need is a browser.

For example, if you find yourself waiting for **pandas** code to finish running and want to go faster, you can switch to a GPU Runtime and use libraries like RAPIDS cuDF that provide zero-code-change acceleration.

To learn more about accelerating pandas on Colab, see the 10 minute guide or US stock market data analysis demo.

Machine learning

With Colab you can import an image dataset, train an image classifier on it, and evaluate the model, all in just a few lines of code.

Colab is used extensively in the machine learning community with applications including:

- Getting started with TensorFlow
- Developing and training neural networks
- Experimenting with TPUs
- Disseminating AI research
- Creating tutorials

To see sample Colab notebooks that demonstrate machine learning applications, see the machine learning examples below.

More Resources

Working with Notebooks in Colab

- Overview of Colab
- Guide to Markdown
- Importing libraries and installing dependencies
- Saving and loading notebooks in GitHub
- Interactive forms
- Interactive widgets

Working with Data

- Loading data: Drive, Sheets, and Google Cloud Storage
- Charts: visualizing data
- Getting started with BigQuery

Machine Learning Crash Course

These are a few of the notebooks from Google's online Machine Learning course. See the full course website for more.

- Intro to Pandas DataFrame
- Intro to RAPIDS cuDF to accelerate pandas
- Linear regression with tf.keras using synthetic data

Using Accelerated Hardware

- TensorFlow with GPUs
- TPUs in Colab

Featured examples

- Retraining an Image Classifier: Build a Keras model on top of a pre-trained image classifier to distinguish flowers.
- Text Classification: Classify IMDB movie reviews as either *positive* or *negative*.
- Style Transfer: Use deep learning to transfer style between images.
- Multilingual Universal Sentence Encoder Q&A: Use a machine learning model to answer questions from the SQuAD dataset.
- Video Interpolation: Predict what happened in a video between the first and the last frame.

```
import pandas as pd
import numpy as np


df=pd.read_csv('/content/Titanic (2).csv')


df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2
1	2	1	1	Cumings, Mrs. John Bradley (Florence	female	38.0	1	0	PC 17599	71.2

```
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.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fa
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000(
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204:
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693-
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000(
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910-
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454:
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000(
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329:

```
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

dtype: int64

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

Sex	count
male	577
female	314

dtype: int64

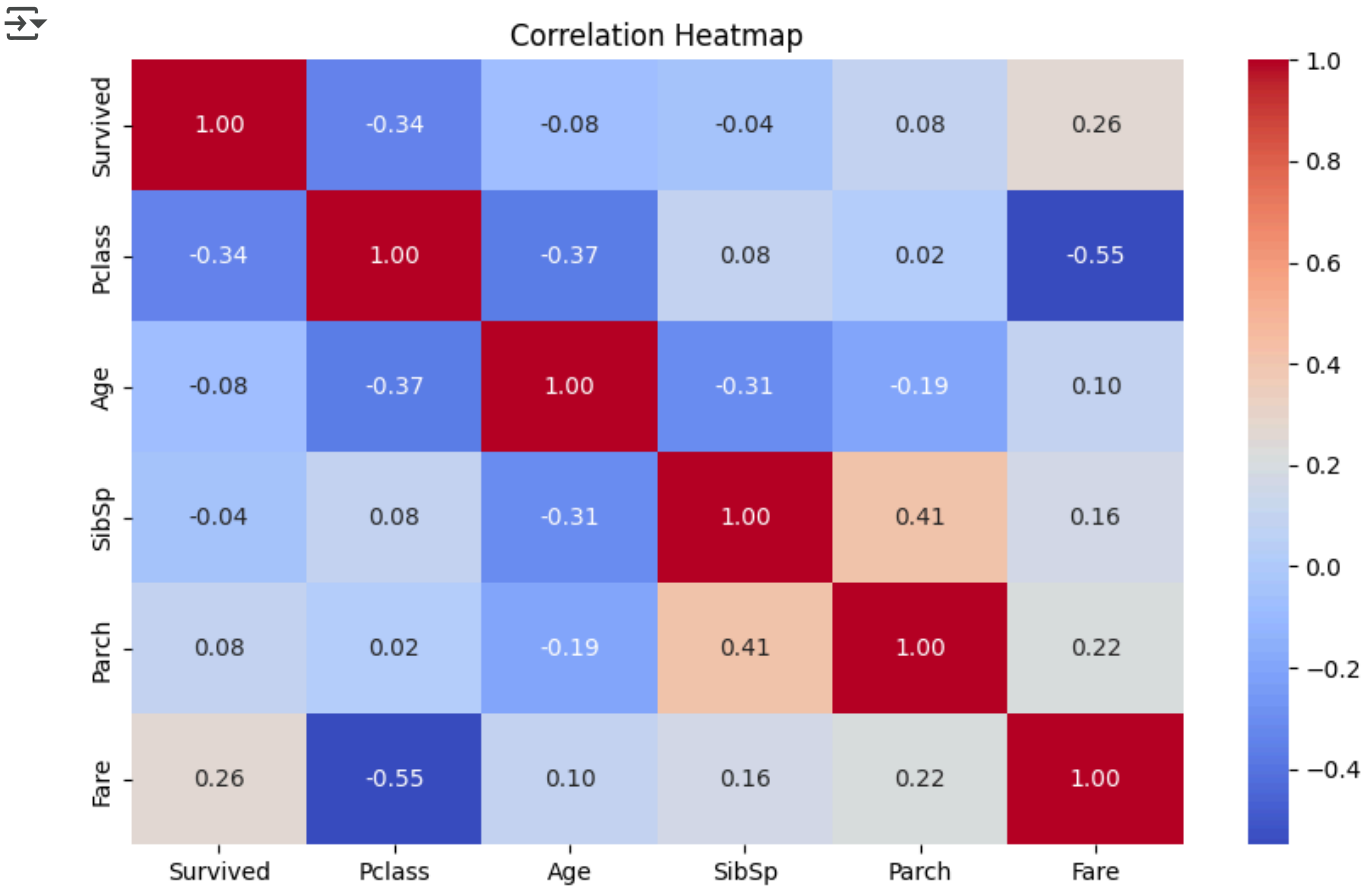
```
df['Survived'].value_counts()
```

Survived	count
0	549
1	342

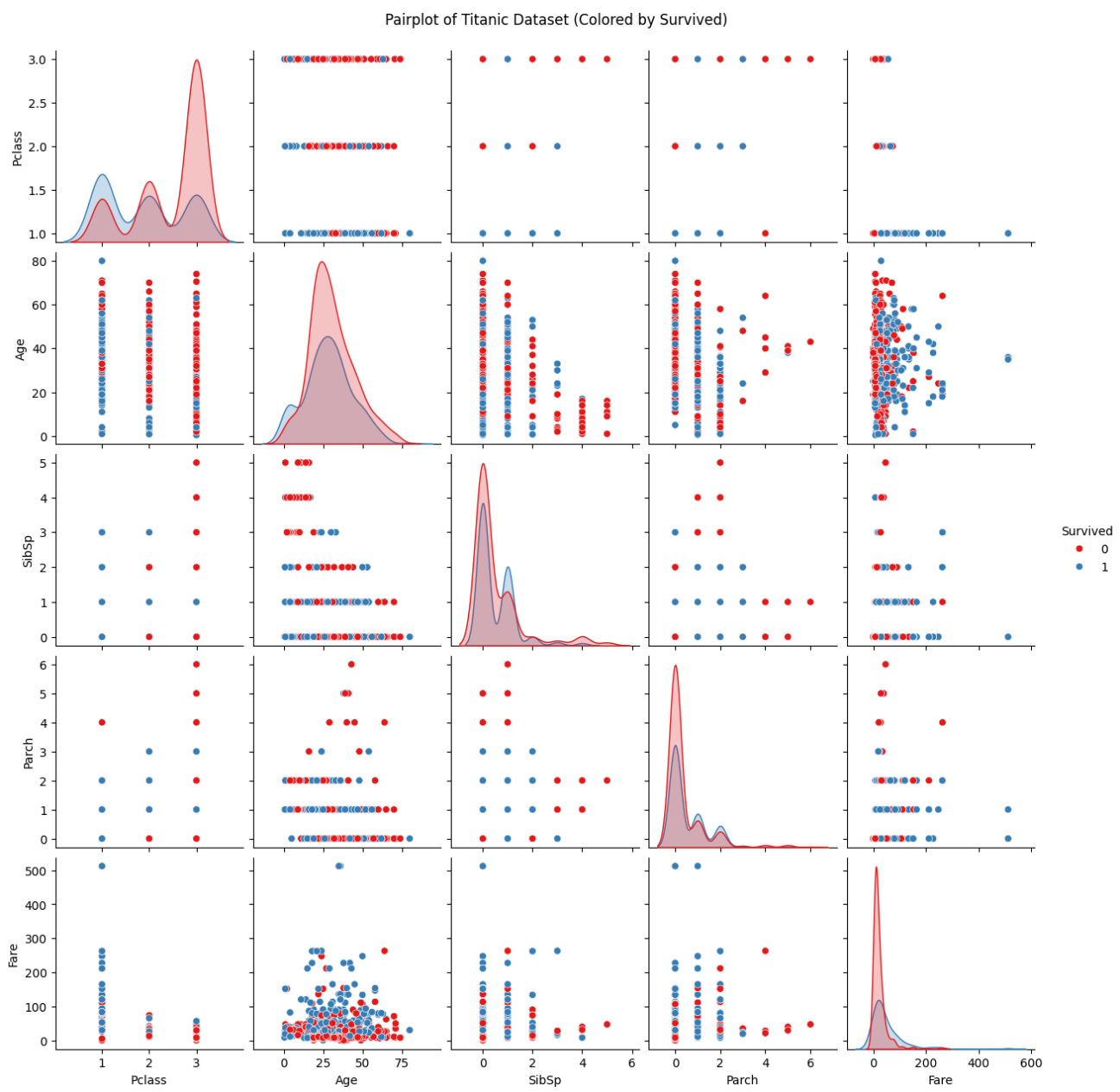
dtype: int64

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

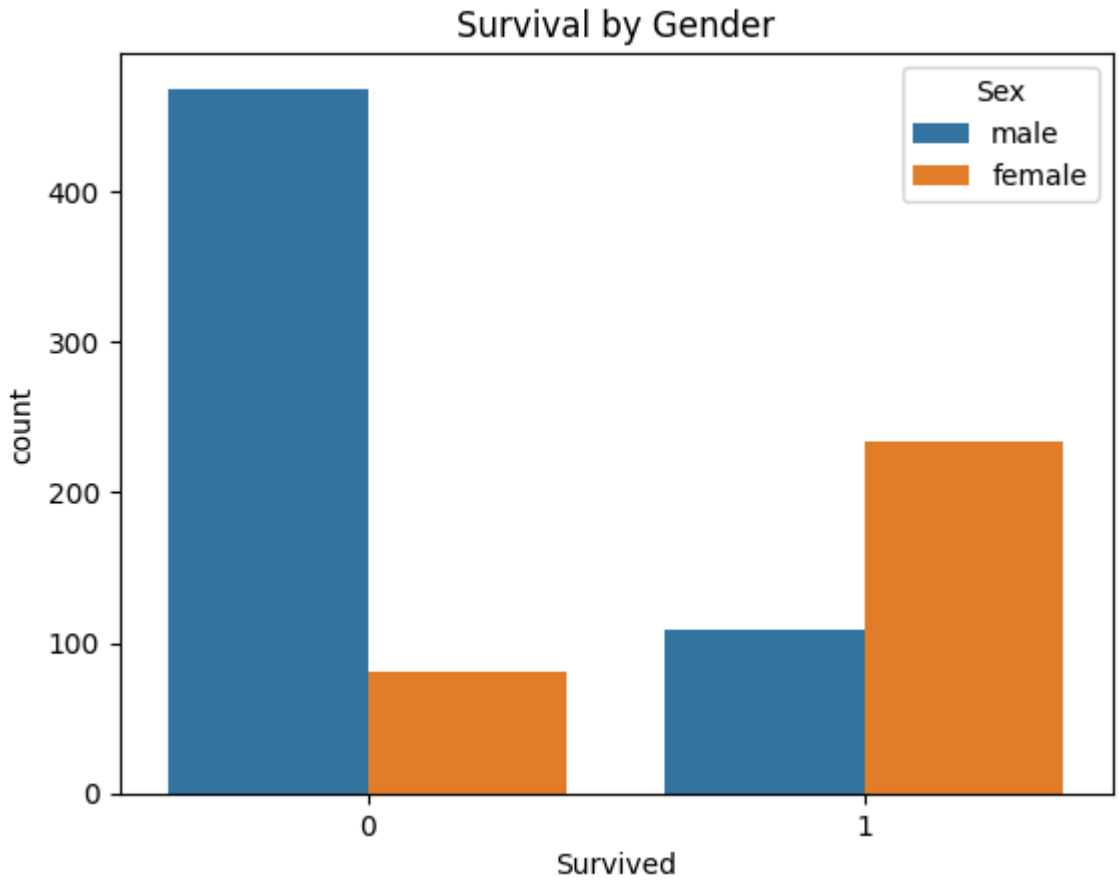
```
import seaborn as sns
import matplotlib.pyplot as plt

# Select numerical columns only
num_cols = ['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']
corr_matrix = df[num_cols].corr()

plt.figure(figsize=(10, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Heatmap')
plt.show()
```

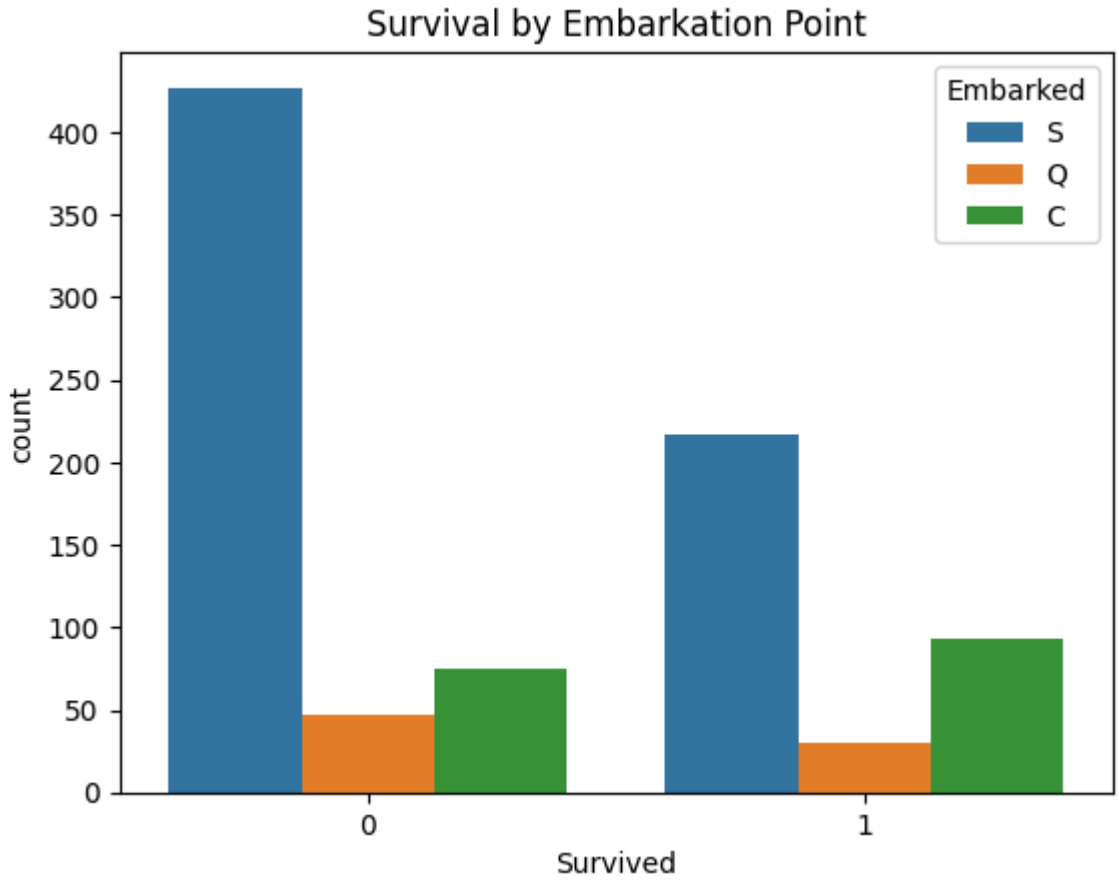



```
num_cols = ['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']

# Drop missing values to avoid errors during plotting
df_clean = df[num_cols].dropna()

# Plot pairplot
sns.pairplot(df_clean, hue='Survived', palette='Set1')
plt.suptitle('Pairplot of Titanic Dataset (Colored by Survived)', y=1.02)
plt.show()
```


Pairplot of Titanic Dataset (Colored by Survived)
Pclass
Age
SibSp
Parch
Fare
Survived
0
1

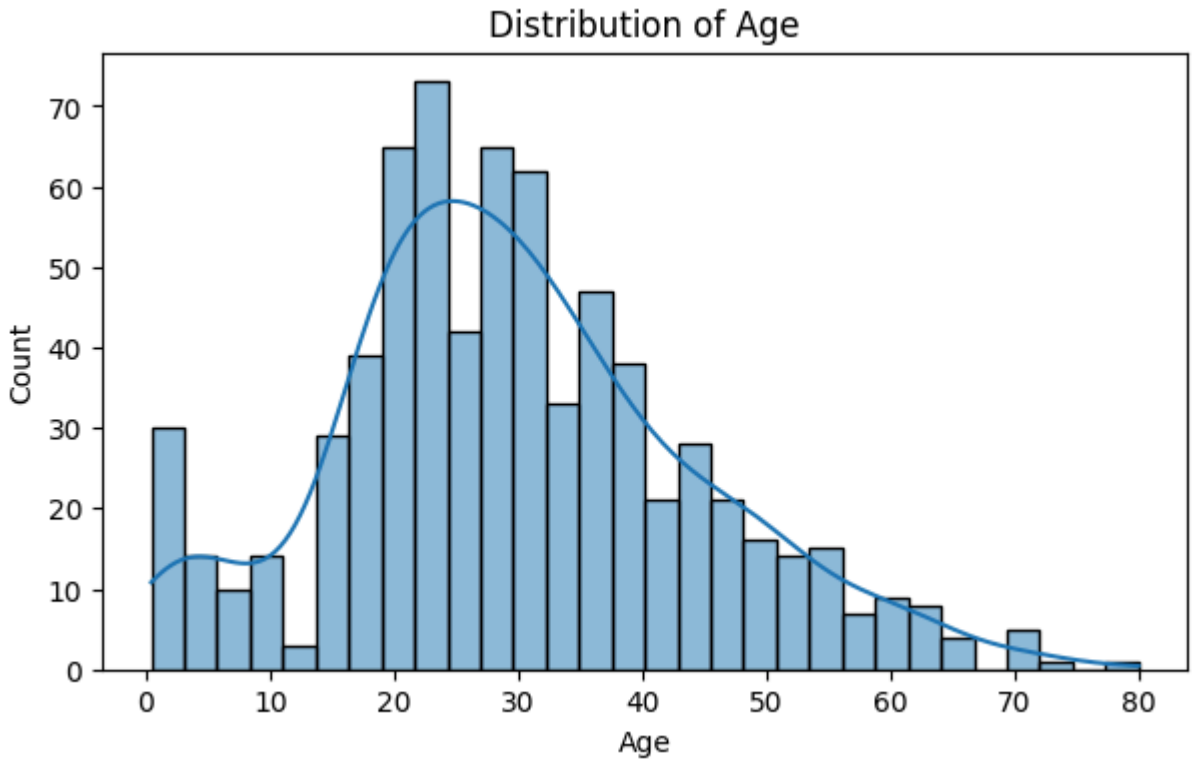
```
# Survival based on Sex
sns.countplot(x='Survived', hue='Sex', data=df)
plt.title('Survival by Gender')
plt.show()

# Survival by Embarkation Port
sns.countplot(x='Survived', hue='Embarked', data=df)
plt.title('Survival by Embarkation Point')
plt.show()
```

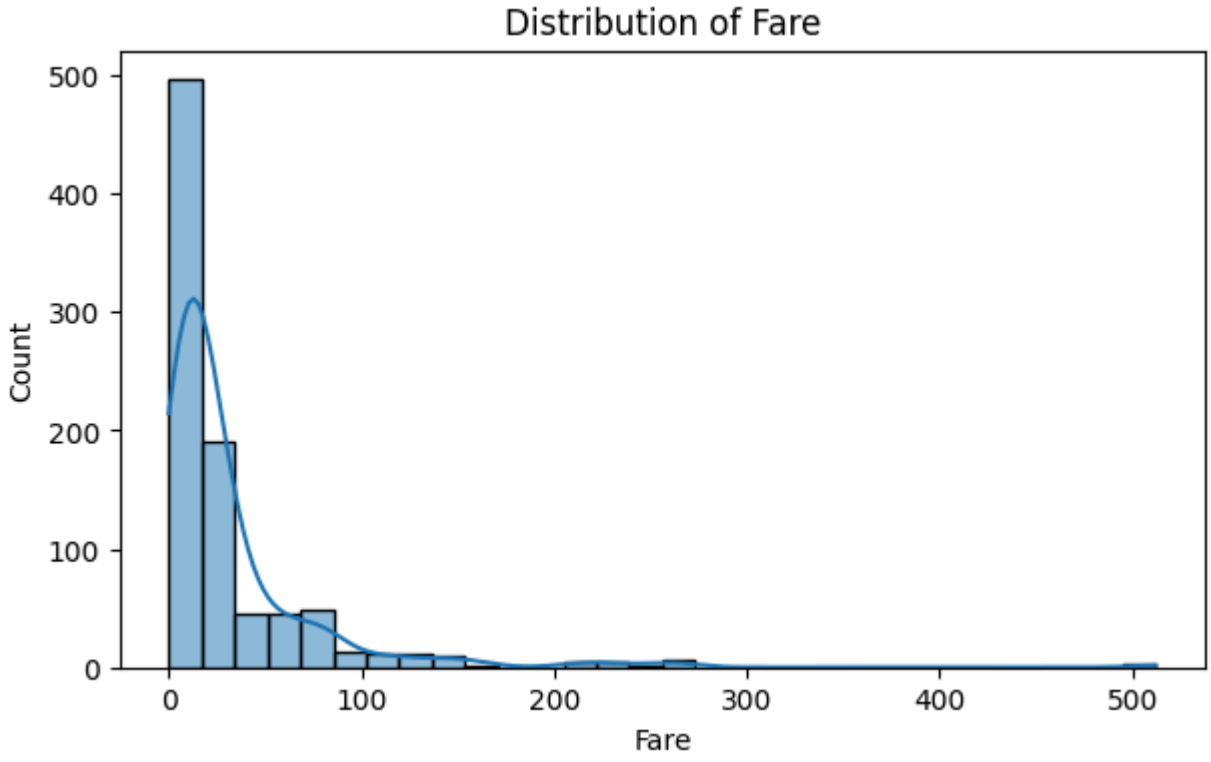


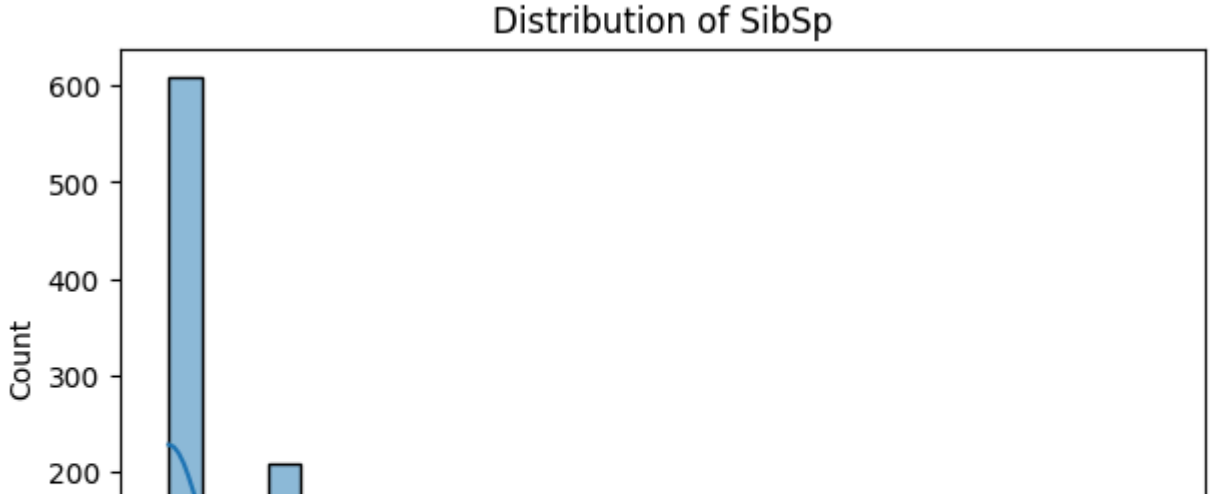


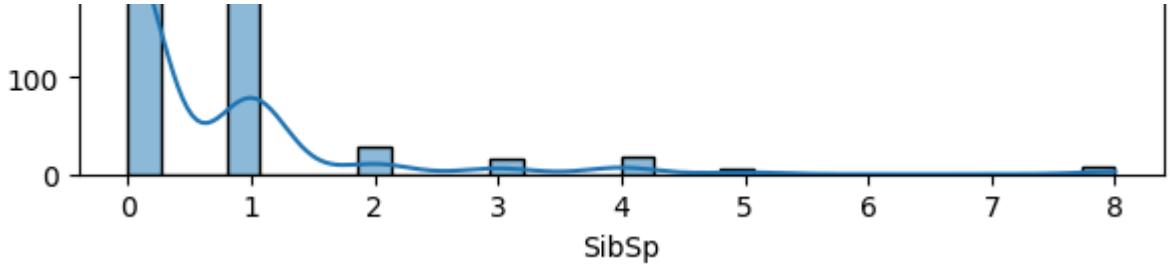

```
# Plot histogram for Age, Fare, SibSp, Parch
num_cols = ['Age', 'Fare', 'SibSp', 'Parch']

for col in num_cols:
    plt.figure(figsize=(7,4))
```

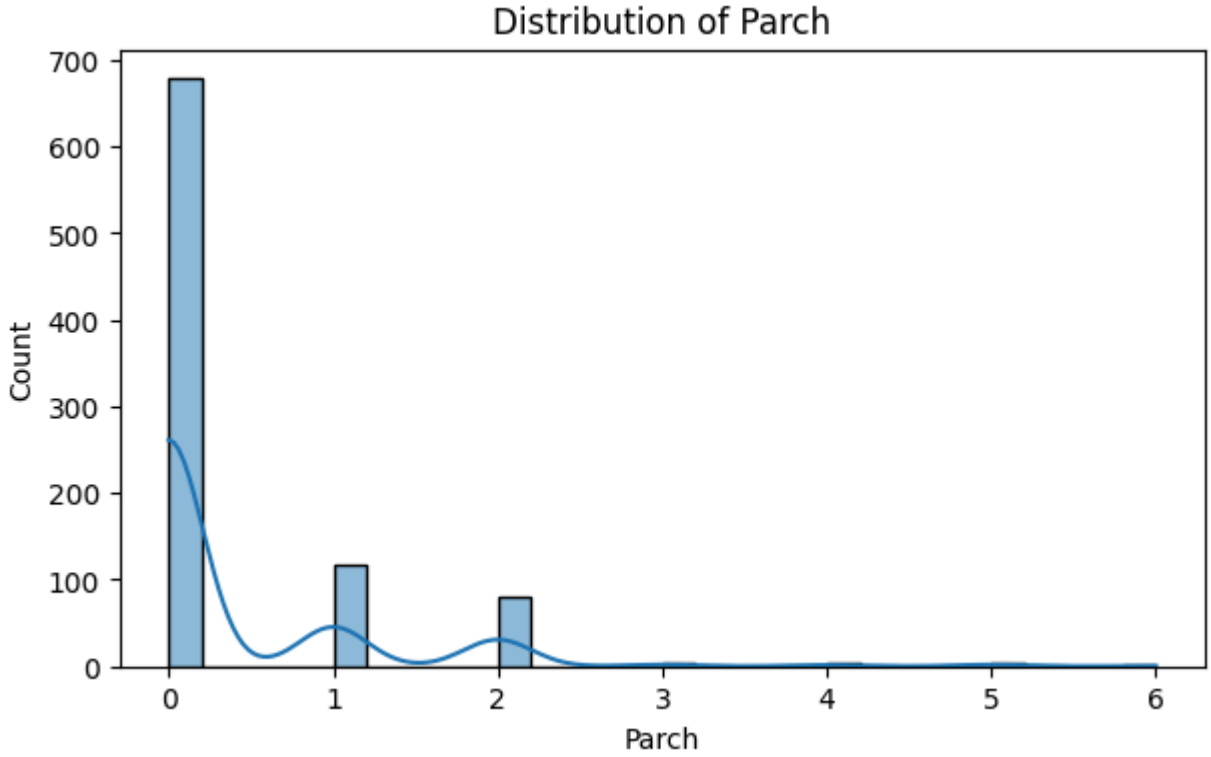
```
        sns.histplot(df[col].dropna(), kde=True, bins=30)
        plt.title(f'Distribution of {col}')
        plt.xlabel(col)
        plt.ylabel('Count')
        plt.show()
```


Distribution of Age
Count
Age
0
10
20
30
40
50
60
70
80


Distribution of Fare
Count
Fare
0
100
200
300
400
500


Distribution of SibSp
Count
600
500
400
300
200


100
0
0
1
2
3
4
5
6
7
8
SibSp


Distribution of Parch
700
600
500
400
300
200
100
0
Count
0
1
2
3
4
5
6
Parch

```
# Boxplots for Age, Fare by Survived
for col in ['Age', 'Fare']:
    plt.figure(figsize=(7,4))
    sns.boxplot(x='Survived', y=col, data=df)
    plt.title(f'{col} vs Survived')
    plt.show()
```