Objective

- The objective of Exploratory data analysis for Haberman dataset is to classify survival status of new patient based on given feature i.e. age,nodes,year.
- Finding important feature of given dataset based on analysis to reach to the conclusion

In [17]:
```
import warnings

warnings.filterwarnings("ignore")
```

In [1]:
```
#Import Libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sb
```

In [2]:
```
#Loading Dataset

haberman=pd.read_csv("C:/Users\91888\Desktop\Applied AI\Assignment\haberman.csv")
```

In [3]:
```
#It will display first 5 rows of dataset

haberman.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]:
```
#Names of columns in dataset
print(haberman.columns)
```

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

In [5]:
```
#How many rows and columns present
print(haberman.shape)
```

```
(306, 4)
```

In [6]:
```
#How many datapoints present for each status
haberman["status"].value_counts()
```

Out[6]:
```
1    225
2     81
Name: status, dtype: int64
```

In [53]:
```
#Basic info of dataset like datatype,no of entries,memory used
print(haberman.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
None
```

In [7]:
```
#Gives mean,std,min max values for all integer columns
print(haberman.describe())
```

```
              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
```

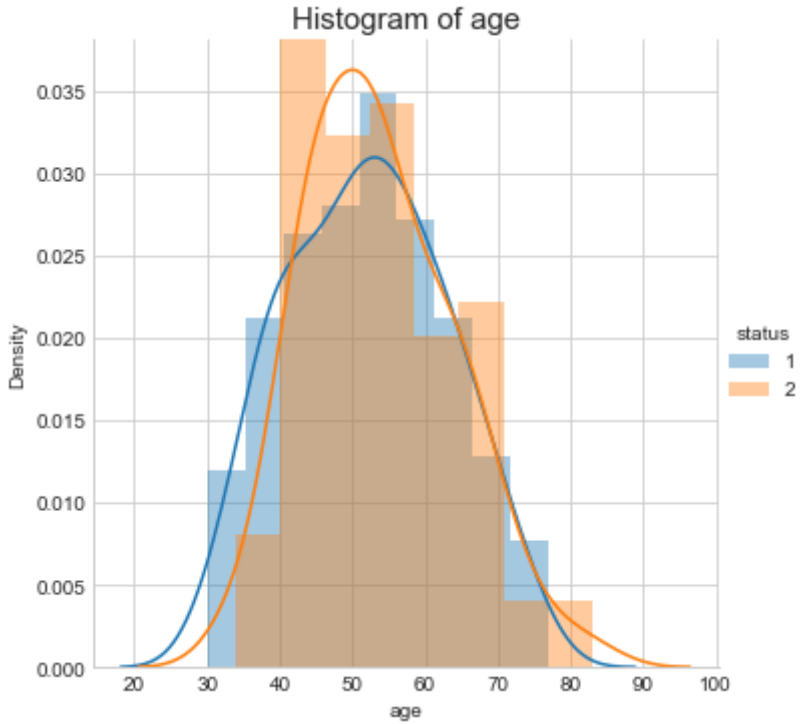
Information of dataset obtained from above data operations

- There are 4 columns i.e. age,year,nodes,status
- Total numbers of rows is 306 and columns is 4
- for status 1(long survival) 225 datapoints are present and for status 2(short survival) 81 are datapoints present
- Data is slightly imbalanced
- Data type of all four attribute is integer
- Mean age of person is 52
- maximun number of nodes found is 52
- About 25% of people have no nodes detected

UNIVARIATE ANALYSIS

Histogram

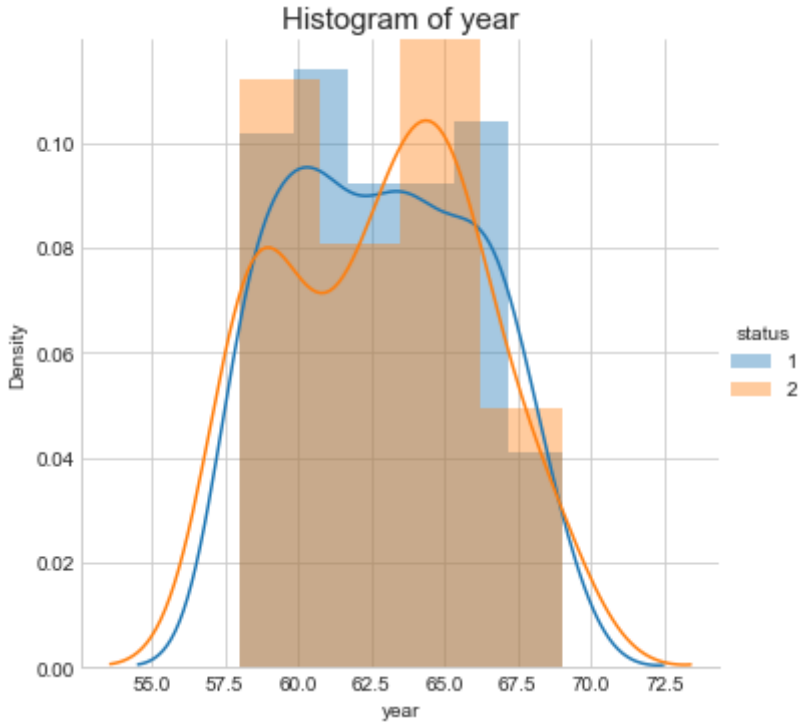
In [64]:
```
import seaborn as sb
sb.FacetGrid(haberman,hue="status",size=5)\
  .map(sb.distplot,"age")\
  .add_legend()
plt.ylabel("Density")
plt.title("Histogram of age",fontsize=15)

plt.show()
```



Observation

- Overlapping is oserved in major part
- we can not decide survival rate by just considering age of person

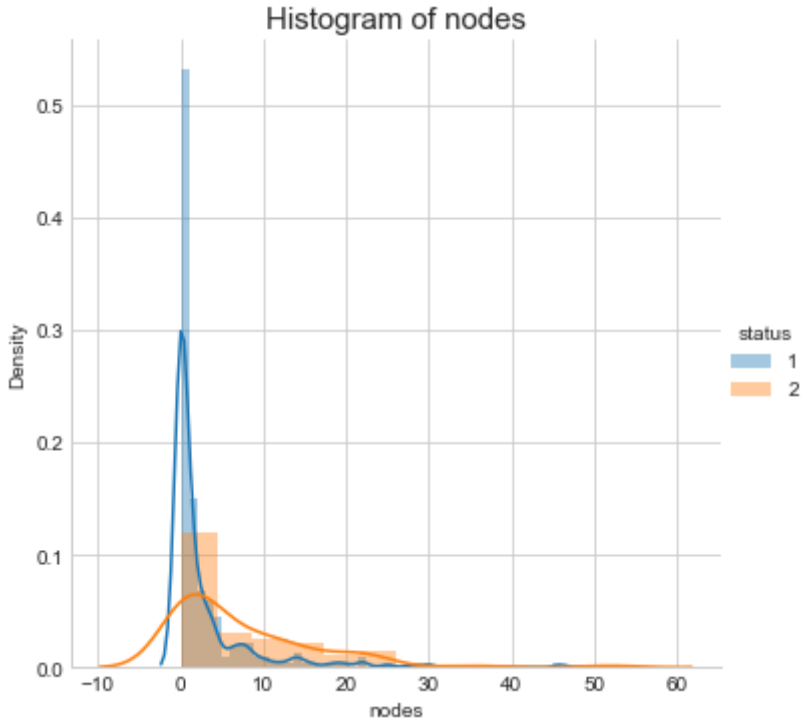
In [63]:
```
sb.FacetGrid(haberman,hue="status",size=5)\
  .map(sb.distplot,"year")\
  .add_legend()
plt.ylabel("Density")
plt.title("Histogram of year",fontsize=15)
plt.show()
```



Observation

- Here also overlapping is observed in major area
- There are more unsucessful operations between 1962 to 1966 as number of death is high
- SO the year alone can not be the good parameter to determine patients survival rate

In [62]:
```
sb.FacetGrid(haberman,hue="status",size=5)\
  .map(sb.distplot,"nodes")\
  .add_legend()
plt.ylabel("Density")
plt.title("Histogram of nodes",fontsize=15)
plt.show()
```



Observation

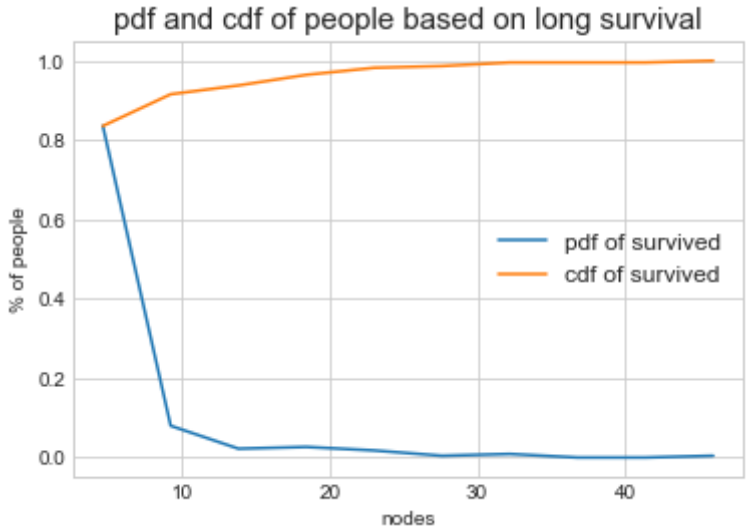
- If the number of nodes are high i.e. more than 23 then patient then there are very less chance of surviving
- People with 0 nodes are more likely to survive
- Also there is more overlapping when number of nodes are within 1 to 10
- so it is difficult to decide survival rate by using only nodes

PDF and CDF

In [11]:
```
import numpy as np
haberman_one=haberman.loc[haberman["status"]==1]
haberman_two=haberman.loc[haberman["status"]==2]
```

In [73]:
```
counts,bin_edges=np.histogram(haberman_one["nodes"],bins=10,density=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.xlabel("nodes")
plt.ylabel("% of people")
plt.title("pdf and cdf of people based on long survival",fontsize=15)
plt.legend(['pdf of survived','cdf of survived'],fontsize=12)
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. ]
```



Observation

- There is 82% chance of survival if number of nodes are less than 5
- As number of nodes increases long survival rate decreases
- People have less chance of survival if number of nodes>30

In [74]:
```
counts,bin_edges=np.histogram(haberman_two["nodes"],bins=10,density=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.xlabel("nodes")
plt.ylabel("% of people")
plt.title("pdf and cdf of people based on short survival",fontsize=15)
plt.legend(['pdf of died','cdf of died'],loc="center",fontsize=12)
plt.show()
```

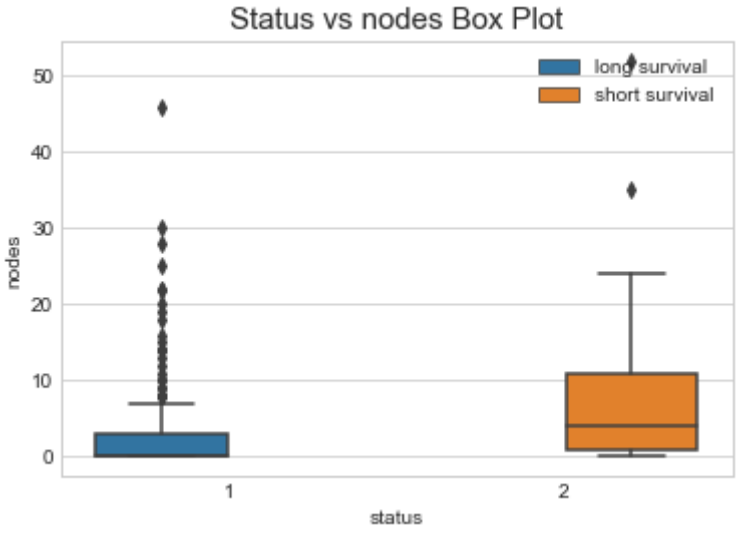
```
[0.56790123 0.14814815 0.13580247 0.04938272 0.07407407 0.
 0.01234568 0.         0.         0.01234568]
[ 0.   5.2 10.4 15.6 20.8 26.  31.2 36.4 41.6 46.8 52. ]
```

observation

- There are 58% of people who have short survival if number of nodes are less than 5.
- If the number of nodes are > 40 then there is 100% short survival

BOX PLOT

In [77]:
```
ax=sb.boxplot(x="status",y="nodes",data=haberman,hue="status")
plt.xlabel("status")
plt.ylabel("nodes")
plt.title("Status vs nodes Box Plot",fontsize=15)
handles, _ = ax.get_legend_handles_labels()
ax.legend(handles, ["long survival", "short survival"],loc="upper right")
plt.show()
```

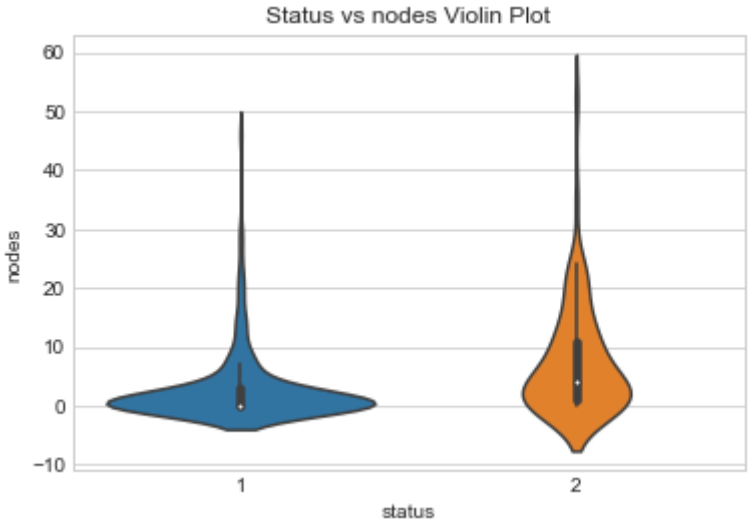


Observation

- For Long survival 25th percentile and 50th percentile are almost same
- Threshold for Long survival is 0 to 7
- For short survival 50th percentile is same as 75th percentile of long survival
- THreshold for short survival is 0 to 25

Violin Plot

In [48]:
```
sb.violinplot(x="status",y="nodes",data=haberman,size=8)
plt.xlabel("status")
plt.ylabel("nodes")
plt.title("Status vs nodes Violin Plot")
plt.show()
```



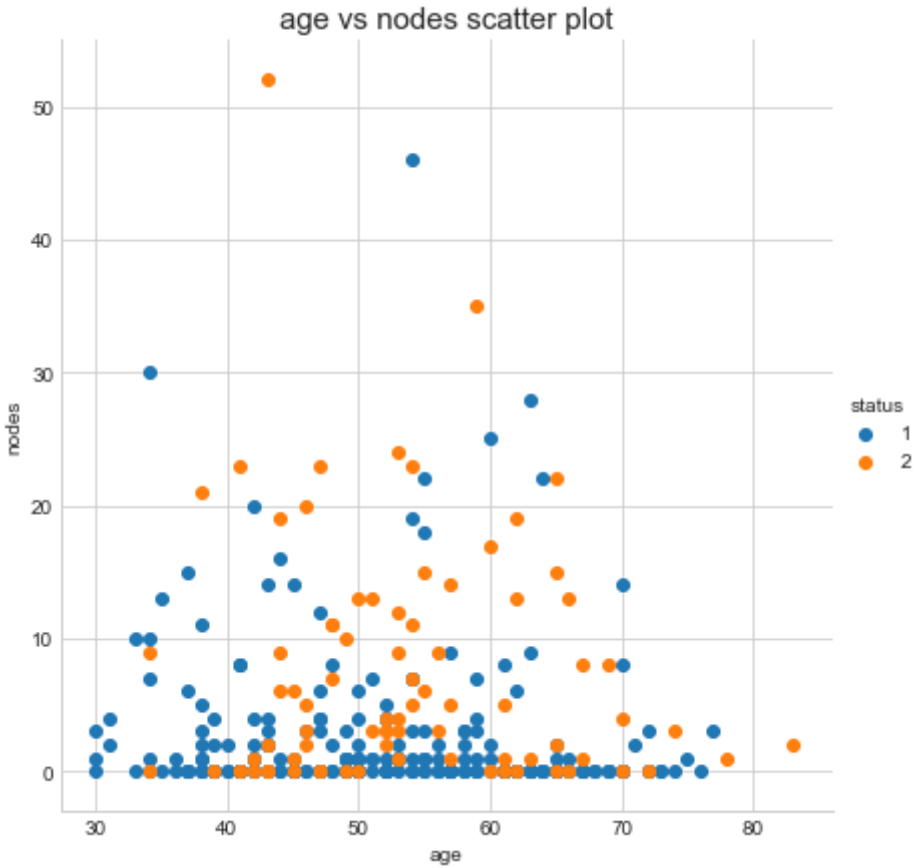
Observation

- Large percentage of people have long survival if the number of nodes are zero
- Also there is small percentage of people have short survival with 0 nodes
- less nodes can not always guarantee survival.

Bi-Variate Analysis

Scatter Plot

In [76]:
```
sb.set_style("whitegrid");
sb.FacetGrid(haberman, hue="status", size=6) \
   .map(plt.scatter, "age", "nodes") \
   .add_legend();
plt.xlabel("age")
plt.ylabel("nodes")
plt.title("age vs nodes scatter plot",fontsize=15)
plt.show();
```

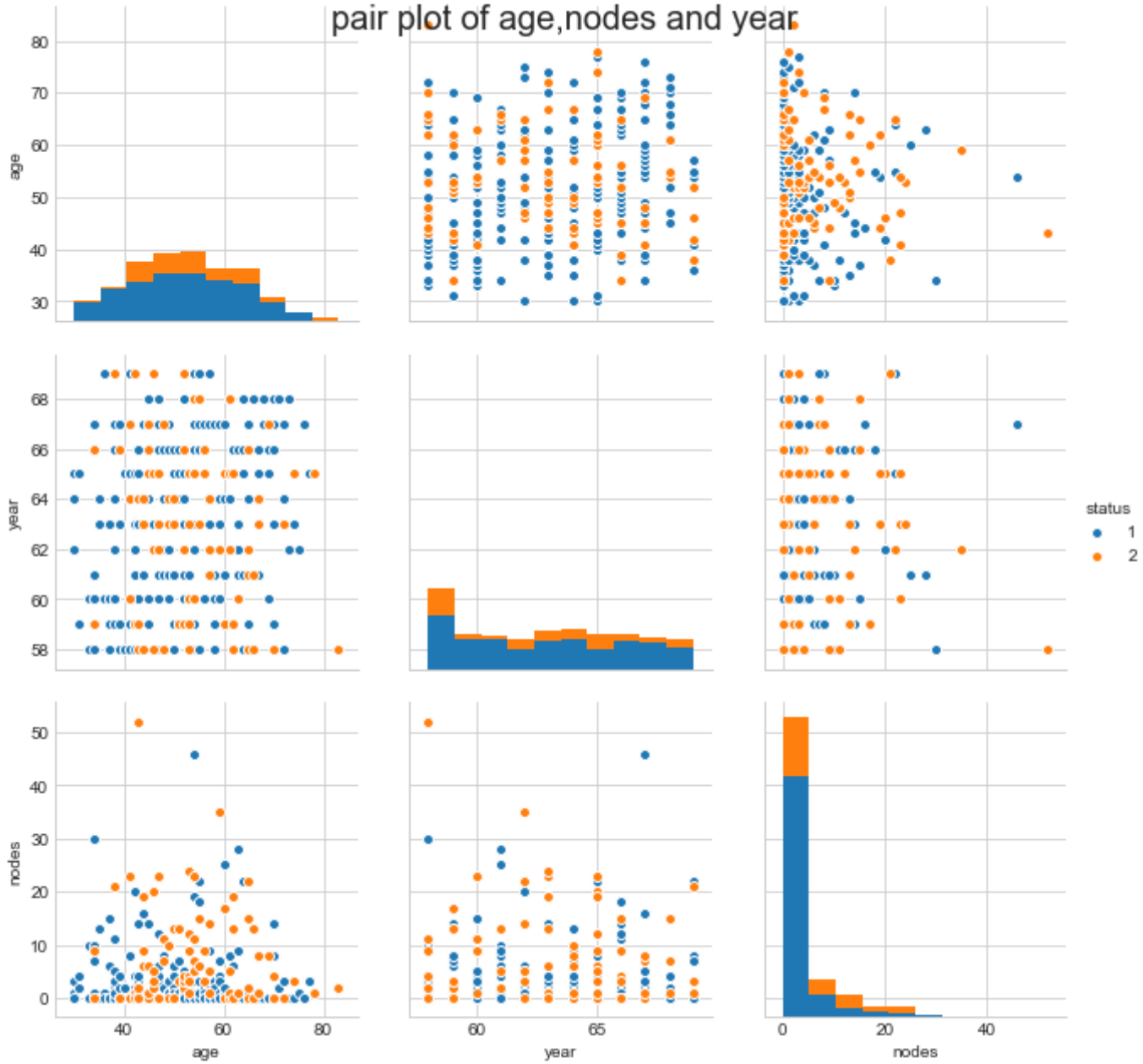


Observation

- There are hardly any patient who have nodes more than 25
- Patient having age more than 50 and nodes more than 10 are less likely to survive
- Patients with 0 nodes are more likely to long survive irrespective of their age

Pair Plot

In [78]:
```
sb.set_style("whitegrid");
sb.pairplot(haberman, hue="status", size=3,vars=['age','year','nodes']);
plt.suptitle("pair plot of age,nodes and year",fontsize=20)
plt.show()
```


pair plot of age,nodes and year
age
year
nodes
status
1
2

Observation

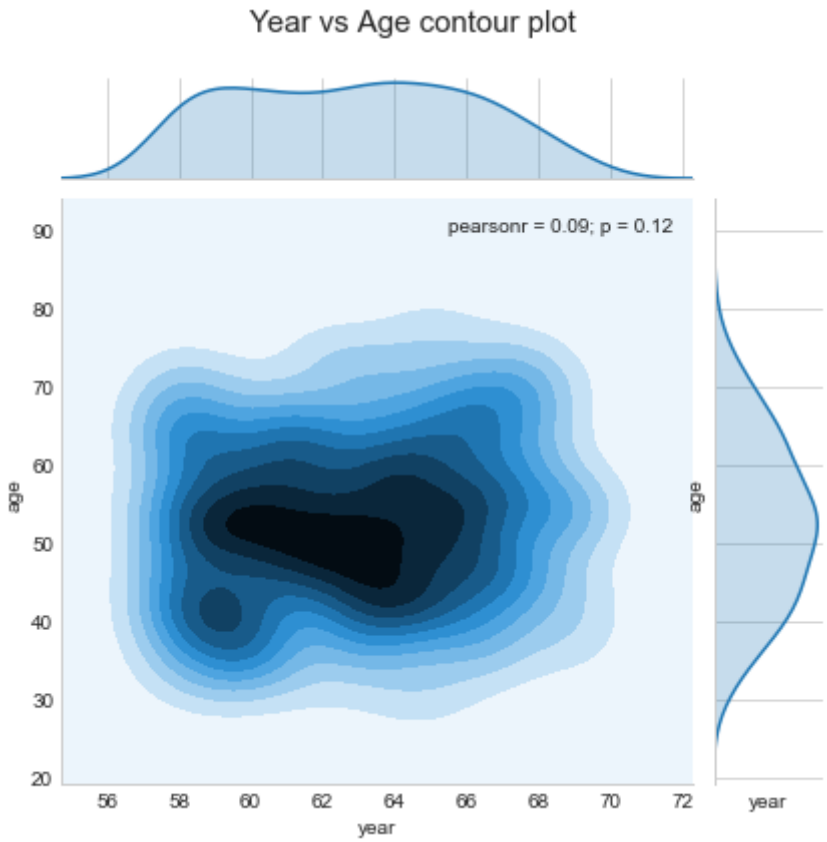
- Plots between age and nodes give distiguish points and it is better than other plots
- We can provide some conclusion based on this graph
- We can consider this two features for futher data operations.

Multivariate Analysis

Contour Plot

In [58]:
```
g=sb.jointplot(x="year",y="age",data=haberman,kind="kde")
plt.xlabel("year")
plt.ylabel("age")

plt.subplots_adjust(top=0.9)
g.fig.suptitle('Year vs Age contour plot',fontsize=15)
```

Out[58]: Text(0.5,0.98,'Year vs Age contour plot')



Observation

- Dark area represents major density and as density is getting low as area gets lighter.
- More operations are done on people in year 1960-1966 in age group 43-58

Conclusion

- Given dataset is imbalanced as the number of datapoints for each class are not same
- There is too much overlapping between data points hence it is difficult to classify
- Nodes is the important feature in dataset
- People with nodes >=1 more likely to die.
- There is good concentration of point when node is zero
- From scatter plot we conclude that people with 0 nodes are likely to survive irrespective of age
- Age is also important feature,people who have age <40 are more likely to long survive inspite of node>=1
- Patient having age more than 50 and nodes more than 10 are less likely to survive
- Patients who have nodes more than 24 are likely to die
- From the box plot we can conclude that large number of patients survived have 0 nodes or doesn't have it .
- Large number of operations are done between year 1960-1966
- There are more unsucessful operations between 1962 to 1966 as more number of patients died withing short period of time
- Patient’s age and operation year alone are not deciding factors for patients survival.
- Survival chances is inversely proportional to number of nodes present but zero nodes does not always guarantee survival.
- Classifying new patients survival status is difficult as data is imbalanced and also there is too much overlapping between data points

In [2]:
```
!jupyter nbconvert --to html EDAAssignment.ipynb
```

```
[NbConvertApp] Converting notebook EDAAssignment.ipynb to html
[NbConvertApp] Writing 590705 bytes to EDAAssignment.html
```