Haberman Dataset

The Haberman's survival data set 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 [24]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels import robust
```

In [25]:
```
#Load haberman.csv into a pandas dataFrame.
hb=pd.read_csv('haberman.csv')
hb.head(10)
```

Out[25]:

	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
5	33	58	10	1
6	33	60	0	1
7	34	59	0	2
8	34	66	9	2
9	34	58	30	1

In [26]:
```
hb.shape
```

Out[26]:
```
(306, 4)
```

In [27]:
```
hb.columns
```

Out[27]:
```
Index(['age', 'year', 'nodes', 'status'], dtype='object')
```

In [28]:
```
hb['status'].value_counts()
```

Out[28]:
```
1    225
2     81
Name: status, dtype: int64
```

Attribute Information

There are 306 examples and 4 features in this dataset. Four features have age, year and nodes as independent features while status as dependent feature.

1. Age of patient
2. Patient's year of operation
3. Number of positive nodes
4. Survival status

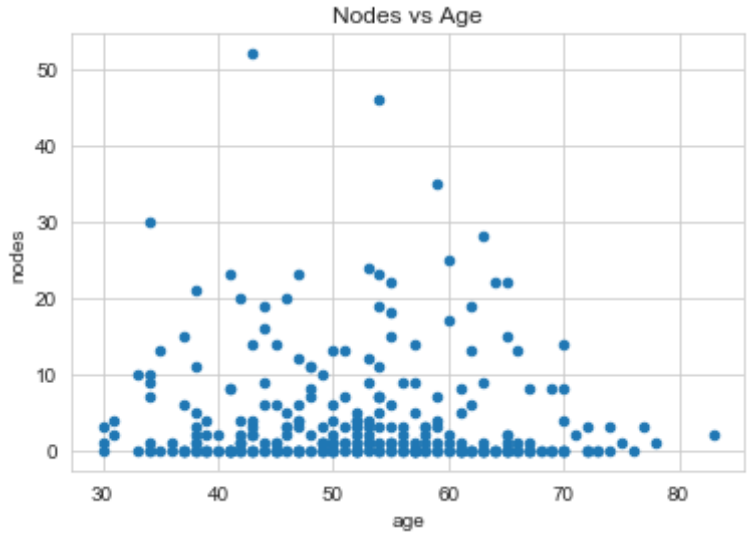
There are two classes(1 and 2). 225 values belong to class 1 while 81 values belong to class 2.

Objective

In this dataset, using features age, year and nodes, we have to classify results in two classes-1 and 2.

In [29]:
```
#Scatter plot of nodes vs age
hb.plot(kind='scatter',x='age',y='nodes')
```

```
plt.title("Nodes vs Age")
plt.show()
```

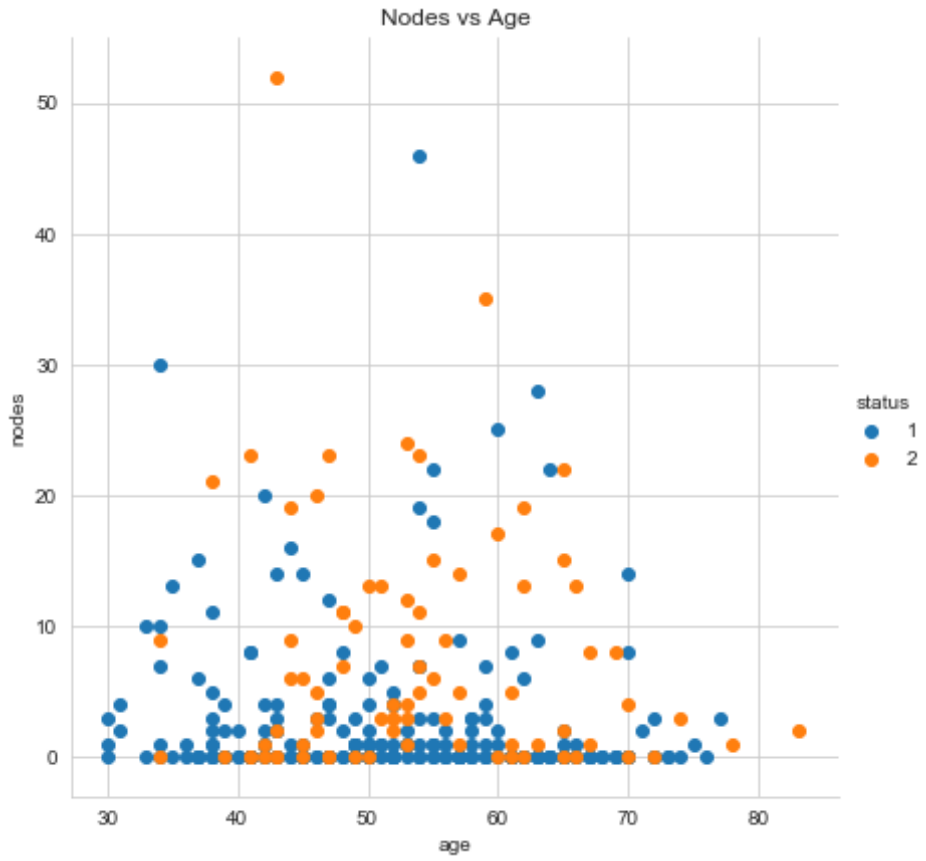


Observation

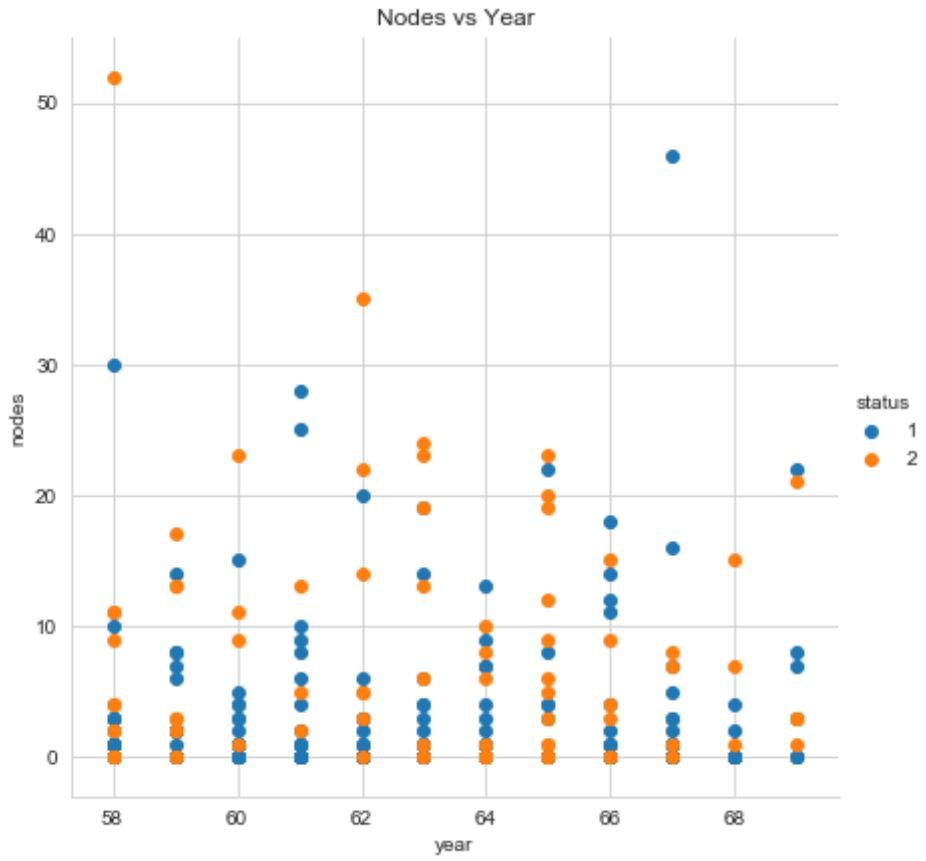
This scatter plot does not give any relevant information regarding our dataset. So we will go for different scatter plots with color coding.

In [30]:
```
#Color-coded scatter plot of year vs age
sns.set_style("whitegrid")
sns.FacetGrid(hb,hue="status",height=6)\
.map(plt.scatter,'age','year')\
.add_legend()
plt.title("Year vs Age")
plt.show()
```

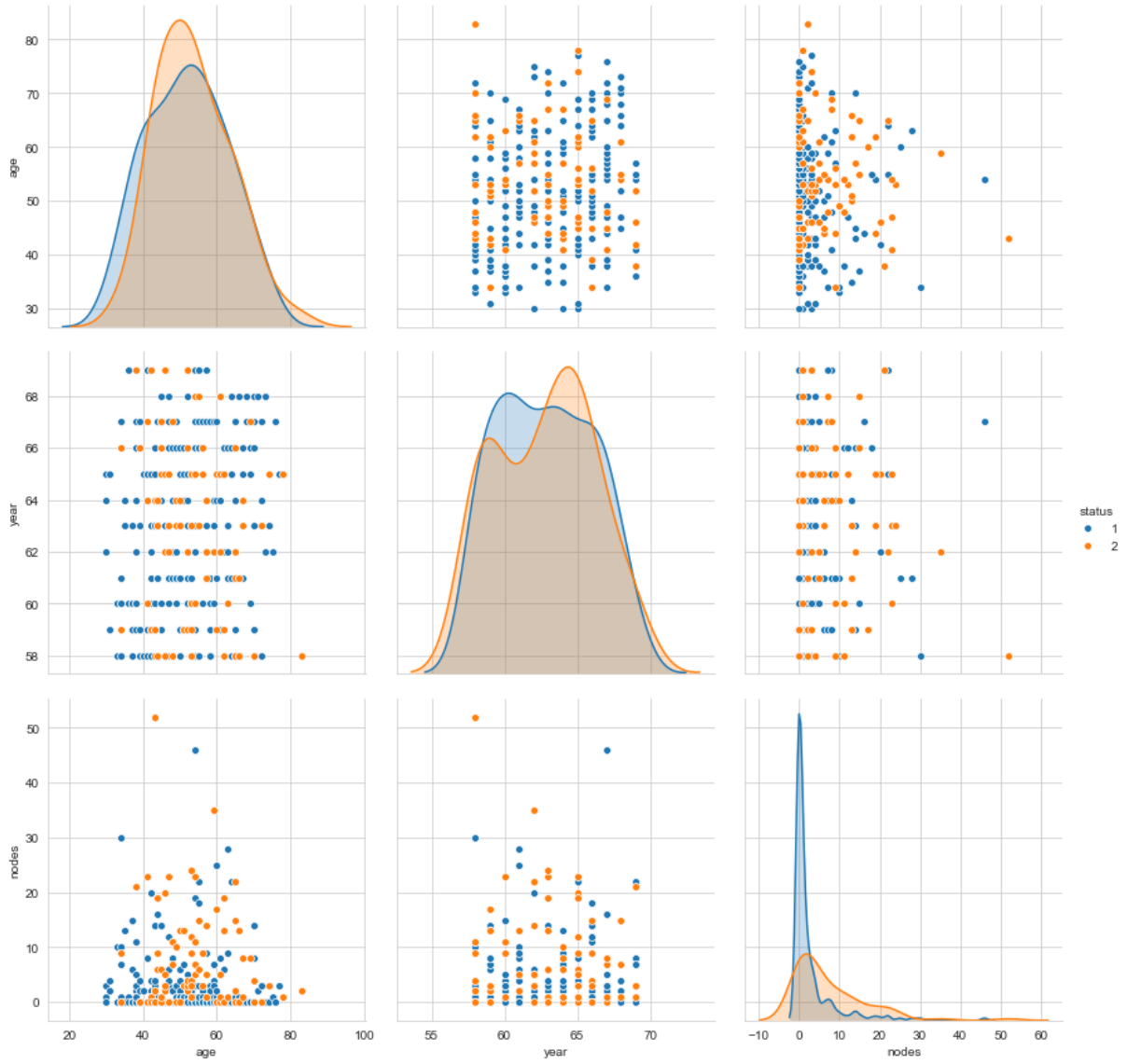
In [31]:
```
#Color-coded scatter plot of nodes vs age
sns.set_style("whitegrid")
sns.FacetGrid(hb,hue="status",height=6)\
.map(plt.scatter,'age','nodes')\
.add_legend()
plt.title("Nodes vs Age")
plt.show()
```



In [32]:
```
#Color-coded scatter plot of nodes vs year
sns.set_style("whitegrid")
sns.FacetGrid(hb,hue="status",height=6)\
.map(plt.scatter,'year','nodes')\
.add_legend()
plt.title("Nodes vs Year")
plt.show()
```

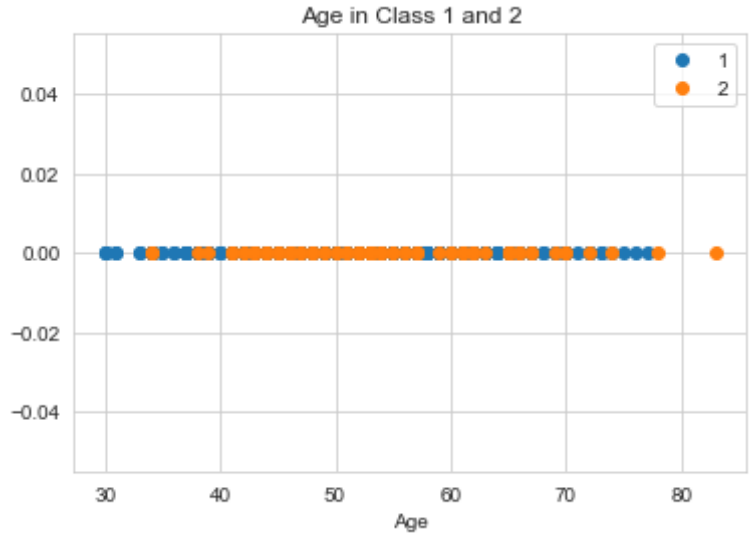


In [33]:
```
#Pair-plot of all the features(age,year,nodes,status)
plt.close();
sns.set_style("whitegrid")
sns.pairplot(hb,hue='status',height=4,vars=["age","year","nodes"])
plt.show()
```

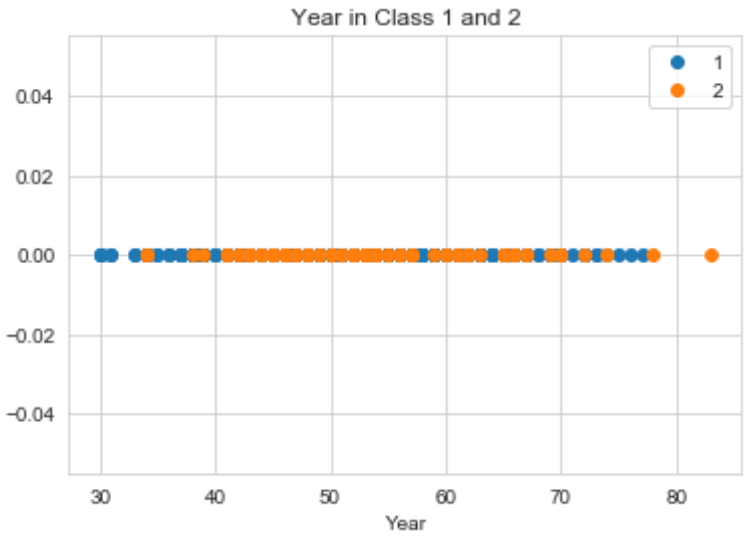


In [34]:
```
#1D plot
one=hb.loc[hb['status']==1]
two=hb.loc[hb['status']==2]
plt.plot(one['age'],np.zeros_like(one['age']),'o')
```

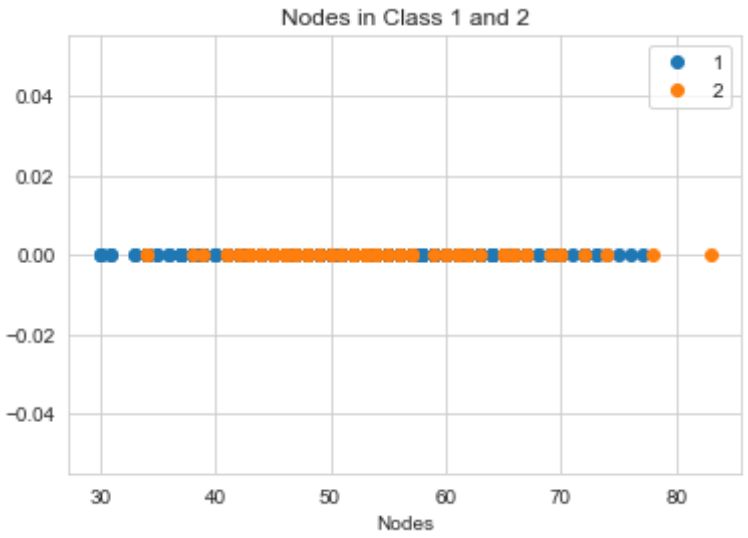
```
plt.plot(two['age'],np.zeros_like(two['age']),'o')
plt.legend("12")
plt.title("Age in Class 1 and 2")
plt.xlabel("Age")
plt.show()
```



In [35]:
```
#1D plot
plt.plot(one['age'],np.zeros_like(one['year']),'o')
plt.plot(two['age'],np.zeros_like(two['year']),'o')
plt.legend("12")
plt.title("Year in Class 1 and 2")
plt.xlabel("Year")
plt.show()
```



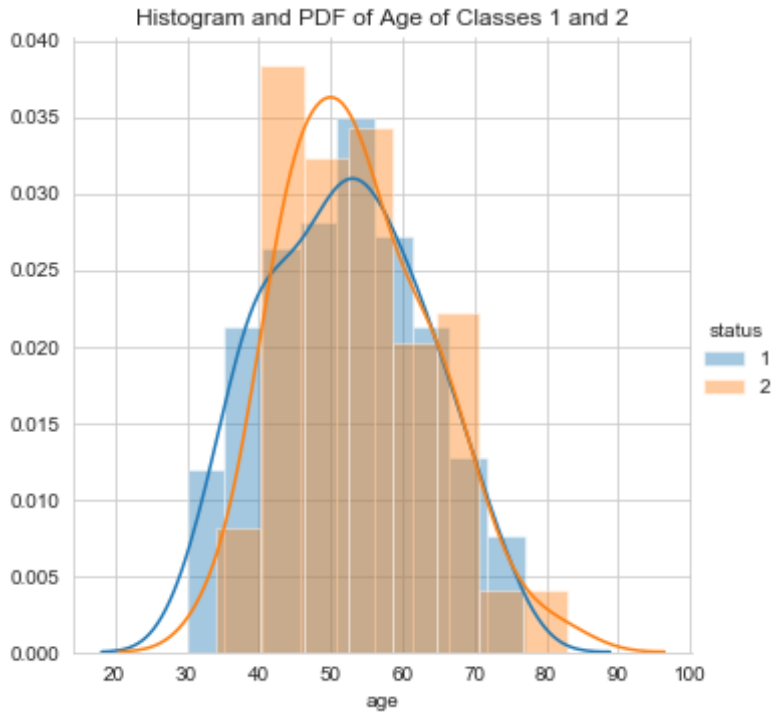
In [36]:
```
#1D plot
plt.plot(one['age'],np.zeros_like(one['nodes']),'o')
plt.plot(two['age'],np.zeros_like(two['nodes']),'o')
plt.legend("12")
plt.title("Nodes in Class 1 and 2")
plt.xlabel("Nodes")
plt.show()
```



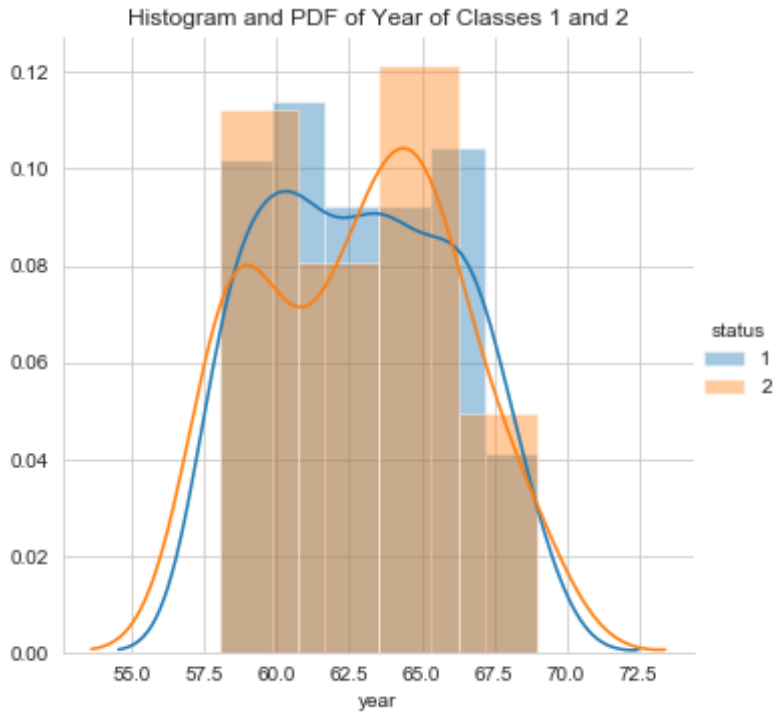
Observation

From the above scatter plots and Pair-plot, we are not able to separate any of the data points. So we cannot classify whether the given observation is of class 1 or class 2. 1-D plots are also not giving any relevant information. So we would go for the histogram plots of features.

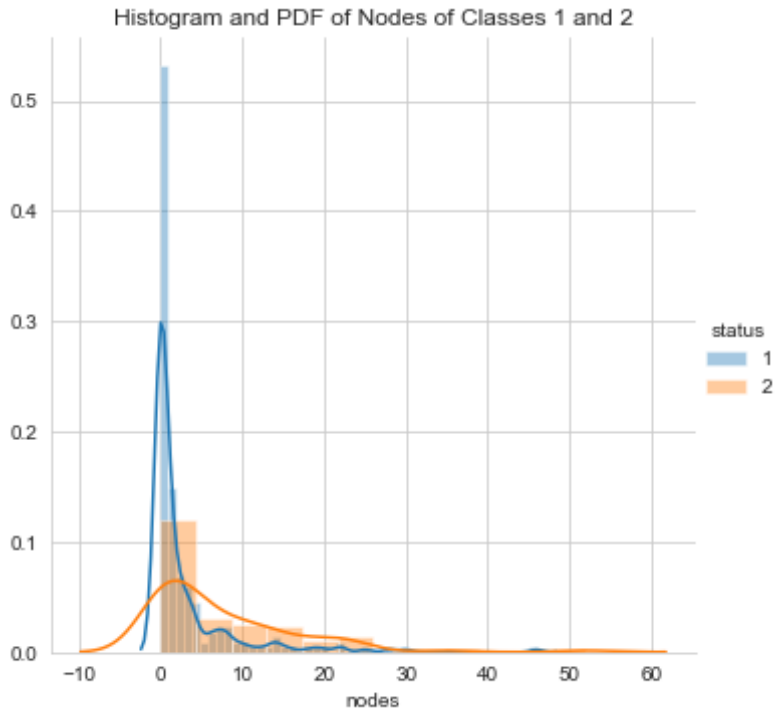
In [37]:
```
#Histogram and PDF of age of Class 1 and Class 2
sns.FacetGrid(hb,hue="status",height=5)\
.map(sns.distplot,'age')\
.add_legend()
plt.title("Histogram and PDF of Age of Classes 1 and 2")
plt.show()
```



In [38]:
```
#Histogram and PDF of year of Class 1 and Class 2
sns.FacetGrid(hb,hue="status",height=5)\
.map(sns.distplot,'year')\
.add_legend()
plt.title("Histogram and PDF of Year of Classes 1 and 2")
plt.show()
```



In [39]:
```
#Histogram and PDF of nodes of Class 1 and Class 2
sns.FacetGrid(hb,hue="status",height=5)\
.map(sns.distplot,'nodes')\
.add_legend()
plt.title("Histogram and PDF of Nodes of Classes 1 and 2")
plt.show()
```

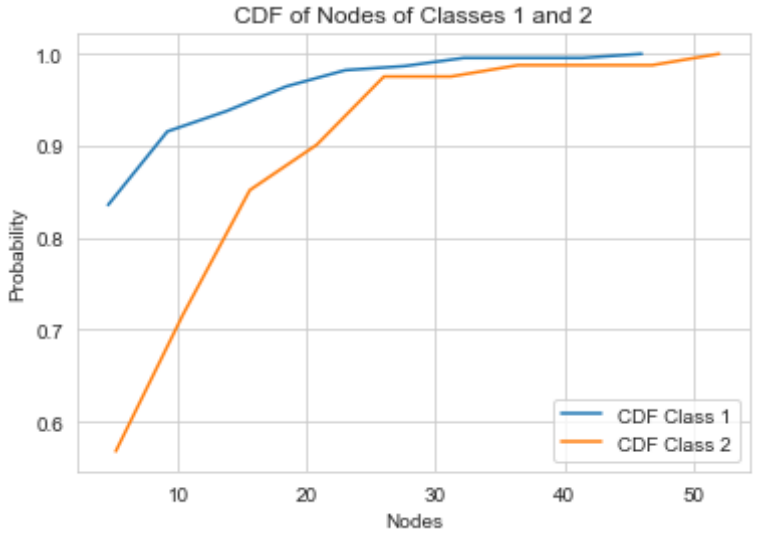


Observation

From the above histograms of age and year,we can see that PDFs of both the classes are overlapped. So we cannot consider any of the two features for classification. But in case of histogram plot of nodes, we can see that even though there is a little bit overlap between PDFs of classes 1 and 2 but there is possibility of making a classifier based on this feature when compared to other two features. So we can make use of this feature only.

In [40]:
```
# CDF plots of nodes of Class 1 and Class 2
counts,bin_edges=np.histogram(one['nodes'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
onecdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 1")
counts,bin_edges=np.histogram(two['nodes'],bins=10,density=True)
```

```
pdf=counts/(sum(counts))
cdf=np.cumsum(pdf)
twocdf=plt.plot(bin_edges[1:],cdf,label="CDF Class 2")
plt.xlabel("Nodes")
plt.ylabel("Probability")
plt.title("CDF of Nodes of Classes 1 and 2")
plt.legend()
plt.show()
```



Observation

From the above CDF curves of nodes we get to know that about 84% of Class 1 people have nodes less than 5 but about 47% of Class 2 people have nodes less than 5. So we have got a classifier boundary at nodes=5.Now we will perform statistical operations on features.

In [41]:
```
print("Mean of age")
print("Class 1=",np.mean(one['age']))
print("Class 2=",np.mean(two['age']))
```

```
Mean of age
```

```
Class 1= 52.01777777777778
Class 2= 53.67901234567901
```

In [42]:
```
print("Mean of year")
print("Class 1=",np.mean(one['year']))
print("Class 2=",np.mean(two['year']))
```

```
Mean of year
Class 1= 62.86222222222222
Class 2= 62.82716049382716
```

In [43]:
```
print("Mean of nodes")
print("Class 1=",np.mean(one['nodes']))
print("Class 2=",np.mean(two['nodes']))
```

```
Mean of nodes
Class 1= 2.7911111111111113
Class 2= 7.45679012345679
```

In [44]:
```
print("Standard Deviation of age")
print("Class 1=",np.std(one['age']))
print("Class 2=",np.std(two['age']))
```

```
Standard Deviation of age
Class 1= 10.98765547510051
Class 2= 10.10418219303131
```

In [45]:
```
print("Standard Deviation of year")
print("Class 1=",np.std(one['year']))
print("Class 2=",np.std(two['year']))
```

```
Standard Deviation of year
Class 1= 3.2157452144021956
Class 2= 3.3214236255207883
```

In [46]:
```
print("Standard Deviation of nodes")
print("Class 1=",np.std(one['nodes']))
print("Class 2=",np.std(two['nodes']))
```

```
Standard Deviation of nodes
Class 1= 5.857258449412131
Class 2= 9.128776076761632
```

In [47]:
```
print("Median of age")
print("Class 1=",np.median(one['age']))
print("Class 2=",np.median(two['age']))
```

```
Median of age
Class 1= 52.0
Class 2= 53.0
```

In [48]:
```
print("Median of year")
print("Class 1=",np.median(one['year']))
print("Class 2=",np.median(two['year']))
```

```
Median of year
Class 1= 63.0
Class 2= 63.0
```

In [49]:
```
print("Median of nodes")
print("Class 1=",np.median(one['nodes']))
print("Class 2=",np.median(two['nodes']))
```

```
Median of nodes
Class 1= 0.0
Class 2= 4.0
```

In [50]:
```
print(np.percentile(one['age'],np.arange(0,100,25)))
print(np.percentile(two['age'],np.arange(0,100,25)))
```

```
[30. 43. 52. 60.]
[34. 46. 53. 61.]
```

In [51]:
```
print(np.percentile(one['year'],np.arange(0,100,25)))
print(np.percentile(two['year'],np.arange(0,100,25)))
```

```
[58. 60. 63. 66.]
```

```
[58. 59. 63. 65.]
```

In [52]:
```
print(np.percentile(one['nodes'],np.arange(0,100,25)))
print(np.percentile(two['nodes'],np.arange(0,100,25)))
```

```
[0. 0. 0. 3.]
[ 0.  1.  4. 11.]
```

In [53]:
```
#Mean Absolute Deviation of age of Class 1 and Class 2
print(robust.mad(one['age']))
print(robust.mad(two['age']))
```

```
13.343419966550417
11.860817748044816
```

In [54]:
```
#Mean Absolute Deviation of year of Class 1 and Class 2
print(robust.mad(one['year']))
print(robust.mad(two['year']))
```

```
4.447806655516806
4.447806655516806
```

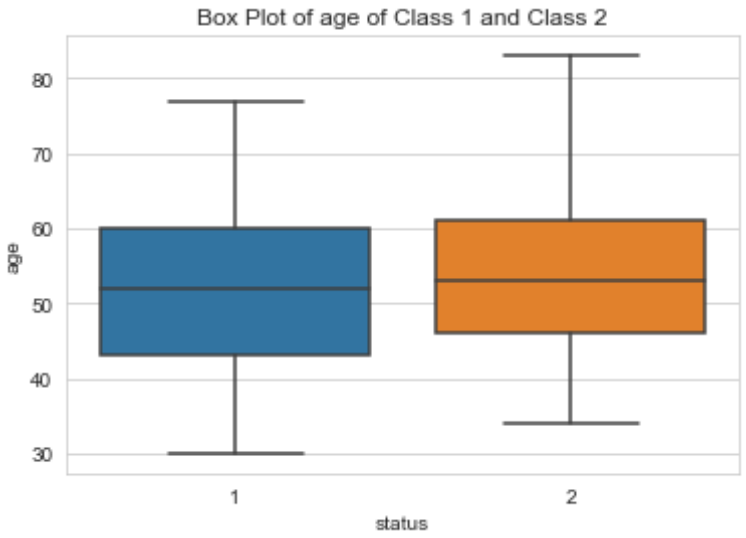
In [55]:
```
#Mean Absolute Deviation of nodes of Class 1 and Class 2
print(robust.mad(one['nodes']))
print(robust.mad(two['nodes']))
```

```
0.0
5.930408874022408
```

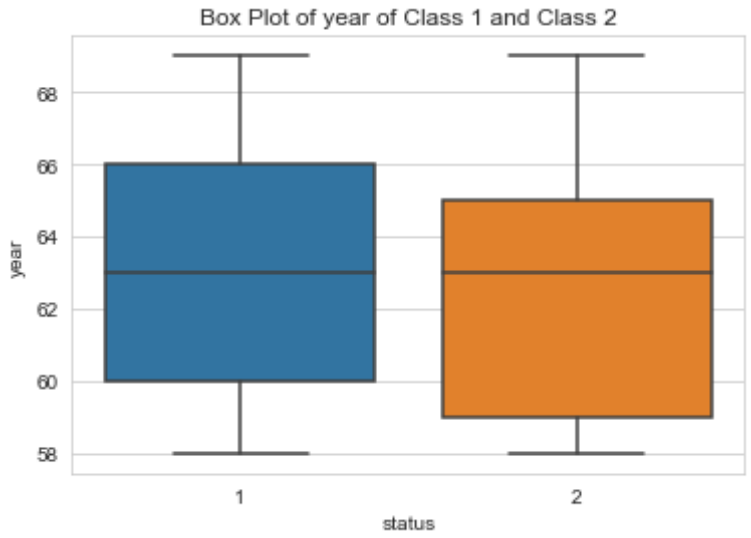
Observation

These statistical operations give us a lot of information. Means, Medians, Standard Deviations and Quantiles of features age and year are almost same. So, we cannot differentiate between them. But in case of feature nodes, they show significant differences. So again it tells we can use feature nodes to make classifier boundary. To differentiate, we will plot boxplots.

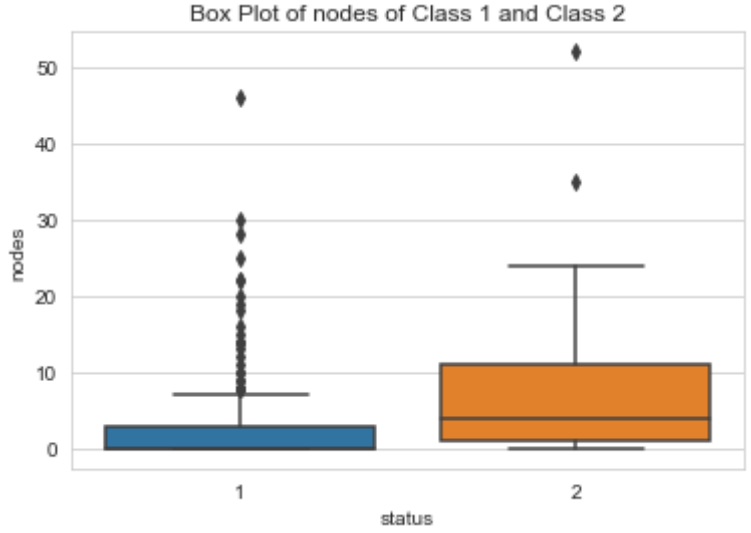
In [56]:
```
#Box Plot of age of Class 1 and Class 2
sns.boxplot(x='status',y='age',data=hb)
plt.title("Box Plot of age of Class 1 and Class 2")
plt.show()
```



In [57]:
```
#Box Plot of year of Class 1 and Class 2
sns.boxplot(x='status',y='year',data=hb)
plt.title("Box Plot of year of Class 1 and Class 2")
plt.show()
```



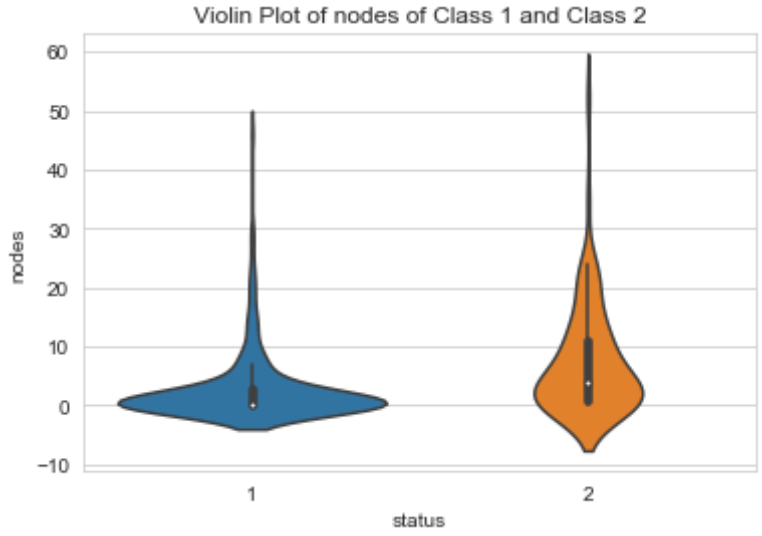
In [58]:
```
#Box Plot of nodes of Class 1 and Class 2
sns.boxplot(x='status',y='nodes',data=hb)
plt.title("Box Plot of nodes of Class 1 and Class 2")
plt.show()
```



In [59]:
```
sns.violinplot(x="status", y="age", data=hb)
plt.title("Violin Plot of age of Class 1 and Class 2")
plt.show()
```

In [60]:
```
sns.violinplot(x="status", y="year", data=hb)
plt.title("Violin Plot of year of Class 1 and Class 2")
plt.show()
```

In [61]:
```
sns.violinplot(x="status", y="nodes", data=hb)
plt.title("Violin Plot of nodes of Class 1 and Class 2")
plt.show()
```



Observation

Again from the boxplots and violin plots, it is clear that featues age and year do not help in classifying as their quantiles do not show very much differences. But from the boxplot of feature nodes, we can make certain inferences as the 75th percentile of class 1 is less than 50th percentile of class 2 which is about 5. This means that 75 percent of nodes in class 1 are less than or equal to 5 while in case of class 2 we have only about 50 percent of nodes which are less than or equal to 5.

Conclusion

From CDF plots, quantiles, boxplots and violin plots, we can conclude that features age and year do not help in making the classification boundary. It is the number of nodes only which somewhat helps us in making the classification boundary for the two classes. If number of nodes is less than or equal to 5 then it belongs to Class 1, else it belongs to Class 2.

In [ ]: