

Restaurantes con Analisis de Colas

Centro de Investigaciones UFM

11/14/2019

Importación de las librerías necesarias para poder correr el código

```
library(readxl)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
##
```

```
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
```

```
##
```

```
##      date
```

Función para formatear variables de duración

Con esta función podemos formatear las variables con duración en minutos y colocarlo en terminos comprensibles en medida que sea posible.

```
Formatear <- function(variable) {
  nueva <- period(hour=(variable%%60), minute=(variable%60/%1), second=((variable%1*60)%1))
  return(nueva)
}
```

Función Generador de Cola

La función Generador_de_Cola declarada a continuación, es necesario ingresar una fila o un vector de datos conteniendo los datos necesarios para generar una cola, además tambien se le ingresa la cantidad de minutos que se simularan, si no se ingresa una cantidad de minutos, por defecto se pasa una hora a la función si no hay ningún otro dato.

```
Generador_de_Cola <- function(my_row, n=60) {
  # Extraer datos de la fila
  cc <- my_row["CC"]
  restaurante <- my_row["Restaurante"]
  t_dia <- as.factor(my_row["FinDe"])
  miu <- as.numeric(my_row["Miu"])
```

```

lambda <- as.numeric(my_row['Lambda'])

#Inicializacion de parametros
# Este vector consiste de
# Id/Tiempo Entre Llegadas/Llegadas/Servicio/Inicio/Tiempo_Cola/Tiempo_Servicio/Final/Tiempo en Sistema
ultimo_cliente<- c(0,0,0,0,0,0,0,0)
historia_restaurante <- c(ultimo_cliente)

# Crear clientes por cada minuto de la simulación
clientes_por_minuto <- rpois(n, lambda)
llegadas <- c()
for (j in 1:length(clientes_por_minuto)) {
  # Se generan horas de llegadas de forma aleatoria entre cada minuto
  horas <- runif(clientes_por_minuto[j], j-1, j) %>% round(digits=2) %>% sort()
  llegadas <- c(llegadas,horas)
}

# Por cada llegada creada se calcula el resto de los datos para completar la simulación, por c/cliente
for(i in 1:length(llegadas)) {
  # Calculos realizado
  llegada <- llegadas[i]
  e_llegadas <- llegada-ultimo_cliente[3]
  servicio <- rexp(n = 1, rate = miu) %>% round(digits=2)
  inicio <- ifelse(llegada>ultimo_cliente[7],llegada,ultimo_cliente[7])
  cola <- (inicio-llegada) %>% round(2)
  final <- inicio + servicio
  en_sistema <- final-llegada

  # Reasigno las variables que van cambiando y agrego la fila a la historia
  cliente_nuevo <- c(i, e_llegadas, llegada, inicio, cola, servicio, final, en_sistema)
  historia_restaurante <- rbind(historia_restaurante, cliente_nuevo)
  ultimo_cliente <- cliente_nuevo
}

# Reformatear la tabla
historia_restaurante <- historia_restaurante[(2:nrow(historia_restaurante)),]
historia_restaurante <- cbind(cc, restaurante, t_dia, historia_restaurante)

colnames(historia_restaurante) <- c("CC", "Restaurante", "FinDe","id", "Entre_Llegadas",
                                   "Llegada", "Inicio", "Cola", "Servicio", "Final", "En_Sistema")
historia_restaurante <- as_data_frame(historia_restaurante)

# Cambiar las variables de tiempo
historia_restaurante<- historia_restaurante %>%
  mutate_at(c("Entre_Llegadas", "Llegada", "Inicio", "Cola", "Servicio", "Final",
              "En_Sistema"), as.numeric)
historia_restaurante<- historia_restaurante %>%
  mutate_at(c("Entre_Llegadas", "Llegada", "Inicio", "Cola", "Servicio", "Final",
              "En_Sistema"), ~Formatear(.))

return(historia_restaurante)
}

```

Lectura y Manipulación de Datos

Este chunk de código contiene tres ciclos anidados (uno dentro del otro), los cuales son útiles para la lectura de los archivos, no importando la cantidad de estos. Está estructurado para considerar las carpetas como centros comerciales, cada archivo debe estar en formato xlsx, incluir la palabra Datos al inicio y Seguido por el nombre del restaurante, sin caracteres especiales ni espacios.

Además de leer los archivos con este código se crean columnas calculadas con los tiempos útiles y finalmente se crea una línea con los datos útiles (resumidos) por cada una de las hojas en todos los archivos. Cada una de estas filas se adjunta a una tabla llamada `datos_iniciales` que almacenará todos los datos que más adelante serán utilizados. De este código únicamente se recibe una línea por cada una de las hojas de excel que es leída. Además se almacena la tabla completa de datos iniciales.

```
## Ingresar a todos los Centros Comerciales
for (folder in list.files("Data")) {
  ## Ingresar a todos los Restaurantes
  for (archivo in list.files(paste("Data", folder, sep = "/" ))) {
    ## Ingresar a todos los días
    for (sheet in 1:length(excel_sheets(paste("Data", folder, archivo, sep = "/")))) {
      # Solo para poder ver que estoy sacando
      print(paste("Leyendo: Data/", folder, '/', archivo, ': hoja-', sheet, sep = ""))

      # Importar tabla de datos
      temp_table <- read_excel(path = paste("Data", folder, archivo, sep = "/"), sheet = sheet,
                               # Se coloca la esquina superior derecha se colocan los tipos de datos
                               range = cell_limits(c(4, NA), c(NA, 4)),
                               col_types = c("numeric", "date", "date", "date"))

      # Renombramos las columnas
      colnames(temp_table) = c('Ingreso', 'Llegada', 'Inicio', 'Final')

      # Se dejan unicamente las filas completas pues el codigo anterior lee
      # mas de las que existen y se laguean las llegadas
      temp_table <- temp_table %>% filter(complete.cases(temp_table)) %>%
        mutate(Llegada_Anterior= lag(Llegada,1), Final_Anterior= lag(Final,1)) %>%
        # Se estiman los diferentes tiempos que se tienen de la toma de datos
        # que son relevantes para las colas.
        mutate(T_Llegadas=as.duration(Llegada-Llegada_Anterior), T_Cola= as.duration(Inicio-Llegada),
               T_Servicio=as.duration(Final-Inicio), T_Sistema= as.duration(Final-Llegada),

               Libre= as.duration(ifelse(Llegada>Final_Anterior, Final-Llegada,0))) %>%
        # Transformar a minutos en el Tiempo de Cola se estiman 6 segundos para considerar
        # la mala toma de datos
        mutate(T_Llegadas=T_Llegadas/dminutes(1),
               T_Cola= ifelse(T_Cola/dminutes(1) < 0.1, 0, T_Cola/dminutes(1)),
               T_Servicio=T_Servicio/dminutes(1), T_Sistema= T_Sistema/dminutes(1),
               Libre= Libre/dminutes(1), Hizo_Cola=ifelse(T_Cola!=0, 1, 0))

      # Con estos ciclos, calculo cual es el promedio de personas en el sistema y el
      # promedio de personas en Cola
      En_Sistema<-c(0)
      En_Cola<-c(0)
      for (i in 2:nrow(temp_table)) {
        contador_s <- 0
        contador_c<- ifelse(temp_table$T_Cola[i]>0,1,0)
        for (j in 1:(i-1)) {
          contador_s <- contador_s + ifelse(temp_table$Final[j]>temp_table$Llegada[i],1,0)
        }
      }
    }
  }
}
```

```

        contador_c <- contador_c + ifelse(temp_table$Inicio[j]>temp_table$Llegada[i],1,0)
    }
    En_Sistema<- c(En_Sistema, contador_s)
    En_Cola<- c(En_Cola, contador_c)
}
temp_table$En_Sistema<- En_Sistema
temp_table$En_Cola <- En_Cola

# Se obtienen los datos mas relevantes de cada archivo y se resumen en esta tabla
datos_utiles <- temp_table %>%
  summarise(CC= folder, Restaurante= substr(archivo, 6, nchar(archivo)-5),
    Fecha= date(Llegada[1]),Cantidad= n(),
    Intervalo= (Llegada[1]%-Llegada[nrow(temp_table)]),
    Intervalo2= (Llegada[1]%-Final[nrow(temp_table)]),
    # Datos para poder realizar simulacion
    Inv_Lambda= mean(T_Llegadas, na.rm = T), Sd_Inv_Lambda = sd(T_Llegadas, na.rm=T) ,
    Miu= mean(T_Servicio), Sd_Miu= sd(T_Servicio),
    Min_T_Llegadas=min(T_Llegadas, na.rm = T),
    Max_T_Llegadas= max(T_Llegadas, na.rm = T), Max_T_Cola= max(T_Cola, na.rm = T),
    # Promedios de tiempos calculados
    T_Cola= mean(T_Cola), T_Sistema_Prom=mean(T_Sistema),
    T_Sistema_Total= sum(T_Sistema), Libre=sum(Libre, na.rm = T), Hizo_Cola= sum(Hizo_Cola),
    En_Cola= mean(En_Cola), En_Sistema= mean(En_Sistema)) %>%
  mutate(D_Semana=wday(Fecha,week_start = 1)) %>% mutate(FinDe = ifelse(D_Semana>=6,1,0)) %>%
  mutate(Tiempo_Tot= as.duration(Intervalo)/dhours(1)) %>%
  mutate(Lambda= Cantidad/(Tiempo_Tot*60)) %>%
  mutate(Total_Minutos= as.duration(Intervalo2)/dminutes(1))

# Si no existe crear tabla datos_iniciales
if (exists("datos_iniciales")) {
  datos_iniciales <-rbind(datos_iniciales, datos_utiles)
} else {
  datos_iniciales <- datos_utiles
}
}
}
}

```

```

## [1] "Leyendo: Data/La Pradera/DatosBurgerKing.xlsx: hoja-1"
## [1] "Leyendo: Data/La Pradera/DatosBurgerKing.xlsx: hoja-2"
## [1] "Leyendo: Data/La Pradera/DatosBurgerKing.xlsx: hoja-3"
## [1] "Leyendo: Data/La Pradera/DatosBurgerKing.xlsx: hoja-4"
## [1] "Leyendo: Data/La Pradera/DatosGoGreen.xlsx: hoja-1"
## [1] "Leyendo: Data/La Pradera/DatosGoGreen.xlsx: hoja-2"
## [1] "Leyendo: Data/La Pradera/DatosGoGreen.xlsx: hoja-3"
## [1] "Leyendo: Data/La Pradera/DatosPandaExpress.xlsx: hoja-1"
## [1] "Leyendo: Data/La Pradera/DatosPandaExpress.xlsx: hoja-2"
## [1] "Leyendo: Data/La Pradera/DatosPandaExpress.xlsx: hoja-3"
## [1] "Leyendo: Data/La Pradera/DatosPolloCampero.xlsx: hoja-1"
## [1] "Leyendo: Data/La Pradera/DatosPolloCampero.xlsx: hoja-2"
## [1] "Leyendo: Data/La Pradera/DatosPolloCampero.xlsx: hoja-3"
## [1] "Leyendo: Data/La Pradera/DatosSubway.xlsx: hoja-1"
## [1] "Leyendo: Data/La Pradera/DatosSubway.xlsx: hoja-2"

```

```

## [1] "Leyendo: Data/La Pradera/DatosSubway.xlsx: hoja-3"
## [1] "Leyendo: Data/La Pradera/DatosSubway.xlsx: hoja-4"
## [1] "Leyendo: Data/La Pradera/DatosTacoBell.xlsx: hoja-1"
## [1] "Leyendo: Data/La Pradera/DatosTacoBell.xlsx: hoja-2"
## [1] "Leyendo: Data/La Pradera/DatosTacoBell.xlsx: hoja-3"
## [1] "Leyendo: Data/Miraflores/DatosBurgerKing.xlsx: hoja-1"
## [1] "Leyendo: Data/Miraflores/DatosBurgerKing.xlsx: hoja-2"
## [1] "Leyendo: Data/Miraflores/DatosBurgerKing.xlsx: hoja-3"
## [1] "Leyendo: Data/Miraflores/DatosGoGreen.xlsx: hoja-1"
## [1] "Leyendo: Data/Miraflores/DatosGoGreen.xlsx: hoja-2"
## [1] "Leyendo: Data/Miraflores/DatosGoGreen.xlsx: hoja-3"
## [1] "Leyendo: Data/Miraflores/DatosKFC.xlsx: hoja-1"
## [1] "Leyendo: Data/Miraflores/DatosKFC.xlsx: hoja-2"
## [1] "Leyendo: Data/Miraflores/DatosKFC.xlsx: hoja-3"
## [1] "Leyendo: Data/Miraflores/DatosMcDonalds.xlsx: hoja-1"
## [1] "Leyendo: Data/Miraflores/DatosMcDonalds.xlsx: hoja-2"
## [1] "Leyendo: Data/Miraflores/DatosMcDonalds.xlsx: hoja-3"
## [1] "Leyendo: Data/Miraflores/DatosPolloCampero.xlsx: hoja-1"
## [1] "Leyendo: Data/Miraflores/DatosPolloCampero.xlsx: hoja-2"
## [1] "Leyendo: Data/Miraflores/DatosPolloCampero.xlsx: hoja-3"
## [1] "Leyendo: Data/Miraflores/DatosPolloCampero.xlsx: hoja-4"
## [1] "Leyendo: Data/Miraflores/DatosSubway.xlsx: hoja-1"
## [1] "Leyendo: Data/Miraflores/DatosSubway.xlsx: hoja-2"
## [1] "Leyendo: Data/Miraflores/DatosSubway.xlsx: hoja-3"
## [1] "Leyendo: Data/Miraflores/DatosTacoBell.xlsx: hoja-1"
## [1] "Leyendo: Data/Miraflores/DatosTacoBell.xlsx: hoja-2"
## [1] "Leyendo: Data/Miraflores/DatosTacoBell.xlsx: hoja-3"
## [1] "Leyendo: Data/Pradera Concepción/DatosBurgerKing.xlsx: hoja-1"
## [1] "Leyendo: Data/Pradera Concepción/DatosBurgerKing.xlsx: hoja-2"
## [1] "Leyendo: Data/Pradera Concepción/DatosBurgerKing.xlsx: hoja-3"
## [1] "Leyendo: Data/Pradera Concepción/DatosGoGreen.xlsx: hoja-1"
## [1] "Leyendo: Data/Pradera Concepción/DatosGoGreen.xlsx: hoja-2"
## [1] "Leyendo: Data/Pradera Concepción/DatosGoGreen.xlsx: hoja-3"
## [1] "Leyendo: Data/Pradera Concepción/DatosMcDonalds.xlsx: hoja-1"
## [1] "Leyendo: Data/Pradera Concepción/DatosMcDonalds.xlsx: hoja-2"
## [1] "Leyendo: Data/Pradera Concepción/DatosMcDonalds.xlsx: hoja-3"
## [1] "Leyendo: Data/Pradera Concepción/DatosPandaExpress.xlsx: hoja-1"
## [1] "Leyendo: Data/Pradera Concepción/DatosPandaExpress.xlsx: hoja-2"
## [1] "Leyendo: Data/Pradera Concepción/DatosPandaExpress.xlsx: hoja-3"
## [1] "Leyendo: Data/Pradera Concepción/DatosPolloCampero.xlsx: hoja-1"
## [1] "Leyendo: Data/Pradera Concepción/DatosPolloCampero.xlsx: hoja-2"
## [1] "Leyendo: Data/Pradera Concepción/DatosPolloCampero.xlsx: hoja-3"
## [1] "Leyendo: Data/Pradera Concepción/DatosSubway.xlsx: hoja-1"
## [1] "Leyendo: Data/Pradera Concepción/DatosSubway.xlsx: hoja-2"
## [1] "Leyendo: Data/Pradera Concepción/DatosSubway.xlsx: hoja-3"
## [1] "Leyendo: Data/Pradera Concepción/DatosTacoBell.xlsx: hoja-1"
## [1] "Leyendo: Data/Pradera Concepción/DatosTacoBell.xlsx: hoja-2"
## [1] "Leyendo: Data/Pradera Concepción/DatosTacoBell.xlsx: hoja-3"

```

Agrupar Datos

De la tabla de datos iniciales se agrupan los datos y se calculan las distribuciones que hay en cada Food Court, se hacen promedios ponderados de las variables mas importantes que además serán utiles más adelante

para la simulación.

```
datos_a_usar <- datos_iniciales %>%
  # La letra S es utilizada para marcar la diferencia de las variables calculadas con pesos
  mutate(LambdaS=(Cantidad*Lambda), MiuS=(Cantidad*Miu), Var_Miu= (Sd_Miu^2),
         Inv_LambdaS=(Cantidad*Inv_Lambda),
         Var_Inv_Lambda=(Sd_Inv_Lambda^2), En_ColaS= Cantidad*En_Cola,
         En_SistemaS= Cantidad*En_Sistema,
         T_ColaS= Cantidad*T_Cola, T_Sistema_PromS= Cantidad*T_Sistema_Prom) %>%
  group_by(CC, Restaurante, FinDe) %>%
  # Resumen de los datos relevantes por agrupacion establecida
  summarise(Cantidad = sum(Cantidad), Lambda= sum(LambdaS)/sum(Cantidad),
            Miu= 1/(sum(MiuS)/sum(Cantidad)), Sd_Miu=1/(sqrt(sum(Var_Miu))),
            Min_T_Llegadas= min(Min_T_Llegadas), Max_T_Llegadas=max(Max_T_Llegadas),
            Max_T_Cola= max(Max_T_Cola),
            Inv_Lambda=sum(Inv_LambdaS)/sum(Cantidad), Sd_Inv_Lambda =sqrt(sum(Var_Inv_Lambda)),
            Inv_Miu = (sum(MiuS)/sum(Cantidad)),
            T_Cola= sum(T_ColaS)/sum(Cantidad), T_Sistema_Prom =sum(T_Sistema_PromS)/sum(Cantidad),
            Tiempo_Tot=sum(Tiempo_Tot),
            # Características de operación calculadas
            Total_Minutos= sum(Total_Minutos), Libre = sum(Libre),
            P_No_Cola= 1-sum(Hizo_Cola)/sum(Cantidad),
            En_Cola=sum(En_ColaS)/sum(Cantidad), En_Sistema=sum(En_SistemaS)/sum(Cantidad)) %>%
  mutate(Por_Ocioso= Libre/Total_Minutos)

# Para poder calcular la distribucion por cada centro comercial
Totales <- datos_a_usar %>% mutate(por_hora = Cantidad/Tiempo_Tot) %>% group_by(CC, FinDe) %>%
  summarise(Total=sum(por_hora))

datos_a_usar<- as_data_frame(datos_a_usar) %>% mutate(por_hora = Cantidad/Tiempo_Tot) %>% full_join(Totales, by=c("CC", "FinDe"))
mutate(Distribucion=por_hora/Total) %>%
  # Reordenar columnas para mostrarlas ordenadas
  select(CC, Restaurante, FinDe, Lambda, Miu, Distribucion, P_No_Cola, Por_Ocioso, T_Cola, T_Sistema_Prom)

## Warning: `as_data_frame()` is deprecated, use `as_tibble()` (but mind the new semantics).
## This warning is displayed once per session.
```

Escribir Archivo

Esta linea de codigo hace que la tabla se cree y escribe un archivo csv con estos datos. Para la visualizacion de este Markdown se mostró las primeras filas de esta tabla.

```
# Se escribe a un archivo donde se puede ver esta tabla a utilizar
write.csv(datos_a_usar, file = "Resultados/Datos Finales.csv", row.names = FALSE)
datos_a_usar[1:13] %>% head(15)
```

```
## # A tibble: 15 x 13
##   CC      Restaurante FinDe Lambda  Miu Distribucion P_No_Cola Por_Ocioso T_Cola
##   <chr> <chr>         <dbl> <dbl> <dbl>         <dbl>         <dbl>         <dbl>
## 1 La P~ BurgerKing    0  0.772 0.419         0.133         0.351         0.226         0.928
## 2 La P~ BurgerKing    1  0.470 0.393         0.121         0.509         0.318         0.419
## 3 La P~ GoGreen       0  0.472 0.257         0.0788        0.510         0.123         1.12
## 4 La P~ GoGreen       1  0.176 0.208         0.0451        0.812         0.00582        0.245
## 5 La P~ PandaExpre~   0  1.80  0.234         0.317         0.183         0           1.51
## 6 La P~ PandaExpre~   1  0.805 0.237         0.206         0.0536        0.00204        1.68
```

```
## 7 La P~ PolloCampe~ 0 1.11 0.265 0.170 0.252 0.0599 1.35
## 8 La P~ PolloCampe~ 1 1.40 0.382 0.359 0.273 0.167 1.99
## 9 La P~ Subway 0 0.715 0.277 0.125 0.0571 0.000568 1.75
## 10 La P~ Subway 1 0.508 0.222 0.122 0.211 0.0690 1.49
## 11 La P~ TacoBell 0 1.00 0.292 0.176 0.128 0.0630 1.89
## 12 La P~ TacoBell 1 0.573 0.231 0.147 0.344 0.00143 0.910
## 13 Mira~ BurgerKing 0 0.284 0.328 0.0885 0.782 0.250 0.278
## 14 Mira~ BurgerKing 1 0.766 0.533 0.159 0.242 0.312 0.960
## 15 Mira~ GoGreen 0 0.230 0.158 0.0737 0.464 0.00287 1.67
## # ... with 4 more variables: T_Sistema_Prom <dbl>, Max_T_Cola <dbl>,
## # En_Cola <dbl>, En_Sistema <dbl>
```

Generar una lista con todas las simulaciones

Se utilizó la función `apply` con `Margin = 1` significa que por cada fila del data frame corra la función y se utiliza este `n` que es ingresado, en este caso son 12 horas. Además se muestra las primeras filas de la simulación de uno de los restaurantes.

```
Simulaciones <- apply(datos_a_usar, MARGIN = 1, FUN = Generador_de_Cola, n=720)
as_data_frame(Simulaciones[[19]]) %>% head(20)
```

```
## # A tibble: 20 x 11
##   CC      Restaurante FinDe id   Entre_Llegadas Llegada Inicio Cola
##   <chr> <chr>         <chr> <chr> <Period>         <Perio> <Period> <Perio>
## 1 Mira~ McDonalds 1 1 5S 5S 5S 0S
## 2 Mira~ McDonalds 1 2 1M 52S 1M 58S 21M 36S 19M 38S
## 3 Mira~ McDonalds 1 3 2M 48S 4M 46S 22M 38S 17M 52S
## 4 Mira~ McDonalds 1 4 3M 25S 8M 12S 30M 31S 22M 18S
## 5 Mira~ McDonalds 1 5 2M 11S 10M 24S 32M 4S 21M 40S
## 6 Mira~ McDonalds 1 6 7M 37S 18M 1S 34M 21S 16M 19S
## 7 Mira~ McDonalds 1 7 30S 18M 31S 42M 15S 23M 43S
## 8 Mira~ McDonalds 1 8 28S 19M 0S 48M 40S 29M 40S
## 9 Mira~ McDonalds 1 9 2M 51S 21M 51S 49M 4S 27M 13S
## 10 Mira~ McDonalds 1 10 3M 15S 25M 6S 52M 43S 27M 37S
## 11 Mira~ McDonalds 1 11 3M 36S 28M 41S 1H 9M 7S 40M 25S
## 12 Mira~ McDonalds 1 12 27S 29M 9S 1H 10M 25S 41M 16S
## 13 Mira~ McDonalds 1 13 4S 29M 14S 1H 12M 9S 42M 55S
## 14 Mira~ McDonalds 1 14 32S 29M 46S 1H 12M 50S 43M 3S
## 15 Mira~ McDonalds 1 15 4M 17S 34M 4S 1H 12M 54S 38M 49S
## 16 Mira~ McDonalds 1 16 2M 6S 36M 10S 1H 15M 4S 38M 53S
## 17 Mira~ McDonalds 1 17 40S 36M 51S 1H 20M 27S 43M 36S
## 18 Mira~ McDonalds 1 18 12S 37M 3S 1H 25M 29S 48M 25S
## 19 Mira~ McDonalds 1 19 8S 37M 12S 1H 29M 41S 52M 29S
## 20 Mira~ McDonalds 1 20 2M 55S 40M 7S 1H 29M 50S 49M 43S
## # ... with 3 more variables: Servicio <Period>, Final <Period>,
## # En_Sistema <Period>
```