haberman

October 3, 2018

1 Haberman Data Set

Haberman Data Set(https://www.kaggle.com/gilsousa/habermans-survival-data-set)

```
In [3]: import numpy as np
        import pandas as pd
        import seaborn as sns
        import matplotlib.pyplot as plt

        ''' download the haberman data set from the link https://www.kaggle.com/gilsousa/haberr
        # read haberman data from the haberman.csv file
        haberman = pd.read_csv("haberman.csv")
        haberman.head(10)
```

Out[3]:

	30	64	1	1.1
0	30	62	3	1
1	30	65	0	1
2	31	59	2	1
3	31	65	4	1
4	33	58	10	1
5	33	60	0	1
6	34	59	0	2
7	34	66	9	2
8	34	58	30	1
9	34	60	1	1

column '30' contains the age of the persons column '60' contains the year on which the operation is performed column '1' contains the number of positive auxilary node detected column '1.1' contains the Survival status (class attribute) 1 = the patient survived 5 years or longer 2 = the patient died within 5 year

```
In [10]: #how many data points and features
         print(haberman.shape)
```

```
(305, 4)
```

so there are 305 data points and 4 features

```
In [11]: # print the features name in the data set
         print(haberman.columns)
```

```
Index(['30', '64', '1', '1.1'], dtype='object')
```

```
In [13]: # How many data points of each classes

         haberman['1.1'].value_counts()
```

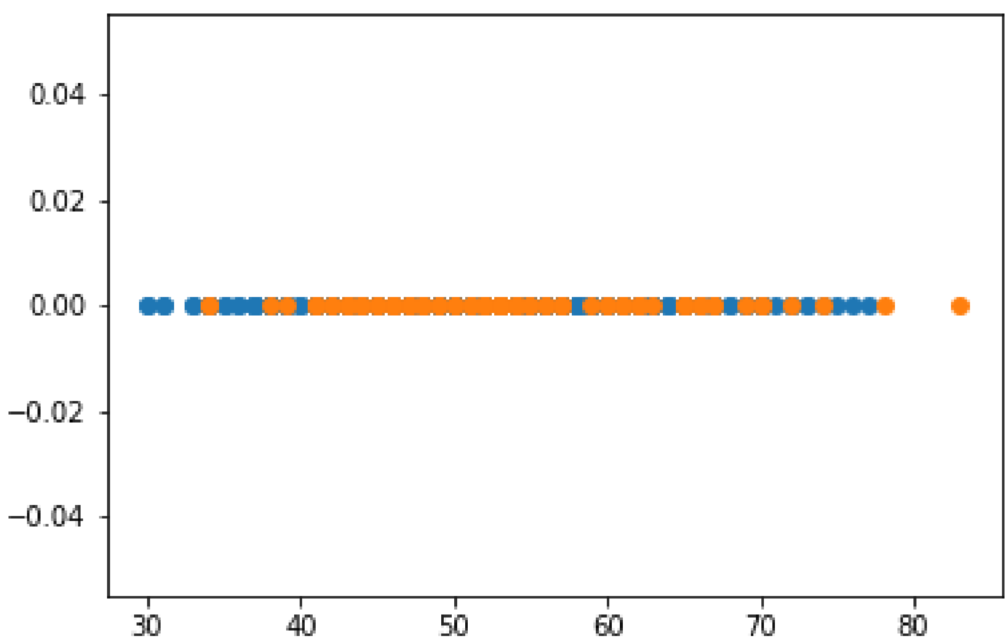
```
Out[13]: 1    224
         2     81
         Name: 1.1, dtype: int64
```

conclusion it have two classes class '1' have 224 datapoints that means 224 people had survived class '2' have 81 datapoints that means 81 people had died It have imbalanced datasets class '1' is approx thrice of class '2'

Objective - it is to check wheather a new data set will fall in class '1' or '2' that mean to predict the survival class of the new patient wheather he/she will die or live a long life

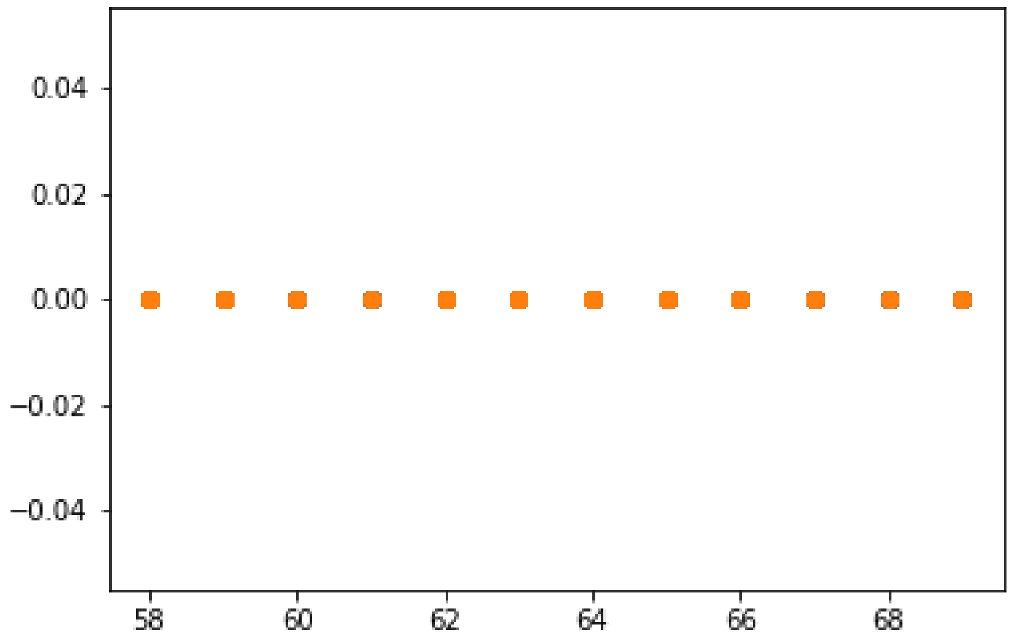
1.1 Univariate Analysis

```
In [22]: # 1-D scatter plot of age
         haberman1 = haberman.loc[haberman["1.1"] == 1];
         haberman2 = haberman.loc[haberman["1.1"] == 2];
         plt.plot(haberman1["30"], np.zeros_like(haberman1['30']), 'o')
         plt.plot(haberman2["30"], np.zeros_like(haberman2['30']), 'o')
         plt.show();
```



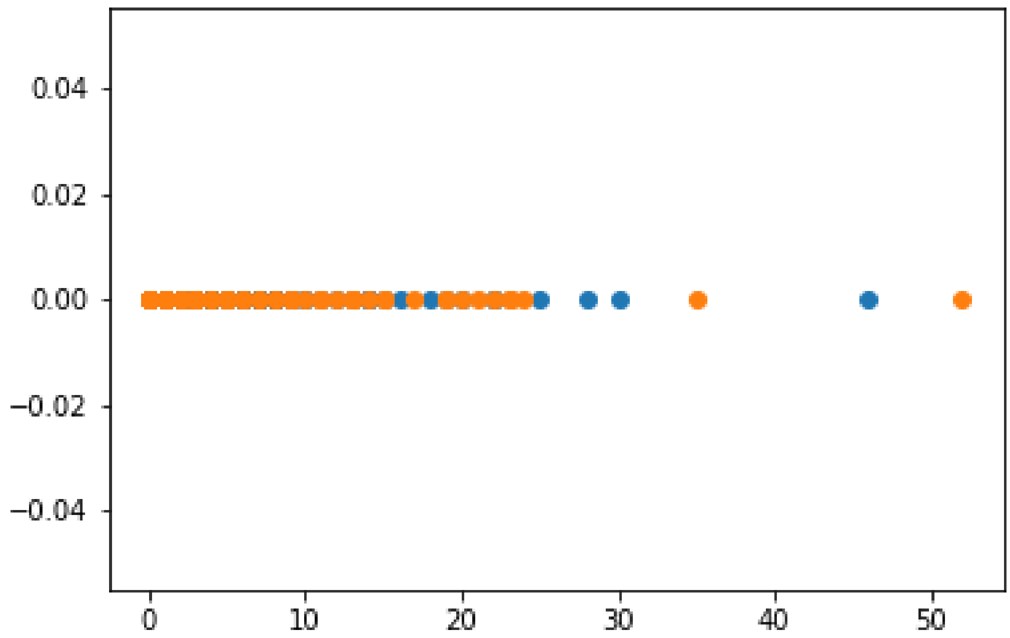
conclusion if age is less than 33 then patient will survive and if the age is greater than 82 patient will die

```
In [23]: # 1-D scatter plot of operation year
         plt.plot(haberman1["64"], np.zeros_like(haberman1['64']), 'o')
         plt.plot(haberman2["64"], np.zeros_like(haberman2['64']), 'o')
         plt.show();
```



conclusion patient year of operation is not of any use

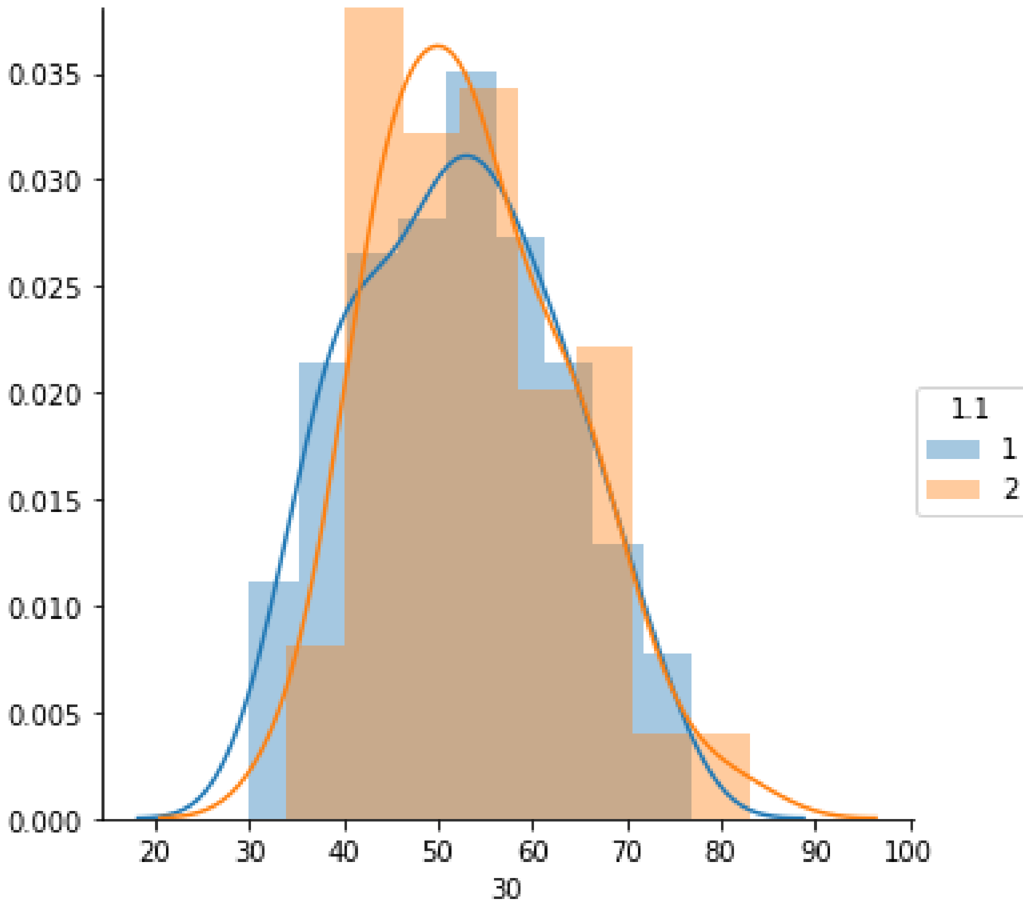
```
In [25]: # 1-D scatter plot of number of positive auxilary node
         plt.plot(haberman1["1"], np.zeros_like(haberman1['1']), 'o')
         plt.plot(haberman2["1"], np.zeros_like(haberman2['1']), 'o')
         plt.show();
```



conclusion if auxilary node is less than 14 then patient will survive

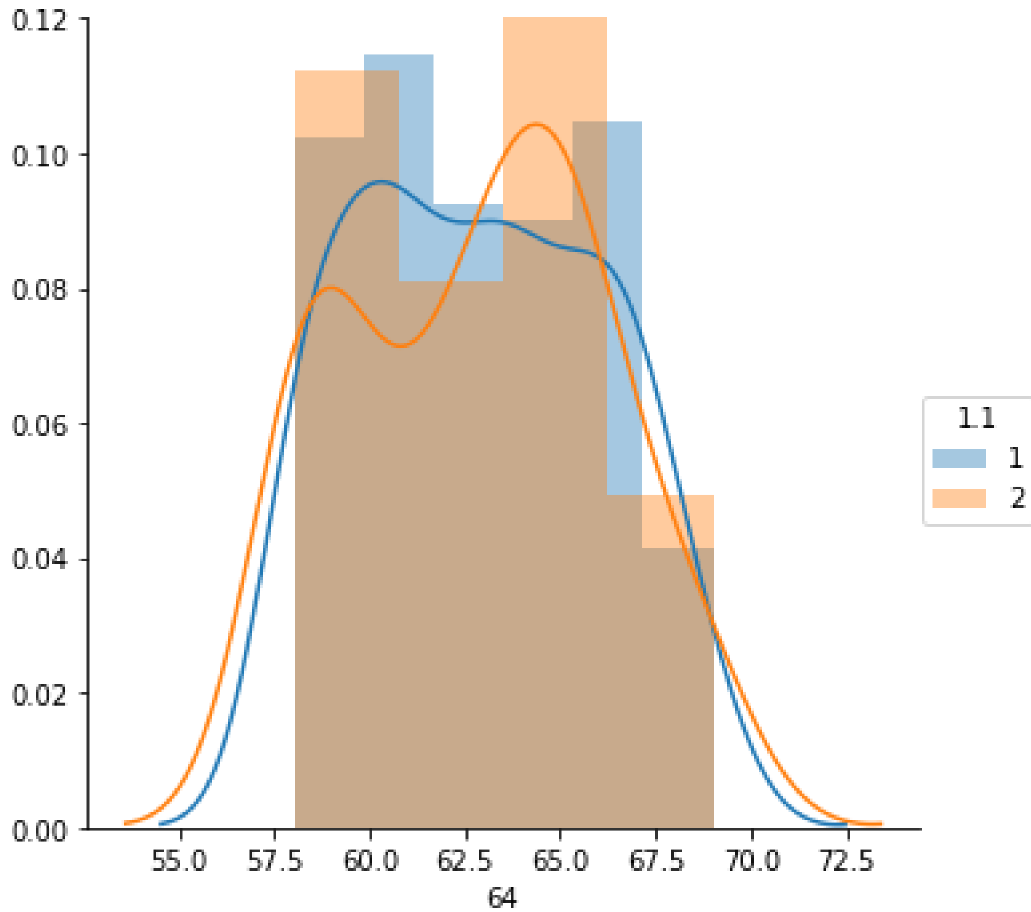
1.2 PDF and Histogram

```
In [4]: #pdf and histogram with respect to age
        import warnings
        warnings.filterwarnings("ignore")
        sns.FacetGrid(haberman, hue="1.1", size=5) \
           .map(sns.distplot, "30") \
           .add_legend();
        plt.show();
```



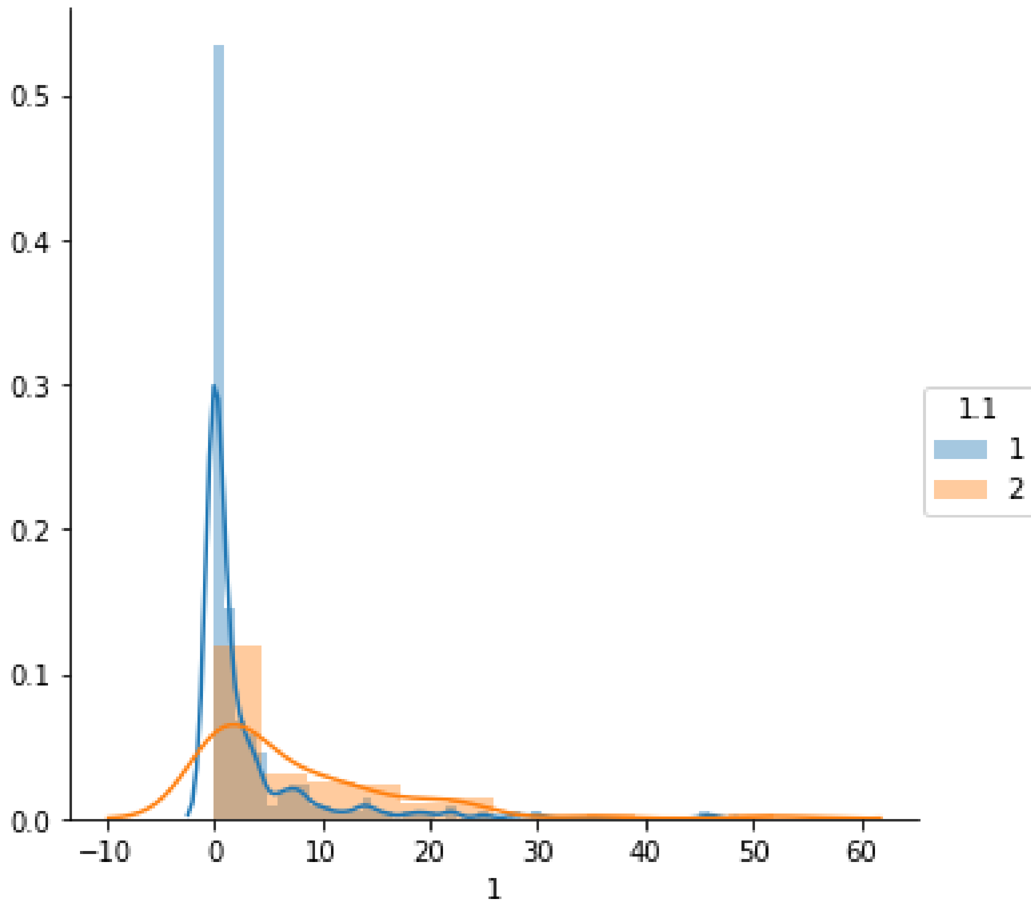
conclusion with respect to age there are lot of overlapping so it not an important features

```
In [5]: #pdf and histogram with respect to operation year

        sns.FacetGrid(haberman, hue="1.1", size=5) \
           .map(sns.distplot, "64") \
           .add_legend();
        plt.show();
```



conclusion with respect to operation year there are lot of overlapping so it not an important features

```
In [6]: #pdf and histogram with respect to number of positive auxilary nodes
        sns.FacetGrid(haberman, hue="1.1", size=5) \
           .map(sns.distplot, "1") \
           .add_legend();
        plt.show();
```



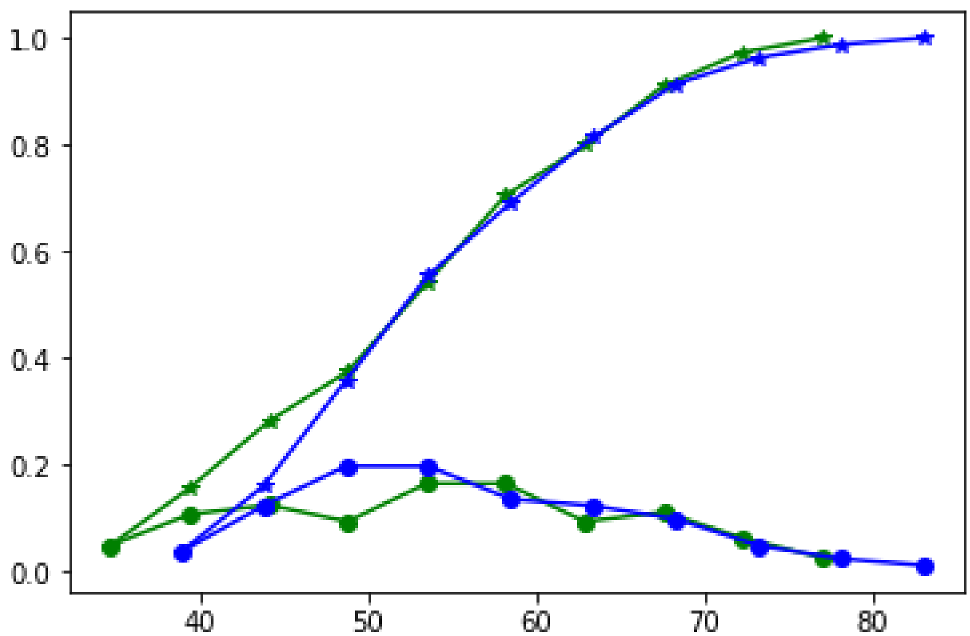
conclusion with respect to age there are lot of overlapping so it is difficult to find that much amount of information but if the number of auxilary nodes is less than 4 the chances of survival is maximum as compared to death of the patient

1.3 CDF and PDF plots

```
In [36]: # Plots of CDF of age only

         # pdf and cdf of survived patient
         counts, bin_edges = np.histogram(haberman1['30'], bins=10,
                                          density = True)
         pdf = counts/(sum(counts))
         print(pdf);
         print(bin_edges)
         cdf = np.cumsum(pdf)
         plt.plot(bin_edges[1:],pdf,"go-")
         plt.plot(bin_edges[1:], cdf,"g*-")


         # pdf and cdf of patient who had died
```

```
counts, bin_edges = np.histogram(haberman2['30'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,"bo-")
plt.plot(bin_edges[1:], cdf,"b*-")

plt.show();
```

```
[0.04910714 0.10714286 0.125      0.09375    0.16517857 0.16517857
 0.09375    0.11160714 0.0625     0.02678571]
[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. ]
```

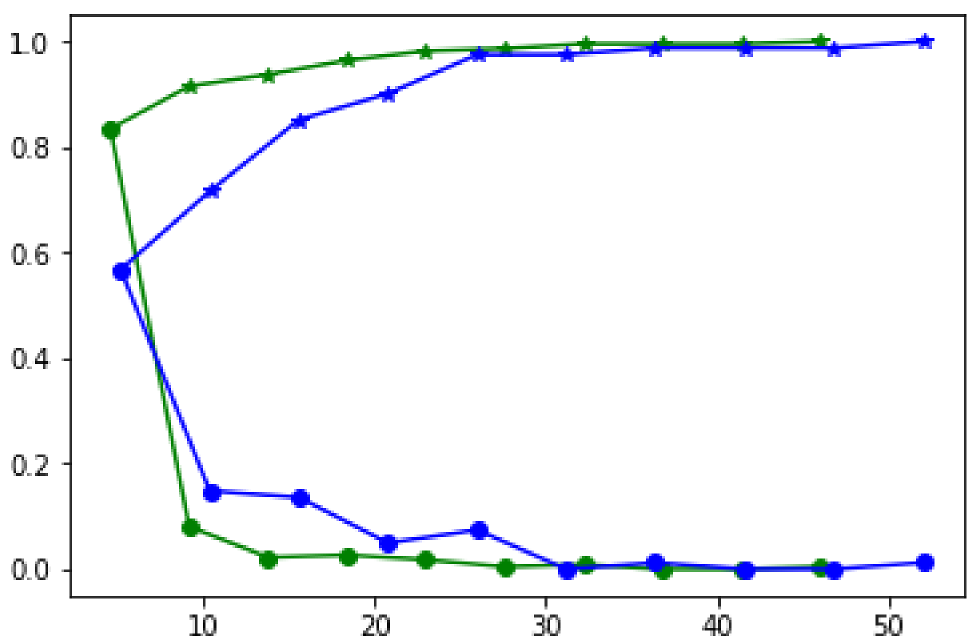


conclusion if the age of patient is less than 38 then it is 100 percent sure that the patient will survive

In [37]:
```
# Plots of CDF of number of positive auxilary nodes only

# pdf and cdf of survived patient
counts, bin_edges = np.histogram(haberman1['1'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
```

```
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,"go-")
plt.plot(bin_edges[1:], cdf,"g*-")


# pdf and cdf of patient who had died
counts, bin_edges = np.histogram(haberman2['1'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,"bo-")
plt.plot(bin_edges[1:], cdf,"b*-")

plt.show();
```

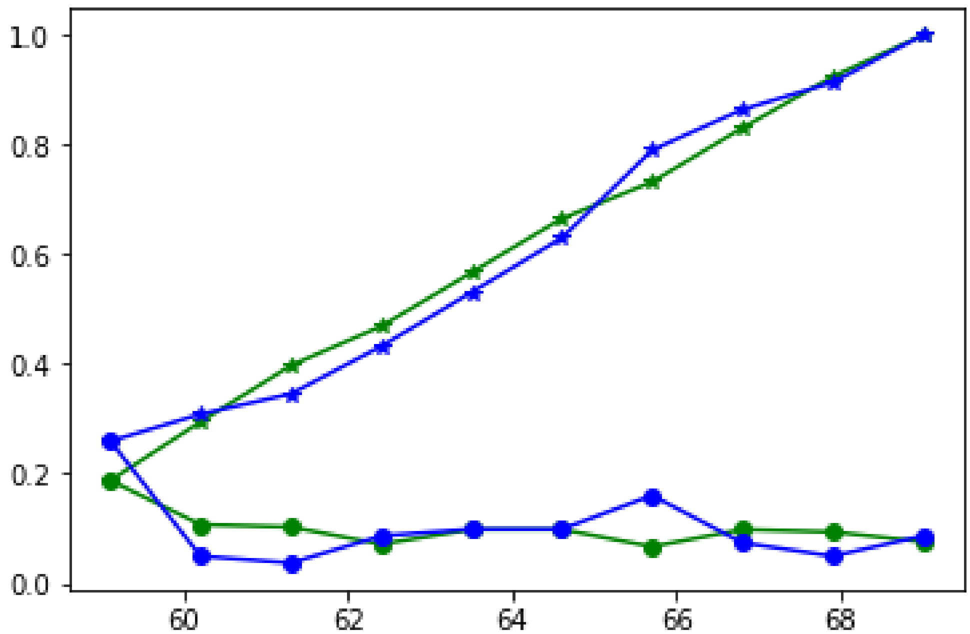
```
[0.83482143 0.08035714 0.02232143 0.02678571 0.01785714 0.00446429
 0.00892857 0.         0.         0.00446429]
[ 0.   4.6  9.2 13.8 18.4 23.  27.6 32.2 36.8 41.4 46. ]
[0.56790123 0.14814815 0.13580247 0.04938272 0.07407407 0.
 0.01234568 0.         0.         0.01234568]
[ 0.   5.2 10.4 15.6 20.8 26.  31.2 36.4 41.6 46.8 52. ]
```



conclusion: if number of auxilary node is greater than 45 than the patient will die

In [38]:
```
# Plots of CDF of year of operations

# pdf and cdf of survived patient
counts, bin_edges = np.histogram(haberman1['64'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,"go-")
plt.plot(bin_edges[1:], cdf,"g*-")


# pdf and cdf of patient who had died
counts, bin_edges = np.histogram(haberman2['64'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf,"bo-")
plt.plot(bin_edges[1:], cdf,"b*-")

plt.show();
```

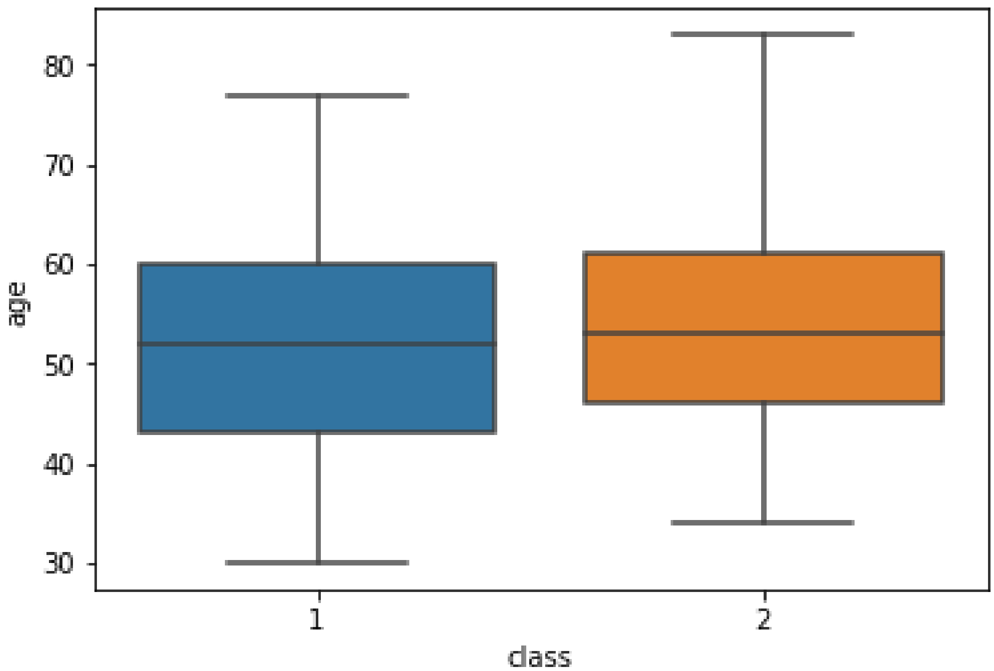
```
[0.1875     0.10714286 0.10267857 0.07142857 0.09821429 0.09821429
 0.06696429 0.09821429 0.09375    0.07589286]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
[0.25925926 0.04938272 0.03703704 0.08641975 0.09876543 0.09876543
 0.16049383 0.07407407 0.04938272 0.08641975]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
```



conclusion: Nothing useful is found

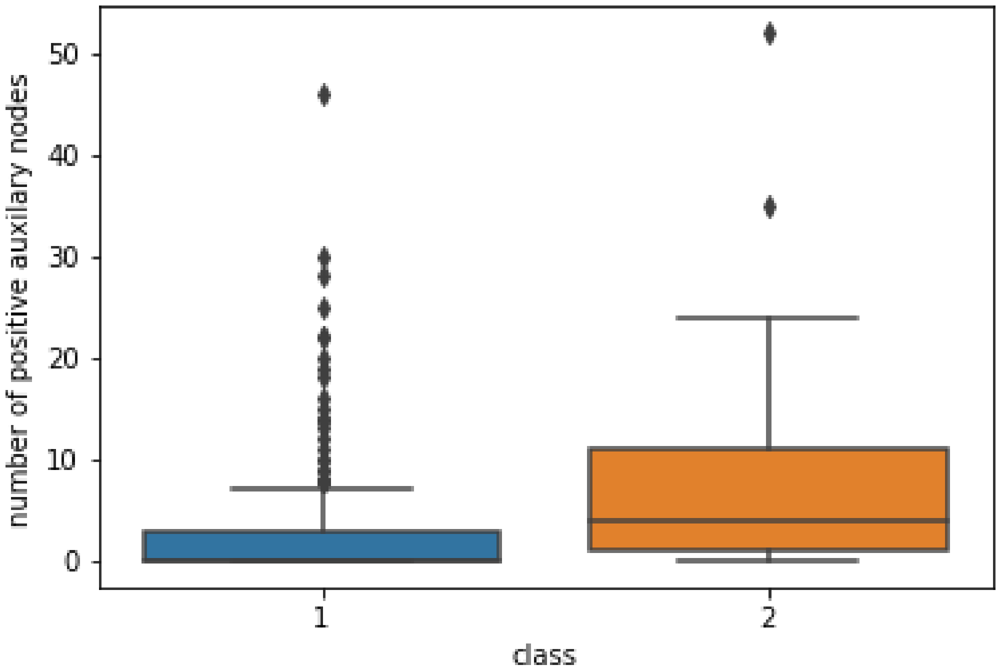
1.4 Box Plot with Whiskers

```
In [41]: # Box plot against age
         sns.boxplot(x='1.1',y='30', data=haberman)
         plt.xlabel('class')
         plt.ylabel('age')
         plt.show()
```



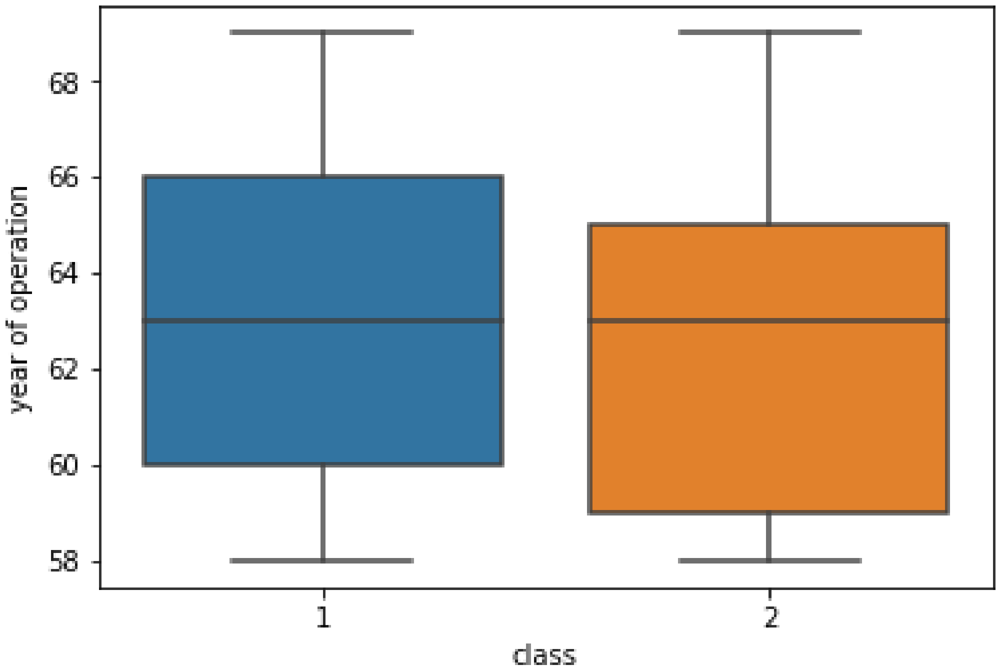
conclusion: 1. Below the age of 45 patient will survive and after the age of 60 patient will die 2. At the age of 52 50 percent of patient will survive and 40

```
In [42]: # Box plot against positive number of auxilary nodes
         sns.boxplot(x='1.1',y='1', data=haberman)
         plt.xlabel('class')
         plt.ylabel('number of positive auxilary nodes')
         plt.show()
```



conclusion: if number of positive auxilary nodes is less than 4 less than 75if number of positive auxilary nodes is more than 9 then chances of patient death is around more than 60

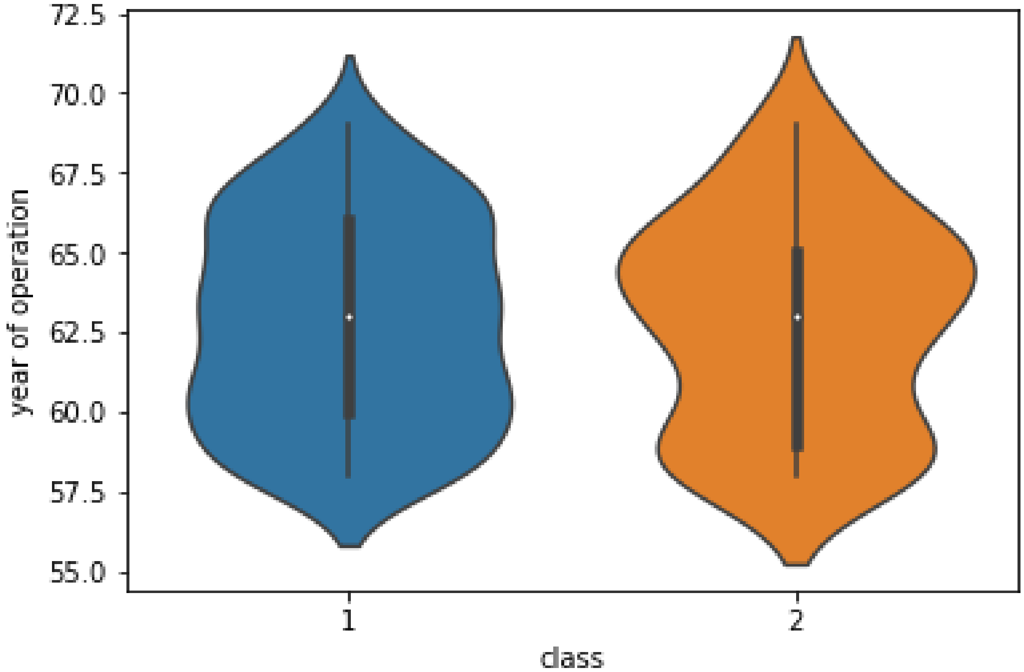
In [43]:
```
# Box plot year of operation
sns.boxplot(x='1.1',y='64', data=haberman)
plt.xlabel('class')
plt.ylabel('year of operation')
plt.show()
```



conclusion operation done in year 60 more than 25

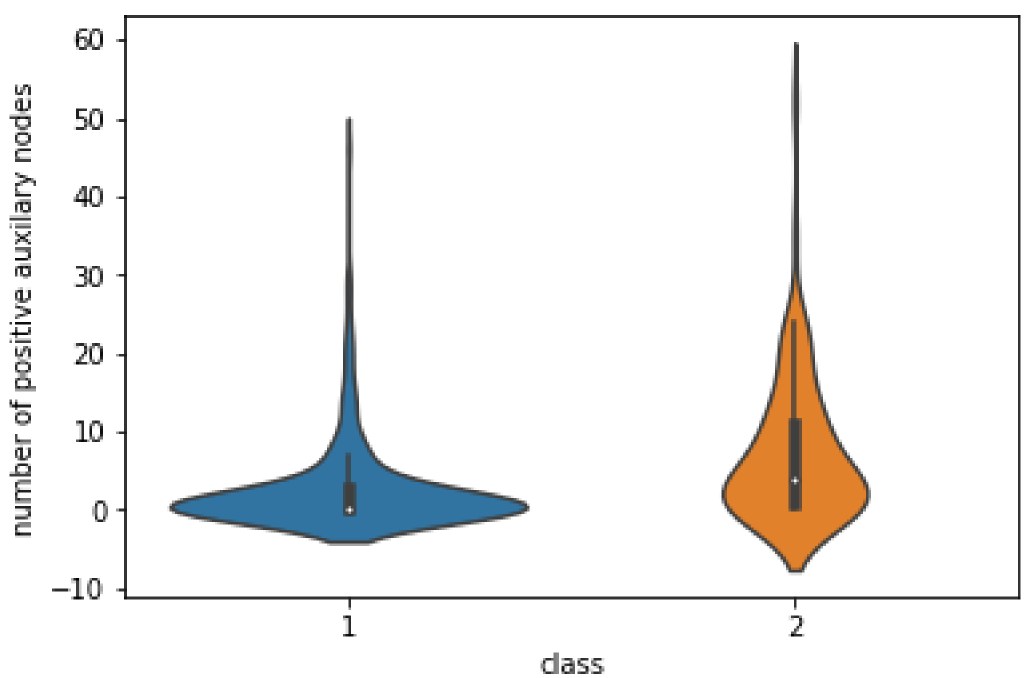
1.5 Violin Plots

In [44]:
```
# Violin plot year of operation
sns.violinplot(x='1.1',y='64', data=haberman)
plt.xlabel('class')
plt.ylabel('year of operation')
plt.show()
```



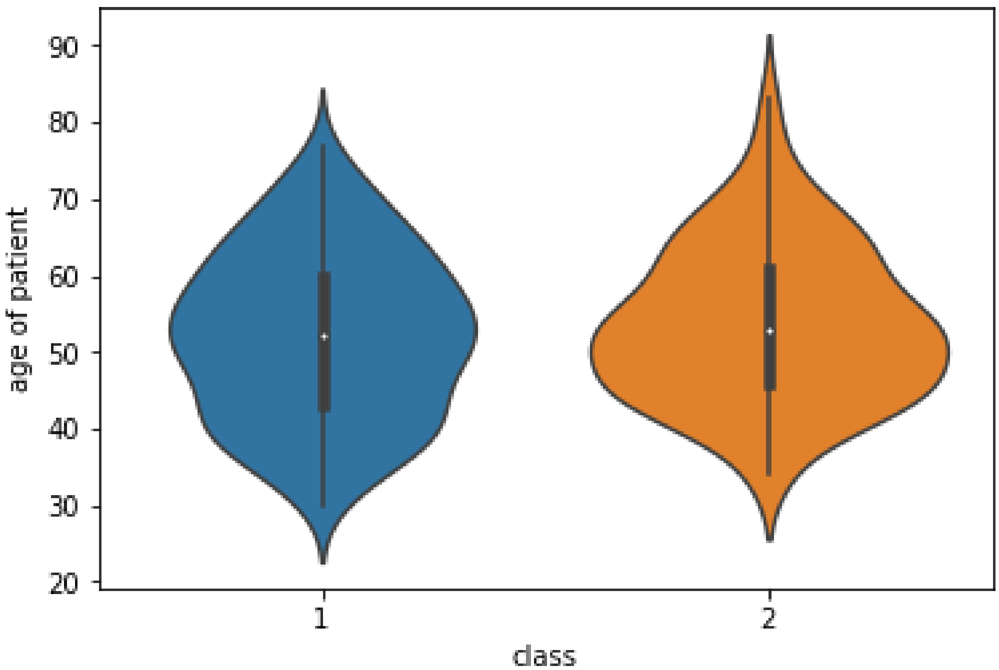
conclusion: both plots are almost similar to each other so nothing is concluded

```
In [46]: # Violin plot number of positive auxilary nodes
         sns.violinplot(x='1.1',y='1', data=haberman)
         plt.xlabel('class')
         plt.ylabel('number of positive auxilary nodes')
         plt.show()
```



conclusion: the pdf is more denser in class '1' the pdf of class '2' is not so denser

```
In [47]: # Violin plot number of age
         sns.violinplot(x='1.1',y='30', data=haberman)
         plt.xlabel('class')
         plt.ylabel('age of patient')
         plt.show()
```

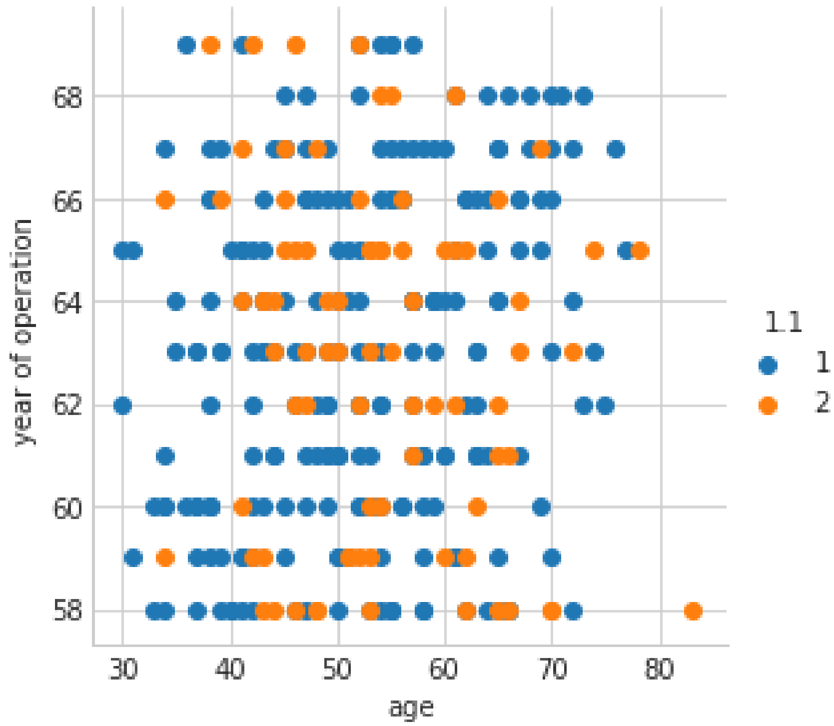


conclusion: both of the plots are almost similar so nothing is concluded

2 Bivariate Classification

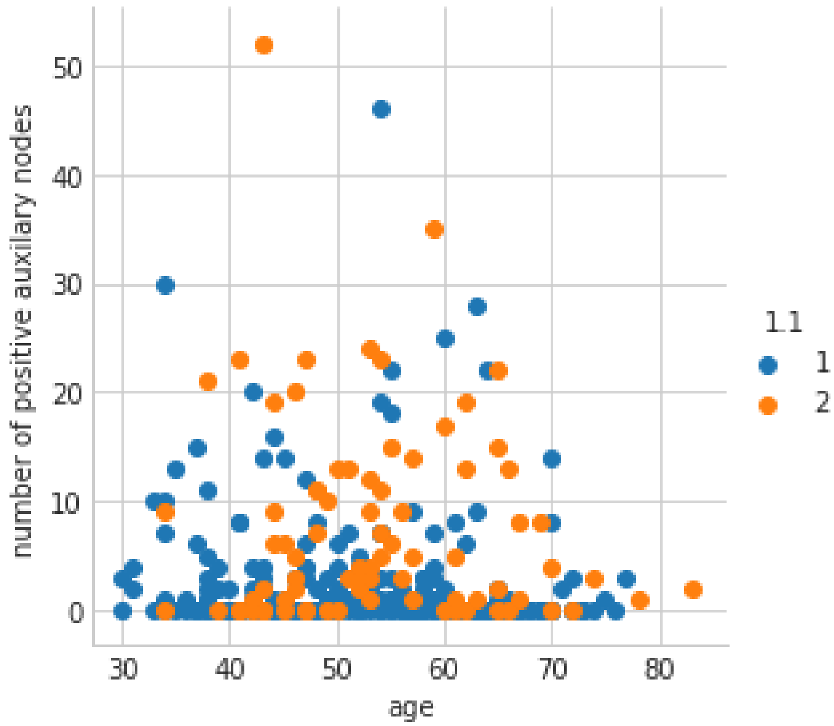
2.1 2-D scatter plot

```
In [50]: # 2-D Scatter plot with color-coding for each live/dead class.
         # age is at x-axis
         # year of operation at y-axis
         sns.set_style("whitegrid");
         sns.FacetGrid(haberman, hue="1.1", size=4) \
            .map(plt.scatter, "30", "64") \
            .add_legend();
         plt.xlabel('age');
         plt.ylabel('year of operation');
         plt.show();
```



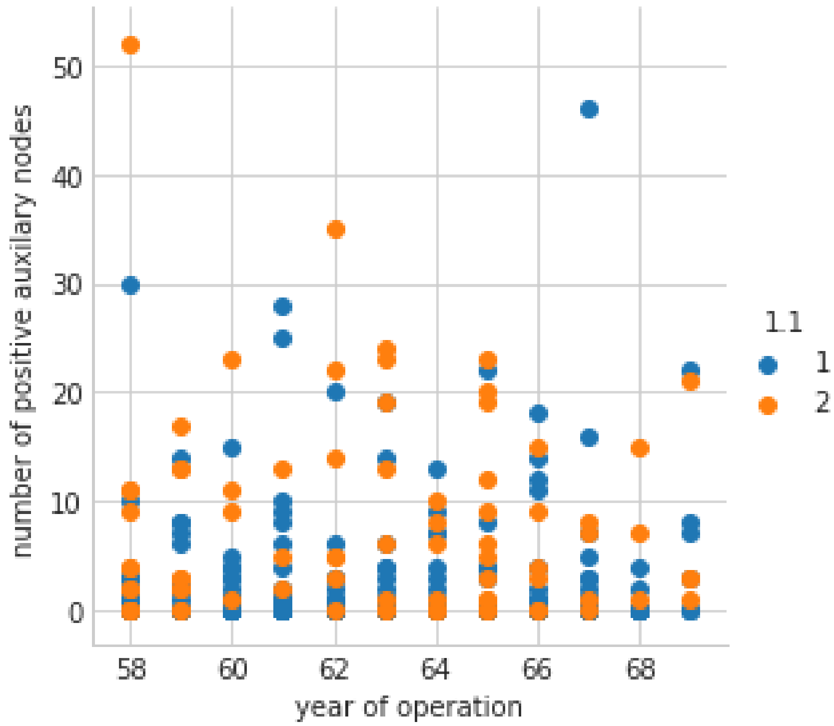
conclusion: all the doth are scattering all over the plane intersecting with others dots of different class. So, nothing useful is concluded

In [52]:
```
# 2-D Scatter plot with color-coding for each live/dead class.
# age is at x-axis
# number of positive auxilary nodes at y-axis
sns.set_style("whitegrid");
sns.FacetGrid(haberman, hue="1.1", size=4) \
   .map(plt.scatter, "30", "1") \
   .add_legend();
plt.xlabel('age');
plt.ylabel('number of positive auxilary nodes');
plt.show();
```



conclusion: Nothing useful concluded as dots of different classes intersect with each other resulting nothing important

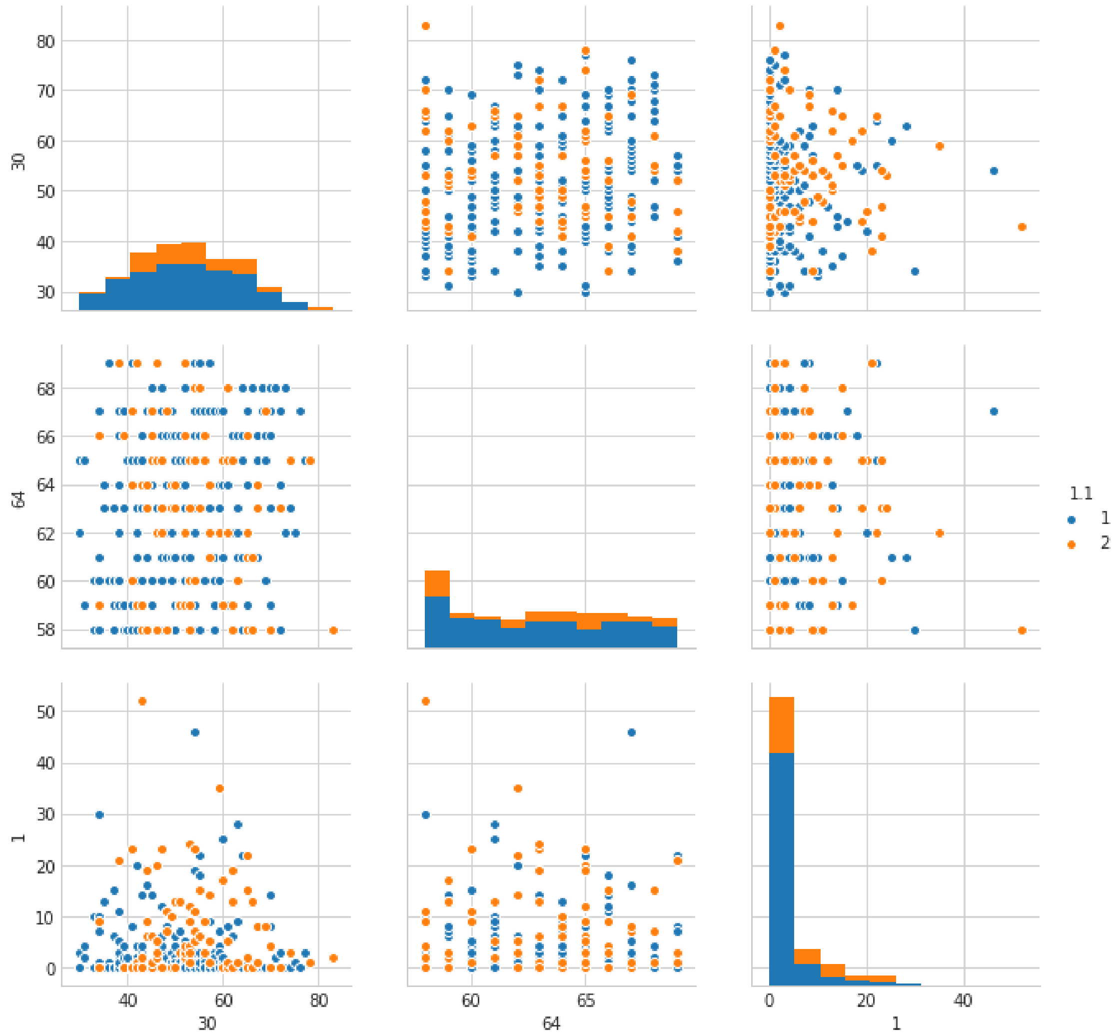
In [53]:
```
# 2-D Scatter plot with color-coding for each live/dead class.
# year of operation is at x-axis
# number of positive auxilary nodes at y-axis
sns.set_style("whitegrid");
sns.FacetGrid(haberman, hue="1.1", size=4) \
   .map(plt.scatter, "64", "1") \
   .add_legend();
plt.xlabel('year of operation');
plt.ylabel('number of positive auxilary nodes');
plt.show();
```



conclusion: Nothing useful concluded as dots of different classes intersect with each other resulting nothing important

2.2 Pair Plots

```
In [59]: # pairwise scatter plot: Pair-Plot
         # Dis-advantages:
         ##Can be used when number of features are high.
         ##Cannot visualize higher dimensional patterns in 3-D and 4-D.
         #Only possible to view 2D patterns.
         plt.close();
         sns.set_style("whitegrid");
         sns.pairplot(haberman, hue="1.1", vars=['30','64','1'], size =3);
         plt.show()
         # NOTE: the diagnol elements are PDFs for each feature. PDFs are expalined below.
```



conclusion: Nothing good is predicted from the pair plots as there are a lot intersection in the plots

2.3 Final Conclusions

1. if auxilary node is less than 14 then patient will survive
2. Below the age of 45 patient will survive and after the age of 60 patient will die
3. if number of auxilary node is greater than 45 than the patient will die
4. if the age of patient is less than 38 then it is 100 percent sure that the patient will survive
5. if age is less than 33 then patient will survive and if the age is greater than 82 patient will die