# Earthquakes

May 22, 2018

## 1 Homework 6 Problem 2

### 1.1 Spatiotemporal Data : Mapping and analysis of earthquakes

In [1]:
```
!pip install --upgrade matplotlib
!conda install -y basemap

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import datetime as dt
import seaborn as sea
from mpl_toolkits.basemap import Basemap

import warnings
warnings.filterwarnings('ignore')
%config InlineBackend.figure_format = 'retina'
```

```
Requirement already up-to-date: matplotlib in /opt/conda/lib/python3.6/site-packages
Requirement already up-to-date: cycler>=0.10 in /opt/conda/lib/python3.6/site-packages/cycler-0.
Requirement already up-to-date: pytz in /opt/conda/lib/python3.6/site-packages (from matplotlib)
Requirement already up-to-date: six>=1.10 in /opt/conda/lib/python3.6/site-packages (from matplo
Requirement already up-to-date: numpy>=1.7.1 in /opt/conda/lib/python3.6/site-packages (from mat
Requirement already up-to-date: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /opt/conda/lib/pytho
Requirement already up-to-date: kiwisolver>=1.0.1 in /opt/conda/lib/python3.6/site-packages (fro
Requirement already up-to-date: python-dateutil>=2.1 in /opt/conda/lib/python3.6/site-packages (
Requirement already up-to-date: setuptools in /opt/conda/lib/python3.6/site-packages (from kiwis
You are using pip version 9.0.1, however version 10.0.1 is available.You should consider upgradi
Fetching package metadata ...
Solving package specifications: .

# All requested packages already installed.
# packages in environment at /opt/conda:
#
basemap                   1.1.0                    py36_4    conda-forge
```

In [2]:
```
quake_df = pd.read_csv('earthquakes.csv')

quake_df.head()
```

Out[2]:

| | Date | Time | Latitude | Longitude | Type | Depth | Depth Error | \ |
|---|---|---|---|---|---|---|---|---|
| 0 | 01/02/1965 | 13:44:18 | 19.246 | 145.616 | Earthquake | 131.6 | NaN | |
| 1 | 01/04/1965 | 11:29:49 | 1.863 | 127.352 | Earthquake | 80.0 | NaN | |
| 2 | 01/05/1965 | 18:05:58 | -20.579 | -173.972 | Earthquake | 20.0 | NaN | |
| 3 | 01/08/1965 | 18:49:43 | -59.076 | -23.557 | Earthquake | 15.0 | NaN | |
| 4 | 01/09/1965 | 13:32:50 | 11.938 | 126.427 | Earthquake | 15.0 | NaN | |

| | Depth Seismic Stations | Magnitude | Magnitude Type | ... | \ |
|---|---|---|---|---|---|
| 0 | NaN | 6.0 | MW | ... | |
| 1 | NaN | 5.8 | MW | ... | |
| 2 | NaN | 6.2 | MW | ... | |
| 3 | NaN | 5.8 | MW | ... | |
| 4 | NaN | 5.8 | MW | ... | |

| | Magnitude Seismic Stations | Azimuthal Gap | Horizontal Distance | \ |
|---|---|---|---|---|
| 0 | NaN | NaN | NaN | |
| 1 | NaN | NaN | NaN | |
| 2 | NaN | NaN | NaN | |
| 3 | NaN | NaN | NaN | |
| 4 | NaN | NaN | NaN | |

| | Horizontal Error | Root Mean Square | ID | Source | Location Source | \ |
|---|---|---|---|---|---|---|
| 0 | NaN | NaN | ISCGEM860706 | ISCGEM | ISCGEM | |
| 1 | NaN | NaN | ISCGEM860737 | ISCGEM | ISCGEM | |
| 2 | NaN | NaN | ISCGEM860762 | ISCGEM | ISCGEM | |
| 3 | NaN | NaN | ISCGEM860856 | ISCGEM | ISCGEM | |
| 4 | NaN | NaN | ISCGEM860890 | ISCGEM | ISCGEM | |

| | Magnitude Source | Status |
|---|---|---|
| 0 | ISCGEM | Automatic |
| 1 | ISCGEM | Automatic |
| 2 | ISCGEM | Automatic |
| 3 | ISCGEM | Automatic |
| 4 | ISCGEM | Automatic |

[5 rows x 21 columns]

In [3]:
```
quake_df = quake_df[['Date', 'Latitude', 'Longitude', 'Magnitude', 'Depth', 'Type']]

quake_df.head()
```

Out[3]:

| | Date | Latitude | Longitude | Magnitude | Depth | Type |
|---|---|---|---|---|---|---|
| 0 | 01/02/1965 | 19.246 | 145.616 | 6.0 | 131.6 | Earthquake |
| 1 | 01/04/1965 | 1.863 | 127.352 | 5.8 | 80.0 | Earthquake |

```
2  01/05/1965   -20.579   -173.972        6.2   20.0  Earthquake
3  01/08/1965   -59.076    -23.557        5.8   15.0  Earthquake
4  01/09/1965    11.938    126.427        5.8   15.0  Earthquake
```

In [4]:
```
quake_df["Date"] = pd.to_datetime(quake_df["Date"])

quake_df.describe()
```

Out[4]:

| | Latitude | Longitude | Magnitude | Depth |
|---|---|---|---|---|
| count | 23412.000000 | 23412.000000 | 23412.000000 | 23412.000000 |
| mean | 1.679033 | 39.639961 | 5.882531 | 70.767911 |
| std | 30.113183 | 125.511959 | 0.423066 | 122.651898 |
| min | -77.080000 | -179.997000 | 5.500000 | -1.100000 |
| 25% | -18.653000 | -76.349750 | 5.600000 | 14.522500 |
| 50% | -3.568500 | 103.982000 | 5.700000 | 33.000000 |
| 75% | 26.190750 | 145.026250 | 6.000000 | 54.000000 |
| max | 86.005000 | 179.998000 | 9.100000 | 700.000000 |

In [5]:
```
# TODO: See what the distribution of values is
quake_df.pivot_table(index = 'Type', values = 'Magnitude', aggfunc=len)
```

Out[5]:
```
Type
Earthquake           23232.0
Explosion                4.0
Nuclear Explosion      175.0
Rock Burst               1.0
Name: Magnitude, dtype: float64
```

In [6]:
```
quake_df=quake_df[quake_df['Type'] == 'Earthquake']
```

### 1.2 Step 2.1 Earthquakes Only, truncate at 2 decimal places

By converting to int * 100. Then group and count.

In [7]:
```
# TODO:
# Truncate to 2 decimal places
#quake_df=quake_df.round(decimals=2)
s1=(100*quake_df['Latitude']).astype(int)
s2=(100*quake_df['Longitude']).astype(int)

quake_df['near_lat']=s1
quake_df['near_lon']=s2
#quake_df=quake_df.drop_duplicates(subset=['Date', 'near_lat','near_lon'])
#grouped=quake_df.groupby(['near_lat'])
#grouped=quake_df.groupby(['near_lat','near_lon']).Date.nunique().reset_index()
grouped=quake_df.groupby(['near_lat','near_lon'])['Date'].count().reset_index()
grouped=grouped.rename(columns={'Date':'quakes'})
grouped=grouped.sort_values(by = ['quakes'], ascending = False)
grouped.head()
```

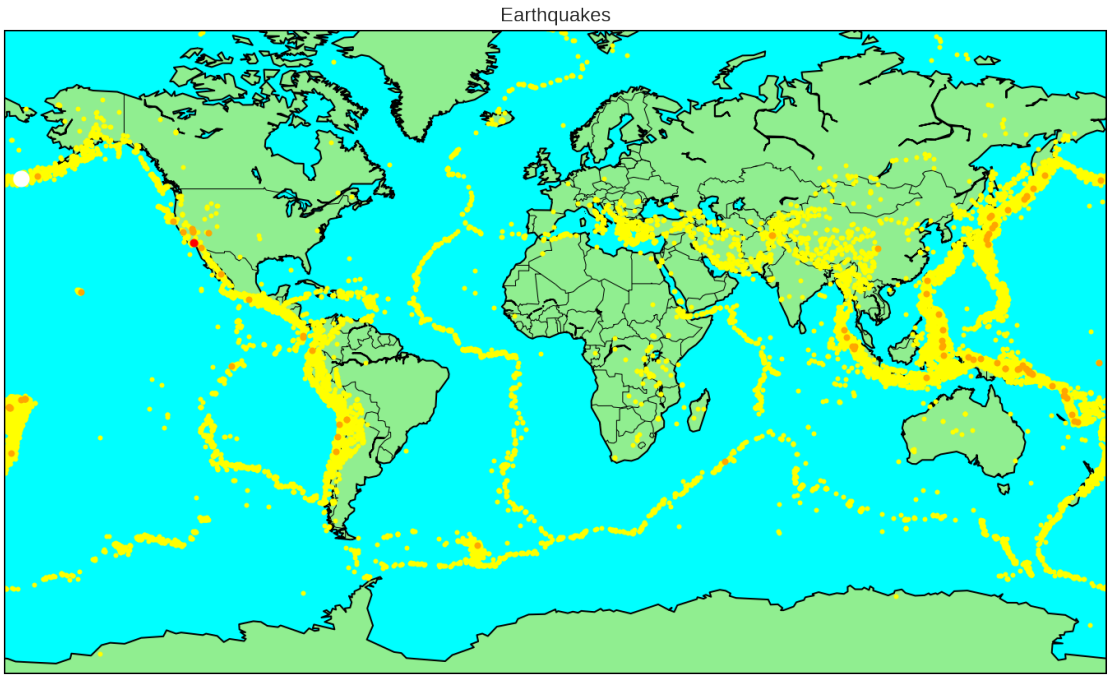
```
Out[7]:        near_lat  near_lon  quakes
        21928      5150    -17480       4
        18556      3441    -11837       3
        13924       492     -8256       2
        4467      -2230     16953       2
        12240      -163     13491       2
```

## 2 Step 2.2 Plot on Map

Color coding frequency within quantile: 1 -> yellow, 2 -> orange, 3 -> red, 4 -> white

```
In [8]: m = Basemap(projection='mill',llcrnrlat=-80,
                    urcrnrlat=80, llcrnrlon=-180,urcrnrlon=180,
                    lat_ts=20,resolution='c')

        plt.figure(figsize=(12,10))
        plt.title("Earthquakes")
        m.fillcontinents(color='lightgreen',lake_color='aqua')
        m.drawmapboundary(fill_color='aqua')
        m.drawcoastlines()
        m.drawcountries()

        # Color-code from 1 --> 4 earthquakes in same area w/in 2 decimal places
        colors = ['none', 'yellow', 'orange', 'red', 'white']

        grouped1=grouped.drop_duplicates()

        df4=grouped[grouped['quakes']==0]
        longitudes = (df4['near_lon'] / 100).tolist()
        latitudes = (df4['near_lat'] / 100).tolist()
        x,y = m(longitudes,latitudes)
        m.plot(x, y, "o", markersize = 3, color = 'none')

        df3=grouped[grouped['quakes']==1]
        longitudes = (df3['near_lon'] / 100).tolist()
        latitudes = (df3['near_lat'] / 100).tolist()
        x,y = m(longitudes,latitudes)
        m.plot(x, y, "o", markersize = 3, color = 'yellow')


        df2=grouped[grouped['quakes']==2]
        longitudes = (df2['near_lon'] / 100).tolist()
        latitudes = (df2['near_lat'] / 100).tolist()
        x,y = m(longitudes,latitudes)
        m.plot(x, y, "o", markersize = 4, color = 'orange')

        df1=grouped[grouped['quakes']==3]
```

```
longitudes = (df1['near_lon'] / 100).tolist()
latitudes = (df1['near_lat'] / 100).tolist()
x,y = m(longitudes,latitudes)
m.plot(x, y, "o", markersize = 5, color = 'red')


df0=grouped[grouped['quakes']==4]

longitudes = (df0['near_lon'] / 100).tolist()
latitudes = (df0['near_lat'] / 100).tolist()
x,y = m(longitudes,latitudes)
m.plot(x, y, "o", markersize = 10, color = 'white')


plt.show()

'''
counts = (grouped['quakes']).tolist()

most = max(counts)
print(counts)

plt.figure(figsize=(12,10))
plt.title("Earthquakes")
m.fillcontinents(color='lightgreen',lake_color='aqua')
m.drawmapboundary(fill_color='aqua')
m.drawcoastlines()
m.drawcountries()

# Color-code from 1 --> 4 earthquakes in same area w/in 2 decimal places
colors = ['none', 'yellow', 'orange', 'red', 'white']

# TODO:
# Instead of plotting everything with the same color -- compute subsets
# of data with 1, 2, .., 4 different quakes, and plot with colors from
# the colors list above
longitudes = (grouped['near_lon'] / 100).tolist()
latitudes = (grouped['near_lat'] / 100).tolist()
x,y = m(longitudes,latitudes)
m.plot(x, y, "o", markersize = 3, color = 'yellow')
'''
```




Out[8]: '\ncounts = (grouped[\'quakes\']).tolist()\n\nmost = max(counts)\nprint(counts)\n\nplt.f

## 3 Step 2.3 Find Locations with Multiple Quakes in a Month

In [9]:
```
# TODO:
# compute DataFrame month_apart with pairs of earthquakes
# in the same (to 2 decimal places) location, which occur
# within a month of each other
quake_df
quake_df1=quake_df
from datetime import date
from dateutil.relativedelta import relativedelta


#quake_df=quake_df.merge(quake_df, left_on=['near_lat'],right_on=['near_lon'])
#quake_df1=pd.merge(quake_df1, quake_df1, left_on='near_lat', right_on='near_lon')
quake_df1=pd.merge(quake_df1, quake_df1, on=['near_lat','near_lon'])
#quake_df1=quake_df1[quake_df1['Date_y']!=quake_df1['Date_x']]
#quake_df1=quake_df1[quake_df1['near_lat_x']!=quake_df1['near_lat_y']]
#quake_df1=quake_df1[quake_df1['near_lon_x']!=quake_df1['near_lon_y']]
quake_df1=quake_df1[quake_df1['Date_x'] < quake_df1['Date_y']]
quake_df1['NEW_DATE'] = quake_df1["Date_x"].apply(lambda x: x + relativedelta(months=1))
#month_apart=quake_df1=quake_df1[quake_df1['Date_y'] <= (quake_df1['Date_x']+ relativede
month_apart=quake_df1[quake_df1['Date_y'] <= quake_df1['NEW_DATE']]
month_apart.head(20)
```

Out[9]:

| | Date_x | Latitude_x | Longitude_x | Magnitude_x | Depth_x | Type_x |
|---|---|---|---|---|---|---|
| 7165 | 1984-09-18 | 34.006 | 141.500 | 6.8 | 47.6 | Earthquake |
| 7738 | 1985-11-17 | -1.639 | 134.911 | 7.1 | 10.0 | Earthquake |
| 12585 | 1995-07-30 | -23.230 | -70.676 | 5.8 | 33.0 | Earthquake |
| 15372 | 2001-04-01 | -34.409 | 55.464 | 5.5 | 10.0 | Earthquake |
| 16607 | 2003-12-25 | -22.305 | 169.531 | 6.0 | 10.0 | Earthquake |
| 17473 | 2005-06-10 | 1.813 | 97.089 | 5.5 | 25.0 | Earthquake |
| 19125 | 2008-07-23 | 32.752 | 105.498 | 5.5 | 4.0 | Earthquake |
| 20693 | 2011-03-15 | 35.209 | 140.994 | 5.7 | 19.4 | Earthquake |
| 21778 | 2013-05-19 | 52.340 | 160.065 | 6.1 | 18.0 | Earthquake |

| | near_lat | near_lon | Date_y | Latitude_y | Longitude_y | Magnitude_y |
|---|---|---|---|---|---|---|
| 7165 | 3400 | 14150 | 1984-09-21 | 34.003 | 141.507 | 5.7 |
| 7738 | -163 | 13491 | 1985-12-06 | -1.636 | 134.910 | 6.0 |
| 12585 | -2323 | -7067 | 1995-08-02 | -23.230 | -70.677 | 6.0 |
| 15372 | -3440 | 5546 | 2001-04-04 | -34.405 | 55.464 | 5.5 |
| 16607 | -2230 | 16953 | 2004-01-03 | -22.300 | 169.535 | 5.5 |
| 17473 | 181 | 9708 | 2005-07-05 | 1.819 | 97.082 | 6.7 |
| 19125 | 3275 | 10549 | 2008-08-05 | 32.756 | 105.494 | 6.0 |
| 20693 | 3520 | 14099 | 2011-03-22 | 35.205 | 140.997 | 5.7 |
| 21778 | 5234 | 16006 | 2013-05-21 | 52.346 | 160.063 | 5.7 |

| | Depth_y | Type_y | NEW_DATE |
|---|---|---|---|
| 7165 | 40.2 | Earthquake | 1984-10-18 |
| 7738 | 24.5 | Earthquake | 1985-12-17 |
| 12585 | 33.0 | Earthquake | 1995-08-30 |
| 15372 | 10.0 | Earthquake | 2001-05-01 |
| 16607 | 10.0 | Earthquake | 2004-01-25 |
| 17473 | 21.0 | Earthquake | 2005-07-10 |
| 19125 | 6.0 | Earthquake | 2008-08-23 |
| 20693 | 20.0 | Earthquake | 2011-04-15 |
| 21778 | 17.8 | Earthquake | 2013-06-19 |

# 4 Step 2.3 Plot Multiple Quakes over Prior Map

In [10]:
```
# TODO: reproduce plot from 2.2.
# Overlay points from month_apart
# Use "o", markersize = 15, color = 'black'
m = Basemap(projection='mill',llcrnrlat=-80,
            urcrnrlat=80, llcrnrlon=-180,urcrnrlon=180,
            lat_ts=20,resolution='c')

plt.figure(figsize=(12,10))
plt.title("Earthquakes")
m.fillcontinents(color='lightgreen',lake_color='aqua')
m.drawmapboundary(fill_color='aqua')
m.drawcoastlines()
```

```
m.drawcountries()


colors = ['none', 'yellow', 'orange', 'red', 'white']

grouped1=grouped.drop_duplicates()

df4=grouped[grouped['quakes']==0]
longitudes = (df4['near_lon'] / 100).tolist()
latitudes = (df4['near_lat'] / 100).tolist()
x,y = m(longitudes,latitudes)
m.plot(x, y, "o", markersize = 3, color = 'none')

df3=grouped[grouped['quakes']==1]
longitudes = (df3['near_lon'] / 100).tolist()
latitudes = (df3['near_lat'] / 100).tolist()
x,y = m(longitudes,latitudes)
m.plot(x, y, "o", markersize = 3, color = 'yellow')


df2=grouped[grouped['quakes']==2]
longitudes = (df2['near_lon'] / 100).tolist()
latitudes = (df2['near_lat'] / 100).tolist()
x,y = m(longitudes,latitudes)
m.plot(x, y, "o", markersize = 4, color = 'orange')

df1=grouped[grouped['quakes']==3]

longitudes = (df1['near_lon'] / 100).tolist()
latitudes = (df1['near_lat'] / 100).tolist()
x,y = m(longitudes,latitudes)
m.plot(x, y, "o", markersize = 5, color = 'red')


df0=grouped[grouped['quakes']==4]

longitudes = (df0['near_lon'] / 100).tolist()
latitudes = (df0['near_lat'] / 100).tolist()
x,y = m(longitudes,latitudes)
m.plot(x, y, "o", markersize = 10, color = 'white')

# Color-code from 1 --> 4 earthquakes in same area w/in 2 decimal places


longitudes = (month_apart['near_lon'] / 100).tolist()
latitudes = (month_apart['near_lat'] / 100).tolist()
x,y = m(longitudes,latitudes)
```

```
m.plot(x, y, "o", markersize = 15, color = 'black')
plt.show()
```