Prueba Técnica Analista de datos ID 382 GITIAD

Juan Esteban Grimaldos León

2024-07-05

## Importar data

Como primera medida se importa las bases de datos de excel a R. Recordemos que el directorio es src. Para efectos prácticos, se realiza una conversión manual de los datos en xls a xlsx.

library(readxl)  
print(getwd()) # dejamos el directorio de trabajo en la carpeta src

## [1] "D:/projects/pruebas-adres-juan-grimaldos/src/scripts"

data\_m <- read\_excel("../data/Municipios.xlsx")  
data\_p <- read\_excel("../data/Prestadores.xlsx")

# Limpieza de datos

## limpieza de datos Departamentos

Como en el Excel Municipios, tenemos el código del departamento pero el nombre de los departamentos está mal escrito, vamos a corregirlo. Para ello debemos insertar la información de los departamentos en la base de datos de municipios.

# generamos dataframes auxiliares  
mask1 <- unique(data.frame(data\_p$depa\_nombre))  
mask2 <- data.frame(data\_m$Departamento, data\_m$Dep)

Como tenemos en Municipios varios datos con caracteres especiales, vamos a limpiar un poco los datos. En adición, vamos a unificar la información con Prestadores.

# juntamos estos dataframes en uno solo  
unique\_df <- unique(merge(  
 mask1, mask2, by.x = "data\_p.depa\_nombre",   
 by.y = "data\_m.Departamento", all.x = TRUE))  
  
# regeneramos el indice  
row.names(unique\_df) <- NULL  
  
# visualizamos las primeras 5 filas  
head(unique\_df)

## data\_p.depa\_nombre data\_m.Dep  
## 1 Amazonas 91  
## 2 Antioquia 05  
## 3 Arauca 81  
## 4 Atlántico 08  
## 5 Barranquilla <NA>  
## 6 Bogotá D.C 11

Como ya tenemos la información de los departamentos con código, vamos a unir esta información con la base de datos de municipios original.

# comparamos las dimensiones de los dataframes antes y después de la unión  
print(dim(data\_m))

## [1] 1118 8

# juntamos los dataframes, la llamaremos df\_m para poder comparar posibles  
# duplicados  
df\_m <- merge(  
 data\_m, unique\_df, by.x = "Dep", by.y = "data\_m.Dep", all.x = TRUE)  
  
# visualizamos las primeras 5 filas  
dim(df\_m)

## [1] 1118 9

Vamos a verificar si hay valores nulos en la columna data\_p.depa\_nombre.

na\_rows <- df\_m[is.na(df\_m$data\_p.depa\_nombre), ]  
dim(na\_rows)

## [1] 0 9

Como no hay valores nulos en la columna data\_p.depa\_nombre, procedemos a realizar el mismo procedimiento pero con la información de municipios.

## limpieza de datos Municipios

# importamos librerias  
library(dplyr)

##   
## Adjuntando el paquete: 'dplyr'

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

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

library(stringr)  
# generamos dataframes auxiliares  
mask1 <- unique(data.frame(data\_p$muni\_nombre))  
mask2 <- data.frame(data\_m$Municipio, data\_m$Depmun)  
  
mask2 <- mask2 %>%  
 mutate(data\_m.Municipio = str\_replace\_all(data\_m.Municipio, "\\s{2,}", " ") %>%  
 str\_replace\_all("[^a-zA-ZáéíóúüñÁÉÍÓÚÜÑ ]", "") %>%  
 str\_trim(side = "left") %>%  
 str\_trim(side = "right") %>%  
 str\_squish() %>%  
 str\_to\_upper() %>%  
 str\_replace\_all("Bogotá D C", "Bogotá D.C"))  
  
# juntamos estos dataframes en uno solo  
unique\_df <- unique(merge(  
 mask1, mask2, by.x = "data\_p.muni\_nombre",   
 by.y = "data\_m.Municipio", all.x = TRUE))  
  
# comparamos las dimensiones de los dataframes antes y después de la unión  
print(paste("Rows of data\_m before merge are:", dim(data\_m)[1]))

## [1] "Rows of data\_m before merge are: 1118"

# juntamos los dataframes, la llamaremos df\_m para poder comparar posibles  
# duplicados  
df\_cleaned <- merge(  
 df\_m, unique\_df, by.x = "Depmun", by.y = "data\_m.Depmun", all.x = TRUE)  
  
# visualizamos las primeras 5 filas  
print(paste("Rows of data\_m after merge are:", dim(df\_cleaned)[1]))

## [1] "Rows of data\_m after merge are: 1118"

# verificamos valores nulos  
na\_rows <- df\_cleaned[is.na(df\_cleaned$data\_p.muni\_nombre), ]  
dim(na\_rows)

## [1] 138 10

Debido a que tenemos valores nulos en la columna data\_p.muni\_nombre, vamos a validar los 138 valores nulos.

readRenviron("../../.Renviron")  
library(RSocrata)  
  
# Access environment variables  
app\_token <- Sys.getenv("SOCRATA\_APP\_TOKEN")  
email <- Sys.getenv("SOCRATA\_EMAIL")  
password <- Sys.getenv("SOCRATA\_PASSWORD")  
  
# Use the environment variables  
df <- read.socrata(  
 "https://www.datos.gov.co/resource/xdk5-pm3f.json",  
 app\_token = app\_token,  
 email = email,  
 password = password  
)  
  
# View data  
head(df)

## region c\_digo\_dane\_del\_departamento departamento  
## 1 Región Eje Cafetero - Antioquia 5 Antioquia  
## 2 Región Eje Cafetero - Antioquia 5 Antioquia  
## 3 Región Eje Cafetero - Antioquia 5 Antioquia  
## 4 Región Eje Cafetero - Antioquia 5 Antioquia  
## 5 Región Eje Cafetero - Antioquia 5 Antioquia  
## 6 Región Eje Cafetero - Antioquia 5 Antioquia  
## c\_digo\_dane\_del\_municipio municipio  
## 1 5.001 Medellín  
## 2 5.002 Abejorral  
## 3 5.004 Abriaquí  
## 4 5.021 Alejandría  
## 5 5.03 Amagá  
## 6 5.031 Amalfi