EDA Assignment

In [11]:

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

Exercise:

1. Download Haberman Cancer Survival dataset from Kaggle. You may have to create a Kaggle account to donwload data. (https://www.kaggle.com/gilsousa/habermans-survival-data-set (https://www.kaggle.com/gilsousa/habermans-survival-data-set))
2. Perform a similar alanlaysis as above on this dataset with the following sections:
3. High level statistics of the dataset: number of points, numer of features, number of classes, data-points per class.
4. Explain our objective.
5. Perform Univaraite analysis(PDF, CDF, Boxplot, Voilin plots) to understand which features are useful towards classification.
6. Perform Bi-variate analysis (scatter plots, pair-plots) to see if combinations of features are useful in classfication.
7. Write your observations in english as crisply and unambigously as possible. Always quantify your results.

In [3]:

```
haberman = pd.read_csv("haberman.csv",names=['Age','Op_Year','axil_nodes','Surv_
status'])
```

In [4]:

```
# (Q) how many data-points and features?
print (haberman.shape)
```

```
(306, 4)
```

In [5]:

```
#(Q) What are the column names in our dataset?
print (haberman.columns)
```

```
Index(['Age', 'Op_Year', 'axil_nodes', 'Surv_status'], dtype='objec
t')
```

In [6]:

```
#(Q) How many data points for each class are present?
haberman["Surv_status"].value_counts()
```

Out[6]:

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

In [19]:

```
from collections import Counter
cnt = Counter()
for word in haberman['axil_nodes']:
    cnt[word]+=1
cnt
```

Out[19]:

```
Counter({1: 41,
         3: 20,
         0: 136,
         2: 20,
         4: 13,
         10: 3,
         9: 6,
         30: 1,
         7: 7,
         13: 5,
         6: 7,
         15: 3,
         21: 1,
         11: 4,
         5: 6,
         23: 3,
         8: 7,
         20: 2,
         52: 1,
         14: 4,
         19: 3,
         16: 1,
         12: 2,
         24: 1,
         46: 1,
         18: 1,
         22: 3,
         35: 1,
         17: 1,
         25: 1,
         28: 1})
```

Objective

Understand which features or combination of features can be useful towards classification

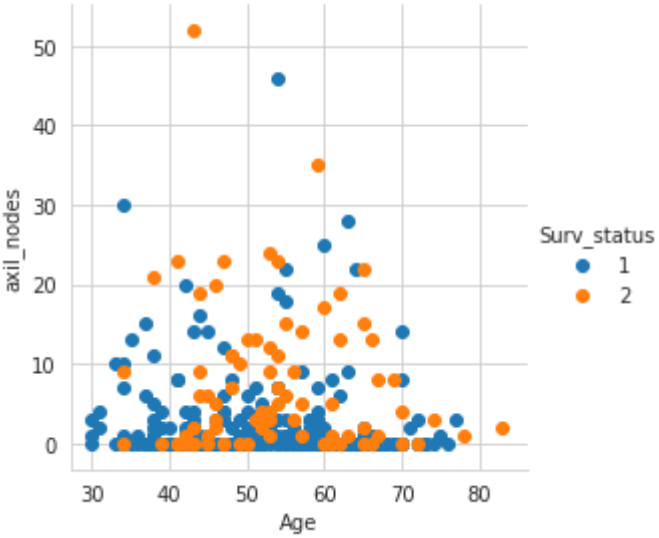
2D plots

Scatter plots

In [7]:

```
sns.set_style("whitegrid");
sns.FacetGrid(haberman, hue="Surv_status", size=4) \
   .map(plt.scatter, "Age", "axil_nodes") \
   .add_legend();
plt.show();
```

