

Tarea Nro. 2 - PANDAS

- Nombre y Apellido: CRISTHIAN MICHAEL JIMENEZ JIMENEZ - Fecha: 19/nov/2020

En esta tarea se examinara datos de terremotos. Comience importando pandas, numpy y matplotlib.

Los datos de los terremotos están localizados en usgs_terremotos_2014.csv. Ni siquiera necesita descargalo, puede abrilo directamente con Pandas.

A continuación resuelva los siguientes ítems.

1) Use la función `read_csv` de Pandas directamente en esta url para abrirlo como un DataFrame

(No use ninguna opción especial). Mostrar las primeras filas y la información del marco de datos.

Debería haber visto que las fechas no se analizaron automáticamente en tipos de fecha y hora.

```
In [3]: import pandas as pd
import matplotlib.pyplot as plt

# Lectura de datos
data = pd.read_csv('usgs_terremotos_2014.csv', parse_dates=['time'], index_col='time',
data.head(5)
```

		latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net
	time										
	2014-01-31 23:53:37.000	60.252000	-152.7081	90.20	1.10		ml	NaN	NaN	NaN	0.2900 ak ak11155
	2014-01-31 23:48:35.452	37.070300	-115.1309	0.00	1.33		ml	4.0	171.43	0.34200	0.0247 nn nn00436
	2014-01-31 23:47:24.000	64.671700	-149.2528	7.10	1.30		ml	NaN	NaN	NaN	1.0000 ak ak11151
	2014-01-31 23:30:54.000	63.188700	-148.9575	96.50	0.80		ml	NaN	NaN	NaN	1.0700 ak ak11151
	2014-01-31 23:30:52.210	32.616833	-115.6925	10.59	1.34		ml	6.0	285.00	0.04321	0.2000 ci ci37171

2) Vuelva a leer los datos de tal manera que todas las columnas de

fechas se identifiquen como fechas y la identificación del terremoto se use como índice

In [4]: `print ("holo")`

holo

3) Obtener las estadísticas básicas de todas las columnas

In [7]: `data.describe()`

	latitude	longitude	depth	mag	nst	gap
count	120108.000000	120108.000000	120107.000000	120065.000000	59688.000000	94935.000000
mean	38.399579	-99.961402	28.375029	1.793958	17.878284	124.048978
std	21.938258	82.996858	62.215416	1.343466	14.911369	68.518595
min	-73.462000	-179.998900	-9.900000	-0.970000	0.000000	9.000000
25%	34.228917	-147.742025	4.100000	0.820000	8.000000	74.000000
50%	38.805300	-120.832000	9.200000	1.400000	14.000000	107.000000
75%	53.889500	-116.068100	22.880000	2.400000	22.000000	155.000000
max	86.651400	179.998000	697.360000	8.200000	365.000000	356.400000

4) Obtener los 20 terremotos más importantes por magnitud

Examina la estructura de la columna `place`. La información del país parece estar allí. ¿Cómo lo sacarías?

In [5]: `# ordenamos de forma de
data.sort_values(by = 'mag', ascending=False).head(20).place`

Out[5]: time place
2014-04-01 23:46:47.260 94km NW of Iquique, Chile
2014-06-23 20:53:09.700 19km SE of Little Sitkin Island, Alaska
2014-04-03 02:43:13.110 53km SW of Iquique, Chile
2014-04-12 20:14:39.300 93km SSE of Kirakira, Solomon Islands
2014-04-19 13:28:00.810 70km SW of Panguna, Papua New Guinea
2014-04-13 12:36:19.230 112km S of Kirakira, Solomon Islands
2014-10-14 03:51:34.460 74km S of Intipuca, El Salvador
2014-04-18 14:27:24.920 33km ESE of Petatlan, Mexico
2014-04-11 07:07:23.130 56km WSW of Panguna, Papua New Guinea
2014-11-15 02:31:41.720 154km NW of Kota Ternate, Indonesia
2014-11-01 18:57:22.380 144km NE of Ndoi Island, Fiji
2014-10-09 02:14:31.440 Southern East Pacific Rise
2014-06-29 07:52:55.170 154km NNW of Visokoi Island,
2014-08-03 00:22:03.680 Federated States of Micronesia region
2014-06-23 19:19:15.940 80km SSE of Raoul Island, New Zealand
2014-02-12 09:19:49.060 272km ESE of Hotan, China
2014-07-21 14:54:41.000 99km NNE of Ndoi Island, Fiji
2014-04-01 23:57:58.790 91km WNW of Iquique, Chile
2014-05-24 09:25:02.440 22km SSW of Kamariotissa, Greece
2014-07-07 11:23:54.780 4km W of Puerto Madero, Mexico
Name: place, dtype: object

5) Extraiga el país utilizando las funciones de datos de texto de Pandas

Agréguelo como una nueva columna al dataframe. (¿Es realmente solo un país? No, algunas filas tienen el nombre de un estado de EE. UU.) Corrija esto <https://www.geeksforgeeks.org/python-pandas-working-with-text-data/>.

Otra librería que puede ser útil <https://pypi.org/project/us/> para identificar si es un estado de EEUU es us

```
In [6]: import us
# data['pais'] = 'Nan'
# data.head()
data.head()
```

	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net	
time											
2014-01-31 23:53:37.000	60.252000	-152.7081	90.20	1.10		ml	NaN	NaN	NaN	0.2900	ak ak11155
2014-01-31 23:48:35.452	37.070300	-115.1309	0.00	1.33		ml	4.0	171.43	0.34200	0.0247	nn nn00436
2014-01-31 23:47:24.000	64.671700	-149.2528	7.10	1.30		ml	NaN	NaN	NaN	1.0000	ak ak11151
2014-01-31 23:30:54.000	63.188700	-148.9575	96.50	0.80		ml	NaN	NaN	NaN	1.0700	ak ak11151
2014-01-31 23:30:52.210	32.616833	-115.6925	10.59	1.34		ml	6.0	285.00	0.04321	0.2000	ci ci37171

```
In [62]: # 'casa' in ['house', 'hola', 'casa']
# import us
# us.states.STATES
lugar_place = str(data.place).split(',')
lugar_place
# Primero definimos la función para poder saber si un estado pertenece o no a 'us'
# def buscar_coindicencias(lugar_place):
#
#     return (x + 5) ** 2
```

```
Out[62]: ['time\n2014-01-31 23:53:37.000      26km S of Redoubt Volcano',
          'Alaska\n2014-01-31 23:48:35.452      32km S of Alamo',
```

```
' Nevada\n2014-01-31 23:47:24.000      12km NNW of North Nenana',
' Alaska\n2014-01-31 23:30:54.000      22km S of Cantwell',
' Alaska\n2014-01-31 23:30:52.210      10km WNW of Progreso',
' Mexico\n                                         ...
0:16.000          29km SW of Valdez',
' Alaska\n2014-12-01 00:09:39.000      102km SSE of Old Iliamna',
' Alaska\n2014-12-01 00:09:25.350      9km WNW of Cobb',
' California\n2014-12-01 00:05:54.000      57km NW of Ester',
' Alaska\n2014-12-01 00:04:05.000      78km WSW of Cordova,
' Alaska\nName: place',
' Length: 120108',
' dtype: object']
```

In [22]:

```
import us
us.states.STATES
```

Out[22]:

```
[<State:Alabama>,
 <State:Alaska>,
 <State:Arizona>,
 <State:Arkansas>,
 <State:California>,
 <State:Colorado>,
 <State:Connecticut>,
 <State:Delaware>,
 <State:Florida>,
 <State:Georgia>,
 <State:Hawaii>,
 <State:Idaho>,
 <State:Illinois>,
 <State:Indiana>,
 <State:Iowa>,
 <State:Kansas>,
 <State:Kentucky>,
 <State:Louisiana>,
 <State:Maine>,
 <State:Maryland>,
 <State:Massachusetts>,
 <State:Michigan>,
 <State:Minnesota>,
 <State:Mississippi>,
 <State:Missouri>,
 <State:Montana>,
 <State:Nebraska>,
 <State:Nevada>,
 <State>New Hampshire>,
 <State>New Jersey>,
 <State>New Mexico>,
 <State>New York>,
 <State:North Carolina>,
 <State:North Dakota>,
 <State:Ohio>,
 <State:Oklahoma>,
 <State:Oregon>,
 <State:Pennsylvania>,
 <State:Rhode Island>,
 <State:South Carolina>,
 <State:South Dakota>,
 <State:Tennessee>,
 <State:Texas>,
 <State:Utah>,
 <State:Vermont>,
 <State:Virginia>,
 <State:Washington>,
 <State:West Virginia>,
```

```
<State:Wisconsin>,
<State:Wyoming>]
```

In [16]: `'Alaska' in us.states.STATES`

Out[16]: `False`

In [31]:

```
# str(us.states.STATES)
# Lugar = '26km S of Redoubt Volcano, Alaska'.split(',')
# Lugar

# 'Washington' in str(us.states.STATES)

def comprobar_estado(valor):
    valor= str(valor).split(',')
    place = valor[-1]
    if (place [0] == ' '):
        place= place[1::]
    if (us.states.lookup(place)):
        place= 'US'
    else:
        place_uno = place.split()
        for n in place_uno:
            if(us.states.lookup(n)):
                place = 'US'
    return place
country = data.place.apply(comprobar_estado)
data ["states"]= country
data.head()
```

Out[31]:

	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net
time										

**2014-01-31
23:53:37.000** 60.252000 -152.7081 90.20 1.10 ml NaN NaN NaN 0.2900 ak ak11155

**2014-01-31
23:48:35.452** 37.070300 -115.1309 0.00 1.33 ml 4.0 171.43 0.34200 0.0247 nn nn00436

**2014-01-31
23:47:24.000** 64.671700 -149.2528 7.10 1.30 ml NaN NaN NaN 1.0000 ak ak11151

**2014-01-31
23:30:54.000** 63.188700 -148.9575 96.50 0.80 ml NaN NaN NaN 1.0700 ak ak11151

**2014-01-31
23:30:52.210** 32.616833 -115.6925 10.59 1.34 ml 6.0 285.00 0.04321 0.2000 ci ci37171



6) Encuentra los 10 países con el mayor número de terremotos

```
In [34]: print ("Los paises con mayor indice de terremos mas fuertes:")
numero = data[data.type == 'earthquake'].states.value_counts()
for pais in numero[0:10].index:
    print(pais)
```

Los paises con mayor indice de terremos mas fuertes:

US
Indonesia
Papua New Guinea
Chile
Japan
New Zealand
Philippines
Mexico
Fiji
Canada

7) Encuentra los 10 principales países donde ocurrieron los terremotos más fuertes y más débiles

```
In [39]: debiles = data[data.type == 'earthquake'].groupby('states').mean().sort_values('mag')
print("TERRMOTOS LEVES:")
for pais in debiles.iloc[0:10,:].index:
    print(pais)
```

TERRMOTOS LEVES:

Off the south coast of Northwest Africa
Sierra Leone
US
Canada
Puerto Rico region
south of the Aleutian Islands
Tennessee-Virginia border region
Saint Martin
Saint Eustatius and Saba
Dominican Republic

```
In [40]: print("TERREMOTOS FUERTES:")
for pais in debiles.iloc[-11:-1,:].index:
    print(pais)
```

TERREMOTOS FUERTES:

Palau
Southern East Pacific Rise
Pacific-Antarctic Ridge
North Korea
Saint Lucia
Drake Passage
Malawi
Easter Island region
Barbados
Bouvet Island region

8) Cree un conjunto de datos filtrado que solo tenga terremotos de magnitud 4 o mayores

```
In [44]: mayores_cuatro=data[data.mag>=4]
mayores_cuatro.head()
```

Out[44]:

localhost:8888/lab

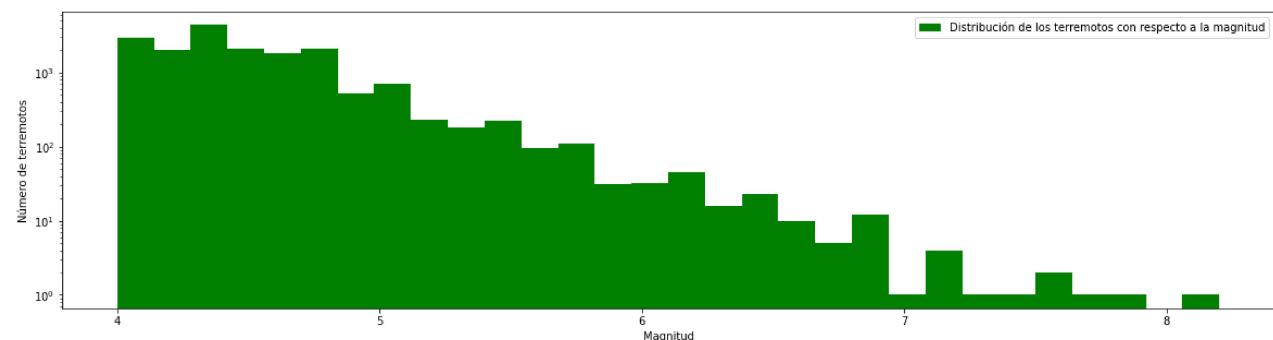
	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	net	id
time											
2014-01-31 23:08:03.660	-4.9758	153.9466	110.18	4.2		mb	NaN	98.0	1.940	0.61	us usc000mqlp
2014-01-31 22:54:32.970	-28.1775	-177.9058	95.84	4.3		mb	NaN	104.0	1.063	1.14	us usc000mqln
2014-01-31 22:49:49.740	-23.1192	179.1174	528.34	4.4		mb	NaN	80.0	5.439	0.95	us usc000mqls
2014-01-31 22:19:44.330	51.1569	-178.0910	37.50	4.2		mb	NaN	NaN	NaN	0.83	us usc000mf1x
2014-01-31 21:56:44.320	-4.8800	153.8434	112.66	4.3		mb	NaN	199.0	1.808	0.79	us usc000mqlm



9) Analice la distribución de las magnitudes del terremoto en la distribución filtrada

Haga un histograma del conteo del terremoto versus la magnitud. Asegúrese de usar una escala logarítmica.

```
In [50]: import matplotlib.pyplot as plt
plt.figure(figsize=(20, 5))
plt.hist(mayores_cuatro.mag, label="Distribución de los terremotos con respecto a la magnitud")
plt.yscale('log', nonpositive='clip')
plt.ylabel("Número de terremotos")
plt.xlabel("Magnitud")
plt.legend()
plt.show()
```



10) Visualice la ubicación de los terremotos haciendo un diagrama de dispersión de su latitud y longitud.

Usa los datos filtrados. Coloréalo por magnitud.

Ej. plt.scatter(x, y, s=s, c=c, cmap=plt.cm.Oranges)

Con s y c podemos modificar el tamaño y el color respectivamente. Para el color, a cada valor numérico se le asigna un color a través de un mapa de colores; ese mapa se puede cambiar con el argumento cmap. Esa correspondencia se puede visualizar llamando a la función colorbar.

```
N = 100
x = np.random.randn(N)
y = np.random.randn(N)
s = 50 + 50 * np.random.randn(N)
c = np.random.randn(N)

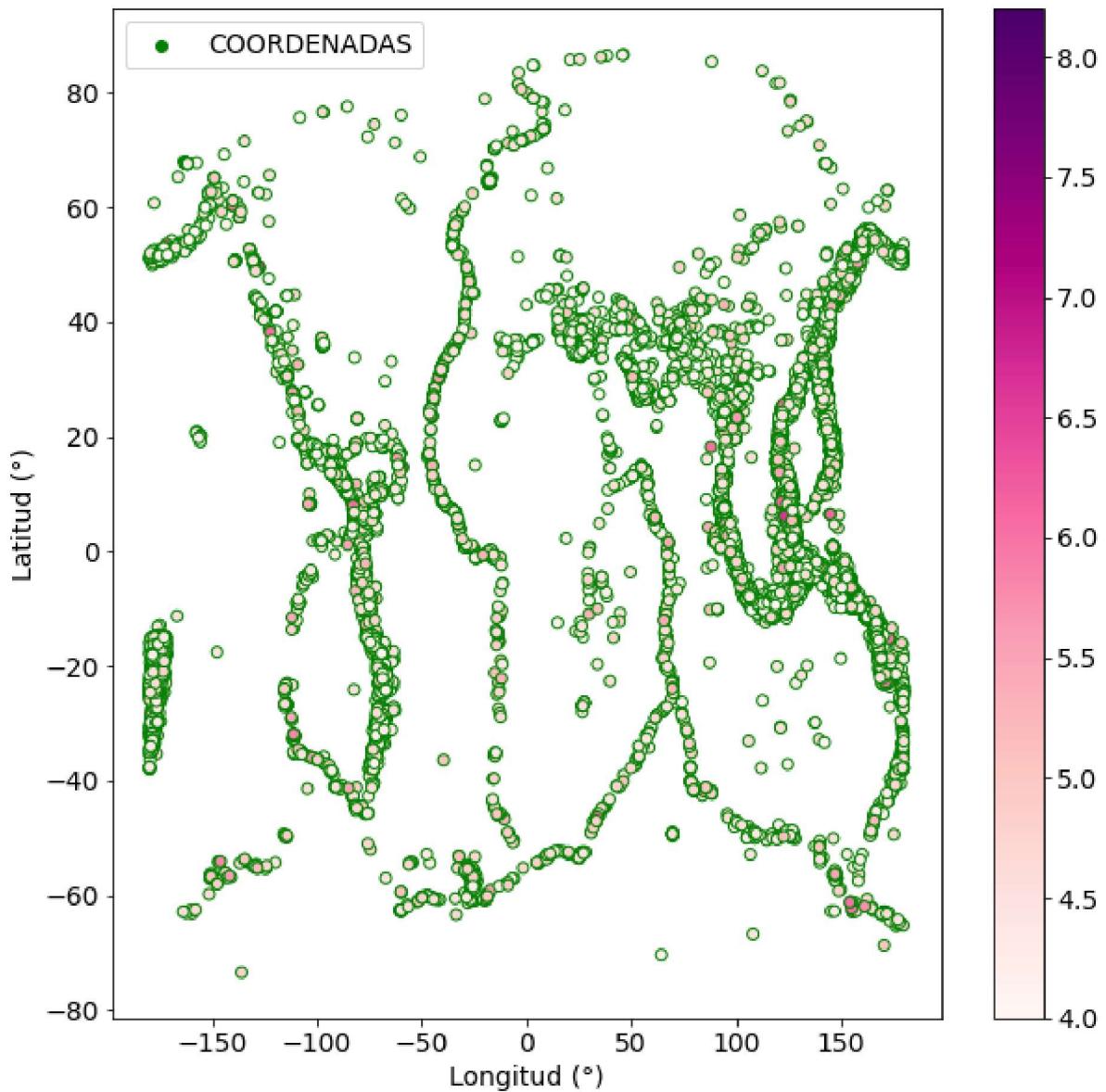
plt.scatter(x, y, s=s, c=c, cmap=plt.cm.Blues)
plt.colorbar()
```

Ref. adicional para colores:

<https://github.com/lisantiago/PythonBasico/raw/d36d9571a1ff6a2df8364a9055f71d70eddd9151/Numpy>

```
In [82]: plt.figure(figsize=(10,10))
plt.rcParams['font.size'] = 14
plt.scatter(mayores_cuatro.longitude, mayores_cuatro.latitude, label="COORDENADAS", c=m
plt.colorbar()
plt.xlabel("Longitud (°)")
plt.ylabel("Latitud (°)")
plt.legend()
plt.show()
```

c:\users\mishu\appdata\local\programs\python\python37\lib\site-packages\ipykernel_launcher.py:3: MatplotlibDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed two minor releases later
This is separate from the ipykernel package so we can avoid doing imports until



11) Haz lo mismo para la profundidad

```
In [77]: colores = ["#00cc44", # Verde
                 "#ff7700", # Naranja
                 "#ff0000" # Rojo
                ]
plt.figure(figsize=(10,10))
plt.rcParams['font.size'] = 14
plt.scatter(mayores_cuatro.longitude, mayores_cuatro.latitude, label="COORDENADAS", c=mayores_cuatro['profundidad'])
plt.xlabel("LONGITUD")
plt.ylabel("LATITUD")
plt.colorbar()
plt.legend()
plt.show()
```

c:\users\mishu\appdata\local\programs\python\python37\lib\site-packages\ipykernel_launcher.py:7: MatplotlibDeprecationWarning: Case-insensitive properties were deprecated in 3.3 and support will be removed two minor releases later
import sys

