

▼ Asignación

- 1. Convertir en coordenadas polares los datos de de y d1
- 2. Graficar un cardioide en coordenadas polares en Python

▼ Cambio de coordenadas en rectangulares y polares

```
import cmath

Z1_rect= 3+5j
type(Z1_rect)

complex

Z1_polar=cmath.polar(Z1_rect)
print(Z1_polar)

(5.830951894845301, 1.0303768265243125)

Z1_mag, Z1_ang=cmath.polar(Z1_rect)
Z1_ang

1.0303768265243125

# Otra forma
Z2_polar=abs(Z1_rect),cmath.phase(Z1_rect)
print(Z2_polar)

(5.830951894845301, 1.0303768265243125)
```

▼ 2. Dibujar un carioide con coordenadas polares en python

Forma del carioide

$$r = a \pm b \times Cos\theta$$
$$r = a \pm b \times Sen\theta$$

$$6 + 6Cos\theta$$

Ejemplo:

$$r = \sqrt{X^2 + Y^2}$$
$$x = Cos\theta$$

Entonces:

$$\frac{x}{r} = Cos\theta = \frac{x}{\sqrt{X^2 + Y^2}}$$

$$\sqrt{X^2 + Y^2} = 6 + 6 \times \frac{x}{\sqrt{X^2 + Y^2}}$$

Como se graficaría en coordenadas polares Damos valores de

θ

y encontramos valores de r

```
from math import cos
#from math import sen
from math import radians
# Voy a crear una tabla en x y y para los diferentes valores
list_grados=[0,15,30,45,60,75,90,105,120,135,150,165,180,195,210,225,240,255,270,285,300,315,
list_grados2=[]

for i in list_grados:
    list_grados2.append(radians(i))

def carioide (grados):
    i=2+(2*cos(grados))
    return i
Lista_r=[]
for i in list_grados2:
    Lista_r.append(carioide(i))
#Lista_r

#list_grados2

from matplotlib.mathtext import List
plt.polar(list_grados2,Lista_r,'r')
```

Segunda forma

```
# Python programa para imprimir un cardioide en coordenadas polares
import numpy as np
from matplotlib import pyplot as plt

theta = np.linspace(0, 2*np.pi, 1000)

r = 2 - 10 * np.sin(theta)

plt.polar(theta, r, 'r')

plt.show()
```

▼ Estadísticas descriptivas

Escala de medida

- Nominal
- Ordinal
- Intervalo (cero relativo)
- Razon (cero absoluto)

```
import pandas as pd
import numpy as np
```

$$a_n = a_1 + r(n - 1)$$

```
def prog(r, n, a1):
    an = a1 + r*(n-1)
    seq = np.arange(start=a1, stop=an, step=r)
    return seq
```

```
prog(r=7, n=20, a1=15)

array([ 15,  22,  29,  36,  43,  50,  57,  64,  71,  78,  85,  92,  99,
        106, 113, 120, 127, 134, 141])
```

```
np.random.seed(123)
```

```
df1 = pd.DataFrame({
    'de': np.sort(np.random.normal(loc = 4, scale = 1, size=96)),
    'dl': np.sort(np.random.normal(loc=4.5, scale=1.2, size=96)),
    'ddd': np.repeat(prog(r=7, n=25, a1=15), 4)
})
```

```
df1['localidad'] = np.repeat(['l1','l2']*24, 2)
df1.head()
```

```
df1.tail()
```

```
df1.plot.scatter(x='ddd', y='de')
```

```
df1.plot.scatter(x='ddd', y='dl')
```

```
import matplotlib.pyplot as plt
```

```
from numpy.ma.core import arange
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
```

```
for i in np.unique(df1['localidad']):
    df1_fil = df1[df1['localidad']==i]
    ax.scatter(df1_fil['ddd'], df1_fil['de'], df1_fil['dl'],
               label = f'localidad: {i}')
```

```
ax.legend()
ax.set_xlabel('DDD')
ax.set_ylabel('Diam Ecuatorial')
```

```
ax.set_zlabel('Diam Longitudinal')
plt.show()
```

```
df1['idx1'] = df1['de']/df1['dl']
df1['idx2'] = df1['dl']/df1['de']
df1.head()
```

```
plt.scatter(df1['ddd'], df1['idx2'])
plt.xlabel('DDD')
plt.ylabel('Indicie')
plt.show()
```

```
plt.scatter(df1[df1['localidad']=='l1']['ddd'],
            df1[df1['localidad']=='l1']['idx2'])
plt.xlabel('DDD')
plt.ylabel('Indicie')
plt.show()
```

```
a = df1['d1']/2
b = df1['de']/2

df1['e'] = np.sqrt(np.abs(1-(b/a)**2))
df1.head()
```

```
plt.scatter(df1['ddd'],
            df1['e'])
plt.xlabel('DDD')
plt.ylabel('E')
plt.show()
```

Asignación

1. Convertir en coordenadas polares los datos de `de` y `d1`
2. Graficar un cardioide en coordenadas polares en Python

✓ 0s completed at 7:19 PM

