Homework 4

Ivan Escalona-Faria

Panther ID: 002395228

Question 0: Show all your code to acquire the dataset in your notebook. If you fail to show how you acquired the dataset, you will earn a zero on the assignment.

In [ ]:
```
import numpy as np
import requests
import pandas as pd
import matplotlib.pyplot as plt
import io
```

In [ ]:
```

years = ['2016','2017','2018','2019']
months = ['01','02', '03', '04', '05', '06', '07', '08', '09', '10', '11
year_month = []

for year in years:

    for month in months:
        if (year == '2019' and (month == '11' or month == '12') ):
            pass
        else:
            year_month.append(f'{year}-{month}')
```

In [ ]:
```


snws/event/1/query?format=csv&starttime={year_month[i]}-01&endtime={year_mont



```

In [ ]:
```
#https://stackoverflow.com/questions/39213597/convert-text-data-from-req

df = pd.DataFrame()

for i in range(0,len(data)):
    temp_data = pd.read_csv(io.StringIO(data[i].content.decode('utf-8'))
    df = pd.concat([df,temp_data])
```

In [ ]:
```
response = requests.get('https://earthquake.usgs.gov/fdsnws/event/1/quer
print(response.status_code)
temp_data = pd.read_csv(io.StringIO(response.content.decode('utf-8')))
df = pd.concat([df,temp_data])
```

In [ ]:
```
data_frame = df.copy()
```

In [ ]:
```
df = data_frame.copy()
```

In [ ]:
```
df.to_csv(path_or_buf='Source_Data.csv', index=False)
```

Question 1: Use describe to get the basic statistics of all the columns (5 points)

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

In [ ]:
```
#Exploring of the data

df.info()
```

Question 2: Get the top 10 earthquakes by magnitude (5 points)

In [ ]:
```
print(df.sort_values(by = 'mag', ascending = False).head(10))
```

Question 3: Handle all Null/empty data by filling it with zeros (10 points)

In [ ]:
```
df.fillna(value = 0, inplace = True)
```

In [ ]:
```
print(df.info())
```

Question 4: Find the top 10 places where the strongest earthquakes occurred (15 points) (Note: Place needs to be parsed nicely to remove the KM location from them. For example: 75km WSW of Illapel, Chile should look like Illapel, Chile).

In [ ]:
```
# https://stackoverflow.com/questions/29247718/map-an-if-statement-in-py
#https://www.geeksforgeeks.org/python-string-isnumeric-application/

df['place'] = df['place'].map(lambda x: ' '.join(x.split(' ')[3:]) if (x
q4 = df.groupby(['place', 'mag']).count().sort_values(by = 'mag', ascend
print(q4.head(10))
```

Question 5: Find the top 10 places where the weakest earthquakes occurred (15 points) (Note: Place needs to be parsed nicely to remove the KM location from them. For example: 75km WSW of Illapel, Chile should look like Illapel, Chile).

In [ ]:
```
q5 = df.groupby(['place', 'mag']).count().sort_values(by = 'mag', ascend
print(q5.head(10))
```

Question 6: 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 [ ]:
```
#Parsing time into just year (First 4 digits of the string)

df['year'] = df['time'].map(lambda x: x[:4])
```

In [ ]:
```
#Creating the required bins

bins = {'4.0-4.5': [4.0, 4.5], '4.5-5.0': [4.5, 5.0], '5.0-6.0': [5.0, 6
bins
```

In [ ]:
```
#Defining the function to classify and then map each earthquake by magni

def magnitude_bin(x):
    if x < 4.5:
        return list(bins.keys())[0]
    elif x < 5.0:
        return list(bins.keys())[1]
    elif x < 6.0:
        return list(bins.keys())[2]
    elif x < 7.0:
        return list(bins.keys())[3]
    else:
        return list(bins.keys())[4]


```

In [ ]:
```
df['mag_bins'] = df['mag'].map(lambda x: magnitude_bin(x))
```

In [ ]:
```
q6 = df.groupby(by = ['year','mag_bins'] ).count()['type']
years = df['year'].unique().tolist()
years
```

In [ ]:
```
hist_bins = q6['2016'].index.to_list()
```

In [ ]:
```
hist_bins
```

In [ ]:
```
colors = ['#18FA18', '#F2F921', '#F92128', '#FAB618']
```

In [ ]:
```
x_positions = list(range(1,6))
```

In [ ]:
```
fig, ax = plt.subplots(1, 1, sharey=True, tight_layout=True)

for i in range(0,len(years)):
    x_locations = list(map(lambda x: float(x)+(0.2*i), x_positions))
    ax.bar(x = x_locations, height = q6[f'{years[i]}'], width = 0.2, col

ax.set_xticks(list(map(lambda x: x+(0.3), x_locations)))
ax.set_ylabel('Count of Earthquakes')
ax.set_xlabel('Earthquake Magnitude Bins')
ax.set_title('Count of Earthquakes vs Magnitude bins')
ax.legend(years)
```

Question 7: Find the 10 countries with the highest number of earthquakes (30 points) (Note: Yes, this is only countries, not full place)

In [ ]:
```
df['country'] = df['place'].map(lambda x: x.split(',')[-1] if ',' in x e

q7 = df.groupby('country')['mag'].count().sort_values(ascending = False)
```

In [ ]:
```
print(q7)
```

Question 8: 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.

In [ ]:
```
fig, ax = plt.subplots(1, 1, sharey=True, tight_layout=True)

ax.hist(x = df['mag'], log = True, width = 0.4, color = 'g')

ax.set_ylabel('Log of count of Earthquakes')
ax.set_xlabel('Earthquake Magnitude')
ax.set_title('Log of count of Earthquakes vs Magnitude')
```

There is an aparent strong linear relation with negative correlation, and a deviation from linear behavior pass the magnitude of 7

Question 9: 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.

In [ ]:
```
#Question 9

fig, ax = plt.subplots(1, 1, sharey=True, tight_layout=True)


ax.hist(x = df['depth'], log = True, width = 65, color = 'r')

ax.set_ylabel('Log of count of Earthquakes')
ax.set_xlabel('Earthquake Depth')
ax.set_title('Log of count of Earthquakes vs Depth')
```

The trend for this data is not as clear as the one on question 8. The linear model would not be as strong, but it would be negatively correlated. A more complex model (quadratic, cubic, bimodal, etc.) could better fit this data.

Question 10: 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 [ ]:
```
fig, ax = plt.subplots(1, 1, sharey=True, tight_layout=True)
ax.scatter(x = df['longitude'], y = df['latitude'], alpha = 0.6, s = df[
ax.set_title('Lat-Long earthquake location map')
ax.set_ylabel('Latitude')
ax.set_xlabel('Longitude')
```