```
from google.colab import files
uploaded = files.upload()
```

Choose files train (2).csv

- **train (2).csv**(text/csv) - 61194 bytes, last modified: 30/06/2025 - 100% done

Saving train (2).csv to train (2) (1).csv

```
import pandas as pd

df = pd.read_csv('train (2).csv')  # or adjust filename if needed
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

Futrelle, Mrs. Jacques Heath (Lily

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

```
# Overview of structure
df.info()
# Summary statistics
df.describe()
# Missing values
df.isnull().sum()
```

```
<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
```

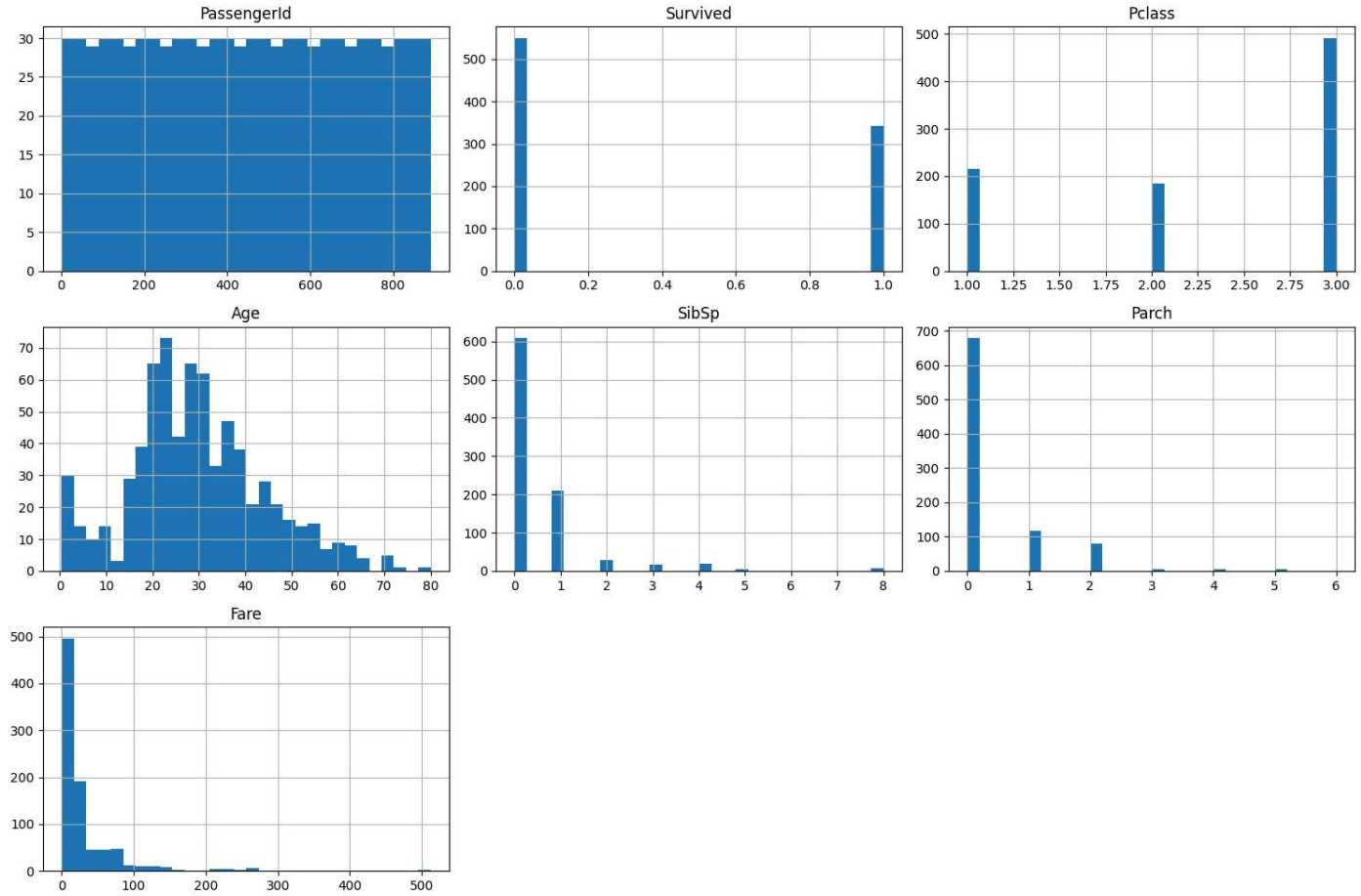
	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	2

```
# Check value counts of all categorical columns
for col in df.select_dtypes(include='object').columns:
    print(f"\n{col} value counts:\n", df[col].value_counts())
```

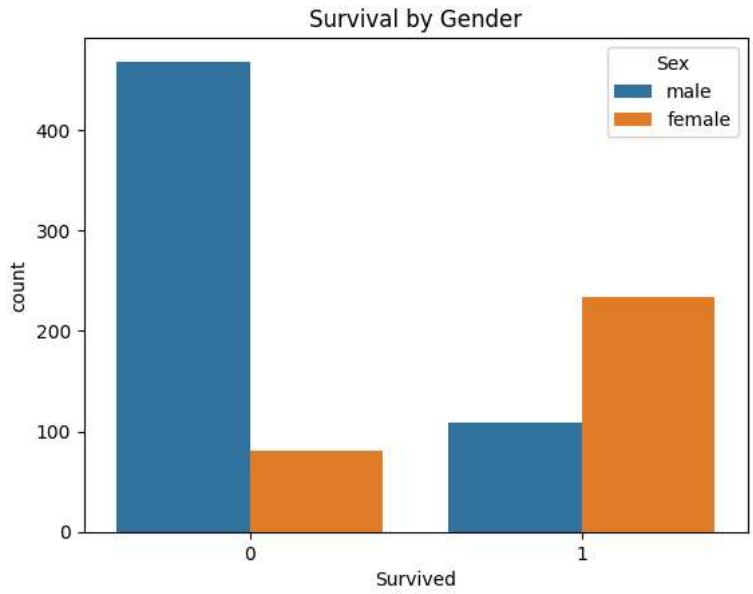
```
Name value counts:
 Name
Dooley, Mr. Patrick                                 1
Braund, Mr. Owen Harris                             1
Cumings, Mrs. John Bradley (Florence Briggs Thayer) 1
Heikkinen, Miss. Laina                              1
Futrelle, Mrs. Jacques Heath (Lily May Peel)        1
                                                   ..
Hewlett, Mrs. (Mary D Kingcome)                     1
Vestrom, Miss. Hulda Amanda Adolfina                1
Andersson, Mr. Anders Johan                         1
Saundercock, Mr. William Henry                      1
Bonnell, Miss. Elizabeth                            1
Name: count, Length: 891, dtype: int64

Sex value counts:
 Sex
male      577
female    314
Name: count, dtype: int64

Ticket value counts:
 Ticket
347082              7
1601                7
CA. 2343            7
3101295             6
CA 2144             6
                   ..
PC 17590            1
17463               1
```

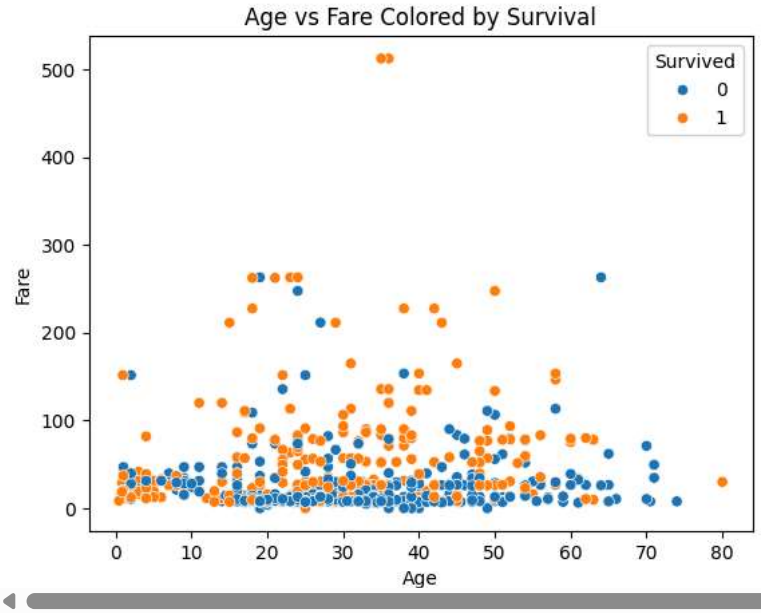
```
330877              1
373450              1
STON/O2. 3101282    1
Name: count, Length: 681, dtype: int64

Cabin value counts:
 Cabin
G6             4
C23 C25 C27    4
B96 B98        4
F2             3
D              3
              ..
E17            1
A24            1
C50            1
B42            1
C148           1
Name: count, Length: 147, dtype: int64

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

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

# Histogram of numeric features
df.hist(bins=30, figsize=(15, 10))
plt.tight_layout()
plt.show()
```

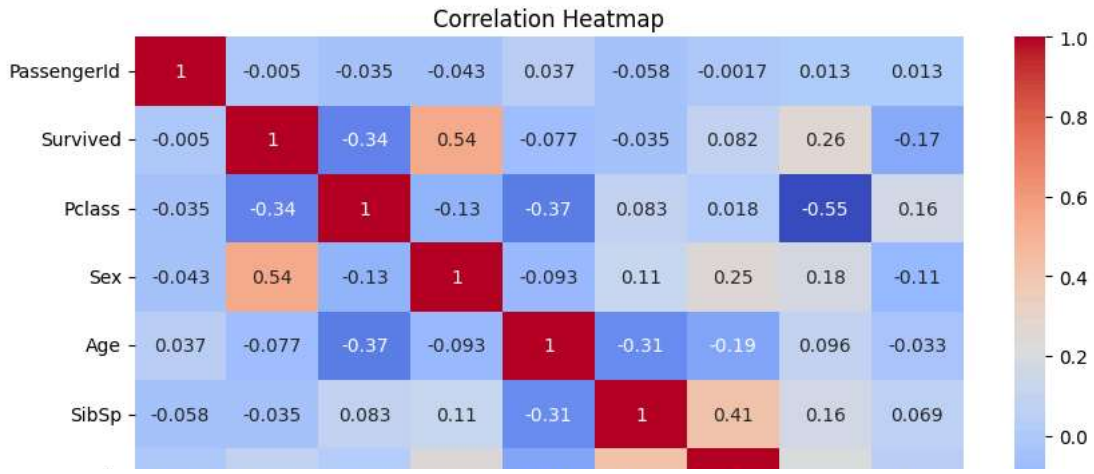



```
# Example: Survival by Gender (change 'Survived' and 'Sex' if columns are different)
sns.countplot(x='Survived', hue='Sex', data=df)
plt.title('Survival by Gender')
plt.show()
# Example: Fare vs Age Scatter Plot
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=df)
plt.title('Age vs Fare Colored by Survival')
plt.show()
```






```
# Encode categorical if needed
df_encoded = df.copy()
df_encoded['Sex'] = df_encoded['Sex'].map({'male': 0, 'female': 1})
df_encoded['Embarked'] = df_encoded['Embarked'].map({'C': 0, 'Q': 1, 'S': 2})

# Drop non-numeric columns before calculating correlation
df_numeric = df_encoded.drop(['Name', 'Ticket', 'Cabin'], axis=1)

# Heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df_numeric.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```




```
# Select useful columns
sns.pairplot(df_encoded[['Survived', 'Age', 'Fare', 'Pclass']], hue='Survived')
plt.suptitle('Pairplot of Key Variables', y=1.02)
plt.show()
```