In [2]:

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

In [3]:

```
data=pd.read_csv("haberman.csv")
```

In [4]:

```
data.head()
```

Out[4]:

	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 [5]:

```
data.shape
```

Out[5]:

```
(306, 4)
```

In [6]:

```
data.columns
```

Out[6]:

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

In [7]:

```
data['nodes'].describe()
```

Out[7]:

```
count    306.000000
mean       4.026144
std        7.189654
min        0.000000
25%        0.000000
50%        1.000000
75%        4.000000
max       52.000000
Name: nodes, dtype: float64
```

In [9]:

```
data['age'].describe()
```

Out[9]:

```
count    306.000000
```

```
mean      52.457516
std       10.803452
min       30.000000
25%       44.000000
50%       52.000000
75%       60.750000
max       83.000000
Name: age, dtype: float64
```

In [10]:

```
data['status'].describe()
```

Out[10]:

```
count    306.000000
mean       1.264706
std        0.441899
min        1.000000
25%        1.000000
50%        1.000000
75%        2.000000
max        2.000000
Name: status, dtype: float64
```

In [11]:

```
data['year'].describe()
```

Out[11]:

```
count    306.000000
mean      62.852941
std        3.249405
min       58.000000
25%       60.000000
50%       63.000000
75%       65.750000
max       69.000000
Name: year, dtype: float64
```

In [12]:

```
print(data.info())
```

```
<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: 9.7 KB
None
```

In [14]:

```
data['status'].unique()
```

Out[14]:

```
array([1, 2], dtype=int64)
```

In [4]:

```
data1=data
```

In [5]:

```
data1.head()
```

Out[5]:

	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 [6]:

```
data['status']=data['status'].replace([1,2],["yes","no"])
```

In [7]:

```
data.head()
```

Out[7]:

	age	year	nodes	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

In [32]:

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

Out[32]:

```
yes    225
no      81
Name: status, dtype: int64
```

OBSERVATIONS

There are no missing values in the data The status column has 1 or 2 which is mapped to 'yes' or 'no' No. of features=3 (age, year and nodes) No. of points= No. of observations= 306 No. of classes= 2 = yes or no No. of data points in 'yes' class= 225 No. of data points in 'no' class = 81

In [28]:

```
data.describe()
```

Out[28]:

	age	year	nodes
count	306.000000	306.000000	306.000000
mean	52.457516	62.852941	4.026144
std	10.803452	3.249405	7.189654
min	30.000000	58.000000	0.000000
25%	44.000000	60.000000	0.000000
50%	52.000000	63.000000	1.000000

	age	year	nodes
75%	60.750000	65.750000	4.000000
max	83.000000	69.000000	52.000000

OBSERVATIONS

The median age group is 52 years 75% of people have 4 or less than 4 positive lymph nodes. Out of which 25% of the people have 0 positive lymph nodes. The data set is imbalanced with 73% values being a yes under the status column

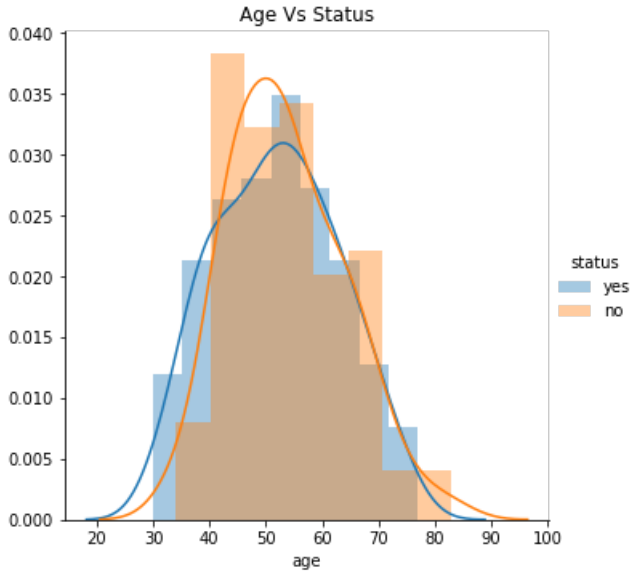
OBJECTIVE

To predict whether the patient will survive after 5 years "yes" or not "no" based on the age, year of treatment and no. of positive lymph nodes

UNIVARIATE ANALYSIS

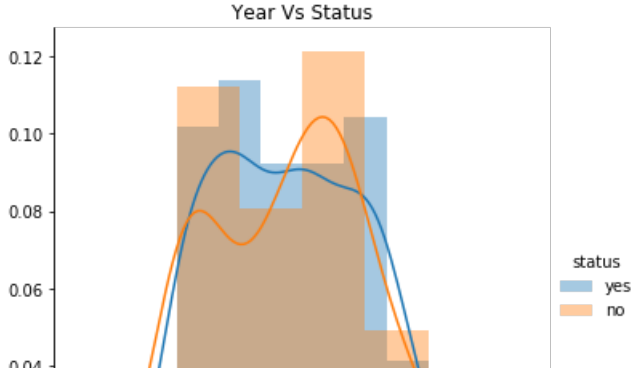
In [13]:

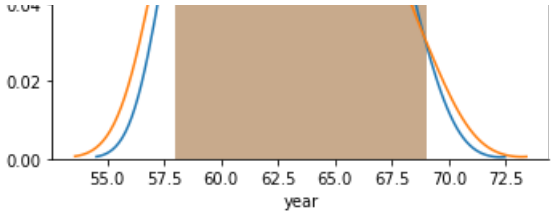
```
fig=sns.FacetGrid(data,hue='status',size=5)
fig.map(sns.distplot,"age").add_legend()
plt.title('Age Vs Status')
plt.show()
```



In [14]:

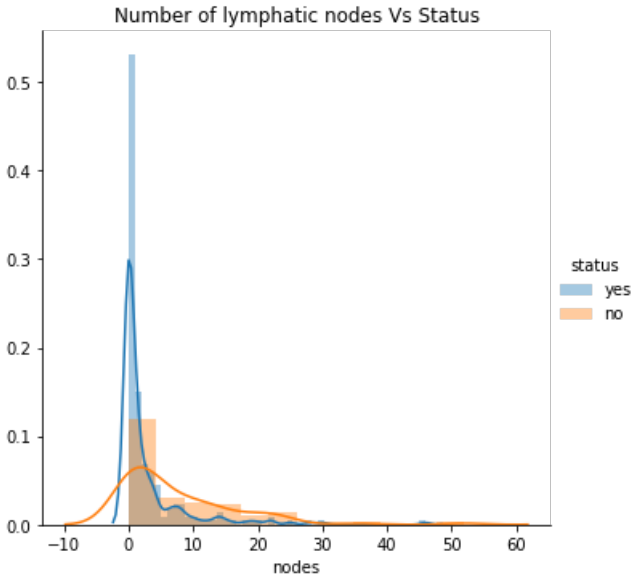
```
fig=sns.FacetGrid(data,hue='status',size=5)
fig.map(sns.distplot,"year").add_legend()
plt.title('Year Vs Status')
plt.show()
```





In [15]:

```
fig=sns.FacetGrid(data,hue='status',size=5)
fig.map(sns.distplot,"nodes").add_legend()
plt.title('Number of lymphatic nodes Vs Status')
plt.show()
```

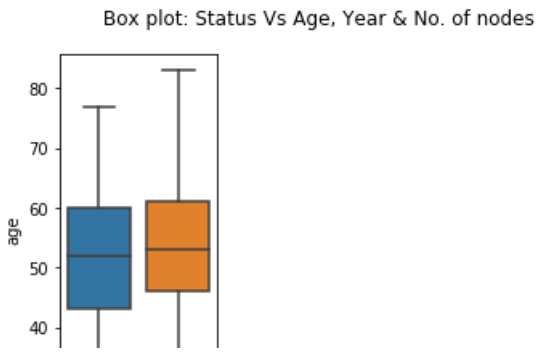


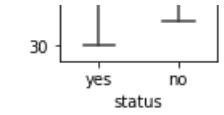
OBSERVATIONS

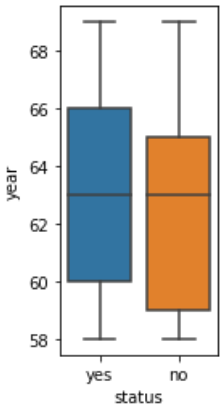
We can clearly state that the two classes cannot be separated using either of the features mentioned such as age, year and nodes

In [19]:

```
fig=plt.figure()
fig.suptitle("Box plot: Status Vs Age, Year & No. of nodes")
plt.subplot(131)
sns.boxplot(x='status',y='age',data=data)
plt.show()

plt.subplot(132)
sns.boxplot(x='status',y='year',data=data)
plt.show()

plt.subplot(133)
sns.boxplot(x='status',y='nodes',data=data)
plt.show
```

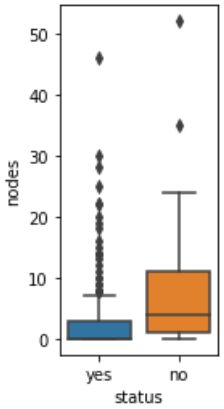






Out[19]:

```
<function matplotlib.pyplot.show(*args, **kw)>
```

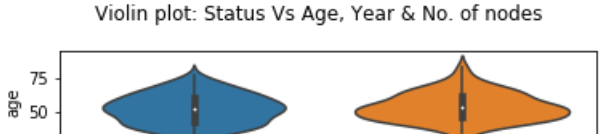


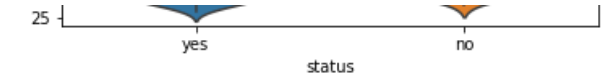
OBSERVATIONS

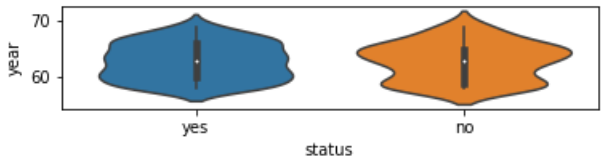
Age of the 75% patients who survived beyond 5 years of treatment is less than or equal to 45 years. Year of treatment for 75% patients who survived beyond 5 years of treatment is less than or equal to 1966. No. of lymph nodes for 75% patients who survived beyond 5 years of treatment is less than or equal to 5

In [20]:

```
fig=plt.figure()
fig.suptitle("Violin plot: Status Vs Age, Year & No. of nodes")


plt.subplot(311)
sns.violinplot(x='status',y='age',data=data)
plt.show()

plt.subplot(312)
sns.violinplot(x='status',y='year',data=data)
plt.show()

plt.subplot(313)
sns.violinplot(x='status',y='nodes',data=data)
plt.show
```

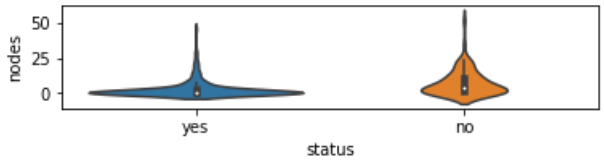






Out[20]:

```
<function matplotlib.pyplot.show(*args, **kw)>
```



OBSERVATIONS

The no. of lymph nodes of the survivors is dense around 0 to 5 The patients treated after 1966 have the slighlty higher chance to surive that the rest. The patients treated before 1959 have the slighlty lower chance to surive that the rest.

In [21]:

```
data_yes=data.loc[data['status']=='yes']
data_no=data.loc[data['status']=='no']
data_yes.head()
```

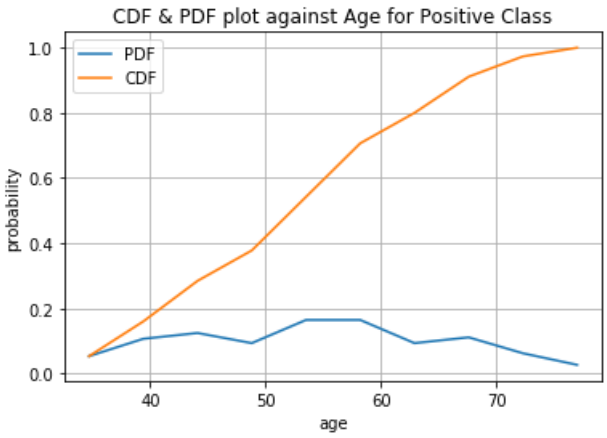
Out[21]:

	age	year	nodes	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

In [34]:

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

cdf=np.cumsum(pdf)
print(cdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)
plt.grid()
plt.xlabel('age')
plt.ylabel('probability')
plt.title('CDF & PDF plot against Age for Positive Class')
plt.legend(['PDF','CDF'])
plt.show()
```

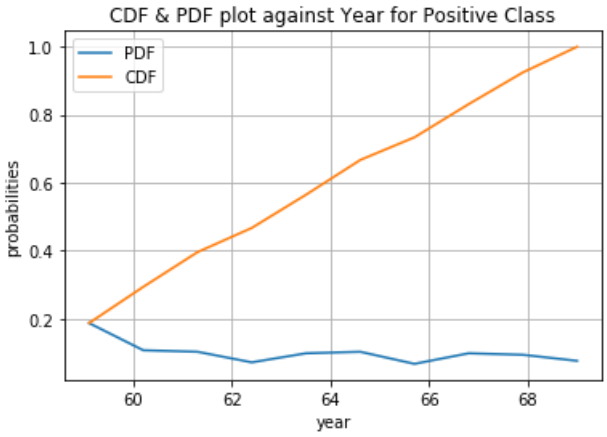
```
[0.05333333 0.10666667 0.12444444 0.09333333 0.16444444 0.16444444
 0.09333333 0.11111111 0.06222222 0.02666667]
[30.  34.7 39.4 44.1 48.8 53.5 58.2 62.9 67.6 72.3 77. ]
[0.05333333 0.16       0.28444444 0.37777778 0.54222222 0.70666667
 0.8        0.91111111 0.97333333 1.        ]
```



In [32]:

```
counts,bin_edges=np.histogram(data_yes['year'],bins=10,density=True)
pdf=counts/sum(counts)
cdf=np.cumsum(pdf)
print(pdf)
print(cdf)
print(bin_edges)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)
plt.grid()
plt.xlabel('year')
plt.ylabel('probabilities')
plt.title('CDF & PDF plot against Year for Positive Class')
plt.legend(['PDF','CDF'])
plt.show()
```

```
[0.18666667 0.10666667 0.10222222 0.07111111 0.09777778 0.10222222
 0.06666667 0.09777778 0.09333333 0.07555556]
[0.18666667 0.29333333 0.39555556 0.46666667 0.56444444 0.66666667
 0.73333333 0.83111111 0.92444444 1.        ]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
```

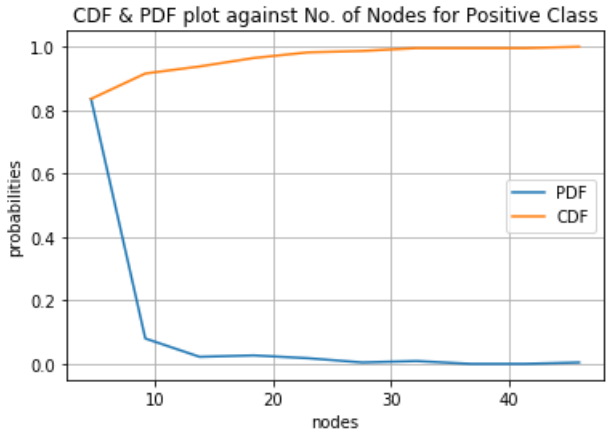


In [33]:

```
counts,bin_edges=np.histogram(data_yes['nodes'],bins=10,density=True)
pdf=counts/sum(counts)
cdf=np.cumsum(pdf)
print(pdf)
print(cdf)
print(bin_edges)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)
plt.grid()
plt.xlabel('nodes')
plt.ylabel('probabilities')
plt.title('CDF & PDF plot against No. of Nodes for Positive Class')
plt.legend(['PDF','CDF'])
```

```
plt.show()
```

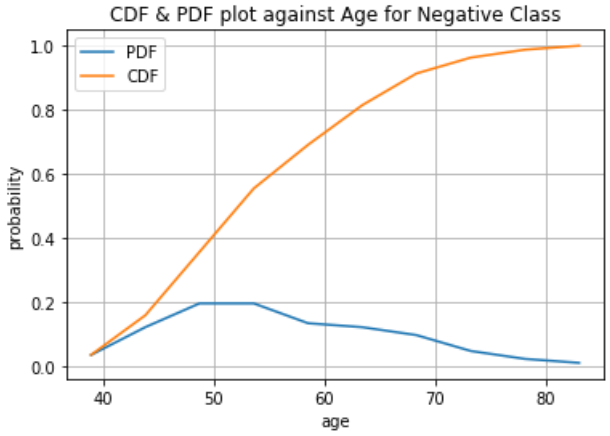
```
[0.83555556 0.08       0.02222222 0.02666667 0.01777778 0.00444444
 0.00888889 0.         0.         0.00444444]
[0.83555556 0.91555556 0.93777778 0.96444444 0.98222222 0.98666667
 0.99555556 0.99555556 0.99555556 1.        ]
[ 0.   4.6  9.2 13.8 18.4 23.  27.6 32.2 36.8 41.4 46. ]
```



In [29]:

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

cdf=np.cumsum(pdf)
print(cdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)
plt.grid()
plt.xlabel('age')
plt.ylabel('probability')
plt.title('CDF & PDF plot against Age for Negative Class')
plt.legend(['PDF','CDF'])
plt.show()
```

```
[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. ]
[0.03703704 0.16049383 0.35802469 0.55555556 0.69135802 0.81481481
 0.91358025 0.96296296 0.98765432 1.        ]
```

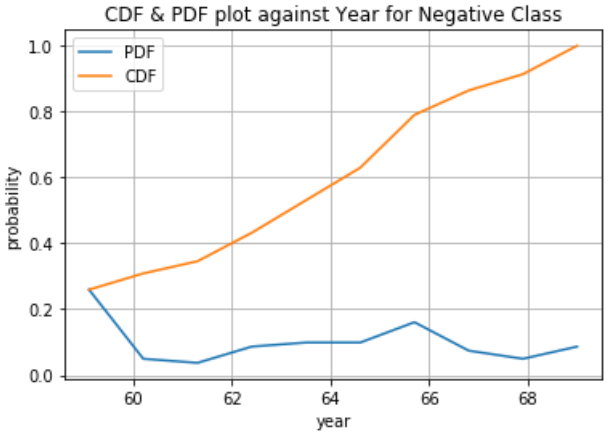


In [30]:

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

```
cdf=np.cumsum(pdf)
print(cdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)
plt.grid()
plt.xlabel('year')
plt.ylabel('probability')
plt.title('CDF & PDF plot against Year for Negative Class')
plt.legend(['PDF','CDF'])
plt.show()
```

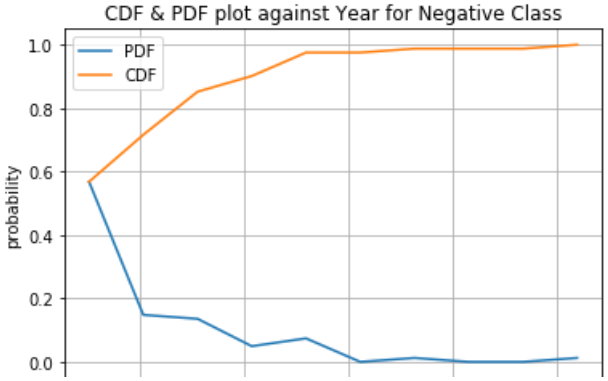
```
[0.25925926 0.04938272 0.03703704 0.08641975 0.09876543 0.09876543
 0.16049383 0.07407407 0.04938272 0.08641975]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
[0.25925926 0.30864198 0.34567901 0.43209877 0.5308642  0.62962963
 0.79012346 0.86419753 0.91358025 1.        ]
```

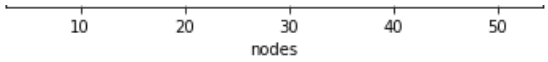


In [31]:

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

cdf=np.cumsum(pdf)
print(cdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)
plt.grid()
plt.xlabel('nodes')
plt.ylabel('probability')
plt.title('CDF & PDF plot against Year for Negative Class')
plt.legend(['PDF','CDF'])
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. ]
[0.56790123 0.71604938 0.85185185 0.90123457 0.97530864 0.97530864
 0.98765432 0.98765432 0.98765432 1.        ]
```





BIVARIATE ANALYSIS

In [37]:

```
plt.close()
sns.set_style("whitegrid")
sns.pairplot(data, hue='status')
plt.suptitle('Pair Plot for positive and negative class against Year, Age and Nodes')
plt.show()
```

OBSERVATIONS

We can clearly see that year of treatment and number of positive lymphatic nodes can be clearly distinguished between the two classes (yes or no)

In [ ]:

```

```