Task

Tell me about this dataset.

Here is all the data you need: "train.csv"

Data loading

Subtask:

Load the dataset "train.csv" into a pandas DataFrame.

Reasoning: Load the dataset "train.csv" into a pandas DataFrame and display its first 5 rows and shape.

```
import pandas as pd

try:
    df = pd.read_csv('train.csv')
    display(df.head())
    print(df.shape)
except FileNotFoundError:
    print("Error: 'train.csv' not found.")
    df = None
except Exception as e:
    print(f"An error occurred: {e}")
    df = None
```

	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

Data exploration

Subtask:

Explore the loaded dataset to understand its characteristics.

Reasoning: Examine the shape, data types, descriptive statistics, missing values, duplicates, and unique values/frequencies for categorical columns of the DataFrame `df`.

```
# Examine the shape of the DataFrame
print("Shape of the DataFrame:", df.shape)

# Determine the data types of each column
print("\nData Types of each column:\n", df.dtypes)

# Generate descriptive statistics for numerical columns
print("\nDescriptive Statistics:\n", df.describe())

# Identify and count missing values
print("\nMissing Values:\n", df.isnull().sum())

# Check for duplicate rows
print("\nNumber of duplicate rows:", df.duplicated().sum())

# Explore unique values and frequencies for categorical columns
categorical_cols = df.select_dtypes(include=['object']).columns
for col in categorical_cols:
    print(f"\nUnique values and frequencies for '{col}':\n{df[col].value_counts()}")
```

```
Shape of the DataFrame: (891, 12)

Data Types of each column:
 PassengerId      int64
```

```
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object

Descriptive Statistics:
        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

Missing Values:
 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

Number of duplicate rows: 0

Unique values and frequencies for 'Name':
Name
Dooley, Mr. Patrick                                     1
```

Data visualization

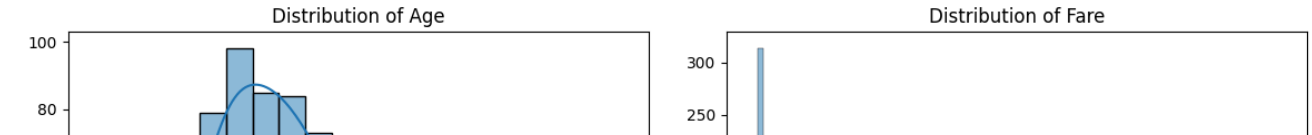
Subtask:

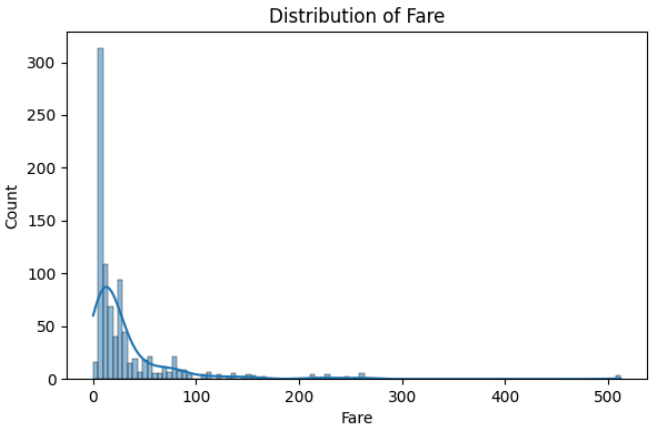
Visualize the distribution of key variables and relationships between them using histograms, box plots, and scatter plots.

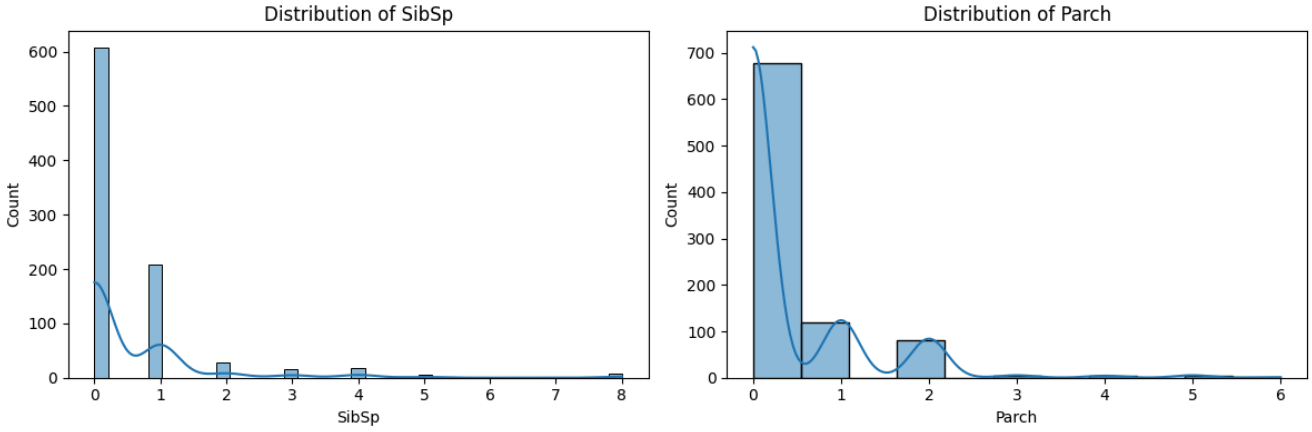
Reasoning: Create histograms, box plots, scatter plots, and bar charts to visualize the distribution of key variables and their relationships, as per the subtask instructions.

```
import matplotlib.pyplot as plt
import seaborn as sns

# Histograms for numerical features
plt.figure(figsize=(12, 8))
plt.subplot(2, 2, 1)
sns.histplot(df['Age'].dropna(), kde=True)
plt.title('Distribution of Age')

plt.subplot(2, 2, 2)
sns.histplot(df['Fare'].dropna(), kde=True)
plt.title('Distribution of Fare')

plt.subplot(2, 2, 3)
sns.histplot(df['SibSp'].dropna(), kde=True)
plt.title('Distribution of SibSp')

plt.subplot(2, 2, 4)
sns.histplot(df['Parch'].dropna(), kde=True)
plt.title('Distribution of Parch')
```

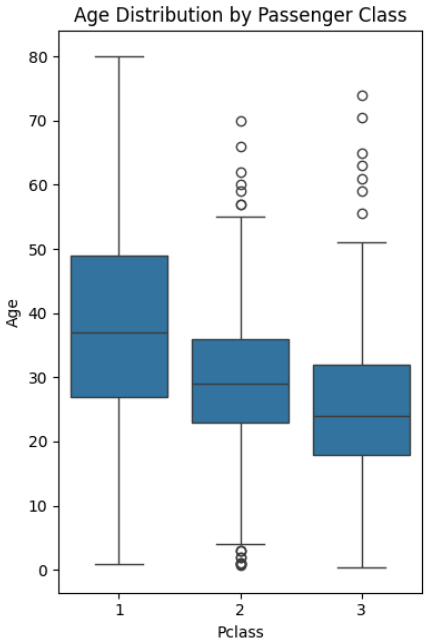
```
plt.tight_layout()
plt.show()


# Box plots for numerical features
plt.figure(figsize=(12, 6))
plt.subplot(1, 3, 1)
sns.boxplot(x='Pclass', y='Age', data=df)
plt.title('Age Distribution by Passenger Class')

plt.subplot(1, 3, 2)
sns.boxplot(x='Pclass', y='Fare', data=df)
plt.title('Fare Distribution by Passenger Class')

plt.subplot(1, 3, 3)
sns.boxplot(x='Pclass', y='SibSp', data=df)
plt.title('SibSp Distribution by Passenger Class')
plt.tight_layout()
plt.show()


# Scatter plots
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=df)
plt.title('Fare vs Age')

plt.subplot(1, 2, 2)
sns.scatterplot(x='Pclass', y='Fare', hue='Survived', data=df)
plt.title('Fare vs Passenger Class')
plt.tight_layout()
plt.show()

# Bar chart for 'Embarked'
plt.figure(figsize=(6, 4))
sns.countplot(x='Embarked', data=df)
plt.title('Number of Passengers Embarked at Each Location')
plt.show()

# Bar chart for 'Sex'
plt.figure(figsize=(4, 4))
sns.countplot(x='Sex', data=df)
plt.title('Number of Passengers by Sex')
plt.show()

# Grouped bar chart for 'Survived' by 'Sex'
plt.figure(figsize=(6, 4))
sns.countplot(x='Sex', hue='Survived', data=df)
plt.title('Survival Count by Sex')
plt.show()
```

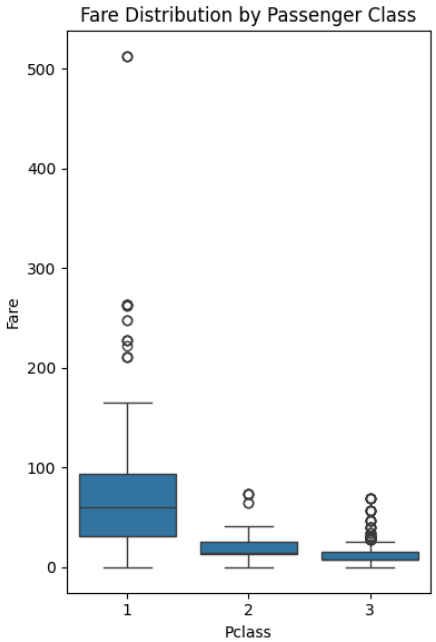

Distribution of Age
Count
Age

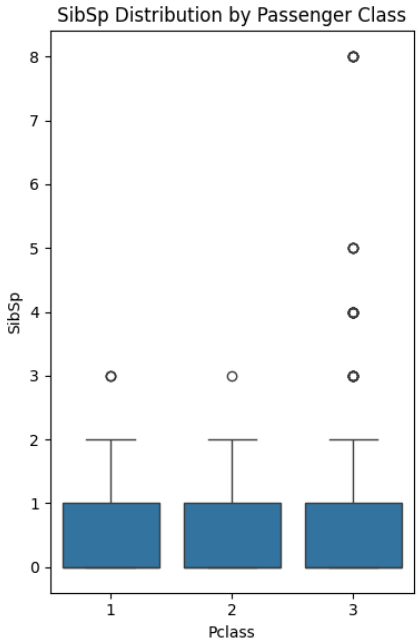

Distribution of Fare
Count
Fare

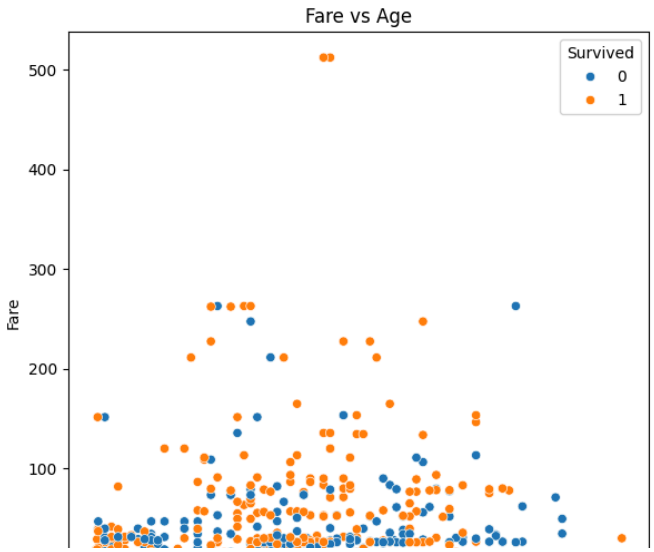

Distribution of SibSp
Count
SibSp

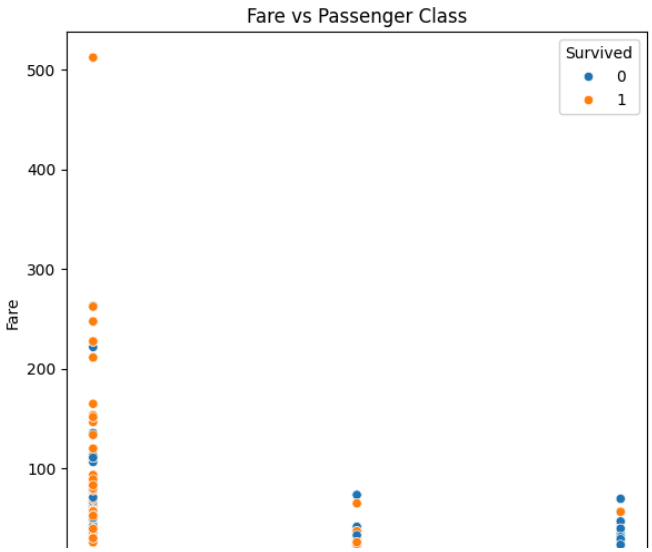
Distribution of Parch
Count
Parch

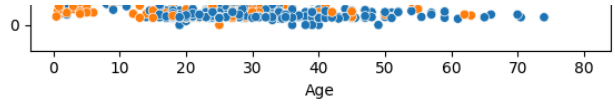

Age Distribution by Passenger Class
Age
Pclass

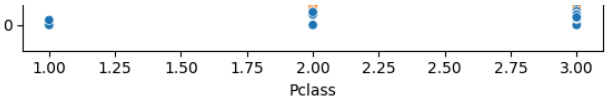

Fare Distribution by Passenger Class
Fare
Pclass

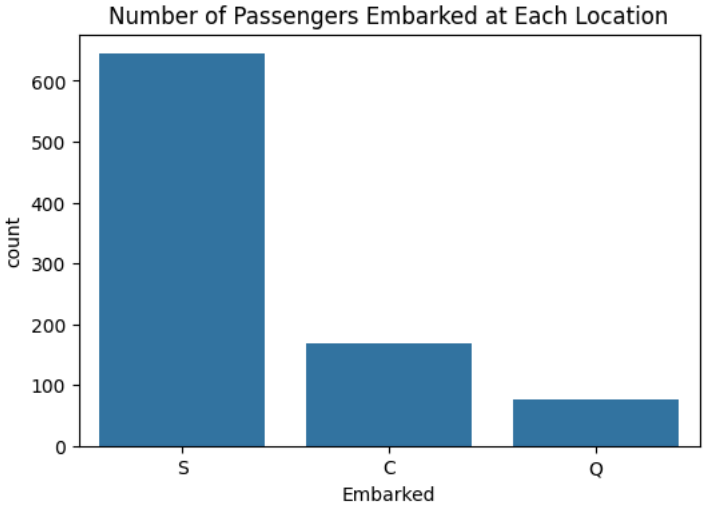

SibSp Distribution by Passenger Class
SibSp
Pclass

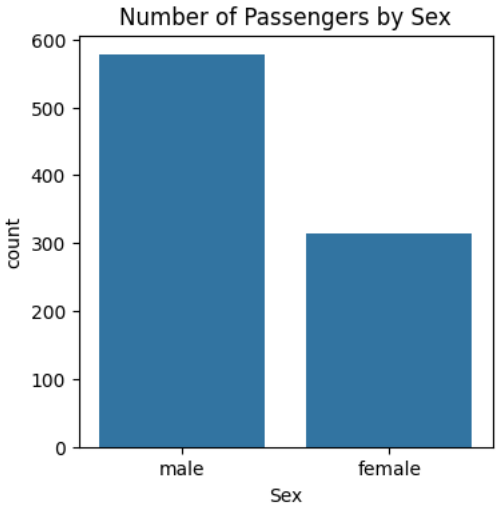

Fare vs Age
Survived
0
1
Fare

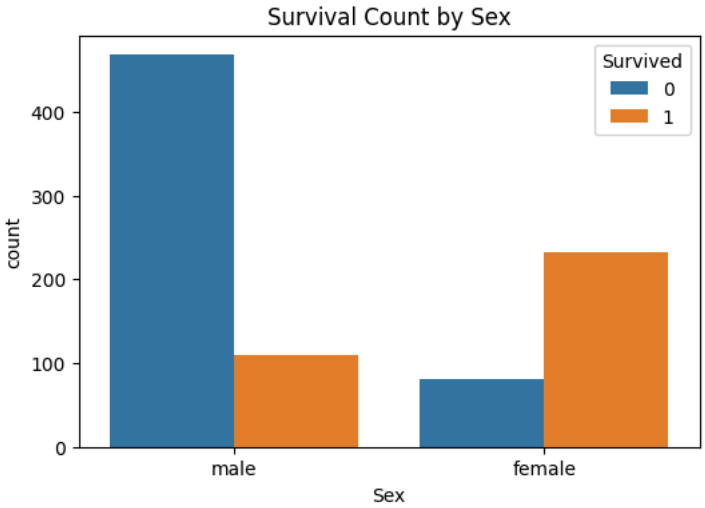

Fare vs Passenger Class
Survived
0
1
Fare


0
0
10
20
30
40
50
60
70
80
Age


0
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
Pclass


Number of Passengers Embarked at Each Location
count
600
500
400
300
200
100
0
S
C
Q
Embarked


Number of Passengers by Sex
count
600
500
400
300
200
100
0
male
female
Sex


Survival Count by Sex
Survived
0
1
count
400
300
200
100
0
male
female
Sex

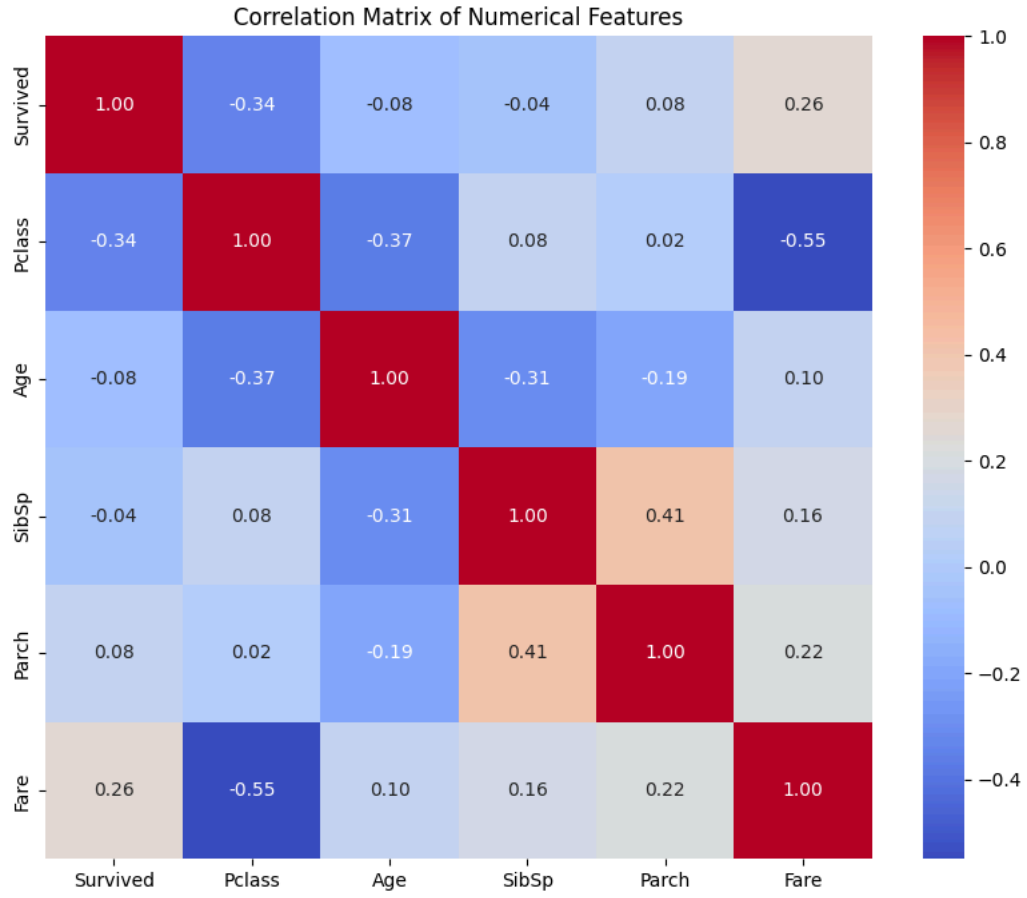
Data analysis

Subtask:

Analyze the data to gain deeper insights.

Reasoning: Calculate the correlation matrix for numerical features, visualize it as a heatmap, and analyze survival rates based on categorical features.

```
import matplotlib.pyplot as plt
import seaborn as sns

# Calculate the correlation matrix
numerical_features = ['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']
correlation_matrix = df[numerical_features].corr()

# Visualize the correlation matrix using a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Numerical Features')
plt.show()

# Analyze survival rates based on categorical features
categorical_features = ['Sex', 'Pclass', 'Embarked']
for feature in categorical_features:
    print(f"\nSurvival Rate by {feature}:")
    print(df.groupby(feature)['Survived'].mean())
```




```
Survival Rate by Sex:
Sex
female    0.742038
male      0.188908
Name: Survived, dtype: float64

Survival Rate by Pclass:
Pclass
1    0.629630
2    0.472826
3    0.242363
Name: Survived, dtype: float64

Survival Rate by Embarked:
Embarked
C    0.553571
Q    0.389610
S    0.336957
```

Reasoning: Investigate the relationship between 'Age' and 'Survived' by creating age bands, calculating survival rates within these bands, and visualizing the results. Then, explore the relationship between 'Fare' and 'Survived', potentially categorizing fares, and visualize this relationship, considering the interaction with 'Pclass'.

```
import matplotlib.pyplot as plt
import seaborn as sns

# Analyze 'Age' and 'Survived'
df['Age_Band'] = pd.cut(df['Age'], bins=[0, 18, 30, 50, 100], labels=['0-18', '18-30', '30-50', '50+'])
print("\nSurvival Rate by Age Band:")
print(df.groupby('Age_Band')['Survived'].mean())
plt.figure(figsize=(8, 6))
sns.barplot(x='Age_Band', y='Survived', data=df)
plt.title('Survival Rate by Age Band')
plt.show()


# Explore 'Fare' and 'Survived'
df['Fare_Band'] = pd.qcut(df['Fare'], 4, labels=['Low', 'Medium', 'High', 'Very High'])
print("\nSurvival Rate by Fare Band:")
print(df.groupby('Fare_Band')['Survived'].mean())
plt.figure(figsize=(8, 6))
sns.barplot(x='Fare_Band', y='Survived', data=df, hue='Pclass')
```