PYTON INTERMEDIO

LECCIÓN NRO. 2 - PRIMER BIMESTRE

TEMA: MATPLOTLIB, NUMPY.LINALG

- Nombre: Renato Balcázar
- Fecha: 01-12-2020

Problema 1 (6 Pts.)

Utilizando el archivo "earthquake.csv" encontrar las gráficas siguientes:

- Trace los datos utilizando puntos como marcas para ver si podemos identificar patrones visibles (tiempo vs magnitud).
- Haz un histograma de la distribución de las magnitudes de los terremoto

Configure para la información de los ejes sea clara.

In [1]:
```
# Descargue los datos ejecutando la siguiente instrucción
!curl -L -o earthquakes.csv "https://risk-engineering.org/static/data/earthquakes-2017.csv"
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  280k  100  280k    0     0   138k      0  0:00:02  0:00:02 --:--:--  138k
```

In [26]:
```
import pandas as pd

data = pd.read_csv('earthquakes.csv', index_col=['time'], parse_dates=['time'])
data.tail()
```

Out[26]:

time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net	...	updated	place	type	horiz
2017-01-02 00:18:49.220000+00:00	-37.1568	52.3287	10.00	5.0	mb	NaN	39.0	9.272	0.71	us	...	2017-03-27T23:53:17.040Z	South Indian Ocean	earthquake	
2017-01-02 00:13:06.300000+00:00	-36.0365	51.9288	10.00	5.7	mwb	NaN	26.0	14.685	1.37	us	...	2017-03-27T23:53:17.040Z	Southwest Indian Ridge	earthquake	
2017-01-01 14:13:34.140000+00:00	3.5887	128.2355	10.00	5.3	mwb	NaN	56.0	5.364	0.61	us	...	2017-03-27T23:53:17.040Z	207km N of Tobelo, Indonesia	earthquake	
2017-01-01 14:12:05.720000+00:00	3.6359	128.5907	11.00	5.2	mwb	NaN	95.0	3.096	0.70	us	...	2017-03-27T23:53:17.040Z	220km NNE of Tobelo, Indonesia	earthquake	
2017-01-01 00:13:25.380000+00:00	2.8327	127.5786	78.93	5.0	mb	NaN	101.0	2.058	0.75	us	...	2017-03-27T23:53:16.040Z	131km NNW of Tobelo, Indonesia	earthquake	

5 rows × 21 columns

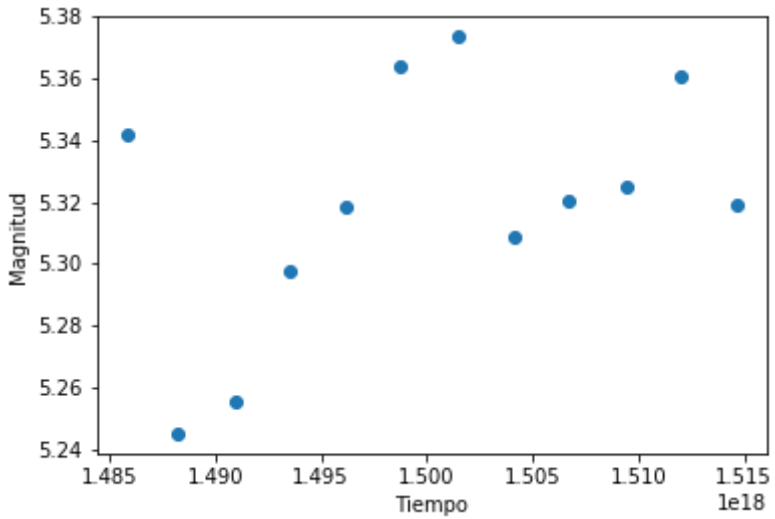
In [42]:
```
import matplotlib.pyplot as plt

# Argupamos los datos de acuerdo al timpo

new_data = data.resample('1m').mean()

new_data.head()
```

Out[42]:

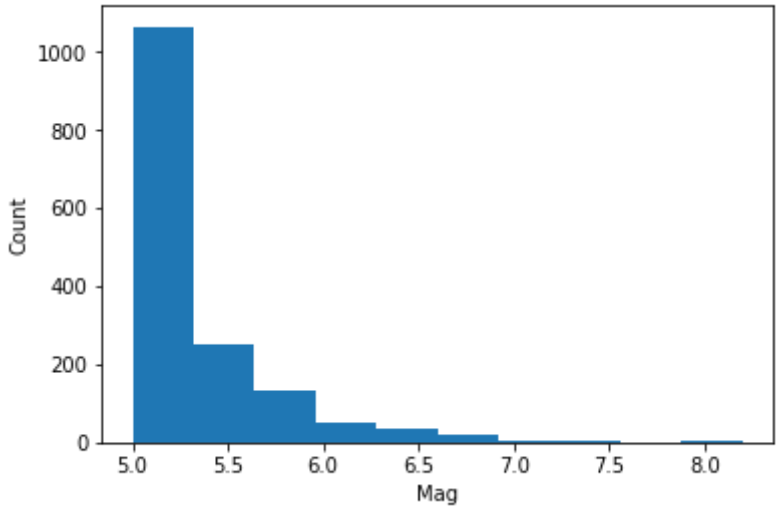
time	latitude	longitude	depth	mag	nst	gap	dmin	rms	horizontalError	depthError	magError	magNst
2017-01-31 00:00:00+00:00	-4.558548	64.640933	59.704641	5.341830	NaN	61.754967	3.998232	0.906863	7.696711	3.377124	0.066376	92.256410
2017-02-28 00:00:00+00:00	-3.495708	27.412058	82.696216	5.245045	NaN	60.809091	5.528809	0.901261	8.043636	3.390991	0.061920	94.470000
2017-03-31 00:00:00+00:00	-1.052508	50.881179	59.346370	5.255556	NaN	55.126866	4.718799	0.884519	7.650746	3.173333	0.063209	94.852713
2017-04-30 00:00:00+00:00	-3.696037	28.472892	43.886985	5.297794	NaN	62.807407	3.629452	0.862132	7.223704	2.957353	0.062015	92.529851
2017-05-31 00:00:00+00:00	-0.013367	-10.098331	44.769200	5.318667	NaN	67.643836	3.875486	0.895133	7.487075	2.817333	0.063865	76.794326

In [43]:
```
plt.plot(new_data.index, new_data['mag'], 'o')
plt.xlabel("Tiempo")
plt.ylabel("Magnitud")
plt.show()
```

```
/usr/lib/python3/dist-packages/matplotlib/cbook/__init__.py:2019: FutureWarning: Support for multi-dimension
al indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version.  Convert to a numpy
array before indexing instead.
  x[:, None]
/usr/lib/python3/dist-packages/matplotlib/axes/_base.py:249: FutureWarning: Support for multi-dimensional in
dexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version.  Convert to a numpy arra
y before indexing instead.
  x = x[:, np.newaxis]
/usr/lib/python3/dist-packages/matplotlib/axes/_base.py:251: FutureWarning: Support for multi-dimensional in
dexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future version.  Convert to a numpy arra
y before indexing instead.
  y = y[:, np.newaxis]
```



In [18]:
```
plt.hist(data.mag.dropna())
plt.xlabel("Magnitud")
plt.ylabel("Frecuencia")
plt.show()
```



Problema 2 (4 Pts)

Suponga que un vendedor de frutas vendió 20 mangos y 10 naranjas en un día por un total de $350. Al día siguiente vendió 17 mangos y 22 naranjas por 500. Si los precios de las frutas se mantuvieron sin cambios en ambos días, ¿cuál fue el precio de un mango y una naranja? Utilice linalg para resolver este problema.

In [4]:
```
import numpy as np

coef = np.array([[20, 10], [17,  22]])
result = np.array([350, 500])

respuesta = np.linalg.solve(coef, result)
respuesta
```

Out[4]:
```
array([10., 15.])
```

Respuesta: Los mangos valen 10 USD cada uno y las naranjas 15 USD