

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Assuming the 'train.csv' file is in the current working directory
data = pd.read_csv('/content/drive/MyDrive/train.csv')

# Viewing the first few rows of the dataset
print(data.head())

# Checking the basic statistics of numerical columns
print(data.describe())

# Checking the data types of each column
print(data.info())

# Checking the number of missing values in each column
print(data.isnull().sum())

```

	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

	PassengerId	Survived	Pclass	Age	SibSp
count	891.000000	891.000000	891.000000	714.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008
std	257.353842	0.486592	0.836071	14.526497	1.102743
min	1.000000	0.000000	1.000000	0.420000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000
50%	446.000000	0.000000	3.000000	28.000000	0.000000
75%	668.500000	1.000000	3.000000	38.000000	1.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000

	Parch	Fare
count	891.000000	891.000000
mean	0.381594	32.204208
std	0.806057	49.693429
min	0.000000	0.000000
25%	0.000000	7.910400
50%	0.000000	14.454200
75%	0.000000	31.000000
max	6.000000	512.329200

```

<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

```

```

Name      0
Sex        0
Age       177
SibSp      0
Parch      0
Ticket     0
Fare       0
Cabin     687
Embarked   2
dtype: int64

```

Question 1: What is the average age of male passengers who survived?

```

average_age_male_survived = data[(data['Sex'] == 'male') & (data['Survived'] == 1)]['Age'].mean()
print("Average age of male passengers who survived: {:.2f}".format(average_age_male_survived))

```

Question 2: What is the survival rate of passengers based on the number of siblings/spouses they had?

```

survival_rate_per_sibsp = data.groupby('SibSp')['Survived'].mean() * 100
print("Survival rate of passengers based on the number of siblings/spouses:")
print(survival_rate_per_sibsp)

```

Question 3: How many passengers had a fare above the 75th percentile?

```

fare_75th_percentile = data['Fare'].quantile(0.75)
passengers_above_75th_percentile = data[data['Fare'] > fare_75th_percentile].shape[0]
print("Number of passengers with a fare above the 75th percentile:", passengers_above_75th_percentile)

```

Question 4: What is the survival rate of passengers with different ticket types (numeric, alphanumeric)?

```

data['TicketType'] = data['Ticket'].str.extract(r'([a-zA-Z]+)')
survival_rate_per_ticket_type = data.groupby('TicketType')['Survived'].mean() * 100
print("Survival rate of passengers based on ticket type:")
print(survival_rate_per_ticket_type)

```

Average age of male passengers who survived: 27.28

Survival rate of passengers based on the number of siblings/spouses:

```

SibSp
0    34.539474
1    53.588517
2    46.428571
3    25.000000
4    16.666667
5     0.000000
8     0.000000

```

Name: Survived, dtype: float64

Number of passengers with a fare above the 75th percentile: 222

Survival rate of passengers based on ticket type:

```

TicketType
A      6.896552
C     45.454545
CA     7.142857
F     66.666667
Fa     0.000000
LINE   25.000000
P     50.000000
PC     65.000000
PP     66.666667
S     14.285714
SC     53.846154
SCO     0.000000
SO    100.000000
SOTON  11.764706
STON   44.444444
SW    100.000000
W      9.090909
WE     50.000000

```

Name: Survived, dtype: float64

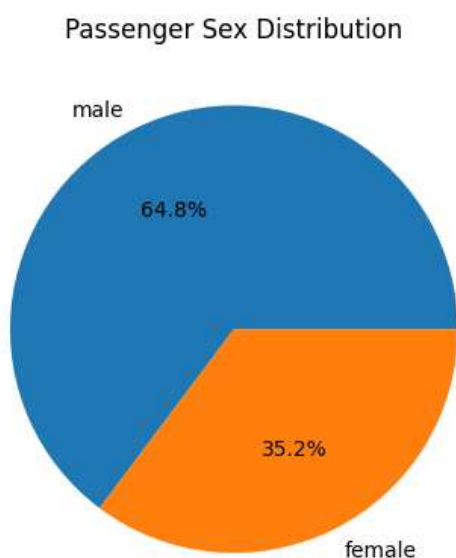
Pie chart for the distribution of passengers by sex

```

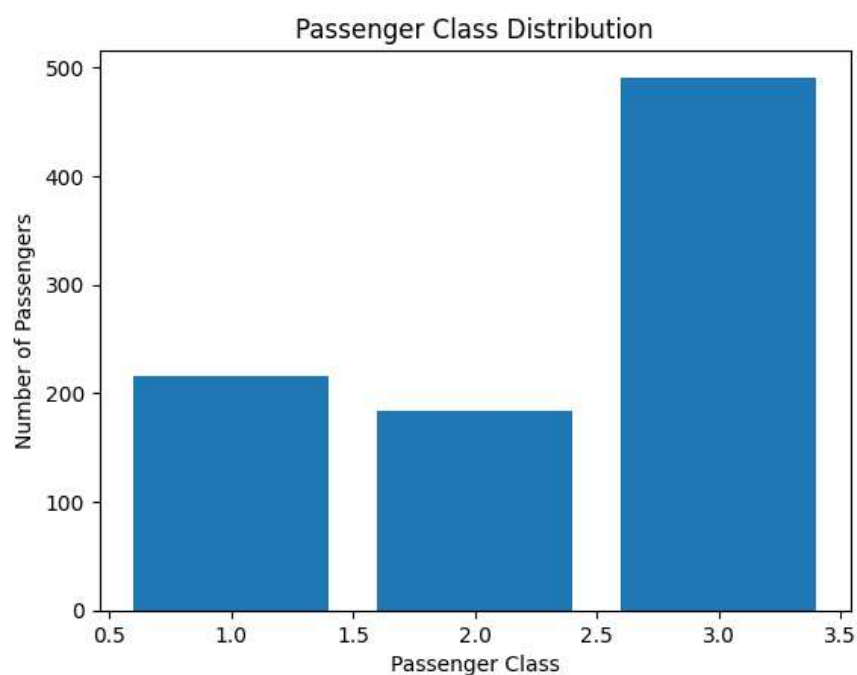
sex_counts = data['Sex'].value_counts()
plt.pie(sex_counts, labels=sex_counts.index, autopct='%1.1f%%')

```

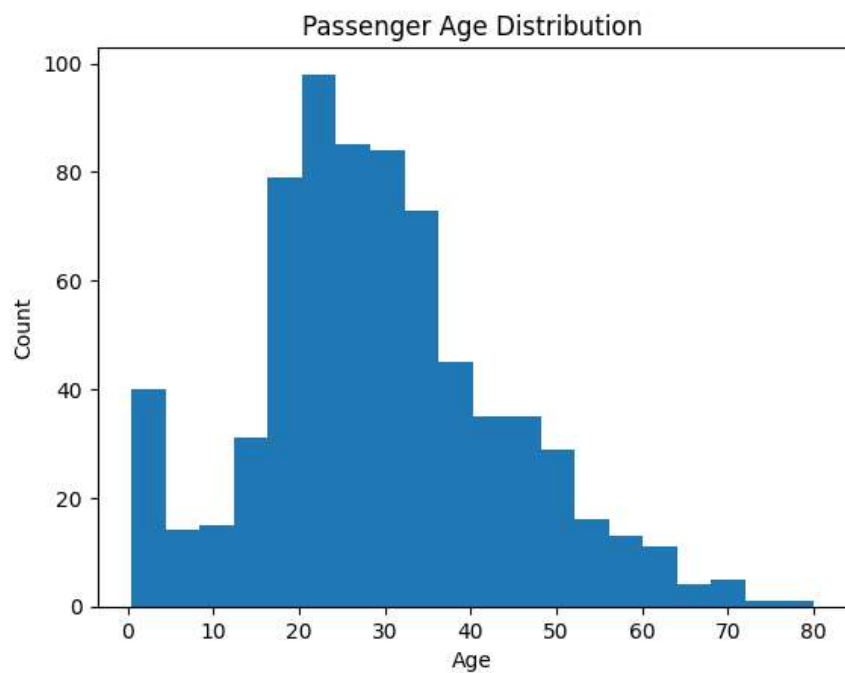
```
plt.title('Passenger Sex Distribution')  
plt.show()
```



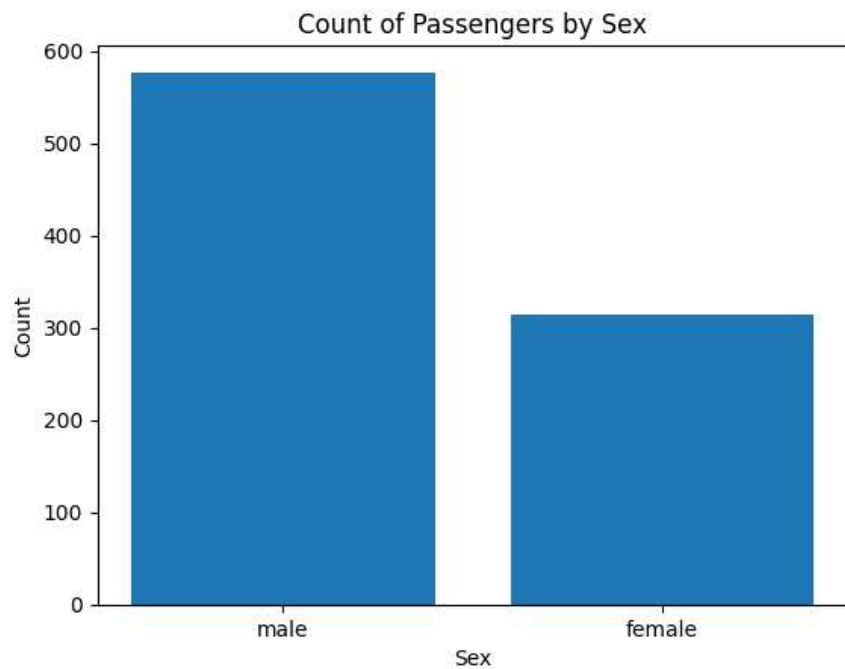
```
# Bar plot for the number of passengers in each passenger class  
class_counts = data['Pclass'].value_counts()  
plt.bar(class_counts.index, class_counts.values)  
plt.xlabel('Passenger Class')  
plt.ylabel('Number of Passengers')  
plt.title('Passenger Class Distribution')  
plt.show()
```



```
# Histogram of passenger ages  
plt.hist(data['Age'].dropna(), bins=20)  
plt.xlabel('Age')  
plt.ylabel('Count')  
plt.title('Passenger Age Distribution')  
plt.show()
```

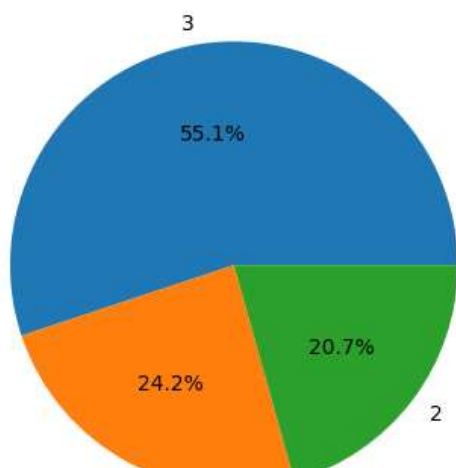


```
# Bar plot for the count of passengers by sex
plt.bar(sex_counts.index, sex_counts.values)
plt.xlabel('Sex')
plt.ylabel('Count')
plt.title('Count of Passengers by Sex')
plt.show()
```



```
# Pie chart for the count of passengers by passenger class
plt.pie(class_counts, labels=class_counts.index, autopct='%1.1f%%')
plt.title('Passenger Class Distribution')
plt.show()
```

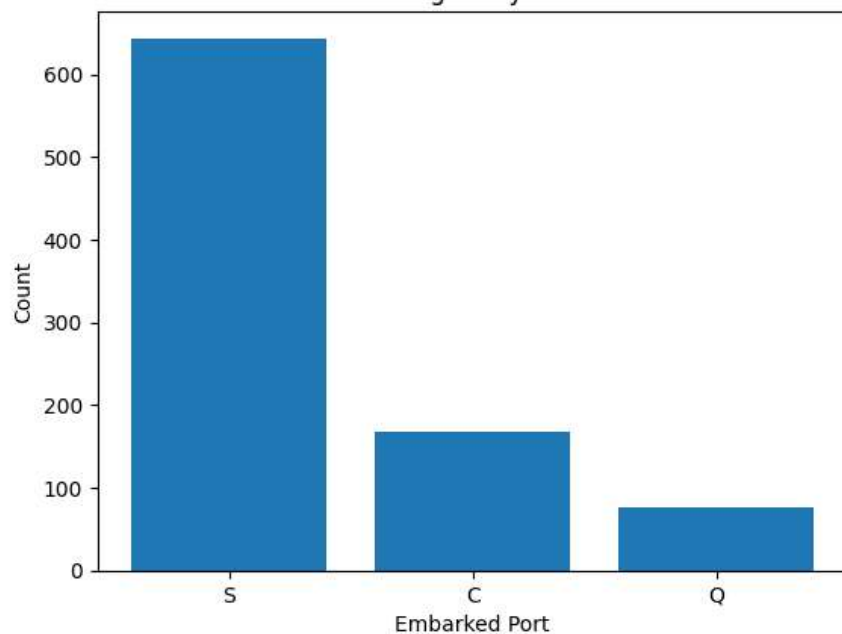
Passenger Class Distribution



```
# Count of passengers by embarked port
embarked_counts = data['Embarked'].value_counts()
print(embarked_counts)
# Bar plot for the count of passengers by embarked port
plt.bar(embarked_counts.index, embarked_counts.values)
plt.xlabel('Embarked Port')
plt.ylabel('Count')
plt.title('Count of Passengers by Embarked Port')
plt.show()
```

```
S    644
C    168
Q     77
Name: Embarked, dtype: int64
```

Count of Passengers by Embarked Port



✓ 0s completed at 11:03 PM

● ×