In [ ]:

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

In [ ]:

```
haber = pd.read_csv("/content/haberman.csv")
```

In [ ]:

```
haber.head()
```

Out[ ]:

	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 [ ]:

```
haber
```

Out[ ]:

	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
...	...	...	...	...
301	75	62	1	1
302	76	67	0	1
303	77	65	3	1
304	78	65	1	2
305	83	58	2	2

306 rows × 4 columns

There is 306 rows and 4 columns

In [ ]:

```
print(haber.shape)
```

```
(306, 4)
```

The column names

In [ ]:

```
haber.columns
```

Out[ ]:

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

number of classes = 2
1 = The patient survivied 5 yrs or longer

2 = The patient died with in 5 yrs

In [ ]:

```
haber['status'].value_counts()
```

Out[ ]:

```
1    225
2     81
Name: status, dtype: int64
```

In [ ]:

```
haber.describe()
```

Out[ ]:

	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

In [ ]:

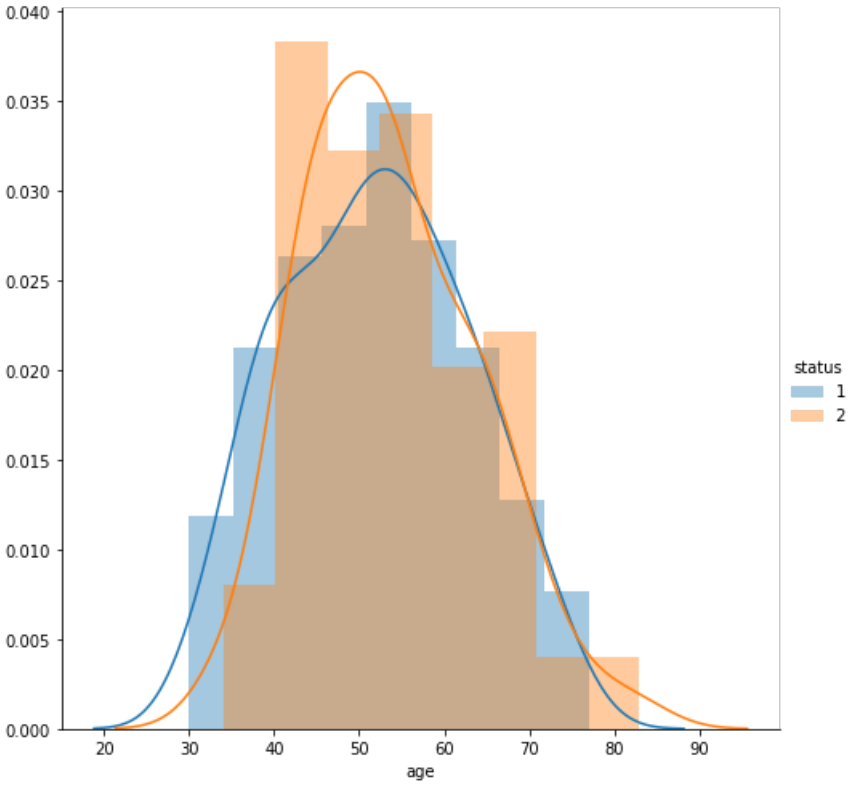
Objective: Analyze the data report the class attribute

Univariante Analysis

In [ ]:

```
#PDF
sb.FacetGrid(haber, hue = 'status', size = 7).map(sb.distplot, "age").add_legend();
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:316: UserWarning: The `size` parameter has be
en renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a
deprecated 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 h
istograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a
deprecated 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 h
istograms).
  warnings.warn(msg, FutureWarning)
```



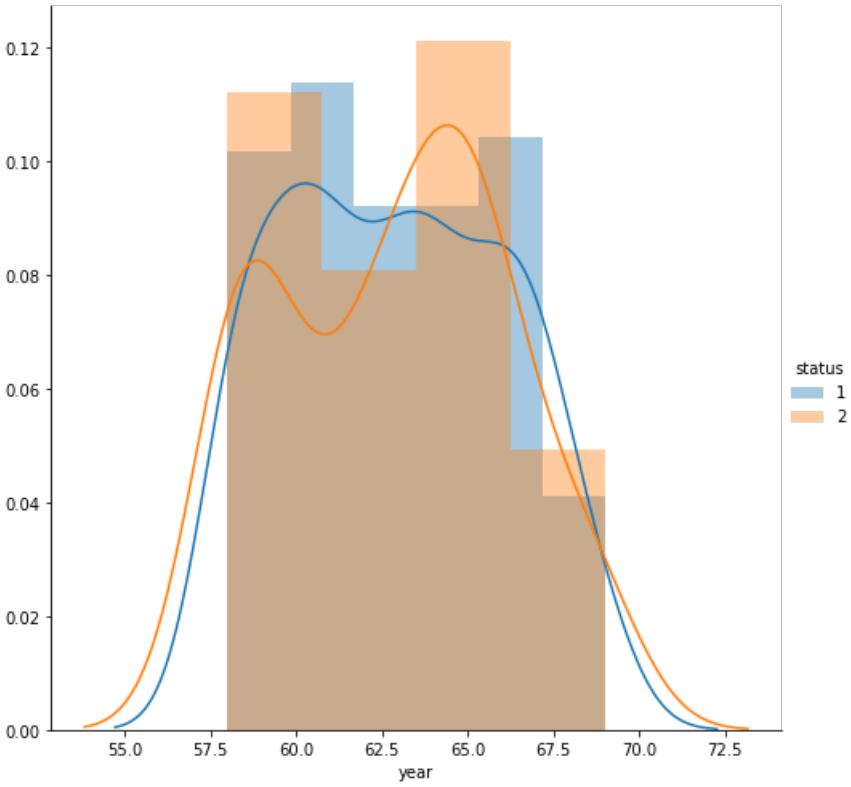
Observation:

1. Age > 30 have more probability for breast cancer
2. Age > 75 have less probability in survival

In [ ]:

```
sb.FacetGrid(haber, hue = 'status', size = 7).map(sb.distplot, "year").add_legend();
```

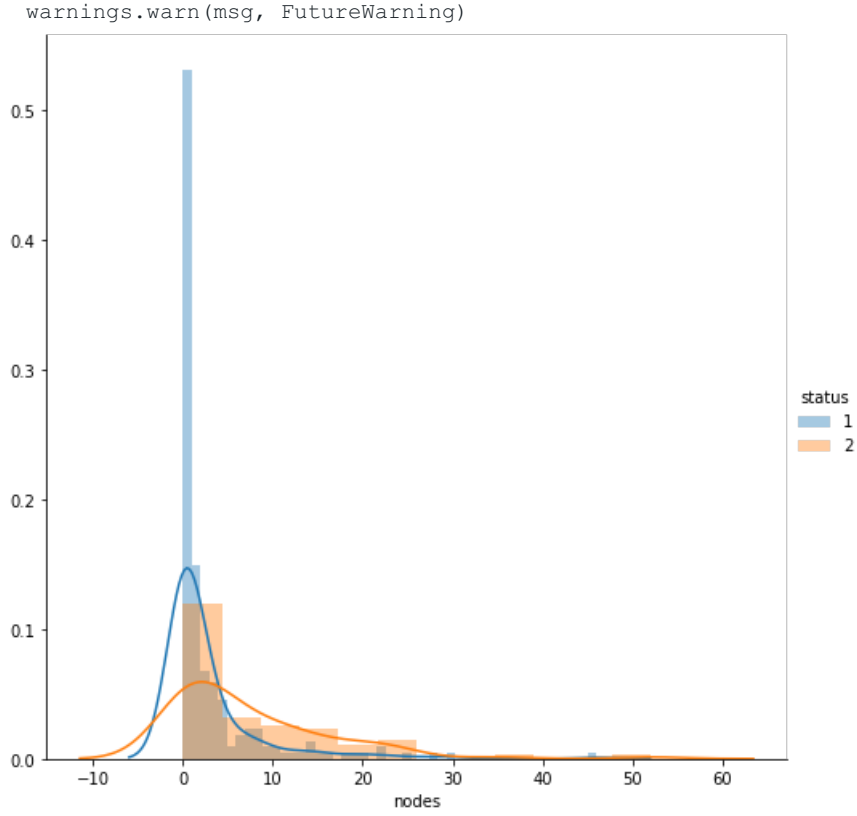
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:316: UserWarning: The `size` parameter has be
en renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a
deprecated 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 h
istograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a
deprecated 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 h
istograms).
  warnings.warn(msg, FutureWarning)
```



In [ ]:

```
sb.FacetGrid(haber, hue = 'status', size = 7).map(sb.distplot, "nodes").add_legend();
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:316: UserWarning: The `size` parameter has be
en renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a
deprecated 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 h
istograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a
deprecated 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 h
istograms).
  warnings.warn(msg, FutureWarning)
```



Observed:

Axillary nodes maximum lie between 0 and 1.

Above 30 nodes death rate is maximum

In [ ]:

```
stage1 = haber.loc[haber['status'] == 1]
stage2 = haber.loc[haber['status'] == 2]
```

In [ ]:

```
stage1.shape
```

Out[ ]:

```
(225, 4)
```

In [ ]:

```
stage1.describe()
```

Out[ ]:

	age	year	nodes	status
count	225.000000	225.000000	225.000000	225.0
mean	52.017778	62.862222	2.791111	1.0
std	11.012154	3.222915	5.870318	0.0
min	30.000000	58.000000	0.000000	1.0
25%	43.000000	60.000000	0.000000	1.0
50%	52.000000	63.000000	0.000000	1.0
75%	60.000000	66.000000	3.000000	1.0
max	77.000000	69.000000	46.000000	1.0

In [ ]:

In [ ]:

```
stage2.shape
```

Out[ ]:

```
(81, 4)
```

In [ ]:

```
stage2.describe()
```

Out[ ]:

	age	year	nodes	status
count	81.000000	81.000000	81.000000	81.0
mean	53.679012	62.827160	7.456790	2.0
std	10.167137	3.342118	9.185654	0.0
min	34.000000	58.000000	0.000000	2.0
25%	46.000000	59.000000	1.000000	2.0
50%	53.000000	63.000000	4.000000	2.0
75%	61.000000	65.000000	11.000000	2.0
max	83.000000	69.000000	52.000000	2.0

In [ ]:

```
stage1.head()
```

Out[ ]:

	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

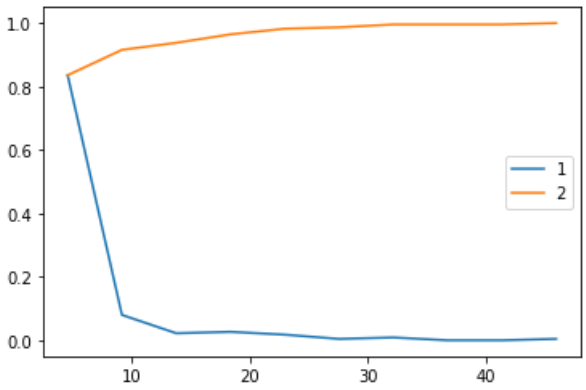
In [ ]:

```
#CDF for status 1 based on nodes
counts, bin_edges = np.histogram(stage1['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.legend(['1', '2'])
```

```
[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. ]
```

Out[ ]:

```
<matplotlib.legend.Legend at 0x7fa52a268d50>
```



Observed:

if auxillary node is 10, 80% possibility for long survival

In [ ]:

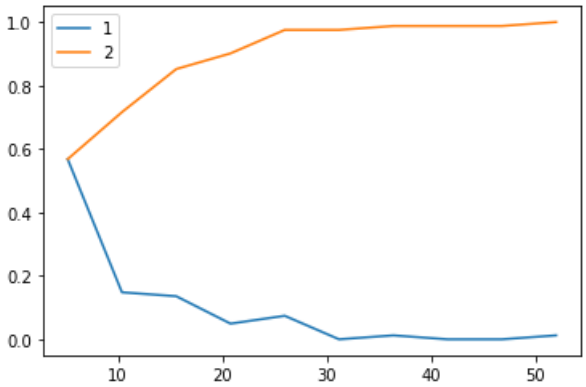
```
#CDF for status 1 based on nodes
counts, bin_edges = np.histogram(stage2['nodes'], bins=10,
```

```
                                    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.legend(['1', '2'])
```

```
[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[ ]:

```
<matplotlib.legend.Legend at 0x7fa52a1dbbd0>
```



Observed

if auxillary node is increases possibility for long survival is minimum.

In [ ]:

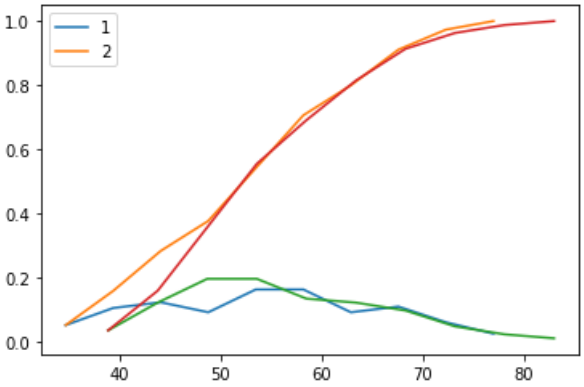
```
#CDF for status 1 based on year
counts, bin_edges = np.histogram(stage1['age'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:], cdf)
plt.legend(['1', '2'])

#CDF for status 2 based on year
counts, bin_edges = np.histogram(stage2['age'], 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.legend(['1', '2'])
```

```
[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. ]
[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. ]
```

Out[ ]:

```
<matplotlib.legend.Legend at 0x7fa529fd6050>
```

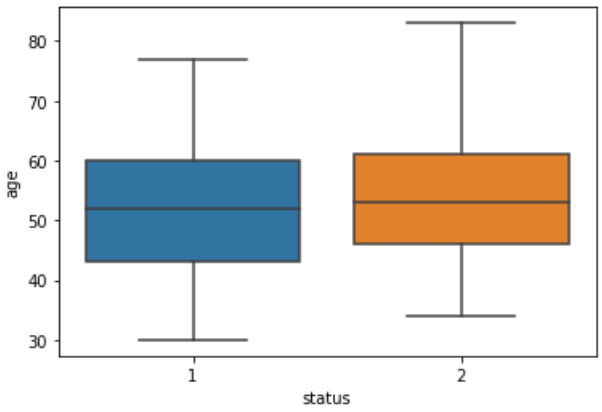


Observed from CDF Plots:

Auxillary Node data is much use full than other data to finding the probability of survival

In [ ]:

```
#box plot
sb.boxplot(x='status',y='age', data=haber)
plt.show()
```



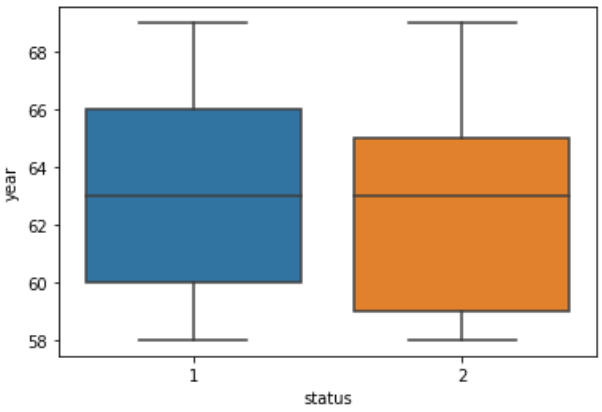
Observation:

Minimum age for getting cancer and operation is 30 also they are most probabily long survived

Maximum age 80 should not survive longer

In [ ]:

```
#box plot
sb.boxplot(x='status',y='year', data=haber)
plt.show()
```

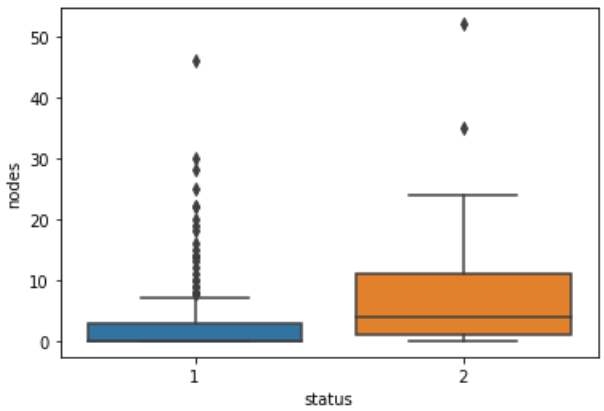


Age 63 is 50th percentile for both long_survival and short_survival

In [ ]:

```
#box plot
```

```
sb.boxplot(x='status',y='nodes', data=haber)
plt.show()
```

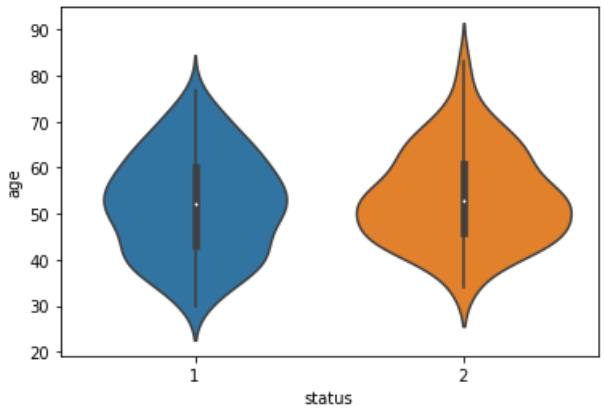


Observation for Boxplot:

Box plot for Age column gives some information related to objective

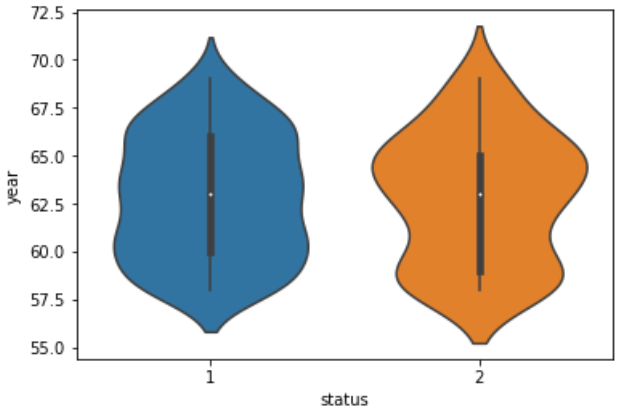
In [ ]:

```
#violin plot of age and status
sb.violinplot(x="status", y="age", data=haber, size=8)
plt.show()
```



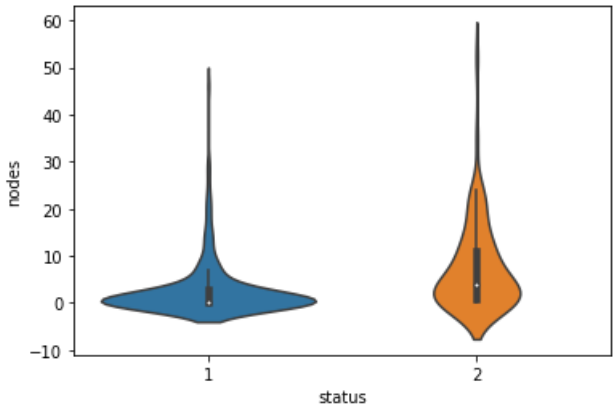
In [ ]:

```
#violin plot of age and status
sb.violinplot(x="status", y="year", data=haber, size=8)
plt.show()
```



In [ ]:

```
#violin plot of age and status
sb.violinplot(x="status", y="nodes", data=haber, size=8)
plt.show()
```

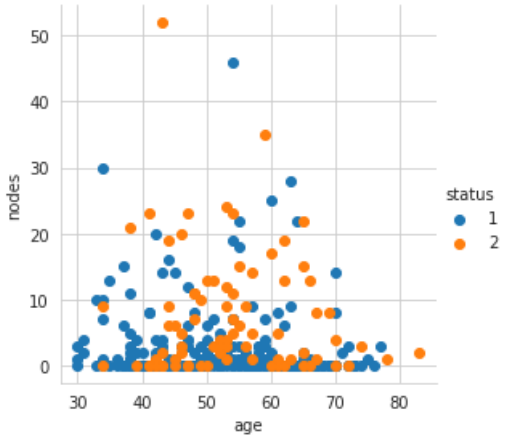


Bivariant Analysis

In [ ]:

```
#2D Scatter Plots
sb.set_style("whitegrid");
sb.FacetGrid(haber, hue="status", size=4).map(plt.scatter, "age", "nodes").add_legend();
plt.show();
```

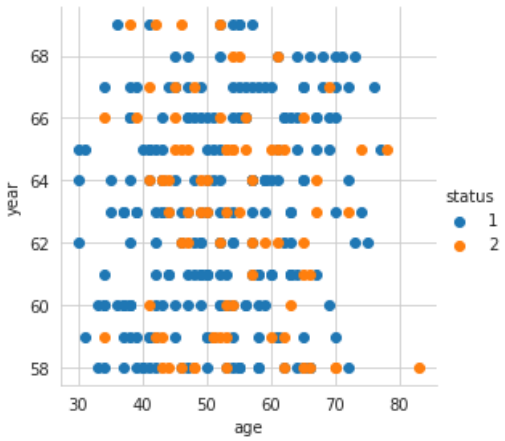
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:316: UserWarning: The `size` parameter has be
en renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```



In [ ]:

```
#2D Scatter Plots
sb.set_style("whitegrid");
sb.FacetGrid(haber, hue="status", size=4).map(plt.scatter, "age", "year").add_legend();
plt.show();
```

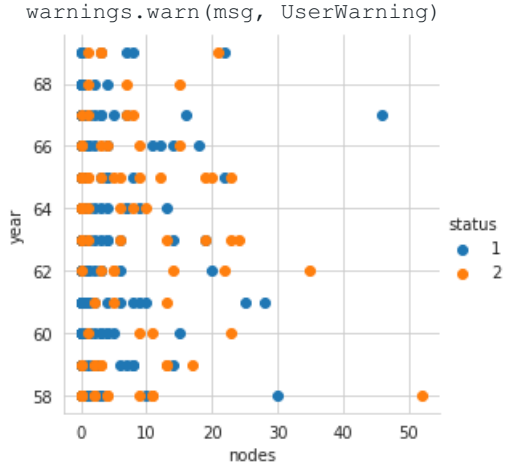
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:316: UserWarning: The `size` parameter has be
en renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```



In [ ]:

```
#2D Scatter Plots
sb.set_style("whitegrid");
sb.FacetGrid(haber, hue="status", size=4).map(plt.scatter, "nodes", "year").add_legend();
plt.show();
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:316: UserWarning: The `size` parameter has be
en renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```



Observations for 2D scatter plot:

for a scatter plot age and node column is little bit scatter. if auxillary nodes > 50 long survival is not possible so count of the patient for operation is also minimum.

Other two combination of columns are overlapped.

In [ ]:

```
#pair plots
sb.set_style("whitegrid")
sb.pairplot(haber, hue="status", palette = 'rainbow', size=3)
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:1969: UserWarning: The `size` parameter has b
een renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```

Observation of Pair plots:

if auxillary node is minimum long survival is possible

More overlappings are there

Overall Observation on haber data

Auxillary node and Age are the important feature to decide survival

There is no possibility for long survival Auxillary node > 50. and operated patients also minimum

Minimum age for getting cancer and operation is 30 also they are most probabily long survived

Maximum age 80 should not survive longer