南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Assignment 2

ESE 5023

张子严

2021年10月21日

1. Significant earthquakes since 2150 B.C.

```
(base) Ziyans-MacBook-Pro:untitled1 ziyanzhang$ cat PS2.py
import pandas as pd
import matplotlib.pyplot as plt


def CountEq_LargestEq(country):
    country = country.upper()
    frame = pd.read_table('earthquakes-2021-10-13_19-14-37_+0800.tsv').groupby('Country')
    grouped_frame = frame.get_group(country).sort_values("Mag", ascending=False)
    print('The total number of earthquakes since 2150 B.C.: %d\n' % grouped_frame.shape[0])
    grouped_frame.rename(columns={'Mo': 'Month', 'Dy': 'Day'}, inplace=True)
    print(grouped_frame[['Country', 'Year', 'Month', 'Day', 'Mag']].head(1))


if __name__ == '__main__':
    Sig_Eqs = pd.read_table('earthquakes-2021-10-13_19-14-37_+0800.tsv')
    grouped_sig = Sig_Eqs.groupby("Country").sum().sort_values("Deaths", ascending=False)
    print(grouped_sig["Deaths"].head(10))
    mag = Sig_Eqs[Sig_Eqs["Mag"] > 6.0]
    grouped_mag = mag[["Year", "Mag"]].groupby("Year").count().plot()
    plt.show()
    CountEq_LargestEq(input('Country:'))
(base) Ziyans-MacBook-Pro:untitled1 ziyanzhang$
```

1.1

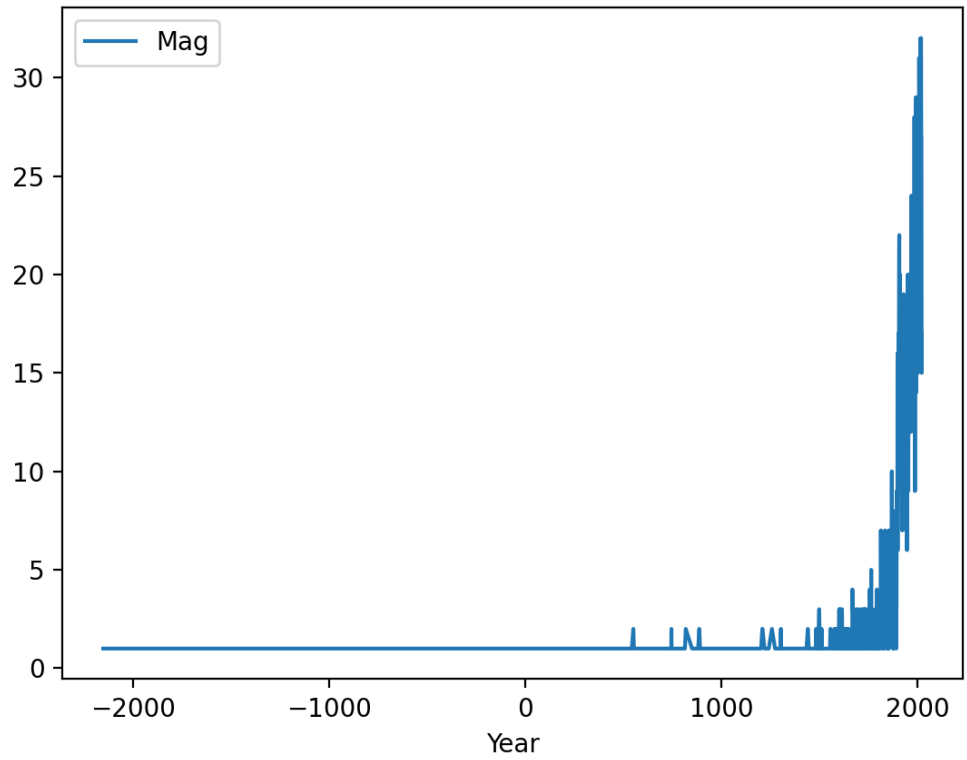
Outputs:

```
/Library/anaconda3/bin/python.app /Users/ziyanzhang/Documents/untitled1/PS2.py
Country
CHINA         2074900.0
TURKEY        1074769.0
IRAN          1011437.0
SYRIA          439224.0
ITALY          434863.0
HAITI          323472.0
AZERBAIJAN     317219.0
JAPAN          278138.0
ARMENIA        191890.0
PAKISTAN       148764.0
```

1.2

Observation: Through our graph, it is easy to find out that the magnitude of earthquake is in recent 1000 years is much higher than the magnitude from 2150 B.C. to 1000 A.C. Hence maybe we can make a prediction that the crustal movement is getting more and more intense.

Graph:



1.3 Output

For example, we set 'China' as the input, the output is showed below:

```
/Library/anaconda3/bin/python.app /Users/ziyanzhang/Documents/untitled1/PS2.py
Country:china
The total number of earthquakes since 2150 B.C.: 610

    Country    Year  Month   Day  Mag
973   CHINA  1668.0    7.0  25.0  8.5

Process finished with exit code 0
```

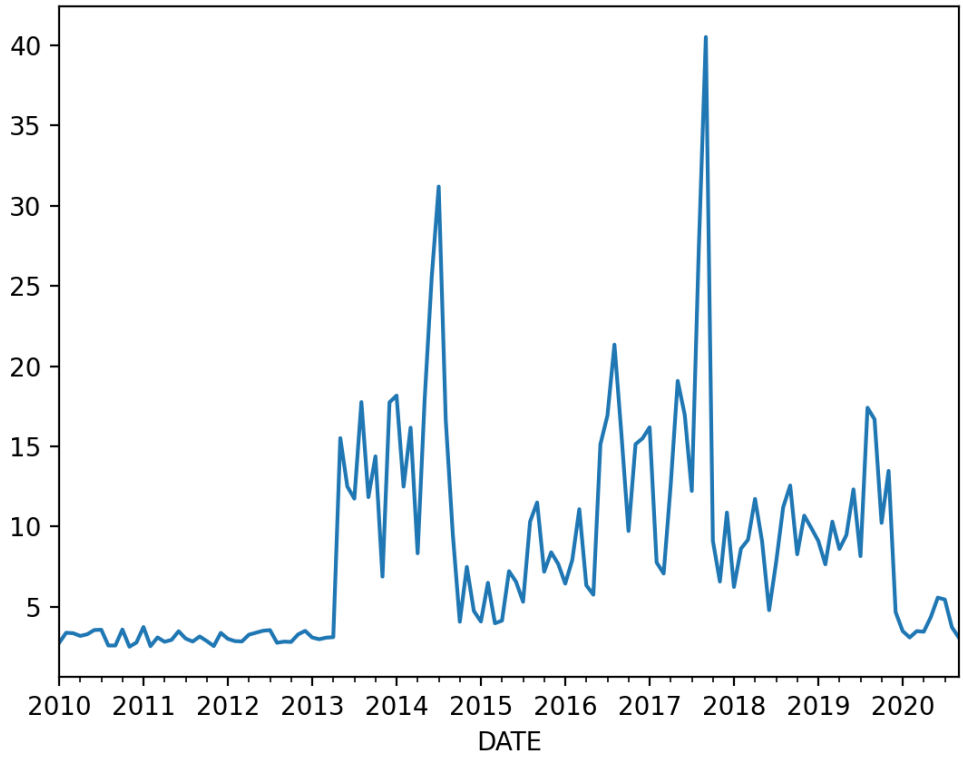
2. Wind speed in Shenzhen during the past 10 years

2.1 Split Method

According to the given guidance, the 'WIND' column consists five parts: direction angle, qualify code of direction angle, wind type code, speed rate and qualify code of wind speed, listed respectively. Simultaneously, there exists a scaling factor as 10 for the wind speed rate.

Hence, we only need the DATE column and the WIND column to calculate monthly averaged wind speed.

2.2 Visualization



The graph of date and monthly average wind speed is demonstrated up there. From our graph, we can indicate that the wind speed reaches a plateau from July to September. Also, we can find that in the past decade, the wind speed is incredibly high in 2017, but suddenly drop to a lower level since 2018.

2.3 Source

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

df = pd.read_csv('2281305.csv', low_memory=False)[['DATE', "WND"]]
date = pd.to_datetime(df['DATE'])
month = date.dt.to_period('M')
df['DATE'] = month
wind = df['WND'].str.split(",")
data = []
for item in wind.values:
    data.append((float(item[3])/10))
speed = pd.Series(data, copy=False)
df['WND'] = speed
df_subset = df[['DATE', 'WND']].groupby('DATE').mean()['WND']
df_subset.plot(kind='bar')
plt.show()
```

3. Explore a data set

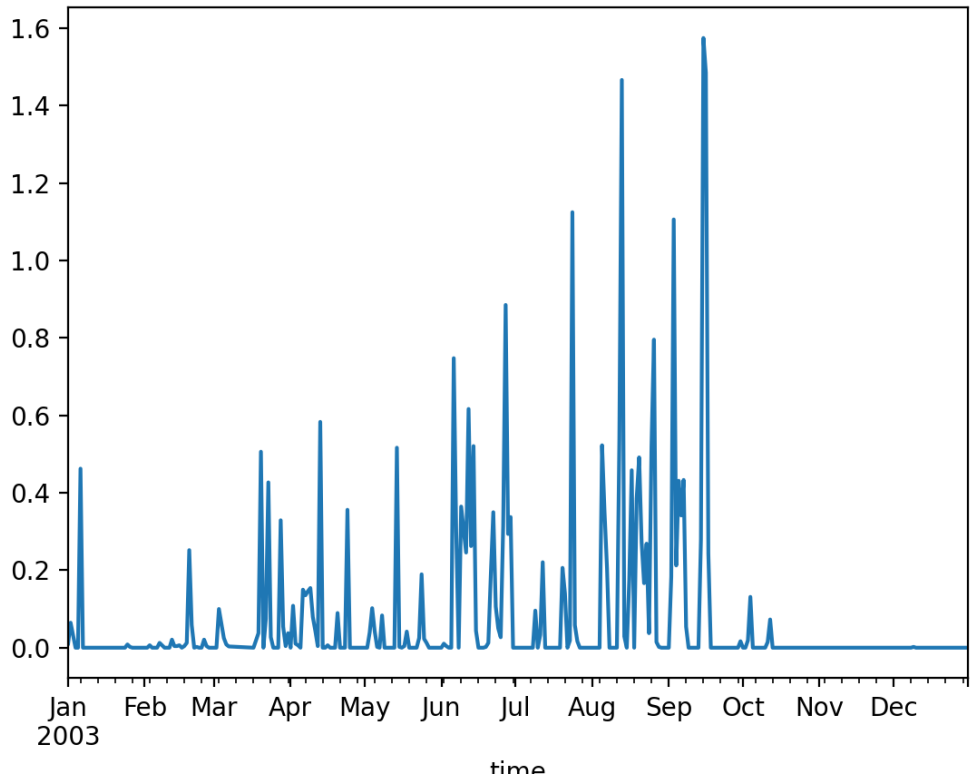
3.1

First, we select a data set that contains the daily rainfall of the DingHu Mountain area. Then, we read this data set through pandas and filter out the missing and uncertain data. We know that the daily rainfall should at least be 0, therefore we drop the rows that contain negative value. Also we regulate the columns name to meet the function input requirement.

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


df = pd.read_excel('总降水量.xlsx', sheet_name='鼎湖山')
df['year'] = df['年']
df['month'] = df['月']
df['day'] = df['日']
df['hour'] = df['时']
df['minute'] = df['分']
df = df[['year', 'month', 'day', 'hour', 'minute', '降水量']]
df = df[df['降水量'] > -1]
time = pd.to_datetime(df[['year', 'month', 'day', 'hour', 'minute']])
df['time'] = time
df = df[['time', '降水量']]
print(df.head())
```

3.2 Plot the data



3.3

(1) Get max rainfall per minute in the whole data set

```
max rainfall per minute:
                     time   降水量
10797 2003-08-13 22:00:00  30.2
```

(2) Get monthly average rainfall in every year

```
time
2003-08    321.8
2003-09    305.4
2003-06    287.6
2003-07     91.9
2003-04     89.9
Freq: M, Name: 降水量, dtype: float64
```

(3) Get the month that has the most plentiful rainfall

```
time
2003-08    321.8
Freq: M, Name: 降水量, dtype: float64
```

(4) Get the average daily rainfall and the day that has the most plentiful rainfall.

```
time
2003-01-01    0.010638
2003-01-02    0.064583
2003-01-03    0.031915
2003-01-04    0.000000
2003-01-05    0.000000
Freq: D, Name: 降水量, dtype: float64
time
2003-01-01    0.010638
Freq: D, Name: 降水量, dtype: float64
```

(5) Plot average monthly rainfall