Importacion

Josep Vicent MORALES

2025-04-03

Contents

0.1 1.Librerias

Conjunto de librerías a utilizar

```
library(pdftools)
```

```
## Warning: package 'pdftools' was built under R version 4.4.2
```

```
## Using poppler version 23.08.0
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.4.2
```

```
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(ggplot2)
library(stringr)
```

0.2 2. Carga Pdfs

Los tickets que queremos analizar se encuentran en formato .pdf, y se encuentran en la carpeta data, por tanto en este bloque importamos todos los .pdf de la carpeta.

```
dir <- "data/"
# Charge all pdfs from folder data
pdfs <- list.files(path = dir, pattern = "\\.pdf$", full.names = TRUE)%>%
        lapply(pdf_text)
```

0.3 3. Transformación

En este apartado realizamos el tratamiento de los datos para dividirlos y reorganizarlos para posteriormente poder consultarlos de manera más fácil y eficiente.

Como hemos observado, en la carpeta data se incluyen tickets que no son de Mercadona o directamente no son tickets por lo que antes de proceder con el transformado, vamos a eliminar estos tickets erróneos.

```
# Filtramos solo los que contienen "MERCADONA"
pdfs <- pdfs[sapply(pdfs, function(x) any(grepl("MERCADONA,", x)))]

# Cabe recalcar que filtramos por "MERCADONA," con una coma ya que hemos encontrado dos tickets con for
# Pero que si pertenecían a Mercadona, debían tener un fallo en la codificación
```

0.3.1 3.1 Tabla Precios

	Descripción	P. Unit	Importe
1	CAFE MOLIDO NATURAL		3,20
1	LAVAVAJILLAS ULTRA		1,90
1	CEBOLLA 2 KG		2,79
1	CACAHUETE SIN SAL		1,60
1	NUEZ CASCARA NATURAL		2,90
1	40 B. CONG. MEDIANAS		0,85
1	BOLSAS.G CONGELADO		1,10
1	PISTACHO TOST 0% SAL		3,35
1	AVELLANA TOSTADA		3,10
2	PAÑUELO CAJA	1,30	2,60
1	NARANJA 5 KG.		5,95

Figure 1: Figura 1: Vista de productos

En la Figura se puede apreciar qué estructura sigue la tabla de manera genérica.

Como las comidas que van por peso como verduras frutas o pescados, se organizan de otra manera en los tickets, mostrandose en una líena la cantidad, y el nombre del elemento, y en la de abajo se muestra el peso del elemento en Kg, seguido del precio en Kg/€ y finalmente a la derecha el precio pagado, se ha creado esta función para poder implementarlo en nuestra tabla.

```
weight_food <- function(df){
      r_remove <- c()

      for (i in c(1:nrow(df))) {
        if (df$Descripcion[i] == "PESCADO" && i + 2 <= nrow(df)){
            df$Cantidad[i +1] <- 1 # Not specify in ticket, by default will be 1
            df$Tipo[i+1] <- "Pescado"
            df$Precio[i+1] <- gsub("[^0-9,]","",df$Precio[i + 2])
            df$Importe[i+1] <- gsub("[^0-9,]","",df$Importe[i + 2])
            df$Peso[i+1] <- gsub("[^0-9,]","",df$Descripcion[i + 2])
            # Rows to remove
            r_remove <- c(r_remove, i, i +2)
        }else{
            if (i < nrow(df) && is.na(df$Precio[i])){
                df$Precio[i] <- gsub("[^0-9,]","",df$Precio[i + 1])
                df$Importe[i] <- gsub("[^0-9,]","",df$Importe[i + 1])
                df$Peso[i] <- gsub("[^0-9,]","",df$Descripcion[i + 1])
                # Rows to remove
                r_remove <- c(r_remove, i+1)
          }
        }
      }
      if (length(r_remove)>0){
          df <- df[-r_remove,]
      }
      return(df)
    }
```

Extraemos por completo los datos referentes a los elementos comprados precios y unidades, y se transforma en una tabla esto se realiza para todos los distintos tickets. Extrae tabla (en pruebas)

Extracción de tabla del pdf.

```
# List to collect all Price Tables
l_compras <- list()

# Price Table Extraction
for (i in pdfs){
  # We take from Description to TOTAL
  tabl <- str_extract(i,regex("Descripción(.*?)TOTAL",dotall = TRUE))
  tabl <- unlist(strsplit(tabl, "\n")) # Division by lines
  tabl <- tabl[-c(1,length(tabl))] # Extraction head and Total line

  if (any(str_detect(i,"ENTRADA"))){
    tabl <- tabl[-c(length(tabl),length(tabl) - 1)]
  }
  l_compras[[length(l_compras) + 1]] <- tabl
}
```

Transformación datos en tabla

```
t_compras <- list()
```

```
i <- l_compras[[2]]
a <- 0 # ticket index counter
for(i in l_compras){
  tabl <- list()
  a <- a + 1 # ticket index increment
    for (e in seq_along(i)){
        line <- i[[e]]

        cantidad <- str_extract(line, "^\\s{0,2}\\d+\\s{1,}")%>%str_trim()
        line <- sub("^\\s{0,2}\\d+\\s{1,}", "", line)%>%str_trim()

        descripcion <- str_extract(line, "^[^\\s].*?\\s{3,}") %>% str_trim()
        line <- sub("^[^\\s].*?\\s{3,}", "", line)%>% str_trim()

        # Next Line Elemts With Weight description is na
        #with that character search
        if (is.na(descripcion)){
            descripcion <- line
            precio <- NA
            importe <- NA

            if (!is.na(line) && line == "PESCADO"){
                tipo <- "Pescado"
            }else{
                tipo <- "Fruta o Verdura"
            }
        }else{
            numeros <- str_split_fixed(line, "\\s{2,}",2)
            precio <- numeros[1]
            importe <- numeros[2]
            tipo <- ""
        }

        dt1 <- data.frame(index = a, #Ticket index
                          Cantidad = cantidad,
                          Descripcion = descripcion,
                          Precio = precio,
                          Importe = importe,
                          Peso = 1, # El valor de Peso será 1 predefinido
                          Tipo = tipo) # (Fruta o verdura) o (Pescado)

        tabl <- append(tabl, list(dt1))
    }



  df <- do.call(rbind, tabl)
  df <- as.data.frame(df)
  df <- weight_food(df)
  # Añadimos el data.frame final a la lista de compras
  t_compras <- append(t_compras, list(df))
}
```

Estos datos presentan algunos problemas, entre ellos, los decimales estan separados por "," en vez de por ".", y a su vez, las tablas no estan llenas de datos, ya que algunos datos del Importe salen desplazados a la columna de Precio debido a que esta está vacía cuando la cantidad de unidades es 1 ya que es el mismo valor, por tanto en el siguiente bloque arreglaremos estos 2 problemas.

```
for (i in seq_along(t_compras)){
   t_compras[[i]] <- t_compras[[i]]%>%# Change <NA> in Importe to Precio
    mutate(Importe = ifelse(Importe == '', Precio, Importe)) %>%
    # Change "," to "." and change type to numeric
    mutate(across(-c(Descripcion, Tipo, index), ~ as.numeric(gsub(",", ".", .))))
}
```

Ahora para una consulta más fácil, unimos todos los tíquets en una misma tabla y usamos la columna index para poder diferenciarlos entre ellos con mayor facilidad.

```
df_p <- data.frame(index = character(),
                   Cantidad = character(),
                   Descripcion = character(),
                   Precio = character(),
                   Importe = character(),
                   Peso = character(),
                   Tipo = character())

for (i in  t_compras){
  df_p <- rbind(df_p,i)
}
```

0.3.2 3.2 Datos Generales

Aqui vamos a tratar de recoger y colocar en una tabla la información general de los tiquets como dirección, teléfono... entre otros.

Extracción bloque cabecera:

```
# List to collect all ticket head elements
l_head <- list()

# Price Table Extraction
for (i in pdfs){
  # We take from MERCADONA to the start of the price table (Descripción)
  tabl <- str_extract(i,regex("MERCADONA(.*?)Descripción",dotall = TRUE))
  tabl <- unlist(strsplit(tabl, "\n")) # Division by lines
  tabl <- tabl[-c(1,length(tabl))] # Extraction Mecadona line and table start line

  l_head[[length(l_head) + 1]] <- tabl
}
```

En este bloque organizamos toda la información de la cabecera de los tiquets en un unico dataframe

```
df_h <- data.frame(index = numeric(),
                   Direccion = character(),
                   Ciudad = character(),
                   CP = character(),
```

```
                  Telefono = character(),
                  Fecha = character(),
                  Hora = character(),
                  OP = character(),
                  Num_Fac_Simp = character(),
                  stringsAsFactors = FALSE
                  )

a <- 0
for (i in l_head){
    a <- a+1
    df_h <- rbind(df_h, data.frame(
    index = a,
    Direccion = i[[1]]%>% str_trim(),
    Ciudad = i[[2]] %>% sub("\\d+", "", .)%>%str_trim(),
    CP =i[[2]] %>% str_extract("\\d+")%>%str_trim(),
    Telefono = i[[3]] %>% sub("TELÉFONO:", "",.)%>%str_trim,
    Fecha = i[[4]] %>% str_extract("\\b\\d{2}/\\d{2}/\\d{4}\\b")
                  %>% as.Date(format = "%d/%m/%Y"),
    Hora = i[[4]] %>% str_extract("\\d{2}[:]\\d{2}")%>% str_trim(),
    OP = i[[4]] %>% str_extract("OP: \\d+") %>% sub("OP: ","",.)%>% str_trim(),
    Num_Fac_Simp = i[[5]] %>% sub("FACTURA SIMPLIFICADA: ","",.) %>% str_trim()
  ))
}
```

Con este bloque extraemos de cada pdf el precio total pagado en euros y lo añadimos al dataframe anterior de datos generales.

```
TOTAL_PAGO <- c()
PARKING_ENTRADA <- c()
PARKING_SALIDA <- c()

for (i in pdfs){
   # Total Payment End Table
   TP <- i %>% str_extract("TOTAL \\(€\\)\\s*(\\d+[.,]\\d{2})") %>%
                sub("TOTAL \\(€\\)\\s*","",.)%>% sub(",",".",.) %>% as.numeric()

   # Parking Entrance
   PE <- i %>% str_extract("ENTRADA \\d{2}[:]\\d{2}") %>%
               sub("ENTRADA","",.) %>%
               str_trim() %>%
               ifelse(is.na(.), "00:00", .)

   # Parking Exit
   PS <- i %>% str_extract("SALIDA \\d{2}[:]\\d{2}") %>%
               sub("SALIDA","",.) %>%
               str_trim() %>%
               ifelse(is.na(.), "00:00", .)

   TOTAL_PAGO <- rbind(TOTAL_PAGO,TP)
   PARKING_ENTRADA <- rbind(PARKING_ENTRADA,PE)
   PARKING_SALIDA  <- rbind(PARKING_SALIDA , PS)
}
```

```
df_h <- cbind(df_h,TOTAL_PAGO,PARKING_ENTRADA,PARKING_SALIDA )
```

0.3.3 3.3 Tabla IVA

En este punto extraemos las tablas de los distintos IVA aplicados en los tiquet.

Con este bloque extraemos como bloque la tabla entera desde el ticket colocandolas en una lista que referencia el tiquet.

```
l_iva <- list()

# Price Table Extraction
for (i in pdfs){
  # We take from IVA to TOTAL
  tabl <- str_extract(i,regex("TARJETA BANCARIA\\s{3,}(.*?)TOTAL",dotall = TRUE))
  tabl <- unlist(strsplit(tabl, "\n")) %>% str_trim() # Division by lines
  tabl <- tabl[-c(1,2,3,length(tabl))] # Extraction head and Total
  tabl <- tabl %>% sub(",",".",.) # change "," to "." for converting to number type
  tabl <- tabl %>% sub("%", "",.) # eliminates "%" for converting to number type

  l_iva[[length(l_iva) + 1]] <- tabl
}
```

En este bloque, colocamos todas las tablas en un unico data frame con una variable de índice para poder saber a que tiquet hacen referencia y enlazandolos con los otros data frames que hemos creado anteriormente.

```
df_iva <- data.frame(index = numeric(),
                     IVA = character(),
                     BASE = character(),
                     CUOTA = character(),
                     stringsAsFactors = FALSE
                     )

a <- 0
for (i in l_iva){
  a <- a +1
  for (e in i){
    iva <- e %>% str_split("\\s{1,}",simplify = TRUE)
    df_iva <- rbind(df_iva,data.frame(
                            index = a,
                            IVA = iva[1],
                            BASE = iva[2],
                            CUOTA = iva[3]
  ))
  }
}
```