

```
In [1]: #Importing necessary libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

In [2]: #Loading the dataset as a Dataframe
df = pd.read_csv("train.csv")
```

```
In [3]: df.head()
```

in [3]: df.head()

Out [3]:

	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	Cummings, 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
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
In [4]: 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            891 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB

In [5]: df.describe()
```

	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

```
In [6]: #Value counts for showing how frequently each categorical value appears
for col in df.select_dtypes(include=['object']).columns:
    print(f'\n--- Value counts for {col} ---')
    print(df[col].value_counts())

--- Value counts for Name ---
Name
Braund, Mr. Owen Harris      1
Boulos, Mr. Hanna            1
Frolicher-Stehli, Mr. Maximilian  1
Gliński, Mr. Eliezer         1
Murdlin, Mr. Joseph          1
..
Kelly, Miss. Anna Katherine "Annie Kate"  1
McCoy, Mr. Bernard           1
Johnson, Mr. William Cahoon Jr  1
Keane, Miss. Nora A          1
Dooley, Mr. Patrick           1
Name: count, Length: 891, dtype: int64

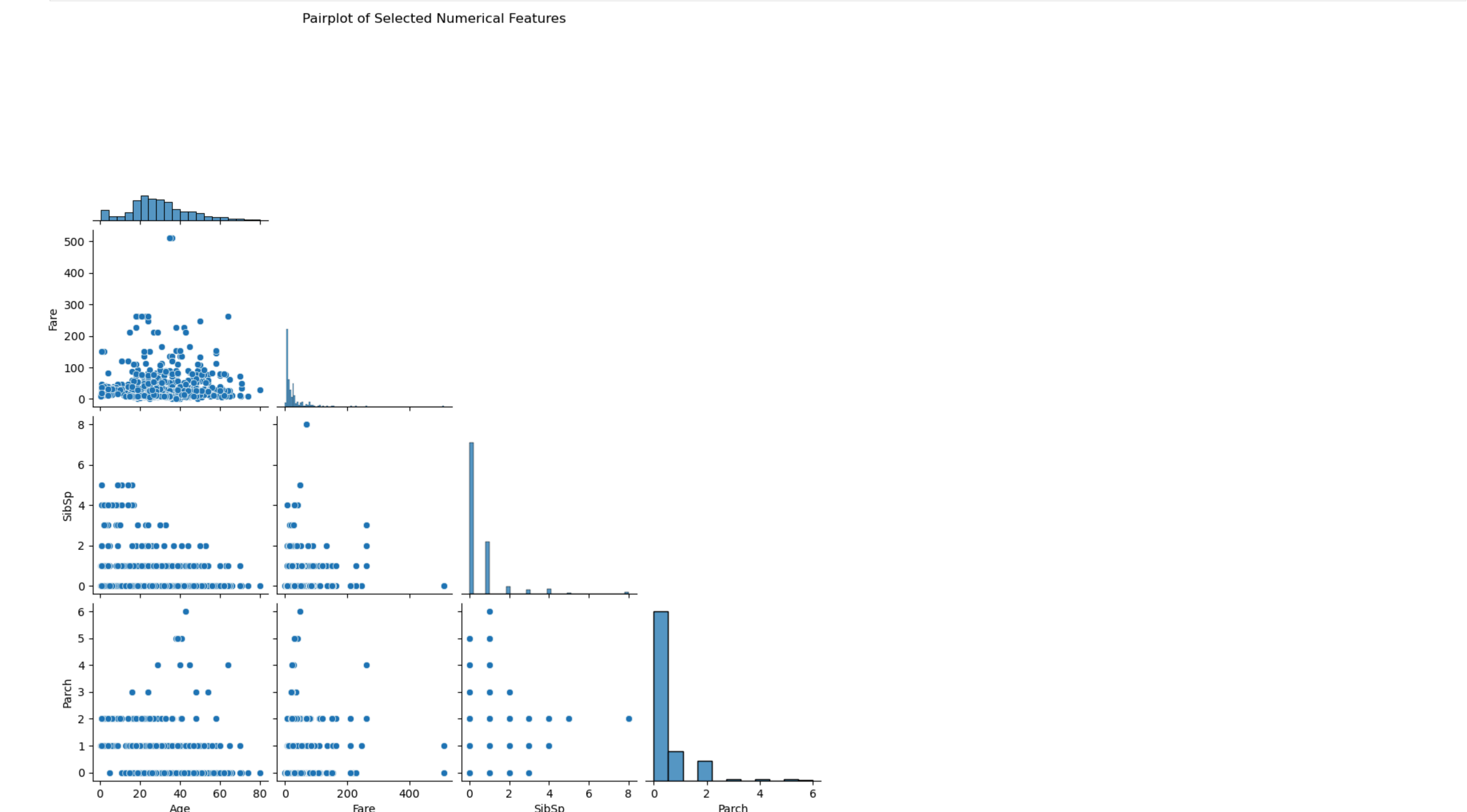
--- Value counts for Sex ---
Sex
male      577
female    314
Name: count, dtype: int64

--- Value counts for Ticket ---
Ticket
347082      7
Ca. 2343    7
1601        7
3101295     6
Ca 2144     6
..
9234        1
19988       1
2693        1
PC 17612    1
370376      1
Name: count, Length: 681, dtype: int64

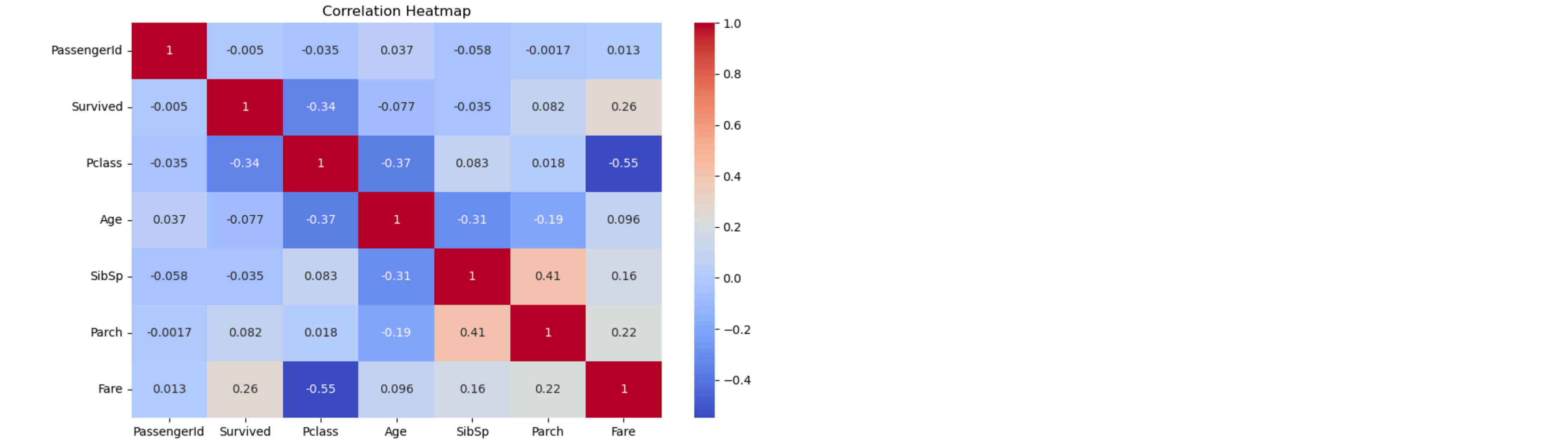
--- Value counts for Cabin ---
Cabin
B96 B98      4
G6           4
C23 C25 C27  4
C22 C26      3
F33          3
..
E34          1
C7           1
C54          1
E36          1
C148         1
Name: count, Length: 147, dtype: int64

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

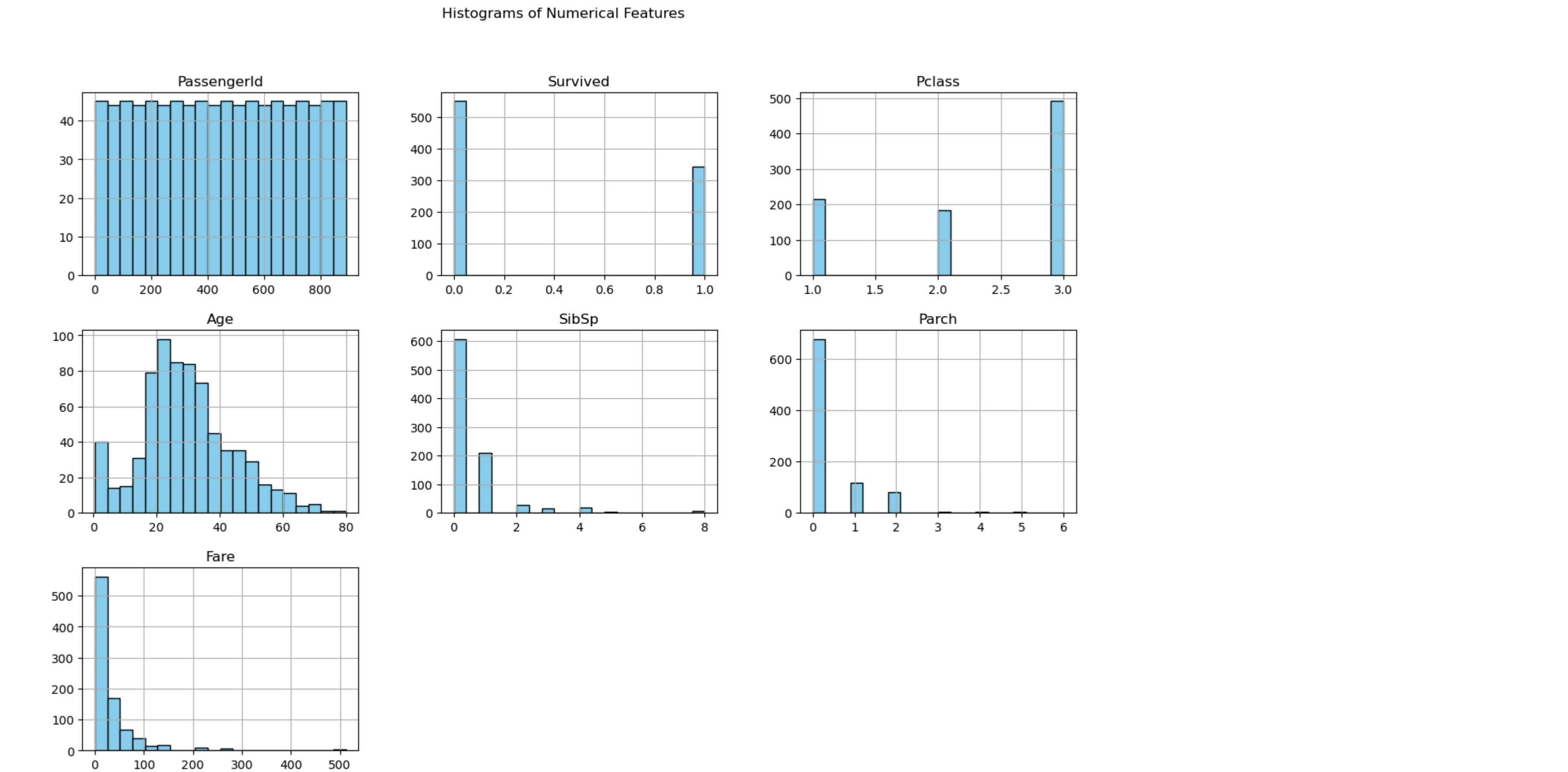
In [14]: #Pairplot of selected important columns that makes it faster and cleaner.
selected_cols = ["Age", "Fare", "SibSp", "Parch"]
sns.pairplot(df[selected_cols], diag_kind="hist", corner=True)
plt.suptitle("Pairplot of Selected Numerical Features", y=1.02)
plt.show()
```



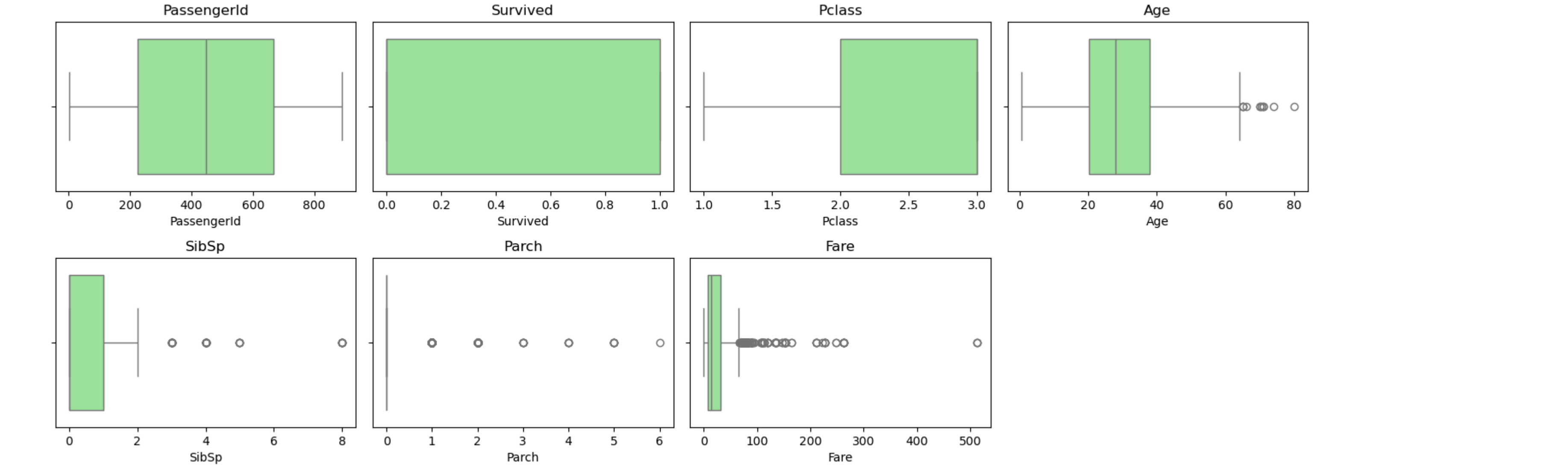
```
In [8]: #Correlation heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```



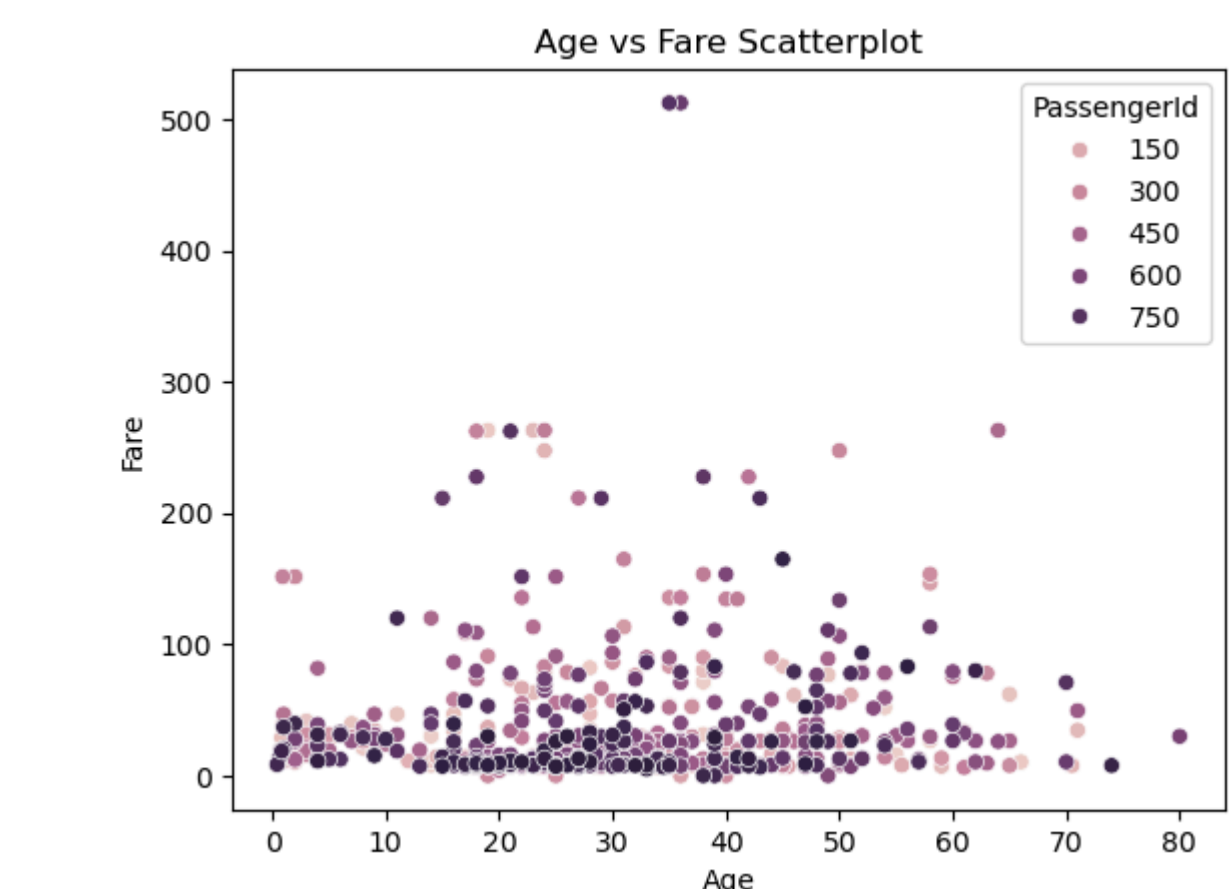
```
In [10]: # Histograms for numerical columns
df.hist(bins=20, figsize=(15, 10), color='skyblue', edgecolor='black')
plt.suptitle("Histograms of Numerical Features")
plt.show()
```



```
In [11]: # Boxplots for outlier detection
num_cols = df.select_dtypes(include=['number']).columns
plt.figure(figsize=(15, 8))
for i, col in enumerate(num_cols, 1):
    plt.subplot(3, 4, i)
    sns.boxplot(x=df[col], color='lightgreen')
    plt.title(col)
plt.tight_layout()
plt.show()
```



```
In [12]: # Scatterplots (Age vs Fare)
if "Age" in df.columns and "Fare" in df.columns:
    sns.scatterplot(x="Age", y="Fare", data=df, hue=df.columns[0])
    plt.title("Age vs Fare Scatterplot")
    plt.show()
```

In []: