chandanachandu124@gmail.com_assignment1

June 7, 2019

```
In [25]: import warnings
         warnings.filterwarnings("ignore")
         import pandas as pd
         import seaborn as sns
         import matplotlib.pyplot as mp
         import numpy as np

         hm=pd.read_csv("haberman.csv")
```

```
In [29]: print(hm.shape)
```

```
(306, 4)
```

''' the observed number of points and features of the data set are shown above'''

```
In [30]: print(hm.columns)
```

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

'''the data set contains the names of the columns as shown above'''

```
In [33]: hm.head(10)
```

Out[33]:

	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

```
In [34]: hm.tail(10)
```

```
Out[34]:      age  year  nodes  status
          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
```

```
In [7]: hm["age"].value_counts()
```

```
Out[7]: 52    14
        54    13
        50    12
        47    11
        53    11
        43    11
        57    11
        55    10
        65    10
        49    10
        38    10
        41    10
        61     9
        45     9
        42     9
        63     8
        59     8
        62     7
        44     7
        58     7
        56     7
        46     7
        70     7
        34     7
        48     7
        37     6
        67     6
        60     6
        51     6
        39     6
        66     5
        64     5
        72     4
        69     4
```

```
40     3
30     3
68     2
73     2
74     2
36     2
35     2
33     2
31     2
78     1
71     1
75     1
76     1
77     1
83     1
Name: age, dtype: int64
```

```
In [8]: hm["year"].value_counts()
```

```
Out[8]: 58    36
        64    31
        63    30
        66    28
        65    28
        60    28
        59    27
        61    26
        67    25
        62    23
        68    13
        69    11
        Name: year, dtype: int64
```

```
In [9]: hm["nodes"].value_counts()
```

```
Out[9]: 0     136
        1      41
        2      20
        3      20
        4      13
        6       7
        7       7
        8       7
        5       6
        9       6
        13      5
        14      4
        11      4
        10      3
```

```
15      3
19      3
22      3
23      3
12      2
20      2
46      1
16      1
17      1
18      1
21      1
24      1
25      1
28      1
30      1
35      1
52      1
Name: nodes, dtype: int64
```

```
In [28]: hm["status"].value_counts()
```

```
Out[28]: 1    225
         2     81
         Name: status, dtype: int64
```

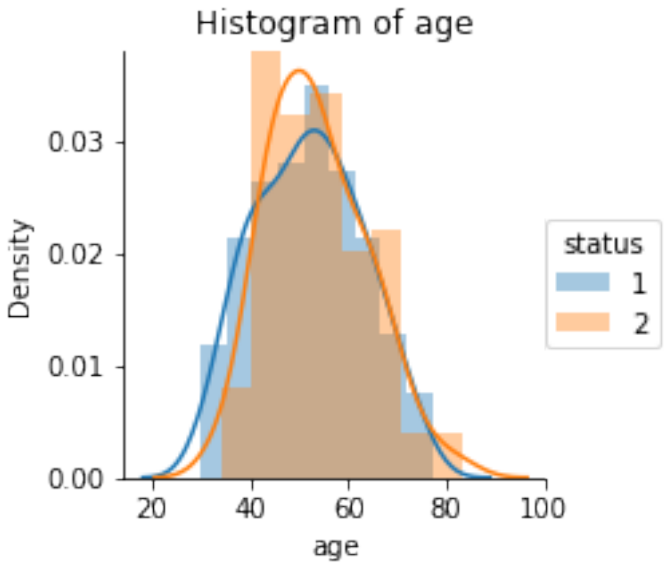
1 Objective

"'by the given data it is found that the data set contains columns named age, year, nodes and status. where the status represents weather the patient lives for more than 5 years or less than 5years, if the survival status is 1 then the patient may survive for 5 years or longer if the status is 2 then the patient is dead within 5 years

here our main objective is to find wheather the patient will survive for more than 5 years or not based upon the patients age, year of treatment and the number of lymph nodes."'

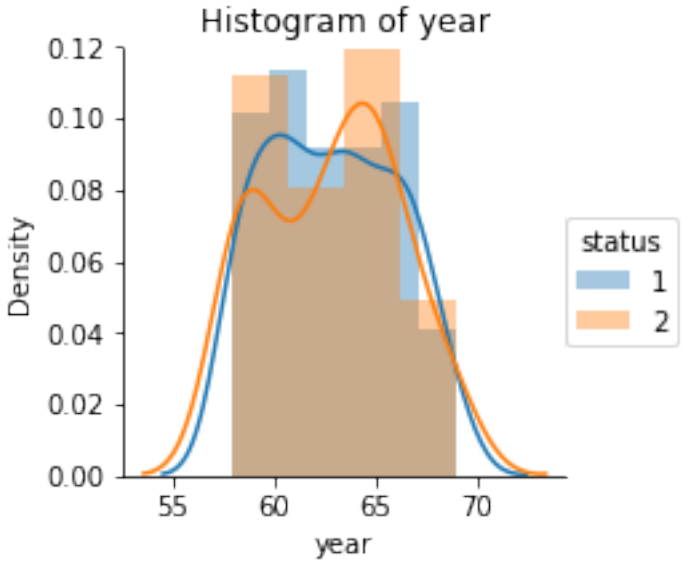
2 Probability Density Function

```
In [40]: sns.FacetGrid(hm, hue="status", size=3) \
             .map(sns.distplot,"age") \
             .add_legend();
         plt.title("Histogram of age")
         plt.ylabel("Density")
         plt.show();
```



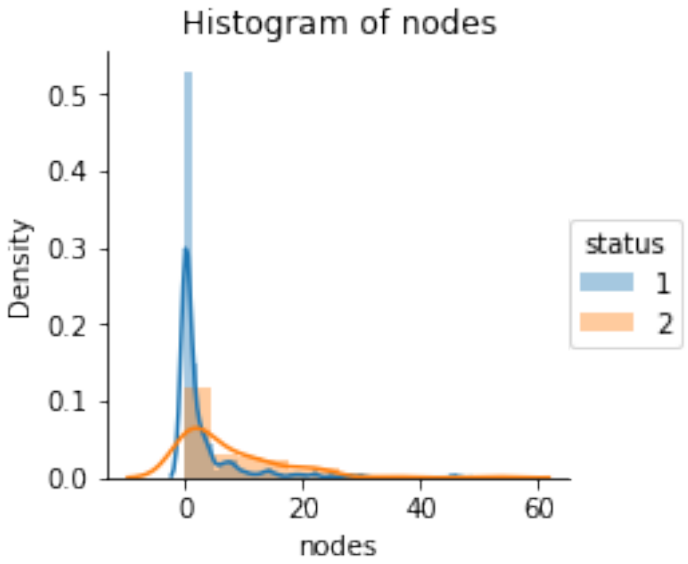
the histogram is drawn between age and density and we cna see the the points are overlapped and some of the blue points got seperated and some sort of conclusion cna be drawm through this,.

```
In [42]: sns.FacetGrid(hm, hue="status", size=3) \
             .map(sns.distplot, "year") \
             .add_legend();
         plt.title("Histogram of year")
         plt.ylabel("Density")
         plt.show();
```



this is a histogram betweendensity and year where the the points are overlapped massively.

```
In [44]: sns.FacetGrid(hm, hue="status", size=3) \
            .map(sns.distplot, "nodes") \
            .add_legend();
         plt.title("Histogram of nodes")
         plt.ylabel("Density")
         plt.show();
```

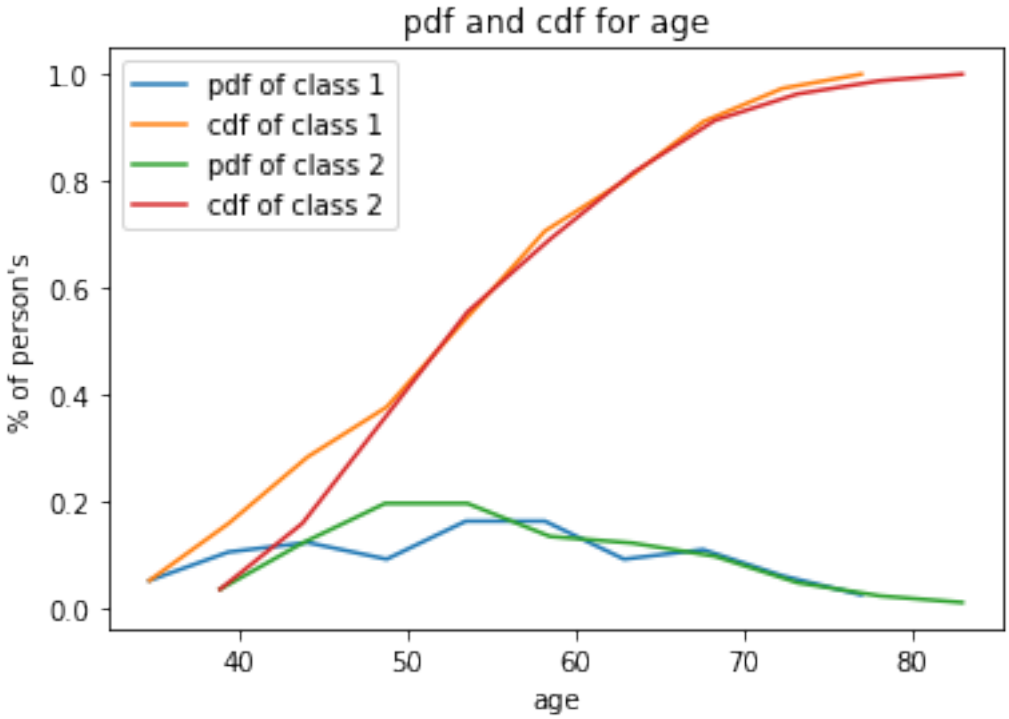


In all the above plots we can see that the features are overlapping each other masively, we can observe that people having 0-5 nodes are 58% survived and 12% died.

3 Cummulative distributive function

```
In [50]: one = hm.loc[hm["status"] == 1]
         two = hm.loc[hm["status"] == 2]
         label = ["pdf of class 1", "cdf of class 1", "pdf of class 2", "cdf of class 2"]
         counts, bin_edges = np.histogram(one["age"], bins=10, density = True)
         pdf = counts/(sum(counts))
         cdf = np.cumsum(pdf)
         plt.title("pdf and cdf for age")
         plt.xlabel("age")
         plt.ylabel("% of person's")
         plt.plot(bin_edges[1:], pdf)
         plt.plot(bin_edges[1:], cdf)

         counts, bin_edges = np.histogram(two["age"], bins=10, density = True)
         pdf = counts/(sum(counts))
```

```
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:], pdf)
plt.plot(bin_edges[1:], cdf)
plt.legend(label)

plt.show()
```



In the above plot class 1 means survived and class 2 means not survived, cdf gives the cumulative probability associated with a function. the cumulative sum of area under curve gives the cdf.

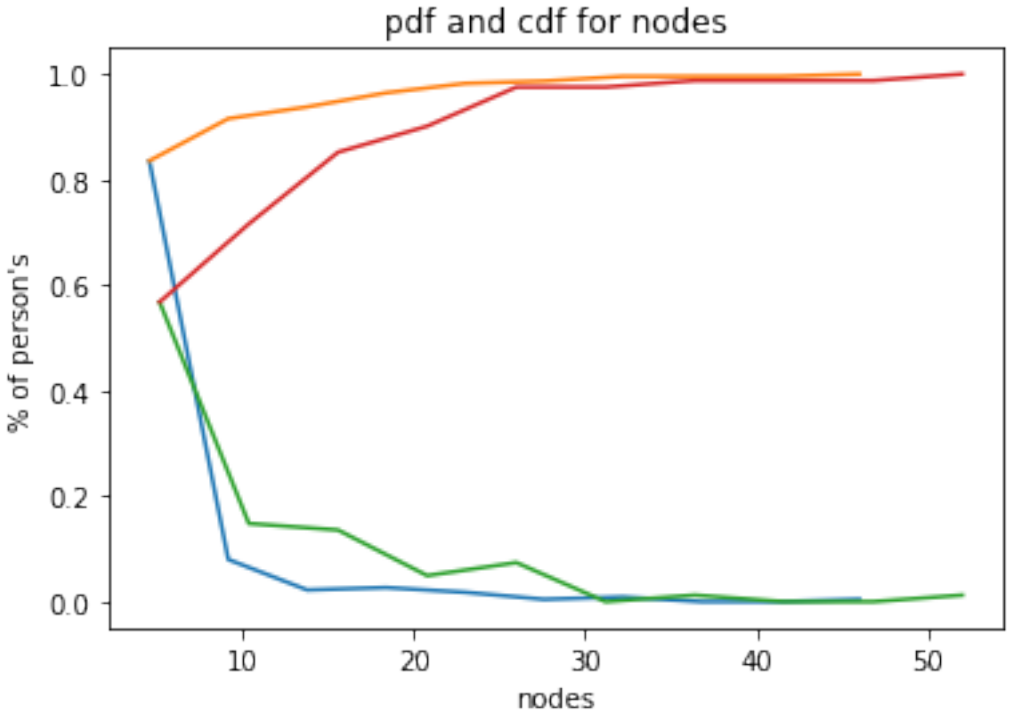
```
In [52]: label = ["pdf of class 1", "cdf of class 1", "pdf of class 2", "cdf of class 2"]
         counts, bin_edges = np.histogram(one["year"], bins=10, density = True)
         pdf = counts/(sum(counts))
         cdf = np.cumsum(pdf)
         plt.plot(bin_edges[1:], pdf)
         plt.plot(bin_edges[1:], cdf)



         counts, bin_edges = np.histogram(two["year"], bins=10, density = True)
         pdf = counts/(sum(counts))
         cdf = np.cumsum(pdf)
         plt.title("pdf and cdf for operation year")
         plt.xlabel("year")
```

```
plt.ylabel("% of person's")
plt.plot(bin_edges[1:], pdf)
plt.plot(bin_edges[1:], cdf)
plt.legend(label)

plt.show();
```

In the above plot class 1 means survived and class 2 means not survived, cdf gives the cumulative probability associated with a function. the cumulative sum of area under curve gives the cdf and proper conclusion cannot be drawn through this.

```
In [54]: label = ["pdf of class 1", "cdf of class 1", "pdf of class 2", "cdf of class 2"]
         counts, bin_edges = np.histogram(one["nodes"], bins=10, density = True)
         pdf = counts/(sum(counts))
         cdf = np.cumsum(pdf)
         plt.plot(bin_edges[1:], pdf)
         plt.plot(bin_edges[1:], cdf)

         counts, bin_edges = np.histogram(two["nodes"], bins=10, density = True)
         pdf = counts/(sum(counts))
         cdf = np.cumsum(pdf)
         plt.title("pdf and cdf for nodes")
         plt.xlabel("nodes")
         plt.ylabel("% of person's")
         plt.plot(bin_edges[1:], pdf)
```

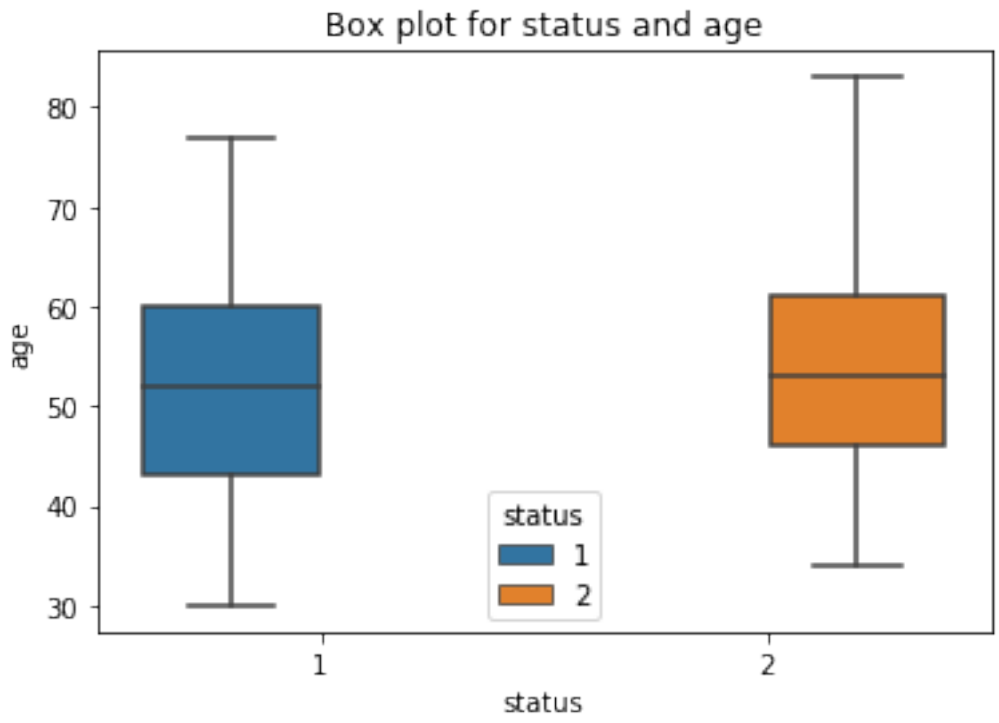
```
plt.plot(bin_edges[1:], cdf)

plt.show();
```



According to the plots we can observe that 15% of the persons who are survived are under or equal to the age group of 37 and persons who are having more than 46 nodes are not survived.

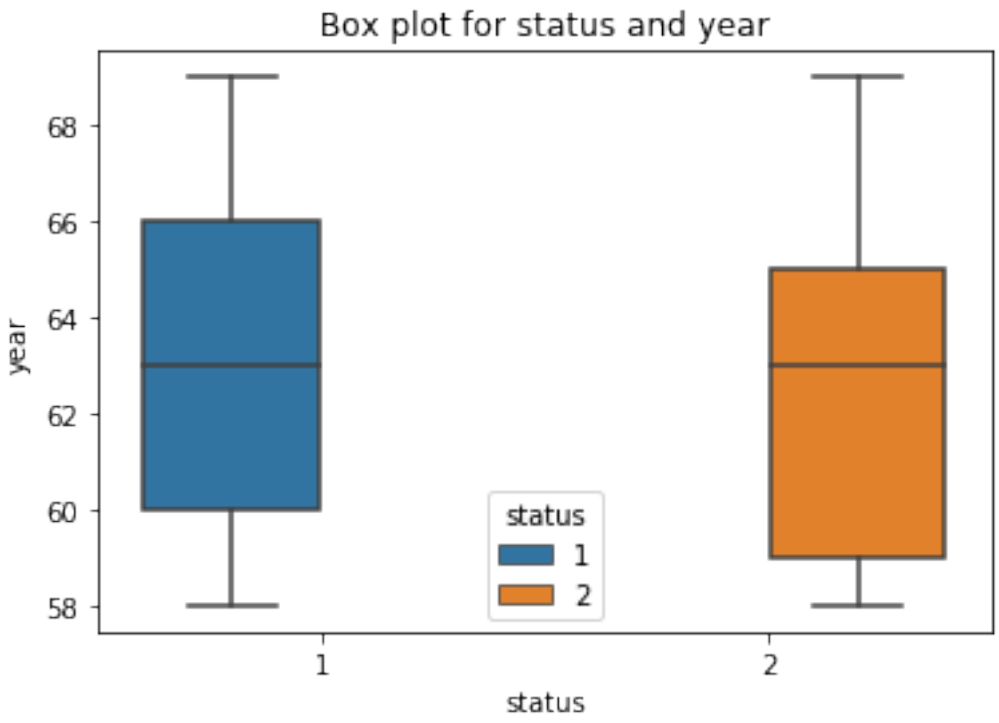
4 Box plot

```
In [58]: sns.boxplot(x = "status", y = "age", hue = "status", data = hm).set_title("Box plot f
         plt.show()
```



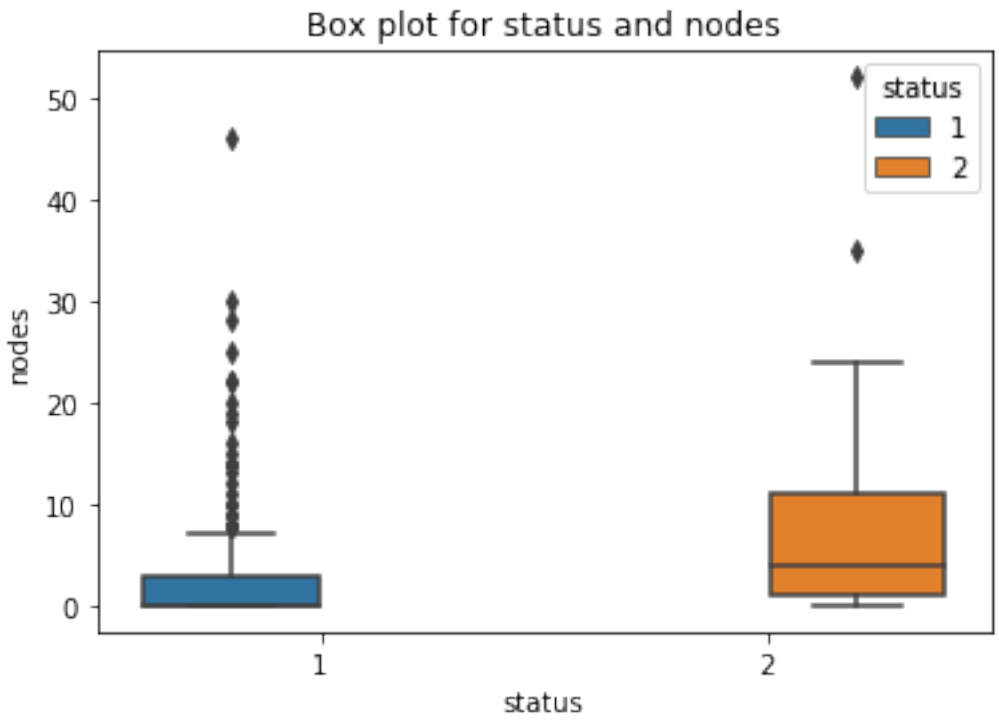
box plots generally gives the statistical summary of data, the horizontal line inside the curve represents the median and t the above box plot is drawn between age and status.where the survived people lies between 50 to 60 and the people dies lies between 55-60.

```
In [59]: sns.boxplot(x = "status", y = "year", hue = "status", data = hm).set_title("Box plot
         plt.show()
```



the box plot between year and status is drawn and the survived people lies between the year 60-66 and not survived people between the years 59-65.

```
In [60]: sns.boxplot(x = "status", y = "nodes", hue = "status", data = hm).set_title("Box plot
         plt.show()
```



here the box plot between nodes and status says that the max points of survival lies between nodes o to 1 and people who are not survived lies from nodes 1 -10

5 Violin plots

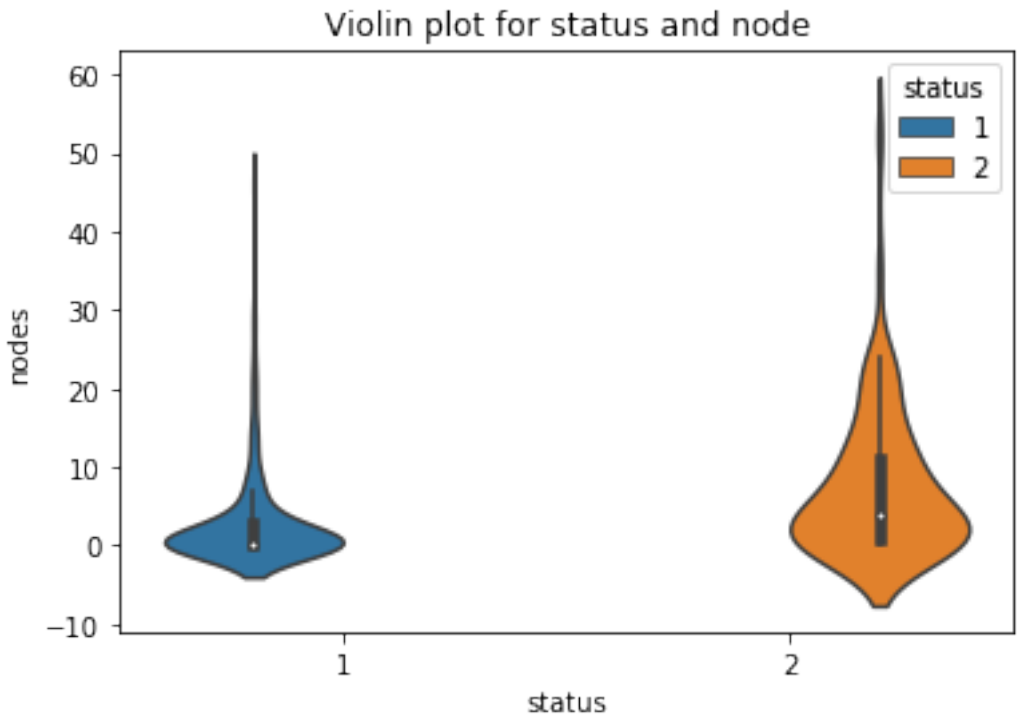
```
In [61]: sns.violinplot(x = "status", y = "age", hue = "status", data = hm)
         plt.title("Violin plot for status and age")
         plt.show()
```

violin plot shows the distribution of data and this violin plot between age and status shows that the maximum survival rate can be seen in between age group 50-60 and max number of people died in age between 40-50.

```
In [62]: sns.violinplot(x = "status", y = "year", hue = "status", data = hm)
         plt.title("Violin plot for status and year")
         plt.show()
```

here the violin plot is drawn between year and status we cna see the highest number of people survided betwwwn the years 59 to 66 and maximum number of people dies between the years 63 to 66 as we cna see the curve.

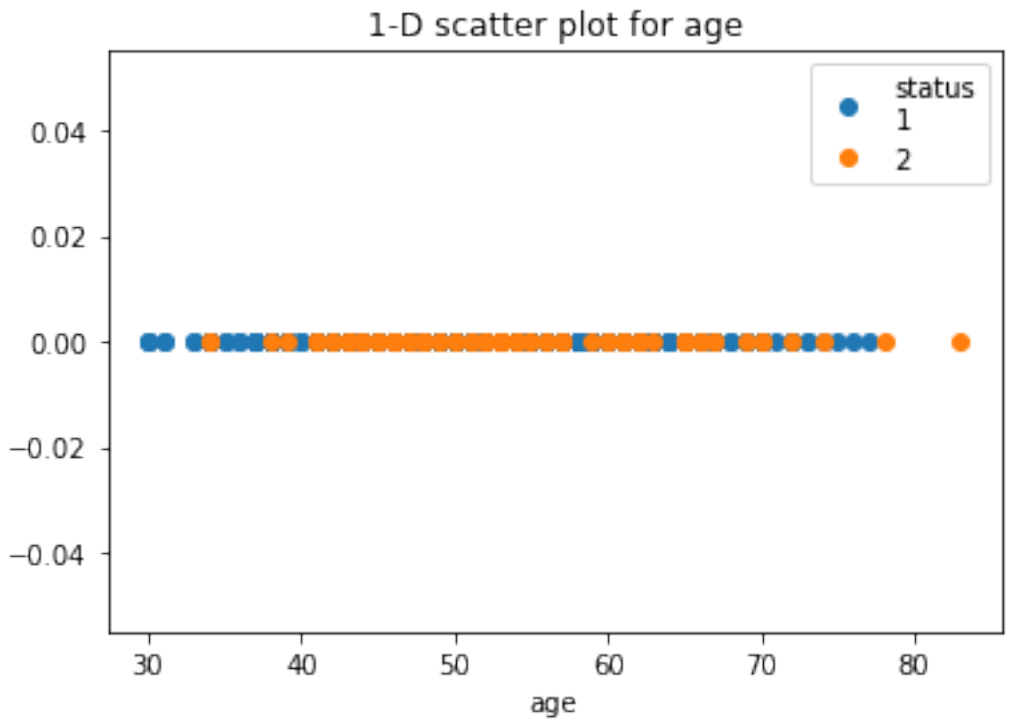
```
In [64]: sns.violinplot(x = "status", y = "nodes", hue = "status", data = hm)
         plt.title("Violin plot for status and node")
         plt.show()
```



this violin curve is drawn between lymph nodes and status and people who are having o-5 lymp nodes are survived and people with more than 5 nodes are not survived

6 Scatter plots

```
In [65]: one = hm.loc[hm["status"] == 1]
         two = hm.loc[hm["status"] == 2]
         plt.plot(one["age"], np.zeros_like(one["age"]), 'o', label = "status\n" "1")
         plt.plot(two["age"], np.zeros_like(two["age"]), 'o', label = "2")
         plt.title("1-D scatter plot for age")
         plt.xlabel("age")
         plt.legend()
         plt.show()
```

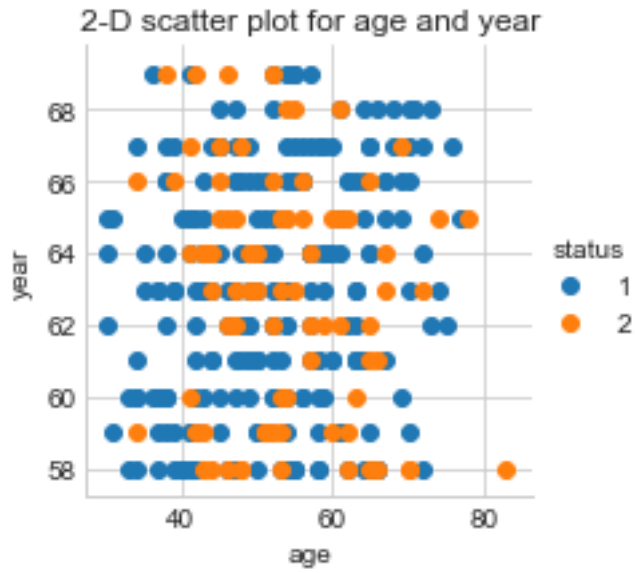


By observing the one dimensional scatter plot we can easily count the number of points that are in the age range survived or not and people who are under age group 41-70 are not survived

```
In [72]: hm.plot(kind = "scatter", x = "age", y = "year")
         plt.title("2-D scatter plot of age")
         plt.show()
```

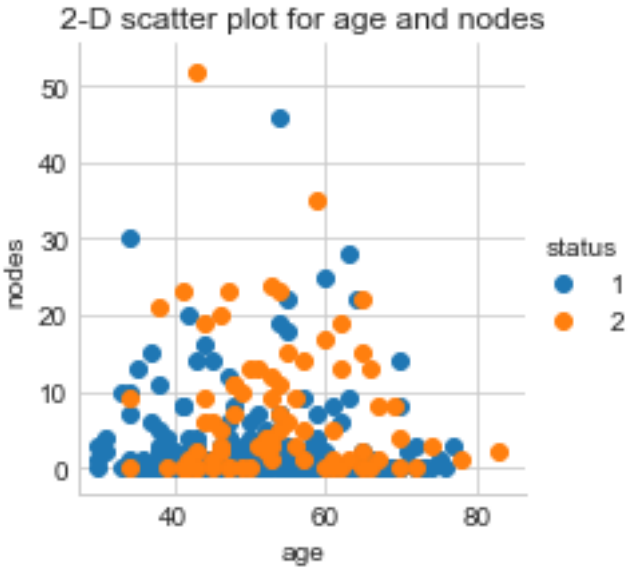
this is a normal two dimensional sctter plot between age and year where all the points are marked in the same color.

```
In [70]: sns.set_style("whitegrid")
         sns.FacetGrid(hm, hue = "status", size = 3).map(plt.scatter, "age", "year").add_legend
         plt.title("2-D scatter plot for age and year")
         plt.show()
```



this is a 2 dimensional scatter plot of age and year and we have used coor coding for each class.

```
In [69]: sns.set_style("whitegrid")
         sns.FacetGrid(hm, hue = "status", size = 3).map(plt.scatter, "age", "nodes").add_leger
         plt.title("2-D scatter plot for age and nodes")
         plt.show()
```



In the above 2d-scatter plots we can see that the class label is not seperable

```
In [73]: sns.set_style("whitegrid")
         sns.pairplot(hm, hue = "status", vars = ["age", "year", "nodes"], size = 3)
         plt.suptitle("pair plot of age, year and nodes")
         plt.show()
```

the data set clearly says that the people survived for longer than 5 years are given the survival status as 1 which is given in blue color and the people who did not survive for more than 5 years are given the survival status of 2 which is in orange color

so, by the data set we can understand that the patients vary from 30 to 83 with the median value of 52 and the maximum number of chances to survive is nearly 75% as the patients are having less nodes and 25% of the patients are having more so, this is an imbalanced data set.

the plot between year and nodes can be a better one for speration of the two classes than the other plots.