In [2]:

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

In [3]:

```
#the following blog post was used as reference
#https://medium.com/@rndayala/eda-on-haberman-data-set-c9ee6d51ab0a#:~:text=Data%20Set%20
Description%3A%20The%20dataset,undergone%20surgery%20for%20breast%20cancer.
#the procedure to set columns, not include the first row as header and changing 1 to yes
and 2 to no was inspired from this post
```

In [4]:

```
#haberman data set published on kaggle contains cases from a study that was conducted bet
ween 1958 and 1970 at
#the University of Chicago's Billings Hospital on the survival of patients who had underg
one surgery for breast cancer.
```

In [5]:

```
#Feature 1-age of patient
#Feature 2- year of test
#Feature 3-No of axillary nodes present in the body
#OBJECTIVE--TO DETERMINE THE SURVIVAL STATUS OF THE PATIENT
```

In [6]:

```
haberman=pd.read_csv("haberman.csv")
#uploading the dataframe using pandas
```

In [7]:

```
#always a good habit to see the dataset and perform basic analysis on it reated to shape,
columns,features etc
haberman
```

Out[7]:

	30	64	1	1.1
0	30	62	3	1
1	30	65	0	1
2	31	59	2	1
3	31	65	4	1
4	33	58	10	1
...	...	...	...	...
300	75	62	1	1
301	76	67	0	1
302	77	65	3	1
303	78	65	1	2
304	83	58	2	2

305 rows × 4 columns

In [8]:

```
#shape is 305x4 but data points mentioned are 306
```

```
#the columnns are also a data point, by default the first row in csv is
#taken as columns, to consider it as a data point as well
haberman=pd.read_csv("haberman.csv",header=None)
haberman
```

Out[8]:

	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
...	...	...	...	...
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 [9]:

```
haberman.shape
```

Out[9]:

```
(306, 4)
```

In [10]:

```
#now we have the correct number of data points in the dataframe, now check the columns
haberman.columns
```

Out[10]:

```
Int64Index([0, 1, 2, 3], dtype='int64')
```

In [11]:

```
#we have to define the column names by ourselves as well
column_names=["age","year","axillary_nodes","survival_status"]
haberman.columns=column_names
haberman
```

Out[11]:

	age	year	axillary_nodes	survival_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
...	...	...	...	...
301	75	62	1	1
302	76	67	0	1
303	77	65	3	1
304	78	65	1	2

	age	year	axillary_nodes	survival_status
305	83	58	2	2

306 rows × 4 columns

In [12]:

```
#survival status=1->survived
#survival status=2->not survived
#we can change them to make more sense
haberman.survival_status=haberman.survival_status.map({1:"yes",2:"no"})
haberman
```

Out[12]:

	age	year	axillary_nodes	survival_status
0	30	64	1	yes
1	30	62	3	yes
2	30	65	0	yes
3	31	59	2	yes
4	31	65	4	yes
...	...	...	...	...
301	75	62	1	yes
302	76	67	0	yes
303	77	65	3	yes
304	78	65	1	no
305	83	58	2	no

306 rows × 4 columns

In [13]:

```
#the dataframe is now ready to be used for Exploratory Data Analysis
#now to check how many data points for each class are present
#i.e to observe how many patients who have survived and how many have not
g=haberman.groupby("survival_status").size()
g
```

Out[13]:

```
survival_status
no       81
yes     225
dtype: int64
```

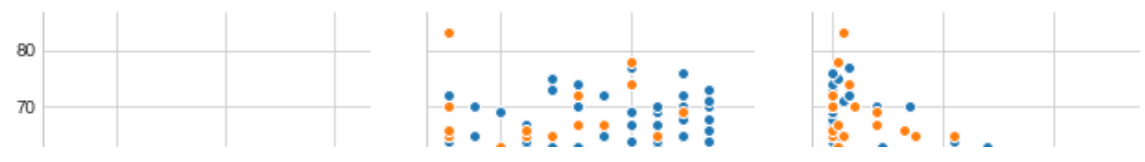
In [14]:

```
#Type of dataset-UNBALANCED DATASET
#set which is a common observation in datasets related to medical records
```

PAIR PLOTS-TO PLOT BETWEEN ALL POSSIBLE PAIRS OF FEATURES TOTAL UNIQUE PLOTS=nC2 (n=number of features)=3C2=3

In [15]:

```
sns.set_style("whitegrid")
sns.pairplot(haberman,hue="survival_status",size=3)
plt.show()
```



OBSERVATIONS FROM PAIR PLOTS-

1) age vs year of operation-overlapping and distinction can't be made

2) year of operation vs no of axillary nodes-overlapping and distinction can't be made

3) age vs no of axillary nodes- overlapping but better than the above mentioned 2 cases. Therefore we will select these 2 features for univariate analysis.

NOTE-WE DONT CONSIDER GRAPHS BELOW THE MAIN DIAGONAL AS THEY ARE JUST THE SAME 3 GRAPHS WITH AXES INTERCHANGED

3D SCATTER PLOT

In [16]:

```
import plotly.express as px
fig=px.scatter_3d(data_frame=haberman,x="age",y="year",z="axillary_nodes",color="survival_status")
fig.show()
```

OBSERVATION FROM 3D SCATTER PLOT-

NO CONCRETE DISTINCTION CAN BE MADE FROM THE 3D SCATTER PLOT OF ALL 3 FEATURES TAKEN TOGETHER

UNIVARIATE ANAYLYSIS

1) FOR AGE OF PATIENT

In [17]:

```
#HISTROGRAM
y=haberman.survival_status=="yes"
n=haberman.survival_status=="no"
```

In [18]:

```
haberman[y]
```

Out[18]:

	age	year	axillary_nodes	survival_status
0	30	64	1	yes
1	30	62	3	yes
2	30	65	0	yes
3	31	59	2	yes
4	31	65	4	yes
...	...	...	...	...
298	73	68	0	yes
300	74	63	0	yes
301	75	62	1	yes
302	76	67	0	yes
303	77	65	3	yes

225 rows × 4 columns

In [19]:

```
haberman[n]
```

Out[19]:

	age	year	axillary_nodes	survival_status
7	34	59	0	no

	age	year	axillary_nodes	survival_status
8	34	66	9	no
24	38	69	21	no
34	39	66	0	no
43	41	60	23	no
...	...	...	...	...
286	70	58	4	no
293	72	63	0	no
299	74	65	3	no
304	78	65	1	no
305	83	58	2	no

81 rows × 4 columns

In [20]:

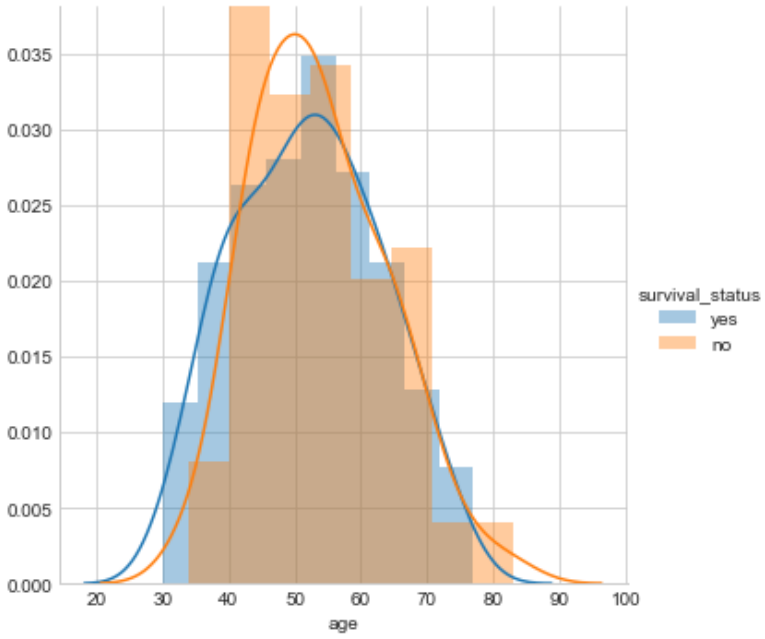
```
sns.FacetGrid(haberman,hue="survival_status",size=5)\
    .map(sns.distplot,"age")\
    .add_legend();
plt.show();
```

```
C:\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

C:\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
```



we can see that the PDF of survival status on the basis of age is highly overlapping and that between ages 40 and 65 the area under both the graphs is almost same-

i.e between ages 40 and 65 the percentage of patients with survival status as yes and the percentage of patients with survival status no is almost the same

2) FOR YEAR OF OPERATION

In [21]:

```
sns.FacetGrid(haberman,hue="survival_status",size=5)\
    .map(sns.distplot,"year")\
```

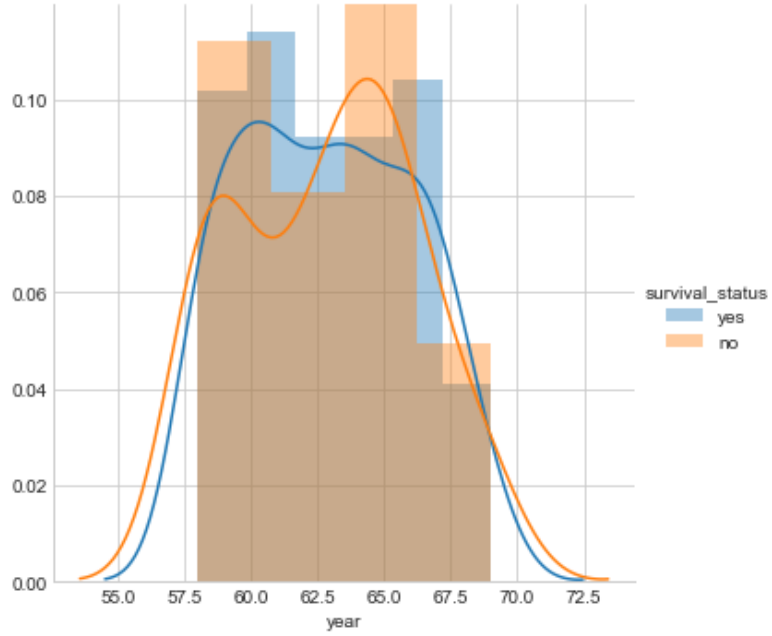
```
    .add_legend();
plt.show();
```

```
C:\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

C:\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
```



we can see that the pdf for survival status on the basis of year of operation is highly overlapping and the are under both the graphs is roughly the same between 1958 and 1968-

i.e- the percentage of patients with survival status as yes and the percentage of patients with survival status as no is the same for the years between 1958 and 1968

3) FOR NUMBER OF AXILLARY NODES

In [22]:

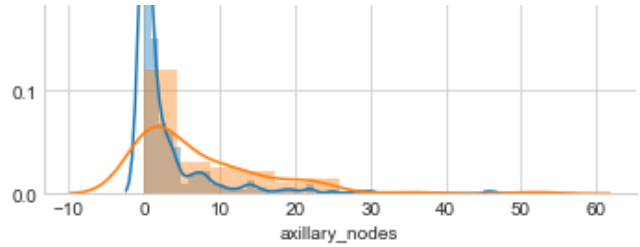
```
sns.FacetGrid(haberman,hue="survival_status",size=5)\
    .map(sns.distplot,"axillary_nodes")\
    .add_legend();
plt.show();
```

```
C:\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

C:\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
```



the pdf of for the survival status on the basis of the number of axillary nodes has the least overlapping for the 2 survival status

thus we can make use of the following if else conditions to get a fairly accurate result-->

if no of axillary nodes>=0 and <=3-->survival status=high

no of axillary nodes>=3-->survuval status=low

DETAILED UNIVARIATE ANALYSIS ON THE BASIS OF NUMBER OF AXILLARY NODES

In [23]:

```
haberman[y]
```

Out[23]:

	age	year	axillary_nodes	survival_status
0	30	64	1	yes
1	30	62	3	yes
2	30	65	0	yes
3	31	59	2	yes
4	31	65	4	yes
...	...	...	...	...
298	73	68	0	yes
300	74	63	0	yes
301	75	62	1	yes
302	76	67	0	yes
303	77	65	3	yes

225 rows × 4 columns

In [24]:

```
#analysis of all the patients withh survival status as yes on the basis of number of axil
lary nodes
print(haberman[y].axillary_nodes.min())
print(haberman[y].axillary_nodes.max())
```

```
0
46
```

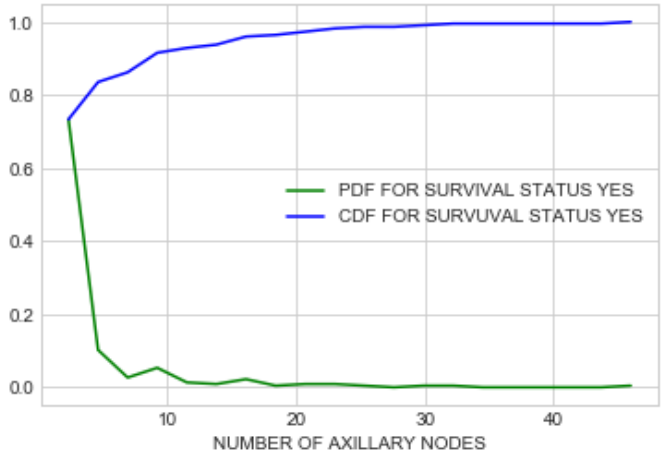
In [25]:

```
count,bin_edges=np.histogram(haberman[y].axillary_nodes,bins=20,density=True)
print(count)
print(bin_edges)
```

```
[0.31884058 0.04444444 0.0115942  0.02318841 0.0057971  0.00386473
 0.00966184 0.00193237 0.00386473 0.00386473 0.00193237 0.
 0.00193237 0.00193237 0.         0.         0.         0.
 0.         0.00193237]
[ 0.   2.3  4.6  6.9  9.2 11.5 13.8 16.1 18.4 20.7 23.  25.3 27.6 29.9
 32.2 34.5 36.8 39.1 41.4 43.7 46. ]
```

In [26]:

```
pdf=count/sum(count)
cdf=np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,'g',label="PDF FOR SURVIVAL STATUS YES")
plt.plot(bin_edges[1:],cdf,'b',label="CDF FOR SURVUVAL STATUS YES")
plt.xlabel("NUMBER OF AXILLARY NODES")
plt.legend()
plt.show()
```



the blue line shows the CDF for all the patients who survived on the basis of the number of axillary nodes present in their body

from the blue line we can clearly see that roughly 85% of the total patients who survived had number of axillary nodes less than 5-->

if axillary nodes>=0 and <=5--> survival status=high--> this statement has 85 percent chance of being correct and 15 percent chance of being incorrect

we can also see from the cdf that when the number of axillary cases reach 10 the chances of survival become very low as for the total patients who survived only 10% had more than 10 nodes.

In [27]:

```
haberman[n]
```

Out[27]:

	age	year	axillary_nodes	survival_status
7	34	59	0	no
8	34	66	9	no
24	38	69	21	no
34	39	66	0	no
43	41	60	23	no
...	...	...	...	...
286	70	58	4	no
293	72	63	0	no
299	74	65	3	no
304	78	65	1	no
305	83	58	2	no

81 rows × 4 columns

In [28]:

```
#analysis of all the patients withh survival status as no on the basis of number of axill
```

```
#analysis of all the patients with survival status as no on the basis of number of axill
ary nodes
print(haberman[n].axillary_nodes.min())
print(haberman[n].axillary_nodes.max())
```

```
0
52
```

In [29]:

```
count,bin_edges=np.histogram(haberman[n].axillary_nodes,bins=20,density=True)
print(count)
print(bin_edges)
```

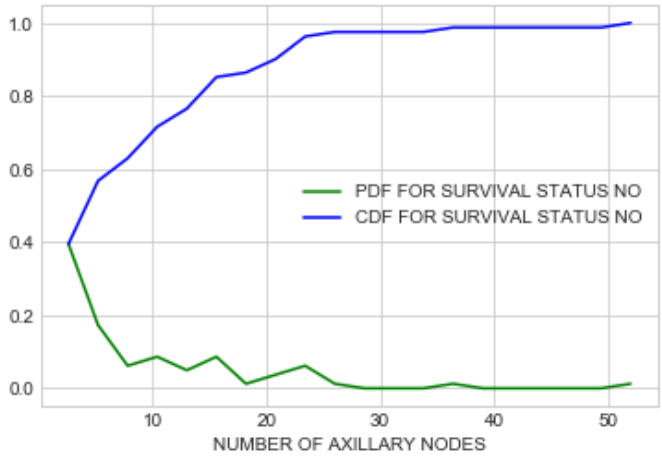
```
[0.15194682 0.06647673 0.02374169 0.03323837 0.01899335 0.03323837
 0.00474834 0.01424501 0.02374169 0.00474834 0.         0.
 0.         0.00474834 0.         0.         0.         0.
 0.         0.00474834]
[ 0.   2.6  5.2  7.8 10.4 13.  15.6 18.2 20.8 23.4 26.  28.6 31.2 33.8
 36.4 39.  41.6 44.2 46.8 49.4 52. ]
```

In [30]:

```
pdf=count/sum(count)
cdf=np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,'g',label="PDF FOR SURVIVAL STATUS NO")
plt.plot(bin_edges[1:],cdf,'b',label="CDF FOR SURVIVAL STATUS NO")
plt.xlabel("NUMBER OF AXILLARY NODES")
plt.legend()
plt.plot()
```

Out[30]:

```
[]
```



we can see from the cdf that for the patients who did not survive more than 60% had number of axillary nodes>=3

if axillary nodes>=3-->survival chance=low-->this statement has 60% percent accuracy

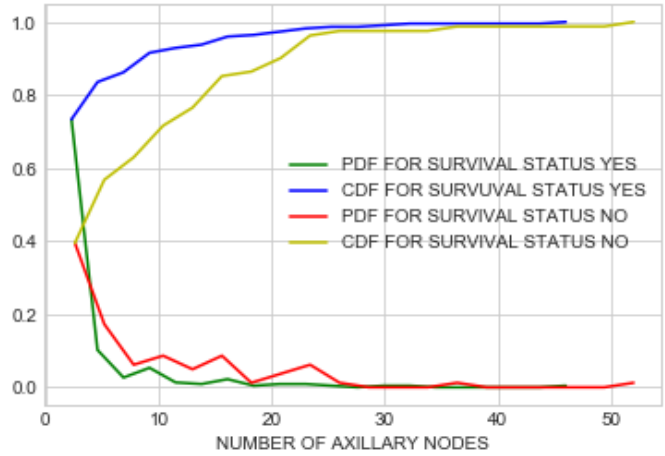
In [31]:

```
#combine the above pdf and cdf for both yes and no on the basis of number of axillary nod
es
count,bin_edges=np.histogram(haberman[y].axillary_nodes,bins=20,density=True)
pdf=count/sum(count)
cdf=np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,'g',label="PDF FOR SURVIVAL STATUS YES")
plt.plot(bin_edges[1:],cdf,'b',label="CDF FOR SURVUVAL STATUS YES")
count,bin_edges=np.histogram(haberman[n].axillary_nodes,bins=20,density=True)
pdf=count/sum(count)
cdf=np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,'r',label="PDF FOR SURVIVAL STATUS NO")
plt.plot(bin_edges[1:],cdf,'y',label="CDF FOR SURVIVAL STATUS NO")
plt.xlabel("NUMBER OF AXILLARY NODES")
plt.legend()
```

```
plt.plot()
```

Out[31]:

```
[]
```



FINAL OBSERVATIONS FOR DISTINCTIONS-

WE CAN USE THE FOLLOWING IF ELSE CONDTIONS FOR PREDICTING SURVIVAL STATUS ON THE BASIS OF NUMBER OF AXILLARY NODES PRESENT IN THE BODY WHICH WAS THE BEST POSSIBLE FEATURE IN THIS SCENARIO-

if number of axillary nodes>=0 && number of axillary nodes<=3-->survival chance=high-->this statement has about 75% accuracy as out of the total patients who survived 75% had less than 3 axillary nodes and 25% had more than 3 axillary nodes

if number of axillary nodes>=3-->survival chance=low-->this statement has about 60% accuracy as out of the total patients who did not survuve 60% had more than 3 axillary nodes and 40% had less than 3 axillary nodes

MORE IN DEPTH UNIVARIATE ANALYSIS FOR NUMBER OF AXILLARY NODES

mean,std,median,iqr and 90th percentile for both survuvl status yes and no for axillary nodes box plot,violin plot,contour plot

MEAN, MEDIAN, STANDARD DEVIATION FOR NO AXILLARY NODES

In [32]:

```
print("for all the patients who survived: \n")
#to check the central value
print("mean number of axillary nodes: {} \n".format(np.mean(haberman[y].axillary_nodes))
)
#to check the spread about the central value
print("standard deviation from the mean number of axillary nodes: {} \n".format(np.std(h
aberman[y].axillary_nodes)))
#to check the central value which is least affected by outliers
print("the middle value or the median number of axillary nodes: {} \n".format(np.median(
haberman[y].axillary_nodes)))
```

```
for all the patients who survived:

mean number of axillary nodes: 2.7911111111111113

standard deviation from the mean number of axillary nodes: 5.857258449412131

the middle value or the median number of axillary nodes: 0.0
```

In [33]:

```
print("for all the patients who did not survive: \n")
#to check the central value
```

```
print("mean number of axillary nodes: {} \n".format(np.mean(haberman[n].axillary_nodes))
)
#to check the spread about the central value
print("standard deviation from the mean number of axillary nodes: {} \n".format(np.std(h
aberman[n].axillary_nodes)))
#to check the central value which is least affected by outliers
print("the middle value or the median number of axillary nodes: {} \n".format(np.median(
haberman[y].axillary_nodes)))
```

```
for all the patients who did not survive:

mean number of axillary nodes: 7.45679012345679

standard deviation from the mean number of axillary nodes: 9.128776076761632

the middle value or the median number of axillary nodes: 0.0
```

PERCENTILE, QUANTILES AND INTER QUANTILE RANGE FOR THE NUMBER OF AXILLARY NODES

In [34]:

```
print("for all the patients who survived: \n")
print("number of axillary nodes at 75th percentile: {} \n".format(np.percentile(haberman
[y].axillary_nodes,75)))
print("the quantiles are nothing but the values at 0th,25th,50th,75th percentile")
print("the quantiles for the number of axillary nodes:")
print(np.percentile(haberman[y].axillary_nodes,np.arange(0,100,25)))
```

```
for all the patients who survived:

number of axillary nodes at 75th percentile: 3.0

the quantiles are nothing but the values at 0th,25th,50th,75th percentile
the quantiles for the number of axillary nodes:
[0. 0. 0. 3.]
```

In [35]:

```
print("the inter quantile range is the difference of the value at the 75th and the 25th p
ercentile")
print("the inter quantile range for number  of axillary nodes in patients who survived is
:")
x=np.percentile(haberman[y].axillary_nodes,75)
y=np.percentile(haberman[y].axillary_nodes,25)
print(x-y)
```

```
the inter quantile range is the difference of the value at the 75th and the 25th percenti
le
the inter quantile range for number  of axillary nodes in patients who survived is:
3.0
```

In [36]:

```
print("for all the patients who did not survive: \n")
print("number of axillary nodes at 75th percentile: {} \n".format(np.percentile(haberman
[n].axillary_nodes,75)))
print("the quantiles are nothing but the values at 0th,25th,50th,75th percentile")
print("the quantiles for the number of axillary nodes:")
print(np.percentile(haberman[n].axillary_nodes,np.arange(0,100,25)))
```

```
for all the patients who did not survive:

number of axillary nodes at 75th percentile: 11.0

the quantiles are nothing but the values at 0th,25th,50th,75th percentile
the quantiles for the number of axillary nodes:
[ 0.  1.  4. 11.]
```

In [37]:

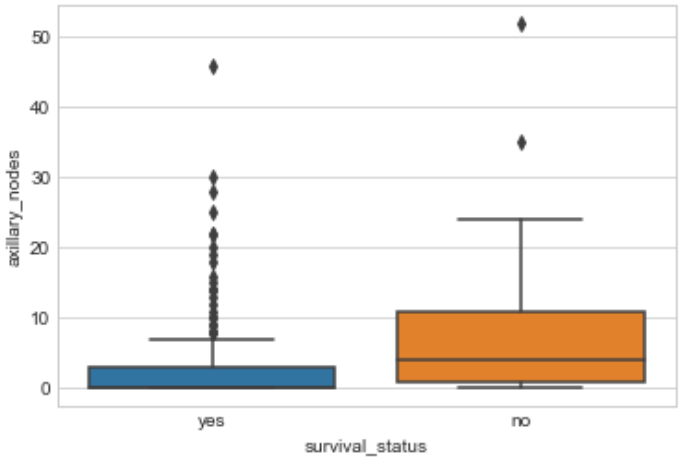
```
print("the inter quantile range is the difference of the value at the 75th and the 25th p
ercentile")
print("the inter quantile range for number  of axillary nodes in patients who did not sur
vive is:")
x=np.percentile(haberman[n].axillary_nodes,75)
y=np.percentile(haberman[n].axillary_nodes,25)
print(x-y)
```

```
the inter quantile range is the difference of the value at the 75th and the 25th percenti
le
the inter quantile range for number  of axillary nodes in patients who did not survive is
:
10.0
```

BOX PLOT WITH WHISKERS

In [38]:

```
#box plots are specifically designed to analyse the quantiles
sns.boxplot(x="survival_status",y="axillary_nodes",data=haberman)
plt.show()
```



INSIDE THE BOX-

LOWER LINE-25TH PERCENTILE

UPPER LINE-75HTH PERCENTILE

MIDDLE LINE-50TH PERCENTILE/MEDIAN

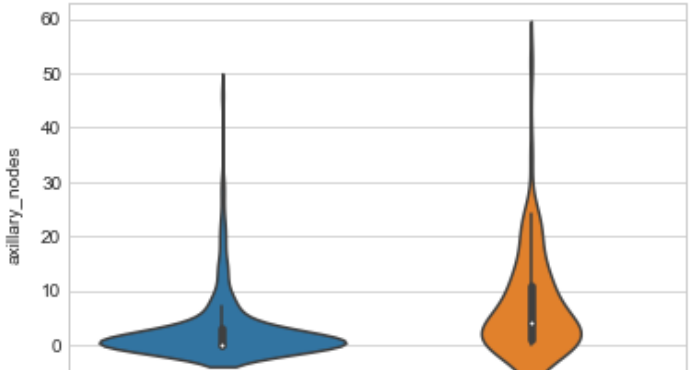
HEIGHT OF THE BOX- INER QUANTILE RANGE= value at 75th percentile- value at 25th percentile

LENGTH OF WHISKER= IQRx 1.5

VIOLIN PLOTS FOR NUMBER OF AXILLARY NODES

In [40]:

```
#best of both histograms and box plots
sns.violinplot(x="survival_status",y="axillary_nodes",data=haberman)
plt.show()
```



The violin plots can work both as box plots and pdf

the box plot inside the violin is the box plot for the class

denser regions with more points are fat and the sparse regions with less points are thin- if we look at the violin plot from sideways it works as the PDF for the class

FINAL OBSERVATIONS FOR DISTINCTIONS-

WE CAN USE THE FOLLOWING IF ELSE CONDTIONS FOR PREDICTING SURVIVAL STATUS ON THE BASIS OF NUMBER OF AXILLARY NODES PRESENT IN THE BODY WHICH WAS THE BEST POSSIBLE FEATURE IN THIS SCENARIO-

if number of axillary nodes>=0 && number of axillary nodes<=3-->survival chance=high-->this statement has about 75% accuracy as out of the total patients who survived 75% had less than 3 axillary nodes and 25% had more than 3 axillary nodes

if number of axillary nodes>=3-->survival chance=low-->this statement has about 60% accuracy as out of the total patients who did not survuve 60% had more than 3 axillary nodes and 40% had less than 3 axillary nodes