

```

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 %>%
  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$Identificador, Awarding_100$Identificador)
96
97 # Filtrar casos en df_placsp_filtrado que coincidan con "Identificador" en
98 # df_precio_percent
99 BD_placsp_criterios <- df_placsp_filtrado[df_placsp_filtrado$Identificador %in% common_identifiers, ]
100
101 # Juntar las variables "WeightNumber" y "Descripcion.criterio.precio"
102 # de df_precio_percent al data.frame resultante
103 BD_placsp_criterios <- merge(
104   BD_placsp_criterios,
105   Awarding_100[c("Identificador", "WeightNumeric", "Descripcion.criterio")],
106   by = "Identificador",
107   all.x = TRUE
108 )
109
110 # Seleccionar casos donde las variables mencionadas (Coste del proyecto) son mayores
111 # de 10000 o NA
112 BD_placsp_criterios <- subset(BD_placsp_criterios,
113   !is.na(`Presupuesto.base.sin.impuestos`) &
114     `Presupuesto.base.sin.impuestos` > 10000 &
115       !is.na(`Presupuesto.base.sin.impuestos.licitación/lote`) &
116         `Presupuesto.base.sin.impuestos.licitación/lote` > 10000 &
117           !is.na(
118             `Importe.adjudicación.sin.impuestos.licitación/lote`) &
119               `Importe.adjudicación.sin.impuestos.licitación/lote` > 10000
120
121
122 # Reorganizar las columnas según la especificación
123 BD_placsp_criterios <- BD_placsp_criterios[, c(
124   "Identificador",
125   "Número.de.expediente",
126   "WeightNumeric",
127   "Descripcion.criterio",
128   "Fecha.actualizacion",
129   "Objeto.del.Contrato",
130   names(BD_placsp_criterios)[!names(BD_placsp_criterios) %in% c("Identificador",
131     "Número.de.expediente", "WeightNumeric", "Descripcion.criterio",
132     "Fecha.actualizacion", "Objeto.del.Contrato")]]
133
134
135

```

```

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 únicamente 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) - Ademas 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 # numero de lote
153
154 BD_constru_criterios <- BD_constru_criterios %>%
155   mutate(
156     Identificador = ifelse(Lote == "Sin lotes", Identificador, paste0(Identificador,
157       "-", Lote)))
158
159 # Si los valores de Identificador WeightNumeric y Descripcion.criterio coinciden
160 # correspondientemente con más de un caso,
161 # me quedo solo con el primero de los casos que aparece
162 BD_constru_criterios <- BD_constru_criterios %>%
163   distinct(Identificador, WeightNumeric, Descripcion.criterio, .keep_all = TRUE)
164
165 save(BD_constru_criterios,file = "BD_constru_criterios.RData")
166
167 ##### Selección criterios económicos -----
168
169 df_economicos<-BD_constru_criterios[c("Identificador"
170                                         , "WeightNumeric"
171                                         , "Descripcion.criterio"
172                                         , "Presupuesto.base.sin.impuestos.licitación/lote"
173                                         ,
174                                         "Importe.adjudicación.sin.impuestos.licitación/lo
175                                         te"
176                                         , "CPV"
177                                         , "Fecha.actualización"
178                                         ,
179                                         "Número.de.ofertas.recibidas.por.licitación/lote"
180                                         , "Tipo.de.contrato" #Demasiados de tipo Obra que
181                                         #no da pie a correlacion
182                                         , "Tipo.de.Administración"
183                                         , "Código.Postal"
184                                         , "Tipo.de.procedimiento"
185                                         , "Tramitación"
186                                         , "Objeto.licitación/lote"
187                                         , "Adjudicatario.licitación/lote"
188                                         , "Órgano.de.Contratación"
189                                         , "Número.de.expediente"
190 )
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
197 # Suma de valores de precio en casos con mas de un criterio de precio
198 df_economicos <- df_economicos %>%
199   group_by(Identificador) %>%
200   summarise(
201     WeightNumeric = sum(WeightNumeric),
202     Descripcion.criterio = "Oferta economica",
203     across(everything(), ~ first(.), .names = "{col}"),
204     .groups = 'drop' # Esto evita el warning en versiones más recientes de dplyr
205   )
206
207 save(df_economicos, file = "BD_constru_precios.RData")
208
209 ### Seleccion de resto de criterios objetivos ----
210 df_objetivos<-BD_constru_criterios[c("Identificador"
211                                         , "WeightNumeric"
212                                         , "Descripcion.criterio"
213                                         )]
214
215 # Seleccionar casos donde en el campo Descripcion.criterio aparecen las palabras clave
216 # str_detect se utiliza para verificar si las palabras clave (en minúsculas y sin
217 # tilde)
218 # tolower se usa para asegurarse de que la comparación sea insensible a mayúsculas y
219 # minúsculas
220 df_objetivos <- df_objetivos %>%
221   filter(
222     str_detect(tolower(str_trim(Descripcion.criterio)),
223     "garantía|garantia|plazo|objetivo|fórmula|formula") &
224     !str_detect(tolower(str_trim(Descripcion.criterio)),
225     "técnica|técnico|técnica|técnico|cualitativo|juicio|precio|económico|económica|c
226     oste|económico|económica")
227   )
228
229 # Filtrar los casos en los que "Descripcion.criterio" contiene ambas palabras "no" y
230 # "fórmula" juntas
231 df_objetivos <- df_objetivos %>%
232   filter(
233     !str_detect(tolower(Descripcion.criterio),
234     "\\\bno\\\b.*\\\bfórmula\\\b|\\\bfórmula\\\b.*\\\bno\\\b")
235   )
236 df_objetivos <- df_objetivos %>%
237   filter(
238     !str_detect(tolower(Descripcion.criterio),
239     "\\\bno\\\b.*\\\bfórmulas\\\b|\\\bfórmulas\\\b.*\\\bno\\\b")
240   )
241 df_objetivos <- df_objetivos %>%
242   filter(
243     !str_detect(tolower(Descripcion.criterio),
244     "\\\bno\\\b.*\\\bformula\\\b|\\\bformula\\\b.*\\\bno\\\b")
245   )
246 df_objetivos <- df_objetivos %>%
247   filter(
248     !str_detect(tolower(Descripcion.criterio),
249     "\\\bno\\\b.*\\\bformulas\\\b|\\\bformulas\\\b.*\\\bno\\\b")
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" )

```

```

252
253 #### Agrupación de dataframes de criterios (añadimos de juicios de valor) ----
254 BD_criterios<-BD_constru_criterios[c("Identificador"
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269 )]
270
271 BD_criterios<-unique(BD_criterios) ##Estan duplicados por varios criterios
272
273 # Seleccionar casos de BD_criterios donde "Identificador" coincide con alguno en
274 df_economicos
275 BD_criterios<-BD_criterios %>%
276   semi_join(df_economicos, by = "Identificador")
277
278 BD_criterios<-unique(BD_criterios) ## Por si acaso, pero no deberia hacer nada
279
280 # Fusionar BD_criterios con df_economicos y renombrar columnas
281 BD_criterios <- merge(BD_criterios, df_economicos[, c("Identificador", "WeightNumeric")]
282   ), by = "Identificador", all.x = TRUE)
283 BD_criterios <- BD_criterios %>% rename(Porcentaje_economico = WeightNumeric)
284
285 # Fusionar BD_criterios con df_resto_objetivos y renombrar columnas
286 BD_criterios <- merge(BD_criterios, df_resto_objetivos[, c("Identificador",
287   "WeightNumeric")], by = "Identificador", all.x = TRUE)
288 BD_criterios <- BD_criterios %>% rename(Porcentaje_resto_objetivos = WeightNumeric)
289
290 # Reemplazar NA por 0 en las nuevas variables
291 BD_criterios$Porcentaje_resto_objetivos <- coalesce(BD_criterios$Porcentaje_resto_objetivos, 0)
292
293 ## Filtramos para quedarnos con los casos en los que la suma de los criterios
294 ## objetivos sea menor que 100
295 BD_criterios <- BD_criterios %>%
296   filter(Porcentaje_economico + Porcentaje_resto_objetivos <= 100)
297
298 # Saco el valor de los criterios de juicio de valor por la resta de las otras
299 BD_criterios <- BD_criterios %>%
300   mutate(
301     Porcentaje_juicio_valor = 100 - (Porcentaje_economico + Porcentaje_resto_objetivos)
302   )
303
304 save(BD_criterios, file = "BD_criterios_final.RData")
305
306 #### Análisis estadístico del criterio de oferta economica ----
307
308 #### Selección de las variables para representar en corrplot
309
310 data_corrplot<-BD_criterios[c("Presupuesto.base.sin.impuestos.licitación/lote"
311
312   , "Importe.adjudicación.sin.impuestos.licitación/lote"
313   , "Fecha.actualizacion"
314   , "Número.de.ofertas.recibidas.por.licitación/lote"
315   , "Porcentaje_economico"
316   , "Porcentaje_resto_objetivos"
317   , "Porcentaje_juicio_valor"

```

```

313 , "Tipo.de.contrato" #Demasiados de tipo Obra que no
314 da pie a correlacion
315 , "Tipo.de.Administración"
316 , "Tipo.de.procedimiento"
317 , "Tramitación"
318 )
319
320 #Cambio la fecha por un valor númerico del año en el que se hizo
321
322 data_corrplot$Fecha.actualizacion<-as.Date(data_corrplot$Fecha.actualizacion,
323 "%d/%m/%Y")
323 data_corrplot$Fecha.actualizacion<-format(data_corrplot$Fecha.actualizacion,"%Y")
324 data_corrplot$Fecha.actualizacion<-as.numeric(data_corrplot$Fecha.actualizacion)
325
326 ## Cambio las tipo factor a tipo numeric:
327
328 data_corrplot <- data_corrplot %>% ## Demasiados de tipo Obras que dan pie a poca
correlacion
329   mutate(
330     Tipo.de.contrato = case_when(
331       Tipo.de.contrato == "Obras" ~ 1L,
332       Tipo.de.contrato == "Servicios" ~ 2L,
333       Tipo.de.contrato == "Suministros" ~ 3L,
334       Tipo.de.contrato == "Gestión de Servicios Públicos" ~ 4L,
335       Tipo.de.contrato == "Privado" ~ 5L,
336       TRUE ~ NA_integer_
337     )
338   )
339 data_corrplot <- data_corrplot %>%
340   mutate(
341     Tramitación = case_when(
342       Tramitación == "Ordinaria" ~ 1L,
343       Tramitación == "Urgente" ~ 2L,
344       Tramitación == "Emergencia" ~ 3L,
345       TRUE ~ NA_integer_
346     )
347   )
348 data_corrplot <- data_corrplot %>%
349   mutate(
350     Tipo.de.Administración = case_when(
351       Tipo.de.Administración == "Administración General del Estado" ~ 1L,
352       Tipo.de.Administración == "Comunidad Autónoma" ~ 2L,
353       Tipo.de.Administración == "Administración Local" ~ 3L,
354       Tipo.de.Administración == "Entidad de Derecho Público" ~ 4L,
355       Tipo.de.Administración == "Otras Entidades del Sector Público" ~ 5L,
356       TRUE ~ NA_integer_
357     )
358   )
359 data_corrplot <- data_corrplot %>%
360   mutate(
361     Tipo.de.procedimiento = case_when(
362       Tipo.de.procedimiento == "Abierto" ~ 1L,
363       Tipo.de.procedimiento == "Abierto simplificado" ~ 2L,
364       Tipo.de.procedimiento == "Derivado de acuerdo marco" ~ 3L,
365       Tipo.de.procedimiento == "Negociado con publicidad" ~ 4L,
366       Tipo.de.procedimiento == "Normas internas" ~ 5L,
367       Tipo.de.procedimiento == "Negociado sin publicidad" ~ 6L,
368       Tipo.de.procedimiento == "Restringido" ~ 7L,
369       TRUE ~ NA_integer_
370     )
371   )
372 ### Calculo y representacion del mapa de calor de correlaciones -----
373 matriz_correlacion <- cor(data_corrplot, method = "spearman",use =
"pairwise.complete.obs")
374
375 # Configurar el dispositivo PNG con alta calidad
376 png("corrplot.png", width = 5000, height = 3500, units = "px", res = 400)
377
378 corrplot(matriz_correlacion, method = "color", type = "full", order = "hclust",tl.cex =
0.65, tl.col = "black",addCoef.col = "black",cl.cex = 0.65,number.cex = 0.6)
379
380 # 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="table_quan.csv", sep=";", dec=".",
na="NA", row.names = T,
col.names = T)
428
429 # Selección de las variables cualitativas
430 data_cual<-BD_criterios[c("Fecha.actualizacion"
431                           , "Tipo.de.contrato" #Demasiados de tipo Obra que no da
432                           pie a correlacion
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 =
451   c("Administración General del Estado", "Comunidad Autónoma", "Administración Local",
452     "Entidad de Derecho Público", "Otras Entidades del Sector Público"))
453 
454 # Traducir los elementos factor de la variable 'Tipo.de.procedimiento'
455 data_cual$Tipo.de.procedimiento <- factor(data_cual$Tipo.de.procedimiento, levels = c(
456   "Abierto", "Abierto simplificado", "Derivado de acuerdo marco", "Negociado con
457   publicidad", "Normas internas", "Negociado sin publicidad", "Restringido"))
458 
459 # Traducir los elementos factor de la variable 'Fecha.actualizacion'
460 data_cual$Fecha.actualizacion <- as.factor(data_cual$Fecha.actualizacion)
461 
462 variables_ingles<-c("Fecha actualizacion"
463   , "Tipo de contrato" #Demasiados de tipo Obra que no da pie a
464   correlacion
465   , "Tipo de Administración"
466   , "Tipo de procedimiento"
467   , "Tramitación"
468 )
469 colnames(data_cual)<-variables_ingles
470 
471 # Cálculo de las estadísticas descriptivas
472 
473 stats2 <- lapply(data_cual[, variables_ingles], function(x) {
474   n_cases <- sum(!is.na(x))
475   prop_na <- sum(is.na(x)) / length(x)
476   table_data <- table(x, useNA = "ifany")
477   prop_data <- prop.table(table_data)
478   data.frame(Frequency = table_data, Proportion = prop_data, N_Cases = n_cases, N_As =
479     sum(is.na(x)))
480 })
481 
482 # Combinar las estadísticas en una tabla final
483 tabla_cual <- do.call(rbind, stats2)
484 write.table(tabla_cual, file="table_cual.csv", sep=";", dec=". ", na="NA", row.names = T,
485   col.names=T)
486 
487 ### Guardar en csv la base de datos final ----
488 BD_criterios_final<-BD_criterios
489 names(BD_criterios_final) <- iconv(names(BD_criterios_final), "UTF-8",
490   "ASCII//TRANSLIT") #quito tildes nombres
491 
492 BD_criterios_final <- data.frame(lapply(BD_criterios_final, function(x) { #quito
493   tildes de variables de tipo texto
494   if(is.factor(x) | is.character(x)) {
495     x <- iconv(x, "UTF-8", "ASCII//TRANSLIT") # Eliminar tildes
496     x <- gsub(";", "", x) # Eliminar caracteres ";"
497     return(x)
498   } else {
499     return(x)
500   }
501 }}))
502 write.table(BD_criterios_final, file="BD_criterios_final.csv", sep=";", dec=". ", na="NA",
503   row.names = F,col.names=T)
504 
505 ### Analisis de casos de mas de 5M € -----
506 
507 BD_criterios_5M <- filter(BD_criterios_final,
508   Presupuesto.base.sin.impuestos.licitacion.lote > 5000000)
509 write.table(BD_criterios_5M, file="BD_criterios_5M.csv", sep=";", dec=". ", na="NA",
510   row.names = F,col.names=T)

```