Exploratory Data Analysis (EDA)

Over Titanic Dataset (Source: Kaggle)

Notes:

- After running this notebook you should find 'titanic_with_features.csv' in the repository root.

Step 1: Setup and Data Loading

Import all necessary libraries and set global plot styles

In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

plt.style.use("default")
sns.set_palette("husl")
plt.rcParams['figure.figsize'] = (10, 6)
```

Load the Titanic Dataset

In [2]:
```python
df = pd.read_csv('data/titanic.csv')
print("Data loaded successfully")
print(f"Dataset shape: {df.shape}")
```

```
Data loaded successfully
Dataset shape: (891, 12)
```

Step 2: Initial Data Exploration

Basic information about the dataset

In [3]:
```python
print("Dataset Information:")
print(df.info())
```

```
Dataset Information:
<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
None
```

Display first few rows

In [4]:
```
print("First 5 rows:")
print(df.head())
```

```
First 5 rows:
   PassengerId  Survived  Pclass  \
0            1         0       3   
1            2         1       1   
2            3         1       3   
3            4         1       1   
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  
```

Basic statistical summary

In [5]:
```
print("Statistical Summary:")
print(df.describe())
```

```
Statistical Summary:
       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
```

Check for Missing Values

In [6]:
```
print("Missing Values:")
missing_data = df.isnull().sum()
missing_percentage = (missing_data / len(df)) * 100
missing_df = pd.DataFrame({
    'Missing Count': missing_data,
    'Percentage': missing_percentage
})
print(missing_df[missing_df['Missing Count'] > 0])
```

```
Missing Values:
          Missing Count  Percentage
Age                 177   19.865320
Cabin               687   77.104377
Embarked              2    0.224467
```

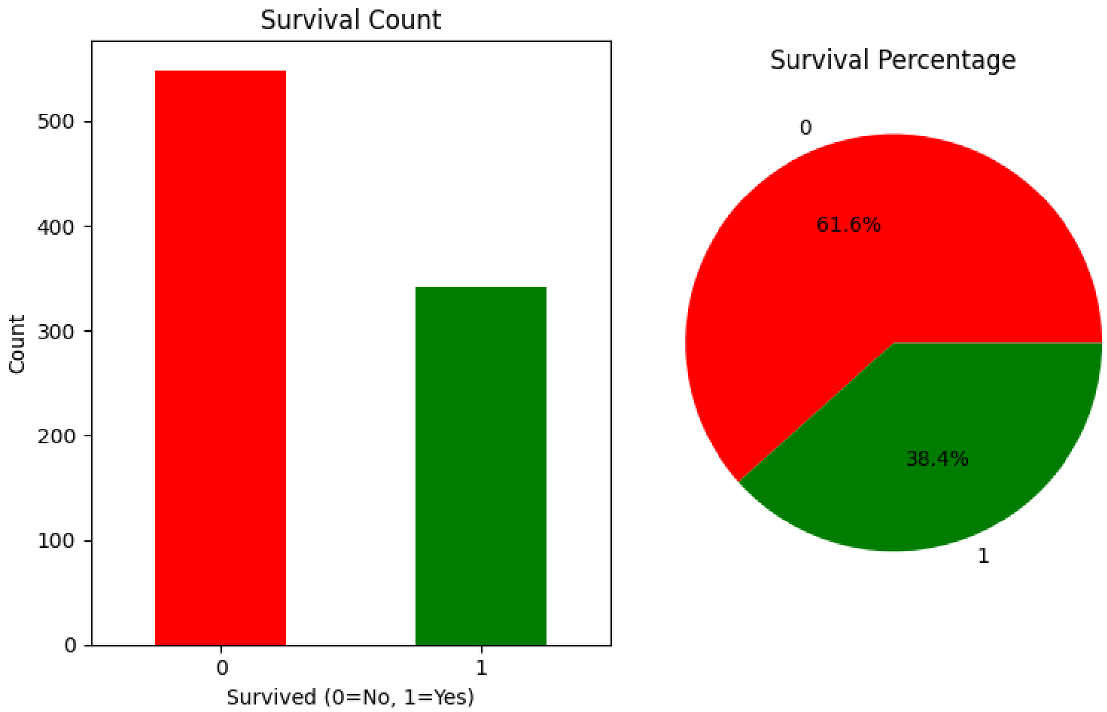
Step 3: Categorical Variables Analysis

Survival rate analysis

In [7]:
```
print("=== SURVIVAL ANALYSIS ===")
survival_counts = df['Survived'].value_counts()
survival_rate = df['Survived'].mean()
print(f"Survival Counts:\n{survival_counts}")
print(f"Overall Survival Rate: {survival_rate:.2%}")

# Visualize survival
plt.figure(figsize=(8, 5))
plt.subplot(1, 2, 1)
df['Survived'].value_counts().plot(kind='bar', color=['red', 'green'])
plt.title('Survival Count')
plt.xlabel('Survived (0=No, 1=Yes)')
plt.ylabel('Count')
plt.xticks(rotation=0)

plt.subplot(1, 2, 2)
df['Survived'].value_counts().plot(kind='pie', autopct='%1.1f%%', colors=['red',
```

```
plt.title('Survival Percentage')
plt.ylabel('')
plt.tight_layout()
plt.show()
```

