```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import folium
from folium.plugins import MarkerCluster, HeatMap
from folium.map import FeatureGroup
from google.colab import drive
from sklearn.metrics import r2_score
```

```
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
df = pd.read_excel('/content/drive/My Drive/V2-global-bleaching-and-environmental-data (2).xlsx')
pd.set_option('display.max_columns', None)
```

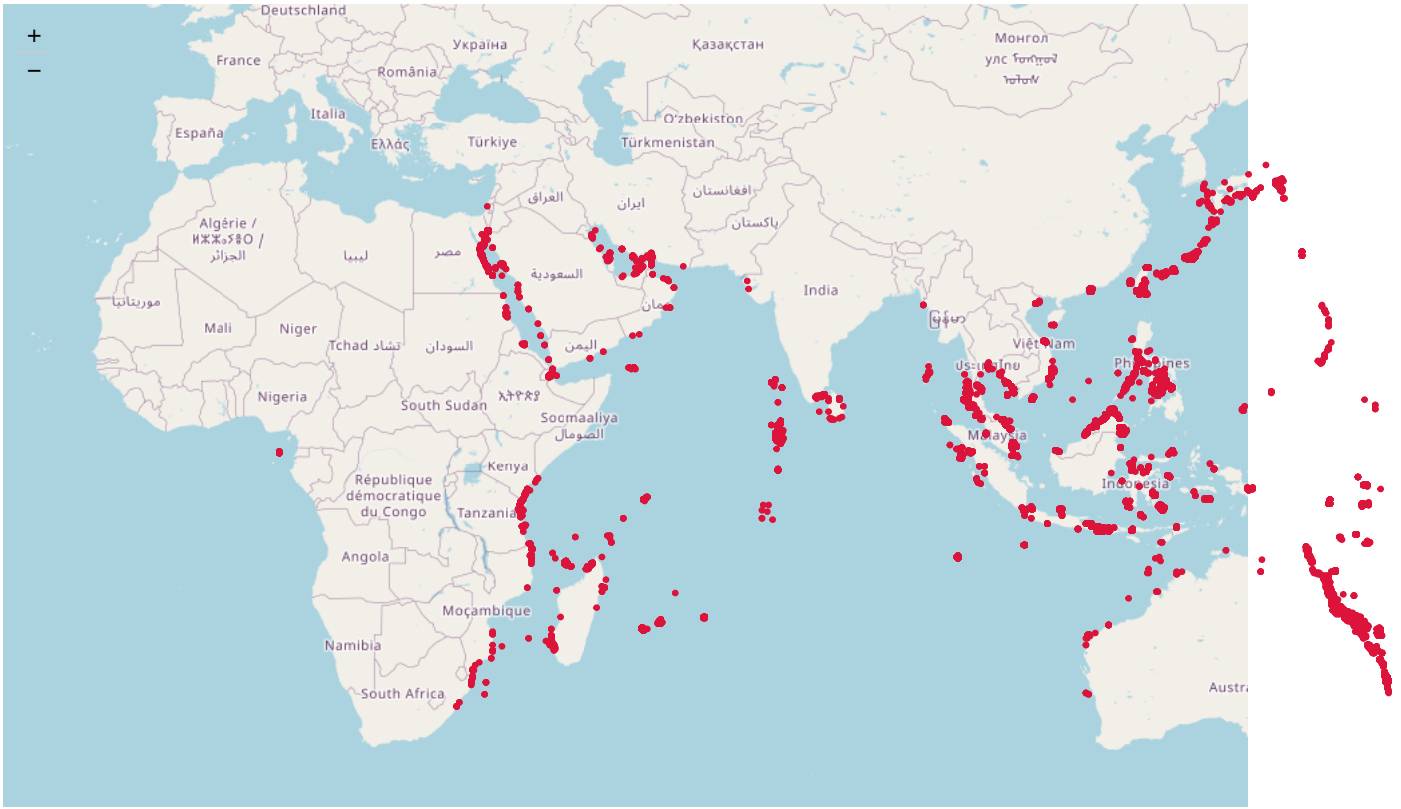
```
df = df.replace('nd',np.nan)
df
```

	Site_ID	Sample_ID	Data_Source	Latitude_Degrees	Longitude_Degrees	Ocean_Name	Reef_ID	Realm_Name	Ecoregion_
0	2501.0	10324336.0	Donner	23.1630	-82.5260	Atlantic	NaN	Tropical Atlantic	Cub Cayman Is
1	3467.0	10324754.0	Donner	-17.5750	-149.7833	Pacific	NaN	Eastern Indo-Pacific	Society Is F Poly
2	1794.0	10323866.0	Donner	18.3690	-64.5640	Atlantic	NaN	Tropical Atlantic	Hispa Puerto Ric Lesser A
3	8647.0	10328028.0	Donner	17.7600	-64.5680	Atlantic	NaN	Tropical Atlantic	Hispa Puerto Ric Lesser A
4	8648.0	10328029.0	Donner	17.7690	-64.5830	Atlantic	NaN	Tropical Atlantic	Hispa Puerto Ric Lesser A
...	...	...	...	...	...	...	...	...	
41356	15446.0	10310562.0	Reef_Check	-8.3651	116.0844	Pacific	116.5.3.9E.8.21.54.4S	Central Indo-Pacific	Lesser S Islands and
41357	15456.0	10310527.0	Reef_Check	-8.3473	116.0503	Pacific	116.3.1.1E.8.20.50.2S	Central Indo-Pacific	Lesser S Islands and
41358	15456.0	10310527.0	Reef_Check	-8.3473	116.0503	Pacific	116.3.1.1E.8.20.50.2S	Central Indo-Pacific	Lesser S Islands and
41359	15457.0	10310536.0	Reef_Check	-8.3445	116.0629	Pacific	116.3.46.548E.8.20.40.236S	Central Indo-Pacific	Lesser S Islands and
41360	15457.0	10310536.0	Reef_Check	-8.3445	116.0629	Pacific	116.3.46.548E.8.20.40.236S	Central Indo-Pacific	Lesser S Islands and

41361 rows × 62 columns

```
### Distribution of Corals Globally
m = folium.Map(location=[39.4699, 0.3763],zoom_start=2.5)

for lat,lon,i in zip(df["Latitude_Degrees"], df["Longitude_Degrees"],range(len(df))):
    ## Differentiate by time period
    # if df.loc[i,'Date_Year']>=2010:
    #   marker = folium.CircleMarker([lat, lon],radius=0.2*df.loc[i,'Percent_Bleaching']+1,
    #                                color='yellow',
    #                                fill_color='yellow')
```

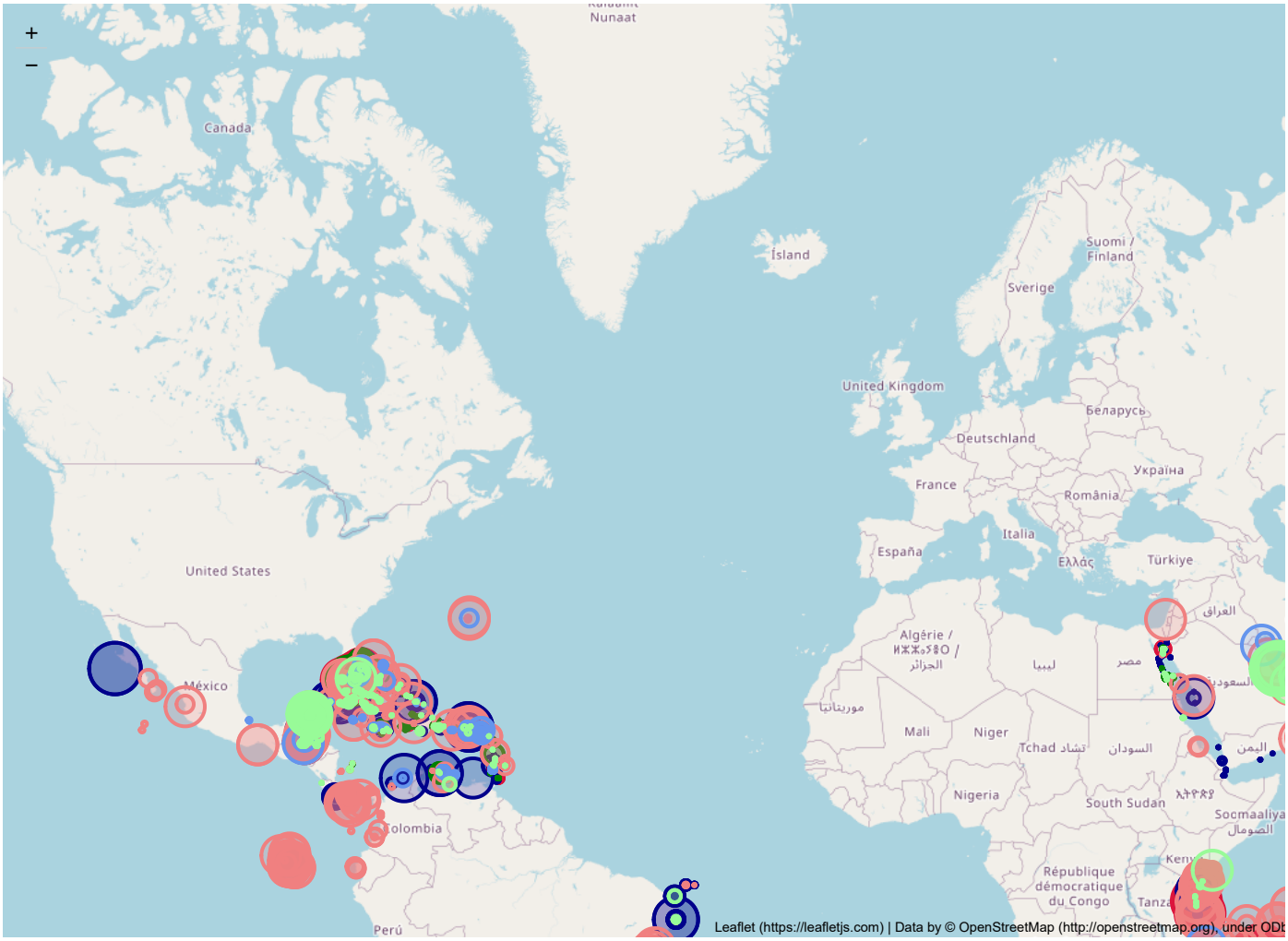
```
    #                             fill_color='yellow')
    #   marker.add_to(m)
    # elif 2000<=df.loc[i,'Date_Year']<2010:
    #   marker = folium.CircleMarker([lat, lon],radius=0.2*df.loc[i,'Percent_Bleaching']+1,
    #                                color='crimson',
    #                                fill_color='crimson')
    #   marker.add_to(m)
    marker = folium.CircleMarker([lat, lon],radius=1,
                                 color='crimson',
                                 fill_color='crimson')
    marker.add_to(m)
m
```



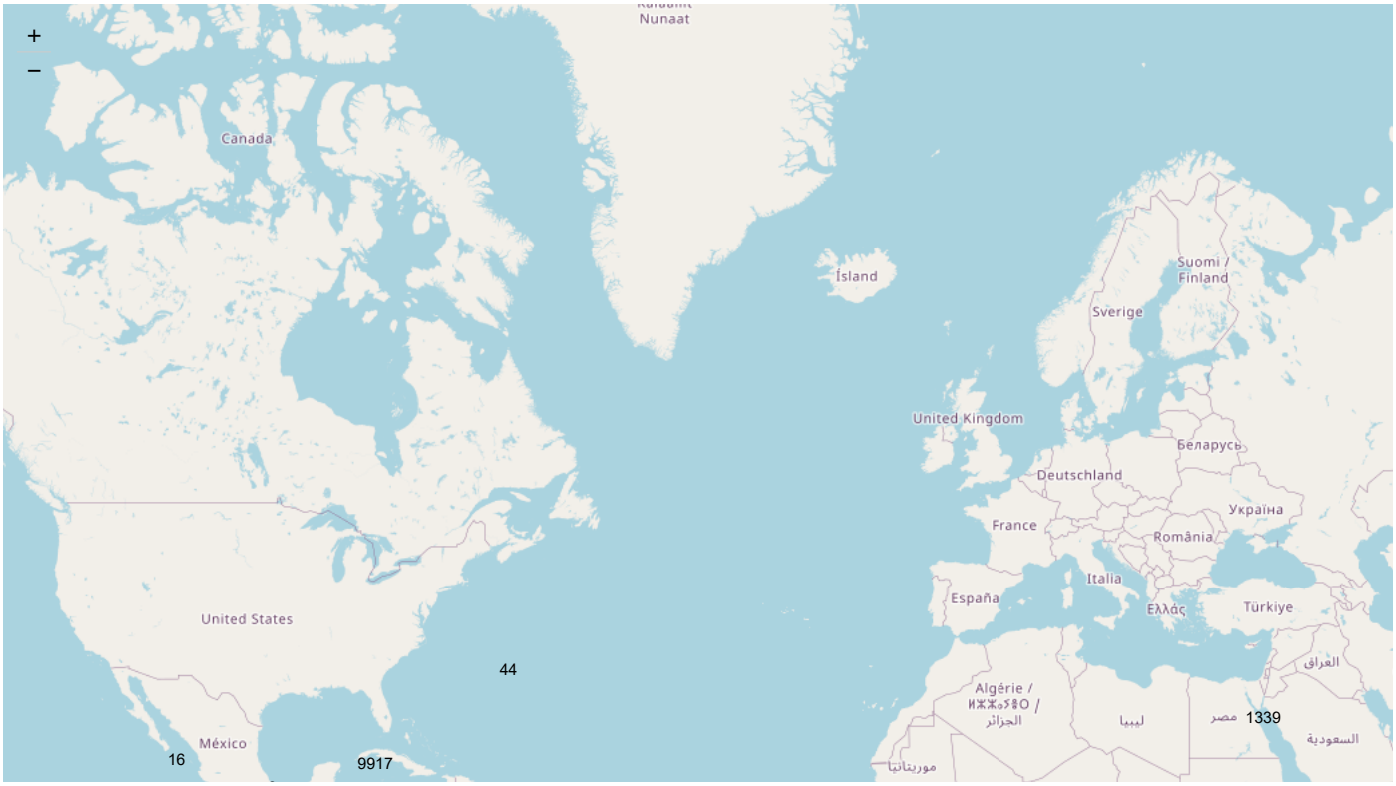

```
from branca.element import Template, MacroElement

template = """
{% macro html(this, kwargs) %}

<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>jQuery UI Draggable - Default functionality</title>
  <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
```

```
    <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
    <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>

    <script>
    $( function() {
      $( "#maplegend" ).draggable({
                      start: function (event, ui) {
                          $(this).css({
                              right: "auto",
                              top: "auto",
                              bottom: "auto"
                          });
                      }
                  });
});

    </script>
</head>
<body>


<div id='maplegend' class='maplegend'
    style='position: absolute; z-index:9999; border:2px solid grey; background-color:rgba(255, 255, 255, 0.8);
     border-radius:6px; padding: 10px; font-size:14px; right: 20px; bottom: 20px;'>

<div class='legend-title'>Legend</div>
<div class='legend-scale'>
  <ul class='legend-labels'>

    <li><span style='background:crimson;opacity:1;'></span>Modern Strong EN Year</li>
    <li><span style='background:DarkBlue;opacity:1;'></span>Modern Strong LN Year</li>
    <li><span style='background:green;opacity:1;'></span>Modern Neutral Year</li>
    <li><span style='background:LightCoral;opacity:1;'></span>Past Strong EN Year</li>
    <li><span style='background:CornflowerBlue;opacity:1;'></span>Past Strong LN Year</li>
    <li><span style='background:PaleGreen;opacity:1;'></span>Past Neutral Year</li>


  </ul>
</div>
</div>

</body>
</html>

<style type='text/css'>
  .maplegend .legend-title {
    text-align: left;
    margin-bottom: 5px;
    font-weight: bold;
    font-size: 90%;
    }
  .maplegend .legend-scale ul {
    margin: 0;
    margin-bottom: 5px;
    padding: 0;
    float: left;
    list-style: none;
    }
  .maplegend .legend-scale ul li {
    font-size: 80%;
    list-style: none;
    margin-left: 0;
    line-height: 18px;
    margin-bottom: 2px;
    }
  .maplegend ul.legend-labels li span {
    display: block;
    float: left;
    height: 16px;
    width: 30px;
    margin-right: 5px;
    margin-left: 0;
    border: 1px solid #999;
    }
  .maplegend .legend-source {
    font-size: 80%;
    color: #777;
```

```
    clear: both;
    }
  .maplegend a {
    color: #777;
    }
</style>
{% endmacro %}"""

macro = MacroElement()
macro._template = Template(template)

    # <li><span style='background:crimson;opacity:1;'></span>Modern Strong EN Year</li>
    # <li><span style='background:DarkBlue;opacity:1;'></span>Modern Strong LN Year</li>
    # <li><span style='background:green;opacity:1;'></span>Modern Neutral Year</li>
    # <li><span style='background:LightCoral;opacity:1;'></span>Past Strong EN Year</li>
    # <li><span style='background:CornflowerBlue;opacity:1;'></span>Past Strong LN Year</li>
    # <li><span style='background:PaleGreen;opacity:1;'></span>Past Neutral Year</li>

from folium.map import FeatureGroup
m = folium.Map(location=[39.4699, 0.3763],zoom_start=2.5)
# reefcheck = df[df['Data_Source']=='Reef_Check'].reset_index()

modernSEN = FeatureGroup(name="Modern Strong El Nino").add_to(m)
modernSLN = FeatureGroup(name="Modern Strong La Nina").add_to(m)
modernmod = FeatureGroup(name="Modern Neutral").add_to(m)
oldSEN = FeatureGroup(name="Past Strong El Nino").add_to(m)
oldSLN = FeatureGroup(name="Past Strong La Nina").add_to(m)
oldmod = FeatureGroup(name="Past Neutral").add_to(m)

for lat,lon,i in zip(df["Latitude_Degrees"], df["Longitude_Degrees"],range(len(df))):

    # Differentiate by time period
    if (df.loc[i,'Date_Year'] in (2015,2016)) :#and (5<=df.loc[i,'Date_Month']<=8): # modern strong El nino
      marker = folium.CircleMarker([lat, lon],radius=0.2*df.loc[i,'Percent_Bleaching']+1,
                                color='crimson', fill_color='crimson')
      modernSEN.add_child(marker)
      # marker.add_to(m)
    elif (df.loc[i,'Date_Year'] in (2007,2008,2010,2011)) :#and (5<=df.loc[i,'Date_Month']<=8): # modern strong La nina
      marker = folium.CircleMarker([lat, lon],radius=0.2*df.loc[i,'Percent_Bleaching']+1,
                                color='DarkBlue', fill_color='DarkBlue')
      modernSLN.add_child(marker)
      # marker.add_to(m)
    elif (df.loc[i,'Date_Year'] in (2012,2013,2014,2019,2020)) :#and (5<=df.loc[i,'Date_Month']<=8): # modern moderate
      marker = folium.CircleMarker([lat, lon],radius=0.2*df.loc[i,'Percent_Bleaching']+1,
                                color='green', fill_color='green')
      modernmod.add_child(marker)
    #  marker.add_to(m)
#########################################################################################################################################
    elif (df.loc[i,'Date_Year'] in (1997,1998,2002,2003)) :#and (5<=df.loc[i,'Date_Month']<=8): # old strong El nino
      marker = folium.CircleMarker([lat, lon],radius=0.2*df.loc[i,'Percent_Bleaching']+1,
                                color='LightCoral', fill_color='LightCoral')
      oldSEN.add_child(marker)
    #  marker.add_to(m)
    elif (df.loc[i,'Date_Year'] in (1999,2000)) :#and (5<=df.loc[i,'Date_Month']<=8): # old strong La nina
      marker = folium.CircleMarker([lat, lon],radius=0.2*df.loc[i,'Percent_Bleaching']+1,
                                color='CornflowerBlue', fill_color='CornflowerBlue')
      oldSLN.add_child(marker)
    #  marker.add_to(m)
    elif (df.loc[i,'Date_Year'] in (1996,2001,2004)) :#and (5<=df.loc[i,'Date_Month']<=8): # old moderate
      marker = folium.CircleMarker([lat, lon],radius=0.2*df.loc[i,'Percent_Bleaching']+1,
                                color='PaleGreen', fill_color='PaleGreen')
      oldmod.add_child(marker)
    #  marker.add_to(m)

folium.LayerControl().add_to(m);

m.get_root().add_child(macro)

m
```

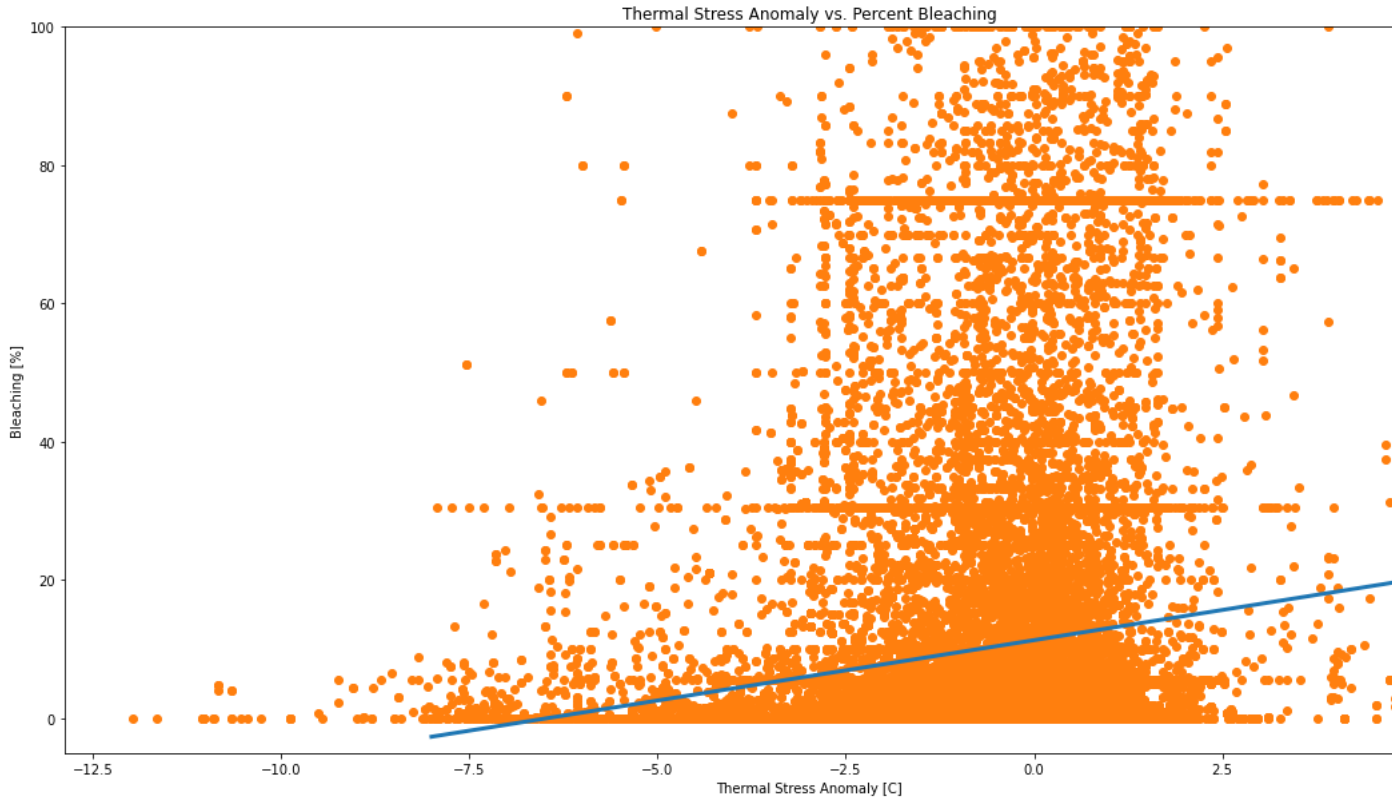



```
### CLusters of Data Points

m = folium.Map(location=[39.4699, 0.3763],zoom_start=2.5)
marker_cluster = MarkerCluster(
    name='Coral Bleaching Data',
    overlay=True,
    control=False,
    icon_create_function=None)

for lat,lon,i in zip(df["Latitude_Degrees"], df["Longitude_Degrees"],range(len(df))):

    marker = folium.CircleMarker([lat, lon],radius=3,
                                 color='crimson',
                                 fill_color='crimson')
    marker_cluster.add_child(marker)

marker_cluster.add_to(m)
m
```




```
### Heatmap of Data Weighted by Percent Bleaching

heatmap = df[['Latitude_Degrees','Longitude_Degrees','Percent_Bleaching']].dropna()
m = folium.Map(location=[39.4699, 0.3763],zoom_start=2.5)

HeatMap(heatmap,
        min_opacity=.35,
        blur = 9
               ).add_to(folium.FeatureGroup(name='Heat Map').add_to(m))
folium.LayerControl().add_to(m)
m
```

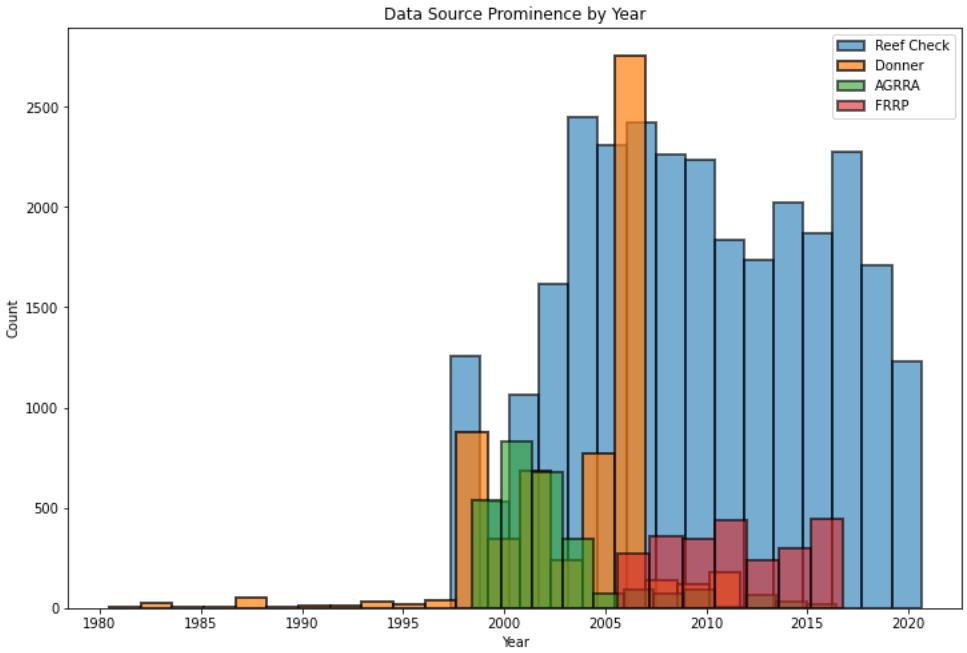

```
x=np.linspace(-8,6,100)
dfsub=df.dropna(subset=['TSA','Percent_Bleaching'])
a = np.poly1d(np.polyfit(dfsub['TSA'],dfsub['Percent_Bleaching'],deg=1))

plt.figure(figsize=(20,10))
plt.plot(df['TSA'],df['Percent_Bleaching'],'o',color='tab:orange')
plt.plot(x,a(x),'tab:blue',linewidth=3,label='Linear Fit')
plt.ylim(bottom=-5,top=100)
plt.title('Thermal Stress Anomaly vs. Percent Bleaching')
plt.xlabel('Thermal Stress Anomaly [C]')
plt.ylabel('Bleaching [%]')
plt.legend()
plt.show()
```




```
print(df.Data_Source.value_counts())

reefcheck = df[df['Data_Source']=='Reef_Check']
donner = df[df['Data_Source']=='Donner']
agrra = df[df['Data_Source']=='AGRRA']
frrp = df[df['Data_Source']=='FRRP']

plt.figure(figsize=(12,8))
plt.hist(reefcheck.Date,label='Reef Check',bins=16,alpha=0.6,edgecolor='k',linewidth=2)
plt.hist(donner.Date,label='Donner',bins=20,alpha=0.7,edgecolor='k',linewidth=2)
plt.hist(agrra.Date,label='AGRRA',bins=12,alpha=0.6,edgecolor='k',linewidth=2)
plt.hist(frrp.Date,label='FRRP',bins=7,alpha=0.6,edgecolor='k',linewidth=2)
plt.title('Data Source Prominence by Year')
plt.xlabel('Year')
```

```
plt.ylabel('Count')
plt.legend()
plt.show()
```

```
Reef_Check    28821
Donner         6307
AGRRA          2848
FRRP           2394
Kumagai         667
McClanahan      234
Safaie           81
Nuryana           5
Setiawan          4
Name: Data_Source, dtype: int64
```



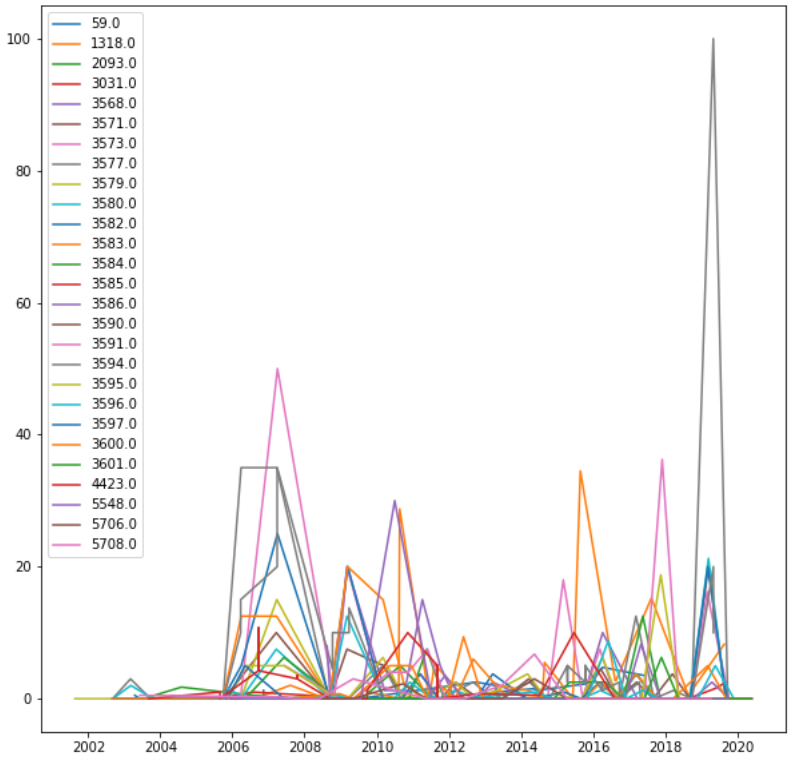

```
print(df['Site_ID'].value_counts())
test = df[df['Site_ID']==3886]
print(test['Depth_m'].value_counts())
```

```
3594.0    90
3886.0    88
3031.0    70
8238.0    68
3015.0    68
          ..
7219.0     1
155.0      1
7220.0     1
7221.0     1
658.0      1
Name: Site_ID, Length: 12702, dtype: int64
8.0    44
4.0    40
5.0     4
Name: Depth_m, dtype: int64
```

```
after1997 = df[df['Date_Year']>1996]
sites = after1997.Site_ID.unique()
years = after1997.Date_Year.unique()

goods=[]
for i in range(len(sites)):

  temp = after1997[after1997['Site_ID']==sites[i]]
  siteyears = temp.Date_Year.unique()
  num = np.isin(years,siteyears)
  trues = num.sum()
  total = len(years)
  prop = trues/total
  if prop>0.5:
    goods.append(sites[i])

goods
```

```
[59.0,
 1318.0,
 2093.0,
 3031.0,
 3568.0,
 3571.0,
 3573.0,
 3577.0,
 3579.0,
 3580.0,
 3582.0,
 3583.0,
 3584.0,
 3585.0,
 3586.0,
 3590.0,
 3591.0,
 3594.0,
 3595.0,
 3596.0,
 3597.0,
 3600.0,
 3601.0,
 4423.0,
 5548.0,
 5706.0,
 5708.0]
```

```
selected = df[np.isin(df['Site_ID'],goods)].sort_values('Date')
plt.figure(figsize=(10,10))
for i in range(len(goods)):
  plt.plot(selected[selected['Site_ID']==goods[i]].Date,selected[selected['Site_ID']==goods[i]].Percent_Bleaching,label=goods[i])

plt.legend()
plt.show()

plt.figure(figsize=(10,10))
plt.plot(selected.Date,selected.Percent_Bleaching)
plt.show()
```



ntratos aquí

✓ 0 s se ejecutó 13:21