Exploratory Data Analysis on Haberman's dataset

Dataset :

https://www.kaggle.com/gilsousa/habermans-survival-data-set

Columns of dataset:

- age = Age of patient at time of operation (numerical)
- Op_Year = Operated year
- axil_nodes = Number of axillary nodes.
- Surv_status = 1(the patient survived 5 years or longer,2(the patient died within 5 years)

Objective :

- To analyse whether the patient will survive or not depending on age of patient,operated year,auxillary nodes.
- Independent variable : age,Op_Year,axil_nodes,Surv_status
- Dependent variable(class) : Surv_status

Loading and visualizing data

In [12]:

```
# import dataset and store it as dataframe.
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

do = pd.read_csv(r'C:\Users\Friend\AI\datasets\haberman.csv',header = None)

# Assigning feature labels to columns.
do.columns = ['age','Op_Year','axil_nodes','Surv_status']

# Converting class label from numeric to string.
label = do['Surv_status']
def renaming(x):
    if x == 1:
        return 'positive'
    else:
        return 'negative'
label = label.map(renaming)
data['Surv_status'] = label

# print data to check data
print(data.head())

# print data to check data
print(data.tail())
```

```
   age  Op_Year  axil_nodes Surv_status
0   30       64           1    positive
1   30       62           3    positive
2   30       65           0    positive
3   31       59           2    positive
4   31       65           4    positive
     age  Op_Year  axil_nodes Surv_status
301   75       62           1    positive
302   76       67           0    positive
303   77       65           3    positive
304   78       65           1    negative
305   83       58           2    negative
positive    225
negative     81
```

```
negative     81
Name: Surv_status, dtype: int64
```

High level statistics of dataset

In [13]:

```
# Number of points and features
print('number of data points are {} \nnumber of features are {}'.format(data.shape[0],data.shape[1]))
```

```
number of data points are 306
number of features are 4
```

In [14]:

```
# Names of features(column names)
print(data.columns)
```

```
Index(['age', 'Op_Year', 'axil_nodes', 'Surv_status'], dtype='object')
```

In [15]:

```
# Data-Points per class
print(data['Surv_status'].value_counts())
```

```
positive    225
negative     81
Name: Surv_status, dtype: int64
```

In [17]:

```
# Is dataset balanced/imbalaced?
gl = data.groupby('Surv_status')
gl['Surv_status'].count()
```

Out[17]:

```
Surv_status
negative     81
positive    225
Name: Surv_status, dtype: int64
```

Observations:

- Since there is no uniformity in class,this dataset could be considered as imbalanced data set. Typically the

In [19]:

```
# Survival Status occured in each year
operated_years = data.groupby('Op_Year')
for operated_year,Status in operated_years:
    print(operated_year)
    print(Status['Surv_status'].value_counts())
```

```
58
positive    24
negative    12
Name: Surv_status, dtype: int64
59
positive    18
negative     9
Name: Surv_status, dtype: int64
60
positive    24
negative     4
Name: Surv_status, dtype: int64
61
```

```
positive    23
negative     3
Name: Surv_status, dtype: int64
62
positive    16
negative     7
Name: Surv_status, dtype: int64
63
positive    22
negative     8
Name: Surv_status, dtype: int64
64
positive    23
negative     8
Name: Surv_status, dtype: int64
65
positive    15
negative    13
Name: Surv_status, dtype: int64
66
positive    22
negative     6
Name: Surv_status, dtype: int64
67
positive    21
negative     4
Name: Surv_status, dtype: int64
68
positive    10
negative     3
Name: Surv_status, dtype: int64
69
positive    7
negative    4
Name: Surv_status, dtype: int64
```

Observations:

- In 1958 , we could see the highest number(24) of survivals.
- In 1965 , we could see the lowest number of deaths(13).

In [57]:

```
# compute statistical values
print(data.describe())
```

```
              age     Op_Year  axil_nodes  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:

- Total number of patients that underwent treatment is 306.
- Age of patients vary from 30 to 83
- This dataset corresponds to patients treated from the year 1958 to 1969.
- There are patients who doesnt have axil nodes.Perhaps 75% of the patients had max of 4 lymph nodes.

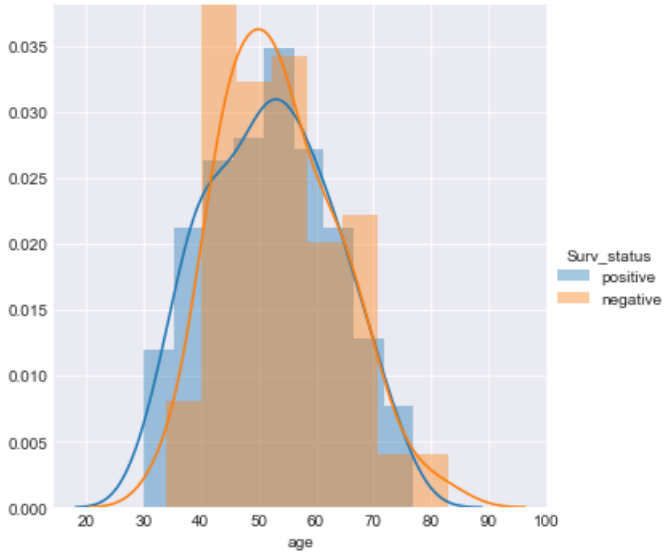
Univariate Analysis

PDF

In [24]:

```
# PDF plot taking 'age' as variable.
sns.FacetGrid(data,hue = 'Surv_status',size = 5).map(sns.distplot,'age').add_legend()
sns.set_style('darkgrid')
plt.show()
```

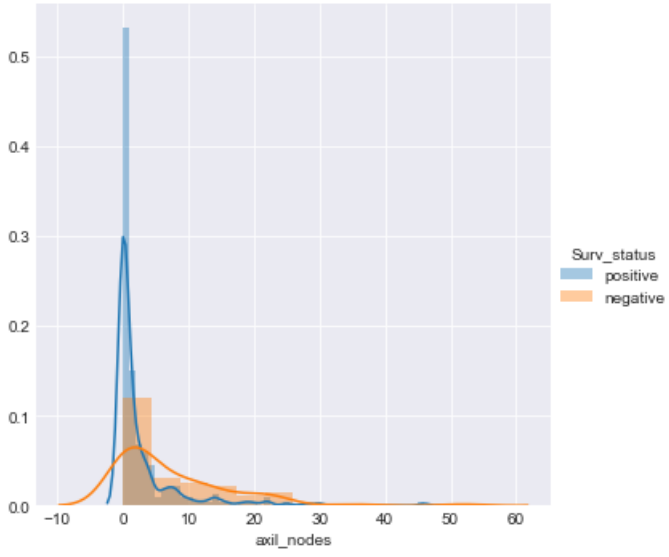
```
C:\Users\Friend\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kw
arg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\Users\Friend\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: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



In [23]:

```
# PDF plot taking 'axil_nodes' as variable.
sns.FacetGrid(data,hue = 'Surv_status',size = 5).map(sns.distplot,'axil_nodes').add_legend()
sns.set_style('darkgrid')
plt.show()
```

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

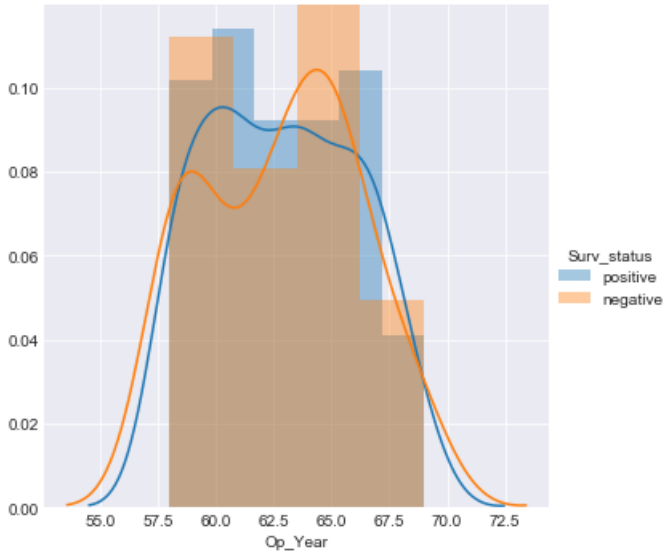


In [25]:

```
# PDF plot taking 'Op_Year' as variable
```

```
# PDF plot taking "Op_Year" as variable.
sns.FacetGrid(data,hue = 'Surv_status',size = 5).map(sns.distplot,'Op_Year').add_legend()
sns.set_style('darkgrid')
plt.show()
```

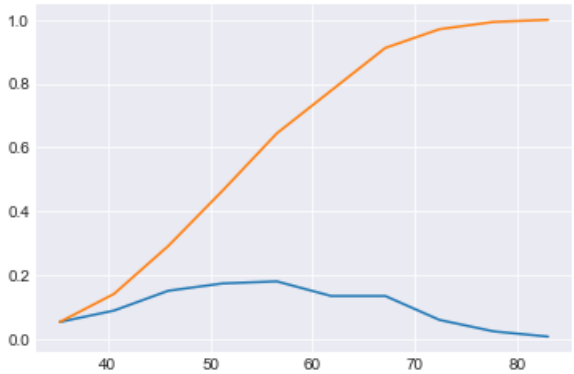
```
C:\Users\Friend\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kw
arg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\Users\Friend\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: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



CDF

In [26]:

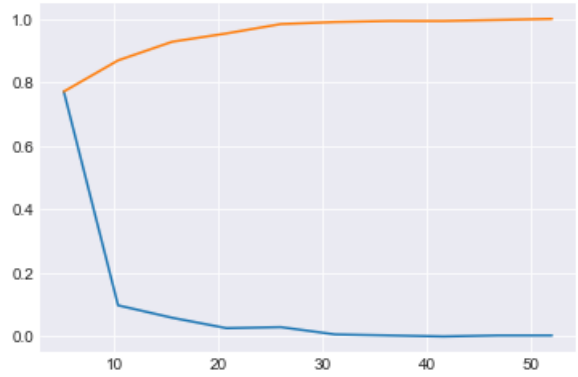
```
# calculate mass and bar width ranges.
count,bar_width = np.histogram(data['age'],bins = 10,density = True)

# plot pdf and cdf
pdf = count/(sum(count))
plt.plot(bar_width[1:],pdf)

# plot CDF
cdf = np.cumsum(pdf)
plt.plot(bar_width[1:],cdf)

plt.show()
```



In [27]:

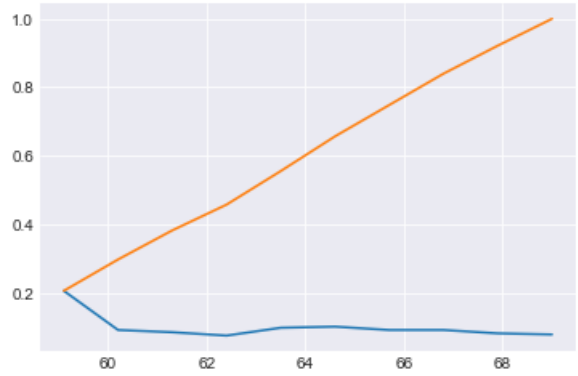
```
# calculate mass and bar width ranges.
count,bar_width = np.histogram(data['axil_nodes'],bins = 10,density = True)

# plot pdf
```

```
pdf = count/(sum(count))
plt.plot(bar_width[1:],pdf)

# plot CDF
cdf = np.cumsum(pdf)
plt.plot(bar_width[1:],cdf)

plt.show()
```



In [28]:

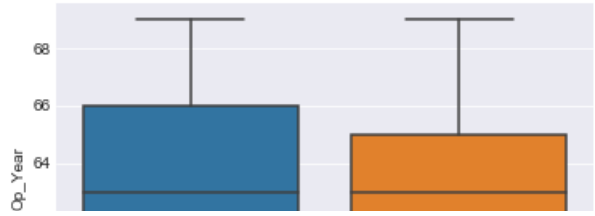
```
# calculate mass and bar width ranges.
count,bar_width = np.histogram(data['Op_Year'],bins = 10,density = True)

# plot pdf
pdf = count/(sum(count))
plt.plot(bar_width[1:],pdf)

# plot CDF
cdf = np.cumsum(pdf)
plt.plot(bar_width[1:],cdf)

plt.show()
```

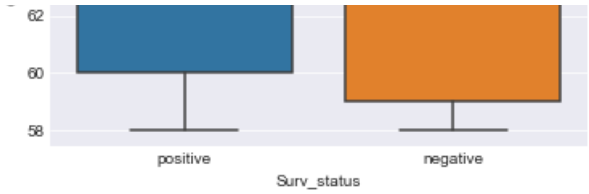


Box plot

In [29]:

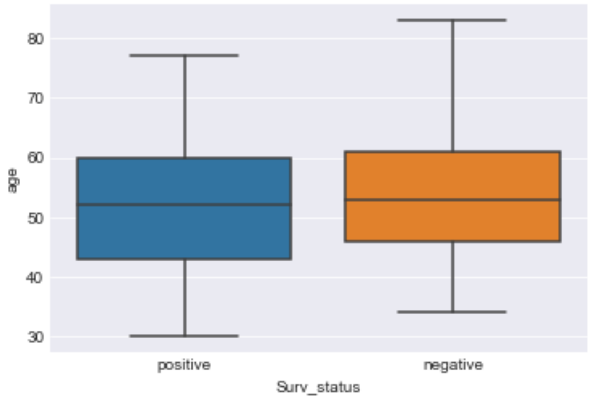
```
sns.boxplot(x = 'Surv_status',y = 'Op_Year',data = data)
sns.set_style('darkgrid')
plt.show()
```





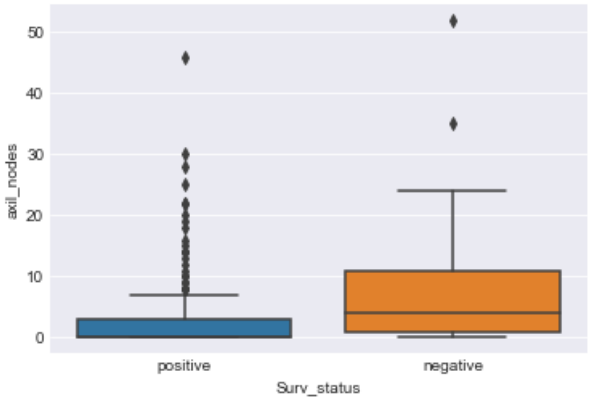
In [30]:

```
sns.boxplot(x = 'Surv_status',y = 'age',data = data)
sns.set_style('darkgrid')
plt.show()
```



In [31]:

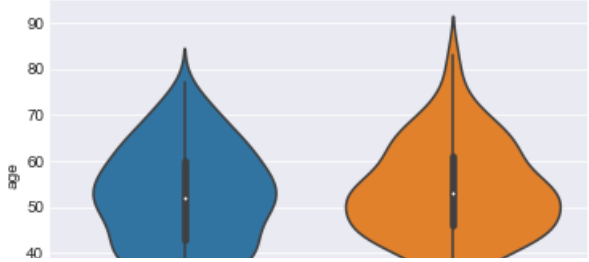
```
sns.boxplot(x = 'Surv_status',y = 'axil_nodes',data = data)
sns.set_style('darkgrid')
plt.show()
```



Violin plot

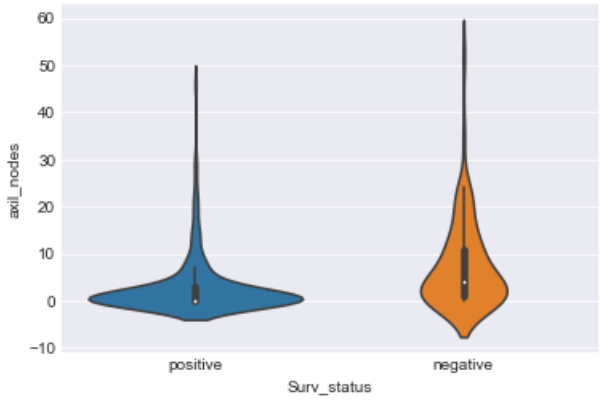
In [32]:

```
sns.violinplot(x ='Surv_status' ,y = 'age',data = data)
sns.set_style('darkgrid')
plt.show()
```



In [33]:

```
sns.violinplot(x ='Surv_status' ,y = 'axil_nodes',data = data)
sns.set_style('darkgrid')
plt.show()
```



In [40]:

```
sns.violinplot(x ='Surv_status' ,y = 'Op_Year',data = data)
sns.set_style('darkgrid')
plt.show()
```

Observations of Univariate Analysis:

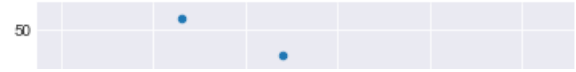
- Patients who have beentreated after 1966 has high chances of survival.
- 75% of patients who have survived more than 5 years have less than around 4 lymph nodes

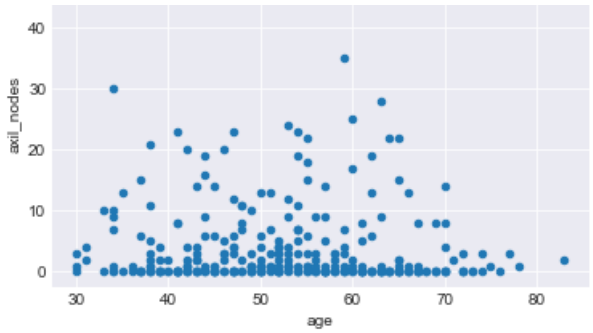
Bivariate analysis

2D-Scatter Plot

In [35]:

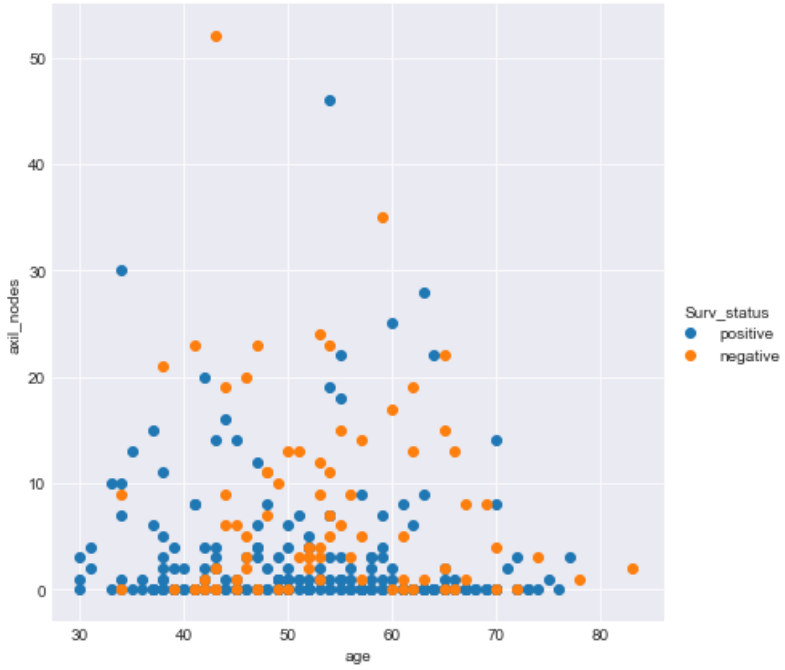
```
data.plot(kind = 'scatter',x ='age',y = 'axil_nodes')
plt.show()
```





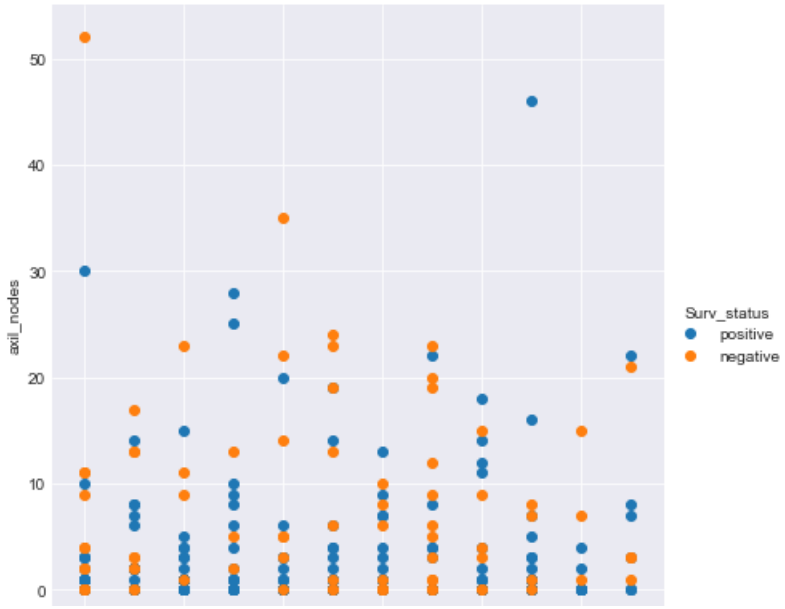
In [36]:

```
sns.FacetGrid(data,hue = 'Surv_status',size = 6).map(plt.scatter,'age','axil_nodes').add_legend()
sns.set_style('darkgrid')
plt.show()
```



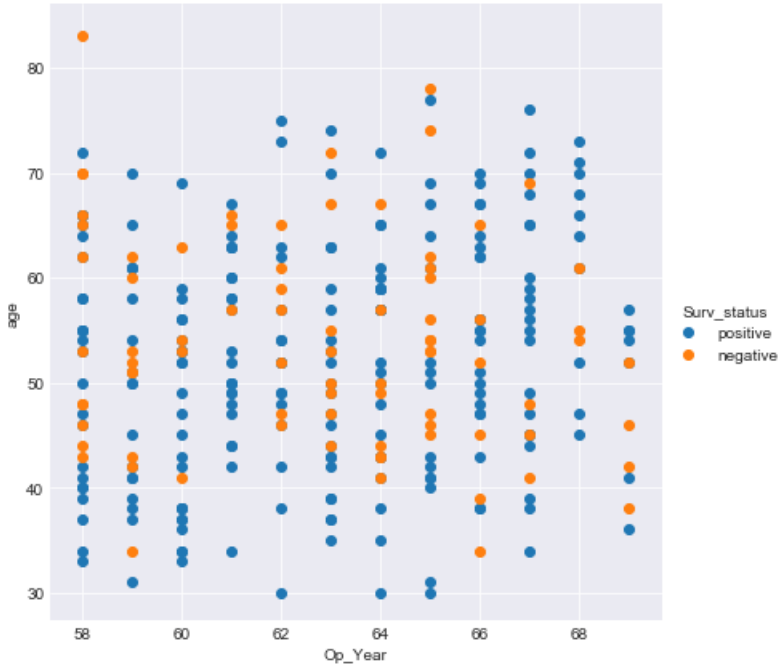
In [37]:

```
sns.FacetGrid(data,hue = 'Surv_status',size = 6).map(plt.scatter,'Op_Year','axil_nodes').add_legend()
sns.set_style('darkgrid')
plt.show()
```



In [38]:

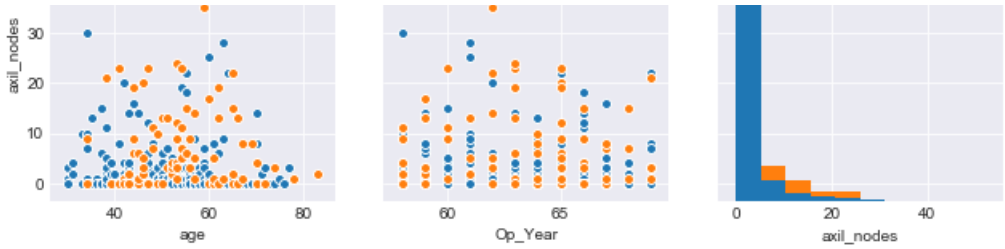
```
sns.FacetGrid(data,hue = 'Surv_status',size = 6).map(plt.scatter,'Op_Year','age').add_legend()
sns.set_style('darkgrid')
plt.show()
```



Pair plot

In [39]:

```
sns.pairplot(data,hue = 'Surv_status',size = 3)
sns.set_style('darkgrid')
plt.show()
```



Observations of Bivariate Analysis:

- By scatter plot drawn across all the features of dataset,we can see that the plot output does not provide us any result.
- None of the features distinguishes the class label.
- Data has thoroughly scattered across all the plot,and hence a distinct analysis could not be made using pair plot.