# Sesión 10

# Teoría de colas

La teoría de colas modeliza matemáticamente el proceso de llegadas y salidas de individuos a un servicio, y permite optimizar éste teniendo en cuenta factores como el tiempo de espera, el coste de los servidores, etc. Un modelo de colas viene determinado por:

- La distribución de las llegadas al sistema.
- La distribución de los tiempos de servicio.
- El número de servidores.
- La disciplina de cola.
- La capacidad máxima del sistema.
- El tamaño de la población.

En esta práctica, vamos a simular el funcionamiento de algunos procesos de colas, suponiendo que la disciplina de cola es FIFO (los clientes son atendidos por orden de llegada), que el tamaño de la población es infinito, y que el sistema tiene capacidad infinita. Existen varios parámetros que se pueden utilizar para describir el funcionamiento de un sistema de colas, como son:

- El tiempo medio de permanencia en el sistema ($W$) y en la cola ($W_q$).
- El número medio de clientes en el sistema ($L$) y en la cola ($L_q$).

Si la tasa media de llegadas al sistema es $\lambda$ y el tiempo medio de servicio es $\frac{1}{\mu}$, estos parámetros vienen relacionados por las siguientes fórmulas:

$$L = \lambda W, \quad L_q = \lambda W_q, \quad W = W_q + \frac{1}{\mu}. \tag{10.1}$$

## 10.1. Simulación de un modelo M/M/1

Comenzamos simulando el funcionamiento de un modelo de colas en el que las llegadas se producen de acuerdo con un proceso de Poisson con parámetro $\lambda$, los tiempos de servicio siguen una distribución exponencial con parámetro $\mu$ y disponemos de un único servidor.

Para ello, vamos a simular los tiempos de llegada al sistema, teniendo en cuenta que el tiempo entre llegadas consecutivas seguirá una distribución $exp(\lambda)$; los tiempos de salida dependerán del tiempo de entrada al servidor más el tiempo de servicio, el cual se distribuye mediante una exponencial $exp(\mu)$; y para determinar los tiempos de entrada al servidor mantendremos un registro del número de clientes en la cola.

```
llegadas=rep(0,1000)
x=rexp(1000,1)
llegadas=cumsum(x)
servicio=rexp(1000,2)
salida=rep(0,1000)
tiempo=rep(0,1000)
salida[1]=llegadas[1]+servicio[1]
for (i in 2:1000){
if (llegadas[i]>salida[i-1]) {salida[i]=llegadas[i]+servicio[i]}
    # si no hay clientes en la cola se les atiende inmediatamente
else {salida[i]=salida[i-1]+servicio[i]}
    # en otro caso hay que esperar a que salga el anterior
}
```

Para obtener el tiempo medio de permanencia en el sistema, basta con comparar los tiempos de llegada y salida de cada cliente:

```
tiempo=salida-llegadas # almacenamos los tiempos de permanencia en el sistema
```

Obtenemos por ejemplo

```
> mean(tiempo)
[1] 1.043182
```

A partir de esto, podemos estimar el resto de elementos de interés, usando la ecuación (10.1).

Nótese que en el ejemplo anterior hemos supuesto que la llegada de clientes al sistema es en media de 1 por minuto, mientras que el tiempo medio de servicio es de 0.5 minutos; si tuviésemos un tiempo medio entre llegadas inferior al tiempo medio de servicio el sistema tendería a colapsarse. Matemáticamente, lo que ocurre es que no existe una distribución estacionaria. Así, si ejecutamos el código anterior con $\lambda = 2$ y $\mu = 1$ (luego hay dos llegadas por minuto en media y tardamos dos minutos en atender a cada cliente), se obtendría:

```
> mean(tiempo)
[1] 255.3057
```

, lo que significa un tiempo de espera medio de más de 4 horas para los 1000 clientes que hemos simulado.

## 10.2. Simulación de un modelo M/G/1

De manera análoga se pueden simular otros modelos de colas con diferentes distribuciones de llegada y de servicio. Si por ejemplo las llegadas se producen de acuerdo con una distribución de Poisson $\mathcal{P}(\lambda)$ y los servicios siguen una distribución arbitaria, únicamente tendríamos que

introducir los parámetros de esta distribución en la simulación. Supongamos por ejemplo que el tiempo de servicio es de 1 minuto con probabilidad 0.6 y de 2 minutos con probabilidad 0.4, y que la tasa de llegadas es de 1 llegada cada 3 minutos ($\lambda = \frac{1}{3}$). En ese caso haríamos lo siguiente:

```
llegadas=rep(0,1000)
x=rexp(1000,1/3)
llegadas=cumsum(x)
servicio=sample(c(1,2),1000,c(0.6,0.4),replace=TRUE)
salida=rep(0,1000)
tiempo=rep(0,1000)
salida[1]=llegadas[1]+servicio[1]
for (i in 2:1000){
if (llegadas[i]>salida[i-1]) {salida[i]=llegadas[i]+servicio[i]}
else {salida[i]=salida[i-1]+servicio[i]}
}
```

Obtendríamos así:

```
> tiempo=salida-llegadas
> mean(tiempo)
[1] 2.091685
```

Es decir, la estimación del tiempo medio en el sistema sería $W = 2{,}091685$, de donde deducimos, aplicando la ecuación (10.1), que

$$W_q = 2{,}091685 - 1{,}4 = 0{,}691685, \quad L = \lambda W = \frac{2{,}091685}{3} = 0{,}6972284, \quad L_q = \lambda W_q = 0{,}2305617.$$

Es decir, el tiempo medio en la cola es 0,691685, el número medio de clientes en el sistema es 0,6972284 y el número medio de clientes en la cola es 0,2305617.

## 10.3. Simulación de un modelo M/M/s

A continuación consideramos un modelo más complejo, en el que disponemos de $s$ servidores en lugar de uno sólo.

En ese caso, vamos a considerar para cada servidor un vector en el que vamos almacenando los momentos en los que queda libre, lo que permite entonces atender al siguiente cliente. En el siguiente ejemplo vamos a suponer que las llegadas siguen una distribución de Poisson con parámetro $\lambda = 2$, los tiempos de servicio una exponencial con parámetro $\mu = 1$ (lo que en el caso de un único servidor conduciría al colapso del sistema), y que disponemos de $s = 3$ servidores.

```
llegada=rep(0,10000)
x=rexp(10000,2)
llegada=cumsum(x) # tiempos de llegada al sistema
servicio=rexp(10000,1) # tiempos de servicio
salida=rep(0,10000)
```

```
tiempo=rep(0,10000)
t=rep(0,3) #almacenamos aquí el momento en que quedan libres los servidores
serv=rep(0,10000) # almacenamos aquí el servidor usado por cada cliente
salida[1]=llegada[1]+servicio[1]
serv[1]=1
t[1]=salida[1]

for (i in 2:10000){

if (llegada[i]> t[1]) { #comprobamos iterativamente si algún servidor está libre
salida[i]=llegada[i]+servicio[i] #de ser así entramos en él
serv[i]=1 # y actualizamos el uso del servidor y los tiempos en que queda libre
t[1]=salida[i]}
else if (llegada[i]> t[2]){
salida[i]=llegada[i]+servicio[i]
serv[i]=2
t[2]=salida[i]}
else if (llegada[i]> t[3]){
salida[i]=llegada[i]+servicio[i]
serv[i]=3
t[3]=salida[i]}
else {
z=which.min(t) #miramos qué servidor es el primero en quedar libre
serv[i]=z
salida[i]=t[z]+servicio[i]
       #la entrada al servidor es el último tiempo de salida almacenado
t[z]=salida[i]
}
}
```

Obtenemos así

```
> tiempo=salida-llegada

> mean(tiempo)
[1] 1.411184
```

Se puede observar cómo en el vector de tiempos de salida, al contrario de lo que ocurría en el caso de un sólo servidor, no es creciente, e individuos que llegan más tarde pueden salir antes del sistema.

## 10.4. Simulación de un modelo M/M/1/k

Consideramos por último el caso en el que existe un límite superior a la capacidad del sistema, de manera que a partir si un cliente llega con el sistema lleno su llegada no se registra hasta que hay alguna salida del sistema. Para simularlo, debemos comparar si la llegada del individuo $i$-ésimo es posterior a la salida del individuo $i-k$ (lo que garantiza que hay algún

hueco libre en el sistema); de no ser así debemos cambiar el tiempo registrado de llegada al momento en el que queda un hueco libre.

A continuación simulamos este proceso cuando las llegadas siguen un proceso de Poisson con tasa $\lambda = 1$ y los tiempos de servicio una exponencial con parámetro $\mu = 2$; de manera análoga se haría la simulación para otras distribuciones. Vamos a suponer que $k = 3$, es decir, que el sistema sólo tiene capacidad para tres individuos (uno en servicio y dos en cola). Esto quiere decir que debemos modificar el algoritmo anterior comprobando, a partir del cuarto individuo, que puede entrar al sistema.

```
llegadas=rep(0,1000)
x=rexp(1000,1)
llegadas=cumsum(x)
servicio=rexp(1000,2)
salida=rep(0,1000)
tiempo=rep(0,1000)
salida[1]=llegadas[1]+servicio[1]
for (i in 2:3){
if (llegadas[i]>salida[i-1]) {salida[i]=llegadas[i]+servicio[i]}
else {salida[i]=salida[i-1]+servicio[i]}
}

for (i in 4:1000){
if (llegadas[i]<salida[i-3]) {
llegadas[i]=salida[i-3]
salida[i]=salida[i-1]+servicio[i]}
else if (llegadas[i]>salida[i-1]) {salida[i]=llegadas[i]+servicio[i]}
else {salida[i]=salida[i-1]+servicio[i]}
}
```

Obtenemos así:

```
> tiempo=salida-llegadas

> mean(tiempo)
[1] 0.7328168
```

Observamos cómo el tiempo medio en el sistema baja ligeramente, puesto que la cola ya no puede crecer indefinidamente; así, individuos que antes estaban en cola ahora ven restringida su entrada hasta que se produce un hueco dentro del sistema. Nótese también que las tasas de llegada al sistema ya no son constantes en $\lambda$, por lo que como hemos visto en teoría las fórmulas de Little cambian ligeramente.

## 10.5. Ejercicios

1. Se considera un modelo de colas en el que las llegadas siguen una distribución $\mathcal{P}(0.8)$, mientras que los tiempos de servicio siguen una distribución uniforme en (0.5,1.5). Estima mediante simulación el tiempo medio de permanencia en el sistema y el número medio de clientes en la cola.

2. Supongamos ahora que la capacidad máxima del sistema es de 4 individuos. ¿Cuál sería ahora el número medio de clientes en la cola?

3. Supongamos por último que el sistema tiene capacidad infinita pero que tenemos 2 servidores. ¿Cuál sería ahora el tiempo medio de permanencia en el sistema?