Data visualization using Haberman Dataset

Necessary Libraries

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

High level statistics of Dataset

In [3]:
```
# Reference: https://www.verywellhealth.com/lymph-node-positive-breast-cancer-429953
# Reference : https://www.kaggle.com/gilsousa/habermans-survival-data-set
# Import required Dataset
haberman = pd.read_csv("C:\\Users\\sai\\Datasets\\haberman.csv")# --> reading csv file
print(haberman.head()) # --> get first 5 data points using head() on the dataframe
print(haberman.shape)# --> we get 306 rows/records and 4 columns/variables
print(haberman.columns)# --> we can view the columns present in the data
print(haberman.describe()) # --> Initial information on each variables
print(haberman['status']) # --> we have 2 class of survival status as 1 and 2
print(haberman['status'].value_counts())# -->data points per class
```

```
   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
(306, 4)
Index(['age', 'year', 'nodes', 'status'], dtype='object')
              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
0      1
1      1
2      1
3      1
4      1
      ..
301    1
302    1
303    1
304    2
305    2
Name: status, Length: 306, dtype: int64
1    225
2     81
Name: status, dtype: int64
```

OBSERVATIONS:

1) we basically have 306 data points with 4 variables in which one variable is categorical and remaining are numerical variables.

2) Using describe() we come to know the comparison of mean - median and variance of each variable along with their range.

3) we also observed that the categorical variable is a class variable with Bi-classes as 1 => the patient survived 5 years or longer & 2 => the patient died within 5 year.

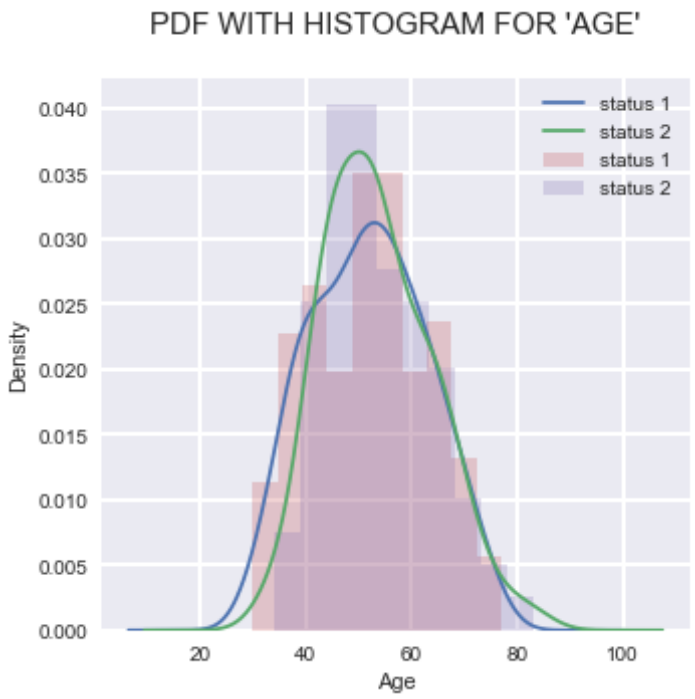
4) No of data points belonging to class 1 are 225 and class 2 are 81---->(It is highly imbalanced).

Objective : This is a Class Problem and hence we are required to analyse features useful for Classification

Univariate Analysis:

In [24]:
```
# PDF along with Histogram for 'age' variable

plt.style.use('seaborn')
plt.figure(figsize =(5,5))

filt1 = haberman['status'] == 1
x1 = haberman.loc[filt1,'age']

filt2 = haberman['status'] == 2
y1 = haberman.loc[filt2,'age']


kwargs1 = {"label":'status 1'}
kwargs2 = {"label":'status 2'}

x1.plot.kde(**kwargs1)
y1.plot.kde(**kwargs2)

plt.hist(x1,label = 'status 1',density = True,alpha = 0.25)
plt.hist(y1,label = 'status 2',density = True,alpha = 0.25)
plt.xlabel("Age")
plt.legend()
plt.title("PDF WITH HISTOGRAM FOR 'AGE'\n",fontsize = 15)
```

```
plt.grid(linewidth = 2)
plt.tight_layout()
plt.show()
```



Observations:

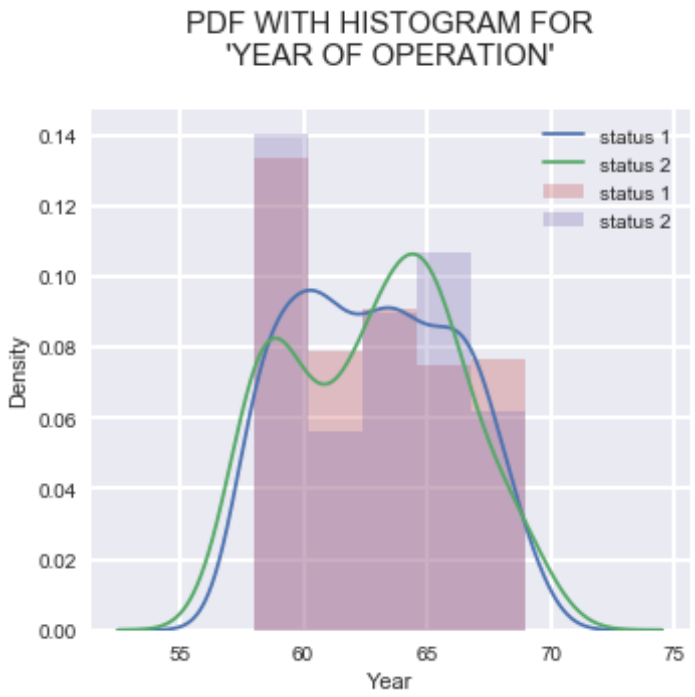
1) Most of data points for each class are overlapping and hence this feature is not providing us a good distinguishable property for our class label.

2) Early 30s-40s aged patients are having high chance of surviving for 5 years or longer.

3) Patients aged between 75-80 are having very less chance to survive.

In [22]:
```
# PDF along with Histogram for 'year of operation' variable
```

```
plt.close()
plt.style.use('seaborn')
plt.figure(figsize =(5,5))

filt1 = haberman['status'] == 1
x2 = haberman.loc[filt1,'year']

filt2 = haberman['status'] == 2
y2 = haberman.loc[filt2,'year']


kwargs1 = {"label":'status 1'}
kwargs2 = {"label":'status 2'}

x2.plot.kde(**kwargs1)
y2.plot.kde(**kwargs2)

plt.hist(x2,label = 'status 1',density = True,alpha = 0.3,bins = 5)
plt.hist(y2,label = 'status 2',density = True,alpha = 0.3,bins = 5)
plt.xlabel("Year")
plt.title("PDF WITH HISTOGRAM FOR\n'YEAR OF OPERATION'\n",fontsize = 15
)
plt.legend()
plt.grid(linewidth = 2)
plt.tight_layout()
plt.show()
```



Observations:

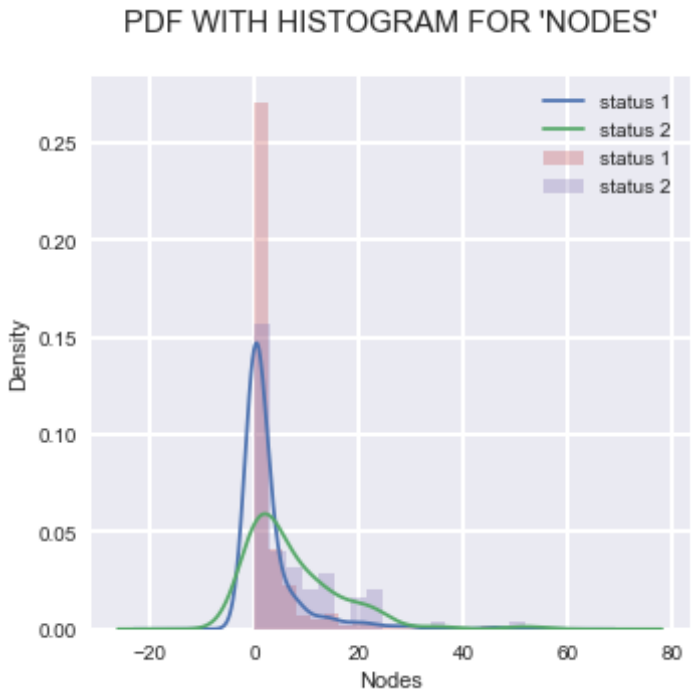
1) Similarly , even in the above plot we can't classify our class variable as there is equal density of datapoints overlapping each other.

2) So, this variable also does not help us in classifying our response variable.

3) Additionally if we observe the smooth curves, we can conclude that patients who had undergone the surgery in the gap of 1960-1962, there are more number of long survival patients than short survival patients and in the year gap of 1963-1965 the patients who had undergone the surgery , there are more number of short survival patients than long survival patients.

In [20]:
```
# PDF along with Histogram for 'Axillary nodes' variable
```

```
plt.close()
plt.style.use('seaborn')
plt.figure(figsize =(5,5))

filt1 = haberman['status'] == 1
x3 = haberman.loc[filt1,'nodes']

filt2 = haberman['status'] == 2
y3 = haberman.loc[filt2,'nodes']


kwargs1 = {"label":'status 1'}
kwargs2 = {"label":'status 2'}

x3.plot.kde(**kwargs1)
y3.plot.kde(**kwargs2)

plt.hist(x3,label = 'status 1',density = True,alpha = 0.3,bins = 17)
plt.hist(y3,label = 'status 2',density = True,alpha = 0.3,bins = 17)
plt.xlabel("Nodes")
plt.title("PDF WITH HISTOGRAM FOR 'NODES'\n",fontsize = 15)
plt.legend()
plt.grid(linewidth = 2)
plt.tight_layout()
plt.show()
```



Observations:

1) Auxilary Nodes provides a good classification feature for classifying status of survival of patients.

2) Patients having 0-3 nodes are more likely to survive for 5 years or longer.

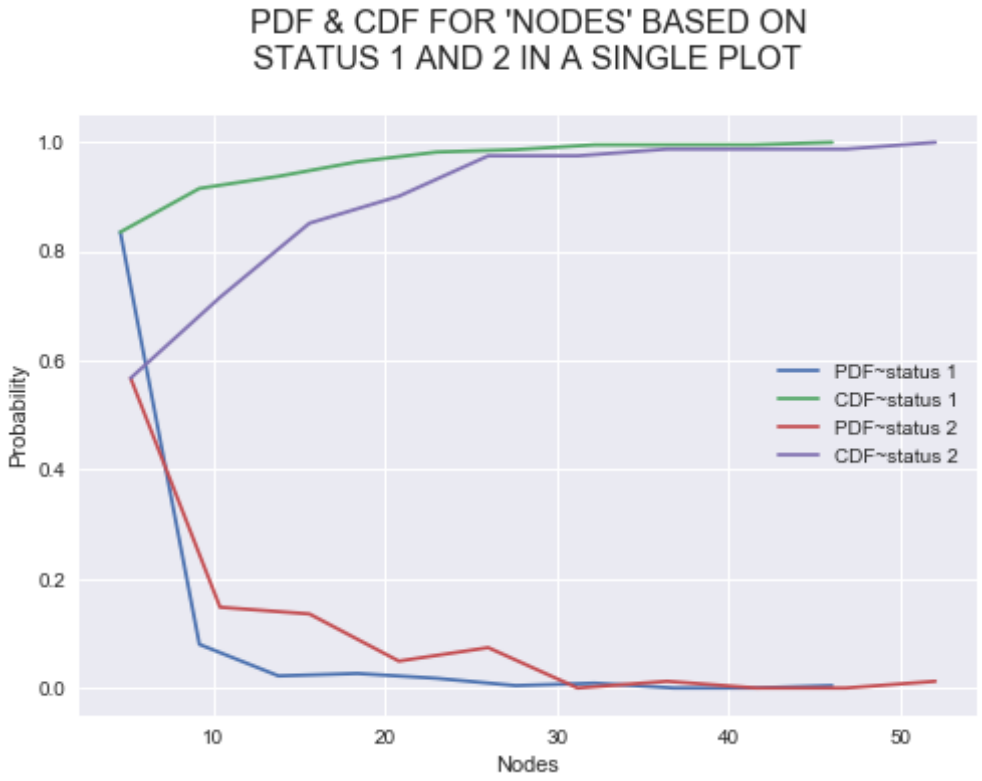
3) Patients having more than 3 nodes are more likely to die within 5 years. I would like to proceed my further analysis with this feature.

In [81]:
```
#Plot CDF  along with PDF for Both feature of status 1 and 2 in a singl
e plot
```

```
plt.close()
plt.style.use('seaborn')
freq, bin_edges = np.histogram(haberman.loc[haberman['status']==1,'node
s'], bins=10,
                               density = True)# for long survival

pdf = freq/(sum(freq))
print("PDF for long survival patient distributed as follows :\n",pdf,en
d="\n");
#print(bin_edges)

#compute CDF
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,label = "PDF~status 1")
plt.plot(bin_edges[1:],cdf,label = "CDF~status 1")
plt.legend()
plt.xlabel('Nodes')
plt.ylabel('Probability')

freq, bin_edges = np.histogram(haberman.loc[haberman['status']==2,'node
s'], bins=10,
                                 density = True)# for short survival
pdf = freq/(sum(freq))
print("PDF for short survival patient distributed as follows :\n",pdf);
#print(bin_edges)

#compute CDF
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,label = "PDF~status 2")
plt.plot(bin_edges[1:],cdf,label = "CDF~status 2")
plt.legend()
plt.xlabel('Nodes')
plt.ylabel('Probability')


plt.title("PDF & CDF FOR 'NODES' BASED ON\n STATUS 1 AND 2 IN A SINGLE
 PLOT \n",fontsize =16)
plt.show();
```

```
PDF for long survival patient distributed as follows :
```

```
 [0.83555556 0.08       0.02222222 0.02666667 0.01777778 0.00444444
 0.00888889 0.         0.         0.00444444]
PDF for short survival patient distributed as follows :
 [0.56790123 0.14814815 0.13580247 0.04938272 0.07407407 0.
 0.01234568 0.         0.         0.01234568]
```



Observations:

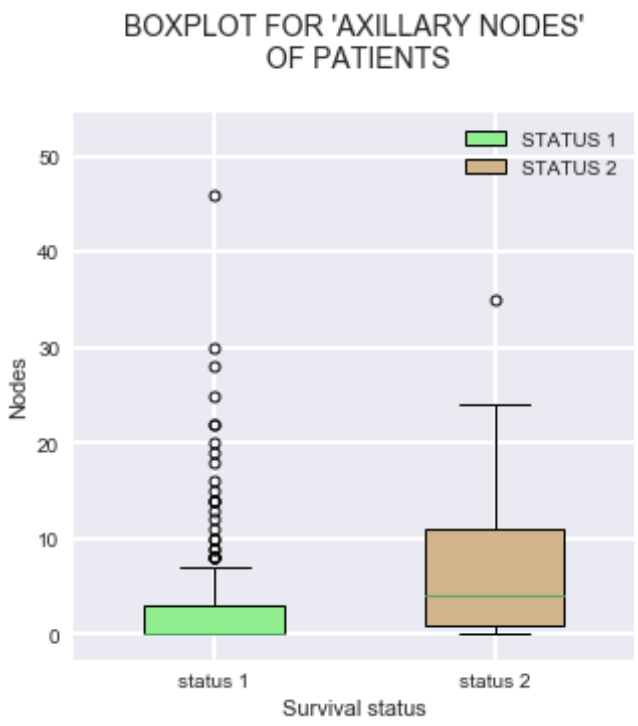
1) There is almost 82% of survival chance for patients having less than 5 axillary nodes.

2) From cumulative plot, we can say that almost 80-85 percent of people in general have high chance of survival.

3) There is only 57% chance that patient having nodes less than 5 die within 5 years.

4) As the nodes increases beyond 10, chances of survival is very less than non-survival status.

In [82]:
```
# Plot BOXPLOT

plt.close()
plt.figure(figsize =(5,5))
plt.style.use('seaborn')
flier_props = dict(marker="o", markersize=5)

box = plt.boxplot([x3,y3],labels = ['status 1','status 2'],
                  flierprops=flier_props,widths = 0.5,patch_artist=True
)

colors = ['lightgreen','tan']

for patch, color in zip(box['boxes'], colors):
    patch.set_facecolor(color)

plt.xlabel('Survival status')

plt.ylabel('Nodes')

plt.title("BOXPLOT FOR 'AXILLARY NODES'\n OF PATIENTS\n",fontsize = 14)

plt.legend([box["boxes"][0], box["boxes"][1]], ['STATUS 1', 'STATUS 2'
], loc='upper right')

plt.grid(linewidth = 2)
plt.show()
```



Observations:

1) The 25th percentile and 50th percentile is nearly equal for the 1st figure in boxplot which is numerically corresponding to zero, interquartile range corresponds to 0-4 and most of the data points until 100th percentile corresponds to 0-7 nodes.

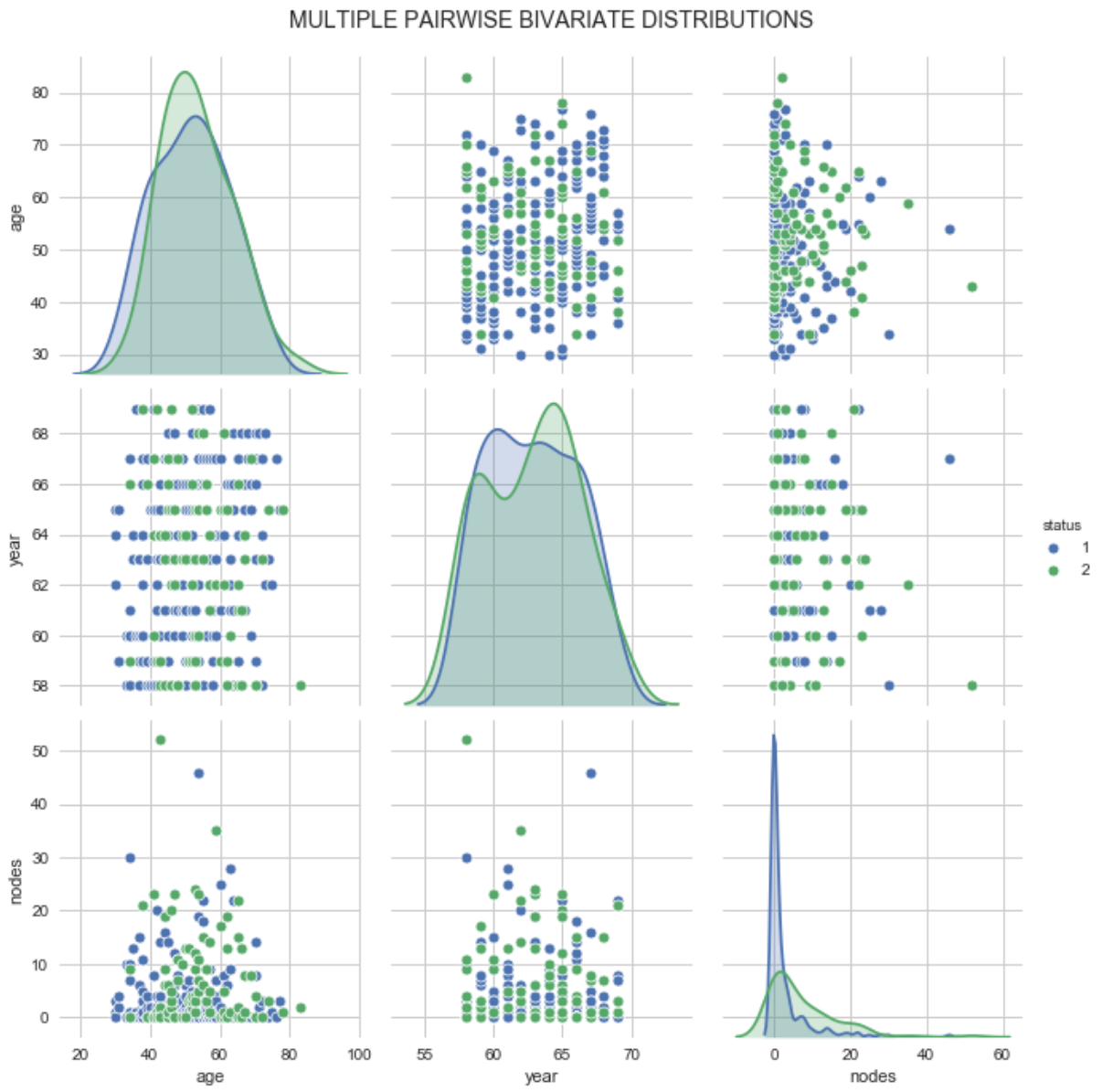
2) In the 2nd figure of the box plot, the median falls at 5 which is 50th percentile of our data , 75th percentile falls at 12 and overall range/ threshold for this plot is 0-25 nodes.

3) Finally, we can also observe that overall range of long survival status has an overlap with short survival which could develop chances of error in classification.

Bivariate Analysis:

Since, we have done univariate analysis and came up with one feature that is --> axillary nodes as a slightly apt feature for our classification. Lets now analyse if there are two variables as a pair or another variable along with nodes which as pair could help us to bring out even more better distinguishable difference for our class variable.

In [37]:
```
# plot PAIRPLOT

plt.close()

sns.set_style('whitegrid') # setting a style

f = sns.pairplot(haberman, hue='status',height =3)

f.fig.suptitle("MULTIPLE PAIRWISE BIVARIATE DISTRIBUTIONS\n",y = 1.04,f
ontsize = 14)

plt.show()
```

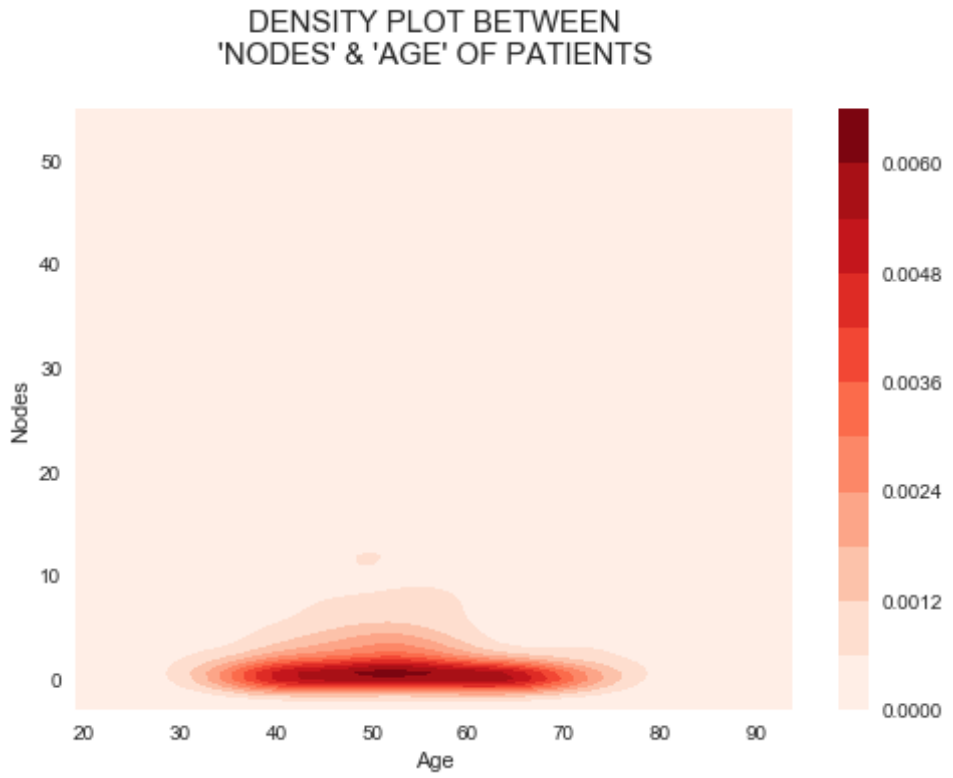


Observations:

1) From the pair plot, the main diagonal figure gives us the smooth histogram curve which we went through in univariate analysis.

2) when we observe the scatter points between year and age, we could not distinguish our class variable as there is high overlap between them, similarly all the other plots in upper triangle plots does not provide us best pair for classifying our class variable / response variable.

In [83]:
```
# plot CONTOUR PLOT(DENSITY PLOT):

plt.close()
sns.set_style('whitegrid') # setting a style

g = sns.kdeplot(haberman.age,haberman.nodes,shade = True,cmap = "Reds",
cbar = True)

plt.xlabel('Age')

plt.ylabel('Nodes')

plt.title("DENSITY PLOT BETWEEN\n'NODES' & 'AGE' OF PATIENTS\n",fontsiz
e = 15)

plt.show()
```



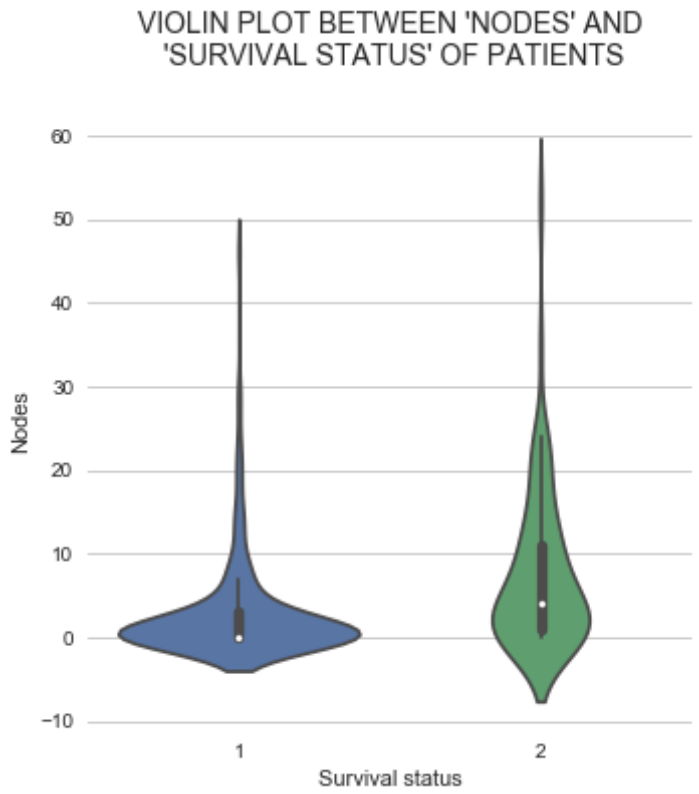
Observations:

1) We can observe that there is high density of points around 0-3 axillary nodes for the patients belonging to the age group of 40-60.

2) We can also notice that there are very less dense points around 4-10 axillary nodes for diverse age groups of patients.

In [84]:
```
# plot VIOLIN PLOT:

plt.close()
```

```
g = sns.catplot(x ='status',y = 'nodes',data = haberman,kind = 'violin'
)

plt.xlabel('Survival status')

plt.ylabel("Nodes")

plt.title("VIOLIN PLOT BETWEEN 'NODES' AND\n 'SURVIVAL STATUS' OF PATIE
NTS\n",fontsize = 14)

plt.show()
```



Observations:

1) For above violin plot we can observe that axillary nodes for long survival status has high density points scattered very close to zero and overall range of it lies between 0-7.5 nodes.

2) we can observe that axillary nodes for short survival status has density points spread slighly far from median and overall range of it lies between 0- 22nodes.

Mean and standard deviation:

In [18]:
```
print("Mean of axillary nodes for patients having long survival status
 :")
print(haberman.loc[haberman['status']== 1 , 'nodes'].mean())

print("Mean of axillary nodes for patients having short survival status
 :")
print(haberman.loc[haberman['status']== 2 , 'nodes'].mean())
```

```
Mean of axillary nodes for patients having long survival status :
2.7911111111111113
Mean of axillary nodes for patients having short survival status :
7.45679012345679
```

Observations:

1) On an average we can say that long survival patients have around 2.5 - 3 nodes.

2) On an average we can say that short survival patients have around 7.5 nodes.

3) But the above conclusion cannot be an apt finding for classifying our class variable since mean is influenced by outlier.

In [19]:
```
print("standard deviation of axillary nodes for patients having long su
rvival status :")
print(haberman.loc[haberman['status']== 1 , 'nodes'].std())
```

```
print("stadard deviation of axillary nodes for patients having short su
rvival status :")
print(haberman.loc[haberman['status']== 2 , 'nodes'].std())
```

```
standard deviation of axillary nodes for patients having long survival
status :
5.870318127719728
stadard deviation of axillary nodes for patients having short survival
status :
9.185653736555782
```

Observation:

1) Axillary nodes are highly scattered / spread around the mean for short survival patients than long survival patients.

Medians, Quantiles ,percentiles and MAD:

In [21]:
```
print("Middle value of axillary nodes for patients having long survival
 status :")
print(haberman.loc[haberman['status']== 1 , 'nodes'].median())

print("Middle value of axillary nodes for patients having short surviva
l status :")
print(haberman.loc[haberman['status']== 2 , 'nodes'].median())
```

```
Middle value of axillary nodes for patients having long survival status
:
0.0
Middle value of axillary nodes for patients having short survival statu
s :
4.0
```

Observation:

1) Since, Median is not highly affected by outliers, we can proceed our findings for optimal value of nodes by taking our median as an average term for each class variable.

2) It is more clear that If patients having zero nodes on an average, tends to survive longer .

3) If patients having 4 nodes on an average, tends to survive shorter.

In [25]:
```
print(" Quantiles : ")
print(np.percentile(haberman.loc[haberman['status']==1,'nodes'],np.aran
ge(0,125,25)))# for long survival
print(np.percentile(haberman.loc[haberman['status']==2,'nodes'],np.aran
ge(0,125,25)))# for short survival
```

```
 Quantiles :
[ 0.  0.  0.  3. 46.]
[ 0.  1.  4. 11. 52.]
```

Observation:

1) Quantiles helps us in showing the 50th percentile of nodes which is zero for long survival and 4 for short survival .

2) It also shows the 75th percentile of nodes which is 3 for long survival and 11 for short survival.

3) 100th percentile of nodes explains us that 100 percent of whole data for long survival and short survival lies below 46 and 52 respectively.

In [27]:
```
print(" Percentiles : ")
print(np.percentile(haberman.loc[haberman['status']==1,'nodes'],90))# f
or long survival
print(np.percentile(haberman.loc[haberman['status']==2,'nodes'],90))# f
or short survival
```

```
 Percentiles :
```

```
8.0

20.0
```

Observations:

1) The 90th percentile for long survival shows that almost 90 percent of the data has axillary nodes less than or equal to 8.0.

2) The 90th percentile for short survival shows that almost 90 percent of the data has axillary nodes less than or equal to 20.0.

In [33]:
```
from statsmodels import robust # necessary package to find median absol
ute deviation
print("Median absolute deviation :")
print(robust.mad(haberman.loc[haberman['status']==1,'nodes'])) # for lo
ng survival
print(robust.mad(haberman.loc[haberman['status']==2,'nodes'])) # for sh
ort survival
```

```
Median absolute deviation :
0.0
5.930408874022408
```

Observations:

1) Median absolute deviation serves us a better understanding of our axillary nodes having --> zero deviation from its median for long survival patients and approx --> 6 deviations from its median for short survival patients.

2) Median value being identified as 4.0 for short survival patients and Median absolute deviation being 6 deviations tends us to believe that there are some values in the axillary nodes of status 2 patients which is not optimal for our classification and corresponds to a considerable error rate in our classification model.

Conclusion:

(1) From the given features it is not a trivial task to classify the patient in terms of survival because there is considerable overlap between the survival status of patients.

(2) In General , patients having less number of axillary nodes tend to survive longer.

(3) As the class attribute comprises an imbalanced weight, our analysis could lead us with high risk of misclassification and errors in the future.

(4) Hence, This kind of exploratory analysis helped us to know more about the data in a descriptive way and also provided us the key features that can be used in a model.