In [94]:
```python
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
```

In [95]:
```python
hbrmn = pd.read_csv("haberman.csv")    # loading data
print(hbrmn.shape)                     # print the shape to understand the row and columns
```

```
(306, 4)
```

In [98]:
```python
print(hbrmn.columns)                   # print the all features
```

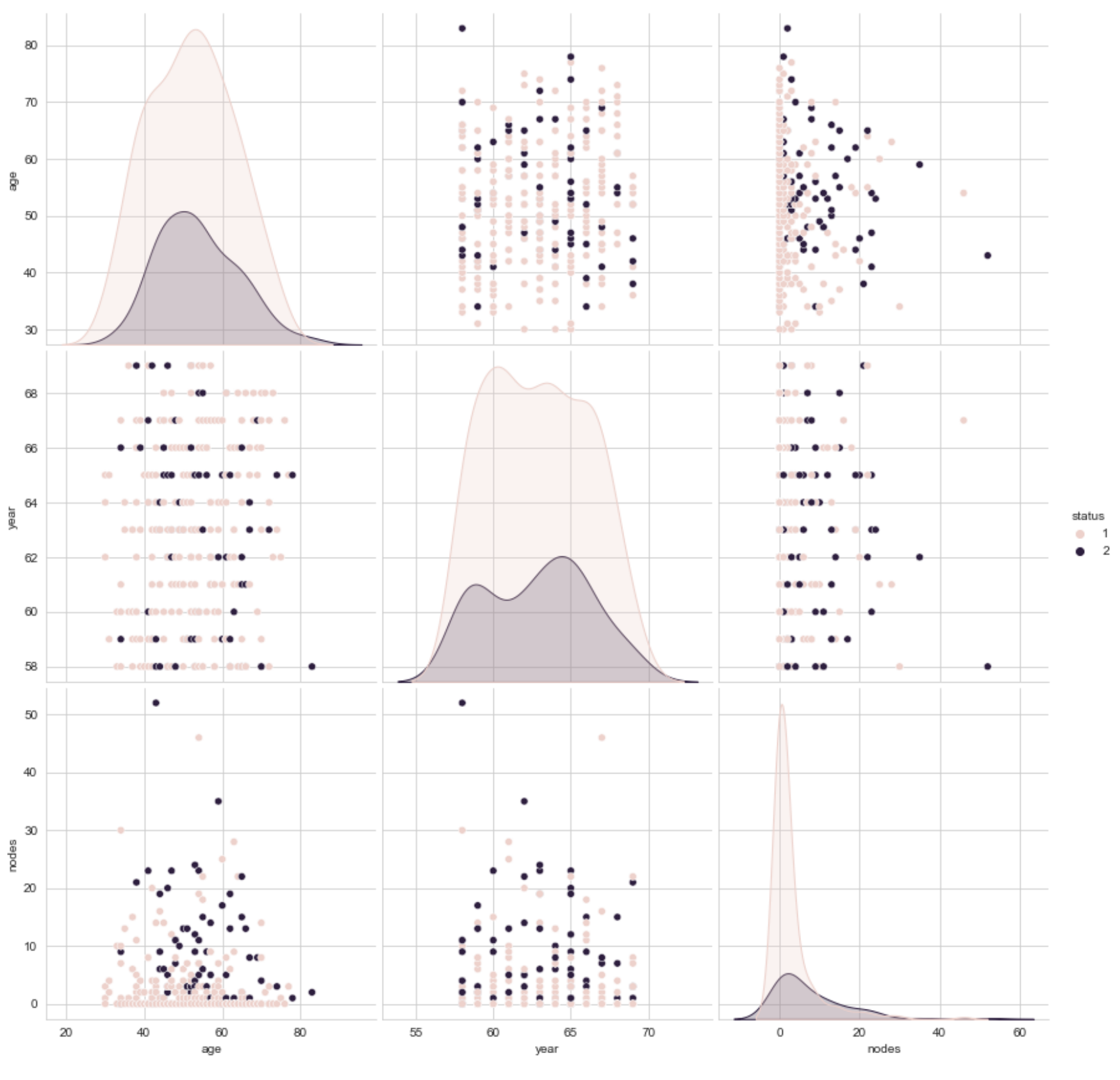
```
Index(['age', 'year', 'nodes', 'status'], dtype='object')
```

In [99]:
```python
hbrmn["status"].value_counts()         # count the no data in each class
```

Out[99]:
```
1    225
2     81
Name: status, dtype: int64
```

ploting the pairplot, it give graphical representation of each feature with respect to other one

In [100…
```python
sns.set_style("whitegrid");
sns.pairplot(hbrmn, hue="status", height=4);
# plt.title("pairplot of hbrmn")
plt.show()
```



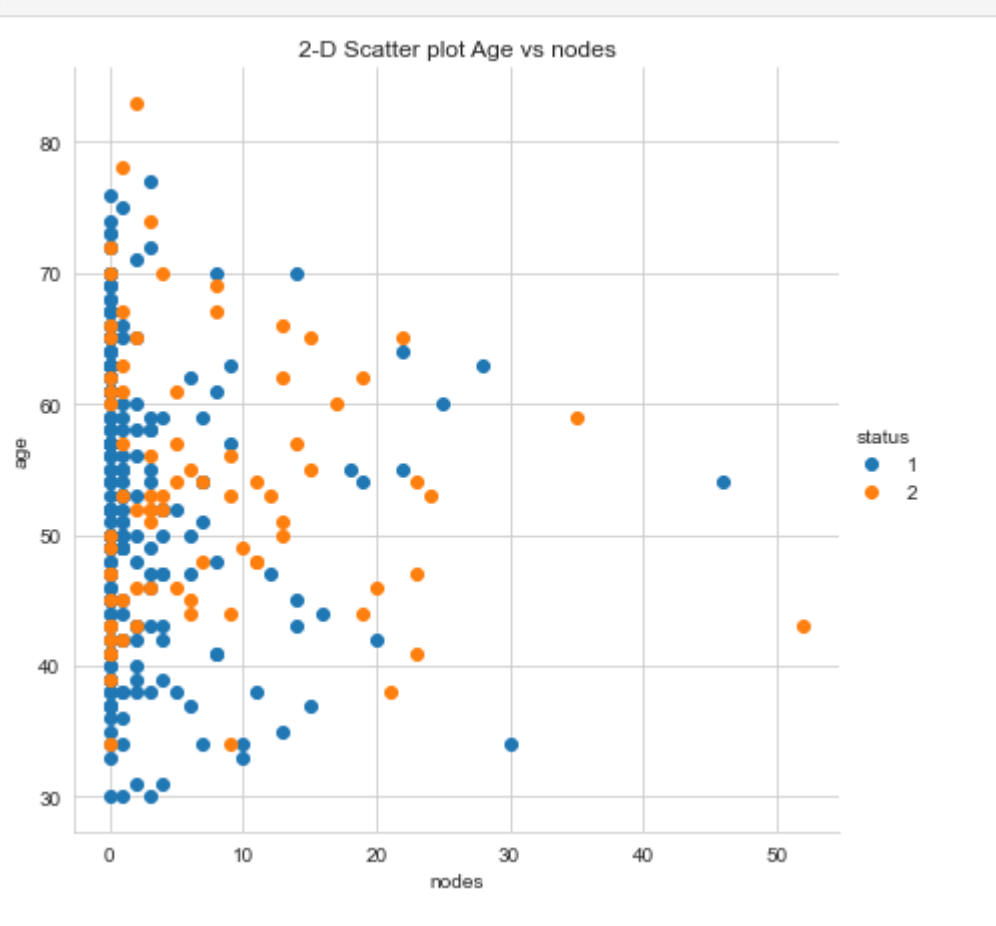
obervations - here we get total six graphs but we see go through above diagonal one

1 - age/year, age/nodes, year/nodes and as we see [age/node] is more clear than the other one

2 - in graph [age/year , year/nodes] we see more overlapping of the points

here we draw seperate graph for the [age/node]

In [101…
```python
sns.set_style("whitegrid");
sns.FacetGrid(hbrmn, hue="status", height=6) \
   .map(plt.scatter,"nodes", "age") \
   .add_legend();
plt.title("2-D Scatter plot Age vs nodes")
plt.show();
```

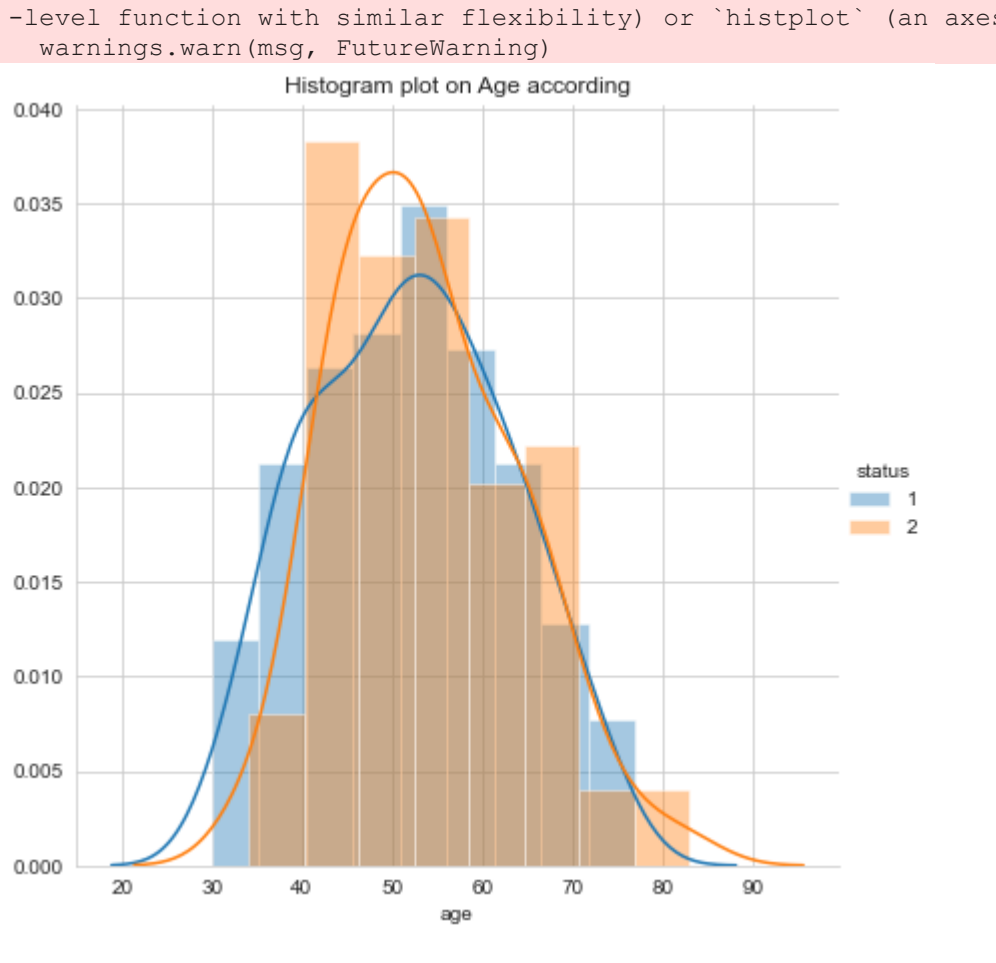


observation - here we clearly see understand the approx(range) of age and nodes according to the status

Histogram

In [102…
```python
sns.FacetGrid(hbrmn, hue="status", height=6) \
   .map(sns.distplot, "age") \
   .add_legend();
plt.title("Histogram plot on Age according")
plt.show();
```

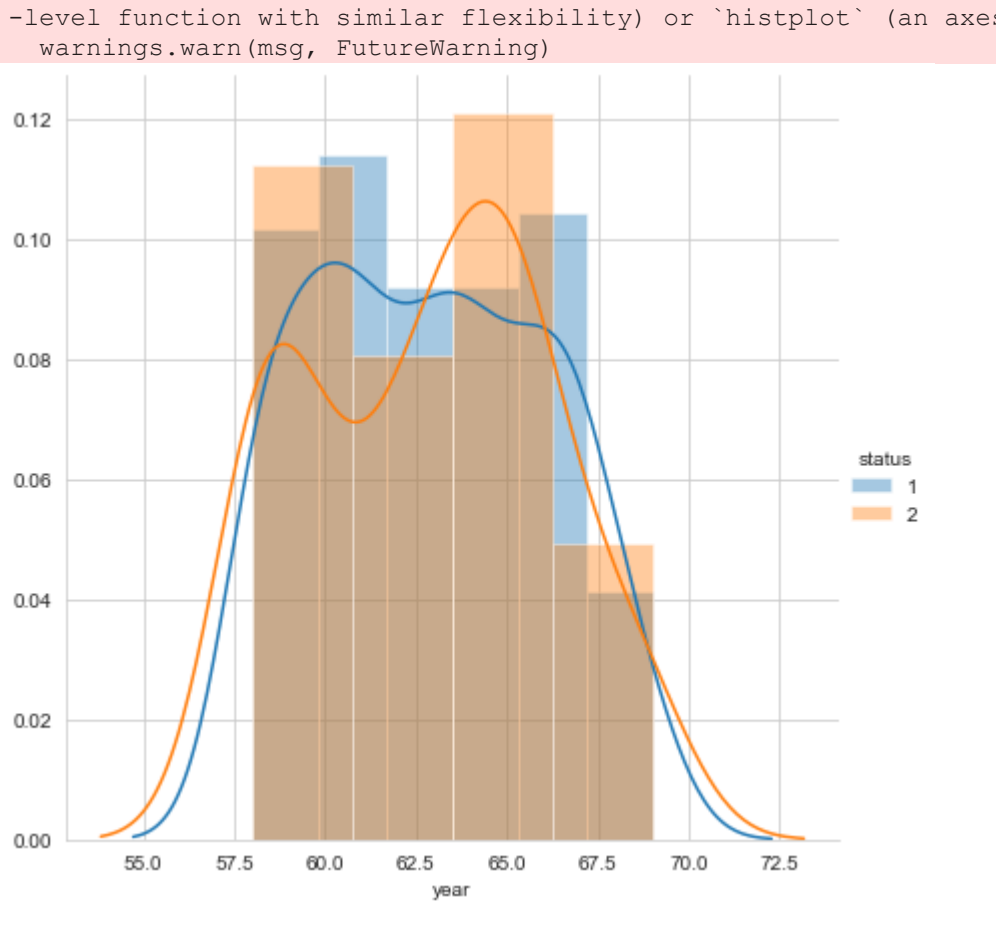
```
C:\Users\dev64\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprec
ated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\dev64\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprec
ated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



observation - very hard to understand , because mostly data overlap.... even the pdf conside

In [103…
```python
sns.FacetGrid(hbrmn, hue="status", height=6) \
   .map(sns.distplot, "year") \
   .add_legend();
plt.show();
```

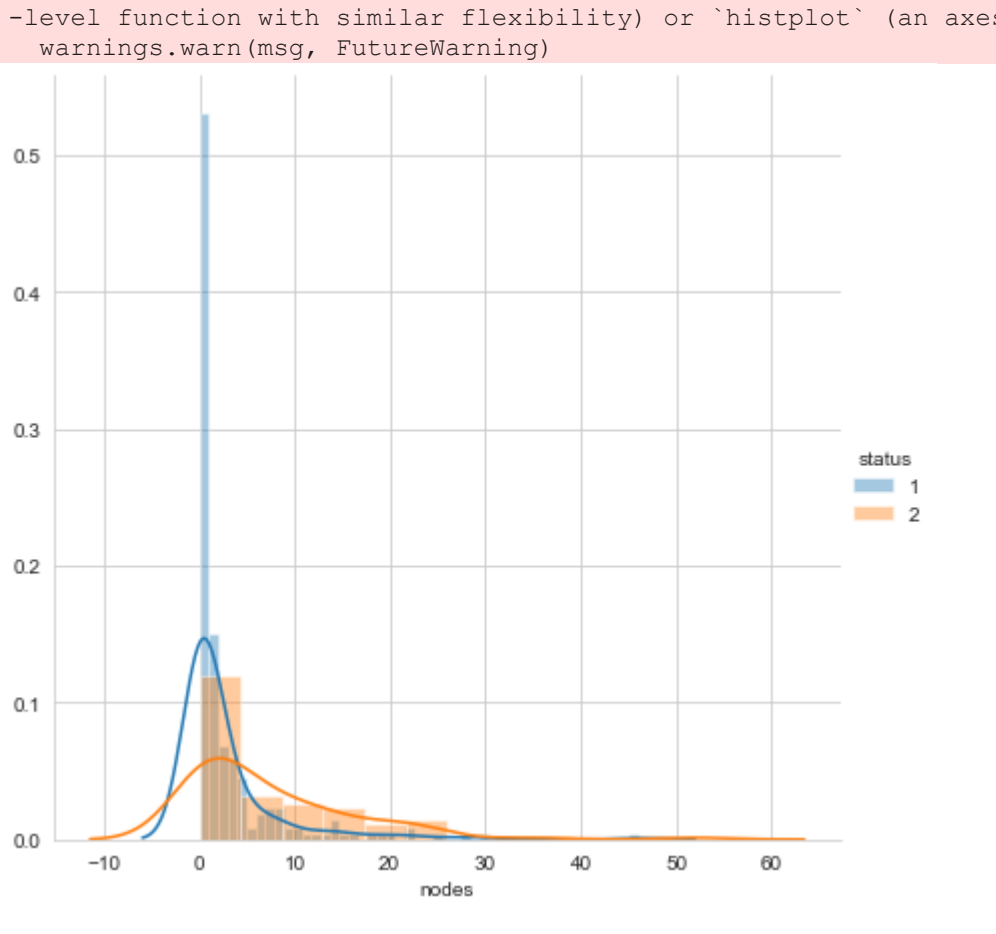
```
C:\Users\dev64\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprec
ated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\dev64\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprec
ated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



observation - very hard to understand , because mostly data overlap even pdf overlap

In [104…
```python
sns.FacetGrid(hbrmn, hue="status", height=6) \
   .map(sns.distplot, "nodes") \
   .add_legend();
plt.show();
```

```
C:\Users\dev64\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprec
ated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\dev64\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprec
ated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



observation - more no. of people who survive have very less no. of the nodes

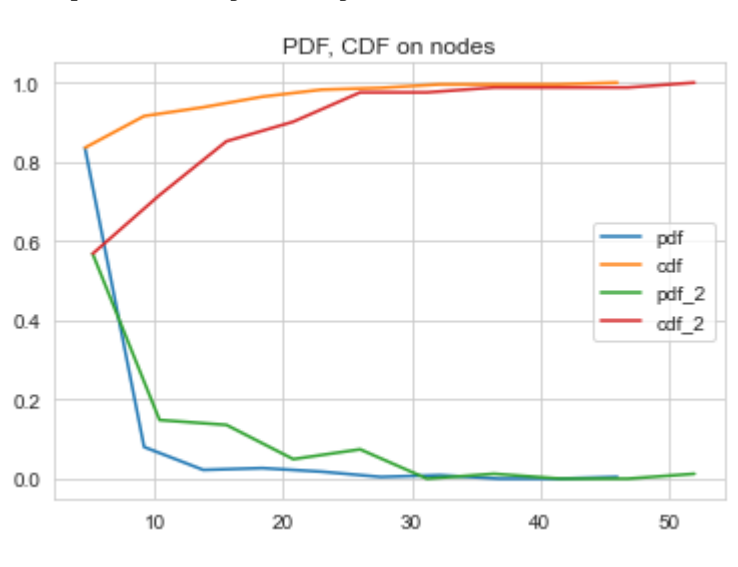
final observation -- nodes >>> age > year

PDF, CDF

In [105…
```python
hbrmn_1 = hbrmn.loc[hbrmn["status"] == 1]
hbrmn_2 = hbrmn.loc[hbrmn["status"] == 2]

counts, bin_edges = np.histogram(hbrmn_1['nodes'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,label = 'pdf')
plt.plot(bin_edges[1:], cdf,label = "cdf")
plt.legend()

counts, bin_edges = np.histogram(hbrmn_2['nodes'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)
cdf = np.cumsum(pdf)
plt.title("PDF, CDF on nodes")
plt.plot(bin_edges[1:],pdf,label = 'pdf_2')
plt.plot(bin_edges[1:], cdf,label = "cdf_2")
plt.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[105…
```
<matplotlib.legend.Legend at 0x1e6c4c1d580>
```



observations : as we see both pdf and cdf line in the graph, A.T.CDF - 81-82% survival chance have nodes less than equal to 4 or 5

observations : as we see both pdf and cdf line in the graph, A.T.CDF_0 - 58-59% non-survival chance have nodes less than equal to 4 or 5

In [106…
```python
#Mean, Variance, Std-deviation,

print("means:")
print(np.mean(hbrmn_1['nodes']))
print(np.mean(hbrmn_2['nodes']))


print("\nSTD-dev:");
print(np.std(hbrmn_1['nodes']))
print(np.std(hbrmn_2['nodes']))
```

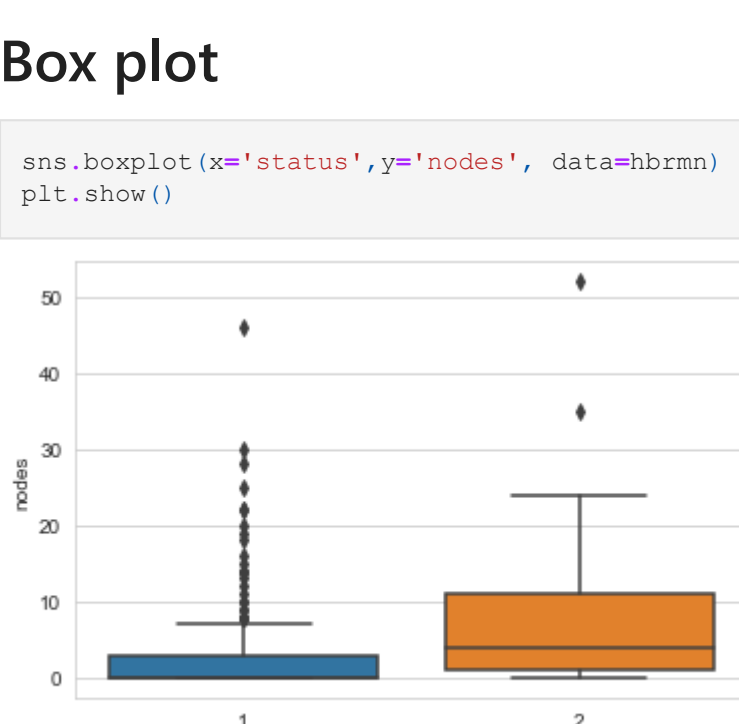
```
means:
2.7911111111111113
7.45679012345679

STD-dev:
5.857258449412131
9.128776076761632
```

In [107…
```python
#Median, Quantiles, Percentiles, IQR.

print("\nMedians:")
print(np.median(hbrmn_1['nodes']))
print(np.median(hbrmn_2['nodes']))

print("\nQuantiles:")
print(np.percentile(hbrmn_1['nodes'],np.arange(0, 100, 25)))
print(np.percentile(hbrmn_2['nodes'],np.arange(0, 100, 25)))


print("\n90th Percentiles:")
print(np.percentile(hbrmn_1['nodes'],90))
print(np.percentile(hbrmn_2['nodes'],90))

from statsmodels import robust
print ("\nMedian Absolute Deviation")
print(robust.mad(hbrmn_1['nodes']))
print(robust.mad(hbrmn_2['nodes']))
```

```
Medians:
0.0
4.0

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

90th Percentiles:
8.0
20.0

Median Absolute Deviation
0.0
5.930408874022408
```

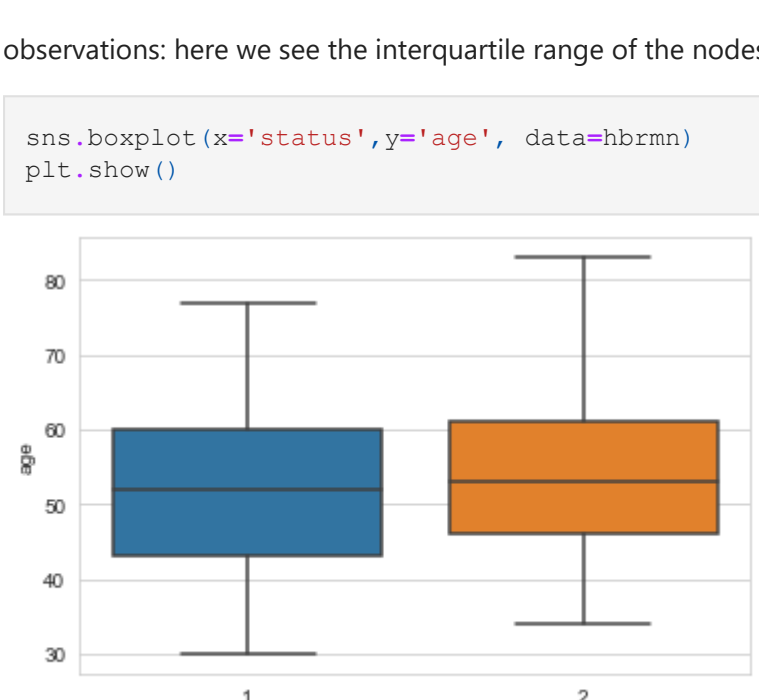
Box plot

In [108…
```python
sns.boxplot(x='status',y='nodes', data=hbrmn)
plt.show()
```



observations: here we see the interquartile range of the nodes, in servival - not more than 4

In [109…
```python
sns.boxplot(x='status',y='age', data=hbrmn)
plt.show()
```

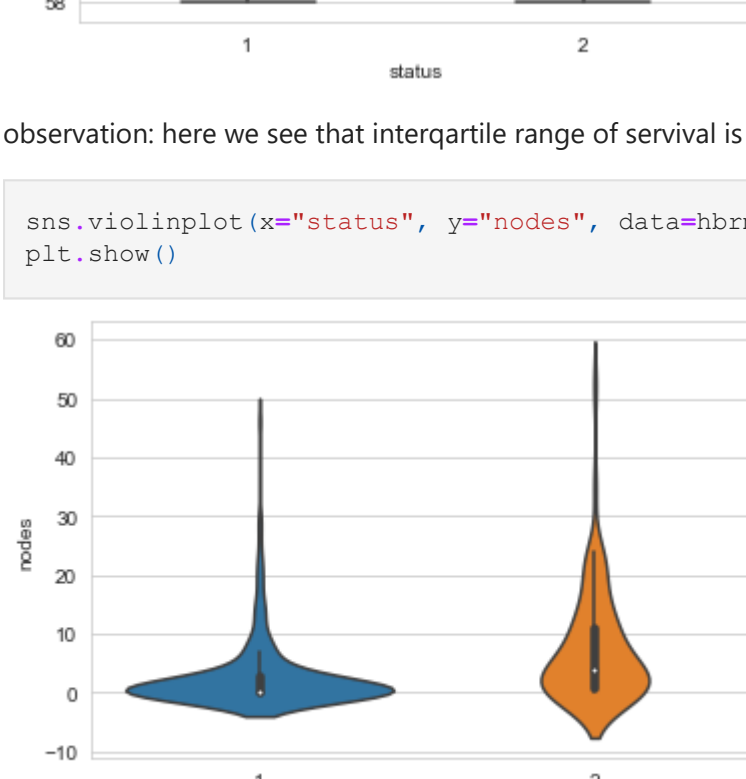


observation: interquartile range of servival in age factor have 44-60

In [110…
```python
sns.boxplot(x='status',y='year', data=hbrmn)
plt.show()
```

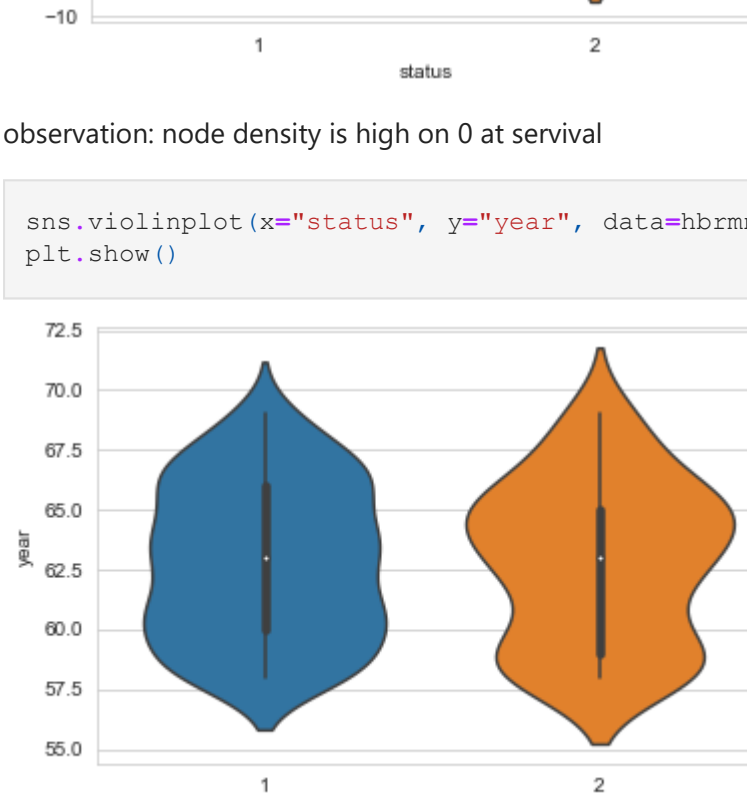
observation: here we see that interqartile range of servival is 60 - 66

In [111…
```python
sns.violinplot(x="status", y="nodes", data=hbrmn, size=8)
plt.show()
```



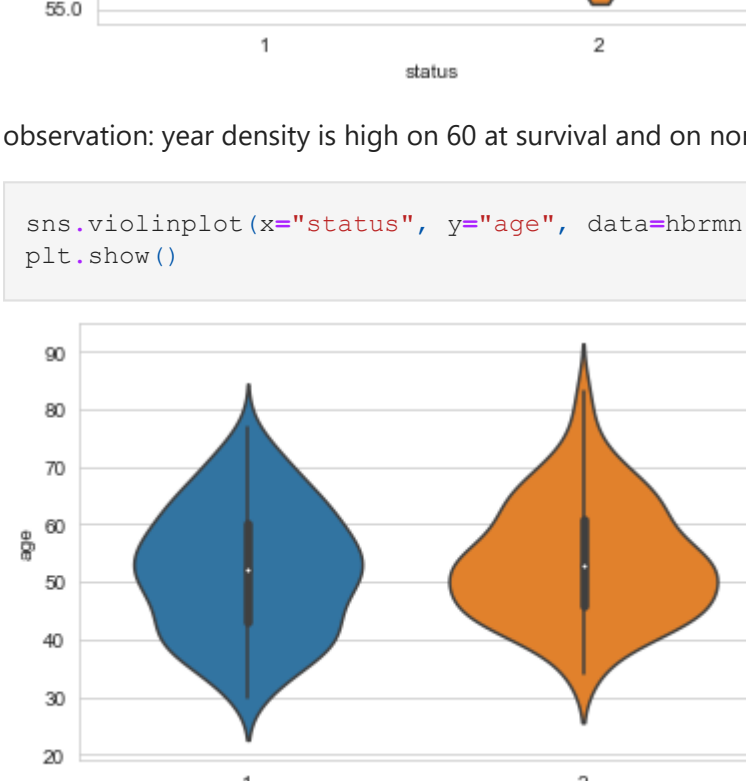
observation: node density is high on 0 at servival

In [112…
```python
sns.violinplot(x="status", y="year", data=hbrmn, size=8)
plt.show()
```



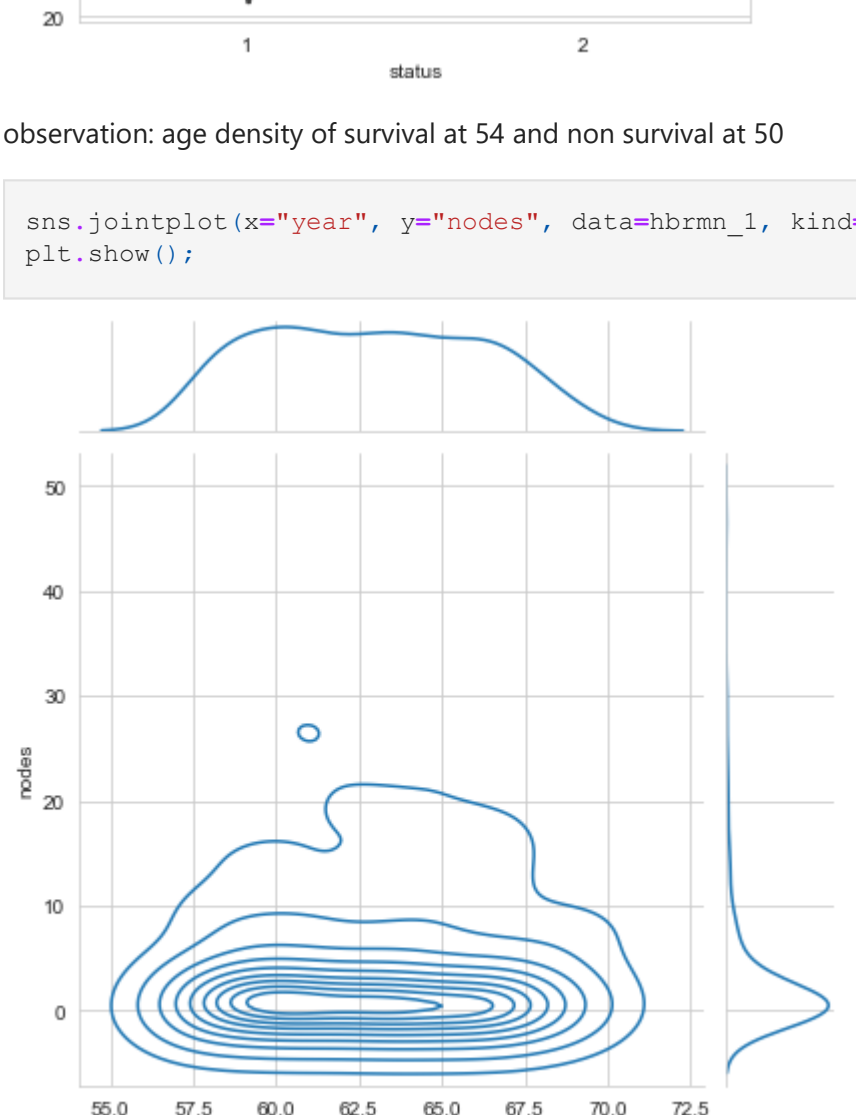
observation: year density is high on 60 at survival and on non survival at 64

In [113…
```python
sns.violinplot(x="status", y="age", data=hbrmn, size=8)
plt.show()
```



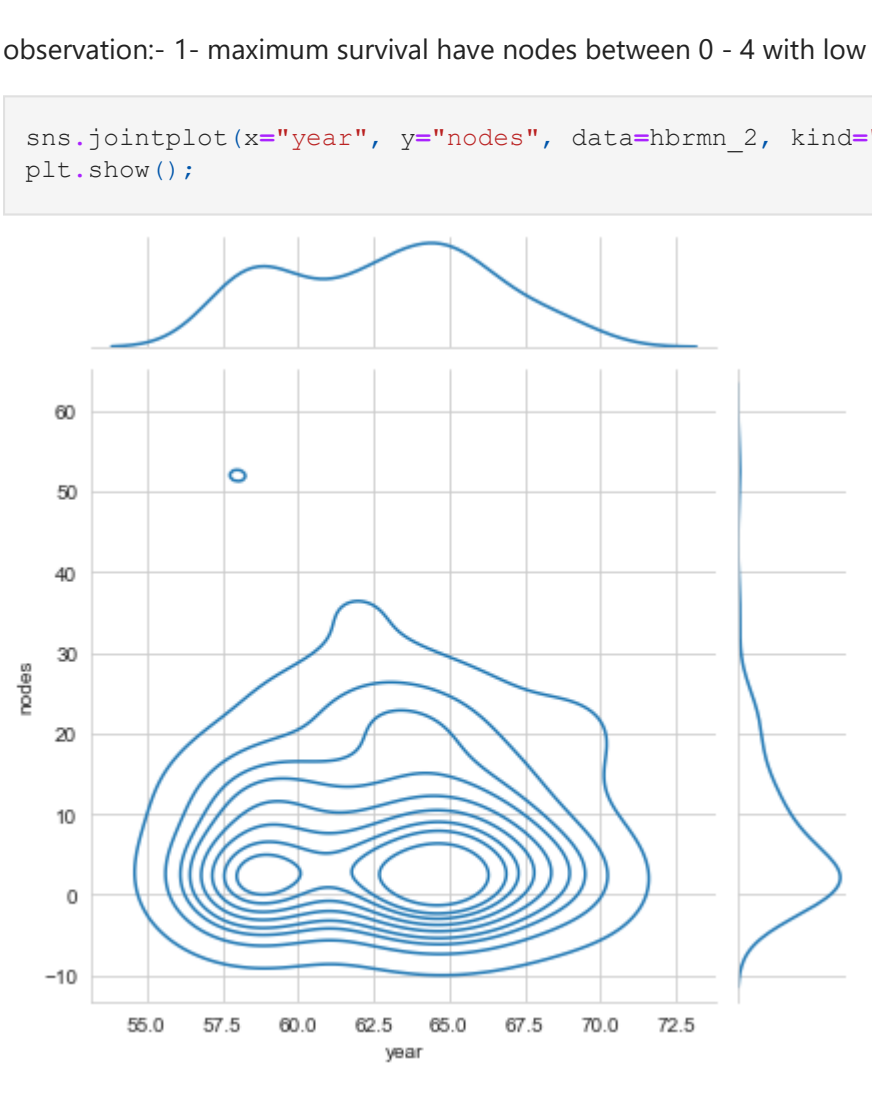
observation: age density of survival at 54 and non survival at 50

In [114…
```python
sns.jointplot(x="year", y="nodes", data=hbrmn_1, kind="kde");
plt.show();
```



observation:- 1- maximum survival have nodes between 0 - 4 with low varience

In [115…
```python
sns.jointplot(x="year", y="nodes", data=hbrmn_2, kind="kde");
plt.show();
```



observation: here in the non-survival the nodes have higher value what high varience