(3.12) Exercise:

1. Download Haberman Cancer Survival dataset from Kaggle. You may have to create a Kaggle account to donwload data. (https://www.kaggle.com/gilsousa/habermans-survival-data-set)
2. Perform a similar alanlaysis as above on this dataset with the following sections:

- High level statistics of the dataset: number of points, numer of features, number of classes, data-points per class.
- Explain our objective.
- Perform Univaraite analysis(PDF, CDF, Boxplot, Voilin plots) to understand which features are useful towards classification.
- Perform Bi-variate analysis (scatter plots, pair-plots) to see if combinations of features are useful in classfication.
- Write your observations in english as crisply and unambigously as possible. Always quantify your results.

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.

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

Attributes:

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

Objectice

To explore the Haberman Cancer Survival Dataset and find which feature or combination of feature are helpfull in determining the status of a person in 5 years after the operation.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
#Load Haberman Dataset to a pandas dataframe

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

In [2]:
```
# (Q) how many data-points and features?
haberman.shape
```

Out[2]:
```
(305, 4)
```

In [3]:
```
#(Q) What are the column names in our dataset?
haberman.columns
```

Out[3]:
```
Index(['30', '64', '1', '1.1'], dtype='object')
```

In [4]:
```
#Columns changed
haberman.columns = ['age', 'operation_year', 'axil_nodes', 'survived_status']
```

In [5]:
```
haberman.columns
```

Out[5]:
```
Index(['age', 'operation_year', 'axil_nodes', 'survived_status'], dtype='object')
```

In [6]:
```
haberman['survived_status'].value_counts()
```

Out[6]:
```
1    224
2     81
Name: survived_status, dtype: int64
```

In [7]:
```
haberman.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 305 entries, 0 to 304
Data columns (total 4 columns):
age                305 non-null int64
operation_year     305 non-null int64
axil_nodes         305 non-null int64
survived_status    305 non-null int64
dtypes: int64(4)
memory usage: 9.6 KB
```

In [8]:
```
haberman.describe()
```

Out[8]:

	age	operation_year	axil_nodes	survived_status
count	305.000000	305.000000	305.000000	305.000000
mean	52.531148	62.849180	4.036066	1.265574
std	10.744024	3.254078	7.199370	0.442364
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%	61.000000	66.000000	4.000000	2.000000
max	83.000000	69.000000	52.000000	2.000000

Observations:
1) Dataset has 4 features/variable and 305 data points.

2) Patient's age lie between 30-83 years those who had undergone cancer surgery in year 1958-1969.

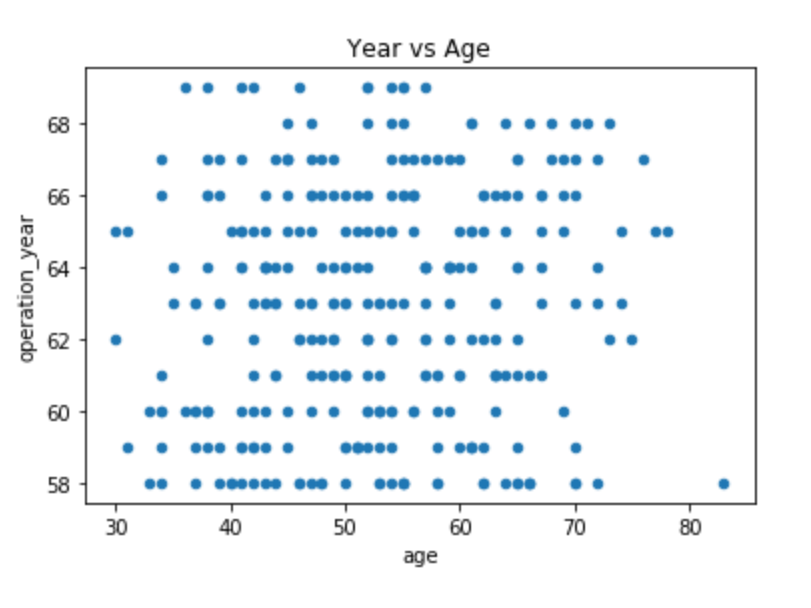
3) 0-4 axil nodes: 75% patient,0 node: 25% patient and very few had up to 52 axil nodes.

4) The dataset has 224 datapoint labeled as "1" and 81 datapoint labeled as "2" viz.,Surv_status:"1=Survived(the patient survived 5 years or longer)", "2=Died(the patient died within 5 year)". Therefore, the dataset is an imbalance dataset

2-D Scatter Plot

In [9]:
```
haberman.plot(kind='scatter', x='age', y='operation_year', title='Year vs Age')
```

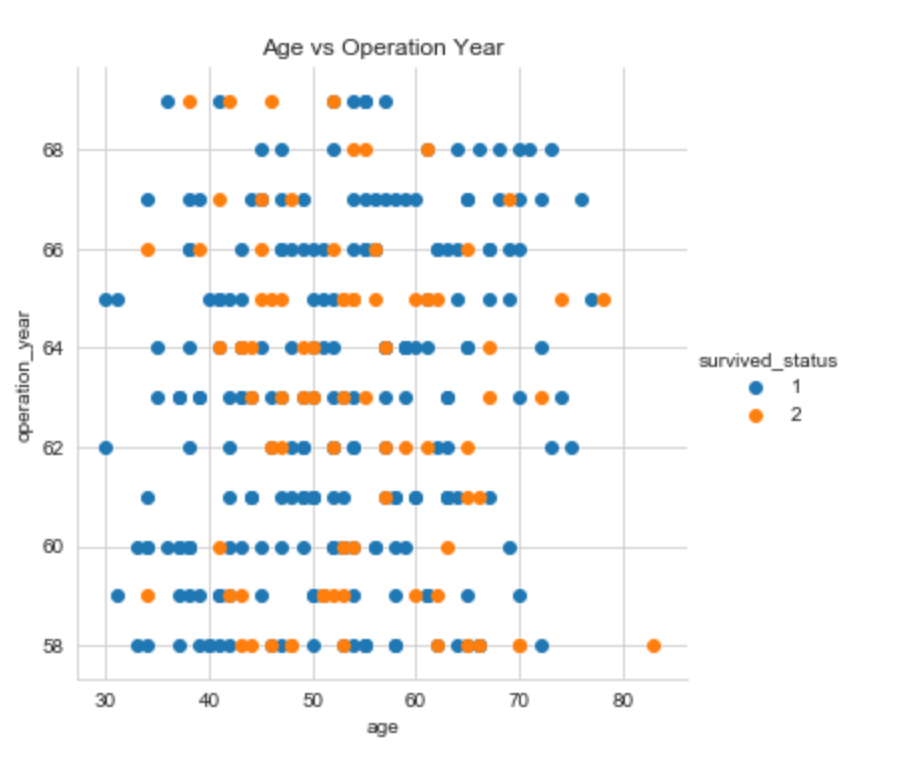
Out[9]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2b2d03cf518>
```



Relation between age and year is not clearly visible with this plot

In [10]:
```
# 2-D Scatter plot with color-coding
sns.set_style('whitegrid')
sns.FacetGrid(haberman, hue='survived_status', size=5) \
    .map(plt.scatter,'age', 'operation_year').add_legend()
plt.title('Age vs Operation Year')
```

Out[10]:
```
Text(0.5, 1.0, 'Age vs Operation Year')
```



Classification is not possible based on year or age realtionship

Pair Plots

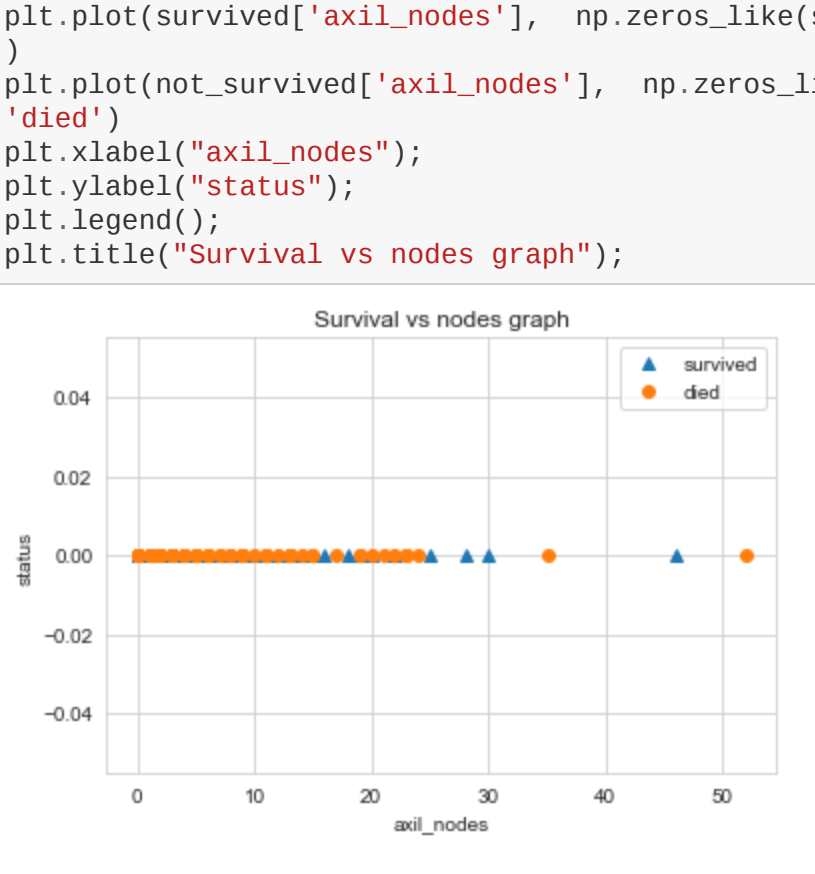
In [11]:
```
sns.set_style('whitegrid')
sns.pairplot(haberman, hue='survived_status', vars = ['age', 'operation_year', 'axil_nodes']
, size=3)
plt.title("pair plots of age, operation year and nodes")
```

Out[11]:
```
Text(0.5, 1.0, 'pair plots of age, operation year and nodes')
```


```
Almost all the points have less than 30 axil nodes. Still there is no clear separation
between status 1 and status 2
```

Histogram, PDF, CDF

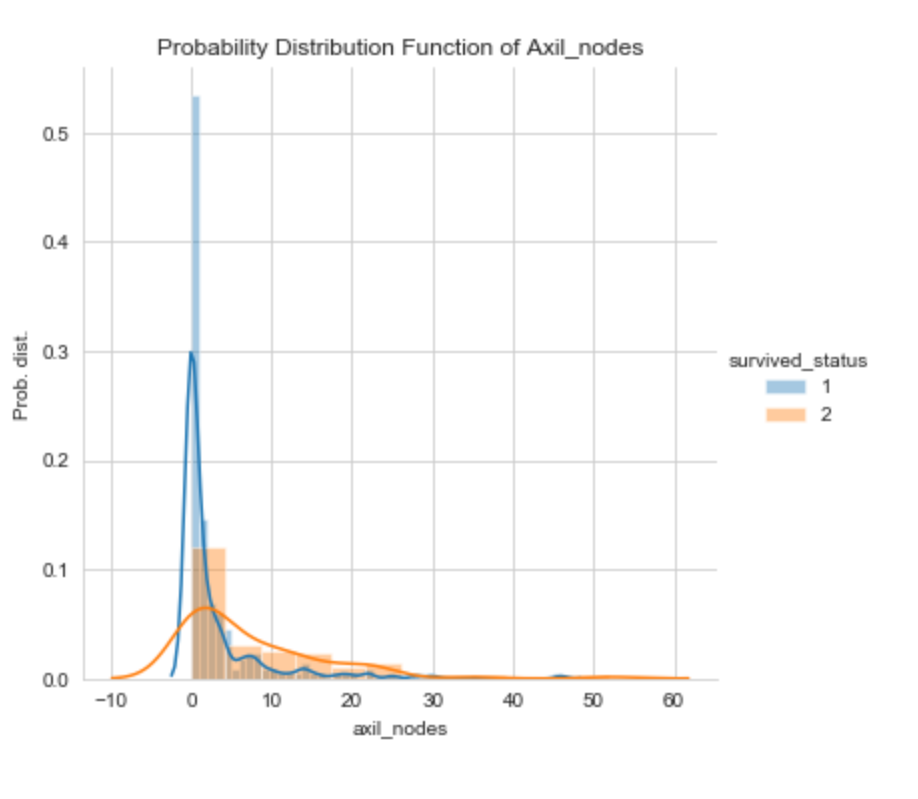
In [12]:
```
survived = haberman[haberman['survived_status'] == 1]
not_survived = haberman[haberman['survived_status'] == 2]
plt.plot(survived['axil_nodes'],  np.zeros_like(survived['axil_nodes']),'^',label='survived'
)
plt.plot(not_survived['axil_nodes'],  np.zeros_like(not_survived['axil_nodes']),'o',label =
'died')
plt.xlabel("axil_nodes");
plt.ylabel("status");
plt.legend();
plt.title("Survival vs nodes graph");
```



Majority of the points are overlapping, hence it is difficult to draw conclusion on this variable

In [13]:
```
sns.FacetGrid(haberman, hue='survived_status', height=5)\
    .map(sns.distplot, 'axil_nodes')\
    .add_legend()
plt.ylabel('Prob. dist.')
plt.title('Probability Distribution Function of Axil_nodes')
```

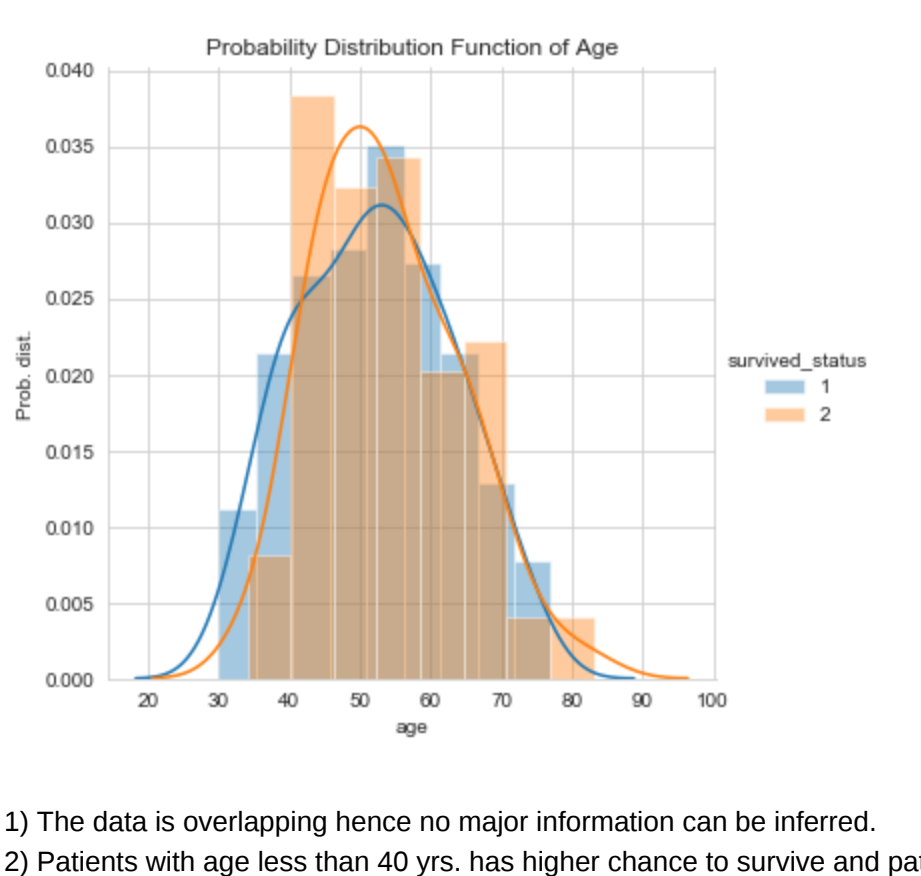
Out[13]:
```
Text(0.5, 1.0, 'Probability Distribution Function of Axil_nodes')
```



We can observe that with 0-3 axil nodes, the chances of survival is more.

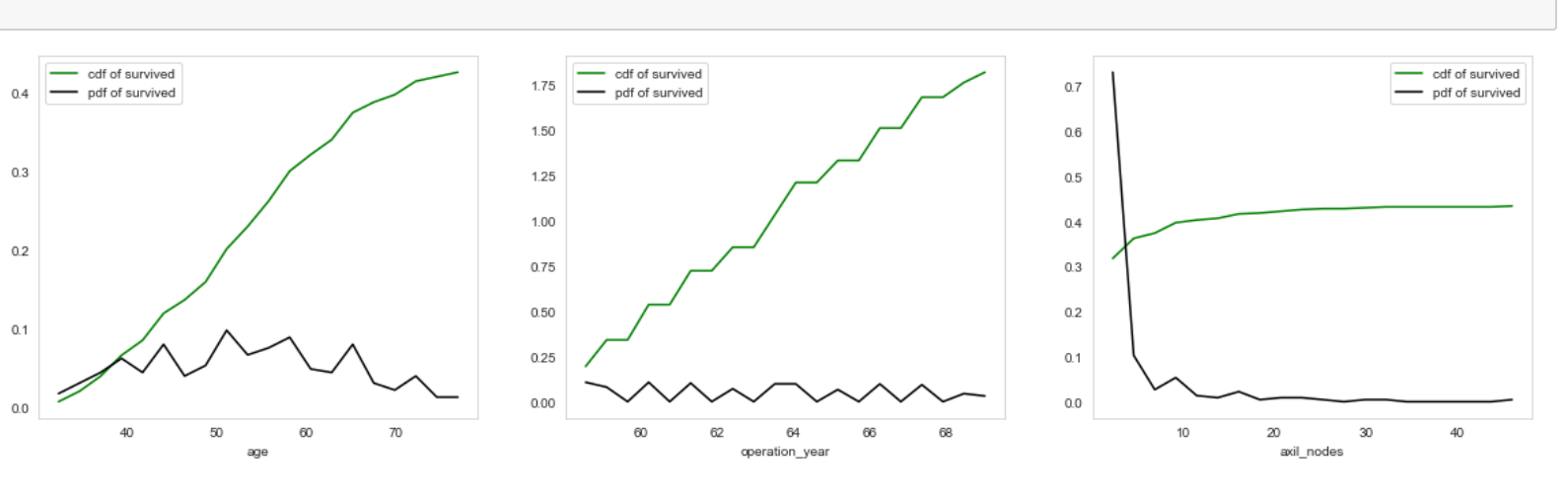
In [14]:
```
sns.FacetGrid(haberman, hue='survived_status', height=5)\
    .map(sns.distplot, 'age')\
    .add_legend()
plt.ylabel('Prob. dist.')
plt.title('Probability Distribution Function of Age')
```

Out[14]:
```
Text(0.5, 1.0, 'Probability Distribution Function of Age')
```

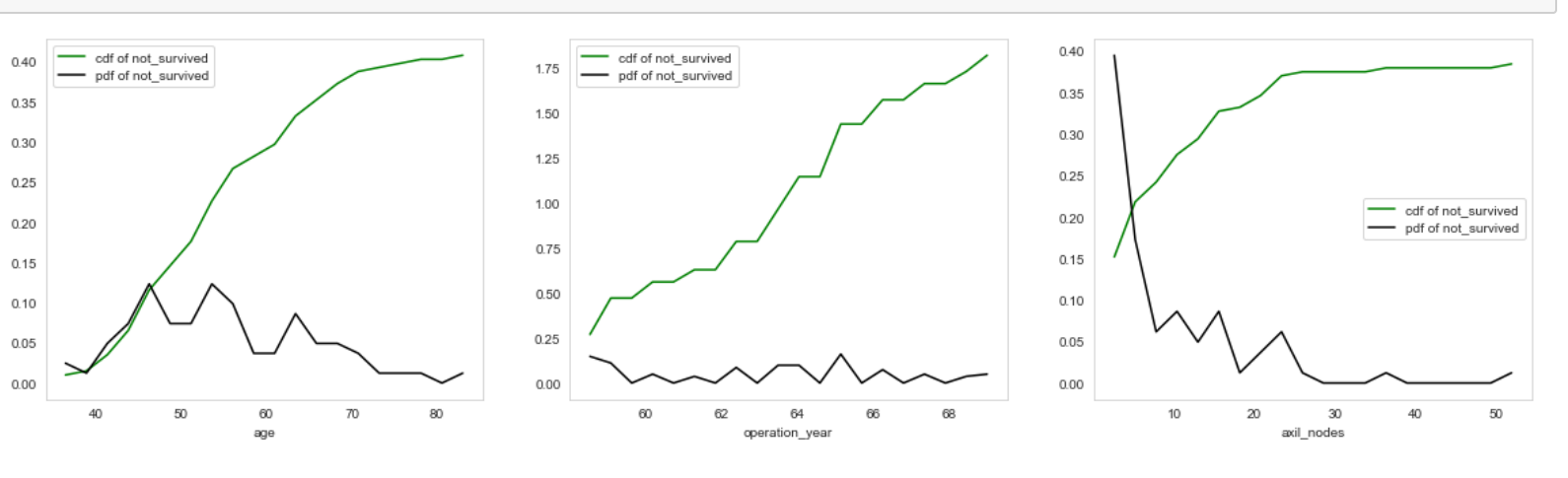


1) The data is overlapping hence no major information can be inferred.
2) Patients with age less than 40 yrs. has higher chance to survive and patient with age more than 78 yrs are most likely to die within 5 yrs. of surgery

In [15]:
```
# pdf and cdf of survived
plt.figure(figsize=(20,5))
i=1
for column in (list(haberman.columns)[:-1]):
    plt.subplot(1,3,i)
    counts, bin_edges = np.histogram(survived[column], bins=20, density=True)
    pdf = counts/sum(counts)
    cdf = np.cumsum(counts)
    plt.plot(bin_edges[1:], cdf, label='cdf of survived', color='green')
    plt.plot(bin_edges[1:], pdf, label = 'pdf of survived', color='black')
    plt.xlabel(column)
    plt.grid()
    plt.legend()
    i+=1
```



In [16]:
```
# pdf and cdf of not_survived
plt.figure(figsize=(20,5))
i=1
for column in (list(haberman.columns)[:-1]):
    plt.subplot(1,3,i)
    counts, bin_edges = np.histogram(not_survived[column], bins=20, density=True)
    pdf = counts/sum(counts)
    cdf = np.cumsum(counts)
    plt.plot(bin_edges[1:], cdf, label='cdf of not_survived', color='green')
    plt.plot(bin_edges[1:], pdf, label = 'pdf of not_survived', color='black')
    plt.xlabel(column)
    plt.grid()
    plt.legend()
    i+=1
```



1) Patient with age between age 32-36 has definitly survived the operation and pataient aged 77-85 has definitly not survived the operation.
2) Patient with axil nodes > 22 are more likely to die.

Mean, Variance and Std-dev

In [17]:
```
print("Means:")
print("survived")
print(np.mean(survived['age']))
print(np.mean(survived['operation_year']))
print(np.mean(survived['axil_nodes']))

print()

print("not survived")
print(np.mean(not_survived['age']))
print(np.mean(not_survived['operation_year']))
print(np.mean(not_survived['axil_nodes']))

print()

print("std:")
print(np.std(haberman['age']))
print(np.std(haberman['operation_year']))
print(np.std(haberman['axil_nodes']))
```

```
Means:
survived
52.11607142857143
62.857142857142854
2.799107142857143

not survived
53.67901234567901
62.82716049382716
7.45679012345679

std:
10.726396748570311
3.2487386178063162
7.187558302814359
```

Median, Percentile, Quantile, IQR, MAD

In [18]:
```
print("\nMedians:")
print(np.median(haberman["age"]))
print(np.median(haberman["operation_year"]))
print(np.median(haberman["axil_nodes"]))


print("\nQuantiles:")
print(np.percentile(haberman["age"],np.arange(0, 100, 25)))
print(np.percentile(haberman["operation_year"],np.arange(0, 100, 25)))
print(np.percentile(haberman["axil_nodes"], np.arange(0, 100, 25)))

print("\n90th Percentiles:")
print(np.percentile(haberman["age"],90))
print(np.percentile(haberman["operation_year"],90))
print(np.percentile(haberman["axil_nodes"], 90))

from statsmodels import robust
print ("\nMedian Absolute Deviation")
print(robust.mad(haberman["age"]))
print(robust.mad(haberman["operation_year"]))
print(robust.mad(haberman["axil_nodes"]))
```

```
Medians:
52.0
63.0
1.0

Quantiles:
[30. 44. 52. 61.]
[58. 60. 63. 66.]
[0. 0. 1. 4.]

90th Percentiles:
67.0
67.0
13.0

Median Absolute Deviation
11.860817748044816
4.447806655516806
1.482602218505602
```

Median Absolute Deviation for age shows the heterogeneity in age is more than other features

IQR of age (75th percentile - 25th percentile) shows that the 50 percent of age lies in range 44 - 52

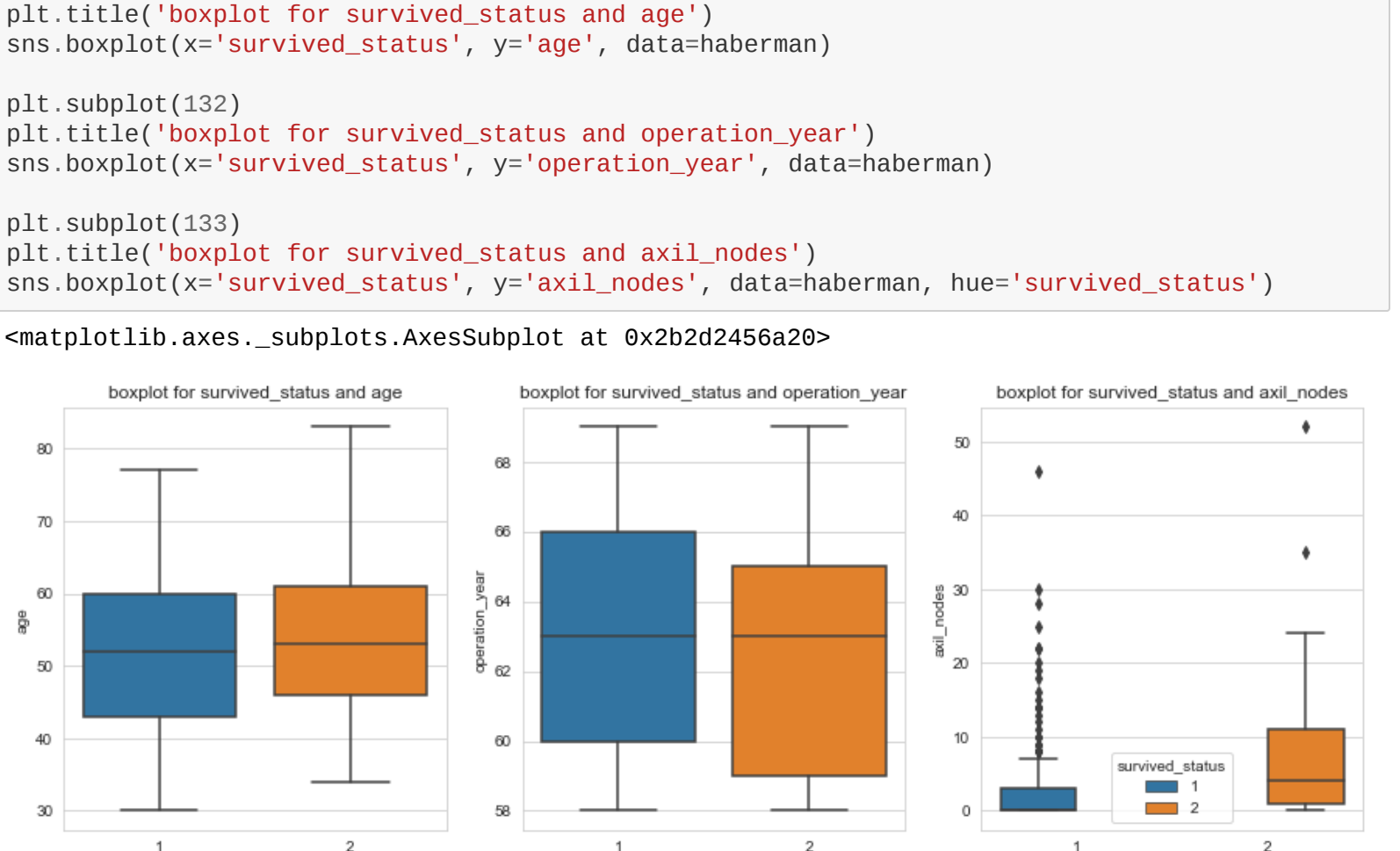
Box Plot and Whiskers

In [19]:
```
plt.figure(figsize=(15,5))
plt.subplot(131)
plt.title('boxplot for survived_status and age')
sns.boxplot(x='survived_status', y='age', data=haberman)

plt.subplot(132)
plt.title('boxplot for survived_status and operation_year')
sns.boxplot(x='survived_status', y='operation_year', data=haberman)

plt.subplot(133)
plt.title('boxplot for survived_status and axil_nodes')
sns.boxplot(x='survived_status', y='axil_nodes', data=haberman, hue='survived_status')
```

Out[19]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2b2d2456a20>
```



1) No major conclusion could be drawn from this plots as the data points are overlapping (i.e. scattered within the same range of values).

2) The number of axil node for survival is dense from 0-5.

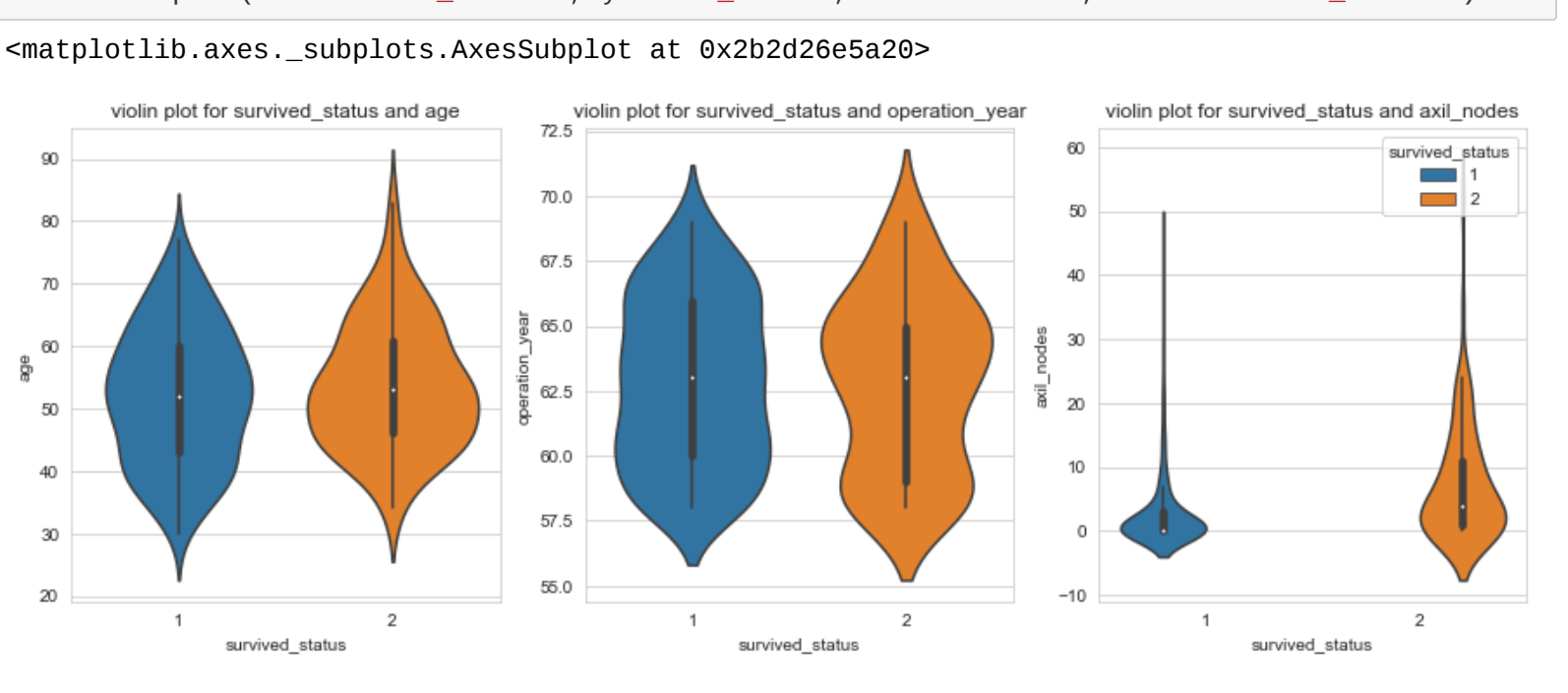
Violin plots

In [20]:
```
plt.figure(figsize=(15,5))
plt.subplot(131)
plt.title('violin plot for survived_status and age')
sns.violinplot(x='survived_status', y='age', data=haberman)

plt.subplot(132)
plt.title('violin plot for survived_status and operation_year')
sns.violinplot(x='survived_status', y='operation_year', data=haberman)

plt.subplot(133)
plt.title('violin plot for survived_status and axil_nodes')
sns.violinplot(x='survived_status', y='axil_nodes', data=haberman, hue='survived_status')
```

Out[20]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2b2d26e5a20>
```



None of the variable-pairs can help us find linearly separable clusters as the data is highly mixed up. We can't find any 'lines' and 'if-else' conditions to build a simple model to classify the survival status of the patient.

In [ ]: