Introdução a Linguagem R

Simulação

Delermando Branquinho Filho

Gerando Números Aleatórios

Funções para distribuições de probabilidade em R - `rnorm` : gerar variáveis aleatórias normais com uma dada média e desvio padrão - `dnorm` : avalie a densidade de probabilidade Normal (com uma dada média / SD) em um ponto (ou vetor de pontos) - `pnorm` : avaliar a função de distribuição cumulativa para uma distribuição Normal - `rpois` : gerar variáveis aleatórias de Poisson com uma determinada taxa

As funções de distribuição de probabilidade geralmente têm quatro funções associadas a elas. As funções são prefixadas com um - `d` para densidade - `r` para geração de números aleatórios - `p` para distribuição cumulativa - `q` para a função quantile

Trabalhar com as distribuições Normal requer a utilização destas quatro funções

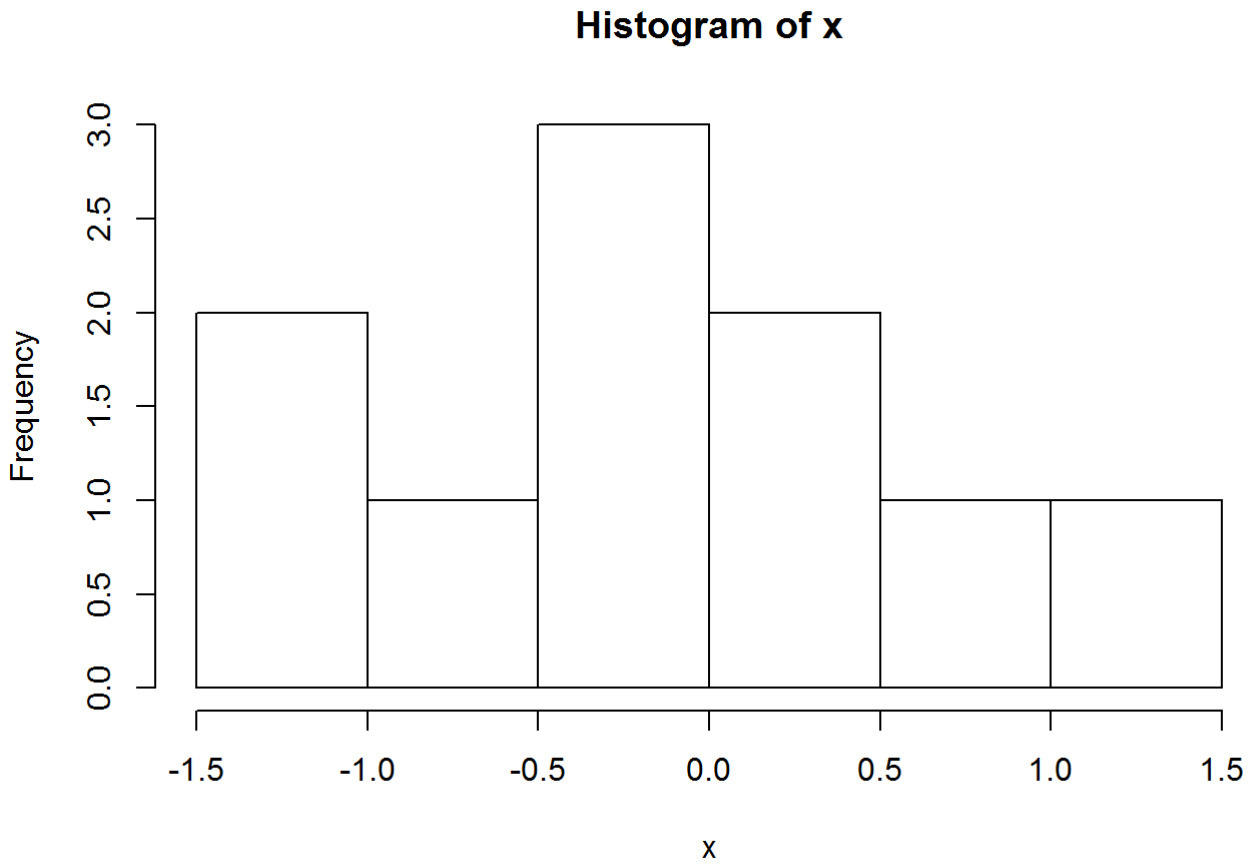
```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Se Φ é a função de distribuição cumulativa para um padrão

Distribuição normal, então `pnorm(q)` = $\Phi(q)$ e `qnorm(p)` = $Phi^{-1}(p)$.

Trabalhar com as distribuições Normal requer a utilização destas quatro funções

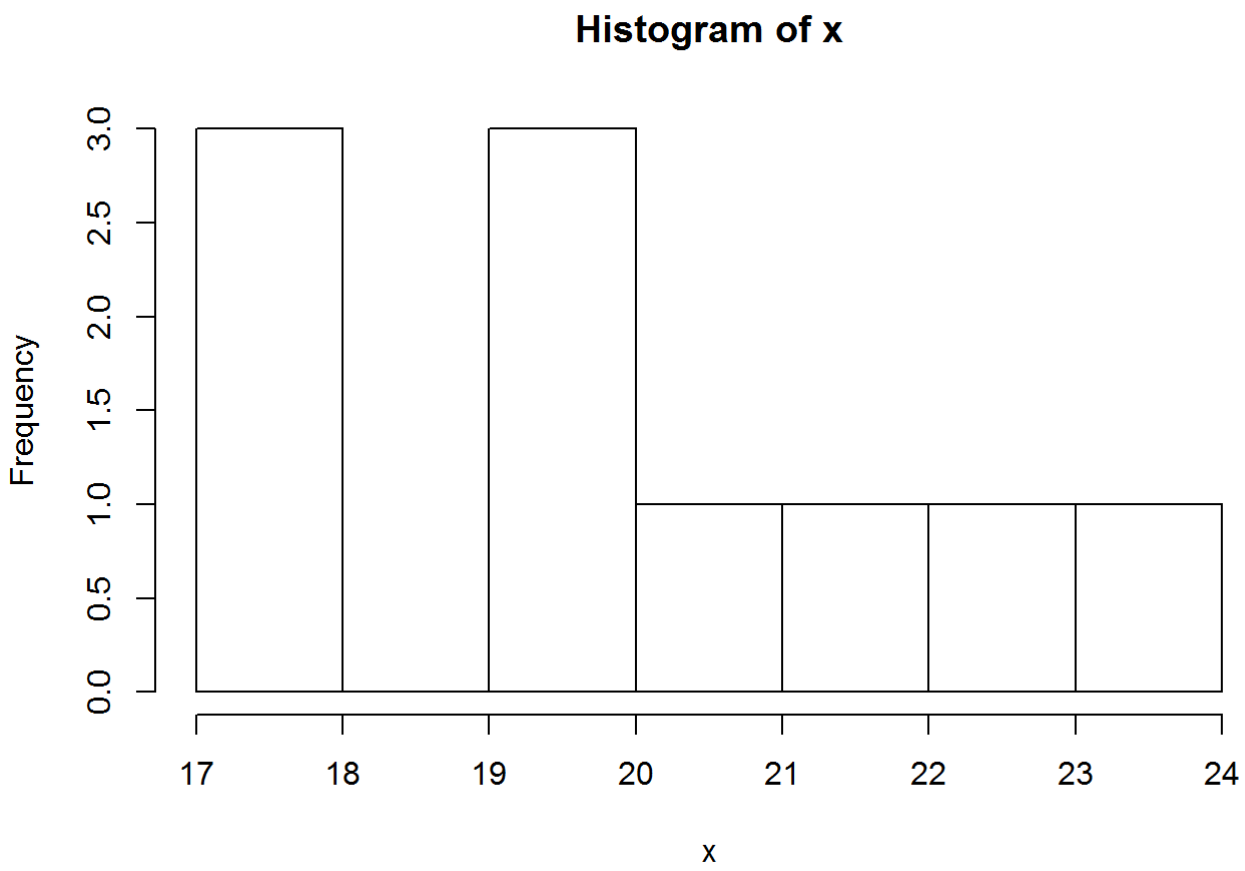
```
x <- rnorm(10)
hist(x)
```




```
x <- rnorm(10, 20, 2)
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.28   18.40   19.78   20.06   21.44   23.86
```

```
hist(x)
```



A definição da semente de número aleatório com `set.seed` assegura a reprodutibilidade

```
set.seed(1)
rnorm(5)
```

```
## [1] -0.6264538  0.1836433 -0.8356286  1.5952808  0.3295078
```

```
rnorm(5)
```

```
## [1] -0.8204684  0.4874291  0.7383247  0.5757814 -0.3053884
```

```
set.seed(1)
rnorm(5)
```

```
## [1] -0.6264538  0.1836433 -0.8356286  1.5952808  0.3295078
```

Always set the random number seed when conducting a simulation!

Gerando dados de Poisson

```
rpois(10, 1)
```

```
##  [1] 0 0 1 1 2 1 1 4 1 2
```

```
rpois(10, 2)
```

```
##  [1] 4 1 2 0 1 1 0 1 4 1
```

```
rpois(10, 20)
```

```
##  [1] 19 19 24 23 22 24 23 20 11 22
```

```
ppois(10, 1)  ## Cumulative distribution
```

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

```
ppois(10, 2)
```

```
## [1] 0.9999917
```

```
ppois(10, 3)
```

```
## [1] 0.9997077
```

Gerando números aleatórios a partir de um modelo linear

Suponha que queremos simular a partir do seguinte modelo linear

$$y = \beta_0 + \beta_1 x + \varepsilon$$

Onde

$\varepsilon \sim \mathcal{N}(0, 2^2)$

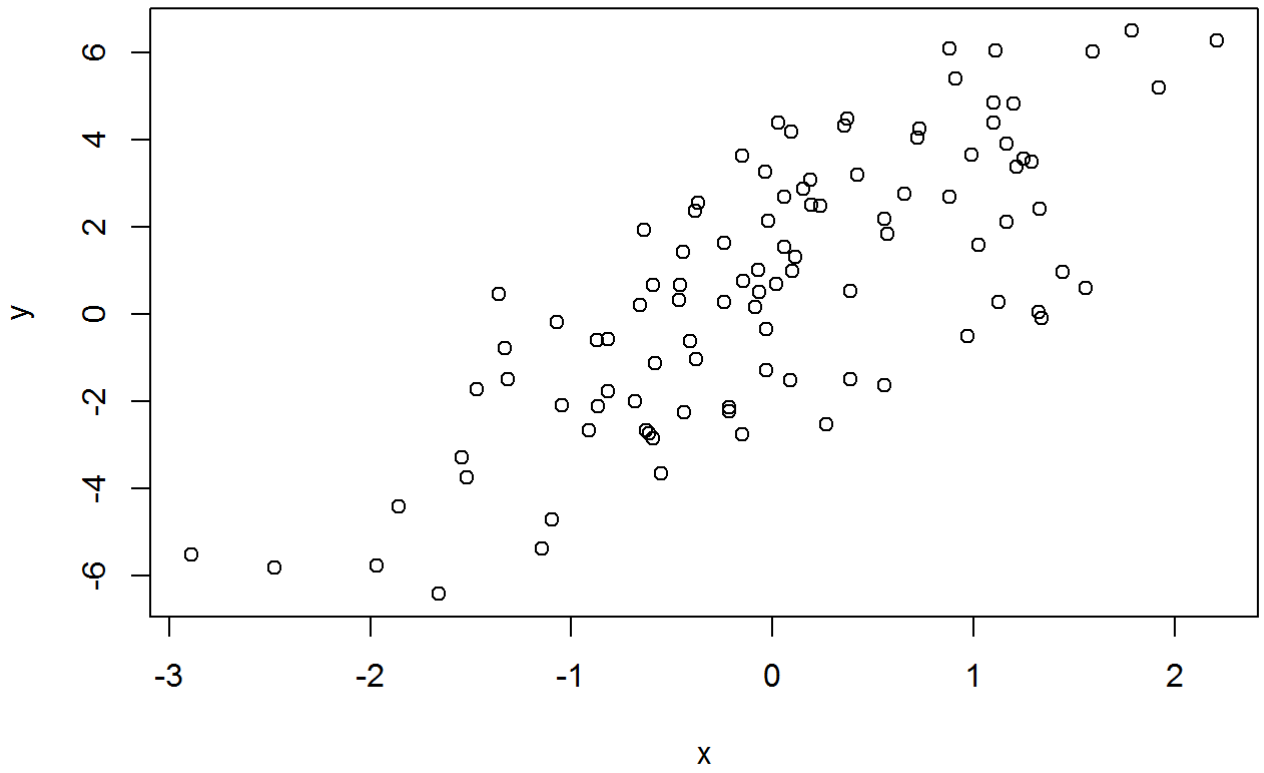
Assume

$x \sim \mathcal{N}(0, 1^2)$, $\beta_0 = 0.5$ and $\beta_1 = 2$

```
set.seed(20)
x <- rnorm(100)
e <- rnorm(100, 0, 2)
y <- 0.5 + 2 * x + e
summary(y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -6.4080 -1.5400  0.6789  0.6893  2.9300  6.5050
```

```
plot(x, y)
```

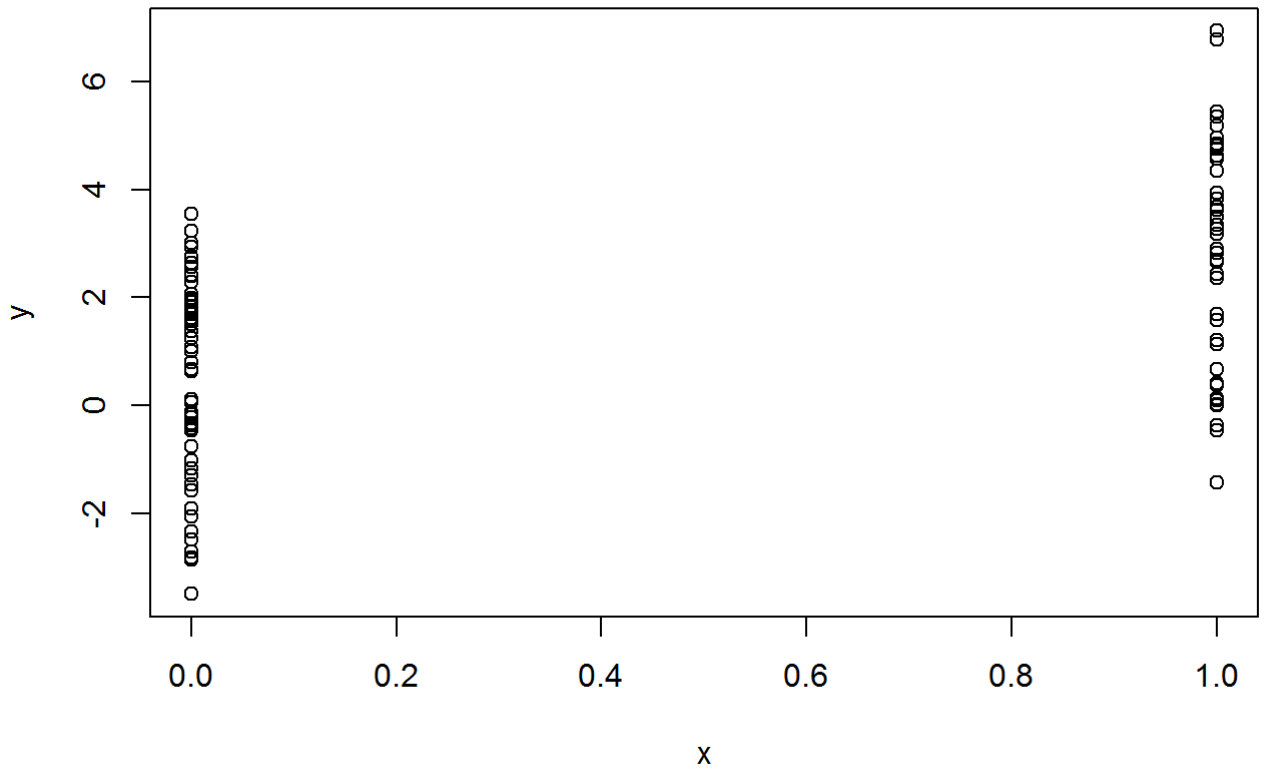


E se `x` for binário?

```
set.seed(10)
x <- rbinom(100, 1, 0.5)
e <- rnorm(100, 0, 2)
y <- 0.5 + 2 * x + e
summary(y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3.4940 -0.1409  1.5770  1.4320  2.8400  6.9410
```

```
plot(x, y)
```



Gerando números aleatórios a partir de um modelo linear generalizado

Suponha que queremos simular a partir de um modelo de Poisson onde

Y ~ Poisson(μ)

log μ = $\beta_0 + \beta_1 x$

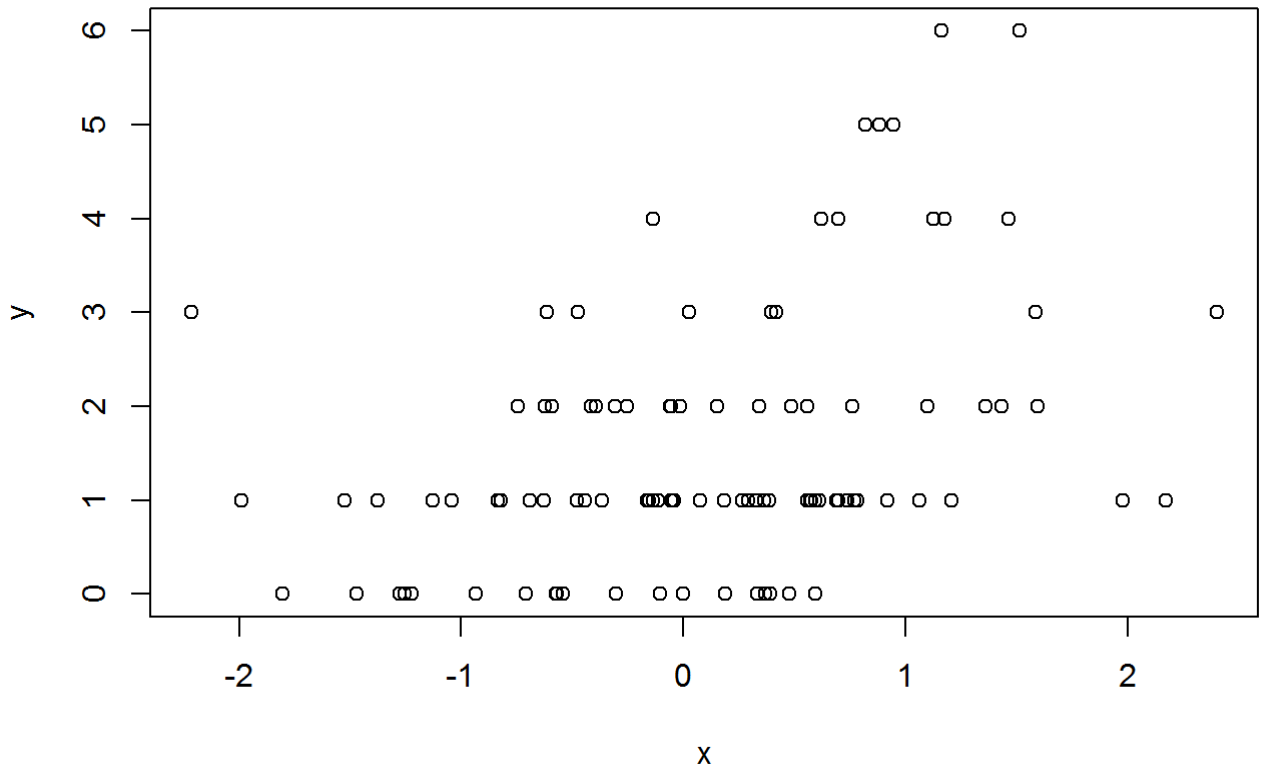
e $\beta_0 = 0.5$ and $\beta_1 = 0.3$

Precisamos usar a função `rpois` para este

```
set.seed(1)
x <- rnorm(100)
log.mu <- 0.5 + 0.3 * x
y <- rpois(100, exp(log.mu))
summary(y)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    1.00    1.00    1.55    2.00    6.00
```

```
plot(x, y)
```



Amostra randômica

A função `sample` desenha aleatoriamente a partir de um conjunto especificado de objetos (escalares), permitindo que você amostra de distribuições arbitrárias.

```
set.seed(1)
sample(1:10, 4)
```

```
## [1] 3 4 5 7
```

```
sample(1:10, 4)
```

```
## [1] 3 9 8 5
```

```
sample(letters, 5)
```

```
## [1] "q" "b" "e" "x" "p"
```

```
sample(1:10)  ## permutação
```

```
##  [1]  4  7 10  6  9  2  8  3  1  5
```

```
sample(1:10)
```

```
##  [1]  2  3  4  1  9  5 10  8  6  7
```

```
sample(1:10, replace = TRUE)  ## Amostra com substituição
```

```
##  [1] 2 9 7 8 2 8 5 9 7 8
```

Resumo

- As amostras de desenho de distribuições de probabilidade específicas podem ser feitas com funções `r*`
- As distribuições padrão são construídas em: Normal, Poisson, Binomial, Exponencial, Gamma, etc.
- A função `sample` pode ser usada para desenhar amostras aleatórias a partir de vetores arbitrários
- Definir a semente do gerador de números aleatórios via set.seed é crítico para a reprodutibilidade

.

The Scientist (http://www.thescientist.com.br)