Aula 3.2 - números aleatórios

Elder Sodre

Wednesday, February 25, 2015

#### Função set.seed()

Em aulas anteriores, vimos duas funções para criar sequências de números aleatórios:

* rnorm(n,mean=0,sd=1) Cria números com uma distribuição normal
* runif(n,min=0,max=1) Cria números com uma distribuição uniforme

O R, na verdade, cria números que parecem aleatórios, mas que podem ser reproduzidos.

Por mais que possa parecer estranho querer gerar a mesma sequência exata de números aleatórios mais de uma vez, isso tem aplicações práticas para reproduzir os resultados de uma análise, por exemplo.

Isto é feito com a função set.seed()

Vejamos alguns exemplos:

rnorm(5)

## [1] -0.70740 0.97733 0.06529 -0.52708 0.76546

rnorm(5)

## [1] 0.84585 -0.05682 1.43871 0.66008 -3.19592

set.seed(1) #Qualquer número inteiro pode ser usado para definir a seed.  
rnorm(5)

## [1] -0.6265 0.1836 -0.8356 1.5953 0.3295

set.seed(1)  
rnorm(5)

## [1] -0.6265 0.1836 -0.8356 1.5953 0.3295

Observe que nas 2 vezes acima que a função rnorm(5) foi chamada, exatamente a mesma sequência de números foi retornada.

Brinque um pouco com set.seed() e a criação de números aleatórios.

#### Funções para gerar números aleatórios

No R, existem uma série de funções que criam números aleatórios de acordo com uma determinada distribuição estatística. Todas elas são formadas por r+nome abreviado da distribuição.

Pode ser útil saber quais elas são. Abaixo há uma pequena lista com estas funções. Não se esqueça de olhar a ajuda da função se for do seu interesse.

* rnorm() #Distribuição normal
* runif() #Distribuição uniforme
* rbinom() #Distribição binomial
* rexp() #Distribuição exponencial
* rlnorm() #distribuição log-normal
* rpois() #Distribuição de poisson

#### Função sample()

A função sample() sorteia um ou mais números de um vetor.

sample (x,size,replace=FALSE,prob=NULL)

* x --> Um vetor com um ou mais elementos para o sorteio.
* size --> A quantidade de itens a serem sorteados
* replace=FALSE --> O sorteio tem reposição? Se replace=TRUE, um elemento que já foi sorteado pode sair de novo
* prob --> Um vetor de probabilidades, caso você quera os elementos tenham probabilidades diferentes de serem sorteados.

sample(1:6,size=1) #Sorteado um número de 1 a 6.

## [1] 2

sample(1:20,4) #Sorteados 4 números de 1 a 20.

## [1] 4 14 7 19

Se você não especificar o argumento size, size será igual ao comprimento de x.

sample(1:6)

## [1] 3 4 5 2 6 1

Quando replace=TRUE, um número pode sair mais de uma vez.

sample(1:10,8,replace=T)

## [1] 3 7 2 3 4 1 4 9

A função set.seed() também pode ser utilizada para obter resultados iguais na função sample.

set.seed(9)  
sample(1:5)

## [1] 2 1 4 3 5

set.seed(9)  
sample(1:5)

## [1] 2 1 4 3 5