02_exploratory_data_analysis

June 24, 2025

[7]:
```
# Step 1: Import necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import os

# Make plots look nice
sns.set(style="whitegrid")
plt.style.use("ggplot")

# Show all columns when printing dataframes
pd.set_option('display.max_columns', None)

# Step 2: Load the cleaned dataset (correct path since we're inside notebooks/)
df = pd.read_csv('../data/titanic_cleaned.csv')

# Step 3: Show first few rows# Step 1: Import necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import os

# Make plots look nice
sns.set(style="whitegrid")
plt.style.use("ggplot")

# Show all columns when printing dataframes
pd.set_option('display.max_columns', None)

# Step 2: Load the cleaned dataset (correct path since we're inside notebooks/)
df = pd.read_csv('../data/titanic_cleaned.csv')

# Step 3: Show first few rows
df.head()
```

```
df.head()
```

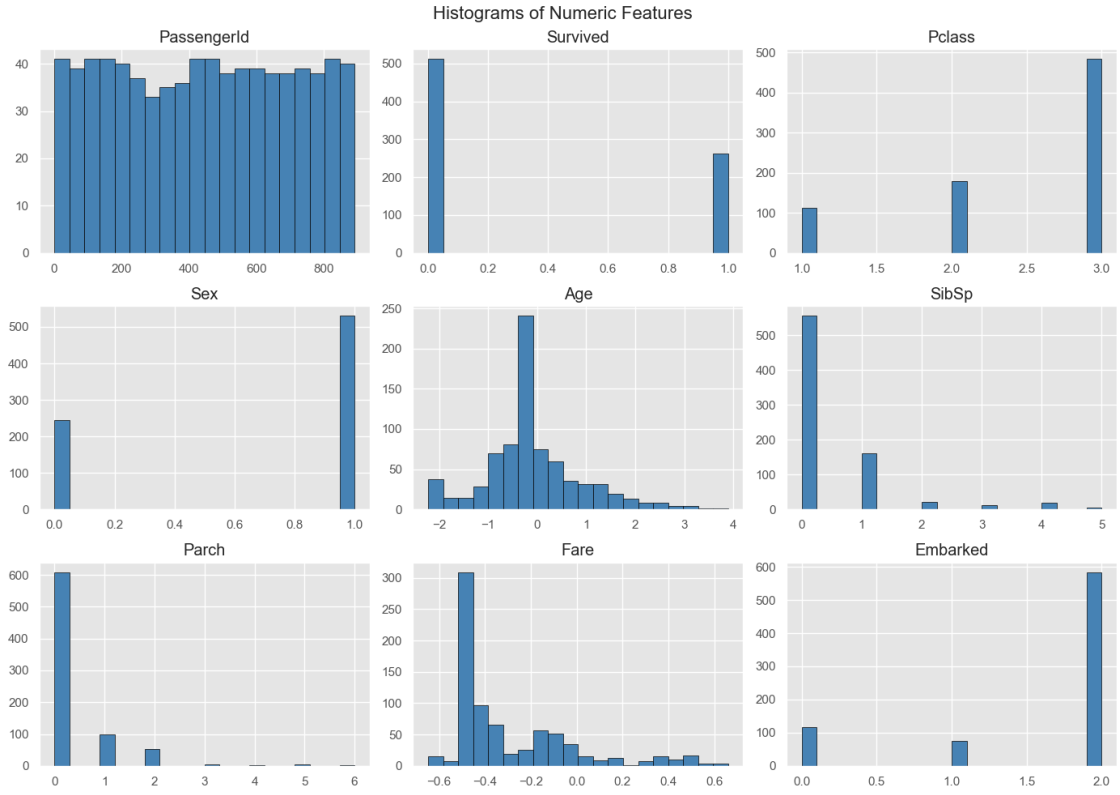
[7]:
```
   PassengerId  Survived  Pclass  Sex       Age  SibSp  Parch      Fare  \
0            1         0       3    1 -0.565736      1      0 -0.502445
1            3         1       3    0 -0.258337      0      0 -0.488854
2            4         1       1    0  0.433312      1      0  0.420730
3            5         0       3    1  0.433312      0      0 -0.486337
4            6         0       3    1 -0.104637      0      0 -0.478116

   Embarked
0         2
1         2
2         2
3         2
4         1
```

[8]:
```
# Step 2: Summary statistics for numerical columns
df.describe().T
```

[8]:
```
             count        mean         std       min         25%         50%  \
PassengerId  775.0  445.806452  260.116285  1.000000  213.500000  450.000000
Survived     775.0    0.339355    0.473796  0.000000    0.000000    0.000000
Pclass       775.0    2.480000    0.734390  1.000000    2.000000    3.000000
Sex          775.0    0.685161    0.464752  0.000000    0.000000    1.000000
Age          775.0   -0.047099    0.982304 -2.224156   -0.565736   -0.104637
SibSp        775.0    0.437419    0.899838  0.000000    0.000000    0.000000
Parch        775.0    0.340645    0.785914  0.000000    0.000000    0.000000
Fare         775.0   -0.289579    0.273391 -0.648422   -0.489442   -0.386671
Embarked     775.0    1.603871    0.734344  0.000000    2.000000    2.000000

                    75%         max
PassengerId  670.500000  891.000000
Survived       1.000000    1.000000
Pclass         3.000000    3.000000
Sex            1.000000    1.000000
Age            0.356462    3.891554
SibSp          1.000000    5.000000
Parch          0.000000    6.000000
Fare          -0.124920    0.660333
Embarked       2.000000    2.000000
```

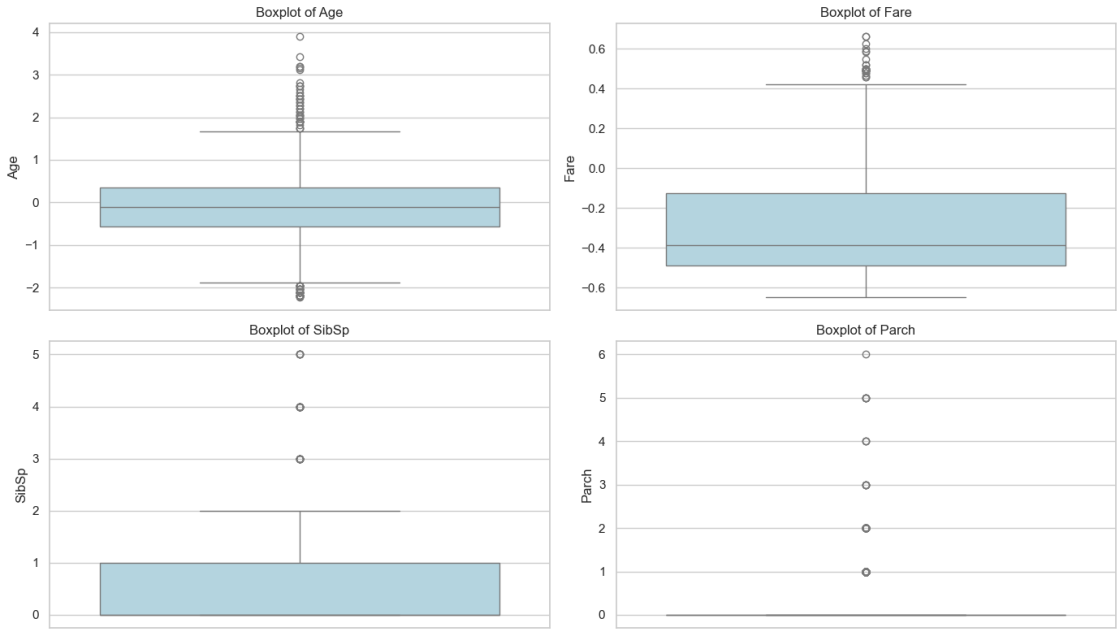
[13]:
```
import os

# Step 3: Histograms for numeric features
df.hist(bins=20, figsize=(14, 10), color='steelblue', edgecolor='black')
plt.suptitle("Histograms of Numeric Features", fontsize=16)
plt.tight_layout()
```

```
#  Ensure images/ folder exists before saving
os.makedirs("../images", exist_ok=True)
plt.savefig("../images/histograms.png")

plt.show()
```



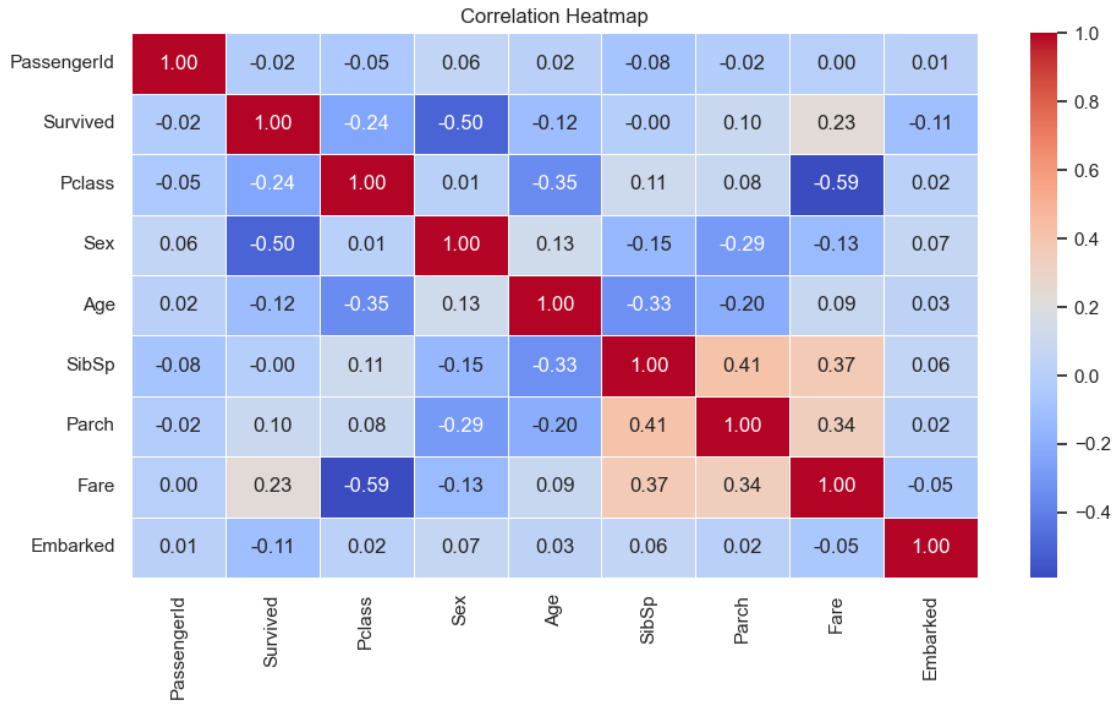
[14]:
```
import os
import matplotlib.pyplot as plt
import seaborn as sns

# step 4: List of numeric columns for boxplots
num_cols = ['Age', 'Fare', 'SibSp', 'Parch']

# Set up plot style
plt.style.use("ggplot")
sns.set(style="whitegrid")

# Create a figure for the boxplots
plt.figure(figsize=(14, 8))
for i, col in enumerate(num_cols, 1):
    plt.subplot(2, 2, i)
```

```
    sns.boxplot(data=df, y=col, color='lightblue')
    plt.title(f'Boxplot of {col}')

plt.tight_layout()

#  Ensure images/ folder exists before saving
os.makedirs("../images", exist_ok=True)
plt.savefig("../images/boxplots.png")

plt.show()
```



[15]:
```
import os
import matplotlib.pyplot as plt
import seaborn as sns

# Set plot styles
plt.style.use("ggplot")
sns.set(style="whitegrid")

# Calculate correlation matrix
corr_matrix = df.corr(numeric_only=True)

# step 5: Plot the heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title("Correlation Heatmap")
```

```
plt.tight_layout()

#  Ensure images/ folder exists before saving
os.makedirs("../images", exist_ok=True)
plt.savefig("../images/correlation_heatmap.png")

plt.show()
```



0.1 EDA Summary – Key Findings

1. **Class & Fare strongly relate to survival** – Passengers in higher classes and those who paid more had a better chance of survival.
2. **Pclass and Fare are inversely correlated** – Higher the class (lower the number), higher the fare.
3. **Most people traveled alone** – Many had `SibSp = 0` and `Parch = 0`.
4. **Age had a broad spread**, with some outliers, but no strong link to survival.
5. **Embarked is mostly 'S' (encoded as 2)** — very few passengers from ports Q or C.

0.2 Ready for Modeling!

The dataset has been cleaned, understood through visualizations and statistics, and is now ready to be used in machine learning models.

[ ]: