In [1]:

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

In [2]:

```
df=pd.read_csv("../input/habermans-survival-data-set/haberman.csv",names=["Age","Operation_Year", "Axilar
y_Node","Survial_status"])
```

In [3]:

```
df.columns
```

Out[3]:

```
Index(['Age', 'Operation_Year', 'Axilary_Node', 'Survial_status'], dtype='object')
```

In [4]:

```
df.head(3)
```

Out[4]:

	Age	Operation_Year	Axilary_Node	Survial_status
0	30	64	1	1
1	30	62	3	1
2	30	65	0	1

In [5]:

```
df["Survial_status"].value_counts()
```

Out[5]:

```
1    225
2     81
Name: Survial_status, dtype: int64
```

In [6]:

```
df.shape
```

Out[6]:

```
(306, 4)
```

In [7]:

```
print(df.info())
```

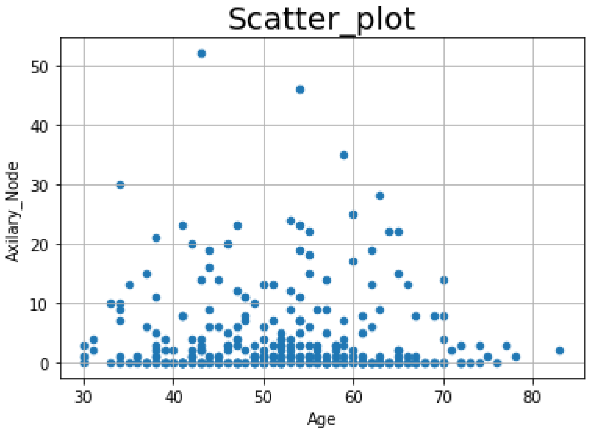
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306 entries, 0 to 305
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Age             306 non-null    int64
 1   Operation_Year  306 non-null    int64
 2   Axilary_Node    306 non-null    int64
 3   Survial_status  306 non-null    int64
dtypes: int64(4)
memory usage: 9.7 KB
None
```

Observations:

- There are no missing values in this data set.
- All the columns are of the integer data type.
- The datatype of the status is an integer, it has to be converted to a categorical datatype
- In the status column, the value 1 can be mapped to ‘yes’ which means the patient has survived 5 years or longer. And the value 2 can be mapped to ‘no’ which means the patient died within 5 years.

2-D Scatter Plot :

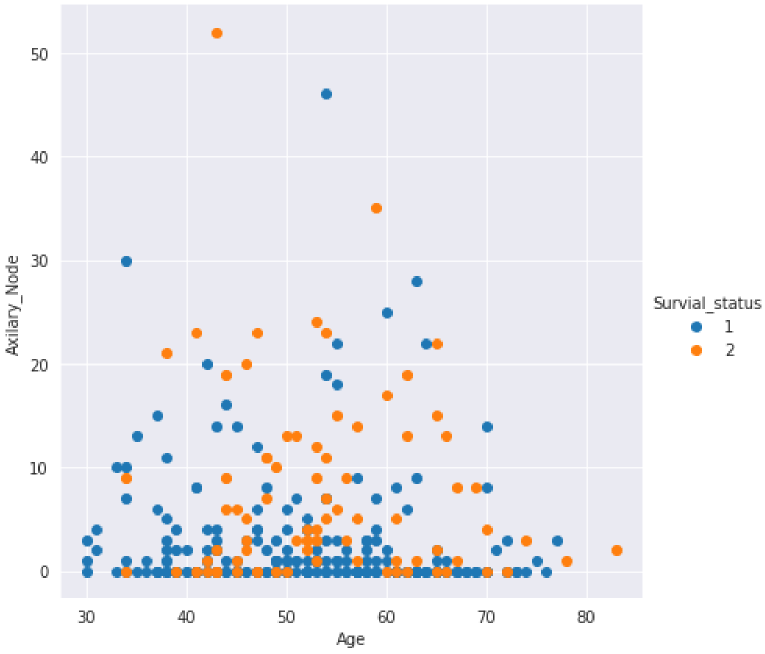
In [8]:
```
df.plot(kind="scatter",x="Age",y="Axilary_Node")
plt.title("Scatter_plot",size=20)
plt.grid()
plt.show()
```



OBSERVATION : By looking above fig we get that it very difficult to find survival and non survival bcoz the point having the same colour .

2-D Scatter plot with color-coding :

In [9]:
```
sns.set_style("darkgrid")
sns.FacetGrid(df,hue="Survial_status",height=6) \
   .map(plt.scatter,"Age","Axilary_Node") \
   .add_legend()
plt.show()
```



OBSERVATION :

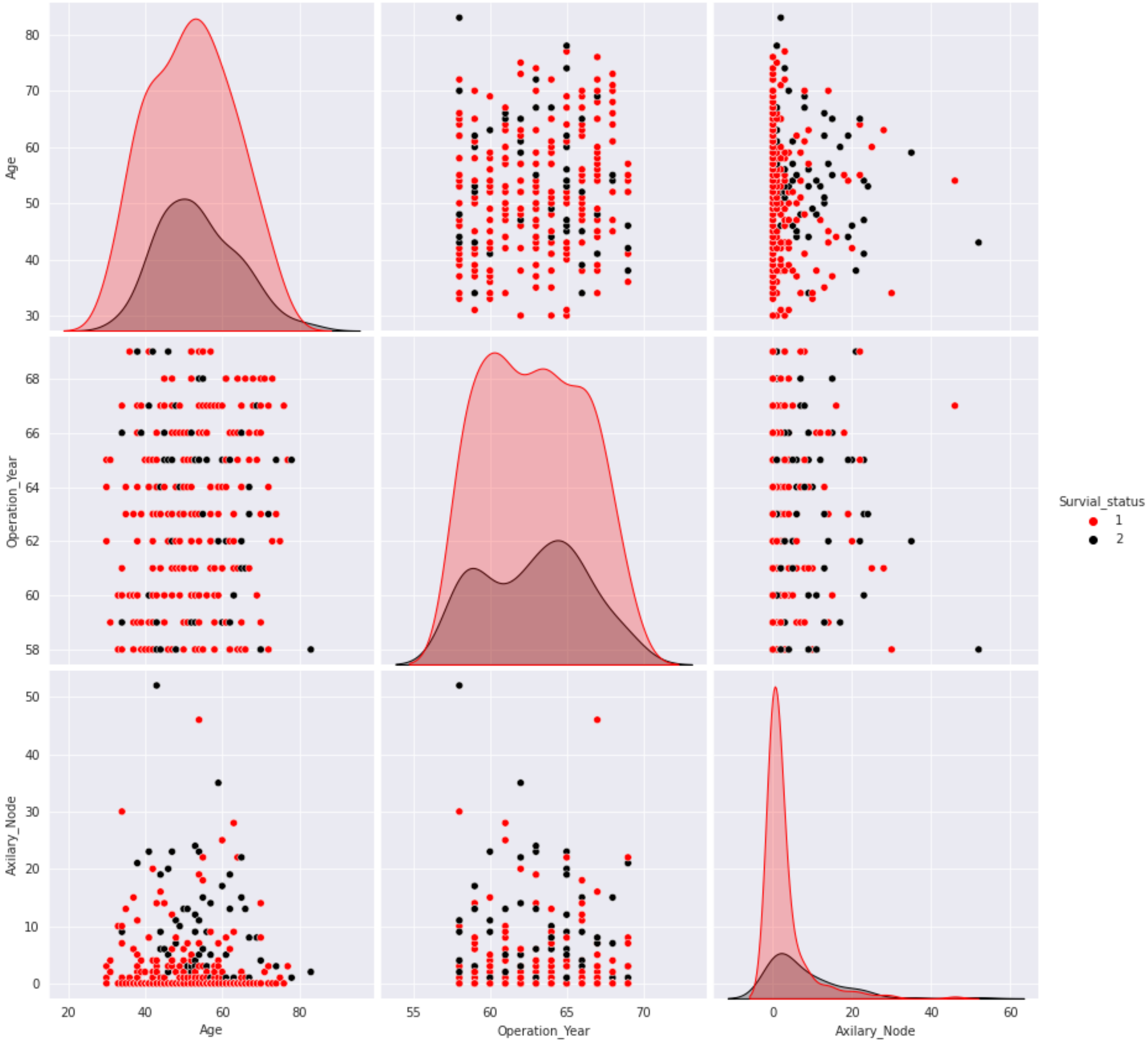
1 - by looking above graph we get that more point are overlapping so we can't get idea by just seeing it

2 - at the age>80 chances of survival is less

3 - in age<35 chance of survival is more

Pair Plot :

```
In [10]: sns.set_style("darkgrid")
         sns.pairplot(df,hue="Survial_status",palette='flag',height=4)#palette is used for colour
         plt.show()
```



Dis-advantages of pair plot:

1 - Can be used when number of features are high.

2 - Cannot visualize higher dimensional patterns in 3-D and 4-D.

3 - Only possible to view 2D patterns.

OBSERVATION :

1 - BY looking above we can't easily find the category of survival and non survival.

2 - IN operation_year and age plot , it very difficult to get than the other plot.

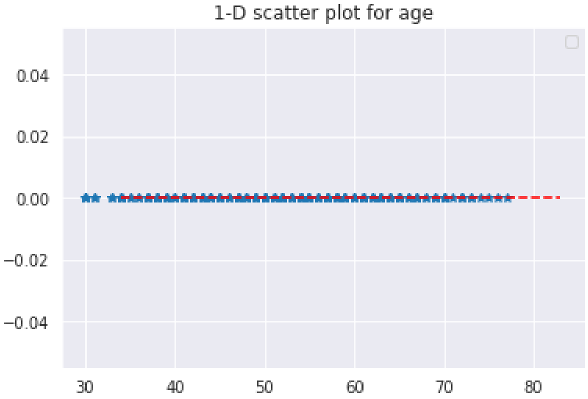
3 - In age vs auxilary_noe plot we can some how get the survival and non survival by their also more point are overlapping, but it is somehow easy than other plots.

In [11]:
```
df1_one = df.loc[df["Survial_status"] == 1];
df1_two = df.loc[df["Survial_status"] == 2];

plt.plot(df1_one["Age"], np.zeros_like(df1_one['Age']), '*')
plt.plot(df1_two["Age"], np.zeros_like(df1_two['Age']), 'r--')

plt.legend('S','N')
plt.title('1-D scatter plot for age')
plt.show()
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:7: UserWarning: Legend does not support 'S'
instances.
A proxy artist may be used instead.
See: https://matplotlib.org/users/legend_guide.html#creating-artists-specifically-for-adding-to-the-lege
nd-aka-proxy-artists
  import sys
```



OBSERVATION :

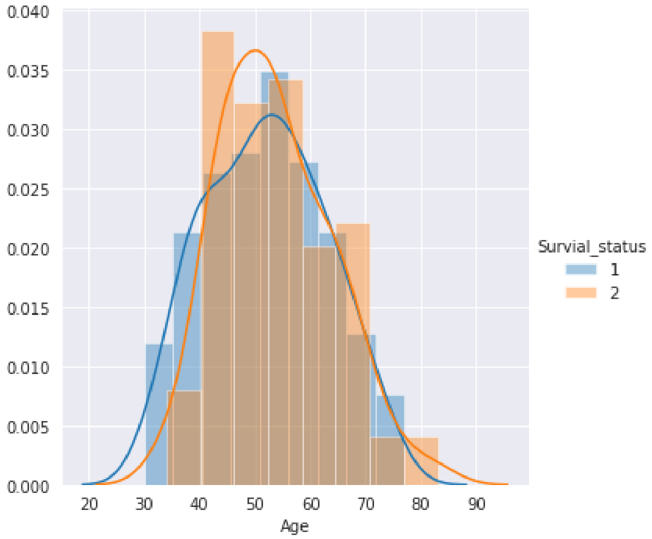
1 . Disadvantages of 1-D scatter plot : Very hard to make sense as points are overlapping a lot.

```
(as above fig show that age of 35-76 are overlapping )
```

Histogram,PDF(Probability Density Functions):

```
In [12]: sns.FacetGrid(df, hue="Survial_status", height=5) \
            .map(sns.distplot, "Age") \
            .add_legend();
         plt.show();
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histogram
s).
  warnings.warn(msg, FutureWarning)
/opt/conda/lib/python3.7/site-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histogram
s).
  warnings.warn(msg, FutureWarning)
```

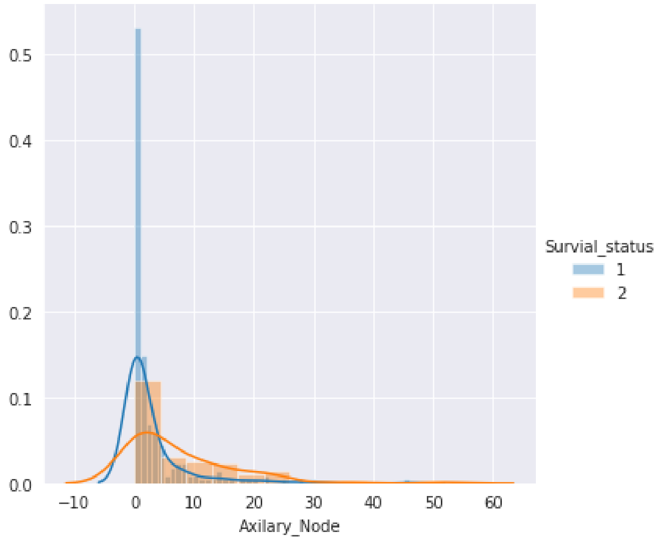


OBSERVATION :

- the data for both the class range between age 30 to 85. both class have mean approximatly equal.

```
In [13]: sns.FacetGrid(df, hue="Survial_status", height=5) \
            .map(sns.distplot, "Axilary_Node") \
            .add_legend();
         plt.show();
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histogram
s).
  warnings.warn(msg, FutureWarning)
/opt/conda/lib/python3.7/site-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histogram
s).
  warnings.warn(msg, FutureWarning)
```



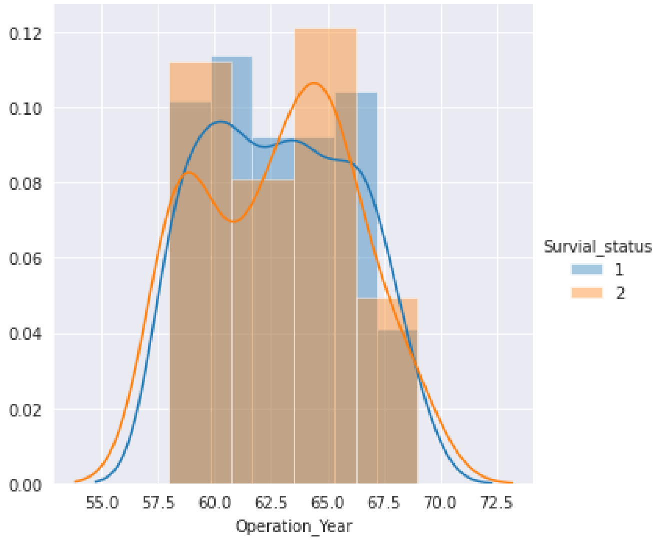
OBSERVATION :

When the number of axillary nodes is roughly between 0-1, chances of survival is maximum. Then the survival rate is gradually declining. But when axillary nodes is more than 20, chances of death are more.

In [14]:

```
sns.FacetGrid(df, hue="Survial_status", height=5) \
   .map(sns.distplot, "Operation_Year") \
   .add_legend();
plt.show();
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histogram
s).
  warnings.warn(msg, FutureWarning)
/opt/conda/lib/python3.7/site-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a dep
recated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histogram
s).
  warnings.warn(msg, FutureWarning)
```



OBSERVATION :

. survival and non survival are overlapping in the Age range (58.0-69.0)

. at Age of 68 survival chance is less

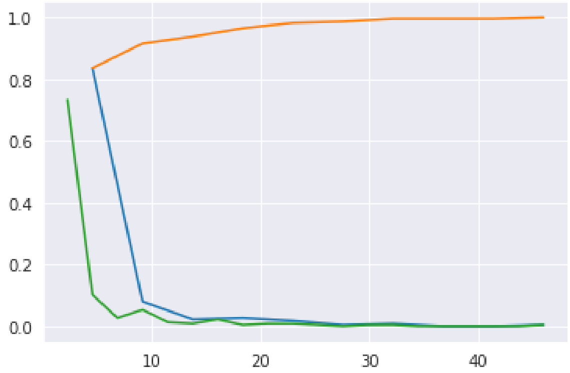
CDF(Cumulative Distribution Function):

In [15]:

```
counts, bin_edges = np.histogram(df1_one['Axilary_Node'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges);
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf);
plt.plot(bin_edges[1:], cdf)

counts, bin_edges = np.histogram(df1_one['Axilary_Node'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
plt.plot(bin_edges[1:],pdf);
plt.show()
```

```
[0.83555556 0.08       0.02222222 0.02666667 0.01777778 0.00444444
 0.00888889 0.         0.         0.00444444]
[ 0.   4.6  9.2 13.8 18.4 23.  27.6 32.2 36.8 41.4 46. ]
```

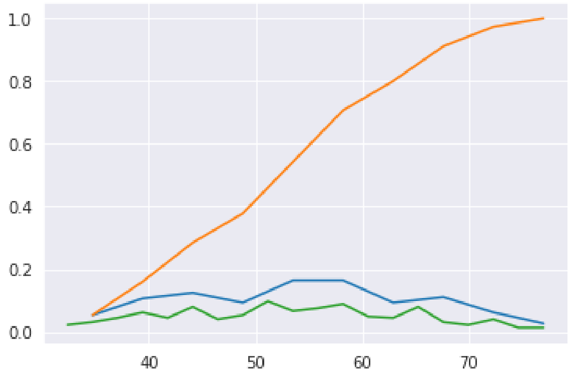


In [16]:

```
counts, bin_edges = np.histogram(df1_one['Age'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges);
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf);
plt.plot(bin_edges[1:], cdf)

counts, bin_edges = np.histogram(df1_one['Age'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
plt.plot(bin_edges[1:],pdf);
plt.show()
```

```
[0.05333333 0.10666667 0.12444444 0.09333333 0.16444444 0.16444444
 0.09333333 0.11111111 0.06222222 0.02666667]
[30.  34.7 39.4 44.1 48.8 53.5 58.2 62.9 67.6 72.3 77. ]
```

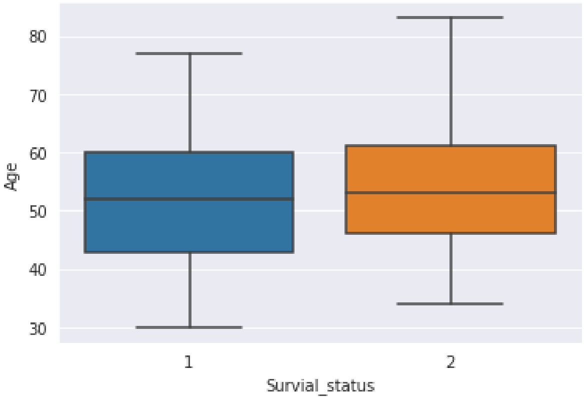


Need for Cumulative Distribution Function (CDF) :

We can visually see what percentage of various category by use of cdf, which can't be get by the pdf

Box Plot:

In [17]:
```
sns.boxplot(x="Survial_status",y="Age",data=df)
plt.show()
```



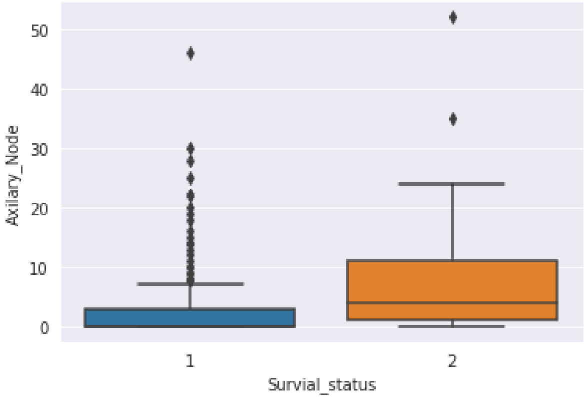
OBSERVATION:

1 . There are no outliers and much can be derived from this plot.

2 . Age of survival lies between 42-60.

3 . Age of non-survival lies between 45-61.

In [18]:
```
sns.boxplot(x="Survial_status",y="Axilary_Node",data=df)
plt.show()
```



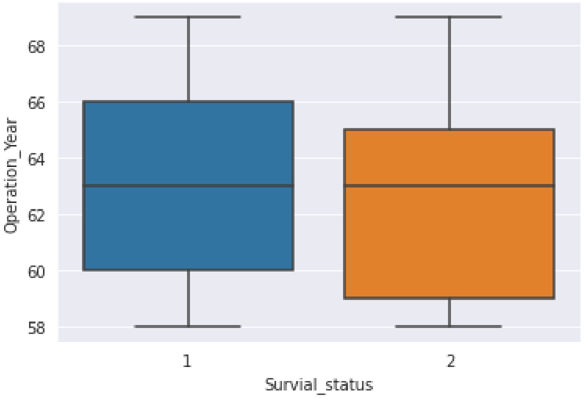
OBSERVATION:

1 . There are a lot of outliers so median is preferred over mean.

2 . Axillary nodes for survival lie between 0-4.

3 . Axillary nodes for non-survival lie between 2-11.

```
In [19]: sns.boxplot(x="Survial_status",y="Operation_Year",data=df)
         plt.show()
```



OBSERVATION :

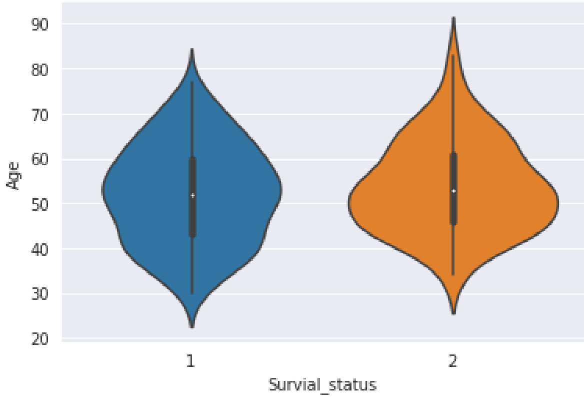
1 . There are no outliers and much can be derived from this plot.

2 . Axillary nodes for survival lie between 60-66

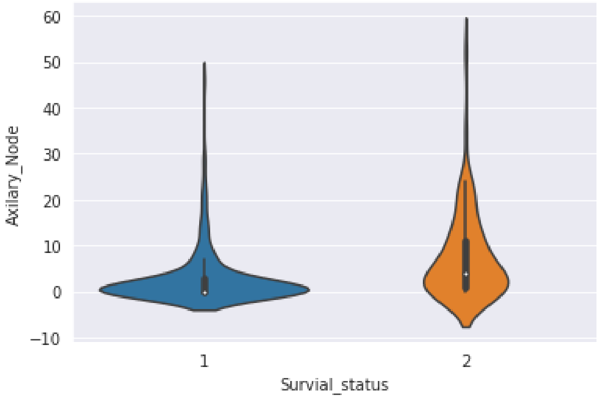
3 . Axillary nodes for non-survival lie between 59-65

Violin Plot :

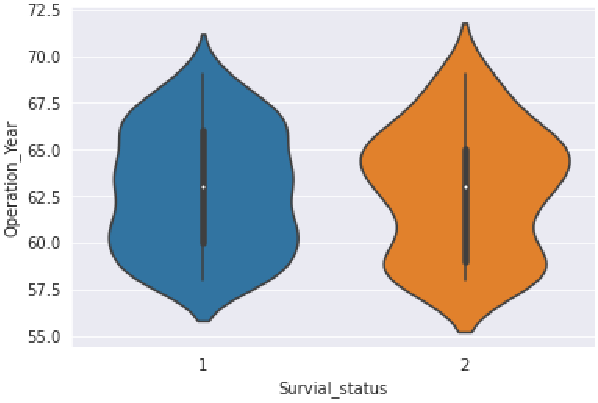
```
In [20]: sns.violinplot(x="Survial_status",y="Age",data=df,height=20)
         plt.show()
```



```
In [21]: sns.violinplot(x="Survial_status",y="Axilary_Node",data=df,height=20)
         plt.show()
```

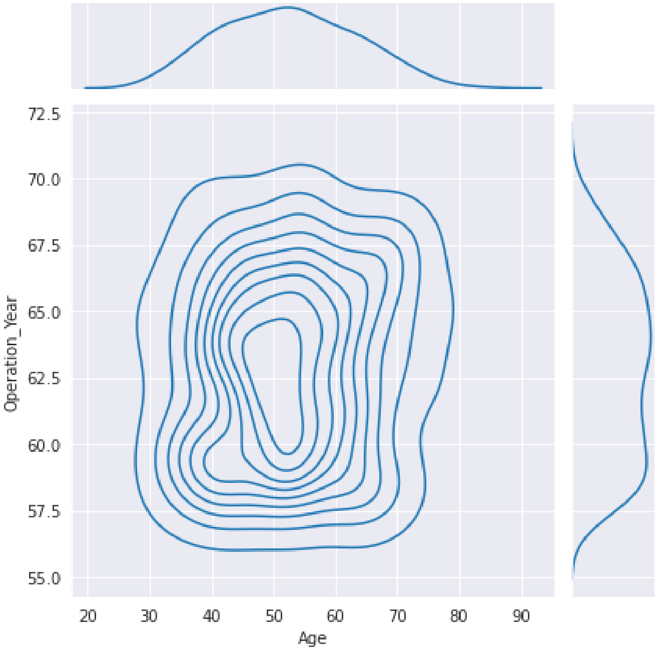


```
In [22]: sns.violinplot(x="Survial_status",y="Operation_Year",data=df,height=20)
         plt.show()
```



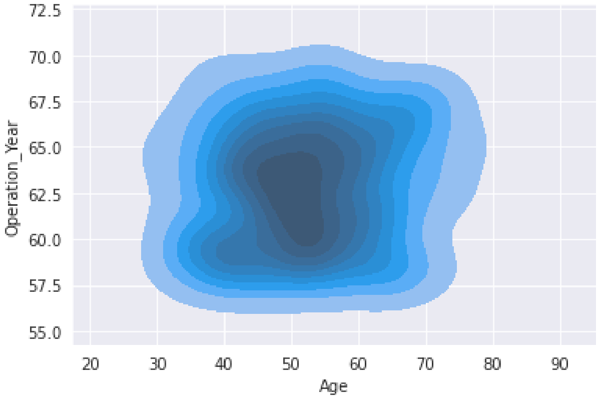
Multivariate probability density, contour plot :

```
In [23]: sns.jointplot(x="Age", y="Operation_Year", data=df, kind='kde')
         plt.show()
```



```
In [24]: sns.kdeplot(data=df, x="Age", y="Operation_Year", fill=True)
```

```
Out[24]: <AxesSubplot:xlabel='Age', ylabel='Operation_Year'>
```



OBSERVATION :

As the the we go deeper in the dark side the chances of the non_survival is increase .