Ass1-Perform EDA on Haberman dataset

February 12, 2019

Q)This is the first assignment on data visualization.

1. The data and reference notebook is attached here, try to document every plot and analysis that you do.
2. Experiment with different functionalities of jupyter notebook and get habituated with its features.
3. Try out as many plotting techniques as you can, but write down your observations for each of them.
4. Please be sure to have proper axes names, title and legend to all the charts that you plot.
5. Have a proper conclusions section where in you summarise your overall observation.
6. If you want to explore more about Haberman's Survival Data Set, you can try out this link https://www.kaggle.com/gilsousa/habermans-survival-data-set/version/1

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

Data Description: The Haberman's survival dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer. source :https://www.kaggle.com/

```
In [2]: # Loading data using pandas and doing a quick look
        haberman_csv="haberman.csv"
        haberman=pd.read_csv(haberman_csv)
        haberman.head()
```

```
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
```

```
In [3]: # Getting general information about the dataset;
        print(haberman.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306 entries, 0 to 305
Data columns (total 4 columns):
age       306 non-null int64
year      306 non-null int64
nodes     306 non-null int64
status    306 non-null int64
dtypes: int64(4)
memory usage: 9.6 KB
None
```

```
In [4]: # Getting the datatype of column in df
        haberman.dtypes
```

```
Out[4]: age       int64
        year      int64
        nodes     int64
        status    int64
        dtype: object
```

0.1 Attribute Information:

1. Age of patient at time of operation (numerical)
2. Patient's year of operation (year - 1900, numerical)
3. Number of positive axillary nodes detected (numerical)
4. Survival status (class attribute)

    - 1 = the patient survived 5 years or longer
    - 2 = the patient died within 5 year

Source : https://archive.ics.uci.edu/ml/datasets/Haberman's+Survival

```
In [5]: # Q) How many Data points and feature ?
        haberman.shape
```

```
Out[5]: (306, 4)
```

0.1.1 Observation:

```
- There are  306  data points and 4 columns .
- 3 input and 1 output
```

```
In [6]: # Getting the column names of our dataset
        haberman.columns
```

```
Out[6]: Index(['age', 'year', 'nodes', 'status'], dtype='object')
```

```
In [7]: # q) How many data points for each classes are present
        haberman["status"].value_counts()
```

```
Out[7]: 1    225
        2     81
        Name: status, dtype: int64
```

0.1.2 Observation:

1. There are 2 classes .

   - 1 : 225 points i.e 225 patients survived 5 years or longer
   - 2 : 81 points i.e 81 patients died within 5 year

2. This is imbalance dataset as one class has 225 points and other one has 81 points

```
In [ ]:
```

```
In [8]: # Getting basic description about our dataset
        haberman.describe()
```

```
Out[8]:                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 [9]: # But first lets us rename our columns for better readability
        haberman.columns=['Age', 'Year_of_treatment','Positive_Lymph_Nodes_counts',
                          'Survival_After_5_Years']
```

```
In [10]: haberman.head()
```

```
Out[10]:    Age  Year_of_treatment  Positive_Lymph_Nodes_counts  Survival_After_5_Years
         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 [11]: # Renaming the classed for better readability
         haberman['Survival_After_5_Years']=haberman['Survival_After_5_Years']\
         .map({1:'Survived',2:'Died'})
         haberman.head()
```

```
Out[11]:    Age  Year_of_treatment  Positive_Lymph_Nodes_counts Survival_After_5_Years
         0   30                 64                            1               Survived
         1   30                 62                            3               Survived
         2   30                 65                            0               Survived
         3   31                 59                            2               Survived
         4   31                 65                            4               Survived
```

1 Objective :

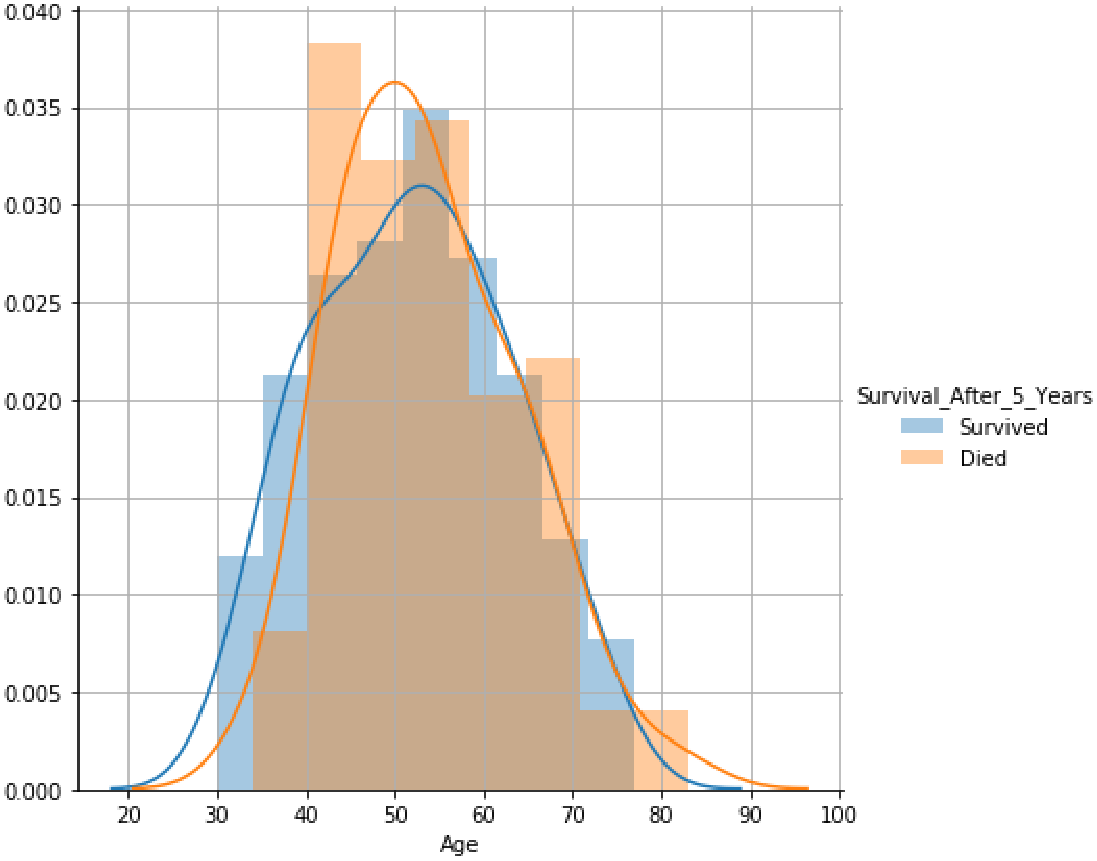
As far as I can think I am going to predict whether a patient will survive for more than 5 1years or not based on certain features/factors

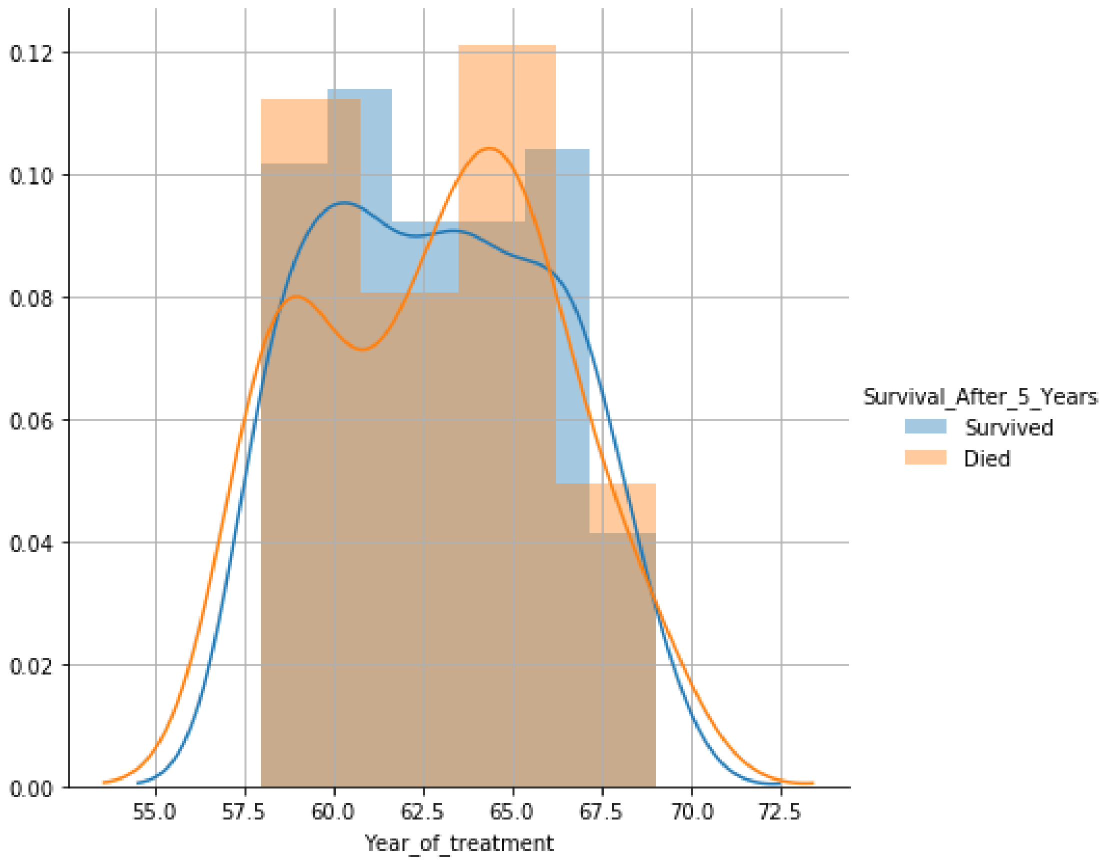
2 Plots

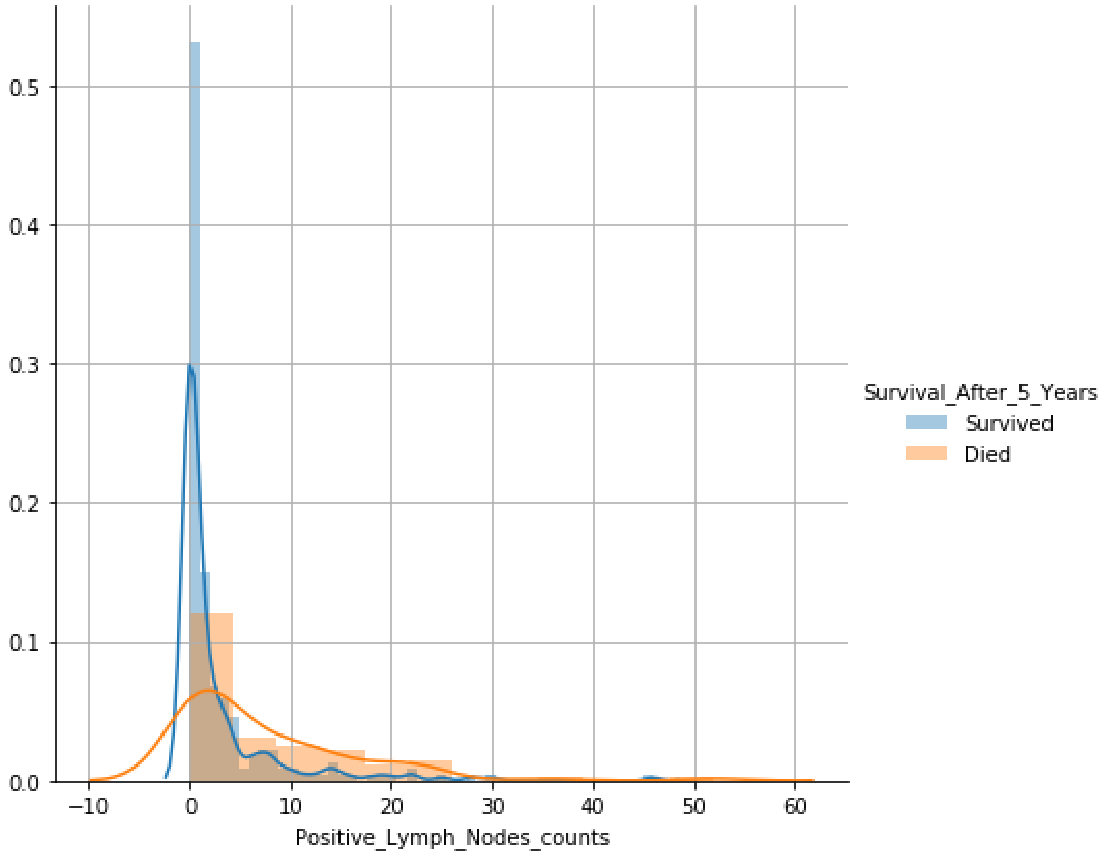
2.1 Univariate Analysis

```
In [12]: # Trying to identify useful feature for usinf pdf and cdf for classification
         patients_survived=haberman[haberman['Survival_After_5_Years']=='Survived']
         patients_died=haberman[haberman['Survival_After_5_Years']=='Died']
```

```
In [13]: # Histogram and PDF
         # I am going to loop over each feature and plot a distplot except the class
         for feature in haberman.columns[:-1]:

             sns.FacetGrid(haberman,hue='Survival_After_5_Years',height=6)\
             .map(sns.distplot,feature).add_legend()
             plt.grid()
             plt.show()
```




0.12
0.10
0.08
0.06
0.04
0.02
0.00
55.0
57.5
60.0
62.5
65.0
67.5
70.0
72.5
Year_of_treatment
Survival_After_5_Years
Survived
Died

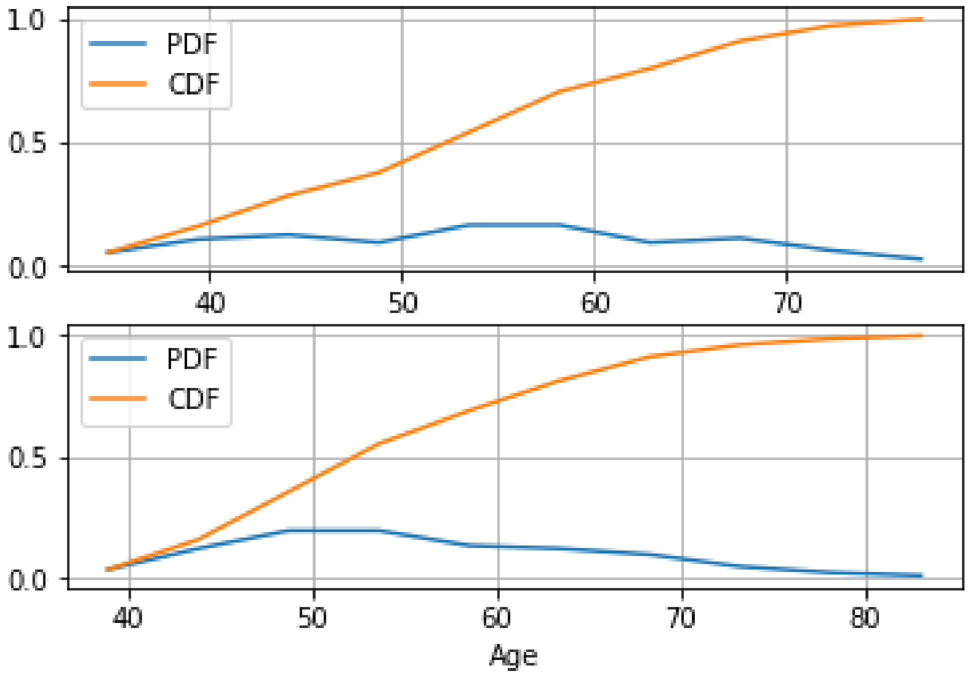


2.1.1 Observations:

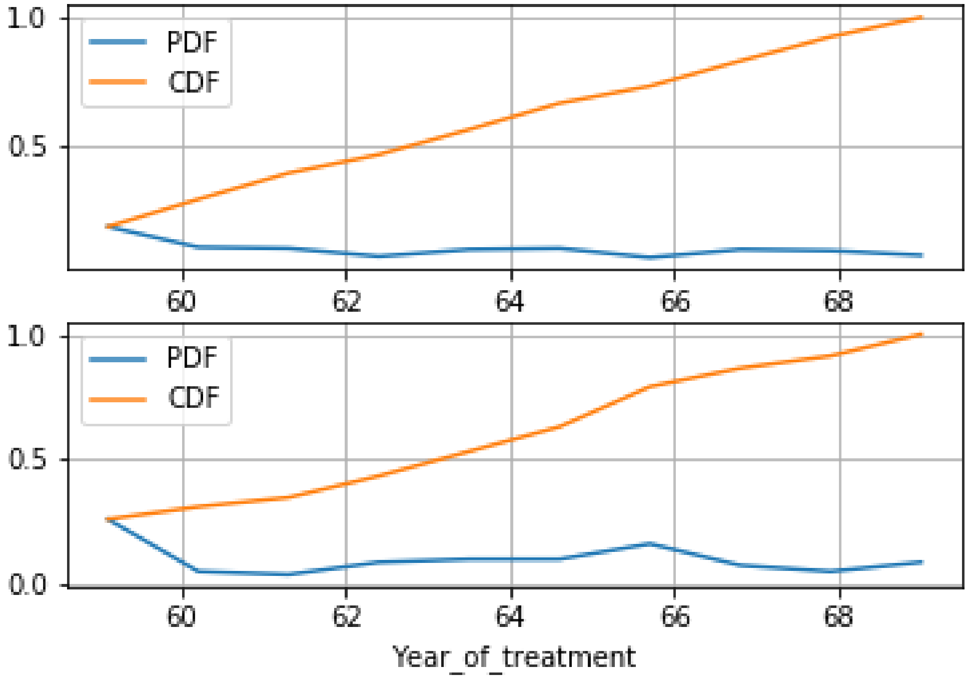
- Huge overlap in Year_of_treament and age columns , so it is very tough to generalise anything
- No of positive nodes is a deciding factor ,. It show that a patient with lymph nodes less than 3(approximately) has higher survival rate

## CDF and PDF

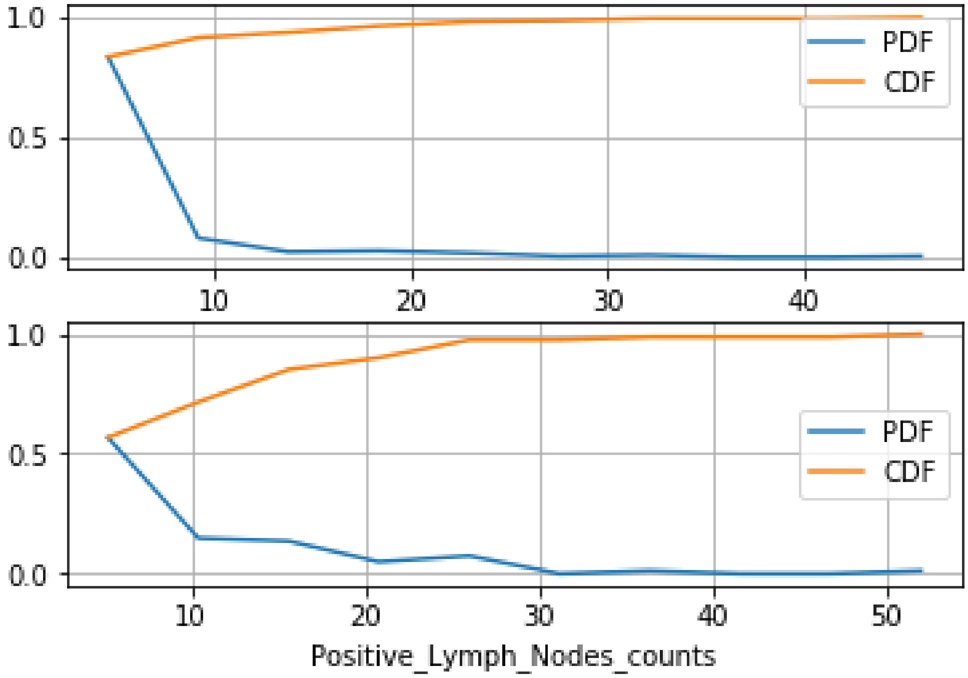
```
In [14]: # using subplots
         # sns.set_style("whitegrid")
         for feature in haberman.columns[:-1]:
             plt.figure(1)
             # Survived

             counts,bin_edges=np.histogram(patients_survived[feature],bins=10,density=True)
             pdf=counts/(sum(counts))
             cdf=np.cumsum(pdf)
             plt.subplot(211)
             plt.grid()
             plt.xlabel(feature)
             plt.plot(bin_edges[1:],pdf,label='PDF')
             plt.plot(bin_edges[1:],cdf,label='CDF')
             plt.legend()

             # Dies
             counts,bin_edges=np.histogram(patients_died[feature],bins=10,density=True)
             pdf=counts/sum(counts)
             cdf=np.cumsum(pdf)
             plt.subplot(212)
             plt.grid()
             plt.xlabel(feature)
             plt.plot(bin_edges[1:],pdf,label='PDF')
             plt.plot(bin_edges[1:],cdf,label='CDF')
             plt.legend()

             plt.show()
             print ("*"*20,feature,"*"*20)
```



```
******************** Age ********************
```



```
******************** Year_of_treatment ********************
```



```
******************** Positive_Lymph_Nodes_counts ********************
```

2.1.2 Observation :

Here we can again see that patient with positive_lymph_nodes_count eqaul to or less than 3 (approx) has much higher chances of survival rate for

3 Mean , Variance and Std-dev

```
In [15]: print("Means")
         print("Average no of lymph node for \
          which patients survived ",np.mean(patients_survived['Positive_Lymph_Nodes_counts']))
         print ("**"*10)
         print("Mean with outliers",np.mean(np.append\
                                            (patients_survived['Positive_Lymph_Nodes_counts'],
         print("STD-DEV")
         print(np.std(patients_died['Age']))
```

```
Means
Average no of lymph node for  which patients survived  2.7911111111111113
********************
Mean with outliers 3.2212389380530975
STD-DEV
10.104182193031312
```

4 Median , Percentile , Quantile , IQR , MAD

```
In [16]: print("Median")
         print(np.median(patients_survived['Positive_Lymph_Nodes_counts']))
         print("Median with outliers",np.median(np.append\
                                            (patients_survived['Positive_Lymph_Nodes_counts']\
                                             ,100)))
         print("Quantile")
         print(np.percentile(patients_survived['Age'],np.arange(0,100,25)))
         print("90th Percentile")
         print(np.percentile(patients_died['Age'],90))
         from statsmodels.robust import mad
         print("Median Absolute Deviation")
         print(mad(patients_survived['Positive_Lymph_Nodes_counts']))
```

```
Median
0.0
Median with outliers 0.0
Quantile
[30. 43. 52. 60.]
```

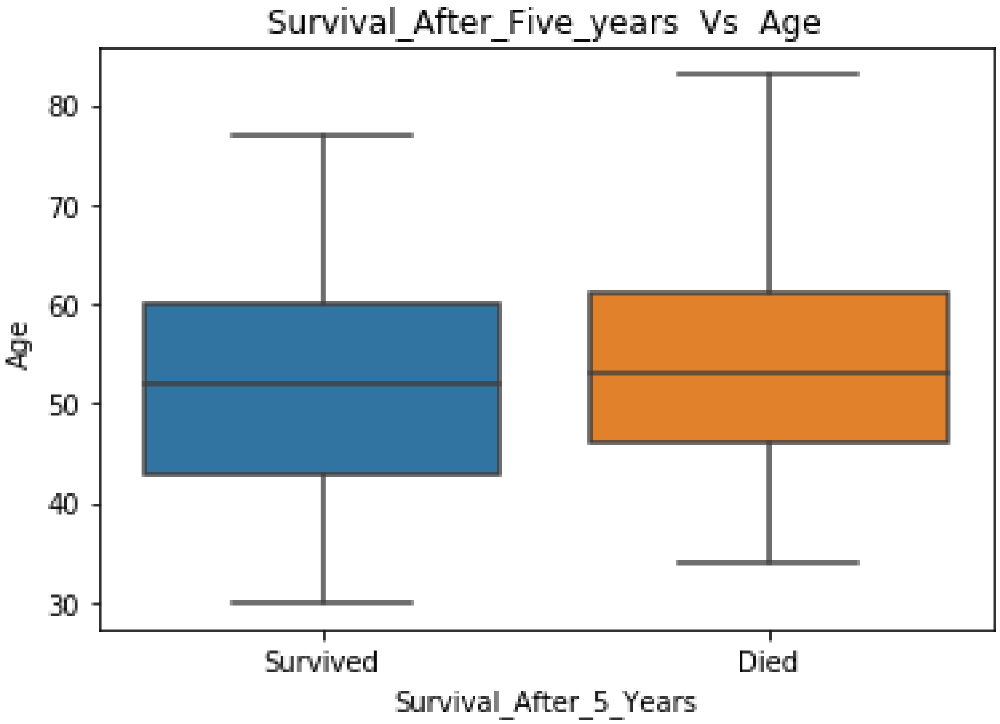
```
90th Percentile
67.0
Median Absolute Deviation
0.0
```

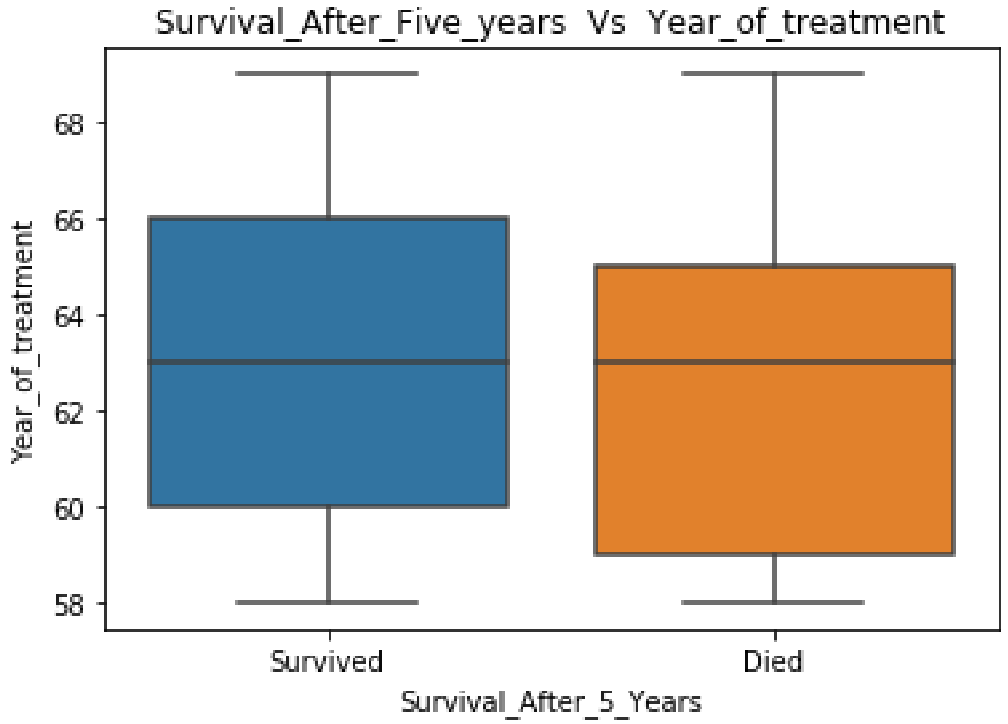
4.0.1 Observation:

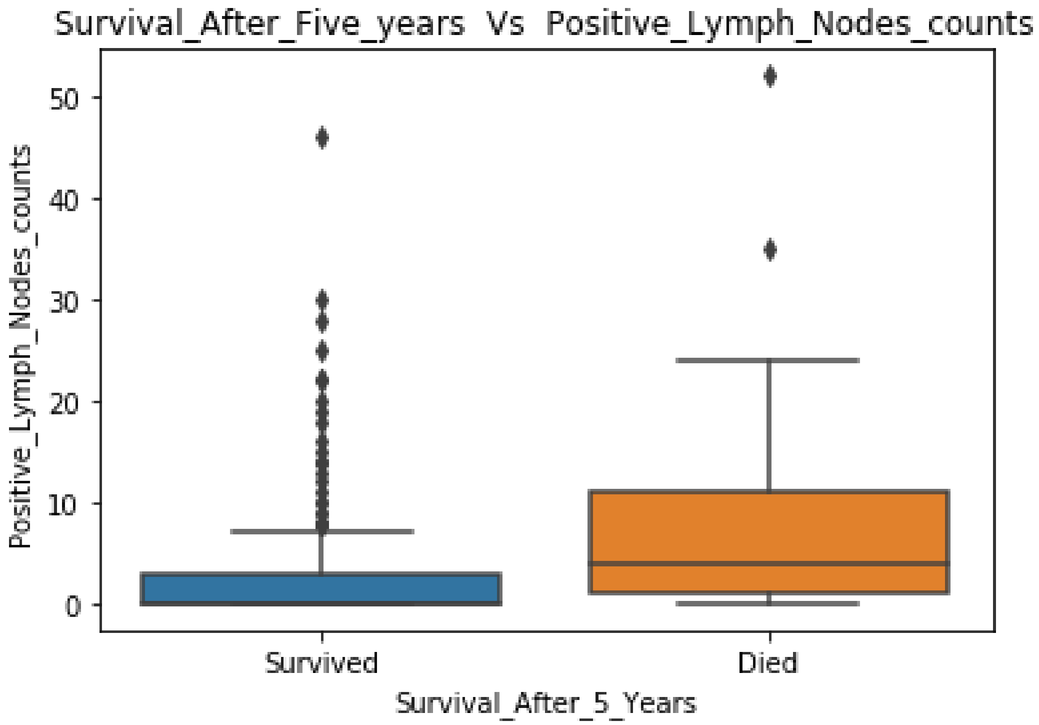
Mean is highly responsive to outliers , but median is not /very less responsive to outliers

4.1 Boxplot

```
In [17]: # plotting boxplot for each other features with the target class
         for feat1 in haberman.columns[:-1]:
                 sns.boxplot(x='Survival_After_5_Years',y=feat1,data=haberman)
                 plt.title("Survival_After_Five_years  Vs  {0}".format(feat1))
                 plt.show()
```




Survival_After_Five_years Vs Year_of_treatment
Year_of_treatment
68
66
64
62
60
58
Survived
Died
Survival_After_5_Years


Survival_After_Five_years Vs Positive_Lymph_Nodes_counts
Positive_Lymph_Nodes_counts
50
40
30
20
10
0
Survived
Died
Survival_After_5_Years

4.2 Observations

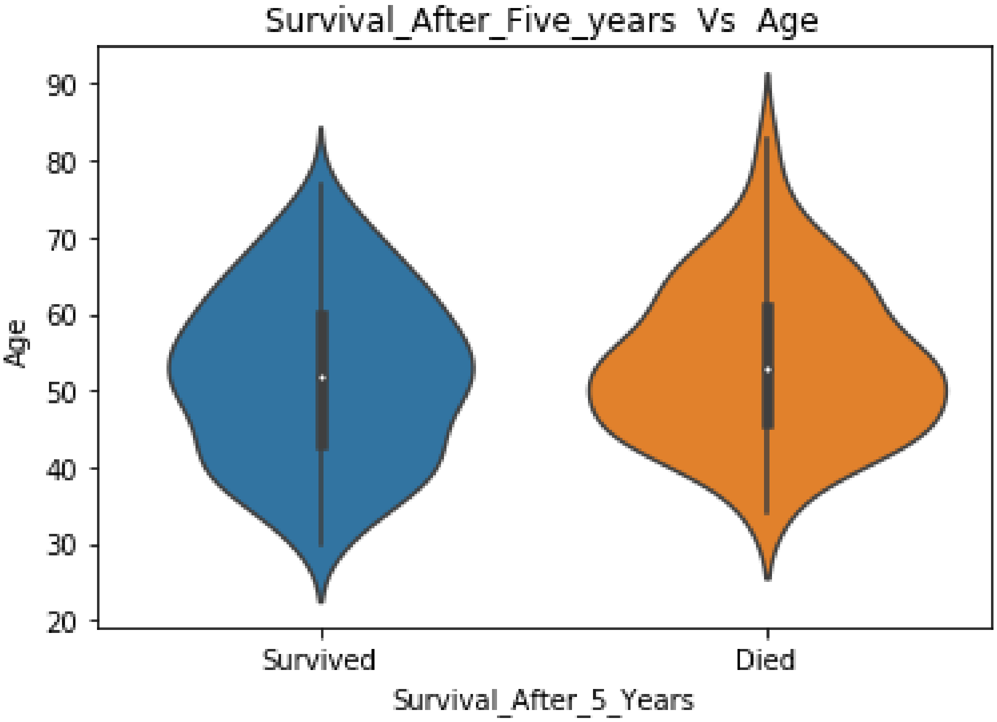
Plot 1: Survival_After_Five_Years Vs Age: - Survived :the patients who survived for 5years or more 50 percentile of them lie between age 42 to 60 years approx. - Died : The patients whose age were between 45 to 62 approx consist of 50 percentile who died before 5 years.

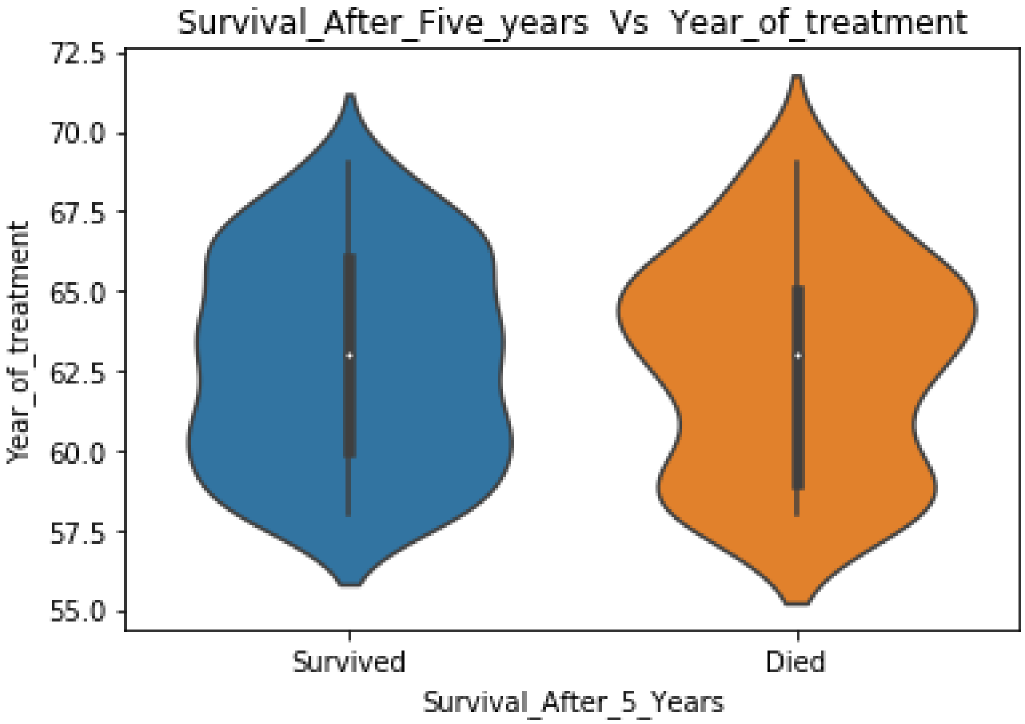
plot 2 : Survival_After_Five_Years Vs Year_of_treatment: - The pateints who were treated in later years have higher chances of survival

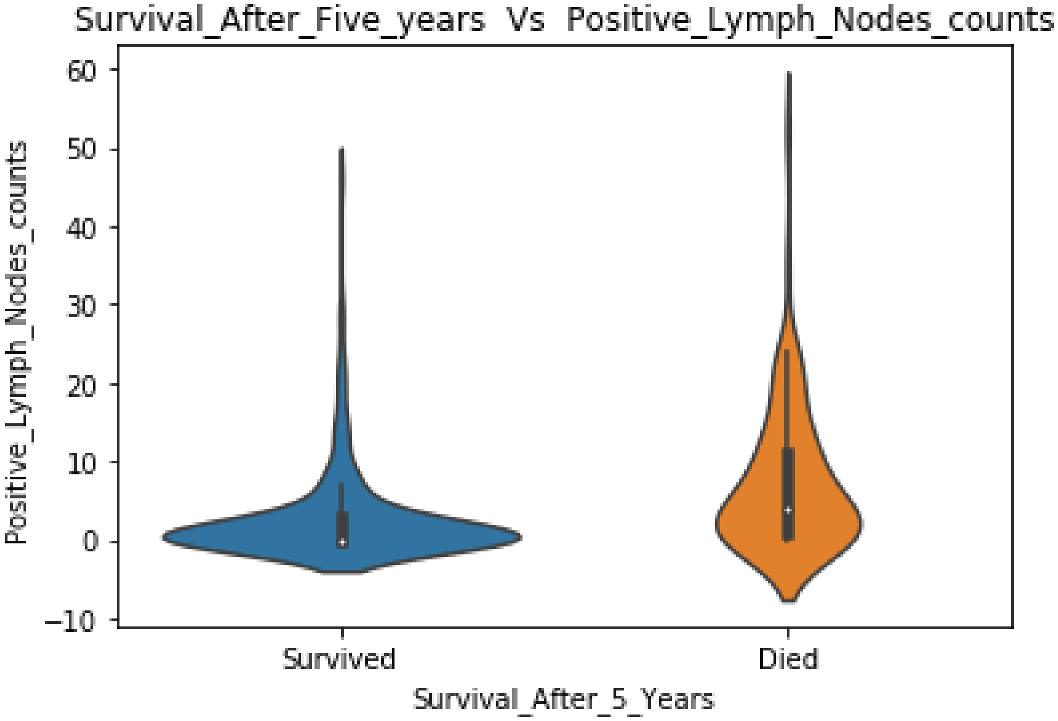
plot 3 : Survival_After_Five_Years Vs Positive_Lymph_Nodes_Counts: - Very high surival rate for No of Positive Lymph Nodes less than 3

4.3 Violin Plot

```
In [18]: # plotting violinplot for each other features with the target class
         for feat1 in haberman.columns[:-1]:
             sns.violinplot(x='Survival_After_5_Years',y=feat1,data=haberman)
             plt.title("Survival_After_Five_years  Vs  {0}".format(feat1))
             plt.show()
```




Survival_After_Five_years Vs Year_of_treatment
Year_of_treatment
72.5
70.0
67.5
65.0
62.5
60.0
57.5
55.0
Survived
Died
Survival_After_5_Years

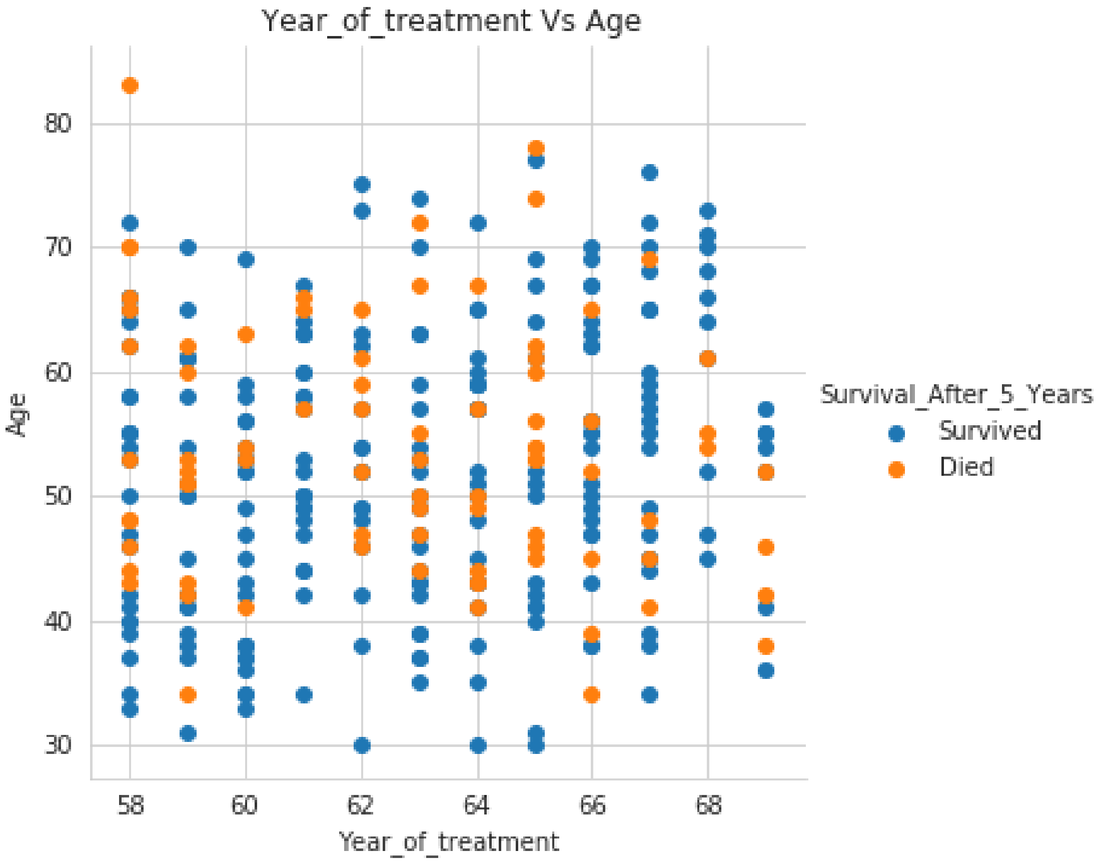

Survival_After_Five_years Vs Positive_Lymph_Nodes_counts
Positive_Lymph_Nodes_counts
60
50
40
30
20
10
0
−10
Survived
Died
Survival_After_5_Years

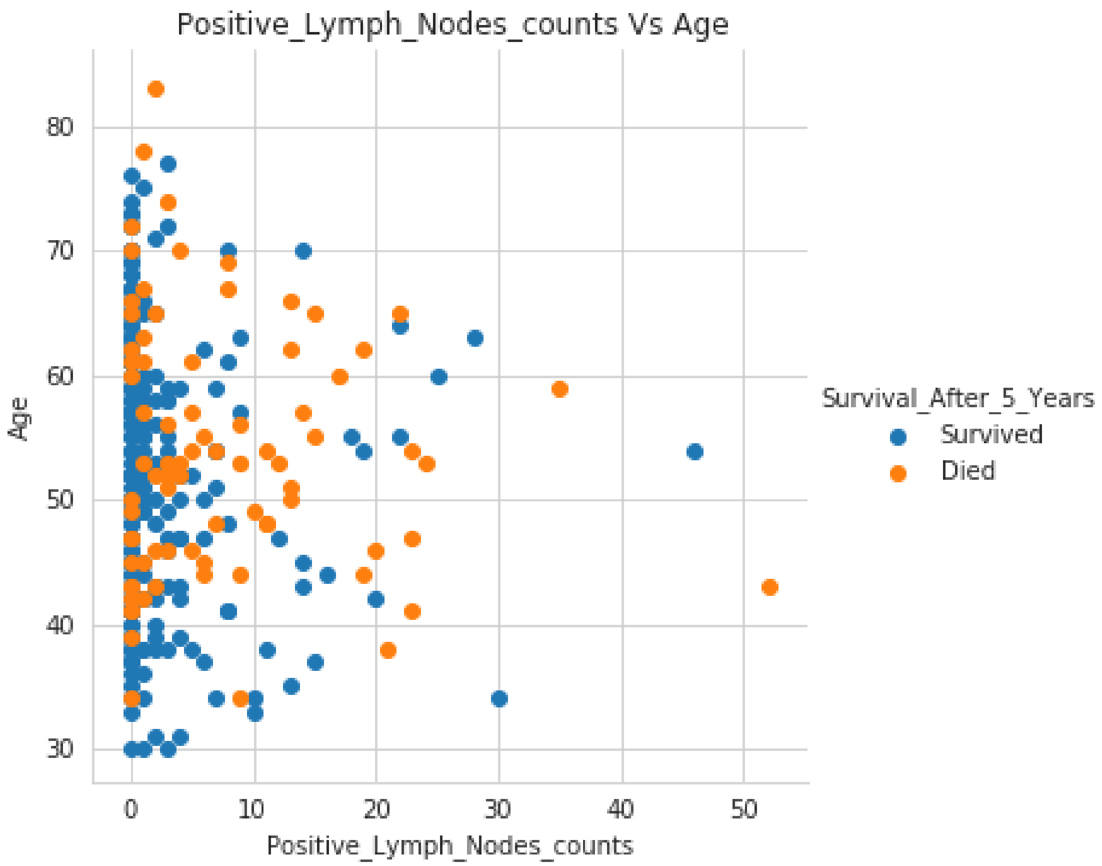
4.4 Observation

The number of Positive Lymph_Node_counts for survival is dense from 0-5.

5 2-D Scatter Plot

```
In [19]: for feat1 in haberman.columns[:1]:
             for feat2 in haberman.columns[:-1]:
                 if feat1!=feat2:
                     sns.set_style("whitegrid")
                     sns.FacetGrid(haberman,hue="Survival_After_5_Years",height=5)\
                     .map(plt.scatter,feat2,feat1).add_legend()

         #               haberman.plot(kind="scatter",x=feat1,y=feat2)
                     plt.title(" {1} Vs {0}".format(feat1,feat2))
                     plt.show()
```





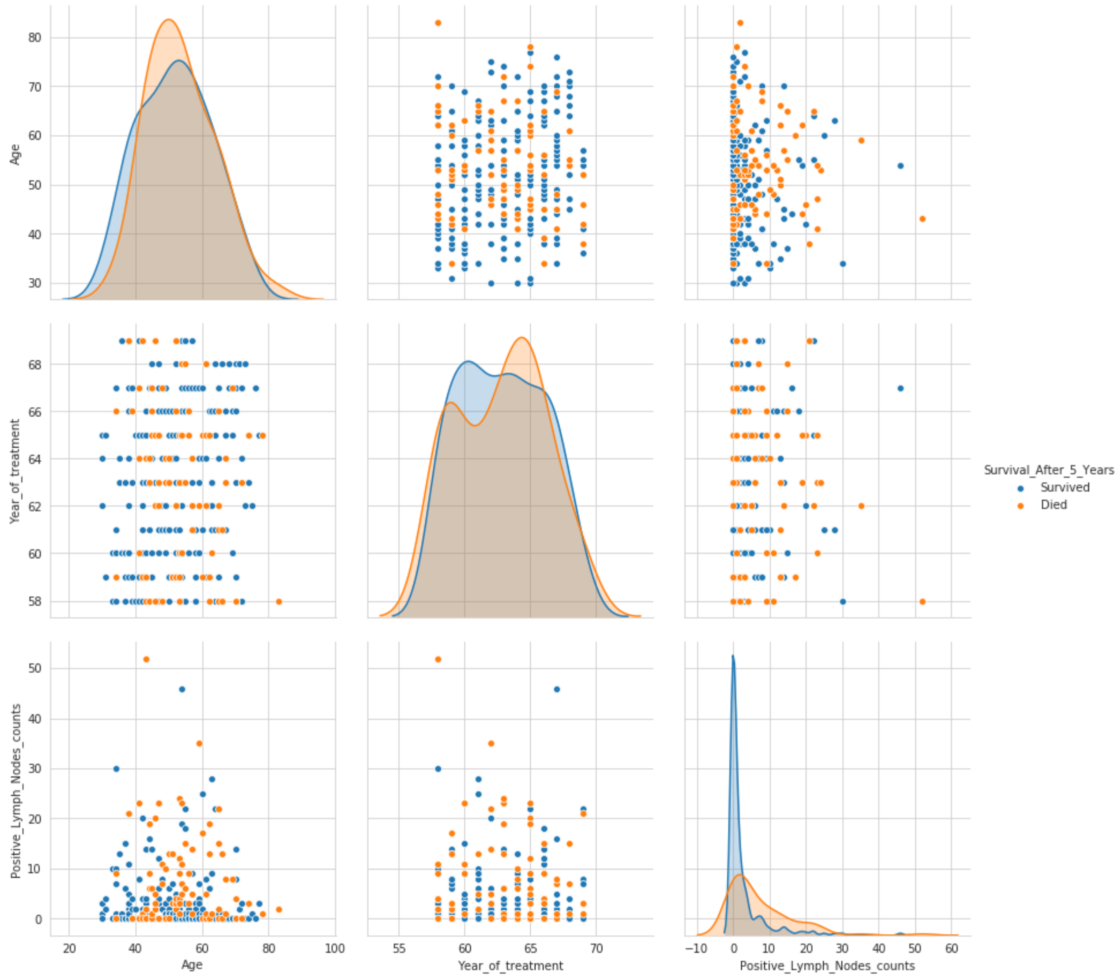
5.1 Observation

For plot1 : - This scatter plot doesn't give much idea , but we can say that majority of operations are performed on people age range between 40 and 68 approx

For Plot 2: - We can see that there is quite good concentration of data point When Lymph is 0

5.2 Pair Plot

```
In [20]: sns.set_style("whitegrid")
         sns.pairplot(haberman,hue="Survival_After_5_Years",height=4)
         plt.show()
```

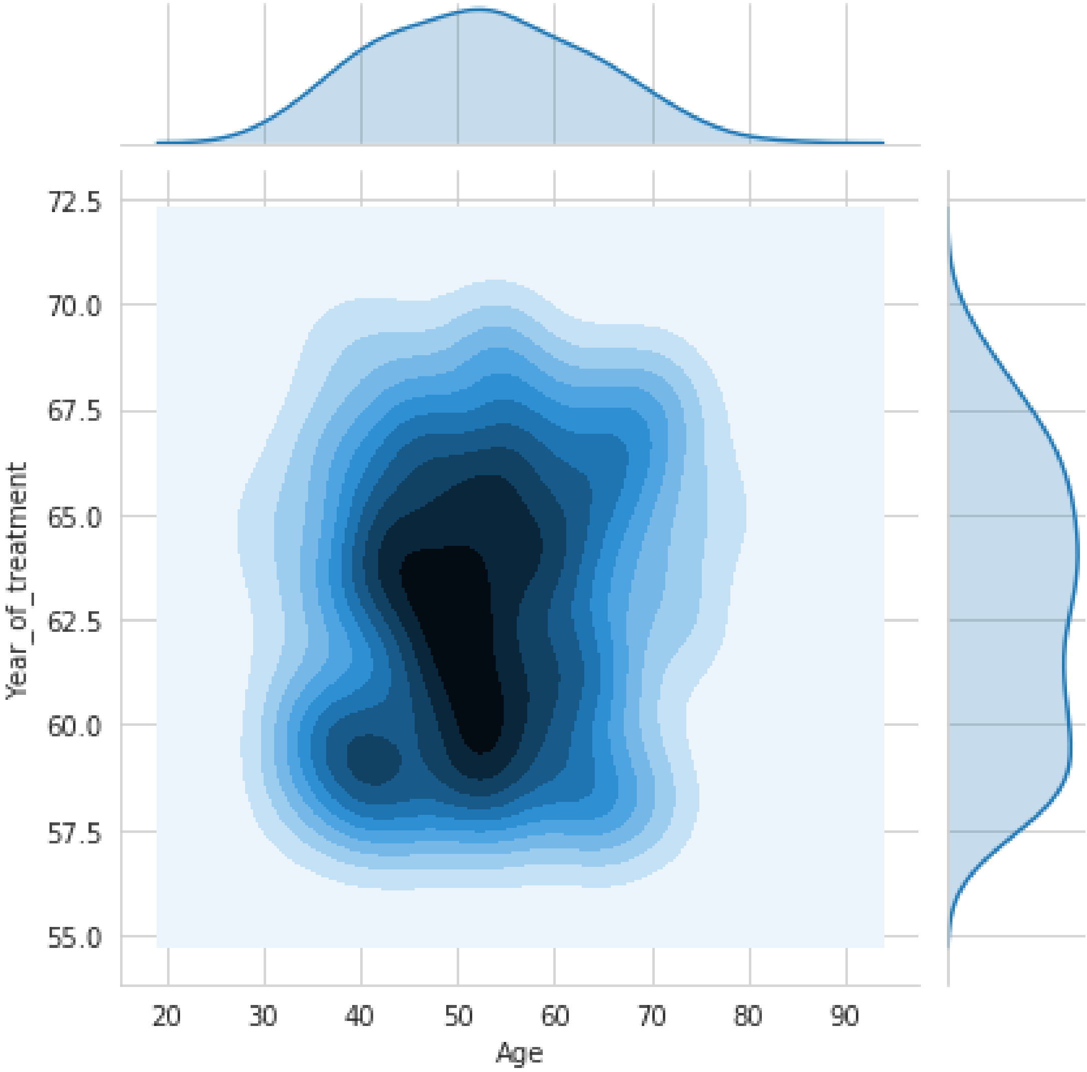


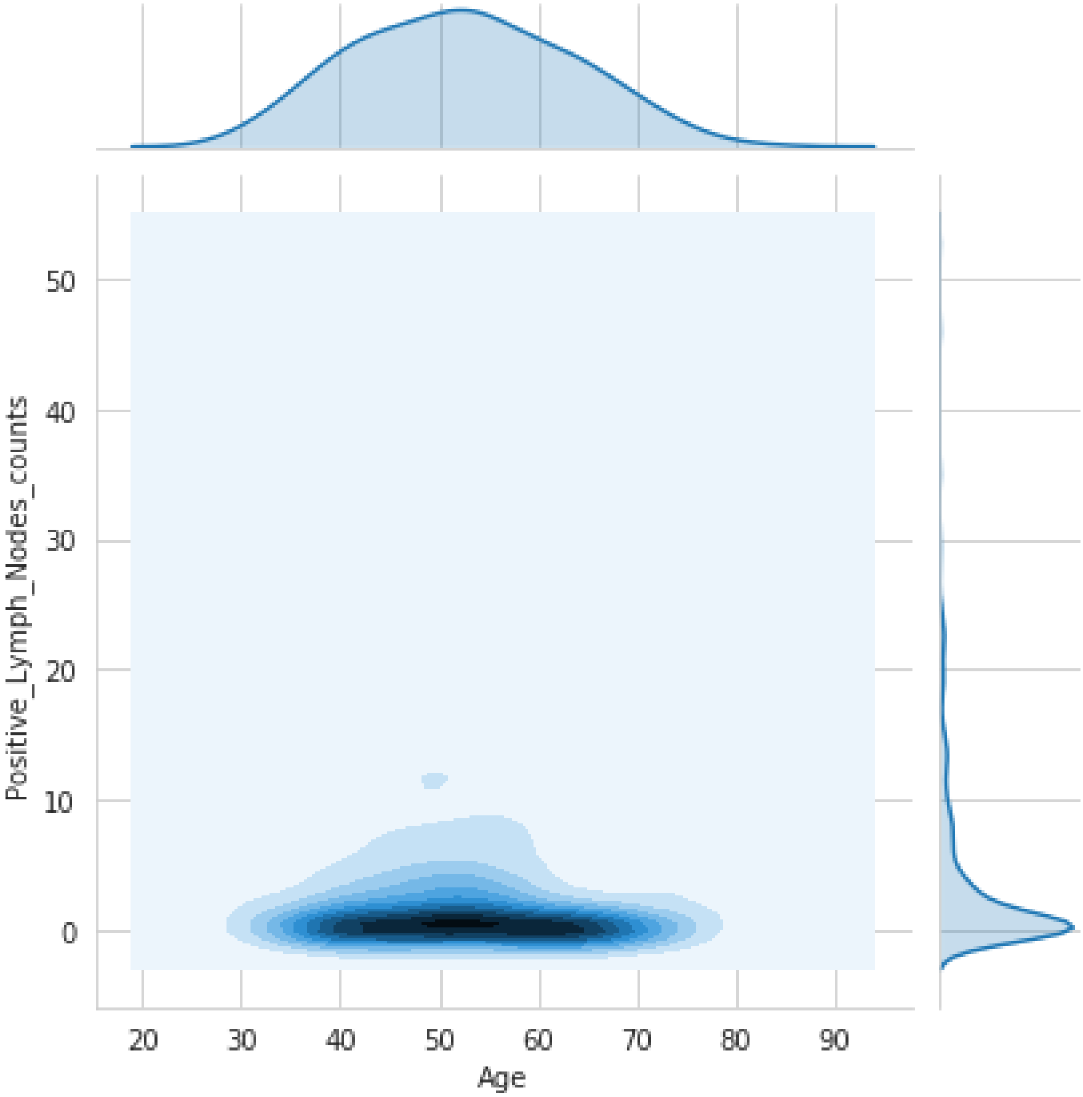
5.3 Observations:

The data is not separable through lines using any feature combinations,hence we can't use if-else condition to separate out

6 JoinPlot

```
In [21]: for feat1 in haberman.columns[:1]:
             for feat2 in haberman.columns[:-1]:
                 if feat1!=feat2:
                     sns.jointplot(x=feat1,y=feat2,data=haberman,kind='kde')
                     plt.show()
```


72.5
70.0
67.5
65.0
62.5
60.0
57.5
55.0
Year_of_treatment
20
30
40
50
60
70
80
90
Age



6.1 Observation:

Plot 1: - The plot is highly concentrated for age 50 to 60 and year 58 to 68
plot 2: - The lesser the number of positive lymph nodes the higher the chances of survival

7 FInal Conclusion

7.1 Patients with lesser (3 approx) postivie lymph nodes survival rate is higher

7.2 No of Positive Lymph nodes is most effective for getting the survival status

7.3 Younger people had more chances of survival also the people who were treated in later years are more likely to survive

```
In [ ]:
```