Import Liberaries

In [1]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import seaborn as sns
```

Load data

In [15]:
```
df = pd.read_csv("test.csv")
```

In [16]:
```
df.shape
```

Out[16]:
```
(418, 11)
```

In [17]:
```
df.head()
```

Out[17]:

	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

EDA - Exploratory Data Analysis

In [18]:
```
df.columns
```

Out[18]:
```
Index(['PassengerId', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch',
       'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

In [19]:
```
df.shape
```

Out[19]:
```
(418, 11)
```

In [20]:
```
df.size
```

Out[20]:
```
4598
```

In [21]:
```
df.index
```

Out[21]:
```
RangeIndex(start=0, stop=418, step=1)
```

Information about the data

In [22]:
```
df.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
```

Descriptive Statistics

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

Out[23]:

	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 [24]:
```
df.describe(include=object)
```

Out[24]:

	Name	Sex	Ticket	Cabin	Embarked
count	418	418	418	91	418
unique	418	2	363	76	3
top	Peter, Master. Michael J	male	PC 17608	B57 B59 B63 B66	S
freq	1	266	5	3	270

In [25]:
```
df.describe(include="number")
```

Out[25]:

	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 [26]:
```
df.describe(include="all")
```

Out[26]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
count	418.000000	418.000000	418	418	332.000000	418.000000	418.000000	418	417.000000	91	418
unique	NaN	NaN	418	2	NaN	NaN	NaN	363	NaN	76	3
top	NaN	NaN	Peter, Master. Michael J	male	NaN	NaN	NaN	PC 17608	NaN	B57 B59 B63 B66	S
freq	NaN	NaN	1	266	NaN	NaN	NaN	5	NaN	3	270
mean	1100.500000	2.265550	NaN	NaN	30.272590	0.447368	0.392344	NaN	35.627188	NaN	NaN
std	120.810458	0.841838	NaN	NaN	14.181209	0.896760	0.981429	NaN	55.907576	NaN	NaN
min	892.000000	1.000000	NaN	NaN	0.170000	0.000000	0.000000	NaN	0.000000	NaN	NaN
25%	996.250000	1.000000	NaN	NaN	21.000000	0.000000	0.000000	NaN	7.895800	NaN	NaN
50%	1100.500000	3.000000	NaN	NaN	27.000000	0.000000	0.000000	NaN	14.454200	NaN	NaN
75%	1204.750000	3.000000	NaN	NaN	39.000000	1.000000	0.000000	NaN	31.500000	NaN	NaN
max	1309.000000	3.000000	NaN	NaN	76.000000	8.000000	9.000000	NaN	512.329200	NaN	NaN

Missing value

In [27]:
```
df.isnull().sum()
```

Out[27]:
```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
```

In [28]:
```
df.isnull().any()
```

Out[28]:
```
PassengerId    False
Pclass         False
Name           False
Sex            False
Age             True
SibSp          False
Parch          False
Ticket         False
Fare            True
Cabin           True
Embarked       False
dtype: bool
```

Fill Missing Values

In [29]:
```
df['Age'] = df['Age'].fillna(df['Age'].median())
```

In [30]:
```
df = df.drop(columns = ['Cabin'])
```

In [31]:
```
df.isnull().sum()
```

Out[31]:
```
PassengerId    0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           1
Embarked       0
dtype: int64
```

In [32]:
```
df['Fare'] = df['Fare'].fillna(df['Fare'].median())
```

In [33]:
```
df.isna().sum()
```

Out[33]:
```
PassengerId    0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Embarked       0
dtype: int64
```

Check Duplicate

In [34]:
```
df.duplicated().sum()
```

Out[34]:
```
np.int64(0)
```

Check Value Counts

In [35]:
```
df.value_counts()
```

Out[35]:
```
PassengerId  Pclass  Name                                          Sex     Age   SibSp  Parch  Ticket   Fare      Em
barked
1309         3       Peter, Master. Michael J                      male    27.0  1      1      2668     22.3583   C
1
892          3       Kelly, Mr. James                              male    34.5  0      0      330911   7.8292    Q
1
1293         2       Gale, Mr. Harry                               male    38.0  1      0      28664    21.0000   S
1
1292         1       Bonnell, Miss. Caroline                       female  30.0  0      0      36928    164.8667  S
1
1291         3       Conlon, Mr. Thomas Henry                      male    31.0  0      0      21332    7.7333    Q
1

..
898          3       Connolly, Miss. Kate                          female  30.0  0      0      330972   7.6292    Q
1
897          3       Svensson, Mr. Johan Cervin                    male    14.0  0      0      7538     9.2250    S
1
896          3       Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female  22.0  1      1      3101298  12.2875   S
1
895          3       Wirz, Mr. Albert                              male    27.0  0      0      315154   8.6625    S
1
894          2       Myles, Mr. Thomas Francis                     male    62.0  0      0      240276   9.6875    Q
1
Name: count, Length: 418, dtype: int64
```

Check DataTypes

In [36]:
```
df.dtypes
```

Out[36]:
```
PassengerId      int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Embarked        object
dtype: object
```

Data Cleaning

In [37]:
```
df['Age'] = df['Age'].astype(int)
```

In [38]:
```
df['Name'] = df['Name'].str.strip()
df['Name'] = df['Name'].str.title()
```

In [39]:
```
df['Fare'] = df['Fare'].fillna(df['Fare'].median())
df['Fare'] = df['Fare'].astype(float)
```

In [40]:
```
df['Ticket'] = df['Ticket'].astype(str).str.upper().str.strip()
```

In [41]:
```
df['Name'] = df['Name'].str.replace(',', '', regex=True)
```

In [42]:
```
df['Sex'] = df['Sex'].str.strip().str.capitalize()
```

In [43]:
```
df.head()
```

Out[43]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked
0	892	3	Kelly Mr. James	Male	34	0	0	330911	7.8292	Q
1	893	3	Wilkes Mrs. James (Ellen Needs)	Female	47	1	0	363272	7.0000	S
2	894	2	Myles Mr. Thomas Francis	Male	62	0	0	240276	9.6875	Q
3	895	3	Wirz Mr. Albert	Male	27	0	0	315154	8.6625	S
4	896	3	Hirvonen Mrs. Alexander (Helga E Lindqvist)	Female	22	1	1	3101298	12.2875	S

Data Preprocessing and modifications

In [44]:
```
# Age group: Child, Young, Adult, Senior
df['Age_Group'] = pd.cut(df['Age'], bins=[0, 12, 18, 35, 60, 100],
                         labels=['Child','Teen','Young','Adult','Senior'])
```

In [45]:
```
df['Status'] = 'Active'  # all rows will have 'Active'
```

In [46]:
```
df.head()
```

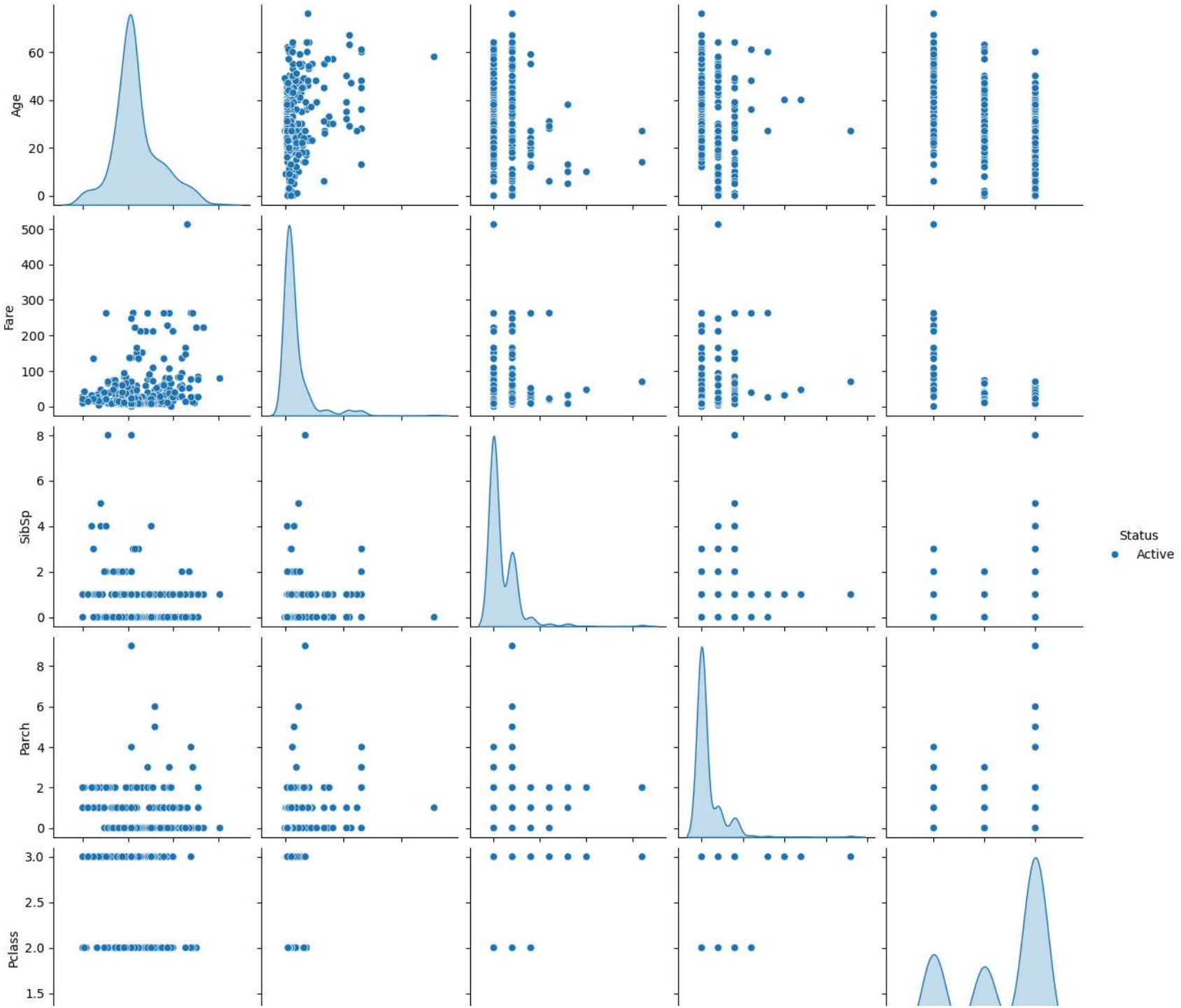
Out[46]:

