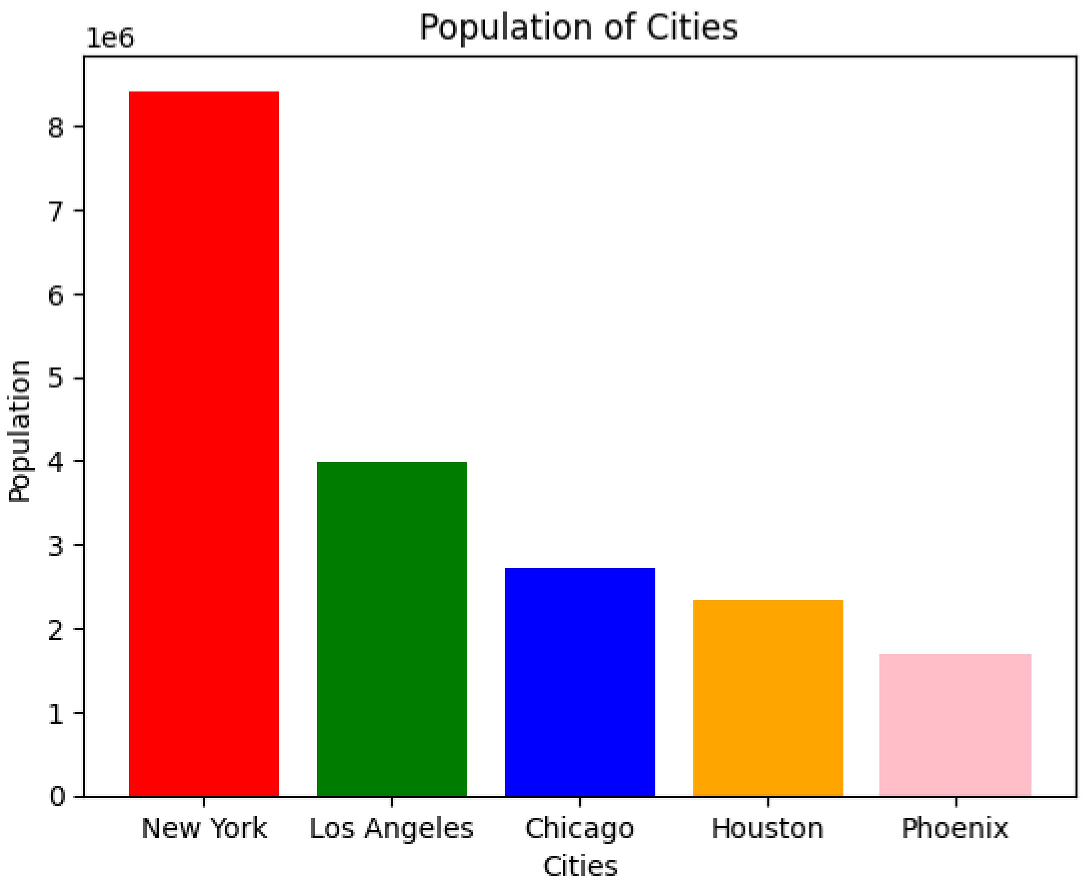
1. Write a Python program to create a bar graph that displays the population of five different cities. Customize the bar graph by adding a title, labels for the axes, and different colors for each bar.

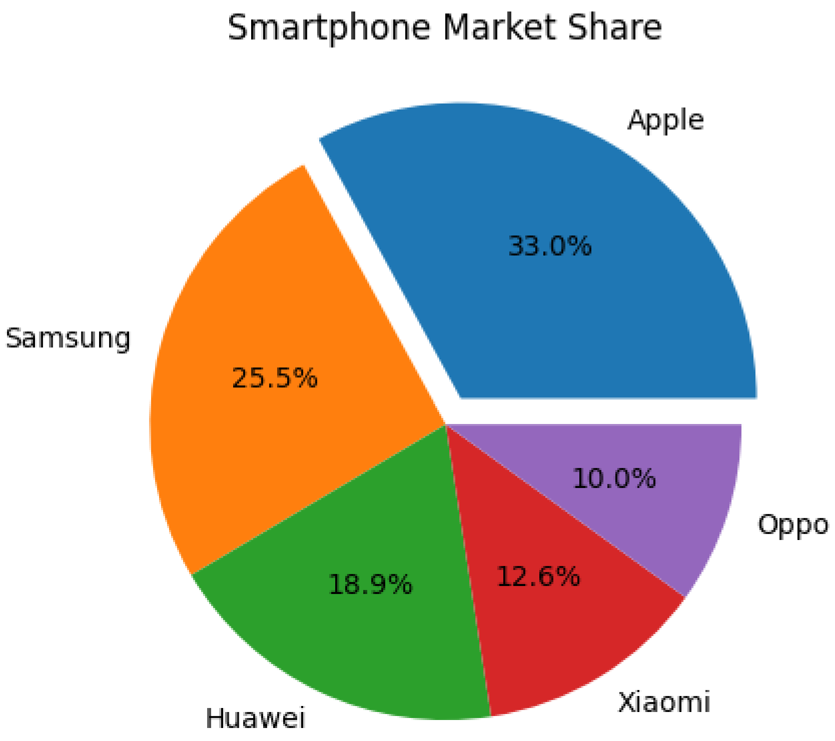
In [2]:
```python
import matplotlib.pyplot as plt
# Data
cities = ['New York', 'Los Angeles', 'Chicago', 'Houston', 'Phoenix']
populations = [8419000, 3980000, 2716000, 2328000, 1690000]
colors = ['red','green', 'blue', 'orange', 'pink']

#Code here::
plt.bar(cities, populations, color=colors)
plt.title('Population of Cities')
plt.xlabel('Cities')
plt.ylabel('Population')
plt.show()
```



2. Write a Python program to create a pie chart that represents the market share of different smartphone brands. Label each wedge with the

brand name and percentage, and highlight (explode) the wedge representing higher market share.

In [6]:
```
# Data
brands = ['Apple', 'Samsung', 'Huawei', 'Xiaomi', 'Oppo']
market_share = [27.5, 21.3, 15.8, 10.5, 8.3]

#Code here::
import matplotlib.pyplot as plt

explode = [0.1 if share == max(market_share) else 0 for share in market_share]

plt.pie(market_share, labels=brands, autopct='%1.1f%%', explode=explode)
plt.title("Smartphone Market Share")
plt.show()
```

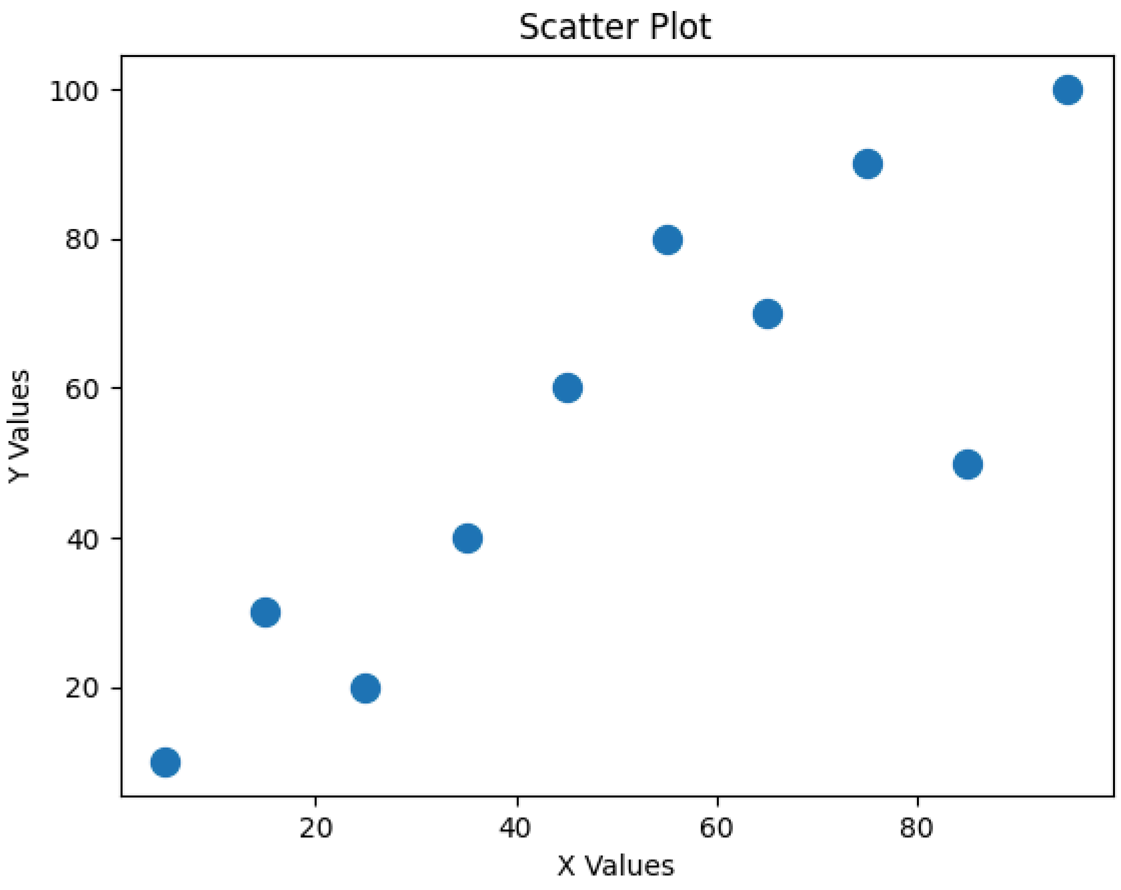


---

3. Create a Python program that generates a scatter plot using the provided data. Customize the plot by: Setting the size of each point to 100, adding a title "Scatter Plot", and labeling the x-axis as "X Values" and the y-axis as "Y Values".

In [7]:
```
# Data
x = [5, 15, 25, 35, 45, 55, 65, 75, 85, 95]
```

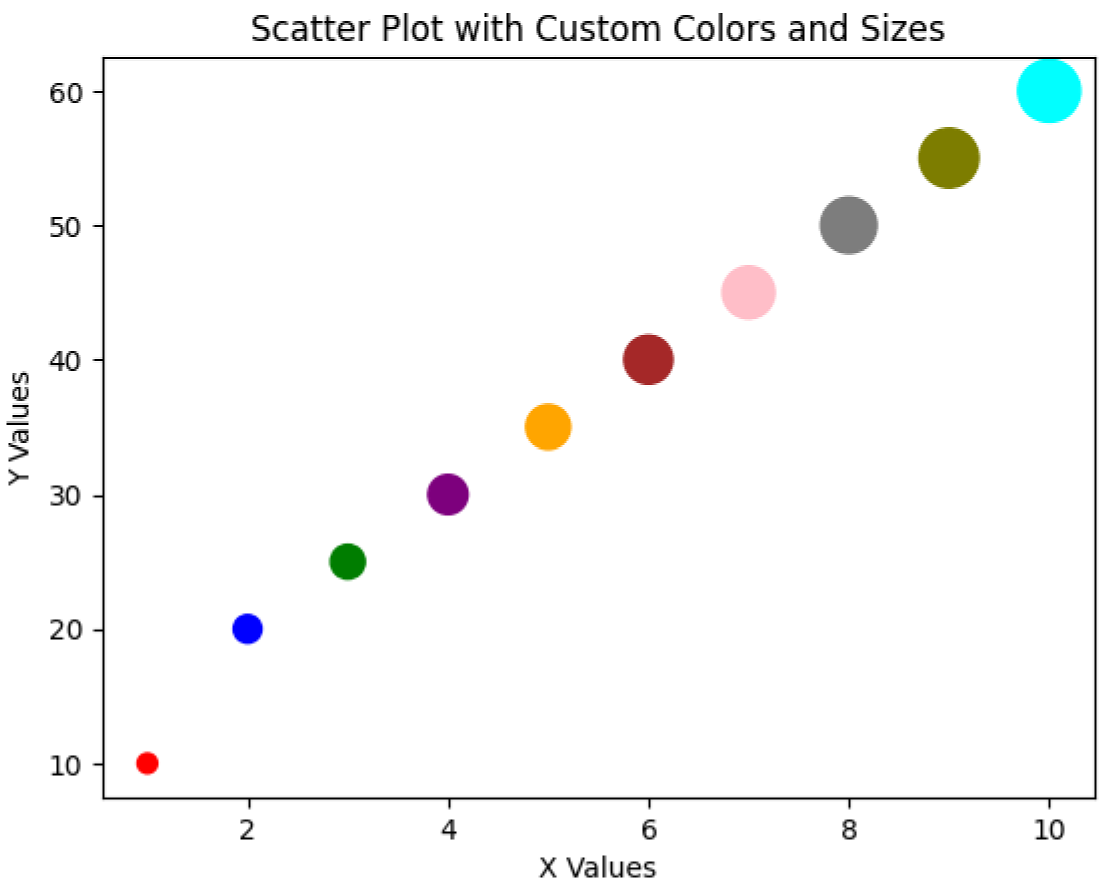
```
y = [10, 30, 20, 40, 60, 80, 70, 90, 50, 100]

#Code here::
plt.scatter(x, y, s=100)
plt.title("Scatter Plot")
plt.xlabel("X Values")
plt.ylabel("Y Values")
plt.show()
```



---

4. Write a Python program to create a scatter plot with custom markers Customize the markers to have different colors and sizes.

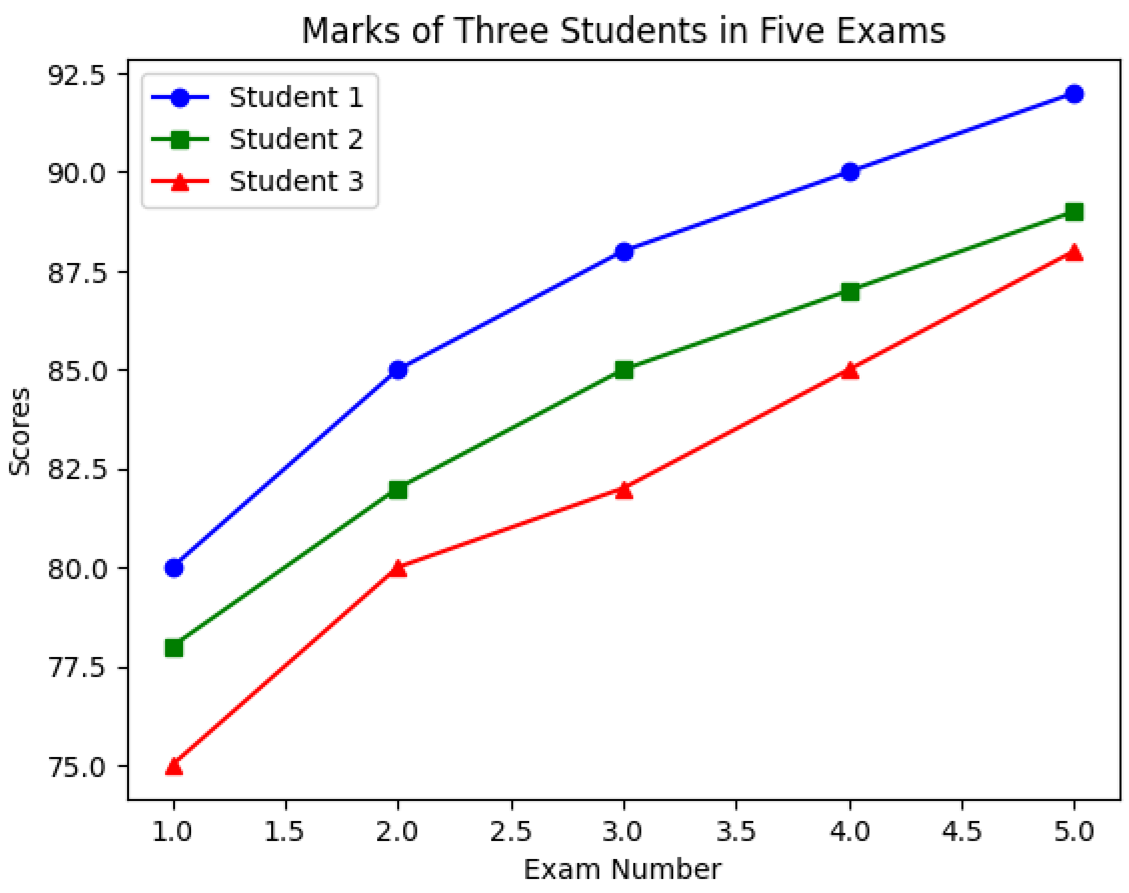
In [8]:
```
# Data
x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y = [10, 20, 25, 30, 35, 40, 45, 50, 55, 60]
colors = ['red', 'blue', 'green', 'purple', 'orange', 'brown', 'pink', 'gray',
sizes = [50, 100, 150, 200, 250, 300, 350, 400, 450, 500]

#Code here::
plt.scatter(x, y, c=colors, s=sizes)
plt.xlabel("X Values")
plt.ylabel("Y Values")
plt.title("Scatter Plot with Custom Colors and Sizes")
plt.show()
```



---

5. Write a Python program to create a line plot that shows the performance of three different students over five exams.Plot all three lines on the same graph, using different colors and markers for each line. Add a legend to identify each student.

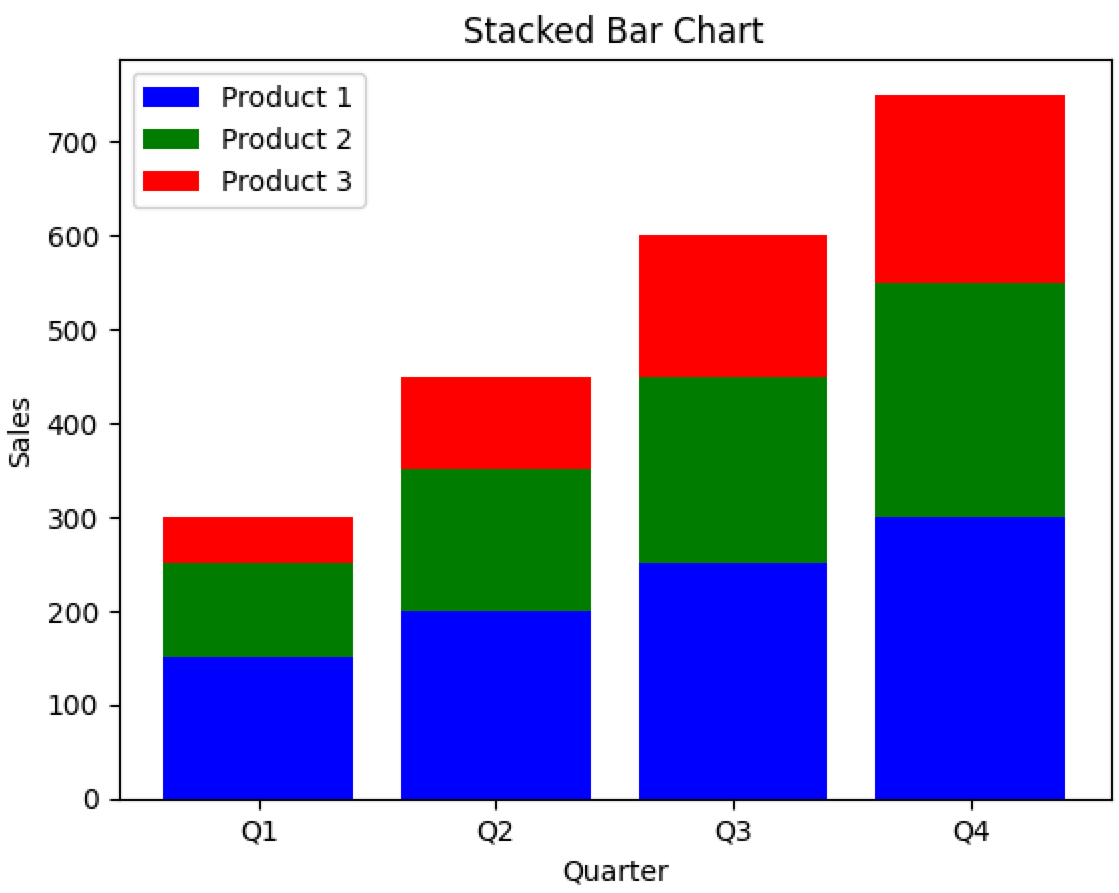
In [10]:
```
# Data
exams = [1, 2, 3, 4, 5]
student1_scores = [80, 85, 88, 90, 92]
student2_scores = [78, 82, 85, 87, 89]
student3_scores = [75, 80, 82, 85, 88]

#Code here::
plt.plot(exams, student1_scores, marker='o', color='blue', label='Student 1')
plt.plot(exams, student2_scores, marker='s', color='green', label='Student 2')
plt.plot(exams, student3_scores, marker='^', color='red', label='Student 3')
plt.xlabel("Exam Number")
plt.ylabel("Scores")
plt.title("Marks of Three Students in Five Exams")
plt.legend()
plt.show()
```



---

6. Write a Python program to create a stacked bar chart that represents the sales of three different products over four quarters. use different colors for each product, and add a legend.

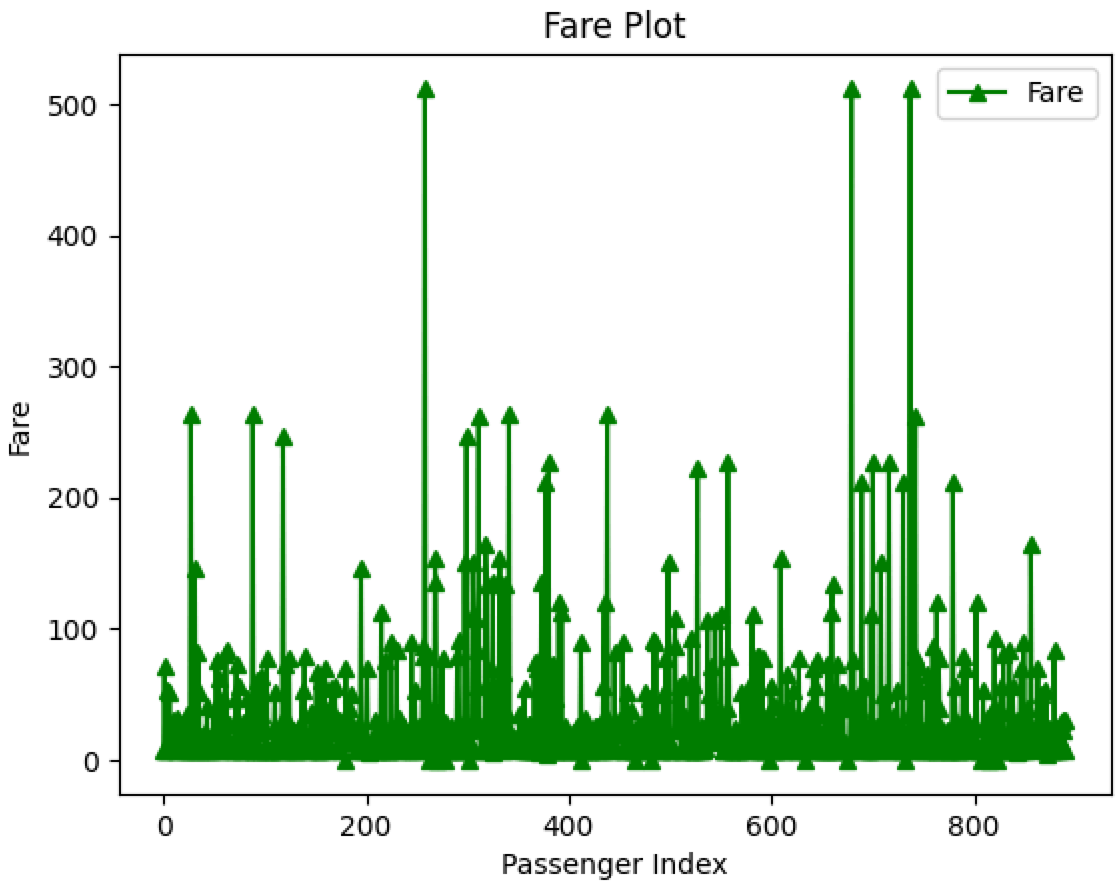
In [11]:
```
# Data
quarters = ['Q1', 'Q2', 'Q3', 'Q4']
product1_sales = [150, 200, 250, 300]
product2_sales = [100, 150, 200, 250]
product3_sales = [50, 100, 150, 200]

#Code here::
plt.bar(quarters, product1_sales, color='blue', label='Product 1')
plt.bar(quarters, product2_sales, bottom=product1_sales, color='green', label=
plt.bar(quarters, product3_sales, bottom=[i+j for i,j in zip(product1_sales, p
plt.xlabel('Quarter')
plt.ylabel('Sales')
plt.title('Stacked Bar Chart')
plt.legend()
plt.show()
```



---

7. Load the titanic.csv dataset using pandas and plot the 'fare' column with a line plot. Customize the plot to have a green line with triangle markers, a title "Fare Plot", and labels for the x and y axes.?

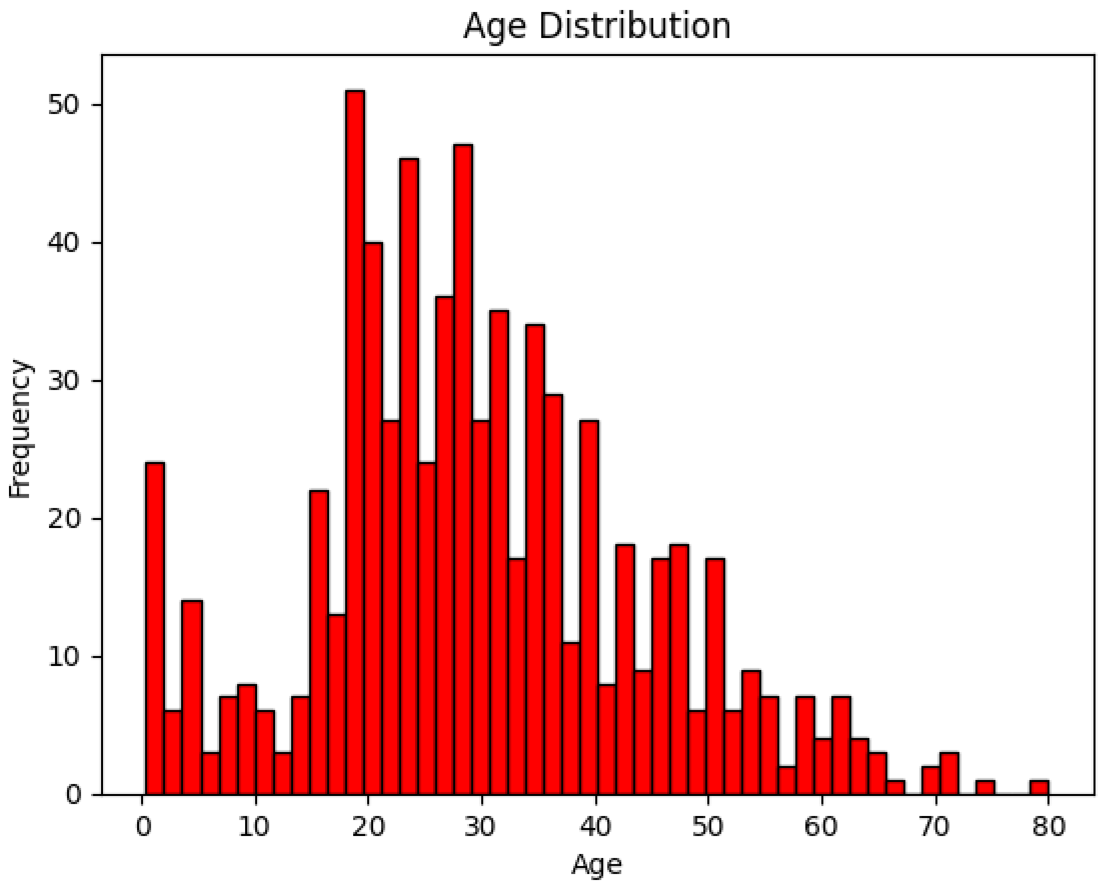
In [12]:
```
#Code here::
import pandas as pd
file_path = r"C:\Users\haris\Downloads\titanic.csv"
data = pd.read_csv(file_path)
fare = data['fare']
plt.plot(fare, color='green', marker='^', label='Fare')

plt.title("Fare Plot")
plt.xlabel("Passenger Index")
plt.ylabel("Fare")
plt.legend()
plt.show()
```



---

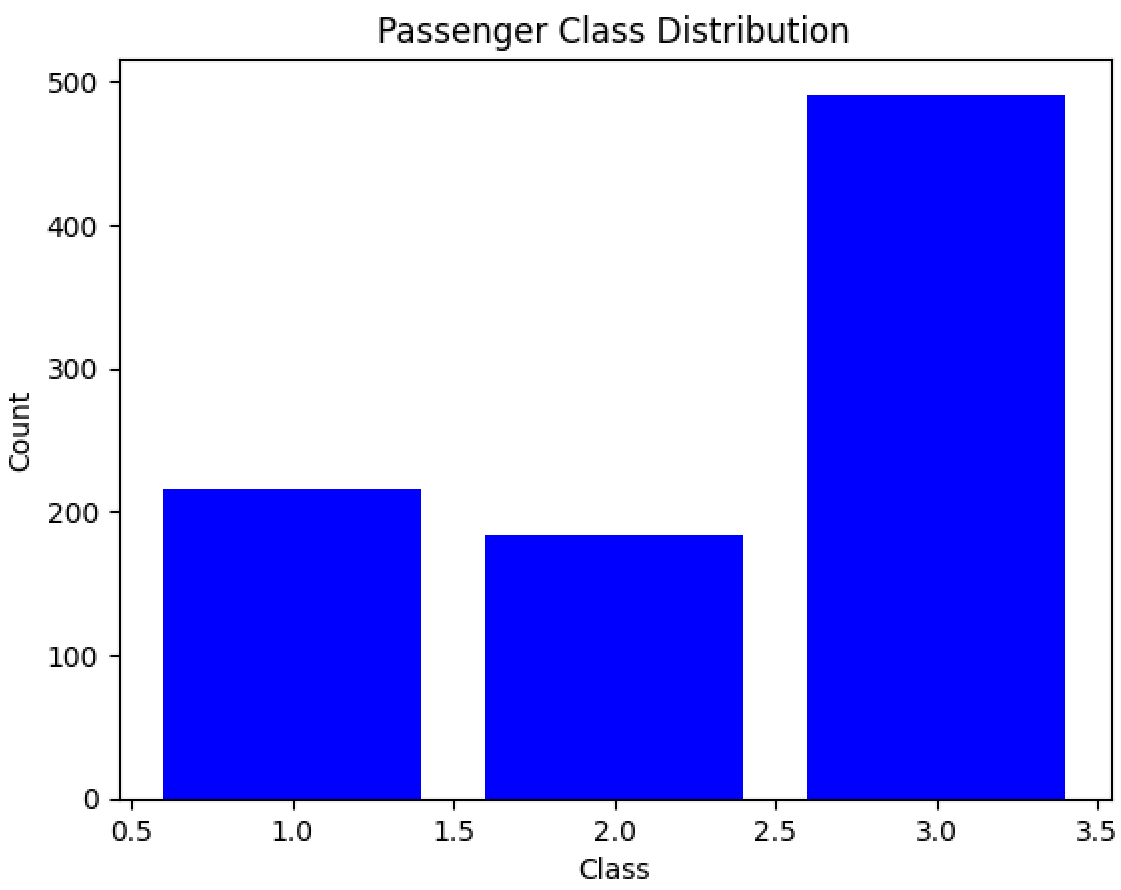
8. Plot a histogram of the 'age' column from the Titanic dataset. Customize the histogram to have 50 bins, a title "Age Distribution", and a red color for the bars.

In [15]:
```
#Code here:
age = data['age'].dropna()
plt.hist(age, bins=50, color='red', edgecolor='black')
plt.title("Age Distribution")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.show()
```



9.Create a bar chart of the 'pclass' column in Titanic dataset, showing the count of each class. Set the color of the bars to blue.

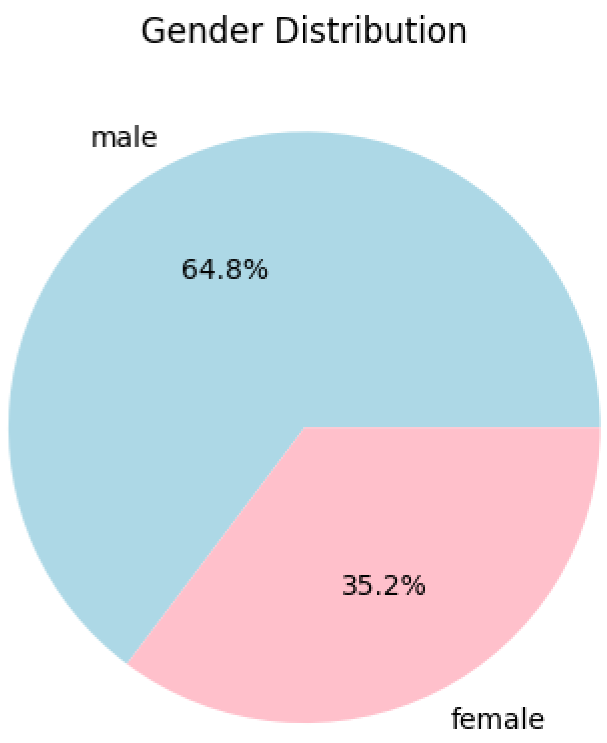
In [16]:
```
#Code here:
import pandas as pd
import matplotlib.pyplot as plt
pclass_counts = data['pclass'].value_counts().sort_index()
plt.bar(pclass_counts.index, pclass_counts.values, color='blue')
plt.title("Passenger Class Distribution")
plt.xlabel("Class")
plt.ylabel("Count")
plt.show()
```



---

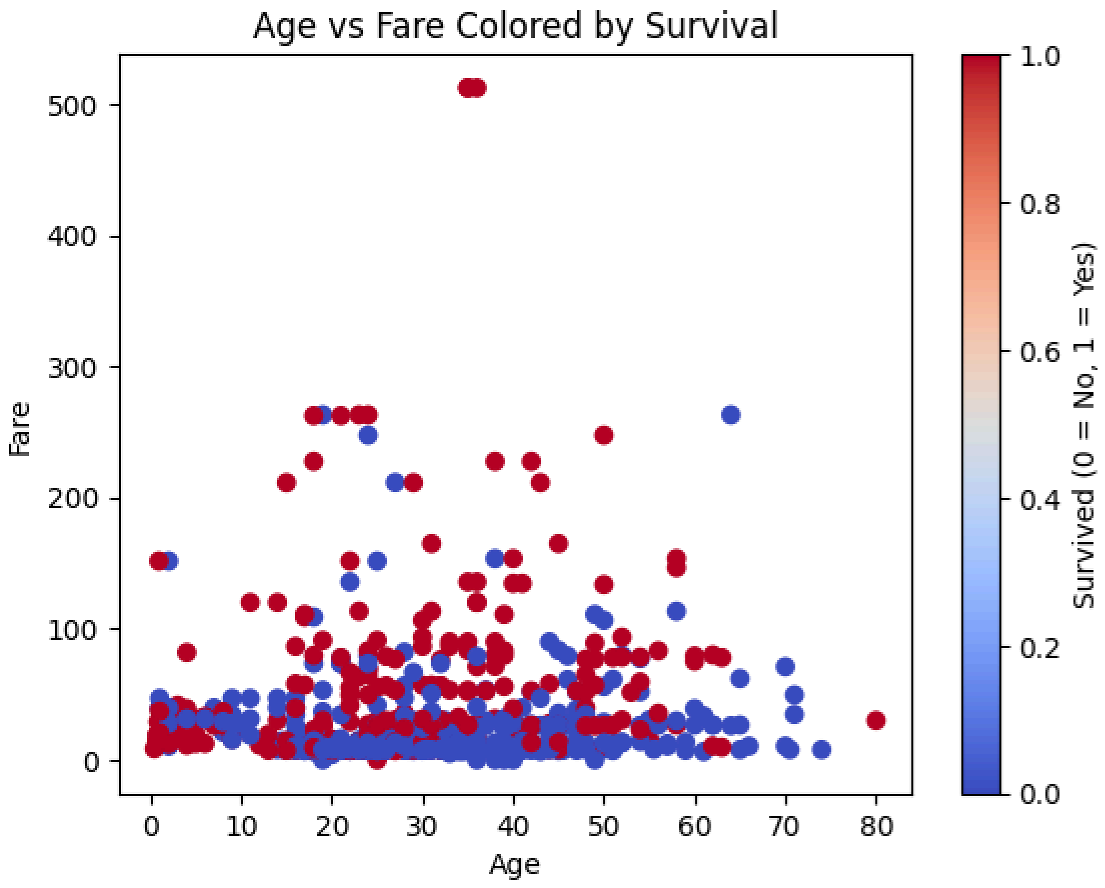
10. Create a pie chart for the 'sex' column in the Titanic dataset. Set the colors to ['lightblue', 'pink'] and add a title "Gender Distribution".

In [17]:
```
#Code here:
gender_counts = data['sex'].value_counts()
plt.pie(gender_counts, labels=gender_counts.index, colors=['lightblue', 'pink'
plt.title("Gender Distribution")
plt.show()
```



11. Using the Titanic dataset, plot a scatter plot of 'age' vs 'fare' and color the points by 'survived'. Set the colormap to 'coolwarm'.

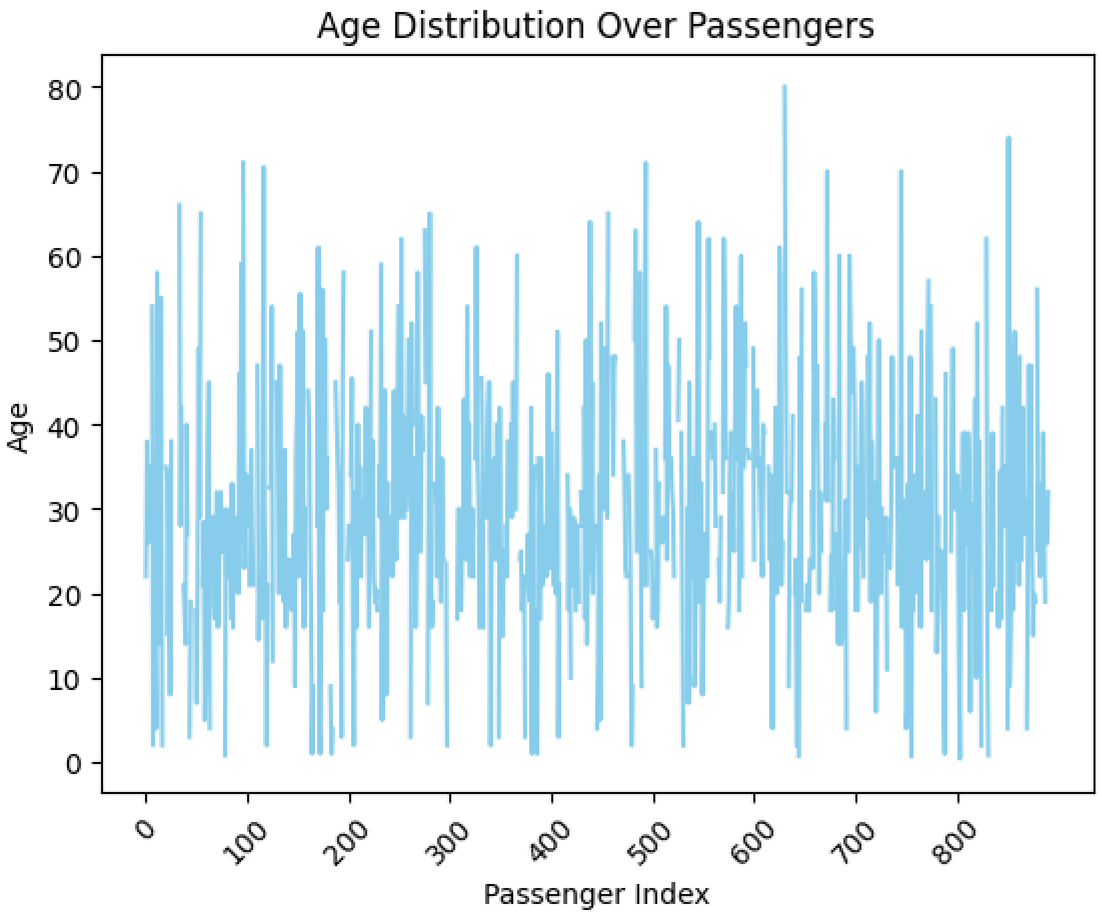
In [18]:
```
#Code here:
filtered_data = data.dropna(subset=['age', 'fare'])
plt.scatter(filtered_data['age'], filtered_data['fare'], c=filtered_data['surv
plt.title("Age vs Fare Colored by Survival")
plt.xlabel("Age")
plt.ylabel("Fare")
plt.colorbar(label='Survived (0 = No, 1 = Yes)')
plt.show()
```



---

12. Using the Titanic dataset, create a line plot of the 'age' column and customize the x-axis and y-axis ticks to show every 100 passengers and every 10 years of age, respectively.

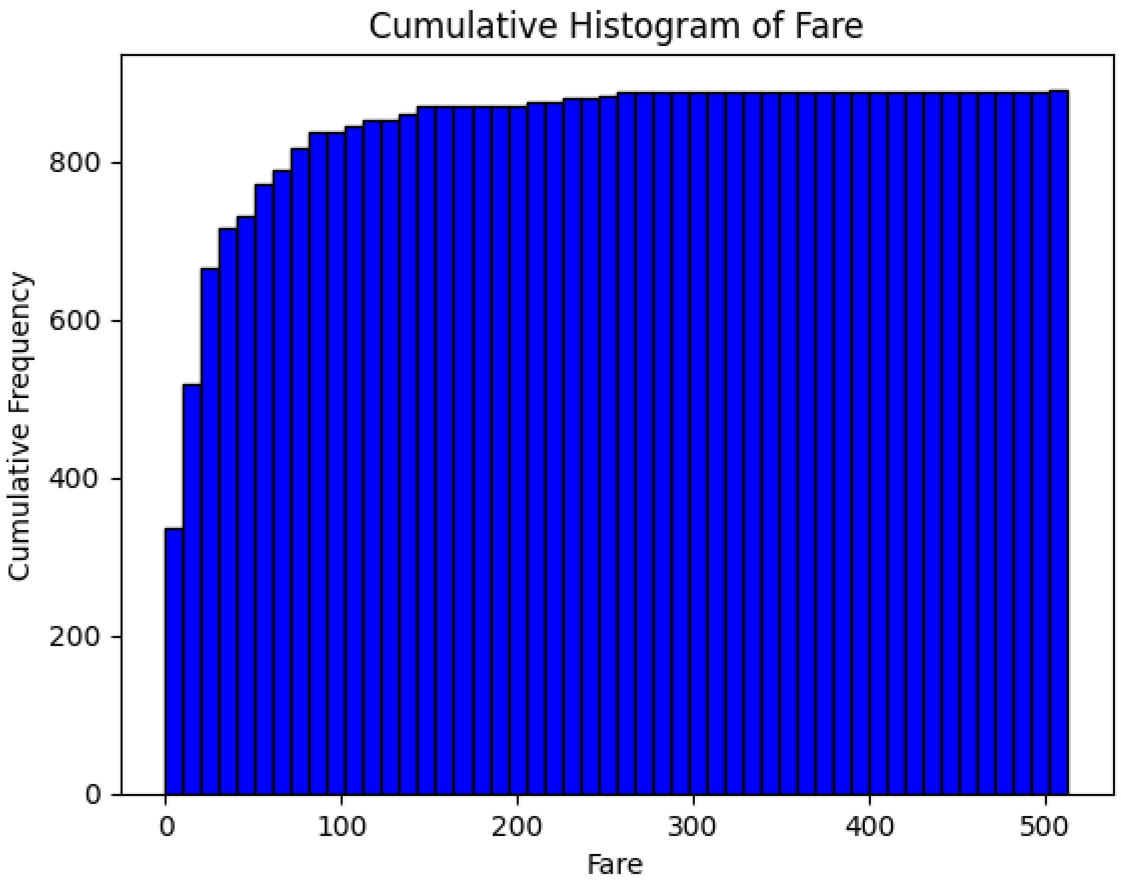
In [20]:
```
#Code here:
plt.plot(data['age'], color='skyblue')
plt.xticks(range(0, len(data), 100), rotation=45)
plt.yticks(range(0, int(data['age'].max()) + 1, 10))
plt.title("Age Distribution Over Passengers")
plt.xlabel("Passenger Index")
plt.ylabel("Age")
plt.show()
```



---

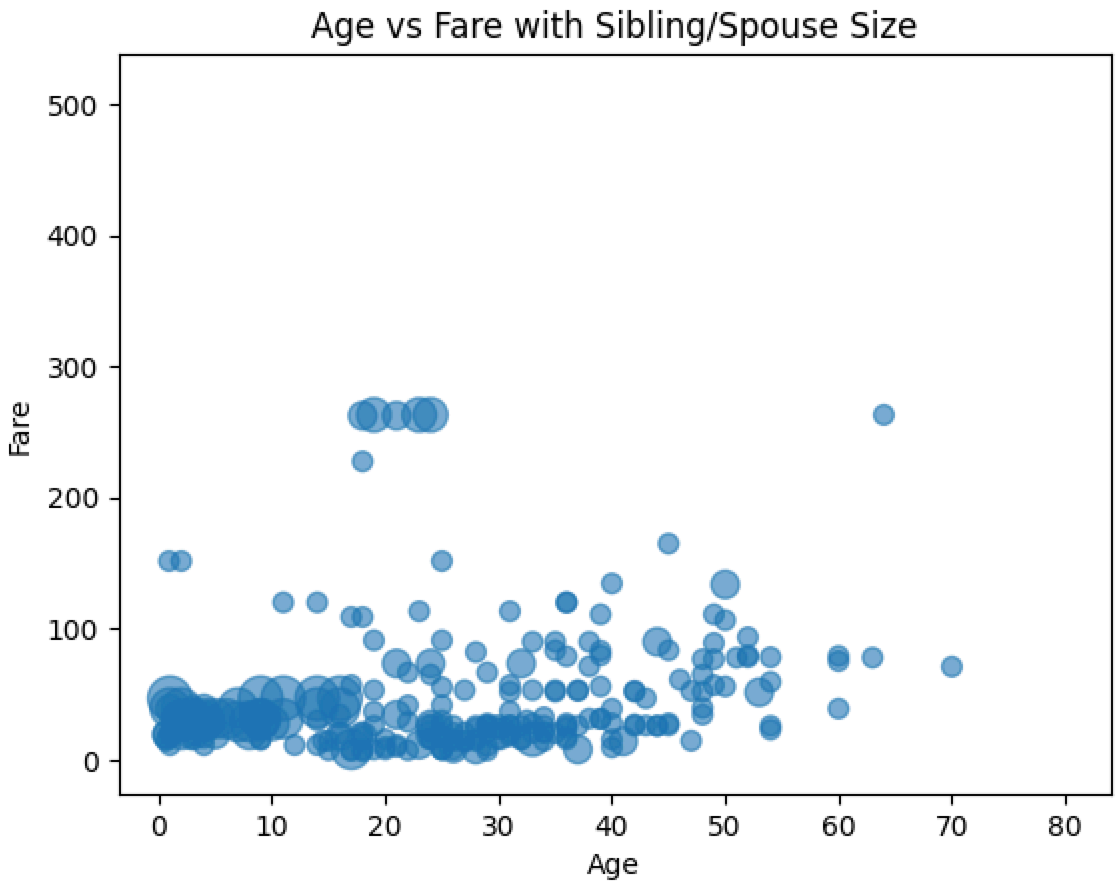
13. Create a cumulative histogram of the 'fare' column from the Titanic dataset with 50 bins and a blue color.

In [26]:
```
#Code here:
plt.hist(data['fare'].dropna(), bins=50, color='blue', cumulative=True, edgecc
plt.title("Cumulative Histogram of Fare")
plt.xlabel("Fare")
plt.ylabel("Cumulative Frequency")
plt.show()
```



14. Using the Titanic dataset plot a customized scatter plot of 'age' vs 'fare' with the size of the points determined by 'sibsp' (number of siblings/spouses aboard).

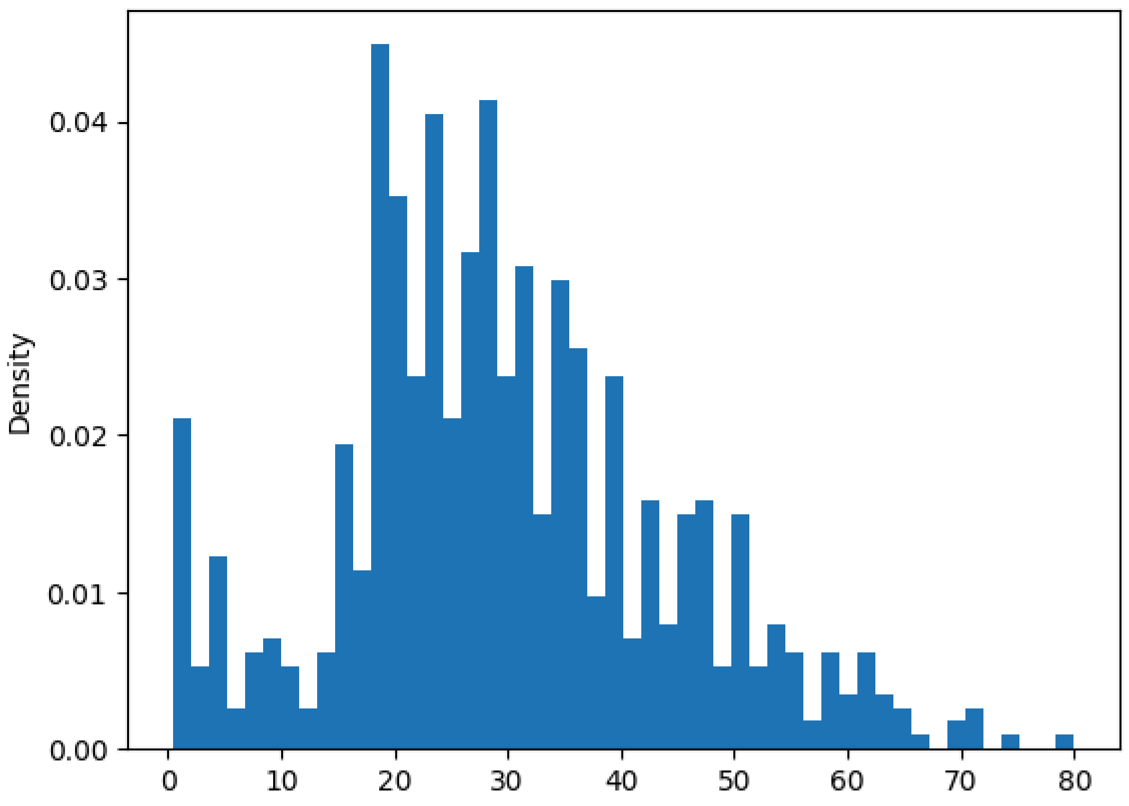
In [27]:
```
#Code here:
filtered_data = data.dropna(subset=['age', 'fare', 'sibsp'])
plt.scatter(filtered_data['age'], filtered_data['fare'], s=filtered_data['sibs
plt.title("Age vs Fare with Sibling/Spouse Size")
plt.xlabel("Age")
plt.ylabel("Fare")
plt.show()
```



---

15. Create a histogram of the 'age' column from the Titanic dataset, setting the y-axis label to 'Density', and enabling density normalization.

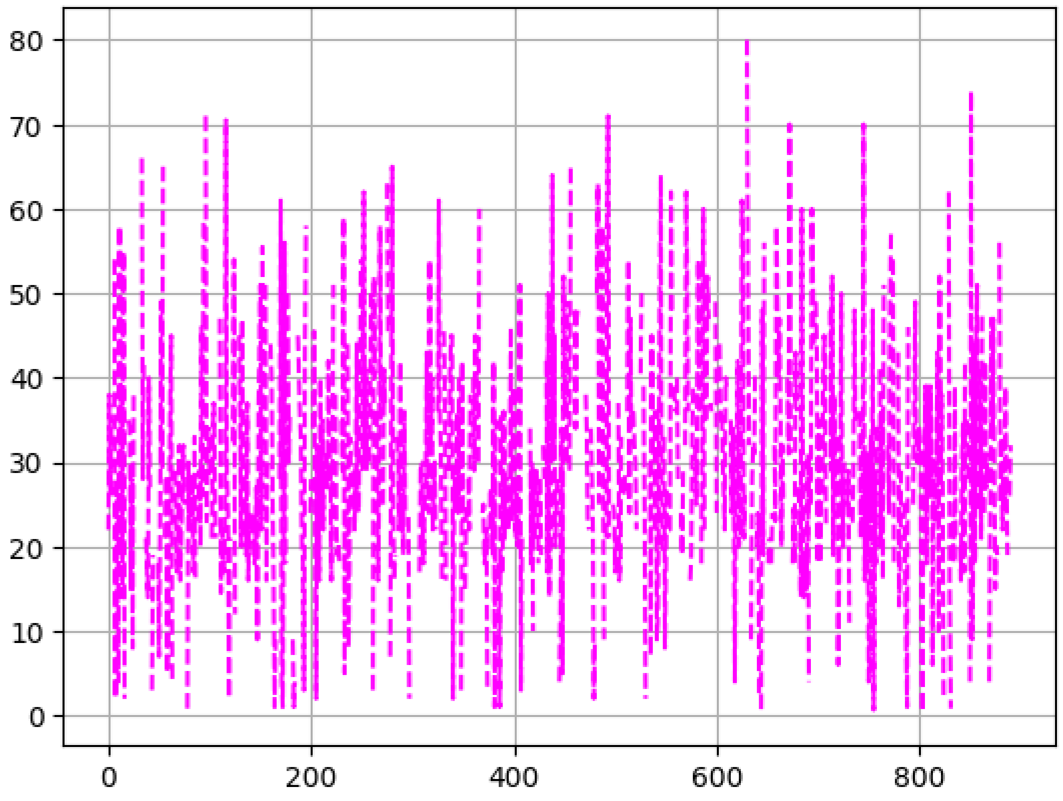
In [28]:
```
#Code here:
plt.hist(data['age'].dropna(), bins=50, density=True)
plt.ylabel('Density')
plt.show()
```



---

16. Using the Titanic dataset and create a line plot of the 'age' column. Customize the plot with a dashed line style, magenta color, and add a grid.

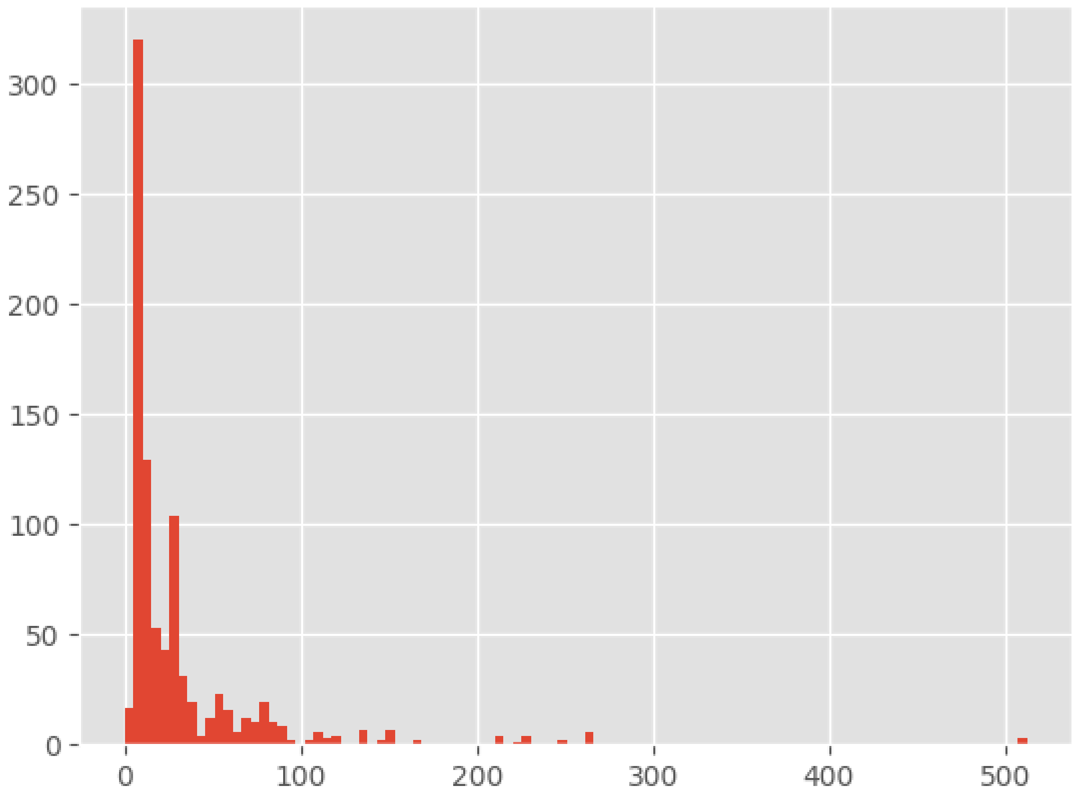
In [29]:
```
#Code here:
plt.plot(data['age'], linestyle='--', color='magenta')
plt.grid(True)
plt.show()
```



---

17. Using the Titanic dataset, plot a histogram of the 'fare' column with 100 bins. Customize the plot to use the 'ggplot' style.

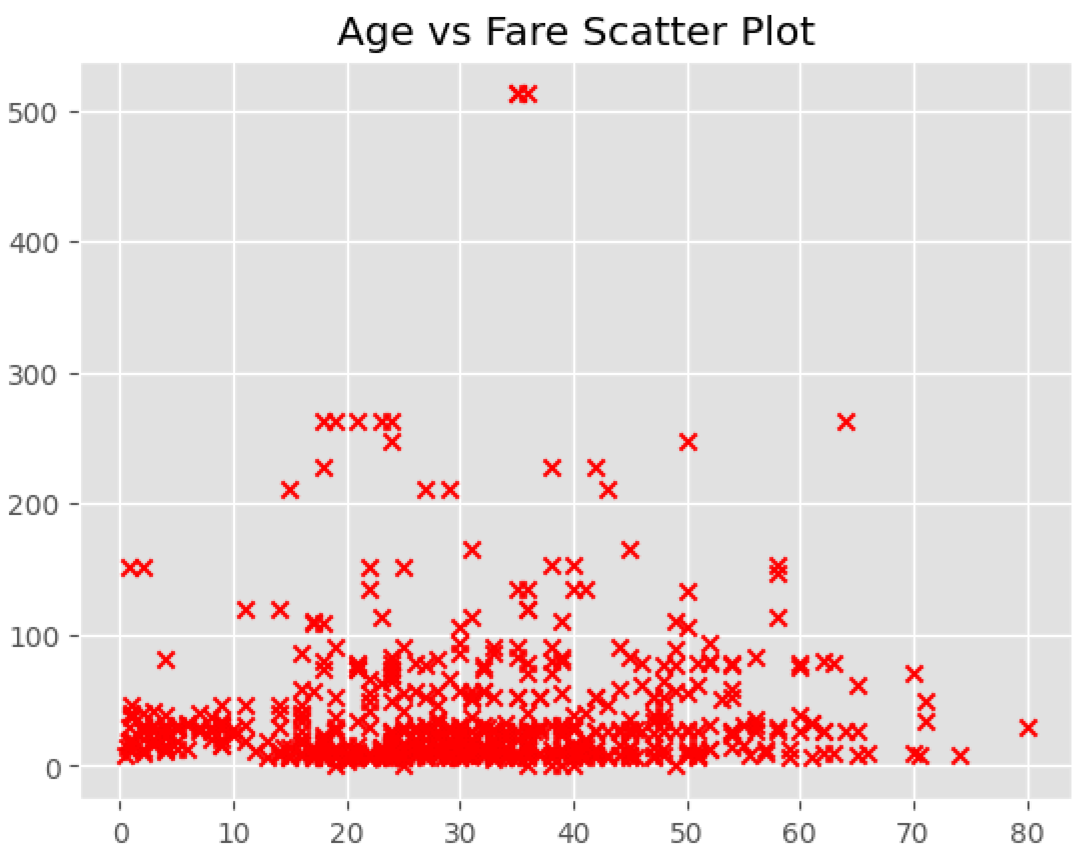
In [30]:
```
#Code here:
plt.style.use('ggplot')
plt.hist(data['fare'].dropna(), bins=100)
plt.show()
```



---

18. Create a scatter plot from the Titanic dataset with 'age' on the x-axis and 'fare' on the y-axis. Customize the plot with a red cross ('x') marker and set the plot title to "Age vs Fare Scatter Plot".

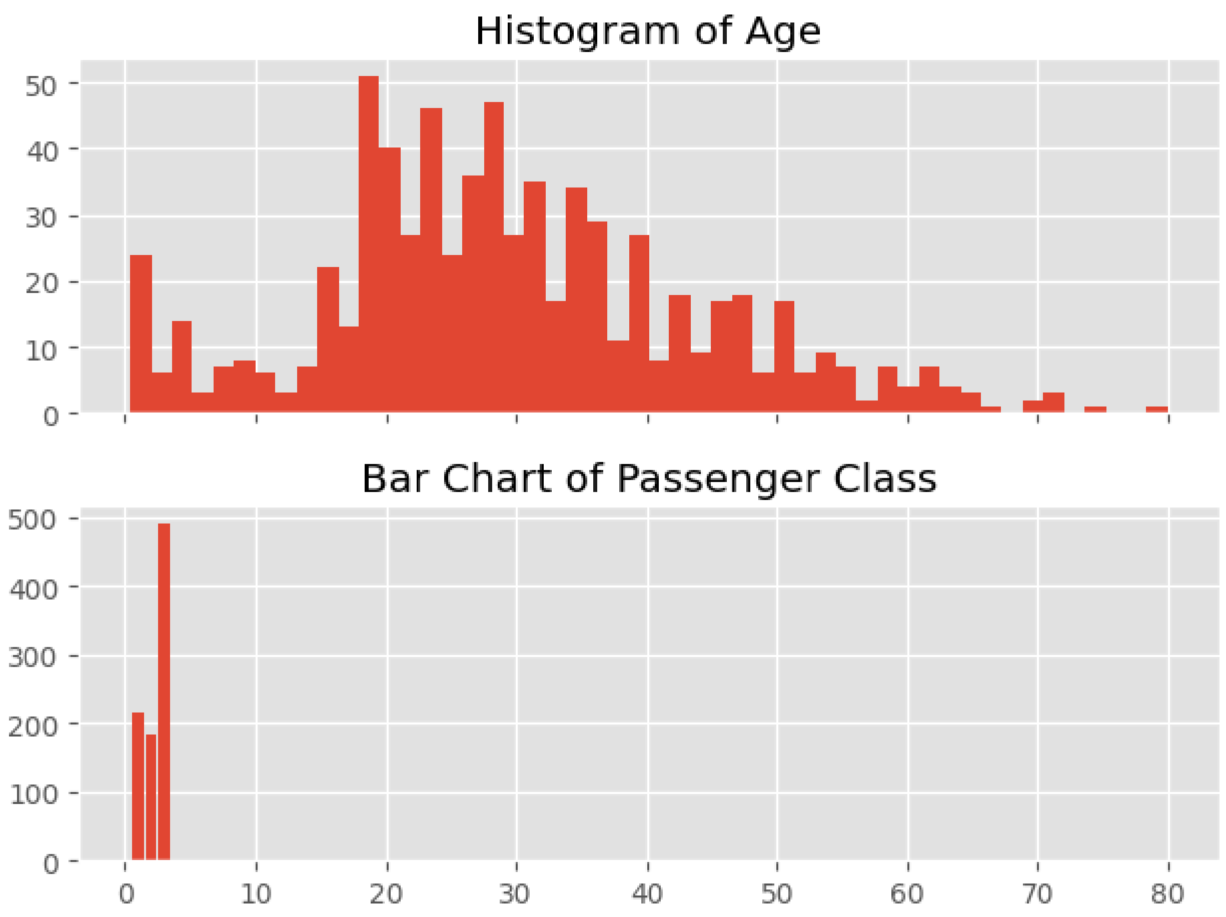
```
In [31]: #Code here:
         plt.scatter(data['age'], data['fare'], marker='x', color='red')
         plt.title("Age vs Fare Scatter Plot")
         plt.show()
```



---

19. Using the Titanic dataset create a figure with two subplots: one subplot showing a histogram of the 'age' column with 50 bins and the second subplot showing a bar chart of the 'pclass' column. Ensure that both plots share the same x-axis.

In [32]:
```
#Code here:
fig, axes = plt.subplots(2, 1, sharex=True)
axes[0].hist(data['age'].dropna(), bins=50)
axes[0].set_title("Histogram of Age")
pclass_counts = data['pclass'].value_counts().sort_index()
axes[1].bar(pclass_counts.index, pclass_counts.values)
axes[1].set_title("Bar Chart of Passenger Class")
plt.tight_layout()
plt.show()
```



---

20. Using the Titanic dataset create a figure with two subplots: the first subplot should display a line plot of the 'age' column, and the second subplot should display a bar chart of the 'sex' column showing the count of each gender.

In [ ]:
```
#Code here:
fig, axes = plt.subplots(2, 1)
axes[0].plot(data['age'])
axes[0].set_title("Line Plot of Age")
sex_counts = data['sex'].value_counts()
axes[1].bar(sex_counts.index, sex_counts.values)
axes[1].set_title("Bar Chart of Gender Distribution")
plt.tight_layout()
plt.show()
```

Good Luck!