In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
sns.set()
import os
os.getcwd()
```

Out[2]:
```
'C:\\Users\\Soni'
```

In [3]:
```
Hb_df = pd.read_csv("haberman.csv", header=None,names=['Age', 'Op_Year', 'Axil_no
```

In [4]:
```
print (Hb_df.shape)
```

```
(306, 4)
```

In [5]:
```
print (Hb_df.head())
```

```
   Age  Op_Year  Axil_nodes_det  Surv_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 [6]:
```
#Printing all the columns
print (Hb_df.columns)
```

```
Index(['Age', 'Op_Year', 'Axil_nodes_det', 'Surv_status'], dtype='object')
```

In [7]:
```
#Print the basic information to find null points
print (Hb_df.info())
```

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

Observations :

1. There are total 305 entries with 4 columns.
2. There is no empty or null values in the data set.
3. Status : 1= Patients survived 5 years or longer.
   2= Patients did not survived 5 years

In [8]:
```
#Top 5 Data Set
Hb_df.head()
```

Out[8]:

	Age	Op_Year	Axil_nodes_det	Surv_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 [9]:
```
#High level statistics of the datase
print (Hb_df.describe())
```

```
              Age     Op_Year  Axil_nodes_det  Surv_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
```

Observations :

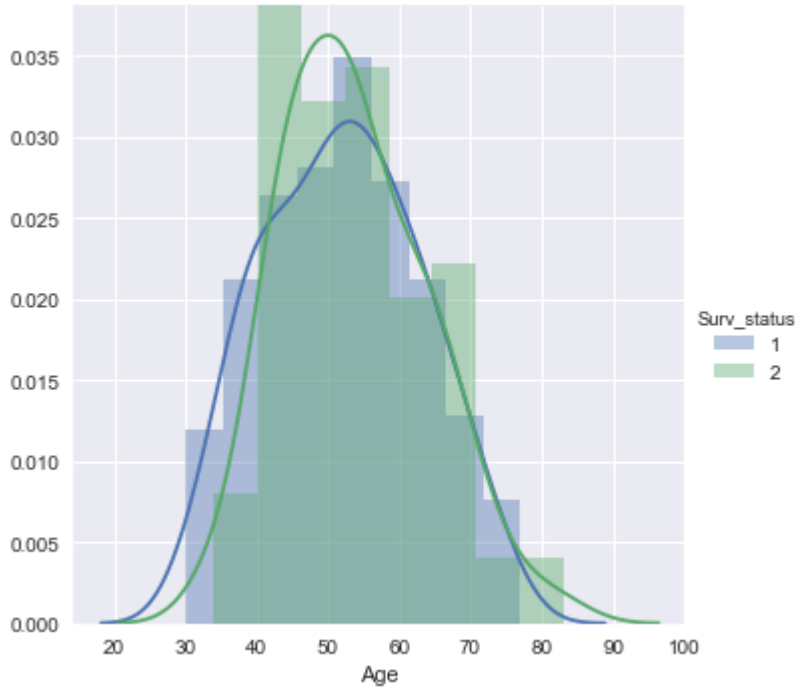
1. The Average mean of the Age is 52 years,those diagnosed with cancer ranging from 30 years to 83 yers.
2. The Axiliary nodes detected ranges from 0 to 52%.
3. The 25% of the patients have 0 Positive axiliary nodes where as 50% have 1 positive axiliary nodes and 75% have 4 positive axiliary nodes.

Objective : The main objective is to see whether the patient survived after 5 years or not based on Age,Operation year and Positive auxiliary nodes.

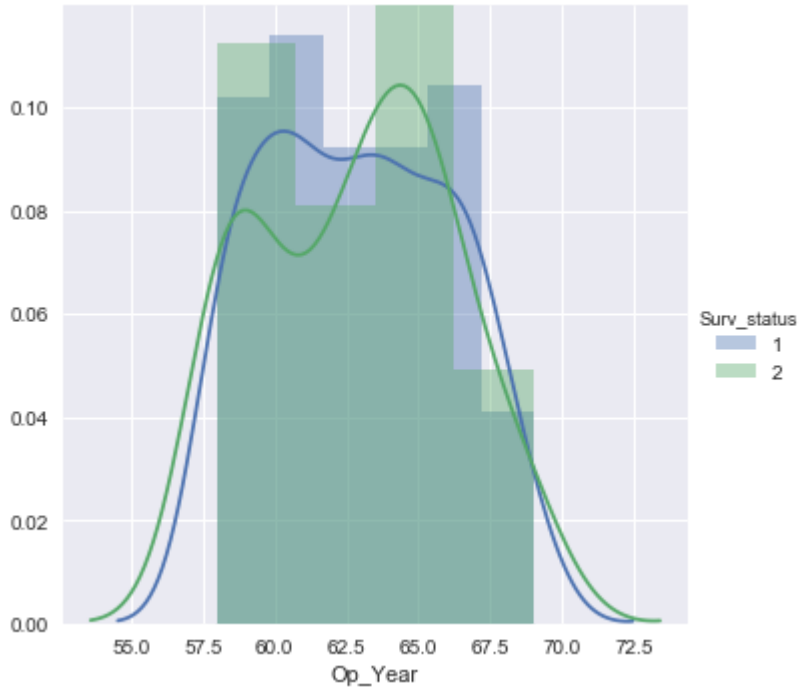
PDF

In [14]:
```
for i, feature in enumerate(list(Hb_df.columns)[:-1]):
    fg = sns.FacetGrid(Hb_df , hue='Surv_status' , size = 5)
    fg.map(sns.distplot, feature).add_legend()
    plt.show()
```

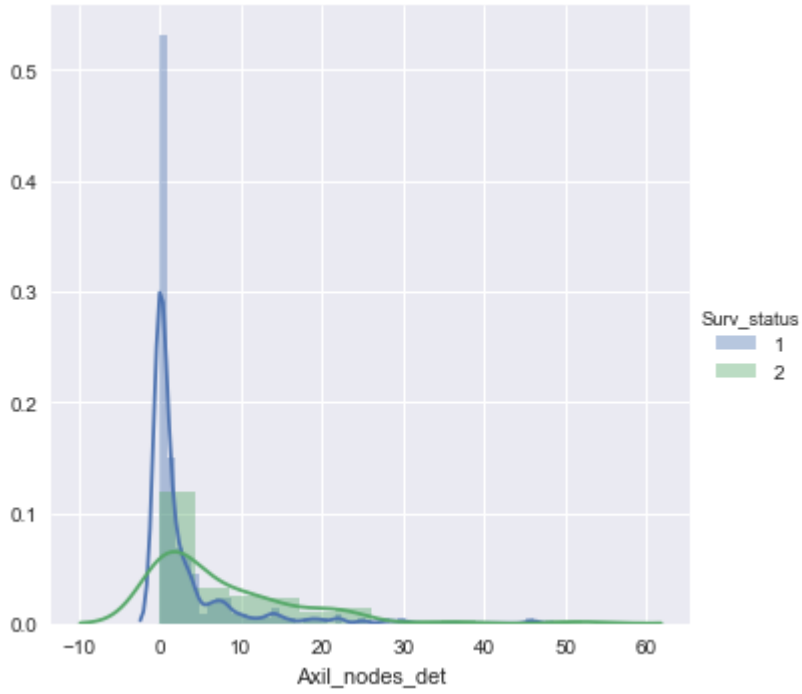
```
C:\Users\Soni\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarn
ing: Using a non-tuple sequence for multidimensional indexing is deprecated; us
e `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpret
ed as an array index, `arr[np.array(seq)]`, which will result either in an erro
r or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
C:\Users\Soni\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWa
rning: The 'normed' kwarg is deprecated, and has been replaced by the 'density'
kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\Users\Soni\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWa
rning: The 'normed' kwarg is deprecated, and has been replaced by the 'density'
kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```




```
C:\Users\Soni\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWa
rning: The 'normed' kwarg is deprecated, and has been replaced by the 'density'
kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\Users\Soni\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWa
rning: The 'normed' kwarg is deprecated, and has been replaced by the 'density'
kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```




```
C:\Users\Soni\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWa
rning: The 'normed' kwarg is deprecated, and has been replaced by the 'density'
kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\Users\Soni\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWa
rning: The 'normed' kwarg is deprecated, and has been replaced by the 'density'
kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



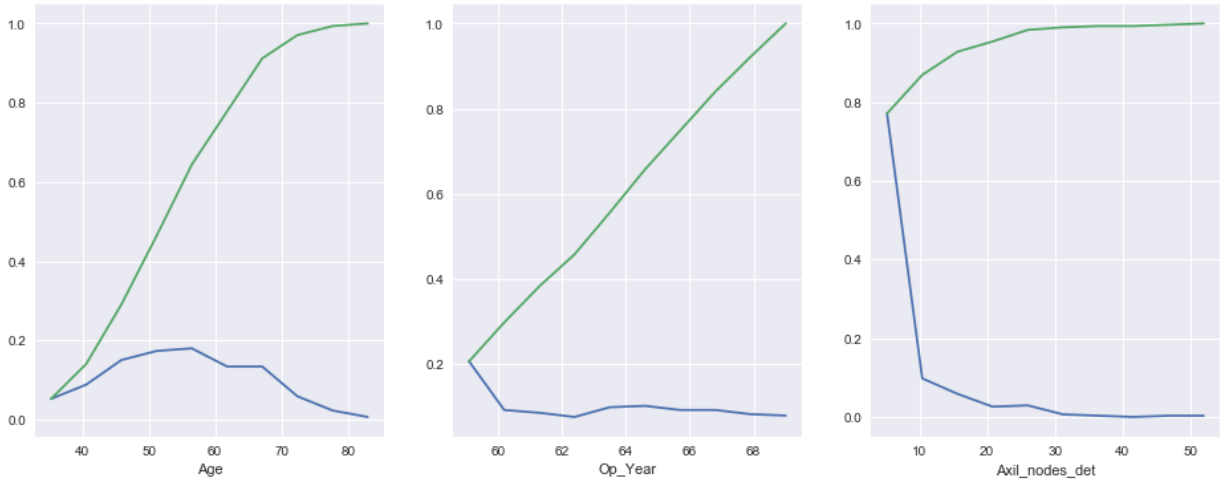
Observations :

1. We get an overlapping graph for Age and operation year for the People who survived and the people didn't survived.

2. For PDF Graph for Positive Axil nodes depicts that they are less overlapping.They re seperable.

In [9]:
```
#CDF of the people didn't survived
plt.figure(figsize=(16,6))
for i, feature in enumerate(list(Hb_df.columns)[:-1]):
    plt.subplot(1, 3, i+1)
    print("********* "+feature+" *********")
    counts, bin_edges = np.histogram(Hb_df[feature], bins=10, density=True)
    print("Bin Edges: {}".format(bin_edges))
    pdf = counts/sum(counts)
    print("PDF: {}".format(pdf))
    cdf = np.cumsum(pdf)
    print("CDF: {}".format(cdf))
    plt.plot(bin_edges[1:], pdf, bin_edges[1:], cdf)
    plt.xlabel(feature)
```

```
********* Age *********
Bin Edges: [30.  35.3 40.6 45.9 51.2 56.5 61.8 67.1 72.4 77.7 83. ]
PDF: [0.05228758 0.08823529 0.1503268  0.17320261 0.17973856 0.13398693
 0.13398693 0.05882353 0.02287582 0.00653595]
CDF: [0.05228758 0.14052288 0.29084967 0.46405229 0.64379085 0.77777778
 0.91176471 0.97058824 0.99346405 1.        ]
********* Op_Year *********
Bin Edges: [58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
PDF: [0.20588235 0.09150327 0.08496732 0.0751634  0.09803922 0.10130719
 0.09150327 0.09150327 0.08169935 0.07843137]
CDF: [0.20588235 0.29738562 0.38235294 0.45751634 0.55555556 0.65686275
 0.74836601 0.83986928 0.92156863 1.        ]
********* Axil_nodes_det *********
Bin Edges: [ 0.   5.2 10.4 15.6 20.8 26.  31.2 36.4 41.6 46.8 52. ]
PDF: [0.77124183 0.09803922 0.05882353 0.02614379 0.02941176 0.00653595
 0.00326797 0.         0.00326797 0.00326797]
CDF: [0.77124183 0.86928105 0.92810458 0.95424837 0.98366013 0.99019608
 0.99346405 0.99346405 0.99673203 1.        ]
```

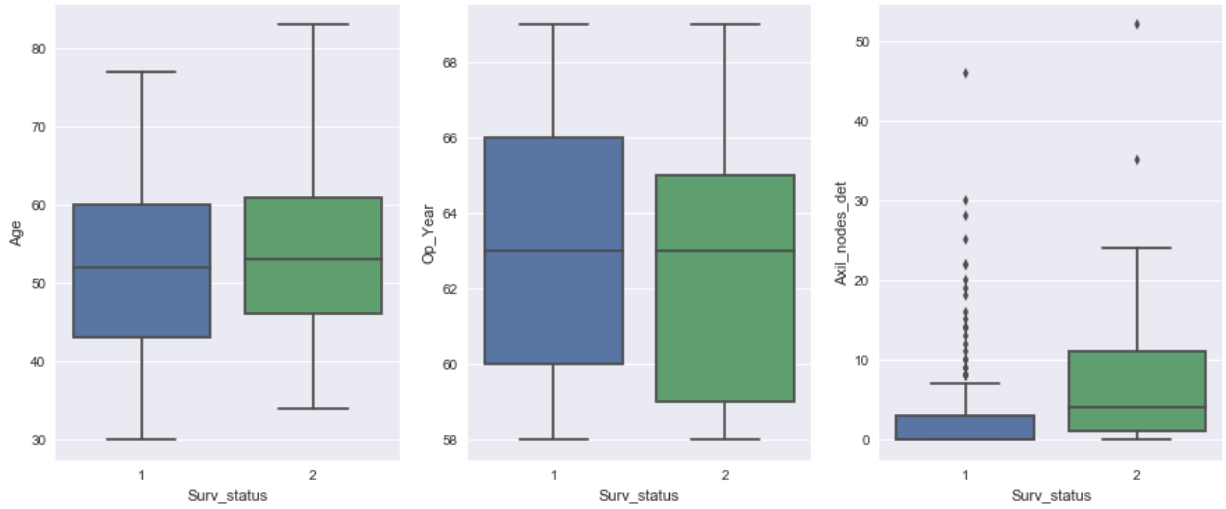


Observation :

1. Below the age of 55, 65% of the patients survived. So lower the age, the higher would be the chance of getting survived.

2. After the Year 1966, higher no. of patients survived the operation and lived longer than 5 years.
3. The positive axilliary node with value 25, 90% of the patients survived.

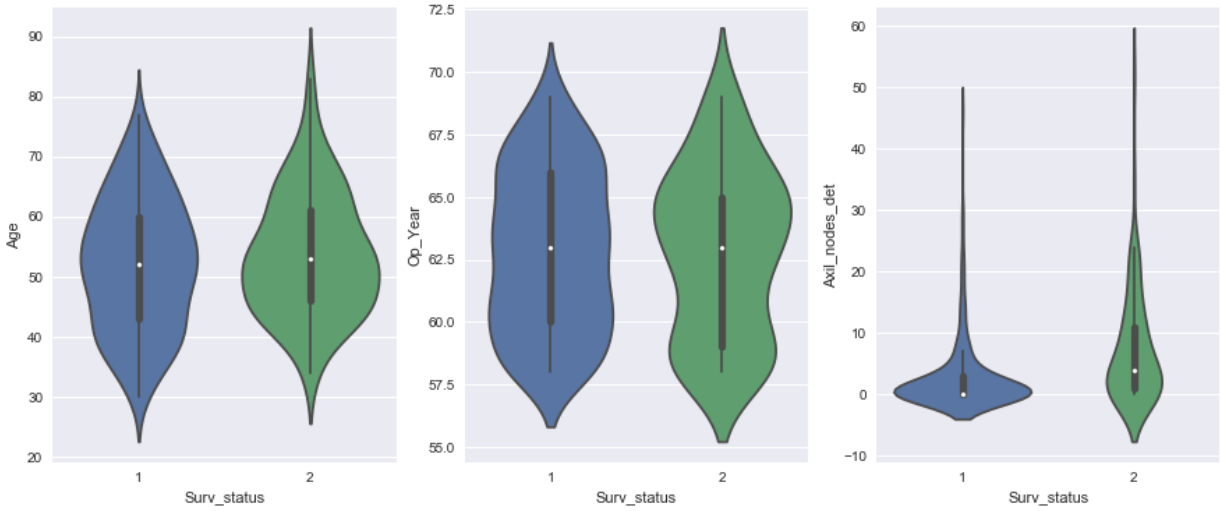
In [12]:
```
#Box Plot
fig,axes = plt.subplots(1, 3 , figsize=(15,6))
for i,feature in enumerate(list(Hb_df.columns)[:-1]):
     sns.boxplot(x='Surv_status',y=feature, data=Hb_df , ax=axes[i])
plt.show()
```



Observations :

1. The 25th Percentile to 75 percentile for Age ranges from 43 to 60 where the patient survives. The 25th Percentile to 75 percentile for Age ranges from 45 to 61 where the patient didn't survived.
2. The 25th Percentile to 75 percentile for Opertion year ranges from 60 to 66 where the patient survives. The 25th Percentile to 75 percentile for Opertion year ranges from 59 to 65 where the patient didn't survived..
3. The 25th Percentile to 75 percentile for Positive auxiliary nodes ranges from 0 to 2 where the patient survives. The 25th Percentile to 75 percentile for Positive auxiliary nodes ranges from 1 to 12 where the patient survives.

In [13]:
```
# Violin Plot
fig,axes = plt.subplots(1, 3 , figsize=(15,6))
for i,feature in enumerate(list(Hb_df.columns)[:-1]):
     sns.violinplot(x='Surv_status',y=feature, data=Hb_df , ax=axes[i])
plt.show()
```



In [37]:
```
#Pair Plot
sns.set_style("whitegrid");
sns.pairplot(Hb_df, hue='Surv_status', vars=[Hb_df.columns[0],Hb_df.columns[1],Hb
plt.show()
```

Observation :

1. The Pair plots determines that both the Survival status,i.e the one which survived and the one which didn't survived, both are not linearly seperable.The graphs overlap each other.