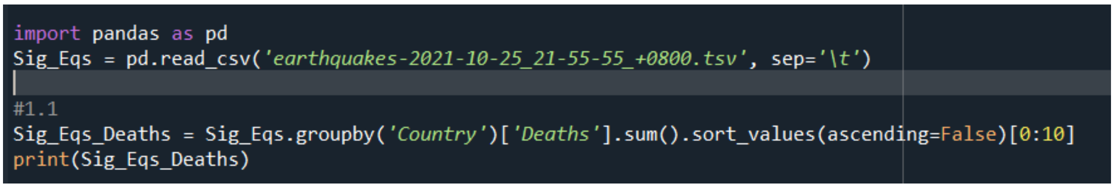
PS2_1

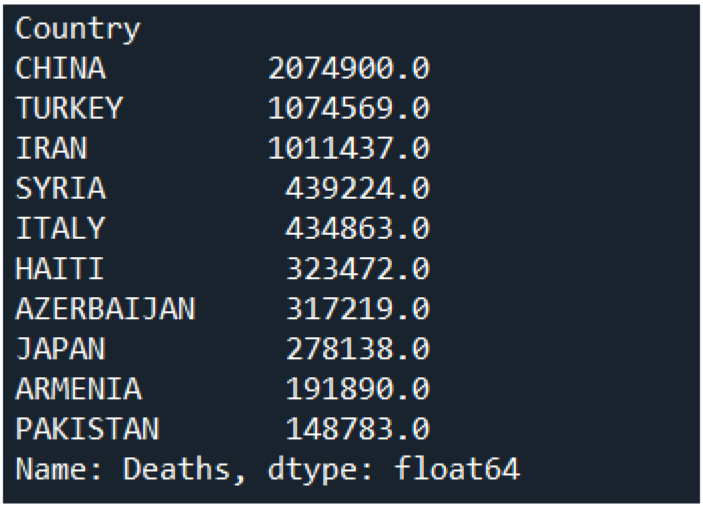
1.1

CODE:



```
import pandas as pd
Sig_Eqs = pd.read_csv('earthquakes-2021-10-25_21-55-55_+0800.tsv', sep='\t')

#1.1
Sig_Eqs_Deaths = Sig_Eqs.groupby('Country')['Deaths'].sum().sort_values(ascending=False)[0:10]
print(Sig_Eqs_Deaths)
```

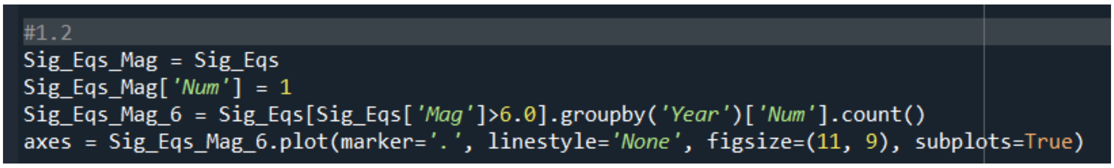
RESULT:



```
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
```

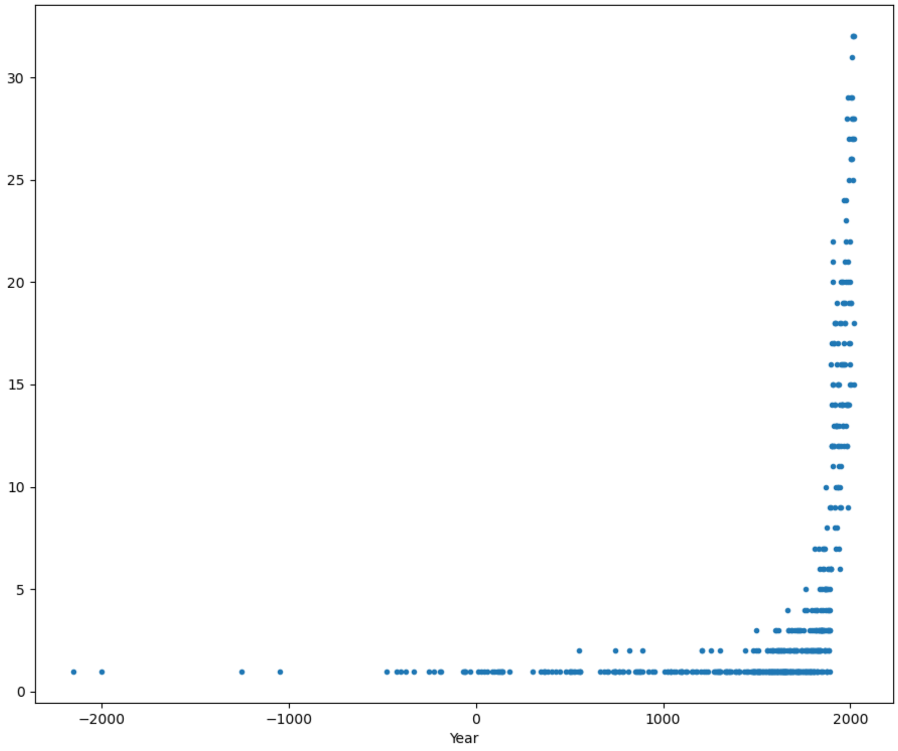
1.2

CODE:



```
#1.2
Sig_Eqs_Mag = Sig_Eqs
Sig_Eqs_Mag['Num'] = 1
Sig_Eqs_Mag_6 = Sig_Eqs[Sig_Eqs['Mag']>6.0].groupby('Year')['Num'].count()
axes = Sig_Eqs_Mag_6.plot(marker='.', linestyle='None', figsize=(11, 9), subplots=True)
```

RESULT:

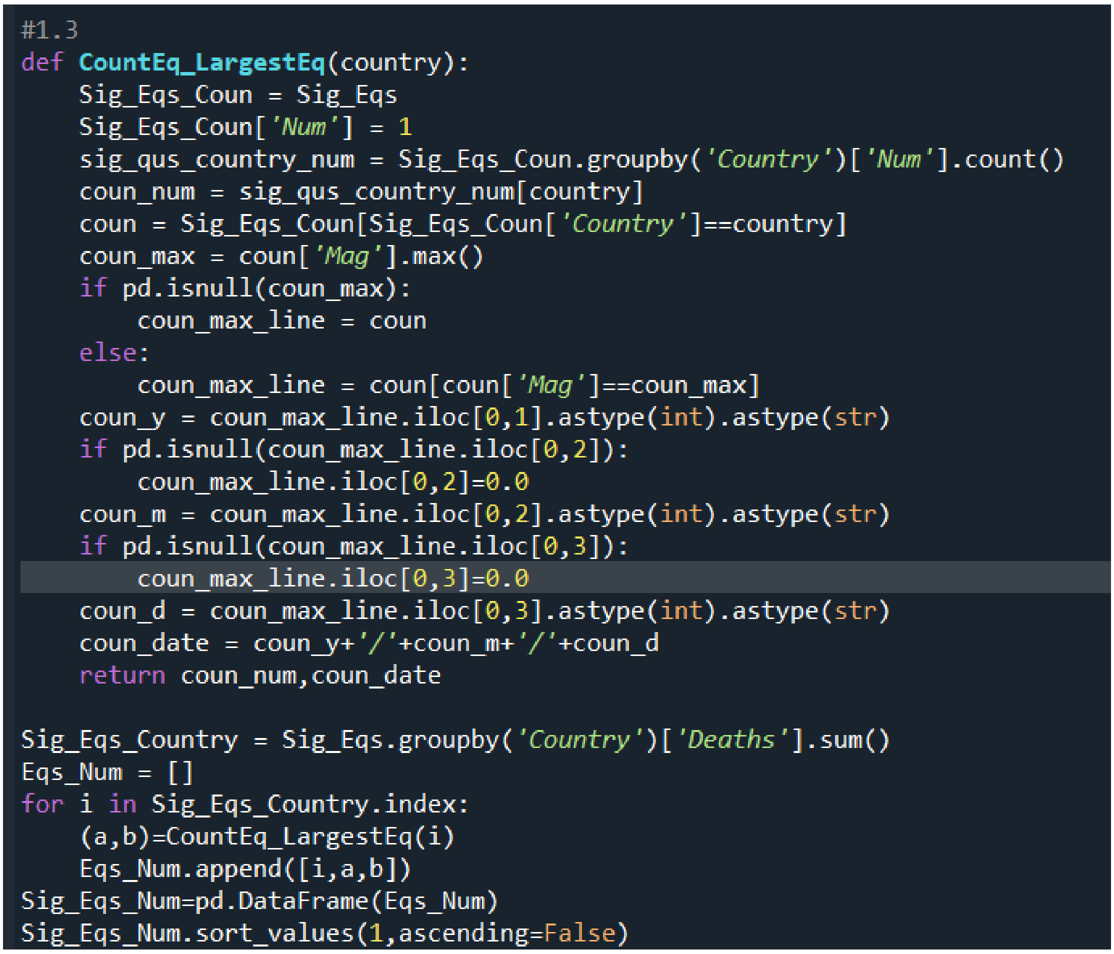


ANSWER:

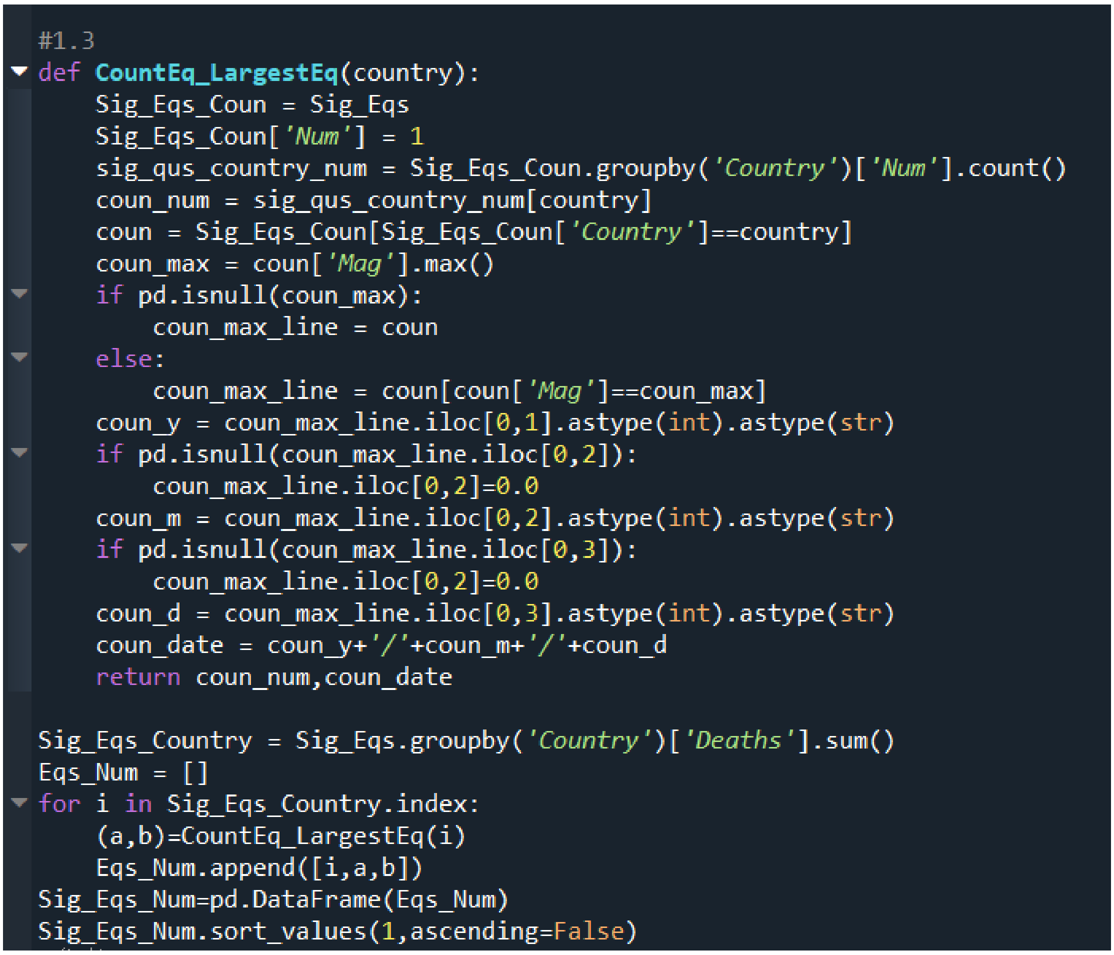
As the methods of observing earthquakes become more sophisticated and the networks become more developed, more earthquakes will be recorded more comprehensively than ever before. It's also possible that the underground plates in inland areas are becoming more active than they used to be.

1.3

CODE:

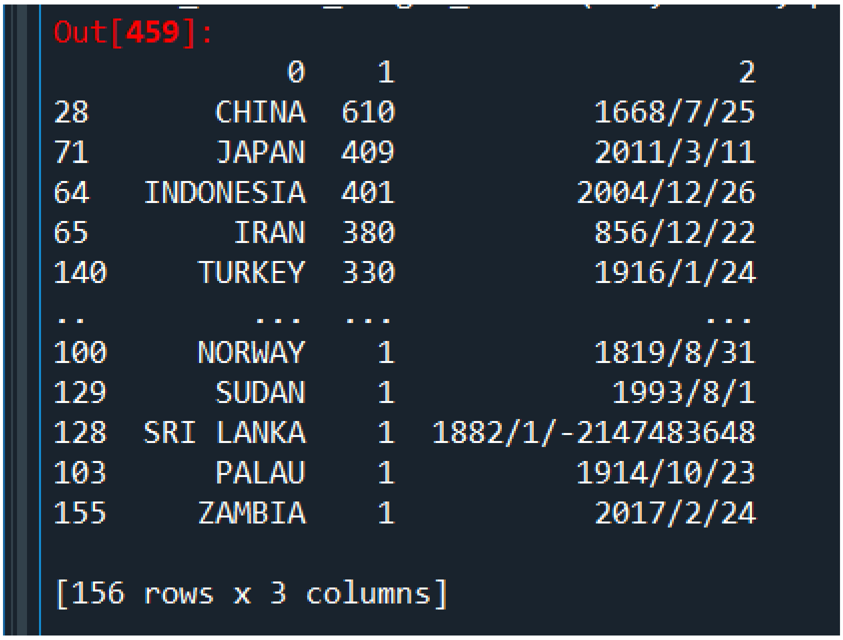


```
#1.3
def CountEq_LargestEq(country):
    Sig_Eqs_Coun = Sig_Eqs
    Sig_Eqs_Coun['Num'] = 1
    sig_qus_country_num = Sig_Eqs_Coun.groupby('Country')['Num'].count()
    coun_num = sig_qus_country_num[country]
    coun = Sig_Eqs_Coun[Sig_Eqs_Coun['Country']==country]
    coun_max = coun['Mag'].max()
    if pd.isnull(coun_max):
        coun_max_line = coun
    else:
        coun_max_line = coun[coun['Mag']==coun_max]
    coun_y = coun_max_line.iloc[0,1].astype(int).astype(str)
    if pd.isnull(coun_max_line.iloc[0,2]):
        coun_max_line.iloc[0,2]=0.0
    coun_m = coun_max_line.iloc[0,2].astype(int).astype(str)
    if pd.isnull(coun_max_line.iloc[0,3]):
        coun_max_line.iloc[0,3]=0.0
    coun_d = coun_max_line.iloc[0,3].astype(int).astype(str)
    coun_date = coun_y+'/'+coun_m+'/'+coun_d
    return coun_num,coun_date

Sig_Eqs_Country = Sig_Eqs.groupby('Country')['Deaths'].sum()
Eqs_Num = []
for i in Sig_Eqs_Country.index:
    (a,b)=CountEq_LargestEq(i)
    Eqs_Num.append([i,a,b])
Sig_Eqs_Num=pd.DataFrame(Eqs_Num)
Sig_Eqs_Num.sort_values(1,ascending=False)
```



```
#1.3
def CountEq_LargestEq(country):
    Sig_Eqs_Coun = Sig_Eqs
    Sig_Eqs_Coun['Num'] = 1
    sig_qus_country_num = Sig_Eqs_Coun.groupby('Country')['Num'].count()
    coun_num = sig_qus_country_num[country]
    coun = Sig_Eqs_Coun[Sig_Eqs_Coun['Country']==country]
    coun_max = coun['Mag'].max()
    if pd.isnull(coun_max):
        coun_max_line = coun
    else:
        coun_max_line = coun[coun['Mag']==coun_max]
    coun_y = coun_max_line.iloc[0,1].astype(int).astype(str)
    if pd.isnull(coun_max_line.iloc[0,2]):
        coun_max_line.iloc[0,2]=0.0
    coun_m = coun_max_line.iloc[0,2].astype(int).astype(str)
    if pd.isnull(coun_max_line.iloc[0,3]):
        coun_max_line.iloc[0,2]=0.0
    coun_d = coun_max_line.iloc[0,3].astype(int).astype(str)
    coun_date = coun_y+'/'+coun_m+'/'+coun_d
    return coun_num,coun_date

Sig_Eqs_Country = Sig_Eqs.groupby('Country')['Deaths'].sum()
Eqs_Num = []
for i in Sig_Eqs_Country.index:
    (a,b)=CountEq_LargestEq(i)
    Eqs_Num.append([i,a,b])
Sig_Eqs_Num=pd.DataFrame(Eqs_Num)
Sig_Eqs_Num.sort_values(1,ascending=False)
```

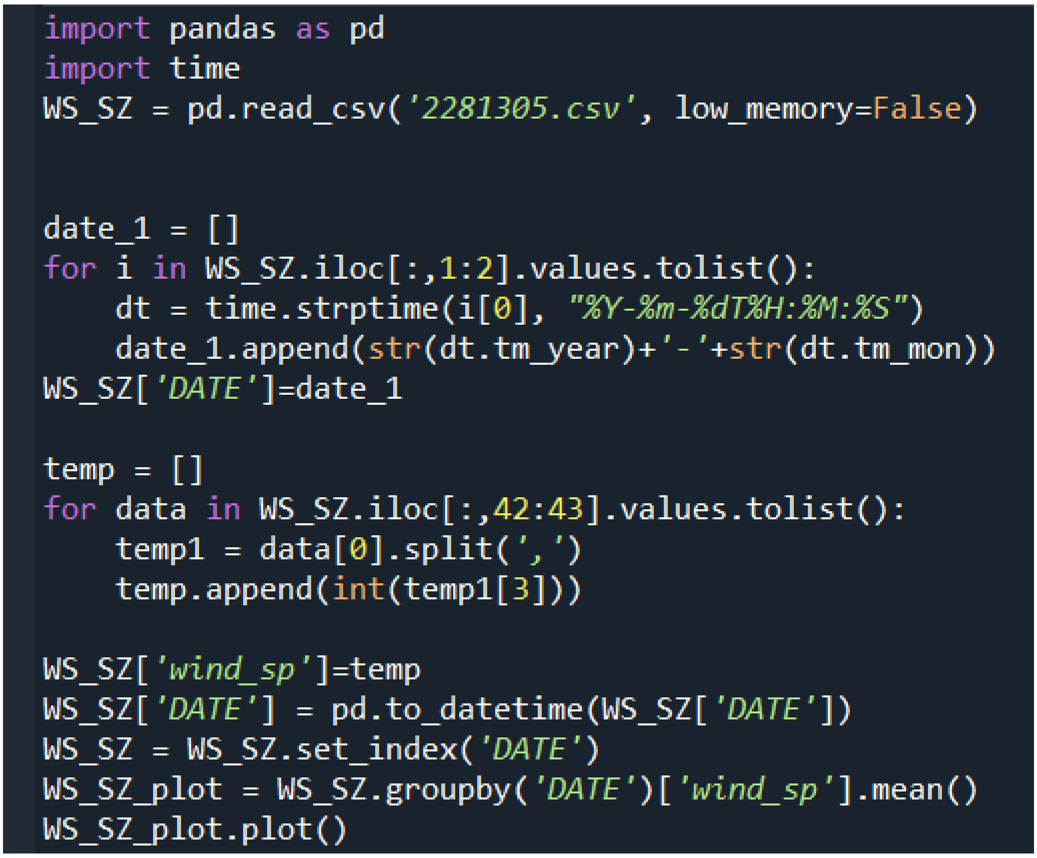
RESULT:



```
Out[459]:
             0    1                 2
28       CHINA  610         1668/7/25
71       JAPAN  409         2011/3/11
64   INDONESIA  401        2004/12/26
65        IRAN  380         856/12/22
140     TURKEY  330         1916/1/24
..         ...  ...               ...
100     NORWAY    1         1819/8/31
129      SUDAN    1          1993/8/1
128  SRI LANKA    1  1882/1/-2147483648
103      PALAU    1        1914/10/23
155     ZAMBIA    1         2017/2/24

[156 rows x 3 columns]
```

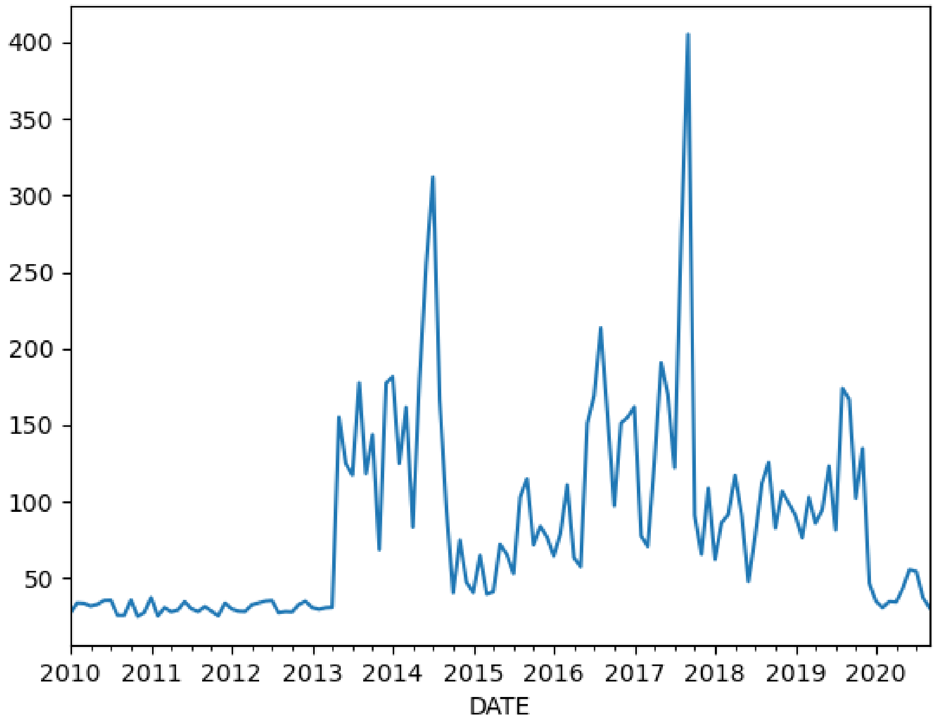
PS2_2

CODE:



```python
import pandas as pd
import time
WS_SZ = pd.read_csv('2281305.csv', low_memory=False)


date_1 = []
for i in WS_SZ.iloc[:,1:2].values.tolist():
    dt = time.strptime(i[0], "%Y-%m-%dT%H:%M:%S")
    date_1.append(str(dt.tm_year)+'-'+str(dt.tm_mon))
WS_SZ['DATE']=date_1

temp = []
for data in WS_SZ.iloc[:,42:43].values.tolist():
    temp1 = data[0].split(',')
    temp.append(int(temp1[3]))

WS_SZ['wind_sp']=temp
WS_SZ['DATE'] = pd.to_datetime(WS_SZ['DATE'])
WS_SZ = WS_SZ.set_index('DATE')
WS_SZ_plot = WS_SZ.groupby('DATE')['wind_sp'].mean()
WS_SZ_plot.plot()
```

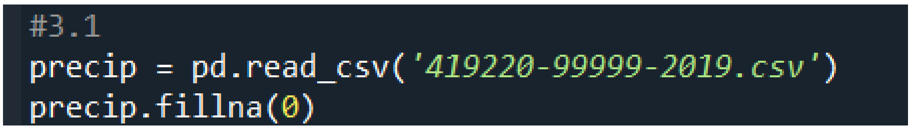
RESULT:



PS2_3

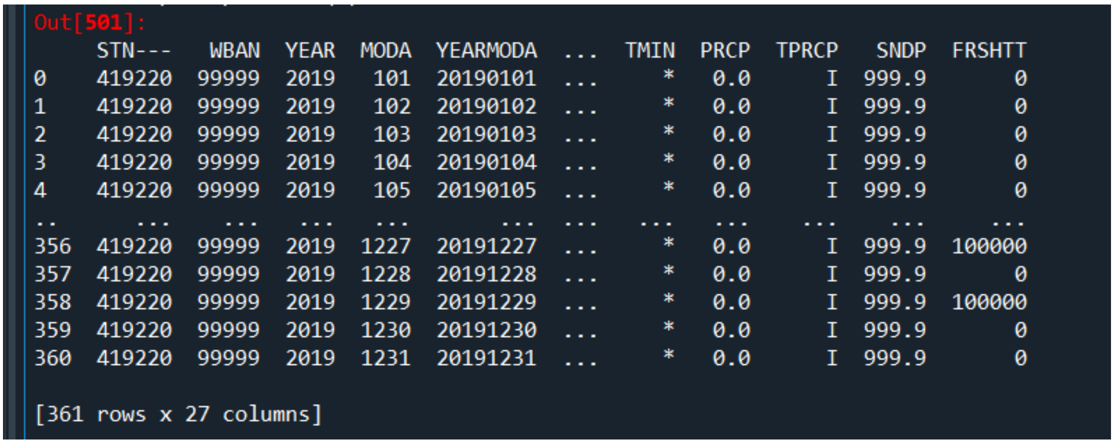
3.1

CODE:



```
#3.1
precip = pd.read_csv('419220-99999-2019.csv')
precip.fillna(0)
```

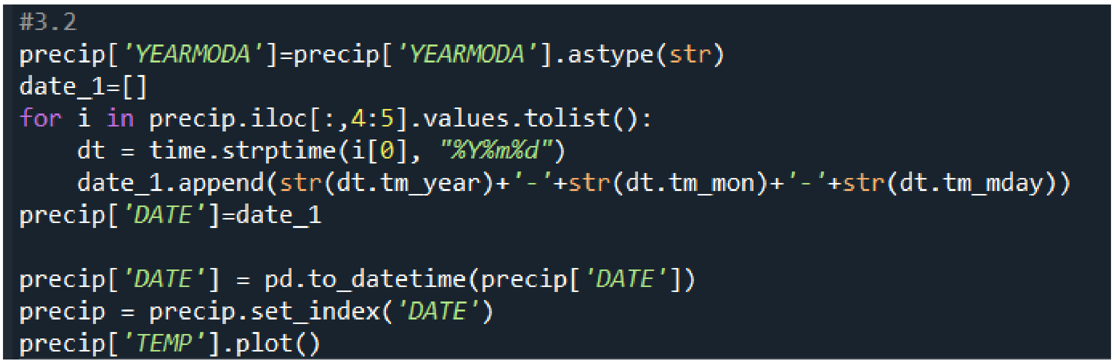
RESULT:



```
Out[501]:
     STN---   WBAN  YEAR  MODA  YEARMODA  ...  TMIN  PRCP  TPRCP   SNDP  FRSHTT
0    419220  99999  2019   101  20190101  ...     *   0.0      I  999.9       0
1    419220  99999  2019   102  20190102  ...     *   0.0      I  999.9       0
2    419220  99999  2019   103  20190103  ...     *   0.0      I  999.9       0
3    419220  99999  2019   104  20190104  ...     *   0.0      I  999.9       0
4    419220  99999  2019   105  20190105  ...     *   0.0      I  999.9       0
..      ...    ...   ...   ...       ...  ...   ...   ...    ...    ...     ...
356  419220  99999  2019  1227  20191227  ...     *   0.0      I  999.9  100000
357  419220  99999  2019  1228  20191228  ...     *   0.0      I  999.9       0
358  419220  99999  2019  1229  20191229  ...     *   0.0      I  999.9  100000
359  419220  99999  2019  1230  20191230  ...     *   0.0      I  999.9       0
360  419220  99999  2019  1231  20191231  ...     *   0.0      I  999.9       0

[361 rows x 27 columns]
```

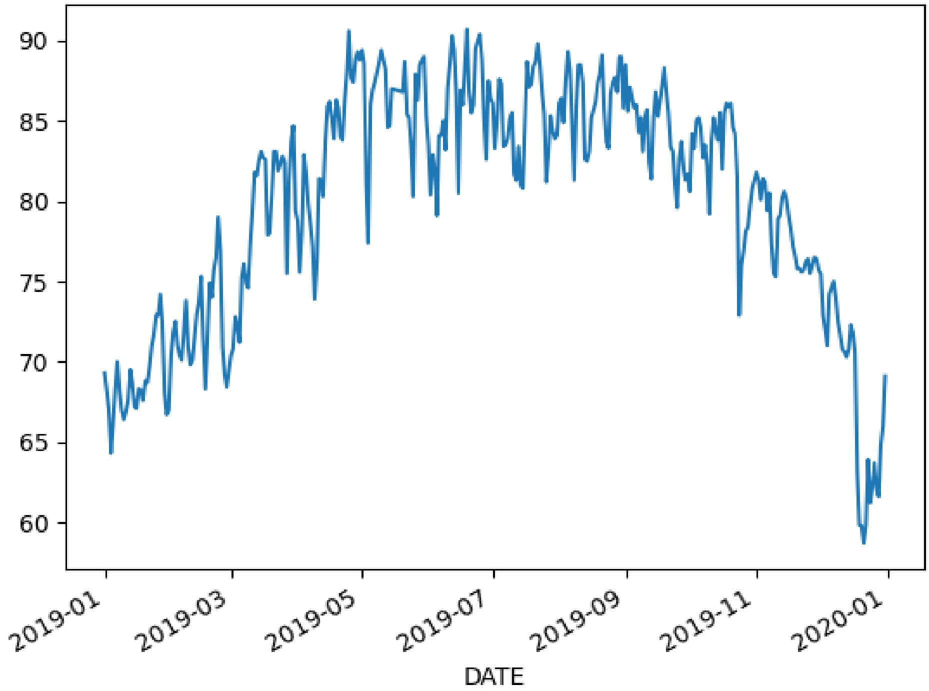
3.2

CODE:



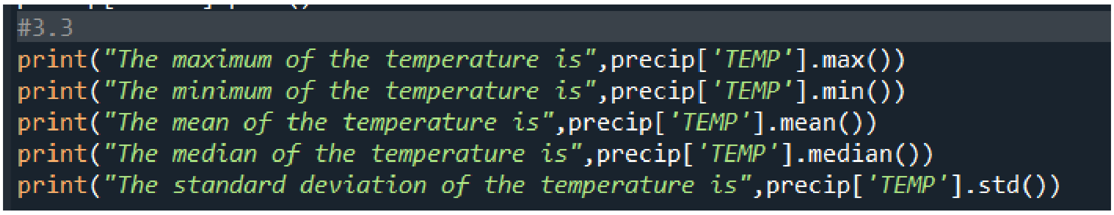
```
#3.2
precip['YEARMODA']=precip['YEARMODA'].astype(str)
date_1=[]
for i in precip.iloc[:,4:5].values.tolist():
    dt = time.strptime(i[0], "%Y%m%d")
    date_1.append(str(dt.tm_year)+'-'+str(dt.tm_mon)+'-'+str(dt.tm_mday))
precip['DATE']=date_1

precip['DATE'] = pd.to_datetime(precip['DATE'])
precip = precip.set_index('DATE')
precip['TEMP'].plot()
```

RESULT:



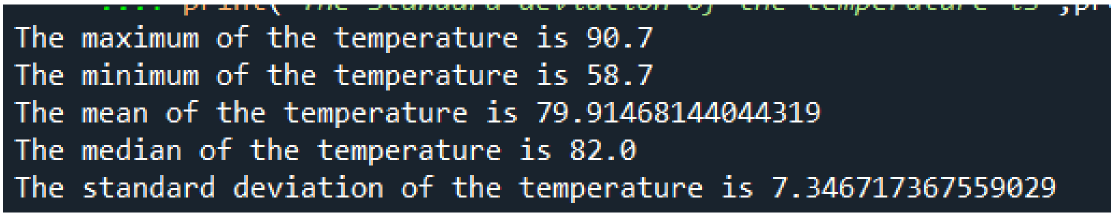
3.3

CODE:



```
#3.3
print("The maximum of the temperature is",precip['TEMP'].max())
print("The minimum of the temperature is",precip['TEMP'].min())
print("The mean of the temperature is",precip['TEMP'].mean())
print("The median of the temperature is",precip['TEMP'].median())
print("The standard deviation of the temperature is",precip['TEMP'].std())
```

RESULT:



```
   ...: print("The standard deviation of the temperature is",pr
The maximum of the temperature is 90.7
The minimum of the temperature is 58.7
The mean of the temperature is 79.91468144044319
The median of the temperature is 82.0
The standard deviation of the temperature is 7.346717367559029
```

REFERENCE：http://www.cainiaoxueyuan.com/bc/6033.html