

Brewing Data Cup 2020 - Reto México

Equipo OrozCuspi Data_Mx: Felipe Orozco & Victor Cuspinera

29/11/2020

Problema

El reto para los participantes en México es de logística, específicamente para **optimizar ruta**, con el objetivo de dividir un centro de distribución en seis territorios de reparto a través de un modelo de optimización que busque minimizar la distancia recorrida, con diversas restricciones: balancear los territorios por volumen sin exceder la capacidad de reparto por día, los territorios deben estar en función de la frecuencia de visita al cliente, y territorios balanceados con exactamente el mismo número de clientes.

Data

Inicialmente cargamos librerías y base de datos.

##	Id_Cliente	id_Agencia	Frecuencia	Vol_Entrega	lat	lon
## 1	1	A1	1	2	20.50605	-98.21238
## 2	2	A1	1	17	20.45813	-98.22125
## 3	3	A1	1	2	20.53043	-98.23686
## 4	4	A1	3	15	20.48866	-98.20294
## 5	5	A1	1	2	20.53325	-98.21967
## 6	6	A1	1	19	20.53106	-98.18264

Repetimos los renglones n veces la 'Frecuencia' correspondiente.

```
# # (Esto no funcionó porque se repiten en una misma ruta algunos algunos puntos: "46.1" "168.1" "46
# data2 = data.frame()
# for (i in 1:3){
#   for (j in 1:i){
#     aux <- data %>% filter(Frecuencia == i)
#     aux$repetition <- j
#     data2 <- rbind(data2, aux)
#   }
# }

data2 = data

# data2$id_unico <- paste0(data2$Id_Cliente, ".", data2$repetition)
# data2
# QUITAR LOS SIGUIENTES RENGLONES
# data2$id_unico <- paste0("id_", data2$Id_Cliente)
data2$id_unico <- as.character(data2$Id_Cliente)
data2 <- data2[(nrow(data2)-1000+1):nrow(data2), ]
```

Calculamos el centroide, suponiendo que esa es la ubicación del Centro de Distribución. El centroide se calcula a partir de Longitud/ Latitud

```
# data[,c("lon", "lat")]
#
# ubicaciones.coord<-data2[,c("lon", "lat")]
# centroide <- centroid(ubicaciones.coord)
# centroide
#
# data2 <- rbind(data2, c(9999, "CEDI", 1, 0, centroide[2], centroide[1]))
# data2[nrow(data2),]
```

Distancia

Calculamos distancias entre clientes

```
latitud = data2[, 5]
longitud = data2[, 6]

matriz_dist <- geo_distance( data.frame(longitud, latitud) )

# añadir penalización
matriz_dist[matriz_dist == 0 ] <- 20* max(matriz_dist)

# nombrar renglones y columnas
rownames(matriz_dist) <- data2$id_unico
colnames(matriz_dist) <- data2$id_unico

# matriz_dist[]
```

```
## [[1]]
## [1] "3550" "3562" "2656" "3293" "3528" "3547" "3107" "3558" "3554" "3551"
## [11] "3021" "3076" "3153" "2849" "3285" "3316" "3584" "2994" "3133" "3588"
## [21] "3478" "3520" "3519" "2820" "3553" "3536" "3112" "2835" "3549" "2808"
## [31] "2816" "2853" "2995" "3077" "3057" "3516" "2899" "3056" "3125" "2930"
## [41] "2903" "2837" "3092" "2975" "3148" "2851" "2785" "2902" "2970" "3100"
## [51] "3604" "2863" "3108" "2955" "3093" "2860" "2947" "3130" "2960" "2770"
## [61] "3270" "2745" "3114" "2821" "3054" "2885" "3101" "3063" "2700" "3591"
## [71] "2630" "3228" "2865" "2756" "2727" "2810" "2676" "2739" "3124" "2685"
## [81] "3257" "3574" "2645" "2689" "2634" "2669" "3039" "3608" "3287" "2755"
## [91] "3069" "2971" "2914" "3068" "2864" "3145" "2888" "3531" "2721" "2632"
## [101] "2737" "2846" "2774" "3126" "2764" "2695" "3046" "2872" "3045" "2782"
## [111] "2951" "3062" "2647" "3004" "3115" "3010" "2969" "2804" "3098" "3308"
## [121] "2715" "3142" "3483" "3141" "2693" "2922" "2921" "3236" "2944" "2882"
## [131] "2811" "2949" "2836" "3157" "2855" "2953" "2847" "2890" "2638" "2983"
## [141] "3080" "2848" "2973" "3602" "3088" "2857" "3116" "3583" "3167" "2940"
## [151] "2749" "2912" "2929" "3229" "3006" "2989" "3105" "3164" "2708" "3597"
## [161] "2719" "3437" "3543" "3271" "3083" "3075" "3218" "2667" "3473" "3620"
## [171] "3200" "2653" "3265"
```

```
# # cuadro_result <- data.frame()
# left <- data$Id_Cliente
# # c_1 <-
# left[data$Id_Cliente == ruta1]
```

Referencias

Arga Adyatama. (2020). “Capacitated Vehicle Routing Problem (CVRP) with Nearest Neighbour and Genetic Algorithm”. RPubs. Consultado el 2020-11-27. Link: <https://rpubs.com/Argaadya/cvrp>

Arga Adyatama. (2020). “Machine Learning Implementation in Manufacturing Industry”. GitHub repository. Consultado el 2020-11-27. Link: <https://github.com/Argaadya/manufacturing>