Heart Failure

December 15, 2024

[3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[4]:
```
import os
os.path.exists(r"C:\Users\jarup\Downloads\data science papers\EDA\assignment␣
 ↪project\heart_failure_clinical_records_dataset.csv")
```

[4]:
```
True
```

[7]:
```
df=pd.read_csv(r"C:\Users\jarup\Downloads\data science papers\EDA\assignment␣
 ↪project\heart_failure_clinical_records_dataset.csv")
```

[9]:
```
df
```

[9]:

	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	
..	…	…	…	…	…	
294	62.0	0	61	1	38	
295	55.0	0	1820	0	38	
296	45.0	0	2060	1	60	
297	45.0	0	2413	0	38	
298	50.0	0	196	0	45	

	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	
..	…	…	…	…	…	…
294	1	155000.00	1.1	143	1	
295	0	270000.00	1.2	139	0	

```
296                    0  742000.00                1.8            138    0
297                    0  140000.00                1.4            140    1
298                    0  395000.00                1.6            136    1

     smoking  time  DEATH_EVENT
0          0     4            1
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]
```

[23]:
```
df.isna().sum()
```

[23]:
```
age                         0
anaemia                     0
creatinine_phosphokinase    0
diabetes                    0
ejection_fraction           0
high_blood_pressure         0
platelets                   0
serum_creatinine            0
serum_sodium                0
sex                         0
smoking                     0
time                        0
DEATH_EVENT                 0
dtype: int64
```

[25]:
```
df.shape
```

[25]:
```
(299, 13)
```

[27]:
```
df.size
```

[27]:
```
3887
```

[29]:
```
df.dtypes
```

[29]:
```
age                         float64
anaemia                       int64
creatinine_phosphokinase      int64
diabetes                      int64
ejection_fraction             int64
high_blood_pressure           int64
platelets                   float64
serum_creatinine            float64
serum_sodium                  int64
sex                           int64
smoking                       int64
time                          int64
DEATH_EVENT                   int64
dtype: object
```

[31]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 299 entries, 0 to 298
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
```

[33]:
```
df.describe()
```

[33]:

	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
mean           38.083612            0.351171  263358.029264
std            11.834841            0.478136   97804.236869
min            14.000000            0.000000   25100.000000
25%            30.000000            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
```

[35]:
```
df.drop_duplicates()
```

[35]:
```
      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
..    …        …                         …         …                 …
294  62.0        0                        61         1                 38
295  55.0        0                      1820         0                 38
296  45.0        0                      2060         1                 60
297  45.0        0                      2413         0                 38
298  50.0        0                       196         0                 45
```

```
     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
..                   …        …                 …           …  …
294                    1  155000.00               1.1           143    1
295                    0  270000.00               1.2           139    0
296                    0  742000.00               0.8           138    0
297                    0  140000.00               1.4           140    1
298                    0  395000.00               1.6           136    1

     smoking  time  DEATH_EVENT
0          0     4            1
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]
```

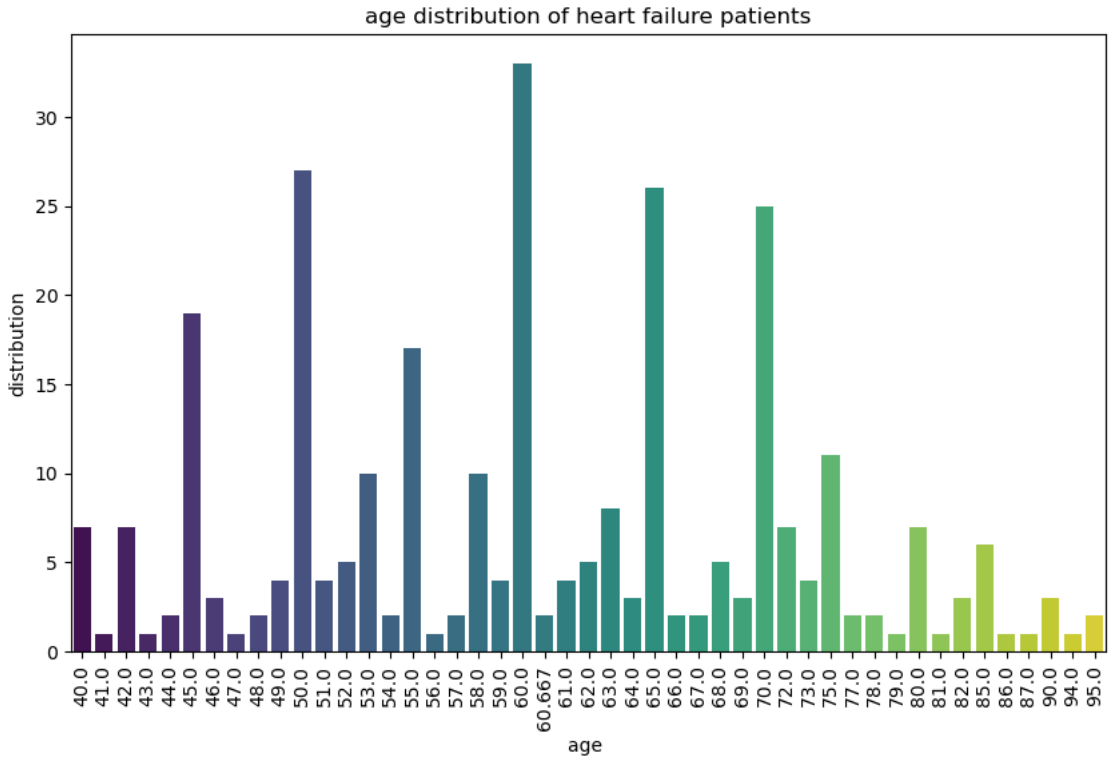
[37]:
```
df.columns
```

[37]:
```
Index(['age', 'anaemia', 'creatinine_phosphokinase', 'diabetes',
       'ejection_fraction', 'high_blood_pressure', 'platelets',
       'serum_creatinine', 'serum_sodium', 'sex', 'smoking', 'time',
       'DEATH_EVENT'],
      dtype='object')
```

[39]:
```
#distribution of age among heart failure patients in the dataset
plt.figure(figsize=(10,6))
sns.countplot(x='age',data =df,palette='viridis')
plt.title("age distribution of heart failure patients")
plt.xlabel('age ')
plt.ylabel('distribution')
plt.xticks(rotation=90)
plt.show()
```

```
C:\Users\jarup\AppData\Local\Temp\ipykernel_15264\2555169396.py:3:
FutureWarning:
```

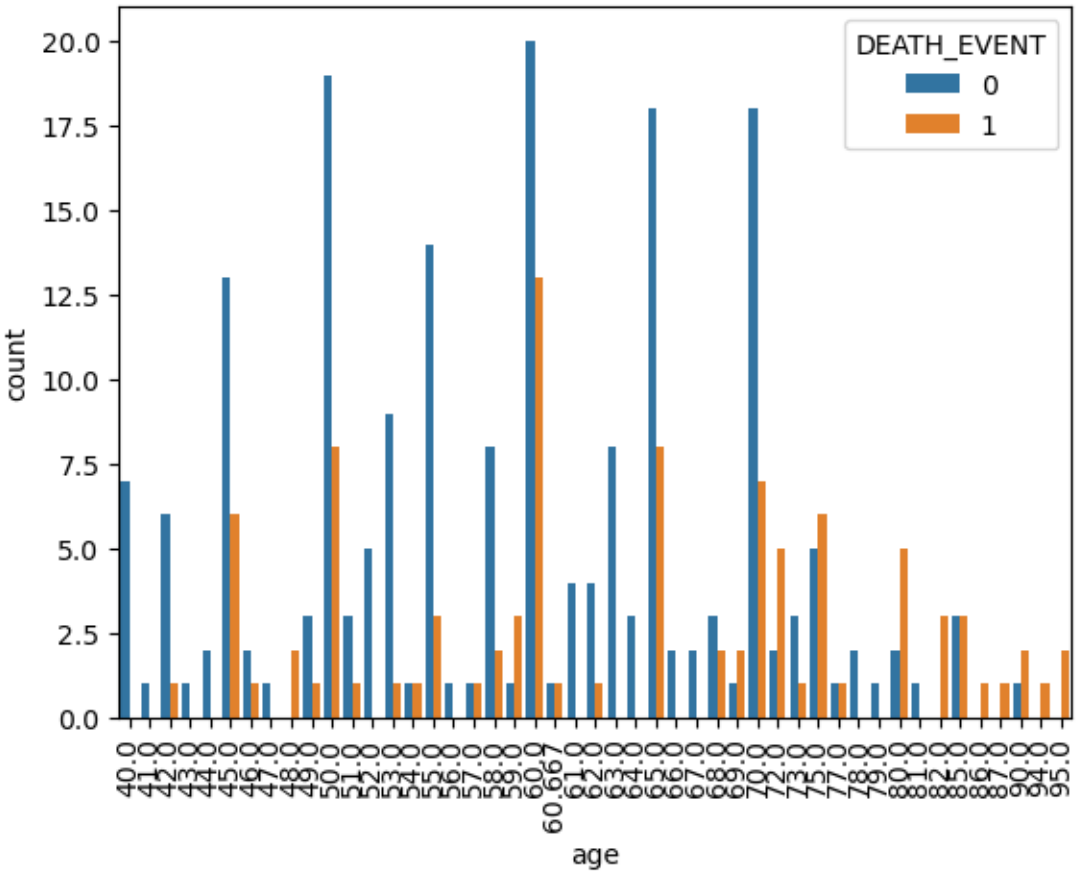
```
Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.countplot(x='age',data =df,palette='viridis')
```



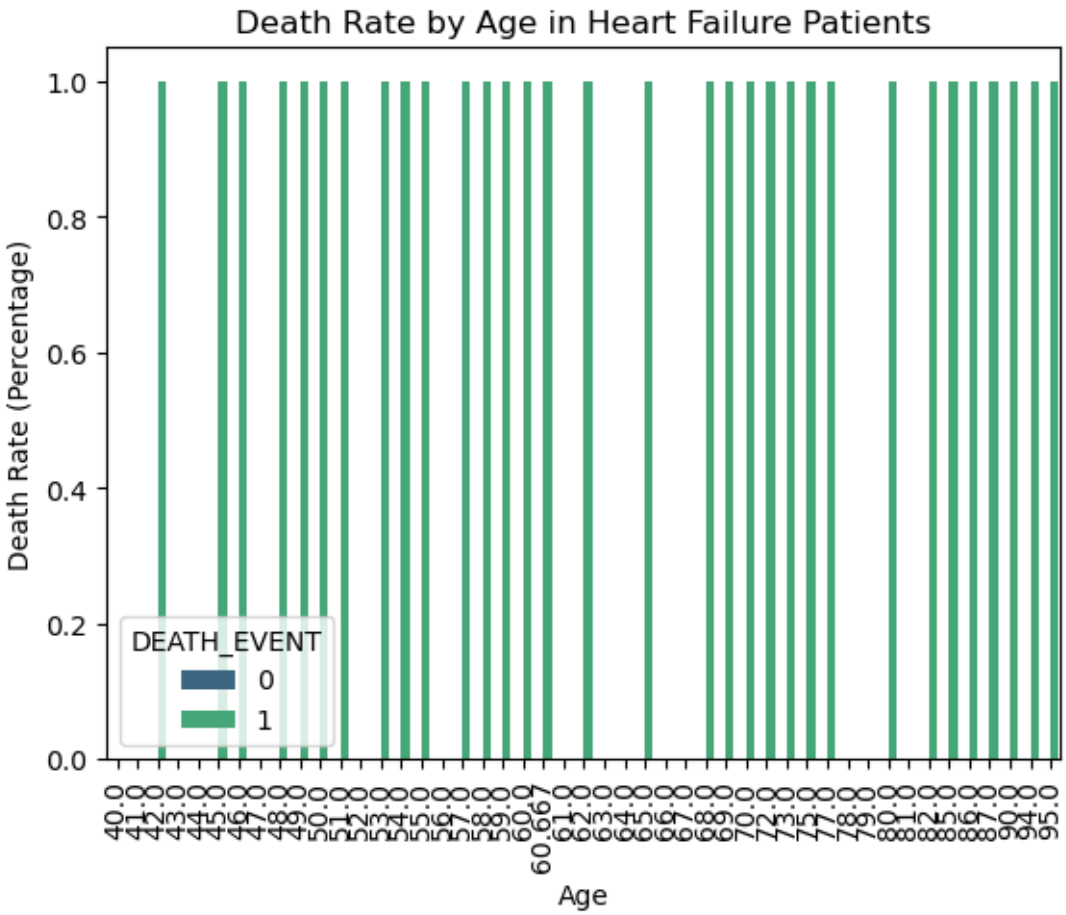
[41]:
```
#distribution of age among heart failure patients in the dataset
plt.figure(figsize=(10,6))
sns.histplot(x='age',data =df,palette='viridis',bins=20)
plt.title("age distribution of heart failure patients")
plt.xlabel('age ')
plt.ylabel('number of patients')
plt.xticks(rotation=90)
plt.show()
```

```
C:\Users\jarup\AppData\Local\Temp\ipykernel_15264\1526458655.py:3: UserWarning:
Ignoring `palette` because no `hue` variable has been assigned.
  sns.histplot(x='age',data =df,palette='viridis',bins=20)
```

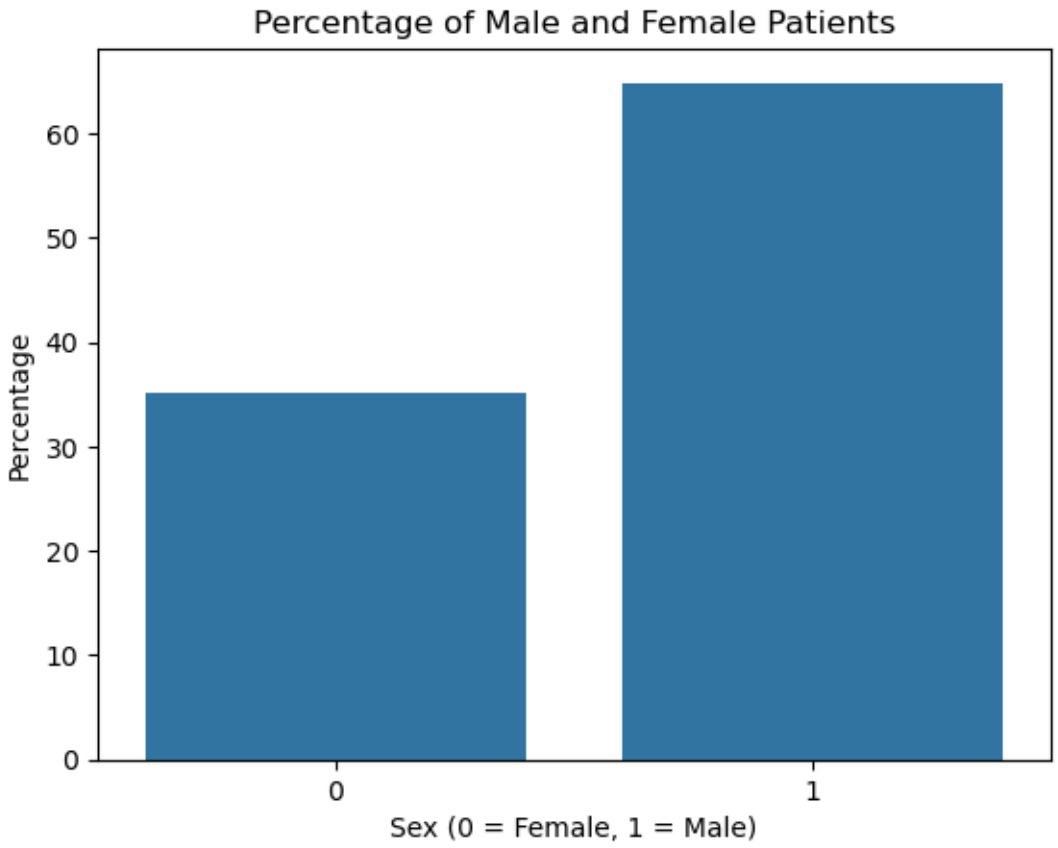
[43]:
```
#the death rate vary with age
sns.countplot(df,x='age',hue='DEATH_EVENT')
plt.xticks(rotation=90)
plt.show()
```



[45]:
```
sns.barplot(x='age', y='DEATH_EVENT', data=df, hue='DEATH_EVENT',␣
 ↪palette='viridis')
plt.title("Death Rate by Age in Heart Failure Patients")
plt.xlabel('Age')
plt.ylabel('Death Rate (Percentage)')
plt.xticks(rotation=90)
plt.show()
```



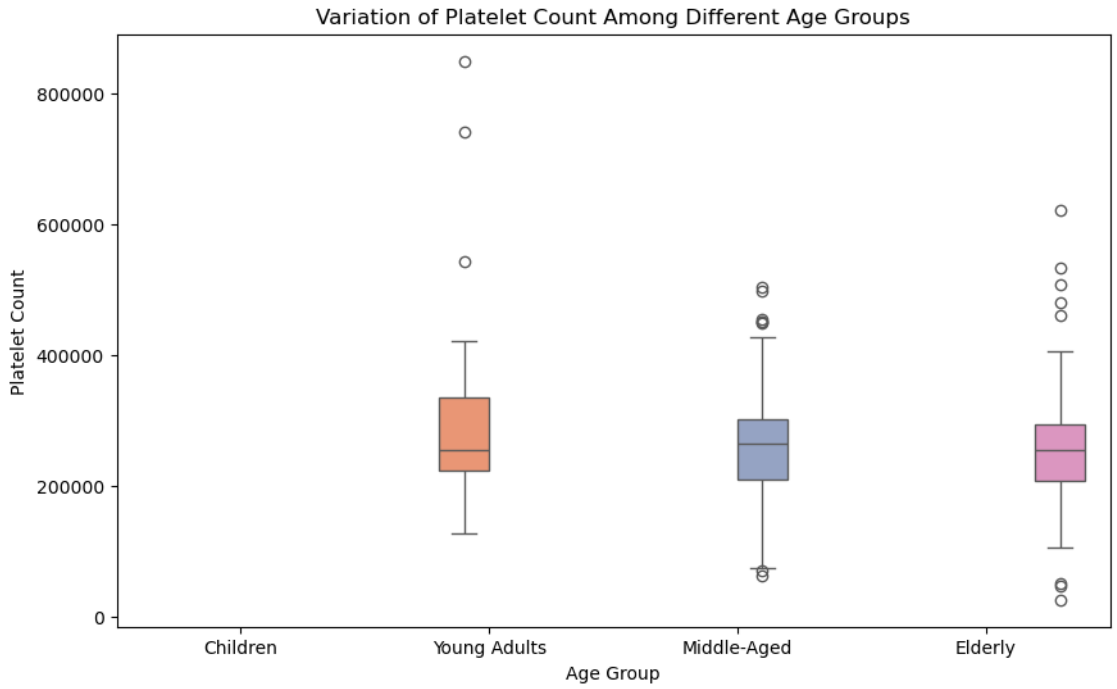
[47]:
```
percentages = df['sex'].value_counts(normalize=True) * 100
sns.barplot(x=percentages.index, y=percentages.values)
plt.xlabel('Sex (0 = Female, 1 = Male)')
plt.ylabel('Percentage')
plt.title('Percentage of Male and Female Patients')
plt.show()
print(percentages)
```




```
sex
1    64.882943
0    35.117057
Name: proportion, dtype: float64
```

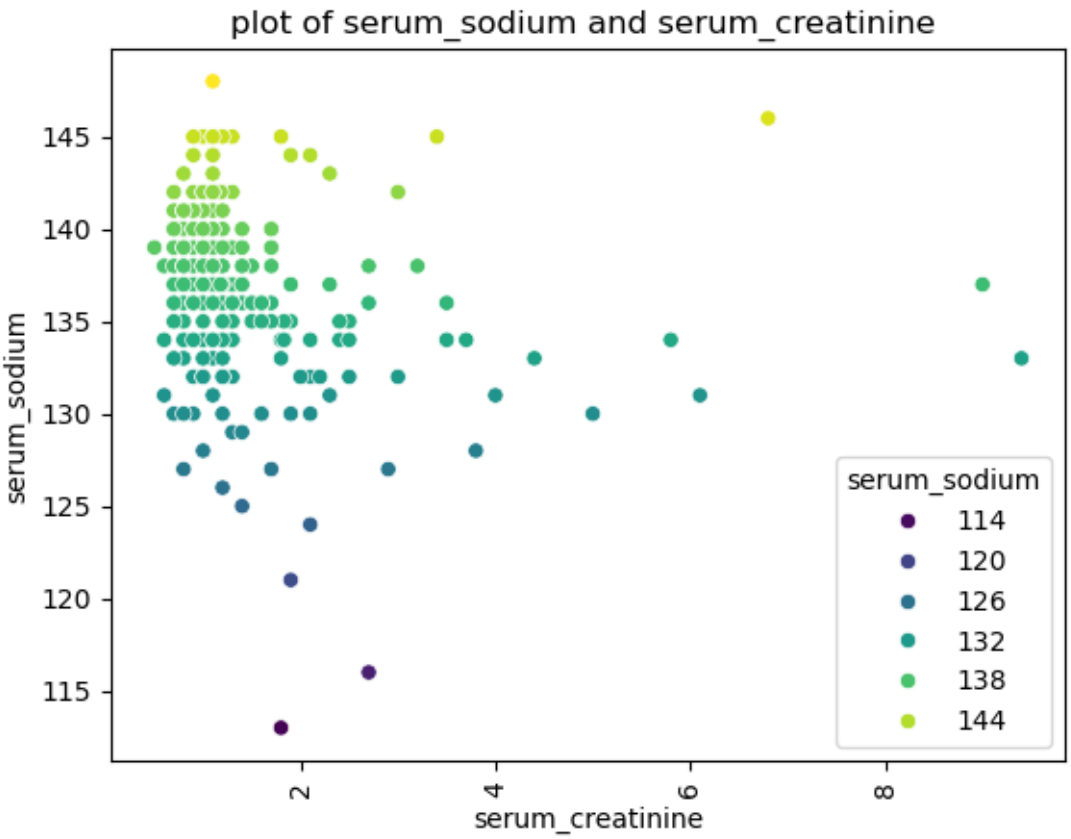
[49]:
```
#4- How does the platelet count vary among different age groups<
```

[51]:
```
bins = [0, 18, 45, 65, 100]
labels = ['Children', 'Young Adults', 'Middle-Aged', 'Elderly']
df['age_group'] = pd.cut(df['age'], bins=bins, labels=labels)
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='age_group', y='platelets',hue='age_group',␣
 ↪palette='Set2')
plt.title('Variation of Platelet Count Among Different Age Groups')
plt.xlabel('Age Group')
plt.ylabel('Platelet Count')
plt.show()
mean_platelet = df.groupby('age_group',observed=True)['platelets'].mean()
print(mean_platelet)
```




```
age_group
Young Adults    295930.597838
Middle-Aged     258123.100347
Elderly         259992.385056
Name: platelets, dtype: float64
```

[71]:
```
#is there a correlation between creatinine and sodium levels in the blood
sns.scatterplot(x='serum_creatinine', y='serum_sodium', data=df,␣
 ↪hue='serum_sodium', palette='viridis')
plt.title("plot of serum_sodium and serum_creatinine ")
plt.xlabel('serum_creatinine')
plt.ylabel('serum_sodium')
plt.xticks(rotation=90)
plt.show()
```

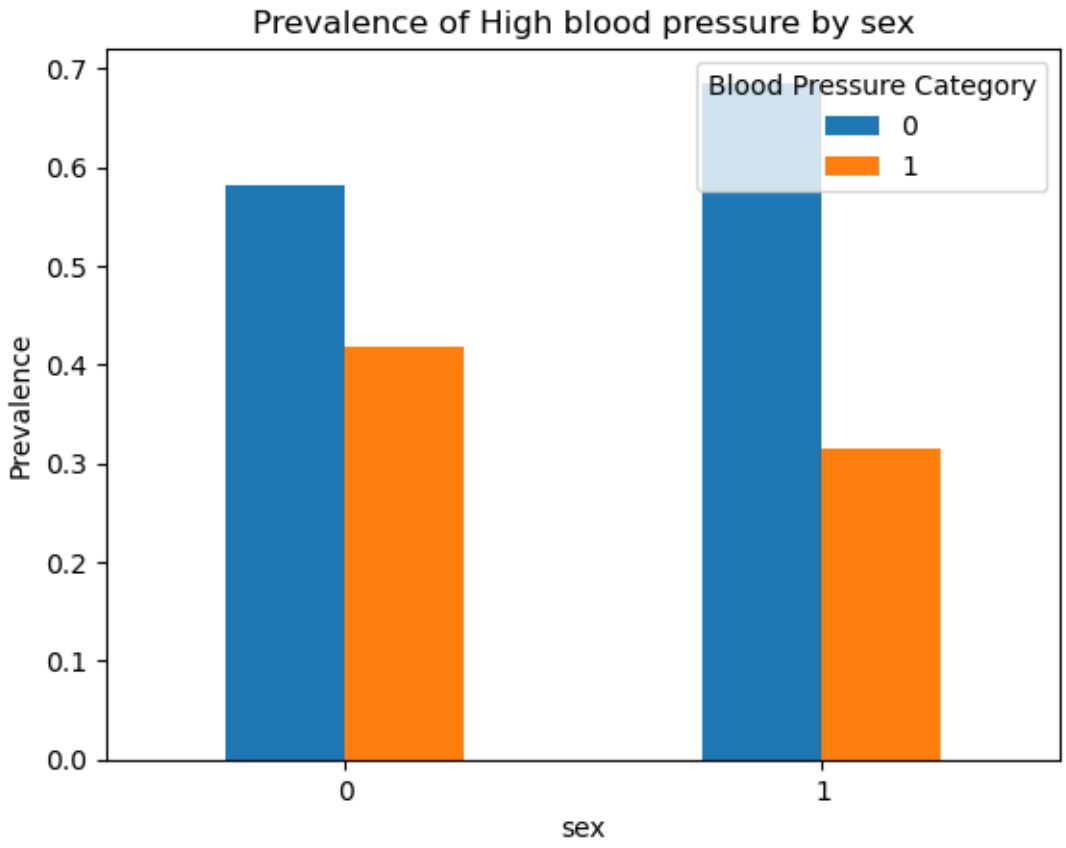


[75]:
```
df.columns
```

[75]:
```
Index(['age', 'anaemia', 'creatinine_phosphokinase', 'diabetes',
       'ejection_fraction', 'high_blood_pressure', 'platelets',
       'serum_creatinine', 'serum_sodium', 'sex', 'smoking', 'time',
       'DEATH_EVENT', 'age_group'],
      dtype='object')
```

[85]:
```
prevalence_by_gender = df.groupby('sex')['high_blood_pressure'].
  ↪value_counts(normalize=True).unstack()
plt.figure(figsize=(8, 6))
prevalence_by_gender.plot(kind='bar')
plt.title('Prevalence of High blood pressure by sex')
plt.xlabel('sex')
plt.ylabel('Prevalence')
plt.xticks(rotation=0)
plt.legend(title='Blood Pressure Category')
plt.show()
```

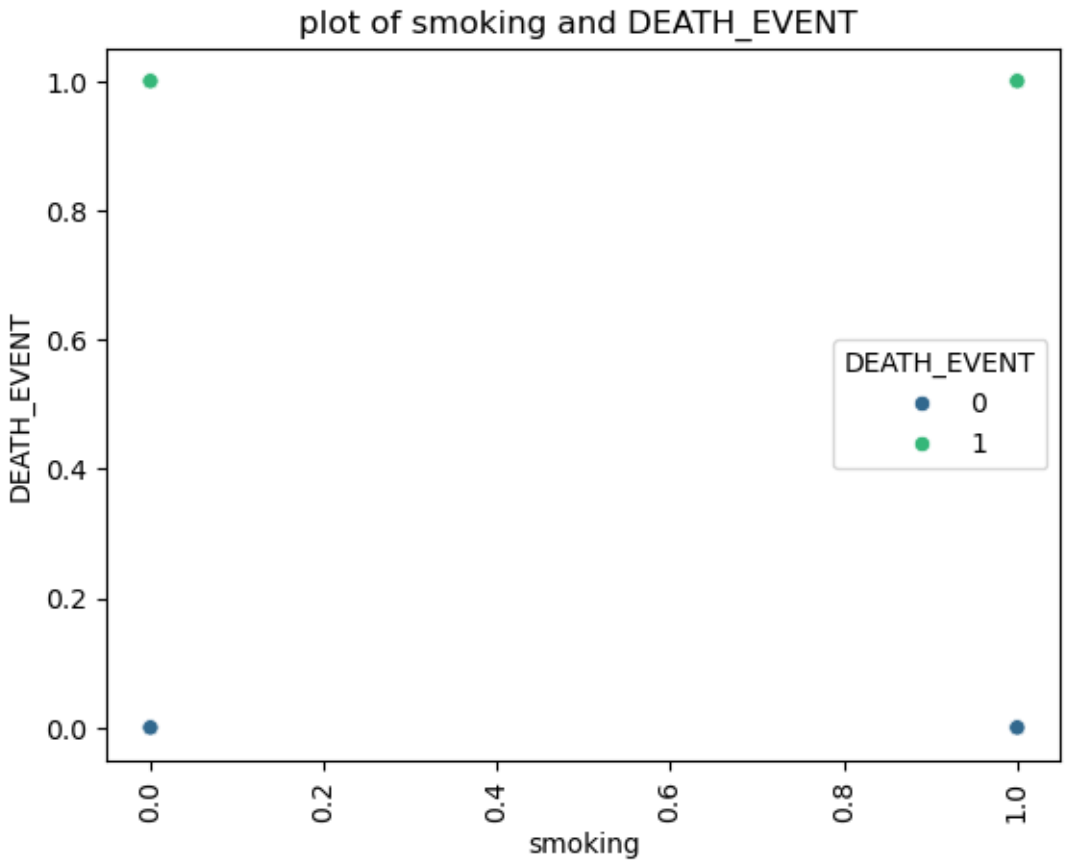
```
<Figure size 800x600 with 0 Axes>
```



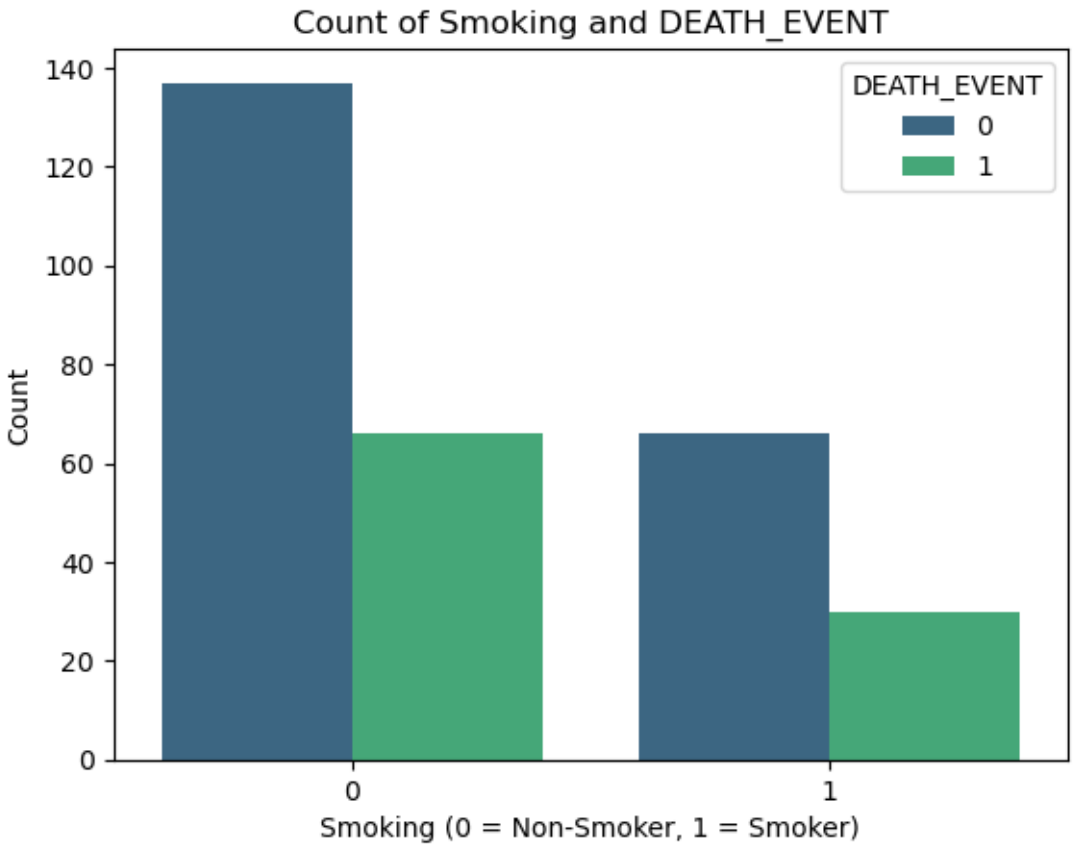
[87]:
```
df.columns
```

[87]:
```
Index(['age', 'anaemia', 'creatinine_phosphokinase', 'diabetes',
       'ejection_fraction', 'high_blood_pressure', 'platelets',
       'serum_creatinine', 'serum_sodium', 'sex', 'smoking', 'time',
       'DEATH_EVENT', 'age_group'],
      dtype='object')
```

[89]:
```
#What is the relationship between smoking habits and the occurrence of heart␣
 ↪failure
sns.scatterplot(x='smoking', y='DEATH_EVENT', data=df, hue='DEATH_EVENT',␣
 ↪palette='viridis')
plt.title("plot of smoking and DEATH_EVENT ")
plt.xlabel('smoking')
plt.ylabel('DEATH_EVENT')
plt.xticks(rotation=90)
plt.show()
```



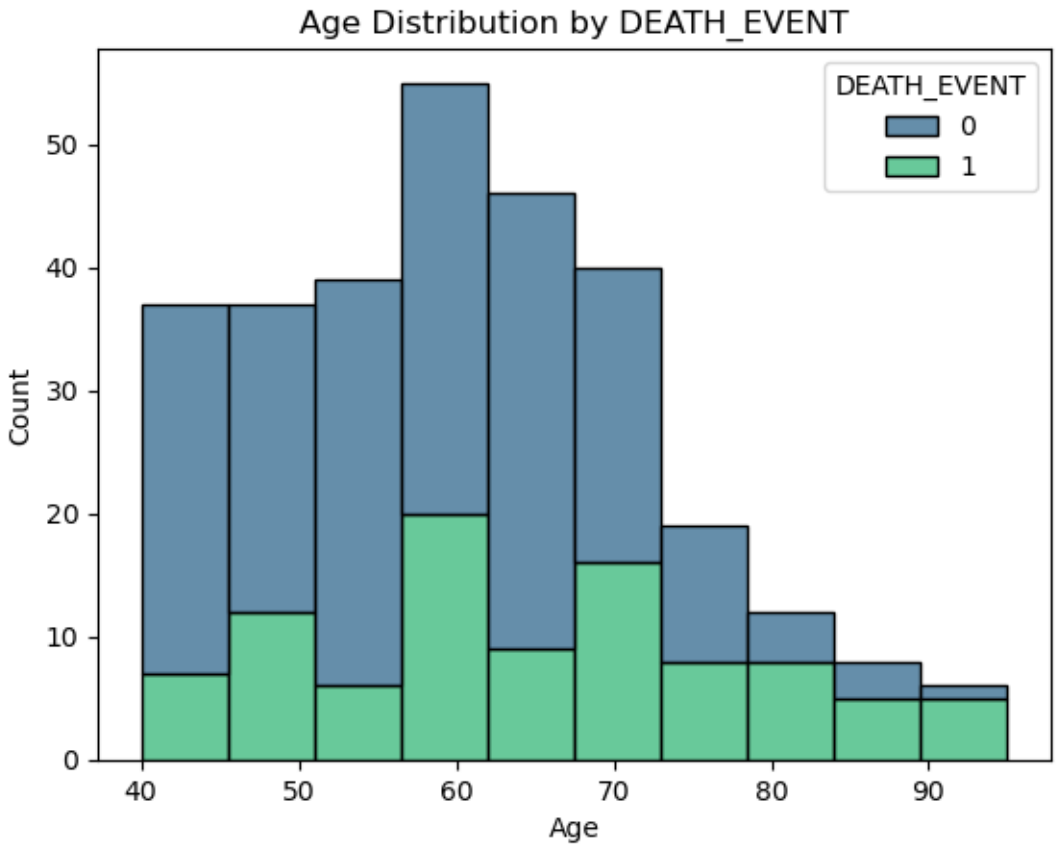
[123]:
```
#What is the relationship between smoking habits and the occurrence of heart␣
 ↪failure
sns.countplot(x='smoking', hue='DEATH_EVENT', data=df, palette='viridis')
plt.title("Count of Smoking and DEATH_EVENT")
plt.xlabel('Smoking (0 = Non-Smoker, 1 = Smoker)')
plt.ylabel('Count')
plt.show()
```



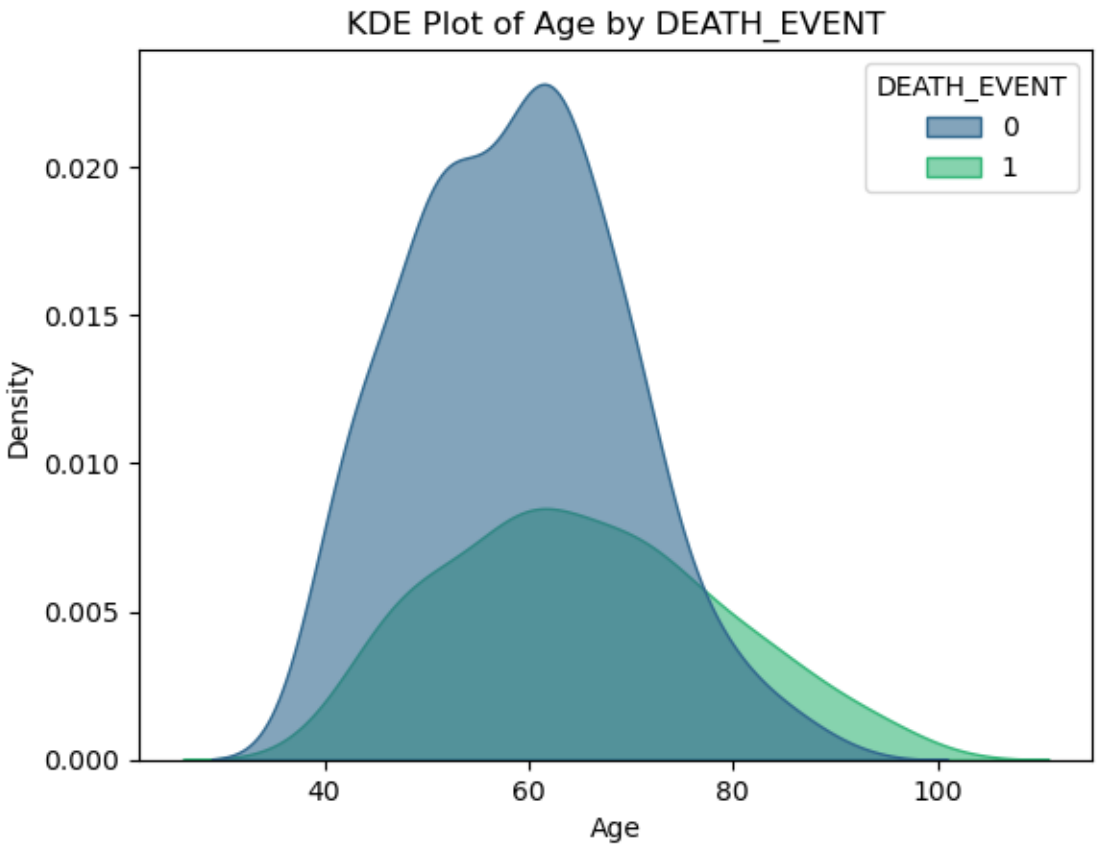
[103]:
```
df.columns
```

[103]:
```
Index(['age', 'anaemia', 'creatinine_phosphokinase', 'diabetes',
       'ejection_fraction', 'high_blood_pressure', 'platelets',
       'serum_creatinine', 'serum_sodium', 'sex', 'smoking', 'time',
       'DEATH_EVENT', 'age_group'],
      dtype='object')
```

[115]:
```
#Are there any noticeable patterns in the distribution of death events across␣
 ↪different age groups
sns.histplot(x='age', hue='DEATH_EVENT', data=df, multiple='stack',␣
 ↪palette='viridis')
plt.title("Age Distribution by DEATH_EVENT")
plt.xlabel('Age')
plt.ylabel('Count')
plt.show()
```



[113]:
```
sns.kdeplot(data=df, x='age', hue='DEATH_EVENT', fill=True, palette='viridis',␣
↪alpha=0.6)
plt.title("KDE Plot of Age by DEATH_EVENT")
plt.xlabel('Age')
plt.ylabel('Density')
plt.show()
```



[125]:
```
df.columns
```

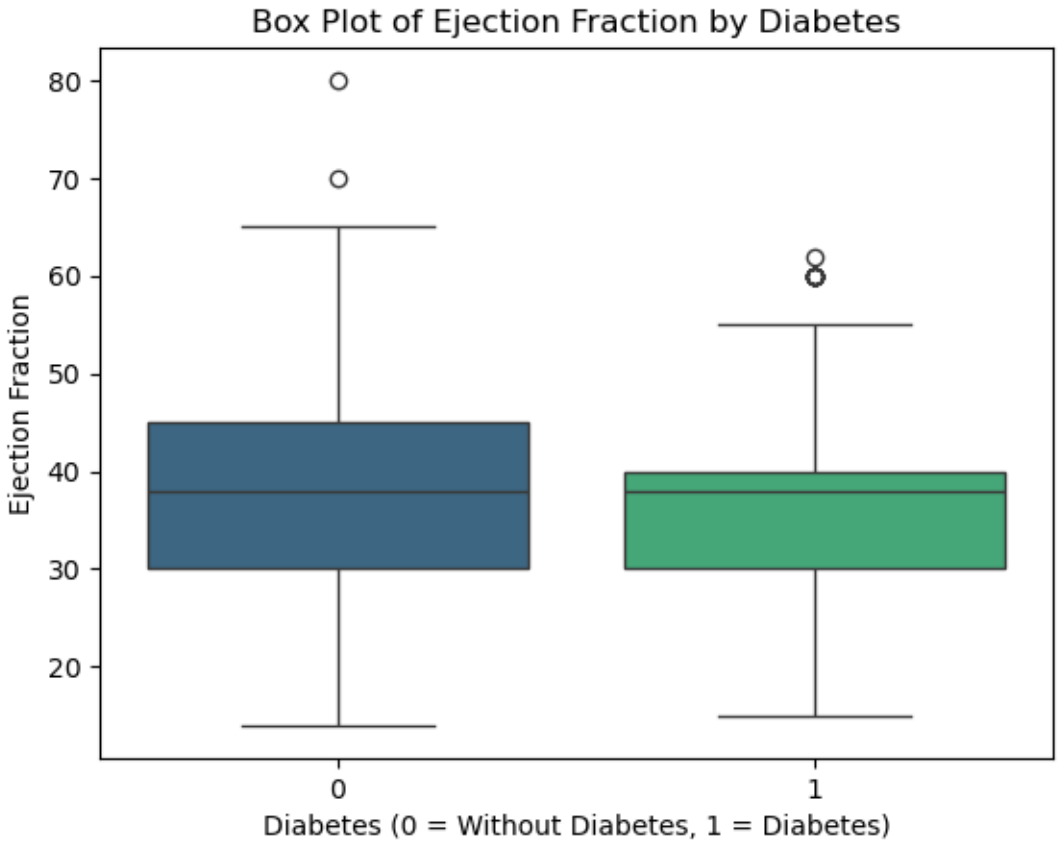
[125]:
```
Index(['age', 'anaemia', 'creatinine_phosphokinase', 'diabetes',
       'ejection_fraction', 'high_blood_pressure', 'platelets',
       'serum_creatinine', 'serum_sodium', 'sex', 'smoking', 'time',
       'DEATH_EVENT', 'age_group'],
      dtype='object')
```

[135]:
```
# Is there any significant difference in ejection fraction between patients␣
↪with and without diabetes
sns.boxplot(x='diabetes', y='ejection_fraction', data=df, palette='viridis')
plt.title("Box Plot of Ejection Fraction by Diabetes")
plt.xlabel('Diabetes (0 = Without Diabetes, 1 = Diabetes)')
plt.ylabel('Ejection Fraction')
plt.show()
```

```
C:\Users\jarup\AppData\Local\Temp\ipykernel_15264\3038049500.py:2:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
```

```
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.
```

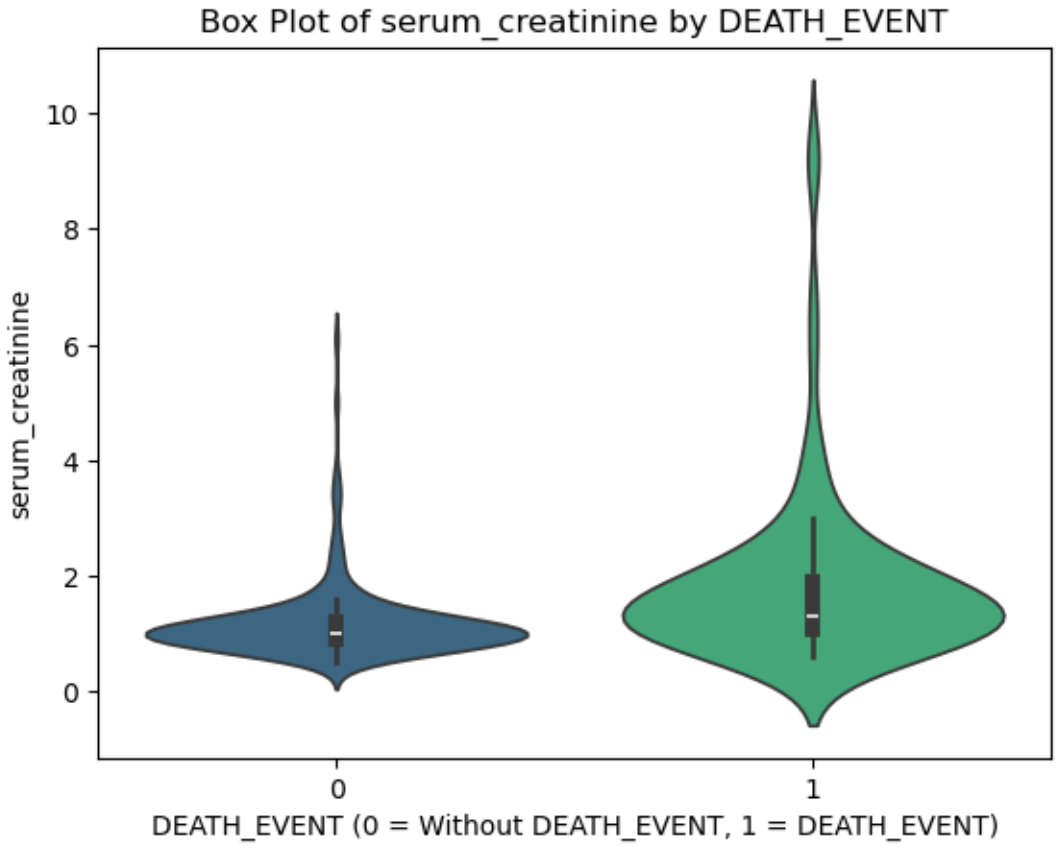
```
  sns.boxplot(x='diabetes', y='ejection_fraction', data=df, palette='viridis')
```



[141]:
```
#How does the serum creatinine level vary between patients who survived and␣
 ↪those who did not?
sns.violinplot(x='DEATH_EVENT', y='serum_creatinine', data=df,␣
 ↪palette='viridis')
plt.title("Box Plot of serum_creatinine by DEATH_EVENT")
plt.xlabel('DEATH_EVENT (0 = Without DEATH_EVENT, 1 = DEATH_EVENT)')
plt.ylabel('serum_creatinine')
plt.show()
```

```
C:\Users\jarup\AppData\Local\Temp\ipykernel_15264\1857119742.py:2:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.
```

```
  sns.violinplot(x='DEATH_EVENT', y='serum_creatinine', data=df,
palette='viridis')
```



[ ]: