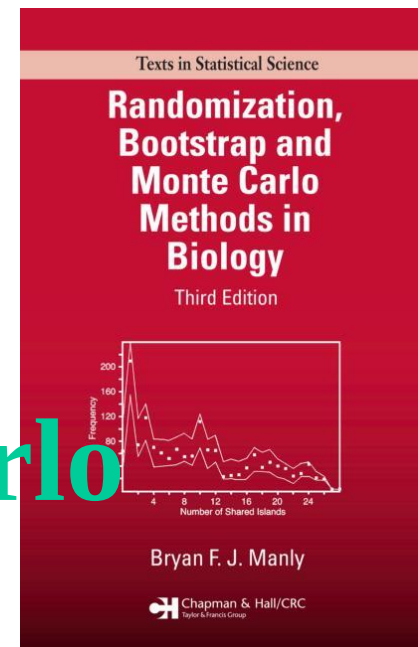


4

Métodos Monte Carlo



- Manly, B. F. J. *Randomization, Bootstrap and Monte Carlo Methods in Biology*. 3^a ed., Chapman & Hall/CR Press, Boca Raton, 2007.

ESQUEMA DO CAPÍTULO

4.1 GERAÇÃO DE NÚMEROS ALEATÓRIOS

4.2 ESTIMAÇÃO MONTE CARLO

4.3 TESTES MONTE CARLO

4.4 OBSERVAÇÕES FINAIS

4.1 Geração de Números Aleatórios

- Os métodos Monte Carlo baseiam-se na ideia de amostragem aleatória, ou seja, na **geração de números aleatórios**;
- O R fornece **várias** funções para geração de número aleatórios, por exemplo:
 - `rnorm(n, mean=0, sd=1)`
que gera n valores com **distribuição normal**, com média = 0 e desvio padrão = 1
 - `rexp(n, rate=1)`
que gera números de uma **distribuição exponencial** com n valores, com taxa $\lambda = 1$ (`rate=1`);
- Veja `help("distributions")`, para conhecer outras formas de gerar dados aleatórios com **diferentes distribuições**
- Há também a função `sample`, para realizar **amostras aleatórias**:
 - `sample(x, size=10, replace=FALSE)`
em que x é o conjunto de dados do qual as amostras serão retiradas, `size` é o tamanho de amostra e `replace` define se a amostra é feita com (TRUE) ou sem reposição (FALSE).

4.1 Geração de Números Aleatórios

Ex. 4.1.1:

- Teste as funções de geração de números aleatórios, via geração de amostras normal e exponencialmente distribuídas, de diversos tamanhos. Lembre-se que a definição de uma **semente** torna os seus resultados **reprodutíveis**.

```
> set.seed(13579)
> runif(n=6, min=0, max=1) # 6 números >uniformemente distribuídos
[1] 0.1084682 0.8792179 0.1051331 0.9017989 0.3994473 0.3237079

> rnorm(n=6, mean=0, sd=1) # 6 números normalmente distribuídos
[1] -1.5266466 1.0971147 2.4887442 0.7794803 0.1883750 -1.0264459

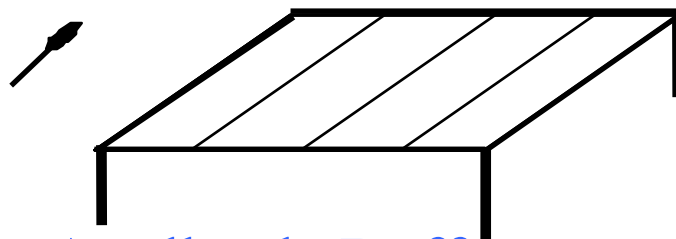
> rexp(n=6, rate=1)          # 6 números exponencialmente distribuídos
[1] 1.2879562 1.8696892 0.5443629 0.5804735 0.3625169 1.1793890

> sample(c(1,2,3,4,5,6),size=6,replace=TRUE) # 6 lançamentos de dados
[1] 2 2 2 5 2 4
```

4.2 Estimação Monte Carlo

- Os métodos de estimação Monte Carlo foram desenvolvidos no final dos anos 1940, após a Segunda Guerra Mundial;
- Entretanto, a ideia **não** era nova:
 - em 1777, o Conde de Buffon utilizou amostragem aleatória para o famoso experimento que ficou conhecido como o **Problema da Agulha de Buffon**.

4.2 Estimação Monte Carlo (cont.)



O Problema da Agulha de Buffon:

- é um método para estimar o valor de π (George Louis Leclerc, c. 1733);
- lança-se uma agulha de tamanho ℓ em uma mesa com linhas espaçadas pela distância d ($>\ell$);
- a probabilidade $P(\text{agulha cruzar a linha}) = 2\ell/(\pi d)$;
- $\hat{p}$ = proporção de vezes que a linha é cruzada;
- portanto, pode-se estimar π por $2\ell/(\hat{p}d)$.

4.2 Estimação Monte Carlo (cont.)

O **Problema da Agulha de Buffon** parece ingênuo, mas apresenta importantes características de simulações:

- experimento para *estimar* algo difícil de calcular exatamente (pelo menos, em 1733);
- *aleatoriedade*, o que torna a estimativa não exata; estimativa do erro de estimação;
- *replicação* (quanto mais, melhor) para reduzir o erro;
- *amostragem sequencial* para controle do erro – continue lançando até que o erro seja suficientemente pequeno;
- *redução de variância* (Buffon Cross).

4.2 Estimação Monte Carlo (cont.)

4.2.1 Estimador Monte Carlo simples

Considere o problema de estimar $\theta = \int_0^1 g(x)dx$. Se $X_1, \dots, X_n$ é uma amostra aleatória Uniforme(0,1), então

$$\hat{\theta} = \overline{g_m(X)} = \frac{1}{m} \sum_{i=1}^m g(X_i)$$

converge para $E[g(X)]$, com probabilidade 1, pela *Lei Forte dos Grandes Números*.

Portanto, um estimador simples para $\int_0^1 g(x)dx$ é $\overline{g_m(X)}$.

4.2 Estimação Monte Carlo (cont.)

4.2.1 Estimador Monte Carlo simples (cont.)

Para estimar $\theta = \int_a^b g(x)dx$, faz-se a transformação linear, $y = (x-a)/(b-a)$ e $dy = (1/(b-a))dx$.

Por exemplo,

$$\theta = \int_a^b g(x)dx = \int_0^1 g[y(b-a) + a](b-a)dy$$

é $(b-a)$ vezes o valor esperado de $g(Y)$, em que Y tem distribuição uniforme em $(0,1)$.

A integral é, portanto, $(b-a)$ vezes o valor esperado de $g(\cdot)$ em $(0,1)$.

4.2 Estimação Monte Carlo (cont.)

4.2.1 Estimador Monte Carlo simples

Ex. 4.2.1.:

Calcule uma estimativa Monte Carlo para

$$\theta = \int_0^1 e^{-x} dx$$

e compare com o valor exato, $\theta = 1 - e^{-1}$.

Solução:

O problema pode ser resolvido com as seguintes linhas de código em R:

```
m <- 1000
x <- runif(m)
theta.chapeu <- mean(exp(-x))
print(theta.chapeu)
print(1-exp(-1))
```

```
[1] 0.6328156
```

```
[1] 0.6321206
```

4.2 Estimação Monte Carlo (cont.)

4.2.1 Estimador Monte Carlo simples

Ex. 4.2.2.:

Use a abordagem Monte Carlo para estimar a função de distribuição acumulada da Normal fda:

$$\Phi(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$$

Solução:

O problema pode ser dividido em 2 casos, $x \geq 0$ e $x \leq 0$, e uso da simetria da densidade da normal. Assim, o problema é estimar:

$$\theta = \int_0^x e^{-t^2/2} dt, \quad x > 0$$

Isso pode ser feito, gerando-se números aleatórios Uniforme(0,x), ou, de forma mais simples, pela mudança de variáveis, $t = xy$ e $dt = xdy$:

$$\theta = \int_0^1 x e^{-(xy)^2/2} dy.$$

4.2 Estimação Monte Carlo (cont.)

4.2.1 Estimador Monte Carlo simples

Ex. 4.2.2.:

Solução (cont.):

Tem-se, assim

$$\hat{\theta} = \overline{g_m(u)} = \frac{1}{m} \sum_{i=1}^m x e^{-(u_i x)^2 / 2}.$$

em que a variável aleatória u tem distribuição Uniforme(0,1). A média amostral $\hat{\theta}$ converge para θ , com $m \rightarrow \infty$. Se $x > 0$, a estimativa é $0,5 + \theta / \sqrt{2\pi}$. Caso contrário, calcule $\Phi(x) = 1 - \Phi(x)$. Veja código em R:

```
x<-seq(0.1,1.0,length=10)
rep<-10000
u<-runif(rep)
fda<-numeric(10)
for (i in 1:10) {
  g<-x[i]*exp(-(u*x[i])^2/2)
  fda[i]<-0.5+mean(g)/sqrt(2*pi)}
print(x)
print(fda)
```

4.3 Testes Monte Carlo

- Em um teste Monte Carlo:
 - a significância de uma estatística de teste observada é determinada por **comparação** com amostras desta estatística obtidas de amostras aleatórias sob algum modelo assumido;
- Se o modelo assumido implica que todas as ordenações dos dados são igualmente prováveis, então isso conduz a um *teste de permutação*, que pode ser visto como caso **particular** dos testes Monte Carlo;
- Similarmente, o *bootstrap* também pode ser visto com um teste Monte Carlo aplicado de uma maneira **particular**;
- Entretanto, há testes Monte Carlo que **não** podem ser vistos como testes de permutação ou *bootstrap*.

4.3 Testes Monte Carlo (cont.)

Exemplo 4.3.1 – Teste de Aleatoriedade Espacial:

- Suponha que a [Figura](#) mostre as posições de 24 plantas em uma área de $2 \times 2 \text{ m}^2$. A questão a ser considerada é se as plantas estão em posições aleatórias. Para ser mais específico, a hipótese nula é que cada uma das plantas seja igualmente provável de ocorrer em qualquer lugar dentro da área, independente das demais plantas.

Várias estatísticas de teste que poderiam ser utilizadas para sumarizar os dados. A primeira que será empregada é a g_1 , que é a distância média entre uma planta e sua vizinha mais próxima. Como há 24 plantas, haverá 24 distâncias contribuindo para o cálculo da média. A segunda é a g_2 , a distância média entre as plantas e sua segunda vizinha mais próxima, que será também a média de 24 valores. E assim por diante.

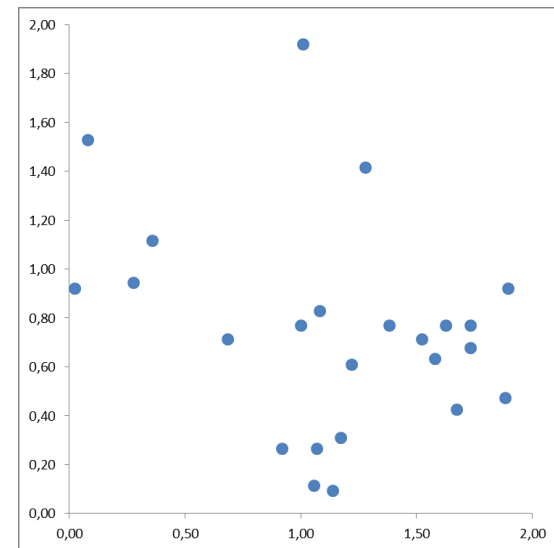


Figura: Posição de 24 plantas

4.3 Testes Monte Carlo (cont.)

Exemplo 4.3.1 – Teste de Aleatoriedade Espacial (cont.):

- A [Tabela](#) mostra os valores das estatísticas g_1 até g_{10} . Para determinação da significância, 999 dados aleatórios foram gerados, de tal forma que 24 pontos fossem alocados em posições aleatórias. O nível de significância para os g_i 's é estimado como sendo a proporção de vezes que se obtém valores simulados mais extremos que os observados na amostra original. A segunda e a terceira linhas mostram essas estimativas.
- Pode ser observado que as distâncias entre as plantas a partir dos seus 3º vizinhos mais próximos e superiores são menores que o razoável de se esperar sob completa aleatoriedade (significância $< 0,05$).

[Tabela](#): Estatísticas do vizinho mais próximo para as 24 plantas

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
Observado	0,203	0,292	0,345	0,408	0,499	0,565	0,611	0,651	0,716	0,761
Média	0,223	0,345	0,444	0,529	0,608	0,680	0,749	0,814	0,878	0,939
Significância	0,226	0,065	0,011	0,003	0,019	0,017	0,012	0,009	0,014	0,012

4.3 Testes Monte Carlo (cont.)

Exemplo 4.3.1 – Teste de Aleatoriedade Espacial (cont.):

- Uma implementação em R da função g_i , que recebe as coordenadas cartesianas x e y das plantas e retorna as distâncias médias do 1º até o i -ésimo vizinho mais próximo, é apresentada a seguir:

```
gi<-function(x,y,i) {
  n<-length(x)
  dist<-matrix(data=0,nrow=n,ncol=n)
  for (j in 1:(n-1)) {
    for (k in (j+1):n) {
      dist[j,k]<-sqrt((x[j]-x[k])^2+(y[j]-y[k])^2)
      dist[k,j]<-dist[j,k] }
    dist[j,]<-sort(dist[j,]) }
  dist[n,]<-sort(dist[n,])
  gi<-numeric(i)
  for (j in 1:i) {
    gi[j]<-mean(dist[, (j+1)]) }
  return(gi)}
```

4.3 Testes Monte Carlo (cont.)

Exemplo 4.3.1 – Teste de Aleatoriedade Espacial (cont.):

- As coordenadas cartesianas x e y das 24 plantas são:

$x \leftarrow$

$c(1.88, 1.90, 1.67, 1.73, 1.73, 1.58, 1.63, 1.52, 1.38, 1.28, 1.14, 1.17, 1.22, 1.06, 1.07, 1.08, 1.01, 1.00, 0.92, 0.69, 0.36, 0.28, 0.02, 0.08)$

$y \leftarrow$

$c(0.47, 0.92, 0.43, 0.68, 0.77, 0.63, 0.77, 0.71, 0.77, 1.41, 0.09, 0.31, 0.61, 0.11, 0.26, 0.83, 1.92, 0.77, 0.26, 0.71, 1.11, 0.94, 0.92, 1.53)$

- As distâncias médias, do 1º ao 10º vizinhos mais próximos, podem ser calculadas pelo trecho de código em R a seguir:

calcular distâncias médias para os dados

$gi.dados \leftarrow gi(x, y, i=10)$

$gi.dados$

4.3 Testes Monte Carlo (cont.)

Exemplo 4.3.1 – Teste de Aleatoriedade Espacial (cont.):

- A simulação Monte Carlo pode ser feita em R como se segue:

```
# simulação Monte Carlo
i<-10
rep<-999
gi.matrix<-
matrix(data=0,nrow=rep,ncol=i)
set.seed(13579)
for (j in 2:rep) {
  # gerar amostras
  xsim<-runif(24,min=0,max=2)
  ysim<-runif(24,min=0,max=2)
  # calcular gi
  gi.matrix[j,]<-gi(xsim,ysim,i)
}
```

- Os valores médios e a significância podem ser calculados pelo seguinte trecho de código em R:

```
# calcular valores médios
medias<-numeric(i)
for (j in 1:i) {
  medias[j]<-mean(gi.matrix[,j])}
medias
# calcular significancia
signif<-numeric(i)
for (j in 1:i) {
  signif[j]<-1
  for (k in 1:rep) {
    if (gi.dados[j]>gi.matrix[k,j])
    {
      signif[j]<-signif[j]+1 }
    }
  signif[j]<-signif[j]/(rep+1)
}
signif
```

4.4 Observações Finais

- Um teste de Monte Carlo é aquele em que uma estatística de teste observada, que foi calculada a partir de um conjunto de dados, é avaliada por comparação com uma distribuição obtida pela geração de um conjunto alternativo de dados, por algum modelo assumido;
- Com esta definição, fica claro que testes que envolvem aleatorização (como os testes de permutação) e testes bootstrap são casos especiais de testes Monte Carlo.