

```

1  ##### SCRIPT 6 - R #####
2  # Tratamiento y limpieza de datos inicial de los criterios de adjudicacion
3  #
4  ##########
5  # Cargar paquetes -----
6  library(ggplot2)
7  library(dplyr)
8  library(openxlsx)
9  library(RODBC)
10 library(quanteda)
11 library(quanteda.textmodels)
12 library(quanteda.textstats)
13 library(readtext)
14 library(spacyr)
15 library(devtools)
16 library(zoo)
17 library(tidyr)
18
19 # Establecer ruta de trabajo -----
20 setwd("C:/Users/guillermo.alonso/Desktop/Tesis Guillermo/_BD actualizada/Datos atom/PLACSP")
21
22 # Cargar archivos AwardingCriteria por año -----
23 años <- 2012:2025
24 lista_df <- list()
25
26 for (año in años) {
27   archivo <- paste0("4.33_AwardingCriteria_", año, ".RData")
28   load(archivo)
29   lista_df[[as.character(año)]] <- Proy
30 }
31
32 # Unir todos los años en un único data frame -----
33 awarding_criteria <- bind_rows(lista_df)
34 awarding_criteria <- unique(awarding_criteria) # Eliminar filas duplicadas
35 rm(Proy)
36
37 # Filtrado de datos -----
38 # Filtrar casos donde la suma de WeightNumeric es 100 o 1
39 # Filtramos casos en los que el peso del criterio es menor que 1.01 (errores o
valores no porcentuales)
40 # Ademas quitamos cualquier proyecto que tuviese uno de esos
41 df <- awarding_criteria
42
43 df_filtered <- df |>
44   group_by(ContractID) |>
45   filter(sum(WeightNumeric) %in% c(100, 1)) |>
46   filter(WeightNumeric >= 1.01) |>
47   anti_join(
48     df |>
49       filter(WeightNumeric < 1.01),
50     by = "ContractID"
51   )
52
53 # Guardar resultado final
54 awarding_100 <- df_filtered
55
56
57 # Seleccionar CRITERIO PRECIO donde en el campo Description aparecen las palabras
clave de precio, y quitar tecnica -----
58 # str_detect se utiliza para verificar si las palabras clave (en minúsculas y sin
tilde)
59 # tolower se usa para asegurarse de que la comparación sea insensible a mayúsculas y
minúsculas
60
61 # Filtrar criterios relacionados con el precio
62 awarding_precio <- awarding_100 |>
63   filter(
64     str_detect(
65       tolower>Description),
66       "\\b(precio|econ[ó]mico|econ[ó]mica|coste|costes|gasto|gastos)\\b"
67     ) &
68     !str_detect(

```

```

69     tolower(str_trim>Description)),
70     "t[eé]cnica"
71   )
72 )
73
74 # Agrupar y sumar pesos por criterio de precio
75 precio_percent <- awarding_precio |>
76   group_by(entryID, ContractID) |>
77   summarise(
78     WeightNumeric = sum(WeightNumeric),
79     Description = "Oferta económica",
80     .groups = "drop"
81   ) |>
82   filter(WeightNumeric != 0)
83
84
85
86 # Seleccionar el RESTO CRITERIOS OBJETIVOS donde en el campo Description aparecen las
87 # palabras clave -----
88 # str_detect se utiliza para verificar si las palabras clave (en minúsculas y sin
89 # tilde)
90 # tolower se usa para asegurarse de que la comparación sea insensible a mayúsculas y
91 # minúsculas
92 # Filtrar criterios objetivos distintos de técnica y precio
93 resto_objetivos <- awarding_100 |>
94   filter(
95     str_detect(
96       tolower(str_trim>Description)),
97       "\b(garant[í]a|plazo|objetivo|f[ó]rmula)\b"
98     ) &
99     !str_detect(
100       tolower(str_trim>Description),
101         "\b(t[eé]cnica|t[eé]cnico|cualitativo|juicio|precio|econ[oó]mico|econ[oó]mica
102         |coste)\b"
103     )
104   )
105
106 # Eliminar los casos que contienen expresiones como "no fórmula
107 resto_objetivos <- resto_objetivos |>
108   filter(
109     !str_detect(
110       tolower>Description),
111       "\bno\b.*\bf[ó]rmul(a|as)\b|\bf[ó]rmul(a|as)\b.*\bno\b"
112     )
113
114 # Eliminar duplicados
115 resto_objetivos <- distinct(resto_objetivos)
116
117 # Agrupar y resumir pesos por identificador
118 resto_objetivos_percent <- resto_objetivos |>
119   group_by(ContractID) |>
120   reframe(
121     entryID = entryID,
122     WeightNumeric = sum(WeightNumeric, na.rm=TRUE),
123     Description = "Resto criterios objetivos"
124   )
125 resto_objetivos_percent<-unique(resto_objetivos_percent) #eliminar duplicados
126
127
128 #### Juntamos en data frame final -----
129 # Unimos ambos data frames
130 awarding_criteria_final <- bind_rows(
131   precio_percent,
132   resto_objetivos_percent
133 )
134
135 # Ordenamos por ContractID
136 awarding_criteria_final <- awarding_criteria_final |>
137   arrange(ContractID)
138
139 #### Agrupamos los criterios por casos concretos, por parejas ContractID-entryID

```

```

137
138 # Paso 1: identificar duplicados por grupo
139 casos_mal <- awarding_criteria_final |>
140   group_by(entryID, ContractID, Description) |>
141   filter(n() > 1) |>
142   filter(WeightNumeric < max(WeightNumeric, na.rm = TRUE)) |>
143   ungroup()
144
145 # Paso 2: quedarnos con el mayor valor por grupo
146 awarding_criteria_limpio <- awarding_criteria_final |>
147   group_by(entryID, ContractID, Description) |>
148   slice_max(WeightNumeric, n = 1, with_ties = FALSE) |>
149   ungroup()
150
151 # Paso 3: pivotear a formato ancho
152 awarding_criteria_final_wide <- awarding_criteria_limpio |>
153   pivot_wider(
154     names_from = Description,
155     values_from = WeightNumeric,
156     values_fill = 0,
157     names_prefix = "%_"
158   ) |>
159   rename(
160     `%_precio` = `%_Oferta economica`,
161     `%_resto_objetivos` = `%_Resto criterios objetivos`
162   )
163
164 ### Limpiamos casos y calculamos %_juicios_valor como la diferencia -----
165
166 # Paso 1: calcular la suma y marcar casos inválidos
167 awarding_criteria_final_wide <- awarding_criteria_final_wide |>
168   mutate(suma_pesos = `%_precio` + `%_resto_objetivos`)
169
170 # Paso 2: extraer los casos incorrectos
171 casos_mal <- awarding_criteria_final_wide |>
172   filter(suma_pesos == 0 | suma_pesos < 1 | suma_pesos > 100)
173
174 # Paso 3: eliminar esos casos del data frame principal
175 awarding_criteria_final_wide <- awarding_criteria_final_wide |>
176   filter(!(suma_pesos == 0 | suma_pesos < 1 | suma_pesos > 100))
177
178 # Paso 4 (opcional): quitar la columna auxiliar si ya no se necesita
179 awarding_criteria_final_wide <- awarding_criteria_final_wide |>
180   select(-suma_pesos)
181 # Paso 5: Saco el valor de los criterios de juicio de valor por la resta de las otras
182 awarding_criteria_final_wide <- awarding_criteria_final_wide %>%
183   mutate(
184     `%_juicios_valor` = 100 - (`%_precio` + `%_resto_objetivos`)
185   )
186
187 ## Guardamos resultados -----
188 DF_Awarding_Final<-awarding_criteria_final_wide
189
190 # Guardamos el dataframe como archivo .RData
191 save(DF_Awarding_Final, file = "DF_Awarding_Final.RData")
192
193 # Exportamos como .csv con separador ";"
194 write.csv2(DF_Awarding_Final, file = "DF_Awarding_Final.csv", sep = ";", row.names =
195 FALSE)

```