Heart Failure - Analysis

Introduction

About - Dataset: Cardiovascular diseases (CVDs) are the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worlwide. Heart failure is a common event caused by CVDs and this dataset contains 12 features that can be used to predict mortality by heart failure.

Most cardiovascular diseases can be prevented by addressing behavioural risk factors such as tobacco use, unhealthy diet and obesity, physical inactivity and harmful use of alcohol using population-wide strategies.

People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidaemia or already established disease) need early detection and management wherein a machine learning model can be of great help.

About - Project: This Exploratory Data Analysis project is a part of my EDA projects. In this project, we'll analyse the relationship between the different features of the heart failure patient included in this dataset namely the distribution of age among the patients, death rate, percentage of male and female patients, variation in the platelets amount, creatinine and sodium level in the blood. The graphical representation and visualisation of data using matplotlib and seaborn library in python helps us to easily understand a lot better about the dataset.

Dataset - Source:\ The dataset is obtained from Kaggle.

Please click here to know more about the dataset.

The dataset consist of column names (attributes) which doesn't provide complete information regarding the data recorded, so we have to refer to the another table / websites to see the complete information regarding the attributes (column names) including measurement units and normal level, if required.

Download the Dataset: There are several options for getting the dataset into Jupyter:

Download the CSV manually and upload it via VSCode.

Use the urlretrieve function from the urllib.request to download CSV files.

Use a helper library, e.g., opendatasets, which contains a collection of curated datasets and provides a helper function for direct download.

Initially, I used the manually download method to download the files from Kaggle without using my username and API key. Later, I uploaded the same dataset to my Github profile, to keep log of any change i made in dataset, just for my convenience.

In [ ]:
```
# Import all modules needed.
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import plotly.express as px
```

In [ ]:
```
# Set style
custom_params = {"axes.spines.right": False, "axes.spines.top": False}
sns.set_theme(style="ticks", rc=custom_params)
sns.set_style('whitegrid')
```

All Modules are Initialized.

In [ ]:
```
# load data set from csv file
df_heart = pd.read_csv('heart_failure.csv')
#print fist 5 rows of data
df_heart.head(5)
```

Out[ ]:

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platele
0	75.0	0	582	0	20	1	265000.(
1	55.0	0	7861	0	38	0	263358.(
2	65.0	0	146	0	20	0	162000.(
3	50.0	1	111	0	20	0	210000.(
4	65.0	1	160	1	20	0	327000.(

Some basic data cleaning and exploring via Pandas & Numpy.

Some info to keep in mind.

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

In [ ]:
```
df_eda = pd.DataFrame()
df_eda['age']=df_heart['age']
df_eda['anaemia']= np.where(df_heart['anaemia']<1,'No','Yes')
df_eda["creatinine_phosphokinase"] = df_heart["creatinine_phosphokinase"]
df_eda["diabetes"] = np.where(df_heart["diabetes"] < 1, "No", "Yes")
```

```
df_eda["ejection_fraction"] = df_heart["ejection_fraction"]
df_eda["high_blood_pressure"] = np.where(df_heart["high_blood_pressure"] < 1, "No", "Y
df_eda["platelets"] = df_heart["platelets"]
df_eda["serum_creatinine"] = df_heart["serum_creatinine"]
df_eda["serum_sodium"] = df_heart["serum_sodium"]
df_eda["sex"] = np.where(df_heart["sex"] < 1, "Female", "Male")
df_eda["smoking"] = np.where(df_heart["smoking"] < 1, "No", "Yes")
df_eda["death_event"] = np.where(df_heart["DEATH_EVENT"] < 1, "No", "Yes")

df_eda.head()
```

Out[ ]:

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platele
0	75.0	No	582	No	20	Yes	265000.(
1	55.0	No	7861	No	38	No	263358.(
2	65.0	No	146	No	20	No	162000.(
3	50.0	Yes	111	No	20	No	210000.(
4	65.0	Yes	160	Yes	20	No	327000.(

In [ ]:
```
# Get some info. on the dataset
df_eda.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 299 entries, 0 to 298
Data columns (total 12 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   age                       299 non-null    float64
 1   anaemia                   299 non-null    object 
 2   creatinine_phosphokinase  299 non-null    int64  
 3   diabetes                  299 non-null    object 
 4   ejection_fraction         299 non-null    int64  
 5   high_blood_pressure       299 non-null    object 
 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    object 
 10  smoking                   299 non-null    object 
 11  death_event               299 non-null    object 
dtypes: float64(3), int64(3), object(6)
memory usage: 28.2+ KB
```

In [ ]:
```
# Get some descrition of the data.
df_heart.describe()
```

Out[ ]:

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pr
count	299.000000	299.000000	299.000000	299.000000	299.000000	299.0
mean	60.833893	0.431438	581.839465	0.418060	38.083612	0.3
std	11.894809	0.496107	970.287881	0.494067	11.834841	0.4
min	40.000000	0.000000	23.000000	0.000000	14.000000	0.0
25%	51.000000	0.000000	116.500000	0.000000	30.000000	0.0
50%	60.000000	0.000000	250.000000	0.000000	38.000000	0.0
75%	70.000000	1.000000	582.000000	1.000000	45.000000	1.0
max	95.000000	1.000000	7861.000000	1.000000	80.000000	1.0

Get some description of the data.

heart_data.describe()

Data Analysed.

1. Age -> Total values are 299. The mean(AVERGAE) age is 60-61. The minimum age is 40 and the maximum age is 95. Total ages = 55
2. Sex -> We see that, most of the candiates of the data are males, up to 60% and 40% females.
3. Smoking -> We see that around 30% of the candiates do smoke, the rest fortunately don't - I WILL VISUALISE TO SEE THIS DATA ENTRY BETTERLY
4. Death? -> 30% ended up dying, the rest fortuntely survived. So this could be a possiblity that the 30% who smoked died - but to make sure of this i'll visualize our dataset.

In [ ]:
```
# shape of the Dataset
df_eda.shape
```

Out[ ]:
```
(299, 12)
```

A bit more of the inner exploration of data, fiddliling with the different columns and finding relationships between them via Pandas.

In [ ]:
```
df_eda['sex'].value_counts()
```

Out[ ]:
```
Male      194
Female    105
Name: sex, dtype: int64
```

In [ ]:
```
df_eda['high_blood_pressure'].value_counts()
```

Out[ ]:
```
No     194
Yes    105
Name: high_blood_pressure, dtype: int64
```

In [ ]:
```
df_eda['diabetes'].value_counts()
```

Out[ ]:
```
No     174
Yes    125
Name: diabetes, dtype: int64
```

In [ ]:
```
df_eda['smoking'].value_counts()
```

Out[ ]:
```
No     203
Yes     96
Name: smoking, dtype: int64
```

In [ ]:
```
df_eda['death_event'].value_counts()
```

Out[ ]:
```
No     203
Yes     96
Name: death_event, dtype: int64
```

Visualise the Data, via Matplotlib

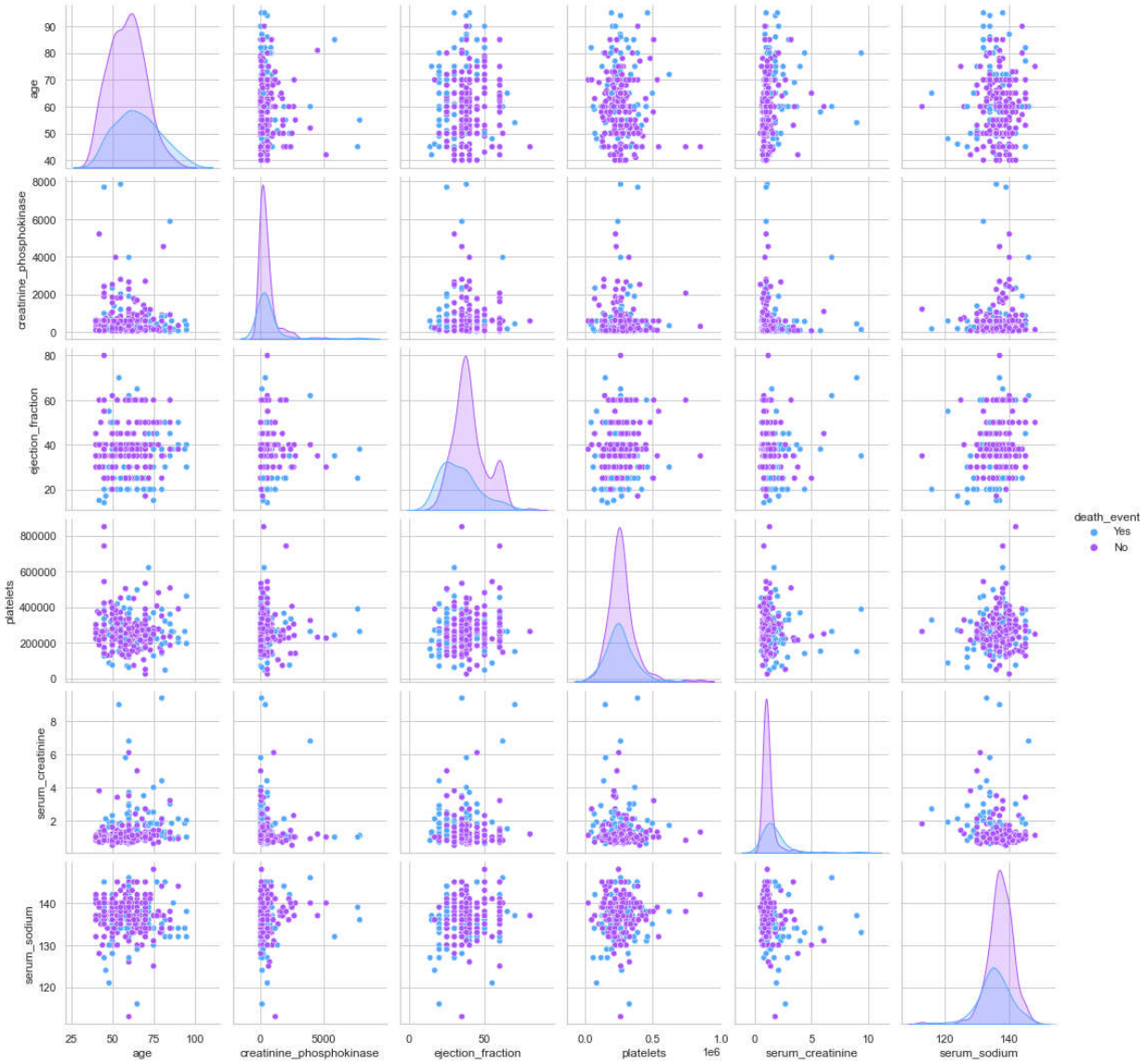
In [ ]:
```
df_eda.head()
```

Out[ ]:

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platele
0	75.0	No	582	No	20	Yes	265000.(
1	55.0	No	7861	No	38	No	263358.(
2	65.0	No	146	No	20	No	162000.(
3	50.0	Yes	111	No	20	No	210000.(
4	65.0	Yes	160	Yes	20	No	327000.(

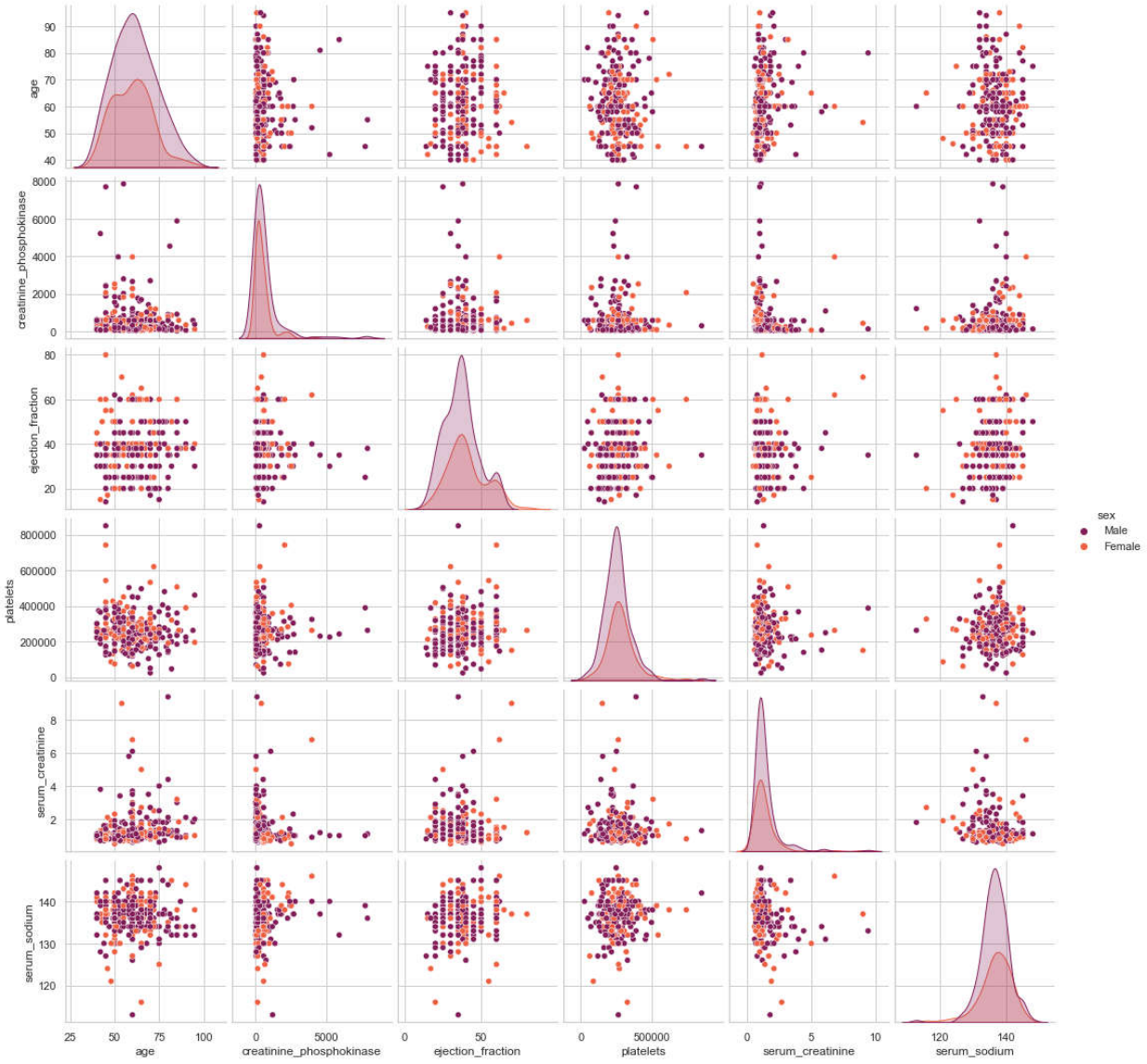
Relationship of the whole datset (with relation to death event)

In [ ]:
```
# Showing the relationship of the whole datset ( with relation to death event)

# Plot
from matplotlib.pyplot import title


sns.pairplot(df_eda,hue='death_event',palette ='cool')
# Show
plt.show()
```



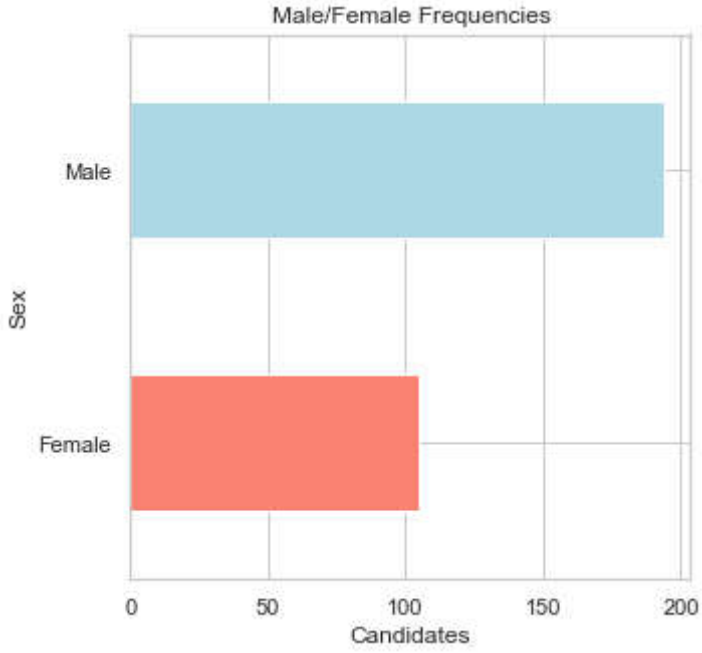
Relationship of the whole dataset (with relation to sex)

In [ ]:
```
# Showing the relationship of the whole dataset (with relation to sex)
# Plot
sns.pairplot(df_eda,hue='sex',palette='rocket')
# Show
plt.show()
```



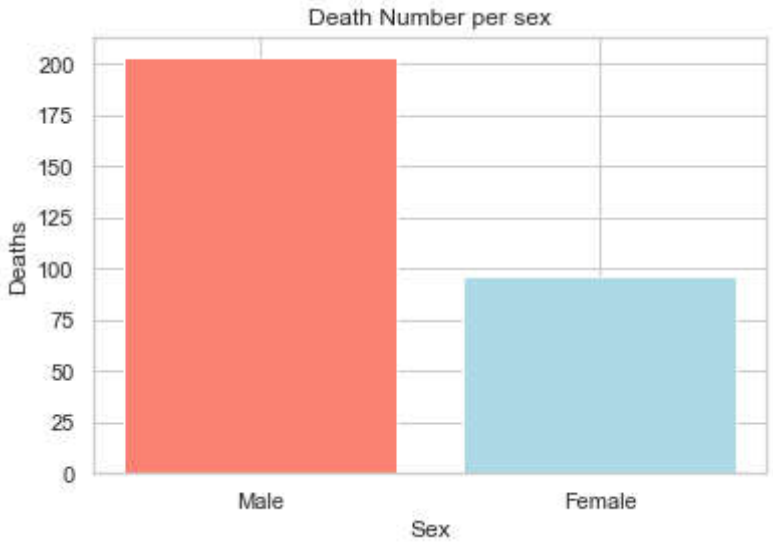
Relationship between categoric variable "sex" and its frequency

In [ ]:
```
# Showing the relationship between categoric variable "sex" and its frequency
plt.figure(figsize=(5,5))
figure_1 = df_eda["sex"].value_counts(ascending = True).plot.barh(color=["salmon", "li
plt.title("Male/Female Frequencies")
plt.ylabel("Sex")
plt.xlabel("Candidates")
plt.show()
```



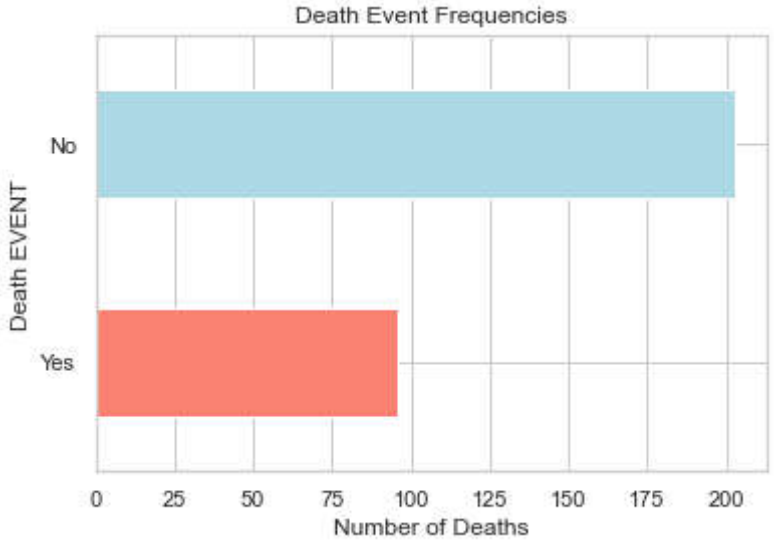
Death event per each sex

In [ ]:
```
# Death event per each sex
figure_3 = plt.bar(df_eda["sex"].value_counts().index, df_eda["death_event"].value_cou
plt.title("Death Number per sex")
plt.xlabel("Sex")
plt.ylabel("Deaths")
plt.show()
```



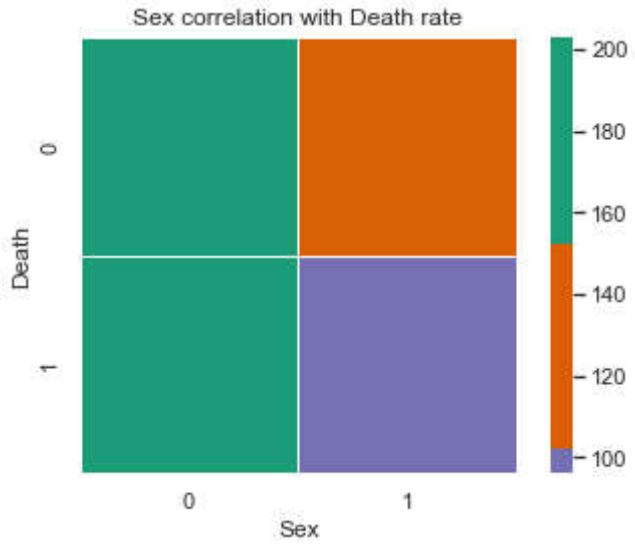
Relationship between categoric variable "death_event" and its frequency

In [ ]:
```
# Showing the relationship between categoric variable "death_event" and its frequency
figure_2 = df_eda["death_event"].value_counts(ascending = True).plot.barh(color=["salm
plt.title("Death Event Frequencies")
plt.ylabel("Death EVENT")
plt.xlabel("Number of Deaths")
plt.show()
```



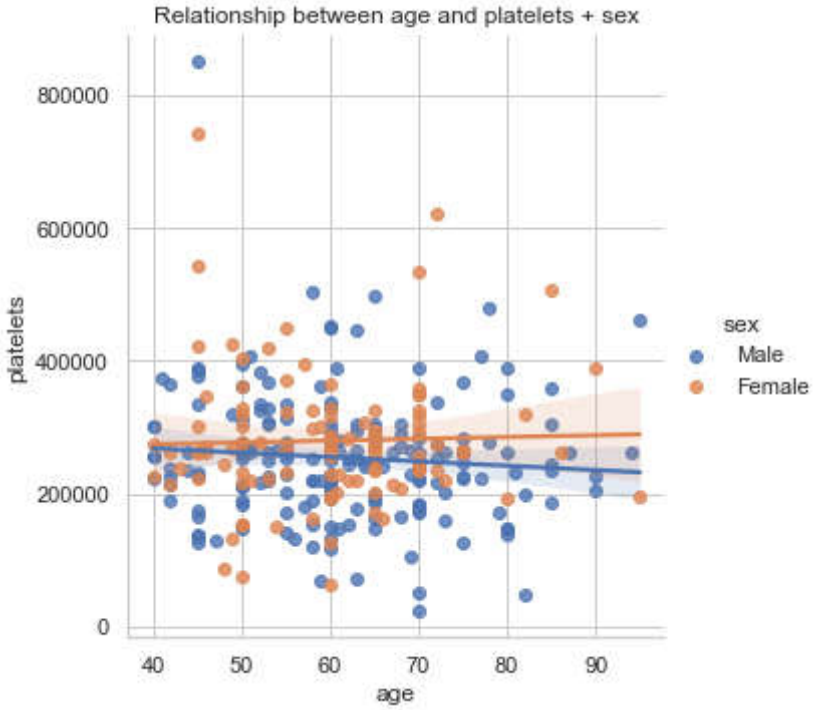
Sex correlation with Death rate

In [ ]:
```
# Sex correlation with Death rate
plt.title('Sex correlation with Death rate')
sns.heatmap((df_heart['sex'].value_counts(),df_heart['DEATH_EVENT'].value_counts()),cm
linewidths=.5,square=True,center=0)
plt.xlabel('Sex') # Female -0 || Male - 1
plt.ylabel('Death') # No -0 || Yes - 1
plt.show()
```



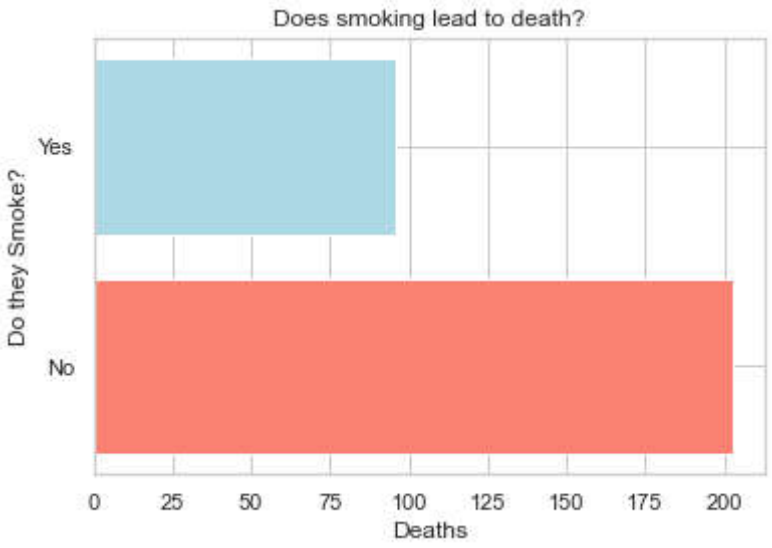
Relationship between age and platelets & sex

In [ ]:
```
# Relationship between age and platelets + sex
sns.lmplot(x = "age",
           y = "platelets",
           hue = "sex",
           data = df_eda).set(title='Relationship between age and platelets + sex')
plt.show()
```



Smoking against Death

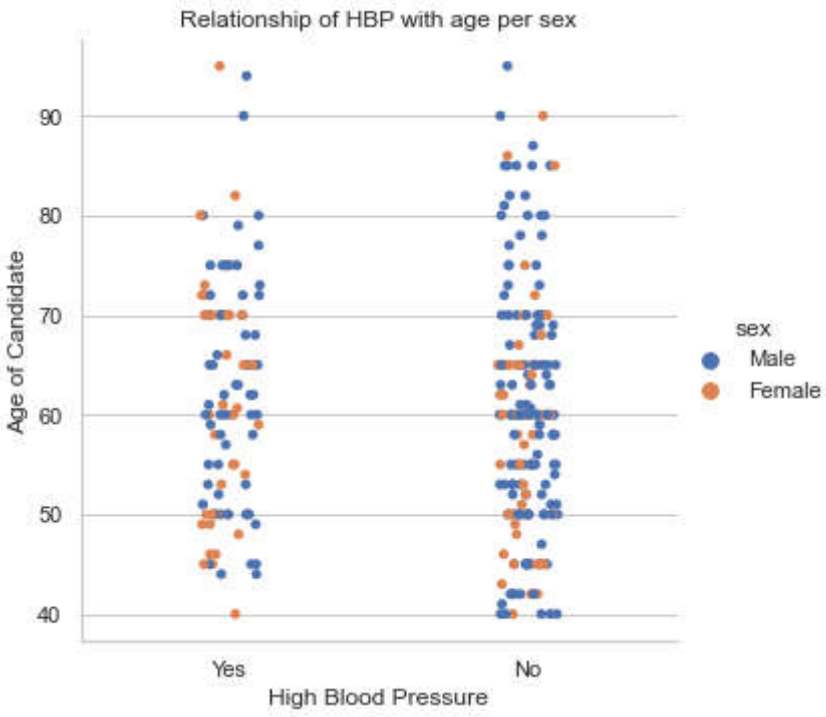
In [ ]:
```
# Smoking against Death
figure_4 = plt.barh(df_eda["smoking"].value_counts().index,df_eda["death_event"].value
plt.title("Does smoking lead to death?")
plt.xlabel("Deaths")
plt.ylabel("Do they Smoke?")
plt.show()
```



High blood pressure with age

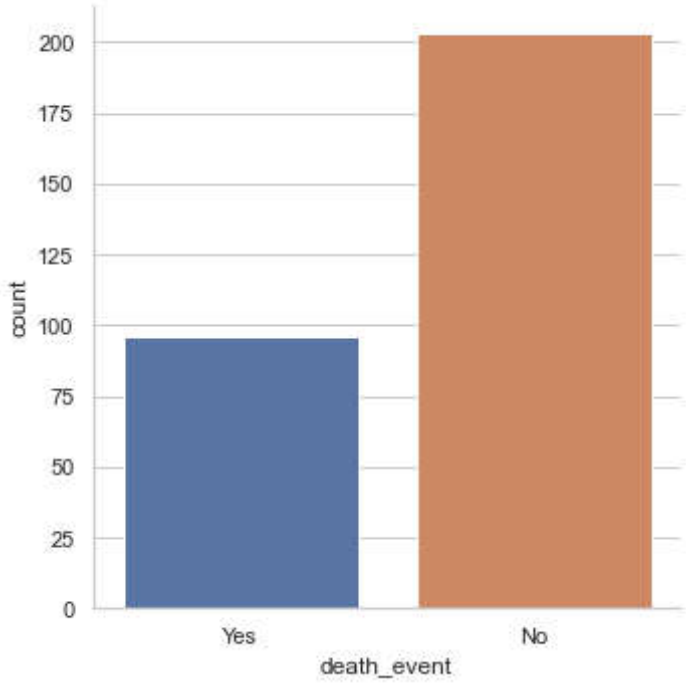
In [ ]:
```
# High blood pressure with age
sns.catplot(
    x = "high_blood_pressure",
    y = "age",
    hue = "sex",
    data = df_eda
```

```
).set(xlabel="High Blood Pressure", ylabel="Age of Candidate", title="Relationship of
plt.show()
```



Value Count for the deaths

In [ ]:
```
# Value Count for the deaths
sns.catplot(x='death_event',
            data=df_eda,
            kind="count")
plt.show()
```



In [ ]: