

In [1]:

```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/indian-subcontinent-earthquake-data-2000-to-2024/Earthquakes.csv
```

In [2]:

```
df=pd.read_csv("/kaggle/input/indian-subcontinent-earthquake-data-2000-to-2024/Earthquakes.csv")
```

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 17805 entries, 0 to 17804
```

```
Data columns (total 22 columns):
```

#	Column	Non-Null Count	Dtype
0	time	17805 non-null	object
1	latitude	17805 non-null	float64
2	longitude	17805 non-null	float64
3	depth	17805 non-null	float64
4	mag	17805 non-null	float64
5	magType	17805 non-null	object
6	nst	13162 non-null	float64
7	gap	16839 non-null	float64
8	dmin	5903 non-null	float64
9	rms	17786 non-null	float64
10	net	17805 non-null	object
11	id	17805 non-null	object
12	updated	17805 non-null	object
13	place	17805 non-null	object
14	type	17805 non-null	object
15	horizontalError	5199 non-null	float64
16	depthError	8777 non-null	float64
17	magError	5852 non-null	float64
18	magNst	16068 non-null	float64
19	status	17805 non-null	object
20	locationSource	17805 non-null	object
21	magSource	17805 non-null	object

```
dtypes: float64(12), object(10)
```

```
memory usage: 3.0+ MB
```

```
In [4]: df.describe()
```

Out[4]:

	latitude	longitude	depth	mag	nst	gap	dmin
count	17805.000000	17805.000000	17805.000000	17805.000000	13162.000000	16839.000000	5903.000000
mean	16.731557	88.992976	30.561463	4.464157	47.685306	120.644789	3.866031
std	11.923713	9.761070	27.774405	0.473279	66.856708	55.824191	3.437627
min	1.793000	60.667800	0.600000	2.600000	4.000000	9.000000	0.027000
25%	5.372000	85.835400	10.000000	4.200000	15.000000	76.900000	1.631500
50%	12.504000	93.393000	29.720000	4.400000	25.000000	111.200000	3.109000
75%	28.854400	94.908000	35.000000	4.700000	49.000000	168.000000	5.073500
max	35.489000	100.895000	264.000000	9.100000	724.000000	337.800000	38.666000

```
In [5]: df.isnull().sum()
```

Out[5]:

time	0
latitude	0
longitude	0
depth	0
mag	0
magType	0
nst	4643
gap	966
dmin	11902
rms	19
net	0
id	0
updated	0
place	0
type	0
horizontalError	12606
depthError	9028
magError	11953
magNst	1737
status	0
locationSource	0
magSource	0
dtype:	int64

In [6]:



```

import matplotlib.pyplot as plt
import seaborn as sns
import folium
from folium.plugins import HeatMap
import geopandas as gpd

df["time"] = pd.to_datetime(df["time"], errors="coerce")
plt.figure(figsize=(12, 5))
df["time"].dt.year.value_counts().sort_index().plot(kind="line", marker="o", color="blue")
plt.title("Earthquake Frequency Over Years")
plt.xlabel("Year")
plt.ylabel("Number of Earthquakes")
plt.grid()
plt.show()

# ◆ Distribution of Magnitude & Depth
fig, axes = plt.subplots(1, 2, figsize=(14, 5))
sns.histplot(df["mag"], bins=30, kde=True, color="blue", ax=axes[0])
axes[0].set_title("Magnitude Distribution")
sns.histplot(df["depth"], bins=30, kde=True, color="red", ax=axes[1])
axes[1].set_title("Depth Distribution")
plt.show()

# ◆ Scatter: Magnitude vs. Depth
plt.figure(figsize=(10, 5))
sns.scatterplot(x=df["depth"], y=df["mag"], alpha=0.5, color="purple")
plt.xlabel("Depth (km)")
plt.ylabel("Magnitude")
plt.title("Depth vs. Magnitude of Earthquakes")
plt.grid()
plt.show()

# ◆ Correlation Matrix
plt.figure(figsize=(8, 6))
sns.heatmap(df[["depth", "mag", "rms", "horizontalError", "depthError", "magError"]].corr(),
            annot=True, cmap="coolwarm", linewidths=0.5)
plt.title("Correlation Matrix of Key Features")
plt.show()
map_center = [df["latitude"].mean(), df["longitude"].mean()]
# ◆ Heatmap of Earthquake Locations
heat_data = df[["latitude", "longitude"]].values.tolist()
folium_map = folium.Map(location=map_center, zoom_start=5)
HeatMap(heat_data).add_to(folium_map)
folium_map.save("earthquake_heatmap.html")

```

```
display(folium_map)
```

```
# ◆ Geospatial Analysis using GeoPandas
```

```
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))  
gdf = gpd.GeoDataFrame(df, geometry=gpd.points_from_xy(df.longitude, df.latitude))
```

```
# ◆ Plot Earthquake Points on a World Map
```

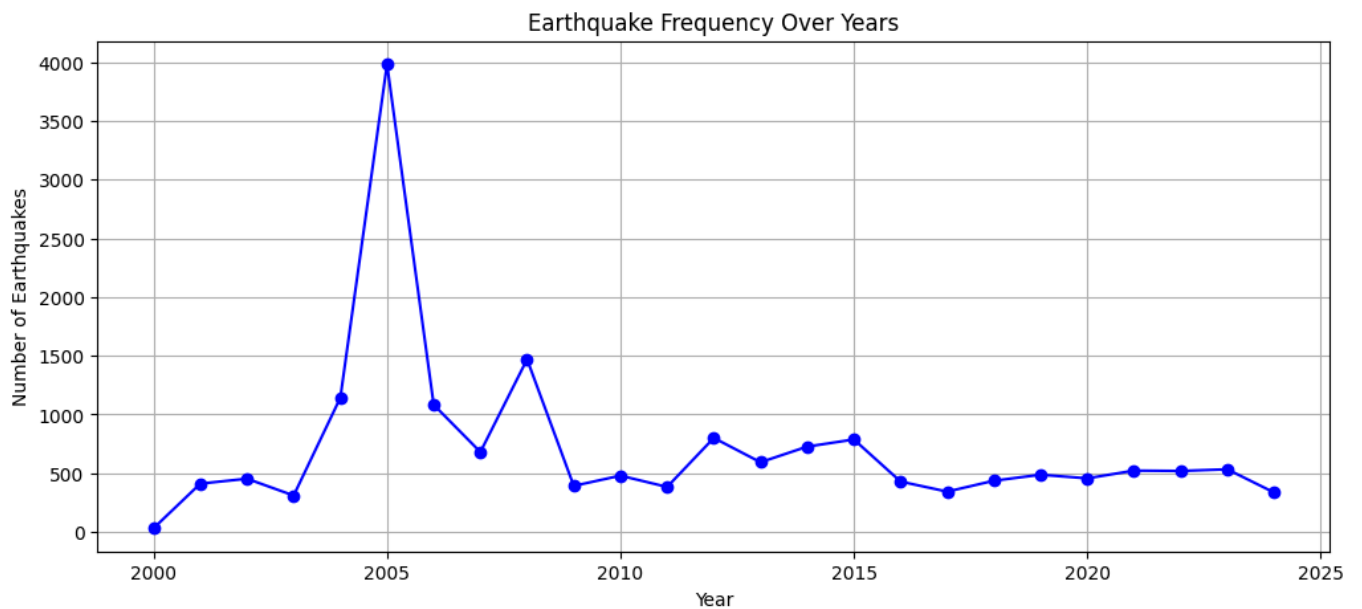
```
fig, ax = plt.subplots(figsize=(12, 8))  
world.plot(ax=ax, color="lightgrey")  
gdf.plot(ax=ax, markersize=df["mag"] * 5, color="red", alpha=0.5)  
plt.title("Earthquake Locations on World Map")  
plt.show()
```

```
# ◆ Earthquakes Per Country (Spatial Join)
```

```
gdf = gdf.set_crs("EPSG:4326") # Set correct CRS  
world = world.to_crs("EPSG:4326")  
joined = gpd.sjoin(gdf, world, how="left", predicate="within")  
country_counts = joined["name"].value_counts()
```

```
# ◆ Plot Top 10 Most Affected Countries
```

```
plt.figure(figsize=(10, 6))  
sns.barplot(x=country_counts[:10].values, y=country_counts[:10].index, palette="coolwarm")  
plt.xlabel("Number of Earthquakes")  
plt.ylabel("Country")  
plt.title("Top 10 Most Affected Countries by Earthquakes")  
plt.grid()  
plt.show()
```

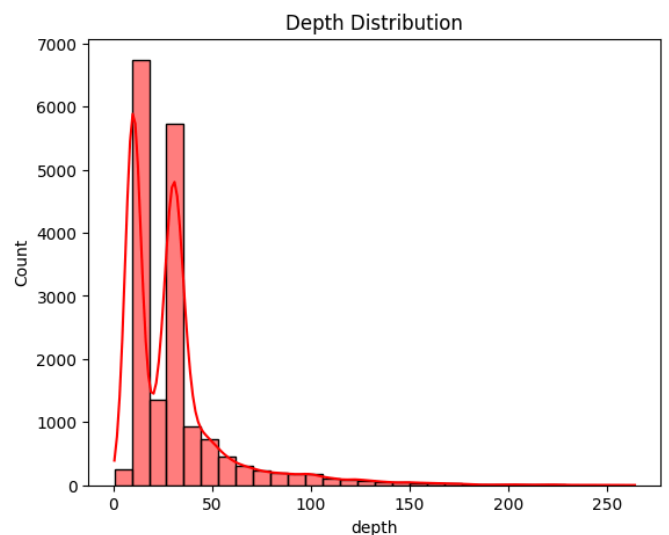
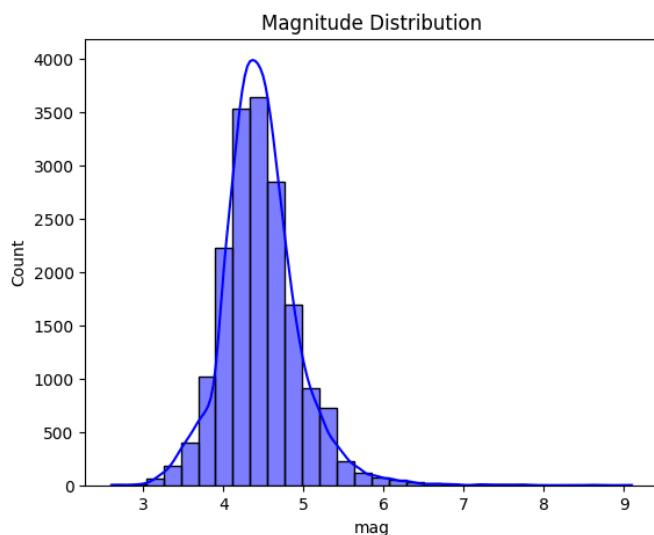


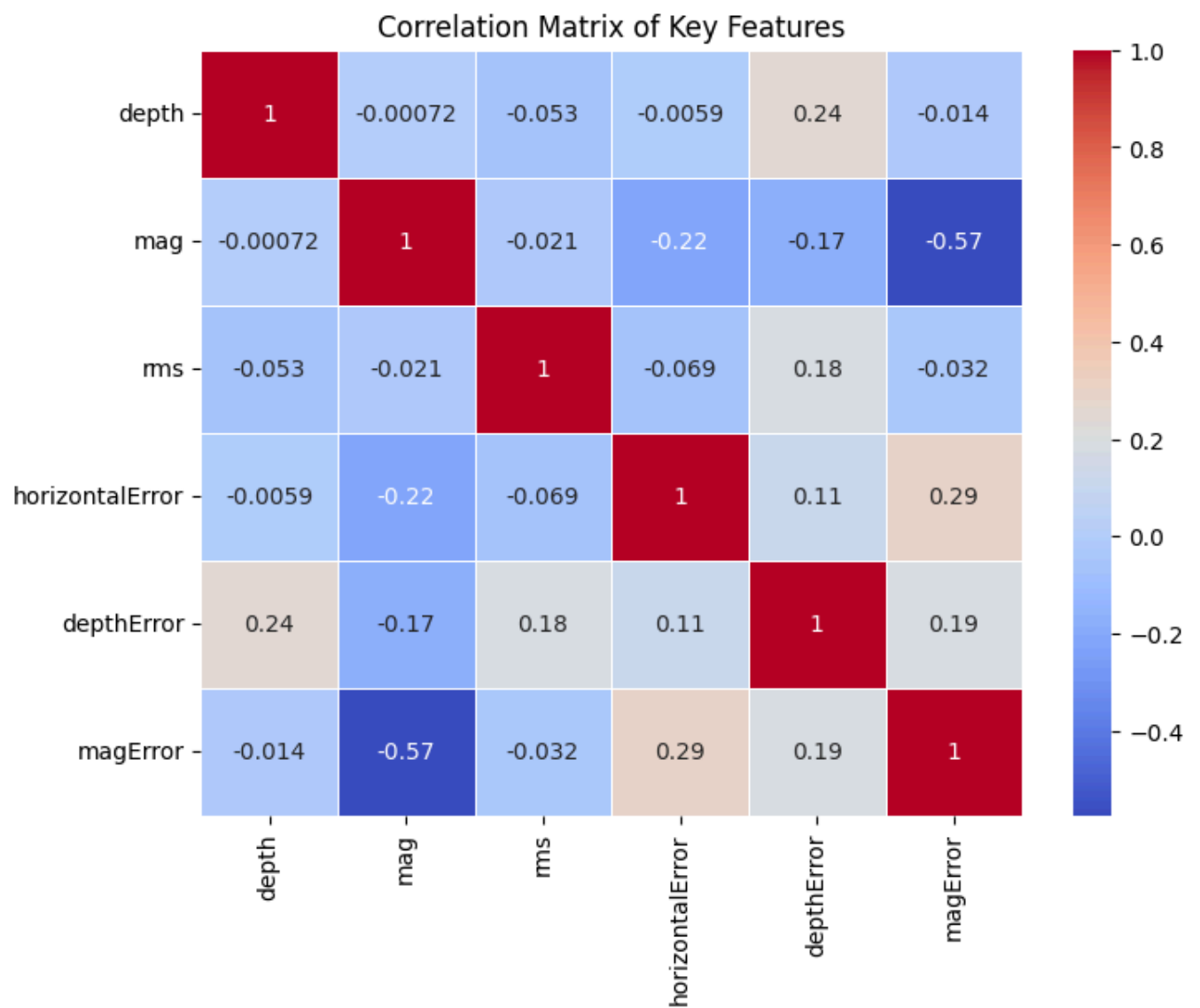
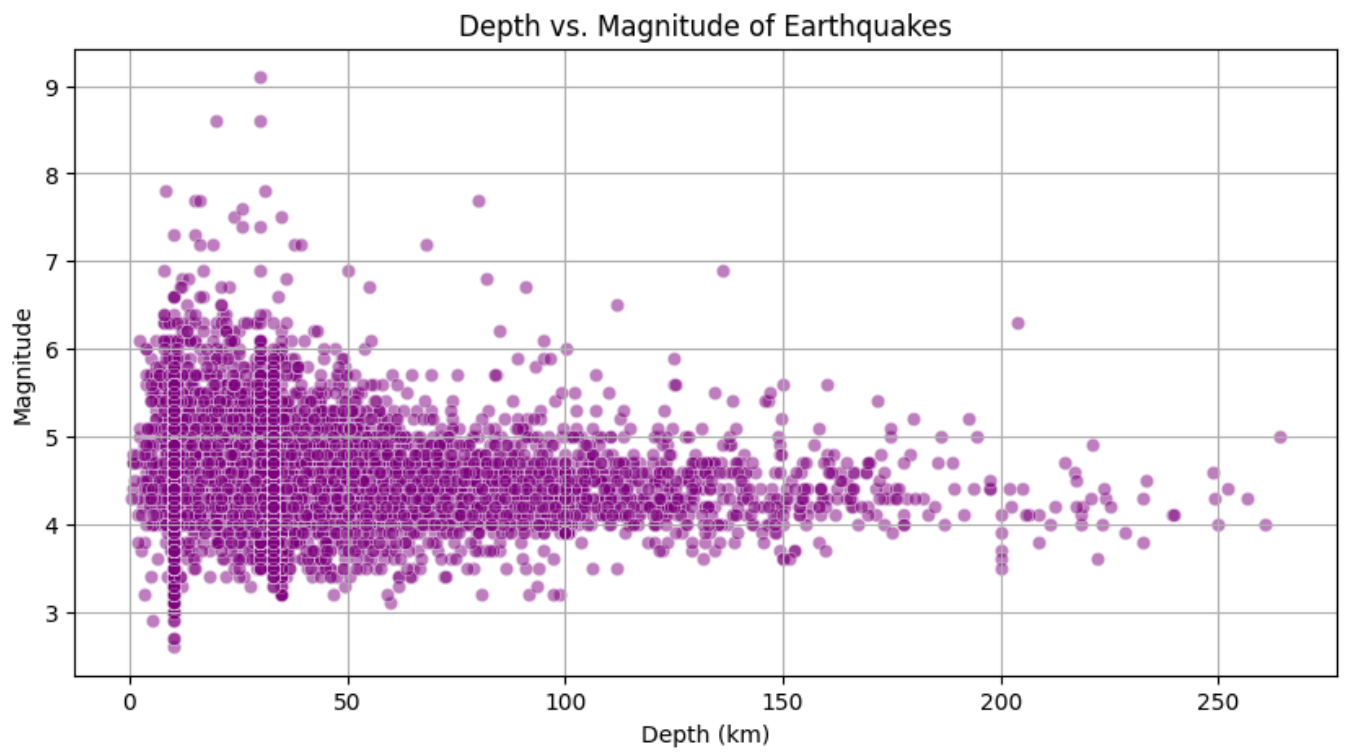
```
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
```

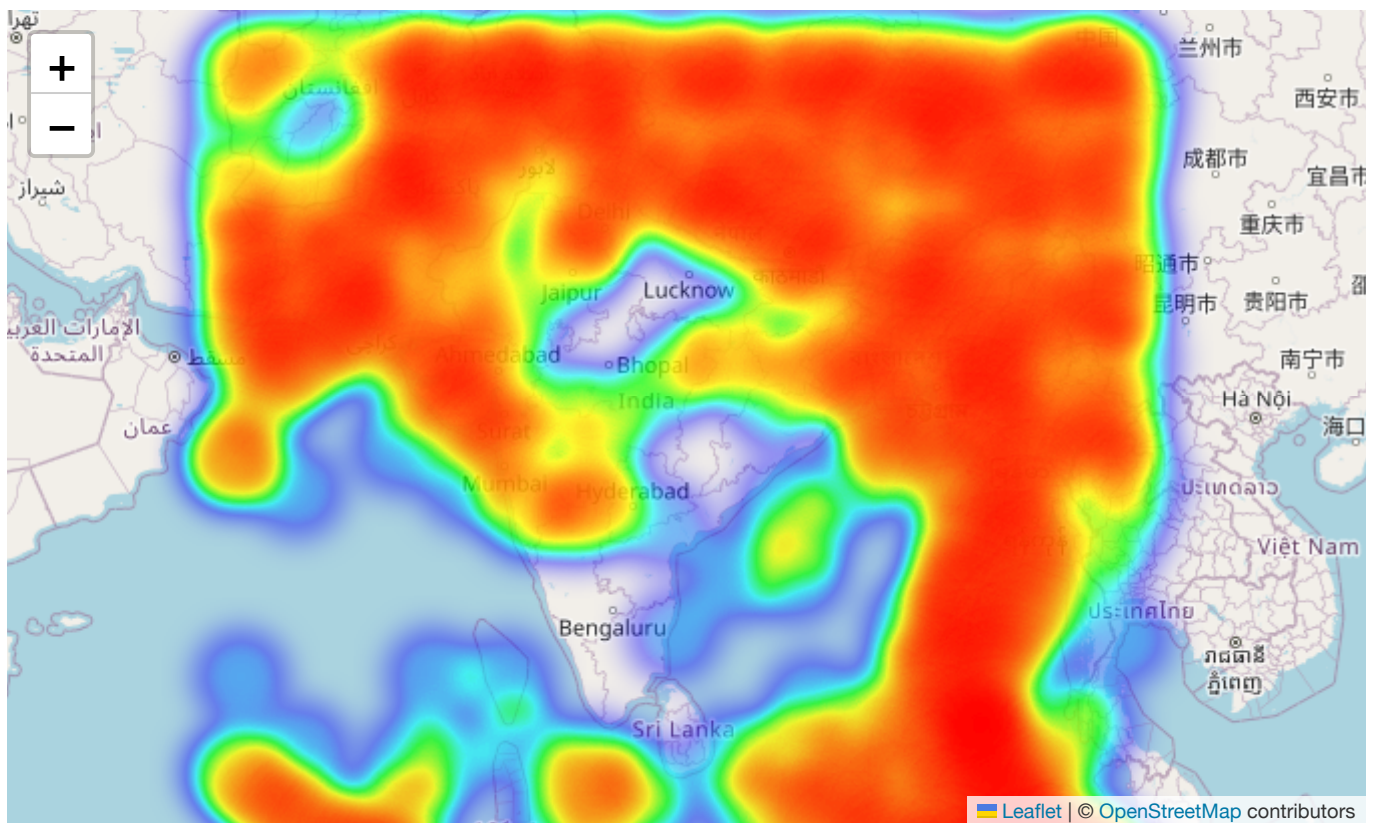
```
with pd.option_context('mode.use_inf_as_na', True):
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
```

```
with pd.option_context('mode.use_inf_as_na', True):
```

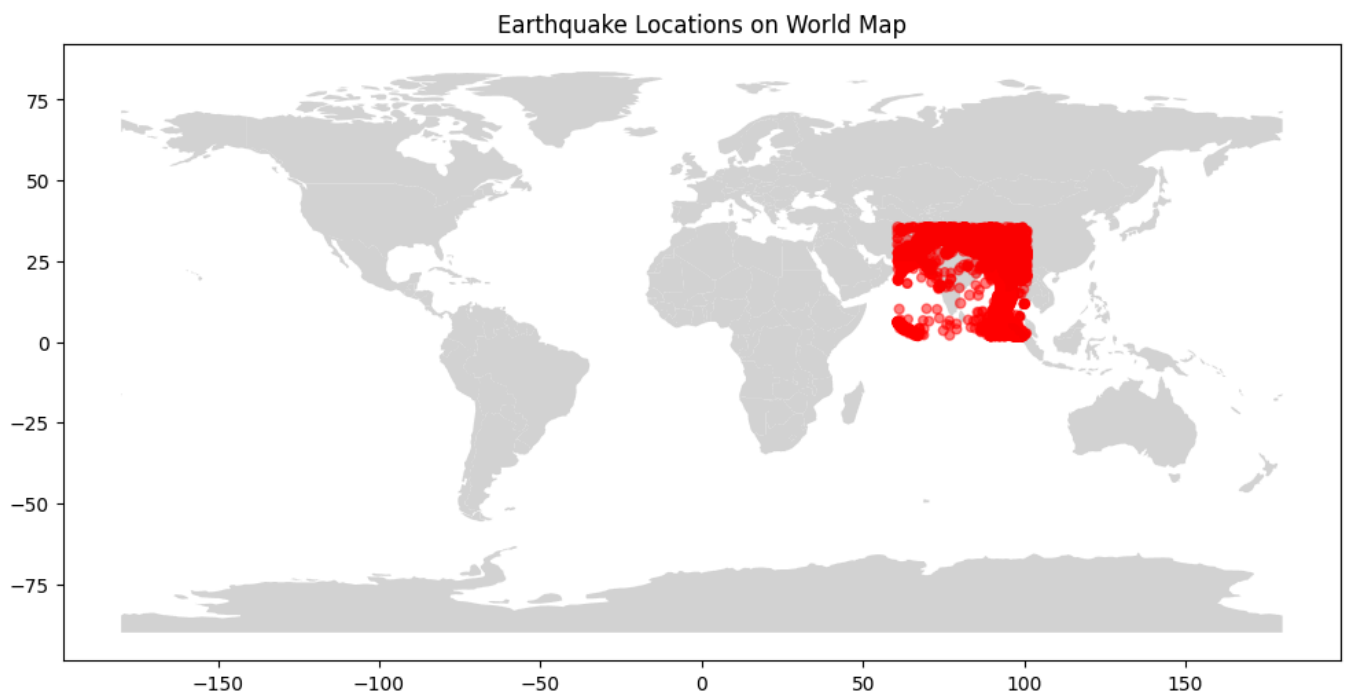






```
<ipython-input-6-8b5041a4fa1b>:48: FutureWarning: The geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get the original 'naturalearth_lowres' data from https://www.naturalearthdata.com/downloads/110m-cultural-vectors/.
```

```
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
```



Top 10 Most Affected Countries by Earthquakes

