1. Download Haberman Cancer Survival dataset from Kaggle. You may have to create a Kaggle account to donwload data. (https://www.kaggle.com/gilsousa/habermans-survival-data-set)

In [32]:
```
import warnings

warnings.filterwarnings("ignore")
```

2. Perform a similar alanlaysis as above on this dataset with the following sections:

In [33]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
#Load Haberman.csv into a pandas dataFrame.
Haberman = pd.read_csv("C:/Users/aaa/Desktop/saraswati aaic/Haberman.csv")
Haberman.head()
```

Out[33]:

	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

### High level statistics

* High level statistics of the dataset: number of points, numer of features, number of classes, data-points per class.

In [34]:
```
# (Q) how many data-points and features?
print ("No. of Datapoints",Haberman.shape[0])
print("No. of features",Haberman.shape[1])
```

```
No. of Datapoints 306
No. of features 4
```

In [35]:
```
# print no of classes and data point of each class

Haberman["status"].value_counts()
```

Out[35]:
```
1    225
2     81
Name: status, dtype: int64
```

Observations

1)The dataset has 4 features and 305 data points.
2)The dataset has a collection of data of patient aged between 30-83 years those who had undergone cancer surgery in year 1958-1969.
3)Almost 75% of the patient had 0-4 nodes where 25% of them had 0 node and very few had up to 52 nodes.
4)The dataset has 224 datapoint labeled as "1" and 81 datapoint labeled as "2"
5)The dataset is an imbalance dataset.

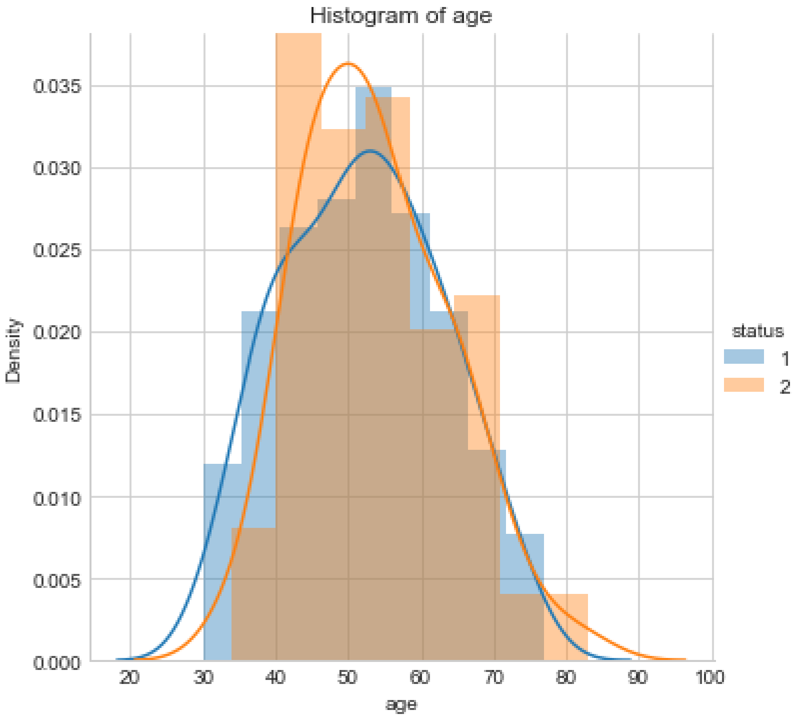
### Explain our objective.

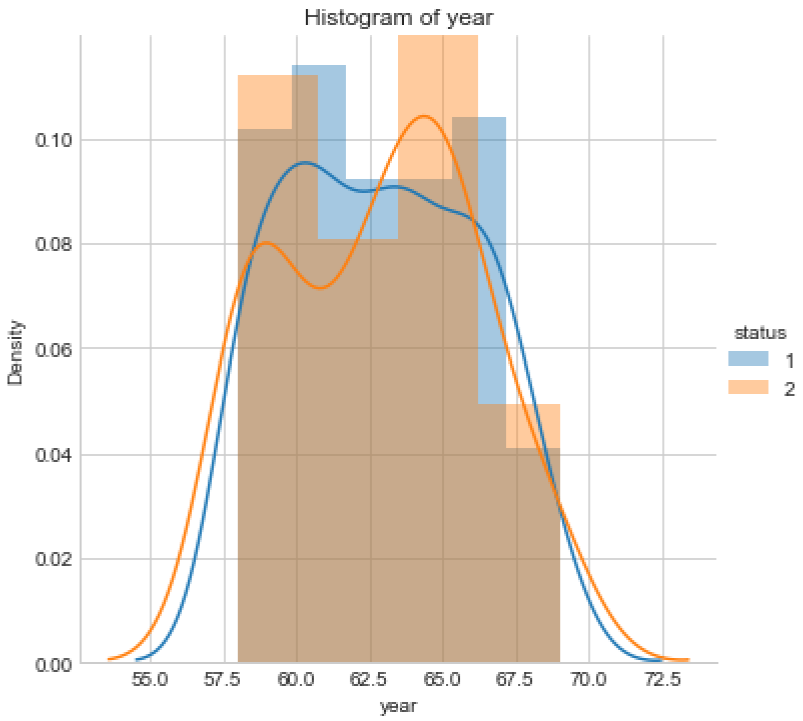
Our objective is to classify a new patient belonging to status 1 or status 2 with the help of given data
i.e age years and nodes

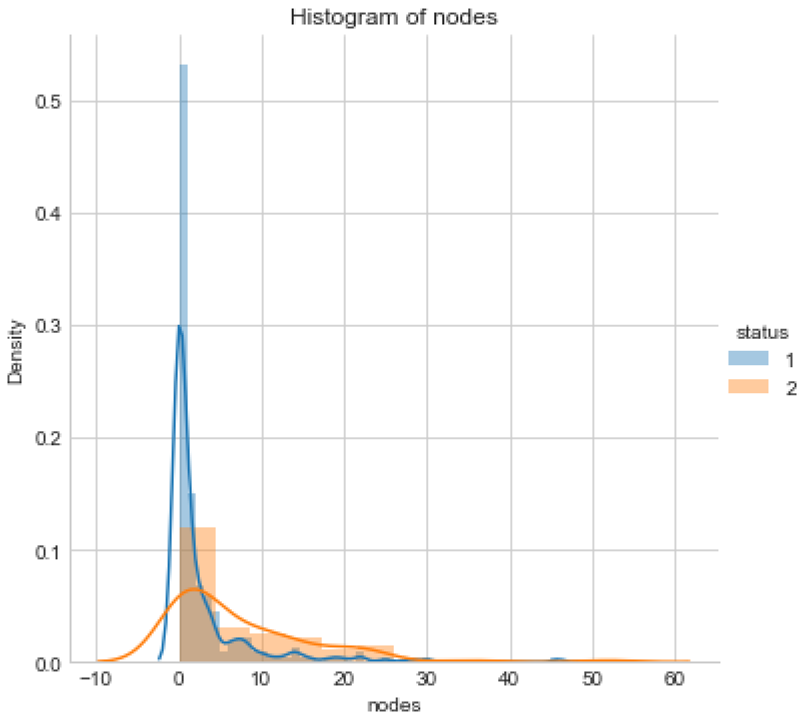
* Perform Univaraite analysis(PDF, CDF, Boxplot, Voilin plots) to understand which features are useful towards classification.

### Histograms ,Pdf and Cdf

```
In [36]: #  plot the pdf of given data
         sns.FacetGrid(Haberman, hue="status", size=5) \
            .map(sns.distplot, "age") \
            .add_legend();
         plt.title("Histogram of age")
         plt.ylabel("Density")
         plt.show();

         sns.FacetGrid(Haberman, hue="status", size=5) \
            .map(sns.distplot, "year") \
            .add_legend();
         plt.title("Histogram of year")
         plt.ylabel("Density")
         plt.show();

         sns.FacetGrid(Haberman, hue="status", size=5) \
            .map(sns.distplot, "nodes") \
            .add_legend();
         plt.title("Histogram of nodes")
         plt.ylabel("Density")
         plt.show();
```







Observations:

1)The survival status corresponding to operation year data points are overlapping,
hence no conclusion about the survival status of the patient could be drawn based on the Year of operation.

2)The data is overlapping hence no major information could be gained.
Patients with age less than 40 years has higher chance to survive and patient with age more than
78 yrs are most likely to die within 5 years of surgery.

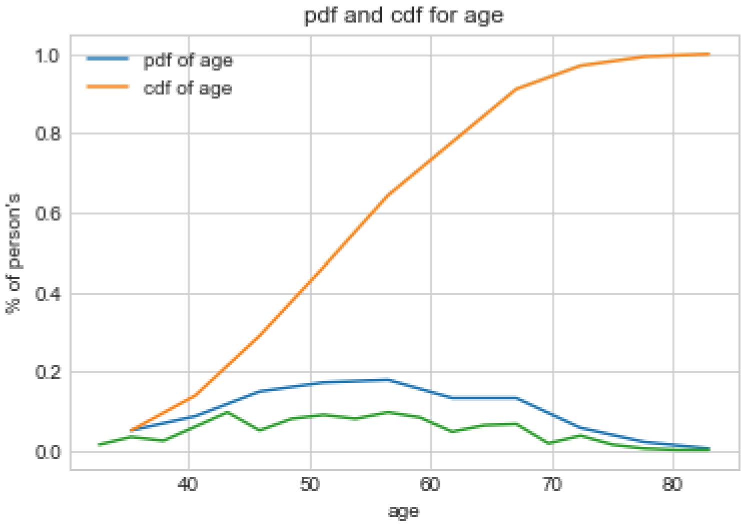
```
3)It is seen that 95% of the patient has nodes between 0 to 25.
 Patient with 0-3 node had higher chances of survival.
 Data is overlapping hence we can't find "point" and "if-else" conditions to
build a simple model.
```

In [37]:

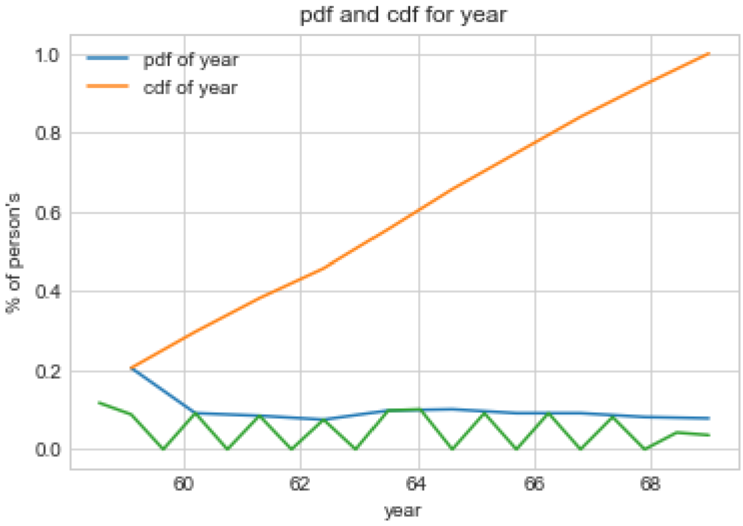
```
# plot pdf and cdf for the same

counts, bin_edges = np.histogram(Haberman['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)


counts, bin_edges = np.histogram(Haberman['age'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
label=["pdf of age","cdf of age"]
plt.legend(label);
plt.title("pdf and cdf for age")
plt.xlabel("age")
plt.ylabel("% of person's")
plt.plot(bin_edges[1:],pdf);

plt.show();

counts, bin_edges = np.histogram(Haberman['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)


counts, bin_edges = np.histogram(Haberman['year'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
label=["pdf of year","cdf of year"]
plt.legend(label);
plt.title("pdf and cdf for year")
plt.xlabel("year")
plt.ylabel("% of person's")
plt.plot(bin_edges[1:],pdf);

plt.show();

counts, bin_edges = np.histogram(Haberman['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)
```

```
counts, bin_edges = np.histogram(Haberman['nodes'], bins=20,
                                 density = True)
pdf = counts/(sum(counts))
label=["pdf of nodes","cdf of nodes"]
plt.legend(label);
plt.title("pdf and cdf for nodes")
plt.xlabel("nodes")
plt.ylabel("% of person's")
plt.plot(bin_edges[1:],pdf);

plt.show();
```

```
[0.05228758 0.08823529 0.1503268  0.17320261 0.17973856 0.13398693
 0.13398693 0.05882353 0.02287582 0.00653595]
[30.  35.3 40.6 45.9 51.2 56.5 61.8 67.1 72.4 77.7 83. ]
```

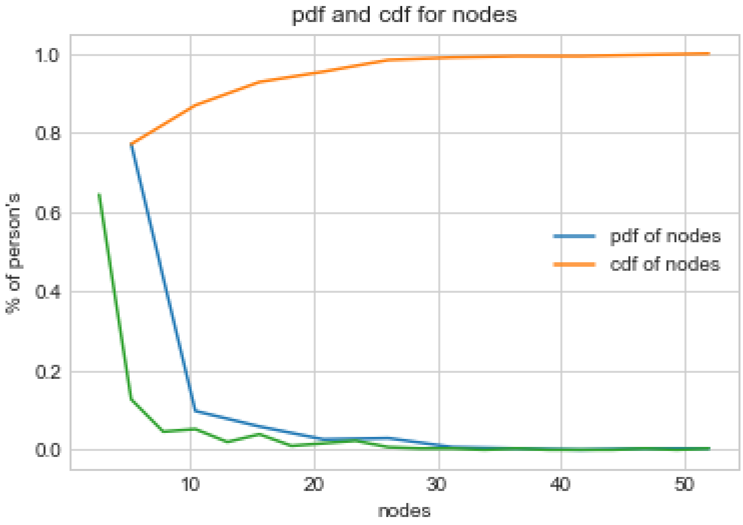


```
[0.20588235 0.09150327 0.08496732 0.0751634  0.09803922 0.10130719
 0.09150327 0.09150327 0.08169935 0.07843137]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
```



```
[0.77124183 0.09803922 0.05882353 0.02614379 0.02941176 0.00653595
 0.00326797 0.         0.00326797 0.00326797]
```

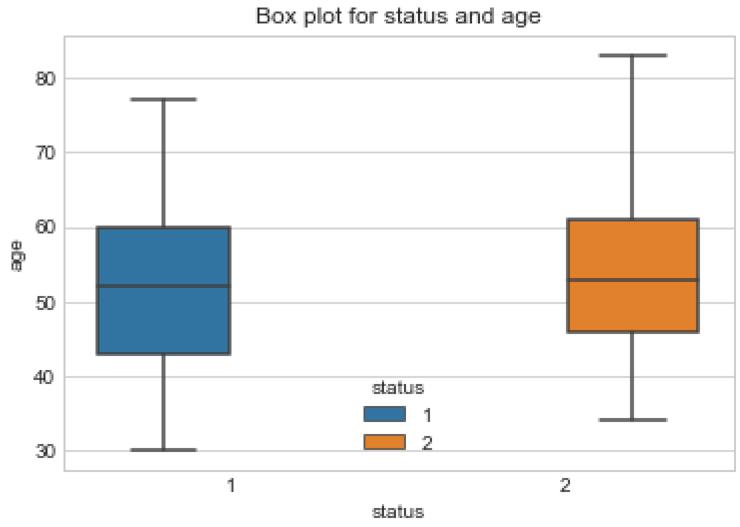
```
[ 0.   5.2 10.4 15.6 20.8 26.  31.2 36.4 41.6 46.8 52. ]
```

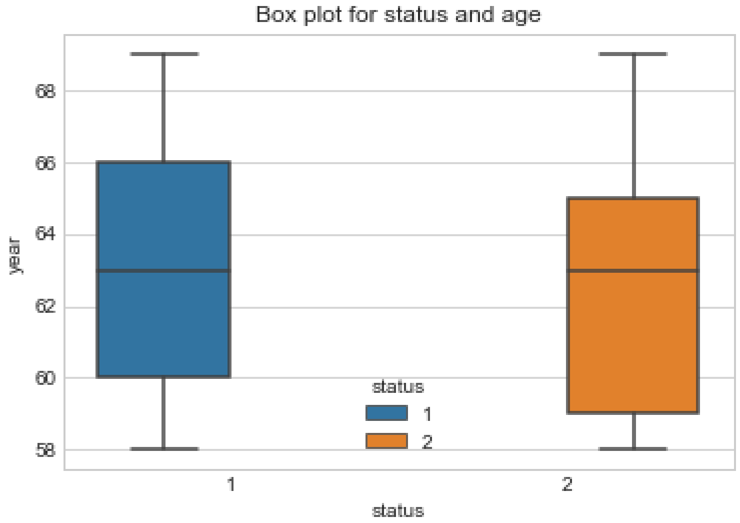


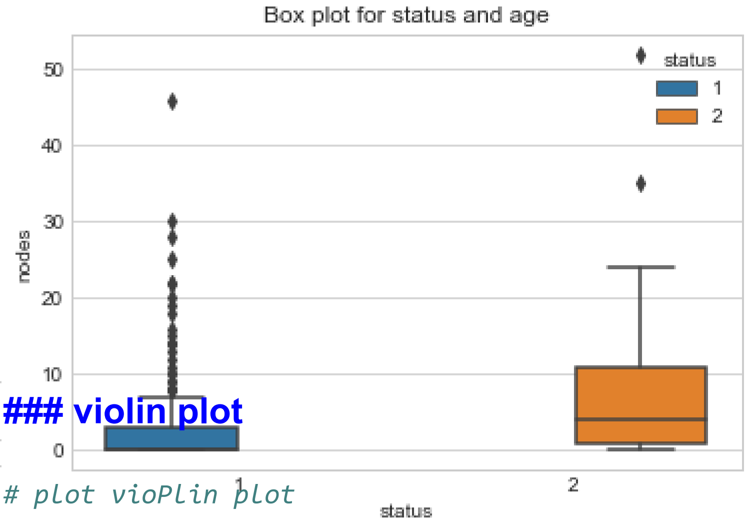
```
Observations:

1) Patient between age 32-36 has survived the operation and pataient aged 77-85
has definitly not survived the operation.
2) as the data for both the case are evenly we cant draw the patient survival
status form the year of operation. Excapt the patient who had surgery between
1961-1965 has high probablity of survival.
3) patient with nodes <22 has has better probability of survival.
```

### Box plot

In [48]:
```
# plot box plot for the same
sns.boxplot(x='status',y='age',hue ="status", data=Haberman).set_title("Box plot 
plt.show()
sns.boxplot(x='status',y='year',hue ="status", data=Haberman).set_title("Box plot
plt.show()
sns.boxplot(x='status',y='nodes',hue ="status", data=Haberman).set_title("Box plo
plt.show()
```



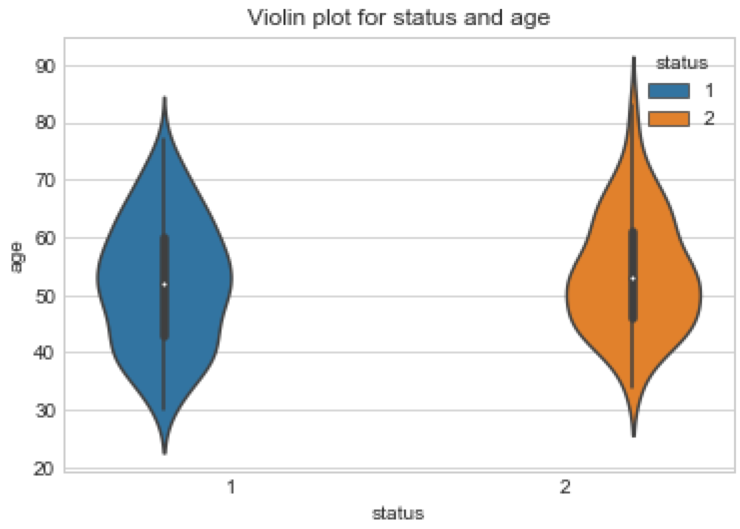




### violin plot

In [47]:

```
# plot vioPlin plot

sns.violinplot(x="status", y="age",hue="status", data=Haberman, size=8)
plt.title("Violin plot for status and age")
plt.show()
sns.violinplot(x="status", y="year",hue="status",data=Haberman, size=8)
plt.title("Violin plot for status and year")
plt.show()
sns.violinplot(x="status", y="nodes",hue="status", data=Haberman, size=8)
plt.title("Violin plot for status and nodes")
plt.show()
```



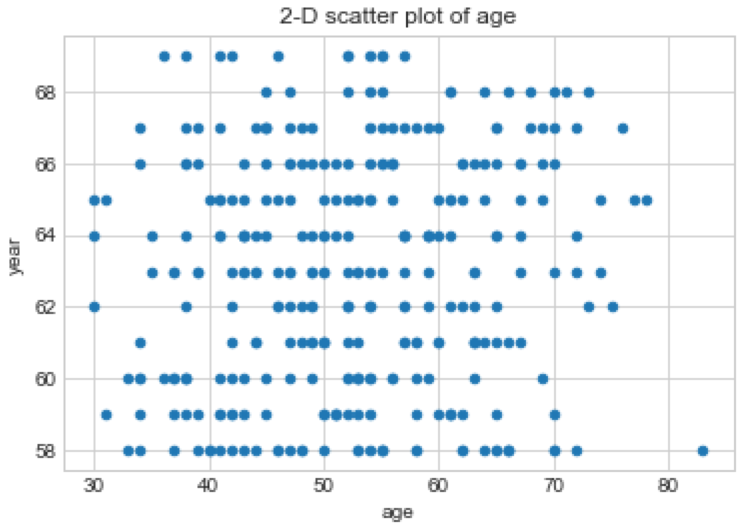
observations:

1) As the data points are overlapping hence no major conclusion could be drawn from this plots
2) The number of node for survival is high or dense from 0-5.

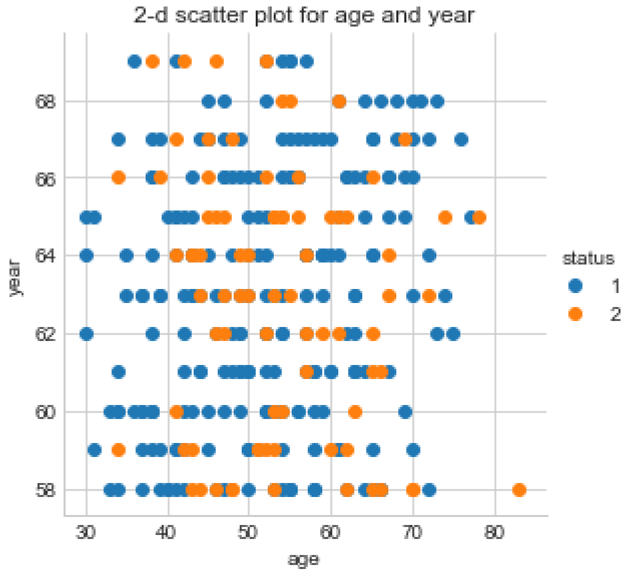
* Perform Bi-variate analysis (scatter plots, pair-plots) to see if combinations of features are useful in classfication.

### Scatter plot

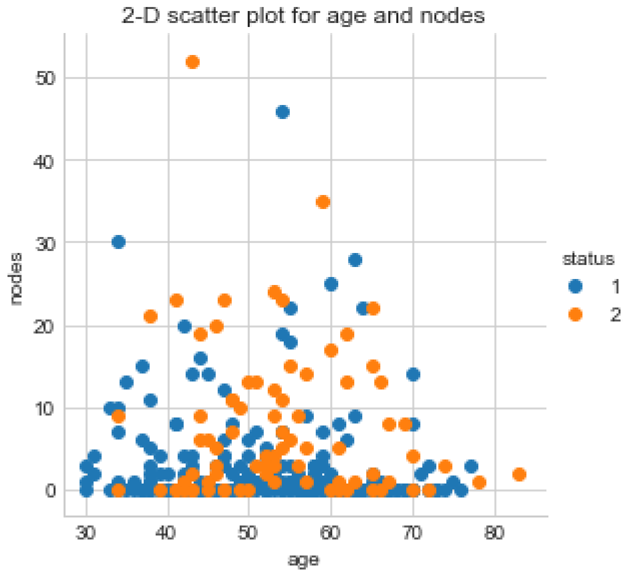
```
In [40]: # 2-D Scatter plot
         Haberman.plot(kind = "scatter", x = "age", y = "year")
         plt.title("2-D scatter plot of age")
         plt.show()
```



```
In [41]: # 2-D Scatter plot with color-coding
         sns.set_style("whitegrid");
         sns.FacetGrid(Haberman, hue="status", size=4) \
            .map(plt.scatter, "age", "year") \
            .add_legend();
         plt.title("2-d scatter plot for age and year")
         plt.show();
```



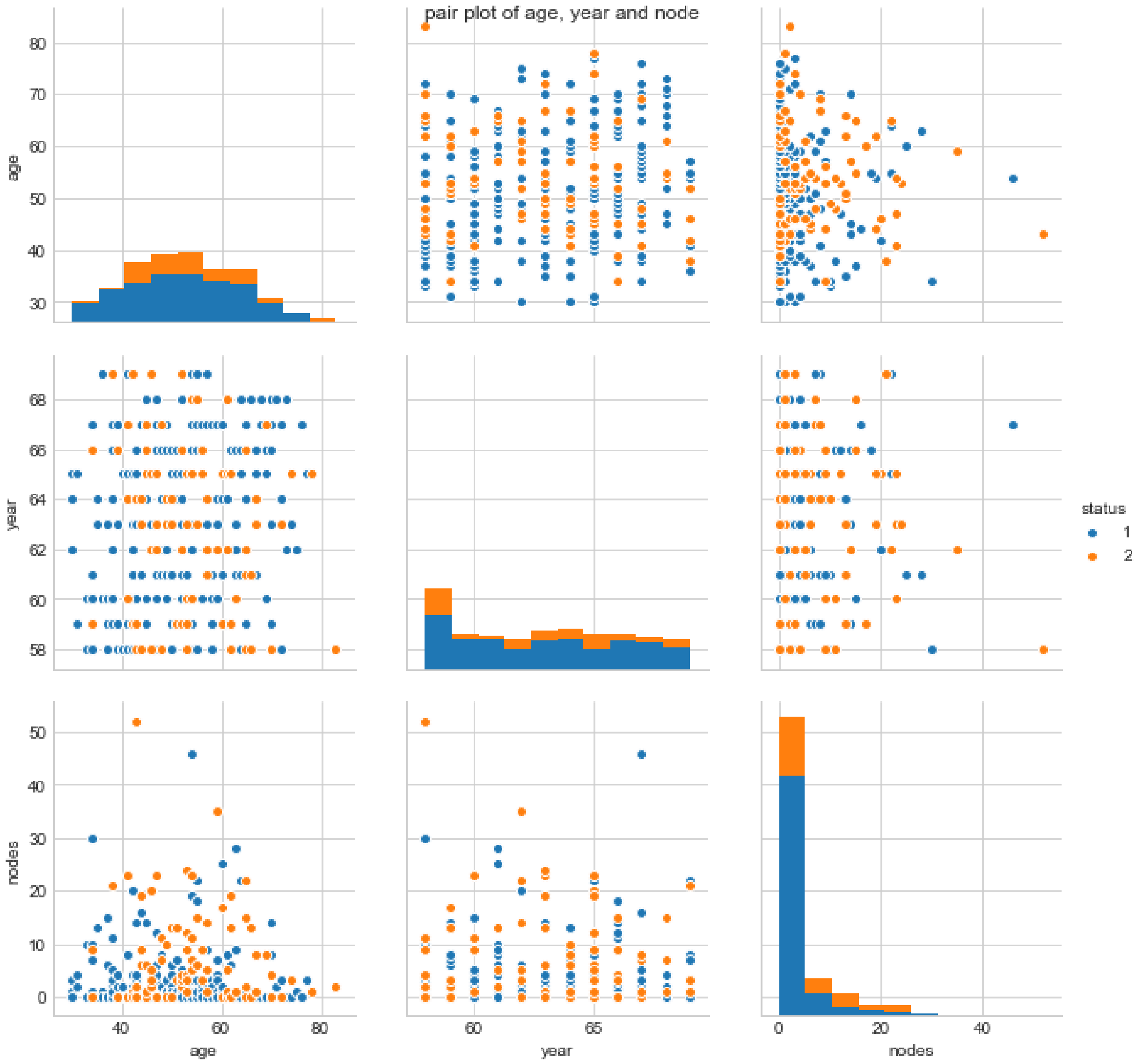
In [42]:
```
sns.set_style("whitegrid")
sns.FacetGrid(Haberman, hue = "status", size = 4).map(plt.scatter, "age", "nodes"
plt.title("2-D scatter plot for age and nodes")
plt.show()
```



```
observations

In the above 2d scatter plot class or status is not linearly seprable
0-5 node person survived and died as well but the died ratio is less than
survive ratio.
```

### Pair plot

In [46]:
```
# pairwise scatter plot: Pair-Plot
sns.set_style("whitegrid")
sns.pairplot(Haberman, hue = "status", vars = ["age", "year", "nodes"], size = 3)
plt.suptitle("pair plot of age, year and node")
plt.show()
```



Observations

The data are highly mixed up, hence we can't find "if-else" conditions to build a simple model to classify the survive status of the patient.