Nicolás Galindo Ramírez

1022409637

28/03/2022

Estadisticas descriptivas

Escalas de medida

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

Escalas de medida

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

```
1 import pandas as pd
2 import numpy as np
```

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

```
1 def prog(r, n, a1):
2   an = a1 + r*(n-1)
3   seq = np.arange(start=a1, stop=an, step=r)
4   return seq
5
6 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])
```

```
 1 np.random.seed(123)
 2
 3 df1 = pd.DataFrame({
 4     'de': np.sort(np.random.normal(loc = 4, scale = 1, size=96)),
 5     'dl': np.sort(np.random.normal(loc=4.5, scale=1.2, size=96)),
 6     'ddd': np.repeat(prog(r=7, n=25, a1=15), 4)
 7 })
 8
 9 df1['localidad'] = np.repeat(['l1','l2']*24, 2)
10 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['dl']/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 dl
2. Graficar un cardioide en coordenadas polares en Python

```
## Coordenadas polares:
import numpy as np
import matplotlib.pyplot as plt
x = df1['de']
y = df1['dl']
# r:
r = np.sqrt(x**2 + y**2)
```

```
# Tetha
t  = np.arctan(y/x)
cop = [[r],[t]]
print(cop)
```

```
[[0     1.868871
1     2.600002
2     2.836354
3     3.165198
4     3.208826
      ...
91    8.798825
92    8.923382
93    9.108493
94    9.265779
95    9.944633
Length: 96, dtype: float64], [0     0.872578
1     0.920876
2     0.847673
3     0.789708
4     0.783921
      ...
91    0.806804
92    0.804706
93    0.821212
94    0.832207
95    0.872654
Length: 96, dtype: float64]]
```

```
## Cardioide:
import numpy as np
import matplotlib.pyplot as plt
import math


plt.axes(projection = 'polar')
a=4
rads = np.arange(0,(2 * np.pi), 0.01)

for rad in rads:
    r = a + (a*np.cos(rad))
    plt.polar(rad,r,'r.')

plt.show()
```

✓ 3 s completado a las 14:56