

Tarea 1

Representación y tabulación de datos

Julio David Ruiz Mendoza

Mayo 2021

Enunciado

Vamos a analizar los datos que se encuentran en el fichero `tiendas.csv` sobre tiendas de campaña.

1. Descarga el fichero de datos `tiendas.csv` situado en <http://gauss.inf.um.es/datos/> en local o localiza la url donde se encuentra.

Lee el fichero (desde local o desde la url) usando correctamente los argumentos de la función `read.table()`: `header`, `sep`, `dec`. Guarda los datos en una variable llamada `tiendaCamp`.

```
tiendaCamp <- read.table(file="http://gauss.inf.um.es/datos/tiendas.csv", header = TRUE,
                           sep = ";", dec = ".", stringsAsFactors = TRUE)
```

2. Utilizando una sola función responde a las siguientes preguntas:

¿Cuántas tiendas de campaña componen el conjunto de datos? ¿Cuántas variables se han recogido?

```
nrow(tiendaCamp) #tiendas de campaña
```

```
## [1] 90  
ncol(tiendaCamp) #variables
```

```
## [1] 7
```

90 tiendas de campaña y 7 variables

3. Utilizando una sola función responde a las siguientes preguntas:

- ¿Qué variables del conjunto de datos son del tipo...

1. Numérico:
2. Entero:
3. Factor:

- ¿Cuántas marcas de tiendas hay?

```
str(tiendaCamp)
```

```
## 'data.frame': 90 obs. of 7 variables:  
## $ precio : num 350 450 160 360 149 ...  
## $ peso : num 960 1450 1700 2160 2210 1530 2690 1840 3200 850 ...  
## $ altura : int 96 107 102 107 107 117 114 122 97 ...  
## $ marca : Factor w/ 8 levels "coleman","kelty",...: 1 1 2 1 8 1 3 1 8 1 ...  
## $ uso : Factor w/ 3 levels "alpinismo","caravana",...: 3 3 3 3 3 3 3 3 3 ...  
## $ estaciones : Factor w/ 3 levels "otono","primavera",...: 2 2 2 2 2 2 2 2 2 ...  
## $ capacidad.person.: int 2 3 2 3 2 2 3 4 4 1 ...
```

```

unique(tiendaCamp$marca)

## [1] coleman   kelty      rei       marmot     milestone msr       quechua
## [8] nemo
## Levels: coleman kelty marmot milestone msr nemo quechua rei
length(unique(tiendaCamp$marca))

```

[1] 8

1. Numérico: precio, peso
2. Entero: altura y capacidad.person
3. Factor: marca, uso, estaciones

Hay 8 marcas: Coleman, Kelty, Marmot, Milestone, Msr, Nemo, Quechua, Rei.

4. Selecciona las variables cuantitativas, guárdalas en una variable de nombre `tiendaCampC` y haz un resumen de las variables.

```

tiendaCampC <- data.frame(tiendaCamp$precio, tiendaCamp$peso, tiendaCamp$altura,
                           tiendaCamp$capacidad.person.)
names(tiendaCampC) <- c("precio", "peso", "altura", "capacidad.person.")
summary(tiendaCampC)

```

```

##      precio          peso          altura      capacidad.person.
##  Min.   : 99.5   Min.   : 850   Min.   : 90.0   Min.   :1.000
##  1st Qu.:261.5   1st Qu.:1855   1st Qu.:102.0   1st Qu.:2.000
##  Median :344.5   Median :2405   Median :112.0   Median :3.000
##  Mean   :349.2   Mean   :3395   Mean   :125.8   Mean   :3.078
##  3rd Qu.:399.9   3rd Qu.:4756   3rd Qu.:151.5   3rd Qu.:4.000
##  Max.   :700.0   Max.   :9554   Max.   :203.0   Max.   :6.000

```

Con dicha información, responde:

¿Qué precio tienen de media las tiendas de campaña? ¿Cuál es el peso máximo que alcanzan?

Precio medio: 349.2

Peso máximo: 9554

5. Crea dos conjuntos de datos denominados `mochi` y `nomochi`, uno con las tiendas de uso mochilero y otro con las tiendas del resto de usos. ¿Son más caras en media las tiendas para mochileros o las de no mochileros?

```

mochi <- tiendaCamp[tiendaCamp$uso == "mochilero",]
nomochi <- tiendaCamp[tiendaCamp$uso != "mochilero",]

mean(mochi$precio)

## [1] 325.3669
mean(nomochi$precio)

## [1] 394.4306

```

Son más caras en media las de **no mochileros**.

6. Añade una variable categórica al conjunto de datos que se llame `pesocat` que provenga de la recodificación de la variable `peso` con los siguientes valores:

- “ligera”: para cuando pesan menos de 2000 kg
- “media”: cuando pesa entre 2000 y 4000 kg (ambos inclusive)
- “pesada”: cuando pesa más de 4000 kg

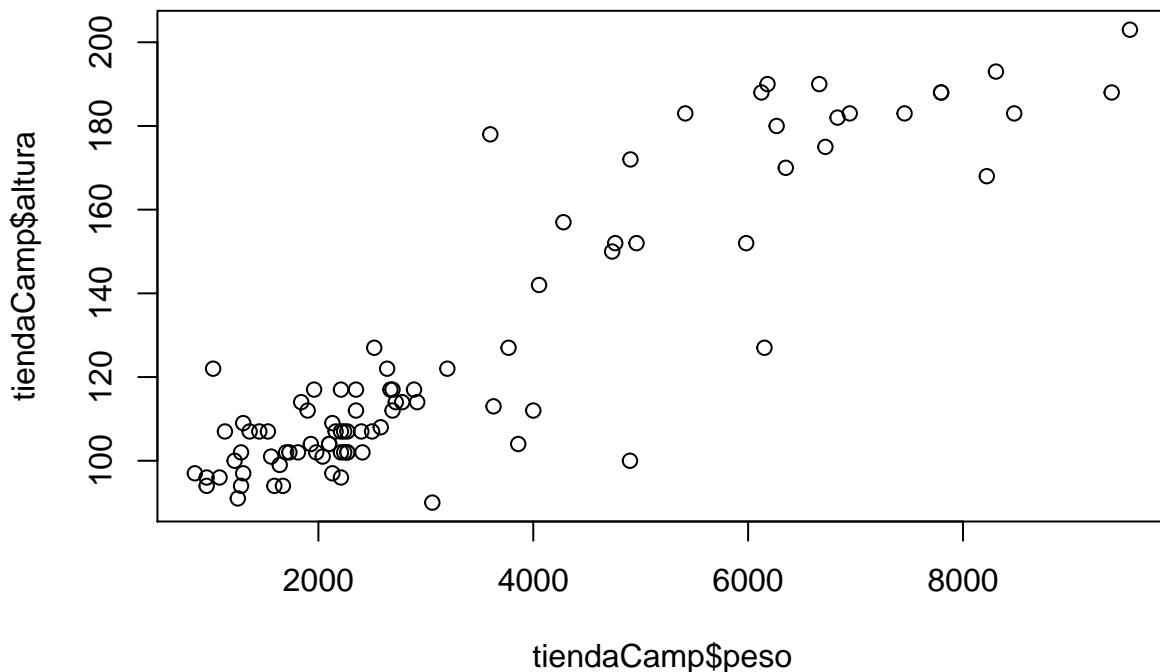
Ayuda: se vió en la sesión 2 de introR (el curso anterior) cómo hacer la recodificación de variables

```
pesocat <- c()  
  
pesocat[tiendaCamp$peso<2000] <- "ligera"  
pesocat[tiendaCamp$peso>=2000 & tiendaCamp$peso<=4000] <- "media"  
pesocat[tiendaCamp$peso>4000] <- "pesada"  
  
tiendaCamp <- cbind(tiendaCamp,pesocat)  
head(tiendaCamp)
```

```
##    precio peso altura marca      uso estaciones capacidad.person. pesocat  
## 1 349.95  960     96 coleman mochilero primavera             2  ligera  
## 2 449.95 1450    107 coleman mochilero primavera             3  ligera  
## 3 159.95 1700    102 kelty mochilero primavera             2  ligera  
## 4 359.95 2160    107 coleman mochilero primavera             3  media  
## 5 149.00 2210    107 rei mochilero primavera             2  media  
## 6 399.95 1530    107 coleman mochilero primavera             2  ligera
```

7. Utiliza Quick-R <https://www.statmethods.net/graphs/creating.html> u otros recursos online para dibujar un gráfico de dispersión simple que enfrente las variables peso y altura.

```
library("ggplot2")  
  
plot(tiendaCamp$peso, tiendaCamp$altura)
```

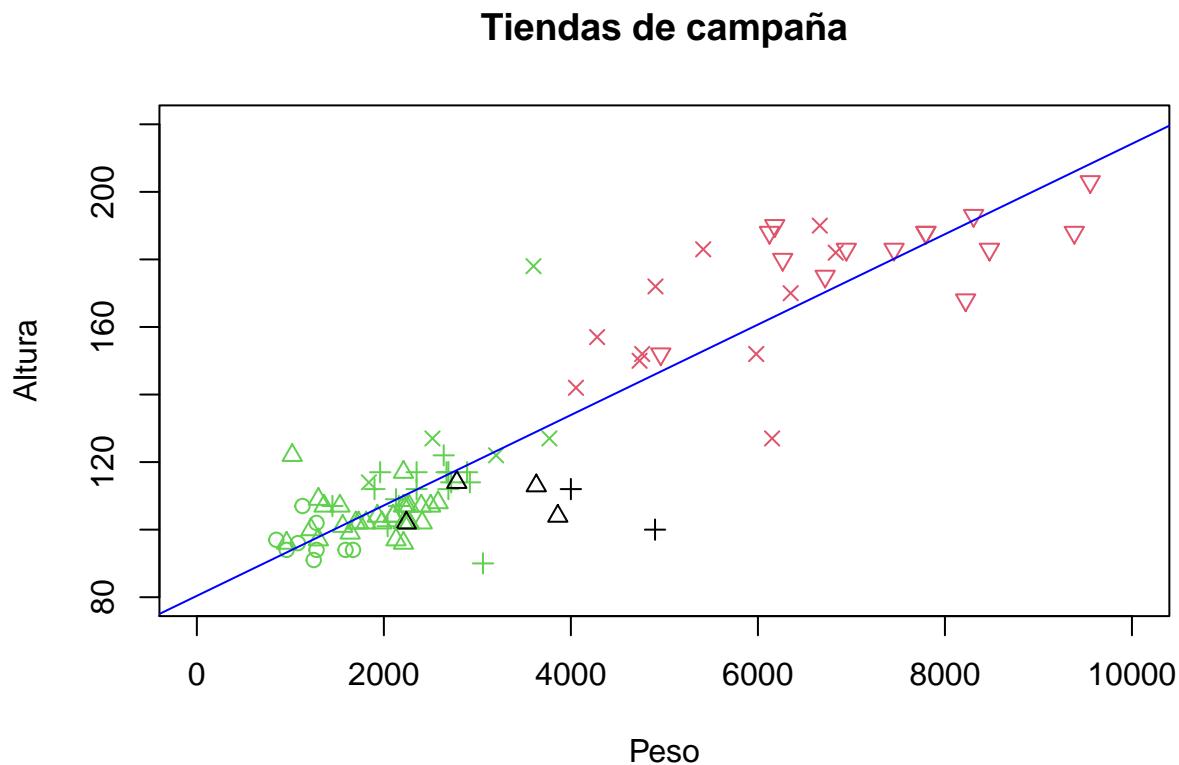


8. Utiliza las ayudas y tu creatividad para mejorar dicho gráfico. Por si necesitas inspiración, puedes:

- Etiquetar los ejes con nombres adecuados.

- Poner un título representativo.
- Especificar el rango de la variable peso entre 0 y 10000.
- Especificar el rango de la variable altura entre 80 y 220.
- Pintar los puntos por colores según la variable uso.
- Utilizar distintos símbolos en función de la capacidad de la tienda.
- Buscar en la ayuda de la función `par()` cómo cambiar el tamaño y color del título.
- Crear un modelo de regresión lineal que se ajuste a la nube de puntos. Utiliza para ello la función `lm(y ~ x, data = tiendaCamp)`. Guarda el modelo en una variable y añádelo al gráfico con la función `abline()`

```
plot(tiendaCamp$peso, tiendaCamp$altura, xlab = 'Peso', ylab = 'Altura',
      main = 'Tiendas de campaña', xlim = c(0,10000), ylim = c(80,220),
      pch = tiendaCamp$capacidad.person., col=tiendaCamp$uso)
ajuste <- lm(altura~peso, data=tiendaCamp)
abline(ajuste, col='blue')
```



¿Qué entregar?

Para realizar esta tarea se utilizará Rmarkdown, por lo tanto, se entregará un fichero .Rmd que contenga tanto el código necesario para responder a las preguntas como la respuesta a las mismas.