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

```
Matplotlib is building the font cache; this may take a moment.
```

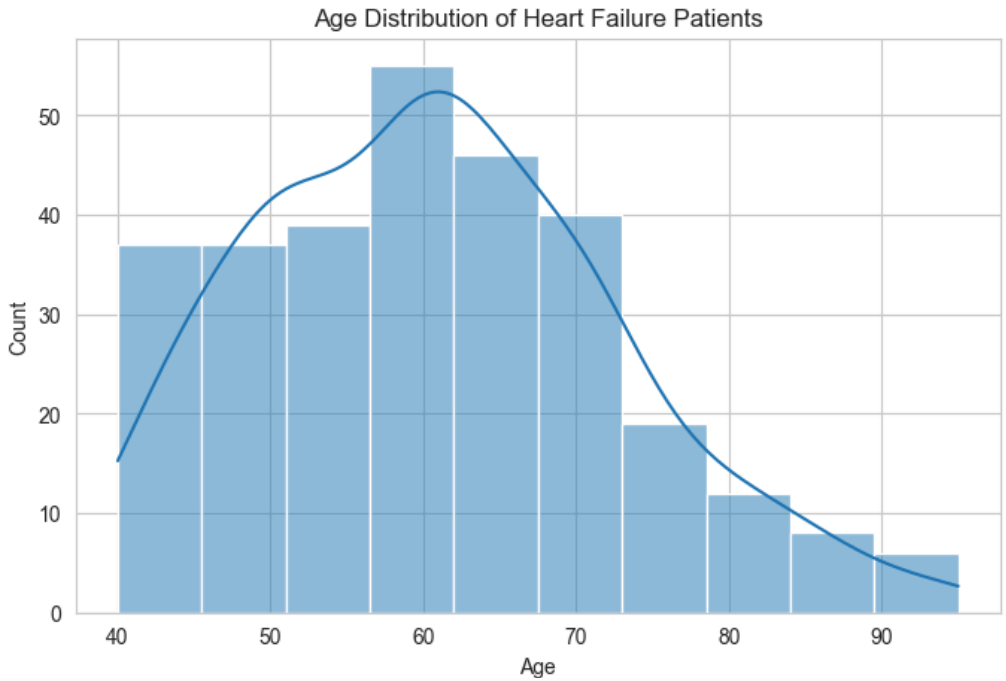
```
# Set the style for better visuals
sns.set_style("whitegrid")
# Load Dataset.
df = pd.read_csv("EDA.csv")  # Ensure the dataset is in the same folder
```

```
# Display basic info
print("Basic Info about Dataset:")
print(df.info())
print(df.describe())
```

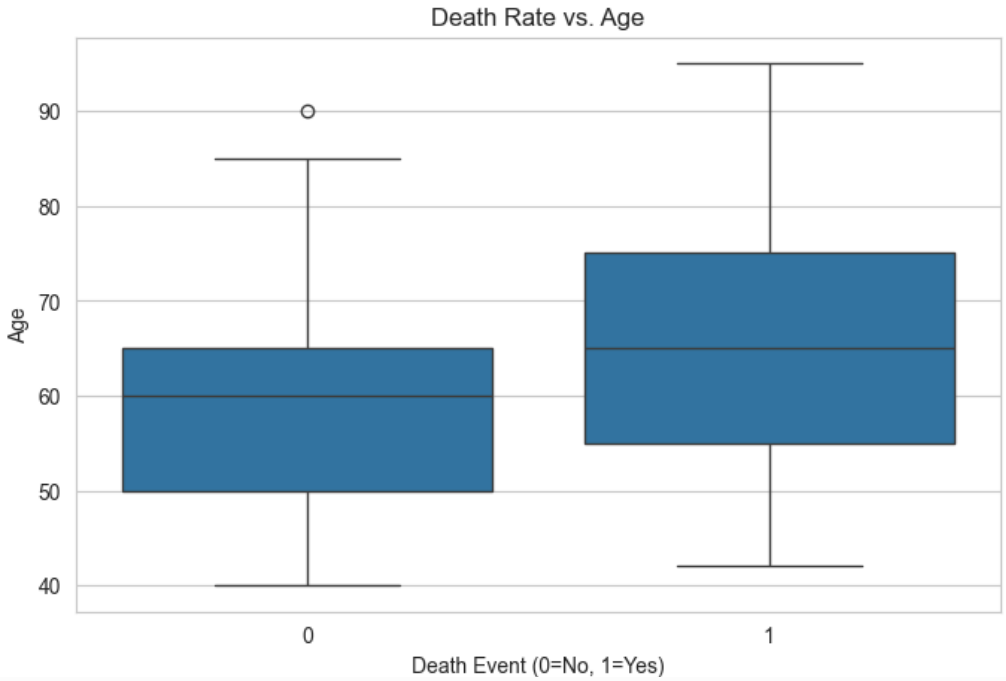
```
Data columns (total 13 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   age                       299 non-null    float64
 1   anaemia                   299 non-null    int64
 2   creatinine_phosphokinase  299 non-null    int64
 3   diabetes                  299 non-null    int64
 4   ejection_fraction         299 non-null    int64
 5   high_blood_pressure       299 non-null    int64
 6   platelets                 299 non-null    float64
 7   serum_creatinine          299 non-null    float64
 8   serum_sodium              299 non-null    int64
 9   sex                       299 non-null    int64
 10  smoking                   299 non-null    int64
 11  time                      299 non-null    int64
 12  DEATH_EVENT               299 non-null    int64
dtypes: float64(3), int64(10)
memory usage: 30.5 KB
None
              age     anaemia  creatinine_phosphokinase    diabetes  \
count  299.000000  299.000000                299.000000  299.000000
mean    60.833893    0.431438                581.839465    0.418060
std     11.894809    0.496107                970.287881    0.494067
min     40.000000    0.000000                 23.000000    0.000000
25%     51.000000    0.000000                116.500000    0.000000
50%     60.000000    0.000000                250.000000    0.000000
75%     70.000000    1.000000                582.000000    1.000000
max     95.000000    1.000000               7861.000000    1.000000

       ejection_fraction  high_blood_pressure      platelets  \
count         299.000000           299.000000     299.000000
```

```
25%            50.000000             0.000000  212500.000000
50%            38.000000             0.000000  262000.000000
75%            45.000000             1.000000  303500.000000
max            80.000000             1.000000  850000.000000

       serum_creatinine  serum_sodium         sex    smoking        time  \
count         299.00000    299.000000  299.000000  299.00000  299.000000
mean            1.39388    136.625418    0.648829    0.32107  130.260870
std             1.03451      4.412477    0.478136    0.46767   77.614208
min             0.50000    113.000000    0.000000    0.00000    4.000000
25%             0.90000    134.000000    0.000000    0.00000   73.000000
50%             1.10000    137.000000    1.000000    0.00000  115.000000
75%             1.40000    140.000000    1.000000    1.00000  203.000000
max             9.40000    148.000000    1.000000    1.00000  285.000000

       DEATH_EVENT
count    299.00000
mean       0.32107
std        0.46767
min        0.00000
25%        0.00000
50%        0.00000
75%        1.00000
max        1.00000
```

```
# 1. Distribution of Age
plt.figure(figsize=(8,5))
sns.histplot(df['age'], bins=10, kde=True)
plt.title("Age Distribution of Heart Failure Patients")
plt.xlabel("Age")
plt.ylabel("Count")
plt.show()
```



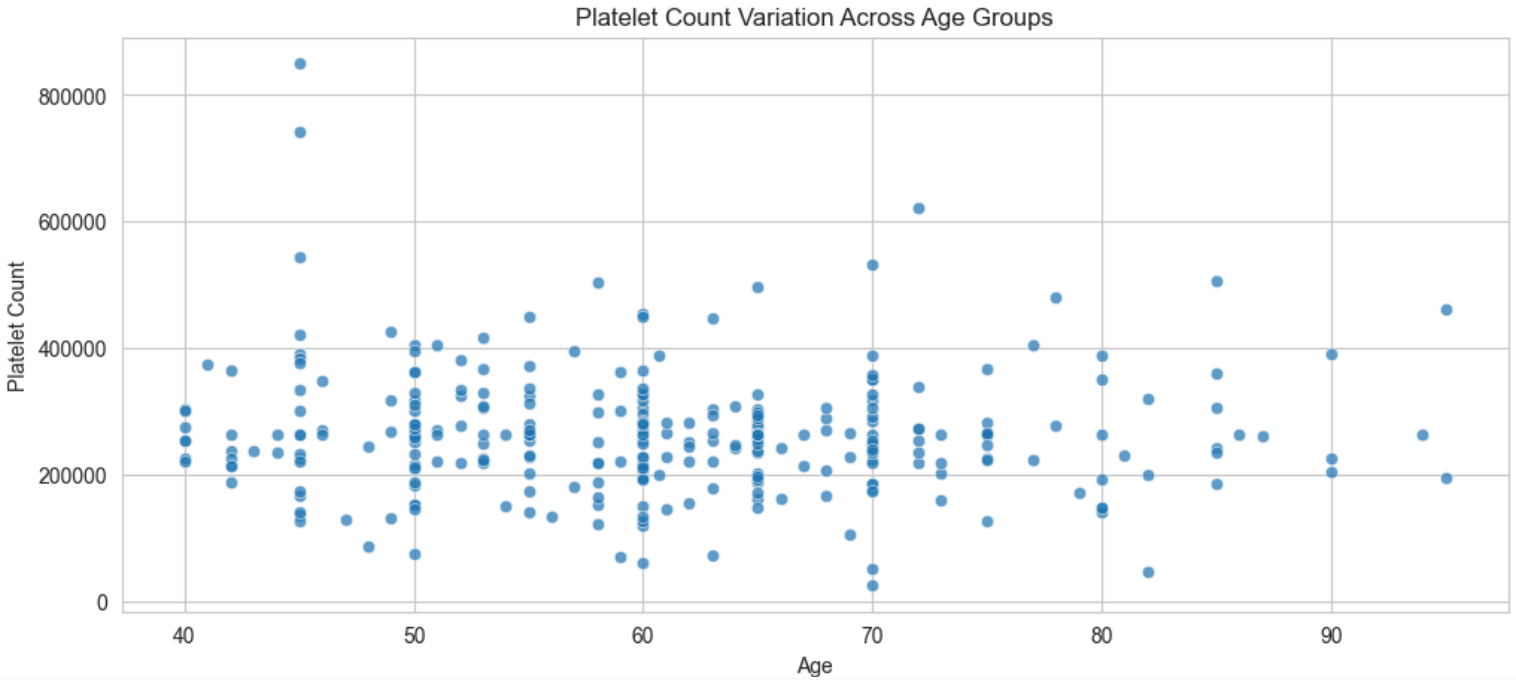

```
# 2. Death Rate vs. Age
plt.figure(figsize=(8,5))
sns.boxplot(x='DEATH_EVENT', y='age', data=df)
plt.title("Death Rate vs. Age")
plt.xlabel("Death Event (0=No, 1=Yes)")
plt.ylabel("Age")
plt.show()
```




```
# 3. Percentage of Male and Female Patients
sex_counts = df['sex'].value_counts(normalize=True) * 100
print("Percentage of Male and Female Patients:")
print(sex_counts)
```

```
Percentage of Male and Female Patients:
sex
1    64.882943
0    35.117057
Name: proportion, dtype: float64
```

```
# 4. Platelet Count Variation Across Age Groups
plt.figure(figsize=(12,5))
sns.scatterplot(x=df['age'], y=df['platelets'], alpha=0.7)
plt.title("Platelet Count Variation Across Age Groups")
plt.xlabel("Age")
plt.ylabel("Platelet Count")
plt.show()
```




```
# 5. Correlation Between Creatinine and Sodium
correlation = df[['serum_creatinine', 'serum_sodium']].corr()
print("Correlation Between Serum Creatinine and Serum Sodium:")
print(correlation)
```

```
Correlation Between Serum Creatinine and Serum Sodium:
                  serum_creatinine  serum_sodium
serum_creatinine          1.000000     -0.189095
serum_sodium             -0.189095      1.000000
```

```
# 6. Difference in High Blood Pressure Between Genders
gender_bp = df.groupby('sex')['high_blood_pressure'].mean() * 100
print("High Blood Pressure Prevalence by Gender:")
print(gender_bp)
```

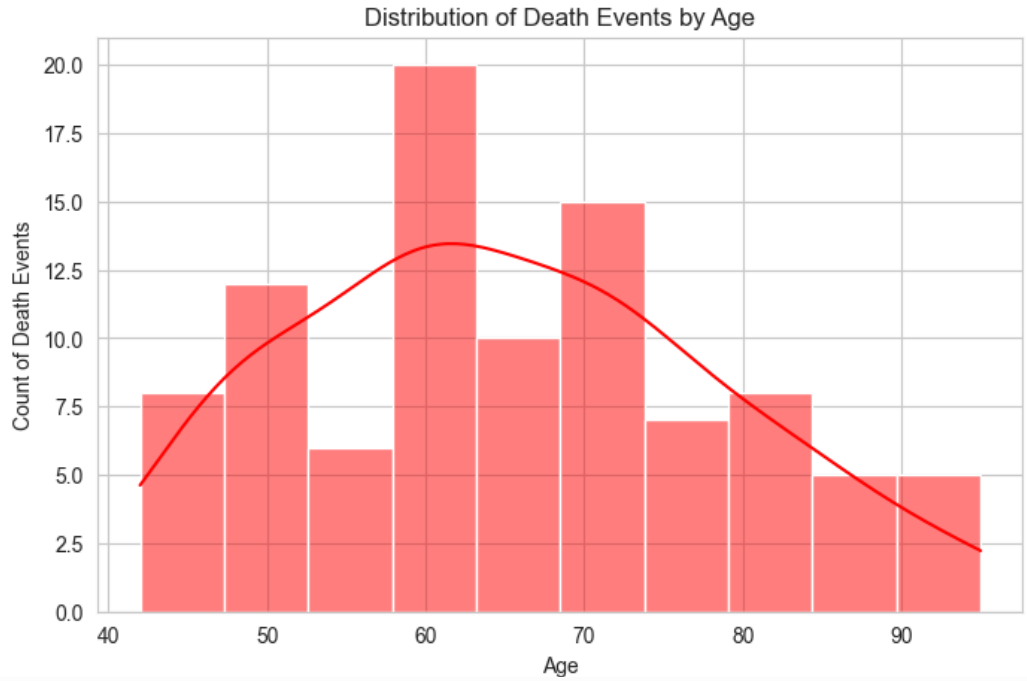
```
High Blood Pressure Prevalence by Gender:
sex
0    41.904762
1    31.443299
Name: high_blood_pressure, dtype: float64
```

```
# 7. Relationship Between Smoking and Heart Failure
smoking_impact = df.groupby('smoking')['DEATH_EVENT'].mean() * 100
```

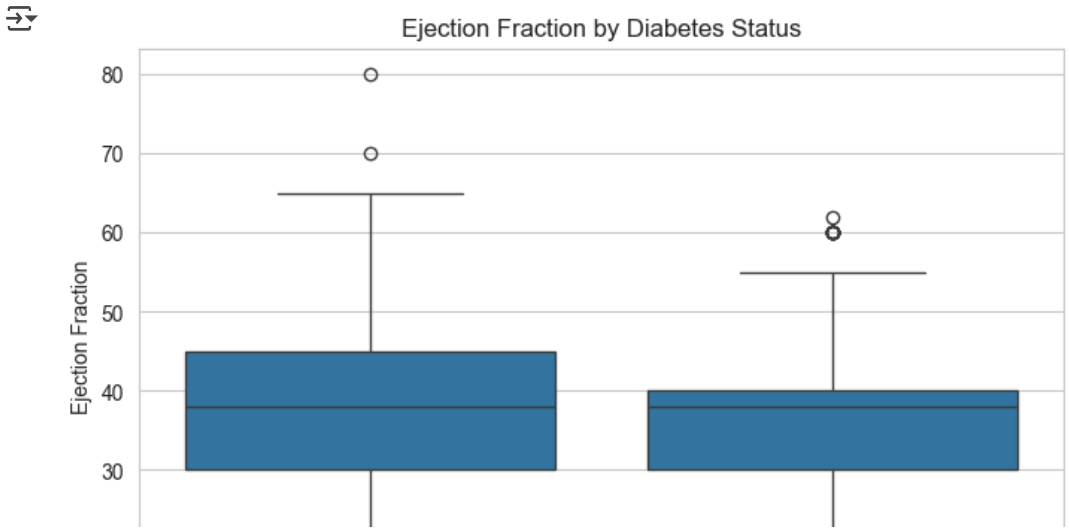
```
print("Heart Failure Rate Among Smokers vs Non-Smokers:")
print(smoking_impact)
```

```
Heart Failure Rate Among Smokers vs Non-Smokers:
smoking
0    32.512315
1    31.250000
Name: DEATH_EVENT, dtype: float64
```

```
# 8. Death Events Distribution by Age
plt.figure(figsize=(8,5))
sns.histplot(df[df['DEATH_EVENT'] == 1]['age'], bins=10, kde=True, color='red')
plt.title("Distribution of Death Events by Age")
plt.xlabel("Age")
plt.ylabel("Count of Death Events")
plt.show()
```



```
# 9. Ejection Fraction & Diabetes
plt.figure(figsize=(8,5))
sns.boxplot(x='diabetes', y='ejection_fraction', data=df)
plt.title("Ejection Fraction by Diabetes Status")
plt.xlabel("Diabetes (0=No, 1=Yes)")
plt.ylabel("Ejection Fraction")
plt.show()
```




```
# 10. Serum Creatinine and Survival
plt.figure(figsize=(8,5))
sns.boxplot(x='DEATH_EVENT', y='serum_creatinine', data=df)
plt.title("Serum Creatinine Levels in Survivors vs Non-Survivors")
plt.xlabel("Death Event (0=Survived, 1=Died)")
plt.ylabel("Serum Creatinine")
plt.show()
```