In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
df = pd.read_csv("train.csv")
```

In [4]:
```
df.head()
```

Out[4]:

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

In [6]:
```
df.dtypes
```

Out[6]:
```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

In [9]:
```
df['Age'] = df['Age'].astype('int64')
```

```
---------------------------------------------------------------------------
IntCastingNaNError                        Traceback (most recent call last)
Cell In[9], line 1
----> 1 df['Age'] = df[     ].astype(       )

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\pandas\core\generi
c.py:6643, in NDFrame.astype(self, dtype, copy, errors)
   6637     results = [
   6638         ser.astype(dtype, copy=copy, errors=errors) for _, ser in self.items
()
   6639     ]
   6641 else:
   6642     # else, only a single dtype is given
-> 6643     new_data = self._mgr.astype(dtype=dtype, copy=copy, errors=errors)
   6644     res = self._constructor_from_mgr(new_data, axes=new_data.axes)
   6645     return res.__finalize__(self, method="astype")

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\pandas\core\interna
ls\managers.py:430, in BaseBlockManager.astype(self, dtype, copy, errors)
    427 elif using_copy_on_write():
    428     copy = False
--> 430 return self.apply(
    431             ,
    432     dtype=dtype,
    433     copy=copy,
    434     errors=errors,
    435     using_cow=using_copy_on_write(),
    436 )

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\pandas\core\interna
ls\managers.py:363, in BaseBlockManager.apply(self, f, align_keys, **kwargs)
    361         applied = b.apply(f, **kwargs)
    362     else:
--> 363         applied = getattr(b, f)(**kwargs)
    364     result_blocks = extend_blocks(applied, result_blocks)
    366 out = type(self).from_blocks(result_blocks, self.axes)

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\pandas\core\interna
ls\blocks.py:758, in Block.astype(self, dtype, copy, errors, using_cow, squeeze)
    755         raise ValueError("Can not squeeze with more than one column.")
    756     values = values[0, :]  # type: ignore[call-overload]
--> 758 new_values = astype_array_safe(values, dtype, copy=copy, errors=errors)
    760 new_values = maybe_coerce_values(new_values)
    762 refs = None

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\pandas\core\dtypes
\astype.py:237, in astype_array_safe(values, dtype, copy, errors)
    234     dtype = dtype.numpy_dtype
    236 try:
--> 237     new_values = astype_array(values, dtype, copy=copy)
    238 except (ValueError, TypeError):
    239     # e.g. _astype_nansafe can fail on object-dtype of strings
    240     #  trying to convert to float
    241     if errors == "ignore":

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\pandas\core\dtypes
```

```
\astype.py:182, in astype_array(values, dtype, copy)
    179     values = values.astype(dtype, copy=copy)
    181 else:
--> 182     values = _astype_nansafe(values, dtype, copy=copy)
    184 # in pandas we don't store numpy str dtypes, so convert to object
    185 if isinstance(dtype, np.dtype) and issubclass(values.dtype.type, str):

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\pandas\core\dtypes
\astype.py:101, in _astype_nansafe(arr, dtype, copy, skipna)
     96     return lib.ensure_string_array(
     97         arr, skipna=skipna, convert_na_value=False
     98     ).reshape(shape)
    100 elif np.issubdtype(arr.dtype, np.floating) and dtype.kind in "iu":
--> 101     return _astype_float_to_int_nansafe(arr, dtype, copy)
    103 elif arr.dtype == object:
    104     # if we have a datetime/timedelta array of objects
    105     # then coerce to datetime64[ns] and use DatetimeArray.astype
    107     if lib.is_np_dtype(dtype, "M"):

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\pandas\core\dtypes
\astype.py:145, in _astype_float_to_int_nansafe(values, dtype, copy)
    141 """
    142 astype with a check preventing converting NaN to an meaningless integer valu
e.
    143 """
    144 if not np.isfinite(values).all():
--> 145     raise IntCastingNaNError(
    146         "Cannot convert non-finite values (NA or inf) to integer"
    147     )
    148 if dtype.kind == "u":
    149     # GH#45151
    150     if not (values >= 0).all():

IntCastingNaNError: Cannot convert non-finite values (NA or inf) to integer
```

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [11]:
```
df.head()
df.info()
df.describe()
df.isnull().sum()
df.nunique()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 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          714 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  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

Out[11]:
```
PassengerId    891
Survived         2
Pclass           3
Name           891
Sex              2
Age             88
SibSp            7
Parch            7
Ticket         681
Fare           248
Cabin          147
Embarked         3
dtype: int64
```

In [12]:
```
median_age = df['Age'].median()
df['Age'].fillna(median_age, inplace=True)
```

In [13]:
```
df.isnull().sum()
```

Out[13]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age              0
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [15]:
```
df['Cabin'].fillna('Unknown')
```

Out[15]:
```
0      Unknown
1          C85
2      Unknown
3         C123
4      Unknown
        ...
886    Unknown
887        B42
888    Unknown
889       C148
890    Unknown
Name: Cabin, Length: 891, dtype: object
```

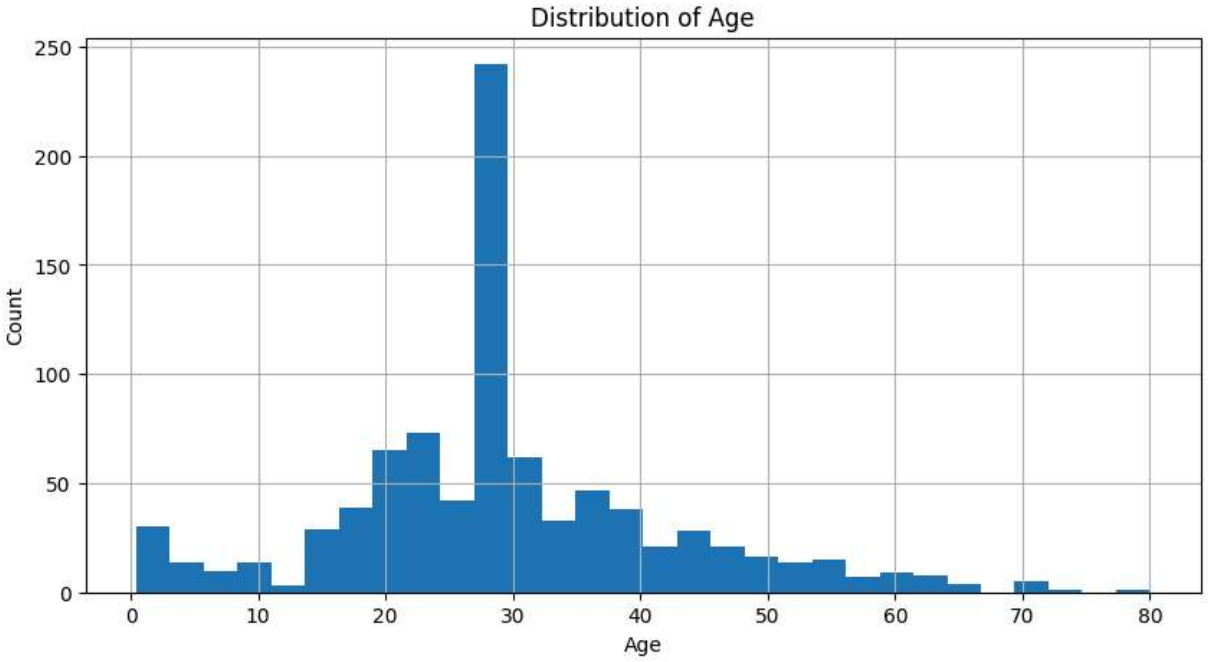
In [17]:
```
# Example for 'Embarked' column
most_frequent_value = df['Embarked'].mode()[0]  # get most frequent value
df['Embarked'].fillna(most_frequent_value)
```

Out[17]:
```
0      S
1      C
2      S
3      S
4      S
      ..
886    S
887    S
888    S
889    C
890    Q
Name: Embarked, Length: 891, dtype: object
```

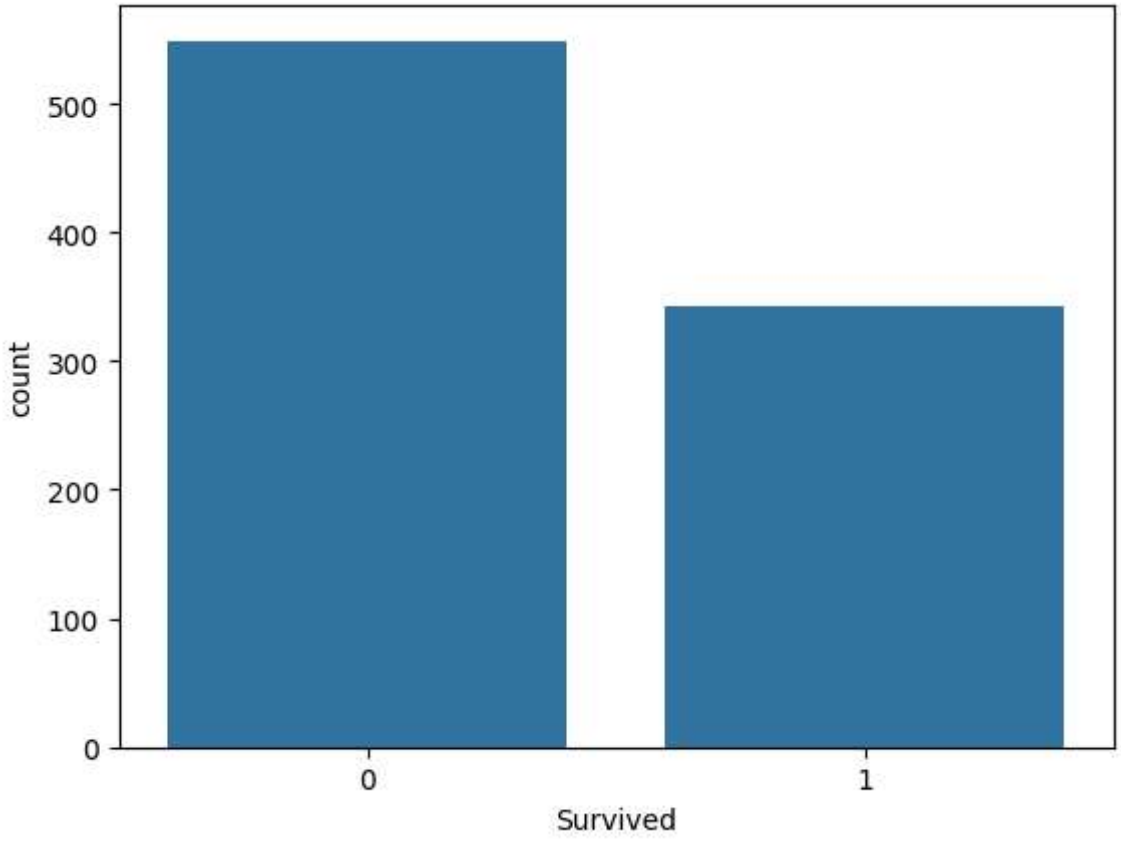
In [18]:
```
df.isnull().sum()
```

Out[18]:
```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

In [19]:
```
# Histogram of Age
df['Age'].hist(bins=30, figsize=(10,5))
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()

# Countplot of Survived
sns.countplot(x='Survived', data=df)
```

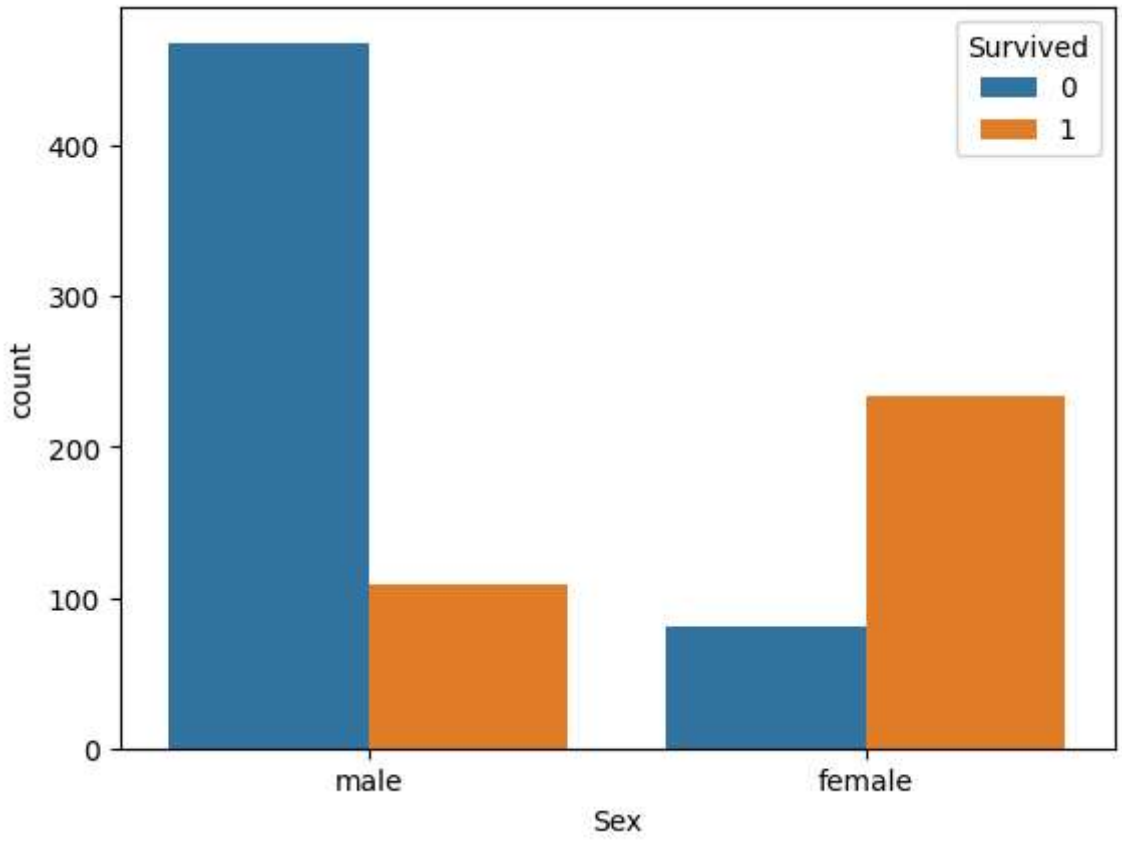


Out[19]: <Axes: xlabel='Survived', ylabel='count'>



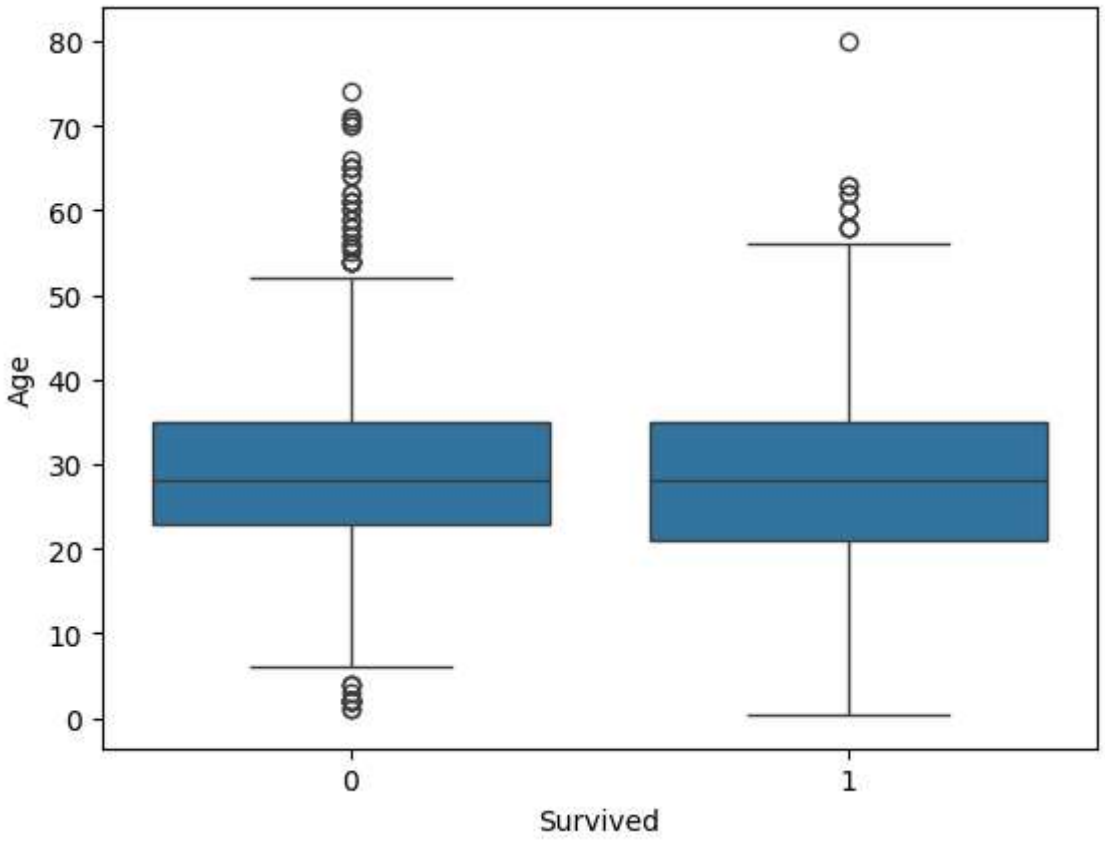
In [22]:
```
# Survived vs Sex
sns.countplot(x='Sex', hue='Survived', data=df)
```

Out[22]: <Axes: xlabel='Sex', ylabel='count'>



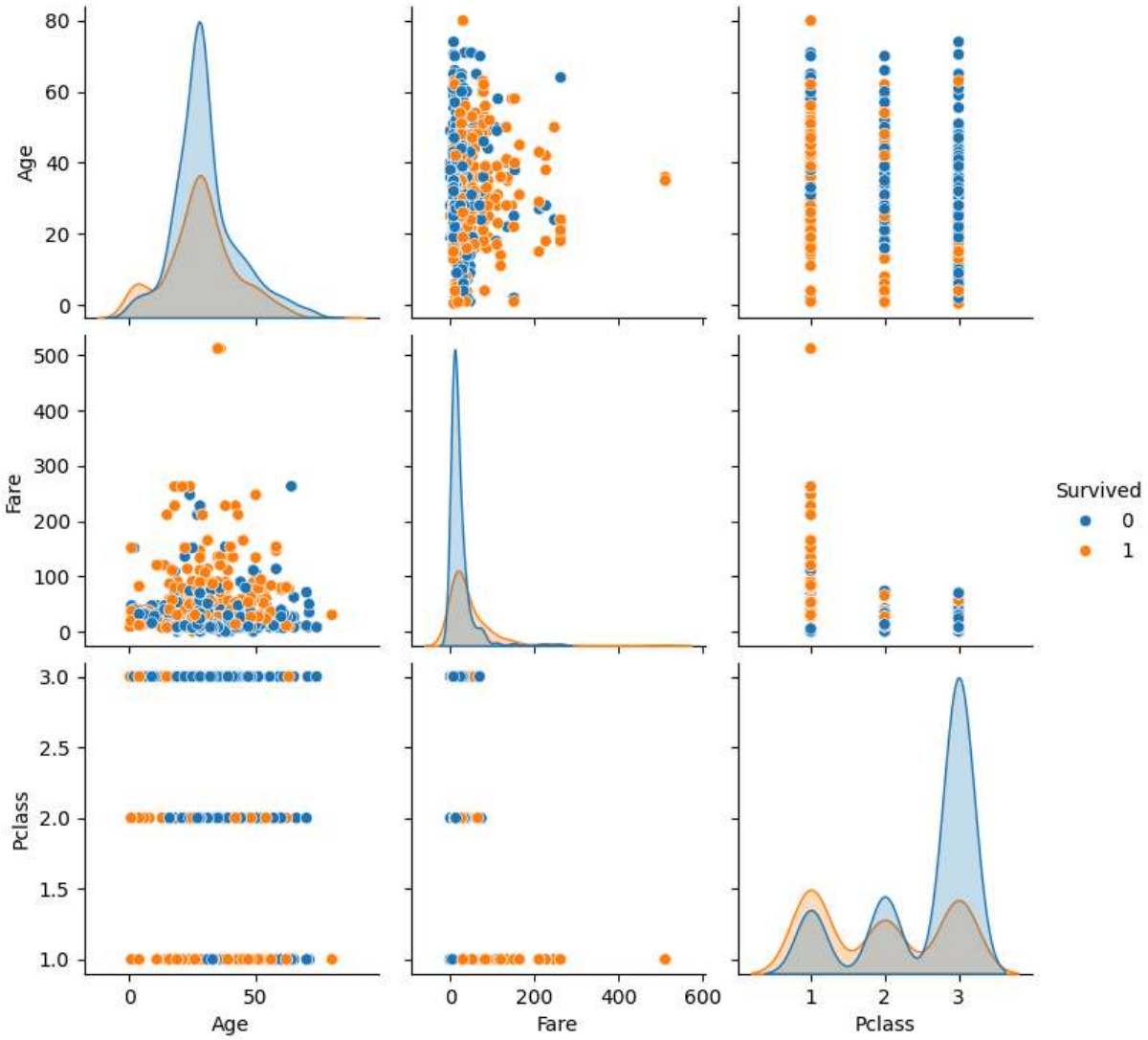
In [23]:
```
# Boxplot of Age vs Survived
sns.boxplot(x='Survived', y='Age', data=df)
```

Out[23]:
```
<Axes: xlabel='Survived', ylabel='Age'>
```



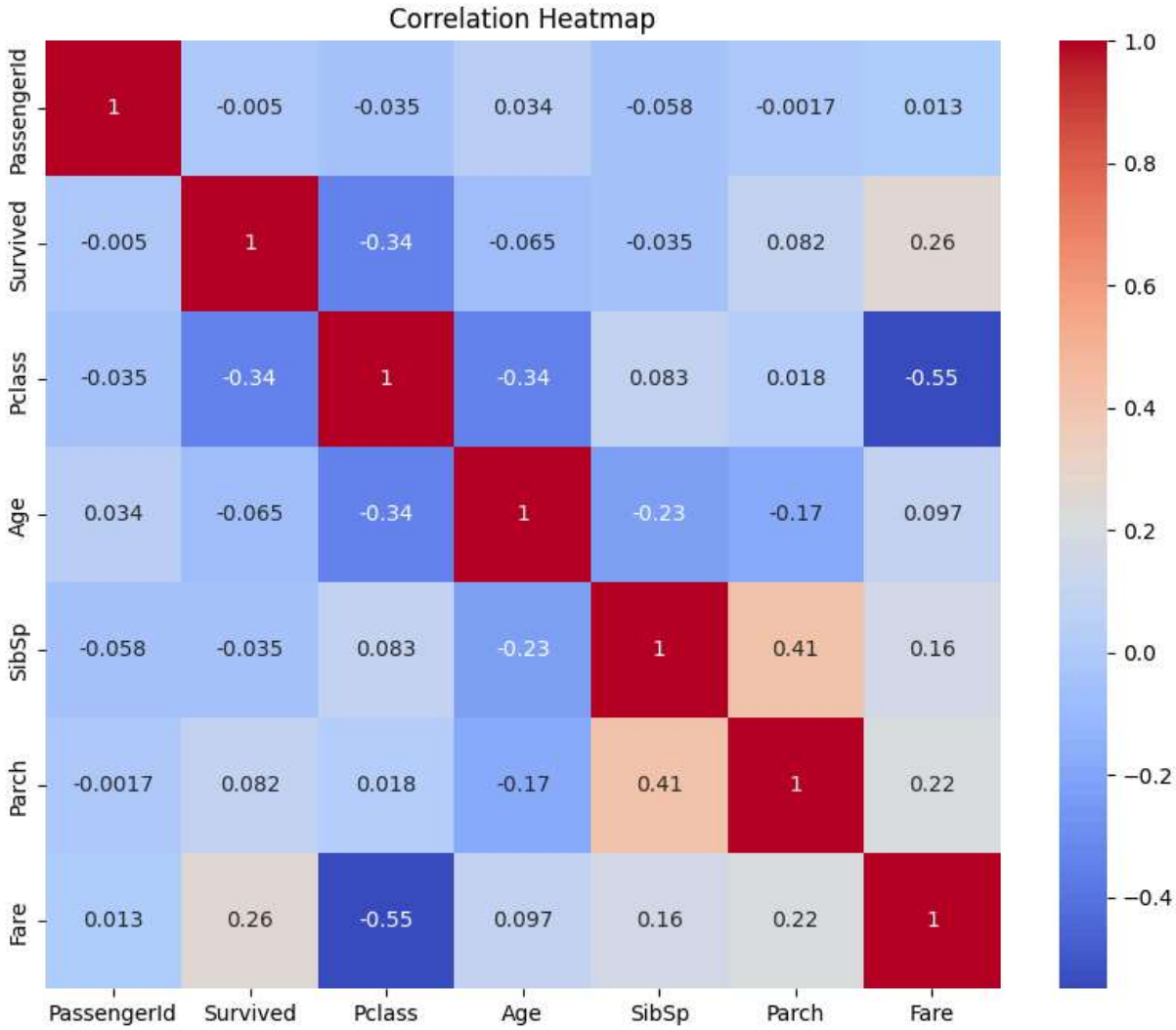
In [24]:
```
# Pairplot
sns.pairplot(df[['Survived', 'Age', 'Fare', 'Pclass']], hue='Survived')

# Heatmap for correlation
plt.figure(figsize=(10,8))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[24], line 6
      4 # Heatmap for correlation
      5 plt.figure(figsize=(10,8))
----> 6 sns.heatmap(df.corr(), annot=True, cmap='coolwarm')

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\pandas\core\frame.p
y:11049, in DataFrame.corr(self, method, min_periods, numeric_only)
  11047 cols = data.columns
  11048 idx = cols.copy()
> 11049 mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
  11051 if method == "pearson":
  11052     correl = libalgos.nancorr(mat, minp=min_periods)

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\pandas\core\frame.p
y:1993, in DataFrame.to_numpy(self, dtype, copy, na_value)
   1991 if dtype is not None:
   1992     dtype = np.dtype(dtype)
-> 1993 result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_value)
   1994 if result.dtype is not dtype:
   1995     result = np.asarray(result, dtype=dtype)

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\pandas\core\interna
ls\managers.py:1694, in BlockManager.as_array(self, dtype, copy, na_value)
   1692         arr.flags.writeable = False
   1693 else:
-> 1694     arr = self._interleave(dtype=dtype, na_value=na_value)
   1695     # The underlying data was copied within _interleave, so no need
   1696     # to further copy if copy=True or setting na_value
   1698 if na_value is lib.no_default:

File ~\AppData\Local\Programs\Python\Python313\Lib\site-packages\pandas\core\interna
ls\managers.py:1753, in BlockManager._interleave(self, dtype, na_value)
   1751     else:
   1752         arr = blk.get_values(dtype)
-> 1753     result[rl.indexer] = arr
   1754     itemmask[rl.indexer] = 1
   1756 if not itemmask.all():

ValueError: could not convert string to float: 'Braund, Mr. Owen Harris'
```



<Figure size 1000x800 with 0 Axes>

In [25]:
```
# Select only numeric columns for correlation
numeric_df = df.select_dtypes(include=['number'])

# Now draw the heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```



In [26]:
```
print(numeric_df.columns)
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare'], dtype
='object')
```

In [ ]: