Segmentación de Tarjetas de Crédito

Ing. Erick Chacón, MAIT | MAF Candidate

2023-08-14

IMPORTAR DATASET

```
TC <- read.table("/Users/erickfernandochaconflores/Downloads/tarjetacredito (1).csv",
                 header = TRUE, sep=";", row.names = "ID")
```

TIPOS DE DATOS

```
str(TC)
```

```
## 'data.frame':    8950 obs. of  17 variables:
##  $ balance               : num  40.9 3202.5 2495.1 1666.7 817.7 ...
##  $ frecuencia.balance    : num  0.818 0.909 1 0.636 1 ...
##  $ compras               : num  95.4 0 773.2 1499 16 ...
##  $ deunavez.maxima       : num  0 0 773 1499 16 ...
##  $ importe.plazos        : num  95.4 0 0 0 0 ...
##  $ adelanto.efectivo     : num  0 6443 0 206 0 ...
##  $ frecuencia.compra     : num  0.1667 0 1 0.0833 0.0833 ...
##  $ deunavez.frecuentes   : num  0 0 1 0.0833 0.0833 ...
##  $ plazos.frecuentes     : num  0.0833 0 0 0 0 ...
##  $ efectivo.frecuentes   : num  0 0.25 0 0.0833 0 ...
##  $ transacciones.adelanto: int  0 4 0 1 0 0 0 0 0 0 ...
##  $ transacciones.compras : int  2 0 12 1 1 8 64 12 5 3 ...
##  $ limite.credito        : num  1000 7000 7500 7500 1200 1800 13500 2300 7000 11000 ...
##  $ monto.pagado          : num  202 4103 622 0 678 ...
##  $ pago.minimo           : num  140 1072 627 0 245 ...
##  $ pago.total.prct       : num  0 0.222 0 0 0 ...
##  $ tenencia              : int  12 12 12 12 12 12 12 12 12 12 ...
```

Todas las variables son númericas asi que no hay necesidad de convertir nada.

ESTIMAR EL NÚMERO DE CLUSTERS

Normalizar Variables

Primero hay que normalizar las variables. Donde:

Z score = (Observación - Media de la variable) / Desviación estandar de la variable

La idea es que la media de la variable sea igual a 0 y la desviación estandar la variable igual 1.

```
TC.norm = scale(TC)
```

Matriz de distancia

No vale la pena hacer la matriz de distancia porque hay muchas variables aunque R puede correrla sin problema la interpretación sería pobre.

Determinar si hay Na´s

```
sum(is.na(x = TC.norm))
```

```
## [1] 0
```

```
TC.norm <- na.omit(TC.norm)
```

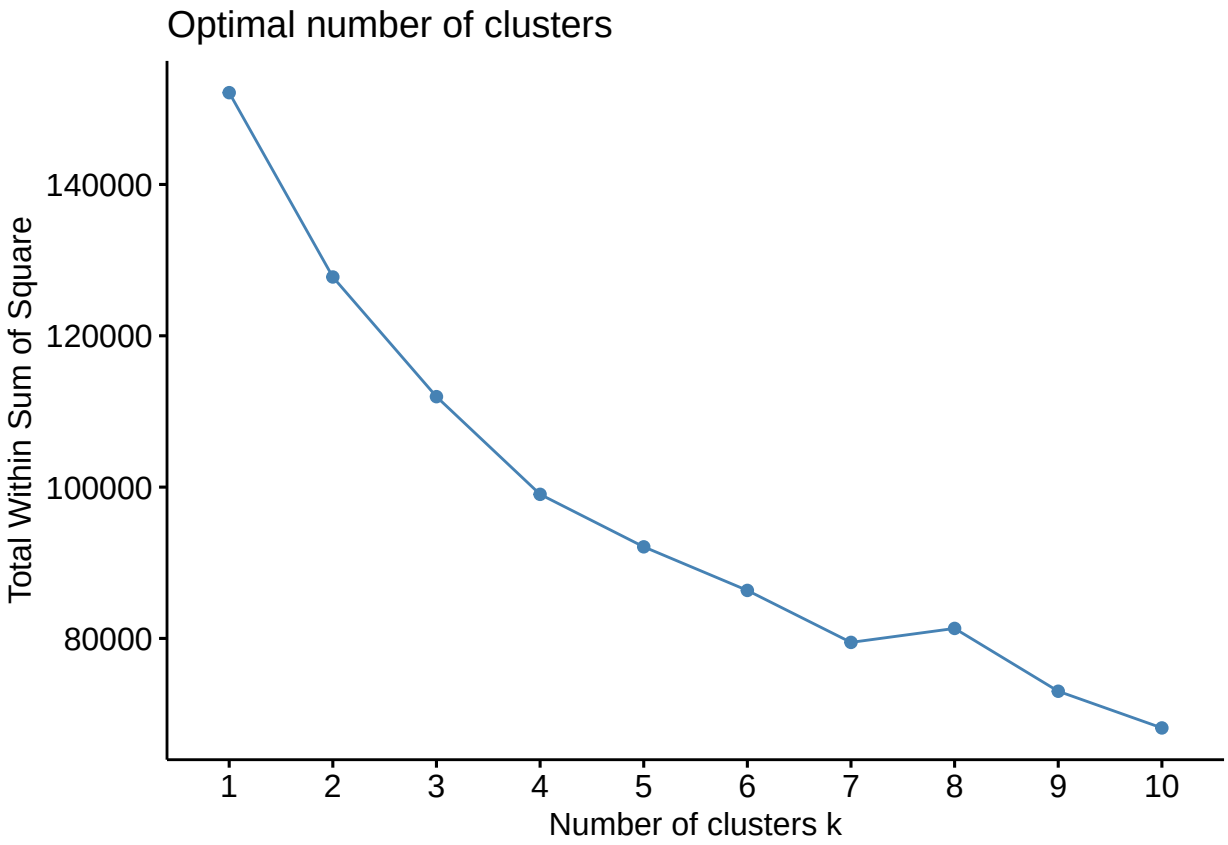
Estimar el número de clusters

```
library(factoextra)
```

```
## Loading required package: ggplot2
```

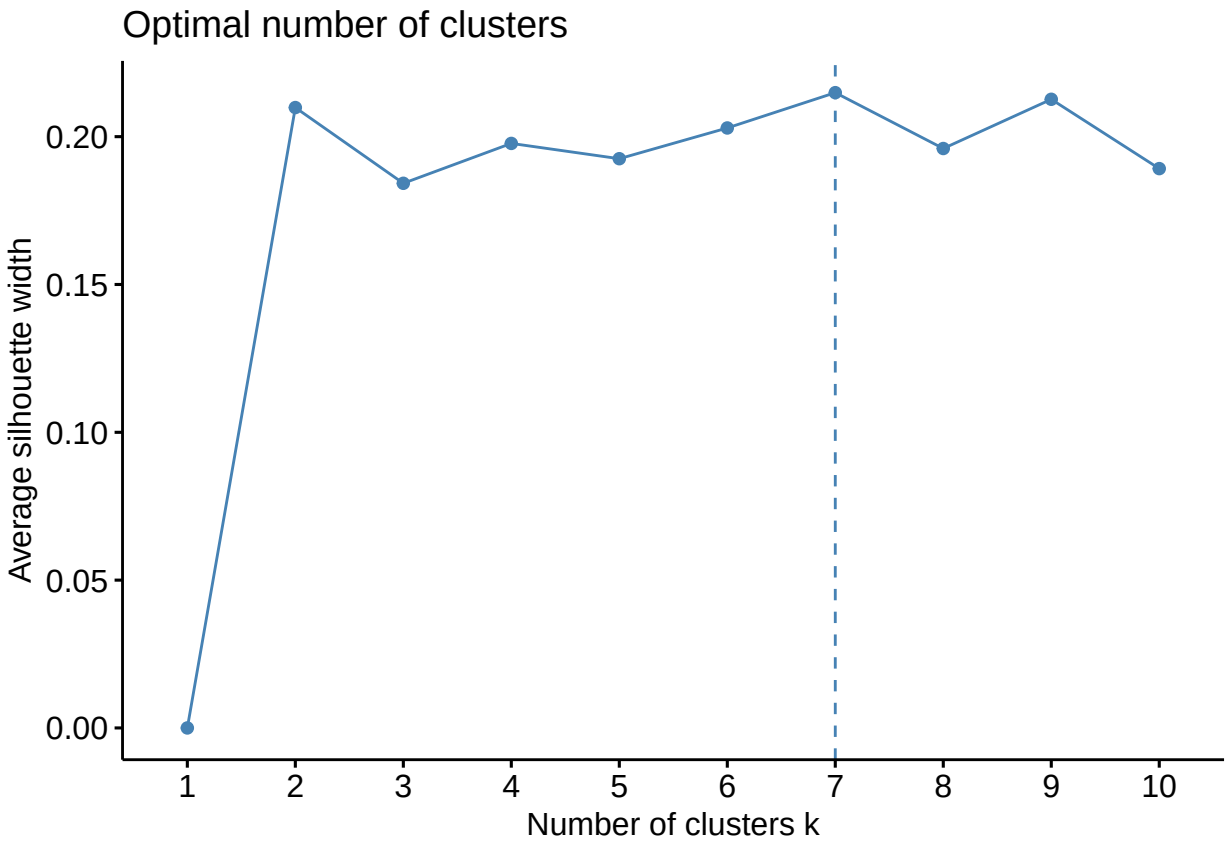
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
fviz_nbclust(TC.norm, kmeans, method ="wss")
```



El 8 es el hombro y el 7 es el codo.

```
fviz_nbclust(TC.norm, kmeans, method ="silhouette")
```

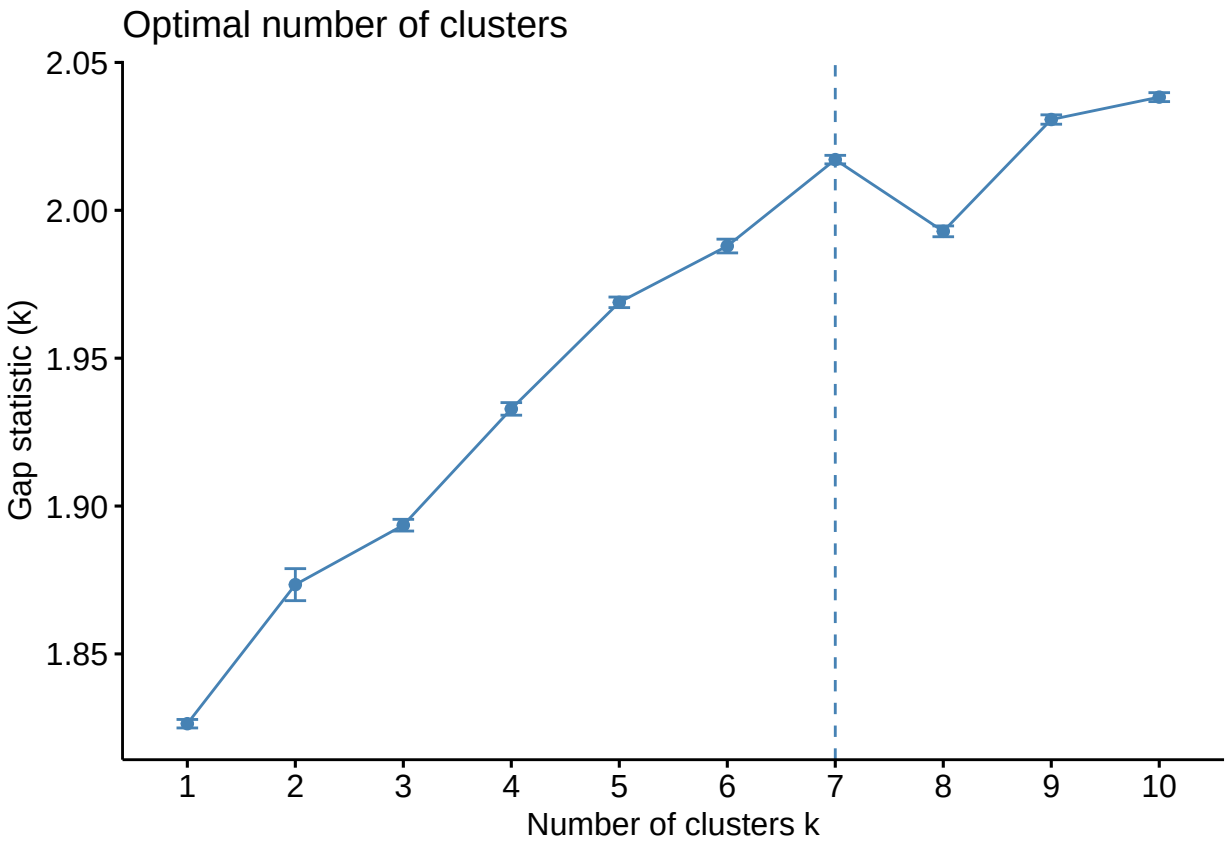


Este método sugiere siete clusters.

```
fviz_nbclust(TC.norm, kmeans, method ="gap_stat")
```

```
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations
```

```
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: Quick-TRANSfer stage steps exceeded maximum (= 447500)

## Warning: Quick-TRANSfer stage steps exceeded maximum (= 447500)

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations
```



Según el método de "gap_stats" nos dice que el optimo es 7 clusters.

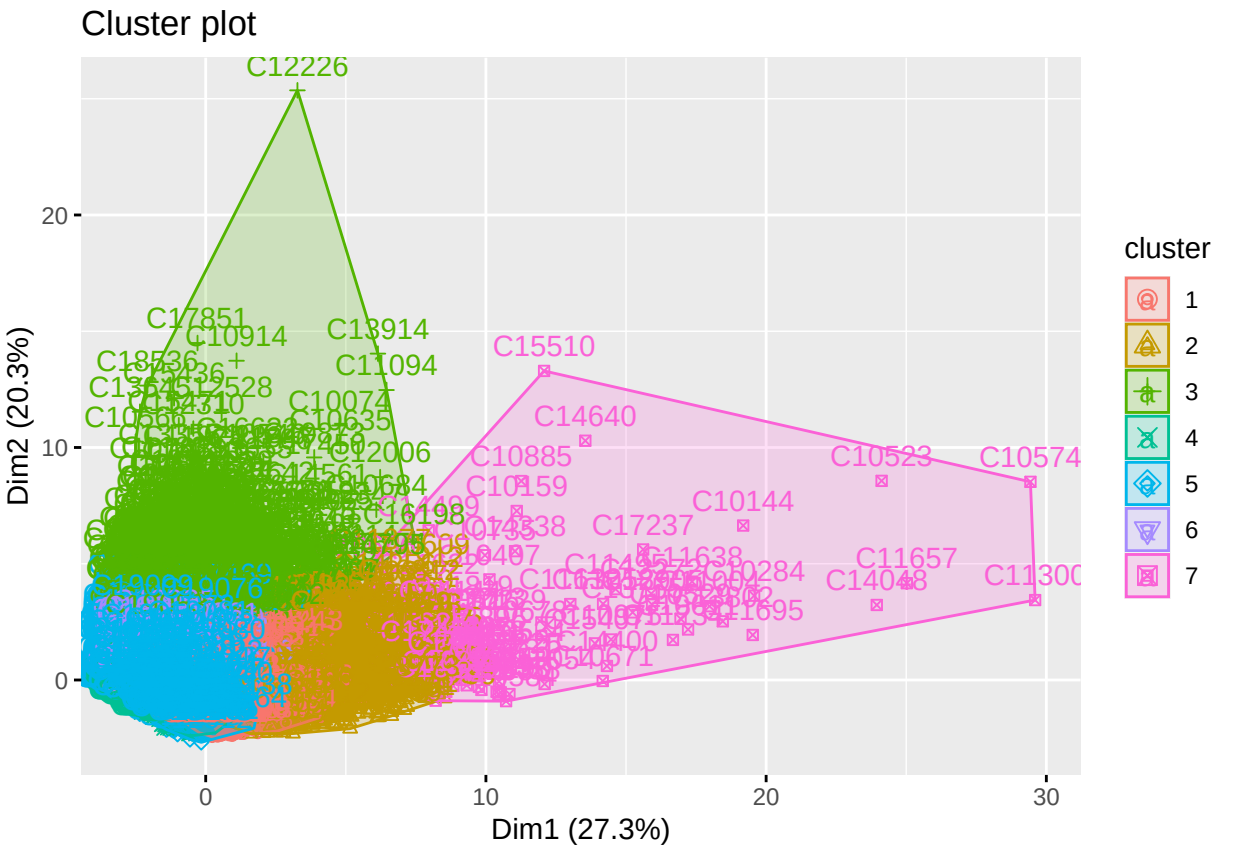
GRÁFICAR LOS CLUSTERS

Primero hay que estimar los clusters. Esto quiere decir que a cada observación se le asignará un cluster.

```
Siete.clusters = kmeans(TC.norm, centers = 7, nstart = 100)
```

Ahora hay que gráficar los clusters.

```
fviz_cluster(Siete.clusters, data=TC.norm)
```



ESTIMAR LAS MEDIAS DE LOS CLUSTERS O CENTROIDES

```
library(dplyr)
```

```
##
## 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
```

```
medias.atributos = TC %>%
  mutate(Cluster = Siete.clusters$cluster) %>%
  group_by(Cluster) %>%
  summarise_all("mean")
```

Visualizemos las medias de los atributos de los clusters. Para poder graficar los atributos de los clusters voy a convertir la base "TC.norm" en un formato de tres columnas utilizando la funcion gather del paquete "tidyr".

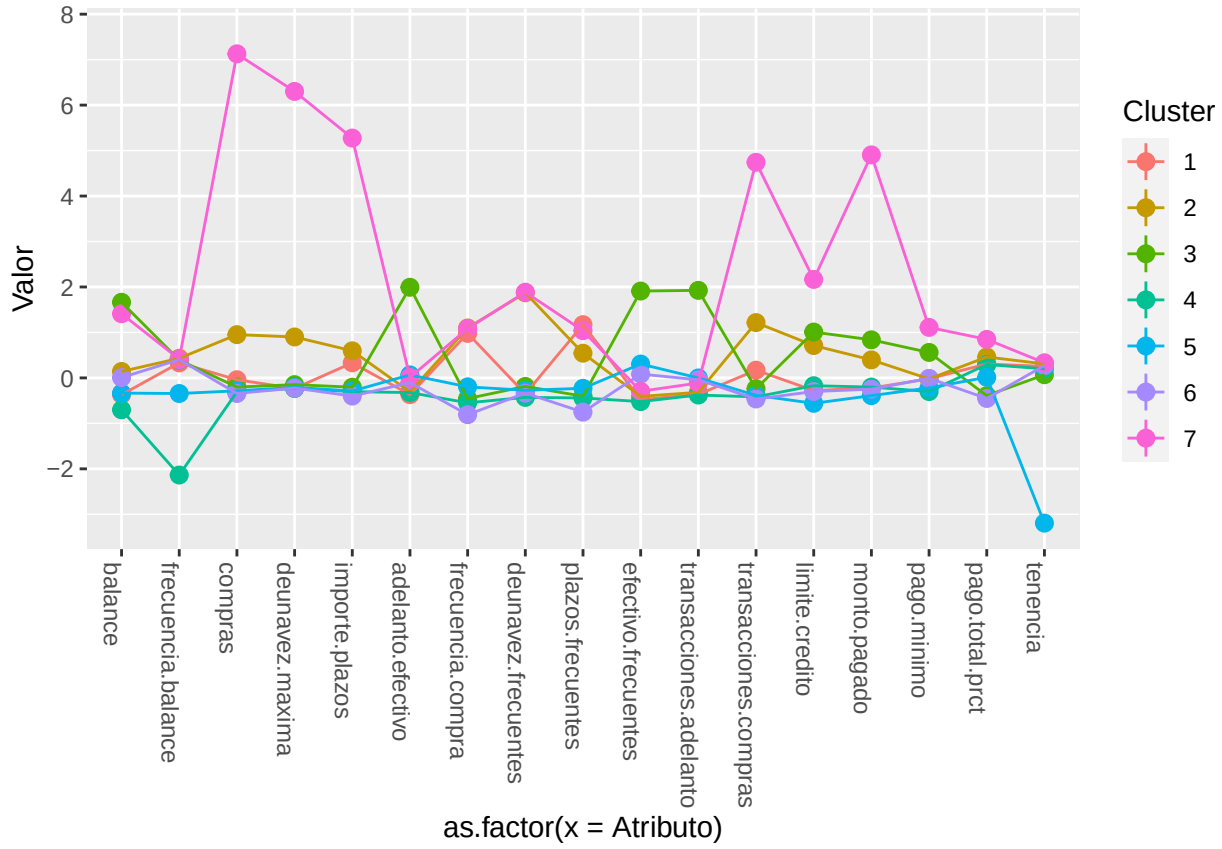
```
library(tidyr)
TC.norm = as.data.frame(TC.norm) #Porque TC.norm se convirtio en una matriz larga.
TC.norm$Cluster = as.factor(Siete.clusters$cluster) #Crear una varible cluster dentro del data set y es
Tres.columnas = gather(TC.norm, Atributo, Valor, balance:tenencia, factor_key = TRUE) #Usar función gat
```

GRÁFICAR LOS CLUSTERS

```
library(ggplot2)
ggplot(Tres.columnas, aes(as.factor(x=Atributo),
                          y= Valor,
                          group = Cluster, colour= Cluster))+
       stat_summary(fun = mean, geom = "pointrange", size =0.5)+
        stat_summary(geom = "line")+
  theme(axis.text.x=element_text(angle = -90, hjust = 0))
```

```
## No summary function supplied, defaulting to `mean_se()`
```

```
## Warning: Removed 119 rows containing missing values (`geom_segment()`).
```



REVISAR CON LAS MEDIAS DE LOS ATRIBUTOS

Revisar con la tabla de las medias de los clusters para ver la similitud entre segmentos.

```
medias.atributos
```

```
## # A tibble: 7 x 18
##   Cluster balance frecuencia.balance compras deunavez.maxima importe.plazos
##     <int>   <dbl>              <dbl>   <dbl>           <dbl>          <dbl>
## 1       1    794.              0.956    917.            207.           711.
## 2       2   1859.              0.979   3036.           2086.           950.
## 3       3   5027.              0.970    572.            346.           226.
## 4       4    103.              0.371    347.            210.           138.
## 5       5    868.              0.796    396.            246.           150.
## 6       6   1580.              0.973    271.            222.            49.7
## 7       7   4501.              0.975  16232.          11051.          5181.
## # i 12 more variables: adelanto.efectivo <dbl>, frecuencia.compra <dbl>,
## #   deunavez.frecuentes <dbl>, plazos.frecuentes <dbl>,
## #   efectivo.frecuentes <dbl>, transacciones.adelanto <dbl>,
## #   transacciones.compras <dbl>, limite.credito <dbl>, monto.pagado <dbl>,
## #   pago.minimo <dbl>, pago.total.prct <dbl>, tenencia <dbl>
```

PERFIL DE CLIENTES