In [2]:

```
import warnings
warnings.filterwarnings("ignore")
```

In [3]:

```
%cd C:\Users\Sumit\Downloads\habermans-survival-data-set
```

```
C:\Users\Sumit\Downloads\habermans-survival-data-set
```

In [4]:

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

In [5]:

```
#Loading the dataset
haberman= pd.read_csv('haberman.csv')
```

Data information

In [6]:

```
haberman.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 305 entries, 0 to 304
Data columns (total 4 columns):
30     305 non-null int64
64     305 non-null int64
1      305 non-null int64
1.1    305 non-null int64
dtypes: int64(4)
memory usage: 9.6 KB
```

In [8]:

```
haberman.shape
```

Out[8]:

```
(305, 4)
```

So we have 305 rows of entry and 4 different columns without attribute.

For categorical plotting we need to name the columns first based on their character.

In [9]:

```
# naming of columns
cancer_df = pd.read_csv('haberman.csv', header=None, names=['age', 'year_of_treatment', 'positive_l
ymph_nodes', 'survival_status_after_5_years'])
```

In [10]:

```
cancer_df.columns
```

Out[10]:

```
Index(['age', 'year_of_treatment', 'positive_lymph_nodes',
       'survival_status_after_5_years'],
      dtype='object')
```

In [11]:

```
cancer_df.head(10)
```

Out[11]:

	age	year_of_treatment	positive_lymph_nodes	survival_status_after_5_years
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	33	58	10	1
6	33	60	0	1
7	34	59	0	2
8	34	66	9	2
9	34	58	30	1

In [12]:

```
cancer_df['survival_status_after_5_years'].unique()
```

Out[12]:

```
array([1, 2], dtype=int64)
```

The dependant variable contains only two unique values such as 1 and 2.

We can change it to 0 and 1 or No and Yes for better understanding.

In [13]:

```
cancer_df['survival_status_after_5_years']=
cancer_df['survival_status_after_5_years'].apply(lambda x:'no' if x==2 else 'yes')
```

In [14]:

```
cancer_df.head(10)
```

Out[14]:

	age	year_of_treatment	positive_lymph_nodes	survival_status_after_5_years
0	30	64	1	yes
1	30	62	3	yes
2	30	65	0	yes
3	31	59	2	yes
4	31	65	4	yes
5	33	58	10	yes
6	33	60	0	yes
7	34	59	0	no
8	34	66	9	no
9	34	58	30	yes

In [15]:

```
#Counting number of 'yes' and 'no'.
cancer_df['survival_status_after_5_years'].value_counts()
```

Out[15]:

```
yes    225
no      81
Name: survival_status_after_5_years, dtype: int64
```

In [16]:

```
# Getting the percentage figure of 'yes' and 'no'
cancer_df.groupby('survival_status_after_5_years').size()/cancer_df['survival_status_after_5_years'
].count()*100
```

Out[16]:

```
survival_status_after_5_years
no     26.470588
yes    73.529412
dtype: float64
```

So this is an imbalanced dataset as surviving number is way more than non surviving number.

Surviving number is 73% while non surviving number is 26%.

Bi-variate analysis

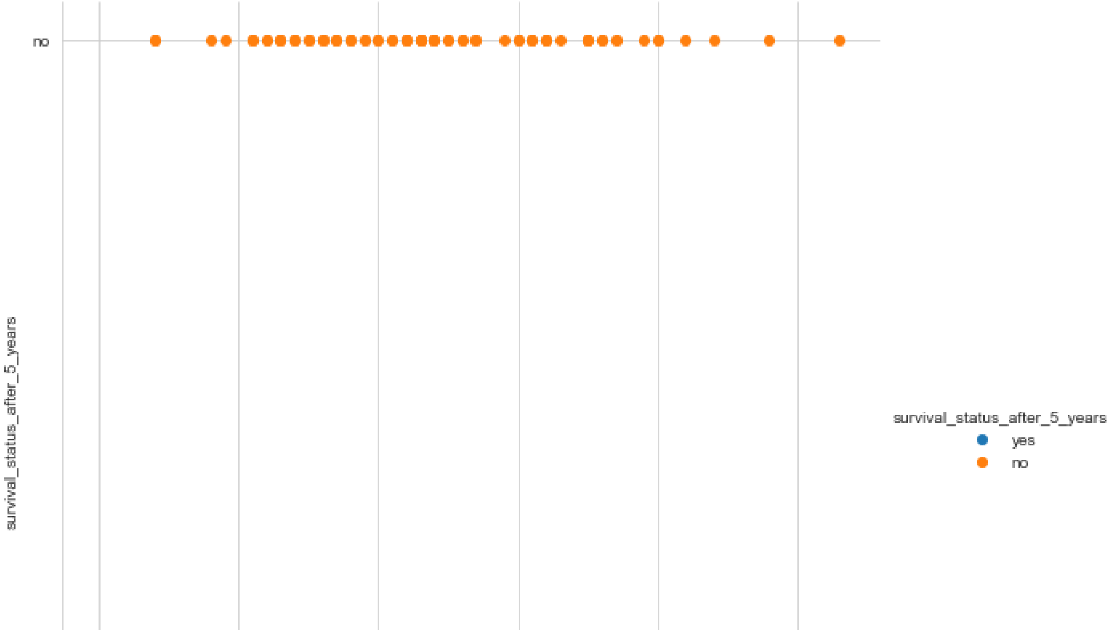
2D Scatter Plot

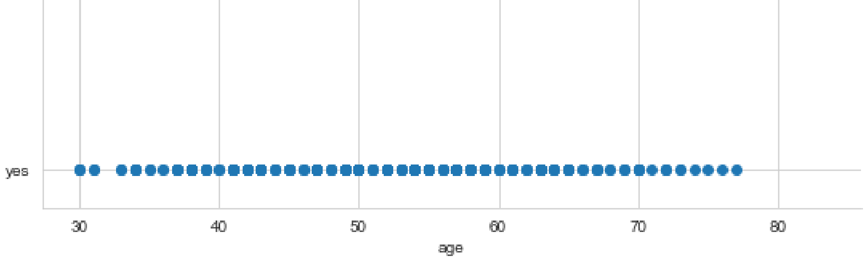
In [17]:

```
sns.set_style('whitegrid')
sns.FacetGrid(cancer_df, hue='survival_status_after_5_years', height=8).map(plt.scatter, 'age', 'su
rvival_status_after_5_years').add_legend()
```

Out[17]:

```
<seaborn.axisgrid.FacetGrid at 0x824bd90>
```





Patients aged betwwen 78 to 80+ could not survive more than 5 years.

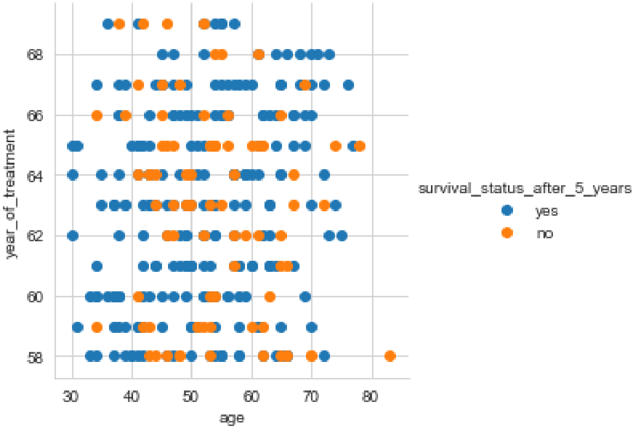
Patients aged betwwen 30 to 33 survived more than 5 years.

In [18]:

```
sns.set_style('whitegrid')
sns.FacetGrid(cancer_df, hue='survival_status_after_5_years', height=4).map(plt.scatter, 'age', 'ye
ar_of_treatment').add_legend()
```

Out[18]:

```
<seaborn.axisgrid.FacetGrid at 0x82f0810>
```



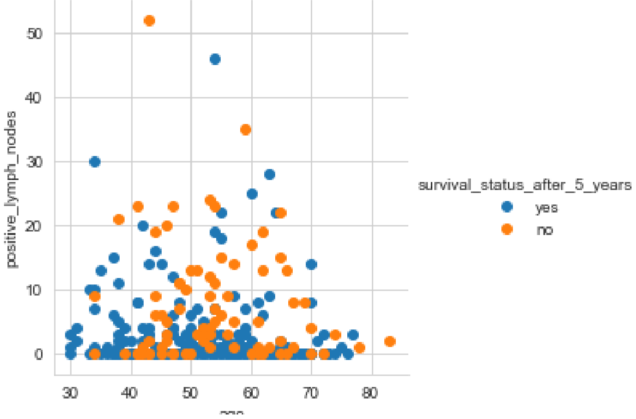
Very diverse plot. can not judge anything.

In [19]:

```
sns.set_style('whitegrid')
sns.FacetGrid(cancer_df, hue='survival_status_after_5_years', height=4).map(plt.scatter, 'age', 'po
sitive_lymph_nodes').add_legend()
```

Out[19]:

```
<seaborn.axisgrid.FacetGrid at 0x837efb0>
```



age

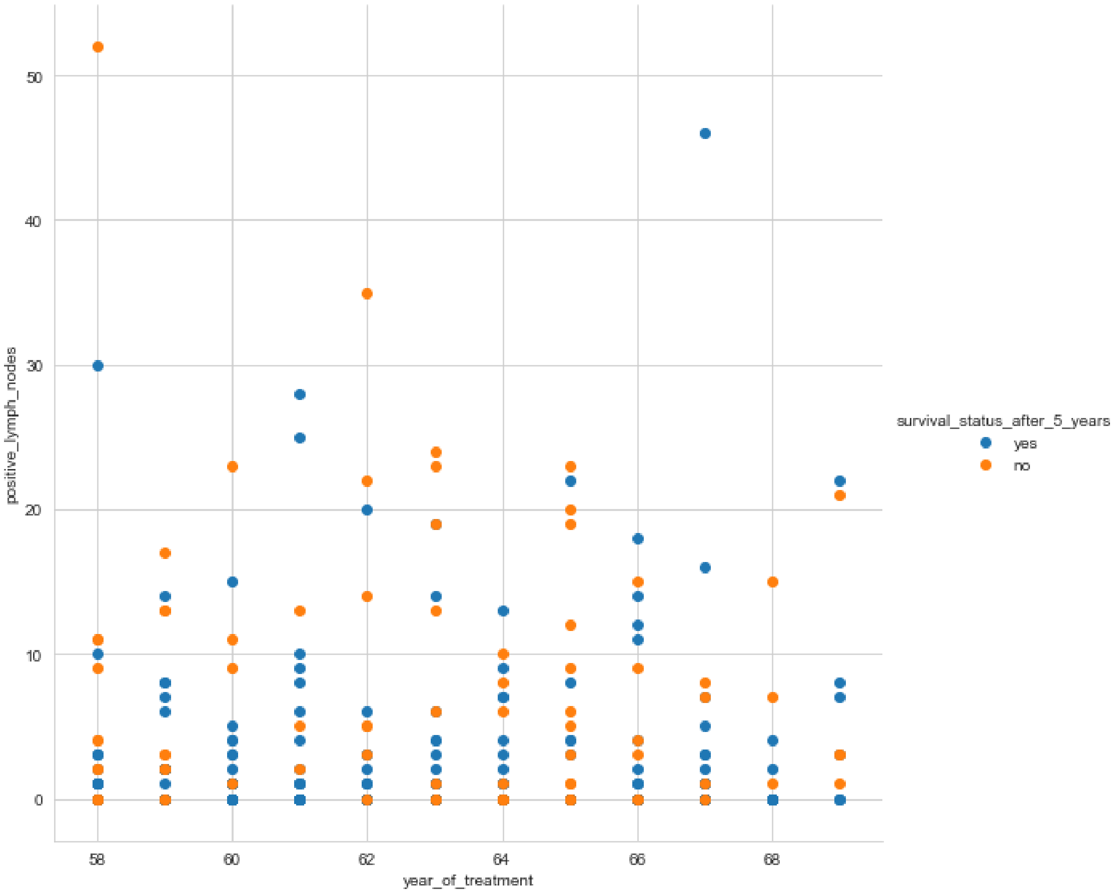
Patients having positive lymph nodes and aged between 78 to 80+ could not survive more than 5 years.

