

```

1  ### SCRIPT 10 - R ###
2  # Analisis de datos especializado de los criterios de adjudicacion
3  #
4  #####
5
6  ### Lectura de Librerias -----
7
8  library(XML)
9  library(corrplot)
10 library(dplyr)
11 library(foreach)
12 library(stringr)
13 library(stringi)
14 library(doParallel)
15 library(DBI)
16 library(ggplot2)
17 library(dplyr)
18 library(openxlsx)
19 library(RODBC)
20 library(quanteda) #paquete recomendado en todos laos
21 library(quanteda.textmodels) #aux
22 library(quanteda.textstats) # aux
23 library(readtext) # sencilla manera de leer data de texto en R
24 library(spacyr) # NLP usando la libreria spaCy, incluyendo etiquetado part-of-speech,
entity recognition y dependency parsing.
25 library(devtools)
26 library(zoo)
27 library(DBI)
28 #setup parallel backend to use many processors
29 cores=detectCores()
30 cl <- makeCluster(cores[1]-1, type = "PSOCK", outfile = "log.txt") #not to overload
your computer #,outfile="log.txt"
31 registerDoParallel(cl)
32
33 f_ini <- Sys.time()
34
35
36 ## Directorio ----
37
38 setwd("C:/Users/guillermo.alonso/Desktop/Tesis Guillermo/2o articulo/AEIPRO 2024")
39
40
41 ### Lectura de datos ----
42
43 load("AwardingCriteria.RData")
44 AwardingCriteria<-Proy_BD
45 AwardingCriteria<-unique(AwardingCriteria) ## Eliminar filas repetidas (mismo
introducido varias veces en la BD)
46 rm(Proy_BD)
47
48 load("BaseDatosCompleta_R.RData")
49 BD_PLACSP<-unique(BD_PLACSP) ## Eliminar filas repetidas (mismo introducido varias
veces en la BD)
50 BD_PLACSP_civil<-unique(BD_PLACSP_civil) ## Eliminar filas repetidas (mismo
introducido varias veces en la BD)
51 BD_PLACSP_constru<-unique(BD_PLACSP_constru) ## Eliminar filas repetidas (mismo
introducido varias veces en la BD)
52 rm(BD_Agregadas)
53 rm(BD_Menores)
54 rm(BD_atom)
55 rm(BD_atom_civil)
56 rm(BD_atom_constru)
57 rm(BD_atom_IngeCivil)
58 rm(BD_PLACSP_IngeCivil)
59
60
61 ### Filtro de casos con datos erróneos y selección de casos ----
62
63 df <- AwardingCriteria
64
65 # Filtrar casos donde la suma de WeightNumeric es 100 o 1
66 df_filtered <- df %>%
67   group_by(entryID) %>%

```

```

68     filter(sum(WeightNumeric) %in% c(100, 1))
69
70 # Comprobar que para todos los entryID con el mismo valor, su ContractID también sea
71 # el mismo valor
72 df_filtered <- df_filtered %>%
73   group_by(entryID) %>%
74   filter(length(unique(ContractID)) == 1)
75
76 # Crear un nuevo data.frame llamado Awarding_100
77 Awarding_100 <- df_filtered
78
79 ### Junto el data-frame de precios con el data.frame de global -----
80
81 # Filtrar casos únicos
82 df_placsp_unicos <- distinct(BD_PLACSP)
83
84 # Quedarse con el caso más reciente por "Identificador"
85 df_placsp_filtrado <- df_placsp_unicos %>%
86   group_by(Identificador) %>%
87   filter(Fecha.actualizacion == max(Fecha.actualizacion))
88
89 # Cambiar los nombres de las variables en Awarding_100
90 Awarding_100 <- Awarding_100 %>%
91   rename(Identificador = entryID, Número.de.expediente = ContractID,
92     Descripcion.criterio = Description)
93
94 # Seleccionar casos comunes en "Identificador"
95 common_identifiers <- intersect(df_placsp_filtrado$Identificador, Awarding_100$
96   Identificador)
97
98 # Filtrar casos en df_placsp_filtrado que coincidan con "Identificador" en
99 # df_precio_percent
100 BD_placsp_criterios <- df_placsp_filtrado[df_placsp_filtrado$Identificador %in%
101   common_identifiers, ]
102
103 # Juntar las variables "WeightNumeric" y "Descripcion.criterio.precio"
104 # de df_precio_percent al data.frame resultante
105 BD_placsp_criterios <- merge(
106   BD_placsp_criterios,
107   Awarding_100[c("Identificador", "WeightNumeric", "Descripcion.criterio")],
108   by = "Identificador",
109   all.x = TRUE
110 )
111
112 # Seleccionar casos donde las variables mencionadas (Coste del proyecto) son mayores
113 # de 10000 o NA
114 BD_placsp_criterios <- subset(BD_placsp_criterios,
115   !is.na(`Presupuesto.base.sin.impuestos`) &
116   `Presupuesto.base.sin.impuestos` > 10000 &
117   !is.na(`Presupuesto.base.sin.impuestos.licitación/lote`) &
118   `Presupuesto.base.sin.impuestos.licitación/lote` >
119   10000 &
120   !is.na(
121     `Importe.adjudicación.sin.impuestos.licitación/lote`) &
122   `Importe.adjudicación.sin.impuestos.licitación/lote` >
123   10000
124 )
125
126 # Reorganizar las columnas según la especificación
127 BD_placsp_criterios <- BD_placsp_criterios[, c(
128   "Identificador",
129   "Número.de.expediente",
130   "WeightNumeric",
131   "Descripcion.criterio",
132   "Fecha.actualizacion",
133   "Objeto.del.Contrato",
134   names(BD_placsp_criterios)[!names(BD_placsp_criterios) %in% c("Identificador",
135     "Número.de.expediente", "WeightNumeric", "Descripcion.criterio",
136     "Fecha.actualizacion", "Objeto.del.Contrato")]
137 )]
```

```

127
128 ### Corrijo el campo del CPV
129 # Extraer el primer CPV de cada cadena
130 BD_placsp_criterios$CPV <- str_extract(BD_placsp_criterios$CPV, "\\b\\d{8}\\b")
131 # Truncar a las 2 primeras cifras
132 BD_placsp_criterios$CPV <- substr(BD_placsp_criterios$CPV, 1, 2)
133 # Convertir la variable "CPV" a tipo numérico
134 BD_placsp_criterios$CPV <- as.numeric(BD_placsp_criterios$CPV)
135
136 save(BD_placsp_criterios, file = "BD_placsp_criterios.RData")
137
138
139 # Filtro para seleccionar unicamente los casos de Construcción
140 BD_constru_criterios <- BD_placsp_criterios %>%
141   filter(CPV == 45)
142
143 # Filtramos casos en los que el peso del criterio es menor que 1.01 (errores o
144 valores no porcentuales) - Además quitamos cualquier proyecto que tuviese uno de esos
145 BD_constru_criterios <- BD_constru_criterios %>%
146   filter(WeightNumeric >= 1.01) %>%
147   anti_join(BD_constru_criterios %>%
148     filter(WeightNumeric < 1.01),
149     by = "Identificador")
150
151 # Separamos los lotes por identificadores diferentes, añadiendo -X, donde X es el
152 número de lote
153 BD_constru_criterios <- BD_constru_criterios %>%
154   mutate(
155     Identificador = ifelse(Lote == "Sin lotes", Identificador, paste0(Identificador,
156       "-", Lote))
157   )
158 BD_constru_criterios <- unique(BD_constru_criterios)
159
160 # Si los valores de Identificador WeightNumeric y Descripcion.criterio coinciden
161 correspondientemente con más de un caso,
162 # me quedo solo con el primero de los casos que aparece
163 BD_constru_criterios <- BD_constru_criterios %>%
164   distinct(Identificador, WeightNumeric, Descripcion.criterio, .keep_all = TRUE)
165
166 save(BD_constru_criterios, file = "BD_constru_criterios.RData")
167
168 ### Seleccion criterios económicos -----
169
170 df_economicos <- BD_constru_criterios[c("Identificador"
171   , "WeightNumeric"
172   , "Descripcion.criterio"
173   , "Presupuesto.base.sin.impuestos.licitación/lote"
174   ,
175   "Importe.adjudicación.sin.impuestos.licitación/lo
176   te"
177   , "CPV"
178   , "Fecha.actualizacion"
179   ,
180   "Número.de.ofertas.recibidas.por.licitación/lote"
181   , "Tipo.de.contrato" # Demasiados de tipo Obra que
182   no da pie a correlacion
183   , "Tipo.de.Administración"
184   , "Código.Postal"
185   , "Tipo.de.procedimiento"
186   , "Tramitación"
187   , "Objeto.licitación/lote"
188   , "Adjudicatario.licitación/lote"
189   , "Órgano.de.Contratación"
190   , "Número.de.expediente"
191 ])
192 df_economicos <- unique(df_economicos)
193
194 # Seleccionar casos donde en el campo Descripcion.criterio aparecen las palabras clave
195 # str_detect se utiliza para verificar si las palabras clave (en minúsculas y sin
196 tilde)
197 # tolower se usa para asegurarse de que la comparación sea insensible a mayúsculas y
198 minúsculas

```

```

190 df_economicos <- df_economicos %>%
191   filter(
192     str_detect(tolower(str_trim(Descripcion.criterio)),
193       "precio|economico|economica|coste|económico|económica") &
194     !str_detect(tolower(str_trim(Descripcion.criterio)), "técnica")
195   )
196 # Suma de valores de precio en casos con mas de un criterio de precio
197 df_economicos <- df_economicos %>%
198   group_by(Identificador) %>%
199   summarise(
200     WeightNumeric = sum(WeightNumeric),
201     Descripcion.criterio = "Oferta economica",
202     across(everything(), ~ first(.), .names = "{col}"),
203     .groups = 'drop' # Esto evita el warning en versiones más recientes de dplyr
204   )
205
206 save(df_economicos, file = "BD_constru_precios.RData")
207
208 ### Seleccion de resto de criterios objetivos ----
209 df_objetivos<-BD_constru_criterios[c("Identificador"
210                                     , "WeightNumeric"
211                                     , "Descripcion.criterio"
212 )]
213
214 # Seleccionar casos donde en el campo Descripcion.criterio aparecen las palabras clave
215 # str_detect se utiliza para verificar si las palabras clave (en minúsculas y sin
216 # tilde)
217 # tolower se usa para asegurarse de que la comparación sea insensible a mayúsculas y
218 # minúsculas
219 df_objetivos <- df_objetivos %>%
220   filter(
221     str_detect(tolower(str_trim(Descripcion.criterio)),
222       "garantía|garantia|plazo|objetivo|fórmula|formula") &
223     !str_detect(tolower(str_trim(Descripcion.criterio)),
224       "técnica|técnico|tecnica|tecnico|cualitativo|juicio|precio|economico|economica|c
225       oste|económico|económica")
226   )
227
228 # Filtrar los casos en los que "Descripcion.criterio" contiene ambas palabras "no" y
229 # "fórmula" juntas
230 df_objetivos <- df_objetivos %>%
231   filter(
232     !str_detect(tolower(Descripcion.criterio),
233       "\\bno\\b.*\\bfórmula\\b|\\bfórmula\\b.*\\bno\\b")
234   )
235 df_objetivos <- df_objetivos %>%
236   filter(
237     !str_detect(tolower(Descripcion.criterio),
238       "\\bno\\b.*\\bfórmulas\\b|\\bfórmulas\\b.*\\bno\\b")
239   )
240 df_objetivos <- df_objetivos %>%
241   filter(
242     !str_detect(tolower(Descripcion.criterio),
243       "\\bno\\b.*\\bformula\\b|\\bformula\\b.*\\bno\\b")
244   )
245 df_objetivos <- df_objetivos %>%
246   filter(
247     !str_detect(tolower(Descripcion.criterio),
248       "\\bno\\b.*\\bformulas\\b|\\bformulas\\b.*\\bno\\b")
249   )
250
251 df_objetivos<-unique(df_objetivos)
252
253 df_resto_objetivos <- df_objetivos %>%
254   group_by(Identificador) %>%
255   summarise(
256     WeightNumeric = sum(WeightNumeric),
257     Descripcion.criterio = "Resto criterios objetivos",
258     .groups = 'drop' # Esto evita el warning en versiones más recientes de dplyr
259   )
260
261 save(df_objetivos,df_resto_objetivos, file="df_resto_objetivos.RData" )

```



```

313         ,"Tipo.de.contrato" #Demasiados de tipo Obra que no
314         da pie a correlacion
315         ,"Tipo.de.Administración"
316         ,"Tipo.de.procedimiento"
317     )]
318
319
320 #Cambio la fecha por un valor numérico del año en el que se hizo
321
322 data_corrplot$Fecha.actualizacion<-as.Date(data_corrplot$Fecha.actualizacion,
323 "%d/%m/%Y")
324 data_corrplot$Fecha.actualizacion<-format(data_corrplot$Fecha.actualizacion,"%Y")
325 data_corrplot$Fecha.actualizacion<-as.numeric(data_corrplot$Fecha.actualizacion)
326
327 ## Cambio las tipo factor a tipo numeric:
328
329 data_corrplot <- data_corrplot %>% ## Demasiados de tipo Obras que dan pie a poca
330 correlacion
331 mutate(
332     Tipo.de.contrato = case_when(
333         Tipo.de.contrato == "Obras" ~ 1L,
334         Tipo.de.contrato == "Servicios" ~ 2L,
335         Tipo.de.contrato == "Suministros" ~ 3L,
336         Tipo.de.contrato == "Gestión de Servicios Públicos" ~ 4L,
337         Tipo.de.contrato == "Privado" ~ 5L,
338         TRUE ~ NA_integer_
339     )
340 )
341 data_corrplot <- data_corrplot %>%
342 mutate(
343     Tramitación = case_when(
344         Tramitación == "Ordinaria" ~ 1L,
345         Tramitación == "Urgente" ~ 2L,
346         Tramitación == "Emergencia" ~ 3L,
347         TRUE ~ NA_integer_
348     )
349 )
350 data_corrplot <- data_corrplot %>%
351 mutate(
352     Tipo.de.Administración = case_when(
353         Tipo.de.Administración == "Administración General del Estado" ~ 1L,
354         Tipo.de.Administración == "Comunidad Autónoma" ~ 2L,
355         Tipo.de.Administración == "Administración Local" ~ 3L,
356         Tipo.de.Administración == "Entidad de Derecho Público" ~ 4L,
357         Tipo.de.Administración == "Otras Entidades del Sector Público" ~ 5L,
358         TRUE ~ NA_integer_
359     )
360 )
361 data_corrplot <- data_corrplot %>%
362 mutate(
363     Tipo.de.procedimiento = case_when(
364         Tipo.de.procedimiento == "Abierto" ~ 1L,
365         Tipo.de.procedimiento == "Abierto simplificado" ~ 2L,
366         Tipo.de.procedimiento == "Derivado de acuerdo marco" ~ 3L,
367         Tipo.de.procedimiento == "Negociado con publicidad" ~ 4L,
368         Tipo.de.procedimiento == "Normas internas" ~ 5L,
369         Tipo.de.procedimiento == "Negociado sin publicidad" ~ 6L,
370         Tipo.de.procedimiento == "Restringido" ~ 7L,
371         TRUE ~ NA_integer_
372     )
373 )
374
375 ### Calculo y representacion del mapa de calor de correlaciones -----
376 matriz_correlacion <- cor(data_corrplot, method = "spearman", use =
377 "pairwise.complete.obs")
378
379 # Configurar el dispositivo PNG con alta calidad
380 png("corrplot.png", width = 5000, height = 3500, units = "px", res = 400)
381
382 corrplot(matriz_correlacion, method = "color", type = "full", order = "hclust", tl.cex
383 = 0.65, tl.col = "black", addCoef.col = "black", cl.cex = 0.65, number.cex = 0.6)
384
385 # Finalizar la salida del PNG

```

```

381 dev.off()
382
383
384 ### Estadística descriptiva de la base de datos -----
385
386 # Selección de las variables cuantitativas continuas
387 variables_quan <- c("Presupuesto.base.sin.impuestos.licitación/lote",
388                   "Importe.adjudicación.sin.impuestos.licitación/lote",
389                   "Número.de.ofertas.recibidas.por.licitación/lote"
390                   , "Porcentaje_economico"
391                   , "Porcentaje_resto_objetivos"
392                   , "Porcentaje_juicio_valor"
393 )
394
395 names_quan <- c("Presupuesto.base.sin.impuestos.licitación/lote",
396                "Importe adjudicación sin impuestos licitación/lote",
397                "Número de ofertas recibidas por licitación/lote"
398                , "Porcentaje_economico"
399                , "Porcentaje_resto_objetivos"
400                , "Porcentaje_juicio_valor"
401 )
402
403 names(variables_quan) <- names_quan
404
405
406 # Cálculo de las estadísticas descriptivas
407 stats <- apply(data_corrplot[, variables_quan], 2, function(x) {
408   n_cases <- sum(!is.na(x))
409   prop_na <- sum(is.na(x)) / length(x)
410   range_iqr <- IQR(x, na.rm = TRUE)
411   c(
412     Min = min(x, na.rm = TRUE),
413     Median = median(x, na.rm = TRUE),
414     Max = max(x, na.rm = TRUE),
415     Mean = mean(x, na.rm = TRUE),
416     SD = sd(x, na.rm = TRUE),
417     Variance = var(x, na.rm = TRUE),
418     N_Cases = n_cases,
419     Prop_NA = prop_na,
420     Range_IQR = range_iqr
421   )
422 })
423
424 # Creación de la tabla de estadísticas descriptivas
425 tabla_quan <- as.data.frame(stats)
426 tabla_quan <- t(tabla_quan)
427 write.table(tabla_quan, file="tabla_quan.csv", sep=";", dec=".", na="NA", row.names = T,
428            col.names = T)
429
430 # Selección de las variables cualitativas
431 data_cual <- BD_criterios[c("Fecha.actualizacion"
432                            , "Tipo.de.contrato" # Demasiados de tipo Obra que no da
433                            , "Tipo.de.Administración"
434                            , "Tipo.de.procedimiento"
435                            , "Tramitación"
436 )]
437
438 # Cambio la fecha por un valor numérico del año en el que se hizo
439 data_cual$Fecha.actualizacion <- as.Date(data_cual$Fecha.actualizacion, "%d/%m/%Y")
440 data_cual$Fecha.actualizacion <- format(data_cual$Fecha.actualizacion, "%Y")
441 data_cual$Fecha.actualizacion <- as.numeric(data_cual$Fecha.actualizacion)
442
443 # Traducir los elementos factor de la variable 'Tipo.de.contrato'
444 data_cual$Tipo.de.contrato <- factor(data_cual$Tipo.de.contrato, levels = c("Obras",
445 "Servicios", "Suministros", "Gestión de Servicios Públicos", "Privado"))
446
447 # Traducir los elementos factor de la variable 'Tramitación'
448 data_cual$Tramitación <- factor(data_cual$Tramitación, levels = c("Ordinaria",
449 "Urgente", "Emergencia"))
450
451 # Traducir los elementos factor de la variable 'Tipo.de.Administración'

```

```

450 data_cual$Tipo.de.Administración <- factor(data_cual$Tipo.de.Administración, levels =
c("Administración General del Estado", "Comunidad Autónoma", "Administración Local",
"Entidad de Derecho Público", "Otras Entidades del Sector Público"))
451
452 # Traducir los elementos factor de la variable 'Tipo.de.procedimiento'
453 data_cual$Tipo.de.procedimiento <- factor(data_cual$Tipo.de.procedimiento, levels = c(
"Abierto", "Abierto simplificado", "Derivado de acuerdo marco", "Negociado con
publicidad", "Normas internas", "Negociado sin publicidad", "Restringido"))
454
455 # Traducir los elementos factor de la variable 'Fecha.actualizacion'
456 data_cual$Fecha.actualizacion <- as.factor(data_cual$Fecha.actualizacion)
457
458
459 variables_ingles<-c("Fecha actualizacion"
460                    , "Tipo de contrato" #Demasiados de tipo Obra que no da pie a
correlacion
461                    , "Tipo de Administración"
462                    , "Tipo de procedimiento"
463                    , "Tramitación"
464                    )
465 colnames(data_cual)<-variables_ingles
466
467
468 # Cálculo de las estadísticas descriptivas
469
470
471 stats2 <- lapply(data_cual[, variables_ingles], function(x) {
472   n_cases <- sum(!is.na(x))
473   prop_na <- sum(is.na(x)) / length(x)
474   table_data <- table(x, useNA = "ifany")
475   prop_data <- prop.table(table_data)
476   data.frame(Frequency = table_data, Proportion = prop_data, N_Cases = n_cases, N_As =
sum(is.na(x)))
477 })
478
479 # Combinar las estadísticas en una tabla final
480 tabla_cual <- do.call(rbind, stats2)
481 write.table(tabla_cual, file="tabla_cual.csv", sep=";", dec=".", na="NA", row.names = T,
col.names=T)
482
483
484 ### Guardar en csv la base de datos final ----
485 BD_criterios_final<-BD_criterios
486 names(BD_criterios_final) <- iconv(names(BD_criterios_final), "UTF-8",
"ASCII//TRANSLIT") #quito tildes nombres
487
488 BD_criterios_final <- data.frame(lapply(BD_criterios_final, function(x) { #quito
tildes de variables de tipo texto
489   if(is.factor(x) | is.character(x)) {
490     x <- iconv(x, "UTF-8", "ASCII//TRANSLIT") # Eliminar tildes
491     x <- gsub(";", "", x) # Eliminar caracteres ";"
492     return(x)
493   } else {
494     return(x)
495   }
496 }))
497 write.table(BD_criterios_final, file="BD_criterios_final.csv", sep=";", dec=".", na="NA",
row.names = F, col.names=T)
498
499 ### Analisis de casos de mas de 5M € -----
500
501 BD_criterios_5M <- filter(BD_criterios_final,
Presupuesto.base.sin.impuestos.licitacion.lote > 5000000)
502 write.table(BD_criterios_5M, file="BD_criterios_5M.csv", sep=";", dec=".", na="NA",
row.names = F, col.names=T)
503
504
505
506
507

```