In [19]:

```
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
haberman=pd.read_csv("C:\\Users\\RAJ\\Desktop\\haberman.csv")
haberman
```

Out[19]:

	age	Op_year	Axil_nodes	Srv_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
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

	age	Op_year	Axil_nodes	Srv_status
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 [20]:

```
# Q) what is the total no of columns and rows
haberman.shape
```

Out[20]:

```
(306, 4)
```

total no of rows=306
total no of columns=4

In [21]:

```
# Q)Name the columns present in data
haberman.columns
```

Out[21]:

```
Index(['age', 'Op_year', 'Axil_nodes', 'Srv_status'], dtype='object')
```

Here total no of columns is 4

In [22]:

```
# Q) how many patients from survey status 1 & 2
haberman["Srv_status"].value_counts()
```

Out[22]:

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

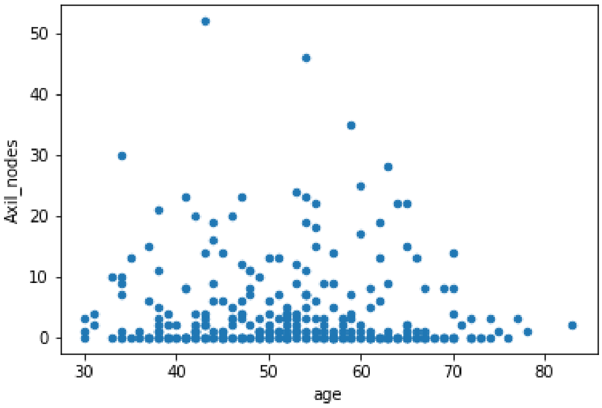
here we see that no of patients who survive 5 year or longer after operation i.e 225

here we see that no of patients who survive 5 year or longer after operation i.e 225
and no of patients who died within 5 year after operation i.e 81

PLOTTING
2-D Scatter Plot

In [30]:

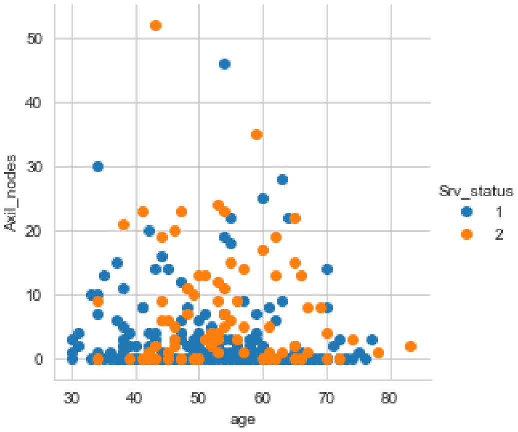
```
haberman.plot(kind="scatter", x="age", y="Axil_nodes");
plt.show()
```



this plot describes that how many positive axillary node detected for age group 30 to 85
maximum positive axillary node counted from 50-60 age group

In [39]:

```
sns.set_style("whitegrid");
sns.FacetGrid(haberman, hue="Srv_status", height=4) \
    .map(plt.scatter, "age", "Axil_nodes") \
    .add_legend();
plt.show();
```

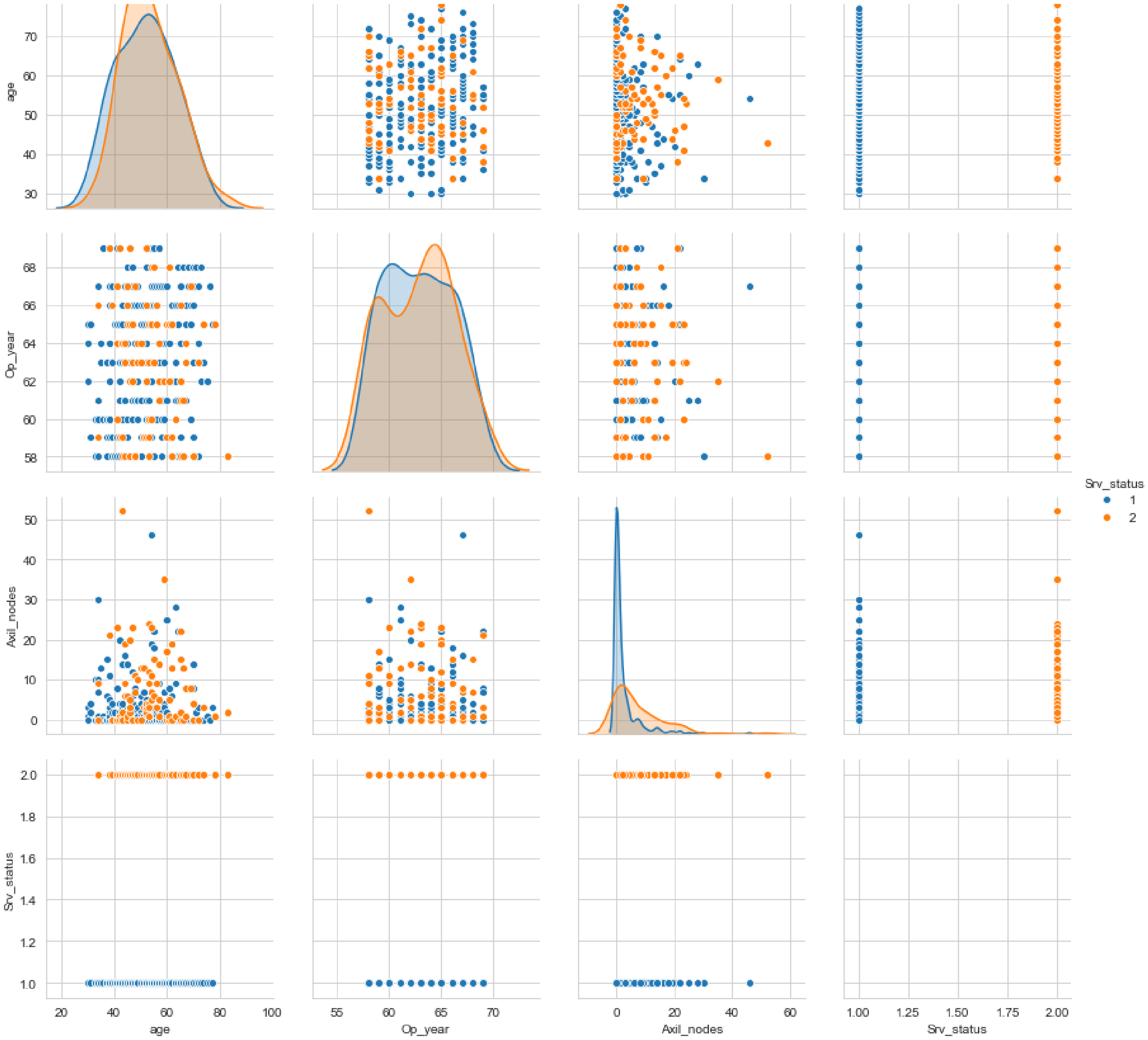


obervation
here both status 1 & 2 are harder to seperate

pair plot

In [45]:

```
plt.close()
sns.set_style("whitegrid")
sns.pairplot(haberman, hue="Srv_status", height=3)
plt.show()
```

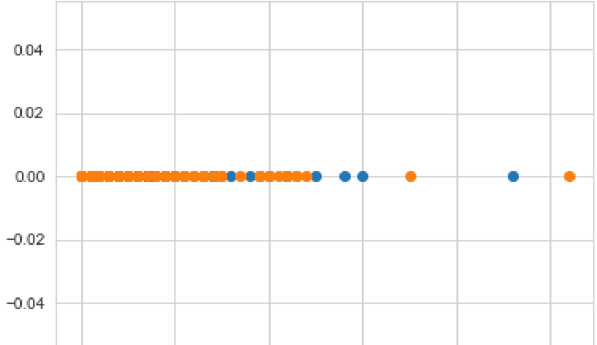


observation
here it is dificult to seperate survey status

Histogram, PDF, CDF

In [56]:

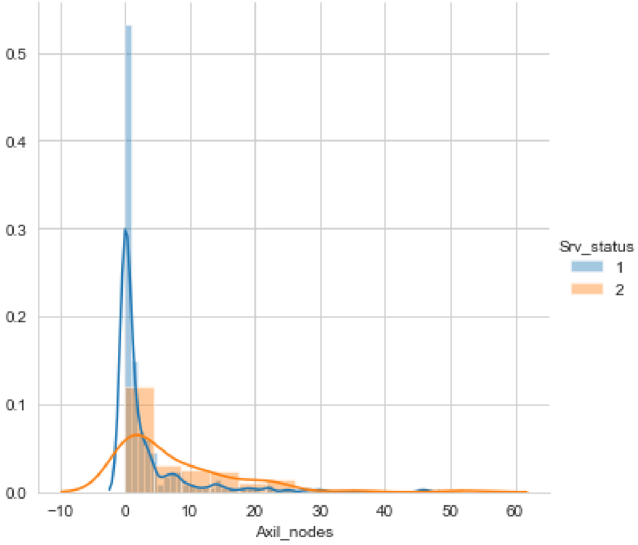
```
import numpy as np
haberman_1= haberman.loc[haberman["Srv_status"] == [1]];
haberman_2= haberman.loc[haberman["Srv_status"] == [2]];

plt.plot(haberman_1["Axil_nodes"], np.zeros_like(haberman1["Axil_nodes"]), 'o')
plt.plot(haberman_2["Axil_nodes"], np.zeros_like(haberman2['Axil_nodes']), 'o')

plt.show()
```



PDFs

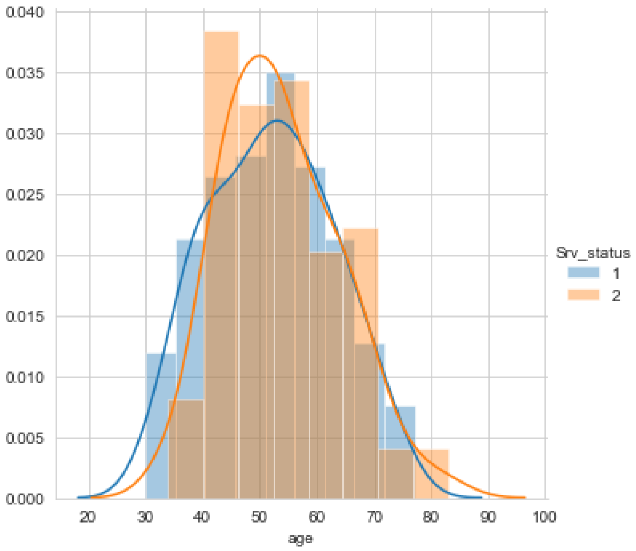
In [59]:

```
sns.FacetGrid(haberman, hue="Srv_status",height=5)\
.map(sns.distplot, "Axil_nodes")\
.add_legend()
plt.show()
```



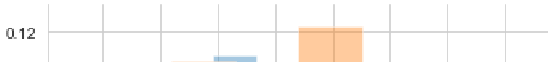
In [60]:

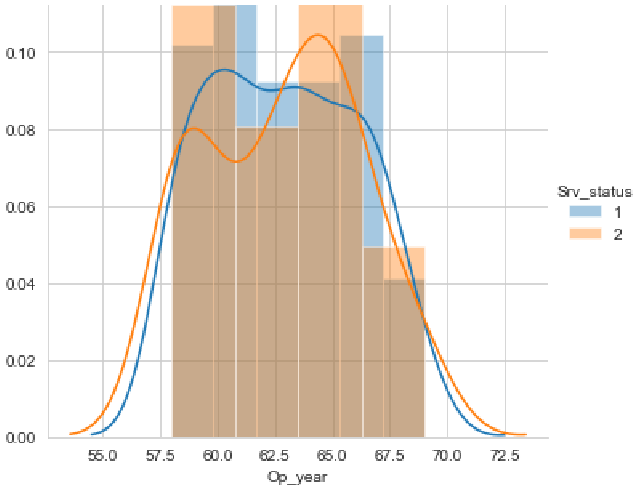
```
sns.FacetGrid(haberman, hue="Srv_status", height=5)\
.map(sns.distplot, "age")\
.add_legend()
plt.show()
```



In [62]:

```
sns.FacetGrid(haberman, hue="Srv_status", height=5)\
.map(sns.distplot, "Op_year")\
.add_legend()
plt.show()
```



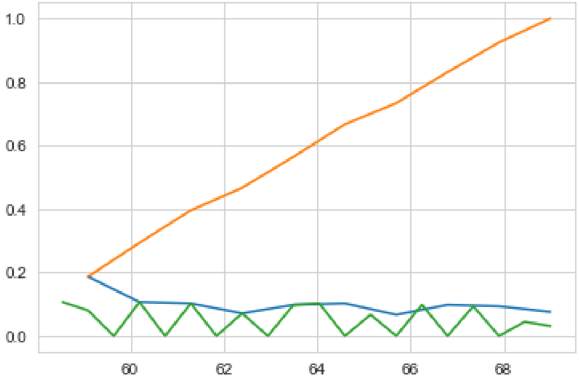


histogram using kde

In [66]:

```
counts, bin_edges = np.histogram(haberman_1['Op_year'], 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(haberman_1['Op_year'], bins=20, density = True)
pdf = counts/(sum(counts))
plt.plot(bin_edges[1:],pdf);


plt.show();
```

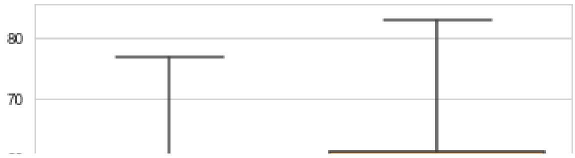
```
[0.18666667 0.10666667 0.10222222 0.07111111 0.09777778 0.10222222
 0.06666667 0.09777778 0.09333333 0.07555556]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
```

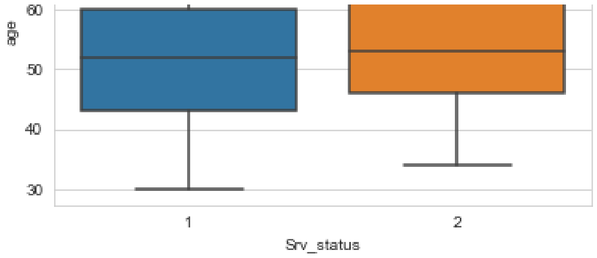


Box plot

In [69]:

```
sns.boxplot(x='Srv_status', y='age',data=haberman)
plt.show()
```



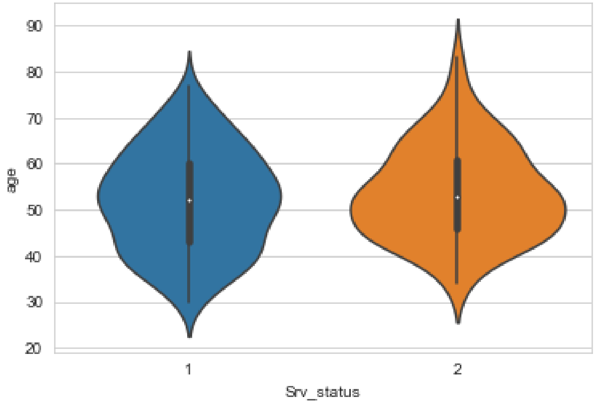


observation
here box plot between age and survey status. that briefs that 25th ,50th and 75th percentile survey status of data are approximately from same age group

violin plot

In [72]:

```
sns.violinplot(x='Srv_status', y='age',data=haberman)
plt.show()
```



observation
here the internal box shows 25th, 50th, 75th percentage whereas whispers shows that minimum and maximum age

In [ ]: