

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

In [5]:

```
df=pd.read_csv("D:\\Drive C\\Downloads\\heart_failure_clinical_records_dataset (1).csv")
```

In [7]:

```
df
```

Out[7]:

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
0	75.0	0	582	0	20	1	265000.00	1.9	130	1	0	4	1
1	55.0	0	7861	0	38	0	263358.03	1.1	136	1	0	6	1
2	65.0	0	146	0	20	0	162000.00	1.3	129	1	1	7	1
3	50.0	1	111	0	20	0	210000.00	1.9	137	1	0	7	1
4	65.0	1	160	1	20	0	327000.00	2.7	116	0	0	8	1
...
294	62.0	0	61	1	38	1	155000.00	1.1	143	1	1	270	0
295	55.0	0	1820	0	38	0	270000.00	1.2	139	0	0	271	0
296	45.0	0	2060	1	60	0	742000.00	0.8	138	0	0	278	0
297	48.0	0	2413	0	38	0	140000.00	1.4	140	1	1	280	0
298	50.0	0	196	0	45	0	395000.00	1.6	136	1	1	285	0

299 rows x 13 columns

In [9]:

```
#!/usr/bin/env python
import pandas as pd
import matplotlib.pyplot as plt
```

In [11]:

```
df=pd.read_csv("D:\\Drive C\\Downloads\\heart_failure_clinical_records_dataset (1).csv")
```

In [13]:

```
df
```

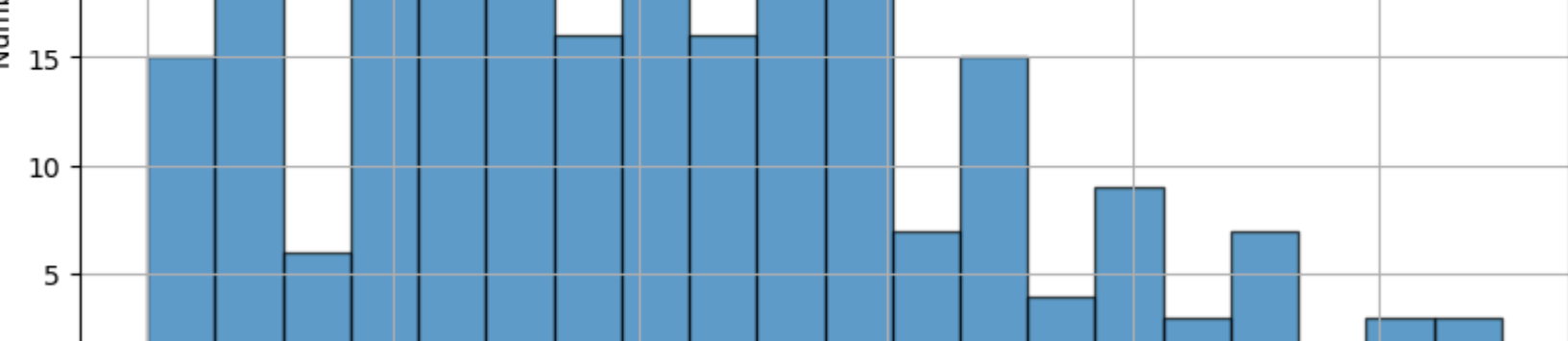
Out[13]:

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
0	75.0	0	582	0	20	1	265000.00	1.9	130	1	0	4	1
1	55.0	0	7861	0	38	0	263358.03	1.1	136	1	0	6	1
2	65.0	0	146	0	20	0	162000.00	1.3	129	1	1	7	1
3	50.0	1	111	0	20	0	210000.00	1.9	137	1	0	7	1
4	65.0	1	160	1	20	0	327000.00	2.7	116	0	0	8	1
...
294	62.0	0	61	1	38	1	155000.00	1.1	143	1	1	270	0
295	55.0	0	1820	0	38	0	270000.00	1.2	139	0	0	271	0
296	45.0	0	2060	1	60	0	742000.00	0.8	138	0	0	278	0
297	48.0	0	2413	0	38	0	140000.00	1.4	140	1	1	280	0
298	50.0	0	196	0	45	0	395000.00	1.6	136	1	1	285	0

299 rows x 13 columns

In [15]:

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("D:\\Drive C\\Downloads\\heart_failure_clinical_records_dataset (1).csv")
plt.figure(figsize=(10, 6))
plt.hist(df['age'], bins=20, edgecolor='k', alpha=0.7)
plt.title('Distribution of Age Among Heart Failure Patients')
plt.xlabel('Age')
plt.ylabel('Number of Patients')
plt.grid(True)
plt.show()
```



In [17]:

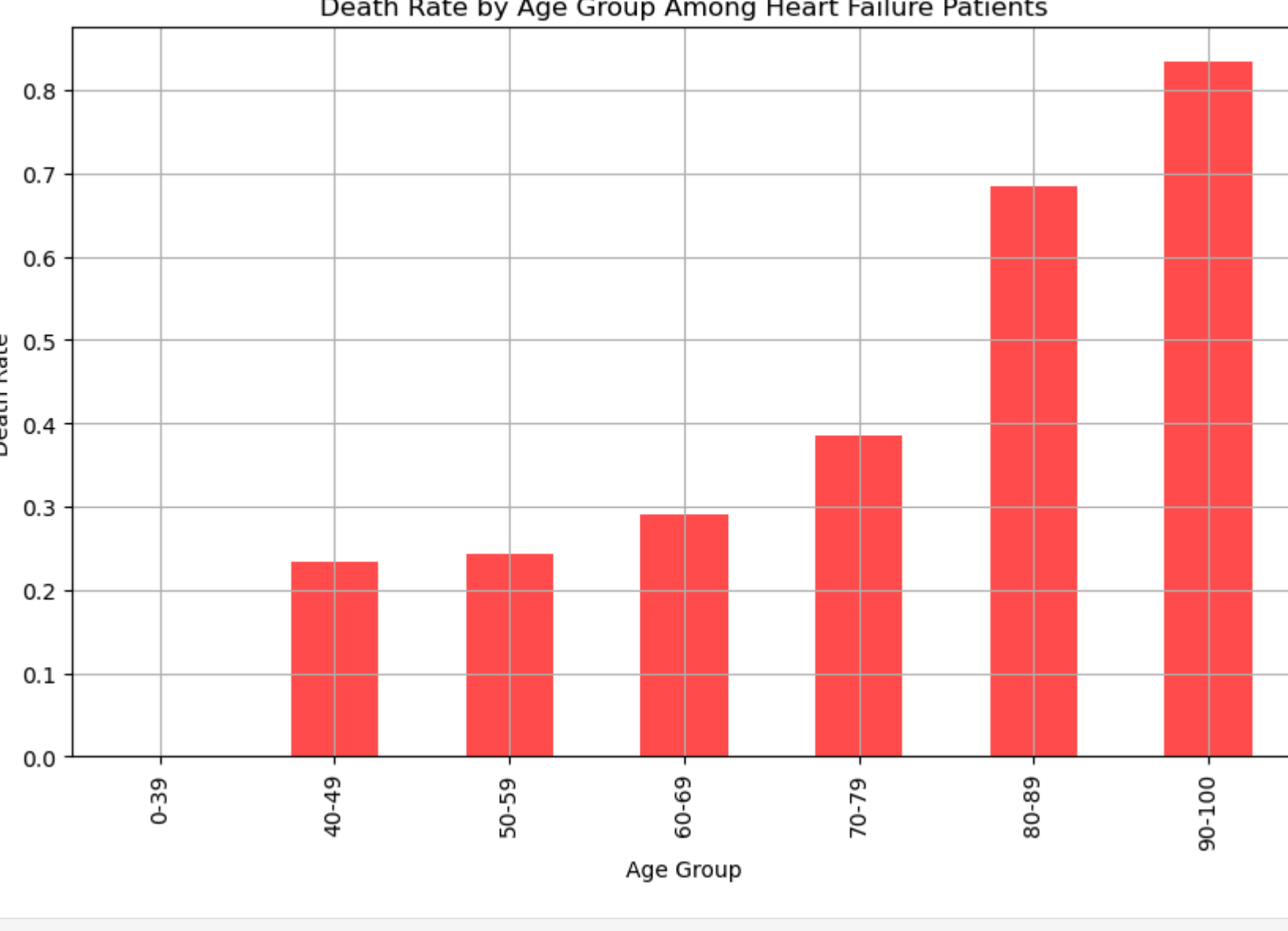
```
#!/usr/bin/env python
```

In [19]:

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("D:\\Drive C\\Downloads\\heart_failure_clinical_records_dataset (1).csv")
age_bins = [0, 40, 50, 60, 70, 80, 90, 100]
age_labels = ['0-39', '40-49', '50-59', '60-69', '70-79', '80-89', '90-100']
df['age_group'] = pd.cut(df['age'], bins=age_bins, labels=age_labels, right=False)
death_rates = df.groupby('age_group')['DEATH_EVENT'].mean()
plt.figure(figsize=(10, 6))
death_rates.plot(kind='bar', color='red', alpha=0.7)
plt.title('Death Rate by Age Group Among Heart Failure Patients')
plt.xlabel('Age Group')
plt.ylabel('Death Rate')
plt.grid(True)
plt.show()
```

C:\Users\Pratishtha\Anaconda\AgData\Local\Temp\ipykernel_14240\2849288253.py:17: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
death_rates = df.groupby('age_group')['DEATH_EVENT'].mean()
```



In [21]:

```
#!/usr/bin/env python
```

In [23]:

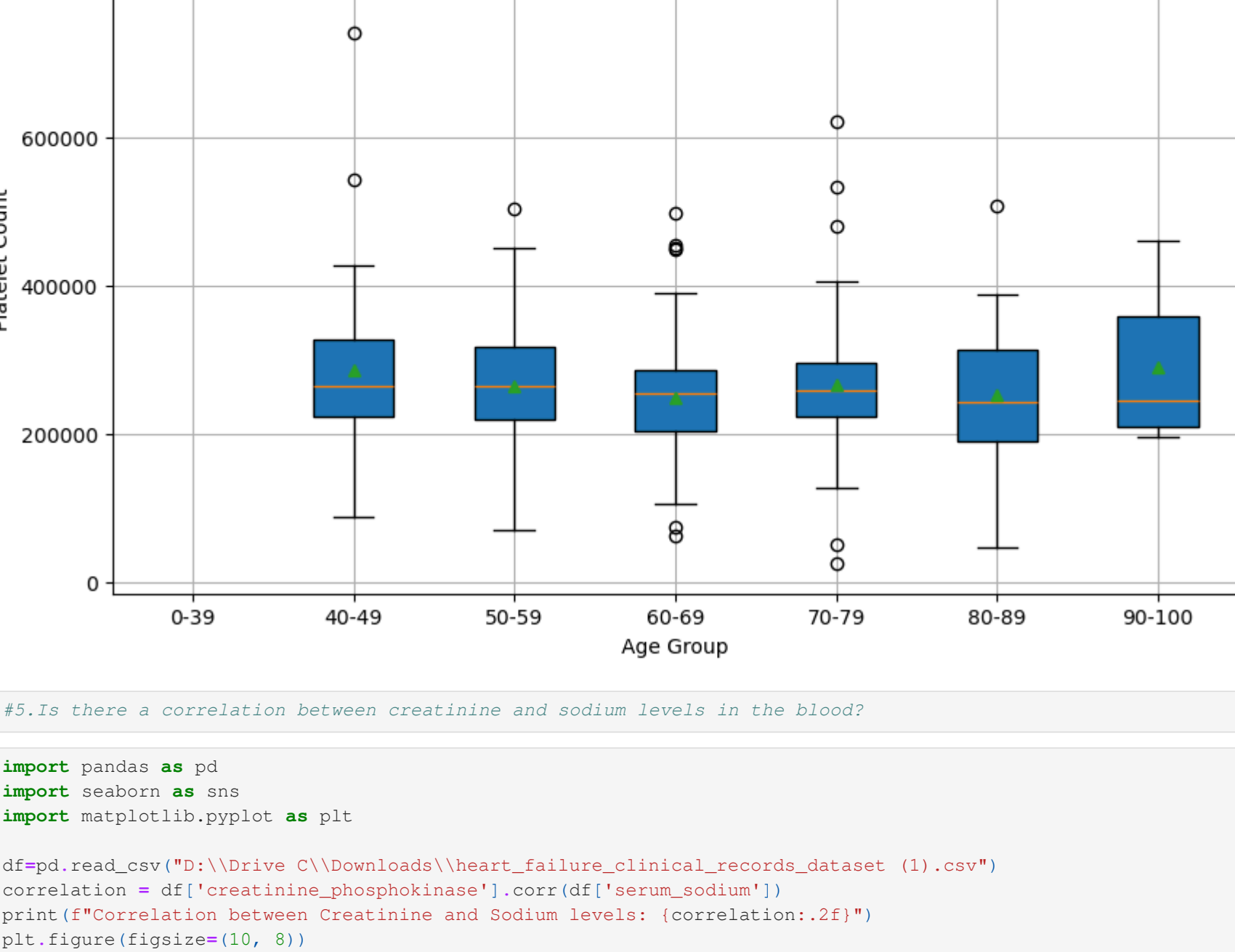
```
import pandas as pd
import numpy as np
df=pd.read_csv("D:\\Drive C\\Downloads\\heart_failure_clinical_records_dataset (1).csv")
gender_counts = df['sex'].value_counts()
total_patients=len(df)
percentage_male = (gender_counts[1]/total_patients) * 100
percentage_female = (gender_counts[0]/total_patients) * 100
print(f"Percentage of male patients: {percentage_male:.2f}%")
print(f"Percentage of female patients: {percentage_female:.2f}%")
percentage of male patients: 64.88%
percentage of female patients: 35.12%
```

In [25]:

```
#!/usr/bin/env python
```

In [27]:

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("D:\\Drive C\\Downloads\\heart_failure_clinical_records_dataset (1).csv")
age_bins = [0, 40, 50, 60, 70, 80, 90, 100]
age_labels = ['0-39', '40-49', '50-59', '60-69', '70-79', '80-89', '90-100']
df['age_group'] = pd.cut(df['age'], bins=age_bins, labels=age_labels, right=False)
plt.figure(figsize=(10, 6))
plt.boxplot(df['platelets'], by=age_group, labels=age_labels, patch_artist=True, showmeans=True)
plt.xlabel('Age Group')
plt.ylabel('Platelet Count')
plt.grid(True)
plt.show()
```



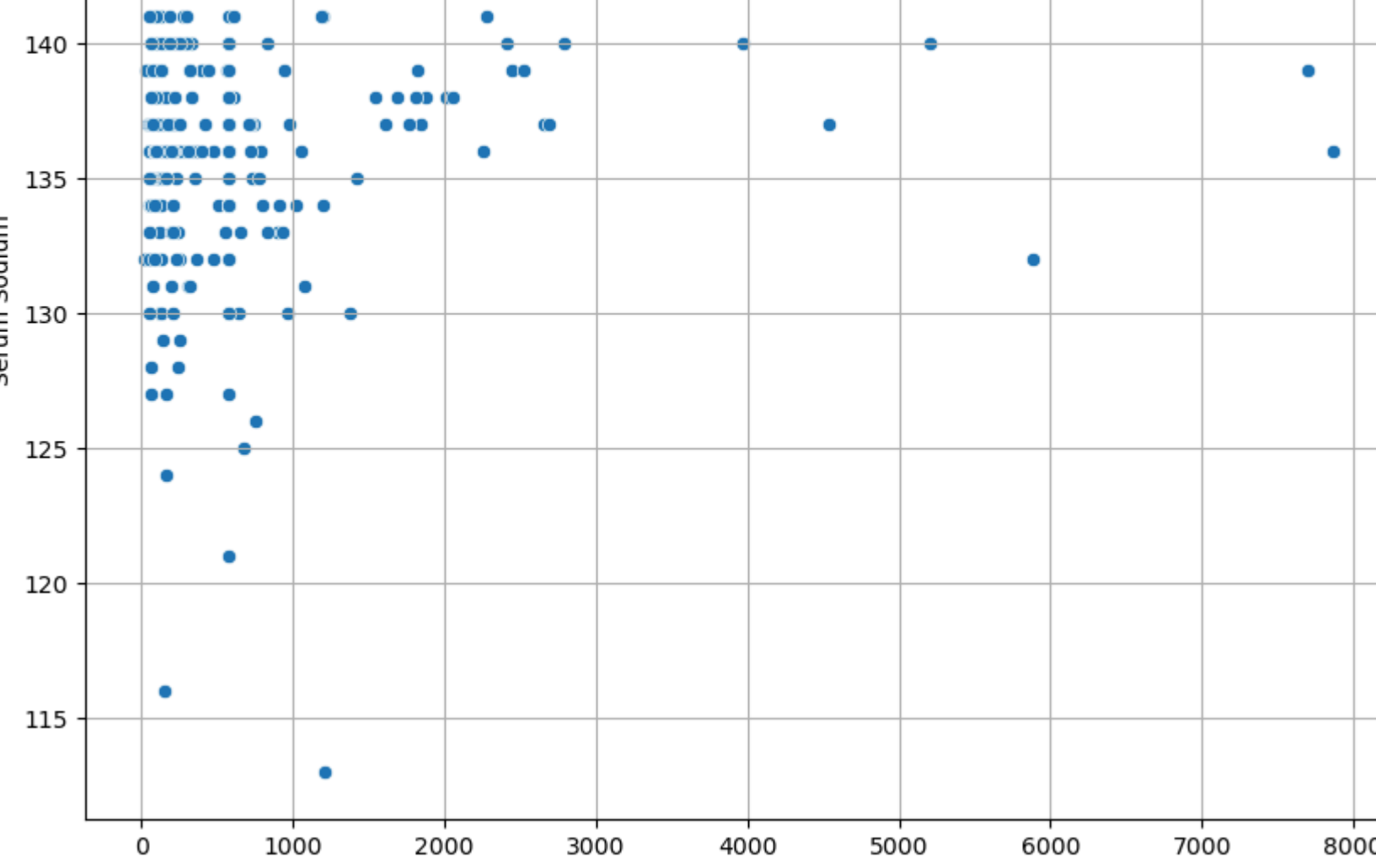
In [29]:

```
#!/usr/bin/env python
```

In [31]:

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("D:\\Drive C\\Downloads\\heart_failure_clinical_records_dataset (1).csv")
correlation = df[['creatinine_phosphokinase', 'serum_sodium']].corr()
print(f"Correlation between Creatinine and Sodium Levels: {correlation['creatinine_phosphokinase', 'serum_sodium']}")
plt.figure(figsize=(10, 8))
sns.scatterplot(x='creatinine_phosphokinase', y='serum_sodium', data=df)
plt.title('Correlation Between Creatinine and Sodium Levels')
plt.xlabel('Creatinine Phosphokinase')
plt.ylabel('Serum Sodium')
plt.grid(True)
plt.show()
```

Correlation between Creatinine and Sodium Levels: 0.06



In [33]:

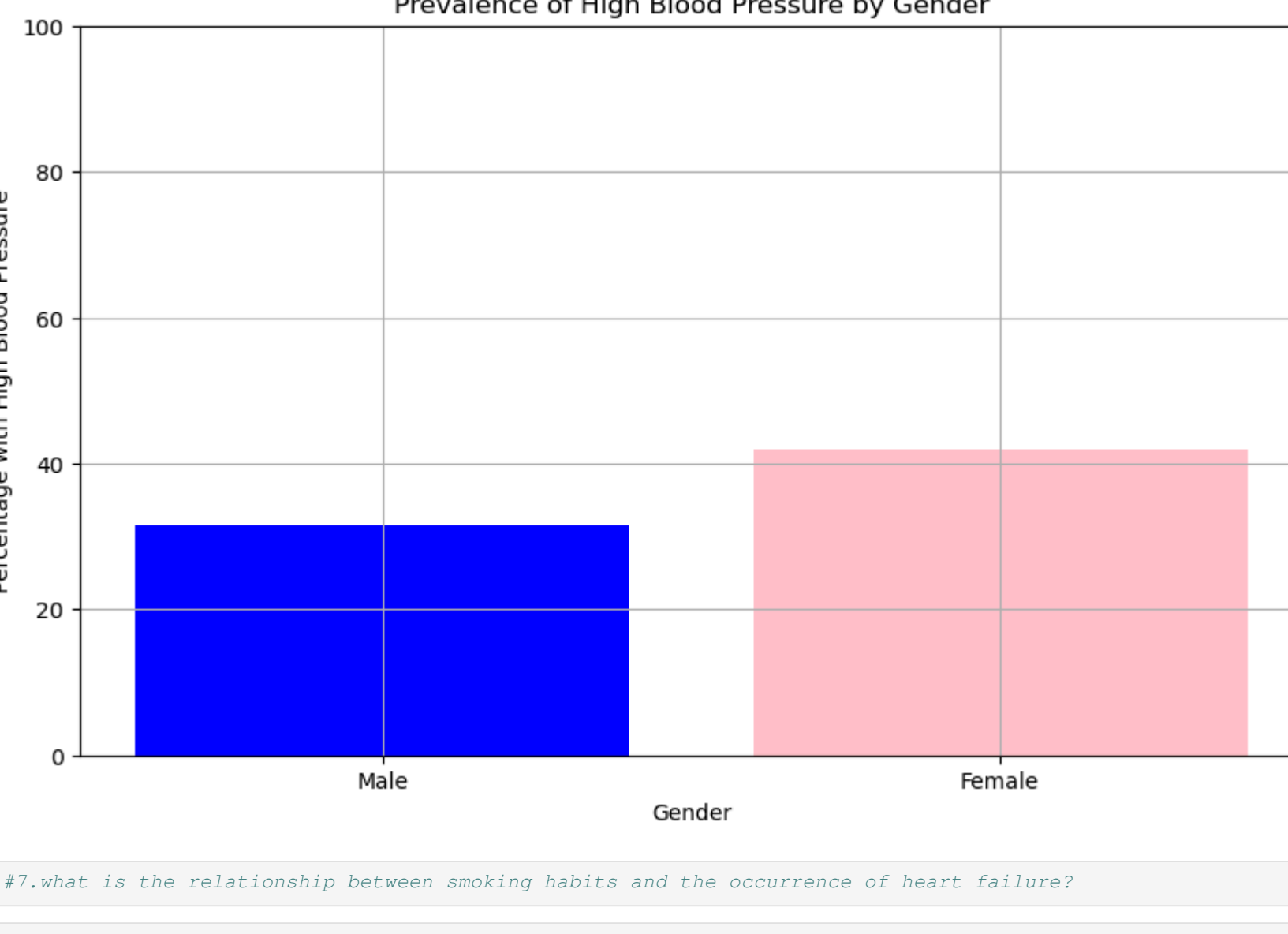
```
#!/usr/bin/env python
```

In [35]:

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("D:\\Drive C\\Downloads\\heart_failure_clinical_records_dataset (1).csv")
gender_counts = df['sex'].value_counts()
high_bp_counts = df.groupby('sex')['high_blood_pressure'].sum()
percentage_male_high_bp = (high_bp_counts[1] / gender_counts[1]) * 100
percentage_female_high_bp = (high_bp_counts[0] / gender_counts[0]) * 100
print(f"Percentage of male patients with high blood pressure: {percentage_male_high_bp:.2f}%")
print(f"Percentage of female patients with high blood pressure: {percentage_female_high_bp:.2f}%")
plt.bar(['Male', 'Female'], [percentage_male_high_bp, percentage_female_high_bp], color=['blue', 'pink'])
plt.title('Prevalence of High Blood Pressure by Gender')
plt.xlabel('Gender')
plt.ylabel('Percentage with High Blood Pressure')
plt.grid(True)
plt.show()
```

Percentage of male patients with high blood pressure: 31.44%

Percentage of female patients with high blood pressure: 41.90%



In [37]:

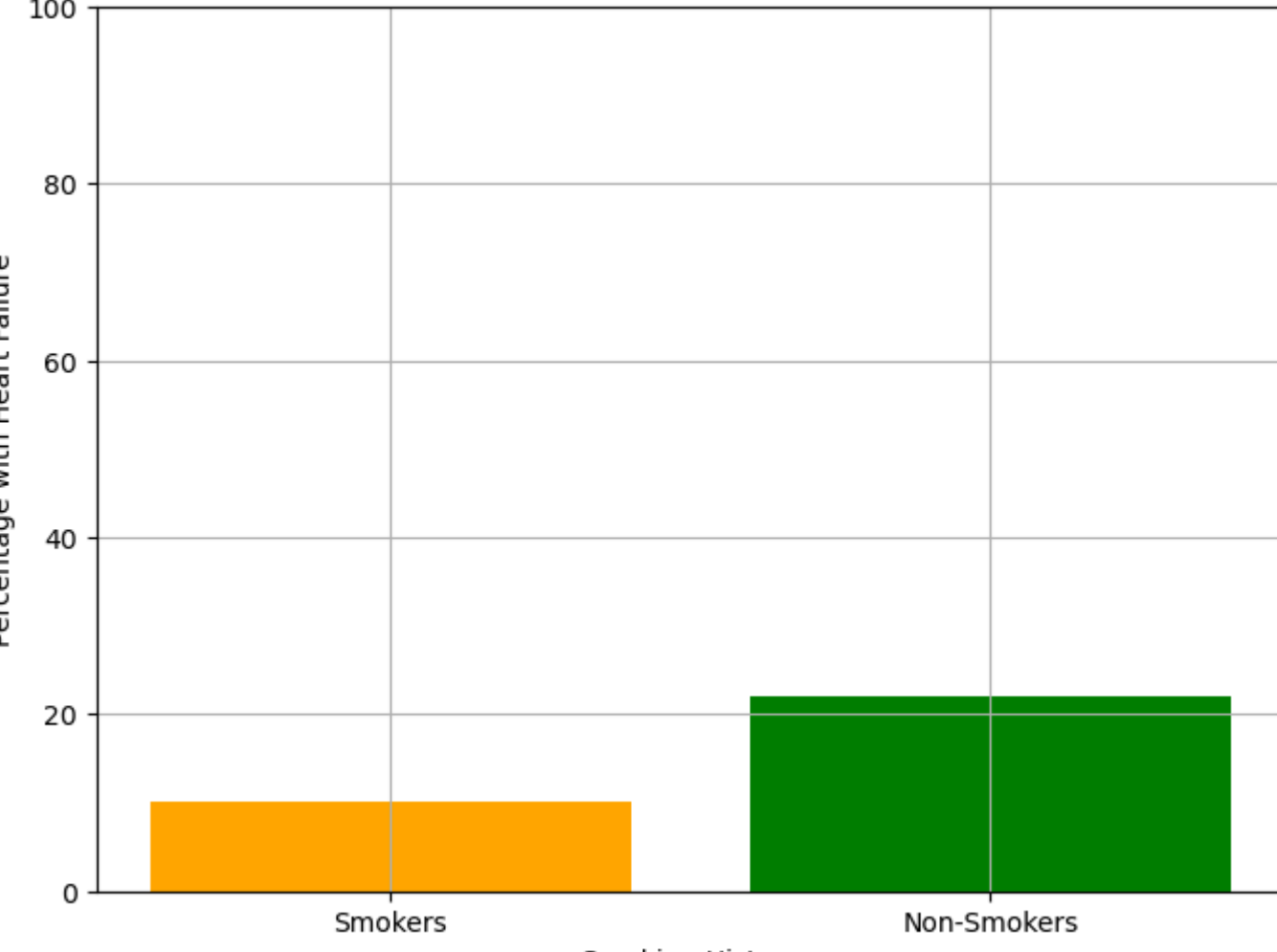
```
#!/usr/bin/env python
```

In [39]:

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("D:\\Drive C\\Downloads\\heart_failure_clinical_records_dataset (1).csv")
total_patients = len(df)
smokers_with_heart_failure = len(df[df['smoking'] == 1])
non_smokers_with_heart_failure = len(df[df['smoking'] == 0])
percentage_smokers_with_heart_failure = (smokers_with_heart_failure / total_patients) * 100
percentage_non_smokers_with_heart_failure = (non_smokers_with_heart_failure / total_patients) * 100
print(f"Percentage of smokers who had heart failure: {percentage_smokers_with_heart_failure:.2f}%")
print(f"Percentage of non-smokers who had heart failure: {percentage_non_smokers_with_heart_failure:.2f}%")
plt.bar(['Smokers', 'Non-Smokers'], [percentage_smokers_with_heart_failure, percentage_non_smokers_with_heart_failure], color=['orange', 'green'])
plt.title('Occurrence of Heart Failure by Smoking History')
plt.xlabel('Smoking History')
plt.ylabel('Percentage with Heart Failure')
plt.grid(True)
plt.show()
```

Percentage of smokers who had heart failure: 10.03%

Percentage of non-smokers who had heart failure: 22.07%



In [41]:

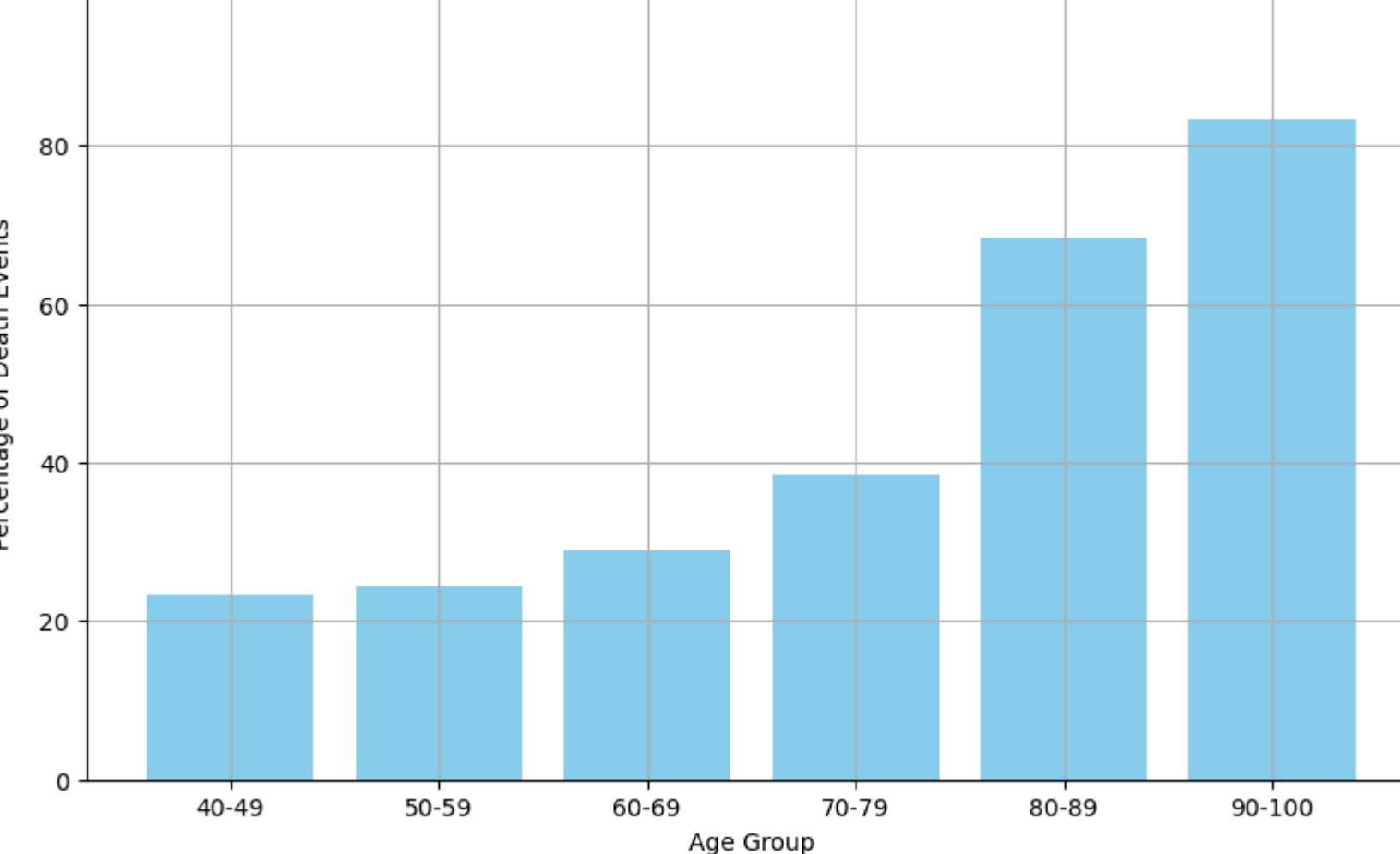
```
#!/usr/bin/env python
```

In [43]:

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("D:\\Drive C\\Downloads\\heart_failure_clinical_records_dataset (1).csv")
age_bins = [0, 40, 50, 60, 70, 80, 90, 100]
age_labels = ['0-39', '40-49', '50-59', '60-69', '70-79', '80-89', '90-100']
df['age_group'] = pd.cut(df['age'], bins=age_bins, labels=age_labels, right=False)
death_event_percentages = df.groupby('age_group')['DEATH_EVENT'].mean() * 100
plt.figure(figsize=(10, 6))
plt.bar(age_labels, death_event_percentages, color='skyblue')
plt.title('Distribution of Death Events Across Age Groups')
plt.xlabel('Age Group')
plt.ylabel('Percentage of Death Events')
plt.grid(True)
plt.show()
```

C:\Users\Pratishtha\Anaconda\AgData\Local\Temp\ipykernel_14240\313049368.py:77: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
death_event_percentages = df.groupby('age_group')['DEATH_EVENT'].mean() * 100
```



In [45]:

```
#!/usr/bin/env python
```

In [47]:

```
import pandas as pd
import scipy.stats as stats
df=pd.read_csv("D:\\Drive C\\Downloads\\heart_failure_clinical_records_dataset (1).csv")
ef_diabetes = df[df['diabetes'] == 1]['ejection_fraction']
ef_no_diabetes = df[df['diabetes'] == 0]['ejection_fraction']
mean_ef_diabetes = ef_diabetes.mean()
mean_ef_no_diabetes = ef_no_diabetes.mean()
std_ef_diabetes = ef_diabetes.std()
std_ef_no_diabetes = ef_no_diabetes.std()
t_statistic, p_value = stats.ttest_ind(ef_diabetes, ef_no_diabetes, equal_var=False)
print(f"Mean ejection fraction for patients with diabetes: {mean_ef_diabetes:.2f}")
print(f"Mean ejection fraction for patients without diabetes: {mean_ef_no_diabetes:.2f}")
print(f"P-value from t-test: {p_value:.4f}")
alpha = 0.05
if p_value < alpha:
    print("There is a significant difference in ejection fraction between patients with and without diabetes.")
else:
    print("There is no significant difference in ejection fraction between patients with and without diabetes.")
```

Mean ejection fraction for patients with diabetes: 38.02

Mean ejection fraction for patients without diabetes: 34.13

P-value from t-test: 0.9325

There is no significant difference in ejection fraction between patients with and without diabetes.

In [49]:

```
#!/usr/bin/env python
```

In [51]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df=pd.read_csv("D:\\Drive C\\Downloads\\heart_failure_clinical_records_dataset (1).csv")
survived = df[df['DEATH_EVENT'] == 0]['serum_creatinine']
non_survived = df[df['DEATH_EVENT'] == 1]['serum_creatinine']
plt.figure(figsize=(10, 6))
sns.kdeplot(survived, bins=20, kde=True, color='blue', label='Survived')
sns.kdeplot(non_survived, bins=20, kde=True, color='red', label='Not Survived')
plt.title('Distribution of Serum Creatinine Levels by Survival Status')
plt.xlabel('Serum Creatinine Level')
plt.ylabel('Frequency')
plt.legend()
plt.grid(True)
plt.show()
```

