In [2]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np


'''downlaod iris.csv from https://raw.githubusercontent.com/uiuc-cse/data-f
#Load Iris.csv into a pandas dataFrame.
df = pd.read_csv("haberman.csv")
```

In [9]:
```
# (Q) how many data-points and features?
print (df.shape)
print(df.columns)
```

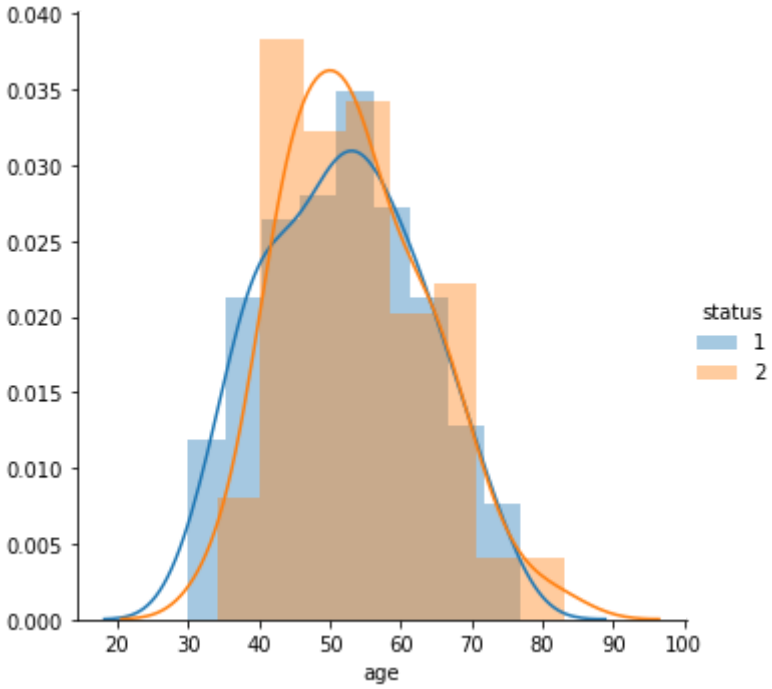
```
(306, 4)
Index(['age', 'year', 'nodes', 'status'], dtype='object')
```

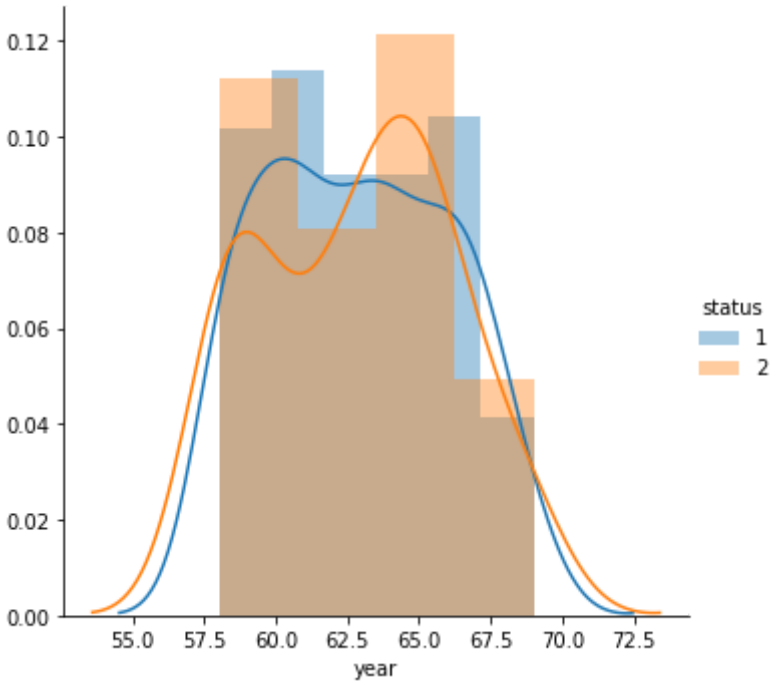
In [8]:
```
# how many data points per class
df.status.value_counts()
```

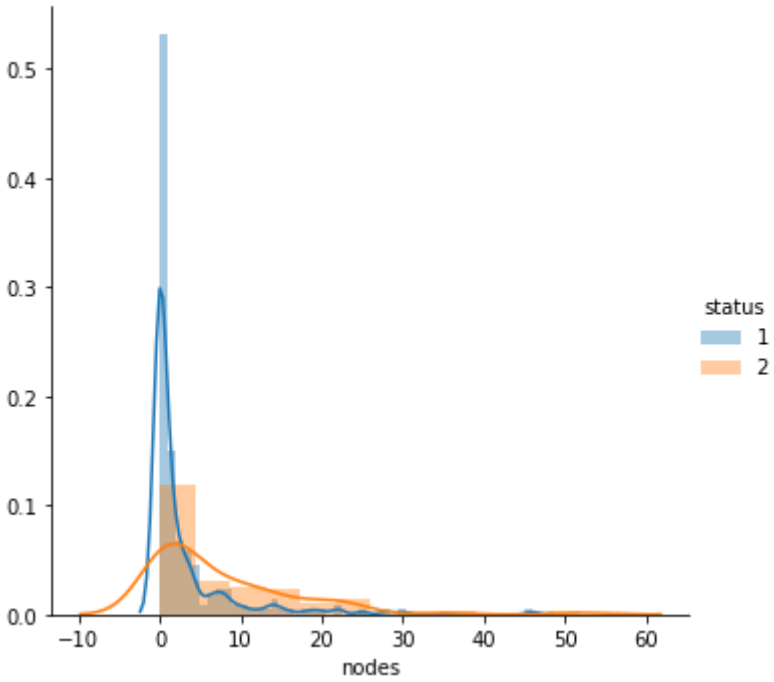
Out[8]:
```
1    225
2     81
Name: status, dtype: int64
```

Our objective is to classify the suvival status based on 3 given features

In [13]:
```
# Prob Dist Function of features
sns.FacetGrid(df, hue="status", height=5) \
   .map(sns.distplot, "age") \
   .add_legend();
plt.show();

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

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



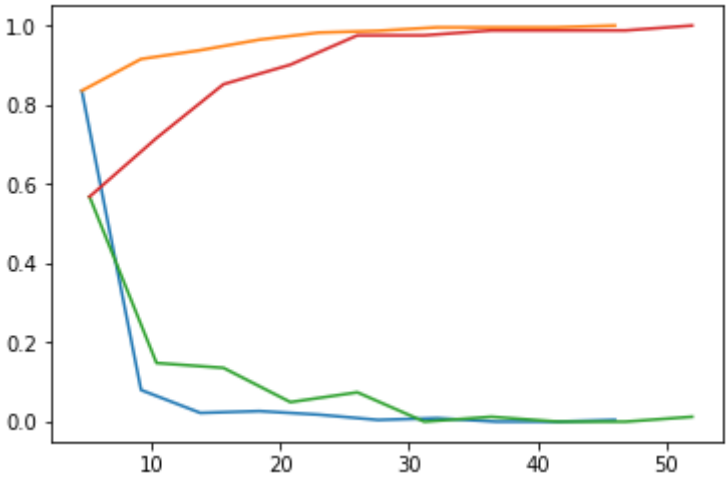

0.12
0.10
0.08
0.06
0.04
0.02
0.00
55.0
57.5
60.0
62.5
65.0
67.5
70.0
72.5
year
status
1
2


0.5
0.4
0.3
0.2
0.1
0.0
−10
0
10
20
30
40
50
60
nodes
status
1
2

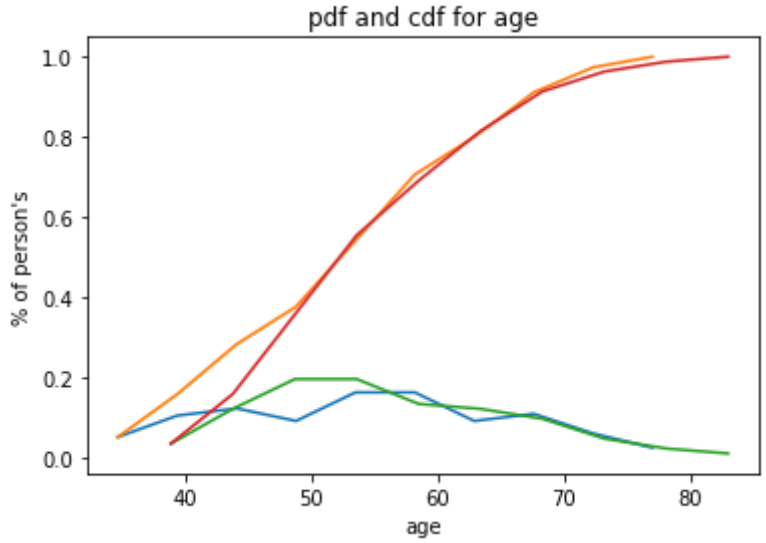
In [23]:
```
df_live = df.loc[df["status"] == 1];
df_dead = df.loc[df["status"] == 2];


counts, bin_edges = np.histogram(df_live['nodes'], bins=10,
                                 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)


# dead
counts, bin_edges = np.histogram(df_dead['nodes'], bins=10,
                                 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)
plt.show()




counts, bin_edges = np.histogram(df_live['age'], bins=10,
                                 density = True)
label = ["pdf of class 1", "cdf of class 1", "pdf of class 2", "cdf of clas

plt.title("pdf and cdf for age")
plt.xlabel("age")
plt.ylabel("% of person's")
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)


# dead
counts, bin_edges = np.histogram(df_dead['age'], bins=10,
                                 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)
plt.show()
```

```
counts, bin_edges = np.histogram(df_live['year'], bins=10,
                                 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)


# dead
counts, bin_edges = np.histogram(df_dead['year'], bins=10,
                                 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)
```

```
[0.83555556 0.08       0.02222222 0.02666667 0.01777778 0.00444444
 0.00888889 0.         0.         0.00444444]
[ 0.   4.6  9.2 13.8 18.4 23.  27.6 32.2 36.8 41.4 46. ]
[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. ]
```

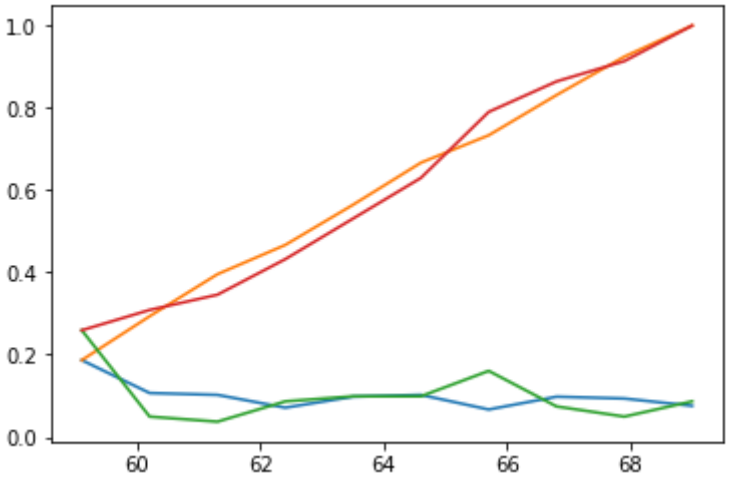



```
[0.05333333 0.10666667 0.12444444 0.09333333 0.16444444 0.16444444
 0.09333333 0.11111111 0.06222222 0.02666667]
[30.  34.7 39.4 44.1 48.8 53.5 58.2 62.9 67.6 72.3 77. ]
[0.03703704 0.12345679 0.19753086 0.19753086 0.13580247 0.12345679
 0.09876543 0.04938272 0.02469136 0.01234568]
[34.  38.9 43.8 48.7 53.6 58.5 63.4 68.3 73.2 78.1 83. ]
```



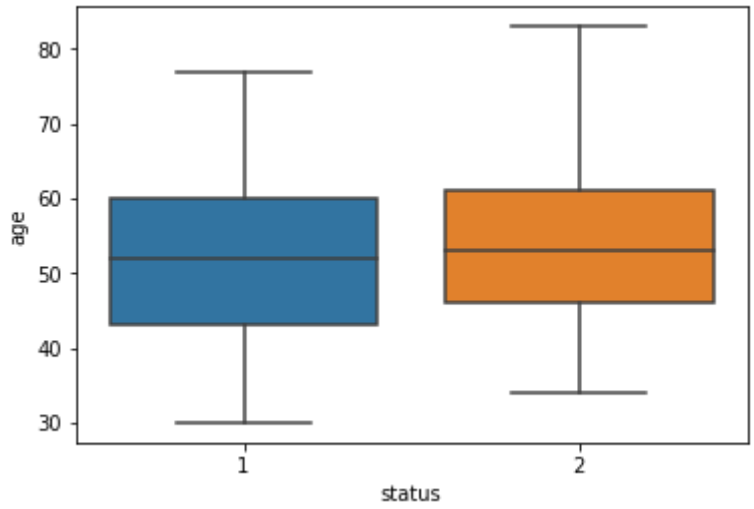

```
[0.18666667 0.10666667 0.10222222 0.07111111 0.09777778 0.10222222
 0.06666667 0.09777778 0.09333333 0.07555556]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
[0.25925926 0.04938272 0.03703704 0.08641975 0.09876543 0.09876543
 0.16049383 0.07407407 0.04938272 0.08641975]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
```

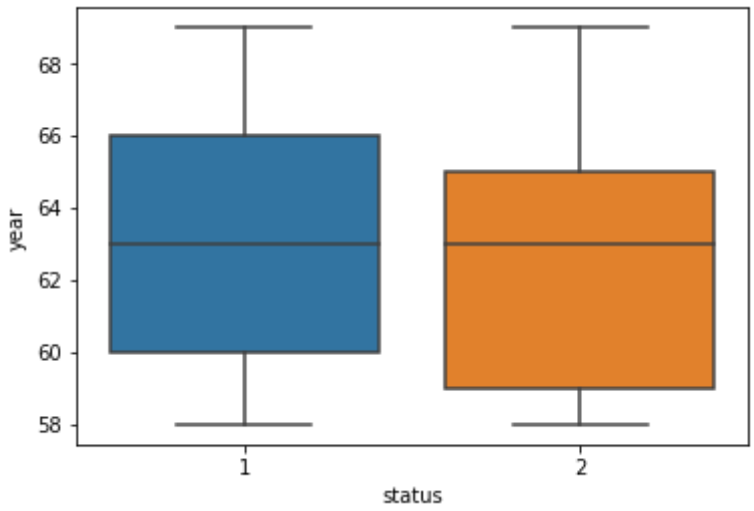
Out[23]: [<matplotlib.lines.Line2D at 0x1a16dfdd68>]

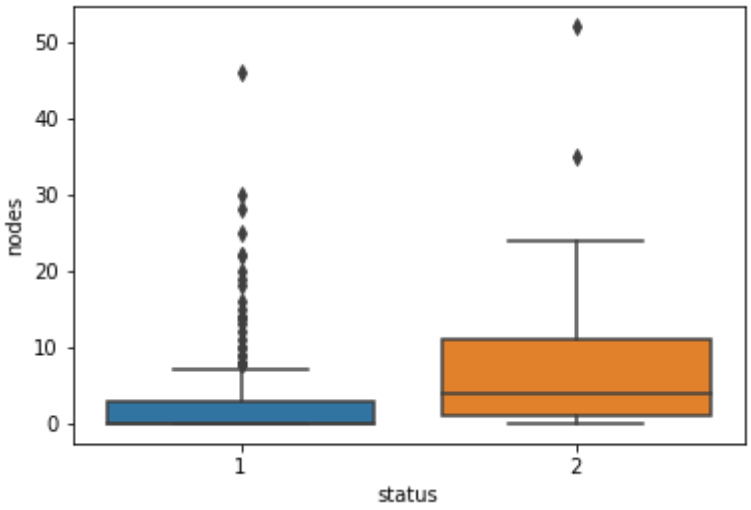


In [25]:

```
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()
```

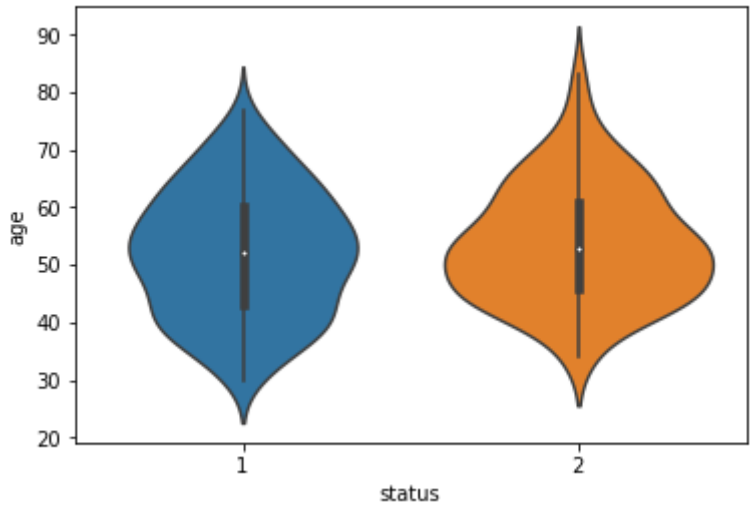


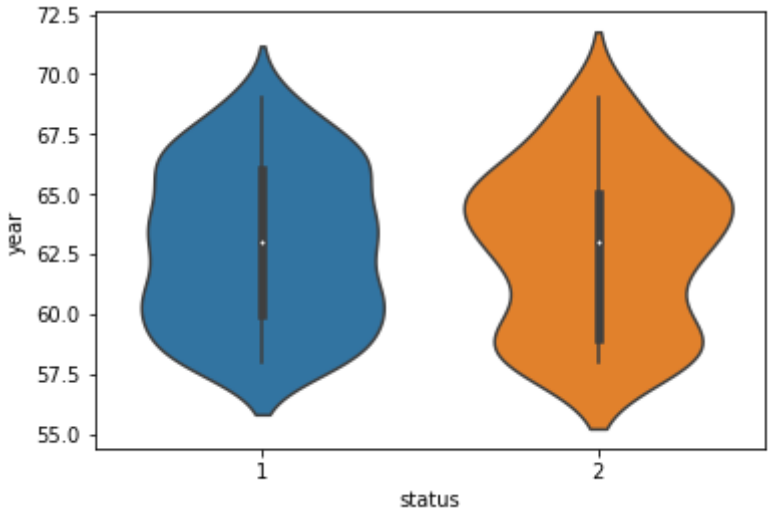


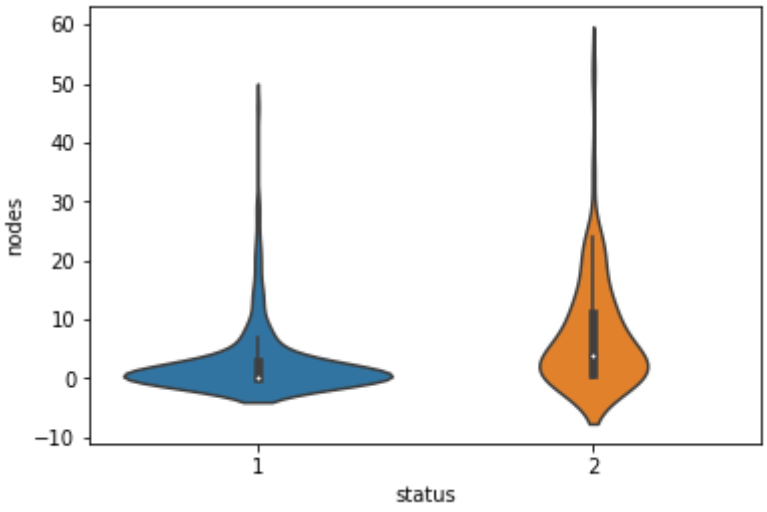

50
40
30
20
10
0
nodes
1
2
status

In [26]:

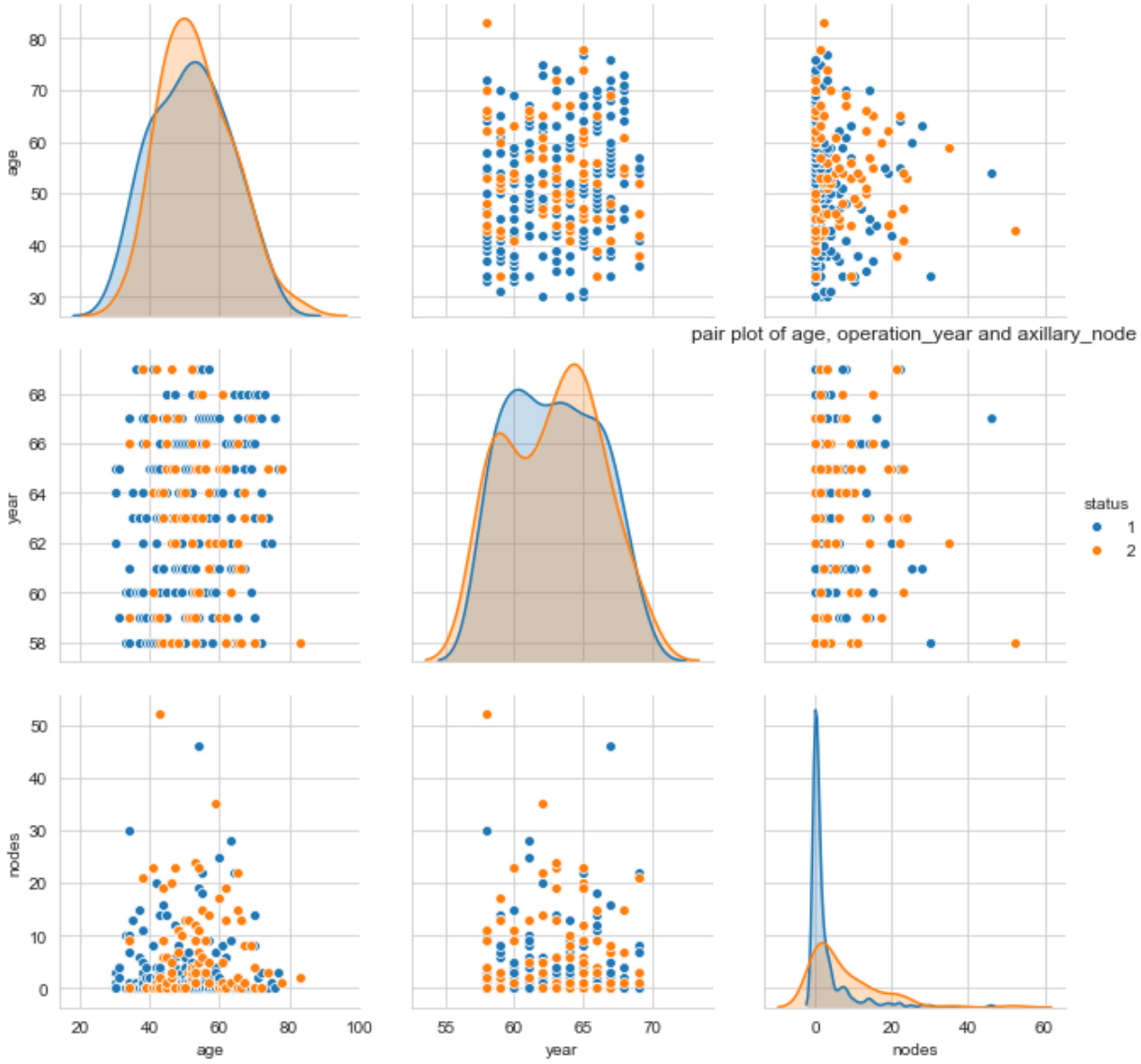
```
sns.violinplot(x='status',y='age', data=df)
plt.show()


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


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






60
50
40
30
20
10
0
−10
nodes
1
2
status

In [5]:
```
plt.close();

sns.set_style("whitegrid")
sns.pairplot(df, hue = "status", vars = ["age", "year", "nodes"], height =
plt.title("pair plot of age, operation_year and axillary_node")
plt.show()
```



Conclusion:

1. The dataset is imbalanced for status
2. From the boxplot and CDF we can conclude nodes< 5 has 90% possibility to survive and nodes>20 has 90% possibility to be dead 3.15% of the person's have less than or equal to age 37 who survived.
3. 60-65 age group, more person died who has less than 6 axillary_lymph_node.

In [ ]: