In [8]:
```
#dataset=https://github.com/venky14/Data-Analysis-Project-Titanic/blob/master/titanic_data.csv
```

In [9]:
```
import pandas as pd
import numpy as np
import random as rnd

# visualization
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import Perceptron
from sklearn.linear_model import SGDClassifier
from sklearn.tree import DecisionTreeClassifier
```

Acquire data

In [10]:
```
train_df = pd.read_csv('titanic_data.csv')
test_df = pd.read_csv('titanic_data.csv')
combine = [train_df, test_df]
```

Analyze

In [12]:
```
print(train_df.columns.values)
```

```
['PassengerId' 'Survived' 'Pclass' 'Name' 'Sex' 'Age' 'SibSp' 'Parch'
 'Ticket' 'Fare' 'Cabin' 'Embarked.']
```

In [13]:
```
train_df.head()
```

Out[13]:

	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 [14]:
```
train_df.tail()
```

Out[14]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked.
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

In [15]:
```
train_df.info()
print('_'*40)
test_df.info()
```

```
<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
```

```
<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
```

In [16]:
```
train_df.describe()
```

Out[16]:

	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 [17]:
```
train_df.describe(include=['O'])
```

Out[17]:

	Name	Sex	Ticket	Cabin	Embarked.
count	891	891	891	204	889
unique	891	2	681	147	3
top	Braund, Mr. Owen Harris	male	347082	B96 B98	S
freq	1	577	7	4	644

In [18]:
```
train_df[['Pclass', 'Survived']].groupby(['Pclass'], as_index=False).mean().sort_values(by='Survived', ascending=Fals
```

Out[18]:

	Pclass	Survived
0	1	0.629630
1	2	0.472826
2	3	0.242363

In [19]:
```
train_df[["Sex", "Survived"]].groupby(['Sex'], as_index=False).mean().sort_values(by='Survived', ascending=False)
```

Out[19]:

	Sex	Survived
0	female	0.742038
1	male	0.188908

In [21]:
```
train_df[["SibSp", "Survived"]].groupby(['SibSp'], as_index=False).mean().sort_values(by='Survived', ascending=False)
```

Out[21]:

	SibSp	Survived
1	1	0.535885
2	2	0.464286
0	0	0.345395
3	3	0.250000
4	4	0.166667
5	5	0.000000
6	8	0.000000

In [23]:
```
train_df[["Parch", "Survived"]].groupby(['Parch'], as_index=False).mean().sort_values(by='Survived', ascending=False)
```

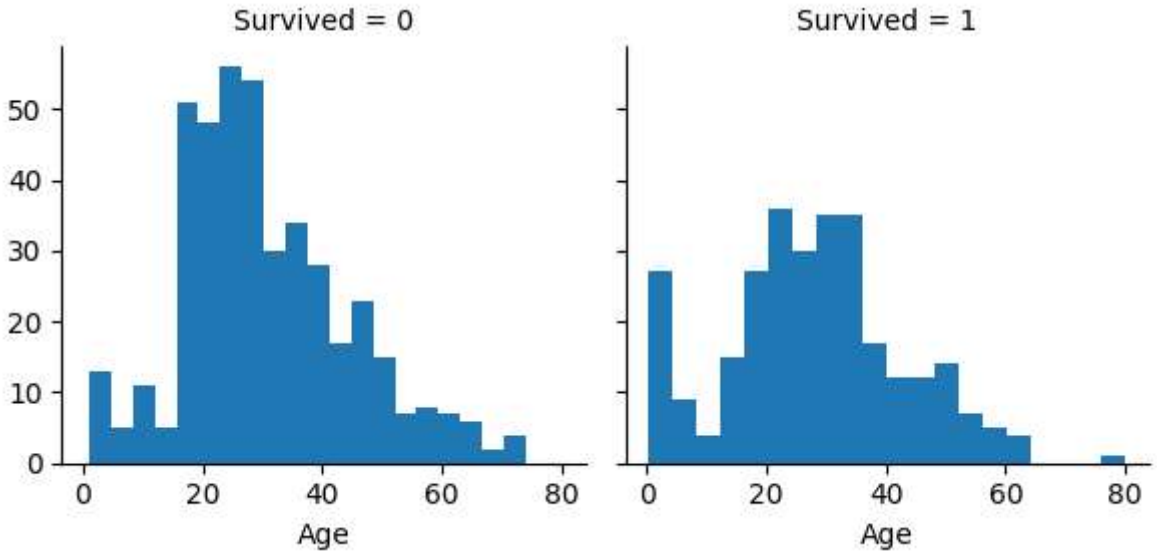
Out[23]:

	Parch	Survived
3	3	0.600000
1	1	0.550847
2	2	0.500000
0	0	0.343658
5	5	0.200000
4	4	0.000000
6	6	0.000000

In [24]:
```
g = sns.FacetGrid(train_df, col='Survived')
g.map(plt.hist, 'Age', bins=20)
```

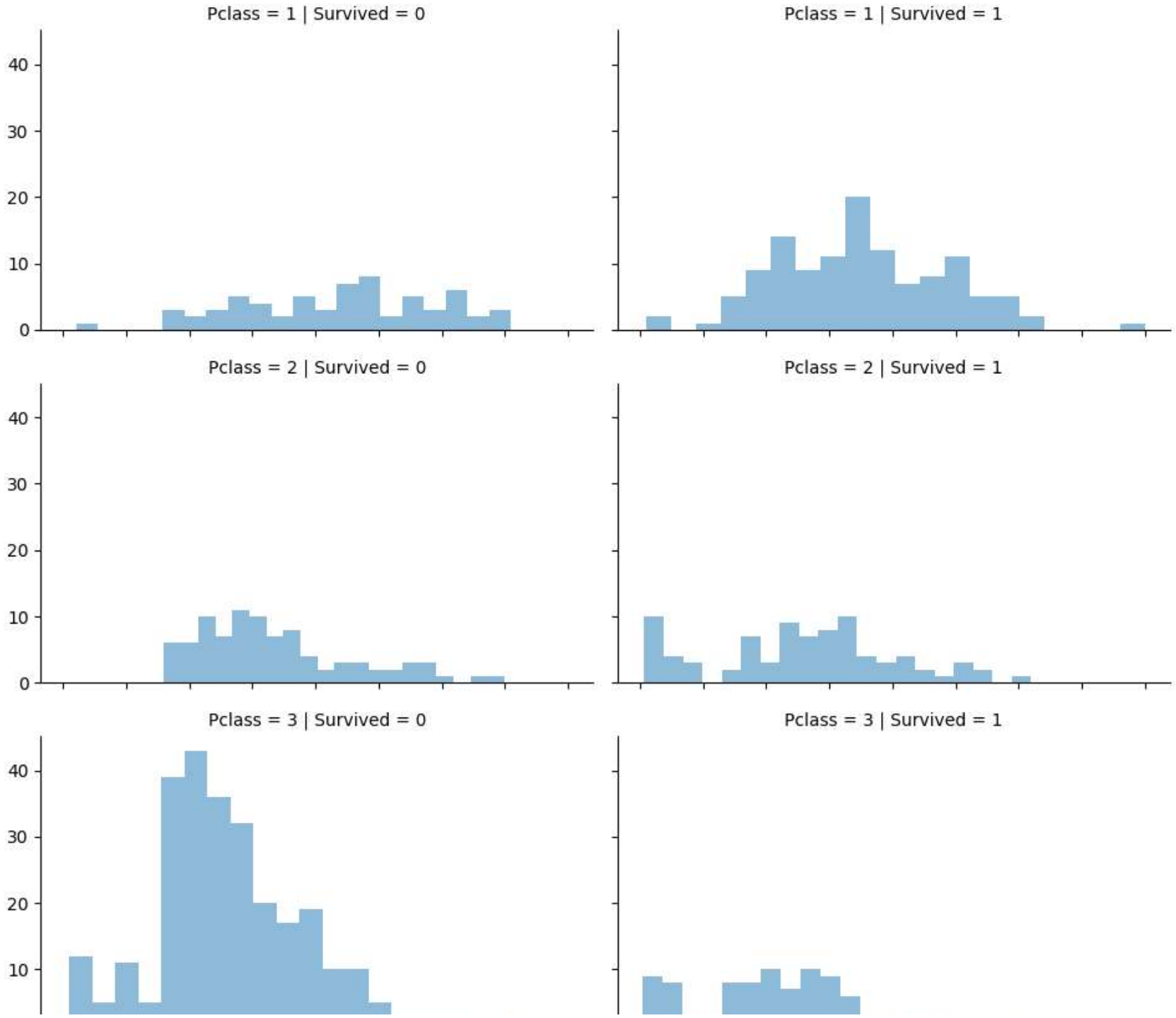
```
C:\Users\Kavisha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The f
igure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

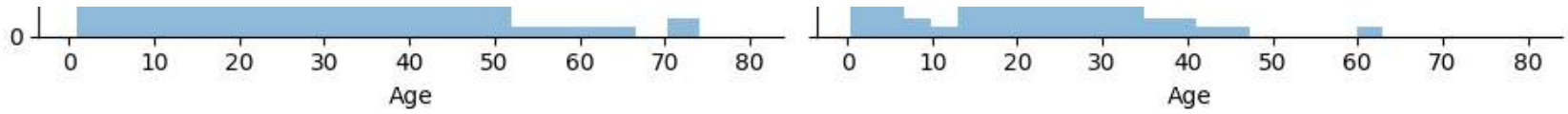
Out[24]: <seaborn.axisgrid.FacetGrid at 0x1beb2a00250>



In [26]:
```
grid = sns.FacetGrid(train_df, col='Survived', row='Pclass', aspect=1.6)
grid.map(plt.hist, 'Age', alpha=.5, bins=20)
grid.add_legend();
```

```
C:\Users\Kavisha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The f
igure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

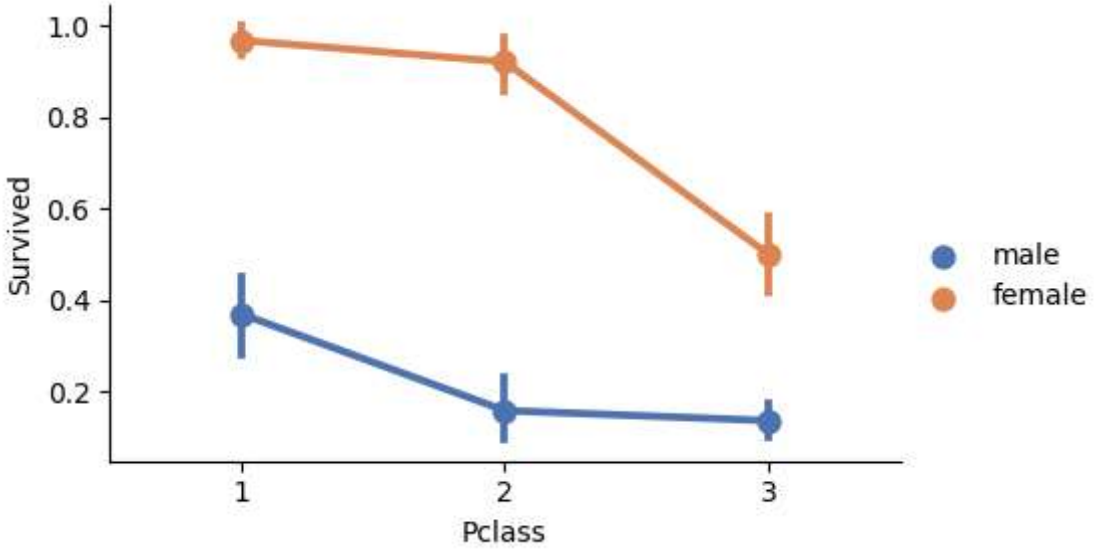

Pclass = 1 | Survived = 0
Pclass = 1 | Survived = 1
Pclass = 2 | Survived = 0
Pclass = 2 | Survived = 1
Pclass = 3 | Survived = 0
Pclass = 3 | Survived = 1
0
10
20
30
40



In [30]:
```
grid = sns.FacetGrid(train_df,   aspect=1.6)
grid.map(sns.pointplot, 'Pclass', 'Survived', 'Sex', palette='deep')
grid.add_legend()
```

```
C:\Users\Kavisha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:712: UserWarning: Using
the pointplot function without specifying `order` is likely to produce an incorrect plot.
  warnings.warn(warning)
C:\Users\Kavisha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:717: UserWarning: Using
the pointplot function without specifying `hue_order` is likely to produce an incorrect plot.
  warnings.warn(warning)
C:\Users\Kavisha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The f
igure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

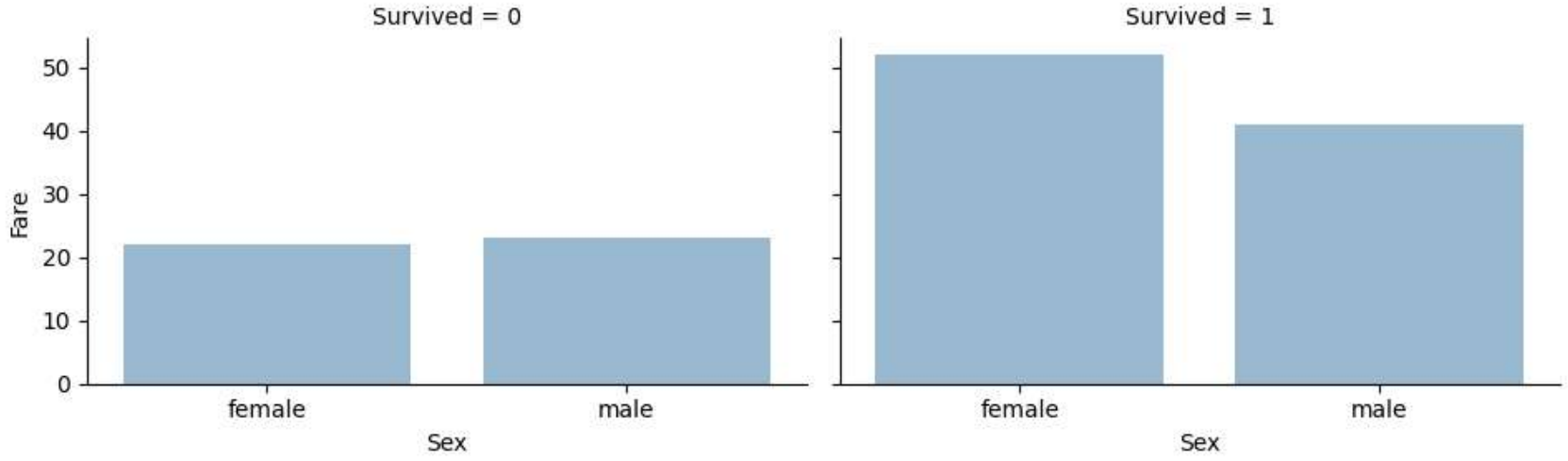
Out[30]: <seaborn.axisgrid.FacetGrid at 0x1beb577d550>



In [32]:
```
grid = sns.FacetGrid(train_df,  col='Survived', aspect=1.6)
grid.map(sns.barplot, 'Sex', 'Fare', alpha=.5, ci=None)
grid.add_legend()
```

```
C:\Users\Kavisha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:712: UserWarning: Using
the barplot function without specifying `order` is likely to produce an incorrect plot.
  warnings.warn(warning)
C:\Users\Kavisha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:848: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  func(*plot_args, **plot_kwargs)
C:\Users\Kavisha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:848: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  func(*plot_args, **plot_kwargs)
C:\Users\Kavisha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The f
igure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[32]: <seaborn.axisgrid.FacetGrid at 0x1beb85a9010>



In [33]:
```
print("Before", train_df.shape, test_df.shape, combine[0].shape, combine[1].shape)

train_df = train_df.drop(['Ticket', 'Cabin'], axis=1)
test_df = test_df.drop(['Ticket', 'Cabin'], axis=1)
combine = [train_df, test_df]

"After", train_df.shape, test_df.shape, combine[0].shape, combine[1].shape
```

```
Before (891, 12) (891, 12) (891, 12) (891, 12)
```

Out[33]:
```
('After', (891, 10), (891, 10), (891, 10), (891, 10))
```

In [34]:
```
for dataset in combine:
    dataset['Title'] = dataset.Name.str.extract(' ([A-Za-z]+)\.', expand=False)

pd.crosstab(train_df['Title'], train_df['Sex'])
```

Out[34]:

Sex	female	male
Title		
Capt	0	1
Col	0	2
Countess	1	0
Don	0	1
Dr	1	6
Jonkheer	0	1
Lady	1	0
Major	0	2
Master	0	40
Miss	182	0
Mlle	2	0
Mme	1	0
Mr	0	517
Mrs	125	0
Ms	1	0
Rev	0	6
Sir	0	1

In [35]:

```
for dataset in combine:
    dataset['Title'] = dataset['Title'].replace(['Lady', 'Countess','Capt', 'Col',\
        'Don', 'Dr', 'Major', 'Rev', 'Sir', 'Jonkheer', 'Dona'], 'Rare')

    dataset['Title'] = dataset['Title'].replace('Mlle', 'Miss')
    dataset['Title'] = dataset['Title'].replace('Ms', 'Miss')
    dataset['Title'] = dataset['Title'].replace('Mme', 'Mrs')
```

```
train_df[['Title', 'Survived']].groupby(['Title'], as_index=False).mean()
```

Out[35]:

	Title	Survived
0	Master	0.575000
1	Miss	0.702703
2	Mr	0.156673
3	Mrs	0.793651
4	Rare	0.347826

In [37]:

```
itle_mapping = {"Mr": 1, "Miss": 2, "Mrs": 3, "Master": 4, "Rare": 5}
for dataset in combine:
    dataset['Title'] = dataset['Title']
    dataset['Title'] = dataset['Title'].fillna(0)

train_df.head(15)
```

Out[37]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked.	Title
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	7.2500	S	Mr
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	71.2833	C	Mrs
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	7.9250	S	Miss
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	53.1000	S	Mrs
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	8.0500	S	Mr
5	6	0	3	Moran, Mr. James	male	NaN	0	0	8.4583	Q	Mr
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	51.8625	S	Mr
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	21.0750	S	Master
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	11.1333	S	Mrs
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	30.0708	C	Mrs
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	16.7000	S	Miss
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	26.5500	S	Miss
12	13	0	3	Saundercock, Mr. William Henry	male	20.0	0	0	8.0500	S	Mr
13	14	0	3	Andersson, Mr. Anders Johan	male	39.0	1	5	31.2750	S	Mr
14	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0	0	0	7.8542	S	Miss

In [38]:
```
train_df = train_df.drop(['Name', 'PassengerId'], axis=1)
test_df = test_df.drop(['Name'], axis=1)
combine = [train_df, test_df]
train_df.shape, test_df.shape
```

Out[38]: ((891, 9), (891, 10))

In [43]:
```
for dataset in combine:
    dataset['Sex'] = dataset['Sex'].map( {'female': 1, 'male': 0} )

train_df.head(15)
```

Out[43]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked.	Title
0	0	3	NaN	22.0	1	0	7.2500	S	Mr
1	1	1	NaN	38.0	1	0	71.2833	C	Mrs
2	1	3	NaN	26.0	0	0	7.9250	S	Miss
3	1	1	NaN	35.0	1	0	53.1000	S	Mrs
4	0	3	NaN	35.0	0	0	8.0500	S	Mr
5	0	3	NaN	NaN	0	0	8.4583	Q	Mr
6	0	1	NaN	54.0	0	0	51.8625	S	Mr
7	0	3	NaN	2.0	3	1	21.0750	S	Master
8	1	3	NaN	27.0	0	2	11.1333	S	Mrs
9	1	2	NaN	14.0	1	0	30.0708	C	Mrs
10	1	3	NaN	4.0	1	1	16.7000	S	Miss
11	1	1	NaN	58.0	0	0	26.5500	S	Miss
12	0	3	NaN	20.0	0	0	8.0500	S	Mr
13	0	3	NaN	39.0	1	5	31.2750	S	Mr
14	0	3	NaN	14.0	0	0	7.8542	S	Miss

In [49]:
```
guess_ages = np.zeros((2,3))
guess_ages
```

Out[49]:
```
array([[0., 0., 0.],
       [0., 0., 0.]])
```

In [53]:
```
for dataset in combine:
    for i in range(0, 2):
        for j in range(0, 3):
            guess_df = dataset[(dataset['Sex'] == i) & \
                               (dataset['Pclass'] == j+1)]['Age'].dropna()

            # age_mean = guess_df.mean()
            # age_std = guess_df.std()
            # age_guess = rnd.uniform(age_mean - age_std, age_mean + age_std)

            age_guess = guess_df.median()



    for i in range(0, 2):
        for j in range(0, 3):
            dataset.loc[ (dataset.Age.isnull()) & (dataset.Sex == i) & (dataset.Pclass == j+1),\
                    'Age'] = guess_ages[i,j]

    dataset['Age'] = dataset['Age']

train_df.head()
```

Out[53]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked.	Title
0	0	3	NaN	22.0	1	0	7.2500	S	Mr
1	1	1	NaN	38.0	1	0	71.2833	C	Mrs
2	1	3	NaN	26.0	0	0	7.9250	S	Miss
3	1	1	NaN	35.0	1	0	53.1000	S	Mrs
4	0	3	NaN	35.0	0	0	8.0500	S	Mr

In [54]:
```
train_df['AgeBand'] = pd.cut(train_df['Age'], 5)
train_df[['AgeBand', 'Survived']].groupby(['AgeBand'], as_index=False).mean().sort_values(by='AgeBand', ascending=Tru
```

Out[54]:

	AgeBand	Survived
0	(0.34, 16.336]	0.550000
1	(16.336, 32.252]	0.369942
2	(32.252, 48.168]	0.404255
3	(48.168, 64.084]	0.434783
4	(64.084, 80.0]	0.090909

In [55]:
```
for dataset in combine:
    dataset.loc[ dataset['Age'] <= 16, 'Age'] = 0
    dataset.loc[(dataset['Age'] > 16) & (dataset['Age'] <= 32), 'Age'] = 1
    dataset.loc[(dataset['Age'] > 32) & (dataset['Age'] <= 48), 'Age'] = 2
    dataset.loc[(dataset['Age'] > 48) & (dataset['Age'] <= 64), 'Age'] = 3
    dataset.loc[ dataset['Age'] > 64, 'Age']
train_df.head()
```

Out[55]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked.	Title	AgeBand
0	0	3	NaN	1.0	1	0	7.2500	S	Mr	(16.336, 32.252]
1	1	1	NaN	2.0	1	0	71.2833	C	Mrs	(32.252, 48.168]
2	1	3	NaN	1.0	0	0	7.9250	S	Miss	(16.336, 32.252]
3	1	1	NaN	2.0	1	0	53.1000	S	Mrs	(32.252, 48.168]
4	0	3	NaN	2.0	0	0	8.0500	S	Mr	(32.252, 48.168]

In [56]:
```
for dataset in combine:
    dataset['FamilySize'] = dataset['SibSp'] + dataset['Parch'] + 1

train_df[['FamilySize', 'Survived']].groupby(['FamilySize'], as_index=False).mean().sort_values(by='Survived', ascend
```

Out[56]:

	FamilySize	Survived
3	4	0.724138
2	3	0.578431
1	2	0.552795
6	7	0.333333
0	1	0.303538
4	5	0.200000
5	6	0.136364
7	8	0.000000
8	11	0.000000

In [57]:
```
for dataset in combine:
    dataset['IsAlone'] = 0
    dataset.loc[dataset['FamilySize'] == 1, 'IsAlone'] = 1

train_df[['IsAlone', 'Survived']].groupby(['IsAlone'], as_index=False).mean()
```

Out[57]:

	IsAlone	Survived
0	0	0.505650
1	1	0.303538

In [58]:
```
X_train = train_df.drop("Survived", axis=1)
Y_train = train_df["Survived"]
X_test  = test_df.drop("PassengerId", axis=1).copy()
X_train.shape, Y_train.shape, X_test.shape
```

Out[58]:
```
((891, 11), (891,), (891, 11))
```

In [ ]:
```

```