Haberman Cancer Survival Dataset

Description about dataset :

This dataset contains some cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of pateints who had undergone for surgery for the breast cancer.

Attributes information :

- Age of patient at time of operation. (Positive integer Value)
- Patient's year of operation. (year - 1900, Positive integer Value)
- Number of positive axillary nodes detected. (Positive integer value)
- Survival status (class attribute) basically, here
    1. '1' for the patient's who survived 5 years or longer and,
    2. '2' for the patient's who died within 5 year.

Objective :

To predict Whether a patient will survive 5 years or more after the operation based on age , year of operation and the number of positive axillary lymph nodes.

In [1]:
```
# import required modules
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sn
from statsmodels import robust

# __doc__ file
''' Download the Habermans Cancer survival dataset from https://www.kag
```

```
gle.com/gilsousa/habermans-survival-data-set '''

# print(__doc__)  # check the doc file to download dataset
# Here no column are present so we defined them.
col_names = ['age', 'operation_year', 'axil_nodes', 'surv_status']

# Load the habermans cancer survival dataset in pandas dataframe:
df = pd.read_csv('haberman.csv', names = col_names)

#df.head() # Show dataset in tabular form
```

In [2]:
```
# Number of datapoint and features:
print("Number of datapoint and features : ", df.shape)
print('\n')
df.info()
```

```
Number of datapoint and features :  (306, 4)


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306 entries, 0 to 305
Data columns (total 4 columns):
age               306 non-null int64
operation_year    306 non-null int64
axil_nodes        306 non-null int64
surv_status       306 non-null int64
dtypes: int64(4)
memory usage: 9.6 KB
```

In [3]:
```
# Column names in this dataset
# | Value         |   column names |
# 30  (age, numerical)            - Age of patient at time of operation
# 640 (operation_year, numerical) - Patient's year of operation
# 1   (axil_nodes, numerical)     - Number of positive axillary nodes d
etected (axil_nodes)
# 1   (surv_status, numerical)    - Survival status (class attribute)
 '1' = the patient survived 5 years or longer '2' = the patient died wi
thin 5 year (Surv_status)
```

```
# Data point present in each Class "1" and class "2" of Survival statu
s.

print("Data points present in each class : ")
print('Class | Values')
df['surv_status'].value_counts()
```

```
Data points present in each class :
Class | Values
```

Out[3]:
```
1    225
2     81
Name: surv_status, dtype: int64
```

Obeservation :

1. There are total 306 rows and 4 column. Indexing is from 0 to 305 and values are all integer type.

1. Columns are age, operation_year, axil_nodes and surv_status.

1. We find there are two classes.

    - Class '1' and Class '2', where Class 1 corresponds to those patients who survived atleast 5 year or longer after the success of the operation.
    - Whereas Class 2 corresponds to those patients who died within 5 years of the operation.

1. Class 1 contains 225 values and Class 2 have 81 values, Hence In Habermans cancer Survival dataset each class values is differ to a significant extent. This dataset is an imbalanced dataset.

Mean, Variance and Std-dev

In [4]:
```
# Now, Generates the descriptive statistics that summarize the central
 tendency, dispersion and shape of a dataset's distribution, excluding
```

```
 NaN (Missing) values.
survived = df.loc[df['surv_status'] == 1] # Class 1
died = df.loc[df['surv_status'] == 2] # Class 2


# Basic Statistics for Class 1 Pateint's :
print('Patients who survived atleast 5 years or longer after success of
 the operation')
survived.describe()
```

```
Patients who survived atleast 5 years or longer after success of the op
eration
```

Out[4]:

	age	operation_year	axil_nodes	surv_status
count	225.000000	225.000000	225.000000	225.0
mean	52.017778	62.862222	2.791111	1.0
std	11.012154	3.222915	5.870318	0.0
min	30.000000	58.000000	0.000000	1.0
25%	43.000000	60.000000	0.000000	1.0
50%	52.000000	63.000000	0.000000	1.0
75%	60.000000	66.000000	3.000000	1.0
max	77.000000	69.000000	46.000000	1.0

In [5]:

```
# Statistic about Class 2 Patient's :
print('Patients who died within 5 year of the operation')
died.describe()
```

```
Patients who died within 5 year of the operation
```

Out[5]:

	age	operation_year	axil_nodes	surv_status
count	81.000000	81.000000	81.000000	81.0

	age	operation_year	axil_nodes	surv_status
mean	53.679012	62.827160	7.456790	2.0
std	10.167137	3.342118	9.185654	0.0
min	34.000000	58.000000	0.000000	2.0
25%	46.000000	59.000000	1.000000	2.0
50%	53.000000	63.000000	4.000000	2.0
75%	61.000000	65.000000	11.000000	2.0
max	83.000000	69.000000	52.000000	2.0

Observation :

1. As, we can see here the Statistics of age about the patient's who survived after 5 year is almost similar to those patient's who died within 5 year. Similarly year of the operation had similar statistics for both classes. But, the axil_nodes (positive axilary nodes) varies the number of positive axillary nodes (axil_nodes) varies for both Class 1 and class 2.

1. Clearly, Above 75% of survived patients had positive axillary nodes less than or equal to 3 and 75 % of dead patients had positive axillarey nodes more than 3 or less than equal to 11 Hence, this parameter varies the most. Note that the patients with less number of positive axillary nodes has the higher chance of survival.

1. Also, the minimum number of Positive Axillary nodes is zero '0' for the dead patients so, Positive axillary nodes alone is not that useful/suitable for determining the survival of patients.

Median, Percentile, Quantile, IQR, MAD

In [6]:
```
#Median, Quantiles, Percentiles, IQR.
```

```
print("\nMedians:")
print(np.median(survived["axil_nodes"]))
print(np.median(died['axil_nodes']))

print("\nQuantiles:")
print(np.percentile(survived["axil_nodes"],np.arange(0, 100, 25)))
print(np.percentile(died["axil_nodes"],np.arange(0, 100, 25)))

print("\n90th Percentiles:")
print(np.percentile(survived["axil_nodes"],90))
print(np.percentile(died["axil_nodes"],90))

print ("\nMedian Absolute Deviation")
print(robust.mad(survived["axil_nodes"]))
print(robust.mad(died["axil_nodes"]))
```

```
Medians:
0.0
4.0

Quantiles:
[0. 0. 0. 3.]
[ 0.  1.  4. 11.]

90th Percentiles:
8.0
20.0

Median Absolute Deviation
0.0
5.930408874022408
```

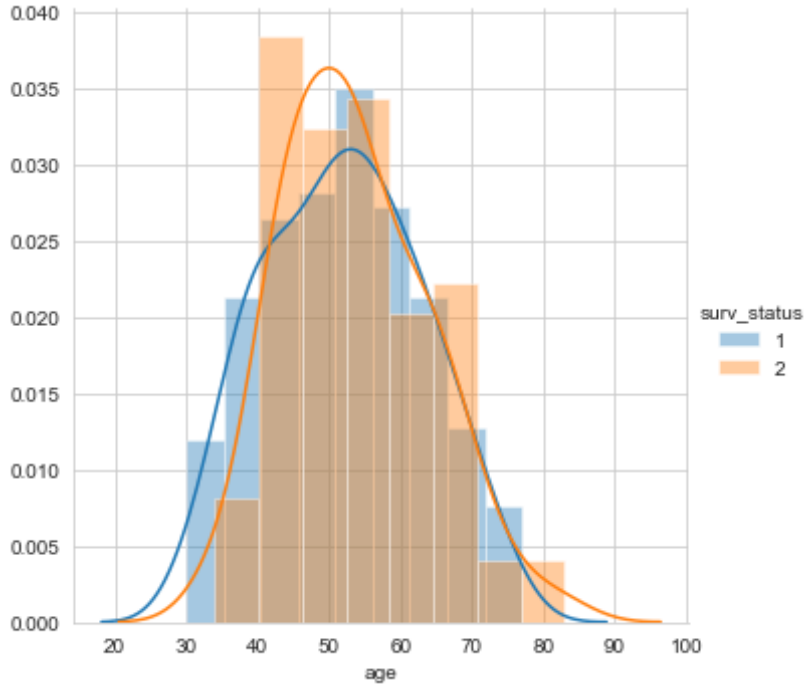
UNIVARIATE ANALYSIS

Histogram, PDF - Age

In [7]:
```
sn.set_style("whitegrid");
```

```
sn.FacetGrid(df , hue = "surv_status" , height = 5) \
.map(sn.distplot , "age") \
.add_legend();

plt.show()
```

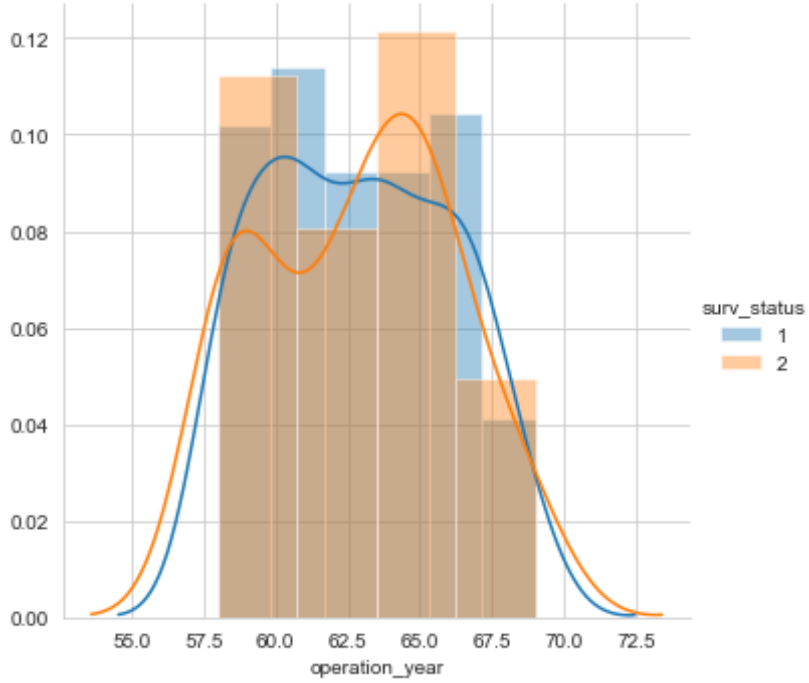
```
E:\Arpit\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureW
arning: Using a non-tuple sequence for multidimensional indexing is dep
recated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future thi
s will be interpreted as an array index, `arr[np.array(seq)]`, which wi
ll result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```



Histogram, PDF - Operation year

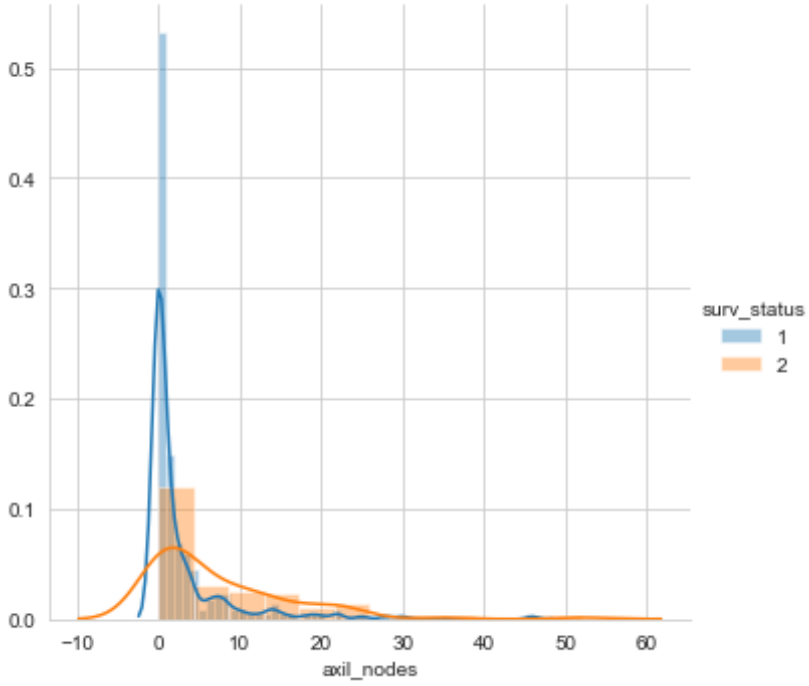
In [8]:
```
sn.set_style("whitegrid");
sn.FacetGrid(df , hue = "surv_status" , height = 5) \
```

```
.map(sn.distplot , "operation_year") \
.add_legend();

plt.show()
```



Histogram, PDF - Positive axillary nodes

In [9]:
```
sn.set_style("whitegrid");
sn.FacetGrid(df , hue = "surv_status" , height = 5) \
.map(sn.distplot , "axil_nodes") \
.add_legend();

plt.show()
```



Observation :

1. PDF's of both the feature 'age' and 'operation_year' are not that good as that of 'axil_nodes'. They look similar for both class labels.

1. In Year 1965, the number of deaths are maximum.

1. 'axil_nodes' seem to be a good/useful feature for the classification of both classes. but from this didtribution we can infer that most survived patients fall in Zero '0' positive axillary nodes.

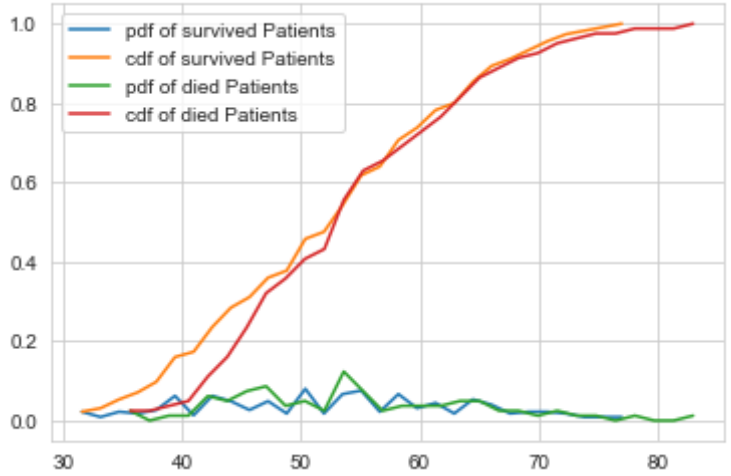
CDF - Age

In [10]:
```
counts, bin_edges = np.histogram(survived['age'], bins=30,
                                 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)

counts, bin_edges = np.histogram(died['age'], bins=30, 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.legend(['pdf of survived Patients', 'cdf of survived Patients','pdf
 of died Patients', 'cdf of died Patients'])

plt.show()
```

```
[0.02222222 0.00888889 0.02222222 0.01777778 0.02666667 0.06222222
 0.01333333 0.06222222 0.04888889 0.02666667 0.04888889 0.01777778
 0.08       0.01777778 0.06666667 0.07555556 0.02222222 0.06666667
 0.03111111 0.04444444 0.01777778 0.05333333 0.04       0.01777778
 0.02222222 0.02222222 0.01777778 0.00888889 0.00888889 0.00888889]
[30.         31.56666667 33.13333333 34.7        36.26666667 37.8333333
3
 39.4        40.96666667 42.53333333 44.1        45.66666667 47.2333333
3
 48.8        50.36666667 51.93333333 53.5        55.06666667 56.6333333
3
 58.2        59.76666667 61.33333333 62.9        64.46666667 66.0333333
3
 67.6        69.16666667 70.73333333 72.3        73.86666667 75.4333333
3
 77.        ]
[0.02469136 0.         0.01234568 0.01234568 0.0617284  0.04938272
 0.07407407 0.08641975 0.03703704 0.04938272 0.02469136 0.12345679
 0.07407407 0.02469136 0.03703704 0.03703704 0.03703704 0.04938272
 0.04938272 0.02469136 0.02469136 0.01234568 0.02469136 0.01234568
 0.01234568 0.         0.01234568 0.         0.         0.01234568]
```

```
[34.         35.63333333 37.26666667 38.9        40.53333333 42.1666666
7
 43.8        45.43333333 47.06666667 48.7        50.33333333 51.9666666
7
 53.6        55.23333333 56.86666667 58.5        60.13333333 61.7666666
7
 63.4        65.03333333 66.66666667 68.3        69.93333333 71.5666666
7
 73.2        74.83333333 76.46666667 78.1        79.73333333 81.3666666
7
 83.        ]
```



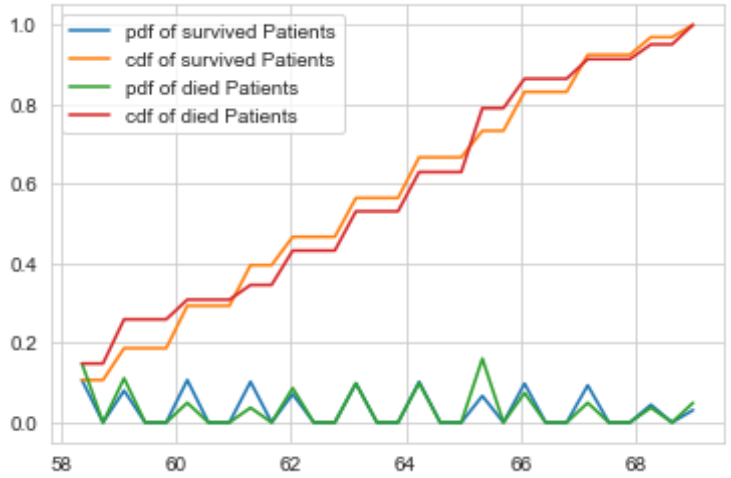
CDF - Operation Year

In [11]:
```
counts, bin_edges = np.histogram(survived['operation_year'], bins=30,
                                 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)
```

```
counts, bin_edges = np.histogram(died['operation_year'], bins=30, densi
ty = 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.legend(['pdf of survived Patients', 'cdf of survived Patients','pdf
 of died Patients', 'cdf of died Patients'])

plt.show()
```

```
[0.10666667 0.         0.08       0.         0.         0.10666667
 0.         0.         0.10222222 0.         0.07111111 0.
 0.         0.09777778 0.         0.         0.10222222 0.
 0.         0.06666667 0.         0.09777778 0.         0.
 0.09333333 0.         0.         0.04444444 0.         0.03111111]
[58.         58.36666667 58.73333333 59.1        59.46666667 59.8333333
3
 60.2        60.56666667 60.93333333 61.3        61.66666667 62.0333333
3
 62.4        62.76666667 63.13333333 63.5        63.86666667 64.2333333
3
 64.6        64.96666667 65.33333333 65.7        66.06666667 66.4333333
3
 66.8        67.16666667 67.53333333 67.9        68.26666667 68.6333333
3
 69.        ]
[0.14814815 0.         0.11111111 0.         0.         0.04938272
 0.         0.         0.03703704 0.         0.08641975 0.
 0.         0.09876543 0.         0.         0.09876543 0.
 0.         0.16049383 0.         0.07407407 0.         0.
 0.04938272 0.         0.         0.03703704 0.         0.04938272]
[58.         58.36666667 58.73333333 59.1        59.46666667 59.8333333
3
 60.2        60.56666667 60.93333333 61.3        61.66666667 62.0333333
3
 62.4        62.76666667 63.13333333 63.5        63.86666667 64.2333333
```

```
3
 64.6         64.96666667 65.33333333 65.7         66.06666667 66.4333333
3
 66.8         67.16666667 67.53333333 67.9         68.26666667 68.6333333
3
 69.         ]
```



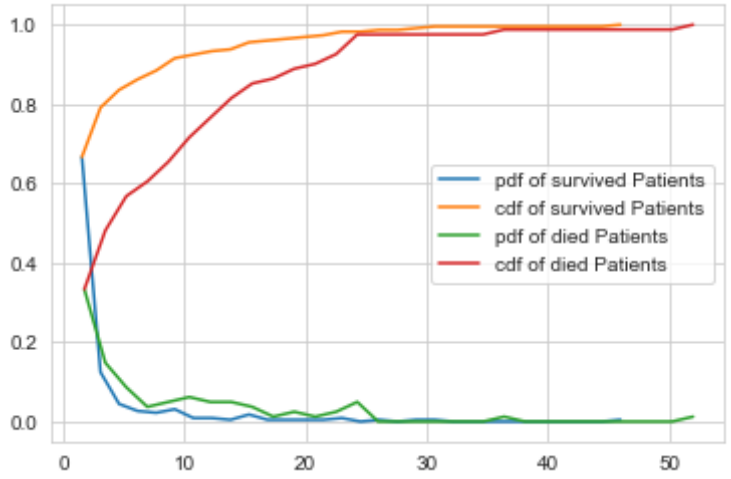
CDF - Positive Axillary nodes

In [12]:
```
counts, bin_edges = np.histogram(survived['axil_nodes'], bins=30,
                                 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)

counts, bin_edges = np.histogram(died['axil_nodes'], bins=30, 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.legend(['pdf of survived Patients', 'cdf of survived Patients','pdf
 of died Patients', 'cdf of died Patients'])

plt.show()
```

```
[0.66666667 0.12444444 0.04444444 0.02666667 0.02222222 0.03111111
 0.00888889 0.00888889 0.00444444 0.01777778 0.00444444 0.00444444
 0.00444444 0.00444444 0.00888889 0.         0.00444444 0.
 0.00444444 0.00444444 0.         0.         0.         0.
 0.         0.         0.         0.         0.         0.00444444]
[ 0.          1.53333333  3.06666667  4.6         6.13333333  7.6666666
7
  9.2        10.73333333 12.26666667 13.8        15.33333333 16.8666666
7
 18.4        19.93333333 21.46666667 23.         24.53333333 26.0666666
7
 27.6        29.13333333 30.66666667 32.2        33.73333333 35.2666666
7
 36.8        38.33333333 39.86666667 41.4        42.93333333 44.4666666
7
 46.        ]
[0.33333333 0.14814815 0.08641975 0.03703704 0.04938272 0.0617284
 0.04938272 0.04938272 0.03703704 0.01234568 0.02469136 0.01234568
 0.02469136 0.04938272 0.         0.         0.         0.
 0.         0.         0.01234568 0.         0.         0.
 0.         0.         0.         0.         0.         0.01234568]
[ 0.          1.73333333  3.46666667  5.2         6.93333333  8.6666666
7
 10.4        12.13333333 13.86666667 15.6        17.33333333 19.0666666
7
 20.8        22.53333333 24.26666667 26.         27.73333333 29.4666666
7
 31.2        32.93333333 34.66666667 36.4        38.13333333 39.8666666
7
 41.6        43.33333333 45.06666667 46.8        48.53333333 50.2666666
```

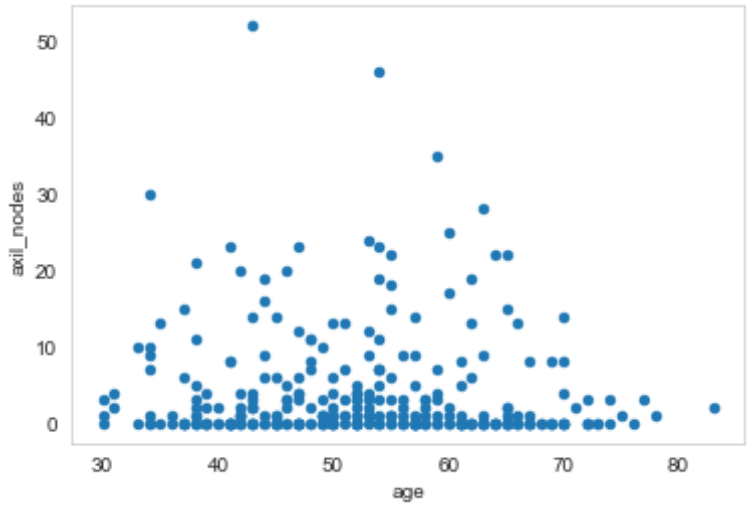
```
7
 52.        ]
```



BI-VARIATE ANALYSIS

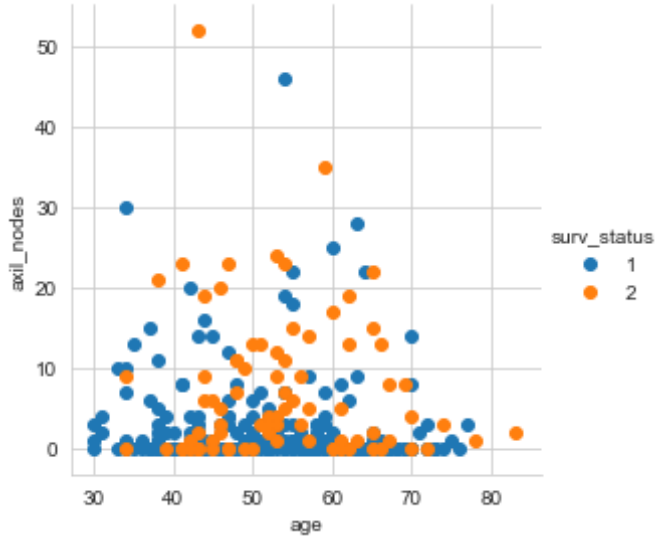
2 - D Scatter Plot

In [13]:
```
#2-D scatter plot:
df.plot(kind='scatter', x='age', y='axil_nodes') ;

#MAke grid:
plt.grid()

#show plot:
plt.show()
```



Observation : The density of number of Positive axillary node is more in between 0 & 5 nodes and Most patients have Zero '0' Positive axillary nodes.

In [14]:
```
# 2-D Scatter plot with color-coding for each patients type/class.
# Here 'sn' corresponds to seaborn.
# Set_style:
sn.set_style("whitegrid");

sn.FacetGrid(df, hue="surv_status", height=4) \
   .map(plt.scatter, "age", "axil_nodes") \
   .add_legend();

plt.show();
```



Observation :

1. Most of the patients have Zero '0' Positive axillary node which are detected as Blue and Orange balls. Clearly these two point are not Linearly separable.

1. These two parameter 'axil_nodes' and 'age' aren't useful by ploting 2-D Scatter Plot. Now let's try the Pair-Plot if we find any Combination for three features 'age', 'operation_year', 'axil_nodes'.

Pair Plot

In [15]:
```
# close plot:
plt.close()

# set_style:
sn.set_style('whitegrid')

# Pair-Plot:
sn.pairplot(df, hue='surv_status', vars=['age', 'operation_year', 'axil
```

```
_nodes'], height=3)

# Show Plot:
plt.show()
```

Observation :

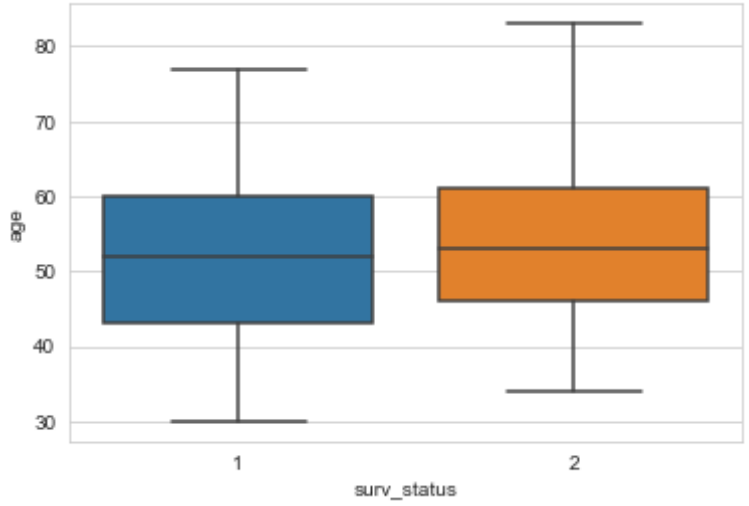
1. As we can see here, most of the points overlapped eachother. Hence, it is difficult to separate them.

1. Classification is difficult of a patient based on his features and positive axillary node doesn't depend on the age of the patients.

Box Plots, Whiskers and Violin Plots

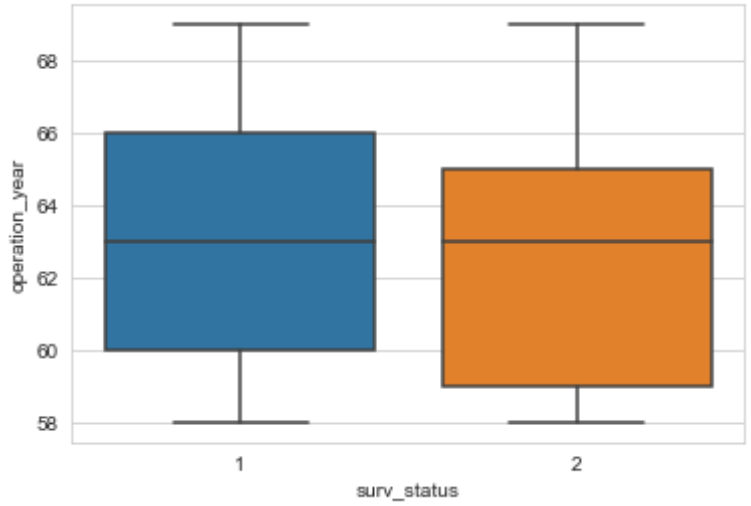
Box Plot - Age

In [16]:
```
sn.boxplot(x='surv_status',y='age', data= df)
plt.show()
```



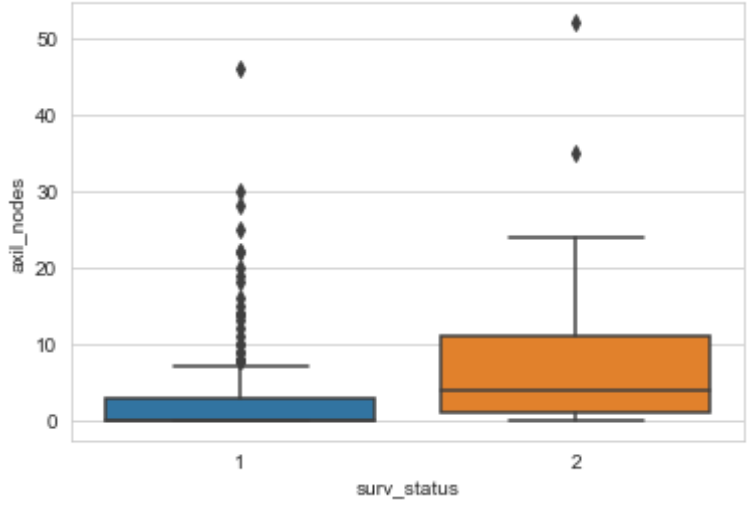
Box Plot - Operation Year

In [17]:
```
sn.boxplot(x='surv_status',y='operation_year', data= df)
plt.show()
```



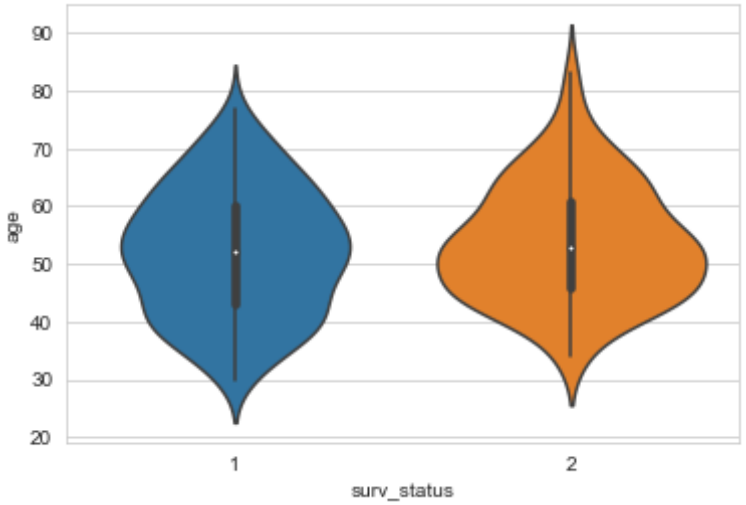
Box Plot - Positive Axillary nodes

In [18]:
```
sn.boxplot(x='surv_status',y='axil_nodes', data= df)
plt.show()
```



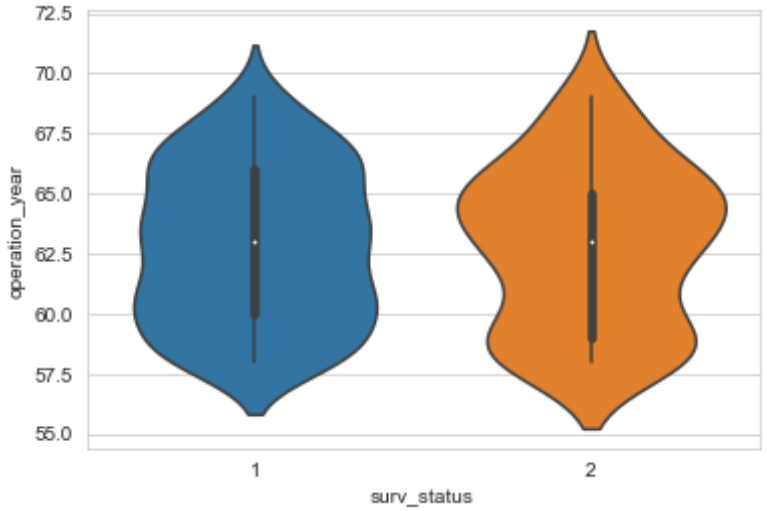
Violin Plot - Age

In [19]:
```
sn.violinplot(x="surv_status", y="age", data = df, size=8)
plt.show()
```



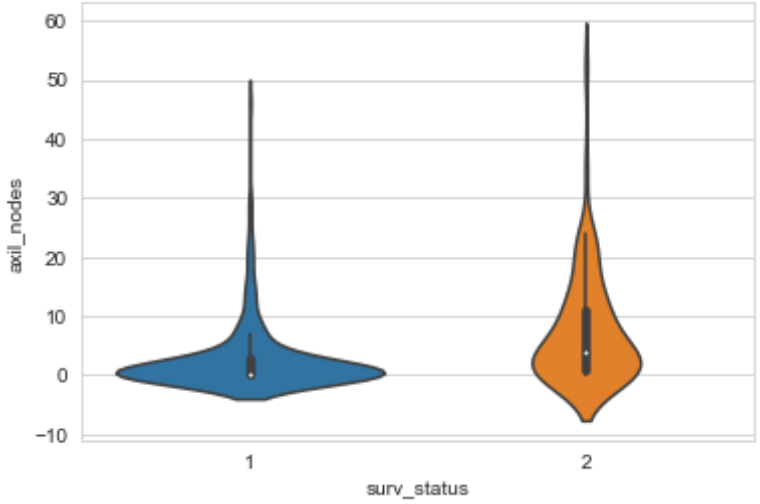
Violin Plot - Operation Year

In [20]:
```
sn.violinplot(x="surv_status", y="operation_year", data = df, size=8)
plt.show()
```



Violin Plot - Positive Axillary nodes

In [21]:
```
sn.violinplot(x="surv_status", y="axil_nodes", data = df, size=8)
plt.show()
```



Observation :

1. The number of positive lymph nodes of the survivors is highly densed from 0 to 5.

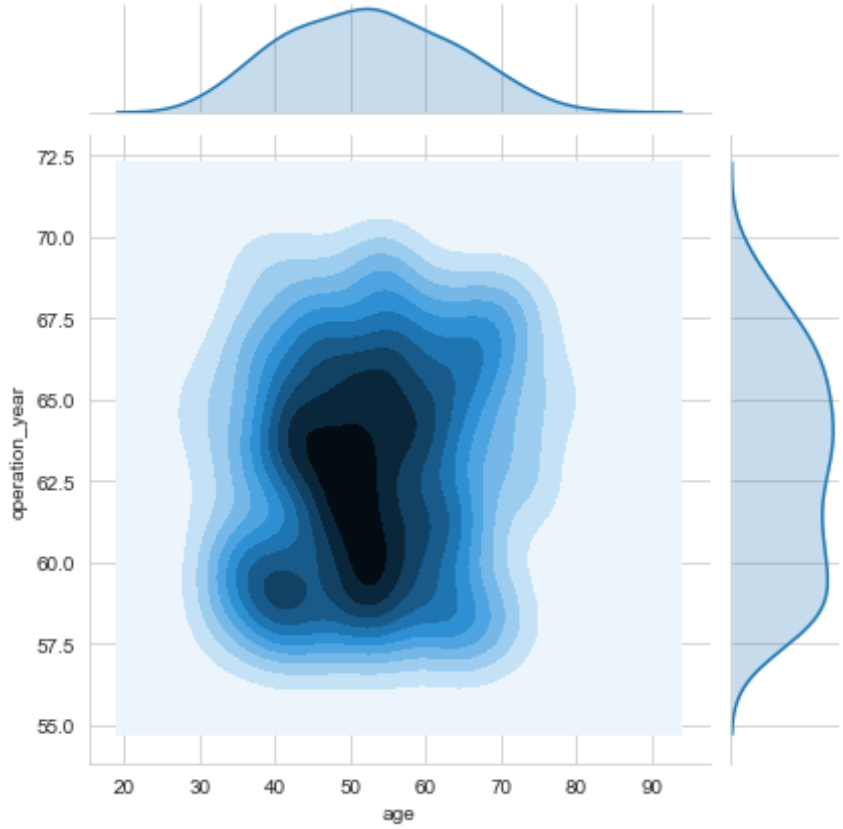
1. Almost 80% of the patients have less than or equal to 5 positive lymph survived more than 5 years.

1. From box plots and violin plots, we can say that more no of patients who are dead have age between 46-62,year between 59-65 and the patients who survived have age between 42-60, year between 60-66.

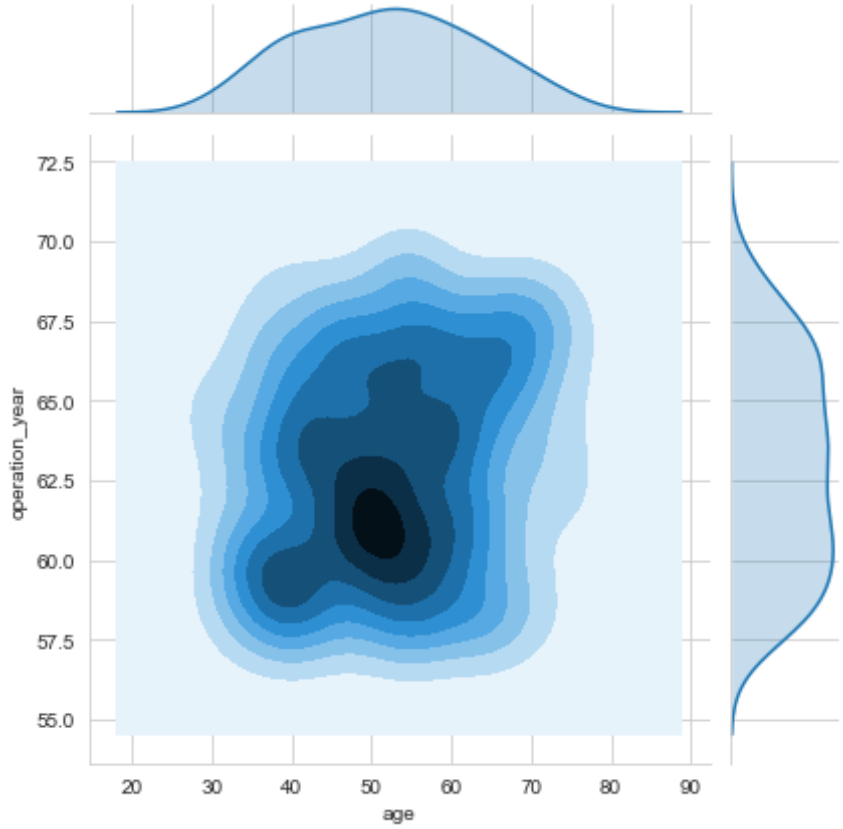
Contour Plot

(x, age) and (y, operation_year)

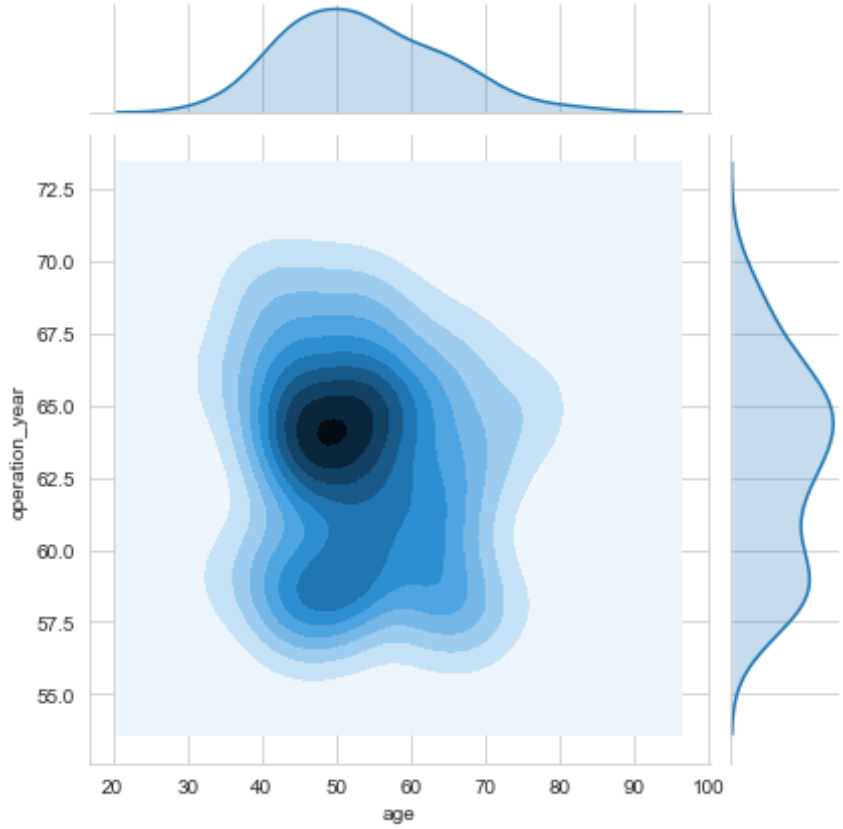
In [22]:
```
sn.jointplot(x="age", y="operation_year", data=df, kind="kde");
plt.show()
```



In [23]:
```
sn.jointplot(x="age", y="operation_year", data=survived, kind="kde");
plt.show()
```



In [24]:
```
sn.jointplot(x="age", y="operation_year", data=died, kind="kde");
plt.show()
```



Observation :

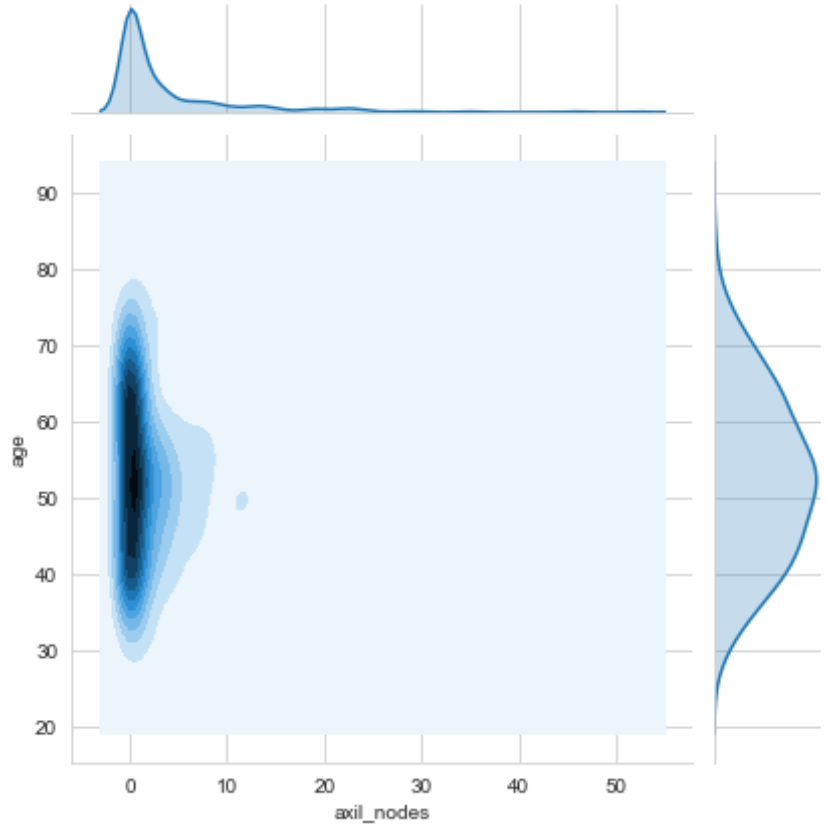
1. Hence, More number of patients of age between 45 - 58 year undergone to operation during the Year 1958 to 1964.

1. Most patients of age 48 - 58 year survived after 5 year of operation during year 1959 to 1962

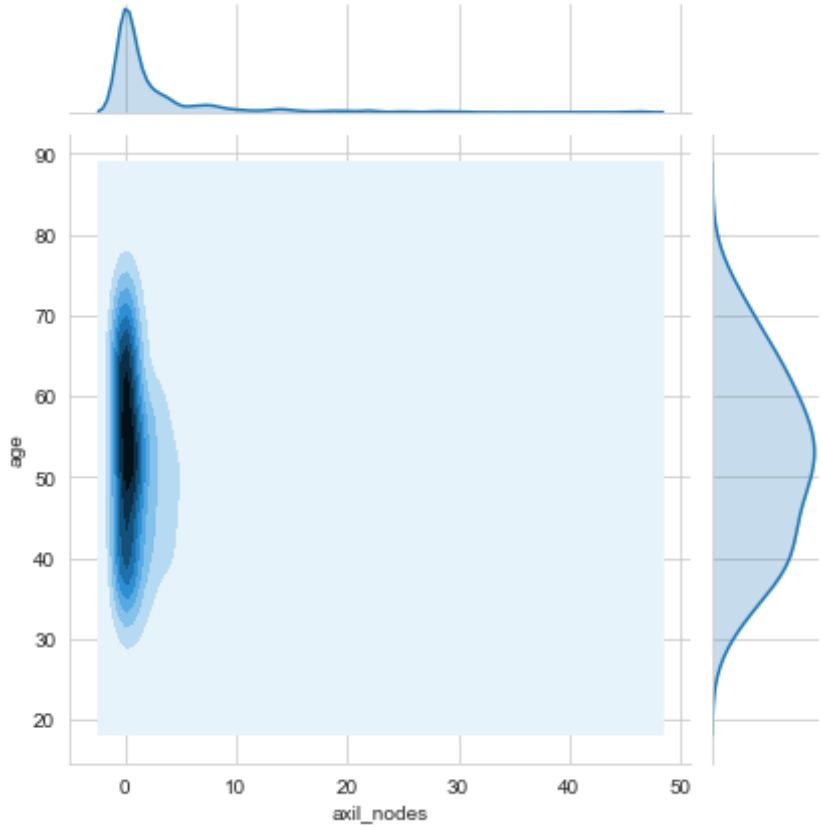
1. Most patients of age 49 - 51 year died within 5 year of operation during 1963 to 1964

(x , axil_nodes) and (y, age)

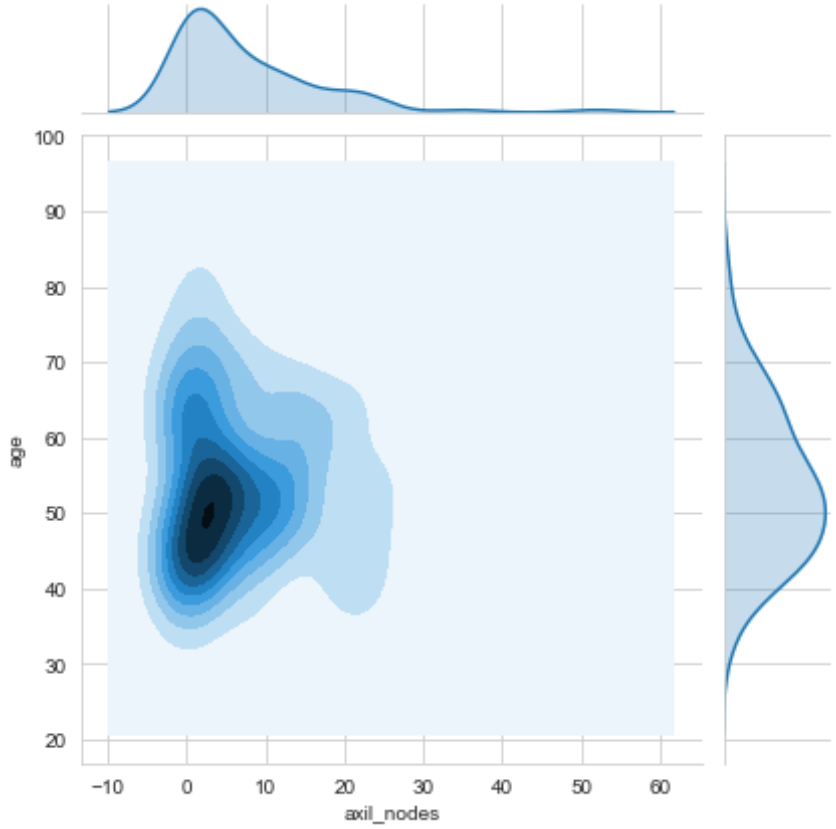
In [25]:
```
sn.jointplot(x="axil_nodes", y="age", data=df, kind="kde");
plt.show()
```



In [26]:
```
sn.jointplot(x="axil_nodes", y="age", data=survived, kind="kde");
plt.show()
```



In [27]:
```
sn.jointplot(x="axil_nodes", y="age", data=died, kind="kde");
plt.show()
```



Operation : Their is no dependency between age and number of positive axillary nodes.

Conclusions:

1. Shape of the dataset is (306, 4). There are 306 observations and 4 features.

1. 225 patients belongs to class '1' and 81 belongs to class '2' Hence, this dataset is imbaanced dataset.

1. So, we find from contour plot that the feature 'age' and 'operation_year' are not the deciding features for the patients survival chances.

1. Hence, Survival chances of patients generally depends upon the feature 'axil_nodes' or the number of Positive Axillary nodes.

1. More number of patients of age between 45 - 58 year undergone to operation during the Year 1958 to 1964. 52 year is the Mean age of patient who survived and 54 year is the Mean age who died.

Submitted by :

Name : ARPIT DUBEY

Email : aarpitdubey@gmail.com

Phone : +91 825-1999-950