Clasificación de los FRE mediante análisis por clusterización

Misión PRI 1901

true

08-10–2021

Abstract

A continuación, se presenta un análisis mediante agrupación por clústers de los FRE de acuerdo a variables de desempeño y complejidad de los mismos

```r
require(readxl)
require(skimr)
require(PerformanceAnalytics)
require(factoextra)
require(patchwork)
require(tidyverse); theme_set(theme_bw())
source(file.path('src', 'models', '900_funcionesAlmacenamientoGrafico.R'), encoding = 'UTF-8')
fig_path <- file.path('figures', '011_clasificacion')
```

```r
data <-
  read_excel(file.path(
    'data',
    'raw',
    'ClasificacionFRE',
    'variablesClasificacionFRE.xlsx'
  ), na = '-') %>%
  mutate(Departamento...2 = str_to_title(Departamento...2))
```

```
## New names:
## * Departamento -> Departamento...1
## * Departamento -> Departamento...2
```

```r
data %>%
  select(!contains('Departamento')) %>%
  chart.Correlation(., histogram = TRUE, pch = 19)
```

```r
if (knitr::is_html_output()) {
  skimr::skim(data)
}
```

1. Análisis por PCA

```r
pca1 <- data %>%
  drop_na() %>%
  column_to_rownames('Departamento...2') %>%
  select(!contains('Departamento')) %>%
  prcomp(., scale = TRUE, center = TRUE)
```

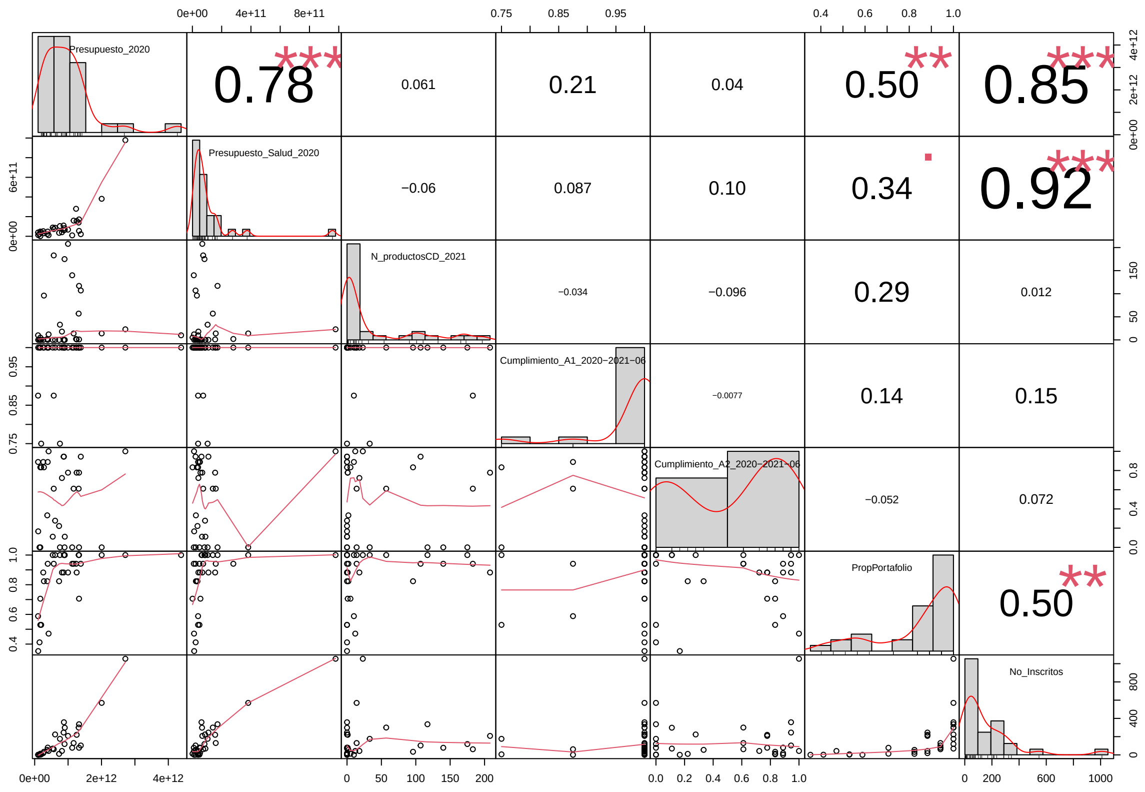

Figure 1: Correlación entre variables de PCA

```r
plot(pca1, main = 'Distribución de varianza')
```

```r
gg1 <- fviz_pca_ind(pca1,
             col.ind = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE
) + labs(title = 'PCA de individuos')
```

```r
gg1
```

```r
guardarGGplot(gg1, '001_fig_ind', 8, 6, fig_path)

gg2 <- fviz_pca_var(
  pca1,
  col.var = "contrib",
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  repel = TRUE
) + labs(title = 'PCA - Explicación de variables')
```

```r
gg2
```

```r
guardarGGplot(gg2, '002_fig_var', 8, 6, fig_path)

clean_data <- data %>%
  drop_na() %>%
```

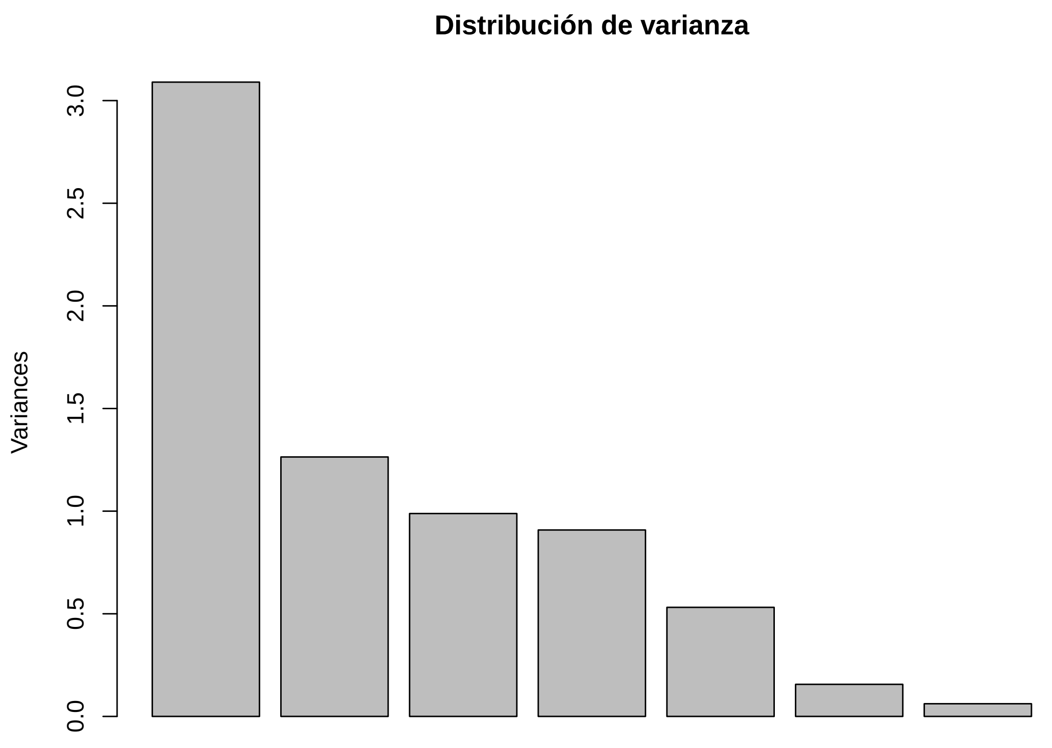

Figure 2: Distribución de las varianzas

```
  column_to_rownames('Departamento...2') %>%
  select(!contains('Departamento'))
```

2. Análisis de Clústers por Kmeans

```
funClusters <- function(data, k) {
  data %>%
    kmeans(k)
}
k.values <- data.frame(k = 1:30)
k.values['TWITH'] <- map_dbl(k.values$k, ~funClusters(clean_data, .x)$tot.withinss)

gg3 <- k.values %>%
  ggplot(aes(x = k, y = TWITH)) +
  geom_point() + geom_line() +
  geom_vline(xintercept = 4, lty = 'dashed', col = 'blue3') +
  ylab('Suma de cuadrados \n dentro de clústers') +
  xlab('N.° de Clusters, k')
```

```
gg3
```

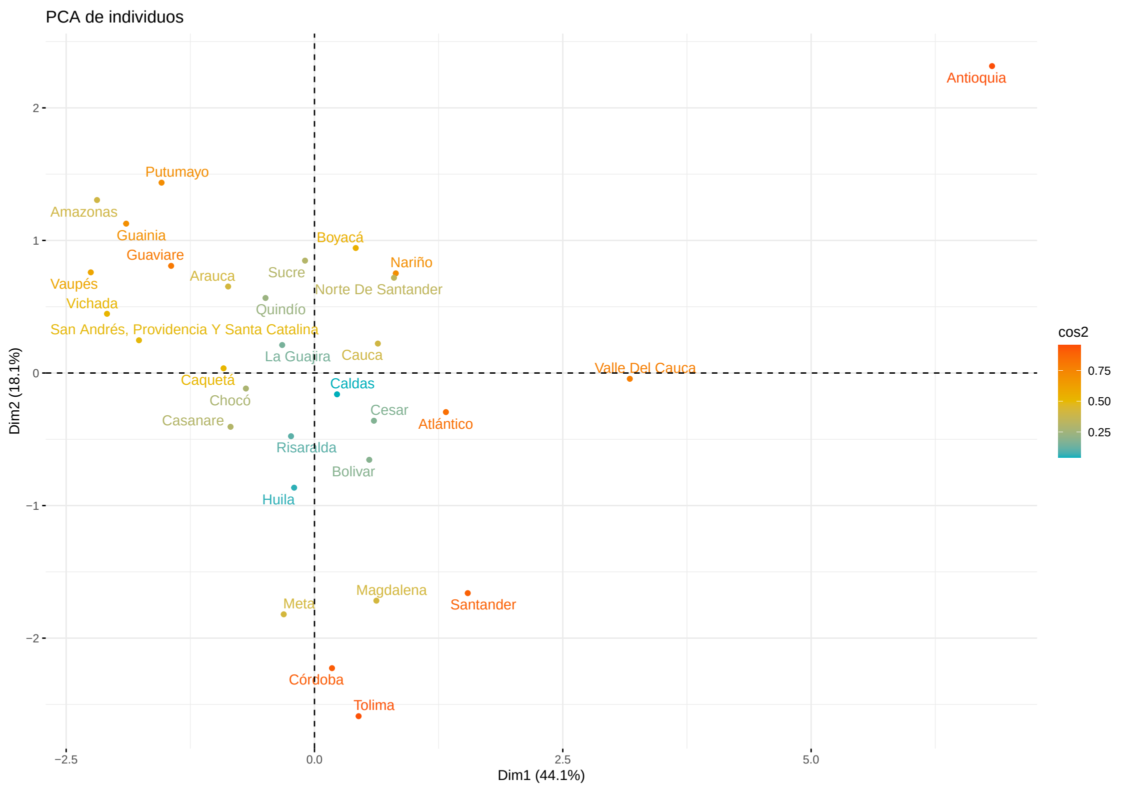

Figure 3: Individuos representados en PCA

```
guardarGGplot(gg3, '003_elbowlKmeans', 6, 4, fig_path)


g1 <- funClusters(clean_data, 4) %>%
  fviz_cluster(., data = clean_data) +
  theme_bw() + labs(title = NULL)
```

```
g1
```

```
guardarGGplot(g1, '004_clusterKmeans', 8, 5, fig_path)


g2 <- funClusters(clean_data, 4) %>%
  fviz_cluster(., data = clean_data, axes = c(1, 2)) +
  theme_bw() + labs(title = NULL)
g3 <- funClusters(clean_data, 4) %>%
  fviz_cluster(., data = clean_data, axes = c(1, 3)) +
  theme_bw() + labs(title = NULL)
g4 <- funClusters(clean_data, 4) %>%
  fviz_cluster(., data = clean_data, axes = c(2, 3)) +
  theme_bw() + labs(title = NULL)
g5 <- funClusters(clean_data, 4) %>%
  fviz_cluster(., data = clean_data, axes = c(1, 4)) +
  theme_bw() + labs(title = NULL)
```

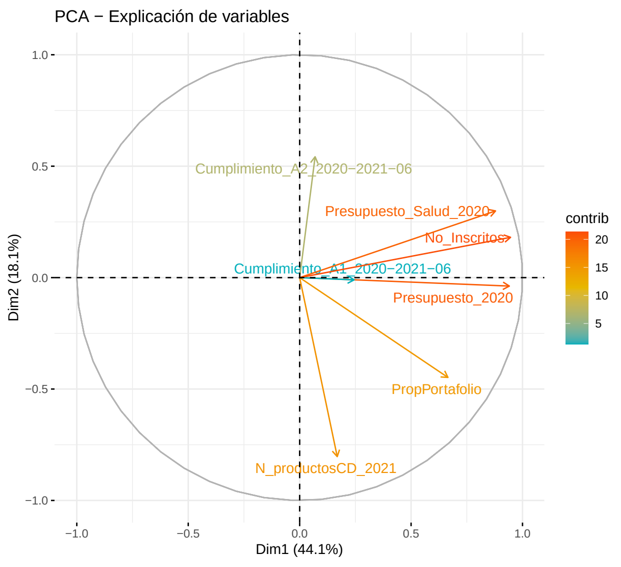

Figure 4: Explicación de variables en PCA

```
ggt <- wrap_plots(g2, g3, g4, g5)
```

```
ggt
```

```
guardarGGplot(ggt, '005_cluz_group', 12, 8, fig_path)
```

3. Clúster jerárquicos

```
funClusters_2 <- function(data, k) {
  t1 <- data %>%
    dist(method = 'euclidean') %>%
    hclust(method = 'complete')

  t2 <- cutree(t1, k)
  return(list(clust = t1, tree = t2))
}
```

```
p1 <- plot(funClusters_2(clean_data, 3)$clust, cex = 0.6, hang = -1, ylab = 'Altura',
     main = 'Dendrograma de clúster', xlab = NULL)
```

```
pdf(file.path(fig_path, '010_dendrograma.pdf'), width = 12, height = 8)
plot(funClusters_2(clean_data, 3)$clust, cex = 0.6, hang = -1, ylab = 'Altura',
     main = 'Dendrograma de clúster', xlab = NULL)
dev.off()
```

```
## pdf
##   2
```

```
saveRDS(p1, file.path(fig_path, '010_dendrograma.rds'))

gg1 <- funClusters_2(clean_data, 3)$clust$height %>%
  as.tibble() %>%
  add_column(groups = length(funClusters_2(clean_data, 3)$clust$height):1) %>%
```

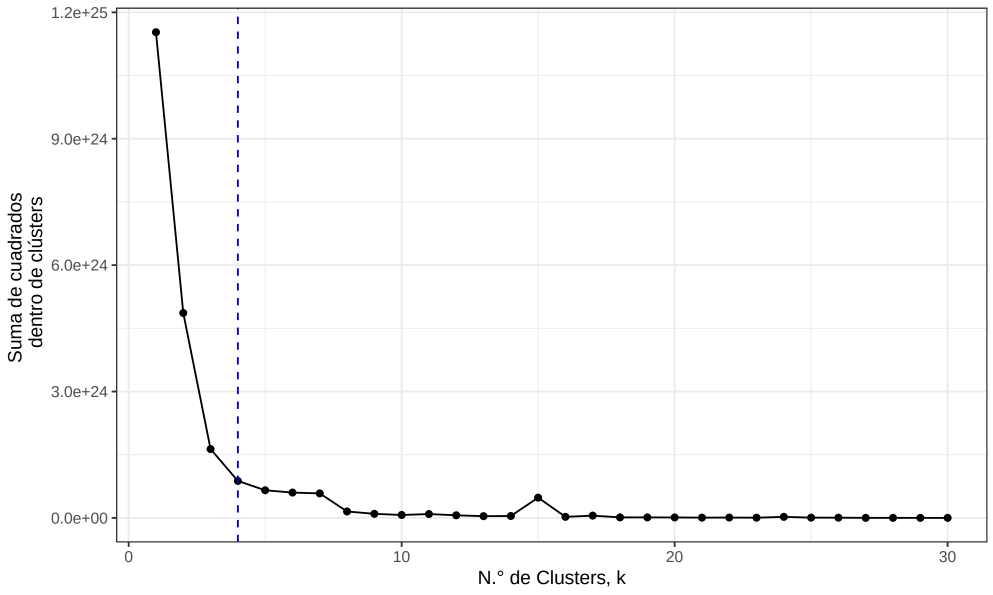

Figure 5: Criterio de codo para clústers por Kmeans

```
  rename(height = value) %>%
  ggplot(aes(x=groups, y = height)) +
  geom_point() + geom_line() +
  geom_vline(xintercept = 5, lty = 'dashed', col = 'blue3') +
  ylab('Altura') +
  xlab('N.° de Clusters, k')
```

```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
```

```
gg1
```

```
guardarGGplot(gg1, '012_elbowlWard', 6, 4, fig_path)

gg2 <- clean_data %>%
  {fviz_cluster(list(data = ., cluster = funClusters_2(., 5)$tree))} +
  theme_bw() + labs(title = NULL) +
  scale_color_discrete(name = 'Clúster') +
  scale_shape_discrete(name = 'Clúster') +
  scale_fill_discrete(name = 'Clúster')
```

```
gg2
```

```
guardarGGplot(gg2, '013_cluz_group', 8, 5, fig_path)


funClusters_3 <- function(axes) {
  clean_data %>%
    {fviz_cluster(list(data = ., cluster = funClusters_2(., 5)$tree),
                  axes = axes, ellipse = T)} +
    theme_bw() + labs(title = NULL) +
    scale_color_discrete(name = 'Clúster') +
```

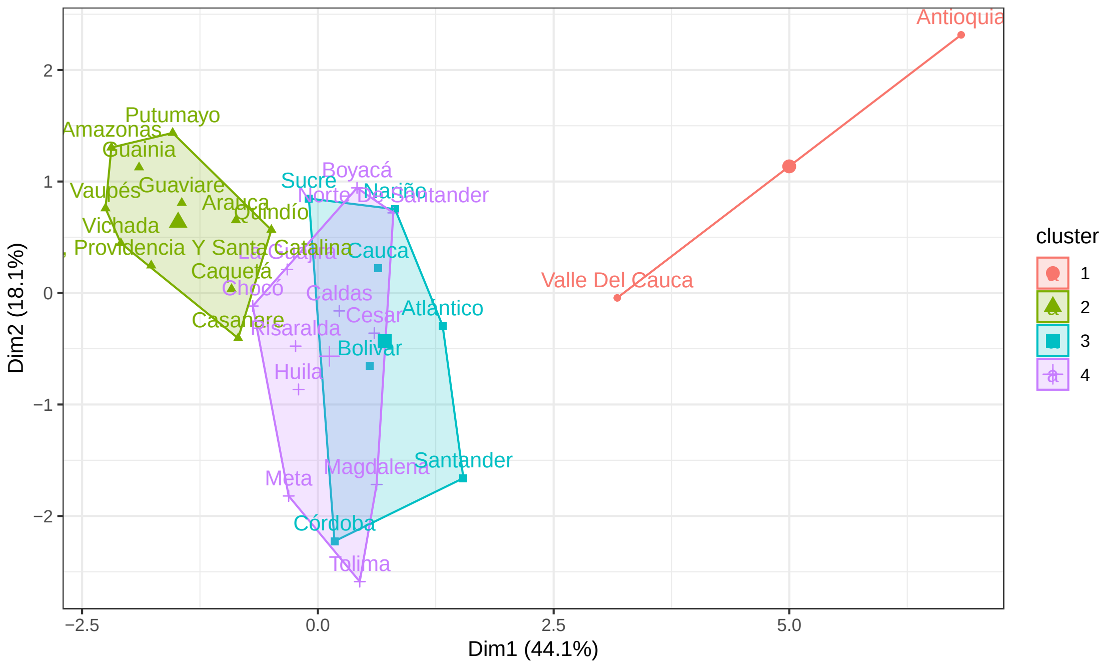

Figure 6: Clústers Kmeans visualizados en primeros dos PC

```
    scale_shape_discrete(name = 'Clúster') +
    scale_fill_discrete(name = 'Clúster')
}



g2 <- funClusters_3(c(1,2))
g3 <- funClusters_3(c(1,3))
g4 <- funClusters_3(c(2,3))
g5 <- funClusters_3(c(1,4))


ggt <- wrap_plots(g2, g3, g4, g5)
```

```
ggt
```

```
guardarGGplot(ggt, '014_cluz_group2', 12, 10, fig_path)
```

```
require(plotly)

trans_data <- as_tibble(pca1$x, rownames = 'Departamentos') %>%
  left_join(funClusters_2(clean_data, 5)$tree %>%
              as_tibble(rownames = 'Departamentos'), by = 'Departamentos') %>%
  rename(Grupo_hclus = value) %>%
  mutate(Grupo_hclus = as.integer(Grupo_hclus))


colors <- c('#4AC6B7', '#1972A4', '#965F8A', '#FF7070', '#C61951')
```

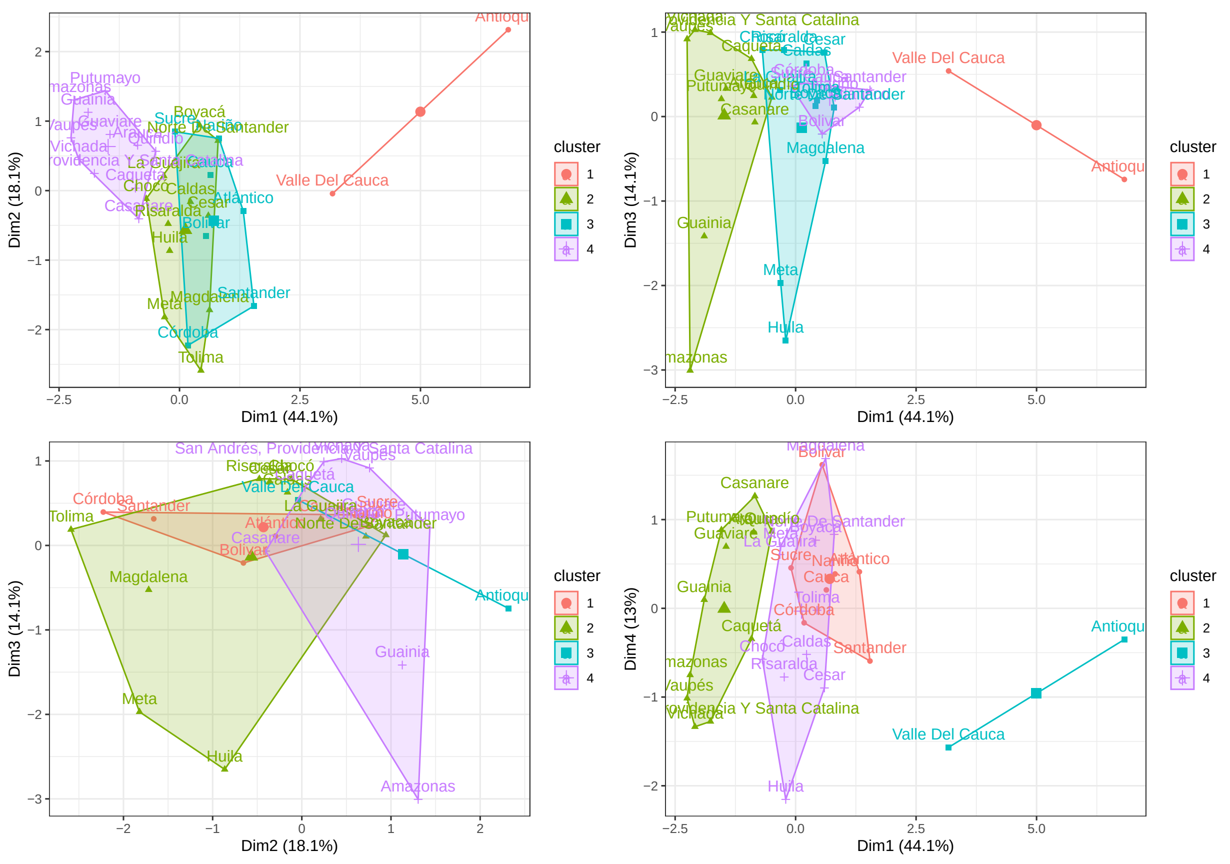

Figure 7: Clústers Kmeans visualizados en varios componentes

```r
fig <- trans_data %>%
  plot_ly(x = ~PC1, y = ~PC2, z = ~PC3, split = ~Grupo_hclus,
          colors = colors, name = ~Grupo_hclus, text = ~Departamentos,
          hovertemplate = "%{text}<br>PC1: %{x}<br>PC2: %{y}<br>PC3: %{z}")
```

```r
if (knitr::is_html_output()) {
  fig %>%
    add_markers()
}




trans_data1 <- data %>%
  left_join(funClusters_2(clean_data, 5)$tree %>%
              as_tibble(rownames = 'Departamentos'),
            by = c('Departamento...2' = 'Departamentos')) %>%
  rename(Grupo_hclus = value, Departamentos = Departamento...2) %>%
  mutate(Grupo_hclus = as.integer(Grupo_hclus))

fig1 <- trans_data1 %>%
  plot_ly(x = ~No_Inscritos, y = ~`Cumplimiento_A2_2020-2021-06`,
          z = ~PropPortafolio, split = ~Grupo_hclus,
```

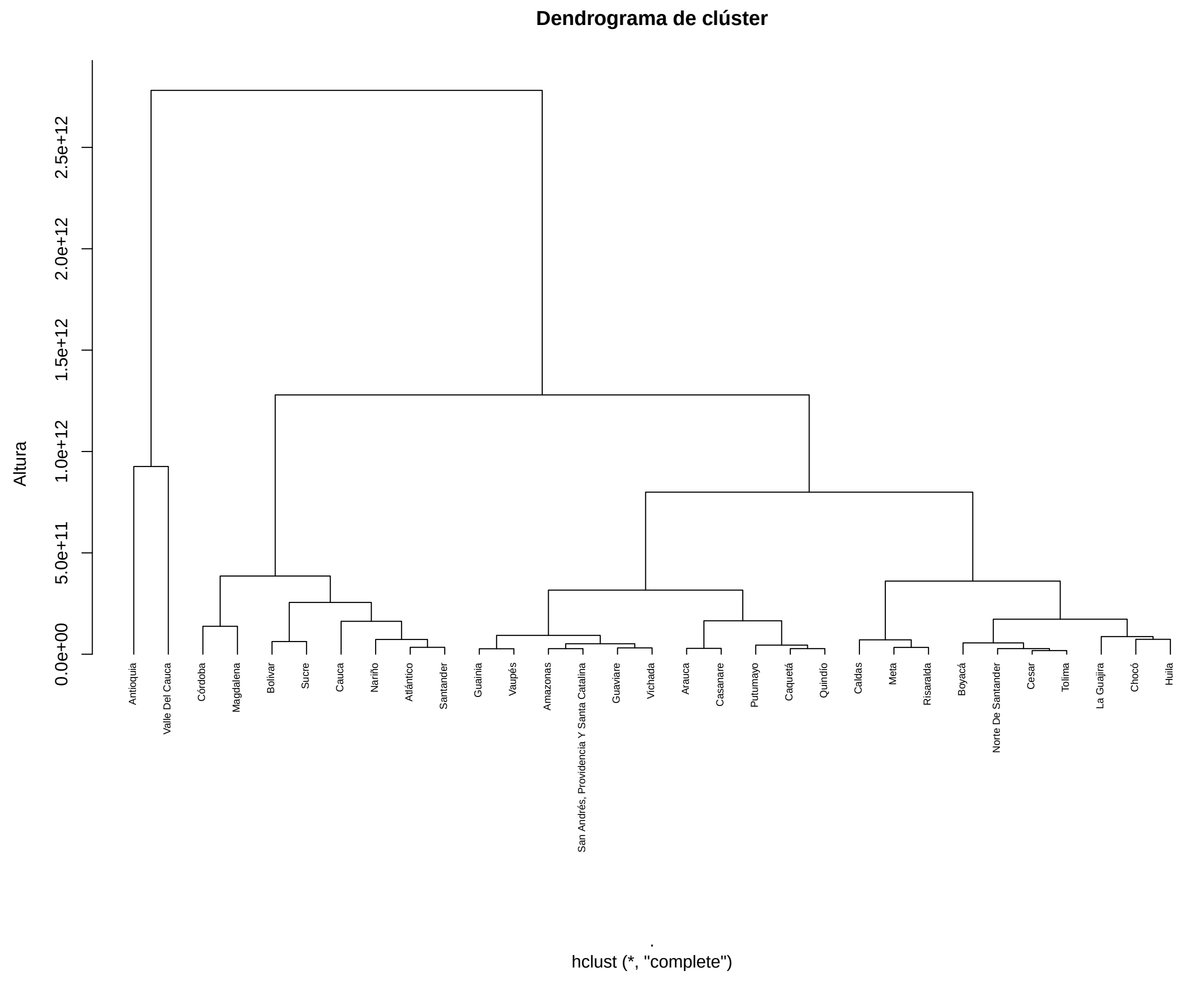

Figure 8: Dendrograma de análisis por clústers

```
           colors = colors, name = ~Grupo_hclus, text = ~Departamentos,
           hovertemplate = "%{text}<br>N.° inscritos: %{x}<br>Cumplimiento A2: %{y}<br>Prop. portafolio:
```

```
if (knitr::is_html_output()) {
  fig1 %>%
    add_markers()
}
```

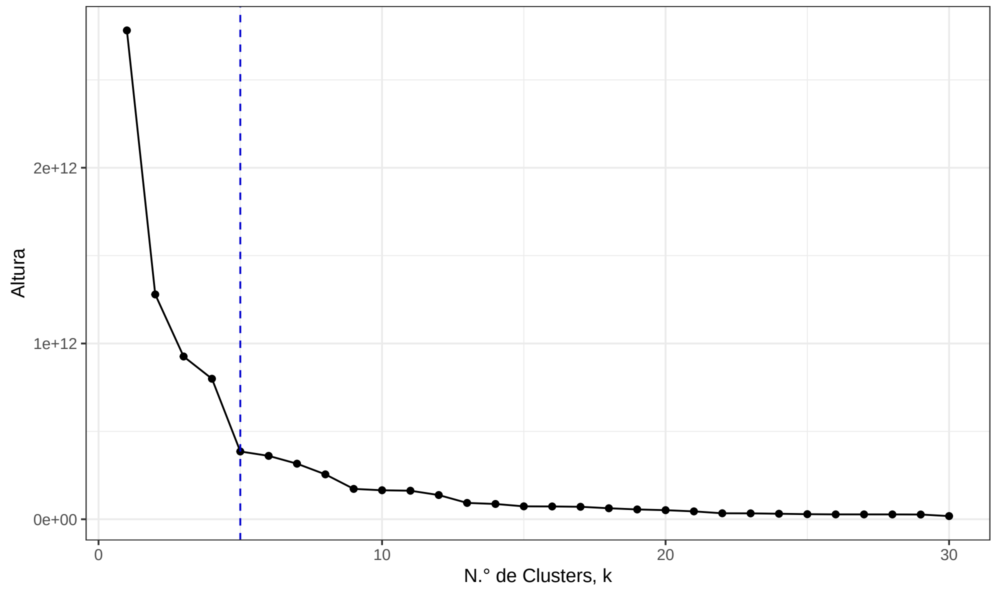

Figure 9: Criterio de codo para clústers jerárquicos

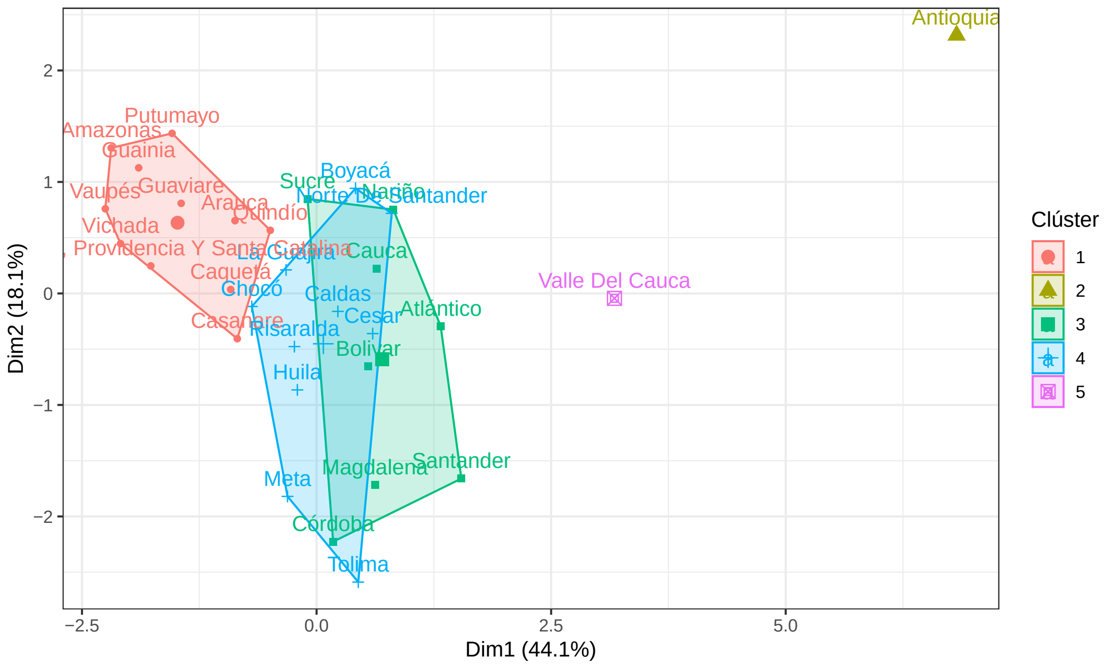

Figure 10: Clústers Jerárquicos visualizados en primeros dos componentes

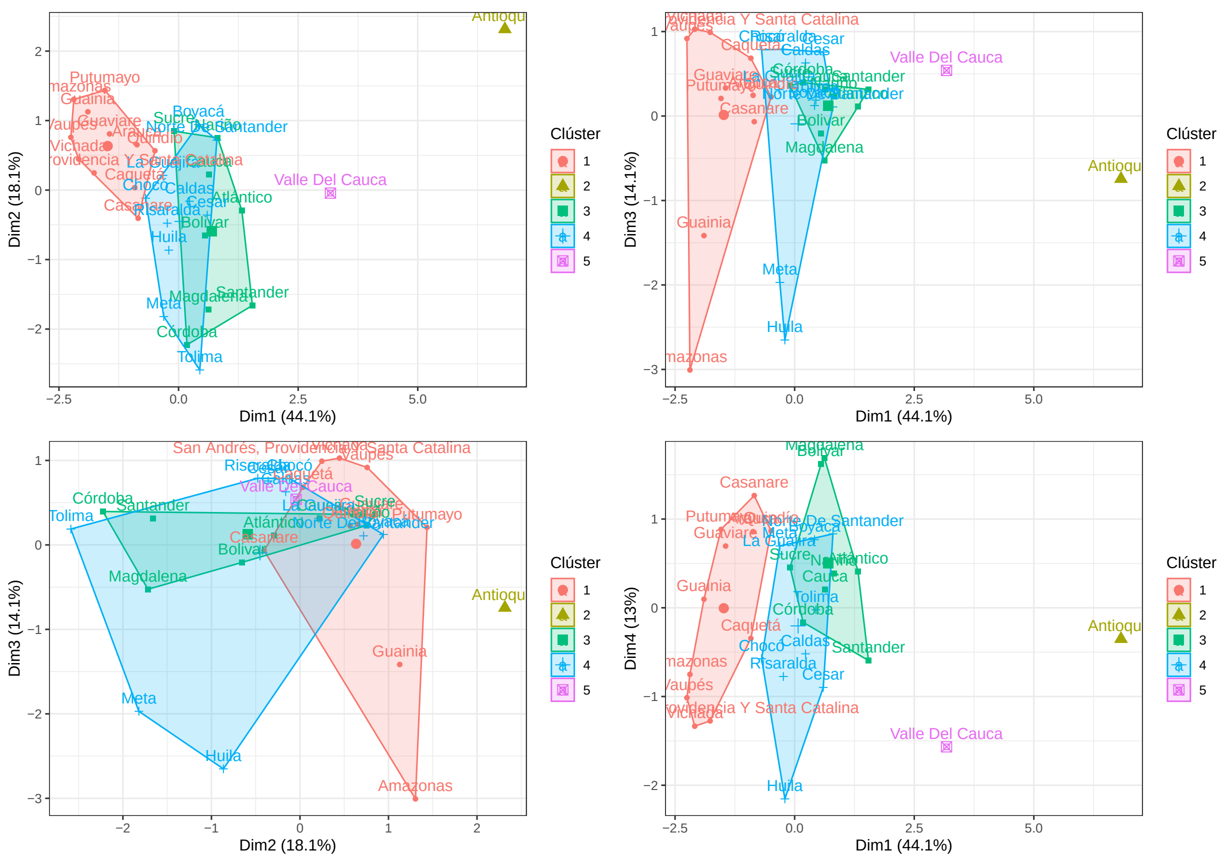
Figure 11: Clústers jerarquizados visualizados en varios componentes