```
# import libraries

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

```
import warnings

# Set the warning filter to 'ignore'
warnings.filterwarnings('ignore')
```

```
# read data set

df = pd.read_csv("/content/Titanic-Dataset.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
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

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.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
```

```
df.shape
```

```
(891, 12)
```

```
df.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

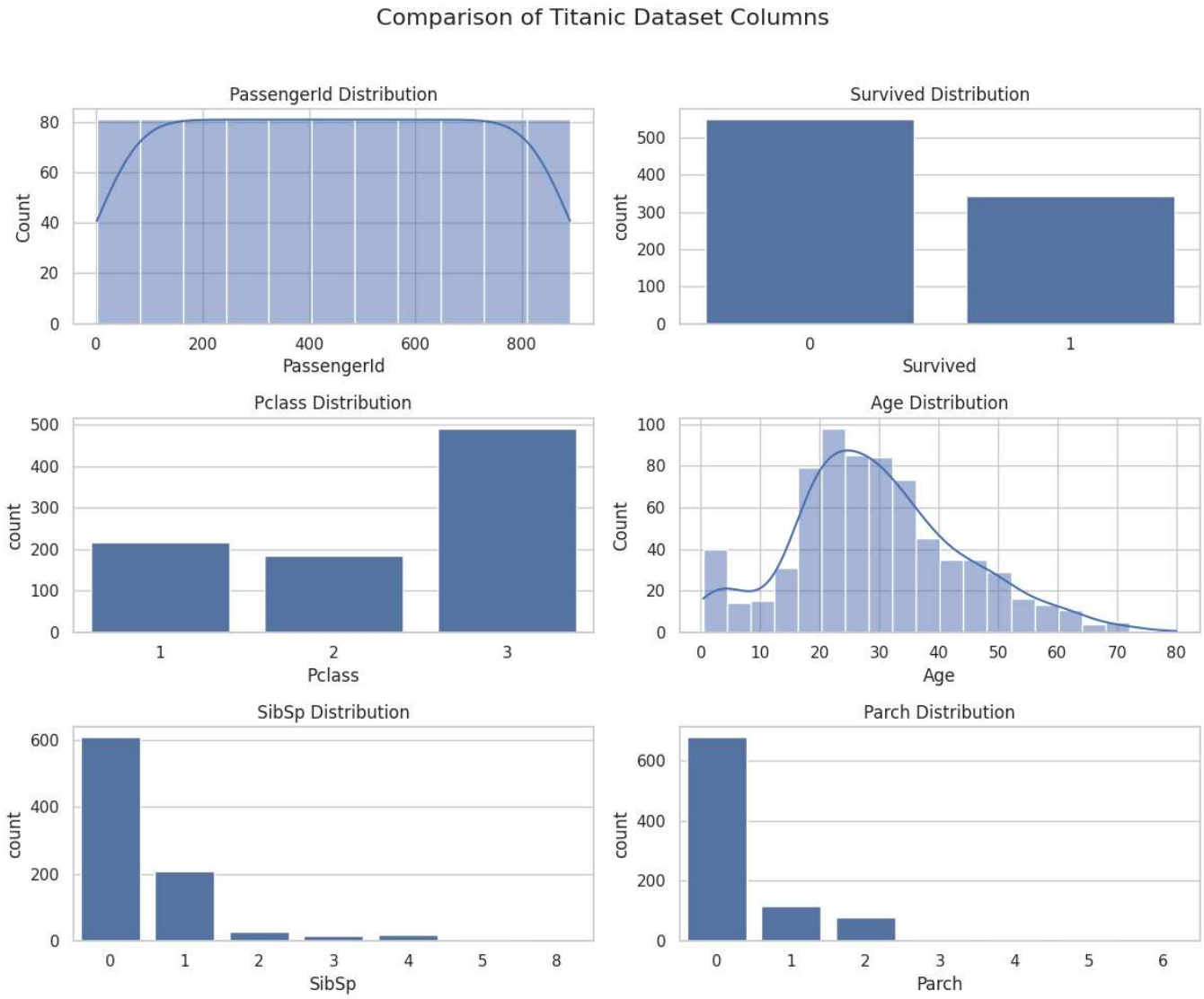
```
df.isnull().sum()
```

	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

```
df.duplicated().sum()
```

```
0
```

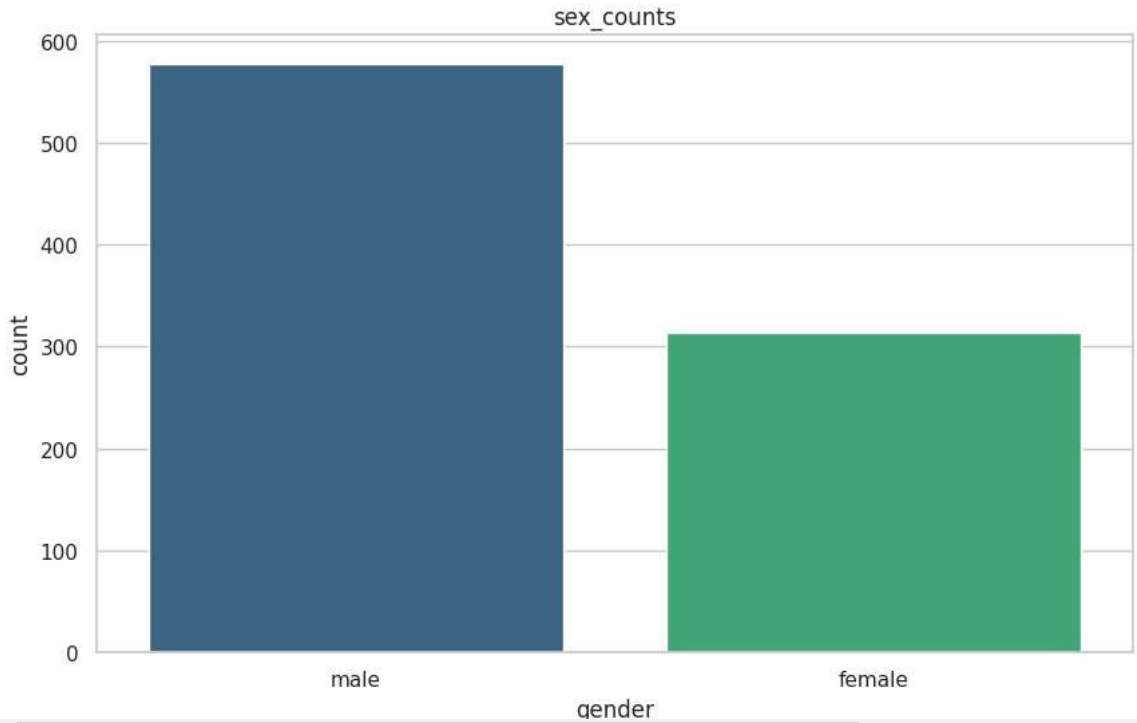
```
# Set the style for seaborn
sns.set(style="whitegrid")

# Set up the subplots
fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(12, 10))
fig.suptitle('Comparison of Titanic Dataset Columns', fontsize=16)

# Visualization for PassengerId
sns.histplot(df['PassengerId'], kde=True, ax=axes[0, 0])
axes[0, 0].set_title('PassengerId Distribution')

# Visualization for Survived
sns.countplot(x='Survived', data=df, ax=axes[0, 1])
axes[0, 1].set_title('Survived Distribution')

# Visualization for Pclass
sns.countplot(x='Pclass', data=df, ax=axes[1, 0])
axes[1, 0].set_title('Pclass Distribution')

# Visualization for Age
sns.histplot(df['Age'].dropna(), kde=True, ax=axes[1, 1])
axes[1, 1].set_title('Age Distribution')

# Visualization for SibSp
sns.countplot(x='SibSp', data=df, ax=axes[2, 0])
axes[2, 0].set_title('SibSp Distribution')

# Visualization for Parch
sns.countplot(x='Parch', data=df, ax=axes[2, 1])
axes[2, 1].set_title('Parch Distribution')

# Adjust layout
plt.tight_layout(rect=[0, 0, 1, 0.96])

# Show the plots
plt.show()
```



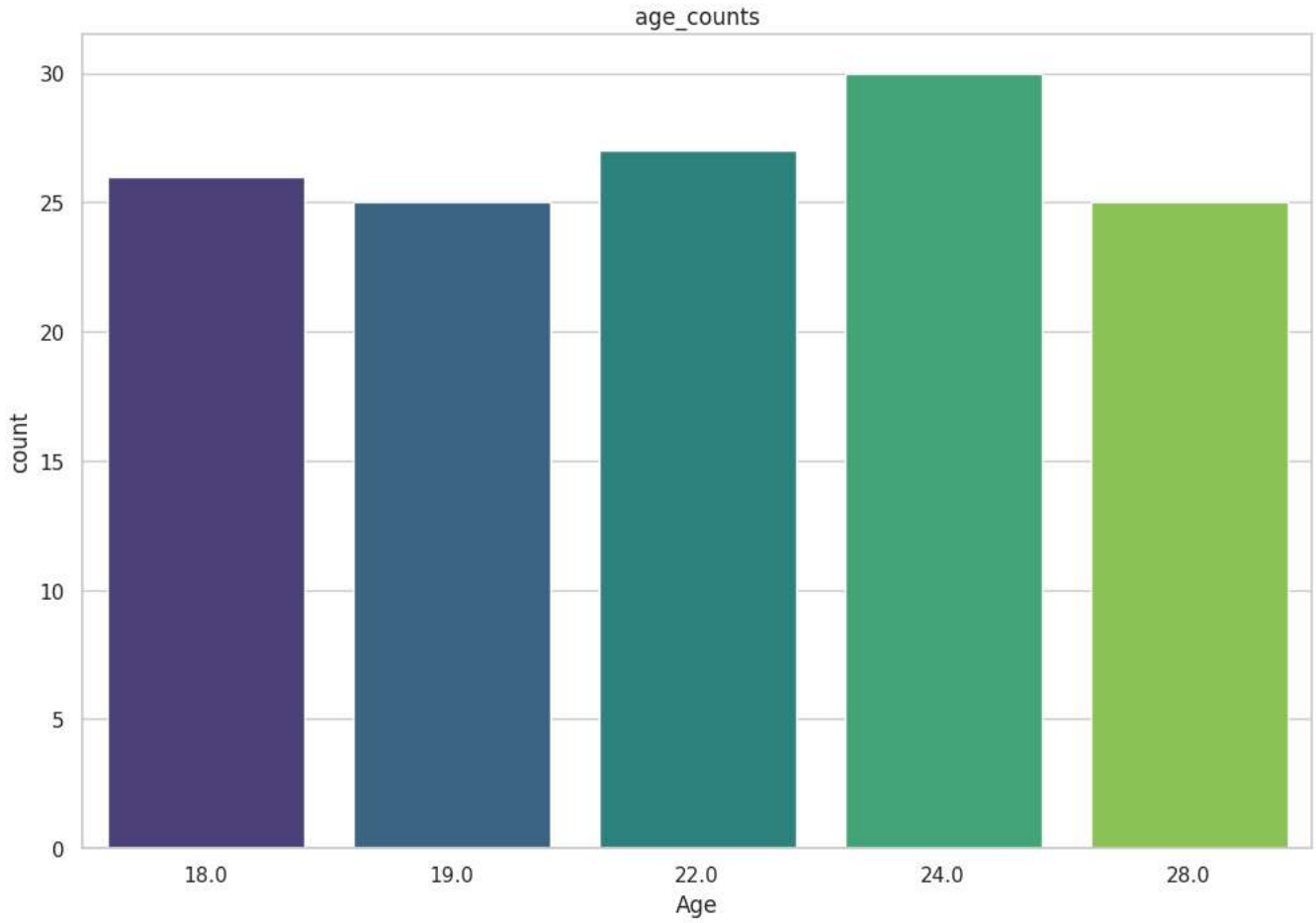

```
sex_counts=df['Sex'].value_counts()
sex_counts
```

Sex	count
male	577
female	314

```
plt.figure(figsize=(10,6))
sns.barplot(x=sex_counts.index,y=sex_counts.values,palette='viridis')
plt.title('sex_counts')
plt.xlabel('gender')
plt.ylabel('count')
plt.show()
```




```
#the most 5 age in data
age_counts=df['Age'].value_counts().head()

plt.figure(figsize=(12,8))
sns.barplot(x=age_counts.index,y=age_counts.values,palette='viridis')
plt.title('age_counts')
plt.xlabel('Age')
plt.ylabel('count')
plt.show()
age_counts
```

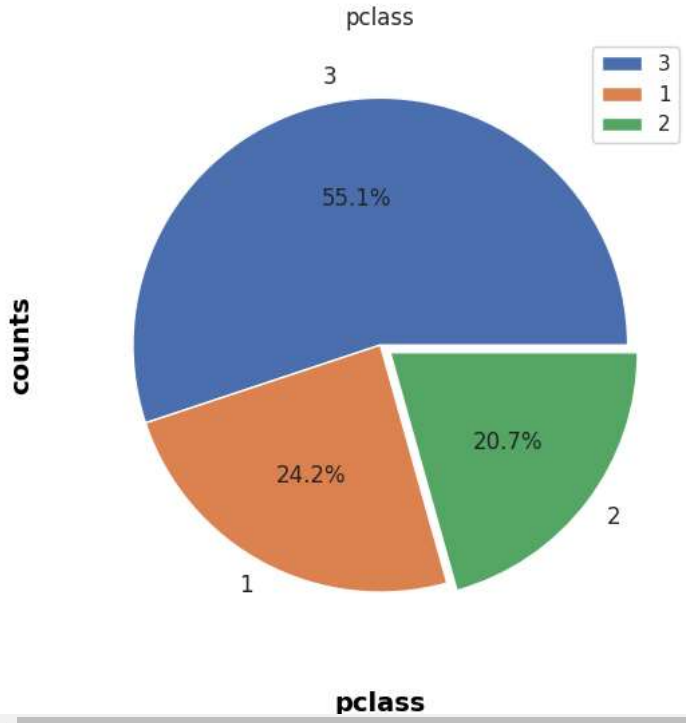


Age	count
24.0	30
22.0	27
18.0	26
19.0	25
28.0	25

```
pclass_counts=df['Pclass'].value_counts()
pclass_counts
```

Pclass	count
3	491
1	216
2	184

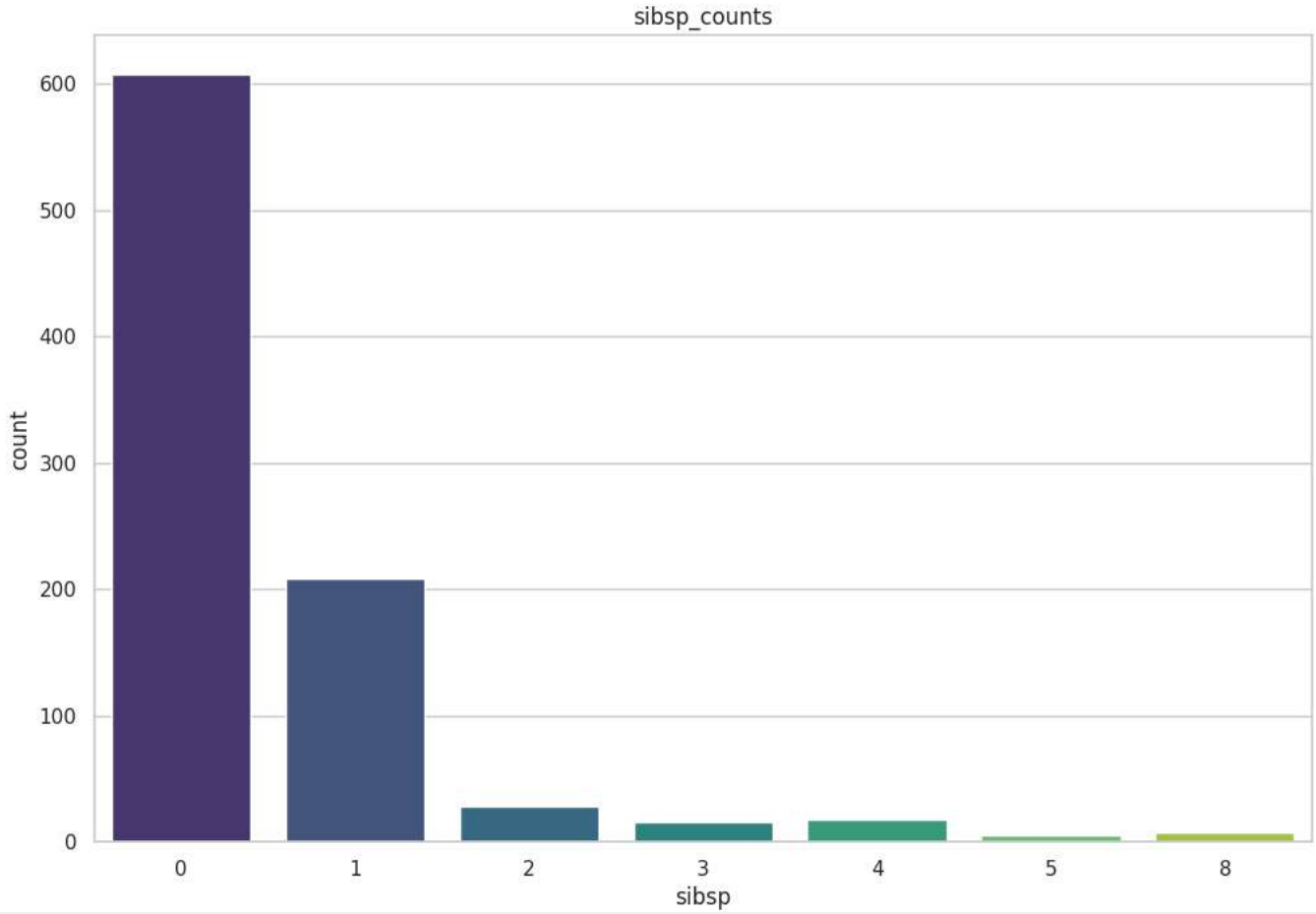
```
plt.figure(figsize = (20, 6))
explode = (0,0,0.05)
pclass_counts.plot(kind = 'pie', fontsize = 12, explode = explode, autopct = '%.1f%%')
plt.title('pclass')
plt.xlabel('pclass', weight = "bold", color = "#000000", fontsize = 14, labelpad = 20)
plt.ylabel('counts', weight = "bold", color = "#000000", fontsize = 14, labelpad = 20)
plt.legend(labels = pclass_counts.index, loc = "best")
plt.show()
```




```
SibSp_counts=df['SibSp'].value_counts()
SibSp_counts
```

SibSp	count
0	608
1	209
2	28
4	18
3	16
8	7
5	5

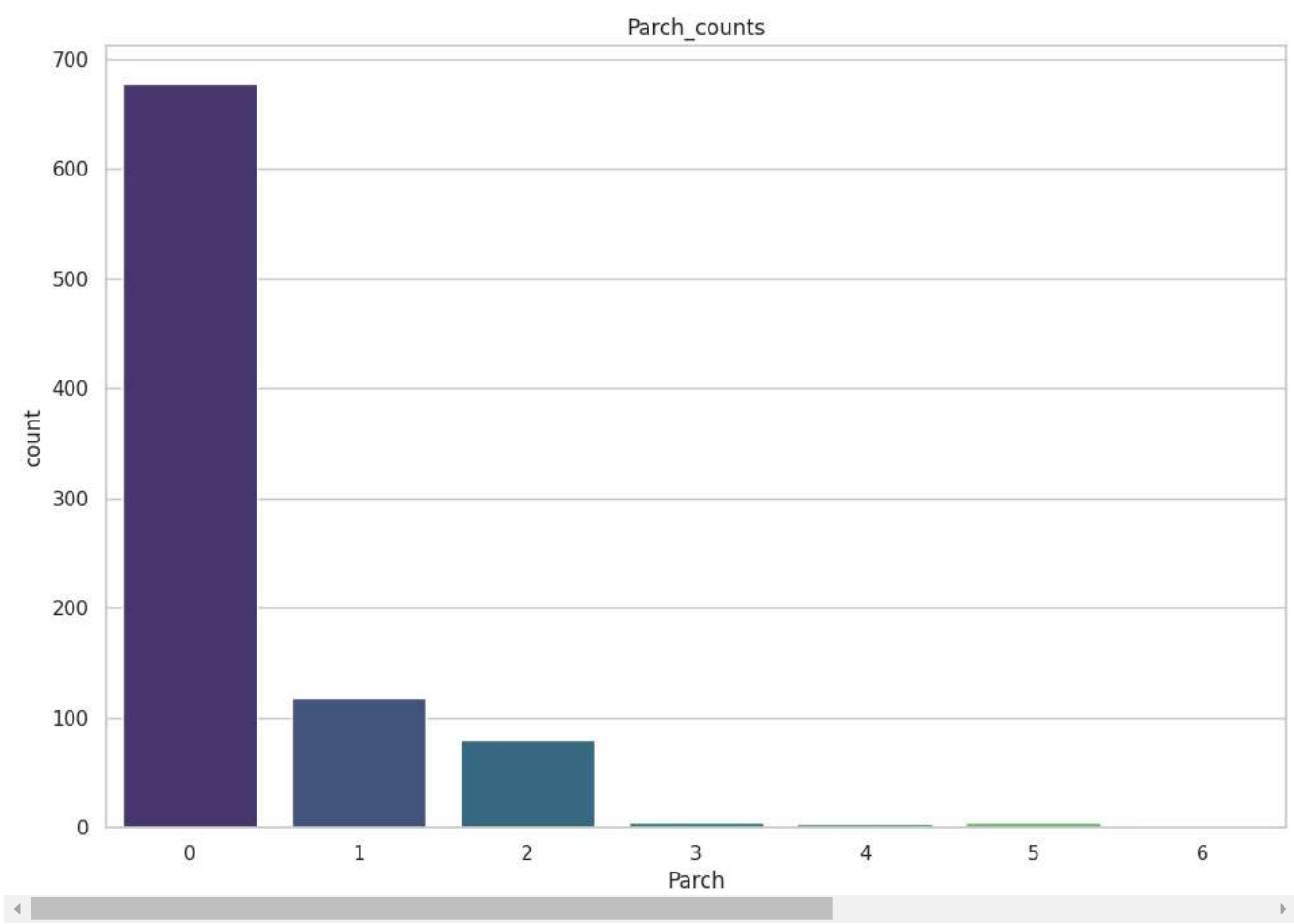
```
plt.figure(figsize=(12,8))
sns.barplot(x=SibSp_counts.index,y=SibSp_counts.values,palette='viridis')
plt.title('sibsp_counts')
plt.xlabel('sibsp')
plt.ylabel('count')
plt.show()
```




```
Parch_counts=df['Parch'].value_counts()
Parch_counts
```

Parch	count
0	678
1	118
2	80
5	5
3	5
4	4
6	1

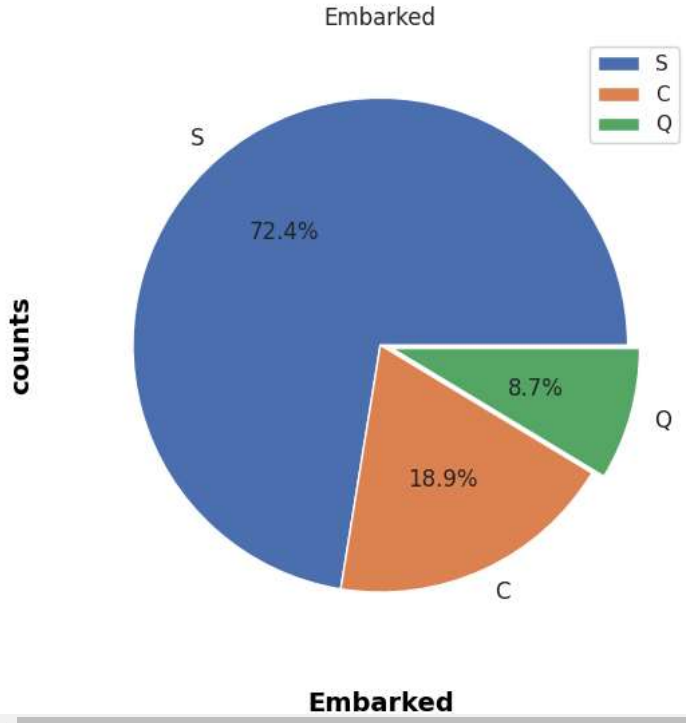
```
plt.figure(figsize=(12,8))
sns.barplot(x=Parch_counts.index,y=Parch_counts.values,palette='viridis')
plt.title('Parch_counts')
plt.xlabel('Parch')
plt.ylabel('count')
plt.show()
```




```
Embarked_counts=df['Embarked'].value_counts()
Embarked_counts
```

	count
Embarked	
S	644
C	168
Q	77

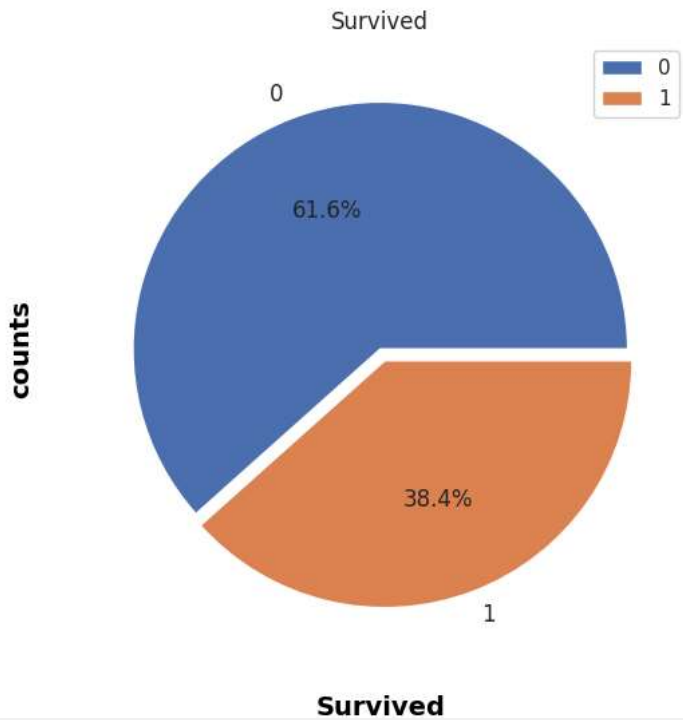
```
plt.figure(figsize = (20, 6))
explode = (0,0,0.05)
Embarked_counts.plot(kind = 'pie', fontsize = 12, explode = explode, autopct = '%.1f%%')
plt.title('Embarked')
plt.xlabel('Embarked', weight = "bold", color = "#000000", fontsize = 14, labelpad = 20)
plt.ylabel('counts', weight = "bold", color = "#000000", fontsize = 14, labelpad = 20)
plt.legend(labels = Embarked_counts.index, loc = "best")
plt.show()
```




```
Survived_counts=df['Survived'].value_counts()
Survived_counts
```

	count
Survived	
0	549
1	342

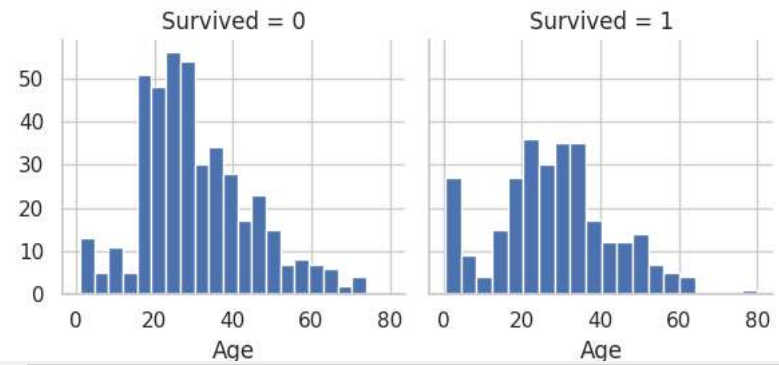
```
plt.figure(figsize = (20, 6))
explode = (0,0.05)
Survived_counts.plot(kind = 'pie', fontsize = 12, explode = explode, autopct = '%.1f%%')
plt.title('Survived')
plt.xlabel('Survived', weight = "bold", color = "#000000", fontsize = 14, labelpad = 20)
plt.ylabel('counts', weight = "bold", color = "#000000", fontsize = 14, labelpad = 20)
plt.legend(labels = Survived_counts.index, loc = "best")
plt.show()
```



visulize ages are survived or not

```
age=sns.FacetGrid(df,col='Survived')
age.map(plt.hist,'Age',bins=20)
```

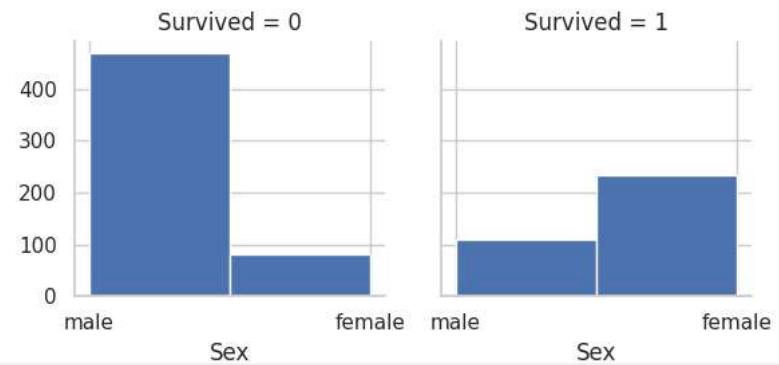
```
<seaborn.axisgrid.FacetGrid at 0x79fa3c22ba60>
```



visulize Gender are survived or not

```
gender=sns.FacetGrid(df,col='Survived')

gender.map(plt.hist,'Sex',bins=2)
```

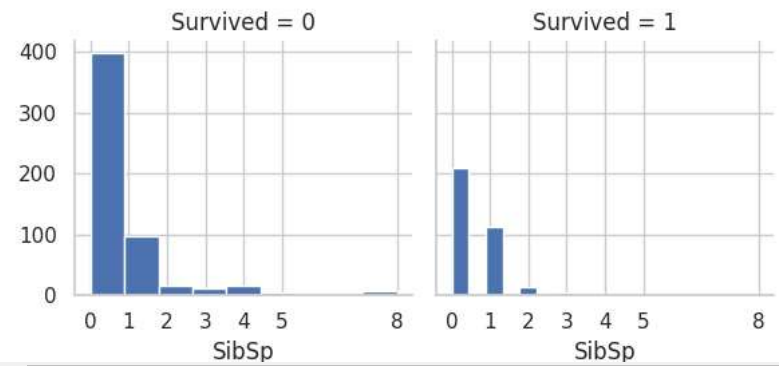
```
<seaborn.axisgrid.FacetGrid at 0x79fa3c106050>
```



visulize sibsp is survived or not

```
sibsp=sns.FacetGrid(df,col='Survived')
plt.xticks(SibSp_counts.index)
sibsp.map(plt.hist,'SibSp',bins=9)
```

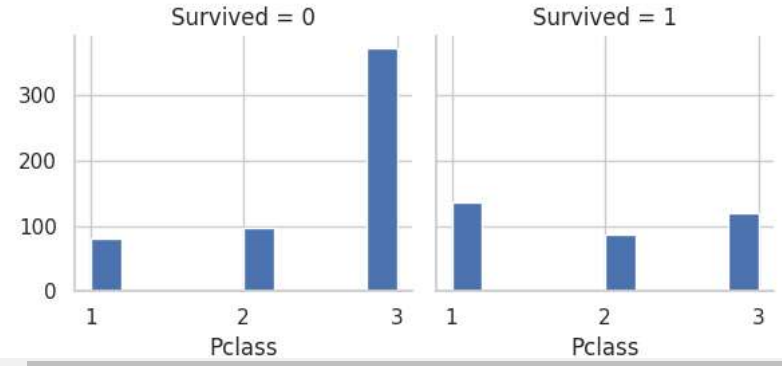
```
<seaborn.axisgrid.FacetGrid at 0x79fa3cfbc250>
```



visulize pclass is survived or not

```
pclass=sns.FacetGrid(df,col='Survived')
plt.xticks([1,2,3])
pclass.map(plt.hist,'Pclass')
```

```
<seaborn.axisgrid.FacetGrid at 0x79fa3ceb9810>
```

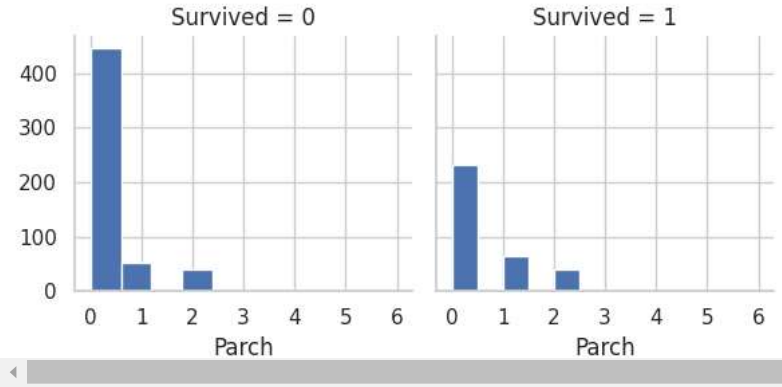


Start coding or generate with AI.

visulize Parch is survived or notParch

```
pclass=sns.FacetGrid(df,col='Survived')
plt.xticks(Parch_counts.index)
pclass.map(plt.hist,'Parch')
```

```
<seaborn.axisgrid.FacetGrid at 0x79fa4504f850>
```




```
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
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

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
# Data cleaning: Convert 'Age' column to numeric, handling errors with coerce
df['Age'] = pd.to_numeric(df['Age'], errors='coerce')
```

```
sns.histplot(df['Age'].dropna(), kde=True, ax=axes[1, 1])
axes[1, 1].set_title('Age Distribution')
```

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

```
# Drop rows with NaN values in the 'Age' column
df = df.dropna(subset=['Age'])
```

```
### convert the gender to binary 0 and 1
df['Sex']=df['Sex'].replace({'male':1,'female':0})
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	53.1000	C123	S

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
# Fill missing values in age column by imputing the median
df['Age'].fillna(df['Age'].median(), inplace=True)
df.isna().sum()
```

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	0
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	529
Embarked	2

```
# Fill missing values in embarked column by imputing the mode
df["Embarked"].fillna(df["Embarked"].mode()[0], inplace=True)
df.isna().sum()
```

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	0
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	529
Embarked	0

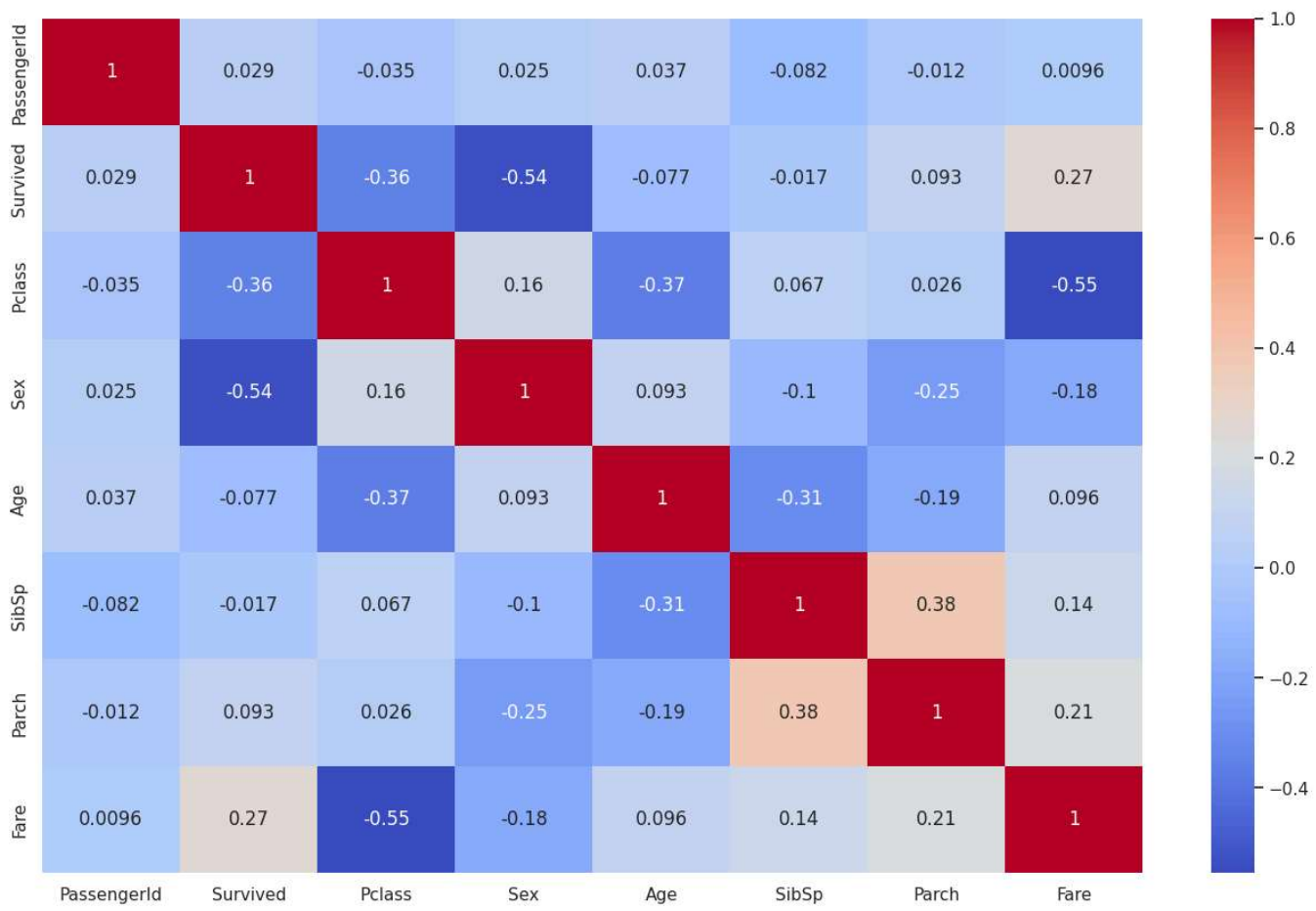
```
df_numeric = df.select_dtypes(include=['number'])  # Select only numeric columns
correlation_matrix = df_numeric.corr()
```

```
df_numeric = df.select_dtypes(include=['float64', 'int64'])
correlation_matrix = df_numeric.corr()
```

```
non_numeric_columns = df.select_dtypes(exclude=['float64', 'int64']).columns
print("Non-numeric columns:", non_numeric_columns)
```

```
Non-numeric columns: Index(['Name', 'Ticket', 'Cabin', 'Embarked'], dtype='object')
```

```
plt.figure(figsize=(16, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.show()
```



Default title text

```
# @title Default title text
df['Embarked']=df['Embarked'].replace({'S':1,'C':2,'Q':3})
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	1	22.0	1	0	A/5 21171	7.2500	NaN	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	38.0	1	0	PC 17599	71.2833	C85	2
2	3	1	3	Heikkinen, Miss. Laina	0	26.0	0	0	STON/O2. 3101282	7.9250	NaN	1
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	35.0	1	0	113803	53.1000	C123	1

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
x=df.drop(['Name','Survived','Cabin','PassengerId','Ticket'],axis=1)
y=df['Survived']
```

```
x.head()
```

	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	1	22.0	1	0	7.2500	1