Exploración gráfica de datos

Álvaro José Álvarez Arranz y Miguel Ángel Calderón Pazmiño

03/12/2023

Objetivos

El objetivo de este ejercicio consiste en utilizar los métodos exploratorios vistos en teoría para obtener información y conclusiones sobre un conjunto de datos de dominio.

Para ello, se usará u dataset relacionado con el resultado de un *Card Sorting* obtenido a través de una evaluación con distintos usuarios. *Card Sorting* es una técnica de investigación y evaluación utilizada en Interacción Persona-Ordenador y Experiencia de usuario que permite la investigación inicial con usuarios y también la evaluación de estructuras de información (por ejemplo, contenidos de una aplicación web o composiciónde un menú de navegación). Esta técnica consiste en suministrar al usuario unas serie de tarjetas que deben ordenar situándolas destro de un conjunto de categorías (estas pueden ser predefinidas, ser creadas por el usuario, o una combinación de ambas). Cuando la ordenación se realiza varias veces con diferentes usuarios, se obtiene información sobre las relaciones entre tarjetas y categorías, lo que permite obtener información sobre el modelo mental del usuario (en fases iniciales de una investigación), o una indicación sobre la idoneidad de estructuras ya existentes (por ejemplo, la idoneidad de las opciones dentro de las categorías de un menú de una aplicación software).

El dataset a utilizar es el resultado de un Card Sorting donde se les pidió a 24 usuarios clasificar 40 tarjetas que representaban distintos alimentos. Al tratarse de un Card Sorting *abierto*, los usuarios fueron los que crearon las categorías que consideraron convenientes, por lo que se tiene un total de 240 categorías clasificatorias. En concreto, el dataset es un únicp fichero CSV que puede obtenerse en el siguiente enlace:

http://cardsorting.net/tutorials/25.csv

Estructura del fichero

El dataset se dispone en forma de matriz (`M`), de forma que las categorías se situan en filas, constituyendo observaciones (*n=240*), mientras que las tarjetas se distribuyen en columnas, constituyendo variables (*p=40*). Por tanto, `M[i,j]=N`implica que `N` usuarios han clasificado la tarjeta `j` en la categoría `i`.

Para el motivo de este ejercicio, se prescindirá de las columnas *Uniqid*, *Startdate*, *Starttime*, *Endtime*, *QID* y *Comment*.

Tareas

1- **Carga del dataset**. Para ello lo que haremos será usar el link que se indica en el enunciado del ejercicio. Para ello usaremos la función 'read.csv":

```
data = read.csv("http://cardsorting.net/tutorials/25.csv")
```

Una vez cargados, vamos a explorar los datos que hay en CSV del enlace. Para ello vamos a usar las funciones `str(data)` y 'head(data) con el objetivo de ver el típo de variables que se usan y cómo se ven las primeras filas:

```
str(data)
```

```
## 'data.frame':    240 obs. of  47 variables:
##  $ Uniqid      : int  2249 2249 2249 2249 2249 2249 2249 2263 2263 2263 ...
##  $ Category    : chr  "Sides" "meat" "dinners" "Snacks" ...
##  $ Startdate   : chr  "10/8/2014" "10/8/2014" "10/8/2014" "10/8/2014" ...
##  $ Starttime   : chr  "13:09:10" "13:09:10" "13:09:10" "13:09:10" ...
##  $ Endtime     : chr  "13:13:10" "13:13:10" "13:13:10" "13:13:10" ...
##  $ QID         : chr  "A1Q77WXLOKP6T" "A1Q77WXLOKP6T" "A1Q77WXLOKP6T" "A1Q77WXLOKP6T" ...
##  $ Carrots     : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Apple       : int  0 0 0 0 0 1 0 1 0 0 ...
##  $ Banana      : int  0 0 0 0 0 1 0 1 0 0 ...
##  $ Bread       : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Broccoli    : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Butter      : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Cake        : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Cereal      : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Cheese      : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Chicken     : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Coffee      : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Cookies     : int  0 0 0 1 0 0 0 0 0 1 ...
##  $ Corn        : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Doughnuts   : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Egg         : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Hamburger   : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ Lettuce     : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Lobster     : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Milk        : int  0 0 0 0 0 0 1 0 1 0 ...
##  $ Muffin      : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Nuts        : int  0 0 0 1 0 0 0 0 0 1 ...
##  $ Onions      : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Orange      : int  0 0 0 0 0 1 0 1 0 0 ...
##  $ Pancakes    : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Pie         : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Pineapple   : int  0 0 0 0 0 1 0 1 0 0 ...
##  $ Pizza       : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ Popcorn     : int  0 0 0 1 0 0 0 0 0 1 ...
##  $ Potato.chips: int  0 0 0 1 0 0 0 0 0 1 ...
##  $ Potatoes    : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Pretzels    : int  0 0 0 1 0 0 0 0 0 1 ...
##  $ Rice        : int  1 0 0 0 0 0 0 0 0 0 ...
##  $ Salmon      : int  0 1 0 0 0 0 0 0 0 0 ...
```

```
##  $ Soda        : int  0 0 0 0 0 0 1 0 1 0 ...
##  $ Spaghetti   : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ Steak       : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ Waffle      : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ Water       : int  0 0 0 0 0 0 1 0 1 0 ...
##  $ Watermelon  : int  0 0 0 0 0 1 0 1 0 0 ...
##  $ Yogurt      : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ Comment     : logi  NA NA NA NA NA NA ...
```

```
head(data)
```

```
##   Uniqid         Category Startdate Starttime  Endtime           QID Carrots
## 1   2249            Sides 10/8/2014  13:09:10 13:13:10 A1Q77WXLOKP6T       0
## 2   2249             meat 10/8/2014  13:09:10 13:13:10 A1Q77WXLOKP6T       0
## 3   2249          dinners 10/8/2014  13:09:10 13:13:10 A1Q77WXLOKP6T       0
## 4   2249           Snacks 10/8/2014  13:09:10 13:13:10 A1Q77WXLOKP6T       0
## 5   2249       breakfasat 10/8/2014  13:09:10 13:13:10 A1Q77WXLOKP6T       0
## 6   2249 Fruit and veggie 10/8/2014  13:09:10 13:13:10 A1Q77WXLOKP6T       1
##   Apple Banana Bread Broccoli Butter Cake Cereal Cheese Chicken Coffee Cookies
## 1     0      0     1        0      1    0      0      0       0      0       0
## 2     0      0     0        0      0    0      0      0       1      0       0
## 3     0      0     0        0      0    0      0      0       0      0       0
## 4     0      0     0        0      0    1      0      1       0      0       1
## 5     0      0     0        0      0    0      1      0       0      0       0
## 6     1      1     0        1      0    0      0      0       0      0       0
##   Corn Doughnuts Egg Hamburger Lettuce Lobster Milk Muffin Nuts Onions Orange
## 1    0         0   0         0       0       0    0      0    0       1      0
## 2    0         0   0         0       0       1    0      0    0       0      0
## 3    0         0   0         1       0       0    0      0    0       0      0
## 4    0         1   0         0       0       0    0      0    1       0      0
## 5    0         0   1         0       0       0    0      1    0       0      0
## 6    1         0   0         0       1       0    0      0    0       0      1
##   Pancakes Pie Pineapple Pizza Popcorn Potato.chips Potatoes Pretzels Rice
## 1        0   0         0     0       0            0        1        0    1
## 2        0   0         0     0       0            0        0        0    0
## 3        0   0         0     1       0            0        0        0    0
## 4        0   1         0     0       1            1        0        1    0
## 5        1   0         0     0       0            0        0        0    0
## 6        0   0         1     0       0            0        0        0    0
##   Salmon Soda Spaghetti Steak Waffle Water Watermelon Yogurt Comment
## 1      0    0         0     0      0     0          0      0      NA
## 2      1    0         0     1      0     0          0      0      NA
## 3      0    0         1     0      0     0          0      0      NA
## 4      0    0         0     0      0     0          0      1      NA
## 5      0    0         0     0      1     0          0      0      NA
## 6      0    0         0     0      0     0          1      0      NA
```

Ahora que tenemos una idea de cómo se estructuran los datos, vamos a pasar a limpiarlos para poder trabajar con ellos.

2- **Transformaciones**.

