Show all your code to acquire the dataset in your notebook

In [ ]:

```
import requests
import pandas as pd
import numpy as np
import io
import matplotlib.pyplot as plt
import us
import plotly.graph_objects as go
#conda install -c plotly/label/test plotly
import seaborn as sns

#If data needs to be loaded from CSV
USGS=pd.read_csv("USGS_BDP_HW4.csv")
```

In [ ]:

```
#If data is loaded from CSV then EXECUTING THIS BLOCK IS NOT REQUIRED
#Run this block only if need current data, and to create CSV
response1 = requests.get("https://earthquake.usgs.gov/fdsnws/event/1/query?format=csv&starttime=2016-01-01&endtime=2017-01-01&minmagnitude=4")
response1_df = pd.read_csv(io.StringIO(response1.content.decode('utf-8')))
response2 = requests.get("https://earthquake.usgs.gov/fdsnws/event/1/query?format=csv&starttime=2017-01-01&endtime=2018-01-01&minmagnitude=4")
response2_df = pd.read_csv(io.StringIO(response2.content.decode('utf-8')))
response3 = requests.get("https://earthquake.usgs.gov/fdsnws/event/1/query?format=csv&starttime=2018-01-01&endtime=2019-01-01&minmagnitude=4")
response3_df = pd.read_csv(io.StringIO(response3.content.decode('utf-8')))
response4 = requests.get("https://earthquake.usgs.gov/fdsnws/event/1/query?format=csv&starttime=2019-01-01&endtime=2019-10-02&minmagnitude=4")
response4_df = pd.read_csv(io.StringIO(response4.content.decode('utf-8')))

USGS=response1_df.append(response2_df)
USGS=USGS.append(response3_df)
USGS=USGS.append(response4_df)
USGS=USGS[USGS['type']=='earthquake']

#data = pd.read_csv("filename.csv")
USGS.to_csv(r'USGS_BDP_HW4.csv')
USGS=pd.read_csv("USGS_BDP_HW4.csv")
```

Use describe to get the basic statistics of all the columns

In [ ]:

```
USGS.describe()
```

```
Get the top 10 earthquakes by magnitude
```

In [ ]:

```
USGS_lar=USGS.nlargest(10,'mag',keep='all')
USGS_lar=USGS_lar.reset_index(drop=True)
USGS_lar.style
```

Handle all Null/empty data by filling it with zeros

In [ ]:

```
#print(USGS.isna().sum())
USGS=USGS.fillna(0)
#print(USGS.isna().sum())
```

Find the top 10 places where the strongest earthquakes occurred

In [ ]:

```
USGS_loc=pd.DataFrame(USGS.nlargest(10,'mag')['place'])
USGS_loc=USGS_loc.apply(lambda x: x[0].split('of')[1],axis=1)
USGS_loc=USGS_loc.reset_index(drop=True)
print(USGS_loc)
```

Find the top 10 places where the weakest earthquakes occurred

In [ ]:

```
USGS_loc=pd.DataFrame(USGS.nsmallest(10,'mag')['place'])
USGS_loc=USGS_loc.apply(lambda x: x[0].split('of')[1],axis=1)
USGS_loc=USGS_loc.reset_index(drop=True)
print(USGS_loc)
```

On a per-year basis, use a bar chart to plot the number of earthquakes for each of the following magnitude groups ranges: Group 1: [4,4.5), Group 2: [4.5,5), Group 3: [5,6), Group 4: [6,7), Group 5: (7,MAX]. Pay close attention to the group ranges. (20 points) Please add labels and colors to the plot

In [ ]:

```
def categorize(USGS_201n, group_interval, group_names):
    group_intervals = [pd.Interval(*gi) for gi in group_interval]
    groups = []
    for mag in USGS_201n:
        GroupN = None
        for i, mag_group in enumerate(group_intervals):
            if mag in mag_group:
                GroupN = group_names[i]
                break
        groups.append(GroupN)
    #print(groups)
    return groups

group_interval=[(4,4.5,'left'),(4.5,5,'left'),(5,6,'left'),(6,7,'left'),(7,10,'right')]
group_names=['G1 [4,4.5)', 'G2 [4.5,5),', 'G3 [5,6),', 'G4 [6,7),', 'G5 (7,MAX),']

years=['2016','2017','2018','2019']
df4plot=pd.DataFrame()
df4plotf=pd.DataFrame()
for i, year in enumerate(years):
    USGS_year=USGS.loc[USGS.apply(lambda x: x['time'].split('-')[0]==year,axis=1),'mag']
    group=categorize(USGS_year, group_interval, group_names)
    df4plot=pd.DataFrame()
    df4plot['group']=group
    df4plot['year']=year
    df4plotf=df4plotf.append(df4plot)

df4plotf=df4plotf.sort_values(by=['group'])

a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
ax = sns.countplot(x="group", hue="year", data=df4plotf)
```

```
Trying other methods for plotting for same que--
On a per-year basis, use a bar chart to plot the number of earthquakes for each of the following
magnitude groups ranges: Group 1: [4,4.5), Group 2: [4.5,5), Group 3: [5,6), Group 4: [6,7), Group 5:
(7,MAX]. Pay close attention to the group ranges. (20 points) Please add labels and colors to the plot.
```

In [ ]:

```
#OR
def categorize(USGS_201n, group_interval, group_names):
    group_intervals = [pd.Interval(*gi) for gi in group_interval]
    groups = []
    for mag in USGS_201n:
        GroupN = None
        for i, mag_group in enumerate(group_intervals):
            if mag in mag_group:
                GroupN = group_names[i]
                break
        groups.append(GroupN)
    #print(groups)
    return groups

group_interval=[(4,4.5,'left'),(4.5,5,'left'),(5,6,'left'),(6,7,'left'),(7,10,'right')]
group_names=['G1 [4,4.5)', 'G2 [4.5,5),', 'G3 [5,6),', 'G4 [6,7),', 'G5 (7,MAX),']

df4plot=pd.DataFrame()
df4plotf=pd.DataFrame()

def garrayf (years):
    for i, year in enumerate(years):
        USGS_year=USGS.loc[USGS.apply(lambda x: x['time'].split('-')[0]==year,axis=1),'mag']
        group=categorize(USGS_year, group_interval, group_names)
        df4plot=pd.DataFrame()
        df4plot['group']=group
        df4plot['year']=year
        #df4plotf=df4plotf.append(df4plot)
        df4plot['freq'] = df4plot.groupby('group')['group'].transform('count')
        df4plot=df4plot[['group','freq']]
        df4plot=df4plot.drop_duplicates(subset='group',keep='first')
        df4plot=df4plot.sort_values(by=['group'], ascending=True)
        df4plot5=df4plot['freq'].head(5)
        garray=df4plot5.tolist()
    return garray

a2016=garrayf(['2016'])
#print(a2016)
a2017=garrayf(['2017'])
#print(a2017)
a2018=garrayf(['2018'])
#print(a2018)
a2019=garrayf(['2019'])
#print(a2019)

barWidth = 0.25
re1 = np.arange(len(a2016))
re2 = [x + barWidth for x in r1]
re3 = [x + barWidth for x in r2]
re4 = [x + barWidth for x in r3]

# Make the plot
plt.bar(re1, a2016, color='#7f6d5f', width=barWidth, edgecolor='white', label='2016')
plt.bar(re2, a2017, color='#557f2d', width=barWidth, edgecolor='white', label='2017')
plt.bar(re3, a2018, color='#2d7f5e', width=barWidth, edgecolor='white', label='2018')
plt.bar(re4, a2019, color='#d4af37', width=barWidth, edgecolor='white', label='2019')

# Add xticks on the middle of the group bars
plt.xlabel('group', fontweight='bold')
plt.xticks([r + barWidth for r in range(len(a2016))], ['G1', 'G2', 'G3', 'G4', 'G5'])

# Create legend & Show graphic
plt.legend()
plt.show()
```

Find the 10 countries with the highest number of earthquakes (30 points) (Note: Yes, this is only countries, not full place)

In [ ]:

```
USGS_con=pd.DataFrame(USGS[['mag','place']])
con=USGS_con['place'].str.split(', ').tolist()
country=[]
for i,sub_list in enumerate(con):
    if(len(sub_list)<2):
        sub_list.append(np.nan)
    country.append(sub_list[1])
USGS_con['place']=country

USGS_con['freq'] = USGS_con.groupby('place')['place'].transform('count')
USGS_con=USGS_con[['place','freq']]
USGS_con=USGS_con.drop_duplicates(subset='place',keep='first')
USGS_con=USGS_con.sort_values(by=['freq'], ascending=False)
USGS_con=USGS_con.reset_index(drop=True)
USGS_con['place'].head(10)
```

Analyze the distribution of the Earthquake magnitudes. This is, make a histogram of the Earthquake count versus magnitude. Make sure to use a Logarithmic scale. What sort of relationship do you see? (20 points) Please add labels and colors to the plot

Answer: The liner relation between logarithmic count and magnitude, shows that there is exponential relationship between the magnitude and its count. Which means the count for small range of magnitute (here 4 to 5) is way more than count of other range of magnitude. Which can also be said as mostly the earthquakes between 2016-01-01 to 2019-10-01 were of magnitude between 4 to 5.

In [ ]:

```
fig, ax = plt.subplots()
fig.text(0.04, 0.5, 'Count', va='center', rotation='vertical')
plt.suptitle('Count versus Magnitude of Earthquake', x=0.5, y=1.05, ha='center', fontsize='xx-large')
USGS.hist('mag', bins=20, color='red',ax=ax)
ax.set_yscale('log')
```

Analyze the distribution of the Earthquake depths. This is, make a histogram of the Earthquake count versus depth. Make sure to use a Logarithmic scale. What sort of relationship do you see? (20 points) Please add labels and colors to the plot.

Answer: The graph of logarithmic count vs depth seems to be bimodular graph. It can be expressed as:
If we divide depth into 3 caterogies : shallow, intermediate and deep
Then we can say most of the times the earthquakes are shallow. And have least frequency of being intermediate if compared to shallow and deep earthquakes.

In [ ]:

```
fig, ax = plt.subplots()
fig.text(0.04, 0.5, 'Count', va='center', rotation='vertical')
plt.suptitle('Count versus Depth of Earthquake', x=0.5, y=1.05, ha='center', fontsize='xx-large')
USGS.hist('depth', bins=50, color='brown',ax=ax)
ax.set_yscale('log')
```

Visualize the locations of earthquakes by making a scatterplot of their latitude and longitude. (20 points) Please add labels and colors to the plot.

In [ ]:

```
plt.figure(figsize=(19, 10))
plt.scatter(USGS['longitude'], USGS['latitude'], c='b', marker='.')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Locations of Earthquakes')
plt.show()
```

Using the US package (https://pypi.org/project/us/), clean the dataset you used previously to only have data from the USA . You need to create a function that accommodates this. (20 points)

In [ ]:

```
def get_only_USA_Data(full_USGS):
    USGS_con=pd.DataFrame(full_USGS[['mag','place']])
    con=USGS_con['place'].str.split(', ').tolist()
    country=[]
    for i,sub_list in enumerate(con):
        if(len(sub_list)<2):
            sub_list.append(np.nan)
        country.append(sub_list[1])
    USGS_con['place']=country

    US_list=[]
    for i in range(len(country)):
        if us.states.lookup(str(country[i])) is None:
            US_list.append(np.nan)
        else:
            US_list.append(country[i])

    USGS['UScheck']=US_list
    USGS_US=USGS.dropna(subset=['UScheck'])
    USGS_US=USGS_US.drop(columns=['UScheck'])
    return USGS_US

USGS_US=get_only_USA_Data(USGS)
USGS_US=USGS_US.reset_index(drop=True)
USGS_US.style
```

Find the top 10 US states where the strongest earthquakes occurred

In [ ]:

```
USGS_con=pd.DataFrame(USGS_US[['mag','place']])
con=USGS_con['place'].str.split(', ').tolist()
country=[]
for i,sub_list in enumerate(con):
    if(len(sub_list)<2):
        sub_list.append(np.nan)
    country.append(sub_list[1])
USGS_con['place']=country

US_list=[]
for i in range(len(country)):
    if us.states.lookup(str(country[i])) is None:
        US_list.append(np.nan)
    else:
        US_list.append(us.states.lookup(str(country[i])))

USGS_con['place']=US_list

USGS_con=USGS_con.sort_values(by=['mag'], ascending=False)
USGS_con=USGS_con.drop_duplicates(subset='place',keep='first')
topten=USGS_con['place'].head(10)
topten=pd.DataFrame(topten)
topten=topten.reset_index(drop=True)
topten.style
```

On a per-year basis, use a bar chart to plot the number of earthquakes for each of the following magnitude groups ranges: Group 1: [4,4.5), Group 2: [4.5,5), Group 3: [5,6), Group 4: [6,7), Group 5: (7,MAX]. Pay close attention to the group ranges. (10 points) Please add labels and colors to the plot.

In [ ]:

```
def categorize(USGS_201n, group_interval, group_names):
    group_intervals = [pd.Interval(*gi) for gi in group_interval]
    groups = []
    for mag in USGS_201n:
        GroupN = None
        for i, mag_group in enumerate(group_intervals):
            if mag in mag_group:
                GroupN = group_names[i]
                break
        groups.append(GroupN)
    #print(groups)
    return groups

group_interval=[(4,4.5,'left'),(4.5,5,'left'),(5,6,'left'),(6,7,'left'),(7,10,'right')]
group_names=['G1 [4,4.5)', 'G2 [4.5,5),', 'G3 [5,6),', 'G4 [6,7),', 'G5 (7,MAX),']

years=['2016','2017','2018','2019']
df4plot=pd.DataFrame()
df4plotf=pd.DataFrame()
for i, year in enumerate(years):
    USGS_year=USGS_US.loc[USGS_US.apply(lambda x: x['time'].split('-')[0]==year,axis=1),'mag']
    group=categorize(USGS_year, group_interval, group_names)
    df4plot=pd.DataFrame()
    df4plot['group']=group
    df4plot['year']=year
    df4plotf=df4plotf.append(df4plot)

df4plotf=df4plotf.sort_values(by=['group'])

a4_dims = (11.7, 8.27)
fig, ax = plt.subplots(figsize=a4_dims)
ax = sns.countplot(x="group", hue="year", data=df4plotf)
```

Visualize the locations of earthquakes by making a scatterplot of their latitude and longitude. Overlay a US map on top of this plot to match the locations. (20 points) Please add labels and colors to the plot.

In [ ]:

```
USGS_US_map=USGS_US
USGS_US_map['info']='mag:'+USGS_US_map['mag'].astype(str)+' loc:'+USGS_US_map['place']

fig = go.Figure(data=go.Scattergeo(
        lon = USGS_US_map['longitude'],
        lat = USGS_US_map['latitude'],
        text = USGS_US_map['info'],
        mode = 'markers',
        marker_color = USGS_US_map['mag'],
        ))

fig.update_layout(
        title = 'Earthquakes in USA<br>(Hover for magnitude)',
        geo_scope='usa',
    )
fig.show()
```