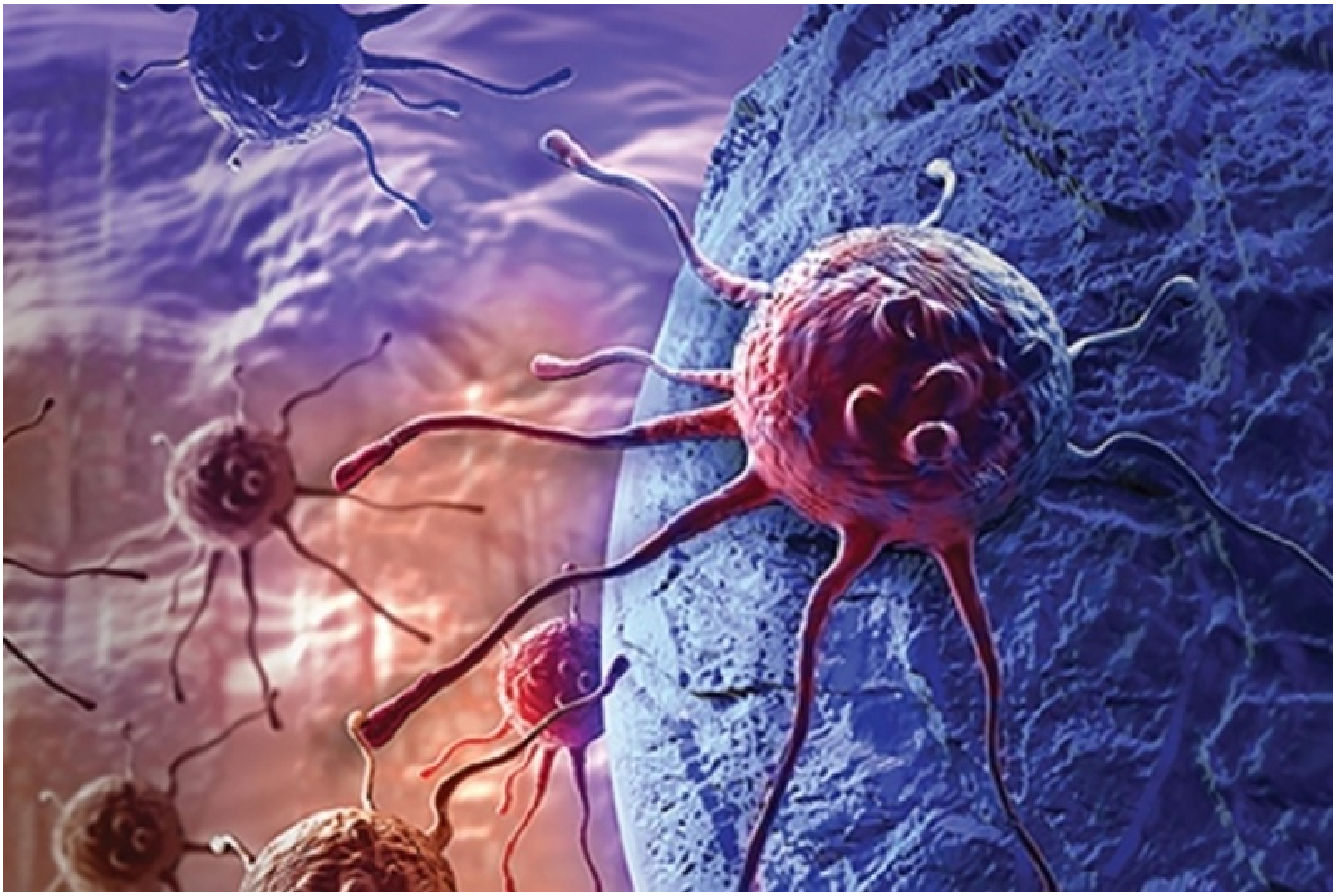
Haberman Cancer Survival



In [1]:

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

In [2]:

```
df= pd.read_csv("C:/Users/anirb/Downloads/archive (6).zip")
df
```

Out[2]:

	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
...	...	...	...	...
300	74	63	0	1
301	75	62	1	1
302	76	67	0	1
303	77	65	3	1
			1	2

Loading [MathJax]/extensions/Safe.js

305 rows × 4 columns

In [3]:

```
df.shape
```

Out[3]:

```
(305, 4)
```

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 305 entries, 0 to 304
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   age     305 non-null    int64
 1   year    305 non-null    int64
 2   nodes   305 non-null    int64
 3   status  305 non-null    int64
dtypes: int64(4)
memory usage: 9.7 KB
```

In [5]:

```
df.columns
```

Out[5]:

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

In [6]:

```
df.describe()
```

Out[6]:

	age	year	nodes	status
count	305.000000	305.000000	305.000000	305.000000
mean	52.357377	62.868852	4.032787	1.262295
std	10.678010	3.242783	7.200528	0.440605
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%	60.000000	66.000000	4.000000	2.000000
max	78.000000	69.000000	52.000000	2.000000

In [7]:

```
df['status']= df['status'].map({1:'Yes',2: 'No'})
df.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 [8]:

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

Out[8]:

```
Yes    225
No      80
Name: status, dtype: int64
```

In [9]:

```
status_yes= df[df['status']== 'Yes']
status_yes.describe()
```

Out[9]:

	age	year	nodes
count	225.000000	225.000000	225.000000
mean	52.017778	62.862222	2.791111
std	11.012154	3.222915	5.870318
min	30.000000	58.000000	0.000000
25%	43.000000	60.000000	0.000000
50%	52.000000	63.000000	0.000000
75%	60.000000	66.000000	3.000000
max	77.000000	69.000000	46.000000

In [10]:

```
status_no= df[df['status']== 'No']
status_no.describe()
```

Out[10]:

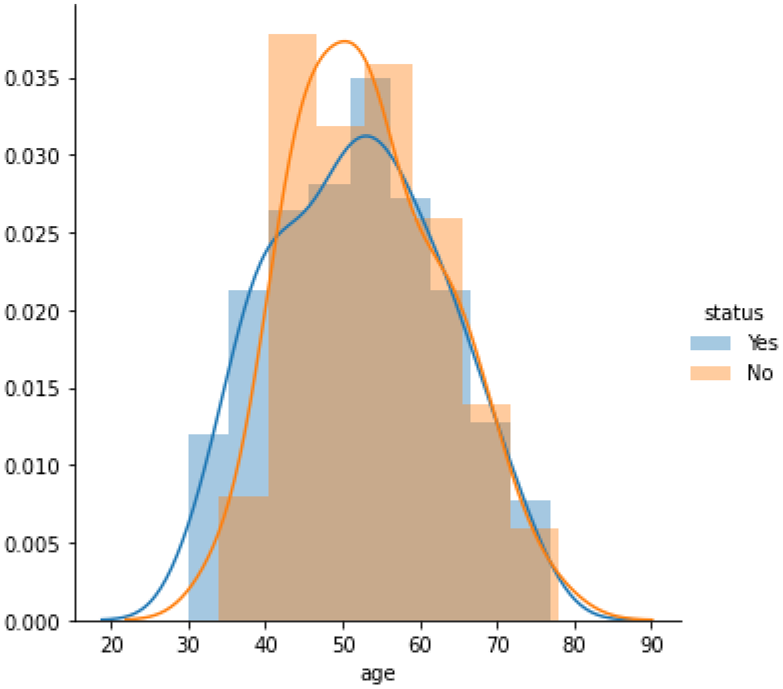
	age	year	nodes
count	80.000000	80.000000	80.000000
mean	53.312500	62.887500	7.525000
std	9.677841	3.318509	9.222942
min	34.000000	58.000000	0.000000
25%	46.000000	59.750000	1.000000
50%	53.000000	63.000000	4.000000
75%	61.000000	65.000000	11.250000
max	78.000000	69.000000	52.000000

Univeriant Analysis- Histogram

In [13]:

```
sns.FacetGrid(df,hue='status',height= 5)\
.map(sns.distplot,'age')\
.add_legend();
plt.show()
```

