Earthquake

August 19, 2019

1 Analysis of Earthquake Frequency

1.0.1 By Royce Schultz

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.optimize import curve_fit
from scipy.special import gamma

from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()
```

[2]:
```
Fields = ['Date','Time','Latitude','Longitude','Type','Magnitude'] # Only import␣
↪used columns
df = pd.read_csv('database.csv', usecols=Fields)
df.head()
```

[2]:

	Date	Time	Latitude	Longitude	Type	Magnitude
0	01/02/1965	13:44:18	19.246	145.616	Earthquake	6.0
1	01/04/1965	11:29:49	1.863	127.352	Earthquake	5.8
2	01/05/1965	18:05:58	-20.579	-173.972	Earthquake	6.2
3	01/08/1965	18:49:43	-59.076	-23.557	Earthquake	5.8
4	01/09/1965	13:32:50	11.938	126.427	Earthquake	5.8

[3]:
```
# Cleaning
df = df[df.Date.str.len() < 15] # Removes 3 rows with malformed dates
df = df[df.Type.isin(['Earthquake'])] # removes explosions and rock bursts
```

[4]:
```
# Calculations
df['Datetime'] = pd.to_datetime(df.Date +' '+ df.Time)
df['Year'] = df['Datetime'].map(lambda x: x.year)

df['Rounded_Magnitude'] = np.floor(df.Magnitude)

df['Last_Quake'] = df.Datetime.diff()
```

```python
df = df[df['Last_Quake'].notna()]
df['Last_Quake_sec'] = df['Last_Quake'].map(lambda x: x.total_seconds()) # Last␣
↪Quake (s)

df.reset_index(drop=True)

df.head()
```

[4]:

```
         Date      Time  Latitude  Longitude        Type  Magnitude  \
1  01/04/1965  11:29:49     1.863    127.352  Earthquake        5.8
2  01/05/1965  18:05:58   -20.579   -173.972  Earthquake        6.2
3  01/08/1965  18:49:43   -59.076    -23.557  Earthquake        5.8
4  01/09/1965  13:32:50    11.938    126.427  Earthquake        5.8
5  01/10/1965  13:36:32   -13.405    166.629  Earthquake        6.7

             Datetime  Year  Rounded_Magnitude      Last_Quake  Last_Quake_sec
1 1965-01-04 11:29:49  1965                5.0 1 days 21:45:31        164731.0
2 1965-01-05 18:05:58  1965                6.0 1 days 06:36:09        110169.0
3 1965-01-08 18:49:43  1965                5.0 3 days 00:43:45        261825.0
4 1965-01-09 13:32:50  1965                5.0 0 days 18:43:07         67387.0
5 1965-01-10 13:36:32  1965                6.0 1 days 00:03:42         86622.0
```

[22]:

```python
df.describe()
```

[22]:

```
           Latitude     Longitude     Magnitude          Year  \
count  23228.000000  23228.000000  23228.000000  23228.000000
mean       1.385304     39.738244      5.882785   1992.719520
std       29.929647    125.755664      0.424059     14.437895
min      -77.080000   -179.997000      5.500000   1965.000000
25%      -18.719500    -76.384500      5.600000   1981.000000
50%       -3.684450    106.307500      5.700000   1994.000000
75%       24.968500    145.290250      6.000000   2005.000000
max       86.005000    179.998000      9.100000   2016.000000

       Rounded_Magnitude                 Last_Quake  Last_Quake_sec  \
count       23228.000000                      23228    23228.000000
mean            5.347641  0 days 19:37:17.121146     70637.121147
std             0.545663  0 days 23:24:29.312558     84269.312559
min             5.000000            0 days 00:00:00       0.000000
25%             5.000000            0 days 03:39:22   13162.000000
50%             5.000000            0 days 11:42:44   42164.000000
75%             6.000000  1 days 03:07:29.500000     97649.500000
max             9.000000           10 days 05:30:13  883813.000000

         Dist_Tokyo  Dist_San_Fran   Dist_Denver
count  23228.000000   23228.000000  23228.000000
mean       1.001038       1.319116      1.390059
```

```
std        0.545300     0.351820     0.370957
min        0.000781     0.007909     0.079930
25%        0.632614     1.175661     1.245043
50%        0.911200     1.361461     1.495918
75%        1.377478     1.559398     1.647515
max        1.994588     1.999079     1.999554
```

[25]:
```
len(df)
```

[25]:
```
23228
```

2 About the data

The dataset is available on Kaggle

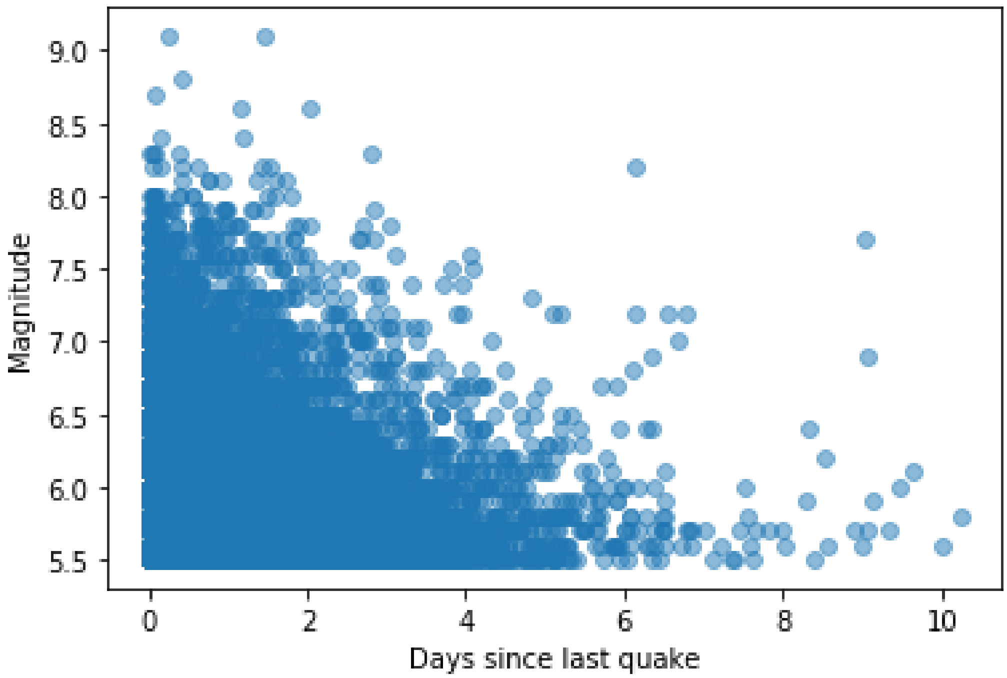
The filtered set contains 23228 quakes with greater than 5.5 magnitude reported between 1965 and 2016.

There are **thousands** of small, unnoticable earthquakes every day that are not contained in this set.

3 Does the strength of an earthquake depend on the time since the last earthquake?

Hypothesis: Assume the earth acts like a spring, constantly storing 'elastic' energy and releasing it in burst that we call earthquakes. Then big earthquakes are the result of a large buildup of energy. If an earthquake has not been recorded for a long time, the probability of a large quake is high.

[5]:
```
plt.scatter(df.Last_Quake_sec/(60*60*24),df.Magnitude, alpha=0.5)

plt.xlabel('Days since last quake')
plt.ylabel('Magnitude')
plt.show()
```



4 Big quakes are preceeded shortly by other quakes

The hypothesis is **not supported**. In fact, the longer time without a quake, the higher probability that the next quake will be small.

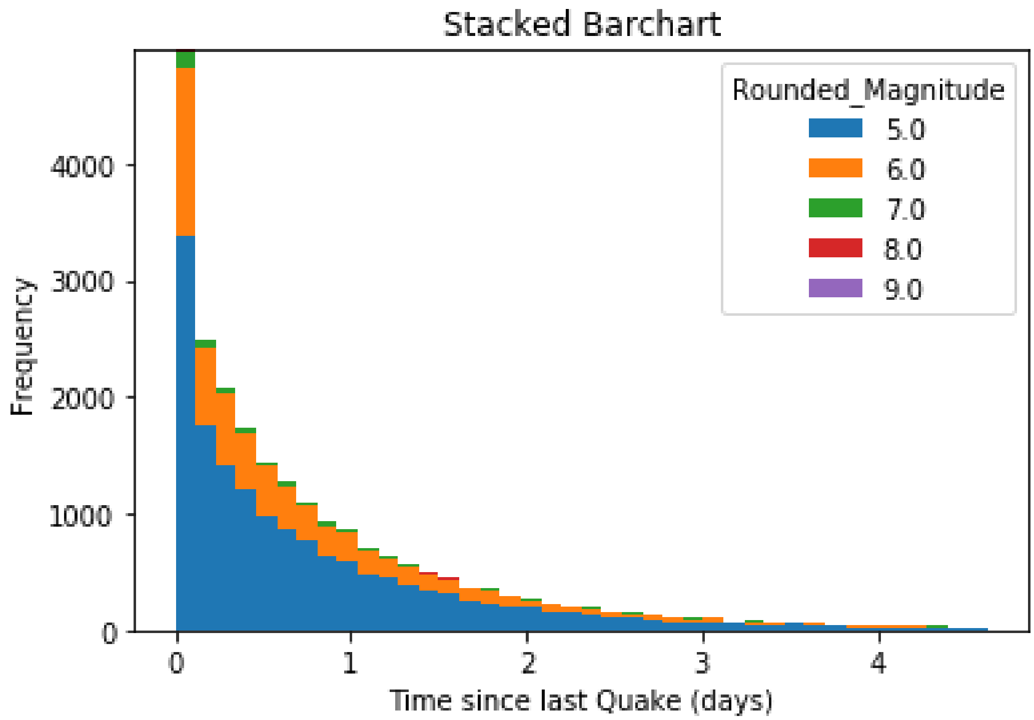
However, this graph does not account for other factors like location and therefore the hypothesis cannot be determined to be inncorrect.

5 Earthquakes are random

Even leading science cannot precisly predict earthquakes, however the frequency of earthquakes can still be modeled statistically. The following figure shows a histogram chart of the time between each recorded quake.

[6]:
```
(df[df.Last_Quake_sec < 4*10**5].
 ↪pivot(columns='Rounded_Magnitude')['Last_Quake_sec']/(60*60*24)).plot(kind =␣
 ↪'hist', stacked=True,bins=40)

plt.title('Stacked Barchart')
plt.xlabel('Time since last Quake (days)')
```

[6]:
```
Text(0.5, 0, 'Time since last Quake (days)')
```



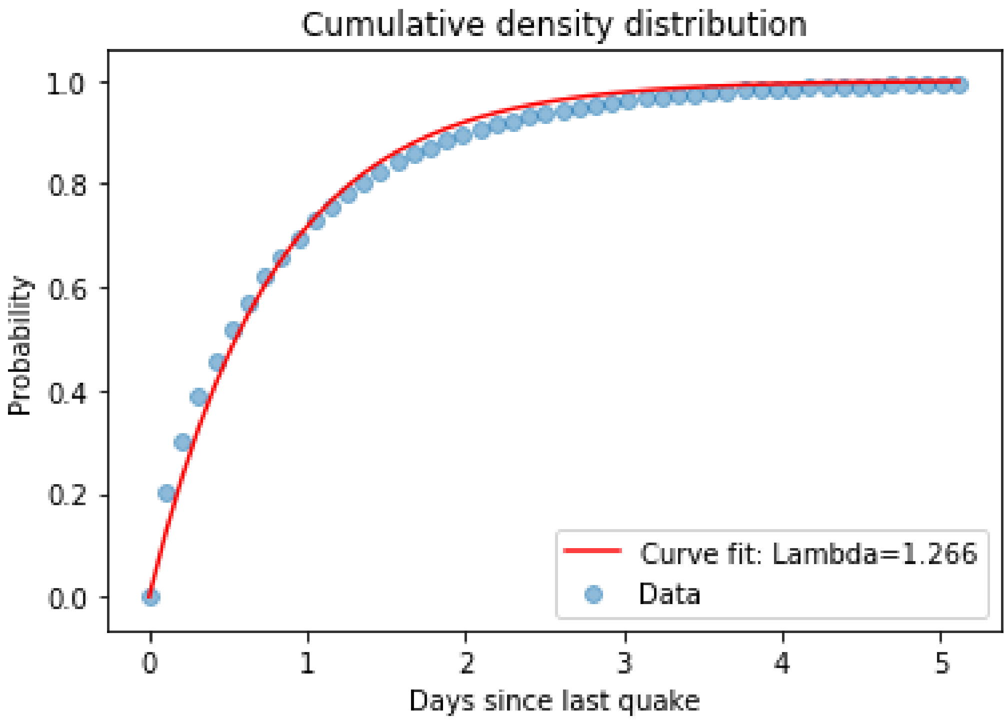
6 The Exponential Distribution

This shape should look familiar to any engineer. It is a decaying exponential. The exponential is a common and well understood distribution. The cumulative and marginal distributions are defined as such,

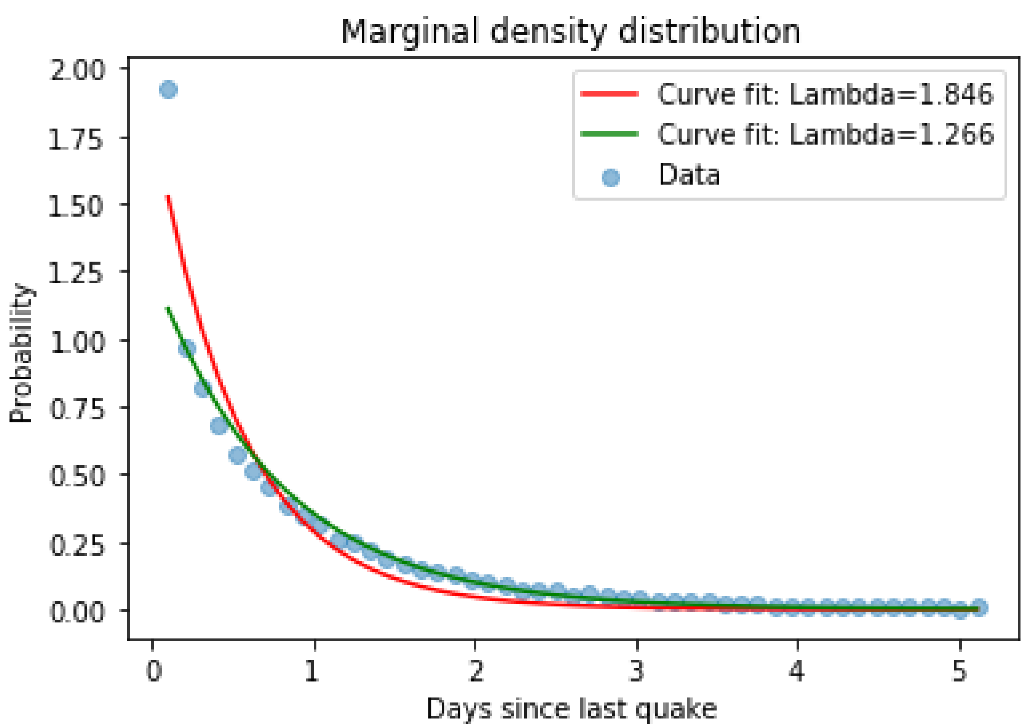
[7]:
```
def cumulative_density(x, lamb):
    return 1 - np.exp(-lamb * x)

def marginal_density(x, lamb):
    return lamb * np.exp(-lamb * x)
```

[20]:
```
# Fit functions to observed data

density = 50 # Plot resolution

# Cumulative Distribution
max_x = df.Last_Quake_sec.max() / (24*60*60)
x = np.linspace(0,.5*max_x,density)
y = np.linspace(0,0,density)
for i in range(density):
    y[i] = len(df[df.Last_Quake_sec < 24*60*60*x[i]]) / len(df)
```

```
plt.scatter(x,y, alpha=0.5, label='Data')

popt, pcov = curve_fit(cumulative_density, x, y)
plt.plot(x, cumulative_density(x, *popt), 'r-',label='Curve fit: Lambda=%5.3f' %␣
↪tuple(popt))

plt.title('Cumulative density distribution')
plt.xlabel('Days since last quake')
plt.ylabel('Probability')
plt.legend()

#Marginal Distribution
plt.figure()
dx = np.diff(x)
dy = np.diff(y)
new_x = x[1:]

plt.scatter(new_x,dy/dx, alpha=0.5, label='Data')

popt, pcov = curve_fit(marginal_density, new_x, dy/dx)
plt.plot(new_x, marginal_density(new_x, *popt), 'r-',label='Curve fit: Lambda=%5.
↪3f' % tuple(popt))
plt.plot(new_x, marginal_density(new_x, 1.266), 'g-',label='Curve fit: Lambda=1.
↪266')

plt.title('Marginal density distribution')
plt.xlabel('Days since last quake')
plt.ylabel('Probability')
plt.legend()

plt.show()
```



Cumulative density distribution
1.0
0.8
0.6
0.4
0.2
0.0
Probability
0
1
2
3
4
5
Days since last quake
Curve fit: Lambda=1.266
Data



Marginal density distribution
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
Probability
0
1
2
3
4
5
Days since last quake
Curve fit: Lambda=1.846
Curve fit: Lambda=1.266
Data

The exponential distribution has the propterty,

$$mean = \frac{1}{\lambda} \tag{1}$$
$$variance = \frac{1}{\lambda} \tag{2}$$

So, **on average** earthquakes occour every 0.78 days.

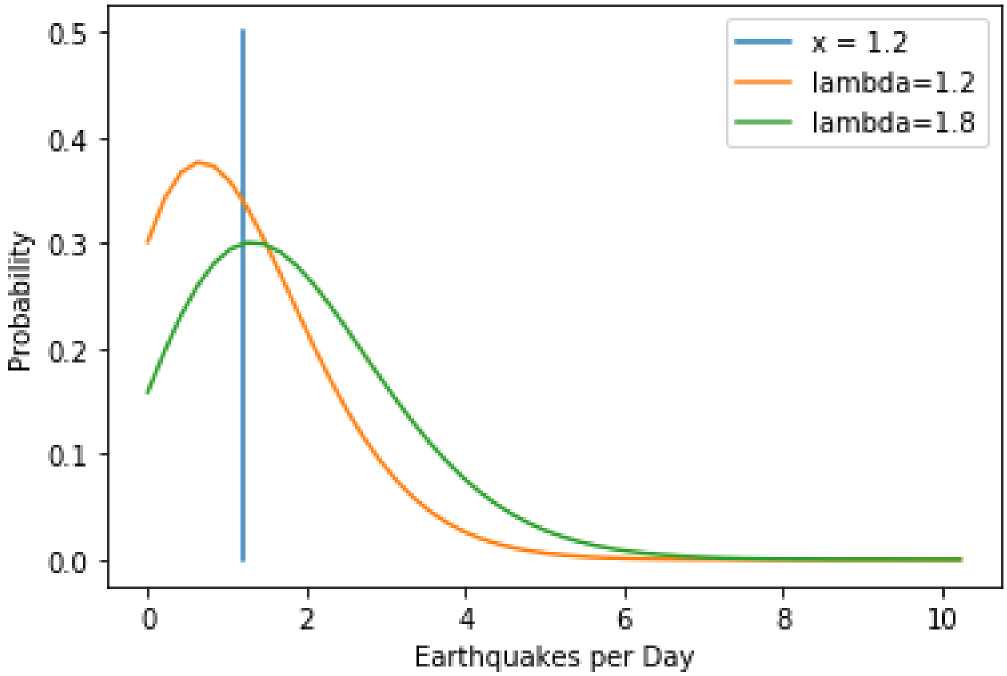
This exponential distribution models the time **betweeen** events, however it is often more useful to predict the number of events in a time period. This is modeled using a sum of exponential distributions that shall be left to the reader to prove. This distribution is commonly refered to as the Poisson Distribution.

7 The Poisson Distribution

[9]:
```
def pois_marginal_density(x,lamb):
    a = lamb**x
    b = np.exp(-lamb)
    c = np.zeros(len(x))
    c = gamma(x+1)
    return (a * b) / c
```

[28]:
```
lamb = 1.2
x = np.linspace(0,max_x)
plt.plot([lamb,lamb],[0,.5], label='x = 1.2') # Mark mean
plt.plot(x,pois_marginal_density(x,lamb),label = 'lambda=1.2')

lamb = 1.84
plt.plot(x,pois_marginal_density(x,lamb),label='lambda=1.8')

plt.xlabel('Earthquakes per Day')
plt.ylabel('Probability')
plt.legend()
```

[28]:
```
<matplotlib.legend.Legend at 0x127f09b00>
```



The Poisson distribution has the propterty,

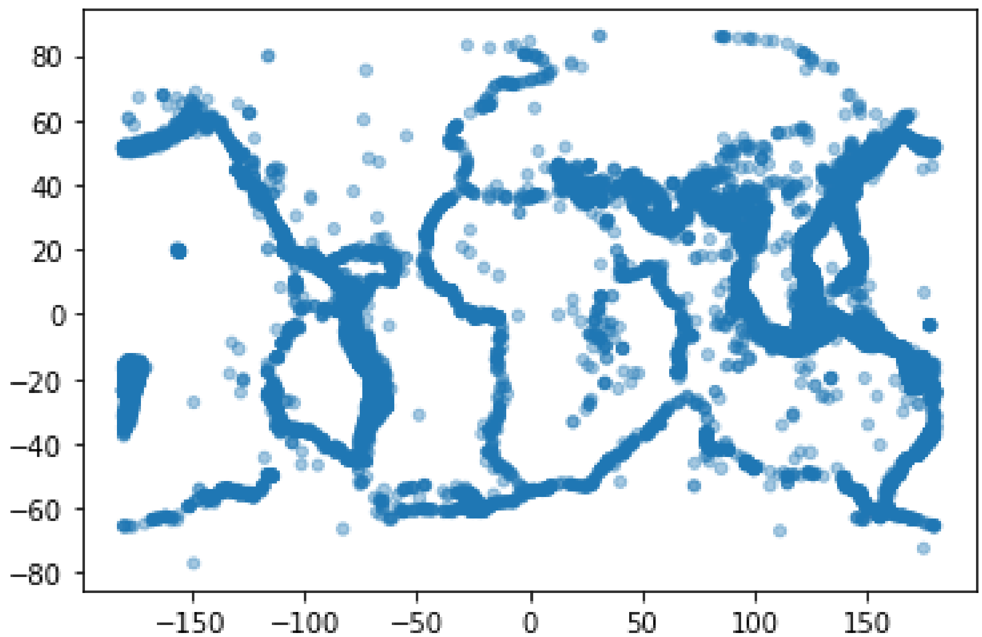
$$mean = \lambda \tag{3}$$
$$variance = \lambda \tag{4}$$

So, **on average** earth will have 1.2 eathquakes per day

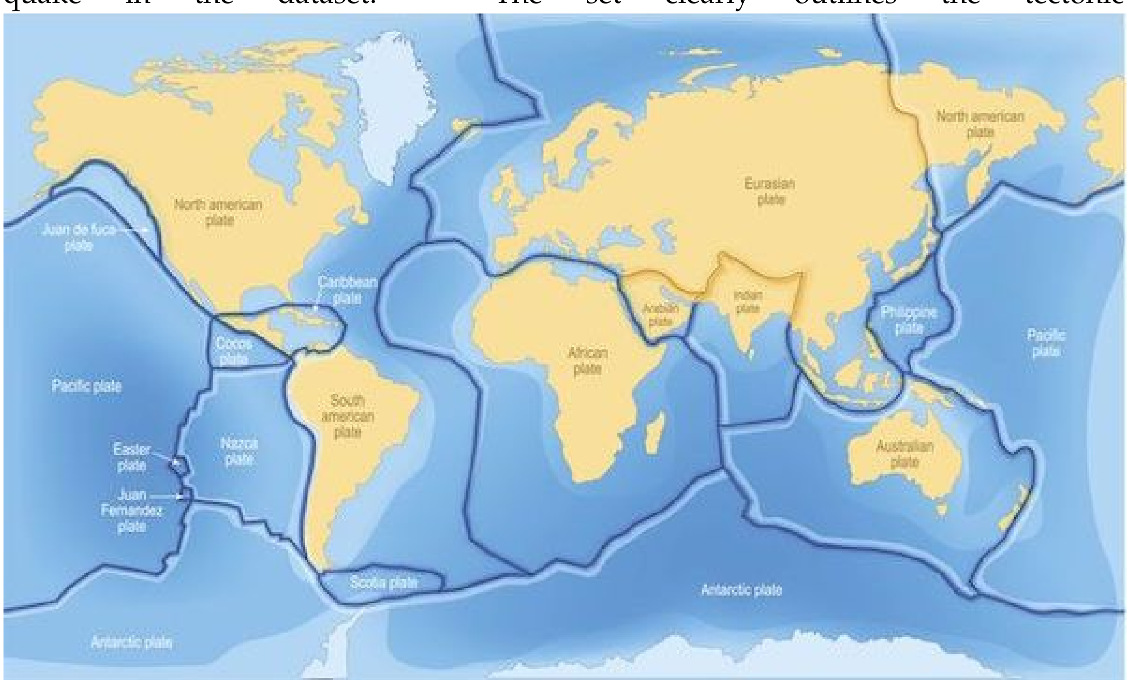
8 Are earthquakes uniformily distributed around the globe?

[11]:
```
plt.scatter(df.Longitude.values,df.Latitude.values, alpha=0.4, s=3*df.Magnitude.
  ↪values)
```

[11]: <matplotlib.collections.PathCollection at 0x126c6dc50>



This scatterplot shows the Latitude and Longitude coordinates of each quake in the dataset. The set clearly outlines the tectonic plates.



9 Which city has greater danger of earthquakes: San Francisco or Tokyo?

This requires aggregating data by distance from a reference point. Some functions need to be defined first.

9.1 About the distance Funciton

Subtracting latitude and longitude of a quake will not return the distance. This is the same effect that makes Greenland appear nearly as large as Africa on a 2d map.

Instead Lat/Long points in spherical space are converted to an x,y,z triple in cartesian space. Then the euclidian distance between quakes and the reference point can be accuratly calculated.

This method should more accuratly assess the proximity of the earthquake since the energy waves travel in a (nearly) straight line through the crust, not along the surface. It should be noted that this does not apply to earthquakes on the opposite side of the earth as the waves do not travel through earths molten core.

[12]:
```python
# Description:
#   Determines Euclidian (straight line) between 2 points. Does not consider arc␣
↪length, just straight distance
#   Assumes earth is a sphere with radius=1
#   So the poles are 2 units apart, the equator is sqrt(2) from each pole, not␣
↪pi and pi/2
# Input:
#   Dataframe with Latitude and Longitude components
#   2 arguments for Lat and Long of reference point
# Output:
#   Series containing distances to the reference point

def distance_from(df, Lat, Long):

    Lat = np.deg2rad(Lat) # Convert degrees to radians for numpy trig
    Long = np.deg2rad(Long)
    x_pos = np.cos(Lat)*np.sin(Long) # Convert spherical coordinates to cartesian
    y_pos = np.cos(Lat)*np.cos(Long) # Assumes earths radius = 1
    z_pos = np.sin(Lat)

    data_Latitude_rad = np.deg2rad(df.Latitude) # Convert dataframe to radians
    data_Longitude_rad = np.deg2rad(df.Longitude)
    data_x_pos = np.cos(data_Latitude_rad)*np.sin(data_Longitude_rad) # Convert␣
↪to spherical
    data_y_pos = np.cos(data_Latitude_rad)*np.cos(data_Longitude_rad)
    data_z_pos = np.sin(data_Latitude_rad)
    return ( (data_x_pos - x_pos)**2 + (data_y_pos - y_pos)**2 + (data_z_pos -␣
↪z_pos)**2)**(1/2) # Pythagoras3D
```

[13]:
```python
Tokyo = [35.67,139.65]
San_Fran = [37.77,-122.42]
Denver = [39.74,-104.99]

df['Dist_Tokyo'] = distance_from(df,Tokyo[0],Tokyo[1])
df['Dist_San_Fran'] = distance_from(df,San_Fran[0],San_Fran[1])
df['Dist_Denver'] = distance_from(df,Denver[0],Denver[1])

df.reset_index(drop=True)
```

[13]:
```
             Date      Time  Latitude  Longitude        Type  Magnitude  \
0      01/04/1965  11:29:49    1.8630   127.3520  Earthquake        5.8
1      01/05/1965  18:05:58  -20.5790  -173.9720  Earthquake        6.2
2      01/08/1965  18:49:43  -59.0760   -23.5570  Earthquake        5.8
3      01/09/1965  13:32:50   11.9380   126.4270  Earthquake        5.8
4      01/10/1965  13:36:32  -13.4050   166.6290  Earthquake        6.7
5      01/12/1965  13:32:25   27.3570    87.8670  Earthquake        5.9
6      01/15/1965  23:17:42  -13.3090   166.2120  Earthquake        6.0
7      01/16/1965  11:32:37  -56.4520   -27.0430  Earthquake        6.0
8      01/17/1965  10:43:17  -24.5630   178.4870  Earthquake        5.8
9      01/17/1965  20:57:41   -6.8070   108.9880  Earthquake        5.9
10     01/24/1965  00:11:17   -2.6080   125.9520  Earthquake        8.2
11     01/29/1965  09:35:30   54.6360   161.7030  Earthquake        5.5
12     02/01/1965  05:27:06  -18.6970  -177.8640  Earthquake        5.6
13     02/02/1965  15:56:51   37.5230    73.2510  Earthquake        6.0
14     02/04/1965  03:25:00  -51.8400   139.7410  Earthquake        6.1
15     02/04/1965  05:01:22   51.2510   178.7150  Earthquake        8.7
16     02/04/1965  06:04:59   51.6390   175.0550  Earthquake        6.0
17     02/04/1965  06:37:06   52.5280   172.0070  Earthquake        5.7
18     02/04/1965  06:39:32   51.6260   175.7460  Earthquake        5.8
19     02/04/1965  07:11:23   51.0370   177.8480  Earthquake        5.9
20     02/04/1965  07:14:59   51.7300   173.9750  Earthquake        5.9
21     02/04/1965  07:23:12   51.7750   173.0580  Earthquake        5.7
22     02/04/1965  07:43:43   52.6110   172.5880  Earthquake        5.7
23     02/04/1965  08:06:17   51.8310   174.3680  Earthquake        5.7
24     02/04/1965  08:33:41   51.9480   173.9690  Earthquake        5.6
25     02/04/1965  08:40:44   51.4430   179.6050  Earthquake        7.3
26     02/04/1965  12:06:08   52.7730   171.9740  Earthquake        6.5
27     02/04/1965  12:50:59   51.7720   174.6960  Earthquake        5.6
28     02/04/1965  14:18:29   52.9750   171.0910  Earthquake        6.4
29     02/04/1965  15:51:25   52.9900   170.8740  Earthquake        5.8
...           ...       ...       ...        ...         ...        ...
23198  12/11/2016  17:26:10  -10.9640   161.5723  Earthquake        5.5
23199  12/14/2016  02:01:23   21.2897   144.4037  Earthquake        6.0
23200  12/14/2016  21:14:56   21.3697   144.2175  Earthquake        5.5
23201  12/16/2016  11:34:58   14.0882   -90.8691  Earthquake        5.5
23202  12/17/2016  10:51:10   -4.5049   153.5216  Earthquake        7.9
```

```
23203  12/17/2016  11:22:40   -4.4244   153.5419  Earthquake        5.6
23204  12/17/2016  11:27:39   -5.6497   153.9975  Earthquake        6.3
23205  12/18/2016  05:46:25  -10.2137   161.2177  Earthquake        5.9
23206  12/18/2016  06:15:46  -34.9886  -107.8694  Earthquake        5.5
23207  12/18/2016  06:39:42   -6.3046   154.3530  Earthquake        5.9
23208  12/18/2016  09:47:05    8.3489   137.6672  Earthquake        6.2
23209  12/18/2016  11:35:48  -10.1904   161.2187  Earthquake        5.5
23210  12/18/2016  13:30:11   -9.9640   -70.9714  Earthquake        6.4
23211  12/20/2016  04:21:29  -10.1773   161.2236  Earthquake        6.4
23212  12/20/2016  10:04:39   37.1442    84.9583  Earthquake        5.6
23213  12/20/2016  12:33:14  -10.1785   160.9149  Earthquake        6.0
23214  12/20/2016  20:07:53  -10.1549   160.7816  Earthquake        5.5
23215  12/21/2016  00:17:15   -7.5082   127.9206  Earthquake        6.7
23216  12/21/2016  16:43:57   21.5036   145.4172  Earthquake        5.9
23217  12/24/2016  01:32:16   -5.2453   153.5754  Earthquake        6.0
23218  12/24/2016  03:58:55   -5.1460   153.5166  Earthquake        5.8
23219  12/25/2016  14:22:27  -43.4029   -73.9395  Earthquake        7.6
23220  12/25/2016  14:32:13  -43.4810   -74.4771  Earthquake        5.6
23221  12/27/2016  23:20:56   45.7192    26.5230  Earthquake        5.6
23222  12/28/2016  08:18:01   38.3754  -118.8977  Earthquake        5.6
23223  12/28/2016  08:22:12   38.3917  -118.8941  Earthquake        5.6
23224  12/28/2016  09:13:47   38.3777  -118.8957  Earthquake        5.5
23225  12/28/2016  12:38:51   36.9179   140.4262  Earthquake        5.9
23226  12/29/2016  22:30:19   -9.0283   118.6639  Earthquake        6.3
23227  12/30/2016  20:08:28   37.3973   141.4103  Earthquake        5.5

                 Datetime  Year  Rounded_Magnitude      Last_Quake  \
0     1965-01-04 11:29:49  1965                5.0 1 days 21:45:31
1     1965-01-05 18:05:58  1965                6.0 1 days 06:36:09
2     1965-01-08 18:49:43  1965                5.0 3 days 00:43:45
3     1965-01-09 13:32:50  1965                5.0 0 days 18:43:07
4     1965-01-10 13:36:32  1965                6.0 1 days 00:03:42
5     1965-01-12 13:32:25  1965                5.0 1 days 23:55:53
6     1965-01-15 23:17:42  1965                6.0 3 days 09:45:17
7     1965-01-16 11:32:37  1965                6.0 0 days 12:14:55
8     1965-01-17 10:43:17  1965                5.0 0 days 23:10:40
9     1965-01-17 20:57:41  1965                5.0 0 days 10:14:24
10    1965-01-24 00:11:17  1965                8.0 6 days 03:13:36
11    1965-01-29 09:35:30  1965                5.0 5 days 09:24:13
12    1965-02-01 05:27:06  1965                5.0 2 days 19:51:36
13    1965-02-02 15:56:51  1965                6.0 1 days 10:29:45
14    1965-02-04 03:25:00  1965                6.0 1 days 11:28:09
15    1965-02-04 05:01:22  1965                8.0 0 days 01:36:22
16    1965-02-04 06:04:59  1965                6.0 0 days 01:03:37
17    1965-02-04 06:37:06  1965                5.0 0 days 00:32:07
18    1965-02-04 06:39:32  1965                5.0 0 days 00:02:26
19    1965-02-04 07:11:23  1965                5.0 0 days 00:31:51
```

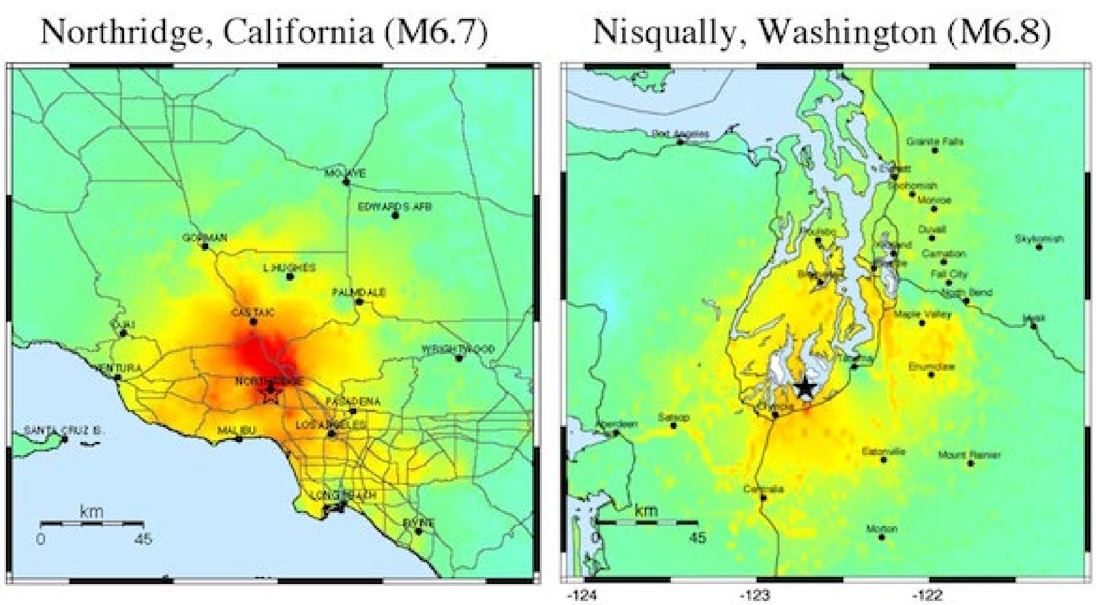
```
20    1965-02-04 07:14:59  1965               5.0 0 days 00:03:36
21    1965-02-04 07:23:12  1965               5.0 0 days 00:08:13
22    1965-02-04 07:43:43  1965               5.0 0 days 00:20:31
23    1965-02-04 08:06:17  1965               5.0 0 days 00:22:34
24    1965-02-04 08:33:41  1965               5.0 0 days 00:27:24
25    1965-02-04 08:40:44  1965               7.0 0 days 00:07:03
26    1965-02-04 12:06:08  1965               6.0 0 days 03:25:24
27    1965-02-04 12:50:59  1965               5.0 0 days 00:44:51
28    1965-02-04 14:18:29  1965               6.0 0 days 01:27:30
29    1965-02-04 15:51:25  1965               5.0 0 days 01:32:56
...                   ...   ...               ...             ...
23198 2016-12-11 17:26:10  2016               5.0 0 days 02:52:57
23199 2016-12-14 02:01:23  2016               6.0 2 days 08:35:13
23200 2016-12-14 21:14:56  2016               5.0 0 days 19:13:33
23201 2016-12-16 11:34:58  2016               5.0 1 days 14:20:02
23202 2016-12-17 10:51:10  2016               7.0 0 days 23:16:12
23203 2016-12-17 11:22:40  2016               5.0 0 days 00:31:30
23204 2016-12-17 11:27:39  2016               6.0 0 days 00:04:59
23205 2016-12-18 05:46:25  2016               5.0 0 days 18:18:46
23206 2016-12-18 06:15:46  2016               5.0 0 days 00:29:21
23207 2016-12-18 06:39:42  2016               5.0 0 days 00:23:56
23208 2016-12-18 09:47:05  2016               6.0 0 days 03:07:23
23209 2016-12-18 11:35:48  2016               5.0 0 days 01:48:43
23210 2016-12-18 13:30:11  2016               6.0 0 days 01:54:23
23211 2016-12-20 04:21:29  2016               6.0 1 days 14:51:18
23212 2016-12-20 10:04:39  2016               5.0 0 days 05:43:10
23213 2016-12-20 12:33:14  2016               6.0 0 days 02:28:35
23214 2016-12-20 20:07:53  2016               5.0 0 days 07:34:39
23215 2016-12-21 00:17:15  2016               6.0 0 days 04:09:22
23216 2016-12-21 16:43:57  2016               5.0 0 days 16:26:42
23217 2016-12-24 01:32:16  2016               6.0 2 days 08:48:19
23218 2016-12-24 03:58:55  2016               5.0 0 days 02:26:39
23219 2016-12-25 14:22:27  2016               7.0 1 days 10:23:32
23220 2016-12-25 14:32:13  2016               5.0 0 days 00:09:46
23221 2016-12-27 23:20:56  2016               5.0 2 days 08:48:43
23222 2016-12-28 08:18:01  2016               5.0 0 days 08:57:05
23223 2016-12-28 08:22:12  2016               5.0 0 days 00:04:11
23224 2016-12-28 09:13:47  2016               5.0 0 days 00:51:35
23225 2016-12-28 12:38:51  2016               5.0 0 days 03:25:04
23226 2016-12-29 22:30:19  2016               6.0 1 days 09:51:28
23227 2016-12-30 20:08:28  2016               5.0 0 days 21:38:09

       Last_Quake_sec  Dist_Tokyo  Dist_San_Fran  Dist_Denver
0            164731.0    0.612724       1.583197     1.702209
1            110169.0    1.166416       1.228930     1.390338
2            261825.0    1.949312       1.782142     1.726013
3             67387.0    0.459641       1.518147     1.635202
```

4	86622.0	0.928365	1.334940	1.501385
5	172553.0	0.755869	1.627749	1.656532
6	294317.0	0.924238	1.338082	1.504309
7	44095.0	1.961063	1.761478	1.699456
8	83440.0	1.154873	1.330660	1.485102
9	36864.0	0.866291	1.767595	1.848748
10	530016.0	0.690026	1.624143	1.739604
11	465853.0	0.421162	0.881912	1.004316
12	244296.0	1.113060	1.242286	1.407445
13	124185.0	0.879631	1.568803	1.562171
14	127689.0	1.383153	1.759665	1.846881
15	5782.0	0.548505	0.730065	0.880204
16	3817.0	0.513466	0.766028	0.911562
17	1927.0	0.489313	0.793756	0.933411
18	146.0	0.520272	0.759192	0.905402
19	1911.0	0.538767	0.739203	0.889347
20	216.0	0.503295	0.776502	0.920722
21	493.0	0.494571	0.785413	0.928610
22	1231.0	0.495355	0.787959	0.927875
23	1354.0	0.507736	0.772399	0.916641
24	1644.0	0.504499	0.776061	0.919531
25	423.0	0.558325	0.720784	0.870975
26	12324.0	0.490678	0.793470	0.932259
27	2691.0	0.510636	0.769293	0.914040
28	5250.0	0.483920	0.801393	0.938664
29	5576.0	0.482036	0.803416	0.940431
...	...	...	...	...
23198	10377.0	0.861411	1.362975	1.527651
23199	203713.0	0.260519	1.279393	1.428313
23200	69213.0	0.258411	1.280620	1.429159
23201	138002.0	1.648671	0.628565	0.492128
23202	83772.0	0.720474	1.390349	1.551059
23203	1890.0	0.719315	1.389525	1.550321
23204	299.0	0.740510	1.394542	1.555064
23205	65926.0	0.848365	1.360280	1.525394
23206	1761.0	1.782615	1.203639	1.214417
23207	1436.0	0.752330	1.396137	1.556629
23208	11243.0	0.473356	1.446170	1.585320
23209	6523.0	0.848032	1.360080	1.525220
23210	6863.0	1.891800	1.114246	0.982731
23211	139878.0	0.847871	1.359926	1.525085
23212	20590.0	0.739758	1.542522	1.560579
23213	8915.0	0.846031	1.362930	1.527759
23214	27279.0	0.844890	1.364030	1.528741
23215	14962.0	0.758405	1.639324	1.756864
23216	59202.0	0.261674	1.267470	1.418142
23217	204499.0	0.732254	1.395547	1.555796

```
23218          8799.0    0.730435        1.395356      1.555598
23219        123812.0    1.945431        1.442333      1.386115
23220           586.0    1.943817        1.440310      1.385084
23221        204523.0    1.269100        1.438249      1.340248
23222         32225.0    1.236486        0.049527      0.189505
23223           251.0    1.236390        0.049631      0.189400
23224          3095.0    1.236486        0.049562      0.189470
23225         12304.0    0.024363        1.192306      1.320417
23226        121888.0    0.827526        1.716739      1.816471
23227         77889.0    0.038961        1.179408      1.308657

[23228 rows x 14 columns]
```

10 How far away can you feel a strong quake?

The US Geological Program (USGS) cites that even somewhat large earhquakes dissapate quickly over an area. The tremors can hardly be felt more than just 100km or about 1 degree away.



PERCEIVED SHAKING	Not felt	Weak	Light	Moderate	Strong	Very strong	Severe	Violent	Extreme
POTENTIAL DAMAGE	none	none	none	Very light	Light	Moderate	Moderate/Heavy	Heavy	Very Heavy
PEAK ACC.(%g)	<.17	.17-1.4	1.4-3.9	3.9-9.2	9.2-18	18-34	34-65	65-124	>124
PEAK VEL.(cm/s)	<0.1	0.1-1.1	1.1-3.4	3.4-8.1	8.1-16	16-31	31-60	60-116	>116
INSTRUMENTAL INTENSITY	I	II-III	IV	V	VI	VII	VIII	IX	X+

There are clearly other factors to consider before estimating damages. Soil content can affect energy dissapation and other secondary effects like tsunamis can cause significant damage

11 Calculating distance threshold

More precicely, 100km == 0.899 deg, but lets round up.

[14]:
```python
earth_radius = 6371 # kilometers
earth_circumfrence = earth_radius * 3.14 * 2
km_per_deg = earth_circumfrence / 360
print(100/km_per_deg)
```

```
0.8997777548945409
```

1 degree maps to 0.017 in the distance space.

[15]:
```python
q = pd.DataFrame.from_dict({'Latitude': [0], 'Longitude': [0]})
distance_from(q,0,1)
```

[15]:
```
0    0.017453
dtype: float64
```

For reference, the distance between SF and LA is 0.088

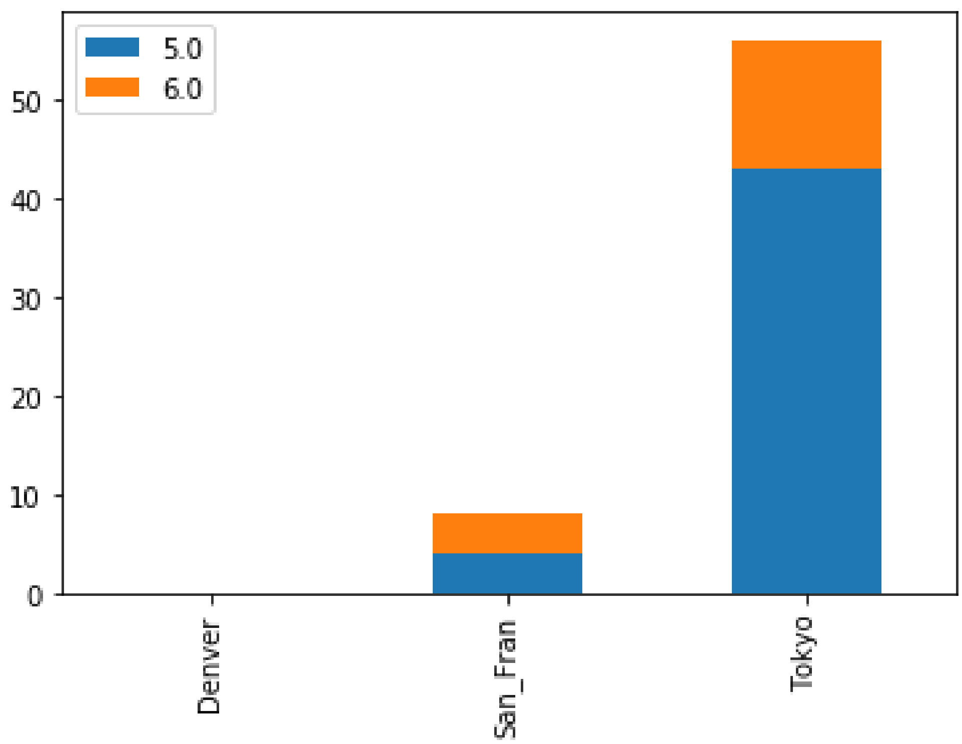
[16]:
```python
print(San_Fran)
q = pd.DataFrame.from_dict({'Latitude': [San_Fran[0]], 'Longitude':␣
↪[San_Fran[1]]})
distance_from(q,34.05,-118.24) # Distance to LA
```

```
[37.77, -122.42]
```

[16]:
```
0    0.08774
dtype: float64
```

[17]:
```python
Local_Dist = .017

Denver_Local_Counts = df[df.Dist_Denver < Local_Dist].Rounded_Magnitude.
↪value_counts()
San_Fran_Local_Counts = df[df.Dist_San_Fran < Local_Dist].Rounded_Magnitude.
↪value_counts()
Tokyo_Local_Counts = df[df.Dist_Tokyo < Local_Dist].Rounded_Magnitude.
↪value_counts()

counts = pd.concat({'Denver':Denver_Local_Counts, 'San_Fran':
↪San_Fran_Local_Counts, 'Tokyo':Tokyo_Local_Counts}, axis = 1).fillna(0)
counts.transpose().plot(kind='bar',stacked=True)
```

[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x126eb1c88>
```



12 Tokyo has significantly more earthquakes than San Francisco or Denver