Haberman's Servival Data Set

1.The Hebermans data set is collection of patiens who had undergone surgery for breast cancer. 2.The data collected in the years between 1958 to 1970. 3.The data set shows the servival of the patiens who undergone the surgery. 4.It contains four columns i.e (age,operatipn_year,axil_nodes,servival_status)

In [81]:
```
import pandas as pd #importing pandas library for data processing.
import numpy as np #importing numpy library for numerical operation like linear algebra matirx operation etc.
import matplotlib.pyplot as plt # importing matplotlib.pyplot for the ploting data.
import seaborn as sns
%matplotlib inline
```

In [82]:
```
df=pd.read_csv("haberman.csv")
df
```

Out[82]:

	age	year	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
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
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 [83]:
```
print(df.shape) #it is the shape of the data set,it contains 306 rows and 4 columns
```

```
(306, 4)
```

In [84]:
```
print(df.columns) # printing the columns of the data set
```

```
Index(['age', 'year', 'nodes', 'status'], dtype='object')
```

In [85]:
```
df["age"].value_counts() #printing the perticular age which occures often in data set
```

Out[85]:
```
52    14
54    13
50    12
47    11
53    11
43    11
57    11
55    10
65    10
49    10
38    10
41    10
61     9
45     9
42     9
63     8
59     8
62     7
44     7
58     7
56     7
46     7
70     7
34     7
48     7
37     6
67     6
60     6
51     6
39     6
66     5
64     5
72     4
69     4
40     3
30     3
68     2
73     2
74     2
36     2
35     2
33     2
31     2
78     1
71     1
75     1
76     1
77     1
83     1
Name: age, dtype: int64
```

In [86]:
```
df["year"][df["year"].value_counts().max()] #prints the highest number of surgeries done in the year
```

Out[86]: 67

In [87]:
```
df["nodes"].value_counts() #it shows the axil counts of the patiens
```

Out[87]:
```
0     136
1      41
2      20
3      20
4      13
6       7
7       7
8       7
5       6
9       6
13      5
14      4
11      4
10      3
15      3
19      3
22      3
23      3
12      2
20      2
46      1
16      1
17      1
18      1
21      1
24      1
25      1
28      1
30      1
35      1
52      1
Name: nodes, dtype: int64
```

In [88]:
```
df["status"].value_counts()
```

Out[88]:
```
1    225
2     81
Name: status, dtype: int64
```

observation 1.servival of the patients more than five years after operation is 228 2.servival of the patients less than five years after operation is 81

[inbalanced data]

In [89]:
```
df.head() #it prints the some sample first five rows dat from the data sat
```

Out[89]:

	age	year	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 [90]:
```
df.tail() #it prints the some sample of last rows of the data
```

Out[90]:

	age	year	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 [91]:
```
df["age"].max() # it prints the highest age of a person in the data set who undergone for the surgery
```

Out[91]: 83

In [92]:
```
df["age"].min() # it prints the lowest age of a person in the data set who undergone for surgery
```

Out[92]: 30

In [93]:
```
df.info() #prints the information of data set
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306 entries, 0 to 305
Data columns (total 4 columns):
age       306 non-null int64
year      306 non-null int64
nodes     306 non-null int64
status    306 non-null int64
dtypes: int64(4)
memory usage: 9.6 KB
```

In [94]:
```
df.describe()#it describe the values calculation of the data set
```

Out[94]:

	age	year	nodes	status
count	306.000000	306.000000	306.000000	306.000000
mean	52.457516	62.852941	4.026144	1.264706
std	10.803452	3.249405	7.189654	0.441899
min	30.000000	58.000000	0.000000	1.000000
25%	44.000000	60.000000	0.000000	1.000000
50%	52.000000	63.000000	1.000000	1.000000
75%	60.750000	65.750000	4.000000	2.000000
max	83.000000	69.000000	52.000000	2.000000

In [95]:
```
df["status"].value_counts(normalize = True)
```

Out[95]:
```
1    0.735294
2    0.264706
Name: status, dtype: float64
```

observation

73% of patients are servive more than five years after operation

27% of patients are not servive more than five years after operation

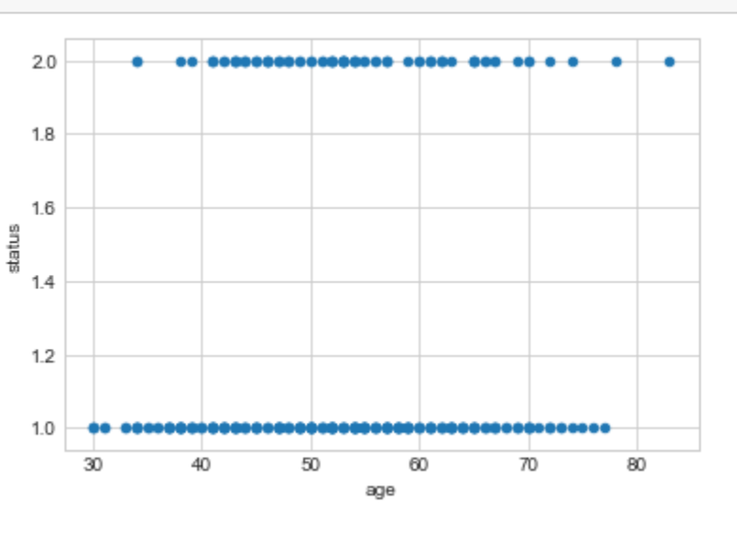
In [96]:
```
np.sum(df.isna())
```

Out[96]:
```
age       0
year      0
nodes     0
status    0
dtype: int64
```

no null values are present in the data set

2-D Scatter Plot

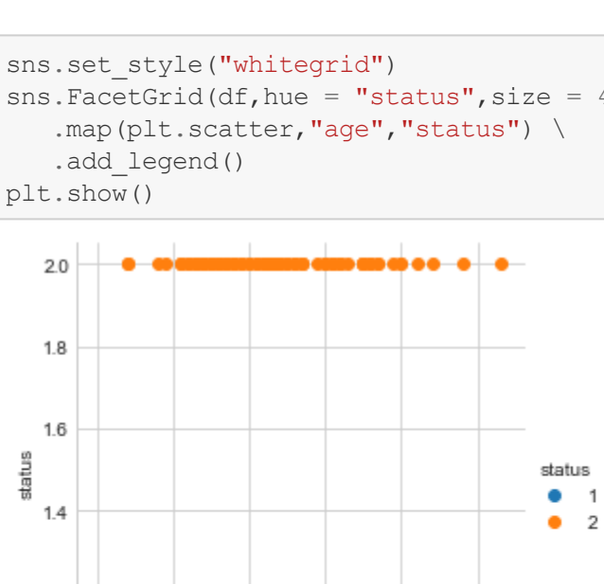
In [97]:
```
df.plot(kind = "scatter" , x = "age",y = "status")
plt.show()
```



observation

1.The above plot shows the age and servival rate of the persons 2.At the age of 30 there more servival rate who undergone surgery 3.As the increases there is less number of servival rate as compare to age decreses

In [98]:
```
sns.set_style("whitegrid")
sns.FacetGrid(df,hue = "status",size = 4) \
   .map(plt.scatter,"age","status") \
   .add_legend()
plt.show()
```



observations: 1.blue color is age and orange color is status 2.at the age of 30 there are more servival of patiens 3.at the above 40 to 5o there are less servival becaluse there is dense of status

In [99]:
```
plt.close()
sns.set_style("whitegrid")
sns.pairplot(df, hue = "status",size = 2)
plt.show()
```

observation:

1.Data cannot be separated linearly

73% of data can be servive more than five years after surgery

printing mean and standard deviation

In [123]: `df["age"].mean()`

Out[123]: 52.45751633986928

In [126]: `df["year"].mean()`

Out[126]: 62.85294117647059

In [133]: `df["nodes"].mean()`

Out[133]: 4.026143790849673

In [128]: `df["status"].mean()`

Out[128]: 1.2647058823529411

In [129]: `df["age"].std()`

Out[129]: 10.80345234930328

In [130]: `df["year"].std()`

Out[130]: 3.249404663223851

In [131]: `df["nodes"].std()`

Out[131]: 7.189653506248565

In [132]: `df["status"].std()`

Out[132]: 0.44189911885403554

Printing Median,Percentile,Quantile,IQR,MAD

In [137]: `df["age"].median()`

Out[137]: 52.0

In [141]: `df["year"].median()`

Out[141]: 63.0

In [144]: `df["nodes"].median()`

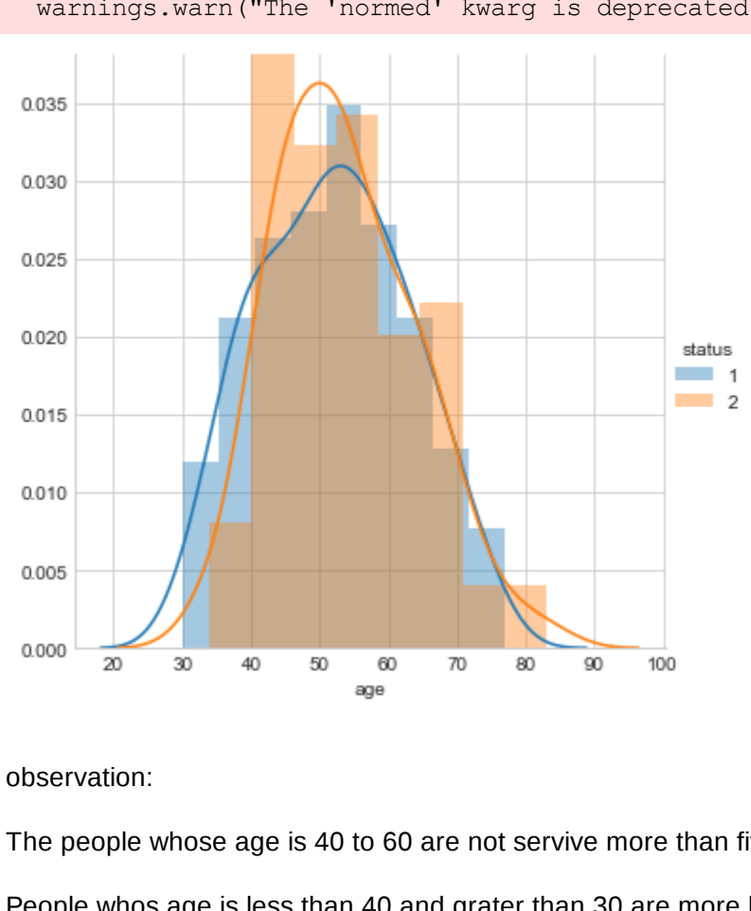
Out[144]: 1.0

In [145]: `df["status"].median()`

Out[145]: 1.0

In [100]:
```
sns.FacetGrid(df,hue = "status",size = 5) \
   .map(sns.distplot,"age") \
   .add_legend()
plt.show()
```

```
C:\Users\sagar\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\Users\sagar\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



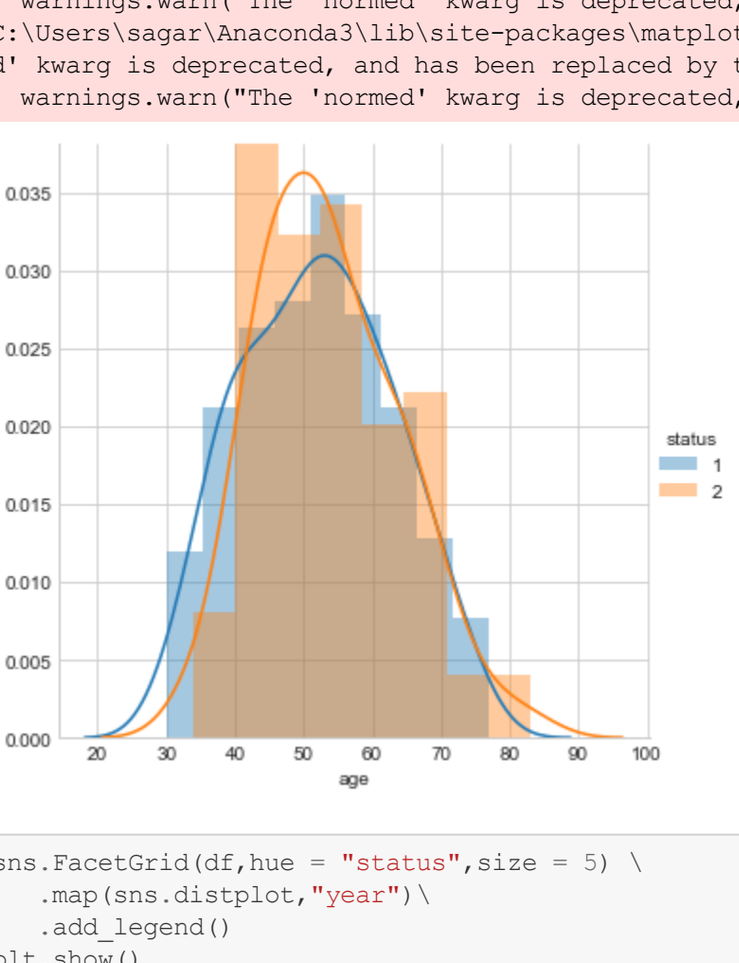
observation:

The people whose age is 40 to 60 are not servive more than five years after surgery

People whos age is less than 40 and grater than 30 are more likely to servive more than five years ofter surgery

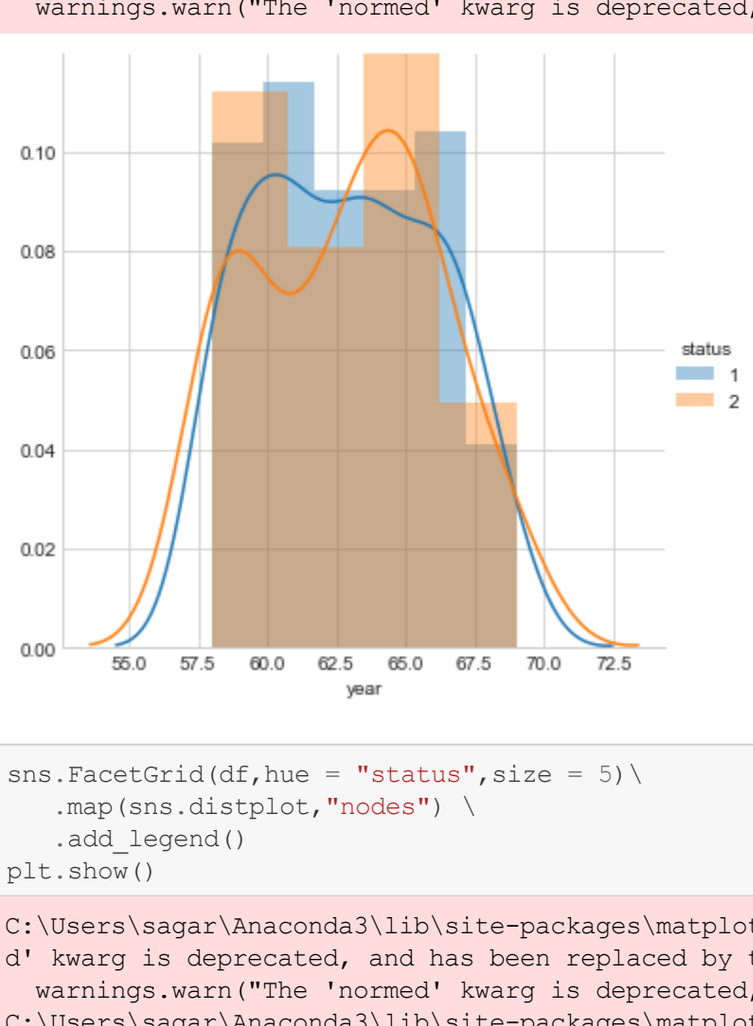
In [101]:
```
sns.FacetGrid(df,hue = "status",size = 5) \
   .map(sns.distplot,"age") \
   .add_legend()
plt.show()
```

```
C:\Users\sagar\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\Users\sagar\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



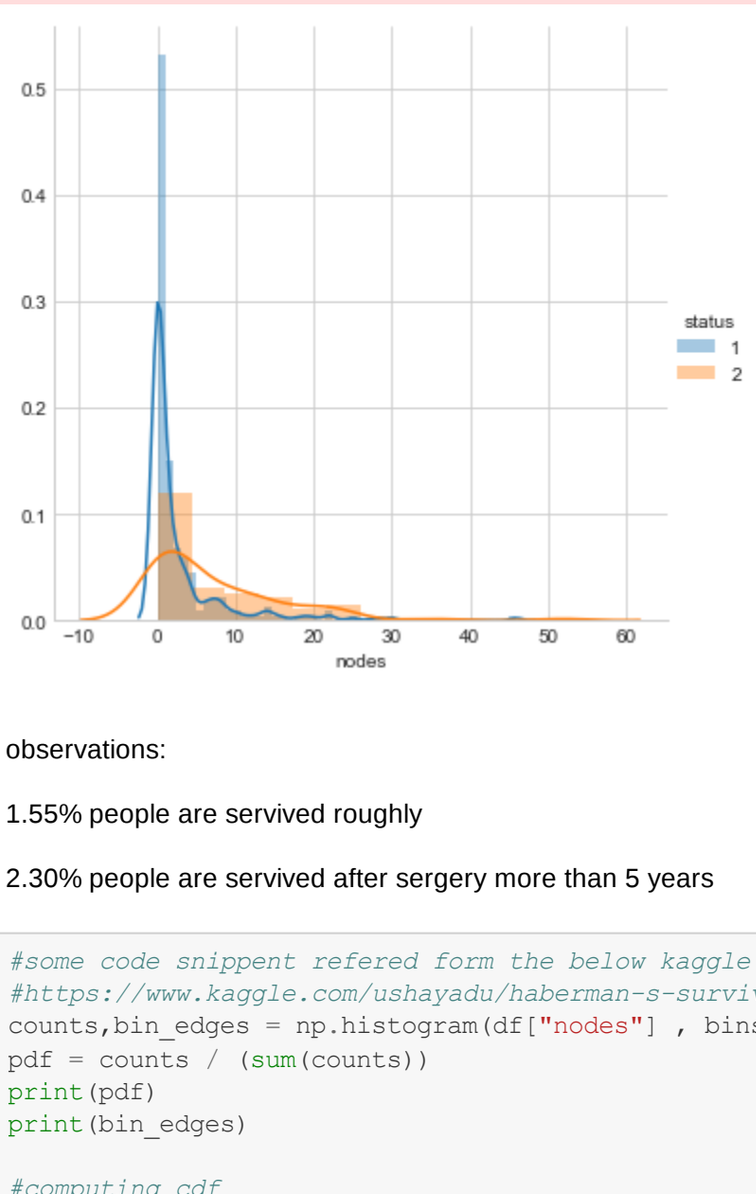
In [102]:
```
sns.FacetGrid(df,hue = "status",size = 5) \
   .map(sns.distplot,"year")\
   .add_legend()
plt.show()
```

```
C:\Users\sagar\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\Users\sagar\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



In [104]:
```
sns.FacetGrid(df,hue = "status",size = 5)\
   .map(sns.distplot,"nodes") \
   .add_legend()
plt.show()
```

```
C:\Users\sagar\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\Users\sagar\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



observations:

1.55% people are servived roughly

2.30% people are servived after sergery more than 5 years

In [1]:
```
#some code snippent refered form the below kaggle ling
#https://www.kaggle.com/ushayadu/haberman-s-survival-dataset-eda
counts,bin_edges = np.histogram(df["nodes"] , bins = 10,density = True)
pdf = counts / (sum(counts))
print(pdf)
print(bin_edges)

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

counts,bin_edges = np.histogram(df["age"] , bins = 10,density = True)
pdf = counts / (sum(counts))
print(pdf)
print(bin_edges)

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

counts,bin_edges = np.histogram(df["year"] , bins = 10,density = True)
pdf = counts / (sum(counts))
print(pdf)
print(bin_edges)

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

counts,bin_edges = np.histogram(df["status"] , bins = 10,density = True)
pdf = counts / (sum(counts))
print(pdf)
print(bin_edges)

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



plt.show()
```

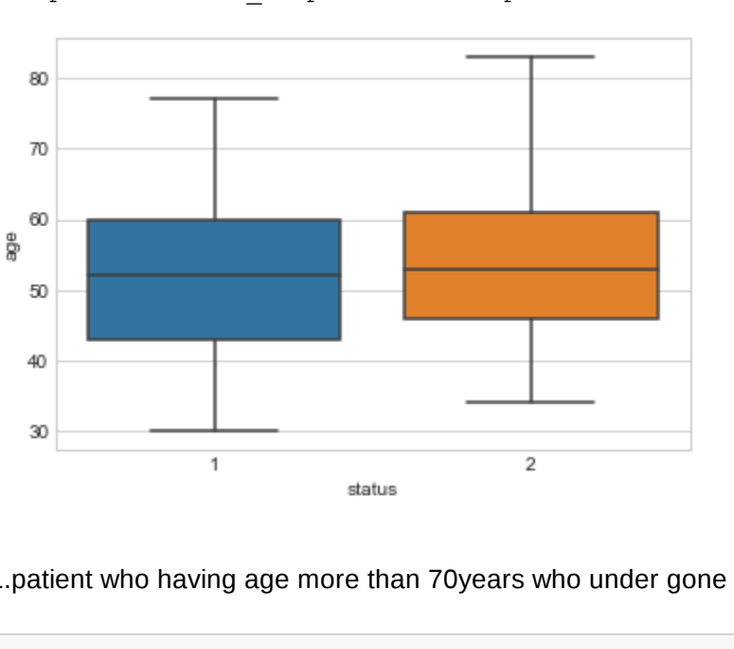
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-e373cb4b94d4> in <module>()
      1 #some code snippent refered form the below kaggle ling
      2 #https://www.kaggle.com/ushayadu/haberman-s-survival-dataset-eda
----> 3 counts,bin_edges = np.histogram(df["nodes"] , bins = 10,density = True)
      4 pdf = counts / (sum(counts))
      5 print(pdf)

NameError: name 'np' is not defined
```

observation:

age is the important feature in the data as age increases the servival rate is decreases

In [114]:
```
sns.boxplot(x = df["status"],y=df["age"])
```

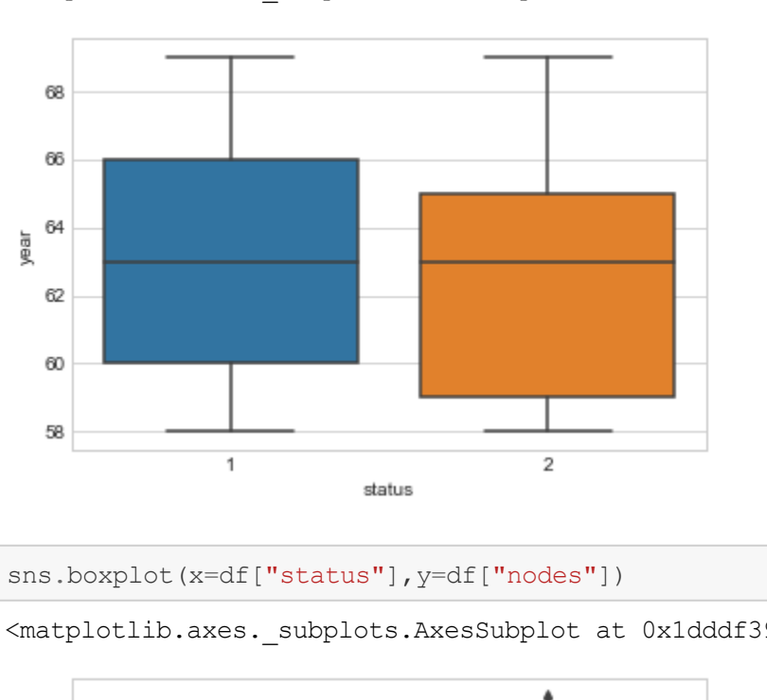
Out[114]: `<matplotlib.axes._subplots.AxesSubplot at 0x1dde0433b00>`



1.patient who having age more than 70years who under gone sergery will not servive more than five years

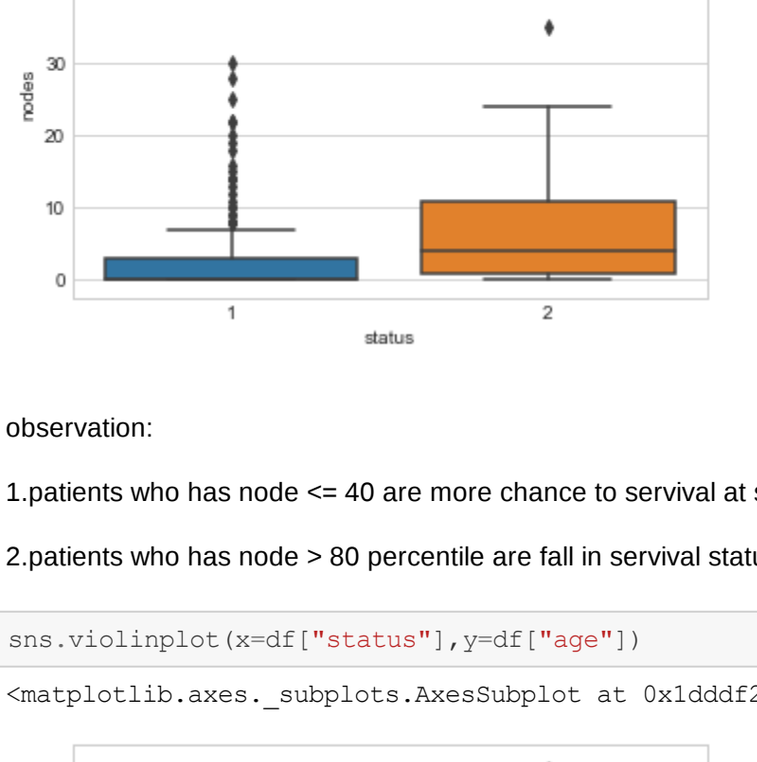
In [115]:
```
sns.boxplot(x=df["status"],y=df["year"])
```

Out[115]: `<matplotlib.axes._subplots.AxesSubplot at 0x1dde0494cc0>`



In [116]:
```
sns.boxplot(x=df["status"],y=df["nodes"])
```

Out[116]: `<matplotlib.axes._subplots.AxesSubplot at 0x1dddf391940>`



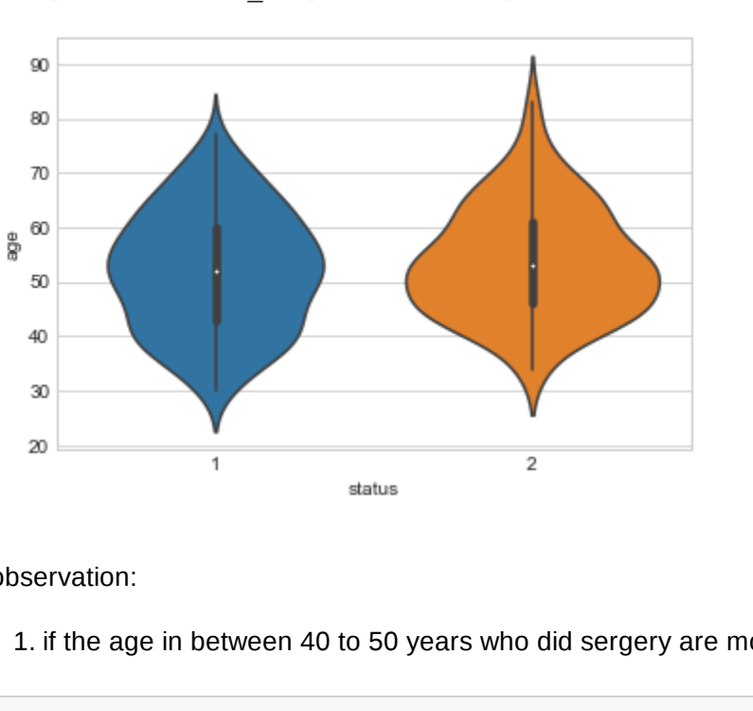
observation:

1.patients who has node <= 40 are more chance to servival at status 1

2.patients who has node > 80 percentile are fall in servival status 2

In [117]:
```
sns.violinplot(x=df["status"],y=df["age"])
```

Out[117]: `<matplotlib.axes._subplots.AxesSubplot at 0x1dddf2cb5f8>`

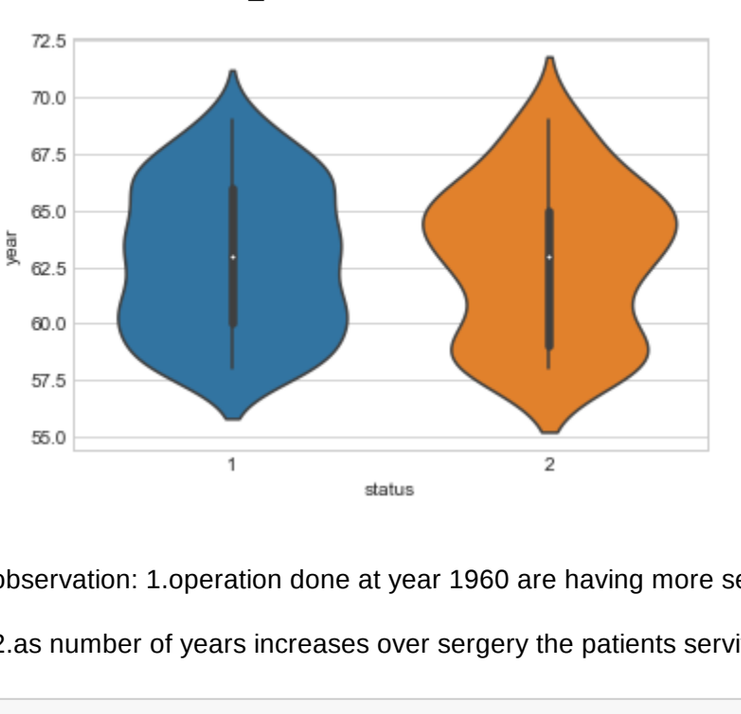


observation:

1. if the age in between 40 to 50 years who did sergery are more likely to be servive more than five years

In [122]:
```
sns.violinplot(x=df["status"],y=df["year"])
```

Out[122]: `<matplotlib.axes._subplots.AxesSubplot at 0x1dde0209b00>`

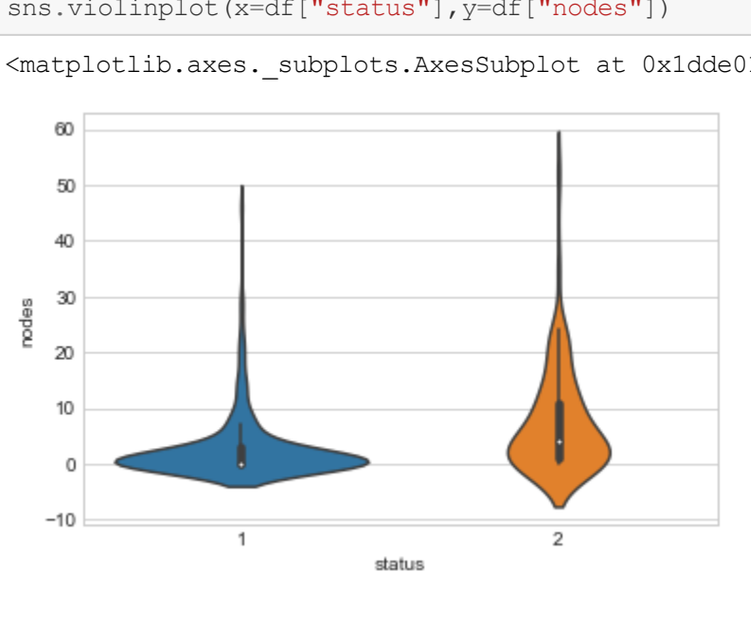


observation: 1.operation done at year 1960 are having more servival rate

2.as number of years increases over sergery the patients servival status being reduced to status 1

In [119]:
```
sns.violinplot(x=df["status"],y=df["nodes"])
```

Out[119]: `<matplotlib.axes._subplots.AxesSubplot at 0x1dde01339b0>`



conclusion