Haberman's Survival Dataset

The 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 [92]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from statsmodels import robust

#Load heberman.csv into a pandas dataFrame.
hm = pd.read_csv("haberman.csv")
hm.head(10)
```

Out[92]:

	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]:
```
# (q)How many data points and features?

print(hm.shape)
```

```
(306, 4)
```

In [27]:
```
# (q) What are the column names in our dataset?
print(hm.columns)
```

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

In [28]:
```
#(Q) How many data points for each class are present?

hm["status"].value_counts()

#Haberman is an unbalanced dataset
```

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

Attribute Information

There are 306 examples and 4 features in this dataset.

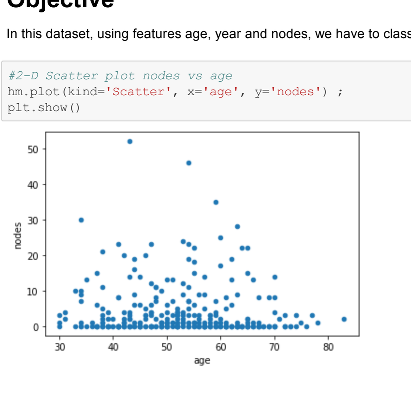
a. Age of patient at time of operation (numerical)\ b.Patient's year of operation (year - 1900, numerical)\ c.Number of positive axillary nodes detected (numerical)\ d.Survival status (class attribute) 1 = the patient survived 5 years or longer ,2 = the patient died within 5 year

Four features have age, year and nodes are independent features while status as dependent feature.

Objective

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

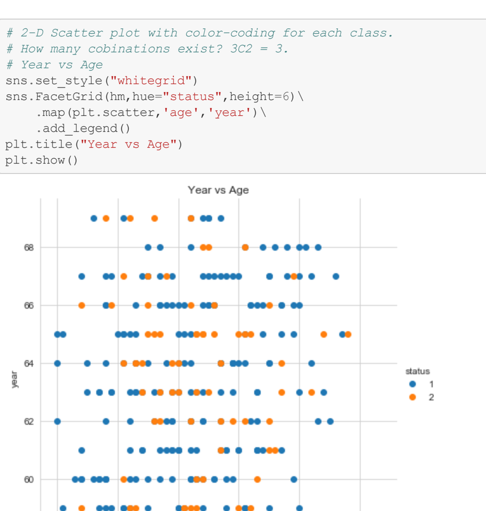
In [30]:
```
#2-D Scatter plot nodes vs age
hm.plot(kind='Scatter', x='age', y='nodes') ;
plt.show()
```



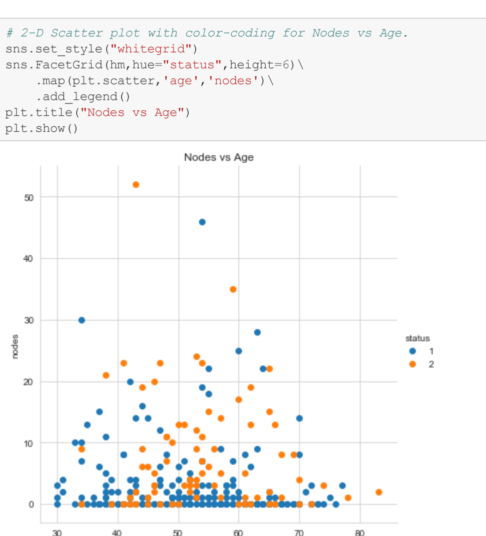
Observation

This scatter plot does not give any relevant information regarding our dataset. \ So we will go fordifferent scatter plots with color coding.We color the points by thier class-label.

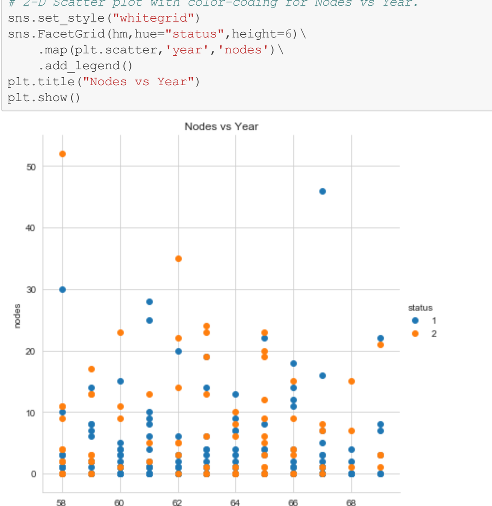
In [44]:
```
# 2-D Scatter plot with color-coding for each class.
# How many cobinations exist? 3C2 = 3.
# Year vs Age
sns.set_style("whitegrid")
sns.FacetGrid(hm,hue="status",height=6)\
    .map(plt.scatter,'age','year')\
    .add_legend()
plt.title("Year vs Age")
plt.show()
```



In [45]:
```
# 2-D Scatter plot with color-coding for Nodes vs Age.
sns.set_style("whitegrid")
sns.FacetGrid(hm,hue="status",height=6)\
    .map(plt.scatter,'age','nodes')\
    .add_legend()
plt.title("Nodes vs Age")
plt.show()
```



In [46]:
```
# 2-D Scatter plot with color-coding for Nodes vs Year.
sns.set_style("whitegrid")
sns.FacetGrid(hm,hue="status",height=6)\
    .map(plt.scatter,'year','nodes')\
    .add_legend()
plt.title("Nodes vs Year")
plt.show()
```



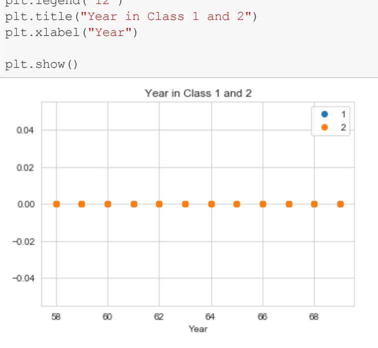
Oberservation

1.orange and blue data points cannot be easily seperated in each case.\ 2.Seperating class 1 from class 2 is much harder as they have considerable overlap.

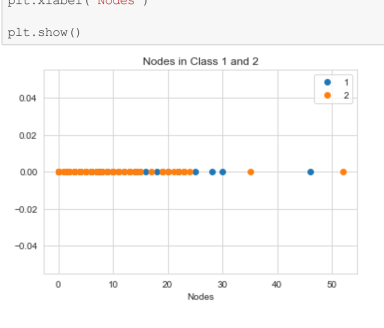
In [49]:
```
# pairwise scatter plot: Pair-Plot

plt.close();
sns.set_style("whitegrid");
sns.pairplot(hm, hue="status", height=3,vars=["age","year","nodes"]);
plt.show()
# NOTE: the diagnol elements are PDFs for each feature. PDFs are expalined below.
```

In [55]:
```
#1-D scatter plot of age
import numpy as np
one = hm.loc[hm["status"] == 1];
two = hm.loc[hm["status"] == 2];

#print(iris_setosa["petal_length"])
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 [58]:
```
#1-D scatter plot of year
import numpy as np
one = hm.loc[hm["status"] == 1];
two = hm.loc[hm["status"] == 2];

#print(iris_setosa["petal_length"])
plt.plot(one["year"], np.zeros_like(one['year']), 'o')
plt.plot(two["year"], np.zeros_like(two['year']), 'o')
plt.legend("12")
plt.title("Year in Class 1 and 2")
plt.xlabel("Year")

plt.show()
```



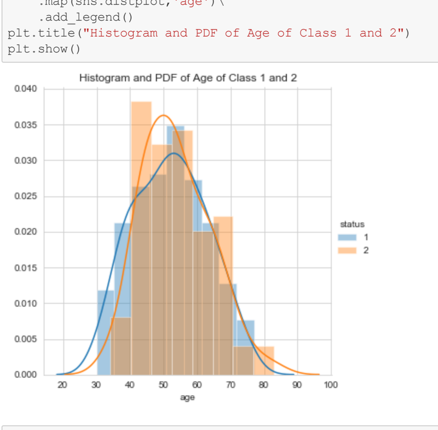
In [59]:
```
#1-D scatter plot of nodes
import numpy as np
one = hm.loc[hm["status"] == 1];
two = hm.loc[hm["status"] == 2];

#print(iris_setosa["petal_length"])
plt.plot(one["nodes"], np.zeros_like(one['nodes']), 'o')
plt.plot(two["nodes"], np.zeros_like(two['nodes']), 'o')
plt.legend("12")
plt.title("Nodes in Class 1 and 2")
plt.xlabel("Nodes")

plt.show()
```



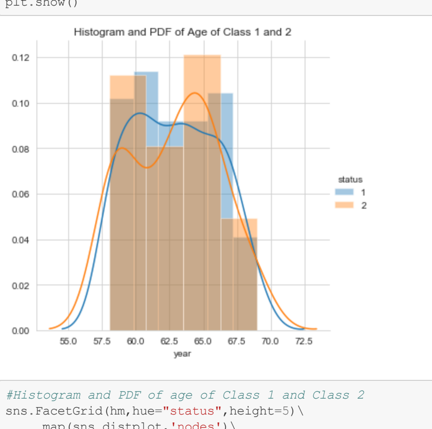
Observation

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

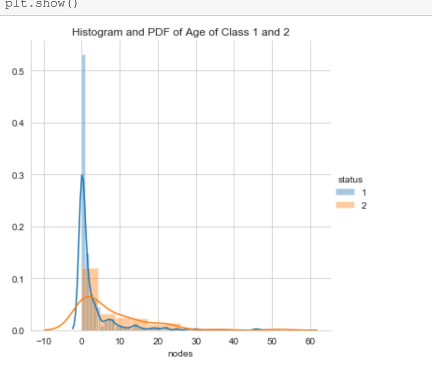
In [62]:
```
#Histogram and PDF of age of Class 1 and Class 2
sns.FacetGrid(hm,hue="status",height=5)\
    .map(sns.distplot,'age')\
    .add_legend()
plt.title("Histogram and PDF of Age of Class 1 and 2")
plt.show()
```



In [63]:
```
#Histogram and PDF of age of Class 1 and Class 2
sns.FacetGrid(hm,hue="status",height=5)\
    .map(sns.distplot,'year')\
    .add_legend()
plt.title("Histogram and PDF of Age of Class 1 and 2")
plt.show()
```



In [64]:
```
#Histogram and PDF of age of Class 1 and Class 2
sns.FacetGrid(hm,hue="status",height=5)\
    .map(sns.distplot,'nodes')\
    .add_legend()
plt.title("Histogram and PDF of Age of Class 1 and 2")
plt.show()
```

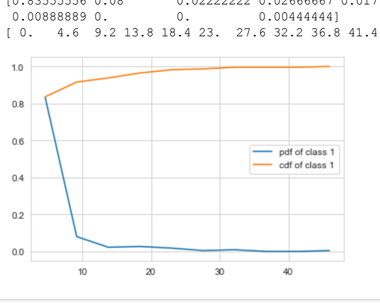


Observation

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.

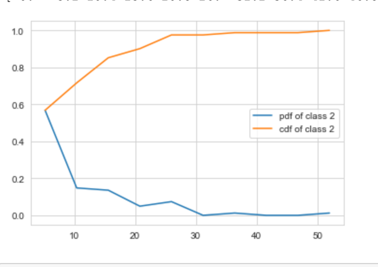
In [72]:
```
#CDF and PDF plots of Nodes of class 1
counts, bin_edges = np.histogram(one['nodes'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges);
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,label="pdf of class 1");
plt.plot(bin_edges[1:], cdf,label="cdf of class 1")
plt.legend()

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. ]
```

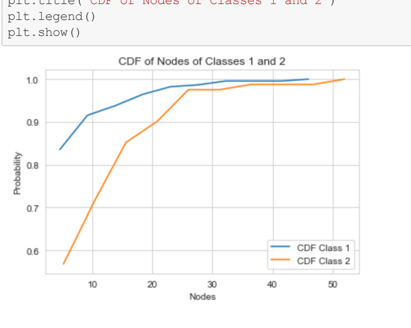


In [73]:
```
#CDF and PDF plots of Nodes of class 2
counts, bin_edges = np.histogram(two['nodes'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges);
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,label="pdf of class 2");
plt.plot(bin_edges[1:], cdf,label="cdf of class 2")
plt.legend()

plt.show();
```

```
[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. ]
```



In [76]:
```
# 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 and\ about 45% of Class 2 people have nodes less than 5.\ So we have got aclassifier boundary at nodes=5.Now we will perform statistical operations on feature.

Means:

In [80]:
```
print("Mean of age")
print("Class 1=",np.mean(one['age']))
print("Class 2=",np.mean(two['age']))
```

```
Mean of age
Class 1= 52.017777777777778
Class 2= 53.67901234567901
```

In [81]:
```
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 [82]:
```
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
```

Std-dev:

In [85]:
```
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 [83]:
```
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 [84]:
```
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
```

Medians:

In [86]:
```
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 [87]:
```
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 [88]:
```
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
```

Quantile

In [89]:
```
print("\nQuantiles:")
print(np.percentile(one['age'],np.arange(0,100,25)))
print(np.percentile(two['age'],np.arange(0,100,25)))
```

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

In [90]:
```
print("\nQuantiles:")
print(np.percentile(one['year'],np.arange(0,100,25)))
print(np.percentile(two['year'],np.arange(0,100,25)))
```

```
Quantiles:
[58. 60. 63. 66.]
[58. 59. 63. 65.]
```

In [91]:
```
print("\nQuantiles:")
print(np.percentile(one['nodes'],np.arange(0,100,25)))
print(np.percentile(two['nodes'],np.arange(0,100,25)))
```

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

MAD

In [93]:
```
print ("\nMedian Absolute Deviation")
print(robust.mad(one["age"]))
print(robust.mad(two["age"]))
```

```
Median Absolute Deviation
13.343419966550417
11.860817748044816
```

In [94]:
```
print ("\nMedian Absolute Deviation")
print(robust.mad(one["year"]))
print(robust.mad(two["year"]))
```

```
Median Absolute Deviation
4.447806655516806
4.447806655516806
```

In [95]:
```
print ("\nMedian Absolute Deviation")
print(robust.mad(one["nodes"]))
print(robust.mad(two["nodes"]))
```

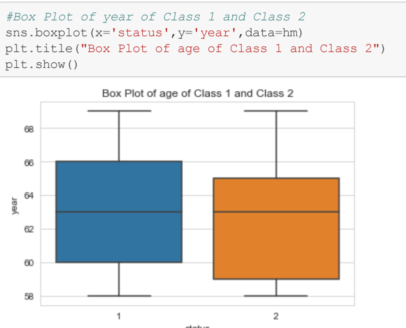
```
Median Absolute Deviation
0.0
5.930408874022408
```

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 nodes, they show significant differences. So again it tells we can use nodes to make classifier boundary. \ To differentiate, we will plot boxplot.

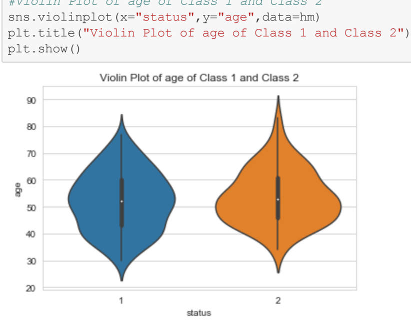
In [96]:
```
#Box Plot of age of Class 1 and Class 2
sns.boxplot(x='status',y='age',data=hm)
plt.title("Box Plot of age of Class 1 and Class 2")
plt.show()
```

In [104]:
```
#Box Plot of year of Class 1 and Class 2
sns.boxplot(x='status',y='year',data=hm)
plt.title("Box Plot of age of Class 1 and Class 2")
plt.show()
```

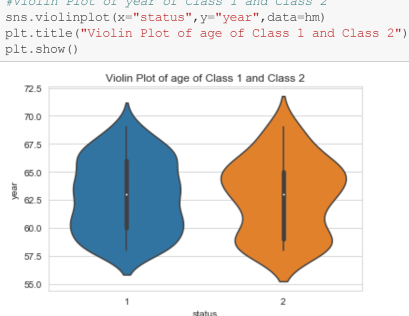


In [105]:
```
#Box Plot of nodes of Class 1 and Class 2
sns.boxplot(x='status',y='nodes',data=hm)
plt.title("Box Plot of age of Class 1 and Class 2")
plt.show()
```

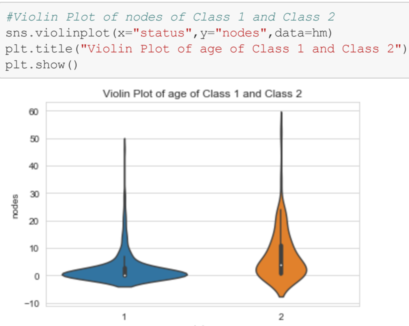
In [106]:
```
#Violin Plot of age of Class 1 and Class 2
sns.violinplot(x="status",y="age",data=hm)
plt.title("Violin Plot of age of Class 1 and Class 2")
plt.show()
```



In [107]:
```
#Violin Plot of year of Class 1 and Class 2
sns.violinplot(x="status",y="year",data=hm)
plt.title("Violin Plot of age of Class 1 and Class 2")
plt.show()
```



In [108]:
```
#Violin Plot of nodes of Class 1 and Class 2
sns.violinplot(x="status",y="nodes",data=hm)
plt.title("Violin Plot of age of Class 1 and Class 2")
plt.show()
```



Obeservation

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 lessthan or equal to 5 while in case of class 2 we have only about 50 percent of nodes which are lessthan or equal to 5.

Conclusion