Analysis of World Earthquake Data

Let's begin by uploading the dataset into a data frame using Pandas.

This dataset was downloaded from [Earthquake dataset](). This is in CSV file format.

Here the name earthquake_df, refers to a data frame that has not been processed yet and may require cleaning, filtering, and modification before it can be used for analysis.

```python
import pandas as pd
```

```
/opt/conda/lib/python3.9/site-packages/pandas/core/computation/expressions.py:21:
UserWarning: Pandas requires version '2.8.0' or newer of 'numexpr' (version '2.7.3'
currently installed).
  from pandas.core.computation.check import NUMEXPR_INSTALLED
/opt/conda/lib/python3.9/site-packages/pandas/core/arrays/masked.py:62: UserWarning:
Pandas requires version '1.3.4' or newer of 'bottleneck' (version '1.3.2' currently
installed).
  from pandas.core import (
```

```python
earthquake_df = pd.read_csv("earthquake_data.csv")
```

```python
earthquake_df
```

	title	magnitude	date_time	cdi	mmi	alert	tsunami	sig	net	nst	dmin	gap	magType	depth
0	M 7.0 - 18 km SW of Malango, Solomon Islands	7.0	22-11-2022 02:03	8	7	green	1	768	us	117	0.509	17.0	mww	14.000
1	M 6.9 - 204 km SW of Bengkulu, Indonesia	6.9	18-11-2022 13:37	4	4	green	0	735	us	99	2.229	34.0	mww	25.000
2	M 7.0 -	7.0	12-11-2022 07:09	3	3	green	1	755	us	147	3.125	18.0	mww	579.000
3	M 7.3 - 205 km ESE of Neiafu, Tonga	7.3	11-11-2022 10:48	5	5	green	1	833	us	149	1.865	21.0	mww	37.000
4	M 6.6 -	6.6	09-11-2022 10:14	0	2	green	1	670	us	131	4.998	27.0	mww	624.464
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
777	M 7.7 - 28 km SSW of Puerto El Triunfo, El Sal...	7.7	13-01-2001 17:33	0	8	NaN	0	912	us	427	0.000	0.0	mwc	60.000

	title	magnitude	date_time	cdi	mmi	alert	tsunami	sig	net	nst	dmin	gap	magType	depth
778	M 6.9 - 47 km S of Old Harbor, Alaska	6.9	10-01-2001 16:02	5	7	NaN	0	745	ak	0	0.000	0.0	mw	36.400
779	M 7.1 - 16 km NE of Port-Olry, Vanuatu	7.1	09-01-2001 16:49	0	7	NaN	0	776	us	372	0.000	0.0	mwb	103.000
780	M 6.8 - Mindanao, Philippines	6.8	01-01-2001 08:54	0	5	NaN	0	711	us	64	0.000	0.0	mwc	33.000
781	M 7.5 - 21 km SE of Lukatan, Philippines	7.5	01-01-2001 06:57	0	7	NaN	0	865	us	324	0.000	0.0	mwc	33.000

782 rows × 19 columns

Let's look at an overview of the dataset.

```
earthquake_df.shape
```

```
(782, 19)
```

The dataset contains 782 records according to earthquakes from 1/1/2001 to 1/1/2023.

Let's view the list of columns in the data frame.

```
earthquake_df.columns
```

```
Index(['title', 'magnitude', 'date_time', 'cdi', 'mmi', 'alert', 'tsunami',
       'sig', 'net', 'nst', 'dmin', 'gap', 'magType', 'depth', 'latitude',
       'longitude', 'location', 'continent', 'country'],
      dtype='object')
```

It seems that abbreviated codes representing various types of information have been utilized as column headings.

To view the complete information type text, we can consult the schema file, which has two columns: "Column" and "Meaning". The schema file can be loaded into a Pandas series where the "Column" values serve as the index and the "Meaning" values serve as the corresponding values.

```
schema_fname = 'schema_new.csv'
schema_raw = pd.read_csv(schema_fname, index_col = "Column").Meaning
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
/tmp/ipykernel_117/3221245431.py in <module>
      1 schema_fname = 'schema_new.csv'
----> 2 schema_raw = pd.read_csv(schema_fname, index_col = "Column").Meaning

/opt/conda/lib/python3.9/site-packages/pandas/io/parsers/readers.py in
read_csv(filepath_or_buffer, sep, delimiter, header, names, index_col, usecols, dtype,
engine, converters, true_values, false_values, skipinitialspace, skiprows, skipfooter,
```

```
nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, parse_dates,
infer_datetime_format, keep_date_col, date_parser, date_format, dayfirst, cache_dates,
iterator, chunksize, compression, thousands, decimal, lineterminator, quotechar,
quoting, doublequote, escapechar, comment, encoding, encoding_errors, dialect,
on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision,
storage_options, dtype_backend)
    946     kwds.update(kwds_defaults)
    947
--> 948     return _read(filepath_or_buffer, kwds)
    949
    950

/opt/conda/lib/python3.9/site-packages/pandas/io/parsers/readers.py in
_read(filepath_or_buffer, kwds)
    609
    610     # Create the parser.
--> 611     parser = TextFileReader(filepath_or_buffer, **kwds)
    612
    613     if chunksize or iterator:

/opt/conda/lib/python3.9/site-packages/pandas/io/parsers/readers.py in __init__(self, f,
engine, **kwds)
   1446
   1447         self.handles: IOHandles | None = None
-> 1448         self._engine = self._make_engine(f, self.engine)
   1449
   1450     def close(self) -> None:

/opt/conda/lib/python3.9/site-packages/pandas/io/parsers/readers.py in
_make_engine(self, f, engine)
   1703                 if "b" not in mode:
   1704                     mode += "b"
-> 1705             self.handles = get_handle(
   1706                 f,
   1707                 mode,

/opt/conda/lib/python3.9/site-packages/pandas/io/common.py in get_handle(path_or_buf,
mode, encoding, compression, memory_map, is_text, errors, storage_options)
    861         if ioargs.encoding and "b" not in ioargs.mode:
    862             # Encoding
--> 863             handle = open(
    864                 handle,
    865                 ioargs.mode,

FileNotFoundError: [Errno 2] No such file or directory: 'schema_new.csv'
```

```
schema_raw
```

```
schema_raw['title']
```

Now that we have successfully imported the dataset, we can proceed to the next stage of preparing and refining the data for our analysis by performing various preprocessing and cleaning operations.

Before that, we will save our work to the Jupyter notebooks.

```
project_name = "analysis-of-world-earthquake-data"
```

```
import jovian
```

```
jovian.commit(project=project_name)
```

```
[jovian] Updating notebook "sithumini2400/analysis-of-world-earthquake-data" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/sithumini2400/analysis-of-world-earthquake-data
```

```
'https://jovian.com/sithumini2400/analysis-of-world-earthquake-data'
```

Data Preparation and Cleaning

Although the dataset provides a vast amount of information, we will restrict our analysis to certain specific areas:

- The location of those earthquakes occurred
- The dates and times that earthquakes occurred
- The magnitude and effects of those earthquakes

We will choose a specific subset of columns that contain the pertinent data for our analysis.

```
selected_columns = [
    "title",
    "magnitude",
    "date_time",
    "alert",
    "tsunami",
    "sig",
    "nst",
    "depth",
    "latitude",
    "longitude",
    "location",
    "continent",
    "country"

]
```

```
len(selected_columns)
```

```
13
```

We will create a new data frame, analysis_df, by extracting a copy of the data solely from the selected columns. This new data frame will allow us to make additional modifications and alterations without altering the original data frame.

```
analysis_df = earthquake_df[selected_columns].copy()
```

```
analysis_df
```

	title	magnitude	date_time	alert	tsunami	sig	nst	depth	latitude	longitude	location	contine
0	M 7.0 - 18 km SW of Malango, Solomon Islands	7.0	22-11-2022 02:03	green	1	768	117	14.000	-9.7963	159.596	Malango, Solomon Islands	Ocear
1	M 6.9 - 204 km SW of Bengkulu, Indonesia	6.9	18-11-2022 13:37	green	0	735	99	25.000	-4.9559	100.738	Bengkulu, Indonesia	Na
2	M 7.0 -	7.0	12-11-2022 07:09	green	1	755	147	579.000	-20.0508	-178.346	NaN	Ocear
3	M 7.3 - 205 km ESE of Neiafu, Tonga	7.3	11-11-2022 10:48	green	1	833	149	37.000	-19.2918	-172.129	Neiafu, Tonga	Na
4	M 6.6 -	6.6	09-11-2022 10:14	green	1	670	131	624.464	-25.5948	178.278	NaN	Na
...	...	...	...	...	...	...	...	...	...	...	...	
777	M 7.7 - 28 km SSW of Puerto El Triunfo, El Sal...	7.7	13-01-2001 17:33	NaN	0	912	427	60.000	13.0490	-88.660	Puerto El Triunfo, El Salvador	Na
778	M 6.9 - 47 km S of Old Harbor, Alaska	6.9	10-01-2001 16:02	NaN	0	745	0	36.400	56.7744	-153.281	Old Harbor, Alaska	Nor Ameri
779	M 7.1 - 16 km NE of Port-Olry, Vanuatu	7.1	09-01-2001 16:49	NaN	0	776	372	103.000	-14.9280	167.170	Port-Olry, Vanuatu	Na
780	M 6.8 - Mindanao, Philippines	6.8	01-01-2001 08:54	NaN	0	711	64	33.000	6.6310	126.899	Mindanao, Philippines	Na
781	M 7.5 - 21 km SE of Lukatan, Philippines	7.5	01-01-2001 06:57	NaN	0	865	324	33.000	6.8980	126.579	Lukatan, Philippines	Na

782 rows × 13 columns

```
schema = schema_raw[selected_columns]
```

```
schema
```

```
analysis_df.shape
```

```
(782, 13)
```

```
schema.shape
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_117/3153010046.py in <module>
----> 1 schema.shape

NameError: name 'schema' is not defined
```

```
analysis_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 782 entries, 0 to 781
Data columns (total 13 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   title      782 non-null    object
 1   magnitude  782 non-null    float64
 2   date_time  782 non-null    object
 3   alert      415 non-null    object
 4   tsunami    782 non-null    int64
 5   sig        782 non-null    int64
 6   nst        782 non-null    int64
 7   depth      782 non-null    float64
 8   latitude   782 non-null    float64
 9   longitude  782 non-null    float64
 10  location   777 non-null    object
 11  continent  206 non-null    object
 12  country    484 non-null    object
dtypes: float64(4), int64(3), object(6)
memory usage: 79.5+ KB
```

Some columns in the data frame have mixed data types or missing values represented by NaN. These columns are classified as "object" data type. Additionally, some columns have fewer non-null values than the total number of rows in the data frame (782), indicating missing values in those columns. We need to explicitly convert each column's data type as needed to handle missing values.

To simplify our analysis, we will convert the "date_time" column to datetime data type and ignore any non-numeric values by converting them to NaN.

```
analysis_df['date_time'] = pd.to_datetime(analysis_df.date_time)
```

```
/tmp/ipykernel_117/3804089582.py:1: UserWarning: Parsing dates in %d-%m-%Y %H:%M format
```

```
when dayfirst=False (the default) was specified. Pass `dayfirst=True` or specify a
format to silence this warning.
  analysis_df['date_time'] = pd.to_datetime(analysis_df.date_time)
```

```
analysis_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 782 entries, 0 to 781
Data columns (total 13 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   title      782 non-null    object
 1   magnitude  782 non-null    float64
 2   date_time  782 non-null    datetime64[ns]
 3   alert      415 non-null    object
 4   tsunami    782 non-null    int64
 5   sig        782 non-null    int64
 6   nst        782 non-null    int64
 7   depth      782 non-null    float64
 8   latitude   782 non-null    float64
 9   longitude  782 non-null    float64
 10  location   777 non-null    object
 11  continent  206 non-null    object
 12  country    484 non-null    object
dtypes: datetime64[ns](1), float64(4), int64(3), object(5)
memory usage: 79.5+ KB
```

Now let's look at some statistic data about numeric columns.

```
analysis_df.describe()
```

	magnitude	date_time	tsunami	sig	nst	depth	latitude	longi
count	782.000000	782	782.000000	782.000000	782.000000	782.000000	782.000000	782.000
mean	6.941125	2012-10-13 08:43:35.677749504	0.388747	870.108696	230.250639	75.883199	3.538100	52.609
min	6.500000	2001-01-01 06:57:00	0.000000	650.000000	0.000000	2.700000	-61.848400	-179.968
25%	6.600000	2007-10-17 20:37:15	0.000000	691.000000	0.000000	14.000000	-14.595600	-71.668
50%	6.800000	2013-06-14 17:10:30	0.000000	754.000000	140.000000	26.295000	-2.572500	109.426
75%	7.100000	2017-10-20 13:43:45	1.000000	909.750000	445.000000	49.750000	24.654500	148.941
max	9.100000	2022-11-22 02:03:00	1.000000	2910.000000	934.000000	670.810000	71.631200	179.662
std	0.445514	NaN	0.487778	322.465367	250.188177	137.277078	27.303429	117.898

Now the dataset has been cleaned, we can begin studying it. Let's take a look at a few sample records from the data frame to get an idea of what the data looks like.

```
analysis_df.sample(10)
```

	title	magnitude	date_time	alert	tsunami	sig	nst	depth	latitude	longitude	location	co
334	M 6.7 - 201 km SSE of Mata-Utu, Wallis and Futuna	6.7	2014-06-29 17:15:00	green	1	691	0	18.00	-14.9831	-175.5100	Mata-Utu, Wallis and Futuna	
278	M 7.5 - Hindu Kush region, Afghanistan	7.5	2015-10-26 09:09:00	orange	0	1448	0	231.00	36.5244	70.3676	Hindu Kush region, Afghanistan	
135	M 7.5 - 111km ESE of Palora, Ecuador	7.5	2019-02-22 10:17:00	yellow	1	1145	0	145.00	-2.1862	-77.0505	Palora, Ecuador	A
377	M 7.1 - 16 km WSW of Atiquipa, Peru	7.1	2013-09-25 16:42:00	green	1	843	0	40.00	-15.8385	-74.5112	Atiquipa, Peru	
23	M 6.9 - 284 km ESE of Tadine, New Caledonia	6.9	2022-03-30 20:56:00	green	1	738	0	10.00	-22.7200	170.2770	Tadine, New Caledonia	
762	M 6.6 - 74 km S of Kokhanok, Alaska	6.6	2001-07-28 07:32:00	NaN	0	754	0	131.80	58.7750	-154.7010	Kokhanok, Alaska	A
349	M 7.6 - 93 km SSE of Kirakira, Solomon Islands	7.6	2014-04-12 20:14:00	green	1	889	0	22.56	-11.2701	162.1480	Kirakira, Solomon Islands	
166	M 7.1 - 136 km W of Iñapari, Peru	7.1	2018-08-24 09:04:00	green	0	776	0	630.00	-11.0355	-70.8284	Iñapari, Peru	A
489	M 7.0 - 19 km NE of Methven, New Zealand	7.0	2010-09-03 16:35:00	NaN	0	1245	365	12.00	-43.5220	171.8300	Methven, New Zealand	
548	M 6.6 - 50 km S of Jurm, Afghanistan	6.6	2009-01-03 20:23:00	NaN	0	765	344	204.80	36.4190	70.7430	Jurm, Afghanistan	

Let us save and upload our work to Jovian before continuing.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "sithumini2400/analysis-of-world-earthquake-data" on
https://jovian.com
```

```
[jovian] Committed successfully! https://jovian.com/sithumini2400/analysis-of-world-earthquake-data
```

```
'https://jovian.com/sithumini2400/analysis-of-world-earthquake-data'
```

Exploratory Analysis and Visualization

Here I hope to analyze and visualize our dataset based on six main topics:

- location
- magnitude
- significance
- time
- alert
- tsunami conditions

Let's begin by importing the 'numpy', 'matplotlib.pyplot', 'seanborn' and 'geopandas' libraries.

```
pip install geopandas
```

```
Requirement already satisfied: geopandas in /opt/conda/lib/python3.9/site-packages (0.14.0)
Requirement already satisfied: shapely>=1.8.0 in /opt/conda/lib/python3.9/site-packages (from geopandas) (2.0.1)
Requirement already satisfied: pyproj>=3.3.0 in /opt/conda/lib/python3.9/site-packages (from geopandas) (3.6.1)
Requirement already satisfied: pandas>=1.4.0 in /opt/conda/lib/python3.9/site-packages (from geopandas) (2.1.1)
Requirement already satisfied: fiona>=1.8.21 in /opt/conda/lib/python3.9/site-packages (from geopandas) (1.9.4.post1)
Requirement already satisfied: packaging in /opt/conda/lib/python3.9/site-packages (from geopandas) (21.2)
Requirement already satisfied: click~=8.0 in /opt/conda/lib/python3.9/site-packages (from fiona>=1.8.21->geopandas) (8.0.3)
Requirement already satisfied: six in /opt/conda/lib/python3.9/site-packages (from fiona>=1.8.21->geopandas) (1.16.0)
Requirement already satisfied: certifi in /opt/conda/lib/python3.9/site-packages (from fiona>=1.8.21->geopandas) (2021.10.8)
Requirement already satisfied: importlib-metadata in /opt/conda/lib/python3.9/site-packages (from fiona>=1.8.21->geopandas) (6.6.0)
Requirement already satisfied: click-plugins>=1.0 in /opt/conda/lib/python3.9/site-packages (from fiona>=1.8.21->geopandas) (1.1.1)
Requirement already satisfied: cligj>=0.5 in /opt/conda/lib/python3.9/site-packages (from fiona>=1.8.21->geopandas) (0.7.2)
```

```
Requirement already satisfied: attrs>=19.2.0 in /opt/conda/lib/python3.9/site-packages
(from fiona>=1.8.21->geopandas) (21.2.0)
Requirement already satisfied: numpy>=1.22.4 in /opt/conda/lib/python3.9/site-packages
(from pandas>=1.4.0->geopandas) (1.26.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/conda/lib/python3.9/site-
packages (from pandas>=1.4.0->geopandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.9/site-packages
(from pandas>=1.4.0->geopandas) (2021.3)
Requirement already satisfied: tzdata>=2022.1 in /opt/conda/lib/python3.9/site-packages
(from pandas>=1.4.0->geopandas) (2023.3)
Requirement already satisfied: pyparsing<3,>=2.0.2 in /opt/conda/lib/python3.9/site-
packages (from packaging->geopandas) (2.4.7)
Requirement already satisfied: zipp>=0.5 in /opt/conda/lib/python3.9/site-packages
(from importlib-metadata->fiona>=1.8.21->geopandas) (3.6.0)
Note: you may need to restart the kernel to use updated packages.
```

```python
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
import geopandas as gpd
import numpy as np
%matplotlib inline

sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (15, 10)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

```
/opt/conda/lib/python3.9/site-packages/scipy/__init__.py:146: UserWarning: A NumPy
version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version
1.26.0
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

Location

First, we get an idea about where the past earthquakes occurred by marking a map. To do this, we use the Geopandas, Seaborn and Matplotlib libraries.

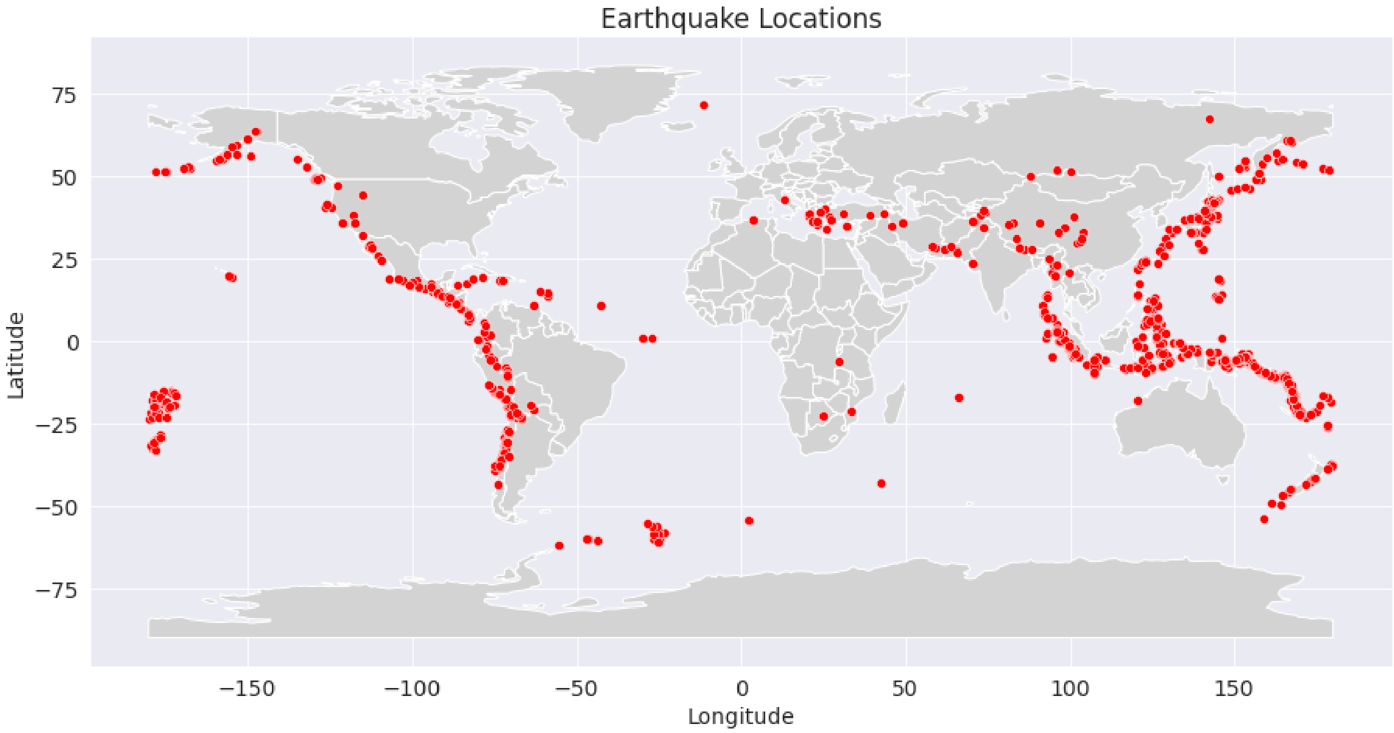
```python
#Create a world map using geopandas
countries = gpd.read_file(
     gpd.datasets.get_path("naturalearth_lowres"))
countries.plot(color="lightgrey")

# Create a scatterplot of earthquake locations using seaborn
sns.scatterplot(x="longitude", y="latitude", data=analysis_df, color = "Red")

# Add a title and axis labels
plt.title("Earthquake Locations")
```

```
plt.xlabel("Longitude")
plt.ylabel("Latitude");
```

```
/tmp/ipykernel_117/3079784467.py:3: FutureWarning: The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from https://www.naturalearthdata.com/downloads/110m-cultural-vectors/.
  gpd.datasets.get_path("naturalearth_lowres"))
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



Upon analyzing the map, it becomes apparent that a significant number of earthquakes have occurred during the previous time period in various parts of the world. However, the highest concentration of earthquakes took place in Asian countries, such as Japan, Indonesia, and China, as well as in North and South American regions, including the west coast of the United States, Mexico, and Chile. This suggests that these regions are prone to seismic activity and may require increased preparedness and preventative measures to mitigate potential damage and loss of life.

To further explore the regions with the highest number of earthquakes, we can employ the use of the value_counts() function. This function will help to provide a more comprehensive view of the frequency and distribution of earthquakes in these regions.

```
most_location = analysis_df["location"].value_counts().head(20)
```

```
most_location
```

```
location
Kirakira, Solomon Islands        17
Kokopo, Papua New Guinea         16
Sola, Vanuatu                    15
Panguna, Papua New Guinea        13
Lata, Solomon Islands            11
Kimbe, Papua New Guinea          11
Tadine, New Caledonia            11
South Sandwich Islands region    10
Bengkulu, Indonesia               8
Kermadec Islands, New Zealand     8
Port-Vila, Vanuatu                8
Namie, Japan                      7
Isangel, Vanuatu                  7
Padang, Indonesia                 7
Sungai Penuh, Indonesia           7
Manokwari, Indonesia              7
Iquique, Chile                    7
Nikolski, Alaska                  6
Sinabang, Indonesia               6
Port-Olry, Vanuatu                6
Name: count, dtype: int64
```

From the above data, it is difficult to get a clear idea. So we visualize the above information with a bar plot to get a very clear idea. For this, we will use the Matplotlib and Seaborn libraries.

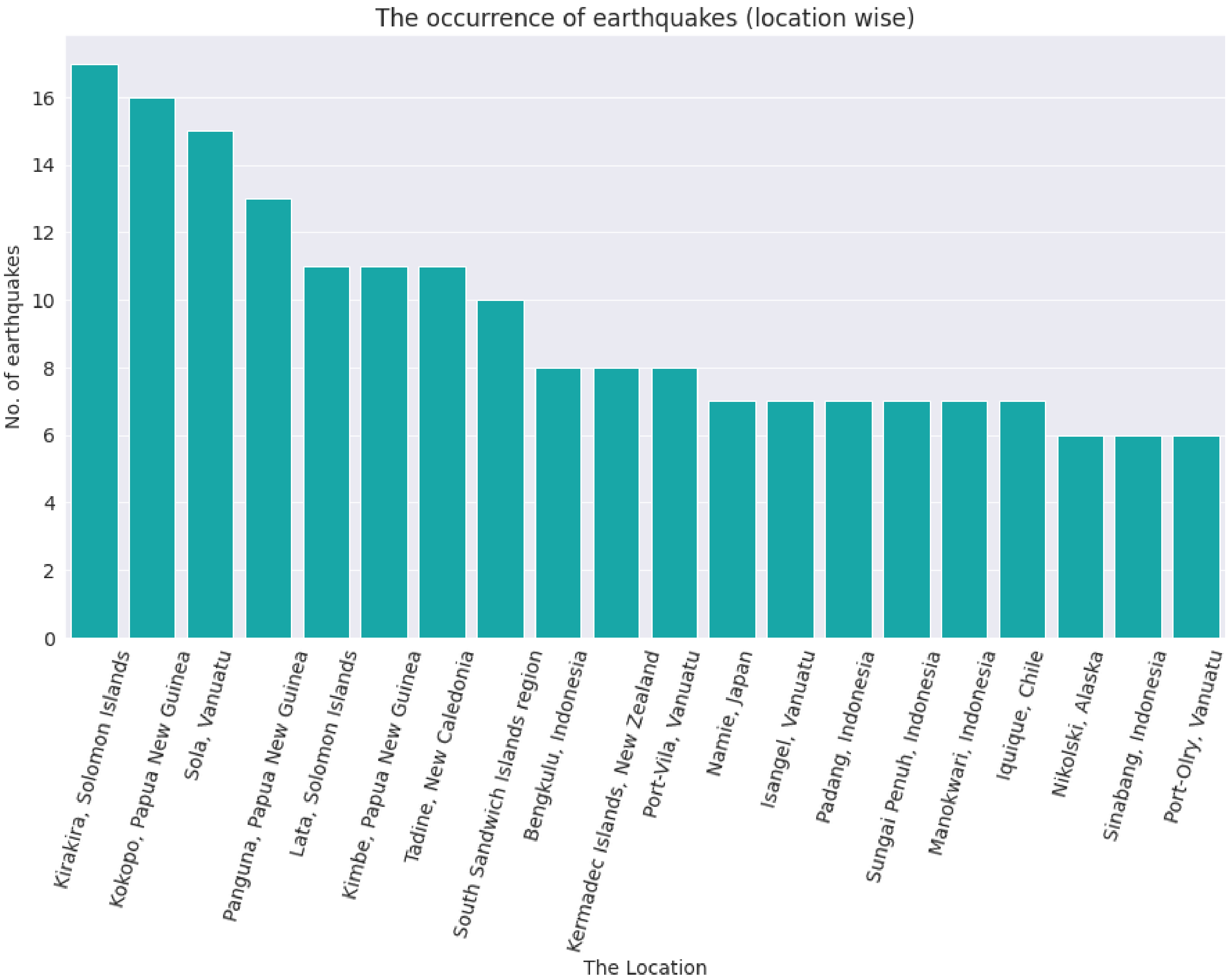
```
#Create a barchart using Seaborn
plt.figure(figsize = (15,8))
sns.barplot( x= most_location.index, y = most_location, color = "c")

#Rotate the x axis labels in 75 degrees
plt.xticks(rotation = 75)

#Set the plot title and axis labels
plt.xlabel("The Location")
plt.ylabel("No. of earthquakes")
plt.title("The occurrence of earthquakes (location wise)");
```

```
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```

```
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



The bar chart provided offers valuable insights into the frequency and distribution of earthquakes during the period from January 1st, 2001 to January 1st, 2023. By analyzing the data presented, we can observe that Kirakira in the Solomon Islands has experienced the highest number of earthquakes with a total of 17 occurrences. This highlights the vulnerability of this region to seismic activity and may require additional preparedness measures to minimize potential damage and loss of life.

Furthermore, the bar chart reveals that Kokopo in Papua New Guinea and Sola in Vanuatu have also experienced a high number of earthquakes during the same period, with 16 and 15 occurrences respectively. This information is crucial for disaster management agencies in these regions, as it can inform the allocation of resources and aid in the development of response plans to mitigate the impacts of future earthquakes.

Overall, the data presented in the bar chart underscores the importance of preparedness and preventative measures in regions with a history of seismic activity. By being aware of areas of high earthquake frequency, appropriate measures can be taken to reduce the impact of these natural disasters and protect the communities affected.

Let us save and upload our work to Jovian before continuing.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "sithumini2400/analysis-of-world-earthquake-data" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/sithumini2400/analysis-of-world-earthquake-data
```

```
'https://jovian.com/sithumini2400/analysis-of-world-earthquake-data'
```

Magnitude

The magnitude of an earthquake is a measure of its size or strength, determined by the amount of energy released during the earthquake. The magnitude scale is logarithmic, meaning that each increase of one unit represents a tenfold increase in energy. It is an important factor in earthquake analysis as it helps to determine the potential impact of the earthquake.

Now let's see the past earthquakes according to their magnitude. For that first we take the magnitude column and get value counts for each magnitude.

```
magnitude_df = analysis_df["magnitude"].value_counts()
```

```
magnitude_df
```

```
magnitude
6.50    131
6.60    115
6.70     98
6.80     78
6.90     77
7.00     49
7.10     43
7.30     31
7.20     30
7.60     22
7.50     22
7.40     18
7.70     16
7.80     15
7.90      9
8.10      6
8.20      6
8.00      5
8.30      3
8.60      2
9.10      2
8.40      2
8.80      1
```

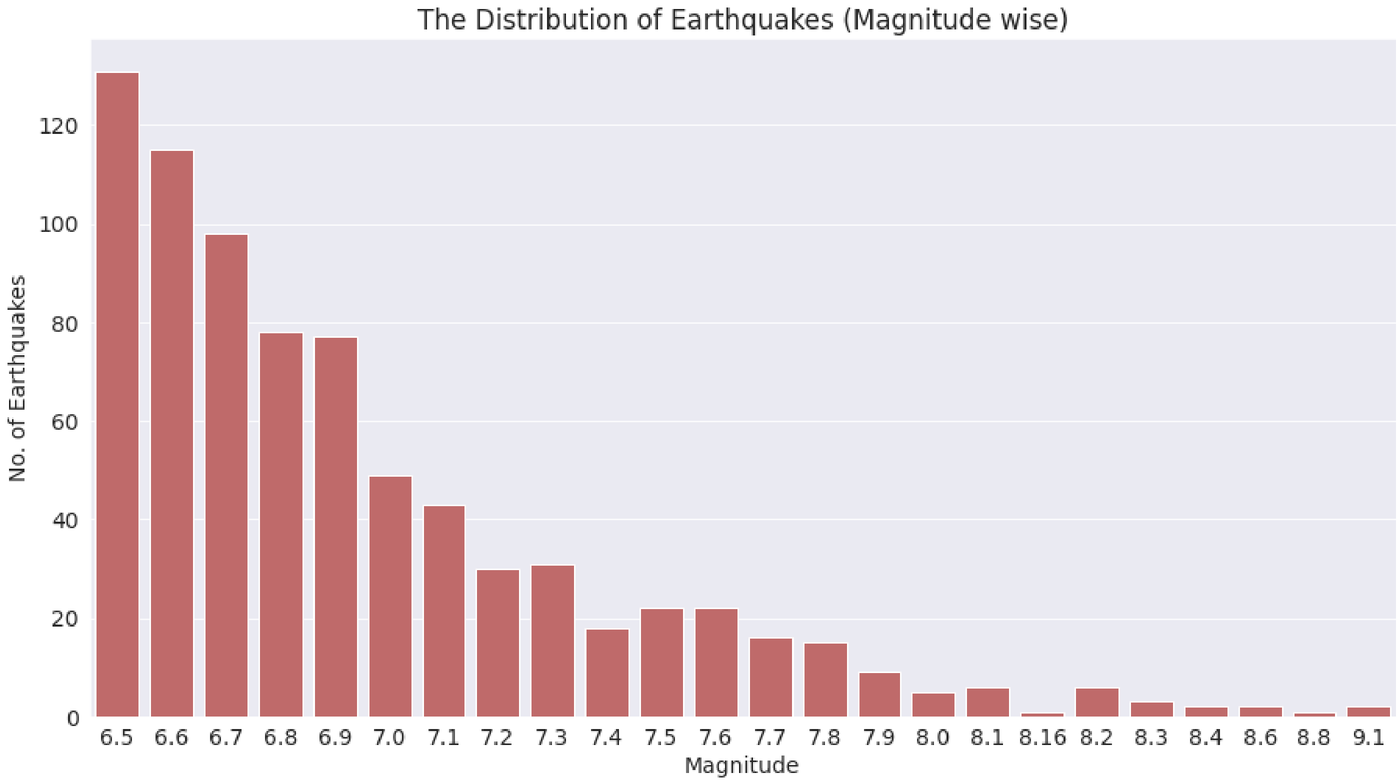
```
8.16      1
Name: count, dtype: int64
```

Now to get a very clear idea, let's create a bar plot including the above information using the Seaborn and Matplotlib libraries.

```
#Create a bar chart using Seaborn and Matplotlib
plt.figure(figsize =(15,8))
sns.barplot(magnitude_df.index, magnitude_df, color = "indianred")


#Set the plot title and axis labels
plt.title("The Distribution of Earthquakes (Magnitude wise)")
plt.xlabel("Magnitude")
plt.ylabel("No. of Earthquakes");
```

```
/opt/conda/lib/python3.9/site-packages/seaborn/_decorators.py:36: FutureWarning: Pass
the following variables as keyword args: x, y. From version 0.12, the only valid
positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



Upon analyzing the bar chart provided, it is evident that most past earthquakes have a magnitude of 6.5. A 6.5 magnitude earthquake is classified as moderately strong on the Richter magnitude scale.

A moderately strong earthquake of this magnitude can cause significant damage to poorly designed or constructed buildings, as well as infrastructure such as bridges. The shaking can also trigger landslides, liquefaction, and other types of ground failure in areas that are susceptible to such events. However, the severity of damage caused by a 6.5 earthquake can vary widely depending on a variety of factors, including the distance of the epicenter from populated areas and the local geology and infrastructure.

For instance, if the epicenter of a 6.5 magnitude earthquake is located in a remote area, with few or no nearby buildings, then the damage may be relatively limited. Conversely, if the earthquake occurs in an urban area with a high population density and poorly constructed buildings, the damage and loss of life can be much more severe.

Overall, the magnitude of an earthquake provides an important indicator of its potential impact and is a critical factor in assessing the level of preparedness and response required to mitigate the damage caused by seismic events.

Let us save and upload our work to Jovian before continuing.

```python
import jovian
```

```python
jovian.commit()
```

```
[jovian] Updating notebook "sithumini2400/analysis-of-world-earthquake-data" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/sithumini2400/analysis-of-world-earthquake-data
```

```
'https://jovian.com/sithumini2400/analysis-of-world-earthquake-data'
```

Significance

The significance of an earthquake is an essential factor to determine, as it can indicate the level of damage and loss resulting from the seismic event. In this dataset,the "sig" column contains numerical values that describe the level of significance of the earthquake.

A higher value in the "sig" column indicates a more significant event. The significance value is calculated based on several factors, such as the earthquake magnitude, the maximum Modified Mercalli Intensity (MMI), the number of reports from people who felt the earthquake, and the estimated impact of the event. Magnitude is a crucial factor in determining the significance of the earthquake since it measures the amount of energy released during the event.

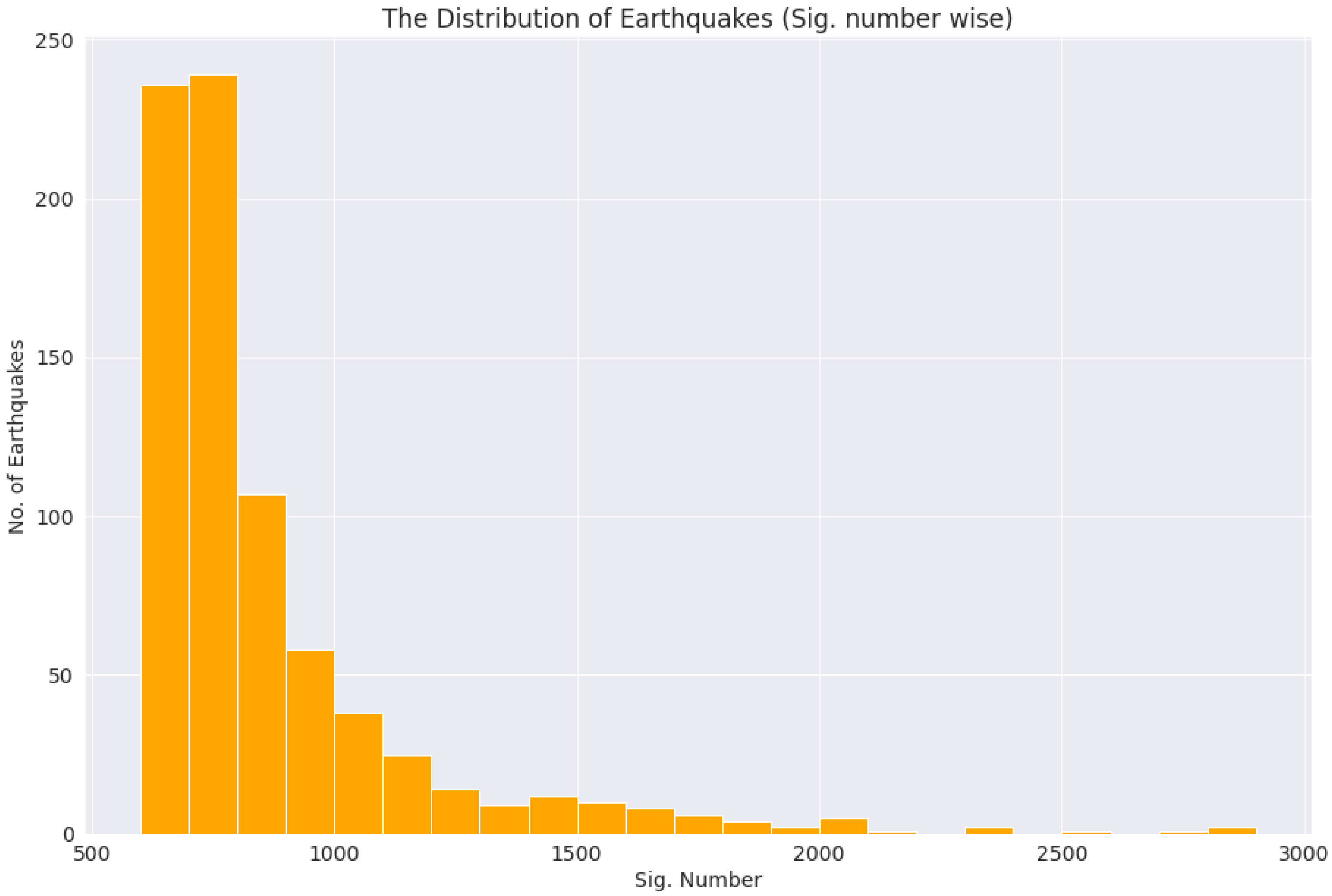
MMI, on the other hand, measures the strength of the earthquake based on how people and structures in the affected area felt the shaking. Felt reports provide valuable data for determining the level of significance of an earthquake as they give insight into the intensity and extent of the shaking. Finally, estimated impact refers to the projected level of damage and loss resulting from the earthquake, considering the population density, building construction, infrastructure quality, and potential for secondary hazards such as landslides and tsunamis.

Now we see our data frame according to their "sig" number.

```
#Extract the "sig" column
sig_df = analysis_df.sig

#Create a histogram using Matplotlib
plt.hist(sig_df, bins=np.arange(600,3000,100), color = "Orange")

#Set the plot title and axis labels
plt.title("The Distribution of Earthquakes (Sig. number wise)")
plt.xlabel("Sig. Number")
plt.ylabel("No. of Earthquakes");
```



Based on the histogram provided, we can observe that the majority of earthquakes in the dataset have "sig" numbers ranging between 600 and 800.

This range of "sig" numbers indicates that these earthquakes are moderately significant events, but with a lower potential for damage and loss of life than earthquakes with higher "sig" numbers.

However, it is crucial to note that the significance of an earthquake cannot be solely determined by its "sig" number, as several other factors can influence the impact of the seismic event. The depth of the earthquake's focus, its location, and the population density of the affected area are some of the critical factors that can significantly impact the severity of an earthquake. Earthquakes with shallow depths and occurring in densely populated areas are more likely to cause significant damage and loss of life, even if their "sig" number is relatively low. Therefore, understanding the overall context and the specific details of an earthquake is essential in assessing its significance and potential impact.

Now we sort our dataset by their sig number and see what is the most significant earthquake that occurred in the past time period from 2001/01/01 to 2023/01/01.

```
biggest_df = analysis_df.sort_values(by = "sig", ascending = False)
```

```
biggest_df
```

	title	magnitude	date_time	alert	tsunami	sig	nst	depth	latitude	longitude	location	conti
507	M 7.2 - 12km SW of Delta, B.C., MX	7.2	2010-04-04 22:40:00	red	0	2910	10	9.987	32.2862	-115.2950	Delta, B.C., MX	N Ame

	title	magnitude	date_time	alert	tsunami	sig	nst	depth	latitude	longitude	location	conti
198	M 8.2 - near the coast of Chiapas, Mexico	8.2	2017-09-08 04:49:00	red	1	2910	0	47.390	15.0222	-93.8993	Chiapas, Mexico	N
235	M 6.6 - 5 km ESE of Preci, Italy	6.6	2016-10-30 06:40:00	red	0	2840	0	8.000	42.8621	13.0961	Preci, Italy	Eur
308	M 7.8 - 67 km NNE of Bharatpur, Nepal	7.8	2015-04-25 06:11:00	red	0	2820	0	8.220	28.2305	84.7314	Bharatpur, Nepal	A
190	M 7.3 - 29 km S of ?alabja, Iraq	7.3	2017-11-12 18:18:00	red	0	2790	0	19.000	34.9109	45.9592	?alabja, Iraq	A
...	...	...	...	...	...	...	...	...	...	...	...	
756	M 6.5 - 3 km NW of Hanamaki, Japan	6.5	2001-12-02 13:01:00	NaN	0	650	605	123.800	39.4020	141.0890	Hanamaki, Japan	A
364	M 6.5 - 32 km W of Sola, Vanuatu	6.5	2014-01-01 16:03:00	green	1	650	0	187.000	-13.8633	167.2490	Sola, Vanuatu	N
375	M 6.5 - Kermadec Islands, New Zealand	6.5	2013-09-30 05:55:00	green	1	650	0	41.540	-30.9255	-178.3230	Kermadec Islands, New Zealand	N
396	M 6.5 - 33 km N of Rabaul, Papua New Guinea	6.5	2013-04-23 23:14:00	green	1	650	264	10.000	-3.8980	152.1270	Rabaul, Papua New Guinea	N
33	M 6.5 - 71 km SE of Nikolski, Alaska	6.5	2022-01-11 12:39:00	NaN	1	650	23	37.000	52.5020	-168.0800	Nikolski, Alaska	N

782 rows × 13 columns

The first row of the above dataframe shows the earthquake that had a significant impact in the past time period. To get information about it, we can use .iloc[ ] function in Python.

```
#Get the first row of the dataframe.
biggest_earthquake = biggest_df.iloc[0]

#Print the information of that earthquake.
print("The following are the details of the earthquake that caused the greatest damage
print(biggest_earthquake)
```

```
The following are the details of the earthquake that caused the greatest damage among
the earthquakes that occured from 1/1/2001 to 1/1/2023.:
```

```
title        M 7.2 - 12km SW of Delta, B.C., MX
magnitude                                  7.2
date_time                  2010-04-04 22:40:00
alert                                      red
tsunami                                      0
sig                                       2910
nst                                         10
depth                                    9.987
latitude                               32.2862
longitude                             -115.295
location                      Delta, B.C., MX
continent                        North America
country                                 Mexico
Name: 507, dtype: object
```

Let us save and upload our work to Jovian before continuing.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "sithumini2400/analysis-of-world-earthquake-data" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/sithumini2400/analysis-of-world-earthquake-data
```

```
'https://jovian.com/sithumini2400/analysis-of-world-earthquake-data'
```

Time

Let's now see how earthquakes have occurred within the last time period. First, we group the earthquake data by the month it occurred. To do this, first we need to add a new column to our data frame.

```
#add new columns "month" and "year" to the analysis_df
analysis_df['month'] = analysis_df['date_time'].dt.month
analysis_df['year'] = analysis_df['date_time'].dt.year
```

```
analysis_df
```

	title	magnitude	date_time	alert	tsunami	sig	nst	depth	latitude	longitude	location	contine
0	M 7.0 - 18 km SW of Malango, Solomon Islands	7.0	2022-11-22 02:03:00	green	1	768	117	14.000	-9.7963	159.596	Malango, Solomon Islands	Ocear
1	M 6.9 - 204 km SW of Bengkulu, Indonesia	6.9	2022-11-18 13:37:00	green	0	735	99	25.000	-4.9559	100.738	Bengkulu, Indonesia	Na

	title	magnitude	date_time	alert	tsunami	sig	nst	depth	latitude	longitude	location	contine
2	M 7.0 -	7.0	2022-11-12 07:09:00	green	1	755	147	579.000	-20.0508	-178.346	NaN	Ocear
3	M 7.3 - 205 km ESE of Neiafu, Tonga	7.3	2022-11-11 10:48:00	green	1	833	149	37.000	-19.2918	-172.129	Neiafu, Tonga	Na
4	M 6.6 -	6.6	2022-11-09 10:14:00	green	1	670	131	624.464	-25.5948	178.278	NaN	Na
...	...	...	...	...	...	...	...	...	...	...	...	
777	M 7.7 - 28 km SSW of Puerto El Triunfo, El Sal...	7.7	2001-01-13 17:33:00	NaN	0	912	427	60.000	13.0490	-88.660	Puerto El Triunfo, El Salvador	Na
778	M 6.9 - 47 km S of Old Harbor, Alaska	6.9	2001-01-10 16:02:00	NaN	0	745	0	36.400	56.7744	-153.281	Old Harbor, Alaska	Nor Ameri
779	M 7.1 - 16 km NE of Port-Olry, Vanuatu	7.1	2001-01-09 16:49:00	NaN	0	776	372	103.000	-14.9280	167.170	Port-Olry, Vanuatu	Na
780	M 6.8 - Mindanao, Philippines	6.8	2001-01-01 08:54:00	NaN	0	711	64	33.000	6.6310	126.899	Mindanao, Philippines	Na
781	M 7.5 - 21 km SE of Lukatan, Philippines	7.5	2001-01-01 06:57:00	NaN	0	865	324	33.000	6.8980	126.579	Lukatan, Philippines	Na

782 rows × 15 columns

Now we should group this dataset by its months and get the number of earthquakes in each month. To do this, we use the .groupby() function and .size() functions.

```
#Group the earthquakes by month and count the number of occurrences
monthly_earthquakes = analysis_df.groupby('month').size()
monthly_earthquakes
```

```
month
1     70
2     63
3     63
4     77
5     58
6     42
7     56
8     68
9     80
10    69
11    80
```

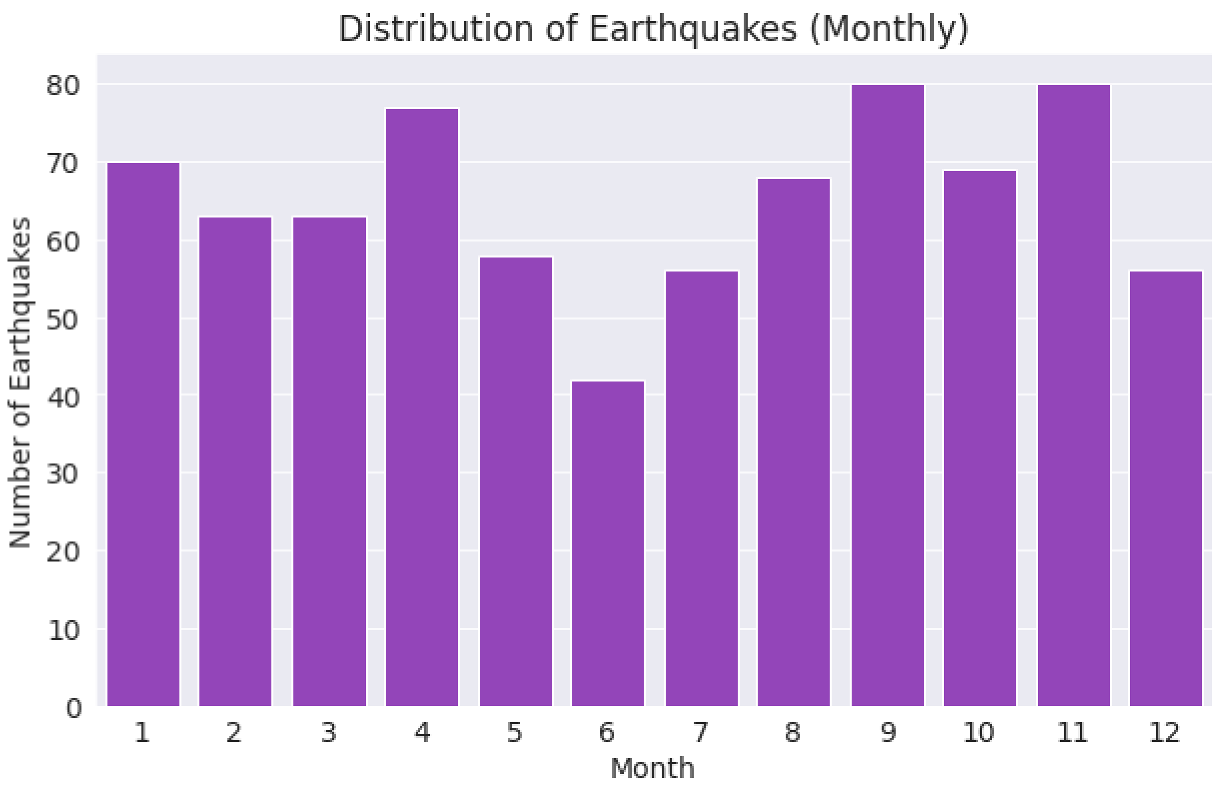
```
12    56
dtype: int64
```

Now we can plot the data in a using the Seaborn and Matplotlib libraries.

```
#Plot the number of earthquakes by month using Seaborn and Matplotlib
plt.figure(figsize=(10, 6))
sns.barplot(x = monthly_earthquakes.index, y = monthly_earthquakes, color = "darkorchic

#Set the plot title and axis labels
plt.title("Distribution of Earthquakes (Monthly)")
plt.xlabel('Month')
plt.ylabel('Number of Earthquakes');
```

```
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



Upon analyzing the bar chart provided, we can observe that the majority of earthquakes that occurred during the last time period were recorded in November, with a total of approximately 90 earthquakes. Additionally, a significant number of earthquakes were also recorded in October. This information suggests that there could be a seasonal pattern in earthquake activity.

However, it is essential to consider that this observation is limited to the dataset analyzed and may not necessarily represent a general trend. Factors such as geographic location, local weather patterns, and geological characteristics of the region can influence the occurrence and frequency of earthquakes. Hence, further research and analysis are necessary to understand the underlying factors that contribute to seasonal patterns in earthquake activity. Additionally, it is crucial to note that earthquakes can occur at any time and are inherently unpredictable, making it necessary to be prepared and have effective measures in place to mitigate their potential impact.

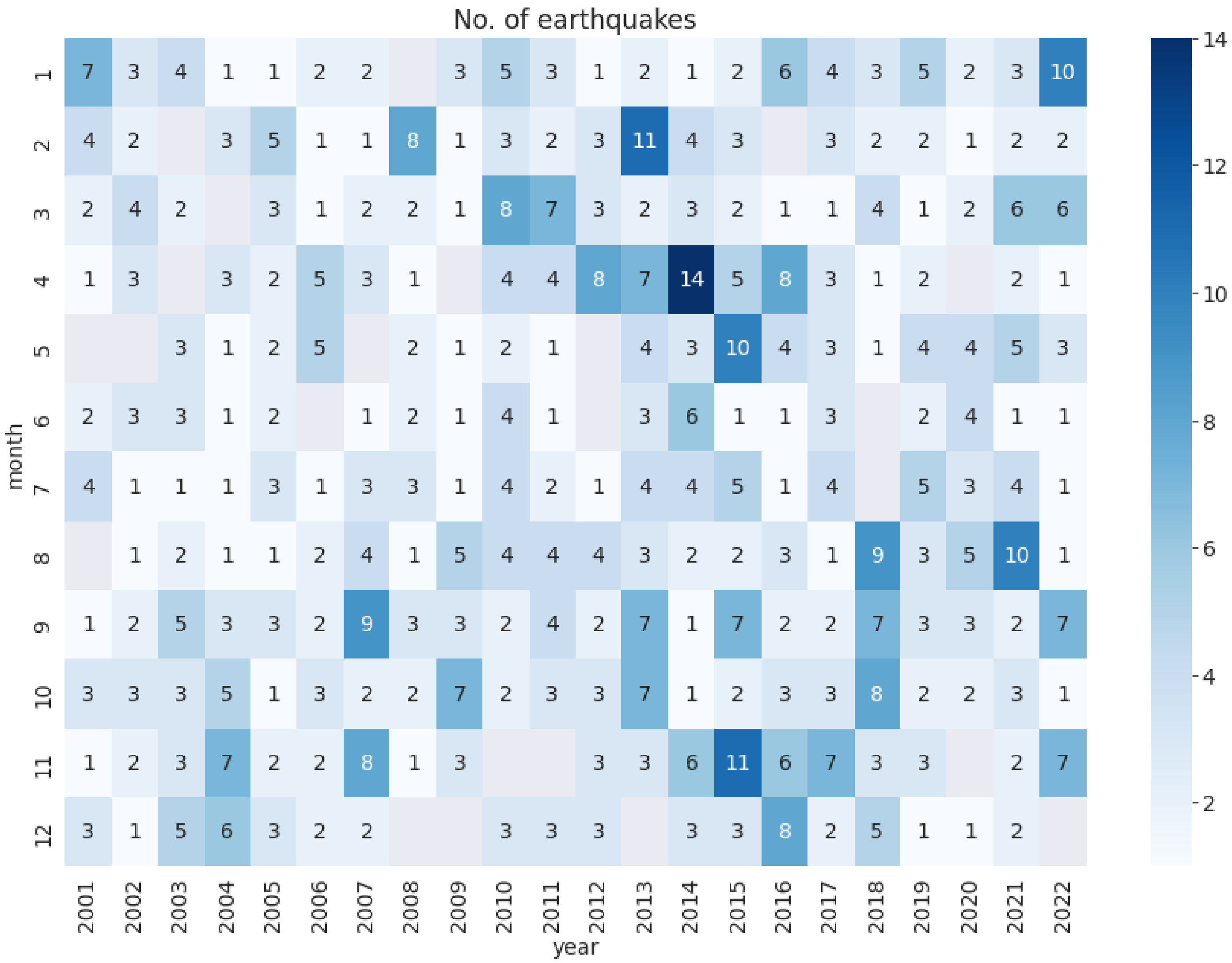
Also we can visualize these data in a heat map to get a clear idea.

```
yearmonth_df = analysis_df.groupby(['month', 'year']).size().unstack()
yearmonth_df
```

year	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	...	2013	2014	2015	2016	2017	201
month																	
1	7.0	3.0	4.0	1.0	1.0	2.0	2.0	NaN	3.0	5.0	...	2.0	1.0	2.0	6.0	4.0	3.
2	4.0	2.0	NaN	3.0	5.0	1.0	1.0	8.0	1.0	3.0	...	11.0	4.0	3.0	NaN	3.0	2.
3	2.0	4.0	2.0	NaN	3.0	1.0	2.0	2.0	1.0	8.0	...	2.0	3.0	2.0	1.0	1.0	4.
4	1.0	3.0	NaN	3.0	2.0	5.0	3.0	1.0	NaN	4.0	...	7.0	14.0	5.0	8.0	3.0	1.
5	NaN	NaN	3.0	1.0	2.0	5.0	NaN	2.0	1.0	2.0	...	4.0	3.0	10.0	4.0	3.0	1.
6	2.0	3.0	3.0	1.0	2.0	NaN	1.0	2.0	1.0	4.0	...	3.0	6.0	1.0	1.0	3.0	Na
7	4.0	1.0	1.0	1.0	3.0	1.0	3.0	3.0	1.0	4.0	...	4.0	4.0	5.0	1.0	4.0	Na
8	NaN	1.0	2.0	1.0	1.0	2.0	4.0	1.0	5.0	4.0	...	3.0	2.0	2.0	3.0	1.0	9.
9	1.0	2.0	5.0	3.0	3.0	2.0	9.0	3.0	3.0	2.0	...	7.0	1.0	7.0	2.0	2.0	7.
10	3.0	3.0	3.0	5.0	1.0	3.0	2.0	2.0	7.0	2.0	...	7.0	1.0	2.0	3.0	3.0	8.
11	1.0	2.0	3.0	7.0	2.0	2.0	8.0	1.0	3.0	NaN	...	3.0	6.0	11.0	6.0	7.0	3.
12	3.0	1.0	5.0	6.0	3.0	2.0	2.0	NaN	NaN	3.0	...	NaN	3.0	3.0	8.0	2.0	5.

12 rows × 22 columns

```
plt.title("No. of earthquakes")
sns.heatmap(yearmonth_df,annot=True, cmap='Blues');
```



After analyzing the heat map of earthquake occurrences, it becomes evident that there were two distinct periods of heightened activity. The first period was in October of 2018, during which a large number of earthquakes were recorded. The second period occurred in November of 2022, where another significant increase in earthquake activity was observed. By paying close attention to earthquake patterns and staying informed about updates from local authorities, individuals living in areas prone to earthquakes can take necessary precautions to ensure their safety and preparedness in the event of future seismic activity.

In addition to the information provided by the heat map, it is also important to consider the magnitude and location of the earthquakes. Earthquakes of higher magnitudes can cause more severe damage, and some areas may be more vulnerable to earthquakes than others due to factors such as local geology and population density.

It is important for individuals and communities in areas prone to earthquakes to be prepared with emergency supplies, evacuation plans, and knowledge of local warning systems. They should also be aware of the potential for tsunamis in coastal areas following a major earthquake.

Governments and organizations can also play a role in earthquake preparedness by implementing building codes and regulations, investing in early warning systems, and providing resources for disaster response and recovery.

Let us save and upload our work to Jovian before continuing.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "sithumini2400/analysis-of-world-earthquake-data" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/sithumini2400/analysis-of-world-earthquake-data
```

```
'https://jovian.com/sithumini2400/analysis-of-world-earthquake-data'
```

Alert

Now let's classify our earthquake data set according to their alert level. These alert levels are shown under four main colors: "green", "yellow", "orange", and "red".

The "alert" column in earthquake data represents the alert level assigned to an earthquake event by seismic monitoring agencies. The alert levels range from "green" to "red," indicating the potential impact of the earthquake on the affected area. "Green" indicates a low risk of damage or injury, while "red" indicates a high risk of significant damage, injuries, or loss of life. However, it is important to note that actual impacts can vary depending on local conditions, so it is crucial to follow official guidance and emergency procedures in the event of an earthquake, regardless of the assigned alert level.

First, we should get the value counts of the dataset from its alert. To do this, we can use .value_counts() function.

```
alert_df = analysis_df["alert"].value_counts()
```

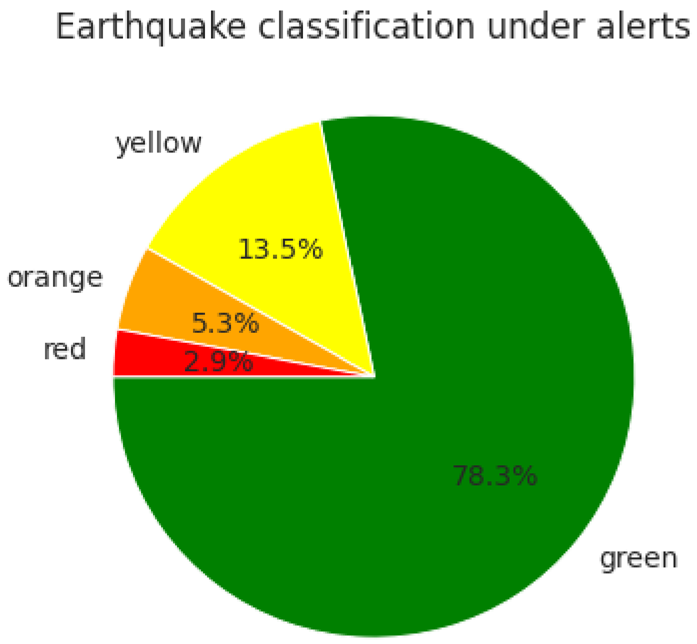
```
alert_df
```

```
alert
green     325
yellow     56
orange     22
red        12
Name: count, dtype: int64
```

Now we can plot this information in a pie chart and it is important to get a clear idea. To do this, we use the Matplotlib library.

```
#Create a pie chart using matplotlib
plt.figure(figsize=(12,6))

#Set the plot title and axis labels
plt.title("Earthquake classification under alerts")
colors = ["Green", "Yellow", "Orange", "Red"]
plt.pie(alert_df, labels=alert_df.index, autopct='%1.1f%%', startangle=180, colors = co
```



Upon examining the data presented, it is evident that most of the earthquakes analyzed had a "green" alert level, indicating that they were relatively minor with a low potential for significant damage, injuries, or loss of life. Conversely, the fewest number of earthquakes had a "red" alert level, signifying that they had a higher potential for significant impact. This observation could suggest that the seismic monitoring agencies responsible for assigning alert levels are effectively identifying earthquakes with varying levels of potential impact.

However, it is crucial to note that the potential impacts of an earthquake can vary depending on local conditions, such as population density, building construction, and proximity to critical infrastructure. Therefore, even minor earthquakes can cause significant damage or injuries under certain circumstances, emphasizing the importance of following official guidance and emergency procedures in the event of an earthquake, regardless of the assigned alert level. Additionally, assigning alert levels to earthquakes is a complex and dynamic process that involves considering multiple factors, including magnitude, depth, and location, among others. Therefore, it is necessary to continue refining and improving these systems to ensure effective and timely responses to earthquake events.

Let us save and upload our work to Jovian before continuing.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "sithumini2400/analysis-of-world-earthquake-data" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/sithumini2400/analysis-of-world-earthquake-data
```

```
'https://jovian.com/sithumini2400/analysis-of-world-earthquake-data'
```

Tsunami

The "tsunami" column of this dataframe contains binary values of either 0 or 1.

The value "1" indicates that the earthquake occurred in an oceanic region and has the potential to generate a tsunami, while the value "0" indicates that the earthquake occurred elsewhere and does not pose a tsunami threat.

Now I will be conducting an analysis to determine whether the earthquakes that have taken place during a specific time period in the past have caused tsunamis or not.

So for that first, I need to take the tsunami column and get value counts for 0 and 1. To do this, we use .value_counts() function.

```
tsunami_df = analysis_df["tsunami"].value_counts()
```

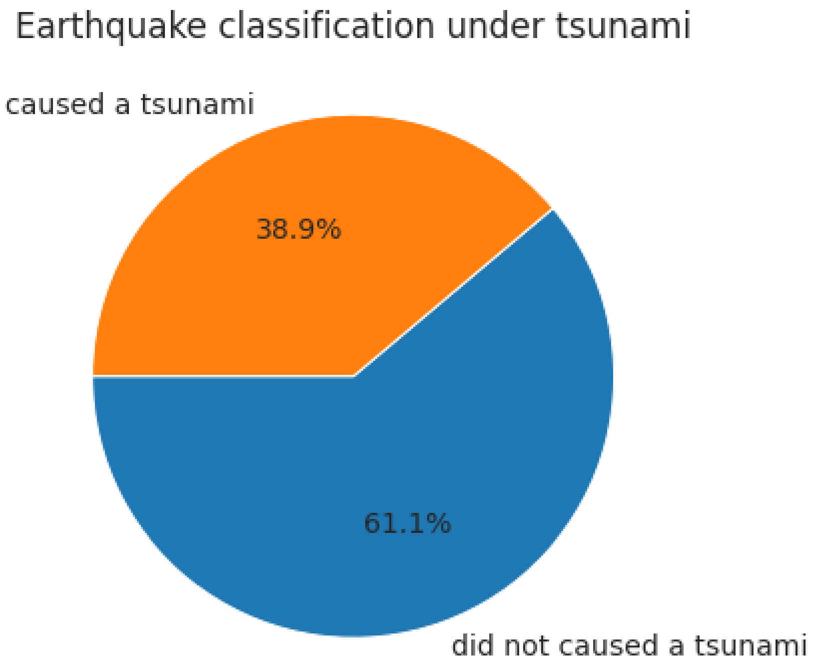
```
tsunami_df
```

```
tsunami
0    478
1    304
Name: count, dtype: int64
```

In our analysis, we observed that out of the 782 earthquakes that occurred during the past time period, 304 resulted in tsunamis, while 478 did not.

These findings can be represented using a pie chart that provides a clear comparison of the proportion of earthquakes that caused tsunamis and those that did not. By visually examining this chart, we can draw meaningful conclusions about the likelihood of a tsunami occurring in the event of an earthquake, which is a crucial factor in planning for and mitigating the potential impacts of such disasters.

```
#Create a pie chart using matplotlib
plt.figure(figsize=(12,6))
plt.title("Earthquake classification under tsunami")
plt.pie(tsunami_df, labels=["did not caused a tsunami","caused a tsunami"], autopct='%1
```



Based on our analysis, we can see that out of all the earthquakes that occurred during the specified time period, 304 of them resulted in tsunamis, while 478 did not. This translates to 38.9% of earthquakes causing tsunamis and 61.1% not causing tsunamis. It is interesting to note that a significant proportion of earthquakes did not lead to tsunamis, indicating that tsunamis are not an inevitable consequence of earthquakes.

Nonetheless, it is crucial to prepare and implement appropriate measures to mitigate the risk of tsunamis, especially in regions prone to such events.

Let us save and upload our work to Jovian before continuing.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "sithumini2400/analysis-of-world-earthquake-data" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/sithumini2400/analysis-of-world-earthquake-data
```

```
'https://jovian.com/sithumini2400/analysis-of-world-earthquake-data'
```

Asking and Answering Questions

Q : What are the areas where earthquakes have caused serious damage during the last time period?

To answer this question, we need to know what the international earthquake magnitude scale is that measures the damage caused by an earthquake.

Here is the international standard earthquake magnitude scale.

- 5.5 to 6.0: Slight damage to buildings and other structures.
- 6.1 to 6.9: May cause a lot of damage in very populated areas.
- 7.0 to 7.9: Major earthquake Serious damage.
- 8.0 or greater: a great earthquake. Can totally destroy communities near the epicenter.

So now let's separate the records related to earthquakes with magnitudes greater than 7.0.

```
highest_mag = analysis_df[analysis_df["magnitude"] >= 7.0]
```

```
highest_mag
```

	title	magnitude	date_time	alert	tsunami	sig	nst	depth	latitude	longitude	location	conti
0	M 7.0 - 18 km SW of Malango, Solomon Islands	7.0	2022-11-22 02:03:00	green	1	768	117	14.000	-9.7963	159.596	Malango, Solomon Islands	Oce
2	M 7.0 -	7.0	2022-11-12 07:09:00	green	1	755	147	579.000	-20.0508	-178.346	NaN	Oce

	title	magnitude	date_time	alert	tsunami	sig	nst	depth	latitude	longitude	location	conti
3	M 7.3 - 205 km ESE of Neiafu, Tonga	7.3	2022-11-11 10:48:00	green	1	833	149	37.000	-19.2918	-172.129	Neiafu, Tonga	
5	M 7.0 - south of the Fiji Islands	7.0	2022-11-09 09:51:00	green	1	755	142	660.000	-26.0442	178.381	the Fiji Islands	
9	M 7.6 - 35 km SSW of Aguililla, Mexico	7.6	2022-09-19 18:05:00	yellow	1	1799	271	26.943	18.3667	-103.252	Aguililla, Mexico	N Ame
...	...	...	...	...	...	...	...	...	...	...	...	
773	M 7.4 - 102 km SSE of Bengkulu, Indonesia	7.4	2001-02-13 19:28:00	NaN	0	842	221	36.000	-4.6800	102.562	Bengkulu, Indonesia	
775	M 7.7 - 17 km NW of Bhach?u, India	7.7	2001-01-26 03:16:00	NaN	0	912	472	16.000	23.4190	70.232	Bhach?u, India	
777	M 7.7 - 28 km SSW of Puerto El Triunfo, El Sal...	7.7	2001-01-13 17:33:00	NaN	0	912	427	60.000	13.0490	-88.660	Puerto El Triunfo, El Salvador	
779	M 7.1 - 16 km NE of Port-Olry, Vanuatu	7.1	2001-01-09 16:49:00	NaN	0	776	372	103.000	-14.9280	167.170	Port-Olry, Vanuatu	
781	M 7.5 - 21 km SE of Lukatan, Philippines	7.5	2001-01-01 06:57:00	NaN	0	865	324	33.000	6.8980	126.579	Lukatan, Philippines	

283 rows × 15 columns

Now we can use the information we have gathered about the earthquakes with the highest magnitudes and plot them on a map to identify their geographical locations. This will help us determine where the earthquakes with the most significant damage occurred during the past time period.

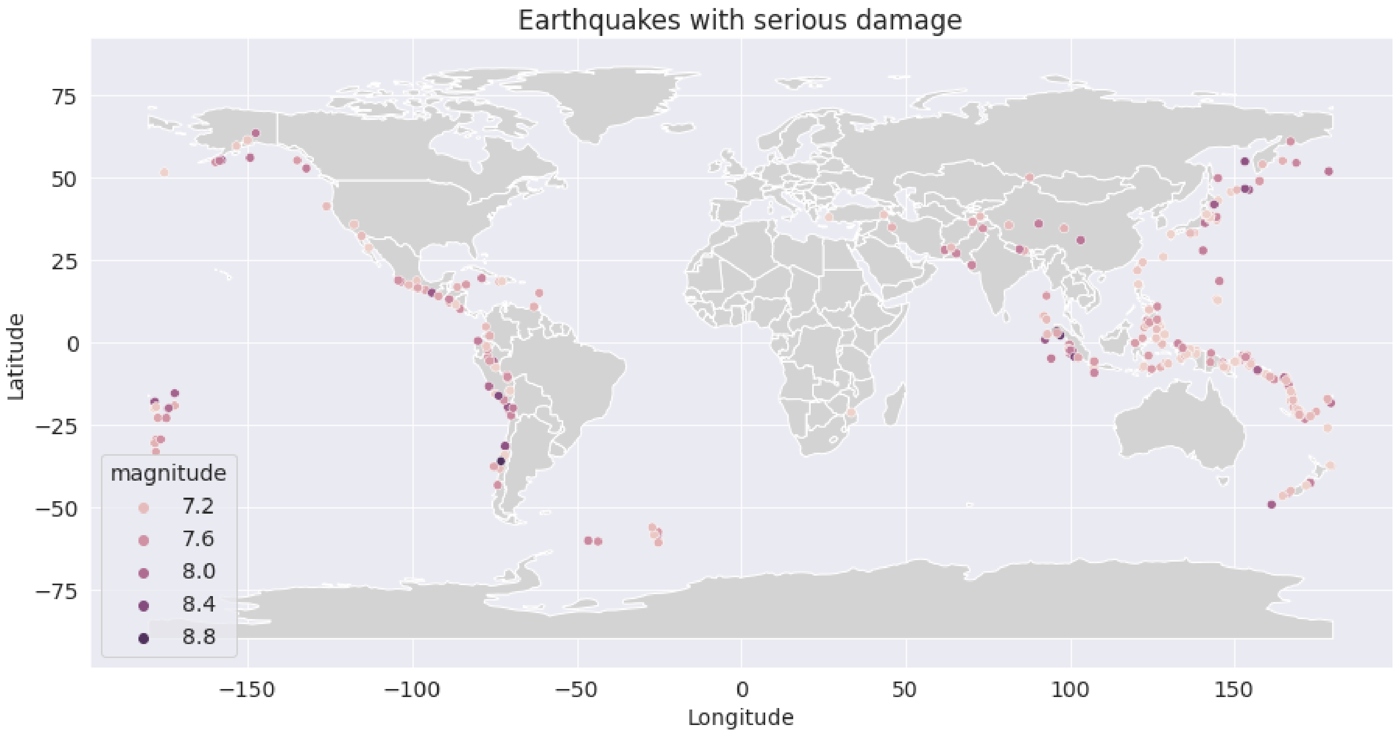
To do that, we use the Matplotlib, Seaborn and Geopandas libraries.

```
#Create a world map using geopandas
countries = gpd.read_file(
     gpd.datasets.get_path("naturalearth_lowres"))
countries.plot(color="lightgrey")

# Create a scatterplot of earthquake locations using seaborn
sns.scatterplot(x="longitude", y="latitude", hue = "magnitude", data=highest_mag)

# Set the title and axis labels
plt.title("Earthquakes with serious damage")
plt.xlabel("Longitude")
plt.ylabel("Latitude");
```

```
/tmp/ipykernel_117/3006795142.py:3: FutureWarning: The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from https://www.naturalearthdata.com/downloads/110m-cultural-vectors/.
  gpd.datasets.get_path("naturalearth_lowres"))
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning: is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



According to our analysis, it appears that a significant number of earthquakes with serious damage occurred in Asian and South American countries during the past time period. This suggests that these regions may be particularly vulnerable to earthquakes and their potential impacts. It is important for these countries to have effective emergency response plans in place, as well as strong building codes and infrastructure designed to withstand earthquakes.

In addition, this information can be used to inform preparedness efforts for earthquakes in other regions that may also be vulnerable. By studying the characteristics of earthquakes that have caused serious damage in the past, scientists and engineers can develop models to better understand the behavior of earthquakes and their potential impacts. This information can be used to develop more effective earthquake prediction and early warning systems, as well as to design more resilient infrastructure and buildings that can withstand earthquakes.

Let us save and upload our work to Jovian before continuing.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "sithumini2400/analysis-of-world-earthquake-data" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/sithumini2400/analysis-of-world-earthquake-data
```

```
'https://jovian.com/sithumini2400/analysis-of-world-earthquake-data'
```

Q : How did the significance of earthquakes change over time?

To answer this question, we need to analyze our earthquake dataset by grouping the data based on the year of the earthquake's occurrence. This can be done using the "groupby()" function in Python. Once the data is grouped by year, we can calculate the mean value of the "sig" number column for each year. This can be done using the "mean()" function in Python.

This analysis will help us understand the trend of earthquake occurrences over the years and whether there has been any significant increase or decrease in the magnitude of earthquakes over time.

```
yearly_df = analysis_df.groupby(pd.Grouper(key='date_time',freq = "Y"))['sig'].mean()
```

```
yearly_df
```

```
date_time
2001-12-31     793.035714
2002-12-31     766.920000
2003-12-31     756.548387
2004-12-31     757.312500
2005-12-31     812.285714
2006-12-31     810.653846
2007-12-31     840.270270
2008-12-31     824.720000
2009-12-31     942.884615
2010-12-31     939.073171
2011-12-31     938.176471
2012-12-31    1032.322581
2013-12-31     829.735849
2014-12-31     773.000000
2015-12-31     847.943396
2016-12-31     935.069767
2017-12-31     961.111111
2018-12-31     947.720930
2019-12-31     870.848485
2020-12-31     941.074074
2021-12-31     927.833333
```

```
2022-12-31    855.425000
Freq: A-DEC, Name: sig, dtype: float64
```

Based on the information provided, it seems challenging to obtain a precise understanding of the significance of the data. Therefore, in order to better comprehend and analyze the information, it would be beneficial to present it visually using a bar plot.

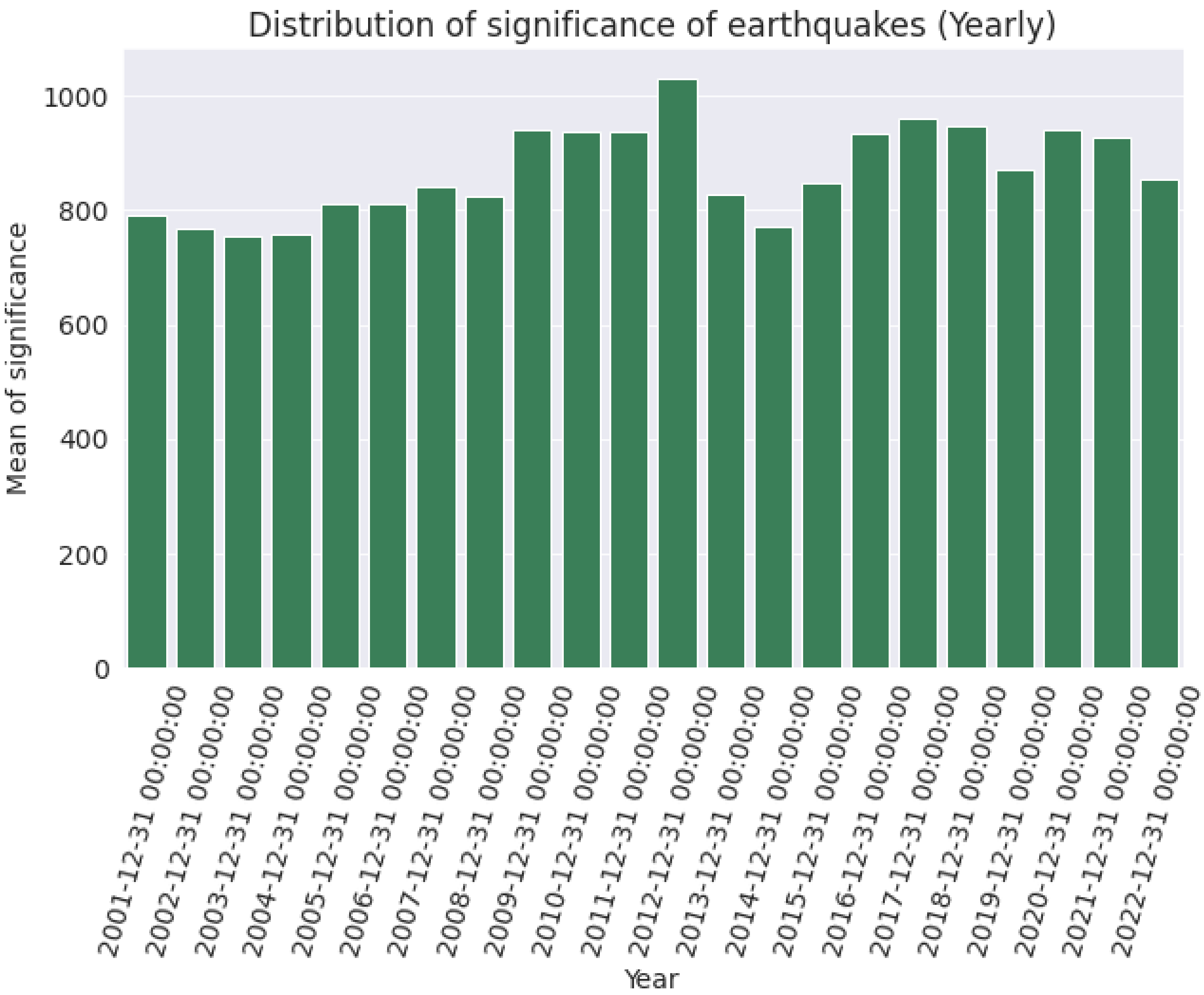
This will enable us to more easily interpret the data and draw meaningful conclusions from it.

```python
#Plot the number of earthquakes by year using Seaborn and Matplotlib
plt.figure(figsize=(10, 6))
sns.barplot(x = yearly_df.index, y = yearly_df, color = "seagreen")

plt.xticks(rotation = 75)

#Set the plot title and axis labels
plt.title('Distribution of significance of earthquakes (Yearly)')
plt.xlabel('Year')
plt.ylabel('Mean of significance');
```

```
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



By studying above bar chart we can see that the highest mean of significance is in 2012 and there is a small incresement in mean of significance during recent years.

some possible reasons could be an increase in the number of seismometers, more accurate measurements of earthquakes, or changes in the earth's crust.

As for solutions to reduce this, for example, if the increase is due to a higher number of seismometers, one solution could be to improve the accuracy of the measurements or to spread the seismometers out to cover a larger area. If the increase is due to changes in the earth's crust, it may be more difficult to find a solution.

Let us save and upload our work to Jovian before continuing.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "sithumini2400/analysis-of-world-earthquake-data" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/sithumini2400/analysis-of-world-earthquake-data
```

```
'https://jovian.com/sithumini2400/analysis-of-world-earthquake-data'
```

Q : What is the relationship between the earthquake alert and its magnitude?

To provide an answer to the question at hand, the initial step is to acquire the number of alerts for each magnitude. This is achieved through the application of two functions, .groupby() and .unstack(). The former function is utilized to group the data by the magnitude and alert columns, while the latter function is used to transform the grouped data into a more readable and understandable format.

Through the use of these functions, we can easily visualize the relationship between the alert and magnitude columns.

```
grouped_df = analysis_df.groupby(['magnitude', 'alert']).size().unstack()
```

```
grouped_df
```

alert	green	orange	red	yellow
magnitude				
6.5	67.0	1.0	NaN	10.0
6.6	51.0	3.0	2.0	7.0
6.7	41.0	3.0	NaN	2.0
6.8	35.0	1.0	NaN	5.0
6.9	43.0	2.0	NaN	7.0
7.0	20.0	1.0	2.0	3.0
7.1	17.0	3.0	NaN	2.0
7.2	7.0	NaN	2.0	2.0
7.3	11.0	2.0	2.0	2.0
7.4	4.0	1.0	NaN	1.0
7.5	7.0	1.0	1.0	3.0
7.6	4.0	NaN	NaN	4.0
7.7	4.0	NaN	1.0	3.0
7.8	4.0	2.0	1.0	NaN
7.9	3.0	NaN	NaN	2.0
8.0	1.0	1.0	NaN	NaN
8.1	2.0	NaN	NaN	NaN
8.2	3.0	NaN	1.0	2.0
8.3	1.0	1.0	NaN	NaN
8.6	NaN	NaN	NaN	1.0

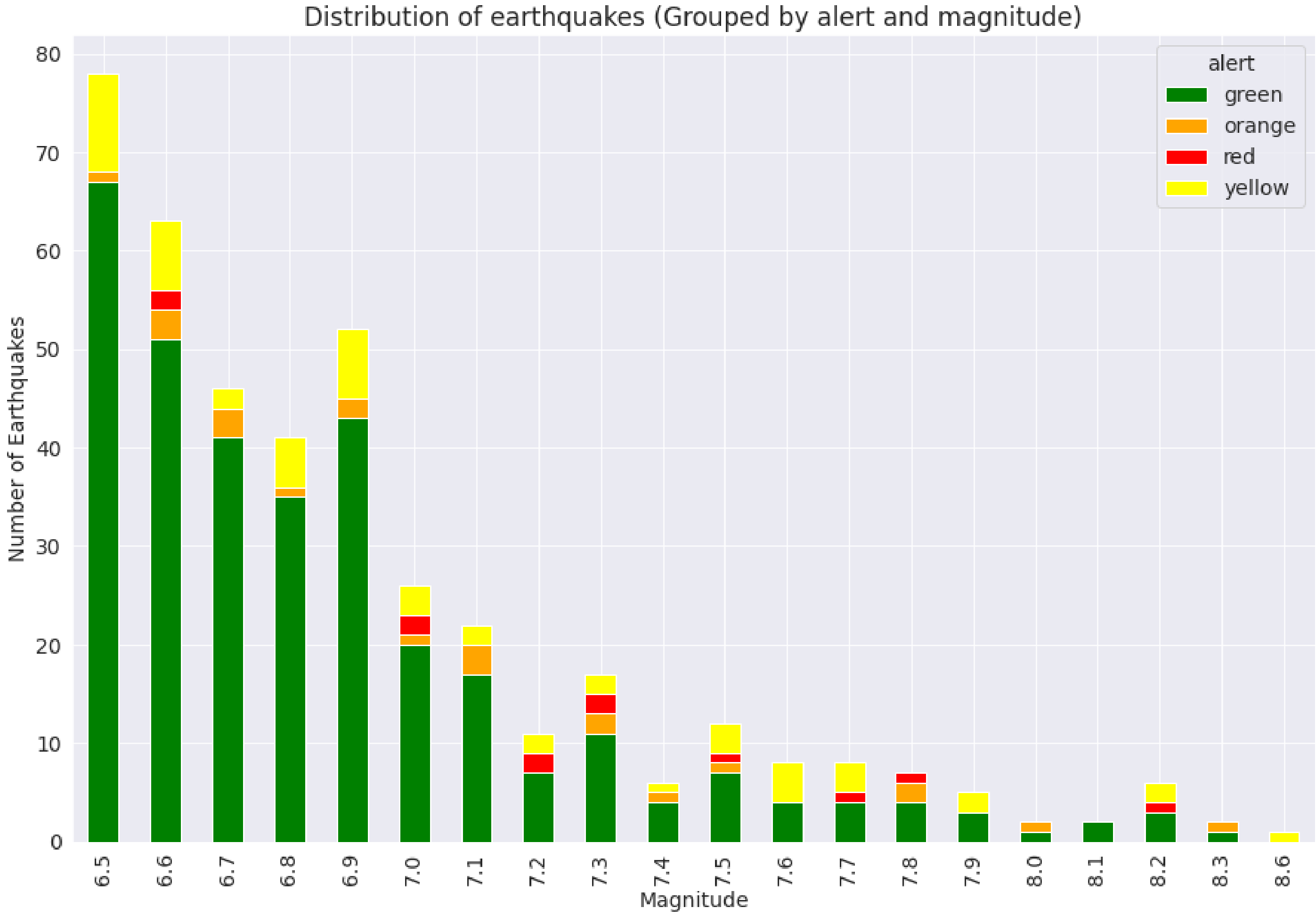
Now we can create a stacked bar chart to visualize the number of alerts for each magnitude. A stacked bar chart is a chart type that shows how different categories contribute to the whole.

In this case, each bar represents a magnitude and the different segments within the bar represent the number of alerts for each color.

By using this stacked bar chart, we can easily see the contribution of each alert color to the total number of alerts for each magnitude.

```
#Create a stacked bar chart using Matplotlib
grouped_df.plot(kind='bar', stacked=True, color = ["Green", "Orange","Red","Yellow"])

#Set the plot title and axis labels
plt.title("Distribution of earthquakes (Grouped by alert and magnitude)")
plt.xlabel("Magnitude")
plt.ylabel("Number of Earthquakes");
```



As mentioned earlier, earthquakes with a magnitude of 7.0 and above can cause significant damage. However, upon examining the stacked bar chart, it becomes apparent that there were green alerts issued for some earthquakes of higher magnitudes. This suggests that the alert system may have some limitations and is not always reliable in predicting the potential damage caused by an earthquake.

Sometimes, even if the magnitude of the earthquake is close to 8, there may be cases where the impact is not very damaging. The reason for this may be that the earthquake occurred in an unpopulated area.

So if the current alert system is not accurately identifying potentially damaging earthquakes, then it may need to be updated or improved.

Another solution could be to increase the number and sensitivity of earthquake monitoring stations to improve the accuracy of earthquake measurements. Additionally, it may be helpful to improve public education and awareness about earthquake safety measures, such as proper building construction and emergency preparedness, to mitigate the impact of potentially damaging earthquakes.

Let us save and upload our work to Jovian before continuing.

```python
import jovian
```

```python
jovian.commit()
```

[jovian] Updating notebook "sithumini2400/analysis-of-world-earthquake-data" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/sithumini2400/analysis-of-world-earthquake-data

'https://jovian.com/sithumini2400/analysis-of-world-earthquake-data'

Q : What are the areas that are more susceptible to tsunamis caused by earthquakes?

To respond to this question, we must initially extract the list of locations where tsunamis have occurred from our dataset.

```python
actual_tsunami = analysis_df[analysis_df["tsunami"]==1]
actual_tsunami
```

	title	magnitude	date_time	alert	tsunami	sig	nst	depth	latitude	longitude	location	conti
0	M 7.0 - 18 km SW of Malango, Solomon Islands	7.0	2022-11-22 02:03:00	green	1	768	117	14.000	-9.7963	159.596	Malango, Solomon Islands	Oce
2	M 7.0 -	7.0	2022-11-12 07:09:00	green	1	755	147	579.000	-20.0508	-178.346	NaN	Oce
3	M 7.3 - 205 km ESE of Neiafu, Tonga	7.3	2022-11-11 10:48:00	green	1	833	149	37.000	-19.2918	-172.129	Neiafu, Tonga	
4	M 6.6 -	6.6	2022-11-09 10:14:00	green	1	670	131	624.464	-25.5948	178.278	NaN	
5	M 7.0 - south of the Fiji Islands	7.0	2022-11-09 09:51:00	green	1	755	142	660.000	-26.0442	178.381	the Fiji Islands	
...	...	...	...	...	...	...	...	...	...	...	...	
408	M 6.9 - 2 km NNE of Yacuanquer, Colombia	6.9	2013-02-09 14:16:00	green	1	904	562	145.000	1.1350	-77.393	Yacuanquer, Colombia	S Am
409	M 7.1 - 32 km SE of Lata, Solomon Islands	7.1	2013-02-08 15:26:00	green	1	780	334	21.000	-10.9280	166.018	Lata, Solomon Islands	
410	M 6.8 - 22 km ESE of Lata, Solomon Islands	6.8	2013-02-08 11:12:00	green	1	711	400	12.000	-10.8380	165.969	Lata, Solomon Islands	

	title	magnitude	date_time	alert	tsunami	sig	nst	depth	latitude	longitude	location	conti
411	M 6.7 - 33 km SSW of Lata, Solomon Islands	6.7	2013-02-07 18:59:00	green	1	691	387	11.000	-10.9970	165.655	Lata, Solomon Islands	
414	M 8.0 - 75 km W of Lata, Solomon Islands	8.0	2013-02-06 01:12:00	green	1	993	460	24.000	-10.7990	165.114	Lata, Solomon Islands	

304 rows × 15 columns

Now we can get the count of how many times each location appears in the subset using the value_counts() function. This will give us an idea of which areas have experienced the most tsunamis due to earthquakes.

```
tsunami_countdf = actual_tsunami["location"].value_counts()
tsunami_countdf
```

```
location
Panguna, Papua New Guinea        9
South Sandwich Islands region    9
Tadine, New Caledonia            9
Lata, Solomon Islands            8
Kirakira, Solomon Islands        8
                                ..
Luwuk, Indonesia                 1
Palora, Ecuador                  1
Molucca Sea                      1
Point MacKenzie, Alaska          1
L'Esperance Rock, New Zealand    1
Name: count, Length: 173, dtype: int64
```

Now we get the list of locations where four or more tsunamis have occurred and then we can consider locations where more susceptible to tsunamis due to earthquakes easily.

```
tsunami_locdf =  tsunami_countdf[tsunami_countdf>=4]
tsunami_locdf
```

```
location
Panguna, Papua New Guinea        9
South Sandwich Islands region    9
Tadine, New Caledonia            9
Lata, Solomon Islands            8
Kirakira, Solomon Islands        8
Kokopo, Papua New Guinea         8
Nikolski, Alaska                 6
Kermadec Islands, New Zealand    6
Perryville, Alaska               5
Port-Olry, Vanuatu               5
the Fiji Islands                 4
```

```
Sola, Vanuatu                4
Levuka, Fiji                 4
Namie, Japan                 4
Name: count, dtype: int64
```

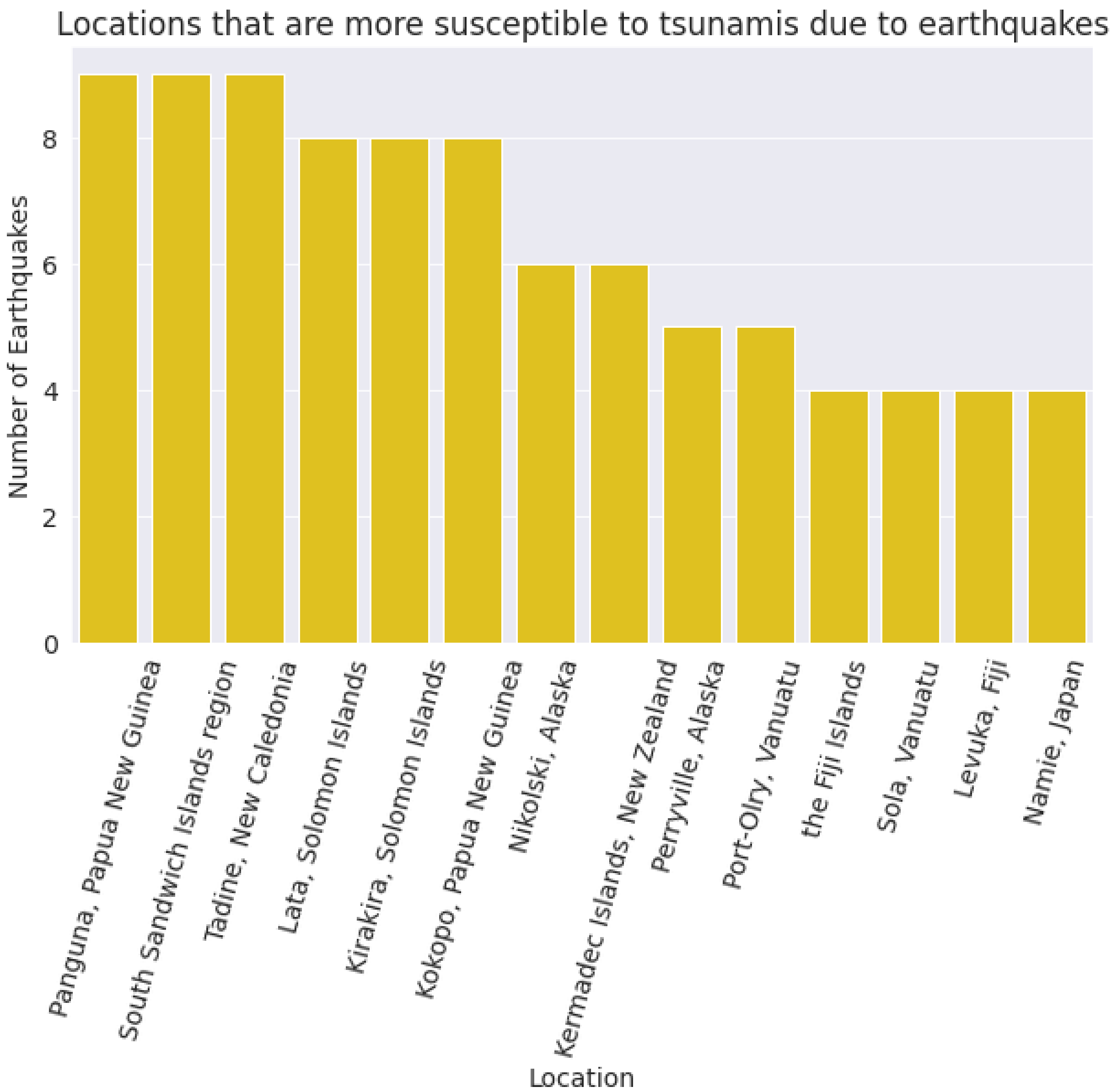
Now we can plot this information in a bar plot and get an idea of which locations are more susceptible to tsunamis due to earthquakes.

```
#Plot the number of locations that are more susceptible to tsunamis due to earthquakes
plt.figure(figsize=(10, 6))
sns.barplot(x = tsunami_locdf.index, y = tsunami_locdf, color = "gold")

plt.xticks(rotation = 75)

#Set the plot title and axis labels
plt.title('Locations that are more susceptible to tsunamis due to earthquakes')
plt.xlabel('Location')
plt.ylabel('Number of Earthquakes');
```

```
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
/opt/conda/lib/python3.9/site-packages/seaborn/_core.py:1225: FutureWarning:
is_categorical_dtype is deprecated and will be removed in a future version. Use
isinstance(dtype, CategoricalDtype) instead
  if pd.api.types.is_categorical_dtype(vector):
```



Based on the analysis bar chart, it appears that certain locations are more susceptible to tsunamis due to earthquakes. These locations include "Panguna, Papua New Guinea", "Tadine, New Caledonia", "South Sandwich Islands region", "Kokopo, Papua New Guinea", "Kirakira, Solomon Islands", and "Lata, Solomon Islands".

For people living in these areas, it is important to be aware of the potential risks associated with tsunamis. It is recommended that individuals create an emergency plan that includes evacuation procedures and the location of safe zones in the event of a tsunami. In addition, it is important to monitor weather and geological reports for any potential warnings or alerts. Being prepared and informed can greatly increase the chances of survival in the event of a tsunami.

Furthermore, individuals living in areas prone to tsunamis should consider participating in community-wide emergency drills and exercises to ensure that everyone knows what to do in the event of an emergency. It is also important to have emergency supplies on hand, such as non-perishable food, water, and first aid kits.

Overall, while living in an area prone to tsunamis due to earthquakes can be risky, being prepared and informed can greatly increase the chances of survival in the event of an emergency.

Let us save and upload our work to Jovian before continuing.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "sithumini2400/analysis-of-world-earthquake-data" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/sithumini2400/analysis-of-world-earthquake-data
```

```
'https://jovian.com/sithumini2400/analysis-of-world-earthquake-data'
```

Inferences and Conclusion

By studying our analysis on earthquake dataset we can make following conclusions.

- The susceptibility of Asian and South American countries to earthquakes and their potential impact. They have marked on following map.

  Earthquakes%20with%20serious%20damage.png

  In conclusion, the analysis of earthquake data has revealed the vulnerability of Asian and South American countries to earthquakes and their potential impact. The implementation of effective emergency response plans, building codes, and resilient infrastructure can significantly reduce the loss of life and property in these regions. Furthermore, the study of past earthquakes can guide preparedness efforts in other vulnerable regions by providing insights into earthquake behavior and its impact. This knowledge can be used to develop models that improve earthquake prediction and early warning systems, design more resilient infrastructure and buildings capable of withstanding earthquakes, and minimize the loss of life and property.

- Analyzing the Mean Significance of Earthquakes Over Time: Possible Reasons and Solutions for Increase in Recent Years

  In conclusion, the bar chart analysis shows that the mean significance of earthquakes has slightly increased in recent years, possibly due to an increase in the number of seismometers, more accurate measurements of earthquakes, or changes in the earth's crust. Possible solutions to reduce this include improving the accuracy of measurements or spreading seismometers out to cover a larger area. However, if the increase is due to changes in the earth's crust, finding a solution may be more difficult. It is important to continue monitoring and studying earthquakes to better understand their behavior and potential impacts, and to develop more effective earthquake prediction and early warning systems.

- The Limitations of Earthquake Alert Systems and Possible Solutions.

  In conclusion, while earthquakes with a magnitude of 7.0 and above can cause significant damage, the alert system used to predict potential damage may have limitations. It is important to note that the impact of an earthquake is not always related to its magnitude, and sometimes, even high magnitude earthquakes may not cause significant damage. Upgrading or improving the current alert system, increasing the number and sensitivity of earthquake monitoring stations, and improving public education and awareness about earthquake safety measures are some ways to mitigate the impact of potentially hazardous earthquakes. By taking proactive measures to improve earthquake preparedness, the risk of loss of life and property damage can be greatly reduced.

- Preparing for Tsunamis in Earthquake-Prone Areas

  In conclusion, it is clear from the analysis of the bar chart that some regions are more vulnerable to tsunamis caused by earthquakes. For individuals living in these areas, taking necessary precautions such as creating an emergency plan, being aware of evacuation procedures and safe zones, monitoring weather and geological reports, participating in community-wide emergency drills, and having emergency supplies on hand can greatly increase the chances of survival in the event of a tsunami. While the risks of living in these areas cannot be eliminated entirely, being prepared and informed can make all the difference when it comes to surviving a natural disaster.

Let us save and upload our work to Jovian before continuing.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "sithumini2400/analysis-of-world-earthquake-data" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/sithumini2400/analysis-of-world-earthquake-data
```

```
'https://jovian.com/sithumini2400/analysis-of-world-earthquake-data'
```

References and Future Work

Here are some potential future works that can be done based on the insights gained from our analysis:

- Conduct further research on the specific factors that contribute to the susceptibility of Asian and South American countries to earthquakes, such as geological characteristics, building codes, and infrastructure design. This can help inform more targeted and effective preparedness and mitigation efforts.
- Explore the potential for utilizing advanced technology, such as artificial intelligence and machine learning, to improve earthquake prediction and early warning systems. This can help reduce the loss of life and property damage caused by earthquakes.
- Conduct studies to identify the long-term impacts of earthquakes on affected communities, including economic, social, and psychological impacts. This can help inform more comprehensive disaster relief efforts.
- Evaluate the effectiveness of current emergency response plans and building codes in earthquake-prone areas, and identify areas for improvement. This can help ensure that communities are better prepared and more resilient in the face of natural disasters.
- Explore the potential for international cooperation and collaboration in disaster relief efforts, particularly in areas where the risk of earthquakes and other natural disasters is high. This can help facilitate the sharing of knowledge and resources, and ultimately improve outcomes for affected communities.

```
import jovian
```

```
jovian.commit()
```

```

```