Cómputo Estadístico

October 2, 2024

Godinez Bravo Diego

Tarea 3 - Bootstrap

Centro de Investigación en Matemáticas

Maestría en Cómputo Estadístico

0.1 Problema 1

- Muestra que existen $\binom{2n-1}{n}$ distintas muestras de bootstrap de tamaño n.

0.1.1 Solución

Supongamos que tenemos una muestra inicial X de tamaño n. Cada muestra bootstrap se crea tomando muestras aleatorias de la muestra original **con reemplazo**. De manera que cada punto de datos de la muestra inicial puede ser elegido cualquier número de veces. Sea k_i la cantidad de veces que un elemento X_i es elegido. Entonces:

$$\sum_{i=1}^{n} k_i = n$$

Por lo tanto, la muestra $\{X_1, X_2, ..., X_n\}$ puede expresarse en términos de las frecuencias de los elementos X_i como $\{k_1, k_2, ..., k_n\}$.

Considerando lo anterior, el problema ahora involucra saber cuántas formas posibles existen para particionar n elementos, partiendo de que el número de elementos es el mismo para cada muestra bootstrap.

Podemos representar el problema mediante un sistema de puntos y barras, en el cual los puntos simbolizan los elementos y cada barra indica una partición, de la siguiente manera:

[161]:
```
n <- 4
y <- rep(0, 2 * n - 1)
y[sample.int(2 * n - 1, n - 1, replace = FALSE)] <- 1
names(y) <- c("_","|")[y + 1]
print(y)
```

```
| _ _ _ | | _
1 0 0 0 1 1 0
```

Note que las muestras bootstrap $X_j^* = \{X_2, X_2, X_3, X_4\}$ y $X_k^* = \{X_4, X_2, X_3, X_2\}$ son las mismas.

El número de puntos situados entre las barras (o entre los 1's) indica la frecuencia con la que se selecciona el punto X_i. De manera que la cantidad de puntos a la derecha de la última barra representa cuántas veces se elige el último elemento del conjunto original.

A partir del sistema mostrado anteriormente, podemos observar que hay $n-1$ barras. Por lo que el problema se reduce a responder la siguiente pregunta: ¿de cuántas maneras diferentes podemos asignar las posiciones de n puntos para que cada asignación sea única? Por lo tanto, si hay n puntos y $n-1$ barras (i.e., $2n-1$ elementos en total), existen $\binom{2n-1}{n}$ formas distintas de colocar los puntos.

- ¿Cuál es la probabilidad de que una muestra de bootstrap sea identica a la original?

0.1.2 Solución

Para cada posición ($i = 1, 2, ..., n$), podemos seleccionar cualquiera de los n elementos, lo que implica que existen n^n formas de asignar n elementos a n posiciones de manera independiente.

Por otro lado, si consideramos el número de formas distintas de ordenar n elementos sin repetición, obtenemos:

$$n(n-1)(n-2)(n-3)...(2)(1) = n!$$

De este modo, la probabilidad de que una muestra bootstrap sea idéntica a la original es:

$$\frac{1}{n^n} n! = \frac{n!}{n^n}$$

donde $\frac{1}{n^n}$ es la probabilidad que tiene cada muestra bootstrap de ser seleccionada.

- ¿Cuál es la muestra de bootstrap que mas probable de ser seleccionada?

0.1.3 Solución

La muestra bootstrap más probable X^* es aquella que presenta el mayor número de permutaciones, es decir, aquella que ofrece el máximo número de formas en que se pueden organizar sus elementos.

Considerando esto, la muestra bootstrap con el mayor número de permutaciones está conformada por todos los elementos X_i^* diferentes, lo que la hace equivalente a la muestra original $\{X_1, X_2, ..., X_n\}$. Esto se debe a que la presencia de incluso una sola repetición disminuiría el número total de permutaciones.

Profundizando en el punto anterior, cuando un elemento se repite en un conjunto, las diferentes formas de organizar dicho conjunto se ven limitadas. Esto ya que las posiciones ocupadas por elementos idénticos no pueden distinguirse entre sí; e.g., en el conjunto $(2, 2, 3)$, si intercambiamos los dos elementos 2, no estamos creando una nueva permutación, ya que son idénticos.

Por lo tanto, la **muestra bootstrap más probable** es aquella que se compone de cada uno de los elementos de la muestra original, sin repeticiones. Esta configuración maximiza el número de maneras en que los elementos pueden ser organizados.

- ¿Cuál es la cantidad promedio de veces que X_i es seleccionada en una muestra de bootstrap

0.1.4 Solución

Dado que cada muestra bootstrap se genera tomando muestras aleatorias de la muestra original **con reemplazo**, un mismo elemento puede ser seleccionado más de una vez. Lo que significa que la selección de un elemento sea un experimento de Bernoulli, donde un "éxito" ocurre cuando seleccionamos el elemento deseado X_i , y un "fracaso" cuando seleccionamos cualquier otro elemento diferente de X_i.

En una extracción cualquiera de la muestra original de tamaño n, la probabilidad de que un elemento X_i sea seleccionado es $\frac{1}{n}$. Recordando que para una muestra bootstrap de tamaño n, se realizan n extracciones, donde en cada una de ellas la probabilidad de seleccionar X_i permanece constance, ya que cada selección es **independiente y con reemplazo**.

Considerando la naturaleza del muestreo, cada una de las muestras bootstrap sigue una **distribución multinomial**. Por lo tanto:

$$E[X_i] = n * p_i = n(\frac{1}{n}) = 1$$

0.2 Problema 2

Sea $x_1, x_2, ..., x_n$ una muestra aleatoria de una normal $N(\theta, 1)$ y suponga que $\bar{x}$ es un estimador de θ.

Sean $X_1^*, X_2^*, ..., X_n^*$ una muestra bootstrap de $N(\bar{x}, 1)$. Muestre que $\bar{X} - \theta$ y $\bar{X}^* - \bar{x}$ tienen la misma distribución $N(0, 1/n)$.

0.2.1 Solución

Dado que $x_1, x_2, ..., x_n$ siguen una distribución $N(\theta, 1)$, entonces cada variable x_i:

$E[x_i] = \theta \quad \text{y} \quad Var(x_i) = 1$

Por lo que la media muestral $\bar{X}$ definida como:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

es una suma de n variables $x_i \sim N(\theta, 1)$.

Por lo tanto:

$$E[\bar{X}] = E\left[\frac{1}{n} \sum_{i=1}^{n} x_i\right] = \frac{1}{n} \sum_{i=1}^{n} E[x_i] = \frac{1}{n}(n)(\theta) = \theta$$

y

$$\begin{aligned} Var(\bar{X}) = Var\left(\frac{1}{n} \sum_{i=1}^{n} x_i\right) &= E[(\frac{1}{n} \sum_{i=1}^{n} x_i - E[\frac{1}{n} \sum_{i=1}^{n} x_i])^2] \\ &= E[\frac{1}{n^2}(\sum_{i=1}^{n} x_i - E[\sum_{i=1}^{n} x_i])^2] \\ &= \frac{1}{n^2} E[(\sum_{i=1}^{n} x_i - E \sum_{i=1}^{n} x_i)^2] \\ &= \frac{1}{n^2} Var(\sum_{i=1}^{n} x_i) \\ &= \frac{1}{n^2}(n)(1) \\ &= \frac{1}{n} \end{aligned}$$

De manera que, si el estimador $\bar{X} \sim N(\theta, \frac{1}{n})$, entonces:

$$\bar{X} - \theta \sim N(0, \frac{1}{n})$$

Por otro lado, la media muestral $\bar{X}^*$ definida como:

$$\bar{X}^* = \frac{1}{n} \sum_{i=1}^{n} X_i^*$$

es la suma de n variables $X_i^* \sim N(\bar{x}, 1)$.

Entonces:

$$E[\bar{X}^*] = E\left[\frac{1}{n}\sum_{i=1}^{n} X_i^*\right] = \frac{1}{n}\sum_{i=1}^{n} E[X_i^*] = \frac{1}{n}(n)(\bar{x}) = \bar{x}$$

y

$$\begin{aligned} Var(\bar{X}^*) = Var\left(\frac{1}{n}\sum_{i=1}^{n} X_i^*\right) &= E[(\frac{1}{n}\sum_{i=1}^{n} X_i^* - E[\frac{1}{n}\sum_{i=1}^{n} X_i^*])^2] \\ &= E[\frac{1}{n^2}(\sum_{i=1}^{n} X_i^* - E[\sum_{i=1}^{n} X_i^*])^2] \\ &= \frac{1}{n^2}E[(\sum_{i=1}^{n} X_i^* - E[\sum_{i=1}^{n} X_i^*])^2] \\ &= \frac{1}{n^2}Var(\sum_{i=1}^{n} X_i^*) \\ &= \frac{1}{n^2}(n)(1) \\ &= \frac{1}{n} \end{aligned}$$

De lo anterior, el estimador $\bar{X}^* \sim N(\bar{x}, \frac{1}{n})$, por consiguiente:

$$\bar{X}^* - \bar{x} \sim N(0, \frac{1}{n})$$

De esta manera se muestra que $\bar{X} - \theta$ y $\bar{X}^* - \bar{x}$ siguen una distribución normal con parámetros $N(0, 1/n)$.

0.3 Problema 3

Considere el conjunto de datos

$$2, 5, 3, 9.$$

Sean $x_1^*, x_2^*, x_3^*, x_4^*$ una muestra bootstrap de este conjunto de datos.

- Encuentre la probabilidad de que el promedio de la muestra bootstrap sea igual a 2.
- Encuentre la probabilidad de que el promedio de la muestra bootstrap sea igual a 9.
- Encuentre la probabilidad de que el promedio de la muestra bootstrap sea igual a 4.

0.3.1 Solución

Dado el conjunto de datos X, el tamaño de cada muestra bootstrap será de 4 observaciones, las cuales serán seleccionadas de este conjunto.

Sea $\bar{X}^*$ el promedio de la muestra bootstrap:

$$\bar{X}^* = \frac{x_1^* + x_2^* + x_3^* + x_4^*}{4},$$

cada elemento x_i^* tiene una probabilidad asignada de ser seleccionada.

- Nos interesa que el valor de $\bar{X}^*$ sea igual a 2, es decir:

$$\frac{x_1^* + x_2^* + x_3^* + x_4^*}{4} = 2$$

Multiplicando ambos lados por 4:

$$x_1^* + x_2^* + x_3^* + x_4^* = 8$$

Por lo tanto, debemos de considerar cada una de las posibles combinaciones de manera que la suma $x_1^* + x_2^* + x_3^* + x_4^*$ sea igual a 8.

Considerando el conjunto de datos, la única combinación posible que satisface la condición es $x_1^* = x_2^* = x_3^* = x_4^* = 2$.

De esta manera, la probabilidad de que el promedio de la muestra bootstrap sea igual a 2 es:

$$P(\bar{X}^* = 2) = \frac{1}{P(n, r)} = \frac{1}{n^r}$$

donde $P(n, r)$ representa las distintas muestras bootstrap del conjunto de datos con $n = 4$ y $r = 4$.

Por lo tanto:

$$P(\bar{X}^* = 2) = \frac{1}{(4)^4} = \frac{1}{256}$$

- Considerando el caso donde $\bar{X}^*$ es igual a 9, entonces:

$$\frac{x_1^* + x_2^* + x_3^* + x_4^*}{4} = 9$$

Resolviendo obtenemos:

$$x_1^* + x_2^* + x_3^* + x_4^* = 36$$

Esto es, debemos de considerar cada una de las posibles combinaciones de manera que la suma $x_1^* + x_2^* + x_3^* + x_4^*$ sea igual a 36.

Similar al caso anterior, la única combinación que satisface la condición es $x_1^* = x_2^* = x_3^* = x_4^* = 9$. Considerando las distintas muestras bootstrap dado el valor de $n = 4$ y $r = 4$, la probabilidad de que el promedio de la muestra bootstrap sea igual a 9 es:

$$P(\bar{X}^* = 9) = \frac{1}{(4)^4} = \frac{1}{256}$$

- Por último, al considerar $\bar{X}^*$ igual a 4, entonces:

$$\frac{x_1^* + x_2^* + x_3^* + x_4^*}{4} = 4$$

Multiplicando ambos lados por 4:

$$x_1^* + x_2^* + x_3^* + x_4^* = 16$$

Es decir, debemos de considerar cada una de las posibles combinaciones de manera que la suma $x_1^* + x_2^* + x_3^* + x_4^*$ sea igual a 16.

En este caso existen dos posibles combinaciones que satisfacen la condición $x_1^* = x_2^* = x_3^* = x_4^* = 16$.

- Combinación 1: $(x_1^* = 9, x_2^* = 3, x_3^* = 2, x_4^* = 2)$
- Combinación 2: $(x_1^* = 5, x_2^* = 5, x_3^* = 3, x_4^* = 3)$

De las cuales podemos considerar 12 y 6 maneras diferentes de ordenar los elementos, respectivamente. De tal manera que la probabilidad de que el promedio de la muestra bootstrap sea igual a 4 es:

$$P(\bar{X}^* = 4) = \frac{12}{(4)^4} + \frac{6}{(4)^4} = \frac{18}{256}$$

0.4 Problema 4

Maximice las siguientes funciones utilizando el algoritmo de recocido simulado.

a) Función 1

$$f(x, y, \alpha, \beta) = \frac{sin^2[(x+\alpha)^2 + (y+\beta)^2] - 0.5}{[1.0 + 0.001 * ((x+\alpha)^2) + (y+\beta)^2)]^2}$$

$$-100 \leq x \leq 100$$

$$-100 \leq y \leq 100$$

$$-\infty \leq \alpha \leq \infty$$

$$-\infty \leq \beta \leq \infty$$

b) Función 2

$$f(x, y) = 21.5 + xsin(4\pi x) + ysin(20\pi y)$$

$$-3.0 \leq x \leq 12.1$$

$$4.1 \leq y \leq 5.8$$

0.4.1 Solución

Maximizar la función:

$$f(x, y, \alpha, \beta) = \frac{sin^2[(x+\alpha)^2 + (y+\beta)^2] - 0.5}{[1.0 + 0.001 * ((x+\alpha)^2) + (y+\beta)^2)]^2}$$

restricciones:

$$-100 \leq x \leq 100$$

$$-100 \leq y \leq 100$$

$$-\infty \leq \alpha \leq \infty$$

$$-\infty \leq \beta \leq \infty$$

[153]:
```
h <- function(x, y, a, b){
    (sin((x + a)^2 + (y + b)^2)^2 - 0.5)/((1 + 0.001 * (x + a)^2 + (y + b)^2)^2)
} # objective function
```

[154]:
```
# Setting parameters

x0 <- 0.99 # initial x and y points
y0 <- 0.99
a0 <- 0.25 # a and b values
b0 <- 0.30

n <- 5000 # no. of iterations
s <- 0.1 # small perturbation value to generate random nearby points

h0 <- h(x0, y0, a0, b0) # value of h function at initial point (x0, y0, a, b)
z <- matrix(0, n, 6) ; z[1,] <- c(x0, y0, h0, 1, a0, b0) # matrix to store x,␣
 ↪y, h, r, a, and b values
rem <- c() # vector to store rejected iterations
```

[155]:
```
for(i in 2:n){
  ti <- 1/(log(1 + i)) # temperature parameter
  xt <- runif(1, x0 - s, x0 + s) # new point (xt, yt)
  yt <- runif(1, y0 - s, y0 + s)
  if(xt < -100 || xt > 100 || yt < -100 || yt > 100){ # constraint
    rem <- c(rem,i) # if new point goes outside the bounds -100 to 100 it is␣
 ↪skipped
    next
  }
  at <- s + runif(1, -10, 10) # update a and b values
  bt <- s + runif(1, -10, 10)
  ht <- h(xt, yt, at, bt) # h function evaluated at new point (xt, yt, a, b)
  dh <- h0 - ht # diff between previous and current h values
  r <- min(exp(dh/ti), 1) # update r value based on the difference and the␣
 ↪temperature parameter

  if(runif(1) < r){ # accept/update or reject the current point
    x0 <- xt
    y0 <- yt
    h0 <- ht
    a0 <- at
    b0 <- at
    z[i,] <- c(xt, yt, ht, r, at, bt)
  }
  else{
    z[i,] <- c(x0, y0, h0, r, a0, b0)
  }
}
```

[156]:
```
head(z) # matrix preview
```

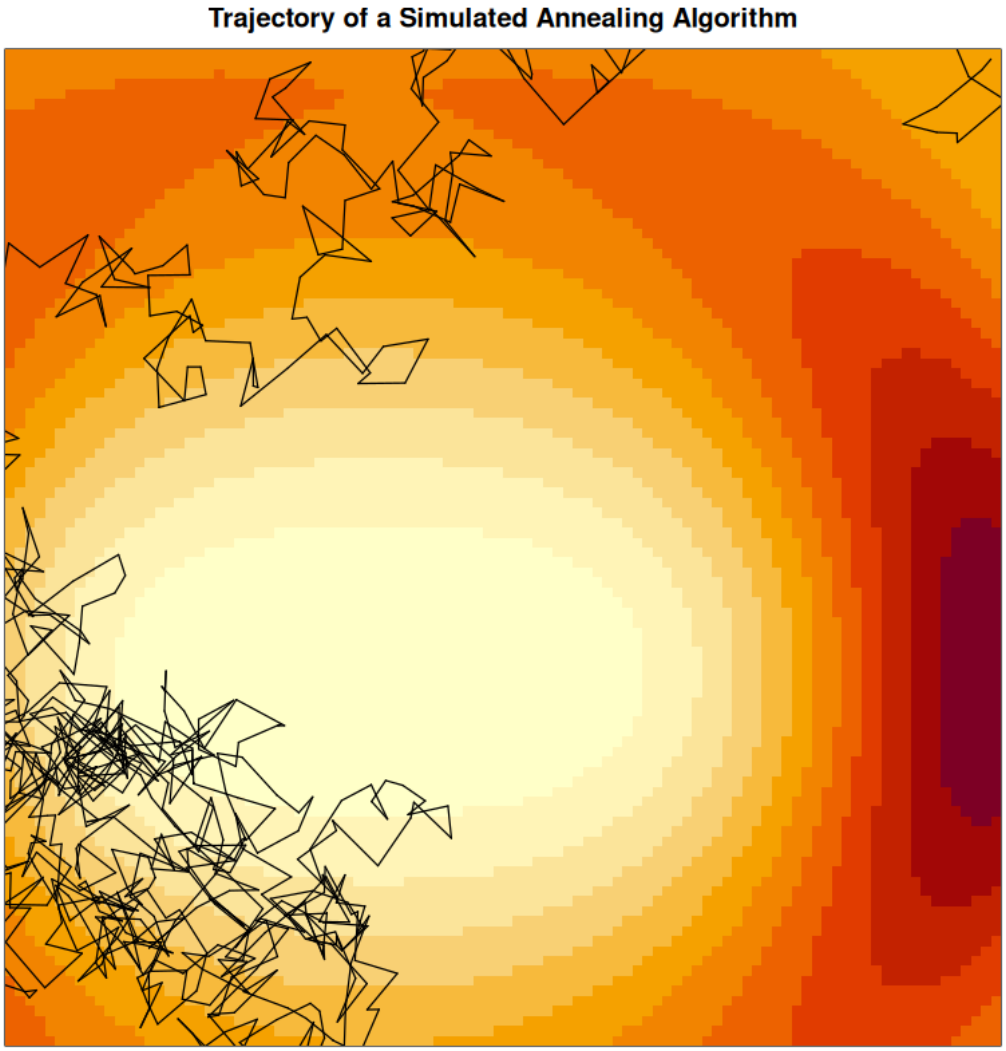
A matrix: 6 × 6 of type dbl

0.9900000	0.9900000	-6.985896e-02	1.0000000	0.250000	0.3000000
0.9713689	0.9678836	-2.513342e-05	0.9261489	7.689145	9.0526648
0.8802466	0.8932900	2.549625e-03	0.9964370	-8.582685	2.4217726
0.8120020	0.8585150	3.226091e-02	0.9533068	1.565773	0.7289883
0.8801000	0.8417009	-4.206830e-05	1.0000000	8.940997	9.2431414
0.9211256	0.8400679	-3.933809e-04	1.0000000	-9.560522	4.8537579

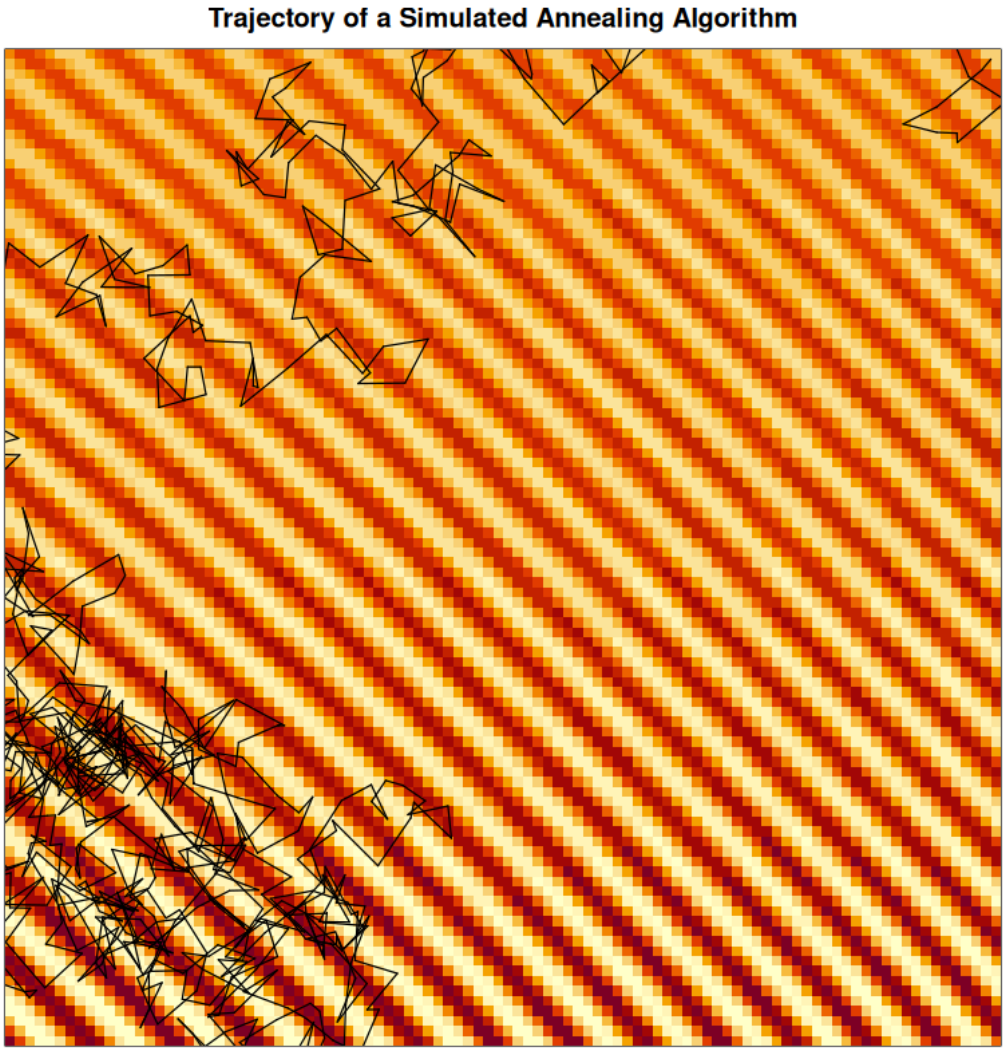
[157]:
```
(z[5000,][3])*-1 # final h value
```

4.34286878292477e-05

[158]:
```
m <- nrow(z)
x <- y <- seq(-1, 1, length = 100)
zi <- outer(x, y, "h", 0.25, 0.25) ; par(mar = c(2, 2, 2, 2))
image(x, y, zi, xaxt = "n", yaxt = "n", ylab = "", xlab = "", cex.main = 1,
main = "Trajectory of a Simulated Annealing Algorithm")
for(i in 1:(m-1)){segments(z[i,1], z[i,2], z[i+1,1], z[i+1,2])}
```



[159]:
```
m <- nrow(z)
x <- y <- seq(-1, 1, length = 100)
zi <- outer(x, y, "h", 10, 10) ; par(mar = c(2, 2, 2, 2))
image(x, y, zi, xaxt = "n", yaxt = "n", ylab = "", xlab = "", cex.main = 1,
main = "Trajectory of a Simulated Annealing Algorithm")
for(i in 1:(m-1)){segments(z[i,1], z[i,2], z[i+1,1], z[i+1,2])}
```



Maximizar la función:

$$f(x, y) = 21.5 + xsin(4\pi x) + ysin(20\pi y)$$

restricciones:

$$-3.0 \leq x \leq 12.1$$

$$4.1 \leq y \leq 5.8$$

[110]:
```r
h <- function(x, y){
    21.5 + (x * sin(4 * pi * x)) + (y * sin(20 * pi * y))
} # objective function
```

[111]:
```r
# Setting parameters

x0 <- runif(1, -3, 12.1) # initial x and y points
y0 <- runif(1, 4.1, 5.8)

n <- 5000 # no. of iterations
s <- 0.1 # small perturbation value to generate random nearby points

h0 <- h(x0, y0) # value of h function at initial point (x0, y0, a, b)
z <- matrix(0, n, 4) ; z[1,] <- c(x0, y0, h0, 1) # matrix to store x, y, h, r,␣
 ↪a, and b values
rem <- c() # vector to store rejected iterations
```

[112]:
```r
for(i in 2:n){
  ti <- 1/(log(1 + i)) # temperature parameter
  xt <- runif(1, x0 - s, x0 + s) # new point (xt, yt)
  yt <- runif(1, y0 - s, y0 + s)
  if( xt < -3 || xt > 12.1 || yt < 4.1 || yt > 5.8 ){ # constraint
    rem <- c(rem,i) # if new point goes outside the bounds it is skipped
    next
  }
  ht <- h(xt, yt) # h function evaluated at new point (xt, yt, a, b)
  dh <- h0 - ht # diff between previous and current h values
  r <- min(exp(dh/ti), 1) # update r value based on the difference and the␣
 ↪temperature parameter

  if(runif(1) < r){ # accept/update or reject the current point
    x0 <- xt
    y0 <- yt
    h0 <- ht
    z[i,] <- c(xt, yt, ht, r)
  }
  else{
    z[i,] <- c(x0, y0, h0, r)
  }
}
```

[113]:
```r
head(z) # matrix preview
```

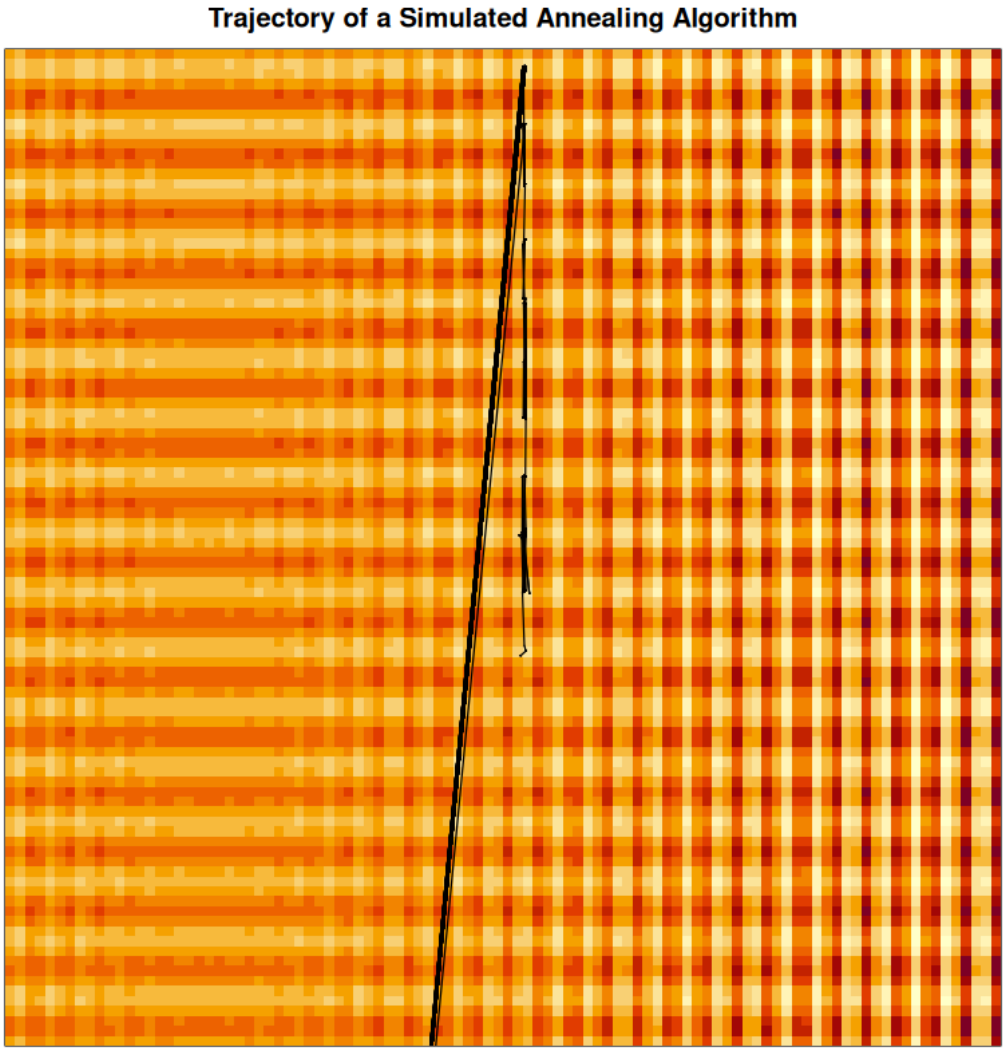
A matrix: 6 × 4 of type dbl

4.819972	4.763671	14.18025	1.000000e+00
4.819972	4.763671	14.18025	2.384370e-02
4.819972	4.763671	14.18025	1.900759e-06
4.819972	4.763671	14.18025	2.851416e-07
4.819972	4.763671	14.18025	5.954546e-06
4.819972	4.763671	14.18025	1.858874e-03

[117]:
```
(z[5000,][3])*-1 # final h value
```

-11.052760677315

[116]:
```
m <- nrow(z)
x <- seq(-3, 12.1, length = 100)
y <- seq(4.1, 5.8, length = 100)
zi <- outer(x, y, "h") ; par(mar = c(2, 2, 2, 2))
image(x, y, zi, xaxt = "n", yaxt = "n", ylab = "", xlab = "", cex.main = 1,
main = "Trajectory of a Simulated Annealing Algorithm")
for(i in 1:(m-1)){segments(z[i,1], z[i,2], z[i+1,1], z[i+1,2])}
```



De los gráficos mostrados con anterioridad, podemos observar la trayectoria del algoritmo, la cual está representada por los segmentos de línea dibujados entre los puntos de la matriz z. Cada punto corresponde a un estado o solución en el espacio de búsqueda, y los segmentos muestran cómo el algoritmo se mueve de un punto al siguiente durante la búsqueda.

Los gráficos muestran la naturaleza intrínseca del algoritmo, que se adentra en el espacio de búsqueda el cual se espera pueda contener múltiples óptimo locales. En este contexto, los valores encontrados de ambas funciones h representan un punto de interés que podría ser un máximo local, aunque su posición en el espacio de búsqueda **no garantiza que sea el óptimo global**.

Es necesario considerar que, aunque los valores obtenidos de h puedan ser significativos, su carácter de máximo local implica que podrían existir mejores soluciones en el espacio de búsqueda.