Clase #15 Computación Estadística

Nicolas Galindo Ramirez - 1022409637

Asignación

[Articulo Referencia]

1. Generar unos datos de la dostribución log-normal, con un promedio de daño del **8%** y una desviación estandar de **3.6** (si algún valor es negativo se convierte a 0)
2. Extraer media, mediana, quartiles y percentiles (desde 5% hasta 95% cada 5)
3. Categorizar la variable según la escala diagramtica del *articulo referencia*
4. En que posición de debo ubicarme en la escala para estimar la severidad real

```
1 import seaborn as sns
2 import pandas as pd
3 import numpy as np
```

```
1 np.random.seed(1022409637)
2 por_sev = np.random.lognormal(mean = 0.08, sigma= 3.6, size= 100)
3 df1 = pd.DataFrame(por_sev,columns=['porcentaje_severidad'])
4 df1
```

```
print("valores menores a 0:",
      df1[df1['porcentaje_severidad']<0].count())
```

```
valores menores a 0: porcentaje_severidad    0
dtype: int64
```

```
print("valores mayores a 100:",
      df1[df1['porcentaje_severidad']>100].count())
```

```
valores mayores a 100: porcentaje_severidad    0
dtype: int64
```

```
df1.loc[df1.porcentaje_severidad>100]=100
df1
```

	porcentaje_severidad
0	44.857806
1	29.118270
2	100.000000
3	0.028331
4	0.121027
...	...
95	0.135961
96	0.003873
97	1.535365
98	11.036898
99	100.000000

100 rows × 1 columns

```
#Media y mediana
import statistics

print('Promedio:'+str(por_sev.mean()))
print('Mediana:'+str(statistics.median(por_sev)))
```

```
Promedio:17.27170550535232
Mediana:0.7979220285982338
```

```
#Quartiles y percentiles
```

```
porenf_serie = pd.Series(por_sev)
porenf_serie.quantile([0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80,
```

```
0.05      0.001911
0.10      0.003817
0.15      0.013464
0.20      0.024733
0.25      0.041433
0.30      0.084208
0.35      0.142894
0.40      0.216876
0.45      0.389495
0.50      0.797922
0.55      1.235497
0.60      1.701593
0.65      2.384348
0.70      3.742066
0.75      7.416645
0.80     13.277109
0.85     50.849293
0.90    100.000000
0.95    100.000000
dtype: float64
```

```
from google.colab import files
from IPython.display import Image
#3) Categorizar la variable según la escala diagramtica del *articulo referencia*
Image('Escala_severidad.jpg', width = 600)
```

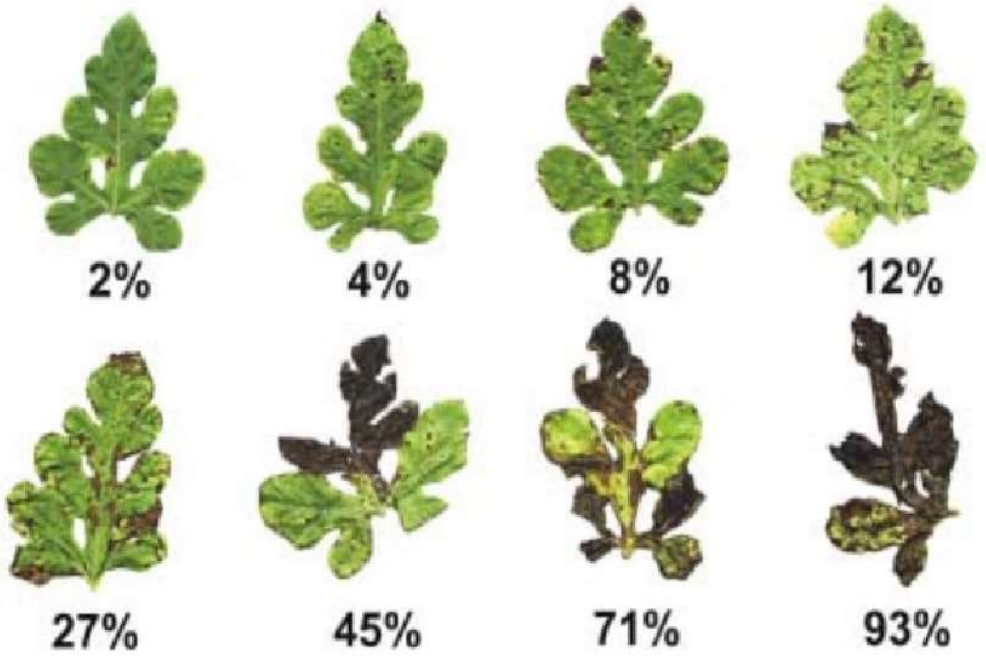


FIG. 1 - Escala diagramática para avaliação da severidade da mancha-de-cercospora em melancia (*Citrullus lanatus*). Os valores representam a porcentagem da área foliar necrosada.

```
cat_sev = []
for i in df1['porcentaje_severidad']:
```

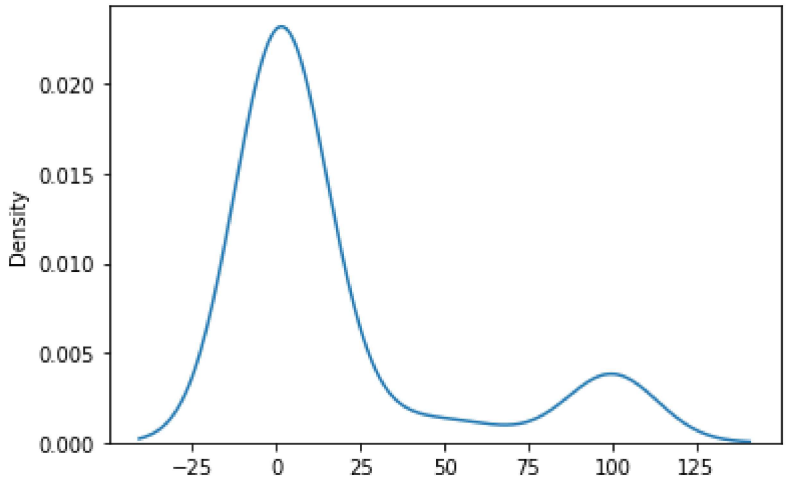
```
    if(i <=2):
      cat_sev.append('2%')
    elif(i <= 4):
      cat_sev.append('4%')
    elif(i <= 8):
      cat_sev.append('8%')
    elif(i <= 12):
      cat_sev.append('12%')
    elif(i <= 27):
      cat_sev.append('27%')
    elif(i <= 45):
      cat_sev.append('45%')
    elif(i <= 71):
      cat_sev.append('71%')
    else:
      cat_sev.append('93%')

cat_sev_serie = pd.Series(cat_sev)
cat_sev_serie.describe()
```

```
count     100
unique      8
top        2%
freq       61
dtype: object
```

```
import seaborn as sns
sns.kdeplot(por_sev)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3d5aac4bd0>
```

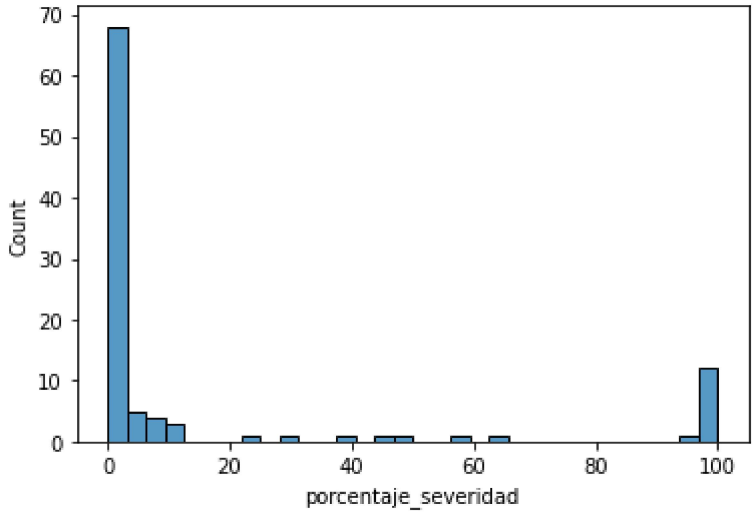



```
cat_sev_serie.value_counts().plot(kind='bar',xlabel='% Severidad', ylabel='N#')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3d5a7a7650>
```


```
1 sns.histplot(df1.porcentaje_severidad)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3d5a8c8c10>
```




```
1  #4)En que posición de debo ubicarme en la escala para estimar la severidad real
2
3  cont = cat_sev_serie.value_counts()
4
5  # Con el punto de la escala
6  frec1 = cont*[2, 4, 8, 12, 27, 45, 71, 93]
7  print(frec1.sum()/100)
8
9  # Con el punto medio
10 frec2 = cont*[1, 3, 6, 10, 19.5, 36, 58, 82]
11 print(frec2.sum()/100)
12
13 # punto percentil 20%
14 frec3 = cont*[0.4, 0.8, 1.6, 2.4, 5.4, 9, 14.2, 18.6]
15 print(frec3.sum()/100)
```

```
8.82
6.655
1.7639999999999998
```

Trabajo en Clase #15:

```
1 import pandas as pd
```

```
1 serie1 = pd.Series(['Fresa', 'Mora', 'Pera', 'Mango'])
2 type(serie1)
3 serie1
```

```
0    Fresa
1     Mora
2     Pera
3    Mango
dtype: object
```

```
1 print(serie1.shape)
2 print(serie1.size)
```

```
(4,)
4
```

```
1 serie2 = pd.Series(
2     {
3         'Fresa':3000,
4      'Mora':2500,
5      'Pera':2700,
6      'Mango':1800
7     }
8 )
9 serie2
```

```
Fresa    3000
Mora     2500
Pera     2700
Mango    1800
dtype: int64
```

```
1 type(serie2)
```

```
pandas.core.series.Series
```

```
1 serie2.shape
```

```
(4,)
```

```
1 serie2.index
```

```
Index(['Fresa', 'Mora', 'Pera', 'Mango'], dtype='object')
```

```
print(serie2[0])
print(serie2['Fresa'])
```

```
3000
3000
```

```
serie2.sum()
```

```
10000
```

```
serie2.cumsum()
```

```
Fresa     3000
Mora      5500
Pera      8200
Mango    10000
dtype: int64
```

```
print(serie2.min())
print(serie2.max())
print(serie2.mean())
print(serie2.std())
```

```
1800
3000
2500.0
509.9019513592785
```

```
import numpy as np

np.random.seed(123)

pcp = np.random.random(30)*10
pcp_serie = pd.Series(pcp)
pcp_serie
```

```
0     6.964692
1     2.861393
2     2.268515
3     5.513148
4     7.194690
5     4.231065
6     9.807642
7     6.848297
8     4.809319
9     3.921175
10    3.431780
11    7.290497
12    4.385722
13    0.596779
14    3.980443
```

```
15    7.379954
16    1.824917
17    1.754518
18    5.315514
19    5.318276
20    6.344010
21    8.494318
22    7.244553
23    6.110235
24    7.224434
25    3.229589
26    3.617887
27    2.282632
28    2.937140
29    6.309761
dtype: float64
```

```
pcp_serie.describe()
```

```
count    30.000000
mean      4.983096
std       2.258412
min       0.596779
25%       3.280137
50%       5.062416
75%       6.935593
max       9.807642
dtype: float64
```

```
serie_temp_min = pd.Series(np.random.uniform(4, 10, 30))
serie_temp_max = pd.Series(np.random.uniform(17, 22, 30))
serie_dia = pd.Series(np.arange(1, 31, 1))
```

```
df1 = pd.DataFrame(
    {
        'temp_min':serie_temp_min,
        'temp_max':serie_temp_max,
        'dia':serie_dia
    }
)
print(type(df1))
df1.head()
```

```
<class 'pandas.core.frame.DataFrame'>
```

	temp_min	temp_max	dia
0	4.552630	20.346569	1

```
df1.describe()
```

	temp_min	temp_max	dia
count	30.000000	30.000000	30.000000
mean	7.073958	19.430973	15.500000
std	1.463612	1.364626	8.803408
min	4.552630	17.080646	1.000000
25%	6.164677	18.294709	8.250000
50%	6.930158	19.823105	15.500000
75%	7.727130	20.438250	22.750000
max	9.913359	21.625662	30.000000

```
np.random.seed(2020)
porc_enf = np.random.exponential(1/10, 30)*100
porc_enf
```

```
array([42.88669551, 20.66659152,  7.12830686,  3.17228589,  4.10857714,
        2.44564175,  3.23623141,  4.2055173 , 19.81653964,  1.70432121,
        1.51855102, 14.15024263, 13.33037684,  4.3953354 ,  4.17172902,
       10.99021524,  2.44751121,  8.24228964,  1.32593293,  3.85275033,
       30.62168555,  1.47754104,  8.4260611 , 37.15859481,  6.99904079,
       11.01609394,  0.34789703,  6.0902548 ,  1.69426689,  6.46357047])
```

```
print(porc_enf.min())
print(porc_enf.max())
```

```
0.34789702531525246
42.8866955085488
```

```
cat_enf = []
for pe_i in porc_enf:
  if(pe_i == 0):
    cat_enf.append('N0')
  elif(pe_i <= 2.5):
    cat_enf.append('N1')
  elif(pe_i <= 5):
    cat_enf.append('N2')
  elif(pe_i <= 10):
    cat_enf.append('N3')
```

```
    elif(pe_i <= 20):
      cat_enf.append('N4')
    elif(pe_i <= 40):
      cat_enf.append('N5')
    elif(pe_i <= 80):
      cat_enf.append('N6')
    else:
      cat_enf.append('N7')

cat_enf_serie = pd.Series(cat_enf)
```

```
cat_enf_serie.describe()
```

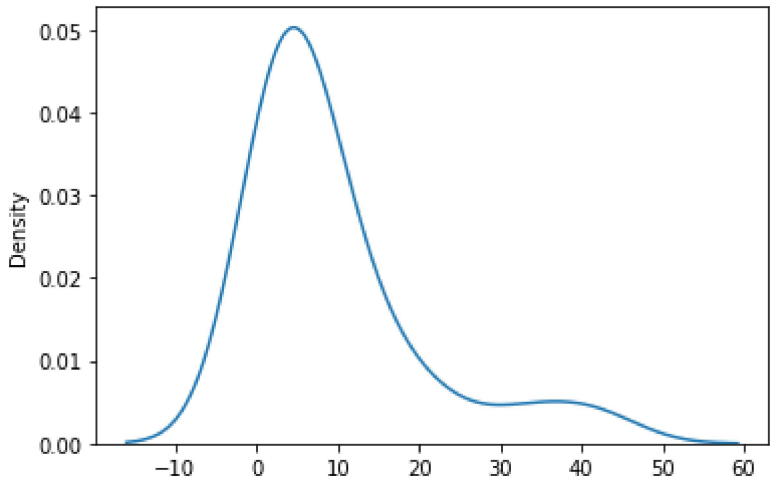
```
count     30
unique     6
top       N1
freq       8
dtype: object
```

```
cat_enf_serie.value_counts()
```

```
N1    8
N2    7
N3    6
N4    5
N5    3
N6    1
dtype: int64
```

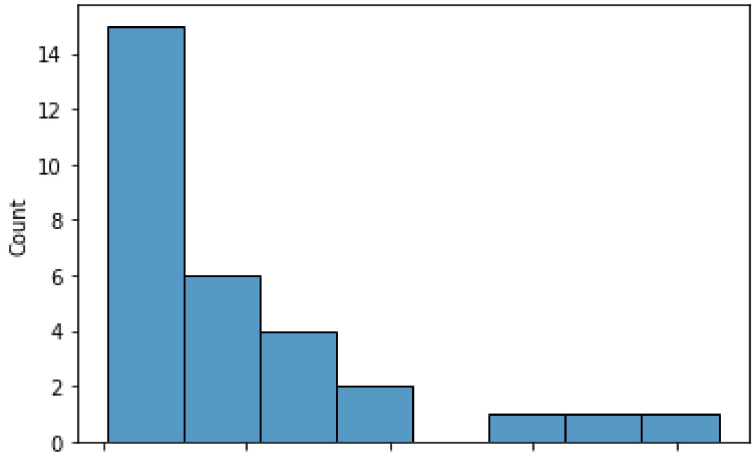
```
import seaborn as sns
sns.kdeplot(porc_enf)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3d5aa25cd0>
```



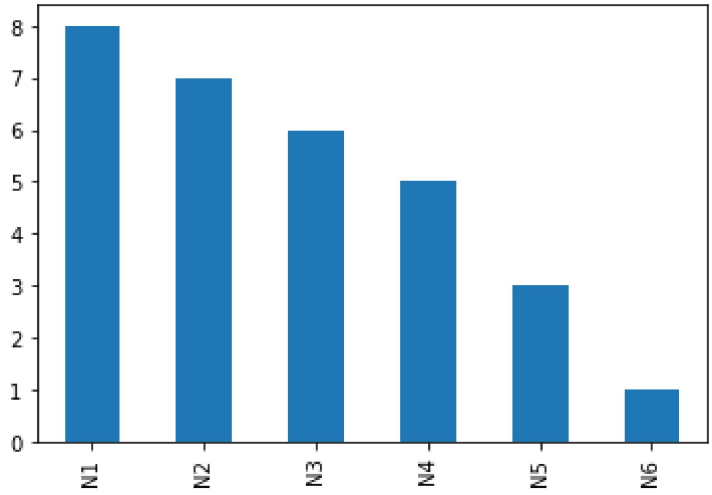

```
sns.histplot(porc_enf)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3d5a8dfad0>
```




```
1 cat_enf_serie.value_counts().plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3d5a552850>
```




```
1 porc_enf.mean()
```

```
9.469688297296576
```

```
1 np.median(porc_enf)
```

```
5.242795098033749
```

```
1 porc_enf_serie = pd.Series(porc_enf)
```

```
1 porc_enf_serie.quantile([0.24, 0.75, 0.90, 0.95])
```

```
0.24     2.447436
0.75    11.009624
0.90    21.662101
0.95    34.216986
dtype: float64
```

```
cont = cat_enf_serie.value_counts()

# Con el punto de corte real
frec1 = cont*[2.5, 5, 10, 20, 40, 80]
print(frec1.sum()/30)

# Con el punto medio
frec2 = cont*[1.25, 3.75, 7.50, 15, 30, 60]
print(frec2.sum()/30)

# Punto percentil 20%
frec3 = cont*[0.50, 3.0, 6.0, 12.0, 24.0, 48.0]
print(frec3.sum()/30)
```

```
13.833333333333334
10.208333333333334
8.033333333333333
```