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

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

```
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
```

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

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

1. Análisis por PCA

```
# 1. Análisis por PCA ----------------------------------------------------

pca1 <- data %>%
```

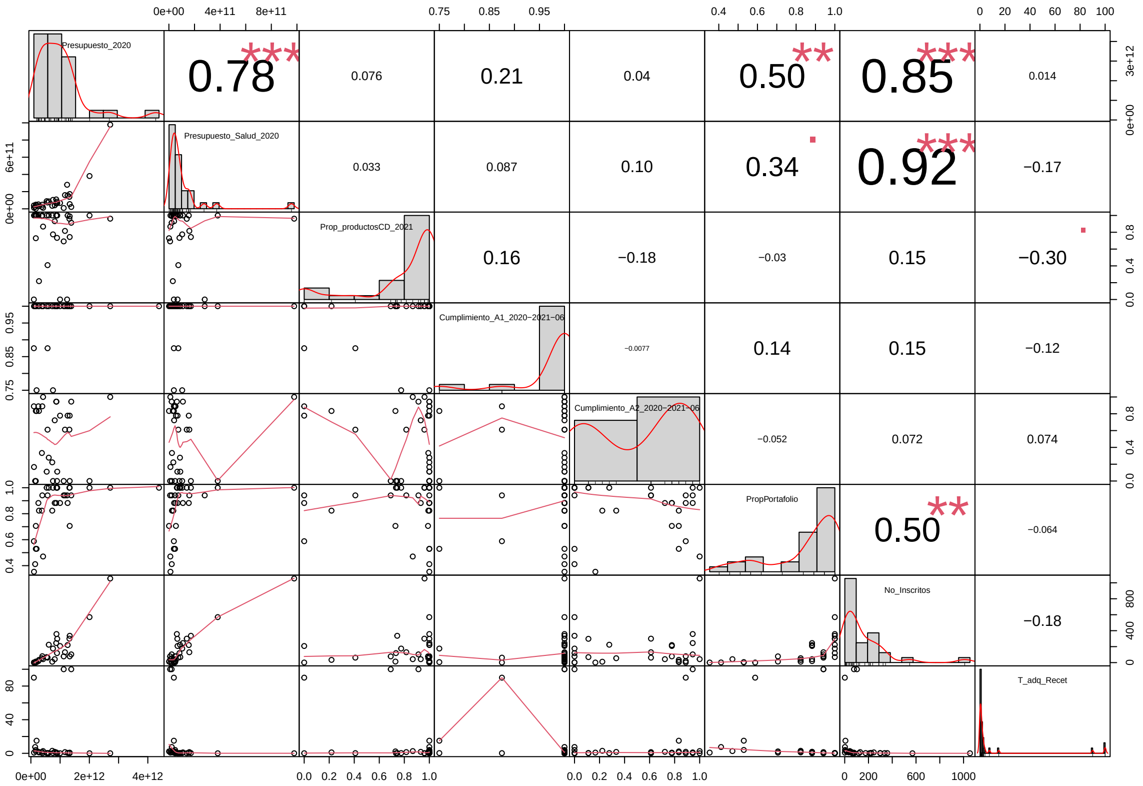

Figure 1: Correlación entre variables de PCA

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

dimVar <- pca1$sdev %>% {.^2*100/sum(.^2)}
```

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

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

```
gg1
```

```
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')
```

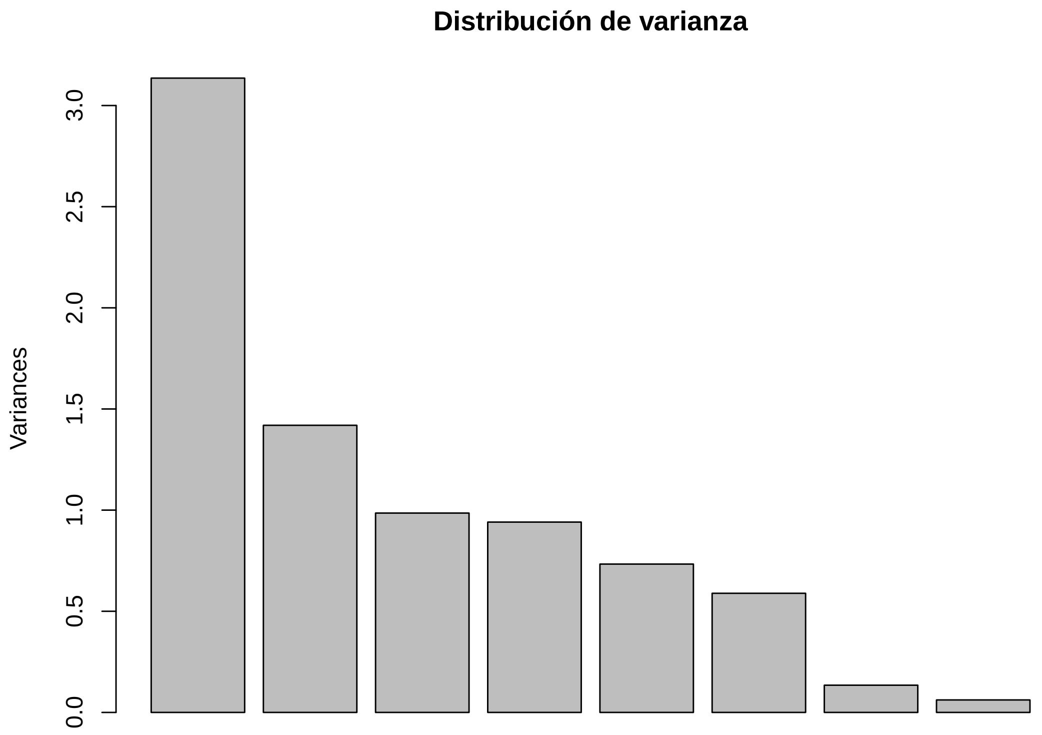

Figure 2: Distribución de las varianzas

```
gg2
```

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

clean_data <- data %>%
  drop_na() %>%
  column_to_rownames('Departamento...2') %>%
  select(!contains('Departamento'))
```

2. Análisis de Clústers por Kmeans

```
# 2. Análisis de Clústers por Kmeans ---------------------------------------

kmValid <- clValid(clean_data, 2:30, clMethods = 'kmeans',
                   validation = c('internal', 'stability'))

kmValid %>% optimalScores()
```

```
##                     Score Method Clusters
## APN          4.032258e-03 kmeans        2
## AD           1.149104e+09 kmeans       30
```

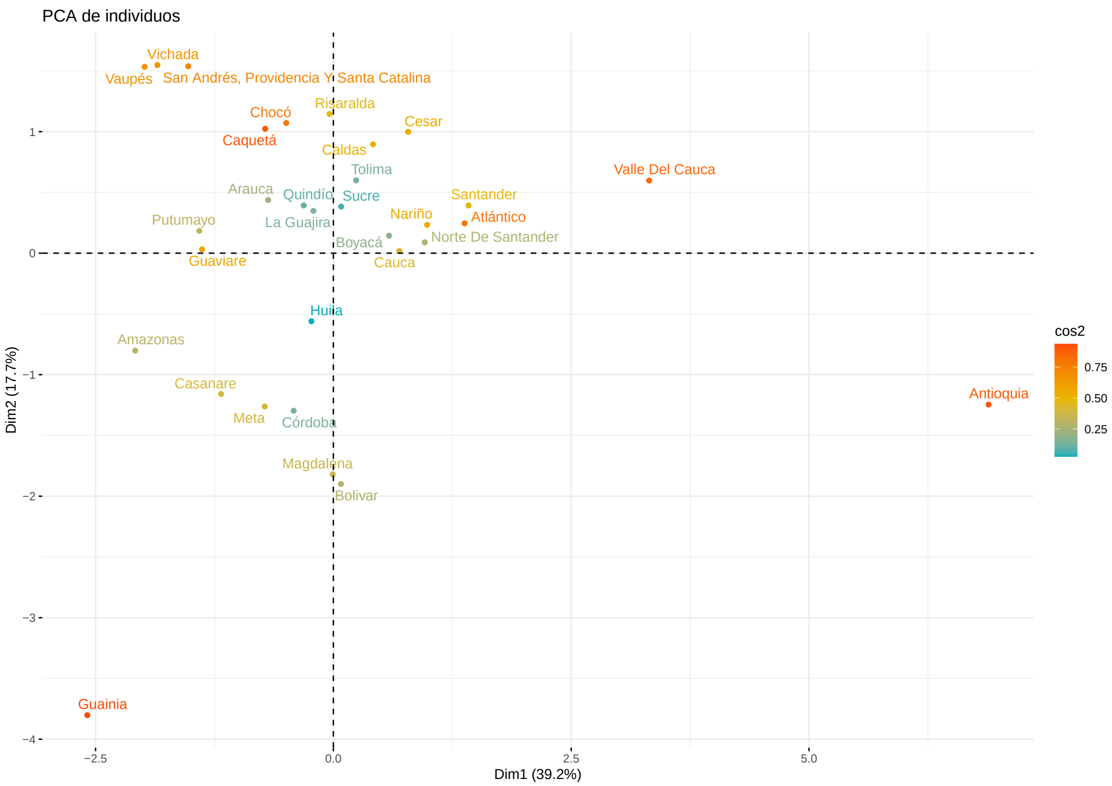

Figure 3: Individuos representados en PCA

```
## ADM          7.114046e+08 kmeans       30
## FOM          1.239604e+10 kmeans       30
## Connectivity 4.715079e+00 kmeans        2
## Dunn         1.451335e+00 kmeans       30
## Silhouette   6.792075e-01 kmeans        2
```

```
kmValidS <- as.data.frame(measures(kmValid)) %>%
  {rownames_to_column(as_tibble(t(.)), var = 'k')} %>%
  mutate(k = as.double(k) + 1,
         tot_withins_ss = map_dbl(k, ~ kmeans(clean_data, .x)$tot.withinss)) %>%
  pivot_longer(cols = !matches('k'))  %>%
  # Selección manual de óptimo de índices de validez
  mutate(koptim = case_when(
    name == 'tot_withins_ss' ~ 4,
    name == 'AD' ~ 5,
    name == 'FOM' ~ 7,
    name == 'Silhouette' ~ 4,
    TRUE ~ 3
  ))


gg3 <- kmValidS  %>%
  ggplot(aes(x = k, y = value)) +
  geom_point() + geom_line() +
  geom_vline(aes(xintercept = koptim), col = 'blue4', lty = 'dashed') +
  ggrepel::geom_label_repel(data = subset(kmValidS, k == koptim),
```

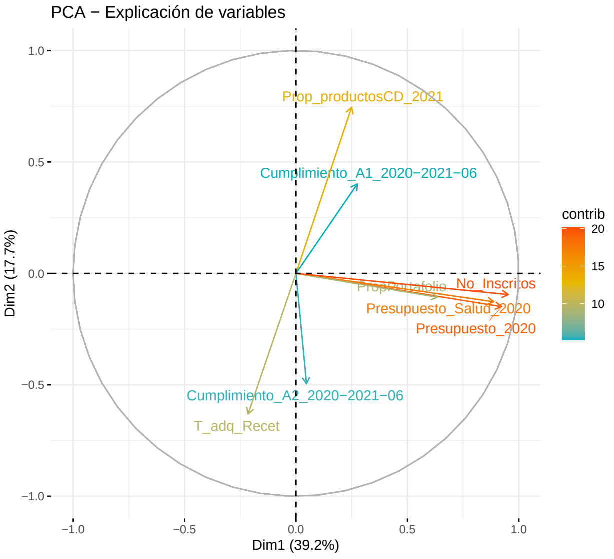

Figure 4: Explicación de variables en PCA

```
            aes(label = koptim)) +
  ylab('Valor') +
  scale_color_manual(values = c('Sí' = 'red', 'No' = NA)) +
  facet_wrap(vars(name), scales = 'free_y')
```

```
gg3
```

```
guardarGGplot(gg3, '003_elbowlKmeans', 8, 6, fig_path)


g1 <- kmeans(clean_data, 3) %>%
  fviz_cluster(., data = clean_data) +
  theme_bw() + labs(title = NULL)
```

```
g1
```

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


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

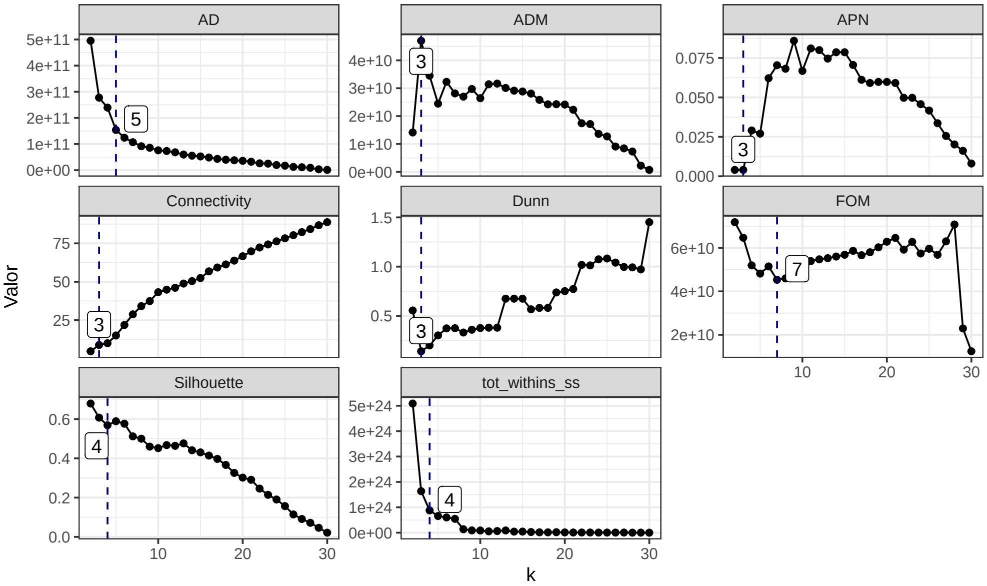

Figure 5: Criterio de codo para clústers por Kmeans

```
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() %>%
```

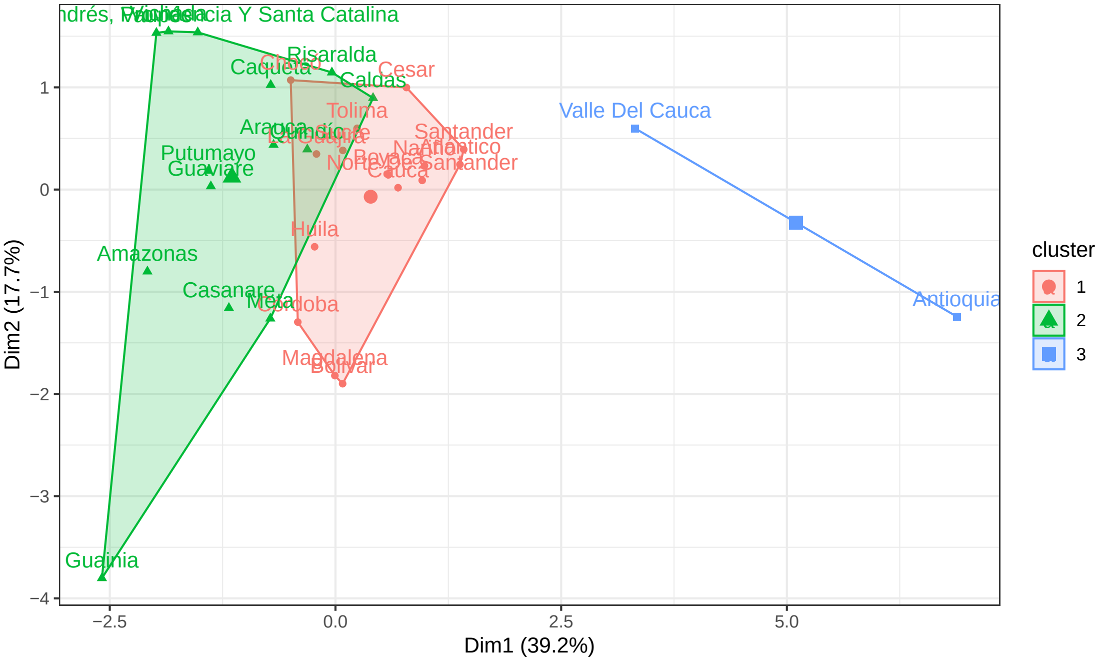

Figure 6: Clústers Kmeans visualizados en primeros dos PC

```
  add_column(groups = length(funClusters_2(clean_data, 3)$clust$height):1) %>%
  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)

# Índices de validez de clúster para algoritmo jerárquico
```

```
gg1b
```

```
guardarGGplot(gg1b, '003b_elbowlKmeans', 8, 6, fig_path)


gg2 <- clean_data %>%
  {fviz_cluster(list(data = ., cluster = funClusters_2(., 5)$tree))} +
  theme_bw() + labs(title = NULL) +
  scale_color_brewer(palette = 'Dark2', name = 'Clúster') +
  scale_fill_brewer(palette = 'Dark2', name = 'Clúster') +
  scale_shape_discrete(name = 'Clúster')
```

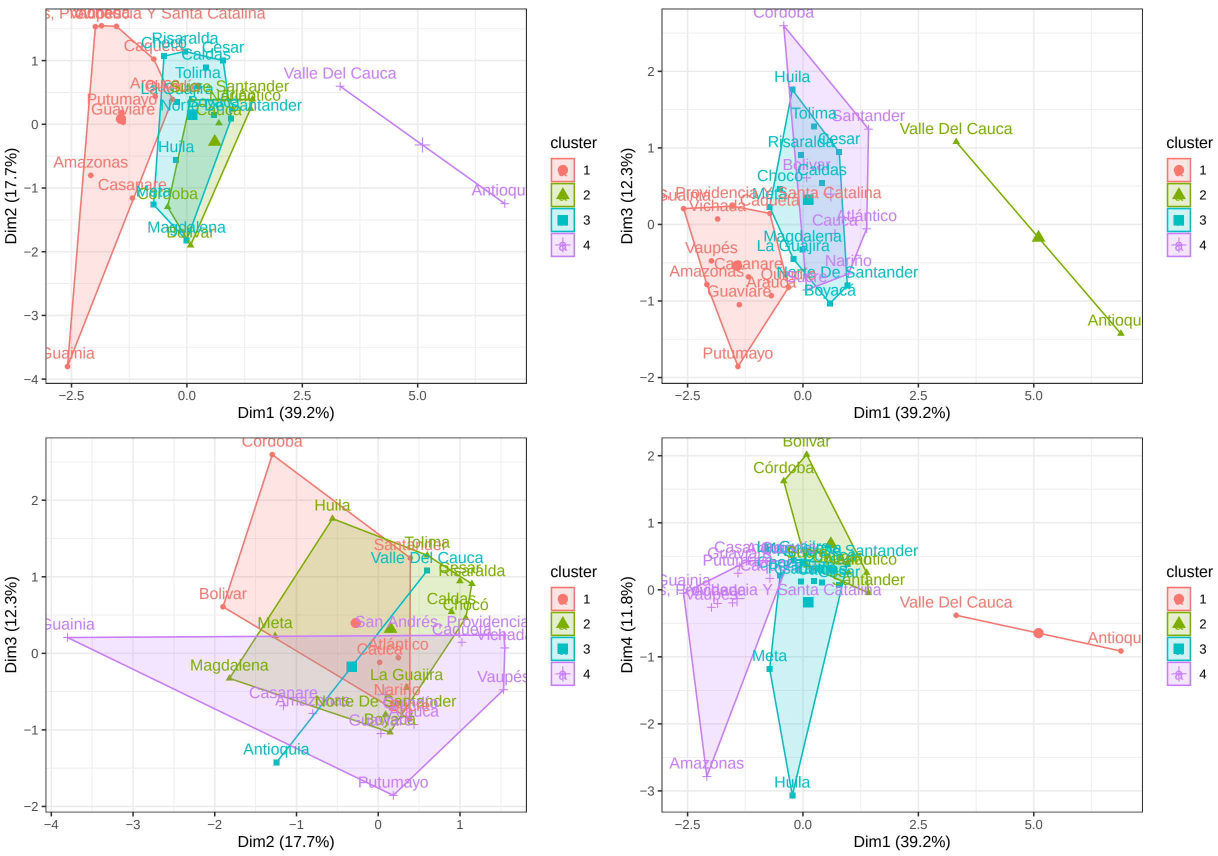

Figure 7: Clústers Kmeans visualizados en varios componentes

```
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') +
    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)
```

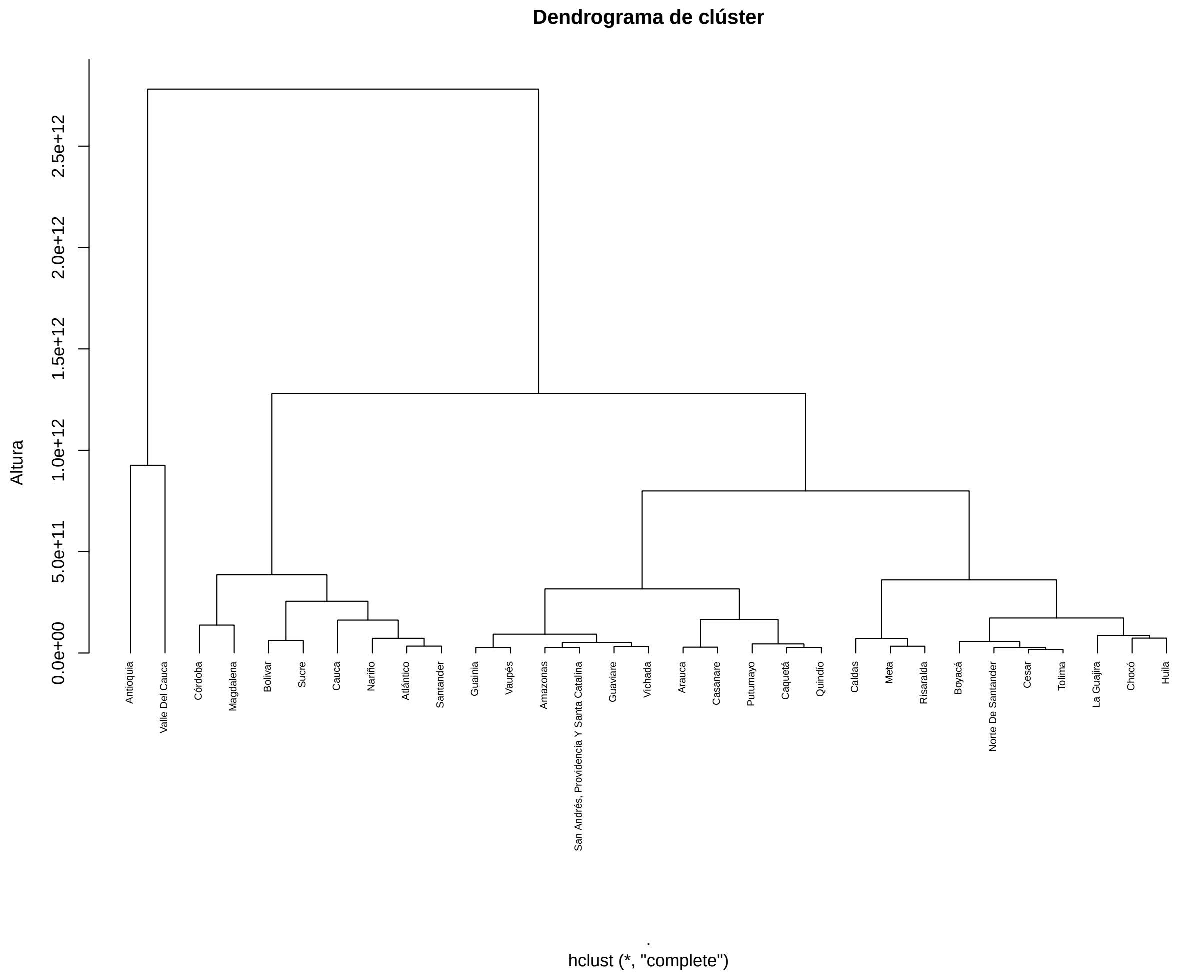

Figure 8: Dendrograma de análisis por clústers

```
ggt
```

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



colors <- c(`1` = '#a705b3', `2` = '#7081ff', `3` = '#ff0000',
            `4` = '#8cffb7', `5` = '#00ff5e')

# Creación de hover
hov_data <- data %>%
  mutate(
    Prec2020 = round(Presupuesto_2020/1e9, 2),
    PrecSalud2020 = round(Presupuesto_Salud_2020/1e9, 2),
    Hover = glue::glue(
      "<b>{Departamento...2}</b> <br>",
      "<i>Complejidad</i> <br>",
      "Presupuesto (miles de millones): {Prec2020} <br>",
      "Presupuesto Salud (miles de millones): {PrecSalud2020} <br>",
```

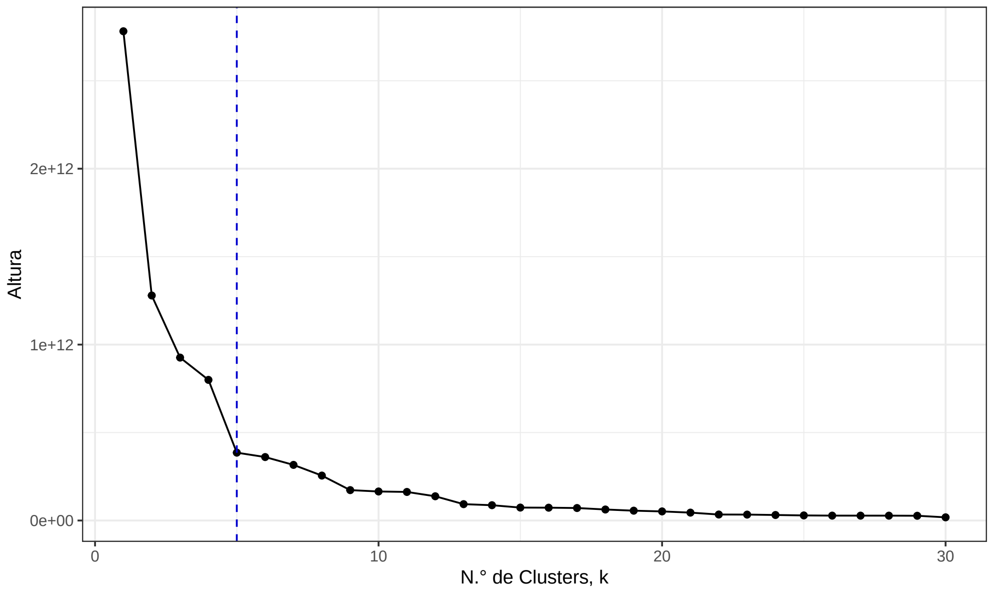

Figure 9: Criterio de codo para clústers jerárquicos

```
      "No. de inscritos: {No_Inscritos} <br>",
      "Prop. uso portafolio: {round(PropPortafolio,2)} <br>",
      "<i>Eficiencia</i> <br>",
      "Prop. ventas por FRE vs CD: {round(Prop_productosCD_2021,2)} <br>",
      "Prop. cumplimiento A1: {round(`Cumplimiento_A1_2020-2021-06`,2)} <br>",
      "Prop. cumplimiento A2: {round(`Cumplimiento_A2_2020-2021-06`,2)} <br>",
      "Raz. tiempo de adq rec vs no. rec: {round(T_adq_Recet, 2)}"
    )
  ) %>%
  select(Departamentos = Departamento...2, Hover)



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(colores = colors[Grupo_hclus],
         Grupo_hclus = factor(Grupo_hclus))



fig <- trans_data %>%
  left_join(hov_data, by = 'Departamentos') %>%
  plot_ly(name = ~Grupo_hclus) %>%
  add_trace(x = ~PC1, y = ~PC2, z = ~PC3,
            customdata = ~Hover,
            hovertemplate = "%{customdata}",
            mode = 'markers',
            type = 'scatter3d',
            color = ~Grupo_hclus,
            colors = "Dark2"
```

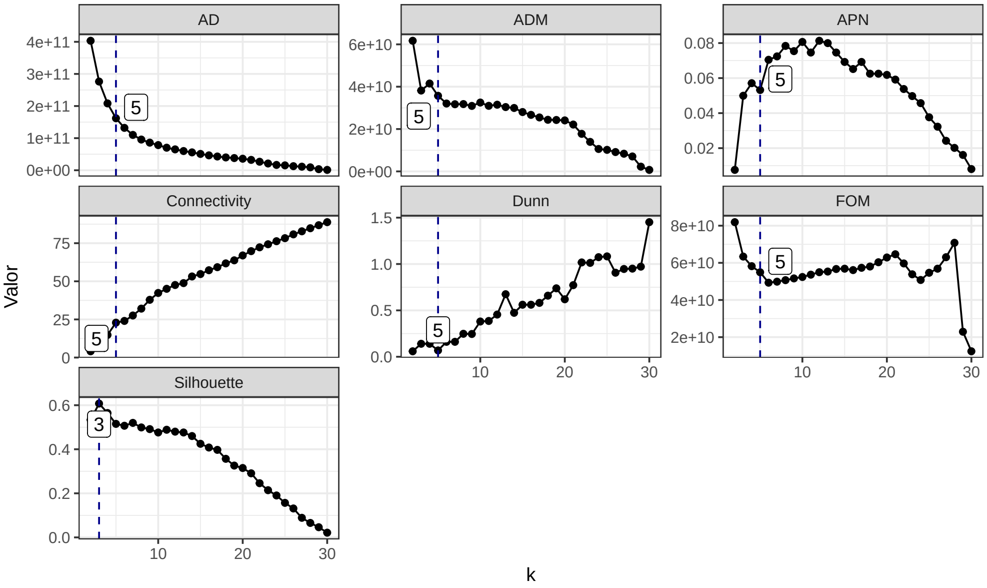

Figure 10: Criterios de codo para clústers por Jerárquico

```
            # marker = list(color = ~colores, size=6)
            ) %>%
  layout(scene = list(
    xaxis = list(title = paste0('PC1 (', round(dimVar[1], 1), '%)'), range = c(-7,+7)),
    yaxis = list(title = paste0('PC2 (', round(dimVar[2], 1), '%)'), range = c(-2.5,+2.5)),
    zaxis = list(title = paste0('PC3 (', round(dimVar[3], 1), '%)'), range = c(-3,+3))
  ))
```

```
if (knitr::is_html_output()) {
  fig
}

guardarPlotly(fig, '020_cluster_1', ruta = fig_path, libdir = 'plotly')

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))

tr_df1 <- trans_data1 %>%
  group_by(Grupo_hclus) %>%
  summarise(across(!matches('Departamento'), mean))

tr_df1 %>%
  write_csv(file.path('references', 'clusterRead.csv'))

trans_data1 %>%
  mutate(across(!matches('Departamento'), mean))
```

```
## # A tibble: 33 x 11
```

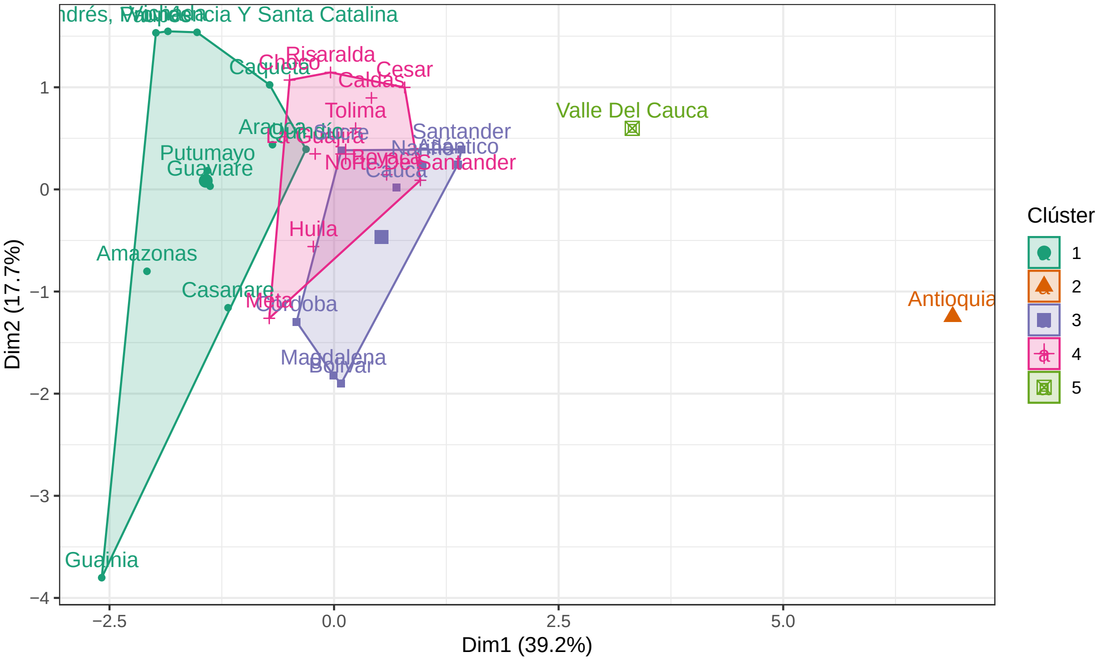

Figure 11: Clústers Jerárquicos visualizados en primeros dos componentes

```
##    Departamento...1 Departamentos Presupuesto_2020 Presupuesto_Salud_2020
##               <dbl> <chr>                    <dbl>                  <dbl>
##  1               22 Amazonas        921484848485.                     NA
##  2                3 Antioquia       921484848485.                     NA
##  3               25 Arauca          921484848485.                     NA
##  4                2 Atlántico       921484848485.                     NA
##  5               33 Bogotá D.c.     921484848485.                     NA
##  6                5 Bolivar         921484848485.                     NA
##  7               11 Boyacá          921484848485.                     NA
##  8               29 Caldas          921484848485.                     NA
##  9               21 Caquetá         921484848485.                     NA
## 10                8 Casanare        921484848485.                     NA
## # ... with 23 more rows, and 7 more variables: Prop_productosCD_2021 <dbl>,
## #   Cumplimiento_A1_2020-2021-06 <dbl>, Cumplimiento_A2_2020-2021-06 <dbl>,
## #   PropPortafolio <dbl>, No_Inscritos <dbl>, T_adq_Recet <dbl>,
## #   Grupo_hclus <dbl>
```

```
# group_by(Grupo_hclus) %>%
#   summarise(across(!matches('Departamento'), function(x) {
#     sum(ifelse(x > mean(x), 1, 0)) / n()
#   }))

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

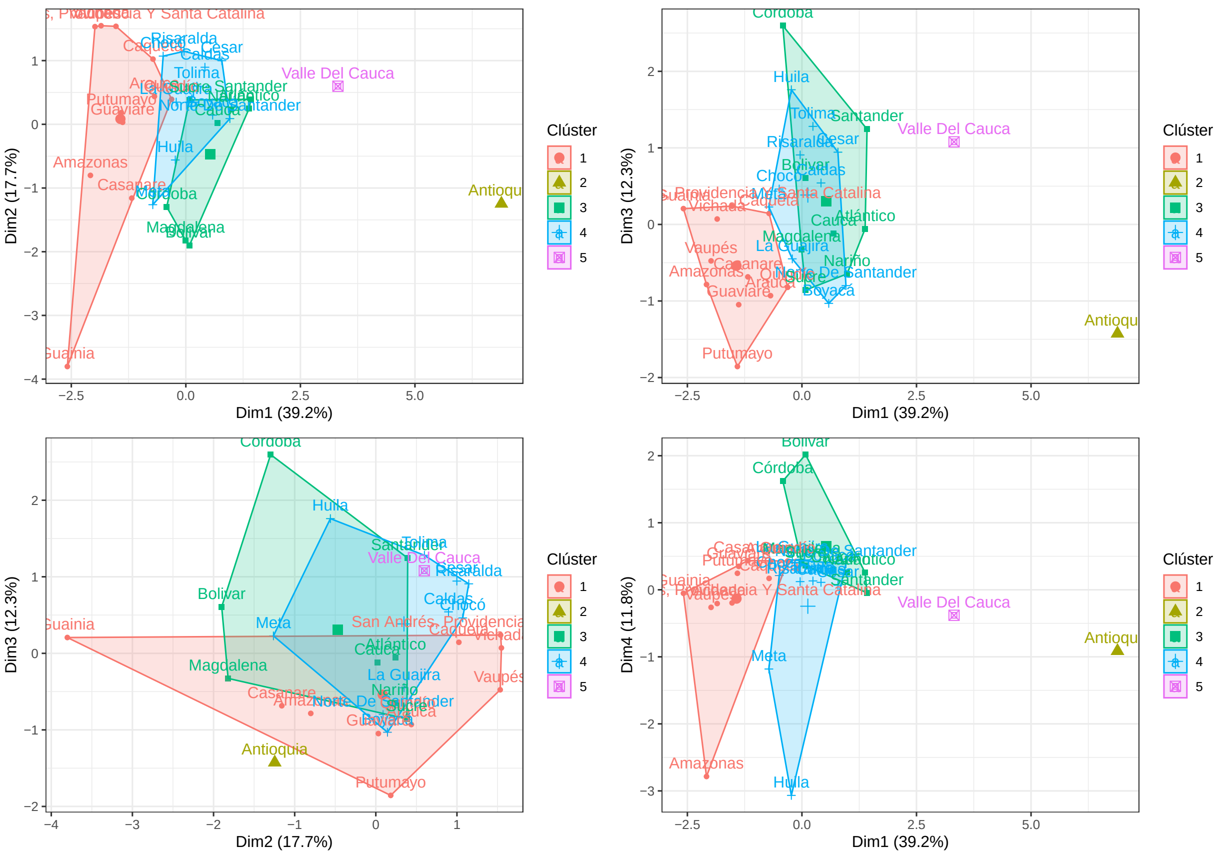

Figure 12: Clústers jerarquizados visualizados en varios componentes

```
}
```