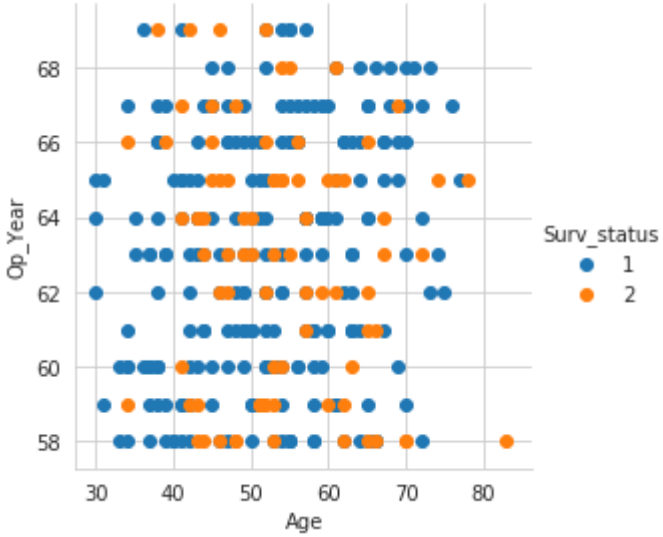
```
/home/admin1/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.
py:230: UserWarning: The `size` paramter has been renamed to `height
`; please update your code.
  warnings.warn(msg, UserWarning)
```



In [8]:

```
sns.set_style("whitegrid");
sns.FacetGrid(haberman, hue="Surv_status", size=4) \
   .map(plt.scatter, "Age", "Op_Year") \
   .add_legend();
plt.show();
```

```
/home/admin1/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.
py:230: UserWarning: The `size` paramter has been renamed to `height
`; please update your code.
  warnings.warn(msg, UserWarning)
```



Summary

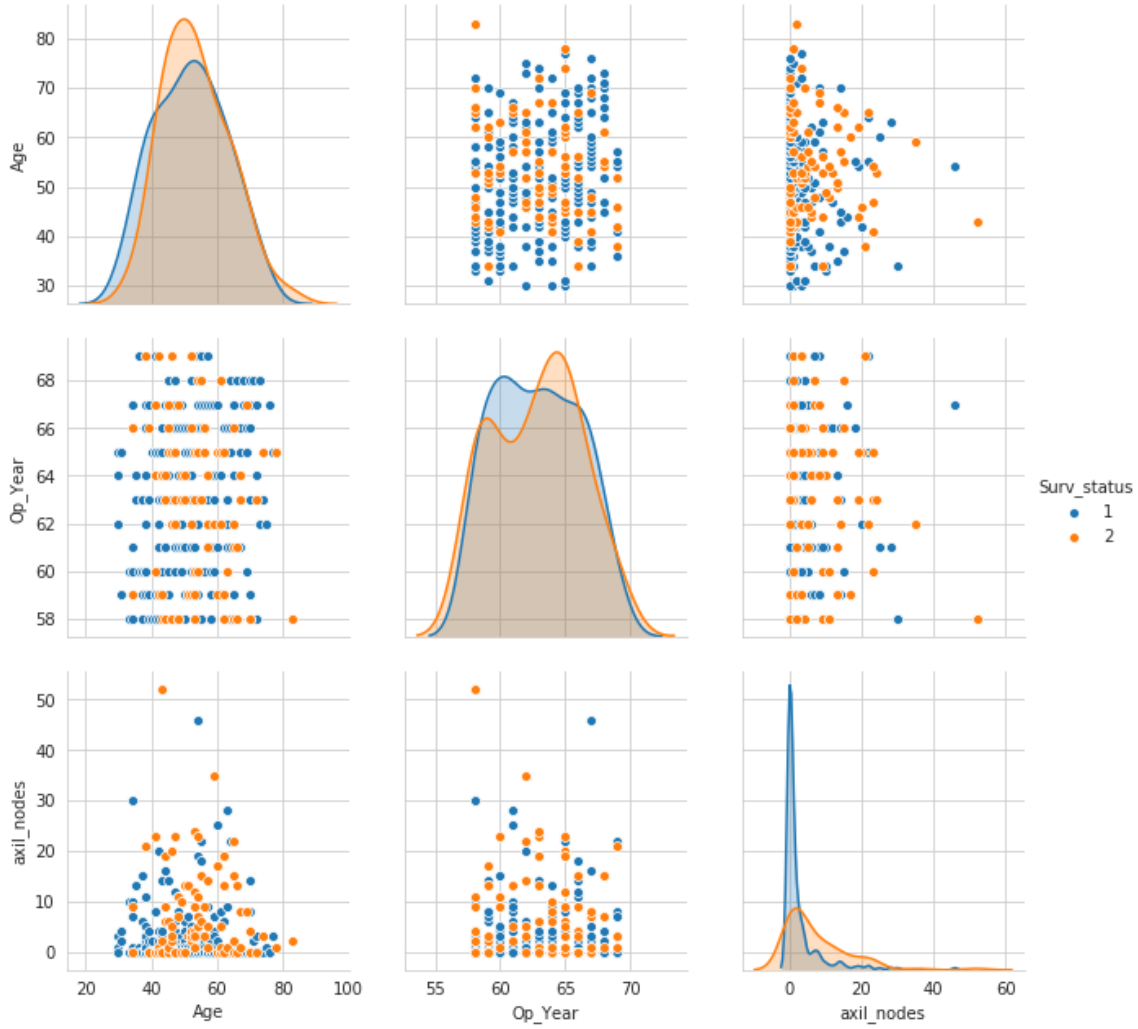
1. Age and axil_nodes cannot used for classification
2. Age and Op_year cannot used for classification

Pair plots

In [9]:

```
plt.close();
sns.set_style("whitegrid");
sns.pairplot(haberman, hue="Surv_status",x_vars=['Age', 'Op_Year', 'axil_nodes'
],y_vars=['Age', 'Op_Year', 'axil_nodes'] ,size=3);
plt.show()
```

```
/home/admin1/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.
py:2065: UserWarning: The `size` parameter has been renamed to `heig
ht`; pleaes update your code.
  warnings.warn(msg, UserWarning)
/home/admin1/anaconda3/lib/python3.7/site-packages/scipy/stats/stat
s.py:1713: FutureWarning: Using a non-tuple sequence for multidimens
ional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr
[seq]`. In the future this will be interpreted as an array index, `a
rr[np.array(seq)]`, which will result either in an error or a differ
ent result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumva
l
```



Summary

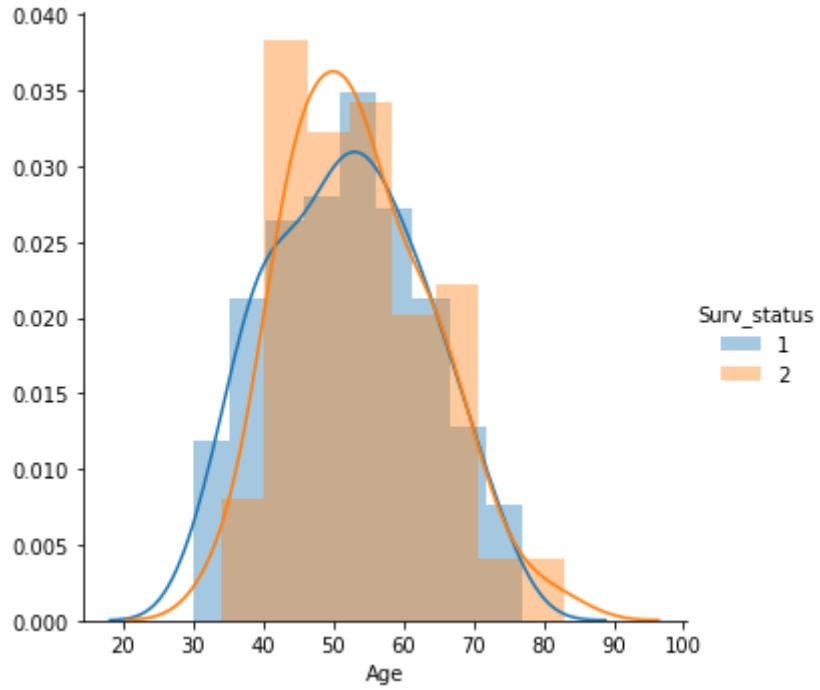
We can identity any two features which can used in classifying

Univariate Analysis

In [18]:

```
sns.FacetGrid(haberman, hue="Surv_status", size=5) \
   .map(sns.distplot, "Age") \
   .add_legend();
plt.show();
```

```
/home/admin1/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.
py:230: UserWarning: The `size` paramter has been renamed to `height
`; please update your code.
  warnings.warn(msg, UserWarning)
/home/admin1/anaconda3/lib/python3.7/site-packages/scipy/stats/stat
s.py:1713: FutureWarning: Using a non-tuple sequence for multidimens
ional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr
[seq]`. In the future this will be interpreted as an array index, `a
rr[np.array(seq)]`, which will result either in an error or a differ
ent result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumva
l
```



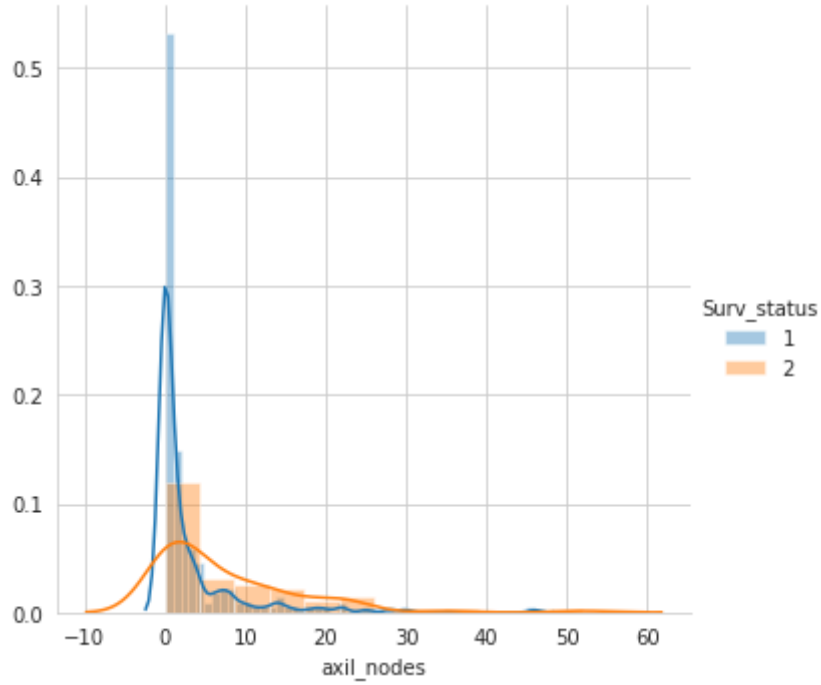
Summary

For age classes 1 and 2 are overlapping. So Age cannot be used as independent feature for classification

In [12]:

```
sns.FacetGrid(haberman, hue="Surv_status", size=5) \
   .map(sns.distplot, "axil_nodes") \
   .add_legend();
plt.show();
```

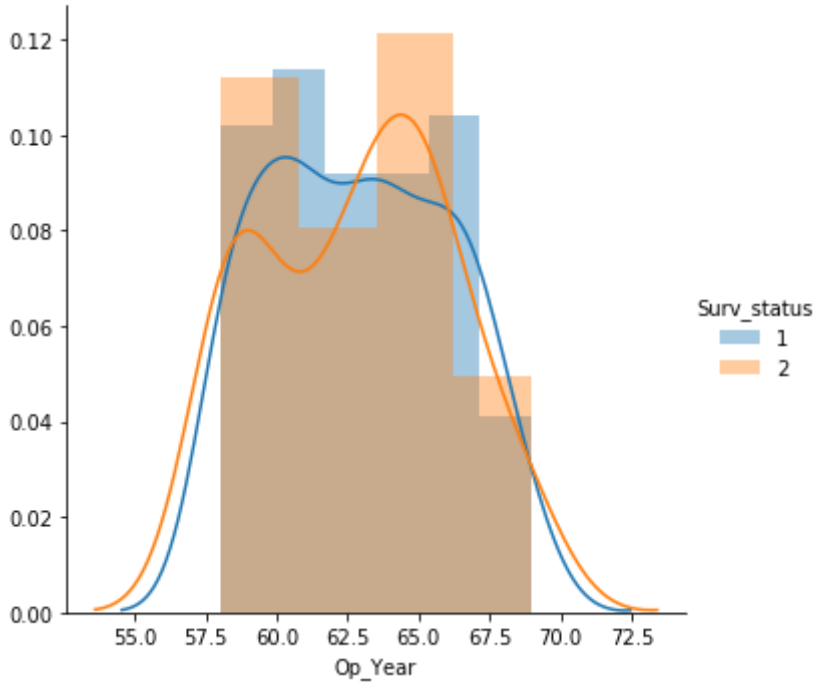
```
/home/admin1/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.
py:230: UserWarning: The `size` paramter has been renamed to `height
`; please update your code.
  warnings.warn(msg, UserWarning)
/home/admin1/anaconda3/lib/python3.7/site-packages/scipy/stats/stat
s.py:1713: FutureWarning: Using a non-tuple sequence for multidimens
ional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr
[seq]`. In the future this will be interpreted as an array index, `a
rr[np.array(seq)]`, which will result either in an error or a differ
ent result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumva
l
```



In [20]:

```
sns.FacetGrid(haberman, hue="Surv_status", size=5) \
   .map(sns.distplot, "Op_Year") \
   .add_legend();
plt.show();
```

```
/home/admin1/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.
py:230: UserWarning: The `size` paramter has been renamed to `height
`; please update your code.
  warnings.warn(msg, UserWarning)
/home/admin1/anaconda3/lib/python3.7/site-packages/scipy/stats/stat
s.py:1713: FutureWarning: Using a non-tuple sequence for multidimens
ional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr
[seq]`. In the future this will be interpreted as an array index, `a
rr[np.array(seq)]`, which will result either in an error or a differ
ent result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumva
l
```



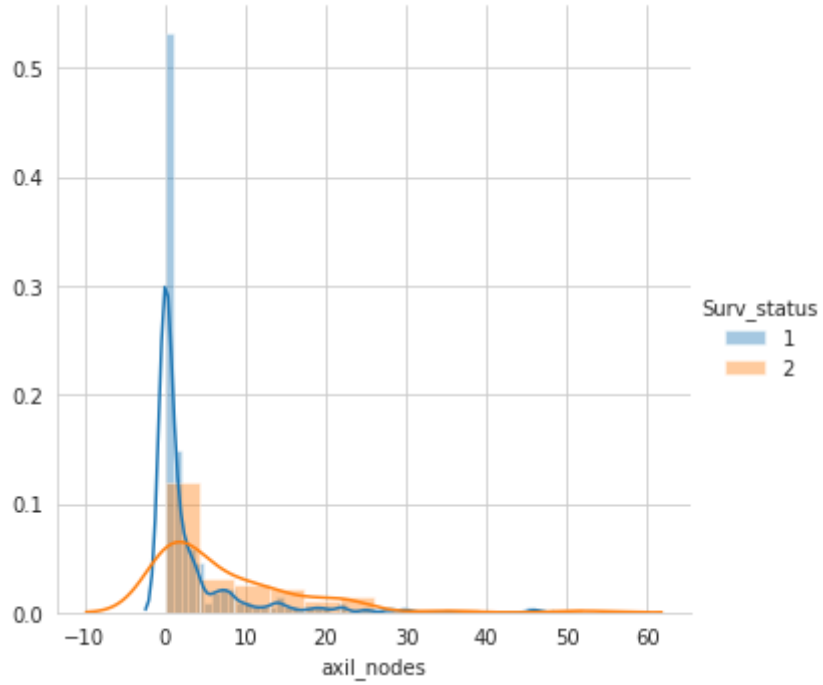
Summary

Op_year as a single feature cannot be used for classification because of the overlap

In [14]:

```
sns.FacetGrid(haberman, hue="Surv_status", size=5) \
   .map(sns.distplot, "axil_nodes") \
   .add_legend();
plt.show();
```

```
/home/admin1/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.
py:230: UserWarning: The `size` paramter has been renamed to `height
`; please update your code.
  warnings.warn(msg, UserWarning)
/home/admin1/anaconda3/lib/python3.7/site-packages/scipy/stats/stat
s.py:1713: FutureWarning: Using a non-tuple sequence for multidimens
ional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr
[seq]`. In the future this will be interpreted as an array index, `a
rr[np.array(seq)]`, which will result either in an error or a differ
ent result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumva
l
```



Summary

axil_nodes as a feature cannot be used as an independent feature for classification

PDF and CDF

In [21]:

```
haberman_survived = haberman.loc[haberman["Surv_status"] == 1];
haberman_dead = haberman.loc[haberman["Surv_status"] == 2];
```

In [31]:

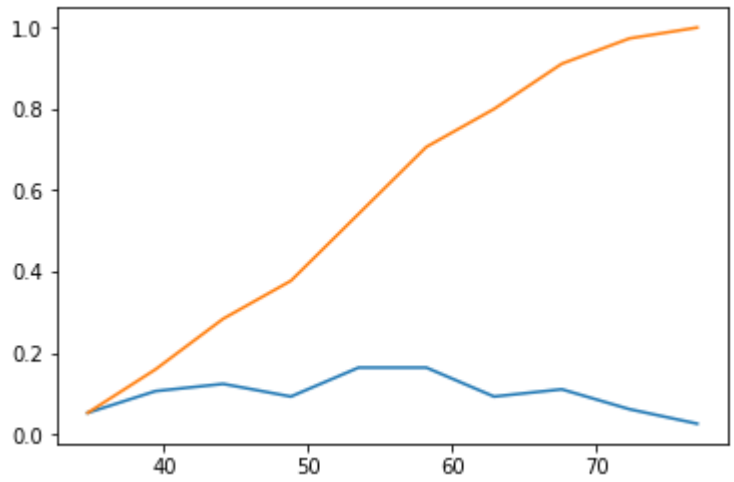
```
def plot_pdf_cdf(data):
    counts, bin_edges = np.histogram(data, bins=10,
                                     density = True)
    pdf = counts/(sum(counts))
    print(pdf);
    print(bin_edges);
    cdf = np.cumsum(pdf)
    plt.plot(bin_edges[1:],pdf);
    plt.plot(bin_edges[1:], cdf)

    plt.show()
```

In [32]:

```
plot_pdf_cdf(haberman_survived['Age'])
```

```
[0.05333333 0.10666667 0.12444444 0.09333333 0.16444444 0.16444444
 0.09333333 0.11111111 0.06222222 0.02666667]
[30.  34.7 39.4 44.1 48.8 53.5 58.2 62.9 67.6 72.3 77. ]
```



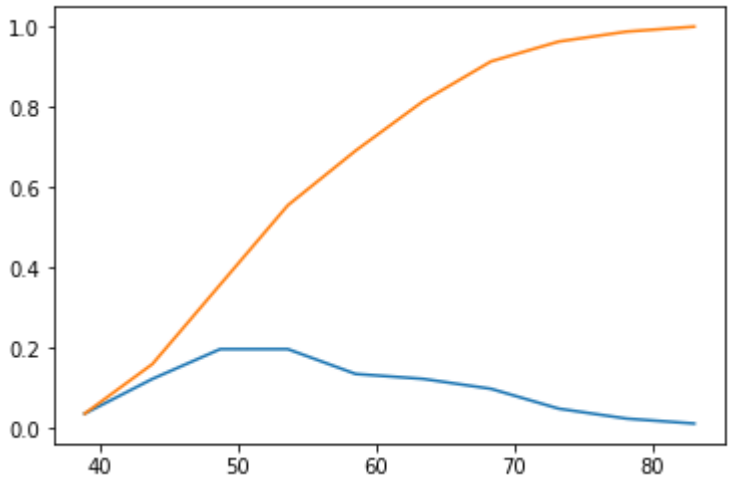
Summary

Out of all the people who have survived approximately 40% are below 50 years

In [33]:

```
plot_pdf_cdf(haberman_dead['Age'])
```

```
[0.03703704 0.12345679 0.19753086 0.19753086 0.13580247 0.12345679
 0.09876543 0.04938272 0.02469136 0.01234568]
[34.  38.9 43.8 48.7 53.6 58.5 63.4 68.3 73.2 78.1 83. ]
```



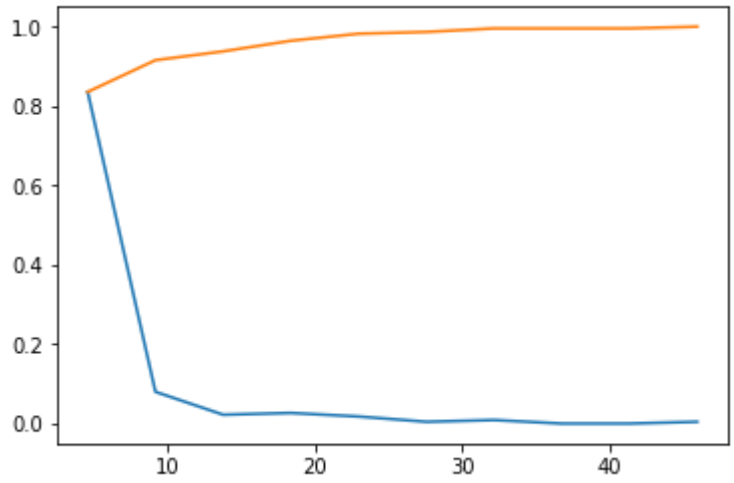
Summary

Out fo all the people whole are dead,appoximately 50% are above 55 years

In [34]:

```
plot_pdf_cdf(haberman_survived['axil_nodes'])
```

```
[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. ]
```



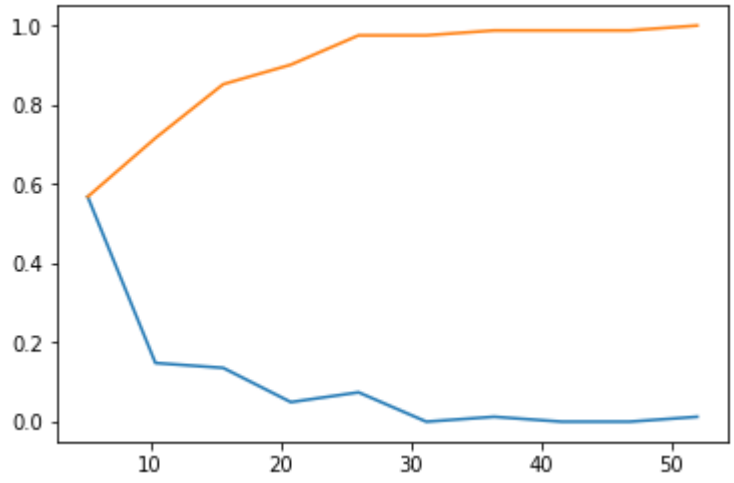
Summary

Out of all the people suvived, 90 % have axil nodes less than 10

In [35]:

```
plot_pdf_cdf(haberman_dead['axil_nodes'])
```

```
[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. ]
```



Summary

Out of all people who are dead , approximately 70 % have axil_nodes less than 10

In [24]:

```
#Mean, Variance, Std-deviation,
print("Means:")
print(np.mean(haberman["Age"]))

print(np.mean(haberman_survived["Age"]))
print(np.mean(haberman_dead["Age"]))

print("\nStd-dev:");
print(np.std(haberman_survived["Age"]))
print(np.std(haberman_dead["Age"]))
```

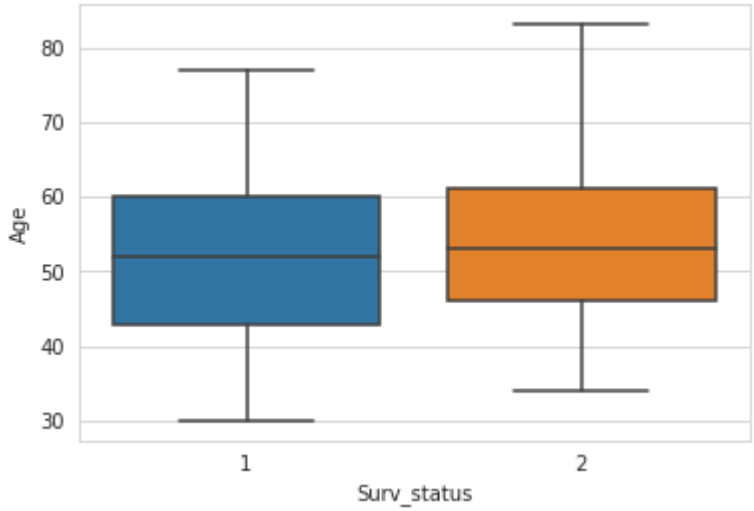
```
Means:
52.45751633986928
52.01777777777778
53.67901234567901

Std-dev:
10.98765547510051
10.10418219303131
```

Box plots

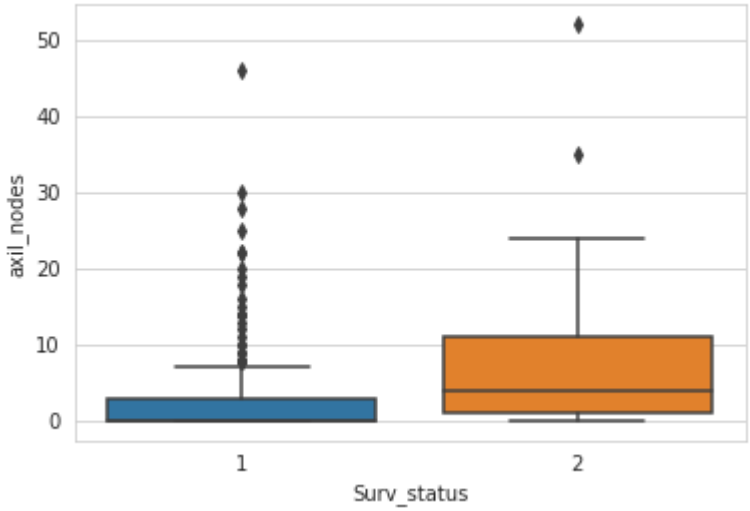
In [26]:

```
sns.boxplot(x='Surv_status',y='Age', data=haberman)
plt.show()
```



In [29]:

```
sns.boxplot(x='Surv_status',y='axil_nodes', data=haberman)
plt.show()
```



Summary

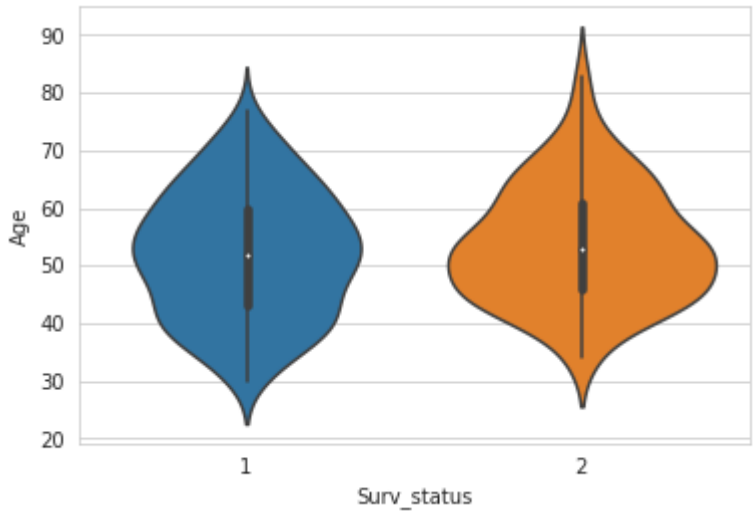
There are almost 50 % overlap between those whose are dead and alive

Violin plots

In [28]:

```
sns.violinplot(x="Surv_status", y="Age", data=haberman, size=8)
plt.show()
```

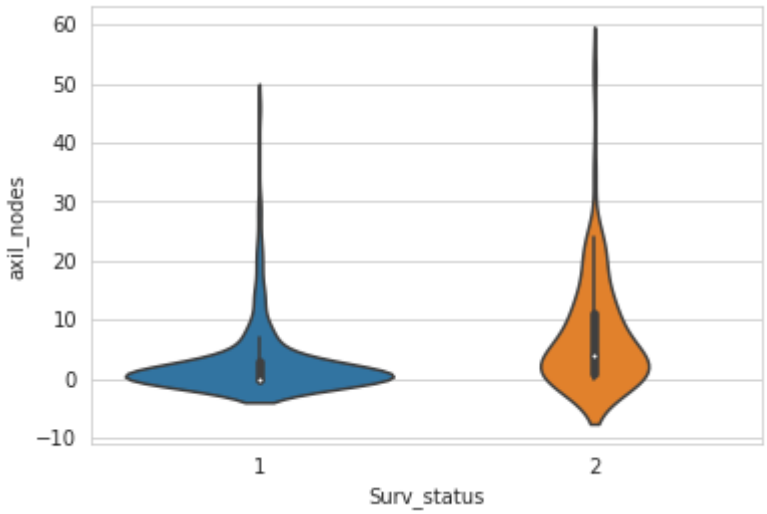
```
/home/admin1/anaconda3/lib/python3.7/site-packages/scipy/stats/stat
s.py:1713: FutureWarning: Using a non-tuple sequence for multidimens
ional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr
[seq]`. In the future this will be interpreted as an array index, `a
rr[np.array(seq)]`, which will result either in an error or a differ
ent result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumva
l
```



In [30]:

```
sns.violinplot(x="Surv_status", y="axil_nodes", data=haberman, size=8)
plt.show()
```

```
/home/admin1/anaconda3/lib/python3.7/site-packages/scipy/stats/stat
s.py:1713: FutureWarning: Using a non-tuple sequence for multidimens
ional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr
[seq]`. In the future this will be interpreted as an array index, `a
rr[np.array(seq)]`, which will result either in an error or a differ
ent result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumva
l
```



In [ ]: