Práctica Limpieza y preprocesado de Datos

David Velasco Herrero

2022-11-20

1. Objetivo: Limpiar el dataset y prepararlo para posibles análisis/modelos.
2. Lectura del dataset en R.
3. Acciones de limpieza explicadas en texto y codificadas en R.

- Se valorará más que se usen data.tables en algún ejemplo, así como las funciones *apply y las librerías dplyr y tidyr.

4. Exportar desde R a un fichero local el dataset limpio resultante.
5. Conclusiones.
6. Opcionalmente, se pueden incluir gráficos si se consideraran necesarios.

Nota: El SCRIPT debe de ser REPRODUCIBLE: No debe depender de las rutas locales (directorios, paths) del equipo del alumno. Se recomienda utilizar setwd(), getwd(), rutas relativas (./, ../) y funciones de modo conveniente.

La entrega será adjuntando el archivo R Markdown al Moodle y el informe generado (PDF, Word, HTML o PPT).

2. Lectura del dataset en R.

Comprobar que el fichero esté en la ruta especificada Si está me lo guardas

```
currentDir <- getwd()
list.files(path="../datos")
```

```
## [1] "anemonefish.xls"          "beer2.csv"
## [3] "DatasetLimpio.xlsx"       "EXAMPLE_DataToClean.xlsx"
## [5] "output"                   "religions.csv"
## [7] "student-mat.csv"          "student-por.csv"
```

```
if (!file.exists("../datos"))
{stop(paste0("Se necesita que el directorio datos esté en: ",currentDir))}


ComprobarInputs <- function(path, dir,file)
{if (!file.exists(paste0(path,"/",dir)))
{stop(paste0("Se necesita que el directorio ", dir, " esté en: ",path))}
```

```
  else if (!file.exists(paste0(path, "/",dir,"/", file)))
    {stop(paste0("Se necesita que ", file," esté en: ", path, "/", dir))}}

parentPath <- dirname(currentDir)

try(ComprobarInputs(parentPath,"datos", "EXAMPLE_DataToClean.xlsx"), FALSE)

library(readxl);

mydata <- read_excel("../datos/EXAMPLE_DataToClean.xlsx",col_names = TRUE)
```

3. Limpieza de los datos

- Previamente hago instalacion de las librerias que necesito.
- Para el cambio de nombre de las columnas hago uso de rename de la librería dplyr.
- En la columna de Area, hago uso de fill down para completar los campos vacios
- En la columna Street hago el cambio del caracter å por ' '(espacio).
- Comprobando que Street y Street2 contienen los mismos datos podemos prescindir de una de ellas.

```
 libs <- c("tidyr","stringr","xlsx","writexl", "dplyr","data.table")

  for (i in libs){
    #print(i)
    if(!require(i, character.only = TRUE))
    { install.packages(i, dependencies=TRUE); library(i) }
  }
```

```
## Loading required package: tidyr

## Loading required package: stringr

## Loading required package: xlsx

## Loading required package: writexl

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: data.table
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(tidyr);
library(dplyr);
library(stringr);

#Checkeo el nombre de las columnas para saber como hacer referencia a ellas
colnames(mydata)
```

```
## [1] "Year"
## [2] "Area (use fill down)"
## [3] "Street (use find and replace to replace the odd character with a space)"
## [4] "Street 2 (in Refine use titlecase and cluster and edit)"
## [5] "Strange HTML (use unescape HTML)"
```

```
#Cambio el nombro mediante rename de la libreria dplyr
mydata <- mydata %>% rename ("Area"= colnames(mydata)[2])
mydata <- mydata %>% rename ("Street"= colnames(mydata)[3])
mydata <- mydata %>% rename ("Street2"= colnames(mydata)[4])
mydata <- mydata %>% rename ("Html"= colnames(mydata)[5])

#Relleno hacia abajo
mydata <- mydata %>% fill("Area", .direction = 'down')

#Quito el caracter 'å' por espacio
mydata$Street<- gsub("å"," ",mydata$Street)

#Pongo la primera en mayuscula de Street y de Street2
mydata$Street = str_to_title(mydata$`Street`)
mydata$Street2 = str_to_title(mydata$`Street2`)

#Limpieza de la columna html de los caracteres especiales
mydata$Html <- gsub(" "," ",mydata$Html)
mydata$Html <- gsub("&amp;","&",mydata$Html)
mydata$Html <- gsub("&lt;","<",mydata$Html)
mydata$Html <- gsub("&gt;",">",mydata$Html)
mydata$Html <- gsub("&quot;","\"",mydata$Html)
mydata$Html <- gsub("&apos;","'",mydata$Html)
mydata$Html <- gsub("&euro;","€",mydata$Html)
mydata$Html <- gsub("&ndash;","-",mydata$Html)
#Hay 6 filas que no tiene el "&" asi que hacemos dos replace para este caso
mydata$Html <- gsub("ndash","-",mydata$Html)

#Como parece que Street y Street2 son iguales lo compruebo.
#Si lo son borro Street2
if(identical(mydata$Street, mydata$Street2)){
  mydata$Street2<-NULL
}
```

- Ejemplo con la libreria dplyr

```
mydata %>% select(Year,Street) %>% tail()
```

```
## # A tibble: 6 x 2
##    Year Street
##   <dbl> <chr>
## 1  2012 Wolverhampton Railway Station
## 2  2012 Wolverhampton Train Station
## 3  2012 Wolverhampton Train Station
## 4  2012 Wright Avenue
## 5  2012 W'ton Racecourse
## 6  2012 W'ton Railway Station
```

```
#Rename
mydata <- mydata %>% rename ("AÑO"= colnames(mydata)[1])
mydata <- mydata %>% rename ("CIUDAD"= colnames(mydata)[2])
mydata <- mydata %>% rename ("CALLE"= colnames(mydata)[3])
mydata <- mydata %>% rename ("HTML"= colnames(mydata)[4])

#Cambiar todo el nombre de las columnas a mayusculas
mydata <- mydata %>%rename_with(tolower)

#Filter
mydata %>%filter(calle=="Alum Rock Road") %>%
select(1:4) %>% head()
```

```
## # A tibble: 6 x 4
##     año ciudad     calle          html
##   <dbl> <chr>      <chr>          <chr>
## 1  2011 Birmingham Alum Rock Road "€300"
## 2  2011 Birmingham Alum Rock Road "alcester road"
## 3  2011 Birmingham Alum Rock Road "\"That silly man\""
## 4  2011 Birmingham Alum Rock Road "<html>"
## 5  2012 Birmingham Alum Rock Road  <NA>
## 6  2012 Birmingham Alum Rock Road  <NA>
```

```
#Arrange(Ordeno el dataset por la columna calle de la A a la Z)
mydata<-mydata %>% arrange(calle)
```

- Ejemplo con la libreria tidyr

```
library(tidyr)
str(mydata)
```

```
## tibble [5,279 x 4] (S3: tbl_df/tbl/data.frame)
##  $ año   : num [1:5279] 2011 2011 2012 2012 2011 ...
##  $ ciudad: chr [1:5279] "Wolverhampton" "Solihull" "Birmingham" "Birmingham" ...
##  $ calle : chr [1:5279] "A And E New Cross Hospital" "A+E Soliohull Hospital" "A38 Expressway, Birmi
##  $ html  : chr [1:5279] NA NA NA NA ...
```

```
unite(mydata, "Direccion", c(calle, ciudad), sep=", ")
```

```
## # A tibble: 5,279 x 3
##      año Direccion                                    html
##    <dbl> <chr>                                        <chr>
##  1  2011 A And E New Cross Hospital, Wolverhampton <NA>
##  2  2011 A+E Soliohull Hospital, Solihull          <NA>
##  3  2012 A38 Expressway, Birmingham, Birmingham    <NA>
##  4  2012 A38 Northfield, Birmingham                <NA>
##  5  2011 Abbey Road, Sandwell                      <NA>
##  6  2011 Abdon Ave, Birmingham                     <NA>
##  7  2012 Abdon Ave, Birmingham                     <NA>
##  8  2012 Abelwell St Walsall, Walsall              <NA>
##  9  2012 Abelwell Street Walsall, Walsall          <NA>
## 10  2012 Aberdeen St Winson Green, Birmingham      <NA>
## # ... with 5,269 more rows
```

- Para practicar con datatable calculo cuantos registros son de 2011 y de 2012 para checkear si se me ha colado algún otro año

```
library(data.table)
dtdata <- as.data.table(mydata)
class(dtdata)
```

```
## [1] "data.table" "data.frame"
```

```
#Contar cuantas de la variable STREET2 se llama Alum Rock Road

dtdata[calle =="Alum Rock Road", .N]
```

```
## [1] 7
```

```
datos2011 <-dtdata[año =="2011", .N]
datos2012 <-dtdata[año =="2012", .N]
registros <- datos2011 + datos2012

#Todos los registros son de 2011 y de 2012
```

- Mediante funciones *apply calcula cuantas filas tienen algún Na

```
mydataNA <- mydata[rowSums(is.na(mydata)) > 0, ]
dim(mydataNA)[1]
```

```
## [1] 5243
```

```
#Quito los NA

mydatab<- mydata
noMissing <- complete.cases(mydatab)
```

```
mydatabNoNa <- mydatab[noMissing,]
mydataNA <- mydatab[!noMissing,]
#Obtengo dos dataframes.
#mydatabNoNa el cual no tienen en ninguna fila un -NA-
#mtdataNA el cual tiene en alguna de las filas un -NA-
#Al ver que están todos en html. Se borra la columna porque no es útil

mydata$html<-NULL
```

4.Exportar datos a un fichero Excel.

```
require(writexl)

write_xlsx(mydata, "../datos/DatasetLimpio.xlsx")
```

5.Conclusiones

- Las columnas Street y Street 2 son iguales por los que se puede borrar una de ellas.
- Se detecta que la columna HTML no aporta nada y tiene muchas celdas sin valor asi que se puede borrar.
- Todos los registros son de 2011(2868) y de 2012(2411) Gracias a los graficos del apartado 6 se observa que:
  - Casi la mitad de los registros son de Birmingham
  - Hay mas registros de 2011 que de 2012

6.Graficos y descriptivos

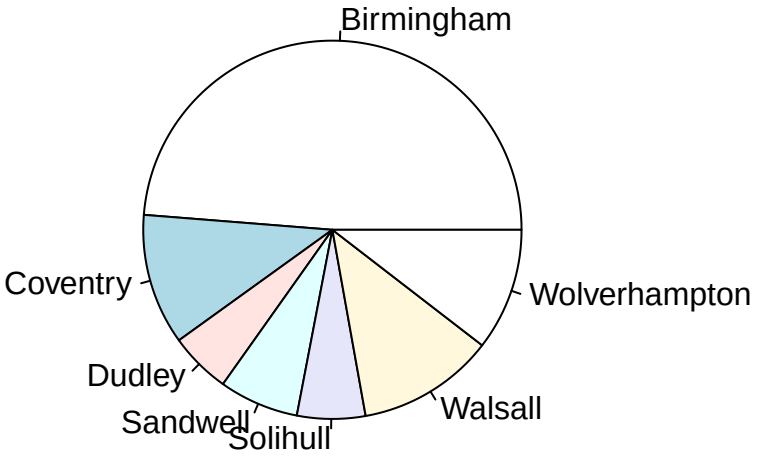
```
table(mydata$año)
```

```
##
## 2011 2012
## 2868 2411
```

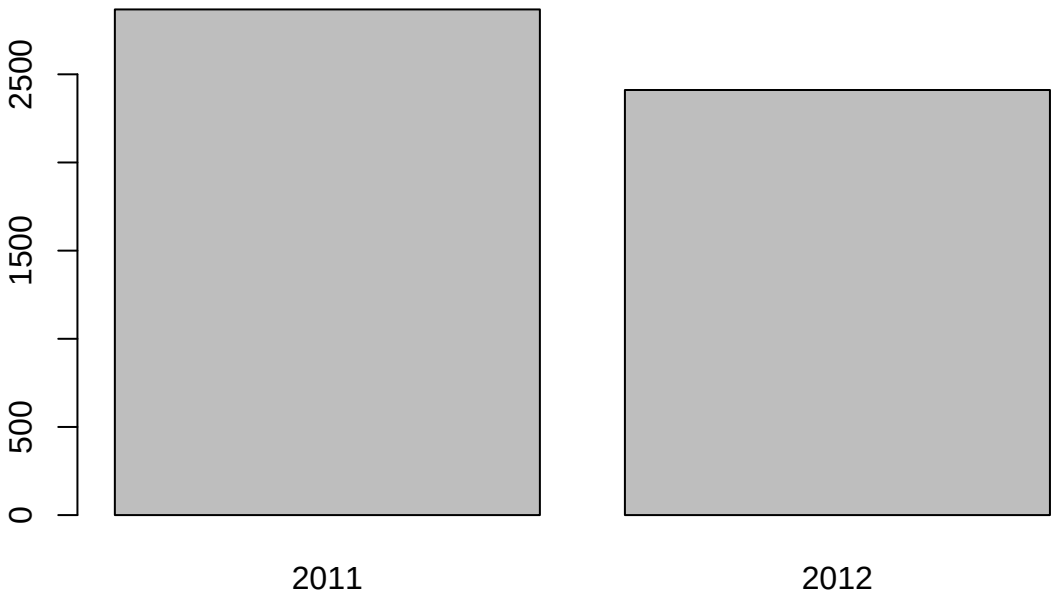
```
#Tabla cruzada entre año y ciudad
table(mydata$año, mydata$ciudad)
```

```
##
##        Birmingham Coventry Dudley Sandwell Solihull Walsall Wolverhampton
##   2011       1376      348    154      192      167     338           293
##   2012       1197      243    123      167      141     281           259
```

```
pie(table(mydata$ciudad))
```




```
barplot(table(mydata$año))
```


0
500
1500
2500
2011
2012