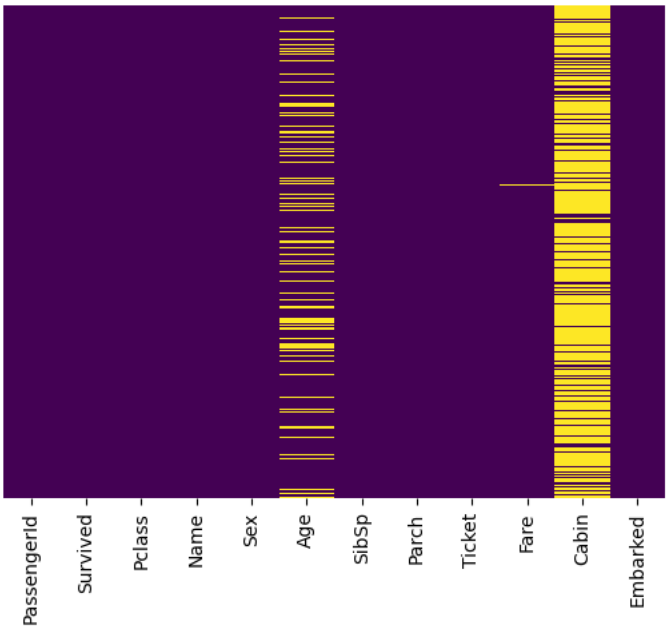
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix, classification_report , accuracy_score
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings("ignore")
```

```
df= pd.read_csv('/content/tested.csv')
df
df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fa
count	418.000000	418.000000	418.000000	332.000000	418.000000	418.000000	417.000(
mean	1100.500000	0.363636	2.265550	30.272590	0.447368	0.392344	35.627
std	120.810458	0.481622	0.841838	14.181209	0.896760	0.981429	55.907
min	892.000000	0.000000	1.000000	0.170000	0.000000	0.000000	0.000(
25%	996.250000	0.000000	1.000000	21.000000	0.000000	0.000000	7.895
50%	1100.500000	0.000000	3.000000	27.000000	0.000000	0.000000	14.454
75%	1204.750000	1.000000	3.000000	39.000000	1.000000	0.000000	31.500(
max	1309.000000	1.000000	3.000000	76.000000	8.000000	9.000000	512.329

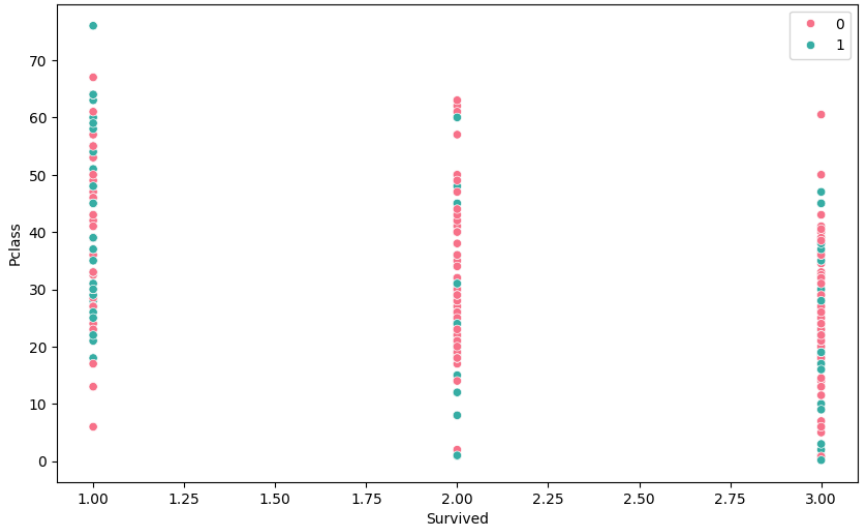
```
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

```
<Axes: >
```



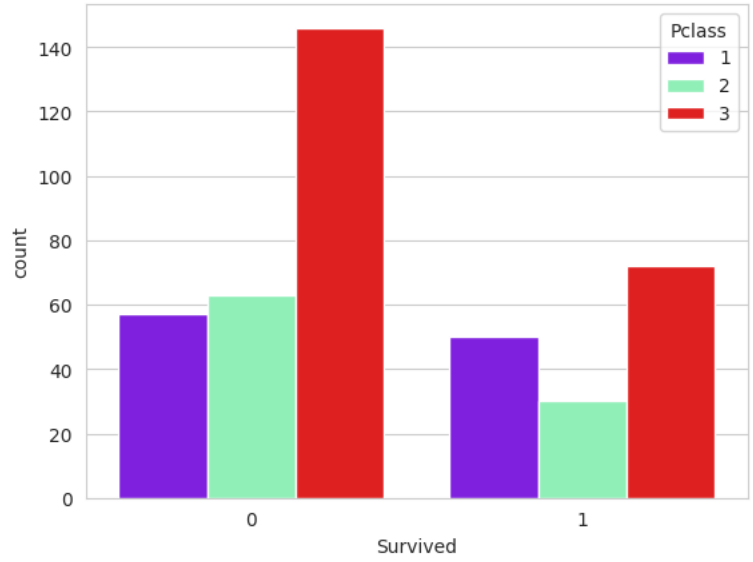

```
plt.figure(figsize=(10, 6))
sns.scatterplot(y=df["Age"] , x=df["Pclass"] ,hue=df["Survived"] ,palette="husl")
plt.xlabel("Survived")
plt.ylabel("Pclass")
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7af053442bc0>
```



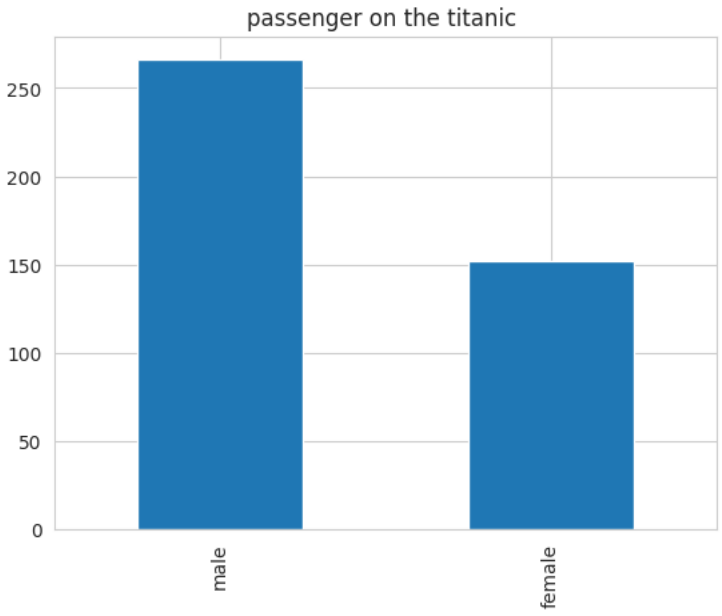

```
sns.set_style('whitegrid')
sns.countplot(x='Survived',hue='Pclass',data=df,palette='rainbow')
```

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



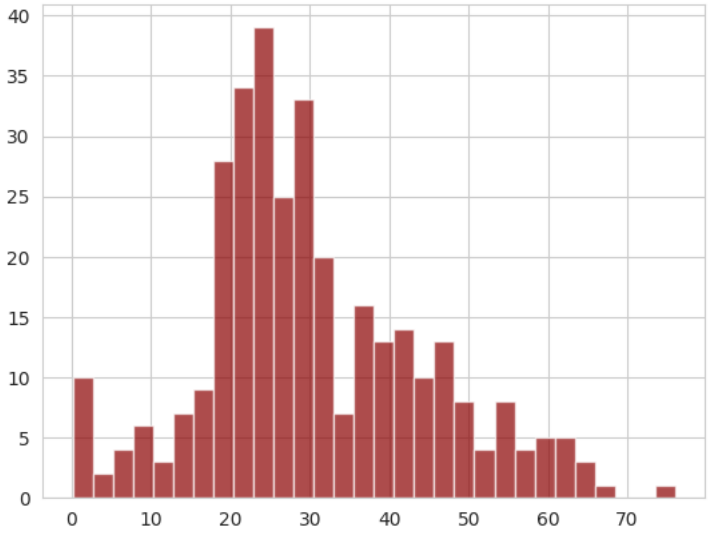

```
df['Sex']. value_counts().plot.bar(title="passenger on the titanic ")
```

```
<Axes: title={'center': 'passenger on the titanic '}>
```




```
df['Age'].hist(bins=30,color='darkred',alpha=0.7)
```

```
<Axes: >
```




```
df.drop('Cabin',axis=1,inplace=True)
df= df.dropna(subset=['Fare', 'Age'])
df
```

	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
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298
...	...	...	...	...	...	...	...	...	...

```
X=df[["Pclass" , "Sex" , "Age" , "Fare"]]
y=df["Survived"]
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
X["Sex"] = encoder.fit_transform(X["Sex"])
X
```

	Pclass	Sex	Age	Fare
0	3	1	34.5	7.8292
1	3	0	47.0	7.0000
2	2	1	62.0	9.6875
3	3	1	27.0	8.6625
4	3	0	22.0	12.2875
...	...	...	...	...
409	3	0	3.0	13.7750
411	1	0	37.0	90.0000
412	3	0	28.0	7.7750
414	1	0	39.0	108.9000
415	3	1	38.5	7.2500

331 rows × 4 columns

+ Code + Text

```
y
```

```
0      0
1      1
2      0
3      0
4      1
      ..
409    1
411    1
412    1
414    1
415    0
Name: Survived, Length: 331, dtype: int64
```

```
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
LR_model=LogisticRegression(max_iter=1500)
LR_model.fit(X_train,y_train)
```

```
▾ LogisticRegression
LogisticRegression(max_iter=1500)
```

```
LR_model.score(X_train,y_train)
LR_pred = LR_model.predict(X_test)
cm1 = confusion_matrix(y_test,LR_pred)
plt.figure(figsize=(10,7))
sns.heatmap(cm1,annot=True)
plt.xlabel('predicted')
plt.ylabel('Truh')
```

```
Text(95.72222222222221, 0.5, 'Truh')
```