Práctica 0. FMAD 2021-2022

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

Rodríguez González, Álvaro

Curso 2021-22. Última actualización: 2021-09-11

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.

Solución:

Primero crearemos el vector de dado_honesto y veremos como es:

```
dado_honesto = sample(1:6,100,replace = TRUE)
dado_honesto
```

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

Tras crear el vector calcularemos las frecuencias absolutas con la función **table**:

```
table(dado_honesto)
```

```
## dado_honesto
##  1  2  3  4  5  6
## 21 16 10 18 15 20
```

También veremos las frecuencias relativas (con 3 decimales) con la función **prop.table**:

```
signif(prop.table(table(dado_honesto)),2)
```

```
## dado_honesto
##    1    2    3    4    5    6
## 0.21 0.16 0.10 0.18 0.15 0.20
```

Por último volvemos a calcular las frecuencias absolutas usando **dplyr**:

```
datos <- data.frame(dado_honesto)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()    masks stats::lag()
```

```
datos %>%
  count(datos$dado_honesto)
```

```
##   datos$dado_honesto  n
## 1                  1 21
## 2                  2 16
## 3                  3 10
## 4                  4 18
## 5                  5 15
## 6                  6 20
```

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.

Solución:

Comenzamos creamos el vector dado_cargado:

```
dado_cargado = sample(1:6,100,replace = TRUE, prob = rep(c(6/7, 12/7), times = c(5, 1)))
dado_cargado
```

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

Repetimos el mismo proceso que en el ejercicio anterior (primero con table tanto frecuencia absoluta como relativa y luego con dplyr unicamente la frecuencia absoluta):

```
table(dado_cargado)
```

```
## dado_cargado
##  1  2  3  4  5  6
## 13 14 13 17 17 26
```

```
signif(prop.table(table(dado_cargado)),2)
```

```
## dado_cargado
##    1    2    3    4    5    6
## 0.13 0.14 0.13 0.17 0.17 0.26
```

```
datos2 <- data.frame(dado_cargado)
library(tidyverse)
datos2 %>%
  count(datos2$dado_cargado)
```

```
##   datos2$dado_cargado  n
## 1                   1 13
## 2                   2 14
## 3                   3 13
## 4                   4 17
## 5                   5 17
## 6                   6 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

Solución:

Primero definiremos los vectores y luego comprobaremos que lo hemos hecho bien viendolos impresos en la pantalla.

```
v1 <- rep(seq(4,1),each = 4)
v2 <- rep(seq(1,5), c(1,2,3,4,5))
v3 <- rep(seq(1,4),4)
```

```
v1
```

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

```
v2
```

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

```
v3
```

```
##  [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:

Hago la selección:

```
mpg2 <- mpg %>%
  filter(class =='pickup') %>%
    select(starts_with('c'))
mpg2
```

```
## # 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:

- ¿Cuáles son las poblaciones totales de las regiones censales?
- Representa esas poblaciones totales en un diagrama de barras (una barra por región censal).
- Ordena los estados por población, de mayor a menor.
- Crea una nueva variable que contenga la tasa de divorcios /matrimonios para cada estado.
- 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.
- 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.

Solución:

Primero importamos los datos con los que vamos a trabajar y los echamos un vistazo para conocer como están estructurados:

```
library(haven)
census <- read_dta("data/census.dta")
```

```
census
```

```
## # 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>
```

Lo primero que queremos ver son las poblaciones totales de las regiones censales, es decir, la frecuencia absoluta de cada región:

```
pob_reg <- census %>%
  group_by(region) %>%
  summarise(Poblacion = sum(pop))

pob_reg
```

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

Otra forma de hacer lo que hemos hecho antes sin usar dplyr seria con el siguiente código:

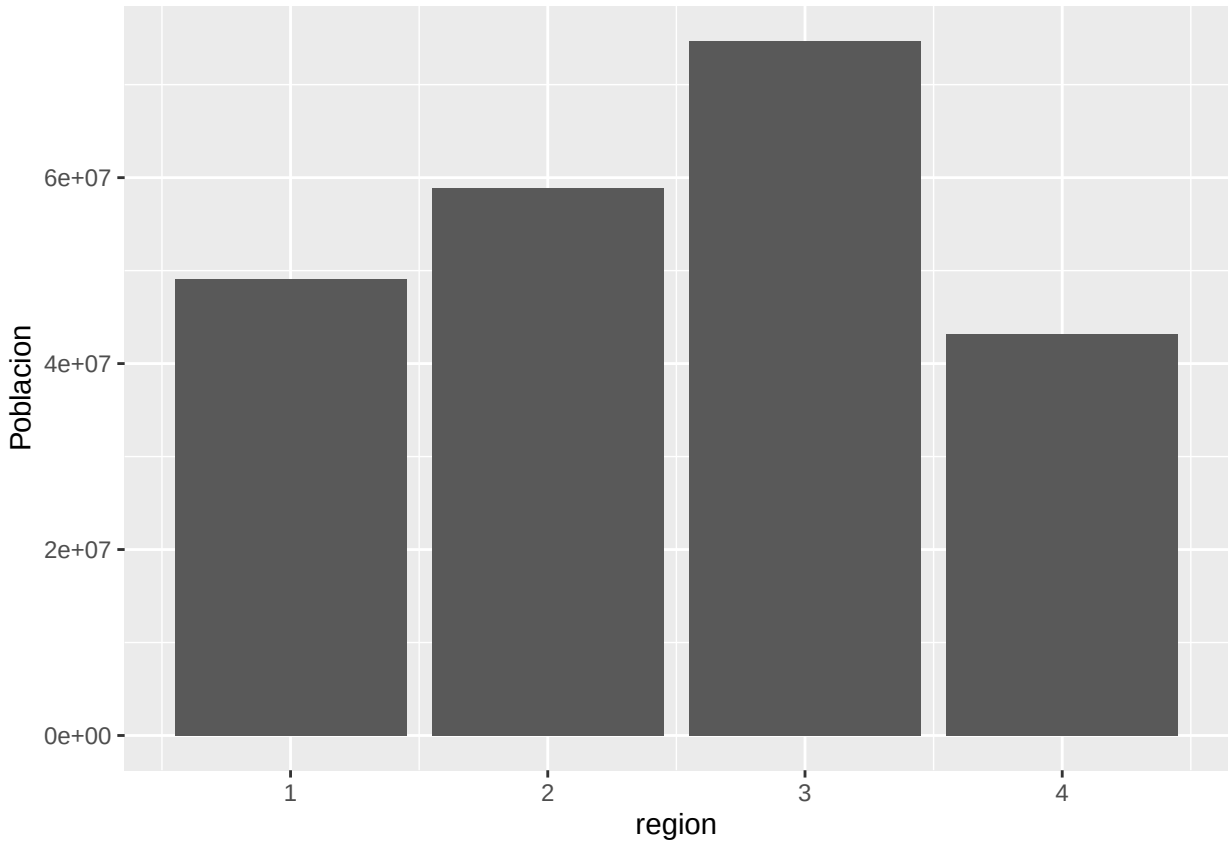
```
aggregate(census$pop, by=list(Category=census$region), FUN=sum)
```

```
##   Category        x
## 1        1 49135283
## 2        2 58865670
## 3        3 74734029
## 4        4 43172490
```

Ahora vamos a representar las poblaciones totales en un diagrama de barras:

```
ggplot(pob_reg, aes(x = region, y = Poblacion)) + geom_bar(stat="identity", position="stack")
```

```
## Don't know how to automatically pick scale for object of type haven_labelled/vctrs_vctr/double. Defau
```



Ahora vamos a ordenar los estados por población, de mayor a menor:

```
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>
```

A continuación crearemos una nueva variable que contendrá la tasa de divorcios /matrimonios para cada estado:

```
ratiodivcas <- census %>%
  mutate(state, ratio = divorce/marriage) %>%
  select(state, ratio)
ratiodivcas
```

```
## # A tibble: 50 x 2
##    state       ratio
##    <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
```

Ahora imprimiremos una tabla donde aparecerán los estados, la edad media de cada uno de ellos y el porcentaje que representa la población mayor de 65 años sobre el total. Además, ordenaremos de mayor a menor según la variable edad media (o medage que es como aparece en la tabla):

```
census %>%
  mutate(state, ratiopob65 = pop65p/pop) %>%
  select(state, medage, ratiopob65) %>%
  top_n(n = 10,medage) %>%
  arrange(desc(medage))
```

```
## # A tibble: 11 x 3
##    state         medage ratiopob65
##    <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
## 11 W. Virginia     30.4      0.122
```

Por último haremos un histograma (con 10 intervalos) de los valores de la variable medage y con la curva de densidad de la variable superpuesta:

```
ggplot(data = census, aes(x = medage)) + geom_histogram(aes(y=stat(density)), bins = 10, fill = "skyblu
```