

Capstone Project-3


EXploratory Data Analysis (EDA) Project

Project-Heart Failure Analysis

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

1) Load the Data File [link text](#)

```
df=pd.read_csv('/content/heart_failure_clinical_records_dataset (1).csv')
df
```



	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_s
0	75.0	0	582	0	20	1	265000.00	1.9	
1	55.0	0	7861	0	38	0	263358.03	1.1	
2	65.0	0	146	0	20	0	162000.00	1.3	
3	50.0	1	111	0	20	0	210000.00	1.9	
4	65.0	1	160	1	20	0	327000.00	2.7	
...	
294	62.0	0	61	1	38	1	155000.00	1.1	
295	55.0	0	1820	0	38	0	270000.00	1.2	
296	45.0	0	2060	1	60	0	742000.00	0.8	
297	45.0	0	2413	0	38	0	140000.00	1.4	
298	50.0	0	196	0	45	0	395000.00	1.6	

299 rows × 13 columns

Next steps:


Generate code with df

 View recommended plots

New interactive sheet

2) Print first 5 rows of data

```
df.head()
```



	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sod
0	75.0	0	582	0	20	1	265000.00	1.9	
1	55.0	0	7861	0	38	0	263358.03	1.1	
2	65.0	0	146	0	20	0	162000.00	1.3	
3	50.0	1	111	0	20	0	210000.00	1.9	
4	65.0	1	160	1	20	0	327000.00	2.7	

Next steps:


Generate code with df

 View recommended plots

New interactive sheet

3)Print last 5 rows of data


```
df.tail()
```



	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_s
294	62.0	0	61	1	38	1	155000.0	1.1	
295	55.0	0	1820	0	38	0	270000.0	1.2	
296	45.0	0	2060	1	60	0	742000.0	0.8	
297	45.0	0	2413	0	38	0	140000.0	1.4	
298	50.0	0	196	0	45	0	395000.0	1.6	

4)Basic cleaning of data (checking null values, missing value)


```
#cheak null values
df.isnull().sum()
```



	0
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

```
#Drop rows with missing values
df=df.dropna()
```



	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_s
0	75.0	0	582	0	20	1	265000.00	1.9	
1	55.0	0	7861	0	38	0	263358.03	1.1	
2	65.0	0	146	0	20	0	162000.00	1.3	
3	50.0	1	111	0	20	0	210000.00	1.9	
4	65.0	1	160	1	20	0	327000.00	2.7	
...
294	62.0	0	61	1	38	1	155000.00	1.1	
295	55.0	0	1820	0	38	0	270000.00	1.2	
296	45.0	0	2060	1	60	0	742000.00	0.8	
297	45.0	0	2413	0	38	0	140000.00	1.4	
298	50.0	0	196	0	45	0	395000.00	1.6	

299 rows × 13 columns

Next steps:

Generate code with df

 View recommended plots

New interactive sheet

5) Some information

- 1.Anaemia: 0 -> No || 1 -> Yes
- 2.diabetes: 0 -> No || 1 -> Yes
- 3.high_blood_pressure: 0 -> No || 1 -> Yes
- 4.sex: 0 -> Female || 1 -> Male
- 5.smoking: 0 -> No || 1 -> Yes
- 6.DEATH_EVENT: 0 -> No || 1 -> Yes

6)Get some info on the dataset

```
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
```

7)Remove un-needed data - time colum

Start coding or generate with AI.

8)Get some description of the data.

```
df.describe()
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine
count	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000
mean	60.833893	0.431438	581.839465	0.418060	38.083612	0.351171	263358.029264	180.925000
std	11.894809	0.496107	970.287881	0.494067	11.834841	0.478136	97804.236869	69.125000
min	40.000000	0.000000	23.000000	0.000000	14.000000	0.000000	25100.000000	54.000000
25%	51.000000	0.000000	116.500000	0.000000	30.000000	0.000000	212500.000000	119.000000
50%	60.000000	0.000000	250.000000	0.000000	38.000000	0.000000	262000.000000	180.000000
75%	70.000000	1.000000	582.000000	1.000000	45.000000	1.000000	303500.000000	241.000000
max	95.000000	1.000000	7861.000000	1.000000	80.000000	1.000000	850000.000000	449.000000

9)Shape of the Datas

```
df.shape
(299, 13)
```

10)Find how many records are there. (value_counts)

- 1) gender
- 2)high blood pressure
- 3)diabetes
- 4)smoking
- 5)death_event

```
#1)gender
df['sex'].value_counts()
```

```
count
sex
1    194
0    105
dtype: int64
```

```
#2)high blood pressure
df['high_blood_pressure'].value_counts()
```

```
↵
```

	count
high_blood_pressure	
0	194
1	105

dtype: int64

```
#3)diabetes
df['diabetes'].value_counts()
```

```
↵
```

	count
diabetes	
0	174
1	125

dtype: int64

```
#4)Smoking
df['smoking'].value_counts()
```

```
↵
```

	count
smoking	
0	203
1	96

dtype: int64

```
#5)Death event
df['DEATH_EVENT'].value_counts()
```

```
↵
```

	count
DEATH_EVENT	
0	203
1	96

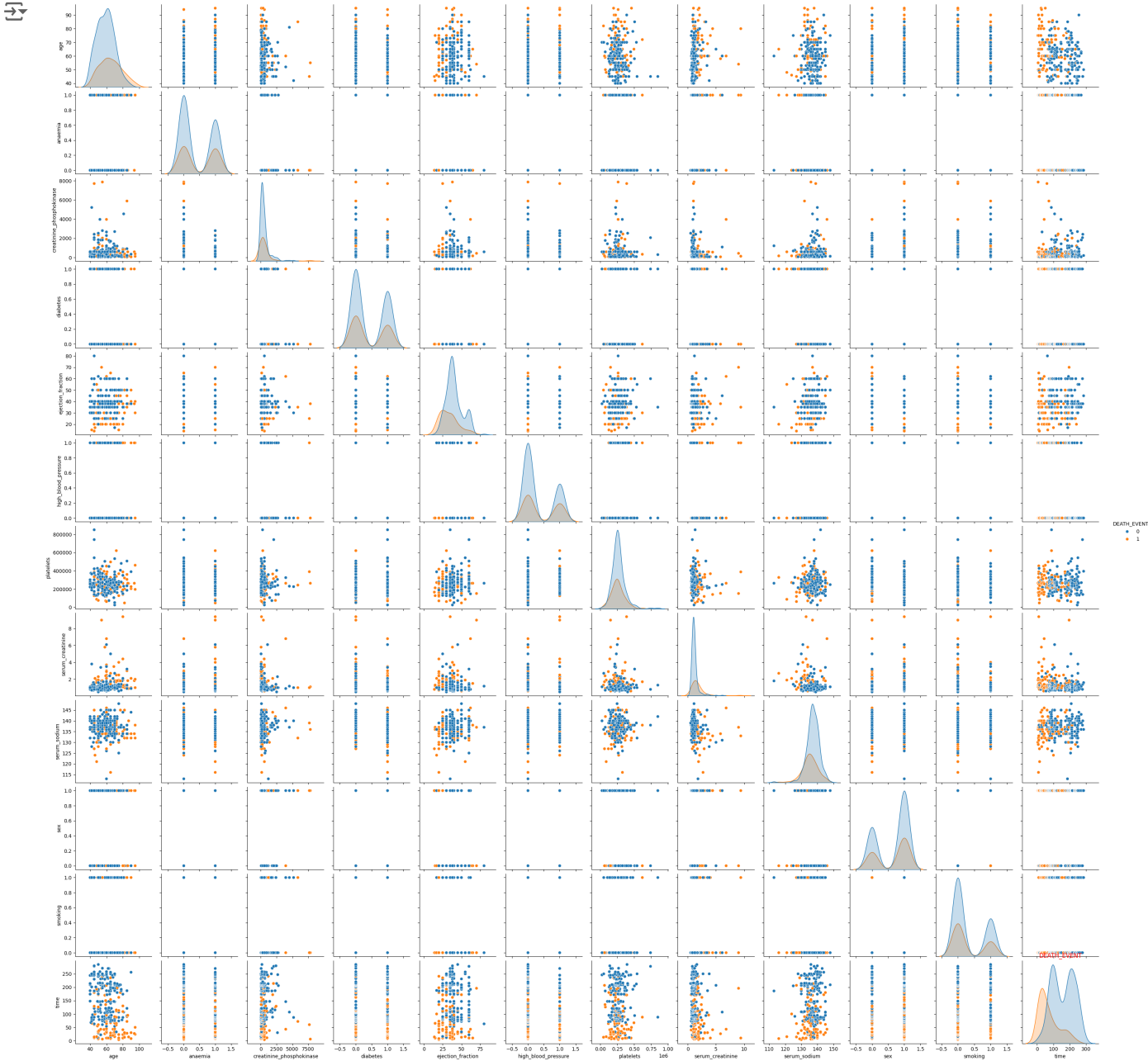
dtype: int64

Visualization

```
import seaborn as sns
import matplotlib.pyplot as plt
```

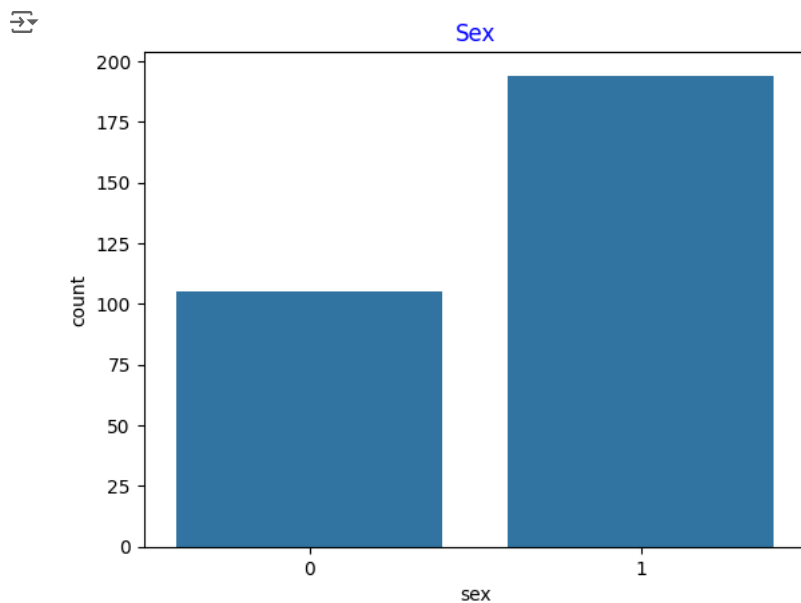
1)Pairplot with relation to death event:

```
sns.pairplot(df,hue='DEATH_EVENT')
plt.title('DEATH_EVENT',color='red')
plt.show()
```



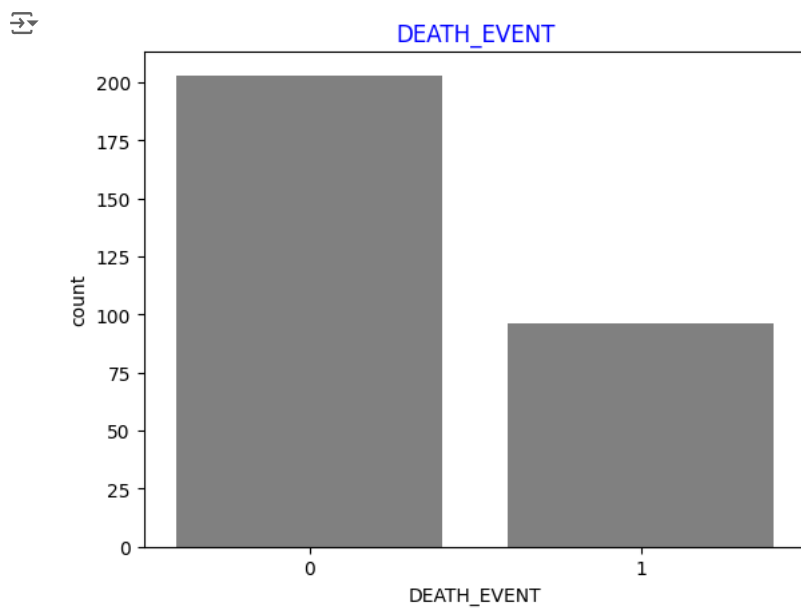
2)Bar plot for 'Sex':

```
sns.countplot(x='sex',data=df)
plt.title('Sex',color='blue')
plt.show()
```



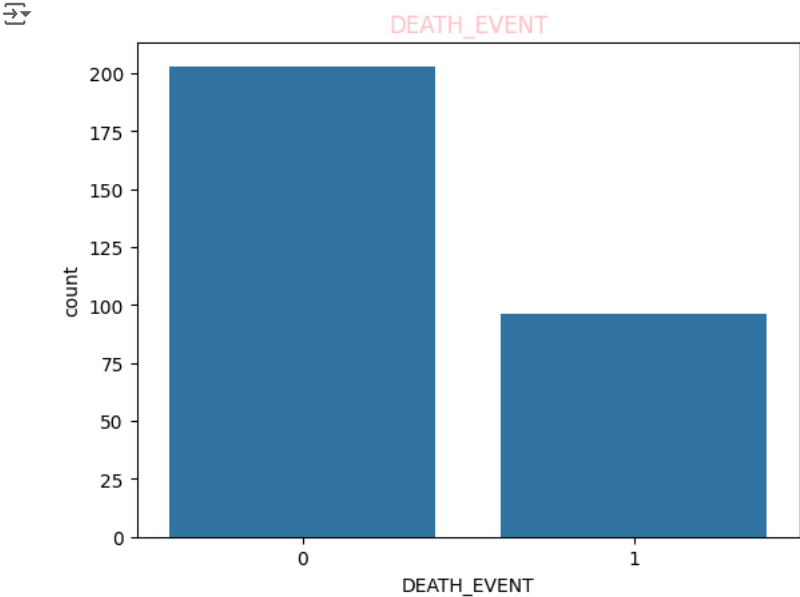
3)Bar plot for 'DEATH_EVENT':

```
sns.countplot(x='DEATH_EVENT',data=df,color='Gray')
plt.title('DEATH_EVENT',color='blue')
plt.show()
```



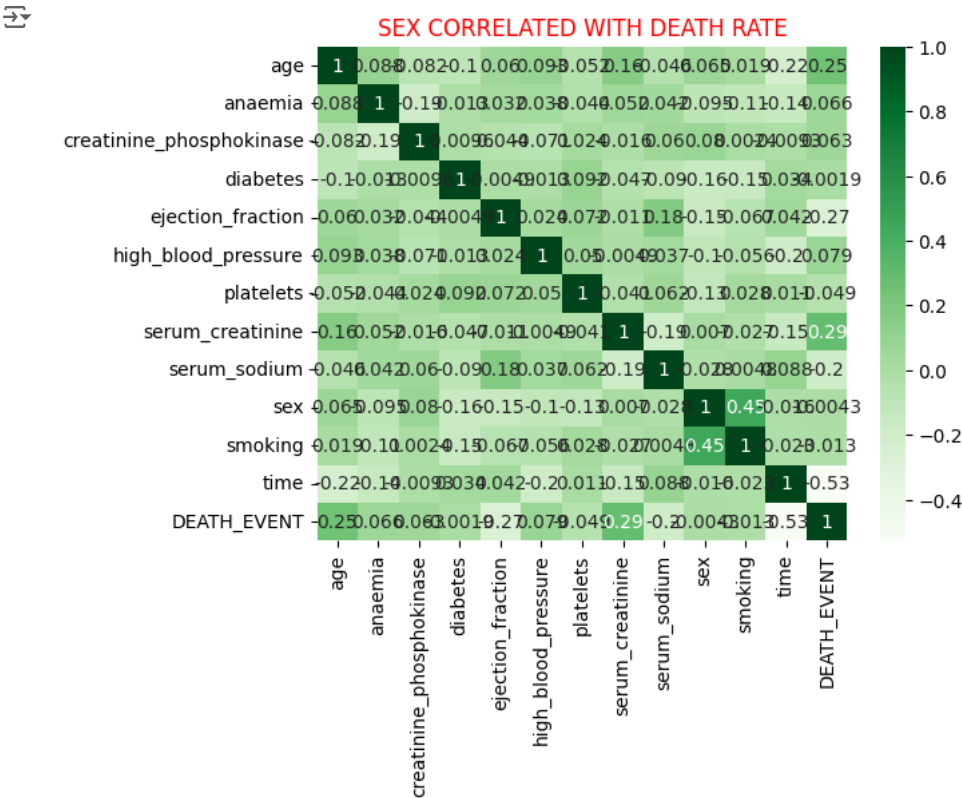
4)Death event per each sex using barplot:

```
sns.countplot(x='DEATH_EVENT',data=df)
plt.title('DEATH_EVENT',color='pink')
plt.show()
```



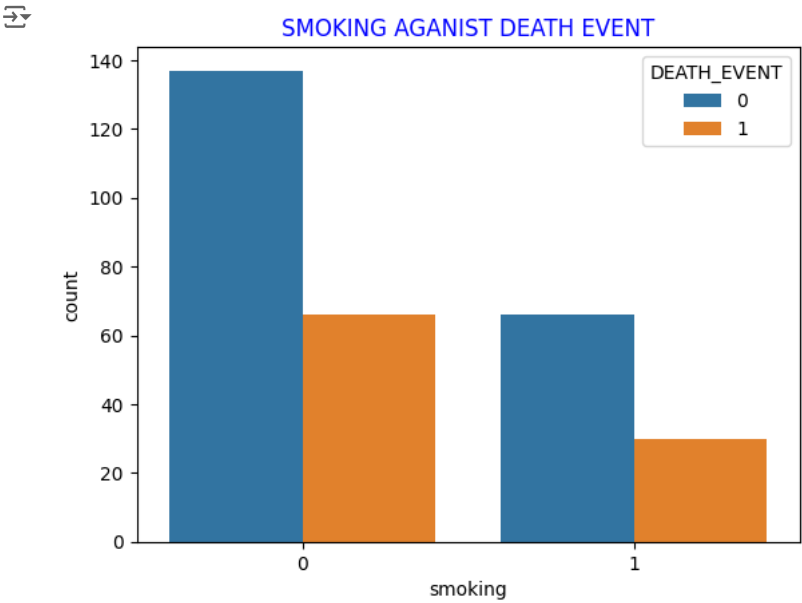
5)Sex correlated with Death rate using heatmap:

```
corr=df.corr()  
sns.heatmap(corr,annot=True,cmap='Greens')  
plt.title('SEX CORRELATED WITH DEATH RATE',color='red')  
plt.show()
```



6)Smoking against Death using Barplot:

```
sns.countplot(x='smoking',hue='DEATH_EVENT',data=df)  
plt.title('SMOKING AGAINST DEATH EVENT',color='blue')  
plt.show()
```



7)High blood pressure with age using catplot:

```
sns.catplot(x='high_blood_pressure',y='age',data=df)
plt.title('HIGH BLOOD PRSSURE',color='red')
plt.show()
```

