Curso: Introducción a la Probabilidad y Estadística CM -274
Ejercicios de Laboratorio de R

Lecturas Importantes

1. Una de las mejores referencias del lenguaje R, es el libro de Hadley Wickham `Advanced in R` : `http://adv-r.had.co.nz/` que abarca temas muy especializados como metaprogramación o rendimiento de código.
2. Una guía de aceleración del código de R, escrito por Noam Ross, de optimización de código: `http://www.noamross.net/blog/2013/4/25/faster-talk.html`.
3. Herramientas y técnicas de depuración con funciones en R de Michael Hannon: `http://www.noamross.net/blog/2013/4/18/r-debug-tools.html`.

Notas importantes

- Todos los ejercicicios resueltos serán subidos a los repositorios de github creado por el estudiante. No se evaluará a un estudiante que no tenga creada un cuenta en github. La forma en la que el estudiante debe presentar este laboratorio es:
  - Debes crear una carpeta local llamada `Practica4`.
  - Dentro de la carpeta `Practica1`, se deben crear las carpetas `Ejercicio1, Ejercicio2, Ejercicio3, Ejercicio4, Ejercicio5, ...` donde se deben alojar las soluciones con la extensión `Rmd` para las preguntas que pidan una solución sin código y `.R` del lenguaje R para las que usen código fuente. Cualquier otra extensión, incluyendo letras minúsculas, será considerada 0 puntos en el ejercicio.
- Los archivos de respuesta deben llevar un comentario inicial con tu nombre y código. Por ejemplo.

```
> # Nombre : Cesar Lara Avila 20122345J
> # Respuesta1: El codigo  muestra el uso del SVD en el calculo
> # de la inversa de matrices
>
>
> library(MASS)
>
> a <- matrix(c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0,
+     0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1), 9, 4)
>
> a.svd <- svd(a)
> a.svd$d
```

- Todo acto de copia de código o de respuestas, será motivo de calificación 0. Este trabajo es individual.
- La fecha de presentación de este laboratorio calificado es hasta las 16:00 PM del dia 10 de diciembre.
- Toda situación o presentación distinta a lo que se pide condiciona tu calificación.
- Puedes descargar este archivo en: `https://github.com/C-Lara/Curso-R/blob/master/Lista-ejercicios/Asignacion10/Asignacion10.pdf`.

Preguntas

1. Considera el siguiente modelo genético simple. Una población consta de un número igual de dos sexos: masculino y femenino. En cada generación, los hombres y las mujeres se emparejan al azar y cada pareja produce exactamente dos hijos, uno de sexo masculino y otro de sexo femenino. Estamos interesados en la distribución de la altura de una generación a otra. Supongamos que la altura de ambos niños es solo el promedio de la altura de sus padres; ¿Cómo cambiará la distribución de la altura entre generaciones?.

   Representa las alturas de la generación actual como un data frame con dos variables, `m` y `f`, para los dos sexos. El comando `rnorm` (100, 160, 20) generará un vector de longitud 100 , de acuerdo con la distribución normal con media 160 y la desviación estándar 20. Usamos esta función para generar aleatoriamente la población en la generación 1:

   ```
   > popular <- data.frame(m = rnorm(100, 160, 20), f = rnorm(100, 160, 20))
   ```

   El comando `sample(x, size = length(x))` devolverá una muestra aleatoria de tamaño `size` tomada del vector `x`(sin reemplazo). (También muestreará con reemplazo, si argumento opcional `replace` se establece en TRUE).

   La siguiente función toma el data frame `popular` y permuta aleatoriamente el orden de los hombres. Los hombres y las mujeres se emparejan según las filas y las alturas para la próxima generación se calculan tomando la media de cada fila. La función devuelve un data frame con la misma estructura, dando las alturas de la próxima generación.

   ```
   > prox.gen <- function(popular) {
   +   popular$m <- sample(popular$m)
   +   popular$m <- apply(popular, 1, mean)
   +   popular$f <- popular$m
   +   return(popular)
   + }
   ```

   Usa la función `prox.gen` para generar nueve generaciones, luego usa la función `histogram` para dibujar la distribución de las alturas masculinas en cada generación. El fenómeno que resulta se llama regresión a la media.

2. Watson le dio una cadena S a Sherlock. Tiene N caracteres de longitud y consiste de unos y ceros. Ahora le pregunta: Dado un entero K, tomaré dos índices i y j al azar entre 1 y N. ¿ Cuál es la probabilidad de que S_i y S_j sean 1 y que $|i - j| \leq K$?. Dé la respuesta como una fracción irreducible con formato a/b.

   Se dará una entrada `Entrada_problema_1.txt` para ejecutar su programa, la cual tendrá el siguiente formato:

   - Primera linea: T, el número de casos que se evaluarán,
   - $2T$ lineas: Casos de prueba,
   - Cada caso de prueba tendrá dos líneas, en la primera se darán N y K, mientras que en la segunda se dará la cadena S.

   Restricciones:

   - $1 \leq T \leq 10^5$
   - $1 \leq N \leq 10^5$
   - $1 \leq K \leq N$
   - $1 \leq$ Suma de los N de todos los casos $\leq 10^5$

Ejemplo de entrada:

2

4 3

1011

4 1

1011

Ejemplo de salida:

9/16

5/16

3. Para $X \sim \texttt{Poisson}(\lambda)$, sea $F(x) = \mathbb{P}(X \leq x)$ y $p(x) = \mathbb{P}(X = x)$. Muestra que la función de probabilidad satisface,

$$p(x+1) = \frac{\lambda}{x+1} p(x).$$

Usando este resultado escribe una función para calcular $p(0), p(1), \ldots, p(x)$ y $F(x) = p(0) + p(1) + \cdots p(x)$.

Si X es una variable aleatoria y $F(x)$ es una función que retorna el CDF F de X, entonces tu puedes simular X usando el siguiente programa:

```
> F.rand <- function () {
+   u <- runif(1)
+   x <- 0
+   while (F(x) < u) {
+     x <- x + 1
+     }
+   return(x)
+ }
```

En el caso de la distribución de Poisson, este programa se puede hacer más eficiente calculando F solo una vez, en lugar de volver a calcularlo cada vez que llame a la función $F(x)$. Al usar dos nuevas variables, `p.x` y `F.x` para $p(x)$ y $F(x)$ respectivamente, modifica este programa para que en lugar de usar la función $F(x)$ actualiza `p.x` y `F.x` dentro del ciclo `while`. Tu programa debe tener la forma

```
> F.rand <- function(lambda) {
+   u <- runif(1)
+   x <- 0
+   # p.x <-?
+   # F.x <- ?
+   while (F.x < u) {
+     x <- x + 1
+     # p.x <-?
+     # F.x <-?
+     }
+   return(x)
+ }
```

Debe asegurarse de que al inicio del ciclo `while` siempre tenga `p.x` igual a $p(x)$ y `F.x` igual a $F(x)$.

4. Claudio tiene una tarea de matemática en la escuela en la que debe evaluar muchas expresiones. El no quiere gastar mucho tiempo. Hay M expresiones en total. Mirando las respuestas de Mandy, Claudio notó que las respuestas a todas las preguntas forman una secuencia no decreciente.

Entonces, decidió que todas sus respuestas estarán entre 1 y N. Llena su hoja de respuestas con una secuencia aleatoria no decreciente de tamaño M donde cada elemento está entre 1 y N.

Aquí es donde empiezan los problemas para Claudio. El no quiere elegir valores muy grandes para N, porque entonces tendrá muchas opciones de donde escoger. También, si elige un valor muy pequeño de N, entonces muchas respuestas se pueden repetir y por ello el profesor sospechará que se ha copiado.

Sea $x = \max_{1 \leq i \leq N} f(i)$, donde $f(i)$ es la frecuencia del número i en la secuencia que eligió, Claudio quiere hallar el valor esperado de x. Ayúdalo a resolver el problema.

Se dará una entrada `Entrada_problema_2.txt` para ejecutar tu programa, la cual tendrá el siguiente formato:

- Primera linea: T, el número de casos que se evaluarán,
- T lineas: Casos de prueba,
- Cada caso de prueba tendrá dos enteros, M y N.

Restricciones:

- $1 \leq T \leq 15$
- $1 \leq M \leq 250$
- $1 \leq N \leq 10^9$

Ejemplo de entrada:

4

1 5

3 3

2 9

9 6

Ejemplo de Salida:

1.0000000000

2.2000000000

1.2000000000

4.3146853147

Importante: Imprime la salida de tal forma de que la respuesta dada respecto a la correcta tenga un error de máximo 10^{-3}, de lo contrario se considerará incorrecta.

5. Desarrolla los siguientes problemas

   (a) El código produce un `gráfico de dispersión`

```
> plot.new()
> plot.window(range(pressure$temperature),
+             range(pressure$pressure))
> box()
> axis(1)
> axis(2)
> points(pressure$temperature, pressure$pressure)
> mtext("temperatura", side=1, line=3)
> mtext("presion", side=2, line=3)
> mtext("Presion de vapor \ncomo una funcion de la Temperatura ",
+       side=3, line=1, font=2)
```

- Describe completamente lo que cada llamada a la función en el código anterior hace,eso incluye una explicación del significado de cada argumento en las llamadas a funciones. Tu respuesta debe incluir una explicación de las diferentes regiones y sistemas de coordenadas creado por este código.
- Describe cómo podría producir el mismo gráfico usando `viewports, layouts, units` en el sistema gráfico **grid**. Esta descripción debe incluir una mención de las funciones de `grid` que se requieren y lo que estas funciones hacen.

(b) Usa la función `curve` para mostrar el gráfico de la función densidad **gamma** con parámetros 1 de forma y 1 de proporción. Usa ahora la función `curve` con el atributo `add = TRUE` para mostrar el gráfico de la densidad de la distribución Gamma, con paramétros de forma k y de proporción 1 para $2, 3$, todos en la misma ventana.

(c) Los datos de `iris` corresponden a las medidas en centímetros de las variables `length, width` de los sépalos y `length , width` de los pétalos respectivamente de 50 flores cada una de tres especies de iris. Hay cuatro variables numéricas correspondientes al sépalo y pétalo y un factor `Species`. Muestra una tabla de medias para `Species` (donde las medias se deben calcularse por separado para cada una de las tres `Species`).

6. Un dia, Alei, Chaous y Woreviam estaban viendo el siguiente problema: Tenemos N grupos de personas, los cuales tienen H_i hombres y M_i mujeres cada uno y se desea organizar un baile. Lastimosamente, estos grupos se han desarrollado de tal forma que los hombres y mujeres de un mismo grupo se odian mutuamente. Por otra parte, estos N grupos están ordenados por mérito y todas las mujeres solamente bailarán con hombres cuyo mérito sea mayor estrictamente que el de ellas. Queremos saber cuántas parejas de baile se pueden formar entre todos los grupos módulo $10^9 + 7$.

   Precisamente Chaous el dia anterior estaba practicando Fuerza Bruta, por lo que propone usar un doble `for` para hallar la suma de los valores, pero Woreviam le hace notar que si N es muy grande eso no serviría, por lo que señala que usando un preprocesamiento (1 `for`) y luego una iteración para hallar la respuesta (otro `for`) será lo suficientemente rápido. Sin embargo, Alei odia el usar muchos `for` en todo el programa, siempre desea que este sea compacto, por lo que dice `Hay una solución que solamente necesita 1 for y dos variables` (además del N y el arreglo de pares) para hallar la respuesta.

   Diseñe y explique la solución que propone Alei, haga el código para hallar la respuesta y pruébelo con la entrada `Entrada_problema_3.txt`, la cual tendrá el siguiente formato:

   - Primera linea: N, el número de grupos en orden de mérito,
   - N lineas: Casos de prueba,
   - Cada caso de prueba tendrá dos enteros, H_i y M_i.

   Restricciones:

   - $10^3 \leq N \leq 10^6$,
   - $1 \leq H_i, M_i \leq 10^4$,

   Ejemplo de Entrada:

   4

   1 2

   2 3

   3 4

   4 5

   Ejemplo de Salida:

   55

   Importante: Recuerde dar la respuesta en módulo $10^9 + 7$.

7. En estadística mucho uso es hecho de Φ^{-1} y es común en los libros de texto proporcionar tablas de Φ^{-1}, llamados puntos porcentuales o cuantiles. Un importante uso de los cuantiles es en el cálculo de los intervalos de confianza. Por ejemplo si $Z \sim N(0,1)$, entonces $\mathbb{P}(Z > 1.6449) = 0.05$ y $\mathbb{P}(Z > 1.9600) = 0.025$. Esto es $\Phi^{-1}(0.95) = 1.6449$ y $\Phi^{-1}(0.975) = 1.9600$. Así 1.6449 es el 95 punto porcentual o el 0.95 cuantil de $N(0,1)$. Sea $Z_\alpha = \Phi^{-1}(\alpha)$ entonces Z_α es la única raíz de la función $\Phi(z) - \alpha$. Por lo que si podemos calcular Φ, entonces podemos encontrar Z_α usando un algoritmo de búsqueda de raices.

   Implementa un algoritmo para encontrar los cuantiles Z_α. ¿ Existe algun función construida en R, que pueda hacer el mismo trabajo?. Si la respuesta es afirmativa compare los resultados.

8. En Macondo existen solo 4 denominaciones para moneda: 1, 2, 5 y 10. Dada una cantidad limitada de estas monedas. ¿ De cuántas formas puede usarlas para formar la cantidad N?.

   Se dará una entrada `Entrada_problema_4.txt` para ejecutar su programa, la cual tendrá el siguiente formato:

   - Primera linea: T, el número de casos que se evaluarán,
   - $2T$ lineas: Casos de prueba,
   - Cada caso de prueba tendrá dos líneas, en la primera se dará en valor N mientras que en la segunda se darán A, B, C y D que son la cantidad de monedas de 1, 2, 5 y 10 respectivamente.

   Restricciones:

   - $1 \leq T \leq 150$,
   - $1 \leq A \leq 10000$,
   - $1 \leq B, C, D \leq 1000$.

   Ejemplo de entrada:

   2

   15

   2 3 1 1

   12

   2 2 1 1

   Ejemplo de salida:

   2

   2