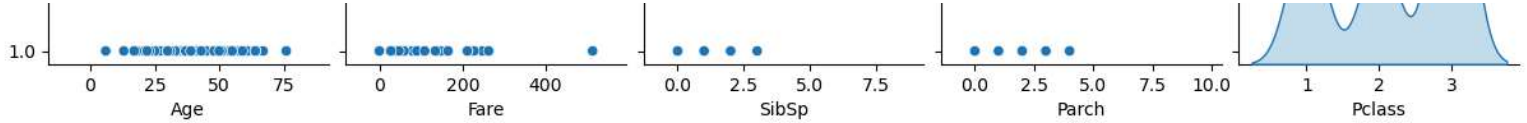
	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Embarked	Age_Group	Status
0	892	3	Kelly Mr. James	Male	34	0	0	330911	7.8292	Q	Young	Active
1	893	3	Wilkes Mrs. James (Ellen Needs)	Female	47	1	0	363272	7.0000	S	Adult	Active
2	894	2	Myles Mr. Thomas Francis	Male	62	0	0	240276	9.6875	Q	Senior	Active
3	895	3	Wirz Mr. Albert	Male	27	0	0	315154	8.6625	S	Young	Active
4	896	3	Hirvonen Mrs. Alexander (Helga E Lindqvist)	Female	22	1	1	3101298	12.2875	S	Young	Active

Visualization

1.Pairplot

In [48]:
```
sns.pairplot(df[['Age', 'Fare', 'SibSp', 'Parch', 'Pclass', 'Status']],
             hue='Status')
plt.show()
```

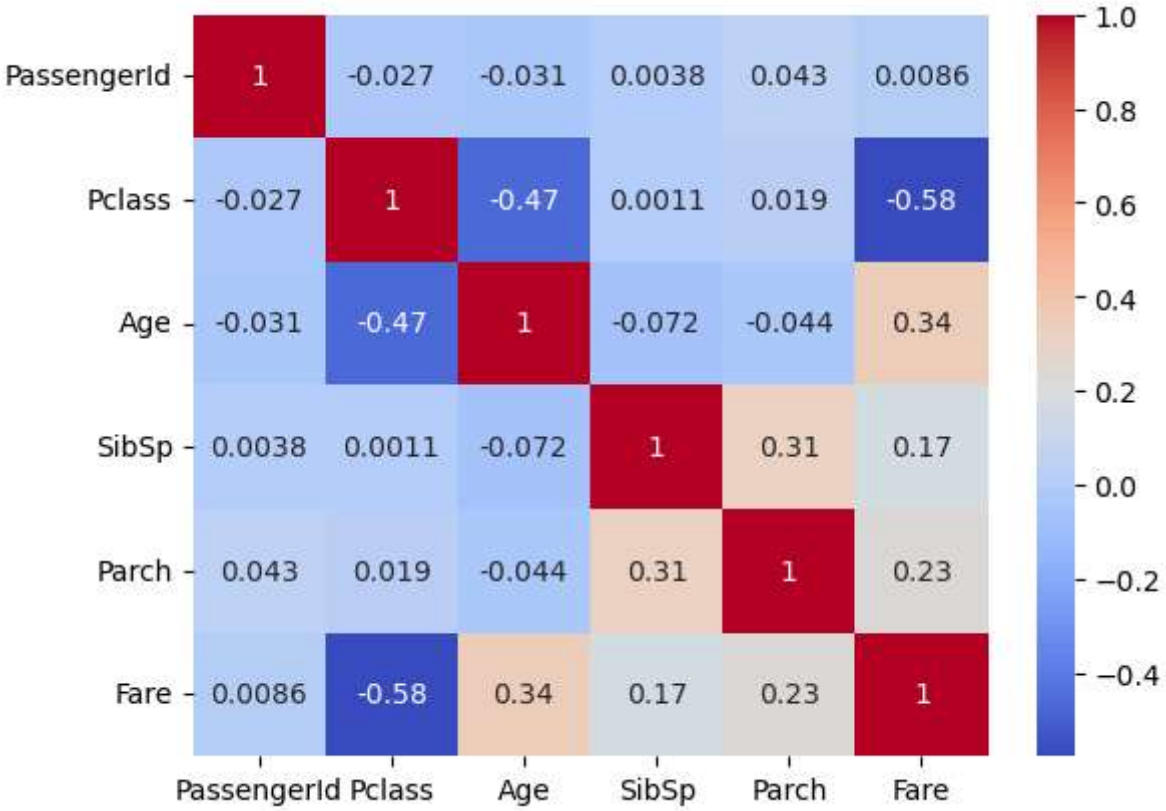

Age
Fare
SibSp
Parch
Pclass
Status
Active



Observations: Survivors usually paid higher fare. Younger passengers had higher survival. Pclass 1 (upper class) shows higher survival.

2.Heatmap (Correlation Matrix)

In [52]:
```
numeric_df = df.select_dtypes(include=['int64', 'float64'])
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.show()
```

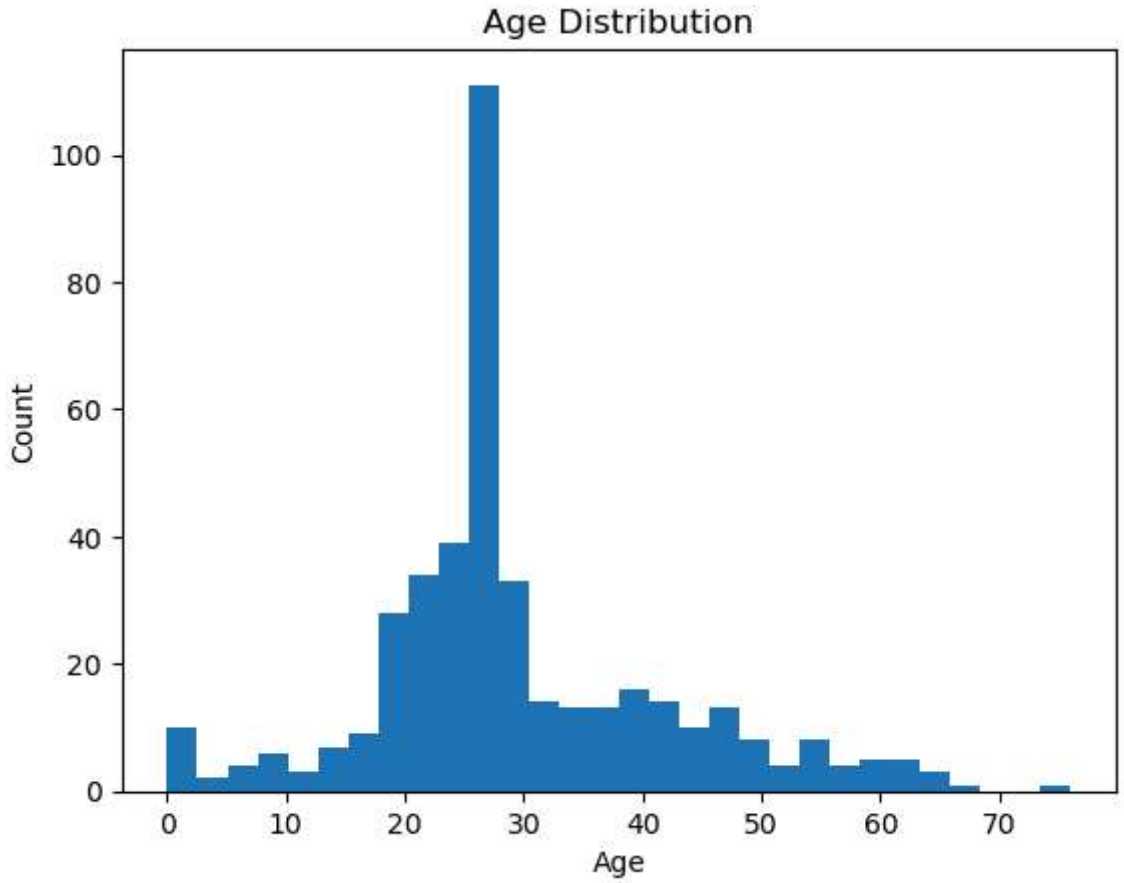


Observations: Fare and Pclass strongly negatively correlated (rich passengers = lower Pclass number). Survival (Status) correlates with Fare and Pclass. Age has weak correlation with survival.

3.Histograms (Distribution Plots)

3.1.Age Distribution

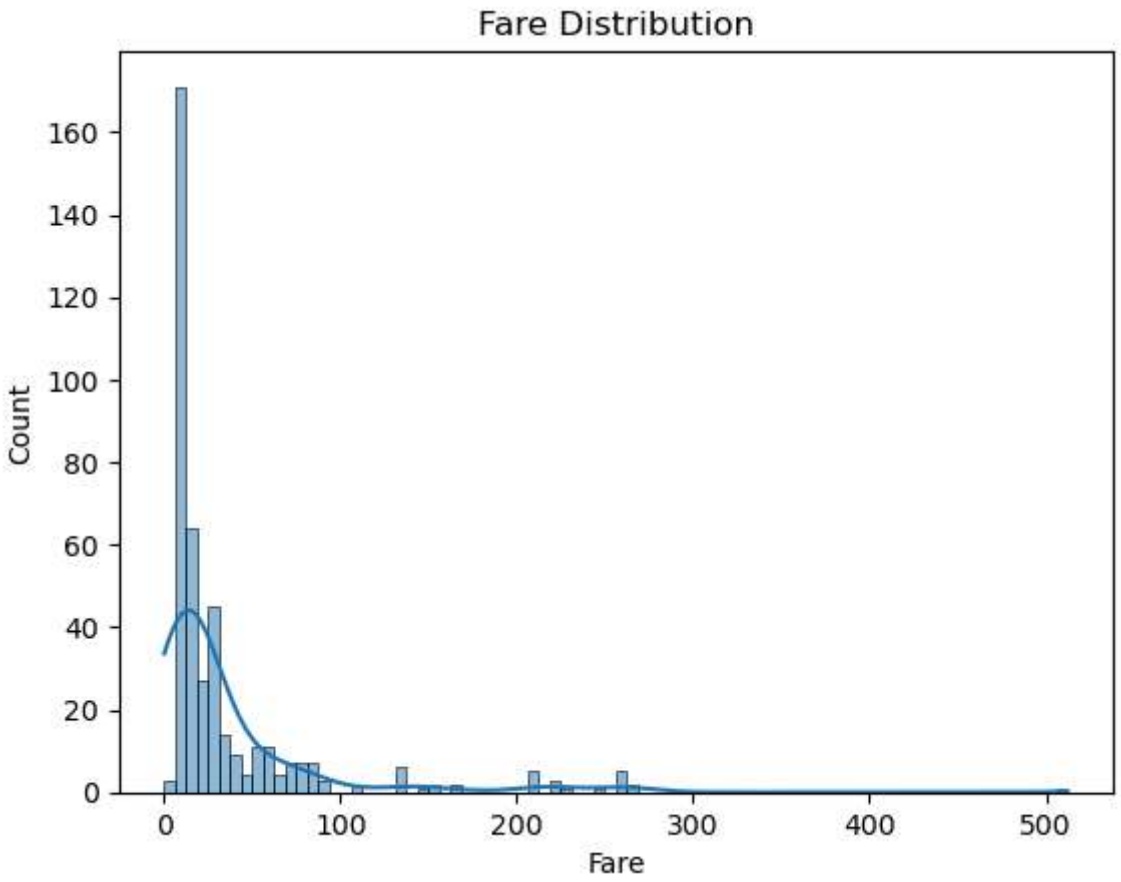
In [54]:
```
plt.hist(df['Age'], bins=30)
plt.title("Age Distribution")
plt.xlabel("Age")
plt.ylabel("Count")
plt.show()
```



Observation: Most passengers are between 20–40 years

3.2Histrogram of Fare

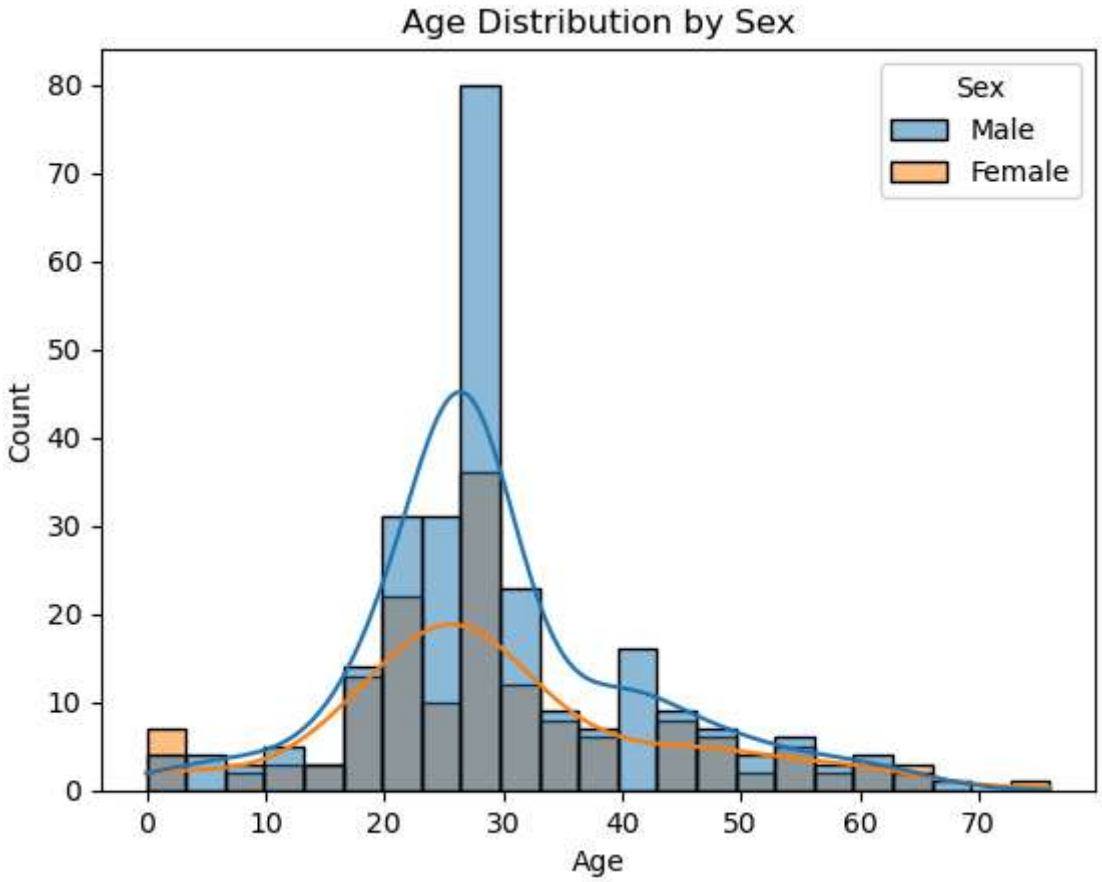
In [58]:
```
sns.histplot(data=df, x='Fare', kde=True)
plt.title("Fare Distribution")
plt.show()
```



Observation: Fare is highly right-skewed (many low fares, few very high).

3.3Histrogram of Age grouped by Sex

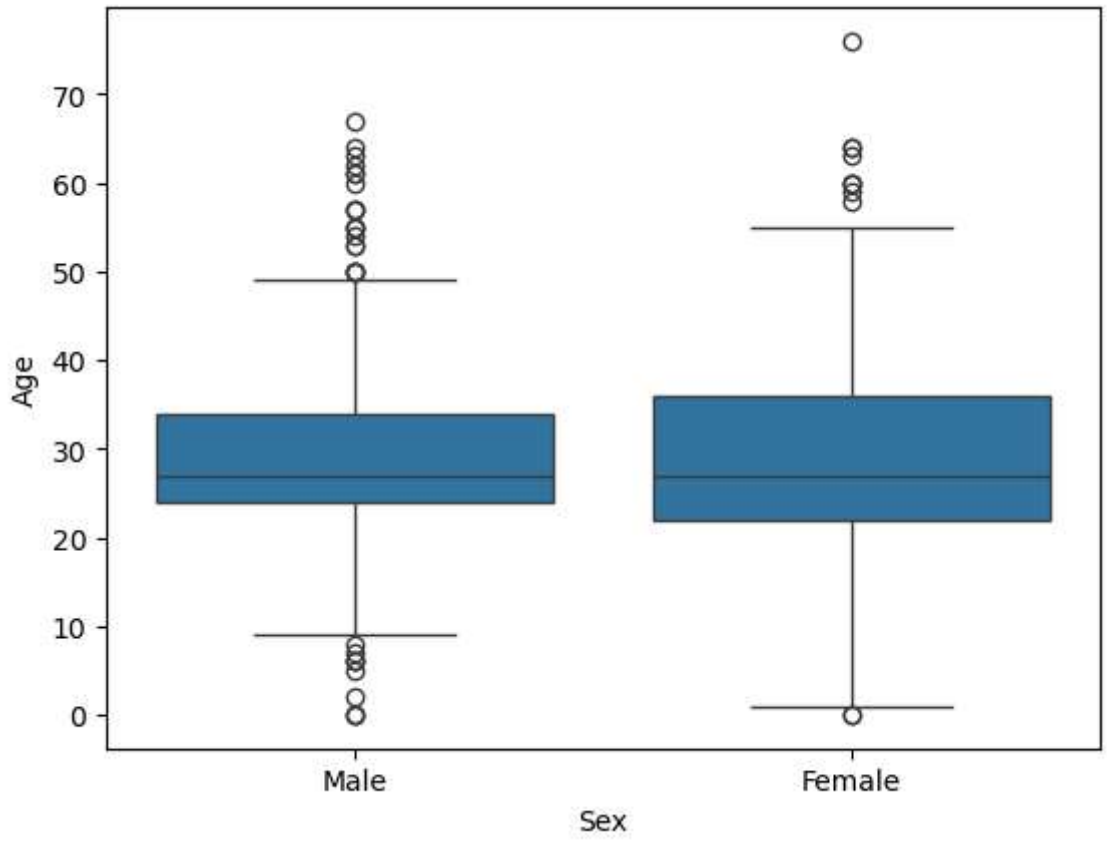
In [59]:
```
sns.histplot(data=df, x='Age', hue='Sex', kde=True)
plt.title("Age Distribution by Sex")
plt.show()
```



4Boxplots

4.1.:Age vs Sex

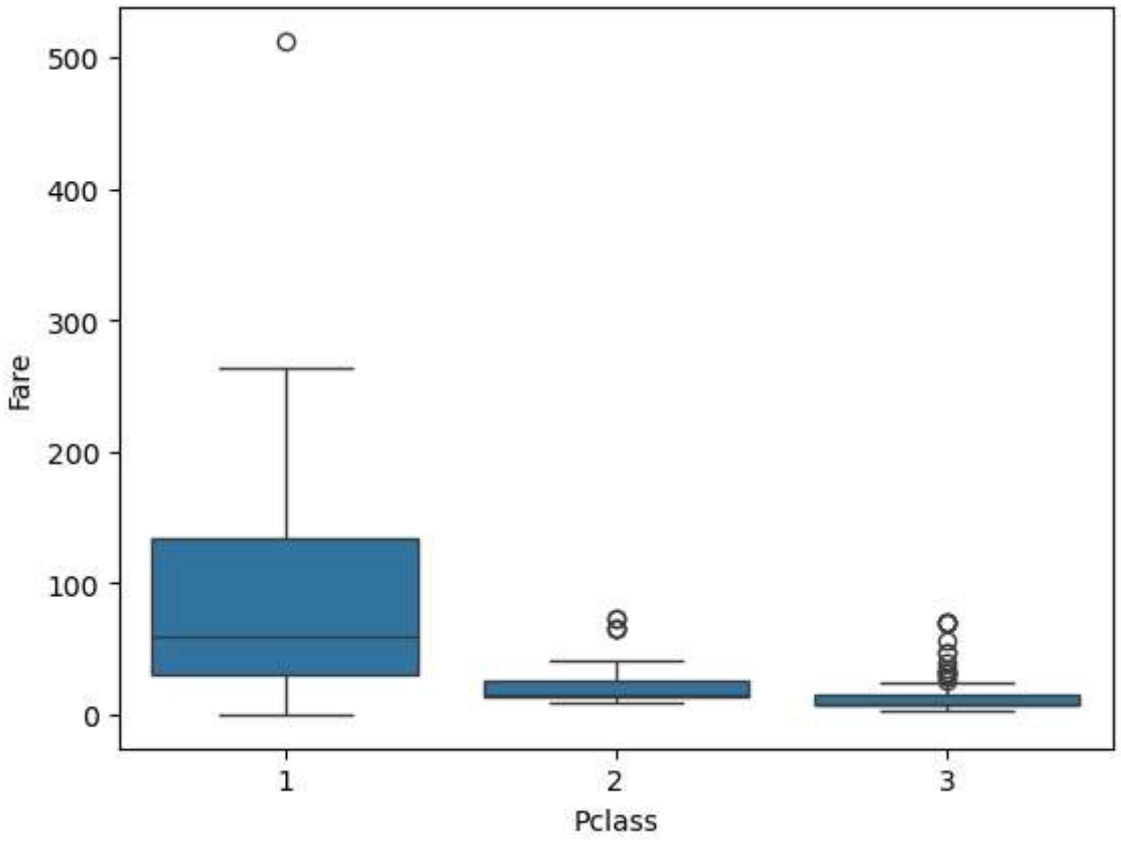
In [60]:
```
sns.boxplot(x='Sex', y='Age', data=df)
plt.show()
```



Observation: 1st class fares are much higher. 3rd class has lowest fare range. Male age distribution is slightly wider; no major outliers.

4.2. Fare vs Pclass

In [61]:
```
sns.boxplot(x='Pclass', y='Fare', data=df)
plt.show()
```

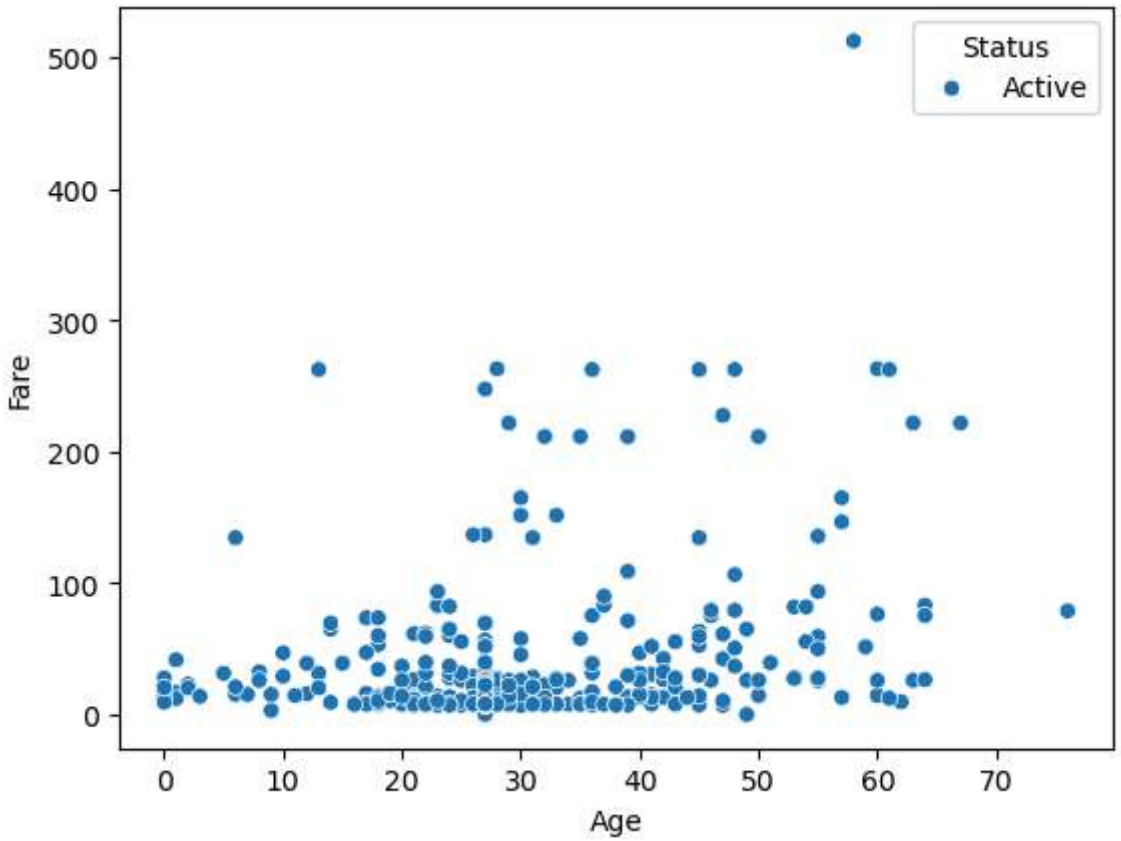


Observation: 1st class fares are much higher. 3rd class has lowest fare range.

5. Scatterplots

5.1 Fare vs Age

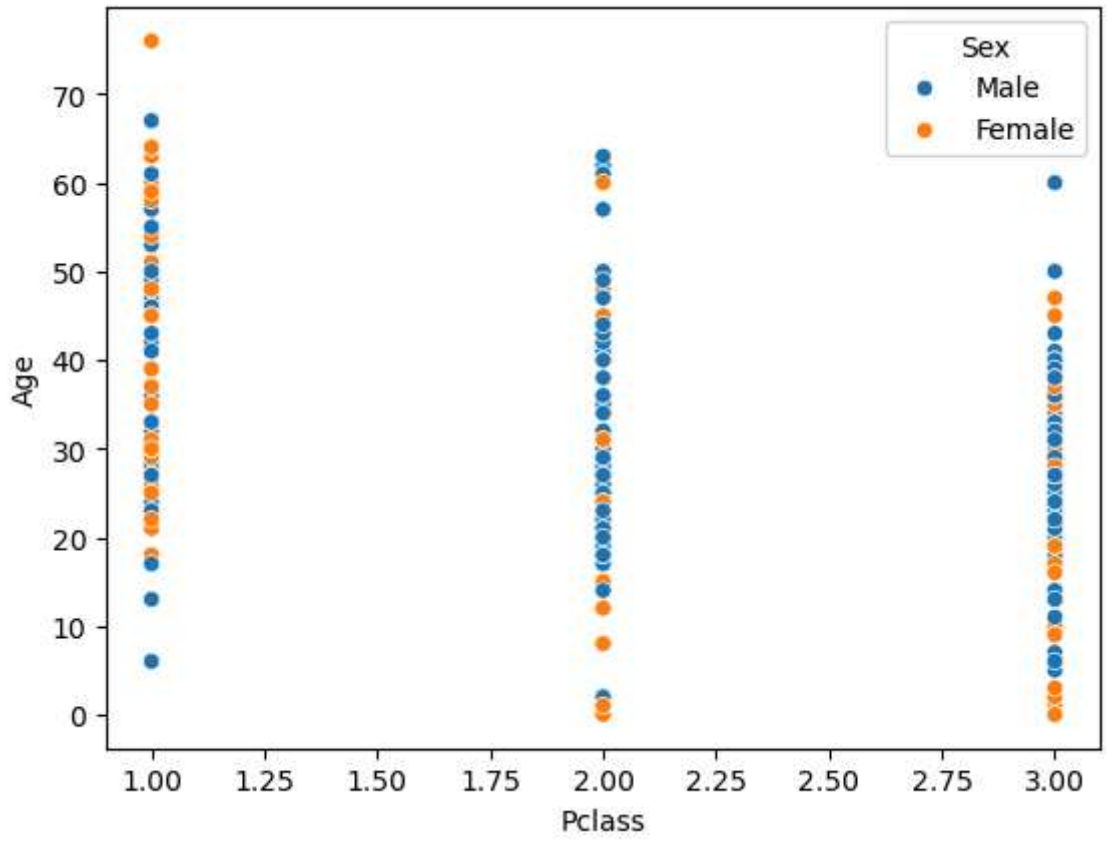
In [62]:
```
sns.scatterplot(x='Age', y='Fare', hue='Status', data=df)
plt.show()
```



Observation: Higher fare passengers were more likely to survive at all ages.

5.2 .Pclass vs Age

In [63]:
```
sns.scatterplot(x='Pclass', y='Age', hue='Sex', data=df)
plt.show()
```



Observation: Class distribution is spread across age groups.

In [ ]: