Earthquakes

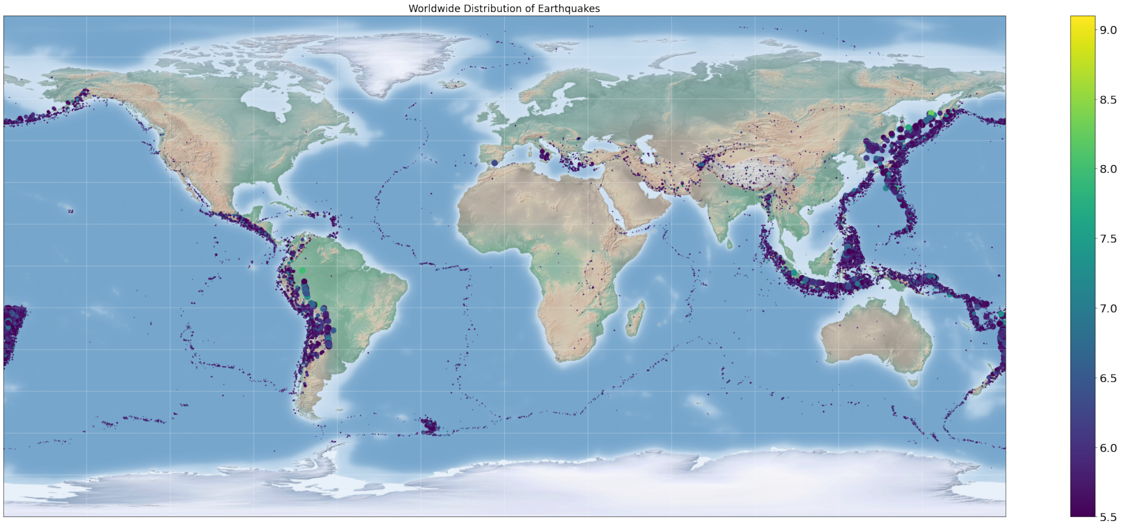
August 18, 2020

[ ]:

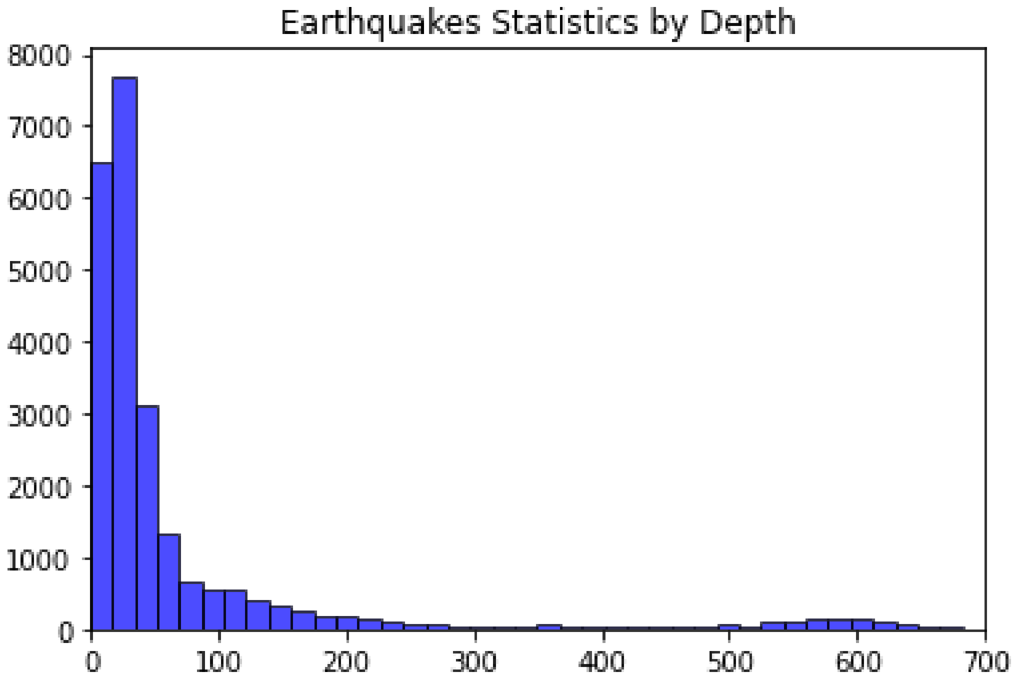
```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
import pandas as pd
from itertools import chain

def draw_map(m, scale=0.2):
    m.shadedrelief(scale=scale)

    lats = m.drawparallels(np.linspace(-90, 90, 13))
    lons = m.drawmeridians(np.linspace(-180, 180, 13))

    lat_lines = chain(*(tup[1][0] for tup in lats.items()))
    lon_lines = chain(*(tup[1][0] for tup in lons.items()))
    all_lines = chain(lat_lines, lon_lines)

    for line in all_lines:
        line.set(linestyle='-', alpha=0.3, color='w')

df = pd.read_csv('database.csv')

fig = plt.figure(figsize = (40, 16))

m = Basemap(projection='cyl', resolution=None,
            llcrnrlat=-90, urcrnrlat=90,
            llcrnrlon=-180, urcrnrlon=180, )

draw_map(m)

m.scatter(df['Longitude'].values,
                  df['Latitude'].values,
                  latlon = True,
                  c = [magn for magn in df['Magnitude']],
                  s = [depth/5 for depth in df['Depth']],
                  cmap = 'viridis',
                  edgecolors = 'none')
```

```
cbar= plt.colorbar()
cbar.ax.tick_params(labelsize=20)

plt.title('Worldwide Distribution of Earthquakes',fontsize='xx-large')
plt.savefig('eq_dis.png')
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/matplotlib/collections.py:885:
RuntimeWarning: invalid value encountered in sqrt
  scale = np.sqrt(self._sizes) * dpi / 72.0 * self._factor
findfont: Font family ['sans-serif'] not found. Falling back to DejaVu Sans.
findfont: Font family ['sans-serif'] not found. Falling back to DejaVu Sans.
```

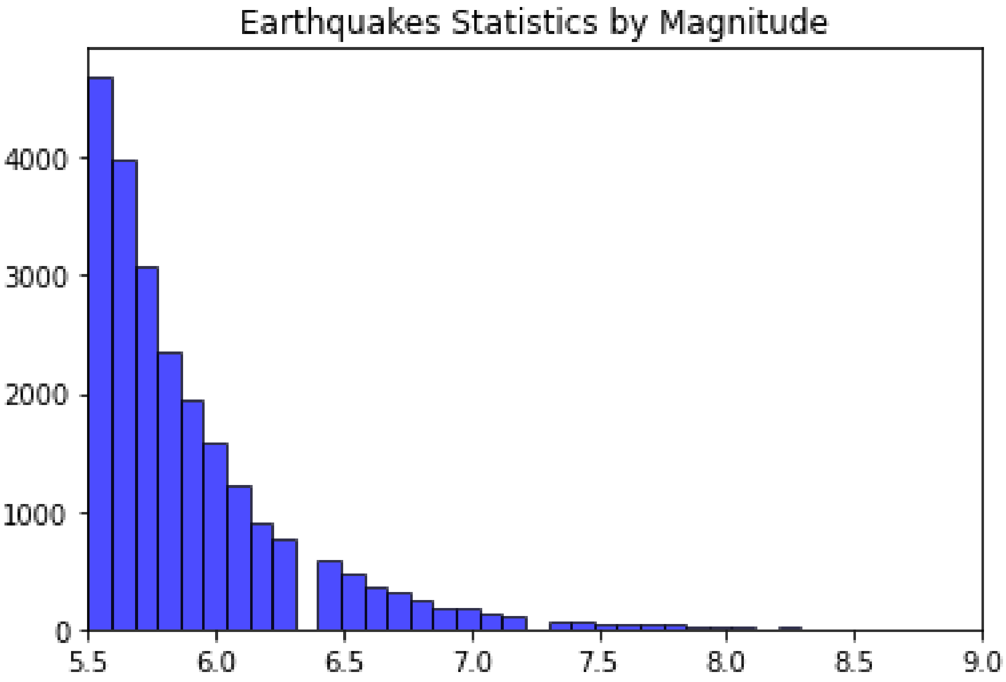


```
[ ]: import matplotlib.pyplot as plt
     import numpy as np
     import matplotlib
     import pandas as pd

     df = pd.read_csv('database.csv')
     data = np.array([float(depth) for depth in df['Depth']])

     plt.hist(data, bins=40, density=0, facecolor="blue", edgecolor="black", alpha=0.
      ↪7)
     plt.xlim(0,700)

     plt.title("Earthquakes Statistics by Depth")
     plt.savefig('eq_depth.png')
     plt.show()
```



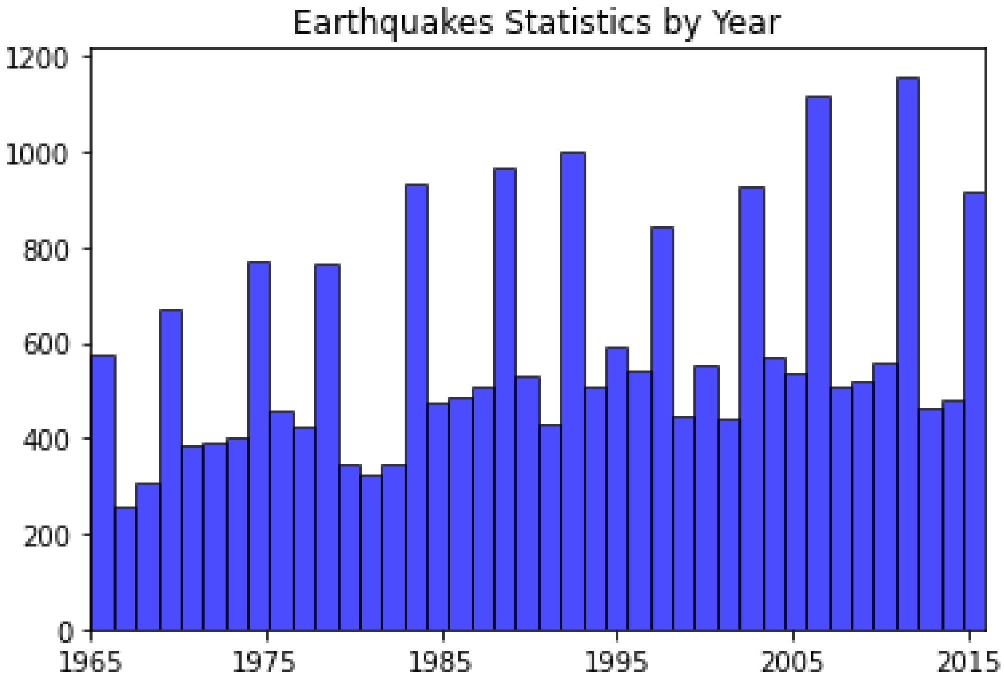
[ ]:

```
import matplotlib.pyplot as plt
import numpy as np
import matplotlib
import pandas as pd

df = pd.read_csv('database.csv')
data = np.array([float(magn) for magn in df['Magnitude']])

plt.hist(data, bins=40, density=0, facecolor="blue", edgecolor="black", alpha=0.
 ↪7)
plt.xlim(5.5,9.0)

plt.title("Earthquakes Statistics by Magnitude")
plt.savefig('eq_magn.png')
plt.show()
```



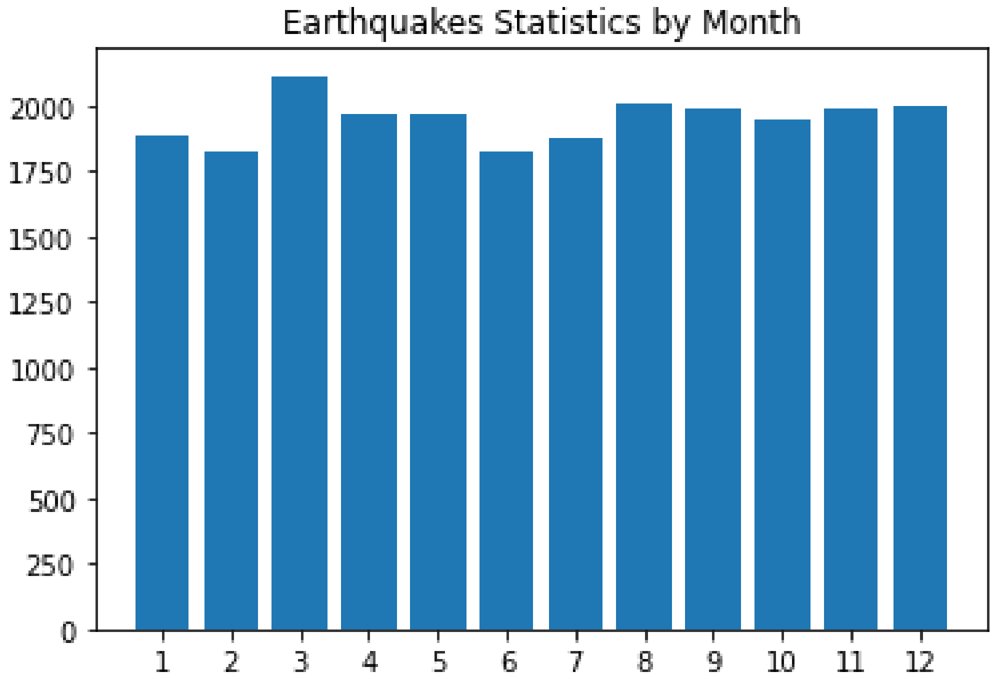
[ ]:

```
import matplotlib.pyplot as plt
import numpy as np
import matplotlib
import pandas as pd

df = pd.read_csv('database.csv')
data = np.array([int(date[-4:]) if date[-1]!='Z' else int(date[:4]) for date in␣
 ↪df['Date']])

plt.hist(data, bins=40, density=0, facecolor="blue", edgecolor="black", alpha=0.
 ↪7)
plt.xlim(1965,2016)
plt.xticks(np.arange(1965,2017,10))

plt.title("Earthquakes Statistics by Year")
plt.savefig('eq_years.png')
plt.show()
```



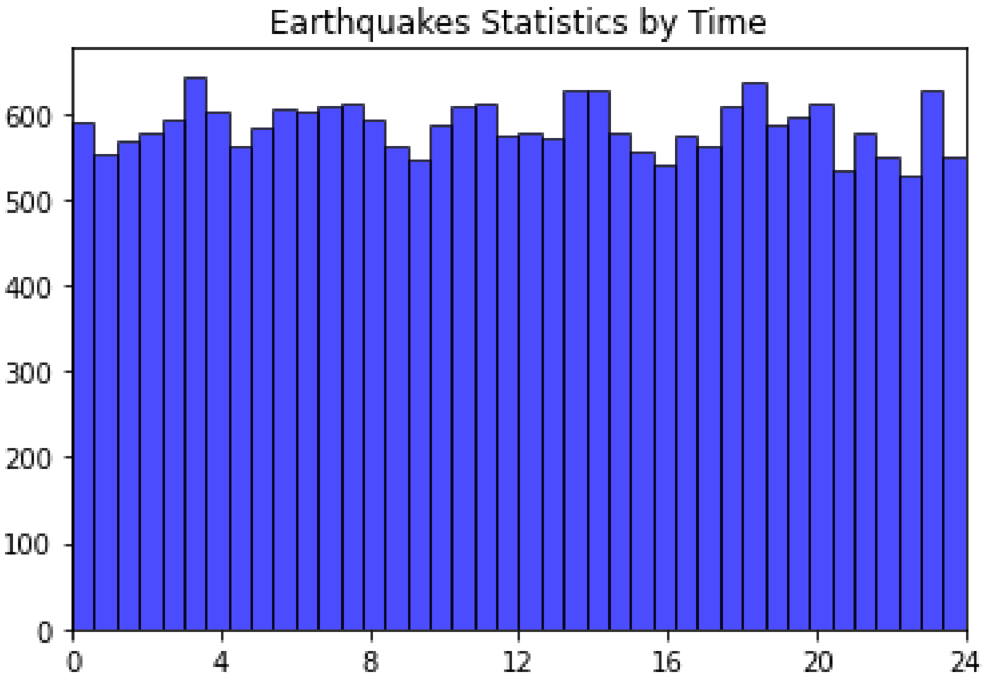
[ ]:

```
import matplotlib.pyplot as plt
import numpy as np
import matplotlib
import pandas as pd

df = pd.read_csv('database.csv')

cnt=[0]*12
for t in df['Date']:
  if t[-1]=='Z':
    cnt[int(t[5:7])-1]=cnt[int(t[5:7])-1]+1
  else:
    month, day, year = t.split('/')
    cnt[int(month)-1]=cnt[int(month)-1]+1

plt.bar(range(1,13), cnt)
plt.xticks(np.arange(1,13,1))

plt.title("Earthquakes Statistics by Month")
plt.savefig('eq_month.png')
plt.show()
```



[ ]:

```
import matplotlib.pyplot as plt
import numpy as np
import matplotlib
import pandas as pd

df = pd.read_csv('database.csv')

data=[]
for t in df['Time']:
  if t[-1]=='Z':
    t=t[-13:-5]
  hour, minute, second = t.split(':')
  data.append((float(hour)*3600.0+float(minute)*60.0+float(second))/3600.0)

data=np.array(data)

plt.hist(data, bins=40, density=0, facecolor="blue", edgecolor="black", alpha=0.
 ↪7)
plt.xlim(0, 24)
plt.xticks(np.arange(0,25,4))

plt.title("Earthquakes Statistics by Time")
plt.savefig('eq_time.png')
plt.show()
```


Earthquakes Statistics by Time
600
500
400
300
200
100
0
0
4
8
12
16
20
24