In [1]:

```
!pip install pandas numpy matplotlib seaborn
```

```
Requirement already satisfied: pandas in /home/sargam/.conda/envs/myenv/li
b/python3.11/site-packages (2.2.3)
Requirement already satisfied: numpy in /home/sargam/.conda/envs/myenv/li
b/python3.11/site-packages (2.0.1)
Requirement already satisfied: matplotlib in /home/sargam/.conda/envs/myen
v/lib/python3.11/site-packages (3.10.1)
Requirement already satisfied: seaborn in /home/sargam/.conda/envs/myenv/l
ib/python3.11/site-packages (0.13.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /home/sargam/.con
da/envs/myenv/lib/python3.11/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /home/sargam/.conda/envs/my
env/lib/python3.11/site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /home/sargam/.conda/envs/
myenv/lib/python3.11/site-packages (from pandas) (2025.2)
Requirement already satisfied: contourpy>=1.0.1 in /home/sargam/.conda/env
s/myenv/lib/python3.11/site-packages (from matplotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /home/sargam/.conda/envs/my
env/lib/python3.11/site-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /home/sargam/.conda/en
vs/myenv/lib/python3.11/site-packages (from matplotlib) (4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /home/sargam/.conda/en
vs/myenv/lib/python3.11/site-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /home/sargam/.conda/env
s/myenv/lib/python3.11/site-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in /home/sargam/.conda/envs/myen
v/lib/python3.11/site-packages (from matplotlib) (11.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in /home/sargam/.conda/env
s/myenv/lib/python3.11/site-packages (from matplotlib) (3.2.3)
Requirement already satisfied: six>=1.5 in /home/sargam/.conda/envs/myenv/
lib/python3.11/site-packages (from python-dateutil>=2.8.2->pandas) (1.17.
0)
```

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:

```
# Load Titanic dataset from seaborn's built-in datasets
dataset = sns.load_dataset("titanic")
```

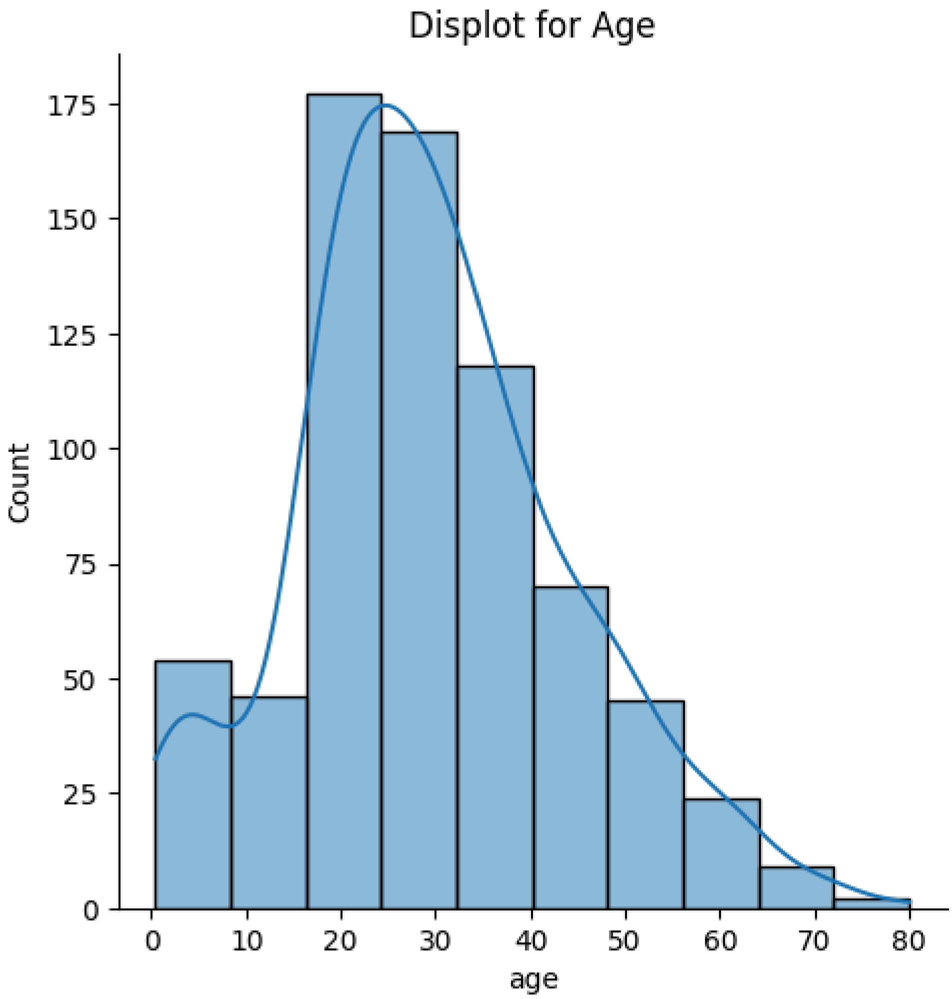
In [4]:

```
# Check the first few rows of the dataset
print(dataset.head())
```

```
   survived  pclass     sex   age  sibsp  parch     fare embarked  class  \
0         0       3    male  22.0      1      0   7.2500        S  Third   
1         1       1  female  38.0      1      0  71.2833        C  First   
2         1       3  female  26.0      0      0   7.9250        S  Third   
3         1       1  female  35.0      1      0  53.1000        S  First   
4         0       3    male  35.0      0      0   8.0500        S  Third   

     who  adult_male deck  embark_town alive  alone  
0    man        True  NaN  Southampton    no  False  
1  woman       False    C    Cherbourg   yes  False  
2  woman       False  NaN  Southampton   yes   True  
3  woman       False    C  Southampton   yes  False  
4    man        True  NaN  Southampton    no   True  
```

In [5]:
```
# Displot for Age
sns.displot(dataset['age'].dropna(), bins=10, kde=True)
plt.title('Displot for Age')
plt.show()
```



In [6]:
```
dataset.shape
```

Out[6]:
```
(891, 15)
```

In [7]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype   
---  ------       --------------  -----   
 0   survived     891 non-null    int64   
 1   pclass       891 non-null    int64   
 2   sex          891 non-null    object  
 3   age          714 non-null    float64 
 4   sibsp        891 non-null    int64   
 5   parch        891 non-null    int64   
 6   fare         891 non-null    float64 
 7   embarked     889 non-null    object  
 8   class        891 non-null    category
 9   who          891 non-null    object  
 10  adult_male   891 non-null    bool    
 11  deck         203 non-null    category
 12  embark_town  889 non-null    object  
 13  alive        891 non-null    object  
 14  alone        891 non-null    bool    
dtypes: bool(2), category(2), float64(2), int64(4), object(5)
memory usage: 80.7+ KB
```

In [8]:

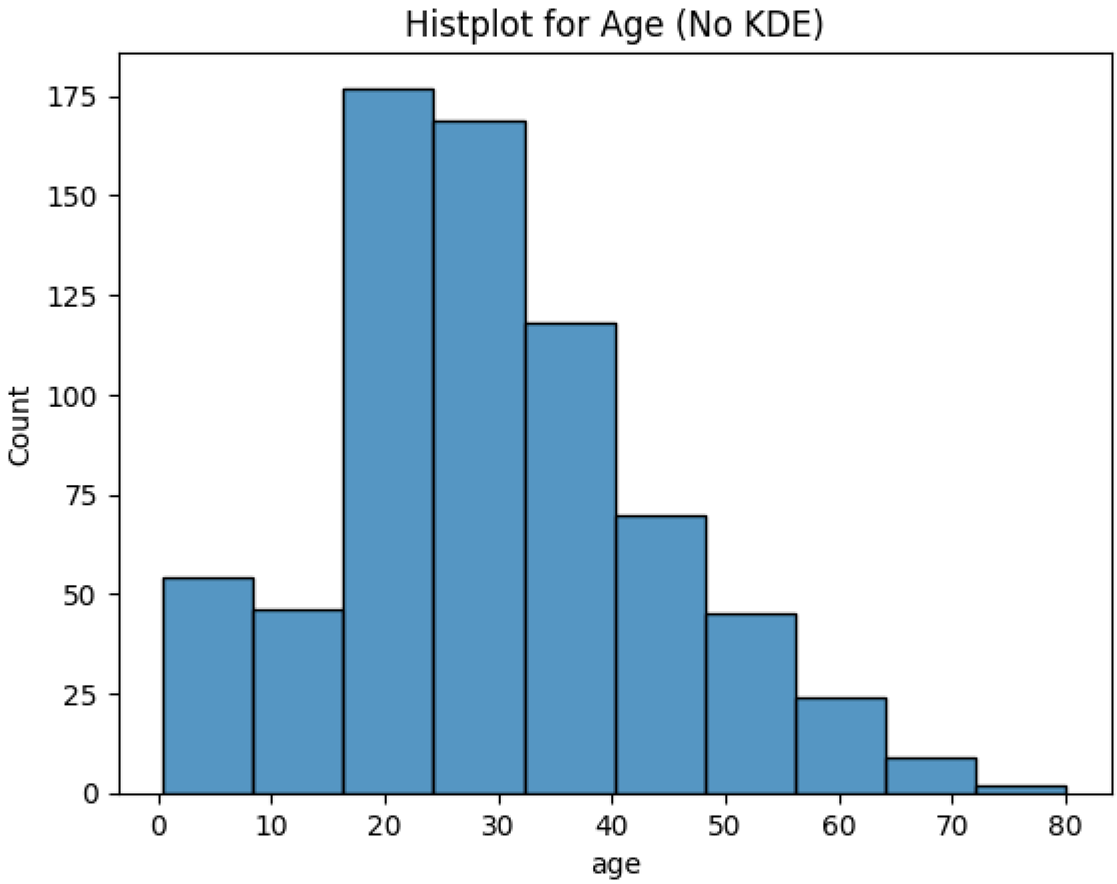
```python
dataset.isna().sum()
```

Out[8]:

