Explorartory Data Analysis on Haberman Cancer Survival Dataset

Dataset link: https://www.kaggle.com/gilsousa/habermans-survival-data-set

By Ansh Rawat

anshrawat1298@gmail.com

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df=pd.read_csv('haberman.csv')
df.shape
```

Out[2]:
```
(306, 4)
```

In [3]:
```
df.head()
```

Out[3]:

	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

In [4]:
```
df.columns
```

Out[4]:
```
Index(['age', 'year', 'nodes', 'status'], dtype='object')
```

In [5]:
```
df['status'].value_counts()
```

Out[5]:
```
1    225
2     81
Name: status, dtype: int64
```

In [6]:
```
df.describe()
```

Out[6]:

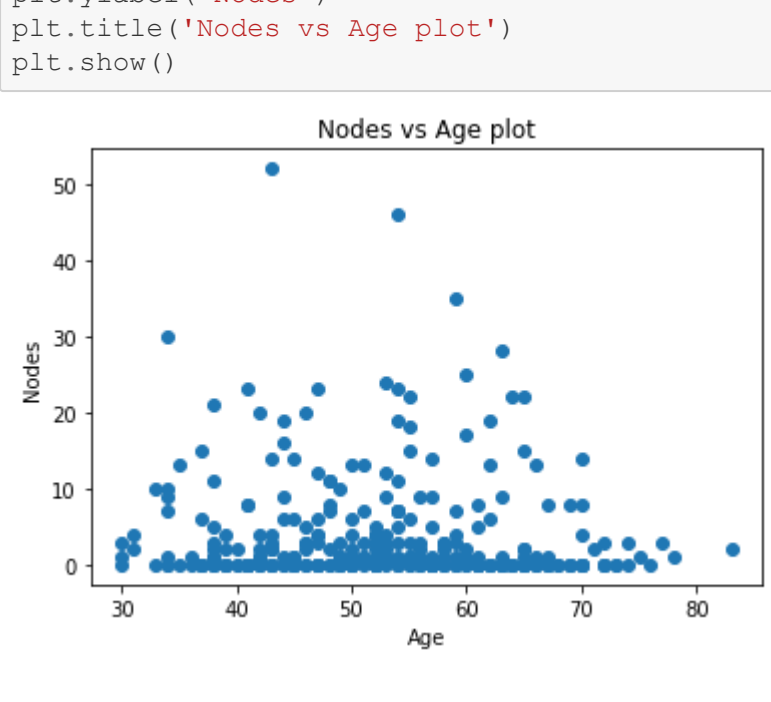
	age	year	nodes	status
count	306.000000	306.000000	306.000000	306.000000
mean	52.457516	62.852941	4.026144	1.264706
std	10.803452	3.249405	7.189654	0.441899
min	30.000000	58.000000	0.000000	1.000000
25%	44.000000	60.000000	0.000000	1.000000
50%	52.000000	63.000000	1.000000	1.000000
75%	60.750000	65.750000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

Haberman Dataset is the breast cancer survival dataset consisting of patients who did/didn't survive from breast cancer after under going the surgery.

From the above lines of code we can say that our data has 306 examples of pateints along which has 4 features out of which 3 features that is Age, Year and Nodes are the independent feature and Status is the dependent feature as per mentioned in Kaggle. Haberman is a unbalanced dataset as there are 225 training examples of patients who survived and only 81 patients who didn't. A summary is can also be seen which gives us a brief idea on the data like mean, standard deviation, min & max values and percentiles.

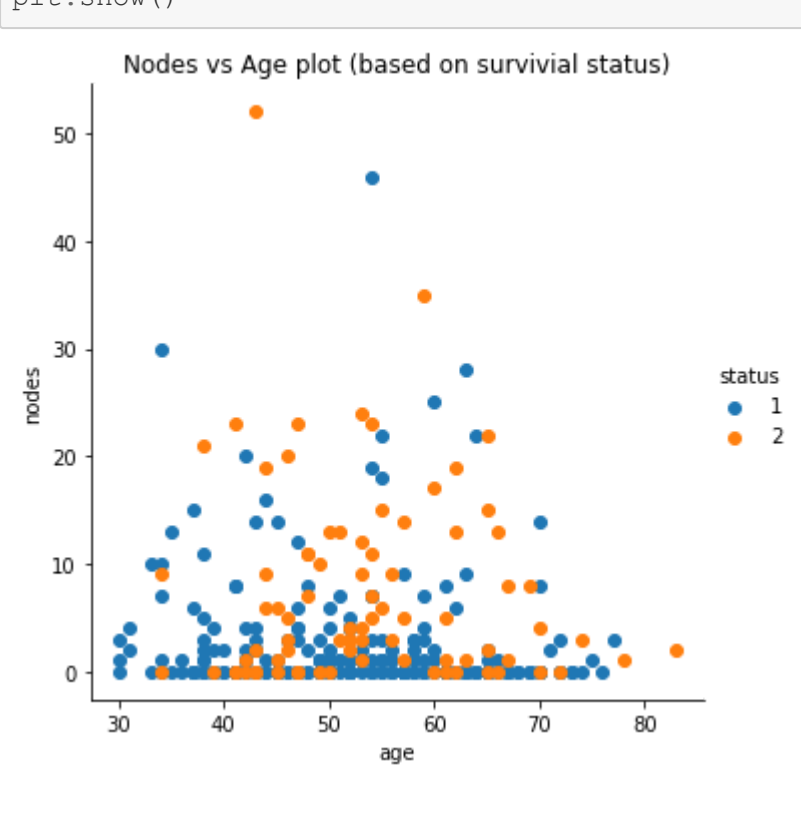
Bivariate Analysis

In [7]:
```
plt.scatter(df['age'],df['nodes'])
plt.xlabel('Age')
plt.ylabel('Nodes')
plt.title('Nodes vs Age plot')
plt.show()
```



Can't really get much information from this plot. Maybe seaborn can provide more details to it.

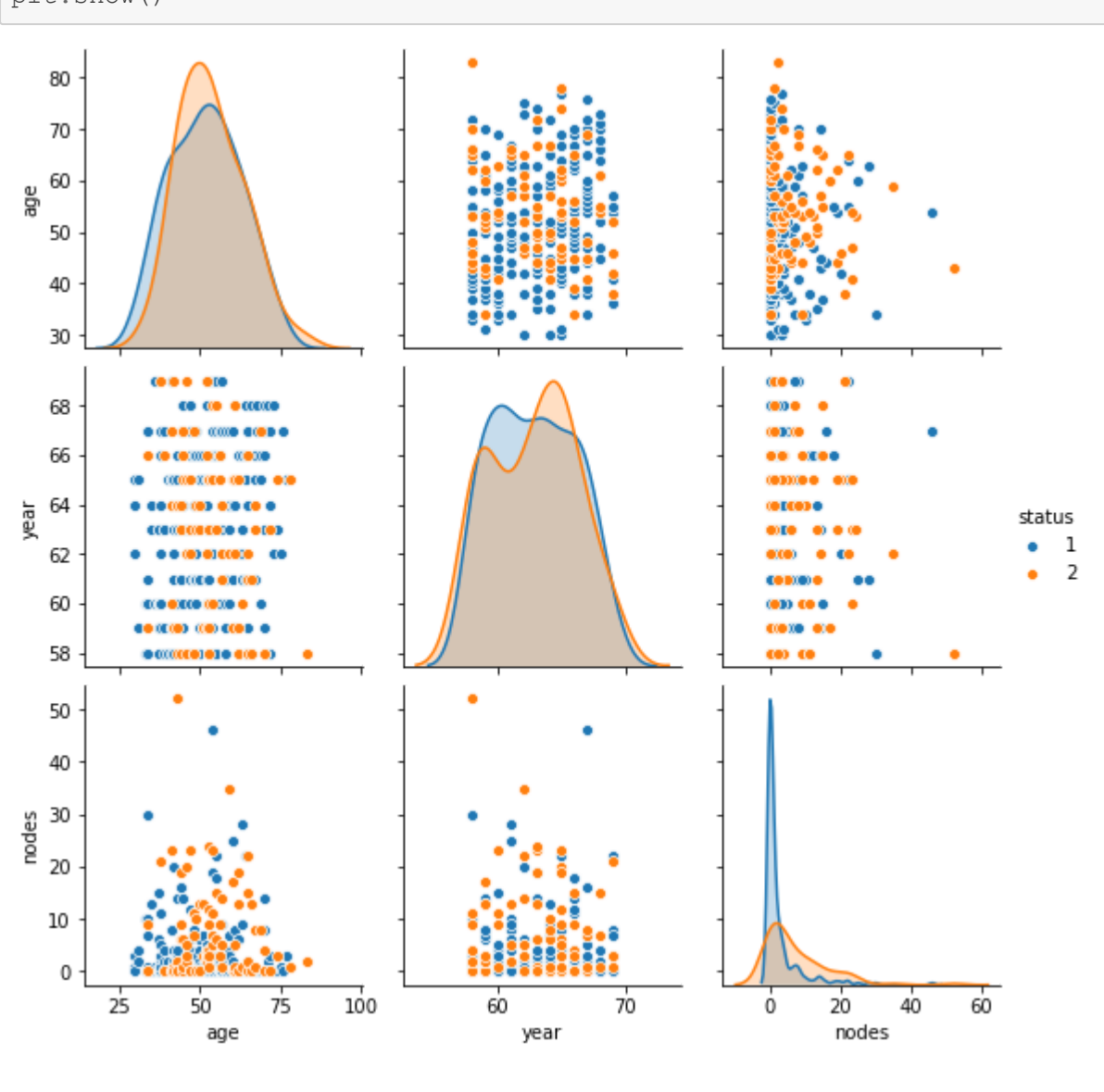
In [8]:
```
sns.FacetGrid(df,hue='status',height=5) \
    .map(plt.scatter,'age','nodes')\
    .add_legend()
plt.title('Nodes vs Age plot (based on survivial status)')
plt.show()
```



A lot of overlapping is there between survival for both kinds of patients. This plot is also not completely linearly separable. We can't make a solid conclusion based on this plot.

Let's try and pair plot all the features and see if there is another more useful features available in the dataset.

In [10]:
```
sns.pairplot(df, hue='status')
plt.show()
```



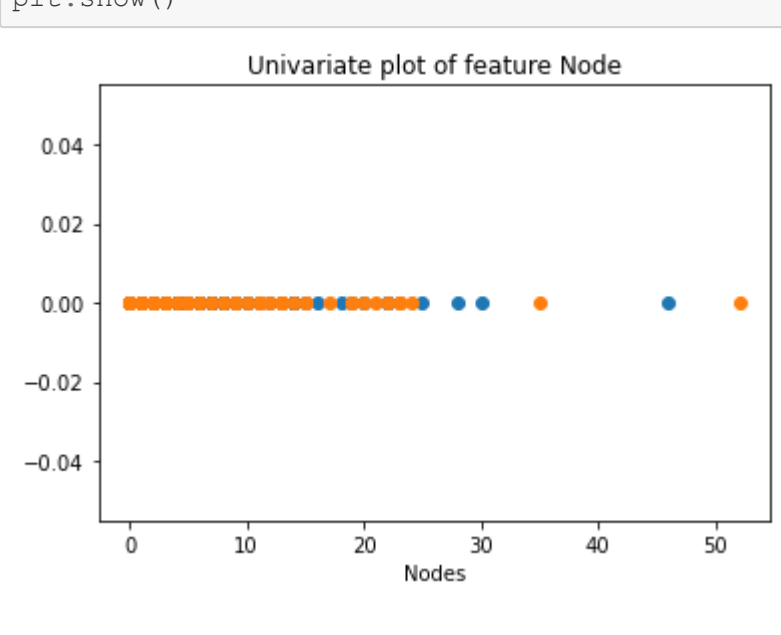
We can clearly see from the above plots that we can't differentiate the patients survival rate just by plotting age, year and nodes against each other.

The plot between nodes and age is better than the other ones but still it is not completely linearly separable. Rest all the features and plots are not linearly separable too.

The PDF (plot of nodes) of **nodes** for patients survived is very steep as compared to the patients who didn't survived.

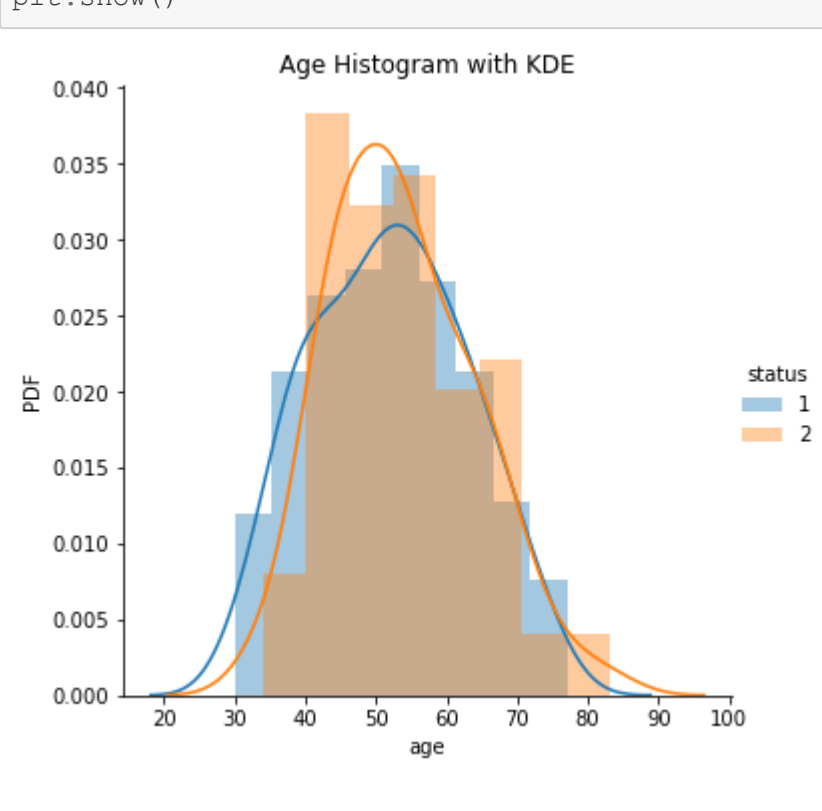
Univariate Analysis

In [11]:
```
sur = df.loc[df['status']==1]
nsur = df.loc[df['status']==2]
plt.plot(sur['nodes'], np.zeros_like(sur['nodes']),'o')
plt.plot(nsur['nodes'], np.zeros_like(nsur['nodes']),'o')
plt.xlabel('Nodes')
plt.title('Univariate plot of feature Node')
plt.show()
```



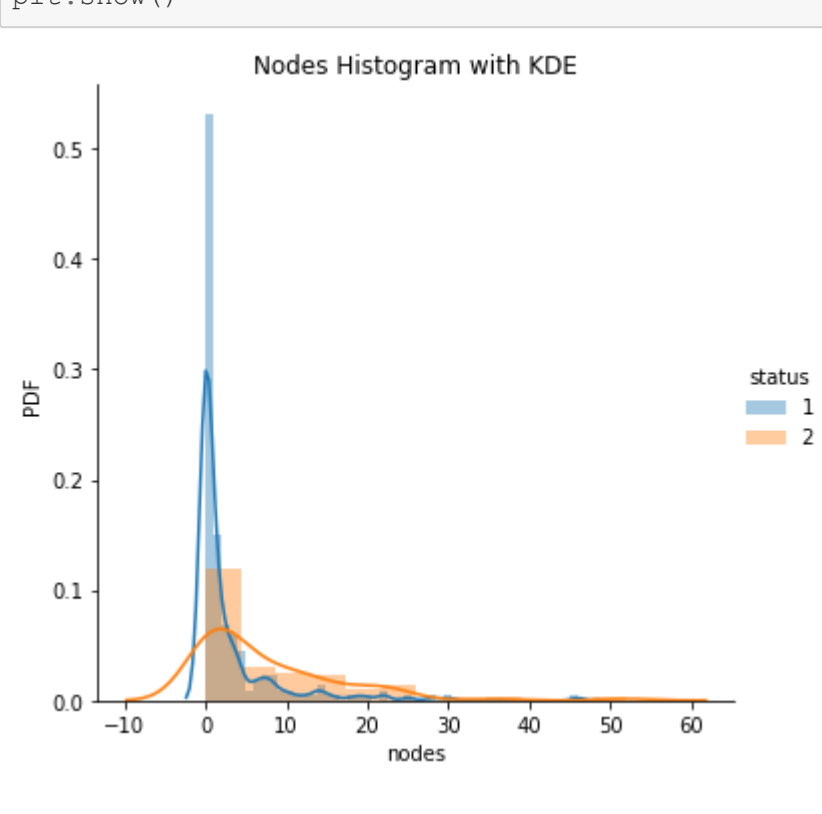
It's still quite hard to differentiate beteween survived or not survived patients based on number of nodes as there is too much overlapping. We shall get better idea by plotting PDFs of the features.

In [13]:
```
sns.FacetGrid(df, hue = 'status', height = 5)\
    .map(sns.distplot, 'age')\
    .add_legend()
plt.ylabel('PDF')
plt.title('Age Histogram with KDE')
plt.show()
```



No major conclusion can be made out of this plot as both the survival rate is well distributed throughout the feature age.

In [14]:
```
sns.FacetGrid(df, hue = 'status', height = 5)\
    .map(sns.distplot, 'nodes')\
    .add_legend()
plt.ylabel('PDF')
plt.title('Nodes Histogram with KDE')
plt.show()
```



From the above plot we can see that there is a peak in the PDF in the initial number of nodes. Also, both the PDFs are more streched towards the right hand side.

Therefore, from the above plot we can conclude that there is postive skewness in the feature nodes and probability of survival of a patient more than 5 years is more likely to happen when the number of nodes are very less.

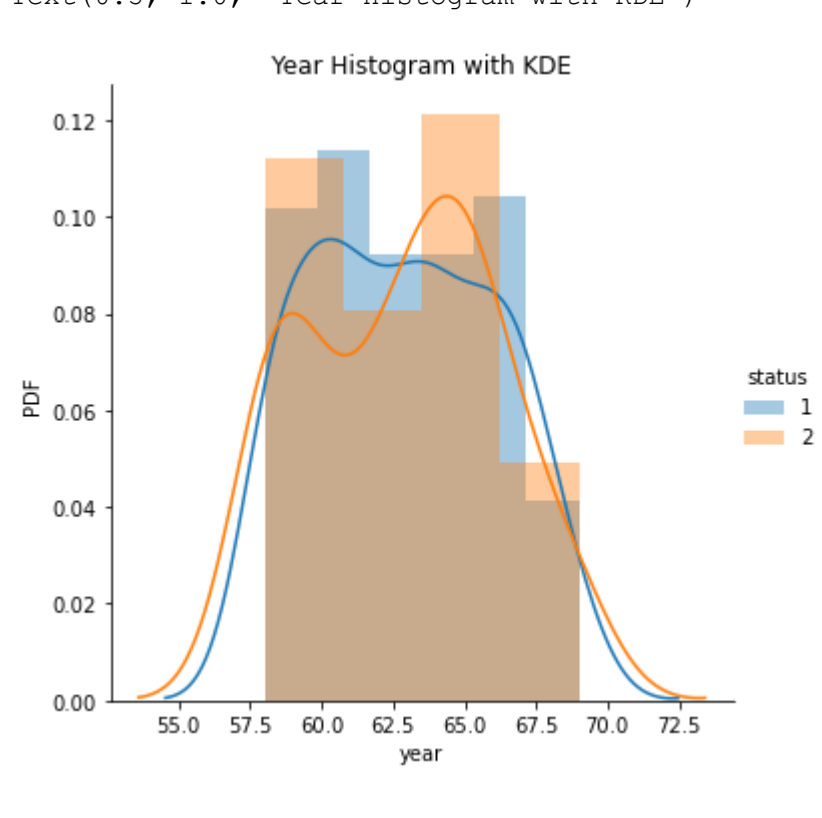
In [15]:
```
print(nsur[nsur['nodes']<=3].shape)
```

```
(39, 4)
```

Still we can't completely separate the survival rates based on simple if else condition because there are few cases in which number of nodes are less but still patient is failed to survive.

In [16]:
```
sns.FacetGrid(df, hue = 'status', height = 5)\
    .map(sns.distplot, 'year')\
    .add_legend()
plt.ylabel('PDF')
plt.title('Year Histogram with KDE')
```

Out[16]:
```
Text(0.5, 1.0, 'Year Histogram with KDE')
```

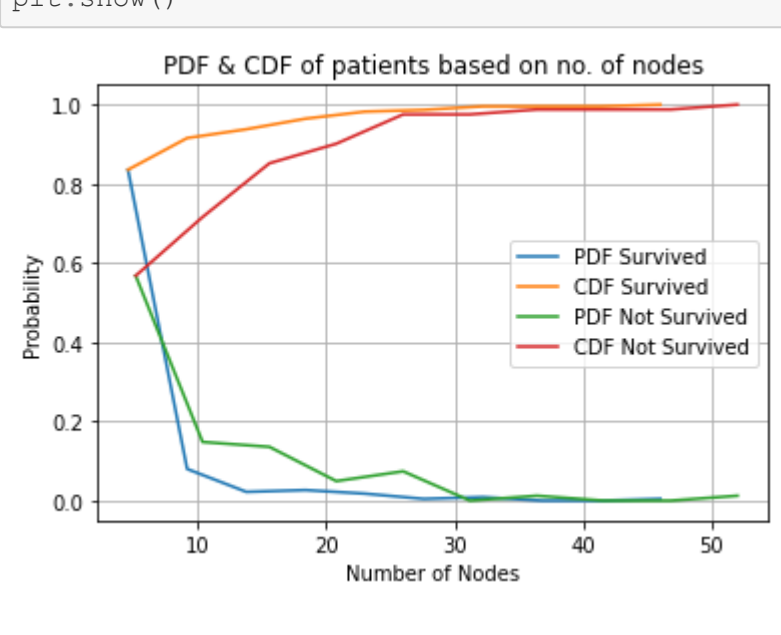


From the above plot no major conclusions can be made out as the status is well distributed among the year of operation feature.

Only the feature nodes come out to be some what useful from the help of PDFs. So let's plot CDF and check whether it gives any value addtions from our data.

In [18]:
```
counts, bins_edges = np.histogram(sur['nodes'])#, bins=10, density=True)
#print(counts)
#print(bins_edges)
pdf=counts/np.sum(counts)
#print(pdf)
cdf=np.cumsum(pdf)
#print(cdf)
plt.plot(bins_edges[1:],pdf, label='PDF Survived')
plt.plot(bins_edges[1:],cdf, label='CDF Survived')

counts, bins_edges = np.histogram(nsur['nodes'])#, bins=10, density=True)
pdf=counts/np.sum(counts)
cdf=np.cumsum(pdf)
plt.plot(bins_edges[1:], pdf, label='PDF Not Survived')
plt.plot(bins_edges[1:], cdf, label='CDF Not Survived')
plt.title('PDF & CDF of patients based on no. of nodes')
plt.grid()
plt.xlabel('Number of Nodes')
plt.ylabel('Probability')
plt.legend()
plt.show()
```



Observations from PDF and CDF

From the PDF and CDF of both survivied and non survived patients based on number of nodes we can conclude that probability or the percentage of survival of a patient is very very less or nearly equal to zero when number of nodes exceeds 30.

82% chances are there that patient will survive and 59% chances are there that patient will not surivive when number of nodes are very less.

Around 98% of our data is available when the number of nodess are less than 25.

But still these observations are not completely accurate as dataset is unbalanced.

Getting insights of data using some statistical tools

In [19]:
```
print('Mean of no. of nodes of patients survived: ',np.mean(sur['nodes']))
print('Mean of no. of nodes of patients not survived: ',np.mean(nsur['nodes']))
print('Mean of age of patients survived: ',np.mean(sur['age']))
print('Mean of age of patients not survived: ',np.mean(nsur['age']))
print('Mean of year of operation of patients survived: ',np.mean(sur['year']))
print('Mean of year of operation of patients survived: ',np.mean(nsur['year']))

print('Standard deviation of no. of nodes of patients survived: ',np.std(sur['nodes']))
print('Standard deviation of no. of nodes of patients not survived: ',np.std(nsur['nodes']))
print('Standard deviation of age of patients survived: ',np.std(sur['age']))
print('Standard deviation of age of patients not survived: ',np.std(nsur['age']))
print('Standard deviation of year of operation of patients survived: ',np.std(sur['year']))
print('Standard deviation of year of operation of patients survived: ',np.std(nsur['year']))
```

```
Mean of no. of nodes of patients survived:  2.7911111111111113
Mean of no. of nodes of patients not survived:  7.45679012345679
Mean of age of patients survived:  52.01777777777778
Mean of age of patients not survived:  53.67901234567901
Mean of year of operation of patients survived:  62.86222222222222
Mean of year of operation of patients survived:  62.82716049382716
Standard deviation of no. of nodes of patients survived:  5.857258449412131
Standard deviation of no. of nodes of patients not survived:  9.128776076761632
Standard deviation of age of patients survived:  10.98765547510051
Standard deviation of age of patients not survived:  10.10418219303131
Standard deviation of year of operation of patients survived:  3.2157452144021956
Standard deviation of year of operation of patients survived:  3.3214236255207883
```

From the average values of features above only the feature number of nodes comes out to be different than others.

Mean of number of nodes comes out to be less for the patients survived and large for the ones who didn't. But there can be outliers in the dataset so let's see median and median absolute deviation for these features as median is not much affected by outliers in the data.

In [20]:
```
print('Median of no. of nodes of patients survived: ',np.median(sur['nodes']))
print('Median of no. of nodes of patients not survived: ',np.median(nsur['nodes']))
print('Median of age of patients survived: ',np.median(sur['age']))
print('Median of age of patients not survived: ',np.median(nsur['age']))
print('Median of year of operation of patients survived: ',np.median(sur['year']))
print('Median of year of operation of patients survived: ',np.median(nsur['year']))

from statsmodels import robust
print('MAD of no. of nodes of patients survived: ',robust.mad(sur['nodes']))
print('MAD of no. of nodes of patients not survived: ',robust.mad(nsur['nodes']))
print('MAD of age of patients survived: ',robust.mad(sur['age']))
print('MAD of age of patients not survived: ',robust.mad(nsur['age']))
print('MAD of year of operation of patients survived: ',robust.mad(sur['year']))
print('MAD of year of operation of patients survived: ',robust.mad(nsur['year']))
```

```
Median of no. of nodes of patients survived:  0.0
Median of no. of nodes of patients not survived:  4.0
Median of age of patients survived:  52.0
Median of age of patients not survived:  53.0
Median of year of operation of patients survived:  63.0
Median of year of operation of patients survived:  63.0
MAD of no. of nodes of patients survived:  0.0
MAD of no. of nodes of patients not survived:  5.930408874022408
MAD of age of patients survived:  13.343419966550417
MAD of age of patients not survived:  11.860817748044816
MAD of year of operation of patients survived:  4.447806655516806
MAD of year of operation of patients survived:  4.447806655516806
```

Now we can say that the most useful feature from the dataset is number of nodes. Rest 2 features age and year are not helping us getting any useful insights of data as of now.

And since the median of patients survived of nodes feature is 0 and also median absolute deviation is 0, so we can say that probability of surivial is higher when the number of nodes are near 0.

We can observe the percentiles and quantiles of this feature to increase our probability even more.

In [21]:
```
print('Quantiles of feature node when patient survive: ',np.percentile(sur['nodes'],(25,50,75,100)))
print('Quantiles of feature node when patient does not survive: ',np.percentile(nsur['nodes'],(25,50,75
,100)))
print('90th Percentile of patient survival: ',np.percentile(sur['nodes'],90))
print('90th Percentile of patient not surviving: ',np.percentile(nsur['nodes'],90))
print('40th Percentile of patient not surviving: ',np.percentile(nsur['nodes'],40))
```

```
Quantiles of feature node when patient survive:  [ 0.  0.  3. 46.]
Quantiles of feature node when patient does not survive:  [ 1.  4. 11. 52.]
90th Percentile of patient survival:  8.0
90th Percentile of patient not surviving:  20.0
40th Percentile of patient not surviving:  3.0
```

Observations from Percentiles and Quantiles

The 50th Percentile of survived patients comes out to be 0 that is minimum or equal to 0 numbers of nodes are there for 50% of patients and 50% of the patients are there in which nodes are greater than 0.

The 75th Percentile of survived patients comes out to be 3 that is minimum or equal to 3 numbers of nodes are there for 75% of patients and 25% of the patients are there in which nodes are greater than 3.

The 50th Percentile of patients who didn't survived comes out to be 4 that is minimum or equal to 4 numbers of nodes are there for 50% of the patients and 50% of the patients are there in which nodes are greater than 4.

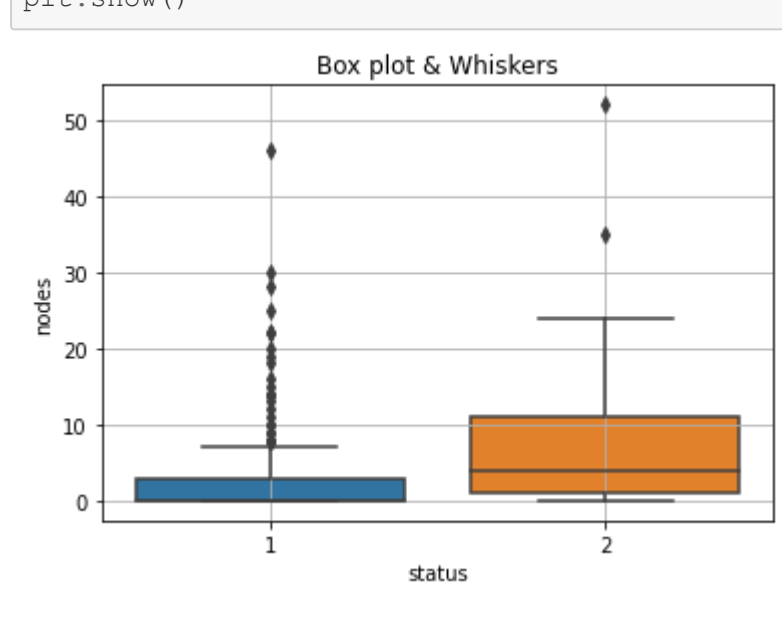
The 75th Percentile of patients who didn't survived comes out to be 11 that is minimum or equal to 11 numbers of nodes are there for 75% of the patients and 25% of the patients are there in which nodes are greater than 11.

We can also conclude that the Inter-Quantile Range (IQR) of patients surived is 3 as 75th percentile is 3 and 25th percentile is 0 and their difference comes out to be 3. Similarly, IQR of patients who didn't survive is 10.

Therefore, we can say that number of nodes is a factor by which probability of a patient survival can be determined.

Box plot, Whiskers and Violin plot

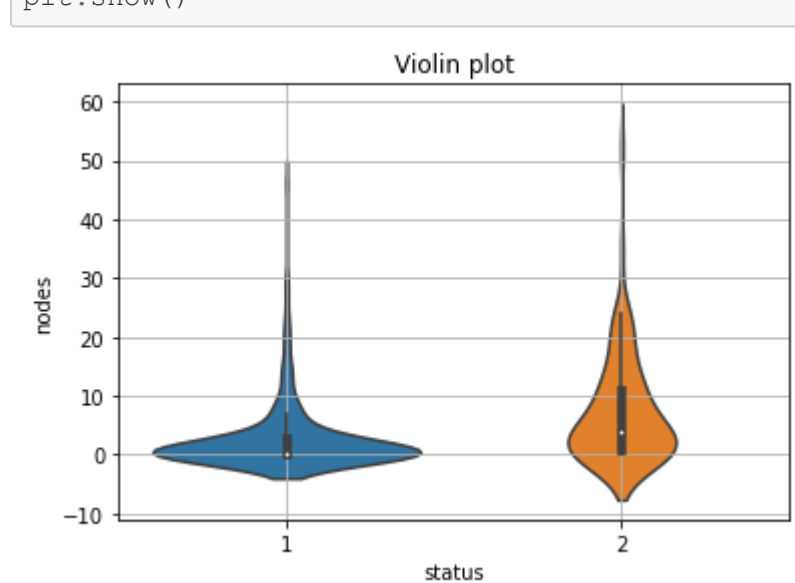
In [22]:
```
sns.boxplot(x='status', y='nodes', data=df)
plt.title('Box plot & Whiskers')
plt.grid()
plt.show()
```



From the above box plot we can conclude that if we say that patients having number of nodes less than 3 (75th percentile value of patients survived) then the patient would survive then, this probablity is would be having an error of 40% as the 40th percentile of patients not surviving is 3 and those patients would fall in the same category.

Therefore, 40% error can be expected if we conclude that patients having nodes less than 3 would survive.

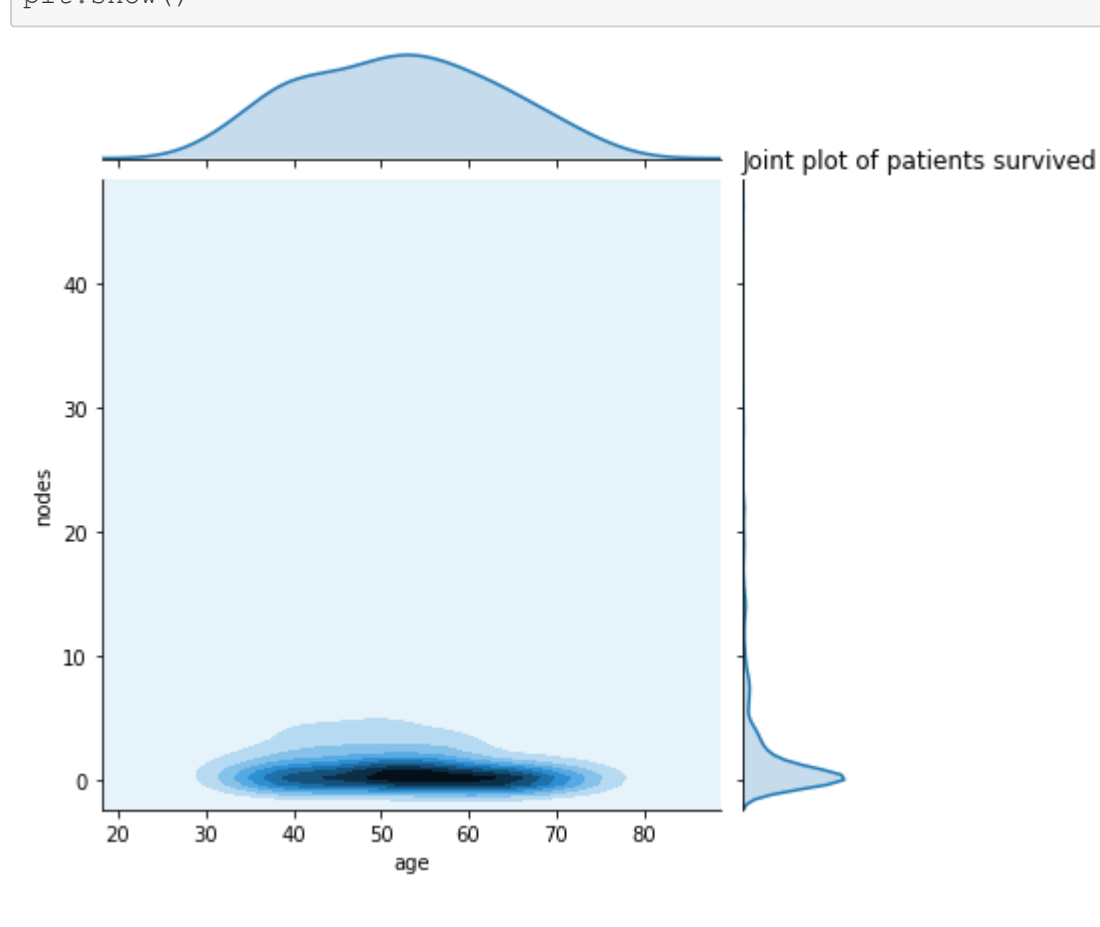
In [23]:
```
sns.violinplot(x='status', y='nodes', data=df)
plt.title('Violin plot')
plt.grid()
plt.show()
```



From the above violin plot we can say that probability of patient survival is higher when number of nodes are 0 as the PDF of survived patients is more steep and 50th percentile is also 0.

Join plot of two features (Bivariate Analysis)

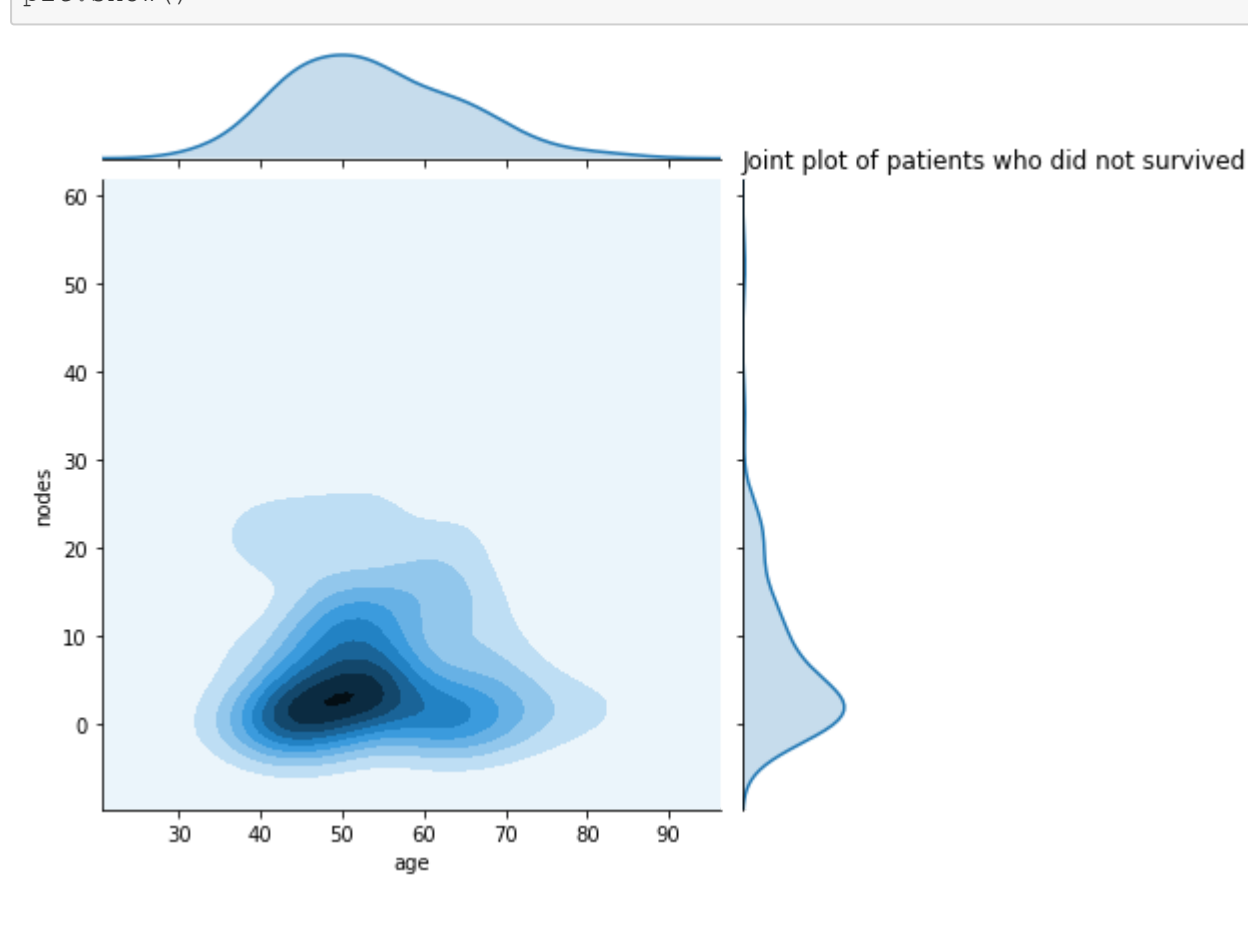
In [24]:
```
sns.jointplot(x='age',y='nodes', data=sur, kind='kde')
plt.title('Joint plot of patients survived', loc='left')
plt.show()
```



From the above joint plot between age and nodes of survived patients we can say that majority of the patients aged between 43 to 65 survived with number of nodes equal to 0.

Also, since the most dense region is near 0 nodes (the black region) then we can again say that the probability of survival is greater when number of nodes are equal to 0.

In [25]:
```
sns.jointplot(x='age', y='nodes', data=nsur, kind='kde')
plt.title('Joint plot of patients who did not survived', loc='left')
plt.show()
```



From the above joint plot between age and nodes of patients who didn't survived we can say that majority of the patients of age between 43 and 53 didn't survived with nodes greater than 0.

Also, this joint plot is more expanded towards greater number of nodes. So, we can also say that greater the number of nodes then greater is the probability of patient not surviving.

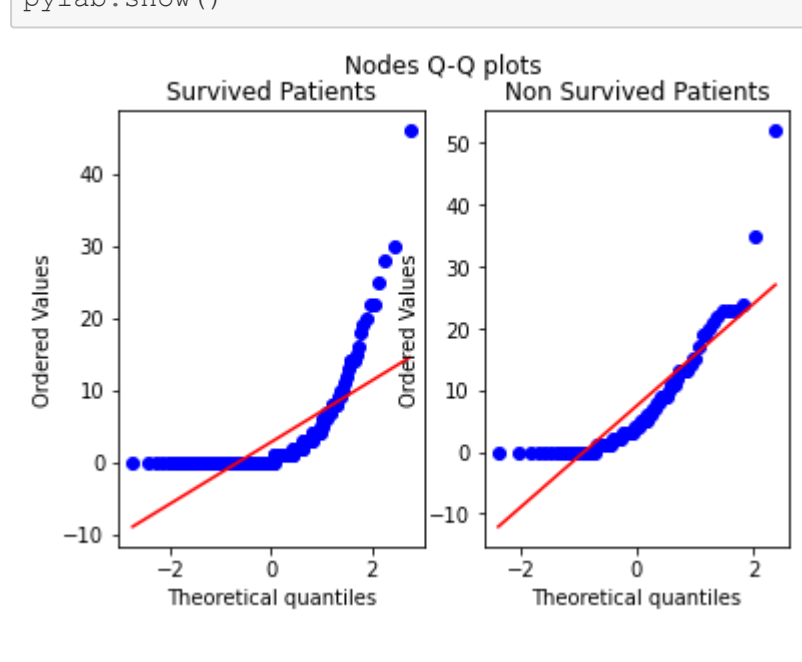
Q-Q Plot

From the PDFs we saw that features of our dataset is not from Guassian or Normal distribution but to be double sure let's do a Q-Q plot of our feature nodes and age.

In [26]:
```
import pylab
from scipy import stats
plt.subplot(1,2,1)
stats.probplot(sur['nodes'], dist='norm', plot=pylab)
plt.title('Survived Patients')

plt.subplot(1,2,2)
stats.probplot(nsur['nodes'], dist='norm', plot=pylab)
plt.title('Non Survived Patients')

plt.suptitle('Nodes Q-Q plots')
pylab.show()
```

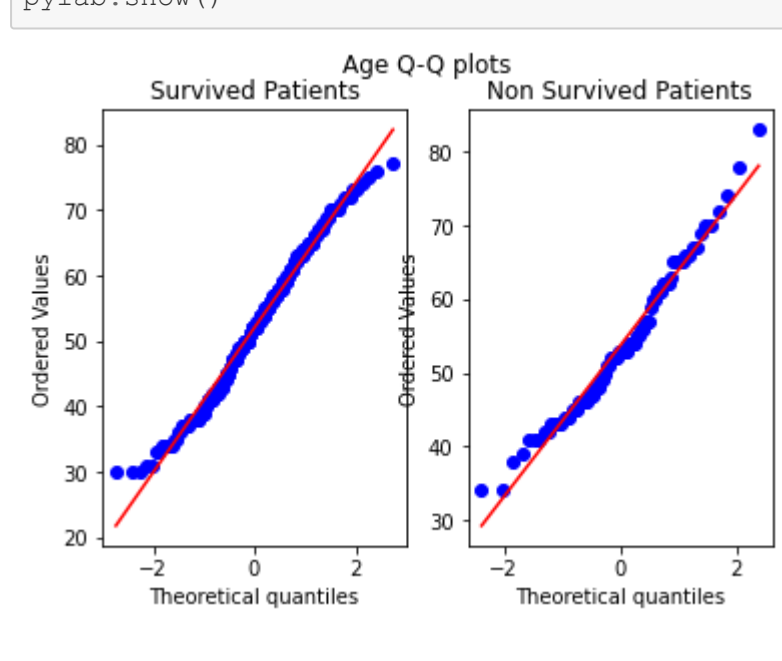


Since the values are not overlapping completely over the theoretical values, therefore we can confirm that our data does not belongs to Gaussian/Normal distribution.

In [27]:
```
plt.subplot(1,2,1)
stats.probplot(sur['age'], dist='norm', plot=pylab)
plt.title('Survived Patients')

plt.subplot(1,2,2)
stats.probplot(nsur['age'], dist='norm', plot=pylab)
plt.title('Non Survived Patients')

plt.suptitle('Age Q-Q plots')
pylab.show()
```



However, the feature age is somewhat better than feature nodes as there are more number of points overlapping the line. But still in none of the plots the points are completely overlaping the line.

Therefore, our dataset is not from Guassian or Normal distribution.

K-S Test

In [28]:
```
print('KS Test for survived patient nodes: ', stats.kstest(sur['nodes'],'norm'))
print('KS Test for survived patient age: ', stats.kstest(sur['age'],'norm'))
print('KS Test for survived patient year: ', stats.kstest(sur['year'],'norm'))

print('KS Test for non survived patient nodes: ', stats.kstest(nsur['nodes'],'norm'))
print('KS Test for non survived patient age: ', stats.kstest(nsur['age'],'norm'))
print('KS Test for non survived patient year: ', stats.kstest(nsur['year'],'norm'))
```

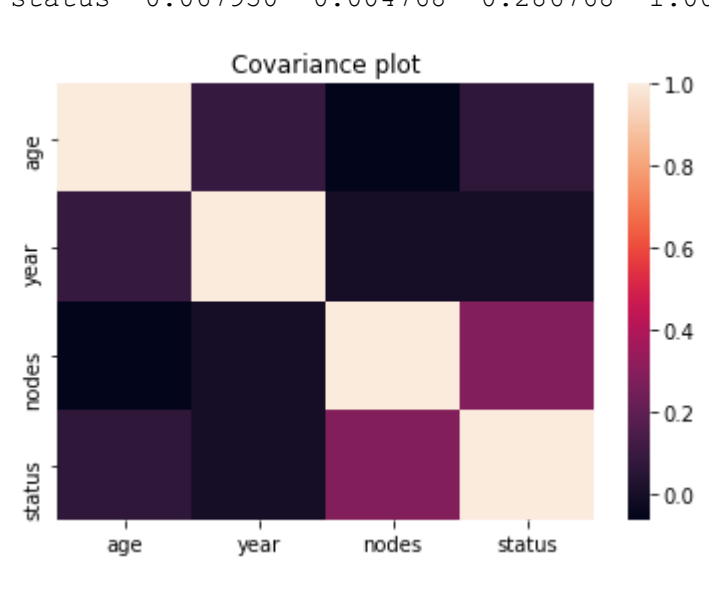
```
KS Test for survived patient nodes:  KstestResult(statistic=0.5, pvalue=1.538448448280327e-52)
KS Test for survived patient age:  KstestResult(statistic=1.0, pvalue=0.0)
KS Test for survived patient year:  KstestResult(statistic=1.0, pvalue=0.0)
KS Test for non survived patient nodes:  KstestResult(statistic=0.6439165347184874, pvalue=3.00899499
73798837e-33)
KS Test for non survived patient age:  KstestResult(statistic=1.0, pvalue=0.0)
KS Test for non survived patient year:  KstestResult(statistic=1.0, pvalue=0.0)
```

Since in none of the case the p-value is near 1. Therefore, our data isn't from Normal or Guassian distribution.

Covariance

In [29]:
```
print('Covariance matrix')
print(df.corr())
sns.heatmap(df.corr())
plt.title('Covariance plot')
plt.show()
```

```
Covariance matrix
             age      year     nodes    status
age     1.000000  0.089529 -0.063176  0.067950
year    0.089529  1.000000 -0.003764 -0.004768
nodes  -0.063176 -0.003764  1.000000  0.286768
status  0.067950 -0.004768  0.286768  1.000000
```



From the plot above we can observe that only number of nodes feature is slightly related to the survival status of patients as its covariance value is around 0.3. Rest two features are not relatable as they have their covariance value nearly equal to 0.

Took some help from GeekforGeeks for this particular plot. https://www.geeksforgeeks.org/