In [ ]:

```
import requests
import io
import numpy as np
import scipy as sp
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline

# parameters = {
#     "format": "csv",
#     "starttime": "2016-01-01",
#     "endtime": "2017-01-01",
#     "minmagnitude": 4,
#     "eventtype": "earthquake"
# }
#response1 = requests.get("https://earthquake.usgs.gov/fdsnws/event/1/qu
ery", params = parameters, timeout=20).content
#parameters2 = {
#     "format": "csv",
#     "starttime": "2017-01-01",
#     "endtime": "2018-01-01",
#     "minmagnitude": 4,
#     "eventtype": "earthquake"
# }
#response2 = requests.get("https://earthquake.usgs.gov/fdsnws/event/1/qu
ery", params = parameters2, timeout=20).content
#parameters3 = {
#     "format": "csv",
#     "starttime": "2018-01-01",
#     "endtime": "2019-01-01",
#     "minmagnitude": 4,
#     "eventtype": "earthquake"
# }
#response3 = requests.get("https://earthquake.usgs.gov/fdsnws/event/1/qu
ery", params = parameters3, timeout=20).content
#parameters4 = {
#     "format": "csv",
#     "starttime": "2019-01-01",
#     "endtime": "2019-10-02",
#     "minmagnitude": 4,
#     "eventtype": "earthquake"
# }
#response4 = requests.get("https://earthquake.usgs.gov/fdsnws/event/1/qu
ery", params = parameters4, timeout=20).content

#open('earthquake_data', 'ab').write(response1)
#open('earthquake_data', 'ab').write(response2)
#open('earthquake_data', 'ab').write(response3)
#open('earthquake_data', 'ab').write(response4)

f=open("earthquake_data", "rb")
if f.mode == 'rb':
    file = f.read()


    #df1 = pd.read_csv(io.StringIO(file.decode('utf-8')))
    #df2 = pd.read_csv(io.StringIO(('earthquake_data2').decode('utf-
8')))
    #df3 = pd.read_csv(io.StringIO(('earthquake_data3').decode('utf-
8')))
    #df4 = pd.read_csv(io.StringIO(('earthquake_data4').decode('utf-
8')))

    df = pd.read_csv(io.StringIO(file.decode('utf-8')))
    df.to_csv('earthquake_data.csv')

    #question1
    print(df.describe())

    #question2
    df_top_mag = df.sort_values(by = 'mag', ascending = False)
    print(df_top_mag.head(10))

    #question3
    df.fillna(0)

    #question4
    df_top_mag = df.sort_values(by = 'mag', ascending = False)
    df_low_mag = df.sort_values(by = 'mag', ascending = True)
    print(df_top_mag['place'].head(10).map(lambda x: x.split('of')[1]))
    print(df_low_mag['place'].head(10).map(lambda x: x.split('of')[1]))

    #question5
    df_2016 = df[df['time'] > '2016'][df['time'] < '2017']
    df_2017 = df[df['time'] > '2017'][df['time'] < '2018']
    df_2018 = df[df['time'] > '2018'][df['time'] < '2019']
    df_2019 = df[df['time'] > '2019']

    #question6
    labels= ['Group1', 'Group2', 'Group3', 'Group4', 'Group5']

    group2016 = [df_2016[df_2016['mag'] >= '4'][df_2016['mag'] < '4.5'][
'mag'].count(), \
              df_2016[df_2016['mag'] >= '4.5'][df_2016['mag'] < '5']['ma
g'].count(), \
             df_2016[df_2016['mag'] >= '5'][df_2016['mag'] < '6']['mag']
.count(), \
             df_2016[df_2016['mag'] >= '6'][df_2016['mag'] < '7']['mag']
.count(),
             df_2016[df_2016['mag'] >= '7']['mag'].count()]

    group2017 = [df_2017[df_2017['mag'] >= '4'][df_2017['mag'] < '4.5'][
'mag'].count(), \
              df_2017[df_2017['mag'] >= '4.5'][df_2017['mag'] < '5']['ma
g'].count(), \
             df_2017[df_2017['mag'] >= '5'][df_2017['mag'] < '6']['mag']
.count(), \
             df_2017[df_2017['mag'] >= '6'][df_2017['mag'] < '7']['mag']
.count(),
             df_2017[df_2017['mag'] >= '7']['mag'].count()]

    group2018 = [df_2018[df_2018['mag'] >= '4'][df_2018['mag'] < '4.5'][
'mag'].count(), \
              df_2018[df_2018['mag'] >= '4.5'][df_2018['mag'] < '5']['ma
g'].count(), \
             df_2018[df_2018['mag'] >= '5'][df_2018['mag'] < '6']['mag']
.count(), \
             df_2018[df_2018['mag'] >= '6'][df_2018['mag'] < '7']['mag']
.count(),
             df_2018[df_2018['mag'] >= '7']['mag'].count()]

    group2019 = [df_2019[df_2019['mag'] >= '4'][df_2019['mag'] < '4.5'][
'mag'].count(), \
              df_2019[df_2019['mag'] >= '4.5'][df_2019['mag'] < '5']['ma
g'].count(), \
             df_2019[df_2019['mag'] >= '5'][df_2019['mag'] < '6']['mag']
.count(), \
             df_2019[df_2019['mag'] >= '6'][df_2019['mag'] < '7']['mag']
.count(),
             df_2019[df_2019['mag'] >= '7']['mag'].count()]

    x = np.arange(len(labels))
    width = 0.2

    men_means = [20, 34, 30, 35, 27]
    women_means = [25, 32, 34, 20, 25]

    fig, ax = plt.subplots()
    rects1 = ax.bar(x - 0.4, group2016, width, label='2016')
    rects2 = ax.bar(x - 0.2, group2017, width, label='2017')
    rects3 = ax.bar(x , group2018, width, label='2018')
    rects4 = ax.bar(x + 0.2, group2019, width, label='2019')

    ax.set_ylabel('# of earthquakes')
    ax.set_title('Earthquakes by Magnitude level')
    ax.set_xticks(x)
    ax.set_xticklabels(labels)
    ax.legend()

    fig.tight_layout()
    plt.show()

    #question 7
    print(df[df['place'].str.contains(',')]['place']\
          .value_counts().index.map(lambda x: x.split(',')[1])\
          .unique().tolist()[:10])

    #question 8
    fig, ax = plt.subplots()
    ax.hist(df['mag'], bins=20, log=True, color='red')
    ax.set_title('Earthquake Count vs Magnitude')
    ax.set_xlabel('Earthquake Magnitude')
    ax.set_ylabel('Earthquake Count')
    plt.show()

    #question 9
    fig, ax = plt.subplots()
    ax.hist(df['depth'], bins=20, log=True color='blue')
    ax.set_title('Earthquake Count vs Depth')
    ax.set_xlabel('Earthquake Depth')
    ax.set_ylabel('Earthquake Count')
    plt.show()

    #question 10
    colors=['green', 'blue']
    plt.scatter(df['latitude'], df['longitude'], c=colors, alpha=0.5)
    plt.title('Earthquake Latitude vs Longitude')
    plt.xlabel('latitude')
    plt.ylabel('longitude')
    plt.show()
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: