Objective-Based on the feature age,operation_year,axil_nodes understanding the survival of patient.

In [1]:

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

In [2]:

```
data=pd.read_csv("haberman.csv")
#loading the data in dataframe "data". The file format is csv hence we used read_csv() function.
```

In [3]:

```
data.tail()
#used tail() function to know the last 5 records of the datasets.
```

Out[3]:

	age	operation_year	axil_nodes	status
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

In [4]:

```
data.head()
#used head() function to know the first 5 records of the datasets.
```

Out[4]:

	age	operation_year	axil_nodes	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

In [5]:

```
data.shape
#shape attribute tells that "data" datasets having 306 rows and 4 columns are there.
```

Out[5]:

```
(306, 4)
```

In [6]:

```
print(data.columns)

#age--->age of the patient when the operation done.
#operation_year------>dataset contains cases from a study that was conducted between 1958 and 1969
.operation_year refers to the particular year.
#axil_nodes------>Number of positive axillary nodes detected
#status---->1 means patient survived 5 years or longer and 2 means the patient died within 5 year
```

```
Index(['age', 'operation_year', 'axil_nodes', 'status'], dtype='object')
```

In [7]:

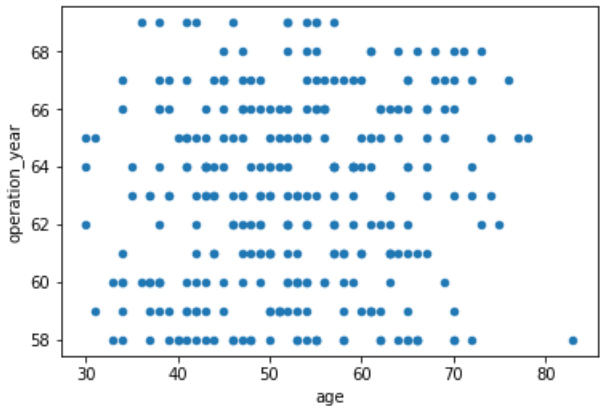
```
data["status"].value_counts()

#1 means patient survived 5 years or longer and the count is 225 and 2 means the patient died with
in 5 year and the count is 81

# this clearly tells that haberman cancer survival datasets having more patient who survived 5 yea
rs or longer than the patient died within 5 year

#datasets related to haberman haberman cancer survival is a imbalanced datasets as the datapoints
for each class i.e. 1(patient survived 5 years or longer) is 225 and 2(patient died within 5 year)
is 81
```

Out[7]:

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

In [8]:

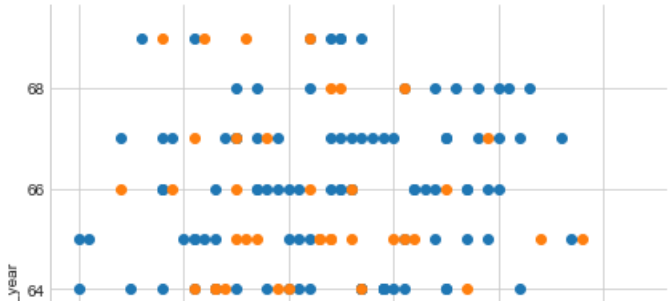
```
data.plot(kind="scatter",x="age",y="operation_year")
plt.show()
```



Observation ### cannot make any observation as the data overlappingin the scatter plot above.

In [35]:

