EDA on Haberman Dataset

OBJECTIVE: To predict whether the paitient will survive after 5 years of operation for give age and operation year and no of axil nodes.

In [2]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv('haberman.csv')#haberman data is loaded into dataframe

df.columns=['age','op_year','axil','survival']#giving names to columns
df.head()#data displayonly first 5
```

Out[2]:

	age	op_year	axil	survival
0	30	62	3	1
1	30	65	0	1
2	31	59	2	1
3	31	65	4	1
4	33	58	10	1

In [3]:

```
df.shape #size of data
```

Out[3]:

```
(305, 4)
```

In [4]:

```
#changing the survival status from 1 and to 2 as yes and no
df['survival']=df['survival'].map({1:'Alive',2:'Died'})
df['survival']=df['survival'].astype('category')
df['survival'].value_counts()#counts data points for each class
```

Out[4]:

```
Alive    224
Died      81
Name: survival, dtype: int64
```

In [5]:

```
df.info()#no,of data points for each variable
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 305 entries, 0 to 304
Data columns (total 4 columns):
age         305 non-null int64
op_year     305 non-null int64
axil        305 non-null int64
survival    305 non-null category
dtypes: category(1), int64(3)
memory usage: 7.6 KB
```

In [6]:

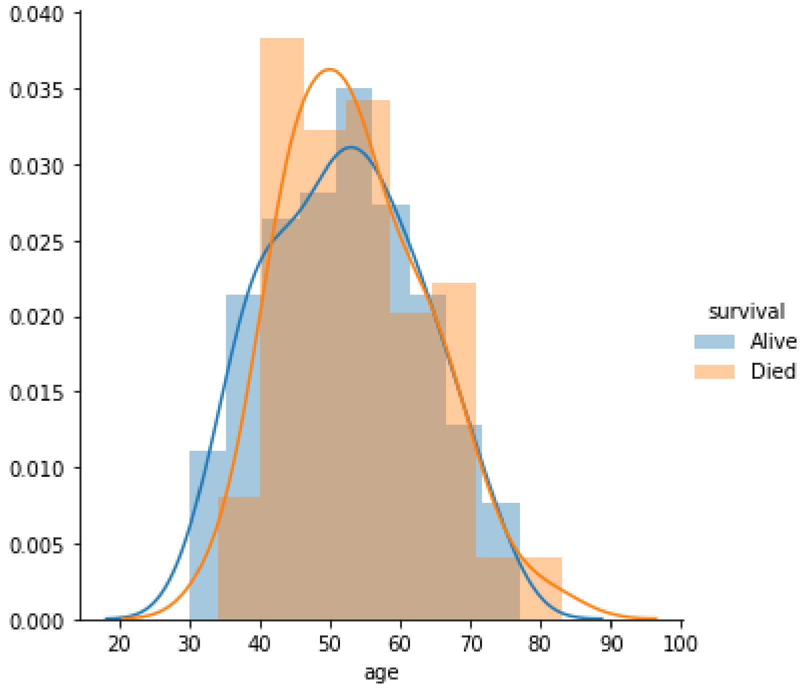
```
#Statstical observations
df.describe()
```

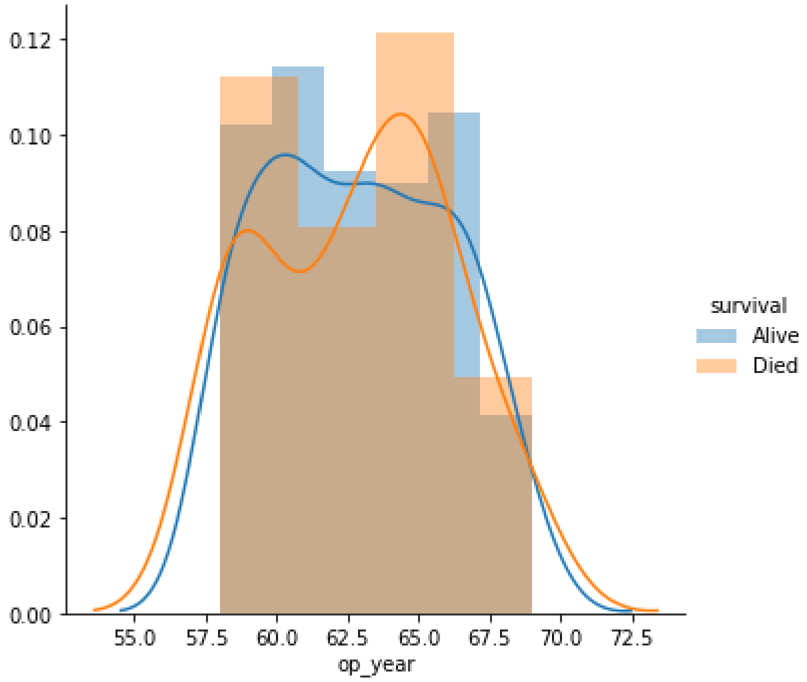
Out[6]:

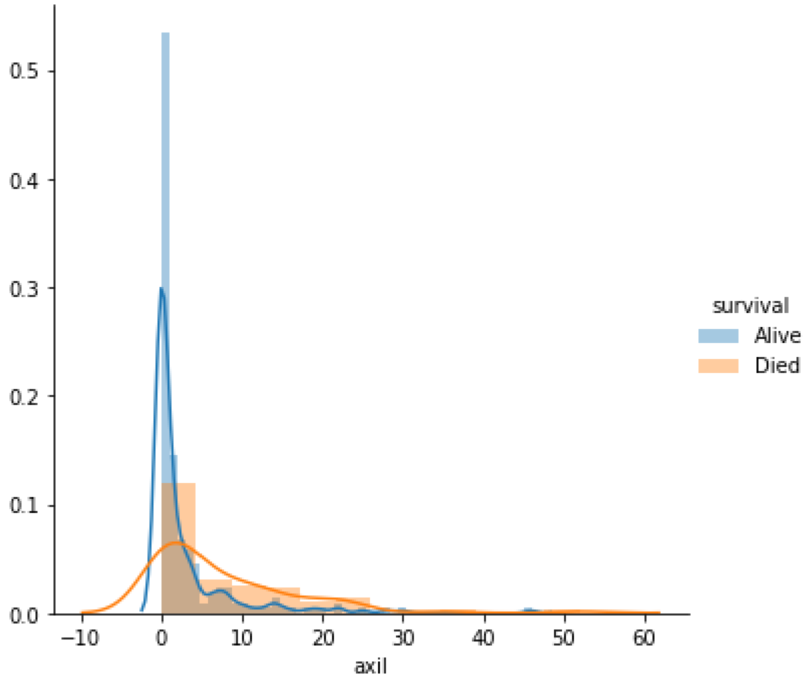
	age	op_year	axil
count	305.000000	305.000000	305.000000
mean	52.531148	62.849180	4.036066
std	10.744024	3.254078	7.199370
min	30.000000	58.000000	0.000000
25%	44.000000	60.000000	0.000000
50%	52.000000	63.000000	1.000000
75%	61.000000	66.000000	4.000000
max	83.000000	69.000000	52.000000

In [7]:

```
#univariate
sns.FacetGrid(df, hue="survival", height=5).map(sns.distplot, "age").add_legend();
plt.show()
sns.FacetGrid(df, hue="survival", height=5).map(sns.distplot, "op_year").add_legend
plt.show()
sns.FacetGrid(df, hue="survival", height=5).map(sns.distplot, "axil").add_legend();
plt.show()
```

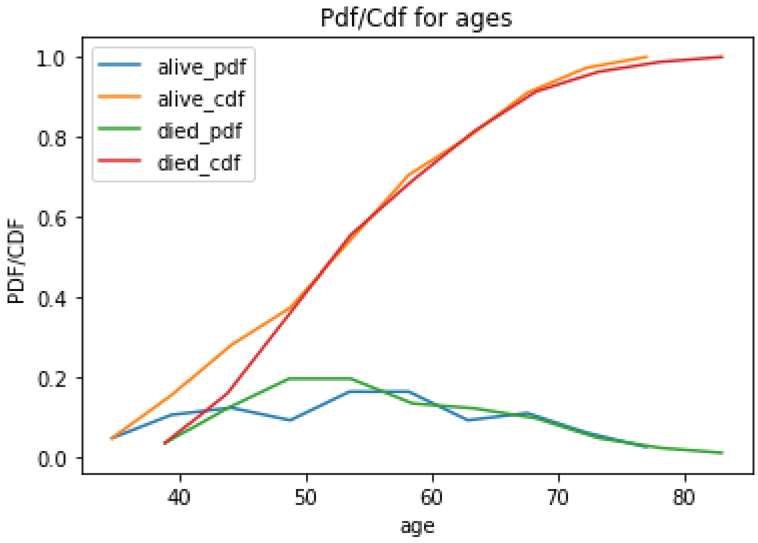


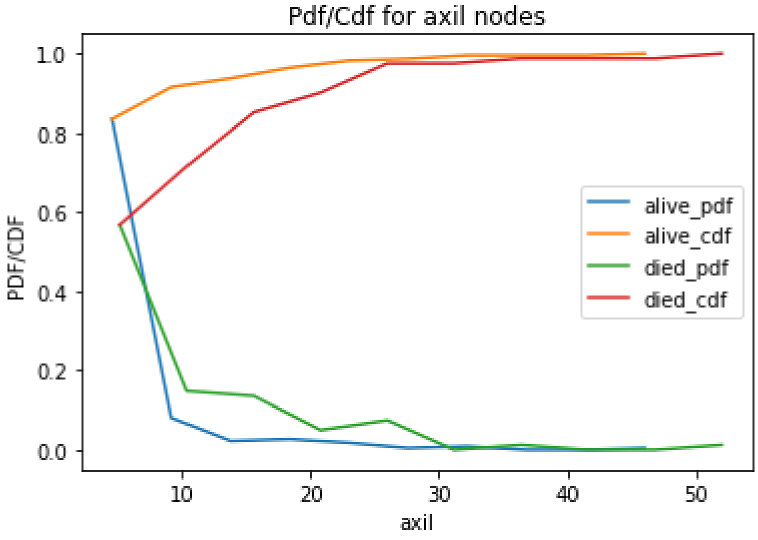


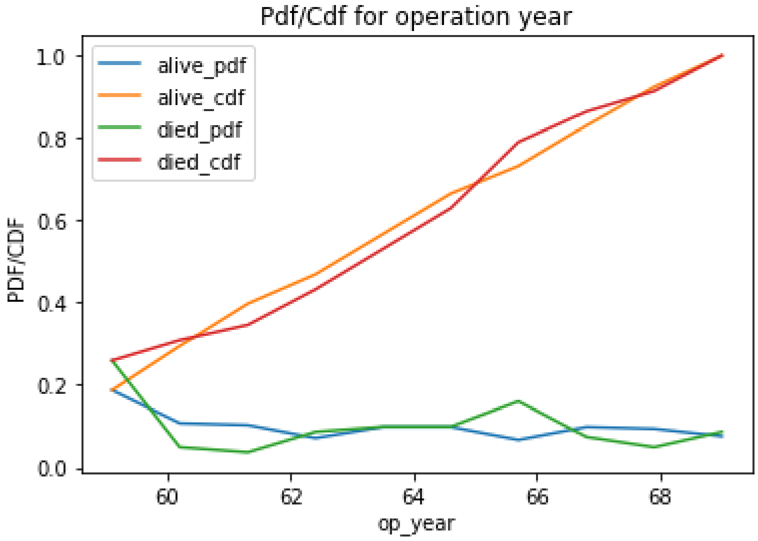

0.5
0.4
0.3
0.2
0.1
0.0
−10
0
10
20
30
40
50
60
axil
survival
Alive
Died

In [8]:

```
#CDF/PDF
#PDF/CDF of age
df_alive=df.loc[df['survival']=="Alive"]
df_died=df.loc[df['survival']=="Died"]
counts,bin_edges=np.histogram(df_alive['age'],bins=10,density=True)
pdf=counts/sum(counts)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,label='alive_pdf')
plt.plot(bin_edges[1:],cdf,label='alive_cdf')


counts,bin_edges=np.histogram(df_died['age'],bins=10,density=True)
pdf=counts/sum(counts)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,label="died_pdf")
plt.plot(bin_edges[1:],cdf,label="died_cdf")
plt.xlabel('age')
plt.ylabel('PDF/CDF')
plt.title('Pdf/Cdf for ages')
plt.legend()
plt.show()


#CDF/PDF of axil nodes
counts,bin_edges=np.histogram(df_alive['axil'],bins=10,density=True)
pdf=counts/sum(counts)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,label='alive_pdf')
plt.plot(bin_edges[1:],cdf,label='alive_cdf')

counts,bin_edges=np.histogram(df_died['axil'],bins=10,density=True)
pdf=counts/sum(counts)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,label="died_pdf")
plt.plot(bin_edges[1:],cdf,label="died_cdf")
plt.xlabel('axil')
plt.ylabel('PDF/CDF')
plt.title('Pdf/Cdf for axil nodes')
plt.legend()
plt.show()


#CDF/PDF of year of operation
counts,bin_edges=np.histogram(df_alive['op_year'],bins=10,density=True)
pdf=counts/sum(counts)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,label='alive_pdf')
plt.plot(bin_edges[1:],cdf,label='alive_cdf')

counts,bin_edges=np.histogram(df_died['op_year'],bins=10,density=True)
pdf=counts/sum(counts)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,label="died_pdf")
plt.plot(bin_edges[1:],cdf,label="died_cdf")
plt.xlabel('op_year')
plt.ylabel('PDF/CDF')
plt.title('Pdf/Cdf for operation year')
plt.legend()

plt.show()
```

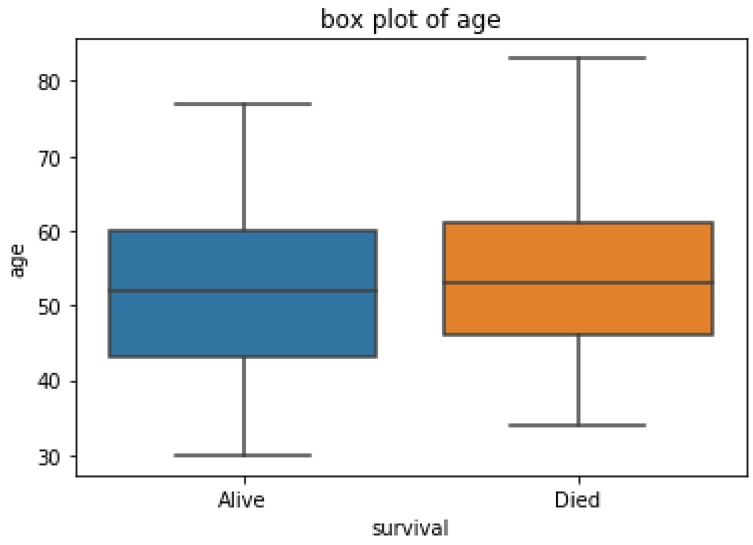


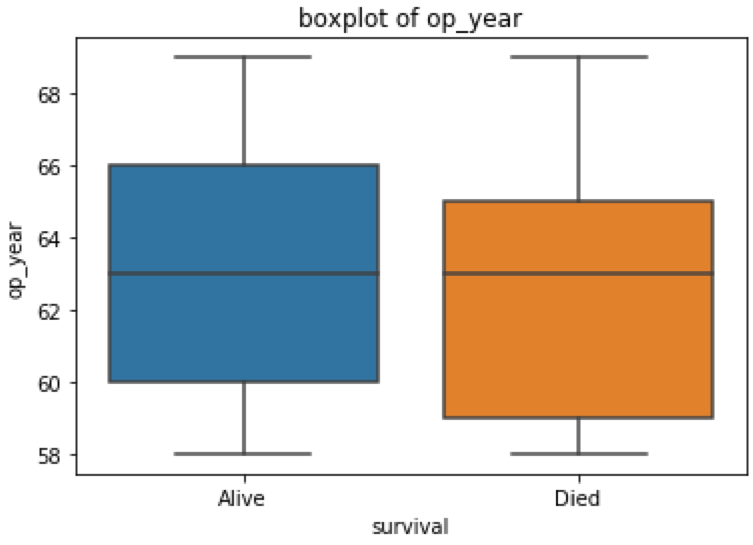


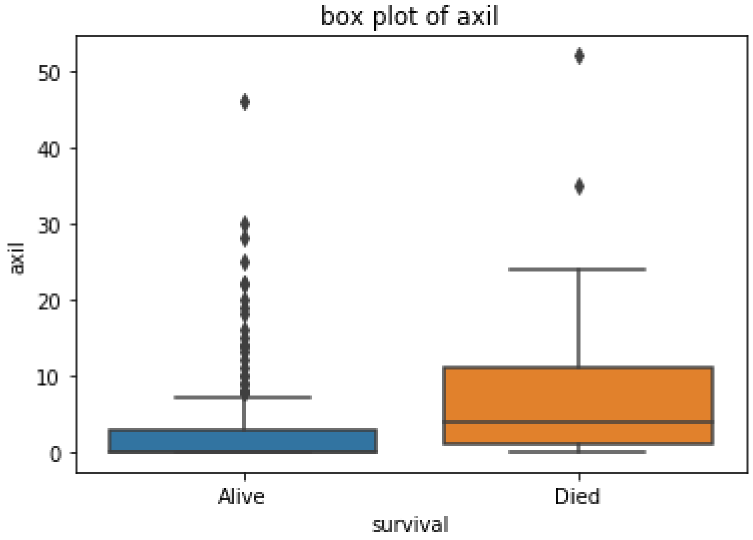

Pdf/Cdf for operation year
alive_pdf
alive_cdf
died_pdf
died_cdf
PDF/CDF
1.0
0.8
0.6
0.4
0.2
0.0
60
62
64
66
68
op_year

In [9]:

```
# Box Plots
sns.boxplot(x="survival",y="age",data=df)
plt.title('box plot of age')
plt.show()
sns.boxplot(x="survival",y="op_year",data=df)
plt.title('boxplot of op_year')
plt.show()
sns.boxplot(x="survival",y="axil",data=df)
plt.title('box plot of axil')
plt.show()
```

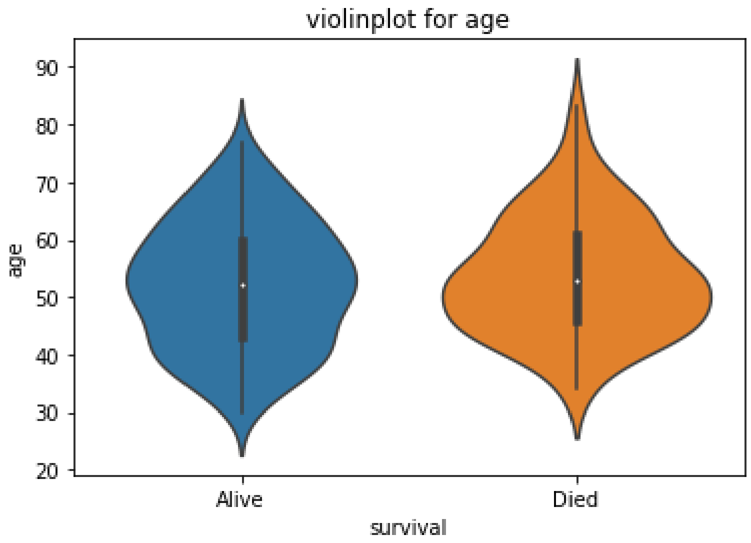

box plot of age

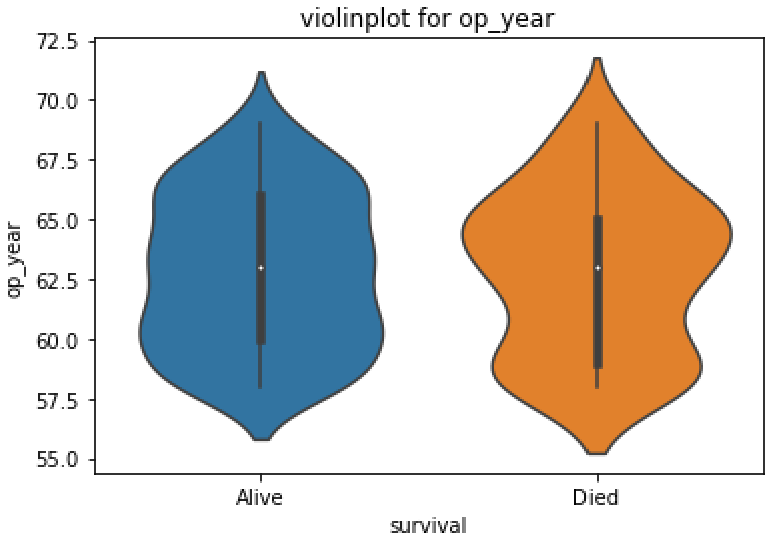

boxplot of op_year

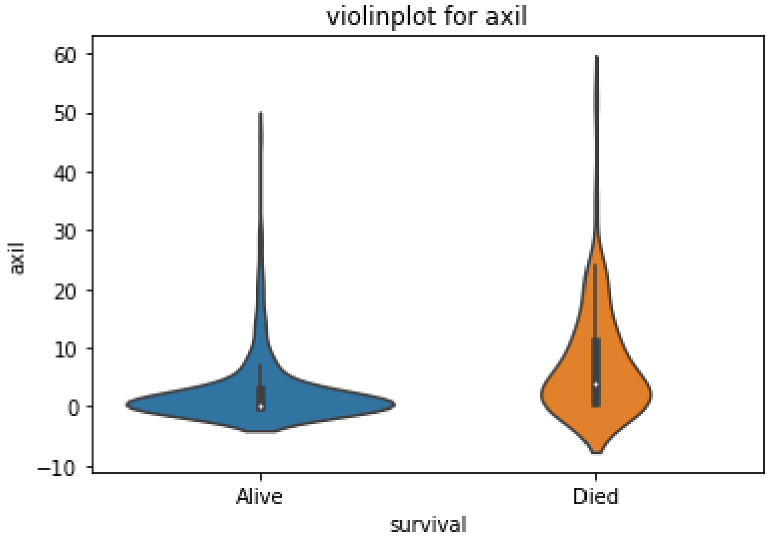

box plot of axil
axil
0
10
20
30
40
50
Alive
Died
survival

In [10]:

```
# voilin Plots
sns.violinplot(x="survival",y="age",data=df)
plt.title('violinplot for age')
plt.show()
sns.violinplot(x="survival",y="op_year",data=df)
plt.title('violinplot for op_year')
plt.show()
sns.violinplot(x="survival",y="axil",data=df)
plt.title('violinplot for axil')
plt.show()
#Violin plots are used here to know the disturbution of data
```

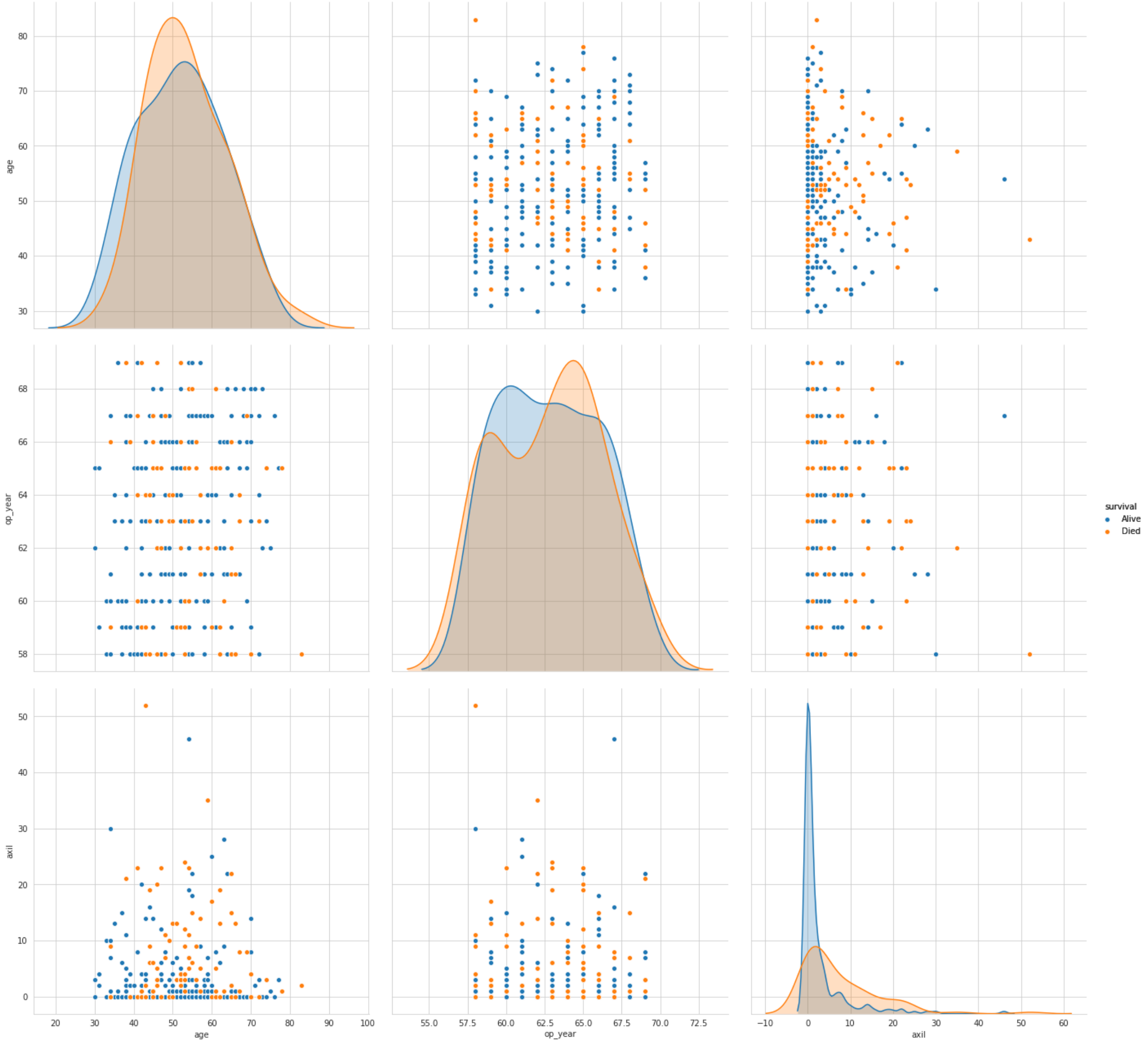






In [11]:

```
#multivariate analysis
sns.set_style('whitegrid')
sns.pairplot(df,hue='survival',height=6).add_legend()
plt.show()
```



OBSERVATION: 1)From all the plots for the given data it is difficult to predict the survival of the patient for the given data 2)We can observe overlap of data using univariate analysis and bi variate analysis.