



Universidad Galileo
Msc. Investigación de Operaciones
Simulación y Modelación I
Ing. Carlos Zelada

PROYECTO NO.1:
TIEMPO ENTRE LLEGADAS DE UN BANCO

Jully Jesmin Berganza López
21000183

Guatemala, 18 de septiembre de 2022.

DEFINICIÓN DEL PROBLEMA

En un banco se quiere evaluar el tiempo que existe entre llegadas por día con la siguiente tabla:

	Tiempo entre llegadas (Minutos)						
	0	1	2	3	4	5	6
Lunes	0.1	0.15	0.1	0.35	0.25	0.05	0
Martes	0.1	0.1	0.15	0.2	0.35	0.1	0
Miercoles	0	0.1	0.1	0.2	0.1	0.25	0.25
Jueves	0	0.15	0.2	0.2	0.15	0.15	0.15
Viernes	0.15	0.15	0.2	0.2	0.1	0.1	0.1
Sabado	0.2	0.15	0.1	0.5	0.05	0	0
Domingo	0.35	0.25	0.2	0.1	0.1	0	0

El horario de atención que tiene el banco es de 10:00 hrs – 16:00 hrs.

Hay n agentes trabajando en el banco. Cada uno de los agentes del banco tarda en promedio 8 minutos en atender a un cliente y tiene una desviación de 5 minutos.

RESOLUCIÓN DEL PROBLEMA

Se cargo la lista ante mencionada:

```
vprob <- list(
  c(0.1, 0.15, 0.1, 0.35, 0.25, 0.05, 0),
  c(0.1, 0.1, 0.15, 0.2, 0.35, 0.1, 0),
  c(0, 0.1, 0.1, 0.2, 0.1, 0.25, 0.25),
  c(0, 0.15, 0.2, 0.2, 0.15, 0.15, 0.15),
  c(0.15, 0.15, 0.2, 0.2, 0.1, 0.1, 0.1),
  c(0.2, 0.15, 0.1, 0.5, 0.05, 0, 0),
  c(0.35, 0.25, 0.2, 0.1, 0.1, 0, 0)
)
```



Se crea la función para los servidores:

```
crearServ <- function(n){  
  out <- data.frame(n=0,tInicio=0,tFin=0)  
  for (i in 1:n) {  
    out <- rbind(out,c(i,0,0))  
  }  
  return(out[-1, ])  
}
```

Se crean funciones para los tiempos de llegada, tiempo de atención, tiempo de atención, tiempo libre del servidor.

```
intDia <- 480  
dias <- 7  
finAt <- intDia * dias  
  
simstage <-function(nServers){  
  tacum <- 0  
  serverF <- 1  
  out <- data.frame(dia=0,customer=0,last=0,service=0,arrival=0,begin=0,end=0,espera=0,server=0)  
  i <- 2  
  for (d in 1:dias) {  
    IDia <- TRUE  
    tacum <- 0  
    servers <- crearServ(nServers)  
    cliente <- 1  
    while(tacum <= intDia){  
      llegada <- sample(c(0:6),1,prob = unlist(vprob[d]))  
      if(tacum + llegada > intDia){  
        break;  
      }  
      rServ <- rnorm(1,mean = 8,sd = 5)  
      serv <- ceiling(ifelse(rServ>=1, yes = rServ, no = 0))  
      last <- ifelse(!IDia,yes = llegada,no=0)  
      service <- serv  
      arrival <- ifelse(!IDia,yes = out[i-1,]$arrival+llegada,no=0)  
      begin <- 0  
    }else{  
      serverF <- (servers %>% filter(tFin == min(tFin)))$n[1]  
      begin <- servers[serverF,]$tFin  
    }  
  
    end <- begin + service  
    servers[serverF,]$tInicio <- begin  
    servers[serverF,]$tFin <- end  
  
    espera<- begin - arrival  
    out <- rbind(out,c(d,cliente,last,service,arrival,begin,end,espera,serverF))  
  
    tacum <- arrival  
    i <- i+1  
    cliente <- cliente+1  
    IDia <- FALSE  
  }  
  return (out[-1, ])  
}  
  
simul<- function(semanas, servidores){  
  out <- data.frame(simul=0, tMedCola=0, tEsperaCola=0, tMaxEspera=0, persNoAt=0)
```



```
for (i in 1:semanas) {
  escenario <- simstage(servidores)

  arrEspera <- 0
  for (j in 1:7) {
    n <- max((escenario %>% filter(dia == j))$arrival)
    arrEsperaAux <- 0
    for (k in 1:n) {
      arrEsperaAux <- append(arrEsperaAux, nrow(escenario %>% filter(dia == j) %>% filter(arrival <= k) %>% filter(arrival <= k)))
    }
    arrEspera <- append(arrEspera, arrEsperaAux[-1])
  }
  tMedCola <- mean(arrEspera[-1])
  persNoAt <- ifelse(tMedCola >= 10, yes = tMedCola - 10, no = 0)
  tEsperaCola <- mean(escenario$espera)
  tMaxEspera <- max(escenario$espera)

  out <- rbind(out, c(i, tMedCola, tEsperaCola, tMaxEspera, persNoAt))
}
return(out[-1,])
}
```

1. Haga una simulación para determinar el tamaño de la cola por día cuando hay 1, 2, 3, 4, 5, 6, 7 servidores. (Cola Infinita)

Se crea la variable para el número de servidor a analizar, esto llamando la función creada anteriormente, pero llamando al simulador a analizar. La función head este funciona al permitir visualizar los primeros 6 datos de la tabla, ya que son muchos datos que pueden llevar un tiempo en cargar.

```
sim1 <- simul(10,1)
```

```
head(sim1)
```

```
mean(sim1$tMedCola)
```

Esto se hace para cada servidor. Las respuestas para servidor serían:

- Para un servidor el tiempo promedio es de 68.92
- Para dos servidores el tiempo promedio es de 43.68
- Para tres servidores el tiempo promedio es de 21.11
- Para cuatro servidores el tiempo promedio es de 9.48
- Para cinco servidores el tiempo promedio es de 5.09
- Para seis servidores el tiempo promedio es de 1.69
- Para siete servidores el tiempo promedio es de 0.44



2. Si no se quiere que un cliente esté más de Y minutos en cola, ¿Cuántos agentes debe tener el banco? Considere n servidores (Cola Infinita)

En este caso se saca la media de cada servidor, creada la variable en el inciso anterior, pero llamando la variable de Tiempo de Espera (tEsperaCola)

Y = menos de 2 minutos

```
mean(sim1$tEsperaCola)
```

Al realizar esta función para los 7 servidores, se determina que se necesitan desde 5 servidores para que los clientes nos esperen más de dos minutos.

3. Como una restricción extra el banco no puede tener en cola más de K personas. Cuántas personas no son aceptadas en promedio por día al banco. Considere n servidores (Cola Finita)

```
#Para 5 servidores  
mean(sim5$persNoAt)
```

Cuando atienden a partir de 5 servidores, todas las personas son atendidas. Con estos 5 servidores, no habrá personas en espera.

PUNTOS EXTRA: En promedio un cliente solo está dispuesto a esperar t minutos en la cola y si la cola es mayor de n personas los clientes deciden no hacer cola e irse del banco. Simule con estas restricciones el banco y determine cuántos clientes abandonan la cola y cuantos deciden no entrar al banco.

Suponiendo cola infinita.

t=10 min

Se realizó el mismo código de simulación, solo que en este caso con un tiempo de 10 minutos.

```
#Para 1 servidor
```

```
simu1 <- simu12(10,1)
```

```
head(simu1)
```

```
mean(simu1$mediaAbandona)
```

```
mean(simu1$mediaNoEntra)
```

```
> #Para 1 servidor
```

```
>
```

```
> simu1 <- simu12(10,1)
```

```
> head(simu1)
```

	mediaAbandona	mediaNoEntra
2	0	187.4286
3	0	185.8571
4	0	184.8571
5	0	180.5714
6	0	186.2857
7	0	191.7143

```
>
```

```
> mean(simu1$mediaAbandona)
```

```
[1] 0
```

```
>
```

```
> mean(simu1$mediaNoEntra)
```

```
[1] 185.9286
```

```
> |
```