

Segunda Tarea P2P

Aurelio Gallardo Rodríguez

2 de Noviembre de 2016

Contents

Introducción.	2
Respecto de este trabajo	2
Condiciones del ejercicio.	2
Apartado a.- Generación de Data Frames	2
Apartado a.1.- Carga del fichero FoodFactsMooc.csv y DF_G (global)	2
Apartado a.2.- Data frames por continentes	2
Apartado a.3.- Crear un data frame global SIN REPETICIONES	3
Apartado a.4.- Crear un data frame SIN REPETICIONES por continente	3
Apartado a.5.- Concatenar los <i>DFU_</i> de cada continente	5
Resumen y conclusión del apartado a	6
Apartado b.- Porcentajes de valores NA en DFG_U	6
Apartado c.- Calculad las correlaciones entre las variables numéricas del dataframe DFU_G	7
Apartado c.1.- Matriz de correlación	7
Apartado c.2.- Heatmap	9
Apartado c.3.- Diagrama de dispersión para las dos variables con mayor correlación: energía y grasa	10
Apartado d.- Vitamina B6 en relación al Actimel de Danone (0.21mg/100g)	11
Apartado d.1.- Data frame con Vitamina B6 disponible. Localizar el producto problema.	11
Apartado d.2.- Tablas de frecuencias. Estadísticos.	12
Apartado d.3.- Los 5 productos con mayor cantidad de vitamina B6	13
Apartado e.- Boxplot Azúcar, Grasa, Sal y N° Aditivos	14
Apartado e.1.- Azúcar	14
Apartado e.2.- Grasas	15
Apartado e.3.- Sal	17
Apartado e.3.- N° de aditivos	18
Apartado f.- Histogramas de bebidas alcohólicas	19
Apartado f.1.- Histograma de DFU_G (datos globales sin repetir)	19
Apartado f.2.- Histogramas por continentes (datos sin repetir DFU_Continentes)	20
Apartado g.- Diagrama de barras del grado de nutrición francés	22
Apartado h.- Diagramas de caja por grado de nutrición	24

Introducción.

Respecto de este trabajo

He intentado, en todo momento, no sólo contestar las preguntas del ejercicio sino también ilustrar el proceso, tanto en su desarrollo lógico, como matemático e instrumental (usando R Markdown y RStudio). Se muestran casi todos los códigos precisamente para mostrar el proceso en RSTUDIO, lo cual no haríamos si el documento fuese más formal.

Condiciones del ejercicio.

El ejercicio consiste en analizar la tabla FoodFactsMooc.csv, la cual es un extracto para el curso de la base de datos nutricional *FoodFacts.csv*, de la web <http://world.openfoodfacts.org/data>. En ella se recogen datos de varios productos, clasificados por países, continentes y valores nutricionales.

Apartado a.- Generación de Data Frames

Apartado a.1.- Carga del fichero FoodFactsMooc.csv y DF_G (global)

Simplemente cargamos el fichero FoodFactsMooc.csv del directorio de trabajo con la orden **read.csv**, usando el parámetro *header=TRUE*. Posteriormente la asignamos al data frame **DF_G** y borramos *FoodFactsMooc*

```
FoodFactsMooc= read.csv("FoodFactsMooc.csv",header = TRUE)
DF_G=data.frame(FoodFactsMooc)
rm(FoodFactsMooc)
```

Apartado a.2.- Data frames por continentes

De este data frame **DF_G**, extraeremos 6 data frames, uno por continente (considerando el americano dividido en Norte y Sur). Para ello, lo primero es obtener dos vectores con los nombres de los continentes; el primero (**continentes**), tal como vienen en los datos (previa conversión a factor)...

```
DF_G$continent=as.factor(DF_G$continent)
continentes=levels(DF_G$continent) # extrae los distintos continentes
```

Y el segundo, **continentes2**, tal como tiene que quedar en la definición de los data frames... (DF_Europa, DF_AmericaS...); eso sí en el mismo orden y correspondencia que el anterior:

```
continentes2=c("Africa","Asia","Europa","AmericaN","Oceania","AmericaS")
```

El siguiente script hace la asignación:

```
# por cada valor del vector continentes
for (i in continentes){ # bucles for
  DFextraer = DF_G[DF_G$continent==i,] # asigna datos filtrados
  ct=trimws(continentes2[which(continentes==i)]) #extrae el nombre correcto
  assign(paste0("DF_",ct),DFextraer) # asigna el nuevo data frame
  rm(DFextraer) #borra el data frame que no necesito
} # Fin del bucle
head(DF_Europa[,1:4],3)
```

```
##           product_name      country continent nutrition_grade_fr
## 10 Luxury Christmas Pudding United Kingdom      Europe          c
## 11 Luxury Christmas Pudding United Kingdom      Europe          c
## 13   Pepsi, Nouveau goût !      France      Europe          e
```

```
head(DF_AmericaS[,1:3],3)
```

```
##           product_name      country      continent
## 3342 Great Value Regular Potato Crisps      Brazil South America
## 7439           Boulgour French Guiana South America
## 9046           Glace rhum-raisins French Guiana South America
```

Por cada elemento del vector *continente* (bucle for), asigno los datos filtrando los que tiene ese nombre en la columna de **DF_G\$continent**; en la variable **ct** extrae el nombre correspondiente al segundo vector *continentes2* en la posición *i*; con **assign** el data frame *DFExtraer* lo copio a otro cuyo nombre construyo usando el prefijo “DF_” y el valor de **ct**, y, por fin, en cada paso del bucle borro el data frame comodín *DFExtraer*. Como ejemplo, veo las primeras filas y columnas del data frame **DF_Europa** y del **DF_AmericaS**

Apartado a.3.- Crear un data frame global SIN REPETICIONES

En **DF_G** hay muchos productos repetidos; tan sólo diferirán en el país y /o continente donde se vendan. Vamos a construir otro data frame en el que no haya productos repetidos (y no aparezca ni el país ni el continente).

```
DFU_G=DF_G # construyo uno único. Hago una copia del Global
DFU_G$continent=NULL # elimino continentes
DFU_G$country=NULL #elimino los países
DFU_G=unique(DFU_G)
cat("El número de repeticiones fueron: ",dim(DF_G)[1]-dim(DFU_G)[1])
```

```
## El número de repeticiones fueron: 1529
```

En primer lugar, hago la copia de **DF_G** a **DFU_G**, para, posteriormente eliminar los vectores correspondientes a países y continentes. Después uso la función *unique* para eliminar los repetidos. Resto las dimensiones de las filas de ambos data frames para obtener el número de repeticiones que hay en **DF_G** (como vemos son **1529** repeticiones)

Apartado a.4.- Crear un data frame SIN REPETICIONES por continente

El script que consigue crear data frames, por continentes, sin repeticiones de productos *dentro de cada continente* se presenta a continuación.

Este script es largo, y hace el cálculo, además, de cuántos productos hay repetidos **en cada continente**. Lo más destacado, a lo que se refería el ejercicio, está consignado por letras (ver comentarios **dentro del script**).

```
# =====
# Continentes SIN REPETICIONES
# =====
filasRepPorContinente=data.frame(Continente=character(0),Repetidos=numeric(0))
filasRepPorContinente$Continente=as.character(filasRepPorContinente$Continente)
for (i in continentes2){ # bucles for [A]
  DFExtraer = get(paste0("DF_",i)) # asigno por nombre [B]
  DFExtraer$country = NULL # le quito la variable país [C]
  DFExtraer=unique(DFExtraer) # quito filas repetidas [D]
  # Filas repetidas por continente
  filasRep=dim(get(paste0("DF_",i)))[1]-dim(DFExtraer)[1]
  filasRepPorContinente[which(continentes2==i),]=c(i,filasRep)
  # cat(paste0("Filas repetidas para ",i),filasRep,"\n")
  assign(paste0("DFU_",i),DFExtraer) # [E]
  rm(DFExtraer) # borro este data frame [F]
  rm(filasRep)
}
filasRepPorContinente$Repetidos=as.numeric(filasRepPorContinente$Repetidos)
knitr::kable(filasRepPorContinente, digits = 2, # dos decimales
caption = 'Filas Repetidas por continente.',
align=c("l","c")) # alineación izq, izq, centrado
```

Table 1: Filas Repetidas por continente.

Continente	Repetidos
Africa	1
Asia	0
Europa	1325
AmericaN	32
Oceania	8
AmericaS	1

```
cat("En total tenemos unas",sum(filasRepPorContinente$Repetidos)," repeticiones por continente")
```

```
## En total tenemos unas 1367 repeticiones por continente
```

```
rm(filasRepPorContinente)
```

- [A] por cada valor del bucle en **continentes2**.
- [B] al data frame DFExtraer le asigno con get el DF correspondiente al nombre en el vector continentes; por ejemplo, si i="Europa", DFExtraer=DF__Europa. La función get accede a "DF__Europa" a través de su nombre (paste0 concatena dos cadenas).
- [C] elimina el vector del data frame que corresponde al país.
- [D] elimina los elementos repetidos.
- [E] Crea un data frame con el nombre **DFU__** y el valor del vector **continentes2**.
- [F] Borro el data frame comodín DFExtraer.

El resto del script permite conocer las diferentes repeticiones que hemos eliminado *por continente*, asignando esta información a un data frame llamado **filasRepPorContinente** Por último, muestro un ejemplo de algunos data frames **DFU__**

```
head(DFU_Asia[,1:3],3)
```

```
##               product_name continent nutrition_grade_fr
## 2045    Beatrix Potter Butter Cookies      Asia
## 2125 Lee Kum Kee Premium Gold Soy Sauce      Asia      b
## 3273      ALMOND ROCA buttercrunch      Asia      e
```

```
head(DFU_Africa[,1:3],3)
```

```
##               product_name continent nutrition_grade_fr
## 1222    Solid Pack Pumpkin      Africa      a
## 3625      Foul Medammas      Africa      a
## 3628 Tahina Natural 100%      Africa      d
```

Apartado a.5.- Concatenar los *DFU__* de cada continente

En este último apartado vamos a concatenar todos los data frames **DFU__**. El resultado se almacenará en un data frame **DFU__Continentes** que contendrá, quizás, líneas repetidas, pero NO DENTRO DE UN MISMO CONTINENTE.

```
# =====
# Data frame Global SIN REPETICIONES dentro del mismo continente
# =====
DFU_Continentes =data.frame()
for (i in continentes2){ # bucles for
  DFExtraer = get(paste0("DFU_",i)) # asigno por nombre
  DFU_Continentes=data.frame(rbind(DFU_Continentes,DFExtraer))
  rm(DFExtraer)
}
rm(i)
```

Tras crearlo vacío, en un bucle, voy extrayendo los nombres del vector **continentes2**. Asigno al data frame comodín DFExtraer el data frame *DFU__ por nombre* (por ejemplo, DFExtraer=DFU_Europa) con la función *get*, y voy asignándoselo de forma recursiva con *rbind* a **DFU__Continentes**. Al final de cada bucle, borro el data frame comodín DFExtraer. Y al terminar, borro la variable *i*.

Por último, extraigo tres datos de ejemplo de **DFU__Continentes**.

```
head(DFU_Continentes[,1:4],3)
```

```
##               product_name continent nutrition_grade_fr additives_n
## 1222    Solid Pack Pumpkin      Africa      a            0
## 3625      Foul Medammas      Africa      a            1
## 3628 Tahina Natural 100%      Africa      d            0
```

Resumen y conclusión del apartado a

Listado de objetos data frame (“**DF**”) creados

```
l1=ls(pattern = "DF") # los que contienen DF
cat("Hemos creado un total de",length(l1),"objetos")
```

```
## Hemos creado un total de 15 objetos
```

```
l1
```

```
## [1] "DF_Africa"      "DF_AmericaN"    "DF_AmericaS"
## [4] "DF_Asia"        "DF_Europa"      "DF_G"
## [7] "DF_Oceania"     "DFU_Africa"     "DFU_AmericaN"
## [10] "DFU_AmericaS"   "DFU_Asia"       "DFU_Continentes"
## [13] "DFU_Europa"     "DFU_G"          "DFU_Oceania"
```

Borrado de objetos que no necesité

```
rm(list=ls()[!(ls() %in% ls(pattern = "DF"))])
```

Apartado b.- Porcentajes de valores NA en DFG_U

Ejecuto el siguiente script:

```
fNonAv=function(x){sum(is.na(x))}
fAv=function(x){sum(!is.na(x))}

a1=sapply(DFU_G[,5:12], FUN=fNonAv)
a2=sapply(DFU_G[,5:12], FUN=fAv)
n=dim(DFU_G[,5:12])[1]

DFU_G_NA=data.frame(Valores.NA=a1,Valores.Válidos=a2)
DFU_G_NA$Por.NA=paste0(round(100*DFU_G_NA$Valores.NA/n,2),"%")
DFU_G_NA$Por.Válidos=paste0(round(100*DFU_G_NA$Valores.Válidos/n,2),"%")
```

Primer defino dos funciones, *fNonAv* y *fAv* que me sumarán respectivamente, columna a columna, si aparece en la misma NA o no. Se lo aplico al data frame **DFU_G**, entre las columnas 5 y 12 (los valores numéricos), y, con los resultados, construyo el data frame **DFU_G_NA**, añadiendo cálculo de porcentajes.

Los valores de porcentaje añadidos, en la columna *Por.NA* y *Por.Válidos* se muestran como caracteres (no numéricos) al concatenarles el símbolo “%”.

Table 2: Porcentajes por parámetros de valores NA y válidos

	Valores.NA	Valores.Válidos	Por.NA	Por.Válidos
energy	2119	35296	5.66%	94.34%
fat	2128	35287	5.69%	94.31%
sugars	5746	31669	15.36%	84.64%

	Valores.NA	Valores.Válidos	Por.NA	Por.Válidos
fiber	15487	21928	41.39%	58.61%
proteins	2546	34869	6.8%	93.2%
sodium	5493	31922	14.68%	85.32%
alcohol	35178	2237	94.02%	5.98%
vitamin_b6	36736	679	98.19%	1.81%

El mayor porcentaje de valores NA lo presentan los parámetros *alcohol* y *vitamina B6*. La explicación es que, en general los alimentos comparten una serie de características como la energía en calorías que aportan, el azúcar, la grasa, etc., en mayor o menor medida; pero el alcohol es un parámetro muy específico, sólo presente en las bebidas alcohólicas, y la vitamina B6 igualmente, en los alimentos que se enriquecen con ellas. Por eso sólo están presentes en un grupo muy determinado de productos y no en la mayoría.

Apartado c.- Calculad las correlaciones entre las variables numéricas del dataframe DFU_G

Apartado c.1.- Matriz de correlación

En primer lugar, debemos obtener de **DFU_G** una data frame con las 8 últimas columnas, correspondientes a los valores numéricos: energía, grasa, sal, etc.

```
DFU_G_ultimas8=DFU_G[,5:12] # últimas 8 filas
head(DFU_G_ultimas8,2)
```

```
##      energy fat sugars fiber proteins      sodium alcohol vitamin_b6
## 10   1284   7    44   4.7      3.9 0.07874016      NA      NA
## 11   1284   7    44   4.7      3.9 0.07870000      NA      NA
```

Y calculamos su matriz de correlación, a la que he llamado **MCor**. Muestro su valor absoluto. La matriz está redondeada a dos decimales.

```
MCor=round(cor(DFU_G_ultimas8,use="pairwise.complete.obs"),2) # matriz correclaciones
knitr::kable(abs(MCor), digits = 2, # dos decimales
caption = 'Matriz de Valores absolutos de Correlación.',
align=rep("c",8)) # alineación centrado
```

Table 3: Matriz de Valores absolutos de Correlación.

	energy	fat	sugars	fiber	proteins	sodium	alcohol	vitamin_b6
energy	1.00	0.79	0.35	0.28	0.23	0.02	0.05	0.01
fat	0.79	1.00	0.02	0.09	0.15	0.01	0.09	0.03
sugars	0.35	0.02	1.00	0.06	0.23	0.09	0.03	0.05
fiber	0.28	0.09	0.06	1.00	0.23	0.02	0.07	0.03
proteins	0.23	0.15	0.23	0.23	1.00	0.08	0.12	0.06
sodium	0.02	0.01	0.09	0.02	0.08	1.00	0.03	0.02
alcohol	0.05	0.09	0.03	0.07	0.12	0.03	1.00	NA
vitamin_b6	0.01	0.03	0.05	0.03	0.06	0.02	NA	1.00

Como puede observarse, con el método elegido `use="pairwise.complete.obs"`, tan sólo las variables *alcohol* y *vitamina B6* no han podido ser, de ninguna forma relacionadas (su factor de correlación está no disponible -NA-). Este método elimina, *por cada pareja de variables*, las filas entre ellas que presenten, al menos, un valor NA.

Vamos a intentar explicar por qué. En primer lugar vamos a calcular un data frame con las filas de **DFU_G** en las que **alcohol y/o vitamina B6 son NA**. Veremos que son la mayoría.

```
AlgunaEsNA=DFU_G_ultimas8[is.na(DFU_G_ultimas8$alcohol) | is.na(DFU_G_ultimas8$vitamin_b6),7:8]
head(AlgunaEsNA,3)
```

```
##      alcohol vitamin_b6
## 10         NA         NA
## 11         NA         NA
## 13         NA         NA
```

En efecto. Este data frame tiene 37395 filas. Con el método *pairwise.complete.obs* que antes mencionábamos, todas estas filas en el cálculo del coeficiente de correlación entre ambas variables **son eliminadas**, no cuentan.

Pero ¿son todas? No. Aún quedan filas. De hecho, si restamos de la cantidad de filas de la matriz **DFU_G**, que son 37415, las filas de **AlgunaEsNA**, 37395, obtenemos que quedan 20. ¿Qué le ocurren a estas filas para que no calculen un coeficiente de correlación? ¡Vamos a encontrar esas filas!

```
NoSonNingunaNA=DFU_G_ultimas8[!(is.na(DFU_G_ultimas8$alcohol) | is.na(DFU_G_ultimas8$vitamin_b6)),7:8]
cat("Hay",dim(NoSonNingunaNA)[1],"filas con valores disponibles")
```

```
## Hay 20 filas con valores disponibles
```

```
NoSonNingunaNA
```

```
##      alcohol vitamin_b6
## 1087         0  0.000400
## 7250         0  0.002000
## 8571         0  0.000800
## 11434        0  0.000400
## 13549        0  0.002200
## 14401        0  0.000360
## 14558        0  0.001500
## 25654        0  0.000260
## 25741        0  0.000900
## 30009        0  0.001000
## 34849        0  0.001190
## 39076        0  0.000750
## 44976        0  0.000200
## 45326        0  0.001900
## 45886        0  0.000800
## 47851        0  0.000070
## 47852        0  0.000070
## 48346        0  0.000760
## 51479        0  0.000052
## 59604        0  0.000700
```


Como podemos observar, las 20 filas de este data frame, presentan valores de *alcohol* nulos. La desviación típica de la variable alcohol es cero (al igual que la media) y, como el factor de correlación $r_{alc\ vB6} = \frac{s_{alc\ vB6}}{s_{alc} \cdot s_{vB6}}$ tiene en el denominador el valor de $s_{alc} = 0$ simplemente no podemos calcularlo, no tiene sentido.

Podemos concluir, que sin datos con los que calcular el coeficiente de correlación, bien porque presenten las variables valores no disponibles -NA- o bien porque en el resto de datos la desviación típica de una de las variables sea nula, el valor de la correlación para la pareja alcohol -vitamina B6 no está disponible.

Apartado c.2.- Heatmap

He optado por un gráfico *corrplot* que presenta más información sobre la matriz de correlación, tanto gráfica como numérica. En él, se observa claramente como las variables *grasa* (*fat*) y *energía* (*energy*) presentan cierta relación (coef. correlación de 0.79), y, que entre el alcohol y la vitamina B6 no existe este coeficiente.

```
library("corrplot")
corrplot(abs(MCor), title = "Correlación DFU_G. Últimas 8 variables",
  method="color", type="full", tl.col="black", tl.cex=0.7,
  cl.cex=0.5, addgrid.col="royalblue4", bg="yellow",
  tl.offset = 1, cl.align.text="c", cl.pos = "n", cl.ratio=0.2,
  addCoef.col=c("black"), mar=c(2,0,4,0), number.cex=0.6)
```

Correlación DFU_G. Últimas 8 variables

	energy	fat	sugars	fiber	proteins	sodium	alcohol	vitamin_b6
energy	1	0.79	0.35	0.28	0.23	0.02	0.05	0.01
fat	0.79	1	0.02	0.09	0.15	0.01	0.09	0.03
sugars	0.35	0.02	1	0.06	0.23	0.09	0.03	0.05
fiber	0.28	0.09	0.06	1	0.23	0.02	0.07	0.03
proteins	0.23	0.15	0.23	0.23	1	0.08	0.12	0.06
sodium	0.02	0.01	0.09	0.02	0.08	1	0.03	0.02
alcohol	0.05	0.09	0.03	0.07	0.12	0.03	1	?
vitamin_b6	0.01	0.03	0.05	0.03	0.06	0.02	?	1

¿Cómo podemos saber si, efectivamente, entre la variable energía y grasa se presenta el máximo de correlación? La tabla es sencilla y se aprecia de un simple vistazo (color más oscuro: 0.79)... ¿Pero y si tuviera más variables y fuera más complicada? Con el siguiente script, descrito en las páginas 14 a 16 de la lección 10ª, obtenemos los coeficientes de correlación ordenados de mayor a menor, y, podemos asegurar, por tanto, cuál es el mayor.

```

n=length(DFU_G_ultimas8)
indices=upper.tri(diag(n))
medidas=names(DFU_G_ultimas8)
med1=matrix(rep(medidas,times=n),nrow=n,byrow=FALSE)[indices]
med2=matrix(rep(medidas,times=n),nrow=n,byrow=TRUE)[indices]
vecMCor=as.vector(MCor)[indices]
corr_df=data.frame(med1,med2,vecMCor,corr.abs=abs(vecMCor))
corr_df_sort=corr_df[order(corr_df$vecMCor,decreasing=TRUE),]
head(corr_df_sort,3)

```

```

##      med1   med2 vecMCor corr.abs
## 1 energy    fat   0.79    0.79
## 2 energy sugars  0.35    0.35
## 4 energy  fiber  0.28    0.28

```

Al ordenar por los valores de correlación **vecMCor** (con su signo además), nos aseguramos que efectivamente es la pareja [**energy, fat, 0.79**] la de mayor coeficiente de correlación, siendo su signo positivo. Por lo tanto, un alimento más graso será, probablemente, más energético.

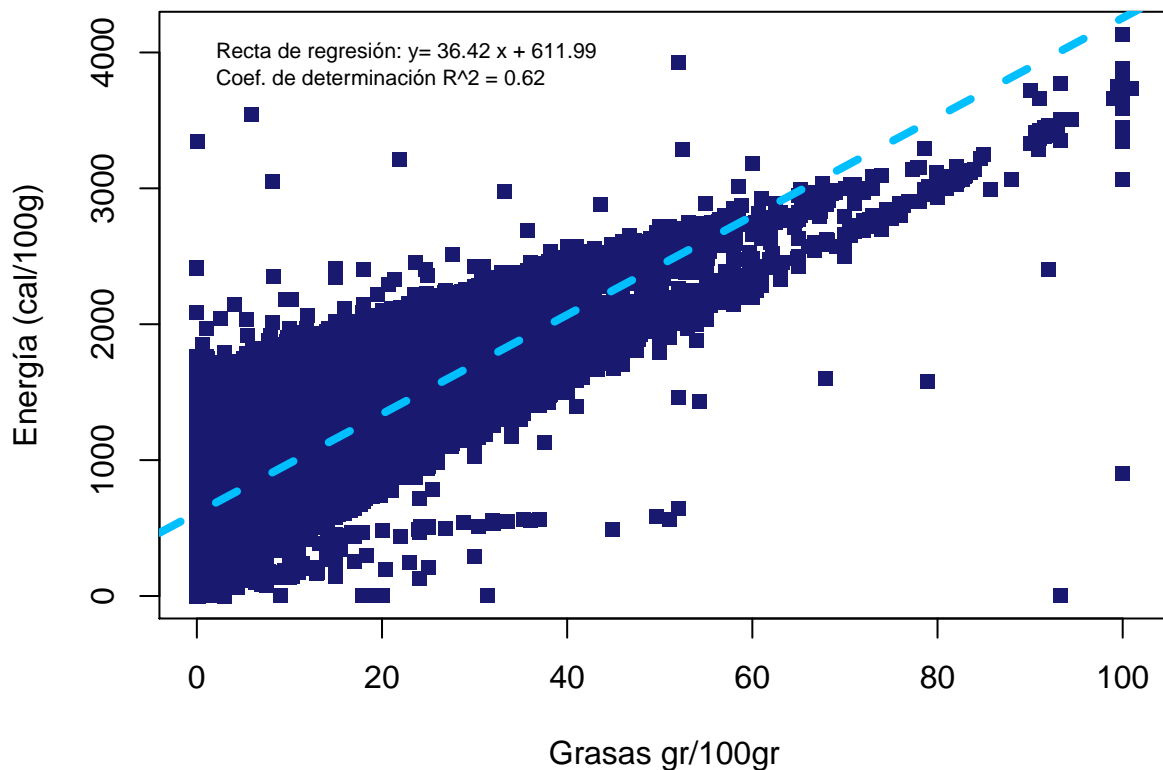
Apartado c.3.- Diagrama de dispersión para las dos variables con mayor correlación: energía y grasa

El ajuste de regresión lineal no es perfecto. Su correlación es de 0.79, por lo que su coeficiente de determinación deberá valer $R^2 = 0.62$. El ajuste sería mucho más fiable si se aproximase a 0.9

```

plot(DFU_G_ultimas8[,2:1],pch=15,col="midnightblue",
     ylab="Energía (cal/100g)",
     xlab="Grasas gr/100gr")
l1=lm(energy~fat,data=DFU_G_ultimas8, na.action = na.omit)
abline(l1,lty="dashed",col="deepskyblue",lwd=4)
a=summary(l1)$coefficients[2]
b=summary(l1)$coefficients[1]
R2=round(summary(l1)$r.squared,2)
textofila2=paste("Recta de regresión: y=",round(a,2),"x +",round(b,2))
text(0,4000,labels=textofila2,cex=0.7,pos=4)
textofila1 = paste("Coef. de determinación R^2 =", R2)
text(0,3800,labels=textofila1,cex=0.7,pos=4)

```



He preferido dibujar la gráfica con las grasas (fat) en el eje de abscisas y en las ordenadas los valores de energía (energy); la gráfica así tiene mejor aspecto y los coeficientes de la recta de regresión también son más comprensibles **energía = 36.424 · grasa + 611.988**. A la vista de la gráfica parece, que en una primera instancia, podemos decir que a más grasa un alimento es más energético o tiene más calorías.

Apartado d.- Vitamina B6 en relación al Actimel de Danone (0.21mg/100g)

Apartado d.1.- Data frame con Vitamina B6 disponible. Localizar el producto problema.

En el siguiente script, calculo un data frame a partir del **DFU_G** en el que la vitamina B6 esté disponible. Lo consigo negando la función *is.na* aplicada a los valores de la variable **vitamin_b6**.

Como los valores en esta variable están especificados en gramos por cada 100 gramos, los transformo en miligramos, multiplicando por 1000. Redondeo los resultados a 3 decimales.

```
DFU_G_vitaminaB6=DFU_G[!is.na(DFU_G$vitamin_b6),c(1,12)]
DFU_G_vitaminaB6$vitamin_b6 = round(DFU_G_vitaminaB6$vitamin_b6*1000,3)
# Encontrar el producto que especifica el problema
library("dplyr") # cargo la biblioteca
a1=filter(DFU_G_vitaminaB6,vitamin_b6 == 0.2100, grepl('Actimel', product_name)) # Filtro el dato
knitr::kable(a1, digits = 2, # dos decimales
caption = 'Producto target encontrado',
align=rep("c",2)) # alineación centrado
```

Table 4: Producto target encontrado

product_name	vitamin_b6
Actimel, Goût Fraise (x 6 Bouteilles)	0.21

Por fin, y aunque no lo pide el ejercicio, compruebo que efectivamente existe un producto cuyo nombre empieza por “Actimel” y cuyo valor es de 0.21mg/100g. Lo consigo usando la orden **filter** de la biblioteca **dplyr**. El producto está localizado en la fila **10135 de DFU_G** y su nombre real es “**Actimel, Goût Fraise (x 6 Bouteilles)**”.

Apartado d.2.- Tablas de frecuencias. Estadísticos.

Puedo encontrar diferentes datos estadísticos de medidas de tendencia central y posición usando la función **summary** aplicado al vector de datos de vitaminas B6: mínimo y máximo, primer, segundo (mediana) y tercer cuartil, además de la media.

```
VB6=DFU_G_vitaminaB6$vitamin_b6
knitr::kable(t(as.matrix(summary(VB6))), digits = 2, # dos decimales
caption = 'Estadísticos de la Vitamina B6',
align=rep("c",6)) # alineación centrado
```

Table 5: Estadísticos de la Vitamina B6

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0	0.35	0.82	1.55	1.2	215

Pero lo que nos resolverá el problema es estudiar su tabla de frecuencias. Calculamos, para cada valor de *Vitamina B6*. También podemos hacerlo, como se describe en la lección 9 para datos cuantitativos. En concreto, calculamos un data frame con los valores de **frecuencia absoluta, relativa, y frecuencia relativa acumulada**, para cada valor de vitamina B6.

```
tVB6.fabs=table(VB6)
tVB6.frel=prop.table(tVB6.fabs)
tVB6.frel.acu=cumsum(tVB6.frel)
# tVB6.frel.acu
DF_VB6_Frecuencias=data.frame(CantidadVB6=row.names(tVB6.fabs)
, fabs=as.numeric(tVB6.fabs)
, frel=as.numeric(tVB6.frel)
, frel.acu=as.numeric(tVB6.frel.acu))
head(DF_VB6_Frecuencias,3)
```

```
## CantidadVB6 fabs      frel      frel.acu
## 1          0      2 0.002945508 0.002945508
## 2       0.024      1 0.001472754 0.004418262
## 3       0.03      1 0.001472754 0.005891016
```

Es importante destacar que al calcular la tabla de frecuencias absolutas con *table*, los valores estudiados se han ordenado de menor a mayor. Con esta premisa en mente(*), podemos seguir el cálculo

Podemos observar, por ejemplo, que hay 43 productos que tienen la misma cantidad de Vitamina B6 que el producto problema, nuestro “**Actimel, Goût Fraise (x 6 Bouteilles)**”. Pero lo que necesitamos es calcular el **porcentaje de productos de la tabla DFU_G que tienen tanta vitamina B6 o más que el “Actimel”**. Puedo hacerlo de varias formas, pero lo haré de la siguiente (describo cada paso del script con letras):

```
por0.21=DF_VB6_Frecuencias[DF_VB6_Frecuencias$CantidadVB6=="0.21",]$frel.acu # [A]
indice0.21=which(DF_VB6_Frecuencias$CantidadVB6=="0.21") # [B]
porMenos0.21=DF_VB6_Frecuencias[indice0.21-1,]$frel.acu # [C]
porMasOIgual0.21=1-porMenos0.21 # [D]
```

- [A] Cálculo $\%(0.21) = 15.91\%$, el porcentaje de productos cuyas frecuencias relativas acumuladas alcanza los 0.21 mg. Gracias a la ordenación de los productos de menor a mayor (*) podemos asegurar que este porcentaje representa la fracción del total de productos que tienen **0.21mg de vitamina B6 o menos**.
 - [A1] Y el *porcentaje de productos que tienen exactamente 0.21mg de vitamina B6* es la frecuencia relativa de esta entrada: **6.33%**
- [B] Pero ese dato no es el que me piden. Antes voy a calcular en qué índice (posición) se encuentra el dato 0.21mg: **32**.
- [C] Y lo que necesito conocer, antes de dar solución a la cuestión definitivamente, es el **porcentaje de productos que tienen estrictamente menos cantidad de 0.21 mg de vitamina B6**. Esto se consigue calculando la frecuencia relativa acumulada del índice **anterior** al calculado en [B]: $\%(<0.21) = 9.57\%$
- [D] **SOLUCIÓN**: por tanto, la fracción de productos que tienen 0.21mg de vitamina B6 o más se consigue restando el resultado del apartado [C] al 100% $\%(>=0.21) = 90.43\%$

Otra forma de hacerlo sería sumando todas las frecuencias relativas desde el índice **32** hasta **158**, el número de filas de nuestro data frame de frecuencias **DF_VB6_Frecuencias**: **90.43%**

Apartado d.3.- Los 5 productos con mayor cantidad de vitamina B6

Aprovechando que tengo cargada la librería *dplyr* uso su función *arrange* para obtener el data frame **VB6_cinco**. En él, ordeno las filas del data frame **DFU_G_vitaminaB6** por orden descendente de cantidad de vitamina B6, y selecciono las cinco primeras.

Una observación. La tabla muestra valores de la variable *vitamin_b6* en miligramos; recordar que en un paso anterior multiplicamos esta variable por 1000.

```
vB6_cinco=head(arrange(DFU_G_vitaminaB6,desc(vitamin_b6)),5)
knitr::kable(vB6_cinco, digits = 2, # dos decimales
caption = '5 productos con mayor vitamina B6',
align=rep("c",2)) # alineación centrado
```

Table 6: 5 productos con mayor vitamina B6

product_name	vitamin_b6
Levure de bière Vitalité Beauté	215.0

product_name	vitamin_b6
Zanahorias "VegaTajo"	200.0
Pop Tarts Hot Fudge Sundae	20.8
bcaa caps	14.0
Levure de bière en paillettes	6.9

Apartado e.- Boxplot Azúcar, Grasa, Sal y N° Aditivos

Pasos previos. Accedemos a la biblioteca RColorBrewer y creo un vector de valores de continente en castellano.

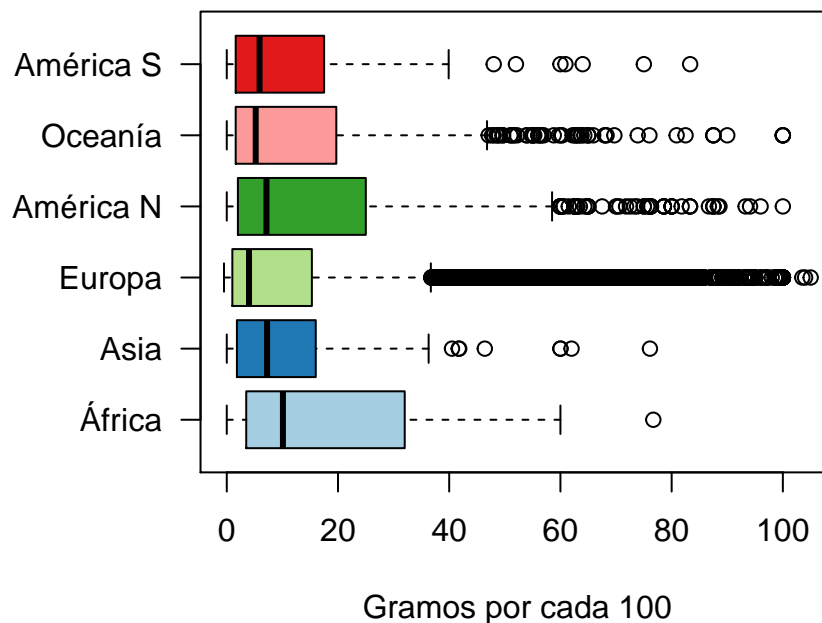
```
continentes2=c("África","Asia","Europa","América N","Oceanía","América S")
library("RColorBrewer")
```

Apartado e.1.- Azúcar

Dibujamos el diagrama de caja correspondiente al azúcar; usamos el data frame **DFU_Continentes**:

```
antiguo.par=par()
par(mar=c(4,8,4,8))
b1 = boxplot(sugars~continent, data=DFU_Continentes, horizontal=TRUE,
  main="Boxplot - Azúcar", xlab="Gramos por cada 100",
  las=1,col=brewer.pal(6,"Paired"), names=continentes2)
# b1$stats
par=antiguo.par
```

Boxplot – Azúcar



Conclusiones

1. Aunque el máximo de las medianas se da en África (10.1) y parece tener además el mayor rango intercuartílico (28.5) y el mayor bigote superior (60), no me atrevería a decir que el mayor consumo de productos con grandes cantidades de azúcar se da en el continente africano. Afirmaría que, en conjunto, los productos africanos contienen más azúcar que la mayoría de productos en Europa (con la menor mediana - 4 - y una pequeña caja intercuartílica).
2. Respecto a la mayoría de productos, tras África encontramos a América del Norte con un comportamiento similar.
3. La primera idea que tuve al ver la cantidad de productos *atípicos* en Europa respecto del azúcar fue pensar que existía una anomalía en una serie de productos respecto a otros continentes; pero esto sólo sería cierto si el número de productos en la lista de cada continente fuera comparable, que no lo es. Lo que sí es cierto es que existen un gran número de productos (respecto al total por cada continente) que se suelen salir de la caja, que tienen más azúcar que la tendencia normal (Ver punto 5)
4. Encontramos los siguientes valores erróneos en Europa, que tienen o azúcar negativa o un contenido de azúcar mayor del 100%. Esto no es posible, y por ello los considero erróneos:

Table 7: Datos erróneos de Azúcar en Europa.

	product_name	continent	sugars
13587	Caprice des dieux	Europe	-0.1
17267	Sirop d'Agave brun bio	Europe	104.0
34163	Sirop d'agave	Europe	105.0
43207	Agavendicksaft	Europe	103.5
57603	Infusion Tilleul	Europe	-0.5

5. En la tabla que sigue, encontramos el porcentaje de productos *atípicos* en cada continente. De la lista de productos que encontramos en cada continente, el porcentaje de ellos cuyas medidas de azúcar no siguen una tendencia normal. Sería pues Oceanía, seguida de Europa, donde se venden, proporcionalmente, más productos con *una tendencia superior a lo normal* en sus niveles de azúcar.

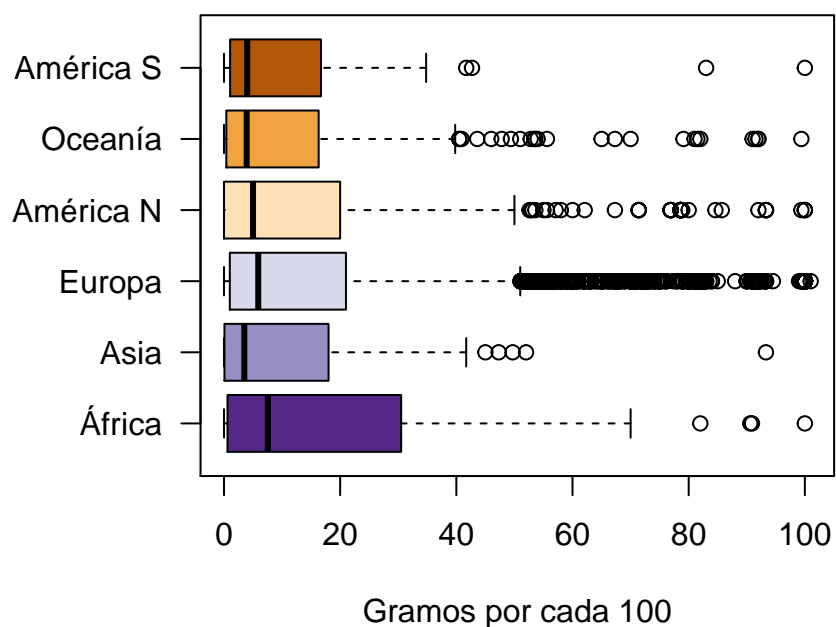
Table 8: Datos atípicos en el DFU_Continentes: azúcar

	África	América N	América S	Asia	Europa	Oceanía
Valores.Atipicos	1.00	70.0	7.00	8.00	3482.00	57.00
Total.Filas	136.00	1428.0	164.00	104.00	35263.00	482.00
Porcentaje	0.74	4.9	4.27	7.69	9.87	11.83

Apartado e.2.- Grasas

Me centraré en los resultados y obviaré los algoritmos de obtención, ya que han sido mostrados en el apartado e.1. Si no, el documento se va a hacer demasiado largo. Sólo se mostrará el código si lo considero ilustrativo.

Boxplot – Grasa



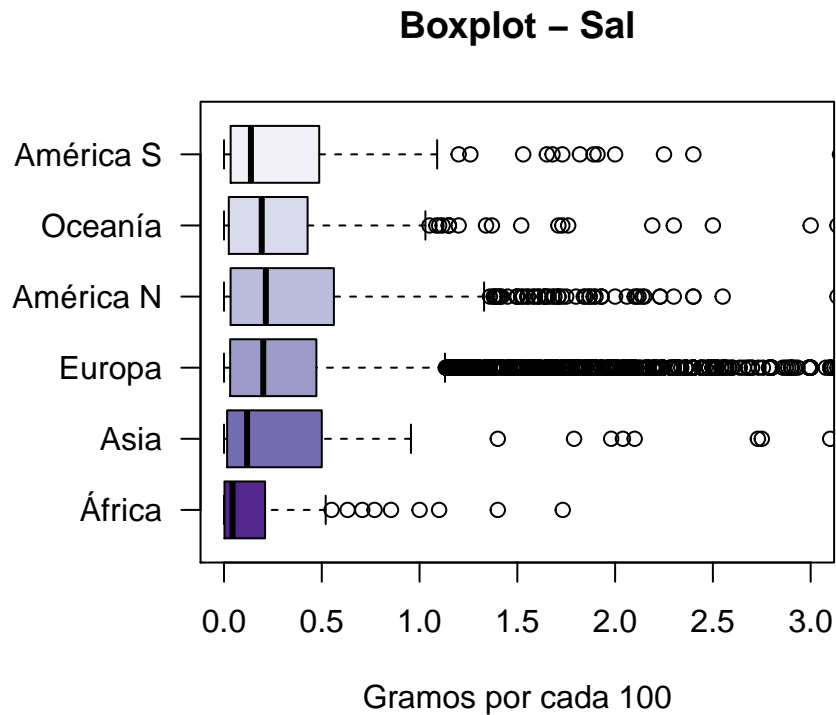
Conclusiones

1. Observamos para la grasa que el continente africano presenta una mayor cantidad de productos grasos: tanto su rango intercuartílico (29.9) y su bigote superior (70) son mayores que el resto. También su mediana 7.55 es mayor que la europea, cuya caja, tanto en la mediana (5.9) como en el tamaño de su rango intercuartílico (20) parece ocupar la segunda posición; con resultados parecidos encontramos a América del Norte.
2. También hay un valor erróneo, cuyo valor de grasa supera los 100g: **Ekstra Jomfru Olivenolie , Europe, 101**
3. Observamos también un comportamiento análogo en los datos *atípicos* respecto de las grasas. Esta vez son Oceanía y Asia los que tienen un mayor porcentaje de productos *atípicos* en cuanto a las grasas. Como su tendencia normal no es de productos grasos - respecto a otros continentes -, los que lo son destacan como atípicos.

Table 9: Datos atípicos en el DFU_Continentes: grasas

	África	América N	América S	Asia	Europa	Oceanía
Valores.Atípicos	4.00	45.00	4.00	5.00	1123.00	25.00
Total.Filas	136.00	1428.00	164.00	104.00	35263.00	482.00
Porcentaje	2.94	3.15	2.44	4.81	3.18	5.19

Apartado e.3.- Sal



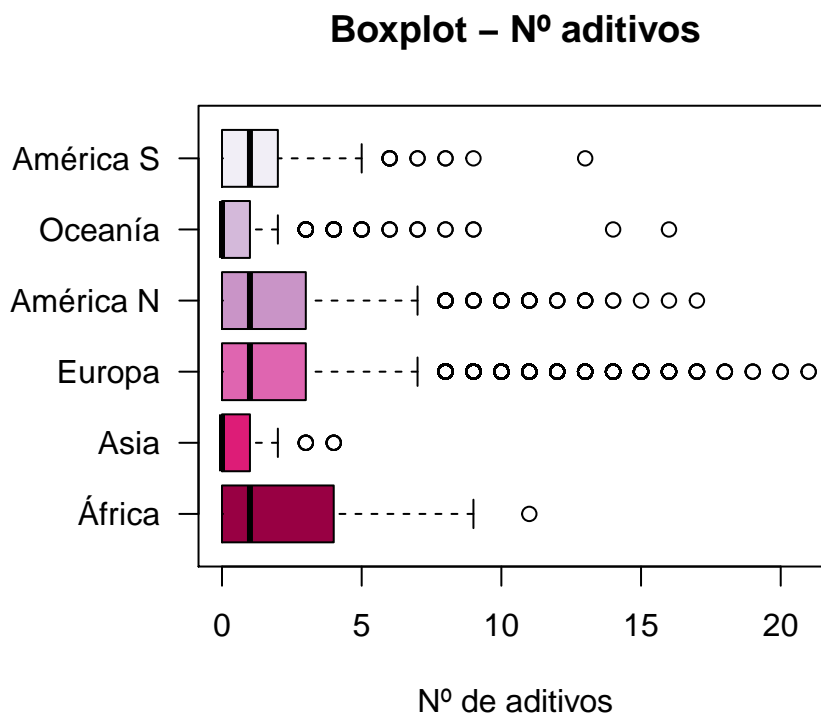
Conclusiones

1. Para la sal he preferido limitar los datos a los resultados para 3g, aunque el máximo sean 40g, ya que las cajas eran muy pequeñas y no se veían. Tal como se observa en el gráfico, la mayor caja la presenta América del Norte con lo que la mayoría de los productos que se venden en este continente tienen más sal que en el resto. Su mediana es (0.21) y su rango intercuartílico (0.53). Detrás parece posicionarse América del Sur, Asia y Europa (con un bigote “whisker” superior mayor). Es también apreciable el poco grado de sal que tienen los productos africanos.
2. No hay valores erróneos.
3. Encontramos también un número apreciable de *atípicos* en Europa. Pero proporcionalmente, Asia y América del Sur tienen un mayor tanto por ciento de productos salados fuera de la tendencia normal.

Table 10: Datos atípicos en el DFU_Continentes: sal

	África	América N	América S	Asia	Europa	Oceanía
Valores.Atipicos	9.00	70.0	20.0	13.0	1469.00	34.00
Total.Filas	136.00	1428.0	164.0	104.0	35263.00	482.00
Porcentaje	6.62	4.9	12.2	12.5	4.17	7.05

Apartado e.3.- N° de aditivos



Conclusiones

1. Como máximo, un producto presenta **21** aditivos. En esta ocasión, son los productos africanos los que parecen presentar un rango intercuartílico mayor (4) seguidos de Europa y América del Norte. Aunque su mediana (1) coincide con las de Europa (1), América del Sur (1) y América del Norte (1).
2. Me parece destacable reseñar que el “whisker” inferior y el límite inferior del rango intercuartílico coinciden en todos los casos (en cero). ¡Al menos, el 25% de los productos no tienen aditivos! y en el caso de Oceanía y Asia también parece coincidir con la mediana, con lo que este porcentaje sube al 50%.
3. Se sigue presentando una gran cantidad de datos *atípicos* primero en Europa y, proporcionalmente en Asia y Oceanía de nuevo; en el caso de estos continentes, como suele haber pocos productos con aditivos, los productos que tienen algunos, se clasifican más fácilmente como atípicos.

Table 11: Datos atípicos en el DFU_Continentes: N° aditivos

	África	América N	América S	Asia	Europa	Oceanía
Valores.Atípicos	1.00	61.00	11.00	10.00	1403.00	55.00
Total.Filas	136.00	1428.00	164.00	104.00	35263.00	482.00
Porcentaje	0.74	4.27	6.71	9.62	3.98	11.41

Apartado f.- Histogramas de bebidas alcohólicas

Apartado f.1.- Histograma de DFU_G (datos globales sin repetir)

Antes de empezar, defino una función que me dibuje el histograma de frecuencias relativas.

```
# =====  
# Histograma de frecuencias relativas  
# =====  
hist_rel=function(x,L,titulo="Frec. relativas y densidad") {  
  h = hist(x,breaks=L,right=FALSE,plot=FALSE) # no lo dibujo aún  
  # Calculo un máximo como o bien la mayor densidad del intervalo h$density  
  # o la mayor de la función density(x)  
  t=round(1.1*max(h$density,max(density(x)[[2]])),2)  
  plot(h, freq=FALSE, xaxt="n",ylim=c(0,t), col="moccasin",  
        main=titulo,  
        xlab="Intervalos", ylab="Densidades")  
  axis(1,at=L)  
  text(h$mids,h$density/2,labels=round(h$counts/length(x),2)  
        ,col="navyblue",srt=90,cex=0.8)  
  lines(density(x),col="maroon4",lwd=2,lty="dashed")  
}
```

A esta función puedo pasarle los datos, el vector que define los intervalos y el título. Dibuja las marcas en el eje X del contenido de alcohol y en el Y las densidades. La frecuencia relativa es el producto de la densidad por la amplitud del intervalo.

Elección del vector de intervalos.

```
BebGlobal=DFU_G[DFU_G$main_category=="Beverages" & DFU_G$alcohol>0  
               & !is.na(DFU_G$alcohol),]$alcohol  
L=c(0,2,seq(4,16,by=2),seq(18,34,by=5),seq(35,49,by=2),seq(50,70,by=5))  
Beb_cut=cut(BebGlobal,breaks=L,right=FALSE)
```

Tras un análisis de los datos, observo que hay como dos zonas donde se concentran la mayor parte de las bebidas alcohólicas: una de baja graduación, entre 4 y 16 gr./100g, y otra entre 35 y 49 gr/100gr de mayor graduación. Elijo pues un vector de intervalos irregulares, definido de 0 a 70 (la mayor graduación es de 69), en el que la amplitud es 2 entre 4 y 16, y entre 35 y 50, y en el resto 5. Los intervalos serían:

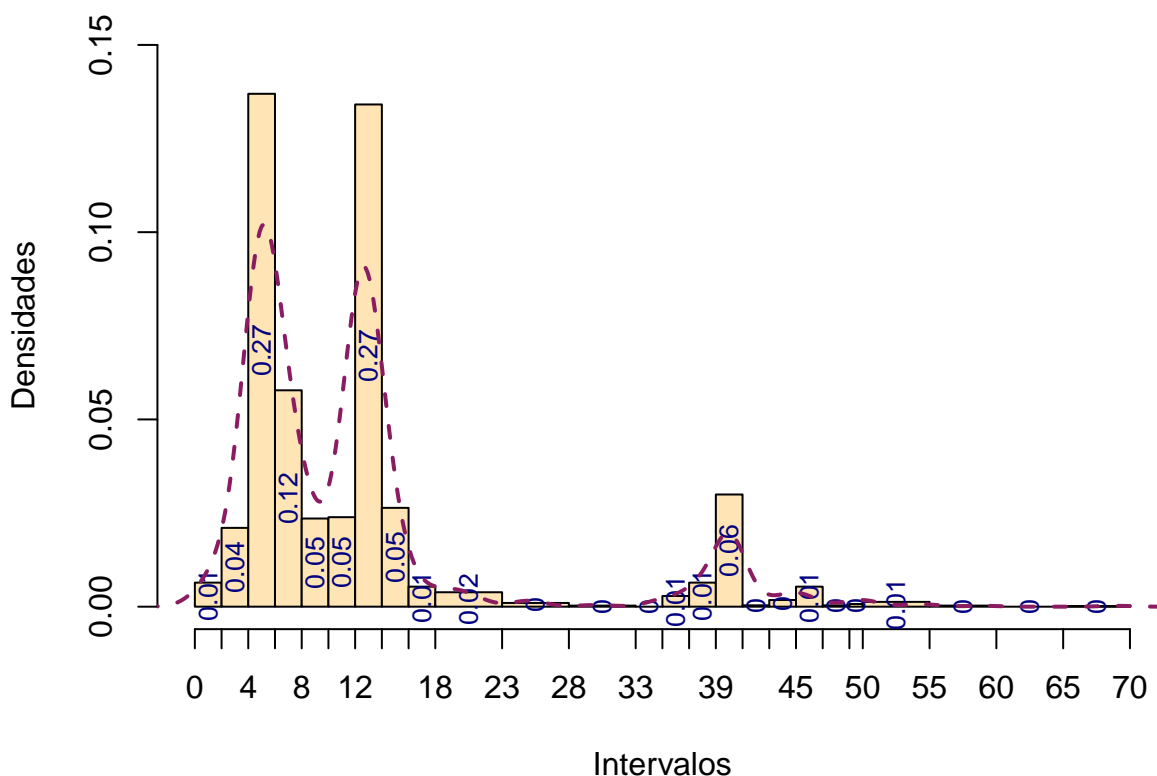
```
levels(Beb_cut)
```

```
## [1] "[0,2)" "[2,4)" "[4,6)" "[6,8)" "[8,10)" "[10,12)" "[12,14)"  
## [8] "[14,16)" "[16,18)" "[18,23)" "[23,28)" "[28,33)" "[33,35)" "[35,37)"  
## [15] "[37,39)" "[39,41)" "[41,43)" "[43,45)" "[45,47)" "[47,49)" "[49,50)"  
## [22] "[50,55)" "[55,60)" "[60,65)" "[65,70)"
```

Para poder comparar, estos intervalos se usarán en todos los histogramas.

Histograma de Frec. Relativas Global

Frec. Relativas de DFU_G

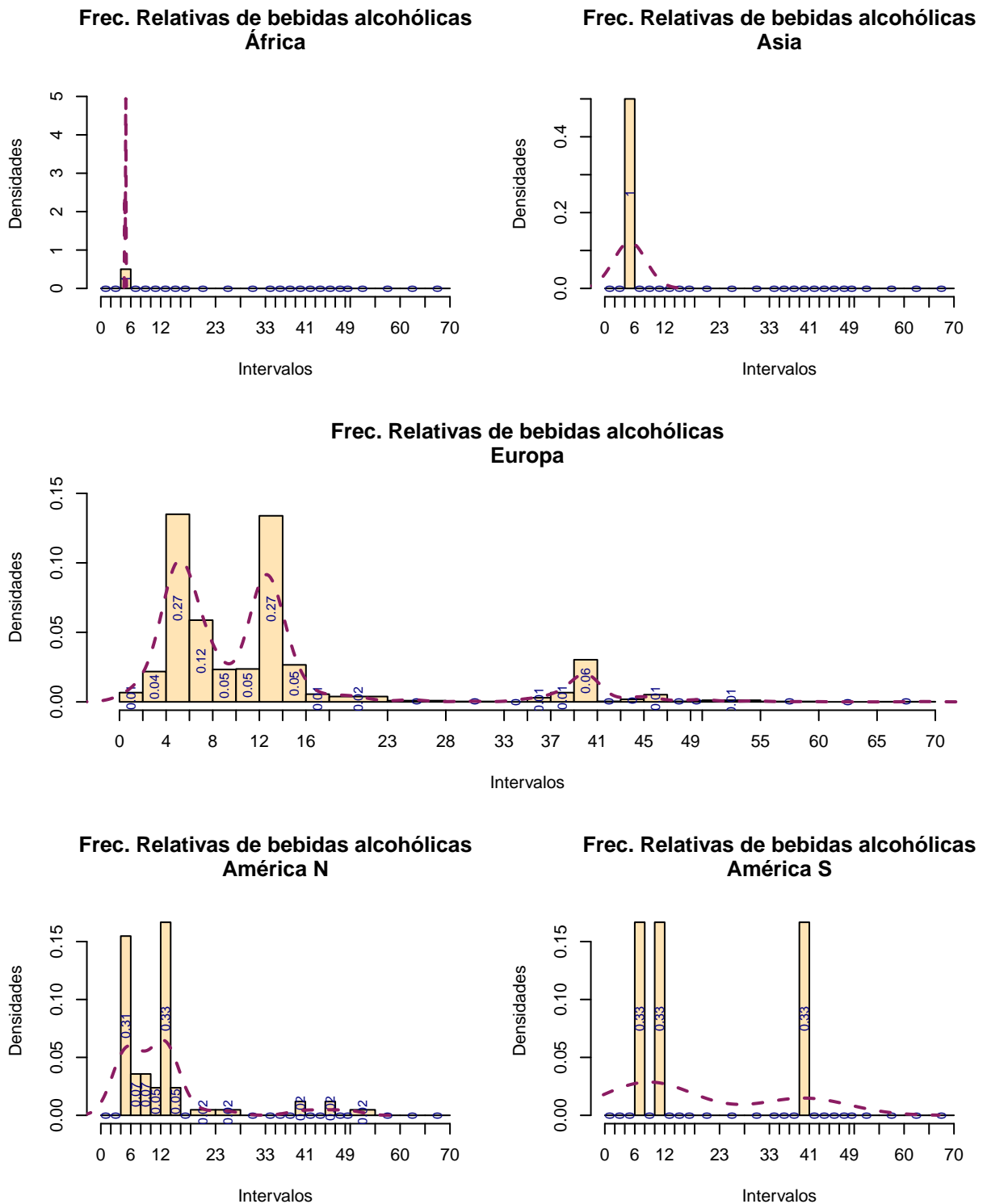


Como puede observarse, casi un 30% de las bebidas tienen entre 4 y 6 gramos de alcohol/100gr. y otro tanto las bebidas entre 12 y 14 gr/100gr. Después hay otro pico, más pequeño, entre 39 y 41gr/100gr. El resto de las bebidas, suelen estar en el intervalo entre 0 y 23gr/100gr, con algunas más alrededor del segundo pico, de mayor graduación.

Apartado f.2.- Histogramas por continentes (datos sin repetir DFU_Continentes)

```
layout(matrix(c(1,2,3,3,4,5),nrow=3,byrow=TRUE))
for (i in continentes){ # bucles for
  Beb = DFU_Continentes[DFU_Continentes$continent==i
                        & DFU_Continentes$main_category=="Beverages"
                        & DFU_Continentes$alcohol>0
                        & !is.na(DFU_Continentes$alcohol),]$alcohol
  ct=trimws(continentes2[which(continentes==i)]) #extrae el nombre correcto
  assign(paste0("Beb_",ct),Beb)
  # Beb_cut=cut(Beb,breaks=L,right=FALSE)
  # maximotabla=max(table(Beb_cut))
  # maximotabla = ceiling(maximotabla/ (10^floor(log10(maximotabla))))*(10^floor(log10(maximotabla)))
  # cat("Maximo tabla: ",maximotabla,"\n")
  # histogramas
  if (length(Beb)>1) {
    hist_rel(Beb,L,paste0("Frec. Relativas de bebidas alcohólicas\n",ct))
    # hist_abs2(Beb,L,maximotabla,paste0("Hist. Frec. Absolutas de beb. alcohólicas\n",ct))
  }
}
```

```
rm(Beb) #borro lo que no necesito
} # Fin del bucle
layout(1)
```



En el script que se muestra, por cada valor del vector continente extraigo un vector de **DFU_Continentes** con los datos de los gramos de alcohol de las bebidas alcohólicas; construyo un vector de nombre diferente

por cada uno y dibujo su histograma (*Beb_Africa, Beb_Asia, etc.*) llamando a la función anterior *hist_rel*. Las líneas comentadas del script se refieren al dibujo del histograma de frecuencias absolutas, que no muestro por no hacer este documento tan largo. Además, rechaza todos aquellos resultados con menos de un dato; el programa lanza un error si calculamos un histograma con tan sólo un dato o ninguno. *Por eso, no aparece reflejado el gráfico de Oceanía, porque sólo tenemos el dato de una bebida alcohólica en la tabla para este continente.*

Respecto de los histogramas, comentar que el global y el europeo son prácticamente idénticos (casi tienen las mismas curvas de densidad). Esto resulta del hecho de que casi todos los datos de bebidas de la tabla son del continente europeo. De hecho, los datos de bebidas alcohólicas en **DFU_G** son 1402 y de 1352 en **DFU_Continentes** lo que representa el **96.43%**.

Las demás gráficas son poco representativas, ya que muestran pocos datos, con excepción quizás de América del Norte. A pesar de que tampoco son muchos los datos de este continente (42) la curva de densidad tiene una forma parecida a la europea y global. Necesitaríamos más datos para asegurar que las bebidas alcohólicas que se consumen en América del Norte tienden a distribuirse, respecto a su cantidad de alcohol, como en Europa o en el resto del mundo.

Apartado g.- Diagrama de barras del grado de nutrición francés

En el siguiente script se realizan las siguientes tareas: primero del data frame **DFU_Continentes** extraigo las columnas correspondientes a *continentes* (2) y a *grado de nutrición francés* (3), pero sólo aquellas en las que el grado de nutrición no sea una cadena vacía; posteriormente, le cambio el nombre a las columnas, a la del grado de nutrición le aplico la función **droplevels** que elimina los niveles que no tiene la variable en ese data frame (elimino así el nivel vacío “”) y, por último, renombro la variable *Continentes*, que es un factor, con las etiquetas correspondientes al vector anterior *continentes2*.

```
a1=DFU_Continentes[!(DFU_Continentes$nutrition_grade_fr==""),c(2,3)]
colnames(a1)=c("Continentes", "Grado.nutrición.francés")
a1$Grado.nutrición.francés=droplevels(a1$Grado.nutrición.francés)
a1$Continentes = factor(a1$Continentes, labels=continentes2)
```

Aplico ahora la función *table* a “a1” (que contiene dos variables) para que me calcule las frecuencias absolutas por cada pareja de variables. Calculo también la frecuencia relativa con *prop.table* y **margin=1**, para calcular la fracción de los grados nutricionales dentro de cada continente (no de forma global)

```
knitr::kable(table(a1), digits = 2, # dos decimales
caption = 'Apariciones de cada elemento \npor
las dos variables continente y grado de nutrición francés',
align=rep("c",6)) # alineación centrado
```

Table 12: Apariciones de cada elemento por las dos variables continente y grado de nutrición francés

	a	b	c	d	e
África	14	7	8	12	20
Asia	2	19	9	13	9
Europa	5581	4444	6240	7592	4813
América N	181	165	211	298	223
Oceanía	74	79	85	106	79
América S	14	14	14	17	10

```
tfrec = round(prop.table(table(a1),margin=1)*100,1)

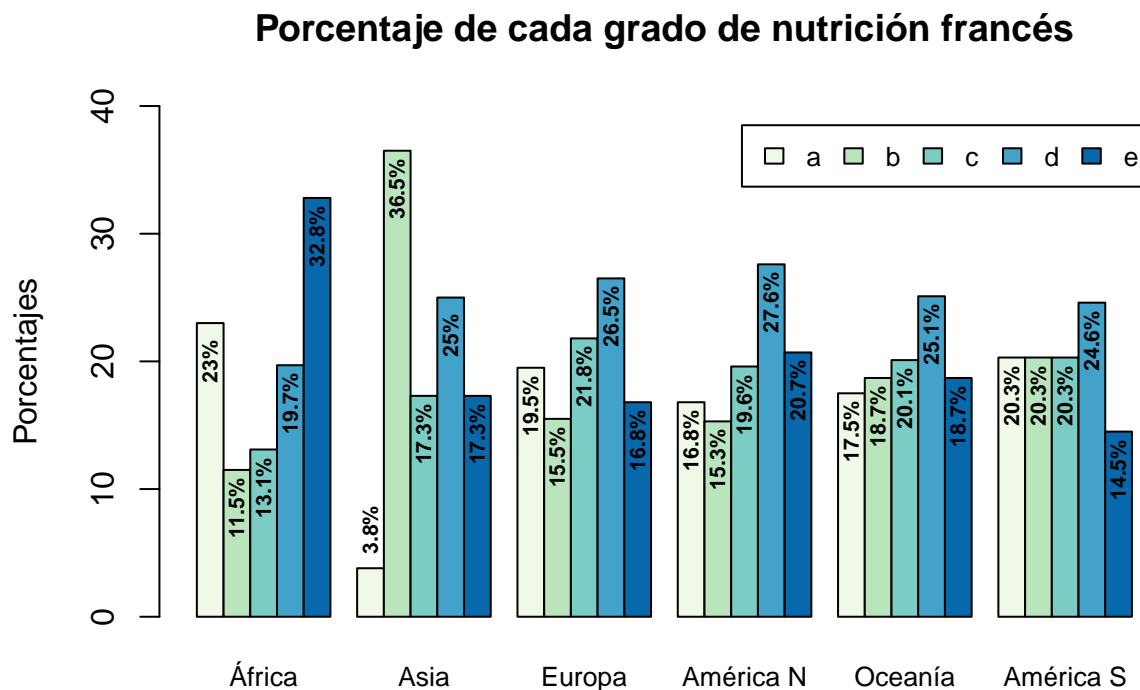
knitr::kable(tfrec, digits = 2, # dos decimales
caption = 'Porcentajes por cada continente\n del grado de nutrición francés',
align=rep("c",6)) # alineación centrado
```

Table 13: Porcentajes por cada continente del grado de nutrición francés

	a	b	c	d	e
África	23.0	11.5	13.1	19.7	32.8
Asia	3.8	36.5	17.3	25.0	17.3
Europa	19.5	15.5	21.8	26.5	16.8
América N	16.8	15.3	19.6	27.6	20.7
Oceanía	17.5	18.7	20.1	25.1	18.7
América S	20.3	20.3	20.3	24.6	14.5

```
# diagrama de barras de los porcentajes
# barplot(t(tfrec),beside=TRUE,legend.text=TRUE)
bp = barplot(t(tfrec),col=brewer.pal(5,"GnBu")
,main="Porcentaje de cada grado de nutrición francés"
,beside = TRUE, legend.text=TRUE, ylab="Porcentajes"
,args.legend = list("topright",cex=0.8, horiz=TRUE)
,ylim = c(0,40), cex.names=0.8)

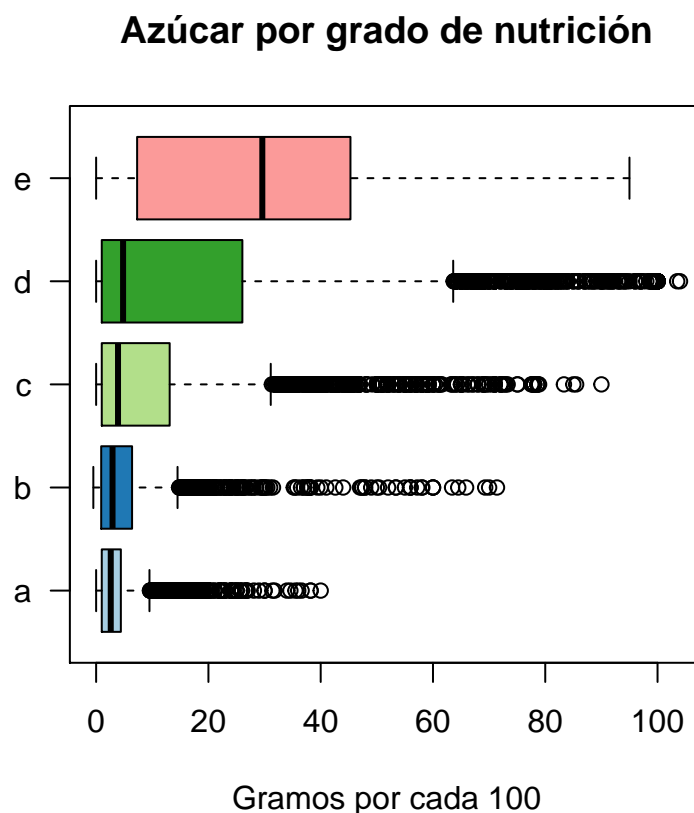
text(bp, t(tfrec)+3*(t(tfrec)<=10)-3*(t(tfrec)>10), paste0(t(tfrec),"%")
,cex=0.7, srt=90, font=2)
```



Apartado h.- Diagramas de caja por grado de nutrición

Partiremos del data frame **DFU_G**, pero retirando los valores en los que *nutrition_grade_fr* es una cadena vacía (Nota técnica: he tenido que instalar texlive-xetex, y activar en Opciones Globales >> Sweave >> Xelatex para que admita el tratamiento de caracteres de los nombres en japonés; si no, hay que eliminar la columna *product_name* en la construcción del data frame **DFU_G_sin**).

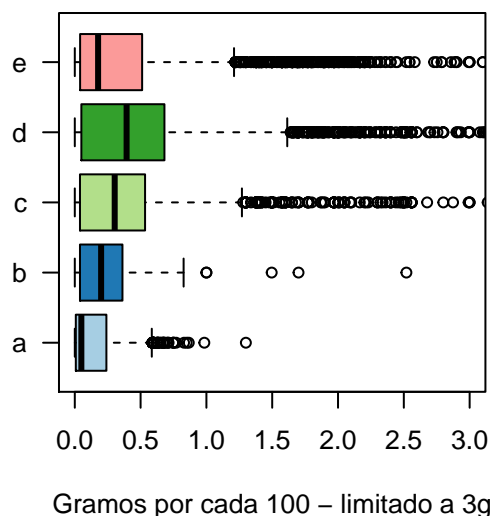
```
DFU_G_sin=DFU_G[!(DFU_G$nutrition_grade_fr==""),]  
DFU_G_sin$nutrition_grade_fr=droplevels(DFU_G_sin$nutrition_grade_fr)  
  
antiguo.par=par()  
par(mar=c(4,8,4,8))  
  
b1 = boxplot(sugars~nutrition_grade_fr, data=DFU_G_sin, horizontal=TRUE,  
             main="Azúcar por grado de nutrición", xlab="Gramos por cada 100",  
             las=1,col=brewer.pal(6,"Paired"))
```



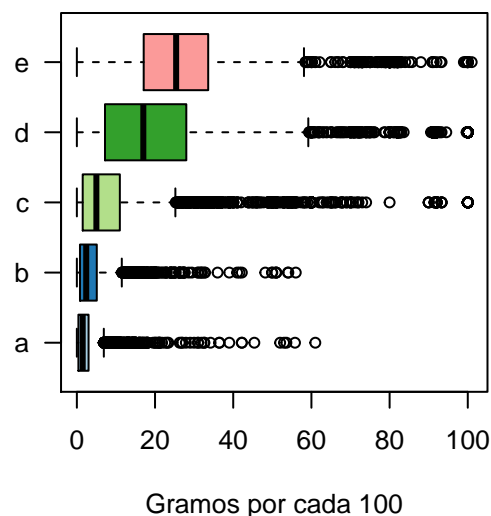
Obtenido el resultado para el azúcar, continúo con la sal, las grasas, la fibra y la proteína. No muestro los scripts que lo llevan a cabo porque son idénticos al script anterior.

Observación: en algunas gráficas, para ver bien las cajas, he limitado el tamaño en las cantidades, no mostrando datos atípicos.

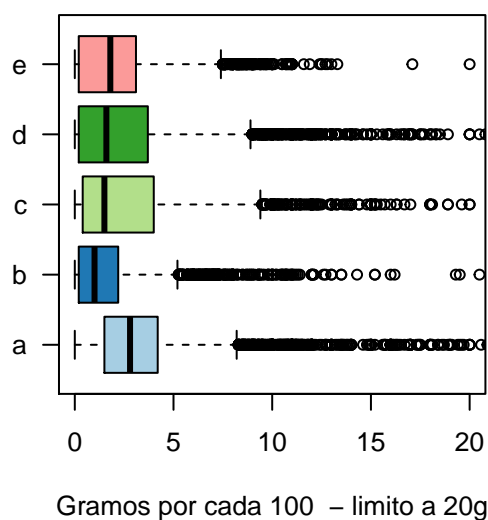
Sal por grado de nutrición



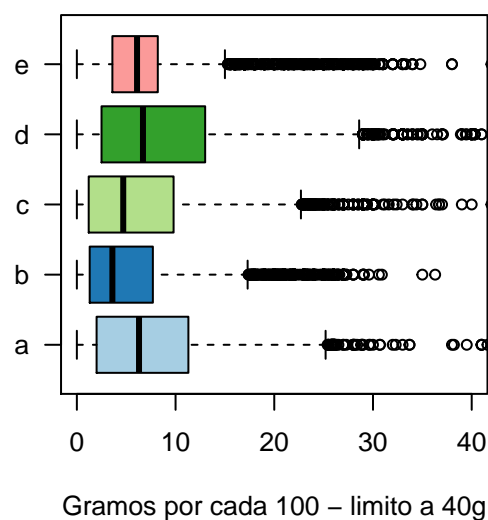
Grasa por grado de nutrición



Fibra por grado de nutrición



Proteínas por grado de nutrición



Explicación: por lo que he podido comprobar en internet (lo siento, no he podido leer el documento porque no entiendo el francés), existe una clasificación de productos de la A a la D (E?) en el que la A indica un producto muy saludable y la D uno poco saludable.

Volviendo a las gráficas, los productos de categoría E tienen cajas grandes en las gráficas del azúcar y la grasa, y algo menos en la de la sal, y no tanta presencia en proteínas y fibra; en el otro extremo, los alimentos de categoría A apenas tienen presencia en grasas, azúcar y sal, y más en fibras y proteínas. La presencia de grasas y azúcares va aumentando, claramente, de la categoría A a la E, y casi en el caso de la sal. Respecto a la fibra y proteínas no está tan claro, pero sí que los alimentos de categoría A presentan más fibra y proteína que otros componentes.

Así que el **grado nutricional francés** es una simple clasificación en base a algún algoritmo que ordena los alimentos en función de su índice saludable, siendo, probablemente **la categoría A muy saludable y la E muy poco saludable**.