Práctica 0. FMAD 2021-2022

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

Monsalve Rodilla, Ignacio

Curso 2021-22. Última actualización: 2021-09-15

Contenido

Ejercicio N

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 I

Enunciado: Usa la función sample para crear 100 números del 1 al 6. Haz la tabla de frecuencias absolutas y relativas

Respuesta:

En primer lugar, definimos el vector dado_honesto

```
set.seed(111)
dado_honesto <-  sample(1:6, 100, replace=TRUE)
dado_honesto
```

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

A continuación, se realiza la tabla de frecuencias absolutas y relativas. En primer lugar se utilizará R normal:

```
table(dado_honesto)
```

```
## dado_honesto
##  1  2  3  4  5  6
## 17 15 14 21 20 13
```

Tabla de frecuencias relativas

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

```
## dado_honesto
##    1    2    3    4    5    6
## 0.17 0.15 0.14 0.21 0.20 0.13
```

Utilizando dplyr:

```
library(tidyverse)

dat = data.frame(dado_honesto)

dat_ej_1 = data.frame(dado_honesto)
```

Se utiliza ‘count’

Frecuencia absoluta

```
dat_ej_1 %>%
  count(dado_honesto)
```

```
##   dado_honesto  n
## 1            1 17
## 2            2 15
## 3            3 14
## 4            4 21
## 5            5 20
## 6            6 13
```

Frecuencia relativa

```
dat_ej_1 %>%
  count(dado_honesto)
```

```
##   dado_honesto  n
## 1            1 17
## 2            2 15
## 3            3 14
## 4            4 21
## 5            5 20
## 6            6 13
```

```
dat_ej_1 %>%
  count(dado_honesto) %>%
  mutate(dado_honesto, relFreq = prop.table(n))
```

```
##   dado_honesto  n relFreq
## 1            1 17    0.17
## 2            2 15    0.15
## 3            3 14    0.14
## 4            4 21    0.21
## 5            5 20    0.20
## 6            6 13    0.13
```

Ejercicio II

Enunciado: Crear un vector 'dado_cargado' siendo la probabilidad de 6 doble que la de cualquier otro Realizar tablas de frecuencias absolutas y relativas.

Respuesta:

```
set.seed(111)
dado_cargado <-  sample(1:6, 100, replace = TRUE, prob = rep(c(6/7, 12/7)
, times = c(5,1)))
dado_cargado
```

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

A continuación, se muestran las tablas como en el Ejercicio I:

```
table(dado_cargado)
```

```
## dado_cargado
##  1  2  3  4  5  6
## 10 14 17 16 19 24
```

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

```
## dado_cargado
##    1    2    3    4    5    6
## 0.10 0.14 0.17 0.16 0.19 0.24
```

Se repite el proceso con dplyr

```
dat_ej_2 = data.frame(dado_cargado)



dat_ej_2 %>%
  count(dado_cargado)
```

```
##   dado_cargado  n
## 1            1 10
## 2            2 14
## 3            3 17
## 4            4 16
## 5            5 19
## 6            6 24
```

La tabla de frecuencias relativas es:

```
dat_ej_2 %>%
  count(dado_cargado) %>%
  mutate(dado_cargado, relFreq = prop.table(n))
```

```
##   dado_cargado  n relFreq
## 1            1 10    0.10
## 2            2 14    0.14
## 3            3 17    0.17
## 4            4 16    0.16
## 5            5 19    0.19
## 6            6 24    0.24
```

Ejercicio III

Enunciado: Utilizar rep y sep para crear tres vectores con una distribución concreta.

Respuesta:

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

Ejercicio IV

Enunciado: Debe utilizarse la tabla mpg y crear una tabla alternativa. En ella todas las filas deben contener 'pickup' y las columnas comenzar por 'c'.

Respuesta:

```
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 V

Enunciado: En primer lugar, se pide descargar el fichero de Census. Desde la ventana de Environment se realiza un 'Import Dataset' y se guarda en la variable census Realmente, se quiere trabajar con dplyr por lo que debe realizarse una conversión

```
library(haven)
census2 = read_dta('./census.dta')

View(census2)
```

Respuesta

- ¿Cuáles son las poblaciones totales de las regiones?

```
Conteo <- sum(census2$pop)

 census2  %>%
  group_by(region) %>%

  summarise(Conteo =  sum(pop))-> poblacion
población
```

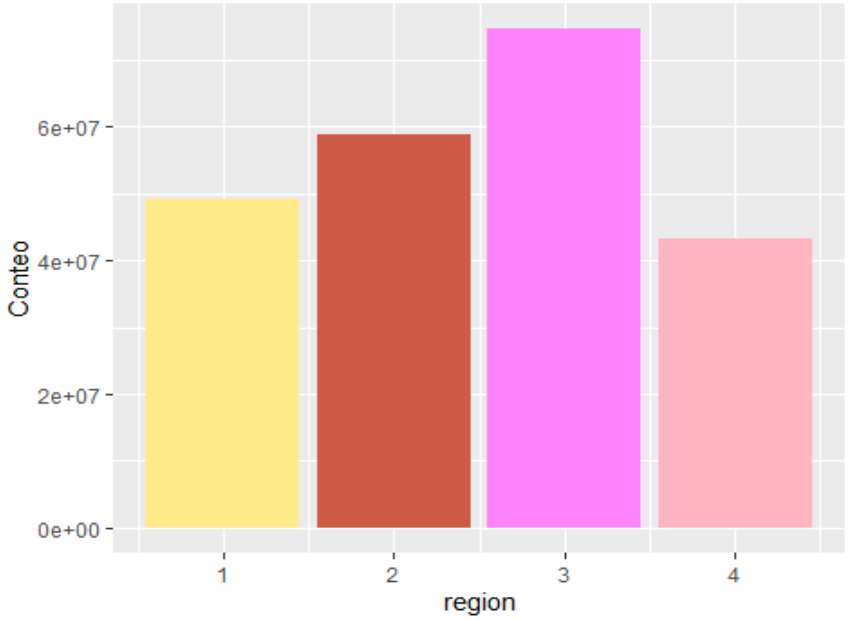
```
## # A tibble: 4 x 2
##        region   Conteo
##     <dbl+lbl>    <dbl>
## 1 1 [NE]      49135283
## 2 2 [N Cntrl] 58865670
## 3 3 [South]   74734029
## 4 4 [West]    43172490
```

- Representa las poblaciones en un diagrama de barras

```
library(viridisLite)

ggplot(poblacion, aes(region, Conteo))+ geom_col(fill=c('lightgoldenrod1'
,'coral3', 'orchid1','lightpink'))

## Don't know how to automatically pick scale for object of type haven_la
belled/vctrs_vctr/double. Defaulting to continuous.
```



- Representa las poblaciones de mayor a menor

```
census2 %>%
  arrange(desc(pop))
```

```
## # A tibble: 50 x 12
##    state       region    pop poplt5 pop5_17 pop18p pop65p popurban med
age  death
##    <chr>    <dbl+lbl>  <dbl>  <dbl>   <dbl>  <dbl>  <dbl>    <dbl>  <d
bl>  <dbl>
##  1 Califor~ 4 [West]  2.37e7 1.71e6 4680558 1.73e7 2.41e6 21607606   2
9.9 186428
##  2 New York 1 [NE]    1.76e7 1.14e6 3551938 1.29e7 2.16e6 14858068   3
1.9 171769
##  3 Texas    3 [South] 1.42e7 1.17e6 3137045 9.92e6 1.37e6 11333017   2
8.2 108019
##  4 Pennsyl~ 1 [NE]    1.19e7 7.47e5 2375838 8.74e6 1.53e6  8220851   3
2.1 123261
##  5 Illinois 2 [N Cnt~ 1.14e7 8.42e5 2400796 8.18e6 1.26e6  9518039   2
9.9 102230
##  6 Ohio     2 [N Cnt~ 1.08e7 7.87e5 2307170 7.70e6 1.17e6  7918259   2
9.9  98268
##  7 Florida  3 [South] 9.75e6 5.70e5 1789412 7.39e6 1.69e6  8212385   3
4.7 104190
##  8 Michigan 2 [N Cnt~ 9.26e6 6.85e5 2066873 6.51e6 9.12e5  6551551   2
8.8  75102
##  9 New Jer~ 1 [NE]    7.36e6 4.63e5 1527572 5.37e6 8.60e5  6557377   3
2.2  68762
## 10 N. Caro~ 3 [South] 5.88e6 4.04e5 1253659 4.22e6 6.03e5  2822852   2
```

```
9.6  48426
## # ... with 40 more rows, and 2 more variables: marriage <dbl>, divorce
<dbl>
```

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

```
variable_tasa <- census2 %>%
  mutate(state, ratio = divorce/marriage) %>%
  select(state, ratio)
variable_tasa
```

```
## # 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
```

- Si nos preguntamos cuáles son los estados más envejecidos podemos responder de dos maneras. Mirando la edad mediana o mirando la franja de mayor edad. Haz una tabla en la que aparezcan los valores de estos criterios

```
census2 %>%
  mutate(state, variable_65 = pop65p/pop) %>%
  select(state, medage, variable_65) %>%
  arrange(desc(medage))
```

```
## # A tibble: 50 x 3
##    state         medage variable_65
##    <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
## # ... with 40 more rows
```

- Haz un histograma de los valores de las variables medage y con la curva de densidad

```
ggplot(census2, aes(x = medage)) + geom_histogram(aes(y=stat(density)), b
ins = 10, fill = 'darkolivegreen1') + geom_density(color='darkolivegreen3
', size=2)
```