untitled9

May 7, 2024

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

[10]:
```
df = sns.load_dataset('titanic')
```

[11]:
```
df.isnull().sum()
```

[11]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

[12]:
```
print(df['age'].mode())
print(df['embark_town'].mode())
print(df['embarked'].mode())
```

```
0    24.0
Name: age, dtype: float64
0    Southampton
Name: embark_town, dtype: object
0    S
Name: embarked, dtype: object
```

[13]:
```
df['age'].fillna(value=24,inplace=True)
df['embark_town'].fillna(value='Southampton',inplace=True)
df['embarked'].fillna(value='S',inplace=True)
```

```
C:\Users\PUSHKAR\AppData\Local\Temp\ipykernel_22512\2395318720.py:1:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  df['age'].fillna(value=24,inplace=True)
C:\Users\PUSHKAR\AppData\Local\Temp\ipykernel_22512\2395318720.py:2:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  df['embark_town'].fillna(value='Southampton',inplace=True)
C:\Users\PUSHKAR\AppData\Local\Temp\ipykernel_22512\2395318720.py:3:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  df['embarked'].fillna(value='S',inplace=True)
```

[14]:
```
df.isnull().sum()
```

[14]:
```
survived         0
pclass           0
sex              0
age              0
sibsp            0
parch            0
fare             0
embarked         0
class            0
who              0
adult_male       0
deck           688
embark_town      0
alive            0
alone            0
dtype: int64
```

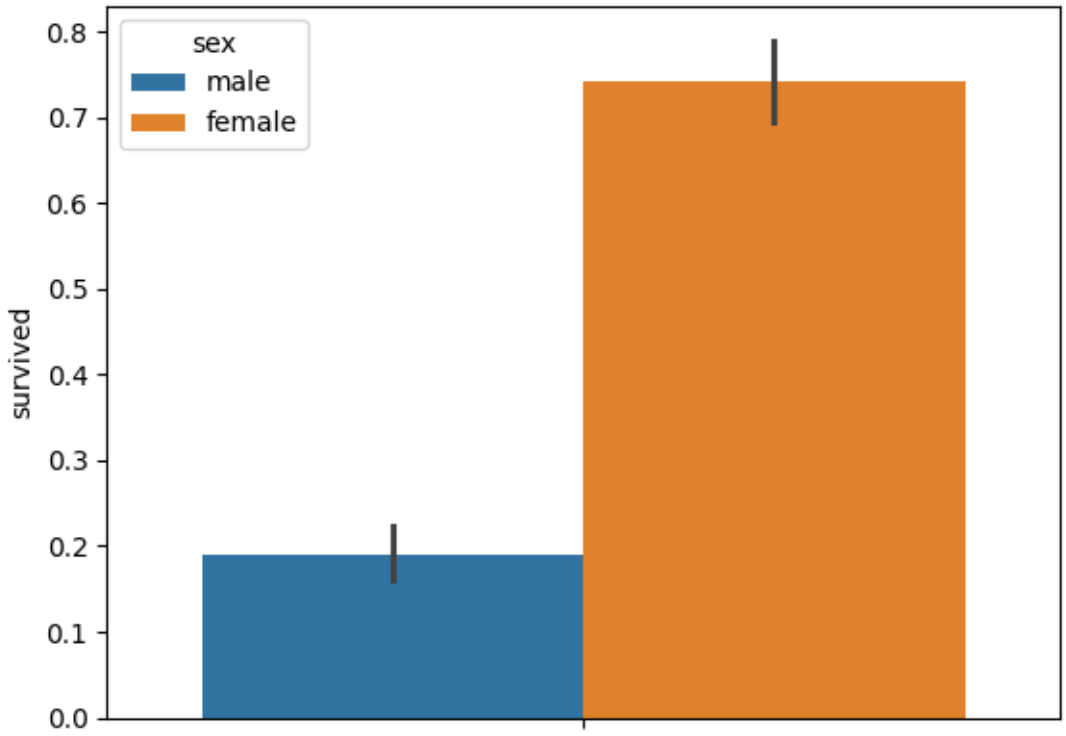
[15]:
```
df = df.drop(['deck'],axis=1)
```

[16]:
```
df.isnull().sum()
```

[16]:
```
survived       0
pclass         0
sex            0
age            0
sibsp          0
parch          0
fare           0
embarked       0
class          0
who            0
adult_male     0
embark_town    0
alive          0
alone          0
dtype: int64
```

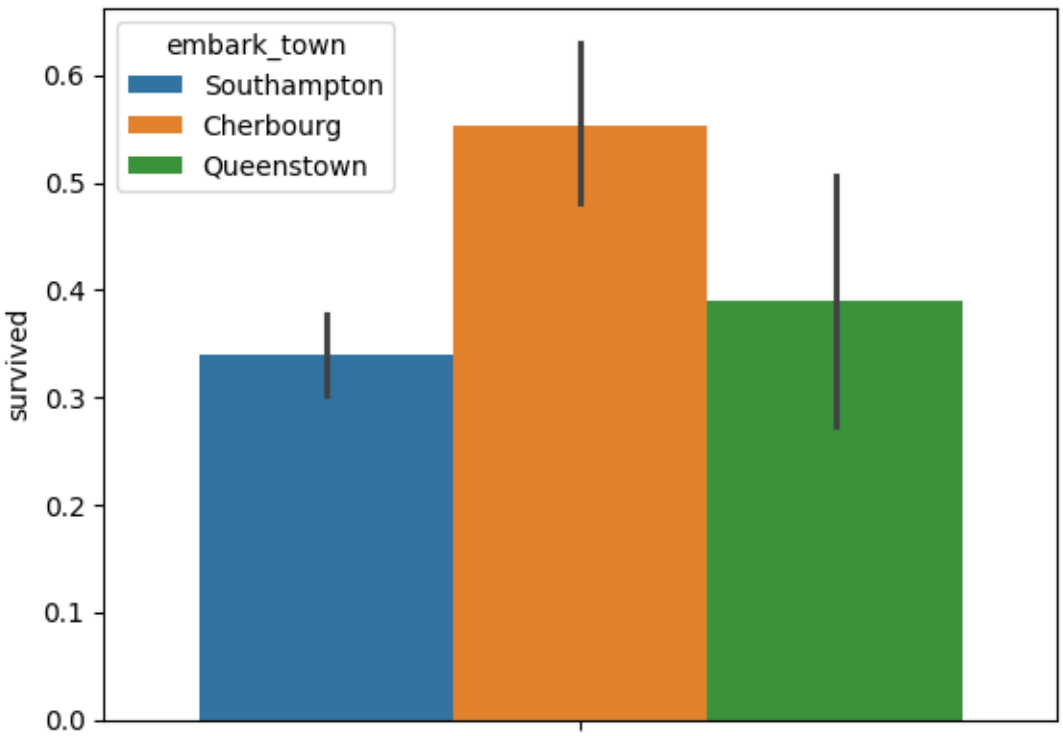
[23]:
```
custom_palette=['blue','orange']
sns.barplot(hue='sex',y='survived',data=df,)
```

[23]:
```
<Axes: ylabel='survived'>
```



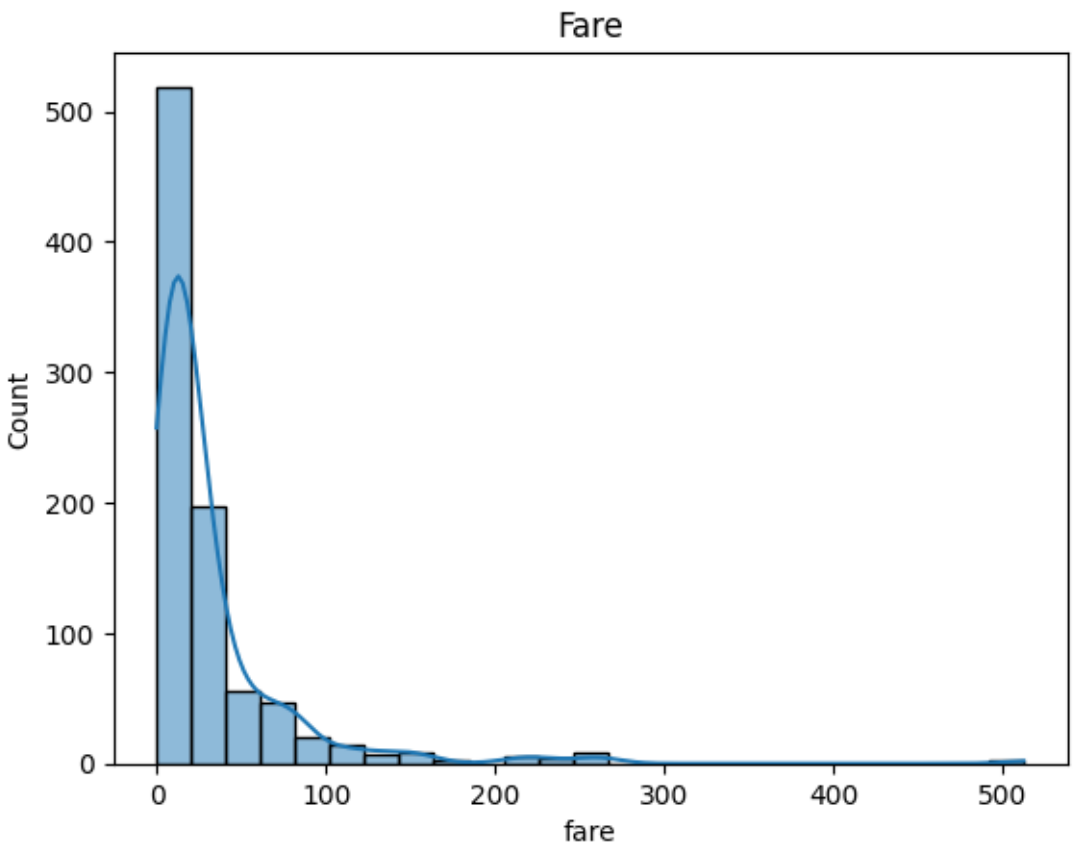
[26]:
```
sns.barplot(hue='embark_town',y='survived',data=df)
```

[26]: <Axes: ylabel='survived'>



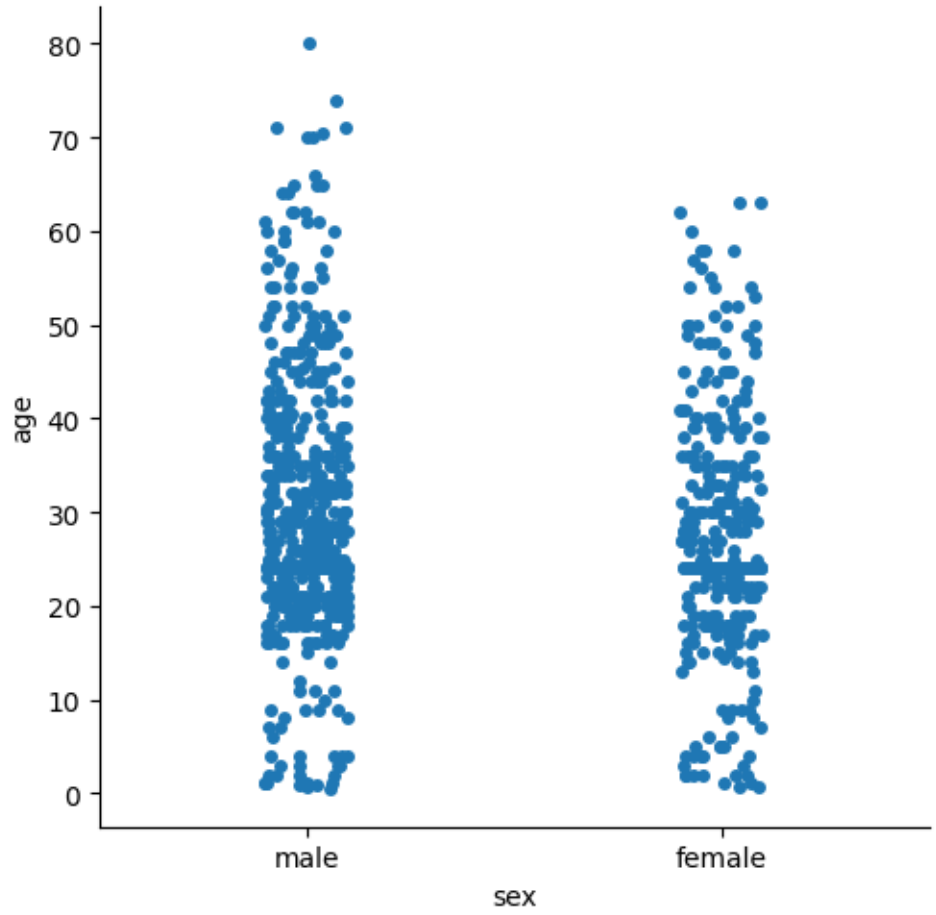
[30]:
```
sns.histplot(data=df['fare'],bins=25,kde=True)
plt.title('Fare')
```

[30]:
```
Text(0.5, 1.0, 'Fare')
```



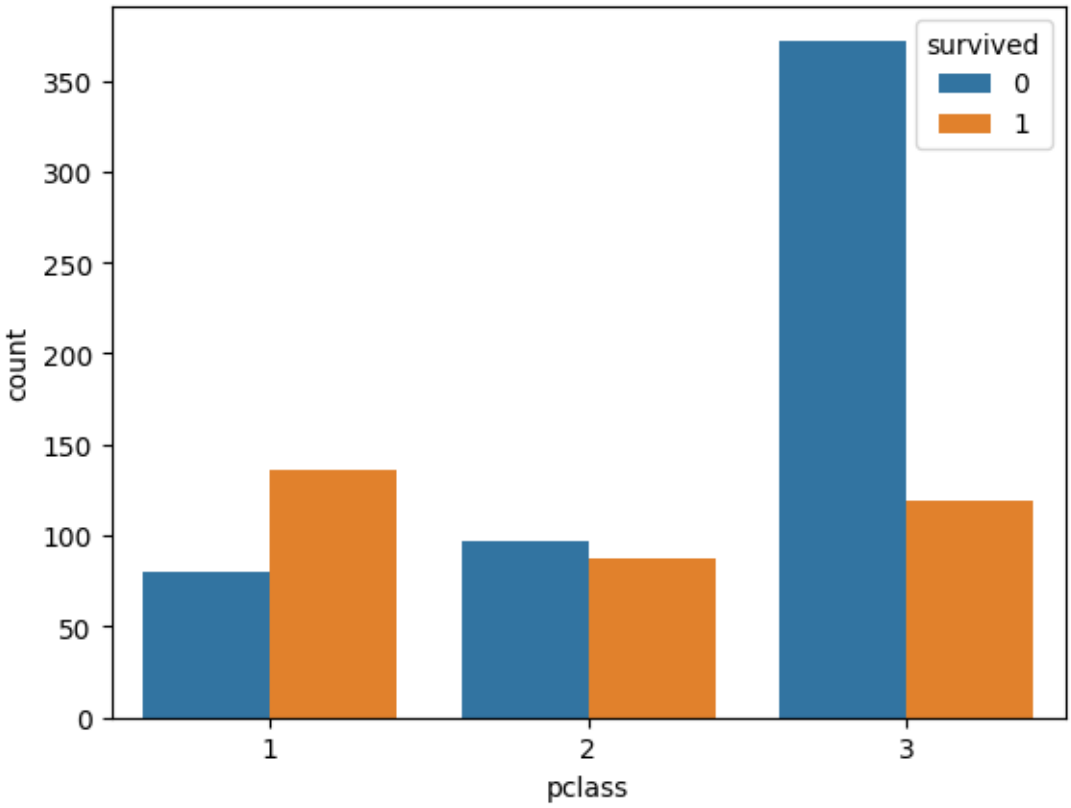
[33]:
```
sns.catplot(x='sex',y='age',data=df)
```

[33]: <seaborn.axisgrid.FacetGrid at 0x2c54d510a40>



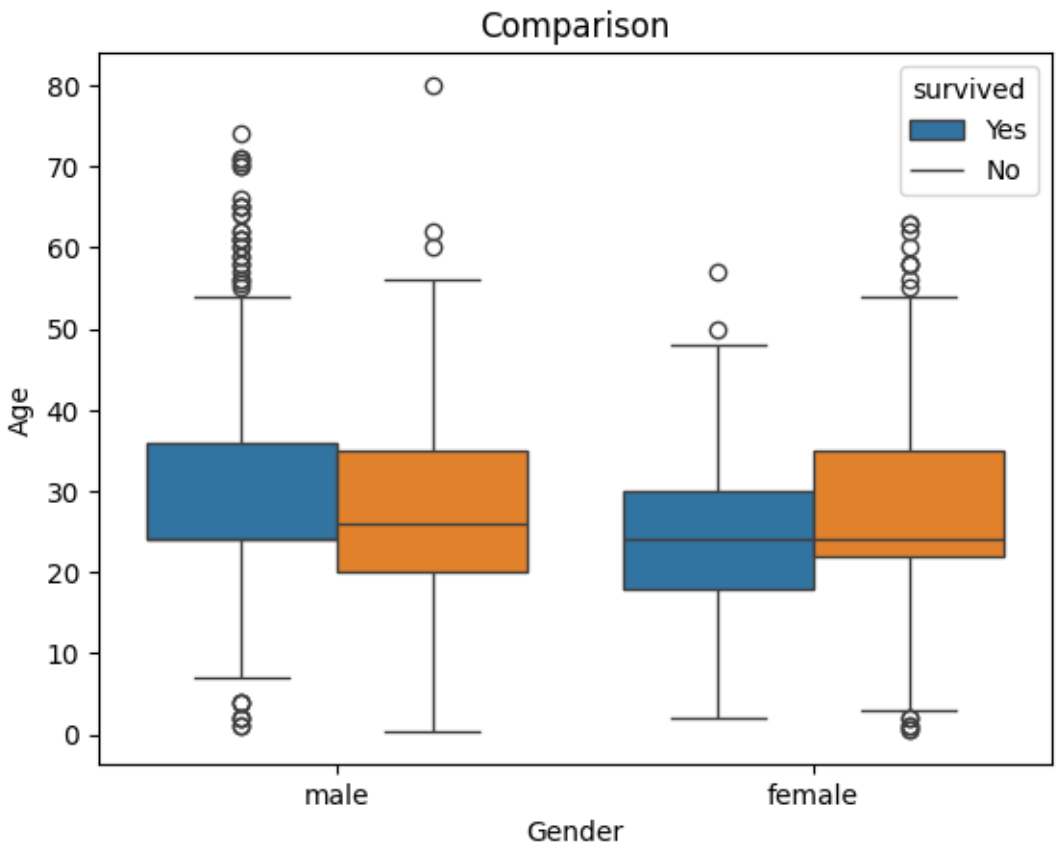
[35]:
```
sns.countplot(x='pclass',hue='survived',data=df)
```

[35]: <Axes: xlabel='pclass', ylabel='count'>



[43]:
```
sns.boxplot(x='sex',y='age',hue='survived',data=df)
plt.legend(title='survived',labels=['Yes','No'])
plt.xlabel('Gender')
plt.ylabel('Age')
plt.title('Comparison')
```

[43]: Text(0.5, 1.0, 'Comparison')


Comparison
survived
Yes
No
Age
80
70
60
50
40
30
20
10
0
male
female
Gender

[ ]: