In [1]:
```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
C:\Users\samil\anaconda3\lib\site-packages\scipy\__init__.py:138: UserWarning: A NumPy version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected version 1.24.4)
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion} is required for this version of "
```

In [2]:
```
pip install geopandas
```

```
Requirement already satisfied: shapely>=1.7.1 in c:\users\samil\anaconda3\lib\site-packages (from geopandas) (2.0.1)
Requirement already satisfied: pandas>=1.1.0 in c:\users\samil\anaconda3\lib\site-packages (from geopandas) (1.2.4)
Requirement already satisfied: packaging in c:\users\samil\anaconda3\lib\site-packages (from geopandas) (20.9)
Requirement already satisfied: pyproj>=3.0.1 in c:\users\samil\anaconda3\lib\site-packages (from geopandas) (3.5.0)
Requirement already satisfied: fiona>=1.8.19 in c:\users\samil\anaconda3\lib\site-packages (from geopandas) (1.9.4.post1)
Requirement already satisfied: click-plugins>=1.0 in c:\users\samil\anaconda3\lib\site-packages (from fiona>=1.8.19->geopandas) (1.1.1)
Requirement already satisfied: importlib-metadata in c:\users\samil\anaconda3\lib\site-packages (from fiona>=1.8.19->geopandas) (3.10.0)
Requirement already satisfied: click~=8.0 in c:\users\samil\anaconda3\lib\site-packages (from fiona>=1.8.19->geopandas) (8.1.5)
Requirement already satisfied: cligj>=0.5 in c:\users\samil\anaconda3\lib\site-packages (from fiona>=1.8.19->geopandas) (0.7.2)
Requirement already satisfied: attrs>=19.2.0 in c:\users\samil\anaconda3\lib\site-packages (from fiona>=1.8.19->geopandas) (20.3.0)
Requirement already satisfied: certifi in c:\users\samil\anaconda3\lib\site-packages (from fiona>=1.8.19->geopandas) (2020.12.5)
Requirement already satisfied: six in c:\users\samil\anaconda3\lib\site-packages (from fiona>=1.8.19->geopandas) (1.15.0)
Requirement already satisfied: colorama in c:\users\samil\anaconda3\lib\site-packages (from click~=8.0->fiona>=1.8.19->geopandas) (0.4.4)
Requirement already satisfied: pytz>=2017.3 in c:\users\samil\anaconda3\lib\site-packages (from pandas>=1.1.0->geopandas) (2021.1)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\samil\anaconda3\lib\site-packages (from pandas>=1.1.0->geopandas) (2.8.1)
Requirement already satisfied: numpy>=1.16.5 in c:\users\samil\anaconda3\lib\site-packages (from pandas>=1.1.0->geopandas) (1.24.4)
Requirement already satisfied: zipp>=0.5 in c:\users\samil\anaconda3\lib\site-packages (from importlib-metadata->fiona>=1.8.19->geopandas) (3.4.1)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\samil\anaconda3\lib\site-packages (from packaging->geopandas) (2.4.7)
Note: you may need to restart the kernel to use updated packages.
```

2023 Earthquake Doublet visulation

In [267]:
```
import geopandas as gpd
```

In [268]:
```
raw_dataset=pd.read_csv("Doublet_EAF_35.csv",sep=",")
```

In [269]:
```
Doublet_EAF_35 = raw_dataset.copy()
Doublet_EAF_35.head()
```

Out[269]:

	Date	Longitude	Latitude	Depth_km	Magnitude
0	2023-07-13 12:32:21	37.208	38.289	12.44	3.8
1	2023-07-09 02:27:35	36.359	37.973	7.00	3.6
2	2023-07-07 18:38:43	37.449	38.023	7.56	3.5
3	2023-07-05 05:55:12	36.633	38.034	6.97	3.6
4	2023-07-03 06:31:50	35.888	36.104	23.45	3.6

In [270]:
```
Doublet_EAF_35.describe()
```

Out[270]:

	Longitude	Latitude	Depth_km	Magnitude
count	1696.000000	1696.000000	1696.000000	1696.000000
mean	37.170608	37.741194	8.275596	3.938797
std	0.877128	0.653008	3.088543	0.438950
min	34.004000	34.090000	0.000000	3.500000
25%	36.503750	37.518000	7.000000	3.600000
50%	37.062500	37.996500	7.010000	3.800000
75%	37.812250	38.103250	8.550000	4.100000
max	39.989000	38.977000	32.910000	7.700000

In [271]:
```
Doublet_EAF_35.shape
```

Out[271]:
```
(1696, 5)
```

In [272]:
```
x= Doublet_EAF_35.iloc[:, 1].values
```

In [273]:
```
y=Doublet_EAF_35.iloc[:, 2].values
```

In [274]:
```
z = Doublet_EAF_35.iloc[:, 3].values
```

In [275]:
```
colors= Doublet_EAF_35.iloc[:, 4].values
```

In [276]:
```
sizes = Doublet_EAF_35.iloc[:, 4].values * 30
```

In [277]:
```
x
```

Out[277]:
```
array([37.208, 36.359, 37.449, ..., 37.043, 34.162, 36.403])
```

In [278]:
```
y
```

Out[278]:
```
array([38.289, 37.973, 38.023, ..., 37.288, 35.018, 37.208])
```

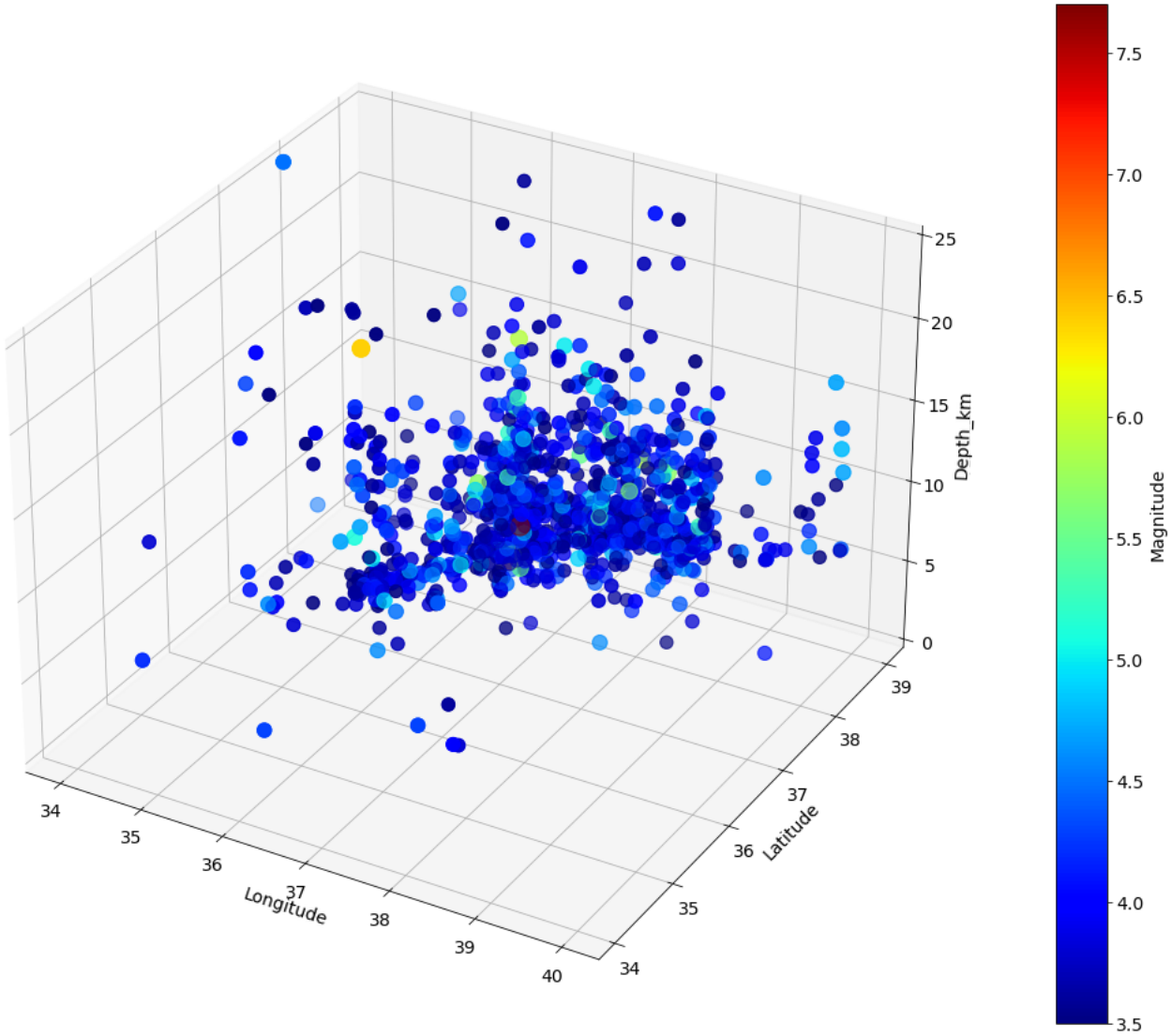
In [279]:
```
colors
```

Out[279]:
```
array([3.8, 3.6, 3.5, ..., 7.7, 3.8, 4.6])
```

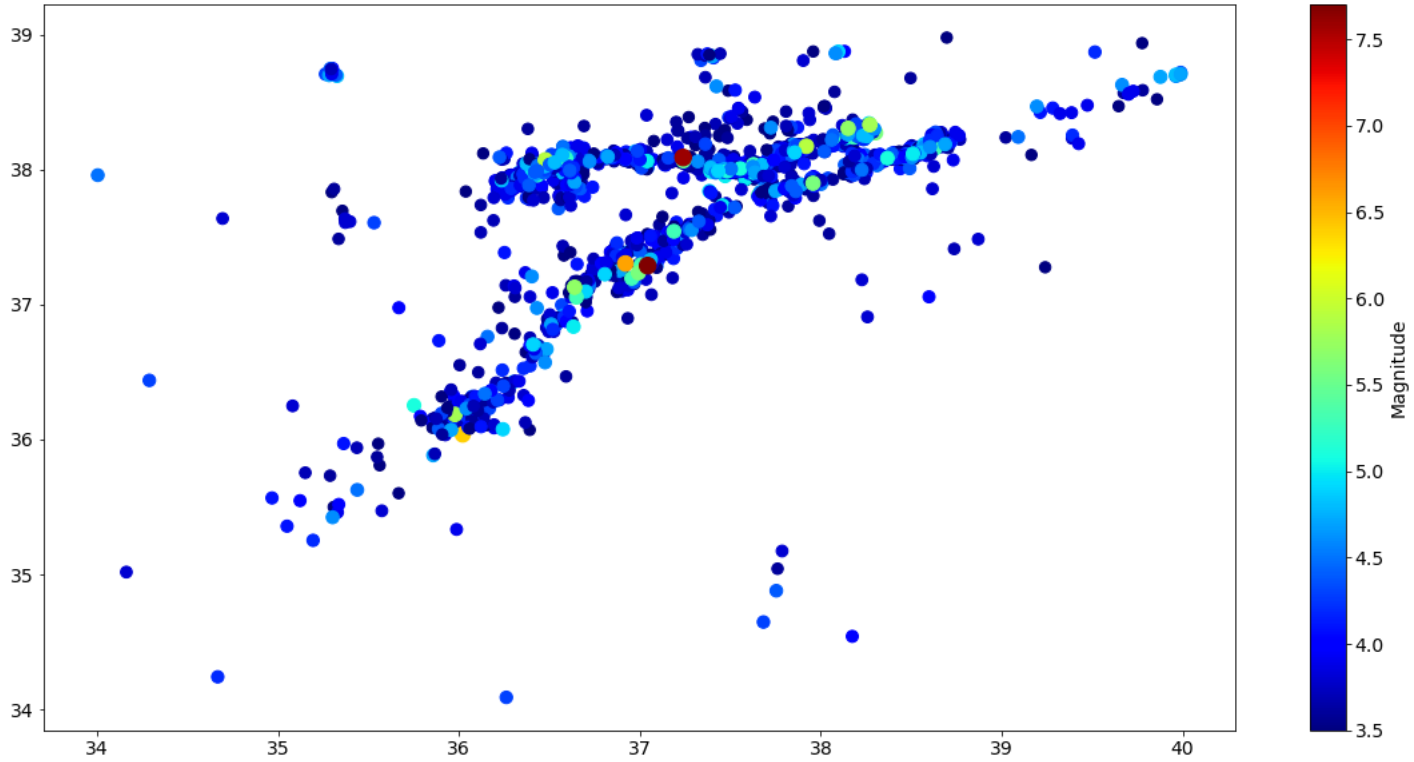
In [280]:
```
sizes
```

Out[280]:
```
array([114., 108., 105., ..., 231., 114., 138.])
```

In [281]:
```
# 3D scatter plot oluşturun

fig = plt.figure(figsize=(25, 15))
ax = fig.add_subplot(111, projection='3d')

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

# Renk çubuğunu ekleyin
cbar = plt.colorbar(ax.scatter(x, y, z, c=colors, s=sizes, cmap='jet'))
cbar.set_label('Magnitude')

# Eksen etiketlerini ayarlayın
ax.set_xlabel('Longitude')
ax.set_ylabel('Latitude')
ax.set_zlabel('Depth_km')

font_size = 700

dpi = (5000)
font_size = 1000

ax.set_zlim(0, 25)

# Grafiği gösterin
plt.show()
```


Depth_km
0
5
10
15
20
25
Longitude
34
35
36
37
38
39
40
Latitude
34
35
36
37
38
39
Magnitude
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5

In [282]:
```
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(20, 10))
my_cmap = plt.get_cmap('hot')

x= Doublet_EAF_35.iloc[:, 1].values

y=Doublet_EAF_35.iloc[:, 2].values

plt.rcParams.update({'font.size': 14})

colors= Doublet_EAF_35.iloc[:, 4].values

sizes = Doublet_EAF_35.iloc[:, 4].values * 20

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)
font_size = 1000




plt.show()
```



In [ ]:

In [ ]:

In [283]:
```
pip install plotly
```

```
Requirement already satisfied: plotly in c:\users\samil\anaconda3\lib\site-packages (5.15.0)Note: you may need to restart the kernel to use updated packages.

Requirement already satisfied: packaging in c:\users\samil\anaconda3\lib\site-packages (from plotly) (20.9)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\samil\anaconda3\lib\site-packages (from plotly) (8.2.2)
Requirement already satisfied: pyparsing>=2.0.2 in c:\users\samil\anaconda3\lib\site-packages (from packaging->plotly) (2.4.7)
```

In [ ]:

In [284]:
```
x
```

Out[284]:
```
array([37.208, 36.359, 37.449, ..., 37.043, 34.162, 36.403])
```

In [285]:
```
y
```

Out[285]:
```
array([38.289, 37.973, 38.023, ..., 37.288, 35.018, 37.208])
```

In [286]:
```
colors
```

Out[286]:
```
array([3.8, 3.6, 3.5, ..., 7.7, 3.8, 4.6])
```

In [287]:
```
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],  # Ülkelere 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 [219]:
```
raw_dataset=pd.read_csv("Doublet_35_day.csv",sep=",")
```

In [220]:
```
raw_dataset=pd.read_csv("Doublet_35_day.csv",sep=",")
```

In [221]:
```
Doublet_35_day= raw_dataset.copy()
Doublet_35_day.head()
```

Out[221]:

	Date	Longitude	Latitude	Depth_km	Magnitude
0	7.429	37.208	38.289	12.44	3.8
1	7.300	36.359	37.973	7.00	3.6
2	7.230	37.449	38.023	7.56	3.5
3	7.170	36.633	38.034	6.97	3.6
4	7.100	35.888	36.104	23.45	3.6

In [223]:
```
Doublet_35_day.describe()
```

Out[223]:

	Date	Longitude	Latitude	Depth_km	Magnitude
count	1696.000000	1696.000000	1696.000000	1696.000000	1696.000000
mean	2.876238	37.170608	37.741194	8.275596	3.938797
std	1.125235	0.877128	0.653008	3.088543	0.438950
min	2.100000	34.004000	34.090000	0.000000	3.500000
25%	2.200000	36.503750	37.518000	7.000000	3.600000
50%	2.330000	37.062500	37.996500	7.010000	3.800000
75%	2.899250	37.812250	38.103250	8.550000	4.100000
max	7.700000	39.989000	38.977000	32.910000	7.700000

In [204]:
```
x= Doublet_35_day.iloc[:, 1].values
```

In [205]:
```
y= Doublet_35_day.iloc[:, 2].values
```

In [206]:
```
z= Doublet_35_day.iloc[:, 4].values
```

In [207]:
```
colors = Doublet_35_day.iloc[:, 0].values
```

In [208]:
```
sizes = Doublet_35_day.iloc[:, 0].values*30
```

In [209]:
```
x
```

Out[209]:
```
array([37.208, 36.359, 37.449, ..., 37.043, 34.162, 36.403])
```

In [210]:
```
y
```

Out[210]:
```
array([38.289, 37.973, 38.023, ..., 37.288, 35.018, 37.208])
```

In [211]:
```
z
```

Out[211]:
```
array([3.8, 3.6, 3.5, ..., 7.7, 3.8, 4.6])
```

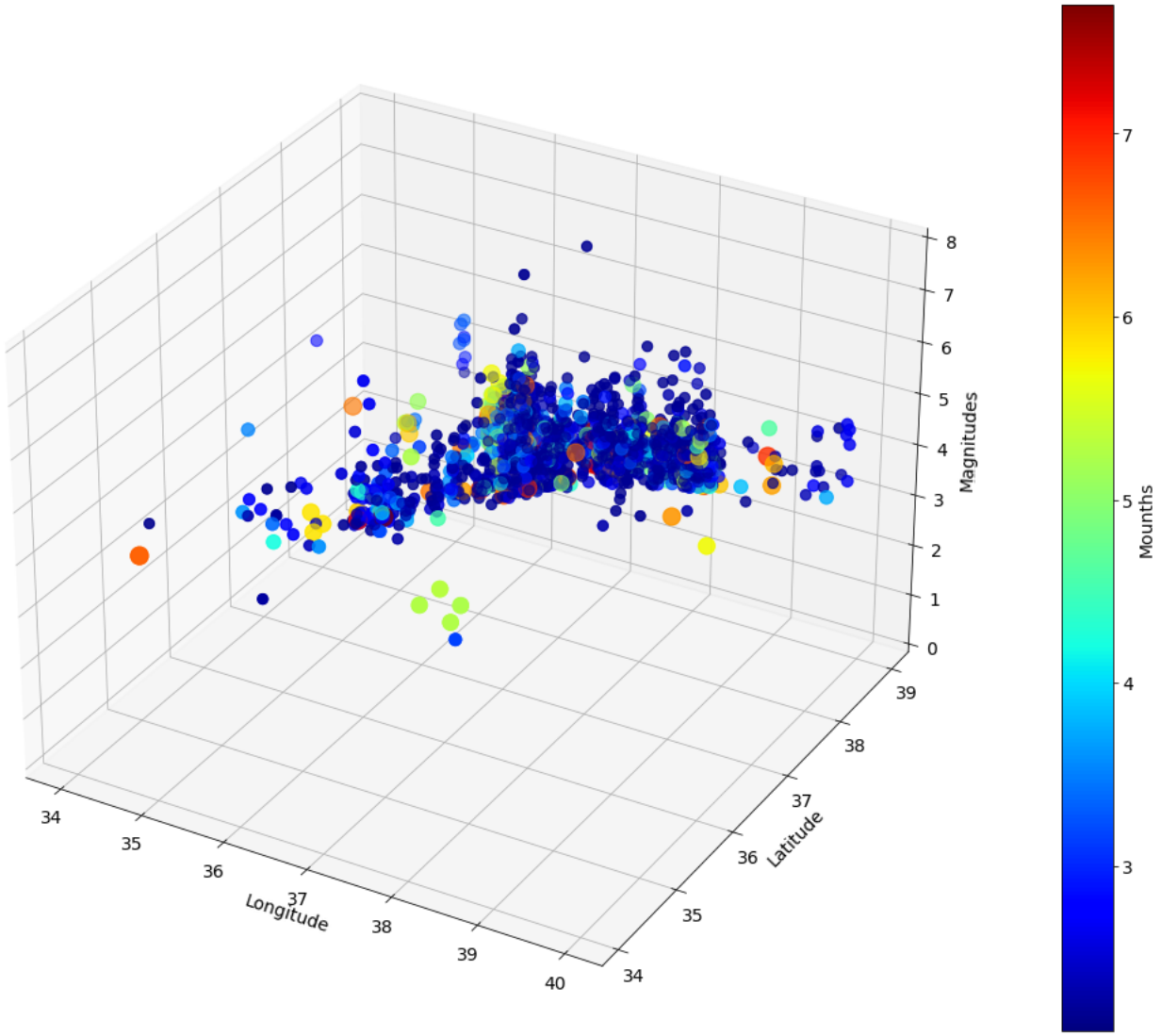
In [212]:
```
colors
```

Out[212]:
```
array([7.429, 7.3  , 7.23 , ..., 2.2  , 2.17 , 2.1  ])
```

In [213]:
```
sizes
```

Out[213]:
```
array([222.87, 219.  , 216.9 , ...,  66.  ,  65.1 ,  63.  ])
```

In [225]:
```
# 3D scatter plot oluşturun

fig = plt.figure(figsize=(25, 15))
ax = fig.add_subplot(111, projection='3d')

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

# Renk çubuğunu ekleyin
cbar = plt.colorbar(ax.scatter(x, y, z, c=colors, s=sizes, cmap='jet'))
cbar.set_label('Mounths')

# Eksen etiketlerini ayarlayın
ax.set_xlabel('Longitude')
ax.set_ylabel('Latitude')
ax.set_zlabel('Magnitudes')

font_size = 12

dpi = 100


ax.set_zlim(0, 8)

# Grafiği gösterin
plt.show()
```



In [ ]:

In [226]:
```
# before earthquakes than 2023 Earthquake Doublet  visulation
```

In [289]:
```
raw_dataset=pd.read_csv("Doublet_EAF_35F.csv",sep=",")
```

In [290]:
```
Doublet_EAF_35F = raw_dataset.copy()
Doublet_EAF_35F.head()
```

Out[290]:

	Date	Longitude	Latitude	Depth	Magnitude
0	05/02/2023 06:06:55	34.162	35.018	30.50	3.8
1	03/02/2023 11:05:08	36.403	37.208	7.00	4.6
2	29/01/2023 16:12:39	35.784	35.884	18.35	4.4
3	21/01/2023 14:27:54	36.374	34.843	18.05	4.1
4	21/01/2023 13:31:59	36.271	34.820	21.33	4.3

In [291]:
```
Doublet_EAF_35.shape
```

Out[291]:
```
(1696, 5)
```

In [292]:
```
x= Doublet_EAF_35F.iloc[:, 1].values
```

In [293]:
```
y= Doublet_EAF_35F.iloc[:, 2].values
```

In [294]:
```
z= Doublet_EAF_35F.iloc[:, 3].values
```

In [295]:
```
colors= Doublet_EAF_35F.iloc[:, 4].values
```

In [296]:
```
sizes = Doublet_EAF_35F.iloc[:, 4].values * 40
```

In [297]:
```
x
```

Out[297]:
```
array([34.162, 36.403, 35.784, ..., 38.   , 40.   , 38.   ])
```

In [298]:
```
y
```

Out[298]:
```
array([35.018, 37.208, 35.884, ..., 40.   , 39.81 , 40.   ])
```

In [299]:
```
z
```

Out[299]:
```
array([30.5 ,  7.  , 18.35, ...,  1.  , 21.  ,  1.  ])
```

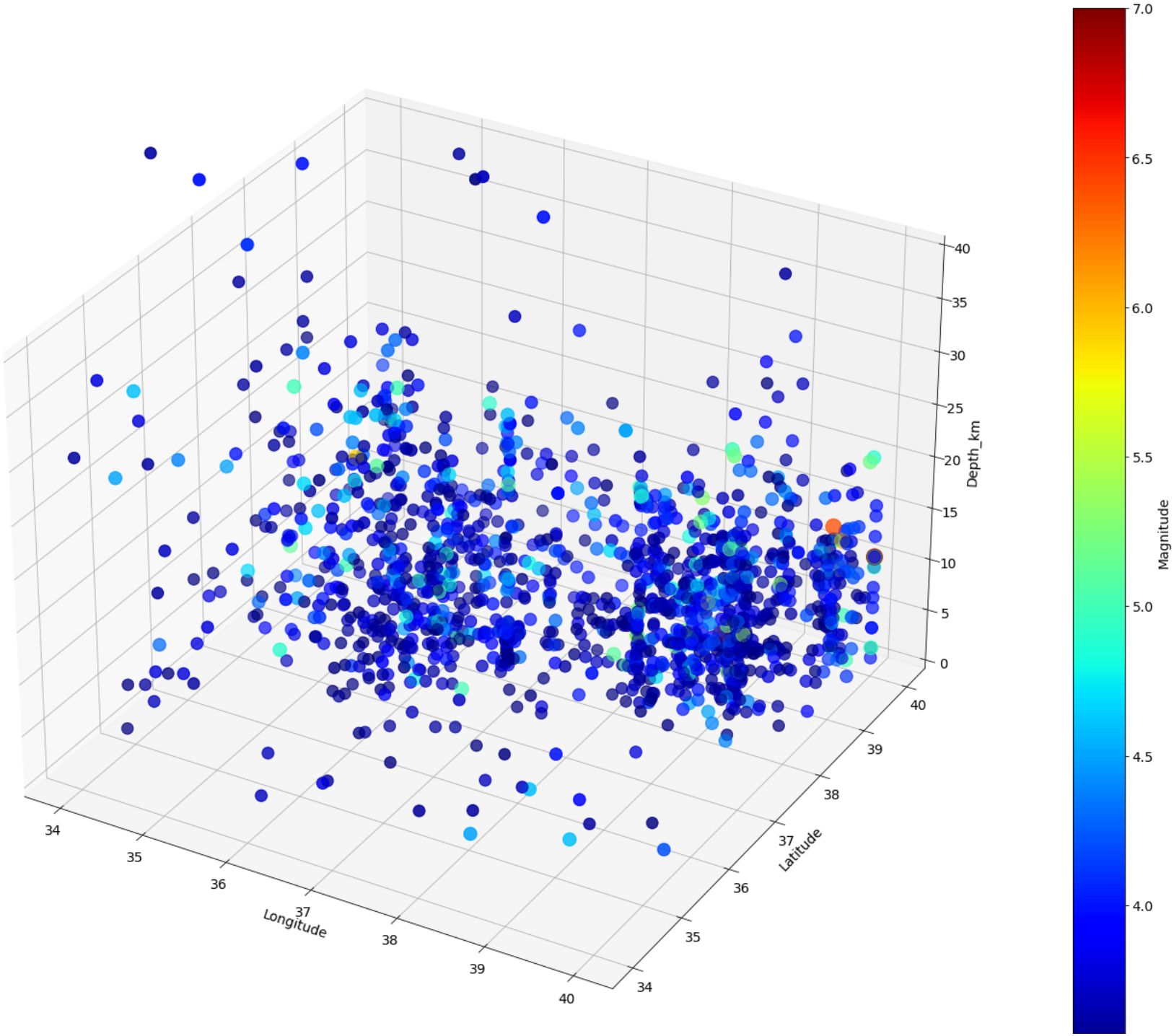
In [300]:
```
colors
```

Out[300]:
```
array([3.8, 4.6, 4.4, ..., 5. , 5.2, 4. ])
```

In [301]:
```
sizes
```

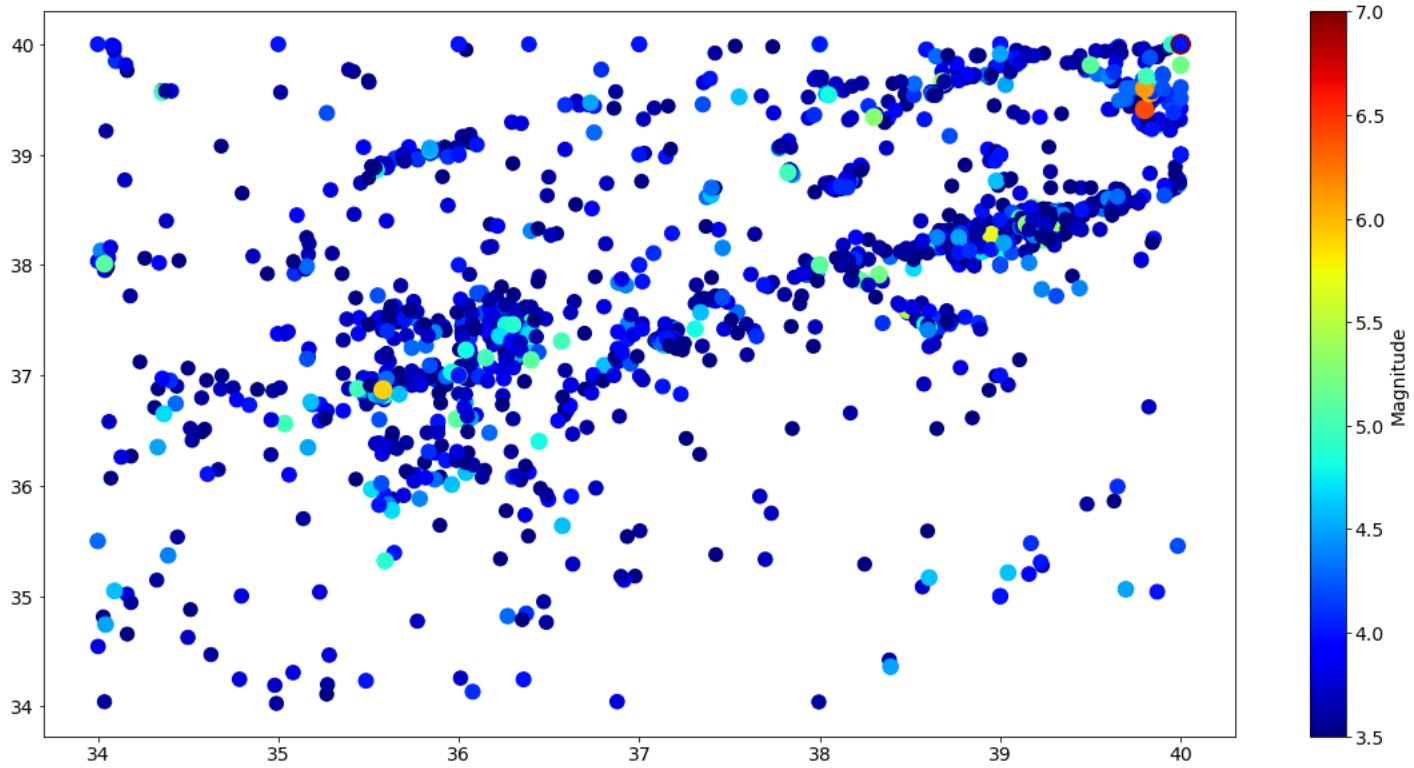
Out[301]:
```
array([152., 184., 176., ..., 200., 208., 160.])
```

In [302]:
```
# 3D scatter plot oluşturun

fig = plt.figure(figsize=(30, 20))
ax = fig.add_subplot(111, projection='3d')

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

# Renk çubuğunu ekleyin
cbar = plt.colorbar(ax.scatter(x, y, z, c=colors, s=sizes, cmap='jet'))
cbar.set_label('Magnitude')

# Eksen etiketlerini ayarlayın
ax.set_xlabel('Longitude')
ax.set_ylabel('Latitude')
ax.set_zlabel('Depth_km')

font_size = 700

dpi = (5000)
font_size = 1000

ax.set_zlim(0, 40)

# Grafiği gösterin
plt.show()
```

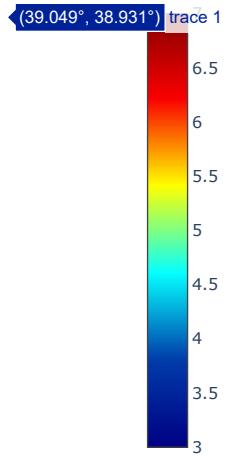

Longitude
34
35
36
37
38
39
40
Latitude
34
35
36
37
38
39
40
Depth_km
0
5
10
15
20
25
30
35
40
Magnitude
4.0
4.5
5.0
5.5
6.0
6.5
7.0


3.5

In [303]:
```
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(20, 10))
my_cmap = plt.get_cmap('hot')

x= Doublet_EAF_35F.iloc[:, 1].values

y=Doublet_EAF_35F.iloc[:, 2].values

plt.rcParams.update({'font.size': 14})

colors= Doublet_EAF_35F.iloc[:, 4].values

sizes = Doublet_EAF_35F.iloc[:, 4].values * 30

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


ax.scatter(x, y,  c=colors, s=sizes, cmap='jet')
cbar = plt.colorbar(ax.scatter(x, y, 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)
font_size = 1000




plt.show()
```


40
39
38
37
36
35
34
34
35
36
37
38
39
40
7.0
6.5
6.0
5.5
5.0
4.5
4.0
3.5
Magnitude

In [304]:
```
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],  # Ülkelere atanacak değerler (hepsi 1 olarak ayarlanmıştır)
    colorscale='Jet',  # Renk skalası adı (Viridis, YlGnBu, Jet vb.)
    zmin=3,
    zmax=7,
    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 [ ]:

In [ ]:

In [ ]:

In [ ]: