Segmentación para una base de datos simulada

```
suppressMessages(library(tidyverse))
suppressMessages(library(ggplot2))
suppressMessages(library(devtools))
```

```
## Warning: package 'devtools' was built under R version 4.0.5
```

```
## Warning: package 'usethis' was built under R version 4.0.5
```

```
suppressMessages(library(ggbiplot))
suppressMessages(library(corrplot))
```

```
## Warning: package 'corrplot' was built under R version 4.0.5
```

```
suppressMessages(library(caret))
```

Función auxiliar para transformar las variables:

```
asignar <- function(fila){
  return (ifelse(sum(fila)==0, 0, which(fila==max(fila))))
}
```

Función auxiliar para segmentar:

```
segmentar_fila <- function(fila, percentiles){
  ## Segmenta una fila según el número de percentiles dados

  ## Argumentos:
  ## fila: Una fila de datos
  ## Percentiles: Percentiles por los cuales se quiere clasificar los datos

  ## Retorno
  ## fila_segmentada: Fila segmentada por percentiles

  fila_segmentada = cut(fila, quantile(fila, probs = percentiles),
                        include.lowest = TRUE, dig.lab = 999, labels = FALSE)

  return(as.numeric(fila_segmentada))
}
```

Lectura de la base de datos:

```r
dtb <- read.csv("base.csv", sep = ";")
```

Canales preferidos:

```r
dtb <- dtb %>% mutate(en_vm_preferido=apply(dtb[2:12], 1, asignar)) %>%
  mutate(en_tx_preferido=apply(dtb[13:23], 1, asignar)) %>%
  mutate(sal_vm_preferido=apply(dtb[24:27], 1, asignar)) %>%
  mutate(sal_tx_preferido=apply(dtb[28:31], 1, asignar))
```

Ponderación canales salientes:

```r
# dtb <- dtb %>% mutate(en_vm_ponderado=segmentar_fila(rowSums(dtb[, 2:12])/11, percentiles = seq(0, 1
#   mutate(en_tx_ponderado = segmentar_fila(rowSums(dtb[, 13:23])/11, percentiles = seq(0, 1 ,by=0.2)))
#   mutate(sal_vm_ponderado = segmentar_fila(rowSums(dtb[, 24:27])/4, percentiles = seq(0, 1 ,by=0.2)))
#   mutate(sal_tx_ponderado = segmentar_fila(rowSums(dtb[, 28:31])/4, percentiles = seq(0, 1 ,by=0.2)))
```

```r
dtb <- dtb %>% mutate(en_vm_ponderado=rowSums(dtb[, 2:12])/11) %>%
  mutate(en_tx_ponderado = rowSums(dtb[, 13:23])/11) %>%
  mutate(sal_vm_ponderado = rowSums(dtb[, 24:27])/4) %>%
  mutate(sal_tx_ponderado = rowSums(dtb[, 28:31])/4)
```

Pagos y recaudos PN y PJ

```r
dtb <- dtb %>% mutate(persona_pagos = as.numeric(dtb$pagos_pj > dtb$pagos_pn)) %>%
  mutate(persona_recaudos = as.numeric(dtb$recaudos_pj > dtb$recaudos_pn))
```

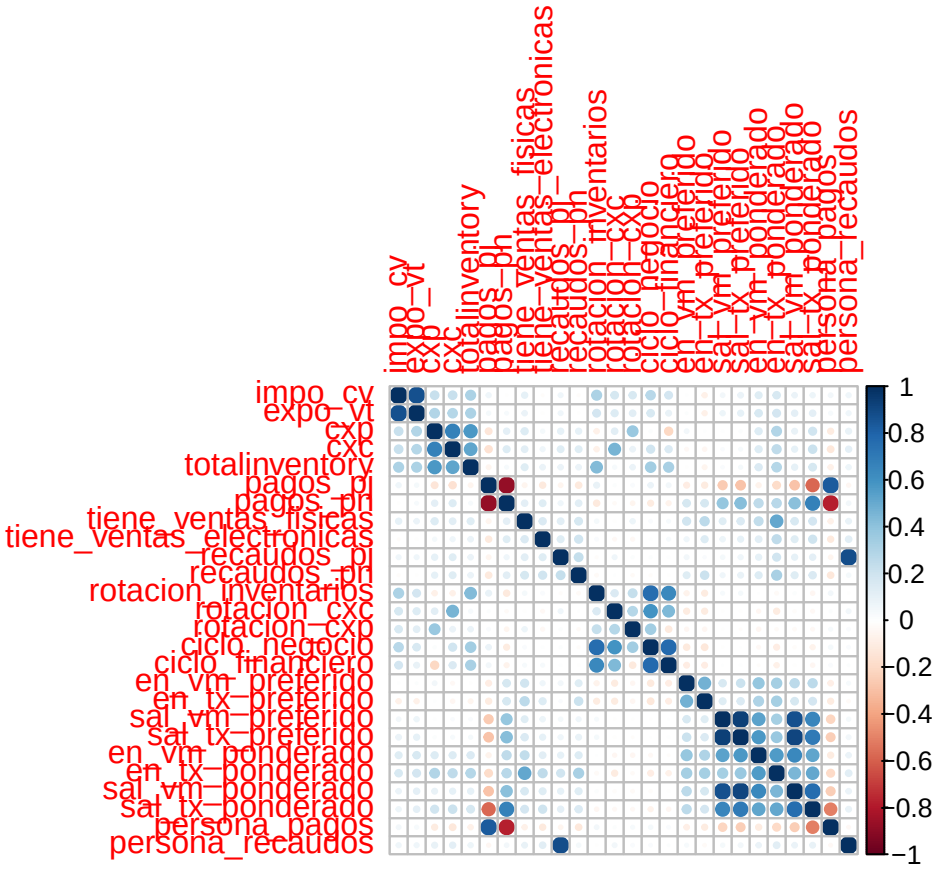
Transformación logarítmica:

```r
dtb <- dtb[, c(1,32:length(dtb))]
indices <- (length(dtb)-2):(length(dtb)-5)
dtb[, indices] <- log(dtb[, indices]+ 1)
```

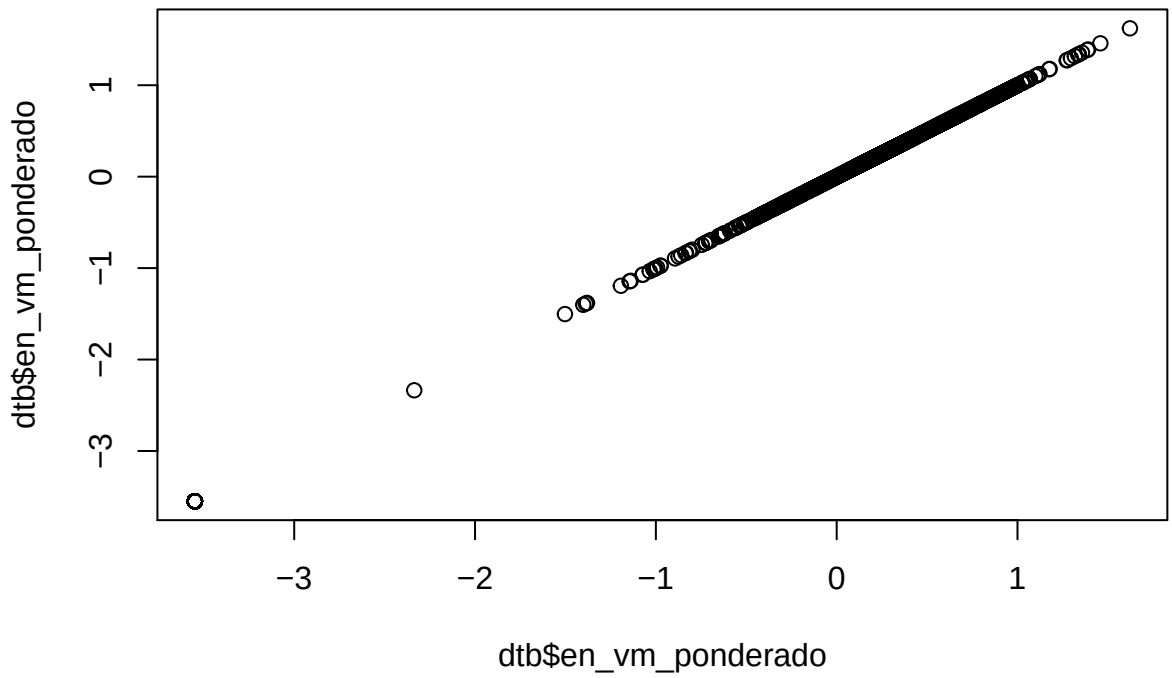
Reescalado de la base de datos y recoger "características":

```r
dtb[, 2:length(dtb)] <- scale(dtb[, 2:length(dtb)], center = TRUE, scale = TRUE)
```

```r
corrplot(cor(dtb[,2:length(dtb)]))
```



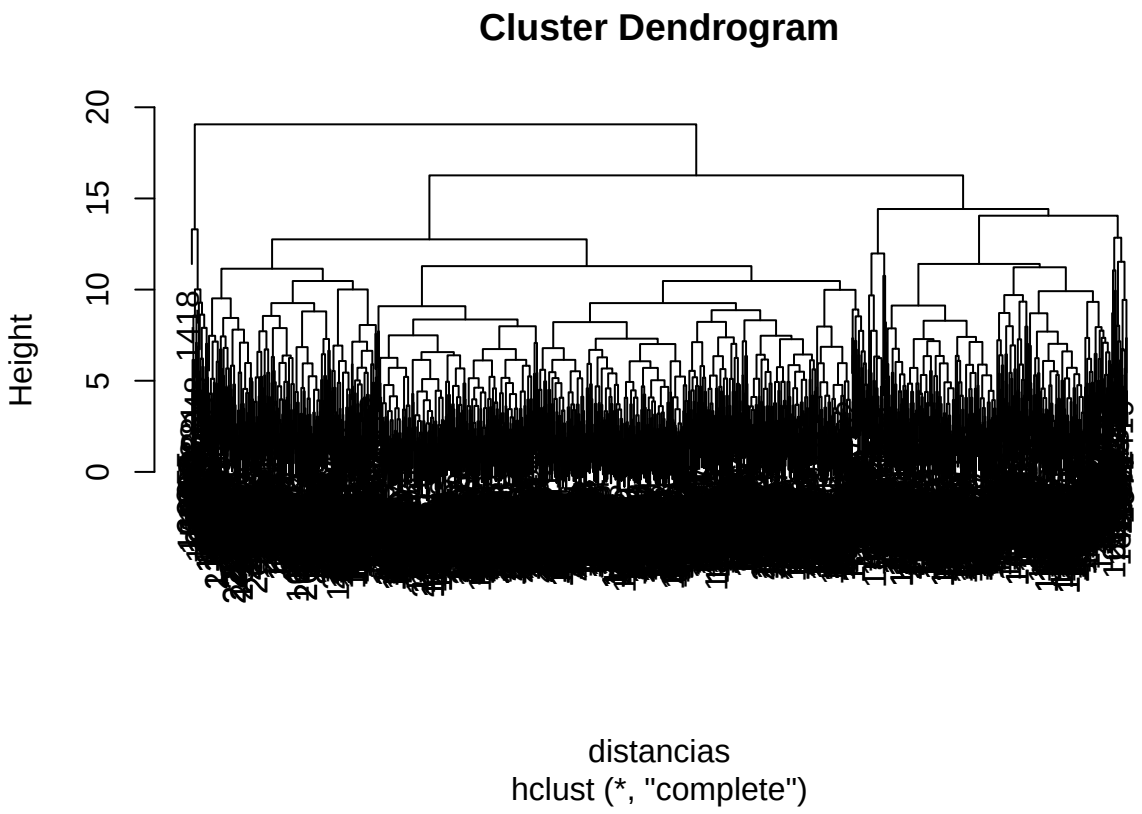

```
plot(dtb$en_vm_ponderado, dtb$en_vm_ponderado)
```



Intento de clustering jerarquico

```
distancias <- dist(dtb[, 2:length(dtb)])
clusters <- hclust(distancias)
```

```
plot(clusters)
```




```
cortes <- cutree(clusters, k = 4)
table(cortes)
```

```
## cortes
##    1    2    3    4
## 1574  574   44   41
```

Análisis de componentes principales de los clusters

```
pca_dtb <- princomp(dtb[, 2:length(dtb)])
summary(pca_dtb)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3     Comp.4     Comp.5
## Standard deviation     2.3666686 1.9174135 1.51914417 1.47924757 1.35625710
## Proportion of Variance 0.2155242 0.1414662 0.08880127 0.08419822 0.07077913
## Cumulative Proportion  0.2155242 0.3569904 0.44579170 0.52998992 0.60076905
##                            Comp.6     Comp.7     Comp.8     Comp.9    Comp.10
## Standard deviation     1.27406881 1.15078705 1.07802107 1.02020017 0.97497009
## Proportion of Variance 0.06246072 0.05095785 0.04471731 0.04004903 0.03657664
## Cumulative Proportion  0.66322977 0.71418762 0.75890493 0.79895396 0.83553059
##                           Comp.11    Comp.12    Comp.13   Comp.14    Comp.15
## Standard deviation     0.90241137 0.89049238 0.71470402 0.6735395 0.55801007
```

```
## Proportion of Variance 0.03133504 0.03051277 0.01965503 0.0174561 0.01198134
## Cumulative Proportion  0.86686564 0.89737840 0.91703343 0.9344895 0.94647087
##                           Comp.16     Comp.17     Comp.18     Comp.19
## Standard deviation     0.48741896 0.435839278 0.398321613 0.382612361
## Proportion of Variance 0.00914168 0.007309269 0.006105046 0.005632993
## Cumulative Proportion  0.95561255 0.962921818 0.969026864 0.974659856
##                           Comp.20    Comp.21     Comp.22     Comp.23
## Standard deviation     0.378395493 0.33415839 0.316308288 0.302099393
## Proportion of Variance 0.005509512 0.00429661 0.003849837 0.003511728
## Cumulative Proportion  0.980169368 0.98446598 0.988315815 0.991827543
##                           Comp.24     Comp.25     Comp.26
## Standard deviation     0.293254007 0.278149325 0.221413132
## Proportion of Variance 0.003309094 0.002976989 0.001886375
## Cumulative Proportion  0.995136637 0.998113625 1.000000000
```

```
ggbiplot(pca_dtb, group=factor(cortes))
```