Mini Proyecto TD 2024

Entrega final

Víctor Álvarez Palomares,
Sergio Martínez Yagüe,
Ferran Medina Mompó,
Carles Pascual i Sivera,
y Úrsula Casaus Fabra

2024-05-14

Introducción

En esta primera parte de la entrega, vamos a importar y a analizar la información que nos aportan una serie de tickets de la compra que se nos proporcionan desde la asignatura.

Carga de librerías y datos necesarios para el análisis

Para empezar, se cargan todas las librerías necesarias para la realización del código. Esto se hace de manera más elegante utilizando el paquete pacman de RStudio. A continuación, se realiza la carga del conjunto de datos, los cuales se encuentran en formato .pdf. No obstante, no podemos importarlos desde este formato, así que vamos a convertirlos a .txt mediante una función que hemos creado en Python con la librería PyPDF2. De esta forma, obtenemos los ficheros .txt para poder importarlos en RStudio y trabajar con ellos.

Características generales de los datos

Los datos de los tickets se van a encontrar en listas de data frames. Cada dataframe contiene registros de 4 variables: "producto" (de tipo caracter), "unidades", "precio_completo" y "precio_individual" (estas de tipo numérico). El número de registros depende de cada data frame, ya que el número de productos varía según la compra.

Análisis de "missing data" en nuestro conjunto de interés

Por suerte, en nuestros conjuntos de datos no hay "missing data", ya que se tratan de tickets de la compra en los cuales cada producto tiene un nombre y precio asignados, y los tickets contienen la cantidad de veces que se añade a la compra.

Importación de los datos

Como ya hemos explicado previamente, hemos convertido nuestros ficheros a .txt desde .pdf. Ahora, vamos a crear otra función para separar las cadenas de caracteres de los ficheros y así importar los datos en los dataframes de forma que nos resulte más fácil trabajar con ellos. También hemos de crear un par de funciones que lean la fruta de forma correcta, pues el formato es distinto al resto de productos.

Preguntas

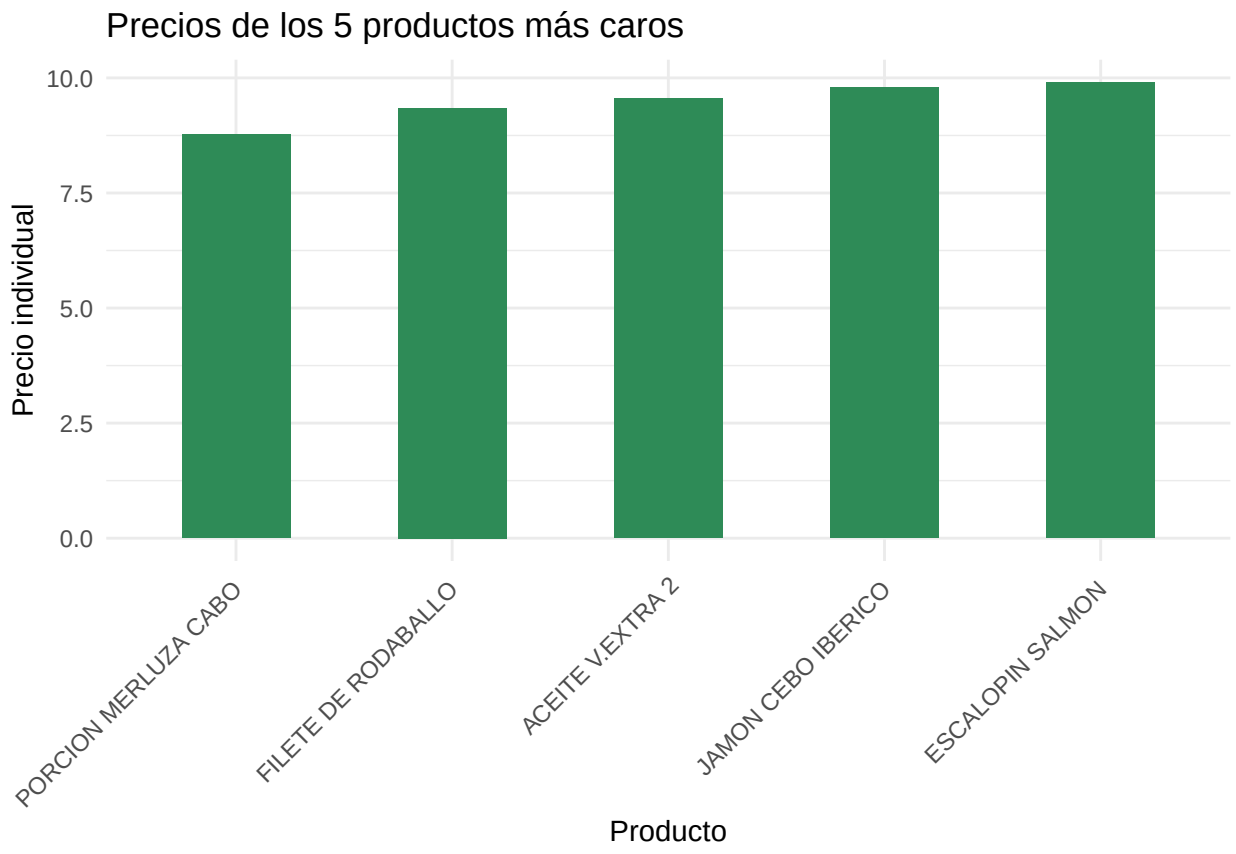
A continuación vamos a plantearnos diferentes cuestiones que nos podrían venir a la cabeza a la hora de hacer un análisis de los tickets de la compra.