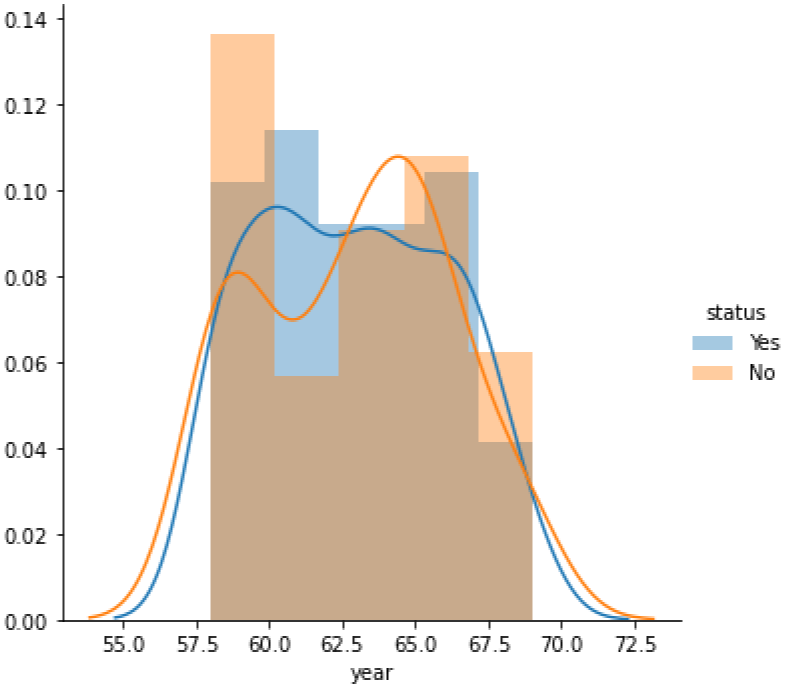
```
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [14]:

```
sns.FacetGrid(df,hue='status',height= 5)\
.map(sns.distplot,'year')\
.add_legend();
plt.show()
```

```
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



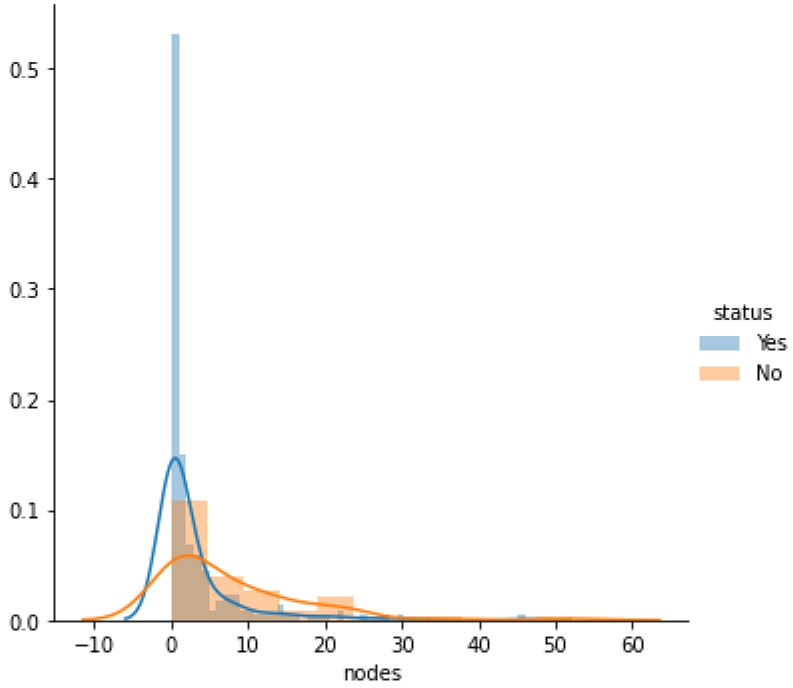
In [16]:

```
sns.FacetGrid(df,hue='status',height= 5)\
.map(sns.distplot,'nodes')\
```

Loading [MathJax]/extensions/Safe.js

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

```
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\anirb\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `
distplot` is a deprecated function and will be removed in a future version. Please adapt y
our code to use either `displot` (a figure-level function with similar flexibility) or `hi
stplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



CDF

In [20]:

```
counts1,bin_edges1= np.histogram(status_yes['nodes'],bins=10)
print(counts1)
print(bin_edges1)
```

```
[188  18   5   6   4   1   2   0   0   1]
[ 0.   4.6  9.2 13.8 18.4 23.  27.6 32.2 36.8 41.4 46. ]
```

In [22]:

```
counts2,bin_edges2= np.histogram(status_no['nodes'],bins=10)
print(counts2)
print(bin_edges2)
```

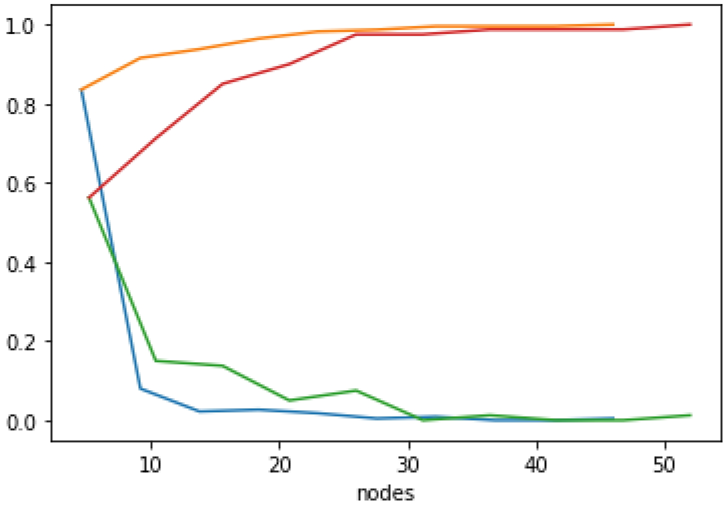
```
[45 12 11  4  6  0  1  0  0  1]
[ 0.   5.2 10.4 15.6 20.8 26.  31.2 36.4 41.6 46.8 52. ]
```

In [23]:

```
pdf1= counts1/(sum(counts1))
print(pdf1)
cdf1= np.cumsum(pdf1)
plt.plot(bin_edges1[1:],pdf1)
plt.plot(bin_edges1[1:],cdf1,label= 'yes')
plt.xlabel('nodes')
print('***********************************************************')
pdf2= counts2/(sum(counts2))
print(pdf2)
um(pdf2)
```

```
plt.plot(bin_edges2[1:],pdf2)
plt.plot(bin_edges2[1:],cdf2,label= 'yes')
plt.xlabel('nodes')
plt.show()
```

```
[0.83555556 0.08       0.02222222 0.02666667 0.01777778 0.00444444
 0.00888889 0.         0.         0.00444444]
********************************************************
[0.5625 0.15   0.1375 0.05   0.075  0.     0.0125 0.     0.     0.0125]
```

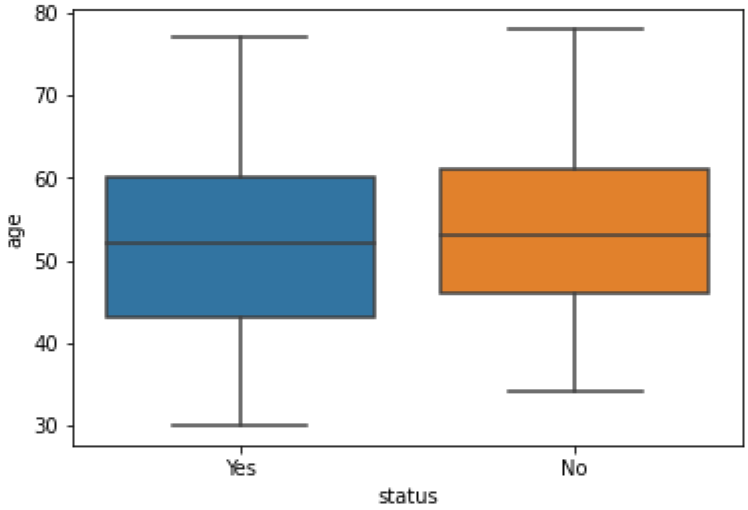


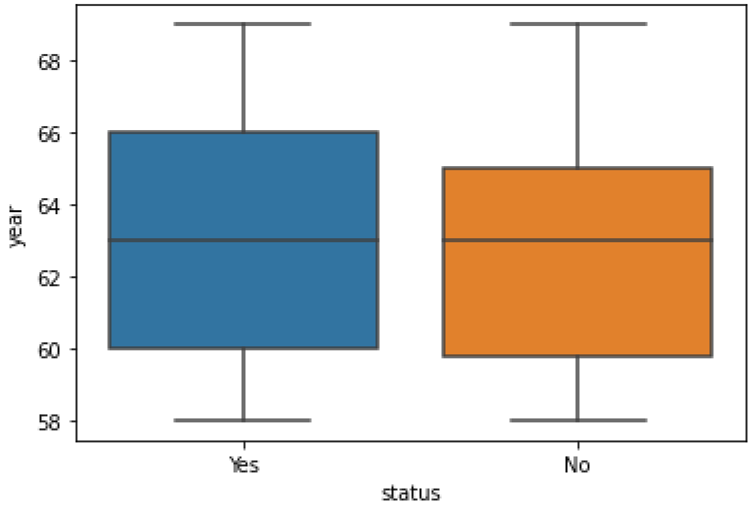
Box Plot :

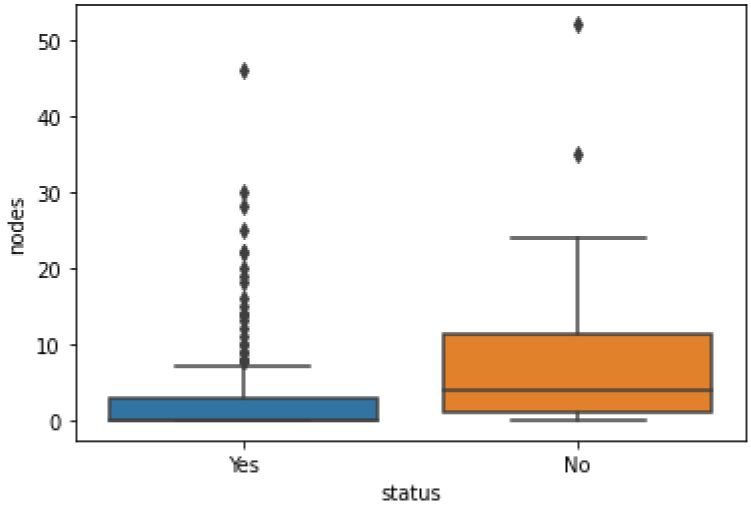
```
In [24]:
sns.boxplot(x='status',y='age',data= df)
plt.show()

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

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





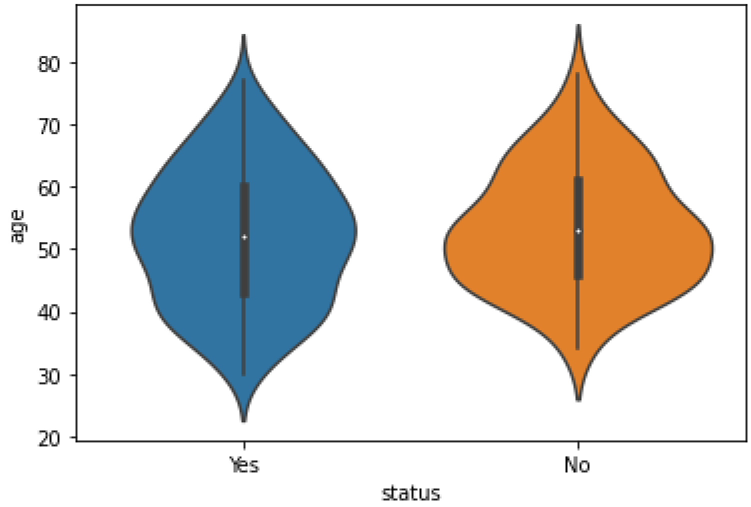


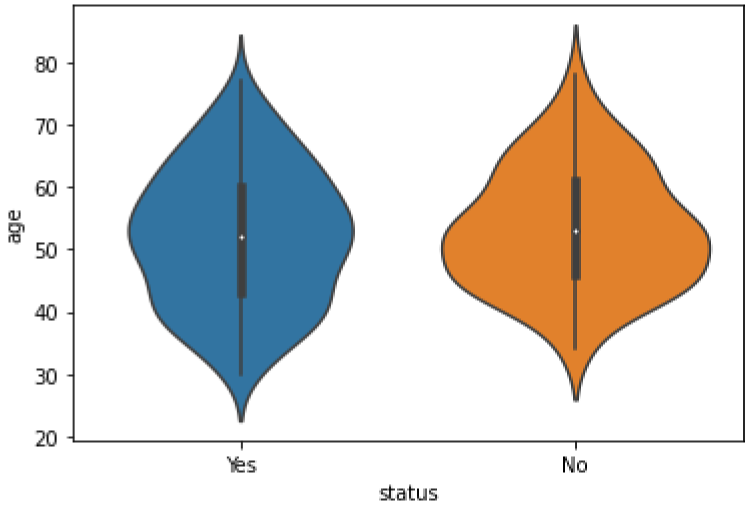
Violin Plot:

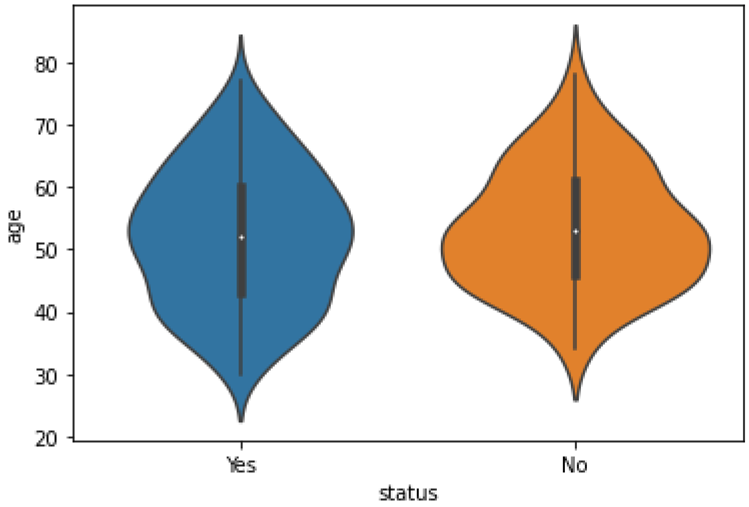
```
In [25]:
sns.violinplot(x='status',y='age',data= df)
plt.show()

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

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



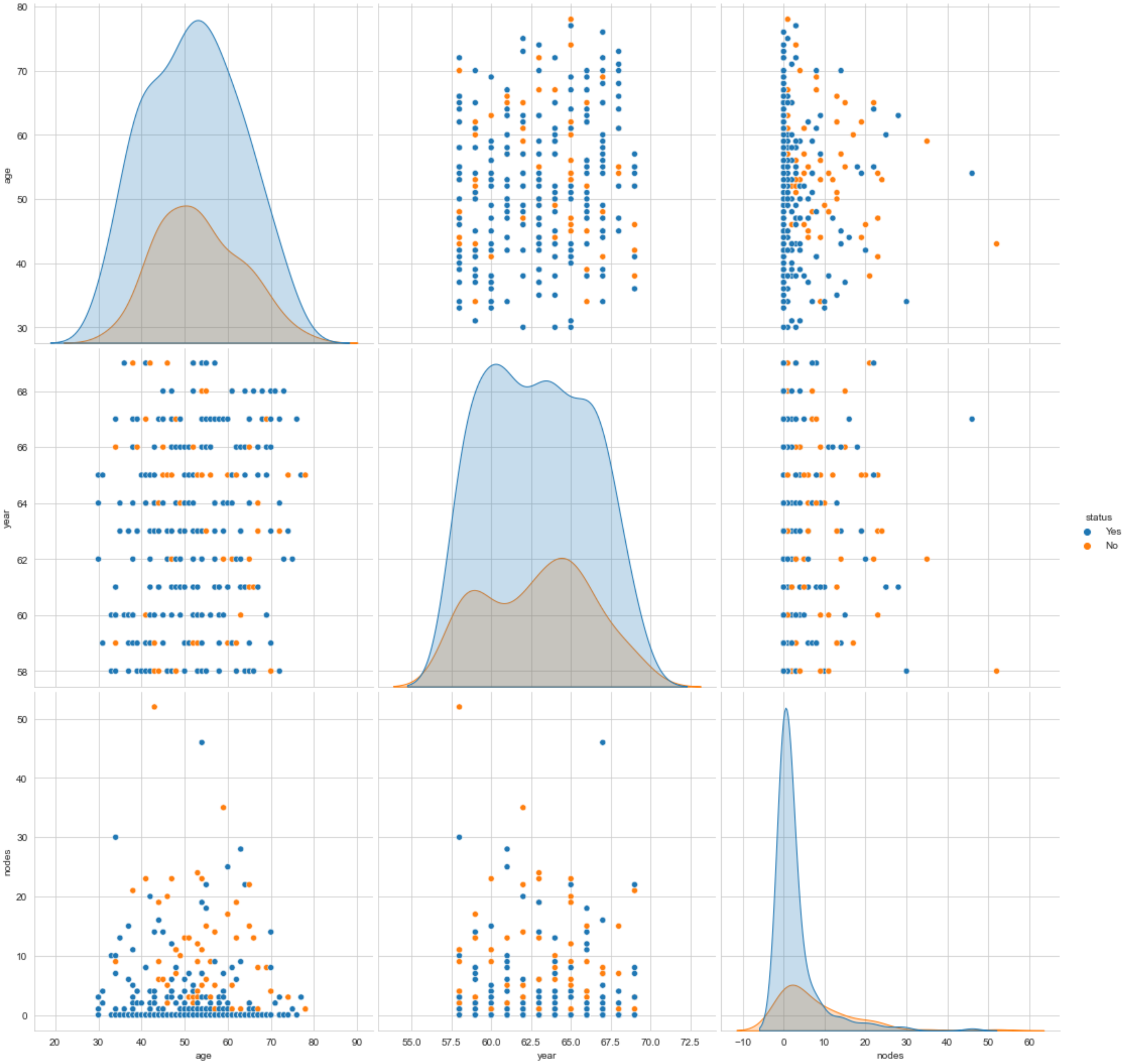




Pair Plot :

In [26]:

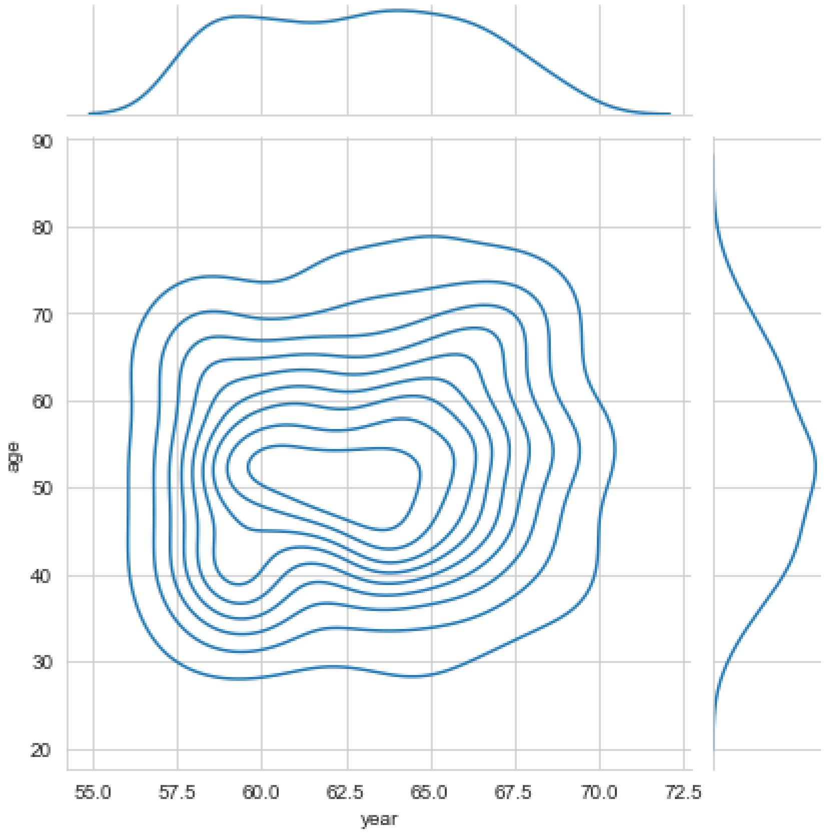
```
sns.set_style('whitegrid')
sns.pairplot(df,hue= 'status',height= 5)
plt.show()
```



Multivariant Analysis:

In [29]:

```
sns.jointplot(x='year',y='age',data= df,kind= 'kde')
plt.show()
```



Thank You

Loading [MathJax]/extensions/Safe.js