```python
import pandas as pd
import numpy as np
from pandas import Series,DataFrame
import matplotlib.pyplot as plt
import seaborn as sns

titanic_df= pd.read_csv('/train.csv')

titanic_df.head()
```

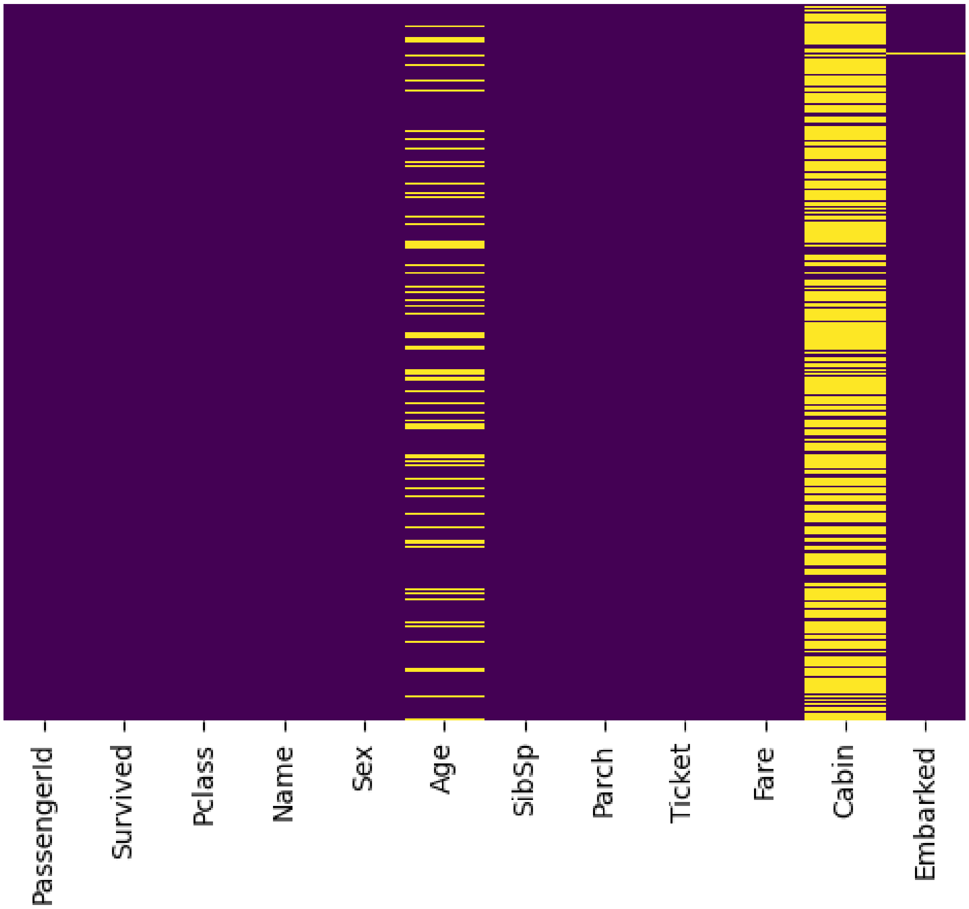
```
{"summary":"{\n  \"name\": \"titanic_df\",\n  \"rows\": 891,\n
\"fields\": [\n    {\n      \"column\": \"PassengerId\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
257,\n        \"min\": 1,\n        \"max\": 891,\n
\"num_unique_values\": 891,\n        \"samples\": [\n          710,\n
440,\n          841\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Survived\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Pclass\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 1,\n
\"max\": 3,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          3,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Name\",\n      \"properties\": {\n        \"dtype\":
\"string\",\n        \"num_unique_values\": 891,\n        \"samples\":
[\n          \"Moubarek, Master. Halim Gonios (\\\"William
George\\\")\",\n          \"Kvillner, Mr. Johan Henrik Johannesson\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Sex\",\n      \"properties\": {\
n        \"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"female\",\n          \"male\"\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Age\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 14.526497332334042,\
n        \"min\": 0.42,\n        \"max\": 80.0,\n
\"num_unique_values\": 88,\n        \"samples\": [\n          0.75,\n
22.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"SibSp\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 1,\n        \"min\": 0,\n        \"max\": 8,\n
\"num_unique_values\": 7,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Parch\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0,\n        \"min\": 0,\n        \"max\": 6,\n
\"num_unique_values\": 7,\n        \"samples\": [\n          0,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Ticket\",\n      \"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 681,\n        \"samples\": [\n
\"11774\",\n          \"248740\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Fare\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 49.6934285971809,\n
\"min\": 0.0,\n        \"max\": 512.3292,\n
\"num_unique_values\": 248,\n        \"samples\": [\n
11.2417,\n          51.8625\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Cabin\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 147,\n
\"samples\": [\n          \"D45\",\n          \"B49\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Embarked\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
3,\n        \"samples\": [\n          \"S\",\n          \"C\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"titanic_df"}
```

```
titanic_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
```

```
sns.heatmap(titanic_df.isnull(),yticklabels=False,cbar=False,cmap='vir
idis')
```

```
<Axes: >
```



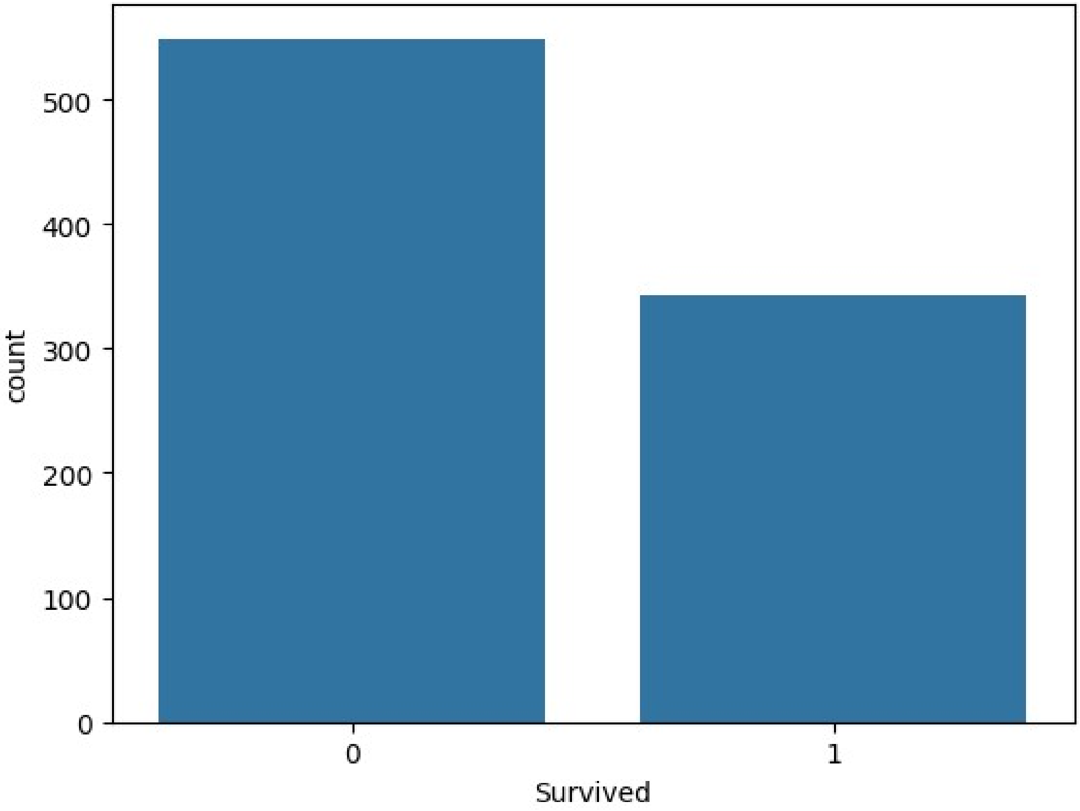
```
titanic_df.describe()
```

```
{"summary":"{\n  \"name\": \"titanic_df\",\n  \"rows\": 8,\n
\"fields\": [\n    {\n      \"column\": \"PassengerId\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
320.8159711429855,\n        \"min\": 1.0,\n        \"max\": 891.0,\n
\"num_unique_values\": 6,\n        \"samples\": [\n          891.0,\n
446.0,\n          668.5\n        ],\n        \"semantic_type\": \"\",\
n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Survived\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 314.8713661874558,\n        \"min\":
0.0,\n        \"max\": 891.0,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          0.3838383838383838,\n          1.0,\n
0.4865924542648575\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Pclass\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 314.2523437079694,\n        \"min\": 0.836071240977049,\n
\"max\": 891.0,\n        \"num_unique_values\": 6,\n
\"samples\": [\n          891.0,\n          2.308641975308642,\n
3.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
```

```
\"Age\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 242.9056731818781,\n        \"min\": 0.42,\n        \"max\":
714.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
29.69911764705882,\n          28.0,\n          714.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"SibSp\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 314.4908277465442,\n
\"min\": 0.0,\n        \"max\": 891.0,\n        \"num_unique_values\":
6,\n        \"samples\": [\n          891.0,\n
0.5230078563411896,\n          8.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Parch\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 314.65971717879,\n
\"min\": 0.0,\n        \"max\": 891.0,\n        \"num_unique_values\":
5,\n        \"samples\": [\n          0.38159371492704824,\n
6.0,\n          0.8060572211299483\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Fare\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 330.6256632228578,\n
\"min\": 0.0,\n        \"max\": 891.0,\n        \"num_unique_values\":
8,\n        \"samples\": [\n          32.204207968574636,\n
14.4542,\n          891.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe"}
```

```
sns.countplot(x='Survived',data=titanic_df)
```

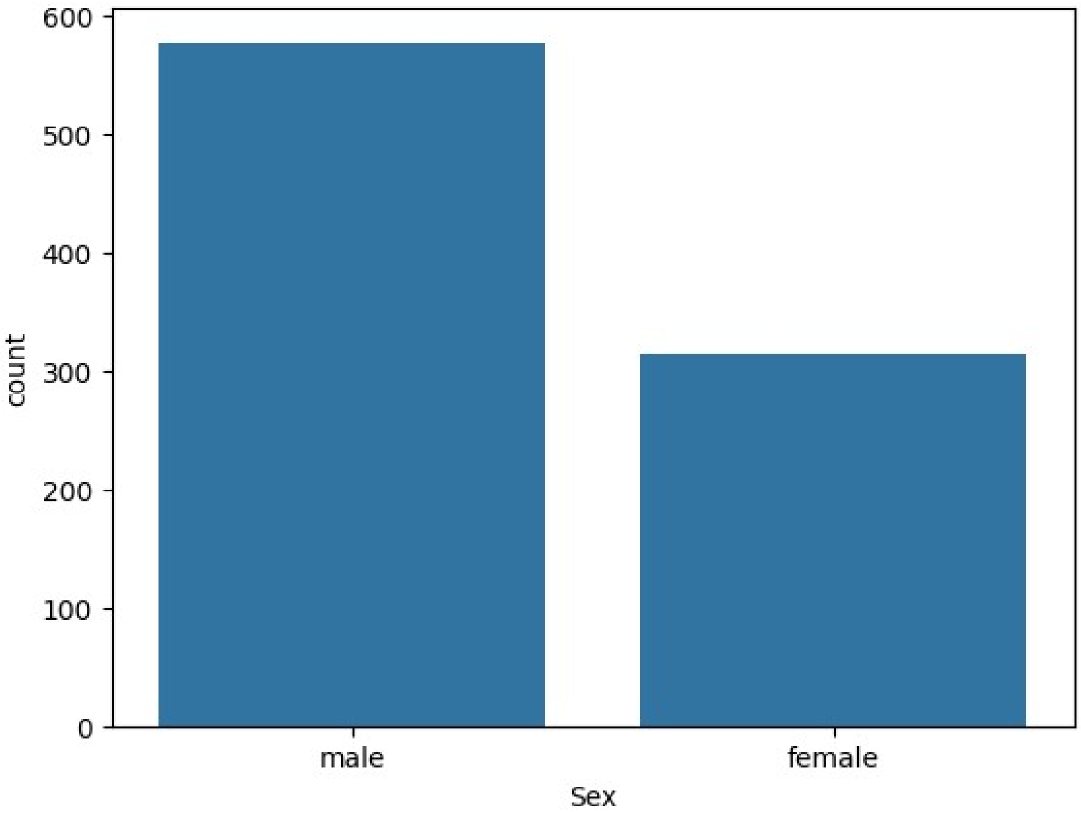
```
<Axes: xlabel='Survived', ylabel='count'>
```



1 survive and 0 not servived

```
sns.countplot(x='Sex',data=titanic_df)
```

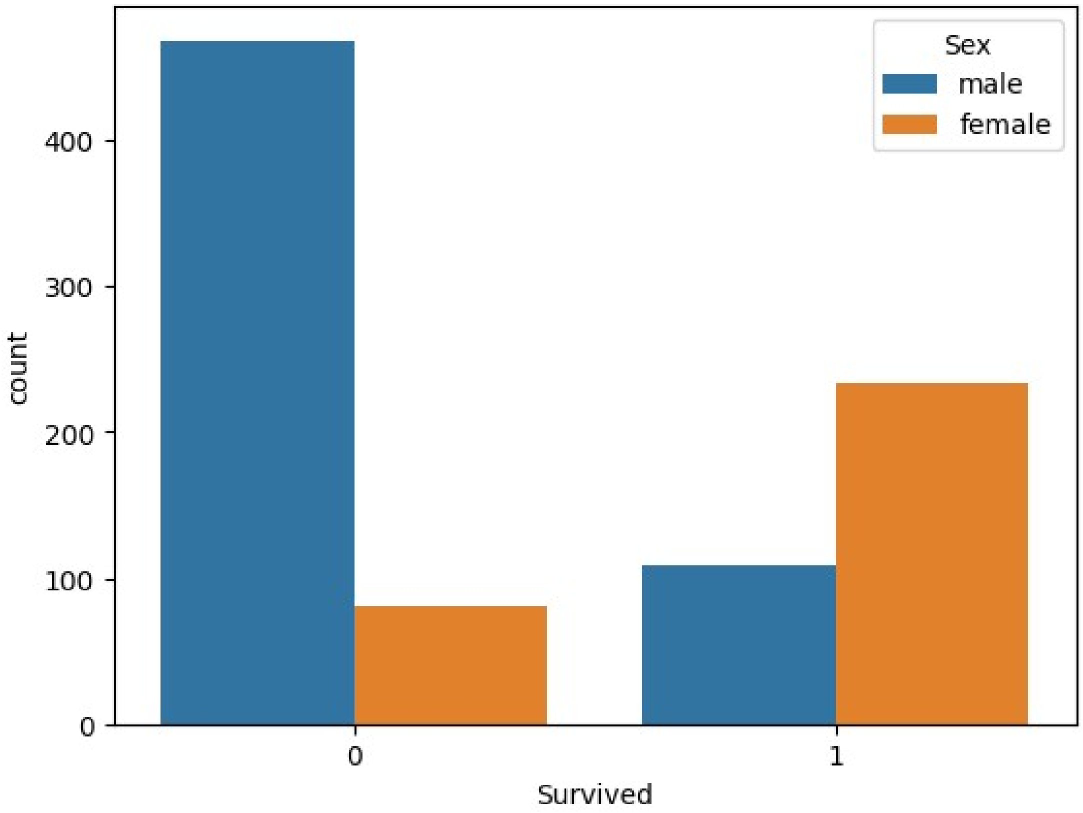
```
<Axes: xlabel='Sex', ylabel='count'>
```



male around 600 and female 300

```
sns.countplot(x='Survived',hue='Sex',data=titanic_df)
```

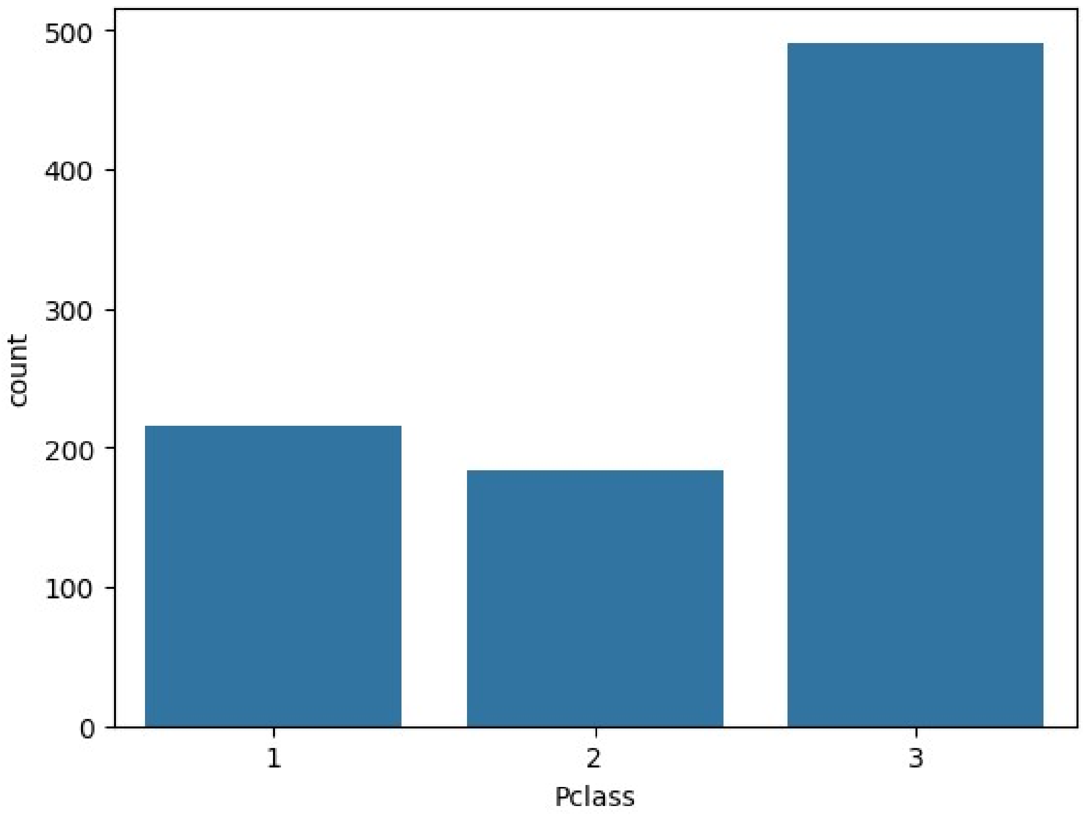
```
<Axes: xlabel='Survived', ylabel='count'>
```



female and child servived are more

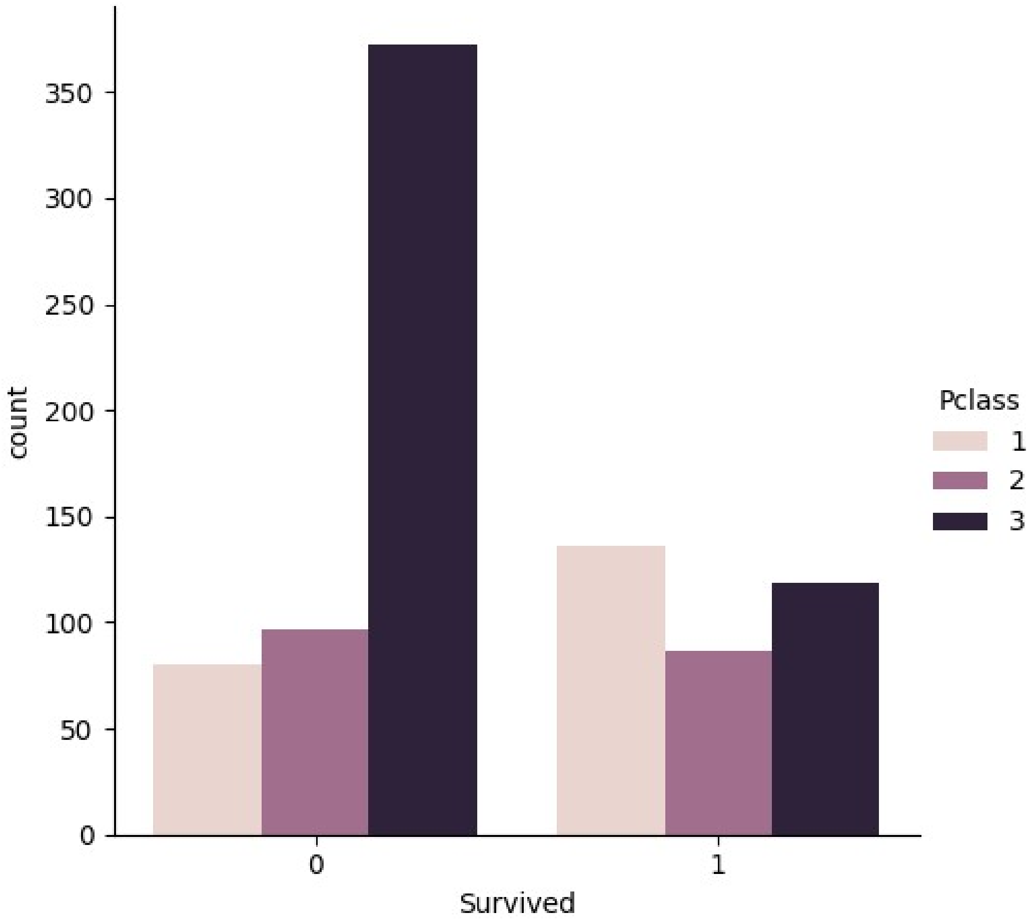
```
sns.countplot(x='Pclass',data=titanic_df)
```

```
<Axes: xlabel='Pclass', ylabel='count'>
```



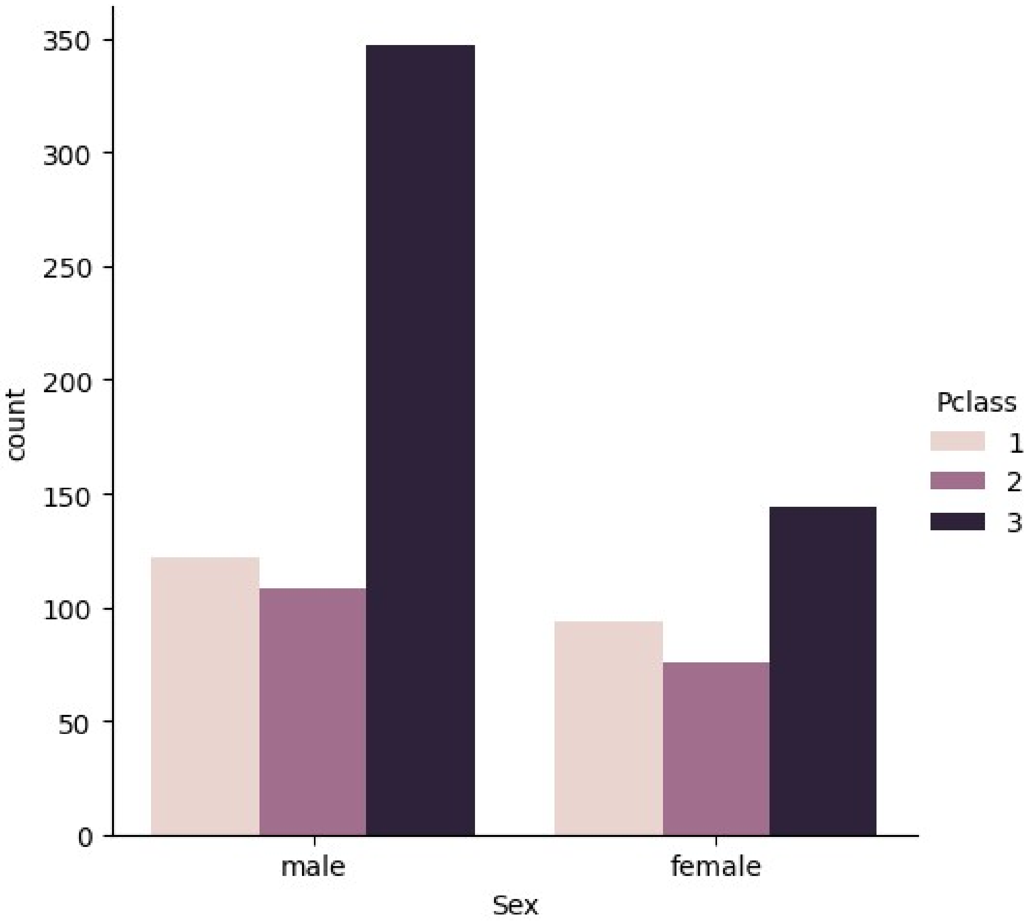
```
sns.catplot(x='Survived',data=titanic_df,hue='Pclass',kind='count')
```

```
<seaborn.axisgrid.FacetGrid at 0x7ae1141fff40>
```



3 rd lower servival

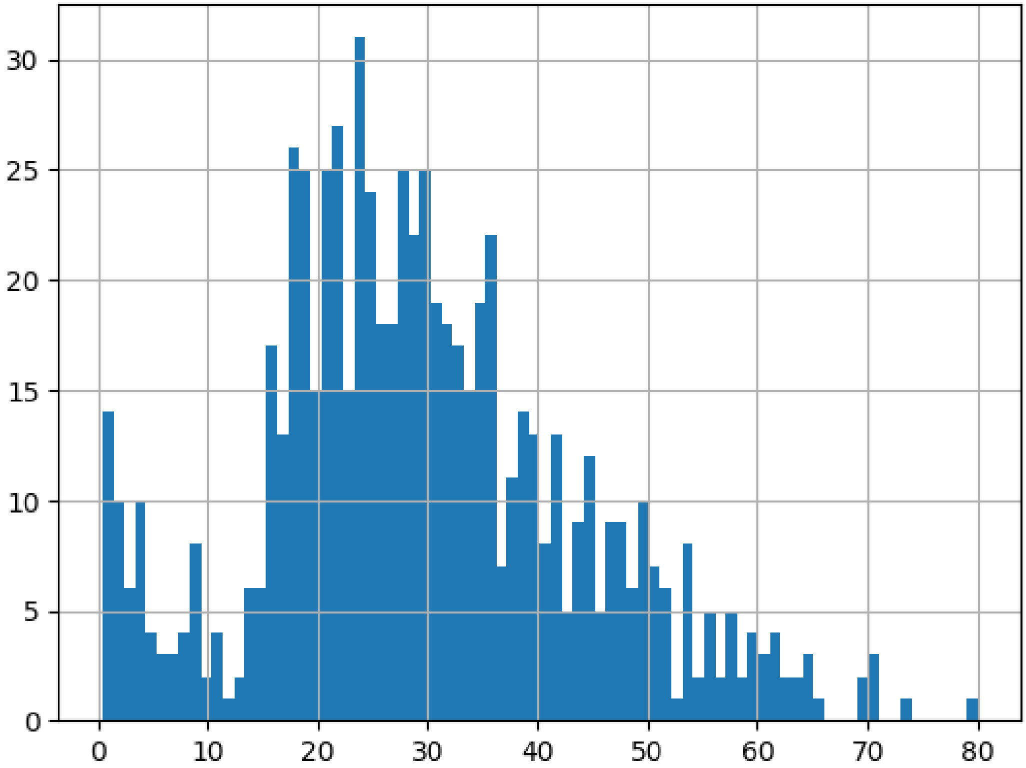
```
sns.catplot(x='Sex',data=titanic_df,hue='Pclass',kind='count')

<seaborn.axisgrid.FacetGrid at 0x7ae1138492a0>
```



female or childern where given prefrences

```
titanic_df['Age'].hist(bins=80)
```

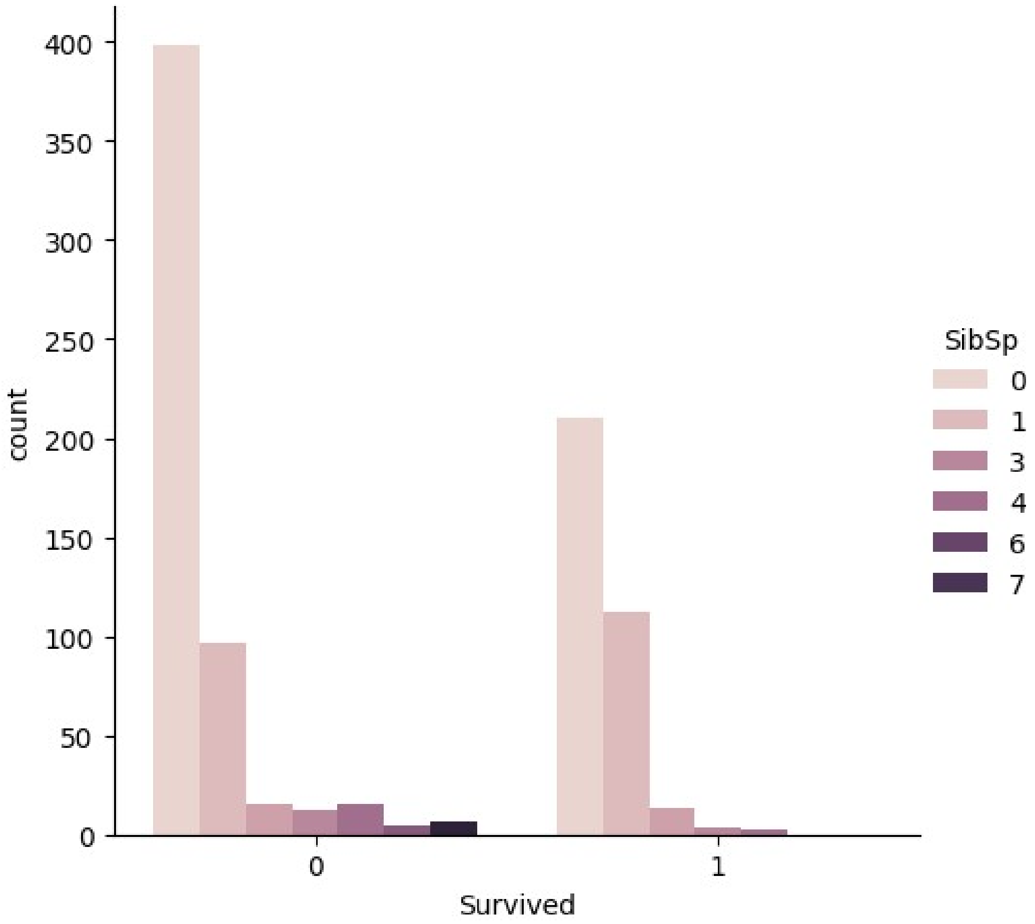
```
<Axes: >
```



how many younger and older where their in the ship ,young people where in the ship

```
sns.catplot(x='Survived',data=titanic_df,hue='SibSp',kind='count')
```

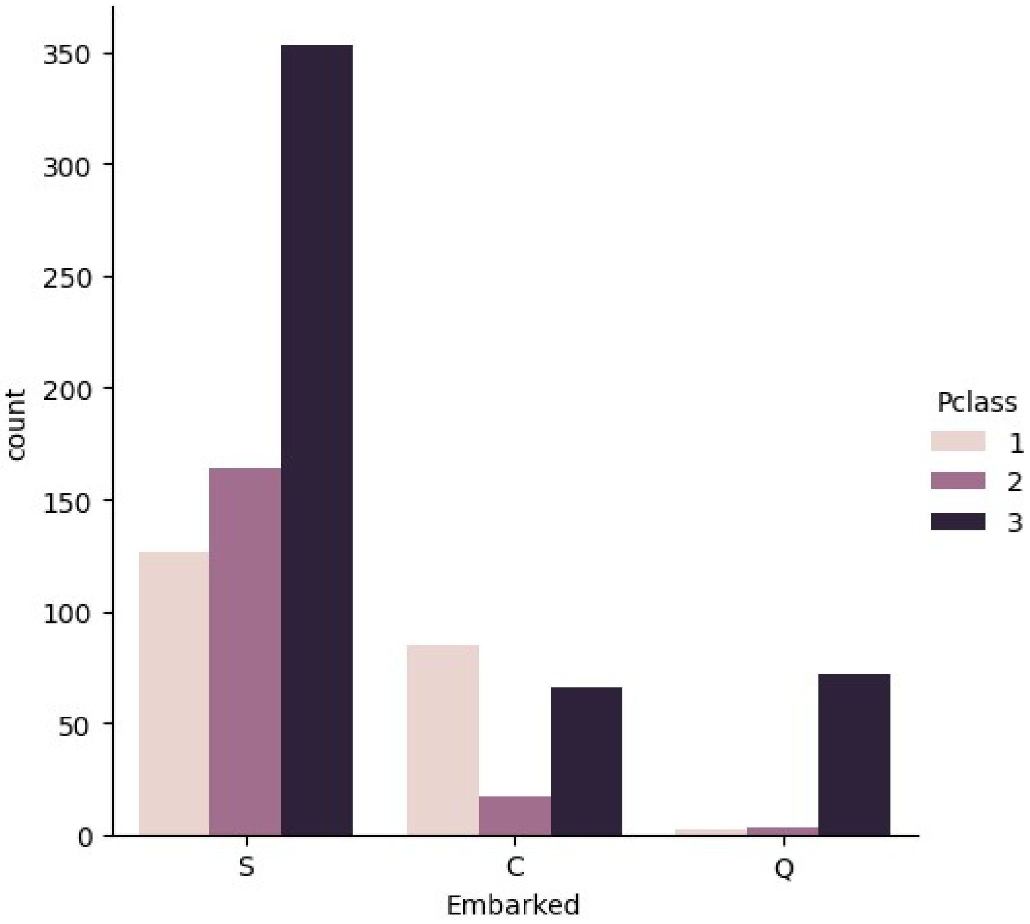
```
<seaborn.axisgrid.FacetGrid at 0x7ae113777730>
```



if their a wife or sibling the chance of the servival is shown

```
sns.catplot(x='Embarked',data=titanic_df,hue='Pclass',kind='count')
```

```
<seaborn.axisgrid.FacetGrid at 0x7ae113777a30>
```



queens town - majority of the people had a sit in 3 rd class

```
titanic_df['Age'].mean()

29.69911764705882
```

data cleaning

```
titanic_df.groupby(by='Pclass')['Age'].mean()

Pclass
1    38.233441
2    29.877630
3    25.140620
Name: Age, dtype: float64
```

pclass group by the class

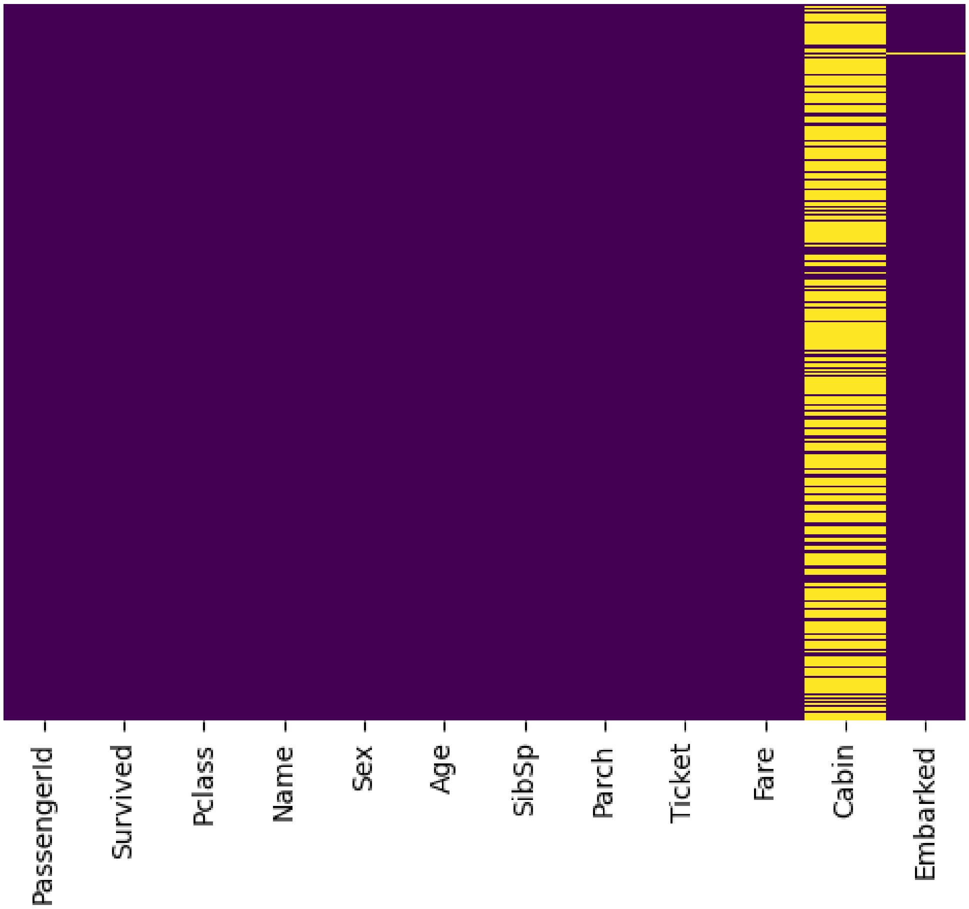
```
def m_age(c):
  Age=c[0]
  Pclass=c[1]
  if pd.isnull(Age):
    if Pclass==1:
      return 38
    elif Pclass==2:
        return 29
    else:
      return 25
  else:
      return(Age)
```

```
titanic_df['Age']=titanic_df[['Age','Pclass']].apply(m_age,axis=1)
```

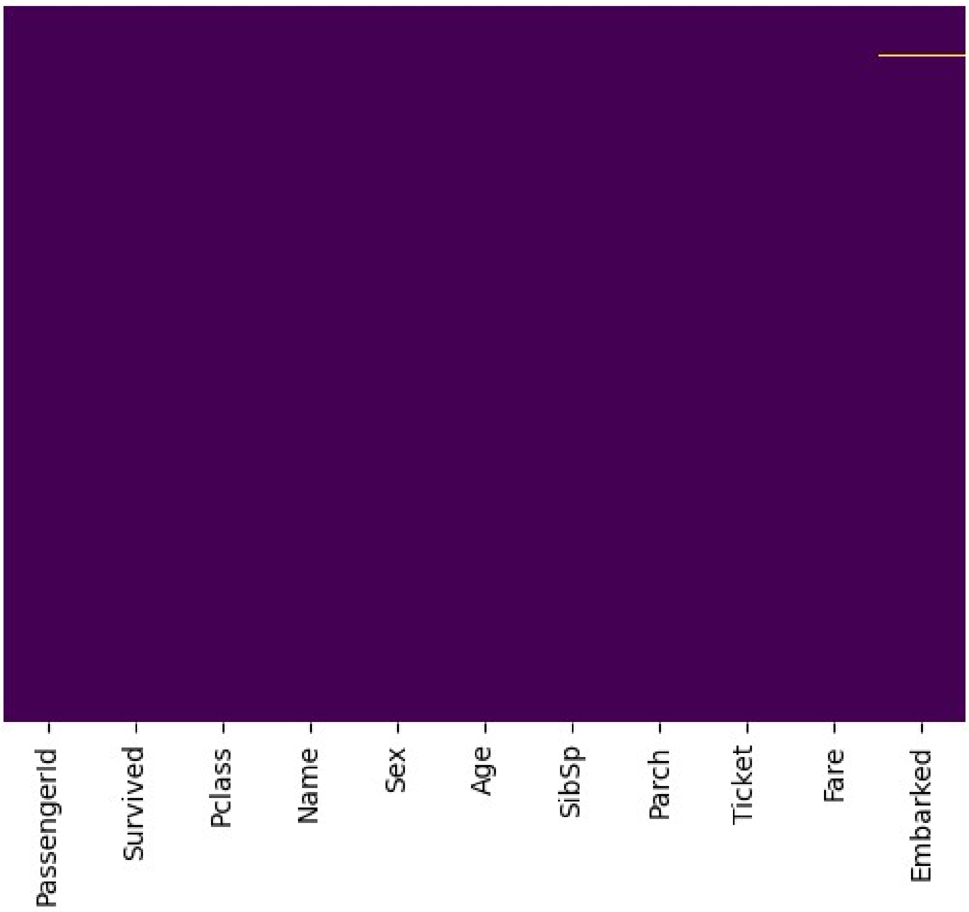
```
<ipython-input-21-d280b1b4ca89>:2: FutureWarning: Series.__getitem__
treating keys as positions is deprecated. In a future version, integer
keys will always be treated as labels (consistent with DataFrame
behavior). To access a value by position, use `ser.iloc[pos]`
  Age=c[0]
<ipython-input-21-d280b1b4ca89>:3: FutureWarning: Series.__getitem__
treating keys as positions is deprecated. In a future version, integer
keys will always be treated as labels (consistent with DataFrame
behavior). To access a value by position, use `ser.iloc[pos]`
  Pclass=c[1]
```

```
sns.heatmap(titanic_df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

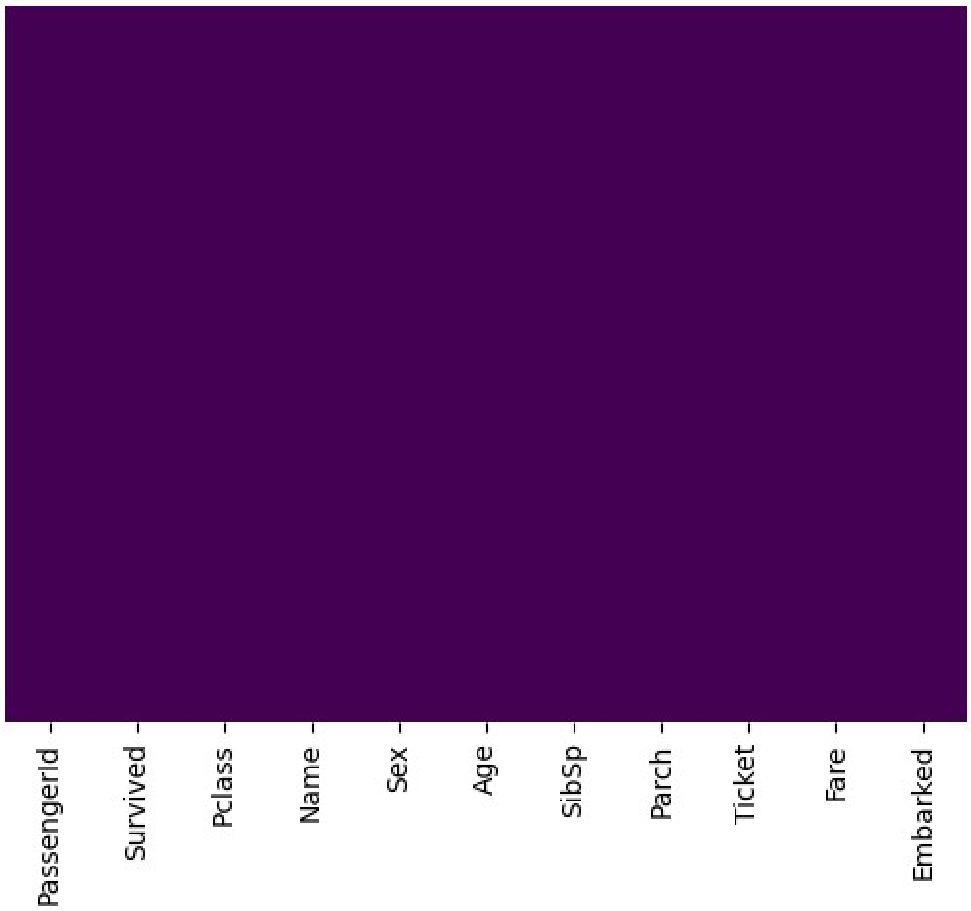
```
<Axes: >
```



```
titanic_df.drop('Cabin',axis=1,inplace=True)

sns.heatmap(titanic_df.isnull(),yticklabels=False,cbar=False,cmap='viridis')

<Axes: >
```



```
titanic_df=titanic_df.dropna()

sns.heatmap(titanic_df.isnull(),yticklabels=False,cbar=False,cmap='viridis')

<Axes: >
```



```
sex=pd.get_dummies(titanic_df['Sex'],drop_first=True)
```

```
sex
```

```
{"summary":"{\n  \"name\": \"sex\",\n  \"rows\": 889,\n  \"fields\":
[\n    {\n      \"column\": \"male\",\n      \"properties\": {\n
\"dtype\": \"boolean\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          false,\n          true\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"sex"}
```

```
embark=pd.get_dummies(titanic_df['Embarked'],drop_first=True)
```

```
titanic_df.drop(['Sex','Embarked','Name','Ticket'],axis=1,inplace=True
)
```

```
<ipython-input-32-c43b8843b89f>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
```

```
titanic_df.drop(['Sex','Embarked','Name','Ticket'],axis=1,inplace=True
)
```

in one column the data is represent

convert female and male to 0 and 1 embarked also as in numeric,drop the name ,drop the ticket

```
embark
```

```
{"summary":"{\n  \"name\": \"embark\",\n  \"rows\": 889,\n
\"fields\": [\n    {\n      \"column\": \"Q\",\n      \"properties\":
{\n        \"dtype\": \"boolean\",\n        \"num_unique_values\": 2,\
n        \"samples\": [\n          true,\n          false\n        ],\
n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"S\",\n      \"properties\": {\n
\"dtype\": \"boolean\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          false,\n          true\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"embark"}
```

```
titanic_df.head()
```

```
{"summary":"{\n  \"name\": \"titanic_df\",\n  \"rows\": 889,\n
\"fields\": [\n    {\n      \"column\": \"PassengerId\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
256,\n        \"min\": 1,\n        \"max\": 891,\n
\"num_unique_values\": 889,\n        \"samples\": [\n          282,\n
436,\n          40\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Survived\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Pclass\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 1,\n
\"max\": 3,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          3,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Age\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 13.177746823022957,\n        \"min\":
0.42,\n        \"max\": 80.0,\n        \"num_unique_values\": 88,\n
\"samples\": [\n          0.75,\n          22.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"SibSp\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 1,\n        \"min\":
0,\n        \"max\": 8,\n        \"num_unique_values\": 7,\n
\"samples\": [\n          1,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Parch\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 0,\n        \"min\":
0,\n        \"max\": 6,\n        \"num_unique_values\": 7,\n
\"samples\": [\n          0,\n          1\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Fare\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 49.697504316707956,\n
\"min\": 0.0,\n        \"max\": 512.3292,\n
\"num_unique_values\": 247,\n        \"samples\": [\n
11.2417,\n          51.8625\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"titanic_df"}
```

```
titanic_df=pd.concat([titanic_df,sex,embark],axis=1)
```

```
titanic_df
```

```
{"summary":"{\n  \"name\": \"titanic_df\",\n  \"rows\": 889,\n
\"fields\": [\n    {\n      \"column\": \"PassengerId\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
256,\n        \"min\": 1,\n        \"max\": 891,\n
\"num_unique_values\": 889,\n        \"samples\": [\n          282,\n
436,\n          40\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Survived\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Pclass\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 1,\n
\"max\": 3,\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          3,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Age\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 13.177746823022957,\n        \"min\":
0.42,\n        \"max\": 80.0,\n        \"num_unique_values\": 88,\n
\"samples\": [\n          0.75,\n          22.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"SibSp\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 1,\n        \"min\":
0,\n        \"max\": 8,\n        \"num_unique_values\": 7,\n
\"samples\": [\n          1,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Parch\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 0,\n        \"min\":
0,\n        \"max\": 6,\n        \"num_unique_values\": 7,\n
\"samples\": [\n          0,\n          1\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Fare\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 49.697504316707956,\n
\"min\": 0.0,\n        \"max\": 512.3292,\n
\"num_unique_values\": 247,\n        \"samples\": [\n
11.2417,\n          51.8625\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"male\",\n      \"properties\": {\n        \"dtype\":
\"boolean\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          false,\n          true\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Q\",\n      \"properties\": {\n
\"dtype\": \"boolean\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          true,\n          false\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"S\",\n      \"properties\": {\n
\"dtype\": \"boolean\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          false,\n          true\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"titanic_df"}
```