

Ejercicio-02JaimeRiquelme

JaimeRiquelme

2023-10-10

1- Instale los paquetes gtools y combinat con las sentencias: `install.packages('gtools')` y `install.packages('combinat')`. Incluya las librerías gtools (`library(gtools)`) y combinat (`library(combinat)`).

```
(library('gtools'))
```

```
## [1] "combinat" "gtools"    "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets" "methods"   "base"
```

```
(library('combinat'))
```

```
## [1] "combinat" "gtools"    "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets" "methods"   "base"
```

Permutación y Combinación

La permutación y la combinación son dos conceptos fundamentales en la teoría de la probabilidad y en la estadística. A continuación, se explicará brevemente la diferencia entre ambas y se presentarán las ecuaciones para calcular cada una de ellas.

Permutación La permutación se refiere a un arreglo ordenado de elementos en los que el orden es importante y se toman todos los elementos. Es decir, el número de permutaciones posibles de n objetos es el número de formas distintas en las que se pueden ordenar estos objetos tomando en cuenta todos los elementos. En las permutaciones sin repetición, cada elemento se puede utilizar una sola vez, mientras que en las permutaciones con repetición, los elementos pueden ser utilizados múltiples veces.

Formulas para calcular permutaciones:

El numero de permutaciones de n diferentes elementos es:

$$n! = \prod_{i=1}^n i$$

El número de permutaciones de un subconjunto de elementos seleccionados de un conjunto de n elementos diferentes es:

$$P_{r,n} = \frac{n!}{(n-r)!}$$

Permutaciones con repetición

El numero de permutaciones de $n = n_1 + n_2 + \dots + n_r$ elementos de diferentes tipos es:

$$P_n = \frac{n!}{n_1!n_2!n_3!\dots n_r!}$$

Combinación

Las combinaciones se enfocan en la selección de elementos, y se diferencian de las permutaciones en que el orden no es importante en las combinaciones. En otras palabras, el número de combinaciones posibles de r elementos seleccionados de un conjunto de n elementos es igual al número de maneras diferentes en que se pueden elegir esos r elementos sin importar el orden en que se seleccionen. En las combinaciones sin repetición, cada elemento solo se puede utilizar una vez, mientras que en las combinaciones con repetición, los elementos se pueden usar varias veces.

Formulas para calcular combinaciones:

El número de combinaciones, para un subconjunto de r elementos que pueden ser seleccionados entre n elementos es denotado como:

$$C_{n,r} = \binom{n}{r} = \frac{n!}{r!(n-r)!}$$

Para realizar una combinación con repetición se utiliza la siguiente fórmula:

$$\binom{n+r-1}{r} = \frac{(n+r-1)!}{r!(n-1)!}$$

Permutacion Circular

Una permutación circular es un tipo especial de permutación en la que el orden de los elementos se considera relativo al punto de inicio, y no hay un inicio o final fijo. Es como organizar objetos en un círculo, donde se puede comenzar a contar desde cualquier punto y aún obtener la misma disposición general.

$$PC_n = (n-1)!$$

Paquetes gtools y permn en R

Paquete ‘gtools’

Combinations: Esta función Genera combinaciones a partir de un conjunto de n elementos, eligiendo r elementos a la vez. Puede especificar si las repeticiones están permitidas y si se utiliza un conjunto diferente de números en lugar del predeterminado 1:n. Ejemplo:

```
combinations(n = 4, r = 2)
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    1    3
## [3,]    1    4
## [4,]    2    3
## [5,]    2    4
## [6,]    3    4
```

Permutations Genera permutaciones de n elementos, tomados r a la vez. A diferencia de las combinaciones, en las permutaciones el orden de los elementos es relevante. Puede especificar si las repeticiones están permitidas y si se utiliza un conjunto diferente de números. Ejemplo:

```
permutations(n = 3, r = 2)
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    1    3
## [3,]    2    1
## [4,]    2    3
## [5,]    3    1
## [6,]    3    2
```

Paquete ‘Combinat’

Permn Calcula todas las permutaciones posibles de un vector dado. Ejemplo:

```
permn(c("A", "B", "C"))
```

```
## [[1]]
## [1] "A" "B" "C"
##
## [[2]]
## [1] "A" "C" "B"
##
## [[3]]
## [1] "C" "A" "B"
##
## [[4]]
## [1] "C" "B" "A"
##
## [[5]]
## [1] "B" "C" "A"
##
## [[6]]
## [1] "B" "A" "C"
```

Combn Esta función, que en realidad es parte del paquete base de R, genera combinaciones de elementos de un vector. Permite especificar cuántos elementos se deben seleccionar a la vez y tiene la capacidad de aplicar una función opcional a cada combinación generada.

```
combn(x = 4, m = 2)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    1    1    2    2    3
## [2,]    2    3    4    3    4    4
```

Diferencias entre ‘gtools’ y ‘combinat’ 1) Las funciones en gtools son más flexibles en términos de especificar el tamaño del conjunto original, el número de elementos a seleccionar, y si se permiten repeticiones.

2) permn simplemente genera todas las permutaciones posibles de un vector.

3) combn se centra en combinaciones y ofrece la posibilidad de aplicar una función a cada combinación.

Calcular:

a) La cantidad de permutaciones posibles con $n = 9$ y $r = 4$ con y sin repetición.

Con repetición:

```
nrow(permutations(9,4, repeats.allowed = TRUE))
```

```
## [1] 6561
```

Sin repetición:

```
nrow(permutations(9,4, repeats.allowed = FALSE))
```

```
## [1] 3024
```

b) Las combinaciones de largo cuatro con las letras f, g, h, i y j con y sin repetición.

Con repetición:

```
letras <- c("f", "g", "h", "i", "j")
combinations(n = length(letras), r = 4, v = letras, repeats.allowed = TRUE)
```

```
##      [,1] [,2] [,3] [,4]
## [1,] "f"  "f"  "f"  "f"
## [2,] "f"  "f"  "f"  "g"
## [3,] "f"  "f"  "f"  "h"
## [4,] "f"  "f"  "f"  "i"
## [5,] "f"  "f"  "f"  "j"
## [6,] "f"  "f"  "g"  "g"
## [7,] "f"  "f"  "g"  "h"
## [8,] "f"  "f"  "g"  "i"
## [9,] "f"  "f"  "g"  "j"
## [10,] "f"  "f"  "h"  "h"
## [11,] "f"  "f"  "h"  "i"
## [12,] "f"  "f"  "h"  "j"
## [13,] "f"  "f"  "i"  "i"
## [14,] "f"  "f"  "i"  "j"
## [15,] "f"  "f"  "j"  "j"
## [16,] "f"  "g"  "g"  "g"
## [17,] "f"  "g"  "g"  "h"
## [18,] "f"  "g"  "g"  "i"
## [19,] "f"  "g"  "g"  "j"
## [20,] "f"  "g"  "h"  "h"
## [21,] "f"  "g"  "h"  "i"
## [22,] "f"  "g"  "h"  "j"
## [23,] "f"  "g"  "i"  "i"
## [24,] "f"  "g"  "i"  "j"
## [25,] "f"  "g"  "j"  "j"
## [26,] "f"  "h"  "h"  "h"
## [27,] "f"  "h"  "h"  "i"
## [28,] "f"  "h"  "h"  "j"
## [29,] "f"  "h"  "i"  "i"
```

```
## [30,] "f" "h" "i" "j"
## [31,] "f" "h" "j" "j"
## [32,] "f" "i" "i" "i"
## [33,] "f" "i" "i" "j"
## [34,] "f" "i" "j" "j"
## [35,] "f" "j" "j" "j"
## [36,] "g" "g" "g" "g"
## [37,] "g" "g" "g" "h"
## [38,] "g" "g" "g" "i"
## [39,] "g" "g" "g" "j"
## [40,] "g" "g" "h" "h"
## [41,] "g" "g" "h" "i"
## [42,] "g" "g" "h" "j"
## [43,] "g" "g" "i" "i"
## [44,] "g" "g" "i" "j"
## [45,] "g" "g" "j" "j"
## [46,] "g" "h" "h" "h"
## [47,] "g" "h" "h" "i"
## [48,] "g" "h" "h" "j"
## [49,] "g" "h" "i" "i"
## [50,] "g" "h" "i" "j"
## [51,] "g" "h" "j" "j"
## [52,] "g" "i" "i" "i"
## [53,] "g" "i" "i" "j"
## [54,] "g" "i" "j" "j"
## [55,] "g" "j" "j" "j"
## [56,] "h" "h" "h" "h"
## [57,] "h" "h" "h" "i"
## [58,] "h" "h" "h" "j"
## [59,] "h" "h" "i" "i"
## [60,] "h" "h" "i" "j"
## [61,] "h" "h" "j" "j"
## [62,] "h" "i" "i" "i"
## [63,] "h" "i" "i" "j"
## [64,] "h" "i" "j" "j"
## [65,] "h" "j" "j" "j"
## [66,] "i" "i" "i" "i"
## [67,] "i" "i" "i" "j"
## [68,] "i" "i" "j" "j"
## [69,] "i" "j" "j" "j"
## [70,] "j" "j" "j" "j"
```

Sin repeticion:

```
letras <- c("f", "g", "h", "i", "j")
combinations(n = length(letras), r = 4, v = letras, repeats.allowed = FALSE)
```

```
##      [,1] [,2] [,3] [,4]
## [1,] "f"  "g"  "h"  "i"
## [2,] "f"  "g"  "h"  "j"
## [3,] "f"  "g"  "i"  "j"
## [4,] "f"  "h"  "i"  "j"
## [5,] "g"  "h"  "i"  "j"
```

c)La cantidad de permutaciones y combinaciones con $n=30$ y $r=20$ sin repetición.

```
#Permutaciones sin repeticion
permutaciones = factorial(30)/factorial(30-20)
print(paste("Permutaciones sin repeticion:", permutaciones))
```

```
## [1] "Permutaciones sin repeticion: 7.30965773291973e+25"
```

```
#Combinaciones sin repeticion
comb = factorial(30)/(factorial(20)*factorial(30-20))
print(paste("combinaciones sin repeticion:", comb))
```

```
## [1] "combinaciones sin repeticion: 30045015"
```

d)La cantidad de permutaciones posibles con $n=7$ y $r=5$ con y sin repetición

```
# Permutaciones con repetición
permutaciones_con_rep = 7^5
print(paste("Permutaciones con repetición:", permutaciones_con_rep))
```

```
## [1] "Permutaciones con repetición: 16807"
```

```
# Permutaciones sin repetición
permutaciones_sin_rep = factorial(7)/factorial(7-5)
print(paste("Permutaciones sin repetición:", permutaciones_sin_rep))
```

```
## [1] "Permutaciones sin repetición: 2520"
```

e)Las combinaciones de largo dos con los números 1, 2, 3, 4 y 5 con y sin repetición.

```
# Combinaciones con repetición
numeros <- c(1, 2, 3, 4, 5)
combinaciones_con_rep = combinations(5,2,v = numeros ,repeats.allowed = TRUE)
print("Combinaciones con repetición:")
```

```
## [1] "Combinaciones con repetición:"
```

```
print(t(combinaciones_con_rep))
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
## [1,]    1    1    1    1    1    2    2    2    2    3    3    3    4    4
## [2,]    1    2    3    4    5    2    3    4    5    3    4    5    4    5
##      [,15]
## [1,]      5
## [2,]      5
```

```
# Combinaciones sin repetición
combinaciones_sin_rep = combinations(5,2,v = numeros,repeats.allowed = FALSE)
print("Combinaciones sin repetición:")
```

```
## [1] "Combinaciones sin repetición:"
```

```
print(t(combinaciones_sin_rep))
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    1    1    1    2    2    2    3    3    4
## [2,]    2    3    4    5    3    4    5    4    5    5
```

f) La cantidad de permutaciones y combinaciones con $n=50$ y $r=10$ sin repetición

```
# Permutaciones sin repetición
permutaciones_sin_rep = factorial(50)/factorial(50-10)
print(paste("Permutaciones sin repetición:", permutaciones_sin_rep))
```

```
## [1] "Permutaciones sin repetición: 37276043023295560"
```

```
# Combinaciones sin repetición
combinaciones_sin_rep = factorial(50)/(factorial(10)*factorial(50-10))
print(paste("Combinaciones sin repetición:", combinaciones_sin_rep))
```

```
## [1] "Combinaciones sin repetición: 10272278169.9999"
```

g) La cantidad de permutaciones circulares posibles con $n=6$.

```
# Permutaciones circulares
permutaciones_circulares = factorial(6-1)
print(paste("Permutaciones circulares:", permutaciones_circulares))
```

```
## [1] "Permutaciones circulares: 120"
```

h) Las combinaciones de largo tres con las letras x, y, z, w y q con y sin repetición.

```
# Combinaciones con repetición
letras <- c("x", "y", "z", "w", "q")
combinaciones_con_rep = expand.grid(letras, letras, letras)
print("Combinaciones con repetición:")
```

```
## [1] "Combinaciones con repetición:"
```

```
print(combinaciones_con_rep)
```

```
##      Var1 Var2 Var3
## 1      x     x     x
## 2      y     x     x
## 3      z     x     x
## 4      w     x     x
## 5      q     x     x
## 6      x     y     x
## 7      y     y     x
## 8      z     y     x
## 9      w     y     x
```

## 10	q	y	x
## 11	x	z	x
## 12	y	z	x
## 13	z	z	x
## 14	w	z	x
## 15	q	z	x
## 16	x	w	x
## 17	y	w	x
## 18	z	w	x
## 19	w	w	x
## 20	q	w	x
## 21	x	q	x
## 22	y	q	x
## 23	z	q	x
## 24	w	q	x
## 25	q	q	x
## 26	x	x	y
## 27	y	x	y
## 28	z	x	y
## 29	w	x	y
## 30	q	x	y
## 31	x	y	y
## 32	y	y	y
## 33	z	y	y
## 34	w	y	y
## 35	q	y	y
## 36	x	z	y
## 37	y	z	y
## 38	z	z	y
## 39	w	z	y
## 40	q	z	y
## 41	x	w	y
## 42	y	w	y
## 43	z	w	y
## 44	w	w	y
## 45	q	w	y
## 46	x	q	y
## 47	y	q	y
## 48	z	q	y
## 49	w	q	y
## 50	q	q	y
## 51	x	x	z
## 52	y	x	z
## 53	z	x	z
## 54	w	x	z
## 55	q	x	z
## 56	x	y	z
## 57	y	y	z
## 58	z	y	z
## 59	w	y	z
## 60	q	y	z
## 61	x	z	z
## 62	y	z	z
## 63	z	z	z

## 64	w	z	z
## 65	q	z	z
## 66	x	w	z
## 67	y	w	z
## 68	z	w	z
## 69	w	w	z
## 70	q	w	z
## 71	x	q	z
## 72	y	q	z
## 73	z	q	z
## 74	w	q	z
## 75	q	q	z
## 76	x	x	w
## 77	y	x	w
## 78	z	x	w
## 79	w	x	w
## 80	q	x	w
## 81	x	y	w
## 82	y	y	w
## 83	z	y	w
## 84	w	y	w
## 85	q	y	w
## 86	x	z	w
## 87	y	z	w
## 88	z	z	w
## 89	w	z	w
## 90	q	z	w
## 91	x	w	w
## 92	y	w	w
## 93	z	w	w
## 94	w	w	w
## 95	q	w	w
## 96	x	q	w
## 97	y	q	w
## 98	z	q	w
## 99	w	q	w
## 100	q	q	w
## 101	x	x	q
## 102	y	x	q
## 103	z	x	q
## 104	w	x	q
## 105	q	x	q
## 106	x	y	q
## 107	y	y	q
## 108	z	y	q
## 109	w	y	q
## 110	q	y	q
## 111	x	z	q
## 112	y	z	q
## 113	z	z	q
## 114	w	z	q
## 115	q	z	q
## 116	x	w	q
## 117	y	w	q

```
## 118    z    w    q
## 119    w    w    q
## 120    q    w    q
## 121    x    q    q
## 122    y    q    q
## 123    z    q    q
## 124    w    q    q
## 125    q    q    q
```

```
# Combinaciones sin repetición
combinaciones_sin_rep = combn(letras, 3, FUN = NULL, simplify = TRUE)
print("Combinaciones sin repetición:")
```

```
## [1] "Combinaciones sin repetición:"
```

```
print(t(combinaciones_sin_rep))
```

```
##      [,1] [,2] [,3]
## [1,] "x"  "y"  "z"
## [2,] "x"  "y"  "w"
## [3,] "x"  "y"  "q"
## [4,] "x"  "z"  "w"
## [5,] "x"  "z"  "q"
## [6,] "x"  "w"  "q"
## [7,] "y"  "z"  "w"
## [8,] "y"  "z"  "q"
## [9,] "y"  "w"  "q"
## [10,] "z"  "w"  "q"
```

5- Considere un problema de un repartidor que debe entregar paquetes en 40 casas y volver al origen sin pasar dos veces por la misma casa

```
CantidadRutas = factorial(40)

ProbabilidadOptima = 1/CantidadRutas

print(paste("La probabilidad de seleccionar la ruta mas corta es de :", ProbabilidadOptima))
```

a) ¿Cuál es la probabilidad de que elija la ruta más corta? ¿Y la más larga?

```
## [1] "La probabilidad de seleccionar la ruta mas corta es de : 1.22561743912839e-48"
```

```
ProbabilidadDos = 1/ (CantidadRutas - 1)

print(paste("La probabilidad de seleccionar la ruta mas corta, distinta a la anterior :", ProbabilidadD
```

b) Si elige una ruta al azar que es distinta a la anterior, ¿cuál es la probabilidad de que sea la ruta más corta?

```
## [1] "La probabilidad de seleccionar la ruta mas corta, distinta a la anterior : 1.22561743912839e-48"
```

6. Un restaurante tiene 4 camareros que deben atender a los clientes. Ana atiende al 20 % de los clientes y olvida 1 de cada 15 órdenes; Bob atiende al 50 % de los clientes y olvida 1 de cada 8 órdenes; Carlos atiende al 20 % de los clientes y olvida 1 de cada 12 órdenes; Diana atiende al 10 % de los clientes y olvida 2 de cada 7 órdenes.

a) (5 puntos) Formule la ecuación para resolver este problema. *Formulando el problema:*

$$P(A_i|O) = \frac{P(O|A_i) \times P(A_i)}{P(O|Ana) \times P(Ana) + P(O|Bob) \times P(Bob) + P(O|Carlos) \times P(Carlos) + P(O|Diana) \times P(Diana)}$$

Además nos entregan:

```
ProbAna = 0.20
ProbBob = 0.50
ProbCarlos = 0.20
ProbDiana = 0.10
```

```
POlvidoAna = 1/15
POlvidoBob = 1/8
POlvidoCarlos = 1/12
POlvidoDiana = 2/7
```

```
PNoOlvidoAna = 1 - (1/15)
PNoOlvidoBob = 1 - (1/8)
PNoOlvidoCarlos = 1 - (1/12)
PNoOlvidoDiana = 1 - (2/7)
```

```
PNoOlvidoTotal = PNoOlvidoAna * ProbAna + PNoOlvidoBob * ProbBob + PNoOlvidoCarlos * ProbCarlos + PNoOlvidoDiana * ProbDiana
PCarlosDadoNoOlvido = (PNoOlvidoCarlos * ProbCarlos) / PNoOlvidoTotal
print(paste(PCarlosDadoNoOlvido))
```

b) (5 puntos) ¿Cuál es la probabilidad de que el cliente haya sido atendido por Carlos?

```
## [1] "0.208587295137478"
```

```
# Probabilidad de que el cliente haya sido atendido por Ana y no olvidara la orden
PAAnaDadoNoOlvido = (PNoOlvidoAna * ProbAna) / PNoOlvidoTotal

# Probabilidad de que el cliente haya sido atendido por Diana y no olvidara la orden
PDianaDadoNoOlvido = (PNoOlvidoDiana * ProbDiana) / PNoOlvidoTotal

# Probabilidad total de que el cliente haya sido atendido por Ana o Diana
```

```
PAnaODianaDadoNoOlvido = PAAnaDadoNoOlvido + PDianaDadoNoOlvido

print(paste("La probabilidad que haya sido atendido por ana o diana es:",PAAnaODianaDadoNoOlvido))
```

c) (5 puntos) ¿Cuál es la probabilidad de que el cliente haya sido atendido por Ana o Diana?

```
## [1] "La probabilidad que haya sido atendido por ana o diana es: 0.293647568738995"
```

```
PtotalOlvido = ProbDiana*POlvidoDiana + ProbCarlos*POlvidoCarlos + ProbBob*POlvidoBob + ProbAna*POlvidoAna

PanaOlvidado = (POlvidoAna*ProbAna)/PtotalOlvido
PbobOlvidado = (POlvidoBob*ProbBob)/PtotalOlvido
PdianaOlvidado = (POlvidoDiana*ProbDiana)/PtotalOlvido
PcarlosOlvidado = (POlvidoCarlos*ProbCarlos)/PtotalOlvido

Ptotal = PanaOlvidado+PbobOlvidado+PdianaOlvidado+PcarlosOlvidado

print(paste("La probabilidad de ser atendido por cualquiera es:",Ptotal))
```

d) (5 puntos) Calcule la probabilidad de que el cliente haya sido atendido por cualquiera de los camareros. ¿Qué observa?

```
## [1] "La probabilidad de ser atendido por cualquiera es: 1"
```

```
#Observamos que la suma de las probabilidades de ser atendido por cada camarero es igual a 1,
#lo que tiene sentido ya que un cliente, en este escenario, definitivamente será atendido por uno
#de los camareros. Esto es una confirmación de que las probabilidades proporcionadas
#son coherentes y cubren todos los escenarios posibles para el cliente.
```

7. (10 puntos) De un grupo de 35 personas, se quiere conocer la opinión de 4 personas (elegidas al azar) sobre la legalización de la marihuana. ###Si se sabe que 18 personas están a favor y 17 en contra, ¿cuál es la probabilidad de que las 4 personas seleccionadas estén en contra?

```
# Probabilidad de que la primera persona seleccionada esté en contra
Pprimero = 17/35
# Probabilidad de que la segunda persona seleccionada esté en contra
Psegundo = 16/34
# Probabilidad de que la tercera persona seleccionada esté en contra
Ptercero = 15/33
# Probabilidad de que la cuarta persona seleccionada esté en contra
Pcuarto = 14/32

# Calculamos la probabilidad total
Ptotal = Pprimero * Psegundo * Ptercero * Pcuarto

print(paste("La probabilidad de que las 4 personas seleccionadas estén en contra es de", Ptotal, "\n"))
```

```
## [1] "La probabilidad de que las 4 personas seleccionadas estén en contra es de 0.0454545454545454 \n"
```