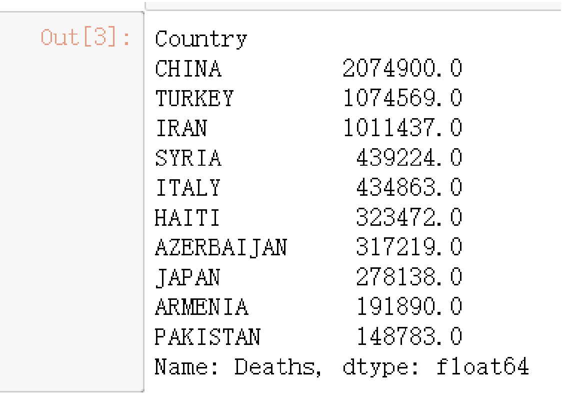
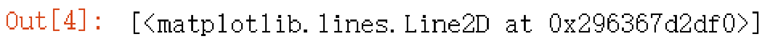
PS2_1

```
import pandas as pd
import numpy as np

#cite:https://blog.csdn.net/IT_SoftEngineer/article/details/107325062
#1.1
#思路：先对国家进行分类并进行加和，然后根据死亡人数进行降序排序，输出前十行即可

Sig_Eqs= pd.read_csv('earthquakes-2021-10-25_21-26-58_+0800.tsv',encoding='gbk',sep='\t')
Sig_Eqs.groupby(['Country']).sum().sort_values('Deaths', ascending=False)['Deaths'].head(10)
```

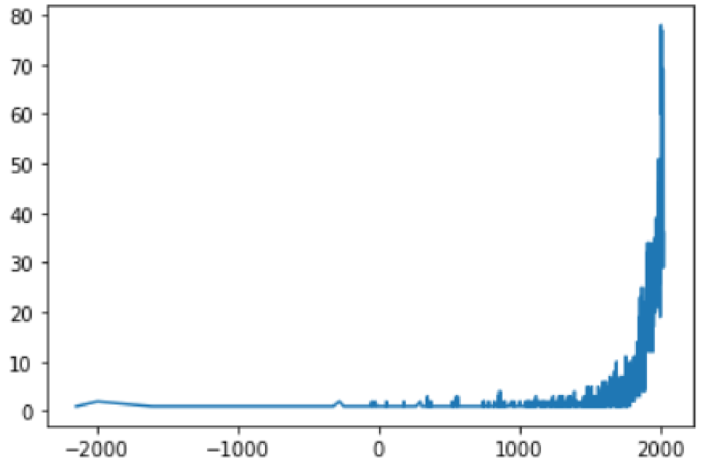


```
Out[3]: Country
        CHINA         2074900.0
        TURKEY        1074569.0
        IRAN          1011437.0
        SYRIA          439224.0
        ITALY          434863.0
        HAITI          323472.0
        AZERBAIJAN     317219.0
        JAPAN          278138.0
        ARMENIA        191890.0
        PAKISTAN       148783.0
        Name: Deaths, dtype: float64
```

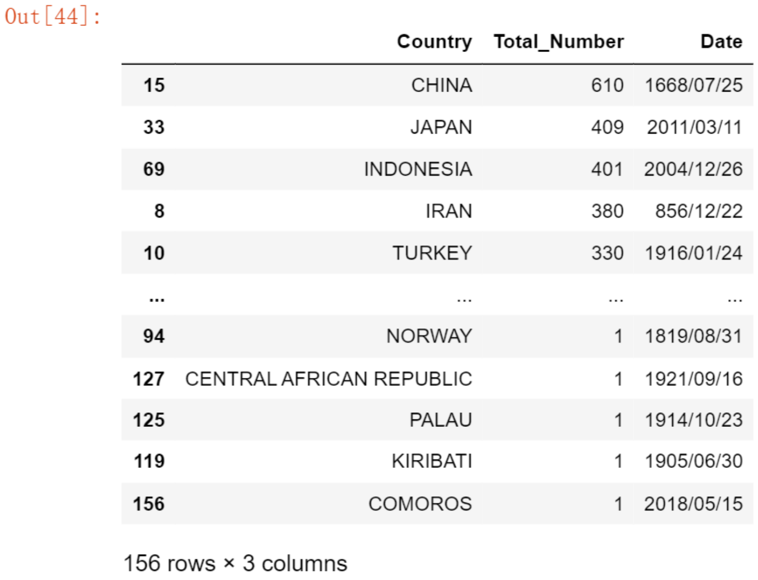
```
#1.2
#思路：先对表格按照震级大于六的数据进行筛选，再增加一列'Total_Number'，每一行赋值
#为 1，便于后面计算总数，在对年份进行分类并进行加和数据，并将其赋值与一个新的
#dataframe，最后进行画图

import matplotlib.pyplot as plt
Sig_Eqs.loc[(Sig_Eqs['Mag'] >6)]
Sig_Eqs['Total_Number']= 1;
Sig_Eqs1= (Sig_Eqs.groupby(['Year']).sum()).reset_index()
plt.plot (Sig_Eqs1['Year'],Sig_Eqs1['Total_Number'])
```



```
Out[4]: [<matplotlib.lines.Line2D at 0x296367d2df0>]
```



```
#1.3

#思路：先对年月日所在列进行类型转换，然后整合到‘Date‘列表示日期，接着定义函数
#CountEq_LargestEq (C,Sig_Eqs)，函数内部思路主要为先找出含有特定国家的所有行并进行
#加和，另外，将以国家为特定值筛选的dataframe赋值给新的dataframe，按照震级惊喜排
#序，输出第一行的date列的值。最后在原dataframe中找到所有的国家并建立一个dataframe
#（df），最后在df中加入两行空列，分别填入函数运行后的值，最后根据总数进行排序即可

Sig_Eqs['cYear']=Sig_Eqs['Year'].astype('Int64').astype(str)
Sig_Eqs['cMo']= Sig_Eqs['Mo'].astype('Int64').astype(str)
Sig_Eqs['cDy']= Sig_Eqs['Dy'].astype('Int64').astype(str)

Sig_Eqs['cMo'][Sig_Eqs['Mo']<10] = '0' + Sig_Eqs['cMo'].astype(str)
Sig_Eqs['cDy'][Sig_Eqs['Dy']<10] = '0' + Sig_Eqs['cDy'].astype(str)
Sig_Eqs['Date'] = Sig_Eqs['cYear'] + '/' + Sig_Eqs['cMo'] + '/' + Sig_Eqs['cDy']

def CountEq_LargestEq (C,Sig_Eqs):

    Total_Number=Sig_Eqs.loc[(Sig_Eqs['Country']== C )].sum()['Total_Number']
    Sig_Eqs2=Sig_Eqs.loc[(Sig_Eqs['Country']==C )]
    Date=Sig_Eqs2.sort_values('Mag',ascending=False).head(1)['Date']
    Date=Date.values.tolist( )
    return Total_Number,Date[0]
#Zhan Yang explained to me what is asked in this problem

df = pd.DataFrame(Sig_Eqs['Country'].unique(),columns=['Country'])
df = df.dropna(axis=0)
df['Total_Number'] = 0
df['Date'] = None
for i in range(df.shape[0]):
    df['Total_Number'].iat[i],df['Date'].iat[i] = CountEq_LargestEq(df['Country'].iat[i],Sig_Eqs)

df=df.sort_values('Total_Number', ascending=False)
df
```



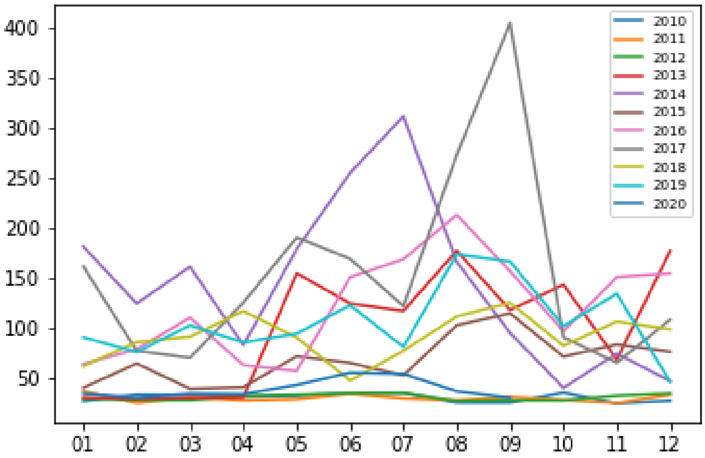
Out[44]:

	Country	Total_Number	Date
15	CHINA	610	1668/07/25
33	JAPAN	409	2011/03/11
69	INDONESIA	401	2004/12/26
8	IRAN	380	856/12/22
10	TURKEY	330	1916/01/24
...	...	...	...
94	NORWAY	1	1819/08/31
127	CENTRAL AFRICAN REPUBLIC	1	1921/09/16
125	PALAU	1	1914/10/23
119	KIRIBATI	1	1905/06/30
156	COMOROS	1	2018/05/15

156 rows × 3 columns

PS2_2

```
#思路：导入文档，将 WND 列中代表风速的数据提取出来赋值到新的一列中，在将月份与日
#期进行相同的操作。然后对'WND1'进行种类的转换（转为 int，用于计算）。最后使用 for 循
#环根据每一个年份进行筛选，并对月份进行分类然后对风速进行均值计算，最后画出每一
#年的月平均风速与月份的曲线即可。

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#cite:https://blog.csdn.net/Asher117/article/details/84346073/

wind= pd.read_csv('2281305.csv')
wind['WND1']= wind['WND'].map(lambda x:x.split(',')[3])
wind['month']= wind['DATE'].map(lambda x:x.split('-')[1])
wind['year']= wind['DATE'].map(lambda x:x.split('-')[0])
wind['WND2']=wind['WND1'].astype('int')
lst=['2010','2011','2012','2013','2014','2015','2016','2017','2018','2019','2020']
for a in lst:

    table=wind.loc[(wind['year'] == a )].groupby(['month'])['WND2'].mean().reset_index()
    plt.plot(table['month'],table['WND2'])
plt.legend(labels=['2010','2011','2012','2013','2014','2015','2016','2017','2018','2019','2020'],fontsize=7)
```

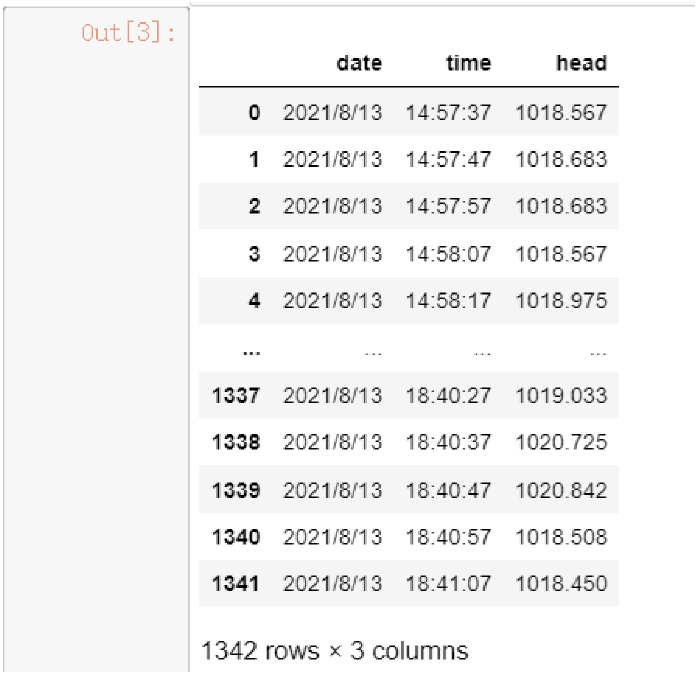


#规律：2013, 2014, 2016, 2017, 2019 在四月份前后月均风速有明显的上升，随后直到八月份
#风速波动上涨，在八月份之后开始回落。2010, 2011, 2012 年无明显的波动，在六月份前后
#有一个小的高峰。总的来说无明显统一趋势。

PS2_3

#思路：采取的数据为实验室的抽水试验数据，读取文件，使用 plot 函数对时间和水位变化进行绘图，随后对水位进行最大值、最小值、均值、方差及标准差进行计算

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#3.1
head= pd.read_csv('head.csv')
head
```

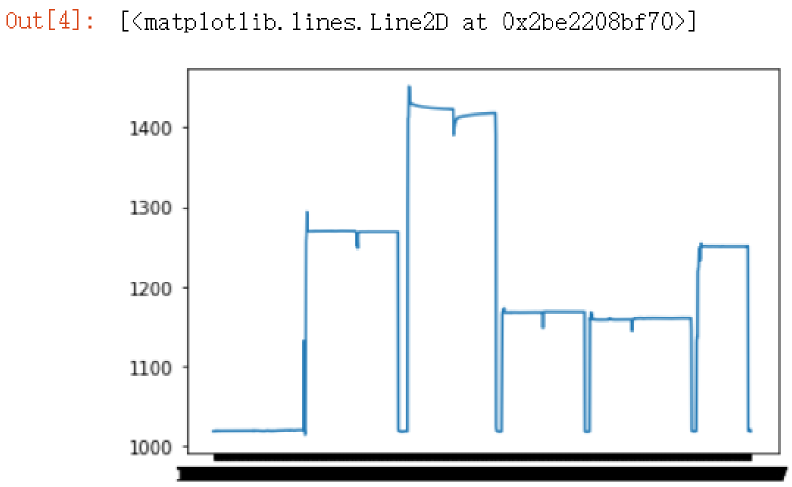


Out[3]:

	date	time	head
0	2021/8/13	14:57:37	1018.567
1	2021/8/13	14:57:47	1018.683
2	2021/8/13	14:57:57	1018.683
3	2021/8/13	14:58:07	1018.567
4	2021/8/13	14:58:17	1018.975
...	...	...	...
1337	2021/8/13	18:40:27	1019.033
1338	2021/8/13	18:40:37	1020.725
1339	2021/8/13	18:40:47	1020.842
1340	2021/8/13	18:40:57	1018.508
1341	2021/8/13	18:41:07	1018.450

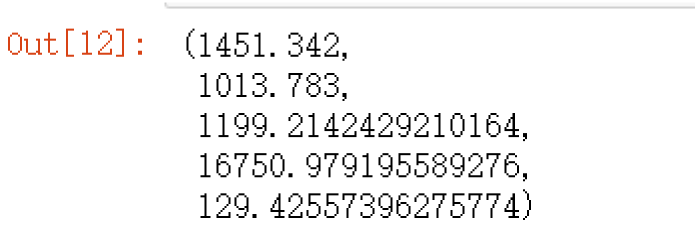
1342 rows × 3 columns

```
#3.2
plt.plot(head['time'],head['head'])
```



```
Out[4]:  [<matplotlib.lines.Line2D at 0x2be2208bf70>]
```

```
#3.3
Max=head['head'].max()
Min=head['head'].min()
Mean=head['head'].mean()
variance=(((head['head']-Mean)**2).sum())/1342
std_deviation=variance**(1/2)
Max,Min,Mean,variance,std_deviation
```



```
Out[12]:  (1451.342,
           1013.783,
           1199.2142429210164,
           16750.979195589276,
           129.42557396275774)
```