

In [11]: `pip install pandas`

```
Requirement already satisfied: pandas in c:\users\samil\anaconda3\lib\site-packages (2.0.3)
Requirement already satisfied: pytz>=2020.1 in c:\users\samil\anaconda3\lib\site-packages (from pandas) (2020.1)
Requirement already satisfied: numpy>=1.20.3; python_version < "3.10" in c:\users\samil\anaconda3\lib\site-packages (from pandas) (1.24.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\samil\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: tzdata>=2022.1 in c:\users\samil\anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\samil\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.15.0)
Note: you may need to restart the kernel to use updated packages.
```

In [60]: `pip install plotly`

```
Requirement already satisfied: plotly in c:\users\samil\anaconda3\lib\site-packages (5.18.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\samil\anaconda3\lib\site-packages (from plotly) (8.2.3)
Requirement already satisfied: packaging in c:\users\samil\anaconda3\lib\site-packages (from plotly) (20.4)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\samil\anaconda3\lib\site-packages (from packaging->plotly) (2.4.7)
Requirement already satisfied: six in c:\users\samil\anaconda3\lib\site-packages (from packaging->plotly) (1.15.0)
Note: you may need to restart the kernel to use updated packages.
```

In [61]: `import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.animation import FuncAnimation
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt`

In [62]: `raw_dataset=pd.read_csv("Doublet_EAF_35F2.csv", sep=",")`

```
In [90]: Doublet_EAF_35F2 = raw_dataset.copy()  
Doublet_EAF_35F2.head()
```

Out[90]:

	Date	Longitude	Latitude	Depth	Magnitude
0	05/02/2023 19:21:54	38.873	38.330	7.00	1.0
1	05/02/2023 18:21:36	35.387	37.362	10.91	1.4
2	05/02/2023 17:10:36	35.053	37.923	7.00	1.3
3	05/02/2023 16:33:47	35.068	37.936	7.02	1.3
4	05/02/2023 15:43:16	35.070	37.937	7.00	1.3

```
In [91]: Doublet_EAF_35F2.shape
```

Out[91]: (13930, 5)

```
In [92]: x = Doublet_EAF_35F2.iloc[:,1].values  
y = Doublet_EAF_35F2.iloc[:,2].values  
z = Doublet_EAF_35F2.iloc[:,3].values  
colors = Doublet_EAF_35F2.iloc[:,4].values  
sizes = Doublet_EAF_35F2.iloc[:,4].values*8
```

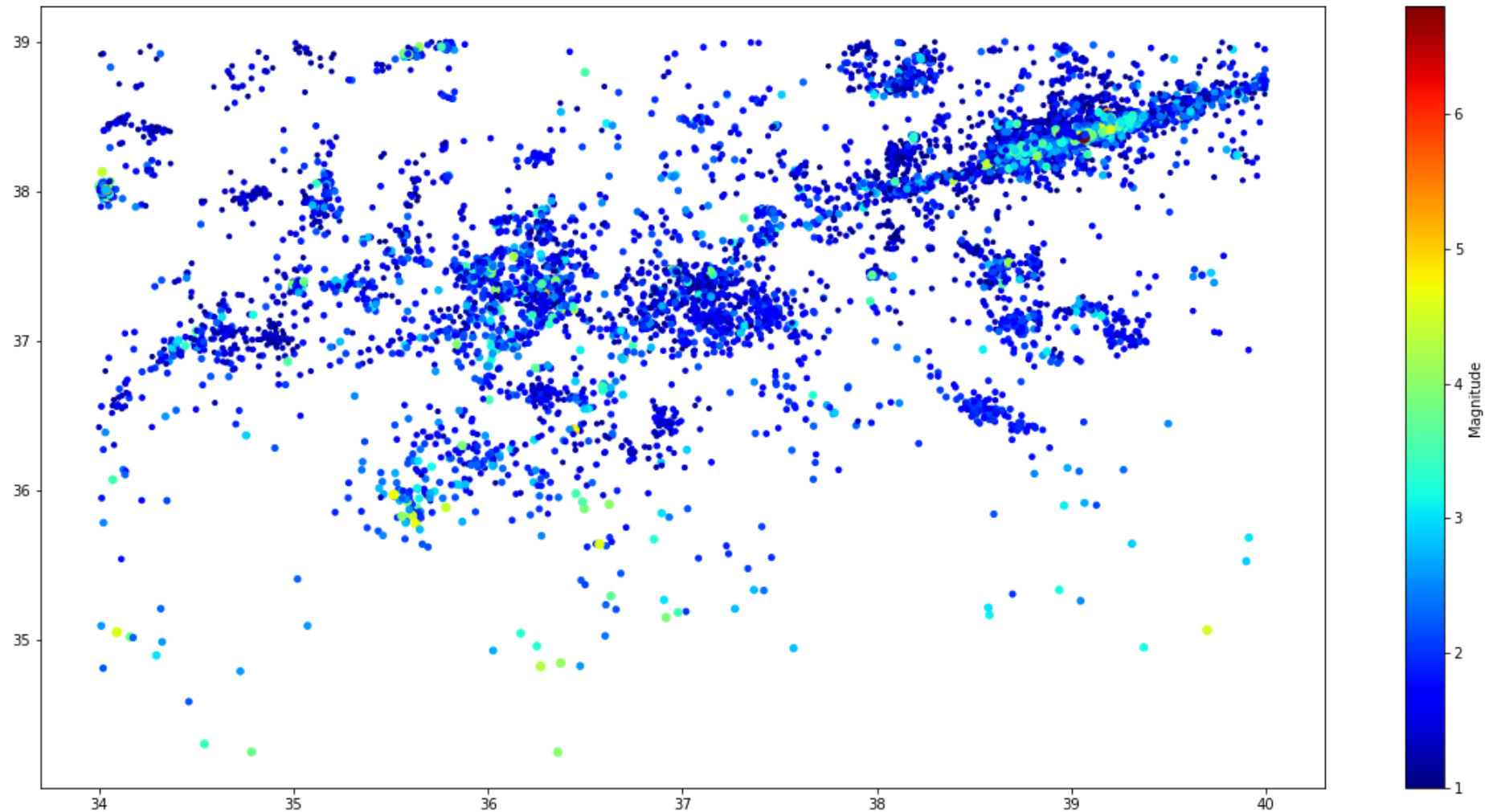
```
In [101]: fig = plt.figure(figsize=(20, 10))
my_cmap = plt.get_cmap('jet')

plt.scatter(x, y, c=colors, s=sizes, cmap='jet')
ax.scatter(x, y, z, c=colors, s=sizes, cmap='jet')
cbar = plt.colorbar(ax.scatter(x, y, z, c=colors, s=sizes, cmap='jet'))

ax.set_xlabel('Longitude')
ax.set_ylabel('Latitude')
cbar.set_label('Magnitude')
ax.set_zlabel('Depth_km')

font_size = 700
dpi = (5000)

plt.show()
```



```
In [102]: import plotly.graph_objects as go
# Yüksek çözünürlüklü dünya haritası verilerini çevrimiçi olarak alın
fig = go.Figure(go.Choroplethmapbox(
    geojson="https://raw.githubusercontent.com/johan/world.geo.json/master/countries.geo.json",
    locations=["USA", "CAN", "MEX", "RUS", "CHN"], # Örnek ülke kodları (ABD, Kanada, Meksika, Rusya, Çin)
    z=[1, 1, 1, 1, 1], # Ükelere atanacak değerler (hepsi 1 olarak ayarlanmıştır)
    colorscale='Jet', # Renk skalası adı (Viridis, YlGnBu, Jet vb.)
    zmin=3,
    zmax=8,
    marker_opacity=0.9, # Ülke sınırlarının opaklığı
    marker_line_width=1, # Ülke sınırlarının kenarlık kalınlığı
))
# Örnek deprem verilerini oluşturun
deprem_verileri = {
    'Longitude': x,
    'Latitude': y,
    'Magnitude': colors,
}
# Scatter plot ile deprem verilerini ekleyin
fig.add_trace(go.Scattermapbox(
    lat=deprem_verileri['Latitude'],
    lon=deprem_verileri['Longitude'],
    mode='markers',
    marker=dict(
        size=deprem_verileri['Magnitude'] * 2, # Magnitude değerine göre nokta boyutlarını belirleme
        color=deprem_verileri['Magnitude'], # Magnitude değerine göre renk skalasını belirleme
        colorscale='Jet', # Renk skalası adı (Viridis, YlGnBu, Jet vb.)
    ),
))
# Harita düzenini ve stilini belirleyin
fig.update_layout(
    mapbox_style="carto-positron", # Harita stilini belirleme (diğer stiller için: "open-street-map", "stamen-terrain" vb.)
    mapbox_zoom=6, # Harita yakınlaştırma düzeyini belirleme
    mapbox_center={"lat": 30.000, "lon": 30.0000}, # Harita merkezini belirleme (ABD'nin merkezi)
)
dpi = (9000)
font_size = 1000
# Grafiği görüntüleyin
fig.show()
```



```
In [103]: fig = plt.figure(figsize=(20, 10))

ax = fig.add_subplot(111, projection='3d')

ax.scatter(x, y, z, c=colors, s=sizes, cmap='jet')

cbar = plt.colorbar(ax.scatter(x, y, z, c=colors, s=sizes, cmap='jet'))

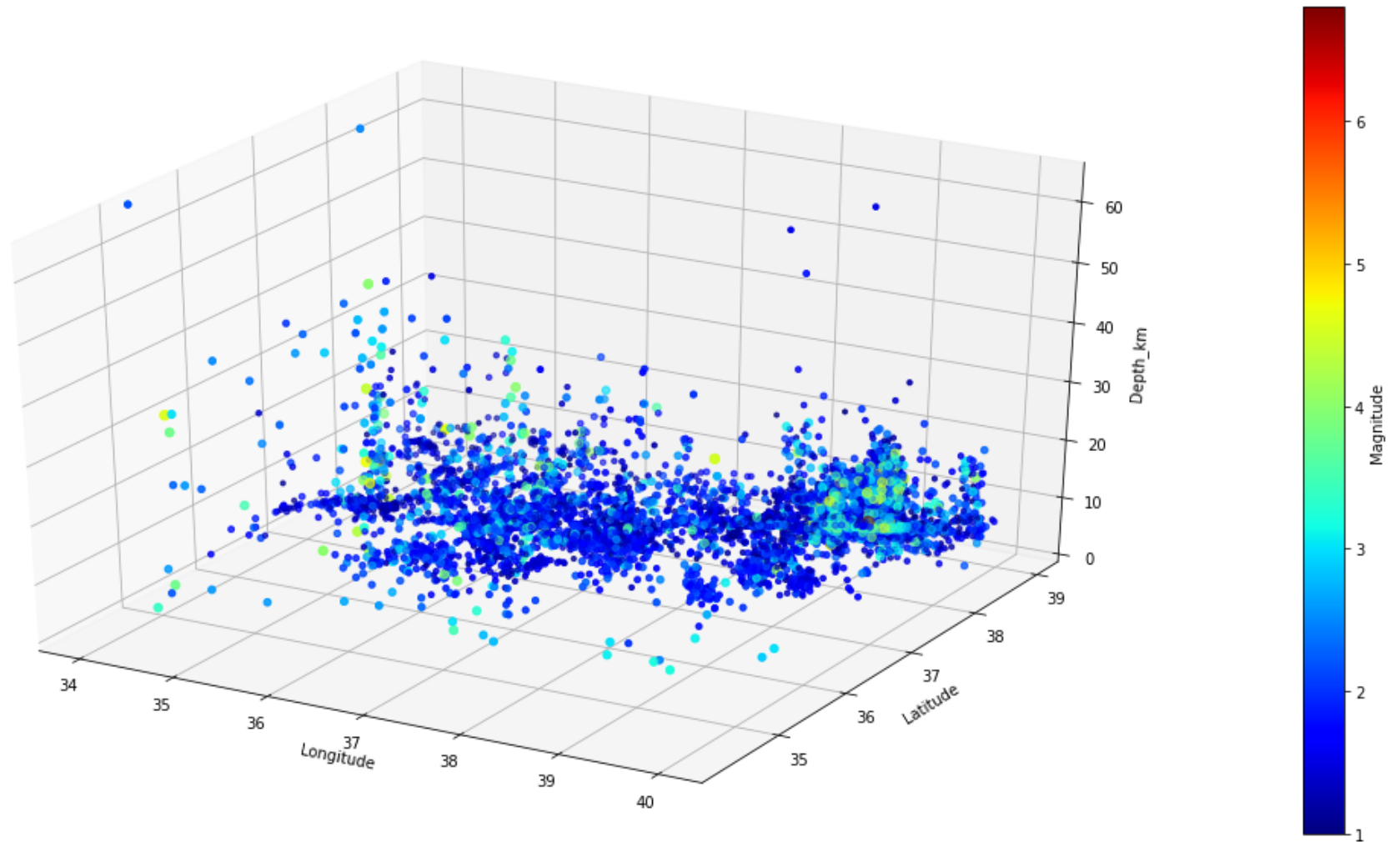
cbar.set_label('Magnitude')

ax.set_xlabel('Longitude')
ax.set_ylabel('Latitude')
ax.set_zlabel('Depth_km')

dpi = (5000)
font_size = 1000

ax.set_zlim(0, 65)

plt.show()
```



```
In [78]: x = Doublet_EAF_35F2.iloc[:,1].values  
y = Doublet_EAF_35F2.iloc[:,2].values  
z = Doublet_EAF_35F2.iloc[:,0].values  
colors = Doublet_EAF_35F2.iloc[:,4].values  
sizes = Doublet_EAF_35F2.iloc[:,4].values*8
```



In [79]:

z

Out[79]: array(['05/02/2023 19:21:54', '05/02/2023 18:21:36',  
              '05/02/2023 17:10:36', ..., '02/01/2020 00:56:49',  
              '01/01/2020 20:36:53', '01/01/2020 19:37:04'], dtype=object)

```
In [81]: from datetime import datetime
# Verileri oluştur
timestamps = z
# Zaman damgalarını aylara dönüştür
years = [datetime.strptime(timestamp, '%d/%m/%Y %H:%M:%S').year for timestamp in timestamps]
print(years) # Ay bilgilerini görüntüle
```

11/30

12/30

13/30

[illegible]

15/30

[illegible]



17/30

18/30

19/30

20/30

21/30

22/30

23/30

24/30



25/30

years

[illegible]

```
x = Doublet_EAF_35F2.iloc[:,1].values
y = Doublet_EAF_35F2.iloc[:,2].values
z = years
colors = Doublet_EAF_35F2.iloc[:,4].values
sizes = Doublet_EAF_35F2.iloc[:,4].values*8
```

```
In [84]: fig = plt.figure(figsize=(20, 10))

ax = fig.add_subplot(111, projection='3d')

ax.scatter(x, y, z, c=colors, s=sizes, cmap='jet')

cbar = plt.colorbar(ax.scatter(x, y, z, c=colors, s=sizes, cmap='jet'))

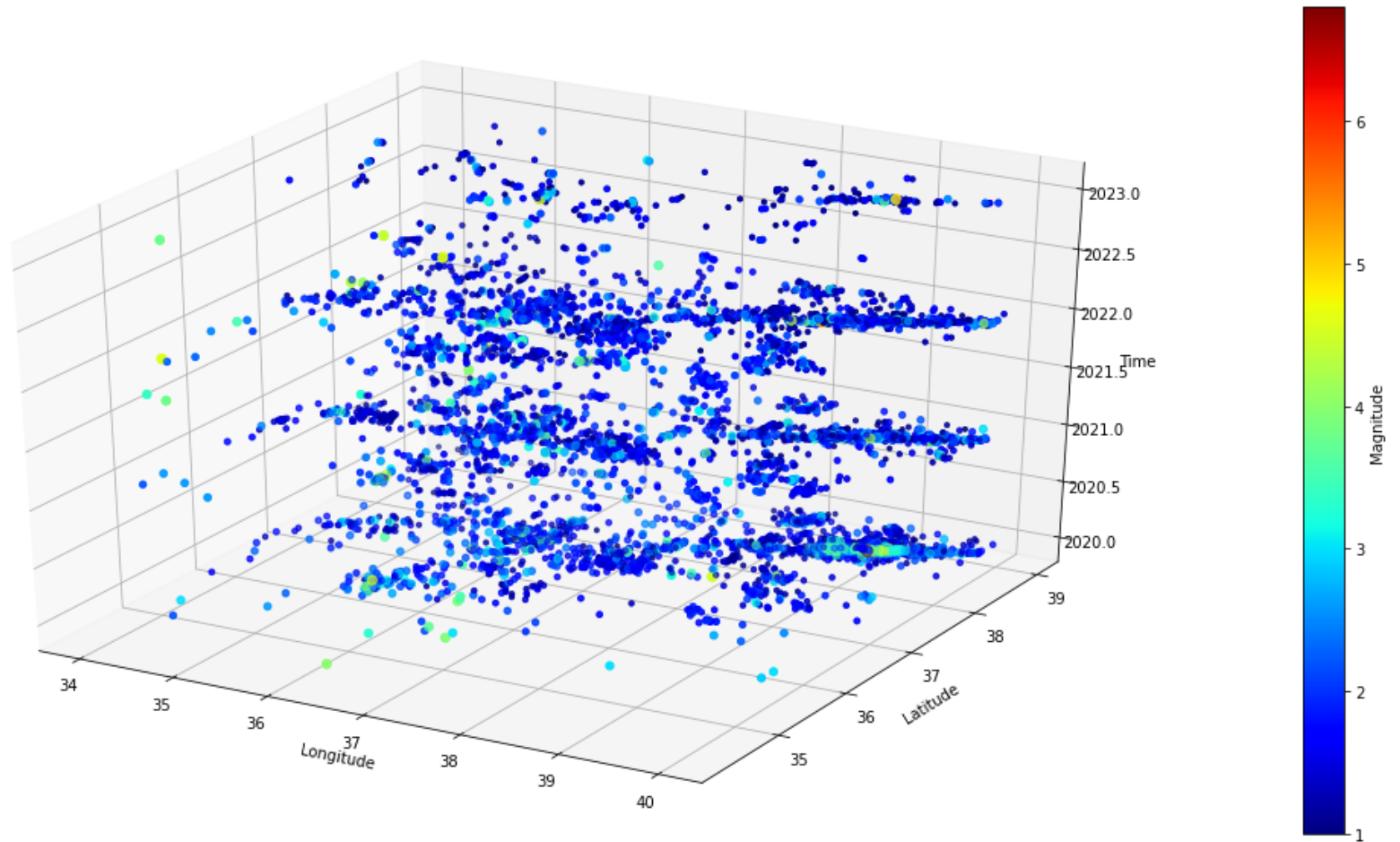
cbar.set_label('Magnitude')

ax.set_xlabel('Longitude')
ax.set_ylabel('Latitude')
ax.set_zlabel('Time')

font_size = 700

dpi = (5000)
font_size = 1000

plt.show()
```



```
In [85]: x = Doublet_EAF_35F2.iloc[:,1].values  
y = Doublet_EAF_35F2.iloc[:,2].values  
z = Doublet_EAF_35F2.iloc[:,4].values  
colors = years  
sizes = Doublet_EAF_35F2.iloc[:,4].values*8
```

```
In [86]: fig = plt.figure(figsize=(20, 10))

ax = fig.add_subplot(111, projection='3d')

ax.scatter(x, y, z, c=colors, s=sizes, cmap='jet')

cbar = plt.colorbar(ax.scatter(x, y, z, c=colors, s=sizes, cmap='jet'))

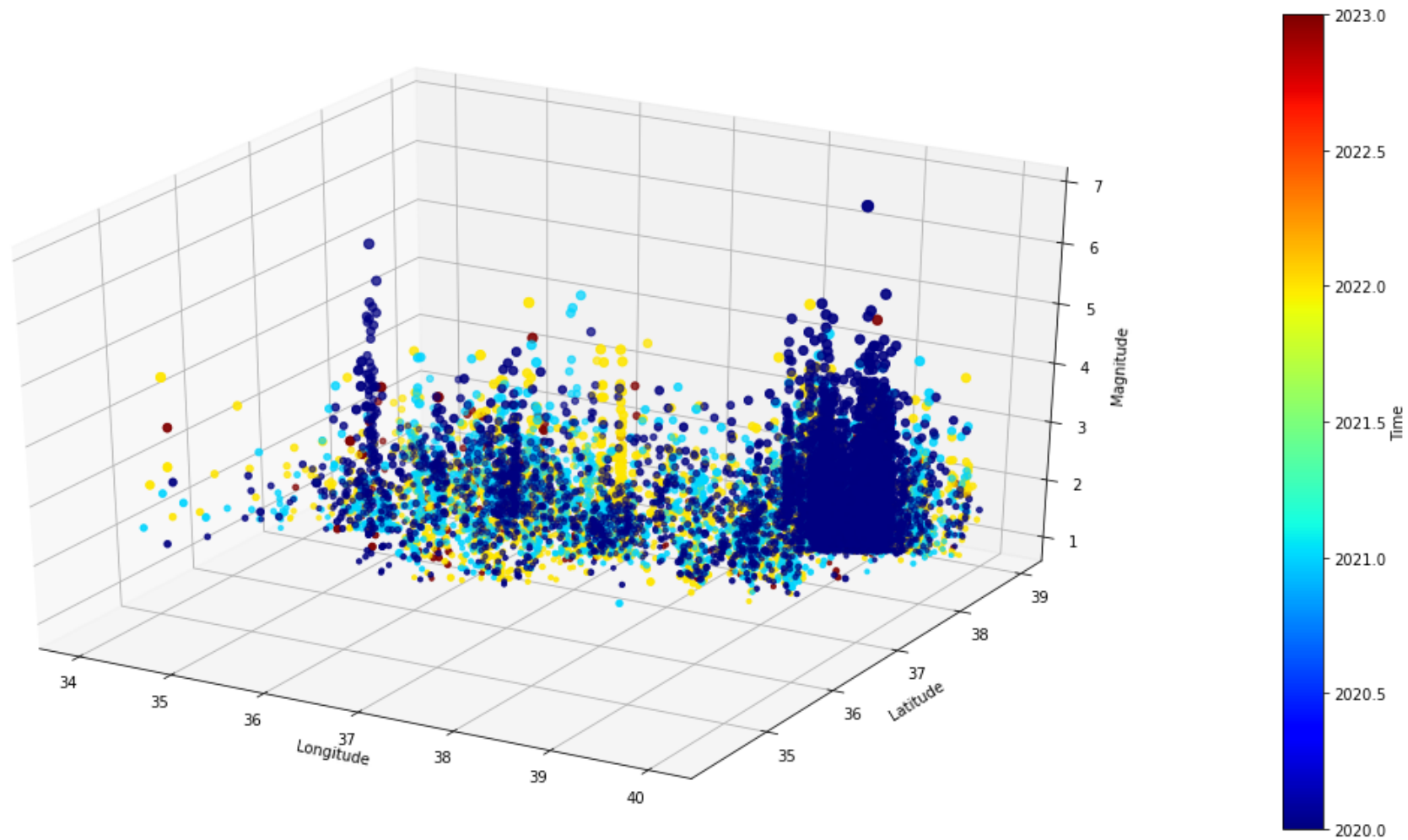
cbar.set_label('Time')

ax.set_xlabel('Longitude')
ax.set_ylabel('Latitude')
ax.set_zlabel('Magnitude')

font_size = 700

dpi = (5000)
font_size = 1000

plt.show()
```



In [ ]:

In [ ]: