Task 5: Exploratory Data Analysis (EDA)

In [4]:
```
import pandas as pd
```

In [ ]:
```
# Read the dataset
```

In [10]:
```
train = pd.read_csv(r"C:\Users\Shabi\Downloads\train.csv")
```

In [11]:
```
train.head()
```

Out[11]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

In [19]:
```
test = pd.read_csv("C:\\Users\\Shabi\\Downloads\\titanic\\test.csv")
```

In [20]:
```
test.head()
```

Out[20]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

In [50]:
```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          891 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Embarked     891 non-null    object 
dtypes: float64(2), int64(5), object(4)
memory usage: 76.7+ KB
```

In [22]:
```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Pclass       418 non-null    int64  
 2   Name         418 non-null    object 
 3   Sex          418 non-null    object 
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64  
 6   Parch        418 non-null    int64  
 7   Ticket       418 non-null    object 
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object 
 10  Embarked     418 non-null    object 
dtypes: float64(2), int64(4), object(5)
memory usage: 36.1+ KB
```

In [23]:
```
train.describe()
```

Out[23]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

In [24]:
```
test.describe()
```

Out[24]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	3.000000	76.000000	8.000000	9.000000	512.329200

In [47]:
```
# Statistical counts
print("\nSurvival counts:")
print(train['Survived'].value_counts())
```

```
Survival counts:
Survived
0    549
1    342
Name: count, dtype: int64
```

In [45]:
```
print("\nGender counts:")
print(train['Sex'].value_counts())
```

```
Gender counts:
Sex
male      577
female    314
Name: count, dtype: int64
```

In [46]:
```
print("\nPassenger Class counts:")
print(train['Pclass'].value_counts())
```

```
Passenger Class counts:
Pclass
3    491
1    216
2    184
Name: count, dtype: int64
```

In [48]:
```
#cleaning the data

train.isnull().sum()
```

Out[48]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

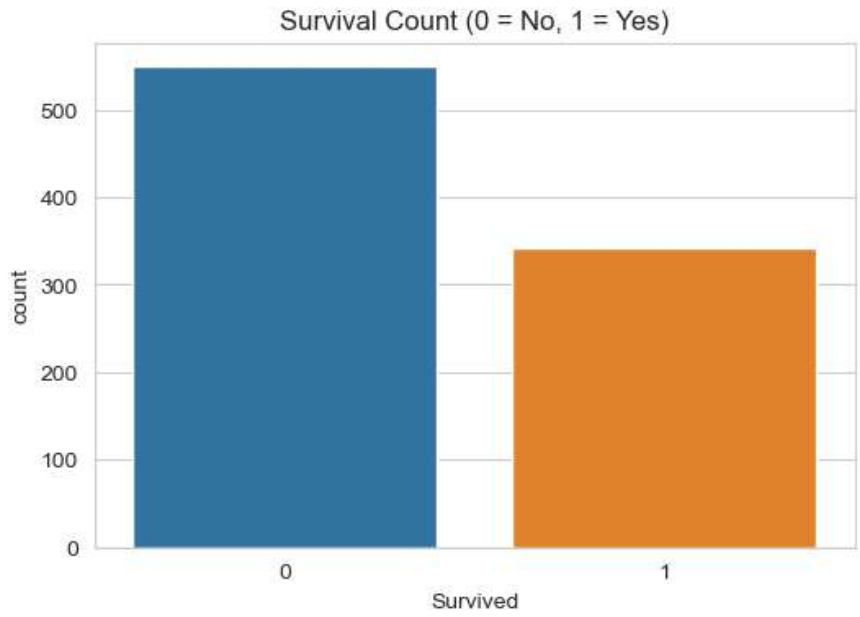
In [ ]:
```
train['Age'].fillna(train['Age'].median(), inplace=True)
train['Embarked'].fillna(train['Embarked'].mode()[0], inplace=True)
train.drop(columns=['Cabin'], inplace=True)
```

In [30]:
```
#visual exploration
```

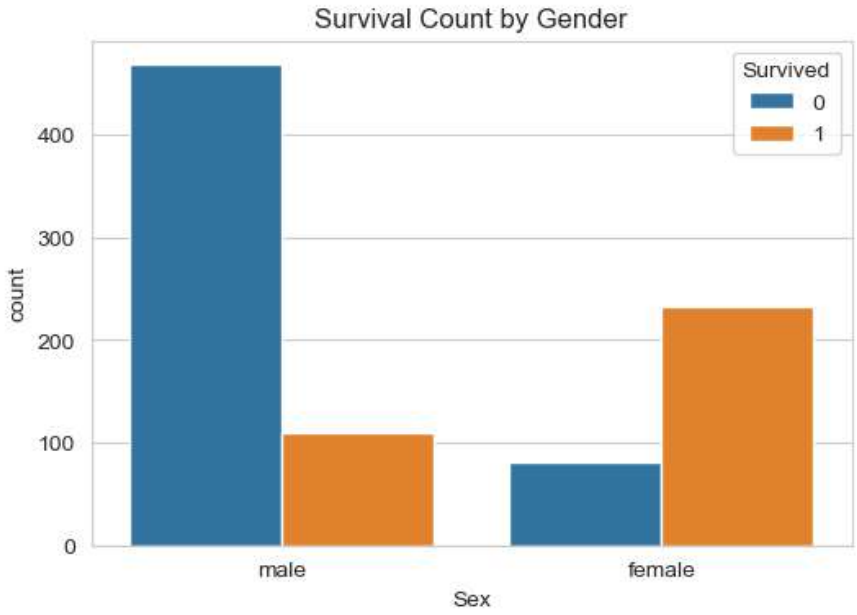
In [31]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [52]:
```
sns.set_style('whitegrid')
```

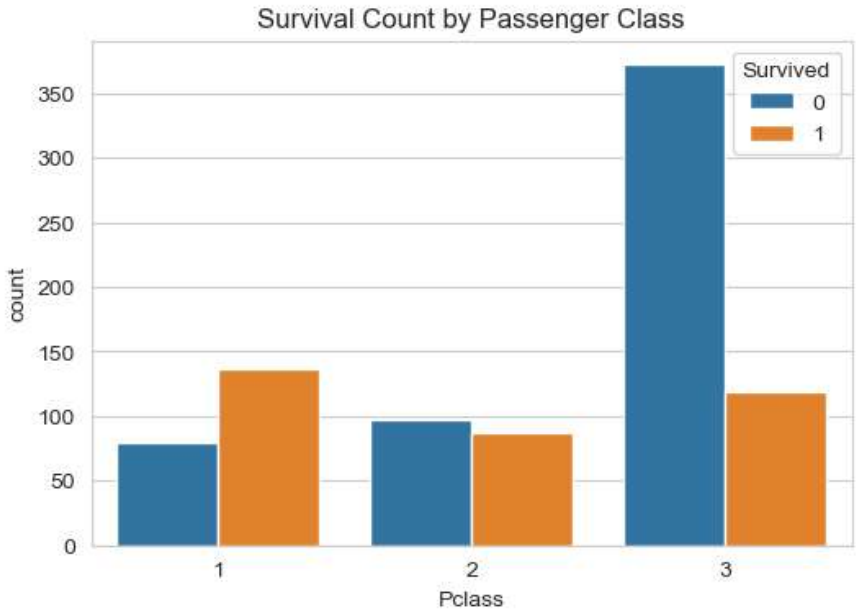
In [53]:
```
# Countplot: Survival count
plt.figure(figsize=(6,4))
sns.countplot(x='Survived', data=train)
plt.title('Survival Count (0 = No, 1 = Yes)')
plt.show()

# Observation: More passengers did not survive than survived.
```



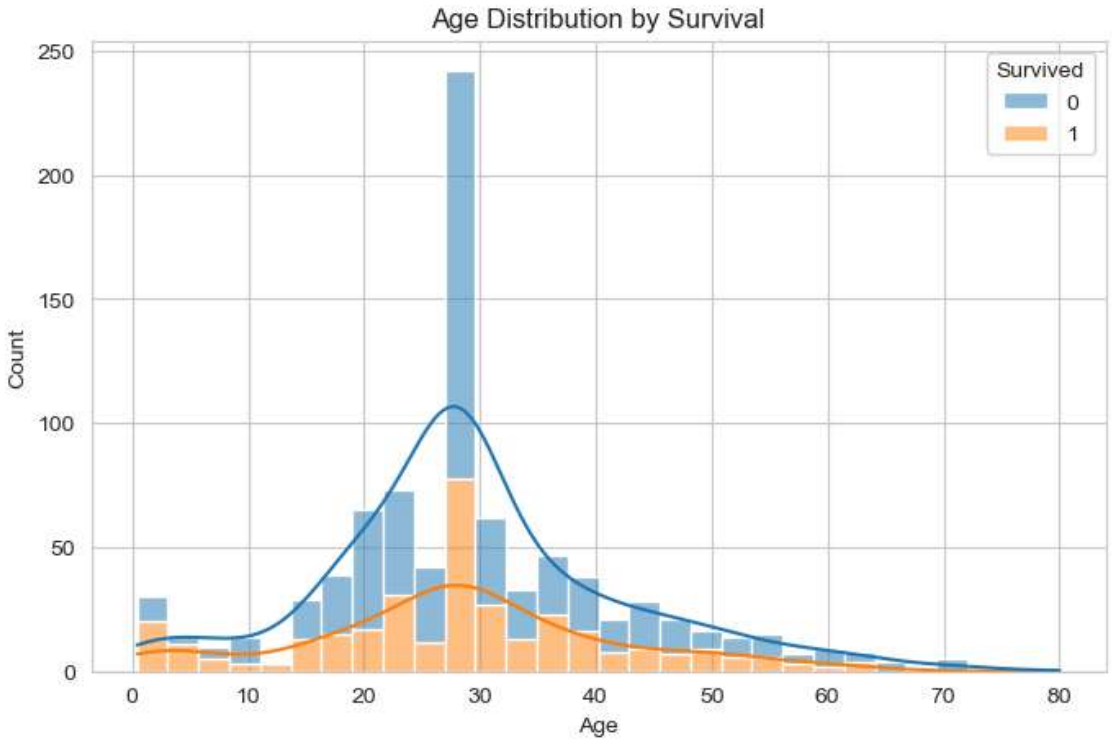
In [63]:
```
# Countplot: Survival by Gender
plt.figure(figsize=(6,4))
sns.countplot(x='Sex', hue='Survived', data=train)
plt.title('Survival Count by Gender')
plt.show()

# Observation: Females had much higher survival rates than males.
```



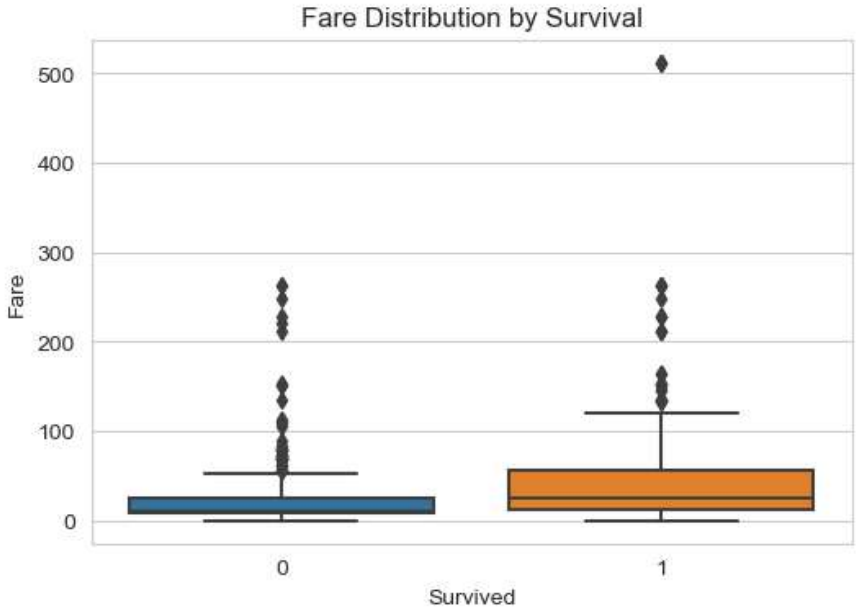
In [55]:
```
# Countplot: Survival by Passenger Class
plt.figure(figsize=(6,4))
sns.countplot(x='Pclass', hue='Survived', data=train)
plt.title('Survival Count by Passenger Class')
plt.show()

# Observation: 1st class passengers had better survival chances than 2nd and 3rd class.
```



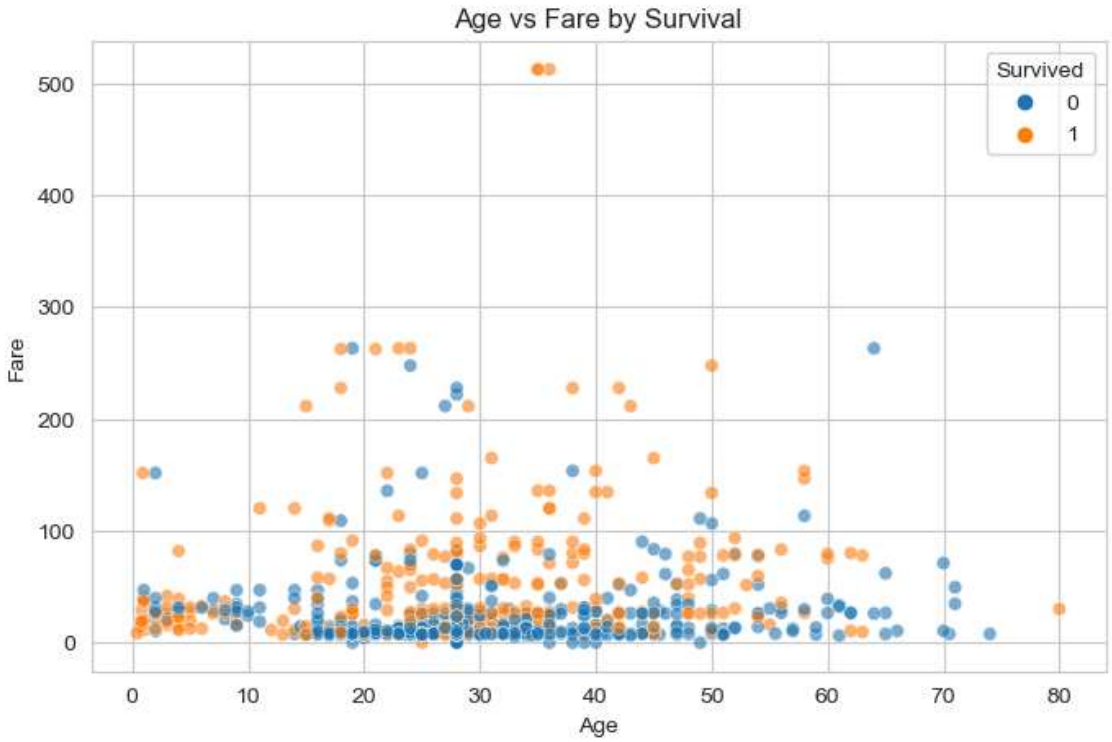
In [56]:
```
# Histogram: Age distribution by Survival
plt.figure(figsize=(8,5))
sns.histplot(data=train, x='Age', bins=30, kde=True, hue='Survived', multiple='stack')
plt.title('Age Distribution by Survival')
plt.show()

# Observation: Younger passengers had a slightly better survival rate.
```



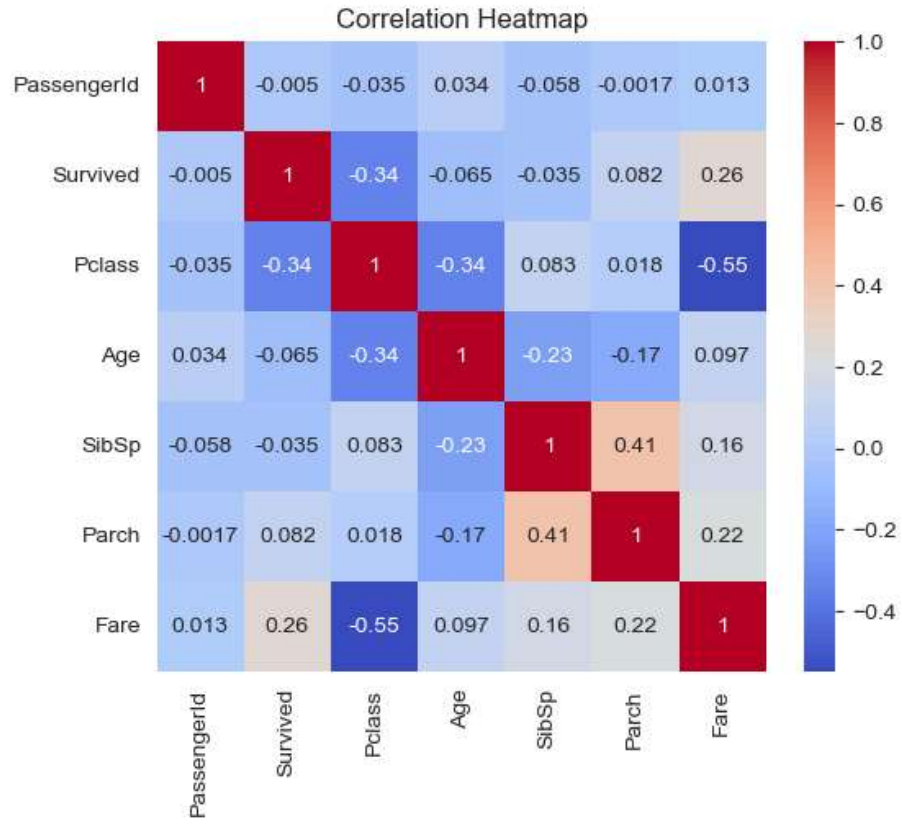
In [57]:
```
#  Boxplot: Fare by Survival
plt.figure(figsize=(6,4))
sns.boxplot(x='Survived', y='Fare', data=train)
plt.title('Fare Distribution by Survival')
plt.show()

# Observation: Passengers who paid higher fares tended to survive more.
```



In [58]:
```
# Scatterplot: Age vs Fare colored by Survival
plt.figure(figsize=(8,5))
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=train, alpha=0.6)
plt.title('Age vs Fare by Survival')
plt.show()

# Observation: Survivors generally paid higher fares; age distribution overlaps.
```

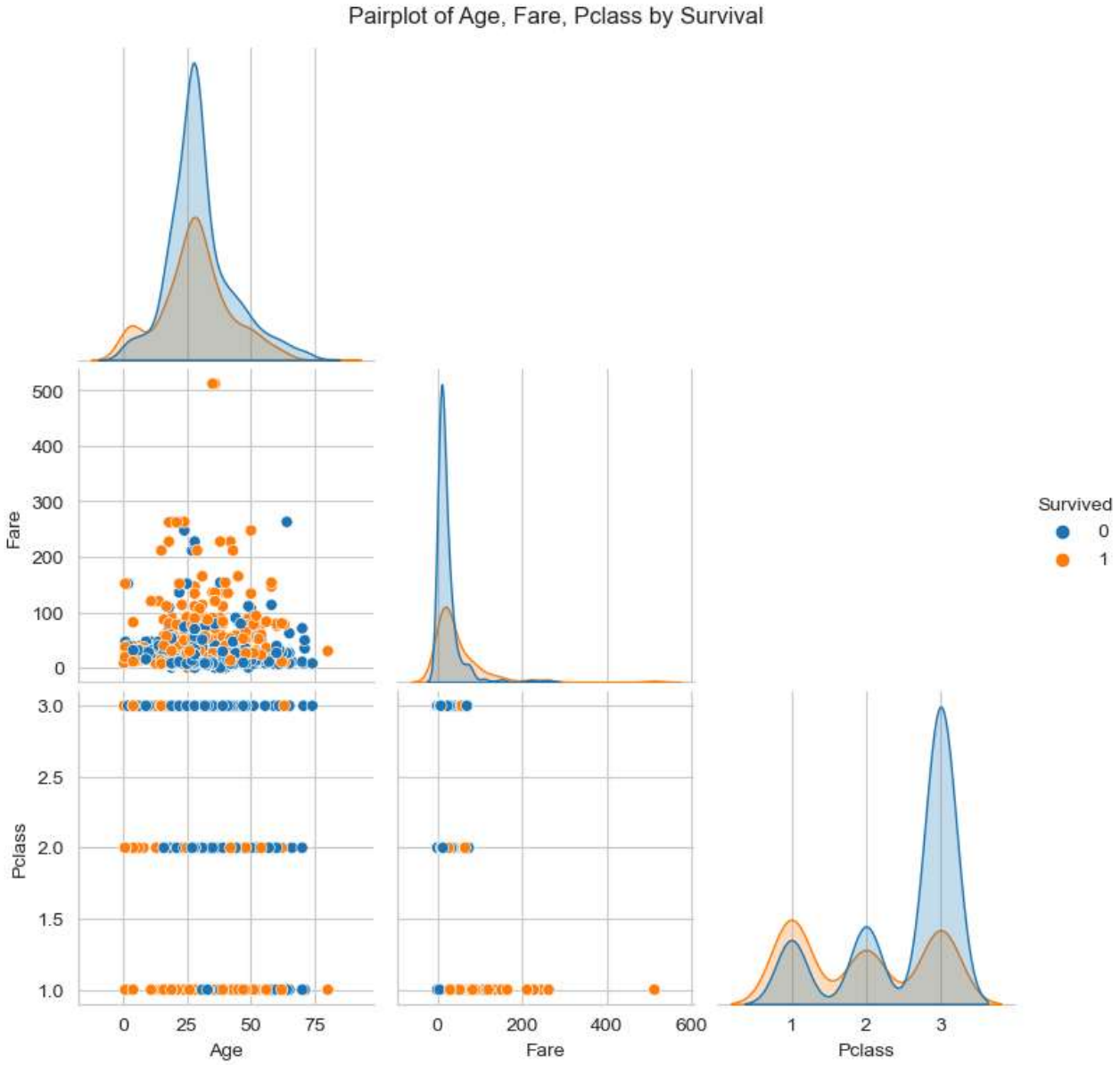


In [59]:
```
# Correlation Heatmap
plt.figure(figsize=(6,5))
sns.heatmap(train.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

# Observation: 'Pclass' is negatively correlated with 'Fare' and positively correlated with 'Survived'
```



In [64]:
```
# Pairplot of selected features colored by Survival
sns.pairplot(train[['Survived', 'Age', 'Fare', 'Pclass']], hue='Survived', diag_kind='kde', corner=Tru
plt.suptitle('Pairplot of Age, Fare, Pclass by Survival', y=1.02)
plt.show()

# Observation:
# - Survivors tend to cluster in different regions, e.g., higher Fare and lower Pclass.
# - Age distribution varies but overlaps across survival classes.
```

```
C:\Users\Shabi\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout ha
s changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



In [62]:
```
# Summary of findings


# Females had significantly higher survival rates than males.
# Passengers in 1st class had better survival chances compared to 2nd and 3rd class.
# Younger passengers and those who paid higher fares had a higher likelihood of survival.
# The 'Pclass' feature is moderately correlated with survival and fare, indicating socio-economic stat
# Missing data was mainly in 'Age' and 'Embarked'; filled with median and mode respectively.
```

In [ ]:
```

```