Práctica 0. FMAD 2021-2022

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

Suero, Jesús

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

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
```

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.

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

Tabla frecuencias absolutas con table:

```
table(dado_honesto)
```

```
## dado_honesto
##  1  2  3  4  5  6
## 19 17 19 15 13 17
```

Tabla frecuencias absolutas con dplyr:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
datos <-
  data.frame(c(1:length(dado_honesto)),dado_honesto)
datos %>%
  count(dado_honesto)
```

```
##   dado_honesto  n
## 1            1 19
## 2            2 17
## 3            3 19
## 4            4 15
## 5            5 13
## 6            6 17
```

Tabla frecuencias relativas:

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

```
## dado_honesto
##    1    2    3    4    5    6
## 0.19 0.17 0.19 0.15 0.13 0.17
```

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.

```
dado_cargado = sample(1:6, size = 100, replace = TRUE, prob = c(1/7,1/7,1/7,1/7,1/7,2/7))
```

Tabla frecuencias absolutas con table:

```
table(dado_cargado)
```

```
## dado_cargado
##  1  2  3  4  5  6
## 16 13 18 13  8 32
```

Tabla frecuencias relativas:

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

```
## dado_cargado
##    1    2    3    4    5    6
## 0.16 0.13 0.18 0.13 0.08 0.32
```

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

```
(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
```

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.

```
library(tidyverse)
```

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

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

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

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

```
## # 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:

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

¿Cuáles son las poblaciones totales de las regiones censales?

```
census %>%
  group_by(region) %>%
  summarize(sum(pop))
```

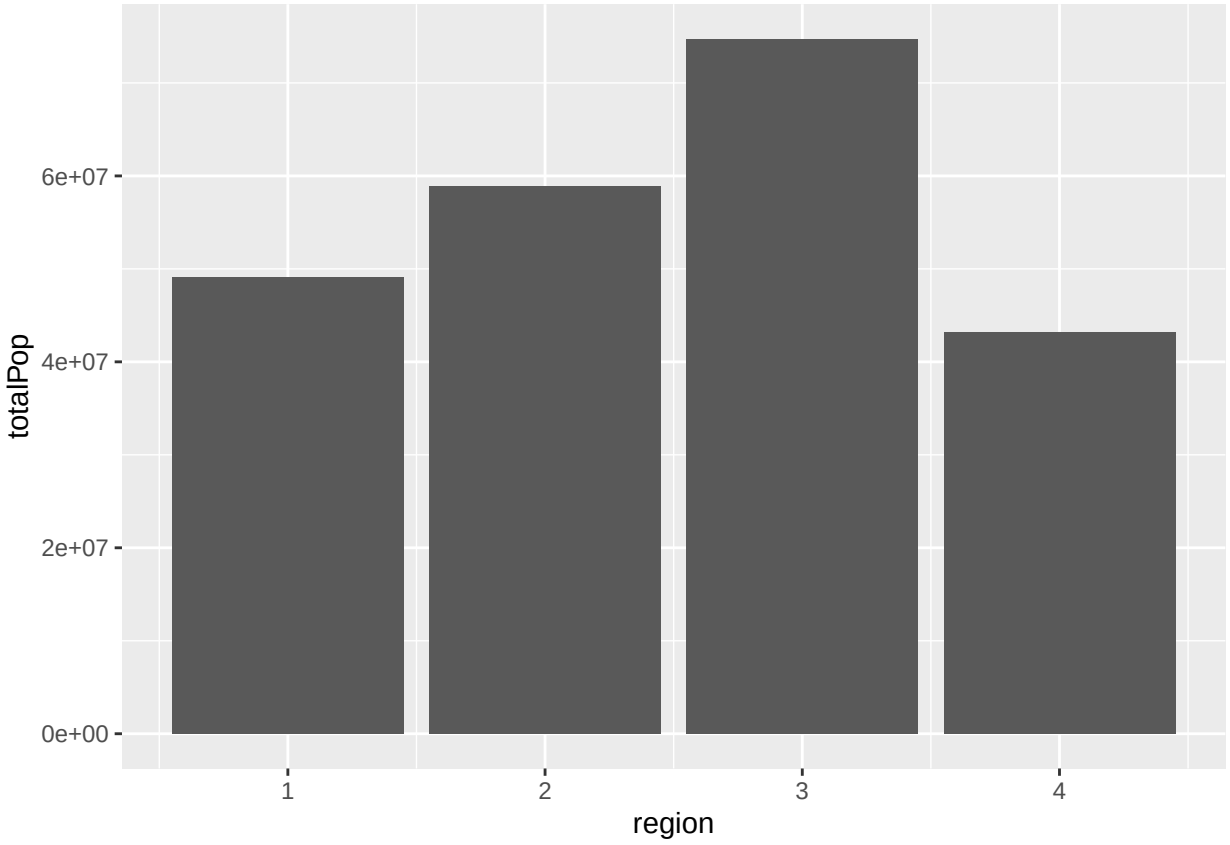
```
## # A tibble: 4 x 2
##        region `sum(pop)`
##     <dbl+lbl>      <dbl>
## 1 1 [NE]        49135283
## 2 2 [N Cntrl]   58865670
## 3 3 [South]     74734029
## 4 4 [West]      43172490
```

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

```
data = census %>%
  group_by(region) %>%
  summarize(totalPop=sum(pop))

ggplot(data, aes(x = region, y=totalPop)) +
  geom_bar(stat="identity")
```

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



Ordena los estados por población, de mayor a menor.

```
census %>%
  select(state,pop) %>%
  arrange(desc(pop))
```

```
## # A tibble: 50 x 2
##    state             pop
##    <chr>           <dbl>
##  1 California   23667902
##  2 New York     17558072
##  3 Texas        14229191
##  4 Pennsylvania 11863895
##  5 Illinois     11426518
##  6 Ohio         10797630
##  7 Florida       9746324
##  8 Michigan      9262078
##  9 New Jersey    7364823
## 10 N. Carolina   5881766
## # ... with 40 more rows
```

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

```
census %>%
  mutate(divorcios_matrimonios = divorce / marriage) %>%
  select(state,divorcios_matrimonios)
```

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

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.

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

```
## # A tibble: 10 x 3
##    state         propMayores medage
##    <chr>               <dbl>  <dbl>
##  1 Florida             0.173   34.7
##  2 New Jersey          0.117   32.2
```

```
##  3 Pennsylvania        0.129   32.1
##  4 Connecticut         0.117   32
##  5 New York            0.123   31.9
##  6 Rhode Island        0.134   31.8
##  7 Massachusetts       0.127   31.2
##  8 Missouri            0.132   30.9
##  9 Arkansas            0.137   30.6
## 10 Maine               0.125   30.4
```

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.

```
ggplot(census, aes(x = medage)) +
  geom_histogram(aes(y=stat(density)), bins=10, fill = "orange", color="black")  +
  geom_density(color="red", size=1.5)
```