Guia 1

Agustin Muñoz Gonzalez

6/5/2020

Limpiamos los registros.

```
rm(list=ls())
```

1

```
x=c(1,2,3)
y=c(6,5,4)
x*2 # Multiplica cada elemento de x por 2
```

```
## [1] 2 4 6
```

```
x*y # Multiplica x_i*y_i
```

```
## [1]  6 10 12
```

```
x[1]*y[2] #Multiplica x_1*y_2
```

```
## [1] 5
```

```
1/x #Devuelve el vector formado por los inversos de cada coord de x
```

```
## [1] 1.0000000 0.5000000 0.3333333
```

```
(1:10)*x[2] #Multiplica cada coord del vector (1,...,10) por x[2]
```

```
##  [1]  2  4  6  8 10 12 14 16 18 20
```

```
rep(c(1,1,2),times=2) #Devuelve el vector (1,1,2,1,1,2)
```

```
## [1] 1 1 2 1 1 2
```

```
#seq(...) devuelve un vector de longitud 5 empezando en 0 y terminando en 10
#Los vectores v_1, v_2 se podrán sumar siempre que length(v_1)=n*length(v_2)
#en tal caso si pensamos a v_1 como n vectores de longitud length(v_2)
#v_1+v_2 le suma a v_2 cada uno de esos n vectores y devuelve el vector de longitud n*length(v_2) resul
seq(from=0,to=10,length.out=5)+1:10
```

```
##  [1]  1.0  4.5  8.0 11.5 15.0  6.0  9.5 13.0 16.5 20.0
```

```
#Veamos que ocurre si se multiplican vectores de distinta longitud
z=c(1,2,3,4)
x*z
```

```
## Warning in x * z: longitud de objeto mayor no es múltiplo de la longitud de uno
## menor
```

```
## [1] 1 4 9 4
```

```
#devuelve error porque length(z)!=n*length(x), para todo n.
```

2

```r
tratamiento=c(rep('A',20),rep('B',18),rep('C',22))
J=seq(1,30,2)
J[1]+J[8]
```

```
## [1] 16
```

3 y 4

```r
sum((1:100))
```

```
## [1] 5050
```

```r
sum((1:100)^2)
```

```
## [1] 338350
```

5

```r
#a) El cjto de datos tiene 153 observaciones y 6 variables.
dim(airquality)
```

```
## [1] 153   6
```

```r
#b) Los nombres de las variables son:
names(airquality)
```

```
## [1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
```

```r
#c) Una forma:
#Gracias a los comandos
attach(airquality) #Para poder buscar las variables sin usar airquality$Ozone, etc. Ver search() para m
Ozone
```

```
##   [1]  41  36  12  18  NA  28  23  19   8  NA   7  16  11  14  18  14  34   6
##  [19]  30  11   1  11   4  32  NA  NA  NA  23  45 115  37  NA  NA  NA  NA  NA
##  [37]  NA  29  NA  71  39  NA  NA  23  NA  NA  21  37  20  12  13  NA  NA  NA
##  [55]  NA  NA  NA  NA  NA  NA  NA 135  49  32  NA  64  40  77  97  97  85  NA
##  [73]  10  27  NA   7  48  35  61  79  63  16  NA  NA  80 108  20  52  82  50
##  [91]  64  59  39   9  16  78  35  66 122  89 110  NA  NA  44  28  65  NA  22
## [109]  59  23  31  44  21   9  NA  45 168  73  NA  76 118  84  85  96  78  73
## [127]  91  47  32  20  23  21  24  44  21  28   9  13  46  18  13  24  16  13
## [145]  23  36   7  14  30  NA  14  18  20
```

```r
Solar.R
```

```
##   [1] 190 118 149 313  NA  NA 299  99  19 194  NA 256 290 274  65 334 307  78
##  [19] 322  44   8 320  25  92  66 266  NA  13 252 223 279 286 287 242 186 220
##  [37] 264 127 273 291 323 259 250 148 332 322 191 284  37 120 137 150  59  91
##  [55] 250 135 127  47  98  31 138 269 248 236 101 175 314 276 267 272 175 139
##  [73] 264 175 291  48 260 274 285 187 220   7 258 295 294 223  81  82 213 275
##  [91] 253 254  83  24  77  NA  NA  NA 255 229 207 222 137 192 273 157  64  71
## [109]  51 115 244 190 259  36 255 212 238 215 153 203 225 237 188 167 197 183
## [127] 189  95  92 252 220 230 259 236 259 238  24 112 237 224  27 238 201 238
## [145]  14 139  49  20 193 145 191 131 223
```

```r
Wind
```

```
##   [1]  7.4  8.0 12.6 11.5 14.3 14.9  8.6 13.8 20.1  8.6  6.9  9.7  9.2 10.9 13.2
##  [16] 11.5 12.0 18.4 11.5  9.7  9.7 16.6  9.7 12.0 16.6 14.9  8.0 12.0 14.9  5.7
##  [31]  7.4  8.6  9.7 16.1  9.2  8.6 14.3  9.7  6.9 13.8 11.5 10.9  9.2  8.0 13.8
##  [46] 11.5 14.9 20.7  9.2 11.5 10.3  6.3  1.7  4.6  6.3  8.0  8.0 10.3 11.5 14.9
##  [61]  8.0  4.1  9.2  9.2 10.9  4.6 10.9  5.1  6.3  5.7  7.4  8.6 14.3 14.9 14.9
##  [76] 14.3  6.9 10.3  6.3  5.1 11.5  6.9  9.7 11.5  8.6  8.0  8.6 12.0  7.4  7.4
##  [91]  7.4  9.2  6.9 13.8  7.4  6.9  7.4  4.6  4.0 10.3  8.0  8.6 11.5 11.5 11.5
## [106]  9.7 11.5 10.3  6.3  7.4 10.9 10.3 15.5 14.3 12.6  9.7  3.4  8.0  5.7  9.7
## [121]  2.3  6.3  6.3  6.9  5.1  2.8  4.6  7.4 15.5 10.9 10.3 10.9  9.7 14.9 15.5
## [136]  6.3 10.9 11.5  6.9 13.8 10.3 10.3  8.0 12.6  9.2 10.3 10.3 16.6  6.9 13.2
## [151] 14.3  8.0 11.5
```

```r
Temp
```

```
##   [1] 67 72 74 62 56 66 65 59 61 69 74 69 66 68 58 64 66 57 68 62 59 73 61 61 57
##  [26] 58 57 67 81 79 76 78 74 67 84 85 79 82 87 90 87 93 92 82 80 79 77 72 65 73
##  [51] 76 77 76 76 76 75 78 73 80 77 83 84 85 81 84 83 83 88 92 92 89 82 73 81 91
##  [76] 80 81 82 84 87 85 74 81 82 86 85 82 86 88 86 83 81 81 81 82 86 85 87 89 90
```

```
## [101] 90 92 86 86 82 80 79 77 79 76 78 78 77 72 75 79 81 86 88 97 94 96 94 91 92
## [126] 93 93 87 84 80 78 75 73 81 76 77 71 71 78 67 76 68 82 64 71 81 69 63 70 77
## [151] 75 76 68
```

```r
Month
```

```
##   [1] 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6
##  [38] 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7
##  [75] 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
## [112] 8 8 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9
## [149] 9 9 9 9 9
```

```r
Day
```

```
##   [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
##  [26] 26 27 28 29 30 31  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
##  [51] 20 21 22 23 24 25 26 27 28 29 30  1  2  3  4  5  6  7  8  9 10 11 12 13 14
##  [76] 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31  1  2  3  4  5  6  7  8
## [101]  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31  1  2
## [126]  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
## [151] 28 29 30
```

```r
#Vemos que las variables con datos faltantes son Ozone y Solar.R.
#(no es muy eficiente esta resolución, pero no se como buscar elementos que cumplan cierta función)
#Otra forma:
#which(Ozone==is.na) #no funca, ver dps de la duda.

#d) El mes de Mayo tiene 31 observaciones. En gral cada mes tiene tanas obs como su cantidad de dias.
length(which(Month==5)) #which devuelve las coord que satisfacen la condición.
```

```
## [1] 31
```

6

```
attach(mtcars)
#a)
mtcars[mtcars$gear==4,] #si no ataché mtcars
```

```
##                 mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Merc 240D      24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230       22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280       19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C      17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Fiat 128       32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Fiat X1-9      27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
mtcars[gear==4,] #si ataché mtcars
```

```
##                 mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Merc 240D      24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230       22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280       19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C      17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Fiat 128       32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Fiat X1-9      27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
#b)
mtcars[disp > 150 & mpg > 20,]
```

```
##                 mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
```

```
#devuelve los autos tq disp > 150 y mpg > 20.

#c)
rownames(mtcars[gear == 4 & am==1,])
```

```
## [1] "Mazda RX4"      "Mazda RX4 Wag"  "Datsun 710"     "Fiat 128"
## [5] "Honda Civic"    "Toyota Corolla" "Fiat X1-9"      "Volvo 142E"
```

```
#d)
#a través de las columnas
colMeans(mtcars[carb == 2,][1])
```

```
##  mpg
```

```
## 22.4
```

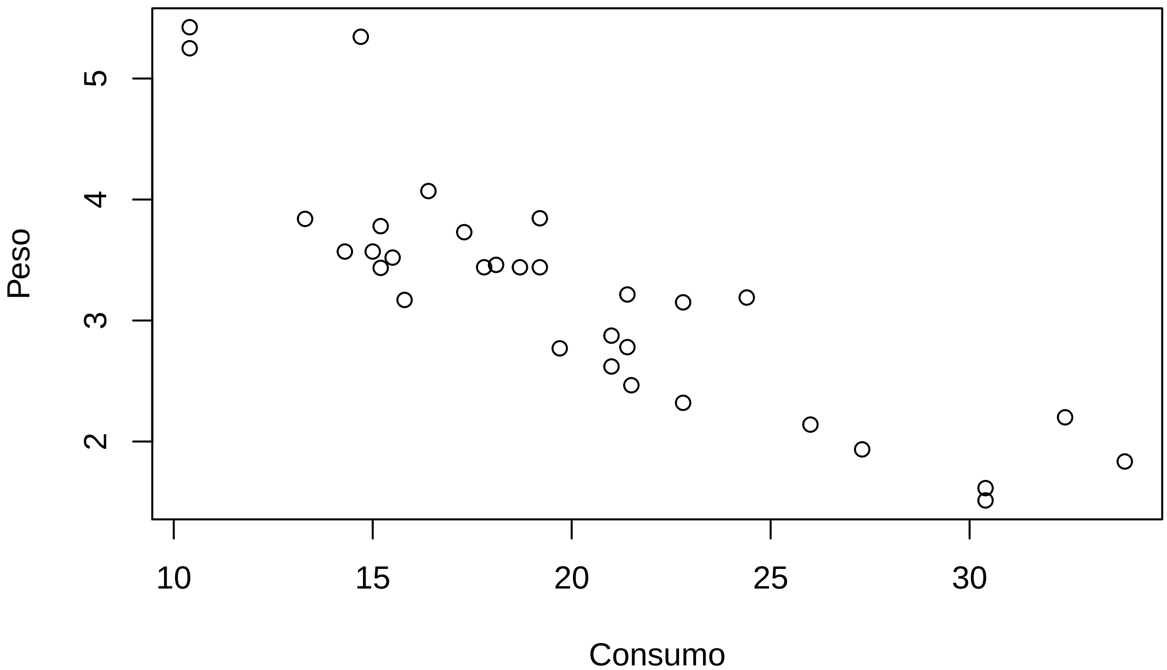
```r
#Una forma de cargar sólo la columna (sin los nombres) es mtcars[carb == 2,][,1]
#en general datos[,n] y datos[n,] carga la columna n y la fila n resp. de lo cjto de datos 'datos'.
#Para buscar por nombre usar las comillas Ej: mtcars["Volvo 142E",]
#a través de las filas
mean(mpg[carb==2])
```

```
## [1] 22.4
```

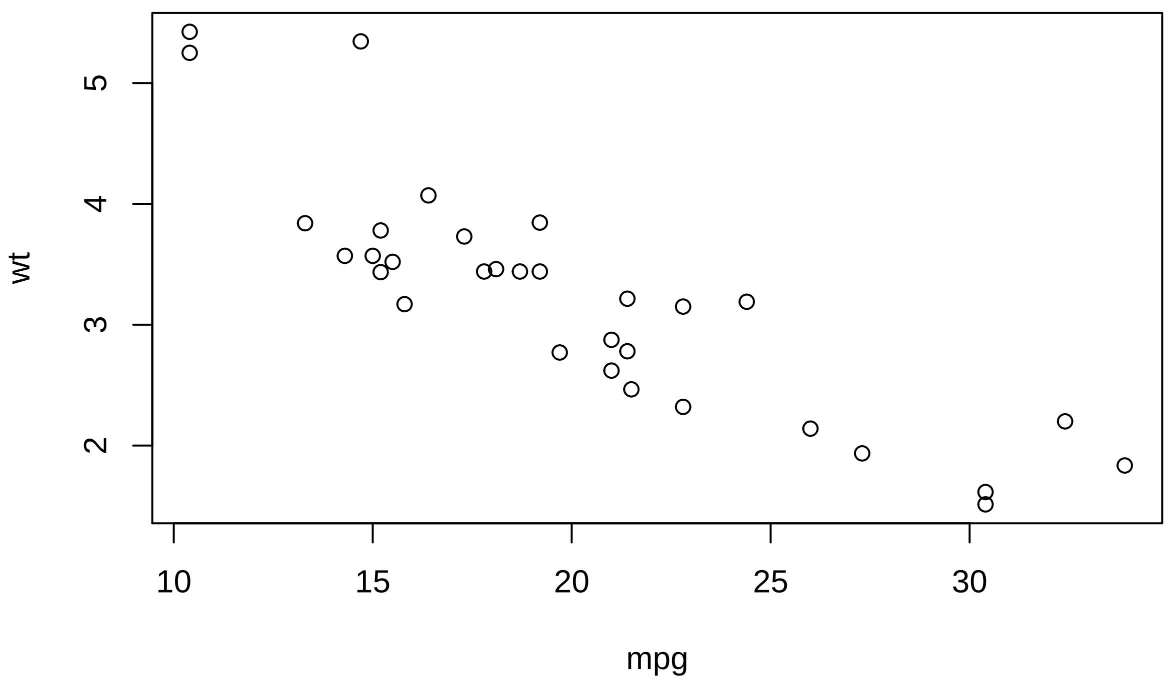
```r
data=data.frame("Desplazamiento"=mtcars["drat"]) #¿cómo se hacía esto?

mtcars[1]
```

```
##                      mpg
## Mazda RX4           21.0
## Mazda RX4 Wag       21.0
## Datsun 710          22.8
## Hornet 4 Drive      21.4
## Hornet Sportabout   18.7
## Valiant             18.1
## Duster 360          14.3
## Merc 240D           24.4
## Merc 230            22.8
## Merc 280            19.2
## Merc 280C           17.8
## Merc 450SE          16.4
## Merc 450SL          17.3
## Merc 450SLC         15.2
## Cadillac Fleetwood  10.4
## Lincoln Continental 10.4
## Chrysler Imperial   14.7
## Fiat 128            32.4
## Honda Civic         30.4
## Toyota Corolla      33.9
## Toyota Corona       21.5
## Dodge Challenger    15.5
## AMC Javelin         15.2
## Camaro Z28          13.3
## Pontiac Firebird    19.2
## Fiat X1-9           27.3
## Porsche 914-2       26.0
## Lotus Europa        30.4
## Ford Pantera L      15.8
## Ferrari Dino        19.7
## Maserati Bora       15.0
## Volvo 142E          21.4
```

```r
#Graficar mpg x wt
#plot(mtcars["mpg"],mtcars["wt"]) #ojo que esto carga las columnas, i.e. como dataframe
plot(mtcars["mpg"][,1],mtcars["wt"][,1],xlab="Consumo",ylab="Peso")
```




```
plot(mpg,wt)#aca los cargo como filas
```



7

```r
#Primero vamos al directorio actual que es donde tenemos el archivo arbolado-en-espacios-verdes.csv
setwd("~/Escritorio/1er Cuatri/Ciencia de Datos con R_ Fundamentos Estadisticos/Prácticas/Guia 1/Resolu


#a)
arboles=read.csv('arbolado-en-espacios-verdes.csv')
#En el bloque de enviroment vemos que el cjto de datos tiene 51502 obs y 17 variables.

#b)
names(arboles)
```

```
##  [1] "long"       "lat"        "id_arbol"   "altura_tot" "diametro"
##  [6] "inclinacio" "id_especie" "nombre_com" "nombre_cie" "tipo_folla"
## [11] "espacio_ve" "ubicacion"  "nombre_fam" "nombre_gen" "origen"
## [16] "coord_x"    "coord_y"
```

```r
#c)
attach(arboles)
mean(altura_tot)
```

```
## [1] 12.1671
```

```r
#d)
length(arboles[espacio_ve == 'ARENALES',1])
```

```
## [1] 198
```

```r
#notar que arboles[espacio_ve == 'ARENALES',1]==arboles[espacio_ve == 'ARENALES',][1]
#pero arboles[espacio_ve == 'ARENALES',][1] lee la col 1 pero como una lista con 1 solo elementos forma
#en cambio arboles[espacio_ve == 'ARENALES',1] lee la columna 1 como una lista (i.e. length=length(col

#e)
arboles_cercanos = data.frame(arboles[espacio_ve == 'LAGO REGATAS',])

#f)
unique(arboles_cercanos$nombre_com)
```

```
##  [1] Eucalipto                          Malus
##  [3] Tipa blanca                        Fresno (Fresno común)
##  [5] Jacarandá                          Fenix
##  [7] Gomero                             Eucalipto (Eucalipto común)
##  [9] Ciprés                             Tuya
## [11] Azarero                            Ligustro
## [13] Arce negundo                       Palma Bangalow  (Palma Rey)
## [15] Morera blanca                      Rafis (Palmerita china)
## [17] Ceibo                              Coculus, Cóculo
## [19] Celtis tala                        Palma de california
## [21] Ombú                               Plátano
## [23] Eucalipto sideroxylon              Ahuehuete
## [25] Roble palustre                     Paraíso
## [27] Corona de cristo                   Pindó
## [29] Ciprés calvo                       Robusta
## [31] Liquidambar                        Lapacho rosado
## [33] Avellano común                     Cedro del Himalaya
## [35] Níspero japonés                    Falso cafeto
```

```
## [37] Sófora japónica                    Olmo
## [39] Tuja                               No Determinable
## [41] Falso Alcanforero                  Palo borracho rosado
## [43] Timbó (Oreja de negro)             Álamo carolina
## [45] Palo borracho                      Ligustrina
## [47] Drácena indivisa                   Limpiatubos
## [49] Bunya-bunya (Araucaria de Bidwill) Ficus
## [51] Acacia blanca
## 337 Levels: Abedul blanco ... Yuca
```

8

```r
library('PASWR2')
```

```
## Loading required package: lattice
## Loading required package: ggplot2
##
## Attaching package: 'ggplot2'
## The following object is masked from 'mtcars':
##
##     mpg
```

```r
tit=TITANIC3
attach(tit)

#a)
mean(tit[pclass=='1st','survived']) #prop_1st
```

```
## [1] 0.619195
```

```r
mean(tit[pclass=='2nd','survived']) #prop_2nd
```

```
## [1] 0.4296029
```

```r
mean(tit[pclass=='3rd','survived']) #prop_3rd
```

```
## [1] 0.2552891
```

```r
#b)
mean(tit[pclass=='1st' & sex =='male','survived']) #prop 1st male
```

```
## [1] 0.3407821
```

```r
mean(tit[pclass=='1st' & sex =='female','survived']) #prop 1st female
```

```
## [1] 0.9652778
```

```r
mean(tit[pclass=='2nd' & sex =='male','survived']) #prop 2nd male
```

```
## [1] 0.1461988
```

```r
mean(tit[pclass=='2nd' & sex =='female','survived']) #prop 2nd female
```

```
## [1] 0.8867925
```

```r
mean(tit[pclass=='3rd' & sex =='male','survived']) #prop 3rd male
```

```
## [1] 0.1521298
```

```r
mean(tit[pclass=='3rd' & sex =='female','survived']) #prop 3rd female
```

```
## [1] 0.4907407
```

```r
#Las mujeres de 3ra tuvieron una tasa mas alta de supervivencia que los varones de 1ra.
#Y en general las mujeres de cada clase tuvieran una tasa de supervivencia muy superior que la de los h

#c)
max(tit[survived == 1 & sex=='female','age'], na.rm=T)
```

```
## [1] 76
```

9

```r
library(PASWR2)
data("CARS2004")
car=CARS2004
attach(car)
```

```
## The following object is masked from package:datasets:
##
##     cars
```

```r
#a)
#una forma
for (i in levels(country))
{print(i)
print(car[country==i,2]*car[country==i,4])}
```

```
## [1] "Austria"
## [1] 4065114
## [1] "Belgium"
## [1] 4854932
## [1] "Cyprus"
## [1] 327040
## [1] "Czech Republic"
## [1] 3809076
## [1] "Denmark"
## [1] 1910892
## [1] "Estonia"
## [1] 472850
## [1] "Finland"
## [1] 2338560
## [1] "France"
## [1] 29411391
## [1] "Germany"
## [1] 45062472
## [1] "Greece"
## [1] 3842268
## [1] "Hungary"
## [1] 2832760
## [1] "Ireland"
## [1] 1550780
## [1] "Italy"
## [1] 33632928
## [1] "Latvia"
## [1] 688743
## [1] "Lithuania"
## [1] 1323264
## [1] "Luxembourg"
## [1] 297868
## [1] "Malta"
## [1] 210000
## [1] "Netherlands"
## [1] 6974682
## [1] "Poland"
## [1] 11991974
```

```
## [1] "Portugal"
## [1] 5991700
## [1] "Slovakia"
## [1] 1194360
## [1] "Slovenia"
## [1] 910176
## [1] "Spain"
## [1] 19224630
## [1] "Sweden"
## [1] 4093056
## [1] "United Kingdom"
## [1] 27618876
```

```r
#otra
for (i in country)
{print(i)
  print(car[country==i,2]*car[country==i,4])}
```

```
## [1] "Belgium"
## [1] 4854932
## [1] "Czech Republic"
## [1] 3809076
## [1] "Denmark"
## [1] 1910892
## [1] "Germany"
## [1] 45062472
## [1] "Estonia"
## [1] 472850
## [1] "Greece"
## [1] 3842268
## [1] "Spain"
## [1] 19224630
## [1] "France"
## [1] 29411391
## [1] "Ireland"
## [1] 1550780
## [1] "Italy"
## [1] 33632928
## [1] "Cyprus"
## [1] 327040
## [1] "Latvia"
## [1] 688743
## [1] "Lithuania"
## [1] 1323264
## [1] "Luxembourg"
## [1] 297868
## [1] "Hungary"
## [1] 2832760
## [1] "Malta"
## [1] 210000
## [1] "Netherlands"
## [1] 6974682
## [1] "Austria"
## [1] 4065114
## [1] "Poland"
```

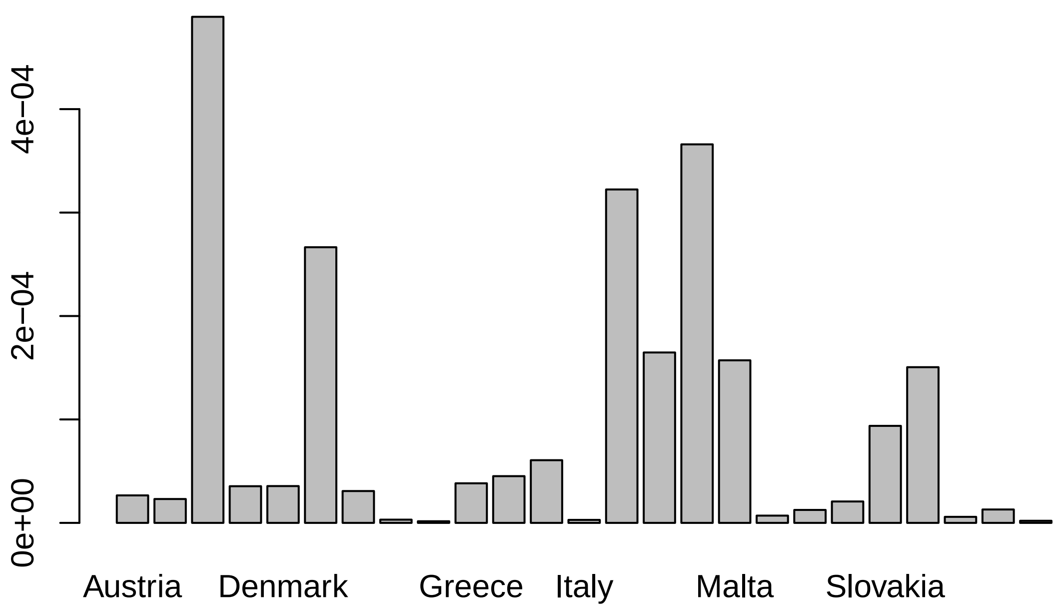
```
## [1] 11991974
## [1] "Portugal"
## [1] 5991700
## [1] "Slovenia"
## [1] 910176
## [1] "Slovakia"
## [1] 1194360
## [1] "Finland"
## [1] 2338560
## [1] "Sweden"
## [1] 4093056
## [1] "United Kingdom"
## [1] 27618876
```

```
#la diferencia entre levels(country) y country, es que levels(country) está ordenado alfabet.

#otra
total_autos=c()
for (i in country)
{total_autos=append(total_autos,c(i,car[country==i,2]*car[country==i,4]))}
#b)
for (i in levels(country))
{print(i)
  print(car[country==i,3]/(car[country==i,2]*car[country==i,4]))}
```

```
## [1] "Austria"
## [1] 2.656752e-05
## [1] "Belgium"
## [1] 2.306932e-05
## [1] "Cyprus"
## [1] 0.0004892368
## [1] "Czech Republic"
## [1] 3.544167e-05
## [1] "Denmark"
## [1] 3.558548e-05
## [1] "Estonia"
## [1] 0.0002664693
## [1] "Finland"
## [1] 3.078818e-05
## [1] "France"
## [1] 3.12804e-06
## [1] "Germany"
## [1] 1.57559e-06
## [1] "Greece"
## [1] 3.825865e-05
## [1] "Hungary"
## [1] 4.518561e-05
## [1] "Ireland"
## [1] 6.061466e-05
## [1] "Italy"
## [1] 2.884078e-06
## [1] "Latvia"
## [1] 0.0003223263
## [1] "Lithuania"
## [1] 0.0001647441
```

```
## [1] "Luxembourg"
## [1] 0.0003659339
## [1] "Malta"
## [1] 0.0001571429
## [1] "Netherlands"
## [1] 7.02541e-06
## [1] "Poland"
## [1] 1.250837e-05
## [1] "Portugal"
## [1] 2.06953e-05
## [1] "Slovakia"
## [1] 9.377407e-05
## [1] "Slovenia"
## [1] 0.0001505203
## [1] "Spain"
## [1] 5.82586e-06
## [1] "Sweden"
## [1] 1.294876e-05
## [1] "United Kingdom"
## [1] 2.027599e-06
```

```r
#c)
tasa_muerte=c()
for (i in levels(country))
{tasa_muerte=append(tasa_muerte,car[country==i,3]/(car[country==i,2]*car[country==i,4]))}
#mortandad=data.frame(pais=levels(country),tasa_muerte)
barplot(height=tasa_muerte,names.arg=levels(country))
```



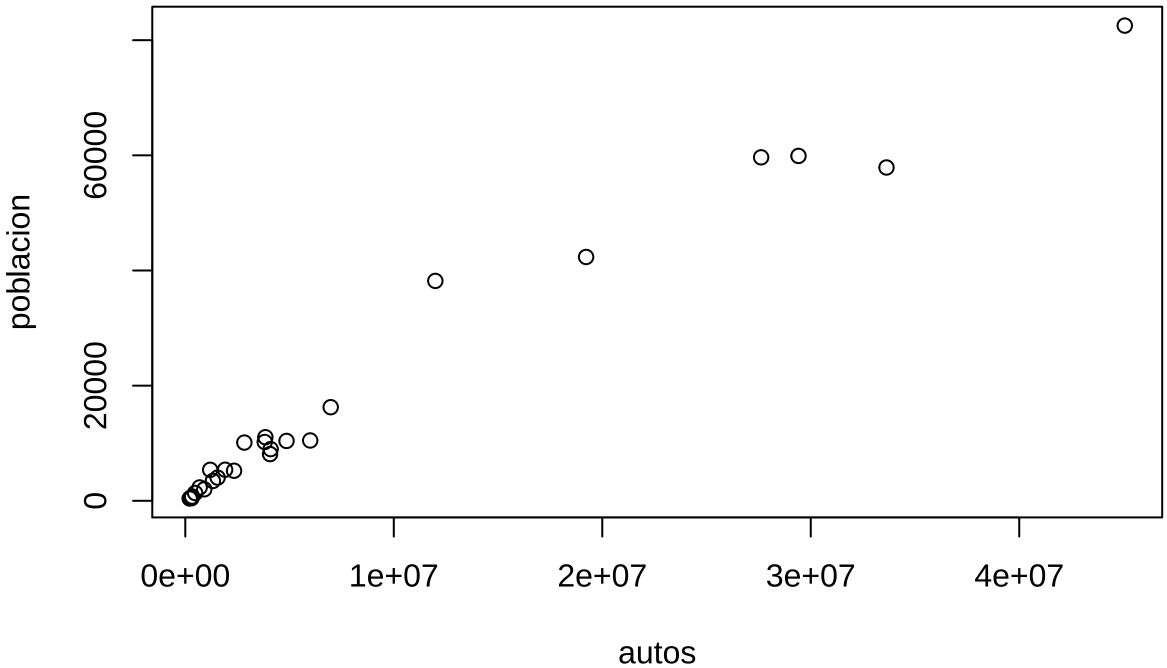

```r
#d)
#El país con la mayor tasa de mortandad es
levels(country)[which.max(tasa_muerte)]
```

```
## [1] "Cyprus"
```

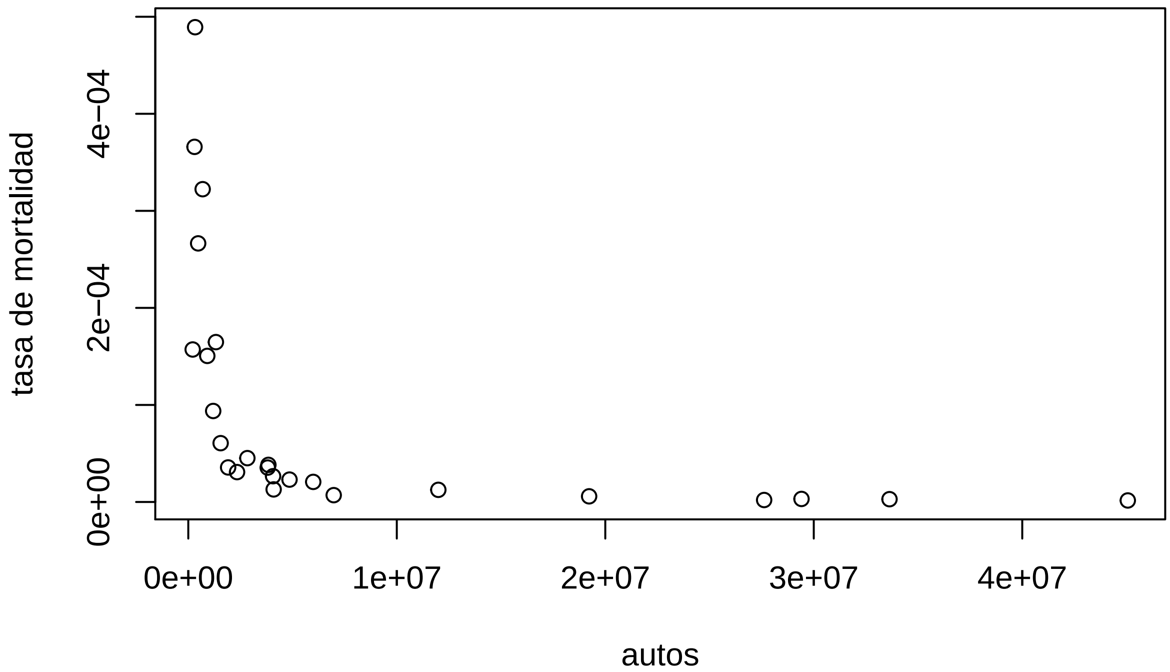
```r
#El país con la menor tasa de mortandad es
levels(country)[which.min(tasa_muerte)]
```

```
## [1] "Germany"
```

```r
#e)
total_autos=c()
for (i in population)
{total_autos=append(total_autos,car[population==i,2]*car[population==i,4])}
plot(total_autos,population,xlab='autos',ylab='poblacion')
```




```r
#f)
total_autos=c()
tasa_muerte=c()
for (i in population)
{total_autos=append(total_autos,car[population==i,2]*car[population==i,4])
tasa_muerte=append(tasa_muerte,car[population==i,3]/(car[population==i,2]*car[population==i,4]))}
plot(total_autos,tasa_muerte,xlab='autos',ylab='tasa de mortalidad')
```


tasa de mortalidad
0e+00
2e−04
4e−04
0e+00
1e+07
2e+07
3e+07
4e+07
autos