Exploratory Data Analysis on Haberman Dataset

December 27, 2019

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

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

0.0.1 Haberman Dataset information present in Kaggle:

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.

Dataset Column Information:

1. Age of patient at time of operation (numerical)
2. Patient's year of operation (year - 1900, numerical)
3. Number of positive axillary nodes detected (numerical)
4. Survival status (class attribute) 1 = the patient survived 5 years or longer 2 = the patient died within 5 year

[2]:
```
print(haberman.shape)
```

```
(306, 4)
```

[3]:
```
print(haberman.columns)
```

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

[4]:
```
haberman.head()
```

[4]:

	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

[5]:
```
haberman.describe()
```

[5]:

	age	year	nodes	status
count	306.000000	306.000000	306.000000	306.000000
mean	52.457516	62.852941	4.026144	1.264706
std	10.803452	3.249405	7.189654	0.441899
min	30.000000	58.000000	0.000000	1.000000
25%	44.000000	60.000000	0.000000	1.000000
50%	52.000000	63.000000	1.000000	1.000000
75%	60.750000	65.750000	4.000000	2.000000
max	83.000000	69.000000	52.000000	2.000000

[6]:
```
haberman["status"].value_counts()
```

[6]:
```
1    225
2     81
Name: status, dtype: int64
```

[7]:
```
haberman['status'].replace(1,'Survived 5 years',inplace=True)
haberman['status'].replace(2,'Died in 5 years',inplace=True)

patient_survived_5_years=haberman.loc[haberman["status"]=='Survived 5 years']
patient_died_in_5_years=haberman.loc[haberman["status"]=='Died in 5 years']
```

[8]:
```
print(patient_survived_5_years.head())
print(patient_died_in_5_years.head())
```

```
   age  year  nodes            status
0   30    64      1  Survived 5 years
1   30    62      3  Survived 5 years
2   30    65      0  Survived 5 years
3   31    59      2  Survived 5 years
4   31    65      4  Survived 5 years
    age  year  nodes           status
7    34    59      0  Died in 5 years
8    34    66      9  Died in 5 years
24   38    69     21  Died in 5 years
34   39    66      0  Died in 5 years
43   41    60     23  Died in 5 years
```

[9]:
```
print(patient_survived_5_years.shape[0]/haberman.shape[0])
```

```
0.7352941176470589
```

Higl Level Statistics - Observations:

1. There are 306 points in this Dataset with 4 features - age of patient, year of operation, number of positive auxillary lymph nodes and Survival status (whether the patient survived

for 5 years)

2. The data is classified based on the Survival Status feature. If the Survival Status feature is '1' - it means that the patient survived for 5 years or longer If the Survival Status feature is '2' - it means that the patient died within 5 years
3. In the dataset, there are 225 patients who survived for 5 years or longer and 81 patients who died within 5 years. This is an imbalanced dataset where 73% of the data is of the patients who survived for 5 years or longer
4. The age of patients vary from 30 to 83 with 50% of patients having age less than 53
5. Even though the highest number of positive auxillary nodes in the dataset is 52, 75% of patients have less than 5 positive auxillary nodes and 25% of the patients have no positive auxillary nodes

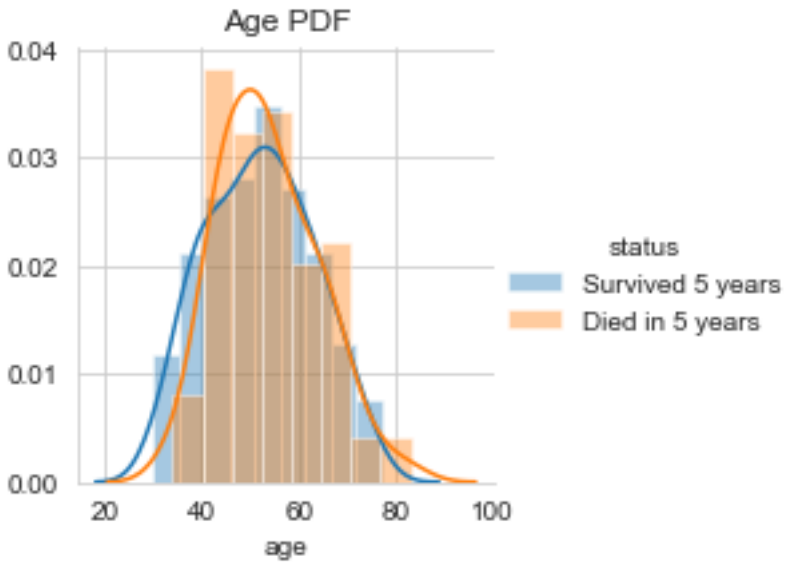
0.1 Objective:

To predict whether a patient will survive for 5 years or longer based on the patients age, year of treatment and the number of positive auxillary nodes

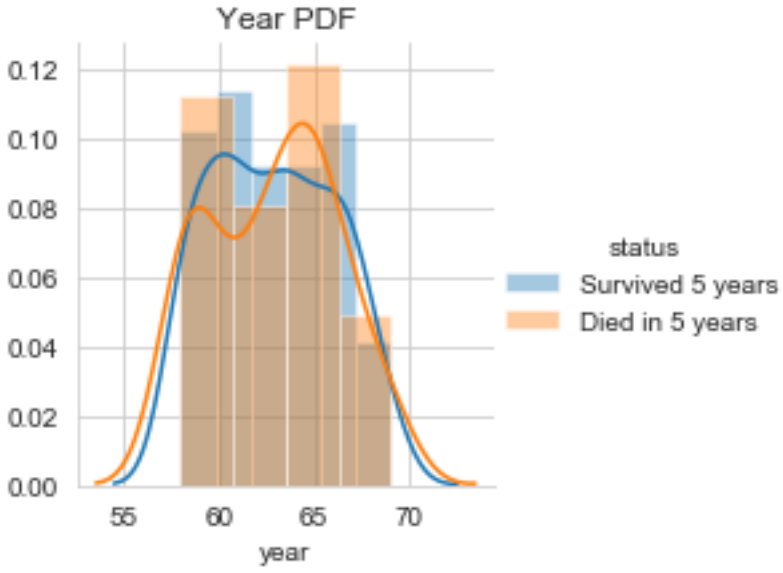
0.1.1 Univariate Analysis:

PDF

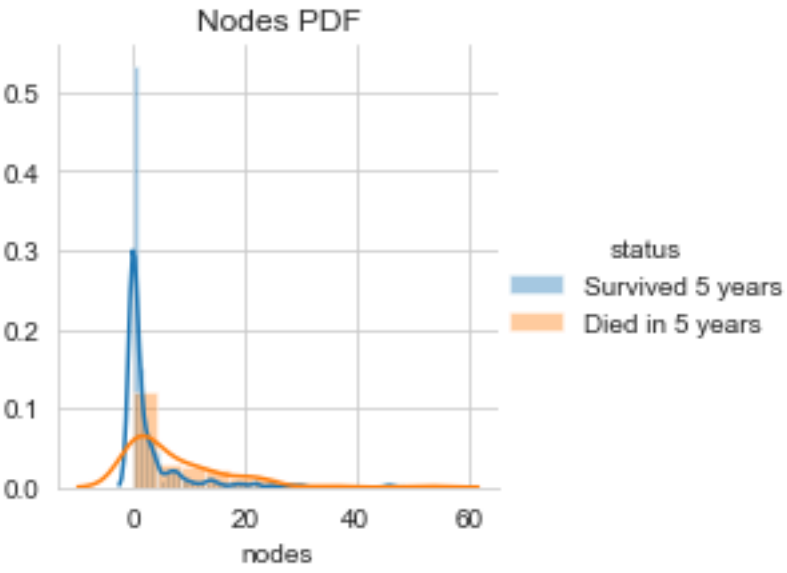
[10]:
```
sns.set_style('whitegrid')
sns.FacetGrid(haberman,hue='status')\
    .map(sns.distplot,"age")\
    .add_legend()
plt.title('Age PDF')
plt.show()
```



[11]:
```
sns.FacetGrid(haberman,hue='status')\
    .map(sns.distplot,'year')\
    .add_legend()
plt.title('Year PDF')
plt.show()
```



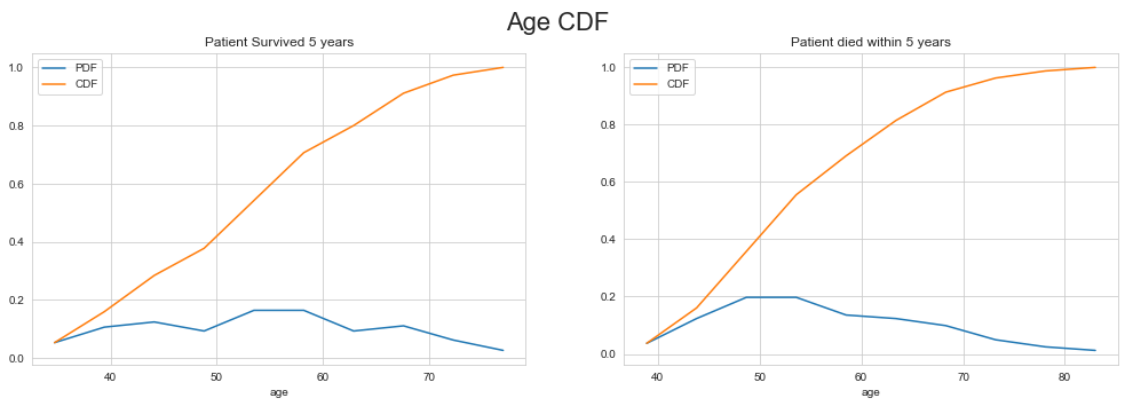
[12]:
```
sns.FacetGrid(haberman,hue='status')\
    .map(sns.distplot,"nodes")\
    .add_legend()
plt.title('Nodes PDF')
plt.show()
```

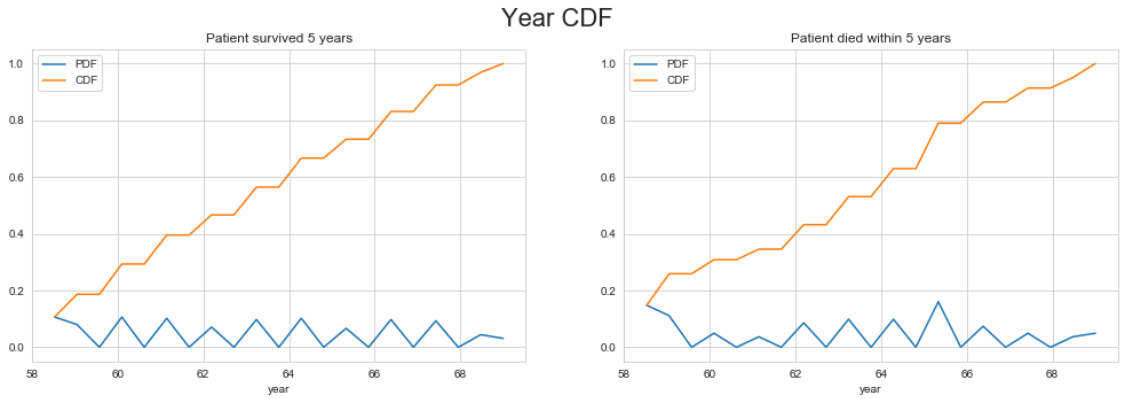


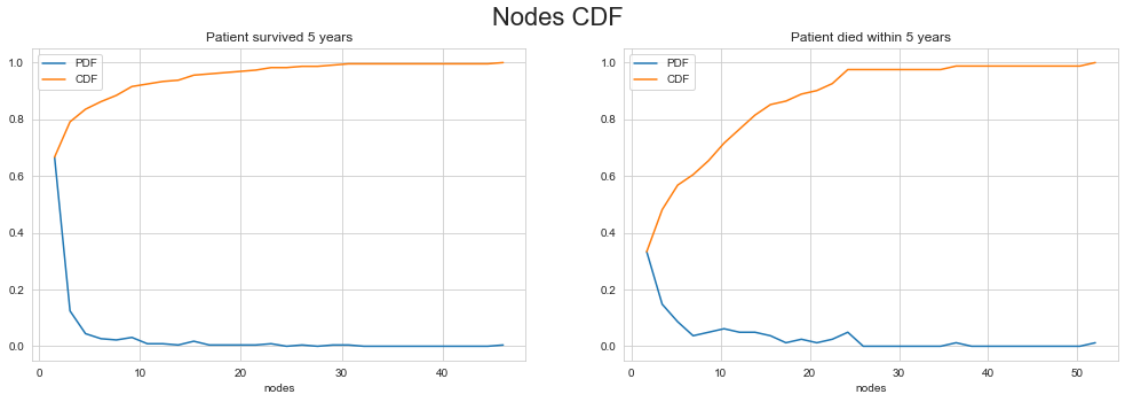
CDF

[13]:
```
plt.figure(1,figsize=(17,5))
plt.suptitle('Age CDF',fontsize=22)
plt.grid()
plt.subplot(121)

counts,bin_edges=np.
 ↪histogram(patient_survived_5_years['age'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf = np.cumsum(pdf)

plt.plot(bin_edges[1:],pdf,label='PDF')
plt.plot(bin_edges[1:],cdf,label='CDF')
plt.title('Patient Survived 5 years')
plt.xlabel('age')
plt.legend()

plt.subplot(122)
counts,bin_edges=np.
 ↪histogram(patient_died_in_5_years['age'],bins=10,density=True)
pdf=counts/(sum(counts))
cdf = np.cumsum(pdf)

plt.plot(bin_edges[1:],pdf,label='PDF')
plt.plot(bin_edges[1:],cdf,label='CDF')
plt.title('Patient died within 5 years')
plt.xlabel('age')
plt.legend()

plt.figure(2,figsize=(17,5))
plt.suptitle('Year CDF',fontsize=22)
plt.subplot(121)
counts,bin_edges=np.
 ↪histogram(patient_survived_5_years['year'],bins=21,density=True)
pdf=counts/(sum(counts))
cdf = np.cumsum(pdf)

plt.plot(bin_edges[1:],pdf,label='PDF')
plt.plot(bin_edges[1:],cdf,label='CDF')
plt.title('Patient survived 5 years')
plt.xlabel('year')
plt.legend()

plt.subplot(122)
```

```
counts,bin_edges=np.
 ↪histogram(patient_died_in_5_years['year'],bins=21,density=True)
pdf=counts/(sum(counts))
cdf = np.cumsum(pdf)

plt.plot(bin_edges[1:],pdf,label='PDF')
plt.plot(bin_edges[1:],cdf,label='CDF')
plt.title('Patient died within 5 years')
plt.xlabel('year')
plt.legend()

plt.figure(3,figsize=(17,5))
plt.suptitle('Nodes CDF',fontsize=22)
plt.subplot(121)
counts,bin_edges=np.
 ↪histogram(patient_survived_5_years['nodes'],bins=30,density=True)
pdf=counts/(sum(counts))
cdf = np.cumsum(pdf)

plt.plot(bin_edges[1:],pdf,label='PDF')
plt.plot(bin_edges[1:],cdf,label='CDF')
plt.title('Patient survived 5 years')
plt.xlabel('nodes')
plt.legend()

plt.subplot(122)
counts,bin_edges=np.
 ↪histogram(patient_died_in_5_years['nodes'],bins=30,density=True)
pdf=counts/(sum(counts))
cdf = np.cumsum(pdf)

plt.plot(bin_edges[1:],pdf,label='PDF')
plt.plot(bin_edges[1:],cdf,label='CDF')
plt.title('Patient died within 5 years')
plt.xlabel('nodes')
plt.legend()



plt.show()
```



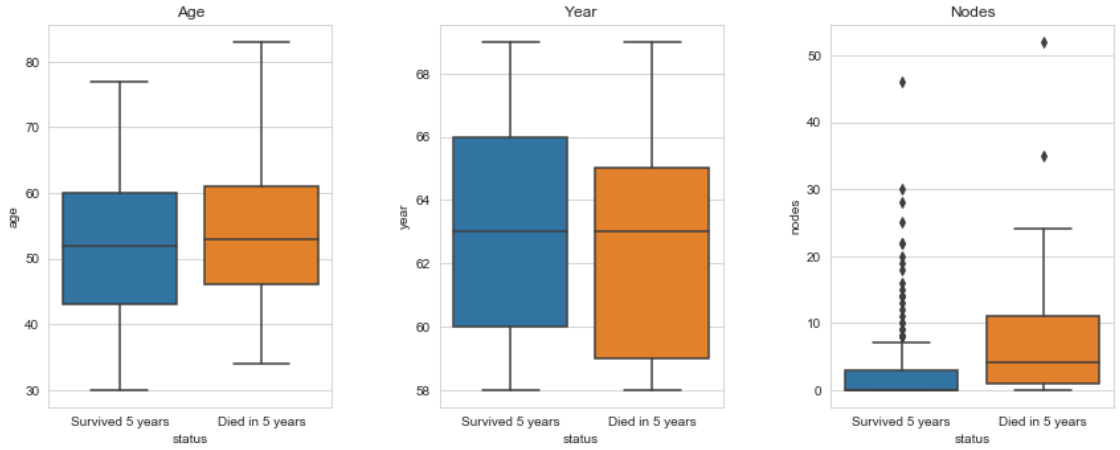




Box Plot

[14]:
```
plt.figure(1,figsize=(13,6))

plt.subplot(131)
```

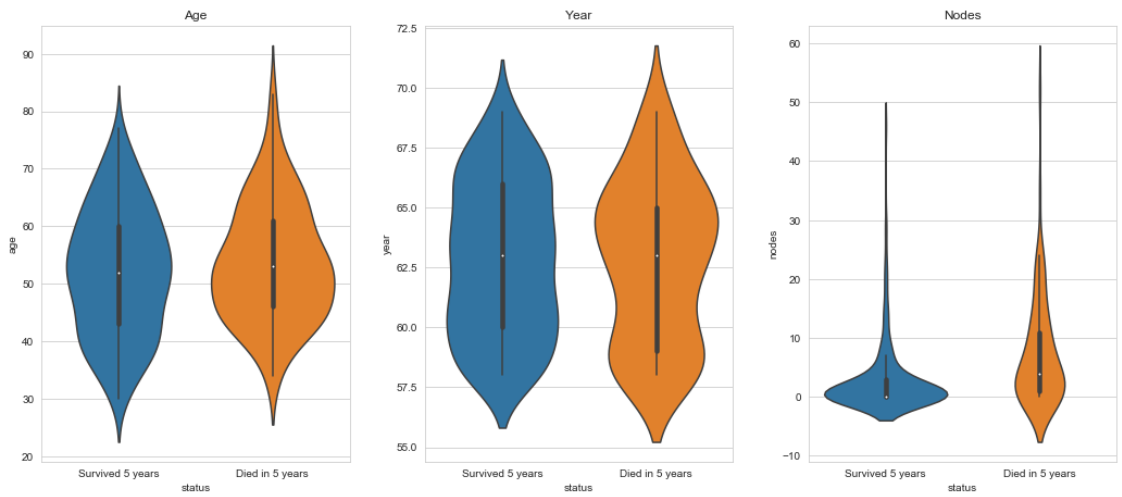
```
plt.title('Age')
sns.boxplot(x='status',y='age', data=haberman)

plt.subplot(132)
plt.title('Year')
sns.boxplot(x='status',y='year', data=haberman)


plt.subplot(133)
plt.title('Nodes')
sns.boxplot(x='status',y='nodes', data=haberman)

plt.tight_layout(pad=5.0)
plt.show()
```



Violin Plots

[15]:
```
plt.figure(1,figsize=(15,7))

plt.subplot(131)
plt.title('Age')
sns.violinplot(x='status',y='age', data=haberman)

plt.subplot(132)
plt.title('Year')
sns.violinplot(x='status',y='year', data=haberman)


plt.subplot(133)
plt.title('Nodes')
sns.violinplot(x='status',y='nodes', data=haberman)
```

```
plt.tight_layout(pad=3.0)
plt.show()
```



[16]:
```
patient_survived_5_years.describe()
```

[16]:

	age	year	nodes
count	225.000000	225.000000	225.000000
mean	52.017778	62.862222	2.791111
std	11.012154	3.222915	5.870318
min	30.000000	58.000000	0.000000
25%	43.000000	60.000000	0.000000
50%	52.000000	63.000000	0.000000
75%	60.000000	66.000000	3.000000
max	77.000000	69.000000	46.000000

[17]:
```
patient_died_in_5_years.describe()
```

[17]:

	age	year	nodes
count	81.000000	81.000000	81.000000
mean	53.679012	62.827160	7.456790
std	10.167137	3.342118	9.185654
min	34.000000	58.000000	0.000000
25%	46.000000	59.000000	1.000000
50%	53.000000	63.000000	4.000000
75%	61.000000	65.000000	11.000000
max	83.000000	69.000000	52.000000

[18]:
```
np.percentile(patient_survived_5_years['nodes'],79)
```

[18]: 3.0

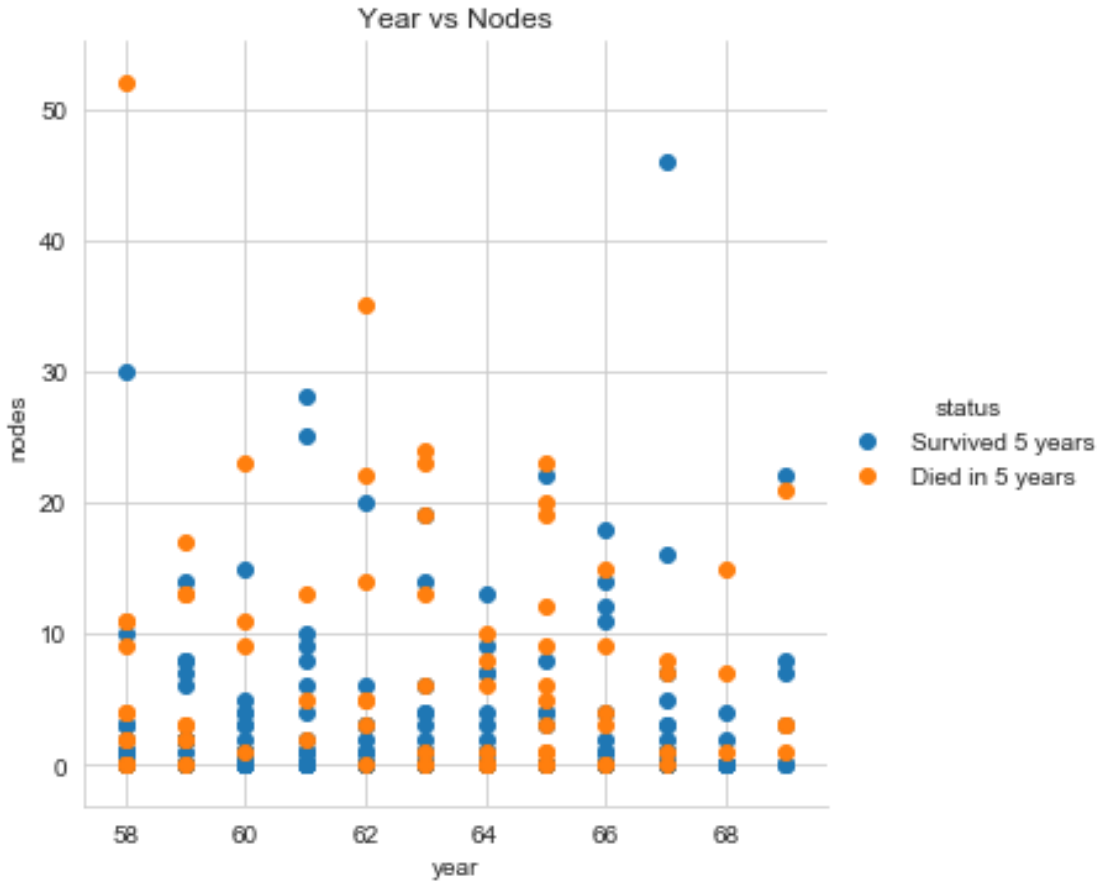
Observations:

1. The feature 'nodes' is the most useful feature toward classification compared to the other features
2. Age and year have too many overlaps between the status that they don't help much in classification
3. 79% of the patients that survived 5 years or longer have auxillary positive nodes less than 4

0.1.2 Bivariate Analysis

2d Scatter plot

[19]:
```
sns.set_style("whitegrid");
sns.FacetGrid(haberman, hue="status", height=5) \
   .map(plt.scatter, "year", "nodes") \
   .add_legend();
plt.title('Year vs Nodes')
plt.show();
```

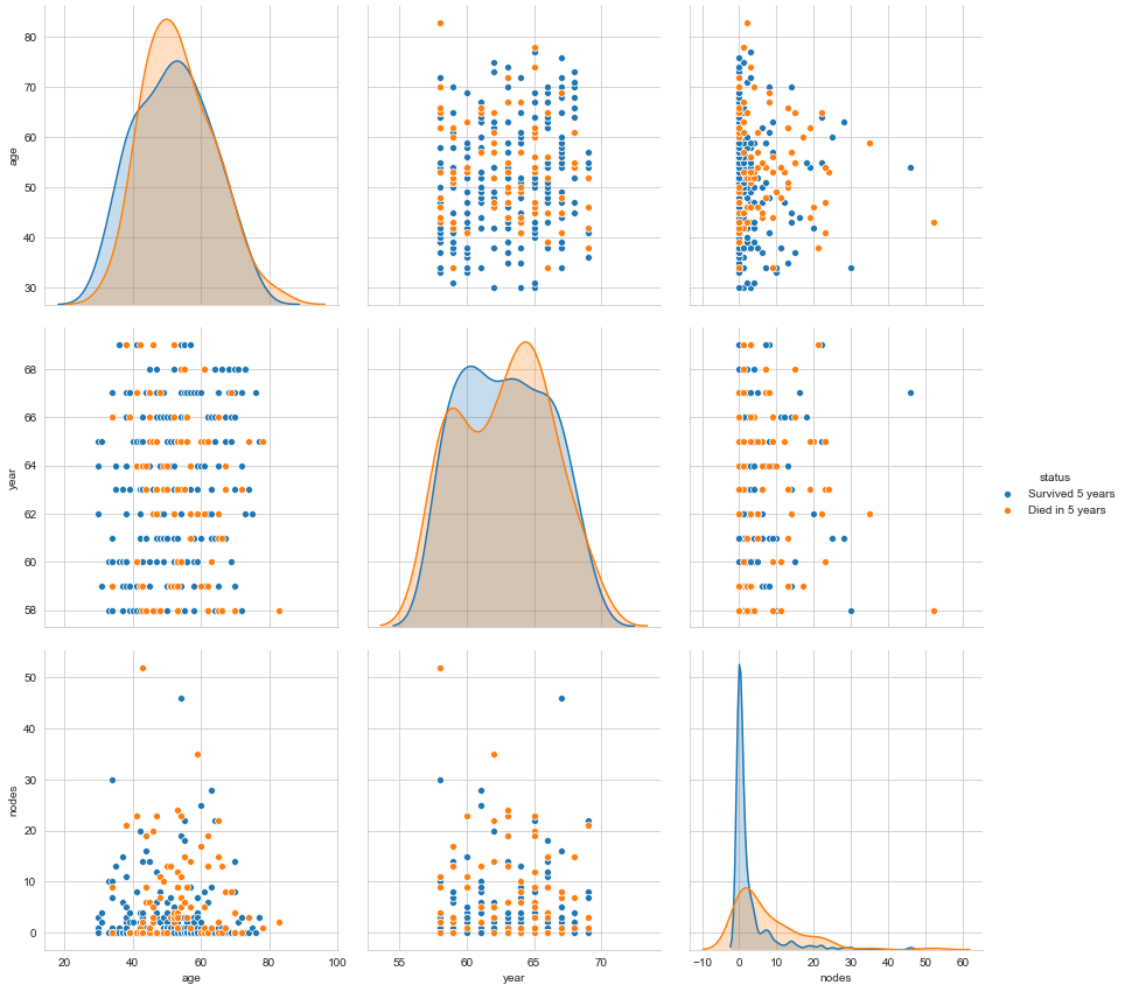


3D Scatter plot

[20]:
```
import plotly.express as px
fig = px.scatter_3d(haberman, x='age', y='year', z='nodes',
                    color='status')
fig.show()
```

Pair Plot

[21]:
```
sns.set_style("whitegrid");
sns.pairplot(haberman, hue="status", height=4);
plt.show()
```



Observation:

1. The pair plots between the features age,year and nodes have many overlaps between the two classes and do not help in separating the two classes