

```
In [1]: # A2Q1
        # independently by Shao Shi
```

```
In [2]: import pandas as pd
import numpy as np
from matplotlib import pyplot as plt

Sig_Eqs = pd.read_csv('earthquakes-2022-10-23_15-23-40_+0800.tsv', sep='\t', header=0)
Sig_Eqs = Sig_Eqs.drop(Sig_Eqs.index[[0]], axis=0) # delete an empty row

# A2Q1_1
countryName = list()
countryDeath = list()
for name, data in Sig_Eqs.groupby(['Country']):
    countryName.append(name)
    countryDeath.append(np.nansum(data['Deaths']))
country_deaths = pd.Series(countryDeath, index=countryName).sort_values(ascending=False)
for i in range(0, 20):
    print(country_deaths.index[i], end = ': ')
    print(country_deaths[i])
```

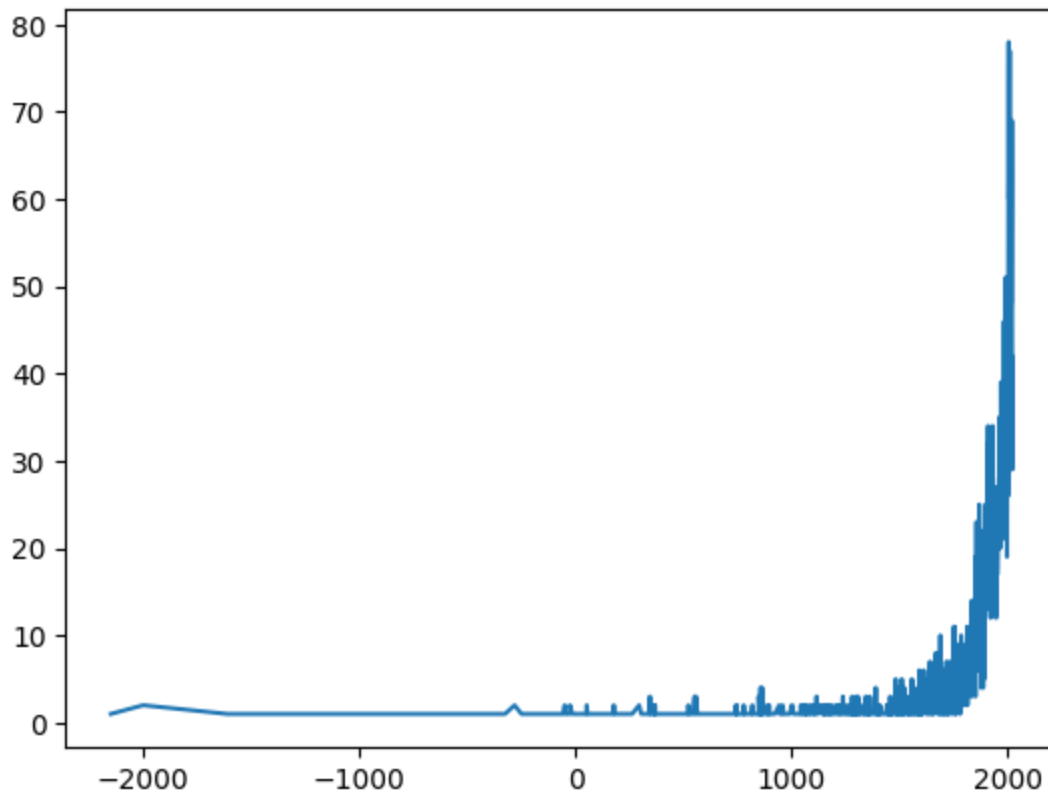
```
CHINA: 2075019.0
TURKEY: 1134569.0
IRAN: 1011446.0
ITALY: 498477.0
SYRIA: 439224.0
HAITI: 323474.0
AZERBAIJAN: 317219.0
JAPAN: 278142.0
ARMENIA: 191890.0
PAKISTAN: 145083.0
IRAQ: 136200.0
ECUADOR: 135479.0
TURKMENISTAN: 117412.0
PERU: 102219.0
ISRAEL: 90388.0
PORTUGAL: 83531.0
GREECE: 79174.0
CHILE: 64276.0
INDIA: 63491.0
TAIWAN: 57135.0
```

C:\Users\Almustafa\AppData\Local\Temp\ipykernel\_3936\2146756341.py:11: FutureWarning: In a future version of pandas, a length 1 tuple will be returned when iterating over a groupby with a grouper equal to a list of length 1. Don't supply a list with a single grouper to avoid this warning.

```
for name, data in Sig_Eqs.groupby(['Country']):
```

```
In [3]: # A2Q1_2
yearName = list()
yearEarthquakeCount = list()
for name, data in Sig_Eqs.groupby(['Year']):
    yearName.append(name)
    yearEarthquakeCount.append(len(data))
# print(yearName)
# print(yearEarthquakeCount)
plt.plot(yearName, yearEarthquakeCount)
plt.show()
```

C:\Users\Almustafa\AppData\Local\Temp\ipykernel\_3936\314223592.py:4: FutureWarning: In a future version of pandas, a length 1 tuple will be returned when iterating over a groupby with a grouper equal to a list of length 1. Don't supply a list with a single grouper to avoid this warning.  
for name, data in Sig\_Eqs.groupby(['Year']):



```
In [4]: # 3 the function defined
def CountEq_LargestEq(Sig_Eqs, countryName):
    countryData = Sig_Eqs[Sig_Eqs["Country"] == countryName]
    eqkNumber = len(countryData)
    largestEqkLoc = ''.join(countryData["Location Name"][countryData["Ms"] == countryData["Ms"].i
    return [eqkNumber, largestEqkLoc]
```

```
In [5]: # 3 run
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from A2Q1_3 import CountEq_LargestEq

# Load data
Sig_Eqs = pd.read_csv('earthquakes-2022-10-23_15-23-40_+0800.tsv', sep='\t', header=0)
Sig_Eqs = Sig_Eqs.drop(Sig_Eqs.index[[0]], axis=0) # delete an empty row
# run A2Q1_3
countryName = Sig_Eqs["Country"].unique()
countryEqkNumber = list(range(len(countryName)))
countryLargestEqkLoc = list(range(len(countryName)))
for i in range(len(countryName)):
    countryEqkNumber[i], countryLargestEqkLoc[i] = CountEq_LargestEq(Sig_Eqs, countryName[i])

country_deaths = pd.Series(countryEqkNumber, index=countryLargestEqkLoc).sort_values(ascending=False)
print(country_deaths.to_string())
```

CHINA: GANSU PROVINCE, SHANXI PROVINCE	616
JAPAN: SANRIKUJAPAN: NANKAIDO	411
INDONESIA: SUMATRA: ACEH: OFF WEST COAST	405
IRAN: DAMGHAN, QUMIS	384
TURKEYTURKEY: ERZINCANTURKEY: ISTANBUL, KOCAELI, SAKARYA	332
ITALY: MARSICA, AVEZZANO, ABRUZZI	331
ALASKA	271
GREECE: CRETE: KNOSSOSGREECE: CRETE	270
PHILIPPINES: MINDANAO, ZAMBOANGA, SULU, ISABELA	222
MEXICO: SAN MARCOS, OAXACAMEXICO: GUERRERO	208
CHILE: VALPARAISO	198
PERU: TRUJILLO, PIURA, SANTAPERU: ICA, NAZCA	190
RUSSIA: KAMCHATKA PENINSULA	152
TAIWANTAIWAN	100
PAPUA NEW GUINEA: NEW IRELAND, DUKE OF YORK	99
INDIA: ASSAM; BANGLADESH	99
PANAMA-COLOMBIA	79
NEW ZEALAND: FJORDLANDNEW ZEALAND: WELLINGTON,WAIOURU,WANGANUI,OTAKI	70
ECUADOR: OFF COAST	67
VENEZUELA: MACUTO	66
AFGHANISTAN: HINDU-KUSH	62
SOLOMON ISLANDS: SANTA CRUZ ISLANDSSOLOMON ISLANDS	61
ALGERIA: NORTHERN	57
ALBANIA: HIMARA, DHERMI, KUC, KUDHESI, VLORE, KANINA	56
VANUATU ISLANDSVANUATU ISLANDS	54
PAKISTAN: MAKRAN COAST	53
BALKANS NW: CROATIA: DUBROVNIK: RAGUSA	53
FRANCE: VERNEGUES,CHARLEVAL,LA ROQUE D'ANTHERON	43
GUAM: AGANA	40
NICARAGUA: LEON, CHINANDEGA, MANAGUA	39
GUATEMALA: NEAR S COAST	39
EL SALVADOR; GUATEMALA	38
COSTA RICA-NICARAGUACOSTA RICA: NICOYA	35
SYRIA: SOUTHWESTERN	33
MYANMAR (BURMA): MANDALAY, MOGOK, MAYMYO	33
	31
SPAIN: ARENAS DEL REY, VEGA, ALHAMA, MALAGA	27
AZORES: TERCEIRA, ANGRA DO HEROISMO	27
TAJIKISTAN: KARATAGTAJIKISTAN: USOY, BARCHIDIV, POSOR, NISURTAJIKISTAN	26
PORTUGAL: LISBON	26
IRAQ: ZURBATIYAH, BADRAH, TURSAQ, BAGHDAD	24
MACQUARIE ISLAND: MACQUARIE STATION; NEW ZEALAND	24
PANAMA: SAN BLAS ARCHIPELAGO	23
NEW CALEDONIA: LOYALTY ISLANDS: LIFOU ISLAND	23
ISRAEL: ZEFAT (SAFED)ISRAEL: ZEFAT (SAFED); LEBANON: BEIRUT (BEYROUT)	23
TONGA ISLANDS	22
BALKANS NW: SLOVENIA: IDRIJA,SKOFJA LOKA	22
SEA OF JAPANYELLOW SEASOUTH KOREA: CHUNGCHONG	21
ARGENTINA: LA RIOJA, SAN JUAN, MENDOZAARGENTINA: SANTIAGO DEL ESTERO PROVINCE	21
MOROCCO: AL HOCEIMA, IMZOURENE, BENI ABDALLAH	20
CANADA: QUEEN CHARLOTTE ISLANDS	20
JAMAICA	19
FIJI ISLANDS	19
KERMADEC ISLANDS: RAOUL	19
HAITI: CAP-HAITIEN	19
DOMINICAN REPUBLIC: NORTHEASTERN COAST	18
BULGARIA: STRUMA	18
ICELAND: VESTMANNAEYJAR, HELLAICELAND: GRIMSNES, SELFOS, EYRARBAKKI, STOKKSEYRI	17
BANGLADESH: SRIMANGAL	17
NEPAL-INDIA	16
AZERBAIJAN: SHEMAKHA (SAMAXI)AZERBAIJAN: SEMACHA; N IRAN	16
BALKANS NW: SERBIA	15

GEORGIA: CAUCASUS	15
ROMANIA: BUCHAREST, KISHINEV-KAGUL	15
EGYPT: NUWAYBI; SAUDI ARABIA; ISRAEL; JORDAN	15
KYRGYZSTAN: TOLUK	14
	14
SOUTH AFRICA: PRINCE EDWARD ISLAND	14
UZBEKISTAN: GAZLIUZBEKISTAN: GAZLI, BUKHARAUZBEKISTAN: GAZLI	14
CUBA: PILON, MANZANILLO	14
ARMENIA: LENINAKAN, SPITAK, KIROVAKAN	13
UK: SCARBOROUGH, GRIMSBY	13
HONDURAS: COASTHONDURAS	13
BALKANS NW: MACEDONIA: DEBAR	12
UKRAINE: CRIMEA: SEBASTOPOL	12
BALKANS NW: BOSNIA-HERZEGOVINA: TITOGRAĐ	11
TURKMENISTAN: UZUN-ADA	11
YEMEN: DHAMAR	10
MARTINIQUE	10
KAZAKHSTAN: ALMA-ATA, TURKESTAN	10
BALKANS NW: MONTENEGRO	10
GUADELOUPE: GRAND BOURG	9
TUNISIA: SIDI ABID,SIDI TOUIL (LA MEDJA),CAILLOUX	9
GERMANY	9
ETHIOPIA: KARAKORE	9
SAMOA ISLANDS	8
TANZANIA: RUKWA	8
TRINIDAD & ST. CHRISTOPHER	8
SOUTH SANDWICH ISLANDS	7
BRAZIL: JOAO CAMARA, NATAL	7
CYPRUS: PAPHOS	7
CONGO: KALEMIE	7
POLAND: KATOWICE	7
AUSTRIA: ARNOLDSTEIN; SLOVENIA: BOVEC, KOBARID	7
NORTH KOREA: YELLOW SEA	6
ATLANTIC OCEAN: NORTHERN	6
MONGOLIA	6
BOLIVIA-NORTHERN CHILEBOLIVIA-NORTHERN CHILE	6
ERITREA: ASMARA	6
RWANDA: GISENYI	5
VIETNAM	5
BALLENY ISLANDS	5
BHUTAN: TASHIGANG	5
JORDAN: AL-KARAK	5
	5
GHANA: ACCRA	5
MICRONESIA, FED. STATES OF: CAROLINE ISLANDS	4
THAILAND: BANGKOK	4
MALAWI: SALIMA, DEDZA, MOHINJI	4
UGANDA: KICHWAMBA, BONDIBOGYO; TANZANIA; DR CONGO	4
MOZAMBIQUE	3
SAUDI ARABIA: WESTERN	3
	3
THE NETHERLANDS: ROERMOND; GERMANY: BONN, HEINSBERG	3
MALAYSIA: SABAH: LAHAD,DATU,KANAK	3
ANTIGUA; SAINT KITTS AND NEVIS	3
INDIAN OCEAN: S	3
KENYA: SUBUKIA	3
SOUTH SUDAN: JUBA, MAYA; UGANDA: NAKURA	3
	2
	2
LAOS: BOKEO; THAILAND: CHIANG RAI, CHIAN SAEN	2
	2
	2

[illegible][illegible][illegible]
$$\begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{matrix}$$
$$\begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{matrix}$$
$$\begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{matrix}$$
$$\begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{matrix}$$