1) ¿Cuáles productos son los más caros?
2) ¿Qué productos se compran más?
3) ¿A qué hora del día hay más compras?
4) ¿Qué días hay más compras?
5) ¿Cuántos productos se compran de media en una compra?
6) ¿Qué productos cambian de precio (por kg)?
7) ¿Qué tipo de IVA recauda más dinero?

1) ¿Cuáles son los productos más caros?

Para poder obtener esta información, vamos a trabajar solo con las variables "producto" y "precio_individual". Vamos a extraer estas columnas de los data frames y vamos a combinar toda esta información en un único data frame que ordenaremos posteriormente de forma descendente. Cuando ya tengamos la información ordenada, crearemos un gráfico que muestre los 5 productos más caros.

```
# Ahora que ya tenemos la información que nos interesa ordenada, vamos a
# mostrarla con una gráfica.

# Como solo queremos mostrar los 5 primeros, los definimos en otro data frame.

df_precios_plot <- head(df_precios, 5)

grafica <- ggplot(df_precios_plot, aes(x = reorder(producto, precio_individual),
                                       y = precio_individual)) +
  geom_bar(stat = "identity", fill = "seagreen", width = 0.5) +
  labs(title = "Precios de los 5 productos más caros",
       x = "Producto",
       y = "Precio individual") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

grafica
```




```
# Si queremos descargarnos el gráfico:

# ggsave("top_5_productos_mas_caros.jpeg", grafica, width = 10, height = 6.5)
```

Como podemos observar en la gráfica, los cinco productos más caros son: el escalopín de salmón (9,90 €), el jamón de cebo ibérico (9,79 €), el aceite de oliva virgen extra (9,55 €), filete de rodaballo (9.35 €) y una porción de merluza de cabo (8,77 €). Estos resultados tienen sentido, ya que los productos mostrados son de alta calidad y de producción costosa. También podemos apreciar que se encuentran más o menos en el mismo intervalo de precio.

2) ¿Qué productos se compran más?

Para calcular los productos más vendidos a partir de los tickets de compra, primero vamos a contabilizar la cantidad de cada producto que se compró en cada ticket, lo que nos da un recuento de unidades para cada producto. Tras ello, sumaremos las unidades de cada producto a través de todos los tickets para obtener un total de unidades vendidas para cada producto. Finalmente, ordenamos los productos por el total de unidades vendidas para identificar los productos más vendidos. Es decir, aquellos productos con más unidades vendidas son los productos más vendidos.

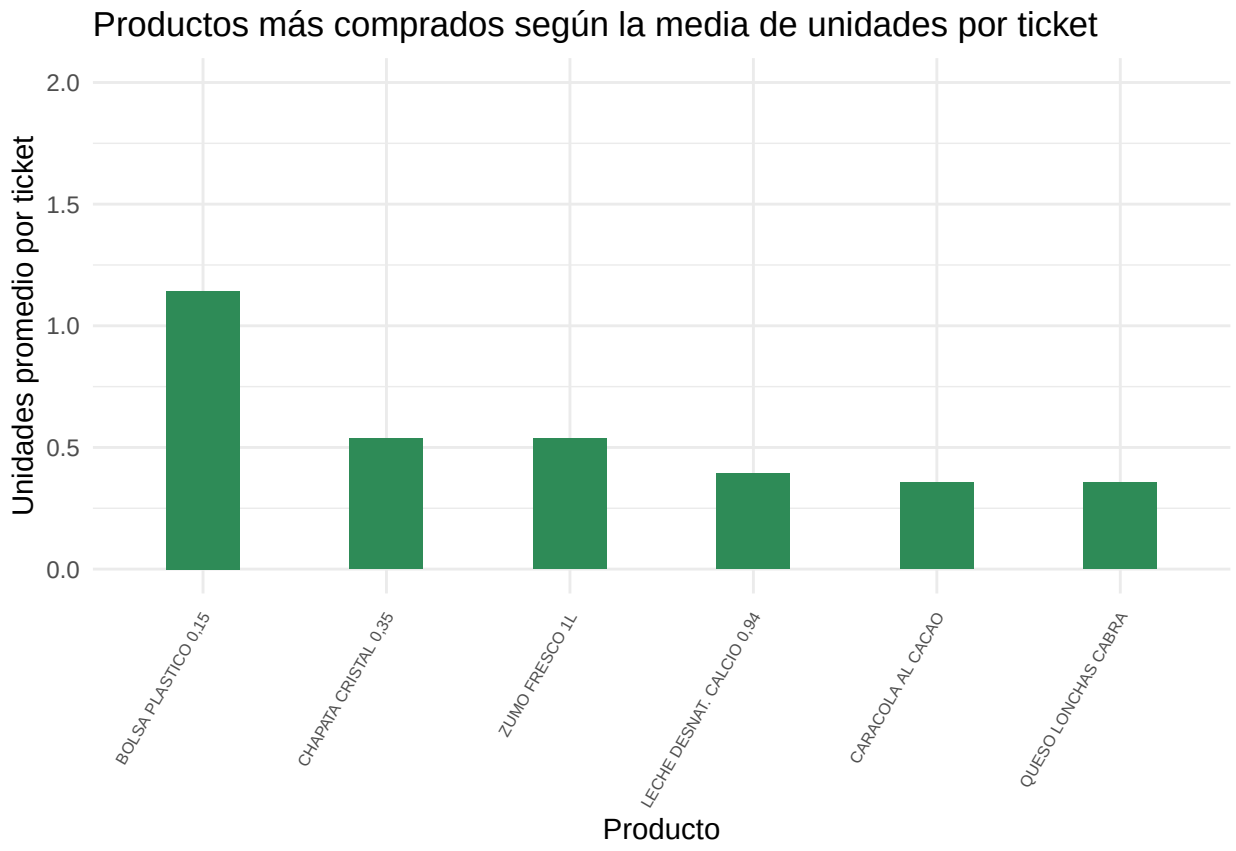
```
# Utilizamos la función que ya tenemos para obtener la lista de dataframes
lista_df <- lista

# Unimos todos los dataframes de los tickets en uno solo
todos_los_tickets <- do.call(rbind, lista_df)
```

```r
# Eliminamos la última fila del dataframe
todos_los_tickets <- head(todos_los_tickets, -1)

# Convertimos la columna 'unidades' a numérica
todos_los_tickets$unidades <- as.numeric(as.character(
  todos_los_tickets$unidades))

# Eliminamos las filas con NA (que son las filas que no se pudieron convertir
#a numérico)
todos_los_tickets <- todos_los_tickets[!is.na(todos_los_tickets$unidades), ]

# A continuación, agrupamos los datos por producto y calculamos las unidades
#totales de cada producto
productos_por_unidades <- todos_los_tickets %>%
  group_by(producto) %>%
  summarise(
    unidades_total = sum(unidades),
    .groups = 'drop'
  )
# Calcula el número total de tickets
num_tickets <- length(lista_df)

# Calcula la media de unidades por ticket para cada producto
productos_por_unidades <- productos_por_unidades %>%
  mutate(unidades_promedio_por_ticket = unidades_total / num_tickets)

# Ahora, ordenamos los productos por sus unidades promedio por ticket
productos_ordenados <- productos_por_unidades %>%
  arrange(desc(unidades_promedio_por_ticket))

# Finalmente, podemos ver los productos que se compran más según la media
#de unidades por ticket
top_productos <- head(productos_ordenados)
top_productos
```

```
## # A tibble: 6 x 3
##   producto                     unidades_total unidades_promedio_por_ticket
##   <chr>                                 <dbl>                        <dbl>
## 1 "BOLSA PLASTICO 0,15 "                   32                        1.14
## 2 "CHAPATA CRISTAL 0,35 "                  15                        0.536
## 3 "ZUMO FRESCO 1L "                        15                        0.536
## 4 "LECHE DESNAT. CALCIO 0,94 "             11                        0.393
## 5 "CARACOLA AL CACAO "                     10                        0.357
## 6 "QUESO LONCHAS CABRA "                   10                        0.357
```

```r
# Creamos un gráfico
grafico <- ggplot(top_productos, aes(x = reorder(producto,
                                                 -unidades_promedio_por_ticket),
                                     y = unidades_promedio_por_ticket)) +
  geom_bar(stat = "identity", fill = "seagreen", width = 0.4) +
  # cambia el ancho de las barras a 0.5
  theme_minimal() + # usa un tema minimalista
  theme(axis.text.x = element_text(angle = 60, hjust = 1, size = 6)) +
```

```
  ylim(0,2) +
  labs(x = "Producto", y = "Unidades promedio por ticket",
       title = "Productos más comprados según la media de unidades por ticket")
grafico
```



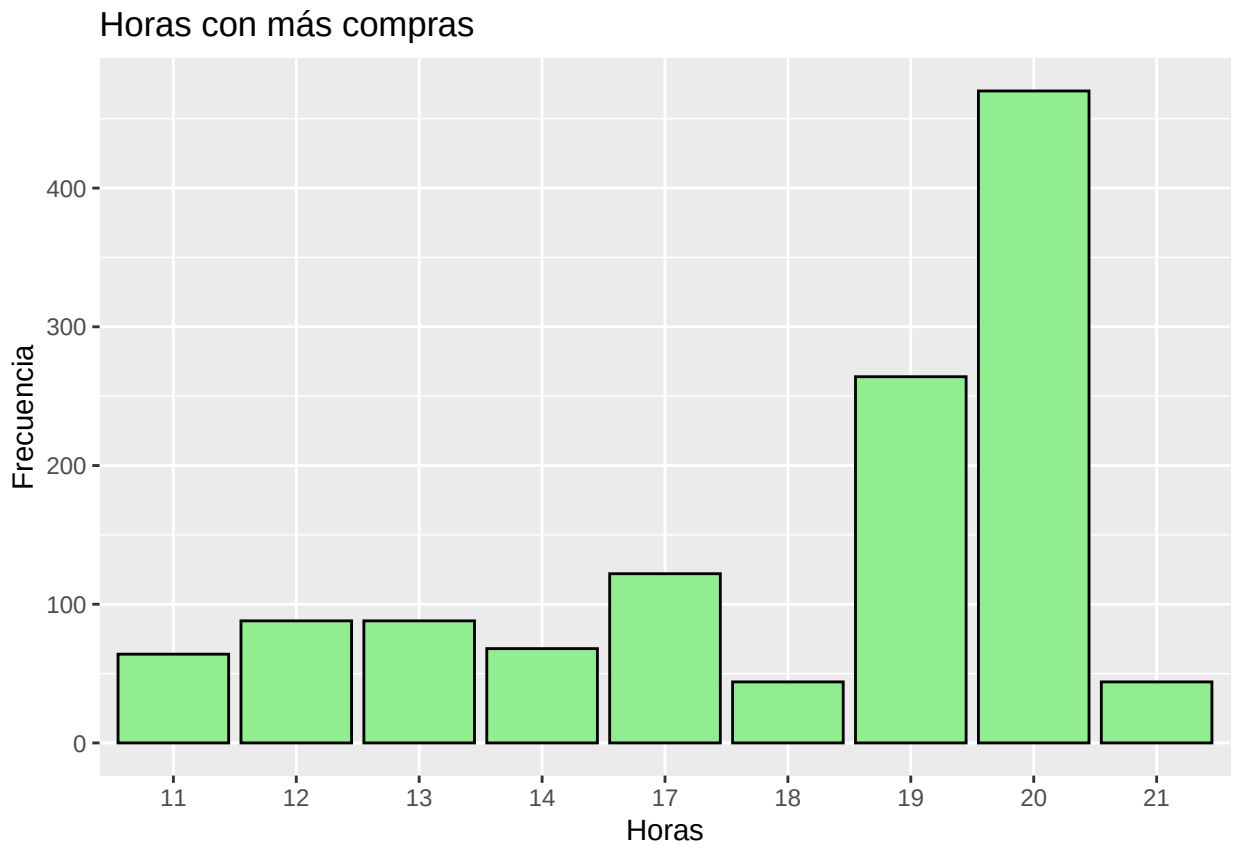

```
#Si queremos descargarnos el gráfico
#ggsave("productos_mas_comprados.jpeg", grafico, width = 10, height = 6.5)
```

Como se puede observar en la gráfica, las bolsas de plástico son el producto más vendido. Con más de 1.1 bolsas de plástico compradas por ticket. Además productos como la chaparta cristal, el zumo fresco, la leche descnatada, la caracola al cacao o el queso a lonchas de cabra también son los más vendidos entre los tickets analizados.

3) ¿A qué hora del día hay más compras?

```
# Seleccionar la columna "Horas" como factor

compras_por_hora <- tidy_ticket %>% select(Horas = Horas)

# Crear el gráfico de barras
plot <- ggplot(data = compras_por_hora, aes(x = Horas)) +
    geom_bar(fill = "lightgreen", color = "black") +
```

```
    labs(title = "Horas con más compras", x = "Horas", y = "Frecuencia")
plot
```

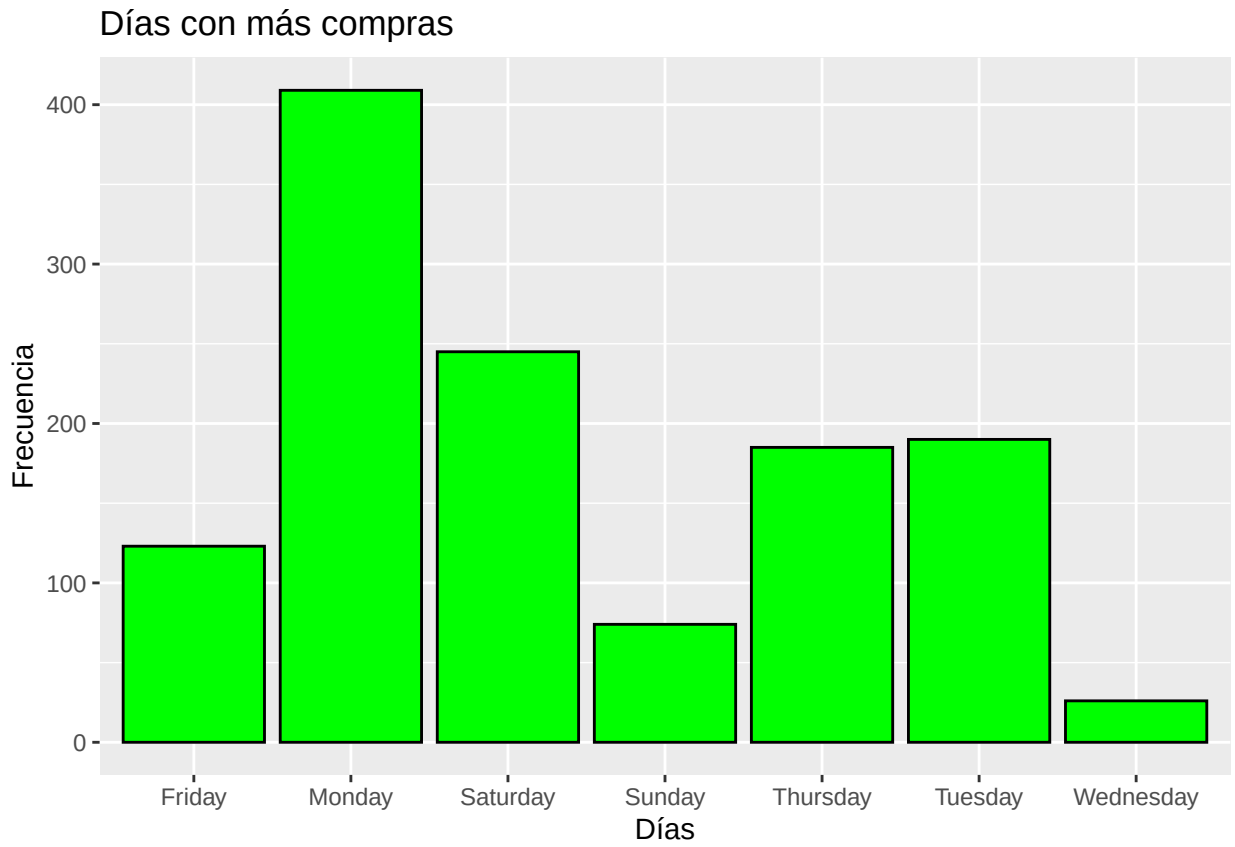



```
ggsave("pregunta3.jpg", plot = plot, width = 8, height = 6, dpi = 300)
```

Como podemos observar las horas con más compras son las 17 de la tarde, que eso nos puede demostrar que la gente va de compras a la hora que acaba de trabajar.

4) ¿Qué días hay más compras?

```
compras_por_dia <- tidy_ticket %>%
  select(dia_semana = dia_semana)

# Crear el gráfico de barras
pregunta_5 <- ggplot(data = compras_por_dia, aes(x = dia_semana)) +
    geom_bar(fill = "green", color = "black") +
    labs(title = "Días con más compras", x = "Días", y = "Frecuencia")

pregunta_5
```




```
ggsave("pregunta5.jpg", plot = pregunta_5, width = 8, height = 6, dpi = 300)
```

Como podemos observar los días con más compras son los lunes, martes y sábado, que eso nos puede demostrar que la gente va de compras a comienzo de semana y los fines de semana.

5) ¿Cuántos productos se compran de media en una compra?

Para saber la media de productos por tickets hemos contabilizado todos los productos y a esta cifra le hemos dividido el número total de tickets.

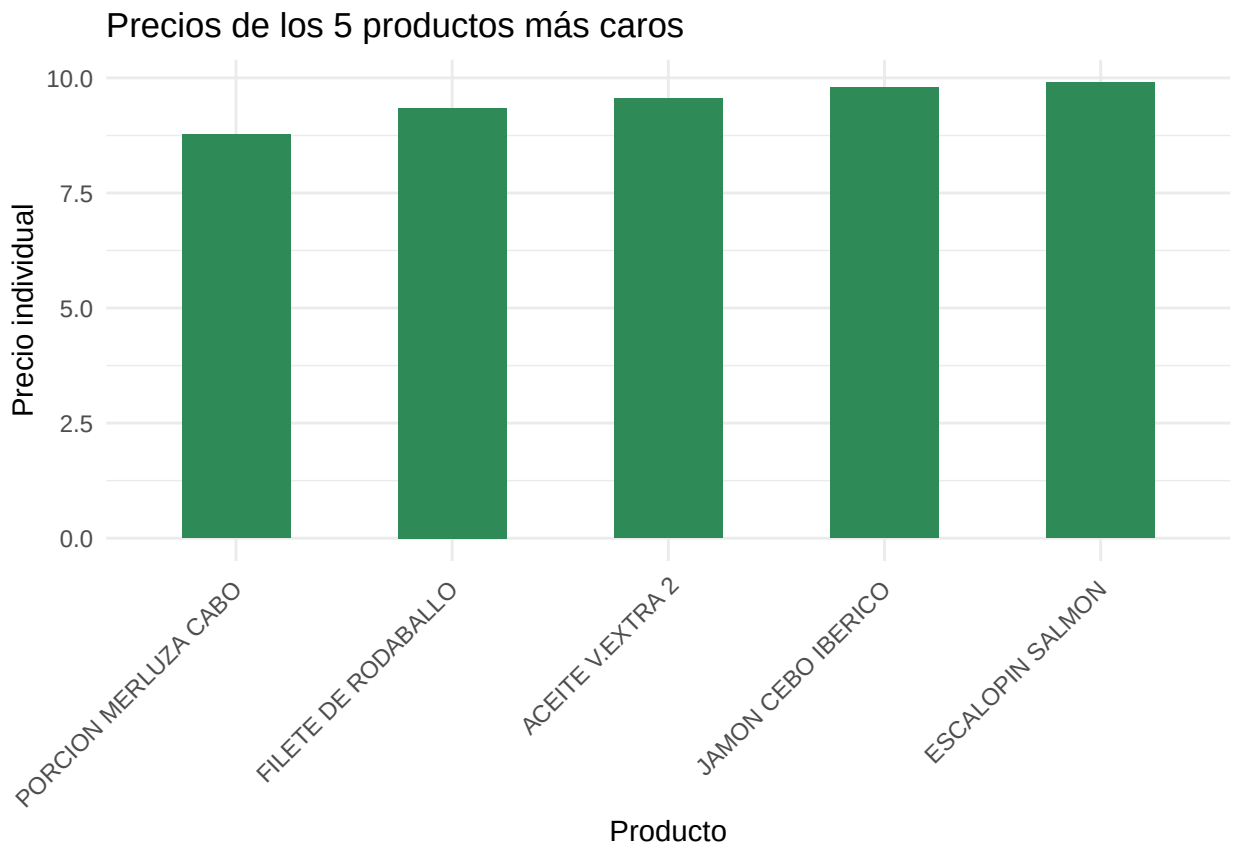
```
# Calculamos de la media de productos por ticket con dos decimales
media_productos_por_ticket = round(sum(nrow(todos_los_tickets)) / num_tickets, 2)
```

Observamos que se obtiene que la media de productos por ticket es de **18.86**. Además sabemos que esto va relacionado con el precio de cada compra. Cuantos más productos compres, más cara saldrá la compra.

6) ¿Qué productos cambian de precio (por kg)?

En este análisis, exploraremos cómo los precios de ciertos productos varían en función de su peso en las compras realizadas en Mercadona. Para ello, hemos desarrollado una función especializada que recorre el contenido de varios tickets almacenados en archivos de texto. Esta función está diseñada para identificar y extraer información sobre los productos vendidos por peso, capturando tanto el nombre del producto como su precio por kilogramo.

```r
extraer_productos_por_peso <- function(rutas_archivos) {
  # Procesamos cada archivo de ticket proporcionado
  for (ruta_archivo in rutas_archivos) {
    # Leemos el contenido del archivo
    lineas <- readLines(ruta_archivo)

    for (i in 2:length(lineas)) {
      # Buscamos las líneas que contienen información de peso y precio por kg
      if (grepl("kg", lineas[i]) && grepl("€/kg", lineas[i])) {
        nombre_producto <- gsub("\\d+", "", lineas[i-1])  # Eliminar números
        # para limpiar el nombre
        nombre_producto <- trimws(nombre_producto)  # Eliminar espacios en
        # blanco sobrantes

        # Extraemos la información de peso y precio por kg
        precio_por_kg_info <- regmatches(lineas[i], regexpr("\\d+,\\d+ €/kg",
                                                            lineas[i]))

        descripcion_producto <- sprintf("%s %s", nombre_producto,
                                        precio_por_kg_info)

        # Mostramos el nombre del producto y su precio por kg
        print(descripcion_producto)
      }
    }
  }
}

grafica <- ggplot(df_precios_plot, aes(x = reorder(producto,
                                                   precio_individual),
                                       y = precio_individual)) +
  geom_bar(stat = "identity", fill = "seagreen", width = 0.5) +
  labs(title = "Precios de los 5 productos más caros",
       x = "Producto",
       y = "Precio individual") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

grafica
```




```
# Si queremos descargarnos el gráfico:

# ggsave("top_5_productos_mas_caros.jpeg", grafica, width = 10, height = 6.5)
```

En resumen, el análisis de los tickets de compra muestra que los precios de productos como plátanos, manzanas y otros vendidos por kilogramo en Mercadona son variables y pueden fluctuar ligeramente. Este comportamiento es común en productos frescos debido a las dinámicas del mercado y las condiciones de suministro, los consumidores deben ser conscientes de la naturaleza variable de los precios de los productos frescos y tomar decisiones informadas para optimizar sus compras.

7) ¿Qué tipo de IVA recauda más dinero?

```
dfIVA <- function(listaArchivos) {
    dfl <- list()   #lista vacía donde se almacenarán los dataframes resultantes

    for (archivo in listaArchivos) {
        texto <- (readLines(archivo, encoding = "windows-1252"))#Extraigo la
        # informacion de los txt
        total1 <- texto[grep("^TOTAL \\(", texto)]
        indice1 <- grep("^IVA BASE", texto)
        indice2 <- grep("^TOTAL \\d", texto)#Obtengo las lineas que contienen el IVA
        IVA <- texto[(indice1 + 1):(indice2 - 1)]
        IVA<- c(IVA,texto[indice2])#Creo un vector con la informacion
```

```
      matrizIVA <- matrix(IVA, ncol = 5, byrow = TRUE)#Lo paso a matriz y le
      # doy la forma adecuada para obtener el df que quiero

      dfIVA <- as.data.frame(matrizIVA)

      df1[[archivo]] <- dfIVA#Almaceno el df en la lista
  }

  df_final<-bind_rows(df1)#Uno todos los df en uno
  df1 <- df_final %>%#Los datos no estan ordenados por columnas,
    # creo nuevas columnas con la condicion de que cada una sea un tipo
    # de IVA o el total
rowwise() %>%
mutate(veintiuno = ifelse(any(startsWith(c_across(everything()), "21")),
                  first(c_across(everything())[startsWith(c_across(everything()),
                                                          "21")]),
                  NA))%>%
  mutate(diez = ifelse(any(startsWith(c_across(everything()), "10")),
                  first(c_across(everything())[startsWith(c_across(everything()),
                                                          "10")]),
                  NA))%>%
mutate(cinco = ifelse(any(startsWith(c_across(everything()), "5")),
                  first(c_across(everything())[startsWith(c_across(everything()),
                                                          "5")]),
                  NA))%>%
mutate(cero = ifelse(any(startsWith(c_across(everything()), "0")),
                  first(c_across(everything())[startsWith(c_across(everything()),
                                                          "0")]),
                  NA))%>%
mutate(total = ifelse(any(startsWith(c_across(everything()), "TOTAL")),
                  first(c_across(everything())[startsWith(c_across(everything()),
                                                          "TOTAL")]),
                  NA))

#Me quedo con las columnas nuevas y selecciono solo los caracteres que
  #me interesan
df <-df1%>%
select(veintiuno,diez,cinco,cero,total)%>%
mutate(base21=substr(veintiuno,5,8))%>%
mutate(base10=substr(diez,5,8))%>%
mutate(base5=substr(cinco,4,8))%>%
mutate(base0=substr(cero,4,8))%>%
mutate(totalBase=substr(total,6,(nchar(total)-4)))%>%
mutate(imponible21=substr(veintiuno,(nchar(veintiuno)-4),
                          nchar(veintiuno)))%>%
mutate(imponible10=substr(diez,(nchar(diez)-4),nchar(diez)))%>%
mutate(imponible5=substr(cinco,(nchar(cinco)-4),nchar(cinco)))%>%
mutate(imponible0=substr(cero,(nchar(cero)-4),nchar(cero)))%>%
mutate(totalImponible=substr(total,(nchar(total)-4),nchar(total)))%>%
    select(base21,base10,base5,base0,imponible21,imponible10,imponible5,
           imponible0,totalBase,totalImponible) %>%
mutate_all(~ str_replace_all(., ",", "."))%>%#Reenplazo comas por puntos
  # y paso a numerico
```

```
   mutate_all(as.numeric)

    return(df)
}
carpeta <- "data"
fichero<-list.files(path = carpeta, full.names = TRUE, recursive = TRUE,
                    pattern = ".txt")

a<-dfIVA(fichero)
```

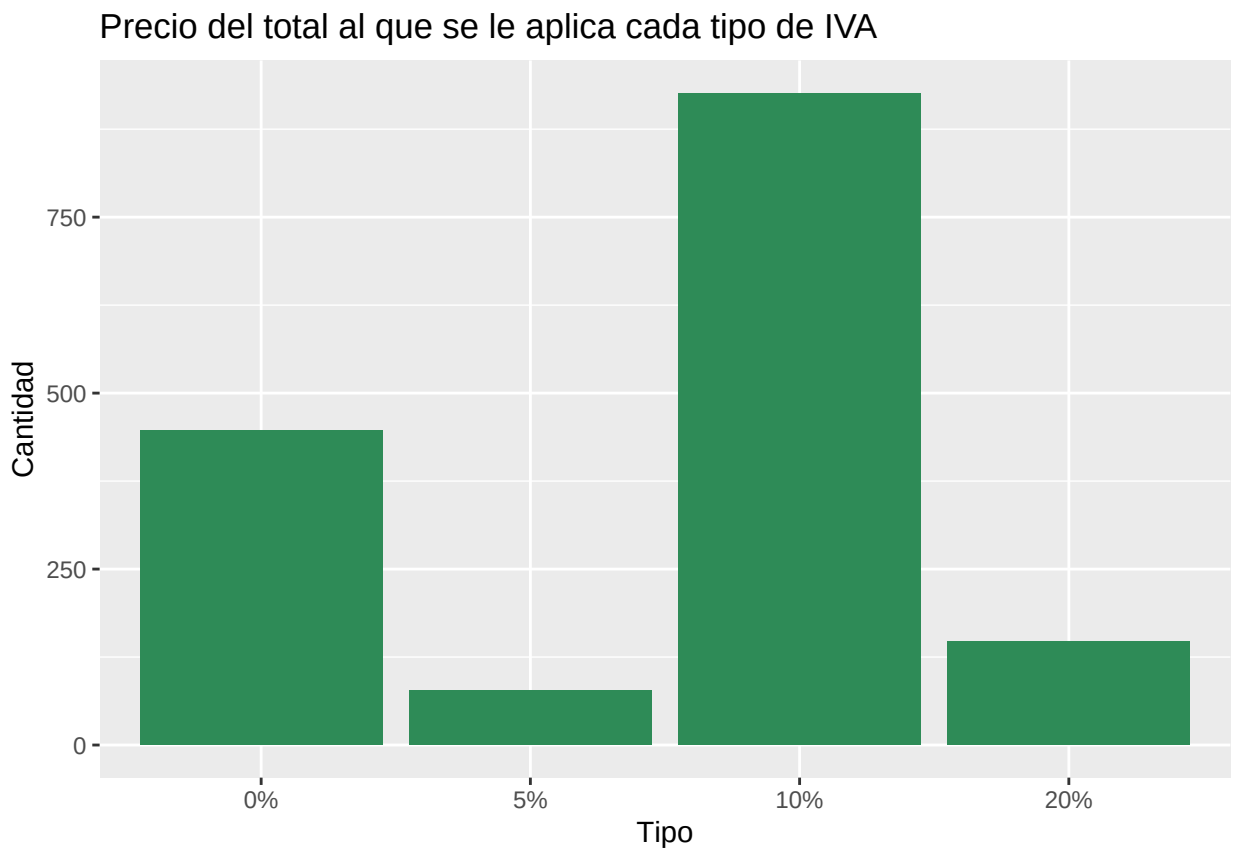
En españa el IVA se divide en tres tipos, general (20%), reducido (10%) y superreducido (5%); también tenemos en cuenta otro tipo, el 0%; es decir, los productos a los que no se les aplica IVA.

Para empezar a estudiar los datos en boxplot, puede ser muy útil ya que sirve para hacerse una idea de los valores y detectar posibles outliers.

En este gráfico se nota que el IVA reducido es el que tiene un rango de valores más amplio, y también se pueden observar unas pocas observaciones atípicas, pero nos centraremos en estudiarlas porque no influyen demasiado en los siguientes gráficos y no son outliers, simplemente compras en las que se ha gastado un poco más de dinero que el promedio.

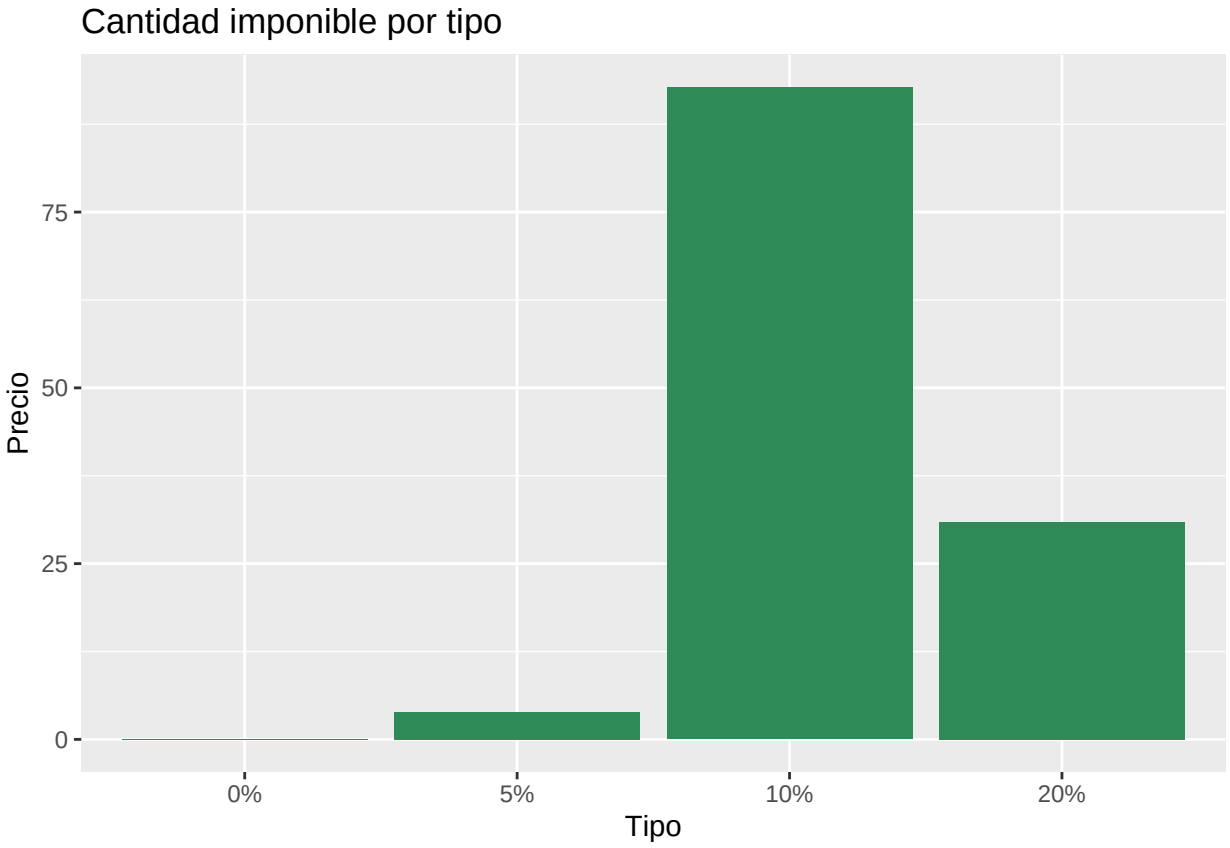
En la siguiente gráfica se puede observar el precio total al que se le aplica cada tipo de IVA, es decir, de los 1598.87 euros, que es la suma del precio de todos los tickets, se puede ver como principalmente se aplica el IVA reducido (10%).

```
## [1] 1598.87
```

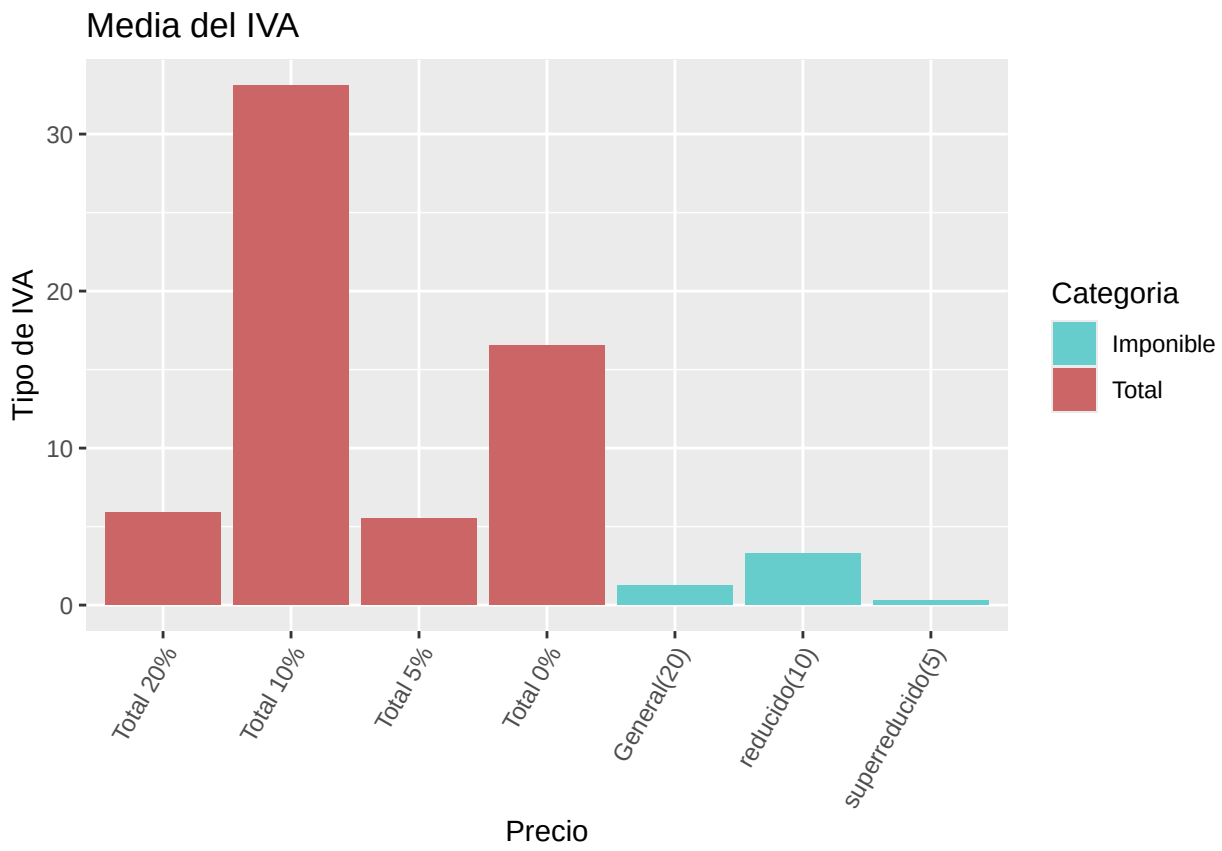


Para continuar, en esta gráfica se puede observar el total imponible de cada tipo de IVA, es decir, de todo el dinero gastado 127.56 euros se han cobrado por el IVA y la mayoría de dinero se ha recaudado por el IVA reducido (10%).

```
## [1] 127.56
```



Para terminar, la siguiente gráfica muestra la media de cada tipo de IVA tanto el total, como la parte imponible.


Media del IVA
Tipo de IVA
30
20
10
0
Total 20%
Total 10%
Total 5%
Total 0%
General(20)
reducido(10)
superreducido(5)
Precio
Categoria
Imponible
Total