PROJECT

Pandas & Numpy functions, Visualisation on Titanic Dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
sns.get_dataset_names()
```

```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

```
# Titanic dataset

d=sns.load_dataset('titanic')
```

+ Code + Text

```
d.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

```
df=pd.read_csv('/content/titanic.csv')
df
```

	index	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False

```
# Q & A on Pandas functions

# 1. What are datatypes of each column ?
# ans:-
print(d.dtypes)
```

```
survived          int64
pclass            int64
sex              object
age             float64
sibsp             int64
parch             int64
fare            float64
embarked         object
class          category
who              object
adult_male         bool
deck           category
embark_town      object
alive            object
alone              bool
dtype: object
```

```
# 2. what is the average age of passengers ?
# ans:-

print(d['age'].mean())
```

```
29.69911764705882
```

```
# 3. how many passengers survived ?
# ans:-

print(d['survived'].sum())
```

```
342
```

```
# 4. What is the most common embarkation port ?
# ans :-

print(d['embarked'].value_counts().index[0])
```

```
S
```

```
# 5. What is he median fare ?
# ans :-

print(d['fare'].median())
```

```
14.4542
```

```
# 6. what is the highest fare paid ?
# ans:-

print(d['fare'].max())
```

```
512.3292
```

```
# 7. how many passengers had more than 2 siblings/spouses onboard ?
# ans:-

print(len(d[d['sibsp']>2]))
```

```
46
```

```
# 8. what is survival rate of female passengers ?
# ans:-

tf=len(d[d['sex']=='female'])
sf=len(d[(d['sex']=='female') & (d['survived']==1)])
print(sf/tf)
```

```
0.7420382165605095
```

```
# 9. what is the total number of missing data ?
# ans:-

print(d.isnull().sum().sum())
```

```
869
```

```
# 10. What is the range of ages of passengers on the Titanic?
# ans:-

print(d['age'].max() - d['age'].min())
```

```
79.58
```

```python
# Q & A on Numpy functions

# 1. what is the variance of passengers ages ?
# ans:-

ages=d['age'].to_numpy()
print(np.var(ages))
```

```
nan
```

```python
# 2. What is the standard deviation of fares ?
# ans :-

fr=d['fare'].to_numpy()
print(np.std(fr))
```

```
49.6655344447741
```

```python
# 3. How many female passengers were onboard ?
# ans:-

fm=d[d[:]=='female']
print(len(fm))
```

```
891
```

```
# 4. What is the pearson correlation coefficient between age and fare ?
# ans :-

ag=d['age'].to_numpy()
fr=d['fare'].to_numpy()
cr=np.corrcoef(ag,fr)[0,1]
print(cr)
```

```
nan
```

```
# 5. What is the survival rate of passengers in third class ?
# ans:-

print(d.loc[d['pclass'] == 3, 'survived'].sum() / d.loc[d['pclass'] == 3, 'survived'].count())
```

```
0.24236252545824846
```

```
# 6. How many passengers were traveling alone?
# ans:-

d['TravelAlone'] = np.where((d['sibsp'] + d['parch']) > 0, 'No', 'Yes')
print(d['TravelAlone'].value_counts()['Yes'])
```

```
537
```

```
# 7. How many unique values are in the 'Sex' column?
# ans:-

print(d['sex'].nunique())
```

```
2
```

```
# 8. What is the survival rate of passengers in second class?
# ans:-

print(d.loc[d['pclass'] == 2, 'survived'].sum() / d.loc[d['pclass'] == 2, 'survived'].count())
```

```
0.47282608695652173
```

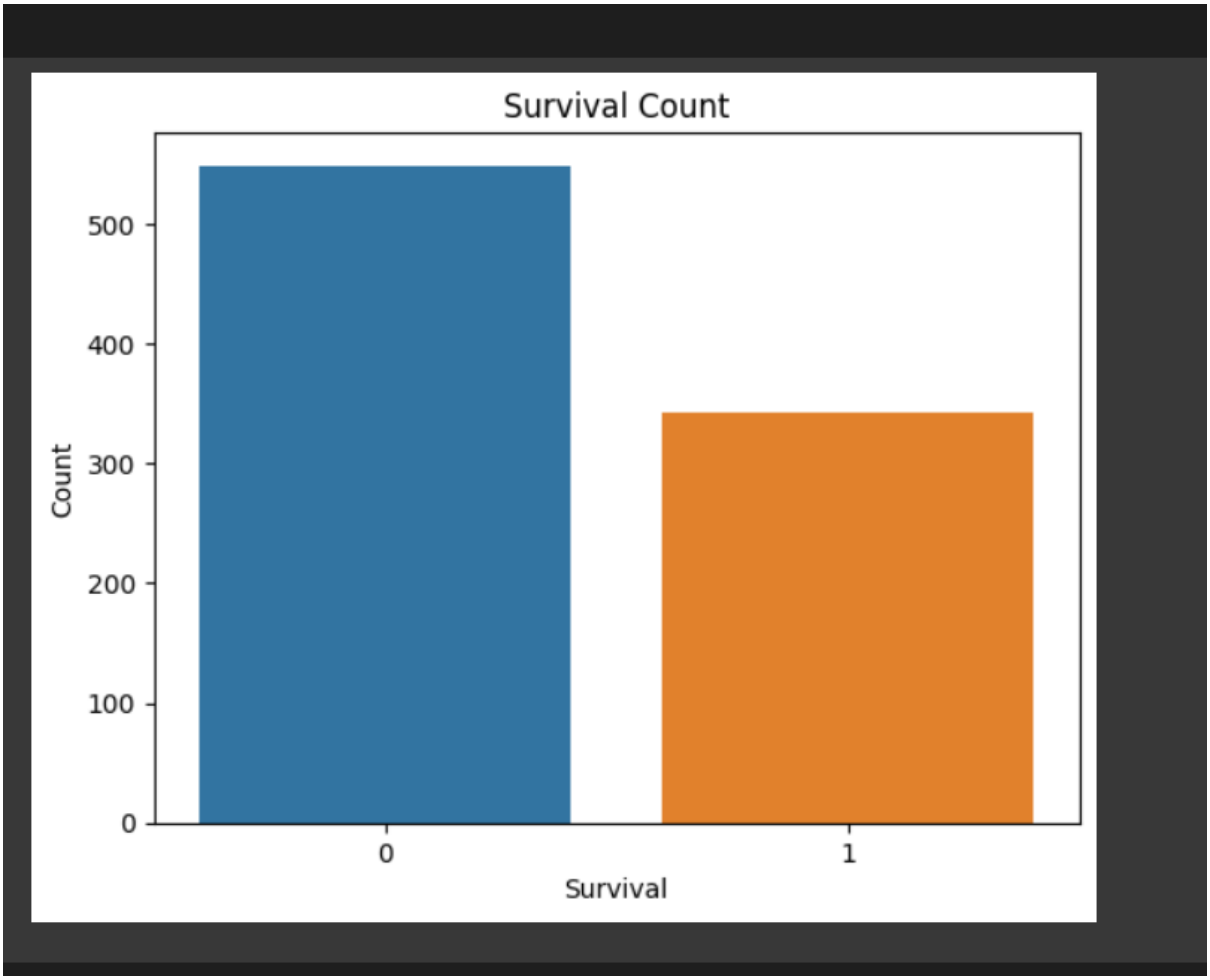
```
# 9. What is the survival rate of passengers who embarked from Southampton?
# ans:-

print(d.loc[d['embarked'] == 'S', 'survived'].sum() / d.loc[d['embarked'] == 'S', 'survived'].count())
```

```
0.33695652173913043
```

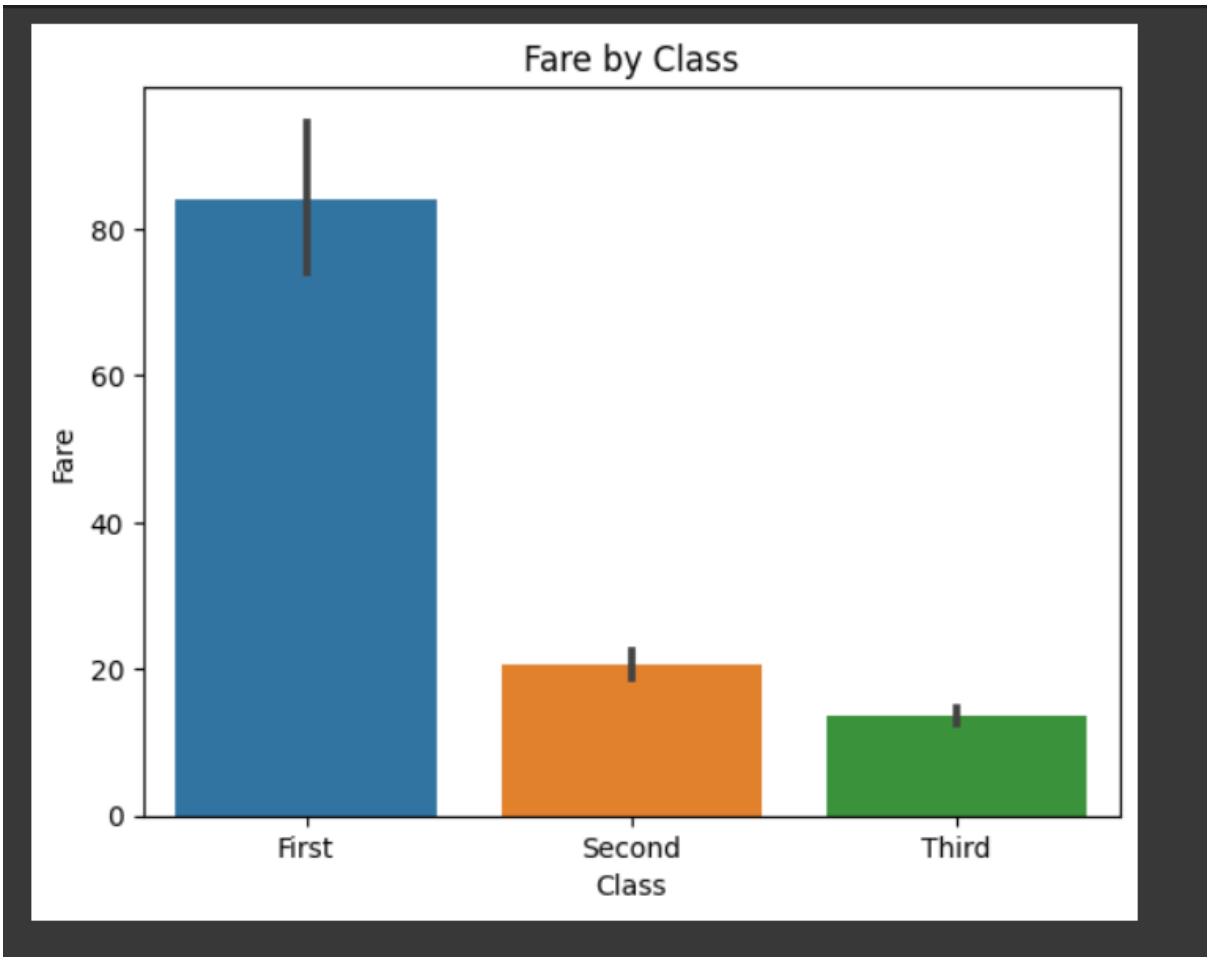
```
# 10. What is the survival rate of male passengers?
# ans:-

print(d.loc[d['sex'] == 'male', 'survived'].sum() / d.loc[d['sex'] == 'male', 'survived'].count())
```

```
0.18890814558058924
```

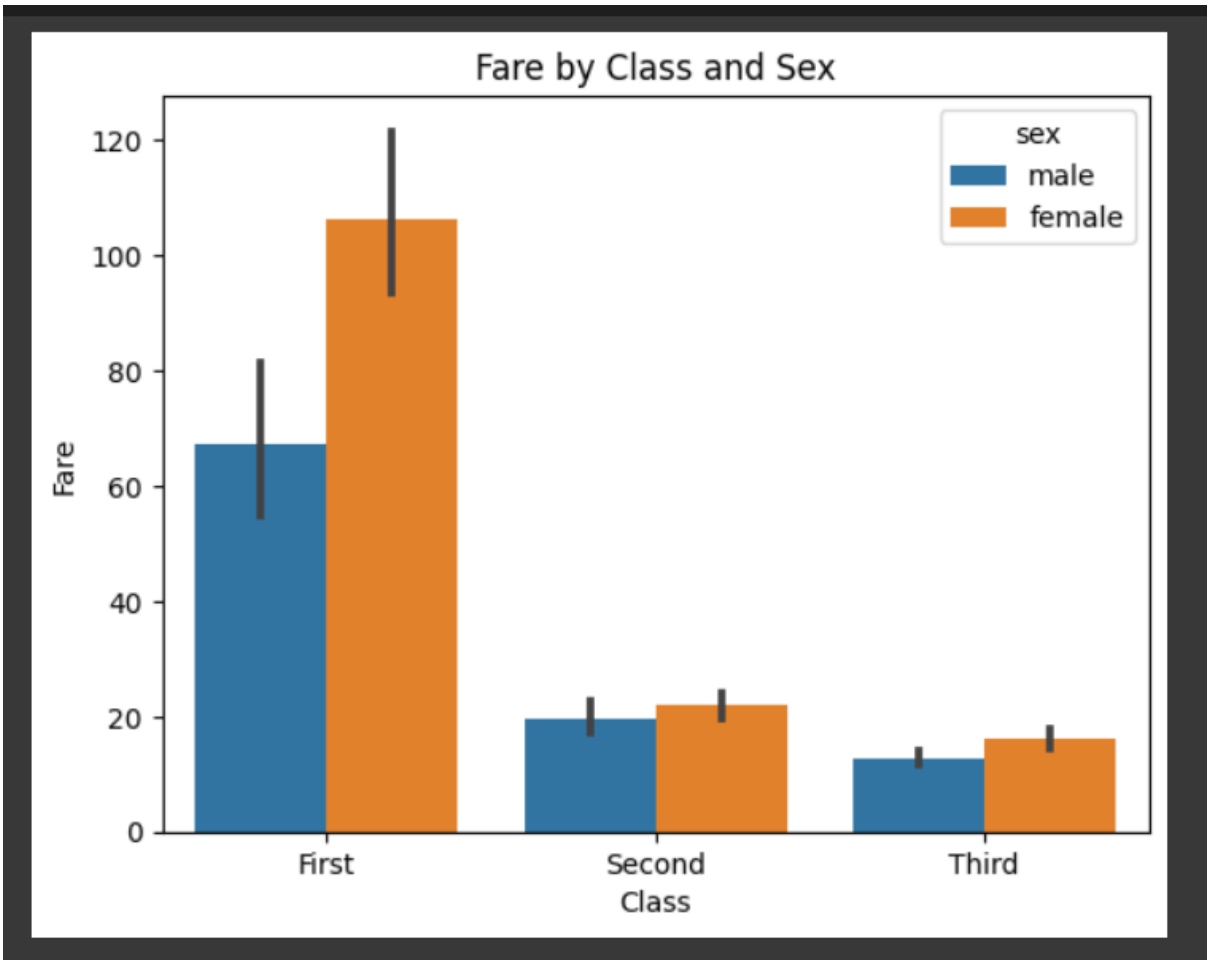
```python
# 20 types of graphs for visualization

# 1. Countplot


titanic = sns.load_dataset('titanic')

sns.countplot(x='survived', data=titanic)
plt.title('Survival Count')
plt.xlabel('Survival')
plt.ylabel('Count')
plt.show()
```



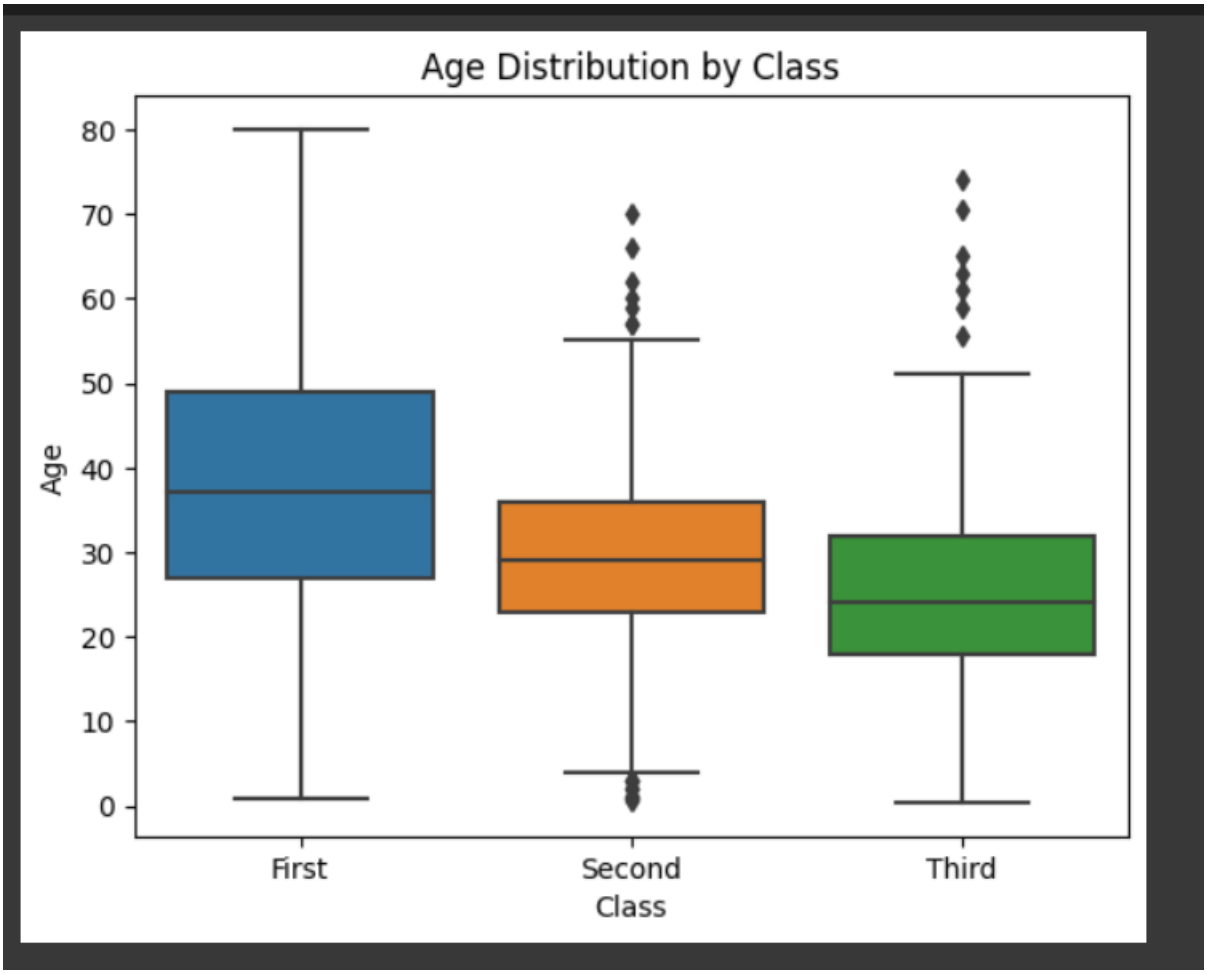
```python
# 2. Bar chart

titanic = sns.load_dataset('titanic')

sns.barplot(x='class', y='fare', data=titanic)
plt.title('Fare by Class')
plt.xlabel('Class')
plt.ylabel('Fare')
plt.show()
```



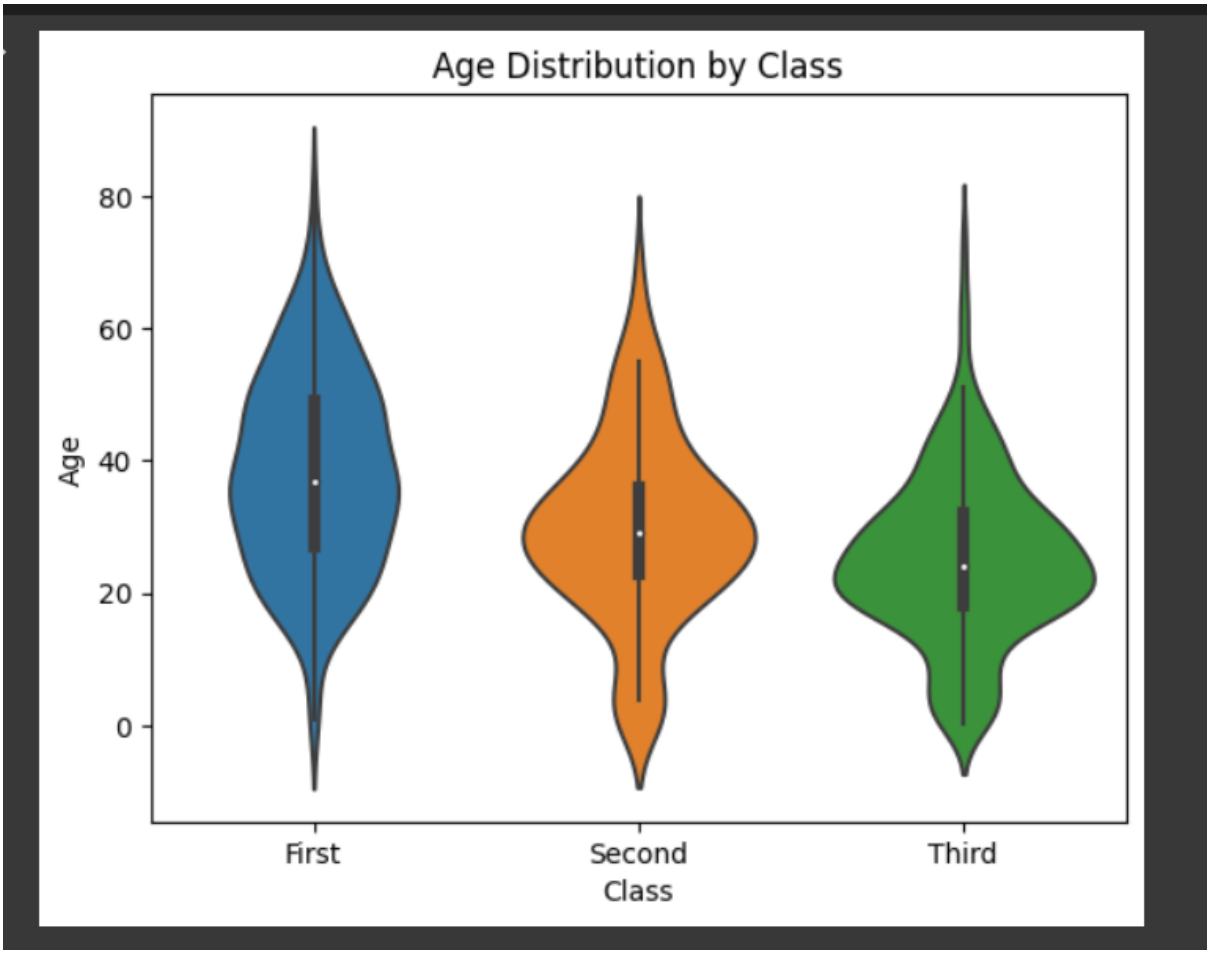
```python
# 3. Stacked bar chart

titanic = sns.load_dataset('titanic')

sns.barplot(x='class', y='fare', hue='sex', data=titanic)
plt.title('Fare by Class and Sex')
plt.xlabel('Class')
plt.ylabel('Fare')
plt.show()
```



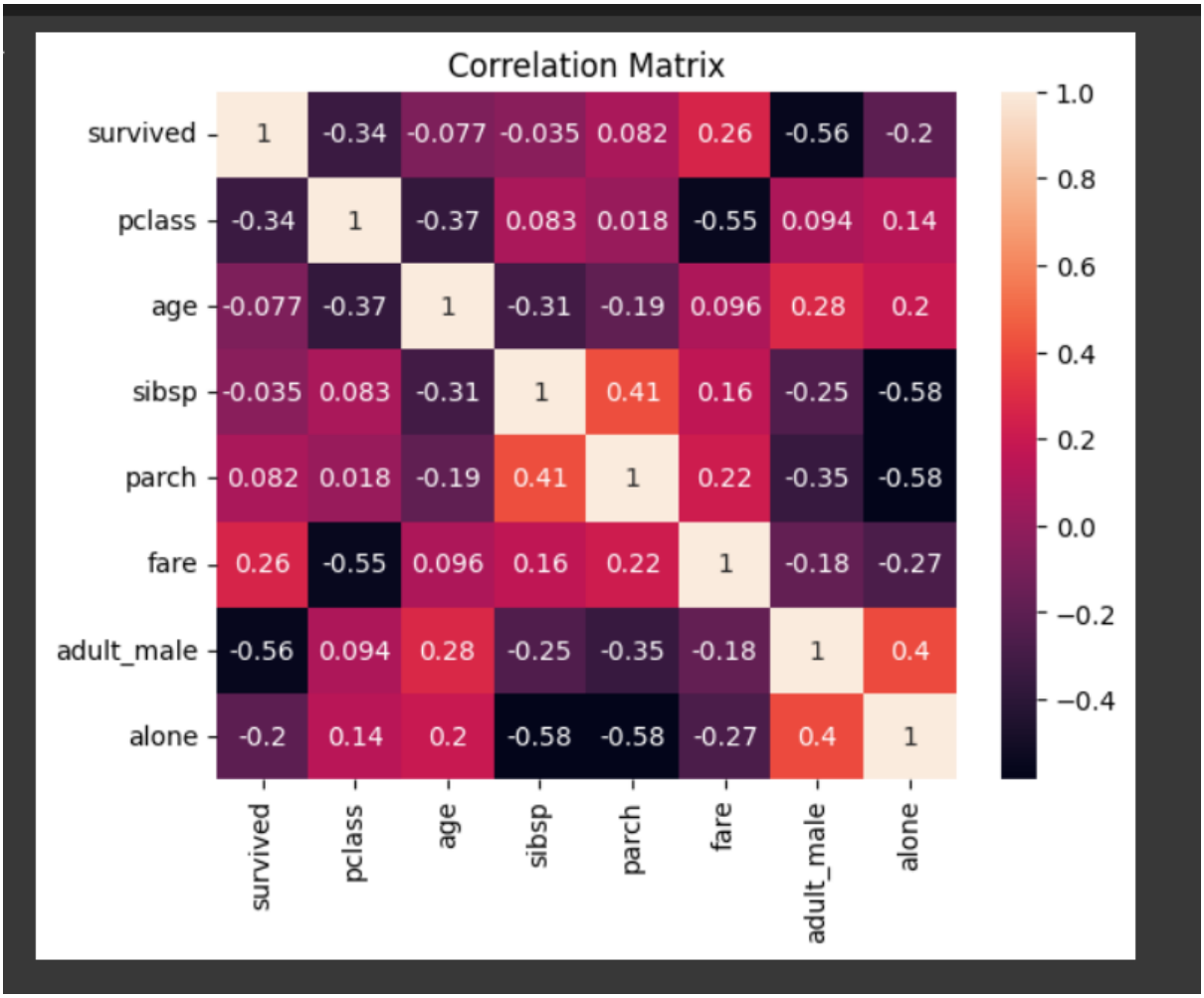
```python
# 4. Box plot

titanic = sns.load_dataset('titanic')

sns.boxplot(x='class', y='age', data=titanic)
plt.title('Age Distribution by Class')
plt.xlabel('Class')
plt.ylabel('Age')
plt.show()
```



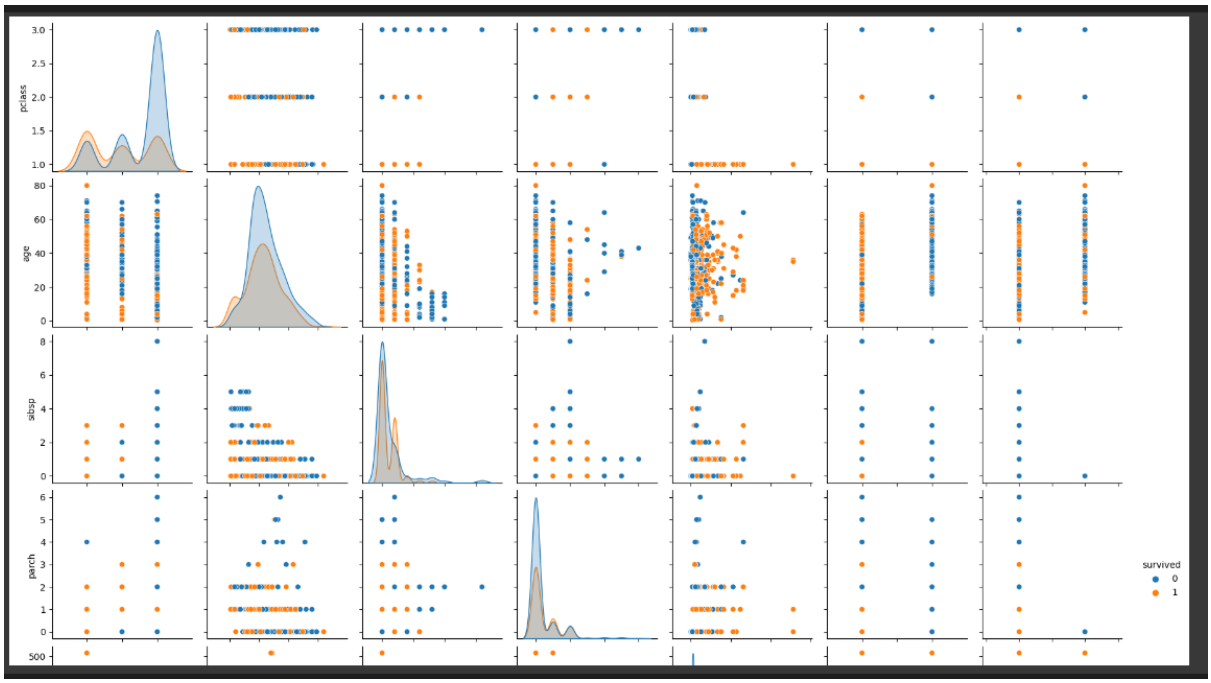
```python
# 5. Violin plot

titanic = sns.load_dataset('titanic')

sns.violinplot(x='class', y='age', data=titanic)
plt.title('Age Distribution by Class')
plt.xlabel('Class')
plt.ylabel('Age')
plt.show()
```



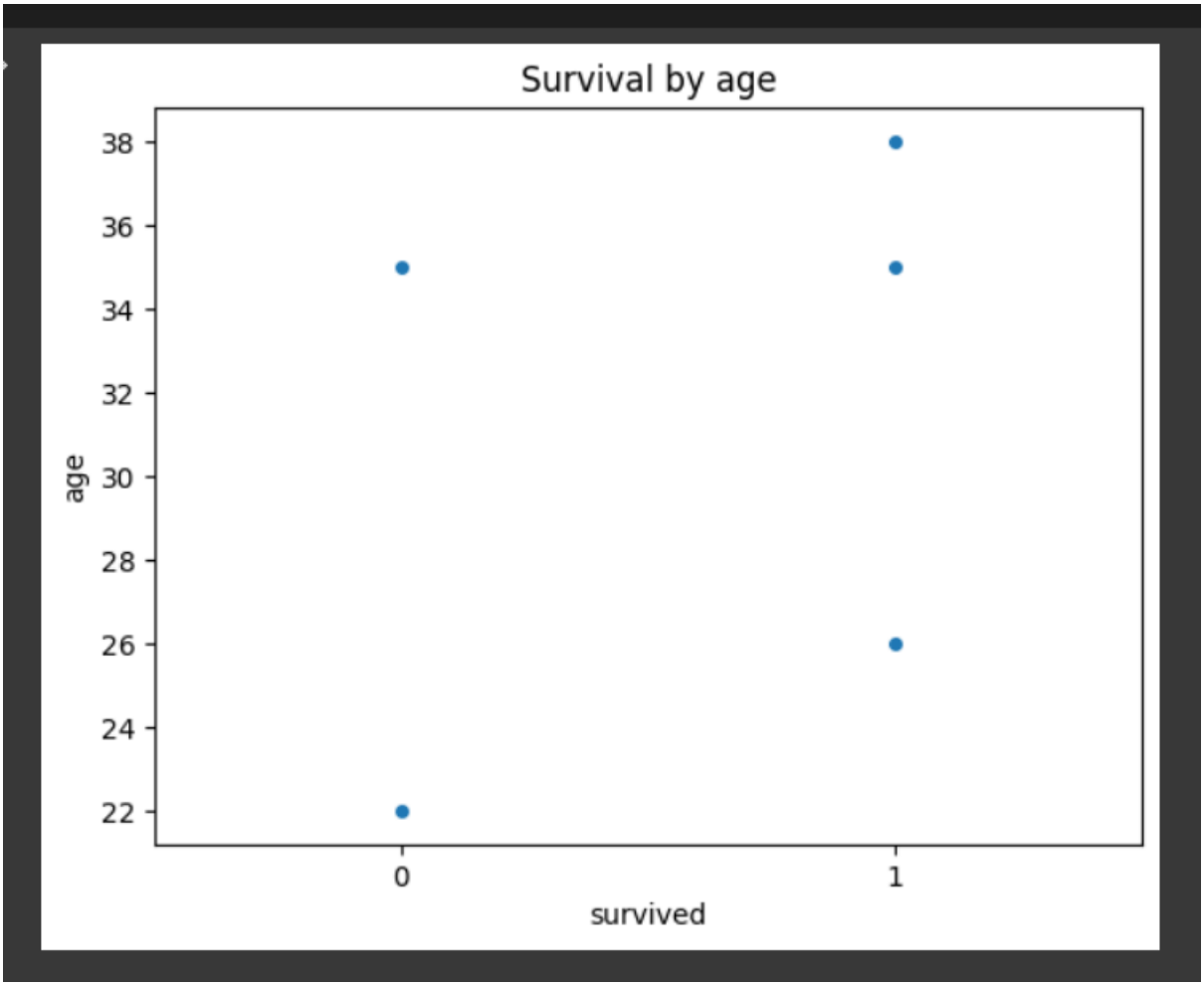
```python
# 6. Heatmap

titanic = sns.load_dataset('titanic')

corr = titanic.corr()
sns.heatmap(corr, annot=True)
plt.title('Correlation Matrix')
plt.show()
```



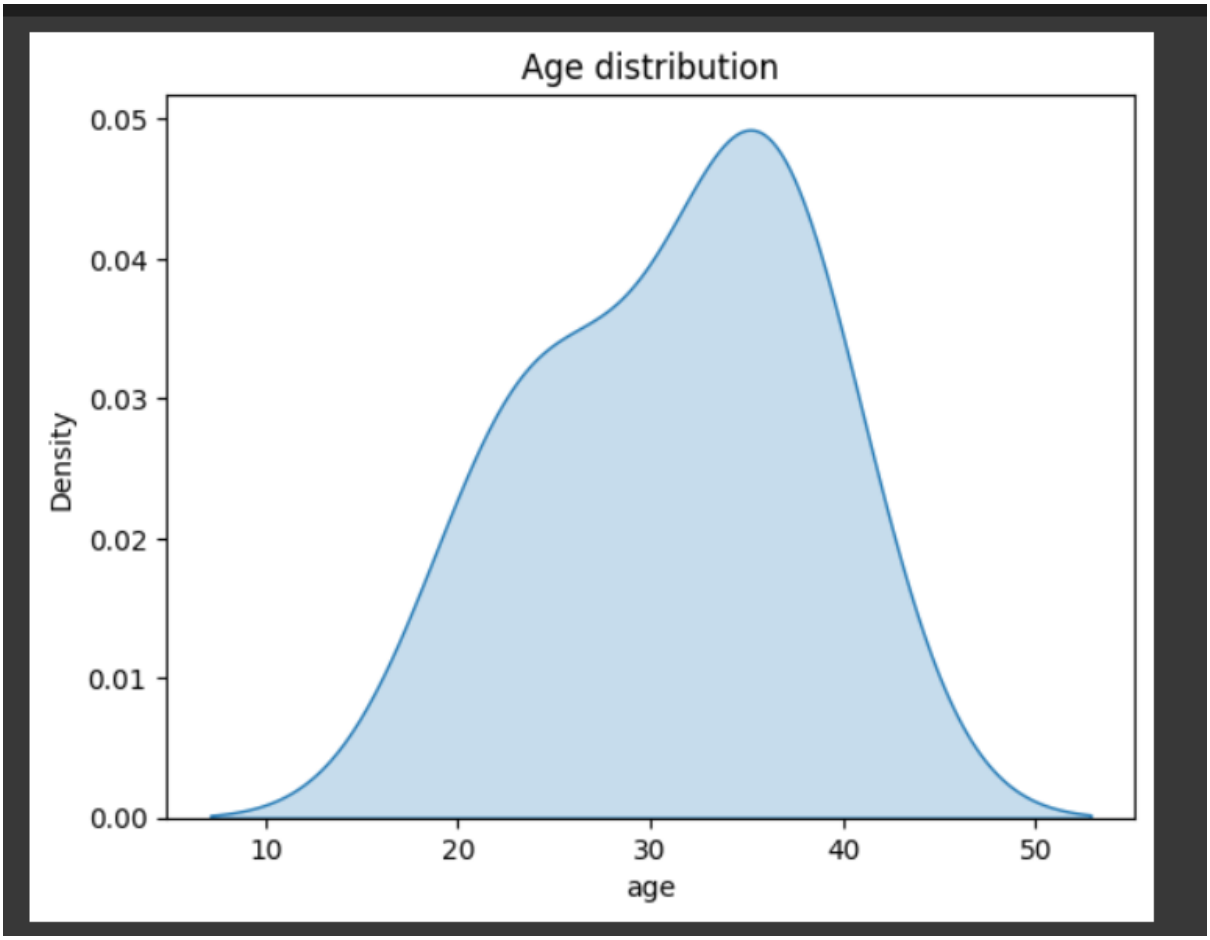
```python
# 7. Pair plot

titanic = sns.load_dataset('titanic')

sns.pairplot(titanic, hue='survived')
plt.title('Pair plot')
plt.show()
```



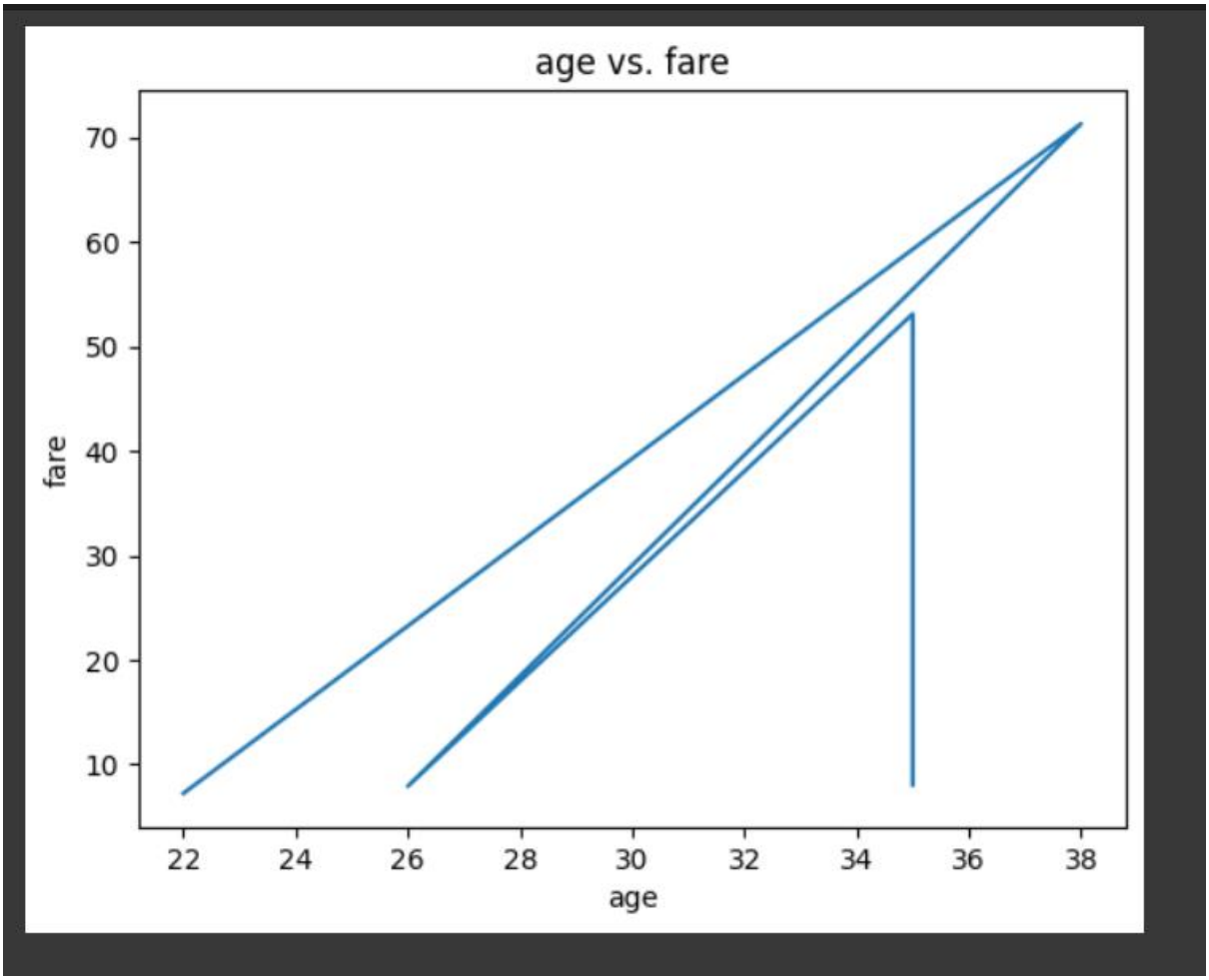
```python
# 8. Swarm plot

data = pd.read_csv('titanic.csv')

sns.swarmplot(x='survived', y='age', data=data)
plt.title('Survival by age')
plt.show()
```



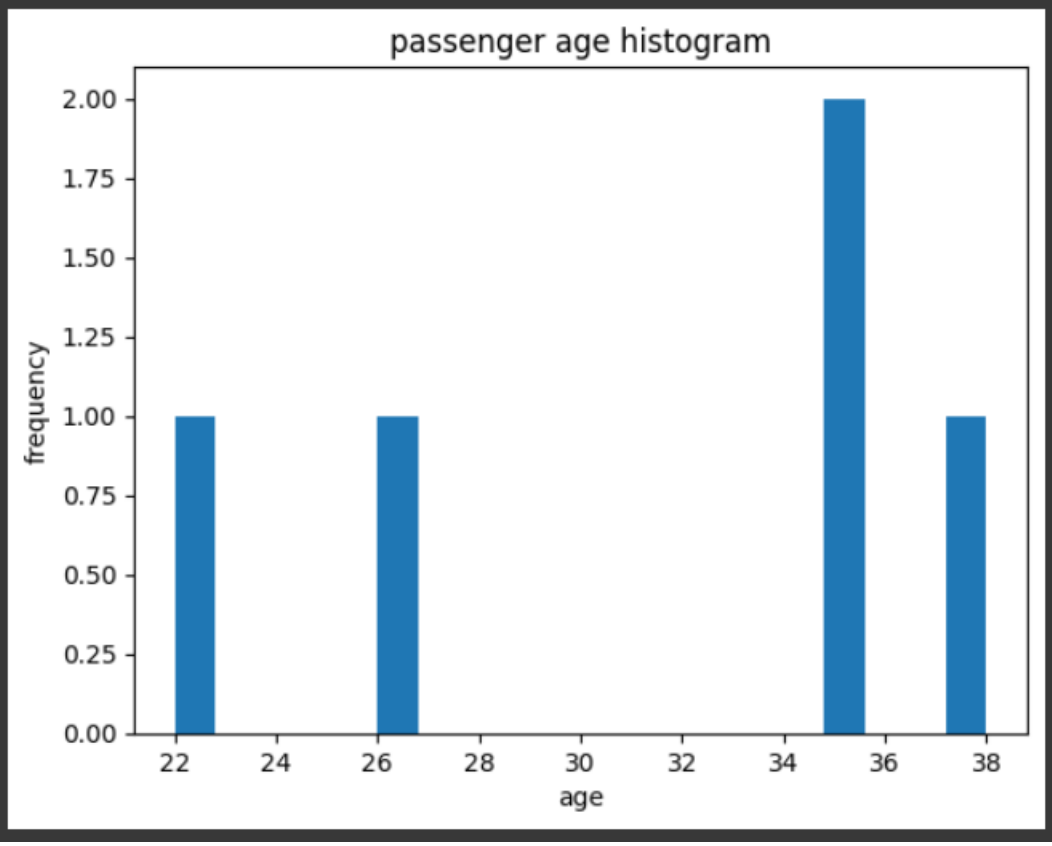
```
# 9. Density plot

data = pd.read_csv('titanic.csv')

sns.kdeplot(data['age'], shade=True)
plt.title('Age distribution')
plt.show()
```



```
# 10. Line plot

data = pd.read_csv('titanic.csv')

plt.plot(data['age'], data['fare'])
plt.title('age vs. fare')
plt.xlabel('age')
plt.ylabel('fare')
plt.show()
```

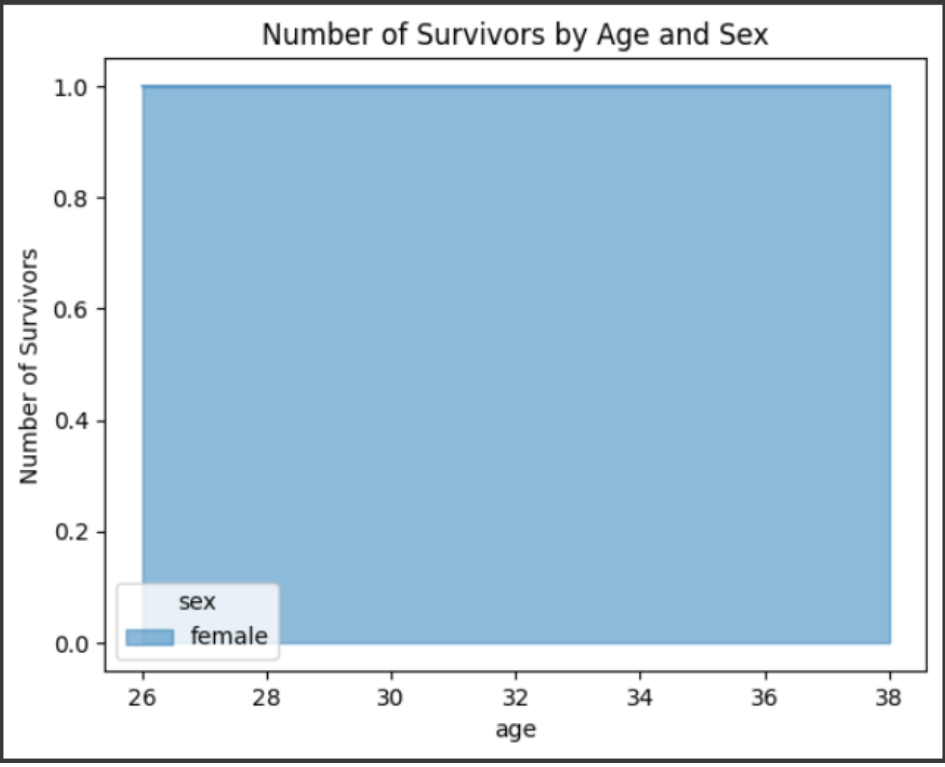


```python
# 11. Histogram

data = pd.read_csv('titanic.csv')

plt.hist(data['age'].dropna(), bins=20)
plt.title('passenger age histogram')
plt.xlabel('age')
plt.ylabel('frequency')
plt.show()
```

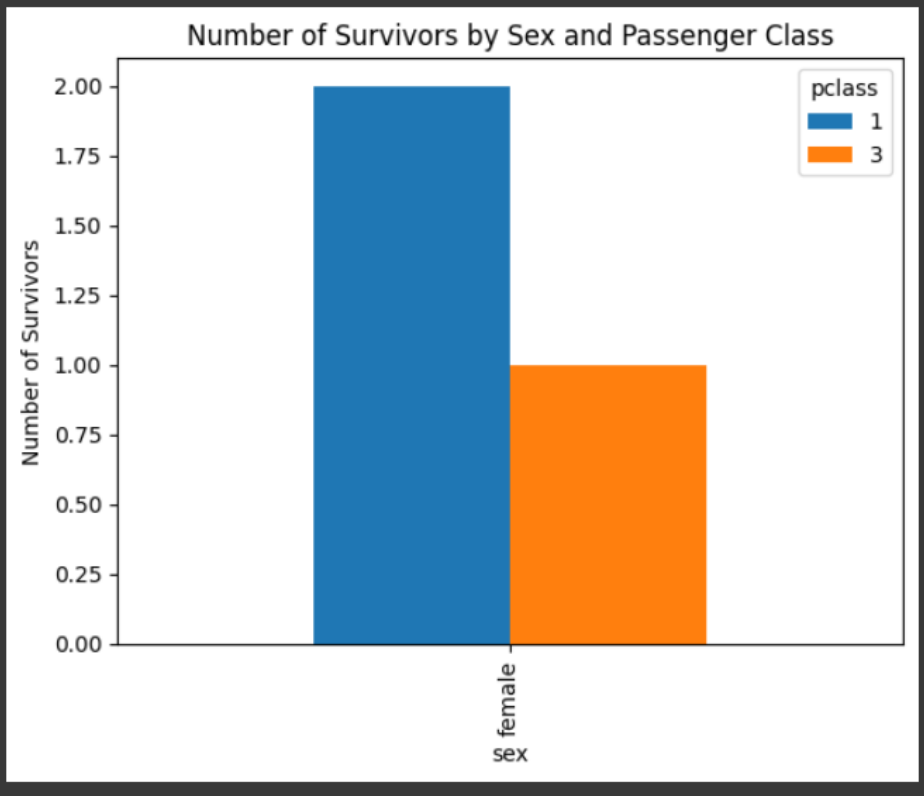


```python
# 12. Donut Chart
df = pd.read_csv('titanic.csv')

survived = df['survived'].value_counts()
fig, ax = plt.subplots()
outer_colors = ['#FF7F7F', '#7FB3D5']
inner_colors = ['#FFAFAF', '#A5C9DD']
ax.pie(survived, colors=outer_colors, wedgeprops=dict(width=0.4, edgecolor='w'))
ax.pie(survived, colors=inner_colors, wedgeprops=dict(width=0.7, edgecolor='w'))
ax.set_title('survival counts')
ax.legend(['died', 'survived'], loc='upper left')
plt.show()
```

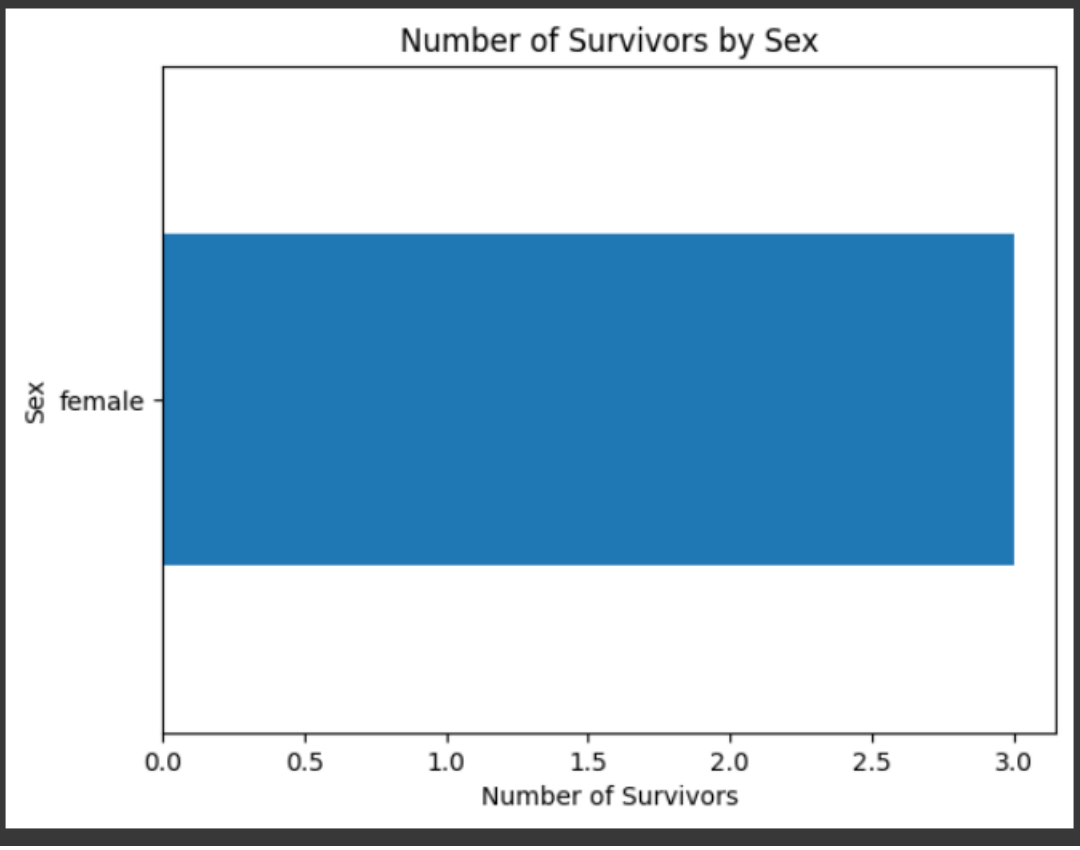
```python
# 13. Area plot
df = pd.read_csv('titanic.csv')
survivors = df[df['survived'] == 1].groupby(['age', 'sex'])['survived'].count().unstack()
fig, ax = plt.subplots()
survivors.plot(kind='area', stacked=True, alpha=0.5, ax=ax)
ax.set_xlabel('age')
ax.set_ylabel('Number of Survivors')
ax.set_title('Number of Survivors by Age and Sex')
plt.show()
```



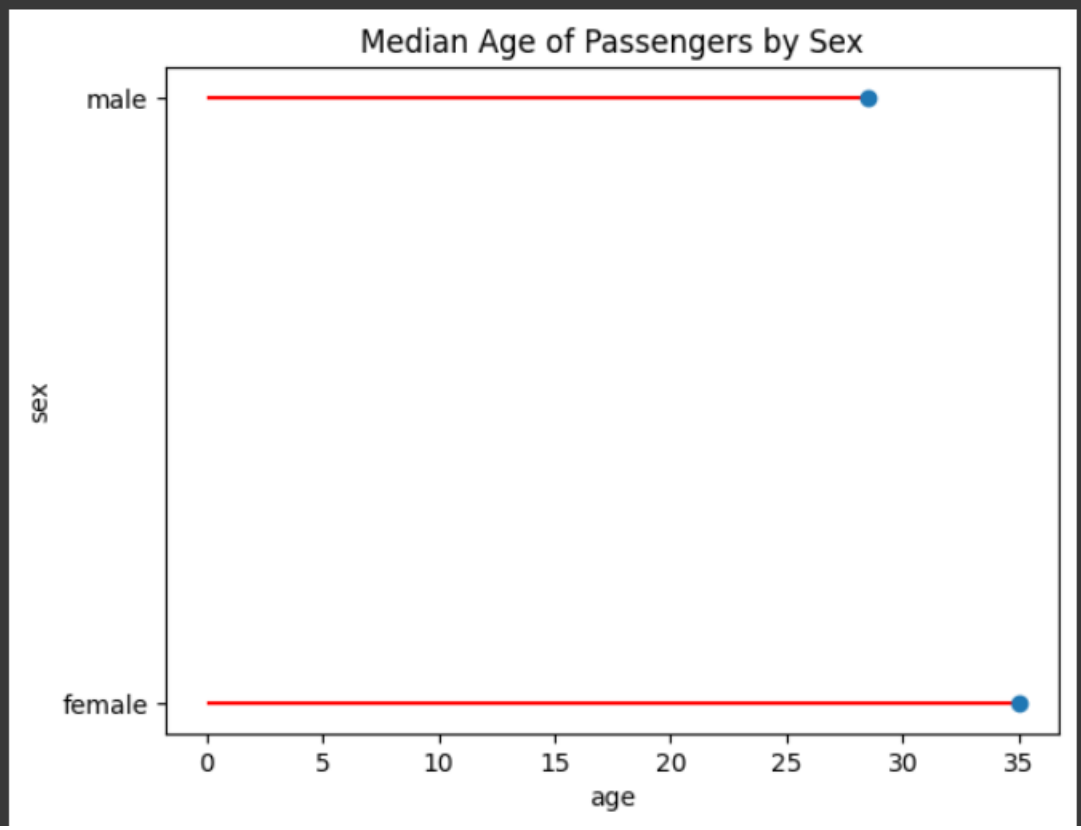
```python
# 14. Grouped Bar Chart
df = pd.read_csv('titanic.csv')
survivors = df[df['survived'] == 1].groupby(['sex', 'pclass'])['survived'].count().unstack()
fig, ax = plt.subplots()
survivors.plot(kind='bar', ax=ax)
ax.set_xlabel('sex')
ax.set_ylabel('Number of Survivors')
ax.set_title('Number of Survivors by Sex and Passenger Class')
plt.show()
```



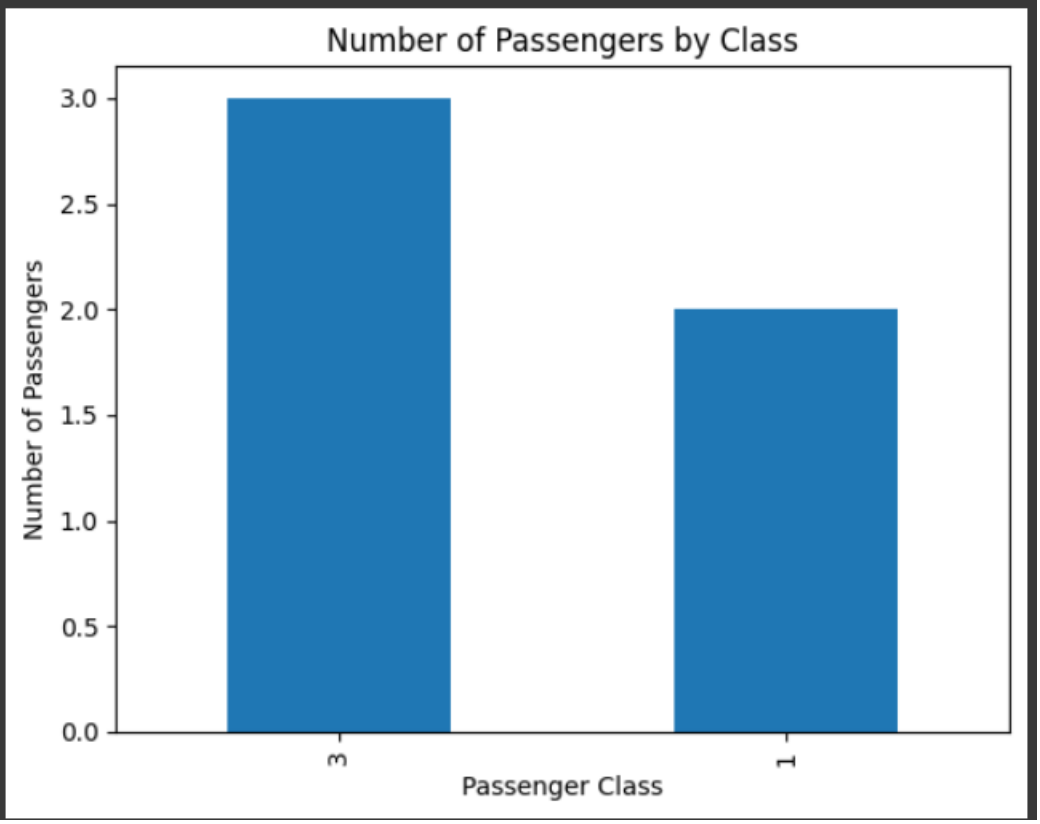
```python
# 15. Horizontal bar chart
df = pd.read_csv('titanic.csv')
survivors = df[df['survived'] == 1]['sex'].value_counts()
fig, ax = plt.subplots()
survivors.plot(kind='barh', ax=ax)
ax.set_xlabel('Number of Survivors')
ax.set_ylabel('Sex')
ax.set_title('Number of Survivors by Sex')
plt.show()
```



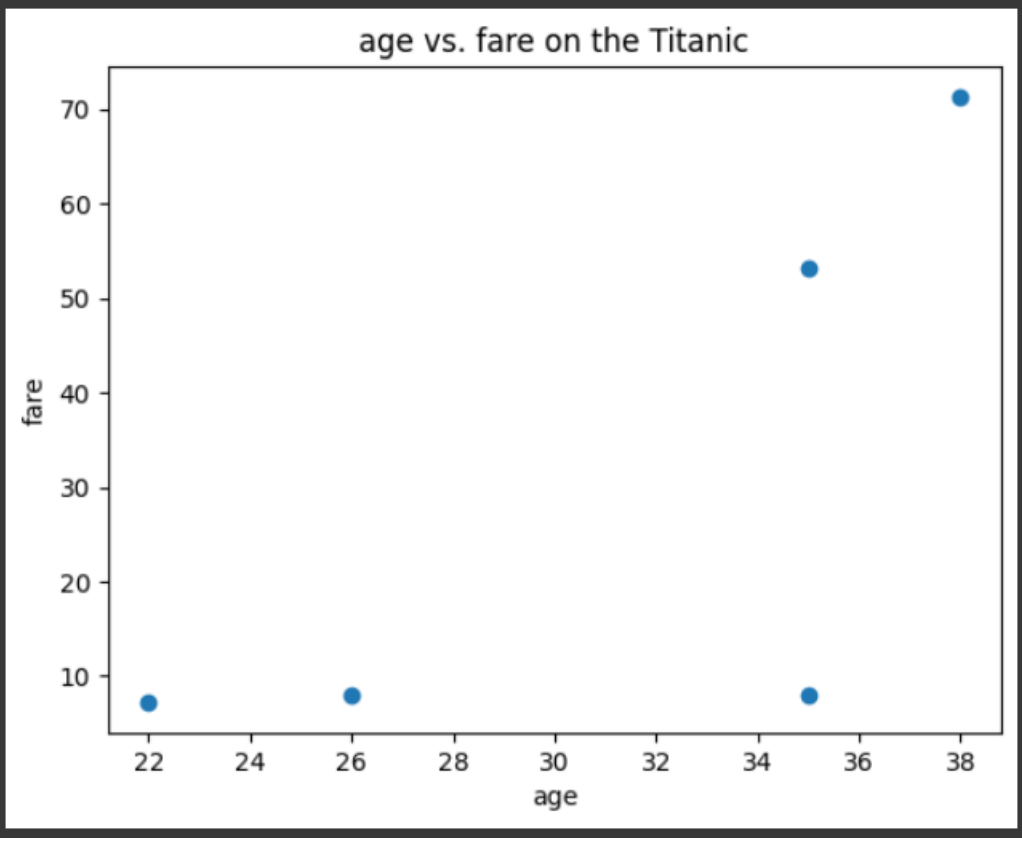
```python
# 16. Lollipop plot
df = pd.read_csv('titanic.csv')
median_age = df.groupby('sex')['age'].median().sort_values(ascending=False)
fig, ax = plt.subplots()
ax.hlines(y=median_age.index, xmin=0, xmax=median_age, color='red')
ax.plot(median_age, median_age.index, "o")
ax.set_xlabel('age')
ax.set_ylabel('sex')
ax.set_title('Median Age of Passengers by Sex')
plt.show()
```



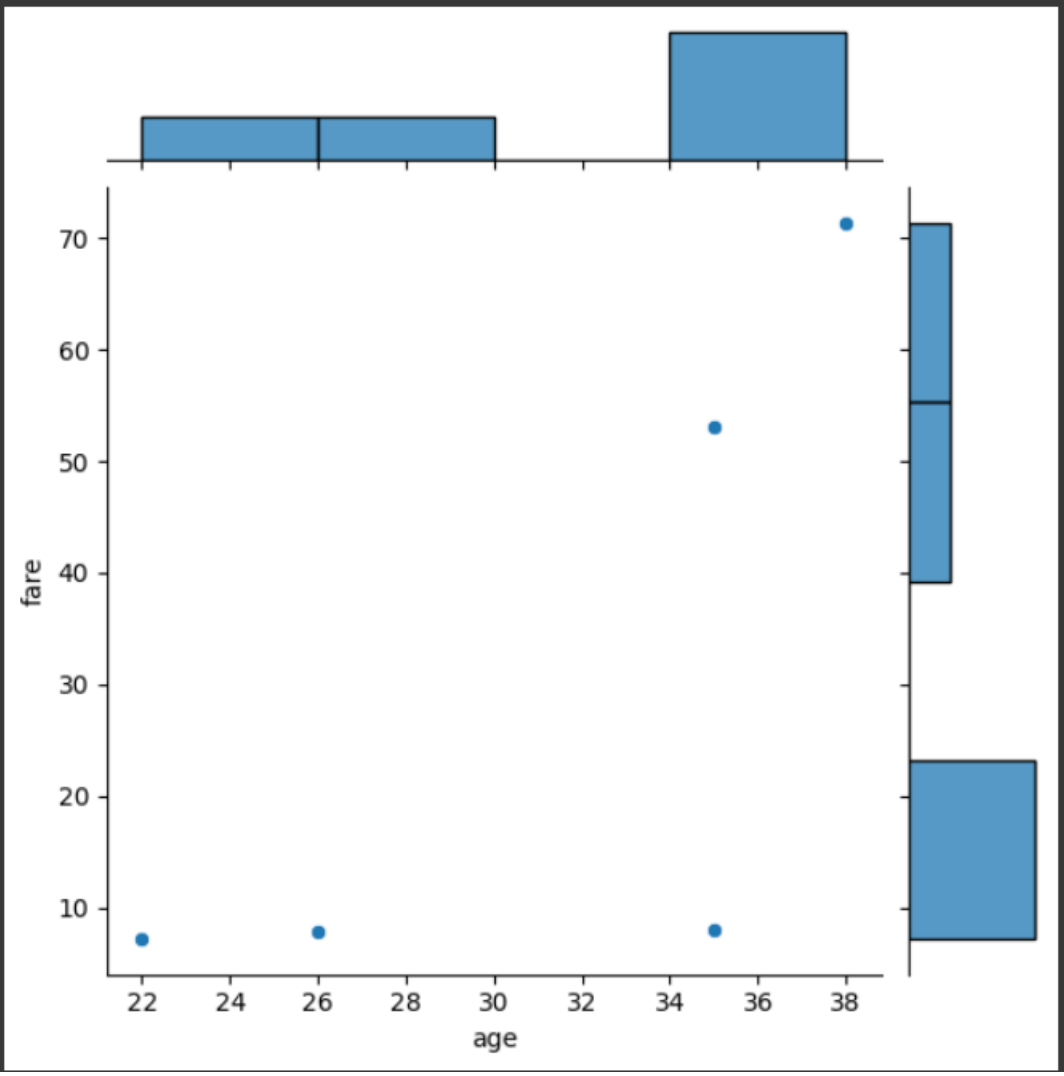
```python
# 17. Column Chart
df = pd.read_csv('titanic.csv')
passenger_count = df['pclass'].value_counts()
fig, ax = plt.subplots()
passenger_count.plot(kind='bar', ax=ax)
ax.set_xlabel('Passenger Class')
ax.set_ylabel('Number of Passengers')
ax.set_title('Number of Passengers by Class')
plt.show()
```



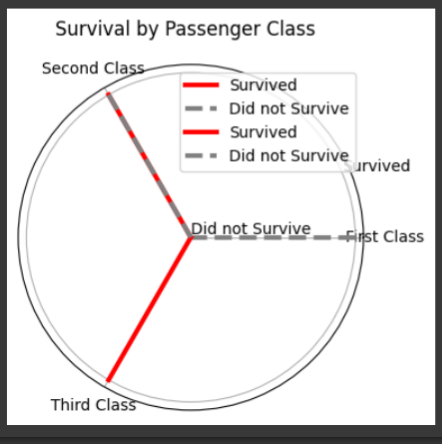
```python
# 18. Scatter plot

df = pd.read_csv('titanic.csv')

plt.scatter(df['age'], df['fare'])

plt.xlabel('age')
plt.ylabel('fare')
plt.title('age vs. fare on the Titanic')

plt.show()
```



```python
# 19. joint plot
df = pd.read_csv('titanic.csv')
sns.jointplot(x='age', y='fare', data=df)
plt.show()
```



```python
# 20. Polar plot
df = pd.read_csv('titanic.csv')
class_survival = df.groupby(['pclass', 'survived']).size().unstack()
class_survival.plot(kind='line', subplots=True, layout=(2,1), sharex=False, figsize=(4, 4),
                    title='Survival by Passenger Class', style=['--', '-'])
plt.subplot(111, polar=True)
plt.xticks([i * 2 * np.pi / 3 for i in range(3)], ['First Class', 'Second Class', 'Third Class'])
plt.yticks([0, 1], ['Did not Survive', 'Survived'])
for i in range(2):
    plt.plot([i * 2 * np.pi / 3, (i + 1) * 2 * np.pi / 3], [0, 1], linewidth=3, color='red', label='Survived')
    plt.plot([i * 2 * np.pi / 3, (i + 1) * 2 * np.pi / 3], [1, 0], linewidth=3, color='grey', linestyle='--', label='Did not Survive')
plt.legend(loc='upper right')
plt.show()
```



```python
# 21. Bubble plot

df = pd.read_csv('titanic.csv')

df = df.dropna()

x = df['age']
y = df['fare']
z = df['survived']

colors = ['red', 'green']

labels = ['Did not survive', 'survived']

fig, ax = plt.subplots()
ax.scatter(x, y, s=z*100, c=colors, alpha=0.5)

ax.set_xlabel('age')
ax.set_ylabel('fare')
ax.set_title('Bubble Plot of Titanic Passengers')

legend_elements = [plt.scatter([],[], s=100, c='red', alpha=0.8, label='Did not survive'),
                   plt.scatter([],[], s=100, c='green', alpha=0.8, label='survived')]
ax.legend(handles=legend_elements)

plt.show()
```