

Introduction to Multivariate Statistics

November 3, 2020

0.0.1 EJEMPLO 1

Para calcular la media aritmética es necesario ocupar la función `mean()`.

```
[1]: # Media del vector

from numpy import array, mean
# Definiendo el vector
v = array([1, 2, 3, 4, 5, 6])
print(v)
print(mean(v)) # Media de v
```

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

```
3.5
```

Es posible calcular la media de una fila o columna de alguna matriz dando el valor 0 o 1 al argumento `axis`.

```
[8]: # Medias de una matriz

from numpy import array, mean
# Definiendo la matriz
M = array([
    [1, 2, 3, 4, 5, 6],
    [1, 2, 3, 4, 5, 6]
])
print(M)
# Media de las columnas
media_col = mean(M, axis = 0)
print(media_col)
# Media de las filas
media_fila = mean(M, axis = 1)
print(media_fila)
```

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

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

```
[1. 2. 3. 4. 5. 6.]
```

```
[3.5 3.5]
```

0.0.2 EJEMPLO 2

Para calcular la varianza poblacional de un vector utilizamos la función `var()`, sin embargo, si queremos calcular la varianza muestral, debemos indicar como argumento `ddof = 1`.

```
[7]: # Varianza del vector

from numpy import array, var
# Definiendo el vector
v = array([1, 2, 3, 4, 5, 6])
print(v)
# Calculando la varianza muestral
print(var(v, ddof = 1))
```

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

```
3.5
```

La función `var()` puede calcular la varianza de las filas o columnas de una matriz si especificamos el valor 0 o 1 en el argumento `axis`.

```
[9]: # Varianzas de una matriz

from numpy import array, var
# Definiendo la matriz
M = array([
    [1, 2, 3, 4, 5, 6],
    [1, 2, 3, 4, 5, 6]
])
print(M)
# Varianza de las columnas
var_col = var(M, ddof = 1, axis = 0)
print(var_col)
# Varianza de las filas
var_fila = var(M, ddof = 1, axis = 1)
print(var_fila)
```

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

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

```
[0. 0. 0. 0. 0. 0.]
```

```
[3.5 3.5]
```

0.0.3 EJEMPLO 3

Ocupamos la función `std()` para calcular la desviación estándar, y al argumento `ddof` les especificamos el valor 1 para calcular la desviación estándar muestral insesgada.

```
[10]: # Desviaciones estándar de una matriz

from numpy import array, std
# Definiendo la matriz
```

```

M = array([
    [1, 2, 3, 4, 5, 6],
    [1, 2, 3, 4, 5, 6]
])
print(M)
# DS de las columnas
ds_col = std(M, ddof = 1, axis = 0)
print(ds_col)
# DS de las filas
ds_fila = std(M, ddof = 1, axis = 1)
print(ds_fila)

```

```

[[1 2 3 4 5 6]
 [1 2 3 4 5 6]]
[0. 0. 0. 0. 0. 0.]
[1.87082869 1.87082869]

```

0.0.4 EJEMPLO 4

Es necesario ocupar la función `cov()`.

```

[13]: # Covarianza de vectores

from numpy import array, cov
# Definiendo el primer y segundo vector
x = array([1, 2, 3, 4, 5, 6, 7, 8, 9])
print(x)
y = array([9, 8, 7, 6, 5, 4, 3, 2, 1])
print(y)
# Calculando la covarianza
Sigma = cov(x,y)[0, 1]
print(Sigma)

```

```

[1 2 3 4 5 6 7 8 9]
[9 8 7 6 5 4 3 2 1]
-7.5

```

0.0.5 EJEMPLO 5

Para calcular el coeficiente de Pearson se ocupa la función `corrcoef()`.

```

[14]: # Correlación de un vector

from numpy import array, corrcoef
# Definiendo el primer y segundo vector
x = array([1, 2, 3, 4, 5, 6, 7, 8, 9])
print(x)
y = array([9, 8, 7, 6, 5, 4, 3, 2, 1])
print(y)

```

```
# Calculando la correlación
r = corrcoef(x,y)[0, 1]
print(r)
```

```
[1 2 3 4 5 6 7 8 9]
[9 8 7 6 5 4 3 2 1]
-1.0
```

0.0.6 EJEMPLO 6

Si nosotros definimos una matriz a la cual le queremos calcular la covarianza, es necesario transponerla para obtener el resultado correcto.

```
[15]: # Matriz de covarianzas

from numpy import array, cov
# Definiendo la matriz de observaciones
X = array([
    [1, 5, 8],
    [3, 5, 11],
    [2, 4, 9],
    [3, 6, 10],
    [1, 5, 10]
])
print(X)
# Calculando la matriz de covarianzas
Sigma = cov(X.T)
print(Sigma)
```

```
[[ 1  5  8]
 [ 3  5 11]
 [ 2  4  9]
 [ 3  6 10]
 [ 1  5 10]]
[[1.  0.25 0.75]
 [0.25 0.5  0.25]
 [0.75 0.25 1.3 ]]
```