In [17]:

```
sns.set_style('whitegrid')
sns.FacetGrid(cancer_df, hue='survival_status_after_5_years', height=8).map(plt.scatter, 'year_of_t
reatment', 'positive_lymph_nodes').add_legend()
```

Out[17]:

```
<seaborn.axisgrid.FacetGrid at 0x860b6f0>
```



In 1961, with positive lymph nodes upto 28 more patients survived more than 5 years compared to the deaths within 5 years.

In 1965, with positive lymph nodes upto 24 most patients could not survived more than 5 years.

Patient having positive lymph nodes more than 50 could not survive more than 5 years.

Patients having positive lymph nodes from 31 to 52+ have surviving chance approximately 1/3 i.e 33.33%.

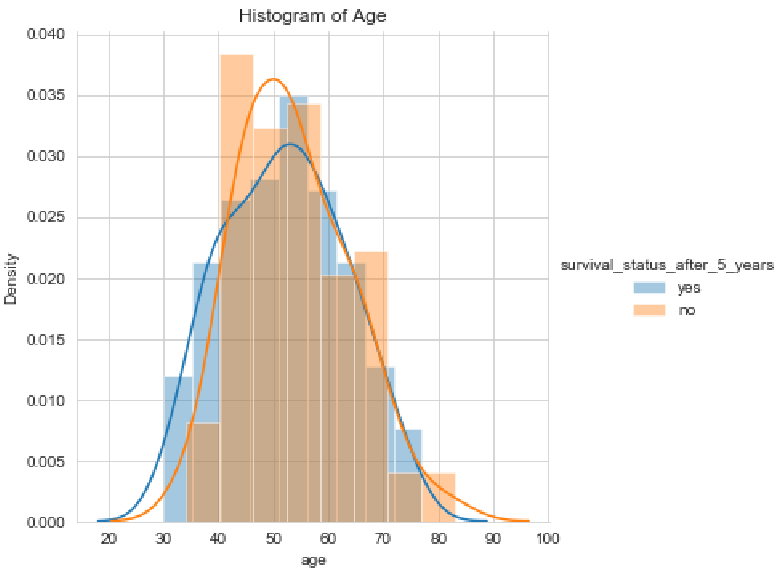
Histogram

In [18]:

```
sns.FacetGrid(cancer_df, hue='survival_status_after_5_years', height=5).map(sns.distplot,'age').add
_legend()
plt.title('Histogram of Age')
plt.ylabel('Density')
plt.ylabel('Density')
```

Out[18]:

```
Text(23.677369791666678, 0.5, 'Density')
```



Its almost completely overlapping each other but somehow we can say greater number of patients approx 38% at the age of 40 to 50 did not survive after 5 years.

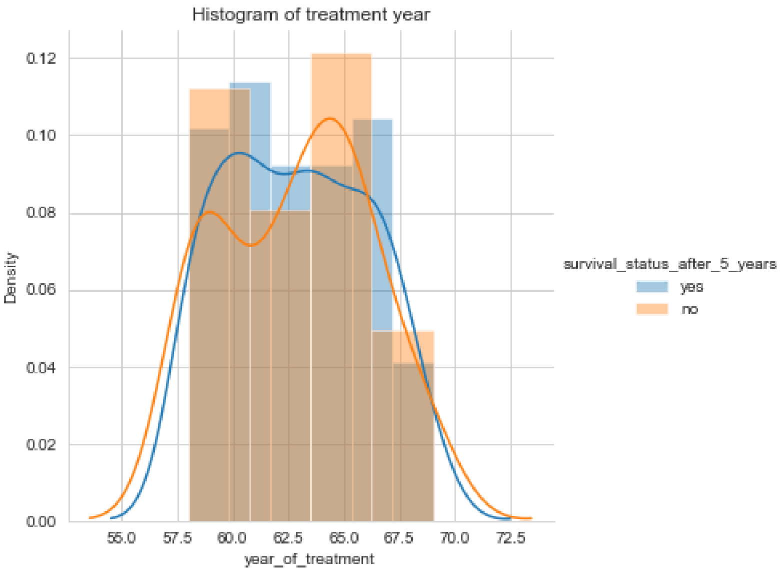
Can not classify because of the huge overlapping.

In [19]:

```
sns.FacetGrid(cancer_df,hue='survival_status_after_5_years',height=5,).map(sns.distplot,'year_of_tr
eatment').add_legend()
plt.title('Histogram of treatment year')
plt.ylabel('Density')
```

Out[19]:

```
Text(21.551631944444452, 0.5, 'Density')
```



Its also massively overlapping each other but we can see a peak of non survival in 1964.

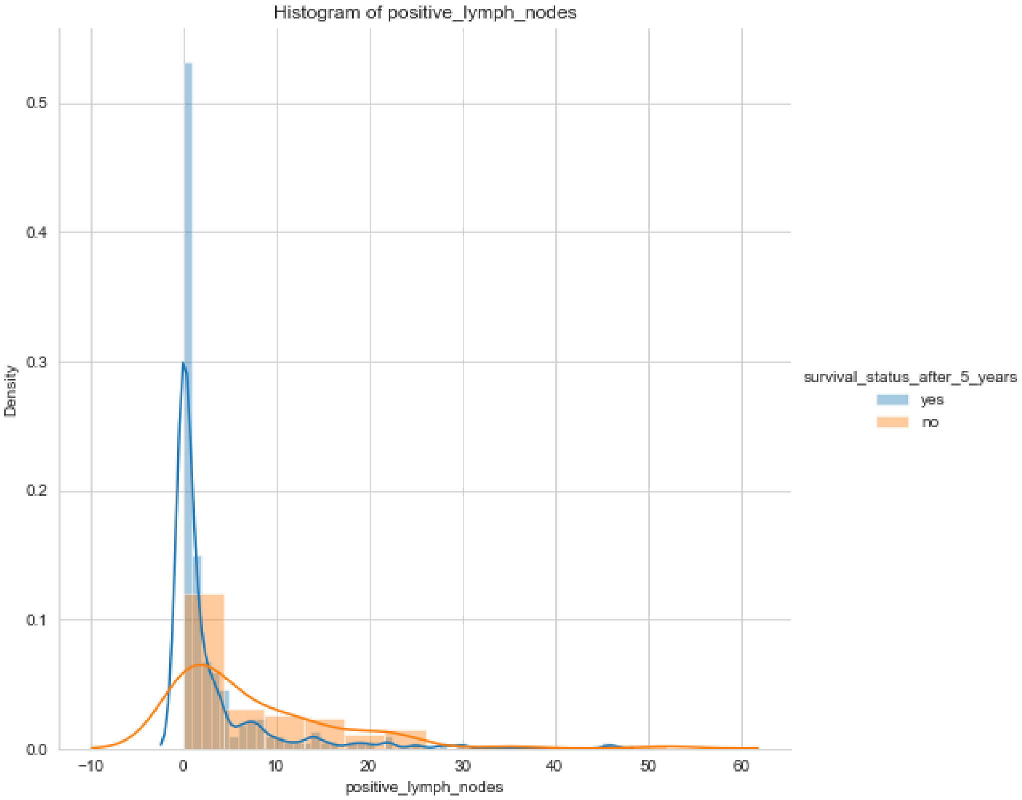
Can not classify because of the huge overlapping.

In [20]:

```
sns.FacetGrid(cancer_df, hue = "survival_status_after_5_years", height = 7).map(sns.distplot, "positive_lymph_nodes").add_legend()
plt.title("Histogram of positive_lymph_nodes")
plt.ylabel("Density")
```

Out[20]:

```
Text(15.747873263888891, 0.5, 'Density')
```



With more positive lymph nodes non survival is greater compared to survival.

Patients with less lymph nodes i.e near to zero have chances of more survival, nearly 30% patients based on PDF.

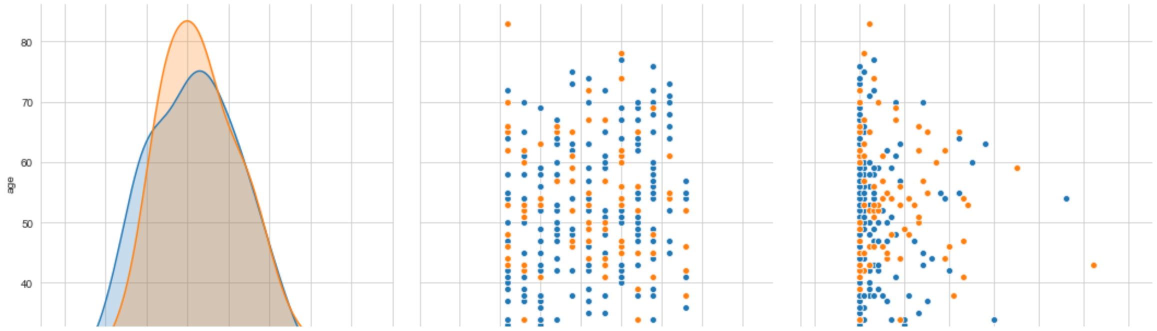
Pair plot

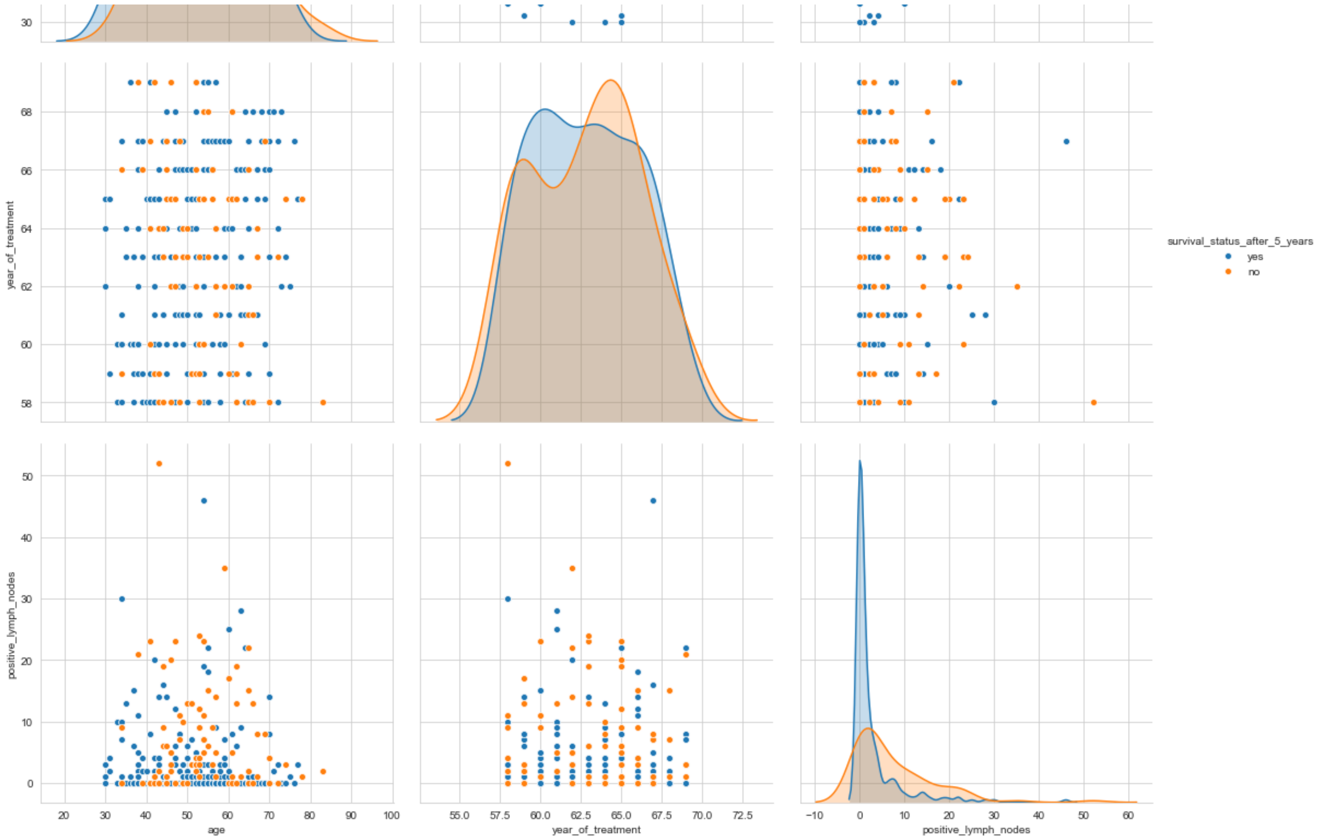
In [21]:

```
sns.set_style("whitegrid")
sns.pairplot(cancer_df, hue='survival_status_after_5_years', height=5)
```

Out[21]:

```
<seaborn.axisgrid.PairGrid at 0x8be3650>
```





Pairplot says the same story but as a whole.

Univariate analysis

Probablity Density Function (PDF)

In [ ]:

```
# PDF plottning using Numpy
```

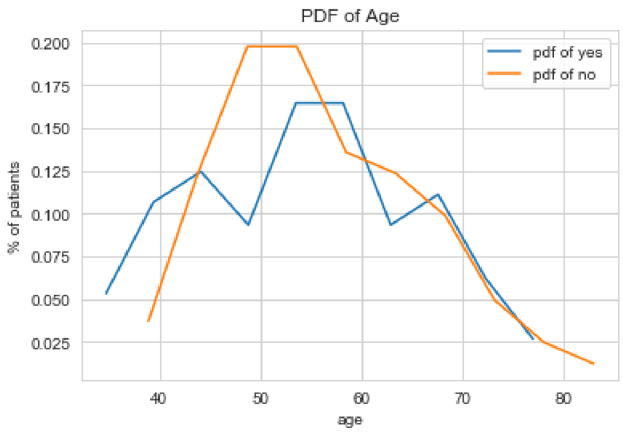
In [22]:

```
yes = cancer_df.loc[cancer_df['survival_status_after_5_years'] == 'yes']
no = cancer_df.loc[cancer_df['survival_status_after_5_years'] == 'no']
Legend=['pdf of yes', 'pdf of no']
counts, bin_edges = np.histogram(yes['age'], bins=10, density = True)
pdf = counts/(sum(counts))
print('yes=',bin_edges)
plt.plot(bin_edges[1:],pdf)


counts, bin_edges = np.histogram(no['age'], bins=10, density = True)
pdf = counts/(sum(counts))
print('no=',bin_edges)
plt.plot(bin_edges[1:],pdf)
plt.title('PDF of Age')
plt.legend(Legend)
plt.xlabel('age')
plt.ylabel('% of patients')
```

```
yes= [30.  34.7 39.4 44.1 48.8 53.5 58.2 62.9 67.6 72.3 77. ]
no= [34.  38.9 43.8 48.7 53.6 58.5 63.4 68.3 73.2 78.1 83. ]
```

Out[22]:

```
Text(0, 0.5, '% of patients')
```



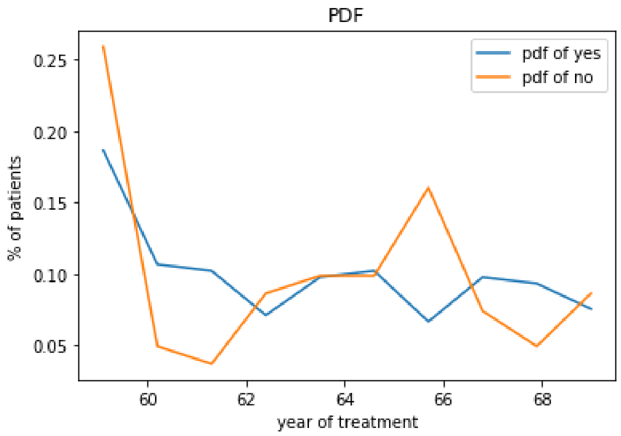
In [16]:

```
Legend=['pdf of yes', 'pdf of no']

counts, bin_edges = np.histogram(yes['year_of_treatment'], bins=10, density = True)
pdf = counts/(sum(counts))
print('yes=',bin_edges)
plt.plot(bin_edges[1:],pdf)


counts, bin_edges = np.histogram(no['year_of_treatment'], bins=10, density = True)
pdf = counts/(sum(counts))
print('no=',bin_edges)
plt.plot(bin_edges[1:],pdf)

plt.title('PDF')
plt.legend(Legend)

plt.xlabel('year of treatment')
plt.ylabel('% of patients')
```

```
yes= [58.   59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
no= [58.   59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
```

Out[16]:

```
Text(0, 0.5, '% of patients')
```



In [23]:

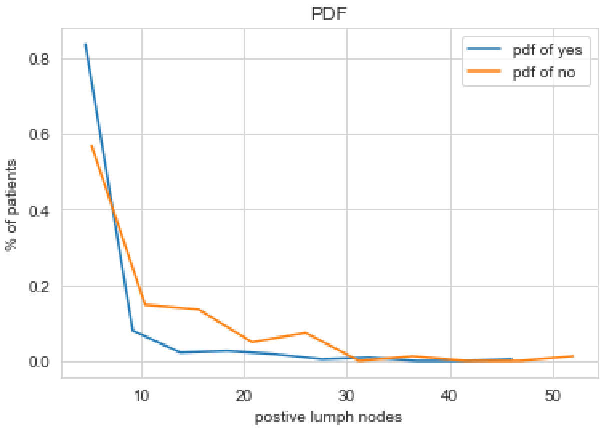
```
Legend=['pdf of yes', 'pdf of no']

counts, bin_edges = np.histogram(yes['positive_lymph_nodes'], bins=10, density = True)
pdf = counts/(sum(counts))
print('yes=',bin_edges)
plt.plot(bin_edges[1:],pdf)


counts, bin_edges = np.histogram(no['positive_lymph_nodes'], bins=10, density = True)
```

```
pdf = counts/(sum(counts))
print('no=',bin_edges)
plt.plot(bin_edges[1:],pdf)

plt.title('PDF')
plt.legend(Legend)

plt.xlabel('postive lumph nodes')
plt.ylabel('% of patients')
```

```
yes= [ 0.   4.6  9.2 13.8 18.4 23.  27.6 32.2 36.8 41.4 46. ]
no= [ 0.   5.2 10.4 15.6 20.8 26.  31.2 36.4 41.6 46.8 52. ]
```

Out[23]:

```
Text(0, 0.5, '% of patients')
```



These PDF plots are same like PDFs in Histogram plotting.

PDF plots show massive overlapping. Can not classify.

Cumulative distribution function (CDF)

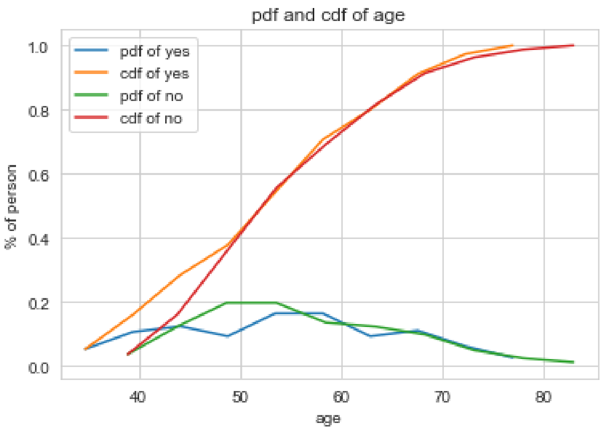
In [24]:

```
Legend=['pdf of yes', 'cdf of yes', 'pdf of no', 'cdf of no']
counts, bin_edges = np.histogram(yes['age'], bins=10, density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
print('PDF of yes=', pdf)
print('CDF of yes=', cdf)
plt.plot(bin_edges[1:], pdf)
plt.plot(bin_edges[1:], cdf)

counts, bin_edges = np.histogram(no['age'], bins=10, density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
print('PDF of no=', pdf)
print('CDF of no=', cdf)
plt.plot(bin_edges[1:], pdf)
plt.plot(bin_edges[1:], cdf)

plt.title('pdf and cdf of age')
plt.xlabel('age')
plt.ylabel('% of person')
plt.legend(Legend)
```

```
PDF of yes= [0.05333333 0.10666667 0.12444444 0.09333333 0.16444444 0.16444444
 0.09333333 0.11111111 0.06222222 0.02666667]
CDF of yes= [0.05333333 0.16       0.28444444 0.37777778 0.54222222 0.70666667
 0.8        0.91111111 0.97333333 1.        ]
PDF of no= [0.03703704 0.12345679 0.19753086 0.19753086 0.13580247 0.12345679
 0.09876543 0.04938272 0.02469136 0.01234568]
CDF of no= [0.03703704 0.16049383 0.35802469 0.55555556 0.69135802 0.81481481
CDF of no= [0.03703704 0.16049383 0.35802469 0.55555556 0.69135802 0.81481481
 0.91358025 0.96296296 0.98765432 1.        ]
```

Out[24]:

```
<matplotlib.legend.Legend at 0xac80a70>
```



Not a very good CDF but we can see the positive survival curve is little higher upto age 48.

That means greater chance of survival.

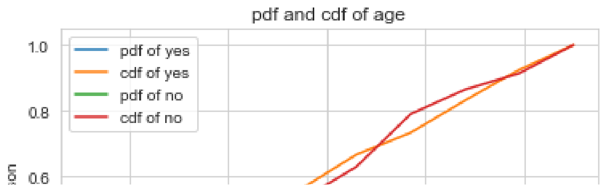
In [23]:

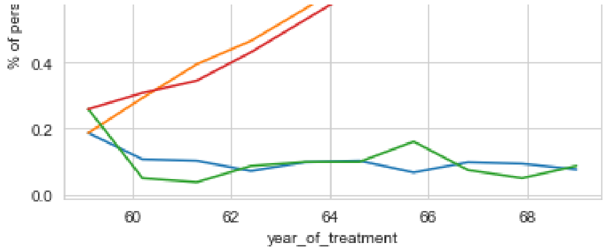
```python
Legend=['pdf of yes', 'cdf of yes', 'pdf of no', 'cdf of no']
counts, bin_edges = np.histogram(yes['year_of_treatment'], bins=10, density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
print('PDF of yes=', pdf)
print('CDF of yes=', cdf)
plt.plot(bin_edges[1:], pdf)
plt.plot(bin_edges[1:], cdf)

counts, bin_edges = np.histogram(no['year_of_treatment'], bins=10, density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
print('PDF of no=', pdf)
print('CDF of no=', cdf)
plt.plot(bin_edges[1:], pdf)
plt.plot(bin_edges[1:], cdf)

plt.title('pdf and cdf of age')
plt.xlabel('year_of_treatment')
plt.ylabel("% of person")
plt.legend(Legend)
```

```
PDF of yes= [0.18666667 0.10666667 0.10222222 0.07111111 0.09777778 0.10222222
 0.06666667 0.09777778 0.09333333 0.07555556]
CDF of yes= [0.18666667 0.29333333 0.39555556 0.46666667 0.56444444 0.66666667
 0.73333333 0.83111111 0.92444444 1.        ]
PDF of no= [0.25925926 0.04938272 0.03703704 0.08641975 0.09876543 0.09876543
 0.16049383 0.07407407 0.04938272 0.08641975]
CDF of no= [0.25925926 0.30864198 0.34567901 0.43209877 0.5308642  0.62962963
 0.79012346 0.86419753 0.91358025 1.        ]
```

Out[23]:

```
<matplotlib.legend.Legend at 0xa916510>
```





A lot of ups and downs. Not a decisive CDF.

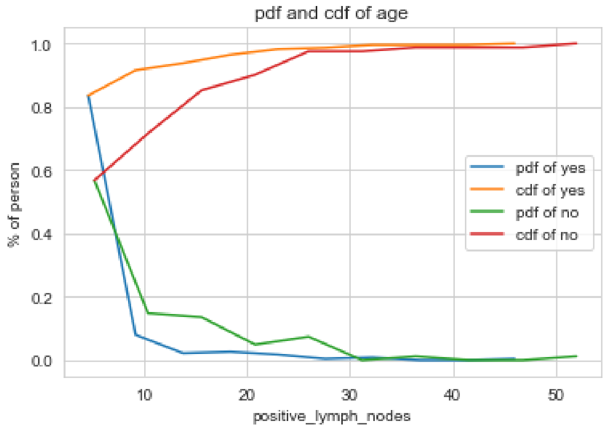
In [25]:

```
Legend=['pdf of yes', 'cdf of yes', 'pdf of no', 'cdf of no']

counts, bin_edges = np.histogram(yes['positive_lymph_nodes'], bins=10, density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
print('PDF of yes=', pdf)
print('CDF of yes=', cdf)
plt.plot(bin_edges[1:], pdf)
plt.plot(bin_edges[1:], cdf)

counts, bin_edges = np.histogram(no['positive_lymph_nodes'], bins=10, density = True)
pdf = counts/(sum(counts))
cdf = np.cumsum(pdf)
print('PDF of no=', pdf)
print('CDF of no=', cdf)
plt.plot(bin_edges[1:], pdf)
plt.plot(bin_edges[1:], cdf)

plt.title('pdf and cdf of age')
plt.xlabel('positive_lymph_nodes')
plt.ylabel('% of person')
plt.legend(Legend)
```

```
PDF of yes= [0.83555556 0.08       0.02222222 0.02666667 0.01777778 0.00444444
 0.00888889 0.         0.         0.00444444]
CDF of yes= [0.83555556 0.91555556 0.93777778 0.96444444 0.98222222 0.98666667
 0.99555556 0.99555556 0.99555556 1.        ]
PDF of no= [0.56790123 0.14814815 0.13580247 0.04938272 0.07407407 0.
 0.01234568 0.         0.         0.01234568]
CDF of no= [0.56790123 0.71604938 0.85185185 0.90123457 0.97530864 0.97530864
 0.98765432 0.98765432 0.98765432 1.        ]
```

Out[25]:

```
<matplotlib.legend.Legend at 0xacc7d30>
```



Positive lymph nodes less than 10 have higher chance of survival as the percentage is over 80%.

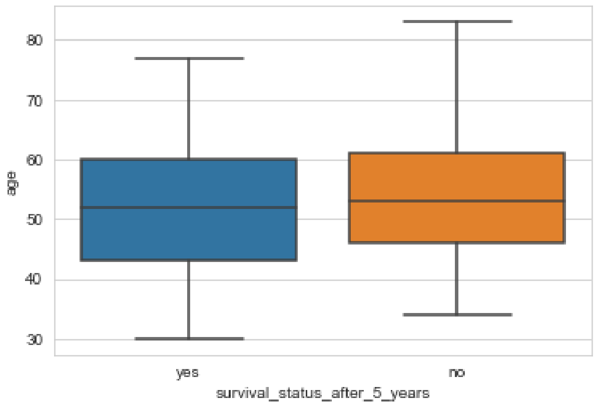
Boxplot and Whiskers

In [28]:

```
sns.boxplot(x='survival_status_after_5_years',y='age', data=cancer_df)
```

Out[28]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xacb3e50>
```



We can observe ages of survived patients lie between 30 to 77 approx and 75% of them are upto 60

Ages of died patients lie between 34 to 85 approx and 75% of them are under 62

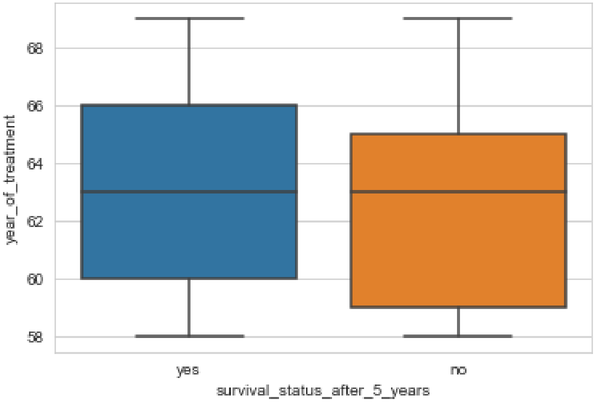
That means patients with age between 30 to 33 are more likely to survive.

In [29]:

```
sns.boxplot(x='survival_status_after_5_years',y='year_of_treatment', data=cancer_df)
```

Out[29]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xadbadb0>
```



In both cases the range of treatment year is same.

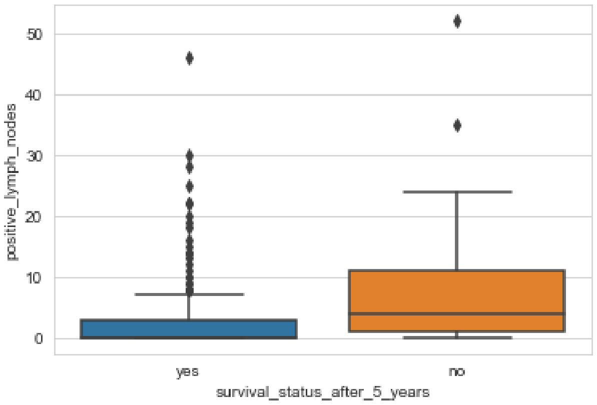
In the year 1965-66, patients are more likely to survive however before 1960 patients had more chance of death.

In [30]:

```
sns.boxplot(x='survival_status_after_5_years',y='positive_lymph_nodes', data=cancer_df)
```

Out[30]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xc1daff0>
```



There are many outliers that makes difficult for any decision.

However we can patients with zero lymph node survived but patients with more than 3 lymph nodes are more likely to die.

Most of the survivors have less than 8 lymph nodes.

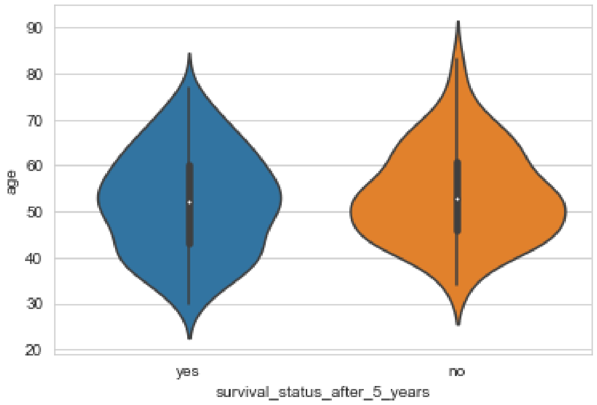
Violin plots

In [31]:

```
sns.violinplot(x='survival_status_after_5_years', y='age', data=cancer_df, size=8)
```

Out[31]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xc219bb0>
```

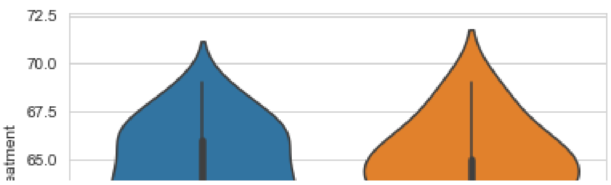


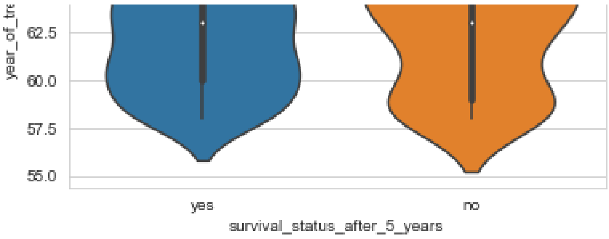
In [32]:

```
sns.violinplot(x='survival_status_after_5_years', y='year_of_treatment', data=cancer_df, size=8)
```

Out[32]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xc2563d0>
```



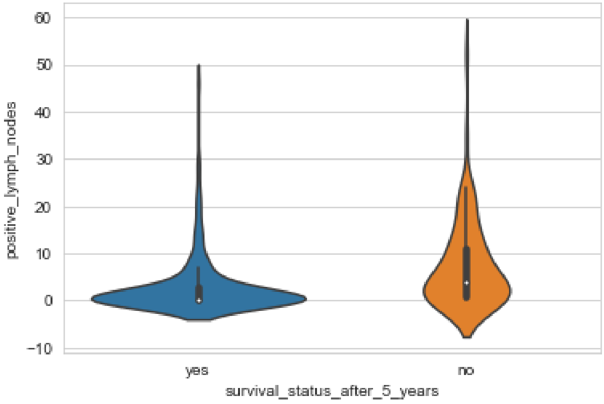


In [33]:

```
sns.violinplot(x='survival_status_after_5_years', y='positive_lymph_nodes', data=cancer_df, size=8)
```

Out[33]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xc285370>
```



Violins also state the same results as Boxplots and Whiskers.

Conclusion

The dataset is very much versatile. plots are overlapping each other.

We can hardly say that Patients with aged less than 40 or more specifically 30 to 33 and having less (less than or equal to 3) positive lymph nodes can survive.

Patients with more age and more positive lymph nodes have less chance to survive more than 5 years.

Patients aged more than 70 having positive lymph nodes are not likely to survive.

But the dataset is very hard to classify directly.

Maybe some pruning can convert the dataset into an eligible dataset for applying decisive algorithms.