ASSIGNMENT 1 - EDA

In [1]:
```
#importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
#importing the dataset
df = pd.read_csv('haberman.csv')
df.head(5)   #to see top 5 rows in the dataset
```

Out[2]:

	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 [3]:
```
df.tail(5)
```

Out[3]:

	age	year	nodes	status
301	75	62	1	1

	age	year	nodes	status
302	76	67	0	1
303	77	65	3	1
304	78	65	1	2
305	83	58	2	2

In [4]:
```
df['status'].unique()
```

Out[4]:
```
array([1, 2], dtype=int64)
```

The dataset contains 3 features as age, year and nodes and it is an binary classification problem and the classes are represented in 'status' column in the dataset where 1 represent as patient survived and 2 represents as not survived

In [5]:
```
#to see the statistics on the dataset
df.describe()
```

Out[5]:

	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

From this there are totally 306 data points present in the dataset

In [6]:
```
#dimension of the dataset
df.shape
```

Out[6]:
```
(306, 4)
```

In [7]:
```
#information about the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306 entries, 0 to 305
Data columns (total 4 columns):
age       306 non-null int64
year      306 non-null int64
nodes     306 non-null int64
status    306 non-null int64
dtypes: int64(4)
memory usage: 9.6 KB
```

From the above result we can see that there is no missing values present in the dataset and all having the same datatype

In [8]:
```
df['status'].value_counts()
```

Out[8]:
```
1    225
2     81
Name: status, dtype: int64
```

It looks that the number of people survived after the operation is 225 and not survived is 81

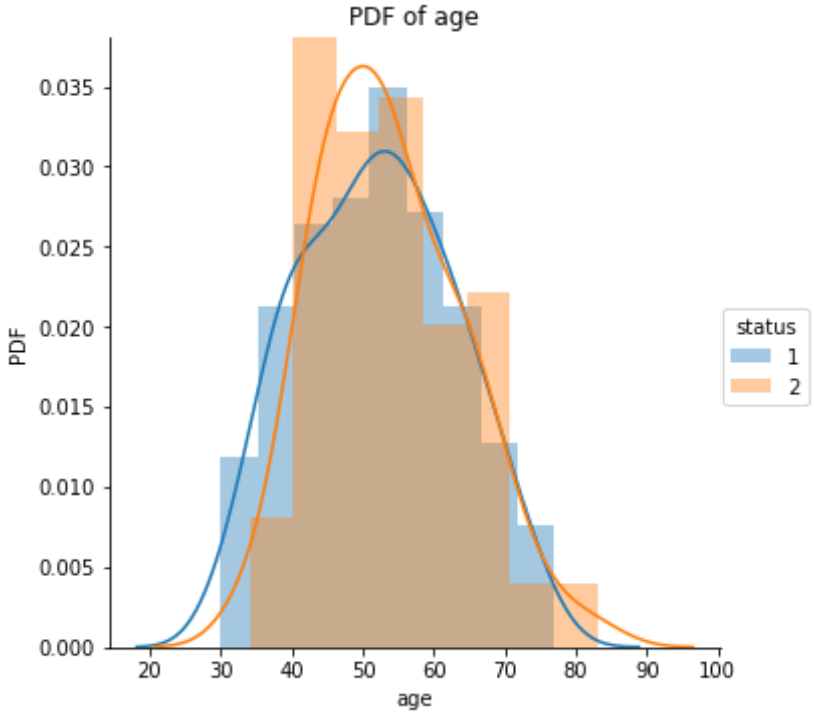
Objective:

We need to find the pattern in the data given whether the patients survived or not.

Univariate Analysis

1. PDF:

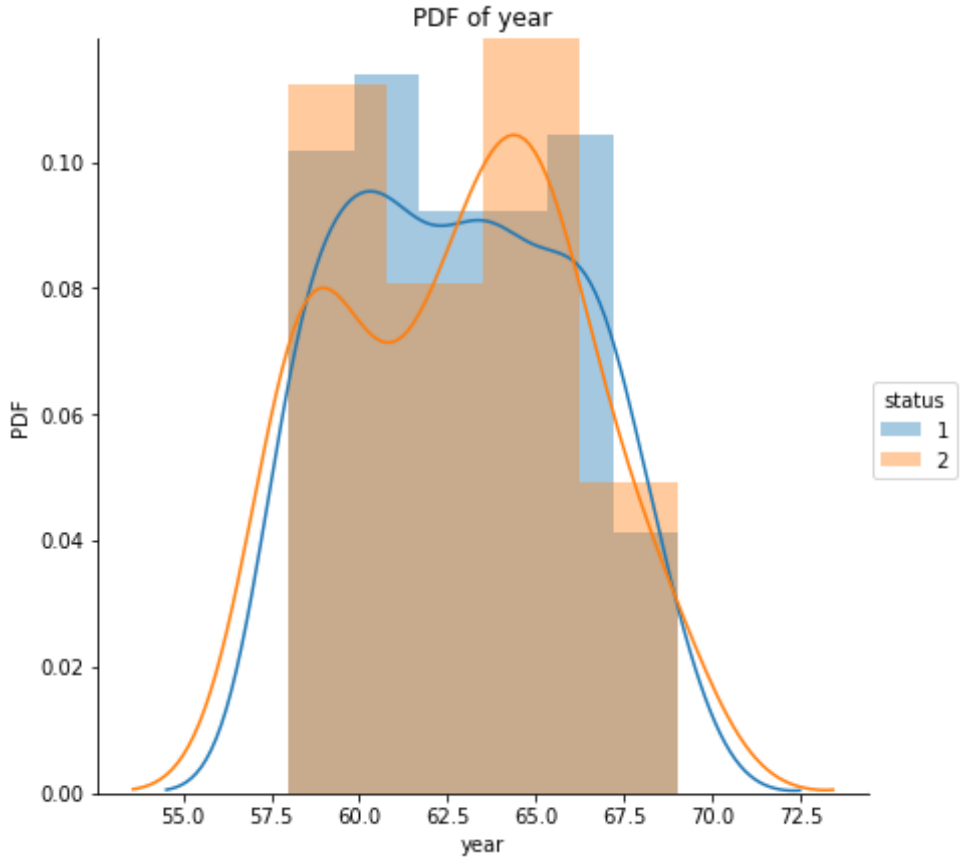
In [11]:
```
#Age
sns.FacetGrid(df, hue='status', size=5).map(sns.distplot, 'age').add_le
gend()
plt.title('PDF of age')
plt.ylabel('PDF')
plt.show()
```



The data of age with status 2 has a symmetric curve and it looks like normally distributed and data with status 1 looks spreaded more than with status 1 and its not perfectly symmetric. THe

data looks like most of the patiets who survived had done the operation when they were around 50-60. The patients age who didn't survive after the operation is around 45.

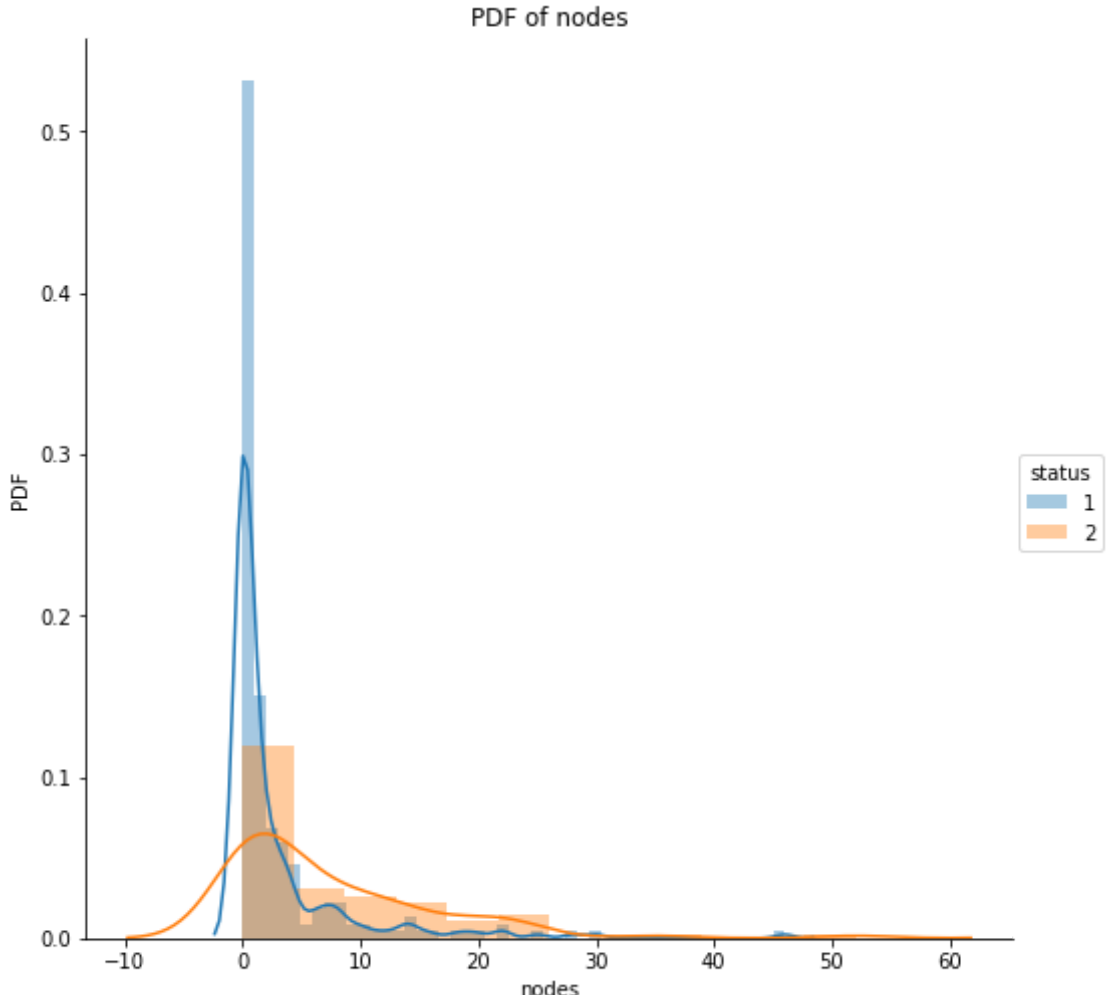
In [12]:
```
#year
sns.FacetGrid(df, hue='status', size=6).map(sns.distplot, 'year').add_l
egend()
plt.title('PDF of year')
plt.ylabel('PDF')
plt.show()
```



From this plot we can see that the patients who survived most during the years 60-62.5 and most

of them didn't survive during the years 64-66 (may be they didn't operated by the best of the doctors during that time).

In [13]:
```
#nodes
sns.FacetGrid(df, hue='status', size=7).map(sns.distplot, 'nodes').add_
legend()
plt.title('PDF of nodes')
plt.ylabel("PDF")
plt.show()
```



The data is positively skewed and it looks like the patients having less nodes are survived more and they didn't survive who had more nodes

2.CDF:

In [14]:
```
df_status_1 = df.loc[df['status'] == 1]
df_status_2 = df.loc[df['status'] == 2]
print(df_status_1.head())
print(df_status_2.head())
```

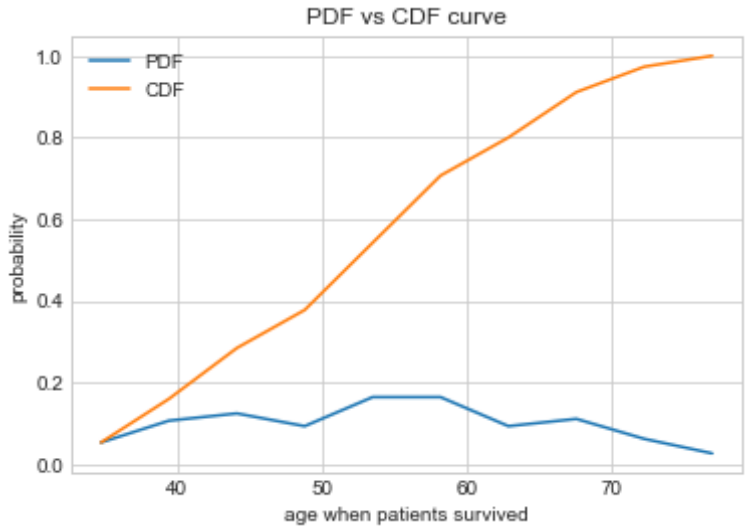
```
   age  year  nodes  status
0   30    64      1       1
1   30    62      3       1
2   30    65      0       1
3   31    59      2       1
4   31    65      4       1
    age  year  nodes  status
7    34    59      0       2
8    34    66      9       2
24   38    69     21       2
34   39    66      0       2
43   41    60     23       2
```

In [57]:
```
#cdf of age
counts, bin_edges = np.histogram(df_status_1['age'], bins=10, density=True)
pdf = counts/sum(counts)
print(pdf)
print(bin_edges)

#compute cdf
cdf = np.cumsum(pdf)
plt.title('PDF vs CDF curve')
plt.xlabel('age when patients survived')
plt.ylabel('probability')
```

```
plt.plot(bin_edges[1:], pdf, label='PDF')
plt.plot(bin_edges[1:], cdf, label='CDF')
plt.legend()
plt.show()
```

```
[0.05333333 0.10666667 0.12444444 0.09333333 0.16444444 0.16444444
 0.09333333 0.11111111 0.06222222 0.02666667]
[30.  34.7 39.4 44.1 48.8 53.5 58.2 62.9 67.6 72.3 77. ]
```



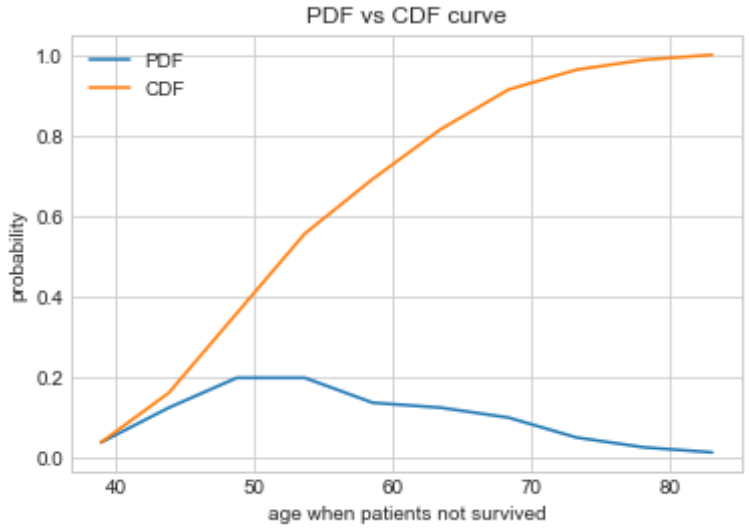
From this cdf graph, there are 60% of people who survived who age less than or equal to 55.

In [59]:
```
#cdf of age
counts, bin_edges = np.histogram(df_status_2['age'], bins=10, density=True)
pdf = counts/sum(counts)
print(pdf)
print(bin_edges)

#compute cdf
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:], pdf, label='PDF')
plt.plot(bin_edges[1:], cdf, label='CDF')
```

```
plt.xlabel('age when patients not survived')
plt.ylabel('probability')
plt.title('PDF vs CDF curve')
plt.legend()
plt.show()
```

```
[0.03703704 0.12345679 0.19753086 0.19753086 0.13580247 0.12345679
 0.09876543 0.04938272 0.02469136 0.01234568]
[34.  38.9 43.8 48.7 53.6 58.5 63.4 68.3 73.2 78.1 83. ]
```



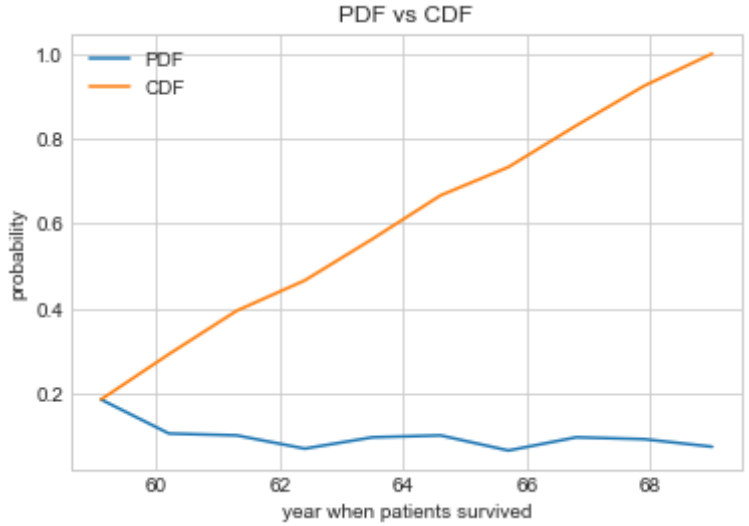
From this graph, the 80% of patients who didn't survive have the age less than or equal to 63.

In [60]:
```
# 2. cdf of year
counts, bin_edges = np.histogram(df_status_1['year'], density=True)
print(counts)
pdf = counts/sum(counts)
print(pdf)
print(bin_edges)

#compute cdf
cdf = np.cumsum(pdf)
plt.title('PDF vs CDF')
```

```
plt.xlabel('year when patients survived')
plt.ylabel('probability')
plt.plot(bin_edges[1:], pdf, label='PDF')
plt.plot(bin_edges[1:], cdf, label='CDF')
plt.legend()
plt.show()
```

```
[0.16969697 0.0969697  0.09292929 0.06464646 0.08888889 0.09292929
 0.06060606 0.08888889 0.08484848 0.06868687]
[0.18666667 0.10666667 0.10222222 0.07111111 0.09777778 0.10222222
 0.06666667 0.09777778 0.09333333 0.07555556]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
```



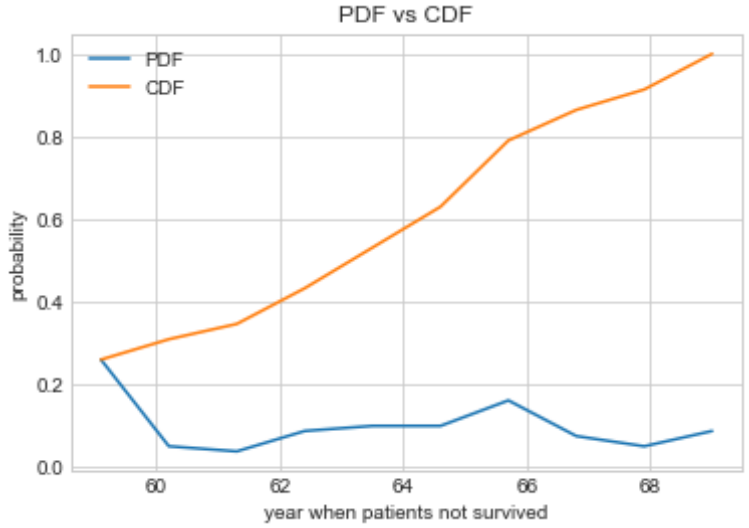
From this graph, there 80% of patients who survived who operated less than 67's year.

In [61]:
```
count, bin_edges = np.histogram(df_status_2['year'], density=True)
pdf = count/sum(count)
print(pdf)
print(bin_edges)

#compute cdf
cdf = np.cumsum(pdf)
```

```
plt.plot(bin_edges[1:], pdf, label='PDF')
plt.plot(bin_edges[1:], cdf, label='CDF')
plt.title('PDF vs CDF')
plt.legend()
plt.xlabel('year when patients not survived')
plt.ylabel('probability')
plt.show()
```

```
[0.25925926 0.04938272 0.03703704 0.08641975 0.09876543 0.09876543
 0.16049383 0.07407407 0.04938272 0.08641975]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
```



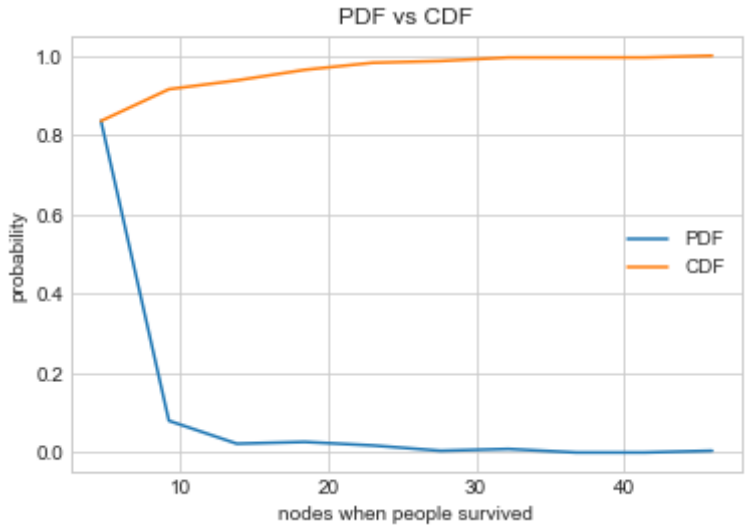
From this graph, most of the patients not survived during the year around 60's but the 25% percentage of patients not survived when the years less than 60's

In [63]:
```
# 3. cdf of nodes
counts, bin_edges = np.histogram(df_status_1['nodes'], density=True)
pdf = counts/sum(counts)
print(pdf)
print(bin_edges)

#compute cdf
```

```
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:], pdf, label='PDF')
plt.plot(bin_edges[1:], cdf, label='CDF')
plt.title('PDF vs CDF')
plt.xlabel('nodes when people survived')
plt.ylabel('probability')
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. ]
```



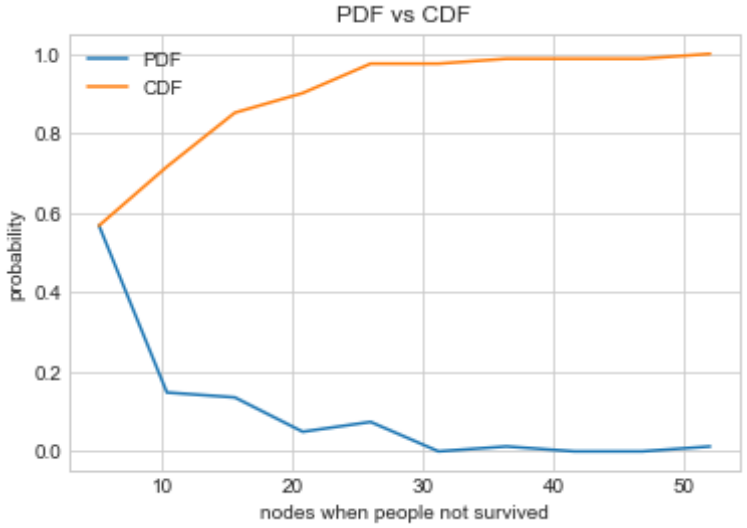
80% of the people survived when they have the nodes less 3.

In [64]:
```
counts, bin_edges = np.histogram(df_status_2['nodes'], density=True)
pdf = counts/sum(counts)
print(pdf)
print(bin_edges)

#compute cdf
cdf = np.cumsum(pdf)
```

```
plt.plot(bin_edges[1:], pdf, label='PDF')
plt.plot(bin_edges[1:],cdf, label='CDF')
plt.title('PDF vs CDF')
plt.legend()
plt.xlabel('nodes when people not survived')
plt.ylabel('probability')
plt.show()
```

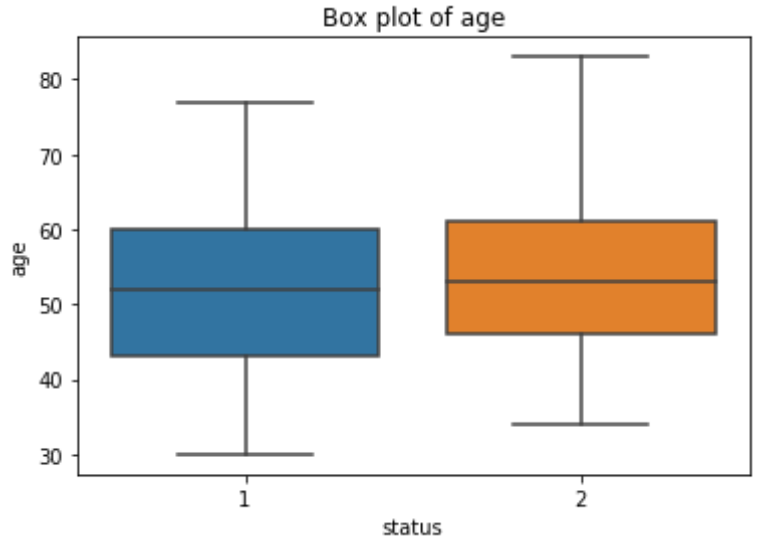
```
[0.56790123 0.14814815 0.13580247 0.04938272 0.07407407 0.
 0.01234568 0.         0.         0.01234568]
[ 0.   5.2 10.4 15.6 20.8 26.  31.2 36.4 41.6 46.8 52. ]
```



From this graph, 70% of the patients not survived when they nodes less than 5

3. Box Plots

In [26]:
```
#age
sns.boxplot(x='status', y='age', data=df)
plt.title('Box plot of age')
plt.show()
```



From the boxplot , we can see that mean age of people who survived is 52 and not survived is 55

In [46]:
```
#90th percentile who survived
np.percentile(df_status_1['age'], 25)
```

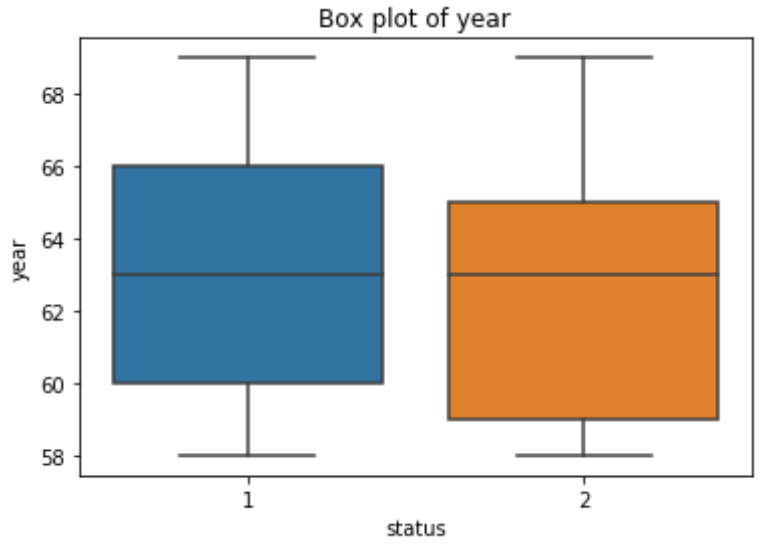
Out[46]: 43.0

In [45]:
```
#90th percentile who didn't survive
np.percentile(df_status_2['age'], 25)
```

Out[45]: 46.0

The boxplot and the above code shows that 25% of patients who survived who had age less than 43 and not survived is 46

In [27]:
```
#year
sns.boxplot(x='status', y='year', data=df)
plt.title('Box plot of year')
plt.show()
```



The above plot shows that most of the patients who didn't surive have done the operation less the year 63

In [56]:
```
np.percentile(df_status_1['year'], 75)
```

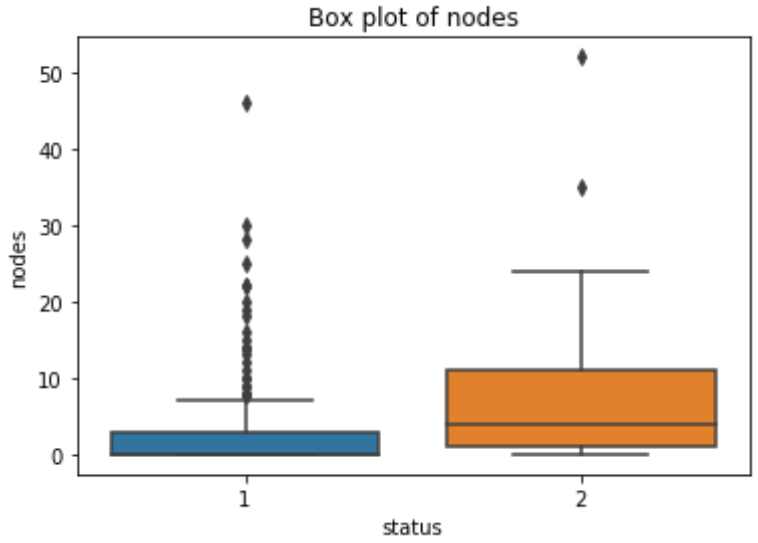
Out[56]: 66.0

In [57]:
```
np.percentile(df_status_2['year'],75)
```

Out[57]: 65.0

The boxplot and the above code shows that 75% of patients who survived who did the operation less than 66 and not survived is 67

In [28]:
```
#nodes
sns.boxplot(x='status', y='nodes', data=df)
plt.title('Box plot of nodes')
plt.show()
```



From this plot , we can see that there are more outliers present in the data especially the patients who survived

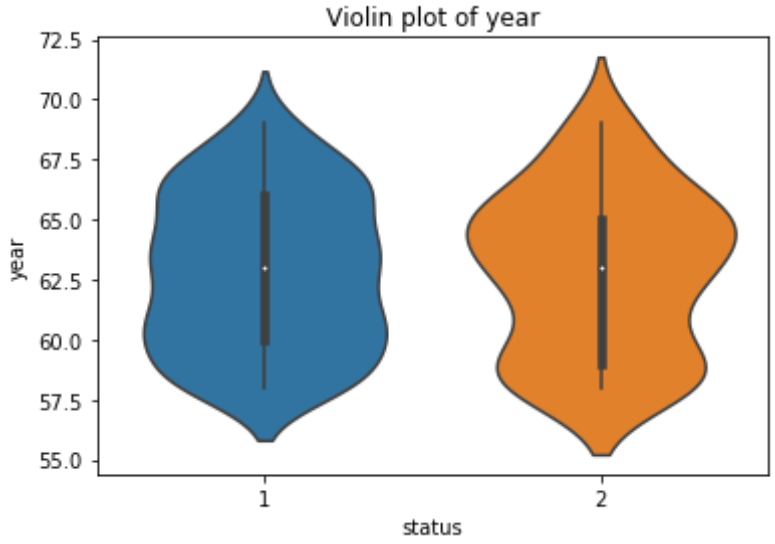
4. Violin Plot

In [29]:
```
#age
sns.violinplot(x='status', y='age', data=df, size=6)
plt.title('violin plot of age')
plt.legend()
plt.show()
```

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

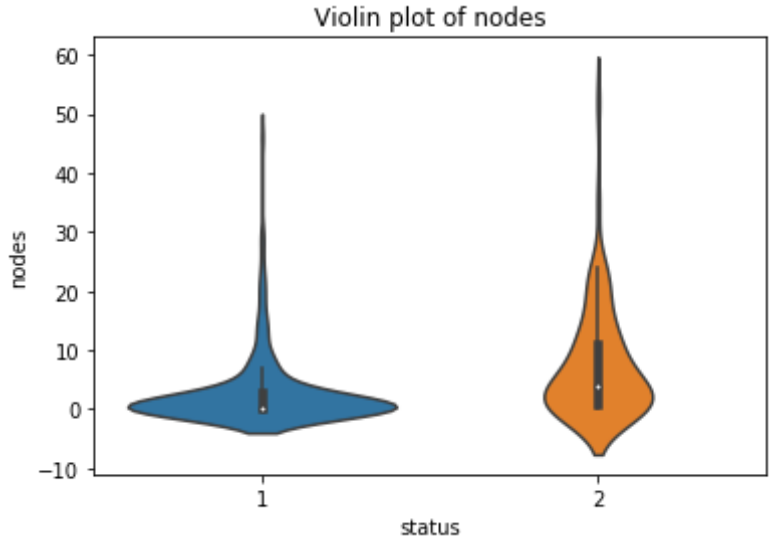
As we discussed earlier, the patients who didn't survive is skewed right side or positively skewed

In [31]:
```
#year
sns.violinplot(x='status', y='year', data=df, size=6)
plt.title('Violin plot of year')
plt.show()
```



The patients who didn't survive looks as a the data is not normally distributed

In [32]:
```
#nodes
sns.violinplot(x='status', y='nodes', data=df, size=6)
plt.title('Violin plot of nodes')
plt.show()
```



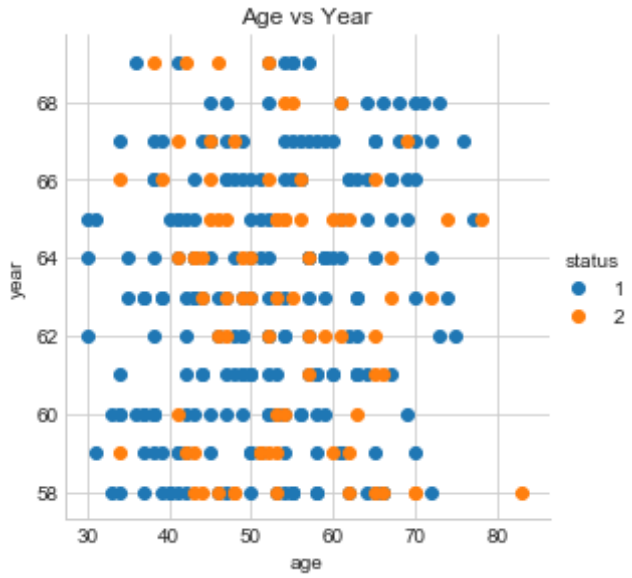
From this plot, the data is skewed for both the results

Bivariate Analysis:

1.Scatter plot

There are three features present in the data and the no of combinations we can plot is 3C2 (i.e 3)

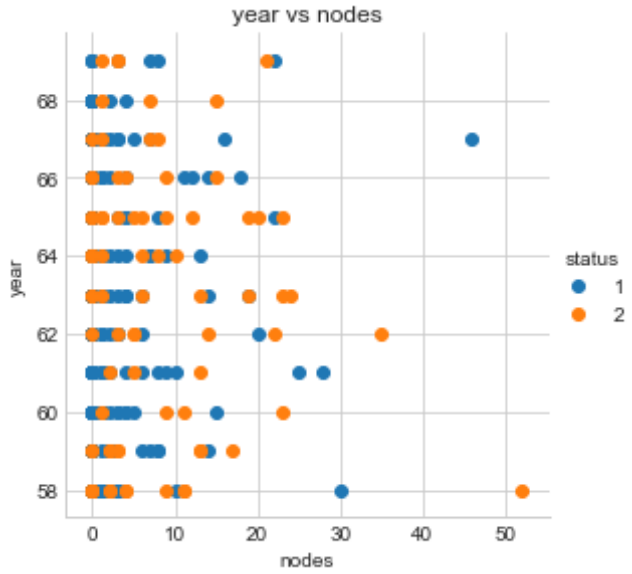
In [33]:
```
# age vs year
sns.set_style('whitegrid')
sns.FacetGrid(df, hue='status', size=4).map(plt.scatter, 'age', 'year')
.add_legend()
plt.title('Age vs Year')
plt.show()
```



We can't come to any conclusions with this plot because the data spread everywhere and there is no pattern in the data

In [34]:
```
#age vs nodes
sns.set_style('whitegrid')
sns.FacetGrid(df, hue='status', size=4).map(plt.scatter, 'nodes', 'age'
).add_legend()
plt.title('Age vs nodes')
plt.show()
```

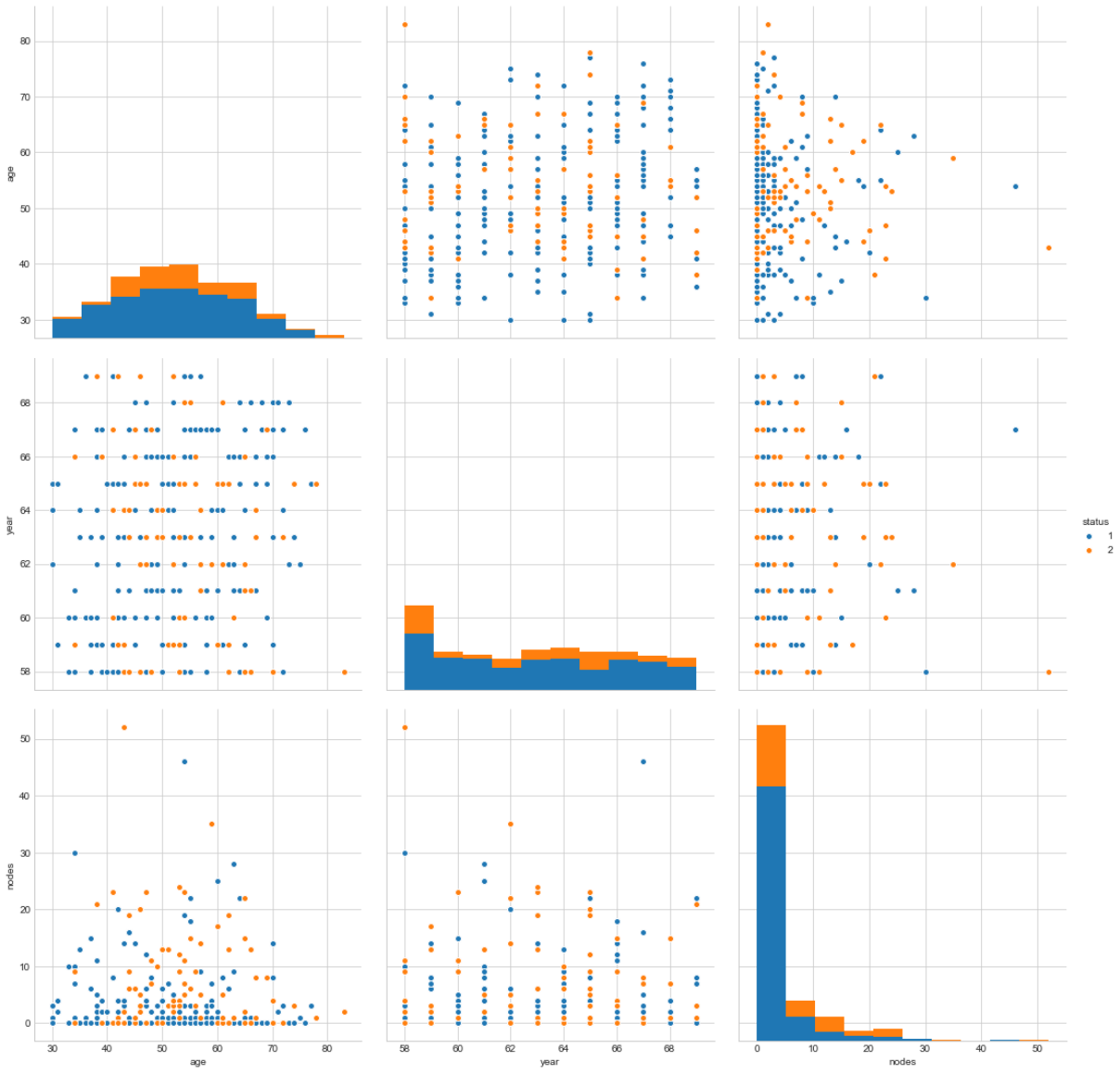
In [35]:
```
#nodes vs years
sns.set_style('whitegrid')
sns.FacetGrid(df, hue='status', size=4).map(plt.scatter, 'nodes', 'yea
r').add_legend()
plt.title('year vs nodes')
plt.show()
```



From the above all scatter plots there is no pattern in the data

2. Pair Plot

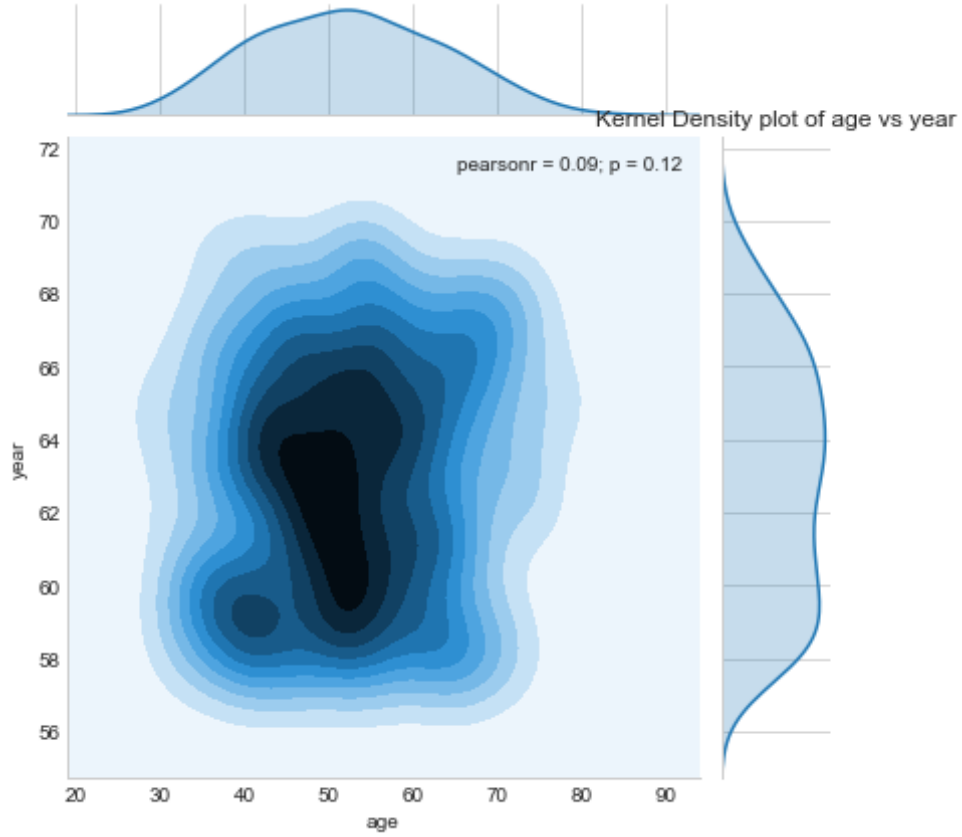
In [4]:
```
plt.close()
sns.set_style('whitegrid')
sns.pairplot(data=df, vars=['age','year','nodes'], hue = 'status', size
=5);
plt.show()
```



Since it is a simulation of scatter plot there is not much information in the pairplot.

3. Jointplot

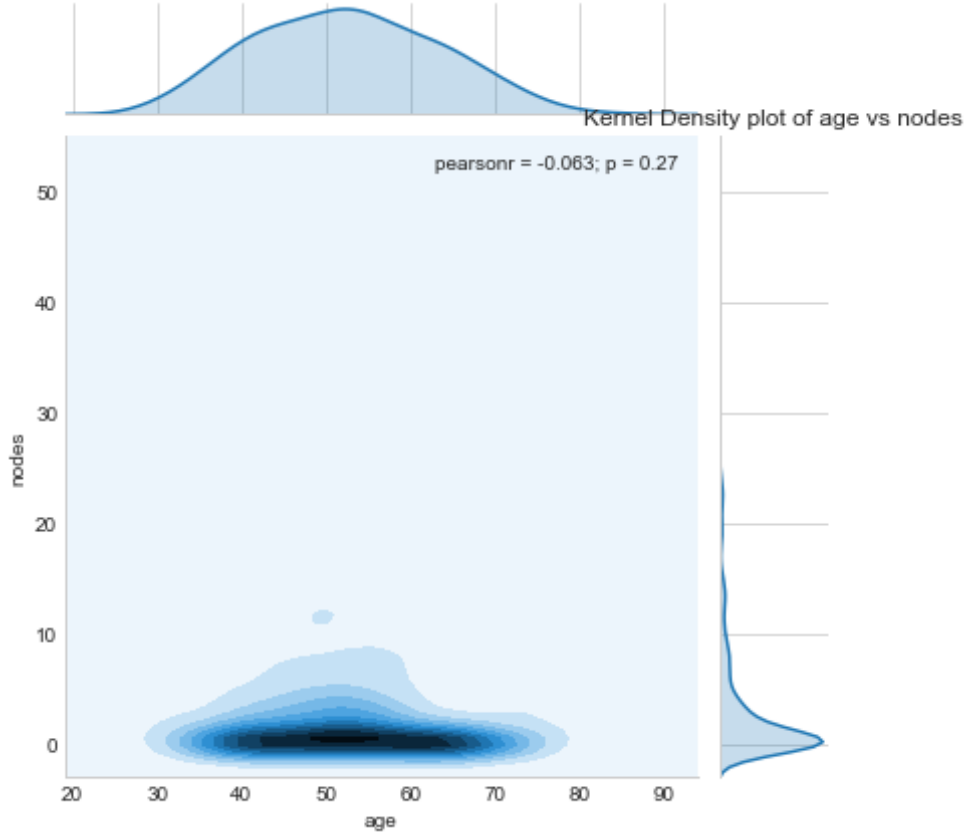
In [40]:
```
#age vs year
sns.set_style('whitegrid')
sns.jointplot(x='age', y='year', data=df, kind='kde')
plt.title('Kernel Density plot of age vs year')
plt.show()
```



From this plot we can see that the most of the data has the age around 50- 60 and with the year 60-64

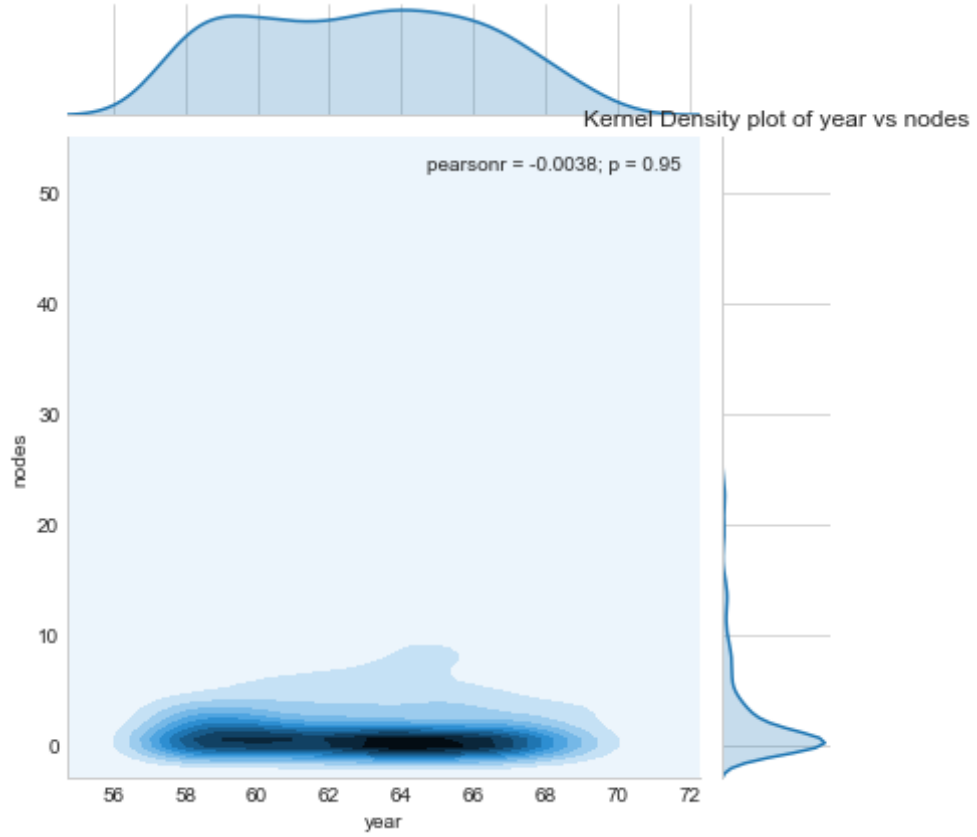
In [41]:
```
#age vs nodes
```

```
sns.set_style('whitegrid')
sns.jointplot(x='age', y='nodes',data=df, kind='kde')
plt.title('Kernel Density plot of age vs nodes')
plt.show()
```



From this plot, we can see that the most of the data lies in the range of 50-60 in age and the nodes is around 0-1

In [42]:
```
#year vs nodes
sns.set_style('whitegrid')
sns.jointplot(x='year', y='nodes', data=df, kind='kde')
plt.title('Kernel Density plot of year vs nodes')
plt.show()
```



From this plot, we can see that most of the data is around 64-66 and the nodes is same as 0-1 as we previously mentioned