objective:

classify the patients data where survives 5 year or longer or died within 5 yrs

In [ ]:
```
Attribute Information:

Feature-1 : Age of patient at time of operation
Feature-2 : Patient's year of operation (year - 1900)
Feature-3 : Number of positive axillary nodes detected
Feature-4 : Survival status

            (class attribute) 1 = the patient survived 5 years or longer

            (class attribute) 2 = the patient died within 5 year
```

In [ ]:
```
import warnings
warnings.filterwarnings("ignore")
```

In [72]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [106]:
```
haberman=pd.read_csv("haberman.csv")
```

In [75]:
```
haberman.shape
```

Out[75]:
```
(306, 4)
```

In [15]:
```
haberman.columns
```

Out[15]:
```
Index(['age', 'year', 'nodes', 'status'], dtype='object')
```

In [20]:
```
haberman['status'].value_counts()
```

Out[20]:
```
1    225
2     81
Name: status, dtype: int64
```

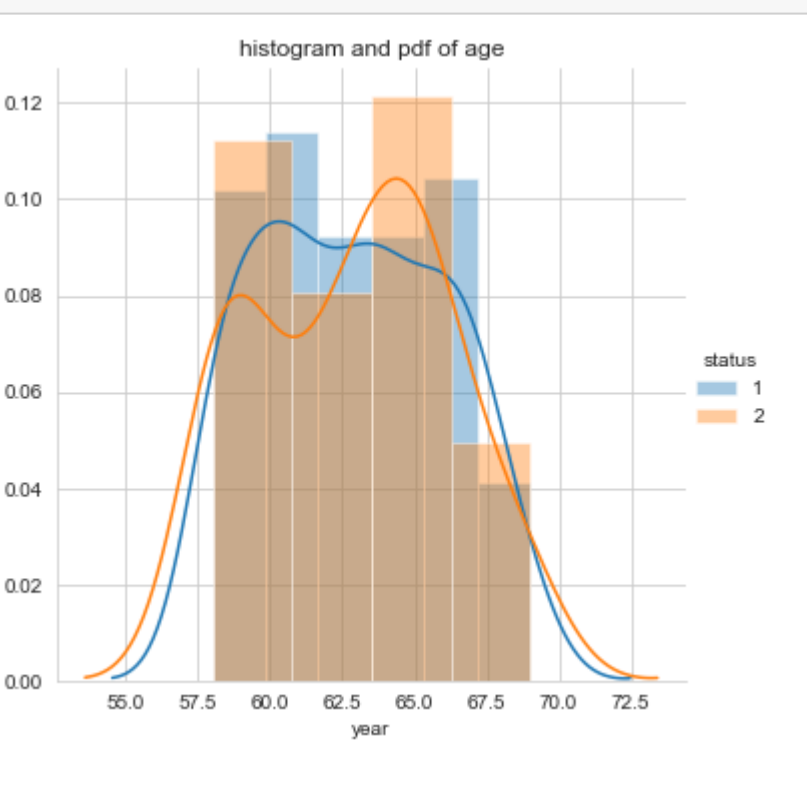
observation:

1. 225 patients survives 5years or longer
2. 81 patients died within 5 yrs

1.univariate analysis

1.1 histogram,pdf,cdf

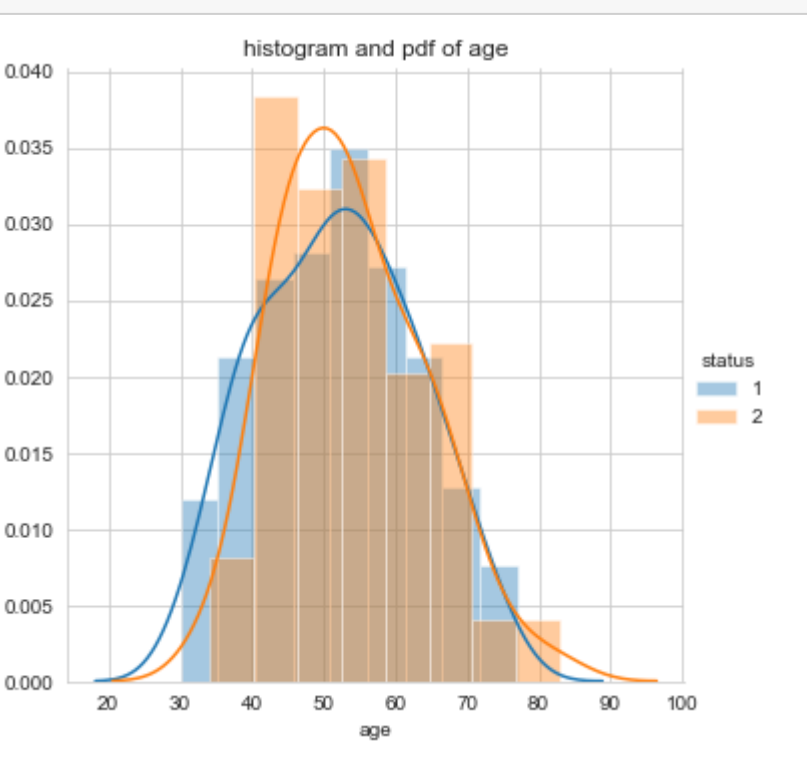
In [107]:
```
sns.FacetGrid(haberman,hue="status",size=5) \
   .map(sns.distplot, "year") \
   .add_legend()
plt.title('histogram and pdf of age')
plt.show()
```



observaion:

As both plots are overlapping on each other cant say much from plot

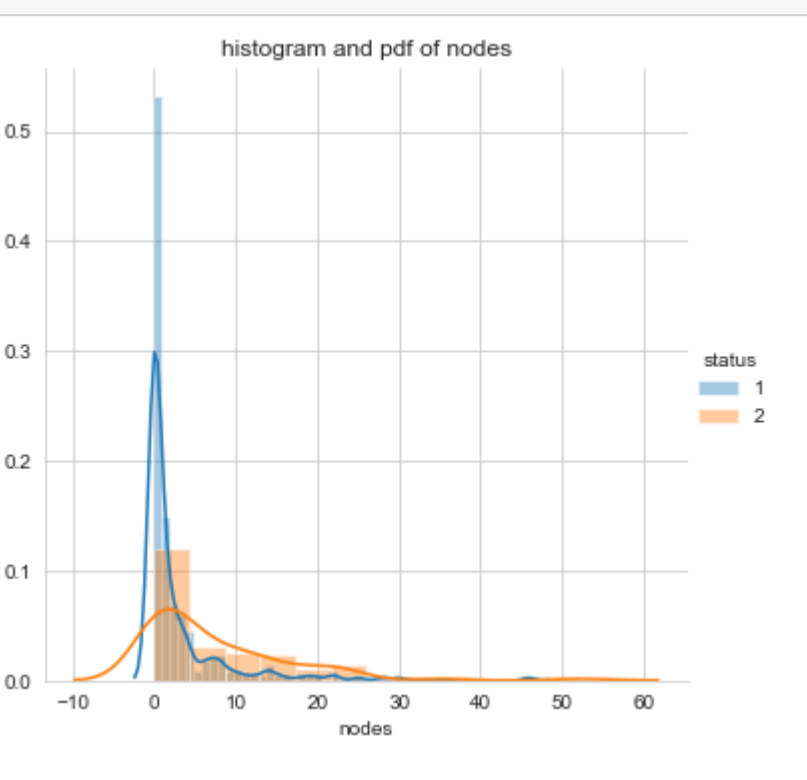
In [77]:
```
sns.FacetGrid(haberman,hue="status",size=5) \
   .map(sns.distplot, "age") \
   .add_legend()
plt.title('histogram and pdf of age')
plt.show()
```



observations:

1.patients from age 30 to 35 survived 5 years or longer after operation 2.patients from age 76 to 83 died within 5 years after operation 3.from age 35 to 76 overlapping is more so cant say much

In [33]:
```
sns.FacetGrid(haberman,hue="status",size=5) \
   .map(sns.distplot,"nodes") \
   .add_legend()
plt.title('histogram and pdf of nodes')
plt.show()
```



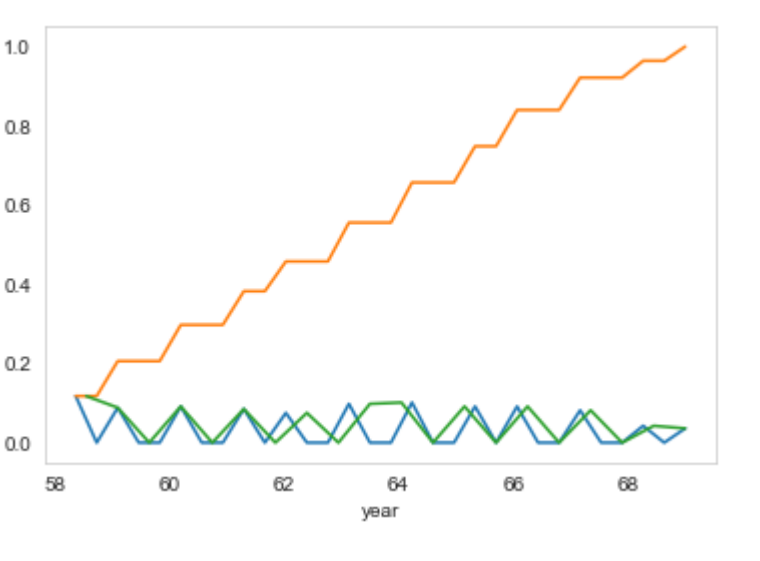
observation:

no. of positive auxillary positive increases the survival status

In [61]:
```
#pdf and cdf

counts,bin_edges = np.histogram(haberman['year'], bins=30,
                                 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(haberman['year'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
plt.plot(bin_edges[1:],pdf);
plt.xlabel('year')
plt.grid()
plt.show();
```

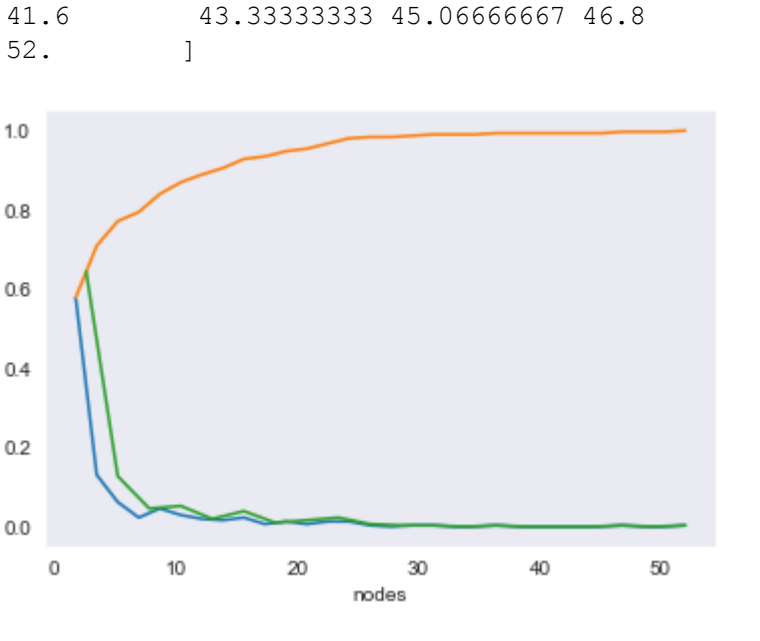
```
[0.11764706 0.         0.08823529 0.         0.         0.09150327
 0.         0.         0.08496732 0.         0.0751634  0.
 0.         0.09803922 0.         0.         0.10130719 0.
 0.         0.09150327 0.         0.09150327 0.         0.
 0.08169935 0.         0.         0.04248366 0.         0.03594771]
[58.         58.36666667 58.73333333 59.1        59.46666667 59.83333333
 60.2        60.56666667 60.93333333 61.3        61.66666667 62.03333333
 62.4        62.76666667 63.13333333 63.5        63.86666667 64.23333333
 64.6        64.96666667 65.33333333 65.7        66.06666667 66.43333333
 66.8        67.16666667 67.53333333 67.9        68.26666667 68.63333333
 69.        ]
```



In [127]:
```
#pdf of nodes
counts,bin_edges = np.histogram(haberman['nodes'], bins=30,
                                 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(haberman['nodes'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
plt.plot(bin_edges[1:],pdf);
plt.xlabel('nodes')
plt.grid()

plt.show();
```

```
[0.57843137 0.13071895 0.0620915  0.02287582 0.04575163 0.02941176
 0.01960784 0.01633987 0.02287582 0.00653595 0.0130719  0.00653595
 0.0130719  0.0130719  0.00326797 0.         0.00326797 0.00326797
 0.         0.         0.00326797 0.         0.         0.
 0.         0.         0.00326797 0.         0.         0.00326797]
[ 0.          1.73333333  3.46666667  5.2         6.93333333  8.66666667
 10.4        12.13333333 13.86666667 15.6        17.33333333 19.06666667
 20.8        22.53333333 24.26666667 26.         27.73333333 29.46666667
 31.2        32.93333333 34.66666667 36.4        38.13333333 39.86666667
 41.6        43.33333333 45.06666667 46.8        48.53333333 50.26666667
 52.        ]
```

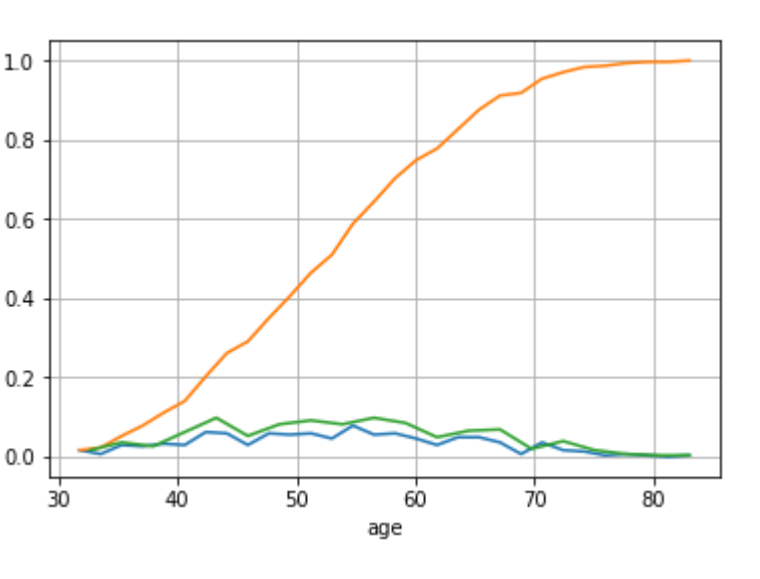


In [87]:
```
#pdf of age
counts,bin_edges = np.histogram(haberman['age'], bins=30,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges);
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf);
plt.plot(bin_edges[1:],cdf)

counts, bin_edges = np.histogram(haberman['age'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
plt.plot(bin_edges[1:],pdf);
plt.xlabel('age')
plt.grid()

plt.show();
```

```
[0.01633987 0.00653595 0.02941176 0.02614379 0.03267974 0.02941176
 0.0620915  0.05882353 0.02941176 0.05882353 0.05555556 0.05882353
 0.04575163 0.07843137 0.05555556 0.05882353 0.04575163 0.02941176
 0.04901961 0.04901961 0.03594771 0.00653595 0.03594771 0.01633987
 0.0130719  0.00326797 0.00653595 0.00326797 0.         0.00326797]
[30.         31.76666667 33.53333333 35.3        37.06666667 38.83333333
 40.6        42.36666667 44.13333333 45.9        47.66666667 49.43333333
 51.2        52.96666667 54.73333333 56.5        58.26666667 60.03333333
 61.8        63.56666667 65.33333333 67.1        68.86666667 70.63333333
 72.4        74.16666667 75.93333333 77.7        79.46666667 81.23333333
 83.        ]
```

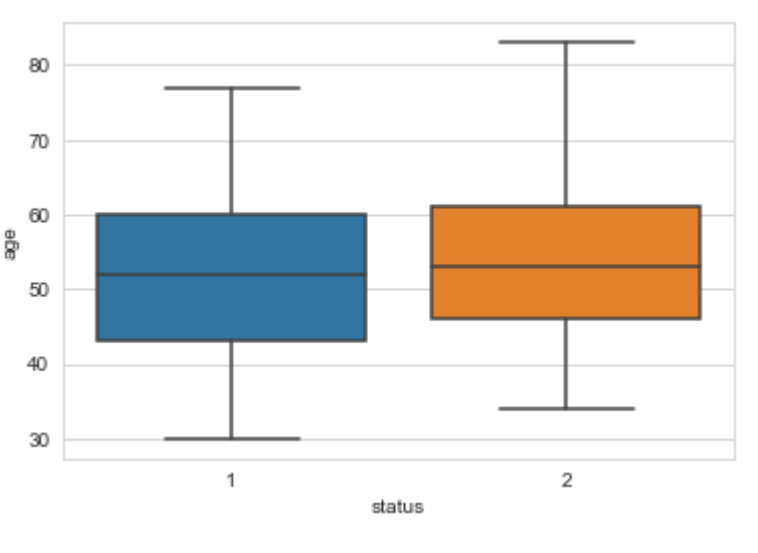


box plot and whisker

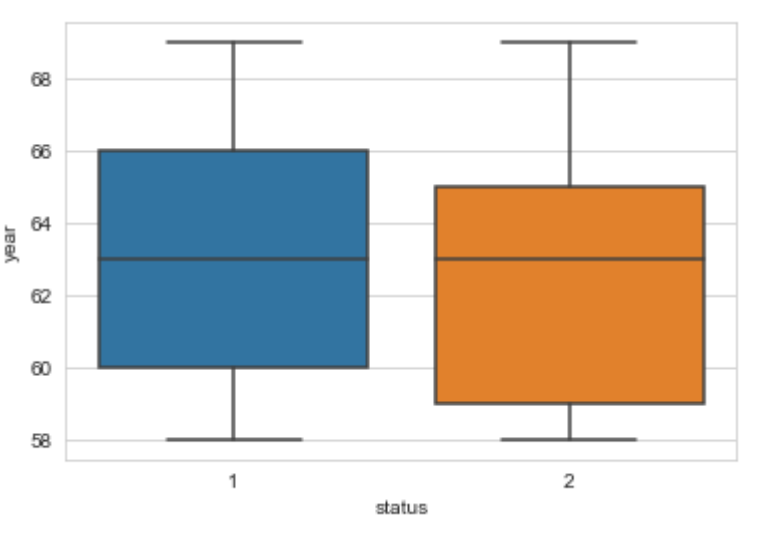
sns.boxplot(x='status',y='year', data=haberman) sns.boxplot(x='status',y='age', data=haberman)

plt.show()

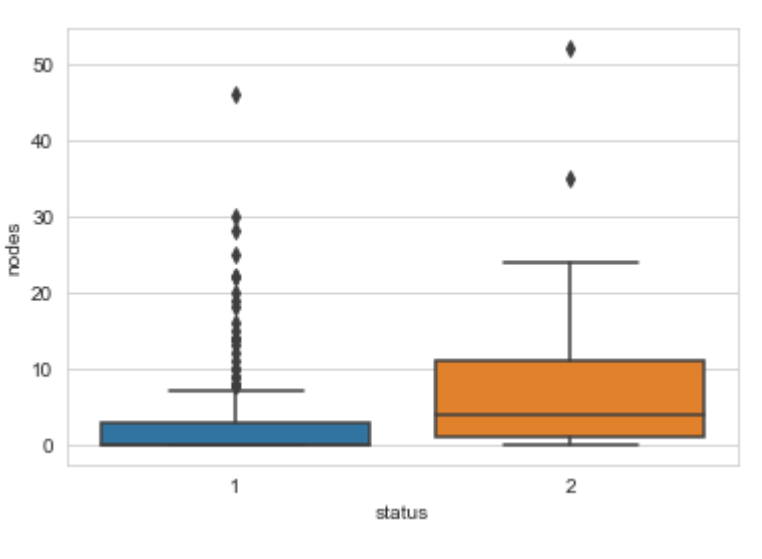
In [109]:
```
sns.boxplot(x='status',y='age', data=haberman)
plt.show()
```



In [110]:
```
sns.boxplot(x='status',y='year', data=haberman)
plt.show()
```



In [121]:
```
sns.boxplot(x='status',y='nodes', data=haberman)
plt.show()
```

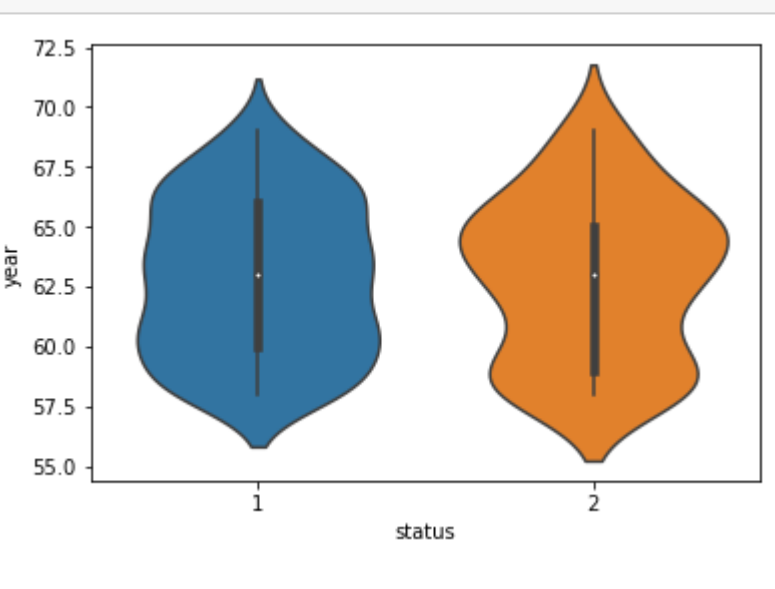


observations:

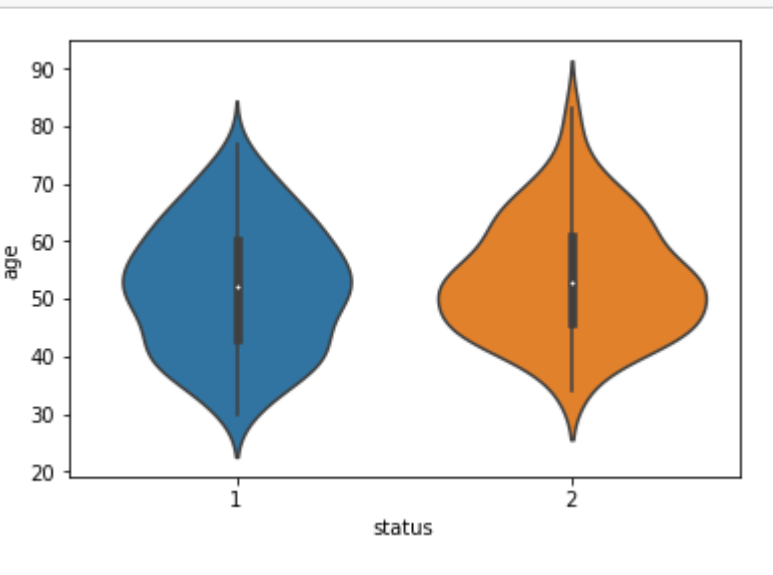
1.From this plot, it is evident that the class 'survived>=5yrs' has 75percentile of count of axillary nodes having value less than 5.

violin plot

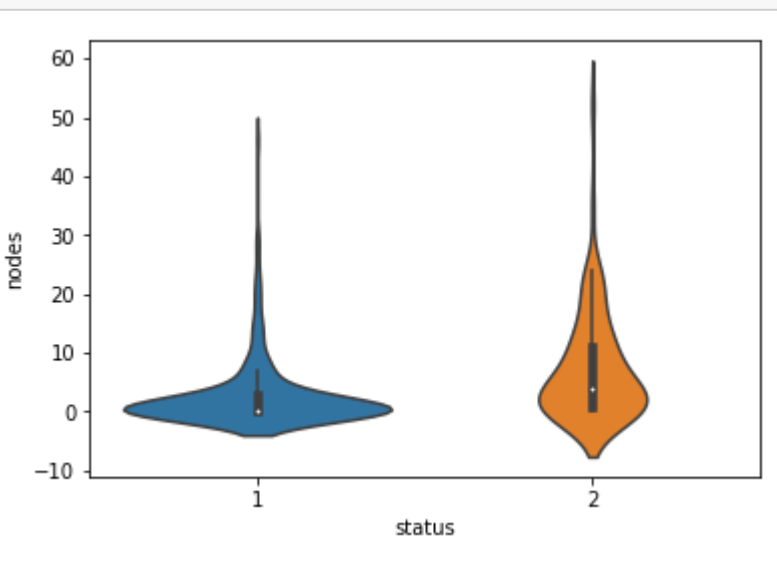
In [95]:
```
sns.violinplot(x="status", y="year", data=haberman, size=8)
plt.show()
```



In [98]:
```
sns.violinplot(x="status", y="age", data=haberman, size=8)
plt.show()
```



In [97]:
```
sns.violinplot(x="status", y="nodes", data=haberman, size=8)
plt.show()
```



observation:

postive auxillary nodes are zero

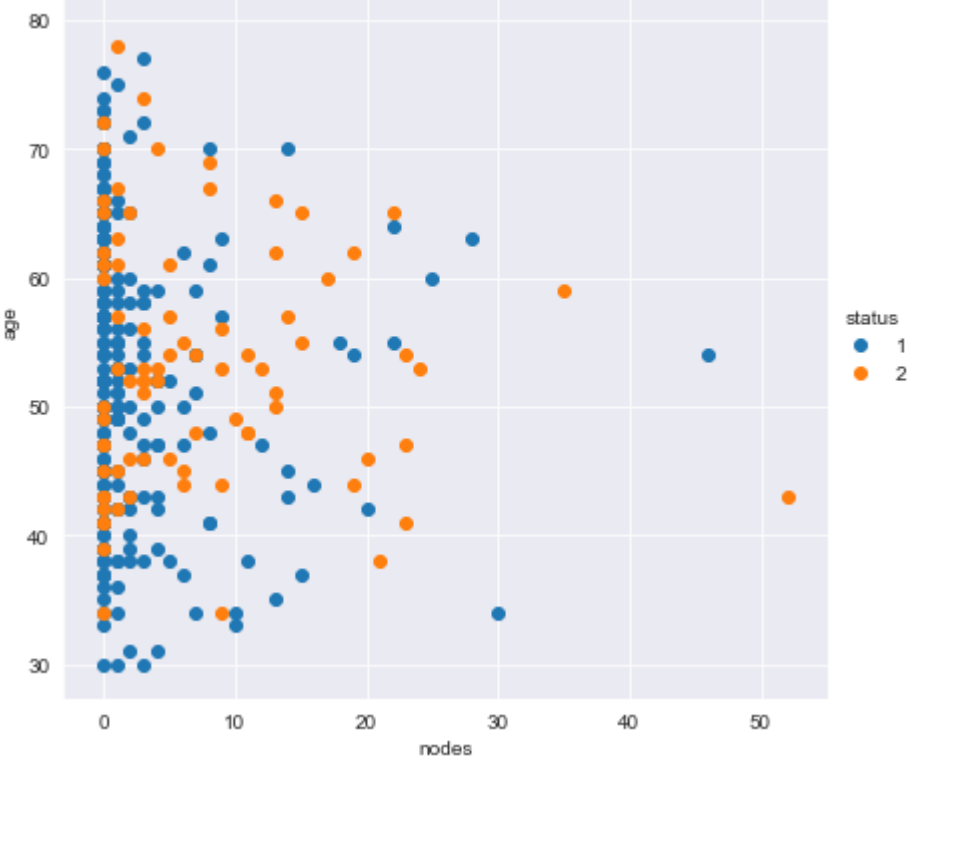
Bivariate analysis

2-D scatter plot

In [108]:
```
sns.set_style("darkgrid");
sns.FacetGrid(haberman, hue="status", size=6) \
   .map(plt.scatter, "year", "age") \
   .add_legend();
plt.show();
```

```
C:\Users\Admin\Anaconda3\lib\site-packages\seaborn\axisgrid.py:230: UserWarning: The `size` param
ter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```

In [124]:
```
sns.set_style("darkgrid");
sns.FacetGrid(haberman, hue="status",size=6) \
   .map(plt.scatter, "nodes", "age") \
   .add_legend();
plt.show();
```



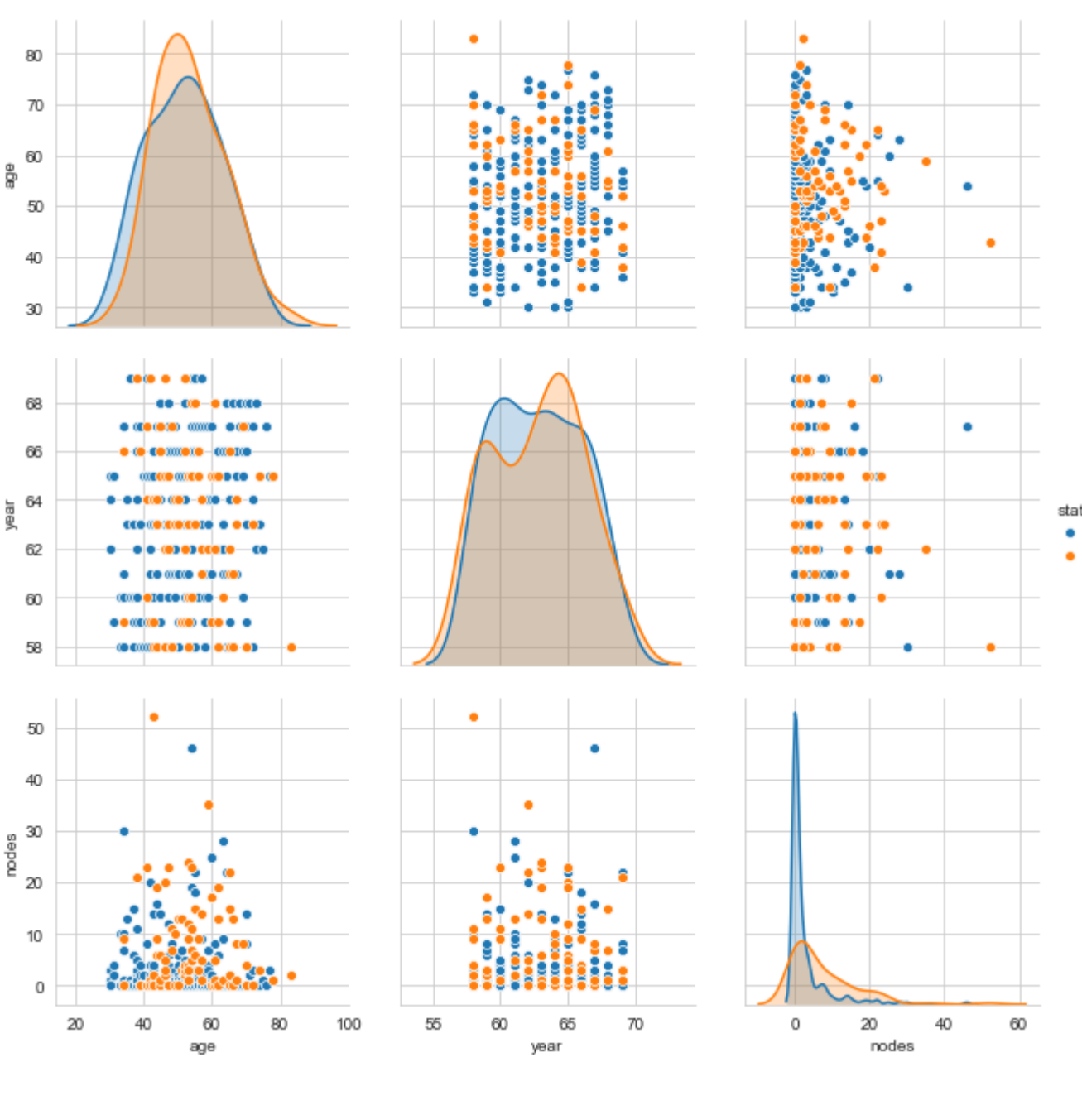
observation:

points are overlapping in both year and nodes data

Pair plot

In [28]:
```
plt.close();
sns.set_style("whitegrid");
sns.pairplot(haberman, hue="status",vars=['age','year','nodes'],size=3);
plt.show()
```

```
C:\Users\Admin\Anaconda3\lib\site-packages\seaborn\axisgrid.py:2065: UserWarning: The `size` para
meter has been renamed to `height`; pleaes update your code.
  warnings.warn(msg, UserWarning)
```



observation:

-positive auxillary nodes is a useful to identify the survival status of cancer patients
-age and year are overlapping on each other

conclusion: