- Tutorial
- Exercícios
- Apostila

8. Reamostragem e Simulação

Função sample

Criar um vetor de LETTERS com letras de “A” a “J” e aplique a ele a função `sample`. Se não utilizar nenhum argumento ela apenas embaralha os atributos do objeto. Com o argumento “replace=T”, ela reamostra cada elemento com reposição, e o argumento “prob” é a reamostragem de cada elemento com probabilidades diferentes.

```
vetor=rep(LETTERS[1:10])
vetor
sample(vetor)
sample(vetor, replace=T)
sample(vetor,40,replace=T)
sample(vetor,prob=c(0.1,0.2,0.05,0.05,0.2,0.1,0.05,0.05,0.1,0.1),replace=T)
## O argumento ''prob'' é padronizado para somar um:
sample(vetor,prob=c(1,2,0.5,0.5,2,1,0.5,0.5,1,1),replace=T) # o argumento
prob pode somar mais que um
```

Dados de mandíbula de Chacal Dourado

Vamos voltar aos dados de Chacal e à pergunta se há diferença no tamanho de mandíbulas entre machos e fêmeas

```
macho=c(120,107,110,116, 114, 111, 113,117,114,112)
femea=c(110,111,107, 108,110,105,107,106,111,111)
macho
femea
sexo=rep(c("macho", "femea"), each=10)
sexo
mf=c(macho,femea)
mf
macho.m=mean(macho)
macho.m
femea.m=mean(femea)
femea.m
macho.m-femea.m
dif.mf=diff(tapply(mf,sexo,mean))
dif.mf
```

PERGUNTAS:

- Essa diferença entre as médias é significativa?
- Qual minha incerteza ao afirmar que essas médias são diferentes?

Se a variação encontrada é devido à variações não relacionadas ao sexo, é possível gerar essa diferença permutando os dados. Caso isso seja verdade encontraremos frequentemente diferenças iguais ou maiores que a observada.

Permutando

```
s1.mf=sample(mf)
s1.mf
diff(tapply(s1.mf,sexo,mean))
##+1
s2.mf=sample(mf)
s2.mf
diff(tapply(s2.mf,sexo,mean))
##+2
diff(tapply(sample(mf),sexo,mean))
##+3
diff(tapply(sample(mf),sexo,mean))
##+1000
### e agora? fazer na mão 1000 vezes? ###
```

Criando ciclos de eventos

Vamos criar um loop!!!!!

```
result<-rep(NA,1000)
result[1]<-diff(tapply(mf,sexo,mean))
for(i in 2:1000)
    {
    dif.dados=diff(tapply(sample(mf),sexo,mean))
    result[i]<-dif.dados
    }
hist(result)
abline(v = result[1], col="red")
abline(v = result[1]*-1, col="red")
```

Cálculo do P

```
## Há diferença entre machos e fêmeas?

bicaudal=sum(result>=result[1]| result<=(result[1]*-1))
bicaudal
length(result)
```

```
p.bi=bicaudal/length(result)
p.bi

## Machos são maiores que as fêmeas?

unicaudal=sum(result>=result[1])
unicaudal
p.uni=unicaudal/length(result)
p.uni
```

Bootstrap

Vamos agora pegar o mesmo exemplo anterior e estimar o intervalo de confiança da média dos machos do chacal dourado. Primeiro vamos ver novamente esses dados e sua média:

```
macho
macho.m
```

Agora, partindo da premissa que esses dados representam o tamanho das mandíbulas do chacal dourado, podemos fazer uma reamostragem dos nossos dados e calcular novamente a média:

```
mean(sample(macho))
```

Essa média não é diferente da anterior, porque mudar a posição dos valores não afeta a estimativa da média. No entanto, se usarmos uma reamostragem com reposição (amostrar um valor e depois retorná-lo, antes de amostrar o próximo), permite que os valores já amostrados apareçam novamente na nova amostra. Vamos fazê-lo:

```
smacho<-sample(macho, replace=TRUE)
mean(smacho)
mean(sample(macho,replace=TRUE))
mean(sample(macho,replace=TRUE))
```

Perceba que as últimas linhas de comando produzem valores diferentes apesar de serem as mesmas. Esse processo é similar ao que usamos para fazer amostras de uma distribuição conhecida com o *rnorm()* e *rpois()*, só que agora os valores passíveis de serem amostrados são aqueles presentes nos nossos dados. Se repetirmos esse procedimento muitas vezes e guardarmos os resultados de cada simulação de amostras com reposição, teremos um conjunto de valores chamados pseudo-valores que representam a distribuição do nosso parâmetro e portanto podemos calcular o intervalo de confiança que desejarmos a partir dessa distribuição. Como repetimos uma operação muitas vezes no R? Usando novamente os ciclos produzidos pela função *for(... in ...)*, vamos fazer então 100 simulações:

```
nsim=100
resulta=rep(NA,nsim)
for(i in 1:nsim)
 {
  resulta[i]<-mean(sample(macho, replace=TRUE))
 }
```

```
## veja os valores calculados
resulta
```

Agora só falta calcular o intervalo de confiança para o limite que interessa (95%, 99%...). Vamos calcular para um intervalo de 90%. Uma forma de faze-lo é ordenando os valores e olhado quais valores estão nos extremos com 5% de cada lado.

```
sort(resulta)
sort(resulta)[6] ## o valore que deixa as 5 menores de fora
sort(resulta)[95] ## o valore que deixa os 5 maiores de fora
```

Podemos também usar a função *quantile()* definindo os quantis de interesse:

```
quantile(resulta, prob=c(0.05, 0.95))
```

Função Vegas

Em aula, se houve tempo, construímos uma função que automatiza a sequência de comandos da primeira parte desse tutorial onde testamos as hipóteses: (1) da mandíbulas de chacais machos e fêmeas serem diferentes e (2) a mandíbula de machos serem maiores que a das fêmeas, em média. Veja se você é capaz de entender o que a função faz a cada linha de comando e se estaria apto a explicá-la a outra pessoa.

função vegas.t

Agora use a função para testar as hipóteses novamente!

Tesourinha e a deriva continental

Vamos agora reproduzir a análise principal do estudo publicado na Nature em 1966 (*Geographical Distribution of the Dermaptera and the Continental Drift Hypothesis*) e descrita no primeiro capítulo do livro do Manly sobre permutação. A ideia era verificar se a ocorrência de taxa de tesourinhas (*Dermaptera*) estava mais correlacionada com a distribuição dos continentes atual ou antes da deriva continental. A informação que partimos é do coeficiente de correlação da ocorrência de taxa de tesourinha entre diferentes regiões biogeográficas: Eurasia, África, Madagascar, Oriente, Austrália, Nova Zelândia, América do Sul e América do norte. Valores positivos próximos a 1 representam composições de comunidades muito parecidas, valores próximos a -1 representam composição muito distintas. Vamos reconstruir essa matriz no objeto `data.cef`:

```
data.coef<-matrix(c(NA, .30, .14, .23, .30, -0.04, 0.02, -0.09, NA, NA,
.50,.50, .40, 0.04, 0.09, -0.06, NA, NA, NA, .54, .50, .11, .14, 0.05,
rep(NA, 4), .61, .03,-.16, -.16, rep(NA, 5), .15, .11, .03, rep(NA, 6), .14,
-.06, rep(NA, 7), 0.36, rep(NA, 8)), nrow=8, ncol=8)
rownames(data.coef) <- c("Eur_Asia", "Africa", "Madag", "Orient", "Austr",
"NewZea", "SoutAm", "NortAm")
colnames(data.coef) <- c("Eur_Asia", "Africa", "Madag", "Orient", "Austr",
"NewZea", "SoutAm", "NortAm")
```

```
data.coef
```

Foram usadas nesse estudo outras duas matrizes de distância, a primeira representando a distância atuais e a outra a distância geográfica antes da deriva continental das mesmas regiões biogeográficas.

```
dist.atual<-matrix(c(NA,1,2,1,2,3,2,1, NA, NA, 1,2,3,4,3,2, NA, NA,
NA,3,4,5,4,3, rep(NA, 4),1,2,3,2, rep(NA, 5), 1,4,3, rep(NA, 6), 5,4,
rep(NA, 7), 1, rep(NA, 8)), nrow=8, ncol=8)
dist.atual
dist.deriva<- matrix(c(NA,1,2,1,2,3,2,1, NA, NA, 1,1,1,2,1,2, NA, NA,
NA,1,1,2,2,3, rep(NA, 4),1,2,2,2, rep(NA, 5), 1,2,3, rep(NA, 6), 3,4,
rep(NA, 7), 1,  rep(NA, 8)), nrow=8, ncol=8)
# colocando nomes nas matrizes
rownames(dist.atual) <- colnames(dist.atual)<- c("Eur_Asia", "Africa",
"Madag", "Orient", "Austr", "NewZea", "SoutAm", "NortAm")

colnames(dist.deriva)<- rownames(dist.deriva)<-  c("Eur_Asia", "Africa",
"Madag", "Orient", "Austr", "NewZea", "SoutAm", "NortAm")
# olhando as matrizes
dist.atual
dist.deriva
```

A primeira parte da análise dos dados e ver qual a correlação entre a matriz de correlação taxonômica e as distâncias geográficas (atual e antes da deriva). Para isso vamos calcular um coeficiente de correlação de *Pearson* entre as matrizes. Esse valor irá nos dizer se as duas matrizes estão correlacionadas, ou seja, os valores de uma variam na mesma direção da outra (+1), em direção contrária (-1) ou não são correlacionadas (0).

$$ r = \frac{\sum_1^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_1^n{(x_i-\bar{x})^2}}\sqrt{\sum_1^n{(y_i-\bar{y})^2}}}$$

```
cor12<-cor(as.vector(data.coef), as.vector(dist.atual), use="complete.obs")
cor13<-cor(as.vector(data.coef), as.vector(dist.deriva), use="complete.obs")
cor12 ## correlação com a distancia atual
cor13 ## correlação com a distancia antes da deriva
```

Ambos os valores de correlação estão nos dizendo que quanto maior a distância geográfica mais diferente é a composição de espécies de tesourinha. Além disso, que a correlação com as distâncias antes da deriva é mais forte. No caso, valores maiores em módulo já que a relação é de correlação negativa (aumento da distância diminui a similaridade florística).

Agora precisamos calcular se esse valores de correlação poderiam ser atribuídos ao acaso. Para isso vamos fazer a permutação de uma das matrizes em e calcular o coeficientes de Pearson após essa permutação. A permutação é simples, vamos mudar as colunas e linhas de lugares de maneira a aleatorizar os valores mas manter a estrutura subjacente ao dados. Uma maneira de fazer é:

```
data.sim<-data.coef # copia da matriz que será aleatorizada
data.sim

# preenchendo o triangulo superior da matriz com os dados correspondentes do
```

```
triangulo inferior
data.sim[upper.tri(data.sim)] <- t(data.coef)[(upper.tri(data.coef))]

data.sim # olhando a matriz
data.sim[8:1, 8:1] # uma matriz baguncada mas que mantem certa estrutura
sim.pos<-sample(1:8) # posicoes permutadas
sim.pos
data.sim<-data.sim[sim.pos, sim.pos] # aqui uma matriz verdadeiramente
permutada
cor12.sim<-cor(as.vector(data.sim), as.vector(dist.atual),
use="pairwise.complete.obs")
cor13.sim<-cor(as.vector(data.sim), as.vector(dist.deriva),
use="pairwise.complete.obs")
cor12.sim
cor13.sim
cor12 ## correlação observada com a distancia atual
cor13 ## correlação observada com a distancia antes da deriva
#########################################################
### Repetir a simulação muitas vezes ####################
#########################################################
res.cor=data.frame(sim12=rep(NA, 5000), sim13=rep(NA,5000))
str(res.cor)
res.cor[1,]<-c(cor12, cor13)
str(res.cor)
for(s in 2:5000)
    {
        sim.pos<-sample(1:8)
        data.sim<-data.sim[sim.pos, sim.pos]
        res.cor[s,1]<-cor(as.vector(data.sim), as.vector(dist.atual),
use="pairwise.complete.obs")
        res.cor[s,2]<-cor(as.vector(data.sim), as.vector(dist.deriva),
use="pairwise.complete.obs")
    }
str(res.cor)
par(mfrow=c(2,1))
hist(res.cor[,1])
abline(v=res.cor[1,1], col="red")
hist(res.cor[,2])
abline(v=res.cor[1,2], col="red")
#### calculando o P ############
p12=sum(res.cor[,1]<= res.cor[1,1])/(dim(res.cor)[1])
p12
p13=sum(res.cor[,2]<= res.cor[1,2])/(dim(res.cor)[1])
p13
```