```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import math
td = pd.read_csv("train.csv")
```

```
td.head(5)
```

```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

**Titanic Dataset Description

1. Data Overview

- Total Entries: 887
- Columns: 8

1. Types of Data:

- Numerical: Survived, Pclass, Age, Siblings/Spouses Aboard, Parents/Children Aboard, Fare
- Categorical: Name, Sex

1. Column Description

- Survived (0 = No, 1 = Yes) → Indicates whether the passenger survived.
- Pclass (1st, 2nd, 3rd) → Passenger class.
- Name → Passenger's full name.
- Sex → Gender of the passenger.
- Age → Age of the passenger.

- Siblings/Spouses Aboard → Number of siblings or spouses aboard.
- Parents/Children Aboard → Number of parents or children aboard.
- Fare → Ticket price paid by the passenger.

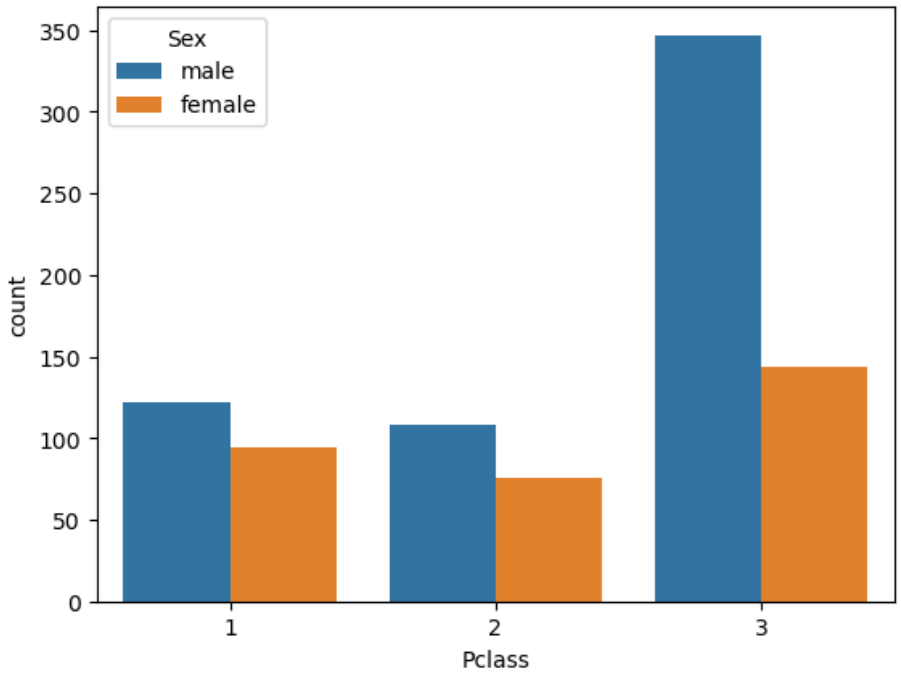
```
plt.figure(figsize=(20, 25))
```

```
<Figure size 2000x2500 with 0 Axes>
```

```
<Figure size 2000x2500 with 0 Axes>
```

```
sns.countplot(x="Pclass", hue="Sex",data=td)
```

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



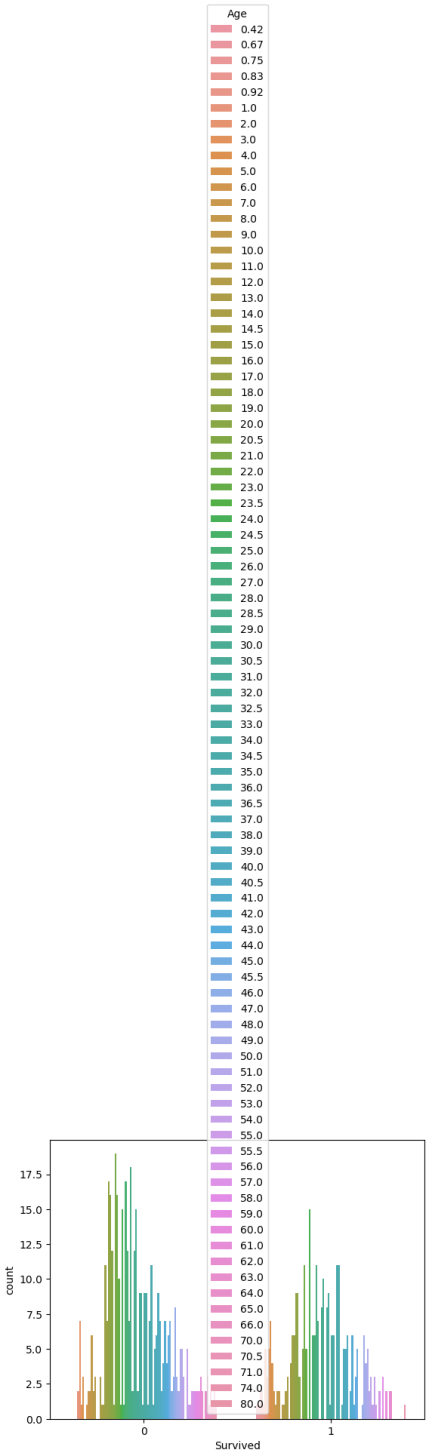

```
sns.countplot(x="Survived", hue="Pclass",data=td)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[1], line 1
----> 1 sns.countplot(x="Survived", hue="Pclass",data=td)

NameError: name 'sns' is not defined
```

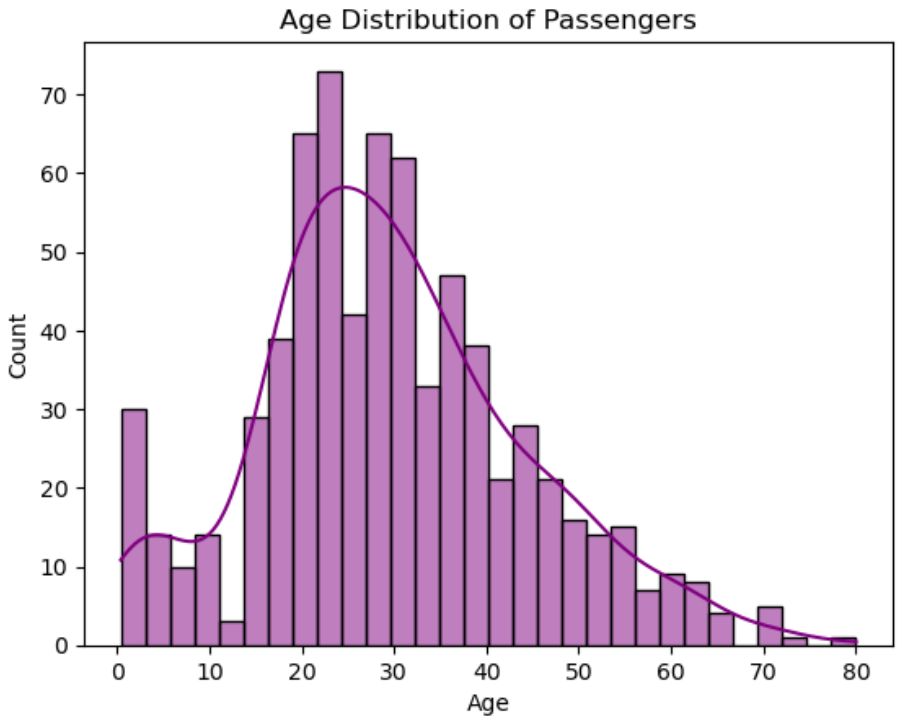
```
sns.countplot(x="Survived", hue="Age",data=td)
```

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


Age
0.42
0.67
0.75
0.83
0.92
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
14.5
15.0
16.0
17.0
18.0
19.0
20.0
20.5
21.0
22.0
23.0
23.5
24.0
24.5
25.0
26.0
27.0
28.0
28.5
29.0
30.0
30.5
31.0
32.0
32.5
33.0
34.0
34.5
35.0
36.0
36.5
37.0
38.0
39.0
40.0
40.5
41.0
42.0
43.0
44.0
45.0
45.5
46.0
47.0
48.0
49.0
50.0
51.0
52.0
53.0
54.0
55.0
55.5
56.0
57.0
58.0
59.0
60.0
61.0
62.0
63.0
64.0
65.0
66.0
70.0
70.5
71.0
74.0
80.0
count
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0
1
Survived

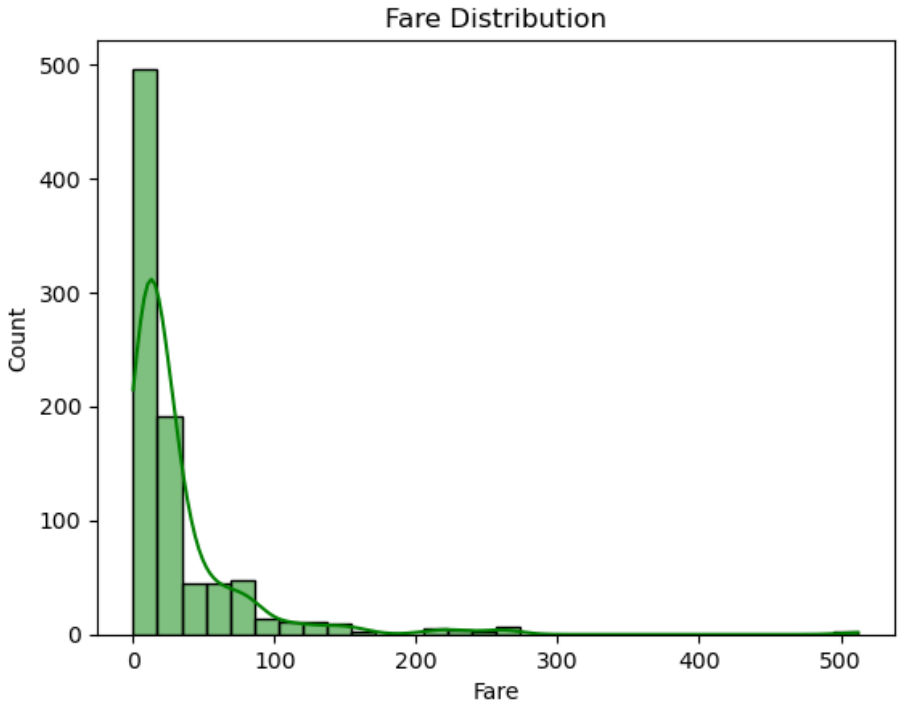
```
sns.histplot(td["Age"], bins=30, kde=True, color="purple")
plt.title("Age Distribution of Passengers")
```

```
Text(0.5, 1.0, 'Age Distribution of Passengers')
```



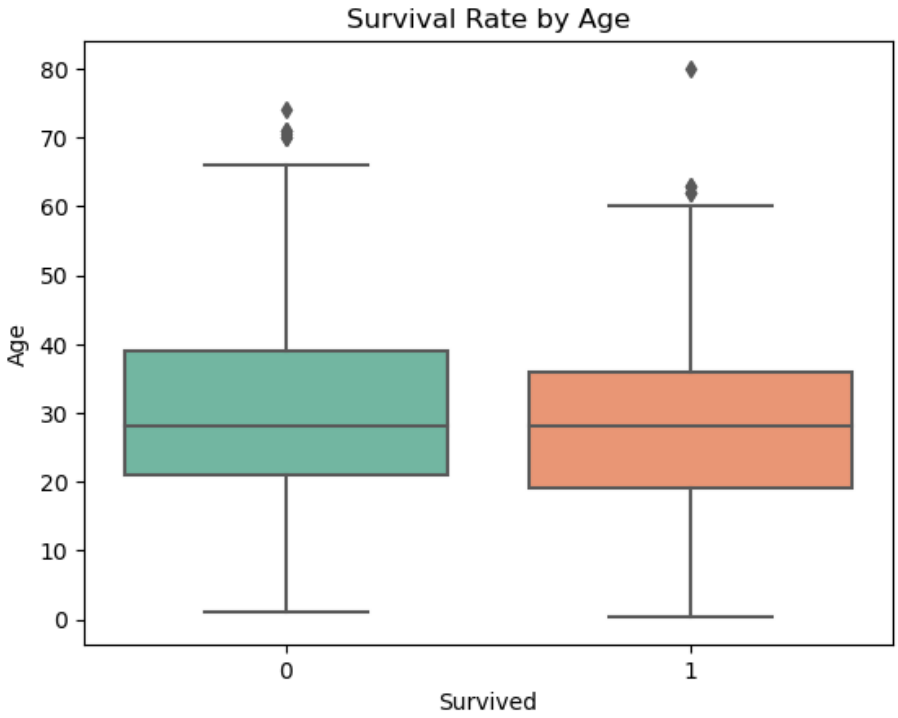

```
sns.histplot(td["Fare"], bins=30, kde=True, color="green")
plt.title("Fare Distribution")
```

```
Text(0.5, 1.0, 'Fare Distribution')
```



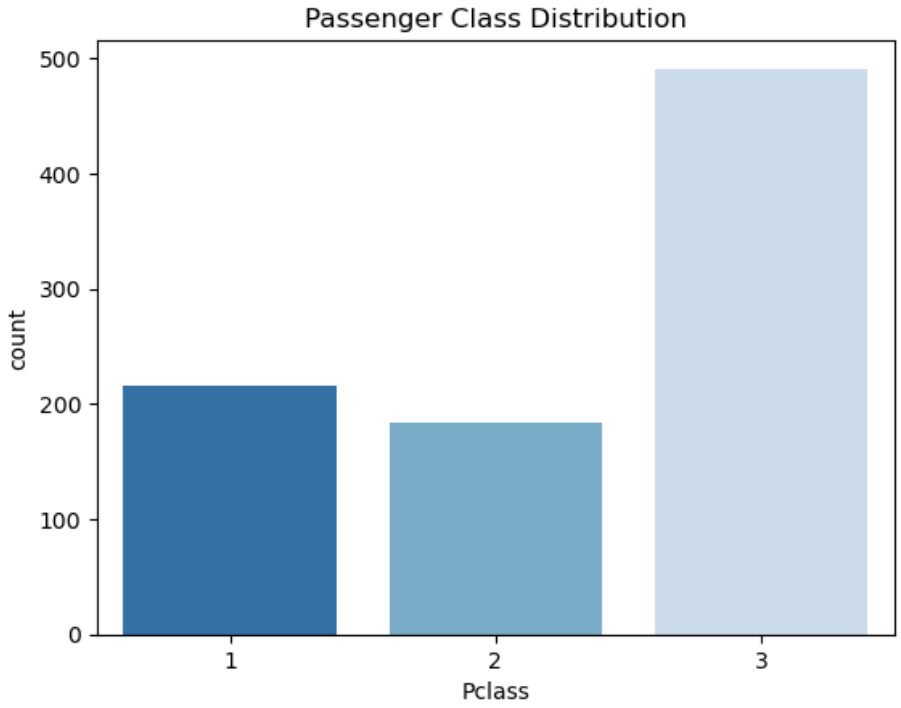

```
sns.boxplot(x="Survived", y="Age", data=td, palette="Set2")
plt.title("Survival Rate by Age")
```

```
Text(0.5, 1.0, 'Survival Rate by Age')
```



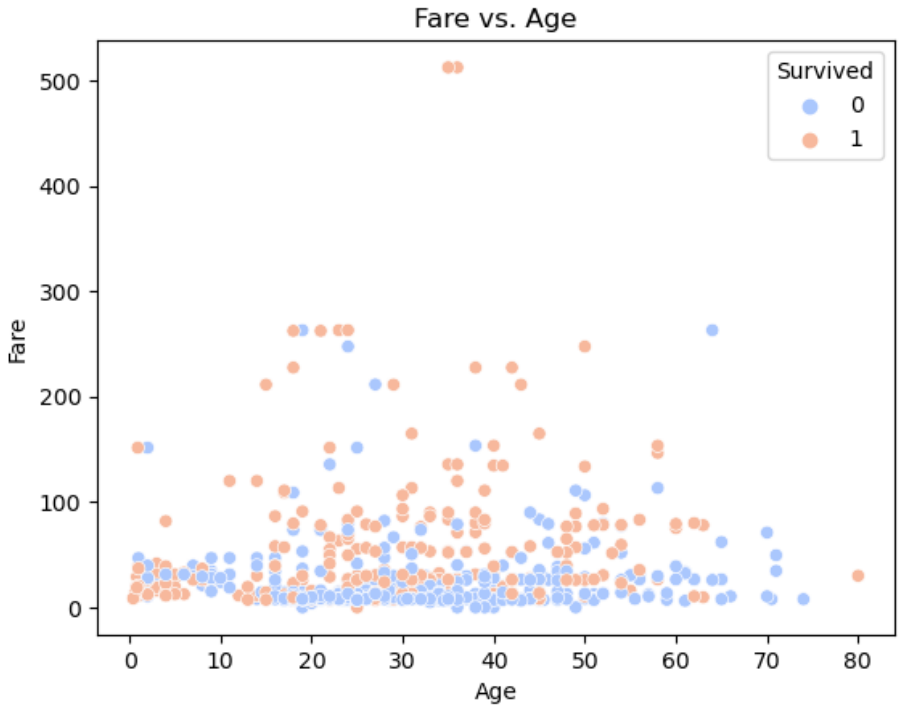

```
sns.countplot(x="Pclass", data=td, palette="Blues_r")
plt.title("Passenger Class Distribution")
```

```
Text(0.5, 1.0, 'Passenger Class Distribution')
```



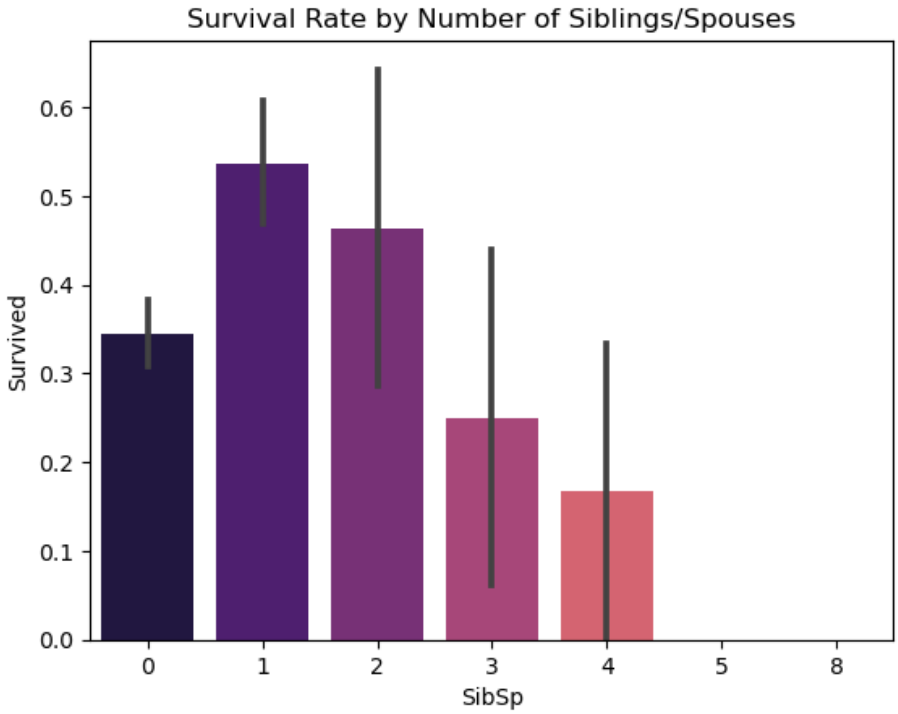

```
sns.scatterplot(x="Age", y="Fare", hue="Survived", data=td, palette="coolwarm")
plt.title("Fare vs. Age")
```

```
Text(0.5, 1.0, 'Fare vs. Age')
```



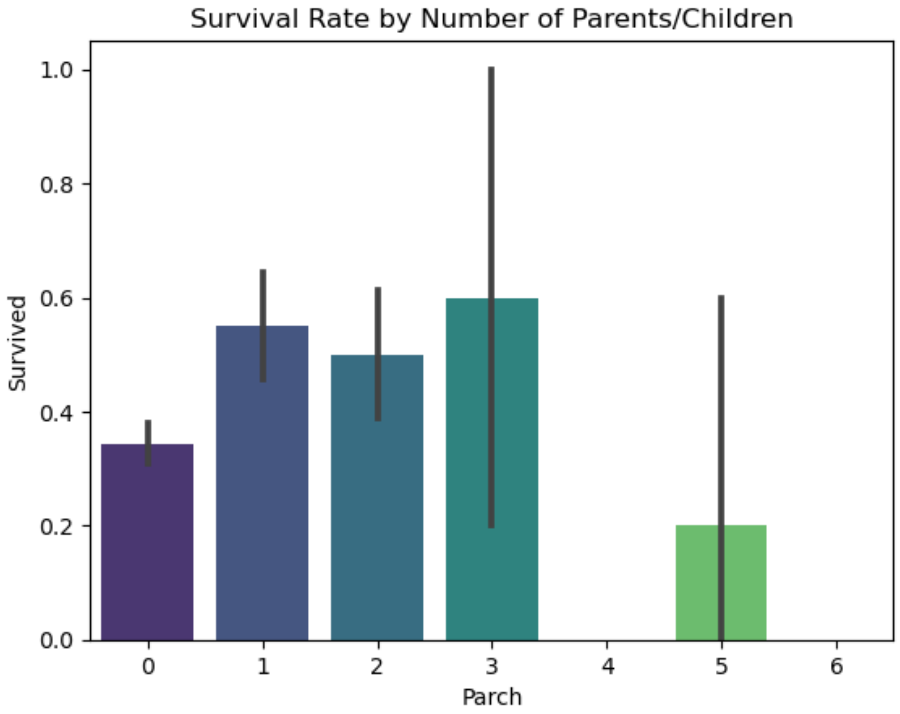

```
sns.barplot(x="SibSp", y="Survived", data=td, palette="magma")
plt.title("Survival Rate by Number of Siblings/Spouses")
```

```
Text(0.5, 1.0, 'Survival Rate by Number of Siblings/Spouses')
```



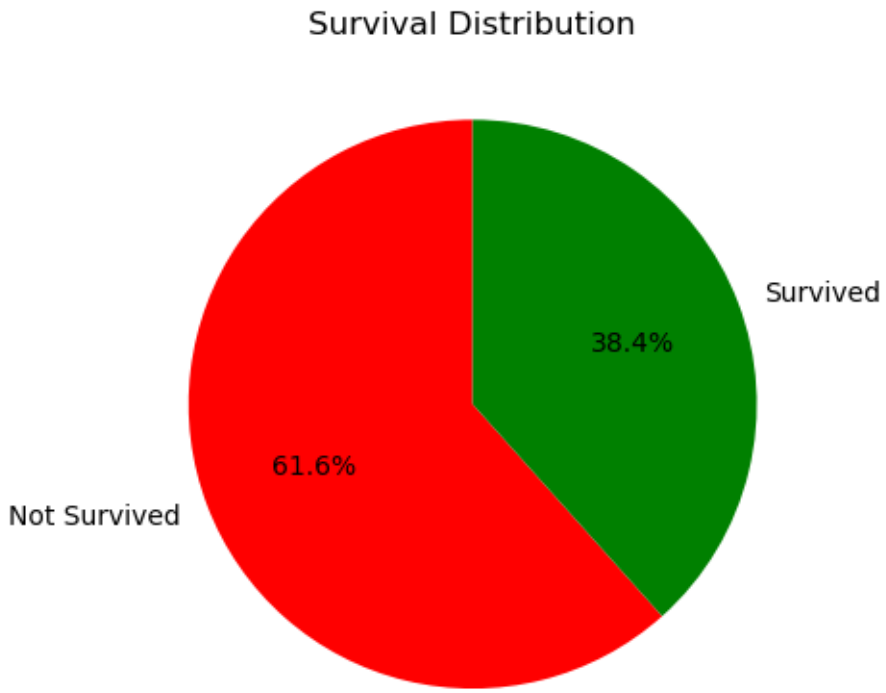

```
sns.barplot(x="Parch", y="Survived", data=td, palette="viridis")
plt.title("Survival Rate by Number of Parents/Children")
```

```
Text(0.5, 1.0, 'Survival Rate by Number of Parents/Children')
```




```
survival_counts = td["Survived"].value_counts()
plt.pie(survival_counts, labels=["Not Survived", "Survived"], autopct="%1.1f%%", colors=["re
plt.title("Survival Distribution")
```

```
Text(0.5, 1.0, 'Survival Distribution')
```




```
td.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
```

```
td.isnull()
```

```
     PassengerId  Survived  Pclass   Name    Sex    Age  SibSp  Parch  Ticket  \
0          False     False   False  False  False  False  False  False   False
1          False     False   False  False  False  False  False  False   False
2          False     False   False  False  False  False  False  False   False
3          False     False   False  False  False  False  False  False   False
4          False     False   False  False  False  False  False  False   False
..           ...       ...     ...    ...    ...    ...    ...    ...     ...
886        False     False   False  False  False  False  False  False   False
887        False     False   False  False  False  False  False  False   False
888        False     False   False  False  False   True  False  False   False
889        False     False   False  False  False  False  False  False   False
890        False     False   False  False  False  False  False  False   False

      Fare  Cabin  Embarked
0    False   True     False
1    False  False     False
2    False   True     False
3    False  False     False
4    False   True     False
..     ...    ...       ...
886  False   True     False
887  False  False     False
888  False   True     False
889  False  False     False
890  False   True     False

[891 rows x 12 columns]
```

```
td.isnull().sum()
```

```
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
```

```
td.drop(["Name","PassengerId","Ticket"],axis=1,inplace=True)
```

```
td.Age.mean()
```

```
29.69911764705882
```

```
td['Age'] = td['Age'].fillna(td['Age'].mean())
```

```
td.isnull().sum()
```

```
Survived      0
Pclass        0
Sex           0
Age           0
SibSp         0
Parch         0
Fare          0
Cabin       687
Embarked      2
dtype: int64
```

```
td.isnull().info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  891 non-null    bool
 1   Pclass    891 non-null    bool
 2   Sex       891 non-null    bool
 3   Age       891 non-null    bool
 4   SibSp     891 non-null    bool
 5   Parch     891 non-null    bool
 6   Fare      891 non-null    bool
 7   Cabin     891 non-null    bool
 8   Embarked  891 non-null    bool
dtypes: bool(9)
memory usage: 8.0 KB
```

```
td.isnull().sum()
```

```
Survived      0
Pclass        0
Sex           0
Age           0
SibSp         0
Parch         0
Fare          0
Cabin       687
Embarked      2
dtype: int64
```

```
td.isnull().info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  891 non-null    bool
 1   Pclass    891 non-null    bool
 2   Sex       891 non-null    bool
 3   Age       891 non-null    bool
 4   SibSp     891 non-null    bool
 5   Parch     891 non-null    bool
 6   Fare      891 non-null    bool
 7   Cabin     891 non-null    bool
 8   Embarked  891 non-null    bool
dtypes: bool(9)
memory usage: 8.0 KB
```

```
td.loc[(td["Survived"] == 1) & (td["Pclass"] == 1) & (td["Cabin"].isna()), "Cabin"] = "B78"
td.loc[(td["Survived"] == 1) & (td["Pclass"] == 2) & (td["Cabin"].isna()), "Cabin"] = "E102"
td.loc[(td["Survived"] == 1) & (td["Pclass"] == 3) & (td["Cabin"].isna()), "Cabin"] = "E104"

td.loc[(td["Survived"] == 0) & (td["Pclass"] == 1) & (td["Cabin"].isna()), "Cabin"] = "C119"
td.loc[(td["Survived"] == 0) & (td["Pclass"] == 2) & (td["Cabin"].isna()), "Cabin"] = "E"
td.loc[(td["Survived"] == 0) & (td["Pclass"] == 3) & (td["Cabin"].isna()), "Cabin"] = "G"
```

```
td["Embarked"].fillna("S", inplace=True)
```

```
td.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  891 non-null    int64
 1   Pclass    891 non-null    int64
 2   Sex       891 non-null    object
 3   Age       891 non-null    float64
 4   SibSp     891 non-null    int64
 5   Parch     891 non-null    int64
 6   Fare      891 non-null    float64
 7   Cabin     891 non-null    object
 8   Embarked  891 non-null    object
dtypes: float64(2), int64(4), object(3)
memory usage: 62.8+ KB
```

```
print(td['Sex'])
```

```
0        male
1      female
2      female
3      female
4        male
        ...
886      male
887    female
888    female
889      male
890      male
Name: Sex, Length: 891, dtype: object
```

```
print(td['Embarked'])
```

```
0      S
1      C
2      S
3      S
4      S
      ..
886    S
887    S
888    S
889    C
890    Q
Name: Embarked, Length: 891, dtype: object
```

```
print(td['Pclass'])
```

```
0      3
1      1
2      3
3      1
4      3
      ..
886    2
887    1
888    3
889    1
890    3
Name: Pclass, Length: 891, dtype: int64
```

```
cabin=pd.get_dummies(td['Cabin'])
```

```
print(cabin)
```

```
     A10  A14  A16  A19  A20  A23  A24  A26  A31  A32  ...  F E69  F G63  \
0      0    0    0    0    0    0    0    0    0    0  ...      0      0
```

```
1      0    0    0    0    0    0    0    0    0    0  ...      0      0
2      0    0    0    0    0    0    0    0    0    0  ...      0      0
3      0    0    0    0    0    0    0    0    0    0  ...      0      0
4      0    0    0    0    0    0    0    0    0    0  ...      0      0
..   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...    ...    ...
886    0    0    0    0    0    0    0    0    0    0  ...      0      0
887    0    0    0    0    0    0    0    0    0    0  ...      0      0
888    0    0    0    0    0    0    0    0    0    0  ...      0      0
889    0    0    0    0    0    0    0    0    0    0  ...      0      0
890    0    0    0    0    0    0    0    0    0    0  ...      0      0

     F G73  F2  F33  F38  F4  G  G6  T
0        0   0    0    0   0  1   0  0
1        0   0    0    0   0  0   0  0
2        0   0    0    0   0  0   0  0
3        0   0    0    0   0  0   0  0
4        0   0    0    0   0  1   0  0
..     ...  ..  ...  ...  .. ..  .. ..
886      0   0    0    0   0  0   0  0
887      0   0    0    0   0  0   0  0
888      0   0    0    0   0  1   0  0
889      0   0    0    0   0  0   0  0
890      0   0    0    0   0  1   0  0

[891 rows x 152 columns]
```

```
td.drop(["Pclass","Sex","Embarked","Cabin"],axis=1,inplace=True)
```

```
td=pd.concat([td,cabin],axis=1)
```

```
print(td.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Columns: 157 entries, Survived to T
dtypes: float64(2), int64(3), uint8(152)
memory usage: 167.2 KB
None
```

```
X=td.drop(["Survived"],axis=1)
```

```
print(X)
```

```
           Age  SibSp  Parch     Fare  A10  A14  A16  A19  A20  A23  ...  \
0    22.000000      1      0   7.2500    0    0    0    0    0    0  ...
1    38.000000      1      0  71.2833    0    0    0    0    0    0  ...
2    26.000000      0      0   7.9250    0    0    0    0    0    0  ...
3    35.000000      1      0  53.1000    0    0    0    0    0    0  ...
4    35.000000      0      0   8.0500    0    0    0    0    0    0  ...
```

```
..          ...    ...    ...      ...  ...  ...  ...  ...  ...  ...  ...
886  27.000000      0      0  13.0000    0    0    0    0    0    0  ...
887  19.000000      0      0  30.0000    0    0    0    0    0    0  ...
888  29.699118      1      2  23.4500    0    0    0    0    0    0  ...
889  26.000000      0      0  30.0000    0    0    0    0    0    0  ...
890  32.000000      0      0   7.7500    0    0    0    0    0    0  ...

     F E69  F G63  F G73  F2  F33  F38  F4  G  G6  T
0        0      0      0   0    0    0   0  1   0  0
1        0      0      0   0    0    0   0  0   0  0
2        0      0      0   0    0    0   0  0   0  0
3        0      0      0   0    0    0   0  0   0  0
4        0      0      0   0    0    0   0  1   0  0
..     ...    ...    ...  ..  ...  ...  .. ..  .. ..
886      0      0      0   0    0    0   0  0   0  0
887      0      0      0   0    0    0   0  0   0  0
888      0      0      0   0    0    0   0  1   0  0
889      0      0      0   0    0    0   0  0   0  0
890      0      0      0   0    0    0   0  1   0  0

[891 rows x 156 columns]
```

```
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Columns: 156 entries, Age to T
dtypes: float64(2), int64(2), uint8(152)
memory usage: 160.2 KB
```

```
y=td["Survived"]
```

```
print(y)
```

```
0      0
1      1
2      1
3      1
4      0
      ..
886    0
887    1
888    0
889    1
890    0
Name: Survived, Length: 891, dtype: int64
```

```
from sklearn.model_selection import train_test_split
```

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=1)
```

```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 623 entries, 114 to 37
Columns: 156 entries, Age to T
dtypes: float64(2), int64(2), uint8(152)
memory usage: 116.8 KB
```

```
X_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 268 entries, 862 to 92
Columns: 156 entries, Age to T
dtypes: float64(2), int64(2), uint8(152)
memory usage: 50.2 KB
```

```
from sklearn.linear_model import LogisticRegression
```

```
lm=LogisticRegression(max_iter=10000)
```

```
print(lm.fit(X_train,y_train))
```

```
LogisticRegression(max_iter=10000)
```

```
Predections=lm.predict(X_test)
```

```
from sklearn.metrics import classification_report
```

```
print(classification_report(y_test,Predections))
```

```
              precision    recall  f1-score   support

           0       0.98      0.90      0.94       153
           1       0.88      0.97      0.92       115

    accuracy                           0.93       268
   macro avg       0.93      0.93      0.93       268
weighted avg       0.93      0.93      0.93       268
```

```
from sklearn.metrics import confusion_matrix
```

```
print(confusion_matrix(y_test,Predections))
```

```
[[137  16]
 [  3 112]]
```

```
from sklearn.metrics import accuracy_score
```

```
print(accuracy_score(y_test,Predections))
```

```
0.9291044776119403
```

Dataset Observations:

1. Survival Rate Observations

- Only ~38% of passengers survived, while ~62% perished.
- Women had a much higher survival rate (~74%) compared to men (~18%).
- Children (age < 10) had a better chance of survival than adults.

1. Gender and Survival Insights

- ~74% of females survived, while only ~18% of males survived.
- "Women and children first" policy led to a higher survival rate for women.
- Male survival was much lower, especially in 3rd Class (~13%).

1. Summary:

- Higher class, higher fare, and being female or a child greatly increased survival chances.
- Men in 3rd class had the worst survival rate (~13%).
- Passengers with large families struggled to survive.

Dropping irrelevant columns (Name, PassengerId, Ticket) Handling missing values by filling missing Age values with the mean. However, I still haven't located the model building and evaluation code. Let me extract more cells to find the relevant information.

The extracted cells still show data cleaning and preprocessing steps, such as:

Filling missing Cabin values based on Survived and Pclass. Filling missing Embarked values with "S". Printing some of the dataset's categorical features (Sex, Embarked, Pclass).

One-hot encoding the Cabin column using pd.get_dummies(). Dropping categorical features (Pclass, Sex, Embarked, and Cabin) and replacing them with the encoded versions. Defining the features (X) and the target (y): X $\rightarrow$ Contains the independent variables (features). y $\rightarrow$ Contains the target variable (Survived).