PS2_1.1

Compute the total number of deaths caused by earthquakes since 2150 B.C. in each country, and then print the top ten countries along with the total number of deaths. 每个国家的总死亡数，输出前十个国家和死亡数

	Country	Deaths
28	CHINA	2074900.0
140	TURKEY	1074569.0
65	IRAN	1011437.0
131	SYRIA	439224.0
69	ITALY	434863.0
58	HAITI	323472.0
10	AZERBAIJAN	317219.0
71	JAPAN	278138.0
6	ARMENIA	191890.0
102	PAKISTAN	148783.0

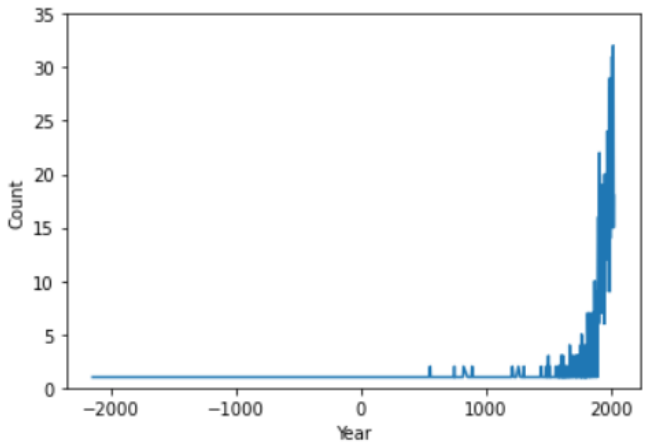
PS2_1.2

Compute the total number of earthquakes with magnitude larger than 6.0 (use column Mag as the magnitude) worldwide each year, and then plot the time series. Do you observe any trend? Explain why or why not? 每年世界>6 级的数量

	Year	Count
298	-2150	1
461	-2000	1
383	-1250	1
382	-1050	1
381	-479	1
...	...	...
2	2017	32
8	2018	28
12	2019	27
72	2020	15
49	2021	18

530 rows × 2 columns

```
Text(0, 0.5, 'Count')
```



Trend: Nearly one thousand year, Earthquakes are incresingly frequent.

There were fewer earthquakes in the earlier period because of, among other things, missing records

PS2_1.3

Write a function CountEq_LargestEq that returns both (1) the total number of earthquakes since 2150 B.C. in a given country AND (2) the date of the largest earthquake ever happened in this country. Apply CountEq_LargestEq to every country in the file, report your results in a descending order

每个国家的[地震总数] 每个国家最大地震的 [日期] 降序

```
###整合到def
Sig_Eqs=pd.read_csv('earthquakes-2021-10-27_15-16-30_+0800.tsv',sep='\t')

def CountEq_LargestEq():
    #(1)the total number of earthquakes since 2150 B.C. in a given country
    Sig_Eqs_CountryCount=Sig_Eqs.value_counts(['Country']).reset_index(name='Count')
    #(2) the date of the largest earthquake ever happened in this country
    Sig_Eqs_CountryMagmax=Sig_Eqs.groupby(['Country'],as_index=False)['Mag'].max()
    Sig_Eqs_CountryCount_CountryMagmax=Sig_Eqs_CountryCount.merge(Sig_Eqs_CountryMagmax,on='Country')
    #Year-->Mo->Dy->Hr->Mn->Sec
    Sig_Eqs['cYear']= Sig_Eqs['Year'].astype(str)
    Sig_Eqs['cMonth']= Sig_Eqs['Mo'].astype(str)
    Sig_Eqs['cDay']= Sig_Eqs['Dy'].astype(str)
    Sig_Eqs['cDate'] = Sig_Eqs['cYear'] + '-' + Sig_Eqs['cMonth'] + '-' + Sig_Eqs['cDay']
    Sig_Eqs_cDate=Sig_Eqs[['cDate','Country','Mag']]
    Sig_Eqs_CountryCount_CountryMagmax_cDate=Sig_Eqs_CountryCount_CountryMagmax.merge(Sig_Eqs_cDate,
                                                                                     on=['Country','Mag'])
    print(Sig_Eqs_CountryCount_CountryMagmax_cDate)

#     return None

CountEq_LargestEq()
```

	Country	Count	Mag	cDate
0	CHINA	610	8.5	1668-7.0-25.0
1	JAPAN	409	9.1	2011-3.0-11.0
2	INDONESIA	401	9.1	2004-12.0-26.0
3	IRAN	380	7.9	856-12.0-22.0
4	TURKEY	330	7.8	1916-1.0-24.0
..	...	...	...	...
173	NORWAY	1	5.8	1819-8.0-31.0
174	SUDAN	1	5.5	1993-8.0-1.0
175	SRI LANKA	1	NaN	1882-1.0-nan
176	PALAU	1	7.6	1914-10.0-23.0
177	ZAMBIA	1	5.9	2017-2.0-24.0

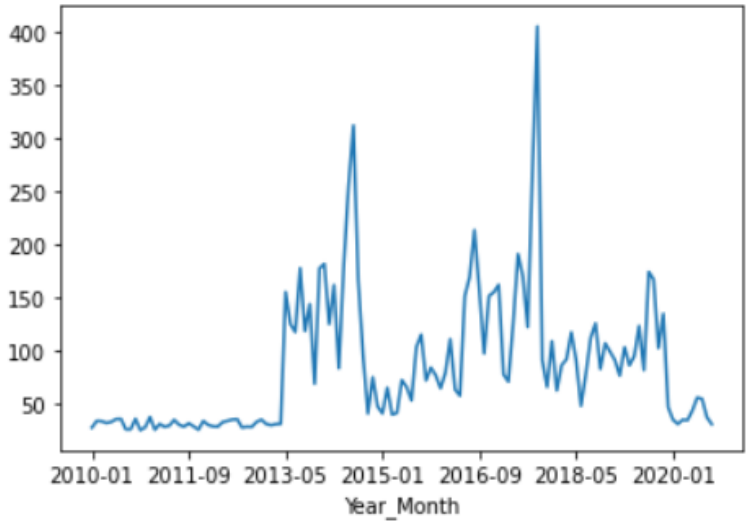
[178 rows x 4 columns]

PS2_2

Explain how you filter the data in your report. #过滤数据
Plot monthly averaged wind speed as a function of the observation time. Is there a trend in monthly averaged wind speed within the past 10 years? #风速月平均

```
#select year and month to string type
data['Year_Month']=data['DATE'].astype('str').str[0:7]
# data['Year_Month']
#select wind speed  to float type
data['WindSpeed']=data['WND'].str[8:12].astype('float')
# data['WindSpeed']
data.groupby(['Year_Month'])['WindSpeed'].mean().plot()
```

```
<AxesSubplot:xlabel='Year_Month'>
```



in monthly averaged wind speed, within the past 10 years, it is very fluctuate

PS2_3.1

clean possible data points with missing values or bad quality.

This data is about Greenland from 1949 to 2020 monthly.

U.S. Department of Commerce
National Oceanic & Atmospheric Administration
National Environmental Satellite, Data, and Information Service
Current Location: Elev: Unknown ft. Lat: Unknown° N Lon: Unknown° W
Station: **Unknown USW00094724**

Global Summary of the Month for 2009
Generated on 03/12/2020

National Centers for Environmental Informati
151 Patton Aven
Asheville, North Carolina 288

Date	Temperature (F)									Number of Days				Precipitation (Inches)						Number of Days			Observed Weather	
Elem ->	TAVG	TMAX	TMIN	HTDD	CLDD	EMXT		EMNT		DX90	DX32	DT32	DT00	PRCP	EMXP		SNOW	EMSD		DP01	DP10	DP1X	DYHF	DYTS
Month	Mean	Mean Max.	Mean Min	Heating Degree Days	Cooling Degree Days	Highest	High Date	Lowest	Low Date	Max >= 90	Max <= 32	Min <= 32	Min <= 0	Total	Greatest Observed Amount	Greatest Observed Date	Snow, Sleet Total Fall	Snow, Sleet Max Depth	Snow, Sleet Max Date	>=.01	>=.10	>=1.0	FG+	TS
Jan	27.1	35.6	18.6	1174	0	53	28	-4	17	0	11	31	1	2.22	0.88	07				9	5	0	1	
Feb	32.4	42.0	22.9	913	0	56	11	4	06	0	4	23	0	0.73	0.18	28				7	5	0	5	
Mar	36.7	45.4	28.0	877	0	63	07	16	04	0	2	21	0	2.58	1.10	29				9	6	1	5	
Apr	48.0	57.6	38.5	513	4	88	26	24	16	0	0	8	0	3.50	1.34	21				9	7	2	7	
May	56.5	64.1	48.9	265	1	79	25	32	13	0	0	1	0	3.01	0.85	06				16	9	0	14	
Jun	61.5	68.9	54.1	117	13	78	30	35	17	0	0	0	0	2.86	0.44	04				20	12	0	13	
Jul	67.7	75.8	59.6	27	109	80	30	42	11	0	0	0	0	6.39	1.18	18				14	10	2	19	
Aug	70.8	78.4	63.1	16	196	88	11	47	09	0	0	0	0	5.35	3.44	29				21	8	1	11	
Sep	60.0	70.4	49.5	164	13	81	05	35	20	0	0	0	0	2.63	1.62	27				6	3	1	7	
Oct	49.9	60.1	39.6	469	0	71	04	25	15	0	0	7	0	6.31	1.63	03				11	8	3	2	
Nov	48.9	56.0	41.9	482	0	66	15	23	07	0	0	5	0	1.97	0.38	23				13	8	0	8	
Dec	34.5	42.3	26.7	946	0	64	03	11	18	0	5	21	0	4.89	1.44	03				9	6	3	3	

Notes

(Blank) Data element not reported or missing.
X Monthly means or totals based on incomplete time series.
\+ Occurred on one or more previous dates during the month. The date in the Date field is the last day of occurrence.

A Accumulated amount.
T Trace Amount.

FG+ Heavy Fog
TS Thunderstorms

```
#delete the missing row if TAVG is missed
dataGL['Temperature']=dataGL['TAVG'].dropna()## https://www.runoob.com/pandas/pandas-cleaning.html
dataGL.shape
dataGL
```

PS2_3.2

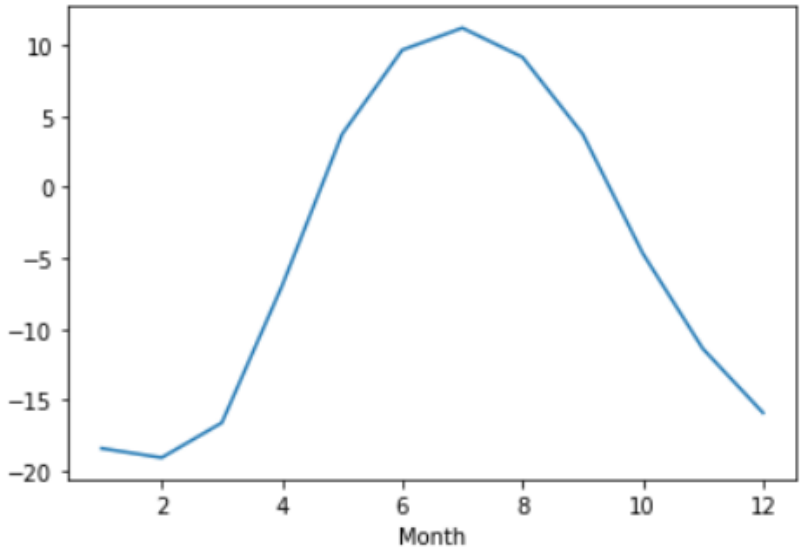
Plot the time series of a certain variable.

```
#create new columns: Year and Month
dataGL['Year'] =pd.to_datetime(dataGL['DATE']).dt.year
dataGL['Month'] = pd.to_datetime(dataGL['DATE']).dt.month
dataGL
```

DATE	...	TMAX_ATTRIBUTES	TMIN	TMIN_ATTRIBUTES	WDFG	WDFG_ATTRIBUTES	WSFG	WSFG_ATTRIBUTES	Temperature	Year	Month
1949/1/1	...	4,,,X	-29.38	4,,,X	NaN	NaN	NaN	NaN	-23.74	1949	1
1949/2/1	...	,,,X	-33.22	,,,X	NaN	NaN	NaN	NaN	-29.59	1949	2
1949/3/1	...	2,,,X	-23.26	2,,,X	NaN	NaN	NaN	NaN	-17.62	1949	3
1949/4/1	...	,,,X	-19.24	,,,X	NaN	NaN	NaN	NaN	-14.75	1949	4
1949/5/1	...	,,,X	-4.96	,,,X	NaN	NaN	NaN	NaN	0.41	1949	5

```
dataGL.groupby(['Month'])['Temperature'].mean().plot()#历年月平均
```

```
<AxesSubplot:xlabel='Month'>
```



PS2_3.3

Conduct at least 5 simple statistical checks with the variable, and report your findings

```
print(dataGL['Temperature'].describe())
print('')
print('mean:',dataGL['Temperature'].mean())
print('max:',dataGL['Temperature'].max())
print('min:',dataGL['Temperature'].min())
print('med:',dataGL['Temperature'].median())
print('std:',dataGL['Temperature'].std())
```

```
count    775.000000
mean      -4.533703
std       11.710298
min      -36.660000
25%      -14.525000
50%       -5.280000
75%        7.140000
max       12.890000
Name: Temperature, dtype: float64

mean: -4.533703225806448
max: 12.89
min: -36.66
med: -5.28
std: 11.710298492795323
```