

Program :

```
import eel
import matplotlib.pyplot as plt
from io import BytesIO
import base64
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns

df = pd.read_csv('../Country Quater Wise Visitors Imputed.csv')

names = [i for i, j in df.items()][1: ]
values = [df[i].sum() for i in names]
names_list = [name[0: 8] for name in names]
index = 2
countries = df['Country of Nationality']
year = names[16]

def bar_graph_1():
    plt.figure(figsize=(15, 7))
    plt.xticks(rotation=45)
    plt.bar(names_list, values, width=0.8)
    plt.title("Tourists over the year 2014 - 2020")
    plt.xlabel("Years in Quarter")
    plt.ylabel("Population in ten thousands")
    return plt

def bar_graph_2():
    country_values = list(df.T[index])[1: ]
    plt.figure(figsize=(15, 7))
    plt.xticks(rotation=45)
    plt.bar(names_list, country_values, width=0.8, color='coral')
    plt.title("Tourists over the year 2014 - 2020 for " + countries[index])
    plt.xlabel("Years in Quarter")
    plt.ylabel("Population in ten thousands")
    return plt

def bar_graph_3():
    country_values = []
    for i, j in df.T.items():
        country_values.append(round(df.T[i][1: ].median(), 2))
    plt.figure(figsize=(20, 7))
    plt.bar(countries, country_values, color='slateblue')
    plt.xticks(rotation=90)
```

```

plt.title("Average Tourists over the year 2014 - 2020 for 63 countries")
plt.xlabel("Countries")
plt.ylabel("Population in ten thousands")
return plt

```

```

def scatter_plot():
    plt.figure(figsize=(25, 10))
    for column_name, _ in df.items():
        if column_name != 'Country of Nationality':
            plt.scatter(countries, df[column_name])
    plt.legend(names_list)
    plt.title("Tourists over the year 2014 - 2020 for 63 countries")
    plt.xlabel("Countries")
    plt.ylabel("Population in ten thousands")
    plt.xticks(rotation=90)
    return plt

```

```

def histogram():
    plt.figure(figsize=(12, 7))
    plt.hist(df[year], color='crimson', bins=15)
    plt.title('Histogram Distribution for ' + year)
    plt.xlabel("Countries")
    plt.ylabel("Population in ten thousands")
    return plt

```

```

def stack_plot():
    plt.figure(figsize=(15, 5))
    plt.stackplot(countries, df[year], color='cornflowerblue')
    plt.xticks(rotation=90)
    plt.title('Stack PLOT Distribution for ' + year)
    plt.xlabel("Countries")
    plt.ylabel("Population in ten thousands")
    return plt

```

```

def box_plot():
    data = []
    for name in names:
        data.append(list(df[name]))
    plt.figure(figsize=(20, 10))
    plt.boxplot(data)
    plt.title('Box Plot for years 2014 - 2020')
    plt.xlabel("Years from 2014 to 2020")
    plt.ylabel("Population in ten thousands")
    return plt

```

```

def violin_plot():

```

```

data = []
for name in names:
    data.append(list(df[name]))
plt.figure(figsize=(20, 10))
plt.violinplot(data, showmeans=True, showmedians=True)
plt.title("Violin Plot for years 2014 - 2020")
plt.xlabel("Years from 2014 to 2020")
plt.ylabel("Population in ten thousands")
return plt

```

```

def pie_chart_1():
    country_values = []
    for i, j in df.T.items():
        country_values.append(round(df.T[i][1: ].median(), 2))
    plt.figure(figsize=(8, 8))
    plt.pie(country_values, labels=countries)
    plt.title('Pie Chart for 63 countries over the year 2014 - 2020')
    return plt

```

```

def pie_chart_2():
    newData = {
        'years': [],
        'population': []
    }
    c = 0
    i = 0
    while i < len(values):
        newData['years'].append(2014+c)
        newData['population'].append(values[i] + values[i+1] + values[i+2] + values[i+3])
        c = c+1
        i += 4
    plt.figure(figsize=(10, 10))
    plt.pie(newData['population'], labels=newData['years'], radius=1)
    plt.title("Pie Chart for years 2014 - 2020")
    plt.pie([5], radius=0.5, colors='white')
    return plt

```

```

def heat_map():
    arrayData = np.loadtxt('../Country Quater Wise Visitors Imputed.csv', skiprows=1, delimiter=',',
usecols=(range(1, 29)))
    plt.figure(figsize = (14,5))
    sns.heatmap( np.transpose(arrayData) , linewidth = 0 , cmap = 'spring' )
    # ax.set_yticks(countries)
    plt.title( "2-D Heat Map of the Dataset" )
    plt.xlabel("63 Countries")
    plt.ylabel("Years from 2014 to 2020")

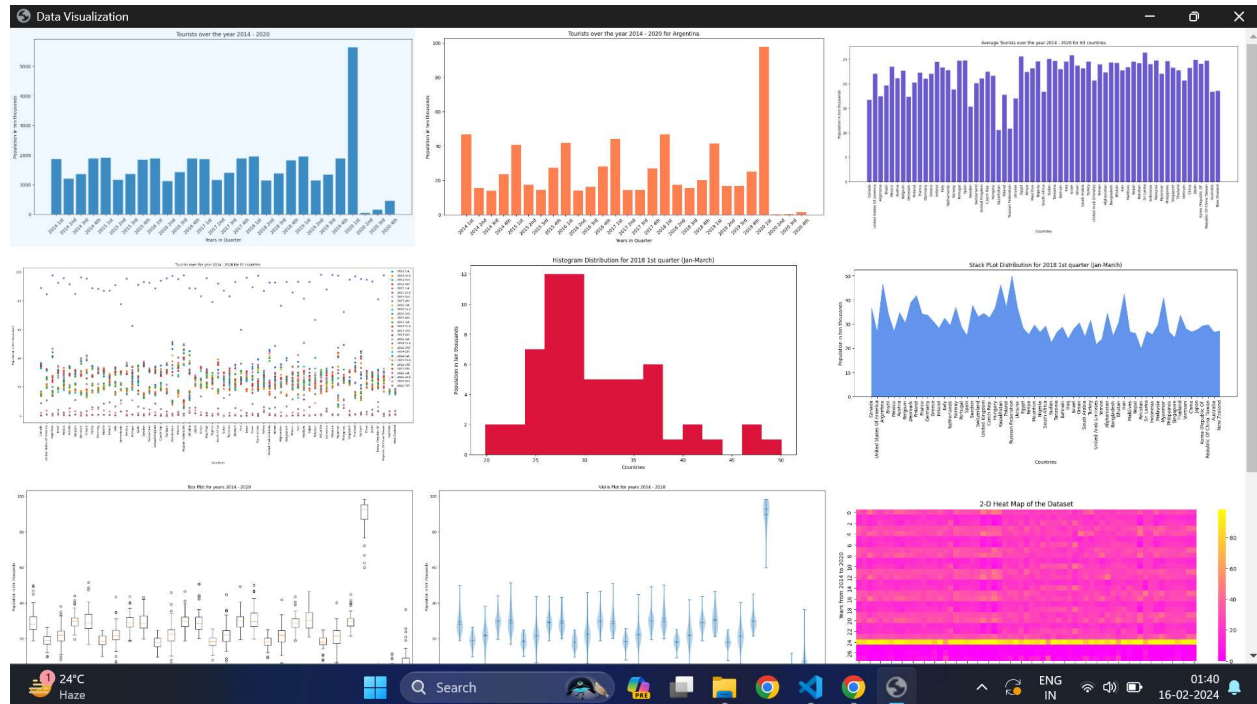
```

```
return plt
```

```
graph_function_list = [  
    ('Bar Graph 1', bar_graph_1),  
    ('Bar Graph 2', bar_graph_2),  
    ('Bar Graph 3', bar_graph_3),  
    ('Scatter Plot', scatter_plot),  
    ('Histogram', histogram),  
    ('Stack Plot', stack_plot),  
    ('Box Plot', box_plot),  
    ('Violin PLOT', violin_plot),  
    ('Heat Map', heat_map),  
    ('Pie Chart 1', pie_chart_1),  
    ('Pie Chart 2', pie_chart_2),  
]
```

```
eel.init('web')  
@eel.expose  
def generate_plot(graph_number):  
    name, run = graph_function_list[graph_number]  
    g = run()  
  
    buffer = BytesIO()  
    g.savefig(buffer, format='png', bbox_inches='tight')  
    buffer.seek(0)  
  
    img_str = base64.b64encode(buffer.read()).decode('utf-8')  
    buffer.close()  
    g.clf()  
  
    return {  
        'name': name,  
        'url': f'data:image/png;base64,{img_str}'  
    }  
  
eel.start('index.html')
```

Output:



1. Bar Graph 1

