ASSIGNMENT 2

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

In [2]:
```
haber=pd.read_csv("haberman.csv")
haber
```

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
5	33	58	10	1
6	33	60	0	1
7	34	59	0	2
8	34	66	9	2
9	34	58	30	1
10	34	60	1	1
11	34	61	10	1
12	34	67	7	1
13	34	60	0	1
14	35	64	13	1

	age	year	nodes	status
15	35	63	0	1
16	36	60	1	1
17	36	69	0	1
18	37	60	0	1
19	37	63	0	1
20	37	58	0	1
21	37	59	6	1
22	37	60	15	1
23	37	63	0	1
24	38	69	21	2
25	38	59	2	1
26	38	60	0	1
27	38	60	0	1
28	38	62	3	1
29	38	64	1	1
...	...	...	...	...
276	67	66	0	1
277	67	61	0	1
278	67	65	0	1
279	68	67	0	1
280	68	68	0	1
281	69	67	8	2
282	69	60	0	1
283	69	65	0	1
284	69	66	0	1

	age	year	nodes	status
285	70	58	0	2
286	70	58	4	2
287	70	66	14	1
288	70	67	0	1
289	70	68	0	1
290	70	59	8	1
291	70	63	0	1
292	71	68	2	1
293	72	63	0	2
294	72	58	0	1
295	72	64	0	1
296	72	67	3	1
297	73	62	0	1
298	73	68	0	1
299	74	65	3	2
300	74	63	0	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

In [3]:
```
print(haber.shape)
```

```
(306, 4)
```

In [4]:
```
print(haber.columns)
```

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

In [5]:
```
haber["status"].value_counts()
```

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

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

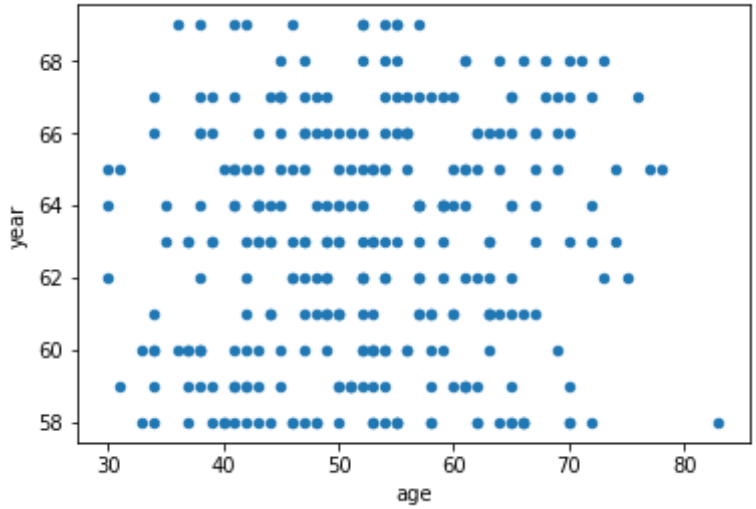
Out[6]:

	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

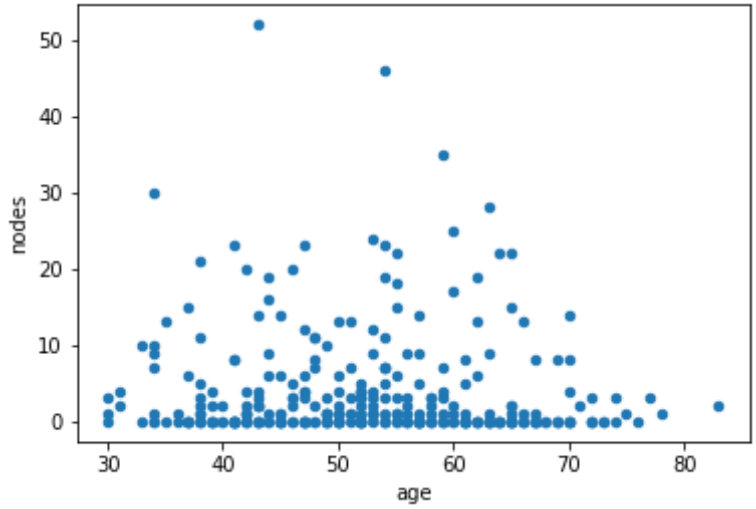
max number of positive nodes are 52 25% people have no positive nodes 75% people have less than 5 positive nodes

the age of patients between 30 to 80 have mean 52.

In [7]:
```
haber.plot(kind='scatter',x='age',y='year')
plt.show()
```



In [8]:
```
haber.plot(kind="scatter",x='age',y='nodes')
plt.show()
```

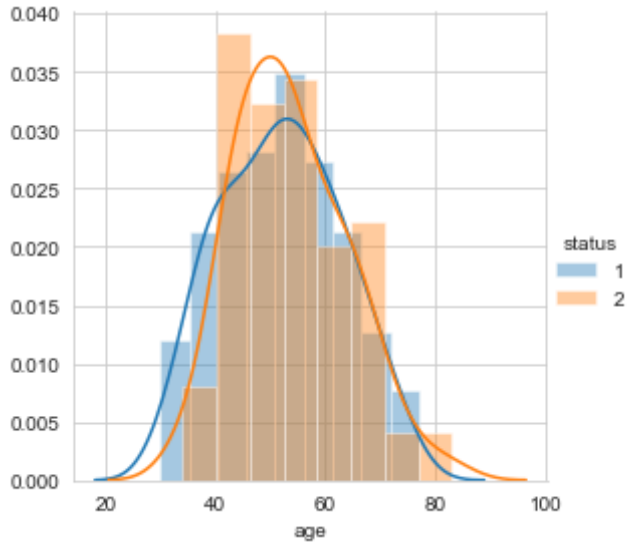


In [9]:
```
def density_plot(feature_var, class_var):
```

```
    sns.set_style(style="whitegrid")
    sns.FacetGrid(data=haber, hue=class_var, size=4) \
    .map(sns.distplot, feature_var) \
    .add_legend()
```

In [10]:
```
density_plot('age','status')
plt.show()
```

```
C:\Users\Himani Mogra\Anaconda3\lib\site-packages\seaborn\axisgrid.py:230: UserWarning: The `size` paramter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```



In [11]:
```
plt.close()
sns.set_style("whitegrid")
sns.pairplot(haber,hue='status',vars=['age','year','nodes'] ,height=3)
plt.show()
```

by the above pairplot scattering we observe that the best results are obtained by the graph between year treatment and

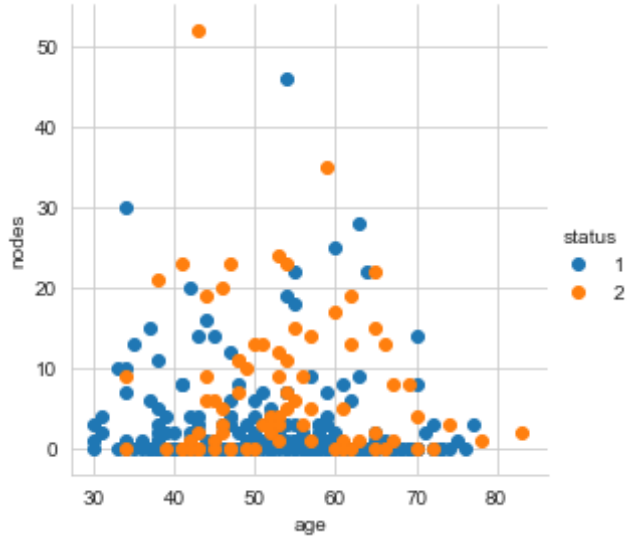
In [12]:
```
sns.set_style("whitegrid")
```

```
sns.FacetGrid(haber,hue="status",size=4)\
.map(plt.scatter,"age","year")\
.add_legend()
plt.show()
```

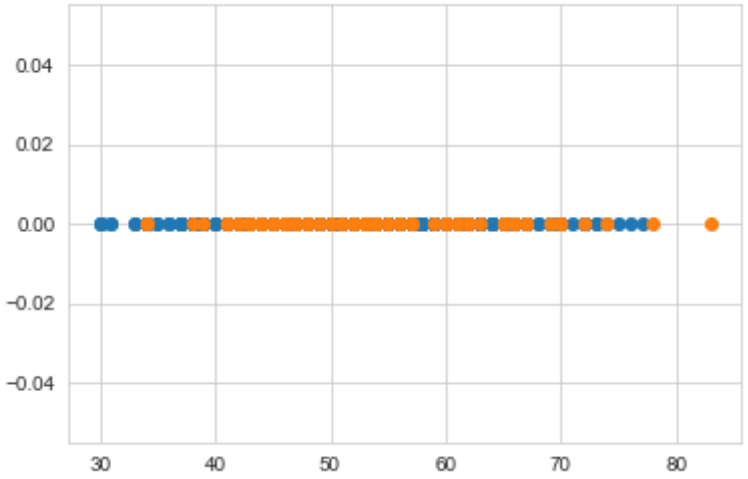
```
C:\Users\Himani Mogra\Anaconda3\lib\site-packages\seaborn\axisgrid.py:2
30: UserWarning: The `size` paramter has been renamed to `height`; plea
se update your code.
  warnings.warn(msg, UserWarning)
```

In [13]:
```
sns.set_style("whitegrid")
sns.FacetGrid(haber,hue="status",size=4)\
.map(plt.scatter,'age','nodes')\
.add_legend()
plt.show()
```

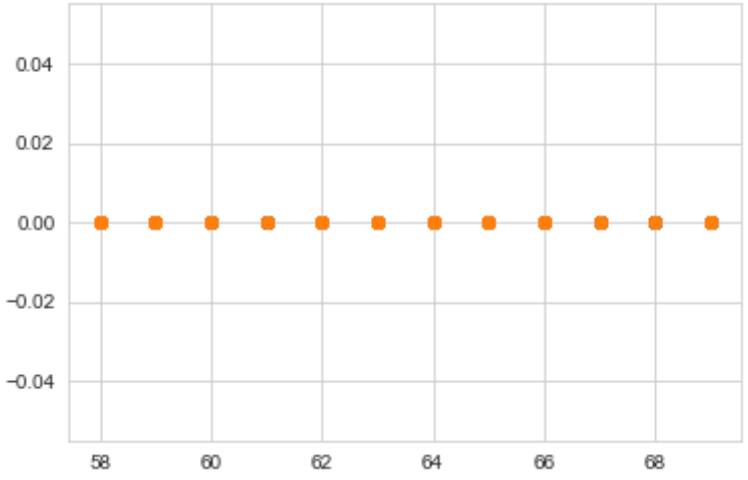
```
C:\Users\Himani Mogra\Anaconda3\lib\site-packages\seaborn\axisgrid.py:2
30: UserWarning: The `size` paramter has been renamed to `height`; plea
se update your code.
  warnings.warn(msg, UserWarning)
```



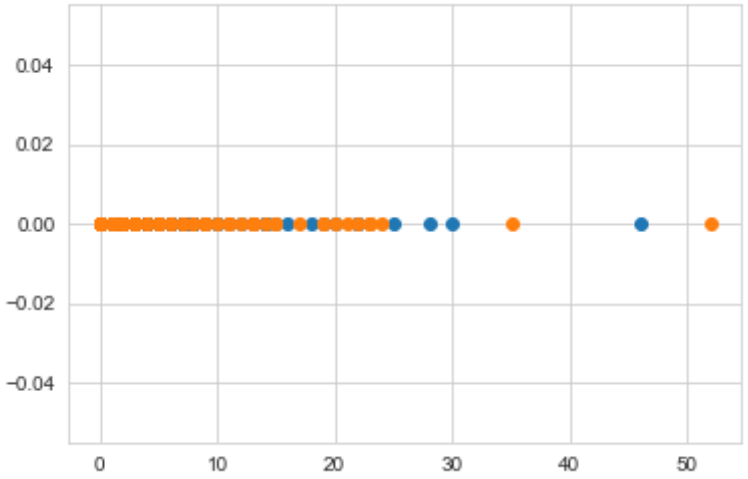
In [14]:
```
import numpy as np
haber_1=haber.loc[haber["status"]==1]
haber_2=haber.loc[haber["status"]==2]

plt.plot(haber_1["age"],np.zeros_like(haber_1["age"]),'o')
plt.plot(haber_2["age"],np.zeros_like(haber_2["age"]),'o')
plt.show()
```



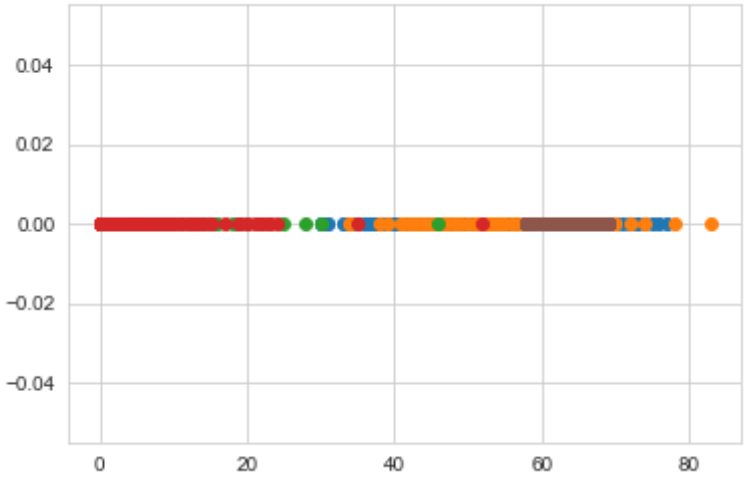
In [15]:
```
haber_1=haber.loc[haber["status"]==1]
haber_2=haber.loc[haber["status"]==2]

plt.plot(haber_1["year"],np.zeros_like(haber_1["year"]),'o')
plt.plot(haber_2["year"],np.zeros_like(haber_2["year"]),'o')
plt.show()
```



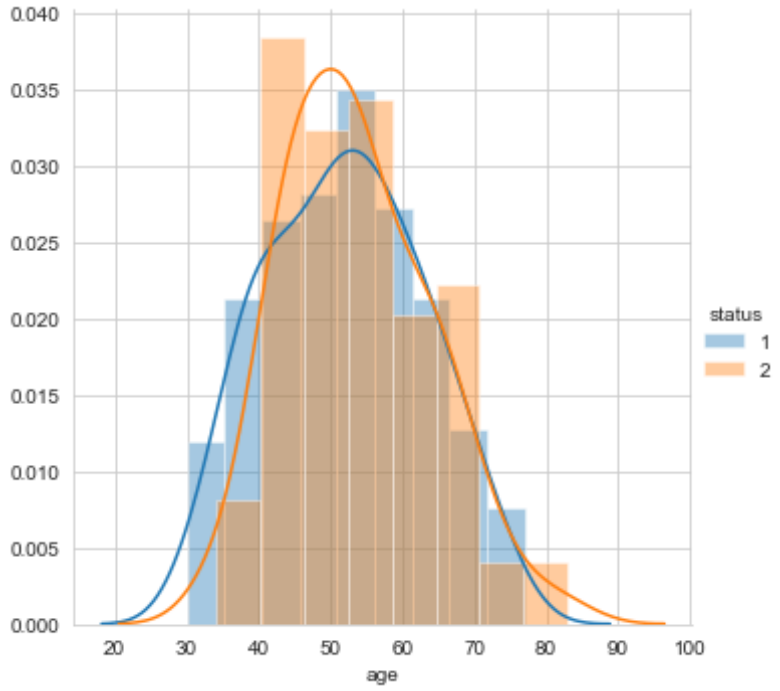
In [16]:
```
haber_1=haber.loc[haber["status"]==1]
haber_2=haber.loc[haber["status"]==2]

plt.plot(haber_1["nodes"],np.zeros_like(haber_1["nodes"]),'o')
plt.plot(haber_2["nodes"],np.zeros_like(haber_2["nodes"]),'o')
plt.show()
```



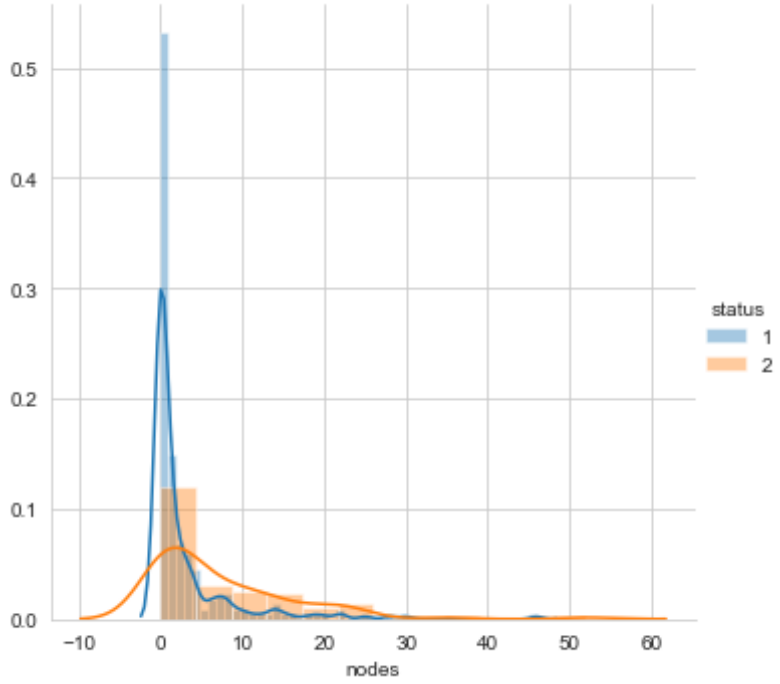
In [17]:
```
haber_1=haber.loc[haber["status"]==1]
haber_2=haber.loc[haber["status"]==2]

plt.plot(haber_1["age"],np.zeros_like(haber_1["age"]),'o')
plt.plot(haber_2["age"],np.zeros_like(haber_2["age"]),'o')

plt.plot(haber_1["nodes"],np.zeros_like(haber_1["nodes"]),'o')
plt.plot(haber_2["nodes"],np.zeros_like(haber_2["nodes"]),'o')

plt.plot(haber_1["year"],np.zeros_like(haber_1["year"]),'o')
plt.plot(haber_2["year"],np.zeros_like(haber_2["year"]),'o')
plt.show()
```



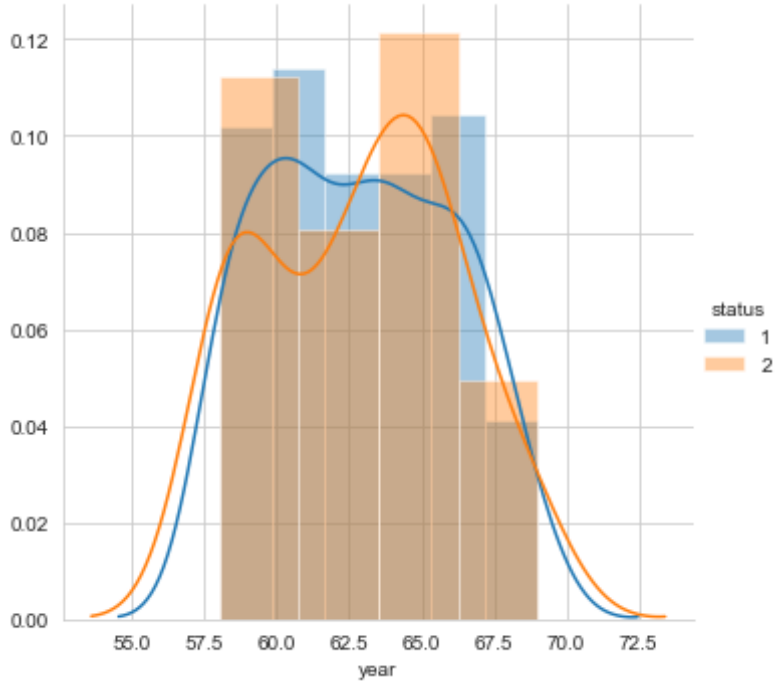
In [18]:
```
sns.FacetGrid(haber,hue="status",height=5)\
.map(sns.distplot,"age")\
.add_legend()
plt.show()
```



In [19]:
```
sns.FacetGrid(haber,hue="status",height=5)\
.map(sns.distplot,"nodes")\
.add_legend()
plt.show()
```

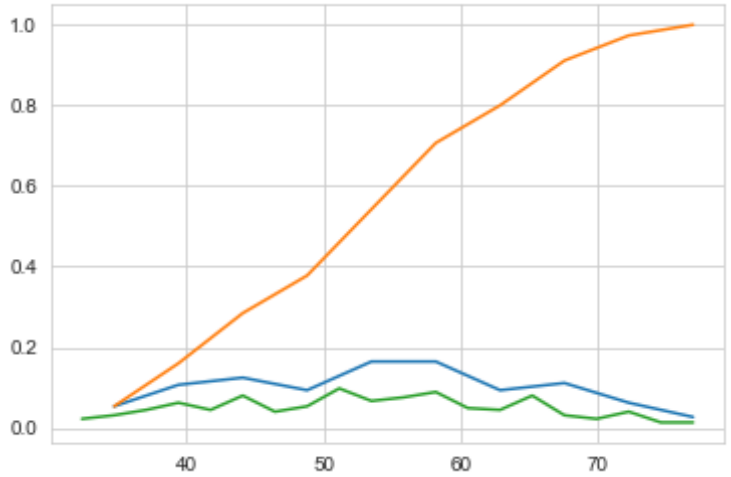


In [20]:
```
sns.FacetGrid(haber,hue="status",height=5)\
.map(sns.distplot,"year")\
.add_legend()
plt.show()
```



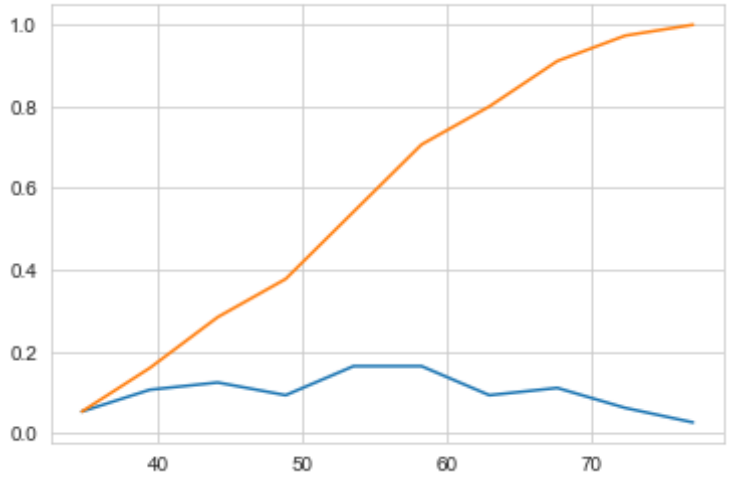
In [21]:
```
import numpy as np
counts,bin_edges=np.histogram(haber_1["age"],bins=10,density=True)

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)

counts,bin_edges=(np.histogram(haber_1["age"],bins=20,density=True))
pdf=counts/(sum(counts))
plt.plot(bin_edges[1:],pdf)
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. ]
```



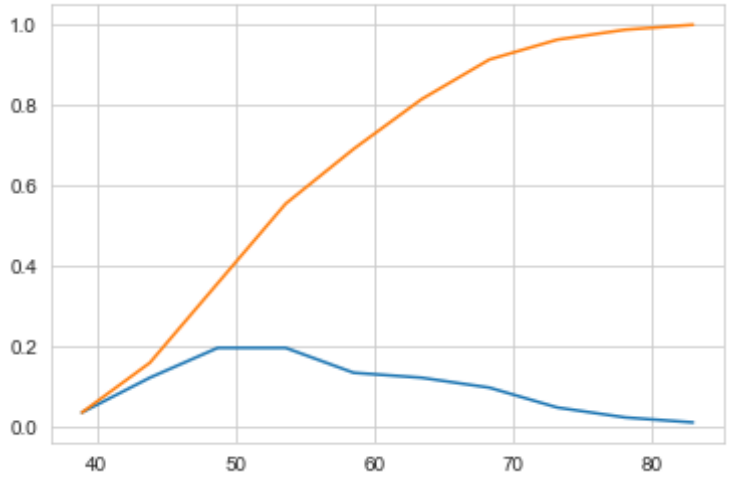
In [22]:
```
counts,bin_edges=np.histogram(haber_1["age"],bins=10,density=True)
pdf=counts/(sum(counts))
print(pdf)
print(cdf)
cdf=np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)
plt.show()
```

```
[0.05333333 0.10666667 0.12444444 0.09333333 0.16444444 0.16444444
 0.09333333 0.11111111 0.06222222 0.02666667]
[0.05333333 0.16       0.28444444 0.37777778 0.54222222 0.70666667
 0.8        0.91111111 0.97333333 1.        ]
```



In [23]:
```
counts,bin_edges=np.histogram(haber_2["age"],bins=10,density=True)
pdf=counts/(sum(counts))
print(pdf)
print(cdf)
cdf=np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)
plt.show()
```

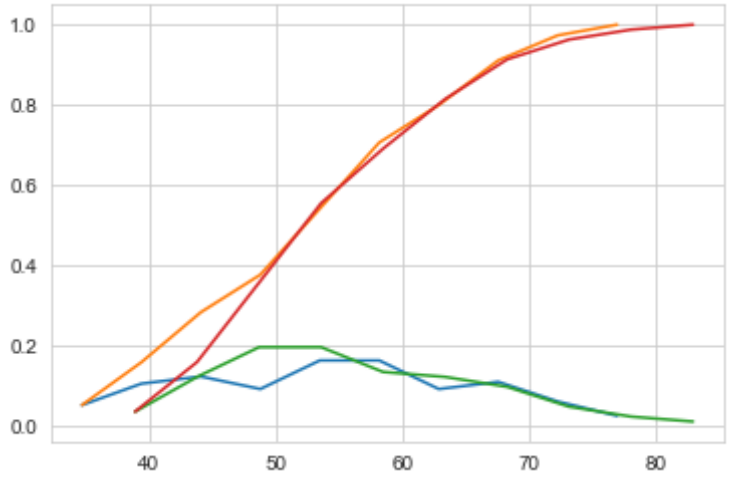
```
[0.03703704 0.12345679 0.19753086 0.19753086 0.13580247 0.12345679
 0.09876543 0.04938272 0.02469136 0.01234568]
[0.05333333 0.16       0.28444444 0.37777778 0.54222222 0.70666667
 0.8        0.91111111 0.97333333 1.        ]
```



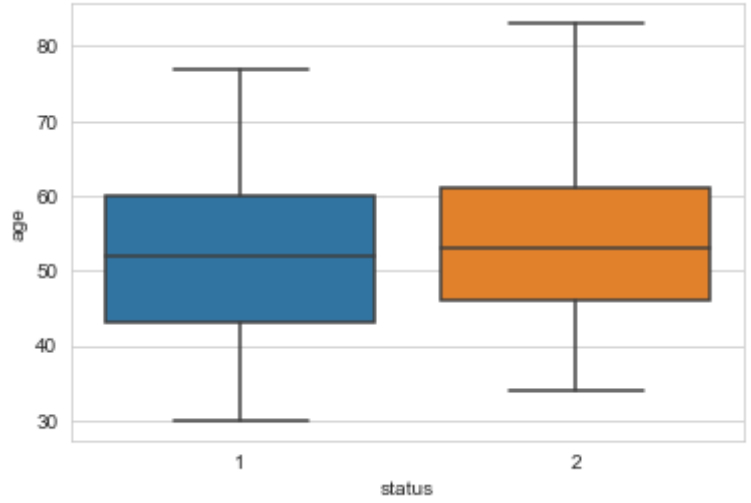
In [24]:
```
counts,bin_edges=np.histogram(haber_1["age"],bins=10,density=True)
pdf=counts/(sum(counts))
print(pdf)
print(cdf)
cdf=np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)


counts,bin_edges=np.histogram(haber_2["age"],bins=10,density=True)
pdf=counts/(sum(counts))
print(pdf)
print(cdf)
cdf=np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:],cdf)
plt.show()
```

```
[0.05333333 0.10666667 0.12444444 0.09333333 0.16444444 0.16444444
 0.09333333 0.11111111 0.06222222 0.02666667]
[0.03703704 0.16049383 0.35802469 0.55555556 0.69135802 0.81481481
 0.91358025 0.96296296 0.98765432 1.         ]
[0.03703704 0.12345679 0.19753086 0.19753086 0.13580247 0.12345679
```

```
 0.09876543 0.04938272 0.02469136 0.01234568]
[0.05333333 0.16       0.28444444 0.37777778 0.54222222 0.70666667
 0.8        0.91111111 0.97333333 1.        ]
```



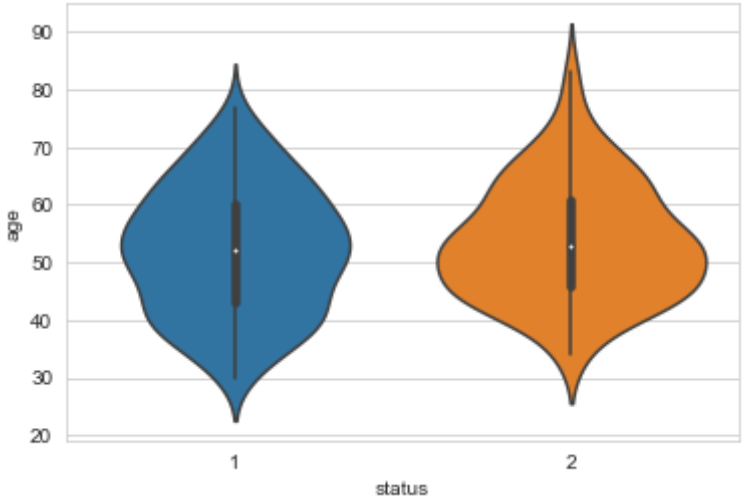
In [25]:
```
sns.boxplot(x="status",y="age",data=haber)
plt.show()
```



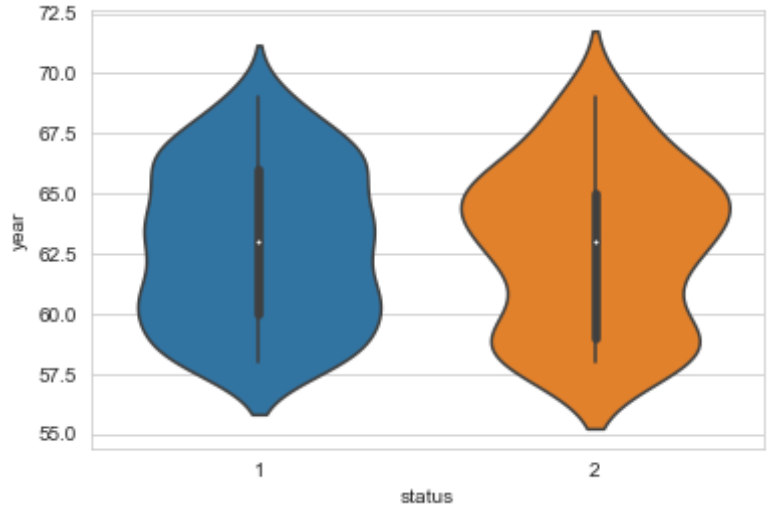
25th percentile values having status 1 lies between age of 30 to 45 50th percentile values having status 1 lies between age of 45 to 60 75th percentile values having status 1 lies between age of 60 to 75

25th percentile values having status 2 lies between age of patients 35 to 47 50th percentile values of patients with status lies between age of 47 to 62 75th percentile values of patients having status 2 lies between age of 62 to 85.

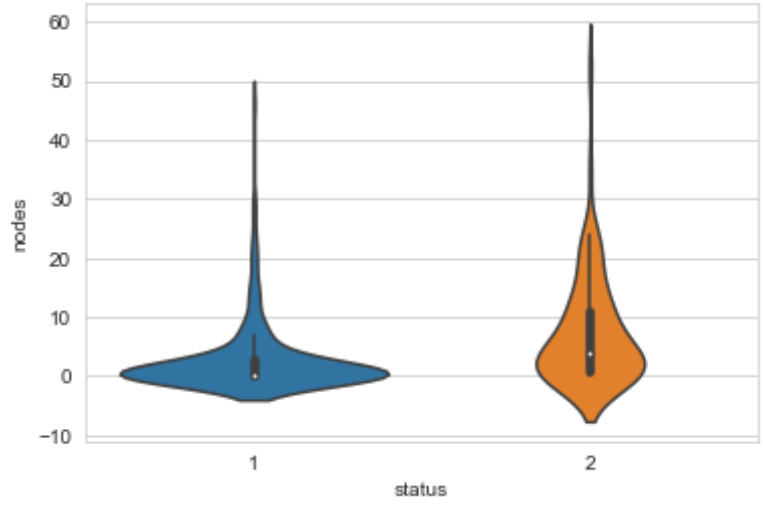
In [26]:
```
sns.violinplot(x="status",y="age",data=haber,height=8)
plt.show()
```



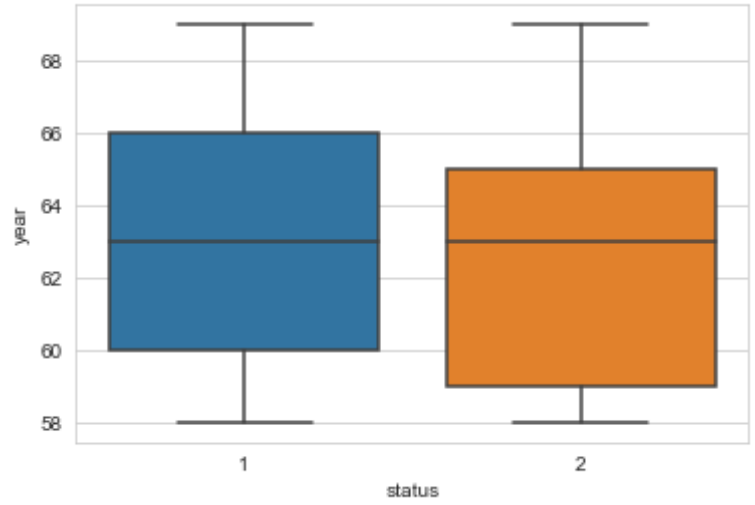
In [27]:
```
sns.violinplot(x="status",y="year",data=haber)
plt.show()
```



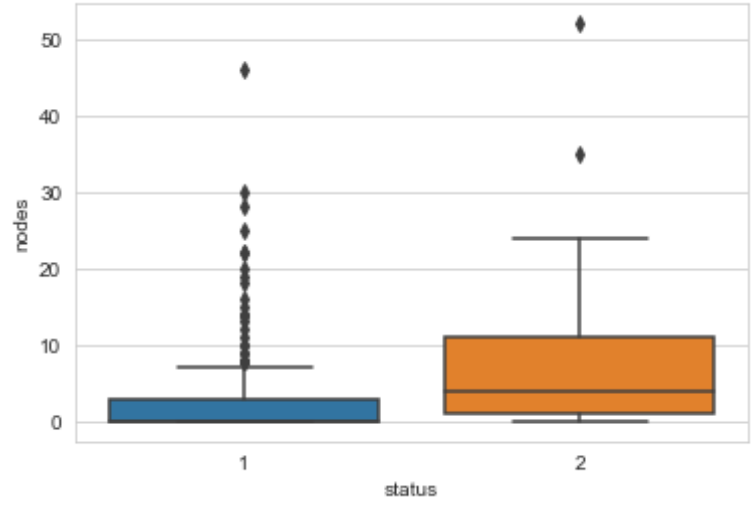
In [28]:
```
sns.violinplot(x="status",y="nodes",data=haber)
plt.show()
```



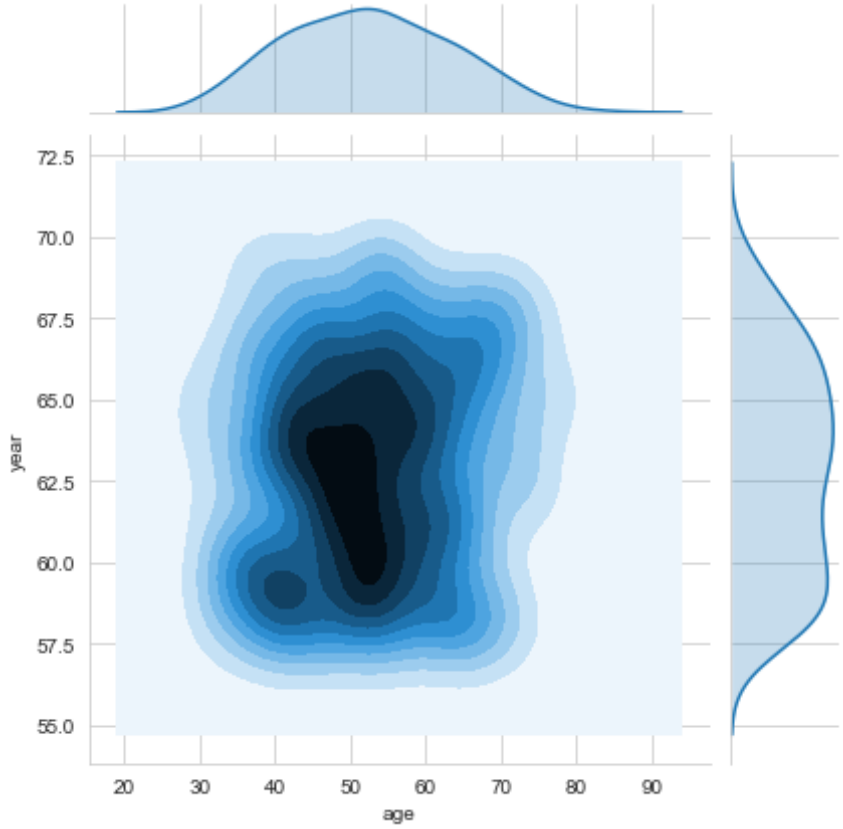
In [29]:
```
sns.boxplot(x="status",y="year",data=haber)
plt.show()
```



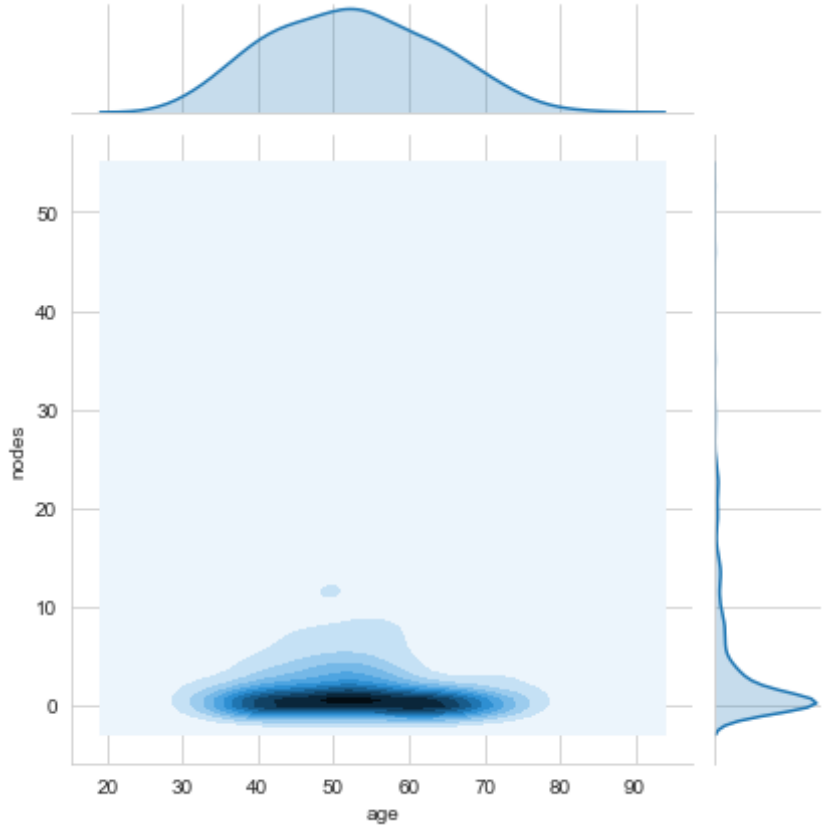
In [30]:
```
sns.boxplot(x="status",y="nodes",data=haber)
plt.show()
```



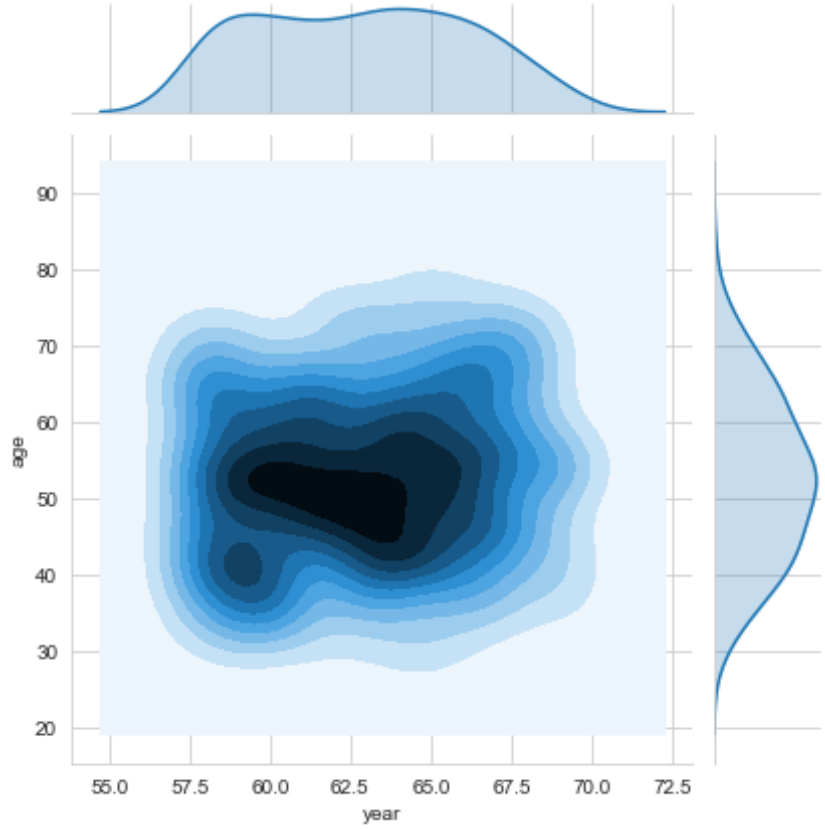
In [31]:
```
sns.jointplot(x="age",y="year",data=haber,kind="kde")
plt.show()
```



In [32]:
```
sns.jointplot(x="age",y="nodes",data=haber,kind="kde")
plt.show()
```



In [33]:
```
sns.jointplot(x="year",y="age",data=haber,kind="kde")
plt.show()
```



In [ ]: