Importing Libraries And Datasets

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

```
df = pd.read_csv("/content/cleaned_titanic.csv")
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	1	-0.542633	1	0	A/5 21171	-0.797316	B96 B98	2
1	3	1	3	Heikkinen, Miss. Laina	0	-0.228652	0	0	STON/O2. 3101282	-0.738993	B96 B98	2
2	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	0.477804	1	0	113803	3.164321	C123	2
											B96	

Generating Summary Statistics

```
df.describe()
```

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
count	750.000000	750.000000	750.000000	750.000000	7.500000e+02	750.000000	750.000000	7.500000e+02	750.000000
mean	445.746667	0.325333	2.508000	0.688000	3.647453e-16	0.438667	0.337333	2.652693e-16	1.614667
std	260.420103	0.468812	0.711766	0.463619	1.000667e+00	0.910317	0.790172	1.000667e+00	0.723106
min	1.000000	0.000000	1.000000	0.000000	-2.236560e+00	0.000000	0.000000	-1.423747e+00	0.000000
25%	214.250000	0.000000	2.000000	0.000000	-5.426332e-01	0.000000	0.000000	-7.415157e-01	2.000000
50%	449.500000	0.000000	3.000000	1.000000	6.171059e-02	0.000000	0.000000	-3.625941e-01	2.000000
75%	670.750000	1.000000	3.000000	1.000000	3.993092e-01	1.000000	0.000000	8.089940e-01	2.000000
max	891.000000	1.000000	3.000000	1.000000	4.010089e+00	5.000000	6.000000	3.164321e+00	2.000000

```
df.info()
```

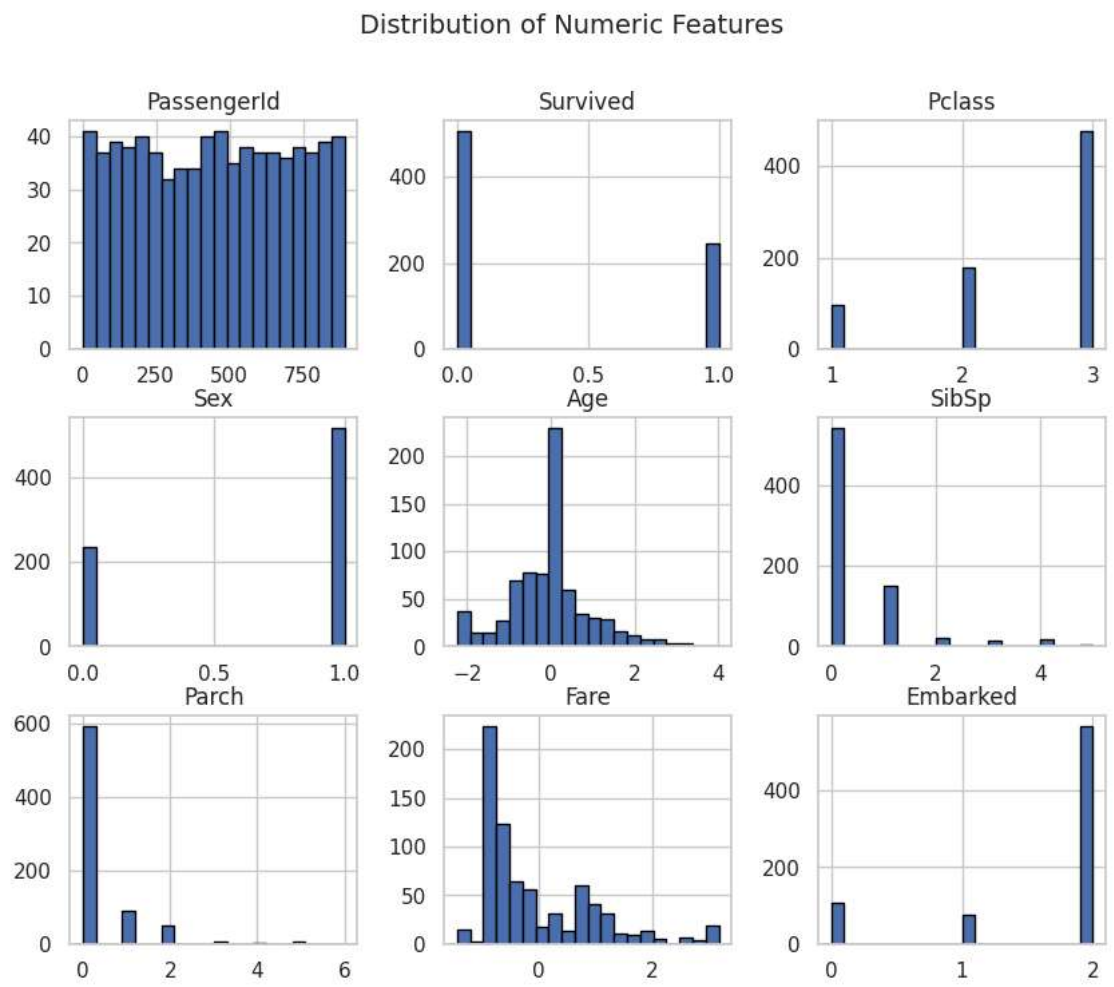
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 750 entries, 0 to 749
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  750 non-null    int64
 1   Survived     750 non-null    int64
 2   Pclass       750 non-null    int64
 3   Name         750 non-null    object
 4   Sex          750 non-null    int64
 5   Age          750 non-null    float64
 6   SibSp        750 non-null    int64
 7   Parch        750 non-null    int64
 8   Ticket       750 non-null    object
 9   Fare         750 non-null    float64
 10  Cabin        750 non-null    object
 11  Embarked     750 non-null    int64
dtypes: float64(2), int64(7), object(3)
memory usage: 70.4+ KB
```

```
df.head()
```

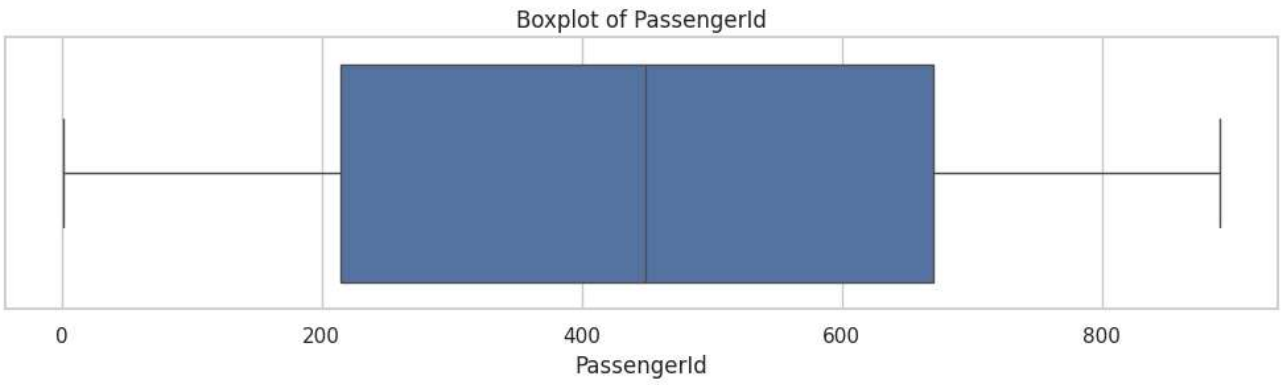
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	1	-0.542633	1	0	A/5 21171	-0.797316	B96 B98	2
1	3	1	3	Heikkinen, Miss. Laina	0	-0.228652	0	0	STON/O2. 3101282	-0.738993	B96 B98	2
2	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	0.477804	1	0	113803	3.164321	C123	2

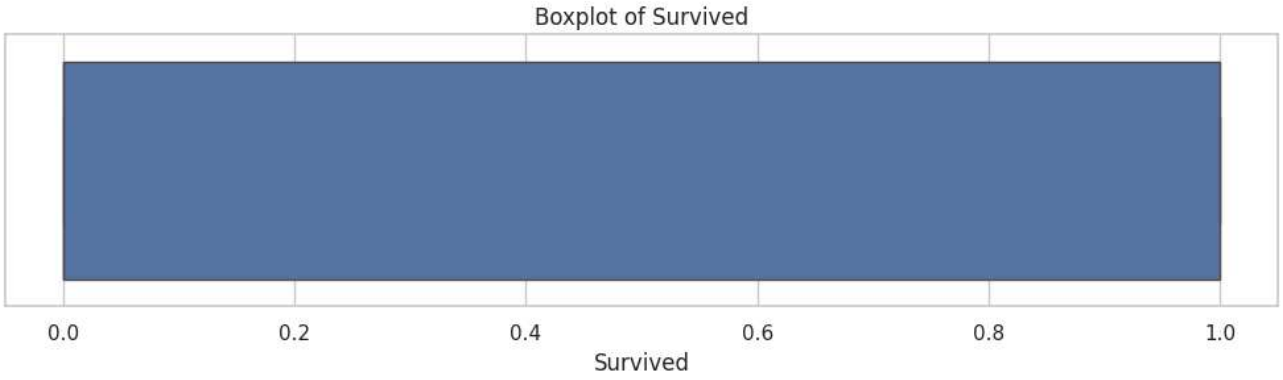
Histograms And Boxplots For Numerical Features

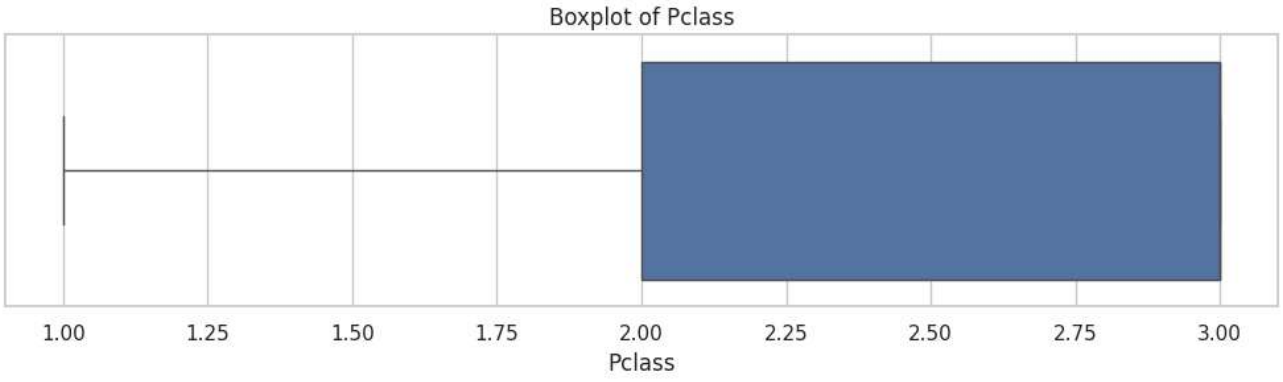
```
num_cols = df.select_dtypes(include = ['int64', 'float64']).columns

df[num_cols].hist(figsize = (10,8), bins = 20, edgecolor = 'black')
plt.suptitle('Histograms Of Numeric Features', fontsize = 14)
plt.show()
```

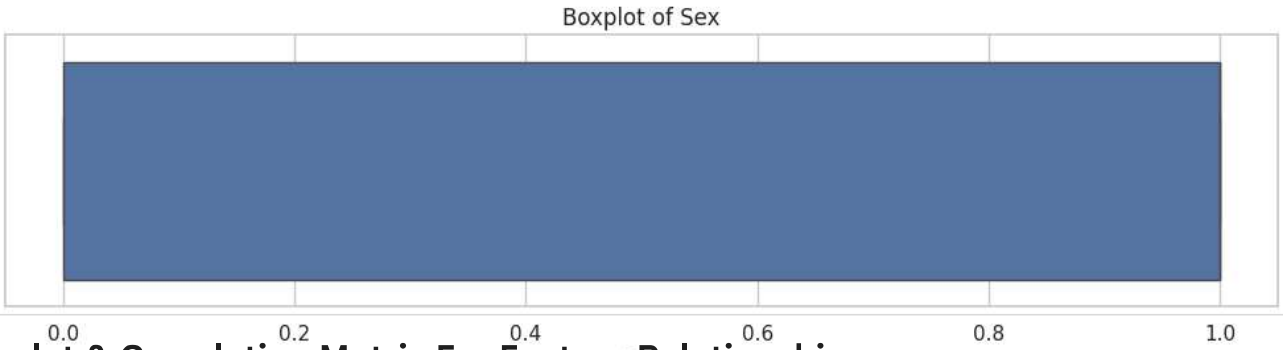



```
plt.figure(figsize = (10, len(num_cols)*3))
for i, col in enumerate(num_cols, 1):
  plt.subplot(len(num_cols), 1,i)
  sns.boxplot(x=df[col])
  plt.title(f'Boxplot of {col}')
plt.tight_layout()
plt.show()
```



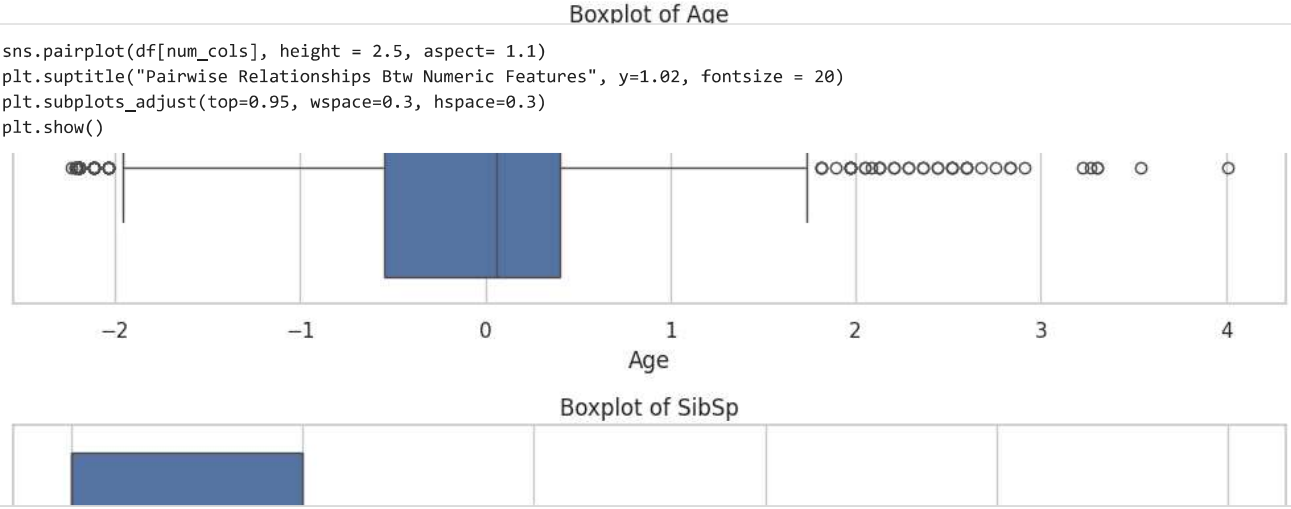


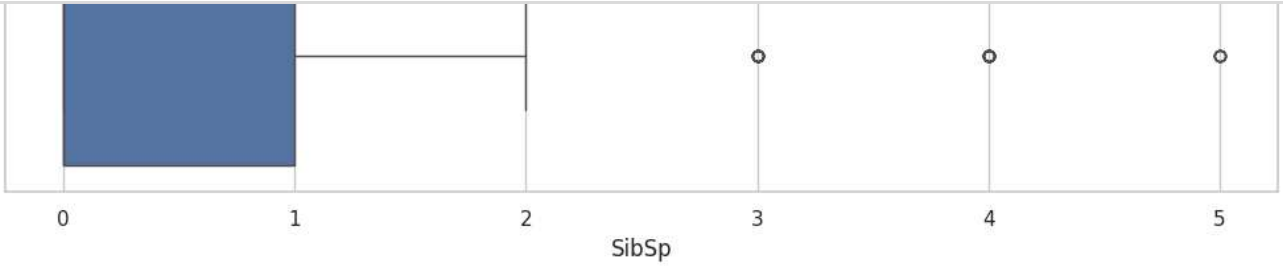


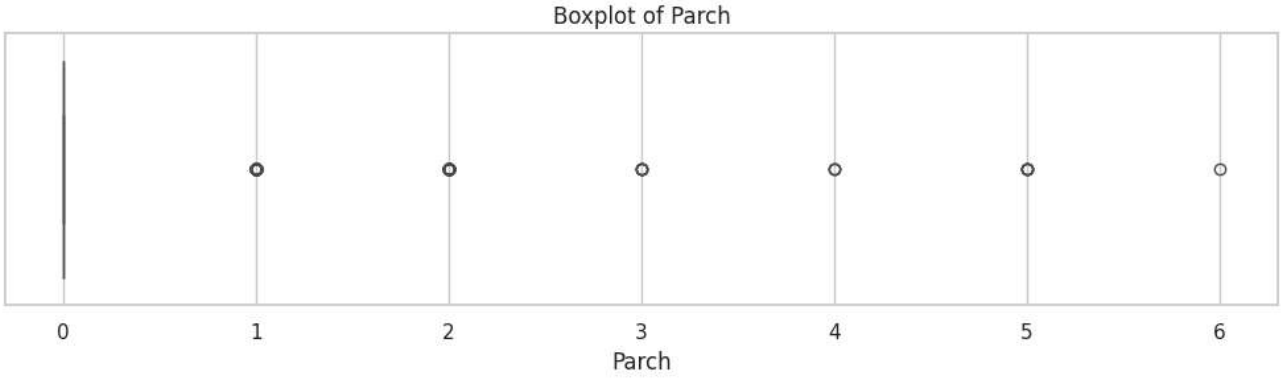


Pairplot & Correlation Matrix For Feature Relationships

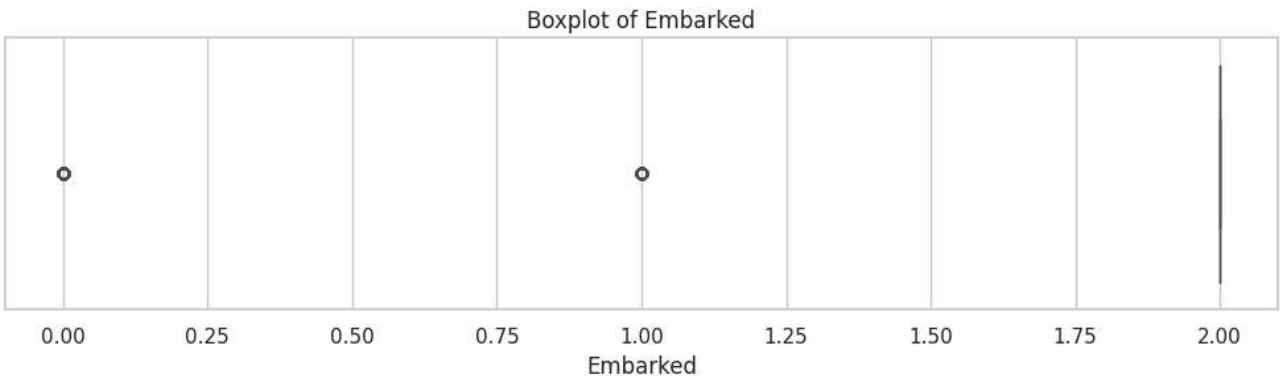
```
sns.pairplot(df[num_cols], height = 2.5, aspect= 1.1)
plt.suptitle("Pairwise Relationships Btw Numeric Features", y=1.02, fontsize = 20)
plt.subplots_adjust(top=0.95, wspace=0.3, hspace=0.3)
plt.show()
```

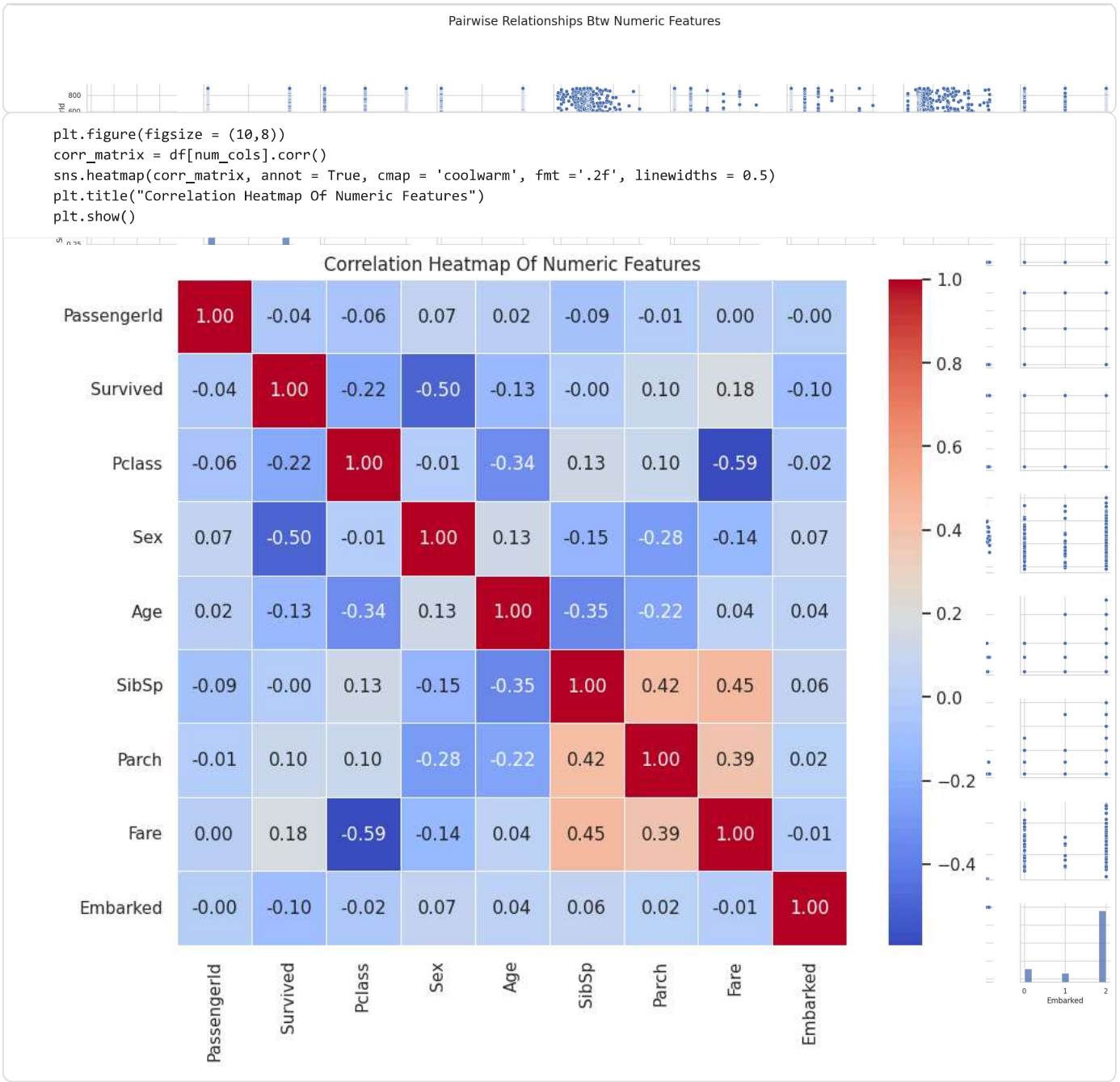



0
1
2
3
4
5
SibSp


Boxplot of Parch
0
1
2
3
4
5
6
Parch


Boxplot of Fare
−1
0
1
2
3
Fare


Boxplot of Embarked
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Embarked




```
plt.figure(figsize = (10,8))
corr_matrix = df[num_cols].corr()
sns.heatmap(corr_matrix, annot = True, cmap = 'coolwarm', fmt ='.2f', linewidths = 0.5)
plt.title("Correlation Heatmap Of Numeric Features")
plt.show()
```

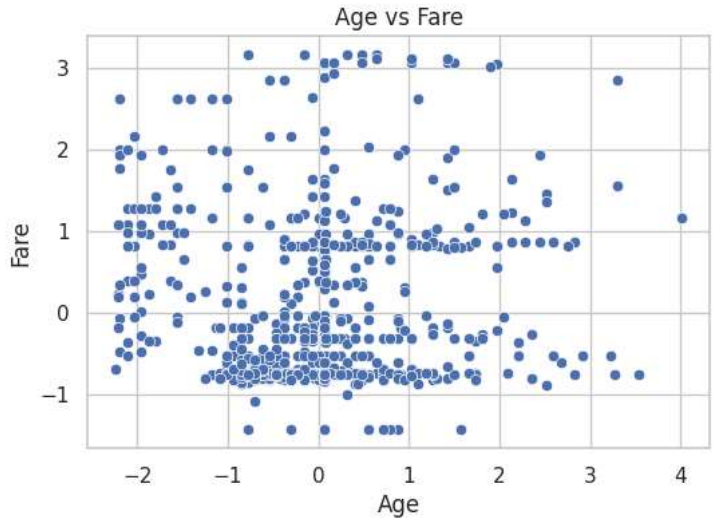
Identifying Patterns, Trends, Anomalies In The Data

```
display(df.describe())

corr = df.corr(numeric_only=True)
display(corr)
```

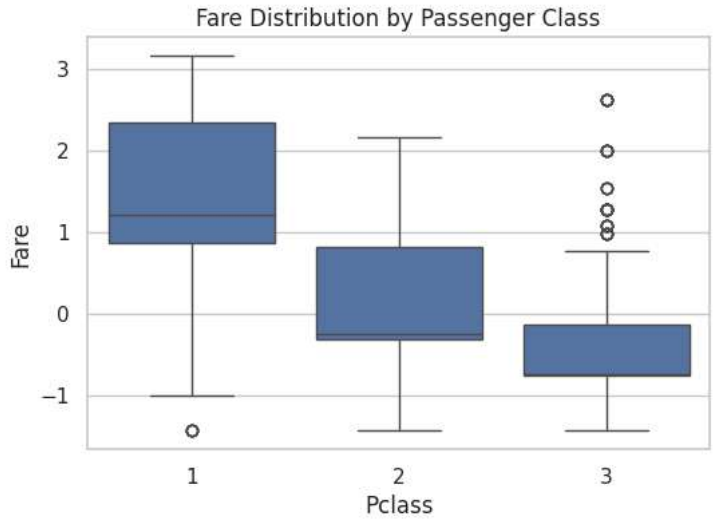
	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
count	750.000000	750.000000	750.000000	750.000000	7.500000e+02	750.000000	750.000000	7.500000e+02	750.000000
mean	445.746667	0.325333	2.508000	0.688000	3.647453e-16	0.438667	0.337333	2.652693e-16	1.614667
std	260.420103	0.468812	0.711766	0.463619	1.000667e+00	0.910317	0.790172	1.000667e+00	0.723106
min	1.000000	0.000000	1.000000	0.000000	-2.236560e+00	0.000000	0.000000	-1.423747e+00	0.000000
25%	214.250000	0.000000	2.000000	0.000000	-5.426332e-01	0.000000	0.000000	-7.415157e-01	2.000000
50%	449.500000	0.000000	3.000000	1.000000	6.171059e-02	0.000000	0.000000	-3.625941e-01	2.000000
75%	670.750000	1.000000	3.000000	1.000000	3.993092e-01	1.000000	0.000000	8.089940e-01	2.000000
max	891.000000	1.000000	3.000000	1.000000	4.010089e+00	5.000000	6.000000	3.164321e+00	2.000000

	PassengerId	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Embarked
PassengerId	1.000000	-0.036090	-0.060263	0.065594	0.021152	-0.087449	-0.012340	0.003882	-0.000980
Survived	-0.036090	1.000000	-0.219870	-0.502914	-0.127323	-0.003237	0.096196	0.180374	-0.098375
Pclass	-0.060263	-0.219870	1.000000	-0.012656	-0.337690	0.127484	0.100833	-0.593969	-0.023834
Sex	0.065594	-0.502914	-0.012656	1.000000	0.132504	-0.146633	-0.284502	-0.139901	0.067033
Age	0.021152	-0.127323	-0.337690	0.132504	1.000000	-0.352495	-0.222272	0.039146	0.042654
SibSp	-0.087449	-0.003237	0.127484	-0.146633	-0.352495	1.000000	0.417657	0.447603	0.058363
Parch	-0.012340	0.096196	0.100833	-0.284502	-0.222272	0.417657	1.000000	0.385990	0.024510
Fare	0.003882	0.180374	-0.593969	-0.139901	0.039146	0.447603	0.385990	1.000000	-0.005163
Embarked	-0.000980	-0.098375	-0.023834	0.067033	0.042654	0.058363	0.024510	-0.005163	1.000000

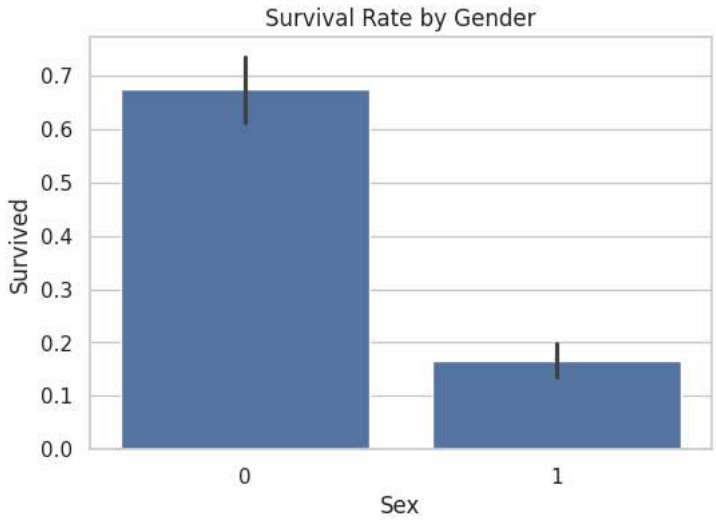
```
plt.figure(figsize=(6,4))
sns.scatterplot(x='Age', y='Fare', data=df)
plt.title('Age vs Fare')
plt.show()
```



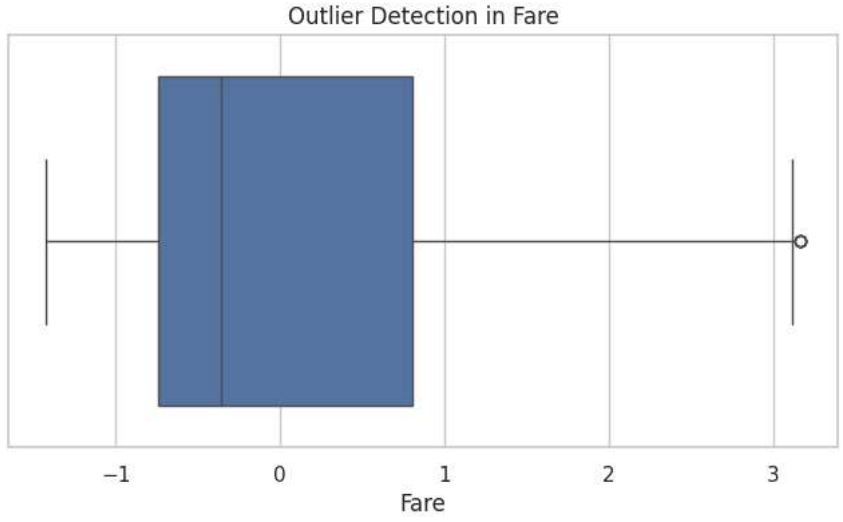

```
plt.figure(figsize=(6,4))
sns.boxplot(x='Pclass', y='Fare', data=df)
plt.title('Fare Distribution by Passenger Class')
plt.show()
```




```
plt.figure(figsize=(6,4))
sns.barplot(x='Sex', y='Survived', data=df)
plt.title('Survival Rate by Gender')
plt.show()
```

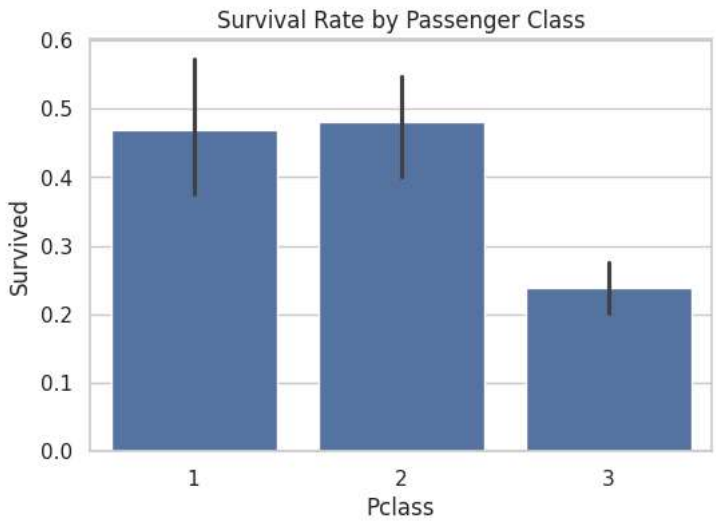



```
plt.figure(figsize=(8,4))
sns.boxplot(x=df['Fare'])
plt.title('Outlier Detection in Fare')
plt.show()
```




```
plt.figure(figsize = (6,4))
sns.barplot(x='Pclass', y='Survived', data=df)
```

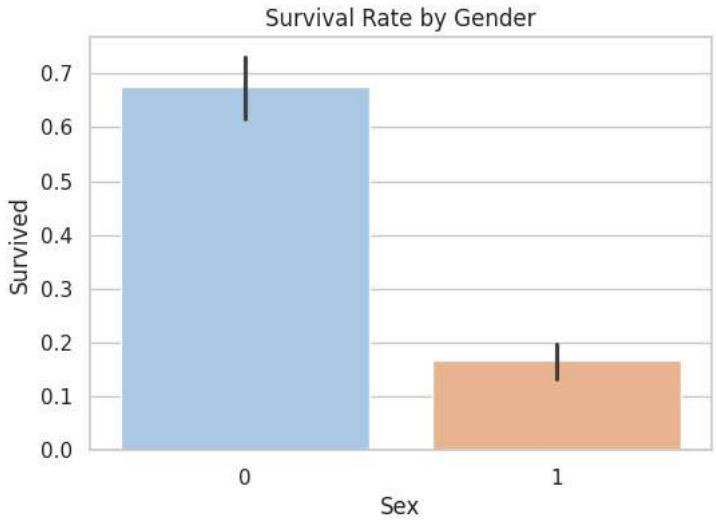
```
plt.title('Survival Rate by Passenger Class')
plt.show()
```



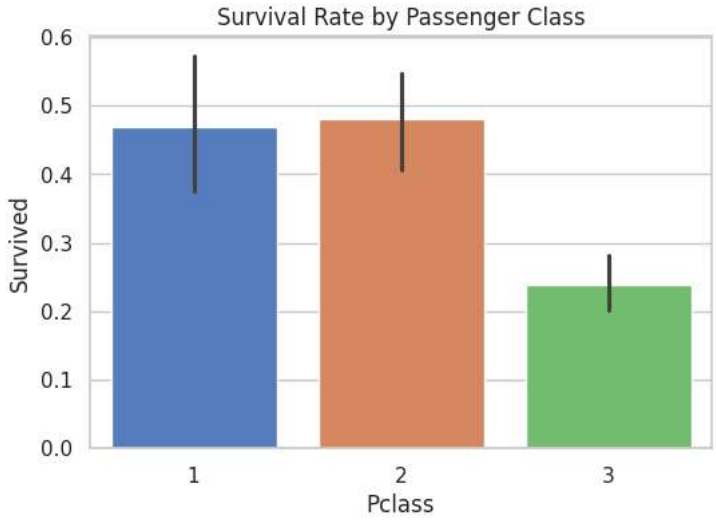
Basic Feature-Level Inferences From Visuals

```
plt.figure(figsize=(6,4))
sns.barplot(x='Sex', y='Survived', data=df, palette='pastel')
plt.title('Survival Rate By Gender')
plt.show()

plt.figure(figsize=(6,4))
sns.barplot(x='Pclass', y='Survived', data=df, palette='muted')
plt.title('Survival Rate By Passenger Class')
plt.show()

plt.figure(figsize=(6,4))
sns.boxplot(x='Pclass', y='Fare', data=df, palette='coolwarm')
plt.title('Fare Distribution By Passenger Class')
plt.show()

plt.figure(figsize=(6,4))
sns.kdeplot(data=df[df['Survived']==1]['Age'], label='Survived', shade=True)
sns.kdeplot(data=df[df['Survived']==0]['Age'], label='Not Survived', shade=True)
plt.title('Age Distribution By Survival')
plt.legend()
plt.show()

plt.figure(figsize=(6,4))
sns.barplot(x='Embarked', y='Survived', data=df, palette='viridis')
plt.title('Survival Rate By Embarkation Port')
plt.show()
```

```
/tmp/ipython-input-3880201168.py:4: FutureWarning:

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

  sns.barplot(x='Sex', y='Survived', data=df, palette='pastel')
```




```
/tmp/ipython-input-3880201168.py:9: FutureWarning:

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

  sns.barplot(x='Pclass', y='Survived', data=df, palette='muted')
```




```
/tmp/ipython-input-3880201168.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

  sns.boxplot(x='Pclass', y='Fare', data=df, palette='coolwarm')
```