TFM_UOC

Ariel Cariaga Martínez

2024-12-25

```r
# Librerías
library(knitr)
library(kableExtra)
library(fastDummies)
library(xgboost)
library(tidyverse)
library(naniar)
library(corrplot)
library(caret)
library(DMwR2) #imputación knn
library(reshape2)
library(FactoMineR)
library(factoextra)
library(ggrepel)
library(MASS) # LDA
library(car)
library(multcomp)
library(pROC)
library(fastshap)

#Otras posibles
#library(readxl)
#library(psych)
#library(DataExplorer)
#library(GGally)
#library(ROSE)
#library(ggfortify)
```

OBTENCIÓN DEL DATASET FINAL

Tras reunión de coordinación, seleccionaremos demográficas y clínicas relevantes.

La columna del espectro clínico de la primera infección tiene NAs, con una proporción de 10 asintomáticos, 35 hospitalizados, 185 en mild-moderated y 7 en UCI. La imputación que generaré es con el valor más común.

```r
#Seleccionamos el espectro clínico de la primera infección
column_spec <- bbddAECM$spec1
selected_spec <- data.frame(spec1 = column_spec)

#Imputación de valores faltantes en la columna 'spec1'
most_frequent <- names(sort(table(selected_spec), decreasing = TRUE))[1]

#Imputamos los valores faltantes con el más frecuente
```

```r
selected_spec[is.na(selected_spec)] <- most_frequent

#Generamos dummies solo para 'spec1'
selected_spec1_dummy <- dummy_cols(selected_spec,
                                   remove_first_dummy = TRUE,
                                   remove_selected_columns = TRUE)
```

Ahora seleccionamos las demográficas de interés.

```r
#Variables sociodemográficas indicadas en la reunión de coordinación.
selected_sociodemographic <- bbddAECM %>%
  dplyr::select(sex, ag, el, ptg19)

#Creamos dummies para las variables demográficas seleccionadas.
selected_sociodemographic_dummies <- dummy_cols(
  selected_sociodemographic,
  remove_first_dummy = TRUE,
  remove_selected_columns = TRUE
)
```

Ahora seleccionamos las variables neuropsicológicas de interés.

```r
#Realizmos selección de tests neuropsicológicos de interés.
selected_neuropsychological <- bbddAECM %>%
  dplyr::select(tn46, tn52, tn36,
                tn38, tn40, tn42,
                tn22, tn44, tn14,
                tn24, tn12, tn6,
                tn30, tn34, tn8,
                tn48, tn50)
```

Y ahora que ya tenemos todas la variables de interés las combinamos en un único dataset.

```r
#Combinamos las variables seleccionadas en un solo dataset.
df_def <- bind_cols(selected_sociodemographic_dummies, selected_spec1_dummy, selected_neuropsychological
```

Resta crear la variable Target sobre la que generaremos el estudio.

```r
# Seleccionamos NoCOVID y No LongCOVID (que será un grupo) vs LongCovid Cog
column_cluster <- bbddAECM$cluster
selected_cluster <- data.frame(cluster = column_cluster)

#Creamos la nueva variable Target
selected_cluster$cluster <- ifelse(
  selected_cluster$cluster %in% c("No COVID", "No LongCOVID"), "No_LC",
  ifelse(selected_cluster$cluster == "LongCOVID Cog", "LC_Cog", NA)  # Excluir "LongCOVID NoCog"
)

#Convertimos Target a factor
selected_cluster$cluster <- factor(selected_cluster$cluster, levels = c("No_LC", "LC_Cog"))

# Convertir a numérico restando 1 para que los niveles sean 0 y 1
selected_cluster$cluster_binary <- as.numeric(selected_cluster$cluster) - 1
```

Finalmente unimos todas las variables definitivamente seleccionadas.

```r
#Unimos la columna de los targets al dataframe final
df_def <- cbind(df_def, Target = selected_cluster$cluster_binary)
```

```r
#-------------------------------------------------------------------------------
##COMPROBAMOS QUE LOS INDICES COINCIDEN EN LA BBDD ORIGINAL Y EN LA RE-CREADA
#-------------------------------------------------------------------------------

#Obtenemos los índices de las observaciones donde cluster es "LongCOVID NoCog"
indices <- which(bbddAECM$cluster == "LongCOVID NoCog")

#Verificamos
print(indices)
```

```
 [1]   1  22  42  55  58  60  61  62  63  64  66  68 184 192 195 199 205 209 211
[20] 230 234 237 238 241 243 250 263
```

```r
#Obtenemos los índices de las observaciones donde Target es NA en el nuevo dataset
indices_na <- which(is.na(df_def$Target))

#Verificamos
print(indices_na)
```

```
 [1]   1  22  42  55  58  60  61  62  63  64  66  68 184 192 195 199 205 209 211
[20] 230 234 237 238 241 243 250 263
```

```r
# ELIMINAMOS TODAS LAS OBSERVACIONES NAS EN TARGET
df_def <- df_def[!is.na(df_def$Target), ]

#Dado que los índices coinciden, df_def será el dataset que utilizaremos para realizar el TFM

#Aquí remodificamos los nombres para evitar problemas
df_def <- df_def %>% rename_with(make.names)
```

```r
#-------------------------------------------------------------------------------
# Análisis general del dataset de trabajo final generado
#-------------------------------------------------------------------------------

# Ver tipos de variables y estructura
str(df_def)
```

```
'data.frame':   241 obs. of  29 variables:
 $ ag                 : num  26.9 60.5 33.1 44.6 61.9 43 55.6 53.6 46.3 51.2 ...
 $ ptg19              : num  38.2 26.3 18.2 24.1 32.8 ...
 $ sex_Woman          : int  1 0 1 1 0 1 1 1 1 1 ...
 $ el_Elementary      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ el_High.School     : int  1 0 0 0 0 0 0 0 0 0 ...
 $ el_Secondary       : int  0 0 0 0 0 0 0 0 0 0 ...
 $ el_Specialist.Master : int  0 1 0 0 1 1 1 0 1 0 ...
 $ el_University.Deg.  : int  0 0 0 1 0 0 0 1 0 1 ...
 $ spec1_Hospitalization: int  0 1 0 0 1 0 0 0 0 1 ...
 $ spec1_Mild.Moderated : int  1 0 1 1 0 1 1 1 1 0 ...
 $ spec1_UCI           : int  0 0 0 0 0 0 0 0 0 0 ...
 $ tn46                : num  0.3 1 0.7 0 0 0.3 -0.3 -0.3 -0.7 -0.3 ...
 $ tn52                : num  1 0 2 -0.6 0 ...
 $ tn36                : num  -1 0 1 0.3 0.3 -0.7 -0.3 -0.3 -1.7 0 ...
 $ tn38                : num  -0.3 0.3 0.3 0 -0.3 -0.3 -0.7 0 -2.3 -0.3 ...
 $ tn40                : num  0 0 1 1 0.3 0.3 -0.3 -0.3 -0.3 -0.3 ...
 $ tn42                : num  -1.3 0 0.3 1.3 -0.3 -0.3 -1 -1.7 -0.7 0.3 ...
```

```
 $ tn22                : num  0.4 0.4 -0.5 -0.9 -0.2 1 0.4 0.3 0.3 0.1 ...
 $ tn44                : num  -0.7 -1.3 -1 1 -0.3 0 -0.7 1 -2 -0.7 ...
 $ tn14                : num  -1.7 1 0.3 -0.3 1.3 0.7 0.3 0.7 -0.3 0.7 ...
 $ tn24                : num  -3 0 -1 -0.7 -0.3 -0.7 0.4 -0.7 -2.7 -0.7 ...
 $ tn12                : num  -2 0 1.3 -0.3 1.3 0 1 0 0.7 0 ...
 $ tn6                 : num  -2.06 0.86 1.7 -0.82 -0.57 1.38 -1.18 -0.2 -0.01 -0.32 ...
 $ tn30                : num  -2.3 0.7 0.32 -0.43 0.4 1.71 -1.22 0.03 -0.07 0.03 ...
 $ tn34                : num  -1 0.3 -1 -1 0.3 0 -0.3 -1 1 -2 ...
 $ tn8                 : num  2 -0.3 0.3 -0.7 -0.3 2.7 0 -0.3 -0.3 0.3 ...
 $ tn48                : num  1 3.1 3.1 1.3 3.1 1.3 1.3 3.1 1 0.8 ...
 $ tn50                : num  -1.7 1.3 -0.3 -0.3 -0.7 0.7 1.3 0.7 0.3 0.7 ...
 $ Target              : num  1 1 0 1 1 1 1 1 1 1 ...
```

```r
# Resumen estadístico
summary(df_def)
```

```
      ag             ptg19          sex_Woman        el_Elementary
 Min.   :25.40   Min.   :16.05   Min.   :0.0000   Min.   :0.00000
 1st Qu.:42.40   1st Qu.:23.01   1st Qu.:1.0000   1st Qu.:0.00000
 Median :48.80   Median :26.19   Median :1.0000   Median :0.00000
 Mean   :48.72   Mean   :27.69   Mean   :0.8008   Mean   :0.04149
 3rd Qu.:55.40   3rd Qu.:30.60   3rd Qu.:1.0000   3rd Qu.:0.00000
 Max.   :70.80   Max.   :87.31   Max.   :1.0000   Max.   :1.00000
                 NA's   :6
 el_High.School    el_Secondary     el_Specialist.Master el_University.Deg.
 Min.   :0.0000   Min.   :0.00000   Min.   :0.0000       Min.   :0.0000
 1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000       1st Qu.:0.0000
 Median :0.0000   Median :0.00000   Median :0.0000       Median :0.0000
 Mean   :0.3278   Mean   :0.03734   Mean   :0.1494       Mean   :0.3776
 3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:0.0000       3rd Qu.:1.0000
 Max.   :1.0000   Max.   :1.00000   Max.   :1.0000       Max.   :1.0000


 spec1_Hospitalization spec1_Mild.Moderated   spec1_UCI            tn46
 Min.   :0.0000        Min.   :0.0000       Min.   :0.0000   Min.   :-2.7000
 1st Qu.:0.0000        1st Qu.:1.0000       1st Qu.:0.0000   1st Qu.:-0.7000
 Median :0.0000        Median :1.0000       Median :0.0000   Median : 0.0000
 Mean   :0.1411        Mean   :0.7967       Mean   :0.0249   Mean   :-0.1021
 3rd Qu.:0.0000        3rd Qu.:1.0000       3rd Qu.:0.0000   3rd Qu.: 0.3000
 Max.   :1.0000        Max.   :1.0000       Max.   :1.0000   Max.   : 2.7000
                                                             NA's   :3
      tn52              tn36             tn38             tn40
 Min.   :-2.30000   Min.   :-3.0000   Min.   :-2.3000   Min.   :-3.0000
 1st Qu.:-0.40000   1st Qu.:-0.7000   1st Qu.:-0.7000   1st Qu.:-0.9250
 Median : 0.00000   Median :-0.3000   Median :-0.3000   Median :-0.3000
 Mean   : 0.02143   Mean   :-0.2453   Mean   :-0.1378   Mean   :-0.2483
 3rd Qu.: 0.40000   3rd Qu.: 0.3000   3rd Qu.: 0.3000   3rd Qu.: 0.3000
 Max.   : 2.00000   Max.   : 2.3000   Max.   : 2.7000   Max.   : 2.7000
 NA's   :3          NA's   :3         NA's   :3         NA's   :3
      tn42              tn22             tn44             tn14
 Min.   :-3.1000   Min.   :-2.3000   Min.   :-3.000   Min.   :-2.3000
 1st Qu.:-1.3000   1st Qu.:-0.1000   1st Qu.:-0.700   1st Qu.:-0.3000
 Median :-0.7000   Median : 0.5000   Median :-0.300   Median : 0.3000
 Mean   :-0.7286   Mean   : 0.3982   Mean   :-0.363   Mean   : 0.4055
 3rd Qu.: 0.0000   3rd Qu.: 0.9000   3rd Qu.: 0.225   3rd Qu.: 1.0000
 Max.   : 2.3000   Max.   : 3.0000   Max.   : 2.300   Max.   : 2.7000
```

```
NA's   :3          NA's   :4          NA's   :3          NA's    :3
     tn24               tn12               tn6                tn30
Min.   :-3.0000    Min.   :-2.0000    Min.   :-5.9400    Min.    :-2.53000
1st Qu.:-1.0000    1st Qu.:-0.3000    1st Qu.:-1.0500    1st Qu.:-0.78000
Median :-0.3000    Median : 0.1500    Median :-0.1900    Median : 0.03000
Mean   :-0.4887    Mean   : 0.2479    Mean   :-0.1951    Mean    : 0.03378
3rd Qu.: 0.0000    3rd Qu.: 1.0000    3rd Qu.: 0.8575    3rd Qu.: 0.97000
Max.   : 2.3000    Max.   : 2.3000    Max.   : 2.2700    Max.    : 1.73000
NA's   :3          NA's   :3          NA's   :3          NA's    :3
     tn34               tn8                tn48               tn50
Min.   :-2.7000    Min.   :-2.0000    Min.   :0.300      Min.    :-2.0000
1st Qu.:-0.7000    1st Qu.:-0.7000    1st Qu.:1.000      1st Qu.: 0.0000
Median :-0.3000    Median :-0.3000    Median :1.300      Median : 0.3000
Mean   :-0.3571    Mean   :-0.1202    Mean   :1.615      Mean    : 0.2294
3rd Qu.: 0.0000    3rd Qu.: 0.3000    3rd Qu.:3.000      3rd Qu.: 0.7000
Max.   : 2.7000    Max.   : 2.7000    Max.   :3.100      Max.    : 3.0000
NA's   :3          NA's   :3          NA's   :3          NA's    :3
     Target
Min.   :0.000
1st Qu.:0.000
Median :1.000
Mean   :0.722
3rd Qu.:1.000
Max.   :1.000
```

```r
# if (!requireNamespace("webshot", quietly = TRUE)) {
#   install.packages("webshot")
#   webshot::install_phantomjs()
# }
# library(webshot)
#
# # Crear un data frame a partir del summary
# summary_table <- as.data.frame(summary(df_def))
#
# save_kable(
#   kable(summary_table, format = "html", caption = "Resumen de df_def") %>%
#     kable_styling(full_width = FALSE, bootstrap_options = c("striped", "hover", "condensed")),
#   file = "tabla_resumen.html"
# )
#
# # Convertir el archivo HTML a JPG
# webshot("tabla_resumen.html", file = "tabla_resumen.jpg", vwidth = 800, vheight = 600)

# Ver las dimensiones y primeras filas
cat("Dimensiones del dataset:", dim(df_def), "\n")
```

```
Dimensiones del dataset: 241 29
```

Vamos a verificar si hay algún valor faltante

```r
#-------------------------------------------------------------------------------
# Valores faltantes
#-------------------------------------------------------------------------------

pdf("outputs/images/01_NAs.pdf", width = 16, height = 10)
```
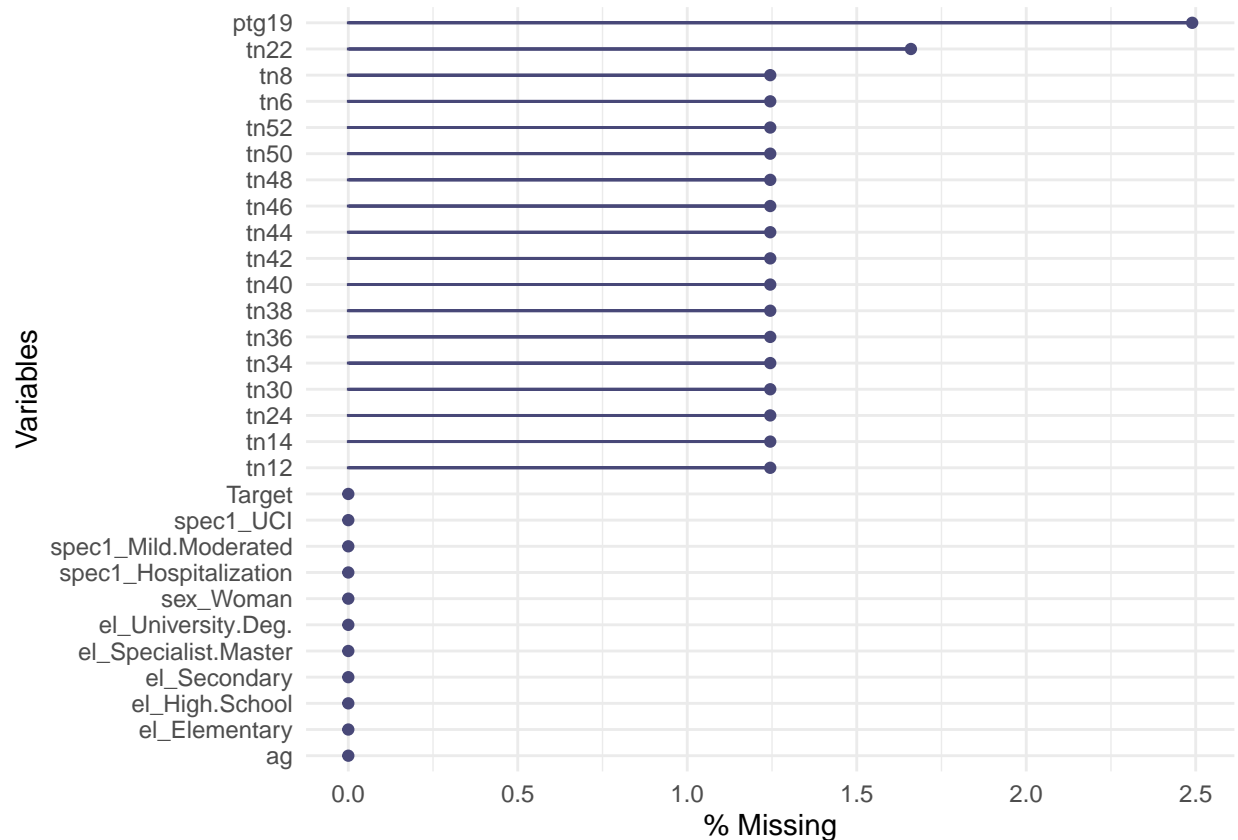
```r
# Resumen de valores faltantes
gg_miss_var(df_def, show_pct = TRUE)

dev.off()
```

pdf
  2

```r
gg_miss_var(df_def, show_pct = TRUE)
```


Como no tenemos una idea global del tipo de valores faltantes, vamos a hacer una imputación con kNN para estos.

```r
#Excluimos columnas específicas
excluded_cols <- c("Target")

#Separamos las columnas excluidas
excluded_data <- df_def %>%
  dplyr::select(all_of(excluded_cols))

#Convertimos todas las columnas numéricas a tipo double
data_for_imputation_selected <- df_def %>% dplyr::select(-all_of(excluded_cols))
data_for_imputation_selected <- data_for_imputation_selected %>%
  mutate(across(where(is.numeric), as.numeric))

#Aplicamos la imputación
selected_data_imputed <- knnImputation(data_for_imputation_selected, k = 10)
```

```r
#Combinamos las columnas imputadas con las excluidas
df_def_clean <- bind_cols(selected_data_imputed, excluded_data)

# Gráfico tras imputación
pdf("outputs/images/02_NAs_tras_imputacion.pdf", width = 16, height = 10)

gg_miss_var(df_def_clean, show_pct = TRUE)

dev.off()
```

```
pdf
  2
```

```r
gg_miss_var(df_def_clean, show_pct = TRUE)
```
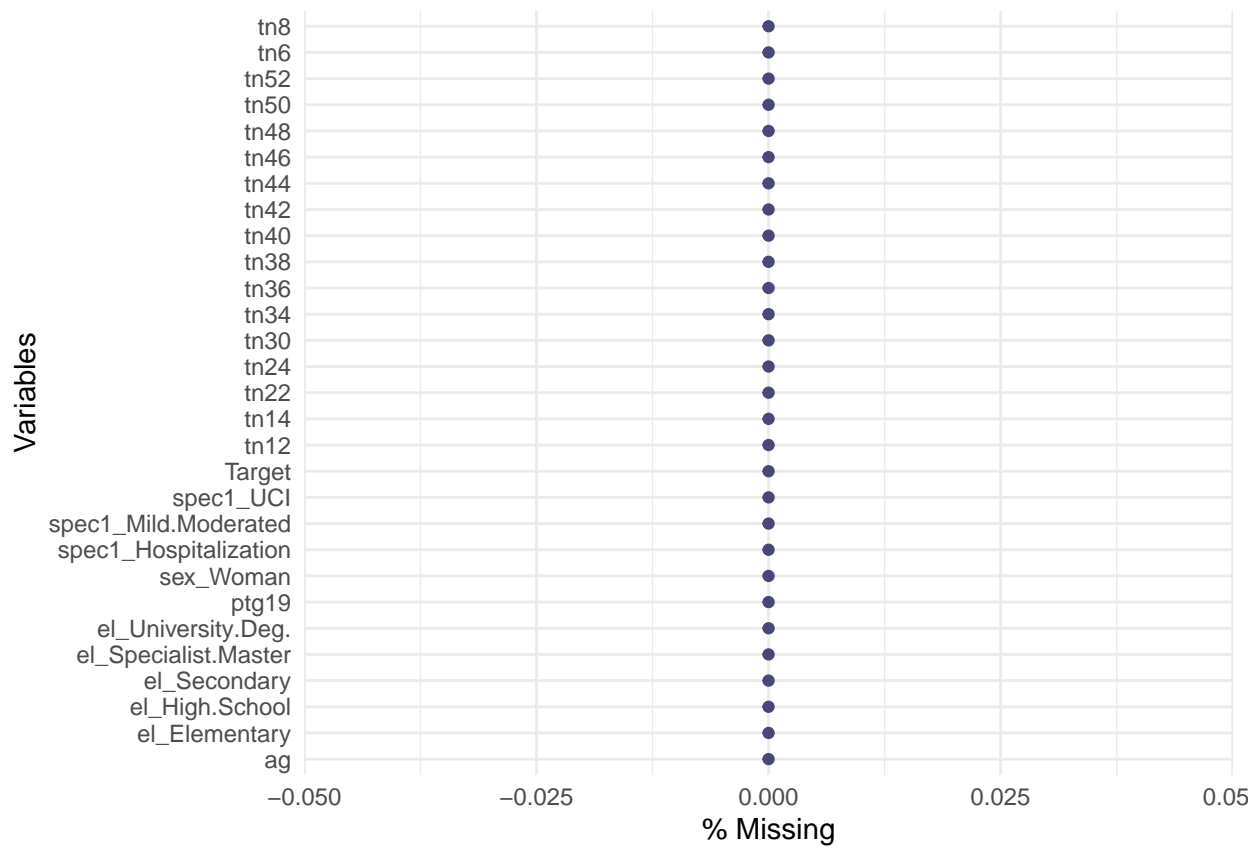

Vamos a dejar el código del balanceo del dataset y guardamos el dataset balanceado por si fuera de interés para utilizarlo posteriormente.

```r
# ----------------------
# Balancear el dataset
# ----------------------

table(df_def_clean$Target) #Hay 64 vs 174
```

```
  0   1
 67 174
```

```
#Si aplicmos ROSE: OJO: DATOS SINTÉTICOS
#df_def_clean_balanced_ROSE <- ROSE(Target ~ ., data = df_combined_clean, seed = 123, method = "under")

# Oversampling manual
minority_class <- df_def_clean %>% filter(Target == "0")
oversampled <- minority_class %>% sample_n(size = nrow(df_def_clean[df_def_clean$Target == "1", ]), repl

#Combinamos con la clase mayoritaria
df_def_clean_balanced <- bind_rows(
  df_def_clean %>% filter(Target == "1"),
  oversampled
)

table(df_def_clean_balanced$Target)
```

```
  0   1
174 174
```

Análisis univariado de variables numéricas

```
# ----------------------
# Análisis univariado
# ----------------------

# a) Seleccionar variables numéricas
numeric_vars_def <- df_def_clean %>% dplyr::select(where(is.numeric))

pdf("outputs/images/03_univariado_num.pdf", width = 16, height = 10)

# Histogramas para variables numéricas
numeric_vars_def %>%
  gather(key = "Variable", value = "Value") %>%
  ggplot(aes(x = Value)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  facet_wrap(~ Variable, scales = "free") +
  theme_minimal() +
  labs(title = "Distribuciones de variables numéricas en conjunto de datos final")

dev.off()
```
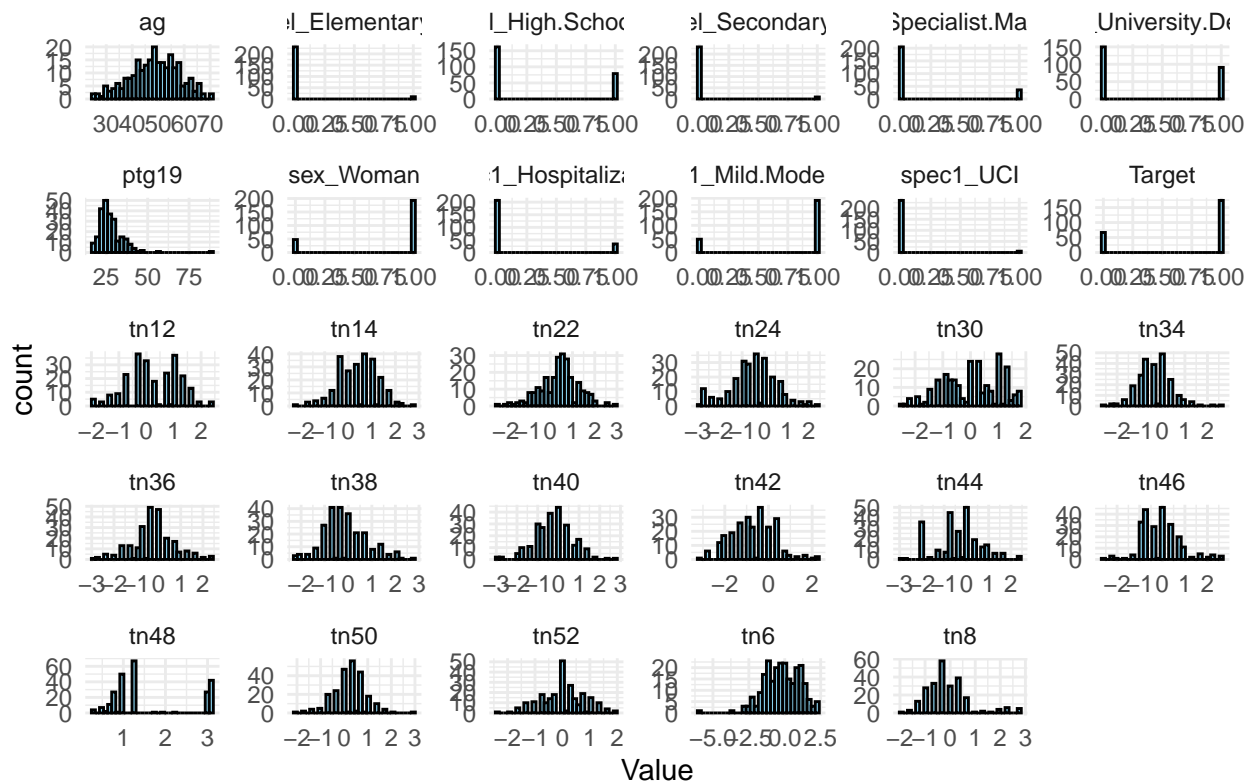
```
pdf
  2
```

```
# Histogramas para variables numéricas
numeric_vars_def %>%
  gather(key = "Variable", value = "Value") %>%
  ggplot(aes(x = Value)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  facet_wrap(~ Variable, scales = "free") +
  theme_minimal() +
  labs(title = "Distribuciones de variables numéricas en conjunto de datos final")
```

Distribuciones de variables numéricas en conjunto de datos final

Análisis univariado de variables categóricas

```r
# b) Variables categóricas
# Seleccionar columnas específicas y transformarlas a factor: mantenemos el nombre cambiado
# para evitar afectar el dataset generado con columnas numéricas que usaremos en los modelos
df_def_clean_factors <- df_def_clean %>%
  mutate(across(
    where(~ is.numeric(.) && all(. %in% c(0, 1)) && n_distinct(.) == 2),
    as.factor
  ))

# b.1) Selección de variables categóricas
categorical_vars_def <- df_def_clean_factors %>% dplyr::select(where(is.factor))

pdf("outputs/images/04_categóricas.pdf", width = 16, height = 10)

# Gráficos de barras para variables categóricas
categorical_vars_def %>%
  gather(key = "Variable", value = "Value") %>%
  ggplot(aes(x = Value, fill = Variable)) +
  geom_bar() +
  facet_wrap(~ Variable, scales = "free") +
  theme_minimal() +
  labs(title = "Distribuciones de variables categóricas (dataset final)")

dev.off()
```

pdf

```r
# Gráficos de barras para variables categóricas
categorical_vars_def %>%
  gather(key = "Variable", value = "Value") %>%
  ggplot(aes(x = Value, fill = Variable)) +
  geom_bar() +
  facet_wrap(~ Variable, scales = "free") +
  theme_minimal() +
  labs(title = "Distribuciones de variables categóricas (dataset final)")
```

Distribuciones de variables categóricas (dataset final)
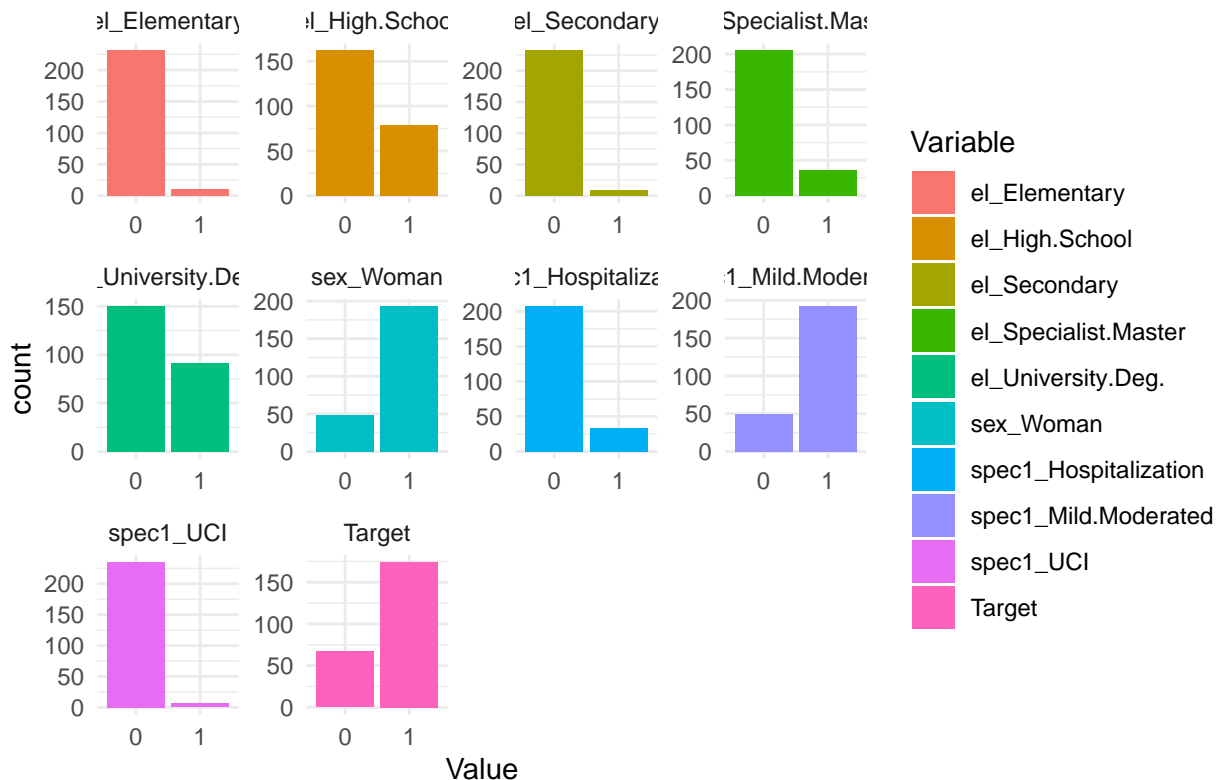
Correlación y relaciones multivariadas

```r
# ----------------------
# Correlación y relaciones multivariadas
# ----------------------

# a) Correlación entre variables numéricas
cor_matrix_def <- cor(numeric_vars_def, use = "pairwise.complete.obs")
corrplot(cor_matrix_def, method = "color", type = "full", tl.cex = 0.7, tl.srt = 45)
```
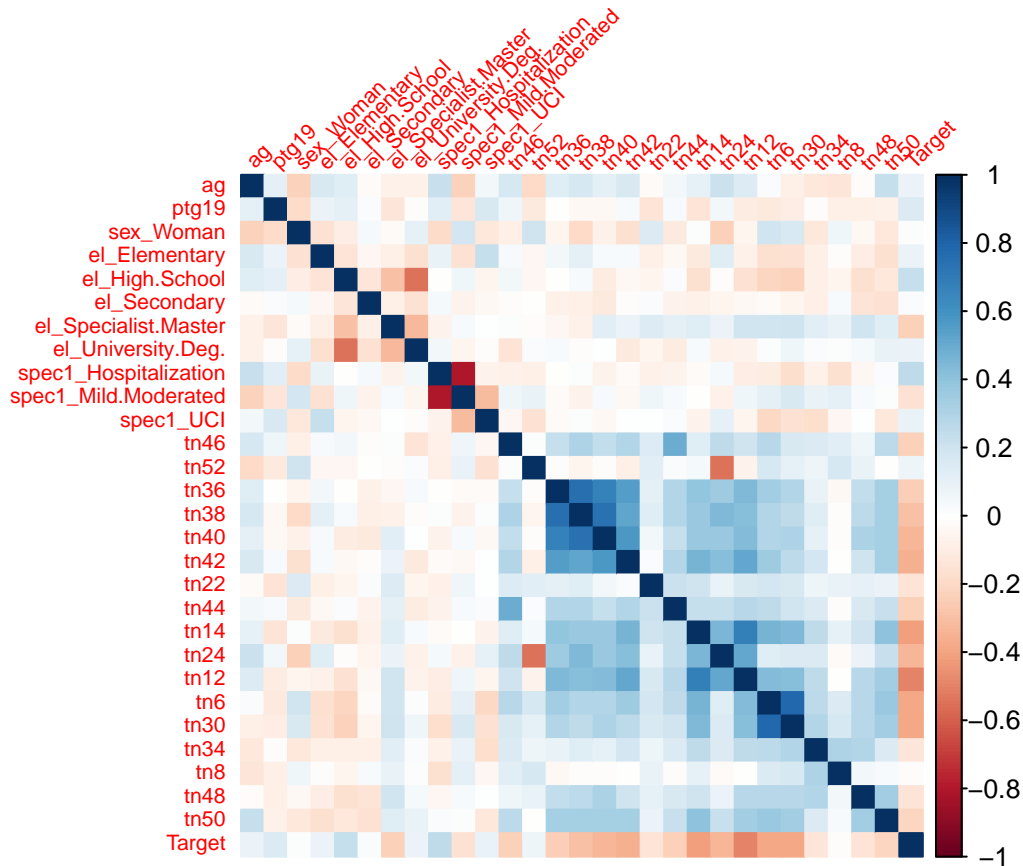
```
# Filtrar correlaciones mayores o iguales a |0.6| (sin incluir la diagonal)
threshold_def <- 0.6
high_correlations_def <- which(abs(cor_matrix_def) >= threshold_def & abs(cor_matrix_def) < 1, arr.ind =

# Crear un data.frame con las variables altamente correlacionadas
cor_df_def <- data.frame(
  Var1 = rownames(cor_matrix_def)[high_correlations_def[, 1]],
  Var2 = colnames(cor_matrix_def)[high_correlations_def[, 2]],
  Correlation = cor_matrix_def[high_correlations_def]
)

# Eliminar duplicados (porque la matriz de correlación es simétrica)
cor_df_def <- cor_df_def[!duplicated(t(apply(cor_df_def, 1, sort))), ]
#write.csv(cor_df, "./outputs/correlaciones.csv")

# Mostrar las correlaciones
print(cor_df_def)
```

```
                  Var1                Var2 Correlation
1       el_High.School      el_High.School   1.0000000
2         el_Secondary        el_Secondary   1.0000000
3  spec1_Mild.Moderated spec1_Hospitalization  -0.8022447
5                 tn52                tn52   1.0000000
6                 tn38                tn36   0.7554991
7                 tn40                tn36   0.6749011
9                 tn38                tn38   1.0000000
10                tn40                tn38   0.7465677
```

11

```
13                tn12              tn14   0.6849510
15                 tn6               tn6   1.0000000
16                tn30               tn6   0.7811306
18                tn30              tn30   1.0000000
19                tn34              tn34   1.0000000
20                tn50              tn50   1.0000000
```

Relaciones numéricas-categóricas

```r
# b) Relaciones numéricas-categóricas
# Ajustando "Target" como variable objetivo
if ("Target" %in% colnames(df_def_clean)) {
  df_def_clean %>%
    mutate(Target = as.factor(Target)) %>% #Considerando que Target es binaria
    gather(key = "Variable", value = "Value", -Target) %>%
    filter(!is.na(Value)) %>%
    ggplot(aes(x = Target, y = Value)) +
    geom_boxplot() +
    facet_wrap(~ Variable, scales = "free") +
    theme_minimal() +
    labs(title = "Relaciones entre variables y Target")
}
```


Relaciones entre variables y Target
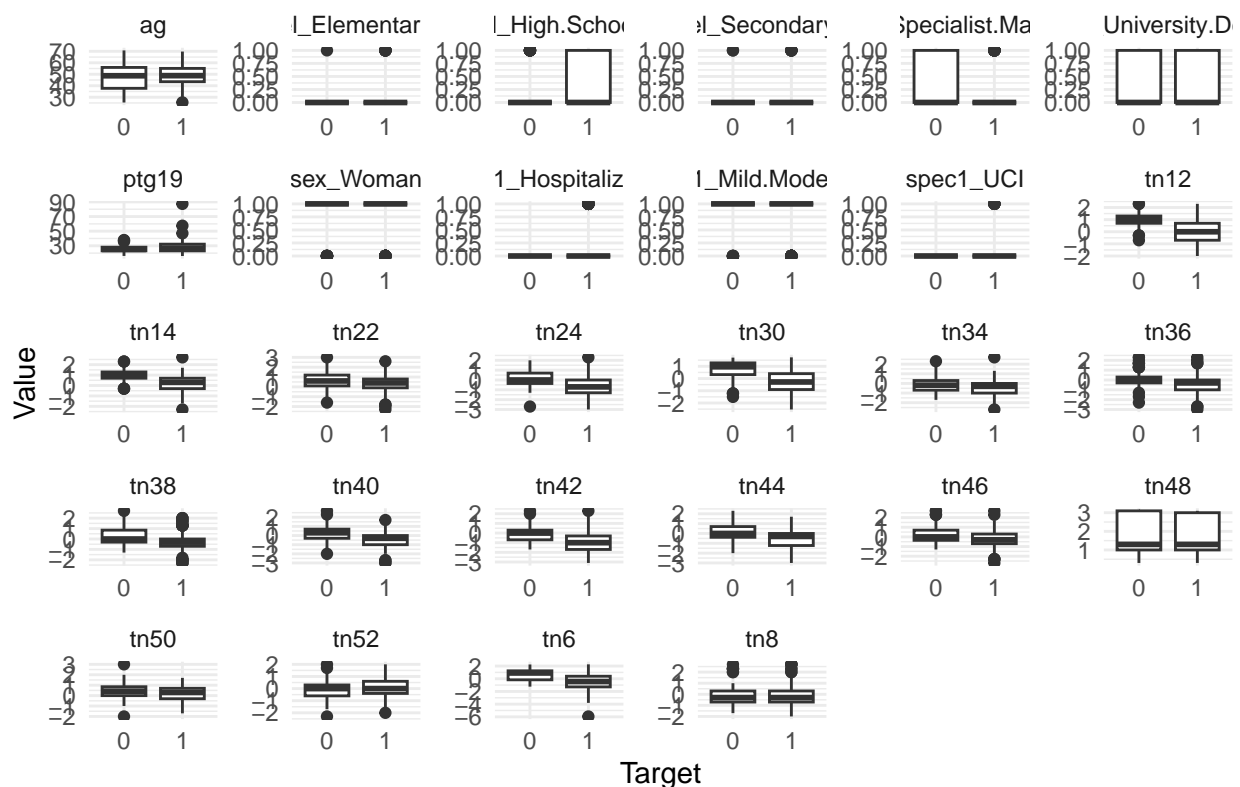
Reducción de la dimensionalidad

```r
# a) Eliminar variables con baja variabilidad: esto es "automático" con el paquete caret
nzv <- nearZeroVar(df_def_clean, saveMetrics = TRUE)
low_variability_vars <- rownames(nzv[nzv$nzv == TRUE, ])
cat("Variables con baja variabilidad eliminadas:", low_variability_vars, "\n")
```

Variables con baja variabilidad eliminadas: el_Elementary el_Secondary spec1_UCI

```r
df_def_clean_low_var <- df_def_clean %>% dplyr::select(-all_of(low_variability_vars))
```

```r
# b) Análisis de componentes principales (PCA)

pca_data_scaled <- scale(df_def_clean)

pca <- prcomp(pca_data_scaled, center = TRUE)
summary(pca)  # Proporción de la varianza explicada
```

```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     2.4283 1.7280 1.36641 1.25357 1.22288 1.15161 1.14405
Proportion of Variance 0.2033 0.1030 0.06438 0.05419 0.05157 0.04573 0.04513
Cumulative Proportion  0.2033 0.3063 0.37067 0.42486 0.47643 0.52216 0.56729
                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
Standard deviation     1.09425 1.06641 1.01097 0.98501 0.97596 0.92702 0.88838
Proportion of Variance 0.04129 0.03922 0.03524 0.03346 0.03284 0.02963 0.02721
Cumulative Proportion  0.60858 0.64780 0.68304 0.71650 0.74934 0.77898 0.80619
                          PC15    PC16    PC17    PC18    PC19    PC20    PC21
Standard deviation     0.8617 0.82799 0.78399 0.73707 0.71217 0.68631 0.64483
Proportion of Variance 0.0256 0.02364 0.02119 0.01873 0.01749 0.01624 0.01434
Cumulative Proportion  0.8318 0.85543 0.87663 0.89536 0.91285 0.92909 0.94343
                          PC22   PC23    PC24    PC25    PC26    PC27    PC28
Standard deviation     0.61088 0.5491 0.50475 0.46039 0.42703 0.39811 0.30577
Proportion of Variance 0.01287 0.0104 0.00879 0.00731 0.00629 0.00547 0.00322
Cumulative Proportion  0.95630 0.9667 0.97548 0.98279 0.98908 0.99454 0.99776
                          PC29
Standard deviation     0.25462
Proportion of Variance 0.00224
Cumulative Proportion  1.00000
```

Necesitamos unas 13 componentes para explicar el 80% de la varianza. Las 2 primeras solo explican un 50% de la varianza.

```r
# Visualzación más sencilla
fviz_pca_ind(pca,
             habillage = df_def_clean$Target,
             label = "none",
             repel = TRUE,
             title = "Gráfico de PCA (Target como variable suplementaria)")
```

Gráfico de PCA (Target como variable suplementaria)
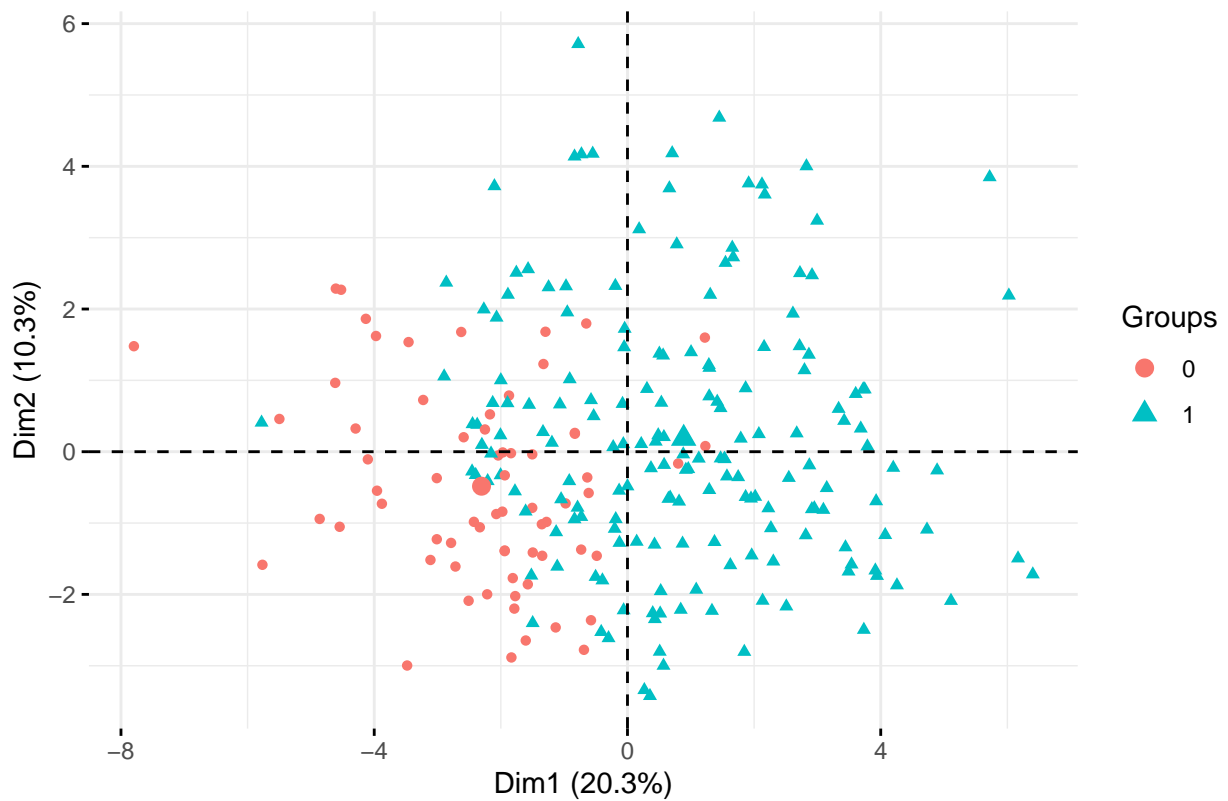

```r
# ----------------------
# Clustering para patrones
# ----------------------
# Matriz de distancias y dendrograma
dist_matrix <- dist(scale(numeric_vars_def))
hclust_model <- hclust(dist_matrix, method = "ward.D2")
plot(hclust_model, main = "Dendrograma de clustering Jerárquico")
```

Dendrograma de clustering Jerárquico
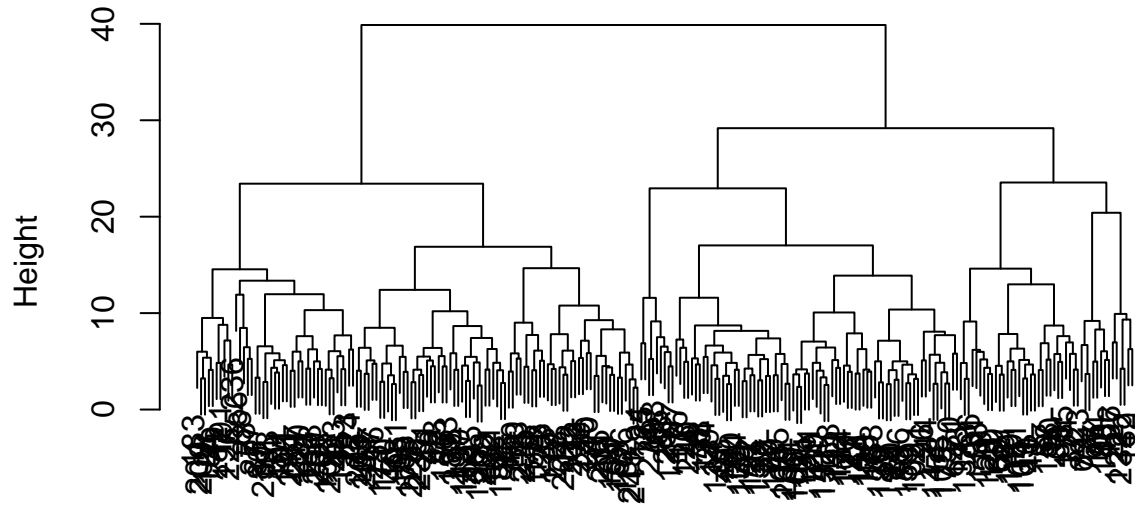
dist_matrix
hclust (*, "ward.D2")

```r
# Cortar el dendrograma en 2 clústeres
clusters <- cutree(hclust_model, k = 2)  # Cambiar "k" al número de clústeres sobre el que se cortará

# Ver asignaciones de clústeres para las observaciones
print(clusters)
```

```
  [1] 1 1 2 1 1 2 2 2 2 1 2 1 2 1 2 2 1 1 2 1 1 2 1 2 1 2 2 2 1 1 1 1 2 1 1 2 1
 [38] 1 1 2 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 1
 [75] 2 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 2 1 1 1 1 2 1 1 1
[112] 1 2 1 2 1 2 2 1 1 1 1 1 1 2 1 1 1 2 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1
[149] 1 2 2 1 2 1 2 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 2 2 1 1 1 1 2 1 2 2 2 2 2 2
[186] 2 2 1 2 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2
[223] 2 2 2 2 1 2 2 2 2 1 1 2 2 2 2 2 2 1 2
```

Verificar como funciona el clúster

```r
# Asociar observaciones con cada clúster
cluster_observations <- split(row.names(numeric_vars_def), clusters)

# Imprimir observaciones de cada clúster
print("Observaciones en el clúster 1:")
```

```
[1] "Observaciones en el clúster 1:"
```

```r
print(cluster_observations[[1]])
```

```
 [1] "1"  "2"  "4"  "5"  "10" "12" "14" "17" "18" "20" "21" "23"
[13] "25" "29" "30" "31" "32" "34" "35" "37" "38" "39" "41" "42"
[25] "43" "44" "58" "59" "60" "61" "62" "63" "64" "66" "67" "69"
[37] "70" "71" "72" "73" "74" "76" "77" "78" "79" "81" "82" "83"
```

```
[49] "84"  "86"  "87"  "88"  "89"  "90"  "91"  "92"  "93"  "94"  "96"  "97"
[61] "98"  "99"  "100" "102" "104" "105" "106" "107" "109" "110" "111" "112"
[73] "114" "116" "119" "120" "121" "122" "123" "125" "126" "127" "129" "130"
[85] "134" "135" "136" "137" "138" "139" "140" "141" "142" "143" "144" "146"
[97] "147" "148" "149" "152" "154" "156" "157" "158" "159" "160" "161" "162"
[109] "164" "165" "166" "168" "169" "170" "174" "175" "176" "177" "179" "188"
[121] "194" "198" "221" "227" "232" "233" "240"
```

```r
print("Observaciones en el clúster 2:")
```

```
[1] "Observaciones en el clúster 2:"
```

```r
print(cluster_observations[[2]])
```

```
  [1] "3"   "6"   "7"   "8"   "9"   "11"  "13"  "15"  "16"  "19"  "22"  "24"
[13] "26"  "27"  "28"  "33"  "36"  "40"  "45"  "46"  "47"  "48"  "49"  "50"
[25] "51"  "52"  "53"  "54"  "55"  "56"  "57"  "65"  "68"  "75"  "80"  "85"
[37] "95"  "101" "103" "108" "113" "115" "117" "118" "124" "128" "131" "132"
[49] "133" "145" "150" "151" "153" "155" "163" "167" "171" "172" "173" "178"
[61] "180" "181" "182" "183" "184" "185" "186" "187" "189" "190" "191" "192"
[73] "193" "195" "196" "197" "199" "200" "201" "202" "203" "204" "205" "206"
[85] "207" "208" "209" "210" "211" "212" "213" "214" "215" "216" "217" "218"
[97] "219" "220" "222" "223" "224" "225" "226" "228" "229" "230" "231" "234"
[109] "235" "236" "237" "238" "239" "241"
```

```r
# Filtrar datos por clústeres si se desea trabajar con subconjuntos
cluster_1 <- numeric_vars_def[clusters == 1, ]
cluster_2 <- numeric_vars_def[clusters == 2, ]

# Mostrar los subconjuntos
print("Datos del clúster 1:")
```

```
[1] "Datos del clúster 1:"
```

```r
print(cluster_1)
```

```
     ag  ptg19 sex_Woman el_Elementary el_High.School el_Secondary
1   26.9 38.23         1             0              1            0
2   60.5 26.34         0             0              0            0
4   44.6 24.14         1             0              0            0
5   61.9 32.80         0             0              0            0
10  51.2 28.66         1             0              0            0
12  36.4 28.45         1             0              1            0
14  48.8 30.98         1             0              0            0
17  54.7 35.75         1             0              0            0
18  47.5 25.44         1             0              1            0
20  46.3 33.01         1             0              1            0
21  33.5 25.10         1             0              0            0
23  53.9 26.57         0             1              0            0
25  55.4 22.68         1             0              0            1
29  39.7 28.13         1             0              0            0
30  46.1 26.41         1             0              0            0
31  63.0 26.47         0             0              0            0
32  42.6 22.21         1             0              0            0
34  63.0 19.70         1             0              0            0
35  39.2 27.71         1             0              1            0
37  59.7 22.10         1             0              1            0
```

16

38	43.0	26.25	0	0	0	0
39	44.7	22.62	1	0	1	0
41	50.4	39.94	1	0	0	0
42	56.6	25.80	1	0	0	0
43	46.7	21.95	1	0	0	0
44	48.8	24.01	1	0	1	0
58	47.0	21.29	1	0	1	0
59	51.4	23.05	0	0	1	0
60	67.2	32.21	1	0	1	0
61	53.8	25.71	1	0	1	0
62	55.3	28.67	0	0	1	0
63	57.8	39.08	1	0	0	0
64	57.9	25.44	1	1	0	0
66	56.7	29.97	0	0	1	0
67	40.3	26.89	1	0	0	0
69	65.7	36.88	1	1	0	0
70	43.6	17.62	1	0	0	0
71	48.5	32.25	0	0	0	0
72	41.8	23.07	1	0	1	0
73	69.8	33.24	0	1	0	0
74	54.4	32.81	1	0	1	0
76	48.1	28.52	1	0	0	1
77	45.5	28.19	1	1	0	0
78	48.3	34.36	1	0	0	0
79	53.7	21.85	1	0	0	0
81	61.8	27.21	1	0	0	0
82	54.9	37.80	1	1	0	0
83	49.8	23.65	1	0	1	0
84	50.3	29.34	1	0	1	0
86	42.0	25.64	1	0	0	1
87	49.5	31.92	1	0	0	0
88	51.4	30.61	1	0	0	1
89	55.6	22.83	1	0	1	0
90	55.6	35.77	0	0	0	0
91	44.8	21.33	1	0	1	0
92	50.2	40.32	1	0	0	0
93	43.3	16.05	1	0	0	0
94	37.9	33.67	1	0	0	1
96	26.4	42.40	1	0	1	0
97	56.4	26.71	0	0	1	0
98	53.3	33.21	0	0	1	0
99	62.6	36.92	0	1	0	0
100	39.8	20.56	1	0	1	0
102	48.2	24.35	1	0	1	0
104	47.2	17.38	1	0	0	0
105	41.5	22.81	1	0	1	0
106	49.1	29.40	0	0	0	0
107	42.2	57.46	1	0	0	0
109	47.5	29.26	1	0	1	0
110	56.9	24.44	0	0	1	0
111	54.3	33.45	1	0	1	0
112	40.5	25.64	0	0	1	0
114	48.6	38.29	1	0	0	0
116	37.9	41.27	0	0	1	0

```
119 56.5 27.24         0              1              0              0
120 53.2 34.17         1              0              0              0
121 50.7 27.21         1              0              0              1
122 61.3 30.59         0              0              0              0
123 55.4 25.75         0              0              1              0
125 39.1 33.57         0              0              0              1
126 46.0 44.23         0              0              1              0
127 47.6 33.87         1              0              0              0
129 59.9 28.93         1              0              1              0
130 40.0 20.00         1              0              0              0
134 59.6 37.97         1              0              1              0
135 44.6 20.90         1              0              0              0
136 48.4 20.62         1              0              1              0
137 52.4 36.31         1              0              0              0
138 48.8 30.21         0              0              1              0
139 48.4 22.31         1              0              1              0
140 48.4 27.97         1              0              1              0
141 51.7 17.94         1              0              1              0
142 53.9 21.15         1              0              1              0
143 45.2 24.16         0              0              1              0
144 41.9 28.95         1              0              0              0
146 50.6 22.32         1              1              0              0
147 37.0 25.30         1              0              1              0
148 57.6 32.78         1              0              0              0
149 57.7 37.72         1              0              0              0
152 54.5 23.84         1              0              1              0
154 49.0 28.76         1              0              0              0
156 50.6 22.50         1              0              1              0
157 43.2 29.30         1              0              0              0
158 57.2 29.34         1              0              0              0
159 54.2 22.82         1              0              1              0
160 52.3 30.35         1              0              0              0
161 39.8 22.34         1              0              0              0
162 54.9 19.54         1              0              1              0
164 44.3 30.51         0              1              0              0
165 40.6 27.79         0              0              0              0
166 54.8 29.79         1              0              0              0
168 47.3 36.06         1              0              1              0
169 50.2 24.01         1              0              0              0
170 38.2 19.14         1              0              0              0
174 44.2 28.28         1              0              0              0
175 39.5 28.26         1              0              1              0
176 46.9 22.47         1              0              0              0
177 35.2 26.40         1              0              1              0
179 53.1 21.21         1              0              0              0
188 31.7 23.18         1              0              1              0
194 43.5 19.43         0              0              0              0
198 67.5 21.88         1              0              0              1
221 34.5 33.24         1              0              0              1
227 35.8 47.02         1              0              1              0
232 54.5 39.36         1              0              0              0
233 48.6 35.23         1              0              0              0
240 51.1 23.83         0              0              0              0
    el_Specialist.Master el_University.Deg. spec1_Hospitalization
```

1	0	0	0
2	1	0	1
4	0	1	0
5	1	0	1
10	0	1	1
12	0	0	0
14	0	1	0
17	0	1	0
18	0	0	0
20	0	0	0
21	0	1	0
23	0	0	1
25	0	0	0
29	0	1	0
30	1	0	0
31	0	1	1
32	0	1	0
34	0	1	1
35	0	0	1
37	0	0	1
38	0	1	0
39	0	0	0
41	0	1	1
42	0	1	0
43	0	1	0
44	0	0	0
58	0	0	0
59	0	0	1
60	0	0	1
61	0	0	0
62	0	0	1
63	0	1	1
64	0	0	0
66	0	0	1
67	0	1	0
69	0	0	1
70	0	1	0
71	0	1	0
72	0	0	1
73	0	0	1
74	0	0	0
76	0	0	0
77	0	0	0
78	0	1	0
79	0	1	0
81	0	1	1
82	0	0	0
83	0	0	0
84	0	0	0
86	0	0	0
87	0	1	1
88	0	0	1
89	0	0	0
90	0	1	1

91	0	0	0
92	0	1	0
93	0	1	0
94	0	0	0
96	0	0	0
97	0	0	0
98	0	0	1
99	0	0	0
100	0	0	0
102	0	0	0
104	0	1	1
105	0	0	0
106	1	0	0
107	0	1	1
109	0	0	0
110	0	0	1
111	0	0	0
112	0	0	0
114	0	1	0
116	0	0	0
119	0	0	0
120	0	1	0
121	0	0	0
122	0	1	0
123	0	0	1
125	0	0	1
126	0	0	0
127	0	1	0
129	0	0	0
130	0	1	0
134	0	0	0
135	0	1	0
136	0	0	0
137	0	1	0
138	0	0	0
139	0	0	0
140	0	0	1
141	0	0	0
142	0	0	0
143	0	0	0
144	0	1	0
146	0	0	0
147	0	0	0
148	0	1	1
149	0	1	1
152	0	0	0
154	0	1	0
156	0	0	0
157	0	1	0
158	1	0	1
159	0	0	0
160	0	1	0
161	0	1	1
162	0	0	0

164	0	0	0
165	0	0	0
166	0	1	0
168	0	0	0
169	0	1	1
170	0	1	0
174	0	1	1
175	0	0	0
176	0	1	0
177	0	0	0
179	0	1	0
188	0	0	0
194	0	1	0
198	0	0	0
221	0	0	0
227	0	0	0
232	0	1	0
233	0	1	0
240	0	1	0

	spec1_Mild.Moderated	spec1_UCI	tn46	tn52	tn36	tn38	tn40	tn42	tn22	tn44
1	1	0	0.3	1.0	-1.00	-0.3	0.0	-1.3	0.4	-0.7
2	0	0	1.0	0.0	0.00	0.3	0.0	0.0	0.4	-1.3
4	1	0	0.0	-0.6	0.30	0.0	1.0	1.3	-0.9	1.0
5	0	0	0.0	0.0	0.30	-0.3	0.3	-0.3	-0.2	-0.3
10	0	0	-0.3	0.7	0.00	-0.3	-0.3	0.3	0.1	-0.7
12	1	0	-0.7	0.7	-2.00	-1.0	-1.0	-1.7	0.1	0.0
14	1	0	-0.7	-0.3	-0.30	-0.3	-1.0	-0.7	1.3	-0.7
17	1	0	-0.7	-2.0	-2.00	-1.3	-1.3	-1.0	-0.5	-2.0
18	1	0	-0.7	0.0	1.30	2.0	0.3	-1.3	-0.9	-0.7
20	1	0	-1.0	0.0	0.00	-0.3	-0.3	-0.3	0.2	0.0
21	1	0	0.3	1.0	-0.70	-0.3	-0.7	-2.7	-0.1	0.0
23	0	0	-2.0	0.0	0.30	1.0	1.3	0.3	0.2	-1.3
25	1	0	0.7	0.0	-0.30	-0.3	-0.3	-0.7	0.8	-0.7
29	1	0	-0.7	-1.3	0.70	-0.7	-0.7	-2.0	0.3	1.0
30	1	0	0.7	0.0	0.70	-0.7	0.3	-2.0	1.2	1.7
31	0	0	-0.7	-1.0	-0.70	0.0	-0.7	-0.7	-0.2	1.0
32	1	0	0.3	-0.3	1.70	-0.3	-0.7	-0.7	-0.2	-2.0
34	0	0	0.0	-1.7	-0.30	0.3	1.0	-0.3	0.7	-0.3
35	0	0	1.3	0.0	0.70	1.7	0.0	0.3	-1.2	0.0
37	0	0	0.0	-0.7	1.30	0.3	0.0	0.0	0.2	0.0
38	0	0	0.0	0.7	0.30	-0.3	0.0	-0.3	-1.3	-1.3
39	1	0	0.3	0.6	0.00	0.0	-0.3	-1.0	1.4	-0.7
41	0	0	-0.3	0.4	-0.30	-0.3	-0.3	-1.0	0.5	-0.7
42	1	0	-0.3	0.4	-1.70	-1.0	-1.0	-2.0	0.5	-0.3
43	1	0	-1.0	1.4	-1.00	-1.0	-1.0	-2.7	-0.6	-0.7
44	1	0	0.3	1.0	0.00	-0.3	-1.3	-0.3	1.4	0.0
58	1	0	0.3	0.0	-1.00	-0.7	-1.3	-2.7	1.6	0.0
59	0	0	0.0	-0.7	-0.30	-0.7	0.0	0.0	-0.1	0.3
60	0	0	0.3	0.0	-0.70	-0.7	-1.3	-1.7	0.7	-0.7
61	1	0	-0.7	0.6	-2.30	-2.0	-2.0	-2.3	0.7	-2.0
62	0	0	0.0	0.4	0.00	0.3	-0.3	-1.7	0.6	-0.3
63	0	0	-0.3	-0.6	1.00	0.7	0.0	0.0	-0.2	0.3
64	1	0	0.0	0.0	-0.70	-0.7	-0.3	-1.0	-0.6	-2.7
66	0	0	-0.3	1.0	0.00	1.3	1.0	0.3	-0.1	1.0

```
67                      1           0 -1.0  1.3  0.30  0.0 -0.7 -1.0  0.5 -0.7
69                      0           0  0.3 -1.3 -0.70 -0.3 -0.7 -2.0  0.5 -1.0
70                      1           0 -1.0 -1.0  1.30  0.3  0.0 -0.3 -1.8 -2.0
71                      1           0  0.3  0.3 -0.70 -0.7 -0.7 -2.3  2.6  0.0
72                      0           0 -0.7 -0.7  0.70  0.3 -0.7 -0.7  0.4  0.0
73                      0           0  0.7 -0.4 -1.00  0.0 -1.0 -1.3  0.7 -0.7
74                      1           0  0.0 -0.3 -1.70 -1.0 -1.0 -0.7 -0.1  0.0
76                      1           0 -1.0 -0.3 -1.00 -0.7 -1.0 -2.3  1.3 -2.0
77                      1           0 -0.7  1.0  0.00  0.7  0.7 -1.0  1.0 -1.7
78                      1           0  0.3 -0.3 -0.30 -0.7 -1.0  0.3 -2.3 -0.7
79                      1           0  0.0 -0.3 -1.00 -0.7 -0.7 -0.7  0.0 -0.7
81                      0           0 -0.7  0.4 -0.70 -0.7 -0.3 -2.3  0.5 -3.0
82                      0           1  0.0  0.0  0.00 -1.0 -1.3 -0.3 -0.8 -0.3
83                      1           0  0.3  0.0 -0.70 -0.3  0.0 -1.7  0.3 -0.7
84                      1           0 -0.7  0.7 -0.30 -1.0 -0.7 -0.3 -0.1 -0.7
86                      1           0 -0.7  2.0  0.00 -0.7 -0.7 -0.3  0.7 -0.7
87                      0           0  1.7  0.7  0.70 -0.3 -0.7 -1.3  0.3  1.7
88                      0           0  0.0  0.3  0.00  0.7  0.3  0.7  0.5 -0.3
89                      1           0  0.0  0.7 -0.30  0.0  0.0 -0.3  0.6 -0.3
90                      0           0 -2.7 -0.4 -0.30  0.3  0.0 -0.7  0.9 -2.0
91                      1           0 -2.3  0.7 -0.30 -1.0 -2.0 -1.0  0.5  0.0
92                      1           0 -0.3 -0.6  0.00  0.0 -0.3 -1.0 -0.5 -2.0
93                      1           0 -1.0 -0.3 -1.70 -1.0 -1.0 -2.7  0.4 -0.7
94                      1           0 -1.0  0.6 -2.30 -2.3 -3.0 -2.7  0.3  0.7
96                      1           0  0.7  0.4 -0.70 -1.0 -1.7 -2.0  0.5  0.0
97                      1           0  0.0 -0.7  0.00  0.0  0.0  0.0 -0.9 -0.3
98                      0           0  0.0 -1.0 -1.00 -0.7 -1.7  0.0 -0.5  0.3
99                      1           0  0.7  0.0  1.00  1.3  1.0 -0.3 -1.1  0.3
100                     1           0 -2.3  0.3 -2.30 -1.8 -2.7 -1.0  0.0 -0.7
102                     1           0 -1.0  0.7  0.00  0.0  0.0 -2.0  1.3  0.0
104                     0           0 -0.7  0.3 -0.30 -0.3 -1.0 -1.3  1.0 -0.7
105                     1           0  0.3  0.0 -1.30 -1.0 -1.0 -2.0 -0.9 -2.0
106                     0           1 -1.3 -1.0 -1.07 -1.0 -1.3 -1.3  1.1 -1.0
107                     0           0 -1.3 -0.7 -1.30 -1.0 -1.0 -2.0 -0.4  0.0
109                     1           0  0.3  0.0 -1.00 -0.7 -1.3 -1.7  0.6 -0.7
110                     0           0 -0.3  0.0  0.00 -0.7 -0.3 -0.3 -1.0 -2.0
111                     1           0  1.0 -0.7 -1.70 -1.0 -1.7 -2.0  0.5  0.3
112                     1           0 -1.0  0.3 -1.00 -0.7 -1.0 -1.0 -0.1 -2.0
114                     1           0 -1.3  0.3 -0.30 -2.0 -0.3 -2.3 -0.9 -2.0
116                     1           0 -0.7 -1.3  0.00 -0.3 -0.7 -1.3 -2.0  0.0
119                     0           1  0.3 -1.3  0.30  0.7  0.0  0.3  1.8  0.3
120                     1           0 -0.7 -0.3 -1.30 -1.0 -1.0 -3.1 -0.6 -2.0
121                     1           0  2.7  0.6  0.30  0.0  0.0  0.3  1.8  1.3
122                     0           1  0.0 -0.6  0.30  1.7  0.3 -1.0  1.0  0.3
123                     0           0 -0.3  0.3 -0.70 -0.3 -0.3 -1.3  0.6 -2.0
125                     0           0 -1.0 -0.7 -1.30 -1.7 -2.0 -2.3 -1.2 -2.0
126                     1           0 -1.0 -1.0 -1.70 -1.7 -1.7 -1.7 -0.3 -2.0
127                     1           0 -0.7  0.3 -0.30 -0.7  0.0 -0.3  0.7 -0.7
129                     1           0 -0.3  0.4 -0.70 -1.0 -0.7 -2.0 -0.6  0.0
130                     1           0 -1.0  0.0 -0.70  0.0  0.0 -0.3 -0.9 -2.0
134                     1           0  0.0  0.3 -1.70 -1.0 -1.7 -1.3  0.0 -1.3
135                     1           0 -0.7  0.3  0.70 -0.3  0.0 -1.0  0.6 -2.0
136                     1           0 -0.7 -0.6 -0.30 -0.3 -1.0 -1.0  0.7 -2.0
137                     1           0 -0.7  0.0  0.30  0.3 -0.3 -1.0  0.4 -0.7
```

```
138                      1          0 -1.0 -1.4 -0.30 -0.3 -0.3 -0.3 -0.8 -2.0
139                      1          0 -1.0 -0.4  0.70  1.3  0.3 -1.3  0.2  0.0
140                      0          0 -0.7  0.0 -1.70 -0.7 -1.0 -2.3  0.7 -2.0
141                      1          0 -0.3 -0.4 -2.00 -1.0 -1.3 -2.3  0.6  0.0
142                      1          0  0.0  1.0 -0.70 -1.0 -0.3  0.3  0.7  0.3
143                      1          0 -0.7 -0.7  0.00  0.3  0.0 -1.0  0.3  0.0
144                      1          0 -1.0  0.0 -2.70 -2.3 -3.0 -1.0 -0.8 -2.0
146                      1          0  0.3 -0.4  1.30  1.7  0.3  1.0 -0.3  1.7
147                      1          0 -0.7  0.3 -0.30 -0.7 -1.7 -1.3  0.4 -0.7
148                      0          0 -0.7 -1.0 -0.70 -1.3 -0.7 -0.7 -0.5 -0.3
149                      0          0 -0.7  0.0 -1.30 -1.3 -2.0 -2.3 -0.1 -1.7
152                      1          0 -0.3  0.0 -1.70 -1.0 -0.7 -2.0  0.2  0.3
154                      1          0 -1.0  0.0 -2.70 -1.3 -1.0 -2.7  0.0 -2.0
156                      1          0 -0.3  0.3 -1.30 -1.3 -1.3 -2.3  0.8 -0.7
157                      1          0 -1.0 -0.4 -0.30 -0.7  0.0 -1.0 -0.8 -2.0
158                      0          0  1.0  0.7 -0.30  0.3 -0.3 -0.3  1.7  1.3
159                      1          0  0.0 -0.4  0.00  0.7  0.0 -1.0  1.2  0.3
160                      1          0 -0.7 -1.6 -1.30 -0.7 -0.7 -2.0  0.5 -0.7
161                      0          0  0.3  0.7  0.00 -0.3  0.0  0.7  0.7 -0.7
162                      1          0 -0.3  0.0 -1.00 -0.7 -0.3 -1.7  0.3 -0.7
164                      1          0  0.7  1.0 -0.70  0.0 -1.0 -1.3 -0.9  0.3
165                      1          0 -1.0  1.3 -0.70 -0.3  0.0 -1.3 -0.3 -1.0
166                      1          0 -0.3  0.6 -0.70 -0.3 -0.3 -0.7  0.0 -2.0
168                      1          0 -0.7  0.7 -1.30 -1.0 -0.7 -1.7  0.8 -2.0
169                      0          0 -0.7  0.4 -0.70 -0.7 -0.7 -1.3  2.1  0.0
170                      1          0  0.0  1.7  1.00  0.0  0.0 -1.0  1.5  0.0
174                      0          0 -1.0 -0.3  0.70  1.3  0.3 -1.3 -0.1  0.0
175                      1          0  0.3  0.3 -0.30 -0.3 -0.3 -2.0  0.7  0.0
176                      1          0  0.7  0.7 -1.30 -1.7 -1.0 -2.0  0.4 -2.0
177                      1          0 -1.0  0.0 -2.30 -2.0 -2.7 -2.3  0.3 -2.0
179                      1          0 -0.3 -0.7  0.00  0.0  0.0 -2.3  0.3 -2.0
188                      1          0 -1.0  0.0  0.00  0.3  0.3 -0.3 -1.5  0.0
194                      1          0 -0.7 -0.4 -0.30  0.0 -1.0 -1.7 -0.2 -0.7
198                      0          0 -0.3 -1.4  0.00  0.3  0.3  0.3 -1.1 -0.7
221                      1          0 -1.0 -1.0 -1.00  0.0 -0.3 -0.3  1.4 -2.0
227                      0          1 -0.7 -1.0 -1.70 -1.0 -1.3 -2.0 -1.4 -0.7
232                      0          1 -0.3 -0.4 -0.30  0.3  1.0  0.3  1.0  0.0
233                      1          0 -0.7  1.0 -0.70 -0.7 -1.0 -1.0  0.0 -2.0
240                      1          0  0.7  0.0 -3.00 -1.0 -0.3  0.3 -1.8  0.0
```

	tn14	tn24	tn12	tn6	tn30	tn34	tn8	tn48	tn50	Target
1	-1.7	-3.0	-2.0	-2.06	-2.30	-1.0	2.0	1.0	-1.7	1
2	1.0	0.0	0.0	0.86	0.70	0.3	-0.3	3.1	1.3	1
4	-0.3	-0.7	-0.3	-0.82	-0.43	-1.0	-0.7	1.3	-0.3	1
5	1.3	-0.3	1.3	-0.57	0.40	0.3	-0.3	3.1	-0.7	1
10	0.7	-0.7	0.0	-0.32	0.03	-2.0	0.3	0.8	0.7	1
12	0.3	-1.0	-0.3	-1.03	-0.03	0.0	-0.3	3.1	-0.7	1
14	0.7	-1.7	-0.7	-2.22	-1.14	-1.0	0.3	1.3	0.7	1
17	-0.3	0.7	0.0	0.04	-0.28	-0.3	0.0	0.7	-1.0	1
18	-0.3	-0.3	0.0	0.01	-0.42	-0.7	-0.3	1.0	-0.7	1
20	-0.3	-0.3	0.0	-1.52	-1.14	-0.7	-0.3	1.3	0.0	1
21	-0.7	-2.0	-1.3	-2.36	-1.46	0.0	0.3	0.8	-0.7	1
23	0.0	0.0	0.0	-1.80	-0.59	0.3	0.3	1.3	-1.3	1
25	1.0	-0.3	1.0	-0.93	-0.90	-0.7	-0.3	0.5	0.7	1
29	0.7	-1.0	0.3	-1.39	-0.75	-1.3	-0.3	1.0	-0.7	1

```
30  -0.7 -0.7 -0.7 -1.87 -0.78  0.0  0.3  1.3  0.0      1
31   0.7 -0.7  0.7 -0.96 -1.60 -0.7 -2.0  0.7  1.3      1
32   0.3 -0.7  0.7  0.24 -0.42 -0.3 -0.3  1.3  0.7      1
34   1.0  1.0  1.3  0.07  0.06  0.0 -0.3  0.8  0.7      1
35   1.3  1.0  1.0  0.98  0.40  1.3  0.7  0.3  0.3      1
37   1.0  0.0  1.0  0.79  0.03  0.0  0.3  3.0  0.7      1
38   0.7 -1.0  0.7 -1.48 -1.60 -0.3 -0.3  1.3 -0.3      1
39   0.7 -0.3  1.0  0.22 -0.78 -0.3 -0.3  1.0  0.0      1
41  -0.3 -0.7 -0.3  0.04  0.03 -0.3 -0.3  1.3  0.0      1
42  -0.3 -1.7 -0.3 -0.93 -0.90 -0.3  0.0  1.0 -0.3      1
43  -0.3 -2.7 -1.3 -1.29 -1.50  0.0  2.3  1.0  0.0      1
44  -0.3 -2.3 -1.0  0.80  1.35  1.3  0.7  3.0 -1.0      1
58  -1.0 -1.0 -0.7 -0.59 -0.07 -0.3  0.7  3.1  0.3      1
59  -0.3 -0.3 -1.0 -2.91 -1.21 -0.3 -1.3  1.3  0.0      1
60  -0.3 -0.3 -1.0 -0.31 -0.93 -1.7 -0.3  0.3  0.3      1
61   0.3 -1.3 -0.7 -1.18 -0.90 -0.7 -0.3  0.5  0.5      1
62   0.0 -0.7 -1.3 -0.69 -0.90 -1.3  0.0  0.8  0.7      1
63   1.3  0.3  2.0  0.42  0.65  0.3 -0.7  3.1  1.7      1
64   0.0 -1.0 -0.3 -0.93 -0.59 -2.3 -0.7  0.5 -0.3      1
66   1.3  0.7  1.0 -0.36 -1.86 -1.0 -1.0  1.0 -0.7      1
67  -0.7 -2.3 -0.3 -0.71 -0.43 -1.0 -0.3  3.0  0.0      1
69  -0.3  0.0 -1.0 -1.22 -0.93 -1.0  0.0  0.7 -0.7      1
70  -0.3 -1.0  0.3 -0.01 -0.07 -1.0  0.7  0.8  0.3      1
71  -1.7 -1.0 -0.7 -0.24 -0.07 -0.7  0.7  0.9  0.3      1
72   0.3 -1.0  0.0  0.22  1.00 -1.3  0.0  1.3  0.0      1
73   0.0  0.7 -0.3 -1.22 -0.93 -0.3 -0.3  0.7  0.3      1
74  -0.3 -2.0 -2.0 -1.43 -0.59 -1.7 -1.0  0.7  0.3      1
76  -0.3 -1.0 -0.7  0.33 -0.42 -0.7  0.7  0.7 -1.7      1
77  -0.3 -1.7  0.0 -0.24  0.28 -1.3 -0.7  3.1  0.3      1
78  -0.7  0.0 -0.7  1.03  1.35 -1.0 -0.7  1.0  0.3      1
79  -0.3 -0.7  0.7 -0.44 -0.59  0.0 -0.3  1.0 -0.3      1
81   0.3 -0.7  0.0 -1.22 -0.60 -0.7  0.3  3.1  0.7      1
82  -0.3  0.0 -0.3 -0.32  0.03 -0.7 -0.3  1.0 -1.0      1
83   0.0 -0.7 -0.3  0.33 -0.07 -2.0 -0.3  0.8  0.0      1
84   0.3 -0.7 -0.3  0.54  0.03 -0.3  0.3  1.3  0.7      1
86   0.7 -2.7  0.3  1.50  1.35  0.0  0.7  1.0 -0.3      1
87   0.3  0.3  0.0  1.84  1.00  0.0  0.3  1.3  0.3      1
88   0.7 -0.3 -0.3 -0.44 -0.59 -1.0  1.7  1.3  0.3      1
89   0.0 -1.7 -0.3 -0.32 -1.22 -0.3  0.3  1.3  0.0      1
90   1.3 -0.3  1.7  0.54  0.34  0.0 -0.3  3.1  1.3      1
91  -0.3 -2.0 -0.3 -1.05 -0.78 -1.7 -1.0  1.3  0.3      1
92  -0.3  0.3 -0.3  1.03  0.34 -0.7 -0.3  0.8  0.8      1
93   0.0 -1.0 -0.3 -0.59  0.28  0.7  0.0  1.3 -0.7      1
94   0.0 -2.3 -0.7 -1.27 -0.75 -0.3 -1.0  0.7 -0.7      1
96  -1.0 -2.7 -2.0 -0.70  0.30 -0.3  0.3  1.0  0.0      1
97   1.0  0.7  0.3 -1.80 -1.21  0.3 -0.3  1.0  0.7      1
98  -0.7 -0.7  2.0 -1.80 -1.53 -0.3 -1.0  1.0 -0.7      1
99  -1.0  0.7  1.0 -2.77 -0.93  0.0  0.0  1.0  0.3      1
100  0.0 -1.3 -0.7 -1.03 -0.39  0.0  0.3  0.6 -0.7      1
102  0.3  0.0 -0.3 -1.40 -0.42 -1.0 -0.3  0.8 -0.7      1
104  0.0  0.0 -0.7 -1.75 -1.85 -1.0 -1.0  1.3  0.7      1
105  0.3 -1.0 -0.3 -2.33 -0.78  0.3  0.7  0.8  0.0      1
106 -0.3 -1.0  0.0 -1.40 -0.78 -0.7 -1.3  1.0 -0.7      1
107 -0.3 -1.0 -0.3 -0.36  1.00  0.0 -1.3  1.0 -0.7      1
```

```
109 -0.3 -1.0 -1.3 -0.36  0.28 -0.7  0.3  1.3  0.3      1
110  1.0 -0.7  0.7 -1.18 -0.59  0.3 -0.7  1.0  0.7      1
111 -1.3 -0.3 -0.3 -2.66 -1.84 -0.3 -0.3  1.3 -0.7      1
112  0.0 -1.0 -1.3 -1.63 -1.14 -1.3 -1.0  1.0  0.0      1
114  1.0 -1.0  0.3 -1.52 -1.14 -0.3 -0.7  1.0 -1.3      1
116 -2.3  0.0 -0.7 -0.67 -0.03 -0.7 -1.0  1.3 -0.3      1
119  1.0  2.3  0.0 -1.30 -0.90 -1.0  0.3  1.3  0.0      1
120 -1.3 -2.0 -2.0 -1.06 -0.59 -1.7 -1.7  0.7  0.0      1
121  0.3 -0.3  1.0 -0.07  0.34 -0.7  1.3  1.3  0.7      1
122  0.7  0.3  0.7 -1.48 -0.93 -1.0 -0.3  3.1  0.3      1
123  0.3 -1.0 -0.3 -2.91 -1.84 -1.0 -1.0  0.8 -0.3      1
125 -1.3 -2.0 -2.0 -1.63 -1.82 -1.3 -1.7  1.0  0.3      1
126 -0.3 -0.7 -0.3 -1.17  0.28  0.3  0.7  0.5 -1.0      1
127  0.7 -0.3 -1.0 -0.12  0.64 -0.3 -0.7  0.8  0.0      1
129  0.0 -1.7 -0.7 -2.54 -2.15 -2.7  0.0  0.7  0.0      1
130 -0.3 -1.0 -1.0 -1.63 -1.46  0.0 -0.3  3.0 -0.3      1
134  1.0  0.0  0.3 -1.80 -0.90  0.0  0.3  1.0  0.3      1
135  1.0  0.0 -1.0 -0.12  0.28  0.0  0.7  1.0  0.7      1
136 -0.7 -0.7 -0.7 -1.29 -0.07 -0.7  0.3  0.3 -1.3      1
137  0.0 -0.3 -0.3 -1.43 -2.15 -0.3 -0.7  1.3  0.3      1
138  0.0  0.7  0.7 -2.33 -0.78 -0.3 -0.3  1.3  0.0      1
139  0.0 -0.3 -0.7  0.80  1.00 -0.7 -0.3  1.3  0.0      1
140 -1.0 -1.0 -1.3 -0.94 -0.07 -0.3 -0.3  1.3  0.0      1
141 -0.7 -1.3 -1.7 -2.29 -1.53 -1.0 -0.3  0.8  0.0      1
142 -0.7 -2.7  0.0 -0.19 -0.90 -1.3 -0.3  1.0 -0.3      1
143 -0.3  0.7  1.0 -5.94 -1.50  0.0 -1.0  1.0  0.0      1
144 -0.7 -2.7 -1.3 -2.33 -1.85 -0.7 -0.3  0.8 -0.7      1
146  0.7  0.7  0.0 -0.69 -1.21  0.0  0.0  1.3  0.0      1
147 -0.3 -1.3 -0.3 -1.03 -0.39 -0.7 -0.3  3.0 -0.3      1
148  1.7  0.0 -0.7 -1.80 -1.53 -1.0 -1.3  1.3  1.0      1
149  0.7 -0.3  0.0 -0.56  0.03 -0.7 -0.7  0.8  0.3      1
152 -1.3 -1.7 -0.7 -1.92 -0.59 -1.7 -1.0  0.7 -0.3      1
154 -0.3 -1.3 -1.7 -1.52  0.07  0.0  0.3  1.3  0.3      1
156 -0.3 -1.0 -0.3 -1.43 -0.59  0.0 -0.7  1.0  0.0      1
157 -0.7 -1.3 -0.7 -1.40 -1.14 -0.7  0.3  1.3  0.3      1
158  1.3 -0.7  0.0  2.27  1.59  0.3 -0.7  3.0  0.0      1
159 -0.3 -0.3 -0.3  0.54 -1.21 -1.3 -0.3  1.0  0.0      1
160  0.3 -0.7  0.0 -0.56 -0.90 -0.3  0.0  3.0  0.0      1
161 -0.3 -1.7  0.0 -1.03 -2.53 -0.7 -0.7  3.0 -0.3      1
162  0.0  0.3 -0.7 -1.18 -0.90 -0.3 -0.7  1.3  0.3      1
164  0.0  0.0  0.0 -1.05 -1.50 -0.3 -0.3  0.8 -0.7      0
165  0.3 -2.3 -1.0 -0.12  1.00  0.0  0.0  1.0  0.3      1
166 -0.3 -1.3 -0.3  1.65  0.34  0.0  0.0  1.0  0.3      1
168 -1.7 -2.7 -1.3 -0.82 -1.50 -1.0 -1.0  3.1  0.3      1
169  0.3 -0.7 -0.3  0.42 -0.59 -0.7 -1.0  0.8  0.0      1
170 -0.3 -1.7 -0.7 -0.19 -0.39 -0.3  0.3  1.0  0.3      1
174 -1.0 -0.7 -0.7  0.10  1.35 -1.0 -1.3  3.0  1.0      1
175  0.3 -1.0 -0.3  0.53  1.39 -0.7  0.3  1.0  0.7      1
176 -1.0 -2.7 -1.0 -1.52 -0.78 -1.3 -0.3  1.0  0.3      1
177 -0.7 -2.7 -1.7 -2.60 -2.17  0.0  0.7  0.5 -0.7      1
179  0.3  0.0 -0.3 -0.69 -0.28 -0.7  0.0  1.3  0.7      1
188  0.7  0.0  0.7  1.13  0.68 -1.3 -1.0  0.8 -0.3      0
194  0.0 -1.3  0.7 -1.05 -0.78  0.0 -1.0  3.0  0.0      1
198  0.3  0.7  1.0  0.07  1.06  0.0 -0.3  0.8  0.0      0
```

```
221 -0.3  1.3  1.3 -0.79 -0.39 -1.3 -0.7  1.0 -2.0     0
227 -0.7 -0.3 -0.3 -3.80 -2.17 -2.3 -1.0  0.5 -1.3     1
232  0.0 -0.3 -0.3 -2.29 -0.90 -1.3  0.3  3.1  1.0     1
233  0.0 -2.7 -0.7  0.10  0.64  0.3  0.7  0.8  0.0     1
240  0.0  0.3  0.0 -0.32  0.65 -0.3 -0.3  1.3  0.0     1
```

```
print("Datos del clúster 2:")
```

[1] "Datos del clúster 2:"

```
print(cluster_2)
```

```
      ag    ptg19 sex_Woman el_Elementary el_High.School el_Secondary
3   33.1 18.24000         1             0              0            0
6   43.0 32.70000         1             0              0            0
7   55.6 25.44000         1             0              0            0
8   53.6 18.24000         1             0              0            0
9   46.3 23.23000         1             0              0            0
11  58.7 26.01000         1             0              1            0
13  42.7 19.37000         1             0              0            0
15  48.7 28.31000         1             0              0            0
16  53.6 20.97000         1             0              0            0
19  55.5 23.65000         1             0              0            0
22  64.6 30.90000         1             0              0            0
24  46.6 20.51000         1             0              0            0
26  29.9 17.54000         1             0              0            0
27  56.9 27.41000         1             0              1            0
28  47.6 23.67000         0             0              1            0
33  45.3 23.53000         1             0              0            0
36  61.6 23.87000         1             0              0            0
40  48.7 25.28000         1             0              0            0
45  36.5 29.40000         1             0              0            0
46  65.1 21.64000         1             0              0            0
47  62.8 24.40000         1             0              0            0
48  52.3 19.43000         1             0              0            0
49  53.0 25.40000         0             0              0            0
50  57.6 23.28000         1             0              1            0
51  51.1 27.57000         1             0              0            0
52  45.2 26.25000         1             0              0            0
53  65.9 24.69000         1             0              0            0
54  31.1 22.97000         1             0              0            0
55  42.6 24.58000         0             0              0            0
56  33.3 27.19237         0             0              0            0
57  61.6 29.72000         0             0              0            0
65  55.2 27.70000         1             0              1            0
68  57.2 45.79000         0             0              1            0
75  25.5 22.21000         1             0              0            0
80  44.8 28.12000         1             0              1            0
85  65.9 25.92000         1             0              1            0
95  33.7 39.51000         1             0              0            0
101 52.7 35.62000         1             0              1            0
103 42.7 20.95000         1             0              0            0
108 39.9 35.13000         1             0              0            0
113 58.3 23.06000         1             0              1            0
115 44.3 27.55000         1             0              0            0
```

117	47.1	24.17000	1	0	0	0
118	50.8	22.18000	1	0	1	0
124	57.0	40.61000	0	0	1	0
128	40.1	30.63000	1	0	0	0
131	46.1	30.76000	1	0	0	0
132	58.3	22.97000	1	0	1	0
133	59.7	24.46000	1	0	1	0
145	52.0	24.17000	1	0	0	0
150	43.1	21.91000	1	0	0	0
151	45.9	41.29000	1	0	1	0
153	41.9	20.42000	1	0	0	0
155	59.2	25.38000	1	0	0	0
163	45.8	27.87000	1	0	0	0
167	30.2	24.14000	1	0	0	0
171	38.4	24.19000	1	0	0	0
172	49.6	25.37000	0	0	0	0
173	58.7	28.55000	1	0	0	0
178	38.2	30.56000	1	0	0	0
180	29.4	17.95000	1	0	0	0
181	42.3	33.41000	1	0	0	0
182	41.5	23.57000	1	0	0	0
183	29.4	24.12000	1	0	0	0
184	40.8	25.43000	1	0	0	0
185	65.3	35.22000	0	0	1	0
186	56.1	23.42000	1	0	0	0
187	33.8	20.78000	1	0	0	0
189	61.0	37.46000	1	0	0	0
190	50.7	25.96000	1	0	0	0
191	62.6	28.57000	0	0	0	0
192	54.3	23.95000	1	0	0	0
193	28.9	26.75000	1	0	0	0
195	65.8	28.06000	1	0	1	0
196	48.7	22.63000	1	0	1	0
197	70.8	24.39000	0	0	0	0
199	40.7	27.42241	1	0	0	0
200	64.6	27.85000	0	0	1	0
201	55.3	26.19000	0	0	1	0
202	54.1	24.85000	0	0	0	0
203	57.8	19.17000	1	0	1	0
204	30.1	21.73000	1	0	0	0
205	57.7	38.63000	1	0	1	0
206	31.9	21.80000	1	0	0	0
207	41.1	27.36000	1	0	0	0
208	63.5	29.96000	1	0	0	0
209	40.9	26.15119	1	0	0	0
210	48.3	25.08000	0	0	0	0
211	49.0	23.54000	1	0	0	0
212	50.1	30.64000	0	0	0	0
213	49.0	22.04000	1	0	0	0
214	55.8	34.02000	1	0	1	0
215	43.3	23.44000	1	0	0	0
216	35.9	20.18000	1	0	0	0
217	51.6	22.79000	1	0	0	0
218	51.8	36.18000	1	0	0	0

219	34.8	28.10000	1	0	0	0
220	51.4	35.75000	1	0	1	0
222	52.6	21.08000	1	0	0	0
223	50.7	25.11000	1	0	0	0
224	45.4	37.82000	1	0	0	0
225	33.0	20.02000	1	0	0	0
226	57.5	30.02000	1	0	0	0
228	37.9	24.93343	1	0	0	0
229	34.7	24.72016	1	0	0	0
230	42.4	26.62936	1	0	0	0
231	46.3	23.24000	1	0	1	0
234	37.0	21.33000	1	0	0	0
235	37.6	24.42000	1	0	0	0
236	51.0	87.31000	0	0	1	0
237	62.1	29.67000	1	0	0	0
238	63.2	27.35000	0	0	1	0
239	47.1	24.10000	0	0	0	0
241	25.4	16.14000	1	0	0	0

	el_Specialist.Master	el_University.Deg.	spec1_Hospitalization
3	0	0	0
6	1	0	0
7	1	0	0
8	0	1	0
9	1	0	0
11	0	0	0
13	0	1	0
15	0	1	0
16	1	0	0
19	1	0	0
22	1	0	0
24	1	0	0
26	1	0	0
27	0	0	0
28	0	0	0
33	1	0	0
36	0	1	0
40	0	1	0
45	0	1	0
46	0	0	0
47	0	0	0
48	0	0	0
49	0	1	0
50	0	0	0
51	0	0	0
52	0	1	0
53	1	0	0
54	1	0	0
55	1	0	0
56	1	0	0
57	0	1	0
65	0	0	0
68	0	0	0
75	0	1	0
80	0	0	0

85	0	0	0
95	0	1	0
101	0	0	0
103	0	1	0
108	0	1	0
113	0	0	0
115	0	1	0
117	1	0	0
118	0	0	0
124	0	0	0
128	0	0	0
131	0	1	0
132	0	0	0
133	0	0	0
145	0	1	0
150	0	1	0
151	0	0	0
153	1	0	0
155	0	1	0
163	0	1	0
167	0	0	0
171	0	1	0
172	0	1	0
173	1	0	0
178	0	0	0
180	0	1	0
181	0	1	0
182	0	1	0
183	0	1	0
184	0	1	0
185	0	0	0
186	0	1	0
187	1	0	0
189	0	1	0
190	1	0	0
191	0	1	0
192	0	1	0
193	1	0	0
195	0	0	0
196	0	0	0
197	1	0	0
199	0	1	0
200	0	0	0
201	0	0	0
202	1	0	0
203	0	0	0
204	0	1	0
205	0	0	0
206	0	1	0
207	1	0	0
208	0	1	0
209	0	1	0
210	0	1	0
211	1	0	0

212	0	1	0
213	0	1	0
214	0	0	0
215	1	0	0
216	0	0	0
217	0	0	0
218	1	0	0
219	1	0	0
220	0	0	0
222	0	1	0
223	0	0	0
224	0	0	0
225	1	0	0
226	1	0	0
228	0	0	0
229	0	0	0
230	0	0	0
231	0	0	0
234	0	1	0
235	1	0	0
236	0	0	0
237	0	1	0
238	0	0	0
239	1	0	0
241	1	0	0

	spec1_Mild.Moderated	spec1_UCI	tn46	tn52	tn36
3	1	0	0.7000000	2.000000000	1.0000000
6	1	0	0.3000000	0.400000010	-0.7000000
7	1	0	-0.3000000	-0.699999990	-0.3000000
8	1	0	-0.3000000	0.400000010	-0.3000000
9	1	0	-0.7000000	1.000000000	-1.7000000
11	1	0	0.0000000	-0.400000010	0.3000000
13	1	0	-0.7000000	1.000000000	-1.7000000
15	1	0	0.0000000	0.699999990	0.0000000
16	1	0	-0.3000000	-1.700000000	-0.3000000
19	1	0	-0.7000000	0.300000010	-0.3000000
22	1	0	-0.3000000	0.000000000	-0.7000000
24	1	0	0.3000000	1.300000000	0.0000000
26	1	0	-0.7000000	0.699999990	-1.3000000
27	1	0	0.0000000	1.000000000	0.0000000
28	1	0	0.3000000	-0.699999990	0.0000000
33	0	0	0.3000000	1.000000000	-1.3000000
36	1	0	0.0000000	0.699999990	1.0000000
40	1	0	0.3000000	-0.400000010	0.0000000
45	1	0	-0.7000000	1.400000000	-0.7000000
46	1	0	-0.3000000	0.000000000	0.0000000
47	1	0	0.7000000	1.400000000	-0.3000000
48	1	0	0.3000000	0.300000010	-0.7000000
49	1	0	1.3000000	-0.699999990	1.7000000
50	1	0	0.0000000	0.000000000	-0.3000000
51	1	0	2.0000000	0.000000000	-0.7000000
52	1	0	-0.7000000	0.000000000	-0.3000000
53	1	0	0.7000000	-2.300000000	0.0000000
54	1	0	0.0000000	-0.600000020	-0.3000000

30

55	1	0	-1.0000000	-1.300000000	-1.7000000
56	1	0	2.3000000	-0.400000010	-0.3000000
57	1	0	0.0000000	0.300000010	-0.3000000
65	1	0	1.0000000	1.300000000	1.7000000
68	1	0	0.7000000	0.899999980	0.0000000
75	1	0	1.7000000	-0.300000010	-1.3000000
80	1	0	-0.7000000	-1.700000000	0.7000000
85	1	0	0.0000000	0.600000020	-0.7000000
95	1	0	-1.0000000	0.600000020	-0.7000000
101	1	0	-0.3000000	-0.800000010	0.3000000
103	1	0	0.7000000	0.300000010	-0.3000000
108	1	0	0.7000000	-0.300000010	-0.3000000
113	1	0	0.7000000	0.000000000	2.3000000
115	1	0	-1.0000000	0.000000000	0.7000000
117	1	0	-1.0000000	1.700000000	-0.3000000
118	1	0	1.0000000	-0.400000010	0.3000000
124	1	0	2.3000000	0.300000010	0.0000000
128	1	0	-1.0000000	-0.600000020	1.3000000
131	1	0	-0.7000000	1.000000000	0.0000000
132	1	0	0.0000000	1.400000000	1.0000000
133	1	0	-1.7000000	-0.600000020	-0.7000000
145	1	0	1.0000000	0.300000010	0.3000000
150	1	0	1.7000000	0.300000010	-0.3000000
151	1	0	-0.7000000	-0.600000020	0.7000000
153	1	0	0.0000000	0.000000000	-2.0000000
155	1	0	0.3000000	0.000000000	0.7000000
163	1	0	0.0000000	0.000000000	0.7000000
167	1	0	0.0000000	0.400000010	-0.7000000
171	1	0	0.0000000	1.600000000	0.0000000
172	1	0	-1.3000000	0.000000000	-0.7000000
173	1	0	0.3000000	0.699999990	-1.3000000
178	1	0	0.7000000	0.300000010	-2.3000000
180	0	0	-1.0000000	0.600000020	0.3000000
181	1	0	0.3000000	0.000000000	-0.3000000
182	1	0	0.3000000	0.300000010	0.0000000
183	1	0	0.3000000	0.300000010	2.0000000
184	1	0	0.0000000	0.000000000	1.3000000
185	1	0	0.3000000	0.000000000	0.0000000
186	1	0	-0.3000000	1.700000000	0.3000000
187	1	0	-1.0000000	0.300000010	-0.7000000
189	1	0	0.7000000	-0.400000010	-0.3000000
190	1	0	0.7000000	-0.699999990	0.0000000
191	1	0	-0.7000000	-0.699999990	1.0000000
192	1	0	-0.3000000	0.000000000	0.7000000
193	1	0	-1.0000000	-0.600000020	0.3000000
195	1	0	2.7000000	-1.000000000	1.7000000
196	1	0	1.7000000	1.000000000	0.0000000
197	1	0	0.0000000	0.400000010	0.7000000
199	1	0	-2.3000000	0.300000010	0.3000000
200	1	0	0.3000000	-0.300000010	0.7000000
201	1	0	1.0000000	-1.000000000	1.7000000
202	1	0	1.0000000	-0.699999990	0.0000000
203	0	0	-0.3000000	1.000000000	0.0000000
204	1	0	0.0000000	-0.699999990	0.3000000

205	1	0	-0.3000000	0.400000010	2.0000000
206	1	0	1.0000000	0.699999990	-0.3000000
207	1	0	0.0000000	0.300000010	-0.3000000
208	0	0	0.7000000	0.000000000	-0.3000000
209	1	0	-0.7000000	-0.300000010	1.3000000
210	1	0	-1.0000000	-1.700000000	0.0000000
211	1	0	-1.0000000	-1.000000000	-0.3000000
212	1	0	2.3000000	0.300000010	2.3000000
213	1	0	-1.0000000	0.000000000	-0.3000000
214	1	0	1.0000000	0.000000000	1.3000000
215	0	0	0.0000000	1.300000000	0.0000000
216	1	0	0.0000000	-0.300000010	-0.3000000
217	1	0	2.0000000	-1.400000000	-0.7000000
218	1	0	-0.3000000	0.400000010	-0.3000000
219	1	0	-0.1936254	0.008326257	-0.3344785
220	1	0	2.3000000	-0.300000010	0.3000000
222	0	0	0.7000000	-0.300000010	-0.3000000
223	1	0	1.0000000	-0.699999990	0.0000000
224	0	0	0.0000000	0.699999990	2.3000000
225	1	0	0.0000000	0.400000010	0.3000000
226	1	0	-0.7000000	0.000000000	1.7000000
228	1	0	0.3389087	0.154617766	-0.3216190
229	1	0	0.0000000	1.300000000	-0.3000000
230	1	0	0.0000000	0.000000000	0.0000000
231	1	0	-0.7000000	-1.300000000	0.0000000
234	1	0	-0.7000000	0.699999990	-1.7000000
235	1	0	0.0000000	-0.300000010	0.0000000
236	1	0	1.7000000	0.000000000	0.7000000
237	1	0	0.0000000	-0.300000010	0.0000000
238	1	0	2.7000000	-1.300000000	0.0000000
239	1	0	0.0000000	0.000000000	-0.3000000
241	1	0	-0.4050728	0.211006487	-0.5695441

	tn38	tn40	tn42	tn22	tn44	tn14
3	0.3000000	1.00000000	0.3000000	-0.5000000	-1.0000000	0.3000000
6	-0.3000000	0.30000000	-0.3000000	1.0000000	0.0000000	0.7000000
7	-0.7000000	-0.30000000	-1.0000000	0.4000000	-0.7000000	0.3000000
8	0.0000000	-0.30000000	-1.7000000	0.3000000	1.0000000	0.7000000
9	-2.3000000	-0.30000000	-0.7000000	0.3000000	-2.0000000	-0.3000000
11	0.3000000	0.70000000	1.0000000	-0.3000000	-1.3000000	1.3000000
13	-1.3000000	-1.00000000	-1.7000000	0.9000000	0.0000000	1.3000000
15	-0.3000000	-1.00000000	-1.7000000	0.3000000	0.0000000	1.0000000
16	-0.7000000	0.30000000	0.0000000	-0.7000000	0.0000000	0.3000000
19	-0.7000000	0.00000000	-0.7000000	0.0000000	1.0000000	1.3000000
22	-0.7000000	-0.70000000	-1.3000000	0.2000000	1.0000000	1.0000000
24	0.7000000	0.70000000	-1.0000000	0.2000000	-0.7000000	-0.3000000
26	0.0000000	-1.00000000	-1.0000000	1.5000000	-0.7000000	1.7000000
27	-0.3000000	1.00000000	-0.3000000	-0.6000000	-0.3000000	0.7000000
28	1.3000000	1.30000000	-0.7000000	-0.1000000	0.0000000	0.7000000
33	-0.3000000	-0.30000000	0.0000000	0.7000000	-0.7000000	0.3000000
36	0.7000000	0.30000000	0.3000000	1.0000000	-1.3000000	1.0000000
40	-0.7000000	0.00000000	-2.0000000	1.0000000	1.0000000	0.3000000
45	-0.7000000	0.00000000	-0.3000000	0.4000000	-0.7000000	1.0000000
46	-0.7000000	-0.70000000	-0.3000000	1.8000000	-0.7000000	1.0000000
47	0.7000000	0.00000000	-0.3000000	-0.1000000	0.0000000	1.7000000

```
48    0.0000000   0.30000000   0.3000000  -0.7000000  -1.0000000   1.0000000
49    1.3000000   1.00000000   2.0000000   0.2000000   2.3000000   0.7000000
50    0.0000000   0.30000000   0.3000000   0.7000000  -1.3000000   1.3000000
51    0.0000000   0.00000000  -0.7000000   0.6000000   0.7000000   0.3000000
52   -0.3000000  -1.00000000  -1.7000000   0.6000000   0.0000000   1.0000000
53    0.3000000  -0.30000000  -1.0000000   1.2000000   0.3000000   0.7000000
54    0.0000000   0.00000000   0.0000000   1.0000000  -0.3000000   2.0000000
55   -0.7000000  -1.70000000  -1.0000000   1.6000000  -1.0000000   0.7000000
56    1.0000000   0.00000000  -0.3000000   0.6331074   2.3000000   0.7000000
57    0.7000000  -0.70000000  -0.3000000   1.5000000   0.3000000   0.0000000
65    2.0000000   1.70000000  -0.3000000   1.2000000   1.3000000   1.0000000
68    0.7000000  -1.70000000   0.3000000   0.9000000   1.7000000   1.3000000
75   -0.3000000  -1.00000000  -1.7000000   1.6000000   0.0000000   0.3000000
80    0.0000000   0.00000000   0.0000000   0.3000000  -0.7000000   1.3000000
85   -0.3000000  -0.30000000  -1.3000000   0.6000000  -0.3000000   0.3000000
95   -1.0000000  -0.30000000  -2.0000000   0.6000000   0.7000000   0.7000000
101  -0.3000000   0.00000000  -0.7000000   1.8000000   0.0000000   0.7000000
103  -0.3000000  -1.70000000  -1.3000000   0.3000000  -2.0000000   1.0000000
108  -1.0000000   0.30000000  -1.3000000  -0.1000000   0.7000000  -0.3000000
113   2.0000000   1.30000000   2.3000000   1.3000000   0.3000000   2.7000000
115   0.7000000   1.00000000   0.3000000   1.2000000   0.0000000   1.0000000
117  -1.0000000   0.00000000  -1.3000000   0.8000000  -0.3000000   0.3000000
118  -0.3000000   0.30000000  -0.3000000   1.7000000   1.3000000   1.3000000
124   1.0000000   0.30000000  -0.3000000   0.6000000   0.3000000   0.7000000
128   1.3000000   0.30000000   0.0000000  -0.4000000  -1.0000000   2.3000000
131  -0.3000000  -0.30000000  -0.3000000   0.8000000  -0.7000000   0.7000000
132   0.7000000   0.30000000  -0.3000000   1.0000000   0.0000000   1.7000000
133  -0.7000000  -0.30000000  -0.7000000   0.3000000  -1.3000000   1.0000000
145   0.3000000   0.00000000  -0.3000000   0.3000000   0.7000000   0.0000000
150  -0.7000000  -0.30000000  -1.7000000   1.0000000   1.0000000   1.0000000
151   0.0000000   0.00000000   0.3000000   1.3000000  -0.7000000   0.7000000
153  -2.0000000  -1.00000000  -1.3000000   0.4000000  -0.3000000   0.3000000
155   1.3000000   0.00000000  -1.0000000   1.0000000  -0.7000000   1.3000000
163   0.7000000   1.00000000  -0.3000000   1.3000000   0.0000000   0.7000000
167  -1.3000000  -1.00000000  -1.3000000   0.5000000  -0.3000000   1.7000000
171  -0.7000000  -0.70000000  -0.7000000   0.9000000   0.3000000   0.3000000
172  -0.7000000  -0.30000000  -1.0000000   0.2700000   1.0000000   0.7000000
173  -1.3000000  -1.30000000  -0.7000000   1.6000000   0.0000000   1.0000000
178  -0.7000000  -0.30000000   0.0000000   0.6000000  -0.3000000   1.3000000
180   0.0000000   0.00000000  -1.3000000   0.5000000  -0.3000000   2.3000000
181   0.0000000   0.30000000  -1.7000000  -0.4000000  -2.0000000   0.7000000
182   0.3000000   0.30000000  -1.3000000   0.7000000  -0.7000000   1.0000000
183   2.7000000   2.70000000  -1.0000000   0.8000000   0.0000000   1.3000000
184   0.0000000   0.00000000   0.7000000  -0.4000000   1.0000000   2.0000000
185   0.7000000   0.30000000   0.0000000  -0.7000000   0.0000000   0.0000000
186   0.3000000   0.70000000   0.3000000   0.2000000   0.0000000   1.3000000
187  -0.7000000  -0.70000000  -1.7000000   2.0000000  -0.3000000   0.3000000
189  -0.7000000  -0.70000000  -0.7000000   1.6000000   0.7000000   1.7000000
190   0.0000000   0.30000000   0.3000000   0.2000000   0.0000000   0.7000000
191   2.0000000   1.30000000   0.7000000  -1.6000000  -1.3000000   2.0000000
192   1.3000000   1.30000000  -1.7000000   0.4000000  -0.7000000   0.7000000
193   1.0000000   1.70000000  -0.3000000   1.2000000  -0.3000000   0.7000000
195   1.0000000   0.70000000   0.3000000   0.5000000   0.0000000   0.0000000
196   0.7000000   1.00000000  -1.0000000   0.6000000   1.0000000   1.7000000
```

```
197  0.7000000   1.00000000   0.0000000   0.0000000   1.0000000   1.7000000
199  1.3000000   1.00000000  -0.7000000   0.2000000   0.0000000   0.7000000
200  0.0000000  -1.00000000  -0.7000000   0.8000000   0.0000000   0.7000000
201  1.7000000   0.70000000   0.3000000   0.8000000   0.3000000   1.0000000
202  0.0000000  -0.70000000  -1.0000000   1.1000000   0.7000000   0.3000000
203  1.0000000  -0.30000000   0.3000000   1.6000000   0.3000000   1.0000000
204  0.0000000   0.30000000   0.3000000   1.8000000   0.0000000   0.7000000
205  0.7000000   0.00000000   0.0000000   1.4000000  -0.3000000   1.3000000
206 -0.3000000  -0.70000000   0.3000000   1.8000000   0.0000000   0.7000000
207 -0.3000000   0.70000000  -0.7000000  -0.4000000  -0.3000000  -0.3000000
208  0.0000000   0.00000000   0.0000000   0.1000000   1.0000000   1.0000000
209  0.7000000   1.00000000   1.0000000   1.4000000  -0.7000000   1.7000000
210  0.3000000   0.30000000   0.3000000   2.0000000  -2.0000000   1.0000000
211 -0.7000000   1.00000000   0.7000000   0.4000000  -0.3000000   0.3000000
212  2.3000000   2.30000000   2.3000000   1.0000000   2.3000000   1.7000000
213  1.0000000   0.70000000  -0.7000000   0.8000000  -2.0000000   1.7000000
214  2.0000000   1.30000000   0.0000000   0.6000000  -0.3000000   0.3000000
215 -0.3000000  -0.30000000  -1.7000000   0.2000000   0.7000000   1.3000000
216 -1.3000000  -2.00000000  -1.3000000   0.8000000   0.7000000   1.0000000
217  1.0000000   1.30000000   0.3000000   1.4000000   0.7000000   0.7000000
218  0.7000000   1.00000000   0.0000000   0.8000000  -2.0000000   0.0000000
219 -0.4843276   0.36919106  -0.3884471   0.6442551  -0.2315443   0.5955351
220  1.3000000   0.30000000   0.3000000   2.6000000   1.3000000   1.0000000
222  0.3000000  -0.30000000   0.0000000   0.1000000  -0.7000000   0.0000000
223 -1.0000000  -1.00000000   0.3000000   1.2000000   0.7000000   1.7000000
224  2.0000000   1.30000000   1.7000000   3.0000000  -0.3000000   1.3000000
225  0.0000000   0.70000000  -0.3000000   0.4000000  -0.3000000   1.0000000
226  0.7000000   1.00000000   0.7000000   1.2000000  -0.7000000   1.0000000
228 -0.2380424  -0.29804675  -0.3265751   0.3526449  -0.2024081   1.2087006
229  0.7000000   0.30000000   0.0000000   0.6000000  -0.3000000   1.3000000
230 -0.3000000  -0.30000000   0.0000000   0.8000000   0.7000000  -0.3000000
231  0.0000000   0.00000000   0.0000000   0.8000000   0.0000000   0.7000000
234 -0.7000000  -1.70000000  -1.3000000   0.6000000   0.0000000   1.0000000
235 -1.0000000   0.70000000   1.7000000  -0.2000000  -0.3000000   1.3000000
236  0.0000000   1.30000000   1.7000000  -0.8000000   1.3000000   0.7000000
237 -0.3000000  -0.30000000  -0.3000000   0.6000000   0.3000000   1.0000000
238  1.3000000   0.70000000   1.3000000  -0.6000000   1.7000000   1.0000000
239  0.0000000   0.70000000   0.0000000   1.8000000   1.7000000   0.7000000
241 -0.2685230  -0.03704518  -0.5917289   0.9404058  -0.4714624   1.0037005
          tn24        tn12         tn6        tn30        tn34         tn8      tn48
3   -1.00000000   1.3000000   1.7000000   0.3200000  -1.0000000   0.3000000 3.100000
6   -0.70000000   0.0000000   1.3800000   1.7100000   0.0000000   2.7000000 1.300000
7    0.40000000   1.0000000  -1.1800000  -1.2200000  -0.3000000   0.0000000 1.300000
8   -0.70000000   0.0000000  -0.2000000   0.0300000  -1.0000000  -0.3000000 3.100000
9   -2.70000000   0.7000000  -0.0100000  -0.0700000   1.0000000  -0.3000000 1.000000
11  -0.30000000   1.0000000   0.4200000  -0.5900000   1.0000000   0.3000000 1.300000
13  -0.30000000   1.0000000   1.2600000   1.7100000  -0.7000000   2.7000000 0.800000
15  -1.00000000   0.3000000   1.0300000   1.3600000   0.0000000  -0.7000000 3.100000
16   1.00000000   0.0000000  -0.8100000   0.6500000  -0.7000000  -0.3000000 0.800000
19  -1.00000000   1.0000000  -0.3200000  -1.2100000  -0.3000000  -0.7000000 1.300000
22  -0.30000000   1.3000000   0.5900000   0.4000000  -0.3000000   0.0000000 3.000000
24  -1.30000000   1.0000000   0.9100000   1.0000000   0.0000000  -1.0000000 3.000000
26   0.30000000   0.7000000  -0.1500000  -0.1300000   0.0000000   0.3000000 3.100000
27  -1.30000000   1.0000000   0.1700000   0.3400000  -0.3000000   0.3000000 3.100000
```

```
28    0.00000000   0.0000000   0.9200000   0.2800000   0.3000000   2.7000000 3.100000
33   -1.00000000   0.7000000   0.2200000   1.0000000   0.0000000  -0.3000000 1.300000
36    0.00000000   1.0000000   0.4600000   1.7300000  -0.7000000   0.0000000 1.300000
40   -0.30000000   0.0000000  -1.0500000  -0.7800000  -0.7000000  -1.0000000 3.000000
45   -1.70000000  -0.3000000  -0.4300000   1.0300000   0.0000000   2.7000000 1.300000
46   -0.30000000   1.0000000   0.9800000   0.0600000  -0.7000000  -0.3000000 3.000000
47   -2.70000000   2.0000000   1.2500000   0.7300000  -1.0000000  -0.7000000 1.000000
48   -1.00000000  -0.7000000   1.5300000   0.9600000  -0.7000000  -1.3000000 3.000000
49    0.70000000   0.0000000   1.1600000   0.3400000  -0.7000000  -0.7000000 1.300000
50    1.70000000   1.7000000   0.7900000   0.3400000   0.3000000  -0.7000000 3.000000
51   -0.30000000   1.0000000  -0.0700000  -1.2100000  -0.7000000  -1.3000000 1.300000
52    0.00000000   0.7000000   0.9100000   0.2800000   0.0000000  -0.3000000 1.300000
53    1.30000000   1.0000000   0.9800000   0.4000000  -1.0000000  -0.7000000 0.800000
54    0.30000000   1.7000000  -0.6700000   0.3200000  -1.3000000  -1.0000000 1.000000
55    2.00000000  -0.3000000  -0.5900000  -0.7800000   0.0000000   0.3000000 1.000000
56    0.70000000   1.3000000   1.9800000   1.3900000   2.3000000   2.3000000 3.100000
57   -1.30000000   0.3000000   0.5900000   1.4000000   0.7000000   2.0000000 3.100000
65    0.00000000   1.0000000  -0.0500000  -0.5900000  -1.0000000   0.0000000 1.300000
68    0.00000000   1.7000000  -0.5600000  -0.9000000   1.3000000   1.3000000 3.100000
75   -0.70000000   0.0000000  -1.5200000  -0.1300000   1.0000000   0.3000000 0.800000
80    1.00000000   1.3000000   0.9200000   1.3500000  -1.0000000  -1.0000000 3.100000
85   -1.30000000   1.0000000   0.8500000   1.7300000   0.0000000  -0.3000000 3.100000
95   -1.30000000   0.0000000   0.8900000   1.0300000   0.3000000   0.0000000 1.300000
101   1.80000000   0.7000000  -0.8100000  -1.2200000   0.3000000   0.3000000 3.100000
103  -0.30000000  -0.3000000   1.5000000   1.3500000   0.7000000   0.3000000 1.000000
108  -0.70000000  -0.3000000   0.5300000   0.3200000   1.0000000   0.3000000 3.000000
113  -0.30000000   0.7000000   2.2700000   1.2800000   0.3000000   0.3000000 3.100000
115   0.00000000   0.7000000  -0.7100000  -0.0700000  -1.3000000  -0.7000000 1.300000
117  -2.70000000  -0.7000000   0.1000000   0.2800000  -0.7000000  -1.0000000 0.700000
118   0.70000000   1.3000000   1.0300000   1.5900000   0.3000000  -0.3000000 1.000000
124   0.00000000   0.3000000  -1.0600000  -0.2800000  -0.7000000  -0.7000000 1.000000
128   0.30000000   1.7000000   0.6900000   1.3500000   0.7000000   0.0000000 3.100000
131   0.00000000   0.7000000  -0.3600000  -0.4200000   2.7000000   0.7000000 3.000000
132   0.30000000   1.7000000   0.0400000   0.0300000  -0.7000000  -0.3000000 1.300000
133   0.30000000   1.0000000  -0.3200000   0.6500000   0.0000000  -0.3000000 0.600000
145  -2.30000000   0.3000000   1.9000000   0.6500000  -0.7000000   0.0000000 1.000000
150  -1.30000000  -0.3000000   0.1000000   0.6400000   0.0000000  -0.3000000 3.000000
151   0.30000000   0.3000000   1.3800000   1.3500000   0.7000000  -0.7000000 1.300000
153  -1.00000000  -0.3000000   1.0300000   1.3500000   0.0000000   2.3000000 3.000000
155   0.00000000   1.7000000   1.2800000   1.5900000  -0.3000000  -0.7000000 1.300000
163   0.00000000   0.7000000  -0.5900000   0.2800000   0.0000000   0.0000000 3.000000
167  -1.70000000   1.3000000   1.0100000   0.3200000  -1.0000000   0.3000000 1.300000
171  -2.30000000   1.0000000   1.2500000   1.0300000  -0.3000000  -1.0000000 3.000000
172  -1.00000000   0.3000000  -0.8300000   1.3500000   0.0000000  -1.3000000 3.000000
173  -1.00000000   1.0000000   1.0300000   0.9600000  -1.3000000   2.0000000 0.300000
178  -0.30000000   1.7000000   1.7300000   1.3900000   0.0000000  -1.0000000 1.300000
180  -0.30000000   0.5000000   0.9400000   1.1700000  -1.0000000   2.3000000 1.300000
181   1.00000000   0.7000000   0.5600000   1.3500000  -1.0000000   0.3000000 3.000000
182  -0.30000000   1.0000000   0.1000000   0.6400000  -0.3000000   0.3000000 1.000000
183   0.00000000   1.0000000  -0.1500000   1.6000000   0.7000000  -0.3000000 3.100000
184   1.00000000   1.7000000   2.1900000   1.7100000   0.7000000  -1.0000000 3.100000
185  -1.30000000  -0.3000000  -0.3100000  -0.0600000  -0.3000000  -0.7000000 1.000000
186  -0.70000000   1.3000000   0.1700000   0.9700000  -0.3000000  -0.3000000 3.100000
187  -0.30000000   1.3000000   0.8900000   1.0300000  -0.7000000   0.0000000 1.300000
```

```
189 -0.30000000   1.3000000   0.7200000   1.4000000 -1.7000000 -1.7000000 0.500000
190  0.70000000   0.3000000  -0.1900000  -0.5900000   0.3000000   0.0000000 1.300000
191  2.00000000   2.3000000  -0.8300000   0.4000000 -0.7000000   0.7000000 3.100000
192 -0.70000000   0.7000000   2.1400000   0.9600000   0.3000000 -0.7000000 0.800000
193  0.30000000   1.0000000  -1.2400000   0.7300000 -1.3000000 -1.3000000 1.300000
195  0.30000000   0.3000000  -0.1800000   1.0600000 -1.0000000   0.0000000 1.000000
196  0.00000000   1.7000000   0.9100000   1.2800000 -0.7000000 -1.3000000 0.800000
197  0.30000000   2.3000000   2.2500000   1.6600000 -0.3000000 -1.3000000 3.000000
199 -1.30000000   0.0000000   0.5700000   1.3500000   0.0000000 -1.3000000 1.000000
200  0.00000000   1.0000000   1.2500000   0.7300000   0.0000000   1.0000000 1.300000
201  0.30000000   0.3000000   1.4000000   0.9700000   0.7000000 -0.7000000 1.000000
202 -0.30000000   0.7000000   1.2800000   0.9700000 -0.7000000 -0.7000000 3.100000
203 -1.30000000   0.7000000   1.6500000   0.0300000   0.3000000 -0.7000000 1.000000
204  0.70000000   1.3000000   1.3700000   1.3900000   0.3000000   0.0000000 1.300000
205 -0.70000000   2.3000000  -0.0700000   0.9700000 -0.7000000 -0.3000000 1.000000
206 -0.70000000   0.3000000  -0.1900000   1.0400000   0.7000000   0.7000000 0.800000
207 -1.30000000   0.7000000   1.2600000   1.7100000 -0.3000000   0.3000000 3.100000
208 -0.30000000   1.0000000   1.2400000   1.4000000   0.0000000   0.3000000 1.300000
209  0.30000000   1.3000000   1.3800000   1.7100000   0.0000000 -1.0000000 1.300000
210  1.00000000   1.3000000  -1.2000000  -0.7800000 -0.7000000   0.0000000 1.300000
211  0.00000000   1.0000000  -0.0100000   0.2800000   1.3000000   0.0000000 3.100000
212  1.70000000   1.7000000   1.5300000   1.2800000   0.3000000   0.0000000 3.100000
213 -0.30000000   1.0000000  -1.0500000   0.2900000   0.3000000   2.7000000 1.300000
214 -0.30000000   1.0000000   0.1700000   0.9700000 -0.3000000   0.0000000 3.100000
215 -1.30000000   0.3000000   1.6100000   1.7100000   1.0000000   2.3000000 3.100000
216  1.00000000   1.3000000  -0.0700000   0.6700000   0.0000000 -1.0000000 1.000000
217  1.70000000   1.3000000   1.1600000   0.9600000   0.0000000 -0.3000000 3.000000
218 -0.70000000   0.0000000   1.5300000   0.6500000 -1.3000000 -0.3000000 3.000000
219 -0.41269192   0.8447977   0.3163209   0.9025117 -0.3208913   0.3926552 1.753575
220  2.00000000   1.3000000  -0.6900000  -0.9000000   0.0000000 -0.3000000 1.000000
222 -0.70000000   0.0000000  -0.4400000   0.0300000 -0.7000000   0.0000000 3.100000
223  0.70000000   1.3000000   2.0200000   1.5900000   0.0000000   2.0000000 1.300000
224  0.00000000   1.0000000   1.0300000   1.0000000   0.0000000 -1.0000000 1.300000
225  0.30000000   1.7000000   1.1300000   1.3900000   1.7000000   2.3000000 3.100000
226  0.00000000   1.3000000   0.6600000   0.9700000 -1.0000000 -0.7000000 3.100000
228 -0.11222345   1.2872673   1.0044066   0.8061191 -0.1814728 -0.3374685 1.937584
229 -1.00000000   1.0000000   0.6500000   1.0300000   1.0000000 -0.7000000 3.100000
230 -1.00000000   1.0000000   1.3800000   0.6400000   0.3000000 -0.7000000 3.100000
231  0.00000000   0.3000000  -0.7000000   0.2900000 -1.3000000 -1.3000000 3.100000
234  0.00000000   0.7000000   0.7700000   1.0300000 -0.3000000 -0.3000000 0.800000
235  0.30000000   1.3000000   1.1300000   1.0300000 -0.7000000   0.3000000 1.300000
236  0.70000000   0.0000000   0.3000000  -0.2800000   0.3000000 -0.3000000 1.300000
237  1.30000000   1.7000000   0.9800000   0.4000000 -1.3000000   0.7000000 1.300000
238  1.30000000   1.0000000   1.2400000   0.4000000   0.7000000   0.7000000 1.300000
239  0.70000000   0.0000000  -1.0500000  -0.7800000   0.3000000 -1.0000000 3.100000
241 -0.09916293   1.0844352   0.2887386   0.6956717 -0.1530272   0.3005667 2.201543
        tn50 Target
3   -0.3000000      0
6    0.7000000      1
7    1.3000000      1
8    0.7000000      1
9    0.3000000      1
11   0.3000000      1
13   0.0000000      1
```

```
15    0.3000000        1
16    0.7000000        1
19    0.7000000        0
22   -0.3000000        1
24    0.7000000        1
26   -0.3000000        1
27    0.3000000        1
28    1.3000000        1
33    0.0000000        1
36    0.7000000        1
40    0.3000000        1
45    0.0000000        1
46    0.3000000        0
47    1.3000000        0
48    0.3000000        0
49    0.3000000        0
50    1.0000000        1
51    0.0000000        0
52    0.0000000        0
53   -0.7000000        0
54    0.3000000        0
55   -1.0000000        0
56    3.0000000        0
57    1.0000000        1
65    0.7000000        1
68    0.7000000        1
75   -0.3000000        1
80    0.3000000        1
85    1.0000000        1
95   -0.3000000        1
101   0.3000000        1
103   0.7000000        1
108   0.3000000        1
113   1.3000000        1
115   0.7000000        1
117   0.3000000        1
118   0.0000000        1
124   0.3000000        1
128   1.0000000        0
131   0.7000000        1
132   0.3000000        1
133   0.0000000        1
145   1.3000000        1
150   0.0000000        1
151  -0.3000000        1
153   1.0000000        1
155   0.3000000        1
163   0.7000000        1
167   0.0000000        1
171   0.0000000        1
172   0.3000000        0
173   0.3000000        0
178   1.0000000        0
180   1.0000000        0
```

```
181   0.7000000       0
182   0.7000000       0
183   1.0000000       0
184   0.7000000       0
185   0.7000000       0
186   0.7000000       0
187   0.3000000       0
189   0.3000000       0
190   0.7000000       0
191   1.7000000       0
192  -0.3000000       0
193   0.0000000       0
195   0.0000000       0
196   0.3000000       0
197   2.0000000       0
199   0.3000000       1
200   1.3000000       0
201   1.0000000       0
202   1.7000000       1
203   0.3000000       0
204   0.0000000       0
205   0.3000000       1
206  -0.3000000       0
207   0.7000000       0
208   1.0000000       0
209   0.7000000       0
210   0.3000000       0
211   0.7000000       0
212   1.3000000       0
213  -0.3000000       0
214   0.0000000       0
215  -0.7000000       0
216   0.7000000       0
217   0.7000000       0
218   0.3000000       1
219   0.4238885       0
220   1.0000000       0
222   0.7000000       0
223   0.7000000       0
224   0.7000000       0
225  -0.3000000       0
226   1.7000000       0
228   0.5523615       0
229   0.3000000       0
230   1.0000000       0
231   0.7000000       1
234   1.0000000       1
235   1.0000000       0
236   0.7000000       1
237   0.0000000       0
238   1.0000000       0
239   0.3000000       0
241   0.1655717       0
```

```r
# Agregar la asignación de clústeres como columna en los datos originales
numeric_vars_def$cluster <- clusters

# Visualizar los datos con la columna de clúster
print("Datos originales con asignación de clúster:")
```

[1] "Datos originales con asignación de clúster:"

```r
print(head(numeric_vars_def))
```

```
    ag ptg19 sex_Woman el_Elementary el_High.School el_Secondary
1 26.9 38.23         1             0              1            0
2 60.5 26.34         0             0              0            0
3 33.1 18.24         1             0              0            0
4 44.6 24.14         1             0              0            0
5 61.9 32.80         0             0              0            0
6 43.0 32.70         1             0              0            0
  el_Specialist.Master el_University.Deg. spec1_Hospitalization
1                    0                  0                     0
2                    1                  0                     1
3                    0                  0                     0
4                    0                  1                     0
5                    1                  0                     1
6                    1                  0                     0
  spec1_Mild.Moderated spec1_UCI tn46 tn52 tn36 tn38 tn40 tn42 tn22 tn44 tn14
1                    1         0  0.3  1.0 -1.0 -0.3  0.0 -1.3  0.4 -0.7 -1.7
2                    0         0  1.0  0.0  0.0  0.3  0.0  0.0  0.4 -1.3  1.0
3                    1         0  0.7  2.0  1.0  0.3  1.0  0.3 -0.5 -1.0  0.3
4                    1         0  0.0 -0.6  0.3  0.0  1.0  1.3 -0.9  1.0 -0.3
5                    0         0  0.0  0.0  0.3 -0.3  0.3 -0.3 -0.2 -0.3  1.3
6                    1         0  0.3  0.4 -0.7 -0.3  0.3 -0.3  1.0  0.0  0.7
  tn24 tn12   tn6  tn30 tn34  tn8 tn48 tn50 Target cluster
1 -3.0 -2.0 -2.06 -2.30 -1.0  2.0  1.0 -1.7      1       1
2  0.0  0.0  0.86  0.70  0.3 -0.3  3.1  1.3      1       1
3 -1.0  1.3  1.70  0.32 -1.0  0.3  3.1 -0.3      0       2
4 -0.7 -0.3 -0.82 -0.43 -1.0 -0.7  1.3 -0.3      1       1
5 -0.3  1.3 -0.57  0.40  0.3 -0.3  3.1 -0.7      1       1
6 -0.7  0.0  1.38  1.71  0.0  2.7  1.3  0.7      1       2
```

```r
# Opcional: Resumen por clúster
cluster_summary <- aggregate(. ~ cluster, data = numeric_vars_def, mean)
print("Resumen por clúster:")
```

[1] "Resumen por clúster:"

```r
print(cluster_summary)
```

```
  cluster       ag    ptg19 sex_Woman el_Elementary el_High.School el_Secondary
1       1 49.16929 28.42748 0.7716535    0.07874016      0.4094488   0.07086614
2       2 48.21404 26.79841 0.8333333    0.00000000      0.2368421   0.00000000
  el_Specialist.Master el_University.Deg. spec1_Hospitalization
1           0.03937008         0.3937008            0.2677165
2           0.27192982         0.3596491            0.0000000
  spec1_Mild.Moderated  spec1_UCI       tn46        tn52        tn36       tn38
1            0.6692913 0.04724409 -0.3299213 -0.02440945 -0.50448819 -0.3881890
2            0.9385965 0.00000000  0.1521071  0.07521009  0.03924876  0.1360448
```

```
          tn40        tn42       tn22         tn44        tn14         tn24        tn12
1 -0.5763780 -1.0795276 0.1574803 -0.67165354 -0.02519685 -0.7598425 -0.2503937
2  0.1239833 -0.3298838 0.6749159 -0.01759136  0.89919242 -0.1791586  0.8247061
           tn6        tn30       tn34         tn8       tn48         tn50       Target
1 -0.8196063 -0.4996850 -0.5622047 -0.2188976 1.319685 -0.01732283 0.9685039
2  0.5199076  0.6482834 -0.1250473 -0.0038969 1.953445  0.50826159 0.4473684
```

```r
# Opcional: Agregar rectángulos al dendrograma para visualizar los clústeres
plot(hclust_model, main = "Dendrograma con clústeres")
rect.hclust(hclust_model, k = 2, border = "red")  # Cambia "k" si es necesario
```

Dendrograma con clústeres
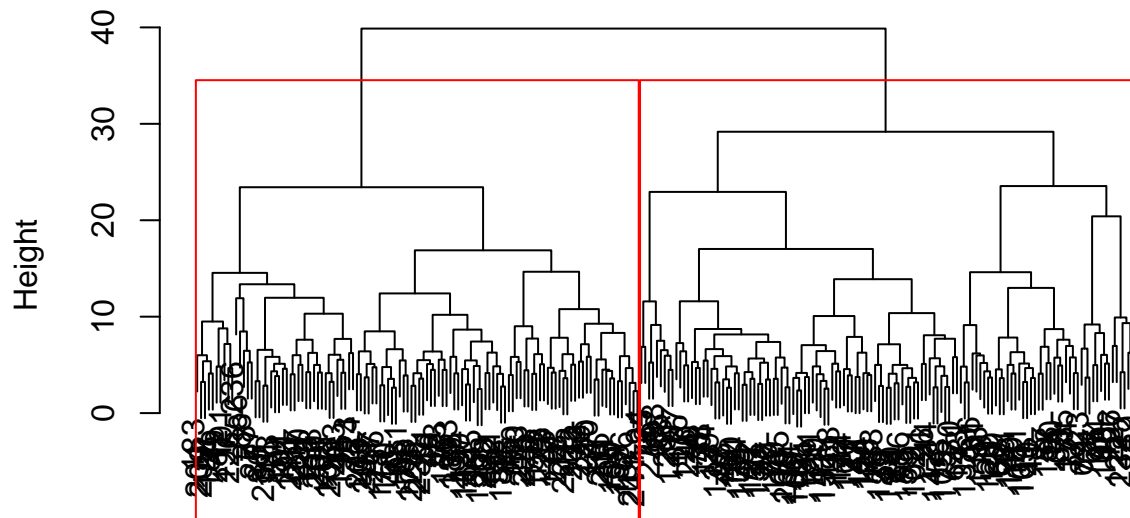
dist_matrix
hclust (*, "ward.D2")

La organización por clústeres no parece ser de utilidad inicial.

Estadística descriptiva e inferial básica.

Vamos a seleccionar las variables que son factores (reales) y las vamos a contrastar con chi-cuadrado

```r
df_chi <- df_def_clean %>%
  dplyr::select(where(~ is.numeric(.) && all(. %in% c(0, 1)))) %>%
  mutate(Target = as.factor(df_def_clean$Target))  # Agregar Target como factor


# -------------------------------------------------------------------------------
# FILTRADO POR FRECUENCIA Y PRUEBAS DE CHI-CUADRADO
# -------------------------------------------------------------------------------


# 1. Calcular las frecuencias mínimas para cada variable dummy
low_freq_vars <- sapply(names(df_chi), function(var) {
  freq_table <- table(df_chi[[var]])  # Calcular la tabla de frecuencias
  min(freq_table)  # Obtener la frecuencia mínima para cada variable
})
```

```r
# 2. Definir el umbral de frecuencia mínima
threshold <- 5  # Cambiar el umbral si es necesario

# 3. Seleccionar variables con frecuencias mínimas mayores o iguales al umbral
valid_vars <- names(low_freq_vars[low_freq_vars >= threshold])

# 4. Filtrar el dataset para mantener solo las variables con frecuencias aceptables
df_chi_filtered <- df_chi %>%
  dplyr::select(all_of(valid_vars))  # Seleccionar las variables válidas

# 5. Realizar pruebas de chi-cuadrado para las variables filtradas contra Target
chi_results_filtered <- sapply(names(df_chi_filtered), function(var) {
  table_var <- table(df_chi_filtered[[var]], df_def$Target)  # Crear tabla de contingencia
  chi_test <- chisq.test(table_var)  # Realizar prueba de chi-cuadrado
  return(chi_test$p.value)  # Extraer el valor p
})

# 6. Convertir los resultados a un data.frame
chi_results_filtered_df <- data.frame(
  Variable = names(chi_results_filtered),  # Nombre de las variables
  P_Value = chi_results_filtered  # Valores p
)

# 7. Ordenar los resultados por valor p
chi_results_filtered_df <- chi_results_filtered_df %>%
  arrange(P_Value)

# -------------------------------------------------------------------------------
# RESULTADOS Y VISUALIZACIÓN
# -------------------------------------------------------------------------------

# Mostrar el resumen de los resultados
print(chi_results_filtered_df)
```

```
                                  Variable      P_Value
Target                              Target 2.866752e-53
spec1_Hospitalization spec1_Hospitalization 2.176275e-04
el_High.School                el_High.School 4.464665e-04
el_Specialist.Master    el_Specialist.Master 6.144517e-04
spec1_Mild.Moderated    spec1_Mild.Moderated 2.872846e-02
el_University.Deg.          el_University.Deg. 2.598910e-01
spec1_UCI                        spec1_UCI 2.810929e-01
el_Elementary                el_Elementary 3.560693e-01
sex_Woman                        sex_Woman 9.553274e-01
el_Secondary                  el_Secondary 9.987447e-01
```

```r
# Guardar los resultados en un archivo CSV si es necesario
# write.csv(chi_results_filtered_df, "chi_squared_results_filtered.csv", row.names = FALSE)

# Opcional: Filtrar variables significativas (p < 0.05)
significant_vars_chi <- chi_results_filtered_df %>%
  filter(P_Value < 0.05)

# Mostrar las variables significativas
```

```
print(significant_vars_chi)
```

```
                             Variable      P_Value
Target                         Target 2.866752e-53
spec1_Hospitalization spec1_Hospitalization 2.176275e-04
el_High.School             el_High.School 4.464665e-04
el_Specialist.Master   el_Specialist.Master 6.144517e-04
spec1_Mild.Moderate    spec1_Mild.Moderate 2.872846e-02
```

Sobre Chi-cuadrado Un p-valor bajo significa que la distribución de las categorías de la variable explicativa difiere significativamente entre las categorías de Target. Es decir, las frecuencias de las categorías no son independientes entre sí, lo que sugiere una relación entre la variable explicativa y Target.

Por ejemplo, si spec1_Hospitalization tiene un p-valor bajo, esto implica que las frecuencias de hospitalización están distribuidas de manera diferente según la categoría de Target (No_LC vs. LC_Cog), lo que podría ser relevante para el modelo.

```
# Librerías necesarias

# Seleccionar variables numéricas continuas (excluyendo las dummy)
numeric_vars_log <- df_def_clean %>%
  dplyr::select(where(is.numeric)) %>%   # Selecciona variables numéricas
  dplyr::select_if(~ length(unique(.)) > 2)  # Excluye las dummies (solo deja aquellas con más de 2 val

numeric_vars_log$Target <- as.factor(df_def_clean$Target)

# Ajustar el modelo de regresión logística
logistic_model <- glm(Target ~ ., data = numeric_vars_log, family = binomial)

# Resumen del modelo
summary(logistic_model)
```

```
Call:
glm(formula = Target ~ ., family = binomial, data = numeric_vars_log)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.32422    1.42136  -1.635 0.102004
ag           0.04644    0.02229   2.084 0.037188 *
ptg19        0.06075    0.03643   1.668 0.095369 .
tn46        -0.27194    0.26750  -1.017 0.309334
tn52         0.40342    0.34525   1.169 0.242605
tn36         0.56483    0.34420   1.641 0.100803
tn38        -0.21690    0.40333  -0.538 0.590730
tn40        -0.56926    0.38874  -1.464 0.143088
tn42        -0.37778    0.29118  -1.297 0.194483
tn22        -0.12377    0.23784  -0.520 0.602794
tn44        -0.26662    0.25533  -1.044 0.296398
tn14        -0.15135    0.40507  -0.374 0.708666
tn24        -0.29628    0.31218  -0.949 0.342588
tn12        -1.24769    0.35217  -3.543 0.000396 ***
tn6         -0.44161    0.28455  -1.552 0.120672
tn30        -0.43698    0.31917  -1.369 0.170971
tn34         0.16351    0.30306   0.540 0.589526
tn8          0.09556    0.22970   0.416 0.677394
```

```
tn48            0.03403    0.22367   0.152 0.879076
tn50            0.31727    0.36141   0.878 0.380020
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 284.89  on 240  degrees of freedom
Residual deviance: 168.21  on 221  degrees of freedom
AIC: 208.21

Number of Fisher Scoring iterations: 6
```

```r
plot(logistic_model)
```


Residuals vs Fitted

Q–Q Residuals

Theoretical Quantiles
glm(Target ~ .)

Scale–Location

Predicted values
glm(Target ~ .)

Residuals vs Leverage

glm(Target ~ .)
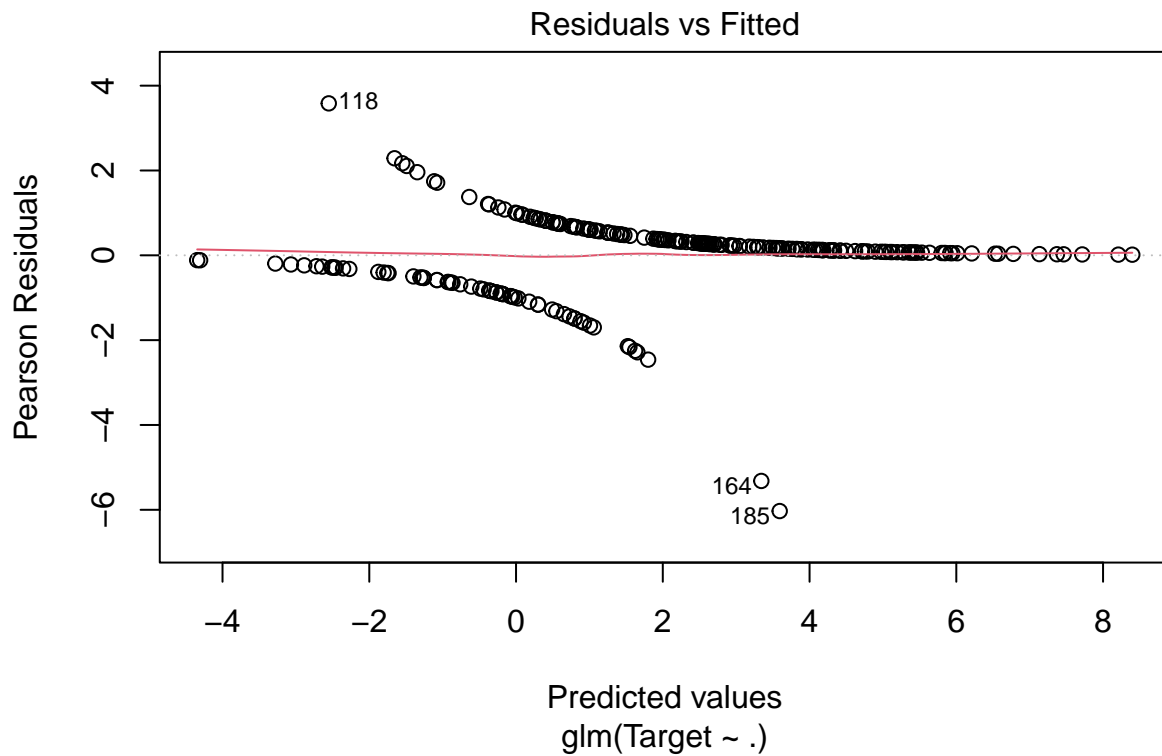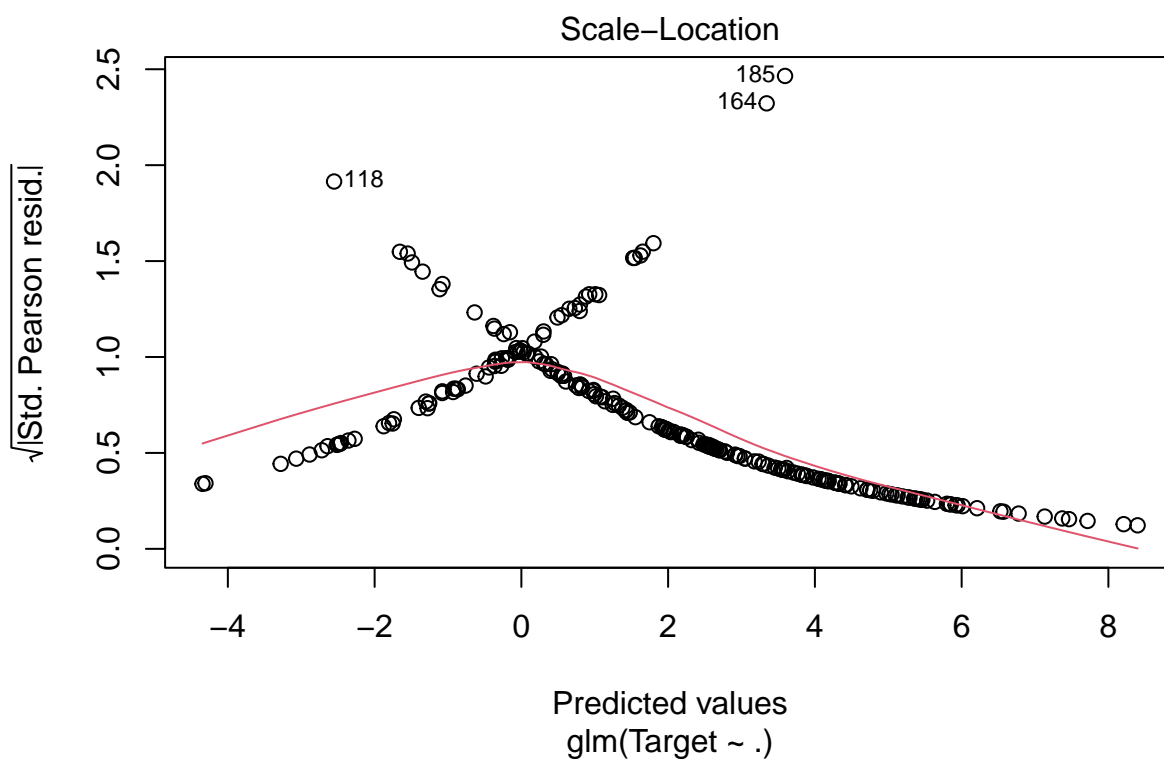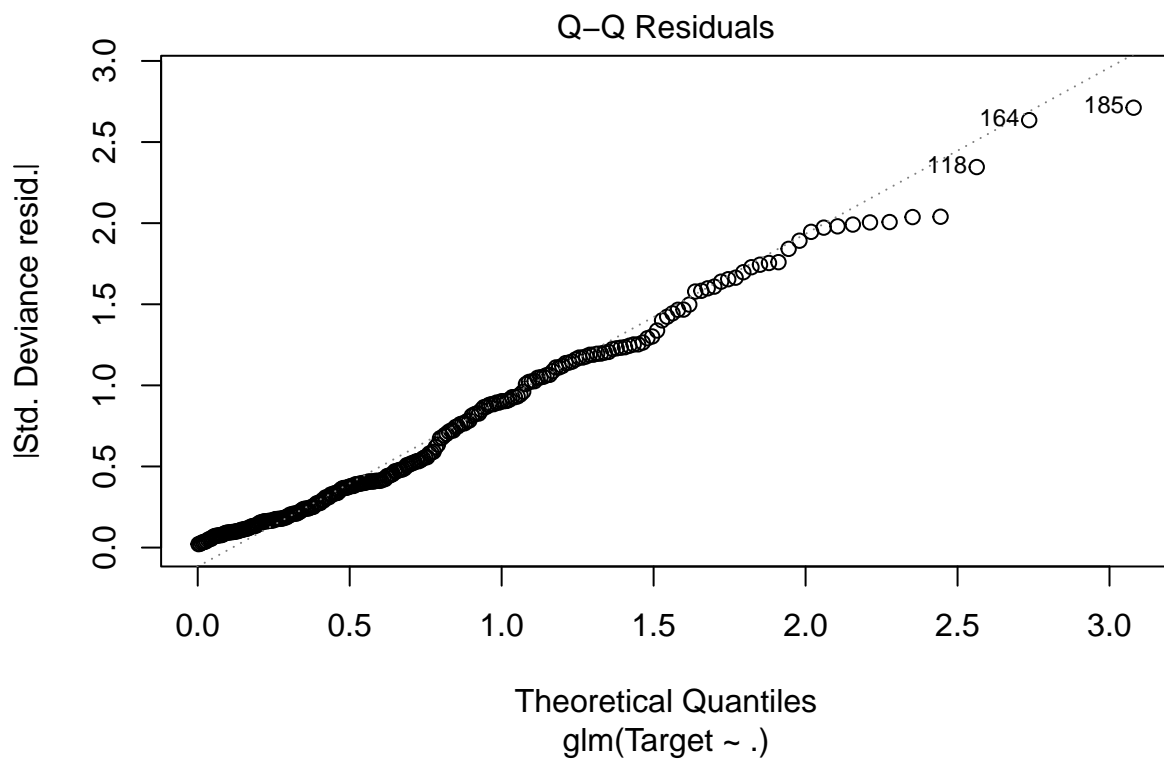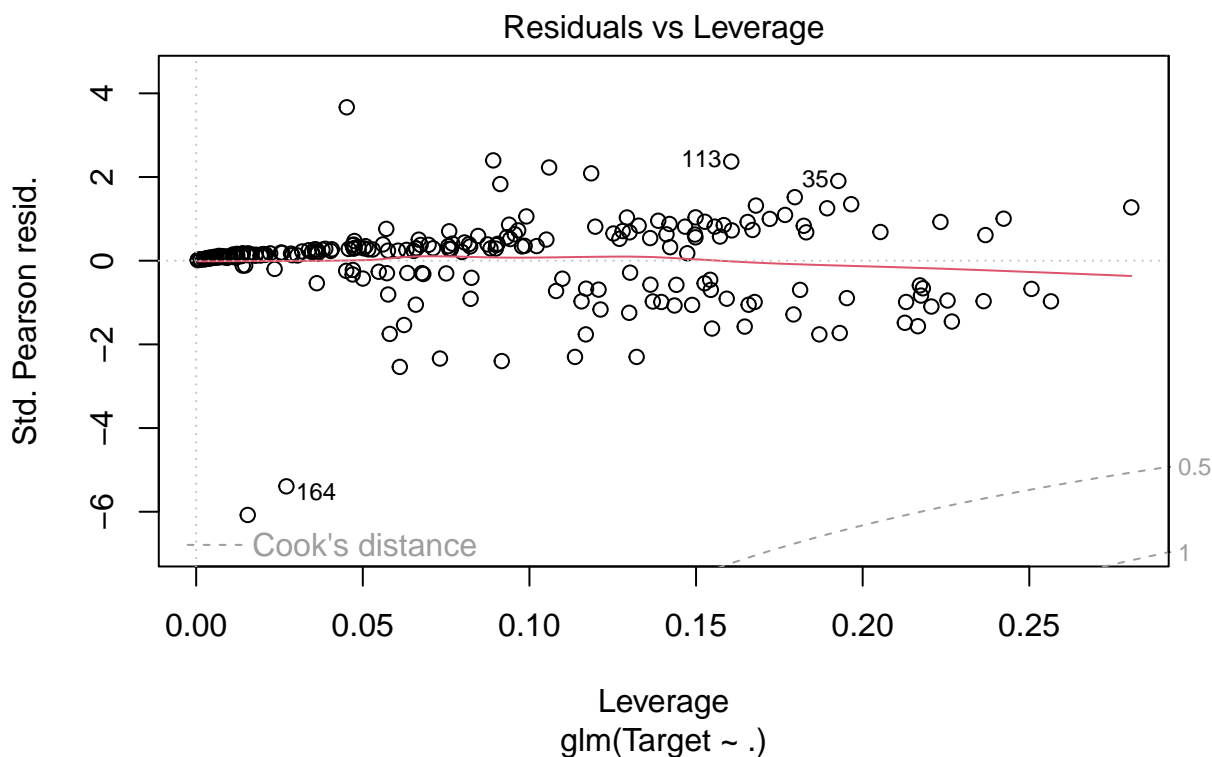
```r
lda_model <- lda(Target ~ ., data = numeric_vars_log)
print(lda_model)

Call:
lda(Target ~ ., data = numeric_vars_log)

Prior probabilities of groups:
        0         1
0.2780083 0.7219917

Group means:
        ag    ptg19       tn46        tn52       tn36       tn38       tn40
0 47.47761 25.81204  0.2259733 -0.06904551  0.1190203  0.2911807  0.2333448
1 49.19483 28.36726 -0.2281609  0.05804598 -0.3883333 -0.3063218 -0.4293103
        tn42      tn22        tn44      tn14       tn24        tn12        tn6
0 -0.1538321 0.6035883  0.01932217 0.9523573  0.07277495  0.98681344  0.5961114
1 -0.9448276 0.3247126 -0.50919540 0.2040230 -0.70000000 -0.02241379 -0.4871264
        tn30       tn34        tn8      tn48      tn50
0   0.6772284 -0.1888864 -0.09170517 1.837205 0.4618182
1 -0.2007471 -0.4195402 -0.12701149 1.535632 0.1425287


Coefficients of linear discriminants:
            LD1
ag      0.04098263
ptg19   0.02624177
tn46   -0.25344260
tn52    0.27158291
tn36    0.40766393
tn38   -0.16422852
tn40   -0.29297120
```

```
tn42   -0.18203216
tn22   -0.06666090
tn44   -0.05936998
tn14   -0.18897618
tn24   -0.08990566
tn12   -0.61817323
tn6    -0.23181184
tn30   -0.18537675
tn34    0.16492408
tn8     0.09127756
tn48    0.08539564
tn50    0.04940253
```

```
plot(lda_model)
```


group 0

group 1
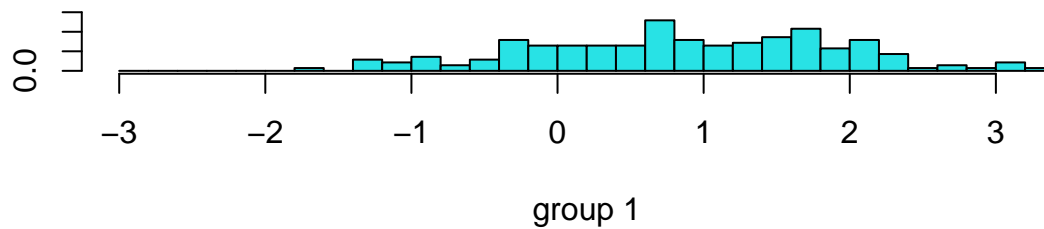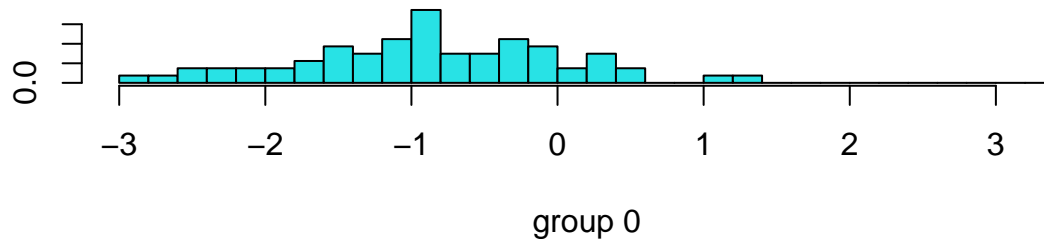
```
# Realizar el test de Kruskal-Wallis para cada variable
kruskal_results <- sapply(colnames(numeric_vars_log)[-which(colnames(numeric_vars_log) == "Target")], fu
  kruskal.test(as.formula(paste(var, "~ Target")), data = numeric_vars_log)$p.value
})

# Convertir los resultados a un data.frame
kruskal_results_df <- data.frame(
  Variable = names(kruskal_results),
  P_Value = kruskal_results
)

# Ordenar por valor p
kruskal_results_df <- kruskal_results_df %>% arrange(P_Value)

# Mostrar los resultados
print(kruskal_results_df)
```

```
      Variable      P_Value
tn12      tn12 1.346705e-15
```

```
tn14        tn14 7.077679e-11
tn6          tn6 5.015610e-10
tn30        tn30 9.667375e-10
tn42        tn42 2.663421e-08
tn40        tn40 1.865985e-07
tn24        tn24 3.572774e-07
tn38        tn38 2.263308e-06
tn50        tn50 8.166312e-04
tn36        tn36 8.963381e-04
tn44        tn44 1.180907e-03
tn46        tn46 2.347338e-03
tn48        tn48 4.823635e-03
ptg19      ptg19 3.622734e-02
tn22        tn22 3.688904e-02
tn34        tn34 6.026432e-02
tn52        tn52 2.036706e-01
ag            ag 2.757166e-01
tn8          tn8 7.211850e-01
```

```r
# Opcional: Filtrar variables significativas (p < 0.05)
significant_vars_krus <- kruskal_results_df %>%
  filter(P_Value < 0.05)
```

Generamos los modelos

```r
# Suponiendo que `df_def_clean` ya contiene las características y la columna `Target`
# Dividir el dataset en entrenamiento (80%) y prueba (20%)
set.seed(2443)  # Para reproducibilidad
#semillas interesantes: (0.8) 1234, 1237, 1238, 1339
#semillas interesantes: (0.7) 1339, 1441 por el recall
#semillas interesantes (0.75) 1234
#Tras mejoras (que es evitar el rol del desbalance y que la clase positiva sea 1), es decir,
#identificar los casos de long covid y 0.75: 1236, 1237, #1239
# tras mejoras y 0,8: 2435, 2429, 2433, 2435

trainIndex <- createDataPartition(df_def_clean$Target, p = 0.8, list = FALSE)
train_data <- df_def_clean[trainIndex, ]
test_data <- df_def_clean[-trainIndex, ]

# Definir la columna objetivo para entrenamiento y prueba
train_target <- train_data$Target
test_target <- test_data$Target

# Eliminar la columna objetivo de los datos de entrenamiento y prueba
train_data <- train_data %>% dplyr::select(-Target)
test_data <- test_data %>% dplyr::select(-Target)

# Convertir los datasets a matrices para XGBoost
train_matrix <- as.matrix(train_data)
test_matrix <- as.matrix(test_data)

# Crear DMatrix para XGBoost
dtrain <- xgb.DMatrix(data = train_matrix, label = train_target)
dtest <- xgb.DMatrix(data = test_matrix, label = test_target)
```

```r
# Calcular el peso para la clase positiva
num_negativos <- sum(train_target == 0)  # Casos de la clase 0
num_positivos <- sum(train_target == 1)  # Casos de la clase 1

scale_pos_weight <- num_negativos / num_positivos

params <- list(
  objective = "binary:logistic",  # Configuración para clasificación binaria
  eval_metric = "logloss",        # Métrica de evaluación para clasificación binaria
    scale_pos_weight = scale_pos_weight  # Peso para manejar el desbalance de clases

)

# Entrenar el modelo de clasificación binaria con XGBoost
modelo_xgb <- xgboost(data = dtrain, params = params, nrounds = 500)
```

```
[1] train-logloss:0.563283
[2] train-logloss:0.481515
[3] train-logloss:0.419843
[4] train-logloss:0.372299
[5] train-logloss:0.333977
[6] train-logloss:0.302680
[7] train-logloss:0.273890
[8] train-logloss:0.254324
[9] train-logloss:0.236549
[10]    train-logloss:0.217855
[11]    train-logloss:0.204182
[12]    train-logloss:0.190731
[13]    train-logloss:0.180605
[14]    train-logloss:0.169654
[15]    train-logloss:0.158723
[16]    train-logloss:0.150307
[17]    train-logloss:0.142247
[18]    train-logloss:0.135132
[19]    train-logloss:0.128481
[20]    train-logloss:0.123429
[21]    train-logloss:0.119813
[22]    train-logloss:0.115978
[23]    train-logloss:0.112281
[24]    train-logloss:0.109127
[25]    train-logloss:0.106008
[26]    train-logloss:0.102319
[27]    train-logloss:0.098278
[28]    train-logloss:0.096241
[29]    train-logloss:0.093938
[30]    train-logloss:0.090129
[31]    train-logloss:0.088019
[32]    train-logloss:0.086471
[33]    train-logloss:0.084705
[34]    train-logloss:0.083406
[35]    train-logloss:0.081934
[36]    train-logloss:0.080555
[37]    train-logloss:0.079598
[38]    train-logloss:0.077407
```

```
[39]    train-logloss:0.076015
[40]    train-logloss:0.074463
[41]    train-logloss:0.073296
[42]    train-logloss:0.071826
[43]    train-logloss:0.070887
[44]    train-logloss:0.069892
[45]    train-logloss:0.068948
[46]    train-logloss:0.067955
[47]    train-logloss:0.067070
[48]    train-logloss:0.065803
[49]    train-logloss:0.065132
[50]    train-logloss:0.064404
[51]    train-logloss:0.063838
[52]    train-logloss:0.063135
[53]    train-logloss:0.062080
[54]    train-logloss:0.060893
[55]    train-logloss:0.060218
[56]    train-logloss:0.059754
[57]    train-logloss:0.059384
[58]    train-logloss:0.058700
[59]    train-logloss:0.058181
[60]    train-logloss:0.057617
[61]    train-logloss:0.057199
[62]    train-logloss:0.056499
[63]    train-logloss:0.056234
[64]    train-logloss:0.055910
[65]    train-logloss:0.055626
[66]    train-logloss:0.055248
[67]    train-logloss:0.054935
[68]    train-logloss:0.054247
[69]    train-logloss:0.053829
[70]    train-logloss:0.053391
[71]    train-logloss:0.053000
[72]    train-logloss:0.052614
[73]    train-logloss:0.052252
[74]    train-logloss:0.051871
[75]    train-logloss:0.051650
[76]    train-logloss:0.051369
[77]    train-logloss:0.050974
[78]    train-logloss:0.050654
[79]    train-logloss:0.050366
[80]    train-logloss:0.050112
[81]    train-logloss:0.049789
[82]    train-logloss:0.049541
[83]    train-logloss:0.049309
[84]    train-logloss:0.048993
[85]    train-logloss:0.048754
[86]    train-logloss:0.048456
[87]    train-logloss:0.048270
[88]    train-logloss:0.047914
[89]    train-logloss:0.047611
[90]    train-logloss:0.047347
[91]    train-logloss:0.047108
[92]    train-logloss:0.046852
```

```
[93]     train-logloss:0.046579
[94]     train-logloss:0.046270
[95]     train-logloss:0.045968
[96]     train-logloss:0.045740
[97]     train-logloss:0.045516
[98]     train-logloss:0.045373
[99]     train-logloss:0.045148
[100]    train-logloss:0.044957
[101]    train-logloss:0.044806
[102]    train-logloss:0.044591
[103]    train-logloss:0.044443
[104]    train-logloss:0.044253
[105]    train-logloss:0.044015
[106]    train-logloss:0.043913
[107]    train-logloss:0.043715
[108]    train-logloss:0.043514
[109]    train-logloss:0.043330
[110]    train-logloss:0.043235
[111]    train-logloss:0.042902
[112]    train-logloss:0.042773
[113]    train-logloss:0.042667
[114]    train-logloss:0.042491
[115]    train-logloss:0.042202
[116]    train-logloss:0.042015
[117]    train-logloss:0.041874
[118]    train-logloss:0.041696
[119]    train-logloss:0.041514
[120]    train-logloss:0.041348
[121]    train-logloss:0.041212
[122]    train-logloss:0.041066
[123]    train-logloss:0.041010
[124]    train-logloss:0.040844
[125]    train-logloss:0.040744
[126]    train-logloss:0.040565
[127]    train-logloss:0.040406
[128]    train-logloss:0.040269
[129]    train-logloss:0.040148
[130]    train-logloss:0.040045
[131]    train-logloss:0.039878
[132]    train-logloss:0.039734
[133]    train-logloss:0.039604
[134]    train-logloss:0.039499
[135]    train-logloss:0.039464
[136]    train-logloss:0.039349
[137]    train-logloss:0.039250
[138]    train-logloss:0.039187
[139]    train-logloss:0.039129
[140]    train-logloss:0.039009
[141]    train-logloss:0.038973
[142]    train-logloss:0.038915
[143]    train-logloss:0.038868
[144]    train-logloss:0.038751
[145]    train-logloss:0.038704
[146]    train-logloss:0.038678
```

```
[147]    train-logloss:0.038569
[148]    train-logloss:0.038528
[149]    train-logloss:0.038420
[150]    train-logloss:0.038394
[151]    train-logloss:0.038334
[152]    train-logloss:0.038238
[153]    train-logloss:0.038196
[154]    train-logloss:0.038135
[155]    train-logloss:0.038014
[156]    train-logloss:0.037953
[157]    train-logloss:0.037904
[158]    train-logloss:0.037819
[159]    train-logloss:0.037702
[160]    train-logloss:0.037651
[161]    train-logloss:0.037574
[162]    train-logloss:0.037494
[163]    train-logloss:0.037416
[164]    train-logloss:0.037383
[165]    train-logloss:0.037335
[166]    train-logloss:0.037231
[167]    train-logloss:0.037178
[168]    train-logloss:0.037099
[169]    train-logloss:0.037071
[170]    train-logloss:0.036991
[171]    train-logloss:0.036920
[172]    train-logloss:0.036874
[173]    train-logloss:0.036822
[174]    train-logloss:0.036783
[175]    train-logloss:0.036683
[176]    train-logloss:0.036584
[177]    train-logloss:0.036536
[178]    train-logloss:0.036493
[179]    train-logloss:0.036471
[180]    train-logloss:0.036420
[181]    train-logloss:0.036401
[182]    train-logloss:0.036355
[183]    train-logloss:0.036324
[184]    train-logloss:0.036294
[185]    train-logloss:0.036262
[186]    train-logloss:0.036229
[187]    train-logloss:0.036163
[188]    train-logloss:0.036164
[189]    train-logloss:0.036084
[190]    train-logloss:0.036043
[191]    train-logloss:0.035999
[192]    train-logloss:0.035979
[193]    train-logloss:0.035945
[194]    train-logloss:0.035906
[195]    train-logloss:0.035876
[196]    train-logloss:0.035847
[197]    train-logloss:0.035812
[198]    train-logloss:0.035797
[199]    train-logloss:0.035770
[200]    train-logloss:0.035710
```

```
[201]    train-logloss:0.035680
[202]    train-logloss:0.035668
[203]    train-logloss:0.035625
[204]    train-logloss:0.035599
[205]    train-logloss:0.035572
[206]    train-logloss:0.035579
[207]    train-logloss:0.035563
[208]    train-logloss:0.035518
[209]    train-logloss:0.035489
[210]    train-logloss:0.035426
[211]    train-logloss:0.035398
[212]    train-logloss:0.035355
[213]    train-logloss:0.035357
[214]    train-logloss:0.035302
[215]    train-logloss:0.035196
[216]    train-logloss:0.035133
[217]    train-logloss:0.035093
[218]    train-logloss:0.035072
[219]    train-logloss:0.035070
[220]    train-logloss:0.035028
[221]    train-logloss:0.035008
[222]    train-logloss:0.035006
[223]    train-logloss:0.034965
[224]    train-logloss:0.034946
[225]    train-logloss:0.034905
[226]    train-logloss:0.034877
[227]    train-logloss:0.034848
[228]    train-logloss:0.034835
[229]    train-logloss:0.034805
[230]    train-logloss:0.034777
[231]    train-logloss:0.034729
[232]    train-logloss:0.034731
[233]    train-logloss:0.034700
[234]    train-logloss:0.034661
[235]    train-logloss:0.034628
[236]    train-logloss:0.034584
[237]    train-logloss:0.034557
[238]    train-logloss:0.034497
[239]    train-logloss:0.034464
[240]    train-logloss:0.034434
[241]    train-logloss:0.034407
[242]    train-logloss:0.034366
[243]    train-logloss:0.034339
[244]    train-logloss:0.034313
[245]    train-logloss:0.034303
[246]    train-logloss:0.034310
[247]    train-logloss:0.034295
[248]    train-logloss:0.034260
[249]    train-logloss:0.034231
[250]    train-logloss:0.034209
[251]    train-logloss:0.034212
[252]    train-logloss:0.034185
[253]    train-logloss:0.034171
[254]    train-logloss:0.034139
```

```
[255]    train-logloss:0.034113
[256]    train-logloss:0.034121
[257]    train-logloss:0.034091
[258]    train-logloss:0.034091
[259]    train-logloss:0.034061
[260]    train-logloss:0.034026
[261]    train-logloss:0.033999
[262]    train-logloss:0.033974
[263]    train-logloss:0.033942
[264]    train-logloss:0.033949
[265]    train-logloss:0.033923
[266]    train-logloss:0.033902
[267]    train-logloss:0.033878
[268]    train-logloss:0.033843
[269]    train-logloss:0.033811
[270]    train-logloss:0.033826
[271]    train-logloss:0.033801
[272]    train-logloss:0.033773
[273]    train-logloss:0.033741
[274]    train-logloss:0.033751
[275]    train-logloss:0.033739
[276]    train-logloss:0.033709
[277]    train-logloss:0.033698
[278]    train-logloss:0.033702
[279]    train-logloss:0.033687
[280]    train-logloss:0.033658
[281]    train-logloss:0.033660
[282]    train-logloss:0.033630
[283]    train-logloss:0.033599
[284]    train-logloss:0.033585
[285]    train-logloss:0.033548
[286]    train-logloss:0.033554
[287]    train-logloss:0.033522
[288]    train-logloss:0.033510
[289]    train-logloss:0.033513
[290]    train-logloss:0.033482
[291]    train-logloss:0.033451
[292]    train-logloss:0.033429
[293]    train-logloss:0.033419
[294]    train-logloss:0.033390
[295]    train-logloss:0.033395
[296]    train-logloss:0.033360
[297]    train-logloss:0.033364
[298]    train-logloss:0.033335
[299]    train-logloss:0.033314
[300]    train-logloss:0.033290
[301]    train-logloss:0.033258
[302]    train-logloss:0.033259
[303]    train-logloss:0.033263
[304]    train-logloss:0.033238
[305]    train-logloss:0.033206
[306]    train-logloss:0.033183
[307]    train-logloss:0.033157
[308]    train-logloss:0.033152
```

```
[309]    train-logloss:0.033131
[310]    train-logloss:0.033108
[311]    train-logloss:0.033102
[312]    train-logloss:0.033070
[313]    train-logloss:0.033049
[314]    train-logloss:0.033035
[315]    train-logloss:0.033019
[316]    train-logloss:0.033001
[317]    train-logloss:0.032988
[318]    train-logloss:0.033001
[319]    train-logloss:0.032991
[320]    train-logloss:0.032963
[321]    train-logloss:0.032969
[322]    train-logloss:0.032944
[323]    train-logloss:0.032926
[324]    train-logloss:0.032904
[325]    train-logloss:0.032909
[326]    train-logloss:0.032885
[327]    train-logloss:0.032859
[328]    train-logloss:0.032848
[329]    train-logloss:0.032852
[330]    train-logloss:0.032825
[331]    train-logloss:0.032828
[332]    train-logloss:0.032802
[333]    train-logloss:0.032780
[334]    train-logloss:0.032769
[335]    train-logloss:0.032741
[336]    train-logloss:0.032744
[337]    train-logloss:0.032716
[338]    train-logloss:0.032688
[339]    train-logloss:0.032669
[340]    train-logloss:0.032654
[341]    train-logloss:0.032647
[342]    train-logloss:0.032622
[343]    train-logloss:0.032605
[344]    train-logloss:0.032615
[345]    train-logloss:0.032592
[346]    train-logloss:0.032565
[347]    train-logloss:0.032549
[348]    train-logloss:0.032556
[349]    train-logloss:0.032536
[350]    train-logloss:0.032541
[351]    train-logloss:0.032518
[352]    train-logloss:0.032508
[353]    train-logloss:0.032483
[354]    train-logloss:0.032465
[355]    train-logloss:0.032456
[356]    train-logloss:0.032433
[357]    train-logloss:0.032413
[358]    train-logloss:0.032396
[359]    train-logloss:0.032404
[360]    train-logloss:0.032383
[361]    train-logloss:0.032369
[362]    train-logloss:0.032350
```

```
[363]    train-logloss:0.032329
[364]    train-logloss:0.032311
[365]    train-logloss:0.032305
[366]    train-logloss:0.032313
[367]    train-logloss:0.032303
[368]    train-logloss:0.032279
[369]    train-logloss:0.032284
[370]    train-logloss:0.032289
[371]    train-logloss:0.032262
[372]    train-logloss:0.032232
[373]    train-logloss:0.032223
[374]    train-logloss:0.032203
[375]    train-logloss:0.032200
[376]    train-logloss:0.032173
[377]    train-logloss:0.032153
[378]    train-logloss:0.032137
[379]    train-logloss:0.032117
[380]    train-logloss:0.032100
[381]    train-logloss:0.032107
[382]    train-logloss:0.032112
[383]    train-logloss:0.032101
[384]    train-logloss:0.032075
[385]    train-logloss:0.032073
[386]    train-logloss:0.032066
[387]    train-logloss:0.032039
[388]    train-logloss:0.032034
[389]    train-logloss:0.032009
[390]    train-logloss:0.032005
[391]    train-logloss:0.031980
[392]    train-logloss:0.031977
[393]    train-logloss:0.031973
[394]    train-logloss:0.031958
[395]    train-logloss:0.031932
[396]    train-logloss:0.031930
[397]    train-logloss:0.031932
[398]    train-logloss:0.031908
[399]    train-logloss:0.031896
[400]    train-logloss:0.031897
[401]    train-logloss:0.031870
[402]    train-logloss:0.031852
[403]    train-logloss:0.031847
[404]    train-logloss:0.031841
[405]    train-logloss:0.031814
[406]    train-logloss:0.031810
[407]    train-logloss:0.031796
[408]    train-logloss:0.031784
[409]    train-logloss:0.031760
[410]    train-logloss:0.031737
[411]    train-logloss:0.031736
[412]    train-logloss:0.031738
[413]    train-logloss:0.031717
[414]    train-logloss:0.031700
[415]    train-logloss:0.031678
[416]    train-logloss:0.031683
```

```
[417]    train-logloss:0.031663
[418]    train-logloss:0.031636
[419]    train-logloss:0.031636
[420]    train-logloss:0.031614
[421]    train-logloss:0.031596
[422]    train-logloss:0.031583
[423]    train-logloss:0.031577
[424]    train-logloss:0.031583
[425]    train-logloss:0.031572
[426]    train-logloss:0.031568
[427]    train-logloss:0.031542
[428]    train-logloss:0.031540
[429]    train-logloss:0.031515
[430]    train-logloss:0.031514
[431]    train-logloss:0.031509
[432]    train-logloss:0.031491
[433]    train-logloss:0.031493
[434]    train-logloss:0.031477
[435]    train-logloss:0.031477
[436]    train-logloss:0.031451
[437]    train-logloss:0.031427
[438]    train-logloss:0.031425
[439]    train-logloss:0.031402
[440]    train-logloss:0.031400
[441]    train-logloss:0.031379
[442]    train-logloss:0.031364
[443]    train-logloss:0.031344
[444]    train-logloss:0.031327
[445]    train-logloss:0.031316
[446]    train-logloss:0.031311
[447]    train-logloss:0.031290
[448]    train-logloss:0.031286
[449]    train-logloss:0.031278
[450]    train-logloss:0.031265
[451]    train-logloss:0.031248
[452]    train-logloss:0.031235
[453]    train-logloss:0.031230
[454]    train-logloss:0.031211
[455]    train-logloss:0.031196
[456]    train-logloss:0.031198
[457]    train-logloss:0.031204
[458]    train-logloss:0.031207
[459]    train-logloss:0.031194
[460]    train-logloss:0.031175
[461]    train-logloss:0.031150
[462]    train-logloss:0.031133
[463]    train-logloss:0.031138
[464]    train-logloss:0.031119
[465]    train-logloss:0.031123
[466]    train-logloss:0.031118
[467]    train-logloss:0.031103
[468]    train-logloss:0.031080
[469]    train-logloss:0.031077
[470]    train-logloss:0.031073
```

```
[471]    train-logloss:0.031059
[472]    train-logloss:0.031039
[473]    train-logloss:0.031041
[474]    train-logloss:0.031018
[475]    train-logloss:0.030997
[476]    train-logloss:0.030978
[477]    train-logloss:0.030966
[478]    train-logloss:0.030959
[479]    train-logloss:0.030941
[480]    train-logloss:0.030935
[481]    train-logloss:0.030941
[482]    train-logloss:0.030943
[483]    train-logloss:0.030939
[484]    train-logloss:0.030914
[485]    train-logloss:0.030904
[486]    train-logloss:0.030882
[487]    train-logloss:0.030865
[488]    train-logloss:0.030853
[489]    train-logloss:0.030835
[490]    train-logloss:0.030836
[491]    train-logloss:0.030816
[492]    train-logloss:0.030807
[493]    train-logloss:0.030807
[494]    train-logloss:0.030795
[495]    train-logloss:0.030785
[496]    train-logloss:0.030776
[497]    train-logloss:0.030756
[498]    train-logloss:0.030762
[499]    train-logloss:0.030745
[500]    train-logloss:0.030733
```

```r
# Generar predicciones en el conjunto de prueba
predicciones <- predict(modelo_xgb, test_matrix)

# Convertir las probabilidades en clases (0 o 1) usando un umbral de 0.5
predicciones_clase <- ifelse(predicciones >= 0.5, 1, 0)

# Crear y mostrar la matriz de confusión comparando predicciones con etiquetas reales
matriz_confusion <- table(Predicho = predicciones_clase, Real = test_target)
print(matriz_confusion)
```

```
        Real
Predicho  0  1
       0 10  4
       1  6 28
```

```r
# Utilizar caret para una matriz de confusión con métricas adicionales
confusion <- confusionMatrix(factor(predicciones_clase),
                             factor(test_target),
                             mode="everything",
                             positive = "1")

print(confusion)
```

```
Confusion Matrix and Statistics
```

```
          Reference
Prediction  0  1
         0 10  4
         1  6 28

               Accuracy : 0.7917
                 95% CI : (0.6501, 0.8953)
    No Information Rate : 0.6667
    P-Value [Acc > NIR] : 0.04225

                  Kappa : 0.5161

 Mcnemar's Test P-Value : 0.75183

            Sensitivity : 0.8750
            Specificity : 0.6250
         Pos Pred Value : 0.8235
         Neg Pred Value : 0.7143
              Precision : 0.8235
                 Recall : 0.8750
                     F1 : 0.8485
             Prevalence : 0.6667
         Detection Rate : 0.5833
   Detection Prevalence : 0.7083
      Balanced Accuracy : 0.7500

       'Positive' Class : 1
```

```r
#predicciones <- predict(modelo_xgb, test_matrix)

# Crear la curva ROC
roc_obj <- roc(test_target, predicciones)

# Graficar la curva ROC
plot(roc_obj, col = "blue", main = "Curva ROC", print.auc = TRUE)
```

Curva ROC
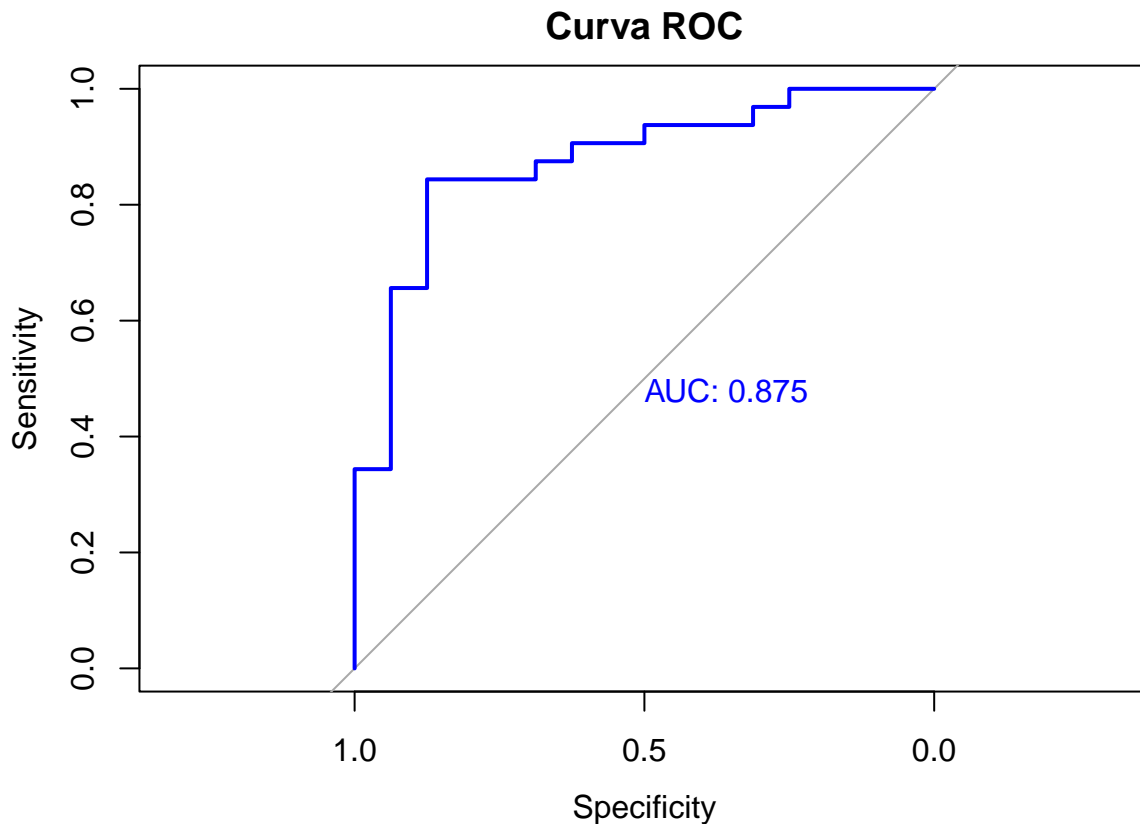

```r
# Calcular el AUC
auc_valor <- auc(roc_obj)
print(paste("El AUC es:", auc_valor))
```

```
[1] "El AUC es: 0.875"
```

```r
# Obtener la importancia de las variables
importancia <- xgb.importance(model = modelo_xgb, feature_names = colnames(train_matrix))

# Convertir la importancia en un data.frame
importancia_df <- as.data.frame(importancia)

# Opcional: ordenar las variables por importancia de mayor a menor
importancia_df <- importancia_df[order(-importancia_df$Gain), ]


# Mostrar la importancia de las variables
print(importancia)
```

```
           Feature        Gain       Cover   Frequency
           <char>       <num>       <num>       <num>
1:            tn12 0.3081574275 0.080903069 0.051162791
2:            tn30 0.0884876880 0.104197717 0.108139535
3:             tn6 0.0756919635 0.111375236 0.111627907
4:            tn22 0.0615674602 0.080054197 0.075581395
5:            tn14 0.0529809393 0.026760600 0.015116279
6:              ag 0.0512532636 0.064178246 0.072093023
7:       sex_Woman 0.0506338962 0.009351063 0.004651163
8:            tn40 0.0444053365 0.058155601 0.051162791
```

```
 9:                  tn24 0.0319256084 0.016915259 0.016279070
10:                  tn42 0.0275703809 0.113162988 0.137209302
11:                  tn44 0.0255161959 0.038014467 0.039534884
12:                  tn38 0.0221177198 0.036401584 0.041860465
13: spec1_Hospitalization 0.0196221524 0.012998049 0.003488372
14:                  tn52 0.0176664100 0.028920006 0.024418605
15:                  tn50 0.0174935906 0.012536896 0.011627907
16:                  tn46 0.0174647243 0.022108332 0.029069767
17:                  tn36 0.0165454101 0.036818706 0.038372093
18:                 ptg19 0.0159604116 0.077150720 0.102325581
19:                   tn8 0.0144680297 0.018145517 0.020930233
20:        el_High.School 0.0137545420 0.022421510 0.015116279
21:   el_Specialist.Master 0.0101955625 0.005362287 0.003488372
22:                  tn34 0.0097462746 0.012136458 0.013953488
23:                  tn48 0.0062923955 0.009682384 0.010465116
24:     el_University.Deg. 0.0004826168 0.002249108 0.002325581
                Feature         Gain        Cover    Frequency
```

```r
# Graficar la importancia de las variables
xgb.plot.importance(importance_matrix = importancia)
```
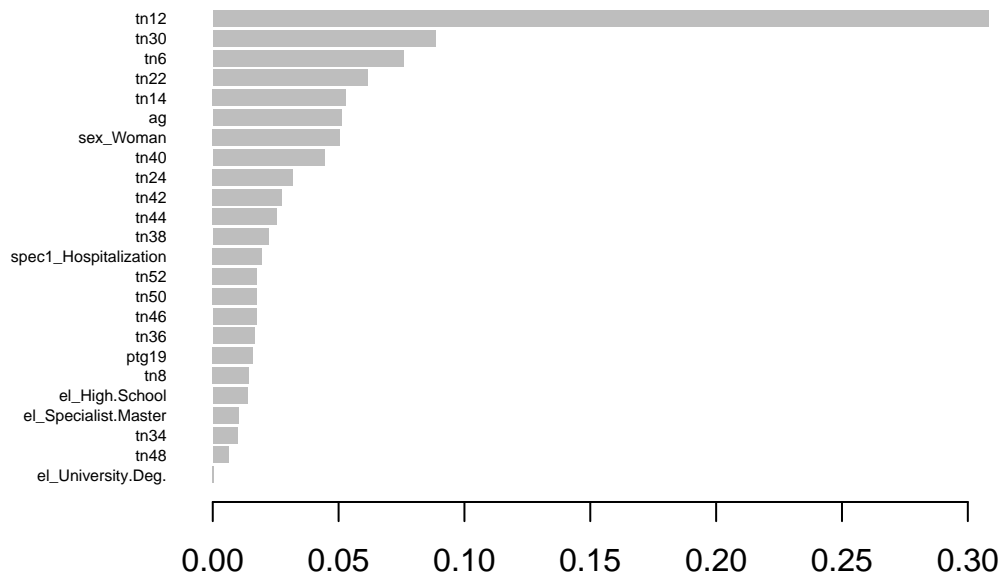

Para darle mayor explicabilidad al modelo vamos a calcular los valores SHAP de observaciones individuales

```r
# Definir la función de predicción personalizada para el modelo xgboost
pred_fun <- function(object, newdata) {
  predict(object, newdata = xgb.DMatrix(data = as.matrix(newdata)))
}


# Convertir test_data a data.frame y seleccionar una observación para explicar
test_df <- as.data.frame(as.matrix(test_data))
new_observation <- test_df[2, , drop = FALSE]  # Seleccionar la primera observación

# Calcular los valores SHAP para la observación seleccionada
set.seed(123)  # Para reproducibilidad
shap_values_fast <- fastshap::explain(
  object = modelo_xgb,            # El modelo de xgboost
  feature_names = colnames(train_data), # Los nombres de las variables
```
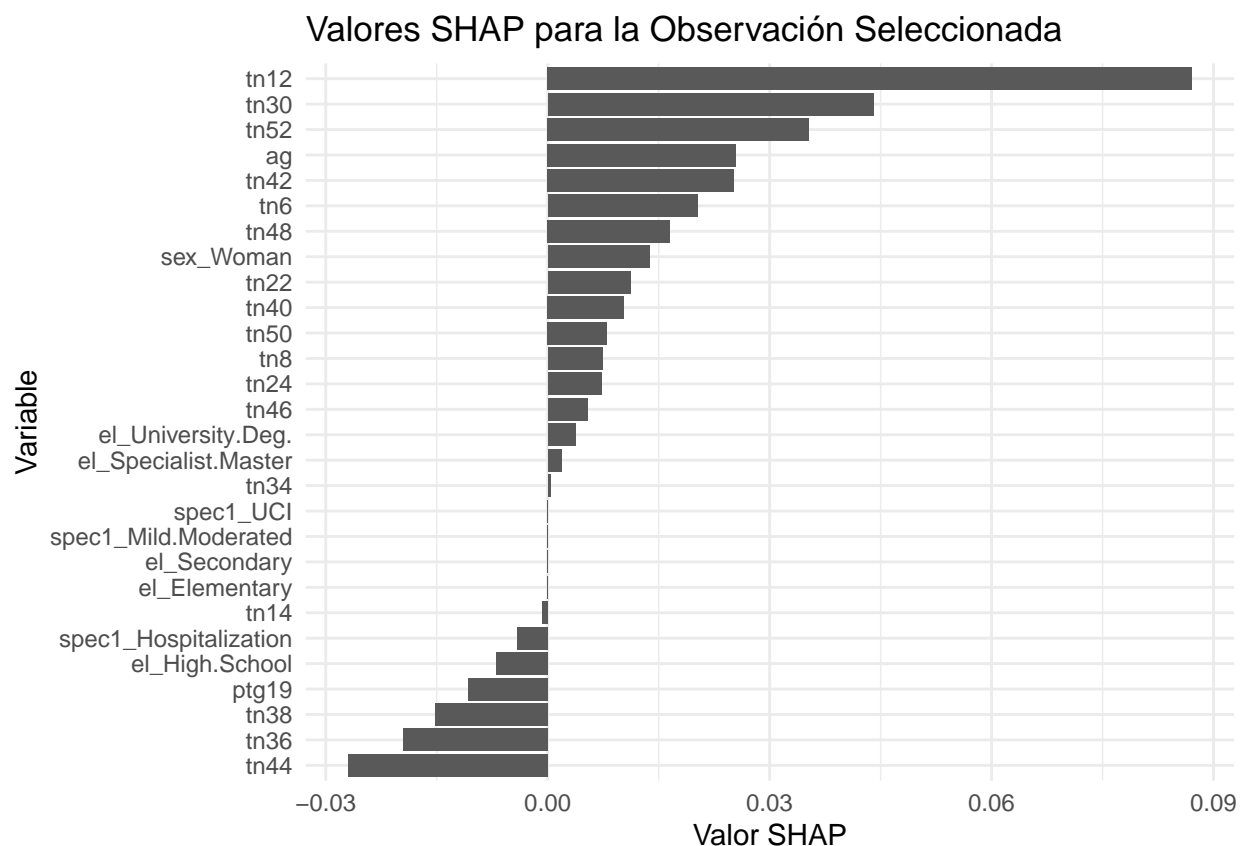
```r
  newdata = new_observation,        # La observación para la cual queremos valores SHAP
  pred_wrapper = pred_fun,          # La función de predicción personalizada
  X = train_data,                   # El conjunto de entrenamiento completo
  nsim = 100                        # Número de simulaciones
)

# Convertir los valores SHAP a un data.frame para graficar
shap_df <- data.frame(
  Variable = colnames(new_observation),
  SHAP = shap_values_fast[1, ]  # Solo la primera fila (si tienes más observaciones)
)

# Graficar los valores SHAP
ggplot(shap_df, aes(x = reorder(Variable, SHAP), y = SHAP)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Valores SHAP para la Observación Seleccionada",
       x = "Variable",
       y = "Valor SHAP") +
  theme_minimal()
```


Valores SHAP para la Observación Seleccionada

```r
########## VARIOS VALORES #############################

# Convertir test_data a data.frame y seleccionar múltiples observaciones para explicar (por ejemplo, la
test_df <- as.data.frame(as.matrix(test_data))
new_observations <- test_df[1:48, , drop = FALSE]  # Seleccionar las primeras 10 observaciones
```

```r
# Calcular los valores SHAP para las observaciones seleccionadas
set.seed(123)  # Para reproducibilidad
shap_values_fast_24 <- fastshap::explain(
  object = modelo_xgb,                # El modelo de xgboost
  feature_names = colnames(train_data), # Los nombres de las variables
  newdata = new_observations,         # Las observaciones seleccionadas
  pred_wrapper = pred_fun,            # La función de predicción personalizada
  X = train_data,                     # El conjunto de entrenamiento completo
  nsim = 100                          # Número de simulaciones
)

# Convertir shap_values_fast a data.frame y añadir un identificador de observación en el proceso de piv
shap_df_24 <- as.data.frame(shap_values_fast_24)
colnames(shap_df_24) <- colnames(train_data)  # Asegurarse de que los nombres de columnas sean los nomb
shap_df_24$Observation <- factor(1:48)  # Añadir la identificación de observación como factor

# Convertir a formato largo para visualización
shap_df_long <- shap_df_24 %>%
  pivot_longer(cols = -Observation, names_to = "Variable", values_to = "SHAP")

# Graficar los valores SHAP para múltiples observaciones
ggplot(shap_df_long, aes(x = reorder(Variable, SHAP), y = SHAP, fill = Observation)) +
  geom_bar(stat = "identity", position = "dodge") +
  coord_flip() +
  labs(title = "Comparación de Valores SHAP para Múltiples Observaciones",
       x = "Variable",
       y = "Valor SHAP") +
  theme_minimal() +
  theme(legend.position = "bottom")
```
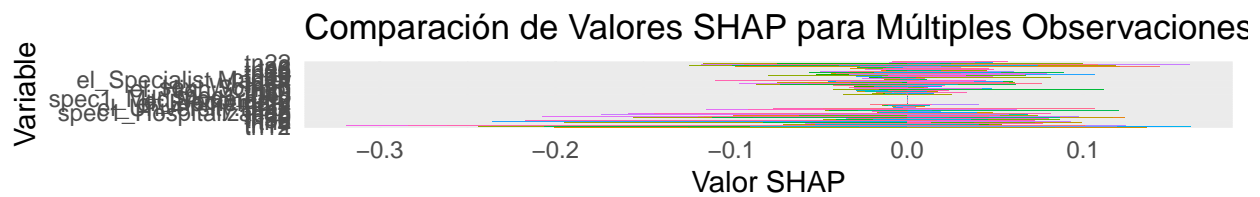
Comparación de Valores SHAP para Múltiples Observaciones


```r
############### COMPARAR TODOS LOS VALORES SHAP #############################

# Calcular la importancia promedio absoluta de cada variable
importance_summary <- shap_df_24 %>%
  dplyr::select(-Observation) %>%  # Excluir la columna de observación
  summarise_all(~ mean(abs(.))) %>%  # Calcular la media absoluta para cada variable
  pivot_longer(cols = everything(), names_to = "Variable", values_to = "Mean_Absolute_SHAP") %>%
  arrange(desc(Mean_Absolute_SHAP))  # Ordenar de mayor a menor importancia

# Graficar la importancia promedio de cada variable
ggplot(importance_summary, aes(x = reorder(Variable, Mean_Absolute_SHAP), y = Mean_Absolute_SHAP)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Importancia promedio de cada variable en las predicciones",
       x = "Variable",
       y = "Mean Absolute SHAP Value") +
  theme_minimal()
```

Importancia promedio de cada variable en las predicciones
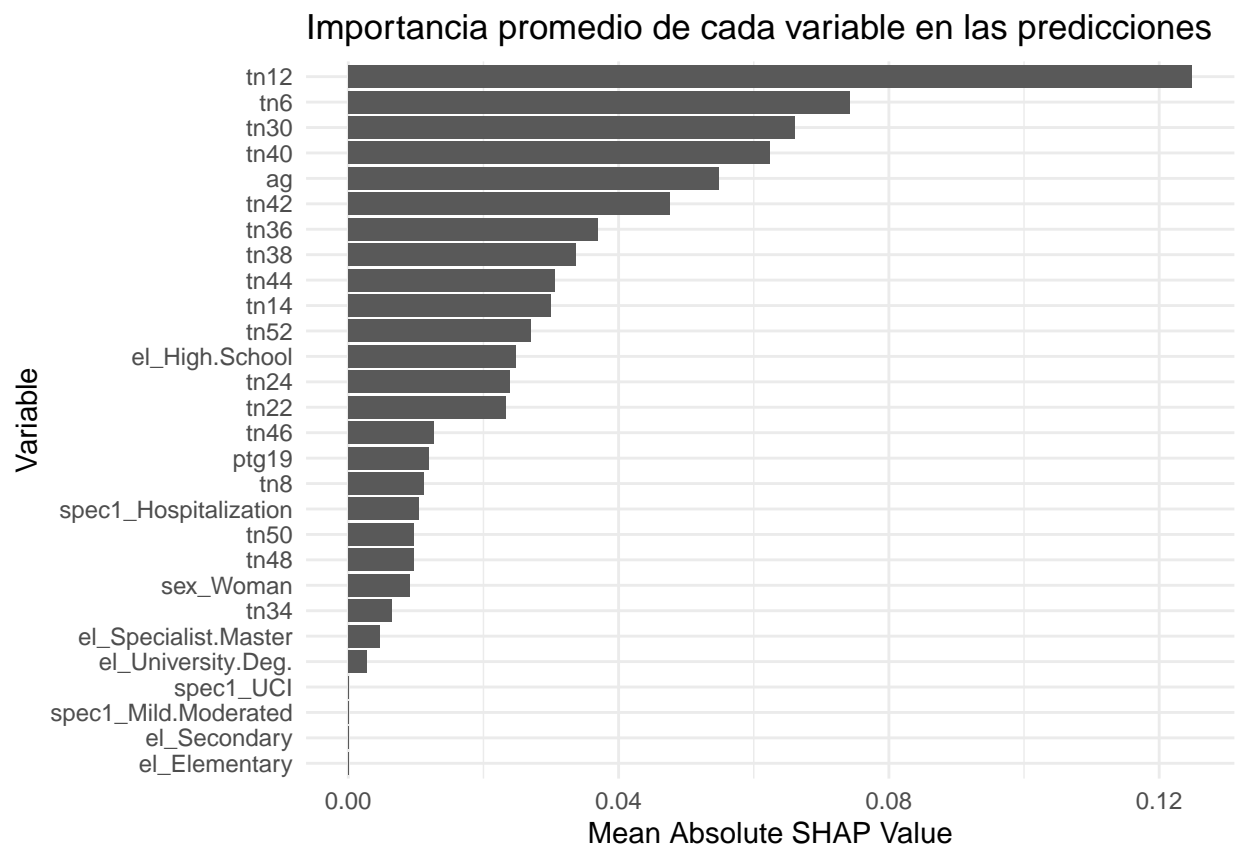
```
# Mostrar el data.frame con la importancia promedio de cada variable
print(importance_summary)
```

```
# A tibble: 28 x 2
   Variable Mean_Absolute_SHAP
   <chr>                 <dbl>
 1 tn12                  0.125
 2 tn6                   0.0742
 3 tn30                  0.0661
 4 tn40                  0.0623
 5 ag                    0.0548
 6 tn42                  0.0475
 7 tn36                  0.0369
 8 tn38                  0.0337
 9 tn44                  0.0306
10 tn14                  0.0299
# i 18 more rows
```