Assignment on Exploratory data analysis of Haberman dataset

Objective-

To perform the analysis of cancer patients data explored from kaggle.com who had undergone breast cancer surgery between the years 1958 and 1970.on the basis of patients who had survied less than 5 year & more than 5 years after lymph node surgery. on the survival of patients.

```
 For data analysis its necesary to explored the data so that we
 can distingusih its features and can perform various data opera
tions on it.here we explored tha data from kaggle.com(https://ww
w.kaggle.com/gilsousa/habermans-survival-data-set/version/1)
The dataset contains cases from a study that was conducted betwe
en 1958 and 1970 at the University of Chicago's Billings Hospita
l on the survival of patients who had undergone surgery for brea
st cancer.
```

Here we have four attributes and 306 number of instances. Information about attributes based on kaggle data.

1-Age of the cancer patient at the time of operation.

2-Patients year of operation(year at which patint undergone surgery i.e.-1958 to 1969)

3-No of positive axillary nodes detected on pateint

4-Survival status

i) 1= the patient who had survied 5 years or longer.

ii) 2= the patient who had survied less than 5 years or died within 5 years.

here 3 feature attribute and 1 class attribute.there is no mising attribute values.so based on this data we are going to start to perform the operations. here we are going to use python language for data analysis primarily because of the fantastic ecosystem of data-centric Python packages and rich in python language which is Here we are going to use pandas,numpy,seaborn,matplotlib to perform the various operations and plots.

In [4]:
```
##Importing Diffrent Pyhon library
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
```

In [6]:
```
haberman = pd.read_csv("haberman.csv")
```

here we import data from csv file format

In [7]:
```
## For No of Rows & Columns in data
print(haberman.shape)
```

```
(306, 4)
```

above code shows the shape of data

i.e. number of rows & columns present in the Haberman cancer data

In [48]:
```
##For Label of Column
print(haberman.columns)
```

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

here we can see the labels of column.

our first column is age,second column is year,third column is nodes & fourth column is staus which is output label dtype means data type which is object type

In [49]:
```
haberman['status'].value_counts()
```

Out[49]:
```
1    225
2     81
Name: status, dtype: int64
```

Observation:

Here we can see dtype of class attribute that is no of (data)patients comes in catageroy -1 and no of data points comes under categery 2.we can no of data points in other feature attribute but our analysis is on survival of patients . so here we are getting 225 data points comes under catery 1 & 81 data points comes under categey 2 i.e 225 patients survied more than 5 years after surgery and 81 patients survied less than 5 years.This is imbalanced dataset

In [50]:
```
haberman.head(6)
```

Out[50]:

	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

Observation: It is just preview to how our data contains.used to return top 6 rows of a dataset.

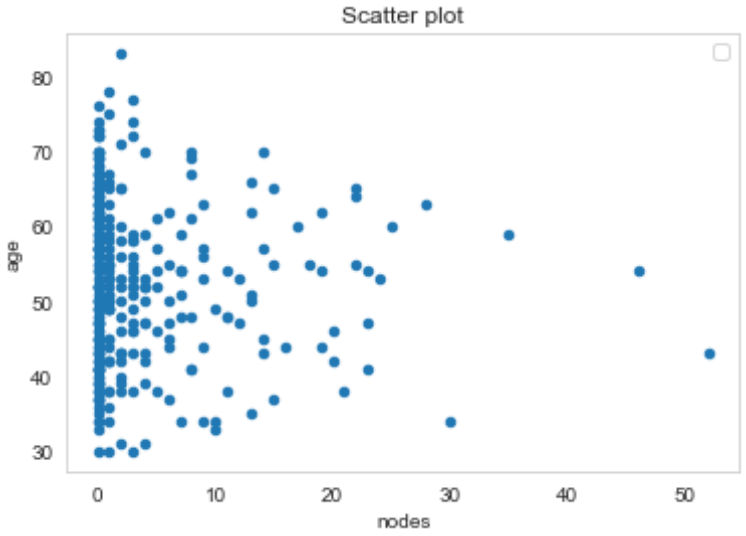
In [20]:
```
haberman.describe()
```

Out[20]:

	age	year	nodes	status
count	306.000000	306.000000	306.000000	306.000000
mean	52.457516	62.852941	4.026144	1.264706
std	10.803452	3.249405	7.189654	0.441899
min	30.000000	58.000000	0.000000	1.000000
25%	44.000000	60.000000	0.000000	1.000000
50%	52.000000	63.000000	1.000000	1.000000
75%	60.750000	65.750000	4.000000	2.000000
max	83.000000	69.000000	52.000000	2.000000

In [18]:
```
##2-D Scatter plot
haberman.plot(x='nodes',y='age',kind = 'scatter');
plt.title("Scatter plot")
plt.grid()
plt.legend()
plt.show()
```

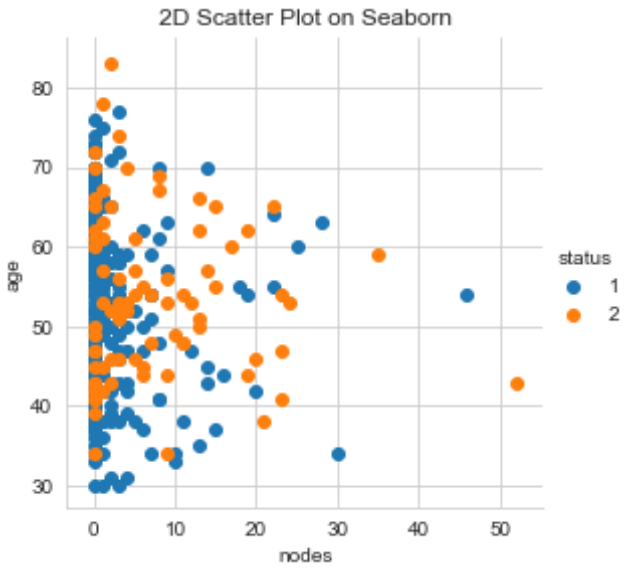
```
No handles with labels found to put in legend.
```



Observation:

above code gives me the scatter plot w.r.t the Nodes on x-axis and Age on y-axis.Our matplotlib library functions grid and show help me to plot data in grid.These is the 2D scatter plot here we are scatterinng the points putting it on the graph.From above plot we can't distingush between data because of same color & overlapping manner by which we can get misconclude To avoid this we importing seaborn packages,Seaborn gives us seaborn gives us very nice tools to plot this things.

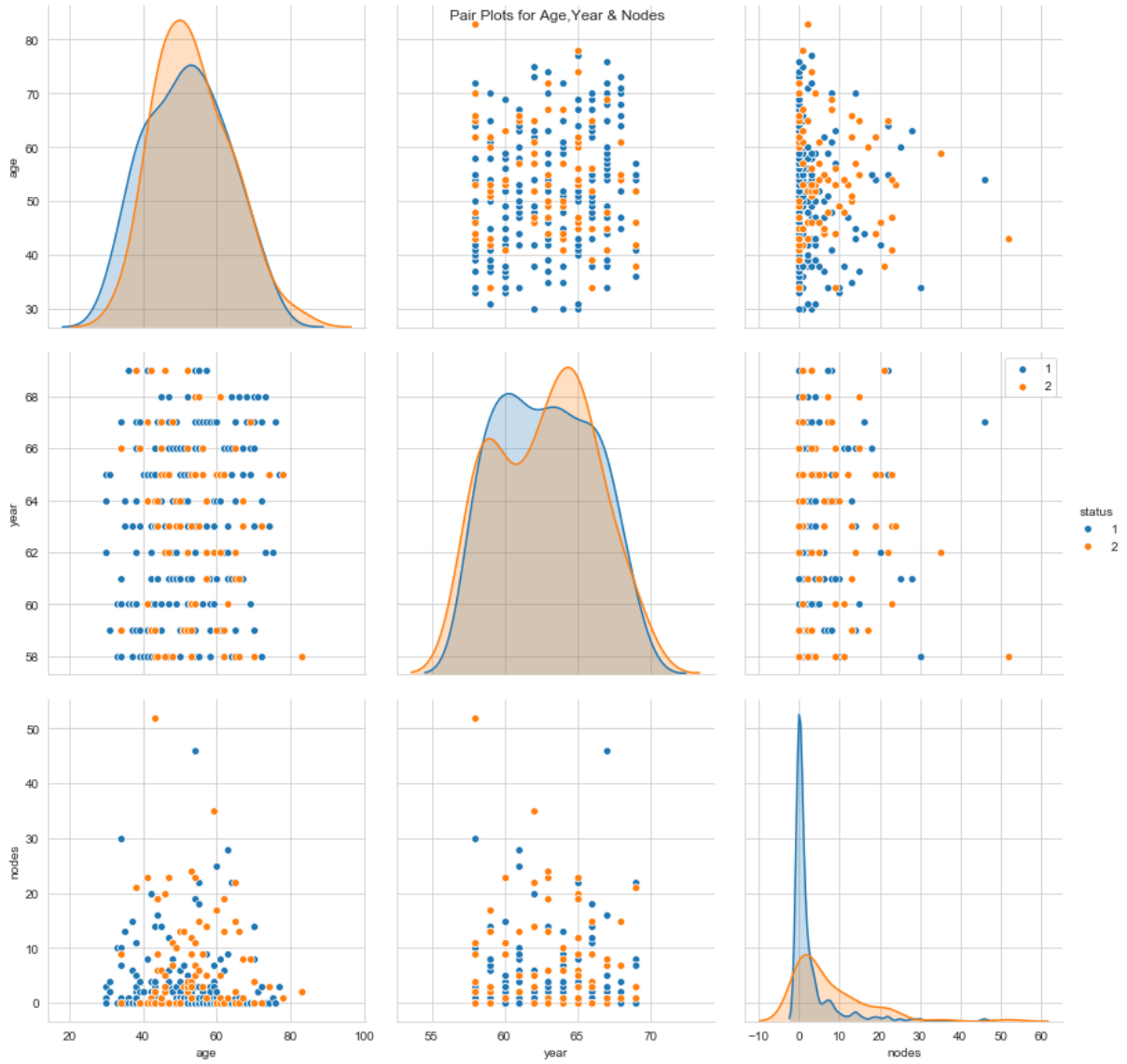
In [8]:
```
sns.set_style('whitegrid');
sns.FacetGrid(haberman,hue='status',size=4)\
   .map(plt.scatter,'nodes','age')\
   .add_legend();
plt.title('2D Scatter Plot on Seaborn')
plt.show()
```



Observation:

Here 'hue=status'by which we can color our dataset based on there values,plotting nodes on X axis & age on Y axis.label says what color gives us what class label. Here blue dot represents long survied status & orange dot represents short survival status.The plot is not linearly seaparable there is considerable overlap between the staus 1 & status 2 to get better plot we wll try for 3D scatter plot

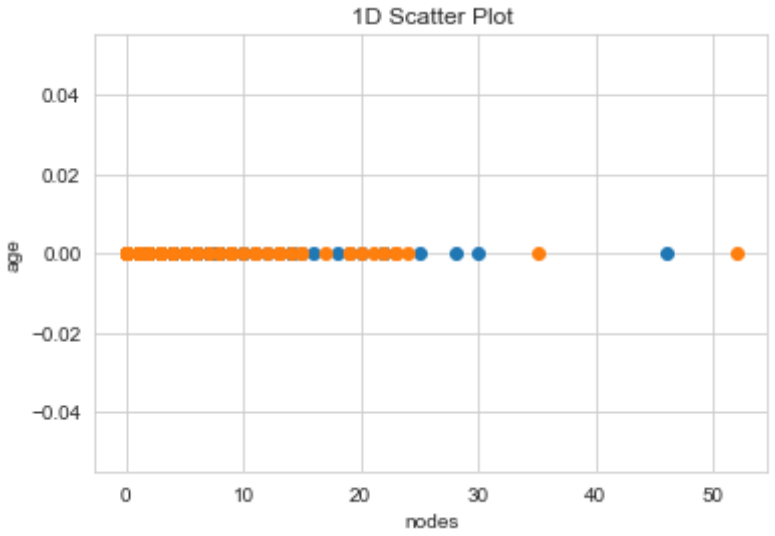
In [28]:
```
## 3D scatter plot using pair plot
plt.close();
sns.set_style('whitegrid');
g=sns.pairplot(haberman,hue='status',size=4,vars=(['age','year','nodes'
]))
g.fig.suptitle("Pair Plots for Age,Year & Nodes")
plt.legend()

plt.show()
```



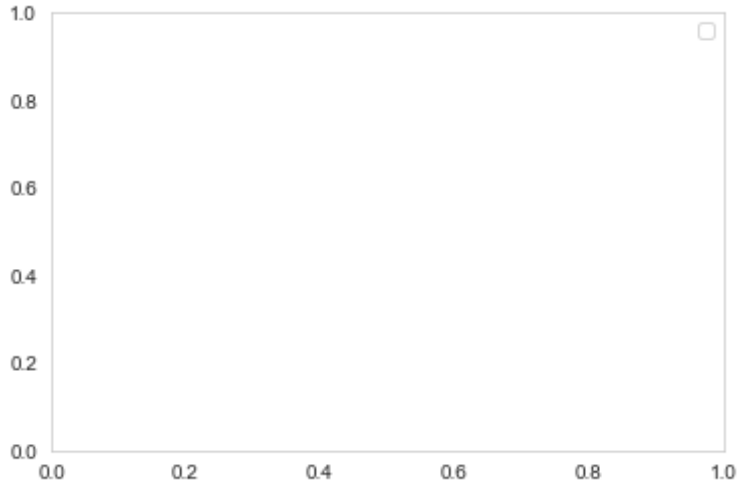
Above type of plot is konwn as pairplot to distingusih all features in the data. The data are highly mixed up,overlapping manner, none of the variable-pairs can help us find linearly separable clusters hence we can't analyse the survival stsus of the cancer patient

Now we are going to plot -D scatter plot by using below snippet on numpy. we are going to plot feature age vs axilary nodes.

In [40]:
```
## 1D scatter plot
plt.close()
haberman_status1= haberman.loc[haberman["status"]== 1];
haberman_status2= haberman.loc[haberman["status"] == 2];
plt.plot(haberman_status1["nodes"], np.zeros_like(haberman_status1['nod
es']),'o')
plt.plot(haberman_status2["nodes"], np.zeros_like(haberman_status2['nod
es']), 'o')
plt.xlabel('nodes')
plt.ylabel('age')
plt.title('1D Scatter Plot')
plt.show()
plt.legend()

plt.grid()
```




```
No handles with labels found to put in legend.
```



We can observe here that data points of status2(short survied patients) is mostly overlapping the status1(long survived patients)by which we can't come to conclusion here.

HISTOGRAM PDF CDF

for getting better results we are going to plot PDF CDF of data points histogram creates far more sense than 1-D scatter plot following code is for hitogram

```
Histogram-are nothing but they represents how many points exist
 how many points exist
```

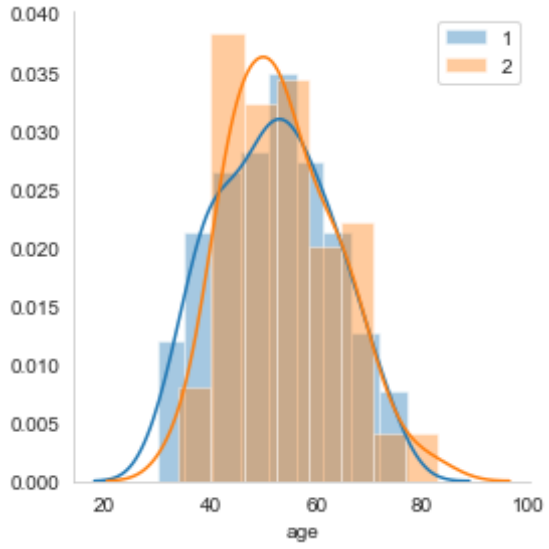
on your of your X axis.Histogram is XY plot wherex axis you have values corresponding to variable that are you are intrested in and Y axis represent how often how many times the data point corresponding to that variable.

PDF -Probability Density Function:-It shows the density of that data or number of data present on that point. PDF will be a peak like structure represents high peak if more number of data present

or else it will be flat/ small peak if number of data present is less. It is smooth graph plot using the edges of histogram

CDF-CDF (Cumulative Distribution Function):- It is representation of cumulative data of PDF ie. it will plot a graph by considering PDF for every data point cumulatively.

In [44]:
```
##PDF PLOT FOR AGE
sns.FacetGrid(haberman,hue="status",size = 4)\
.map(sns.distplot,'age')
plt.legend()
plt.grid()
plt.show()
```

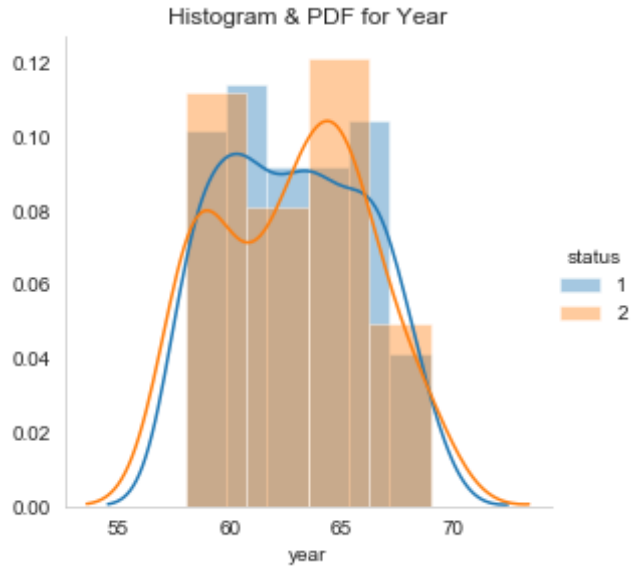


Observation:

This is PDF of age of patient at the time of operation.It is seen that the datapoints between 30-75 has nearabout same status because of overlapping from this we can't mention anything.

In [45]:
```
##PDF PLOT FOR YEAR
```

```
sns.FacetGrid(haberman,hue="status",size = 4)\
.map(sns.distplot,'year')\
.add_legend()

plt.grid()
plt.title('Histogram & PDF for Year')
plt.show()
```

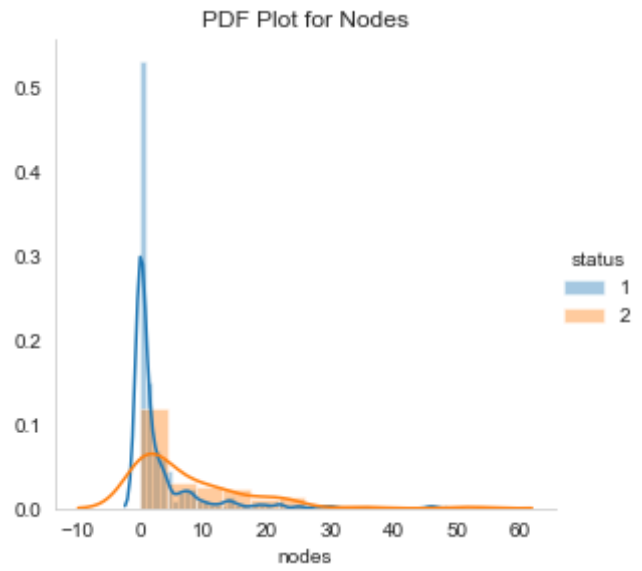


Observation:

This is PDF of Operation year of the paitent most data points overlapping each other so we could not predict from this.

In [57]:
```
##PDF PLOT FOR NODES

sns.FacetGrid(haberman,hue="status",size = 4)\
.map(sns.distplot,'nodes')\
.add_legend()
plt.title('PDF Plot for Nodes')
```

```
plt.grid()
plt.show()
```



Observation:

This is the PDF of Auxilary nodes.the dark blue line is called PDF it is a smooth form of our histogram From the above histogram we can observe that

1-Most of the patients survived at the axilary node 0-25.

2-Maximum people survived more than 5 years if they have less axilary node detected,here we can observe that patients who had undergone 0-3 axilary node surgery had higher chances of survival.

3-Here also we are getting morer overlapping of data points still it is hard to conclude if else condition.

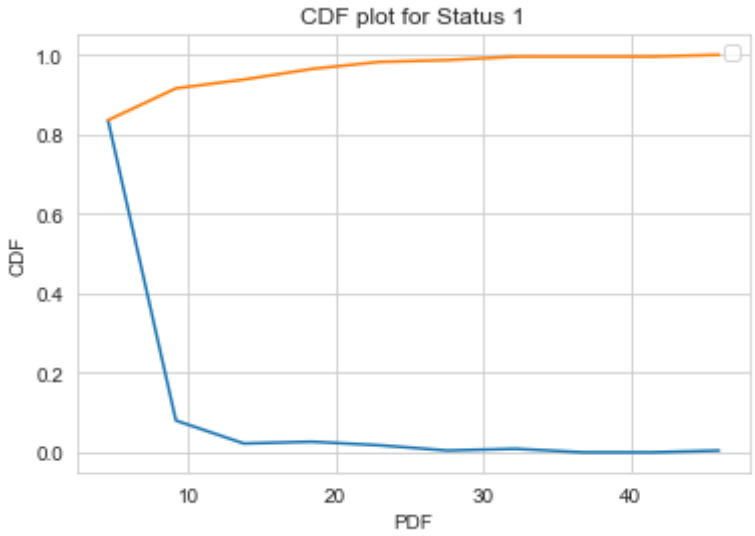
CDF PLOT

Now we are going to plot CDF for selected feature i.e.Auxilary node PDF tells us no of points in particular range CDF gives us idea about percentage

In [49]:
```
## CDF PLOT FOR STATUS1
counts,bin_edges = np.histogram(haberman_status1['nodes'],bins=10,densi
ty=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.grid()
plt.xlabel('PDF')
plt.ylabel('CDF')


plt.title('CDF plot for Status 1')
plt.legend()
plt.grid()
```

```
No handles with labels found to put in legend.
```

```
[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. ]
```



Observation:

1-From above CDF we can observe that orange line shows CDF and blue line shows PDF.

2-There is a 83% chance of survival more than 5 years if number of axillary nodes detected are less than 5.

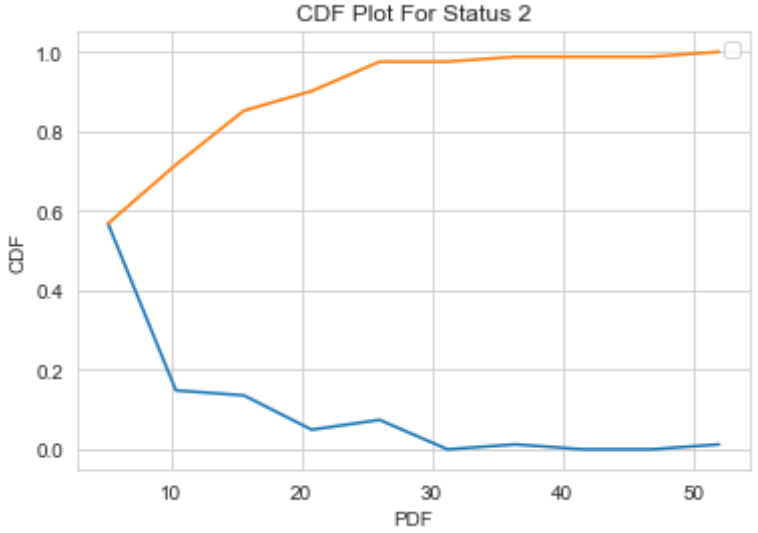
3-We can observe as number of axillary nodes increases survival chances also reduces means it is clearly observed that 80%—85% of people have good chances of survival if they have less no of auxillary nodes detected & as nodes increases the survival status also decreases as a result 100% of people have less chances of survival if nodes increases >40.

In [57]:
```
## CDF PLOT FOR STATUS2
counts,bin_edges = np.histogram(haberman_status2['nodes'],bins=10,densi
ty=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.grid()

plt.xlabel('PDF')
plt.ylabel('CDF')
plt.title('CDF Plot For Status 2')
plt.legend()
plt.grid()
plt.show()
```

```
No handles with labels found to put in legend.
```

```
[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. ]
```



Observation:

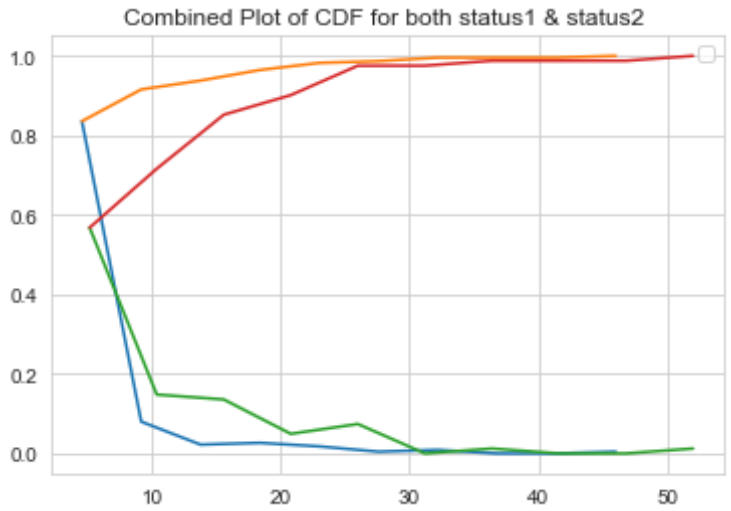
From above plot we can observe that near about 55% of people having axilary nodes less than 5 also nearly 100% of people in status 2 (short surviaval)if nodes are greater than 45.

In [11]:
```
##  COMBINED CDF PLOT FOR STATUS1 & STATUS2
counts,bin_edges = np.histogram(haberman_status1['nodes'],bins=10,densi
ty=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.grid()
plt.legend()
plt.title('Combined Plot of CDF for both status1 & status2 ')
counts,bin_edges = np.histogram(haberman_status2['nodes'],bins=10,densi
ty=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.grid()
plt.legend()
```

```
No handles with labels found to put in legend.
No handles with labels found to put in legend.
```

```
[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. ]
```

Out[11]:
```
<matplotlib.legend.Legend at 0x453cd08240>
```



Observation:

Above plot shows combination of pdf cdf of status 1 & status2.

We can predict patients status by applying mathematical formulae in numpy liabrary. SUCH AS mean,standard deviation.

In [21]:
```
## MEAN & STANDARD DEVIATION FOR NODE
print("Means:")
print(np.mean(haberman_status1['nodes']))

print(np.append(haberman_status1['nodes'],50))
print(np.mean(haberman_status2['nodes']))

print("\nStd-dev:");
print(np.std(haberman_status1['nodes']))
print(np.std(haberman_status2['nodes']))
```

```
Means:
```

```
2.7911111111111113
[ 1  3  0  2  4 10  0 30  1 10  7  0 13  0  1  0  0  0  0  6 15  0  2
0
  0  3  1  0 11  1  5  0  0  0  2  4  2  0  0  0  8  0  0  8  0  0  0
1
  2  4 20  0  1 14  2  3  0  2  0  4  0  1  0 16  0  0 14  0  0  1  0
3
  0  0  6  0  0  3  4  4 12  8  2  0  0  1  0  0  1  1  3  0  1  0  6
0
  1  1  2  0  0  4  1  7  1  0  1  0  4  0  4  5  0  1  0  0  0  1  1
2
  1  0  7  3  0 46  0  7 19  1  0  1  0  1 18  0  3 22  1  0  2  1  0
0
  9  0  0  0  0  0  0  0  0  3  1  0  0  3  2  0  0  1  4  0  7  3  1
2
 25  0  0  0  0  8  0  0  6  0  0  0  0  0  0  0  0  9 28  0 22  0  0
0
  0  0  0  2  0  1  0  1  0  0  0  0  0  0  0  0  0  0 14  0  0  8  0
2
  0  0  3  0  0  0  1  0  3 50]
7.45679012345679

Std-dev:
5.857258449412131
9.128776076761632
```

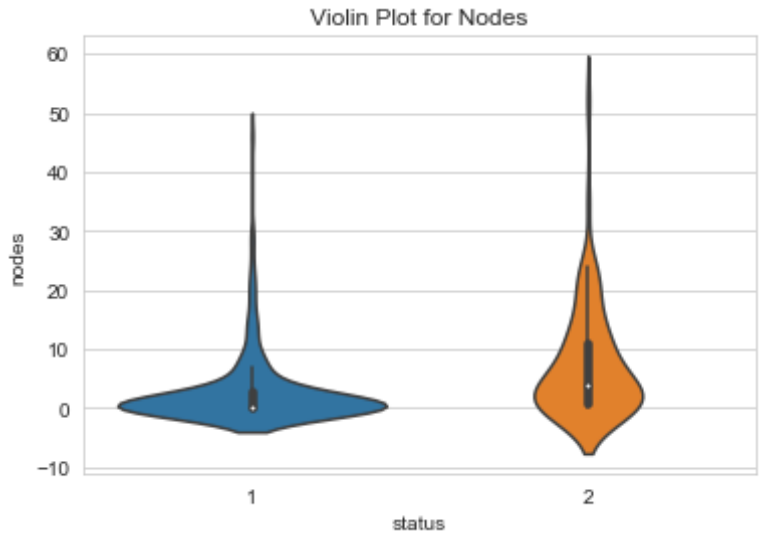
In [25]:

```
##MEAN & STANDARD DEVIATION FOR AGE
print("Means:")
print(np.mean(haberman_status1['age']))

print(np.append(haberman_status1['age'],50))
print(np.mean(haberman_status2['age']))

print("\nStd-dev:");
print(np.std(haberman_status1['age']))
print(np.std(haberman_status2['age']))
```

```
Means:
52.01777777777778
```

```
[30 30 30 31 31 33 33 34 34 34 34 34 35 35 36 36 37 37 37 37 37 37 38 3
8
 38 38 38 38 38 38 38 39 39 39 39 39 40 40 40 41 41 41 41 41 41 41 42 4
2
 42 42 42 42 42 43 43 43 43 43 43 43 44 44 44 44 45 45 45 45 45 45 46 4
6
 46 47 47 47 47 47 47 47 47 48 48 48 48 49 49 49 49 49 49 49 49 50 50 5
0
 50 50 50 50 50 50 50 51 51 51 51 52 52 52 52 52 52 52 52 52 52 53 53 5
3
 53 53 54 54 54 54 54 54 54 54 54 55 55 55 55 55 55 55 55 56 56 56 56 5
6
 57 57 57 57 57 57 57 57 58 58 58 58 58 58 58 59 59 59 59 59 59 59 60 6
0
 60 60 61 61 61 61 61 61 62 62 62 62 63 63 63 63 63 63 63 64 64 64 64 6
4
 65 65 65 65 65 65 66 66 66 67 67 67 67 68 68 69 69 69 70 70 70 70 70 7
1
 72 72 72 73 73 74 75 76 77 50]
53.67901234567901

Std-dev:
10.98765547510051
10.10418219303131
```

In [60]:
```
## VIOLIN PLOT FOR NODES
sns.violinplot(x="status", y="nodes", data=haberman, size=8)
plt.title('Violin Plot for Nodes')
plt.show()
```



Observation:

Violin plot is the combination of histogram and box olt both features including it

1-Central thick black portion gives us value of box plot and thin black lines are whiskers, white dot represents 50 th percentile whereas top & bottom line of thick black portion represents 75th and 25 th percentiles.

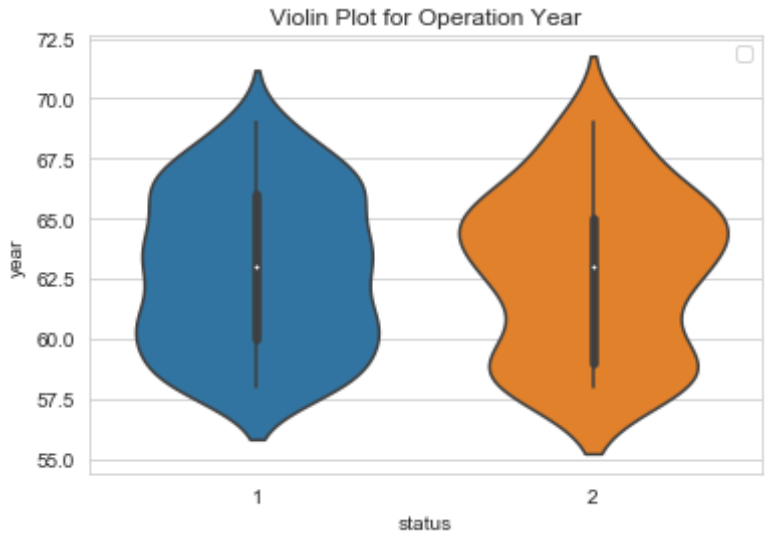
2-The region like bell curve reprresents distribution of data points.

In [62]:
```
## VIOLIN PLOT FOR AGE
sns.violinplot(x="status", y="age", data=haberman, size=8)
plt.title('Violin Plot for Age')
plt.legend()

plt.show()
```

```
No handles with labels found to put in legend.
```

In [63]:
```
## VIOLIN PLOT FOR YEAR
sns.violinplot(x="status", y="year", data=haberman, size=8);
plt.legend()
plt.title('Violin Plot for Operation Year')
plt.show()
```

```
No handles with labels found to put in legend.
```



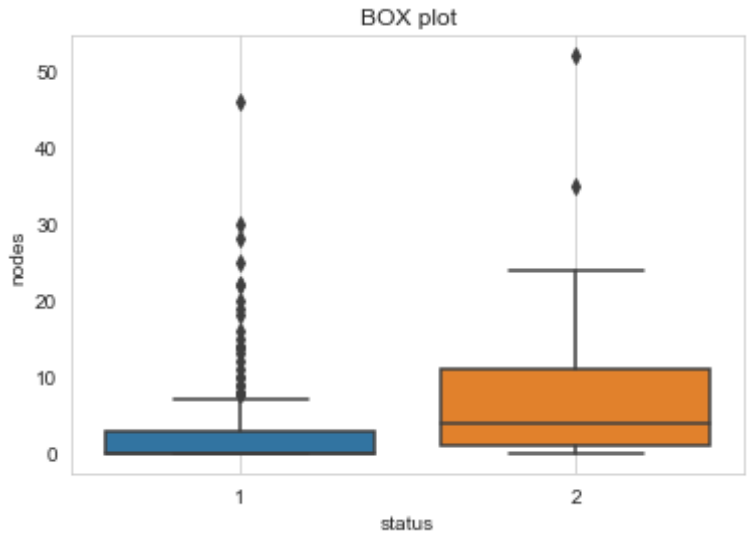
Observation:

From above two violin plot we can observe more no of patients who were dead have age between 46-62,year between 59-65 and the patients who survived have age between 42-60, year between 60-66.

BOX PLOT

Histogram not gives us nth percentile value i.e. 25th percentile value,75 percentile value whereas CDF gives us partially idea about it ,there are some other plot like BOxplotwhich can gives us clear idea about it.box plot uses percentiles.

In [68]:
```
plt.title('BOX plot')
sns.boxplot(x="status",y="nodes",data=haberman)

plt.grid()
plt.show()
```



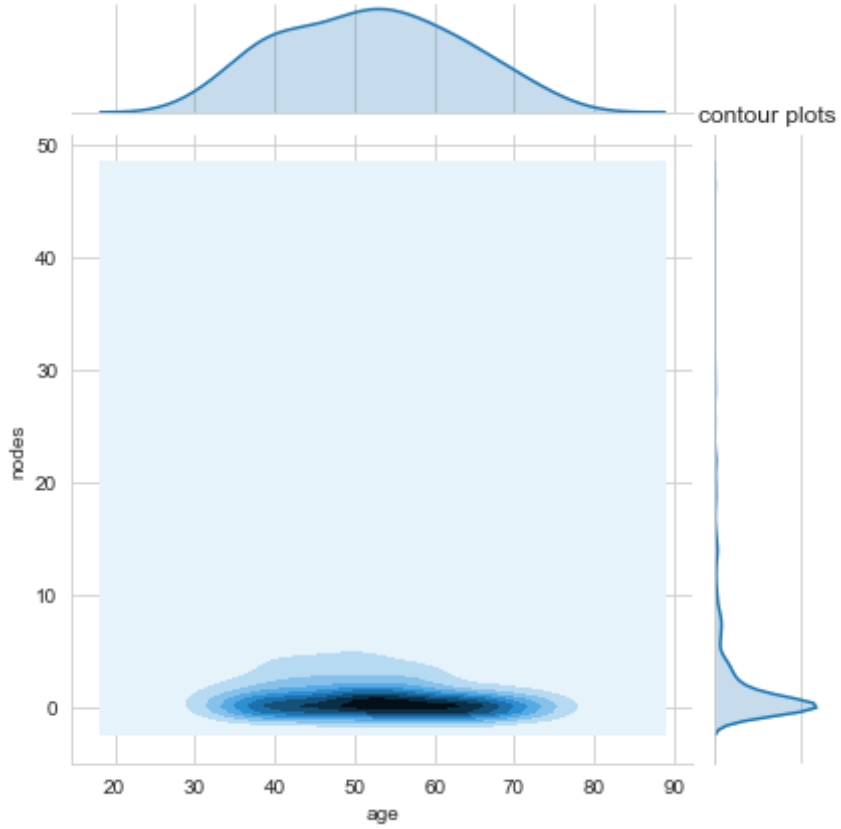
Observation:

Box plot gives percentile values.width of it has no significance.only horizontal value of the box represents 25-50-75th pertile value.

1-In above box plot we can seethat 25th percentile & 50th percentile are nearly the same its 0 75% is near about 5 for status 1(long survival).Thershold value for it ranges from 0-7 nodes.

2-Threshold for the Short survival 0 to 25 nodes and 75th% is 12 and 25th% is 1 or 2

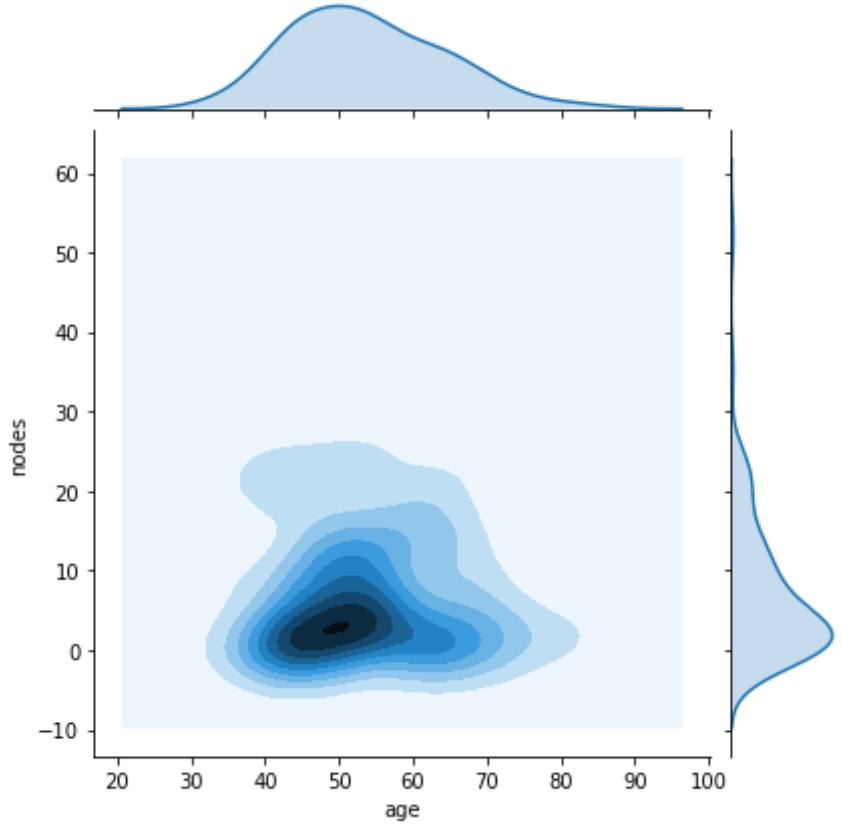
In [18]:
```
## CONTOUR PLOT
sns.jointplot(x="age",y="nodes",data=haberman_status1,kind="kde")
plt.grid()
plt.title('contour plots')
plt.show()
```



Observation:

Above is the 2D density plot for status 1(long survival)by using feature age and axilary nodes.The density of point for status 1 is more from age 45-65 and axilary nodes 0-2.Here area with dark shade shows maximum density whereas area with light shade shows low density.

In [18]:
```
sns.jointplot(x="age",y="nodes",data=haberman_status2,kind="kde" )
plt.show()
```



Observation:

From above contour plot we can observe that cancer patients with age between 45-55 has short survival of chanceswith axilary nodes 0-5.

In [11]:
```
##MEDIAN QUANTICLES 90TH PERCENTILE
print("\nMedians:")
print(np.median(haberman_status1["nodes"]))

print(np.median(np.append(haberman_status1["nodes"],50)))
print(np.median(haberman_status2["nodes"]))
```

```
print("\nQuantiles:")
print(np.percentile(haberman_status1["nodes"],np.arange(0,100,25)))
print(np.percentile(haberman_status2["nodes"],np.arange(0,100,25)))

print("\n90th Percentile:")
print(np.percentile(haberman_status1["nodes"],90))
print(np.percentile(haberman_status2["nodes"],90))

from statsmodels import robust
print("\nMedian Absolute Deviation")
print(robust.mad(haberman_status1["nodes"]))
print(robust.mad(haberman_status2["nodes"]))
```

```
Medians:
0.0
0.0
4.0

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

90th Percentile:
8.0
20.0

Median Absolute Deviation
0.0
5.930408874022408
```

CONCLUSION

1-There are 4 attribute out of which status is the output feature.

2-This dataset is unbalanced dataset out of 306 oprationsperformed

a)225 cancer patients have survival status1 who had survived more than 5 years.

b)81 cancer patients who had survival status 2 who has short survived i.e less than 5 years.

3- Age of the patients vary from 30-83 with average age of 52. Maximum number of positive lymph nodes observed is 52,nearly 75% have 4 positive lymph nodes and 25% have 0 positive lymph nodes.

4-Bivariate analysis with scatter plot a)We observed that blue datapoints overlaps with orange datapoints so we cant make any conclusion from 2D scatter plot.so we tried Bivariate analysis using 3D scatter plot here also its hard to conclude because of overlapping nature of all plots.

5-Univariate analysis by using HISTOGRAM,CDF,PDF.

a)From the PDF plot of patients age & operation year we observed that no any linearly seaparable nature of any plot most cases data points aresame for both status so we cant make any concluison from both of them.Axilary lymph node shows somewhat useful plot that we can see some difference n distribution.patients with 0-3 nodes have better probability of survival.

b)From CDF plot we can conclude that as the no of axialry lymph nodes increses patients survival chances also decreases.

6- From stastical data the mean of status 2 is greater than mean of status1.Thus haberman data will be helpful to diagnose cancer by using above different python progrmming.

In [ ]: