

Ejercicio 02: Probabilidades, Permutaciones y Combinaciones

Reinaldo Pacheco

2023-10-08

Ejercicios 02: Probabilidades, Permutaciones y Combinaciones

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

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

```
##
## Attaching package: 'combinat'

## The following object is masked from 'package:utils':
##
##      combn

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

2- Defina los siguientes conceptos y proporcione las fórmulas para calcularlos: Permutación, Combinación, Permutación Circular, Permutación con Repetición.

Permutación

La permutación es el número de formas diferentes en las que se pueden organizar elementos distintos en un orden particular. Se denota como $P(n, r)$ y su fórmula es la siguiente:

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

- n es el número total de elementos.
- r es el número de elementos seleccionados para la permutación.
- $n!$ es el factorial de n

Combinación

La combinación es el número de formas diferentes en las que se pueden seleccionar r elementos de un conjunto de n . A diferencia de la permutación, no importa el orden de los elementos. Se denota como $C(n, r)$ y su fórmula es la siguiente:

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

- n es el número total de elementos.
- r es el número de elementos seleccionados para la combinación.
- $n!$ es el factorial de n .
- $r!$ es el factorial de r .
- $(n-r)!$ es el factorial de $n-r$.

Permutación Circular

La permutación circular es el número de formas diferentes en las que se pueden organizar n elementos en un círculo. Se denota como $PC(n)$ y su fórmula es la siguiente:

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

- n es el número total de elementos.

Permutación con Repetición

La permutación con repetición es el número de formas diferentes en las que se pueden organizar n elementos, se pueden repetir. Se denota como $PR(n; n_1, n_2, \dots, n_k)$ y su fórmula es la siguiente:

$$PR(n; n_1, n_2, \dots, n_k) = \frac{n!}{n_1! * n_2! * \dots * n_k!}$$

- n es el número total de elementos.
- n_1, n_2, \dots, n_k son las repeticiones de cada elemento.
- $n!$ es el factorial de n .
- $n_1!, n_2!$ son los factoriales de n_1, n_2

3- Busque en la ayuda de R las funciones combinations, permutations del paquete gtools y permn, combn del paquete combinat. Explique brevemente cómo funcionan y qué diferencias hay entre ellas.

Paquete gtools

1. **combinations:** La función combinations del paquete gtools genera todas las combinaciones posibles de un conjunto de elementos dados.
2. **permutations:** La función permutations del paquete gtools genera todas las permutaciones posibles de un conjunto de elementos dados.

Paquete combinat

1. **permn**: La función **permn** del paquete **combinat** genera todas las permutaciones del vector proporcionado sin usar repetición.
2. **combn**: La función **combn** del paquete **combinat** genera todas las combinaciones posibles de un conjunto de elementos dados.

El paquete **gtools** genera combinaciones y permutaciones sin repetición a partir de tamaños especificados. En contraste, **combinat** crea permutaciones y combinaciones sin repetición directamente desde un vector

4- Calcule:

- a) Permutaciones con y sin repetición con $n=9$ y $r=4$

```
permutacion_con_repeticion = nrow(permutations(9, 4 ,repeats.allowed = TRUE))
print(paste("Permutaciones con repetición:", permutacion_con_repeticion))
```

```
## [1] "Permutaciones con repetición: 6561"
```

```
permutacion_sin_repeticion = nrow(permutations(9, 4 , repeats.allowed = FALSE))
print(paste("Permutaciones sin repetición:", permutacion_sin_repeticion))
```

```
## [1] "Permutaciones sin repetición: 3024"
```

- b) Combinaciones de largo cuatro con las letras f, g, h, i y j con y sin repetición

```
letras = c('f', 'g', 'h', 'i', 'j')
combinaciones_sin_repeticion = choose(length(letras), 4)
print(paste("Combinaciones sin repetición:", combinaciones_sin_repeticion))
```

```
## [1] "Combinaciones sin repetición: 5"
```

```
combinaciones_repeticion = choose(length(letras) + 4 - 1, 4)
print(paste("Combinaciones con repetición:", combinaciones_repeticion))
```

```
## [1] "Combinaciones con repetición: 70"
```

- c) Permutaciones y combinaciones con $n=30$ y $r=20$ sin repetición

```
n = 30
r = 20

permutaciones_sin_repeticion = factorial(n) / factorial(n - r)
print(paste("Permutaciones sin repetición:", permutaciones_sin_repeticion))
```

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

```
combinaciones_sin_repeticion = choose(n, r)
print(paste("Combinaciones sin repetición:", combinaciones_sin_repeticion))
```

```
## [1] "Combinaciones sin repetición: 30045015"
```

d) Permutaciones con y sin repetición con $n=7$ y $r=5$

```
n = 7
r = 5

permutaciones_sin_repeticion = factorial(n) / factorial(n - r)
print(paste("Permutaciones sin repetición:", permutaciones_sin_repeticion))
```

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

```
permutaciones_repeticion = n^r
print(paste("Permutaciones con repetición:", permutaciones_repeticion))
```

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

e) Combinaciones de largo 2 con los números 1, 2, 3, 4 y 5 con y sin repetición

```
numeros = c(1,2,3,4,5)

combinaciones_sin_repeticion = choose(length(numeros), 2)
print(paste("Combinaciones sin repetición:", combinaciones_sin_repeticion))
```

```
## [1] "Combinaciones sin repetición: 10"
```

```
combinaciones_repeticion = choose(length(numeros) + 2 - 1, 2)
print(paste("Combinaciones con repetición:", combinaciones_repeticion))
```

```
## [1] "Combinaciones con repetición: 15"
```

f) Permutaciones y combinaciones con $n=50$ y $r=10$ sin repetición

```
n = 50
r = 10

permutaciones_sin_repeticion = factorial(n) / factorial(n - r)
print(paste("Permutaciones sin repetición:", permutaciones_sin_repeticion))
```

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

```
combinaciones_sin_repeticion = choose(n, r)
print(paste("Combinaciones sin repetición:", combinaciones_sin_repeticion))
```

```
## [1] "Combinaciones sin repetición: 10272278170"
```

g) Permutaciones circulares posibles con $n=6$

```
n = 6
permutaciones_circulares = factorial(n - 1)

print(paste("Permutaciones circulares:", permutaciones_circulares))
```

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

h) combinaciones de largo 3 con las letras x, y, z, w y q con y sin repetición.

```
letras = c('x', 'y', 'z', 'w', 'q')

# Combinaciones sin repetición
combinaciones_sin_repeticion = choose(length(letras), 3)
print(paste("Combinaciones sin repetición:", combinaciones_sin_repeticion))
```

```
## [1] "Combinaciones sin repetición: 10"
```

```
# Combinaciones con repetición
combinaciones_repeticion = length(letras)^3
print(paste("Combinaciones con repetición:", combinaciones_repeticion))
```

```
## [1] "Combinaciones con repetición: 125"
```

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

```
# Total de rutas posibles
total_rutas = factorial(40)

# Probabilidad de elegir la ruta más corta o la más larga
prob_1 = 1/total_rutas

print(paste("Probabilidad Ruta más corta o larga:", prob_1))
```

```
## [1] "Probabilidad Ruta más corta o larga: 1.22561743912839e-48"
```

```
# Probabilidad de elegir la ruta más corta si la ruta previamente elegida fue diferente
prob_2 = 1/(total_rutas - 1)
print(paste("Probabilidad de que sea la ruta más corta distinta a la anterior:", prob_2))
```

```
## [1] "Probabilidad de que sea la ruta más corta distinta a la anterior: 1.22561743912839e-48"
```

Si se intenta usar funciones como combinations o permutations para generar todas las posibles rutas puede ocurrir un problema debido a las limitaciones de memoria. Es menos eficiente generar todas las permutaciones, que calcular probabilidades usando las propiedades y las fórmulas matemáticas anteriormente explicadas.

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.

- Formule la ecuación para resolver este problema.
- ¿Cuál es la probabilidad de que el cliente haya sido atendido por Carlos?
- ¿Cuál es la probabilidad de que el cliente haya sido atendido por Ana o Diana?
- Calcule la probabilidad de que el cliente haya sido atendido por cualquiera de los camareros. ¿Qué observa?

Este problema se puede representar mediante el Teorema de Bayes

$$P(\text{Camarero}|\neg\text{Olvido}) = \frac{P(\neg\text{Olvido}|\text{Camarero}) \times P(\text{Camarero})}{P(\neg\text{Olvido})}$$

- ¿Cuál es la probabilidad de que el cliente haya sido atendido por Carlos?

```
prob_ana = 0.20
olvido_ana = 1/15
no_olvido_ana = (1-olvido_ana)

prob_bob = 0.50
olvido_bob = 1/8
no_olvido_bob = (1-olvido_bob)

prob_carlos = 0.20
olvido_carlos = 1/12
no_olvido_carlos = (1-olvido_carlos)

prob_diana = 0.10
olvido_diana = 2/7
no_olvido_diana = (1-olvido_diana)

p_atendido_carlos = (prob_carlos*no_olvido_carlos)/
    (prob_ana*no_olvido_ana + prob_bob*no_olvido_bob+
    prob_diana*no_olvido_diana+prob_carlos*no_olvido_carlos)

cat("La probabilidad es:", p_atendido_carlos)
```

La probabilidad es: 0.2085873

- ¿Cuál es la probabilidad de que el cliente haya sido atendido por Ana o Diana?

```
prob_ana = 0.20
olvido_ana = 1/15
no_olvido_ana = (1-olvido_ana)

prob_bob = 0.50
olvido_bob = 1/8
no_olvido_bob = (1-olvido_bob)

prob_carlos = 0.20
olvido_carlos = 1/12
no_olvido_carlos = (1-olvido_carlos)

prob_diana = 0.10
olvido_diana = 2/7
```

```

no_olvido_diana = (1-olvido_diana)

p_atendido_ana = (prob_ana*no_olvido_ana)/
    (prob_carlos*no_olvido_carlos
    + prob_bob*no_olvido_bob+prob_diana*no_olvido_diana
    +prob_ana*no_olvido_ana)

p_atendido_diana = (prob_diana*no_olvido_diana)/
    (prob_carlos*no_olvido_carlos + prob_bob*no_olvido_bob+
    prob_ana*no_olvido_ana+
    prob_diana*no_olvido_diana)

p_atendido_ana_diana = p_atendido_ana + p_atendido_diana

prob_ana_o_diana = prob_ana + prob_diana
cat("La probabilidad es:", p_atendido_ana_diana)

```

```
## La probabilidad es: 0.2936476
```

d) Calcule la probabilidad de que el cliente haya sido atendido por cualquiera de los camareros.

```

p_olvido = prob_ana * olvido_ana +
    prob_bob * olvido_bob+
    prob_carlos * olvido_carlos +
    prob_diana * olvido_diana

p_ana_dado_olvido = (olvido_ana * prob_ana) / p_olvido
p_bob_dado_olvido = (olvido_bob * prob_bob) / p_olvido
p_carlos_dado_olvido = (olvido_carlos * prob_carlos) / p_olvido
p_diana_dado_olvido = (olvido_diana * prob_diana) / p_olvido

cat("Probabilidad es de:", p_ana_dado_olvido +
    p_bob_dado_olvido +
    p_carlos_dado_olvido +
    p_diana_dado_olvido)

```

```
## Probabilidad es de: 1
```

se observa que la probabilidad es de 1 o 100% esto quiere decir que un cliente siempre será atendido por cualquiera de los camareros, lo cual es lógico considerando el enunciado de que los camareros deben atender a todos los clientes.

7- 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?

```

prob_1ero = 17/35
prob_2do = 16/34
prob_3ero = 15/33
prob_4to = 14/32

prob_total = (prob_1ero*prob_2do*prob_3ero*prob_4to)
print(paste("La probabilidad es de:", prob_total))

```

```
## [1] "La probabilidad es de: 0.0454545454545454"
```