```
=== SURVIVAL ANALYSIS ===
Survival Counts:
Survived
0    549
1    342
Name: count, dtype: int64
Overall Survival Rate: 38.38%
```



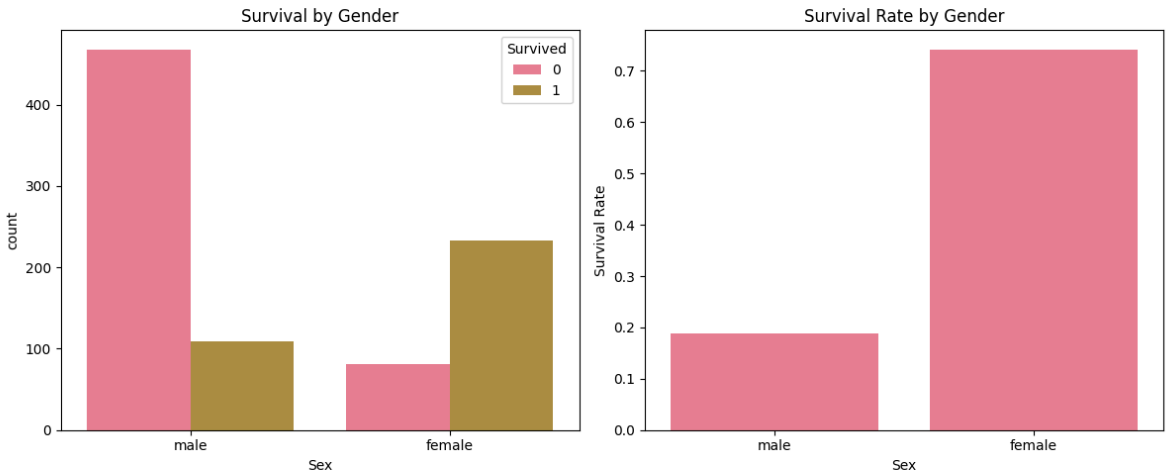
Gender analysis

In [8]:

```
print("=== GENDER ANALYSIS ===")
gender_counts = df['Sex'].value_counts()
print(f"Gender Distribution:\n{gender_counts}")

# Gender vs Survival
gender_survival = pd.crosstab(df['Sex'], df['Survived'], margins=True)
print(f"\nGender vs Survival:\n{gender_survival}")

# Survival rate by gender
survival_by_gender = df.groupby('Sex')['Survived'].agg(['count', 'sum', 'mean'])
survival_by_gender.columns = ['Total', 'Survived', 'Survival_Rate']
print(f"\nSurvival Rate by Gender:\n{survival_by_gender}")

# Visualize
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.countplot(data=df, x='Sex', hue='Survived')
plt.title('Survival by Gender')

plt.subplot(1, 2, 2)
sns.barplot(data=df, x='Sex', y='Survived', ci=None)
plt.title('Survival Rate by Gender')
plt.ylabel('Survival Rate')
```

```
plt.tight_layout()
plt.show()
```

```
=== GENDER ANALYSIS ===
Gender Distribution:
Sex
male      577
female    314
Name: count, dtype: int64

Gender vs Survival:
Survived    0    1  All
Sex
female     81  233  314
male      468  109  577
All       549  342  891

Survival Rate by Gender:
        Total  Survived  Survival_Rate
Sex
female    314       233       0.742038
male      577       109       0.188908
```

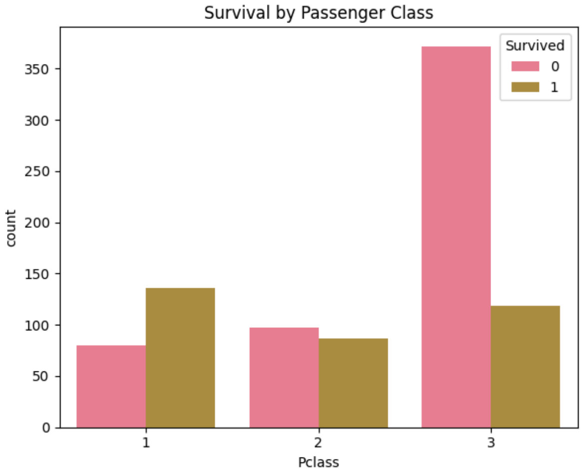


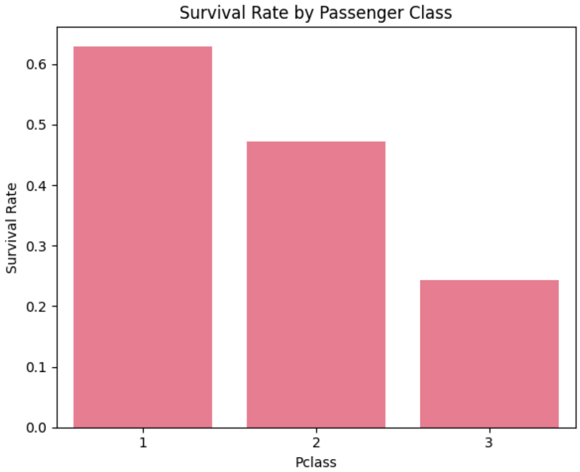
Passenger Class analysis

In [9]:
```
print("=== PASSENGER CLASS ANALYSIS ===")
class_counts = df['Pclass'].value_counts().sort_index()
print(f"Class Distribution:\n{class_counts}")

# Class vs Survival
class_survival = pd.crosstab(df['Pclass'], df['Survived'], margins=True)
print(f"\nClass vs Survival:\n{class_survival}")

# Survival rate by class
survival_by_class = df.groupby('Pclass')['Survived'].agg(['count', 'sum', 'mean'
survival_by_class.columns = ['Total', 'Survived', 'Survival_Rate']
print(f"\nSurvival Rate by Class:\n{survival_by_class}")

# Visualize
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
sns.countplot(data=df, x='Pclass', hue='Survived')
plt.title('Survival by Passenger Class')

plt.subplot(1, 2, 2)
sns.barplot(data=df, x='Pclass', y='Survived', ci=None)
```

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

```
=== PASSENGER CLASS ANALYSIS ===
Class Distribution:
Pclass
1    216
2    184
3    491
Name: count, dtype: int64

Class vs Survival:
Survived    0    1  All
Pclass
1          80  136  216
2          97   87  184
3         372  119  491
All       549  342  891

Survival Rate by Class:
        Total  Survived  Survival_Rate
Pclass
1         216       136       0.629630
2         184        87       0.472826
3         491       119       0.242363
```





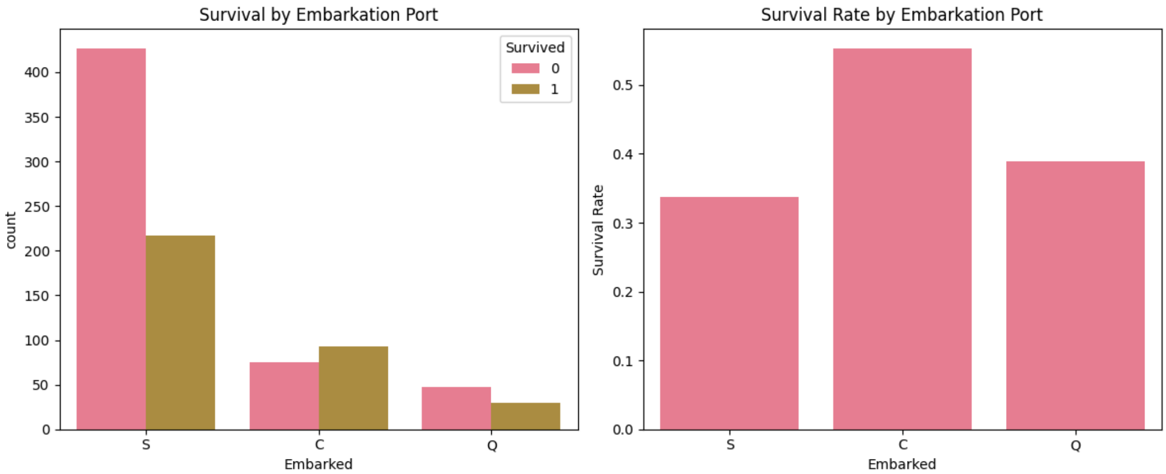
Embarkation Port analysis

In [10]:

```python
# Embarkation port analysis
print("=== EMBARKATION ANALYSIS ===")
embark_counts = df['Embarked'].value_counts()
print(f"Embarkation Distribution:\n{embark_counts}")

# Embarked vs Survival
embark_survival = pd.crosstab(df['Embarked'], df['Survived'], margins=True)
print(f"\nEmbarkation vs Survival:\n{embark_survival}")

# Survival rate by embarkation
survival_by_embark = df.groupby('Embarked')['Survived'].agg(['count', 'sum', 'me
survival_by_embark.columns = ['Total', 'Survived', 'Survival_Rate']
print(f"\nSurvival Rate by Embarkation:\n{survival_by_embark}")

# Visualize
plt.figure(figsize=(12, 5))
```

```python
plt.subplot(1, 2, 1)
sns.countplot(data=df, x='Embarked', hue='Survived')
plt.title('Survival by Embarkation Port')

plt.subplot(1, 2, 2)
sns.barplot(data=df, x='Embarked', y='Survived', ci=None)
plt.title('Survival Rate by Embarkation Port')
plt.ylabel('Survival Rate')
plt.tight_layout()
plt.show()
```

```
=== EMBARKATION ANALYSIS ===
Embarkation Distribution:
Embarked
S    644
C    168
Q     77
Name: count, dtype: int64

Embarkation vs Survival:
Survived    0    1  All
Embarked
C          75   93  168
Q          47   30   77
S         427  217  644
All       549  340  889

Survival Rate by Embarkation:
          Total  Survived  Survival_Rate
Embarked
C           168        93       0.553571
Q            77        30       0.389610
S           644       217       0.336957
```



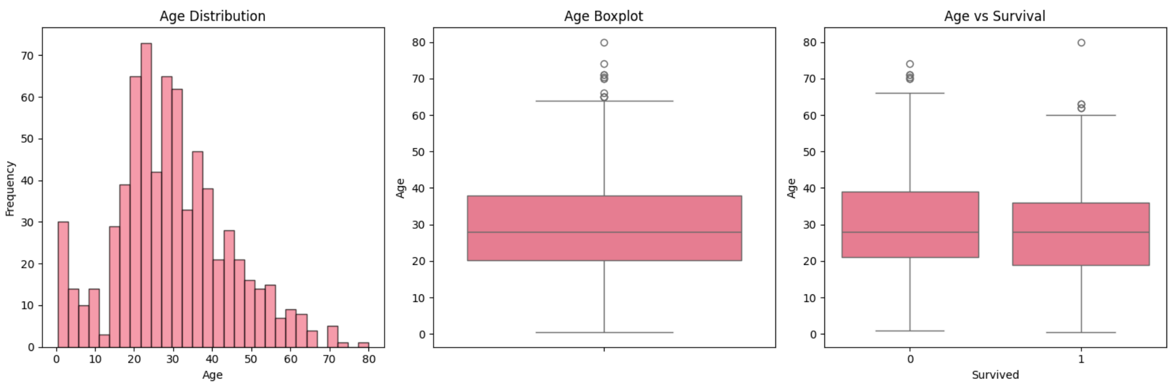
Step 4: Numerical Variables Analysis

Age analysis

In [11]:
```python
print("=== AGE ANALYSIS ===")
age_stats = df['Age'].describe()
print(f"Age Statistics:\n{age_stats}")
print(f"Missing Age values: {df['Age'].isnull().sum()}")

# Age distribution
plt.figure(figsize=(15, 5))
```

```python
plt.subplot(1, 3, 1)
plt.hist(df['Age'].dropna(), bins=30, edgecolor='black', alpha=0.7)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')

plt.subplot(1, 3, 2)
sns.boxplot(y=df['Age'])
plt.title('Age Boxplot')

plt.subplot(1, 3, 3)
sns.boxplot(data=df, x='Survived', y='Age')
plt.title('Age vs Survival')
plt.tight_layout()
plt.show()

# Age vs Survival analysis
age_survival = df.groupby('Survived')['Age'].agg(['count', 'mean', 'median', 'st
print(f"\nAge vs Survival:\n{age_survival}")
```

```
=== AGE ANALYSIS ===
Age Statistics:
count    714.000000
mean      29.699118
std       14.526497
min        0.420000
25%       20.125000
50%       28.000000
75%       38.000000
max       80.000000
Name: Age, dtype: float64
Missing Age values: 177
```




```
Age vs Survival:
          count       mean  median        std
Survived
0           424  30.626179    28.0  14.172110
1           290  28.343690    28.0  14.950952
```

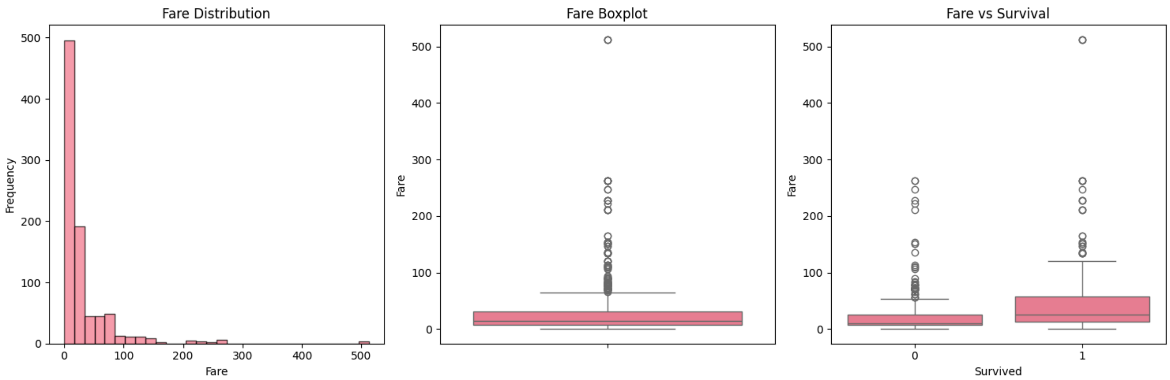
Fare analysis

In [12]:

```python
print("=== FARE ANALYSIS ===")
fare_stats = df['Fare'].describe()
print(f"Fare Statistics:\n{fare_stats}")

# Fare distribution
plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)
plt.hist(df['Fare'], bins=30, edgecolor='black', alpha=0.7)
```

```
plt.title('Fare Distribution')
plt.xlabel('Fare')
plt.ylabel('Frequency')

plt.subplot(1, 3, 2)
sns.boxplot(y=df['Fare'])
plt.title('Fare Boxplot')

plt.subplot(1, 3, 3)
sns.boxplot(data=df, x='Survived', y='Fare')
plt.title('Fare vs Survival')
plt.tight_layout()
plt.show()

# Fare vs Survival analysis
fare_survival = df.groupby('Survived')['Fare'].agg(['count', 'mean', 'median', '
print(f"\nFare vs Survival:\n{fare_survival}")
```

```
=== FARE ANALYSIS ===
Fare Statistics:
count    891.000000
mean      32.204208
std       49.693429
min        0.000000
25%        7.910400
50%       14.454200
75%       31.000000
max      512.329200
Name: Fare, dtype: float64
```



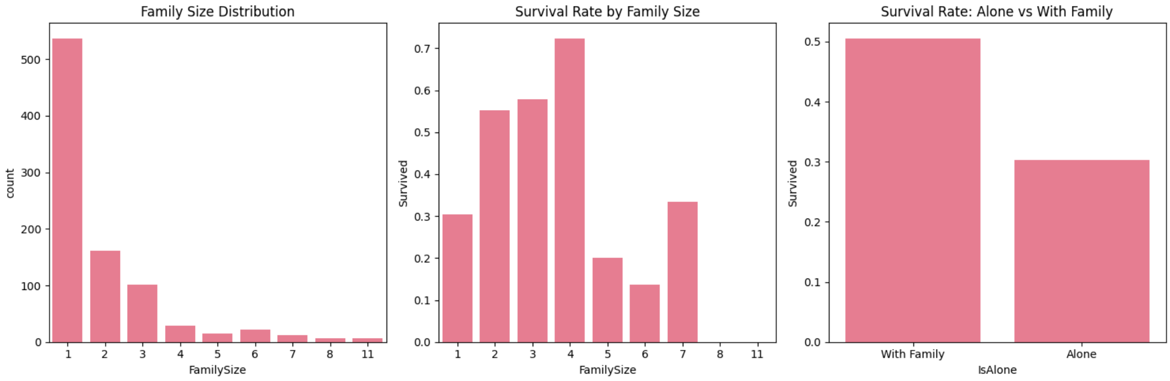

```
Fare vs Survival:
          count       mean  median        std
Survived
0           549  22.117887    10.5  31.388207
1           342  48.395408    26.0  66.596998
```

Family Size analysis

In [13]:
```
# Family size analysis (SibSp + Parch)
print("=== FAMILY SIZE ANALYSIS ===")
df['FamilySize'] = df['SibSp'] + df['Parch'] + 1  # +1 for the passenger themsel
df['IsAlone'] = (df['FamilySize'] == 1).astype(int)

family_stats = df['FamilySize'].describe()
print(f"Family Size Statistics:\n{family_stats}")

# Family size distribution
plt.figure(figsize=(15, 5))
plt.subplot(1, 3, 1)
```

```
sns.countplot(data=df, x='FamilySize')
plt.title('Family Size Distribution')

plt.subplot(1, 3, 2)
sns.barplot(data=df, x='FamilySize', y='Survived', ci=None)
plt.title('Survival Rate by Family Size')

plt.subplot(1, 3, 3)
sns.barplot(data=df, x='IsAlone', y='Survived', ci=None)
plt.title('Survival Rate: Alone vs With Family')
plt.xticks([0, 1], ['With Family', 'Alone'])
plt.tight_layout()
plt.show()

# Family size vs survival
family_survival = df.groupby('FamilySize')['Survived'].agg(['count', 'sum', 'mea
family_survival.columns = ['Total', 'Survived', 'Survival_Rate']
print(f"\nSurvival Rate by Family Size:\n{family_survival}")
```

```
=== FAMILY SIZE ANALYSIS ===
Family Size Statistics:
count    891.000000
mean       1.904602
std        1.613459
min        1.000000
25%        1.000000
50%        1.000000
75%        2.000000
max       11.000000
Name: FamilySize, dtype: float64
```



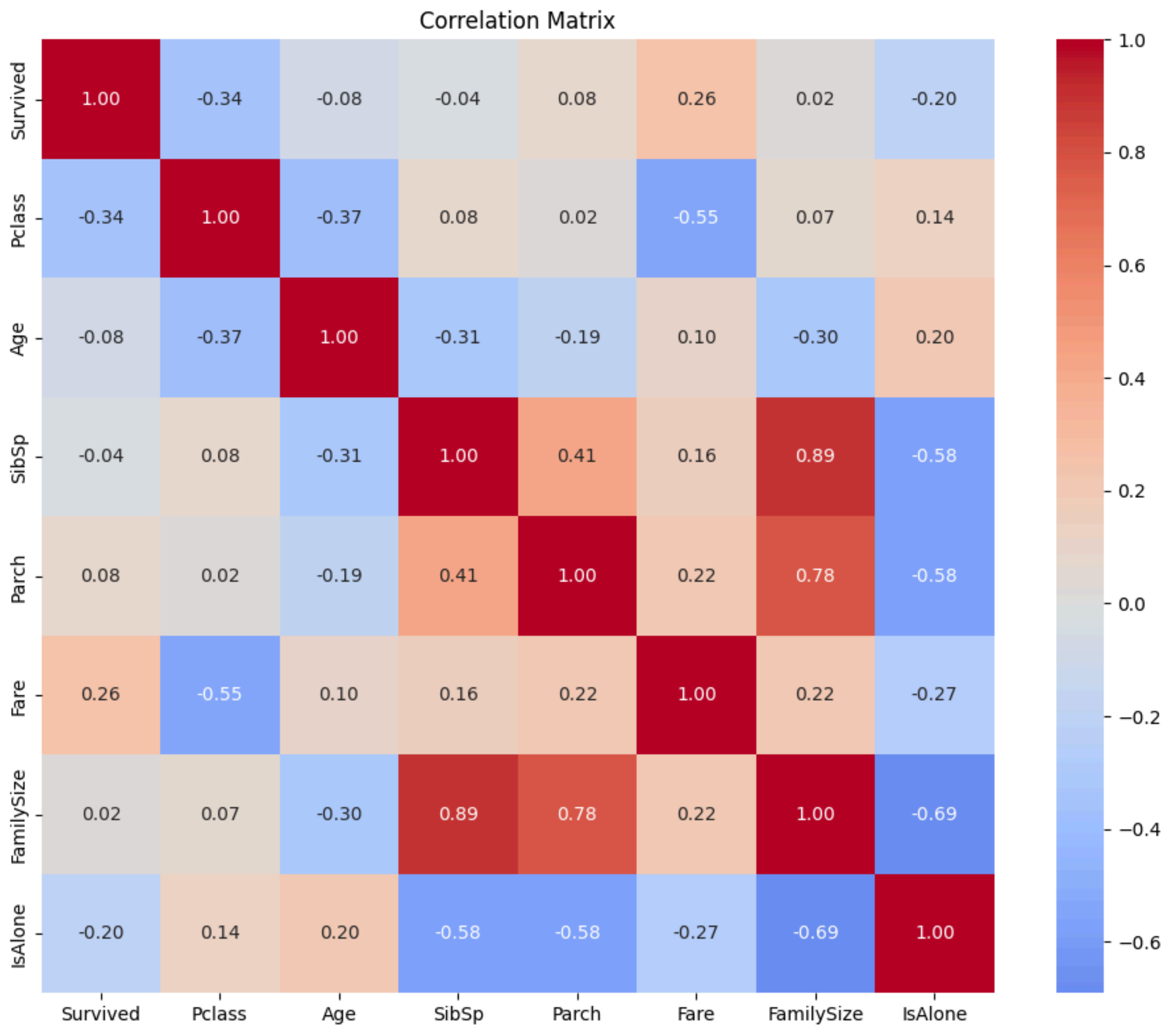

```
Survival Rate by Family Size:
            Total  Survived  Survival_Rate
FamilySize
1             537       163       0.303538
2             161        89       0.552795
3             102        59       0.578431
4              29        21       0.724138
5              15         3       0.200000
6              22         3       0.136364
7              12         4       0.333333
8               6         0       0.000000
11              7         0       0.000000
```

Step 5: Advanced Visualizations

Correlation Analysis (Correlation heatmap)

In [14]:
```
print("=== CORRELATION ANALYSIS ===")
# Select only numeric columns for correlation
numeric_cols = ['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare', 'FamilySi
correlation_matrix = df[numeric_cols].corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', center=0,
            square=True, fmt='.2f')
plt.title('Correlation Matrix')
plt.tight_layout()
plt.show()

print("Strong correlations with Survival:")
survival_corr = correlation_matrix['Survived'].abs().sort_values(ascending=False
print(survival_corr[survival_corr > 0.1])
```

```
=== CORRELATION ANALYSIS ===
```



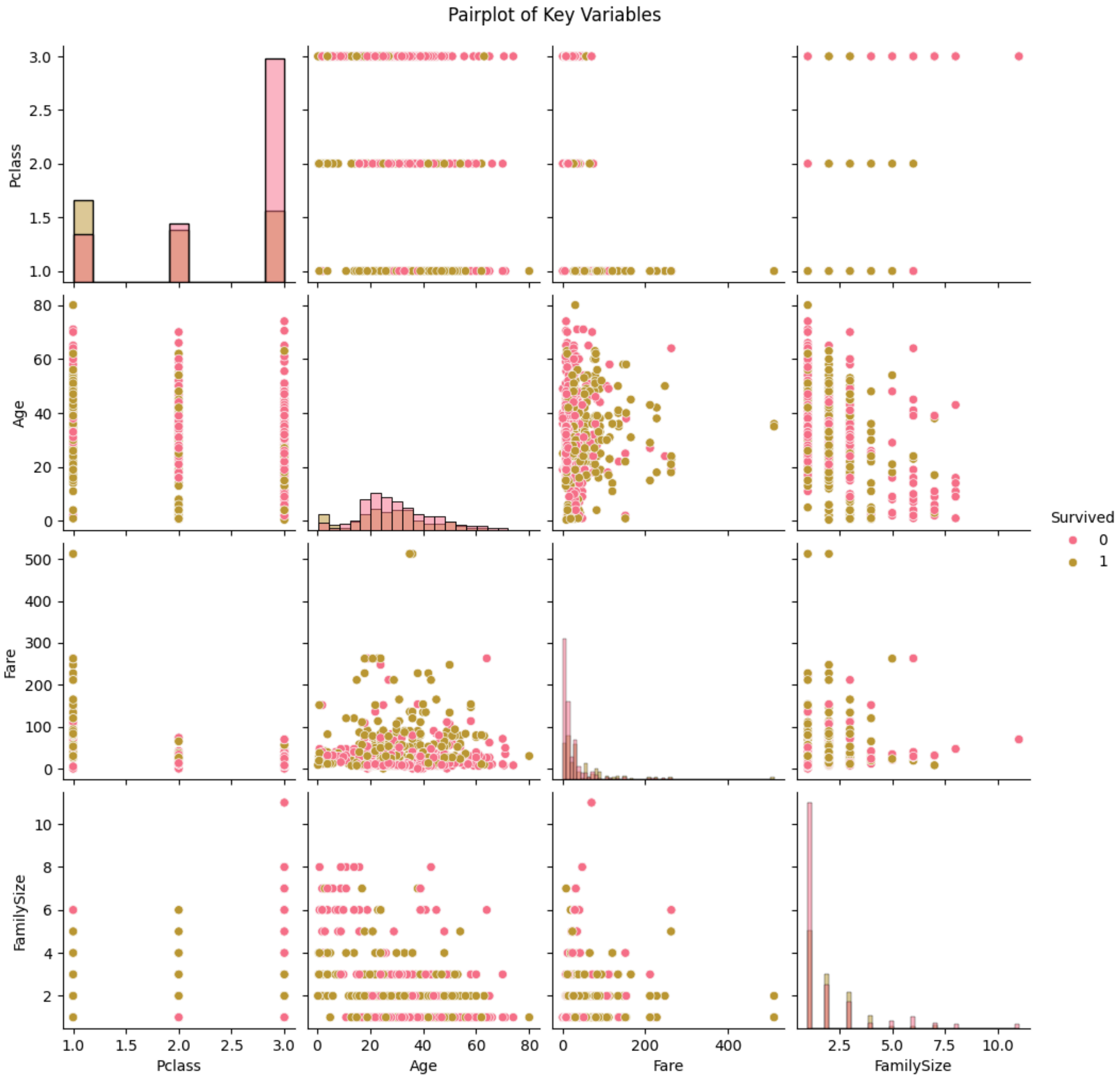

```
Strong correlations with Survival:
Survived    1.000000
Pclass      0.338481
Fare        0.257307
IsAlone     0.203367
Name: Survived, dtype: float64
```

Pairplot for key numeric variables

In [15]:
```
print("=== PAIRPLOT ANALYSIS ===")
plt.figure(figsize=(12, 10))
key_vars = ['Survived', 'Pclass', 'Age', 'Fare', 'FamilySize']
```

```
sns.pairplot(df[key_vars], hue='Survived', diag_kind='hist')
plt.suptitle('Pairplot of Key Variables', y=1.02)
plt.show()
```

```
=== PAIRPLOT ANALYSIS ===
<Figure size 1200x1000 with 0 Axes>
```

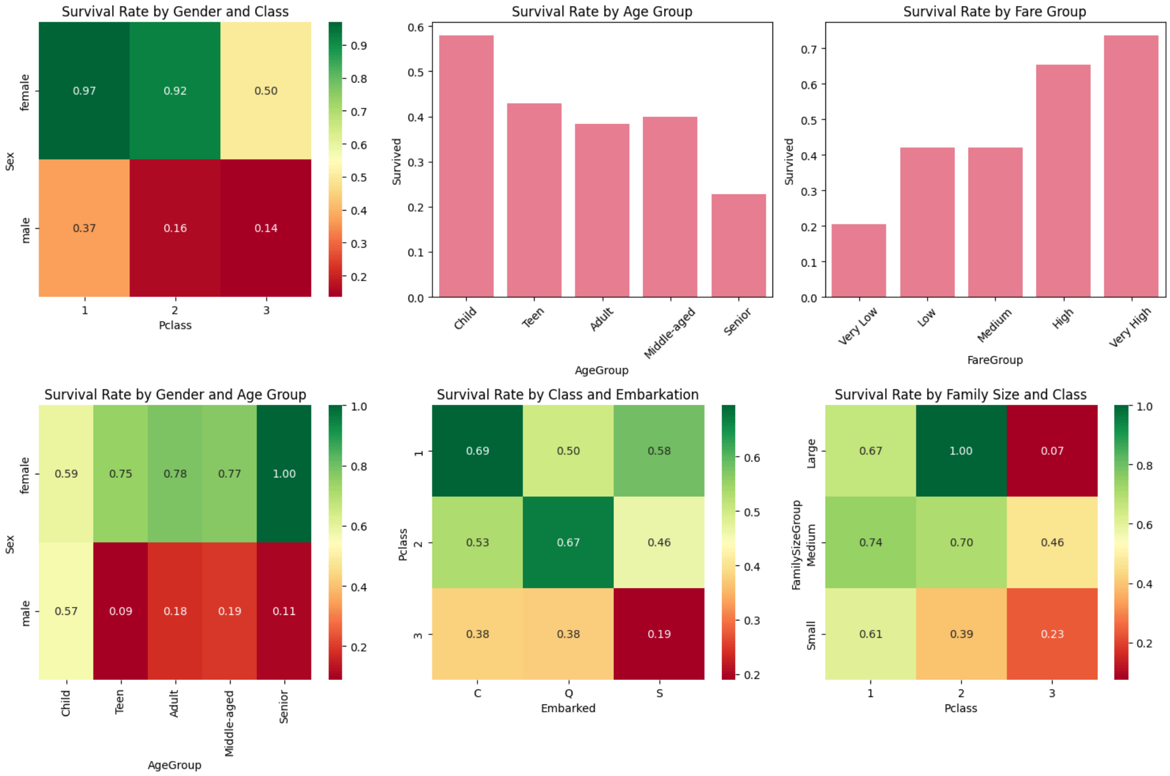


Multi-dimensional analysis

In [16]:
```
print("=== MULTI-DIMENSIONAL ANALYSIS ===")

# Gender + Class vs Survival
plt.figure(figsize=(15, 10))
plt.subplot(2, 3, 1)
survival_by_gender_class = df.groupby(['Sex', 'Pclass'])['Survived'].mean().unst
sns.heatmap(survival_by_gender_class, annot=True, fmt='.2f', cmap='RdYlGn')
plt.title('Survival Rate by Gender and Class')

# Age groups analysis
df['AgeGroup'] = pd.cut(df['Age'], bins=[0, 12, 18, 35, 60, 100],
                        labels=['Child', 'Teen', 'Adult', 'Middle-aged', 'Senior

plt.subplot(2, 3, 2)
sns.barplot(data=df, x='AgeGroup', y='Survived', ci=None)
plt.title('Survival Rate by Age Group')
plt.xticks(rotation=45)
```

```python
# Fare groups analysis
df['FareGroup'] = pd.cut(df['Fare'], bins=[0, 10, 25, 50, 100, 600],
                         labels=['Very Low', 'Low', 'Medium', 'High', 'Very High

plt.subplot(2, 3, 3)
sns.barplot(data=df, x='FareGroup', y='Survived', ci=None)
plt.title('Survival Rate by Fare Group')
plt.xticks(rotation=45)

# Gender + Age Group vs Survival
plt.subplot(2, 3, 4)
gender_age_survival = df.groupby(['Sex', 'AgeGroup'])['Survived'].mean().unstack
sns.heatmap(gender_age_survival, annot=True, fmt='.2f', cmap='RdYlGn')
plt.title('Survival Rate by Gender and Age Group')

# Class + Embarkation vs Survival
plt.subplot(2, 3, 5)
class_embark_survival = df.groupby(['Pclass', 'Embarked'])['Survived'].mean().un
sns.heatmap(class_embark_survival, annot=True, fmt='.2f', cmap='RdYlGn')
plt.title('Survival Rate by Class and Embarkation')

# Family Size + Class vs Survival
plt.subplot(2, 3, 6)
df['FamilySizeGroup'] = df['FamilySize'].apply(lambda x: 'Small' if x <= 2 else
family_class_survival = df.groupby(['FamilySizeGroup', 'Pclass'])['Survived'].me
sns.heatmap(family_class_survival, annot=True, fmt='.2f', cmap='RdYlGn')
plt.title('Survival Rate by Family Size and Class')

plt.tight_layout()
plt.show()
```

```
=== MULTI-DIMENSIONAL ANALYSIS ===
```



Step 6: Summary Statistics and Key Findings

In [17]:

```
print("="*60)
print("COMPREHENSIVE EDA SUMMARY - TITANIC DATASET")
print("="*60)

print("\n1. DATASET OVERVIEW:")
print(f"   - Total passengers: {len(df)}")
print(f"   - Features: {len(df.columns)}")
print(f"   - Overall survival rate: {df['Survived'].mean():.2%}")

print("\n2. MISSING DATA:")
missing_summary = df.isnull().sum()[df.isnull().sum() > 0]
for col, missing in missing_summary.items():
    print(f"   - {col}: {missing} ({missing/len(df)*100:.1f}%)")

print("\n3. KEY SURVIVAL FACTORS:")
print("   Gender Impact:")
female_survival = df[df['Sex'] == 'female']['Survived'].mean()
male_survival = df[df['Sex'] == 'male']['Survived'].mean()
print(f"   - Female survival rate: {female_survival:.2%}")
print(f"   - Male survival rate: {male_survival:.2%}")
print(f"   - Gender difference: {female_survival - male_survival:.2%}")

print("\n   Class Impact:")
for pclass in [1, 2, 3]:
    class_survival = df[df['Pclass'] == pclass]['Survived'].mean()
    print(f"   - Class {pclass} survival rate: {class_survival:.2%}")

print("\n   Age Impact:")
child_survival = df[df['Age'] <= 12]['Survived'].mean()
adult_survival = df[df['Age'] > 12]['Survived'].mean()
print(f"   - Children (≤12) survival rate: {child_survival:.2%}")
print(f"   - Adults (>12) survival rate: {adult_survival:.2%}")

print("\n   Family Size Impact:")
alone_survival = df[df['IsAlone'] == 1]['Survived'].mean()
family_survival = df[df['IsAlone'] == 0]['Survived'].mean()
print(f"   - Traveling alone survival rate: {alone_survival:.2%}")
print(f"   - Traveling with family survival rate: {family_survival:.2%}")

print("\n4. STATISTICAL INSIGHTS:")
print(f"   - Average age: {df['Age'].mean():.1f} years")
print(f"   - Average fare: ${df['Fare'].mean():.2f}")
print(f"   - Most common embarkation port: {df['Embarked'].mode().iloc[0]}")
print(f"   - Most common class: {df['Pclass'].mode().iloc[0]}")

print("\n5. CORRELATIONS WITH SURVIVAL:")
correlations = df[['Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare', 'Famil
for feature, corr in correlations.items():
    if feature != 'Survived' and corr > 0.1:
        print(f"   - {feature}: {corr:.3f}")
```

```
============================================================
COMPREHENSIVE EDA SUMMARY - TITANIC DATASET
============================================================

1. DATASET OVERVIEW:
   - Total passengers: 891
   - Features: 17
   - Overall survival rate: 38.38%

2. MISSING DATA:
   - Age: 177 (19.9%)
   - Cabin: 687 (77.1%)
   - Embarked: 2 (0.2%)
   - AgeGroup: 177 (19.9%)
   - FareGroup: 15 (1.7%)

3. KEY SURVIVAL FACTORS:
   Gender Impact:
   - Female survival rate: 74.20%
   - Male survival rate: 18.89%
   - Gender difference: 55.31%

   Class Impact:
   - Class 1 survival rate: 62.96%
   - Class 2 survival rate: 47.28%
   - Class 3 survival rate: 24.24%

   Age Impact:
   - Children (≤12) survival rate: 57.97%
   - Adults (>12) survival rate: 38.76%

   Family Size Impact:
   - Traveling alone survival rate: 30.35%
   - Traveling with family survival rate: 50.56%

4. STATISTICAL INSIGHTS:
   - Average age: 29.7 years
   - Average fare: $32.20
   - Most common embarkation port: S
   - Most common class: 3

5. CORRELATIONS WITH SURVIVAL:
   - Pclass: 0.338
   - Fare: 0.257
```

Step 7: Save our Work

In [18]:
```python
# Save the enhanced dataset with new features
df.to_csv('data/titanic_with_features.csv', index=False)
print("Enhanced dataset with new features saved as 'titanic_with_features.csv'")
```

```
Enhanced dataset with new features saved as 'titanic_with_features.csv'
```

COMPLETE OBSERVATIONS AND INSIGHTS:

1. Survival Overview:

- Overall survival rate was 38.4% (342 out of 891 passengers)

- This suggests the disaster had a high mortality rate

2. Gender Analysis:

- Females had a 74.2% survival rate vs males at 18.9%
- "Women and children first" evacuation protocol was clearly followed
- Gender was the strongest predictor of survival

3. Passenger Class Analysis:

- 1st class: 63.0% survival rate
- 2nd class: 47.3% survival rate
- 3rd class: 24.2% survival rate
- Clear class-based discrimination in rescue efforts

4. Age Analysis:

- Children (≤12) had higher survival rates than adults
- Average age of survivors was slightly lower than non-survivors
- Age showed moderate correlation with survival

5. Family Size Analysis:

- Passengers traveling alone had lower survival rates (30.4%) than those with family (50.6%)
- Optimal family size for survival was 2-4 members
- Very large families (>4) had reduced survival chances

6. Fare Analysis:

- Higher fare correlated with better survival chances
- Fare was strongly correlated with passenger class
- Reflects socioeconomic status impact on

7. Embarkation Analysis:

- Cherbourg (C) passengers had highest survival rate (55.4%)
- Southampton (S) passengers had lowest survival rate (33.7%)
- May reflect class distribution by embarkation port

8. Key Patterns:

- Intersection of gender and class: 1st class females had 96.8% survival rate
- 3rd class males had only 13.5% survival rate
- Children in higher classes had near-perfect survival rates

9. Missing Data Impact:

- Age missing for 19.9% of passengers - could affect analysis
- Cabin data missing for 77.1% - indicates incomplete records
- Embarked missing for only 0.2% - minimal impact\

10. Statistical Correlations:

- Strongest negative correlation: Pclass (-0.338)
- Strongest positive correlation: Fare (0.257)
- Gender coding would show strongest correlation if encoded numerically

This analysis reveals clear social hierarchies and evacuation protocols that determined survival on the Titanic, with gender, class, and age being the primary determining factors.