In [37]:
```python
import pandas as pa
import matplotlib.pyplot as mp
import seaborn as sea
import numpy as np
import warnings

#loading haberman in to paandas

haberman = pa.read_csv('haberman.csv')
warnings.filterwarnings("ignore")
```

In [38]:
```python
# by printing the shape we can get the total number of points

print("shape of data set =",haberman.shape )

#So number of points are 305 and as there are 4 coloums in which 3 are features and 1 is class label
```

```
shape of data set = (305, 4)
```

In [39]:
```python
#to find the colums that we can find in the dataset
print("columns =",haberman.columns)
```

```
columns = Index(['Age', 'Op_year', 'axil_nodes_det', 'Surv_status'], dtype='object')
```

In [40]:
```python
#In the data set they gave numbers instead of all these age op_year columns ... i changed the csv fi
le
#As there are 4 columns  'Age', 'Op_year', 'axil_nodes_det' are features and class attribute is 'Sur
v_status'
#To know how many data points for each class are present
haberman['Surv_status'].value_counts()
#1 means the person is alive for more than 5 year after cancer
#2 means that the person is not alive after 5 years after cancer
#It is an imbalanced data set because class with 1 are a lot more than that of 2.
```

Out[40]:
```
1    224
2     81
Name: Surv_status, dtype: int64
```

Observations

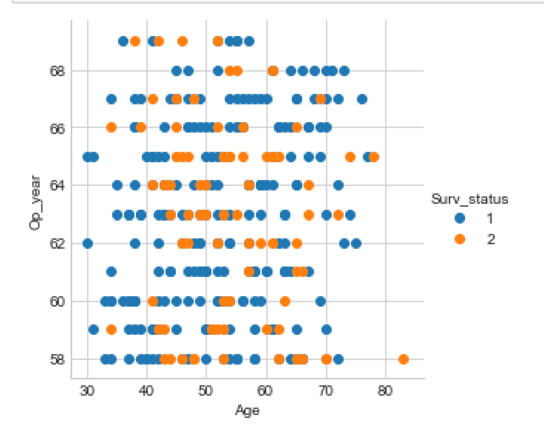
1)Dataset has 4 columns and 305 rows 2)Dataset has features as 'Age', 'Op_year','axil_nodes_det' and class label as 'Surv_status 3)There are totally 224 points of Surv_status as class1 (people are alive after 5 years) and 81 points as class-2 (people who are not alive after 5 years)

In [41]:
```python
#Scatter plot
haberman.plot(kind='scatter',x='Age',y='Op_year')
mp.title('2-D scatter plot')
mp.show();
```

observations

1)As everything is in same color we are unable to classify

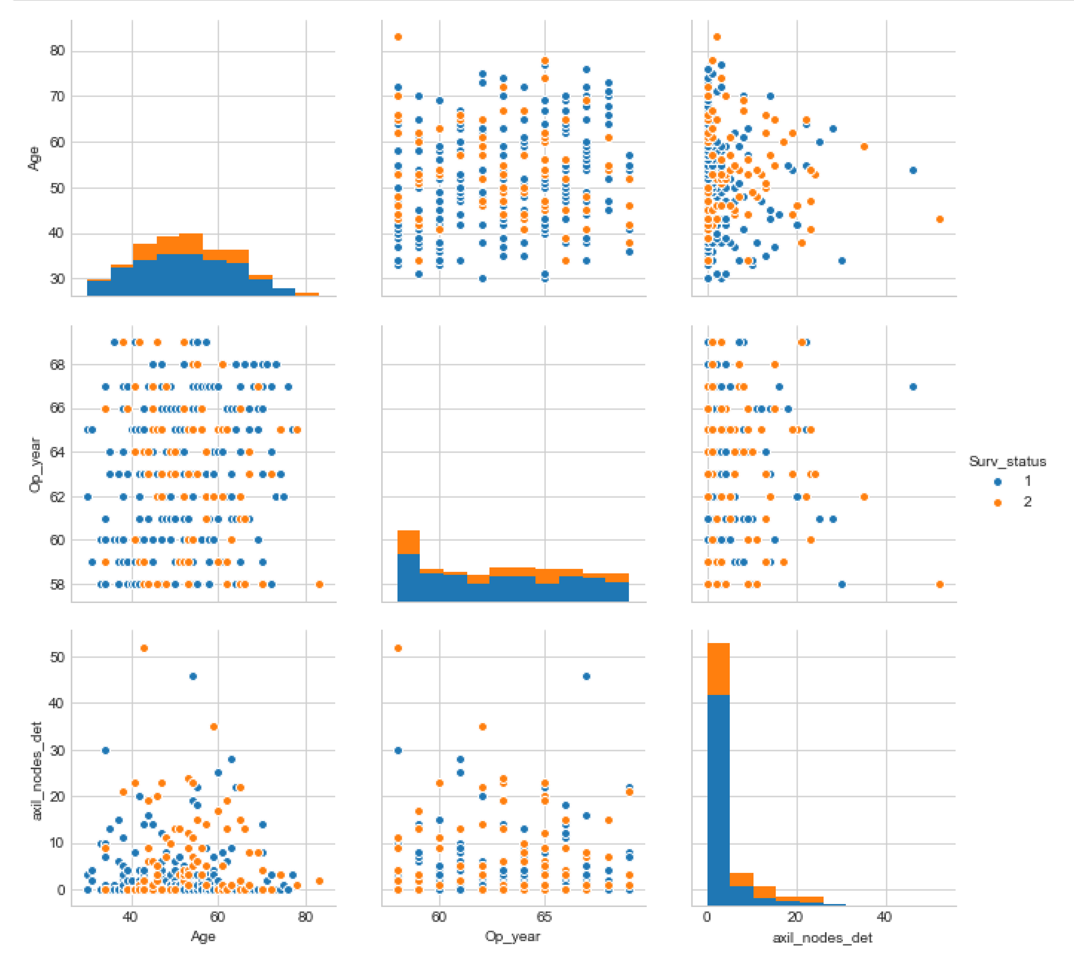
In [42]:
```python
#using seaborn we can give colors
sea.set_style("whitegrid");
sea.FacetGrid(haberman, hue='Surv_status',size=4)\
   .map(mp.scatter, 'Age','Op_year')\
   .add_legend();
mp.show();
```



Observations

1)Here the both class labels are in different colors but, using these 2 attributes classification is very difficult as the both colors are mixed

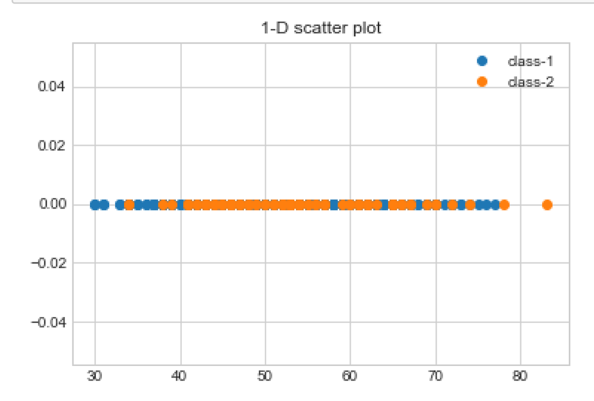
In [43]:
```python
#It is really very dangerous man....!!! how to distinguish now??
#So that let us use the pair plot so that we can find the attributes which effectively differentiate
mp.close();
sea.set_style("whitegrid");
sea.pairplot(haberman, hue="Surv_status",size=3,vars=['Age', 'Op_year', 'axil_nodes_det']);
mp.show();
```



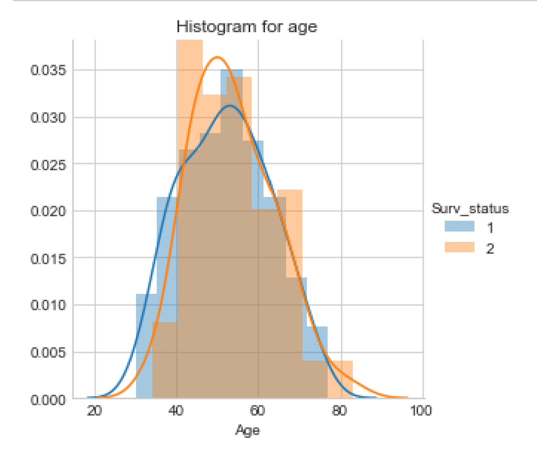
Observations

1)Here in scatter pair-plot the one with the attributes (Op_year,axil_nodes_det) can somewhat good in classifying compared to others.

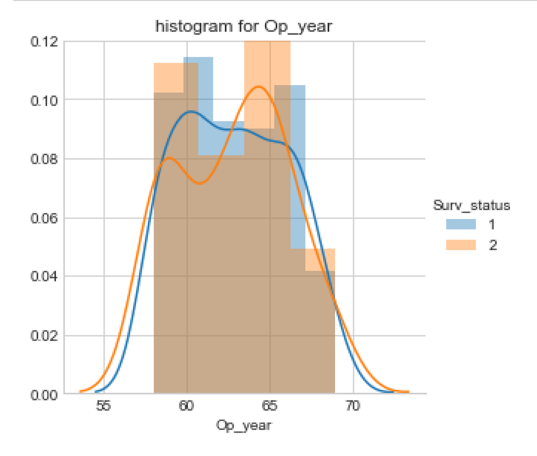
In [44]:
```python
# try with 1-D scatter plot
haberman_1 = haberman.loc[haberman["Surv_status"]==1];
haberman_2 = haberman.loc[haberman["Surv_status"]==2];

mp.title('1-D scatter plot')
mp.plot(haberman_1["Age"], np.zeros_like(haberman_1['Age']), 'o',label='class-1')
mp.plot(haberman_2["Age"], np.zeros_like(haberman_2['Age']),'o',label='class-2')
mp.legend();
mp.show();
```



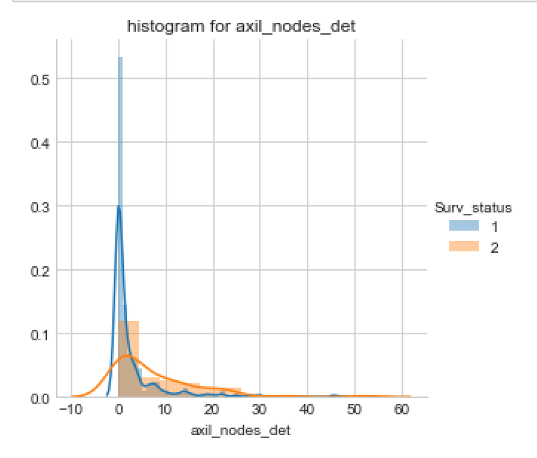
In [45]:
```python
#Now let us draw histograms for this
sea.FacetGrid(haberman, hue='Surv_status' , size=4)\
   .map(sea.distplot, "Age")\
   .add_legend();
mp.title("Histogram for age")
mp.show();
```



In [46]:
```python
#histogram for Op_year
sea.FacetGrid(haberman, hue='Surv_status' , size=4)\
   .map(sea.distplot, "Op_year")\
   .add_legend();
mp.title("histogram for Op_year")
mp.show();
```



In [47]:
```python
#histogram for axil_nodes_det
sea.FacetGrid(haberman, hue='Surv_status' , size=4)\
   .map(sea.distplot, "axil_nodes_det")\
   .add_legend();
mp.title("histogram for axil_nodes_det")
mp.show();
```

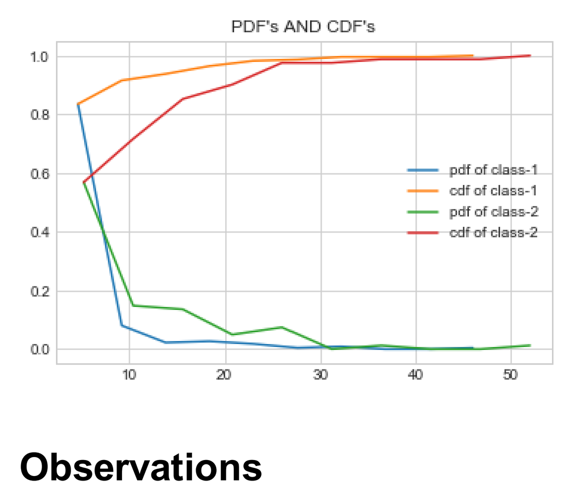


Observations

1)Here , in 1-D scatter plot we cannot be able to count how many points are there 2)So, when we go with histograms we can find that histograms with the attributes 'Age' and 'Op_date' are very complicated to classify compared with axil_nodes_det.

In [48]:
```python
#I think this is really a complicated data and nothing can distinguish it properly and axil_nodes is
 somewhat better so let us draw cdf and pdf for this
#PDF-Probability Distribution function
#CDF-Cumilative Distribution function
mp.title("PDF's AND CDF's")
counts, bin_edges = np.histogram(haberman_1["axil_nodes_det"], bins=10,density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges);
cdf = np.cumsum(pdf)
mp.plot(bin_edges[1:],pdf,label='pdf of class-1');
mp.plot(bin_edges[1:], cdf,label='cdf of class-1');
mp.legend()

#2
counts, bin_edges = np.histogram(haberman_2["axil_nodes_det"], bins=10,density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges);
cdf = np.cumsum(pdf)
mp.plot(bin_edges[1:],pdf,label='pdf of class-2');
mp.plot(bin_edges[1:], cdf,label='cdf of class-2');
mp.legend()

mp.show();
```

```
[0.83482143 0.08035714 0.02232143 0.02678571 0.01785714 0.00446429
 0.00892857 0.         0.         0.00446429]
[ 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. ]
```

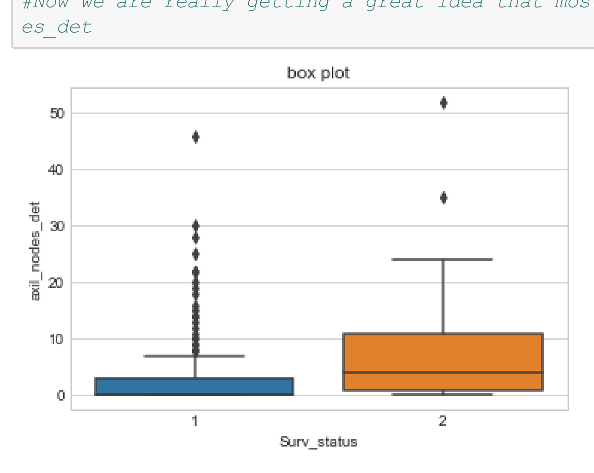


Observations

1)PDF"""============================== explaining 'bin edges and counts ============================== bin edges : [1. 11. 21. 31. 41. 51.] counts per each bin : [5 5 7 2 1] ============================== explaining 'density=True' parameter ============================== manual calculated densities for each bin [0.025 0.025 0.035 0.01 0.005] bin edges : [1. 11. 21. 31. 41. 51.] counts per each bin using density=True: [0.025 0.025 0.035 0.01 0.005] ============================== explaining counts/sum(counts) ============================== bin edges : [1. 11. 21. 31. 41. 51.] counts per each bin using density=True: [0.25 0.25 0.35 0.1 0.05] I am really confused about bin_edges and counts and atlast i got clear by seeing this explanation""" Pdf's is like percentage , we will have counts with density, it is like division of count to the sum of the counts.

2)CDf is percentile , that is as said in the lecture differentiation of cdf is pdf and integration of pdf is cdf

In [49]:
```python
#Box plot
sea.boxplot(x='Surv_status',y='axil_nodes_det',data=haberman)
mp.title("box plot")
mp.show()
#Now we are really getting a great Idea that most of them of status 2 are present with more axil_nod
es_det
```



Observation

1)This will let us know that the box is b/w 25% to 75% of class-1 and class-2 are with what axil_node_det, that is majority of the class-1 and class-2 class labels have what axil_nodes_det.

In [50]:
```python
#Violin plot
sea.violinplot(x='Surv_status',y='axil_nodes_det',data = haberman , size = 8)
mp.title("violin plot")
mp.show()
```