EarthQuake

By Dor Mizrahi & Tal Lilo

Will there be a Tsunami in the next earthquake?

01/2023

Data Science Project

EarthQuakes a little BackGround

From Wikipedia:

"An earthquake is the ground shaking caused by a sudden slip on a fault. Stresses in the earth's outer layer push the sides of the fault together. Stress builds up and the rocks slip suddenly, releasing energy in waves that travel through the earth's crust and cause the shaking that we feel during an earthquake".

We chose the topic of earthquakes, because we thought it is a very interesting and intriguing topic, we wanted a challenging topic that we could explore and research .

The Connection Between EarthQuake to Tsunami

From Wikipedia:

"When a great earthquake ruptures, the faulting can cause vertical slip that is large enough to disturb the overlying ocean, thus generating a tsunami that will travel outwards in all directions".

The Connection Between EarthQuake to Tsunami

as we learn about earthquakes, and research it we found it's impact on a tsunami
as a result we wanted to focus on it.
our research question is:
is it possible to predict a tsunami caused by an earthquake?
in addition we got some more questions like:
How are earthquakes that cause tsunamis characterized?
Can a tsunami happen anywhere in the world?
Are there places where the tsunami is less common?
We will try to answer these questions during the project.

Our Project Steps

Obtating Data
Web Crawling for colletion and obtating the data

Scrubbing Data
Cleaning, formatting and filtering the data

Exploring Data
Visualizing & Understanding the data

Modeling Data
Clustering the data into groups, modeling & algorithm

Interpreting Data
presentation of data understanding & delivering the results + final predict model

WEB CRAWLING

we found a goverment site which contains all the earthquakes of the world, we looked for the earthquakes of the last years - 1900 and above.

Only List Earthquakes Shown on Map

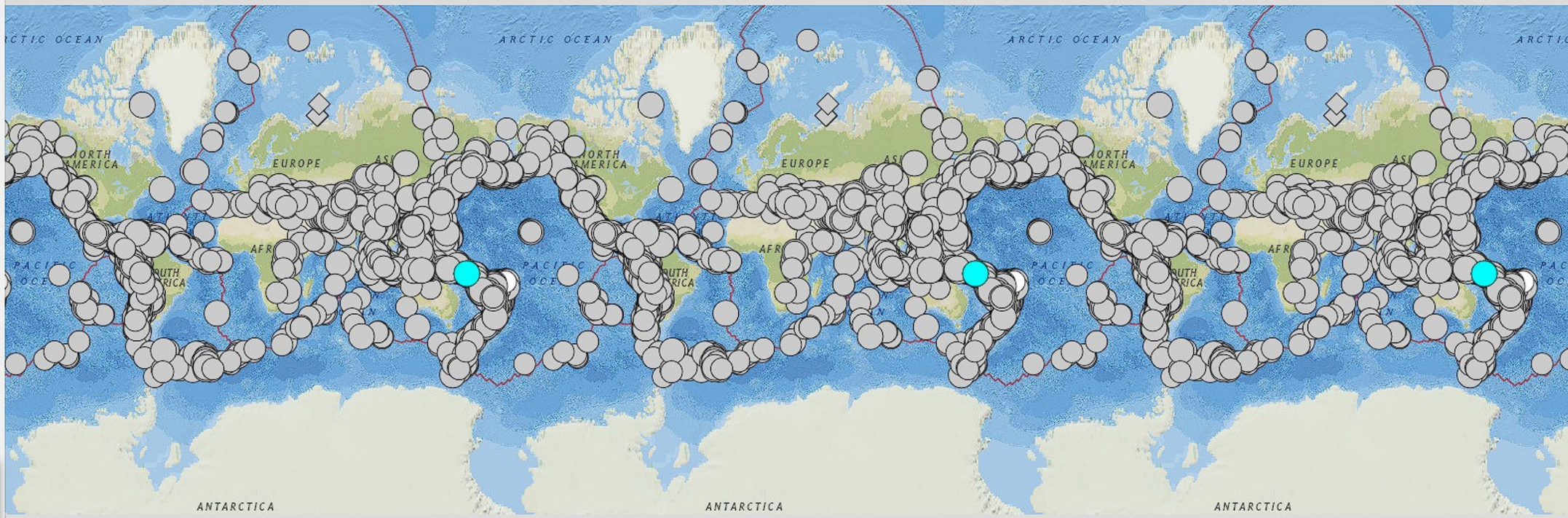
Format: Magnitude | Sort: Newest First

6.8	Samoa Islands region 2022-12-04 19:24:15 (UTC)	38.0 km
7.0	Solomon Islands 2022-11-22 02:03:06 (UTC)	14.0 km
6.9	204 km SW of Bengkulu, Ind... 2022-11-18 13:37:08 (UTC)	25.0 km
7.0	Fiji region 2022-11-12 07:09:13 (UTC)	579.0 km
7.3	205 km ESE of Neiafu, Tonga 2022-11-11 10:48:46 (UTC)	37.0 km
6.6	south of the Fiji Islands 2022-11-09 10:14:33 (UTC)	624.5 km

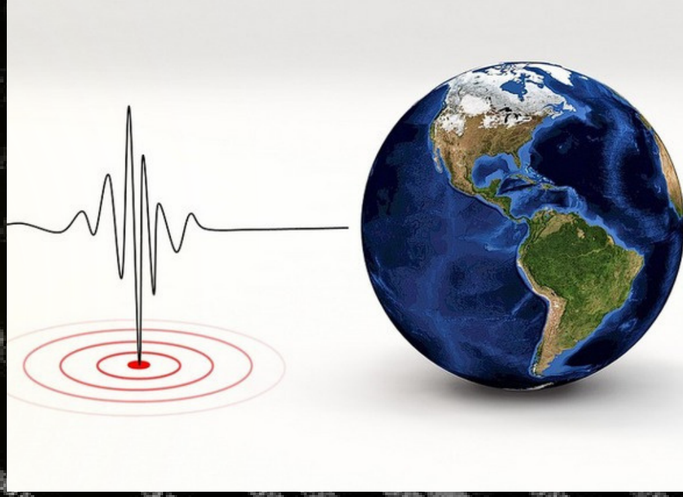
M 7.0 - Solomon Islands

VIII DYFI | VII ShakeMap | GREEN PAGER

Main Tools : Selenium



the data



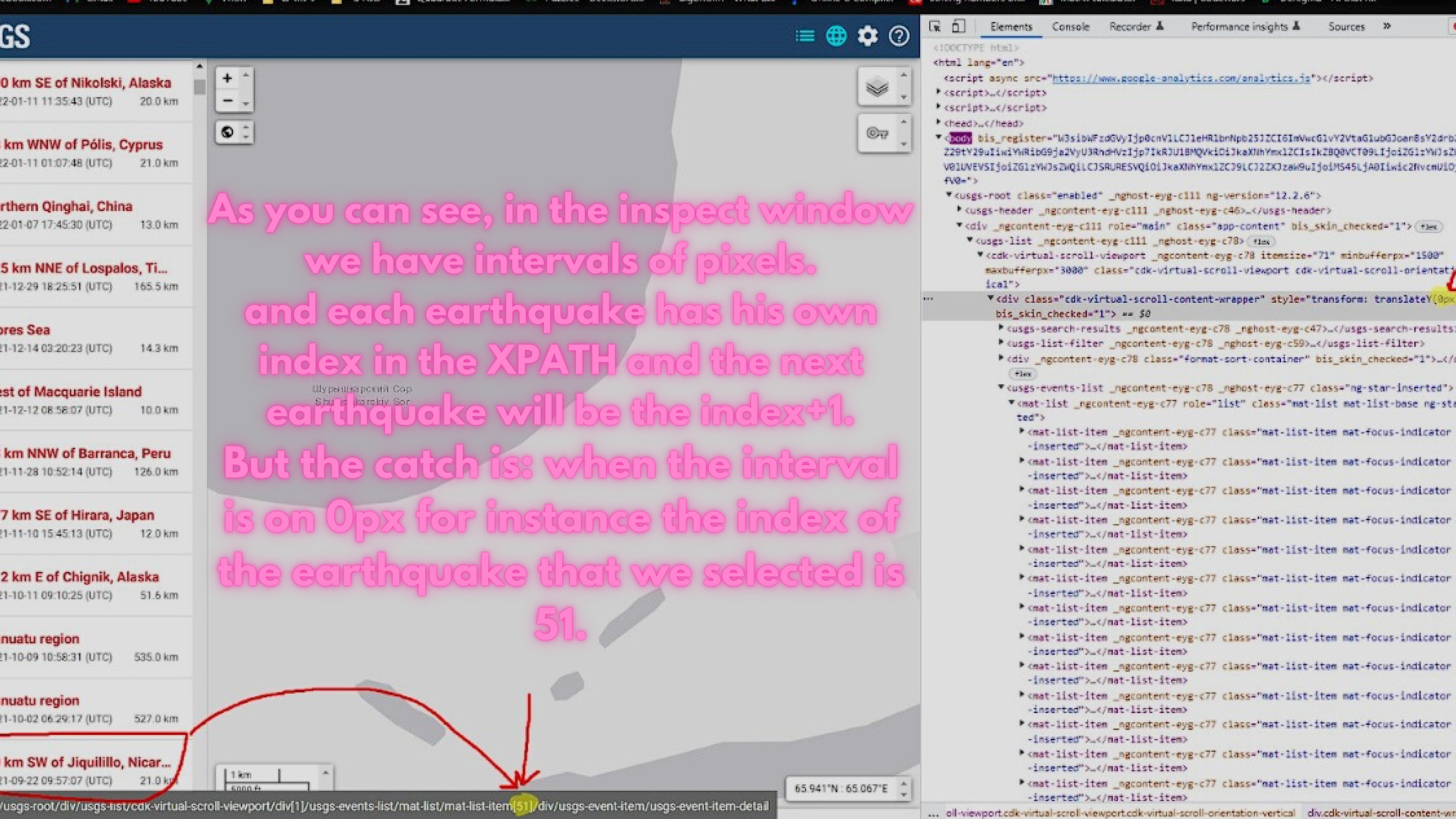
Location
Magnitude
Depth
DYFI(Did You Feel IT)
PAGER(The Damage)
ShakeMap(The Intensity of the earthquake)
Tsunami(Yes/No)
Latitude
Longitude


We had
some
problems…

Lazy Loading

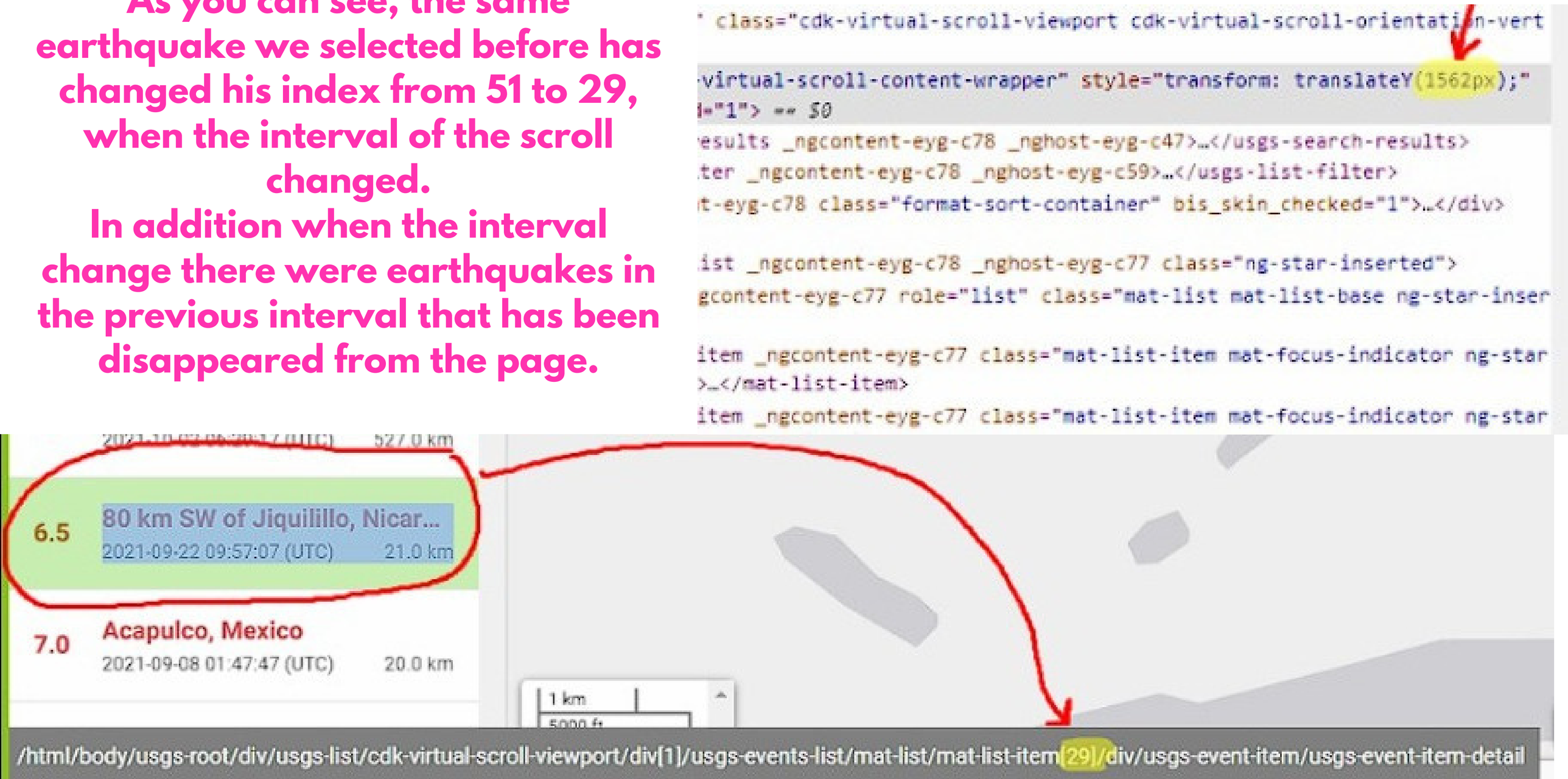
Lazy loading is the practice of delaying load or initialization of resources or objects until they're actually needed to improve performance and save system resources.

In our case we didn't only just suffer from lazy loading, every time we scrolled, elements were replaced with other elements


As you can see, in the inspect window we have intervals of pixels.
and each earthquake has his own index in the XPATH and the next earthquake will be the index+1.
But the catch is: when the interval is on 0px for instance the index of the earthquake that we selected is 51.
/usgs-root/div/usgs-list/cdk-virtual-scroll-viewport/div[1]/usgs-events-list/mat-list/mat-list-item[51]/div/usgs-event-item/usgs-event-item-detail

The Catch

As you can see, the same earthquake we selected before has changed his index from 51 to 29, when the interval of the scroll changed.
In addition when the interval change there were earthquakes in the previous interval that has been disappeared from the page.



BUT WE OVERCAME IT

We noticed that the intervals of the scrolling are raising in linear way, therefore we calculated the intervals which were effected by the browser size, and we scrolled with the selenium accordingly.

```
for load_page in range(525):


    for j in range(1,75):
        driver.execute_script(
        f"document.querySelector('cdk-virtual-scroll-viewport').scrollTop={jump}")
        time.sleep(1)
        print(j)
        try:
```

```
    close = driver.find_element(
        By.XPATH, "/html/body/usgs-root/div/usgs-details-info-box/mat-card/mat-card-actions/button/span[1]"
    close.click()
jump+=1562
```

The DataFrame

The site was collapsed every 100 minutes, therefore we had to create several data frames and then to concat it.

```
frame=[pf, pf2, pf3, pf4, pf5, pf6, pf7, pf8, pf9,
 pf10, pf11, pf12, pf13, pf14, pf15,
 pf16, pf17, pf18, pf19, pf20,
  pf21,pf22]
```

[11586 rows x 11 columns]

AND FINALLY....

Magnitude	Location	Time	Depth	DYFI	ShakeMap	PAGER	Tsunami	Longitude	Latitude
3.8	4 km SW of Fuig, Puerto Rico	2022-12-24 11:51:57 (UTC)	7.0 km	V	III	NaN	0	17.965°N	66.948°W
3.7	4 km WSW of Guánica, Puerto Rico	2022-12-24 04:27:28 (UTC)	8.0 km	V	III	NaN	0	17.957°N	66.947°W
3.7	NaN	2022-12-22 17:57:33 (UTC)	78.0 km	III	II	NaN	0	63.249°N	148.390°W
5.5	NaN	2022-12-22 13:13:55 (UTC)	157.0 km	V	IV	GREEN	0	8.042°S	74.503°W
3.8	6km NW of Petrolia, CA	2022-12-22 11:49:55 (UTC)	28.7 km	III	III	NaN	0	40.360°N	124.335°W
...	...	...	...	...	...	...	...	...	...
6.7	3 km ESE of Browns Point, Washington	1965-04-29 15:28:45 (UTC)	64.7 km	VIII	VII	NaN	0	47.288°N	122.406°W
9.2	1964 Prince William Sound Earthquake, Alaska	1964-03-28 03:36:16 (UTC)	25.0 km	IX	VIII	NaN	0	60.908°N	147.339°W
	6km WNW of	1952-07-21							

Dealing with duplicates

we had a lot of duplicates in each partial dataframe
therefore we dropped all of them, and we assembled them

FORMATTING

Importing every non - numeric variable into a fitted categorical data, Also transferring the column "Date" to datatime type for using it more easily to approach the year month or day accordingly.

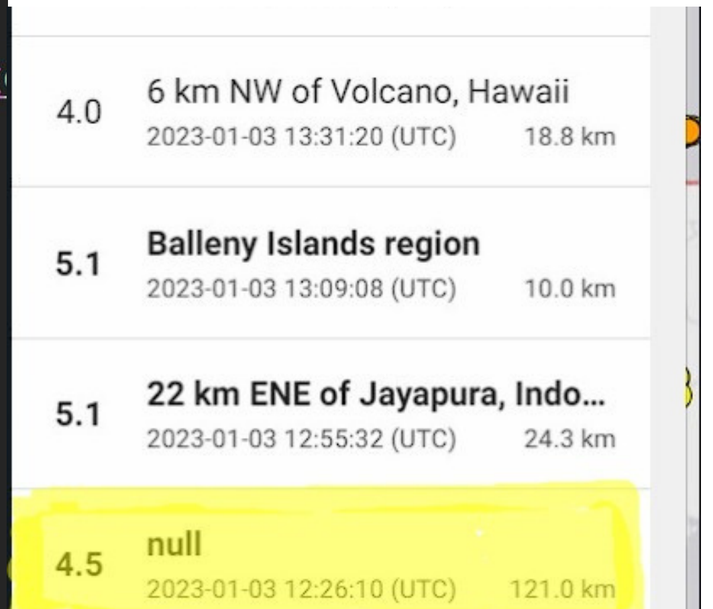
In addition the latitude and longitude data was presented as gps coordinates with the degree symbol and N-north, W-west, S-south, E-east letters, we had to dispose the symbol and convert it to homogenic float, we multiplied by -1 if it has S or W.

scrubbing data

Dealing with missing values -

In the PAGER column we had more than 50% missing values, therefore we dropped the entire column.
In addition we had some missing locations, that's why we used API which uses Latitude and Longitude that we had, to locate the exact location of the earthquake.

```
tuple_element = df[['Longitude', 'Latitude']].apply(lambda x: tuple(x), axis=1)
states=[]
for cordinate in tuple_element:
    url = f'https://nominatim.openstreetmap.org/reverse?lat={cordinate[1]}&lon={cordinate[
    time.sleep(1)
    response = requests.get(url)

# Check the response status code to make sure the request was successful
    if response.status_code == 200:
    # Parse the JSON response
        res = response.json()
        try:
            states.append(res["display_name"])


        except:
            states.append("Sea")
```



RING OF FIRE

From Wikipedia

"The Ring of Fire (also known as the Pacific Ring of Fire, the Rim of Fire, the Girdle of Fire or the Circum-Pacific belt) is a region around much of the rim of the Pacific Ocean where many volcanic eruptions and earthquakes occur."

We have read about the pacific ring of fire, and we understood that in this area has a lot of potential to have a severe earthquakes.

We chose to add another column that indicates if each of the earthquakes were in the Ring of Fire zone.

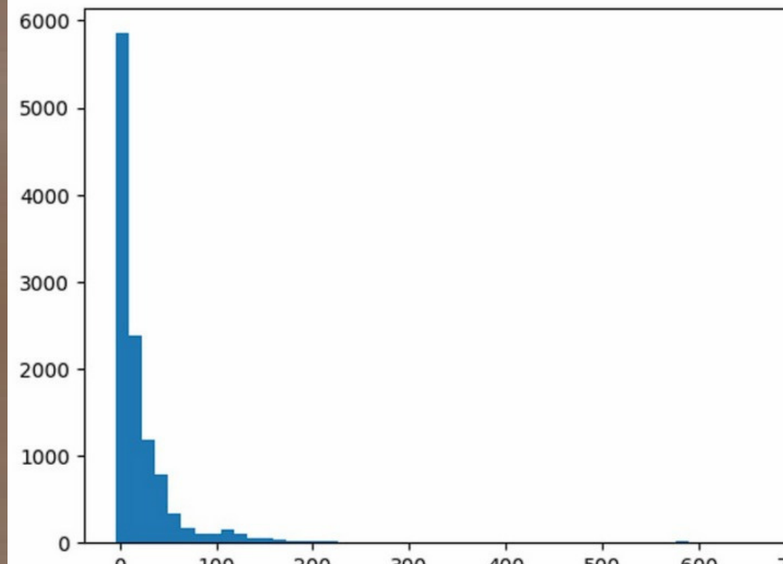
IS SEA

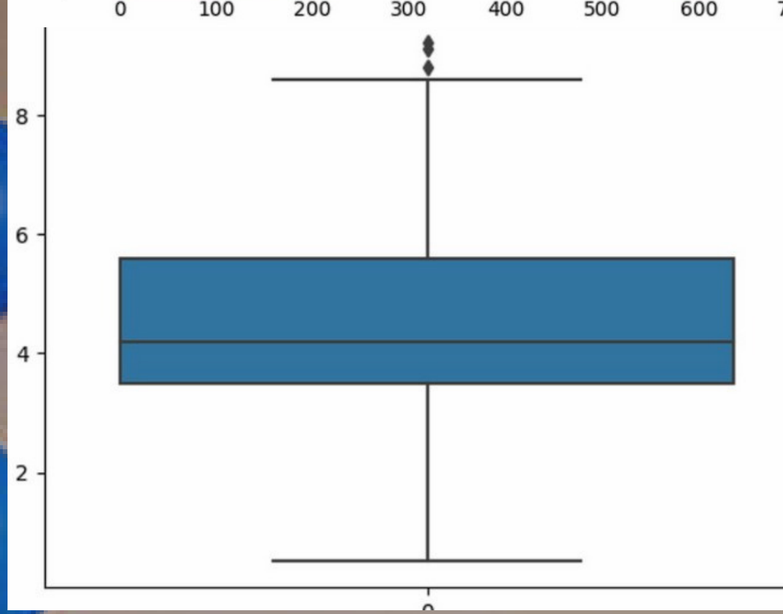
We chose to add another column which indicates if the earthquake source were in sea or not, that is because earthquakes iccur in the water is more likely to have Tsunami.

Outliers

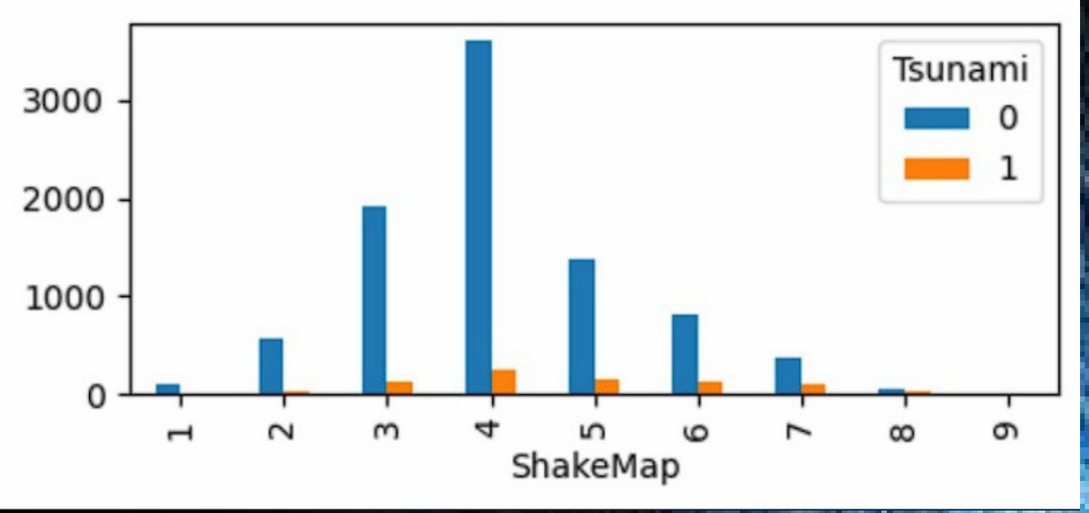
We have searched for outliers using boxplot histogram and IQR

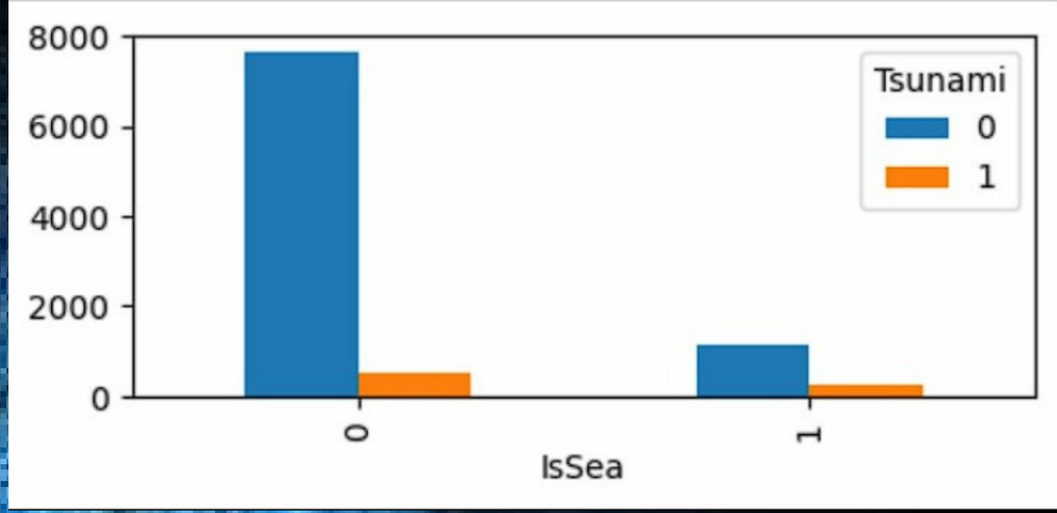
Even though we did found "outliers" we decided to keep them because they represent valid data in our data frame.

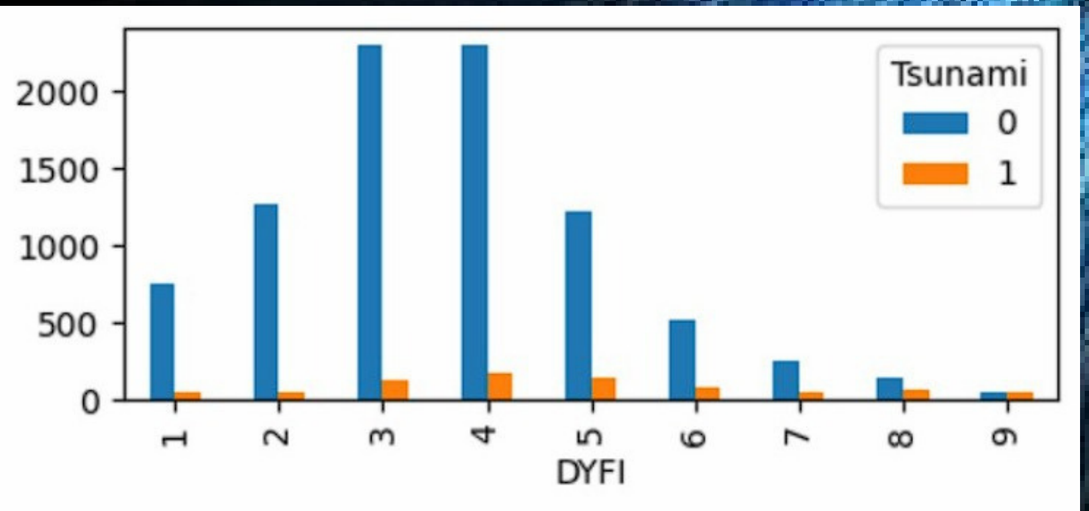


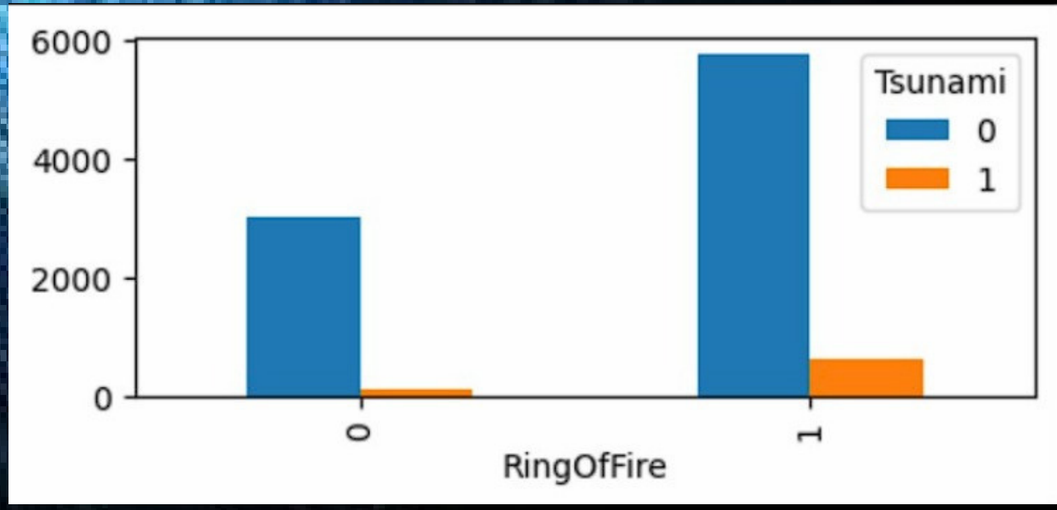


Exploring & Visualizing Data


Tsunami
0
1
3000
2000
1000
0
1
2
3
4
5
6
7
8
9
ShakeMap


Tsunami
0
1
8000
6000
4000
2000
0
0
1
IsSea


Tsunami
0
1
2000
1500
1000
500
0
1
2
3
4
5
6
7
8
9
DYFI


Tsunami
0
1
6000
4000
2000
0
0
1
RingOfFire

(Chi)^2 Contingency

Correlation between: Tsunami and RingOfFire

Tsunami and IsSea

```
ct1 = pd.crosstab(df3['IsSea'], df3['Tsunami'])
ct1.plot(kind='bar',figsize=(5,2))
ct1
chi2_contingency(ct1)
#  corolation
```

```
(200.48981235525645,
 1.6328433302440943e-45,
 1,
```

Tsunami and RingOfFire

```
ct1 = pd.crosstab(df3['RingOfFire'], df3['Tsunami'])
ct1.plot(kind='bar',figsize=(5,2))
ct1
chi2_contingency(ct1)
# good coralation
```

```
(108.26914278853367,
 2.346494203859814e-25,
 1,
```

Tsunami and ShakeMap

```
ct1 = pd.crosstab(df3['ShakeMap'], df3['Tsunami'])
ct1.plot(kind='bar',figsize=(5,2))
ct1
chi2_contingency(ct1)

#good corr
```

```
(305.1827930001843,
 3.246410220355922e-61,
 8,
```

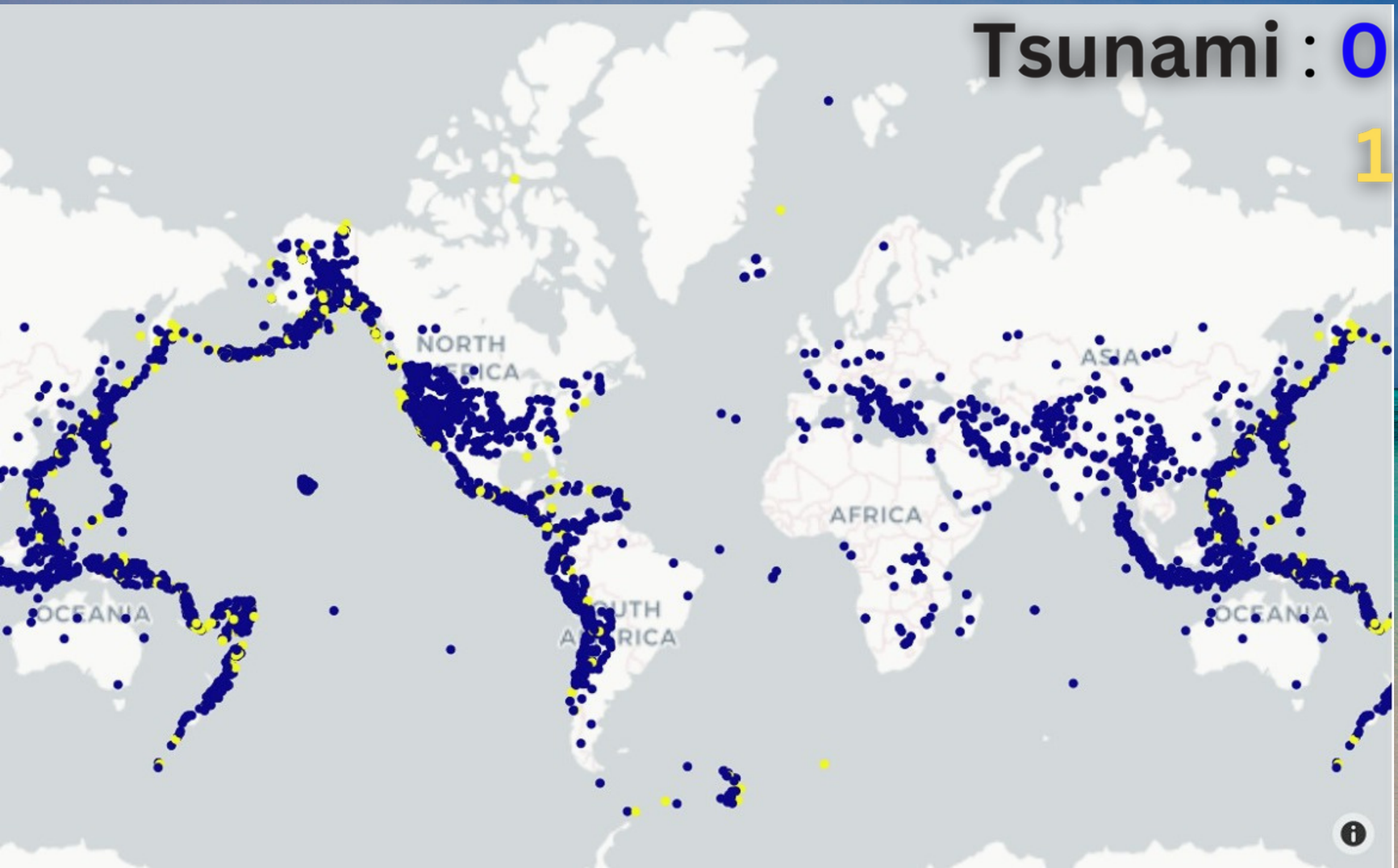
Tsunami and DYFI

```
ct1 = pd.crosstab(df3['DYFI'], df3['Tsunami'])
ct1.plot(kind='bar',figsize=(5,2))
ct1
chi2_contingency(ct1)
#good corr
```

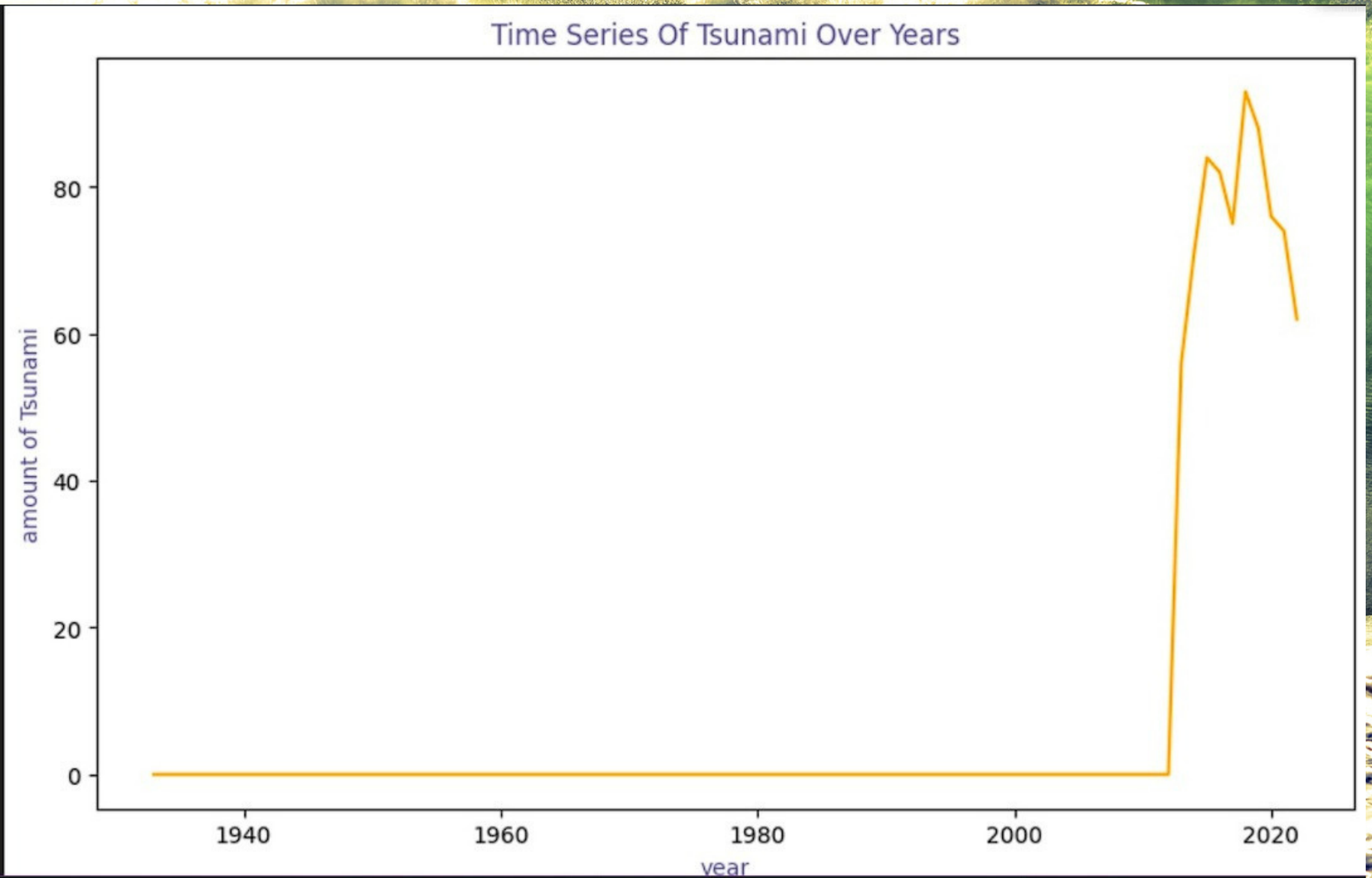
✓ 0.4s

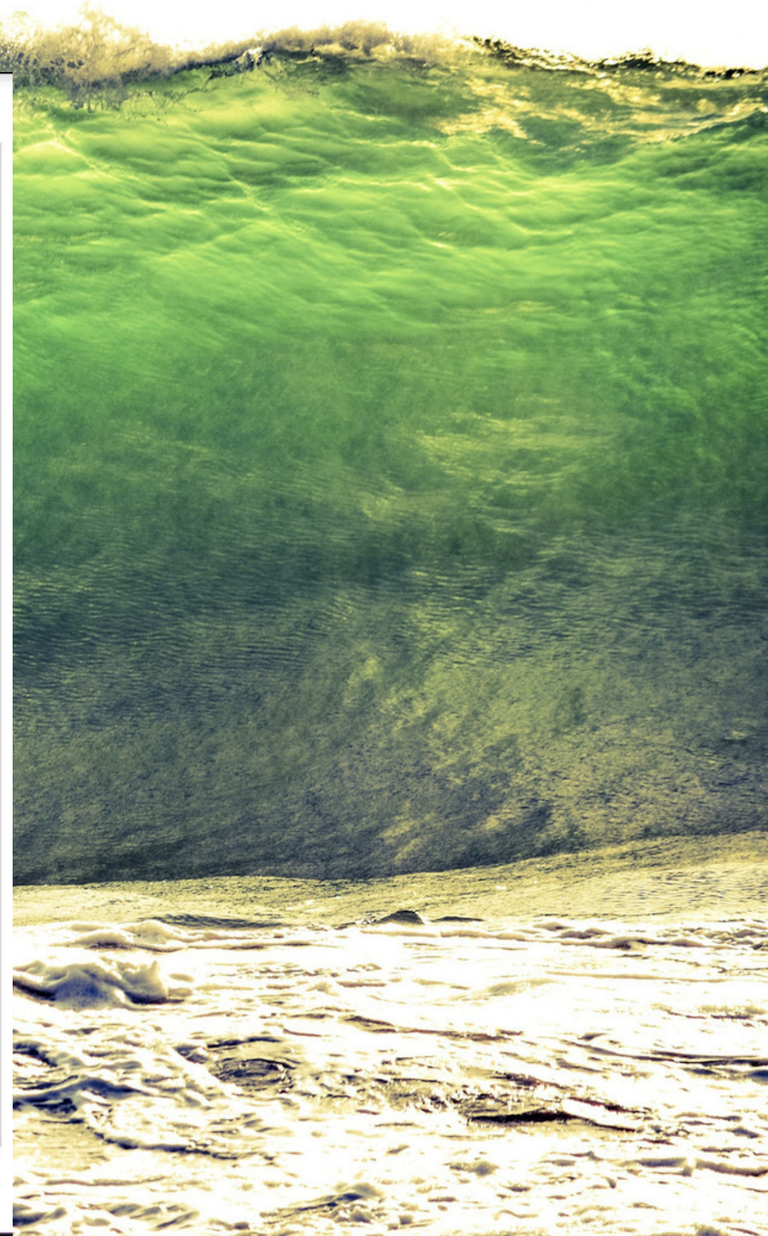
```
(464.3026733505944,
 3.182306674990775e-95,
 8,
```

Can you tell where there are more yellows?



During the EDA process we noticed that there were 0 tsunamis before the 20's century, and this is of course not correct.



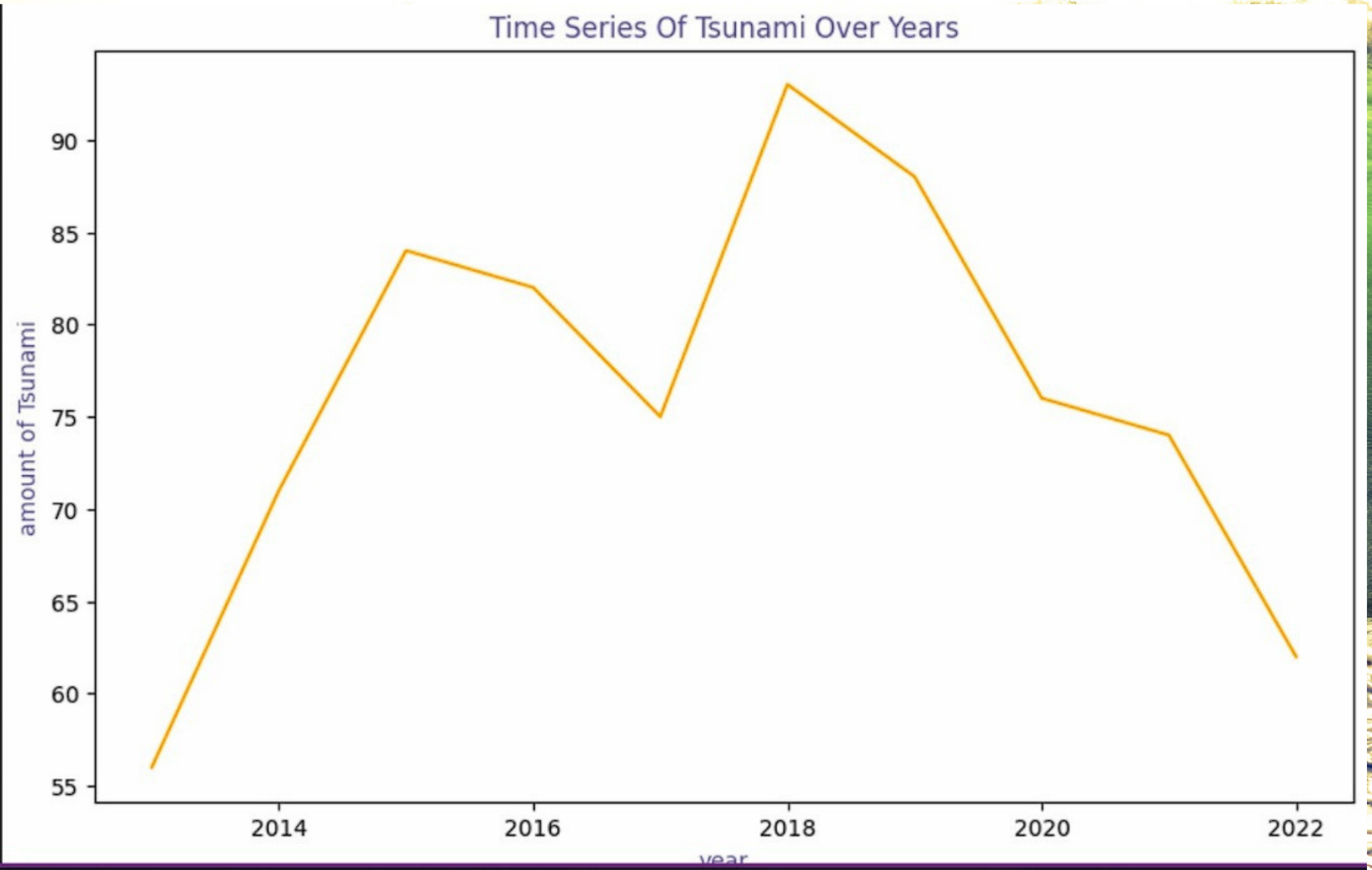


We assumed that the reason might be that the website didn't report the tsunamis until a certain year.

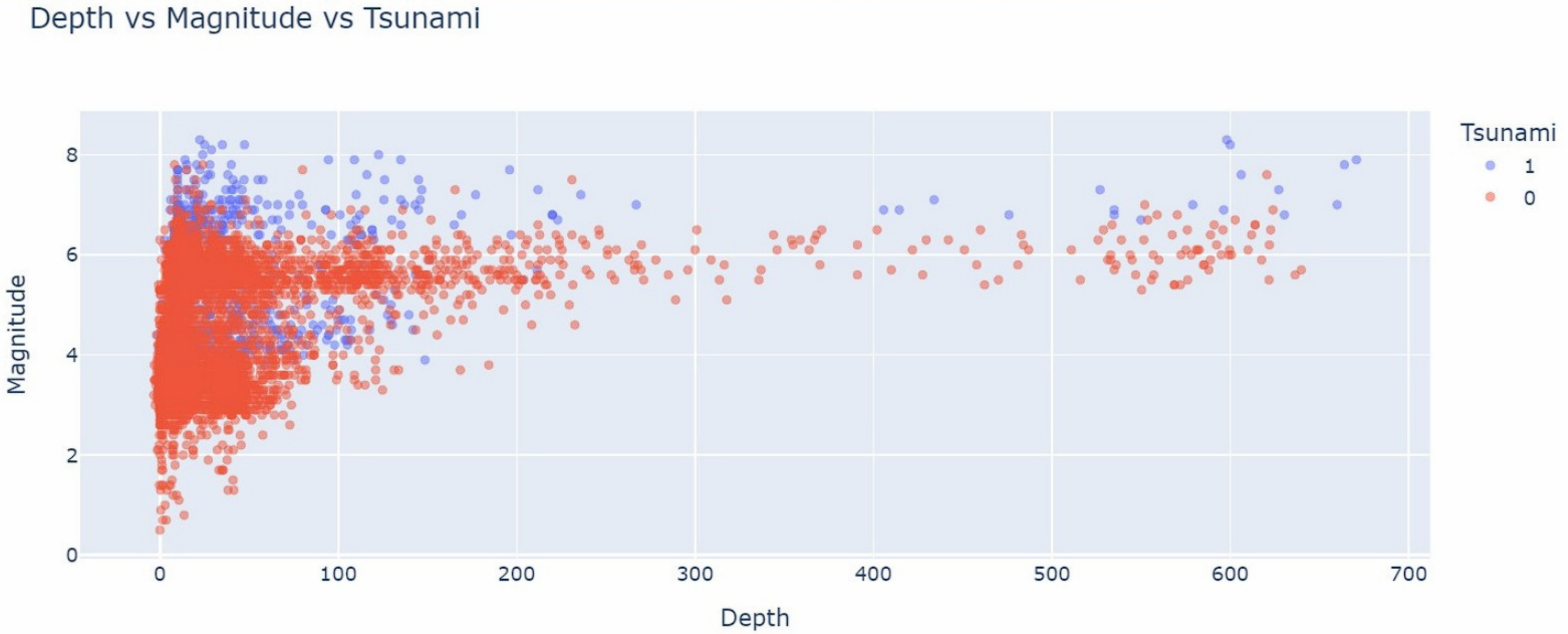
```
df3.sort_values(by='Date', ascending=True, inplace=True)
df3.reset_index(inplace=True)

for i,element in enumerate(df3["Tsunami"]):
    if element ==1:
        print(i)
        break

df3=df3.iloc[1728:]
```

```
1728
```

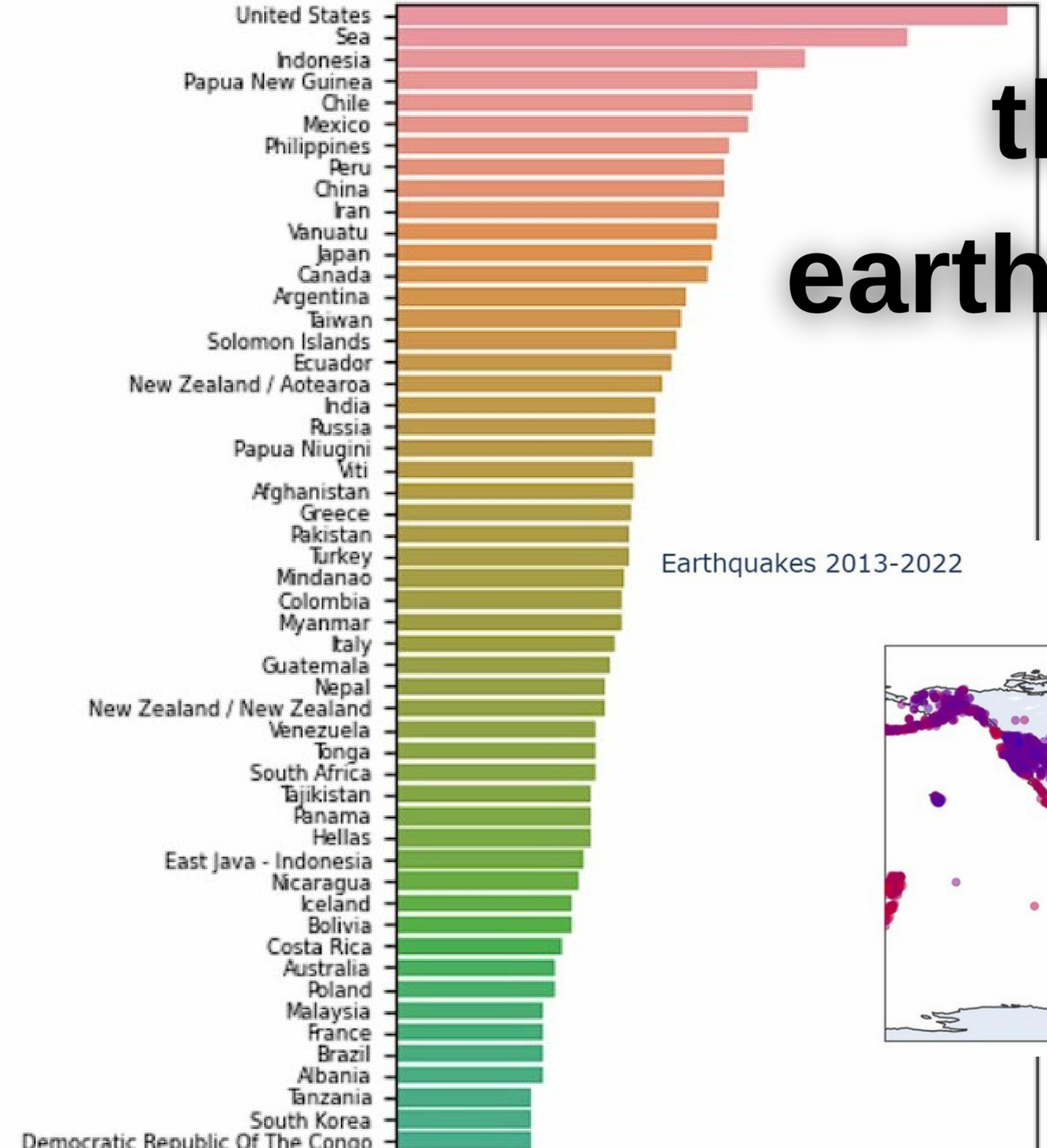
After the cleaning

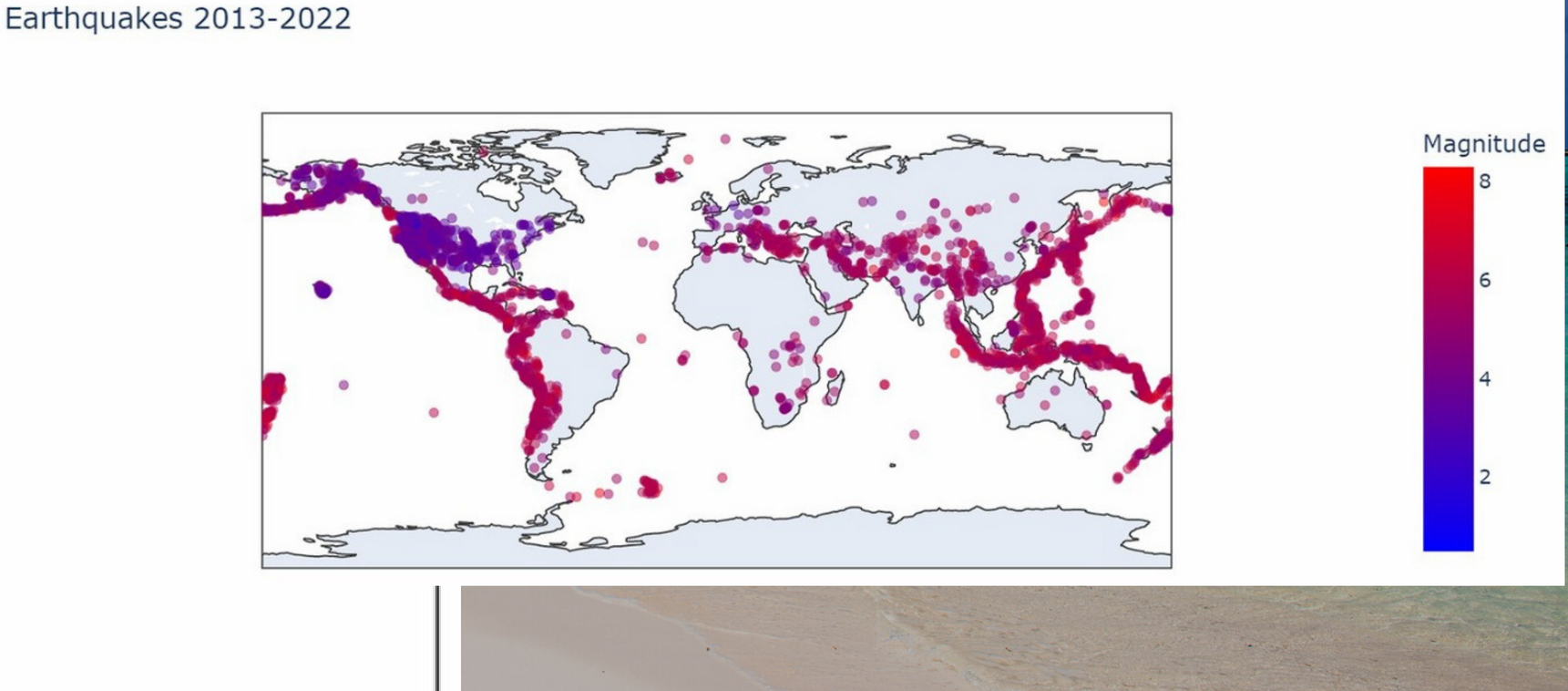


We noticed that the graph has logarithm shape, and the most of the tsunamis occur in shallow depth and high magnitude



the amount of earthquakes per state





MODELING DATA!

After The EDA phase we noticed that the most effecting features on the Tsunami occurring are:
Magnitude, Depth, Latitude, Longitude, DYFI, ShakeMap, IsSea, RingofFire.

We splitted it to Discrete and Continues columns

```
X_discrete= df3[["DYFI","ShakeMap","IsSea","RingOfFire"]]


features2 = df3[["Longitude", "Latitude",
                 "Depth", 'Magnitude']]
```

SUPERVISED LEARNING - CLASSIFICATION

we will try and predict which earthquake will cause to tsunami by supervise learning -classification

Logistic Regression

Because most of the earthquakes doesn't contain tsunamis we have "Bias".
therefore we decided to give more weight for the smaller group, and less for the wider.

```
859]

..   Predicting...
            displaying information
     Model train Accuracy:  0.8844944997380828
     Model test Accuracy:  0.8696335078534031
     [[1560  203]
      [  46  101]]
     the precision is:0.3322368421052631
     the recall_score is:0.6870748299319728
     the f1_score is:0.4478935698447893
```

```
model = LogisticRegression(class_weight={1: 0.85, 0: 0.15})
model.fit(X_train_processed, y_train)
```

Random forest classifier

Because random forest efficiency depends on the parameters given,
we used CLF algorithm to detect which parameters are the best for the machine learning

```
model2 = RandomForestClassifier(max_depth=7, min_samples_split=15)
model2.fit(X_train_processed, y_train)
describe_output_for_model(model, X_test_processed, y_test)
```

```
Predicting...
        displaying information
Model train Accuracy:  0.891042430591933
Model test Accuracy:  0.8732984293193717
[[1566  197]
 [  45  102]]
the precision is:0.3411371237458194
the recall_score is:0.6938775510204082
the f1_score is:0.4573991031390134
```

Decision Tree

```
from sklearn.model_selection import GridSearchCV
from sklearn import tree

clf=tree.DecisionTreeClassifier()
params={"max_depth":[2,3,4,5,6,7],"min_samples_split":[5,10,15,20,25,30]}
clfCV=GridSearchCV(clf,params,cv=10)
clfCV.fit(X_train_processed, y_train)
print(f"best params are:{clfCV.best_params_}")
print(f"best score are:{clfCV.best_score_}")
```

```
model2 = tree.DecisionTreeClassifier(max_depth=7, min_samples_split=10)
```

```
model2.fit(X_train_processed, y_train)
describe_output_for_model(model2,X_test_processed,y_test)
```

```
Predicting...
        displaying information
Model train Accuracy:  0.961891042430592
Model test Accuracy:  0.9256544502617801
[[1693   70]
 [  72   75]]
the precision is:0.5172413793103449
the recall_score is:0.5102040816326531
the f1_score is:0.5136986301369864
```

Neural Network

After many efforts and push backs, we found the best method for our model.

We tried to find with CLF the best parameters for the neuron network machine learning but it seems to take a long time, as we read online, it could take months... that's why we tried to find good combinations and we did.

```
from sklearn.neural_network import MLPClassifier
clf = MLPClassifier(random_state=1, max_iter=370, hidden_layer_sizes=300,
                    activation="relu").fit(X_train_processed, y_train)
describe_output_for_model(clf, X_test_processed, y_test)
```

```
Predicting...
        displaying information
Model train Accuracy:  0.9577003666841278
Model test Accuracy:  0.9403141361256544
[[1702   61]
 [  53   94]]
the precision is:0.6064516129032258
the recall_score is:0.6394557823129252
the f1_score is:0.6225165562913908
```


WE GOT VERY GOOD RESULT FROM THE NEURAL NETWORK, BUT CAN WE MAXMIZE OUR RESULT EVEN MORE?

WE TRIED TO ACHIEVE MORE FROM OUR DATA..

The website report about the distance of the earthquake from the populated location

We thought that it might be a correlation between the distance to the effect on tsunamis

1	0	46 km E of Pedro Bay, Alaska	46.0	United States
1	0	23 km SSE of San Pedro de Atacama, Chile	23.0	Chile
1	0	17 km NW of Ciranjang-hilir, Indonesia	17.0	Indonesia
1	0	91 km NNW of Hihifo, Tonga	91.0	Sea

```
dist_from_source=[]

for element in df["Source"]:
    try:
        dist_from_source.append(int(element.split("km")[0].strip()))
    except:
        dist_from_source.append(None)
df["dis_from_source"]=dist_from_source
```

We indeed found correlation between the distance to the other features.

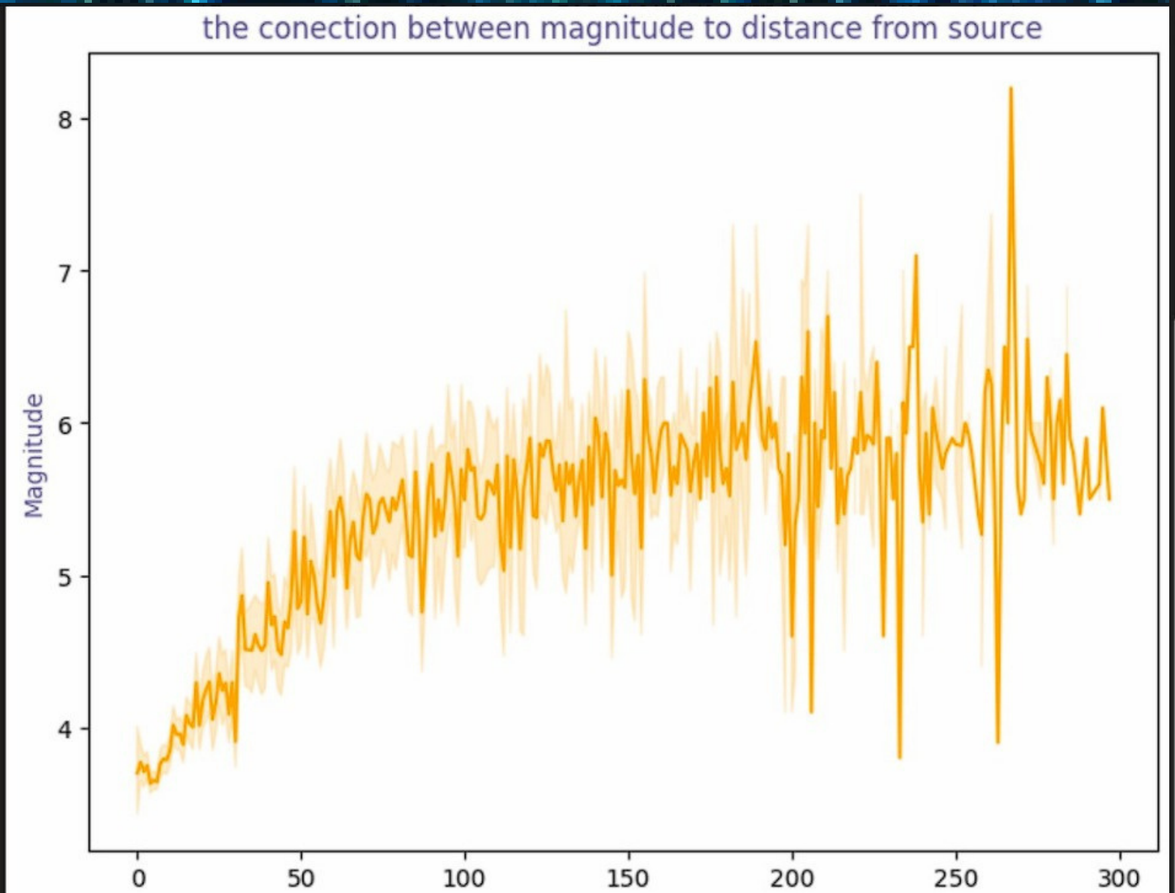
```
bins = [0,10,30,50,70,90,110,130,150,200]
labels = ['0-20','20-40','40-60','60-80','80-100','100-120','120-140','140-160','>160']
df3['dis_from_source_discrete'] = pd.cut(df3['dis_from_source'], bins, labels=labels)
```

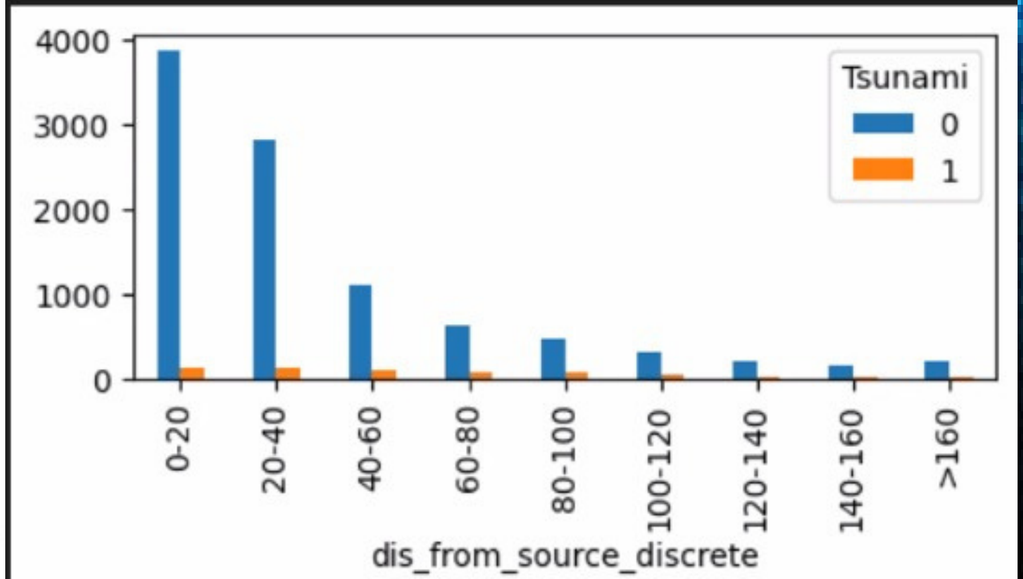
[15] ✓ 0.2s

```
ct1 = pd.crosstab(df3['dis_from_source_discrete'], df3['Tsunami'])
ct1.plot(kind = 'bar', figsize = (5, 2))
chi2_contingency(ct1)
```

[16] ✓ 1.6s

```
(264.33168202008585,
 1.5711065699315177e-52,
 8,
 array([[3738.50908222,  246.49091778]
```





CAN WE GET MORE FORE THE DATA?

As you can see, there is a correlation between the state to the Tsunami

But...
There are to many different states, and that will lead to a problem in the machine learning.
and that is why we decided to find the ratio between the amount of earthquakes of each state to the amount of the tsunamis

```
ct1 = pd.crosstab(df3["State"], df3['Tsunami'])
chi2_contingency(ct1)
```

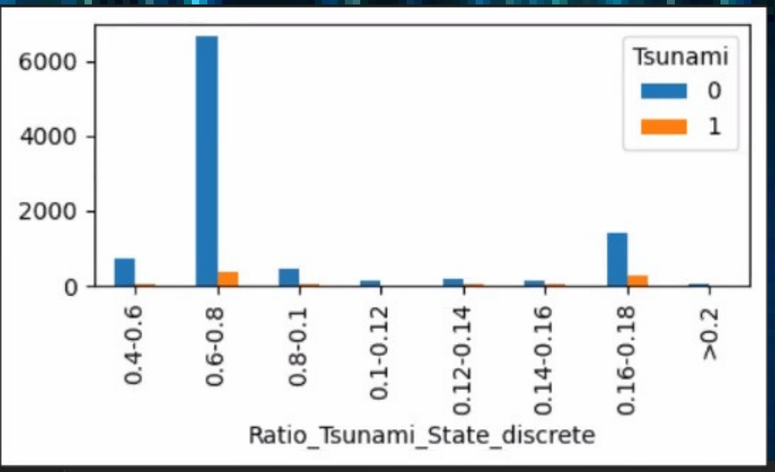
```
(367.2928711223977,
 2.1492191905417132e-26,
 122,
```

```
df3["Tsunami"] = df3["Tsunami"].astype("int")
grouped = df3.groupby("State")
counts = {state: grouped.get_group(
    state)['Tsunami'].sum() for state in df3["State"].unique()}
amount= {state: grouped.get_group(
    state)["State"].count() for state in df3["State"].unique()}
ratio_tsunami_for_state={state: counts[state]/amount[state] for state in df3["State"].unique()}
```

```
rank_list = [ratio_tsunami_for_state[state] for state in df3["State"]]
df3["Ratio_Tsunami_State"]=rank_list
```

THE EFFECT

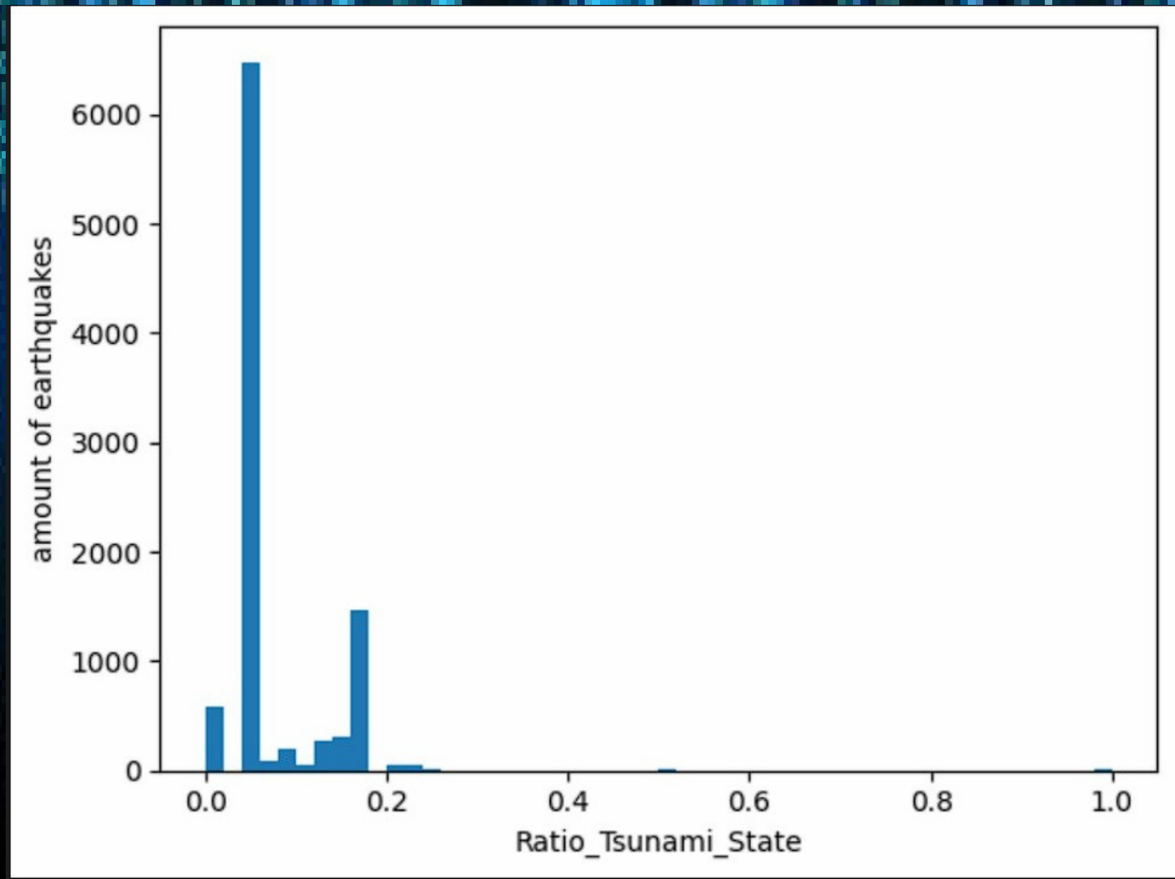
We are able to see there is a huge correlation between Ratio of the, tsunami to earthquake for each state.




```
ct1 = pd.crosstab(df3["Ratio_Tsunami_State_discrete"], df3['Tsunami'])
ct1.plot(kind='bar', figsize=(5, 2))
chi2_contingency(ct1)
```

[25] ✓ 1.8s

```
(247.56219977342676,
 9.149547258607137e-50,
 7,
 array([[6.78781508e+02, 5.32184922e+01],
        [6.50498945e+03, 5.10010551e+02],
        [4.72921542e+02, 3.70784577e+01],
        [1.15912143e+02, 9.08785728e+00],
```



The Machine learning with the additional features

After the adding that we did, The new features we provide to the machine learning are:

```
X_discrete= df3[["DYFI","ShakeMap","RingOfFire"]]
```

```
df3[["Longitude", "Latitude",
     "Depth", 'Magnitude', "Ratio_Tsunami_State", "dis_from_source"]
```

After we made the "dis_from_source" column, the earthquakes that are located in the sea weren't selected of course. that is why we dropped the "IsSea" column for the machine learning

THE MACHINE LEARNING RESULTS

LOGISTIC REGRESSION

```
Predicting...
        displaying information
Model train Accuracy:  0.899806522940851 3
Model test Accuracy:  0.870718232044198 8
[[1474  191]
 [  43  102]]
the precision is:0.34812286689419797
the recall_score is:0.7034482758620689
the f1_score is:0.465753424657534 3
```

Random Forest

```
Predicting...
         displaying information
Model train Accuracy:  0.8998065229408513
Model test Accuracy:  0.8707182320441988
[[1474  191]
 [  43  102]]
the precision is:0.34812286689419797
the recall_score is:0.7034482758620689
the f1_score is:0.4657534246575343
```

Decision Tree

```
model2 = tree.DecisionTreeClassifier(max_depth=6, min_samples_split=5)

model2.fit(X_train_processed, y_train)
describe_output_for_model(model2,X_test_processed,y_test)
```

✓ 0.2s

```
Predicting...
        displaying information
Model train Accuracy:  0.9582642343836374
Model test Accuracy:  0.9342541436464088
[[1607   58]
 [  61   84]]
the precision is:0.5915492957746479
the recall_score is:0.5793103448275863
the f1_score is:0.5853658536585367
```

NEURAL NETWORK

```
from sklearn.neural_network import MLPClassifier
clf = MLPClassifier(random_state=1, max_iter=370, hidden_layer_sizes=400,
                    activation="relu").fit(X_train_processed, y_train)
describe_output_for_model(clf, X_test_processed, y_test)
```

✓ 48.3s

```
Predicting...
        displaying information
Model train Accuracy:  0.9617191818684356
Model test Accuracy:  0.9248618784530387
[[1583   82]
 [  54   91]]
the precision is:0.5260115606936416
the recall_score is:0.6275862068965518
the f1_score is:0.5723270440251572
```

As you can see the adding of additional features wasn't successful and the first model was the BEST

THE FIRST MODEL BEST RESULT:

```
        displaying information
Model train Accuracy:  0.9577003666841278
Model test Accuracy:  0.9403141361256544
[[1702   61]
 [  53   94]]
the precision is:0.606451612903225
the recall_score is:0.6394557823129252
the f1_score is:0.6225165562913908
```

Interpreting Data

CONCLUSIONS :

- The earthquakes in the ring of fire are significantly has more chance for tsunami.
- The Depth & Magnitude & Intensity of the earthquake has huge factor for a potential Tsunami.
- In the machine learning phase we discovered that less is more - when we added more features we had worse result.
- and some machine learning algorithms works better with different data.

THE FINAL CONCLUSION

We succeed to anticipate tsunami in earthquake, by -Magnitude, Depth, Location and Intensity.
The model was 62% accurate.

Concerns

The data of the earthquakes that we took were from the last 10 years, therefore the model might not be suitable to predict Tsunamis for the long term.

Thank you For Watching

Tal Lilo

tallilo206361321@gmail.com

Dor Mizrahi

DorMizrahi1209@gmail.com