Sam Logsdon - Assignment4

October 23, 2019

0.0.1 Fetch data

[ ]:

```
import requests
import os
from datetime import datetime


dates = [('2016-01-01', '2017-01-01'), ('2017-01-01', '2018-01-01'),␣
 ↪('2018-01-01', '2019-01-01'), ('2019-01-01', '2019-10-02')]

if os.path.isfile('data.csv'):
    os.remove('data.csv')

def fetch(url, params):
    response = requests.get(url, params)
    if response.status_code == 200:
        return response.text
    else:
        print(response.text)

first = True
for start, end in dates:
    params = {'format': 'csv', 'minmagnitude': 4.0, 'starttime': start,␣
 ↪'endtime': end}
    data = fetch('https://earthquake.usgs.gov/fdsnws/event/1/query', params)
    with open('data.csv', 'a') as fp:
        if not first:
            lines = data.split('\n')
            for line in lines[1:]:
                fp.write(line + '\n')
        else:
            fp.write(data)
            first = False
```

0.0.2 Run this block first

[ ]:

```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
df = pd.read_csv('data.csv')
df.time = pd.to_datetime(df.time)
```

0.0.3 Question 1

Use describe to get basic statistics for all the columns

[ ]:

```
with pd.option_context('display.max_columns', 40):
    print(df.describe(include='all'))
```

0.0.4 Question 2

Get the top 10 earthquakes by magnitude

[ ]:

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

0.0.5 Question 3

Handle null/empty values by filling it with zeroes

[ ]:

```
df = df.fillna(0)
```

0.0.6 Question 4

Find the top 10 places where the strongest earthquakes occurred

[ ]:

```
def extract_place(place: str):
    return place.split('of')[-1]

df.place = df.place.map(extract_place)
df.sort_values('mag', ascending=False).place.head(10)
```

0.0.7 Question 5

Find the top 10 places where the weakest earthquakes occurred

[ ]:

```
df.sort_values('mag', ascending=True).place.head(10)
```

0.0.8 Question 6

[ ]:
```
intervals = [
    (4,4.5),
    (4.5,5),
    (5,6),
    (6,7),
    (7, df.mag.max())
]
df['magbins'] = pd.cut(df.mag, pd.IntervalIndex.from_tuples(intervals))
df['year'] = df.time.map(lambda d: d.year)
data = df.groupby(['year', 'magbins']).agg(
    freq = pd.NamedAgg(column='mag', aggfunc='count')
)
ax = data.unstack().plot(kind='bar', title='Number of Earthquakes by Magnitude␣
 ↪per Year')
ax.legend(['4.0-4.5', '4.5-5.0', '5.0-6.0', '6.0-7.0', '7.0-8.2'])
```

0.0.9 Question 7

[ ]:
```
def extract_country(place: str):
    s = place.split(' of ')[-1]
    return s.split(',')[-1]
df['country'] = df.place.map(extract_country)
df.groupby('country').country.agg('count').sort_values(ascending=False).head(10)
```

0.0.10 Question 8

The frequency appears to be inversely proportional to the magnitude.

[ ]:
```
plt.hist(x=df.mag, bins=40, log=True, color='purple', alpha=0.5)
plt.title('Distribution of Earthquakes by Magnitude')
plt.xlabel('Magnitude')
plt.ylabel('Earthquake Frequency')
```

0.0.11 Question 9

Similar to the relationship between magnitude and frequency, but less even across the distribution

[ ]:
```
plt.hist(x=df.depth, log=True, bins=40, color='green', alpha=0.5)
plt.title('Distribution of Earthquakes by Depth')
plt.xlabel('Depth')
plt.ylabel('Earthquake Frequency')
```

0.0.12 Question 10

[ ]:

```
#fig, ax = plt.subplots(figsize=(12, 4))
plt.rcParams['figure.figsize'] = [9.5, 6]
plt.scatter(df.latitude, df.longitude, c=df.mag)
plt.xlabel('latitude')
plt.ylabel('longitude')
plt.title('Earthquake Locations')
plt.colorbar(label='Magnitude')
```