Práctica 0. FMAD 2021-2022

ICAI. Master en Big Data. Fundamentos Matemáticos del Análisis de Datos (FMAD).

García Vázquez, Carlos

Curso 2021-22. Última actualización: 2021-09-14

Preliminares

Cargamos en memoria los paquetes que van a ser necesarios durante la ejecución de la práctica.

```
library(tidyverse)
library(haven)
library(dplyr)
library(ggplot2)
```

Ejercicio 0 (ejemplo).

Enunciado: Usa la función `seq` de R para fabricar un vector `v` con los múltiplos de 3 del 0 al 300. Muestra los primeros 20 elementos de `v` usando `head` y calcula:

- la suma del vector `v`,
- su media,
- y su longitud.

Respuesta:

```
v = seq(from = 0, to = 300, by = 3)
head(v, 20)
```

```
##  [1]  0  3  6  9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57
```

Suma de `v`

```
sum(v)
```

```
## [1] 15150
```

Media:

```
mean(v)
```

```
## [1] 150
```

Longitud:

```
length(v)
```

```
## [1] 101
```

EJERCICIOS (pág 12-13)

Ejercicio 1.

Enunciado: Usando la función sample crea un vector dado_honesto con 100 números del 1 al 6. Haz una tabla de frecuencias absolutas (de dos maneras, con table y dplyr) y una tabla de frecuencias relativas.

Respuesta:

Creamos el vector dado_honesto:

```
set.seed(2021)
(dado_honesto=sample(1:6,size=100,replace=TRUE))
```

```
##   [1] 6 6 2 4 4 6 6 3 6 6 5 1 4 3 4 2 3 4 5 3 6 2 4 5 6 2 3 4 5 6 5 1 6 2 3 3 2
##  [38] 6 6 6 2 5 6 3 2 1 1 6 5 4 4 6 3 3 2 1 2 1 1 1 1 5 3 3 1 4 6 6 6 2 1 3 4 2
##  [75] 5 2 6 6 4 6 6 3 4 5 1 6 5 3 1 5 3 1 3 6 4 6 6 5 1 3
```

Tabla de frecuencias absolutas (table):

```
table(dado_honesto)
```

```
## dado_honesto
##  1  2  3  4  5  6
## 15 13 18 14 13 27
```

Tabla de frecuencias relativas (table):

```
prop.table(table(dado_honesto))
```

```
## dado_honesto
##    1    2    3    4    5    6
## 0.15 0.13 0.18 0.14 0.13 0.27
```

Convertimos el vector en un DataFrame para hacer uso del paquete dplyr

```
data_dado_honesto=data.frame(dado_honesto)
```

Tabla de frecuencias absolutas (dplyr):

```
data_dado_honesto %>%
  count(dado_honesto)
```

```
##   dado_honesto  n
## 1            1 15
## 2            2 13
## 3            3 18
## 4            4 14
## 5            5 13
## 6            6 27
```

Tabla de frecuencias relativas (dplyr):

```
data_dado_honesto %>%
  count(dado_honesto) %>%
    mutate(dado_honesto, relFreq = prop.table(n), n=NULL)
```

```
##   dado_honesto relFreq
## 1            1    0.15
## 2            2    0.13
## 3            3    0.18
## 4            4    0.14
## 5            5    0.13
## 6            6    0.27
```

Ejercicio 2.

Enunciado: A continuación crea un nuevo vector dado_cargado de manera que la probabilidad de que el número elegido valga 6 sea el doble que la probabilidad de elegir cualquiera de los cinco números restantes. Lee la ayuda de sample si lo necesitas. De nuevo, haz tablas de frecuencias absolutas y relativas de este segundo vector.

Respuesta:

Creamos el vector dado_cargado con las características definidas:

```
set.seed(2021)
(dado_cargado=sample(1:6,size=100,replace=TRUE,prob=c(rep(1/7,5),2/7)))
```

```
##   [1] 4 2 5 3 5 5 5 6 2 1 6 2 5 4 2 6 4 1 1 4 6 1 3 6 5 1 1 1 1 2 6 1 1 1 4 6 1
##  [38] 6 1 4 6 2 3 1 6 2 1 6 6 2 4 5 6 4 3 6 4 3 1 2 6 4 6 3 2 6 6 5 6 6 6 5 6 1
##  [75] 6 4 4 3 5 1 4 3 1 5 3 6 1 2 4 3 6 5 1 4 6 1 5 2 2 5
```

Tabla de frecuencias absolutas (table):

```
table(dado_cargado)
```

```
## dado_cargado
##  1  2  3  4  5  6
## 22 13 10 15 14 26
```

Tabla de frecuencias relativas (table):

```
prop.table(table(dado_cargado))
```

```
## dado_cargado
##    1    2    3    4    5    6
## 0.22 0.13 0.10 0.15 0.14 0.26
```

Convertimos el vector en un DataFrame para hacer uso del paquete dplyr

```
data_dado_cargado=data.frame(dado_cargado)
```

Tabla de frecuencias absolutas (dplyr):

```
data_dado_cargado %>%
  count(dado_cargado)
```

```
##   dado_cargado  n
## 1            1 22
## 2            2 13
## 3            3 10
## 4            4 15
## 5            5 14
## 6            6 26
```

Tabla de frecuencias relativas (dplyr):

```
data_dado_cargado %>%
  count(dado_cargado) %>%
    mutate(dado_cargado, relFreq = prop.table(n), n=NULL)
```

```
##   dado_cargado relFreq
## 1            1    0.22
## 2            2    0.13
## 3            3    0.10
## 4            4    0.15
## 5            5    0.14
## 6            6    0.26
```

Ejercicio 3.

Enunciado: Utiliza las funciones rep y seq para crear tres vectores v1, v2 y v3 con estos elementos respectivamente:

4, 4, 4, 4, 3, 3, 3, 3, 2, 2, 2, 2, 1, 1, 1, 1

1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5

1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4

Respuesta:

Creación de vectores (1ª Forma)

```
(v1= rep(seq(4,1),each = 4))
```

```
##  [1] 4 4 4 4 3 3 3 3 2 2 2 2 1 1 1 1
```

```
(v2= rep(seq(1,5),times=1:5))
```

```
##  [1] 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5
```

```
(v3=rep(seq(1,4),times=4))
```

```
##  [1] 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
```

Creación de vectores (2ª Forma)

```
(v1= rep(4:1, times = 1, length.out = NA, each = 4))
```

```
##  [1] 4 4 4 4 3 3 3 3 2 2 2 2 1 1 1 1
```

```
(v2= rep(1:5,1:5))
```

```
##  [1] 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5
```

```
(v3=rep(1:4,4))
```

```
##  [1] 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4
```

Ejercicio 4.

Enunciado: Utilizando la tabla mpg de la librería tidyverse crea una tabla mpg2 que:

- contenga las filas en las que la variable class toma el valor pickup.
- y las columnas de la tabla original cuyos nombres empiezan por c. No se trata de que las selecciones a mano, por sus nombres. Busca información sobre funciones auxiliares para select en la Sección 5.4 de R4DS.

Respuesta:

```
mpg %>%
  select(starts_with("c")) %>%
  filter(class=='pickup')
```

```
## # A tibble: 33 x 3
##      cyl   cty class
##    <int> <int> <chr>
##  1     6    15 pickup
##  2     6    14 pickup
##  3     6    13 pickup
##  4     6    14 pickup
##  5     8    14 pickup
##  6     8    14 pickup
##  7     8     9 pickup
##  8     8    11 pickup
##  9     8    11 pickup
## 10     8    12 pickup
## # ... with 23 more rows
```

Ejercicio 5.

Enunciado: Descarga el fichero census.dta. Averigua de qué tipo de fichero se trata y usa la herramienta Import DataSet del panel Environment de RStudio para leer con R los datos de ese fichero. Asegúrate de copiar en esta práctica los dos primeros comandos que llevan a cabo la importación (excluye el comando View) y que descubrirás al usar esa herramienta. Después completa los siguientes apartados con esos datos y usando dplyr y ggplot:

Respuesta:

Leemos fichero de tipo stata:

```
#Volvemos a cargar la librería para mostrar los 2 comandos necesarios en la importación juntos
library(haven)
(census = read_dta("./data/census.dta"))
```

```
## # A tibble: 50 x 12
##    state       region    pop poplt5 pop5_17 pop18p pop65p popurban medage  death
##    <chr>       <dbl+l>  <dbl>  <dbl>   <dbl>  <dbl>  <dbl>    <dbl>  <dbl>  <dbl>
##  1 Alabama     3 [Sou~ 3.89e6 2.96e5  865836 2.73e6 4.40e5  2337713   29.3  35305
##  2 Alaska      4 [Wes~ 4.02e5 3.89e4   91796 2.71e5 1.15e4   258567   26.1   1604
##  3 Arizona     4 [Wes~ 2.72e6 2.14e5  577604 1.93e6 3.07e5  2278728   29.2  21226
##  4 Arkansas    3 [Sou~ 2.29e6 1.76e5  495782 1.62e6 3.12e5  1179556   30.6  22676
##  5 California  4 [Wes~ 2.37e7 1.71e6 4680558 1.73e7 2.41e6 21607606   29.9 186428
##  6 Colorado    4 [Wes~ 2.89e6 2.16e5  592318 2.08e6 2.47e5  2329869   28.6  18925
##  7 Connecticut 1 [NE]  3.11e6 1.85e5  637731 2.28e6 3.65e5  2449774   32    26005
##  8 Delaware    3 [Sou~ 5.94e5 4.12e4  125444 4.28e5 5.92e4   419819   29.8   5123
##  9 Florida     3 [Sou~ 9.75e6 5.70e5 1789412 7.39e6 1.69e6  8212385   34.7 104190
## 10 Georgia     3 [Sou~ 5.46e6 4.15e5 1231195 3.82e6 5.17e5  3409081   28.7  44230
## # ... with 40 more rows, and 2 more variables: marriage <dbl>, divorce <dbl>
```

Observamos las columnas que forman el DataFrame:

```
names(census)
```

```
##  [1] "state"    "region"   "pop"      "poplt5"   "pop5_17"  "pop18p"
##  [7] "pop65p"   "popurban" "medage"   "death"    "marriage" "divorce"
```

Enunciado: ¿Cuáles son las poblaciones totales de las regiones censales?

Respuesta:

```
(PopReg=census %>%
  group_by(region) %>%
  summarise(PopTot=sum(pop)))
```

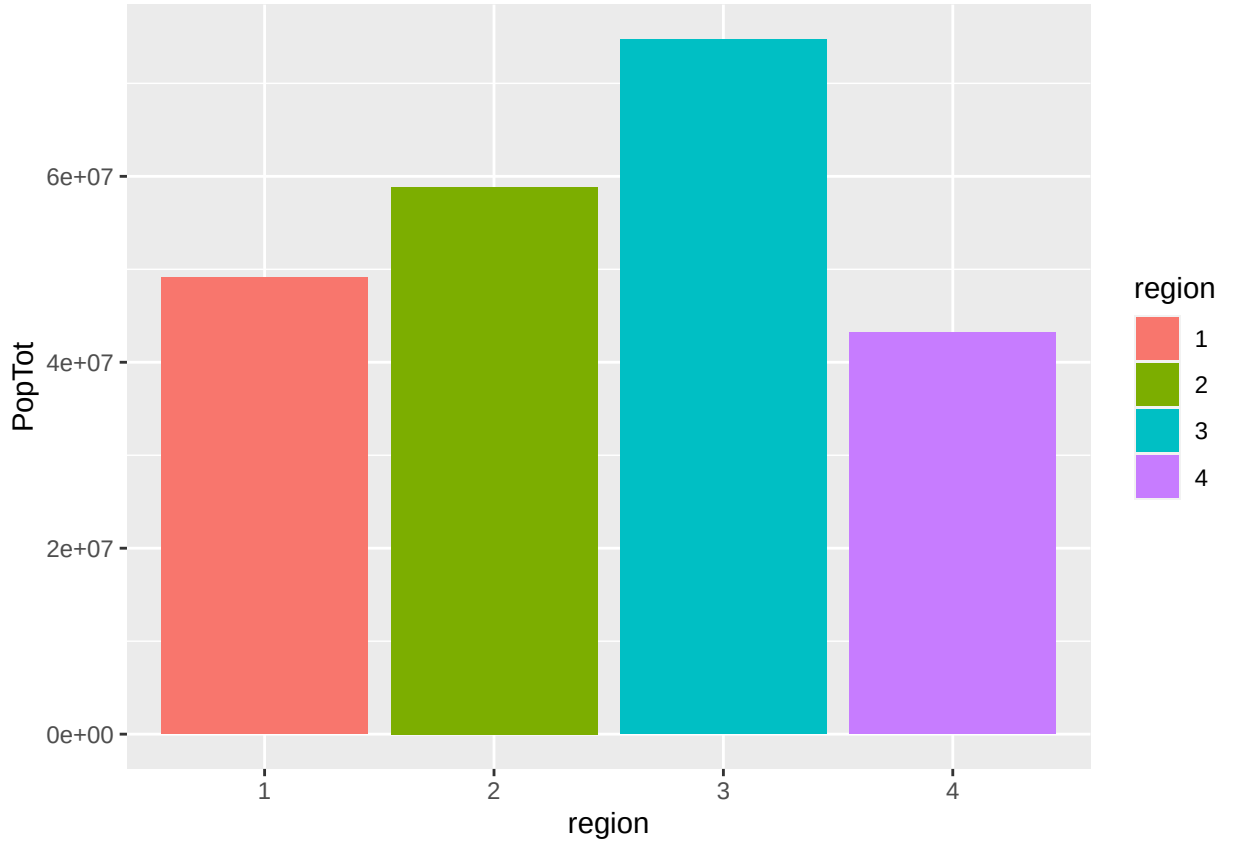
```
## # A tibble: 4 x 2
##        region   PopTot
##     <dbl+lbl>    <dbl>
## 1 1 [NE]      49135283
## 2 2 [N Cntrl] 58865670
## 3 3 [South]   74734029
## 4 4 [West]    43172490
```

Enunciado: Representa esas poblaciones totales en un diagrama de barras (una barra por región censal).

Respuesta:

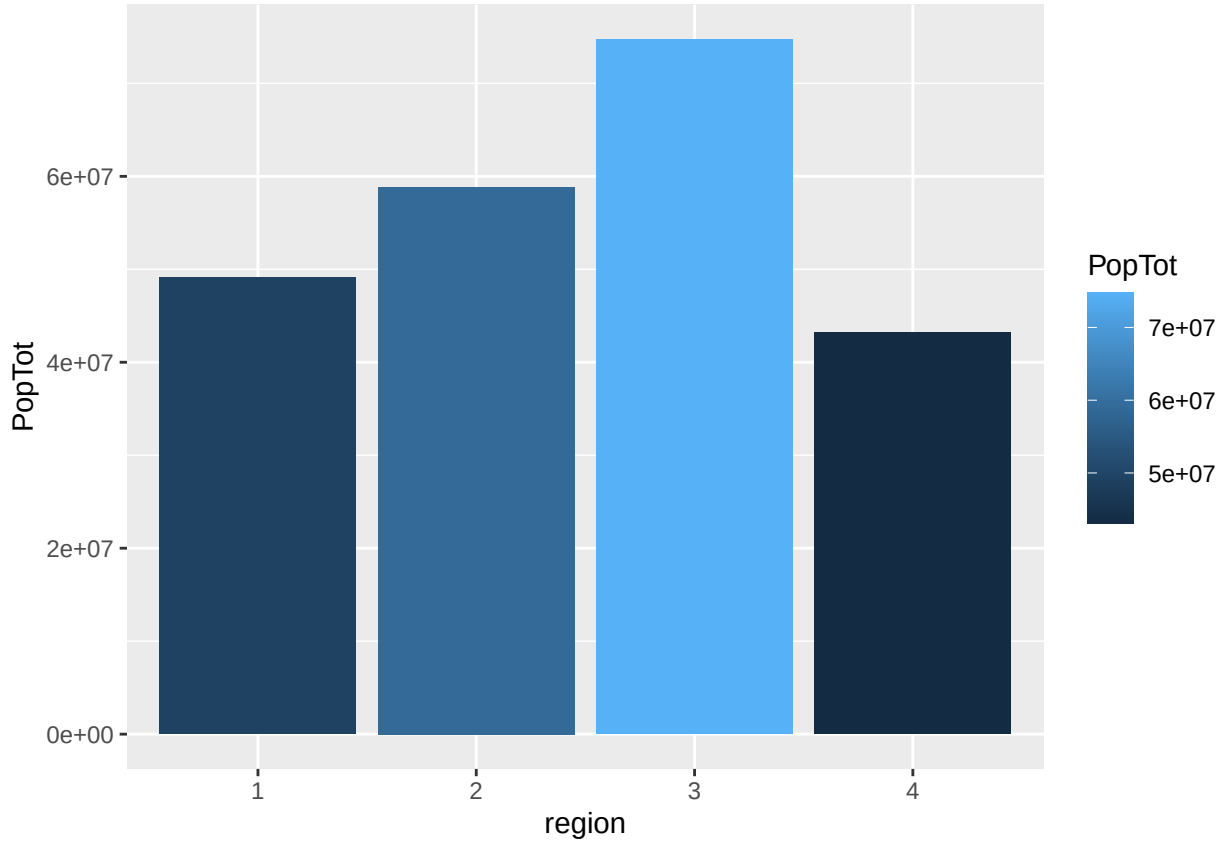
Primero realizamos la gráfica haciendo uso del geom_bar:

```
PopReg$region <- as.factor(PopReg$region)
ggplot(PopReg) +
  geom_bar(aes(x=region, y=PopTot, fill=region),stat = "identity")
```



Como añadido, mostramos la gráfica haciendo uso del geom_col, estableciendo el color en función de la población total en cada región:

```
ggplot(PopReg,aes(x=region,y=PopTot)) +
  geom_col(aes(fill=PopTot))
```



Enunciado: Ordena los estados por población, de mayor a menor

Respuesta:

Lo hacemos con la función arrange de la librería dplyr:

```
census %>%
  arrange(desc(pop))
```

```
## # A tibble: 50 x 12
##    state       region    pop poplt5 pop5_17 pop18p pop65p popurban medage  death
##    <chr>    <dbl+lbl>  <dbl>  <dbl>   <dbl>  <dbl>  <dbl>    <dbl>  <dbl>  <dbl>
##  1 Califor~ 4 [West]  2.37e7 1.71e6 4680558 1.73e7 2.41e6 21607606   29.9 186428
##  2 New York 1 [NE]    1.76e7 1.14e6 3551938 1.29e7 2.16e6 14858068   31.9 171769
##  3 Texas    3 [South] 1.42e7 1.17e6 3137045 9.92e6 1.37e6 11333017   28.2 108019
##  4 Pennsyl~ 1 [NE]    1.19e7 7.47e5 2375838 8.74e6 1.53e6  8220851   32.1 123261
##  5 Illinois 2 [N Cnt~ 1.14e7 8.42e5 2400796 8.18e6 1.26e6  9518039   29.9 102230
##  6 Ohio     2 [N Cnt~ 1.08e7 7.87e5 2307170 7.70e6 1.17e6  7918259   29.9  98268
##  7 Florida  3 [South] 9.75e6 5.70e5 1789412 7.39e6 1.69e6  8212385   34.7 104190
##  8 Michigan 2 [N Cnt~ 9.26e6 6.85e5 2066873 6.51e6 9.12e5  6551551   28.8  75102
##  9 New Jer~ 1 [NE]    7.36e6 4.63e5 1527572 5.37e6 8.60e5  6557377   32.2  68762
## 10 N. Caro~ 3 [South] 5.88e6 4.04e5 1253659 4.22e6 6.03e5  2822852   29.6  48426
## # ... with 40 more rows, and 2 more variables: marriage <dbl>, divorce <dbl>
```

Enunciado: Crea una nueva variable que contenga la tasa de divorcios /matrimonios para cada estado.

Respuesta:

Incluimos el campo en la tabla y lo mostramos:

```
census=census %>%
    mutate(divorce_rate=signif(divorce/marriage,3))
#Para que se aprecie bien, mostraré cada estado con su correspondiente tasa de divorcio
census %>%
    select(state,divorce_rate)
```

```
## # A tibble: 50 x 2
##    state       divorce_rate
##    <chr>              <dbl>
##  1 Alabama            0.546
##  2 Alaska             0.656
##  3 Arizona            0.659
##  4 Arkansas           0.599
##  5 California         0.633
##  6 Colorado           0.532
##  7 Connecticut        0.518
##  8 Delaware           0.521
##  9 Florida            0.661
## 10 Georgia            0.492
## # ... with 40 more rows
```

Enunciado: Si nos preguntamos cuáles son los estados más envejecidos podemos responder de dos maneras. Mirando la edad mediana o mirando en qué estados la franja de mayor edad representa una proporción más alta de la población total. Haz una tabla en la que aparezcan los valores de estos dos criterios, ordenada según la edad mediana decreciente y muestra los 10 primeros estados de esa tabla.

Respuesta:

Creamos la nueva columna con la proporción y seleccionamos las variables de interés, para luego ordenar los registros y mostrar únicamente los 10 primeros:

```
census %>%
  mutate(PropMayAge=signif(pop65p/pop,3)) %>%
  select(state,medage,PropMayAge) %>%
  arrange(desc(medage)) %>%
  head(10)
```

```
## # A tibble: 10 x 3
##    state         medage PropMayAge
##    <chr>          <dbl>      <dbl>
##  1 Florida         34.7      0.173
##  2 New Jersey      32.2      0.117
##  3 Pennsylvania    32.1      0.129
##  4 Connecticut     32        0.117
##  5 New York        31.9      0.123
##  6 Rhode Island    31.8      0.134
##  7 Massachusetts   31.2      0.127
##  8 Missouri        30.9      0.132
##  9 Arkansas        30.6      0.137
## 10 Maine           30.4      0.125
```

Enunciado: Haz un histograma (con 10 intervalos) de los valores de la variable medage (edad mediana) y con la curva de densidad de la variable superpuesta.

Respuesta:

Marcamos los cortes para los intervalos:

```
cortes = seq(min(census$medage,na.rm=TRUE), max(census$medage,na.rm=TRUE), length.out = 11)
```

Gráficamos los datos mediante el histograma:

```
ggplot(census, aes(x = medage)) +
  geom_histogram(aes(y=stat(density)),
                 breaks = cortes, fill = "orange", color="black")  +
  geom_density(color="red", size=1.5)
```