```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.io as pio
pio.renderers.default = "notebook_connected"
```

```
data = pd.read_csv("/kaggle/input/titanic/train.csv")
data.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
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

```
data.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
```

```
data = data.drop(columns=['Name','PassengerId','Ticket'])
data.head()
```

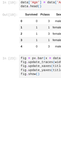
	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Cabin	Embarked
0	0	3	male	22.0	1	0	7.2500	NaN	S
1	1	1	female	38.0	1	0	71.2833	C85	C
2	1	3	female	26.0	0	0	7.9250	NaN	S
3	1	1	female	35.0	1	0	53.1000	C123	S
4	0	3	male	35.0	0	0	8.0500	NaN	S

```
print("data shape",data.shape)
data.isnull().sum()
```

```
data shape (891, 9)
Survived    0
Pclass      0
Sex         0
Age       177
SibSp       0
Parch       0
Fare        0
Cabin     687
Embarked    2
dtype: int64
```

```
data = data.drop(columns=["Cabin"])
```

```
fig = px.bar(x = data['Embarked'].value_counts().keys(),y = data['Embarked'].value_counts().values)
fig.update_traces(width = 0.5)
fig.update_xaxes(title="Embarked")
fig.update_yaxes(title="count")
fig.show()
```



```
data['Embarked'] = data['Embarked'].fillna(data['Embarked'].mode(dropna=True))
print("data_frame",data.isnull().sum)
fig.show()
```

```
def find_outlier_percentage(data:pd.DataFrame,column:str) -> None:
    Q1,Q3 = np.percentile(data[column].dropna(),[25,75])
    IQR = Q3-Q1
    Low = Q1 - 1.5*IQR
    Hi = Q3 + 1.5*IQR
    len, = data[(data[column] < Low) | (data[column] > Hi)].shape[0]
    print(f"{column} outliers: {len} / {data[column].shape[0]}")
find_outlier_percentage(data=data,column='Age')
```

```
Age 3.23
```

```
data['Age'].describe()
```

```
count    714.000000
mean      29.699118
std       14.526497
min        0.420000
25%       20.125000
50%       28.000000
75%       38.000000
max       80.000000
Name: Age, dtype: float64
```

```
data['Age'].median()
```

```
28.0
```

```
data['Age'] = data['Age'].fillna(data['Age'].median())
data.head()
```

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	0	3	male	22.0	1	0	7.2500	S
1	1	1	female	38.0	1	0	71.2833	C
2	1	3	female	26.0	0	0	7.9250	S
3	1	1	female	35.0	1	0	53.1000	S
4	0	3	male	35.0	0	0	8.0500	S

```
fig = px.bar(x = data['Survived'].value_counts().keys(),y = data['Survived'].value_counts().values)
fig.update_traces(width = 0.5)
fig.update_xaxes(title="Survived")
fig.update_yaxes(title="count")
fig.show()
```



```
fig = px.bar(x = data['Pclass'].value_counts().keys(),y = data['Pclass'].value_counts().values)
fig.update_traces(width = 0.5)
fig.update_xaxes(title="Pclass")
fig.update_yaxes(title="count")
fig.show()
```

```
fig = px.bar(x = data['Sex'].value_counts().keys(),y = data['Sex'].value_counts().values)
fig.update_traces(width = 0.5)
fig.update_xaxes(title="Sex")
fig.update_yaxes(title="count")
fig.show()
```

```
fig = px.histogram(data_frame=data,x = "Age",marginal="rug")
fig.update_xaxes(title="Age")
fig.update_yaxes(title="count")
fig.show()
```

```
fig = px.histogram(data_frame=data,x = "SibSp",marginal="rug")
fig.update_xaxes(title="SibSp")
fig.update_yaxes(title="count")
fig.show()
```

```
fig = px.histogram(data_frame=data,x = "Fare",marginal="rug")
fig.update_xaxes(title="Fare")
fig.update_yaxes(title="count")
fig.show()
```



```
Fare 13.92
```

```
fig = px.histogram(data_frame=data,x = "Parch",marginal="rug")
fig.update_xaxes(title="Parch")
fig.update_yaxes(title="count")
fig.show()
```

```
plt.figure(figsize=(16,14))
sns.heatmap(data.corr(numeric_only=True),annot=True)
plt.show()
```

```
data = pd.get_dummies(data=data,drop_first=True)
data.head()
```

	Survived	Pclass	Age	SibSp	Parch	Fare	Sex_female	Sex_male	Embarked_C	Embarked_Q	Embarked_S
0	0	3	22.0	1	0	7.2500	0	1	0	0	1
1	1	1	38.0	1	0	71.2833	1	0	1	0	0
2	1	3	26.0	0	0	7.9250	1	0	0	0	1
3	1	1	35.0	1	0	53.1000	1	0	0	0	1
4	0	3	35.0	0	0	8.0500	0	1	0	0	1

```
[data['Fare'] >= 0].sum()
```

```
Fare    0.0
dtype: float64
```

```
data['Fare'] = np.power(data['Fare'],0.25)
data['Fare'].sum()
```

```
0.12857999999999
```

```
def normalize(data:pd.DataFrame,column:str):
    mean = data[column].mean()
    std = data[column].std()
    data[column] = data[column].apply(lambda x : (x-mean)/std)
normalize(data=data,column='Fare')
data['Fare'].head()
```

```
0   -0.565419
1    0.663488
2   -0.258143
3    0.433102
4   -0.433102
Name: Fare, dtype: float64
```

```
data.head()
```

	Survived	Pclass	Age	SibSp	Parch	Fare	Sex_female	Sex_male	Embarked_C	Embarked_Q	Embarked_S
0	0	3	0.434597	1	0	-0.565419	0	1	0	0	1
1	1	1	0.846647	1	0	0.663488	1	0	1	0	0
2	1	3	-0.284547	0	0	-0.257857	1	0	0	0	1
3	1	1	0.633566	1	0	0.433102	1	0	0	0	1
4	0	3	0.633566	0	0	-0.433102	0	1	0	0	1

```
data.describe()
```

	Survived	Pclass	Age	SibSp	Parch	Fare	Sex_female	Sex_male	Embarked_C	Embarked_Q	Embarked_S
count	891.000000	891.000000	8.910000e+02	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	0.383838	2.308642	2.311624e-18	0.523008	0.381594	2.111111	0.352413	0.647587	0.188552	0.086420	0.722783
std	0.486592	0.836071	1.000562e+00	1.102743	0.806057	0.811008	0.477990	0.477990	0.391372	0.281141	0.447876
min	0.000000	1.000000	-2.223964e+00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	2.000000	5.454545e-01	0.000000	0.000000	1.677966	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	3.000000	-1.042767e-01	0.000000	0.000000	2.500000	0.000000	1.000000	0.000000	0.000000	1.000000
75%	1.000000	3.000000	6.666667e-01	1.000000	0.000000	2.964411	1.000000	1.000000	0.000000	0.000000	1.000000
max	1.000000	3.000000	3.666667e+00	8.000000	6.000000	4.767583	1.000000	1.000000	1.000000	1.000000	1.000000

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(data.drop(columns=['Survived']),data.sample(1)['Survived'],data.sample(1,random_state=42,train_size=
```

```
X_train
```

```
array([[ 1.        , -0.947009,  0.      ,  0.      ,  0.      ,
        -0.      ,  1.      ,  0.      ,  0.      ,  1.      ],
       [ 1.        , -0.104704,  0.      ,  0.      ,  0.      ,
         0.      ,  0.      ,  1.      ,  0.      ,  0.      ],
       [ 1.        , -0.170103,  1.      ,  0.      ,  0.      ,
         0.      ,  0.      ,  1.      ,  0.      ,  1.      ],
       [ 1.        ,  0.002000,  1.      ,  0.      ,  0.      ,
         0.      ,  0.      ,  1.      ,  0.      ,  1.      ],
       [ 1.        ,  0.170103,  0.      ,  0.      ,  0.      ,
         0.      ,  0.      ,  1.      ,  0.      ,  1.      ],
       [ 1.        ,  0.622334,  0.      ,  1.      ,  0.      ,
         0.      ,  0.      ,  1.      ,  0.      ,  1.      ],
       [ 1.        ,  0.622334,  0.      ,  0.      ,  0.      ,
         0.      ,  0.      ,  1.      ,  0.      ,  1.      ],
       [ 1.        ,  1.000000,  0.      ,  0.      ,  0.      ,
         0.      ,  0.      ,  1.      ,  0.      ,  1.      ],
       [ 1.        ,  0.622334,  1.      ,  0.      ,  0.      ,
         0.      ,  0.      ,  1.      ,  0.      ,  1.      ],
       [ 1.        ,  0.622334,  0
```