Dado que podemos ver que están las columnas *Uniqid*, *Startdate*, *Starttime*, *Endtime*, *QID* y *Comment*, vamos a eliminar dichas columnas del dataset:

```r
#install.packages('dplyr')
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.2
```

```
##
## 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
```

```r
data2 = select(data, -Uniqid, -Startdate, -Starttime, -Endtime, -QID, -Comment)
head(data2)
```

```
##           Category Carrots Apple Banana Bread Broccoli Butter Cake Cereal
## 1            Sides       0     0      0     1        0      1    0      0
## 2             meat       0     0      0     0        0      0    0      0
## 3          dinners       0     0      0     0        0      0    0      0
## 4           Snacks       0     0      0     0        0      0    1      0
## 5       breakfasat       0     0      0     0        0      0    0      1
## 6 Fruit and veggie       1     1      1     0        1      0    0      0
##   Cheese Chicken Coffee Cookies Corn Doughnuts Egg Hamburger Lettuce Lobster
## 1      0       0      0       0    0         0   0         0       0       0
## 2      0       1      0       0    0         0   0         0       0       1
## 3      0       0      0       0    0         0   0         1       0       0
## 4      1       0      0       1    0         1   0         0       0       0
## 5      0       0      0       0    0         0   1         0       0       0
## 6      0       0      0       0    1         0   0         0       1       0
##   Milk Muffin Nuts Onions Orange Pancakes Pie Pineapple Pizza Popcorn
## 1    0      0    0      1      0        0   0         0     0       0
## 2    0      0    0      0      0        0   0         0     0       0
## 3    0      0    0      0      0        0   0         0     1       0
## 4    0      0    1      0      0        0   1         0     0       1
## 5    0      1    0      0      0        1   0         0     0       0
## 6    0      0    0      0      1        0   0         1     0       0
##   Potato.chips Potatoes Pretzels Rice Salmon Soda Spaghetti Steak Waffle Water
## 1            0        1        0    1      0    0         0     0      0     0
## 2            0        0        0    0      1    0         0     1      0     0
## 3            0        0        0    0      0    0         1     0      0     0
## 4            1        0        1    0      0    0         0     0      0     0
## 5            0        0        0    0      0    0         0     0      1     0
## 6            0        0        0    0      0    0         0     0      0     0
##   Watermelon Yogurt
## 1          0      0
## 2          0      0
## 3          0      0
## 4          0      1
```

```
## 5          0      0
## 6          1      0
```

Ahora que tenemos las columnas que no nos interesan eliminadas, lo primero que vamos a hacer es comprobar si existe algún valor faltante en el dataset, y cuantos son. Para ello usaremos la función `sum()` y `is.na()`:

```
sum(is.na(data2))
```

```
## [1] 0
```

Como podemos observar, no hay valores *NA* en el connjunto de datos. Ahora vamos a ver las categorías.

```
sort(data2$Category)
```

```
##   [1] "1"                    "1"                    "1"
##   [4] "1"                    "1"                    "1"
##   [7] "1"                    "1"                    "1"
##  [10] "1"                    "10"                   "10"
##  [13] "11"                   "11"                   "12"
##  [16] "13"                   "14"                   "2"
##  [19] "2"                    "2"                    "2"
##  [22] "2"                    "2"                    "2"
##  [25] "2"                    "2"                    "3"
##  [28] "3"                    "3"                    "3"
##  [31] "3"                    "3"                    "3"
##  [34] "3"                    "4"                    "4"
##  [37] "4"                    "4"                    "4"
##  [40] "4"                    "4"                    "4"
##  [43] "5"                    "5"                    "5"
##  [46] "5"                    "5"                    "5"
##  [49] "5"                    "5"                    "6"
##  [52] "6"                    "6"                    "6"
##  [55] "6"                    "6"                    "6"
##  [58] "6"                    "7"                    "7"
##  [61] "7"                    "7"                    "7"
##  [64] "7"                    "7"                    "8"
##  [67] "8"                    "8"                    "8"
##  [70] "8"                    "9"                    "9"
##  [73] "9"                    "9"                    "9"
##  [76] "Baked Goods/Breakfast" "bakery"              "Beverage"
##  [79] "beverages"            "beverages"            "Beverages"
##  [82] "Beverages"            "Beverages"            "Beverages"
##  [85] "Bread"                "Bread"                "Breads"
##  [88] "Breafast Food"        "breakfasat"           "breakfast"
##  [91] "breakfast"            "Breakfast"            "Breakfast"
##  [94] "Breakfast"            "Breakfast"            "breakfast foods"
##  [97] "Breakfast Foods"      "Breakfast items"      "Breakfast Items"
## [100] "Butter!"              "Carbohydrates"        "Chicken"
## [103] "Crunchy produce"      "dairy"                "dairy"
## [106] "Dairy"                "Dairy"                "Dairy"
## [109] "Dairy"                "Dairy"                "Dairy"
## [112] "Dairy"                "Dairy Products"       "Deserts"
```

```
## [115] "dessert"               "Dessert"               "Desserts"
## [118] "dinner"                "dinners"               "Dinners"
## [121] "drinks"                "Drinks"                "Drinks"
## [124] "Drinks"                "Drinks"                "Drinks"
## [127] "Drinks"                "Drinks"                "Drinks"
## [130] "Fats"                  "Fats/Junk Food"        "Fish"
## [133] "Fruit"                 "Fruit"                 "Fruit"
## [136] "Fruit"                 "Fruit"                 "Fruit and veggie"
## [139] "fruits"                "Fruits"                "Fruits"
## [142] "Fruits"                "Fruits & Veggies"      "Fruits & Veggies"
## [145] "Fruits & Veggies"      "Fruits & Veggies"      "Fruits & Veggies"
## [148] "Fruits & Veggies"      "Fruits & Veggies"      "Fruits & Veggies"
## [151] "Fruits & Veggies"      "Fruits & Veggies"      "Fruits & Veggies"
## [154] "fruits and vegetables" "frutis and vegies"     "Grain"
## [157] "Grains"                "Grains"                "Grains"
## [160] "Grains"                "Grains"                "Grains"
## [163] "Junk"                  "lunch"                 "Main Course"
## [166] "Main dishes"           "Meals & Fast Food"     "meat"
## [169] "Meat"                  "Meat"                  "Meat"
## [172] "Meat"                  "meat and seafood"      "meats"
## [175] "Meats"                 "Meats"                 "Misc. healthy"
## [178] "Nothing"               "Nuts"                  "pantry food"
## [181] "Pasta"                 "Pasta"                 "pizza"
## [184] "produce"               "Produce"               "Protein"
## [187] "Protein"               "Protein"               "Protein"
## [190] "Protein"               "Protein"               "Protein"
## [193] "Protein"               "Protein"               "Protein"
## [196] "Protein meal bases"    "protein non meat"      "Quick energy"
## [199] "seafood"               "Seafood"               "Seafood"
## [202] "Seafood"               "Side Dishes"           "Sides"
## [205] "Snack Foods"           "snacks"                "snacks"
## [208] "snacks"                "snacks"                "Snacks"
## [211] "Snacks"                "Snacks"                "Snacks"
## [214] "Snacks"                "Snacks"                "Snacks"
## [217] "Snacks"                "Snacks"                "Snacks"
## [220] "Snacks"                "starches"              "Starchs/Grains"
## [223] "Substantial snacks"    "Sweets"                "Toppings"
## [226] "Treats"                "Treats"                "Treats"
## [229] "Treats"                "Unhealthy liquids"     "vegetables"
## [232] "Vegetables"            "Vegetables"            "Vegetables"
## [235] "Vegetables"            "Vegetables"            "Vegetables"
## [238] "Vegetables"            "Veggies"               "Whole Meal"
```

Dado que las categorías son varias y creadas por los usuarios, al ser una prueba abierta, vamos a asegurarnos de que estén bien y en minúsculas:

```
data2$Category = trimws(tolower(data2$Category))

sort(data2$Category)
```

```
##   [1] "1"                   "1"                   "1"
##   [4] "1"                   "1"                   "1"
##   [7] "1"                   "1"                   "1"
```

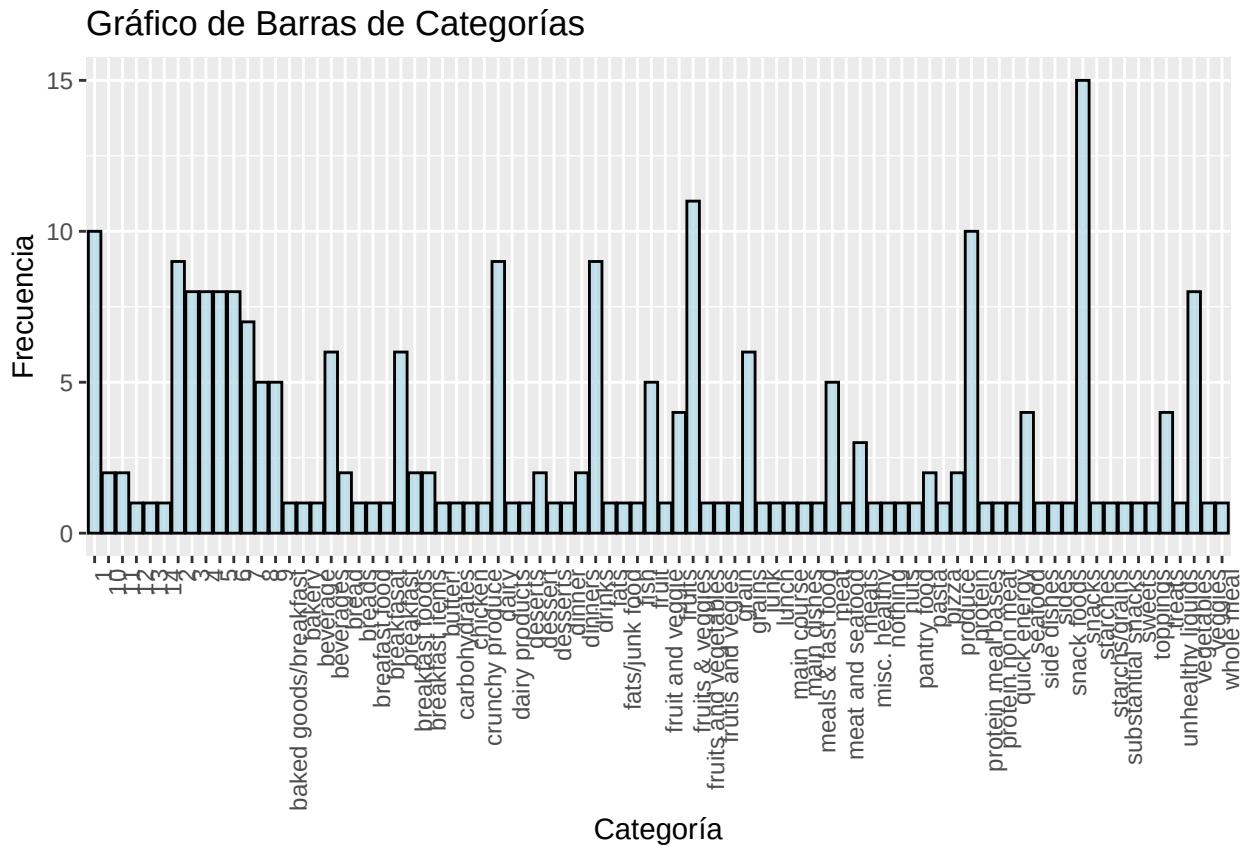
```
##  [10] "1"                    "10"                   "10"
##  [13] "11"                   "11"                   "12"
##  [16] "13"                   "14"                   "2"
##  [19] "2"                    "2"                    "2"
##  [22] "2"                    "2"                    "2"
##  [25] "2"                    "2"                    "3"
##  [28] "3"                    "3"                    "3"
##  [31] "3"                    "3"                    "3"
##  [34] "3"                    "4"                    "4"
##  [37] "4"                    "4"                    "4"
##  [40] "4"                    "4"                    "4"
##  [43] "5"                    "5"                    "5"
##  [46] "5"                    "5"                    "5"
##  [49] "5"                    "5"                    "6"
##  [52] "6"                    "6"                    "6"
##  [55] "6"                    "6"                    "6"
##  [58] "6"                    "7"                    "7"
##  [61] "7"                    "7"                    "7"
##  [64] "7"                    "7"                    "8"
##  [67] "8"                    "8"                    "8"
##  [70] "8"                    "9"                    "9"
##  [73] "9"                    "9"                    "9"
##  [76] "baked goods/breakfast" "bakery"              "beverage"
##  [79] "beverages"            "beverages"            "beverages"
##  [82] "beverages"            "beverages"            "beverages"
##  [85] "bread"                "bread"                "breads"
##  [88] "breafast food"        "breakfasat"           "breakfast"
##  [91] "breakfast"            "breakfast"            "breakfast"
##  [94] "breakfast"            "breakfast"            "breakfast foods"
##  [97] "breakfast foods"      "breakfast items"      "breakfast items"
## [100] "butter!"              "carbohydrates"        "chicken"
## [103] "crunchy produce"      "dairy"                "dairy"
## [106] "dairy"                "dairy"                "dairy"
## [109] "dairy"                "dairy"                "dairy"
## [112] "dairy"                "dairy products"       "deserts"
## [115] "dessert"              "dessert"              "desserts"
## [118] "dinner"               "dinners"              "dinners"
## [121] "drinks"               "drinks"               "drinks"
## [124] "drinks"               "drinks"               "drinks"
## [127] "drinks"               "drinks"               "drinks"
## [130] "fats"                 "fats/junk food"       "fish"
## [133] "fruit"                "fruit"                "fruit"
## [136] "fruit"                "fruit"                "fruit and veggie"
## [139] "fruits"               "fruits"               "fruits"
## [142] "fruits"               "fruits & veggies"     "fruits & veggies"
## [145] "fruits & veggies"     "fruits & veggies"     "fruits & veggies"
## [148] "fruits & veggies"     "fruits & veggies"     "fruits & veggies"
## [151] "fruits & veggies"     "fruits & veggies"     "fruits & veggies"
## [154] "fruits and vegetables" "frutis and vegies"   "grain"
## [157] "grains"               "grains"               "grains"
## [160] "grains"               "grains"               "grains"
## [163] "junk"                 "lunch"                "main course"
## [166] "main dishes"          "meals & fast food"    "meat"
## [169] "meat"                 "meat"                 "meat"
```

```
## [172] "meat"                  "meat and seafood"      "meats"
## [175] "meats"                 "meats"                 "misc. healthy"
## [178] "nothing"               "nuts"                  "pantry food"
## [181] "pasta"                 "pasta"                 "pizza"
## [184] "produce"               "produce"               "protein"
## [187] "protein"               "protein"               "protein"
## [190] "protein"               "protein"               "protein"
## [193] "protein"               "protein"               "protein"
## [196] "protein meal bases"    "protein non meat"      "quick energy"
## [199] "seafood"               "seafood"               "seafood"
## [202] "seafood"               "side dishes"           "sides"
## [205] "snack foods"           "snacks"                "snacks"
## [208] "snacks"                "snacks"                "snacks"
## [211] "snacks"                "snacks"                "snacks"
## [214] "snacks"                "snacks"                "snacks"
## [217] "snacks"                "snacks"                "snacks"
## [220] "snacks"                "starches"              "starchs/grains"
## [223] "substantial snacks"    "sweets"                "toppings"
## [226] "treats"                "treats"                "treats"
## [229] "treats"                "unhealthy liquids"     "vegetables"
## [232] "vegetables"            "vegetables"            "vegetables"
## [235] "vegetables"            "vegetables"            "vegetables"
## [238] "vegetables"            "veggies"               "whole meal"
```

3- **Creación de un histograma**: Ahora que los datos ya están correctos podemos crear un histograma para ver cómo se reparten los datos. Dado que los datos a representar son variables discretas, lo que haremos será usar la función `geom_bar()`:

```r
library(ggplot2)

ggplot(data2, aes(x = Category)) +
  geom_bar(fill = "lightblue", color = "black", alpha = 0.7) +
  labs(title = "Gráfico de Barras de Categorías", x = "Categoría", y = "Frecuencia") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



4- **Matriz de distancia**: Ahora vamos a comparar las distancias entre las tarjetas. Dado que hay una cantidad considerable de tarjetas, la mejor forma de hacer esta representación es la de usar una matriz con mapa de calor para encontrar las tarjetas más distantes, y para ello lo vamos a hacer de la siguiente manera:

```
# Bibliotecas necesarias
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
## v readr     2.1.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
var_names = names(data2)[-1]

data_distancias = data2[-1]

matriz_distancias <- dist(t(data_distancias))

# Convertir la matriz de distancias a un dataframe
df_distancias <- as.data.frame(as.matrix(matriz_distancias))
```

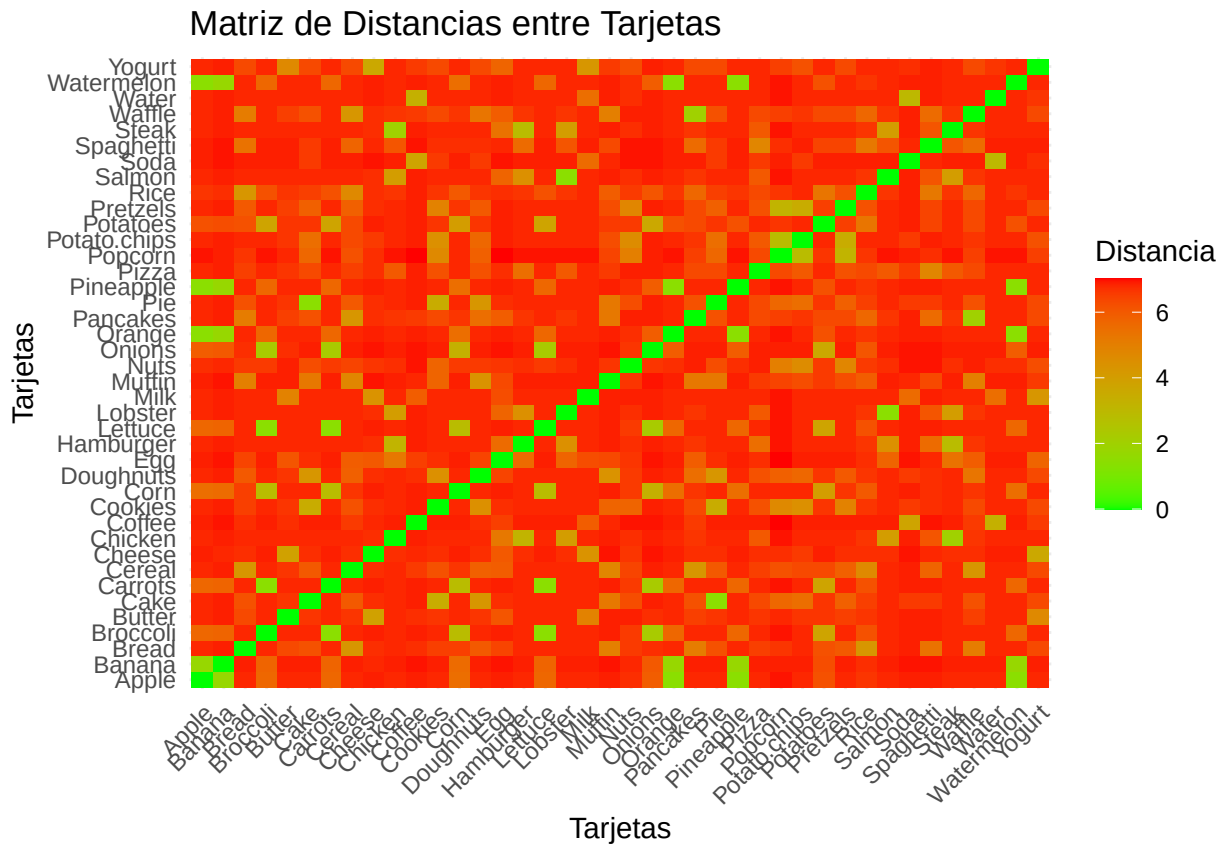
```r
# Asignar nombres a las filas y columnas del dataframe
rownames(df_distancias) <- var_names
colnames(df_distancias) <- var_names

# Convertir el dataframe de distancias a formato largo y añadir nombres de filas y columnas como variab
df_distancias_largo <- df_distancias %>%
  rownames_to_column(var = "Variable1") %>%
  gather(key = "Variable2", value = "Distancia", -Variable1)

mapa_calor <- ggplot(df_distancias_largo, aes(Variable1, Variable2, fill = Distancia)) +
  geom_tile() +
  scale_fill_gradient(low = "green", high = "red") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Matriz de Distancias entre Tarjetas",
       x = "Tarjetas",
       y = "Tarjetas")

# Añadir una leyenda en la parte derecha y de orientación vertical
mapa_calor <- mapa_calor +
  theme(legend.position = "right",
        legend.direction = "vertical")

mapa_calor
```



Puede observarse que pese a la gran cantidad de categorías, puede verse las cercanías entre tarjetas mejor que las lejanías.

5- **Representación gráfica**: Ahora vamos a representar las relaciones de las tarjetas con un grafo. Para ello se usará `qgraph`.

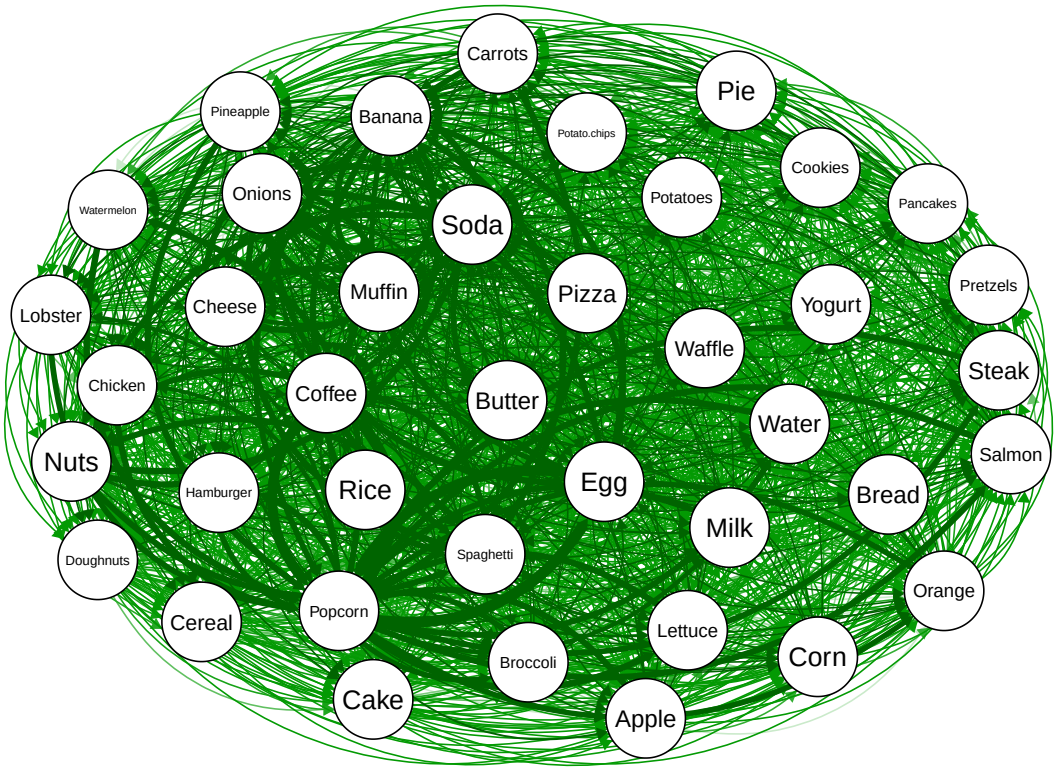
```
# install.packages("qgraph")
library(qgraph)
```

```
## Warning: package 'qgraph' was built under R version 4.3.2
```

```
qgraph(input=df_distancias_largo,
  var.names = var_names,
  layout = "spring",  # Puedes ajustar el tipo de layout según tus preferencias
  esize = df_distancias_largo$Distancia,  # Especificar el tamaño de los bordes según la distancia
  vsize = 7,  # Ajustar el tamaño de los nodos
  legend.cex = 0.9  # Ajustar el tamaño de la leyenda
)
```

```
## Warning in qgraph(input = df_distancias_largo, var.names = var_names, layout =
## "spring", : The following arguments are not documented and likely not arguments
## of qgraph and thus ignored: var.names
```

```
## Warning in avgW * (esize - 1): longitud de objeto mayor no es múltiplo de la
## longitud de uno menor
```



Podemos observar que debido a la gran cantidad de tarjetas, es muy dificil ver cómo se relacionan, pero sí que podemos ver relaciones más fuertes que otras como “Huevo” y “Pizza” por estar estrechamente relacionados en algunos platos.

6- **Conclusiones**: Se han visto algunas formas en las que los datos pueden ser visualizados como las *barras*, matrices configuradas como mapas de calor o gráficos de relaciones. Debido a la gran cantidad de categorías, que al ser una prueba abierta impica que los participantes podían crear las suyas propias, y a la gran cantidad de tarjetas la visualización de los datos de relaciones se vuelve dificil, necesitando una posible normalización de categorías y/o reducción de tarjetas para su clasificación.