```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
url = "https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv"
df = pd.read_csv(url)
```

```
df.shape        # Rows & columns
df.head()       # First 5 rows
df.info()       # Column names & data types
df.describe()   # Summary statistics
```

```
<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
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
df.isnull().sum()
```

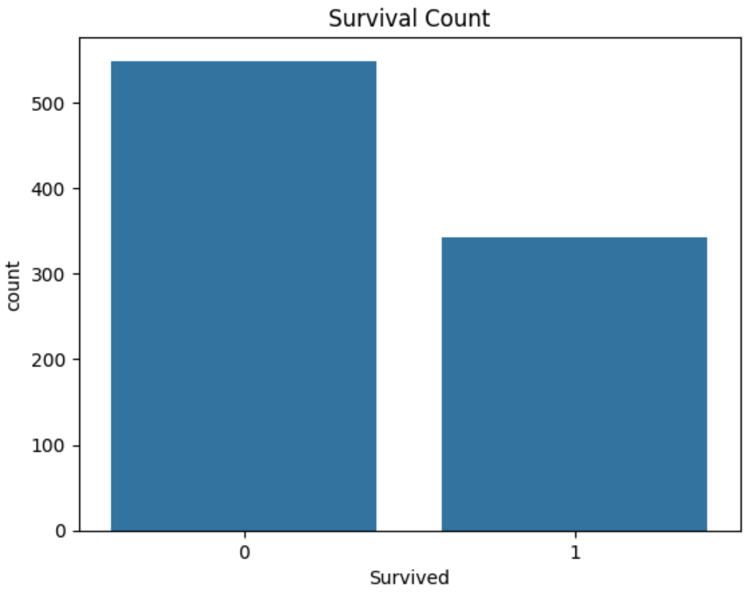
	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	177
SibSp	0
Parch	0
Ticket	0
Fare	0
Cabin	687
Embarked	2

dtype: int64

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

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

sns.countplot(x='Survived', data=df)
plt.title("Survival Count")
plt.show()
```



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

# First 5 rows
print("\nFirst 5 rows:")
print(df.head())

# Column names, data types & null counts
print("\nInfo:")
print(df.info())

# Statistical summary (only numeric columns)
print("\nSummary statistics:")
print(df.describe())
```

```
4           5         0       3

                                              Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   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




 1   Survived     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
None

Summary 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
```

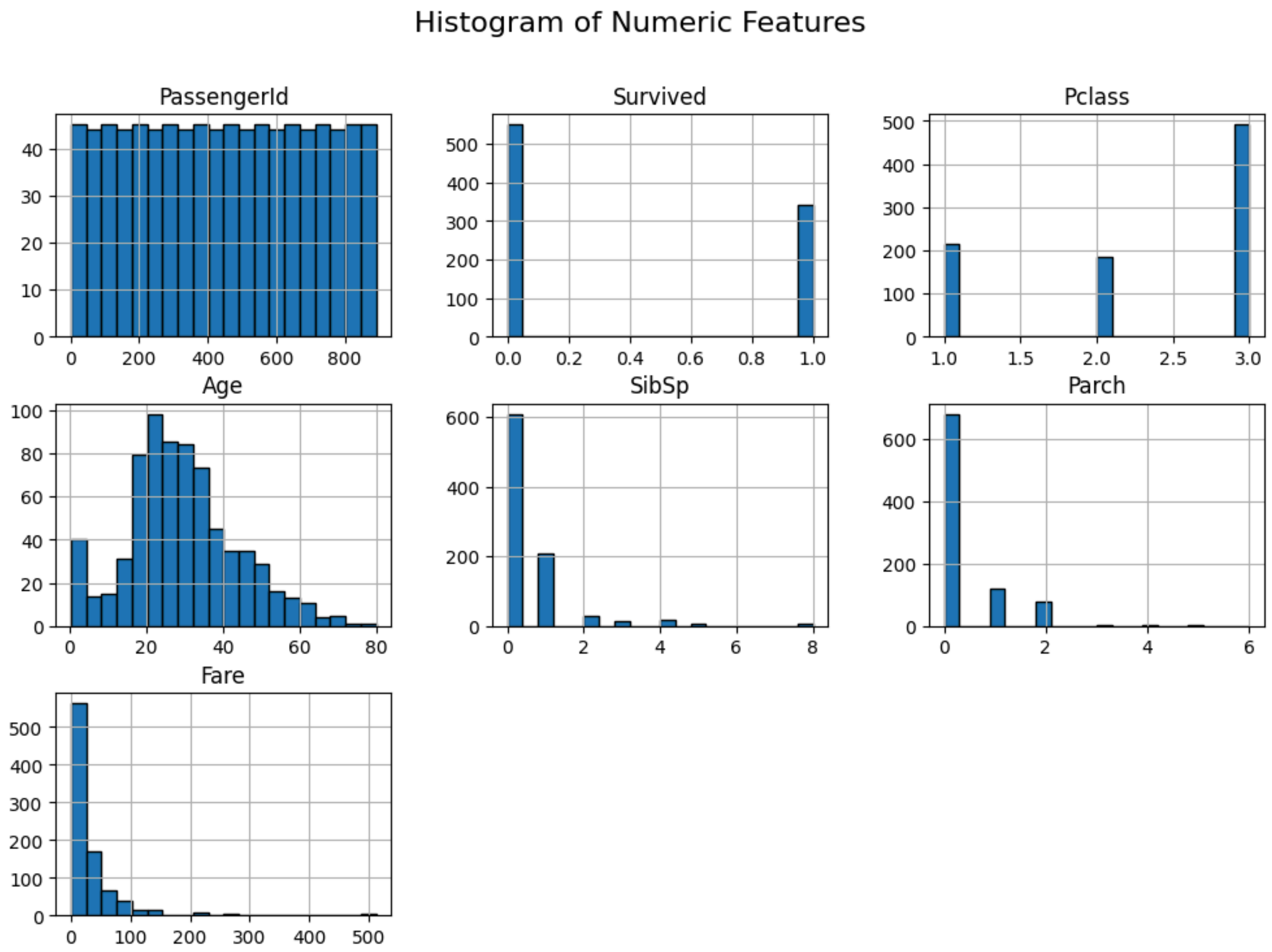
```
print("\nMissing values in each column:")
print(df.isnull().sum())
```

```
Missing values in each column:
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
```

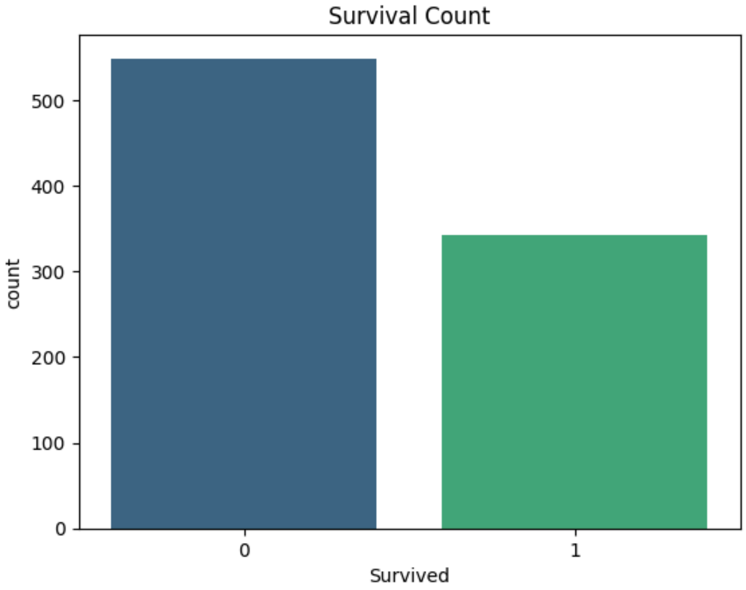
```
for col in df.select_dtypes(include=['object']).columns:
    print(f"\nValue counts for {col}:\n", df[col].value_counts())
```

```
Value counts for Name:
 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

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

Value counts for Ticket:
 Ticket
 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

Value counts for Cabin:
 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

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

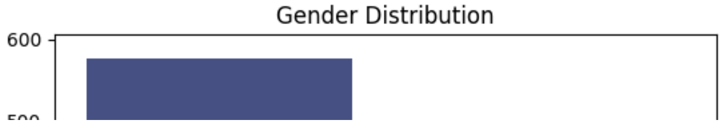
```
import seaborn as sns
import matplotlib.pyplot as plt

# =======================
# 📌 Univariate Analysis
# =======================

# Histogram for numeric features
df.hist(figsize=(12, 8), bins=20, edgecolor='black')
plt.suptitle("Histogram of Numeric Features", fontsize=16)
plt.show()

# Survival count
sns.countplot(x='Survived', data=df, palette='viridis')
plt.title("Survival Count")
plt.show()

# Gender distribution
sns.countplot(x='Sex', data=df, palette='mako')
plt.title("Gender Distribution")
plt.show()

# =======================
# 📌 Bivariate Analysis
# =======================

# Survival by Gender
sns.countplot(x='Sex', hue='Survived', data=df, palette='coolwarm')
plt.title("Survival Count by Gender")
plt.show()

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

# =======================
# 📌 Outlier Detection
# =======================

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

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

# =======================
# 📌 Correlation Analysis
# =======================

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

# =======================
# 📌 Pairplot
# =======================

sns.pairplot(df[['Survived', 'Age', 'Fare', 'Pclass']], hue='Survived', palette='husl')
plt.show()
```

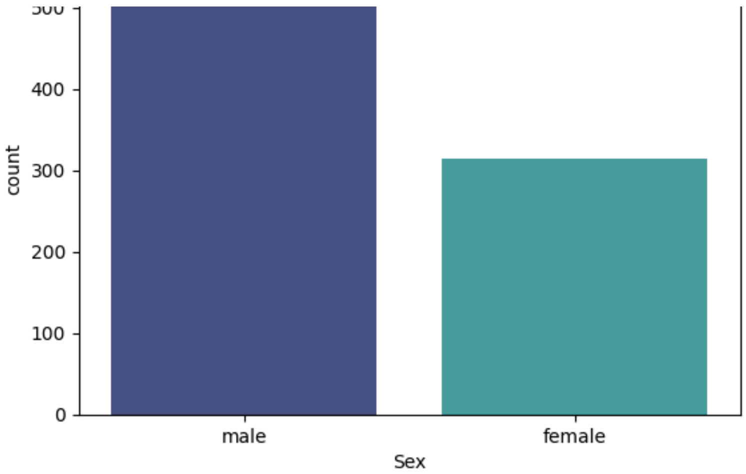


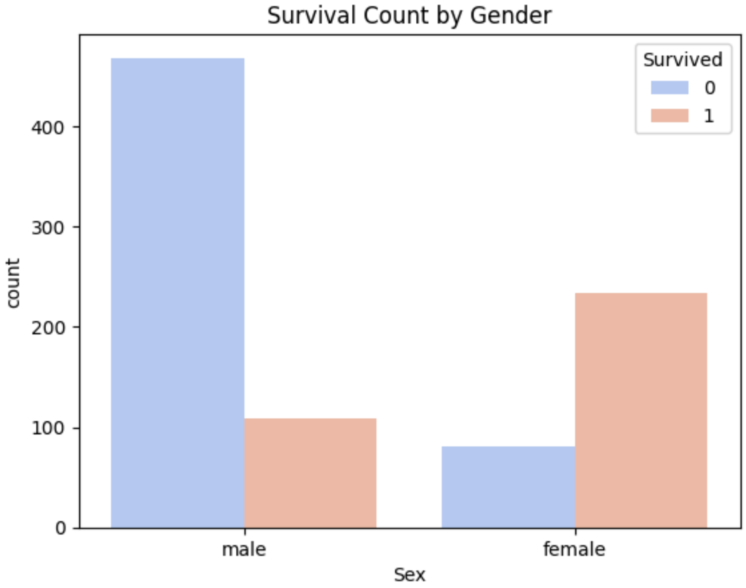
```
/tmp/ipython-input-2791850807.py:14: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `lege

  sns.countplot(x='Survived', data=df, palette='viridis')
```

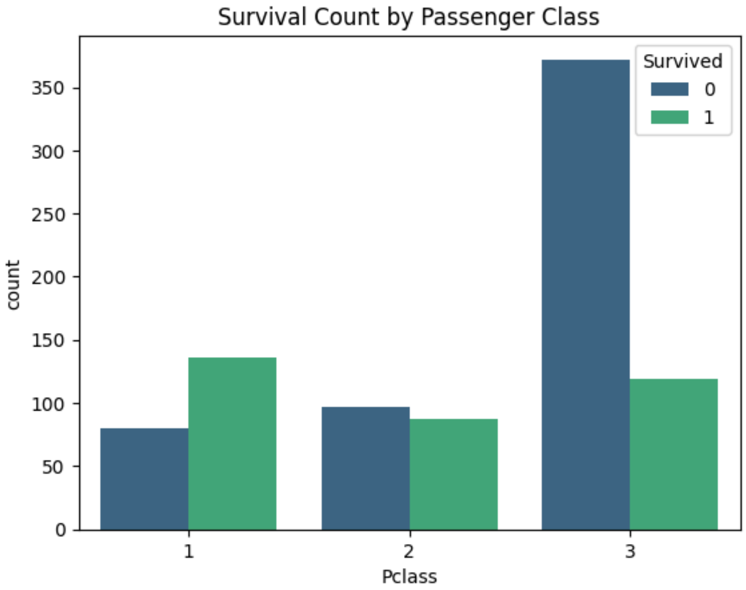


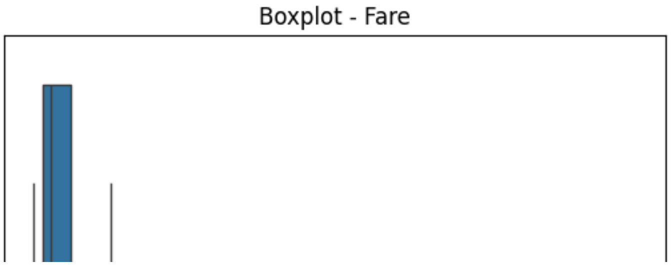
```
/tmp/ipython-input-2791850807.py:19: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `lege

  sns.countplot(x='Sex', data=df, palette='mako')
```

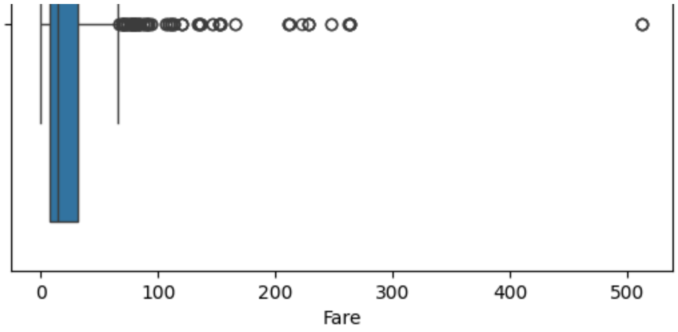


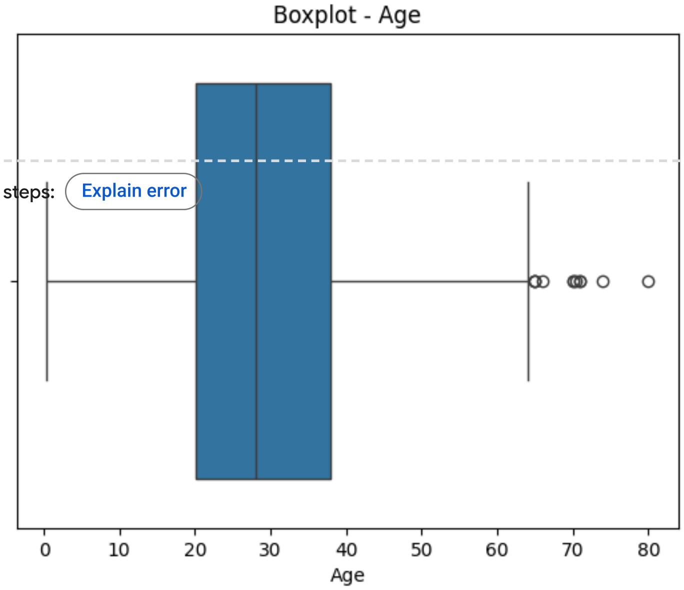

500
400
300
200
100
0
count
male
female
Sex


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


Survival Count by Passenger Class
Survived
0
1
350
300
250
200
150
100
50
0
count
1
2
3
Pclass


Boxplot - Fare





Next steps: Explain error

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/tmp/ipython-input-2791850807.py in <cell line: 0>()
     52 
     53 plt.figure(figsize=(10, 6))
---> 54 sns.heatmap(df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
     55 plt.title("Correlation Heatmap")
     56 plt.show()

                                3 frames
/usr/local/lib/python3.11/dist-packages/pandas/core/internals/managers.py in _interleave(self, dtype, na_value)
   1751             else:
   1752                 arr = blk.get_values(dtype)
-> 1753             result[rl.indexer] = arr
   1754             itemmask[rl.indexer] = 1
   1755 

ValueError: could not convert string to float: 'Braund, Mr. Owen Harris'

<Figure size 1000x600 with 0 Axes>
```

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

# =======================
# 📌 Univariate Analysis
# =======================

# Histogram for numeric features only
numeric_df = df.select_dtypes(include=['number'])
numeric_df.hist(figsize=(12, 8), bins=20, edgecolor='black')
plt.suptitle("Histogram of Numeric Features", fontsize=16)
plt.show()

# Survival count
sns.countplot(x='Survived', data=df, palette='viridis')
plt.title("Survival Count")
plt.show()

# Gender distribution
sns.countplot(x='Sex', data=df, palette='mako')
plt.title("Gender Distribution")
plt.show()

# =======================
# 📌 Bivariate Analysis
# =======================

# Survival by Gender
sns.countplot(x='Sex', hue='Survived', data=df, palette='coolwarm')
plt.title("Survival Count by Gender")
plt.show()

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

# =======================
# 📌 Outlier Detection
# =======================

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

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

# =======================
# 📌 Correlation Analysis (Only Numeric Columns)
# =======================
numeric_df = df.select_dtypes(include=['number'])
plt.figure(figsize=(10, 6))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()

# =======================
# 📌 Pairplot (Only Selected Numeric Columns)
# =======================
sns.pairplot(df[['Survived', 'Age', 'Fare', 'Pclass']], hue='Survived', palette='husl')
plt.show()
```