1. Significant earthquakes since 2150 B.C.

The Significant Earthquake Database contains information on destructive earthquakes from 2150 B.C. to the present. On the top left corner, select all columns and download the entire significant earthquake data file in `.tsv` format by clicking the `Download TSV File` button. Click the variable name for more information. Read the file (e.g., `earthquakes-2023-1024_16-20-01_+0800.tsv`) as an object and name it `Sig_Eqs`.

1.1 [5 points] 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.

1.2 [10 points] 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?

1.3 [10 points] 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.

Program:

```
#File Reading and Instruction Calling
import pandas as pd import numpy as
np from matplotlib import pyplot as
plt from pandas import Series,
DataFrame
Sig_Eqs=pd.read_csv('earthquakes-2023-10-24_16-20-01_+0800.tsv', delimiter='\t')
#Q1.1
top_deaths=Sig_Eqs.groupby(['Country'])['Deaths'].sum()
top_10_countries=top_deaths.nlargest(10) print(top_10_countries)
#Q1.2
Mig6=Sig_Eqs[Sig_Eqs['Mag']>6.0] earthquake_counts =
Mig6.groupby('Year').size() plt.bar(earthquake_counts.index,
earthquake_counts.values,10) plt.xlabel('Year')
plt.ylabel('Number of Earthquakes')
```

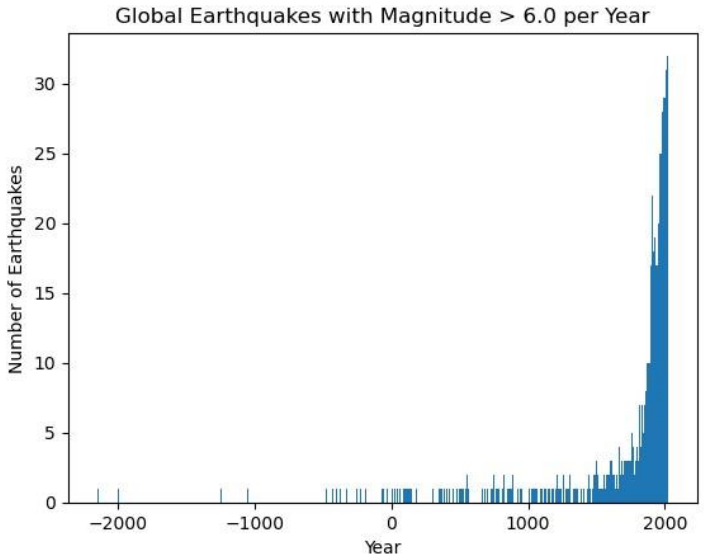
```
plt.title('Global Earthquakes with Magnitude > 6.0 per Year')
plt.show() #Q1.3 data = Sig_Eqs
Sig_Eqs['Year']=Sig_Eqs['Year'].fillna(0).astype(int)
Sig_Eqs['Mo']=Sig_Eqs['Mo'].fillna(0).astype(int)
Sig_Eqs['Dy']=Sig_Eqs['Dy'].fillna(0).astype(int)
def CountEq_LargestEq(data):     results = []

# Group the data by country     grouped_data
= data.groupby('Country')

   for country, group in grouped_data:
      total_earthquakes = group[(group['Year'] >= -2150)]['Year'].count()

      largest_earthquake = group[group['Mag'] == group['Mag'].max()]
largest_earthquake_date = None

      if not largest_earthquake.empty:
         year = largest_earthquake['Year'].values[0]
month = largest_earthquake['Mo'].values[0]            day =
largest_earthquake['Dy'].values[0]
         largest_earthquake_date = f"{year}-{month}-{day}"

      results.append((country, total_earthquakes, largest_earthquake_date))

   return results

# Apply CountEq_LargestEq to the earthquake data results
= CountEq_LargestEq(data)

# Create a DataFrame from the results results_df = pd.DataFrame(results,
columns=['Country', 'Total Earthquakes', 'Date of Largest Earthquake'])

# Sort the results DataFrame in descending order by the date of the largest earthquake
results_df = results_df.sort_values('Date of Largest Earthquake', ascending=False)

# Print the results for index, row in
results_df.iterrows():
      print(f"Country: {row['Country']}, Total Earthquakes since 2150 B.C.: {row['Total
Earthquakes']}, Date of Largest Earthquake: {row['Date of Largest Earthquake']}")
```

The Result:

Q1-1

```
Country
CHINA          2075045.0
TURKEY         1188881.0
IRAN           1011449.0
ITALY           498478.0
SYRIA           439224.0
HAITI           323478.0
AZERBAIJAN      317219.0
JAPAN           279085.0
ARMENIA         191890.0
PAKISTAN        145083.0
Name: Deaths, dtype: float64
```

Q1-2



Q1-3

```
Country: SPAIN, Total Earthquakes since 2150 B.C.: 34, Date of Largest Earthquake: 881-5-26
Country: IRAN, Total Earthquakes since 2150 B.C.: 384, Date of Largest Earthquake: 856-12-22
Country: LEBANON, Total Earthquakes since 2150 B.C.: 14, Date of Largest Earthquake: 551-7-9
Country: GREECE, Total Earthquakes since 2150 B.C.: 270, Date of Largest Earthquake: 365-7-21
Country: MOROCCO, Total Earthquakes since 2150 B.C.: 21, Date of Largest Earthquake: 2023-9-8
Country: CUBA, Total Earthquakes since 2150 B.C.: 14, Date of Largest Earthquake: 2020-1-28
Country: COMOROS, Total Earthquakes since 2150 B.C.: 1, Date of Largest Earthquake: 2018-5-15
Country: ZAMBIA, Total Earthquakes since 2150 B.C.: 1, Date of Largest Earthquake: 2017-2-24
Country: MADAGASCAR, Total Earthquakes since 2150 B.C.: 1, Date of Largest Earthquake: 2017-1-11
Country: RWANDA, Total Earthquakes since 2150 B.C.: 5, Date of Largest Earthquake: 2015-8-7
Country: THAILAND, Total Earthquakes since 2150 B.C.: 4, Date of Largest Earthquake: 2014-5-5
```

Problem-solving ideas：

In question 1.1, group by country and calculate the total number to obtain a new column of data. Then, extract the top ten values and corresponding countries using the nlargest function. In question 1.2, first extract rows with Mag greater than 6, then group them by year, calculate the number of data in each group, and plot. In question 1.3, convert the non empty data of the month month day column to an integer, create a function, and finally return an array. The function's function includes counting, finding the maximum value of the earthquake, and returning the time.

2. Wind speed in Shenzhen during the past 10 years

In this problem set, we will examine how wind speed changes in Shenzhen during the past `10` years, we will take a look at the hourly weather data measured at the BaoAn International Airport. The data set is from NOAA Integrated Surface Dataset. Download the file 2281305.zip, where the number `2281305` is the site ID. Extract the zip file, you should see a file named `2281305.csv`. Save the `.csv` file to your `working directory`.

Read page `8-9` (`POS 65-69` and `POS 70-70`) of the comprehensive user guide for the detailed format of the wind data. Explain how you filter the data in your report.

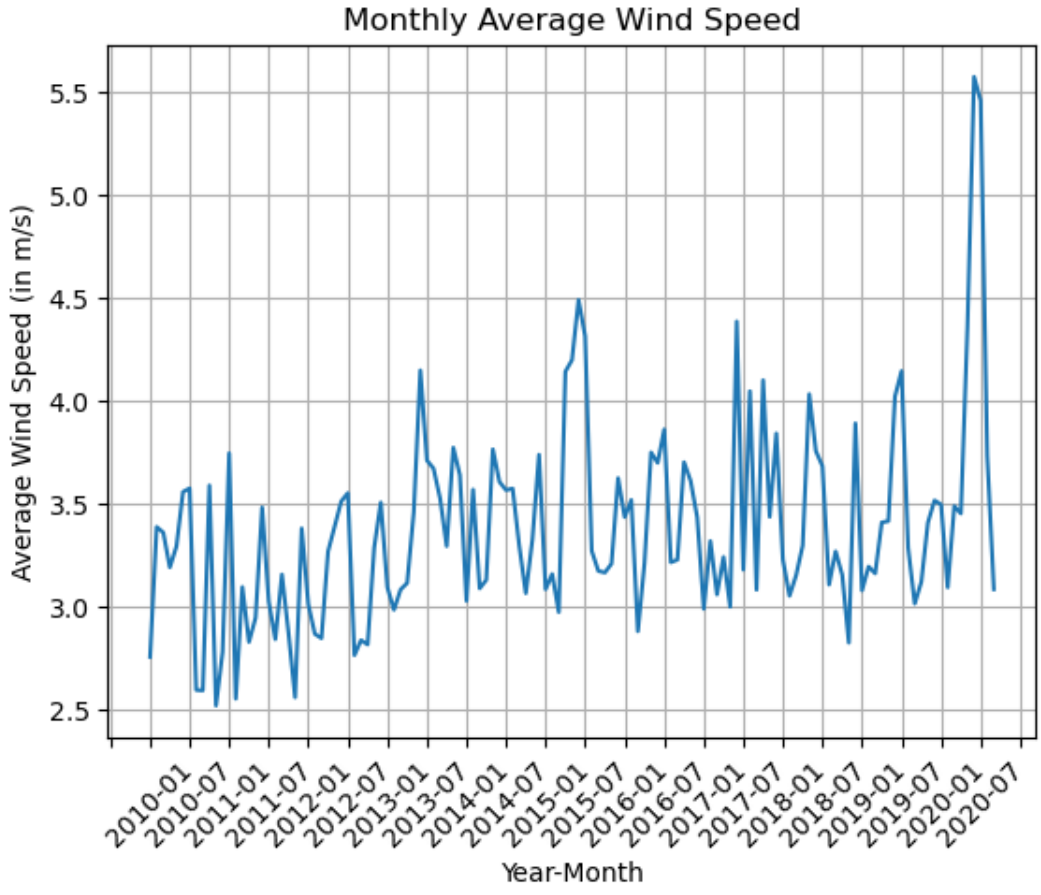
[10 points] Plot monthly averaged wind speed as a function of the observation time. Is there a trend in monthly averaged wind speed within the past `10` years?

Program:

```
#File Reading and Instruction Calling
import pandas as pd import numpy as
np from matplotlib import pyplot as
plt import matplotlib.ticker as ticker
from pandas import Series, DataFrame

#Extract data from tables
SZ_wind=pd.read_csv('2281305.csv')
SZ_wind['year'] = SZ_wind['DATE'].str.split('-').str[0] SZ_wind['month']
= SZ_wind['DATE'].str.split('-').str[1]
SZ_wind['type_code'] = SZ_wind['WND'].str.split(',').str[2]
SZ_wind['speed_rate'] = SZ_wind['WND'].str.split(',').str[3]
SZ_wind = SZ_wind[SZ_wind['speed_rate'] != 9999]
SZ_wind['speed_rate'] = SZ_wind['speed_rate'].astype(float) / 100
SZ_wind['speed_quality_code'] = SZ_wind['WND'].str.split(',').str[4]

#Calculate monthly average wind speed
monthly_avg_speed=SZ_wind.groupby(['year','month'])['speed_rate'].mean().reset_in
dex()

x = [f'{year}-{month}' for year, month in
zip(monthly_avg_speed['year'],monthly_avg_speed['month'])] y
= monthly_avg_speed.speed_rate
```

```
ticker_spacing = 6 #Plot
the diagram
fig, ax = plt.subplots()

plt.xlabel('Year-Month')
plt.ylabel('Average Wind Speed (in m/s)')
plt.title('Monthly Average Wind Speed')

ax.plot(x, y)
ax.xaxis.set_major_locator(ticker.MultipleLocator(ticker_spacing))
plt.xticks(rotation=45) plt.grid(True)
plt.show()
```

The Result:



Problem-solving ideas：

Due to the presence of multiple data in some cells, the split function is used for segmentation and str is used for extraction, followed by data processing_ The rate parameter 9999 needs to be deleted. After deletion, the value needs to be divided by ten

before use. By grouping the year and month, the average value is calculated, and the plot is made after calculation.

3. Explore a data set

Browse the CASEarth, National Centers for Environmental Information (NCEI), or Advanced Global Atmospheric Gases Experiment (AGAGE) website. Search and download a data set you are interested in. You are also welcome to use data from your group in this problem set. But the data set should be in `csv`, `XLS`, or `XLSX` format, and have temporal information.

3.1 [5 points] Load the `csv`, `XLS`, or `XLSX` file, and clean possible data points with missing values or bad quality.

3.2 [5 points] Plot the time series of a certain variable.

3.3 [5 points] Conduct at least `5` simple statistical checks with the variable, and report your findings.

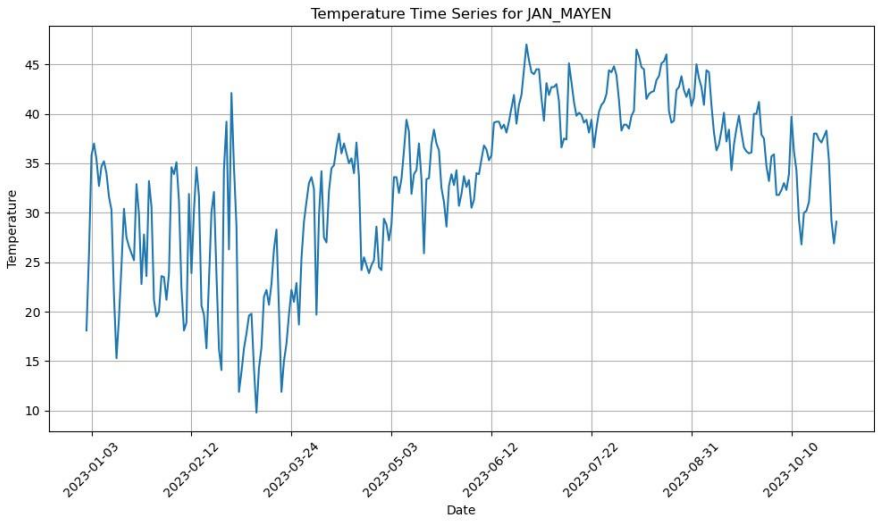
Program:

```
#Q3.1
#File Reading and Instruction Calling
import pandas as pd import numpy as
np
from matplotlib import pyplot as plt

JAN_MAYEN = pd.read_csv('JAN_MAYEN.csv')
JAN_MAYEN['TEMP'] = JAN_MAYEN['TEMP'].replace(999.9, np.nan)
#Q3.2
# Convert the 'DATE' column to datetime format
JAN_MAYEN['DATE'] = pd.to_datetime(JAN_MAYEN['DATE'])

# Plotting the time series
plt.figure(figsize=(10, 6))
plt.plot(JAN_MAYEN['DATE'], JAN_MAYEN['TEMP'])

# Customize the plot
plt.title('Temperature Time Series for JAN_MAYEN')
plt.xlabel('Date') plt.ylabel('Temperature')
plt.grid(True)
plt.gca().xaxis.set_major_locator(ticker.MaxNLocator(nbins=10))
# Set number of ticks on x-axis plt.xticks(rotation=45)
plt.tight_layout()
```

```
# Show the plot
plt.show() #Q3.3
# Compute the statistical measures mean_temp
= JAN_MAYEN['TEMP'].mean() max_temp =
JAN_MAYEN['TEMP'].max() min_temp =
JAN_MAYEN['TEMP'].min() var_temp =
JAN_MAYEN['TEMP'].var()
range_temp = max_temp - min_temp

# Print the results
print("Mean Temperature:", mean_temp)
print("Maximum Temperature:", max_temp)
print("Minimum Temperature:", min_temp)
print("Variance of Temperature:", var_temp) print("Range
of Temperature:", range_temp)
```

The Result:

Q3.2



Q3.3

```
Mean Temperature: 33.342
Maximum Temperature: 47.0
Minimum Temperature: 9.8
Variance of Temperature: 65.63501939799333
Range of Temperature: 37.2
```

Problem-solving ideas：

Find climate information about Jay Mayen region in 2023 on NCEI, and read_ The CSV function extracts the CSV file and replaces the missing values (999.9). Using days as the x-axis and temperature as the y-axis, plot a time function of temperature, and then calculate its maximum, minimum, variance, range, average, and discuss the climate situation in the region.