```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

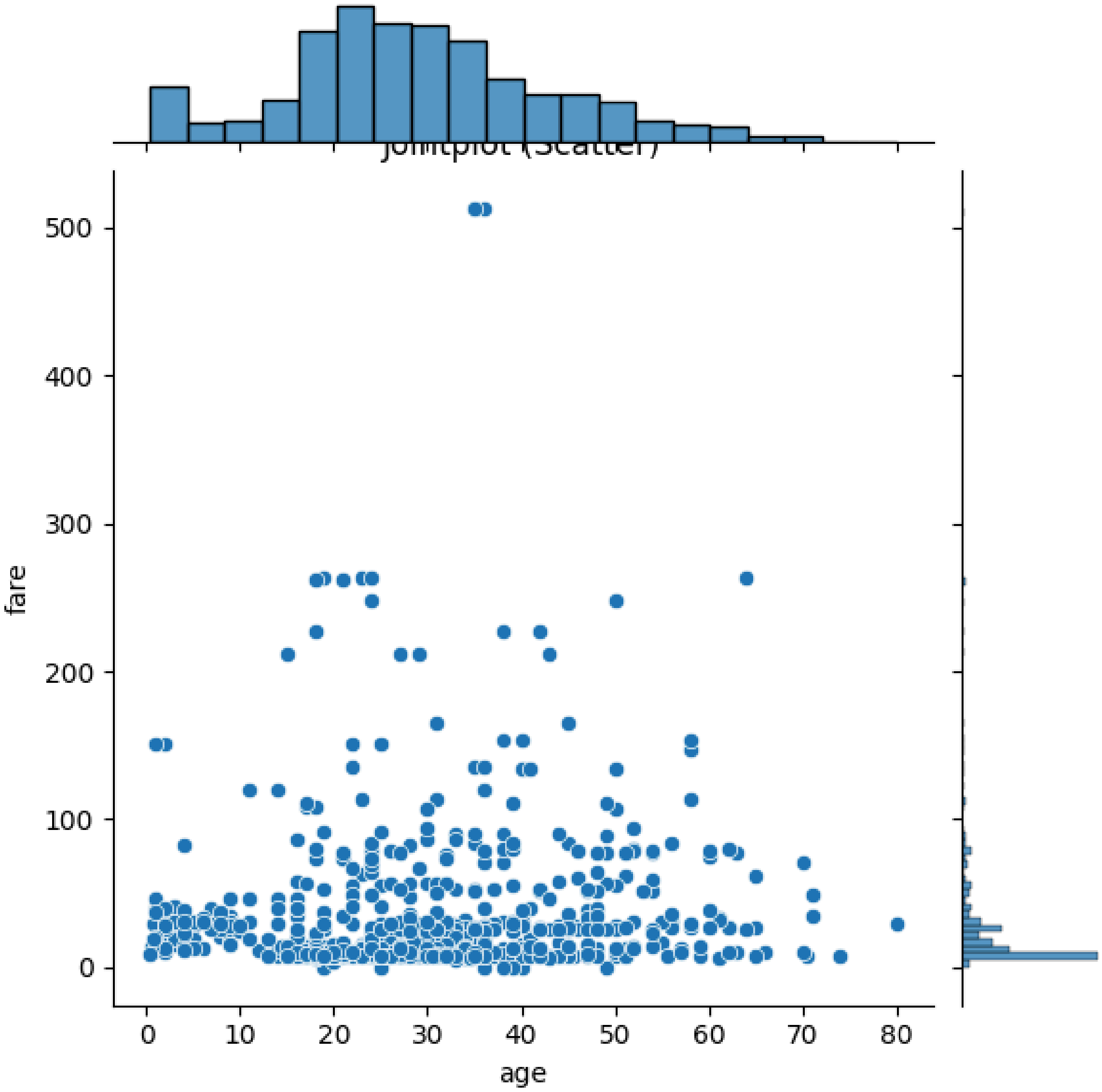
In [9]:

```python
# Histplot for Age without KDE
sns.histplot(dataset['age'].dropna(), bins=10, kde=False)
plt.title('Histplot for Age (No KDE)')
plt.show()
```



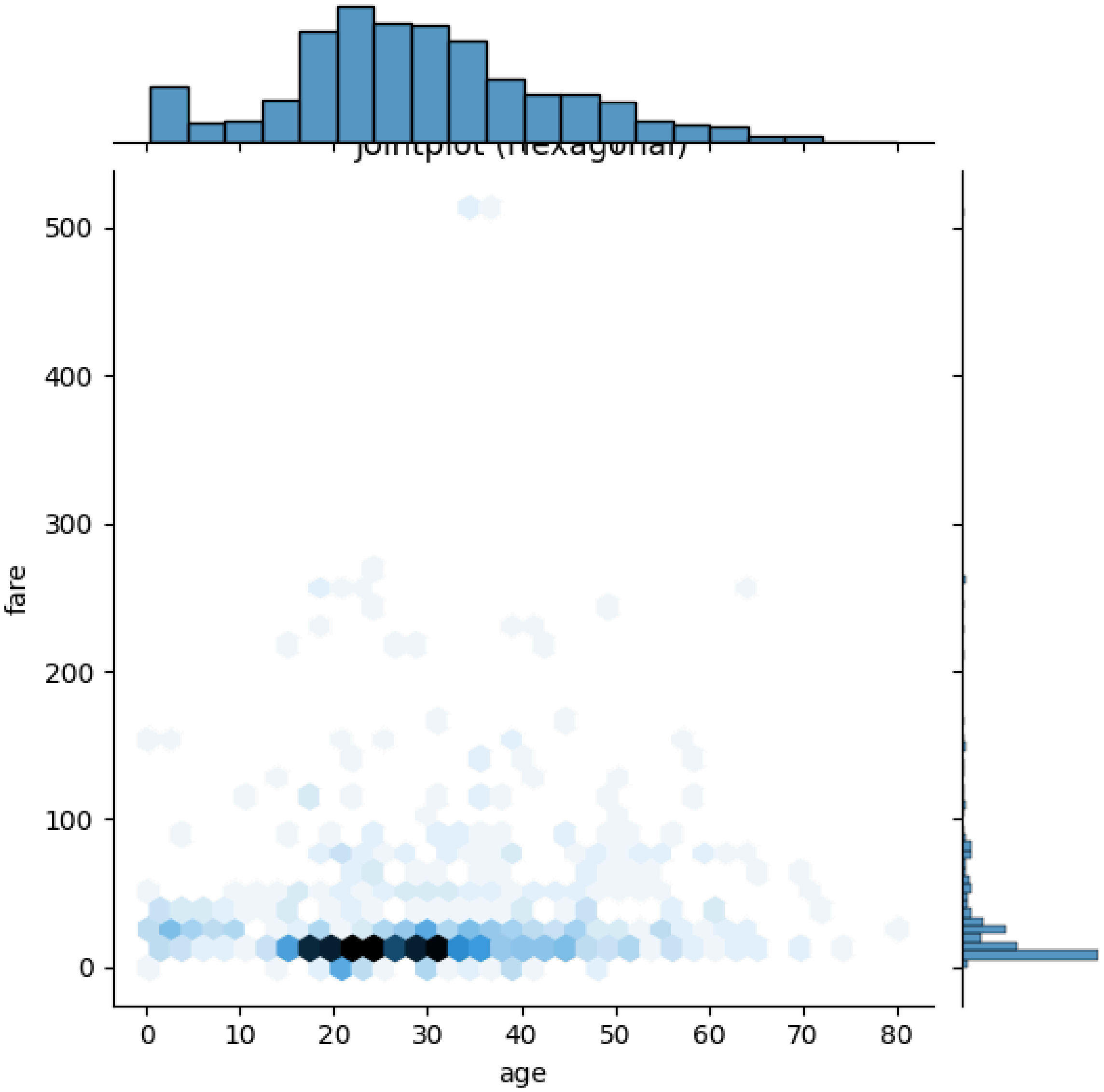
In [10]:

```
# Jointplot (Scatter) between Age and Fare
sns.jointplot(x='age', y='fare', kind='scatter', data=dataset)
plt.title('Jointplot (Scatter)')
plt.show()
```



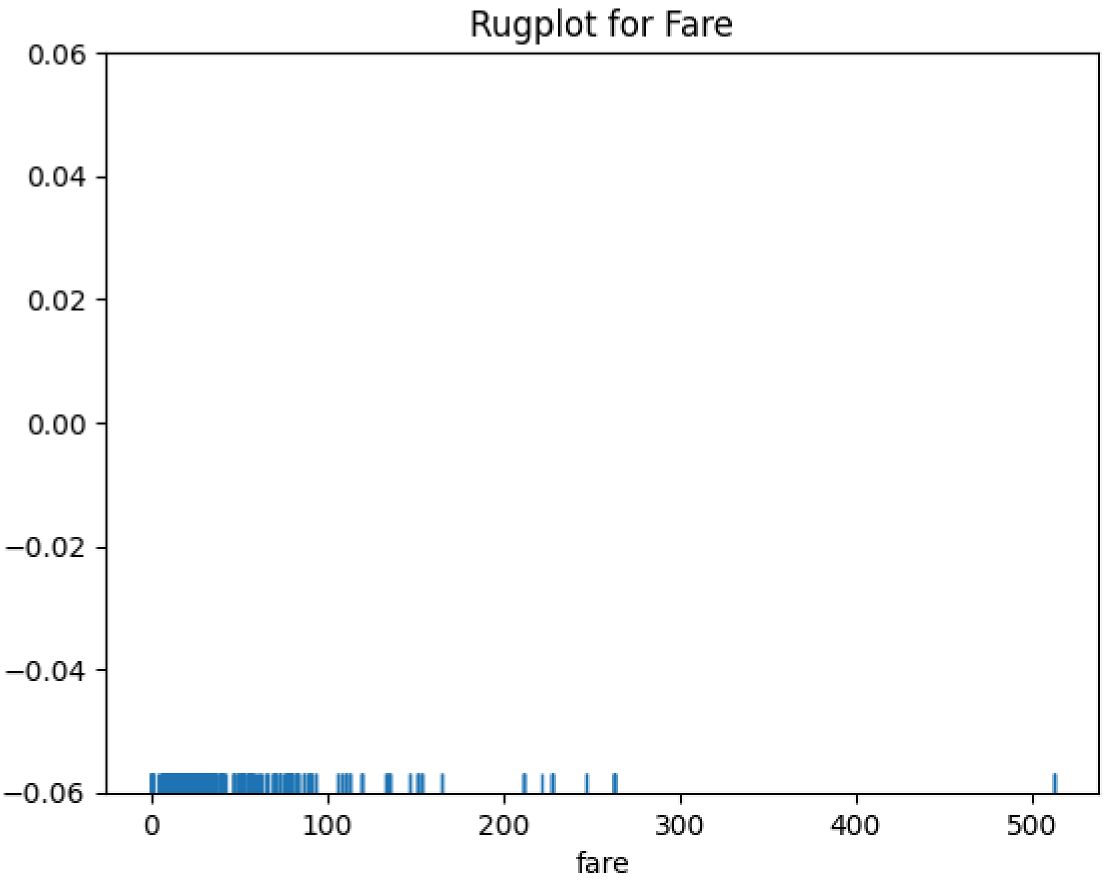
In [11]:

```
# Jointplot (Hexagonal) between Age and Fare
sns.jointplot(x='age', y='fare', kind='hex', data=dataset)
plt.title('Jointplot (Hexagonal)')
plt.show()
```

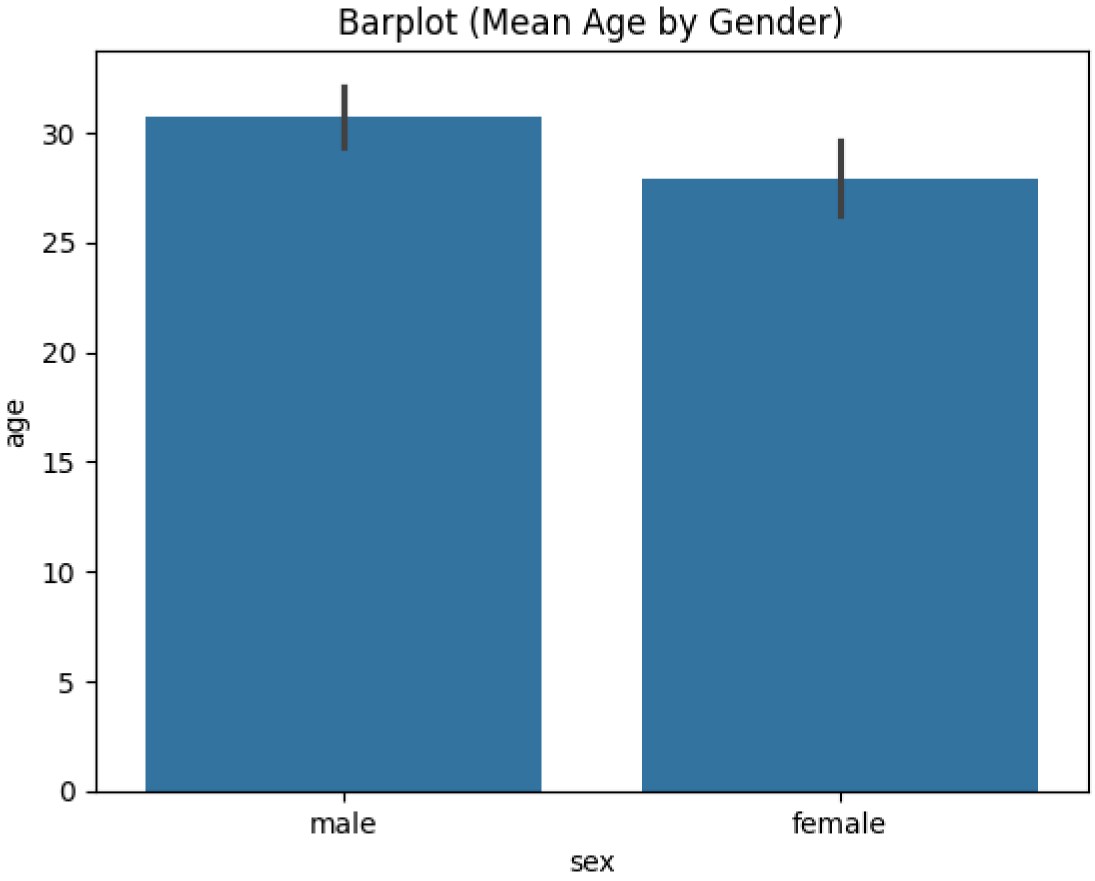


In [12]:

```
# Rugplot for Fare
sns.rugplot(dataset['fare'])
plt.title('Rugplot for Fare')
plt.show()
```

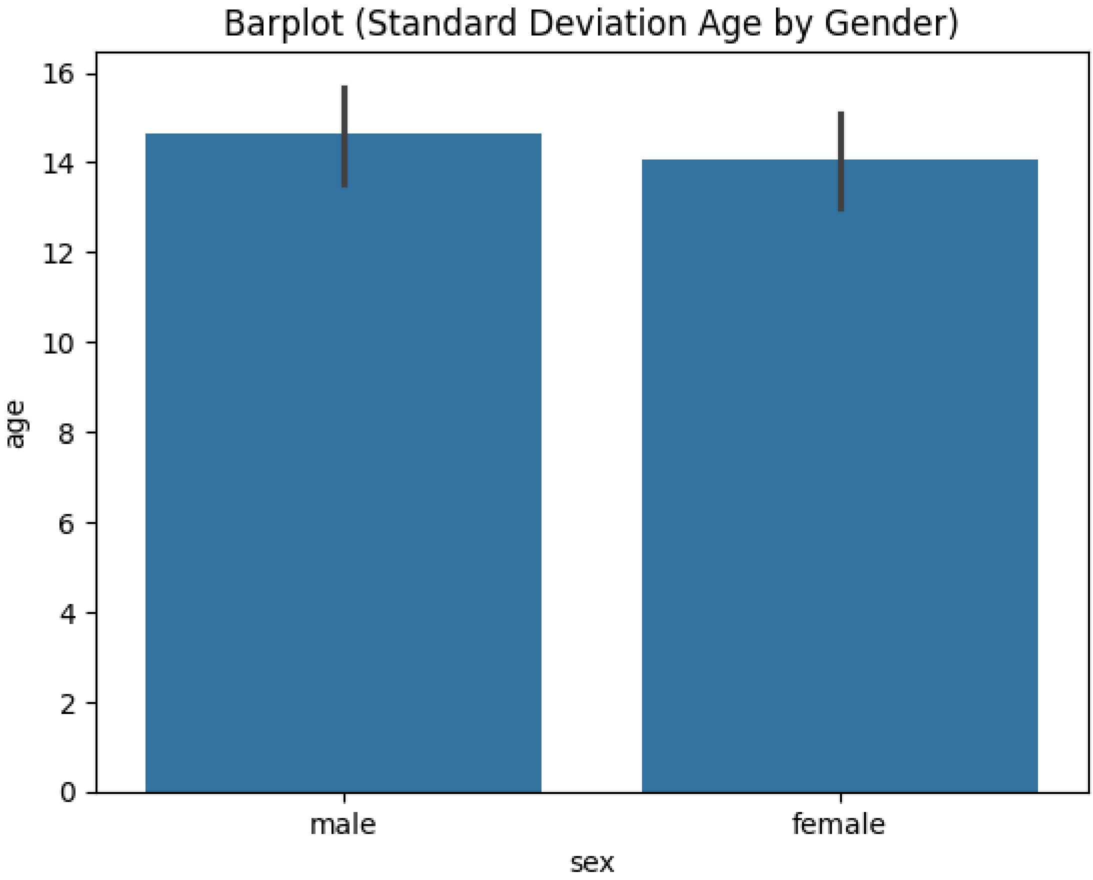


```
In [13]: # Barplot (Mean Age by Gender)
         sns.barplot(x='sex', y='age', data=dataset)
         plt.title('Barplot (Mean Age by Gender)')
         plt.show()
```



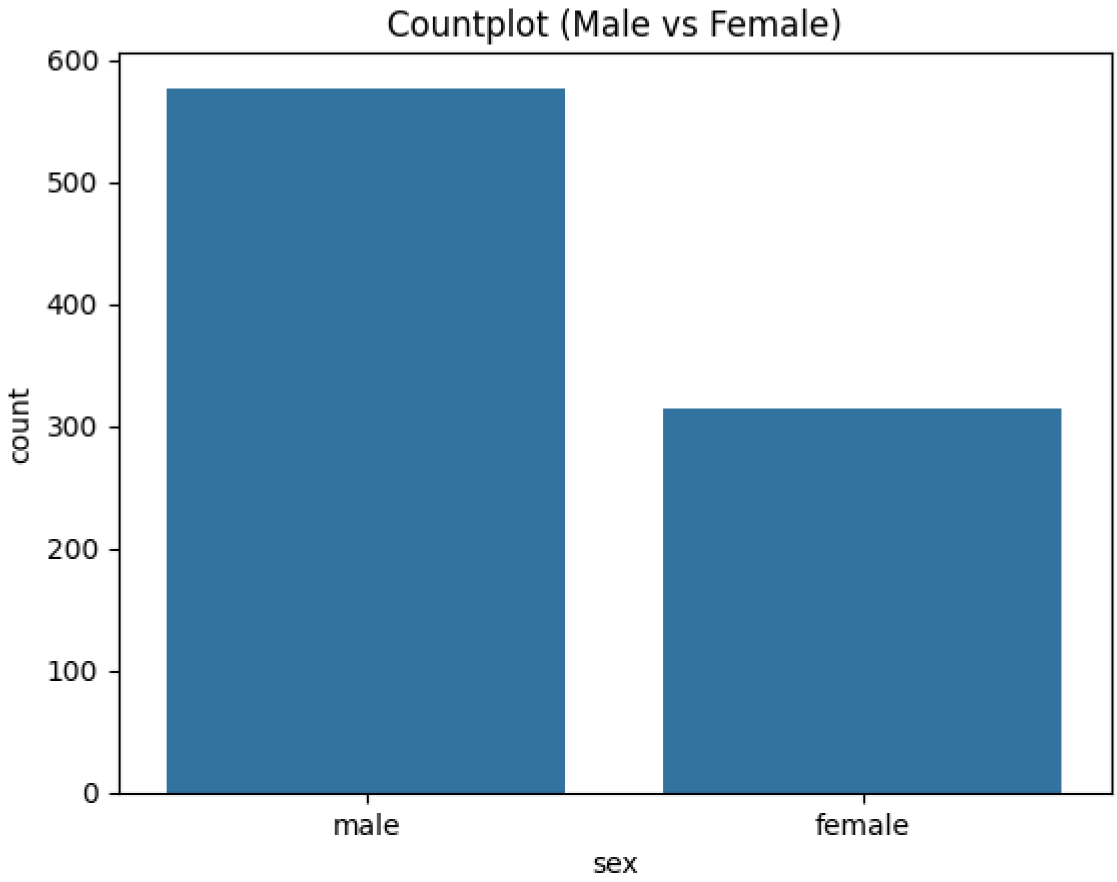
In [14]:

```
# Barplot (Standard Deviation Age by Gender)
sns.barplot(x='sex', y='age', data=dataset, estimator=np.std)
plt.title('Barplot (Standard Deviation Age by Gender)')
plt.show()
```



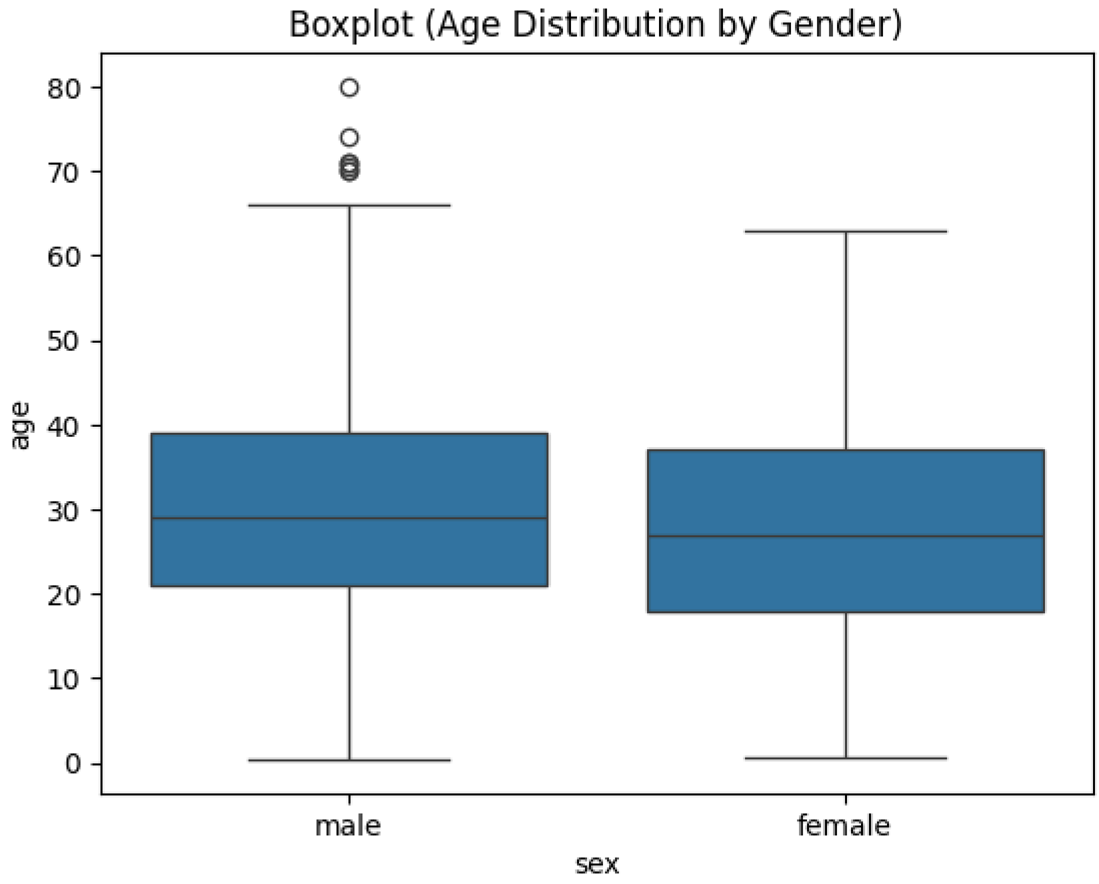
In [15]:

```
# Countplot (Male vs Female)
sns.countplot(x='sex', data=dataset)
plt.title('Countplot (Male vs Female)')
plt.show()
```



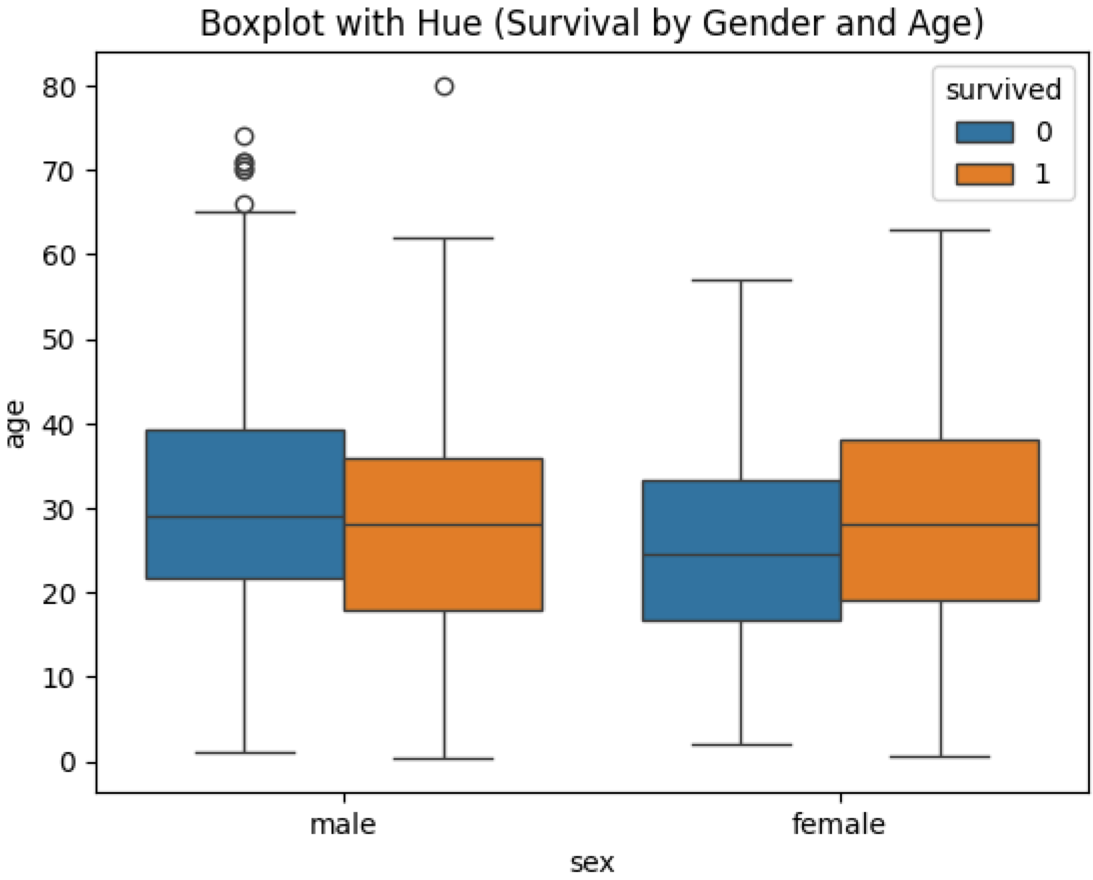
In [16]:

```
# Boxplot (Age Distribution by Gender)
sns.boxplot(x='sex', y='age', data=dataset)
plt.title('Boxplot (Age Distribution by Gender)')
plt.show()
```



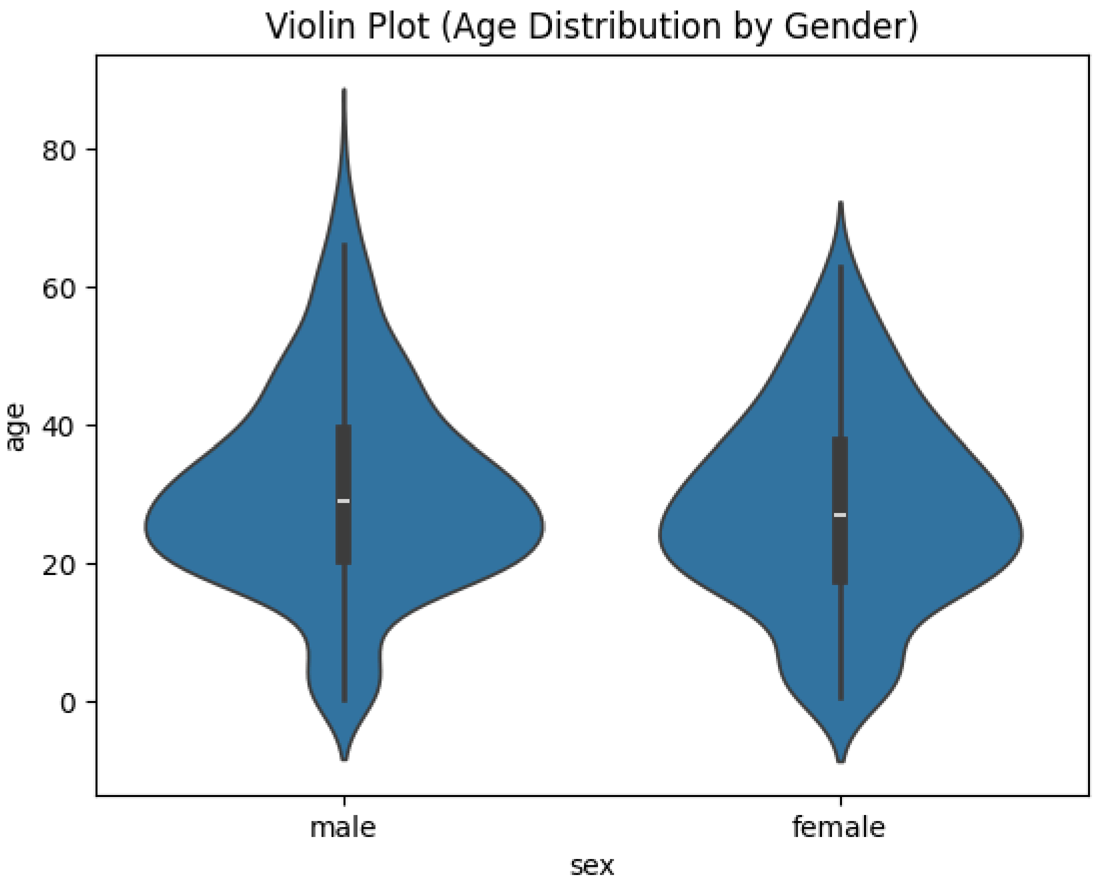
In [17]:

```python
# Boxplot with Hue (Survival by Gender and Age)
sns.boxplot(x='sex', y='age', data=dataset, hue="survived")
plt.title('Boxplot with Hue (Survival by Gender and Age)')
plt.show()
```



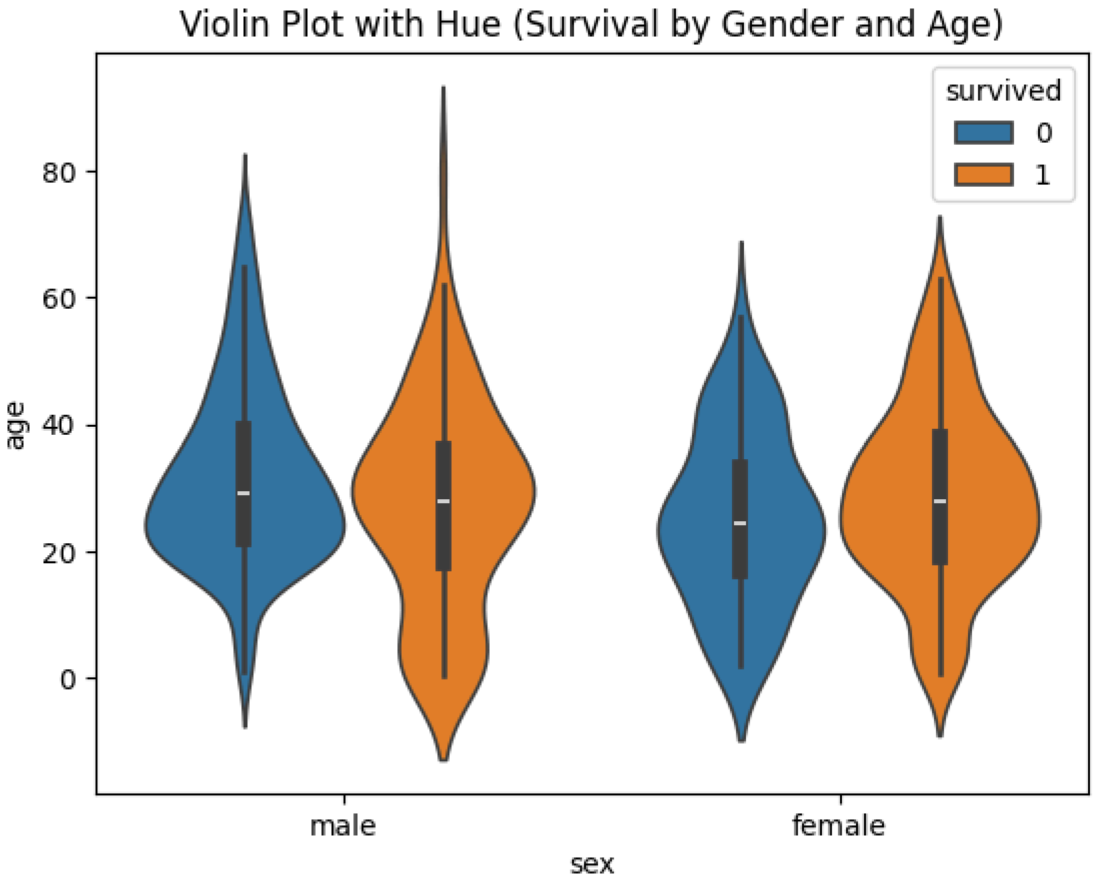
In [18]:

```python
# Violin Plot (Age Distribution by Gender)
sns.violinplot(x='sex', y='age', data=dataset)
plt.title('Violin Plot (Age Distribution by Gender)')
plt.show()
```



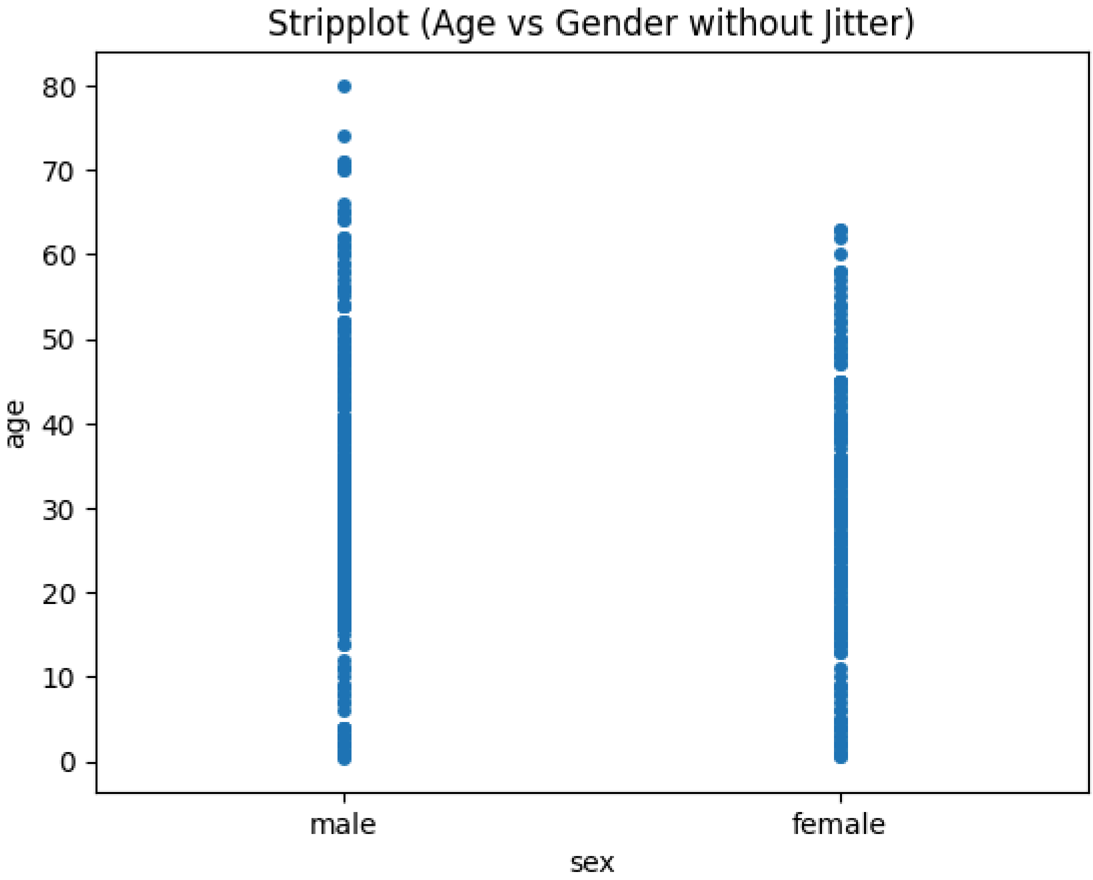
In [19]:

```python
# Violin Plot with Hue (Survival by Gender and Age)
sns.violinplot(x='sex', y='age', data=dataset, hue='survived')
plt.title('Violin Plot with Hue (Survival by Gender and Age)')
plt.show()
```



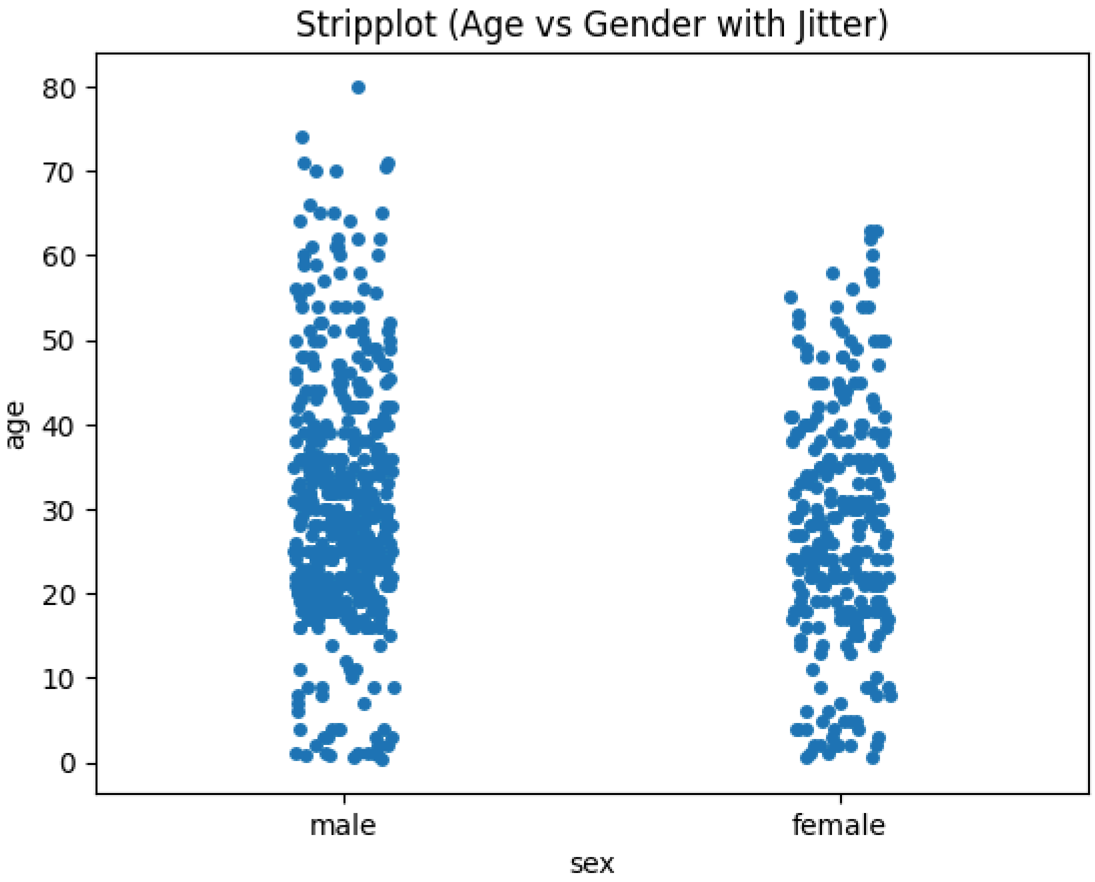
In [20]:

```
# Stripplot (Age vs Gender without Jitter)
sns.stripplot(x='sex', y='age', data=dataset, jitter=False)
plt.title('Stripplot (Age vs Gender without Jitter)')
plt.show()
```



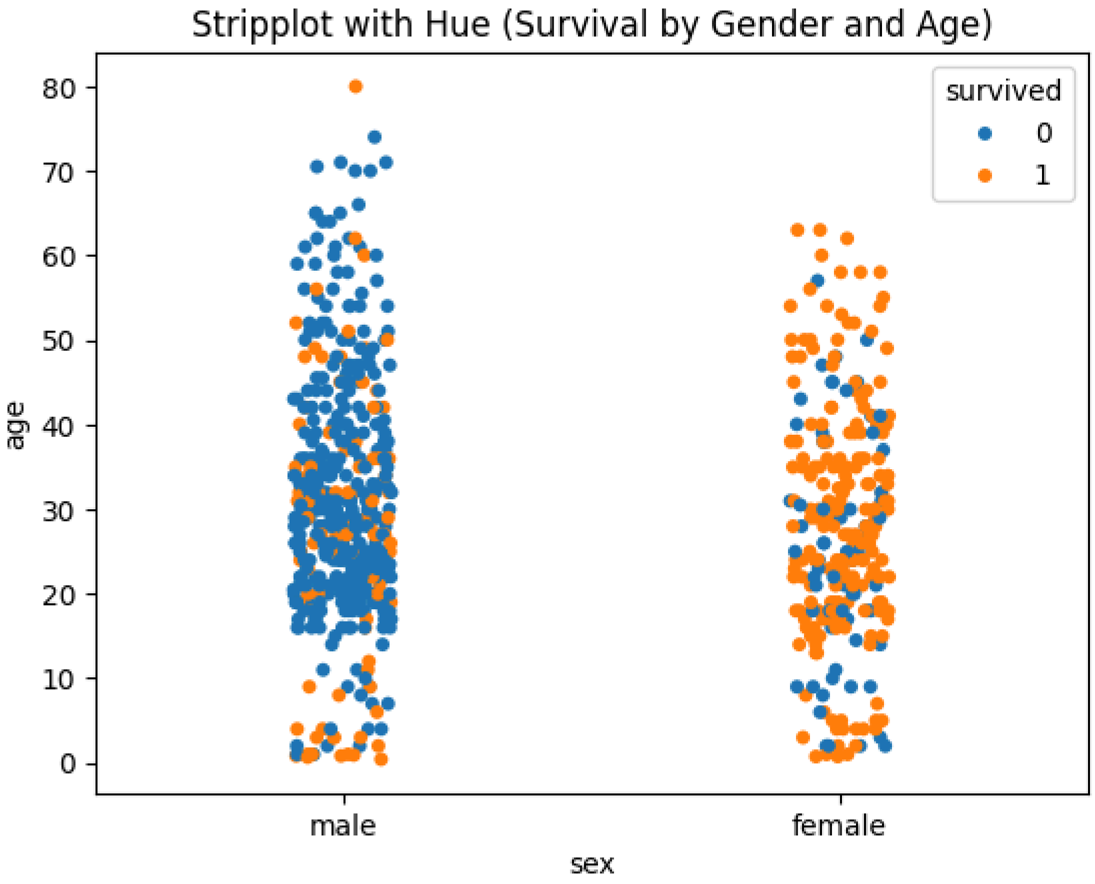
In [21]:

```
# Stripplot (Age vs Gender with Jitter)
sns.stripplot(x='sex', y='age', data=dataset, jitter=True)
plt.title('Stripplot (Age vs Gender with Jitter)')
plt.show()
```



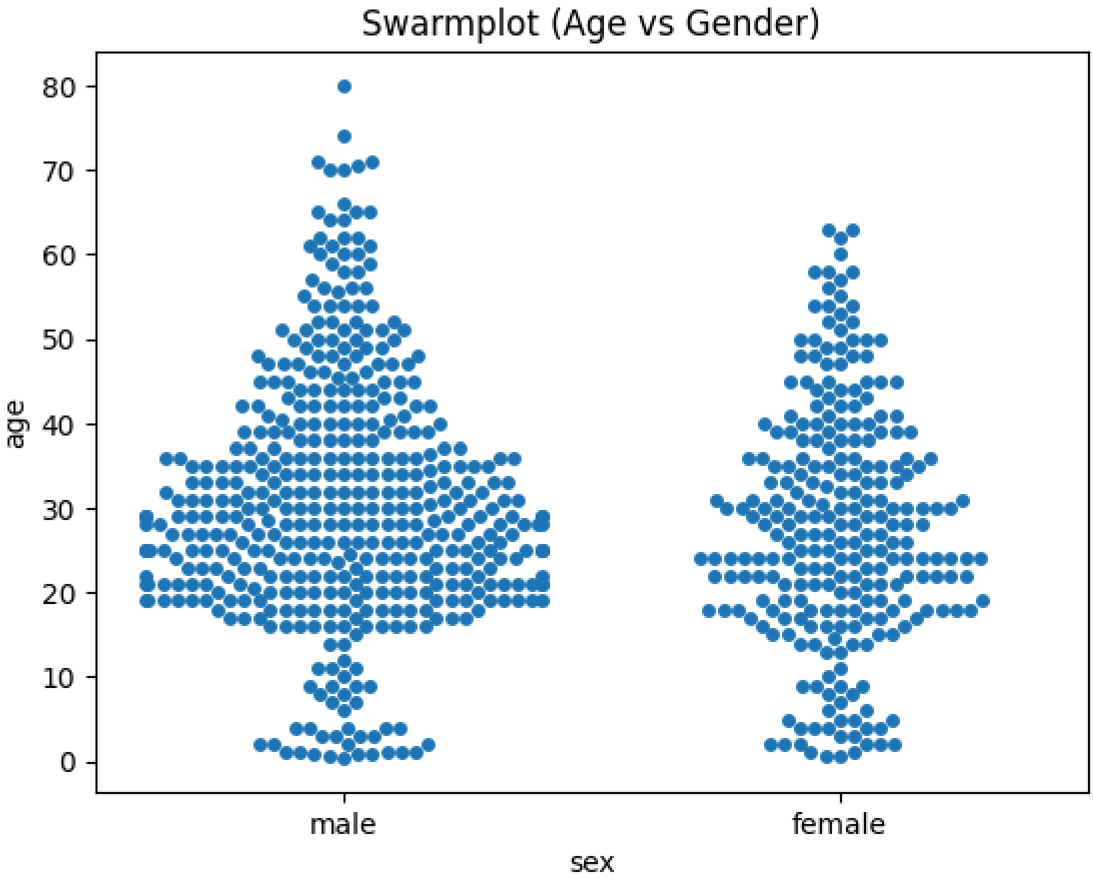
In [22]:

```
# Stripplot with Hue (Survival by Gender and Age)
sns.stripplot(x='sex', y='age', data=dataset, jitter=True, hue='survived'
plt.title('Stripplot with Hue (Survival by Gender and Age)')
plt.show()
```



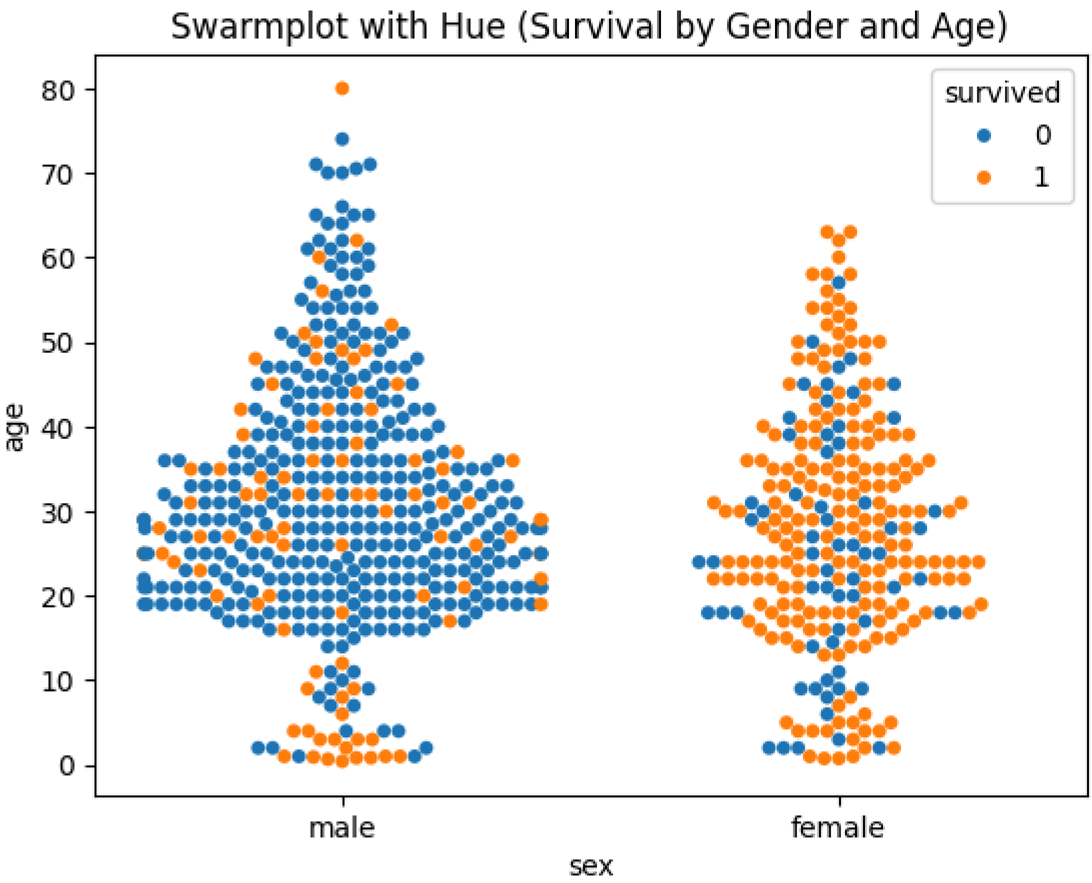
In [23]:

```
# Swarmplot (Age vs Gender)
sns.swarmplot(x='sex', y='age', data=dataset)
plt.title('Swarmplot (Age vs Gender)')
plt.show()
```



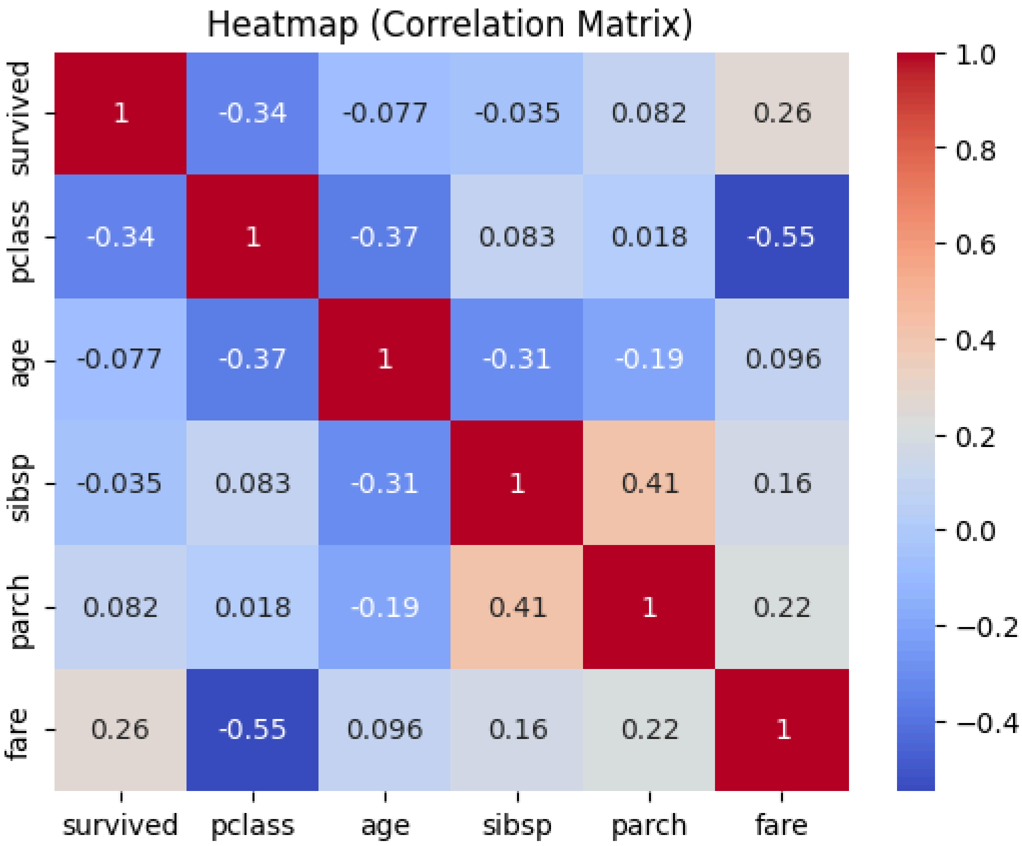
In [24]:

```
# Swarmplot with Hue (Survival by Gender and Age)
sns.swarmplot(x='sex', y='age', data=dataset, hue='survived')
plt.title('Swarmplot with Hue (Survival by Gender and Age)')
plt.show()
```

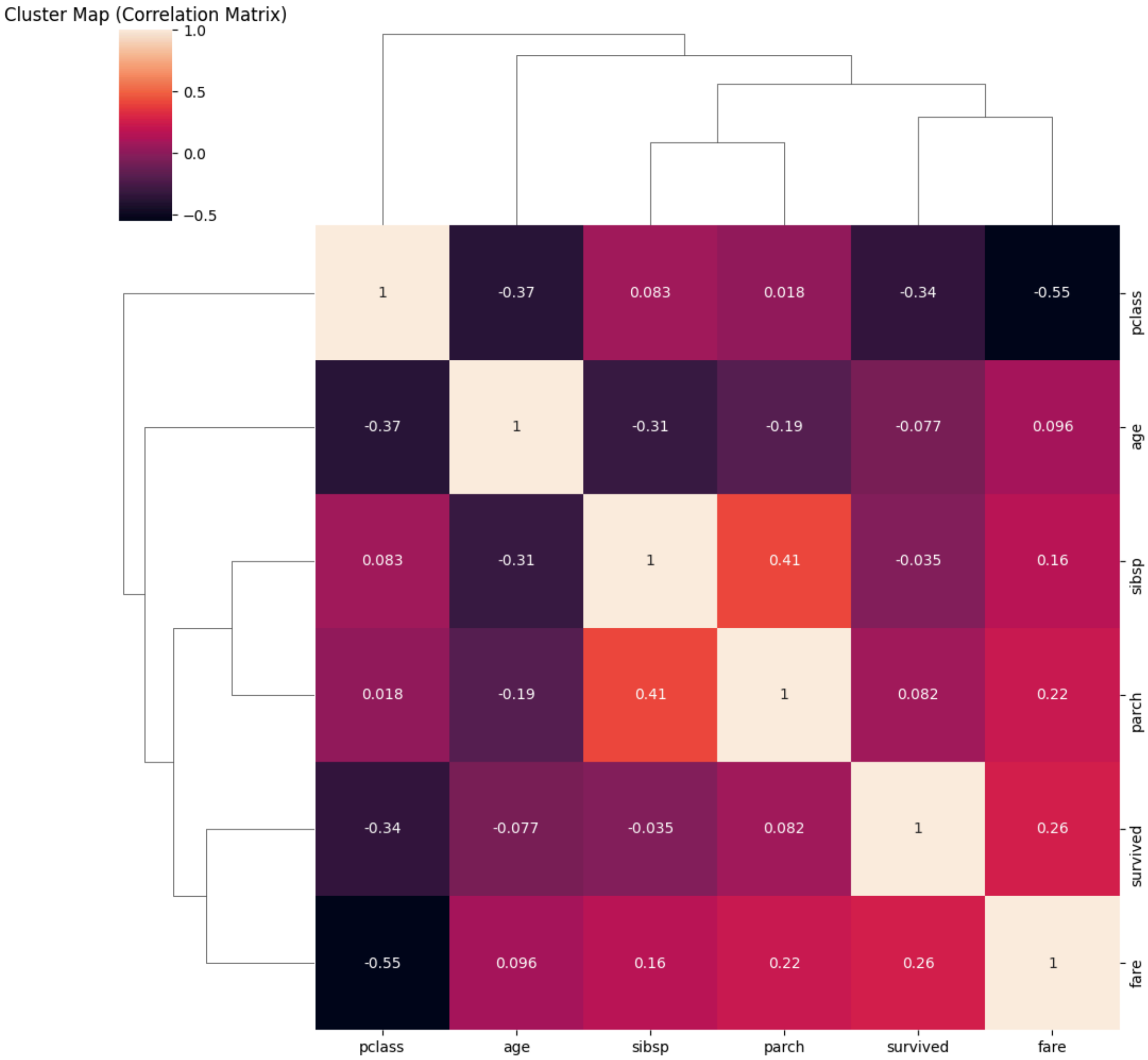


In [25]:

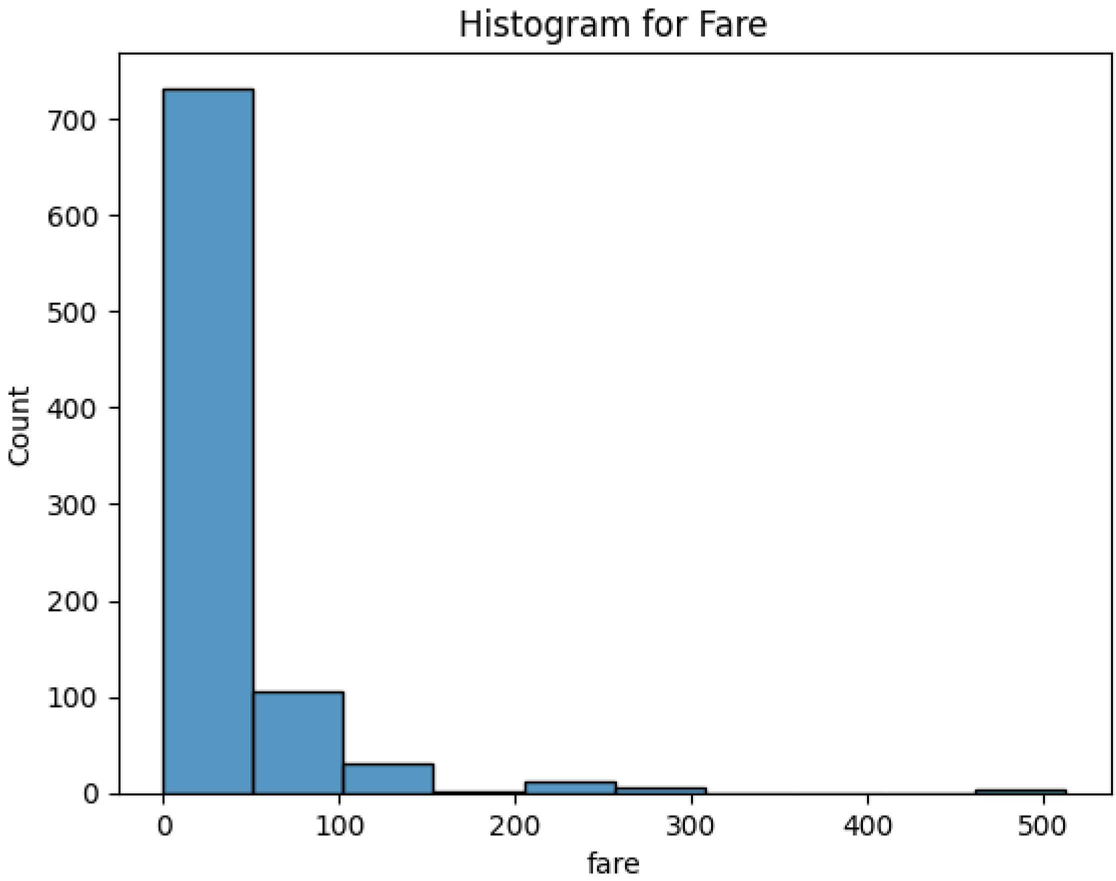
```
# Correlation Matrix Heatmap
numeric_cols = dataset.select_dtypes(include=[np.number])
corr = numeric_cols.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Heatmap (Correlation Matrix)')
plt.show()
```



```
In [26]: # Cluster Map (Correlation Matrix)
         sns.clustermap(corr, annot=True)
         plt.title('Cluster Map (Correlation Matrix)')
         plt.show()
```

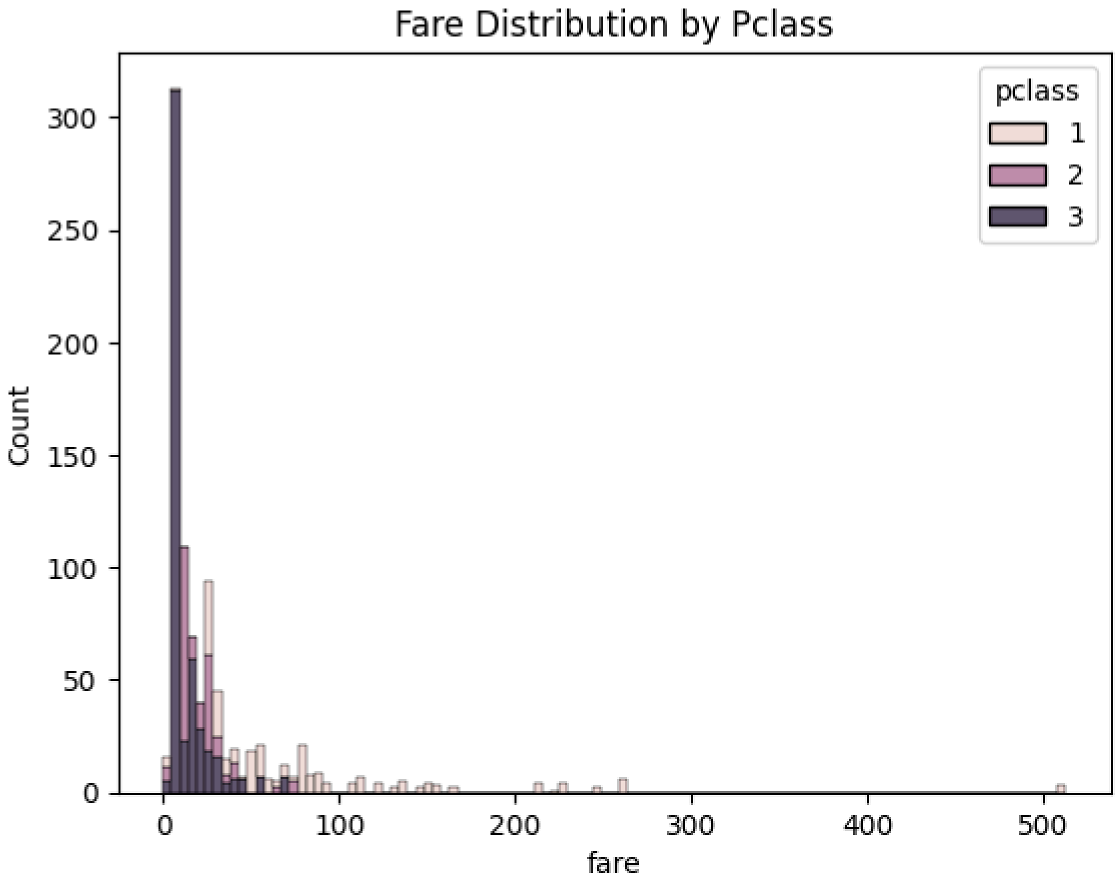


In [27]:
```
# Histogram for Fare
sns.histplot(dataset['fare'].dropna(), kde=False, bins=10)
plt.title('Histogram for Fare')
plt.show()
```

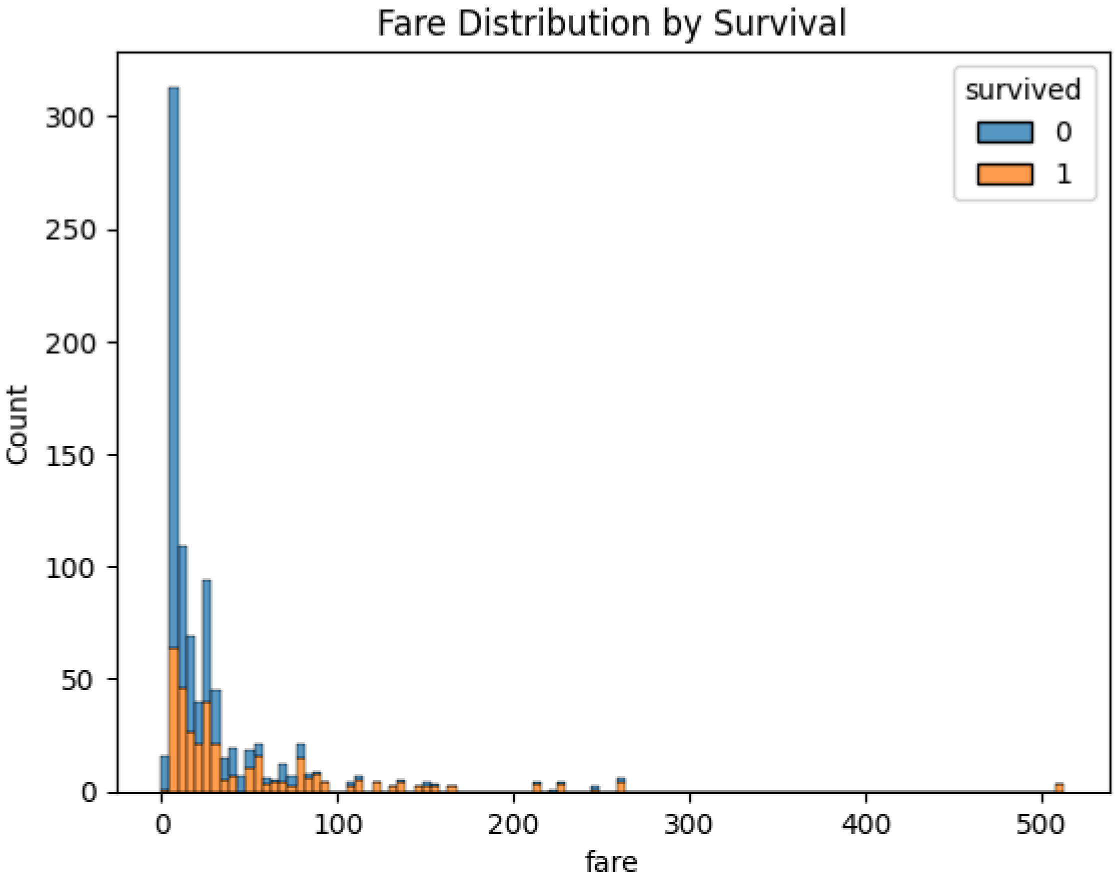

Histogram for Fare

In [30]:

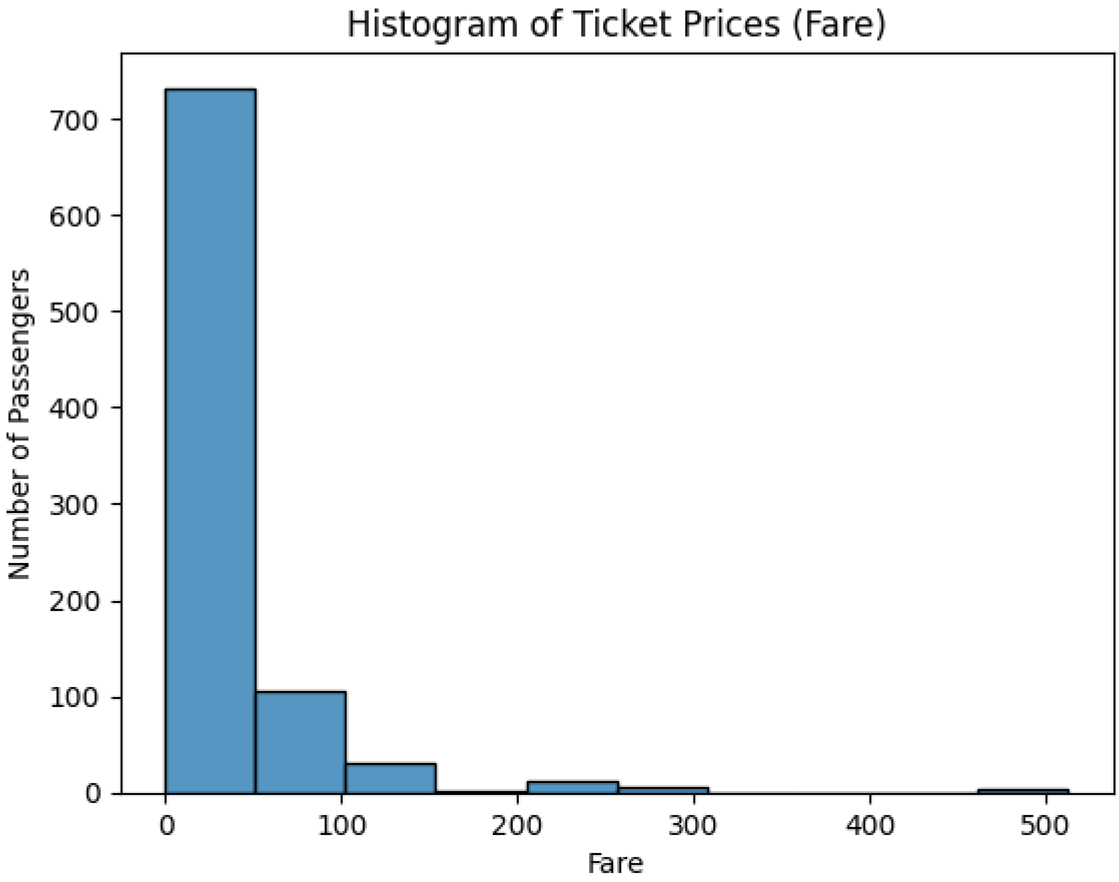
```
sns.histplot(data=dataset, x='fare', hue='pclass', multiple="stack")
plt.title('Fare Distribution by Pclass')
plt.show()
```


Fare Distribution by Pclass

In [32]:
```
sns.histplot(data=dataset, x='fare', hue='survived', multiple="stack")
plt.title('Fare Distribution by Survival')
plt.show()
```



In [33]:
```
# Plot histogram for 'fare'
sns.histplot(dataset['fare'].dropna(), kde=False, bins=10)
plt.title('Histogram of Ticket Prices (Fare)')
plt.xlabel('Fare')
plt.ylabel('Number of Passengers')
plt.show()
```


Histogram of Ticket Prices (Fare)
Number of Passengers
0
100
200
300
400
500
600
700
Fare
0
100
200
300
400
500

In [ ]: