EDA on Haberman Cacer survival data set

The dataset contains attributes information

1. Age of patient at time of operation (numerical)
2. Patient's year of operation (numerical)
3. Number of positive axillary nodes detcted (numerical)
4. Survival status 1 - patient survived 2 - patient not survived

objective : To classify whether patient survied or not. There are 3 features age, year and axillary nodes based on which we need to predict survival status whether patient is survied or not

In [18]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

In [19]:

```
#Load haberman datset
haberman = pd.read_csv("haberman.csv", names = ['age', 'year', 'axillarynodes', 'status'])
```

In [30]:

```
haberman.head()
```

Out[30]:

	age	year	axillarynodes	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

In [16]:
```
# How many data points and features
haberman.shape
```

Out[16]:
```
(306, 4)
```

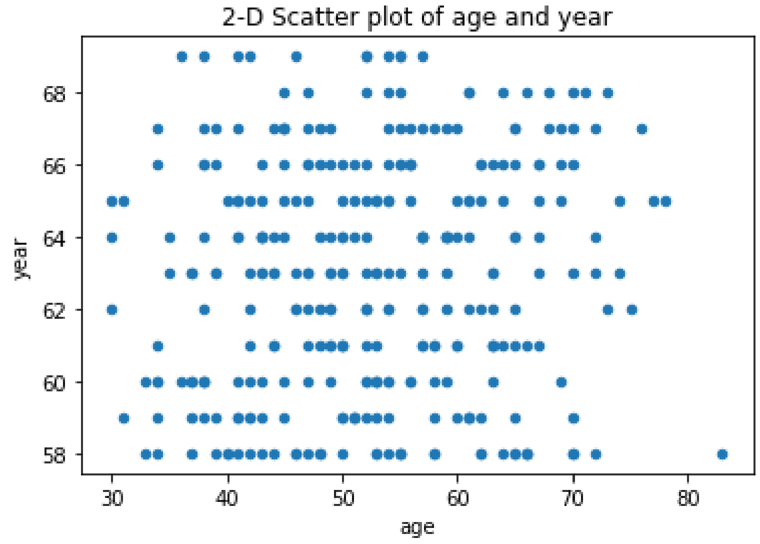
In [18]:
```
#How many data points for each status are present?
#How many patients survived and how many are not?
haberman["status"].value_counts()
```

Out[18]:
```
1    225
2     81
Name: status, dtype: int64
```

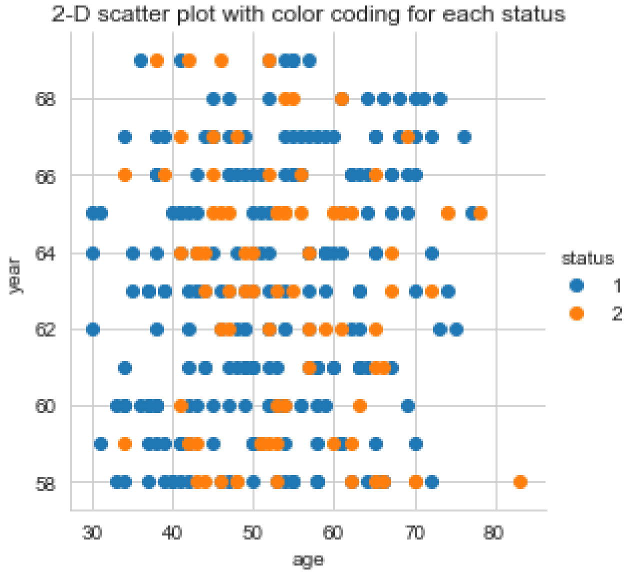
Based on above data, we can say it is imbalanced dataset, as the counts of patients who survived is no where matching with not survived

2-D Scatter Plot

In [20]:
```
haberman.plot(kind='scatter', x='age', y='year')
plt.title('2-D Scatter plot of age and year')
plt.show()

#with the below graph we cannot make any sense out of it
```



In [21]:
```
#2-D scatter plot with color coding for each status
sns.set_style("whitegrid");
sns.FacetGrid(haberman, hue="status", height=4)\
    .map(plt.scatter, "age", "year")\
    .add_legend();
plt.title('2-D scatter plot with color coding for each status')
plt.show();
```



Observations

Seperating survied and not survived data is harder since these data points are overlapping

Pair Plot

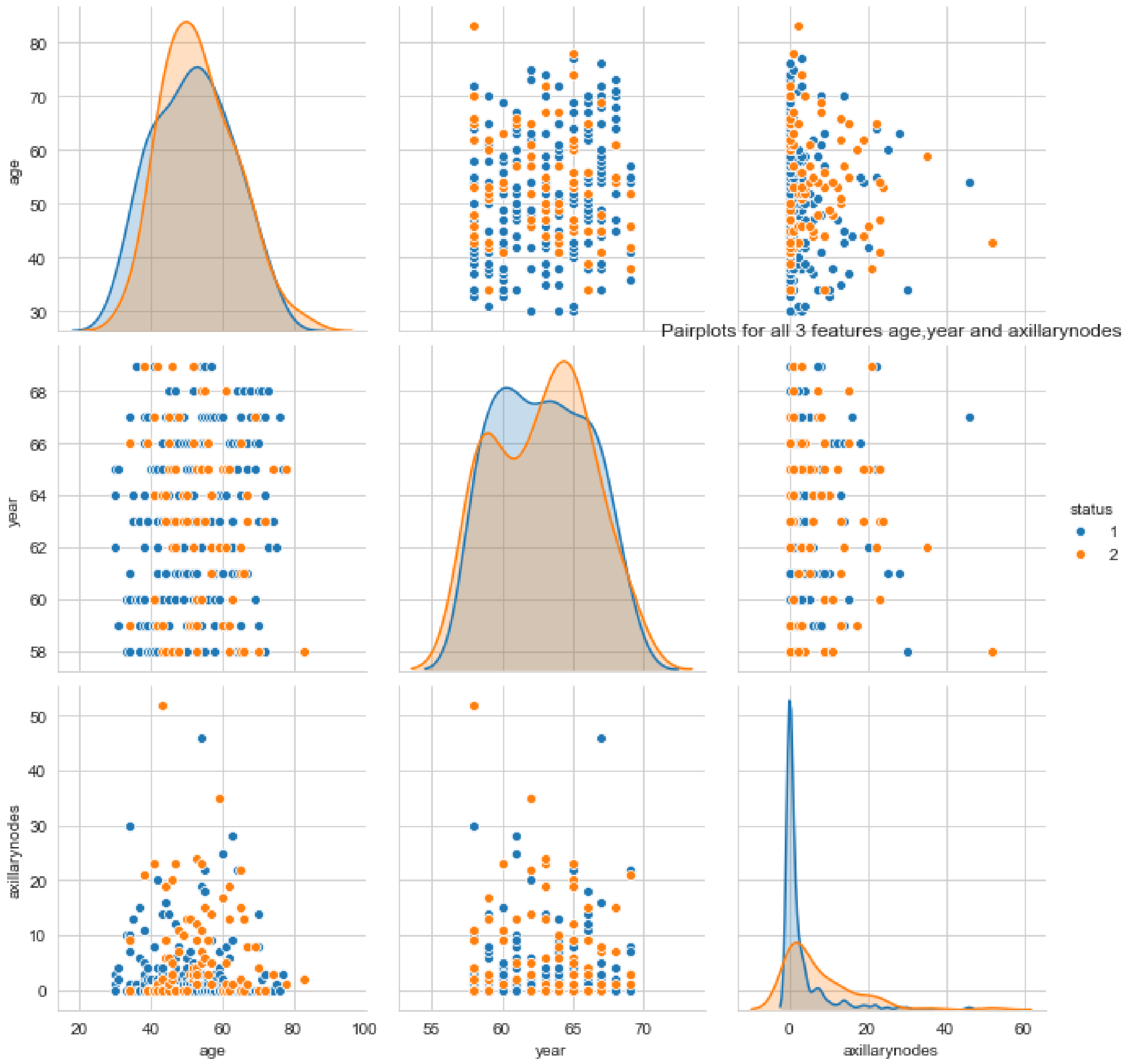
Since we cannot arrive any relationship with the features towards objective using 2-D scatterplots, so we are going for pairplots to identify some relationship with the features

In [22]:

```
plt.close()
```

In [22]:

```
sns.set_style("whitegrid");
sns.pairplot(haberman, hue="status", height=3);
plt.title('Pairplots for all 3 features age,year and axillarynodes')
plt.show()
```



Observations

```
1. axillarynodes is the most usful feature to identify whether
patient survived or not, Though they are overlapping
```

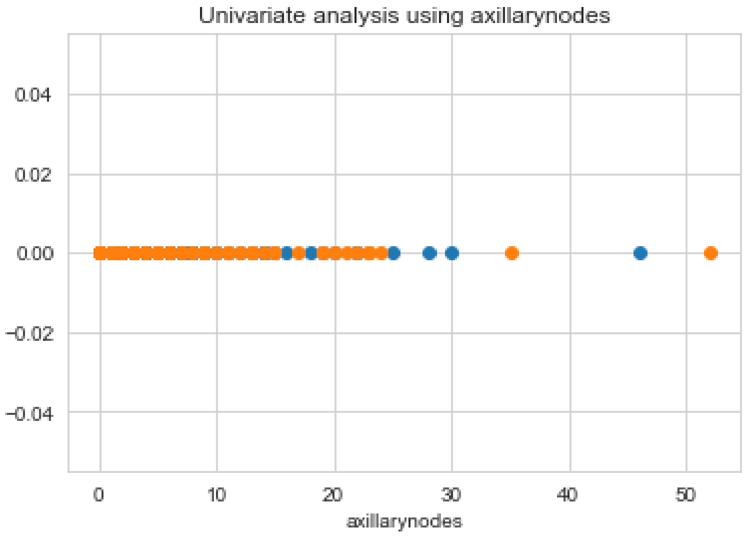
```
   a bit, but it can be analysed with some thresolds using PDF
and CDF

2. age, year both of these features are not lineraly sepearble
since they are overlapping more
```

Univariate Analysis using PDF

Since we have few features age,year and other important feature axillarynodes, we will see how to plot 1-D scatter plot for age, year and axillarynodes individually

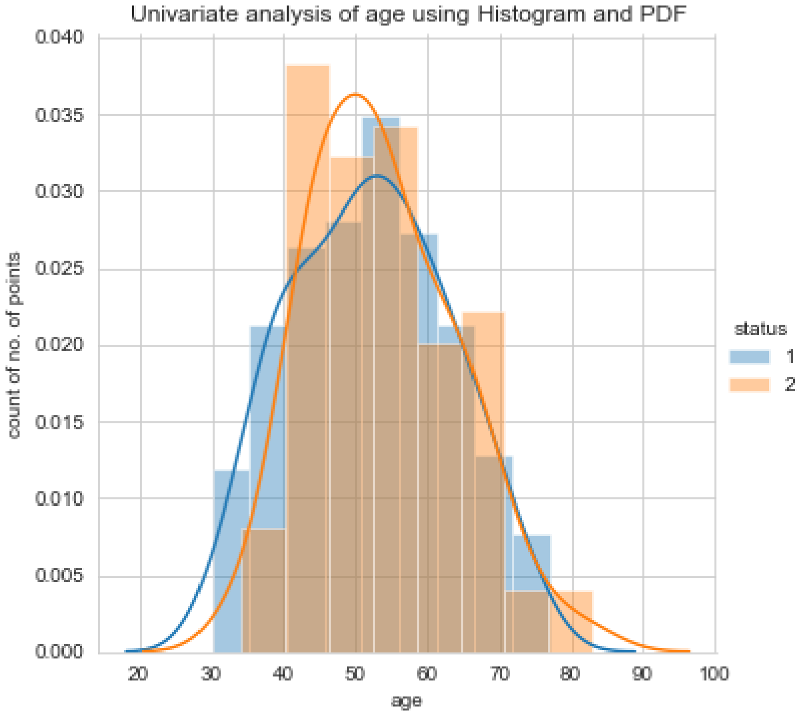
In [23]:

```
import numpy as np
haberman_survived = haberman.loc[haberman["status"] == 1];
haberman_notsurvived = haberman.loc[haberman["status"] == 2];

plt.plot(haberman_survived["axillarynodes"], np.zeros_like(haberman_sur
vived['axillarynodes']),'o')
plt.plot(haberman_notsurvived["axillarynodes"], np.zeros_like(haberman_
notsurvived['axillarynodes']),'o')
plt.xlabel('axillarynodes')
plt.title('Univariate analysis using axillarynodes')
plt.show()

#here the points are overlapping so we cannot  derive or differentiate
 any observations
```



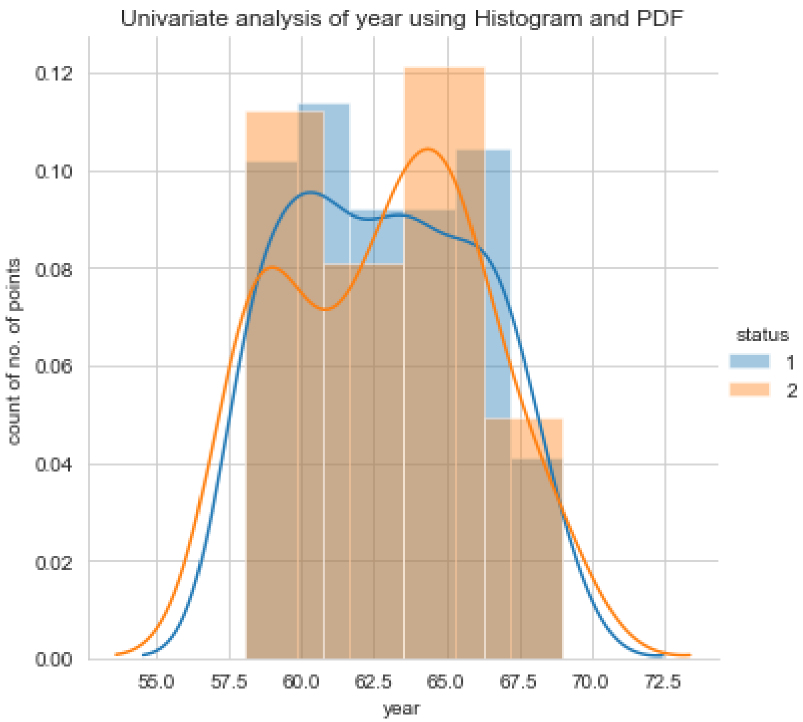
In [29]:

```
sns.FacetGrid(haberman, hue="status", height=5)\
    .map(sns.distplot, "age")\
    .add_legend();
plt.title('Univariate analysis of age using Histogram and PDF')
plt.ylabel('count of no. of points')
plt.show();
```



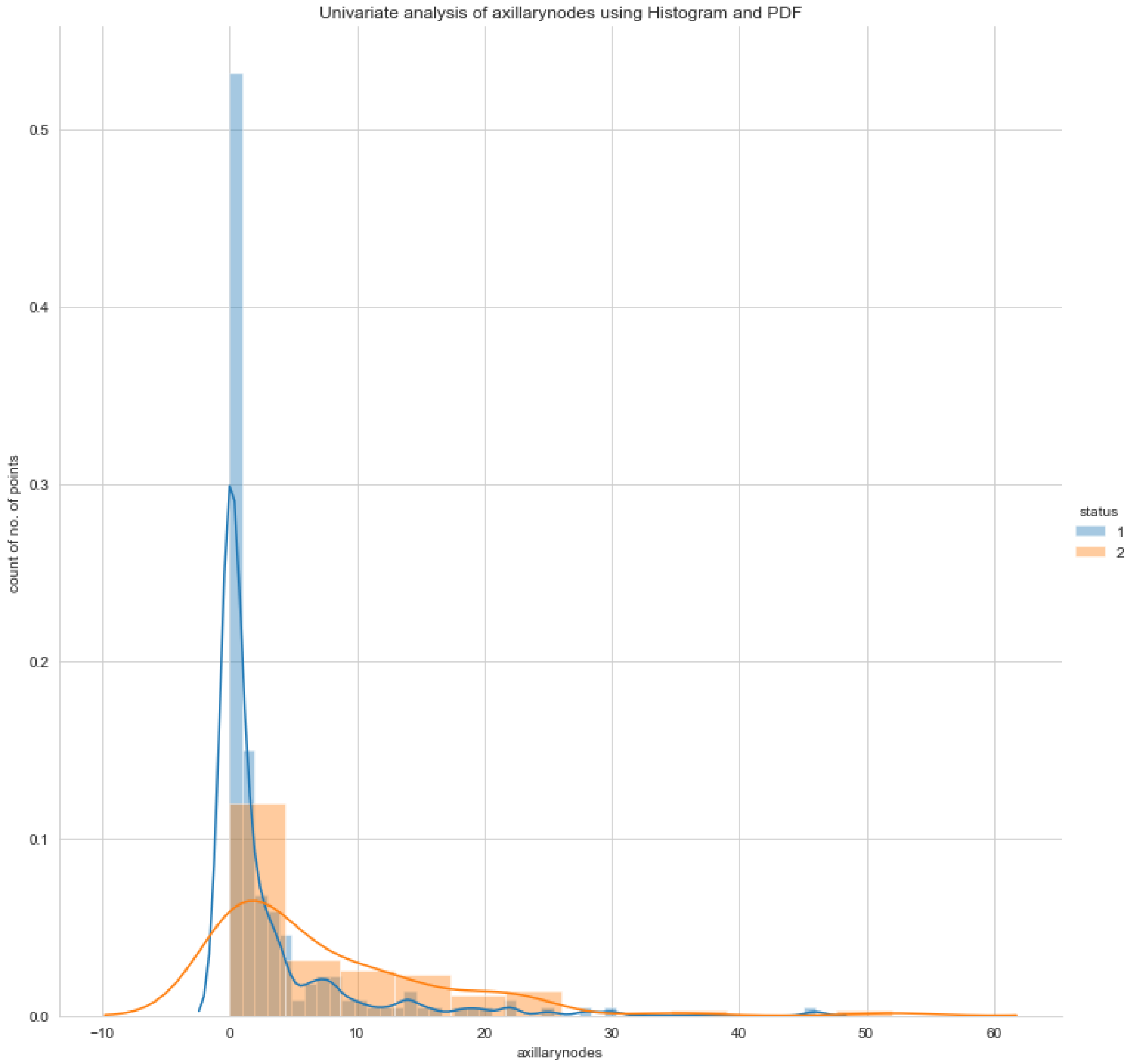
In [30]:

```
sns.FacetGrid(haberman, hue="status", height=5)\
    .map(sns.distplot, "year")\
    .add_legend();
plt.title('Univariate analysis of year using Histogram and PDF')
plt.ylabel('count of no. of points')
plt.show();
```



In [31]:

```
sns.FacetGrid(haberman, hue="status", height=10)\
    .map(sns.distplot, "axillarynodes")\
    .add_legend();
plt.title('Univariate analysis of axillarynodes using Histogram and PD
F')
plt.ylabel('count of no. of points')
plt.show();
```



Observations:

1. Once again it is proved that, age and year have massive overlapping compared to axillarynodes histogram. hence we can say that the histogram of axillarynodes is making sense compared to other histograms.
2. Most of the survived patients are having axillarynodes from 0 to 1
3. Most of the unsurvived patients(died) are having axillarynodes from 0 to 5

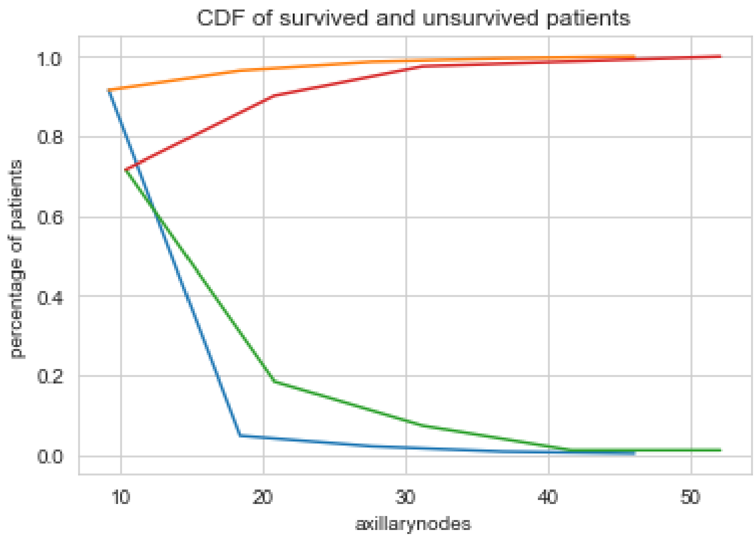
In [34]:

```
counts, bin_edges = np.histogram(haberman_survived['axillarynodes'], bi
ns=5,
                                 density = True)
plt.title('CDF of survived and unsurvived patients')
plt.xlabel('axillarynodes')
plt.ylabel('percentage of patients')
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)

# notsurvived
counts, bin_edges = np.histogram(haberman_notsurvived['axillarynodes'],
 bins=5,
                                 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)
```

```
[0.91555556 0.04888889 0.02222222 0.00888889 0.00444444]
[ 0.   9.2 18.4 27.6 36.8 46. ]
[0.71604938 0.18518519 0.07407407 0.01234568 0.01234568]
[ 0.  10.4 20.8 31.2 41.6 52. ]
```

Out[34]:

```
[<matplotlib.lines.Line2D at 0x1da0f4c6ec8>]
```



Observations:

Here if we take a thresold value of intersection point 13 , when axillarynodes<=13, then aprox 90% times it is correct that patient can survive, 10% chances patient may die. also when axillarynodes>13, 60% times it is correct that patient can die, 40% chances that patient may survive.

Majority of patients survived when they have axillarynodes<=30

Mean, Variance and Std-dev

In [5]:
```
print("Means: ")
print(np.mean(haberman_survived["axillarynodes"]))
print(np.mean(haberman_notsurvived["axillarynodes"]))

print("\nStd-dev: ")
print(np.std(haberman_survived["axillarynodes"]))
print(np.std(haberman_notsurvived["axillarynodes"]))
```

```
Means:
2.7911111111111113
7.45679012345679

Std-dev:
5.857258449412131
9.128776076761632
```

Average deviation of points from mean is std-dev, For patients who are survived, the spread of axillarynodes is small For patients who are died, the spread of axillarynodes is large

Majority of survived patients are having axillarynodes in the ranges 2.79-5.85 to 2.79+5.85) Majority of survived patients are having axillarynodes in the ranges 7.45-9.12 to 7.45+9.12)

Median, Percentile, Quantile, IQR, MAD

Mean/Std-dev can be corrupted easily by a small outlier, hence to overcome this we will calculate Median,Percentile, Quantile and MAD to

In [11]:

```
print("\n Medians: ")
print(np.median(haberman_survived["axillarynodes"]))
#Median with an outlier
print(np.median(np.append(haberman_survived["axillarynodes"],50)))
print(np.median(haberman_notsurvived["axillarynodes"]))
```

```
 Medians: 
0.0
0.0
4.0
```

Though we have an outlier, the median has not corrupted, the value of median is same with outlier and without outlier

In [7]:

```
print("\n Quantiles: ")
print(np.percentile(haberman_survived["axillarynodes"],np.arange(0, 100
, 25)))
print(np.percentile(haberman_notsurvived["axillarynodes"],np.arange(0,
100, 25)))
```

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

In [8]:

```
print("\n 90th Percentiles: ")
print(np.percentile(haberman_survived["axillarynodes"],90))
print(np.percentile(haberman_notsurvived["axillarynodes"],90))
```

```
 90th Percentiles: 
8.0
20.0
```

90% percentage of survived patients are having axillarynodes in the range of 0 to 8 90% of died patients are having axillarynodes in the range of 0 to 20

In [10]:

```
from statsmodels import robust
print("\n Median Absolute Deviation")
print(robust.mad(haberman_survived["axillarynodes"]))
print(robust.mad(haberman_notsurvived["axillarynodes"]))
```

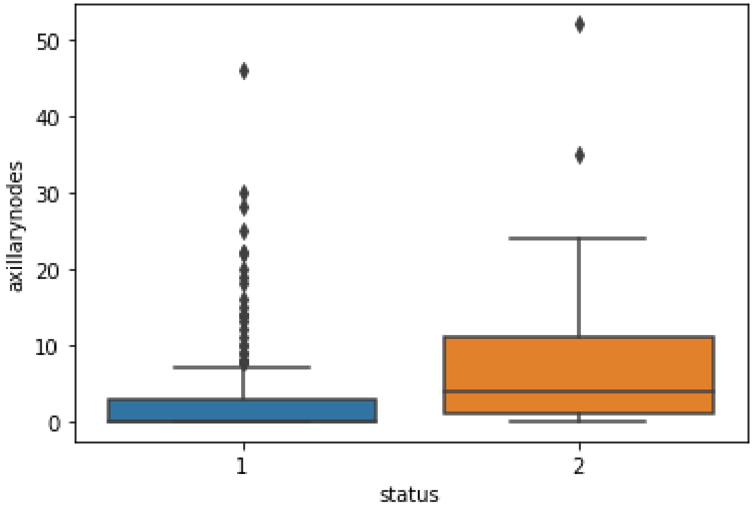
```
 Median Absolute Deviation
0.0
5.930408874022408
```

For majority of survived patients, The average variablility/spread of axillarynodes are 0 For majority of unsurvived patients, The average variablility/spread of axillarynodes are in the range from mode of(4.0-5.9) till (4.0+5.9)

Box plot and Whiskers

We can get 75%,25%,50% percentage values by CDF but we need to draw horizontal and vertical lines to find these values. To avoid drawing horizontal and vertical lines, and to find these percentage values we will draw box and whiskers

In [16]:
```
sns.boxplot(x='status', y='axillarynodes', data=haberman)
plt.show()
```



Observations:

For majority of survived patients, axillarynodes are between min=0, max<=8, rest all are outliers 50 percentile of axillary nodes are between 0 to 4

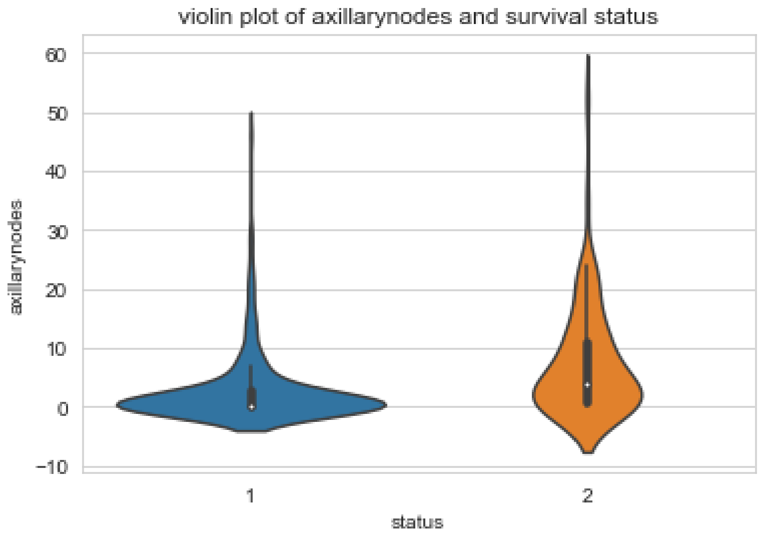
For majority of died patients, axillarynodes are between min=0, max<=25, rest all are outliers 25 percentile of axillary nodes are at 1 50 percentile of axillary nodes are at 5 75 percentile of axillary nodes are at 10

Violin plots

Combination of PDF and histogram

In [35]:

```
sns.violinplot(x="status", y="axillarynodes", data=haberman, height=8)
plt.title('violin plot of axillarynodes and survival status')
plt.show()
```



Summary of plots

Observations:

For majority of survived patients, axillary nodes are from min=0 and max=8 50 percentage of them having axillary nodes are 0 75 percetage of them having axillary nodes 5

hence, we can say that if a patient is having axillary nodes <=8 then probability of survival is more

For majority of unsurvived/died patients, axillary nodes are from min=0 and max=25 50 percentage of them having axillary nodes are 5 75 percetage of them having axillary nodes are 10

hence, we can say that if a patient is having axillary nodes >=10 then probability of survival is less

In [ ]: