Análisis de datos del IV CENAGRO

Marvin Quispe

4/9/2020

Analisis de datos del IV CENAGRO

El IV Censo Nacional Agropecuario 2012 (IV CENAGRO), es la investigación estadística más importante del Sector Agrario, que ha realizado el Instituto Nacional de Estadística e Informática (INEI), que proporciona datos actualizados para el conocimiento de la base productiva agropecuaria mediante el recojo de las declaraciones de todos los productores agropecuarios del país (INEI, 2012).

El análisis estadístico del IV CENAGRO permitirá evaluar las necesidades del sector agrario, lo que servirá de fundamento para el desarrollo de políticas y planes de desarrollo. El análisis realizado tiene como objetivo la clasificación socieconómica de agricultores del Perú en base al IV CENAGRO, actividad desarrollada en marco del proyecto "Late Bilght" y cuenta con las siguientes etapas:

Primero importamos las librerías o paquetes necesarios para el análisis y las bases de datos del IV CENAGRO para el análisis, posteriormente se seleccionan solo los datos para los cultivos de interés así como las variables de mayor importancia.

```
###################################
# IV CENSO NACIONAL AGROPECUARIO 2012 - LECTURA DE DATOS
# INPUT : CENAGRO database
# OUTPUT : Clustering and plots
# @MarvinQuispeSedano
# Nota: Tildes omitidas en los comentarios de codigo
###################################

# Librerias
pkgs = c("data.table", "cluster", "dplyr", "ggplot2", "Rtsne")
# install.packages(pkgs)
lapply(pkgs, library, character.only = TRUE)

###################################

# Configurar el directorio de trabajo
setwd("C:/Users/Asus/Documents/R/lateblight_tests/cenagro")
wd_data <- list.files("C:/Users/Asus/Documents/R/lateblight_tests/cenagro/combined/",
                      pattern=".rds", full=TRUE)

# Importar la primera base de datos
df1 <- readRDS(wd_data[1])
setDT(df1)

# Importar la segunda base de datos
df2 <- readRDS(wd_data[3])
setDT(df2)
```

```r
# Importar un .csv con el codigo de los cultivos
cc <- read.csv("code.csv",sep = ";", header = T)

# Hacer un merge entre las bases de datos 1 y 2 para las variables de interes
df_merge = merge(df1[,c(2:6,25:49,53:54)],
                 df2[,c(2:6,10:17)],
                 by=c("P001","P002","P003","P007X","P008"))

# Eliminar archivos para liberar la memoria
rm(df1);rm(df2)

# Obtener los datos segun nuestros requerimientos
df_crop <-df_merge[df_merge$P024_03 == "2612"]
df_crop <- merge(df_crop, cc, by.x="P024_03", by.y="code")
rm(df_merge)

################################
# Seleccionar las columnas que dispongan datos
colSums(is.na(df_crop))
df_crop2 <- df_crop[, !c("P027","P029_01", "P029_02", "P029_03")]
df_crop3 <- na.omit(df_crop2)
```

Lo segundo es obtener una muestra significativa de datos respecto y evaluar el número eficiente de clusters para el análisis con distancia de gower (k-medoid) y t-SNE para darle dimensionalidad al gráfico para su adecuada visualización.

```r
# Obtener una muestra de 10k datos y verificar significancia estadistica
sample_size = 10000
set.seed(1)
idxs = sample(1:nrow(df_crop3),sample_size,replace=F)

df_crop3 <- as.data.frame(df_crop3)
df_crop4 <- as.data.frame(df_crop3[idxs,])

pvalues = list()
for (col in names(df_crop3)) {
  if (class(df_crop3[,col]) %in% c("numeric","integer")) {
    # Numeric variable. Using Kolmogorov-Smirnov test

    pvalues[[col]] = ks.test(df_crop4[[col]],df_crop3[[col]])$p.value

  } else {
    # Categorical variable. Using Pearson's Chi-square test

    probs = table(df_crop3[[col]])/nrow(df_crop3)
    pvalues[[col]] = chisq.test(table(df_crop4[[col]]),p=probs)$p.value

  }
}

pvalues
```

```r
# Obtenemos la distancia de gower
gower_dist <- daisy(df_crop4, metric = "gower")

# Buscamos de 2 a 8 clusters
sil_width <- c(NA)
for(i in 2:8){
  pam_fit <- pam(gower_dist, diss = TRUE, k = i)
  sil_width[i] <- pam_fit$silinfo$avg.width
}

plot(1:8, sil_width,
     xlab = "Number of clusters",
     ylab = "Silhouette Width")
lines(1:8, sil_width)

# Obtenemos resultados para 5 clusters
k <- 5
pam_fit <- pam(gower_dist, diss = TRUE, k)

pam_results <- df_crop4 %>%
  mutate(cluster = pam_fit$clustering) %>%
  group_by(cluster) %>%
  do(the_summary = summary(.))

pam_results$the_summary

# Obtener SNE - BARNES
tsne_obj <- Rtsne(gower_dist, is_distance = TRUE)

tsne_data <- tsne_obj$Y %>%
  data.frame() %>%
  setNames(c("X", "Y")) %>%
  mutate(cluster = factor(pam_fit$clustering))
ggplot(aes(x = X, y = Y), data = tsne_data) +
  geom_point(aes(color = cluster))
```

Por último se puede agregar la metodología “Hierarchical Clustering” para observar los datos agrupados en un gráfico tipo dendograma.

```r
# Hierarchical Clustering

hc <- hclust(gower_dist, "ward.D2")
rect.hclust(hc , k = 5, border = 2:6)
abline(h = 5, col = 'red')

cut_avg <- cutree(hc, k = 5)
df_crop_cluster <- mutate(df_crop4, cluster = cut_avg)

ggplot(df_crop_cluster, aes(x=LONG_DECI, y = LAT_DECI, color = factor(cluster))) +
  geom_point()
```