

In [39]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
pd.set_option('display.notebook_repr_html', True)
import warnings;
warnings.simplefilter('ignore')
```

Information From Kaggle Data

Relevant Information: The dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer.

Number of Instances: 306

Number of Attributes: 4 (including the class attribute)

Attribute Information:

- Age of patient at time of operation (numerical)
- Patient's year of operation (year - 1900, numerical)
- Number of positive axillary nodes detected (numerical)
- Survival status (class attribute) 1 = the patient survived 5 years or longer 2 = the patient died within 5 year

In [40]:

```
haberman_data_raw = pd.read_csv("haberman.csv", header = None)  
haberman_data_raw # Contains no Columns Labels
```

Out[40]:

	0	1	2	3
0	30	64	1	1
1	30	62	3	1
2	30	65	0	1
3	31	59	2	1
4	31	65	4	1
5	33	58	10	1
6	33	60	0	1
7	34	59	0	2
8	34	66	9	2
9	34	58	30	1
10	34	60	1	1
11	34	61	10	1
12	34	67	7	1
13	34	60	0	1
14	35	64	13	1
15	35	63	0	1
16	36	60	1	1
17	36	69	0	1
18	37	60	0	1
19	37	63	0	1
20	37	58	0	1
21	37	59	6	1
22	37	60	15	1
23	37	63	0	1
24	38	69	21	2
25	38	59	2	1
26	38	60	0	1
27	38	60	0	1
28	38	62	3	1
29	38	64	1	1
...
276	67	66	0	1
277	67	61	0	1

	0	1	2	3
278	67	65	0	1
279	68	67	0	1
280	68	68	0	1
281	69	67	8	2
282	69	60	0	1
283	69	65	0	1
284	69	66	0	1
285	70	58	0	2
286	70	58	4	2
287	70	66	14	1
288	70	67	0	1
289	70	68	0	1
290	70	59	8	1
291	70	63	0	1
292	71	68	2	1
293	72	63	0	2
294	72	58	0	1
295	72	64	0	1
296	72	67	3	1
297	73	62	0	1
298	73	68	0	1
299	74	65	3	2
300	74	63	0	1
301	75	62	1	1
302	76	67	0	1
303	77	65	3	1
304	78	65	1	2
305	83	58	2	2

306 rows × 4 columns

In [41]:

```
haberman_data = pd.read_csv('haberman.csv', header=None, names=['Age', 'Operation_Y  
ear', 'axil_nodes', 'Surv_status'])  
haberman_data
```

Out[41]:

	Age	Operation_Year	axil_nodes	Surv_status
0	30	64	1	1
1	30	62	3	1
2	30	65	0	1
3	31	59	2	1
4	31	65	4	1
5	33	58	10	1
6	33	60	0	1
7	34	59	0	2
8	34	66	9	2
9	34	58	30	1
10	34	60	1	1
11	34	61	10	1
12	34	67	7	1
13	34	60	0	1
14	35	64	13	1
15	35	63	0	1
16	36	60	1	1
17	36	69	0	1
18	37	60	0	1
19	37	63	0	1
20	37	58	0	1
21	37	59	6	1
22	37	60	15	1
23	37	63	0	1
24	38	69	21	2
25	38	59	2	1
26	38	60	0	1
27	38	60	0	1
28	38	62	3	1
29	38	64	1	1
...
276	67	66	0	1
277	67	61	0	1

	Age	Operation_Year	axil_nodes	Surv_status
278	67	65	0	1
279	68	67	0	1
280	68	68	0	1
281	69	67	8	2
282	69	60	0	1
283	69	65	0	1
284	69	66	0	1
285	70	58	0	2
286	70	58	4	2
287	70	66	14	1
288	70	67	0	1
289	70	68	0	1
290	70	59	8	1
291	70	63	0	1
292	71	68	2	1
293	72	63	0	2
294	72	58	0	1
295	72	64	0	1
296	72	67	3	1
297	73	62	0	1
298	73	68	0	1
299	74	65	3	2
300	74	63	0	1
301	75	62	1	1
302	76	67	0	1
303	77	65	3	1
304	78	65	1	2
305	83	58	2	2

306 rows × 4 columns

In [42]:

```
# Classes attributes are 1 and 2 show the year of survival of the patient after  
the operations  
# No of data points per class  
print(haberman_data["Surv_status"].value_counts())
```

```
1    225
```

```
2     81
```

```
Name: Surv_status, dtype: int64
```

Observations From the Data:

1. Shows class 1 : 225 patients survived 5 years and more after the operation
2. and class 2 : 81 data points shows patients survived only 5 years

In [43]:

```
print(haberman_data.columns)
```

```
Index(['Age', 'Operation_Year', 'axil_nodes', 'Surv_status'], dtype  
='object')
```

Observations

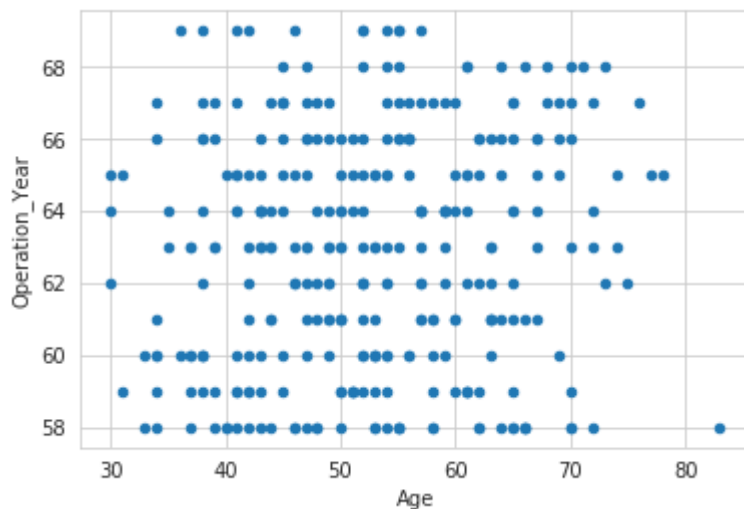
1: Columns : Age , Operation_Year and axil_nodes can be used to classify whether Surv_status is 1 or 2

Bi-variate analysis

2D PLOTS

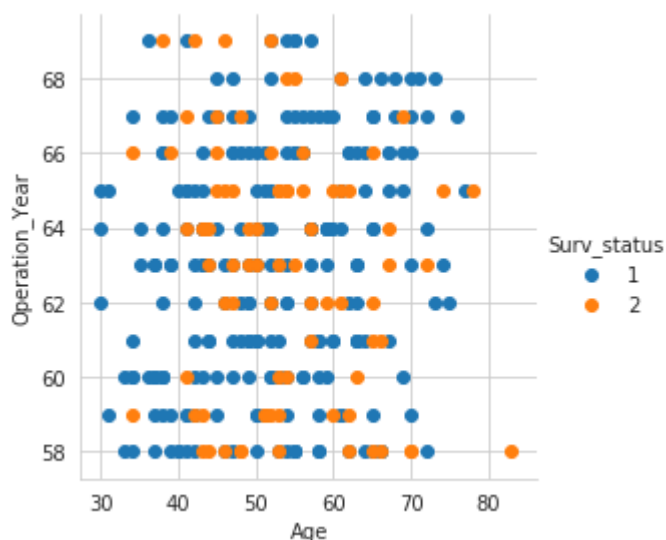
In [44]:

```
# FIRST WE WILL PLOT THE SCATTER PLOT TO SEE
haberman_data.plot(kind="scatter",x="Age",y="Operation_Year")
plt.show() # cannot make much sense out of this , can't tell which point belong
to which survival status
```



In [45]:

```
# using color coding to see if anything comes out
import seaborn as sns
sns.set_style("whitegrid")
sns.FacetGrid(haberman_data,hue="Surv_status",size=4).map(plt.scatter,"Age","Ope
ration_Year").add_legend();
plt.show();
```



Observations:

1. Age and Operation Year can't separate the survival status of the patients
2. Separation of Survival Status is difficult as there is a lot of overlap

Try With Pair Plot

In [46]:

```
#close the already shown plot
plt.close()
sns.set_style("whitegrid")
sns.pairplot(haberman_data,hue="Surv_status",size=4,vars=['Age', 'Operation_Yea
r', 'axil_nodes'],
plt.show())
```



Observations:

1. Pairplot is not able to distinguish the data
2. We are not able to distinguish the class using the two feature at a time and hence bivariate analysis is not helpful

In [47]:

```
# We will now seprate the data on the bases on the class and will apply some var  
iable based visualisation  
# class 1 and class data seperation  
class1_data = haberman_data.loc[haberman_data["Surv_status"]==1]  
class2_data = haberman_data.loc[haberman_data["Surv_status"]==2]  
  
print("People Who Survived 5 Years and More")  
class1_data.describe()
```

People Who Survived 5 Years and More

Out[47]:

	Age	Operation_Year	axil_nodes	Surv_status
count	225.000000	225.000000	225.000000	225.0
mean	52.017778	62.862222	2.791111	1.0
std	11.012154	3.222915	5.870318	0.0
min	30.000000	58.000000	0.000000	1.0
25%	43.000000	60.000000	0.000000	1.0
50%	52.000000	63.000000	0.000000	1.0
75%	60.000000	66.000000	3.000000	1.0
max	77.000000	69.000000	46.000000	1.0

In [48]:

```
print("People Who Survived 5 Years and less")  
class2_data.describe()
```

People Who Survived 5 Years and less

Out[48]:

	Age	Operation_Year	axil_nodes	Surv_status
count	81.000000	81.000000	81.000000	81.0
mean	53.679012	62.827160	7.456790	2.0
std	10.167137	3.342118	9.185654	0.0
min	34.000000	58.000000	0.000000	2.0
25%	46.000000	59.000000	1.000000	2.0
50%	53.000000	63.000000	4.000000	2.0
75%	61.000000	65.000000	11.000000	2.0
max	83.000000	69.000000	52.000000	2.0

#Observations

1. Age and Operation Year are not changing that much in both the classes statistics and have almost same values in both the sets , the variable that is changing is axillary nodes in both the classes
2. If we see the survived patients then 75 % of the patients have axillary nodes values ≤ 3 In case of the patients of class 2 who died 75% of the patients have axillary nodes values ≤ 11
3. If we see the axil_nodes max values than in case of survived patients it is 46 and in case of dead patients it is 52 means . Alone axil_nodes values can't decide the survival as there are cases of axil_nodes values = 1 but still they didn't survive

Univariate Analysis

In [49]:

```
# We will analyse the data based on one feature
```

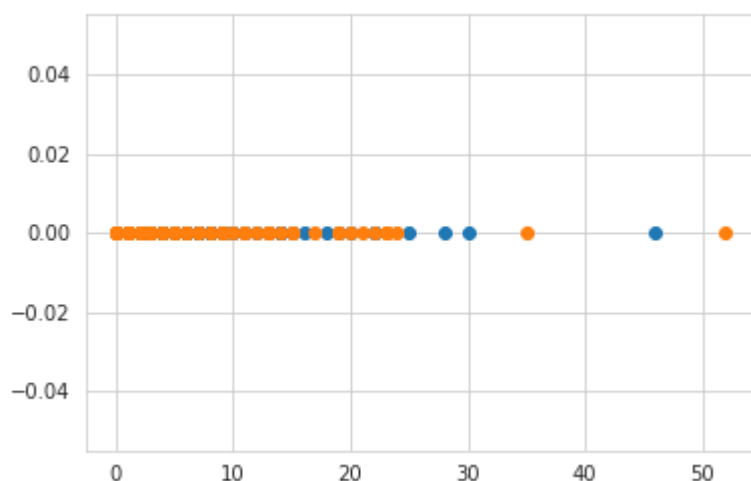
1-D Scatter plots

In [50]:

```
# we will draw 1-D Scatter plot on the Auxillary Nodes data points
```

In [51]:

```
# Using axil_nodes for 1-D scatter plot as we know it changes marginally  
plt.plot(class1_data["axil_nodes"], np.zeros_like(class1_data["axil_nodes"]), 'o')  
plt.plot(class2_data["axil_nodes"], np.zeros_like(class2_data["axil_nodes"]), 'o')  
plt.show()
```



#Observations

1. Hard to make sense from the plot because points are overlapping each other

Using Only One Feature (Univariate Based Analysis)

In [52]:

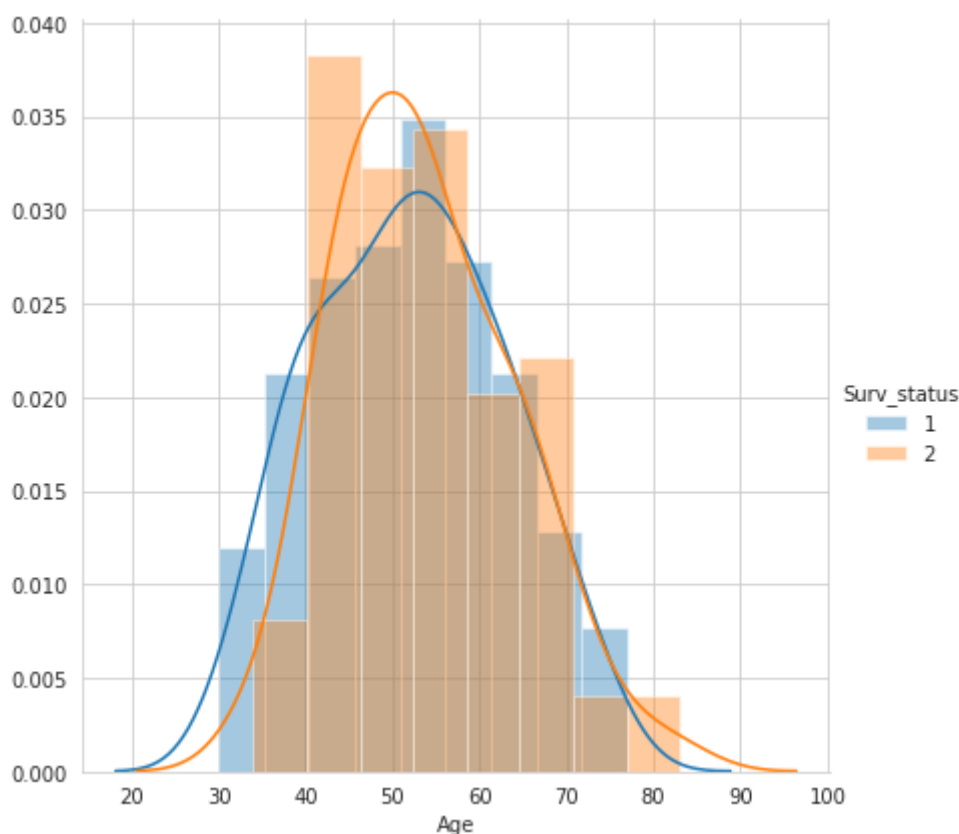
```
# we will use Histogram , PDF , CDF to Analyse the data of the patients
```

1. Using Age Feature

Histogram

In [53]:

```
sns.set_style("whitegrid")
sns.FacetGrid(haberman_data, hue="Surv_status", size=6).map(sns.distplot, "Age").add_legend();
plt.show();
```



PDF AND CDF Using Age

In [54]:

```
counts, bin_edges = np.histogram(class1_data["Age"], bins=10, density=True)
```

In [55]:

```
counts # shows the no of pdf values for each bin
```

Out[55]:

```
array([0.01134752, 0.02269504, 0.02647754, 0.01985816, 0.03498818,  
       0.03498818, 0.01985816, 0.02364066, 0.01323877, 0.00567376])
```

In [56]:

```
bin_edges
```

Out[56]:

```
array([30. , 34.7, 39.4, 44.1, 48.8, 53.5, 58.2, 62.9, 67.6, 72.3, 7  
       7. ])
```

In [57]:

```
print(len(counts))  
print(len(bin_edges)) # need to start from the second
```

10

11

In [58]:

```
pdf = counts/sum(counts)
print("\nFor Class : 1")
print(pdf)
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)

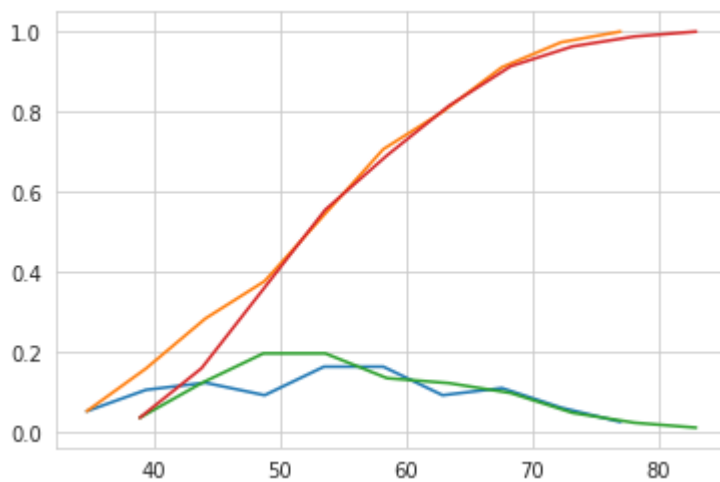
# plot for class 2
counts,bin_edges = np.histogram(class2_data["Age"],bins=10,density=True)
pdf = counts/sum(counts)
print("\n For Class 2 : ")
print(pdf)
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)
plt.show()
```

For Class : 1

```
[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. ]
```

For Class 2 :

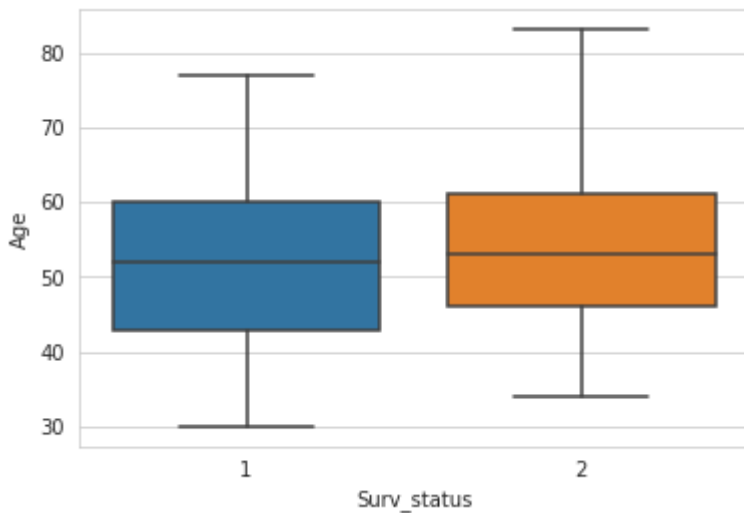
```
[0.03703704 0.12345679 0.19753086 0.19753086 0.13580247 0.12345679
 0.09876543 0.04938272 0.02469136 0.01234568]
[34.  38.9 43.8 48.7 53.6 58.5 63.4 68.3 73.2 78.1 83. ]
```



Using Box Plot

In [59]:

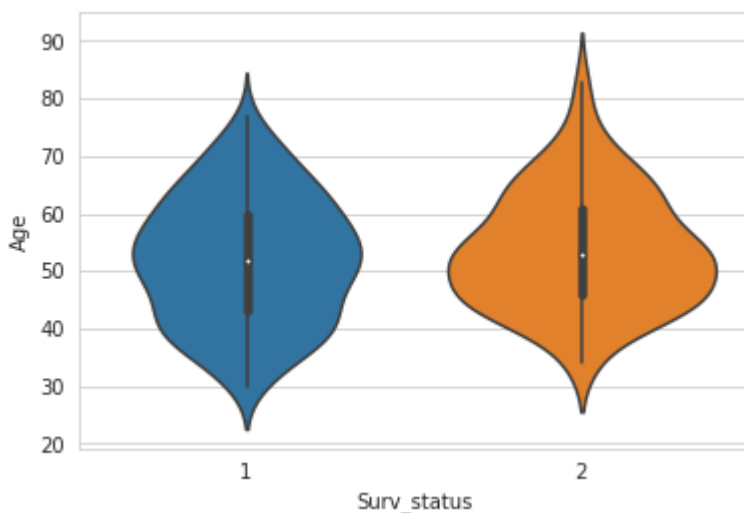
```
# Box plot on the Age Feature
sns.boxplot(x="Surv_status",y="Age",data=haberman_data)
plt.show()
# 25th , 50th , 75th percentile are overlapping each other not able to distinguish
```



Violin Plot

In [60]:

```
sns.violinplot(x="Surv_status",y="Age",data=haberman_data)
plt.show()
# same problem of overlapping in the percentiles
```



#Final Ovservations:

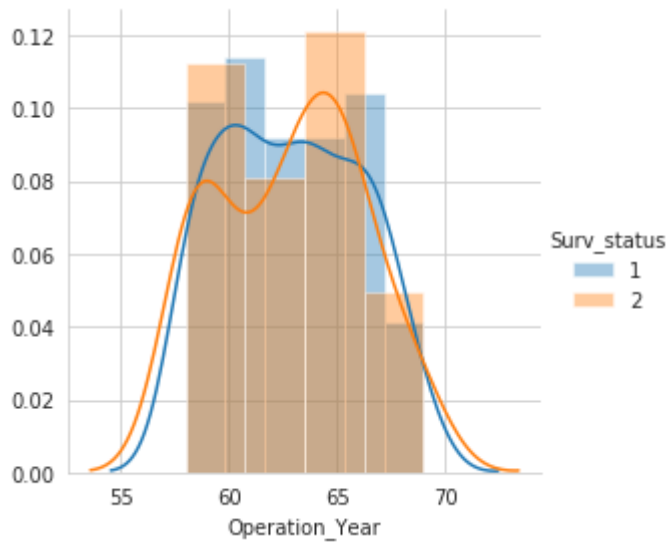
1. Age plots are almost same for both the classes 1 and 2.
2. There is lot of overlapping in maximum plots when we draw it using the Patient Age
3. This show that Patient Age can't be used for classification into the two ages
4. Patient Age doesn't affect the survival status of the patient to that extent.

2. Using Operation Year Feature

Histogram

In [61]:

```
sns.set_style("whitegrid")
sns.FacetGrid(haberman_data, hue="Surv_status", size=4).map(sns.distplot, "Operation_Year").add_legend();
plt.show();
```



CDF and PDF

In [62]:

```
counts,bin_edges = np.histogram(class1_data["Operation_Year"],bins=10,density=True)
pdf = counts/sum(counts)
print("\nFor Class : 1")
print(pdf)
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)

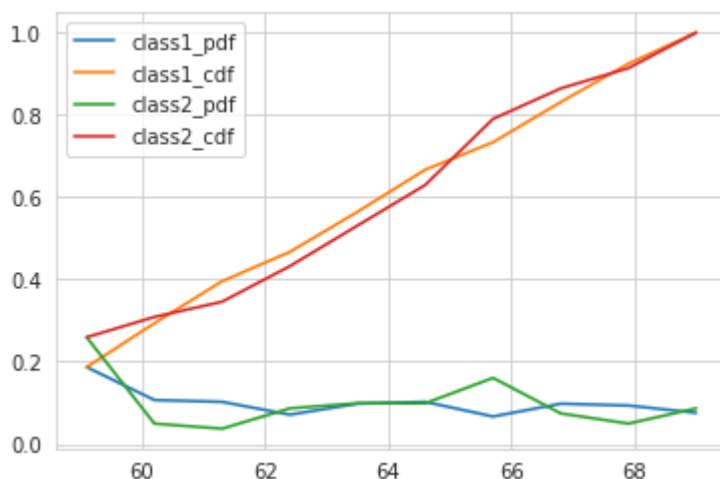
# plot for class 2
counts,bin_edges = np.histogram(class2_data["Operation_Year"],bins=10,density=True)
pdf = counts/sum(counts)
print("\n For Class 2 : ")
print(pdf)
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)
plt.legend(["class1_pdf","class1_cdf","class2_pdf","class2_cdf"])
plt.show()
```

For Class : 1

```
[0.18666667 0.10666667 0.10222222 0.07111111 0.09777778 0.10222222
 0.06666667 0.09777778 0.09333333 0.07555556]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
```

For Class 2 :

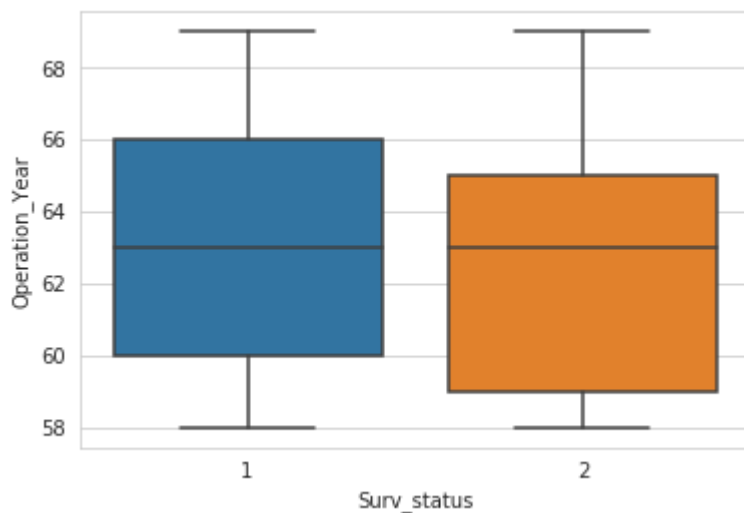
```
[0.25925926 0.04938272 0.03703704 0.08641975 0.09876543 0.09876543
 0.16049383 0.07407407 0.04938272 0.08641975]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
```



Bos Plot Using the Operation Year

In [63]:

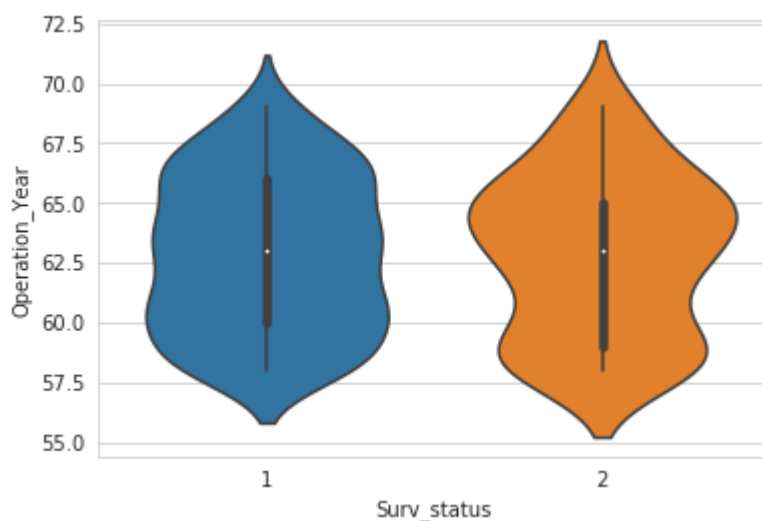
```
sns.set_style("whitegrid")
sns.boxplot(x="Surv_status",y="Operation_Year",data=haberman_data);
```



Violin Plot

In [64]:

```
sns.violinplot(x="Surv_status",y="Operation_Year",data=haberman_data);
```



#Final Observations:

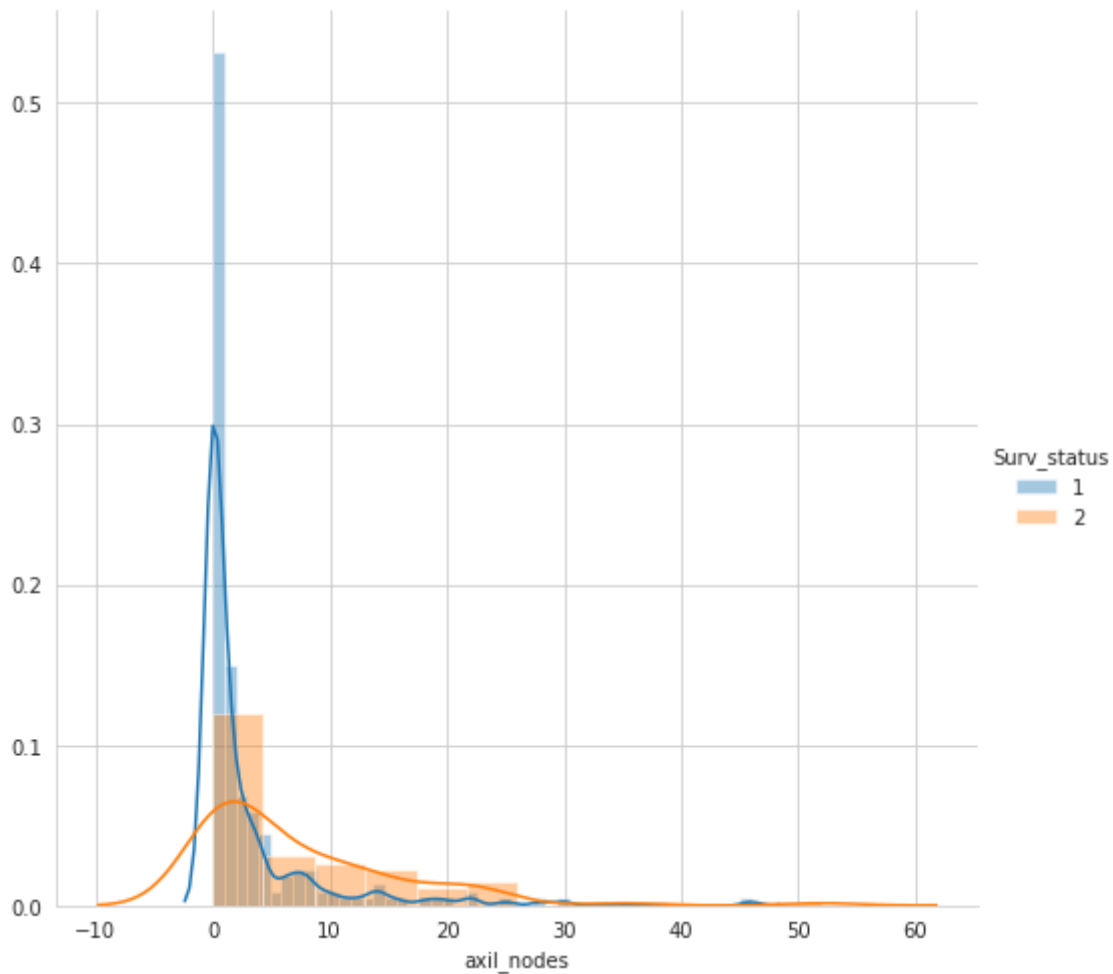
1. Operation Year doesn't affect the survival status of the patient upto that extent
2. Histogram , CDF , PDF, Box , Violin Plots are not able to classify and seperate the data becuase of overlapping
3. Operation year cannot be considered as the important feature in the haberman data set.

3 . Using the Axillary Nodes Values

Histogram

In [65]:

```
# using the axil_nodes column values now
import warnings;
warnings.simplefilter('ignore')
sns.set_style("whitegrid")
sns.FacetGrid(haberman_data, hue="Surv_status", height=7).map(sns.distplot, "axil_n
odes").add_legend();
plt.show();
```



PDF , CDF Using Axillary Nodes

In [66]:

```
counts,bin_edges = np.histogram(class1_data["axil_nodes"],bins=10,density=True)
pdf = counts/sum(counts)
print("\nFor Class : 1")
print(pdf)
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)

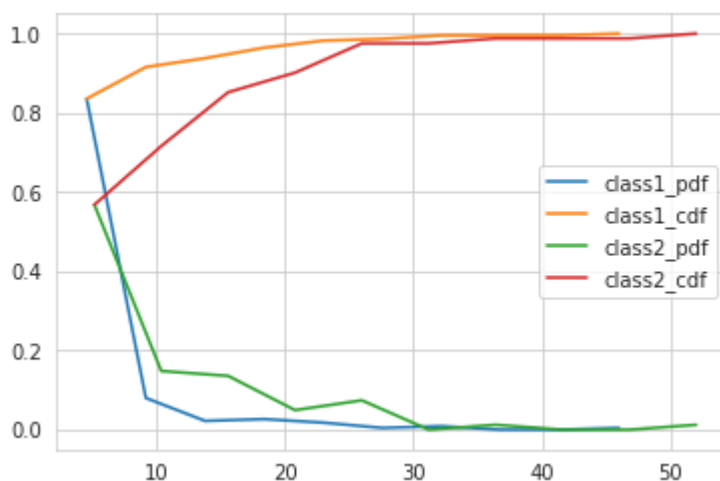
# plot for class 2
counts,bin_edges = np.histogram(class2_data["axil_nodes"],bins=10,density=True)
pdf = counts/sum(counts)
print("\n For Class 2 : ")
print(pdf)
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)
plt.legend(["class1_pdf","class1_cdf","class2_pdf","class2_cdf"])
plt.show()
```

For Class : 1

```
[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. ]
```

For Class 2 :

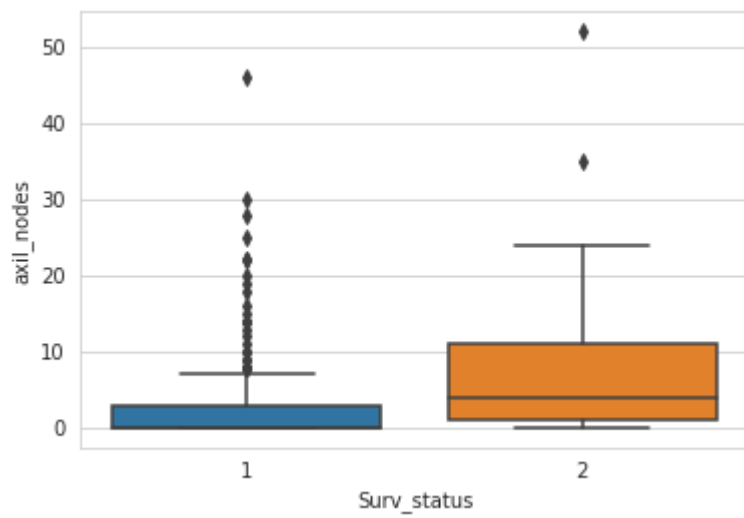
```
[0.56790123 0.14814815 0.13580247 0.04938272 0.07407407 0.
 0.01234568 0.        0.        0.01234568]
[ 0.   5.2 10.4 15.6 20.8 26.  31.2 36.4 41.6 46.8 52. ]
```



Box Plots

In [67]:

```
sns.boxplot(x="Surv_status",y="axil_nodes",data=haberman_data)  
plt.show()
```



In [68]:

```
# some percentage calculation on the dat sets

print("\nPercentage of the patients having 0 axillary nodes counts : {} %"
      .format((len(haberman_data[haberman_data["axil_nodes"]==0])/len(haberman_data)) *100))

print("\nPercentage of the patients having axillary nodes counts < 3 : {} %"
      .format((len(haberman_data[haberman_data["axil_nodes"]<=3])/len(haberman_data)) *100))

print("\nPercentage of the patients having axillary nodes counts > 3 and less 10 : {} %"
      .format((len(haberman_data[(haberman_data["axil_nodes"]>3) & (haberman_data["axil_nodes"]<10)])/len(haberman_data)) *100))

print("\nPercentage of the patients having axillary nodes counts less 10 : {} %"
      .format((len(haberman_data[haberman_data["axil_nodes"]<10])/len(haberman_data)) *100))

print("\nPercentage of the patients having axillary nodes counts > 10 : {} %"
      .format((len(haberman_data[haberman_data["axil_nodes"]>10])/len(haberman_data)) *100))
```

Percentage of the patients having 0 axillary nodes counts : 44.44444444444444 %

Percentage of the patients having axillary nodes counts < 3 : 70.91503267973856 %

Percentage of the patients having axillary nodes counts > 3 and less 10 : 15.032679738562091 %

Percentage of the patients having axillary nodes counts less 10 : 85.94771241830065 %

Percentage of the patients having axillary nodes counts > 10 : 13.071895424836603 %

In [69]:

```
# some percentage calculation on the data sets

print("\nPercentage of the patients having 0 auxillary nodes counts and survival
status 1 : {} %"
      .format((len(haberman_data[(haberman_data["axil_nodes"]==0) & (haberman_data
["Surv_status"]==1)])/len(haberman_data)) *100))

print("\nPercentage of the patients having auxillary nodes counts < 3 and survi
val status 1 : {} %"
      .format((len(haberman_data[(haberman_data["axil_nodes"]<=3 & (haberman_data[
"Surv_status"]==1)])/len(haberman_data)) *100))

print("\nPercentage of the patients having auxillary nodes counts > 3 and less
10 and class 1 : {} %"
      .format((len(haberman_data[(haberman_data["axil_nodes"]>3) & (haberman_data
["axil_nodes"]<10) & (haberman_data["Surv_status"]==1)])/len(class1_data)) *100
))
```

Percentage of the patients having 0 auxillary nodes counts and survi
val status 1 : 38.23529411764706 %

Percentage of the patients having auxillary nodes counts < 3 and su
rvival status 1 : 55.22875816993464 %

Percentage of the patients having auxillary nodes counts > 3 and le
ss 10 and class 1 : 12.444444444444445 %

In [70]:

```
# some percentage calculation on the data sets

print("\nPercentage of the patients having 0 axillary nodes counts and survival
      status 2 : {} %"
      .format((len(haberman_data[(haberman_data["axil_nodes"]==0) & (haberman_data[
"Surv_status"]==2)]/len(haberman_data)) *100))

print("\nPercentage of the patients having axillary nodes counts < 3 and survival
      status 2 : {} %"
      .format((len(haberman_data[(haberman_data["axil_nodes"]<=3 & (haberman_data[
"Surv_status"]==2)]/len(haberman_data)) *100))

print("\nPercentage of the patients having axillary nodes counts > 3 and less 10
      and class 2 : {} %"
      .format((len(haberman_data[(haberman_data["axil_nodes"]>3) & (haberman_data[
"axil_nodes"]<10) & (haberman_data["Surv_status"]==2)]/len(class1_data)) *100
      ))

print("\nPercentage of the patients having axillary nodes counts >10 and survival
      status 2 : {} %"
      .format((len(haberman_data[(haberman_data["axil_nodes"]>10 & (haberman_data[
"Surv_status"]==2)]/len(haberman_data)) *100))
```

Percentage of the patients having 0 axillary nodes counts and survival status 2 : 6.209150326797386 %

Percentage of the patients having axillary nodes counts < 3 and survival status 2 : 47.05882352941176 %

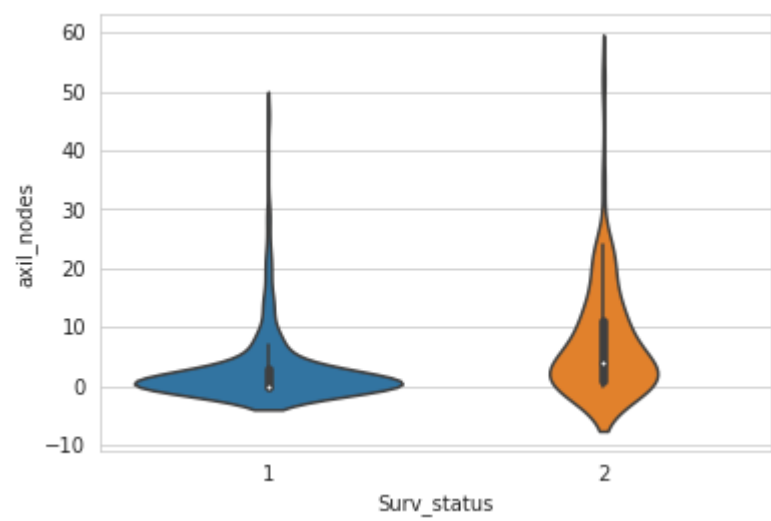
Percentage of the patients having axillary nodes counts > 3 and less 10 and class 2 : 8.0 %

Percentage of the patients having axillary nodes counts >10 and survival status 2 : 55.55555555555556 %

Violin Plot

In [71]:

```
sns.violinplot(x="Surv_status",y="axil_nodes",data=haberman_data)
plt.show()
```



In [72]:

```
class1_data.describe()
```

Out[72]:

	Age	Operation_Year	axil_nodes	Surv_status
count	225.000000	225.000000	225.000000	225.0
mean	52.017778	62.862222	2.791111	1.0
std	11.012154	3.222915	5.870318	0.0
min	30.000000	58.000000	0.000000	1.0
25%	43.000000	60.000000	0.000000	1.0
50%	52.000000	63.000000	0.000000	1.0
75%	60.000000	66.000000	3.000000	1.0
max	77.000000	69.000000	46.000000	1.0

In [73]:

```
class2_data.describe()
```

Out[73]:

	Age	Operation_Year	axil_nodes	Surv_status
count	81.000000	81.000000	81.000000	81.0
mean	53.679012	62.827160	7.456790	2.0
std	10.167137	3.342118	9.185654	0.0
min	34.000000	58.000000	0.000000	2.0
25%	46.000000	59.000000	1.000000	2.0
50%	53.000000	63.000000	4.000000	2.0
75%	61.000000	65.000000	11.000000	2.0
max	83.000000	69.000000	52.000000	2.0

#Observations

1. Not able to clearly separate the data points but upto some extent but overlapping is very less compared to Age and Operation Year
2. As we can see that 50% of the patients having 0 axillary nodes have survived.
3. Inference can be drawn that as the no of the axillary nodes increased the survival chances decreases.
4. There are some cases where having small axillary nodes values doesn't guarantee the survival of the patients

#Further Observations:

1. As we have seen that axillary nodes can classify the data upto some extent .
2. We can see the data points of axillary nodes and age so that we can find something to classify

In [74]:

```
# points where the age is less than 35 and count of auxil_nodes count <=3 and survived
print("Patients whose age < 40 and axil_nodes <=3 among the whole data set : {}".format(len(haberman_data[(haberman_data["Age"]<40) & (haberman_data["axil_nodes"]<=3)])))

print("Patients whose age > 40 and axil_nodes <=3 among the whole data set : {}".format(len(haberman_data[(haberman_data["Age"]>40) & (haberman_data["axil_nodes"]<=3)])))

print("Patients whose age > 40 and axil_nodes <=3 among 225 data points for class 1 : {}".format(len(class1_data[(class1_data["Age"]>40) & (class1_data["axil_nodes"]<=3)])))
```

Patients whose age < 40 and axil_nodes <=3 among the whole data set : 27

Patients whose age > 40 and axil_nodes <=3 among the whole data set : 187

Patients whose age > 40 and axil_nodes <=3 among 225 data points for class 1 : 150

In [75]:

```
# points where the age is less than 35 and count of auxil_nodes count <=3 and survived
print("Patients whose age > 40 and axil_nodes < 3 among the whole data set : {}".format(len(haberman_data[(haberman_data["Age"]>40) & (haberman_data["axil_nodes"]<3)])))

print("Patients whose age > 40 and axil_nodes >3 among the whole data set : {}".format(len(haberman_data[(haberman_data["Age"]>40) & (haberman_data["axil_nodes"]>3)])))

print("Patients whose age > 40 and axil_nodes > 3 among 225 data points for class 1 : {}".format(len(class1_data[(class1_data["Age"]>40) & (class1_data["axil_nodes"]>3)])))
```

Patients whose age > 40 and axil_nodes < 3 among the whole data set : 169

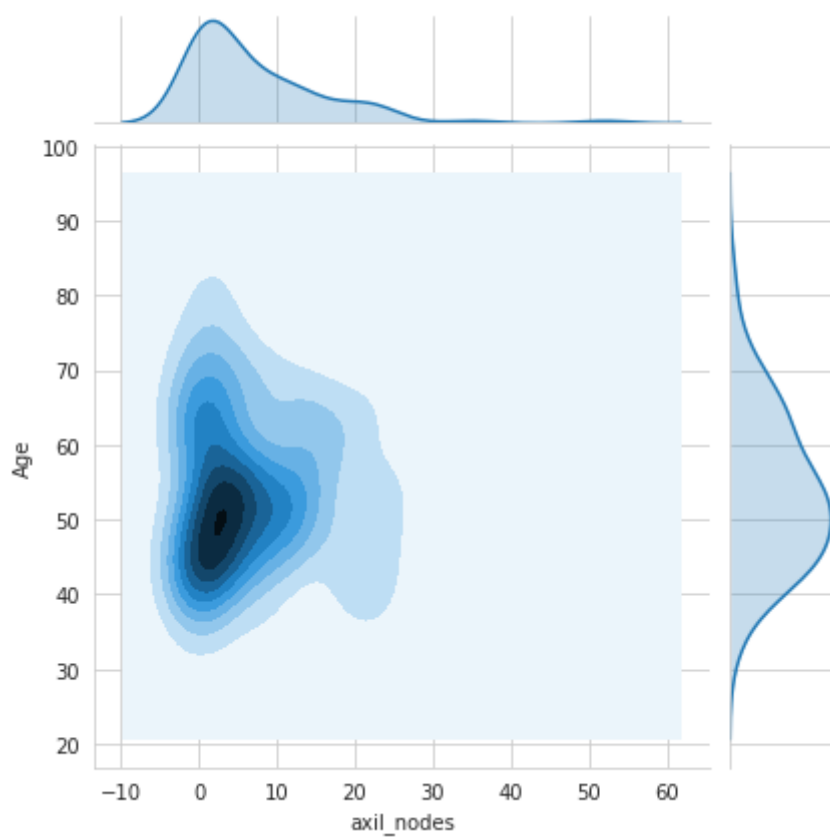
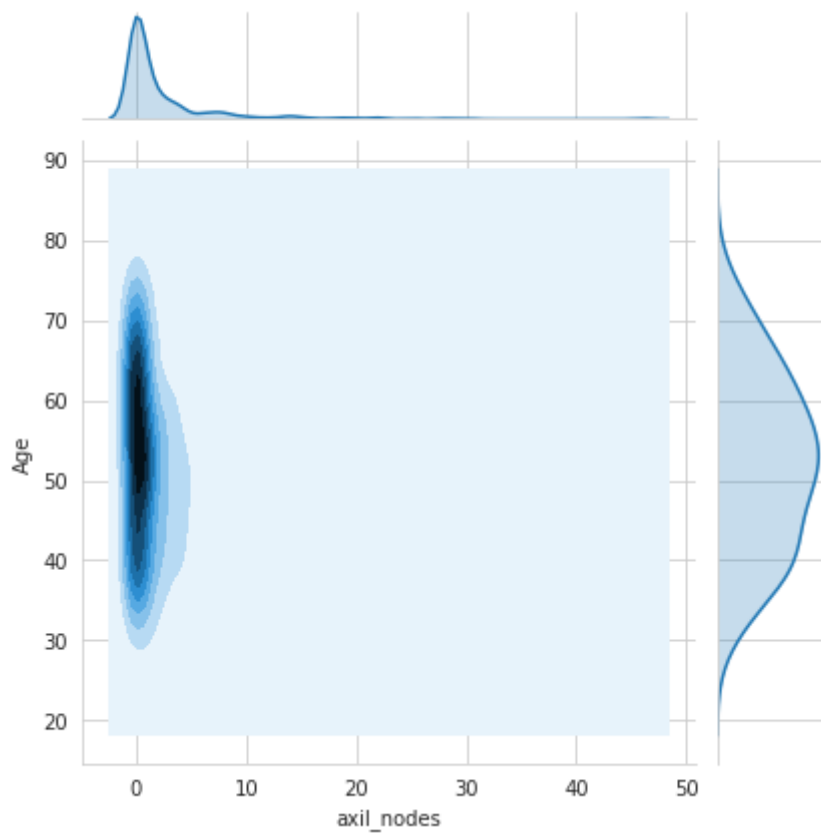
Patients whose age > 40 and axil_nodes >3 among the whole data set : 76

Patients whose age > 40 and axil_nodes > 3 among 225 data points for class 1 : 36

Contour Plot

In [76]:

```
sns.jointplot(x="axil_nodes",y="Age",data=class1_data,kind="kde")  
  
plt.show()  
sns.jointplot(x="axil_nodes",y="Age",data=class2_data,kind="kde")  
  
plt.show()
```



#Observations

1. Contour plots didn't tell much about the the class :
2. But we can see that most of points in the class 1 have axil_nodes < 3 and
3. In case of class 2 most of the points are having axil_nodes > 3

FINAL CONCLUSION

Final Observations:

1. Age and Year of Operation is not a deciding feature for classification
2. Smaller the no of the positive axillary nodes higher the survival chances.
3. Larger the positive axillary nodes count the chances of survival of the patient decreases . But there are some cases
that also tell that smaller the axillary count doesn't make sure the survival of the patient as patients with 0 axillary counts were not able to survive for more than 5 years.
4. From the data analysis we can say that

```
if positive axillary nodes < 3 its class is 1
if age > 40 and positive axillary nodes < 3 chances are that it can
lie in class 1 fewer chances for it to lie
in class 2
```

5. There can be mistakes in clasifying the data and chances are higher.
6. Classification with these feature is a difficult task and can misclassify for a particular data point