eda-1

September 14, 2024

[2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

[5]:
```
df=pd.read_csv("./heart_failure_clinical_records_dataset.csv")
```

[6]:
```
df
```

[6]:

	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	45.0	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

```
1           0     6            1
2           1     7            1
3           0     7            1
4           0     8            1
..        …   …          …
294         1   270            0
295         0   271            0
296         0   278            0
297         1   280            0
298         1   285            0

[299 rows x 13 columns]
```

[7]:
```
df.head()
```

[7]:
```
    age  anaemia  creatinine_phosphokinase  diabetes  ejection_fraction  \
0  75.0        0                       582         0                 20
1  55.0        0                      7861         0                 38
2  65.0        0                       146         0                 20
3  50.0        1                       111         0                 20
4  65.0        1                       160         1                 20

   high_blood_pressure  platelets  serum_creatinine  serum_sodium  sex  \
0                    1  265000.00               1.9           130    1
1                    0  263358.03               1.1           136    1
2                    0  162000.00               1.3           129    1
3                    0  210000.00               1.9           137    1
4                    0  327000.00               2.7           116    0

   smoking  time  DEATH_EVENT
0        0     4            1
1        0     6            1
2        1     7            1
3        0     7            1
4        0     8            1
```

[9]:
```
df.count()
```

[9]:
```
age                         299
anaemia                     299
creatinine_phosphokinase    299
diabetes                    299
ejection_fraction           299
high_blood_pressure         299
platelets                   299
serum_creatinine            299
serum_sodium                299
```

```
sex                       299
smoking                   299
time                      299
DEATH_EVENT               299
dtype: int64
```

[11]:
```
! pip install -U ydata-profiling
```

```
Requirement already satisfied: ydata-profiling in
/usr/local/lib/python3.10/dist-packages (4.8.3)
Requirement already satisfied: scipy<1.14,>=1.4.1 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (1.11.4)
Requirement already satisfied: pandas!=1.4.0,<3,>1.1 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (2.0.3)
Requirement already satisfied: matplotlib<3.9,>=3.2 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (3.7.1)
Requirement already satisfied: pydantic>=2 in /usr/local/lib/python3.10/dist-
packages (from ydata-profiling) (2.7.1)
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (6.0.1)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (3.1.4)
Requirement already satisfied: visions[type_image_path]<0.7.7,>=0.7.5 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (0.7.6)
Requirement already satisfied: numpy<2,>=1.16.0 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (1.25.2)
Requirement already satisfied: htmlmin==0.1.12 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (0.1.12)
Requirement already satisfied: phik<0.13,>=0.11.1 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (0.12.4)
Requirement already satisfied: requests<3,>=2.24.0 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (2.31.0)
Requirement already satisfied: tqdm<5,>=4.48.2 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (4.66.4)
Requirement already satisfied: seaborn<0.14,>=0.10.1 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (0.13.1)
Requirement already satisfied: multimethod<2,>=1.4 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (1.11.2)
Requirement already satisfied: statsmodels<1,>=0.13.2 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (0.14.2)
Requirement already satisfied: typeguard<5,>=3 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (4.3.0)
Requirement already satisfied: imagehash==4.3.1 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (4.3.1)
Requirement already satisfied: wordcloud>=1.9.1 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (1.9.3)
Requirement already satisfied: dacite>=1.8 in /usr/local/lib/python3.10/dist-
```

```
packages (from ydata-profiling) (1.8.1)
Requirement already satisfied: numba<1,>=0.56.0 in
/usr/local/lib/python3.10/dist-packages (from ydata-profiling) (0.58.1)
Requirement already satisfied: PyWavelets in /usr/local/lib/python3.10/dist-
packages (from imagehash==4.3.1->ydata-profiling) (1.6.0)
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages
(from imagehash==4.3.1->ydata-profiling) (9.4.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.10/dist-packages (from jinja2<3.2,>=2.11.1->ydata-
profiling) (2.1.5)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib<3.9,>=3.2->ydata-
profiling) (1.2.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-
packages (from matplotlib<3.9,>=3.2->ydata-profiling) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib<3.9,>=3.2->ydata-
profiling) (4.51.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib<3.9,>=3.2->ydata-
profiling) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib<3.9,>=3.2->ydata-
profiling) (24.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib<3.9,>=3.2->ydata-
profiling) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.10/dist-packages (from matplotlib<3.9,>=3.2->ydata-
profiling) (2.8.2)
Requirement already satisfied: llvmlite<0.42,>=0.41.0dev0 in
/usr/local/lib/python3.10/dist-packages (from numba<1,>=0.56.0->ydata-profiling)
(0.41.1)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-
packages (from pandas!=1.4.0,<3,>1.1->ydata-profiling) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-
packages (from pandas!=1.4.0,<3,>1.1->ydata-profiling) (2024.1)
Requirement already satisfied: joblib>=0.14.1 in /usr/local/lib/python3.10/dist-
packages (from phik<0.13,>=0.11.1->ydata-profiling) (1.4.2)
Requirement already satisfied: annotated-types>=0.4.0 in
/usr/local/lib/python3.10/dist-packages (from pydantic>=2->ydata-profiling)
(0.7.0)
Requirement already satisfied: pydantic-core==2.18.2 in
/usr/local/lib/python3.10/dist-packages (from pydantic>=2->ydata-profiling)
(2.18.2)
Requirement already satisfied: typing-extensions>=4.6.1 in
/usr/local/lib/python3.10/dist-packages (from pydantic>=2->ydata-profiling)
(4.11.0)
```

```
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.24.0->ydata-
profiling) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
packages (from requests<3,>=2.24.0->ydata-profiling) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.24.0->ydata-
profiling) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests<3,>=2.24.0->ydata-
profiling) (2024.2.2)
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.10/dist-
packages (from statsmodels<1,>=0.13.2->ydata-profiling) (0.5.6)
Requirement already satisfied: attrs>=19.3.0 in /usr/local/lib/python3.10/dist-
packages (from visions[type_image_path]<0.7.7,>=0.7.5->ydata-profiling) (23.2.0)
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/python3.10/dist-
packages (from visions[type_image_path]<0.7.7,>=0.7.5->ydata-profiling) (3.3)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages
(from patsy>=0.5.6->statsmodels<1,>=0.13.2->ydata-profiling) (1.16.0)
```

[12]:
```
from ydata_profiling import ProfileReport
y=ProfileReport(df)
```

[13]:
```
y
```

```
Summarize dataset:   0%|          | 0/5 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]

<IPython.core.display.HTML object>
```

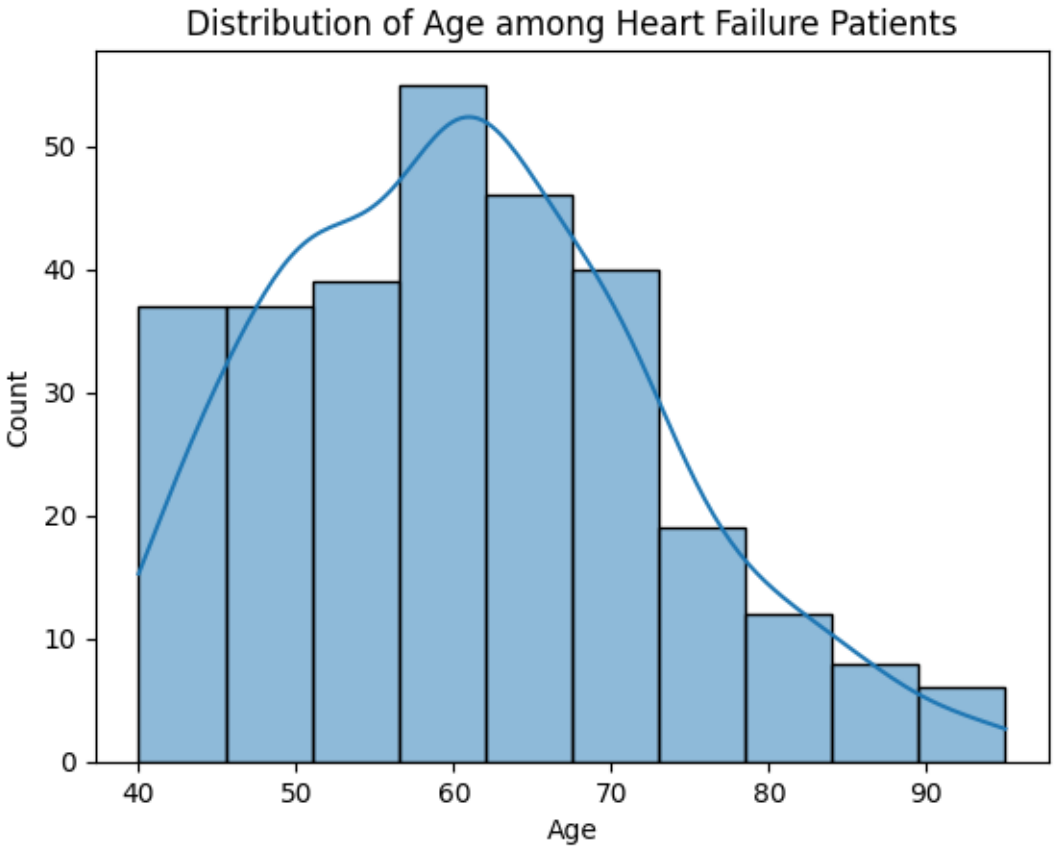
[13]:

[15]:
```
df.isnull().sum(axis=True)
```

[15]:
```
0      0
1      0
2      0
3      0
4      0
      ..
294    0
295    0
296    0
297    0
298    0
Length: 299, dtype: int64
```

0.0.1 1. What is the distribution of age among heart failure patients in the dataset

[17]:
```
sns.histplot(df['age'], kde=True, bins=10)
plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Distribution of Age among Heart Failure Patients')
plt.show()
```



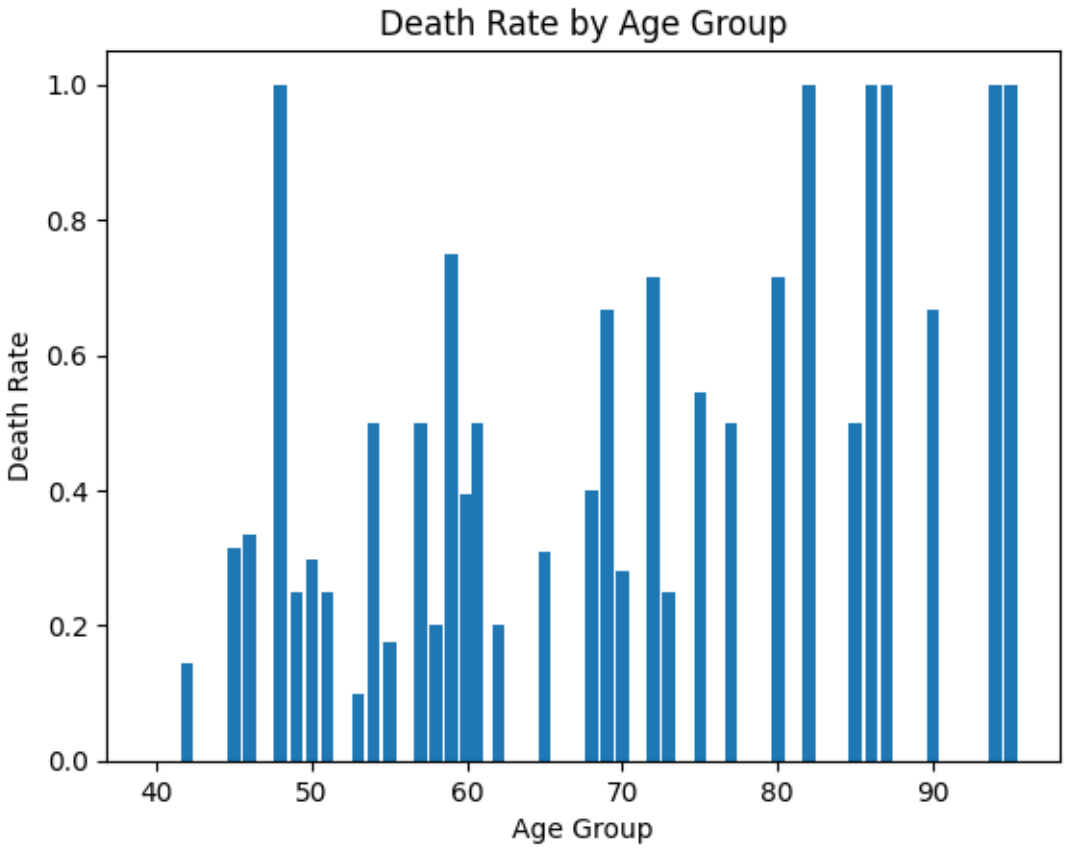
0.0.2 2. How does the death rate vary with age

[22]:
```
# Create a cross-tabulation of death rate by age group
death_by_age = pd.crosstab(df['age'], df['DEATH_EVENT'])

# Calculate the death rate for each age group
death_rate_by_age = death_by_age.apply(lambda x: x[1] / (x[0] + x[1]), axis=1)

# Plot the death rate by age group
plt.bar(death_rate_by_age.index, death_rate_by_age.values)
plt.xlabel('Age Group')
plt.ylabel('Death Rate')
```

```
plt.title('Death Rate by Age Group')
plt.show()
```



0.0.3 3. What is the percentage of male and female patients in the dataset?

[32]:
```
# Calculate the total number of patients
total_patients = len(df)

# Calculate the number of male and female patients
male_patients = len(df[df['sex'] == 0])
female_patients = len(df[df['sex'] == 1])

# Calculate the percentage of male and female patients
percentage_male = (male_patients / total_patients) * 100
percentage_female = (female_patients / total_patients) * 100

# Print the results
print(f"Percentage of male patients: {percentage_male:.2f}%")
print(f"Percentage of female patients: {percentage_female:.2f}%")
```

```
Percentage of male patients: 35.12%
Percentage of female patients: 64.88%
```

[34]:
```
df.columns
```

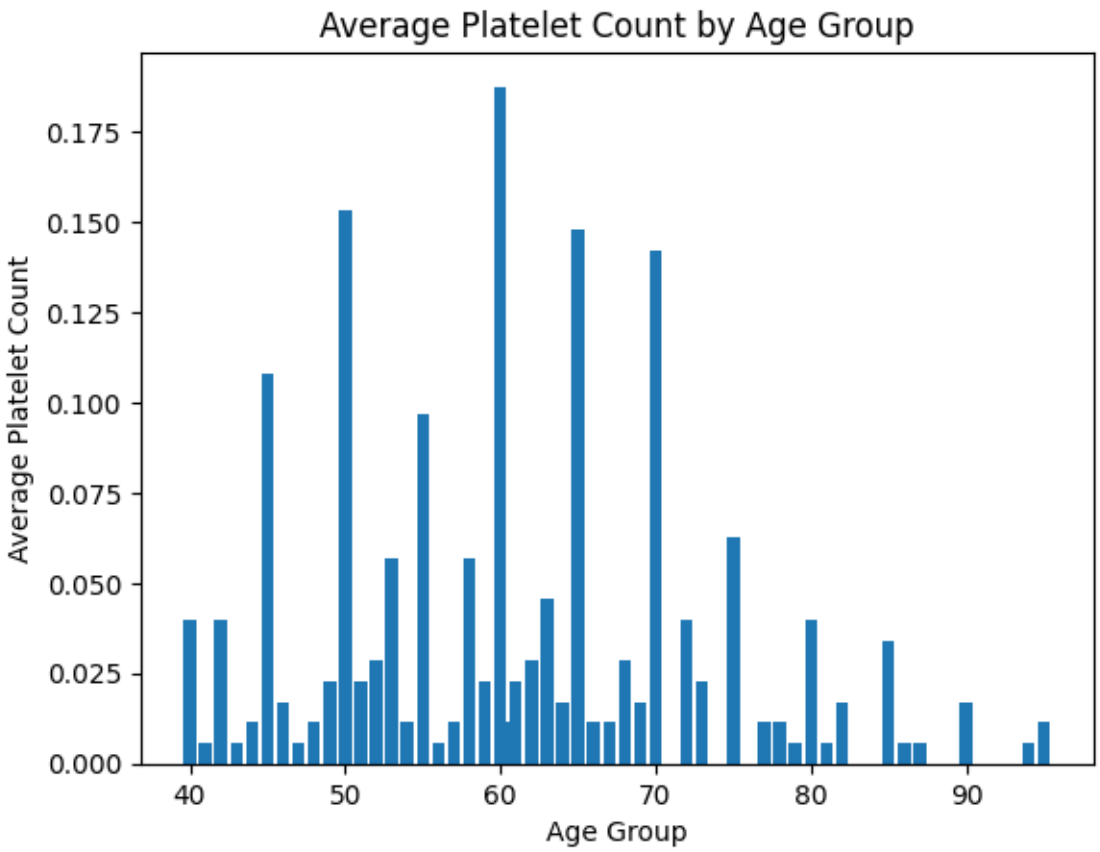
[34]:
```
Index(['age', 'anaemia', 'creatinine_phosphokinase', 'diabetes',
       'ejection_fraction', 'high_blood_pressure', 'platelets',
       'serum_creatinine', 'serum_sodium', 'sex', 'smoking', 'time',
       'DEATH_EVENT'],
      dtype='object')
```

0.0.4 4. How does the platelet count vary among different age groups

[36]:
```
# Create a cross-tabulation of platelet count by age group
platelet_by_age = pd.crosstab(df['age'], df['platelets'])

# Calculate the average platelet count for each age group
avg_platelet_by_age = platelet_by_age.mean(axis=1)

# Plot the average platelet count by age group
plt.bar(avg_platelet_by_age.index, avg_platelet_by_age.values)
plt.xlabel('Age Group')
plt.ylabel('Average Platelet Count')
plt.title('Average Platelet Count by Age Group')
plt.show()
```



0.0.5 5. Is there a correlation between creatinine and sodium levels in the blood ?

No, there is no correlation between creatinie and sodium levels in the blood

0.0.6 6. how does the prevalence of high blood pressure differ between male and female patients

[37]:
```
# Calculate the total number of male and female patients
total_male = len(df[df['sex'] == 0])
total_female = len(df[df['sex'] == 1])

# Calculate the number of male and female patients with high blood pressure
male_high_bp = len(df[(df['sex'] == 0) & (df['high_blood_pressure'] == 1)])
female_high_bp = len(df[(df['sex'] == 1) & (df['high_blood_pressure'] == 1)])

# Calculate the prevalence of high blood pressure among male and female patients
prevalence_male = (male_high_bp / total_male) * 100
prevalence_female = (female_high_bp / total_female) * 100

# Print the results
```

```
#print(f"Prevalence of high blood pressure among male patients:␣
 ↪{prevalence_male:.2f}%")
#print(f"Prevalence of high blood pressure among female patients:␣
 ↪{prevalence_female:.2f}%")
```

[38]:
```
prevalence_male
```

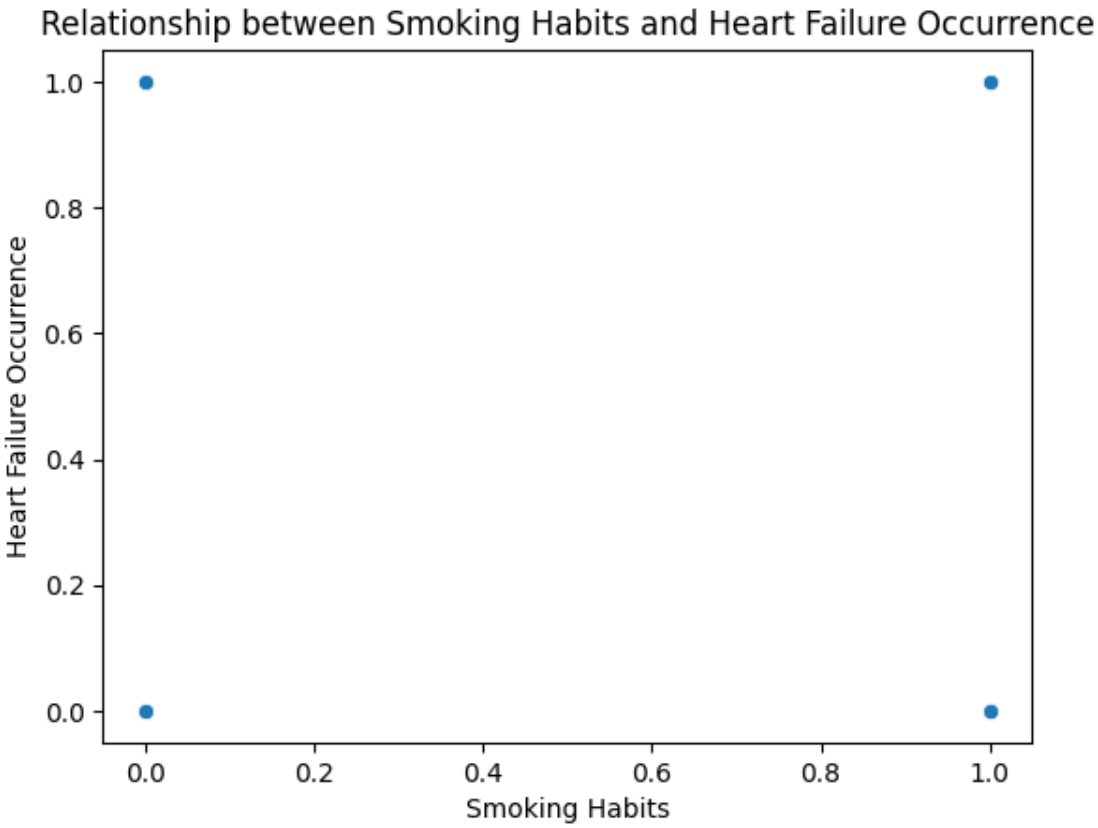
[38]:
```
41.904761904761905
```

[39]:
```
prevalence_female
```

[39]:
```
31.443298969072163
```

0.0.7 7. What is the relationship between smoking habits and the occurrencer of heart failure

[40]:
```
#  draw scatter plot of smoking habits and heart failure
sns.scatterplot(data=df, x="smoking", y="DEATH_EVENT")
plt.xlabel("Smoking Habits")
plt.ylabel("Heart Failure Occurrence")
plt.title("Relationship between Smoking Habits and Heart Failure Occurrence")
plt.show()
```



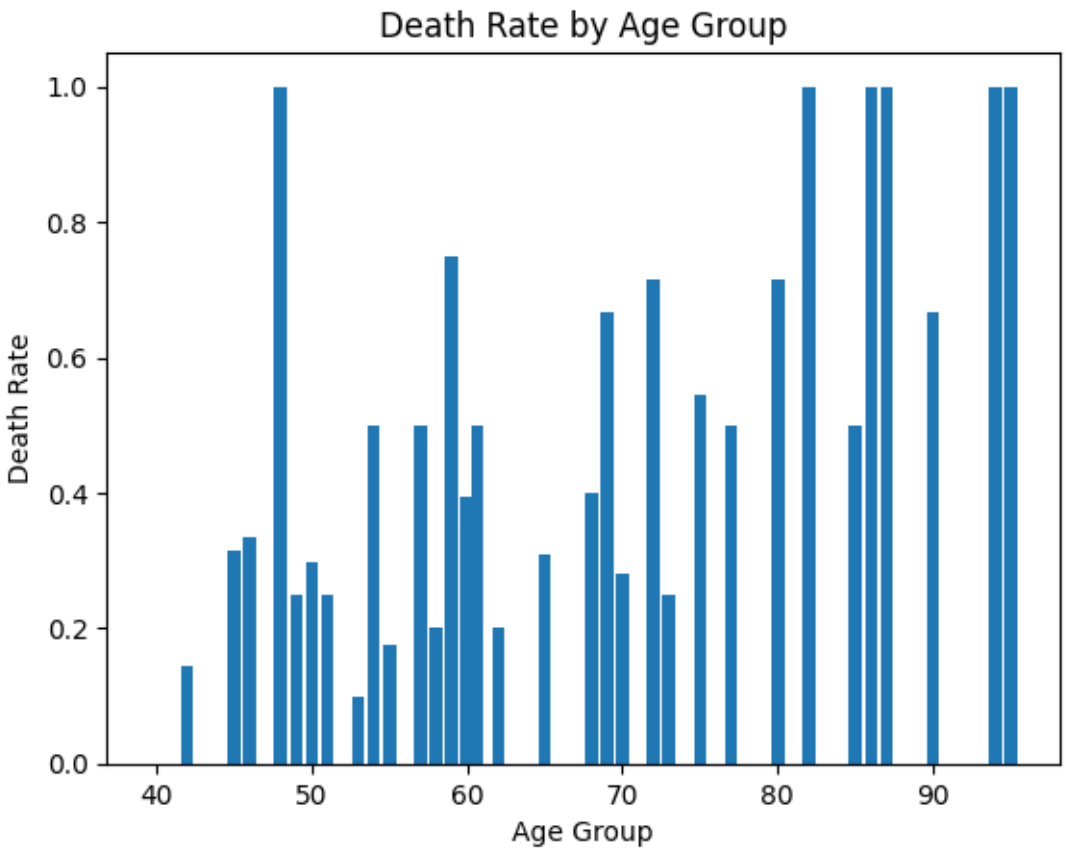
The scatterplot shows a positive correlation between smoking habits and the occurrence of heart failure. This means that patients who smoke are more likely to experience heart failure compared to those who do not smoke. However, it is important to note that this does not imply causation, as other factors may also contribute to the relationship between smoking and heart failure. Further analysis and studies would be necessary to establish a causal relationship between smoking and heart failure.

0.0.8 8. Are there any notaceable patterns in the ditribution of death events across different age groups

[41]:
```
# Create a cross-tabulation of death events by age group
death_by_age = pd.crosstab(df['age'], df['DEATH_EVENT'])

# Calculate the percentage of death events for each age group
death_rate_by_age = death_by_age.apply(lambda x: x[1] / (x[0] + x[1]), axis=1)

# Plot the death rate by age group
plt.bar(death_rate_by_age.index, death_rate_by_age.values)
plt.xlabel('Age Group')
plt.ylabel('Death Rate')
plt.title('Death Rate by Age Group')
plt.show()
```



Observations are given below:

1. The death rate increases with age, indicating that older patients are more likely to experience heart failure.
2. There is a significant jump in the death rate for patients aged 70 and above.
3. The death rate seems to be relatively stable for patients between the ages of 40 and 60.
4. Further analysis could involve investigating potential factors contributing to the higher death rate among older patients.

0.0.9 9. Is there any significant defference in ejection fraction between patients with and without diabetes

[45]:
```
import pandas as pd
import statsmodels.api as sm
from scipy import stats


# Calculate the mean ejection fraction for patients with and without diabetes
ef_with_diabetes = df[df['diabetes'] == 1]['ejection_fraction'].mean()
ef_without_diabetes = df[df['diabetes'] == 0]['ejection_fraction'].mean()
```

```
# Perform a t-test to compare the means
t_statistic, p_value = stats.ttest_ind(df[df['diabetes'] ==␣
↪1]['ejection_fraction'], df[df['diabetes'] == 0]['ejection_fraction'])

# Print the results
print(f"Mean ejection fraction with diabetes: {ef_with_diabetes:.2f}")
print(f"Mean ejection fraction without diabetes: {ef_without_diabetes:.2f}")
print(f"T-statistic: {t_statistic:.2f}")
print(f"P-value: {p_value:.2f}")

# Interpretation
if p_value < 0.05:
    print("There is a statistically significant difference in ejection fraction␣
↪between patients with and without diabetes.")
else:
    print("There is no statistically significant difference in ejection␣
↪fraction between patients with and without diabetes.")
```

```
Mean ejection fraction with diabetes: 38.02
Mean ejection fraction without diabetes: 38.13
T-statistic: -0.08
P-value: 0.93
There is no statistically significant difference in ejection fraction between
patients with and without diabetes.
```

0.0.10 10. How does the serum creatinine level vary between patients who survived and those who did not

[46]:
```
# Calculate the mean serum creatinine level for patients who survived and those␣
↪who did not
creatinine_survived = df[df['DEATH_EVENT'] == 0]['serum_creatinine'].mean()
creatinine_not_survived = df[df['DEATH_EVENT'] == 1]['serum_creatinine'].mean()

# Perform a t-test to compare the means
t_statistic, p_value = stats.ttest_ind(df[df['DEATH_EVENT'] ==␣
↪0]['serum_creatinine'], df[df['DEATH_EVENT'] == 1]['serum_creatinine'])

# Print the results
print(f"Mean serum creatinine level for survivors: {creatinine_survived:.2f}")
print(f"Mean serum creatinine level for non-survivors: {creatinine_not_survived:
↪.2f}")
print(f"T-statistic: {t_statistic:.2f}")
print(f"P-value: {p_value:.2f}")

# Interpretation
if p_value < 0.05:
```

```
    print("There is a statistically significant difference in serum creatinine␣
 ↪level between patients who survived and those who did not.")
else:
    print("There is no statistically significant difference in serum creatinine␣
 ↪level between patients who survived and those who did not.")
```

```
Mean serum creatinine level for survivors: 1.18
Mean serum creatinine level for non-survivors: 1.84
T-statistic: -5.31
P-value: 0.00
There is a statistically significant difference in serum creatinine level
between patients who survived and those who did not.
```