Double-click (or enter) to edit

```
#Capstone Project:3- Exploratory Data Analysis Projects
#HEALTH FAILURE ANALYSIS


#MODULES NEEDED: Numpy, Pandas, Matplotlip, Seaborn

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

#load the data
heart_df=pd.read_csv("/content/heart_failure_clinical_records_dataset.csv")
print(heart_df)
```

```
      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]
```

```
#load the data in first 5 rows
heart_df.head()
```

	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

```
#Load the data in last 5 rows
heart_df.tail()
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
294	62.0	0	61	1	38	1	155000.0	1.1	143	1	1	270	0
295	55.0	0	1820	0	38	0	270000.0	1.2	139	0	0	271	0
296	45.0	0	2060	1	60	0	742000.0	0.8	138	0	0	278	0
297	45.0	0	2413	0	38	0	140000.0	1.4	140	1	1	280	0
298	50.0	0	196	0	45	0	395000.0	1.6	136	1	1	285	0

```
#cleaning of data for checking null values
heart_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

```
heart_df.isnull().any().any()
```

```
False
```

```
#Info on the dataset
heart_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
```

```
#Some dataset was cleaning

heart_data = pd.DataFrame()

heart_data["age"] = heart_df["age"]
heart_data["anaemia"] = np.where(heart_df["anaemia"]< 1,"No","yes")
heart_data["diabetes"] = np.where(heart_df["diabetes"]< 1,"No","yes")
heart_data["high_blood_pressure"] = np.where(heart_df["high_blood_pressure"]< 1,"No","yes")
heart_data["platelets"] = heart_df["platelets"]
heart_data["serum_creatinine"] = heart_df["serum_creatinine"]
heart_data["serum_sodium"] = heart_df["serum_sodium"]
heart_data["sex"] = np.where(heart_df["sex"]< 1,"Female","Male")
heart_data["smoking"] = np.where(heart_df["smoking"]< 1,"No","yes")
heart_data["DEATH_EVENT"] = np.where(heart_df["DEATH_EVENT"]< 1,"No","yes")
heart_data.head()
```

	age	anaemia	diabetes	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	DEATH_EVENT
0	75.0	No	No	yes	265000.00	1.9	130	Male	No	yes
1	55.0	No	No	No	263358.03	1.1	136	Male	No	yes
2	65.0	No	No	No	162000.00	1.3	129	Male	yes	yes
3	50.0	yes	No	No	210000.00	1.9	137	Male	No	yes
4	65.0	yes	yes	No	327000.00	2.7	116	Female	No	yes

```
#some info on the dataset

heart_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 299 entries, 0 to 298
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   age                   299 non-null    float64
 1   anaemia               299 non-null    object 
 2   diabetes              299 non-null    object 
 3   high_blood_pressure   299 non-null    object 
 4   platelets             299 non-null    float64
 5   serum_creatinine      299 non-null    float64
 6   serum_sodium          299 non-null    int64  
 7   sex                   299 non-null    object 
 8   smoking               299 non-null    object 
 9   DEATH_EVENT           299 non-null    object 
dtypes: float64(3), int64(1), object(6)
memory usage: 23.5+ KB
```

```
#Remove un-needed data - time column

heart_df.drop(['time'], axis = 1, inplace = True)


# describe of the data

heart_df.describe()
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	DEATH_EVENT
count	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.000000	299.00000	299.000000	299.000000	299.00000	299.00000
mean	60.833893	0.431438	581.839465	0.418060	38.083612	0.351171	263358.029264	1.39388	136.625418	0.648829	0.32107	0.32107
std	11.894809	0.496107	970.287881	0.494067	11.834841	0.478136	97804.236869	1.03451	4.412477	0.478136	0.46767	0.46767
min	40.000000	0.000000	23.000000	0.000000	14.000000	0.000000	25100.000000	0.50000	113.000000	0.000000	0.00000	0.00000
25%	51.000000	0.000000	116.500000	0.000000	30.000000	0.000000	212500.000000	0.90000	134.000000	0.000000	0.00000	0.00000
50%	60.000000	0.000000	250.000000	0.000000	38.000000	0.000000	262000.000000	1.10000	137.000000	1.000000	0.00000	0.00000
75%	70.000000	1.000000	582.000000	1.000000	45.000000	1.000000	303500.000000	1.40000	140.000000	1.000000	1.00000	1.00000
max	95.000000	1.000000	7861.000000	1.000000	80.000000	1.000000	850000.000000	9.40000	148.000000	1.000000	1.00000	1.00000

```
#shape of the data

heart_data.shape
```

```
(299, 10)
```

```
#Value count of the data
#"ANAEMIA"
#"HIGH_BLOOD_PRESSURE"
#"DIABETES"
#"SMOKING"
#"DEATH_EVENT"


#Anaemia

heart_data['anaemia'].value_counts()
```

anaemia	count
No	170
yes	129

dtype: int64

```
heart_data['high_blood_pressure'].value_counts()
```

high_blood_pressure	count
No	194
yes	105

dtype: int64

```
heart_data['diabetes'].value_counts()
```

diabetes	count
No	174
yes	125

dtype: int64

```
heart_data['smoking'].value_counts()
```

smoking	count
No	203
yes	96

dtype: int64

```
heart_data['DEATH_EVENT'].value_counts()
```

DEATH_EVENT	count
No	203
yes	96

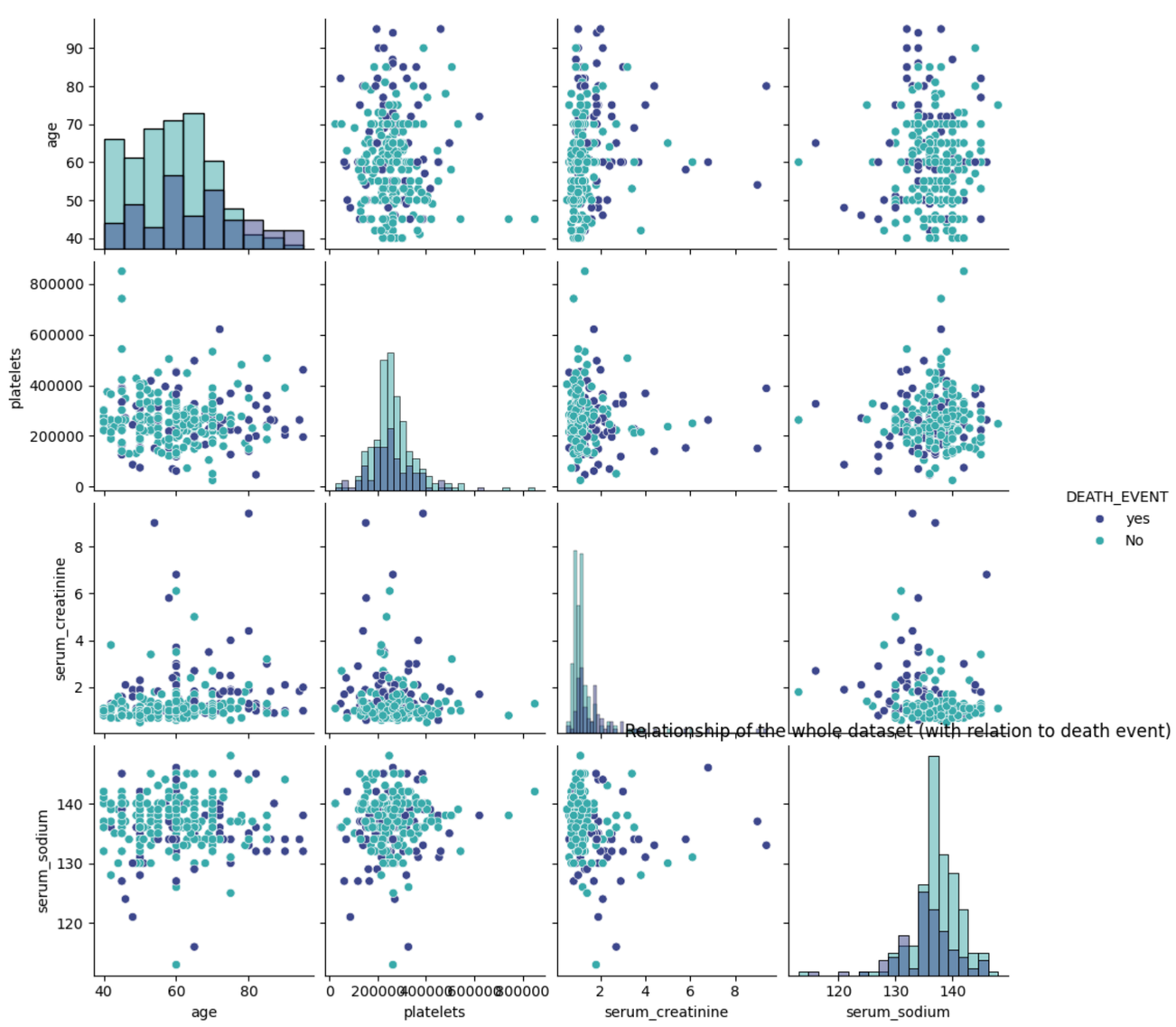
dtype: int64

```
#Visualization the data Matplotlib & Seaborn
#the relationship of the whole dataset (with relation to death event) using pairplot
import matplotlib.pyplot as plt
import seaborn as sns
heart_data.head()
```

	age	anaemia	diabetes	high_blood_pressure	platelets	serum_creatinine	serum_sodium	sex	smoking	DEATH_EVENT
0	75.0	No	No	yes	265000.00	1.9	130	Male	No	yes
1	55.0	No	No	No	263358.03	1.1	136	Male	No	yes
2	65.0	No	No	No	162000.00	1.3	129	Male	yes	yes
3	50.0	yes	No	No	210000.00	1.9	137	Male	No	yes
4	65.0	yes	yes	No	327000.00	2.7	116	Female	No	yes

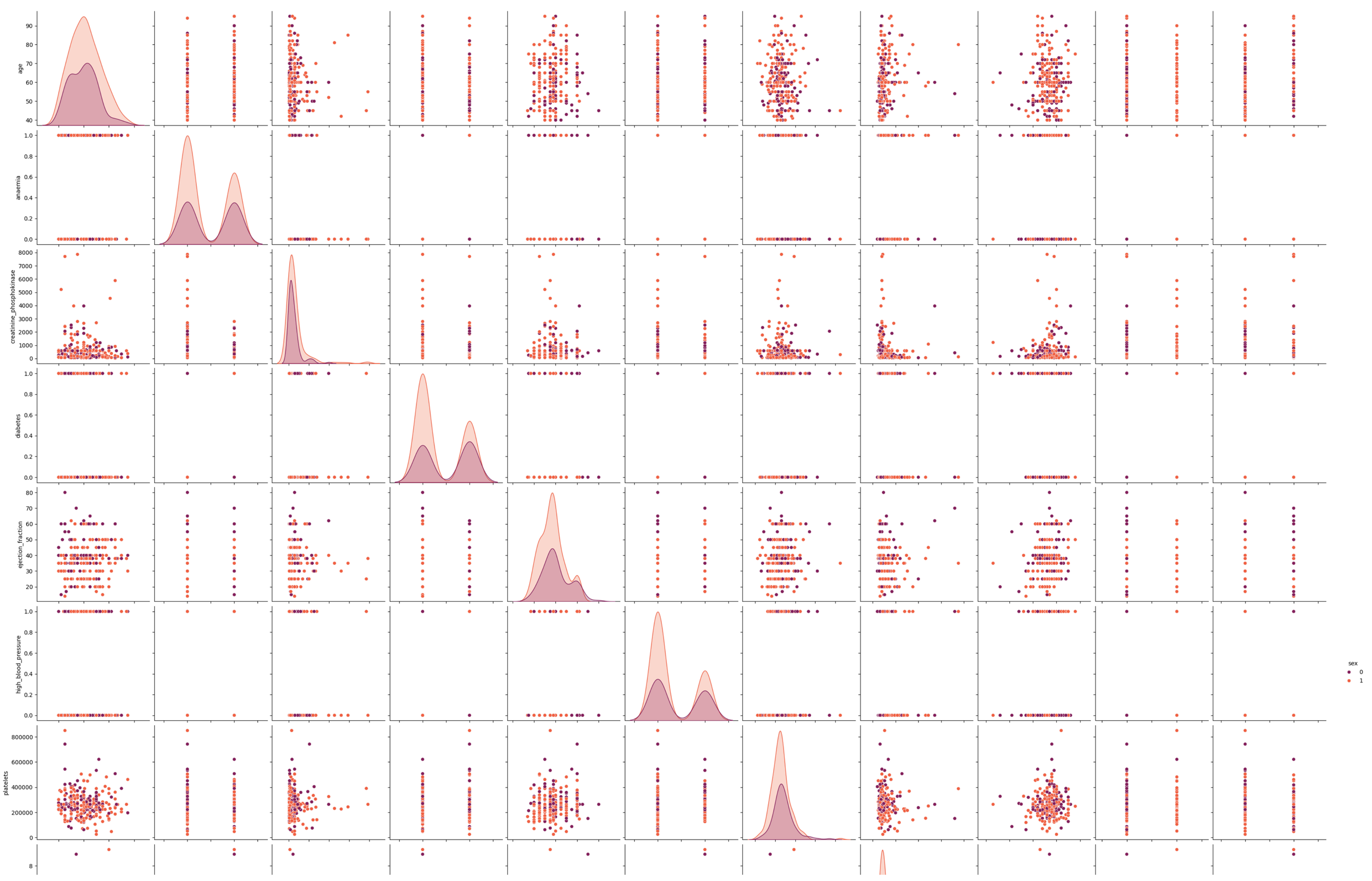
```
#Show the relationship of the whole dataset (with relation to death event) using pairplot.
#plot
sns.pairplot(heart_data, hue = 'DEATH_EVENT',diag_kind="hist", palette='mako')

#add title

plt.title("Relationship of the whole dataset (with relation to death event)")

#add X-axis label & Y-axis label
```

```
plt.xlabel("age")
plt.ylabel("platelets")

#show
plt.show()
```



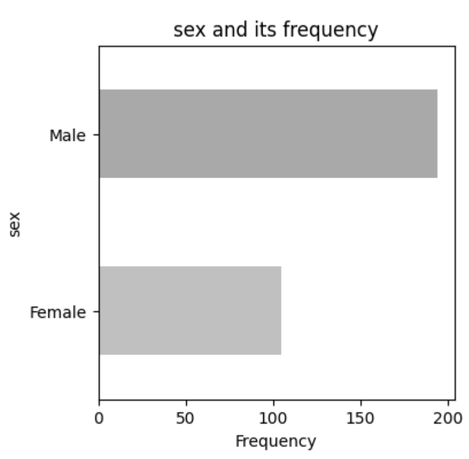
Relationship of the whole dataset (with relation to death event)

```
#Show the pairplot with heart failure dataset
sns.pairplot(heart_df, hue="sex",palette="rocket" , height=3)

# add title
plt.title("Pairplot with heart failure dataset")

#add X-axis label & Y-axis label
plt.xlabel("age")
plt.ylabel("platelets")


# Show
plt.show()
```

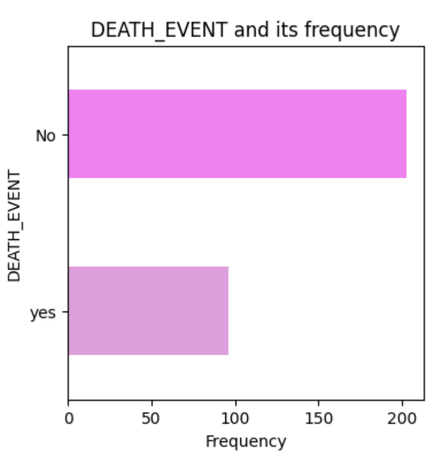



```
#Show the relationship between categoric variable "sex" and its frequency using


#plot
plt.figure(figsize=(4,4))
figure = heart_data["sex"].value_counts(ascending = True).plot.barh(color=['silver','darkgray'])


#add title
plt.title("sex and its frequency")

#add X-axis label & Y-axis label
plt.xlabel("Frequency")
plt.ylabel("sex")
```

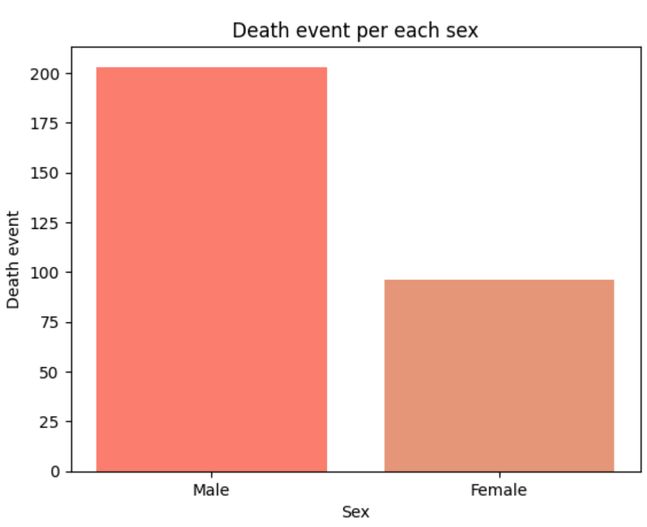
```
#show
plt.show()
```




```
 #Showing the relationship between categoric variable "death_event" and its frequency using bar plot.

#plot
plt.figure(figsize=(4,4))
figure = heart_data["DEATH_EVENT"].value_counts(ascending = True).plot.barh(color=['plum','violet'])

#add title
plt.title("DEATH_EVENT and its frequency")

#add x-axis label & y-axis label
plt.xlabel("Frequency")
plt.ylabel("DEATH_EVENT")

#show
plt.show()
```

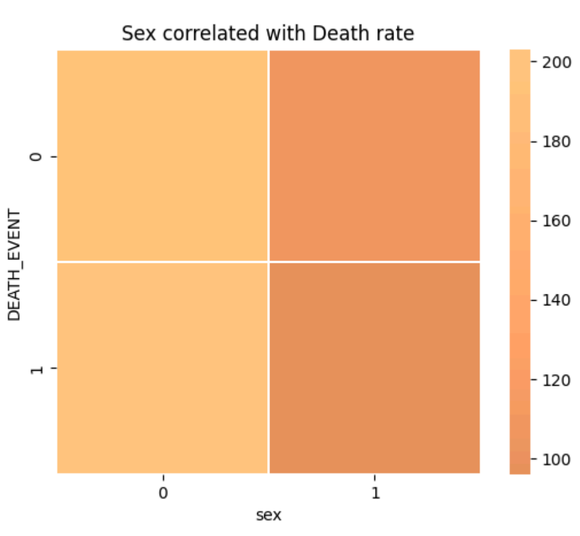



```
# Death event per each sex using bar plot
#plot
plt.bar(heart_data['sex'].value_counts().index, heart_data['DEATH_EVENT'].value_counts(),color=['salmon','darksalmon'])

#add title
plt.title(' Death event per each sex ')

#add x-axis label & y-axis label
plt.xlabel('Sex')
plt.ylabel('Death event')

#show
plt.show()
```

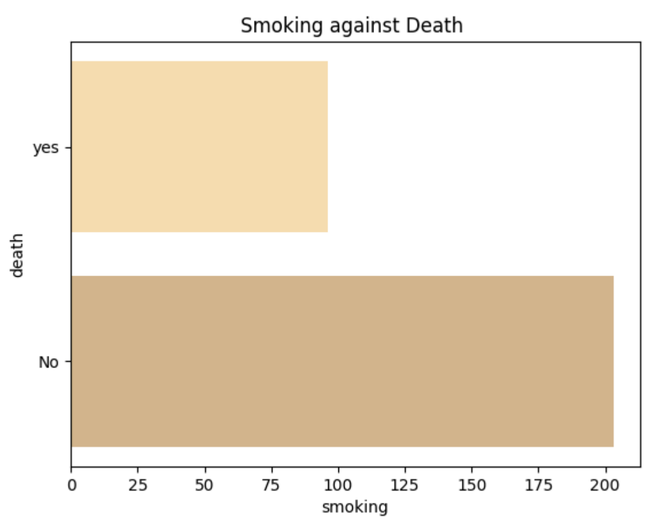


Double-click (or enter) to edit

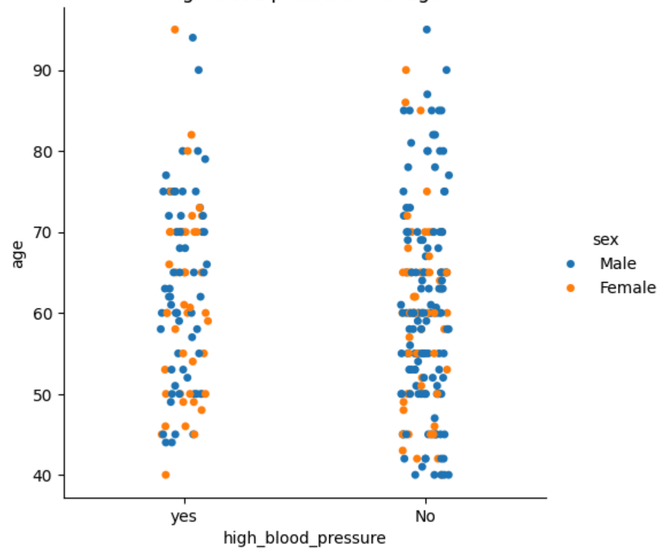
```
#Sex correlated with Death rate (use heatmap)
#plot
sns.heatmap((heart_data['sex'].value_counts(),heart_data['DEATH_EVENT'].value_counts()),cmap='copper',linewidth=.2,square=True,center=0)

#add title
plt.title('Sex correlated with Death rate')

#x-axis &y-axis label
plt.xlabel('sex')
plt.ylabel('DEATH_EVENT')

#show
plt.show()
```



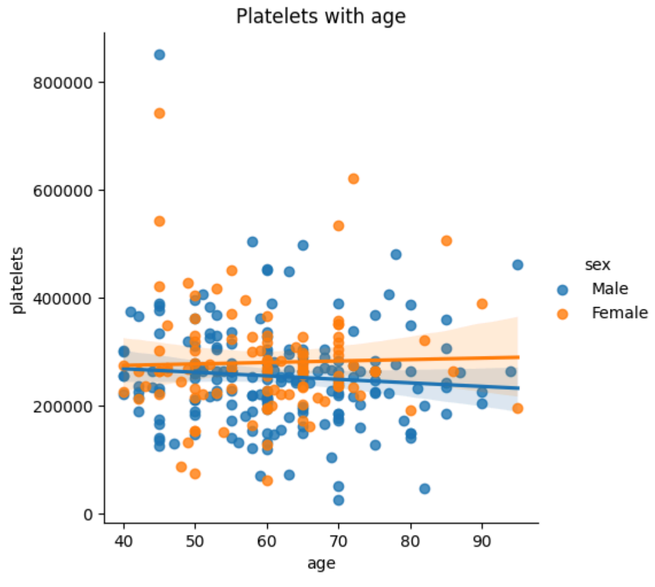

```
#Smoking against Death using bar plot
plt.barh(heart_data["smoking"].value_counts().index,heart_data["DEATH_EVENT"].value_counts(), color=['tan','wheat'])

#title
plt.title("Smoking against Death ")

#x-axis & y-axis label
plt.xlabel("smoking")
plt.ylabel("death")

#show
plt.show()
```



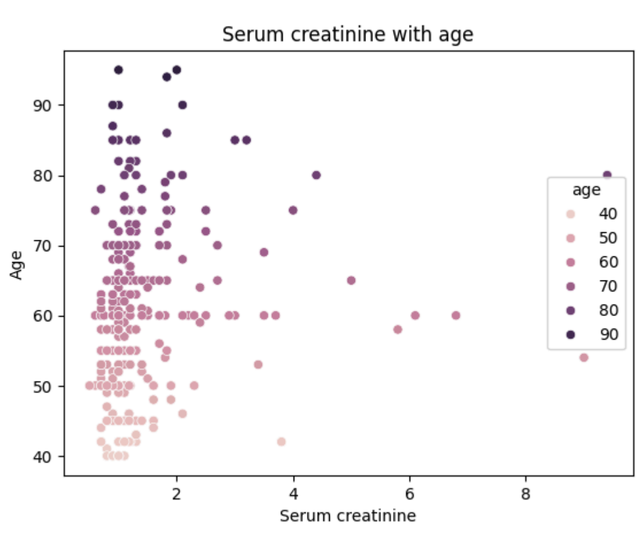

```
#High blood pressure with age using catplot.
#plot
sns.catplot(data=heart_data,x = "high_blood_pressure",y = "age", hue = "sex",)

#add title
plt.title("High blood pressure with age ")

#add x-axis label & y-axis label
plt.xlabel("high_blood_pressure")
plt.ylabel("age")

#show
plt.show()
```



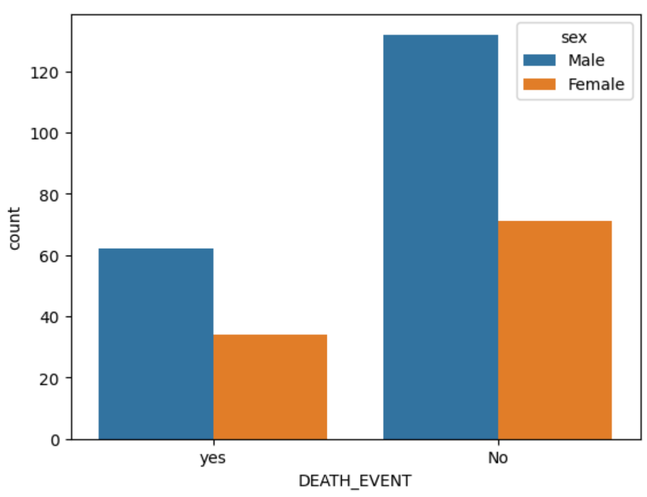

```
#Show the lineplot age with  platelets

sns.lmplot(x='age',y='platelets',hue='sex',data=heart_data)

#add tltle
plt.title("Platelets with age")

#x-axis & y-axis label
plt.xlabel("age")
plt.ylabel("platelets")

#show
plt.show()
```




```
#show the scatterplot serum_creatinine with age
#plot
sns.scatterplot(x='serum_creatinine',y='age',hue='age',data=heart_data)

#add title
plt.title("Serum creatinine with age")

#add x-axis label & y-axis label
plt.xlabel("Serum creatinine")
plt.ylabel("Age")

#show
plt.show()
```




```
#show countplot with death event for heart failure analysis
#plot

sns.countplot(data=heart_data,x='DEATH_EVENT',hue='sex')

#add title
plt.title=("Which gender was more DEATH_EVENT?")

#add x-axis label & y-axis label
plt.xlabel=("DEATH_EVENT")
plt.ylabel=("count")

#show
plt.show()
```




```
#show countplot with age of heart failure analysis
#plot
sns.countplot(data=heart_data,x='age',hue='sex')

#add title
plt.title=("Which gender was more age in death event?")
#add x-axis & y-axis label
plt.xlabel=("age")
plt.ylabel=("count")
plt.show()
```