Plotting Earthquake magnitude vs time

In [704]:

```
import json
import requests
import pandas as pd
import datetime
import time
import pytz
import numpy as np
import matplotlib.pyplot as plt
from pandas.plotting import register_matplotlib_converters
import urllib
import dateutil.parser
from datetime import datetime
from pytz import timezone
import os
```

In [705]:

```
!pip install folium
import folium
from folium.plugins import TimestampedGeoJson
import seaborn as sns
from folium import plugins
```

```
Requirement already satisfied: folium in c:\users\lenovo\anaconda3\new folder\lib\site-packages (0
.10.1)
Requirement already satisfied: jinja2>=2.9 in c:\users\lenovo\anaconda3\new folder\lib\site-
packages (from folium) (2.10.3)
Requirement already satisfied: numpy in c:\users\lenovo\anaconda3\new folder\lib\site-packages
(from folium) (1.16.5)
Requirement already satisfied: branca>=0.3.0 in c:\users\lenovo\anaconda3\new folder\lib\site-
packages (from folium) (0.4.0)
Requirement already satisfied: requests in c:\users\lenovo\anaconda3\new folder\lib\site-packages
(from folium) (2.22.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\lenovo\anaconda3\new folder\lib\site-p
ackages (from jinja2>=2.9->folium) (1.1.1)
Requirement already satisfied: six in c:\users\lenovo\anaconda3\new folder\lib\site-packages (from
branca>=0.3.0->folium) (1.12.0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\users\lenovo\anaconda3\new folder\lib\site-packages (from requests->folium) (1.24.2)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\lenovo\anaconda3\new
folder\lib\site-packages (from requests->folium) (2019.9.11)
Requirement already satisfied: idna<2.9,>=2.5 in c:\users\lenovo\anaconda3\new folder\lib\site-pac
kages (from requests->folium) (2.8)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in c:\users\lenovo\anaconda3\new
folder\lib\site-packages (from requests->folium) (3.0.4)
```

In [753]:

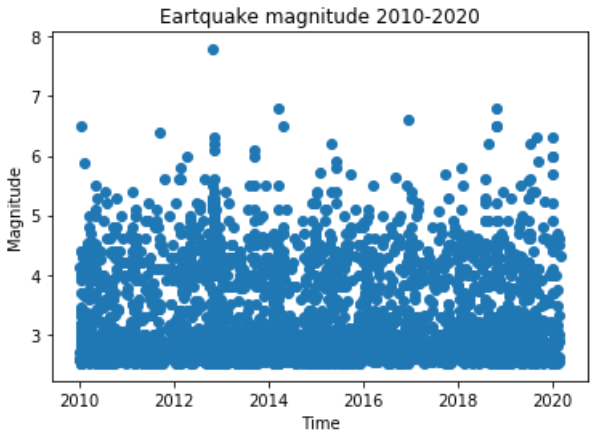
```
data=pd.read_csv("Project3PlateTectonics.csv")
data.shape

time=data.time
magnitude=data.mag
latitude=data.latitude
longitude=data.longitude
```

In [767]:

```
time1=pd.to_datetime(time, format='%Y-%m-%dT%H:%M:%S.%fZ')
plt.scatter(time1,mag)
plt.ylabel('Magnitude')
plt.xlabel('Time')
plt.title('Eartquake magnitude 2010-2020')
```

Out[767]:

```
Text(0.5, 1.0, 'Eartquake magnitude 2010-2020')
```



In [772]:

```
print("minimum", min(mag))
```

```
minimum 2.5
```

In [773]:

```
print("maximum", max(mag))
```

```
maximum 7.8
```

In [782]:

```
avg = sum(mag)/len(mag)
print("average", avg)
```

```
average 3.3513366802635796
```

Earthquake locations (lat and lon) on a map with magnitude

In [ ]:

```

```

In [694]:

```
m=folium.Map(location=[47.7589 , -130 ], zoom_start=5)


for index, row in data.iterrows():
    folium.CircleMarker([row['latitude'], row['longitude']],
                        radius= [row['mag']],
                        popup=print(row['mag']),
                        #fill_color="#3db7e4", # divvy color
                       ).add_to(m)
```

```
4.33
2.55
3.0
3.32
4.5
2.62
2.9
2.9
4.6
```

2.55
4.7
2.88
2.82
3.06
2.51
3.76
3.16
3.47
3.0
4.5
2.61
4.0
3.1
2.85
2.95
3.1
2.9
2.91
2.66
2.58
4.6
2.5
3.7
2.51
2.72
2.66
2.7
2.9
4.5
4.5
3.7
4.5
4.0
6.3
4.9
4.1
4.4
6.0
6.0
5.7
5.2
4.06
2.5
3.2
3.07
2.87
2.82
2.56
3.17
4.0
3.54
4.53
2.58
2.56
2.73
3.09
2.8
2.6
3.11
4.6
2.82
3.5
3.21
3.18
3.58
5.24
3.2
2.77
4.6
3.01
2.88
4.8
3.23
2.52
3.2
2.68

2.92
3.09
3.9
3.49
3.1
2.79
2.57
2.87
2.75
2.8
3.0
4.1
2.6
2.71
3.1
4.1
5.9
3.7
3.24
3.1
3.0
3.1
2.9
6.3
2.56
2.51
2.53
2.84
3.3
2.62
3.3
3.4
5.4
3.8
4.1
2.61
2.91
3.3
4.0
2.93
2.67
2.68
2.69
4.4
2.55
2.51
2.77
3.35
2.5
2.81
2.73
3.3
2.84
2.65
3.74
2.68
2.96
5.4
2.7
2.8
2.92
2.78
2.79
2.59
2.57
2.52
4.6
5.6
4.7
3.7
6.2
3.7
4.1
2.66
4.1
3.8
3.12

3.1
4.1
2.55
2.9
3.87
5.58
2.8
3.1
2.8
4.6
3.3
4.7
4.2
4.1
4.2
5.3
4.1
5.3
3.6
4.9
3.2
4.1
2.61
4.0
2.65
2.92
2.62
2.89
2.7
2.68
4.0
2.71
3.9
3.78
3.38
2.5
2.56
3.0
3.57
4.2
3.3
2.7
4.1
4.2
2.62
3.0
4.1
2.67
2.61
3.1
2.61
3.2
3.1
5.0
4.3
3.3
3.1
2.9
2.73
2.64
4.32
2.52
3.07
2.65
2.83
3.7
3.9
3.08
5.5
4.9
2.78
3.29
2.66
2.7
2.8
2.67
2.55

4.8
3.0
4.7
2.7
2.56
3.0
4.19
3.0
2.58
3.4
2.68
4.35
3.04
2.69
2.82
3.1
3.3
3.1
2.87
2.67
2.55
2.82
2.84
2.7
3.79
2.57
3.2
2.87
4.08
2.99
3.0
2.67
2.77
4.0
2.77
4.1
5.3
4.0
4.1
3.4
2.64
4.2
4.1
2.54
2.63
2.66
2.51
4.39
3.05
2.51
2.53
3.46
4.19
3.11
4.48
2.5
2.83
3.64
3.13
2.53
2.94
3.05
4.3
2.7
2.62
2.86
2.6
3.37
3.7
3.2
2.7
4.4
2.81
3.0
2.57
2.51
3.05

4.1
2.84
4.9
3.81
4.08
2.64
2.52
4.25
3.5
2.95
3.2
2.62
4.1
3.26
2.61
2.53
2.62
2.7
3.2
2.93
3.1
2.5
2.67
2.94
3.1
3.0
2.98
4.6
4.6
5.4
3.2
3.4
2.88
4.7
4.1
2.72
3.13
4.07
2.64
2.51
3.11
3.9
4.3
3.8
4.5
4.1
3.9
2.52
3.79
2.8
2.64
4.3
2.64
4.9
2.9
4.0
3.2
3.1
2.94
3.2
4.5
4.0
3.0
5.2
4.6
4.3
4.2
2.64
4.4
4.3
4.1
4.9
4.5
4.4
6.5
6.8
6.5

3.2
2.59
2.55
2.56
2.83
2.73
2.9
2.54
3.0
3.3
3.18
2.52
2.88
3.83
2.81
4.3
2.61
3.0
2.52
2.99
3.1
2.67
3.27
2.92
2.61
2.62
2.56
2.69
2.6
3.07
3.0
2.76
2.53
3.1
2.8
2.51
3.1
3.2
2.9
3.5
3.03
3.0
2.57
3.1
3.9
2.8
4.0
3.1
3.1
3.1
2.55
3.8
4.5
3.2
3.5
6.2
4.9
2.69
3.19
3.35
4.2
2.64
2.57
2.6
3.2
2.6
3.2
2.7
2.67
4.3
4.4
3.8
3.0
3.2
5.3
2.7
2.9

2.65
2.53
4.53
3.6
3.1
3.1
4.9
2.9
2.8
5.6
4.3
3.2
4.3
3.5
5.2
4.9
2.73
3.05
3.9
2.59
2.88
2.81
3.0
2.65
2.69
4.1
2.82
2.67
2.73
2.75
2.62
2.68
2.88
2.9
2.68
3.43
3.72
3.19
2.9
2.6
3.2
3.1
2.51
2.53
2.78
2.87
3.0
2.61
2.74
2.65
4.2
2.95
4.4
2.81
2.73
2.61
4.6
2.61
2.82
2.94
2.93
3.8
3.1
3.2
3.0
2.67
4.5
2.8
4.47
2.8
3.18
2.54
4.0
3.8
2.79
4.3
2.6

2.6
4.0
3.1
3.1
2.51
2.58
3.28
3.7
2.8
4.4
2.67
3.5
2.55
2.87
2.6
3.6
2.54
2.85
4.66
4.42
2.53
2.55
2.8
2.93
4.0
4.6
2.8
2.94
3.1
2.64
2.84
2.87
2.8
4.48
4.1
2.74
3.3
4.6
3.1
3.2
2.81
3.5
3.3
3.7
3.4
3.4
2.51
2.58
2.75
3.0
2.67
2.55
3.06
4.33
2.8
2.8
2.9
2.76
5.5
4.1
3.3
3.3
5.1
5.8
2.9
4.0
2.57
2.57
3.2
2.68
4.0
4.3
4.0
4.4
4.7
4.5
3.4
4.0

4.0
2.68
2.94
3.1
4.52
3.6
4.1
3.76
2.98
3.82
2.84
4.3
3.8
2.73
2.7
4.7
3.87
4.26
2.7
2.84
2.8
2.8
4.4
2.65
4.8
3.03
2.58
4.7
3.2
4.9
3.1
3.01
2.75
4.5
2.66
2.98
2.73
2.55
2.5
2.9
2.9
3.0
3.1
4.2
3.1
2.8
2.8
2.8
2.7
2.9
2.5
3.38
2.8
3.94
5.3
2.9
4.5
3.5
3.5
3.3
2.75
2.97
2.9
2.52
2.66
3.0
2.6
2.9
4.4
2.77
2.8
4.3
3.0
3.9
2.51
3.2
2.9
2.9

2.9
3.91
2.87
3.0
5.7
3.01
3.0
2.76
2.68
2.56
2.54
3.3
3.3
3.3
2.76
2.58
2.9
2.7
3.8
2.82
4.3
4.1
2.54
4.7
2.54
3.9
2.99
3.3
2.89
4.8
3.0
2.57
2.53
2.56
4.58
4.2
5.08
2.84
2.6
2.8
2.69
2.56
3.0
2.58
4.1
3.74
2.57
4.1
2.62
4.03
3.8
3.9
2.7
3.14
2.91
2.55
2.64
2.57
3.53
4.8
2.62
3.4
3.1
2.54
2.62
2.8
3.1
2.88
2.9
4.3
3.96
2.58
3.6
4.4
2.9
3.3
2.84
4.4

4.4
4.2
2.92
2.85
2.64
3.33
2.7
3.9
2.52
4.2
2.81
2.71
2.82
2.66
3.07
4.1
2.55
4.2
2.89
3.21
2.6
3.3
2.8
4.0
3.13
2.94
2.67
4.04
2.96
2.58
3.85
4.2
4.2
2.71
3.12
2.84
5.2
3.3
4.5
5.1
2.85
4.0
3.3
3.9
2.68
2.72
5.0
2.74
3.68
4.0
2.86
2.99
2.88
2.6
3.58
2.58
3.05
3.8
3.3
2.61
3.0
3.12
4.3
4.6
4.6
4.8
5.4
4.2
4.4
2.7
4.6
3.9
5.3
4.2
3.2
3.3
3.2
4.2

4.3
4.1
2.85
4.3
3.1
2.86
3.2
2.6
2.94
2.94
2.68
4.3
2.76
2.57
2.91
3.3
3.3
3.4
4.7
2.51
3.3
2.91
6.6
4.3
2.78
4.35
3.3
2.92
2.95
2.57
4.2
5.5
3.0
2.56
4.5
3.84
3.2
3.2
3.11
2.9
2.6
2.95
2.85
2.52
2.64
2.91
2.74
4.06
2.53
3.82
2.55
3.55
2.92
4.11
2.65
2.53
2.6
5.0
4.1
4.5
4.1
3.4
2.61
2.58
2.52
2.51
2.79
2.74
2.8
4.6
2.93
4.9
2.91
2.83
3.7
2.59
4.5

2.77
4.9
3.0
3.25
3.4
2.9
2.87
3.0
2.77
5.64
3.0
2.7
4.49
2.92
4.4
3.2
3.2
3.2
2.9
2.67
3.0
3.3
2.67
4.7
2.9
3.9
3.17
3.5
3.8
2.57
2.5
3.3
2.86
4.81
2.74
2.66
4.0
2.6
2.81
4.3
3.1
3.69
2.82
2.51
2.89
2.67
3.0
2.61
2.81
2.88
2.76
2.8
3.2
2.56
4.6
2.87
2.65
2.56
3.1
2.8
3.11
4.5
3.2
2.6
3.03
3.4
5.0
2.56
2.59
2.72
2.7
4.2
3.2
4.3
4.0
4.0
3.3

2.66
2.77
3.0
3.8
3.18
3.16
2.88
2.9
2.53
2.9
3.1
4.1
2.6
2.9
2.62
2.64
2.7
2.68
2.77
2.56
3.7
2.73
2.9
2.55
3.6
3.1
4.3
2.8
2.56
3.1
2.8
2.9
2.8
4.6
2.7
2.66
3.4
5.5
3.3
2.93
2.54
2.82
4.4
4.2
3.08
3.8
2.82
2.61
2.86
2.81
2.62
2.77
2.57
2.57
3.1
3.4
2.99
2.71
3.8
3.7
2.74
4.6
5.0
3.93
5.1
3.2
3.4
4.1
3.2
3.2
4.1
3.8
4.3
3.9
4.0
2.68
3.6

2.9
4.9
4.4
2.85
3.89
2.64
2.69
3.7
2.58
3.8
2.92
3.66
4.31
2.83
3.02
3.5
2.76
3.1
3.1
4.44
2.58
2.65
3.2
4.79
2.84
4.87
2.98
4.3
3.32
3.6
4.2
3.3
2.8
2.85
4.5
3.3
2.58
3.5
3.5
2.85
2.52
3.04
2.76
2.94
3.32
4.3
2.78
4.0
4.8
2.65
4.5
4.5
2.54
3.25
3.2
3.14
2.8
2.79
2.77
2.62
2.71
2.51
2.7
4.33
2.61
2.79
3.07
2.7
2.96
5.1
2.57
2.9
2.75
2.82
2.92
3.1
3.9

3.08
2.7
2.52
2.84
2.68
2.7
3.0
5.7
2.63
2.75
4.7
4.8
2.8
2.8
2.64
2.56
2.55
3.1
4.5
2.59
3.1
2.5
4.3
3.04
3.56
2.62
2.87
2.5
2.66
2.56
2.76
4.4
2.75
4.7
4.4
4.5
3.8
2.52
2.59
4.2
2.86
2.55
2.5
4.4
3.9
5.0
4.1
2.58
3.3
3.45
2.51
2.8
3.9
3.5
2.53
2.77
2.66
2.63
3.8
2.67
2.69
2.55
2.8
2.59
2.7
2.9
2.85
2.55
3.8
2.71
3.0
4.2
3.2
2.91
3.2
3.0
3.1

3.1
2.77
2.5
3.13
2.8
2.68
2.68
2.56
2.63
2.76
3.9
4.0
3.1
3.8
4.2
3.9
3.6
5.9
4.4
5.4
4.1
4.0
4.2
3.9
4.3
4.2
4.1
5.2
5.8
2.86
4.34
2.71
4.1
2.51
2.98
3.0
2.77
3.4
4.4
2.73
2.81
2.64
2.55
2.99
4.4
2.7
2.87
2.75
2.73
2.95
2.63
4.2
3.3
4.1
4.1
4.2
4.6
6.2
3.8
3.7
3.4
2.59
3.5
3.0
5.4
3.0
2.95
2.76
2.74
2.59
4.0
2.59
2.91
2.53
4.2
4.7
2.5

2.55
3.06
4.5
2.61
3.9
3.6
2.72
4.0
3.9
5.1
2.69
3.2
3.31
2.6
2.93
3.1
2.74
3.8
2.54
2.52
2.59
2.75
2.61
4.1
2.92
2.8
3.79
4.9
2.67
2.59
4.4
2.78
3.0
5.1
2.9
2.63
3.0
2.58
2.5
4.0
3.02
2.64
3.04
2.58
3.3
4.0
3.8
2.59
3.4
3.0
2.8
2.57
2.55
2.5
2.62
2.7
2.98
2.89
2.51
2.94
2.61
2.95
4.25
2.74
2.71
2.88
2.85
2.5
2.97
2.59
3.1
5.72
2.86
3.0
4.5
3.16
2.59

2.76
3.9
4.28
2.98
2.62
2.5
2.85
3.04
3.1
2.86
2.77
4.8
2.7
2.69
2.59
4.5
3.27
2.6
2.54
5.4
2.91
5.36
4.5
2.79
3.63
2.97
2.5
3.1
4.6
4.5
2.9
4.7
4.3
5.2
4.0
5.2
4.2
4.7
4.4
4.2
3.08
2.71
4.4
3.6
3.8
4.6
2.57
2.96
3.6
2.59
4.7
3.2
2.53
2.5
2.57
2.82
2.6
4.29
3.21
2.91
3.31
2.64
2.72
2.52
2.78
2.84
2.83
2.94
2.5
4.9
3.3
5.2
3.4
4.2
3.7
2.53
2.65

2.51
2.53
3.2
3.3
2.59
4.3
2.54
2.54
2.66
2.54
2.92
4.4
2.68
3.2
3.0
3.39
4.1
2.97
2.52
2.92
2.84
2.95
2.52
3.01
4.24
2.54
2.9
2.71
2.53
3.7
2.71
3.64
3.06
3.0
3.93
3.85
4.3
2.6
3.3
2.99
4.7
2.7
2.75
2.94
3.2
2.79
2.73
2.84
2.89
2.52
2.65
4.9
3.0
2.99
2.81
2.57
3.0
3.6
2.65
2.61
2.66
2.6
2.94
2.94
4.5
2.77
2.73
4.5
2.69
2.94
2.95
2.51
3.01
2.67
2.74
2.6
2.97

2.59
3.8
3.22
2.52
2.59
2.51
2.97
2.58
2.6
2.62
4.6
2.62
2.65
2.84
3.1
2.87
2.62
3.0
4.2
2.6
3.03
2.98
3.2
2.64
2.57
4.1
3.8
2.9
2.82
3.01
2.58
2.9
3.2
2.67
2.6
3.31
3.2
3.2
2.54
2.57
2.91
2.71
2.59
2.72
2.64
2.88
2.8
3.01
2.9
4.14
2.85
2.88
2.67
3.7
2.58
2.59
2.59
3.02
2.94
2.75
2.75
3.2
2.71
2.63
2.64
2.65
3.2
5.1
4.1
2.77
2.84
2.93
3.19
3.2
2.92
2.55
3.9

2.6
2.56
2.86
3.9
4.7
4.2
4.1
4.0
4.4
2.63
4.8
4.7
4.3
6.5
3.32
2.82
2.8
2.69
3.3
2.6
2.79
2.8
3.05
2.65
2.9
4.2
2.56
2.75
2.94
4.7
2.58
2.87
2.84
2.83
2.81
2.82
2.62
4.4
2.62
2.74
2.7
2.56
2.84
3.79
2.88
2.58
3.22
2.59
2.63
2.83
2.87
3.95
3.32
2.76
2.5
2.53
3.3
3.04
2.83
4.41
4.01
2.97
2.84
2.87
2.56
2.86
2.83
4.0
2.73
2.9
4.6
5.5
2.57
2.83
3.6
2.99
2.91

2.57
4.0
2.97
5.1
2.83
3.2
3.14
2.83
2.9
2.88
2.81
2.52
2.78
2.91
2.73
2.91
2.9
2.84
2.94
3.65
2.98
2.98
3.28
2.61
2.8
3.05
3.46
4.47
3.24
2.95
2.65
3.71
3.64
2.51
2.69
2.62
2.84
2.5
3.0
2.72
2.91
2.99
2.68
2.93
3.03
2.78
2.5
2.55
2.63
3.33
3.22
2.61
2.76
2.82
4.35
2.93
3.03
3.79
3.39
4.3
6.8
3.07
2.65
2.5
2.59
2.84
3.1
3.1
2.7
2.61
3.4
2.57
2.62
2.96
2.81
2.51
2.9

2.56
2.51
2.97
3.08
2.58
4.5
4.1
3.0
2.69
2.56
3.37
3.5
3.4
4.2
2.51
2.72
2.99
3.2
3.56
2.52
2.9
3.3
4.0
2.8
3.0
3.0
3.2
4.7
2.54
3.0
3.83
2.95
4.63
2.65
2.75
3.1
2.9
2.58
4.0
3.8
3.7
3.0
2.77
2.72
2.63
2.63
4.02
2.67
4.3
2.9
4.11
3.0
2.58
4.6
2.59
3.08
3.8
2.62
2.75
2.8
2.61
2.57
4.3
2.58
2.64
2.75
3.1
3.0
2.9
3.0
3.1
5.5
4.2
2.53
3.4
4.5
2.65

2.75
3.41
3.65
2.8
2.64
4.0
4.1
3.2
4.0
2.9
2.76
4.0
2.59
5.0
2.76
4.65
2.79
4.0
3.2
4.6
3.8
3.0
2.54
3.19
2.98
2.69
2.9
3.9
2.66
3.1
2.92
2.56
3.26
2.69
3.8
2.53
4.94
2.57
2.62
2.52
3.9
2.94
3.84
2.6
2.57
2.8
3.5
3.8
2.57
4.5
2.54
2.5
2.5
2.66
4.02
2.61
3.17
4.3
4.2
4.6
4.3
2.59
2.67
4.3
4.9
4.6
6.0
4.7
5.5
4.2
5.1
4.5
6.1
2.6
3.9
2.5
3.0

3.0
2.76
2.66
2.88
2.57
2.76
2.78
2.74
3.17
4.3
2.86
2.54
2.76
2.54
2.55
4.5
3.0
4.0
2.9
3.9
5.5
4.47
2.7
2.82
2.53
2.84
2.71
2.81
2.76
4.6
3.9
5.1
4.0
2.9
3.2
2.68
3.4
2.61
4.6
2.51
2.5
3.5
3.13
2.56
2.82
2.81
2.64
3.26
2.76
3.9
3.0
2.53
3.1
2.5
3.8
2.6
2.63
2.61
2.65
2.64
2.72
4.1
2.8
2.84
3.78
3.3
2.64
3.9
2.76
2.67
2.8
2.9
4.1
4.0
2.7
2.7
2.51
3.9

3.9
4.4
4.2
2.9
2.94
4.2
4.3
4.2
4.3
3.1
2.9
2.83
2.6
3.78
2.6
4.28
2.6
2.7
2.53
3.57
2.52
2.73
3.03
2.7
2.74
3.3
2.5
2.6
2.58
2.54
2.51
2.5
3.86
2.54
2.69
2.63
2.81
3.06
2.52
3.11
2.85
3.03
4.3
3.06
3.01
2.69
2.58
2.96
2.55
3.8
4.81
2.77
2.6
2.94
4.29
3.23
2.76
4.3
2.68
2.58
4.5
4.5
4.3
4.5
2.65
2.94
2.82
3.5
3.8
2.68
4.18
3.04
2.78
2.96
5.2
4.9
2.76
2.85

2.85
4.3
2.68
2.57
2.89
3.36
2.56
2.8
2.51
3.07
4.0
3.8
2.62
5.4
3.45
2.52
2.88
2.76
3.0
3.0
2.82
4.02
2.63
2.72
2.55
2.65
4.3
2.66
2.59
2.62
2.77
4.02
3.24
4.1
3.31
2.83
2.61
4.3
3.0
2.87
3.9
2.92
2.54
4.5
2.64
3.06
2.52
3.09
3.9
2.87
3.96
4.1
4.5
3.59
2.78
4.2
3.79
4.3
2.58
4.8
2.5
4.6
3.52
4.1
2.75
4.4
3.11
2.7
2.54
4.2
2.52
2.94
3.34
2.62
3.08
2.62
2.53
4.0

4.0
2.61
3.72
2.89
2.8
2.91
4.3
2.5
2.71
4.7
3.47
4.7
5.1
4.1
4.1
4.2
4.3
2.6
4.1
4.7
2.78
4.4
6.1
4.2
4.1
4.7
4.8
4.0
5.1
4.2
4.0
4.0
4.0
4.5
4.3
2.89
4.5
4.0
4.6
4.0
4.5
2.65
3.1
4.8
5.0
4.5
4.3
4.4
4.6
5.2
4.0
3.21
4.2
4.7
4.1
4.1
4.0
5.1
5.0
4.2
4.1
5.3
6.2
4.1
4.1
4.0
4.0
4.3
4.9
4.3
4.2
4.4
4.0
4.4
4.4
4.1
4.0

4.0
4.0
4.0
4.0
4.0
4.4
5.0
4.6
4.3
4.4
4.0
4.1
4.1
4.0
5.0
4.3
4.0
4.7
4.1
4.7
4.1
4.2
4.6
4.0
4.7
4.6
4.6
5.2
5.6
5.4
4.9
5.2
6.3
4.3
4.5
4.0
4.3
4.5
4.7
4.4
4.4
4.4
4.0
4.0
4.3
4.0
4.4
4.5
4.2
4.0
4.2
4.1
5.1
4.2
4.1
4.0
4.0
4.0
5.1
5.2
4.1
4.3
4.0
4.3
5.1
4.0
4.1
4.6
4.5
4.2
4.4
4.2
4.6
4.1
4.4
5.3
4.2

4.1
4.4
4.3
4.3
4.2
4.0
4.0
4.0
5.2
4.9
4.4
4.9
4.0
4.4
4.0
4.0
4.0
4.0
5.2
4.4
4.1
4.1
4.5
4.2
4.0
4.1
4.0
4.2
4.2
4.4
4.3
4.3
4.2
4.3
4.3
4.4
4.5
4.3
4.1
4.1
4.1
5.3
4.7
5.1
5.2
5.5
4.6
4.2
4.3
7.8
3.3
2.6
2.51
3.69
2.7
2.79
3.55
3.15
4.0
3.34
2.54
2.63
2.73
3.2
3.2
2.59
4.7
4.3
3.8
2.7
2.78
2.79
2.67
4.2
2.6
2.68
4.5

3.84
2.75
4.51
2.5
2.52
2.65
3.51
2.6
2.9
2.54
2.67
4.1
3.1
2.57
3.0
2.61
4.7
3.09
3.96
3.05
4.43
2.68
3.12
2.85
2.5
2.57
2.58
2.91
2.51
3.3
2.97
2.66
2.59
2.53
2.86
4.1
2.8
2.98
2.8
3.9
4.5
2.9
3.0
3.0
4.5
5.6
2.53
2.61
2.65
2.56
2.66
4.38
4.5
2.63
4.9
2.65
2.82
2.87
2.57
2.8
2.98
2.66
2.63
4.0
2.75
2.79
2.79
4.2
2.88
2.76
3.49
2.59
2.76
2.92
5.19
5.1
4.0

2.87
2.63
2.63
2.64
3.99
3.0
2.65
4.42
4.6
4.6
2.62
2.62
2.78
3.56
3.97
3.1
4.8
3.47
2.69
5.2
3.4
2.73
2.56
3.4
2.71
2.7
2.95
2.81
3.55
2.68
2.75
2.63
2.56
4.1
3.8
4.0
2.57
2.89
2.51
4.4
2.73
3.05
2.83
2.55
2.86
2.79
2.53
2.56
4.08
2.61
4.7
2.69
3.3
2.57
2.98
2.51
5.0
4.2
2.78
4.8
4.1
4.0
2.56
4.5
4.7
4.3
4.5
4.9
4.2
3.83
3.0
2.57
4.0
4.7
2.53
2.88
4.15

2.56
3.9
2.6
2.54
3.0
2.74
3.4
2.57
2.74
6.0
2.63
2.82
2.85
2.54
2.62
2.86
2.76
3.4
2.96
2.72
4.4
3.9
2.56
2.59
2.93
3.7
3.17
2.9
3.61
3.34
2.56
4.2
3.12
2.66
4.0
4.4
4.1
2.67
2.89
3.7
2.8
4.1
3.73
2.95
3.0
3.12
2.55
4.28
4.2
3.4
3.86
2.5
3.4
2.95
2.85
3.07
2.73
2.55
4.7
2.7
2.52
2.74
5.8
4.1
2.91
2.77
5.6
2.97
2.62
3.22
2.75
2.53
3.4
3.2
4.8
5.6
4.3

2.6
4.0
4.0
2.6
2.79
3.3
4.0
2.55
4.28
2.51
2.58
5.2
2.8
3.18
3.21
3.2
2.56
2.64
2.94
2.71
4.1
4.5
3.9
3.02
4.3
3.03
4.02
2.8
3.5
3.5
2.62
2.73
2.66
5.0
2.7
2.67
4.1
2.64
2.9
2.76
3.15
2.59
2.8
2.62
3.1
2.77
3.9
2.76
2.71
4.1
5.6
4.4
2.5
2.86
2.63
4.5
2.62
2.6
2.8
2.6
2.61
2.86
4.1
3.42
2.94
4.6
3.04
3.5
2.79
6.4
2.51
2.63
2.81
2.81
4.5
2.91
2.53

2.57
4.8
2.9
2.9
2.7
3.2
3.42
2.6
2.76
2.71
5.1
4.9
3.5
2.53
2.51
4.1
3.7
2.56
2.84
4.8
3.55
2.8
2.66
2.99
2.65
2.99
2.6
3.85
3.0
2.61
2.8
2.88
2.53
2.74
2.69
2.5
3.33
2.7
2.73
4.1
3.5
2.79
2.59
2.55
2.81
4.3
2.93
2.75
3.6
3.8
2.52
5.2
4.2
4.9
3.69
2.6
2.72
2.92
2.73
2.91
4.0
2.53
2.67
2.88
3.0
2.59
2.61
2.5
3.3
4.9
2.6
2.9
2.52
3.74
2.54
4.6
2.92

2.86
2.89
3.0
3.0
2.66
2.63
2.51
2.6
4.2
2.69
3.9
3.25
2.6
2.54
2.92
2.87
4.4
4.2
3.9
4.2
2.54
3.6
2.64
4.4
3.0
2.54
3.47
2.83
3.2
3.53
4.6
2.7
2.81
3.35
2.97
2.5
4.6
4.4
5.1
4.0
3.9
5.0
5.0
2.55
4.54
4.5
2.5
3.4
3.4
4.1
2.77
3.22
3.1
4.3
3.0
3.0
4.4
4.1
3.5
3.5
2.85
2.59
5.4
4.8
2.73
2.57
3.04
2.7
4.6
3.2
4.7
4.0
2.85
2.7
2.55
3.1
2.56

2.72
2.82
2.75
2.59
4.2
2.53
2.8
3.79
3.68
3.69
2.51
2.89
2.57
2.61
2.6
2.96
2.62
4.0
3.95
2.53
2.61
4.2
2.82
2.7
2.81
4.6
2.7
5.0
4.2
2.54
3.05
4.4
2.73
2.92
2.81
2.81
4.2
2.69
3.4
3.43
2.64
4.8
2.63
4.3
4.1
2.6
2.66
4.0
4.39
5.3
2.83
4.7
2.55
4.3
3.31
4.8
2.51
3.1
2.56
2.59
3.7
2.8
3.31
3.6
2.51
4.8
2.88
3.9
2.5
2.75
2.74
2.7
2.93
3.9
3.8
2.96
4.4

2.9
3.5
2.67
3.9
3.28
3.18
3.7
2.91
2.52
4.6
2.8
3.3
4.37
2.87
3.15
2.64
2.76
3.0
2.98
2.9
2.51
2.62
3.5
5.2
2.79
4.6
2.7
5.4
2.76
2.93
2.92
3.05
3.9
3.6
3.81
2.84
2.8
2.75
2.5
3.45
4.9
4.0
2.74
2.74
2.53
3.05
3.0
2.6
3.21
4.0
2.78
2.56
2.9
4.4
2.74
2.72
3.81
4.25
3.2
2.74
2.5
2.81
2.5
2.57
2.95
2.84
2.51
2.8
3.42
2.73
4.4
4.0
2.97
3.36
2.69
4.0
4.0

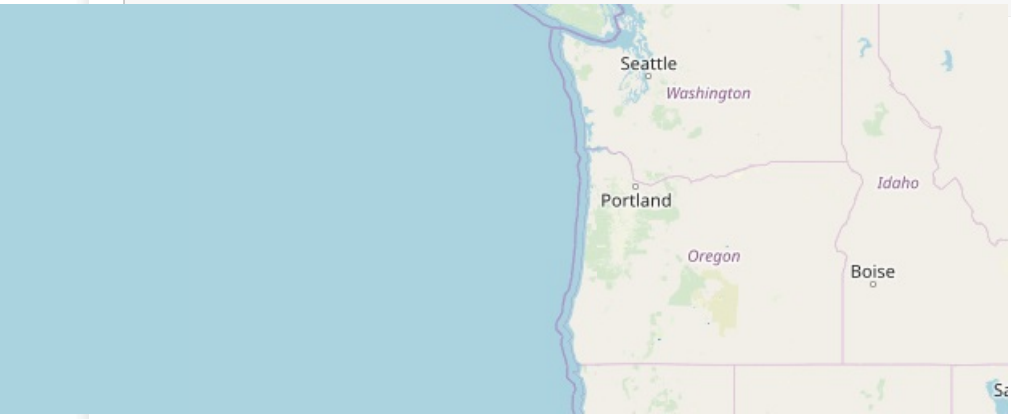
2.72
4.0
2.78
3.6
5.3
2.7
3.9
5.0
2.8
3.7
4.3
4.1
3.2
4.2
3.9
4.5
5.1
2.7
5.5
4.5
3.4
2.5
2.56
2.69
2.71
2.91
2.94
2.65
2.64
2.98
4.5
3.24
2.87
4.52
2.55
3.91
2.81
2.72
4.2
2.6
4.7
4.5
3.1
2.65
2.76
3.3
3.9
3.2
4.13
2.8
3.0
2.57
3.6
3.7
2.68
3.27
4.1
3.74
3.9
4.8
4.3
3.1
2.87
2.7
4.0
4.2
5.0
4.3
4.2
3.26
4.1
2.51
2.77
3.7
4.5
4.53
2.96

2.76
3.9
2.81
2.85
2.55
2.55
3.12
2.7
2.86
2.7
2.66
3.37
2.83
2.53
2.9
3.1
2.64
4.4
2.87
2.74
2.58
3.16
2.65
3.63
3.32
2.77
2.7
2.91
5.88
2.79
2.8
3.0
2.67
2.94
2.7
4.3
2.99
2.61
2.51
2.74
2.62
2.63
3.33
2.89
3.15
4.4
2.98
2.52
3.18
2.81
3.12
2.87
2.75
4.05
2.69
2.66
2.52
2.65
2.69
2.66
2.52
2.73
2.79
2.83
3.91
2.76
4.12
2.7
2.86
2.74
2.99
2.76
2.85
2.95
3.97
2.79
4.41

```
2.69
2.96
2.64
2.74
2.67
3.39
2.63
2.71
3.15
3.32
2.94
3.71
3.46
6.5
4.0
3.09
2.6
2.61
2.59
2.57
2.7
4.12
2.59
```

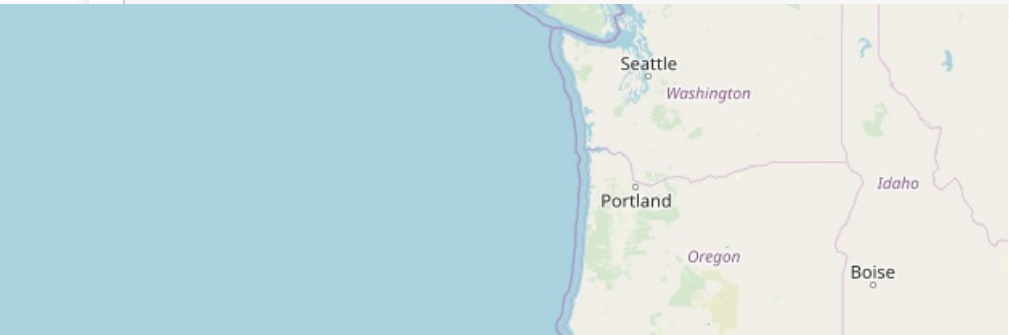
In [745]:

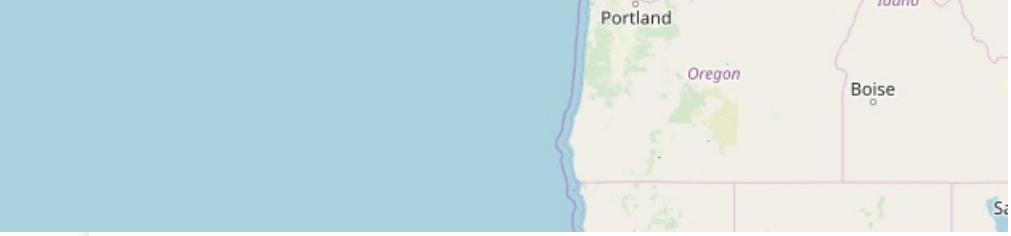
```
coordinates=[(40.5, -128.040335),(40.5, -124)],[(45, -130.040335),(43, -126)]
aline=folium.PolyLine(locations=coordinates,weight=2,color = 'red')
m.add_child(aline)
```



In [744]:

```
m
```





In [697]:

```
for index, row in data.iterrows():
    folium.CircleMarker([row['latitude'], row['longitude']],
                        radius='magnitude',
                        #popup=row['name'],
                        fill_color="#3db7e4", # divvy color
                       ).add_to(m)
```

EXTRA CREDIT (Interactive Map with scroll to move)

In [698]:

```
df=data
def create_geojson_features(df):
    print('> Creating GeoJSON features...')
    features = []
    for _, row in data.iterrows():
        feature = {
            'type': 'Feature',
            'geometry': {
                'type':'Point',
                'coordinates':[row['longitude'],row['latitude']]
            },
            'properties': {
                'time': row['time'],
                'style': {'color' :'red'},
                'icon': 'circle',
                'iconstyle':{
                   # 'fillColor': row['color'],
                    'fillOpacity': 0.8,
                    'stroke': 'true',
                    'radius': row['mag']
                }
            }
        }
        features.append(feature)
    return features
```

In [699]:

```
def make_map(features):
    print('> Making map...')
    coords_US=[50.5039, -130]
    pollution_map = folium.Map(location=coords_US, control_scale=True, zoom_start=5)

    TimestampedGeoJson(
        {'type': 'FeatureCollection',
        'features': features}
        , period='P1M'
```

```
        , add_last_point=True
        , auto_play=False
        , loop=False
        , max_speed=1
        , loop_button=True
        , date_options='YYYY/MM'
        , time_slider_drag_update=True
    ).add_to(pollution_map)
    print('> Done.')
    return pollution_map

def plot_pollutant(df):
    #print('Mapping {} pollution in Belgium in 2013-2015'.format(pollutants[pollutant_ID]
['name']))
    #df = load_data(pollutant_ID)
    # df = clean_data(df)
    #df = prepare_data(df, pollutant_ID)
    features = create_geojson_features(df)
    return make_map(features), df
```

In [700]:

```
pollution_map, df = plot_pollutant(df)
pollution_map
```

```
> Creating GeoJSON features...
> Making map...
> Done.
```

Out[700]:

In [22]:

```
BBox = ((data.longitude.min(),   data.longitude.max(),
    data.latitude.min(), data.latitude.max()))
print(BBox)
```

```
(-133.0153, -123.13566670000002, 38.6885, 52.9)
```

In [ ]:

Map earthquake April 2015

In [749]:

```
data=pd.read_csv("Project3Ocean_2015data.csv")
data.shape

#T_raw=data.loc[:,'time']
#Time = []
#new_format= "%Y-%m-%d %H:%M:%S"
#for i in range (len(T_raw)):
#    datecheck=datetime.strptime(T_raw[i],"%Y-%m-%dT%H:%M:%S.%fz").replace(microsecond=0)
#    datecheck.strftime(new_format)
#    Time.append(datecheck)

time1=data.time1
mag1=data.mag1
latitude=data.latitude
longitude=data.longitude


m1=folium.Map(location=[47.7589 , -130 ], zoom_start=3)


for index, row in data.iterrows():
    folium.CircleMarker([row['latitude'], row['longitude']],
                        radius= row['mag1'],
                        #popup=print(row['mag']),
                        # divvy color
```

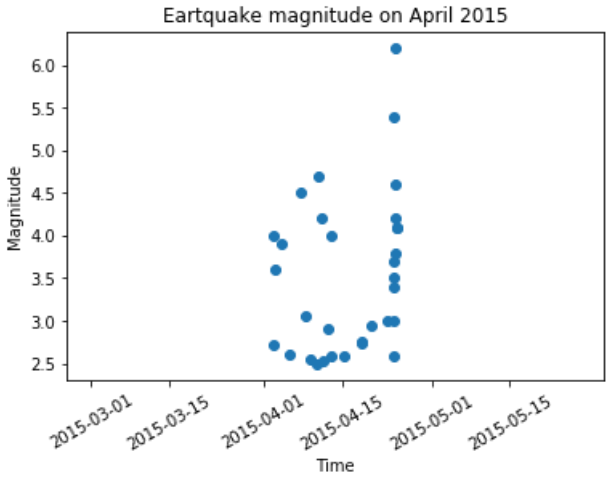
In [750]:

```
time2=pd.to_datetime(time2, format='%Y-%m-%dT%H:%M:%S.%fZ')
```

```
plt.scatter(time2,magnitudee)
plt.xticks(rotation=30)
plt.ylabel('Magnitude')
plt.xlabel('Time')
plt.title('Eartquake magnitude on April 2015')
```

Out[750]:

```
Text(0.5, 1.0, 'Eartquake magnitude on April 2015')
```



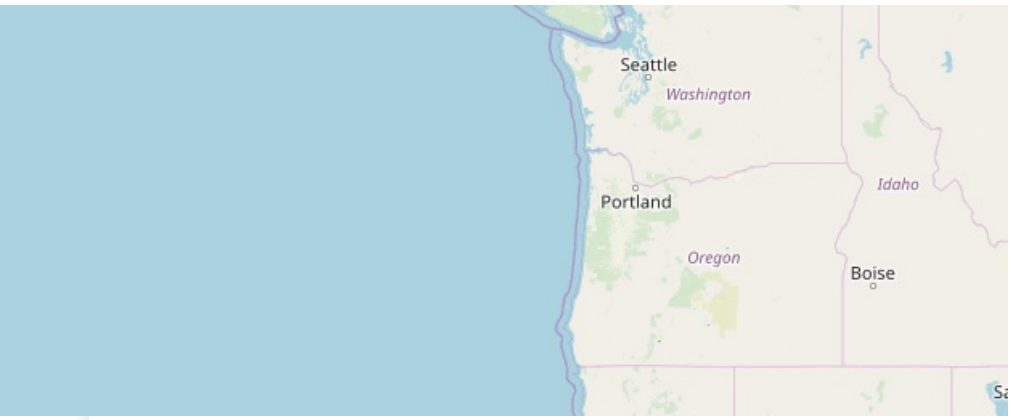
Transform Boundary

In [712]:

```
data=pd.read_csv("Divergrent_and_Boundarydata.csv")
data.shape

#T_raw=data.loc[:,'time']
#Time = []
#new_format= "%Y-%m-%d %H:%M:%S"
#for i in range (len(T_raw)):
#    datecheck=pd.to_datetime(time2, format='%Y-%m-%dT%H)
#    Time.append(datecheck)

time3=[]
for i in range (len(data)):
    time3.append(dateutil.parser.isoparse(data['time3'][i]))


#time2=data.time1
mag3=data.mag3
latitude=data.latitude
longitude=data.longitude


m1=folium.Map(location=[47.7589 , -130 ], zoom_start=5)


for index, row in data.iterrows():
    folium.CircleMarker([row['latitude'], row['longitude']],
                        radius= row['mag3'],
                        #popup=print(row['mag']),
                        #fill_color="#3db7e4", # divvy color
                       ).add_to(m1)

    coordinates=[(40.5, -128.040335),(40.5, -124)],[(45, -130.040335),(43, -126)]
    aline=folium.PolyLine(locations=coordinates,weight=2,color = 'red')
    m1.add_child(aline)
```

In [714]:

```
m1
```

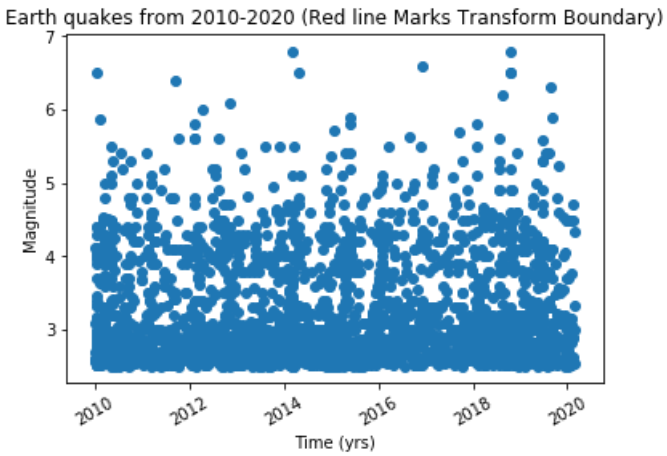


In [717]:

```
plt.scatter(time3,mag3)
plt.xticks(rotation=30)
plt.ylabel('Magnitude')
plt.xlabel('Time (yrs)')
plt.title('Earth quakes from 2010-2020 (Red line Marks Transform Boundary)')
```

Out[717]:

```
Text(0.5, 1.0, 'Earth quakes from 2010-2020 (Red line Marks Transform Boundary)')
```



Divergent Boundary

In [751]:

```
data=pd.read_csv("Divergrent_and_Boundarydata.csv")
data.shape

#T_raw=data.loc[:,'time']
#Time = []
#new_format= "%Y-%m-%d %H:%M:%S"
#for i in range (len(T_raw)):
```

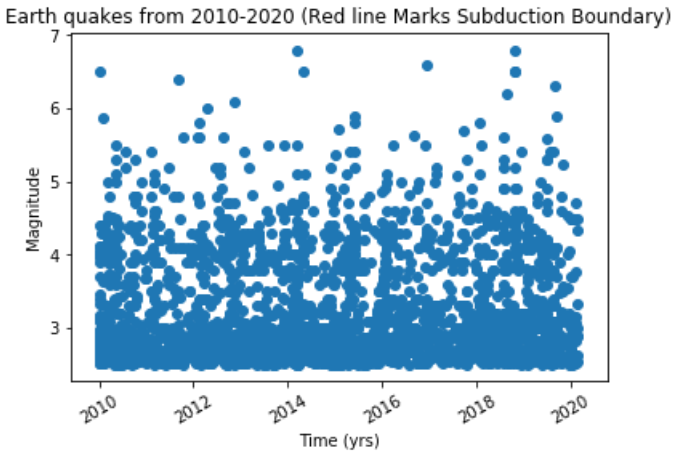
```
#     datecheck=pd.to_datetime(time2, format='%Y-%m-%d %H)
#     Time.append(datecheck)

time3=[]
for i in range (len(data)):
    time3.append(dateutil.parser.isoparse(data['time3'][i]))


#time2=data.time1
mag3=data.mag3
latitude=data.latitude
longitude=data.longitude


m1=folium.Map(location=[45.7589 , -128 ], zoom_start=5)


for index, row in data.iterrows():
    folium.CircleMarker([row['latitude'], row['longitude']],
                        radius= row['mag3'],
                        #popup=print(row['mag']),
                        #fill_color="#3db7e4", # divvy color
                       ).add_to(m1)

    coordinates=[(40.5, -127.040335),(42.5, -126.5)],[(45, -130.040335),(49, -128)]
    aline=folium.PolyLine(locations=coordinates,weight=2,color = 'green')
    m1.add_child(aline)
```

In [741]:

```
plt.scatter(time3,mag3)
plt.xticks(rotation=30)
plt.ylabel('Magnitude')
plt.xlabel('Time (yrs)')
plt.title('Earth quakes from 2010-2020 (Red line Marks Subduction Boundary)')
```

Out[741]:

```
Text(0.5, 1.0, 'Earth quakes from 2010-2020 (Red line Marks Subduction Boundary)')
```


Earth quakes from 2010-2020 (Red line Marks Subduction Boundary)
Magnitude
Time (yrs)
7
6
5
4
3
2010
2012
2014
2016
2018
2020

In [ ]: