

Práctica 0. FMAD 2021-2022

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

Apellido, Nombre

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

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: Preliminares

```
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()
```

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

```
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:

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

```
table(dado_honesto)
```

```
## dado_honesto
##  1  2  3  4  5  6
## 27 17 17 11 14 14
```

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

```
## dado_honesto
##    1    2    3    4    5    6
## 0.27 0.17 0.17 0.11 0.14 0.14
```

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

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

```
##   dado_honesto   n
## 1              1 27
## 2              2 17
## 3              3 17
## 4              4 11
## 5              5 14
## 6              6 14
```

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

```
##   dado_honesto relFreq
## 1             1    0.27
## 2             2    0.17
## 3             3    0.17
## 4             4    0.11
## 5             5    0.14
## 6             6    0.14
```

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, 100, replace=TRUE,prob=c(rep(1/7,5),2/7))
table(dado_cargado)
```

```
## dado_cargado
## 1  2  3  4  5  6
## 12 18 16 14 17 23
```

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

```
## dado_cargado
##   1    2    3    4    5    6
## 0.12 0.18 0.16 0.14 0.17 0.23
```

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

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

```
##   dado_cargado  n
## 1             1 12
## 2             2 18
## 3             3 16
## 4             4 14
## 5             5 17
## 6             6 23
```

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

```
##   dado_cargado relFreq
## 1             1    0.12
```

```
## 2      2    0.18
## 3      3    0.16
## 4      4    0.14
## 5      5    0.17
## 6      6    0.23
```

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

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

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

```
(v1=rep(4:1, times=1, 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, times=4, each=1))
```

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

Utilizando la tabla mpg de la librería tidyverse crea una tabla mpg2 que: I contenga las filas en las que la variable class toma el valor pickup. I y las columnas de la tabla original cuyos nombres empiezan por c. No se trata de que las selecciones a mano, por sus nombres.

```
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
```

```
(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>
```

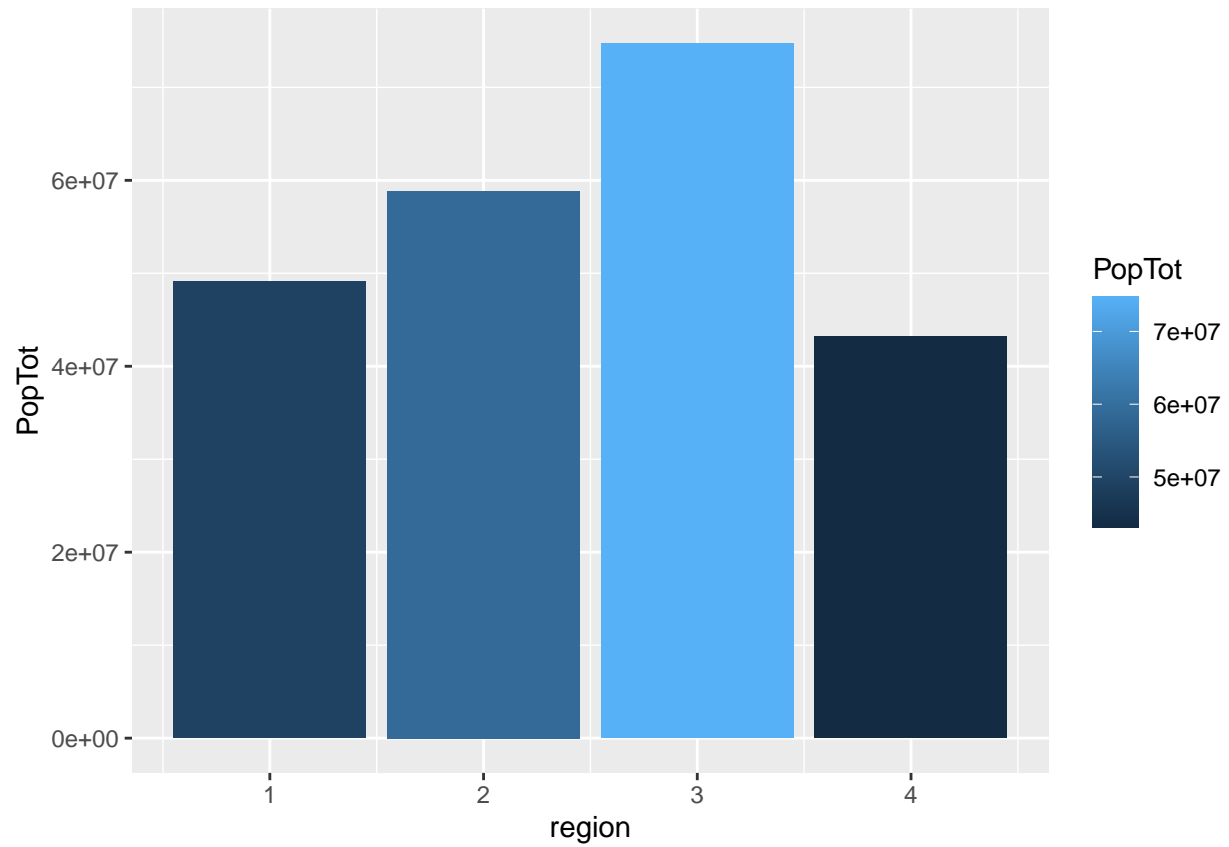
¿Cuáles son las poblaciones totales de las regiones censales?

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

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

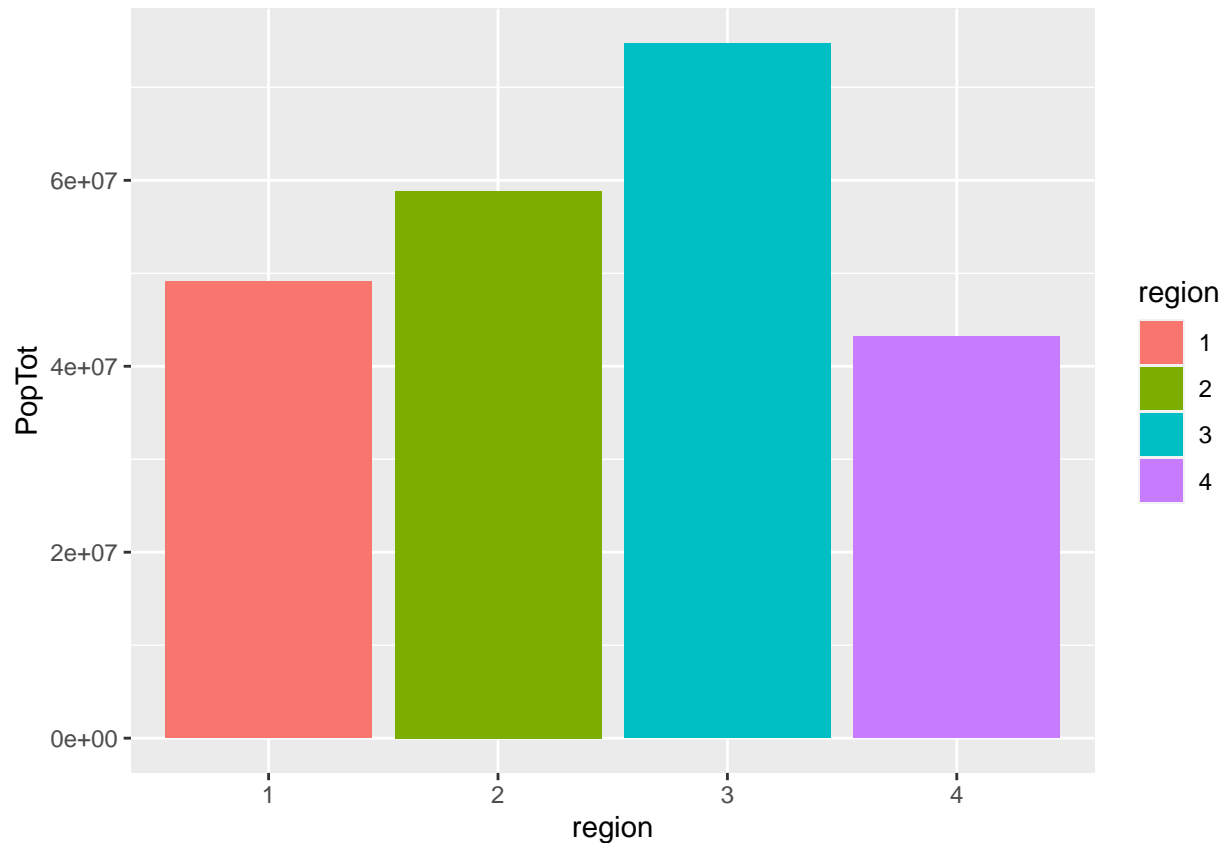
```
library(viridisLite)
ggplot(PopReg) +
  geom_col(aes(x=region,y=PopTot, fill=PopTot))
```

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



```
PopReg$region <- as.factor(PopReg$region)

ggplot(PopReg, aes(x=region, y=PopTot, fill=region)) +
  geom_bar(stat = "identity")
```



Ordena 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 Pennsylv~ 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>
```

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

```
TasaDiv=census %>%
  mutate(TasaDivorcio=divorce/marriage)
(TasaDiv)
```

```
## # A tibble: 50 x 13
##   state      region    pop poplt5 pop5_17 pop18p pop65p popurban medage  death
##   <chr>      <dbl+1> <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 3 more variables: marriage <dbl>, divorce <dbl>,
## #   TasaDivorcio <dbl>
```

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 %>%
  arrange(desc(medage)) %>%
  mutate(Proporcion65plus=pop65p/pop) %>%
  select(state, medage,Proporcion65plus)
```

```
## # A tibble: 50 x 3
##   state      medage Proporcion65plus
##   <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
```

```
head(10)
```

```
## [1] 10
```

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.

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

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