```
print("Hello! Python is working")
```

```
Hello! Python is working
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df= pd.read_csv("https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv")
df.head()
```

	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

```
df.shape
```

```
(891, 12)
```

```
df.columns
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

```
df.isnull().sum()
```

	0
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

```
df['Age'].fillna(df['Age'].median(),inplace=True)
```

```
/tmp/ipython-input-1527141296.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]


  df['Age'].fillna(df['Age'].median(),inplace=True)
```

```
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
```

```
/tmp/ipython-input-3744086084.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]
```

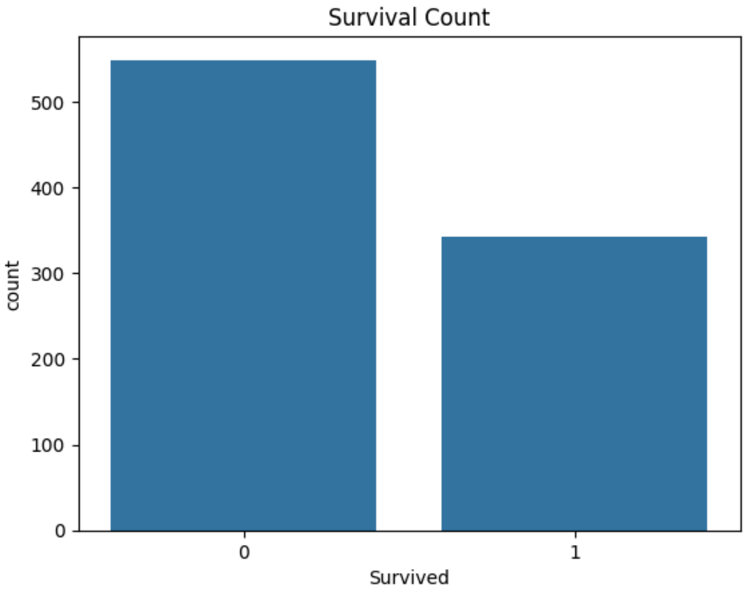
```
  df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
```

```
df.drop('Cabin', axis=1, inplace=True,errors='ignore')
```

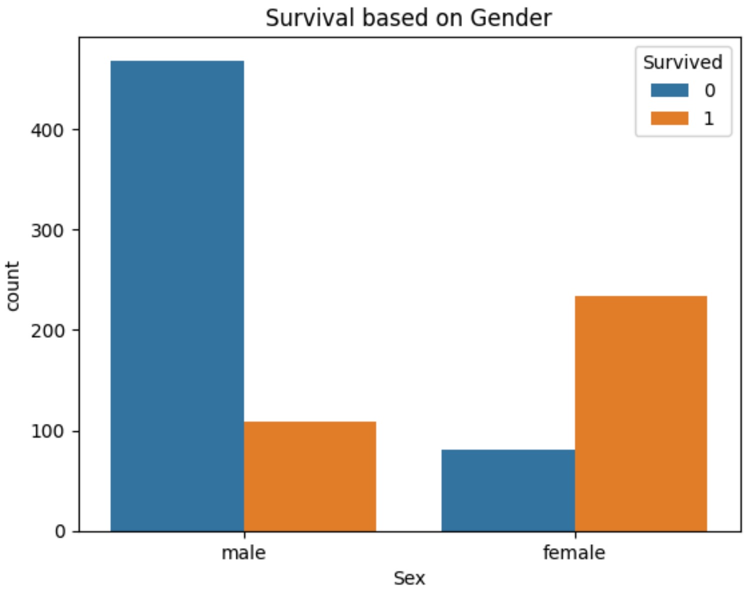
```
df.columns
```

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

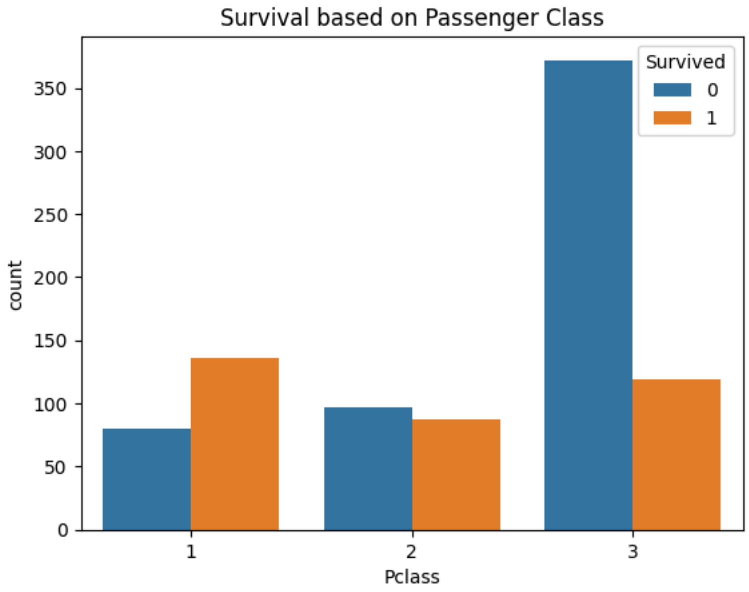
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.countplot(x='Survived',data=df)
plt.title("Survival Count")
plt.show()
```



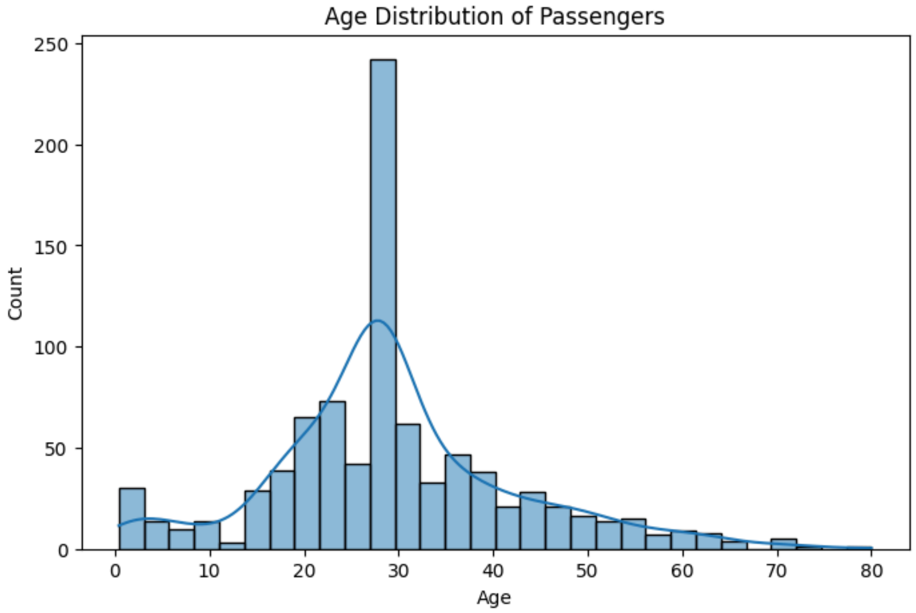
```
sns.countplot(x='Sex', hue='Survived', data=df)
plt.title("Survival based on Gender")
plt.show()
```



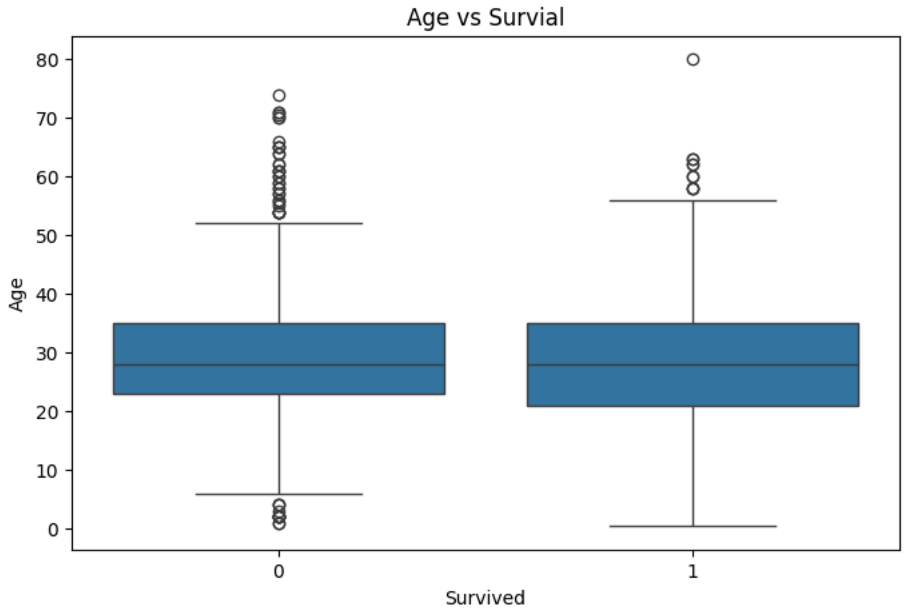
```
sns.countplot(x='Pclass', hue='Survived', data=df)
plt.title("Survival based on Passenger Class")
plt.show()
```



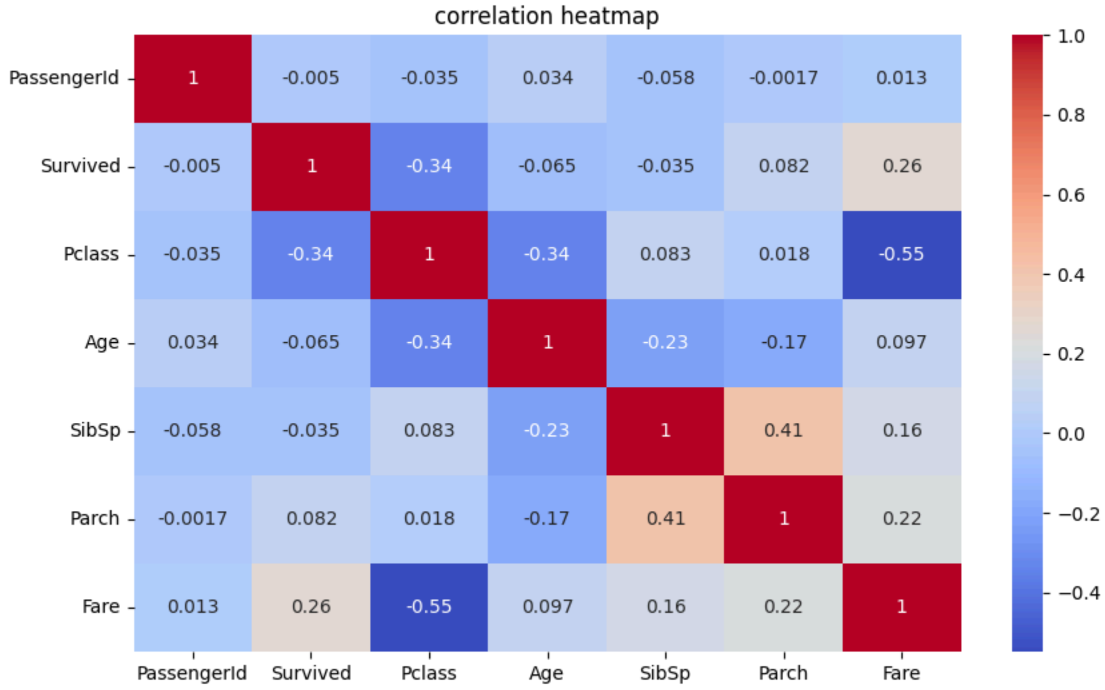
```
plt.figure(figsize=(8,5))
sns.histplot(df['Age'], bins=30, kde=True)
plt.title("Age Distribution of Passengers")
plt.xlabel("Age")
plt.ylabel("Count")
plt.show()
```



```
plt.figure(figsize=(8,5))
sns.boxplot(x='Survived', y='Age',data=df)
plt.title("Age vs Survial")
plt.show()
```



```
from numpy._core.defchararray import title
plt.figure(figsize=(10,6))
sns.heatmap(df.corr(numeric_only=True),annot=True,cmap='coolwarm')
plt.title("correlation heatmap")
plt.show()
```



```
Start coding or generate with AI.
```

Observations:

1. Passengers in 1st class had higher survival rate compared to 2nd and 3rd class.
2. Female passengers survived more than male passengers.
3. Younger passengers had slightly better survival chances.
4. Fare and Pclass show correlation with survival.
5. Age had missing values which were handled using median.

```
# Observations:
# 1. Passengers in 1st class had higher survival rate
# 2. Female passengers survived more
# 3. Younger passengers survived more
```

```
df['Age'].fillna(df['Age'].median(),inplace=True)
```

```
/tmp/ipython-input-1527141296.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]


  df['Age'].fillna(df['Age'].median(),inplace=True)
```

```
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
```

```
/tmp/ipython-input-3744086084.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]


  df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)
```

```
df['Sex'] = df['Sex'].map({'male': 0, 'female': 1})
df['Embarked'] = df['Embarked'].map({'S': 0, 'C': 1, 'Q': 2})
```

```
df.head()
```

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

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

```
X = df[['Pclass', 'Sex', 'Age', 'Fare']]
y = df['Survived']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

```
model = LogisticRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
accuracy_score(y_test, y_pred)
```

```
0.8044692737430168
```

```
X.isnull().sum()
```

	0
Pclass	0
Sex	0
Age	0
Fare	0

dtype: int64

```
df['Age'].fillna(df['Age'].median(),inplace=True)
df['Fare'].fillna(df['Fare'].median(),inplace=True)
```

```
/tmp/ipython-input-1020745694.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]


  df['Age'].fillna(df['Age'].median(),inplace=True)
/tmp/ipython-input-1020745694.py:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through ch
```

```
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col]


  df['Fare'].fillna(df['Fare'].median(),inplace=True)
```

```
df['Sex'] = df['Sex'].map({'male': 0, 'female': 1})
df['Sex'].fillna(df['Sex'].mode()[0], inplace=True)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/range.py in get_loc(self, key)
    412             try:
--> 413                 return self._range.index(new_key)
    414             except ValueError as err:

ValueError: 0 is not in range

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
                                  3 frames
/usr/local/lib/python3.12/dist-packages/pandas/core/indexes/range.py in get_loc(self, key)
    413                 return self._range.index(new_key)
    414             except ValueError as err:
--> 415                 raise KeyError(key) from err
    416         if isinstance(key, Hashable):
    417             raise KeyError(key)

KeyError: 0
```

```
X = df[['Pclass', 'Sex', 'Age', 'Fare']]
y = df['Survived']
```

```
X.isnull().sum()
```

```
df_model = df[['Survived', 'Pclass', 'Sex', 'Age', 'Fare']]
```

Double-click (or enter) to edit

```
df_model['Sex'] = df_model['Sex'].map({'male': 0, 'female': 1})
```

```
df_model['Sex'].value_counts()
```

```
df_model['Age'] = df_model['Age'].fillna(df_model['Age'].median())
df_model['Fare'] = df_model['Fare'].fillna(df_model['Fare'].median())
```

```
df_model.isnull().sum()
```

```
X = df_model.drop('Survived',axis=1)
y = df_model['Survived']
```

Double-click (or enter) to edit

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
accuracy_score(y_test, y_pred)
```

```
0.8044692737430168
```

```
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test,y_pred)
cm
```

```
array([[90, 15],
       [20, 54]])
```

```
from sklearn .metrics import classification_report

print(classification_report(y_test,y_pred))
```

```
              precision    recall  f1-score   support

           0       0.82      0.86      0.84       105
           1       0.78      0.73      0.76        74

    accuracy                           0.80       179
   macro avg       0.80      0.79      0.80       179
weighted avg       0.80      0.80      0.80       179
```

```
# Example passenger
# Pclass=3, Sex=female(1), Age=26, Fare=7.25
sample = [[3, 1, 26, 7.25]]S

prediction = model.predict(sample)
prediction
```

```
/usr/local/lib/python3.12/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names,
  warnings.warn(
array([1])
```