1.import the necessary libraries

[ ] ↳ *1 cell hidden*

2.import the dataset

```
dataset=pd.read_csv("tested.csv")
```

```
dataset
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ti
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	3
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	3
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	2
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	3
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	31

```
dataset.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272

```
dataset.tail()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ti
413	1305	0	3	Spector, Mr. Woolf	male	NaN	0	0	A.5.
414	1306	1	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 1

```
dataset.shape
```

```
(418, 12)
```

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

```
dataset.describe()
```

```
     PassengerId  Survived  Pclass    Age   SibSp   Parch
```

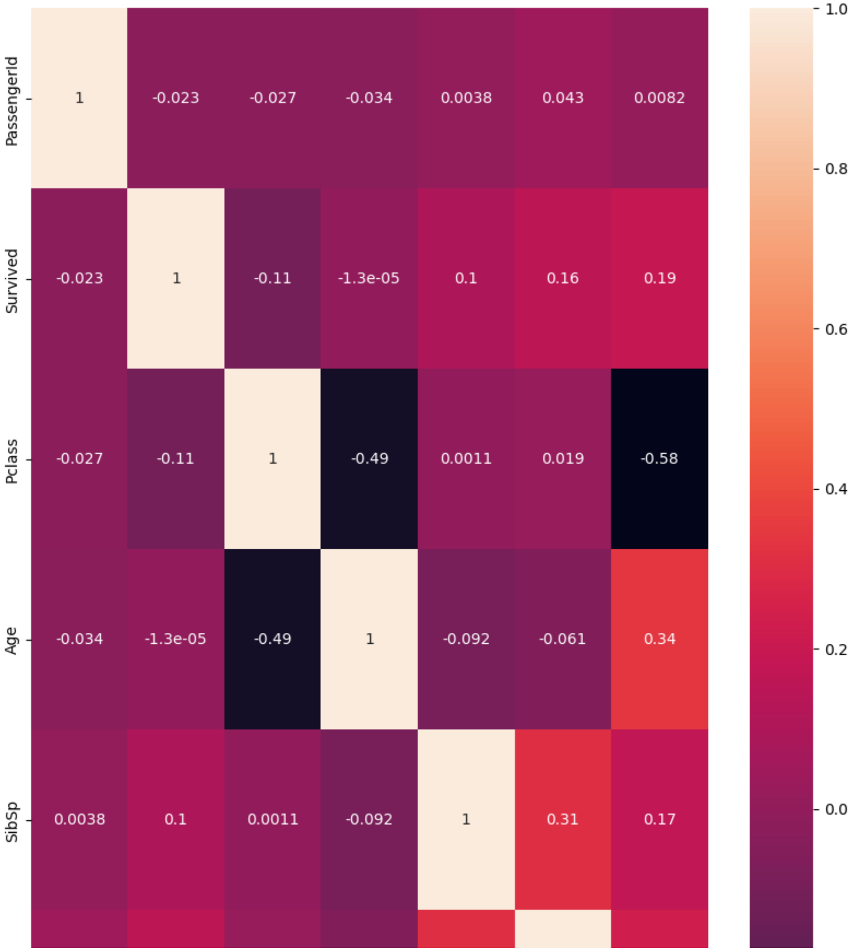
```
corr=dataset.corr()
corr
```

```
<ipython-input-11-f22ca9e9dc13>:1: FutureWarning: The default value of nume
  corr=dataset.corr()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	
PassengerId	1.000000	-0.023245	-0.026751	-0.034102	0.003818	0.043080	0.00
Survived	-0.023245	1.000000	-0.108615	-0.000013	0.099943	0.159120	0.19
Pclass	-0.026751	-0.108615	1.000000	-0.492143	0.001087	0.018721	-0.57
Age	-0.034102	-0.000013	-0.492143	1.000000	-0.091587	-0.061249	0.33
SibSp	0.003818	0.099943	0.001087	-0.091587	1.000000	0.306895	0.17
Parch	0.043080	0.159120	0.018721	-0.061249	0.306895	1.000000	0.23
Fare	0.008211	0.191514	-0.577147	0.337932	0.171539	0.230046	1.00

```
plt.subplots(figsize=(10,15))
sns.heatmap(corr,annot=True)
```

```
<Axes: >
```



```
dataset.Survived.value_counts()
```

```
0    266
1    152
Name: Survived, dtype: int64
```

```
dataset.Sex.value_counts()
```

```
male      266
female    152
Name: Sex, dtype: int64
```

```
dataset.Pclass.value_counts()
```

```
    3    218
    1    107
    2     93
    Name: Pclass, dtype: int64
```

Double-click (or enter) to edit

3.Handling null values

```
dataset.isnull().any()
```

```
    PassengerId    False
    Survived       False
    Pclass         False
    Name           False
    Sex            False
    Age             True
    SibSp          False
    Parch          False
    Ticket         False
    Fare            True
    Cabin           True
    Embarked       False
    dtype: bool
```

```
dataset.isnull().sum()
```

```
    PassengerId      0
    Survived         0
    Pclass           0
    Name             0
    Sex              0
    Age             86
    SibSp            0
    Parch            0
    Ticket           0
    Fare             1
    Cabin          327
    Embarked         0
    dtype: int64
```

```
dataset ["Fare"].fillna(dataset ["Fare"] .mean (), inplace=True)
```

```
dataset ["Age"].fillna(dataset ["Age"] .mean (), inplace=True)
```

```
dataset.isnull().any()
```

```
    PassengerId    False
    Survived       False
    Pclass         False
```

```
Name           False
Sex            False
Age            False
SibSp          False
Parch          False
Ticket         False
Fare           False
Cabin           True
Embarked       False
dtype: bool
```

```
dataset.drop(["Cabin"],axis=1)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
0	892	0	3	Kelly, Mr. James	male	34.50000	0	0
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.00000	1	0
2	894	0	2	Myles, Mr. Thomas Francis	male	62.00000	0	0
3	895	0	3	Wirz, Mr. Albert	male	27.00000	0	0
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.00000	1	1

4.Data Visualisation

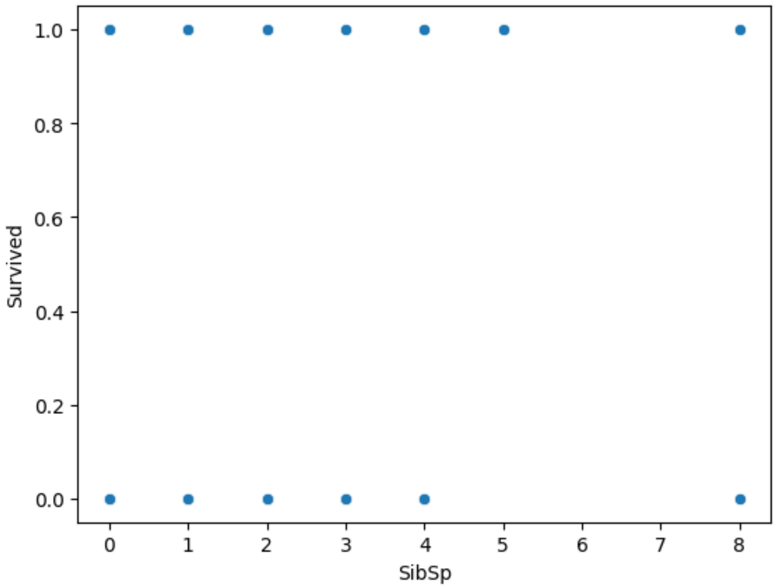
```
sns.scatterplot(x="Age" ,y= "Survived",data=dataset)
```

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

1.0
0.8
0.6
urvived

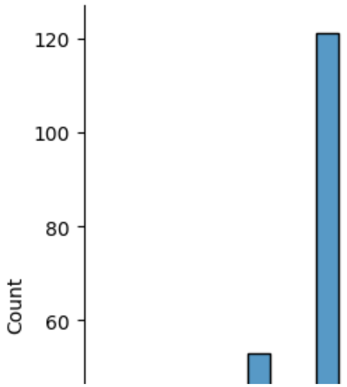
```
sns.scatterplot(x="SibSp" ,y="Survived",data=dataset)
```

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



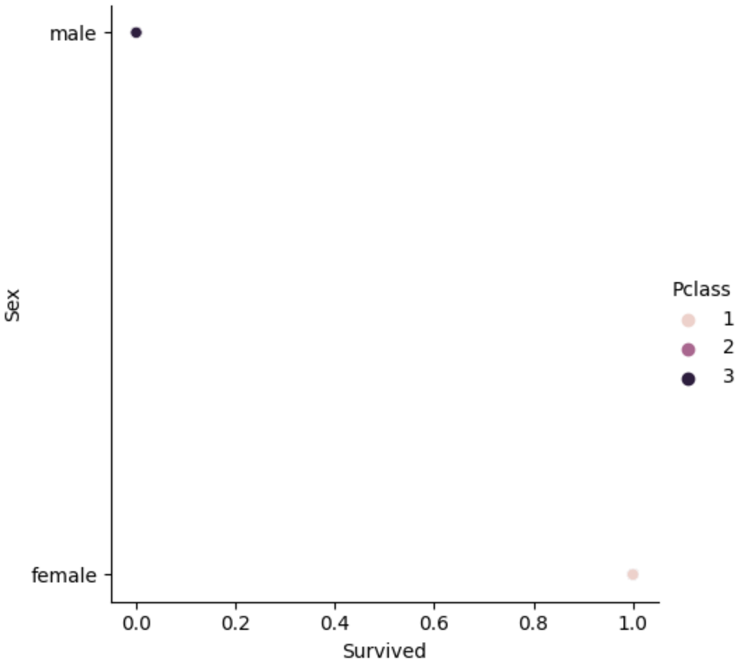
```
sns.displot(dataset["Age"])
```

```
<seaborn.axisgrid.FacetGrid at 0x78073fb8feb0>
```



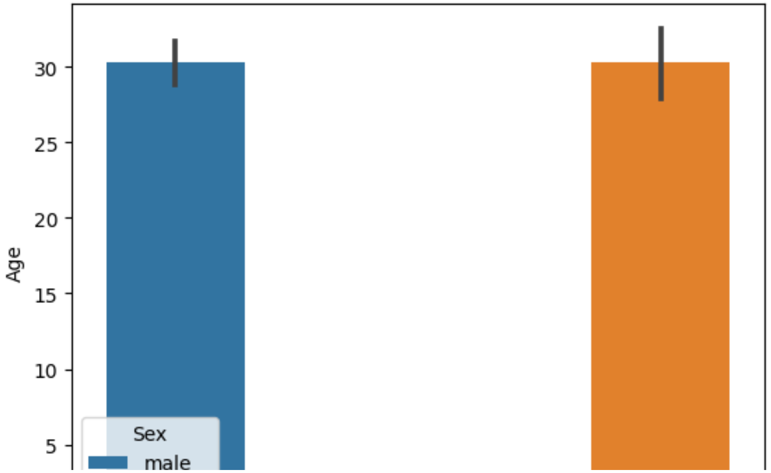
```
sns.relplot(x="Survived",y="Sex",data=dataset,hue="Pclass")
```

```
<seaborn.axisgrid.FacetGrid at 0x78073fa02c20>
```



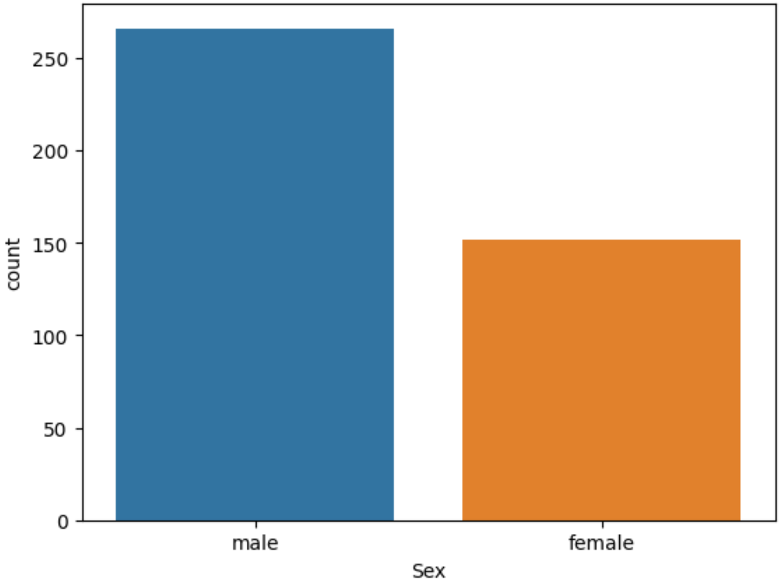
```
sns.barplot(data=dataset,x="Survived",y="Age",hue="Sex")
```

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



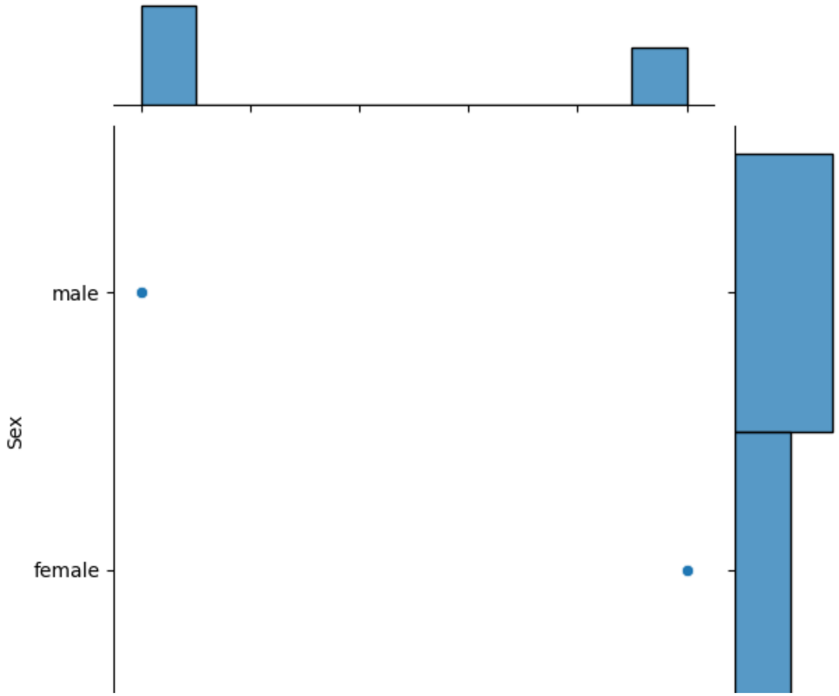
```
sns.countplot(x='Sex',data=dataset)
```

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



```
sns.jointplot(x="Survived",y='Sex',data=dataset)
```

```
<seaborn.axisgrid.JointGrid at 0x78073f7fb880>
```



5.Outliers

```
sns.boxplot(dataset.Age)
```

```
<Axes: >
```

6.Separating Dependent and Independent Varriables

```
dependent_variable = dataset['Survived']
dependent_variable.head()
```

```
0    0
1    1
2    0
3    0
4    1
Name: Survived, dtype: int64
```

```
independent_variables = dataset[['PassengerId','Name','Pclass', 'Sex', 'Age', 'SibSp', 'Parch', 'Fare', 'Embarked']]
```

```
independent_variables.head()
```

	PassengerId	Name	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	892	Kelly, Mr. James	3	male	34.5	0	0	7.8292	Q
1	893	Wilkes, Mrs. James (Ellen Needs)	3	female	47.0	1	0	7.0000	S

```
dependent_variable.shape
```

```
(418,)
```

```
independent_variables.shape
```

```
(418, 9)
```

Double-click (or enter) to edit

```
from sklearn.preprocessing import LabelEncoder
```

```
le=LabelEncoder()
```

```
independent_variables["Sex"] = le.fit_transform(independent_variables["Sex"])
```

```
<ipython-input-41-c1630205b919>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  independent_variables["Sex"] = le.fit_transform(independent_variables["Sex"])
```

7.Encoding

```
from sklearn.model_selection import train_test_split
independent_variables_train, independent_variables_test, dependent_variable_train,dependent_variable_test=train_test_split(independent_variables,dependent_variable,test_size=0.
```

```
independent_variables_train.shape, independent_variables_test.shape, dependent_variable_train.shape,dependent_variable_test.shape
```

```
((292, 9), (126, 9), (292,), (126,))
```

8.Feature scaling

```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler ()
```

```
independent_variables = independent_variables.drop(columns=["Name"])
```

```
independent_variables.head()
```

	PassengerId	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	892	3	1	34.5	0	0	7.8292	Q
1	893	3	0	47.0	1	0	7.0000	S
2	894	2	1	62.0	0	0	9.6875	Q
3	895	3	1	27.0	0	0	8.6625	S
4	896	3	0	22.0	1	1	12.2875	S