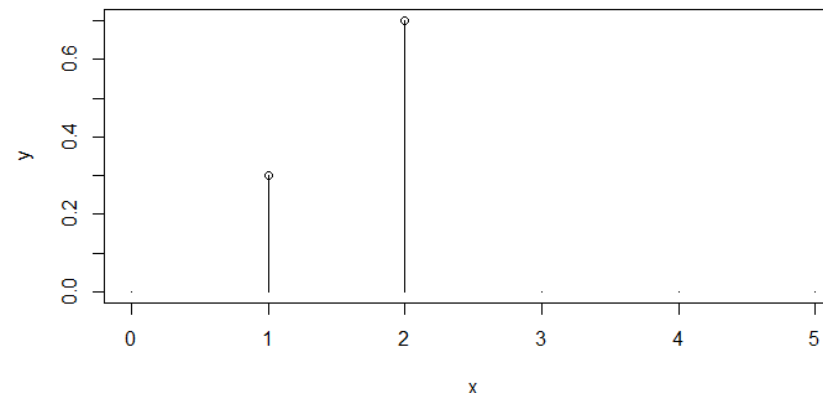
Podemos definir de forma geral a **probabilidade** de ocorrência de um determinado evento como:

$$P(X = x) = f(x)$$

$$P(X = x) = f(x) = \begin{cases} 0,3, & \text{se } x = 1 \\ 0,7, & \text{se } x = 2 \\ 0, & \text{caso contrário} \end{cases}$$

Função de probabilidade.

```
x <- 0:5  
y <- c(0, 0.3, 0.7, 0, 0, 0)  
plot(x, y, type = 'h')  
points(1, 0.3)  
points(2, 0.7)
```



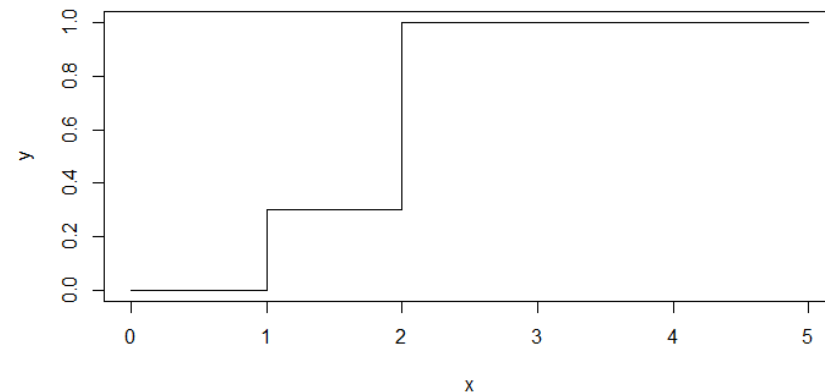
Exemplo 1, continuação) Uma urna possui 10 bolas sendo 3 com o número 1 e 7 com o número 2.

a) Definimos a variável X como o número de bolas obtidas em uma extração aleatória.

Função de probabilidade acumulada:

$$P(X \leq x) = F(x) = \begin{cases} 0, & \text{se } x < 1 \\ 0,3, & \text{se } 1 \leq x < 2 \\ 1, & \text{se } x \geq 2 \end{cases}$$

```
x <- 0:5  
y <- c(0, 0.3, 1, 1, 1, 1)  
plot(x, y, type='s')
```

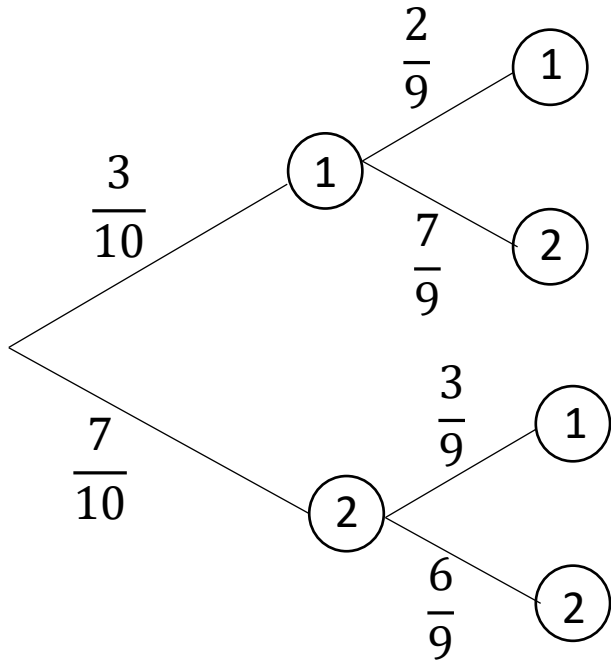


Distribuições Discretas



Exemplo 1 , continuação) Uma urna possui 10 bolas sendo 3 com o número 1 e 7 com o número 2.

b) Definimos a variável Y como o número de bolas com o número 1 obtidas em duas extrações



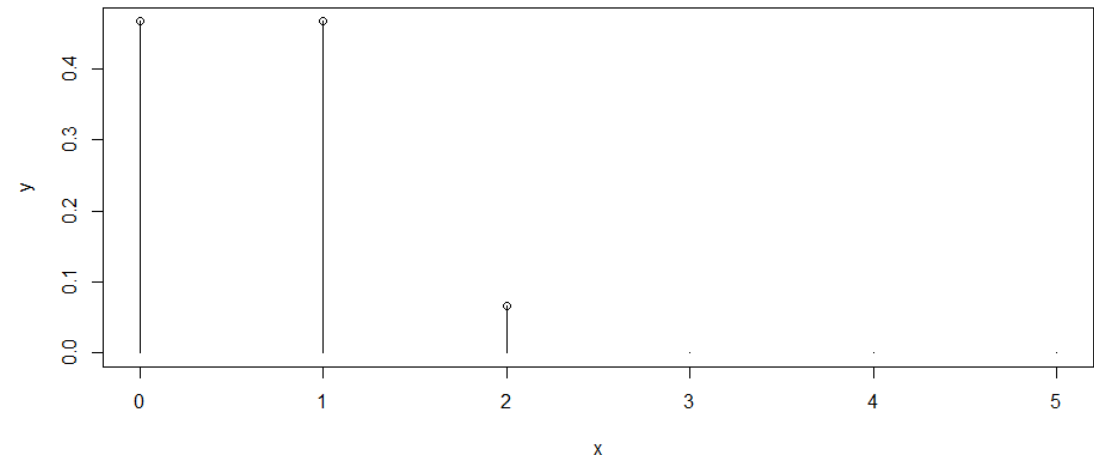
Resultados Possíveis		Probabilidades	Y
1	1	$6/90$	2
1	2	$21/90$	1
2	1	$21/90$	1
2	2	$42/90$	0

Distribuições Discretas

Exemplo 1 , continuação) Uma urna possui 10 bolas sendo 3 com o número 1 e 7 com o número 2.

b) Definimos a variável Y como o número de bolas com o número 1 obtidas em duas extrações.

$$P(Y = y) = f(y) = \begin{cases} \frac{14}{30}, & \text{se } y = 0 \\ \frac{14}{30}, & \text{se } y = 1 \\ \frac{2}{30}, & \text{se } y = 2 \end{cases}$$



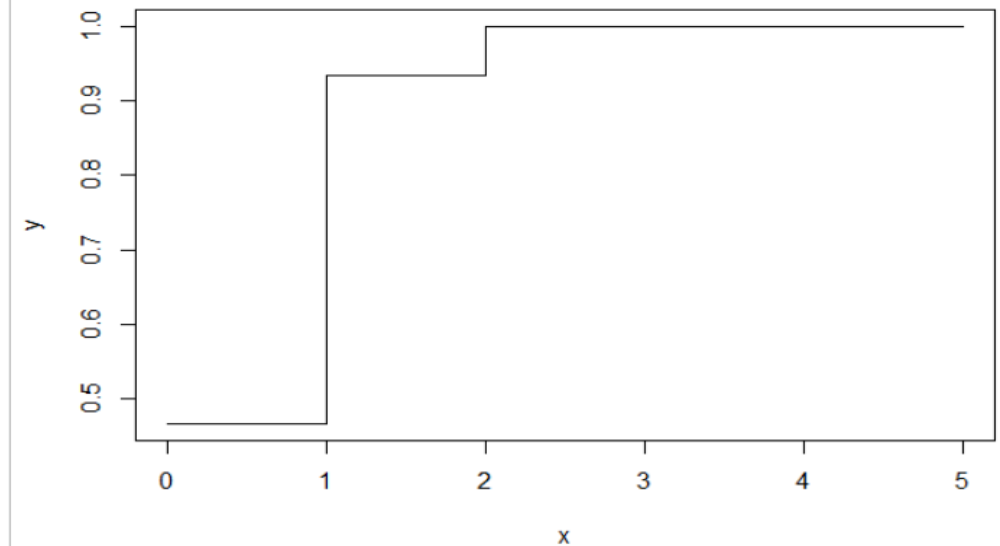
```
x <- 0:5
y <- c (14/30, 14/30, 2/30, 0, 0, 0)
plot (x, y, type = 'h')
points (0, 14/30)
points (1, 14/30)
points (2, 2/30)
```


Exemplo 1 , continuação) Uma urna possui 10 bolas sendo 3 com o número 1 e 7 com o número 2.

b) Definimos a variável Y como o número de bolas com o número 1 obtidas em duas extrações

Função de probabilidade acumulada:

$$P(Y \leq y) = F(y) = \begin{cases} 0, & \text{se } y < 0 \\ \frac{14}{30}, & \text{se } 0 \leq y < 1 \\ \frac{28}{30}, & \text{se } 1 \leq y < 2 \\ \frac{30}{30}, & \text{se } y \geq 2 \end{cases}$$



```
x <- 0:5
y <- c (14/30, 28/30, 30/30, 30/30, 30/30, 30/30)
plot (x, y, type ='s')
```

Variáveis aleatórias



A função **sample()** do R permite criar uma amostra de números pseudo-aleatórios de um vetor qualquer com ou sem reposição.

```
n <- 100000 #100000 escolhas a realizar, ou seja, um número muito grande
bolinhas <- 1:10 #bolinhas de 1 a 10
caixa <- bolinhas #caixa com 10 bolinhas
numero.bolinhas <- length(bolinhas)
total.bolinha.escolhida <- rep (0, numero.bolinhas)
probabilidade.bolinha.escolhida <- rep (0, numero.bolinhas)
#n escolhas de bolinhas com repetição
for (i in 1:n) {
  bolinha <- sample (caixa, 1)
  total.bolinha.escolhida [bolinha] <- total.bolinha.escolhida [bolinha] + 1
}
#probabilidade de escolha de cada bolinha
for (i in 1:numero.bolinhas) {
  probabilidade <- total.bolinha.escolhida [i] / n
  print (paste ('Probabilidade da bolinha [' , i , ' ] = ' , probabilidade))
}
```

Variáveis aleatórias



Como esperado, ao repetir o processo infinitas vezes, a probabilidade de escolha de cada bolinha é de 0,1.

Probabilidade da bolinha [1]	=	0.10044
Probabilidade da bolinha [2]	=	0.09918
Probabilidade da bolinha [3]	=	0.09919
Probabilidade da bolinha [4]	=	0.09972
Probabilidade da bolinha [5]	=	0.10138
Probabilidade da bolinha [6]	=	0.09983
Probabilidade da bolinha [7]	=	0.10104
Probabilidade da bolinha [8]	=	0.09925
Probabilidade da bolinha [9]	=	0.09932
Probabilidade da bolinha [10]	=	0.10065

As funções **choice()**, **choices()** e **sample()** do Python permitem criar amostras de números pseudo-aleatórios de um vetor qualquer com ou sem reposição dependendo da função usada.

```
import random

n = 100000 #100000 escolhas a realizar, ou seja, um número muito grande
bolinhas = [1,2,3,4,5,6,7,8,9,10] #bolinhas de 1 a 10
caixa = bolinhas #caixa com 10 bolinhas
numerobolinhas = len(bolinhas)
totalbolinhaescolhida = [0]*numerobolinhas # vetor de tamanho numerobolinhas com 0
probabilidadebolinhaescolhida = [0]*numerobolinhas
#n escolhas de bolinhas com repetição
for i in range(1, n):
    bolinha = random.choice(caixa)
    totalbolinhaescolhida [bolinha - 1] = totalbolinhaescolhida [bolinha - 1] + 1
#probabilidade de escolha de cada bolinha
for i in range(1, numerobolinhas):
    probabilidade = totalbolinhaescolhida [i - 1] / n
    print ('Probabilidade da bolinha [' + str(i) + '] = ' + str(probabilidade))
```

Agora vamos retomar o Exemplo 1. Uma urna possui 10 bolas sendo 3 com o número 1 e 7 com o número 2 e vamos refazer os itens a) e b) utilizando simulação.

a) Definimos a variável X como o número de bolas obtidas em uma extração aleatória.

```
import random
n = 100000 #100000 escolhas a realizar, ou seja, um número muito grande
bolinhas = [1,1,1,2,2,2,2,2,2,2] # bolinhas 1 e 2
caixa = bolinhas # caixa com 10 bolinhas
numerobolinhas = len(bolinhas)
totalbolinhaescolhida = [0]*numerobolinhas # vetor de tamanho numerobolinhas com 0s
probabilidadebolinhaescolhida = [0]*numerobolinhas # vetor de tamanho numerobolinhas com 0s
contb1 = 0
contb2 = 0
for i in range(n):
    bolinha = random.choice(caixa)
    if (bolinha == 1):
        contb1 = contb1 + 1
    if (bolinha == 2):
        contb2 = contb2 + 1
probb1 = (contb1/n)
print(probb1)
probb2 = (contb2/n)
print(probb2)
```

b) Definimos a variável Y como o número de bolas com o número 1 obtidas em duas extrações.

```
import random
n = 100000 # escolhas a realizar, ou seja, um número muito grande
bolinhas = [1,1,1,2,2,2,2,2,2,2] # bolinhas 1 e 2
caixa = bolinhas #caixa com 10 bolinhas
numerobolinhas = len(bolinhas)
totalbolinhaescolhida = [0]*numerobolinhas
probabilidadebolinhaescolhida = [0]*numerobolinhas
contb11 = 0
contb12 = 0
contb21 = 0
contb22 = 0
for i in range(n):
    bolas = random.sample(caixa, 2) # sem reposição
    if (bolas[0] == 1):
        if (bolas[1] == 1):
            contb11 = contb11 + 1
        else:
            contb12 = contb12 + 1
    else:
        if (bolas[1] == 1):
            contb21 = contb21 + 1
        else:
            contb22 = contb22 + 1
```

```
probb11 = (contb11/n)
print(probb11)
probb12 = (contb12/n)
print(probb12)
probb21 = (contb21/n)
print(probb21)
probb22 = (contb22/n)
print(probb22)
```

Distribuição Binomial



A Distribuição Binomial descreve o **número de sucessos** que ocorrem em um determinado número de experimentos de **Bernoulli**. Um ensaio de Bernoulli é definido por:

- Em cada ensaio considera-se somente a ocorrência ou não ocorrência de um certo evento que será denominado sucesso e cuja não ocorrência será denominada falha.
- Os ensaios são independentes.
- A probabilidade de sucesso, p , é a mesma para cada ensaio.
- A probabilidade de falha, $1 - p$, é a mesma para cada ensaio.

Para um experimento que consiste na realização de n ensaios independentes de Bernoulli, a probabilidade de um ponto amostral com sucessos nos k primeiros ensaios e falhas nos $n - k$ ensaios seguintes é $p^k (1 - p)^{n-k}$.

Distribuição Binomial



O número de pontos do espaço amostral que satisfaz essa condição é igual ao número de maneiras com que podemos escolher k ensaios para a ocorrência de sucesso dentre o total de n ensaios, pois nos $n - k$ restantes deverão ocorrer falhas. Este número é igual ao número de combinações de n elementos tomados k a k , ou seja:

$$\binom{n}{k} = \frac{n!}{k! (n - k)!}$$

Seja X o número de sucessos obtidos na realização de n ensaios de Bernoulli independentes. Se X tem distribuição **Binomial** com parâmetros n e p , em que p é a probabilidade de sucesso em cada ensaio, então sua fdp é:

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k} = \left(\frac{n!}{k! (n-k)!} \right) p^k (1 - p)^{n-k}$$

Distribuição Binomial



A função acumulada da distribuição da **Binomial** (n, p) é dada por:

$$P(X \leq k) = \sum_{x=0}^k \binom{n}{x} p^x (1-p)^{n-x}$$

A função acumulada $F(x)$ da distribuição **Binomial** é o somatório das probabilidades individuais de cada ponto da $f(x)$ até que um determinado valor k seja obtido.

Exemplo 1) Suponha que, para uma máquina que produz peças automotivas, a probabilidade de se obter uma peça defeituosa (sucesso) é $p = 0,1$. Das peças produzidas por esta máquina, seleciona-se uma amostra de 10 peças para análise. Qual a probabilidade de se obter:

a) Uma peça defeituosa?

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k} = \left(\frac{10!}{1!(10-1)!} \right) 0,1^1 (1 - 0,1)^{10-1} = 0,387$$

b) Nenhuma peça defeituosa?

$$P(X = 0) = \binom{10}{0} 0,1^0 (1 - 0,1)^{10-0} = 0,349$$

c) Duas peças defeituosas?

$$P(X = 2) = \binom{10}{2} 0,1^2 (1 - 0,1)^{10-2} = 0,194$$

Exemplo 1) Suponha que, para uma máquina que produz peças automotivas, a probabilidade de se obter uma peça defeituosa (sucesso) é $p = 0,1$. Das peças produzidas por esta máquina, seleciona-se uma amostra de 10 peças para análise. Qual a probabilidade de se obter:

d) No mínimo duas peças defeituosas?

$$P(X \geq 2) = P(X = 2) + \cdots + P(X = 10) = 1 - [P(X = 0) + P(X = 1)] = 0,264$$

e) No máximo duas peças defeituosas?

$$P(X \leq 2) = P(X = 0) + P(X = 1) + P(X = 2) = 0,930$$

No R, o trabalho com uma **variável aleatória** X está baseado em 4 procedimentos identificados pelas letras **p** , **d** , **q** e **r** que, adicionadas previamente ao código das **distribuições**, permitem o seu uso com **diferentes propósitos**:

- **rdistrib**: Gera **números aleatórios** baseados na distribuição definida. Necessita de argumentos especificando o tamanho da amostra, além dos parâmetros requeridos pela distribuição de interesse;
- **pdistrib**: Calcula a **função de probabilidade acumulada** $F(x)$;
- **ddistrib**: Calcula a **distribuição de probabilidade** $f(x)$;
- **qdistrib**: Calcula o **quantil** correspondente a uma dada probabilidade.

No Python, o trabalho com uma **variável aleatória** X pode ser realizado através do módulo stats da biblioteca SciPy. Ele apresenta métodos comuns, como rvs, pdf, cdf, ppf e stats, às distribuições que permitem o seu uso com **diferentes propósitos**:

- **rvs**: Gera **números aleatórios** baseados na distribuição definida;
- **cdf**: Calcula a **função de probabilidade acumulada** $F(x)$;
- **pdf**: Calcula a **distribuição de probabilidade** $f(x)$;
- **ppf**: Percent Point Function (inverse of cdf — percentiles).
- **stats**: Return mean, variance, skew, and/or kurtosis.

Distribuição Binomial no R



Distribuição	Sintaxe	Opções
Binomial (binom)	dbinom (x, n, p, opções)	x : número de sucessos em n ensaios de Bernoulli. p : probabilidade de sucesso. q : número máximo de sucessos em ensaios de Bernoulli. prob : probabilidade em n ensaios de Bernoulli. obs : número de observações. n : número de ensaios de Bernoulli.
	pbinom (q, n, p, opções)	
	qbinom (prob, n, p, opções)	
	rbinom (obs, n, p)	

Distribuição Binomial



Exemplo 2) Suponha que, para uma máquina que produz peças automotivas, a probabilidade de se obter uma peça defeituosa (sucesso) é $p = 0,1$. Das peças produzidas por esta máquina, seleciona-se uma amostra de 10 peças para análise. Qual a probabilidade de se obter:

a) Uma peça defeituosa?

```
dbinom(1, 10, 0.1)  
0.3874205
```

b) Nenhuma peça defeituosa?

```
dbinom(0, 10, 0.1)  
0.3486784
```

c) Duas peças defeituosas?

```
dbinom(2, 10, 0.1)  
0.1937102
```

Exemplo 2) Suponha que, para uma máquina que produz peças automotivas, a probabilidade de se obter uma peça defeituosa (sucesso) é $p = 0,1$. Das peças produzidas por esta máquina, seleciona-se uma amostra de 10 peças para análise. Qual a probabilidade de se obter:

d) No mínimo duas peças defeituosas?

```
1 - pbinom(1, 10, 0.1)  
0.2639011
```

e) No máximo duas peças defeituosas?

```
pbinom(2, 10, 0.1)  
0.9298092
```


Distribuição Binomial

```
nsamples <- 1000000
namostra <- 10; cont.defeituosa <- 0; npeca.sem.defeito <- 0; npeca.defeituosa <- rep(0, 10)
for (i in 1:nsamples) {
  cont.defeituosa <- 0
  for (j in 1:namostra) { # seleciono 10 peças
    peca.selecionada.defeituosa <- sample(c(T, F), 1, prob = c(0.1, 0.9))
    if (peca.selecionada.defeituosa) {
      cont.defeituosa <- cont.defeituosa + 1
    }
  }
  if (cont.defeituosa > 0) { # npeca.defeituosa[i] contém o total de peças i defeituosas
    npeca.defeituosa[cont.defeituosa] <- npeca.defeituosa[cont.defeituosa] + 1
  } else {
    npeca.sem.defeito <- npeca.sem.defeito + 1
  }
}
prob.nenhuma.defeituosa <- npeca.sem.defeito/nsamples
print(prob.nenhuma.defeituosa)
prob.uma.defeituosa <- npeca.defeituosa[1]/nsamples
print(prob.uma.defeituosa)
prob.duas.defeituosas <- npeca.defeituosa[2]/nsamples
print(prob.duas.defeituosas)
prob.minimo.duas.defeituosas <- sum(npeca.defeituosa[2:10])/nsamples # filtro no vetor
print(prob.minimo.duas.defeituosas)
prob.maximo.duas.defeituosas <- (npeca.sem.defeito + sum(npeca.defeituosa[1:2]))/nsamples
print(prob.maximo.duas.defeituosas)
```

Distribuição Binomial



Exemplo 3) Considere que a probabilidade de certa peça artesanal ser produzida com perfeição por uma pessoa que está em treinamento é igual a 0,5. Considere que a pessoa produza 6 peças por vez.

a) Obter a distribuição de probabilidades do número de peças perfeitas produzidas pela pessoa em 6 peças.

```
vetor.probababilidade <- dbinom (0:6, 6, 0.5)
vetor.probababilidade
0.015625 0.093750 0.234375 0.312500 0.234375 0.093750
0.015625
intervalo <- 0:6
intervalo
0 1 2 3 4 5 6
```

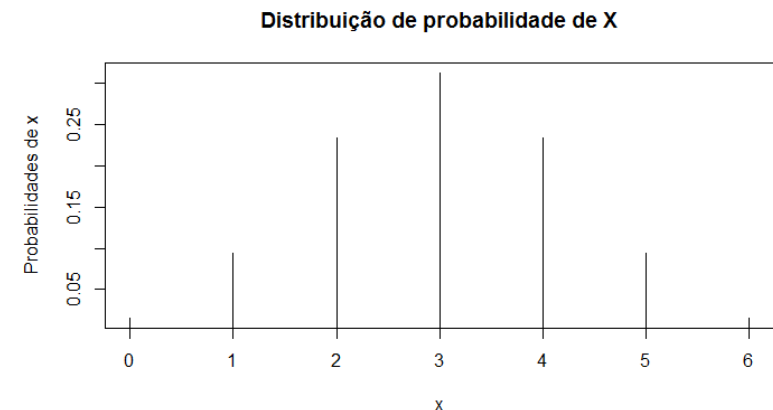
Distribuição Binomial

Exemplo 3) Considere que a probabilidade de certa peça artesanal ser produzida com perfeição por uma pessoa que está em treinamento é igual a 0,5. Considere que a pessoa produza 6 peças por vez.

b) Construir o gráfico de distribuição de probabilidade obtida no item a.

```
vetor.proabilidade
0.015625 0.093750 0.234375 0.312500 0.234375 0.093750 0.015625
intervalo <- 0:6
intervalo
0 1 2 3 4 5 6
```

```
plot(intervalo, #intervalo desejado
      vetor.proabilidade, #valores de probabilidade
      type="h", #traço do eixo ao ponto
      xlab='x',
      ylab='Probabilidades de x',
      main='Distribuição de probabilidade de X')
```



Distribuição Binomial

```
nsamples <- 1000000
namostra <- 6; cont.defeituosa <- 0; npeca.sem.defeito <- 0; npeca.defeituosa <- rep(0, 6)
for (i in 1:nsamples) {
  cont.defeituosa <- 0
  for (j in 1:namostra) { # são produzidas 6 peças
    peca.selecionada.defeituosa <- sample(c(T, F), 1, prob = c(0.5, 0.5))
    if (peca.selecionada.defeituosa) {
      cont.defeituosa <- cont.defeituosa + 1
    }
  }
  if (cont.defeituosa > 0) { # npeca.defeituosa[i] contém o total de peças i defeituosas
    npeca.defeituosa[cont.defeituosa] <- npeca.defeituosa[cont.defeituosa] + 1
  } else {
    npeca.sem.defeito <- npeca.sem.defeito + 1
  }
}
vetor.probabilidade <- c(npeca.sem.defeito, npeca.defeituosa)
vetor.probabilidade <- vetor.probabilidade / nsamples
intervalo <- 0:6
plot(intervalo,
      vetor.probabilidade, #valores de probabilidade
      type="h", #traço do eixo ao ponto
      xlab='x',
      ylab='Probabilidades de x',
      main='Distribuição de probabilidade de X')
```

Distribuição Poisson



Em muitas situações, nos deparamos com situações em que o número de ensaios n é grande ($n \rightarrow \infty$) e p é pequeno ($p \rightarrow 0$), no cálculo da função **Binomial**, o que nos leva a algumas dificuldades. Nestes casos, trabalhamos com a distribuição **Poisson**.

A distribuição de **Poisson** é utilizada quando não é prático ou mesmo possível determinar o número de fracassos ou o número total de provas de um experimento.

É muito útil para descrever as probabilidades do número de ocorrências num campo ou intervalo contínuo, em geral, de tempo ou espaço.

A probabilidade de que existam exatamente k ocorrências (k sendo um inteiro não negativo, $k = 0, 1, 2, \dots$) é:

$$f(k; \lambda) = P(X = k) = \frac{e^{-\lambda} \lambda^k}{k!}$$

λ é um número real, que indica a taxa de ocorrência por unidade de medida.

Distribuição Poisson



Exemplo 1) A análise das estradas feita por uma empresa aponta que ocorrem, em média, 4 buracos por Km. A análise aponta, ainda, que a quantidade de buracos segue uma distribuição Poisson. Qual a probabilidade de que um Km qualquer contenha:

a) Um buraco?

$$P(X = k) = \frac{e^{-\lambda} \lambda^k}{k!} = \frac{e^{-4} 4^1}{1!} = 0,073$$

b) Dois buracos?

$$P(X = k) = \frac{e^{-\lambda} \lambda^k}{k!} = \frac{e^{-4} 4^2}{2!} = 0,147$$

c) Nenhum buraco?

$$P(X = k) = \frac{e^{-\lambda} \lambda^k}{k!} = \frac{e^{-4} 4^0}{0!} = 0,018$$

d) Pelo menos um buraco?

$$P(X \geq k) = 1 - P(X < k) = 1 - P(X < 1) = 1 - P(X = 0) = 1 - \frac{e^{-\lambda} \lambda^0}{0!} = 0,982$$

No R, o trabalho com uma **variável aleatória** X está baseado em 4 procedimentos identificados pelas letras **p** , **d** , **q** e **r** que, adicionadas previamente ao código das **distribuições**, permitem o seu uso com **diferentes propósitos**:

- **rdistrib**: Gera **números aleatórios** baseados na distribuição definida. Necessita de argumentos especificando o tamanho da amostra, além dos parâmetros requeridos pela distribuição de interesse;
- **pdistrib**: Calcula a **função de probabilidade acumulada** $F(x)$;
- **ddistrib**: Calcula a **distribuição de probabilidade** $f(x)$;
- **qdistrib**: Calcula o **quantil** correspondente a uma dada probabilidade.

Distribuição Poisson no R

Distribuição	Sintaxe	Opções
Poisson (pois)	dpois (x, lambda, opções)	x : vetor contendo o número de ocorrências. lambda : número médio de eventos ocorrendo no intervalo considerado. q : vetor contendo os quantis. prob : vetor contendo as probabilidades. obs : número de observações.
	ppois (q, lambda, opções)	
	qpois (prob, lambda, opções)	
	rpois (obs, lambda)	

Distribuição Poisson



Exemplo 2) A análise das estradas feita por uma empresa aponta que ocorrem, em média, 4 buracos por Km. A análise aponta, ainda, que a quantidade de buracos segue uma distribuição Poisson. Qual a probabilidade de que um Km qualquer contenha:

a) Um buraco?

```
dpois (1, 4)  
0.07326256
```

b) Dois buracos?

```
dpois (2, 4)  
0.1465251
```

c) Nenhum buraco?

```
dpois (0, 4)  
0.01831564
```

d) Pelo menos um buraco?

```
1 - dpois (0, 4)  
0.9816844
```

Distribuição Poisson



Exemplo 2, continuação) A análise das estradas feita por uma empresa aponta que ocorrem, em média, 4 buracos por Km. A análise aponta, ainda, que a quantidade de buracos segue uma distribuição Poisson. Qual a probabilidade de que um Km qualquer contenha:

a) Um buraco?

```
nsamples <- 1000000; nkm.sem.buraco <- 0; nkm.com.buraco <- rep(0, 30)
for (i in 1:nsamples) {
  buraco <- rpois(1, 4) # uso de rpois com lambda = 4 e pegando uma observação
  if (buraco > 0) {
    nkm.com.buraco[buraco] <- nkm.com.buraco[buraco] + 1
  } else {
    nkm.sem.buraco <- nkm.sem.buraco + 1
  }
}
prob.um.buraco <- nkm.com.buraco[1] / nsamples
print(prob.um.buraco)
```

Distribuição Poisson



Exemplo 2, continuação) A análise das estradas feita por uma empresa aponta que ocorrem, em média, 4 buracos por Km. A análise aponta, ainda, que a quantidade de buracos segue uma distribuição Poisson. Qual a probabilidade de que um Km qualquer contenha:

b) Dois buracos?

```
nsamples <- 1000000; nkm.sem.buraco <- 0; nkm.com.buraco <- rep(0, 30)
for (i in 1:nsamples) {
  buraco <- rpois(1, 4) # uso de rpois com lambda = 4 e pegando uma observação
  if (buraco > 0) {
    nkm.com.buraco[buraco] <- nkm.com.buraco[buraco] + 1
  } else {
    nkm.sem.buraco <- nkm.sem.buraco + 1
  }
}
prob.dois.buracos <- nkm.com.buraco[2] / nsamples
print(prob.dois.buracos)
```

Distribuição Poisson



Exemplo 2, continuação) A análise das estradas feita por uma empresa aponta que ocorrem, em média, 4 buracos por Km. A análise aponta, ainda, que a quantidade de buracos segue uma distribuição Poisson. Qual a probabilidade de que um Km qualquer contenha:

c) Nenhum buraco?

```
nsamples <- 1000000; nkm.sem.buraco <- 0; nkm.com.buraco <- rep(0, 30)
for (i in 1:nsamples) {
  buraco <- rpois(1, 4) # uso de rpois com lambda = 4 e pegando uma observação
  if (buraco > 0) {
    nkm.com.buraco[buraco] <- nkm.com.buraco[buraco] + 1
  } else {
    nkm.sem.buraco <- nkm.sem.buraco + 1
  }
}
prob.nenhum.buraco <- nkm.sem.buraco / nsamples
print(prob.nenhum.buraco)
```

Distribuição Poisson



Exemplo 2, continuação) A análise das estradas feita por uma empresa aponta que ocorrem, em média, 4 buracos por Km. A análise aponta, ainda, que a quantidade de buracos segue uma distribuição Poisson. Qual a probabilidade de que um Km qualquer contenha:

d) Pelo menos um buraco?

```
nsamples <- 1000000; nkm.sem.buraco <- 0; nkm.com.buraco <- rep(0, 30)
for (i in 1:nsamples) {
  buraco <- rpois(1, 4) # uso de rpois com lambda = 4 e pegando uma observação
  if (buraco > 0) {
    nkm.com.buraco[buraco] <- nkm.com.buraco[buraco] + 1
  } else {
    nkm.sem.buraco <- nkm.sem.buraco + 1
  }
}
prob.pelo.menos.um.buraco <- sum(nkm.com.buraco[1:10]) / nsamples
print(prob.pelo.menos.um.buraco)
```

Distribuição Geométrica



Descreve a quantidade de experimentos de **Bernoulli** que precisam ser realizados até a ocorrência do primeiro sucesso.

Para introduzir o modelo de variável geométrica, vamos retomar uma sequência de ensaios de **Bernoulli**, que é definida por:

- Em cada ensaio, considera-se somente a ocorrência ou não ocorrência de um certo evento que será denominado sucesso e cuja não ocorrência será denominada falha.
- Os ensaios são independentes.
- A probabilidade de sucesso, p , é a mesma para cada ensaio.
- A probabilidade de falha, $1 - p$, é a mesma para cada ensaio.

Seja X a variável aleatória que fornece o número de falhas até o primeiro sucesso, ela tem distribuição **Geométrica** com parâmetro p , $0 < p < 1$, se sua função de probabilidade é dada por:

$$P(X = k) = (1 - p)^{k-1}p$$

Exemplo 1) Suponha que a probabilidade de um equipamento eletrônico ser defeituoso é de 0,2. Realizaram-se testes na produção de um determinado lote. Determine a probabilidade do primeiro defeito encontrado:

a) Ocorrer no sétimo equipamento do teste:

$$P(7) = (0,2)(1 - 0,2)^{7-1} = 0,0524 = 5,24\%$$

b) Não ocorrer no quarto equipamento testado:

$$\begin{aligned} 1 - P(4) &= 1 - [(0,2)(1 - 0,2)^{4-1}] \\ &= 1 - 0,1024 = 0,8976 = 89,76\% \end{aligned}$$

No R, o trabalho com uma **variável aleatória** X está baseado em 4 procedimentos identificados pelas letras **p** , **d** , **q** e **r** que, adicionadas previamente ao código das **distribuições**, permitem o seu uso com **diferentes propósitos**:

- **rdistrib**: Gera **números aleatórios** baseados na distribuição definida. Necessita de argumentos especificando o tamanho da amostra, além dos parâmetros requeridos pela distribuição de interesse;
- **pdistrib**: Calcula a **função de probabilidade acumulada** $F(x)$;
- **ddistrib**: Calcula a **distribuição de probabilidade** $f(x)$;
- **qdistrib**: Calcula o **quantil** correspondente a uma dada probabilidade.

Distribuição Geométrica no R



Distribuição	Sintaxe	Opções
Geométrica (geom)	dgeom (x, prob, opções)	x : vetor contendo o número de falhas ocorridas em uma sequência de Bernoulli antes do sucesso. prob : probabilidade de sucesso em cada tentativa. q : vetor contendo os quantis. p : vetor contendo as probabilidades . n : número de observações.
	pgeom (q, prob, opções)	
	qgeom (p, prob, opções)	
	rgeom (n, prob)	

Exemplo 2) Suponha que a probabilidade de um equipamento eletrônico ser defeituoso é de 0,2. Realizaram-se testes na produção de um determinado lote. Determine a probabilidade do primeiro defeito encontrado:

a) Ocorrer no sétimo equipamento do teste:

```
dgeom(6, 0.2)
```

```
0.0524288
```

b) Não ocorrer no quarto equipamento testado:

```
1 - dgeom(3, 0.2)
```

```
0.8976
```

Exemplo 2, continuação) Suponha que a probabilidade de um equipamento eletrônico ser defeituoso é de 0,2. Realizaram-se testes na produção de um determinado lote. Determine a probabilidade do primeiro defeito encontrado:

a) Ocorrer no sétimo equipamento do teste:

```
nsamples <- 10000000
n.sem.defeito <- 0; n.com.defeito <- rep(0, 100)
for (i in 1:nsamples) {
  defeito <- rgeom(1, 0.2) # uso de rgeom com prob = 0.2 e pegando uma observação
  if (defeito > 0) {
    n.com.defeito[defeito] <- n.com.defeito[defeito] + 1
  } else {
    n.sem.defeito <- n.sem.defeito + 1
  }
}
prob.ocorrer.setimo <- n.com.defeito[6] / nsamples # seis sem defeito e o sétimo com defeito
print(prob.ocorrer.setimo)
```

Exemplo 2, continuação) Suponha que a probabilidade de um equipamento eletrônico ser defeituoso é de 0,2. Realizaram-se testes na produção de um determinado lote. Determine a probabilidade do primeiro defeito encontrado:

b) Não ocorrer no quarto equipamento testado:

```
nsamples <- 10000000
n.sem.defeito <- 0; n.com.defeito <- rep(0, 100)
for (i in 1:nsamples) {
  defeito <- rgeom(1, 0.2) # uso de rgeom com prob = 0.2 e pegando uma observação
  if (defeito > 0) {
    n.com.defeito[defeito] <- n.com.defeito[defeito] + 1
  } else {
    n.sem.defeito <- n.sem.defeito + 1
  }
}
# para não ocorrer no quarto não há três sucessos seguido de um erro
prob.nao.ocorrer.quarto <- (n.sem.defeito + sum(n.com.defeito) - n.com.defeito[3]) / nsamples
print(prob.nao.ocorrer.quarto)
```

Distribuição Geométrica

Exemplo 3) A probabilidade de uma regulagem bem sucedida na montagem de um determinado tipo de produto é de 0.80. Assuma que as tentativas são independentes.

- a) Qual é a probabilidade de que a primeira regulagem bem sucedida requeira exatamente quatro tentativas?

```
dgeom (3, 0.8)
0.0064
```

Na função $dgeom(x, p)$ x representa o número de fracassos, sendo neste caso $x = 4 - 1 = 3$.

- b) Qual é a probabilidade de que a primeira regulagem bem sucedida requeira no máximo quatro tentativas ?

```
dgeom (0, 0.8) + dgeom (1, 0.8) + dgeom (2, 0.8) + dgeom (3, 0.8)
0.9984
```

- c) Qual é a probabilidade de que a primeira regulagem bem sucedida requeira ao menos quatro tentativas ?

```
1 - (dgeom (0, 0.8) + dgeom (1, 0.8) + dgeom (2, 0.8))
0.008
```

Exemplo 3, continuação) A probabilidade de uma regulagem bem sucedida na montagem de um determinado tipo de produto é de 0.80. Assuma que as tentativas são independentes.

a) Qual é a probabilidade de que a primeira regulagem bem sucedida requeira exatamente quatro tentativas?

```
nsamples <- 10000000
n.bem.sucedida <- 0; n.mal.sucedida <- rep(0, 100)
for (i in 1:nsamples) {
  regulagem <- rgeom(1, 0.8) # uso de rgeom com prob = 0.8 e pegando uma observação
  if (regulagem > 0) {
    n.mal.sucedida[regulagem] <- n.mal.sucedida[regulagem] + 1
  } else {
    n.bem.sucedida <- n.bem.sucedida + 1
  }
}
prob.quatro.tentativas <- n.mal.sucedida[3] / nsamples
print(prob.quatro.tentativas)
```

Exemplo 3, continuação) A probabilidade de uma regulagem bem sucedida na montagem de um determinado tipo de produto é de 0.80. Assuma que as tentativas são independentes.

b) Qual é a probabilidade de que a primeira regulagem bem sucedida requeira no máximo quatro tentativas ?

```
nsamples <- 10000000
n.bem.sucedida <- 0; n.mal.sucedida <- rep(0, 100)
for (i in 1:nsamples) {
  regulagem <- rgeom(1, 0.8) # uso de rgeom com prob = 0.8 e pegando uma observação
  if (regulagem > 0) {
    n.mal.sucedida[regulagem] <- n.mal.sucedida[regulagem] + 1
  } else {
    n.bem.sucedida <- n.bem.sucedida + 1
  }
}
prob.no.maximo.quatro <- (n.bem.sucedida + sum(n.mal.sucedida[1:3])) / nsamples
print(prob.no.maximo.quatro)
```

Exemplo 3, continuação) A probabilidade de uma regulagem bem sucedida na montagem de um determinado tipo de produto é de 0.80. Assuma que as tentativas são independentes.

c) Qual é a probabilidade de que a primeira regulagem bem sucedida requeira ao menos quatro tentativas ?

```
nsamples <- 10000000
n.bem.sucedida <- 0; n.mal.sucedida <- rep(0, 100)
for (i in 1:nsamples) {
  regulagem <- rgeom(1, 0.8) # uso de rgeom com prob = 0.8 e pegando uma observação
  if (regulagem > 0) {
    n.mal.sucedida[regulagem] <- n.mal.sucedida[regulagem] + 1
  } else {
    n.bem.sucedida <- n.bem.sucedida + 1
  }
}
prob.ao.menos.quatro <- (1 - sum(n.mal.sucedida[3:100])) / nsamples
print(prob.ao.menos.quatro)
```


Distribuição Hipergeométrica



É utilizada quando se consideram extrações aleatórias, sem reposição, de uma população dividida segundo dois atributos.

Considere uma população com N objetos nos quais M são classificados como do tipo A e $N - M$ são classificados como do tipo B . Seja X a variável aleatória que conta o número de objetos classificados como do tipo A em uma amostra de tamanho n . Então a distribuição de probabilidade de X é dada por:

$$P(X = k) = \frac{\binom{M}{k} \binom{N-M}{n-k}}{\binom{N}{n}}$$

Distribuição Hipergeométrica



Exemplo 1) Um banco de investimentos possui 20 clientes com mais de R\$ 100 *milhões* aplicados. Destes, 5 moram fora do Brasil:

a) Ao retirar 3 ao acaso, qual a probabilidade de que 2 deles morem fora do Brasil?

$$P(X = 2) = \frac{C(5)_2 C(20-5)_{3-2}}{C(20)_3} = \frac{10 \times 15}{1140} = 0,1315 = 13,15\%$$

b) Ao retirar 5 ao acaso, qual a probabilidade de que todos morem no Brasil?

$$P(X = 5) = \frac{C(5)_0 C(20-5)_5}{C(20)_5} = 0,1937 = 19,37\%$$

No R, o trabalho com uma **variável aleatória** X está baseado em 4 procedimentos identificados pelas letras **p** , **d** , **q** e **r** que, adicionadas previamente ao código das **distribuições**, permitem o seu uso com **diferentes propósitos**:

- **rdistrib**: Gera **números aleatórios** baseados na distribuição definida. Necessita de argumentos especificando o tamanho da amostra, além dos parâmetros requeridos pela distribuição de interesse;
- **pdistrib**: Calcula a **função de probabilidade acumulada** $F(x)$;
- **ddistrib**: Calcula a **distribuição de probabilidade** $f(x)$;
- **qdistrib**: Calcula o **quantil** correspondente a uma dada probabilidade.

Distribuição Hipergeométrica no R



Distribuição	Sintaxe	Opções
Hipergeométrica (hyper)	dhyper (x, m, n, k, opções)	x : vetor contendo o número de elementos com a característica A extraídos sem reposição de uma urna que contém elementos com características A e B. m : número de elementos com a característica A. n : número de elementos com a característica B. k : número de elementos extraídos da urna. q : vetor contendo os quantis. p : vetor contendo as probabilidades. nn : número de observações.
	phyper (q, m, n, k, opções)	
	qhyper (p, m, n, k, opções)	
	rhyper (nn, m, n, k)	

Distribuição Hipergeométrica



Exemplo 1, continuação) Um banco de investimentos possui 20 clientes com mais de R\$ 100 *milhões* aplicados. Destes, 5 moram fora do Brasil:

a) Ao retirar 3 ao acaso, qual a probabilidade de que 2 deles morem fora do Brasil?

```
dhypcr(2, 5, 15, 3)  
0.1315789
```

b) Ao retirar 5 ao acaso, qual a probabilidade de que todos morem no Brasil?

```
dhypcr(0, 5, 15, 5)  
0.193692
```

Distribuição Hipergeométrica



Exemplo 1, continuação) Um banco de investimentos possui 20 clientes com mais de R\$ 100 *milhões* aplicados. Destes, 5 moram fora do Brasil:

a) Ao retirar 3 ao acaso, qual a probabilidade de que 2 deles morem fora do Brasil?

```
nsamples <- 10000000
nobservacao <- 20
n.obs.fora <- 5
n.obs.Brasil <- 15
n.extraido <- 3
n.mora.fora <- rep(0, 100)
n.mora.brasil <- 0
for (i in 1:nsamples) {
  cliente <- rhyper(1, n.obs.fora, n.obs.Brasil, n.extraido) # uma observação
  if (cliente > 0) {
    n.mora.fora[cliente] <- n.mora.fora[cliente] + 1
  } else {
    n.mora.brasil <- n.mora.brasil + 1
  }
}
prob.dois.mora.fora <- n.mora.fora[2] / nsamples
print(prob.dois.mora.fora)
```

Distribuição Hipergeométrica



Exemplo 1, continuação) Um banco de investimentos possui 20 clientes com mais de R\$ 100 *milhões* aplicados. Destes, 5 moram fora do Brasil:

b) Ao retirar 5 ao acaso, qual a probabilidade de que todos morem no Brasil?

```
nsamples <- 10000000
nobservacao <- 20
n.obs.fora <- 5
n.obs.Brasil <- 15
n.extraido <- 5
n.mora.fora <- rep(0, 100)
n.mora.brasil <- 0
for (i in 1:nsamples) {
  cliente <- rhyper(1, n.obs.fora, n.obs.Brasil, n.extraido) # uma observação
  if (cliente > 0) {
    n.mora.fora[cliente] <- n.mora.fora[cliente] + 1
  } else {
    n.mora.brasil <- n.mora.brasil + 1
  }
}
prob.todos.mora.Brasil <- n.mora.brasil / nsamples
print(prob.todos.mora.Brasil)
```

Distribuição Hipergeométrica



Exemplo 2) Uma urna contém 15 bolas, sendo 10 brancas e 5 vermelhas. Uma amostra de extensão 4, não ordenada e sem reposição, é retirada ao acaso da urna. Determine qual é a probabilidade de que três bolas sejam vermelhas.

```
dhypcr (3, 5, 10, 4)  
0.07326007
```


Distribuição Hipergeométrica



Exemplo 2, continuação) Uma urna contém 15 bolas, sendo 10 brancas e 5 vermelhas. Uma amostra de extensão 4, não ordenada e sem reposição, é retirada ao acaso da urna. Determine qual é a probabilidade de que três bolas sejam vermelhas.

```
nsamples <- 10000000
nobservacao <- 15
n.obs.branca <- 10
n.obs.vermelha <- 5
n.extraido <- 4
n.bola.vermelha <- rep(0, 100)
n.bola.branca <- 0
for (i in 1:nsamples) {
  bola <- rhyper(1, n.obs.vermelha, n.obs.branca, n.extraido) # uma observação
  if (bola > 0) {
    n.bola.vermelha[bola] <- n.bola.vermelha[bola] + 1
  } else {
    n.bola.branca <- n.bola.branca + 1
  }
}
prob.tres.bola.vermelha <- n.bola.vermelha[3] / nsamples
print(prob.tres.bola.vermelha)
```

Utilizando o Python (módulo stats da biblioteca SciPy), refaça todos os exemplos deste material que foram resolvidos em R, inclusive os que foram resolvidos por simulação.

Utilize métodos como `rvs`, `pdf`, `cdf`, `ppf` e stats relacionados às distribuições.

- **rvs**: Gera **números aleatórios** baseados na distribuição definida;
- **cdf**: Calcula a **função de probabilidade acumulada** $F(x)$;
- **pdf**: Calcula a **distribuição de probabilidade** $f(x)$;
- **ppf**: Percent Point Function (inverse of cdf — percentiles).
- **stats**: Return mean, variance, skew, and/or kurtosis.