Analysis of Haberman's Survival Dataset

- Objective: Classify a patient died(whithin 5yrs.) or survived(more than 5yrs.) given the 3 features age, op_year, axil_node.
- axillary node(axil_node) : https://www.healthline.com/human-body-maps/axillary-lymph-nodes
- op_year : The year in which operation of patient was performed
- age : It is the age of the patient.

1. Importing Libraries

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
from statsmodels import robust
```

In [2]:

```
warnings.filterwarnings('ignore')
```

2. Load Dataset

In [3]:

```
data = pd.read_csv('haberman.csv', names = ['age', 'op_year', 'axil_nodes', 'surv_status'])
data.head()
```

Out[3]:

	age	op_year	axil_nodes	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

```
There are 3 independent variables : age, operation year(op_year), axillary node(axil_node)
1 dependent variable survival status(surv_status)
```

Replace 1 with survived and 2 with died

surv_status 1 means patient survive 5 year or more and 2 means patient died within 5 year

In [4]:

```
data.surv_status.replace(to_replace = [1, 2], value = ['survived', 'died'], inplace = True)
```

Brief summary of data set

In [5]:

```
# Columns in the data set
data.columns
```

Out[5]:

```
Index(['age', 'op_year', 'axil_nodes', 'surv_status'], dtype='object')
```

In [6]:

```
# Shape of data set
data.shape
```

Out[6]:

```
(306, 4)
```

```
    Rows : 306
    Columns : 4
```

In [7]:

```
data.describe()
```

Out[7]:

	age	op_year	axil_nodes
count	306.000000	306.000000	306.000000
mean	52.457516	62.852941	4.026144
std	10.803452	3.249405	7.189654
min	30.000000	58.000000	0.000000
25%	44.000000	60.000000	0.000000
50%	52.000000	63.000000	1.000000
75%	60.750000	65.750000	4.000000
max	83.000000	69.000000	52.000000

In [8]:

```
# There are 2 Classes in survival_status column 1. survived and 2. died
data.surv_status.value_counts()
```

Out[8]:

```
survived    225
died         81
Name: surv_status, dtype: int64
```

Observation :

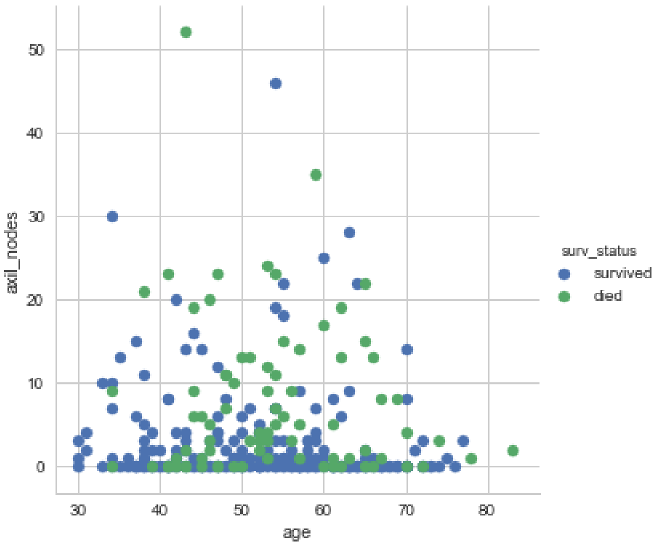
This dataset is imbalanced because the patient survived(more than 5 year) is more than the patient died(within 5 years).

Bivariate Analysis

1. Scatter Plot

In [9]:

```
# Plot between the age and axil_nodes
sns.set(style = 'whitegrid')
grid = sns.FacetGrid(data, hue = 'surv_status', size = 5)
grid.map(plt.scatter, 'age', 'axil_nodes')
grid.add_legend()
plt.show()
```



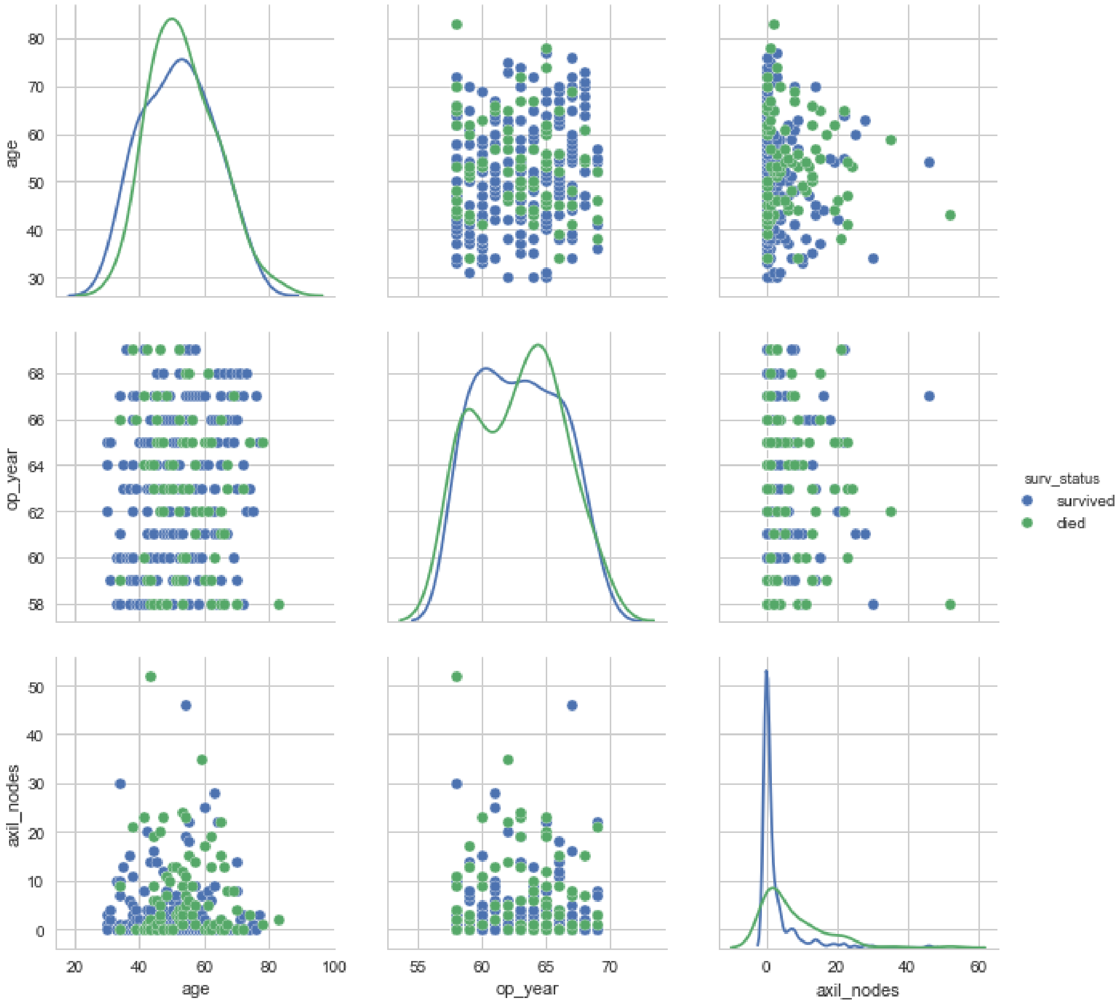
Observations :

1. The points are highly overlapped
2. We cannot easily classify by simply looking at the scatter plot between age and axile node

2. Pair Plot

In [10]:

```
# There are 3C2(= 3) combinations to select x and y axis
sns.pairplot(data, hue = 'surv_status', size = 3, diag_kind = 'kde')
plt.show()
```



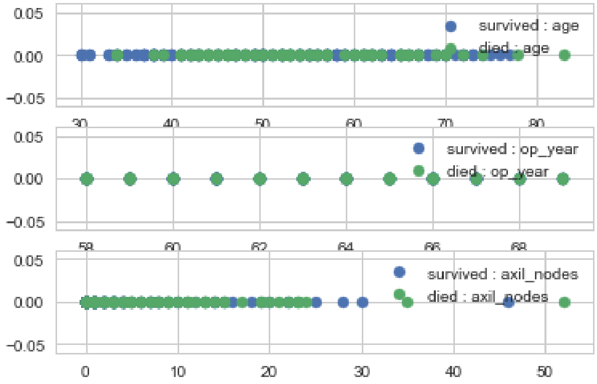
Observations :

1. It is difficult to do classification by simply looking at the scatter plots beacuse the points are highly overlapping.
2. op_year cannot be used to do classification because if we look at the plots it just shows how many patient had admited in the hospiltal. It has no relation to surv_status(whether the pateint died or survived).

Univariate Analysis

In [11]:

```
# 1D plot
survived = data[data.surv_status == 'survived']
died = data[data.surv_status == 'died']

sns.set(style = 'whitegrid')

# age
plt.subplot(311)
plt.scatter(survived.age, np.zeros_like(survived.age), label = 'survived : age')
plt.scatter(died.age, np.zeros_like(died.age), label = 'died : age')
plt.legend()

# op_year
plt.subplot(312)
plt.scatter(survived.op_year, np.zeros_like(survived.op_year), label = 'survived : op_year')
plt.scatter(died.op_year, np.zeros_like(died.op_year), label = 'died : op_year')
plt.legend()

# axil_nodes
plt.subplot(313)
plt.scatter(survived.axil_nodes, np.zeros_like(survived.axil_nodes), label = 'survived :
axil_nodes')
plt.scatter(died.axil_nodes, np.zeros_like(died.axil_nodes), label = 'died : axil_nodes')
plt.legend()

plt.show()
```



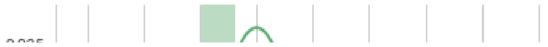
Observations :

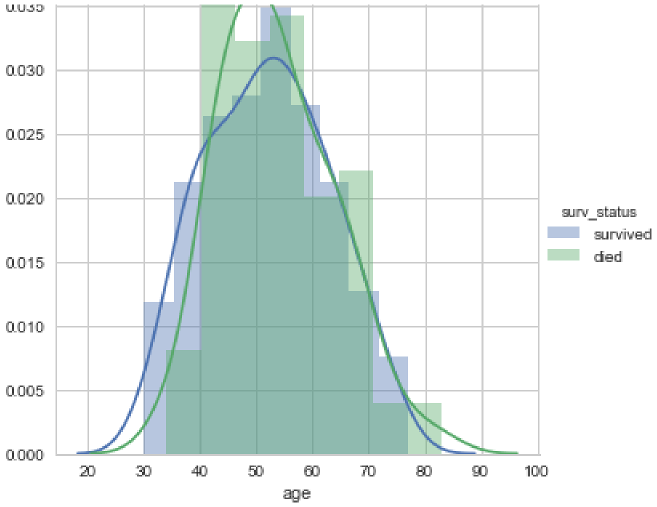
1. By looking the 1D plot of op_year it become clear that op_year has nothing to do with classification of survival of patients
2. But still we cannot do classification based on age and axil_nodes because there is very large overlaping of points

Histograms

In [12]:

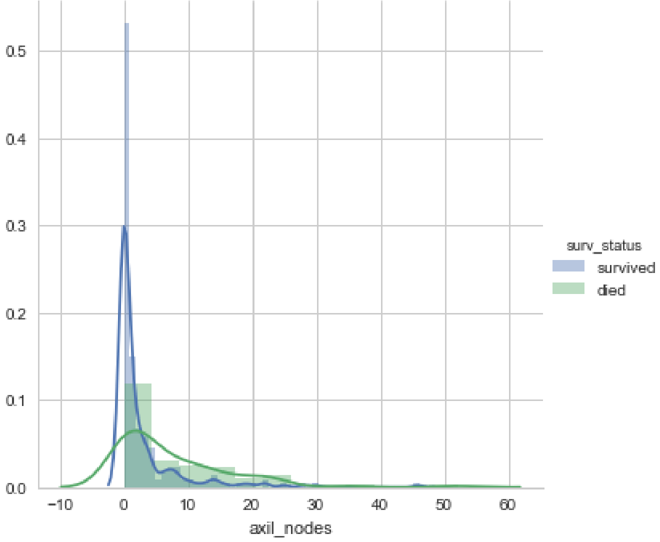
```
# age
grid = sns.FacetGrid(data, hue = 'surv_status', size = 5)
grid.map(sns.distplot, 'age')
grid.add_legend()
plt.show()
```





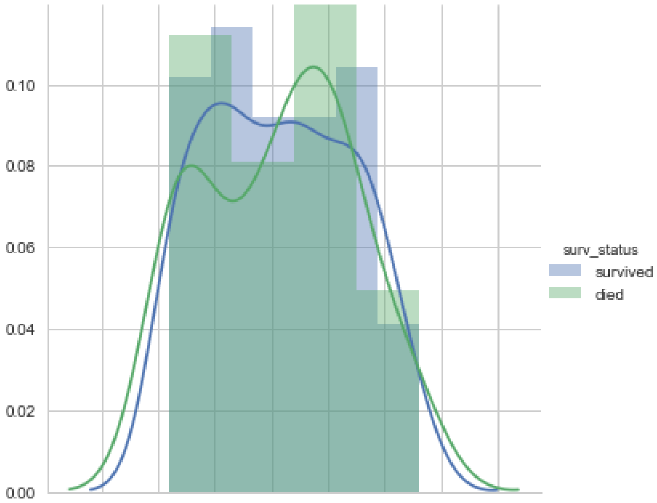
In [13]:

```
# axil_nodes
grid = sns.FacetGrid(data, hue = 'surv_status', size = 5)
grid.map(sns.distplot, 'axil_nodes')
grid.add_legend()
plt.show()
```



In [14]:

```
# op_year
grid = sns.FacetGrid(data, hue = 'surv_status', size = 5)
grid.map(sns.distplot, 'op_year')
grid.add_legend()
plt.show()
```



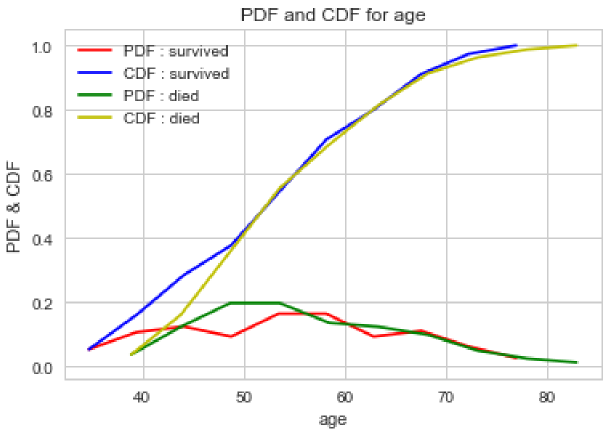
Observation:

1. There is very large overlapping in the histogram of age.
2. If we look at the histogram of axile_nodes then we can atleast say that the most patient survived who had less axil_nodes(about 0 - 2) and most patient died who had large number of axil_nodes(3 onwards)

PDF and CDF

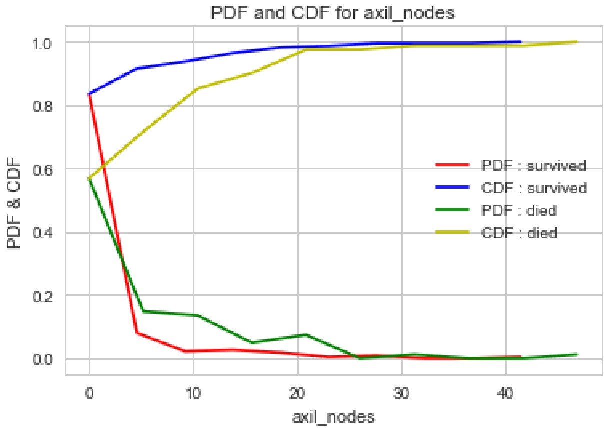
In [15]:

```
# For age

# survived
counts, bins = np.histogram(survived.age, bins = 10, density = False)
pdf = counts / counts.sum()
cdf = np.cumsum(pdf)
plt.plot(bins[1:], pdf, 'r', label = 'PDF : survived')
plt.plot(bins[1:], cdf, 'b', label = 'CDF : survived')

# died
counts, bins = np.histogram(died.age, bins = 10, density = False)
pdf = counts / counts.sum()
cdf = np.cumsum(pdf)
plt.plot(bins[1:], pdf, 'g', label = 'PDF : died')
plt.plot(bins[1:], cdf, 'y', label = 'CDF : died')

plt.legend()
plt.title('PDF and CDF for age')
plt.xlabel('age')
plt.ylabel('PDF & CDF')
plt.show()
```



In [16]:

```
# For axil_nodes

# survived
counts, bins = np.histogram(survived.axil_nodes, bins = 10, density = False)
pdf = counts / counts.sum()
cdf = np.cumsum(pdf)
plt.plot(bins[:-1], pdf, 'r', label = 'PDF : survived')
plt.plot(bins[:-1], cdf, 'b', label = 'CDF : survived')

# died
counts, bins = np.histogram(died.axil_nodes, bins = 10, density = False)
pdf = counts / counts.sum()
cdf = np.cumsum(pdf)
plt.plot(bins[:-1], pdf, 'g', label = 'PDF : died')
plt.plot(bins[:-1], cdf, 'y', label = 'CDF : died')

plt.legend()
plt.title('PDF and CDF for axil_nodes')
plt.xlabel('axil_nodes')
plt.ylabel('PDF & CDF')
plt.show()
```



Observation:

1. If we look at the cdf of age then we can say that age has barely any affects on the surv_status of a patient.
2. In case of axiel node, 90% of patient who survived and 70% of patient who died had axil_nodes less than 4. As axil_nodes exceeds 4 the probability of paitient had died is more.

Mean and Variance / Standard deviation

1. Patient who had survived(more than 5yrs.)

In [17]:

```
print('Mean : ')
print('age : ', survived.age.mean())
print('axil_nodes : ', survived.axil_nodes.mean())
print('op_year : ', survived.op_year.mean())
print('*' * 30)
print('Standard Deviation : ')
print('age : ', survived.age.std())
print('axil_nodes : ', survived.axil_nodes.std())
print('op_year : ', survived.op_year.std())
```

```
Mean :
age :  52.01777777777778
axil_nodes :  2.7911111111111113
op_year :  62.86222222222222
******************************
Standard Deviation :
age :  11.012154179929546
axil_nodes :  5.870318127719728
op_year :  3.222915223781498
```

2. Patient who had died(whithin 5yrs.)

In [18]:

```
print('Mean : ')
print('age : ', died.age.mean())
print('axil_nodes : ', died.axil_nodes.mean())
print('op_year : ', died.op_year.mean())
print('*' * 30)
print('Standard Deviation : ')
print('age : ', died.age.std())
print('axil_nodes : ', died.axil_nodes.std())
print('op_year : ', died.op_year.std())
```

```
Mean :
age :  53.67901234567901
axil_nodes :  7.45679012345679
```

```
op_year :  62.82716049382716
******************************
Standard Deviation :
age :  10.16713720829741
axil_nodes :  9.185653736555782
op_year :  3.34211805393223
```

Median, Percentile, Quantile, Inter Qunatile Range(IQR), Meadin Absolute Deviation(MAD)

1. Patient who had survived(more than 5yrs.)

In [19]:

```
print('Median : ')
print('age : ', survived.age.median())
print('axil_nodes : ', survived.axil_nodes.median())
print('op_year : ', survived.op_year.median())
print('*' * 30)
print('90th percentile : ')
print('age : ', np.percentile(survived.age, 90))
print('axil_nodes : ', np.percentile(survived.axil_nodes, 90))
print('op_year : ', np.percentile(survived.op_year, 90))
print('*' * 30)
print('Qunatile : ')
print('age : ', np.percentile(survived.age, np.arange(25, 101, 25)))
print('axil_nodes : ', np.percentile(survived.axil_nodes, np.arange(25, 101, 25)))
print('op_year : ', np.percentile(survived.op_year, np.arange(25, 101, 25)))
print('*' * 30)
print('Inter Qunatile Range : ')
print('age : ', np.percentile(survived.age, 75) - np.percentile(survived.age, 25))
print('axil_nodes : ', np.percentile(survived.axil_nodes, 75) - np.percentile(survived.axil_nodes,
25))
print('op_year : ', np.percentile(survived.op_year, 25) - np.percentile(survived.op_year, 25))
print('*' * 30)
print('Meadin Absolute Deviation : ')
print('age : ', robust.mad(survived.age))
print('axil_nodes : ', robust.mad(survived.axil_nodes))
print('op_year : ', robust.mad(survived.op_year))
```

```
Median :
age :  52.0
axil_nodes :  0.0
op_year :  63.0
******************************
90th percentile :
age :  67.0
axil_nodes :  8.0
op_year :  67.0
******************************
Qunatile :
age :  [43. 52. 60. 77.]
axil_nodes :  [ 0.  0.  3. 46.]
op_year :  [60. 63. 66. 69.]
******************************
Inter Qunatile Range :
age :  17.0
axil_nodes :  3.0
op_year :  0.0
******************************
Meadin Absolute Deviation :
age :  13.343419966550417
axil_nodes :  0.0
op_year :  4.447806655516806
```

2. Patient who had died(within 5yrs.)

In [20]:

```
print('Median : ')
print('age : ', died.age.median())
print('axil_nodes : ', died.axil_nodes.median())
print('op_year : ', died.op_year.median())
print('op_year : ', died.op_year.median())
print('*' * 30)
print('90th percentile : ')
print('age : ', np.percentile(died.age, 90))
print('axil_nodes : ', np.percentile(died.axil_nodes, 90))
print('op_year : ', np.percentile(died.op_year, 90))
print('*' * 30)
print('Qunatile : ')
print('age : ', np.percentile(died.age, np.arange(25, 101, 25)))
print('axil_nodes : ', np.percentile(died.axil_nodes, np.arange(25, 101, 25)))
print('op_year : ', np.percentile(died.op_year, np.arange(25, 101, 25)))
print('*' * 30)
print('Inter Qunatile Range : ')
print('age : ', np.percentile(died.age, 75) - np.percentile(died.age, 25))
print('axil_nodes : ', np.percentile(died.axil_nodes, 75) - np.percentile(died.axil_nodes, 25))
print('op_year : ', np.percentile(died.op_year, 25) - np.percentile(died.op_year, 25))
print('*' * 30)
print('Meadin Absolute Deviation : ')
print('age : ', robust.mad(died.age))
print('axil_nodes : ', robust.mad(died.axil_nodes))
print('op_year : ', robust.mad(died.op_year))
```

```
Median :
age :  53.0
axil_nodes :  4.0
op_year :  63.0
******************************
90th percentile :
age :  67.0
axil_nodes :  20.0
op_year :  67.0
******************************
Qunatile :
age :  [46. 53. 61. 83.]
axil_nodes :  [ 1.  4. 11. 52.]
op_year :  [59. 63. 65. 69.]
******************************
Inter Qunatile Range :
age :  15.0
axil_nodes :  10.0
op_year :  0.0
******************************
Meadin Absolute Deviation :
age :  11.860817748044816
axil_nodes :  5.930408874022408
op_year :  4.447806655516806
```

Observations:

1. 50% patient who survived had 0 axil_nodes and 75% patient who survived had atmost 3 axil_nodes.
2. 25% patient who died had atmost 1 axil_nodes and 50% patient who died had atmost 4 axil_nodes.
3. 90% patient either survived or dead had age atmost 67yrs and op_year was atmost 1967.
4. If we see the qunatiles of age and op_year for either dead or survived patient is almost same which explains that surv_status is barely affected by these two feature.

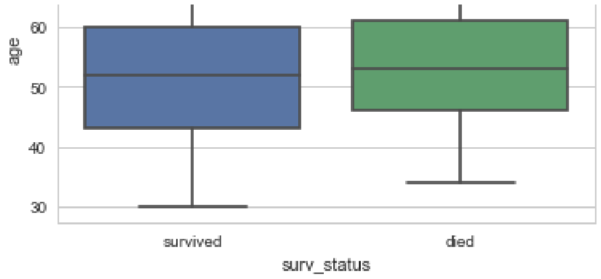
Box Plots and Whiskers

In [21]:

```
# For age
sns.boxplot(data = data, x = 'surv_status', y = 'age')
```

Out[21]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x201a85aa710>
```

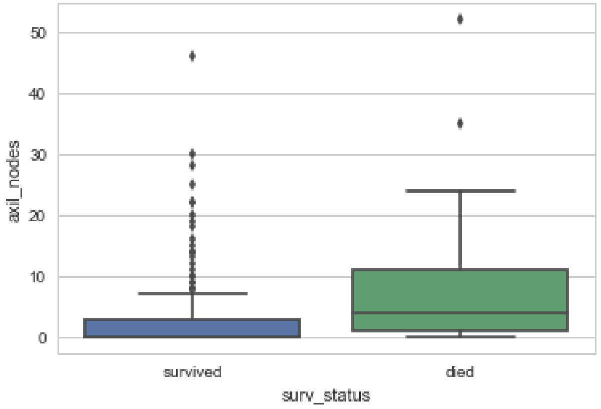


In [22]:

```
# For axil_nodes
sns.boxplot(data = data, x = 'surv_status', y = 'axil_nodes')
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x201a84cf1d0>
```



Observation:

1. 50% patient survived who had zero axil_nodes.
2. 25% patient died who had approximately 2 - 3 axil_nodes. 50% patient died who had approx. 4 axil_nodes.
3. very few patient survived who had axil_nodes more than 4.
4. So we can say there is high chance of survival if the patient has axil_nodes less than 4.

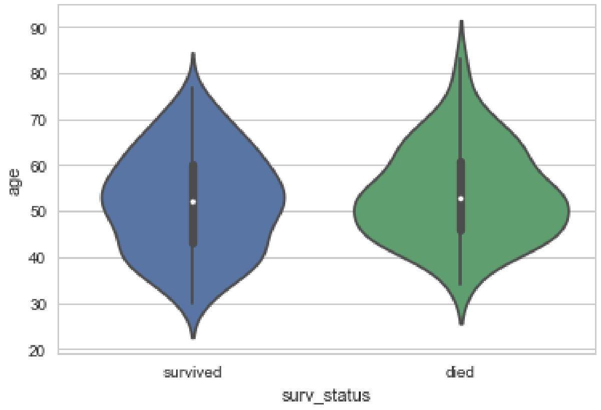
Violin Plots

In [23]:

```
# For age
sns.violinplot(data = data, x = 'surv_status', y = 'age')
```

Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x201a83835f8>
```



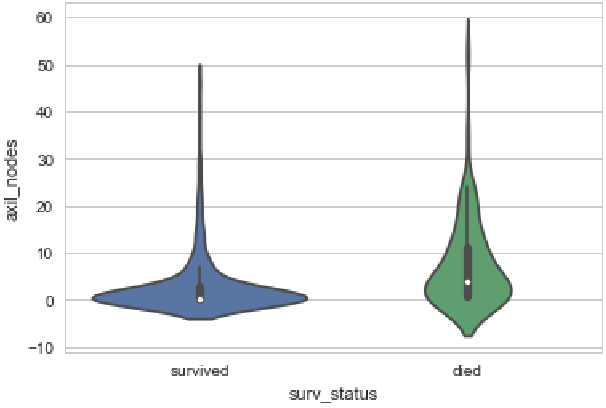
In [24]:

In [24]:

```
# For axil_nodes
sns.violinplot(data = data, x = 'surv_status', y = 'axil_nodes')
```

Out[24]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x201a832e7f0>
```



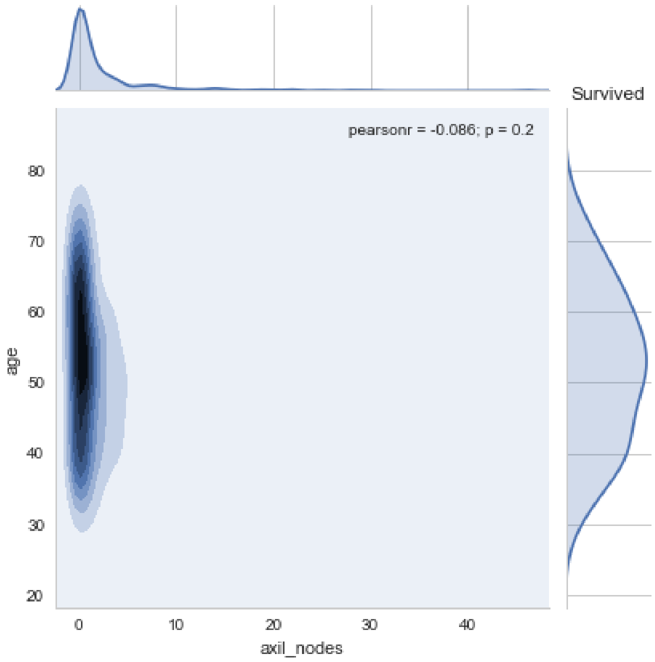
Observation:

1. By looking at the violin plot of axil_nodes it became more clear that there is high probability of survival of patient if it has 0 - 3 axil_nodes.

Multivariate Probability Density / Contour Plot

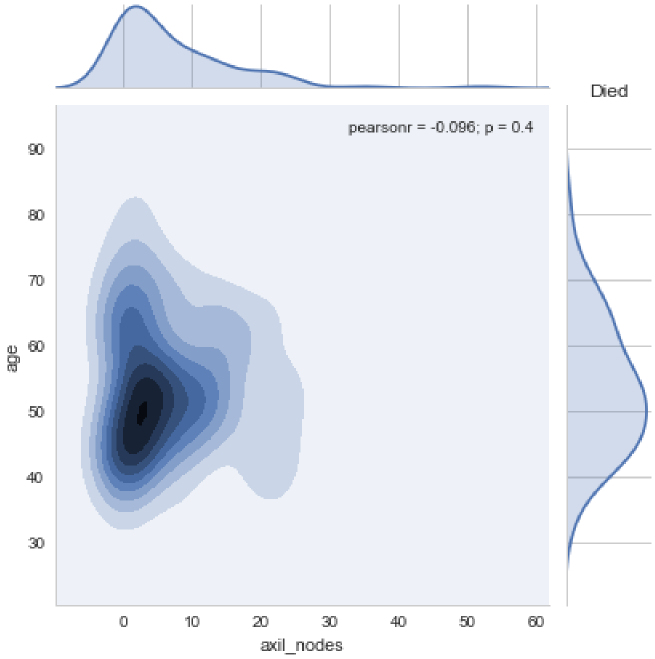
In [25]:

```
# Countour plot between age and axil_nodes of survived
sns.jointplot('axil_nodes', 'age', survived, kind = 'kde')
plt.title('Survived')
plt.show()
```



In [26]:

```
# Countour plot between age and axil_nodes of died
sns.jointplot('axil_nodes', 'age', died, kind = 'kde')
plt.title('Died')
plt.show()
```



Observation:

1. The patinet who had axil_nodes less than 3 and Irrespective of age has high probability of survival.

Conclusion

1. Best feature to do classification are as follows: axil_nodes >> age > op_year
2. op_year and age barely affect the surv_status
3. Those patient who had axil_nodes less than 3 had higher chance of survival