Haberman's Survival Dataset

- Haberman's Survival dataset: [https://www.kaggle.com/gilsousa/habermans-survival-data-set/home]
- This dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer.
- In this Dataset there are 306 rows and 4 column.
- Number of Attributes: 4 (including the class attribute)
- coulmn names- 'age', 'year', 'nodes', 'status'.
- Here 'nodes' mean- Axiliary Node: https://ww5.komen.org/BreastCancer/LymphNodeStatus.html Breast cancer frequently spreads to the axillary (armpit) lymph nodes so, by checking nodes doctor can decide the level of the breast cancer.
- Status is called class attribute or class label. Status or survival time is basically divided in two part:

```
(a)   Survival time is 5 years : 1
(b)   Survival time less than 5 years : 2
```

In [82]:

```
import pandas as pd
#pandas is so powerful python module used in data science.
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
haber = pd.read_csv("haberman.csv")
#csv: stands for "comma seperated value"
#Above pandas is used by pd to read the csv file
```

In [83]:

```
print(haber.shape)
#Shape is used here to print total number of rows and column in the dataset.
```

```
(306, 4)
```

In [84]:

```
print(haber.columns)
#columns is used to find column's name in the given dataset.
#Age, year and nodes are  called as variables.
#status is called as class label
```

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

In [85]:

```
haber["status"].value_counts()
#Value_counts() is used for counting all the groups of same element in given class label
#given below there are two group survival time either 1 or 2 so value_counts() give the total numb
er of people in 1 and 2
```

Out[85]:

```
1    225
2     81
Name: status, dtype: int64
```

In (class label)status number of 1 - 225 and no of 2 - 81. It shows that Given dataset is unbalanced dataset.

Bi-variate Analysis

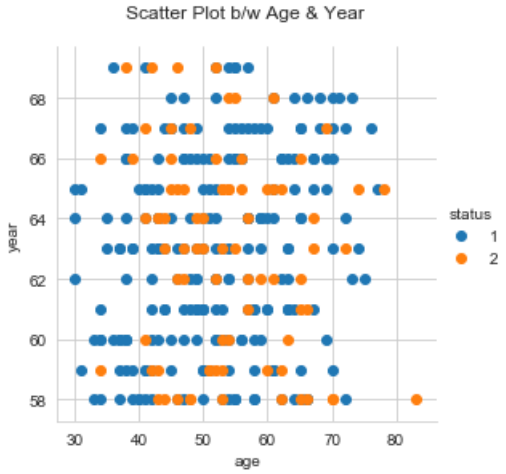
1. This type of data involves two different variables.
2. The analysis of this type of data deals with causes and relationships and the analysis is done to find out the relationship

among the two variables.
3. Reference- https://www.geeksforgeeks.org/univariate-bivariate-and-multivariate-data-and-its-analysis/

Scatter Plot

In [86]:

```
sns.set_style("whitegrid");
s1 = sns.FacetGrid(haber, hue="status", height=4)
s1.map(plt.scatter, "age", "year")
s1.add_legend()
plt.title("Scatter Plot b/w Age & Year \n")
plt.show();
```

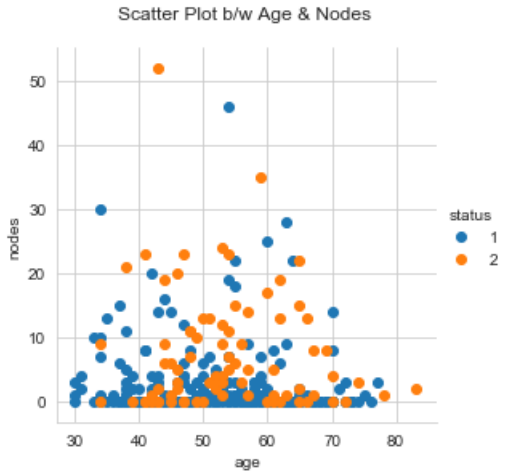


Observations :

1. Our objective is to find the patient who will survive 5 years(denoted by - 1) or less than 5 years(denoted by - 2).
2. In this Graph overlapping is very high so we can not seperate it very efficiently.
3. But we can see from age 30-40 and year 1958 -1964 the survival time of the patient is 5 years.
4. Patient who has less than 5 years survival time(status : 2) is mainly belongs to age 40 to 80 approximately.

In [87]:

```
sns.set_style("whitegrid");
s1 = sns.FacetGrid(haber, hue="status", height=4)
s1.map(plt.scatter, "age", "nodes")
s1.add_legend()
plt.title("Scatter Plot b/w Age & Nodes \n")
plt.show();
```



Observations

1. Approximately Age 30-60 and Nodes 0-10 has the maximum number of patient without any overlapping who has 5 years survival time.
2. Patient ,whom Axiliary Nodes between 0-25 and Age between 35-80, comes under 5 year survival time. 2. This is very complex and overlapped ploting so we can not analyse more.

Pair-plot

In [88]:

```
plt.close();
sns.set_style("whitegrid");
#whitegrid is used for making grid in the graph
sns.pairplot(haber,vars = ["age","year","nodes"],hue = "status")
#hue(color) used for catagorised 'status' by color during ploting of the curve
#vars is used for specific multiple variable ploting
plt.show()
```

Observations

1. Pair plot is the most useful plotting for the analysis. 2. Haberman dataset is unbalanced dataset so it is difficult to analyse easily. 3. I can see in the diagram overlapping in the nodes(age vs nodes & year vs nodes) is less than age and year. 4. Here I'm not able to analyse properly but conclusion is that nodes has less overlapped diagram.

Univariate Analysis

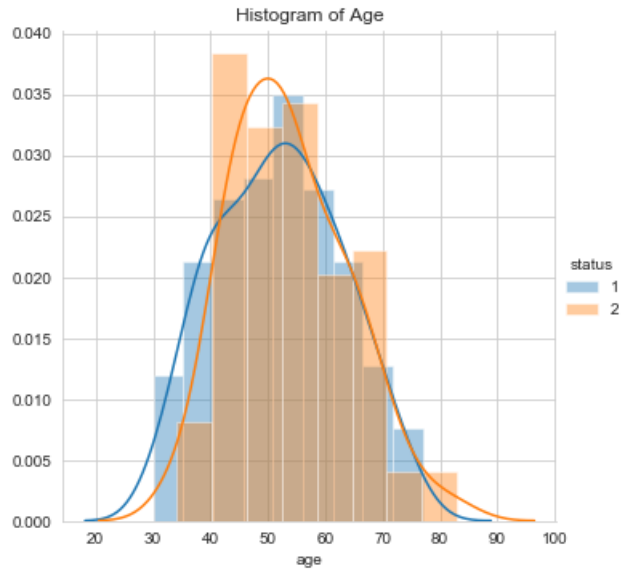
1. This type of data consists of only one variable.
2. The analysis of univariate data is thus the simplest form of analysis since the information deals with only one quantity that changes
3. Reference- https://www.geeksforgeeks.org/univariate-bivariate-and-multivariate-data-and-its-analysis/

Histogram

In [89]:

```
sns.FacetGrid(haber, hue="status", height=5) \
   .map(sns.distplot, "age") \
```

```
    .add_legend();
plt.title("Histogram of Age")
plt.show();
```

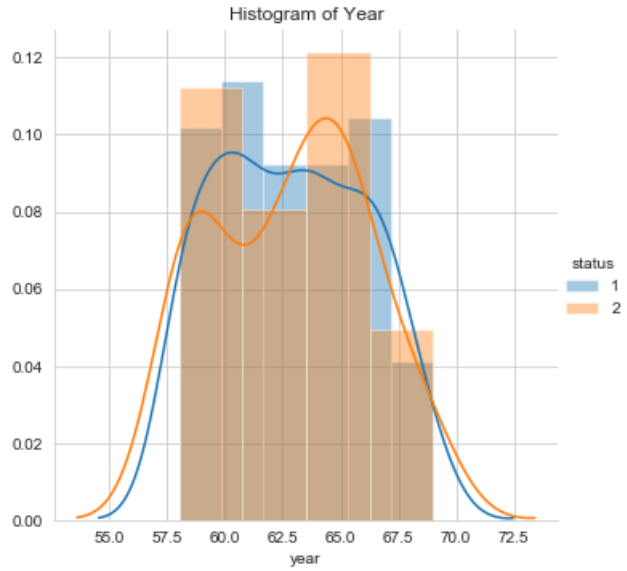


Observations

1. Age from 35 to 75 the overlapping is very high. 2. Any definition is not properly working here to classify the status 1 or 2. 3. Y axis defines the number of point on specific age. 4. From 30 to 40 age if I observes number of patient, whom survival time is 5 year, is less overlapped so We can classify the class label with some error.

In [90]:

```
sns.FacetGrid(haber, hue="status", height=5) \
   .map(sns.distplot, "year") \
   .add_legend();
plt.title("Histogram of Year")
plt.show();
```

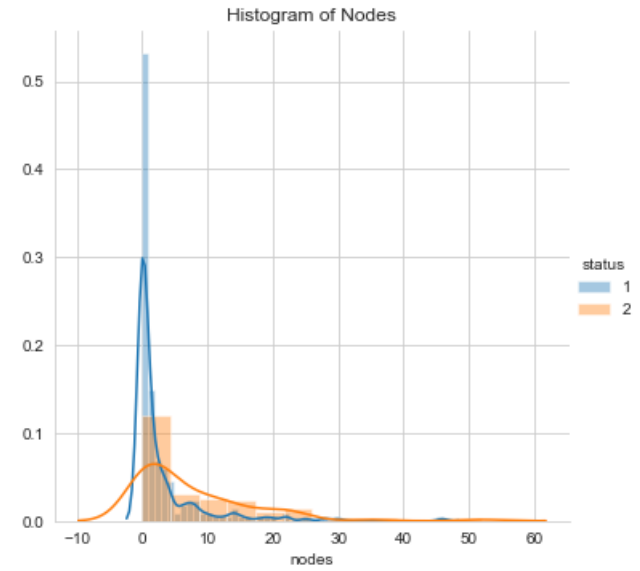


Observations

1. Histogram of age is less overlapped than year histogram. 2. It is more complex to classify the class label.

In [91]:

```
sns.FacetGrid(haber, hue="status", height=5) \
   .map(sns.distplot, "nodes") \
   .add_legend();
plt.title("Histogram of Nodes")
plt.show();
```



Observations

1. Above all histogram except histogram of nodes are the least overlapped diagram to classify the class label. 2. It's very clear plotting to decide the most important variable from given variables to classify the class label. 3. So according to diagram nodes is the most important factor for the classification of the class label. 4. After observing all the histogram of the variables the order of the important variables is year

PDF & CDF

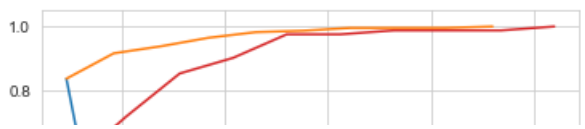
- PDF stands for Probability Density Function.
- CDF stands for Cumulative Density Function.

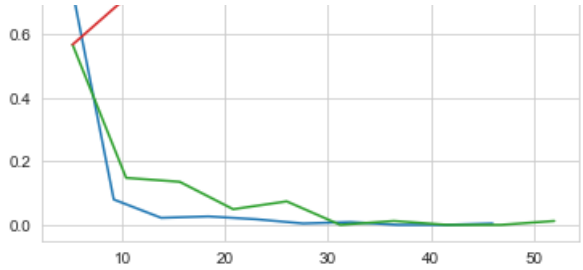
In [92]:

```
#Here I take nodes for the ploting.
haber_1 = haber.loc[haber["status"] == 1 ];
haber_2 = haber.loc[haber["status"]== 2];
counts, bin_edges = np.histogram(haber_1['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(haber_2['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();
```

```
[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. ]
```





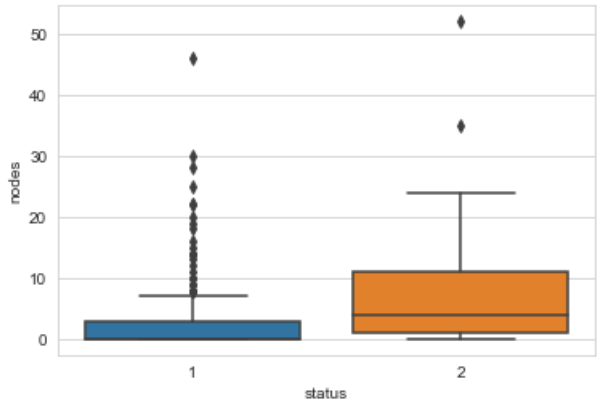
Observations

1. CDF = area under curve of (PDF) 2. Here is difficult to decide the CDF and PDF because both are looking same. 3. Any rules isn't worth for it. 4. I cann't say much after seeing this diagram.

Box plot and Whiskers

In [93]:

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



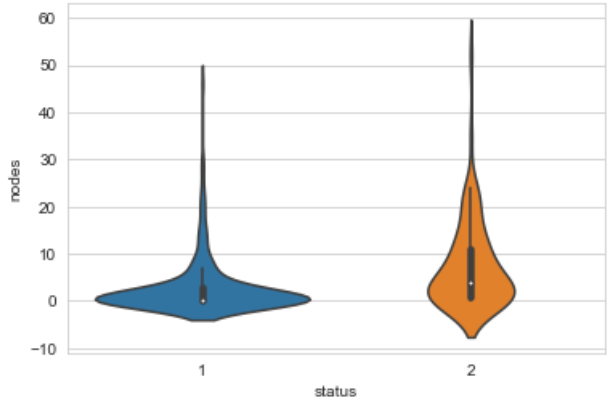
Observations

1. Maximum Whisker : 8 & Minimum Whisker : 0 for status: 1 2. Maximum Whisker : 25 & Minimum Whisker : 0 for status: 2 3. Nodes >=0 && nodes<8 belongs to status: 1 with 80% error. 4. nodes>0 && nodes<25 belongs to status : 2 with 80% error.

Violin plots

In [94]:

```
sns.violinplot(x="status", y="nodes", data=haber, size=8)
plt.show()
```



Observations

1. Violin plot is a combination of histogram and box-plot. 2. Violine plot uses the box-plot in the middle like 25,50 & 7 percentile.

Summary & Conclusion of the haberman's survival Dataset

1. Unbalanced Dataset. 2. Complex to analyse by using one diagram. We have to see many types of plotting to classify the class label. 3. Order of precedence variable for classify the status is- year