```python
In [2]:  import seaborn as sns
         import pandas as pd

         # Load the Titanic dataset
         df = sns.load_dataset('titanic')

         # Display the first few rows
         print(df.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
```

```python
In [8]:  # Check structure and missing values
         df.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
```

```python
In [10]:  # Statistical summary of numerical columns
          df.describe()
```

Out[10]:

|	survived	pclass	age	sibsp	parch	fare
count	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [12]:
```python
sns.countplot(x='survived', data=df)
plt.title('Survival Count')
plt.xlabel('Survived (0 = No, 1 = Yes)')
plt.ylabel('Count')
plt.show()
```
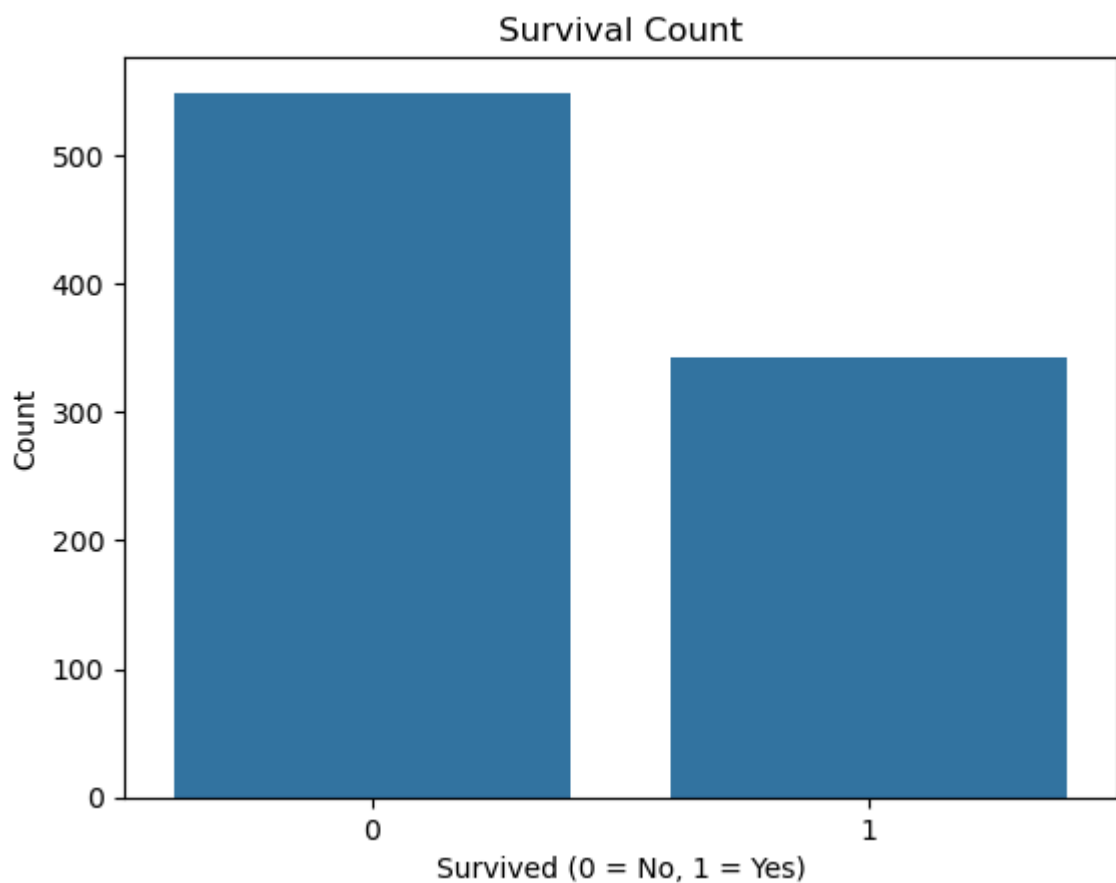

In [14]:
```python
sns.countplot(x='sex', hue='survived', data=df)
plt.title('Survival by Sex')
plt.xlabel('Sex')
plt.ylabel('Count')
plt.legend(title='Survived')
plt.show()
```

Survival by Sex
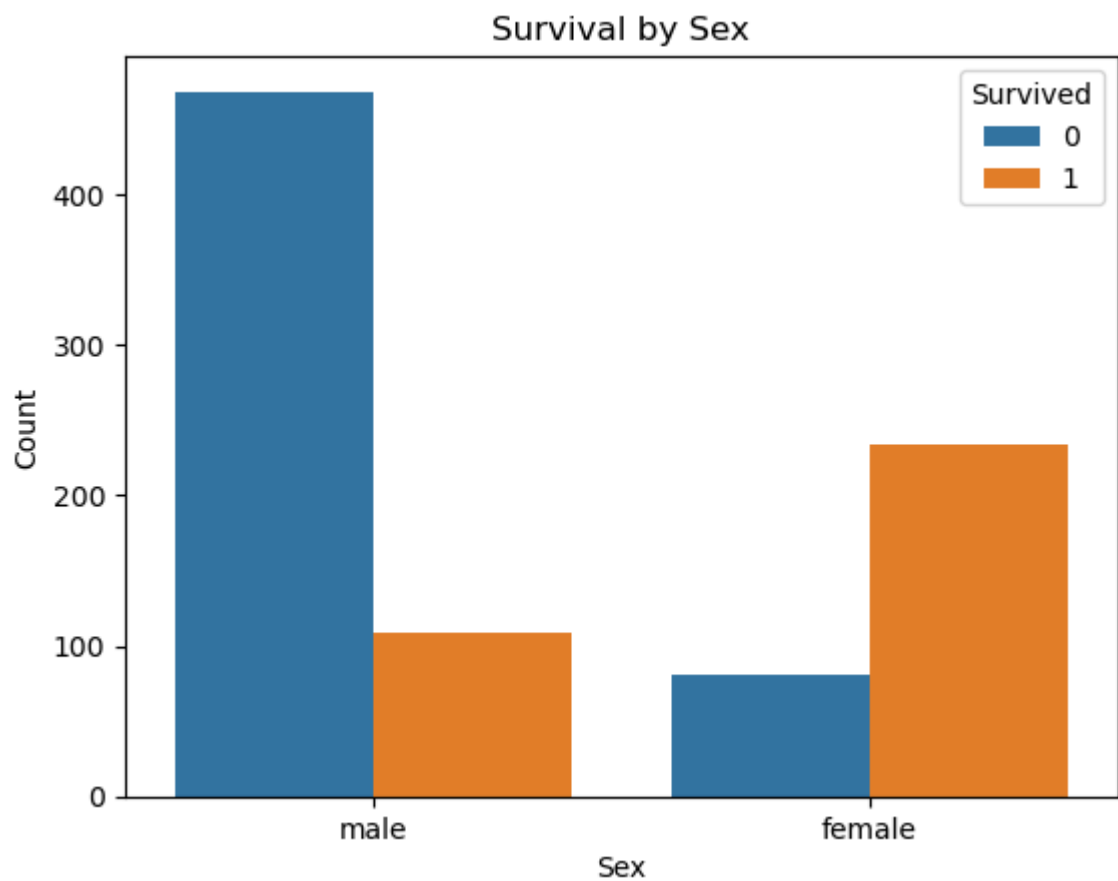

```python
In [16]:  sns.histplot(df['age'].dropna(), bins=30, kde=True)
          plt.title('Age Distribution')
          plt.xlabel('Age')
          plt.ylabel('Frequency')
          plt.show()
```
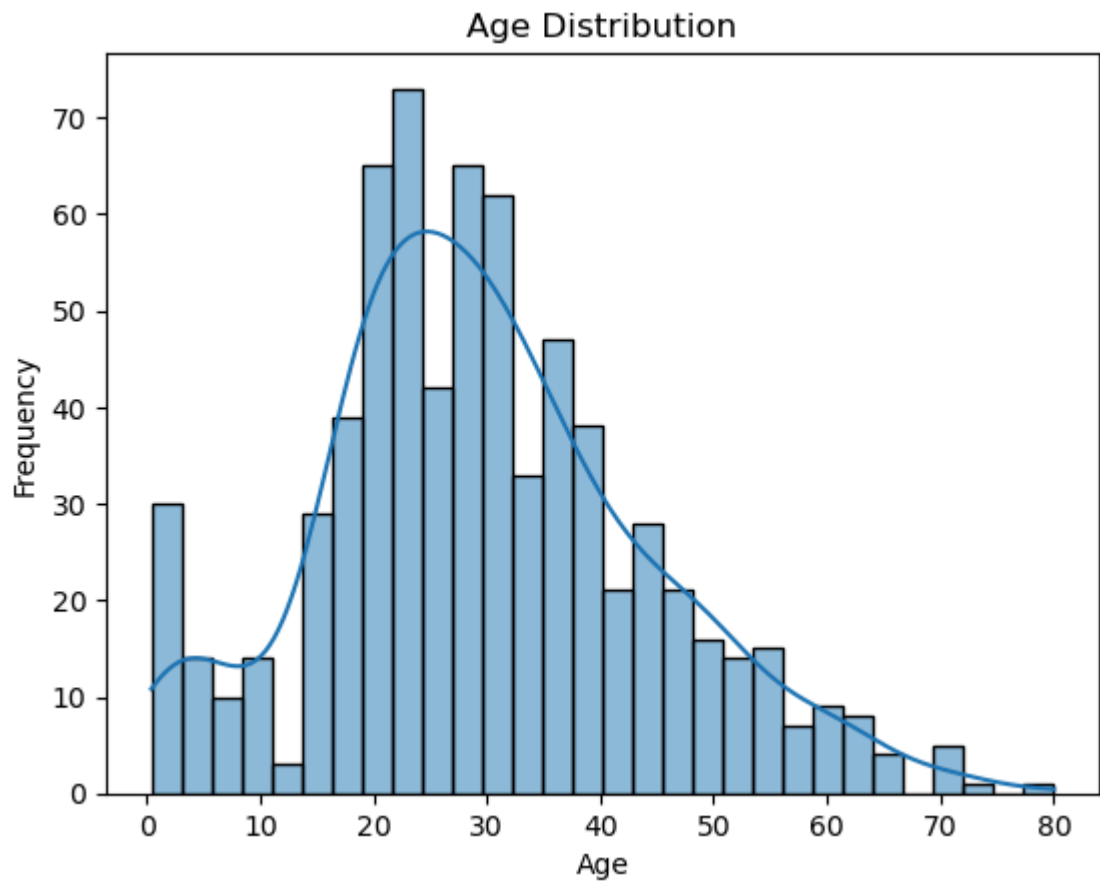
## Age Distribution


```
In [18]:  sns.boxplot(x='pclass', y='age', data=df)
          plt.title('Age Distribution by Passenger Class')
          plt.xlabel('Passenger Class')
          plt.ylabel('Age')
          plt.show()
```

Age Distribution by Passenger Class
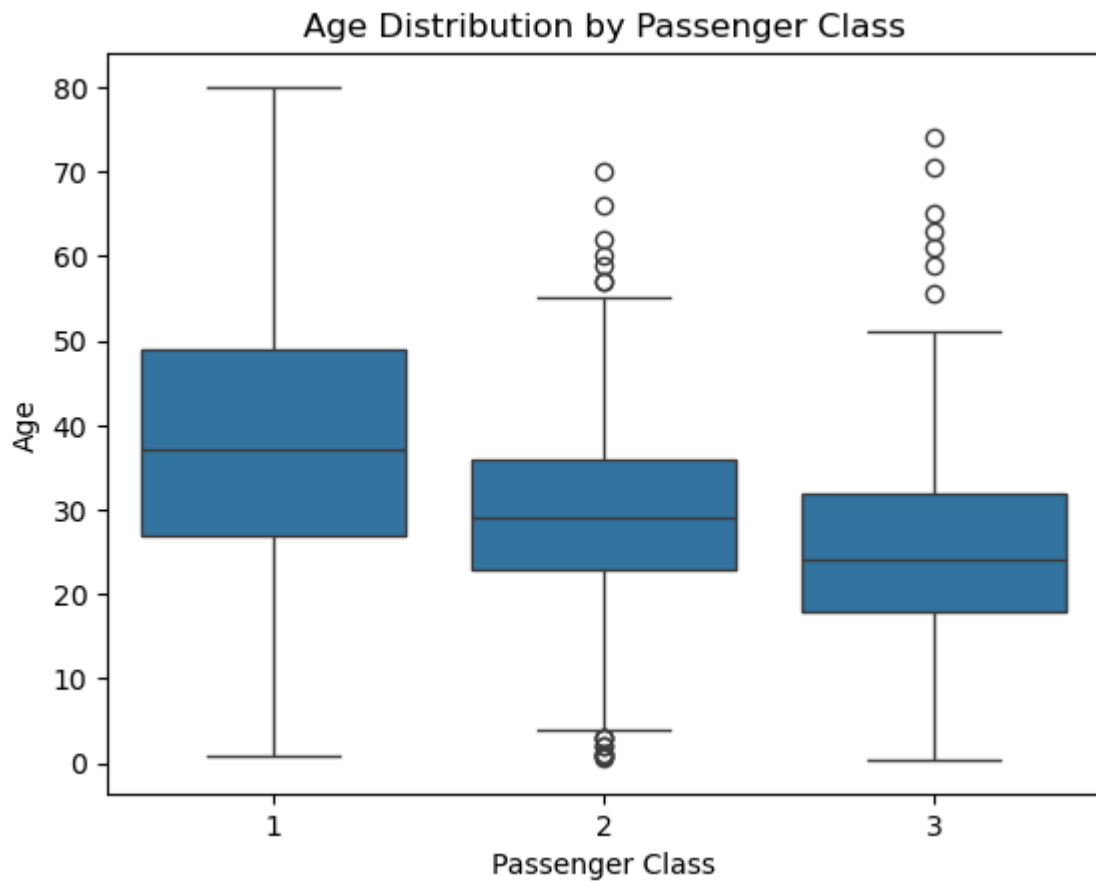
In [20]:
```python
# Select numerical columns
numeric_df = df.select_dtypes(include='number')

# Correlation matrix
plt.figure(figsize=(10, 6))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

## Correlation Heatmap
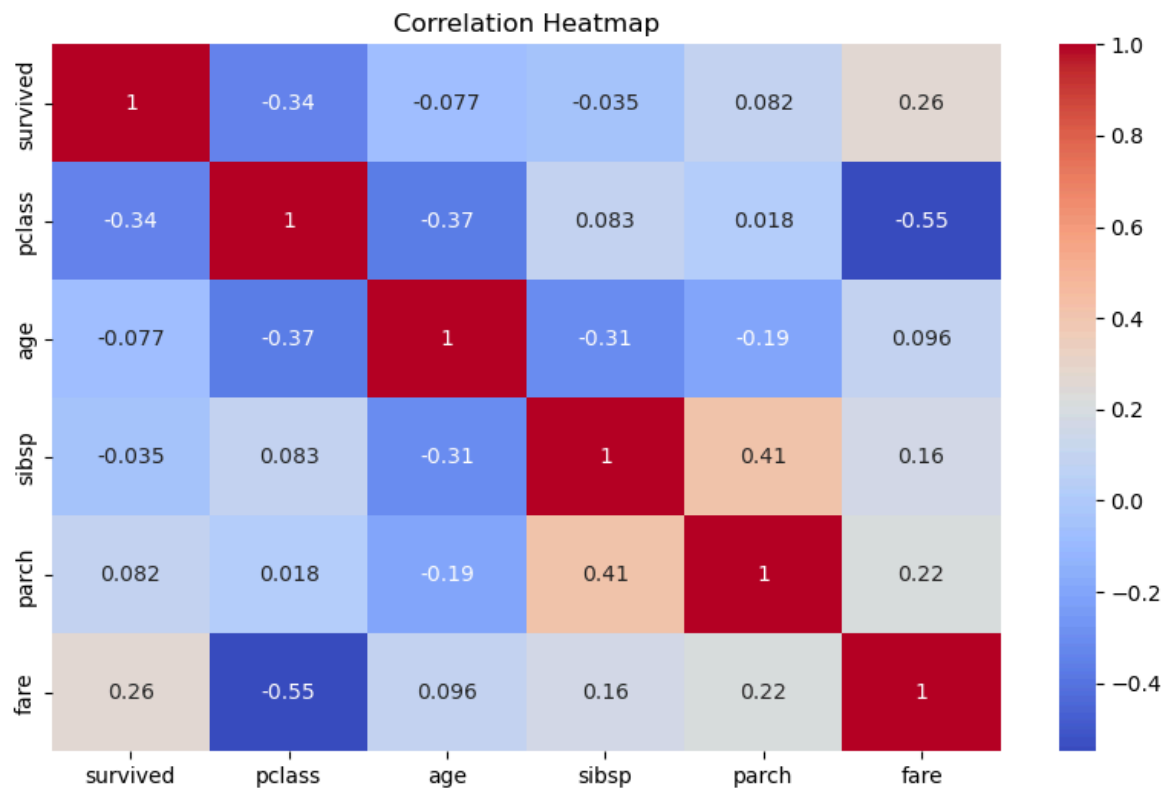


```
In [22]:   # We'll create a new DataFrame with selected columns
           pairplot_data = df[['age', 'fare', 'pclass', 'survived', 'sex']]
```

```
In [24]:   # Convert 'sex' to numeric (optional, but helps with coloring)
           pairplot_data['sex'] = pairplot_data['sex'].map({'male': 0, 'female': 1})
```

```
C:\Users\sande\AppData\Local\Temp\ipykernel_12600\1987081823.py:2: SettingWithCop
yWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  pairplot_data['sex'] = pairplot_data['sex'].map({'male': 0, 'female': 1})
```
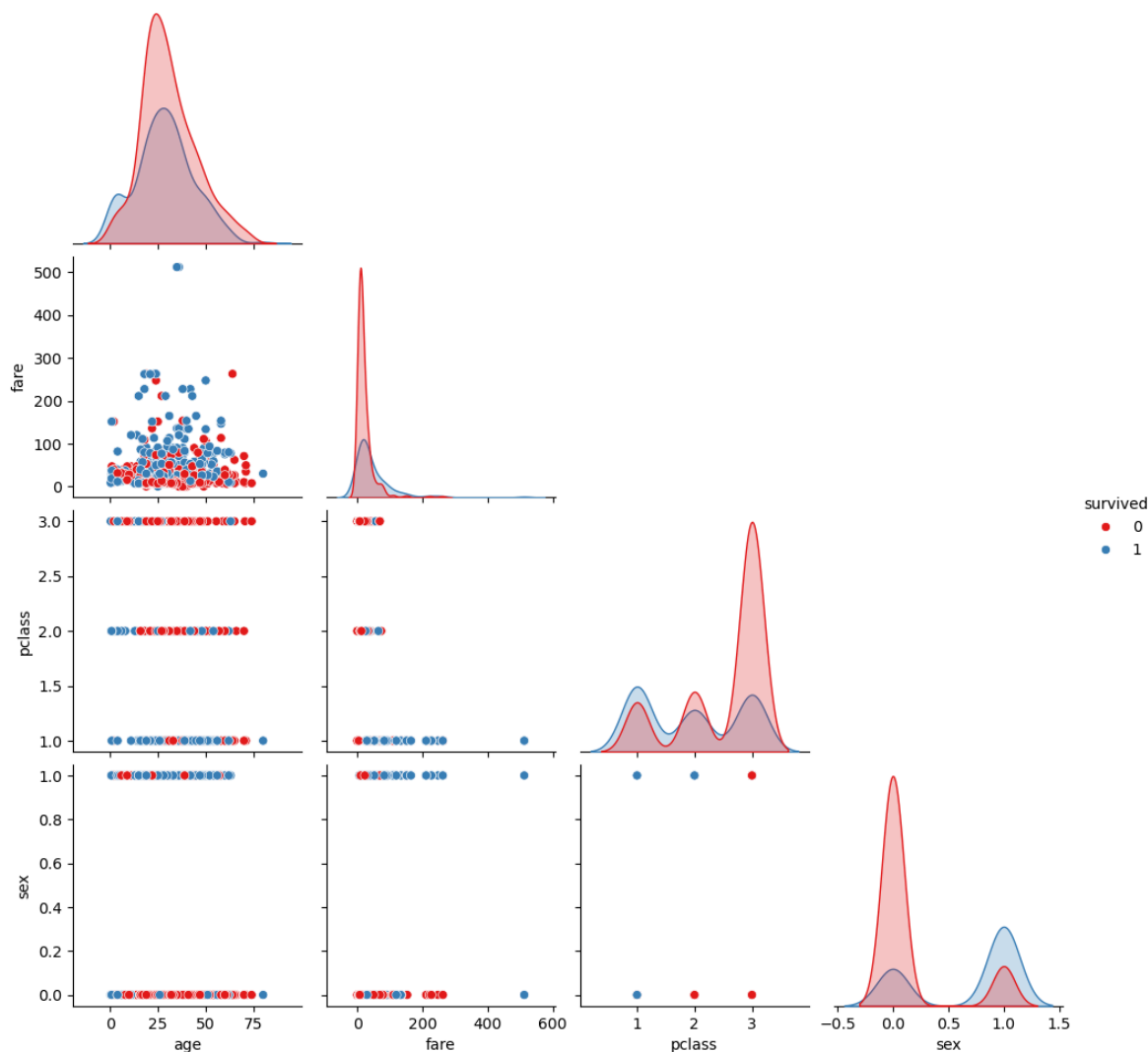
```
In [30]:   sns.pairplot(pairplot_data, hue='sex', palette='coolwarm', diag_kind='kde')
           plt.suptitle("Pairplot of Titanic Features", y=1.02)
           plt.show()
```

Pairplot of Titanic Features
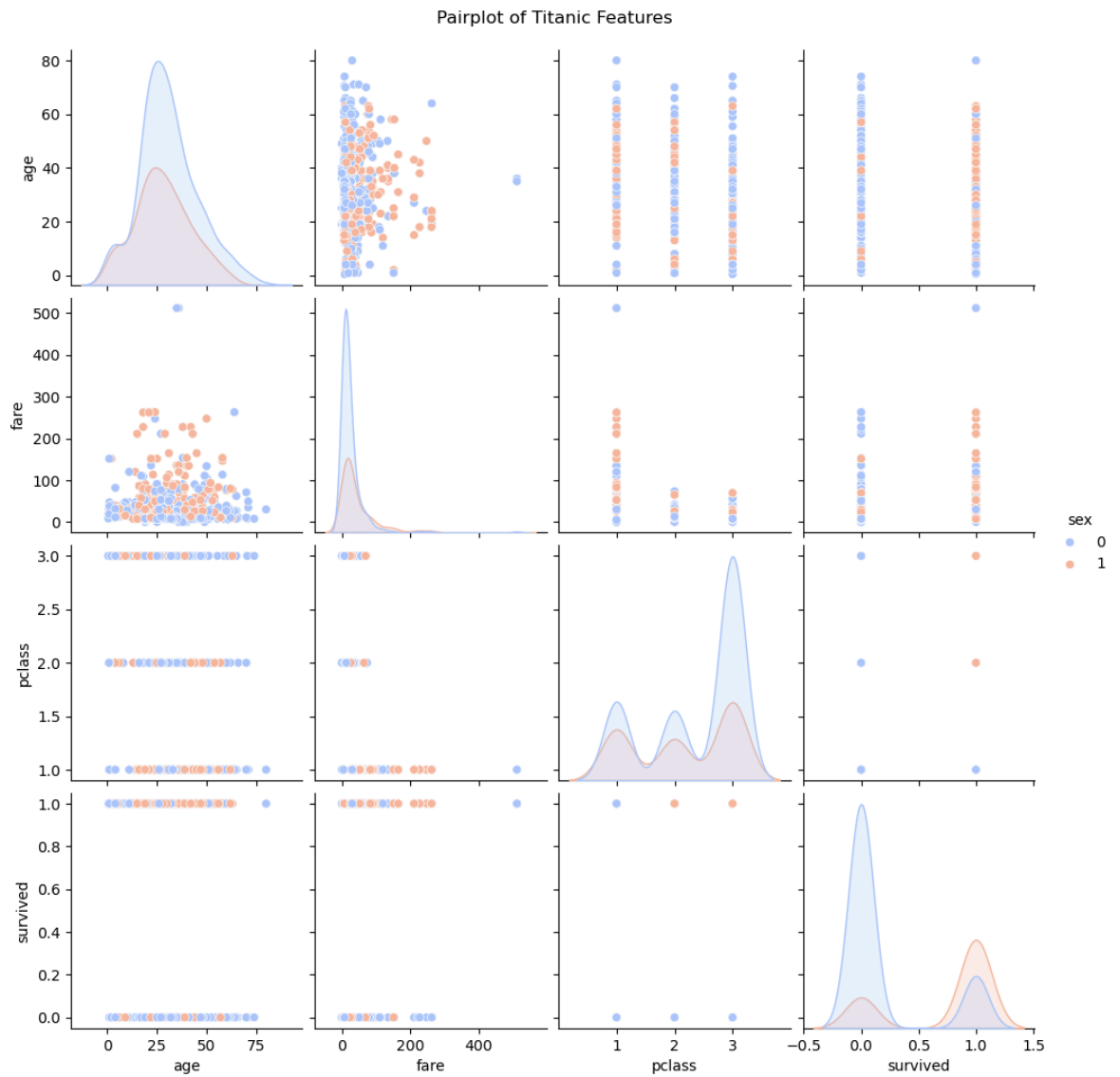

```
In [28]: sns.pairplot(pairplot_data, hue='survived', palette='Set1', diag_kind='kde', cor
```

Out[28]: <seaborn.axisgrid.PairGrid at 0x1e9f73e9550>

In [1]: `pip install reportlab`

```
Collecting reportlab
  Downloading reportlab-4.3.1-py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: pillow>=9.0.0 in c:\users\sande\anaconda3\lib\site
-packages (from reportlab) (10.4.0)
Requirement already satisfied: chardet in c:\users\sande\anaconda3\lib\site-packa
ges (from reportlab) (4.0.0)
Downloading reportlab-4.3.1-py3-none-any.whl (1.9 MB)
   ---------------------------------------- 0.0/1.9 MB ? eta -:--:--
   ---------------------------------------- 1.9/1.9 MB 10.7 MB/s eta 0:00:00
Installing collected packages: reportlab
Successfully installed reportlab-4.3.1
Note: you may need to restart the kernel to use updated packages.
```

In [ ]: