Tareas de Cómputo Científico para Probabilidad y Estadística

Alumnos del curso, Ago-Dic 2017

Índice

1. Descomposición LU y Cholesky

1. Implementar los algoritmos de *Backward* y *Forward substitution*.

2. Implementar el algoritmo de eliminación gaussiana con pivotteo parcial LUP, 21.1 del Trefethen (p. 160).

3. Dar la descomposición LUP para una matriz aleatoria de entradas $U(0,1)$ de tamaño 5×5, y para la matriz

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 1 \\ -1 & -1 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 \end{pmatrix} \tag{1}$$

4. Usando la descomposición LUP anterior, resolver el sistema de la forma

$$Dx = b \tag{2}$$

   donde D son las matrices del problema 3, para 5 diferentes b aleatorios con entradas $U(0,1)$. Verificando si es o no posible resolver el sistema.

5. Implementar el algoritmo de descomposición de Cholesky 23.1 del Trefethen (p. 175).

6. Comparar la complejidad de su implementación de los algoritmos de factorización de Cholesky y LUP mediante la medición de los tiempos que tardan con respecto a la descomposición de una matriz aleatoria hermitiana definida positiva. Graficar la comparación.

2. Descomposición QR y mínimos cuadrados

1. Implementar el algoritmo de Gram-Schmidt modificado 8.1 del Trefethen (p. 58) para generar la descomposición QR.

2. Implementar el algoritmo que calcula el estimador de mínimos cuadrados en una regresión usando la descomposición QR.

3. Generar $\mathbf{Y}$ compuesto de $y_i = sen(x_i) + \epsilon_i$ donde $\epsilon_i \sim N(0, \sigma)$ con $\sigma = 0.11$ para $x_i = \frac{4\pi i}{n}$ para $i = 1, \ldots, n$.

   Hacer un ajuste de mínimos cuadrados a $\mathbf{Y}$, con descomposición QR, ajustando un polinomio de grado $p - 1$.

   - Considerar los 12 casos: $p = 3, 4, 6, 100$ y $n = 100, 1000, 10000$.
   - Graficar el ajuste en cada caso.
   - Medir tiempo de ejecución de su algoritmo, comparar con descomposición QR de scipy y graficar los resultados.

4. Hacer $p = 0.1n$, o sea, diez veces más observaciones que coeficientes en la regresión, ¿Cual es la n máxima que puede manejar su computadora?

3. Estabilidad

1. Sea A una matriz de tamaño 20×50 creenla aleatoreamente, fíjenla y calculen su descomposición QR. Sean $\lambda_1 > \lambda_2 > ... \geq \lambda_{20} = 1 > 0$ y

   $$B = Q^* diag(\lambda_1, \lambda_2, ..., \lambda_{20})Q \text{ y } B_\varepsilon = Q^* diag(\lambda_1 + \varepsilon_1, \lambda_2 + \varepsilon_2, ..., \lambda_{20} + \varepsilon_{20})Q,$$

   con $\varepsilon_i \sim N(0, \sigma)$, con $\sigma = 0.01\lambda_{20} = 0.01$.

   a) Comparar la descomposición de Cholesky de B y de B_ε usando el algoritmo de la tarea 1. Considerar los casos cuando B tiene un *buen* número de condición y un *mal* número de condición.

   b) Con el caso mal condicionado, comparar el resultado de su algoritmo con el del algoritmo de Cholesky de scipy.

   c) Medir el tiempo de ejecución de su algoritmo de Cholesky con el de scipy.

2. Resolver el problema de mínimos cuadrados,

   $$y = X\beta + \varepsilon, \ \varepsilon_i \sim N(0, \sigma)$$

   usando su implementación de la descomposición QR; β es de tamaño $n \times 1$ y X de tamaño $n \times d$.

   Sean $d = 5, n = 20, \beta = (5, 4, 3, 2, 1)'$ y $\sigma = 0.15$.

a) Hacer X con entradas aleatorias $U(0,1)$ y simular y. Encontrar $\hat{\beta}$ y compararlo con el obtenido $\hat{\beta}_p$ haciendo $X + \Delta X$, donde las entradas de ΔX son $N(0, \sigma = 0.01)$. Comparar a su vez con $\hat{\beta}_c = ((X+\Delta X)'(X+\Delta X))^{-1}(X+\Delta X)y$ usando el algoritmo genérico para invertir matrices scipy.linalg.inv .

b) Lo mismo que el anterior pero con X mal condicionada (ie. con casi colinealidad).

4. Cálculo de eigenvalores

1. Dado el siguiente
   Teorema (Gershgorin):
   Dada una matriz $A = a_{ij}$ de $m \times m$, cada eigenvalor de A está en al menos uno de los discos en el plano complejo con centro en a_{ii} y radio $\sum_{j \neq i} |a_{ij}|$. Además, si n de estos discos forman un dominio conexo, disjunto de los otros $m - n$ discos, entonces hay exactamente n eigenvalores en ese dominio.

   Deduce estimaciones de los eigenvalores de

$$A = \begin{pmatrix} 8 & 1 & 0 \\ 1 & 4 & \epsilon \\ 0 & \epsilon & 1 \end{pmatrix}$$

2. Implementa la iteración QR con shift. Aplícala a la matriz A del Ejercicio 1 con $\epsilon = 10^N$ para $N = 1, .., 5$.

3. Determina todos los eigenvalores y eigenvectores de una matriz de Householder.

4. Demuestra que no es posible construir la transformación de similaridad del teorema de Schur con un número finito de transformaciones de similaridad de Householder.

5. ¿Qué pasa si aplicas la iteración QR sin shift a una matriz ortogonal?

5. Simulación Estocástica, introducción

1. Definir la cdf inversa generalizada F_X^- y demostrar que en el caso de variables aleatorias continuas esta coincide con la inversa usual. Demostrar además que en general para simular de X podemos simular $u \sim U(0,1)$ y $F_X^-(u)$ se distribuye como X.

2. Implementar el siguiente algoritmo para simular variables aleatorias uniformes:

$$x_i = 107374182 x_{i-1} + 104420 x_{i-5} \mod 2^{31} - 1$$

regresa x_i y recorrer el estado, esto es $x_{j-1} = x_j; j = 1, 2, 3, 4, 5$; ¿parecen $U(0,1)$?

3. ¿Cuál es el algoritmo que usa *scipy.stats.uniform* para generar números aleatorios ? ¿Cómo se pone la semilla? ¿y en R?

4. ¿En *scipy* que funciones hay para simular una variable aleatoria genérica discreta? ¿tienen preproceso?

5. Implementar el algoritmo Adaptive Rejection Sampling y simular de una $Gama(2,1)$ 10,000 muestras. ¿cuando es conveniente dejar de adaptar la envolvente?

6. MCMC: Metropolis-Hastings

1. Simular $n = 5$ y $n = 35$ v.a Bernoulli $Be(1/3)$; sea r el número de éxitos en cada caso.

2. Implementar el algoritmo Metropolis-Hastings para simular de la posterior

$$f(p|\bar{x}) \propto p^r (1-p)^{n-r} \cos(\pi p) I_{[0,\frac{1}{2}]}(p),$$

con los dos casos de n y r de arriba. Para ello poner la propuesta $(p'|p) = p' \sim Beta(r+1, n-r+1)$ y la distribución inicial de la cadena $\mu \sim U(0, \frac{1}{2})$.

3. Argumentar porque la cadena es f-irreducible y porque es ergódica. Implementar el algoritmo con los datos descritos y discutir los resultados.

4. Implementar el algoritmo Metropolis-Hastings con la posterior de arriba tomando una propuesta diferente.

7. MCMC: Metropolis-Hastings II

Con el algoritmo Metropolis-Hastings (MH), simular lo siguiente:

1. Sean $x_i \sim Ga(\alpha, \beta); i = 1, 2, \ldots, n$. Simular datos x_i con $\alpha = 3$ y $\beta = 100$ considerando los casos $n = 3$ y 30.

   Con $\alpha \sim$ U(1,4), $\beta \sim$ exp(1) distribuciones a priori, se tiene la posterior

   $$f(\alpha, \beta|\bar{x}) \propto \frac{\beta^{n\alpha}}{\Gamma(\alpha)^n} r_1^{\alpha-1} e^{-\beta(r_2+1)} 1(1 \leq \alpha \leq 4) 1(\beta > 1),$$

   con $r_2 = \sum_{i=1}^{n} x_i$ y $r_1 = \prod_{i=1}^{n} x_i$.

   En ambos casos, grafica los contornos para visualizar dónde está concentrada la posterior.

   Utilizar la propuesta

   $$q\left(\begin{pmatrix}\alpha_p \\ \beta_p\end{pmatrix} \mid \begin{pmatrix}\alpha \\ \beta\end{pmatrix}\right) = \begin{pmatrix}\alpha \\ \beta\end{pmatrix} + \begin{pmatrix}\varepsilon_1 \\ \varepsilon_2\end{pmatrix},$$

   donde

   $$\begin{pmatrix}\varepsilon_1 \\ \varepsilon_2\end{pmatrix} \sim \mathcal{N}_2\left(\begin{pmatrix}0 \\ 0\end{pmatrix}, \begin{pmatrix}\sigma_1^2 & 0 \\ 0 & \sigma_2^2\end{pmatrix}\right).$$

2. Simular de la distribución Gamma(α,1) con la propuesta Gamma([α], 1), donde [α] denota la parte entera de α.

   Además, realizar el siguiente experimento: poner como punto inicial $x_0 = 1,000$ y graficar la evolución de la cadena, es decir,
   $f(X_t)$ vs t.

3. Implementar Random Walk Metropolis Hasting (RWMH) donde la distribución objetivo es $\mathcal{N}_2(\mu, \Sigma)$, con

   $$\mu = \begin{pmatrix}3 \\ 5\end{pmatrix} \quad \Sigma = \begin{pmatrix}1 & 0.9 \\ 0.9 & 1\end{pmatrix}.$$

Utilizar como propuesta $\varepsilon_t \sim \mathcal{N}_2(\mathbf{0}, \sigma I)$. ¿Cómo elegir σ para que la cadena sea eficiente? ¿Qué consecuencias tiene la elección de σ?

Como experimento, elige como punto inicial $x_o = \begin{pmatrix} 1000 \\ 1 \end{pmatrix}$ y comenta los resultados.

Para todos los incisos del ejercicio anterior:

- Establece cual es tu distribución inicial.
- Grafica la evolución de la cadena.
- Indica cuál es el Burn-in.
- Comenta qué tan eficiente es la cadena.
- Implementa el algoritmo MH considerando una propuesta diferente.

8. MCMC: MH con Kerneles Híbridos y Gibbs Sampler

1. Aplique el algoritmo de Metropolis-Hastings considerando como función objetivo la distribución normal bivariada:

$$f_{X_1,X_2}(\bar{x}) = \frac{1}{2\pi}|\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\bar{x}-\mu)'\Sigma^{-1}(\bar{x}-\mu)\right\}$$

donde,

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \quad \Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

Así, se tienen las siguientes distribuciones condicionales:

$$X_1|X_2 = x_2 \sim N\left(\mu_1 + \rho\frac{\sigma_1}{\sigma_2}(x_2-\mu_2)\ ,\ \sigma_1^2(1-\rho^2)\right)$$

$$X_2|X_1 = x_1 \sim N\left(\mu_2 + \rho\frac{\sigma_2}{\sigma_1}(x_1-\mu_1)\ ,\ \sigma_2^2(1-\rho^2)\right)$$

Considere las siguientes propuestas:

$$q_1\left((x_1', x_2')|(x_1, x_2)\right) = f_{X_1|X_2}(x_1'|x_2)1(x_2' = x_2)$$

$$q_2\left((x_1', x_2')|(x_1, x_2)\right) = f_{X_2|X_1}(x_2'|x_1)1(x_1' = x_1)$$

A partir del algoritmo MH usando Kerneles híbridos simule valores de la distribución normal bivariada, fijando $\sigma_1 = \sigma_2 = 1$, considere los casos $\rho = 0.8$ y $\rho = 0.99$[1]

2. Consideré los tiempos de falla $t_1, \ldots t_n$ con distribución $Weibull(\alpha, \lambda)$:

$$f(t_i|\alpha, \lambda) = \alpha\lambda t_i^{\alpha-1} e^{-t_i^\alpha \lambda}$$

Se asumen como a priori $\alpha \sim exp(c)$ y $\lambda|\alpha \sim Gama(\alpha, b)$, por lo tanto, $f(\alpha, \lambda) = f(\lambda|\alpha)f(\alpha)$[2]. Así, para la disitribución posterior se tiene:

$$f(\alpha, \lambda|\bar{t}) \propto f(\bar{t}|\alpha, \lambda)f(\alpha, \lambda)$$

A partir del algoritmo MH usando Kerneles híbridos simule valores de la distribución posterior $f(\alpha, \lambda|\bar{t})$, considerando las siguientes propuestas:

Propuesta 1:

$\lambda_p|\alpha, \bar{t} \sim Gama\left(\alpha + n\ ,\ b + \sum_{i=1}^{n} t_i^\alpha\right)$ y dejando α fijo.

Propuesta 2:

[1]Ver la tesis de Cricelio Montesinos para una explicación más extensa del Gibbs, Montesinos, C (2016) "Distribución de Direcciones en el Gibbs Sampler Generalizado", MSc Dissertation, CIMAT. https://www.cimat.mx/es/Tesis_digitales/. También vean la Enciclopedia de Estadística de Wiley, la entrada de *Gibbs Sampler*: https://www.cimat.mx/~jac/2016WileytStatsRef_GibbsSampling.pdf.

[2]Este ejemplo aparece en Kundu, D. (2008), "Bayesian Inference and Life Testing Plan for the Weibull Distribution in Presence of Progressive Censoring", *Technometrics*, **50**(2), 144–154

$\alpha_p|\lambda, \bar{t} \sim Gama\,(n+1\,,\ -\log(b) - \log(r_1) + c)$, con $r_1 = \prod_{i=1}^{n} t_i$ y dejando λ fijo.

Propuesta 3:

$\alpha_p \sim exp(c)$ y $\lambda_p|\alpha_p \sim Gama(\alpha_p, b)$.

Propuesta 4 (RWMH):

$\alpha_p = \alpha + \epsilon$, con $\epsilon \sim N(0, \sigma)$ y dejando λ fijo.

Simular datos usando $\alpha = 1$ y $\lambda = 1$ con $n = 20$. Para la a priori usar $c = 1$ y $b = 1$.

3. Considere el ejemplo referente al número de fallas de bombas de agua en una central nuclear[3], donde p_i representa el número de fallas en el tiempo de operación t_i, con $i = 1, \ldots n$.

   Se considera el modelo $p_i \sim Poisson(\lambda_i t_i)$, (las λ_i son independientes entre si), con distribuciones a priori $\lambda_i|\beta \sim Gama(\alpha, \beta)$ y $\beta \sim Gama(\gamma, \delta)$, por lo tanto:

$$f(\lambda_1, \ldots, \lambda_n, \beta) = f(\lambda_1|\beta) f(\lambda_2|\beta) \ldots f(\lambda_n|\beta) f(\beta)$$

   Para la distribución posterior se tiene:

$$f(\lambda_1, \ldots, \lambda_n, \beta|\bar{p}) \propto L(\bar{p}, \bar{\lambda}, \beta) f(\lambda_1, \ldots, \lambda_n, \beta)$$

   Simule valores de la distribución posterior $f(\lambda_1, \ldots, \lambda_n, \beta|\bar{p})$, usando un kernel híbrido, considerando las propuestas:

$$\lambda_i|\bar{\lambda}_{-i}, \beta, \bar{t} \sim Gama(t_i p_i + \alpha\,,\ \beta + 1)$$

[3]Este ejemplo fué usado en el artículo original del Gibbs sampler del Gelfand y Smith (1990). Vea también Norton, R.A., Christen, J.A. y Fox, C. (2017), "Sampling hyperparameters in hierarchical models: improving on Gibbs for high-dimensional latent fields and large data sets" *Communications in Statistics - Simulation and Computation*, http://dx.doi.org/10.1080/03610918.2017.1353618 .

Bomba (i)	1	2	3	4	5	6	7	8	9	10
T. de uso (t_i)	94.32	15.72	62.88	125.76	5.24	31.44	1.05	1.05	2.1	10.48
# de fallas (p_i)	5	1	5	14	3	18	1	1	4	22

Cuadro 1: Datos de bombas de agua en centrales nucleares (Robert y Casella, p. 385) para el ejemplo 8.3.

$$\beta|\bar{\lambda}, \bar{t} \ \sim \ Gama\left(n\alpha + \gamma \ , \ \delta + \sum_{i=1}^{n} \lambda_i\right).$$

Verifique que estas son propuestas Gibbs.

Use los datos del Cuadro 1 con los parámetros a priori $\alpha = 1.8$, $\gamma = 0.01$ y $\delta = 1$.

9. MCMC: Tarea Final

1. **(Problema en ecología)** Sean $X_1, ...X_m$ variables aleatorias donde X_i denota el número de individuos de una especie en cierta región. Suponga que $X_i|N, p \sim$ Binomial(N, p), entonces

$$f(\bar{x}|N, p) = \prod_{i=1}^{m} \frac{N!}{x_i!(N - x_i)!} p^{x_i}(1 - p)^{N - x_i}.$$

Asumiendo la distribución a priori $p \sim$ Beta(α, β) y $N \sim h(\cdot)$, donde h es una dist. discreta en $\{0, 1, 2, \ldots, N_{max}\}$, se tiene definida la distribución posterior $f(N, P|\bar{x})$.

A partir del algoritmo MH, simule valores de la distribución posterior usando un kernel híbrido. Para ello considere **como sugerencia** la siguiente distribución inicial para el MCMC

$$p \sim \mathrm{U}(0, 1) \quad \text{y} \quad N \sim \mathrm{U}_d\left\{\max_{i\in\{1,...m\}}(x_i), \max_{i\in\{1,...m\}}(x_i) + 1, ..., N_{max}\right\}$$

y las propuestas

- Propuesta 1: De la condicional total de p (kernel Gibbs).

- Propuesta 2: De la a priori.
- Propuesta 3: Propuesta hipergeométrica (¿?).
- Propuesta 4: Poisson: $N_p \sim \text{máx}_{i\in\{1,\dots m\}}(x_i) + \text{Poisson}(?)$.
- Propuesta 5: Caminata aleatoria

$$N_p = N + \epsilon, \quad \mathbb{P}(\epsilon = 1) = \frac{1}{2} = \mathbb{P}(\epsilon = -1).$$

Los datos son estos: 6, 4, 9, 7, 8, 2, 8, 7, 5, 5, 3, 9, 4, 5, 9, 8, 7, 5, 3, 2; $m = 20$. A priori, esperamos que sea difícil observar a los individuos entonces $\alpha = 1, \beta = 20$. La especie no es muy abundante y entonces $N_{max} = 1000$ y $h(N) = 1/(N_{max} + 1); N \in \{0, 1, 2, \dots, N_{max}\}$.

Las propuestas y distribución inicial para el MCMC de arriba son **solamente sugerencia**, propongan otras propuestas, experimenten y comenten.

2. (**Estudio de mercado**) Se tiene un producto y se realiza una encuesta con el fin de estudiar cuánto se consume dependiendo de la edad. Sea Y_i el monto de compra y X_i la covariable la cual representa la edad.

   Suponga que $Y_i \sim Po(\lambda_i)$ (distribución Poisson con intensidad λ_i)

$$\lambda_i = c g_b(x_i - a)$$

   para g_b la siguiente función de liga

$$g_b(x) = \exp\left(-\frac{x^2}{2b^2}\right).$$

   O sea, se trata de regresión Poisson con una función liga no usual. Si $\lambda_i = 0$ entonces $P(Y_i = 0) = 1$. $a =$ años medio del segmento (años), $c =$ gasto promedio (pesos), $b =$ "amplitud" del segmento (años).

   Considere las distribuciones a priori

$$a \sim N(35, 5), \quad c \sim Gama(3, 3/950), \quad b \sim Gama(2, 2/5).$$

   El segundo parámetro de la normal es desviación estandard y el segundo parámetro de las gammas es taza (*rate*).

   Usando MH simule de la distribución posterior de a, c y b.

Los datos son estos, $n = 100$:

X = array([28, 17, 14, 51, 16, 59, 16, 54, 52, 16, 31, 31, 54, 26, 19, 13, 59, 48, 54, 23, 50, 59, 55, 37, 61, 53, 56, 31, 34, 15, 41, 14, 13, 13, 32, 46, 17, 52, 54, 25, 61, 15, 53, 39, 33, 52, 65, 35, 65, 26, 54, 16, 47, 14, 42, 47, 48, 25, 15, 46, 31, 50, 42, 23, 17, 47, 32, 65, 45, 28, 12, 22, 30, 36, 33, 16, 39, 50, 13, 23, 50, 34, 19, 46, 43, 56, 52, 42, 48, 55, 37, 21, 45, 64, 53, 16, 62, 16, 25, 62]) # a nos

Y = array([493, 165, 9 , 0, 72, 0, 89, 0, 0, 70, 79, 96, 0, 1127, 548, 4, 0, 0, 0, 1522, 0, 0, 0, 0, 0, 0, 0, 80, 5, 38, 0, 11, 8, 4, 31, 0, 174, 0, 0, 1305, 0, 39, 0, 0, 18, 0, 0, 4, 0, 1102, 0, 94, 0, 13, 0, 0, 0, 1308, 33, 0, 90, 0, 0, 1466, 156, 0, 39, 0, 0, 496, 2, 1368, 190, 0, 12, 76, 0, 0, 5, 1497, 0, 6, 533, 0, 0, 0, 0, 0, 0, 0, 0, 1090, 0, 0, 0, 93, 0, 88, 1275, 0]) # pesos

3. Investiga y describe muy brevemente los softwares OpenBugs, Nimble, JAGS, DRAM, Rtwalk, Mcee Hammer, PyMCMC.

10. Optimización Clásica

1. EM: Considere que se tienen los siguientes datos

$$x = \left(y_{(1)}, y_{(2)}, \ldots, y_{(m)}, z\right)$$

donde $z = (a, \ldots, a)_{n-m}$. $x_i \sim N(\theta, 1)$ si $x_i < a$, $(z_i|a, \theta$ tiene distribución Normal truncada).

$$f(z_i|a,\theta) = \frac{1/\sqrt{2\pi} exp\left\{-1/2(z_i-\theta)^2\right\}}{1-\Phi(a-\theta)} I_{(a,\infty)}(z_i).)$$

El logaritmo de la distribución conjuta es

$$\log\left(f(x,z|\theta)\right) \propto -\frac{1}{2}\sum_{i=1}^{m}(x_i-\theta)^2 - \frac{1}{2}\sum_{i=m+1}^{n}(z_i-\theta)^2$$

y la función de verosimilitud es

$$L(\theta|x) = \frac{1}{(2\pi)^{m/2}} \exp\left\{-\frac{1}{2}\sum_{i=1}^{m}(x_i-\theta)^2\right\}\left[1-\Phi(a-\theta)\right]^{n-m}$$

donde $\Phi(\cdot)$ es la función de distribución de una normal estándar. Implementar el algoritmo EM para obtener el estimador máximo verosímil de θ.

Los datos son:

array([18.221853, 18.417174, 18.720224, 19.067637, 19.128777, 19.402623, 19.507782, 19.580571, 19.632340, 19.930952, 20.116566, 20.142095, 20.445327, 20.461254, 20.646856, 21.000000, 21.000000, 21.000000, 21.000000, 21.000000,])

$n = 20, m = 15, a = 21.$

2. MCMC: Bajo el mismo esquema de datos del ejercicio anterior, poner la distribución a priori para $\theta \sim N(18, 1)$ e implementar un MCMC cuya función objetivo es

$$f(\theta, z_{m+1}, \ldots, z_n|x).$$

Es decir, se agregan parámetros espurios z para salvar el problema de la censura; la misma idea de EM. Utilice solamente kérneles Gibbs. Para simular de las condicionales totales $f(z_i|\theta, x)$ utilice el método de la Transformada Inversa.

3. Método de Newton-Raphson: Bajo el mismo esquema de EM, implementar el método de Newton-Raphson para encontrar a $\hat{\theta}$ que maximice la log-verosimilitud $\log L(x|\theta)$ (MLE).

11. Métodos de remuestreo

Por definir.