

EDA on Haberman Survival Dataset

Haberman's Survival Dataset:

Reference :

<https://towardsdatascience.com/will-habermans-survival-data-set-make-you-diagnose-cancer-8f40b3449673>
(<https://towardsdatascience.com/will-habermans-survival-data-set-make-you-diagnose-cancer-8f40b3449673>) this link included some amazing explanation on how to talk to the plots and gain insights from the same

Attribute Information on Haberman.csv file

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

Objective :

To predict Whether a patient will survive 5 years or more after the operation based on age , year of operation and the number of positive axillary nodes.

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

column = ['Patient_age', 'Year_of_operation', 'positive_axillary_nodes', 'Survival_status']
data = pd.read_csv("haberman.csv" , names = column)
data.head()
```

Out[1]:

	Patient_age	Year_of_operation	positive_axillary_nodes	Survival_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 [2]:

```
print("The number of data points and features respectively:" , data.shape)
print("\n")
print("Data points per class:")
data['Survival_status'].value_counts()
```

The number of data points and features respectively: (306, 4)

Data points per class:

Out[2]:

```
1    225
2     81
Name: Survival_status, dtype: int64
```

Observation:

- 1) There are two classes 1 and 2 , class 1 corresponds to patients who survived 5 years or longer after the operation and class 2 corresponds to patients who died within 5 years of operation.
- 2) class 1 contains 225 values and class 2 contain 81 values , since the number of values in each class differ substantially the dataset is imbalanced(225:81)).

In [3]:

```
data_survived = data.loc[data["Survival_status"] == 1]
data_died = data.loc[data["Survival_status"] == 2]

print("Basic statistics about patients who survived 5 years or longer after operation")
data_survived.describe()
```

Basic statistics about patients who survived 5 years or longer after operation

Out[3]:

	Patient_age	Year_of_operation	positive_axillary_nodes	Survival_status
count	225.000000	225.000000	225.000000	225.0
mean	52.017778	62.862222	2.791111	1.0
std	11.012154	3.222915	5.870318	0.0
min	30.000000	58.000000	0.000000	1.0
25%	43.000000	60.000000	0.000000	1.0
50%	52.000000	63.000000	0.000000	1.0
75%	60.000000	66.000000	3.000000	1.0
max	77.000000	69.000000	46.000000	1.0

In [4]:

```
print("Basic statistics about patients who died within 5 years of operation")  
data_died.describe()
```

Basic statistics about patients who died within 5 years of operation

Out[4]:

	Patient_age	Year_of_operation	positive_axillary_nodes	Survival_status
count	81.000000	81.000000	81.000000	81.0
mean	53.679012	62.827160	7.456790	2.0
std	10.167137	3.342118	9.185654	0.0
min	34.000000	58.000000	0.000000	2.0
25%	46.000000	59.000000	1.000000	2.0
50%	53.000000	63.000000	4.000000	2.0
75%	61.000000	65.000000	11.000000	2.0
max	83.000000	69.000000	52.000000	2.0

Observations:

About 75% of survived patients had positive axillary nodes less than or equal to 3 and 75% of patients who died had positive axillary nodes less than or equal to 11. Thus the parameter which varies the most in the above statistics is the number of positive axillary nodes. The patient with less number of positive axillary nodes has higher chance of survival. Number of positive axillary nodes alone is not suitable in determining the survival of patient as there are cases where the patient died within 5 years of operation even when she had no positive axillary nodes. so we cant depend on it.

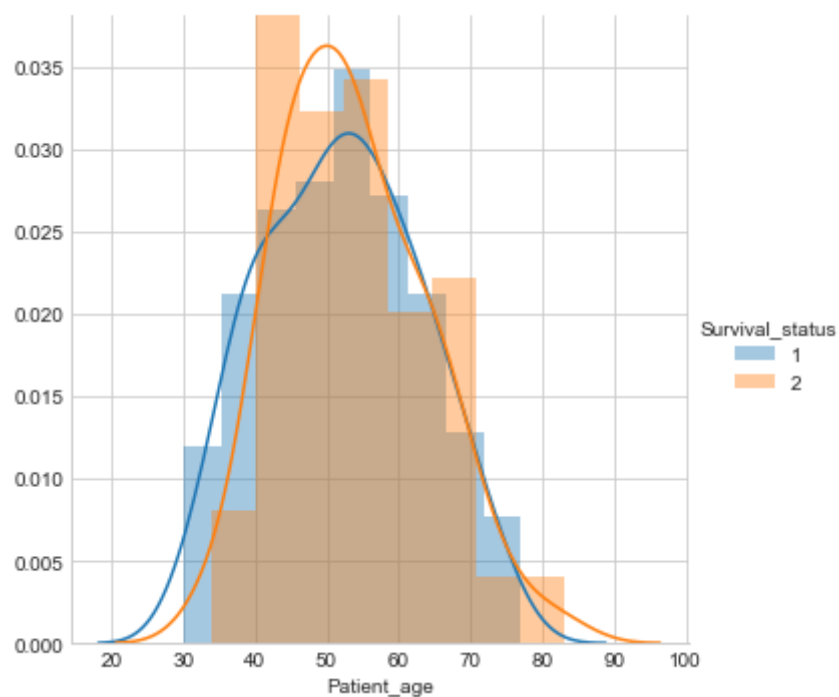
Univariate Analysis based on different features

1. Patient_age

1.1 Histogram with PDF

In [5]:

```
sns.set_style("whitegrid");  
sns.FacetGrid(data , hue = "Survival_status" , size = 5).map(sns.distplot , "Patient_age").add_legend();  
plt.show()
```



1.2 PDF and CDF plot

In [6]:

```

counts, bin_edges = np.histogram(data_survived['Patient_age'], bins=20, 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(data_died['Patient_age'], bins=20, 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.legend(['sur_pdf', 'sur_cdf', 'died_pdf', 'died_cdf'])

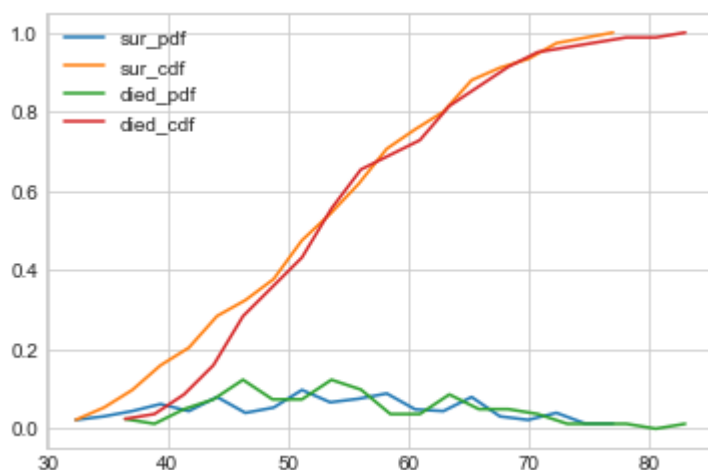
plt.show();

```

```

[0.02222222 0.03111111 0.04444444 0.06222222 0.04444444 0.08
 0.04      0.05333333 0.09777778 0.06666667 0.07555556 0.08888889
 0.04888889 0.04444444 0.08      0.03111111 0.02222222 0.04
 0.01333333 0.01333333]
[30.   32.35 34.7  37.05 39.4  41.75 44.1  46.45 48.8  51.15 53.5  55.85
 58.2  60.55 62.9  65.25 67.6  69.95 72.3  74.65 77.   ]
[0.02469136 0.01234568 0.04938272 0.07407407 0.12345679 0.07407407
 0.07407407 0.12345679 0.09876543 0.03703704 0.03703704 0.08641975
 0.04938272 0.04938272 0.03703704 0.01234568 0.01234568 0.01234568
 0.      0.01234568]
[34.   36.45 38.9  41.35 43.8  46.25 48.7  51.15 53.6  56.05 58.5  60.95
 63.4  65.85 68.3  70.75 73.2  75.65 78.1  80.55 83.   ]

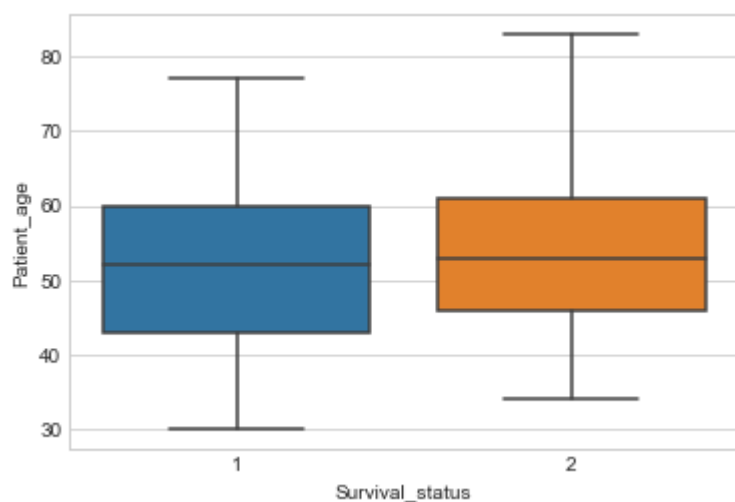
```



1.3 Box Plot

In [7]:

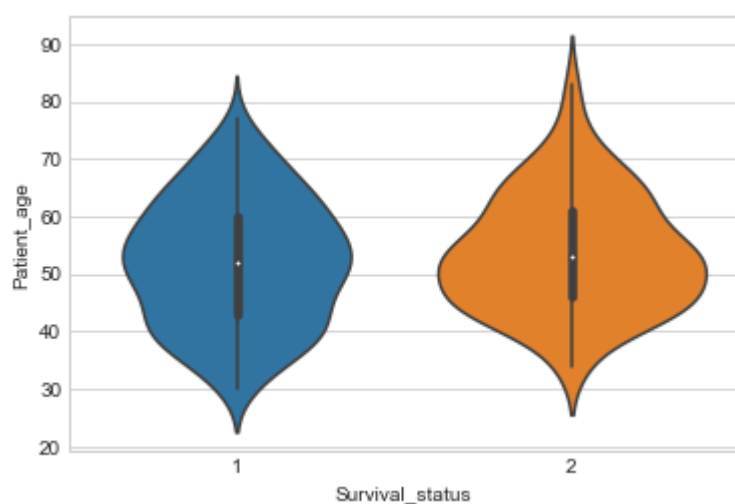
```
sns.boxplot(x='Survival_status',y='Patient_age', data= data)
plt.show()
```



1.4 Violin Plot

In [8]:

```
sns.violinplot(x="Survival_status", y="Patient_age", data = data, size=8)
plt.show()
```



Observation:

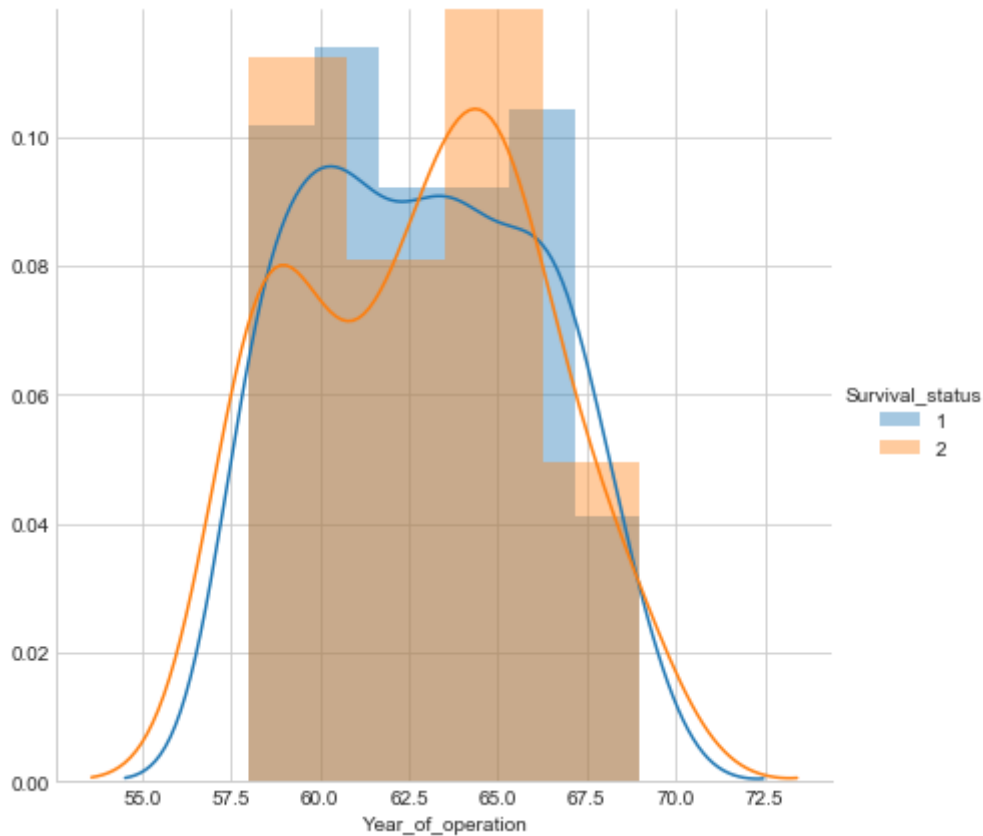
Univariate analysis on age of the patients gives similar show too much similarity (high overlap) age of the patient alone is not an important parameter in determining the survival.

2. Year_of_operation

2.1 Histogram with PDF

In [9]:

```
sns.set_style("whitegrid");  
sns.FacetGrid(data , hue = "Survival_status" , size = 6).map(sns.distplot , "Year_of_op  
eration").add_legend();  
plt.show()
```



2.2 PDF and CDF plot

In [10]:

```
counts, bin_edges = np.histogram(data_survived['Year_of_operation'], bins= 30, 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(data_died['Year_of_operation'], bins=30, 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.legend(['sur_pdf', 'sur_cdf', 'died_pdf', 'died_cdf'])

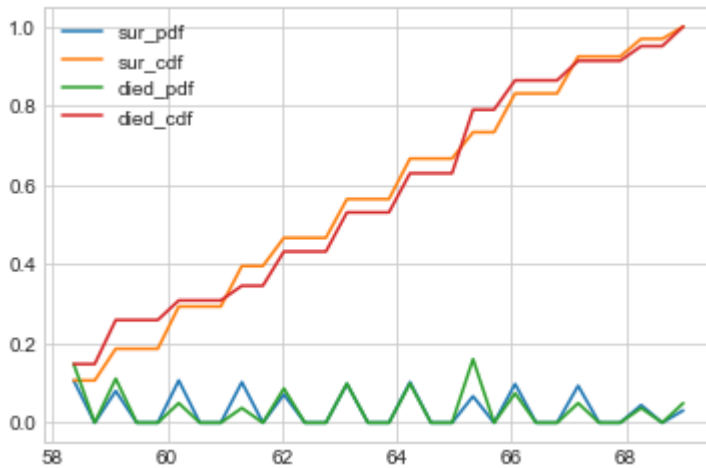
plt.show();
```



```

[0.10666667 0.          0.08          0.          0.          0.10666667
 0.          0.          0.10222222 0.          0.07111111 0.
 0.          0.09777778 0.          0.          0.10222222 0.
 0.          0.06666667 0.          0.09777778 0.          0.
 0.09333333 0.          0.          0.04444444 0.          0.03111111]
[58.          58.36666667 58.73333333 59.1          59.46666667 59.83333333
 60.2          60.56666667 60.93333333 61.3          61.66666667 62.03333333
 62.4          62.76666667 63.13333333 63.5          63.86666667 64.23333333
 64.6          64.96666667 65.33333333 65.7          66.06666667 66.43333333
 66.8          67.16666667 67.53333333 67.9          68.26666667 68.63333333
 69.          ]
[0.14814815 0.          0.11111111 0.          0.          0.04938272
 0.          0.          0.03703704 0.          0.08641975 0.
 0.          0.09876543 0.          0.          0.09876543 0.
 0.          0.16049383 0.          0.07407407 0.          0.
 0.04938272 0.          0.          0.03703704 0.          0.04938272]
[58.          58.36666667 58.73333333 59.1          59.46666667 59.83333333
 60.2          60.56666667 60.93333333 61.3          61.66666667 62.03333333
 62.4          62.76666667 63.13333333 63.5          63.86666667 64.23333333
 64.6          64.96666667 65.33333333 65.7          66.06666667 66.43333333
 66.8          67.16666667 67.53333333 67.9          68.26666667 68.63333333
 69.          ]

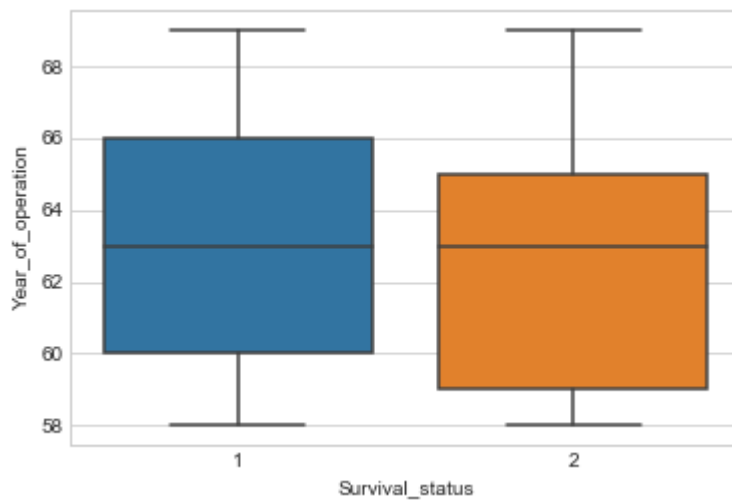
```



2.3 Box Plot

In [11]:

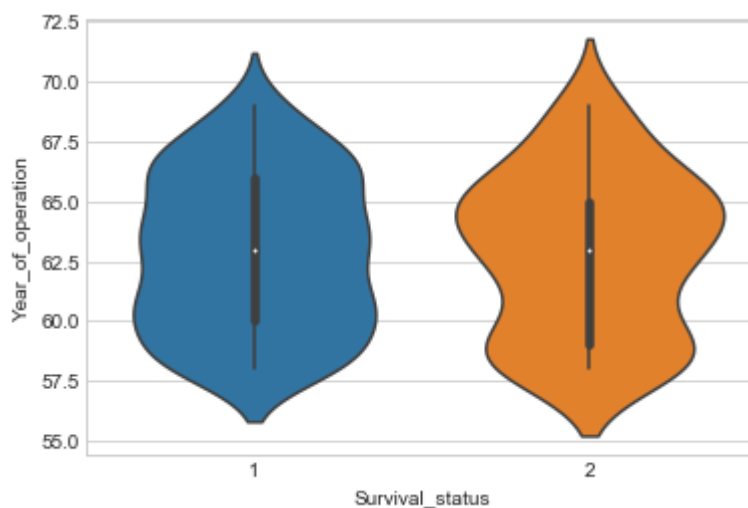
```
sns.boxplot(x='Survival_status',y='Year_of_operation', data= data)  
plt.show()
```



2.4 Violin Plot

In [12]:

```
sns.violinplot(x="Survival_status", y="Year_of_operation", data = data, size=8)  
plt.show()
```



Observation:

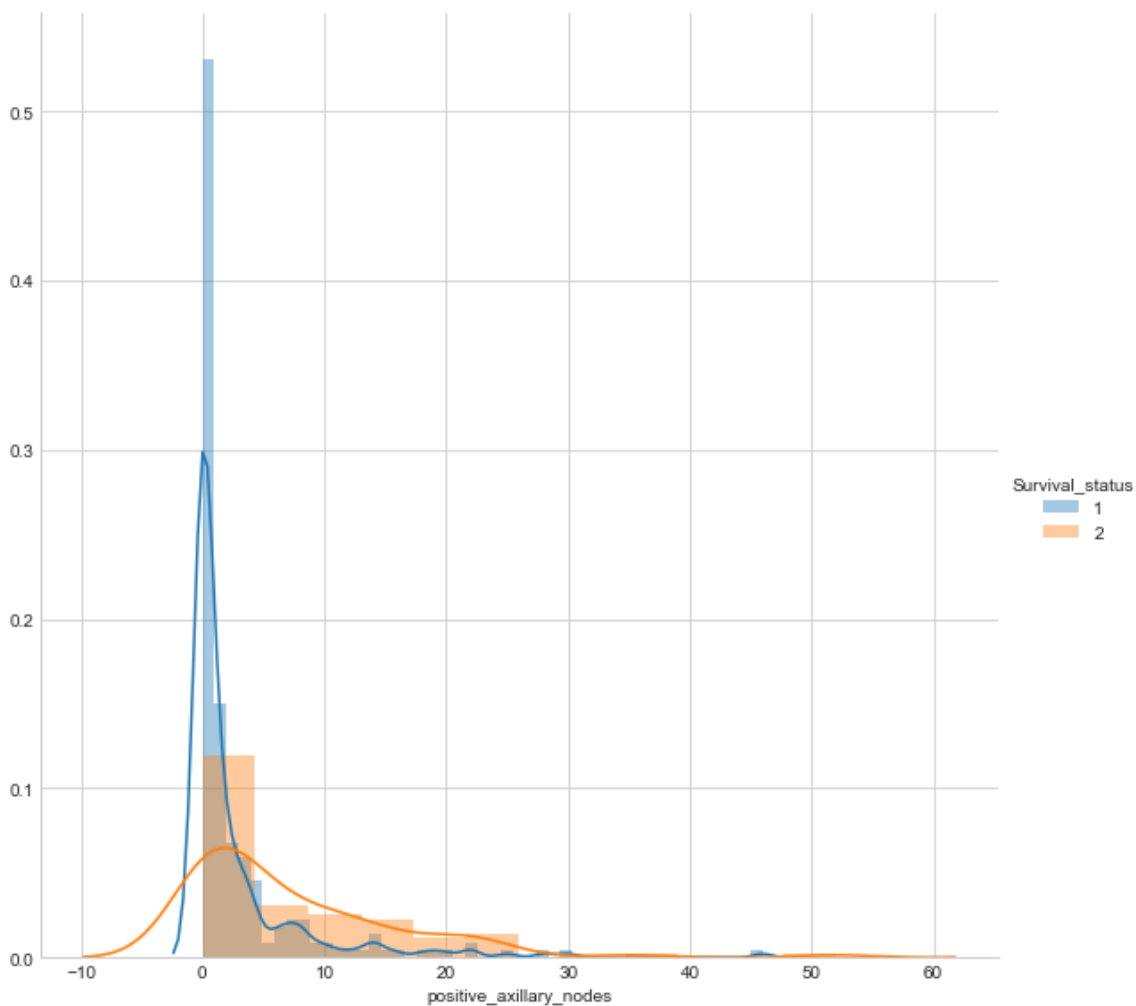
Survival doesn't depend on the year of operation

3. positive_axillary_nodes

3.1 Histogram with PDF

In [13]:

```
sns.set_style("whitegrid");  
sns.FacetGrid(data , hue = "Survival_status" , size = 8).map(sns.distplot , "positive_axillary_nodes").add_legend();  
plt.show()
```



3.2 PDF and CDF Plot

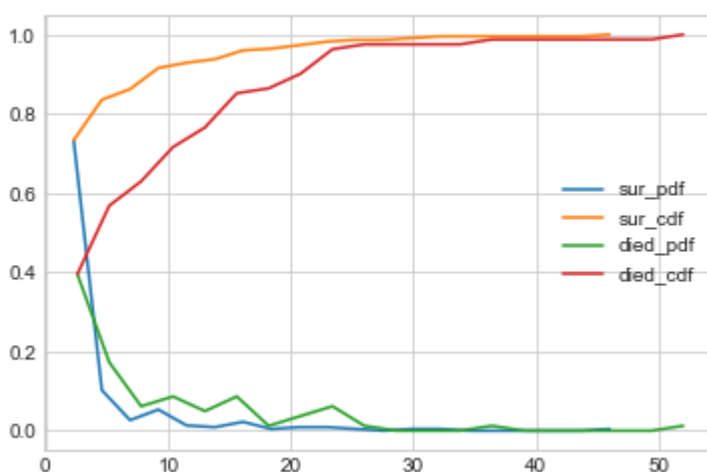
In [14]:

```
counts, bin_edges = np.histogram(data_survived['positive_axillary_nodes'], bins=20, 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(data_died['positive_axillary_nodes'], bins=20, 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.legend(['sur_pdf', 'sur_cdf', 'died_pdf', 'died_cdf'])

plt.show()
```

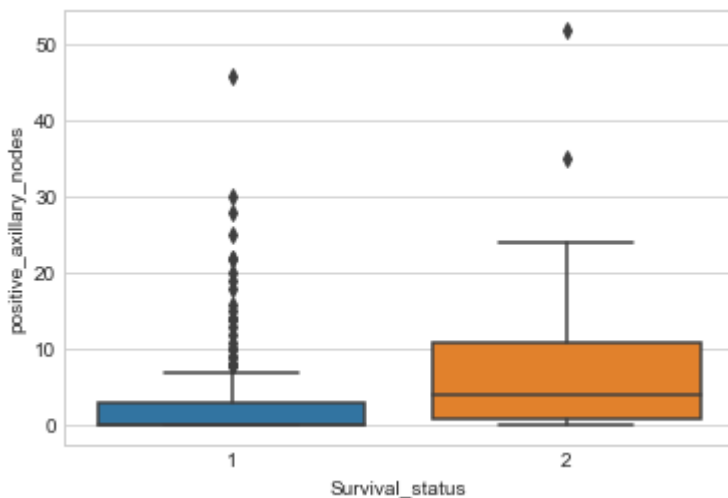
```
[0.73333333 0.10222222 0.02666667 0.05333333 0.01333333 0.00888889
 0.02222222 0.00444444 0.00888889 0.00888889 0.00444444 0.
 0.00444444 0.00444444 0. 0. 0. 0.
 0. 0.00444444]
[ 0.  2.3  4.6  6.9  9.2 11.5 13.8 16.1 18.4 20.7 23.  25.3 27.6 29.9
 32.2 34.5 36.8 39.1 41.4 43.7 46. ]
[0.39506173 0.17283951 0.0617284  0.08641975 0.04938272 0.08641975
 0.01234568 0.03703704 0.0617284  0.01234568 0. 0.
 0. 0.01234568 0. 0. 0. 0.
 0. 0.01234568]
[ 0.  2.6  5.2  7.8 10.4 13.  15.6 18.2 20.8 23.4 26.  28.6 31.2 33.8
 36.4 39.  41.6 44.2 46.8 49.4 52. ]
```



3.3 Box Plot

In [15]:

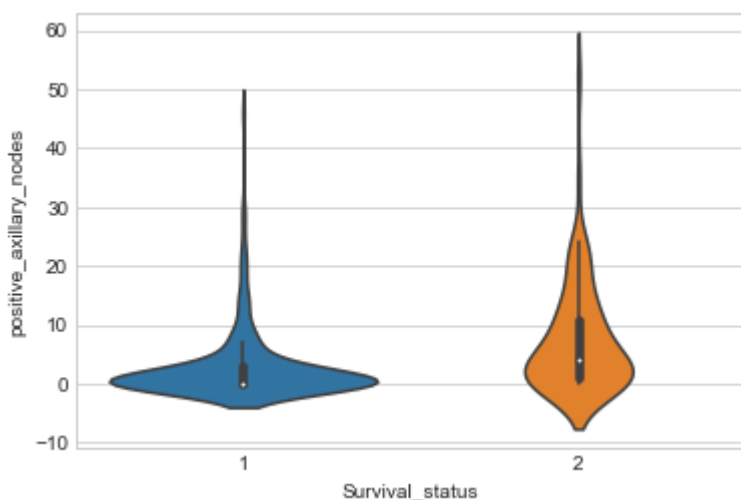
```
sns.boxplot(x='Survival_status',y='positive_axillary_nodes', data= data)
plt.show()
```



3.4 Violin Plot

In [16]:

```
sns.violinplot(x="Survival_status", y="positive_axillary_nodes", data = data, size=8)
plt.show()
```



Observation

Overlap of data points is less in comparison to other features. difficult to set a threshold for positive axillary nodes which will differentiate one class from another. About 50% of the patients who survived had no positive axillary nodes. As number of positive axillary nodes increase the chance of survival decreases accordingly. Absence of positive axillary nodes cannot always guarantee survival as a small percentage of patients who had zero positive axillary nodes dies within 5 yrs of treatment

Bivariate Analysis

Pair Plot

In [17]:

```
plt.close();
data["Survival_status"] = data["Survival_status"].apply(lambda x: "Survived" if x == 1
else "Died")
sns.set_style("whitegrid");
sns.pairplot(data , hue="Survival_status", size=3);
plt.show();
```



Observation:

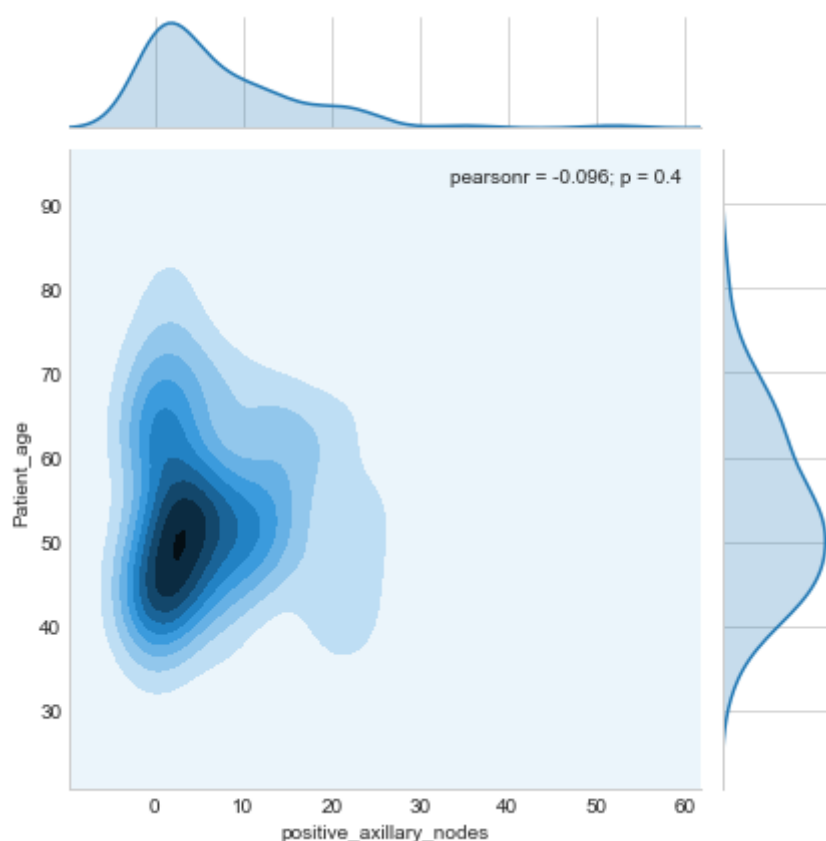
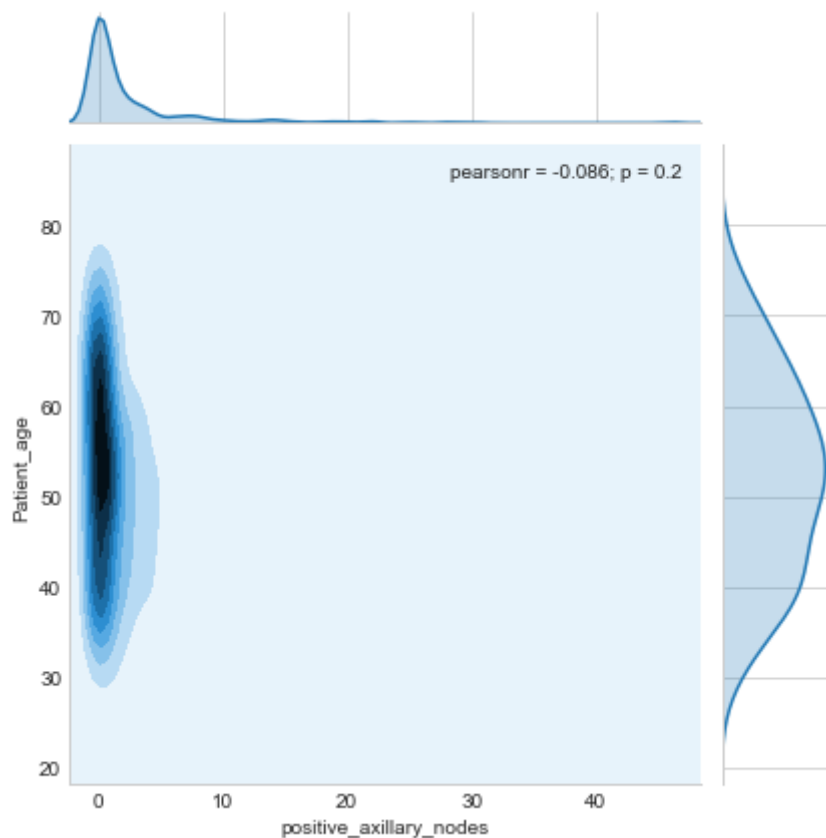
it is difficult to classify a patient based on his features

Multivariate Analysis

Contour Plot

In [18]:

```
sns.jointplot(x="positive_axillary_nodes", y="Patient_age", data = data_survived , kind  
="kde");  
plt.show();  
  
sns.jointplot(x="positive_axillary_nodes", y="Patient_age", data = data_died , kind="kd  
e");  
plt.show();
```



Observation:

age and number of positive axillary nodes are independent of one another