Exploratory data analysis (EDA) on Haberman's Survival dataset

About the Dataset

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

Number of Instances (rows): **306**

Number of Attributes (columns): **4** (including the class attribute and all the attributes have **numeric** values)

Attribute Information

1. **Age** of patient at time of operation
2. Patient's **year** of operation
3. Number of positive axillary **nodes** detected
   (**Positive axillary nodes** means the lymph nodes in the underarm area do contain cancer.
   These are also the first lymph nodes where breast cancer is likely to spread.)
4. Survival **status** (class attribute)
   1 = the patient survived 5 years or longer
   2 = the patient died within 5 year

Missing Attribute Values: None

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

OBJECTIVE

Classify a new patient who had undergone breast cancer surgery as belonging to one of the 2 classes i.e, (patient will survive more than 5 years or die within 5 years) based on the 3 features (age, year, nodes).

1. importing libraries

In [30]:
```
# importing the required python libraries

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

2. Loading dataset

In [31]:
```
#loading data to pandas dataframe
haberman_df = pd.read_csv('haberman.csv')
```

3. Basic analysis of df

In [32]:
```
#to view the dataframe
haberman_df
```

Out[32]:

	age	year	nodes	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
...	...	...	...	...
301	75	62	1	1
302	76	67	0	1
303	77	65	3	1
304	78	65	1	2
305	83	58	2	2

306 rows × 4 columns

Observations :

difficult to understand the **status** of patient, as the class attribute holds numerical values
1 = the patient **survived** 5 years or longer
2 = the patient **died** within 5 year

In [33]:
```
# replacing 1 as survived and 2 as died using map()
haberman_df['status']= haberman_df['status'].map({1:'survived',2:'died'})
haberman_df
```

Out[33]:

	age	year	nodes	status
0	30	64	1	survived
1	30	62	3	survived
2	30	65	0	survived
3	31	59	2	survived
4	31	65	4	survived
...	...	...	...	...
301	75	62	1	survived
302	76	67	0	survived
303	77	65	3	survived
304	78	65	1	died
305	83	58	2	died

306 rows × 4 columns

In [34]:
```
#Checking the column names
haberman_df.columns
```

Out[34]:
```
Index(['age', 'year', 'nodes', 'status'], dtype='object')
```

In [35]:
```
#Checking the dimensions(rows,columns)
haberman_df.shape
```

Out[35]:
```
(306, 4)
```

In [36]:
```
#returns first 5 rows of the df
haberman_df.head()
```

Out[36]:

	age	year	nodes	status
0	30	64	1	survived
1	30	62	3	survived
2	30	65	0	survived
3	31	59	2	survived
4	31	65	4	survived

In [37]:
```
#returns last 5 rows of the df
haberman_df.tail()
```

Out[37]:

	age	year	nodes	status
301	75	62	1	survived
302	76	67	0	survived
303	77	65	3	survived
304	78	65	1	died
305	83	58	2	died

4. High level statistics of df

In [38]:
```
#to get a concise summary of the df
haberman_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306 entries, 0 to 305
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   age     306 non-null    int64 
 1   year    306 non-null    int64 
 2   nodes   306 non-null    int64 
 3   status  306 non-null    object
dtypes: int64(3), object(1)
memory usage: 9.7+ KB
```

Observations :

- **no null values present in the df,** as the no.of non-null counts of each attributes are equal to the no.of instances(306)
- **all the features have numeric values,** as the datatype of each features belongs to integer.
- **class attribute have non numeruc values,** as its datatype belongs to object.

In [39]:
```
#computes a summary statistics of numeric data
haberman_df.describe()
```

Out[39]:

	age	year	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

Observations :

- **age** of patients ranges from 30 to 83 (min, max) with an average age of 52 (mean) and deviation of age from mean is 10 (std)
- **year** of operation between 1958 - 1969 (min, max)
- **nodes** - highest positive axillary nodes of patient is 52(max) but 75% patient has less than 4 nodes and 25% has no nodes

In [40]:
```
#Counting number of datapoints in each class
haberman_df['status'].value_counts()
```

Out[40]:
```
survived    225
died         81
Name: status, dtype: int64
```

Observations:

haberman_df is a **imbalanced dataset**, as the number of data points for the 2 classes are not equal

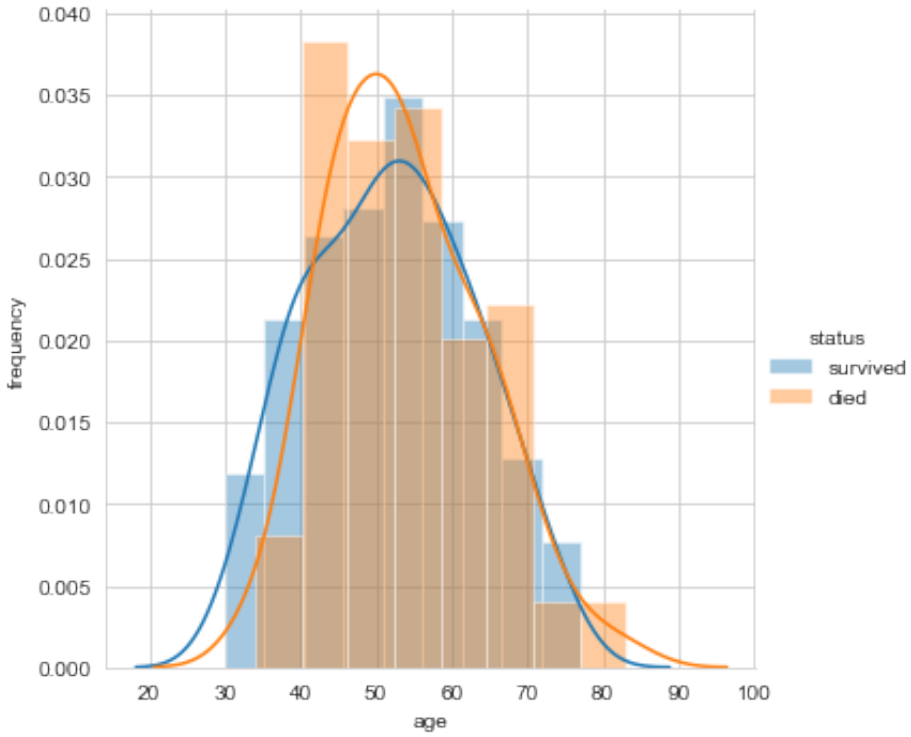
5. Univaraite analysis

univaraite analysis (Distribution plots, PDF and CDF, Boxplot, Voilin plots) is performed to understand which features are useful towards classification.

5.1 Distribution plots

The Seaborn module along with the Matplotlib module is used to depict the **distplot**, which shows the variation in the data distribution through histogram and density line on it.

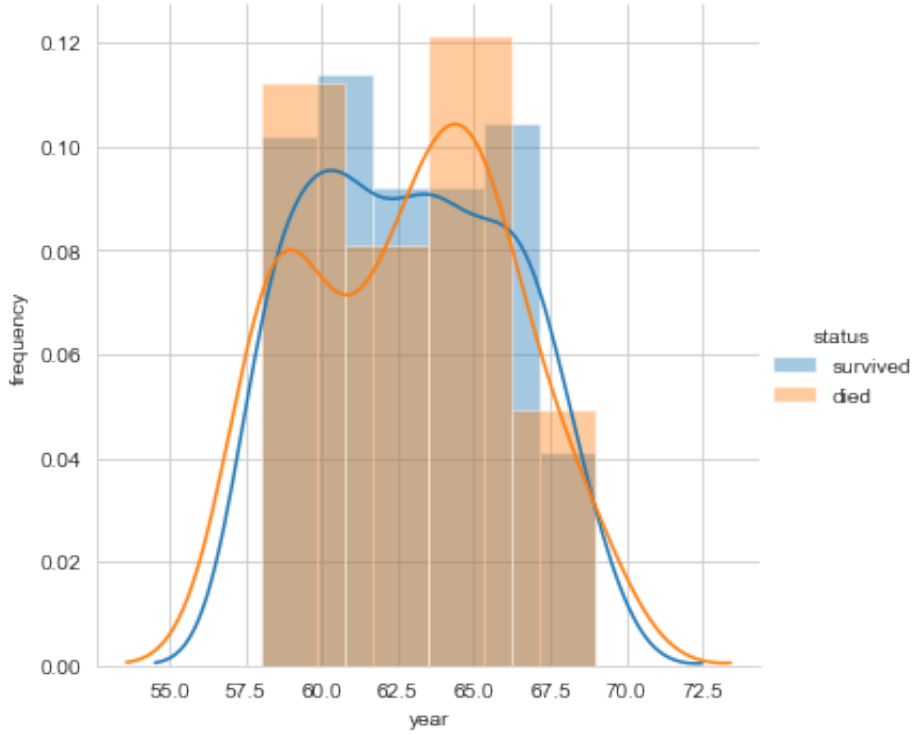
In [41]:
```
#Univariate analysis on the "age" variable using distplot
sns.FacetGrid(haberman_df, hue='status', height=5) \
   .map(sns.distplot, "age") \
   .add_legend()
plt.ylabel('frequency')
plt.show()
```



Observations:

- patients with age between 30 and 34 have survived more than 5 years
- patients with age between 77 and 83 have died with in 5 years
- for the rest of the patients, can't conclude anything from the plot as points are overlapping.

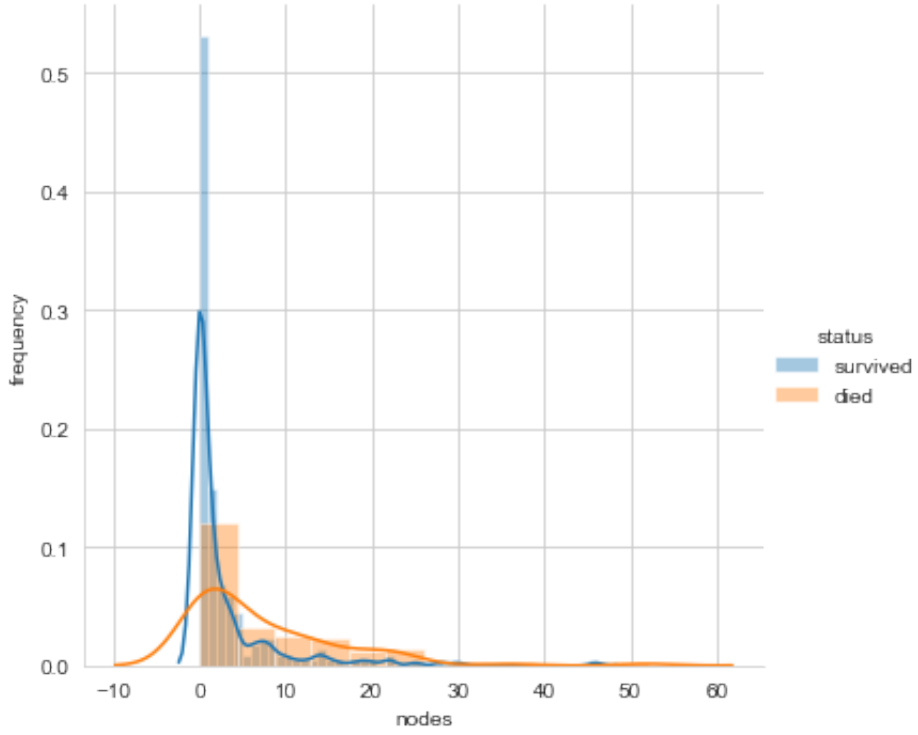
In [42]:
```
#Univariate analysis on the "year" variable using distplot
sns.FacetGrid(haberman_df, hue='status', height=5) \
   .map(sns.distplot, "year") \
   .add_legend()
plt.ylabel('frequency')
plt.show()
```



Observations:

- can't conclude anything from the plot as points are much overlapping , so **year of operation** not useful in classifying the patient.

In [43]:
```
#Univariate analysis on the "nodes" variable using distplot
sns.FacetGrid(haberman_df, hue='status', height=5) \
   .map(sns.distplot, "nodes") \
   .add_legend()
plt.ylabel('frequency')
plt.show()
```



Observations:

- eventhough overlapping seen in this plot, but it clearly shows that patients with less number of postive axillary nodes have survived more than 5 years.

Hence from distplot, **nodes** seems to be much useful in classifying the patient's survival status than other features.

5.2 PDF and CDF

PDF and CDF are used to find the probabilistic relations between the variables.

PDF (Probability Denstiy Funtion)

- finds the density of probability for continues random variable
- to understand the distribution of data visually without knowing the exact probability for a certain range of values.

CDF (Cummulative Distributive Function)

- finds the cumulative probability of random variables either it is continuous or discrete
- to determine the probability that a random varible that is taken from the population will be less than or equal to a certain value.

In [44]:
```
#creating dataframe of patients who survived
survived_df = haberman_df.loc[haberman_df["status"]== 'survived']

#creating dataframe of patients who died
died_df = haberman_df.loc[haberman_df["status"]== 'died']
```

In [45]:
```
#Plotting PDF abd CDF for age of patients who survived
counts, bin_edges = np.histogram(survived_df['age'], bins=10, densi
ty = True)
pdf = counts/(sum(counts))
print("pdf_survived :\n",pdf);
print('bin_edges_survived :\n',bin_edges)
cdf = np.cumsum(pdf)
print("cdf_survived :\n=",cdf);
plt.plot(bin_edges[1:],pdf,label='pdf_survived')
plt.plot(bin_edges[1:],cdf,label='cdf_survived')

print('*'*60)

#Plotting PDF abd CDF for age of patients who died
counts, bin_edges = np.histogram(died_df['age'], bins=10, density =
True)
pdf = counts/(sum(counts))
print("pdf_died :\n", pdf);
print('bin_edges_died:\n',bin_edges)
cdf = np.cumsum(pdf)
print("cdf_died:\n =",cdf);
plt.plot(bin_edges[1:],pdf, label='pdf_died')
plt.plot(bin_edges[1:], cdf, label='cdf_died')

plt.xlabel('age')
plt.ylabel('probability')
plt.title('PDF and CDF of age')
plt.legend()
plt.show()
```

```
pdf_survived :
 [0.05333333 0.10666667 0.12444444 0.09333333 0.16444444 0.1644444
4
 0.09333333 0.11111111 0.06222222 0.02666667]
bin_edges_survived :
 [30.  34.7 39.4 44.1 48.8 53.5 58.2 62.9 67.6 72.3 77. ]
cdf_survived :
= [0.05333333 0.16       0.28444444 0.37777778 0.54222222 0.706666
67
 0.8        0.91111111 0.97333333 1.        ]
**********************************************************
pdf_died :
 [0.03703704 0.12345679 0.19753086 0.19753086 0.13580247 0.1234567
9
 0.09876543 0.04938272 0.02469136 0.01234568]
bin_edges_died:
 [34.  38.9 43.8 48.7 53.6 58.5 63.4 68.3 73.2 78.1 83. ]
cdf_died:
 = [0.03703704 0.16049383 0.35802469 0.55555556 0.69135802 0.81481
481
 0.91358025 0.96296296 0.98765432 1.        ]
```

Observations:

From pdf

- patients with the age between 45 and 55 have died more than people who survived.

From cdf

- Patient with age less than 45 has more probability of surviving than the probability of dying within 5 years

From pdf and cdf

- No patients with the age between 30 and 34 have died within 5 years
- No patients with the age between 77 and 83 have survived more than 5 years

But **age** not seems to be much useful in classifying the patient as pdf and cdf plot of both classes are found to be overlapped.

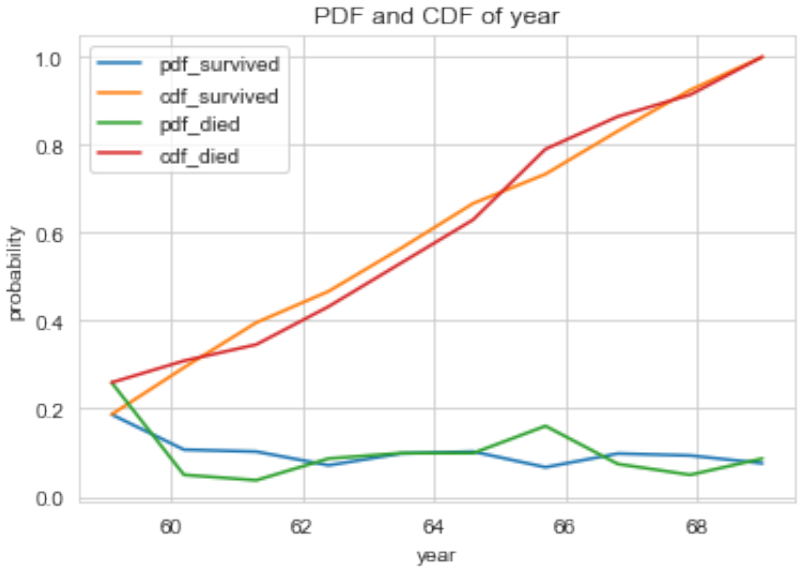
In [46]:
```
#Plotting PDF abd CDF for year of operation of patients who survive
d
counts, bin_edges = np.histogram(survived_df['year'], bins=10, dens
ity = True)
pdf = counts/(sum(counts))
print("pdf_survived :\n",pdf);
print('bin_edges_survived :\n',bin_edges)
cdf = np.cumsum(pdf)
print("cdf_survived :\n=",cdf);
plt.plot(bin_edges[1:],pdf,label='pdf_survived')
plt.plot(bin_edges[1:], cdf, label='cdf_survived')

print('*'*60)

#Plotting PDF abd CDF for year of operation of patients who died
counts, bin_edges = np.histogram(died_df['year'], bins=10, density
= True)
pdf = counts/(sum(counts))
print("pdf_died :\n", pdf);
print('bin_edges_died:\n',bin_edges)
cdf = np.cumsum(pdf)
print("cdf_died:\n =",cdf);
plt.plot(bin_edges[1:],pdf, label='pdf_died')
plt.plot(bin_edges[1:], cdf, label='cdf_died')

plt.xlabel('year')
plt.ylabel('probability')
plt.title('PDF and CDF of year')
plt.legend()
plt.show()
```

```
pdf_survived :
 [0.18666667 0.10666667 0.10222222 0.07111111 0.09777778 0.1022222
2
 0.06666667 0.09777778 0.09333333 0.07555556]
bin_edges_survived :
 [58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
cdf_survived :
= [0.18666667 0.29333333 0.39555556 0.46666667 0.56444444 0.666666
67
 0.73333333 0.83111111 0.92444444 1.        ]
**********************************************************
pdf_died :
 [0.25925926 0.04938272 0.03703704 0.08641975 0.09876543 0.0987654
3
 0.16049383 0.07407407 0.04938272 0.08641975]
bin_edges_died:
 [58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
cdf_died:
 = [0.25925926 0.30864198 0.34567901 0.43209877 0.5308642  0.62962
963
 0.79012346 0.86419753 0.91358025 1.        ]
```



Observations:

From pdf

- patients who has been operated between 1960 and 1962 have survived more than people who died.
- patients who has been operated between 1965 and 1967 have died more than people who survived.

From cdf

- does not depicts any valuable inference as it overlaps for both survival status.

Hence, from both pdf and cdf plot, **year** is also not seems to be useful in classifying the patient.

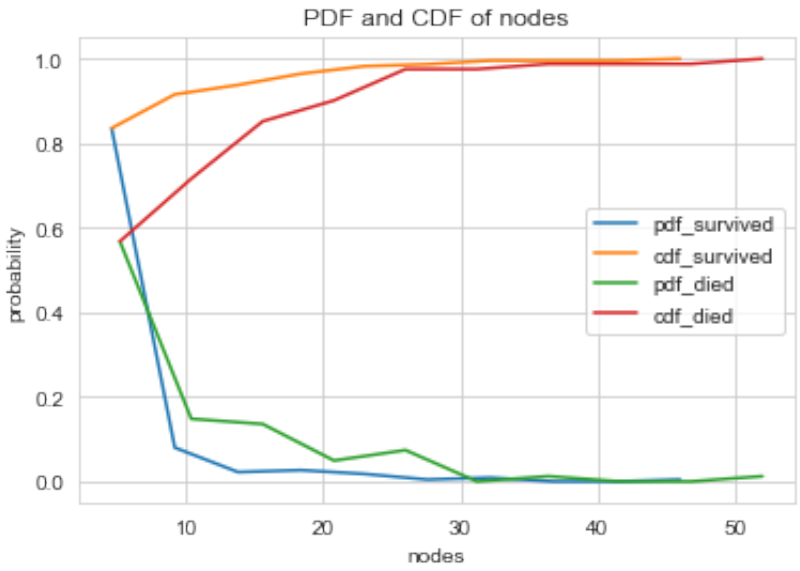
In [47]:
```
#Plotting PDF abd CDF for no.of positive axillary nodes of patients
who survived
counts, bin_edges = np.histogram(survived_df['nodes'], bins=10, den
sity = True)
pdf = counts/(sum(counts))
print("pdf_survived :\n",pdf);
print('bin_edges_survived :\n',bin_edges)
cdf = np.cumsum(pdf)
print("cdf_survived :\n=",cdf);
plt.plot(bin_edges[1:],pdf,label='pdf_survived')
plt.plot(bin_edges[1:], cdf, label='cdf_survived')

print('*'*60)

#Plotting PDF abd CDF for no.of positive axillary nodes of patients
who died
counts, bin_edges = np.histogram(died_df['nodes'], bins=10, density
= True)
pdf = counts/(sum(counts))
print("pdf_died :\n", pdf);
print('bin_edges_died:\n',bin_edges)
cdf = np.cumsum(pdf)
print("cdf_died:\n =",cdf);
plt.plot(bin_edges[1:],pdf, label='pdf_died')
plt.plot(bin_edges[1:], cdf, label='cdf_died')

plt.xlabel('nodes')
plt.ylabel('probability')
plt.title('PDF and CDF of nodes')
plt.legend()
plt.show()
```

```
pdf_survived :
 [0.83555556 0.08       0.02222222 0.02666667 0.01777778 0.0044444
4
 0.00888889 0.         0.         0.00444444]
bin_edges_survived :
 [ 0.   4.6  9.2 13.8 18.4 23.  27.6 32.2 36.8 41.4 46. ]
cdf_survived :
= [0.83555556 0.91555556 0.93777778 0.96444444 0.98222222 0.986666
67
 0.99555556 0.99555556 0.99555556 1.        ]
**********************************************************
pdf_died :
 [0.56790123 0.14814815 0.13580247 0.04938272 0.07407407 0.
 0.01234568 0.         0.         0.01234568]
bin_edges_died:
 [ 0.   5.2 10.4 15.6 20.8 26.  31.2 36.4 41.6 46.8 52. ]
cdf_died:
 = [0.56790123 0.71604938 0.85185185 0.90123457 0.97530864 0.97530
864
 0.98765432 0.98765432 0.98765432 1.        ]
```



Observations:

From pdf and cdf

- For the patients having less than 5 positive axillary nodes, the probability of surviving (~82%) is higher than the probability of dying (~58%).
- No patients having more than 46 nodes have survived more than 5 years.

The PDF and CDF plot for 'nodes' of both srvival status are also overlapping, but it clearly shows that the patients with less number of postive axillary nodes have more chances of survival.

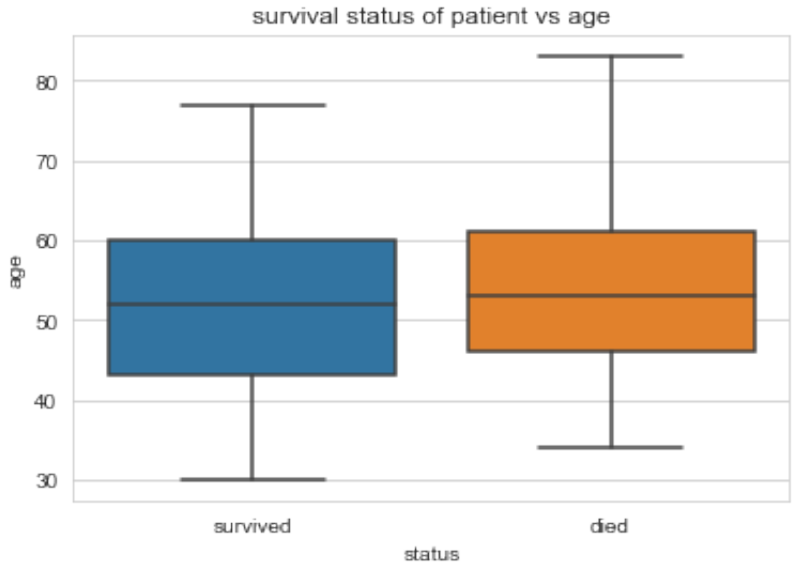
Hence from all PDF and CDF plot, nodes seems to be much useful in classifying the patient's survival status than other features.

5.3 Boxplot

A box plot is a useful way to illustrate the central tendency, variability, and skewness of a distribution and also an excellent way to detect outliers and extreme values.

- box represents the interquartile range(IQR)
- top line in the box is the 75th percentile(3rd quartile)
- center line in the box is the 50th percentile(median).
- bottom line in the box is the 25th percentile(1st quartile)
- Whiskers extend from the box to the left and right.
- the left and right fences represent the minimum and maximum value.

In [48]:
```
#boxplot for survival status of patient vs age
sns.boxplot(x='status',y='age', data=haberman_df)
plt.title('survival status of patient vs age')
plt.show()
```

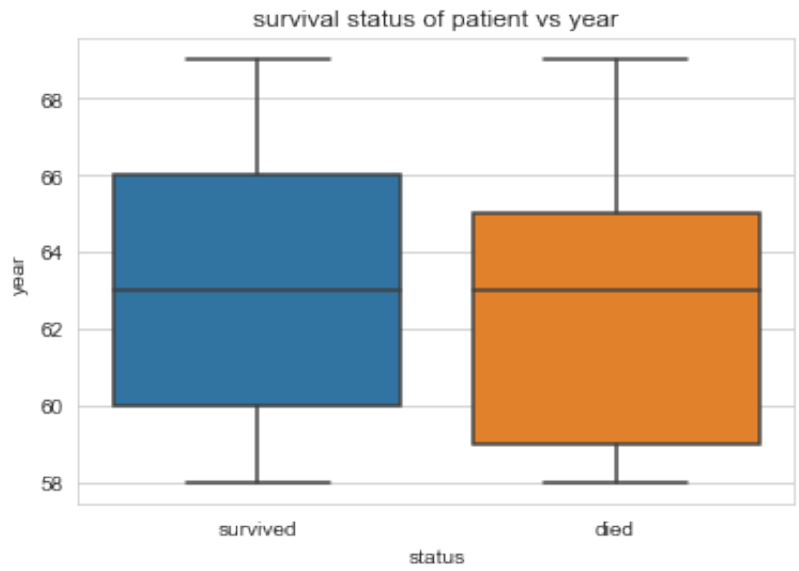


Observations:

- 50% of patient's age who survived more than 5 years are ranges from 43 to 60 (IQR)
- 50% of patient's age who died within 5 years are ranges from 46 to 61 (IQR)

From the box plot we found that ~75% of the age of patients overlapped for both survival status. Hence, **age** is not much useful in classifying the patient.

In [49]:
```
#boxplot for survival status of patient vs year
sns.boxplot(x='status',y='year', data=haberman_df)
plt.title('survival status of patient vs year')
plt.show()
```

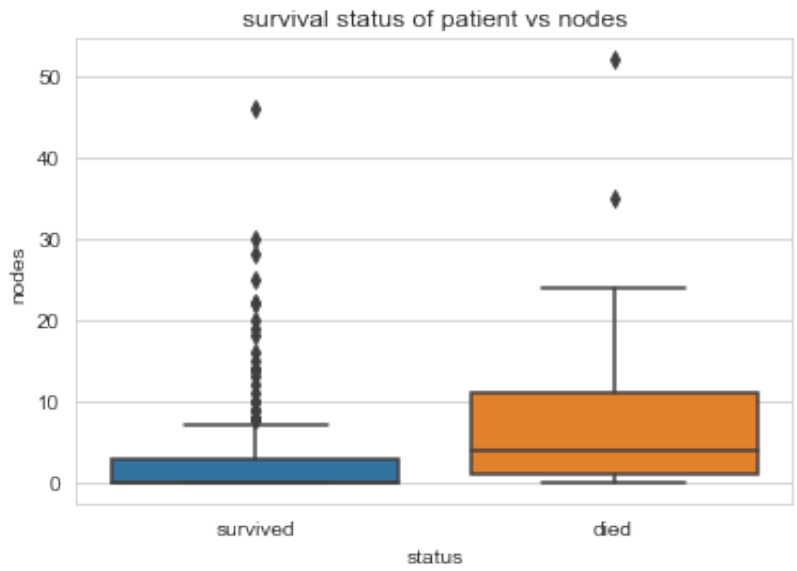


Observations:

- 50% operations happened in between 1960 to 1966 for patients who survived more than 5 years (IQR)
- 50% operations happened in between 1959 to 1965 for patients who died within 5 years (IQR)

From the box plot we found that ~75% of the year of operations overlapped for both survival status. Hence, **year** is also not much useful in classifying the patient.

In [50]:
```
#boxplot for survival status of patient vs nodes
sns.boxplot(x='status',y='nodes', data=haberman_df)
plt.title('survival status of patient vs nodes')
plt.show()
```



Observations:

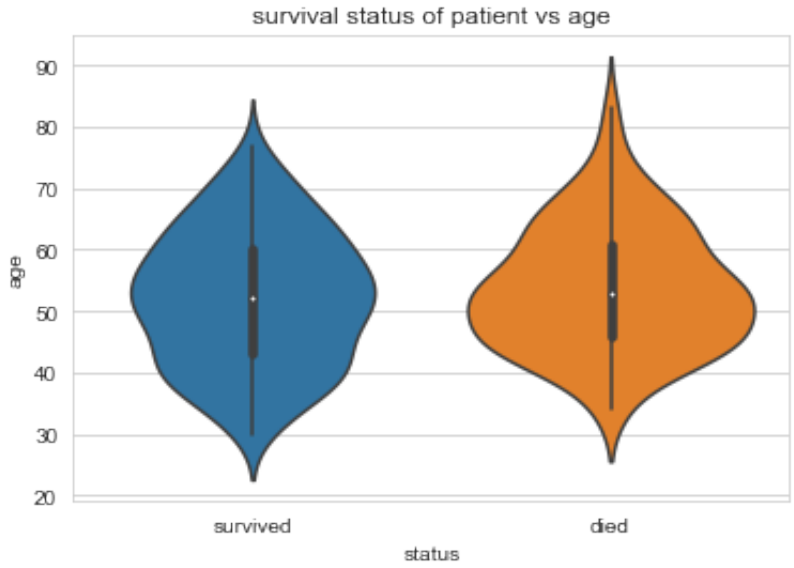
- 50% of patients who survived more than 5 years are having 0 positive axillary nodes(median)
- 50% of patients who died within 5 years are having the positive axillary nodes within the range of 1 to 11(IQR)
- Almost patients who survived more than 5 years having their positive nodes less than 7(max). Hence,we see some outliers above the whiskers.

From the box plot we found that only 25% of the patients having positive nodes overlapped for both survival status. Hence, nodes seems to be much useful in classifying the patient's survival status than other features.

5.4 Violin plot

- It is similar to a box plot, with the addition of PDF on each side and they look like a voilin, hence named as violin plot
- In a violin plot Denser regions of the data are fatter, and sparser ones are thinner.

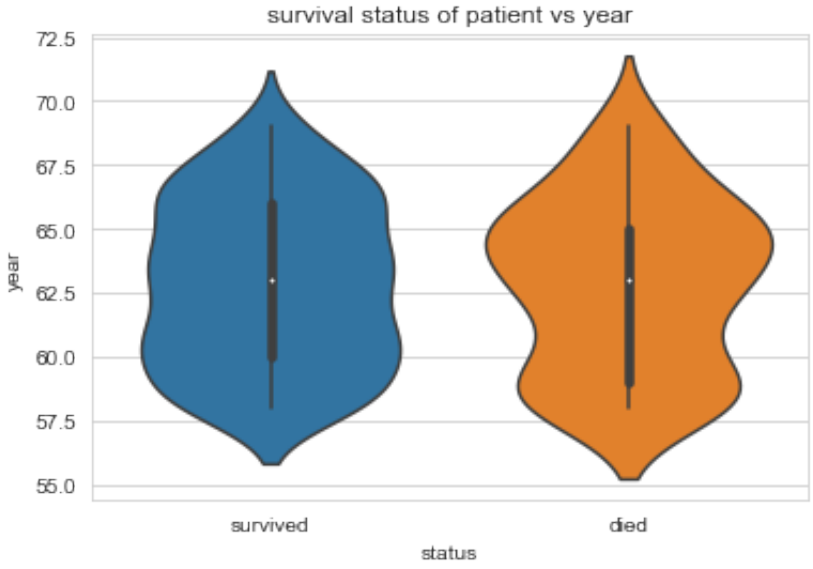
In [51]:
```
#violinplot for survival status of patient vs age
sns.violinplot(x="status", y="age", data=haberman_df, size=8)
plt.title('survival status of patient vs age')
plt.show()
```



Observations:

- age of patients who died within 5 years is highly densed between 45 and 55

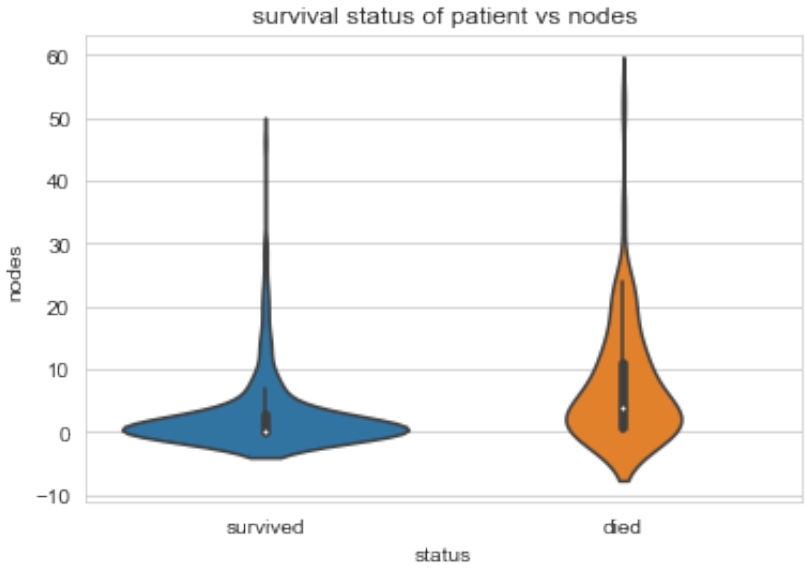
In [52]:
```
#violinplot for survival status of patient vs year
sns.violinplot(x="status", y="year", data=haberman_df, size=8)
plt.title('survival status of patient vs year')
plt.show()
```



Observations:

- year of operation of patients who survived more than 5 years is highly densed in 1960
- year of operation of patients who died within 5 years is highly densed in 1965

In [53]:
```
#violinplot for survival status of patient vs nodes
sns.violinplot(x="status", y="nodes", data=haberman_df, size=8)
plt.title('survival status of patient vs nodes')
plt.show()
```



Observations:

- Positive axillary nodes of patients who survived more than 5 years is highly densed for 0 to 2 nodes.
- Positive axillary nodes of patients who died within 5 years is highly densed for 4 to 7 nodes.

From the violin plot we found that most people who survived more than 5 years have zero positive axillary nodes.Hence, nodes seems to be much useful in classifying the patient's survival status than other features.

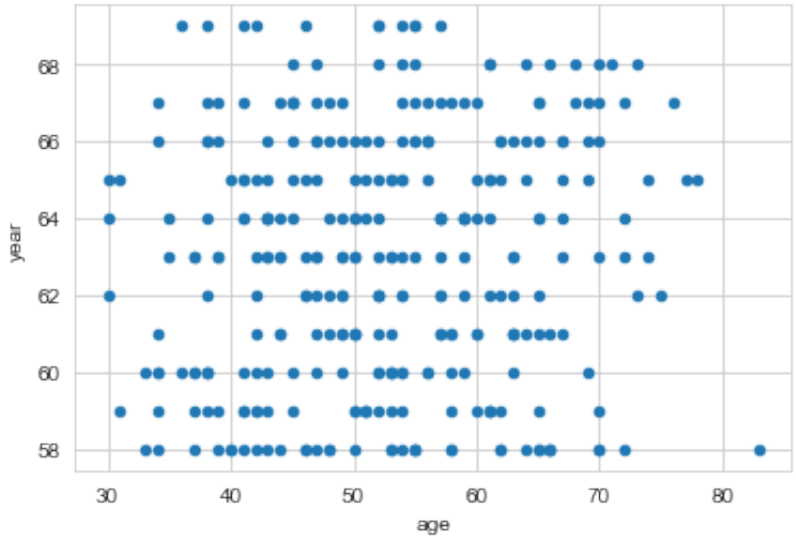
6. Bivariate analysis

Bivariate analysis (scatter plots, pair-plots) is performed to understand if combinations of features are useful in classfication.

6.1.Scatter plot

A scatter plot shows the relationship between two numerical variables where each value in the data set is represented by a dot.

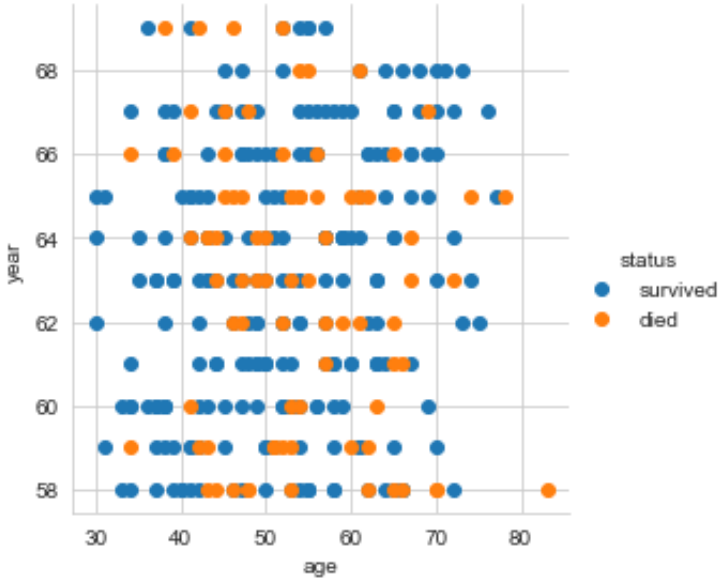
In [54]:
```
#2-D scatter plot between age and year
haberman_df.plot(kind='scatter', x='age', y='year') ;
plt.show()
```



Observations:

can't say much about this plot because both variables are represented by the same color

In [55]:
```
# 2-D Scatter plot with color-coding for each class.
sns.set_style("whitegrid")
sns.FacetGrid(haberman_df, hue='status', height=4) \
   .map(plt.scatter, "age", "year") \
   .add_legend();
plt.show();
```



Observations :

now it is easy to differentiate between two variables based on the survival status but this combination of features (**age and year**) are not much useful in classifying the patient, as the points have overlapping.

6.2 pair plot

Instead of plotting scatter plot individually for each combination, we can plot a pair plot which shows a clear view of relationship between all combination of features.

In [56]:
```
# pairwise scatter plot: Pair-Plot
sns.pairplot(haberman_df, hue ="status", diag_kind = "hist", height
= 3)
plt.show()
```

Observations:

- The histogram on the diagonal shows the distribution of a single variable while the scatter plots on the upper and lower side of the diagonal shows the relationship between two variables.
- From **age and nodes** combination, patients with age between 30 to 40 and having lesser number of positive axillary nodes seems to be more survived.
- In all combinations, there is so much overlapping, thus no plot can be linearly separable and not seems to be much useful in classifying the patient.

CONCLUSION

Haberman_df is a imbalanced dataset, as the number of data points for the 2 classes are not equal.

We plotted multiple plots above to classify a new patient as belonging to one of the 2 classes based on the 3 features.

1.age

```
if age > 30 && age < 34, then survived more than 5
if age >77, then died within 5 years
```

for the rest of the patients, it is not possible to write working if/else code, hence 'age' won't totally help in classifying.

2.year

```
'year' does not depicts any valuable inference as it overlaps for both
survival status.
```

3.nodes

```
from all the plots,'nodes' gives us a clear idea that the patients havi
ng 0 or less number of positive axilary nodes have survived more than 5
years after the operation.
```

Since there is too much overlapping in data points, it is difficult to create simple linearly seperable model to classify a new patient as belonging to one of the 2 classes based on the 3 features.

But it can be possible to assume that as the number of positive axillary nodes and age increases, the chance of survival of patients decreases.