GUI APPLICATION

In [2]:

```
#start of predefine code
import pandas as pd
from ipywidgets import Button, Dropdown, Output, VBox, Layout, widgets
from IPython.display import display
from tkinter import Tk, filedialog
import matplotlib.pyplot as plt
import numpy as np

graph_type = ['Choose one.. ','bubble','bar']
funtionality = ['Choose One','Sort','Filter']
sort_option = ['ascending','descending']
df = ''
new_df = ''
input_box = ''
input_fontsize = ''
input_title = ''
#end of predefine code

def select_files(b):
    # Create a Tkinter window and hide it
    root = Tk()
    root.withdraw()

    # Get path of CSV file using file dialog
    file_path = filedialog.askopenfilename(filetypes=[("CSV Files", "*.csv")])

    # Read data from CSV into a Pandas DataFrame
    data = pd.read_csv(file_path)

    # Create a drop-down menu for selecting a column to operate on
    column_dropdown = widgets.Dropdown(options=data.columns.tolist(), description="Selec

    # Create buttons for sorting and filtering
    sort_button = widgets.Button(description="Sort")
    filter_button = widgets.Button(description="Filter")

    # Create input box for filter value
    filter_value_input = widgets.Text(description="Filter Value:")

    # Create output widget for displaying sorted or filtered data
    output = widgets.Output()

    # Define sorting and filtering functions
    def sort_data(column):
        # Sort data by selected column
        sorted_data = data.sort_values(by=column)
        with output:
            display(sorted_data)

    def filter_data(column, filter_value):
        # Filter data by selected column and value
        filtered_data = data[data[column] == filter_value]
        with output:
            display(filtered_data)

    # Connect buttons to sorting and filtering functions
    sort_button.on_click(lambda b: sort_data(column_dropdown.value))
```

```
    filter_button.on_click(lambda b: filter_data(column_dropdown.value, filter_value_inp

    # Display widgets
    display(column_dropdown)
    display(filter_value_input)
    display(filter_button)
    display(sort_button)
    display(output)

#start of predefine code
def display_plot(xaxis, yaxis, graph_type):
    global new_df
    global input_title
    global input_fontsize
    if(graph_type == 'bubble'):
        plt.subplots(figsize=(19,8))
        rgb = np.random.rand(3)
        #Write Condition here

        #End of write condition here
        plt.title(input_title.value, fontsize=input_fontsize.value)
        plt.xlabel(xaxis, fontsize=input_fontsize.value)
        plt.xticks(rotation='vertical')
        plt.ylabel(yaxis, fontsize=input_fontsize.value)
        plt.show()
    elif(graph_type == 'bar'):
        plt.subplots(figsize=(19,8))
        plt.bar(new_df[xaxis], new_df[yaxis], color=['red', 'green','blue','yellow','pir
        plt.title(input_title.value, fontsize=input_fontsize.value)
        plt.xlabel(xaxis, fontsize=input_fontsize.value)
        plt.xticks(rotation='vertical')
        plt.ylabel(yaxis, fontsize=input_fontsize.value)
        plt.show()
    else:
        print("Choose valid graph")
fileselect = widgets.Button(description="File select")
fileselect.on_click(select_files)
display(fileselect)
#end of predefined
```

File select

Select Colu… Embarked

Filter Value: 55

Filter

Sort

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin

	PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512
125	126	1	3	Nicola-Yarred, Master. Elias	male	12.0	1	0	2651	11
354	355	0	3	Yousif, Mr. Wazli	male	NaN	0	0	2647	7
352	353	0	3	Elias, Mr. Tannous	male	15.0	1	1	2695	7
128	129	1	3	Peter, Miss. Anna	female	NaN	1	1	2668	22
...	...	...	...	...	...	...	...	...	...	
355	356	0	3	Vanden Steen, Mr. Leo Peter	male	28.0	0	0	345783	9
344	345	0	2	Fox, Mr. Stanley Hubert	male	36.0	0	0	229236	13
445	446	1	1	Dodge, Master. Washington	male	4.0	0	2	33638	81
61	62	1	1	Icard, Miss. Amelie	female	38.0	0	0	113572	80
829	830	1	1	Stone, Mrs. George Nelson (Martha Evelyn)	female	62.0	0	0	113572	80

In [ ]:

891 rows × 12 columns