Task 5 - Exploratory Data Analysis (Titanic Dataset)

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

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

Next steps: Generate code with df New interactive sheet

```
df.info()         # data types and missing counts
df.describe(include='all').T   # numeric + non-numeric summary
```

```
<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
```

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
PassengerId	891.0	NaN	NaN	NaN	446.0	257.353842	1.0	223.5	446.0	668.5	891.0
Survived	891.0	NaN	NaN	NaN	0.383838	0.486592	0.0	0.0	0.0	1.0	1.0
Pclass	891.0	NaN	NaN	NaN	2.308642	0.836071	1.0	2.0	3.0	3.0	3.0
Name	891	891	Dooley, Mr. Patrick	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Sex	891	2	male	577	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Age	714.0	NaN	NaN	NaN	29.699118	14.526497	0.42	20.125	28.0	38.0	80.0
SibSp	891.0	NaN	NaN	NaN	0.523008	1.102743	0.0	0.0	0.0	1.0	8.0
Parch	891.0	NaN	NaN	NaN	0.381594	0.806057	0.0	0.0	0.0	0.0	6.0
Ticket	891	681	347082	7	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Fare	891.0	NaN	NaN	NaN	32.204208	49.693429	0.0	7.9104	14.4542	31.0	512.3292
Cabin	204	147	G6	4	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Embarked	889	3	S	644	NaN	NaN	NaN	NaN	NaN	NaN	NaN

```
print("Survived counts:\n", df['Survived'].value_counts())
print("\nPclass counts:\n", df['Pclass'].value_counts())
print("\nSex counts:\n", df['Sex'].value_counts())
print("\nEmbarked counts:\n", df['Embarked'].value_counts(dropna=False))
```

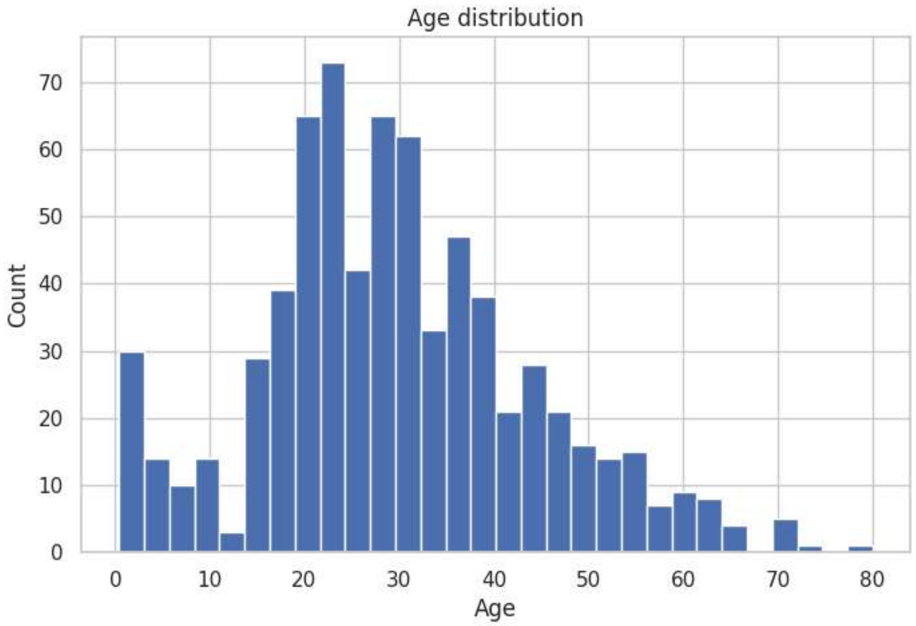
```
Survived counts:
 Survived
0    549
1    342
Name: count, dtype: int64

Pclass counts:
 Pclass
3    491
1    216
2    184
Name: count, dtype: int64

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

Embarked counts:
 Embarked
S      644
C      168
Q       77
NaN      2
Name: count, dtype: int64
```

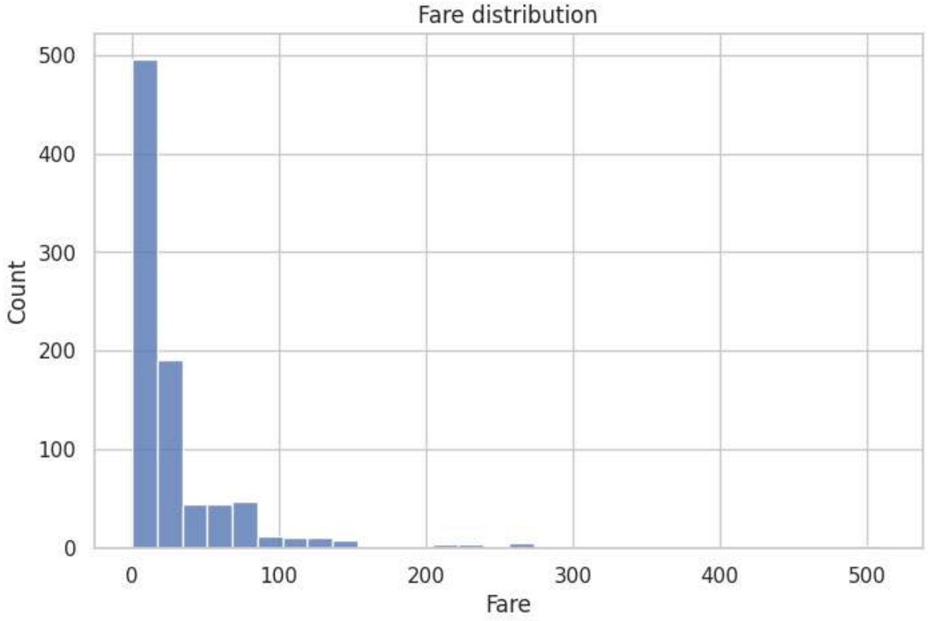
```
missing = df.isnull().sum().sort_values(ascending=False)
missing = pd.DataFrame({'missing_count': missing, 'missing_pct': missing/len(df)*100})
missing
```

	missing_count	missing_pct
Cabin	687	77.104377
Age	177	19.865320
Embarked	2	0.224467
PassengerId	0	0.000000
Name	0	0.000000
Pclass	0	0.000000
Survived	0	0.000000
Sex	0	0.000000
Parch	0	0.000000
SibSp	0	0.000000
Fare	0	0.000000
Ticket	0	0.000000

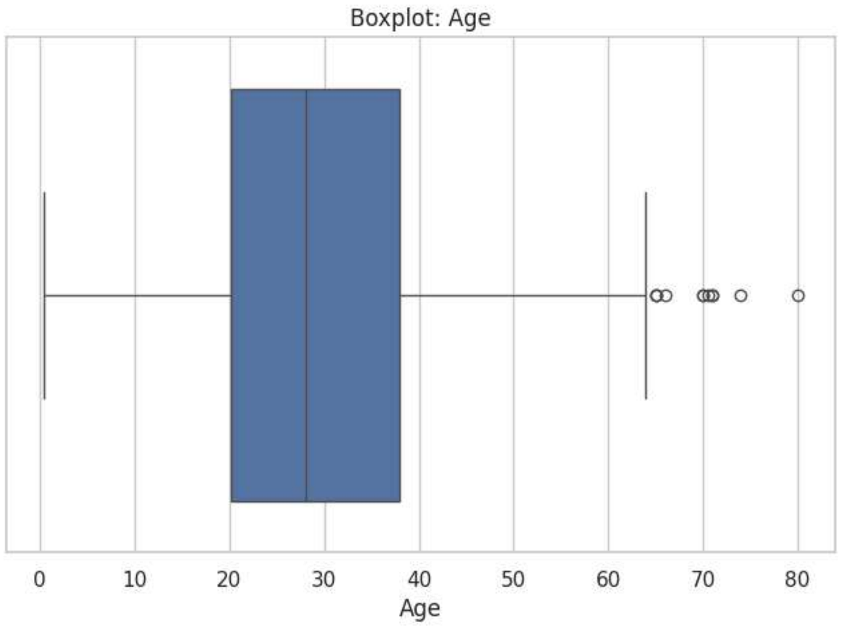
Next steps: Generate code with missing New interactive sheet

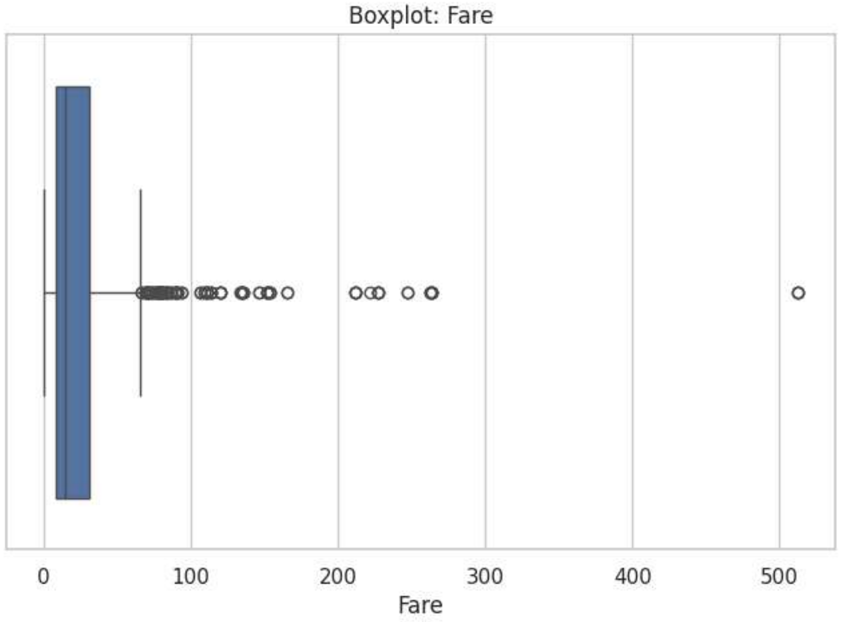
```
# Age histogram
df['Age'].hist(bins=30)
plt.title('Age distribution')
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()

# Fare distribution (log scale if skewed)
sns.histplot(df['Fare'].dropna(), bins=30)
plt.title('Fare distribution')
plt.show()
```



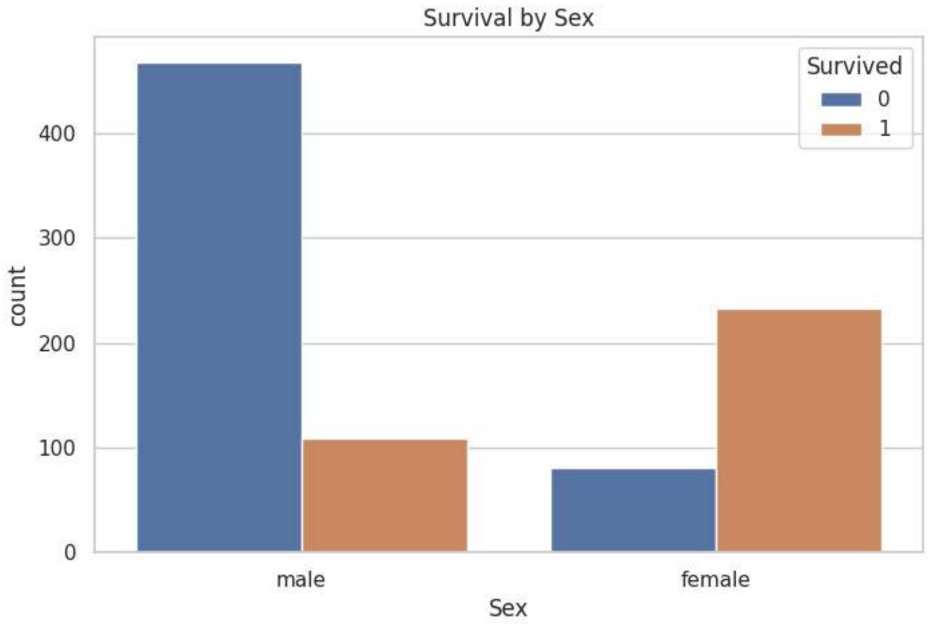


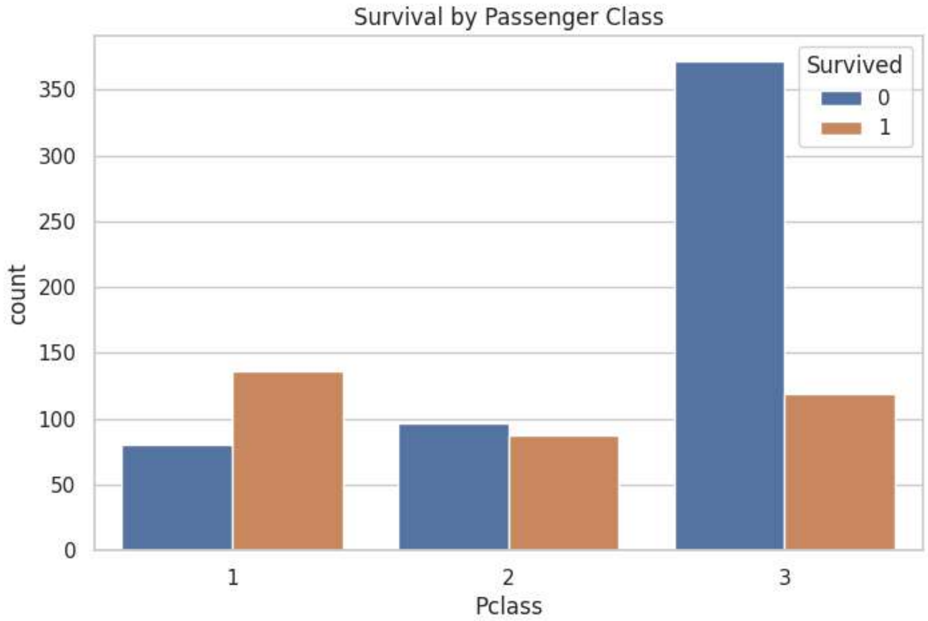
```
sns.boxplot(x=df['Age'])
plt.title('Boxplot: Age')
plt.show()

sns.boxplot(x=df['Fare'])
plt.title('Boxplot: Fare')
plt.show()
```



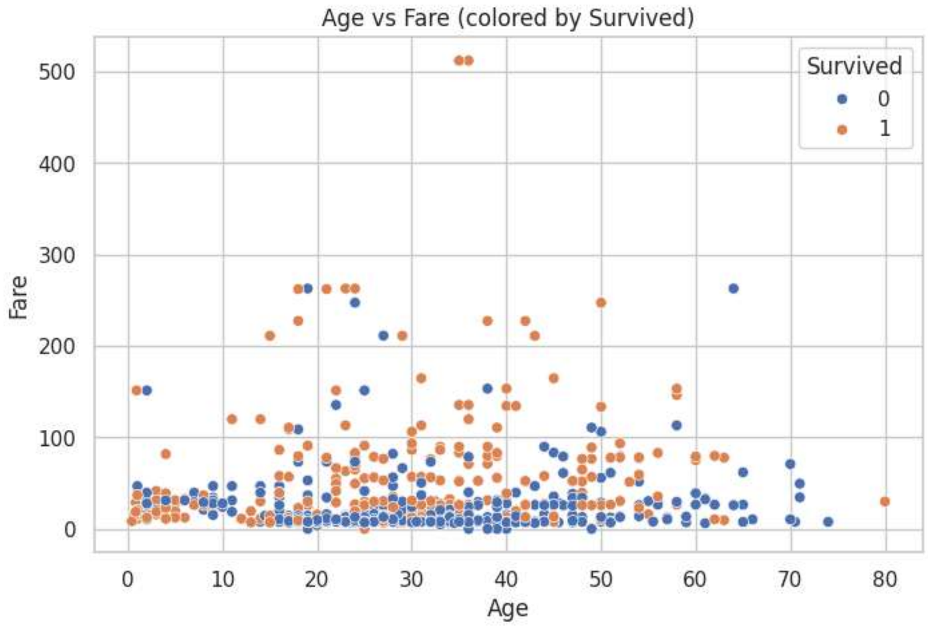


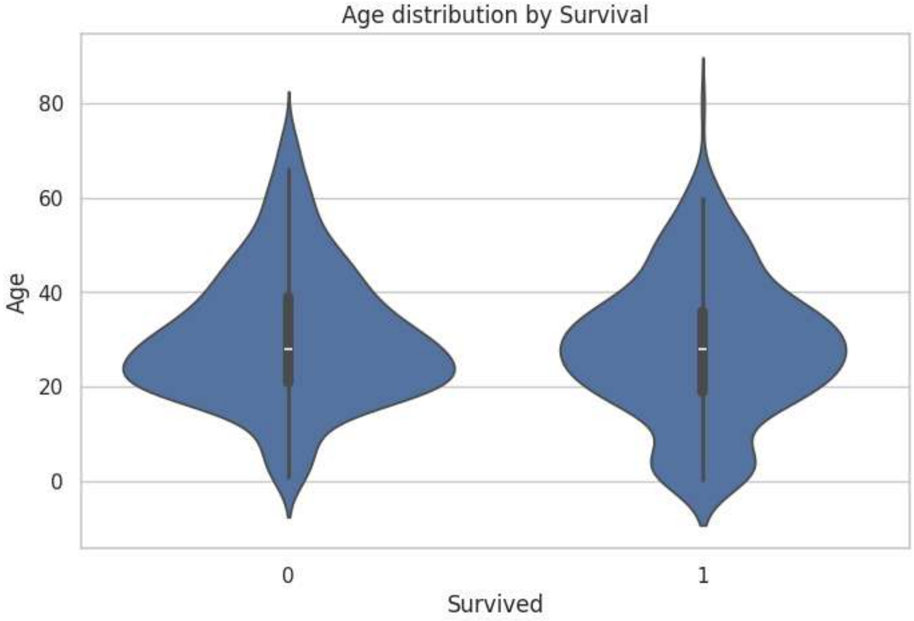
```
# Survived by Sex
sns.countplot(x='Sex', hue='Survived', data=df)
plt.title('Survival by Sex')
plt.show()

# Survived by Pclass
sns.countplot(x='Pclass', hue='Survived', data=df)
plt.title('Survival by Passenger Class')
plt.show()
```



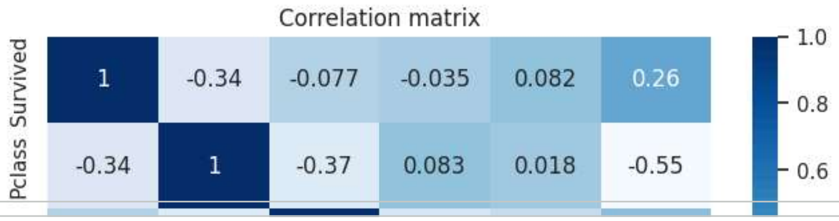


```
# Age vs Fare colored by Survived
sns.scatterplot(x='Age', y='Fare', hue='Survived', data=df)
plt.title('Age vs Fare (colored by Survived)')
plt.show()

# Age distribution by Survived (violin)
sns.violinplot(x='Survived', y='Age', data=df)
plt.title('Age distribution by Survival')
plt.show()
```

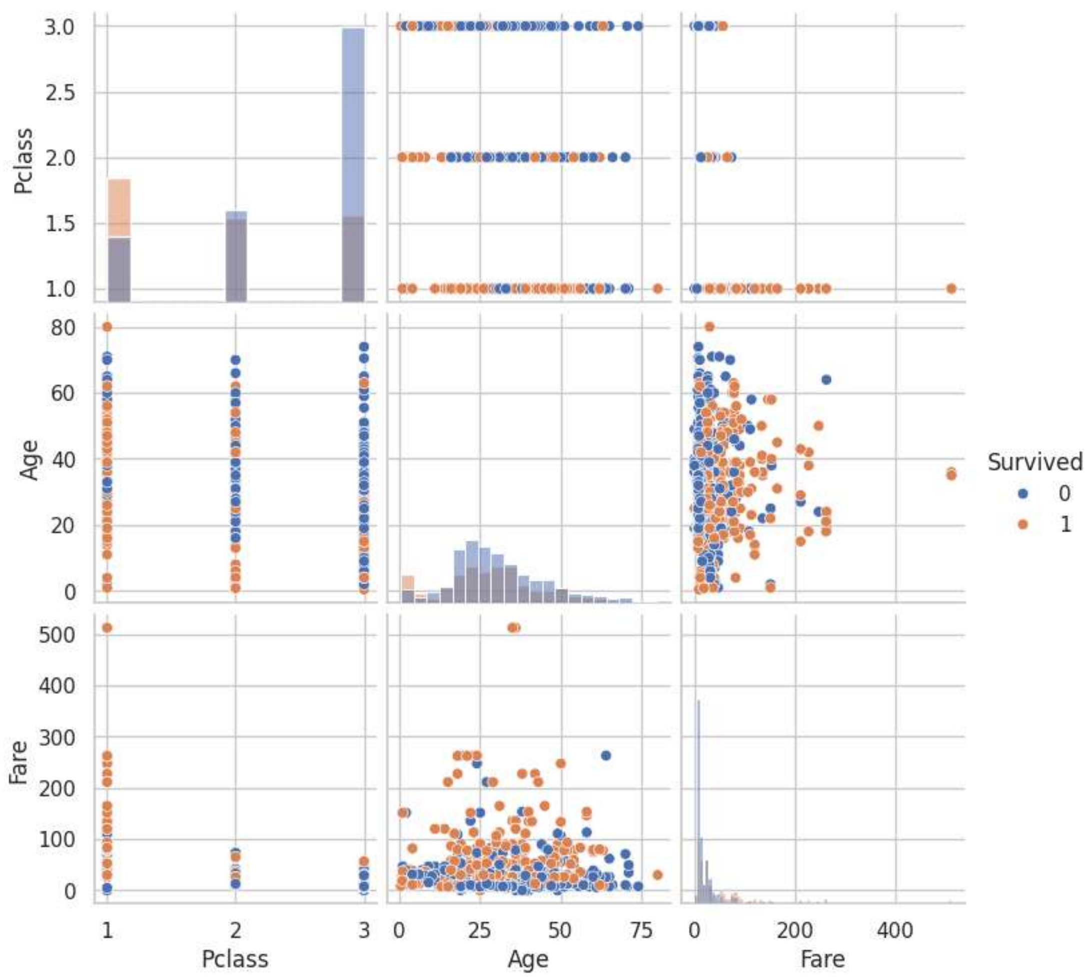




```
# First convert non-numeric if needed for correlation
corr_df = df[['Survived','Pclass','Age','SibSp','Parch','Fare']].corr()
corr_df

sns.heatmap(corr_df, annot=True, cmap='Blues')
plt.title('Correlation matrix')
plt.show()
```



```
# select small subset of cols to avoid clutter
sns.pairplot(df[['Survived','Pclass','Age','Fare']].dropna(), hue='Survived', diag_kind='hist')
plt.show()
```



```
# Survival rate by sex
surv_by_sex = df.groupby('Sex')['Survived'].mean().reset_index()
surv_by_sex

# Survival rate by Pclass
surv_by_class = df.groupby('Pclass')['Survived'].mean().reset_index()
surv_by_class

# Survival by Embarked
surv_by_embarked = df.groupby('Embarked')['Survived'].mean().reset_index()
surv_by_embarked
```

	Embarked	Survived
0	C	0.553571
1	Q	0.389610
2	S	0.336957

Next steps: Generate code with surv_by_embarked New interactive sheet