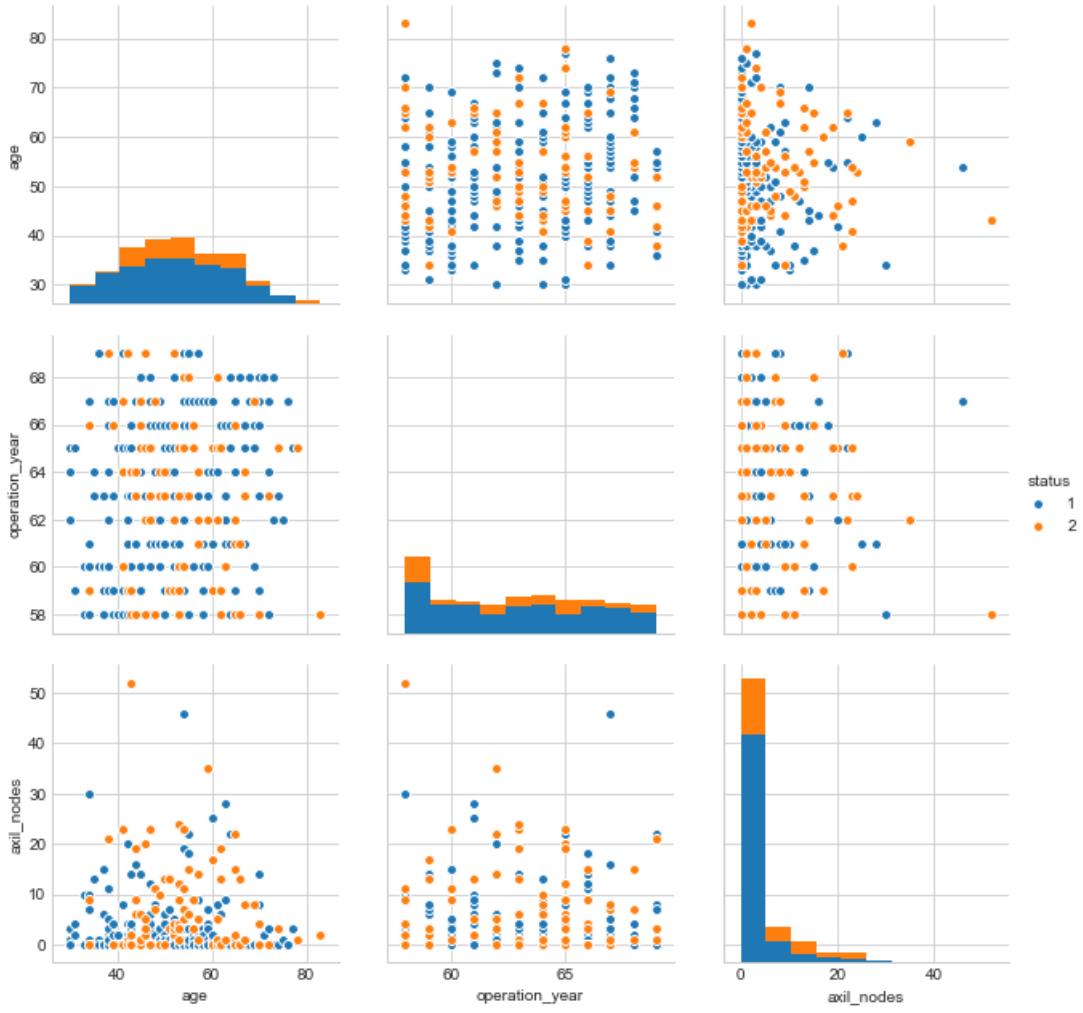
```
sns.set_style("whitegrid")
sns.FacetGrid(data, hue="status", size=6).map(plt.scatter, "age", "operation_year").add_legend();
plt.show()
```



observations #---->blue and orange points are overlapping/also very close to each other and hence cannot separate blue points from orange points.

In [39]:

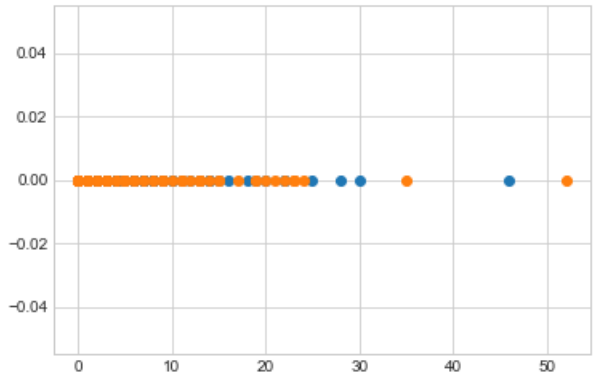
```
#2D scatter plot in combination of feature

plt.close()

sns.set_style("whitegrid");
sns.pairplot(data,vars=["age","operation_year","axil_nodes",],hue="status",size=3);
plt.show()
```



observation # Blue points and orange points are very close and overlapping with each other hence cannot make any observation.

In [41]:

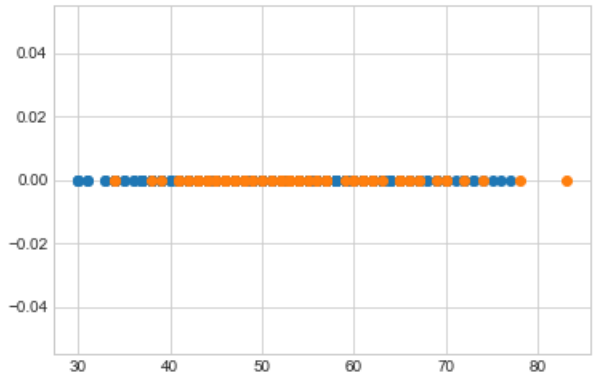
```
import numpy as np

data_1=data.loc[data["status"]==1]


data_2=data.loc[data["status"]==2]

plt.plot(data_1["axil_nodes"],np.zeros_like(data_1["axil_nodes"]),"o")
plt.plot(data_2["axil_nodes"],np.zeros_like(data_2["axil_nodes"]),"o")

plt.show()
```



observation #Blue points and orange points are very close and overlapping with each other hence cannot make any observation.

In [59]:

```
import numpy as np

data_1=data.loc[data["status"]==1]


data_2=data.loc[data["status"]==2]

plt.plot(data_1["age"],np.zeros_like(data_1["age"]),"o")
plt.plot(data_2["age"],np.zeros_like(data_2["age"]),"o")

plt.show()
```

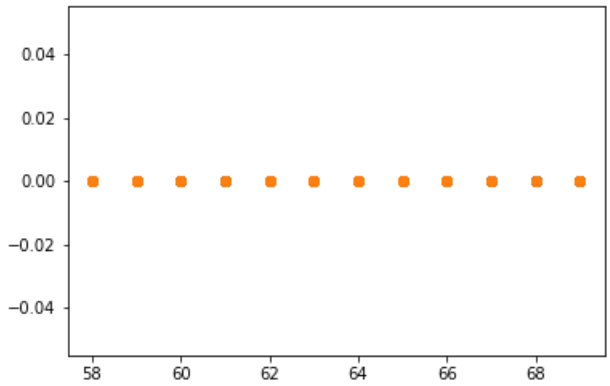


observation # Blue points and orange points are very close and overlapping with each other hence cannot make any observation.

In [18]:

```
import numpy as np
```

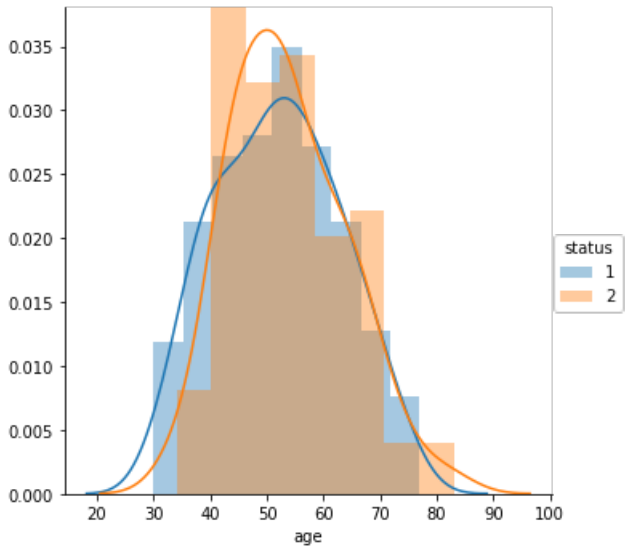
```
data_1=data.loc[data["status"]==1]


data_2=data.loc[data["status"]==2]

plt.plot(data_1["operation_year"],np.zeros_like(data_1["operation_year"]),"o")
plt.plot(data_2["operation_year"],np.zeros_like(data_2["operation_year"]),"o")

plt.show()
```



observation # Blue points and orange points are very close and overlapping with each other hence cannot make any observation.

In [19]:

```
import warnings
warnings.filterwarnings("ignore")

sns.FacetGrid(data, hue="status", size=5) \
   .map(sns.distplot, "age") \
   .add_legend();
plt.show();
```

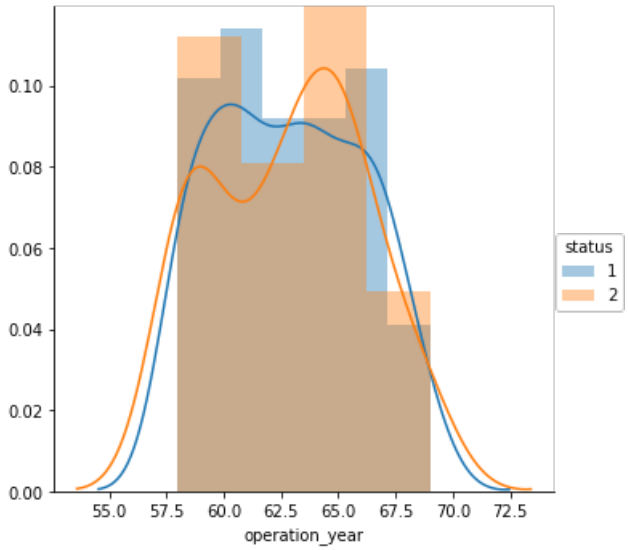


obeservation #-cannot make any observation as the data is overlapping.

In [20]:

```
sns.FacetGrid(data, hue="status", size=5) \
   .map(sns.distplot, "operation_year") \
   .add_legend();
```

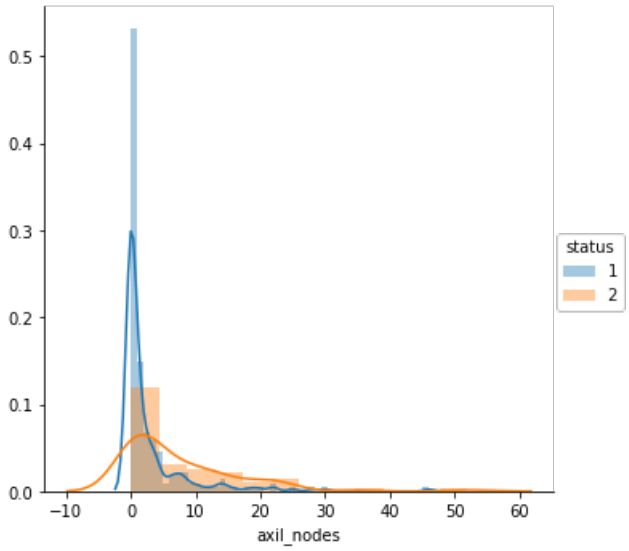
```
plt.show();
```



obeservation #-cannot make any observation as the data is overlapping.

In [21]:

```
sns.FacetGrid(data, hue="status", size=5) \
   .map(sns.distplot, "axil_nodes") \
   .add_legend();
plt.show();
```



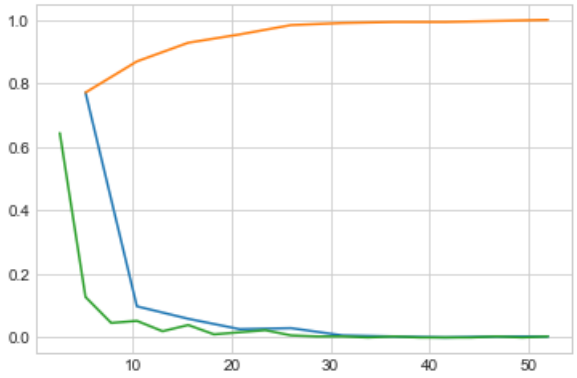
observation # when the auxilary nodes are between 0 to 3 then the survival of patient is increased and with increased auxilary nodes the survival of patient decreased.

In [53]:

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

plt.show();
```

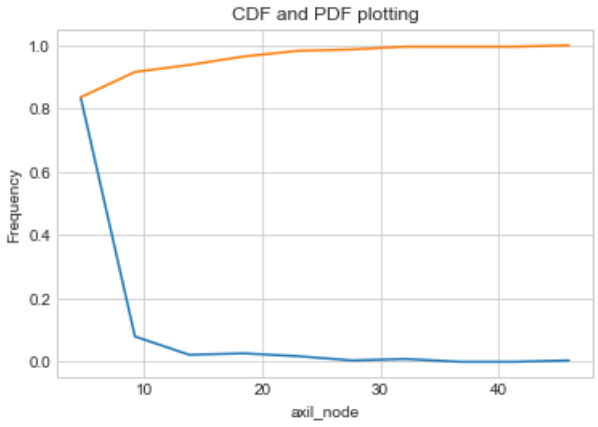
```
[0.77124183 0.09803922 0.05882353 0.02614379 0.02941176 0.00653595
 0.00326797 0.         0.00326797 0.00326797]
[ 0.   5.2 10.4 15.6 20.8 26.  31.2 36.4 41.6 46.8 52. ]
```



In [64]:

```
counts, bin_edges = np.histogram(data_1['axil_nodes'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)

#compute CDF
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:], cdf)



plt.xlabel('axil_node')
plt.ylabel('Frequency')
plt.title('CDF and PDF plotting')

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. ]
```

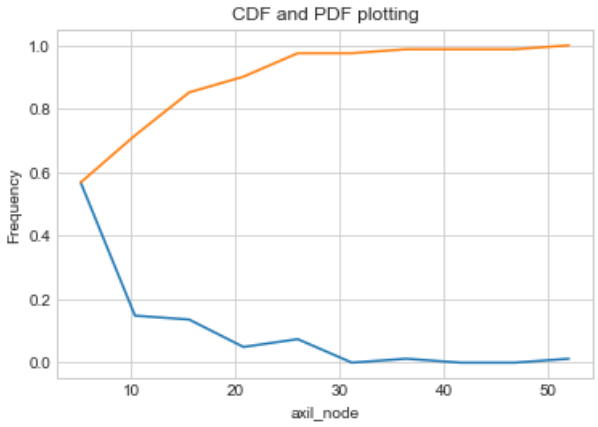


In [65]:

```
counts, bin_edges = np.histogram(data_2['axil_nodes'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
```

```
print(pdf);
print(bin_edges)

#compute CDF
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:], cdf)

plt.xlabel('axil_node')
plt.ylabel('Frequency')
plt.title('CDF and PDF plotting')

plt.show();
```

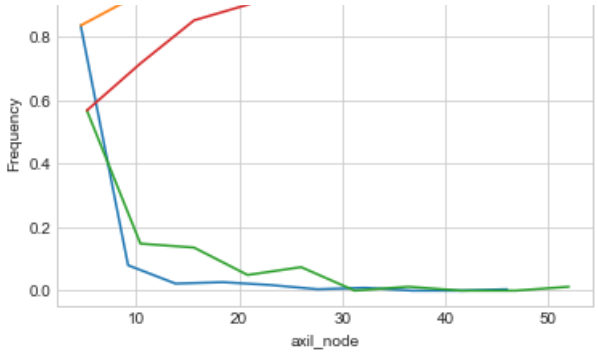
```
[0.56790123 0.14814815 0.13580247 0.04938272 0.07407407 0.
 0.01234568 0.         0.         0.01234568]
[ 0.   5.2 10.4 15.6 20.8 26.  31.2 36.4 41.6 46.8 52. ]
```



In [66]:

```
counts, bin_edges = np.histogram(data_1['axil_nodes'], 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(data_2['axil_nodes'], 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)


plt.xlabel('axil_node')
plt.ylabel('Frequency')
plt.title('CDF and PDF plotting')


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. ]
[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. ]
```



In [25]:

```
print("Means:")
#Mean with an outlier.
#print(np.mean(np.append(data["axil_nodes"],50)));
print(np.mean(data_1["axil_nodes"]))
print(np.mean(data_2["axil_nodes"]))

print("\nStd-dev:");
print(np.std(data_1["axil_nodes"]))
print(np.std(data_2["axil_nodes"]))
```

```
Means:
2.7911111111111113
7.45679012345679

Std-dev:
5.857258449412131
9.128776076761632
```

observation #---->The average value of axil_nodes of the patient status 1 is 2.8

observation---->The average value of axil_nodes of the patient status 1 is 7.45

observation # This clearly states patient who having less auxillary nodes survival is more than more and patient havong more auxillary node survival is less.

observation #standard deviation value of axil_nodes for status 1 patient is 5.85 which tells the spread value of axil_nodes around the average.

observation # standard deviation value of axil_nodes for status 2 patient is 9.12 which tells the spread value of axil_nodes around the average.

In [26]:

```
print("Means:")
#Mean with an outlier.
#print(np.mean(np.append(data["axil_nodes"],50)));
print(np.mean(data_1["age"]))
print(np.mean(data_2["age"]))

print("\nStd-dev:");
print(np.std(data_1["age"]))
```

```
print(np.std(data_2["age"]))
```

```
Means:
52.01777777777778
53.67901234567901

Std-dev:
10.98765547510051
10.10418219303131
```

observation---->The average value of age for status 1 patient is 52.01 and the spread around average is 10.98

observation---->The average value of age for status 2 patient is 53.67 and the spread around average is 10.10

We cannot make any inference based on the age at which patient undergone operation. As the value of mean and standard deviation of age the status1 and status 2 is very close to each other.

In [85]:

```
print("Means:")
#Mean with an outlier.
#print(np.mean(np.append(data["axil_nodes"],50)));
print(np.mean(data_1["operation_year"]))
print(np.mean(data_2["operation_year"]))

print("\nStd-dev:");
print(np.std(data_1["operation_year"]))
print(np.std(data_2["operation_year"]))
```

```
Means:
62.86222222222222
62.82716049382716

Std-dev:
3.2157452144021956
3.3214236255207883
```

observation---->The average value of operation_year for status 1 patient is 52.01 and the spread around average is 62.86

observation---->The average value of operation_year for status 2 patient is 53.67 and the spread around average is 62.82

We cannot make any inference based on the operation_year at which patient undergone operation. As the value of mean and standard deviation of age the status1 and status 2 is very close to each other.

In [93]:

```
data_1["axil_nodes"]
```

Out[93]:

```
0       1
1       3
```

```
2       0
3       2
4       4
5      10
6       0
9      30
10      1
11     10
12      7
13      0
14     13
15      0
16      1
17      0
18      0
19      0
20      0
21      6
22     15
23      0
25      2
26      0
27      0
28      3
29      1
30      0
31     11
32      1
       ..
265     2
266     0
267     1
270     0
271     1
272     0
275     0
276     0
277     0
278     0
279     0
280     0
282     0
283     0
284     0
287    14
288     0
289     0
290     8
291     0
292     2
294     0
295     0
296     3
297     0
298     0
300     0
301     1
302     0
303     3
Name: axil_nodes, Length: 225, dtype: int64
```

In [86]:

```
#Median, Quantiles, Percentiles, IQR.
print("\nMedians:")
print(np.median(data_1["axil_nodes"]))
print(np.median(data_2["axil_nodes"]))

print("\nQuantiles:")
print(np.percentile(data_1["axil_nodes"],np.arange(0, 100, 25)))
print(np.percentile(data_2["axil_nodes"],np.arange(0, 100, 25)))

print("\n90th Percentiles:")
print(np.percentile(data_1["axil_nodes"],90))
print(np.percentile(data_2["axil_nodes"],90))

from statsmodels import robust
```

```
from statsmodels import robust
print ("\nMedian Absolute Deviation")
print(robust.mad(data_1["axil_nodes"]))
print(robust.mad(data_2["axil_nodes"]))
```

```
Medians:
0.0
4.0

Quantiles:
[0. 0. 0. 3.]
[ 0.  1.  4. 11.]

90th Percentiles:
8.0
20.0

Median Absolute Deviation
0.0
5.930408874022408
```

observation--->median value for axil_nodes for status 1 patient is 0 the reason is if we closely observe data_1["axil_nodes"] most of the value are 0 and if we sort them and find themiddle value then it get value of 0 #observation--->median value for axil_nodes for status 2 patient is 4 the reason is if we closely observe data_2["axil_nodes"] most of the value are 0 and if we sort them and find themiddle value then it get value of 0 #observation-Clearly we can say status 1 patient having axil_nodes value less than status 2 patient. #observation--->50 percentile value of axil node is 0 for status 1 patient #observation--->75 percentile value of axil node is 3 for status 1 patient #observation--->remaining percentile value of axil node is more than 3 #observation--->50 percentile value of axil node is 4 for status 2 patient #observation--->75 percentile value of axil node is 11 for status 2 patient #observation--->remaining percentile value of axil node is more than 11 #observation---->90 percentile value of axil_nodes is below 8 #10 percentile value of axil_nodes is more than 8 #MAD of axil_nodes for status 1 patient is same as median for status 1 patient. #MAD of axil_nodes for status 2 patient is increased compared to median of status 2 patient.

In [87]:

```
#Median, Quantiles, Percentiles, IQR.
print("\nMedians:")
print(np.median(data_1["age"]))
print(np.median(data_2["age"]))

print("\nQuantiles:")
print(np.percentile(data_1["age"],np.arange(0, 100, 25)))
print(np.percentile(data_2["age"],np.arange(0, 100, 25)))

print("\n90th Percentiles:")
print(np.percentile(data_1["age"],90))
print(np.percentile(data_2["age"],90))

from statsmodels import robust
print ("\nMedian Absolute Deviation")
print(robust.mad(data_1["age"]))
print(robust.mad(data_2["age"]))
```

```
Medians:
52.0
53.0

Quantiles:
[30. 43. 52. 60.]
[34. 46. 53. 61.]

90th Percentiles:
67.0
67.0

Median Absolute Deviation
13.343419966550417
11.860817748044816
```

#cannot make any observations as the data lies very closely or overlappings.

In [91]:

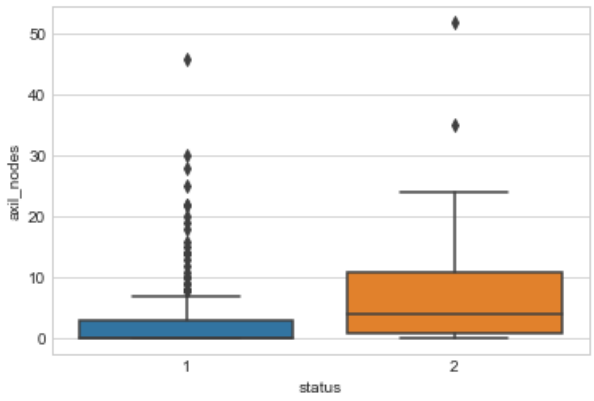
```
#Median, Quantiles, Percentiles, IQR.
print("\nMedians:")
print(np.median(data_1["operation_year"]))
print(np.median(data_2["operation_year"]))

print("\nQuantiles:")
print(np.percentile(data_1["operation_year"],np.arange(0, 100, 25)))
print(np.percentile(data_2["operation_year"],np.arange(0, 100, 25)))

print("\n90th Percentiles:")
print(np.percentile(data_1["operation_year"],90))
print(np.percentile(data_2["operation_year"],90))

from statsmodels import robust
print ("\nMedian Absolute Deviation")
print(robust.mad(data_1["operation_year"]))
print(robust.mad(data_2["operation_year"]))
```

```
Medians:
63.0
63.0

Quantiles:
[58. 60. 63. 66.]
[58. 59. 63. 65.]

90th Percentiles:
63.0
63.0

Median Absolute Deviation
4.447806655516806
4.447806655516806
```

#cannot make any observations as the data lies very closely or overlappings.

In [71]:

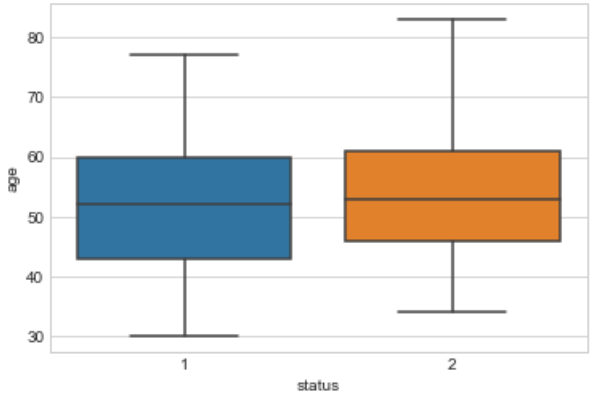
```
sns.boxplot(x='status',y='axil_nodes', data=data)
plt.show()

#observation--->For status 1 patient 25 percentile value lies in 0 and rest 50 to 75 perentile the value of axil_nodes is 4

#for status 2 patient 75 percentile value of axil_nodes value is above 10 and 50 percentile value of axil_nodes is between 4 to 10 and rest 25 percentile below 4
```



In [73]:

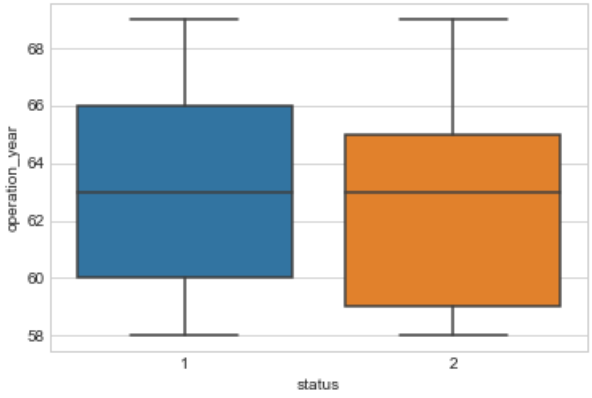
```
sns.boxplot(x='status',y='age', data=data)
plt.show()

#age of status 1 patient between 45 years to 60 years
```

```
#age of status 2 patient is slightly higher than status 1 patient.
```



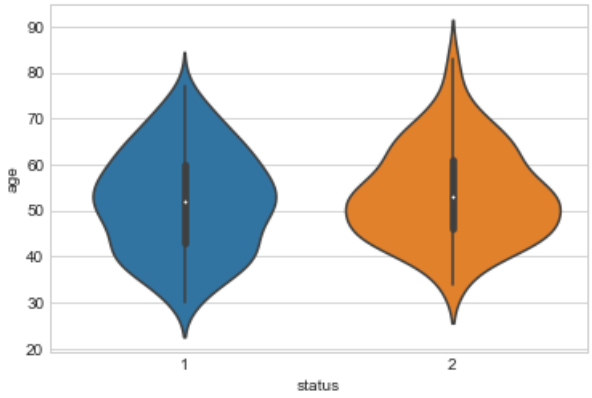
In [75]:

```
sns.boxplot(x='status',y='operation_year', data=data)
plt.show()

#25 percentile value of operation year for status 1 patient is 60
#50 percentile value of operation year for status 1 patient is 63
#75 percentile value of operation year for status 1 patient is 66

#25 percentile value of operation year for status 1 patient is below 60
#50 percentile value of operation year for status 1 patient is 63
#75 percentile value of operation year for status 1 patient is below 66
```



In [78]:

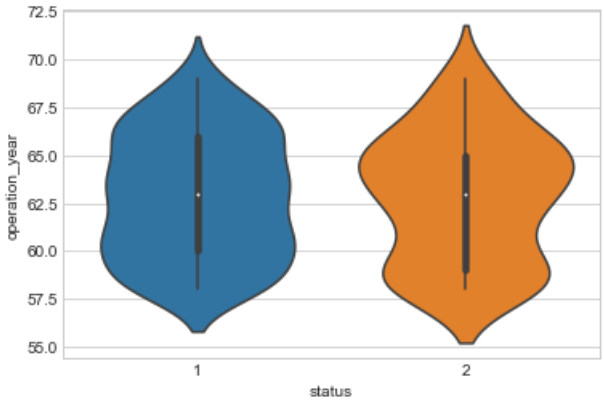
```
sns.violinplot(x="status", y="age", data=data, size=8)
plt.show()

# the value of age for both status 1 and status 2 patient is more dense betwen 40 to 60
```



In [80]:

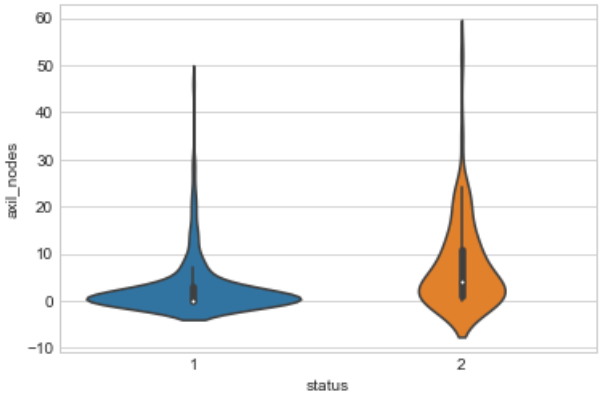
```
sns.violinplot(x="status", y="operation_year", data=data, size=8)
plt.show()
```

```
#The status 1 patient are more dense between 57.5 to 60 and status 2 patient are more dense betwee
n 62.5 to 66.0
```



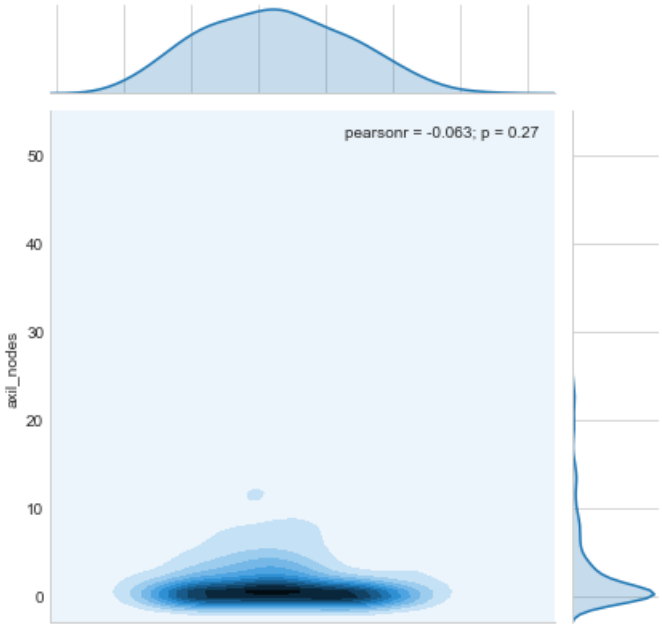
In [82]:

```
sns.violinplot(x="status", y="axil_nodes", data=data, size=8)
plt.show()

#through violin plot it can be observed as axil_nodes value for status 1 patient is more dense aro
und value 0 compared to status 2 axil_node value.
```



In [83]:

```
sns.jointplot(x="age", y="axil_nodes", data=data, kind="kde");
plt.show();

#the age of patient are more dense between 40 to 65 and corresponding axil_nodes value is ranging
from 0 to 7
```



Conclusion-Most of feature of status 1 and staus 2 patient is overlapping however auxil_nodes have significant relationship with patient survival.Patient with less auxil_nodes survival is more than compared to patient having more auxilary nodes.