Práctica grupal

Teoría de juegos y teoría de colas

Investigación Operativa en la empresa

Teoría de juegos

Para trabajar en Teoría de Juegos en R utilizaremos el archivo de funciones llamado FuncionesJuegos.R. Este archivo necesita algunos paquetes, por tanto, necesitaremos instalarlos primero en nuestro RStudio.

Para instalar dichos paquetes vamos a RStudio, y en la ventana inferior derecha pulsamos sobre "Packages" y posteriormente "Install", y en packages escribiremos ktsolve, BB, Deriv y esperaremos a que se instale todo.

Una vez hayamos hecho esto, descargamos el archivo con las funciones FuncionesJuegos.R, nos situamos en el directorio en el que lo hayamos puesto, realizamos el More / Set As Working Directory, y desde un nuevo script en la misma carpeta escribiremos source("FuncionesJuegos.R"). A partir de aquí podremos utilizar las funciones de dicho archivo.

Para crear un juego debemos crear dos matrices que representan los pagos de cada uno de los dos jugadores. Por otra parte necesitamos indicar los nombres de las acciones (estrategias) de cada uno de los jugadores. A continuación, se muestra un juego en forma normal y su traducción a R

		Jugador nº 2	
		X	*Y*
Jugador nº 1	*A*	3, 2	1, 4
	B	0, 3	2, 1

```
m1 <- matrix(c(3,1,0,2),nrow=2,byrow=TRUE)
m2 <- matrix(c(2,4,3,1),nrow=2,byrow=TRUE)
e1 <- c("A","B")
e2 <- c("X","Y")
j<-list(utilidad_J1=m1,utilidad_J2=m2,estrategias_J1=e1,estrategias_J2=e2)
```

Ahora en la variable **j** (en este caso) tendríamos definido el juego, y lo podríamos mostrar con la función **muestra_juego**

`muestra_juego(j)` nos mostraría lo siguiente

```
                 Jugador 2
               -------------
                  X      Y
               -------------
Jugador 1 A    | 3,2 | 1,4 |
               -------------
          B    | 0,3 | 2,1 |
               -------------
```

Aquí tenemos un ejemplo para otro juego, del que podéis practicar para encontrar su definición en R.

```
                 Jugador 2
               ------------------
                   H        M
               ------------------
Jugador 1 H    | 10,10 | 20, 0 |
               ------------------
          M    |  0,20 | 15,15 |
               ------------------
```

La función `quita_eed(j,em)` nos quita las estrategias estrictamente dominadas, utilizando o sin utilizar estrategias mixtas

En nuestro último ejemplo
`jeed <- quita_eed(j,FALSE)` nos muestra

```
Quito la fila 2 al estar estrictamente dominada por la fila 1
Quito la columna 2 al estar estrictamente dominada por la columna 1
```

Si ahora hacemos `muestra_juego(jeed)` obtenemos el juego con las estrategias dominadas eliminadas.

```
                 Jugador 2
               ---------
                   H
               ---------
Jugador 1 H    | 10,10 |
               ---------
```

Si definimos ahora el siguiente juego en R

		Jugador nº 2		
		X	Y	Z
Jugador nº 1	A	8, 11	6, 14	10, 12
	B	9, 10	9, 3	0, 6

Si probáramos por estrategias estrictamente dominadas no eliminaríamos nada pero si añadimos estrategias mixtas mediante `jeedm <- quita_eed(j,TRUE)` obtenemos

```
Quito la columna 3 al estar dominada por al menos una estrategia mixta:
   >> Ejemplo: (Columna 1)*(0.43)+(Columna 2)*(0.57)
Quito la fila 1 al estar estrictamente dominada por la fila 1
Quito la columna 2 al estar estrictamente dominada por la columna 1
```

`muestra_juego(jeedm)` nos mostraría ahora

```
                  Jugador 2
                 ---------
                     X
                 ---------
Jugador 1 B      |  9,10 |
                 ---------
```

Definimos ahora el siguiente juego en R

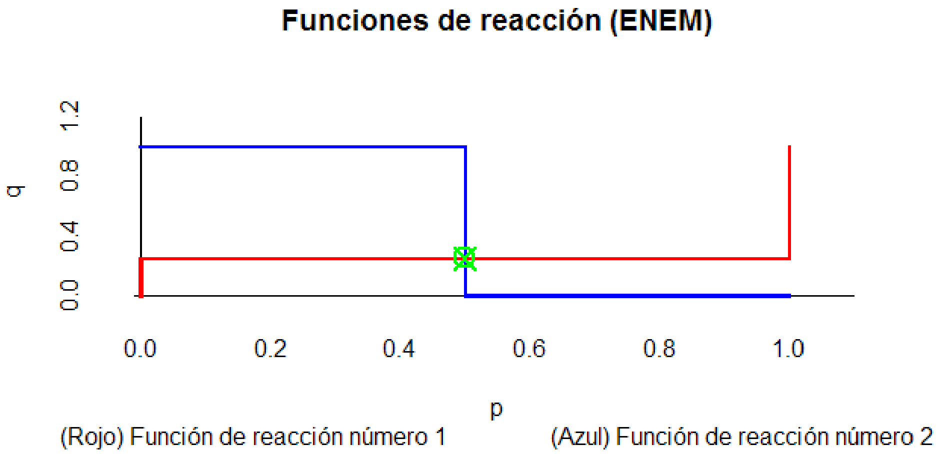
		Jugador nº 2	
		X	Y
Jugador nº 1	A	<u>3</u>, 2	1, <u>4</u>
	B	0, <u>3</u>	<u>2</u>, 1

Si escribimos `enash_ep(j)` nos mostrará los equilibrios de Nash en estrategias puras que tenga el juego, en este caso no tendrá ninguna. El subrayado se muestra a los lados de las utilidades.

```
                         Jugador 2
                 -----------------------------
                           X             Y
                 -----------------------------
Jugador 1 A      | __3__,     2 |     1,__4__ |
                 -----------------------------
          B      |     0,__3__ | __2__,     1 |
                 -----------------------------
```

```
Ningún Equilibrio de Nash en EP encontrado
```

Sin embargo, podemos buscar equilibrios de Nash en estrategias mixtas utilizando `enash_em(j)` que nos mostraría lo siguiente:




```
Equilibrio de Nash en EM en (p=0.5,q=0.25)
   >> ENEM = (0.5·A+0.5·B,0.25·X+0.75·Y)
```

Teoría de colas – Colas especializadas

Gracias a los compañeros del curso 2015/2016:

- Estefanía Jiménez Arroyo
- Samuel López Hernández
- Lucía López-Rey Sánchez-Cano

Disponemos de funciones que nos calculan las medidas de desempeño en colas M/M
Las cargamos con *source('FuncionesColas.R')* y podremos acceder a:

Funciones M/M/1

MM1_medidas(lambda,mu): Medidas de desempeño, devuelve un vector con $Ls, Lq, Ws, Wq, \bar{c}$
MM1_pn(lambda,mu,n): Probabilidad de que haya n clientes en el sistema
MM1_PLsMayorN(lambda,mu,n): Probabilidad de que el número de clientes en el sistema sea mayor que n
MM1_PWsMayorT(lambda,mu,t): Probabilidad de que el tiempo de espera en el sistema sea mayor que t
MM1_PWqMayorT(lambda,mu,t): Probabilidad de que el tiempo de espera la cola sea mayor que t

Funciones M/M/1:DG/N/inf

MM1N_pn(lambda,mu,N,n): Probabilidad de que haya n clientes en el sistema para un sistema de N clientes
MM1N_medidas(lambda,mu,N): Medidas de desempeño, para sistemas con N clientes, devuelve un vector con $Ls, Lq, Ws, Wq, \bar{c}$

Funciones M/M/c

MMc_p0(lambda,mu,c): Probabilidad de que no haya clientes con c servidores
MMc_pn(lambda,mu,c,n): Probabilidad de que haya n clientes con c servidores
MMc_medidas(lambda,mu,c): Medidas de desempeño, para sistemas con c servidores, devuelve un vector con $Ls, Lq, Ws, Wq, \bar{c}$

Funciones M/M/c:DG/N/inf(c<N)

MMcN_p0(lambda,mu,c,N): Probabilidad de que no haya clientes con c servidores para sistemas con N clientes
MMcN_pn(lambda,mu,c,n,N): Probabilidad de que haya n clientes con c servidores para sistemas con N clientes
MMcN_medidas(lambda,mu,c,N): Medidas de desempeño, para sistemas con c servidores para sistemas con N clientes, devuelve un vector con $Ls, Lq, Ws, Wq, \bar{c}$

Ejemplo. *Un lavadero para coches recibe a los mismos siguiendo una distribución de Poisson, con 4 coches por hora, que pueden esperar en el parking de la instalación si el lavadero está ocupado. El tiempo para lavar un coche es exponencial, con una media de 10 minutos. El parking tiene únicamente 4 plazas, y si está lleno, no podrán esperar y se tendrán que ir a otro lado.*

a) *Ayude al propietario del lavadero a conocer el impacto que tiene la limitación de parking y cómo se van los clientes a la competencia.*

```
lambda <- 4
mu <- 6
N <- 5
n <- 5
p5 <- MM1N_pn(lambda,mu,N,n) # Probabilidad del sistema lleno
lambda.perdido <- p5*lambda
lambda.ef <- 4-lambda.perdido
```

b) *La probabilidad de que un automóvil que llegue pase de inmediato al lavado.*

```
p0 <- MM1N_pn(lambda,mu,N,0)
```

c) *El tiempo estimado de espera para que se inicie un servicio.*

```
aux <- MM1N_medidas(lambda,mu,N)
aux[4]
```

d) *Cantidad esperada de lugares de estacionamiento vacíos.*

```
5-aux[1]
```

e) *Probabilidad de que estén ocupados todos las plazas de parking*

p5 (ya calculado antes)

f) *Determine la cantidad de plazas de estacionamiento tal que el porcentaje de automóviles que no puedan encontrar espacio no sea mayor que el 1 por ciento.*

Gracias a R podemos construir un fragmento de código como este

```
for(clientes in 1:15){
    # Probabilidad del sistema lleno
    plleno <- MM1N_pn(lambda,mu,clientes,clientes)
    porc <- round(plleno*100,2)
    print(paste("Con ",clientes-1," plazas probabilidad del ",porc,'%',sep=""))
}
```

Obteniendo:

```
"Con 0 plazas probabilidad del 40%"
"Con 1 plazas probabilidad del 21.05%"
"Con 2 plazas probabilidad del 12.31%"
"Con 3 plazas probabilidad del 7.58%"
"Con 4 plazas probabilidad del 4.81%"
"Con 5 plazas probabilidad del 3.11%"
"Con 6 plazas probabilidad del 2.03%"
"Con 7 plazas probabilidad del 1.34%"
"Con 8 plazas probabilidad del 0.88%"
"Con 9 plazas probabilidad del 0.58%"
"Con 10 plazas probabilidad del 0.39%"
"Con 11 plazas probabilidad del 0.26%"
"Con 12 plazas probabilidad del 0.17%"
"Con 13 plazas probabilidad del 0.11%"
"Con 14 plazas probabilidad del 0.08%"
```

Es decir, que harían falta disponer de **al menos 8 plazas de parking**

Entrega de práctica

Resuelva los siguientes ejercicios utilizando R

1. Dado el siguiente juego en forma normal, se pide:
   a) Indicar qué estrategias sobreviven a la eliminación de estrategias estrictamente dominadas
   b) Para el juego resultante del apartado a) calcular (si hubiera) los equilibrios de Nash en estrategias puras
   c) Para el juego resultante del apartado a) calcular (si hubiera) los equilibrios de Nash en estrategias mixtas

		Jugador #2		
		X	**Y**	**Z**
Jugador #1	**A**	0,0	1,0	2,1
	B	6,3	7,-1	1,0
	C	4,5	5,2	0,3

2. Un pequeño restaurante con una sola trabajadora, Luisa, que vende comida para llevar, tiene un aforo muy pequeño que permite que, además del cliente que está siendo atendido, únicamente 3 clientes hagan cola dentro del local. Por otra parte, tiene el inconveniente de que el Ayuntamiento de su ciudad, incomprensiblemente y alegando cuestiones de movilidad, prohíbe que los peatones hagan cola fuera del local, por lo que los clientes que lleguen una vez el restaurante esté lleno tendrán que comer en otro lugar. Se sabe que los clientes llegan siguiendo una distribución de Poisson con un ritmo de 20 clientes por hora, y que el tiempo para atender a un cliente sigue una distribución exponencial, con una media de un cliente cada 2 minutos, conteste a lo siguiente:

   a) Escriba la notación Kendall que define a este sistema.
   b) Cantidad promedio de clientes en el sistema.
   c) Se sabe que el Ayuntamiento multa diariamente con 50€ a los locales que incumplen la ordenanza de la espera en la calle. Indique a Luis si le convendría (en términos únicamente económicos) saltarse la limitación si cada cliente se gasta una media de 10€ y el local está abierto 8 horas al día.

3. Una tienda de barrio dispone de una única cola, y del tendero Antonio que cobra a los clientes. Tanto los tiempos entre llegadas de clientes como entre la atención de clientes del tendero siguen distribuciones exponenciales. Se sabe que llegan 4 clientes cada hora, y que el tendero tarda 9 minutos en atender a cada cliente. Sabiendo esto se pide:

a) ¿Qué porcentaje del tiempo del trabajo estará Antonio ocioso?
b) Calcula el número promedio de clientes que tendrá Antonio en la tienda y el tiempo que pasa desde que el cliente llega a la tienda hasta que se va.
c) Antonio ha tenido suerte en las últimas Navidades y ha ganado una pedrea, por lo que se ha animado a contratar a un ayudante, el cual trabaja al mismo ritmo que él. ¿Cómo mejorarán las medidas de desempeño con este cambio? (medidas del apartado b)
d) Él había observado, que de media, cada cliente, cada 10 minutos, le robaba productos por un valor de 50 céntimos. El nuevo empleado cobra 20€ la hora. ¿Cuánto ha supuesto realmente el aumento de coste horario para Antonio?

19 de diciembre de 2022