Haberman Cancer Survival Analysis

Creating dataframe "survival"

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

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

In [2]:
```
survival.columns = ['age','year','axil','expectancy']     #Assigning na
mes to the columns
survival.head(5)
```

Out[2]:

	age	year	axil	expectancy
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

Additional Info:

'1' Denotes the person survived for 5 years or more i.e Survived class

'2' Denotes the person died withing 5 years of operation i.e Not survived Class

Checking the number of entries of each class of the dataframe

In [3]:
```
survival["expectancy"].value_counts()
```

Out[3]:
```
1    224
2     81
dtype: int64
```

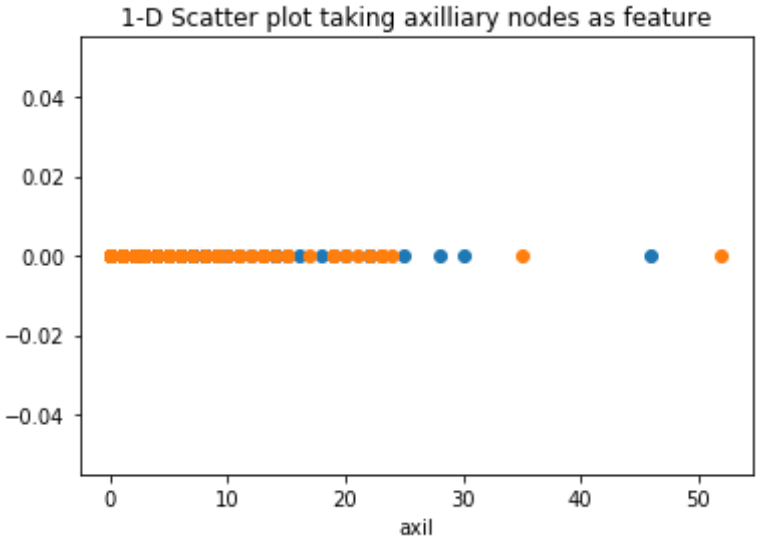
Obeservation:

1. Imbalance dataset containing ~74% data of class 1.

Univariate Analysis

1-D Scatter Plot (Feature : Number of Positive Axilliary Nodes)

In [5]:
```
import numpy as np
Yes = survival.loc[survival['expectancy'] == 1];
No = survival.loc[survival['expectancy'] == 2];

plt.plot(Yes["axil"], np.zeros_like(Yes['axil']), 'o')
plt.plot(No["axil"], np.zeros_like(No['axil']), 'o')
plt.xlabel("axil")
plt.title('1-D Scatter plot taking axilliary nodes as feature')
plt.show()
```

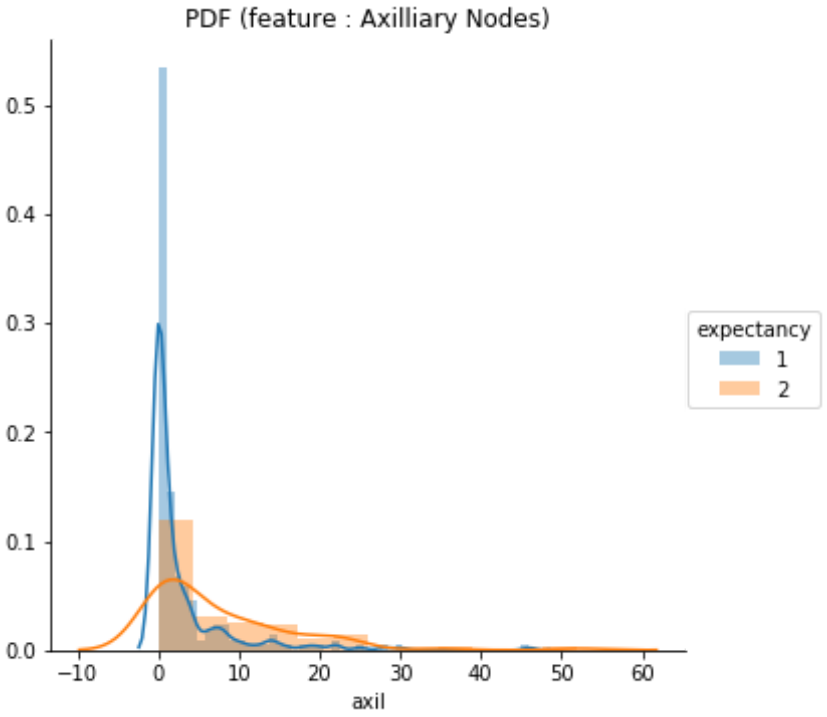


Obeservation:

1. Using auxilliary nodes as features, we can distinguish the classes.
1. In this graph (axil < 24) has more points belonging to "Survived" class.

Probability Density Function (Number of Positive Axilliary nodes as feature)

In [27]:
```
sns.FacetGrid(survival, hue="expectancy", size=5) \
   .map(sns.distplot, "axil") \
   .add_legend();
plt.title("PDF (feature : Axilliary Nodes)")
plt.show()
```



Obeservations:

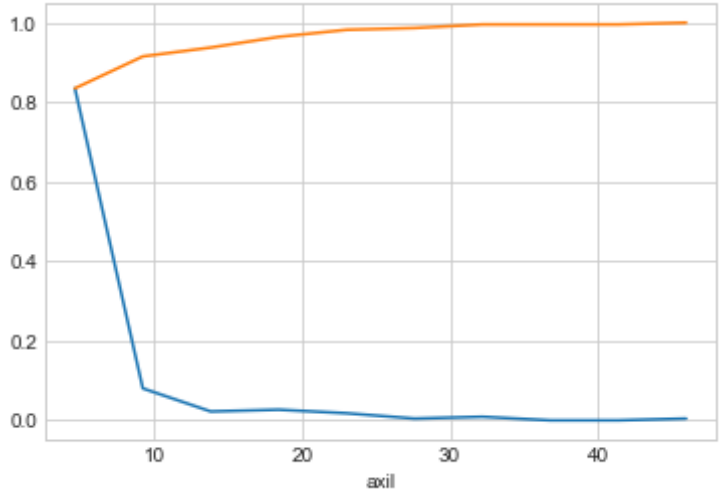
1. Graph is higly overlaped, hence it can't classify the classes clearly.
2. Number of positive auxilliary nodes in the range 0 to 4 has more points belonging to '1'.
3. Number of positive auxilliary nodes in range 4 to 30 has more points belonging to '2'.
4. Maximum number class '1' points occurs when no Positive Auxiliary Nodes found.

Plotting CDF (For survived class)

In [34]:
```
#Person Survived
counts, bin_edges = np.histogram(Yes['axil'], 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.xlabel("axil")
```

```
[0.83482143 0.08035714 0.02232143 0.02678571 0.01785714 0.00446429
 0.00892857 0.         0.         0.00446429]
[ 0.   4.6  9.2 13.8 18.4 23.  27.6 32.2 36.8 41.4 46. ]
```

Out[34]: Text(0.5,0,'axil')



Obeservation:

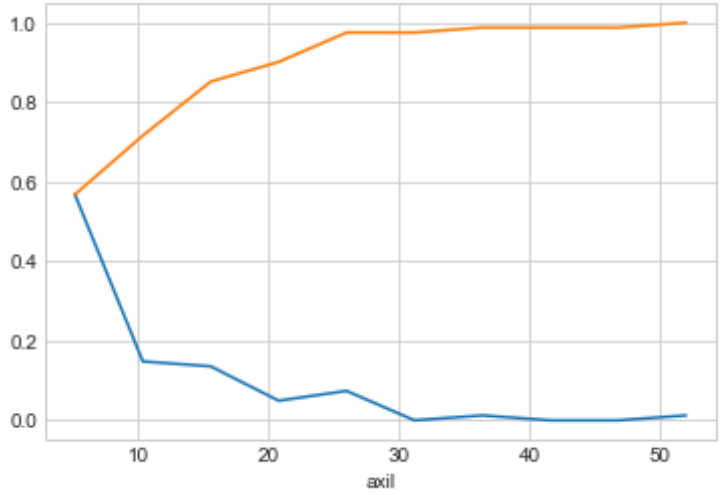
1. We can't draw any exact conclusions or if else module from th is CDF plot.

Plotting CDF (For not survived class)

In [26]:
```
# Person not survived
counts, bin_edges = np.histogram(No['axil'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.xlabel("axil")
plt.plot(bin_edges[1:], cdf)
```

```
[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. ]
```

Out[26]:
```
[<matplotlib.lines.Line2D at 0x20dd9fb6b38>]
```

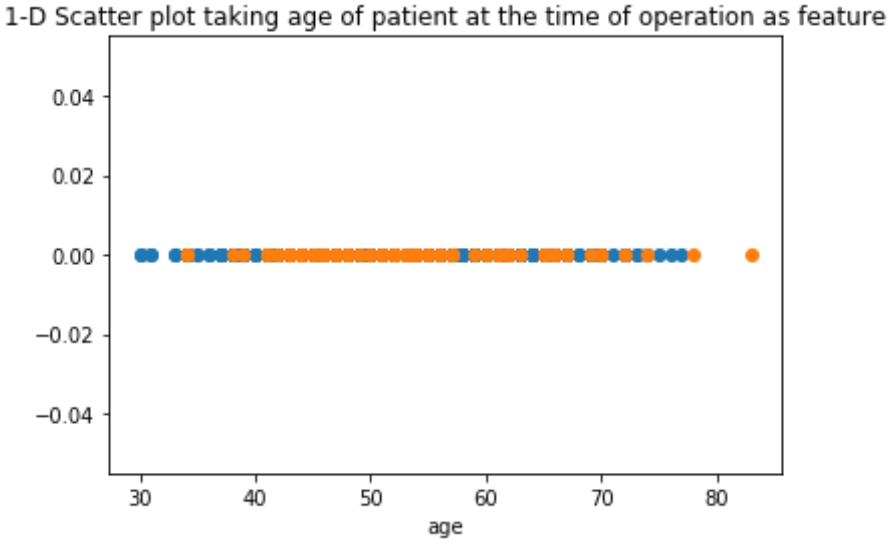


Obeservation:

1. We can't draw any exact conclusion from this graph.

1-D Scatter Plot (Feature : Age of Patient)

In [25]:
```
import numpy as np
Yes = survival.loc[survival['expectancy'] == 1];
No = survival.loc[survival['expectancy'] == 2];
plt.xlabel("age")
plt.title('1-D Scatter plot taking age of patient at the time of operat
ion as feature')
plt.plot(Yes["age"], np.zeros_like(Yes['age']), 'o')
plt.plot(No["age"], np.zeros_like(No['age']), 'o')
plt.show()
```

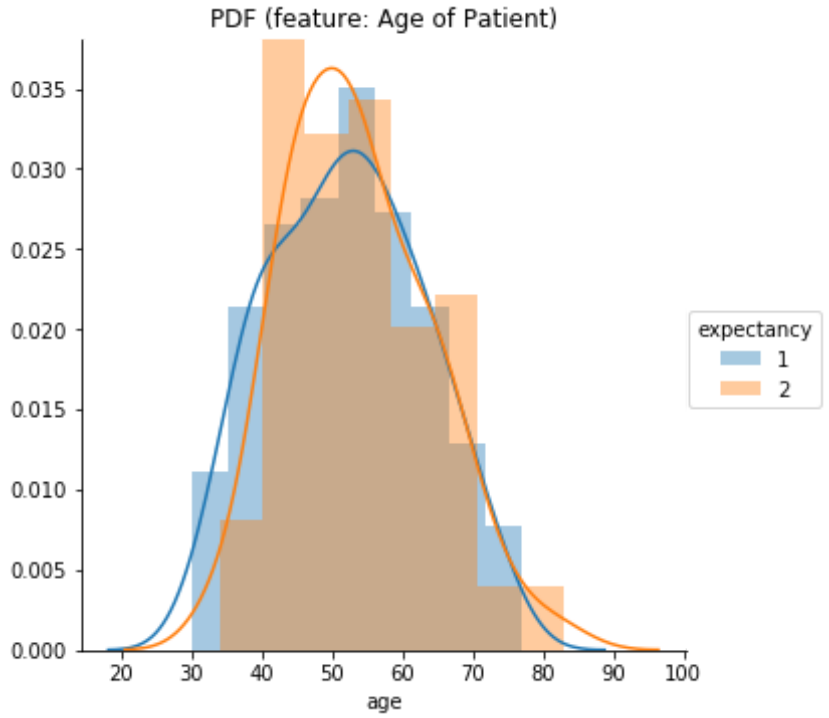


Obeservations:

1. Using age, we can somehow distinguish the two classes.
2. Age of range 40 - ~62 has more points belonging to survived class.

Probability Density Function (Age of Patient as feature)

In [28]:
```
sns.FacetGrid(survival, hue="expectancy", size=5) \
   .map(sns.distplot, "age") \
   .add_legend();
plt.title("PDF (feature: Age of Patient)")
plt.show();
```



Obeservations:

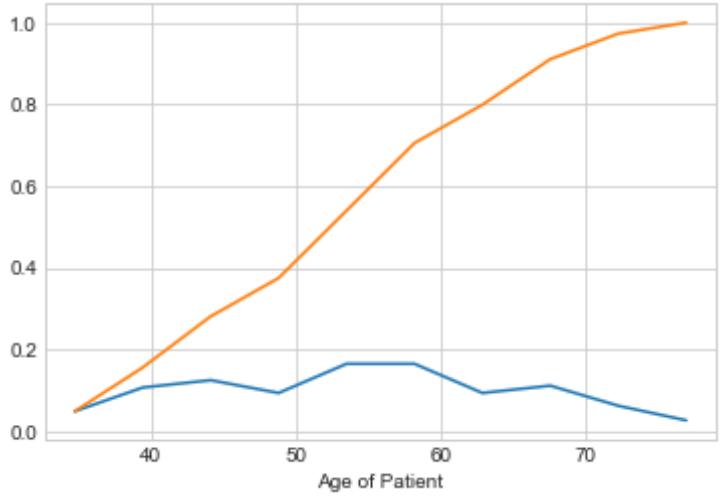
1.Graph is higly overlaped, hence it can't distinguish the classes clearly.

Plotting CDF (For Survived class)

In [22]:
```
counts, bin_edges = np.histogram(Yes['age'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.xlabel("Age of Patient")
plt.plot(bin_edges[1:], cdf)
```

```
[0.04910714 0.10714286 0.125      0.09375    0.16517857 0.16517857
 0.09375    0.11160714 0.0625     0.02678571]
[30.  34.7 39.4 44.1 48.8 53.5 58.2 62.9 67.6 72.3 77. ]
```

Out[22]:
```
[<matplotlib.lines.Line2D at 0x20dda07b1d0>]
```



Obeservations:

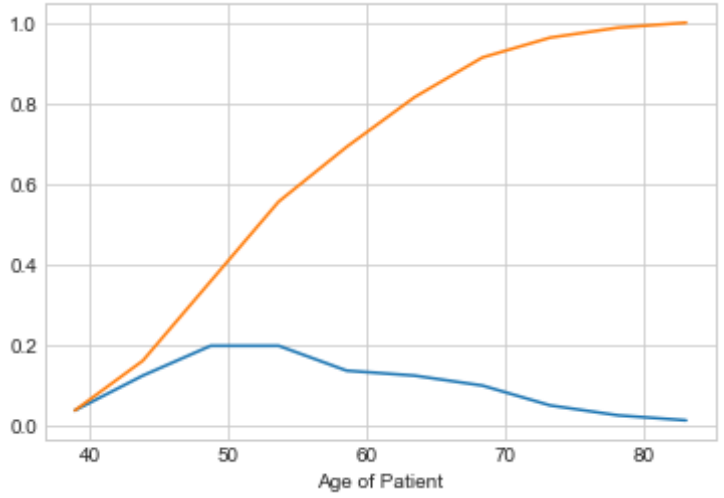
1.We can't draw any exact conclusion from this graph.

Plotting CDF (For Not Survived class)

In [29]:
```
counts, bin_edges = np.histogram(No['age'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.xlabel("Age of Patient")
plt.plot(bin_edges[1:], cdf)
```

```
[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. ]
```

Out[29]:
```
[<matplotlib.lines.Line2D at 0x20ddb13bc18>]
```



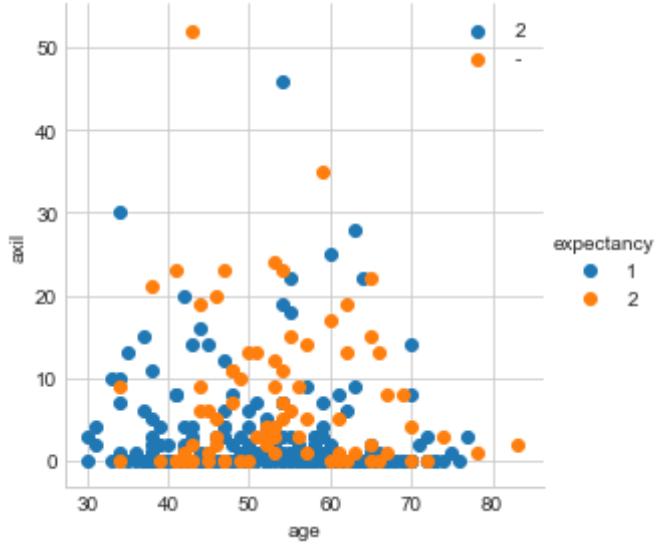
Obeservations:

1.We can't draw any exact conclusion from this graph.

Bi-variate Analysis

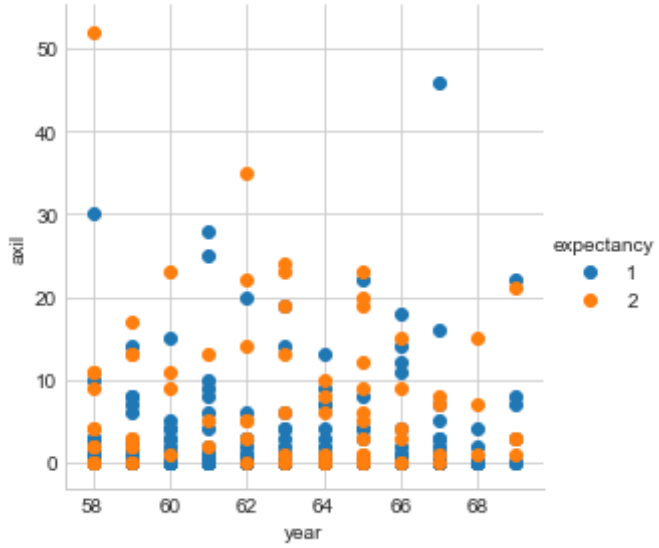
2-D Scatter plt (features: axil and age)

In [16]:
```
sns.set_style("whitegrid");
sns.FacetGrid(survival, hue="expectancy", size=4) \
   .map(plt.scatter, "age", "axil") \
   .add_legend();
plt.legend("2-D Scatter plot (X-axis: age, Y-axis:axilliary nodes")
plt.show();
```



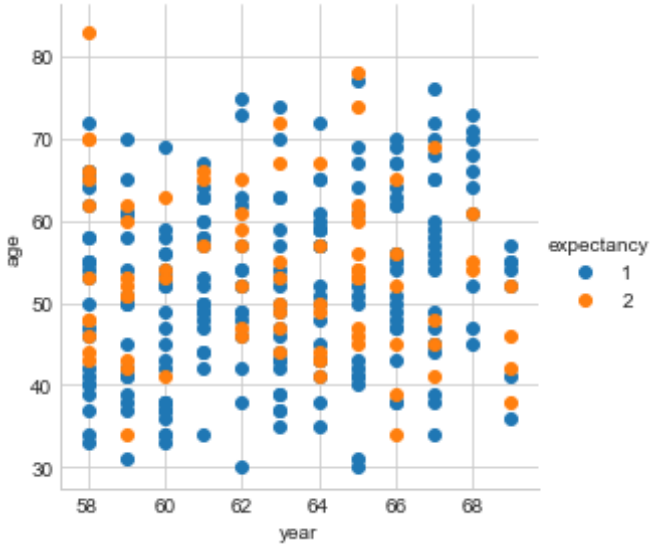
2-D Scatter plt (features: year and axil)

In [19]:
```
sns.set_style("whitegrid");
sns.FacetGrid(survival, hue="expectancy", size=4) \
   .map(plt.scatter, "year", "axil") \
   .add_legend();
plt.show();
```



2-D Scatter plt (features: year and age)

In [20]:
```
sns.set_style("whitegrid");
sns.FacetGrid(survival, hue="expectancy", size=4) \
   .map(plt.scatter, "year", "age") \
   .add_legend();
plt.show()
```



Obeservation:

1. These plots are not much informative as point are very overlaped and not linearly separable.

2-D Pair Plots

In [22]:
```
plt.close();
sns.set_style("whitegrid");
sns.pairplot(survival, hue="expectancy", size=3, vars=("age","axil","ye
ar"));
plt.show()
```

Obeservations:

1.These plots are not much informative as point are very overlaped and not linearly separable.

2.We can't find "lines" and "if-else" conditions to build a simple model to classify the classes.

Median, Quantiles and 90th percentile (Feature: Axilliary nodes)

In [31]:
```
print("\nMedians:")
print(np.median(Yes["axil"]))
print(np.median(No["axil"]))

print("\nQuantiles:")
print(np.percentile(Yes["axil"],np.arange(0, 100, 25)))
print(np.percentile(No["axil"],np.arange(0, 100, 25)))

print("\n90th Percentiles:")
print(np.percentile(Yes["axil"],90))
print(np.percentile(No["axil"],90))
```

```
Medians:
0.0
4.0

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

90th Percentiles:
8.0
20.0
```

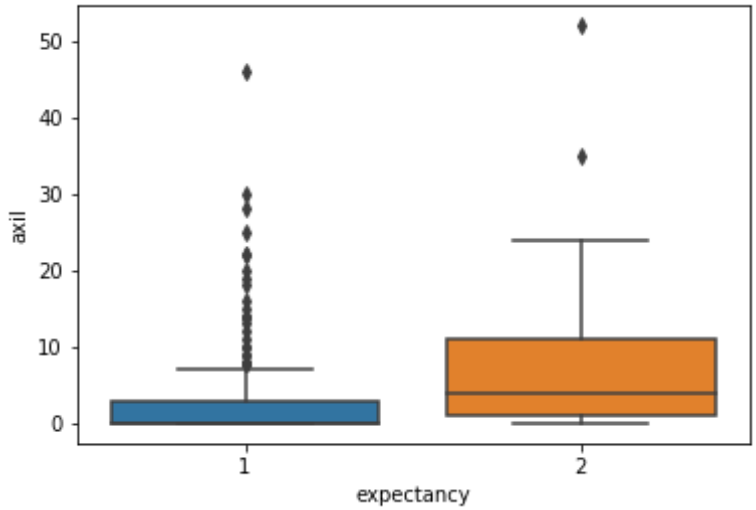
Median, Quantile and 90th Percentile (Feature : Patient's Age)

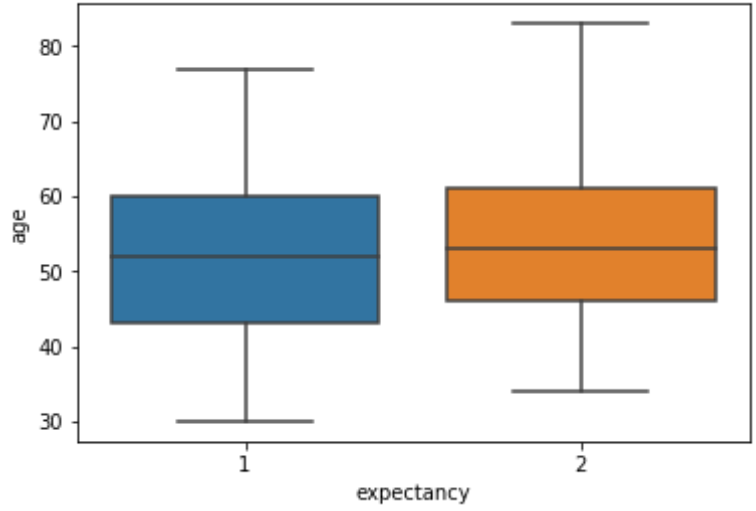
In [32]:
```
print("\nMedians:")
print(np.median(Yes["age"]))
print(np.median(No["age"]))
```

```
print("\nQuantiles:")
print(np.percentile(Yes["age"],np.arange(0, 100, 25)))
print(np.percentile(No["age"],np.arange(0, 100, 25)))

print("\n90th Percentiles:")
print(np.percentile(Yes["age"],90))
print(np.percentile(No["age"],90))
```

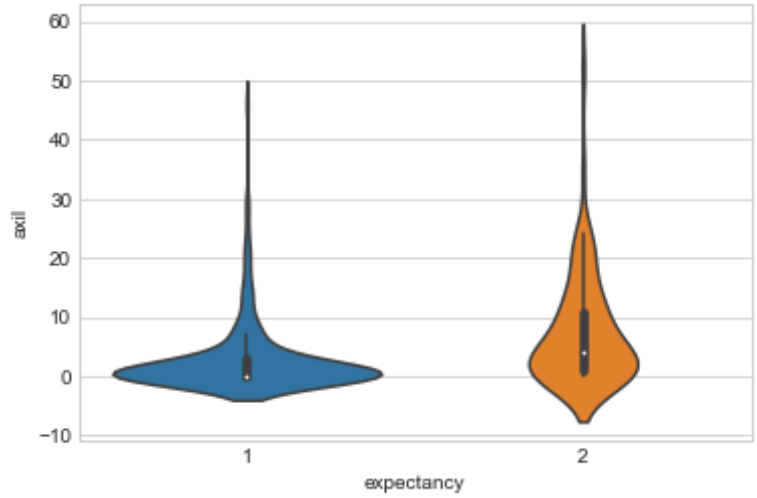
```
Medians:
52.0
53.0

Quantiles:
[30. 43. 52. 60.]
[34. 46. 53. 61.]

90th Percentiles:
67.0
67.0
```

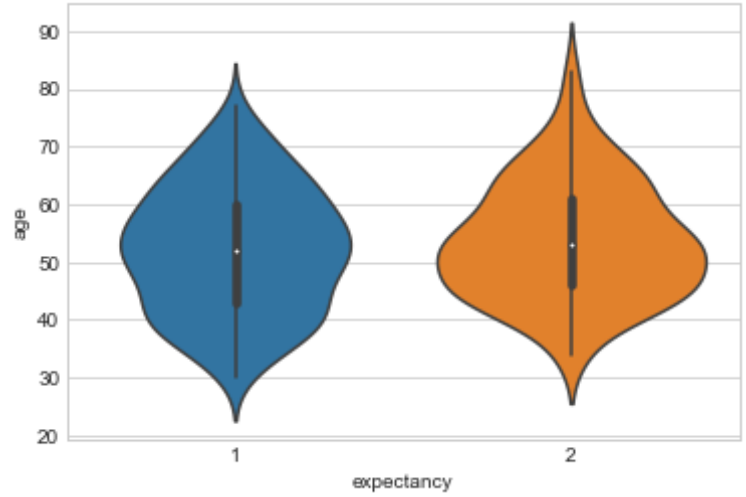
In [29]:
```
#Box plot (feature: Number of positive axilliary node)

sns.boxplot(x='expectancy',y='axil', data=survival)
plt.show()

sns.boxplot(x='expectancy',y='age', data=survival)
plt.show()

'''
Obeservation :
   1. axil>0.5 belongs to not survived class.
'''
```





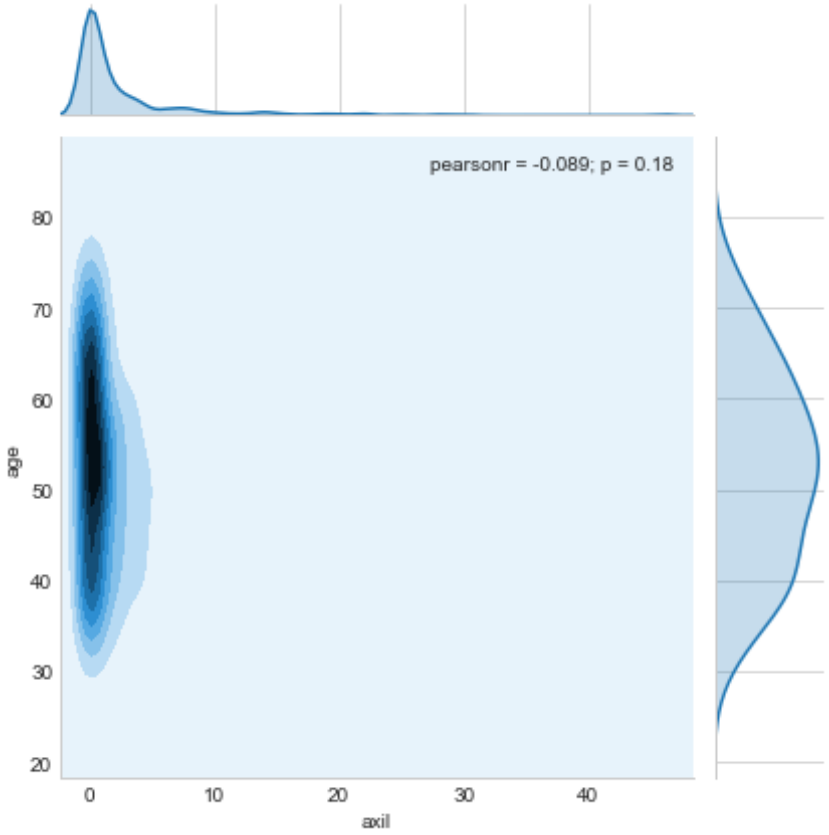
In [28]:
```
#Voilin Plot
sns.violinplot(x="expectancy", y="axil", data=survival, size=8)
plt.show()

sns.violinplot(x="expectancy", y="age", data=survival, size=8)
plt.show()
```





Multivariate probability density, contour plot.

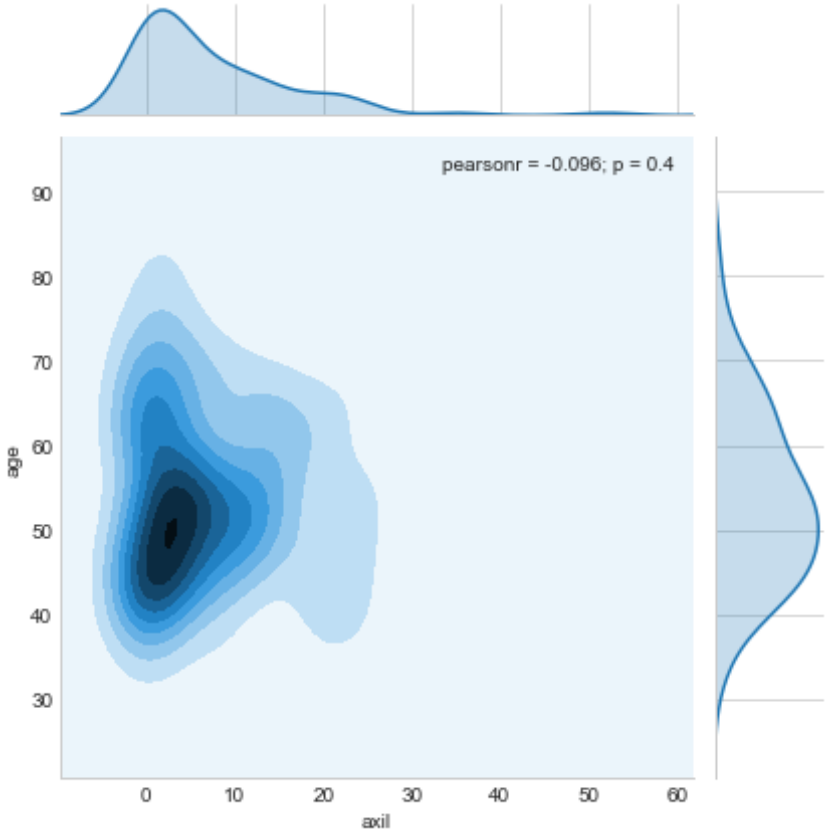
In [36]:
```
sns.jointplot(x="axil", y="age", data=Yes, kind="kde");
plt.show();
```



Obeservation:

1. At age of range 42 to 65 and having 0 positive axilliary nodes has huge desity of survived people.

In [37]:
```
sns.jointplot(x="axil", y="age", data=No, kind="kde");
plt.show();
```



Obeservation:

1. Small number of people of age of 48 to 51 and number of axiliary having nodes between 2 to 4 are not survived people

Conclusions:

1. The given data contains ~74% data of the patient survived.

2. The dataset is not linearly separable.

3. Graphs are higly overlaped, hence we cannot classify the classes perfectly.

4. The number of positive axilliary nodes is the most important feature.

5. Number of positive auxilliary nodes in the range 0 to 4 has more points belonging to '1'.

6. Number of positive auxilliary nodes in range 4 to 30 has more points belonging to '2'.

7. Maximum class '1' points occurs when no Positive Auxilliary Nodes found.

8. Person of Age(when they were operated) of range 40 to ~65 has survived the most.