Haberman

February 13, 2019

```
In [54]:  #(1.1) DESCRIBE THE BASIC TERMINOLOGY
```

```
In [40]: import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns

         haber =pd.read_csv("haberman.csv")

         print("shape of the data")
         print(haberman.shape) #shape number of row and column
```

```
shape of the data
(305, 4)
```

```
In [41]: haber.head() #print first top value
```

```
Out[41]:    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
```

```
In [9]: print(haberman.columns) #what are the number of columns in data set
```

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

```
In [42]: #haberman data is not label. so how to label the data?
         columns=['patients_age','year_of_operation','no_of_axillary_nodes','classes']

         haberman=pd.read_csv("haberman.csv", names=columns)

         haberman.head()
```

```
Out[42]:    patients_age  year_of_operation  no_of_axillary_nodes  classes
         0            30                 64                     1        1
```

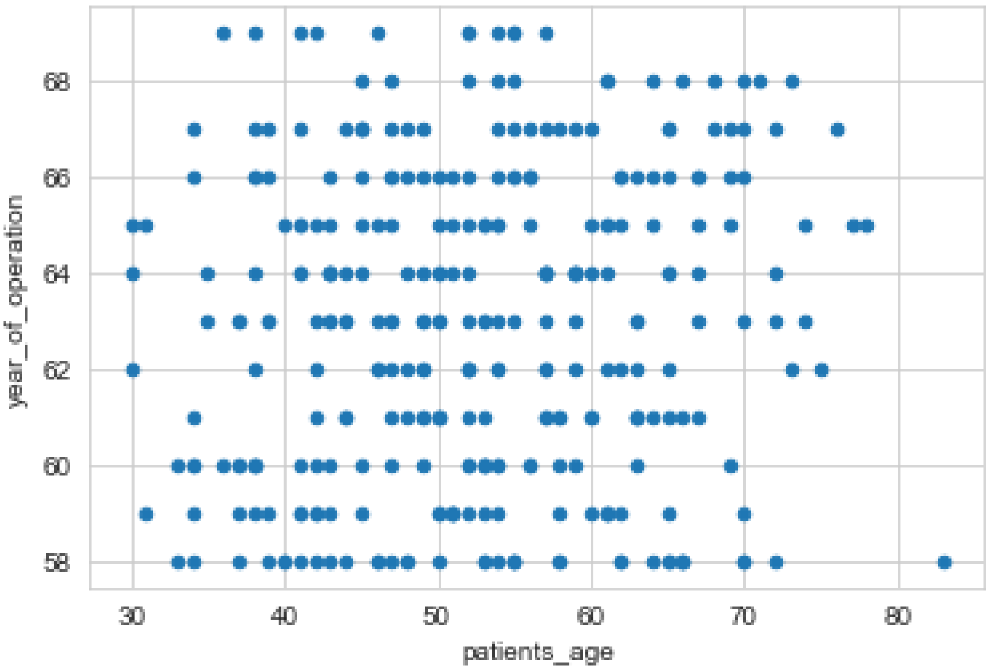
```
1             30                  62                     3         1
2             30                  65                     0         1
3             31                  59                     2         1
4             31                  65                     4         1
```

```
In [20]: haberman.describe()
```

```
Out[20]:        patients_age  year_of_operation  no_of_axillary_nodes     classes
         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 [25]: # 2-D SCATTER PLOT (BIVARIATE)
```

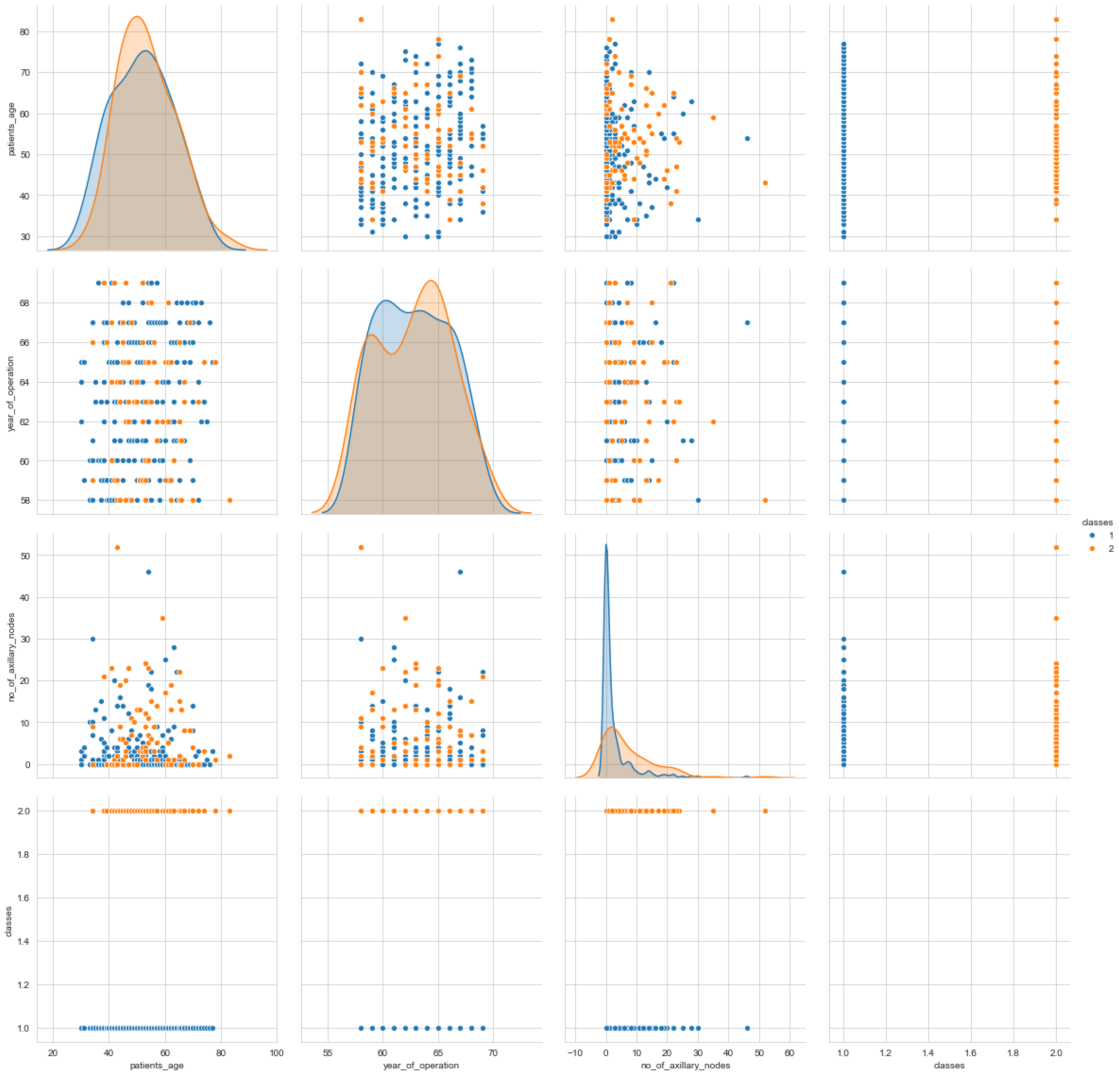
```
In [48]: haberman.plot(kind='scatter',x='patients_age',y='year_of_operation');
         plt.show()
```



```
In [36]: #(1.3) Pair plot
         # In which we can virulize the data in 2-D
```

```
In [51]: plt.close()
         sns.set_style("whitegrid");
         sns.pairplot(haberman,hue="classes",size=4);
         plt.show()
```

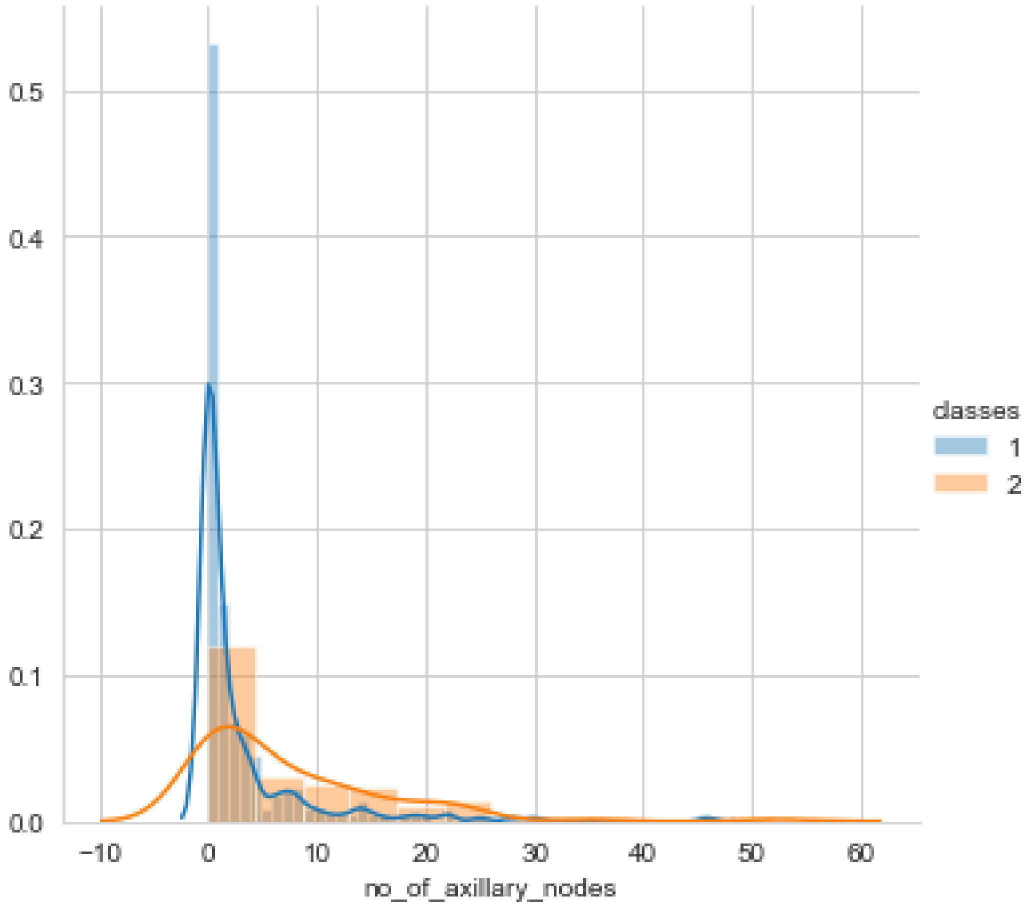
```
/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.py:2065: UserWarning: The `size` parame
  warnings.warn(msg, UserWarning)
/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tu
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
/anaconda3/lib/python3.7/site-packages/statsmodels/nonparametric/kde.py:488: RuntimeWarning: in
  binned = fast_linbin(X, a, b, gridsize) / (delta * nobs)
/anaconda3/lib/python3.7/site-packages/statsmodels/nonparametric/kdetools.py:34: RuntimeWarning
  FAC1 = 2*(np.pi*bw/RANGE)**2
/anaconda3/lib/python3.7/site-packages/numpy/core/fromnumeric.py:83: RuntimeWarning: invalid va
  return ufunc.reduce(obj, axis, dtype, out, **passkwargs)
```



```
In [55]: #(1.4) PDF and CDF
```

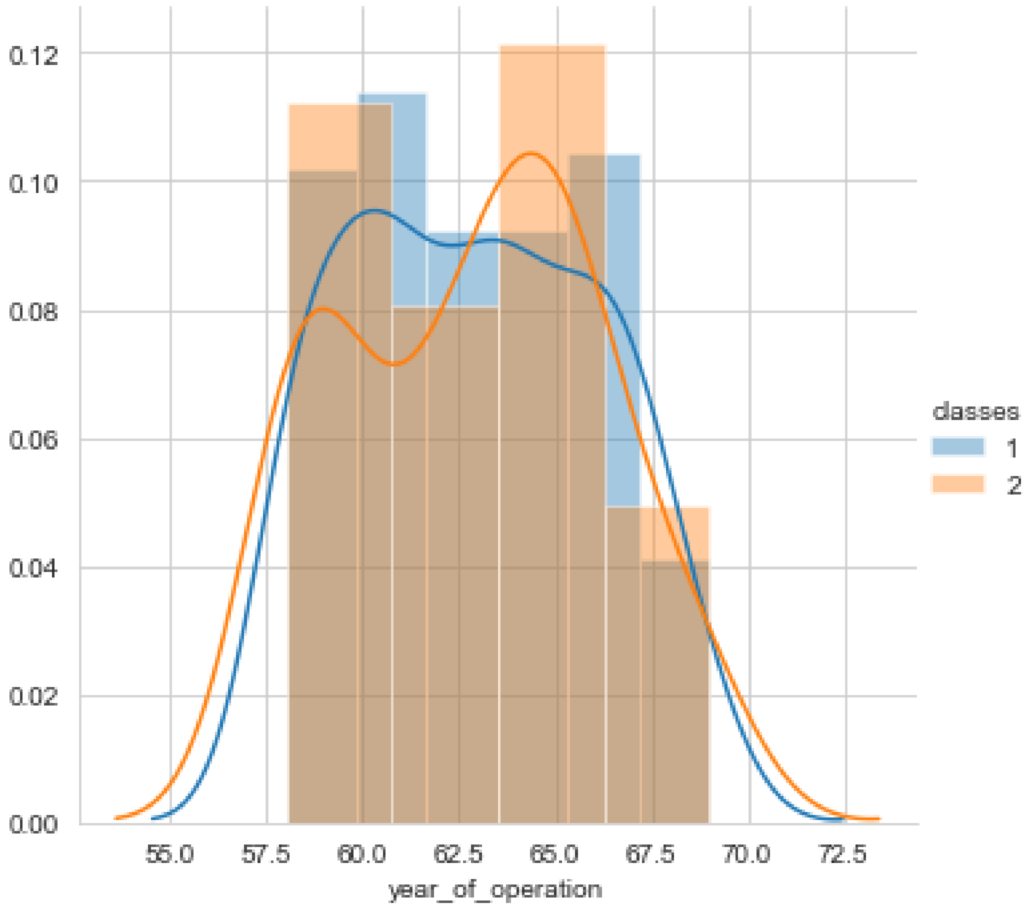
```
In [60]: sns.FacetGrid(haberman,hue="classes",size=5)\
             .map(sns.distplot,"no_of_axillary_nodes")\
             .add_legend()
         plt.show();
```

```
/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.py:230: UserWarning: The `size` paramte
  warnings.warn(msg, UserWarning)
/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tu
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



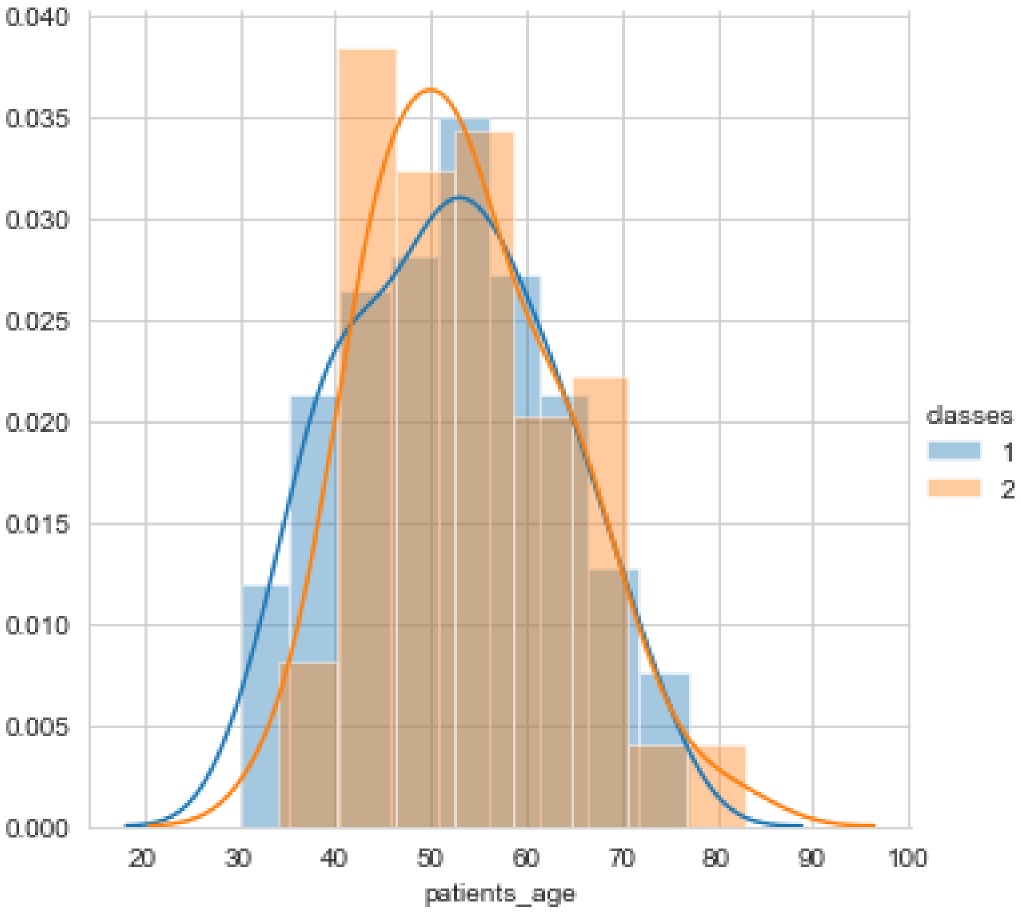
```
In [63]: sns.FacetGrid(haberman,hue="classes",size=5)\
                      .map(sns.distplot,"year_of_operation")\
                      .add_legend()
         plt.show()
```

```
/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.py:230: UserWarning: The `size` paramt
  warnings.warn(msg, UserWarning)
/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tu
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



```
In [66]: sns.FacetGrid(haberman,hue="classes",size=5)\
                       .map(sns.distplot,"patients_age")\
                       .add_legend()
         plt.show()
```

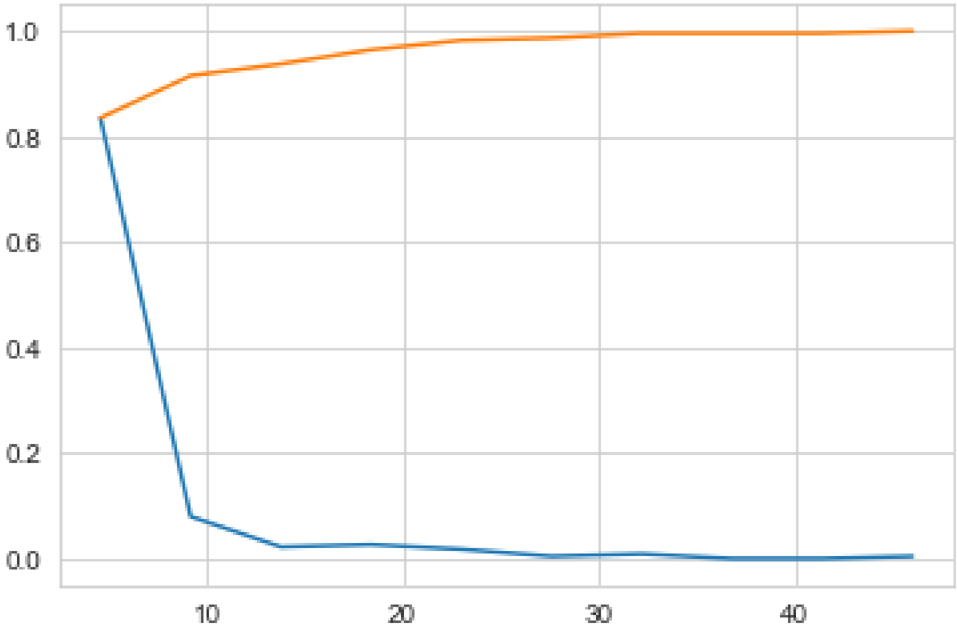
```
/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.py:230: UserWarning: The `size` paramt
  warnings.warn(msg, UserWarning)
/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tu
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



```
In [72]: five_more=haberman.loc[haberman["classes"]==1]
         five_less=haberman.loc[haberman["classes"]==2]

         counts,bin_edges=np.histogram(five_more['no_of_axillary_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.show();
```

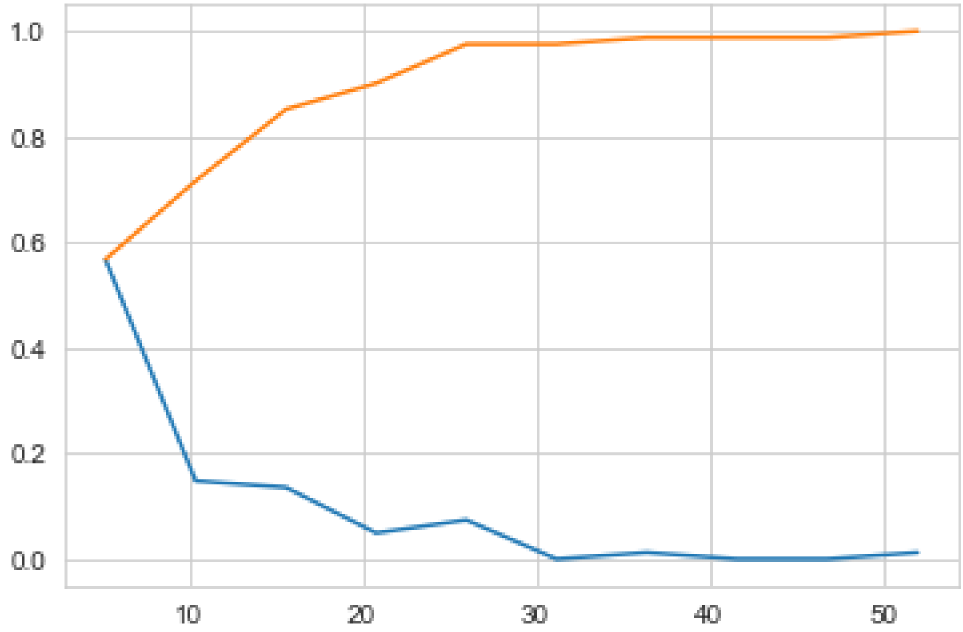
```
[0.83555556 0.08       0.02222222 0.02666667 0.01777778 0.00444444
 0.00888889 0.         0.         0.00444444]
```

```
[ 0.   4.6  9.2 13.8 18.4 23.  27.6 32.2 36.8 41.4 46. ]
```



```
In [74]: counts,bin_edges=np.histogram(five_less[],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.show()
```

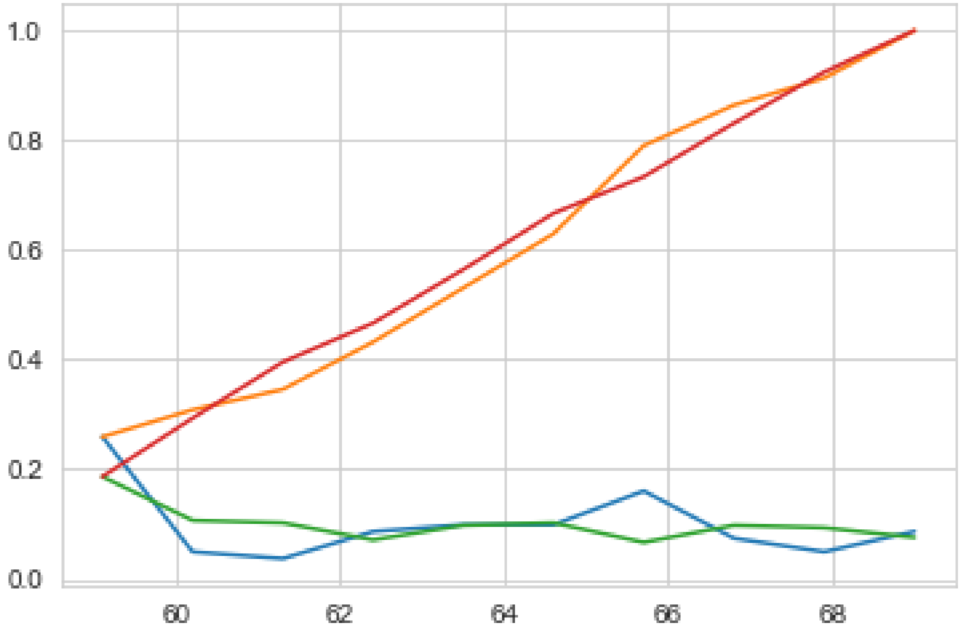
```
[0.56790123 0.14814815 0.13580247 0.04938272 0.07407407 0.
 0.01234568 0.         0.         0.01234568]
[ 0.   5.2 10.4 15.6 20.8 26.  31.2 36.4 41.6 46.8 52. ]
```



```
In [75]: #(1.5) BOX PLOT

In [80]: #classes 2
         counts,bin_edges=np.histogram(five_less['year_of_operation'],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);


         #classes 1
         counts,bin_edges=np.histogram(five_more['year_of_operation'],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.show()
```

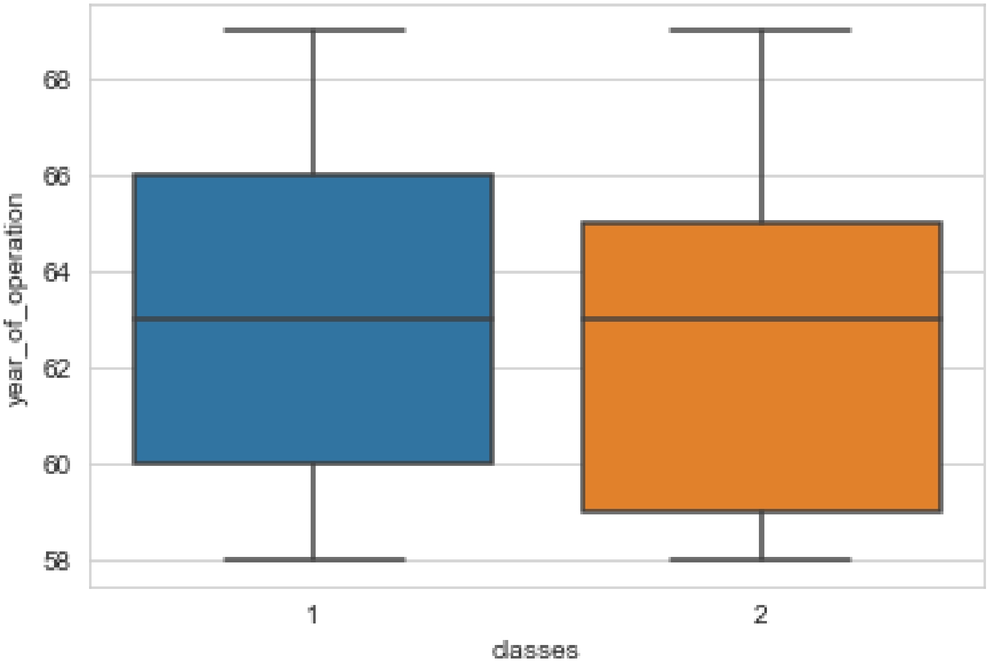
```
[0.25925926 0.04938272 0.03703704 0.08641975 0.09876543 0.09876543
 0.16049383 0.07407407 0.04938272 0.08641975]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
[0.18666667 0.10666667 0.10222222 0.07111111 0.09777778 0.10222222
 0.06666667 0.09777778 0.09333333 0.07555556]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
```



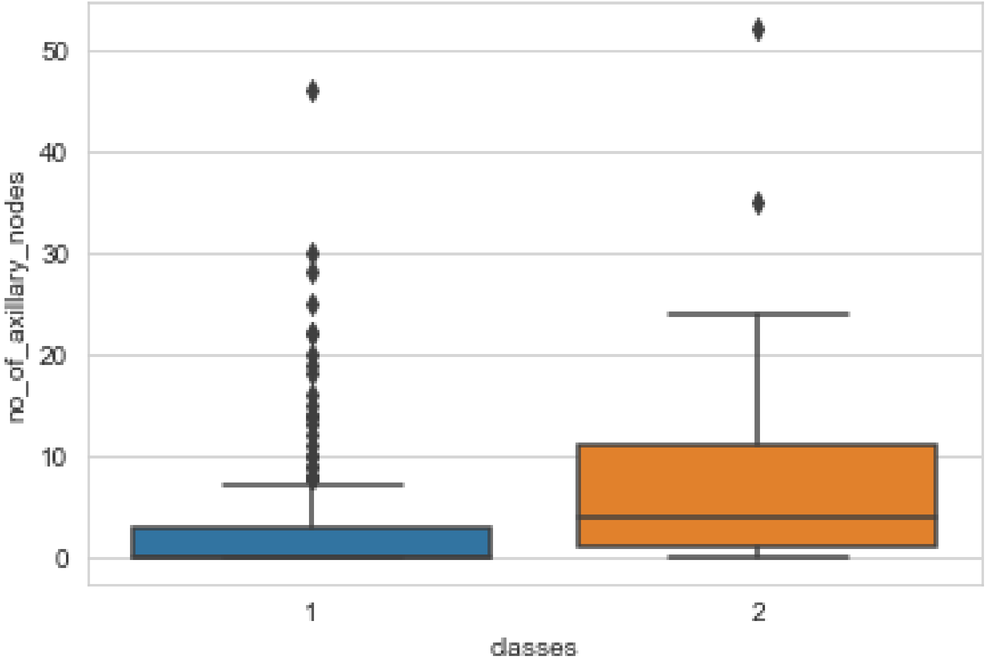
```
In [81]: #(1.6) BOX PLOT AND WHISKERS
```

```
In [83]: sns.boxplot(x='classes',y='year_of_operation',data=haberman)
         plt.show
```

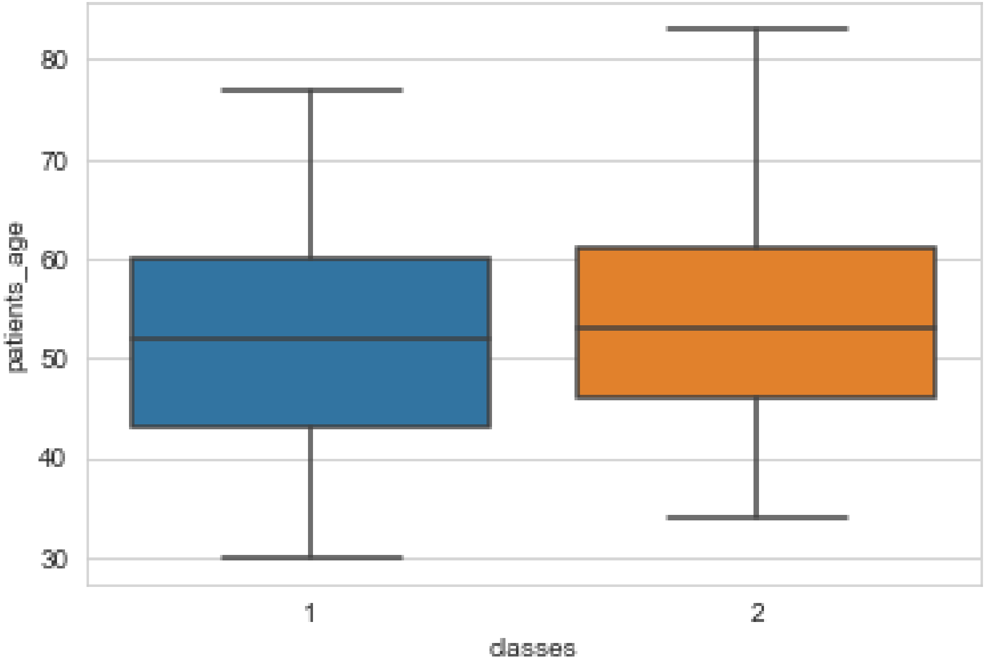
```
Out[83]: <function matplotlib.pyplot.show(*args, **kw)>
```



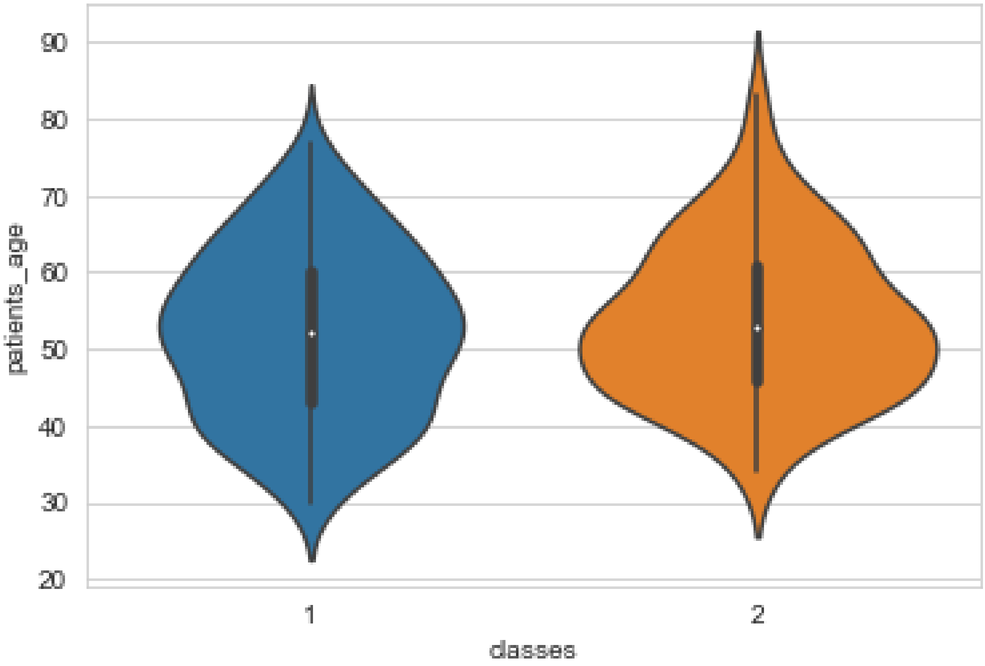
```
In [85]: sns.boxplot(x='classes',y='no_of_axillary_nodes',data=haberman)
         plt.show

Out[85]: <function matplotlib.pyplot.show(*args, **kw)>
```



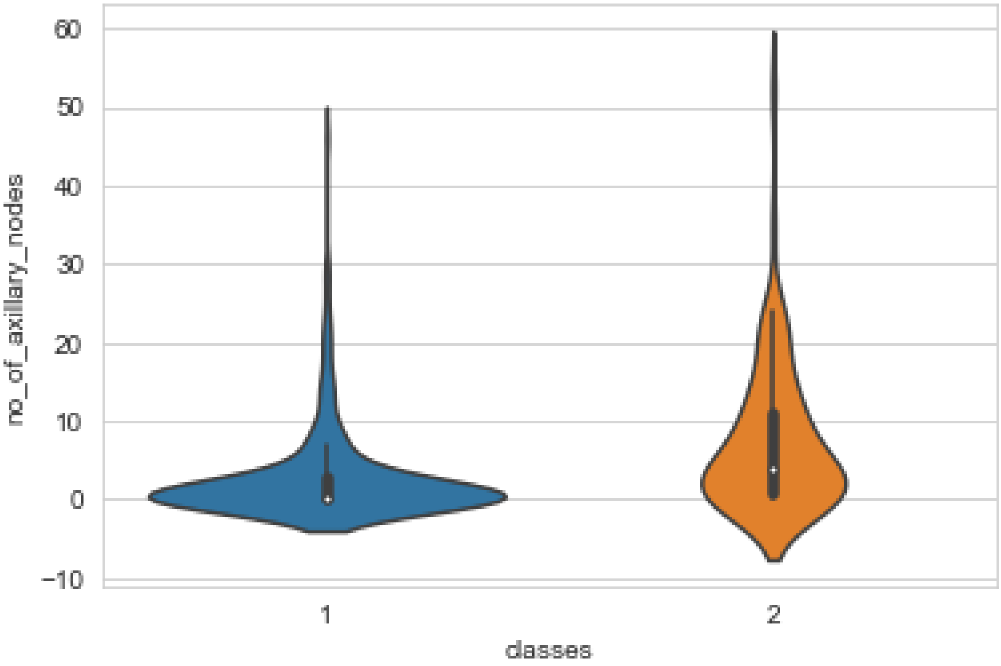
```
In [86]: sns.boxplot(x='classes',y='patients_age',data=haberman)
         plt.show

Out[86]: <function matplotlib.pyplot.show(*args, **kw)>
```



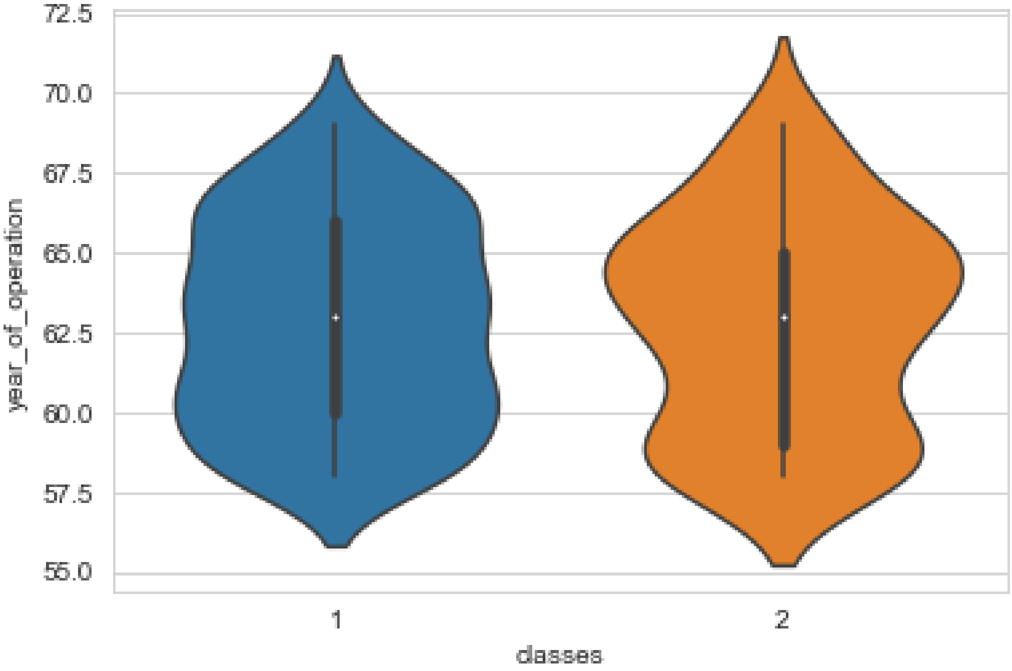
```
In [87]: #(1.7) Violin plot

In [90]: sns.violinplot(x="classes", y="patients_age", data=haberman)
         plt.show

/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tu
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval


Out[90]: <function matplotlib.pyplot.show(*args, **kw)>
```



```
In [93]: sns.violinplot(x='classes', y='no_of_axillary_nodes', data=haberman)
         plt.show

/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tu
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval


Out[93]: <function matplotlib.pyplot.show(*args, **kw)>
```



```
In [92]: sns.violinplot(x="classes", y="year_of_operation", data=haberman)
         plt.show

/anaconda3/lib/python3.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tu
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval


Out[92]: <function matplotlib.pyplot.show(*args, **kw)>
```


72.5
70.0
67.5
65.0
62.5
60.0
57.5
55.0
year_of_operation
1
2
classes