EXPLORATORY DATA ANALYSIS OF HABERMAN DATASET

In [19]:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import warnings
warnings.filterwarnings("ignore")
```

In [7]:

```
ds=pd.read_csv("haberman.csv")
```

Dataset

In [11]:

```
print(ds.head(7))
```

```
   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
5  33  60   0    1
6  34  59   0    2
```

Dimensions of the data set

In [16]:

```
ds.shape
```

Out[16]:

```
(305, 4)
```

Number of rows = 305 Number of columns = 4

Columns of the dataset

In [15]:

```
ds.columns
```

Out[15]:

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

Attribute Information[1]

1. '30'= Age of patient at time of operation
2. '64'= Patient's year of operation (year - 1900, numerical)
3. '1' = Number of positive axillary nodes detected (numerical)
4. '1.1'= Survival status (class attribute) 1 = the patient survived 5 years or longer, 2 = the patient died within 5 year

In [12]:

```
ds.rename(
  columns={
    "30" : 'Age',
    "64" : 'YOP',
    "1" : 'Nodes',
    "1.1" : 'SurvivalStatus'
  },
  inplace=True
)
print(ds.head(7))
```

```
   Age  YOP  Nodes  SurvivalStatus
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
5   33   60      0               1
6   34   59      0               2
```

Number of patients of each age

In [13]:

```
ds['Age'].value_counts().head(5)
```

Out[13]:

```
52    14
54    13
50    12
43    11
47    11
Name: Age, dtype: int64
```

Maximum number of patients are 52 years old.

Number of operations in each year

In [24]:

```
ds['YOP'].value_counts().head(5)
```

Out[24]:

```
58    36
64    30
63    30
66    28
65    28
Name: YOP, dtype: int64
```

In year 1958 , maximum number of operations happened.

Description of the dataset[2]

The 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.

Objective

Our objective is to find out the survival status of the patients who have undergone breast cancer

operation. We have to analyse the various attributes like age of the patient , year of operation and number of positive axillary nodes detected and then predict the survival status.

Number of patients dying within 5 years and number of patients living for more than 5 years

In [55]:

```
ds["SurvivalStatus"].value_counts()
```

Out[55]:

```
1    224
2     81
Name: SurvivalStatus, dtype: int64
```

Thus we see 224 patients have survived for more than 5 years and 81 years survived within 5 years.

Univariate Analysis

Probability Density Function

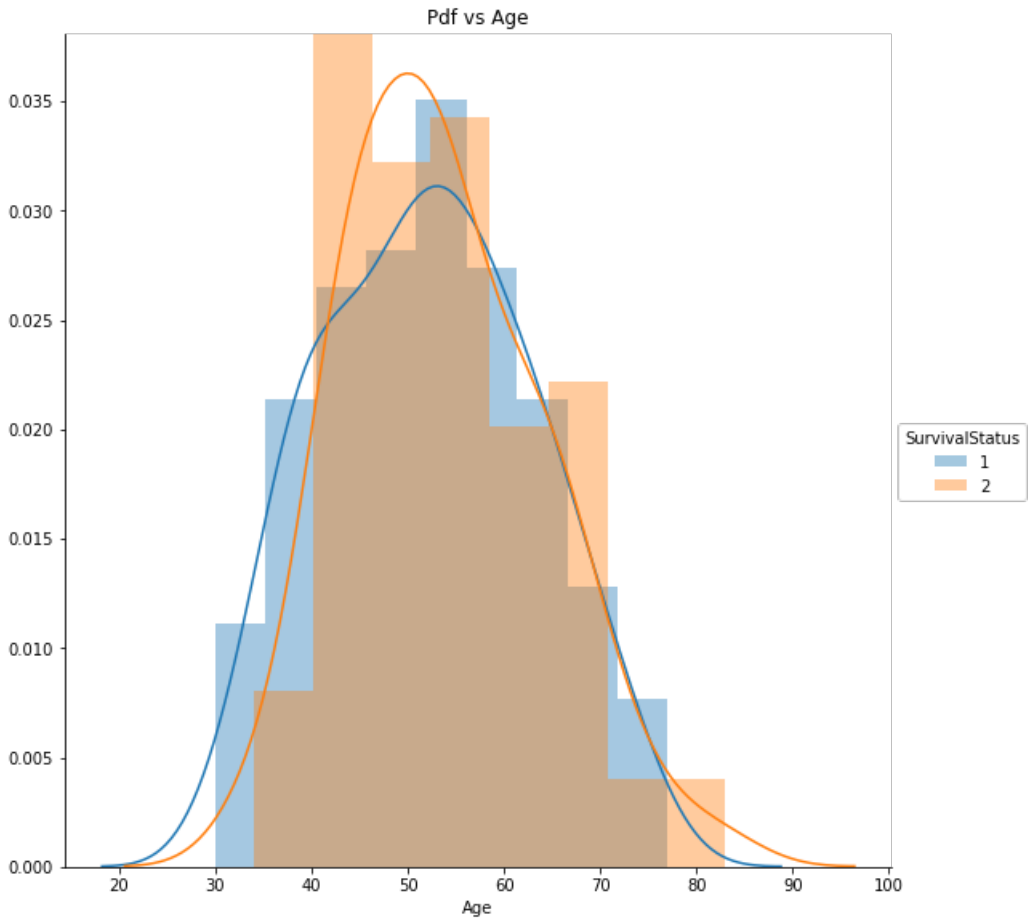
I have found the probablity density function of various age , year of operation and number of axillary nodes for each of the survival status.

In [17]:

```
sns.FacetGrid(ds,hue="SurvivalStatus",size=8).map(sns.distplot,"Age").add_legend()
plt.title("Pdf vs Age")
```

Out[17]:

```
Text(0.5,1,'Pdf vs Age')
```

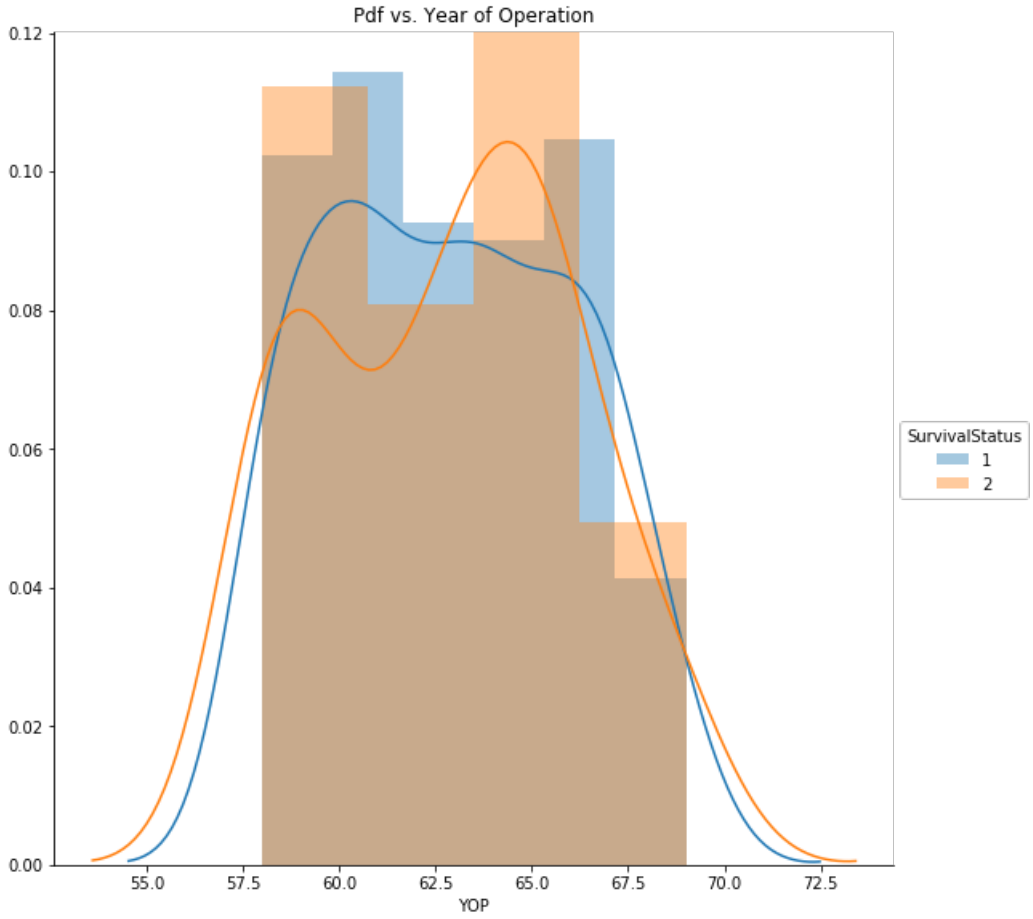


In [20]:

```
sns.FacetGrid(ds,hue="SurvivalStatus",size=8).map(sns.distplot,"YOP").add_legend()
plt.title("Pdf vs. Year of Operation")
```

Out[20]:

```
Text(0.5,1,'Pdf vs. Year of Operation')
```

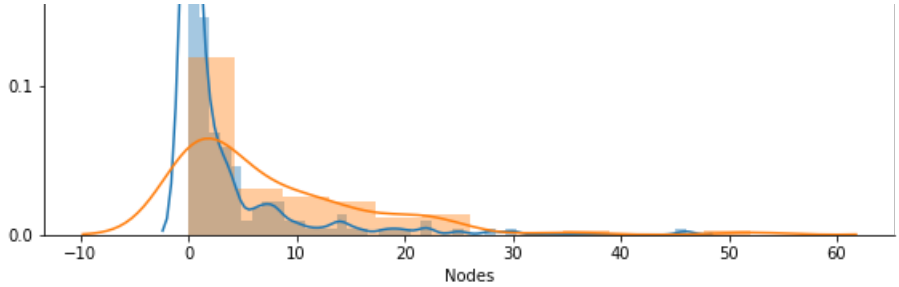


In [21]:

```
sns.FacetGrid(ds,hue="SurvivalStatus",size=8).map(sns.distplot,"Nodes").add_legend()
plt.title("Pdf vs. Nodes")
```

Out[21]:

```
Text(0.5,1,'Pdf vs. Nodes')
```



In mostly all of the graphs I see overlapping pdfs and so I cannot reach to any kind of conclusion of using any one attribute for classification.In the 3rd graph , I see that patients having less than 2 nodes have greater probability of survival status 1.

Cumulative Density Function

Now let me find the cdfs of the attributes for each of the survival status.

In [60]:

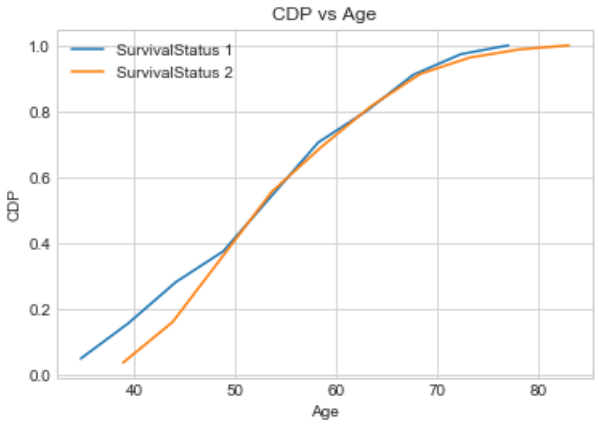
```
status1=ds.loc[ds["SurvivalStatus"]==1] ## 5 years or longer
status2=ds.loc[ds["SurvivalStatus"]==2] ## Within  5 years
```

In [77]:

```
c1,b1=np.histogram(status1["Age"],bins=10,density=True)

cdf=np.cumsum(pdf)
c2,b2=np.histogram(status2["Age"],bins=10,density=True)

cdf2=np.cumsum(pdf2)


plt.plot(b1[1:],cdf,label="SurvivalStatus 1")

plt.plot(b2[1:],cdf2,label="SurvivalStatus 2")
plt.title("CDP vs Age")
plt.xlabel("Age")
plt.ylabel("CDP")
plt.legend()
```

Out[77]:

```
<matplotlib.legend.Legend at 0x5fb025c0>
```



From this cdf I conclude that using age as the attribute I can accurately classify 10% patients as survival status of 5 years and longer and 2% patients as survival status of within 5 years.

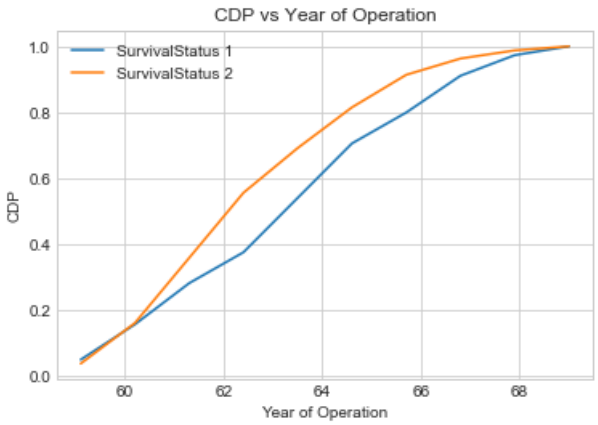
In [76]:

```
c1,b1=np.histogram(status1["YOP"],bins=10,density=True)

cdf=np.cumsum(pdf)
c2,b2=np.histogram(status2["YOP"],bins=10,density=True)
```

```
cdf2=np.cumsum(pdf2)


plt.plot(b1[1:],cdf,label="SurvivalStatus 1")

plt.plot(b2[1:],cdf2,label="SurvivalStatus 2")
plt.title("CDP vs Year of Operation")
plt.xlabel("Year of Operation")
plt.ylabel("CDP")
plt.legend()
```

Out[76]:

```
<matplotlib.legend.Legend at 0x60e02908>
```



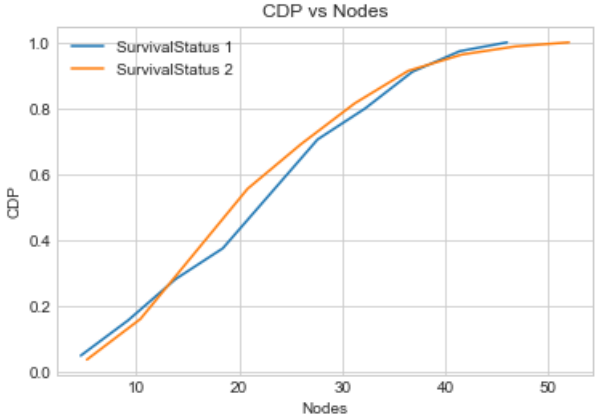
Here also we get overlapping cdfs.

In [78]:

```
c1,b1=np.histogram(status1["Nodes"],bins=10,density=True)

cdf=np.cumsum(pdf)
c2,b2=np.histogram(status2["Nodes"],bins=10,density=True)

cdf2=np.cumsum(pdf2)


plt.plot(b1[1:],cdf,label="SurvivalStatus 1")

plt.plot(b2[1:],cdf2,label="SurvivalStatus 2")
plt.title("CDP vs Nodes")
plt.xlabel("Nodes")
plt.ylabel("CDP")
plt.legend()
```

Out[78]:

```
<matplotlib.legend.Legend at 0x60d077b8>
```



Here 97% of the two cdfs are overlapping , for the last 3% patients I can classify them correctly as survival status less than 5 years

Here 81% of the two calls are overlapping , for the last 9% patients I can classify them correctly as survival status less than 5 years.

Thus I find out that using cdf , age attribute provides the best results .

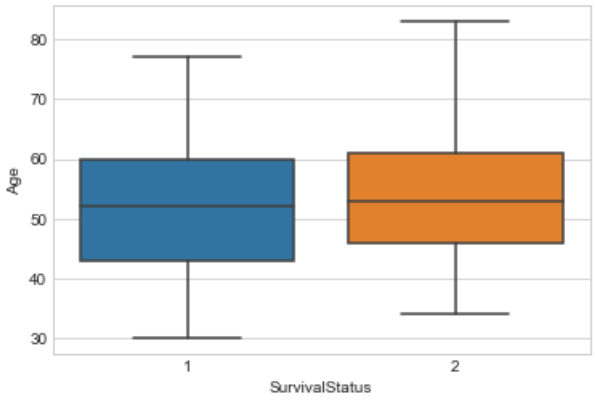
Boxplots for the 3 attributes

In [65]:

```
sns.boxplot(x="SurvivalStatus",y="Age",data=ds)
```

Out[65]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e15c4e0>
```

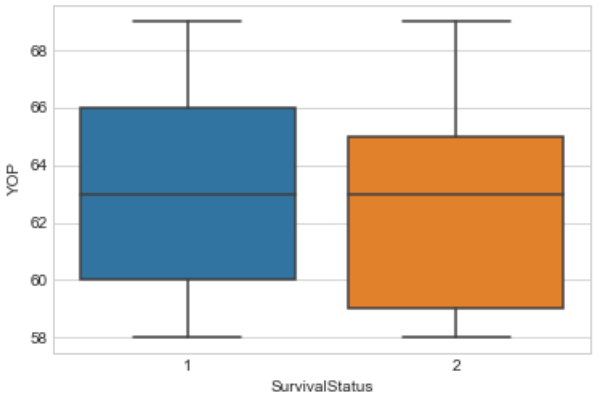


In [66]:

```
sns.boxplot(x="SurvivalStatus",y="YOP",data=ds)
```

Out[66]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e5dd160>
```

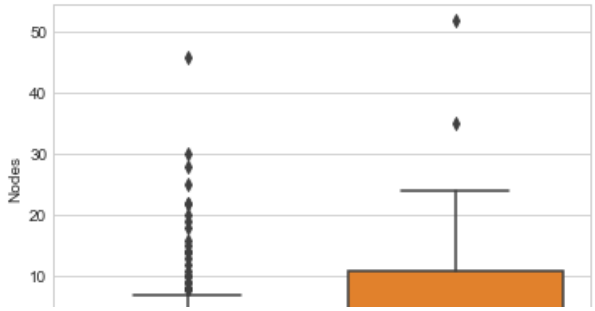


In [67]:

```
sns.boxplot(x="SurvivalStatus",y="Nodes",data=ds)
```

Out[67]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x56d78048>
```



In the age boxplot, I can classify 8 percentile of patients as living more than 5 years. In the year of operation boxplot , I can classify 8 percentile of patients as living more than 5 years and 8 percentile of patients as living less than 5 years. In the number of axillary nodes boxplot ,I can correctly classify around 50 percentile of patients as living less than 5 years.

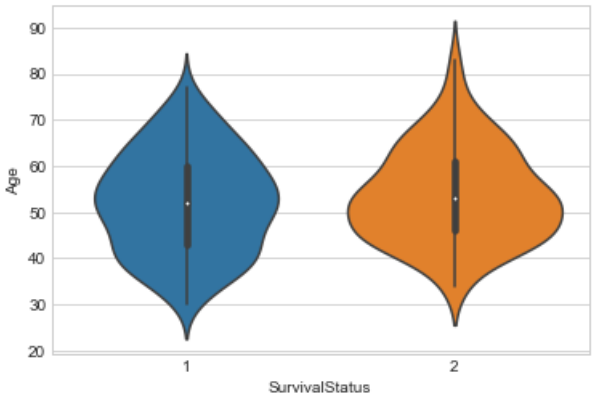
Violin Plots

In [68]:

```
sns.violinplot(y="Age",x="SurvivalStatus",data=ds)
```

Out[68]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x56de48d0>
```

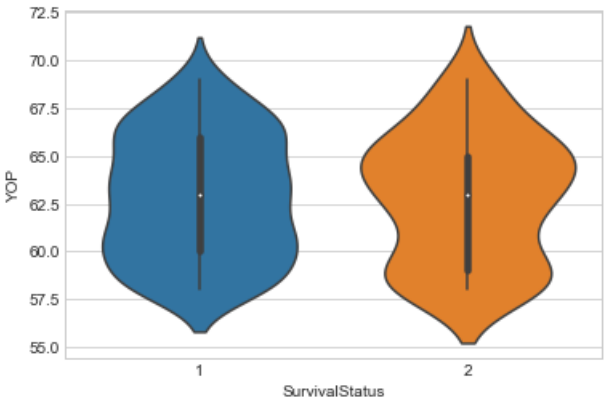


In [69]:

```
sns.violinplot(y="YOP",x="SurvivalStatus",data=ds)
```

Out[69]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x56dd5128>
```

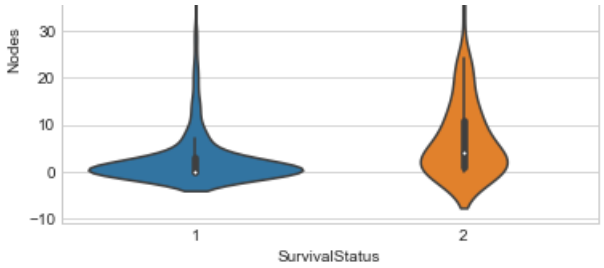


In [70]:

```
sns.violinplot(y="Nodes",x="SurvivalStatus",data=ds)
```

Out[70]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x56e3c6a0>
```



We do not infer anything significant from the violin plots.

Bivariate Analysis

Pairplots for the 3 attributes

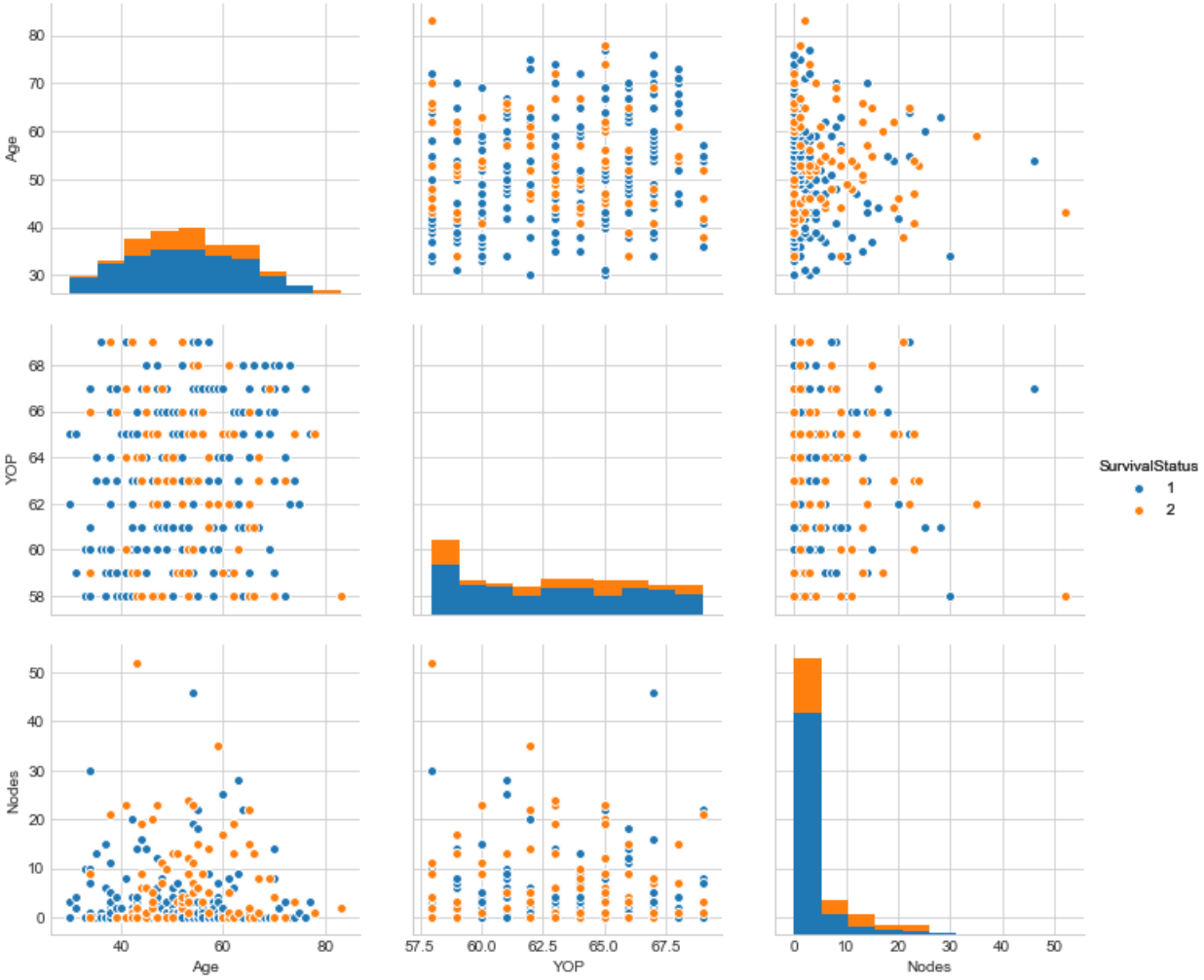
I find out the pair plots for the 3 attributes to find any relationship between them.

In [23]:

```
sns.set_style("whitegrid")
sns.pairplot(ds,hue="SurvivalStatus",size=3,vars=["Age","YOP","Nodes"]).add_legend()
```

Out[23]:

```
<seaborn.axisgrid.PairGrid at 0xb89c780>
```



I cannot infer anything signficant from the pairplots.As the scatterplots are plots from the pairplots only , I do not perform it again.

Inference

After conducting Univariate and Bivariate analysis, I conclude using the pdf and boxplot of number of axillary nodes that axillary node

After conducting Univariate and Bivariate analysis, I conclude using the pdf and boxplot of number of axillary nodes that axillary node can be used as an attribute for classification of Survival Status with an accuracy greater than the other two attributes.

References

1. https://www.kaggle.com/gilsousa/habermans-survival-data-set
2. https://www.kaggle.com/gilsousa/habermans-survival-data-set
3. https://chartio.com/resources/tutorials/how-to-rename-columns-in-the-pandas-python-library/