HOMEWORK 1: CAUTION! CONTENTS ARE HOT 🌋

DUE: *FEBRUARY 20, 2025 @ 11:59 PM*

24-HR LATE DUE DATE WITH A 15% PENALTY: *FEBRUARY 21, 2025 @ 11:59 PM*

The NCEI/WDS Global Significant Volcanic Eruptions Database is a very comprehensive collection of +600 volcanic eruptions dating from 4360 BC to the present. Due to the nature of this assignment, we will be dealing with relatively newer volcanoes (in which some are still obviously still older than anyone on Earth currently). Each eruption in the database is classified as significant if it meets one or more criteria, such as causing fatalities, incurring **damage on property** (**+$1 million**), reaching a **Volcanic Explosivity Index (VEI)** of **6 or higher**, generating a tsunami, or being linked to a significant earthquake. The VEI is a scale that measures the explosiveness of volcanic eruptions, providing insight into the magnitude and potential consequences of the eruptions. The database includes detailed information on the location, type of volcano, last known eruption, VEI, casualties, property damage, and much more.

We are going to dive straight into these volcanoes (well... their dataset), to swim our way into Pandas proficiency!

You will find the Pandas Documentation helpful. There are also some helpful links to guide you along the way! Don't get burned 🔥

DO NOT REMOVE ANY PART OF ANY OF THE QUESTIONS OR YOU LOSE CREDIT

No Hardcoding either 😋❤️‍🔥

REMEMBER TO SHOW ALL CODE OUTPUT (NO CREDIT OTHERWISE)

Part 1: Maintenance 🤩 (25 POINTS TOTAL)

First, we're going to familiarize ourselves with the process. As in most languages, Python looks best when its modules are imported first before any other code is written ✨

In [73]:
```
# Make sure these code blocks run properly and that you have properly instal
import pandas as pd
import requests
# import other libraries here

# Don't remove this
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)
```

As you may have noticed, there's another library aside from Pandas called "requests." **The requests library allows you to send HTTP requests to a server, retrieve the content, and process it at ease.** It's very beginner friendly for those attempting to get into webscraping (super important for collecting and creating datasets). We also recommend looking into BeautifulSoup (yeah, soup LOL), another wonderful library that can be paired with the requests library for webscraping.

As shown below, sometimes specific websites require specific headers in order to process a request to access the data.

To check if a request was processed successfully, use the status_code function to see if the process returned 200.

In [74]:
```
# API URL and headers in case request gets denied.
api_url = "https://www.ngdc.noaa.gov/hazel/hazard-service/api/v1/volcanoes"

headers = {
    'accept': '*/*'
}
```

TASK 1.0: Cute Webscraping (5 points)

- To make our cute webscraper we need to **create a GET request** using the hints above.
- This particular dataset NOAA returns data from the API as ***json*** when a user makes a request.
- The json data has a particular format, so we will **extract our needed information only from the field called items** to make a dataframe (you may need to store this data before turning it into a dataframe).
- After properly scraping the data, **name this dataframe *df***

- Save this dataframe into a **CSV file named 'volcanoes.csv'**

You won't need to run this cell more than once

In [75]:
```
response = requests.get(api_url, headers = headers)
json = response.json()
items = json.get('items', [])
df = pd.DataFrame(items)
df.to_csv('volcanoes.csv', index = False)
```

TASK 1.1: 1 Liner Thingz (3 points)

We need to get an idea of what this dataset is going to look. In order to do that, let's take a look at some of the most basic things our dataframe has.

Read the directions carefully and code your answer with only one line of code.

CAN'T USE LOOPS. DO NOT DISPLAY THE DATAFRAME, JUST YOUR CODE OUTPUT HERE.

1.1.1: In one line of code and **using only one single attribute call**, output ***only the numbers*** of **datapoints and features** in the dataframe.

Hint: The output's going to be a tuple

In [76]:
```
df.shape
```

Out[76]:
```
(200, 43)
```

1.1.2: In one line of code and **using only one single attribute call**, list the **names** of ***all the features*** in the dataframe.

In [77]:
```
df.columns
```

Out[77]:
```
Index(['id', 'year', 'month', 'day', 'tsunamiEventId', 'earthquakeEventId',
       'volcanoLocationId', 'volcanoLocationNewNum', 'volcanoLocationNum',
       'name', 'location', 'country', 'latitude', 'longitude', 'elevation',
       'morphology', 'agent', 'deathsTotal', 'deathsAmountOrderTotal',
       'damageAmountOrderTotal', 'significant', 'publish', 'eruption',
       'status', 'timeErupt', 'vei', 'deathsAmountOrder', 'damageAmountOrde
r',
       'housesDestroyedAmountOrderTotal', 'deaths', 'injuries',
       'injuriesAmountOrder', 'injuriesTotal', 'injuriesAmountOrderTotal',
       'housesDestroyedAmountOrder', 'housesDestroyed', 'housesDestroyedTot
al',
       'missingAmountOrder', 'missingAmountOrderTotal', 'missing',
       'missingTotal', 'damageMillionsDollars', 'damageMillionsDollarsTota
l'],
      dtype='object')
```

We won't be using some of the data because there is a lot of missing data.

1.1.3: *In one line of code, create a **new dataframe** called **new_df** that **discards** all the features of the **old** dataframe **except for the following**:*

`id, year, month, day, tsunamiEventId, earthquakeEventId, volcanoLocationId, volcanoLocationNewNum, name, country, elevation, morphology, deathsTotal, vei, deaths` *Hint: Don't use any drop function here*

In [78]:
```
new_df = df[['id', 'year', 'month', 'day', 'tsunamiEventId', 'earthquakeEver
new_df # KEEP THIS. It will display the whole dataframe.
```

Out[78]:

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
0	1	1169	2.0	4.0	2852.0	421.0	10106
1	2	1329	7.0	NaN	NaN	NaN	10106
2	3	1883	3.0	NaN	NaN	NaN	30301
3	4	1888	3.0	13.0	1175.0	NaN	50107
4	5	1850	NaN	NaN	NaN	NaN	50214
5	6	1832	11.0	1.0	NaN	NaN	10106
6	7	1977	1.0	10.0	NaN	NaN	20303
7	8	787	NaN	NaN	NaN	NaN	10102
8	9	1779	8.0	8.0	NaN	NaN	10102
9	10	1302	NaN	NaN	NaN	NaN	10103
10	11	1907	8.0	4.0	NaN	NaN	201112
11	12	1905	3.0	10.0	NaN	NaN	10102
12	13	1986	7.0	24.0	NaN	NaN	10104
13	14	1536	3.0	23.0	NaN	NaN	10106
14	15	1904	2.0	25.0	NaN	NaN	30301
15	16	1878	1.0	NaN	1095.0	6195.0	50214
16	17	1737	5.0	20.0	NaN	NaN	10102
17	18	1917	4.0	1.0	NaN	NaN	40105
18	19	1899	NaN	NaN	NaN	NaN	50102
19	20	1979	3.0	8.0	NaN	NaN	50103
20	21	1970	3.0	2.0	NaN	NaN	10101

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
21	22	1972	6.0	9.0	NaN	NaN	30302
22	23	1682	8.0	12.0	NaN	NaN	10102
23	24	1805	8.0	11.0	NaN	NaN	10102
24	25	1919	5.0	22.0	1476.0	NaN	10104
25	26	1930	9.0	11.0	4218.0	NaN	10104
26	27	1903	8.0	30.0	NaN	NaN	40105
27	28	1928	11.0	2.0	NaN	NaN	10106
28	29	1872	4.0	24.0	NaN	NaN	10102
29	30	1979	7.0	5.0	NaN	NaN	10106
30	31	1987	4.0	17.0	NaN	NaN	10106
31	32	1914	9.0	10.0	NaN	NaN	40104
32	33	1886	8.0	31.0	NaN	NaN	40311
33	34	1912	12.0	3.0	NaN	NaN	20302
34	35	1895	6.0	17.0	NaN	NaN	50103
35	36	1538	9.0	29.0	NaN	NaN	10101
36	37	1853	6.0	24.0	NaN	NaN	40311
37	38	1954	8.0	3.0	NaN	NaN	50101
38	39	1944	3.0	27.0	NaN	NaN	10102
39	40	1984	8.0	15.0	5807.0	NaN	20403
40	41	-141	NaN	NaN	NaN	NaN	10106
41	42	1928	1.0	23.0	NaN	NaN	10204
42	43	1631	2.0	14.0	NaN	920.0	201141

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
43	44	1843	11.0	17.0	NaN	NaN	10106
44	45	1944	12.0	4.0	NaN	NaN	60320
45	49	1979	2.0	20.0	NaN	NaN	60320
46	61	1587	NaN	NaN	NaN	NaN	60325
47	64	1672	8.0	4.0	NaN	NaN	60325
48	69	1822	12.0	27.0	NaN	NaN	60325
49	71	1832	12.0	25.0	NaN	NaN	60325
50	81	1872	4.0	15.0	NaN	NaN	60325
51	82	1872	11.0	3.0	NaN	NaN	60325
52	87	1902	12.0	NaN	NaN	NaN	60325
53	92	1920	7.0	25.0	NaN	NaN	60325
54	94	1930	12.0	18.0	NaN	NaN	60325
55	100	1953	3.0	23.0	NaN	NaN	60325
56	104	1961	5.0	8.0	NaN	NaN	60325
57	105	1969	1.0	7.0	NaN	NaN	60325
58	111	1986	10.0	15.0	NaN	NaN	60325
59	112	1994	11.0	22.0	NaN	NaN	60325
60	116	1311	NaN	NaN	NaN	NaN	60328
61	117	1334	NaN	NaN	NaN	NaN	60328
62	118	1376	NaN	NaN	NaN	NaN	60328
63	119	1385	NaN	NaN	NaN	NaN	60328
64	126	1586	NaN	NaN	NaN	NaN	60328
65	128	1716	7.0	20.0	NaN	NaN	60328
66	132	1826	10.0	11.0	NaN	NaN	60328
67	133	1848	5.0	16.0	NaN	NaN	60328
68	135	1864	1.0	3.0	NaN	NaN	60328
69	136	1875	1.0	29.0	NaN	NaN	60328
70	137	1901	5.0	22.0	NaN	NaN	60328
71	138	1919	5.0	19.0	NaN	NaN	60328
72	140	1951	8.0	31.0	NaN	NaN	60328

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
73	141	1966	4.0	26.0	NaN	NaN	60328
74	144	1990	2.0	10.0	NaN	NaN	60328
75	159	1860	4.0	NaN	NaN	NaN	60330
76	167	1885	4.0	18.0	NaN	NaN	60330
77	176	1895	5.0	22.0	NaN	NaN	60330
78	191	1909	9.0	NaN	NaN	NaN	60330
79	193	1911	11.0	8.0	NaN	NaN	60330
80	195	1913	6.0	23.0	NaN	NaN	60330
81	196	1941	9.0	21.0	NaN	NaN	60330
82	198	1946	2.0	NaN	NaN	NaN	60330
83	199	1946	10.0	29.0	NaN	NaN	60330
84	200	1950	8.0	28.0	NaN	NaN	60330
85	201	1963	5.0	5.0	NaN	NaN	60330
86	202	1967	8.0	31.0	NaN	NaN	60330
87	203	1985	5.0	10.0	NaN	NaN	60330
88	204	1978	9.0	19.0	NaN	NaN	60330
89	205	1976	8.0	31.0	NaN	NaN	60330
90	206	1981	3.0	29.0	NaN	NaN	60330
91	261	1843	8.0	NaN	NaN	NaN	60332
92	269	1869	8.0	NaN	NaN	NaN	60332
93	294	1593	NaN	NaN	NaN	NaN	60334
94	295	1597	NaN	NaN	NaN	NaN	60334
95	296	1638	NaN	NaN	NaN	NaN	60334
96	297	1730	NaN	NaN	NaN	NaN	60334
97	298	1817	NaN	NaN	NaN	NaN	60334
98	333	1817	1.0	24.0	NaN	NaN	60335
99	345	1963	9.0	5.0	NaN	NaN	60401
100	351	1963	3.0	18.0	1942.0	NaN	60402
101	353	1963	5.0	16.0	NaN	4292.0	60402
102	367	1815	4.0	10.0	613.0	NaN	60404
103	389	1969	1.0	28.0	NaN	NaN	60411
104	394	1928	8.0	4.0	1607.0	NaN	60415

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
105	395	1964	1.0	1.0	NaN	NaN	60415
106	397	1973	1.0	NaN	NaN	NaN	60415
107	398	1981	9.0	5.0	NaN	NaN	60415
108	410	1869	7.0	7.0	NaN	NaN	60418
109	411	1907	9.0	28.0	NaN	NaN	60418
110	439	1870	NaN	NaN	NaN	NaN	60425
111	441	1948	4.0	7.0	NaN	NaN	60425
112	445	1973	12.0	NaN	5761.0	NaN	60425
113	447	1983	8.0	17.0	4702.0	NaN	60425
114	450	1953	6.0	NaN	NaN	NaN	60427
115	457	1659	11.0	11.0	285.0	NaN	60505
116	458	1660	2.0	NaN	NaN	NaN	60505
117	466	1692	6.0	4.0	NaN	NaN	60507
118	472	1598	NaN	NaN	NaN	NaN	60509
119	474	1615	3.0	NaN	NaN	NaN	60509
120	477	1694	11.0	20.0	NaN	NaN	60509
121	482	1820	6.0	11.0	NaN	NaN	60509
122	486	1988	5.0	9.0	NaN	NaN	60509
123	488	1983	7.0	23.0	NaN	NaN	60601
124	489	1845	2.0	8.0	741.0	1865.0	60603
125	525	1958	7.0	12.0	NaN	NaN	60611
126	530	1870	8.0	NaN	NaN	NaN	60701
127	533	1871	3.0	3.0	1022.0	NaN	60701
128	552	1940	6.0	20.0	NaN	NaN	60702

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
129	561	1974	2.0	11.0	NaN	NaN	60702
130	562	1976	9.0	15.0	NaN	NaN	60702
131	569	1889	9.0	6.0	1181.0	6309.0	60703
132	578	1711	12.0	11.0	NaN	NaN	60704
133	580	1812	8.0	6.0	NaN	NaN	60704
134	581	1856	3.0	2.0	859.0	1995.0	60704
135	584	1892	6.0	7.0	1197.0	NaN	60704
136	585	1913	3.0	14.0	1407.0	2973.0	60704
137	589	1966	8.0	12.0	NaN	NaN	60704
138	594	1550	11.0	NaN	NaN	NaN	60801
139	599	1673	5.0	20.0	312.0	NaN	60804
140	607	1608	7.0	18.0	222.0	NaN	60806
141	617	1772	5.0	9.0	504.0	NaN	60806
142	618	1773	2.0	2.0	NaN	NaN	60806
143	619	1775	NaN	NaN	NaN	NaN	60806
144	627	1838	2.0	26.0	NaN	NaN	60806
145	629	1840	2.0	2.0	718.0	NaN	60806
146	643	1871	8.0	7.0	NaN	NaN	60806
147	657	1962	12.0	31.0	NaN	NaN	60806
148	663	1550	NaN	NaN	5863.0	NaN	60807
149	664	1646	7.0	19.0	5754.0	10519.0	60807
150	665	1760	NaN	NaN	NaN	NaN	60807
151	666	1861	12.0	29.0	NaN	NaN	60807
152	669	1890	6.0	29.0	NaN	NaN	60807
153	677	1873	1.0	16.0	NaN	NaN	70106
154	679	1871	4.0	30.0	1020.0	2179.0	70108
155	680	1950	9.0	15.0	NaN	NaN	70108
156	693	1996	8.0	10.0	NaN	NaN	70202
157	697	1933	12.0	25.0	1666.0	NaN	70301
158	698	1978	7.0	29.0	NaN	NaN	70301

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
159	704	1766	7.0	20.0	NaN	NaN	70303
160	706	1800	10.0	30.0	NaN	NaN	70303
161	707	1814	2.0	1.0	NaN	NaN	70303
162	712	1853	7.0	13.0	NaN	NaN	70303
163	714	1858	1.0	NaN	NaN	NaN	70303
164	716	1871	12.0	8.0	NaN	NaN	70303
165	724	1887	3.0	9.0	NaN	NaN	70303
166	730	1897	5.0	23.0	NaN	NaN	70303
167	732	1928	1.0	NaN	NaN	NaN	70303
168	733	1938	6.0	5.0	NaN	NaN	70303
169	735	1947	1.0	7.0	NaN	NaN	70303
170	736	1968	4.0	21.0	NaN	NaN	70303
171	739	1984	9.0	9.0	NaN	NaN	70303
172	745	1572	NaN	NaN	NaN	NaN	70307
173	752	1716	9.0	24.0	387.0	1205.0	70307
174	753	1749	8.0	11.0	431.0	1304.0	70307
175	754	1754	11.0	28.0	2853.0	1328.0	70307
176	756	1874	7.0	19.0	NaN	NaN	70307
177	761	1911	1.0	30.0	1392.0	8391.0	70307
178	762	1965	9.0	28.0	1977.0	NaN	70307
179	777	1969	3.0	21.0	2872.0	NaN	70402
180	787	1853	10.0	29.0	809.0	NaN	80103
181	821	1841	4.0	NaN	NaN	NaN	80205
182	823	1933	12.0	24.0	NaN	NaN	80205
183	843	764	NaN	NaN	NaN	NaN	80208
184	844	766	7.0	20.0	61.0	7341.0	80208
185	846	1471	11.0	3.0	NaN	NaN	80208
186	858	1779	11.0	8.0	NaN	NaN	80208
187	859	1781	4.0	11.0	529.0	NaN	80208
188	867	1914	1.0	12.0	1412.0	2994.0	80208

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
189	874	1946	1.0	NaN	NaN	NaN	80208
190	878	1955	10.0	13.0	NaN	NaN	80208
191	904	1566	10.0	31.0	NaN	NaN	80209
192	915	1716	11.0	9.0	NaN	NaN	80209
193	916	1717	2.0	7.0	NaN	NaN	80209
194	932	1792	5.0	21.0	568.0	1515.0	80210
195	933	1991	6.0	3.0	NaN	NaN	80210
196	949	1331	12.0	NaN	NaN	NaN	80211
197	960	1485	1.0	NaN	NaN	NaN	80211
198	990	1826	10.0	NaN	NaN	NaN	80211
199	994	1854	2.0	NaN	NaN	NaN	80211

TASK 1.2: 1 Liner Shenaniganz (7 points)

We're going to tidy up the **new dataframe** a little more with some more advanced 1 liner code.

Read the directions carefully and code your answer with only one line of code.

For this section, keep the method of display that is already in the box. Write your code as indicated.

YOU CAN'T USE ONE LINE LOOPS OR ANY KIND OF LOOP.

1.2.1: *In one line of code and **using only one single function call**, **drop any row** that contains **NaN** in **any one** of the columns indicating a measure of **time**.*

In [79]:
```
new_df = new_df.dropna(subset= ['year', 'month', 'day'])
new_df # KEEP THIS. It will display the whole dataframe.
```

Out[79]:

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
0	1	1169	2.0	4.0	2852.0	421.0	10106
3	4	1888	3.0	13.0	1175.0	NaN	50107
5	6	1832	11.0	1.0	NaN	NaN	10106
6	7	1977	1.0	10.0	NaN	NaN	20303
8	9	1779	8.0	8.0	NaN	NaN	10102
10	11	1907	8.0	4.0	NaN	NaN	201112
11	12	1905	3.0	10.0	NaN	NaN	10102
12	13	1986	7.0	24.0	NaN	NaN	10104
13	14	1536	3.0	23.0	NaN	NaN	10106
14	15	1904	2.0	25.0	NaN	NaN	30301
16	17	1737	5.0	20.0	NaN	NaN	10102
17	18	1917	4.0	1.0	NaN	NaN	40105
19	20	1979	3.0	8.0	NaN	NaN	50103
20	21	1970	3.0	2.0	NaN	NaN	10101
21	22	1972	6.0	9.0	NaN	NaN	30302
22	23	1682	8.0	12.0	NaN	NaN	10102
23	24	1805	8.0	11.0	NaN	NaN	10102
24	25	1919	5.0	22.0	1476.0	NaN	10104
25	26	1930	9.0	11.0	4218.0	NaN	10104
26	27	1903	8.0	30.0	NaN	NaN	40105
27	28	1928	11.0	2.0	NaN	NaN	10106
28	29	1872	4.0	24.0	NaN	NaN	10102
29	30	1979	7.0	5.0	NaN	NaN	10106

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
30	31	1987	4.0	17.0	NaN	NaN	10106
31	32	1914	9.0	10.0	NaN	NaN	40104
32	33	1886	8.0	31.0	NaN	NaN	40311
33	34	1912	12.0	3.0	NaN	NaN	20302
34	35	1895	6.0	17.0	NaN	NaN	50103
35	36	1538	9.0	29.0	NaN	NaN	10101
36	37	1853	6.0	24.0	NaN	NaN	40311
37	38	1954	8.0	3.0	NaN	NaN	50101
38	39	1944	3.0	27.0	NaN	NaN	10102
39	40	1984	8.0	15.0	5807.0	NaN	20403
41	42	1928	1.0	23.0	NaN	NaN	10204
42	43	1631	2.0	14.0	NaN	920.0	201141
43	44	1843	11.0	17.0	NaN	NaN	10106
44	45	1944	12.0	4.0	NaN	NaN	60320
45	49	1979	2.0	20.0	NaN	NaN	60320
47	64	1672	8.0	4.0	NaN	NaN	60325
48	69	1822	12.0	27.0	NaN	NaN	60325
49	71	1832	12.0	25.0	NaN	NaN	60325
50	81	1872	4.0	15.0	NaN	NaN	60325
51	82	1872	11.0	3.0	NaN	NaN	60325
53	92	1920	7.0	25.0	NaN	NaN	60325
54	94	1930	12.0	18.0	NaN	NaN	60325
55	100	1953	3.0	23.0	NaN	NaN	60325
56	104	1961	5.0	8.0	NaN	NaN	60325

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
57	105	1969	1.0	7.0	NaN	NaN	60325
58	111	1986	10.0	15.0	NaN	NaN	60325
59	112	1994	11.0	22.0	NaN	NaN	60325
65	128	1716	7.0	20.0	NaN	NaN	60328
66	132	1826	10.0	11.0	NaN	NaN	60328
67	133	1848	5.0	16.0	NaN	NaN	60328
68	135	1864	1.0	3.0	NaN	NaN	60328
69	136	1875	1.0	29.0	NaN	NaN	60328
70	137	1901	5.0	22.0	NaN	NaN	60328
71	138	1919	5.0	19.0	NaN	NaN	60328
72	140	1951	8.0	31.0	NaN	NaN	60328
73	141	1966	4.0	26.0	NaN	NaN	60328
74	144	1990	2.0	10.0	NaN	NaN	60328
76	167	1885	4.0	18.0	NaN	NaN	60330
77	176	1895	5.0	22.0	NaN	NaN	60330
79	193	1911	11.0	8.0	NaN	NaN	60330
80	195	1913	6.0	23.0	NaN	NaN	60330
81	196	1941	9.0	21.0	NaN	NaN	60330
83	199	1946	10.0	29.0	NaN	NaN	60330
84	200	1950	8.0	28.0	NaN	NaN	60330
85	201	1963	5.0	5.0	NaN	NaN	60330
86	202	1967	8.0	31.0	NaN	NaN	60330
87	203	1985	5.0	10.0	NaN	NaN	60330
88	204	1978	9.0	19.0	NaN	NaN	60330
89	205	1976	8.0	31.0	NaN	NaN	60330
90	206	1981	3.0	29.0	NaN	NaN	60330
98	333	1817	1.0	24.0	NaN	NaN	60335
99	345	1963	9.0	5.0	NaN	NaN	60401
100	351	1963	3.0	18.0	1942.0	NaN	60402
101	353	1963	5.0	16.0	NaN	4292.0	60402
102	367	1815	4.0	10.0	613.0	NaN	60404
103	389	1969	1.0	28.0	NaN	NaN	60411

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
104	394	1928	8.0	4.0	1607.0	NaN	60415
105	395	1964	1.0	1.0	NaN	NaN	60415
107	398	1981	9.0	5.0	NaN	NaN	60415
108	410	1869	7.0	7.0	NaN	NaN	60418
109	411	1907	9.0	28.0	NaN	NaN	60418
111	441	1948	4.0	7.0	NaN	NaN	60425
113	447	1983	8.0	17.0	4702.0	NaN	60425
115	457	1659	11.0	11.0	285.0	NaN	60505
117	466	1692	6.0	4.0	NaN	NaN	60507
120	477	1694	11.0	20.0	NaN	NaN	60509
121	482	1820	6.0	11.0	NaN	NaN	60509
122	486	1988	5.0	9.0	NaN	NaN	60509
123	488	1983	7.0	23.0	NaN	NaN	60601
124	489	1845	2.0	8.0	741.0	1865.0	60603
125	525	1958	7.0	12.0	NaN	NaN	60611
127	533	1871	3.0	3.0	1022.0	NaN	60701
128	552	1940	6.0	20.0	NaN	NaN	60702
129	561	1974	2.0	11.0	NaN	NaN	60702
130	562	1976	9.0	15.0	NaN	NaN	60702
131	569	1889	9.0	6.0	1181.0	6309.0	60703
132	578	1711	12.0	11.0	NaN	NaN	60704
133	580	1812	8.0	6.0	NaN	NaN	60704
134	581	1856	3.0	2.0	859.0	1995.0	60704
135	584	1892	6.0	7.0	1197.0	NaN	60704
136	585	1913	3.0	14.0	1407.0	2973.0	60704
137	589	1966	8.0	12.0	NaN	NaN	60704
139	599	1673	5.0	20.0	312.0	NaN	60804

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
140	607	1608	7.0	18.0	222.0	NaN	60806
141	617	1772	5.0	9.0	504.0	NaN	60806
142	618	1773	2.0	2.0	NaN	NaN	60806
144	627	1838	2.0	26.0	NaN	NaN	60806
145	629	1840	2.0	2.0	718.0	NaN	60806
146	643	1871	8.0	7.0	NaN	NaN	60806
147	657	1962	12.0	31.0	NaN	NaN	60806
149	664	1646	7.0	19.0	5754.0	10519.0	60807
151	666	1861	12.0	29.0	NaN	NaN	60807
152	669	1890	6.0	29.0	NaN	NaN	60807
153	677	1873	1.0	16.0	NaN	NaN	70106
154	679	1871	4.0	30.0	1020.0	2179.0	70108
155	680	1950	9.0	15.0	NaN	NaN	70108
156	693	1996	8.0	10.0	NaN	NaN	70202
157	697	1933	12.0	25.0	1666.0	NaN	70301
158	698	1978	7.0	29.0	NaN	NaN	70301
159	704	1766	7.0	20.0	NaN	NaN	70303
160	706	1800	10.0	30.0	NaN	NaN	70303
161	707	1814	2.0	1.0	NaN	NaN	70303
162	712	1853	7.0	13.0	NaN	NaN	70303
164	716	1871	12.0	8.0	NaN	NaN	70303
165	724	1887	3.0	9.0	NaN	NaN	70303
166	730	1897	5.0	23.0	NaN	NaN	70303
168	733	1938	6.0	5.0	NaN	NaN	70303
169	735	1947	1.0	7.0	NaN	NaN	70303
170	736	1968	4.0	21.0	NaN	NaN	70303
171	739	1984	9.0	9.0	NaN	NaN	70303
173	752	1716	9.0	24.0	387.0	1205.0	70307
174	753	1749	8.0	11.0	431.0	1304.0	70307
175	754	1754	11.0	28.0	2853.0	1328.0	70307
176	756	1874	7.0	19.0	NaN	NaN	70307
177	761	1911	1.0	30.0	1392.0	8391.0	70307

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
178	762	1965	9.0	28.0	1977.0	NaN	70307
179	777	1969	3.0	21.0	2872.0	NaN	70402
180	787	1853	10.0	29.0	809.0	NaN	80103
182	823	1933	12.0	24.0	NaN	NaN	80205
184	844	766	7.0	20.0	61.0	7341.0	80208
185	846	1471	11.0	3.0	NaN	NaN	80208
186	858	1779	11.0	8.0	NaN	NaN	80208
187	859	1781	4.0	11.0	529.0	NaN	80208
188	867	1914	1.0	12.0	1412.0	2994.0	80208
190	878	1955	10.0	13.0	NaN	NaN	80208
191	904	1566	10.0	31.0	NaN	NaN	80209
192	915	1716	11.0	9.0	NaN	NaN	80209
193	916	1717	2.0	7.0	NaN	NaN	80209
194	932	1792	5.0	21.0	568.0	1515.0	80210
195	933	1991	6.0	3.0	NaN	NaN	80210

1.2.2: *In one line of code, **change** the **index column** of the dataframe so that it has **1-based indexing**.*

In [80]:
```
new_df.index = range(1, len(new_df) + 1)
new_df # KEEP THIS. It will display the whole dataframe.
```

Out[80]:

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
1	1	1169	2.0	4.0	2852.0	421.0	10106
2	4	1888	3.0	13.0	1175.0	NaN	50107
3	6	1832	11.0	1.0	NaN	NaN	10106
4	7	1977	1.0	10.0	NaN	NaN	20303
5	9	1779	8.0	8.0	NaN	NaN	10102
6	11	1907	8.0	4.0	NaN	NaN	201112
7	12	1905	3.0	10.0	NaN	NaN	10102
8	13	1986	7.0	24.0	NaN	NaN	10104
9	14	1536	3.0	23.0	NaN	NaN	10106
10	15	1904	2.0	25.0	NaN	NaN	30301
11	17	1737	5.0	20.0	NaN	NaN	10102
12	18	1917	4.0	1.0	NaN	NaN	40105
13	20	1979	3.0	8.0	NaN	NaN	50103
14	21	1970	3.0	2.0	NaN	NaN	10101
15	22	1972	6.0	9.0	NaN	NaN	30302
16	23	1682	8.0	12.0	NaN	NaN	10102
17	24	1805	8.0	11.0	NaN	NaN	10102
18	25	1919	5.0	22.0	1476.0	NaN	10104
19	26	1930	9.0	11.0	4218.0	NaN	10104
20	27	1903	8.0	30.0	NaN	NaN	40105
21	28	1928	11.0	2.0	NaN	NaN	10106
22	29	1872	4.0	24.0	NaN	NaN	10102
23	30	1979	7.0	5.0	NaN	NaN	10106

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
24	31	1987	4.0	17.0	NaN	NaN	10106
25	32	1914	9.0	10.0	NaN	NaN	40104
26	33	1886	8.0	31.0	NaN	NaN	40311
27	34	1912	12.0	3.0	NaN	NaN	20302
28	35	1895	6.0	17.0	NaN	NaN	50103
29	36	1538	9.0	29.0	NaN	NaN	10101
30	37	1853	6.0	24.0	NaN	NaN	40311
31	38	1954	8.0	3.0	NaN	NaN	50101
32	39	1944	3.0	27.0	NaN	NaN	10102
33	40	1984	8.0	15.0	5807.0	NaN	20403
34	42	1928	1.0	23.0	NaN	NaN	10204
35	43	1631	2.0	14.0	NaN	920.0	201141
36	44	1843	11.0	17.0	NaN	NaN	10106
37	45	1944	12.0	4.0	NaN	NaN	60320
38	49	1979	2.0	20.0	NaN	NaN	60320
39	64	1672	8.0	4.0	NaN	NaN	60325
40	69	1822	12.0	27.0	NaN	NaN	60325
41	71	1832	12.0	25.0	NaN	NaN	60325
42	81	1872	4.0	15.0	NaN	NaN	60325
43	82	1872	11.0	3.0	NaN	NaN	60325
44	92	1920	7.0	25.0	NaN	NaN	60325
45	94	1930	12.0	18.0	NaN	NaN	60325
46	100	1953	3.0	23.0	NaN	NaN	60325
47	104	1961	5.0	8.0	NaN	NaN	60325

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
48	105	1969	1.0	7.0	NaN	NaN	60325
49	111	1986	10.0	15.0	NaN	NaN	60325
50	112	1994	11.0	22.0	NaN	NaN	60325
51	128	1716	7.0	20.0	NaN	NaN	60328
52	132	1826	10.0	11.0	NaN	NaN	60328
53	133	1848	5.0	16.0	NaN	NaN	60328
54	135	1864	1.0	3.0	NaN	NaN	60328
55	136	1875	1.0	29.0	NaN	NaN	60328
56	137	1901	5.0	22.0	NaN	NaN	60328
57	138	1919	5.0	19.0	NaN	NaN	60328
58	140	1951	8.0	31.0	NaN	NaN	60328
59	141	1966	4.0	26.0	NaN	NaN	60328
60	144	1990	2.0	10.0	NaN	NaN	60328
61	167	1885	4.0	18.0	NaN	NaN	60330
62	176	1895	5.0	22.0	NaN	NaN	60330
63	193	1911	11.0	8.0	NaN	NaN	60330
64	195	1913	6.0	23.0	NaN	NaN	60330
65	196	1941	9.0	21.0	NaN	NaN	60330
66	199	1946	10.0	29.0	NaN	NaN	60330
67	200	1950	8.0	28.0	NaN	NaN	60330
68	201	1963	5.0	5.0	NaN	NaN	60330
69	202	1967	8.0	31.0	NaN	NaN	60330
70	203	1985	5.0	10.0	NaN	NaN	60330
71	204	1978	9.0	19.0	NaN	NaN	60330
72	205	1976	8.0	31.0	NaN	NaN	60330
73	206	1981	3.0	29.0	NaN	NaN	60330
74	333	1817	1.0	24.0	NaN	NaN	60335
75	345	1963	9.0	5.0	NaN	NaN	60401
76	351	1963	3.0	18.0	1942.0	NaN	60402
77	353	1963	5.0	16.0	NaN	4292.0	60402
78	367	1815	4.0	10.0	613.0	NaN	60404
79	389	1969	1.0	28.0	NaN	NaN	60411

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
80	394	1928	8.0	4.0	1607.0	NaN	60415
81	395	1964	1.0	1.0	NaN	NaN	60415
82	398	1981	9.0	5.0	NaN	NaN	60415
83	410	1869	7.0	7.0	NaN	NaN	60418
84	411	1907	9.0	28.0	NaN	NaN	60418
85	441	1948	4.0	7.0	NaN	NaN	60425
86	447	1983	8.0	17.0	4702.0	NaN	60425
87	457	1659	11.0	11.0	285.0	NaN	60505
88	466	1692	6.0	4.0	NaN	NaN	60507
89	477	1694	11.0	20.0	NaN	NaN	60509
90	482	1820	6.0	11.0	NaN	NaN	60509
91	486	1988	5.0	9.0	NaN	NaN	60509
92	488	1983	7.0	23.0	NaN	NaN	60601
93	489	1845	2.0	8.0	741.0	1865.0	60603
94	525	1958	7.0	12.0	NaN	NaN	60611
95	533	1871	3.0	3.0	1022.0	NaN	60701
96	552	1940	6.0	20.0	NaN	NaN	60702
97	561	1974	2.0	11.0	NaN	NaN	60702
98	562	1976	9.0	15.0	NaN	NaN	60702
99	569	1889	9.0	6.0	1181.0	6309.0	60703
100	578	1711	12.0	11.0	NaN	NaN	60704
101	580	1812	8.0	6.0	NaN	NaN	60704
102	581	1856	3.0	2.0	859.0	1995.0	60704
103	584	1892	6.0	7.0	1197.0	NaN	60704
104	585	1913	3.0	14.0	1407.0	2973.0	60704
105	589	1966	8.0	12.0	NaN	NaN	60704
106	599	1673	5.0	20.0	312.0	NaN	60804

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
107	607	1608	7.0	18.0	222.0	NaN	60806
108	617	1772	5.0	9.0	504.0	NaN	60806
109	618	1773	2.0	2.0	NaN	NaN	60806
110	627	1838	2.0	26.0	NaN	NaN	60806
111	629	1840	2.0	2.0	718.0	NaN	60806
112	643	1871	8.0	7.0	NaN	NaN	60806
113	657	1962	12.0	31.0	NaN	NaN	60806
114	664	1646	7.0	19.0	5754.0	10519.0	60807
115	666	1861	12.0	29.0	NaN	NaN	60807
116	669	1890	6.0	29.0	NaN	NaN	60807
117	677	1873	1.0	16.0	NaN	NaN	70106
118	679	1871	4.0	30.0	1020.0	2179.0	70108
119	680	1950	9.0	15.0	NaN	NaN	70108
120	693	1996	8.0	10.0	NaN	NaN	70202
121	697	1933	12.0	25.0	1666.0	NaN	70301
122	698	1978	7.0	29.0	NaN	NaN	70301
123	704	1766	7.0	20.0	NaN	NaN	70303
124	706	1800	10.0	30.0	NaN	NaN	70303
125	707	1814	2.0	1.0	NaN	NaN	70303
126	712	1853	7.0	13.0	NaN	NaN	70303
127	716	1871	12.0	8.0	NaN	NaN	70303
128	724	1887	3.0	9.0	NaN	NaN	70303
129	730	1897	5.0	23.0	NaN	NaN	70303
130	733	1938	6.0	5.0	NaN	NaN	70303
131	735	1947	1.0	7.0	NaN	NaN	70303
132	736	1968	4.0	21.0	NaN	NaN	70303
133	739	1984	9.0	9.0	NaN	NaN	70303
134	752	1716	9.0	24.0	387.0	1205.0	70307
135	753	1749	8.0	11.0	431.0	1304.0	70307
136	754	1754	11.0	28.0	2853.0	1328.0	70307
137	756	1874	7.0	19.0	NaN	NaN	70307
138	761	1911	1.0	30.0	1392.0	8391.0	70307

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
139	762	1965	9.0	28.0	1977.0	NaN	70307
140	777	1969	3.0	21.0	2872.0	NaN	70402
141	787	1853	10.0	29.0	809.0	NaN	80103
142	823	1933	12.0	24.0	NaN	NaN	80205
143	844	766	7.0	20.0	61.0	7341.0	80208
144	846	1471	11.0	3.0	NaN	NaN	80208
145	858	1779	11.0	8.0	NaN	NaN	80208
146	859	1781	4.0	11.0	529.0	NaN	80208
147	867	1914	1.0	12.0	1412.0	2994.0	80208
148	878	1955	10.0	13.0	NaN	NaN	80208
149	904	1566	10.0	31.0	NaN	NaN	80209
150	915	1716	11.0	9.0	NaN	NaN	80209
151	916	1717	2.0	7.0	NaN	NaN	80209
152	932	1792	5.0	21.0	568.0	1515.0	80210
153	933	1991	6.0	3.0	NaN	NaN	80210

The **deathsTotal** and **deaths** columns have approximations of the same data with alternating NaNs in each.

1.2.3: *In one line of code, make a **new column** called **'totalDeaths'** that takes the **max** of the values given between those **two*** *columns.

- If there is **NaN** in ***one column*** and a **numerical** value in the **other**, it will ***take the numerical value***.
- **Only** if there are **NaNs** in ***both columns***, the **new column will have NaN.**

In [81]:
```
new_df = new_df.assign(totalDeaths = new_df[['deaths','deathsTotal']].max(ax
new_df # KEEP THIS. It will display the whole dataframe.
```

Out[81]:

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
1	1	1169	2.0	4.0	2852.0	421.0	10106
2	4	1888	3.0	13.0	1175.0	NaN	50107
3	6	1832	11.0	1.0	NaN	NaN	10106
4	7	1977	1.0	10.0	NaN	NaN	20303
5	9	1779	8.0	8.0	NaN	NaN	10102
6	11	1907	8.0	4.0	NaN	NaN	201112
7	12	1905	3.0	10.0	NaN	NaN	10102
8	13	1986	7.0	24.0	NaN	NaN	10104
9	14	1536	3.0	23.0	NaN	NaN	10106
10	15	1904	2.0	25.0	NaN	NaN	30301
11	17	1737	5.0	20.0	NaN	NaN	10102
12	18	1917	4.0	1.0	NaN	NaN	40105
13	20	1979	3.0	8.0	NaN	NaN	50103
14	21	1970	3.0	2.0	NaN	NaN	10101
15	22	1972	6.0	9.0	NaN	NaN	30302
16	23	1682	8.0	12.0	NaN	NaN	10102
17	24	1805	8.0	11.0	NaN	NaN	10102
18	25	1919	5.0	22.0	1476.0	NaN	10104
19	26	1930	9.0	11.0	4218.0	NaN	10104
20	27	1903	8.0	30.0	NaN	NaN	40105
21	28	1928	11.0	2.0	NaN	NaN	10106
22	29	1872	4.0	24.0	NaN	NaN	10102
23	30	1979	7.0	5.0	NaN	NaN	10106

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
24	31	1987	4.0	17.0	NaN	NaN	10106
25	32	1914	9.0	10.0	NaN	NaN	40104
26	33	1886	8.0	31.0	NaN	NaN	40311
27	34	1912	12.0	3.0	NaN	NaN	20302
28	35	1895	6.0	17.0	NaN	NaN	50103
29	36	1538	9.0	29.0	NaN	NaN	10101
30	37	1853	6.0	24.0	NaN	NaN	40311
31	38	1954	8.0	3.0	NaN	NaN	50101
32	39	1944	3.0	27.0	NaN	NaN	10102
33	40	1984	8.0	15.0	5807.0	NaN	20403
34	42	1928	1.0	23.0	NaN	NaN	10204
35	43	1631	2.0	14.0	NaN	920.0	201141
36	44	1843	11.0	17.0	NaN	NaN	10106
37	45	1944	12.0	4.0	NaN	NaN	60320
38	49	1979	2.0	20.0	NaN	NaN	60320
39	64	1672	8.0	4.0	NaN	NaN	60325
40	69	1822	12.0	27.0	NaN	NaN	60325
41	71	1832	12.0	25.0	NaN	NaN	60325
42	81	1872	4.0	15.0	NaN	NaN	60325
43	82	1872	11.0	3.0	NaN	NaN	60325
44	92	1920	7.0	25.0	NaN	NaN	60325
45	94	1930	12.0	18.0	NaN	NaN	60325
46	100	1953	3.0	23.0	NaN	NaN	60325
47	104	1961	5.0	8.0	NaN	NaN	60325

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
48	105	1969	1.0	7.0	NaN	NaN	60325
49	111	1986	10.0	15.0	NaN	NaN	60325
50	112	1994	11.0	22.0	NaN	NaN	60325
51	128	1716	7.0	20.0	NaN	NaN	60328
52	132	1826	10.0	11.0	NaN	NaN	60328
53	133	1848	5.0	16.0	NaN	NaN	60328
54	135	1864	1.0	3.0	NaN	NaN	60328
55	136	1875	1.0	29.0	NaN	NaN	60328
56	137	1901	5.0	22.0	NaN	NaN	60328
57	138	1919	5.0	19.0	NaN	NaN	60328
58	140	1951	8.0	31.0	NaN	NaN	60328
59	141	1966	4.0	26.0	NaN	NaN	60328
60	144	1990	2.0	10.0	NaN	NaN	60328
61	167	1885	4.0	18.0	NaN	NaN	60330
62	176	1895	5.0	22.0	NaN	NaN	60330
63	193	1911	11.0	8.0	NaN	NaN	60330
64	195	1913	6.0	23.0	NaN	NaN	60330
65	196	1941	9.0	21.0	NaN	NaN	60330
66	199	1946	10.0	29.0	NaN	NaN	60330
67	200	1950	8.0	28.0	NaN	NaN	60330
68	201	1963	5.0	5.0	NaN	NaN	60330
69	202	1967	8.0	31.0	NaN	NaN	60330
70	203	1985	5.0	10.0	NaN	NaN	60330
71	204	1978	9.0	19.0	NaN	NaN	60330
72	205	1976	8.0	31.0	NaN	NaN	60330
73	206	1981	3.0	29.0	NaN	NaN	60330
74	333	1817	1.0	24.0	NaN	NaN	60335
75	345	1963	9.0	5.0	NaN	NaN	60401
76	351	1963	3.0	18.0	1942.0	NaN	60402
77	353	1963	5.0	16.0	NaN	4292.0	60402
78	367	1815	4.0	10.0	613.0	NaN	60404
79	389	1969	1.0	28.0	NaN	NaN	60411

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
80	394	1928	8.0	4.0	1607.0	NaN	60415
81	395	1964	1.0	1.0	NaN	NaN	60415
82	398	1981	9.0	5.0	NaN	NaN	60415
83	410	1869	7.0	7.0	NaN	NaN	60418
84	411	1907	9.0	28.0	NaN	NaN	60418
85	441	1948	4.0	7.0	NaN	NaN	60425
86	447	1983	8.0	17.0	4702.0	NaN	60425
87	457	1659	11.0	11.0	285.0	NaN	60505
88	466	1692	6.0	4.0	NaN	NaN	60507
89	477	1694	11.0	20.0	NaN	NaN	60509
90	482	1820	6.0	11.0	NaN	NaN	60509
91	486	1988	5.0	9.0	NaN	NaN	60509
92	488	1983	7.0	23.0	NaN	NaN	60601
93	489	1845	2.0	8.0	741.0	1865.0	60603
94	525	1958	7.0	12.0	NaN	NaN	60611
95	533	1871	3.0	3.0	1022.0	NaN	60701
96	552	1940	6.0	20.0	NaN	NaN	60702
97	561	1974	2.0	11.0	NaN	NaN	60702
98	562	1976	9.0	15.0	NaN	NaN	60702
99	569	1889	9.0	6.0	1181.0	6309.0	60703
100	578	1711	12.0	11.0	NaN	NaN	60704
101	580	1812	8.0	6.0	NaN	NaN	60704
102	581	1856	3.0	2.0	859.0	1995.0	60704
103	584	1892	6.0	7.0	1197.0	NaN	60704
104	585	1913	3.0	14.0	1407.0	2973.0	60704
105	589	1966	8.0	12.0	NaN	NaN	60704
106	599	1673	5.0	20.0	312.0	NaN	60804

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
107	607	1608	7.0	18.0	222.0	NaN	60806
108	617	1772	5.0	9.0	504.0	NaN	60806
109	618	1773	2.0	2.0	NaN	NaN	60806
110	627	1838	2.0	26.0	NaN	NaN	60806
111	629	1840	2.0	2.0	718.0	NaN	60806
112	643	1871	8.0	7.0	NaN	NaN	60806
113	657	1962	12.0	31.0	NaN	NaN	60806
114	664	1646	7.0	19.0	5754.0	10519.0	60807
115	666	1861	12.0	29.0	NaN	NaN	60807
116	669	1890	6.0	29.0	NaN	NaN	60807
117	677	1873	1.0	16.0	NaN	NaN	70106
118	679	1871	4.0	30.0	1020.0	2179.0	70108
119	680	1950	9.0	15.0	NaN	NaN	70108
120	693	1996	8.0	10.0	NaN	NaN	70202
121	697	1933	12.0	25.0	1666.0	NaN	70301
122	698	1978	7.0	29.0	NaN	NaN	70301
123	704	1766	7.0	20.0	NaN	NaN	70303
124	706	1800	10.0	30.0	NaN	NaN	70303
125	707	1814	2.0	1.0	NaN	NaN	70303
126	712	1853	7.0	13.0	NaN	NaN	70303
127	716	1871	12.0	8.0	NaN	NaN	70303
128	724	1887	3.0	9.0	NaN	NaN	70303
129	730	1897	5.0	23.0	NaN	NaN	70303
130	733	1938	6.0	5.0	NaN	NaN	70303
131	735	1947	1.0	7.0	NaN	NaN	70303
132	736	1968	4.0	21.0	NaN	NaN	70303
133	739	1984	9.0	9.0	NaN	NaN	70303
134	752	1716	9.0	24.0	387.0	1205.0	70307
135	753	1749	8.0	11.0	431.0	1304.0	70307
136	754	1754	11.0	28.0	2853.0	1328.0	70307
137	756	1874	7.0	19.0	NaN	NaN	70307
138	761	1911	1.0	30.0	1392.0	8391.0	70307

	id	year	month	day	tsunamiEventId	earthquakeEventId	volcanoLocationId
139	762	1965	9.0	28.0	1977.0	NaN	70307
140	777	1969	3.0	21.0	2872.0	NaN	70402
141	787	1853	10.0	29.0	809.0	NaN	80103
142	823	1933	12.0	24.0	NaN	NaN	80205
143	844	766	7.0	20.0	61.0	7341.0	80208
144	846	1471	11.0	3.0	NaN	NaN	80208
145	858	1779	11.0	8.0	NaN	NaN	80208
146	859	1781	4.0	11.0	529.0	NaN	80208
147	867	1914	1.0	12.0	1412.0	2994.0	80208
148	878	1955	10.0	13.0	NaN	NaN	80208
149	904	1566	10.0	31.0	NaN	NaN	80209
150	915	1716	11.0	9.0	NaN	NaN	80209
151	916	1717	2.0	7.0	NaN	NaN	80209
152	932	1792	5.0	21.0	568.0	1515.0	80210
153	933	1991	6.0	3.0	NaN	NaN	80210

TASK 1.3: Tailoring Time (10 Points)

It's pretty obvious that the year, month, and day look pretty weird in the dataset. We're going to have to do some hardcore cleaning on the time. We will learn more about data cleaning in class soon, but here we will perform some basic data cleaning.

- **We need to have only ONE column called "*date*" that contains the full date in the following format YYYY-MM-DD, not separated into three columns.**
- **Make sure there are no floating point values in the date.**
- **Sort the data from most recent to least.**
- **Remove the old columns and place the new column next to the 'id' column.**

YOU CAN USE MULTIPLE LINES OF CODE BUT CAN'T USE LOOPS OR HARDCODE.

Note: It is alright to have only a **maximum of 12 NaTs** for some dates that often go further back in time because the **datetime module** in Pandas has a year limit (unless otherwise guided).

In [46]:
```
new_df = new_df.assign(date = pd.to_datetime(new_df[['year', 'month', 'day']
new_df.insert(1, 'date', new_df.pop('date'))
new_df # KEEP THIS. It will display the whole dataframe.
```

Out[46]:

	id	date	tsunamiEventId	earthquakeEventId	volcanoLocationId	volcanoLocat
120	693	1996-08-10	NaN	NaN	70202	
50	112	1994-11-22	NaN	NaN	60325	
153	933	1991-06-03	NaN	NaN	80210	
60	144	1990-02-10	NaN	NaN	60328	
91	486	1988-05-09	NaN	NaN	60509	
24	31	1987-04-17	NaN	NaN	10106	
49	111	1986-10-15	NaN	NaN	60325	
8	13	1986-07-24	NaN	NaN	10104	
70	203	1985-05-10	NaN	NaN	60330	
133	739	1984-09-09	NaN	NaN	70303	
33	40	1984-08-15	5807.0	NaN	20403	
86	447	1983-08-17	4702.0	NaN	60425	
92	488	1983-07-23	NaN	NaN	60601	
82	398	1981-09-05	NaN	NaN	60415	
73	206	1981-03-29	NaN	NaN	60330	
23	30	1979-07-05	NaN	NaN	10106	
13	20	1979-03-08	NaN	NaN	50103	

	id	date	tsunamiEventId	earthquakeEventId	volcanoLocationId	volcanoLocat
38	49	1979-02-20	NaN	NaN	60320	
71	204	1978-09-19	NaN	NaN	60330	
122	698	1978-07-29	NaN	NaN	70301	
4	7	1977-01-10	NaN	NaN	20303	
98	562	1976-09-15	NaN	NaN	60702	
72	205	1976-08-31	NaN	NaN	60330	
97	561	1974-02-11	NaN	NaN	60702	
15	22	1972-06-09	NaN	NaN	30302	
14	21	1970-03-02	NaN	NaN	10101	
140	777	1969-03-21	2872.0	NaN	70402	
79	389	1969-01-28	NaN	NaN	60411	
48	105	1969-01-07	NaN	NaN	60325	
132	736	1968-04-21	NaN	NaN	70303	
69	202	1967-08-31	NaN	NaN	60330	
105	589	1966-08-12	NaN	NaN	60704	
59	141	1966-04-26	NaN	NaN	60328	
139	762	1965-09-28	1977.0	NaN	70307	
81	395	1964-01-01	NaN	NaN	60415	

	id	date	tsunamiEventId	earthquakeEventId	volcanoLocationId	volcanoLocat
75	345	1963-09-05	NaN	NaN	60401	
77	353	1963-05-16	NaN	4292.0	60402	
68	201	1963-05-05	NaN	NaN	60330	
76	351	1963-03-18	1942.0	NaN	60402	
113	657	1962-12-31	NaN	NaN	60806	
47	104	1961-05-08	NaN	NaN	60325	
94	525	1958-07-12	NaN	NaN	60611	
148	878	1955-10-13	NaN	NaN	80208	
31	38	1954-08-03	NaN	NaN	50101	
46	100	1953-03-23	NaN	NaN	60325	
58	140	1951-08-31	NaN	NaN	60328	
119	680	1950-09-15	NaN	NaN	70108	
67	200	1950-08-28	NaN	NaN	60330	
85	441	1948-04-07	NaN	NaN	60425	
131	735	1947-01-07	NaN	NaN	70303	
66	199	1946-10-29	NaN	NaN	60330	
37	45	1944-12-04	NaN	NaN	60320	
32	39	1944-03-	NaN	NaN	10102	

	id	date	tsunamiEventId	earthquakeEventId	volcanoLocationId	volcanoLocat
		27				
65	196	1941-09-21	NaN	NaN	60330	
96	552	1940-06-20	NaN	NaN	60702	
130	733	1938-06-05	NaN	NaN	70303	
121	697	1933-12-25	1666.0	NaN	70301	
142	823	1933-12-24	NaN	NaN	80205	
45	94	1930-12-18	NaN	NaN	60325	
19	26	1930-09-11	4218.0	NaN	10104	
21	28	1928-11-02	NaN	NaN	10106	
80	394	1928-08-04	1607.0	NaN	60415	
34	42	1928-01-23	NaN	NaN	10204	
44	92	1920-07-25	NaN	NaN	60325	
18	25	1919-05-22	1476.0	NaN	10104	
57	138	1919-05-19	NaN	NaN	60328	
12	18	1917-04-01	NaN	NaN	40105	
25	32	1914-09-10	NaN	NaN	40104	
147	867	1914-01-12	1412.0	2994.0	80208	
64	195	1913-06-23	NaN	NaN	60330	
104	585	1913-03-14	1407.0	2973.0	60704	

	id	date	tsunamiEventId	earthquakeEventId	volcanoLocationId	volcanoLocat
27	34	1912-12-03	NaN	NaN	20302	
63	193	1911-11-08	NaN	NaN	60330	
138	761	1911-01-30	1392.0	8391.0	70307	
84	411	1907-09-28	NaN	NaN	60418	
6	11	1907-08-04	NaN	NaN	201112	
7	12	1905-03-10	NaN	NaN	10102	
10	15	1904-02-25	NaN	NaN	30301	
20	27	1903-08-30	NaN	NaN	40105	
56	137	1901-05-22	NaN	NaN	60328	
129	730	1897-05-23	NaN	NaN	70303	
28	35	1895-06-17	NaN	NaN	50103	
62	176	1895-05-22	NaN	NaN	60330	
103	584	1892-06-07	1197.0	NaN	60704	
116	669	1890-06-29	NaN	NaN	60807	
99	569	1889-09-06	1181.0	6309.0	60703	
2	4	1888-03-13	1175.0	NaN	50107	

	id	date	tsunamiEventId	earthquakeEventId	volcanoLocationId	volcanoLocat
128	724	1887-03-09	NaN	NaN	70303	
26	33	1886-08-31	NaN	NaN	40311	
61	167	1885-04-18	NaN	NaN	60330	
55	136	1875-01-29	NaN	NaN	60328	
137	756	1874-07-19	NaN	NaN	70307	
117	677	1873-01-16	NaN	NaN	70106	
43	82	1872-11-03	NaN	NaN	60325	
22	29	1872-04-24	NaN	NaN	10102	
42	81	1872-04-15	NaN	NaN	60325	
127	716	1871-12-08	NaN	NaN	70303	
112	643	1871-08-07	NaN	NaN	60806	
118	679	1871-04-30	1020.0	2179.0	70108	
95	533	1871-03-03	1022.0	NaN	60701	
83	410	1869-07-07	NaN	NaN	60418	
54	135	1864-01-03	NaN	NaN	60328	
115	666	1861-12-29	NaN	NaN	60807	
102	581	1856-03-02	859.0	1995.0	60704	
141	787	1853-10-29	809.0	NaN	80103	

	id	date	tsunamiEventId	earthquakeEventId	volcanoLocationId	volcanoLocat
126	712	1853-07-13	NaN	NaN	70303	
30	37	1853-06-24	NaN	NaN	40311	
53	133	1848-05-16	NaN	NaN	60328	
93	489	1845-02-08	741.0	1865.0	60603	
36	44	1843-11-17	NaN	NaN	10106	
111	629	1840-02-02	718.0	NaN	60806	
110	627	1838-02-26	NaN	NaN	60806	
41	71	1832-12-25	NaN	NaN	60325	
3	6	1832-11-01	NaN	NaN	10106	
52	132	1826-10-11	NaN	NaN	60328	
40	69	1822-12-27	NaN	NaN	60325	
90	482	1820-06-11	NaN	NaN	60509	
74	333	1817-01-24	NaN	NaN	60335	
78	367	1815-04-10	613.0	NaN	60404	
125	707	1814-02-01	NaN	NaN	70303	
101	580	1812-08-06	NaN	NaN	60704	
17	24	1805-08-11	NaN	NaN	10102	
124	706	1800-10-30	NaN	NaN	70303	

	id	date	tsunamiEventId	earthquakeEventId	volcanoLocationId	volcanoLocat
152	932	1792-05-21	568.0	1515.0	80210	
146	859	1781-04-11	529.0	NaN	80208	
145	858	1779-11-08	NaN	NaN	80208	
5	9	1779-08-08	NaN	NaN	10102	
109	618	1773-02-02	NaN	NaN	60806	
108	617	1772-05-09	504.0	NaN	60806	
123	704	1766-07-20	NaN	NaN	70303	
136	754	1754-11-28	2853.0	1328.0	70307	
135	753	1749-08-11	431.0	1304.0	70307	
11	17	1737-05-20	NaN	NaN	10102	
151	916	1717-02-07	NaN	NaN	80209	
150	915	1716-11-09	NaN	NaN	80209	
134	752	1716-09-24	387.0	1205.0	70307	
51	128	1716-07-20	NaN	NaN	60328	
100	578	1711-12-11	NaN	NaN	60704	
89	477	1694-11-20	NaN	NaN	60509	
88	466	1692-06-04	NaN	NaN	60507	

	id	date	tsunamiEventId	earthquakeEventId	volcanoLocationId	volcanoLocat
16	23	1682-08-12	NaN	NaN	10102	
1	1	NaT	2852.0	421.0	10106	
9	14	NaT	NaN	NaN	10106	
29	36	NaT	NaN	NaN	10101	
35	43	NaT	NaN	920.0	201141	
39	64	NaT	NaN	NaN	60325	
87	457	NaT	285.0	NaN	60505	
106	599	NaT	312.0	NaN	60804	
107	607	NaT	222.0	NaN	60806	
114	664	NaT	5754.0	10519.0	60807	
143	844	NaT	61.0	7341.0	80208	
144	846	NaT	NaN	NaN	80208	
149	904	NaT	NaN	NaN	80209	

Part 2: Volcanic Matryoshkas 🪆 (30 POINTS TOTAL)

Now, that most of the data has been tidied up. We will organize the data into more sizable pieces of information in order to extract useful information. **You can use loops in the section if you wish, however your results must be displayed in a viewable manner.**

2.1.1: *(10 points here)*

Use the **groupby function in Pandas** to create separate dataframes for each unique country.

- **Each table must only have the columns**: 'date' 'country', 'name', and 'vei'
- **Sort** the dataframe of **each country** by **highest to lowest 'vei'**
- Use the **display** function to show **each sorted table**

You MUST use the groupby function here and display your results.

In [47]:
```
group_df = new_df[['date', 'country', 'name', 'vei']].sort_values('vei', asc
for country in group_df.groups:
```

```
    display(group_df.get_group(country))
```

	date	country	name	vei
33	1984-08-15	Cameroon	Oku Volcanic Field	NaN

	date	country	name	vei
10	1904-02-25	Comoros	Karthala	2.0

	date	country	name	vei
27	1912-12-03	Congo, DRC	Nyamulagira	3.0
4	1977-01-10	Congo, DRC	Nyiragongo	1.0

	date	country	name	vei
6	1907-08-04	Ethiopia	Alayta	2.0
35	NaT	Ethiopia	Dama Ali	NaN

	date	country	name	vei
34	1928-01-23	Greece	Santorini	2.0

	date	country	name	vei
78	1815-04-10	Indonesia	Tambora	7.0
76	1963-03-18	Indonesia	Agung	5.0
106	NaT	Indonesia	Gamkonora	5.0
57	1919-05-19	Indonesia	Kelud	4.0
59	1966-04-26	Indonesia	Kelud	4.0
105	1966-08-12	Indonesia	Awu	4.0
92	1983-07-23	Indonesia	Colo	4.0
115	1861-12-29	Indonesia	Kie Besi	4.0
43	1872-11-03	Indonesia	Merapi	4.0
114	NaT	Indonesia	Kie Besi	4.0
42	1872-04-15	Indonesia	Merapi	4.0
60	1990-02-10	Indonesia	Kelud	4.0
80	1928-08-04	Indonesia	Paluweh	3.0
44	1920-07-25	Indonesia	Merapi	3.0
40	1822-12-27	Indonesia	Merapi	3.0
45	1930-12-18	Indonesia	Merapi	3.0
41	1832-12-25	Indonesia	Merapi	3.0
52	1826-10-11	Indonesia	Kelud	3.0
63	1911-11-08	Indonesia	Semeru	3.0
111	1840-02-02	Indonesia	Gamalama	3.0
50	1994-11-22	Indonesia	Merapi	3.0
56	1901-05-22	Indonesia	Kelud	3.0
103	1892-06-07	Indonesia	Awu	3.0
102	1856-03-02	Indonesia	Awu	3.0
101	1812-08-06	Indonesia	Awu	3.0
71	1978-09-19	Indonesia	Semeru	3.0
100	1711-12-11	Indonesia	Awu	3.0
39	NaT	Indonesia	Merapi	3.0
97	1974-02-11	Indonesia	Karangetang	3.0
73	1981-03-29	Indonesia	Semeru	3.0
79	1969-01-28	Indonesia	Iya	3.0
69	1967-08-31	Indonesia	Semeru	3.0

	date	country	name	vei
70	1985-05-10	Indonesia	Semeru	3.0
77	1963-05-16	Indonesia	Agung	3.0
47	1961-05-08	Indonesia	Merapi	3.0
108	1772-05-09	Indonesia	Gamalama	3.0
46	1953-03-23	Indonesia	Merapi	3.0
58	1951-08-31	Indonesia	Kelud	3.0
107	NaT	Indonesia	Gamalama	3.0
54	1864-01-03	Indonesia	Kelud	2.0
83	1869-07-07	Indonesia	Lewotobi	2.0
95	1871-03-03	Indonesia	Ruang	2.0
53	1848-05-16	Indonesia	Kelud	2.0
93	1845-02-08	Indonesia	Soputan	2.0
110	1838-02-26	Indonesia	Gamalama	2.0
74	1817-01-24	Indonesia	Ijen	2.0
109	1773-02-02	Indonesia	Gamalama	2.0
51	1716-07-20	Indonesia	Kelud	2.0
112	1871-08-07	Indonesia	Gamalama	2.0
85	1948-04-07	Indonesia	Iliwerung	2.0
94	1958-07-12	Indonesia	Mahawu	2.0
113	1962-12-31	Indonesia	Gamalama	2.0
68	1963-05-05	Indonesia	Semeru	2.0
75	1963-09-05	Indonesia	Batur	2.0
81	1964-01-01	Indonesia	Paluweh	2.0
48	1969-01-07	Indonesia	Merapi	2.0
72	1976-08-31	Indonesia	Semeru	2.0
98	1976-09-15	Indonesia	Karangetang	2.0
82	1981-09-05	Indonesia	Paluweh	2.0
49	1986-10-15	Indonesia	Merapi	2.0
37	1944-12-04	Indonesia	Dieng Volcanic Complex	2.0
66	1946-10-29	Indonesia	Semeru	2.0
65	1941-09-21	Indonesia	Semeru	2.0
61	1885-04-18	Indonesia	Semeru	2.0

	date	country	name	vei
99	1889-09-06	Indonesia	Banua Wuhu	2.0
116	1890-06-29	Indonesia	Kie Besi	2.0
62	1895-05-22	Indonesia	Semeru	2.0
96	1940-06-20	Indonesia	Karangetang	2.0
84	1907-09-28	Indonesia	Lewotobi	2.0
104	1913-03-14	Indonesia	Awu	2.0
64	1913-06-23	Indonesia	Semeru	2.0
38	1979-02-20	Indonesia	Dieng Volcanic Complex	1.0
67	1950-08-28	Indonesia	Semeru	1.0
86	1983-08-17	Indonesia	Iliwerung	1.0
55	1875-01-29	Indonesia	Kelud	0.0

	date	country	name	vei
19	1930-09-11	Italy	Stromboli	3.0
18	1919-05-22	Italy	Stromboli	3.0
22	1872-04-24	Italy	Vesuvius	3.0
9	NaT	Italy	Etna	3.0
16	1682-08-12	Italy	Vesuvius	3.0
29	NaT	Italy	Campi Flegrei	3.0
5	1779-08-08	Italy	Vesuvius	2.0
36	1843-11-17	Italy	Etna	2.0
3	1832-11-01	Italy	Etna	2.0
11	1737-05-20	Italy	Vesuvius	2.0
17	1805-08-11	Italy	Vesuvius	2.0
32	1944-03-27	Italy	Vesuvius	2.0
23	1979-07-05	Italy	Etna	2.0
8	1986-07-24	Italy	Stromboli	2.0
24	1987-04-17	Italy	Etna	2.0
7	1905-03-10	Italy	Vesuvius	2.0
21	1928-11-02	Italy	Etna	1.0
14	1970-03-02	Italy	Campi Flegrei	NaN
1	NaT	Italy	Etna	NaN

	date	country	name	vei
144	NaT	Japan	Aira	5.0
145	1779-11-08	Japan	Aira	4.0
142	1933-12-24	Japan	Kuchinoerabujima	4.0
147	1914-01-12	Japan	Aira	4.0
146	1781-04-11	Japan	Aira	4.0
149	NaT	Japan	Kirishimayama	3.0
150	1716-11-09	Japan	Kirishimayama	3.0
151	1717-02-07	Japan	Kirishimayama	3.0
148	1955-10-13	Japan	Aira	3.0
143	NaT	Japan	Aira	3.0
152	1792-05-21	Japan	Unzendake	2.0
153	1991-06-03	Japan	Unzendake	1.0

	date	country	name	vei
12	1917-04-01	New Zealand	Okataina	1.0
20	1903-08-30	New Zealand	Okataina	1.0
25	1914-09-10	New Zealand	Whakaari/White Island	NaN

	date	country	name	vei
88	1692-06-04	Pacific Ocean	Serua	4.0
87	NaT	Pacific Ocean	Teon	4.0
89	1694-11-20	Pacific Ocean	Banda Api	3.0
91	1988-05-09	Pacific Ocean	Banda Api	3.0
90	1820-06-11	Pacific Ocean	Banda Api	2.0

	date	country	name	vei
31	1954-08-03	Papua New Guinea	Bam	2.0
13	1979-03-08	Papua New Guinea	Karkar	2.0
2	1888-03-13	Papua New Guinea	Ritter Island	2.0
28	1895-06-17	Papua New Guinea	Karkar	2.0

	date	country	name	vei
139	1965-09-28	Philippines	Taal	4.0
134	1716-09-24	Philippines	Taal	4.0
136	1754-11-28	Philippines	Taal	4.0
125	1814-02-01	Philippines	Mayon	4.0
138	1911-01-30	Philippines	Taal	3.0
129	1897-05-23	Philippines	Mayon	3.0
126	1853-07-13	Philippines	Mayon	3.0
128	1887-03-09	Philippines	Mayon	3.0
127	1871-12-08	Philippines	Mayon	3.0
133	1984-09-09	Philippines	Mayon	3.0
132	1968-04-21	Philippines	Mayon	3.0
135	1749-08-11	Philippines	Taal	3.0
123	1766-07-20	Philippines	Mayon	3.0
119	1950-09-15	Philippines	Camiguin	3.0
118	1871-04-30	Philippines	Camiguin	2.0
124	1800-10-30	Philippines	Mayon	2.0
120	1996-08-10	Philippines	Kanlaon	2.0
117	1873-01-16	Philippines	Ragang	2.0
131	1947-01-07	Philippines	Mayon	2.0
140	1969-03-21	Philippines	Didicas	2.0
122	1978-07-29	Philippines	Bulusan	2.0
137	1874-07-19	Philippines	Taal	2.0
130	1938-06-05	Philippines	Mayon	2.0
121	1933-12-25	Philippines	Bulusan	2.0

	date	country	name	vei
15	1972-06-09	Reunion	Fournaise, Piton de la	2.0

	date	country	name	vei
141	1853-10-29	Taiwan	Unnamed	2.0

	date	country	name	vei
26	1886-08-31	Tonga	Niuafo'ou	4.0
30	1853-06-24	Tonga	Niuafo'ou	0.0

2.1.2: *(5 points here)*

Using **groupby again**, **print** out the **maximum 'vei'** for **each unique country.**

You MUST use the groupby function here and print your results.

- **Print** out your results in this format: "Country: {country_name}, Highest VEI: {vei}"

In [48]:
```
group_df = new_df.groupby('country')['vei'].max()
for country_name, vei in group_df.items():
    print(f'Country: {country_name}, Highest VEI: {vei}')
```

```
Country: Cameroon, Highest VEI: nan
Country: Comoros, Highest VEI: 2.0
Country: Congo, DRC, Highest VEI: 3.0
Country: Ethiopia, Highest VEI: 2.0
Country: Greece, Highest VEI: 2.0
Country: Indonesia, Highest VEI: 7.0
Country: Italy, Highest VEI: 3.0
Country: Japan, Highest VEI: 5.0
Country: New Zealand, Highest VEI: 1.0
Country: Pacific Ocean, Highest VEI: 4.0
Country: Papua New Guinea, Highest VEI: 2.0
Country: Philippines, Highest VEI: 4.0
Country: Reunion, Highest VEI: 2.0
Country: Taiwan, Highest VEI: 2.0
Country: Tonga, Highest VEI: 4.0
```

Finally, we have ALMOST REACHED THE END!! Since there is still quite a bit of missing data, we want to make use of what is still available.

A very powerful tool in Python's magnificent collection of libraries is its beautiful graphing tools.

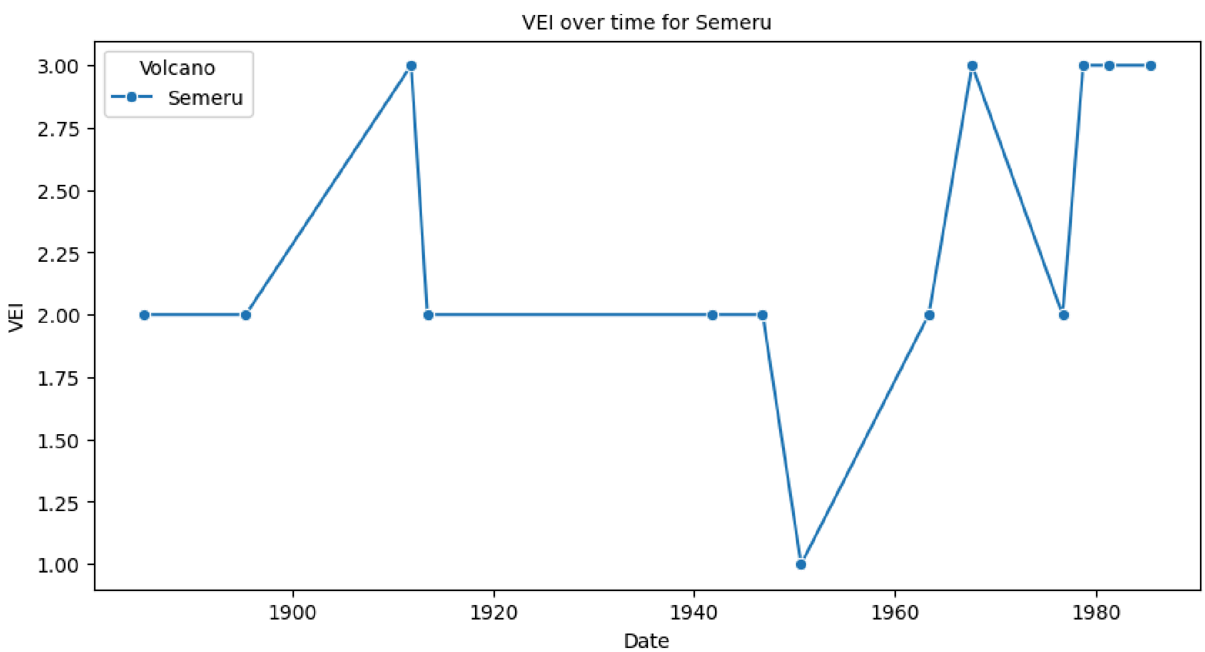
Check out libraries such as Seaborn or Matplotlib to create meaningful visualizations! **Your final task in this section requires the use of these libraries**

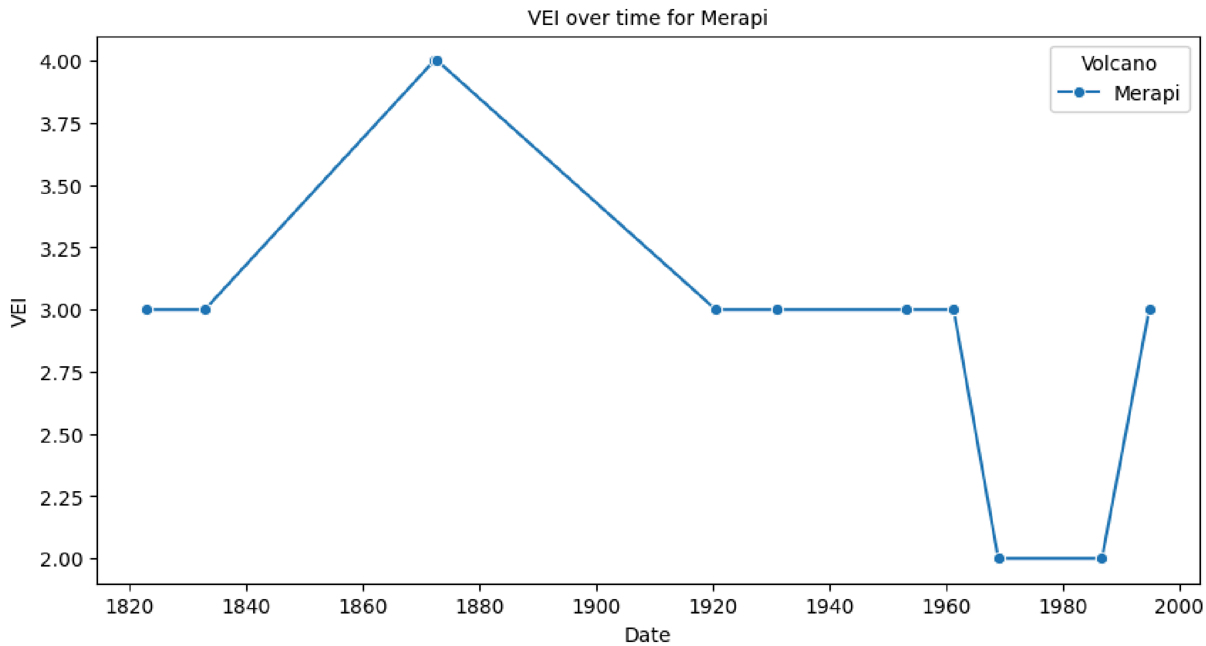
2.1.3: *(15 points here)*

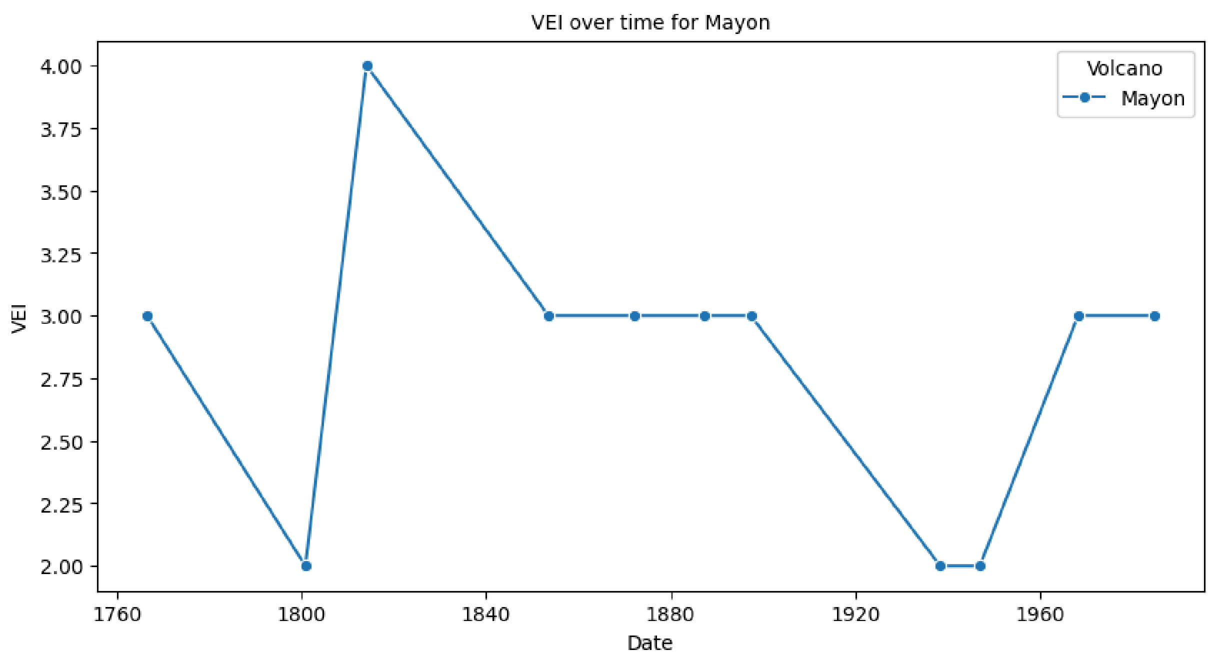
- Based on the **unique names of volcanoes**, **filter names that have more than 3 datapoints under their name.**
- Each datapoint in the dataframe refers to a recorded instance of a volcanic eruption.
- Make **separate line graphs for each volcano** and **plot their VEIs over time.**
- You **must display each graph** to receive credit.

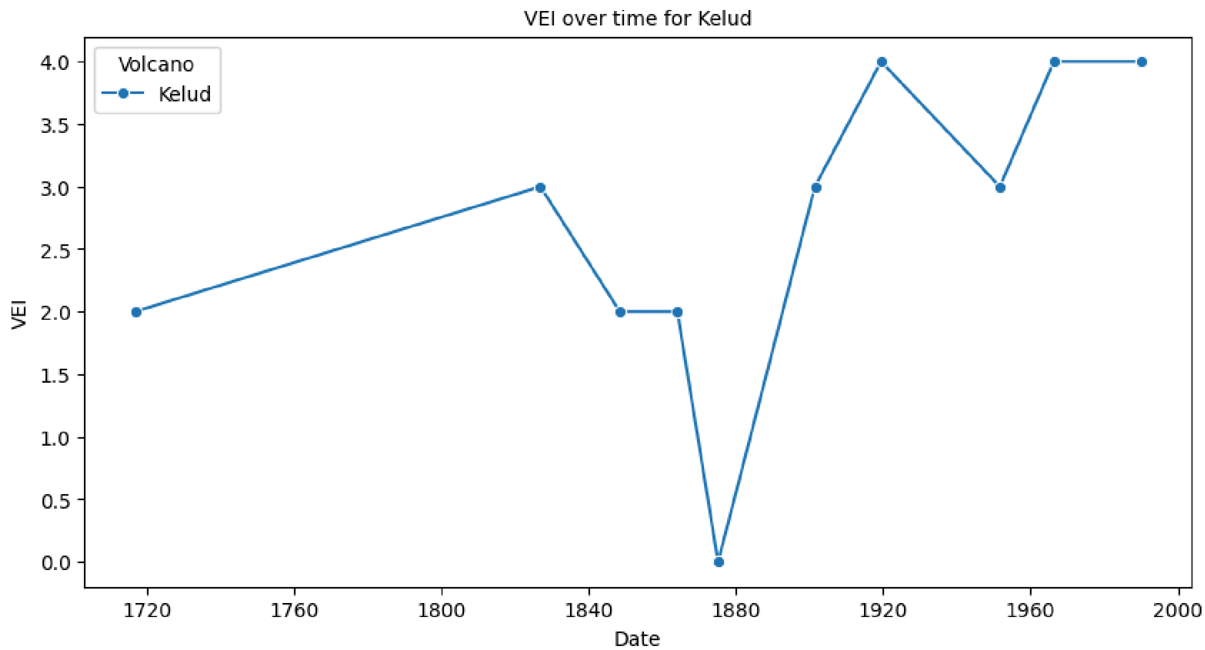
Make sure to properly label all parts of the graph appropriately to receive credit 👀
(like title, axes, legend, etc...)

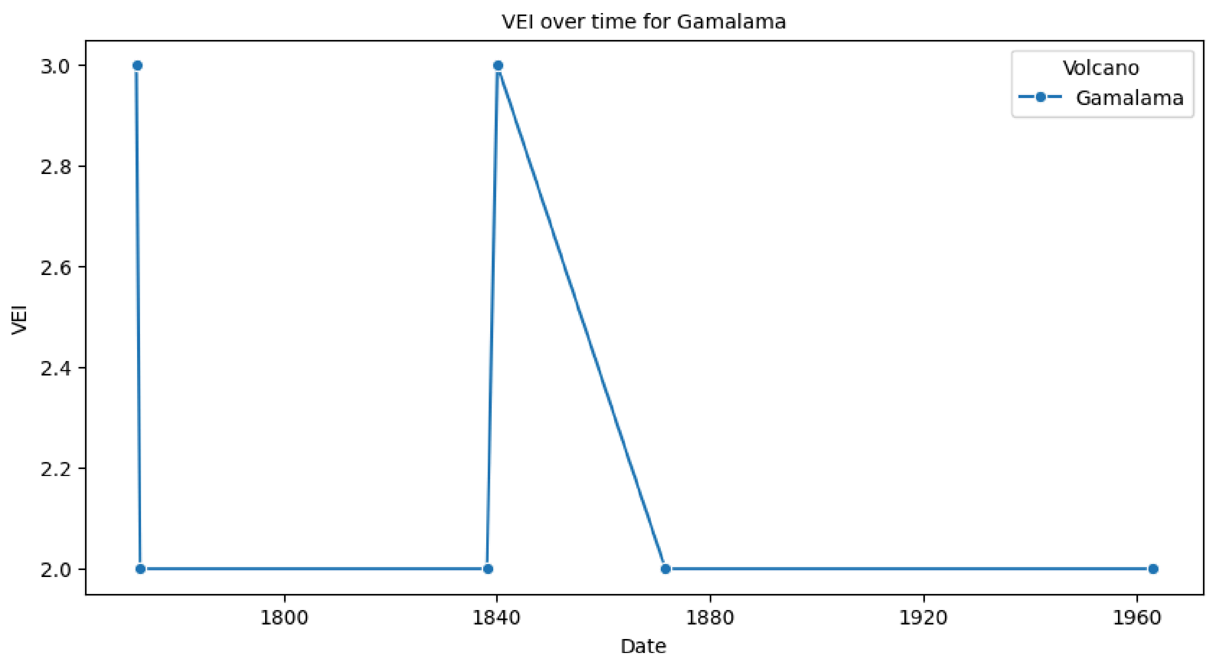
In [49]:
```
import matplotlib.pyplot as plt
import seaborn as sns
volcano_count = new_df['name'].value_counts()
more_than_3 = volcano_count[volcano_count > 3].index
for volcano in more_than_3:
    volcano_data = new_df[new_df['name'] == volcano]
    volcano_sort = volcano_data.sort_values('date')
    plt.figure(figsize=(10, 5))
    sns.lineplot(data=volcano_sort, x='date', y='vei', marker='o', label=vol
    plt.title(f'VEI over time for {volcano}', fontsize=10)
    plt.xlabel('Date', fontsize=10)
    plt.ylabel('VEI', fontsize=10)
    plt.legend(title= 'Volcano', fontsize=10)
    plt.show()
```

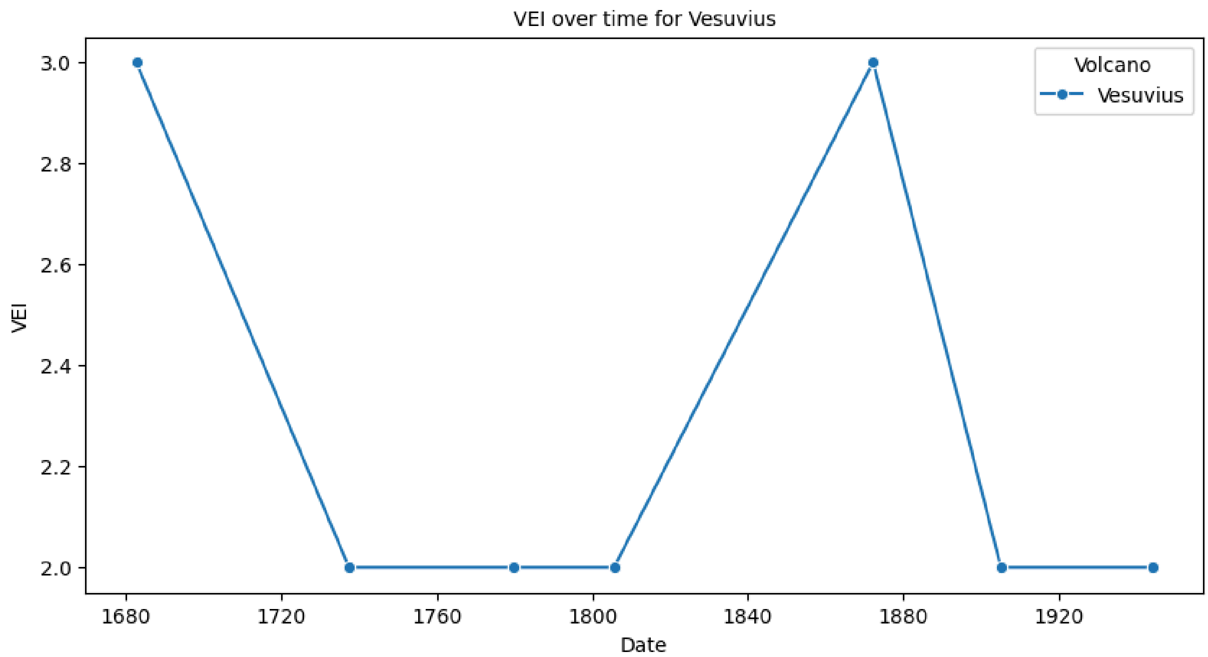


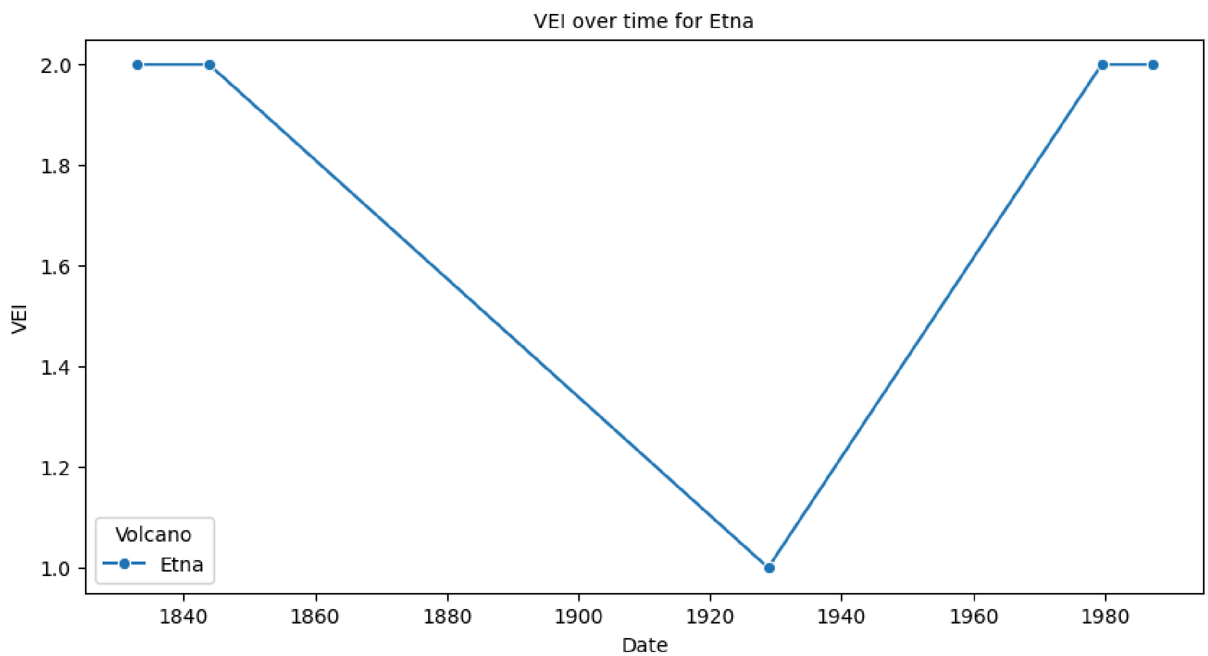

VEI over time for Merapi
Volcano
Merapi
VEI
4.00
3.75
3.50
3.25
3.00
2.75
2.50
2.25
2.00
1820
1840
1860
1880
1900
1920
1940
1960
1980
2000
Date

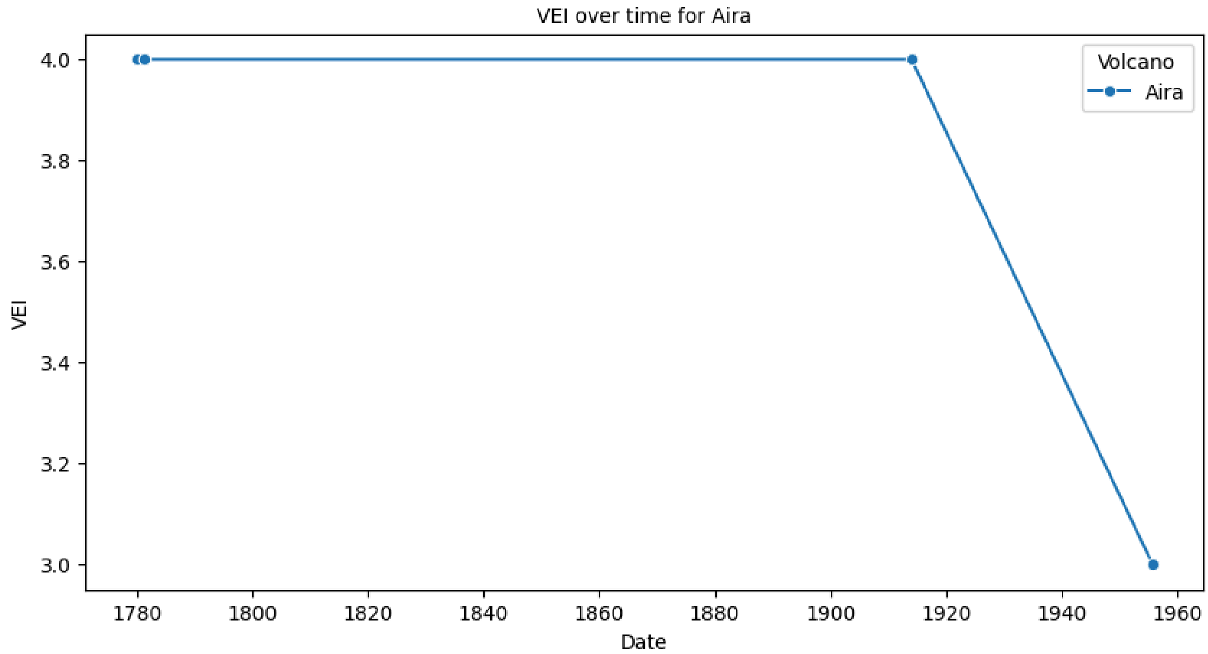

VEI over time for Mayon
Volcano
Mayon
VEI
4.00
3.75
3.50
3.25
3.00
2.75
2.50
2.25
2.00
1760
1800
1840
1880
1920
1960
Date

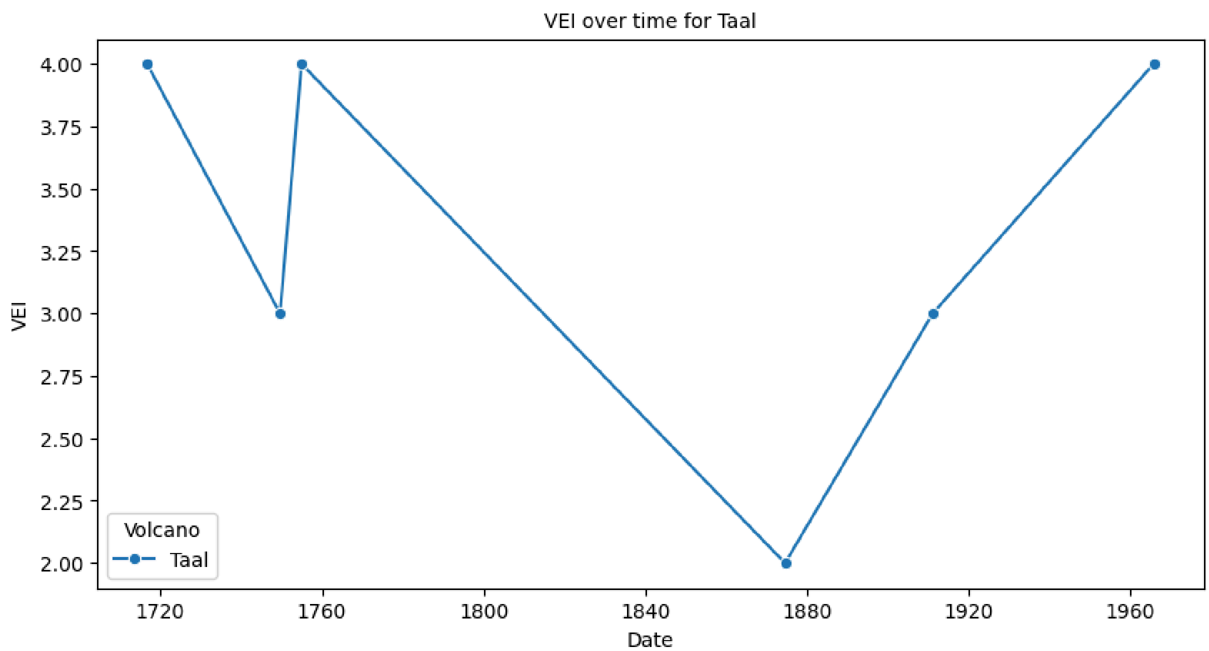

VEI over time for Kelud
Volcano
Kelud
VEI
Date
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1720
1760
1800
1840
1880
1920
1960
2000

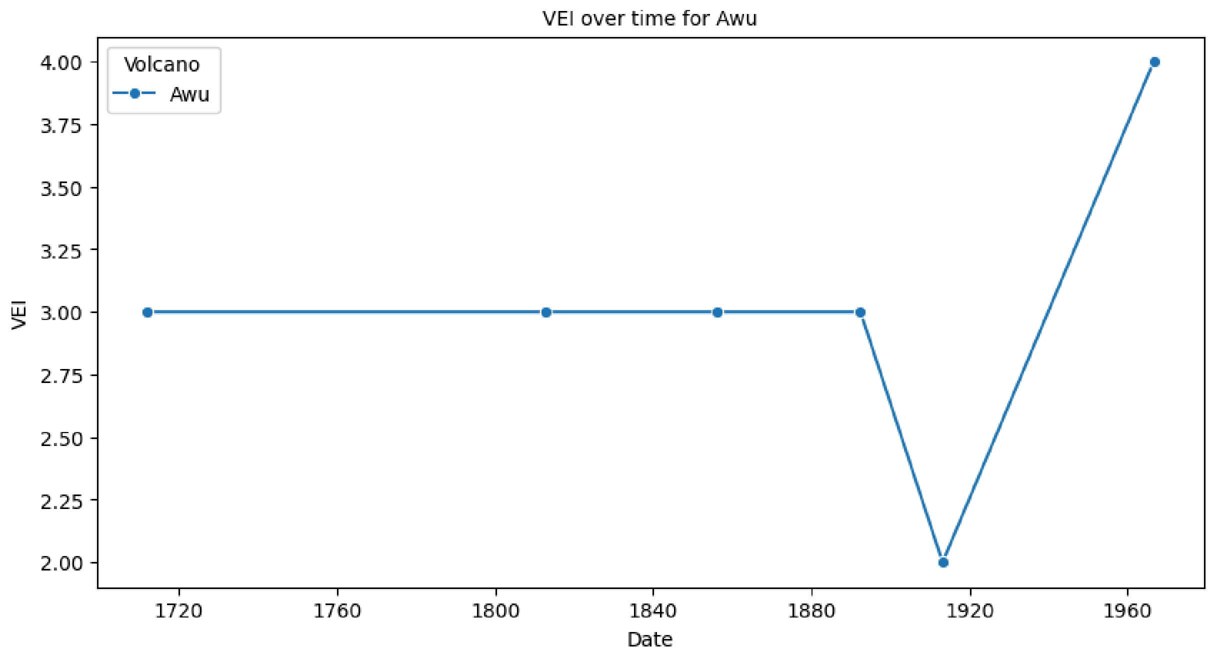

VEI over time for Gamalama
Volcano
Gamalama
VEI
Date
3.0
2.8
2.6
2.4
2.2
2.0
1800
1840
1880
1920
1960


VEI over time for Vesuvius
3.0
2.8
2.6
2.4
2.2
2.0
VEI
1680
1720
1760
1800
1840
1880
1920
Date
Volcano
Vesuvius


VEI over time for Etna
2.0
1.8
1.6
1.4
1.2
1.0
VEI
1840
1860
1880
1900
1920
1940
1960
1980
Date
Volcano
Etna


VEI over time for Aira
VEI
Date
4.0
3.8
3.6
3.4
3.2
3.0
1780
1800
1820
1840
1860
1880
1900
1920
1940
1960
Volcano
Aira


VEI over time for Taal
VEI
Date
4.00
3.75
3.50
3.25
3.00
2.75
2.50
2.25
2.00
1720
1760
1800
1840
1880
1920
1960
Volcano
Taal



Part 3: Fiery Jobs 🚒 (15 POINTS TOTAL)

Proficiency in **SQL** is also super important. SQL databases are essentially relational databases in which there are vast amounts of tabular data. which can often be used to connect with related tablular data. This is a pretty good intro into learning more about SQL.

Check out this tutorial for some clarifications on SQL.

Now! We'll be using **`sqlite`** to access a database.

- Start by downloading the sql lite file and putting it in the same directory as this notebook (hit the 'download' button in the upper right).
- Check out the description of the data so you know the table / column names.

The following code will use `sqlite3` to create a database connection. `sqlite3` is the library in Python that assists in navigating through SQL databases.

Note: If you are working on this assignment via Google Colab, sometimes the runtime resets and it will throw errors.

Instead of running through the entire notebook, run the notebook from the following code block and onwards:

- Click anywhere on the next code block.
- Go up to where it says **'Runtime'** in the toolbar (right under the title of the notebook and **in between 'Insert' and 'Tools'**)
- Hover over it and **click on the option** that says **'Run cell and below'**

In [50]:
```
import sqlite3
import pandas as pd

conn = sqlite3.connect("database.sqlite")
crsr = conn.cursor()
```

In [51]:
```
# This code will let you check out the different tables within the database.
query = "SELECT name FROM sqlite_master WHERE type='table';"
tables = crsr.execute(query).fetchall()
print(tables)
```

```
[('Salaries',), ('SALARIES_2011',), ('SALARIES_2012',), ('SALARIES_2013',),
('SALARIES_2014',)]
```

Remember that each problem should be solved with a single SQL query.

Note: All outputs must be shown

- Only include whatever fields are mentioned throughout each question, nothing more and nothing less.
- Follow each instruction clearly

3.1.1: 2 Points

From the **Salaries** table, get the **average base pay** for firefighters (all job titles consisting of the word "firefighter" **(not case-sensitive)**) between the **years 2012 to 2014**.

Remember that firefighters that also occupy other professions are still considered firefighters.

Hint: Look into this 👀

In [52]:
```
query = "SELECT AVG(BASEPAY) AS AVERAGE_BASE_PAY FROM SALARIES WHERE LOWER(J

# KEEP THIS. It will display the whole dataframe.
df = pd.read_sql(query, conn)
df
```

Out[52]:

	AVERAGE_BASE_PAY
0	101657.66285

3.1.2: 2 Points

From the **Salaries** table, get all the firefighters (all job titles consisting of the word "firefighter" **(not case-sensitive)**) in the **year 2012** making under **$90,000 as a base pay. Sort** them in **descending** order by their pay.

Remember that firefighters that also occupy other professions are still considered firefighters.

In [53]:
```
query = "SELECT * FROM SALARIES WHERE LOWER(JOBTITLE) LIKE '%firefighter%' A

# KEEP THIS. It will display the whole dataframe.
df = pd.read_sql(query, conn)
df
```

Out[53]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherP
0	40505	Vincent Pampanin	Firefighter	89014.12	43348.93	14071.
1	39771	Brook Mancinelli	EMT/Paramedic/Firefighter	86469.94	61993.52	6555.
2	46472	Adam Lewis	EMT/Paramedic/Firefighter	85436.51	4646.71	13845.
3	45756	Peter Johnson	Firefighter	84466.16	8504.68	13114.
4	41897	Michael Craig	Firefighter	84466.15	30232.18	15369.
5	45333	Matthew Estrada	Firefighter	84466.15	16600.28	8493.
6	45250	Alberto Jaime Lopez	Firefighter	84386.06	11170.30	13865
7	45373	Nick Oxford	Firefighter	84305.99	8898.97	15379
8	47645	Daniel Sankey	Firefighter	84305.99	3583.40	8993.
9	44841	Catherine Abrams	Firefighter	84305.98	15591.26	12152.
10	44059	Ketric Mahoney	Firefighter	84305.97	20272.22	12420.
11	46639	Edwin Marsullo Jr	Firefighter	84296.64	4386.42	11940.
12	43165	Marilyn Barton	Firefighter	84145.82	31922.76	8043.
13	46333	Jarrod Cariola	Firefighter	84136.47	12157.42	7161.
14	45400	Truc Nguyen	Firefighter	84065.74	10935.90	13482.
15	50377	Edgardo Vergara	EMT/Paramedic/Firefighter	83550.17	1786.89	3472.
16	41887	Patrick Reyes	Firefighter	83546.57	34953.92	13604.
17	44577	Travis Hemenez	Firefighter	83546.57	17897.68	12900.
18	44650	Gregory Ginotti	Firefighter	83546.57	22417.40	8597.
19	46751	Dominic Fasso	Firefighter	83546.57	3039.23	13762.

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherP
20	47339	Jimmy Cheung	Firefighter	83546.57	8740.48	6120.
21	47957	Jesse Maurer	Firefighter	83546.57	3557.05	8534.
22	46396	Steven Rodriguez	Firefighter	83546.56	5445.18	13495.
23	47276	Kenichi Noguchi	Firefighter	83495.80	3101.46	11501.
24	46110	Joseph Kilgore	Firefighter	83466.48	9874.84	10924.
25	48374	Shaun Mooney	Firefighter	83466.48	7242.16	3992.
26	44615	Gail Readdie	Firefighter	83386.40	19715.60	11353.
27	43474	Damon Robertson	Firefighter	83226.22	24243.02	13830.
28	46981	Jared Cooper	EMT/Paramedic/Firefighter	83216.55	6300.29	12909.
29	50690	Jennifer Balestrieri	EMT/Paramedic/Firefighter	83216.55	0.00	3701.
30	49279	Jonathan Honda	EMT/Paramedic/Firefighter	83216.54	4104.15	5520.
31	47387	James Novello Jr	EMT/Paramedic/Firefighter	83216.53	7525.57	9709.
32	47585	Christopher Chambre	EMT/Paramedic/Firefighter	83216.52	13333.40	4312.
33	48693	Tan Nguyen	EMT/Paramedic/Firefighter	83216.52	3016.11	8670.
34	50253	William Wong	EMT/Paramedic/Firefighter	83216.52	5208.31	1181.
35	48179	Joseph Del Grande	EMT/Paramedic/Firefighter	83216.50	1848.37	11709.
36	48962	Ryan Jamison	EMT/Paramedic/Firefighter	83215.75	4963.92	5845.
37	49802	Brian Buna	EMT/Paramedic/Firefighter	83212.17	3660.19	4366.
38	48973	Peter Ong	EMT/Paramedic/Firefighter	83199.44	5509.99	5371.
39	49503	Christopher Sandoval	EMT/Paramedic/Firefighter	83167.28	3766.58	5156.

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherP
40	47343	Ryan Watson	EMT/Paramedic/Firefighter	83090.36	6077.03	11521.
41	49117	Shane Pinaula	EMT/Paramedic/Firefighter	83074.28	1322.37	9090.
42	46762	Tsz Lap Ko	EMT/Paramedic/Firefighter	82742.37	16122.65	5629
43	50580	John Kavanaugh	EMT/Paramedic/Firefighter	82725.16	225.73	4578.
44	48200	Christina Clark	EMT/Paramedic/Firefighter	82692.73	7886.46	7048.
45	46113	James McGuigan	Firefighter	82374.64	9939.15	12180.
46	48853	Ryan Towner	EMT/Paramedic/Firefighter	82213.14	2873.09	9436.
47	49232	Clarence Hom	Firefighter	81960.80	2567.50	2462.
48	48789	Arthur Julaton	EMT/Paramedic/Firefighter	81471.39	4807.63	9183
49	48619	Michael Horta	EMT/Paramedic/Firefighter	80682.71	6675.23	8844.
50	46116	Michelle Estrada	EMT/Paramedic/Firefighter	80468.80	10622.02	15990
51	50547	Alec Kauf	EMT/Paramedic/Firefighter	80329.79	5117.17	3049
52	49280	Angelo Manalo	Firefighter	79215.67	0.00	10755.
53	51199	Jason Landivar	EMT/Paramedic/Firefighter	79210.54	779.77	5980.
54	49897	Prisco Somontan	EMT/Paramedic/Firefighter	78944.31	6559.10	6479.
55	50793	Tyrone Harper	EMT/Paramedic/Firefighter	78505.61	4805.80	4877.
56	47000	Anna Rensi	Firefighter	78411.95	8495.29	13359.
57	49827	Scott Ward	EMT/Paramedic/Firefighter	78273.10	7309.28	6774.
58	46000	Robi Tse	Firefighter	77696.84	17700.81	10848
59	51420	Joshua Smith	EMT/Paramedic/Firefighter	76119.01	960.07	11349.

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherP
60	54603	Jennifer Glickman	Firefighter	75156.36	0.00	8246.
61	48564	Thomas Ro	EMT/Paramedic/Firefighter	74118.62	15060.35	7864.
62	50337	Matthew Faris	EMT/Paramedic/Firefighter	73980.07	4540.24	11809.
63	50300	Clark Irey	EMT/Paramedic/Firefighter	73839.08	5342.11	10328
64	50513	Brian Machado	EMT/Paramedic/Firefighter	73839.04	4547.27	9857
65	50708	Sylvia Rivera	EMT/Paramedic/Firefighter	73839.02	2079.98	11431.
66	51013	Jane Kang	EMT/Paramedic/Firefighter	73515.91	8202.09	5194.
67	51325	Che Soriano	EMT/Paramedic/Firefighter	73469.46	1739.61	9746.
68	47330	Destin Rey Tianero	EMT/Paramedic/Firefighter	73407.76	20589.74	8954.
69	53880	Michael Fields	EMT/Paramedic/Firefighter	73157.83	37.96	3652.
70	52980	Zachary Beatty	EMT/Paramedic/Firefighter	71075.31	1981.89	6990
71	54228	Philip Telesforo	Firefighter	70921.30	2663.41	5633.
72	52071	Henry Truong	EMT/Paramedic/Firefighter	69923.40	3414.32	11380
73	46710	Michael Menefee	EMT/Paramedic/Firefighter	67678.01	27545.56	16786.
74	57052	Denise Elarms	Firefighter	67432.26	0.00	2533.
75	54073	Kenneth Lincoln	EMT/Paramedic/Firefighter	64082.80	8527.99	6619
76	57311	Sarah Bartel	EMT/Paramedic/Firefighter	63983.09	1800.98	2797
77	55568	Travis Rail	Firefighter	63277.35	6285.31	6397.
78	53658	Guillermo Casillas	Firefighter	62221.42	3727.71	18008.
79	45630	John Diluzio	Firefighter	61028.00	0.00	68357.

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherP
80	43108	Peter Walker	Firefighter	60688.63	96.52	74800
81	49851	Frances Focha	Firefighter	60688.61	10620.88	27978.
82	59077	Ira Burroughs	EMT/Paramedic/Firefighter	59826.84	1378.08	3129.
83	57196	Lesley Sudduth	EMT/Paramedic/Firefighter	59569.51	2875.28	8759.
84	51164	Tracy Cavaretta	EMT/Paramedic/Firefighter	59188.86	0.00	42929.
85	58956	Gregg Takeuchi	Firefighter	58978.66	1127.03	5595.
86	60230	Allen Posey	Firefighter	58861.88	0.00	3787.
87	60447	James Lockhart	EMT/Paramedic/Firefighter	58842.00	0.00	3801.
88	57472	Robert Craig Gordon	EMT/Paramedic/Firefighter	58614.74	1043.93	7239.
89	44540	Randall Henderson	Firefighter	58564.00	24847.86	43458.
90	60519	Dominic Fuentes	Firefighter	58345.60	0.00	2821.
91	60715	Michael Ferry	Firefighter	54718.29	109.67	5397.
92	61379	Joaquinn Villarreal	EMT/Paramedic/Firefighter	53588.98	0.00	4722
93	59718	Edward Bird	EMT/Paramedic/Firefighter	50833.09	5033.60	2971.
94	59149	Gregory Wong	EMT/Paramedic/Firefighter	50822.26	4784.05	8937
95	62419	Clyde Watarai	Firefighter	50186.00	0.00	5186.
96	63097	Rita Kearns	EMT/Paramedic/Firefighter	44869.12	0.00	740.
97	62721	Daniel Nakagawa	EMT/Paramedic/Firefighter	43876.22	584.04	3521
98	53930	Christian Collier	Firefighter	42331.63	28513.76	18607.
99	63600	Christopher Olsen	EMT/Paramedic/Firefighter	37801.15	1559.42	2745.

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherP
100	48652	Eugene Eden Jr	Firefighter	35047.04	3796.52	75866.
101	63922	Ronald Perez	Firefighter	32809.20	0.00	6460.
102	63548	Heather McCarthy	Firefighter	31377.50	4262.88	6605.
103	64290	Richard Platt	EMT/Paramedic/Firefighter	29730.26	1005.22	2383
104	64128	Douglas Mei	EMT/Paramedic/Firefighter	29579.00	3376.47	1601.
105	64138	Dennis Luong	EMT/Paramedic/Firefighter	29579.00	4023.27	1214.
106	64154	Alexander Lamond	EMT/Paramedic/Firefighter	29579.00	1544.82	2960.
107	64165	Paul Hobbs	EMT/Paramedic/Firefighter	29579.00	2765.59	1650
108	64274	Miles DeGraffenreid	EMT/Paramedic/Firefighter	29579.00	212.70	3041.
109	64297	Michael Mason	EMT/Paramedic/Firefighter	29579.00	1298.90	2171.
110	64320	Lucas Muncal	EMT/Paramedic/Firefighter	29579.00	1408.23	1677.
111	64323	Jay Weber	EMT/Paramedic/Firefighter	29579.00	50.42	3015.
112	64331	Graham Hoffman	EMT/Paramedic/Firefighter	29579.00	1002.18	2279.
113	64355	Daniel Nazzareta	EMT/Paramedic/Firefighter	29579.00	783.14	2053.
114	64362	Barry Beere	EMT/Paramedic/Firefighter	29579.00	297.78	2343.
115	64368	Kurt Knuepfel	EMT/Paramedic/Firefighter	29579.00	355.73	2169.
116	64401	Sherry Mahoney	EMT/Paramedic/Firefighter	29579.00	0.00	2402.
117	64405	Benjamin Lopez	EMT/Paramedic/Firefighter	29579.00	56.78	2103.
118	64407	Michael Clements	EMT/Paramedic/Firefighter	29579.00	0.00	2167.
119	64411	Seaborn Chiles	EMT/Paramedic/Firefighter	29579.00	551.46	1626.

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherP
120	64423	Brandon Yukich	EMT/Paramedic/Firefighter	29579.00	93.68	2122.
121	64460	Michael Pendergast	EMT/Paramedic/Firefighter	29579.00	721.61	1305.
122	64738	Sean Wehrman	EMT/Paramedic/Firefighter	29579.00	298.12	3266.
123	64264	Anthony Dawson	EMT/Paramedic/Firefighter	29310.10	0.00	3155
124	64061	Nathan Moore	EMT/Paramedic/Firefighter	29175.65	4065.00	2371.
125	64262	Emily Anderson	EMT/Paramedic/Firefighter	29175.65	2139.69	2169.
126	64310	Phillip Romero	EMT/Paramedic/Firefighter	29175.65	1725.26	2288.
127	64395	Christina Altenberg	EMT/Paramedic/Firefighter	29175.65	1645.44	1554
128	64437	Daniel Goepel	EMT/Paramedic/Firefighter	29175.65	1171.45	1558
129	64477	Mark Murphy	EMT/Paramedic/Firefighter	29175.65	1121.82	1188.
130	64489	Maneka Michelle Spidle	EMT/Paramedic/Firefighter	29175.65	767.30	1287
131	64648	Nicole Cabaud	Firefighter	29057.60	0.00	180
132	64252	Angela Jovel	EMT/Paramedic/Firefighter	28822.72	3508.31	1282
133	63745	Eric Marshall	EMT/Paramedic/Firefighter	28503.40	9212.97	2227.
134	63751	Steven Rascon	Firefighter	27146.50	0.00	15180.
135	65438	Jonathan Pettey	Firefighter	25095.87	0.00	2534.
136	65168	Xavier Brown	Firefighter	24656.00	0.00	0.
137	65178	Michael Crehan	Firefighter	24656.00	0.00	0.
138	65202	John Ayers	Firefighter	24656.00	0.00	0.
139	65223	Stephen Marchisio	Firefighter	24656.00	0.00	0.

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherP
140	65226	Jamie Serchia	Firefighter	24656.00	0.00	0.
141	65227	Travis Scott	Firefighter	24656.00	0.00	0.
142	65228	Grayson Ward	Firefighter	24656.00	0.00	0.
143	65231	Andrew Yee	Firefighter	24656.00	0.00	0.
144	65232	Matthew Barr	Firefighter	24656.00	0.00	23
145	65235	Scott Mason	Firefighter	24656.00	0.00	0.
146	65239	Philip Korn	Firefighter	24656.00	0.00	0.
147	65245	Jennifer Romanini	Firefighter	24656.00	28.89	0.
148	65247	Don Noble	Firefighter	24656.00	0.00	0.
149	65248	Tanna Hall	Firefighter	24656.00	0.00	0.
150	65249	Dustin Stewart	Firefighter	24656.00	0.00	0.
151	65250	Frederick Calonico III	Firefighter	24656.00	0.00	0.
152	65251	Johnny Hong	Firefighter	24656.00	0.00	0.
153	65252	Brian Sullivan	Firefighter	24656.00	0.00	0.
154	65253	Patrick Ryan	Firefighter	24656.00	0.00	0.
155	65254	Nicholas Sabella	Firefighter	24656.00	0.00	0.
156	65255	Colin Carter	Firefighter	24656.00	0.00	0.
157	65258	Scott Bryant	Firefighter	24656.00	0.00	0.
158	65260	Simon Lewis	Firefighter	24656.00	0.00	0.
159	65261	Joseph Egan	Firefighter	24656.00	0.00	0.

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherP
160	65264	Daniel Murphy	Firefighter	24656.00	0.00	0.
161	65265	Brandon Murray	Firefighter	24656.00	0.00	0.
162	65266	John Vagenas	Firefighter	24656.00	0.00	0.
163	65267	James James	Firefighter	24656.00	0.00	0.
164	65269	Jonathan Truppa	Firefighter	24656.00	0.00	0.
165	65274	Hashim Anderson	Firefighter	24656.00	0.00	0.
166	65275	Jennifer Risse	Firefighter	24656.00	0.00	0.
167	65571	David Filkins	Firefighter	24656.00	0.00	0.
168	65580	Dwight Nackord	Firefighter	24656.00	0.00	0.
169	65192	Emily O'Rourke	Firefighter	24578.95	0.00	0.
170	65283	Christopher Campbell	Firefighter	24559.69	0.00	0.
171	64602	Russell Zimmerman	EMT/Paramedic/Firefighter	24109.01	545.99	10603
172	65319	Robert Schwartz	Firefighter	20884.00	1215.22	3039.
173	66153	Michael Estrada	EMT/Paramedic/Firefighter	20771.45	0.00	561.
174	62094	Theodore Tom	Firefighter	18738.39	12600.07	31398.
175	66069	Lorena Gutierrez	Firefighter	18492.00	0.00	0.
176	66499	Kathleen Zepeda	Firefighter	14283.88	0.00	3408
177	67327	Daniel Alderete	Firefighter	12328.00	0.00	0.
178	67508	Frank Cuffe	Firefighter	10953.67	0.00	1646.
179	64137	Dean Lundie	Firefighter	10390.50	4897.77	27644.

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherP
180	68270	Victoria Bowen	Firefighter	9246.00	0.00	0.
181	68561	Karlene Allen	Firefighter	8629.60	0.00	0.
182	65745	Michael Sanders	Firefighter	6192.19	1401.83	18482.
183	70751	Daren Brannan	EMT/Paramedic/Firefighter	2737.52	0.00	342
184	65156	William Cody	Firefighter	0.00	0.00	35282
185	66555	Rock Crawford	Firefighter	0.00	0.00	21283.
186	70099	Keith Wong	Firefighter	0.00	0.00	6119
187	70275	Sarah Hamilton	Firefighter	0.00	0.00	5477.
188	72201	James Schrick	Firefighter	0.00	27.46	936

3.1.3: 4 Points

From the **Salaries** table, first get the **averages** of **base pay**, **benefits**, and **overtime pay** for firefighters (all job titles consisting of the word "firefighter" **(not case-sensitive)**).

- Then, make a **column with the sum** of these **three averages**
- Finally, **exclude** job titles containing "FIREFIGHTER" **(case-sensitive)**

Remember that firefighters that also occupy other professions are still considered firefighters.

In [54]:
```
query = "SELECT AVG(BASEPAY) AS AVERAGE_BASE_PAY, AVG(BENEFITS) AS AVERAGE_E

# KEEP THIS. It will display the whole dataframe.
df = pd.read_sql(query, conn)
df
```

Out[54]:

	AVERAGE_BASE_PAY	AVERAGE_BENEFITS	OVERTIME_PAY	TOTAL_AVERAGE_PAY
0	101339.991492	34712.580523	21146.158965	157198.73098

3.1.4: 7 Points

Finally, we'll make our own table in our database.

- Separate the **Salaries table** by **years**, and add it back to the database.
- Using a loop might be helpful.
- You may use basic python to complete the task. However, using querying on SQL is **mandatory**.
- Feel free to **use multiple lines of code for this problem only.**

Hint: Check this out

In [55]:
```
crsr.execute("SELECT DISTINCT YEAR FROM SALARIES;")
years = [year[0] for year in crsr.fetchall()]
for year in years:
    create_table = f"CREATE TABLE IF NOT EXISTS SALARIES_{year} AS SELECT *
    crsr.execute(create_table)
```

In [56]:
```
# Run this code to check if you successfully added your table.
cursor = conn.cursor()
cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
print(cursor.fetchall())
```

```
[('Salaries',), ('SALARIES_2011',), ('SALARIES_2012',), ('SALARIES_2013',),
('SALARIES_2014',)]
```

Part 4: BONUS SECTION (Pandas 'Group By')

This flowchart is taken from our lecture class presentation and illustrates the process of transforming data using the Pandas GroupBy operation. First, the data is input, followed by applying the GroupBy function to one or more columns of the DataFrame. Once the data is grouped, an aggregation function (such as sum(), mean(), or count()) is applied to compute summary statistics for each group

bonus

Your task is to translate the workflow shown in the flowchart into Pandas queries that perform these operations step by step.

Notes: Your task is to translate the workflow shown in the flowchart into Pandas queries. Ensure that the **exact input and exact output** from the flowchart are replicated using Pandas queries, step by step.

4.1 (Point 1)

Create a **sample dataset** that includes columns for ***account, order,*** and ***ext price***.

In [57]:
```
# Create the Sample dataset from above flowchart
input_data = {
    'account' : [383080, 383080, 383080, 412290, 412290, 412290, 412290, 412
    'order' : [10001, 10001, 10001, 10005, 10005, 10005, 10005, 10005, 10006
    'ext price' : [235.83, 232.32, 107.97, 2679.36, 286.02, 832.95, 3472.04,
}

# Create DataFrame
df = pd.DataFrame(input_data)
# Display Dataframe (DONT REMOVE THE CODE)
df.style.format({'ext price' : '{:.2f}'}).hide(axis='index')
```

Out[57]:

account	order	ext price
383080	10001	235.83
383080	10001	232.32
383080	10001	107.97
412290	10005	2679.36
412290	10005	286.02
412290	10005	832.95
412290	10005	3472.04
412290	10005	915.12
218895	10006	3061.12
218895	10006	518.65
218895	10006	216.90
218895	10006	-72.18

4.2 (Point 1+1 =2)

Group by **order** and **show** the **intermediate results**

In [58]:
```
# Group by 'order' and show intermediate result

group = df.groupby('order')
```

```
group

# Display intermediate result for each group; hints: you have to use 'for lo
print("\nIntermediate Grouped Data (Before Aggregation):")

for grouping, order in group:
    display(order[['order', 'ext price']].style.format({'ext price' : '{:.2f
    print()
```

```
Intermediate Grouped Data (Before Aggregation):
```

order	ext price
10001	235.83
10001	232.32
10001	107.97

order	ext price
10005	2679.36
10005	286.02
10005	832.95
10005	3472.04
10005	915.12

order	ext price
10006	3061.12
10006	518.65
10006	216.90
10006	-72.18

4.3 (Point 1) Apply the **Sum Aggregation** for **Each** Group

Now we'll apply the sum aggregation to get the ***total ext price*** for ***each order***:

In [59]:
```
# Repeat group by 'order' again and then apply aggregation (sum of 'ext_pric

aggregated_result = df.groupby('order')['ext price'].sum().reset_index()
# Show the aggregated result after re-grouping  (DONT REMOVE THE CODE)
print("\nAggregated Data (Sum of 'ext_price' per 'order'):")
for index, row in aggregated_result.iterrows():
    temp = pd.DataFrame([row])
    display(temp[['order', 'ext price']].style.format({'order' : '{:.0f}', '
    print()
```

```
Aggregated Data (Sum of 'ext_price' per 'order'):
```

order	ext price
10001	576.12

order	ext price
10005	8185.49

order	ext price
10006	3724.49

4.4 (Point 1) Combine the Results into One Final Table.

Finally, we will **reset the index** and create a combined table that shows order and the sum of the ext price for each group:

Notes: In pandas, `reset_index()` is a method used to reset the index of a DataFrame to its default integer-based index. By default, when you perform certain operations like `groupby()`, the resulting DataFrame may have a new index (e.g., the grouped column). The `reset_index()` method allows you to convert the current index back to a default sequential integer index and optionally, move the current index values into a regular column.

In [60]:
```python
# Reset index to combine result into a single DataFrame
final_result = aggregated_result.reset_index()

# Rename the columns for clarity
final_result = final_result.drop(columns=['index'])
final_result.columns = ['order', 'Order_Total']

# Show the final result  (DONT REMOVE THE CODE)
print("\nFinal Combined Result (Order and Total 'ext_price'):")
temp = pd.DataFrame(final_result)
display(temp.style.format({'Order_Total' : '{:.2f}'}).hide(axis='index'))
```

```
Final Combined Result (Order and Total 'ext_price'):
```

order	Order_Total
10001	576.12
10005	8185.49
10006	3724.49

THE END!