

Seismes

February 26, 2018

1 A. Extraction des données

1.1 1.

```
In [1]: import pandas as pa
        from datetime import datetime
```

```
In [2]: T=pa.read_csv("http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_month.csv",s
```

1.2 2.

```
In [3]: T.head()
```

```
Out[3]:
```

	time	latitude	longitude	depth	mag	magType	nst	\
0	2018-02-26T16:25:54.470Z	33.578667	-116.635000	12.65	0.69	ml	18.0	
1	2018-02-26T16:25:06.770Z	38.801834	-122.781998	2.65	1.65	md	18.0	
2	2018-02-26T16:16:49.370Z	38.804001	-122.817665	2.18	0.96	md	14.0	
3	2018-02-26T16:16:31.950Z	37.314499	-121.684670	7.47	1.85	md	28.0	
4	2018-02-26T16:02:20.620Z	32.734333	-117.413833	3.61	1.67	ml	17.0	

	gap	dmin	rms	...	updated	\
0	79.0	0.01253	0.19	...	2018-02-26T16:29:37.050Z	
1	76.0	0.01207	0.04	...	2018-02-26T16:39:03.527Z	
2	107.0	0.01022	0.05	...	2018-02-26T16:43:03.081Z	
3	67.0	0.04313	0.05	...	2018-02-26T16:41:09.153Z	
4	239.0	0.23760	0.24	...	2018-02-26T16:06:10.284Z	

	place	type	horizontalError	depthError	\
0	4km NE of Anza, CA	earthquake	0.44	0.55	
1	4km NW of The Geysers, CA	earthquake	0.27	0.92	
2	6km WNW of The Geysers, CA	earthquake	0.34	0.50	
3	14km ESE of Alum Rock, CA	earthquake	0.16	0.40	
4	18km SW of La Jolla, CA	earthquake	1.70	4.14	

	magError	magNst	status	locationSource	magSource
0	0.260	14.0	automatic	ci	ci
1	0.100	5.0	automatic	nc	nc
2	0.200	4.0	automatic	nc	nc

```

3      0.130    20.0  automatic          nc      nc
4      0.217    25.0  automatic          ci      ci

```

[5 rows x 22 columns]

1.3 3. Nombre de séismes

In [4]: `len(T)`

Out[4]: 9343

2 B. Etude du temps d'attente entre 2 séismes

Fonction datetime du module datetime La fonction **datetime** prend 6 paramètres (entiers) qui sont années - mois - jour - heure - minute -seconde

Exemple : le 3 mars 2018 à 15h30 correspond à

In [5]: `datetime(2018,3,14,15,30,0)`

Out[5]: `datetime.datetime(2018, 3, 14, 15, 30)`

Remarque : on peut effectuer la différence entre deux datetime.

2.1 1. Conversion d'un temps UTC en format datetime

Chaque date contenue dans la colonne **timing** du tableau T est une chaîne de caractères, écrite au format **UTC**

2.1.1 a) Fonction conversion qui transforme une date s (au format UTC) en la liste des 6 paramètres de datetime

```

In [6]: def conversion(s):
        k=0
        mot=''
        while s[k]!='.':
            if s[k]!='-' and s[k]!='T' and s[k]!=':':
                mot=mot+s[k]
            else:
                mot=mot+'-'
            k=k+1
        Liste_string=mot.split('-')
        return [int(a) for a in Liste_string]

```

2.1.2 b) Temps du dernier séisme

In [7]: `conversion(T.loc[0,'time'])`

Out[7]: [2018, 2, 26, 16, 25, 54]

```
In [8]: t=conversion(T.loc[0,'time'])
        datetime(t[0],t[1],t[2],t[3],t[4],t[5])
```

```
Out[8]: datetime.datetime(2018, 2, 26, 16, 25, 54)
```

2.1.3 c) Fonction horaire qui pour une date (chaîne de caractères écrit au format UTC de la colonne timing) renvoie son datetime

```
In [9]: def horaire(UTC):
        t=conversion(UTC)
        return datetime(t[0],t[1],t[2],t[3],t[4],t[5])
```

2.2 2. Création du tableau T1 ne concernant que les séismes d'amplitudes >5

```
In [10]: T1=T[T['mag']>5]
```

```
In [11]: T1.head()
```

```
Out[11]:
```

	time	latitude	longitude	depth	mag	magType	nst	\
12	2018-02-26T15:17:59.470Z	-6.4950	143.2550	13.81	6.2	mww	NaN	
23	2018-02-26T13:34:53.170Z	-2.7690	126.7076	6.30	6.0	mww	NaN	
32	2018-02-26T12:34:54.350Z	-6.3147	143.3448	26.70	5.2	mb	NaN	
36	2018-02-26T11:56:07.450Z	-6.5004	143.5538	13.33	5.7	mww	NaN	
67	2018-02-26T08:26:59.340Z	-6.4274	143.2706	23.64	6.0	mww	NaN	

	gap	dmin	rms	...	updated	\
12	19.0	4.828	0.90	...	2018-02-26T16:42:51.013Z	
23	38.0	3.578	0.72	...	2018-02-26T15:39:17.267Z	
32	38.0	4.869	0.94	...	2018-02-26T13:20:33.040Z	
36	20.0	4.591	0.91	...	2018-02-26T14:42:34.040Z	
67	39.0	4.856	1.01	...	2018-02-26T14:20:36.040Z	

	place	type	horizontalError	\
12	56km SW of Mendi, Papua New Guinea	earthquake	4.1	
23	46km NE of Airbuaya, Indonesia	earthquake	6.8	
32	37km WSW of Mendi, Papua New Guinea	earthquake	9.5	
36	37km SSW of Mendi, Papua New Guinea	earthquake	5.0	
67	50km WSW of Mendi, Papua New Guinea	earthquake	7.4	

	depthError	magError	magNst	status	locationSource	magSource
12	3.1	0.048	41.0	reviewed	us	us
23	3.0	0.046	45.0	reviewed	us	us
32	4.1	0.055	108.0	reviewed	us	us
36	4.1	0.073	18.0	reviewed	us	us
67	4.8	0.073	18.0	reviewed	us	us

```
[5 rows x 22 columns]
```

```
In [12]: T1.describe()
```

```

Out [12]:
    latitude longitude depth mag nst gap \
count 102.000000 102.000000 102.000000 102.000000 0.0 102.000000
mean 1.122188 29.813399 59.838137 5.474510 NaN 67.676471
std 31.699477 123.498268 125.432734 0.413554 NaN 45.158283
min -65.861700 -179.452100 4.300000 5.100000 NaN 15.000000
25% -15.730350 -78.541425 10.000000 5.200000 NaN 35.000000
50% -3.484950 94.677750 13.880000 5.300000 NaN 52.000000
75% 18.135475 142.646550 37.305000 5.600000 NaN 87.750000
max 85.324700 178.827500 631.160000 7.500000 NaN 247.000000

    dmin rms horizontalError depthError magError \
count 102.000000 102.000000 102.000000 102.000000 102.000000
mean 4.496706 0.927647 7.589216 3.250000 0.065873
std 5.236466 0.230390 2.386177 1.733065 0.019421
min 0.069000 0.480000 2.600000 0.700000 0.026000
25% 1.323500 0.752500 6.000000 1.900000 0.052250
50% 2.480500 0.890000 7.400000 2.450000 0.064500
75% 5.532250 1.040000 8.700000 4.100000 0.073000
max 32.330000 1.510000 15.200000 8.400000 0.127000

    magNst
count 102.000000
mean 64.774510
std 85.256504
min 6.000000
25% 19.000000
50% 32.000000
75% 73.250000
max 492.000000

```

2.3 3. Fonction attentes qui pour un tableau de données Tab, renvoie la liste des temps d'attente entre 2 séismes

```

In [13]: def attentes(Tab):
    L=[]
    Liste_Temps=list(Tab['time'])
    for k in range(1,len(Liste_Temps)):
        d1=horaire(Liste_Temps[k-1])
        d2=horaire(Liste_Temps[k])
        d=d1-d2
        L.append(d.seconds)
    return L

```

```

In [14]: L=attentes(T1)

```

2.4 4. Modélisation

Soit variable S qui pour un séisme choisi au hasard, associe le temps d'attente (exprimé en seconde) du prochain séisme. Quelle loi de probabilité peut-on associer à S ?

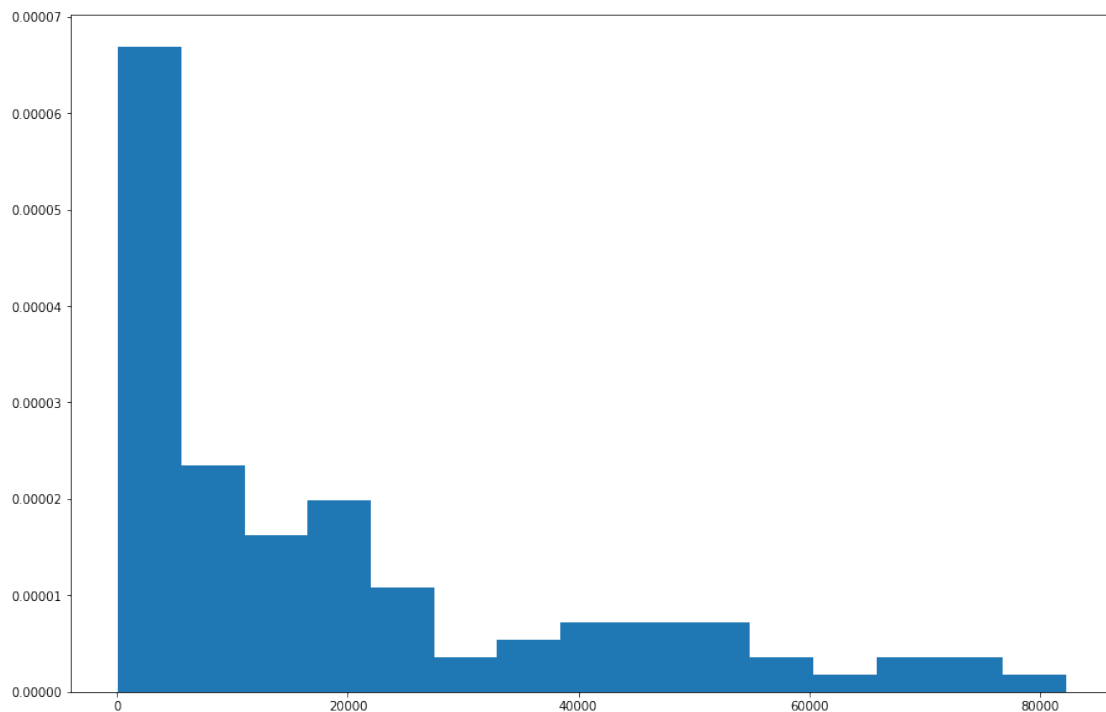
2.4.1 a) Histogramme des fréquences associées à L

```
In [15]: import pylab as pl
```

```
In [16]: fig=pl.figure(figsize=(15,10))  
         pl.hist(L,bins=15,normed=True)
```

```
Out[16]: (array([ 6.68497507e-05,  2.34877502e-05,  1.62607502e-05,  
                  1.98742502e-05,  1.08405001e-05,  3.61350004e-06,  
                  5.42025005e-06,  7.22700007e-06,  7.22700007e-06,  
                  7.22700007e-06,  3.61350004e-06,  1.80675002e-06,  
                  3.61350004e-06,  3.61350004e-06,  1.80675002e-06]),  
         array([ 84.,  5564., 11044., 16524., 22004., 27484., 32964.,  
                38444., 43924., 49404., 54884., 60364., 65844., 71324.,  
                76804., 82284.]),  
         <a list of 15 Patch objects>)
```

```
In [17]: pl.show()
```



2.4.2 b) Une proposition de densité de probabilité

```
In [18]: import numpy as np
```

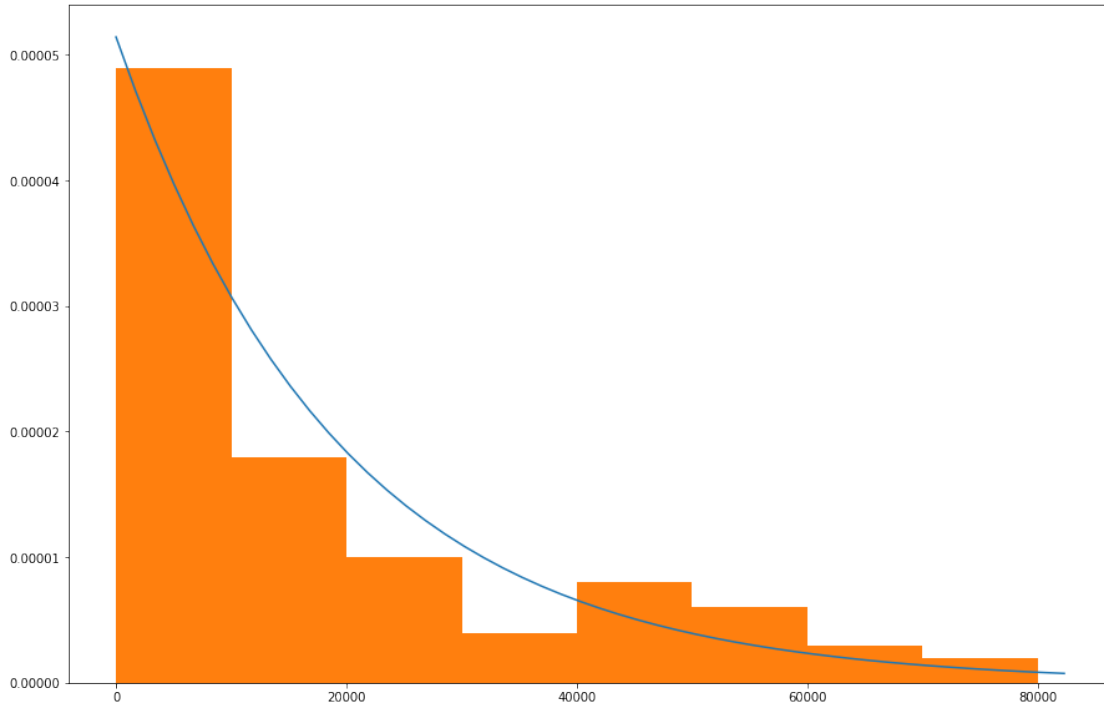
```
In [19]: m=np.mean(L)  
         a=1/m
```

```

def densité(a,x):
    return a*np.exp(-a*x)
X=np.linspace(0,max(L))

pl.figure(figsize=(15,10))
pl.plot(X,densité(a,X))
pl.hist(L,[10000*k for k in range(9)],normed=True)
pl.show()

```



- Les valeurs de **L** forme un échantillon de plusieurs réalisations de **S**.
- L'histogramme des fréquences associé à cet échantillon semble être ajusté par la densité de probabilité f de la loi exponentielle de paramètres $a = \frac{1}{m}$ avec m valeur moyenne des temps contenus dans **L** :

$$f(x) = a \times e^{(-ax)}$$

2.4.3 Valeurs des paramètres :

In [20] : a

Out [20] : 5.1426526005783193e-05

In [21] : m

Out [21] : 19445.217821782178

2.4.4 Conversion de m en heure :

```
In [22]: m/60/60
```

```
Out[22]: 5.4014493949394939
```

2.4.5 -----

2.4.6 Coin Prof :

- On émet l'hypothèse que S suit une loi exponentielle de paramètre a.
- La question est de déterminer si oui ou non on accepte cette hypothèse avec un risque de se tromper fixé à 5%.
- On effectue pour cela un test de Kolmogorov Smirnov afin de mener à bien ce test d'hypothèse :

```
In [23]: import scipy.stats as sc
         sc.kstest(L, 'expon', args=(0,m))
```

```
Out[23]: KstestResult(statistic=0.12355181883501992, pvalue=0.084014531839118334)
```

Puisque la p-value est supérieure à 5%, on accepte l'hypothèse émise sur la loi de probabilité.

2.4.7 -----

2.5 2. Probabilité pour ne pas observer de nouveau séisme durant au moins un jour complet :

```
In [24]: np.exp(-a*24*60*60)
```

```
Out[24]: 0.011757642234223138
```

3 C. Carte des séismes

```
In [25]: import folium
```

```
In [26]: carte = folium.Map(location=[0,0],tiles='Mapbox Control Room', zoom_start=2)
```

```
In [27]: carte
```

```
Out[27]: <folium.folium.Map at 0x1a17300ac8>
```

```
In [28]: for lat,long,mag in zip(T1['latitude'],T1['longitude'],T1['mag']):
         folium.CircleMarker(location=[lat,long],
                             fill=True,
                             fill_color='red',
                             fill_opacity=0.5,
                             color='yellow',
                             popup=str(mag),
                             radius=1*np.sqrt(mag)).add_to(carte)
```

```
In [29]: carte
```

```
Out[29]: <folium.folium.Map at 0x1a17300ac8>
```

```
In [30]: #carte.save('C:/Lycee(henrimoissan)/Prof (henri-moissan)/Annee 2017/philippe/Activite/S
```

3.0.1 Petit Bonus : détermination du seisme d'amplitude maximale

```
In [31]: T2=T1[T1['mag']==max(T1['mag'])]  
T2
```

```
Out[31]:
```

	time	latitude	longitude	depth	mag	magType	nst	\
207	2018-02-25T17:44:44.380Z	-6.1488	142.7663	35.0	7.5	mww	NaN	
	gap	dmin	rms	...		updated	\	
207	30.0	5.423	1.16	...		2018-02-26T15:54:54.861Z		
					place	type	horizontalError	\
207					89km SSW of Porgera, Papua New Guinea	earthquake	7.1	
	depthError	magError	magNst	status	locationSource	magSource		
207	1.9	0.071	19.0	reviewed		us	us	

[1 rows x 22 columns]

```
In [32]: folium.CircleMarker(location=[T.loc[207,'latitude'],T.loc[207,'longitude']],  
                             fill=True,  
                             fill_color='red',  
                             fill_opacity=0.5,  
                             color='blue',  
                             popup=str(T.loc[207,'mag']),  
                             radius=1*np.sqrt(T.loc[207,'mag'])).add_to(carte)
```

```
Out[32]: <folium.features.CircleMarker at 0x1a1759bef0>
```

```
In [33]: carte
```

```
Out[33]: <folium.folium.Map at 0x1a17300ac8>
```

4 D. Activité sismique en Insulide

Les pays de l'Asie du sud-est (Indonésie, Singapour, etc) appartiennent à la zone (appelée **In-sulide**) ayant : - une latitude comprise entre -13° à 15° - une longitude comprise entre 90° et 170°

Pour représenter cette zone sur la carte, en guise de bonus, voici les instructions à écrire :

```
In [34]: ls = folium.PolyLine(  
         locations=[[15, 90], [15, 170], [-13, 170], [-13, 90], [15, 90]],  
         color='purple'  
         )  
ls.add_to(carte)
```



```
Out [34]: <folium.features.PolyLine at 0x1a175f9320>
```

```
In [35]: carte
```

```
Out [35]: <folium.folium.Map at 0x1a17300ac8>
```

```
In [36]: #carte.save('C:/Lycee(henrimoissan)/Prof (henri-moissan)/Annee 2017/philippe/Activite/S
```

4.1 1. Formation du tableau T2 qui ne contient que les lignes de T1, associées aux séismes qui se sont produits en *Insulide* :

```
In [37]: Insulide=(T1['latitude']>-13)&(T1['latitude']<15)&(T1['longitudo']>90)&(T1['longitudo']
```

```
In [38]: T2=T1[Insulide]
```

4.2 2. Proportion de séismes produits en Insulide

```
In [39]: len(T2)
```

```
Out [39]: 36
```

```
In [40]: proportion=len(T2)/len(T1)
```

```
In [41]: proportion
```

```
Out [41]: 0.35294117647058826
```

4.3 3. Que représente l'aire de zone par rapport à celle de la Terre ?

```
In [42]: Surface=(170-90)/360*(15-(-13))/180
```

```
In [43]: Surface
```

```
Out [43]: 0.0345679012345679
```

5 E. Nombre de séismes sur une période

5.1 1. Proportion de séisme par seconde :

```
In [44]: p=len(T1)/(30*24*60*60)
```

```
In [45]: p
```

```
Out [45]: 3.935185185185185e-05
```

5.2 2. Probabilité qu'il se produise *au moins* un séisme de magnitude >5 en l'espace d'un séisme

Une journée comporte $n = 24 \times 60 \times 60 = 86400$ secondes. Pour chacune d'elles, il y a deux issues : - Issue "succès" : il se produit un séisme avec une probabilité p - Issue "echec" : il ne se produit pas de séisme.

On répète cette épreuve n fois de suite, et on considère l'indépendance entre deux séismes. On obtient donc un processus de Bernoulli d'ordre n .

- La variable aléatoire X qui compte le nombre de séismes qui se produisent sur un jour complet suit une loi binomiale de paramètres n et p .
- Il s'agit de calculer

$$P(X \geq 1)$$

```
In [46]: 1-(1-p)**(24*60*60)
```

```
Out[46]: 0.9666289626336273
```

5.3 3. A partir de quelle heure, avons-nous plus d'une chance sur deux d'observer au moins un séisme (de magnitude >5) dans le monde ?

```
In [47]: for k in range(25):  
         print(k, 1-(1-p)**(60*60*k))
```

```
0 0.0  
1 0.1320919075158461  
2 0.24673554300051725  
3 0.34623568198946697  
4 0.43259265782127443  
5 0.5075425759881587  
6 0.5725922164962226  
7 0.6290493259063563  
8 0.6780489080416747  
9 0.7205760419052594  
10 0.7574856855356216  
11 0.789519863933119  
12 0.8173225866003881  
13 0.8414527945964037  
14 0.8623955973894714  
15 0.8805720254128746  
16 0.8963474943868419  
17 0.9100391515720809  
18 0.9219222516426686  
19 0.9322356903577308  
20 0.9411868072798726  
21 0.9489555540933713  
22 0.9556981123212672  
23 0.9615500331713038  
24 0.9666289626336273
```

```
In [48]: k=0
         while 1-(1-p)**(60*60*k)<1/2:
             k=k+1
         print(k)
```

5

Il faut attendre 5 heures.