3. Plotting for Exploratory data analysis (EDA)

(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 (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.

In [5]:

```
#Importing Libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
```

In [48]:

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

Out[48]:

	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

Description , shape and feature-size of the dataset

There are 4 columns in the dataset.
Age - Age of the person during operation
Year - Year during the operation
Nodes - No. of positive auxillary nodes
Status - Survived (1) and Dead (2)

In [7]:

```
# Description of columns of dataset
df.describe()
```

Out[7]:

	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

In [8]:

```
#Shape of the dataset
df.shape
```

Out[8]:

```
(306, 4)
```

In [9]:

```
# No. of patients in each class
df['status'].value_counts()
```

Out[9]:

```
1    225
2     81
Name: status, dtype: int64
```

The data is imbalance, as there are unequal datapoints for each class.

Of total 306 patients, 225 survived and 81 were dead.

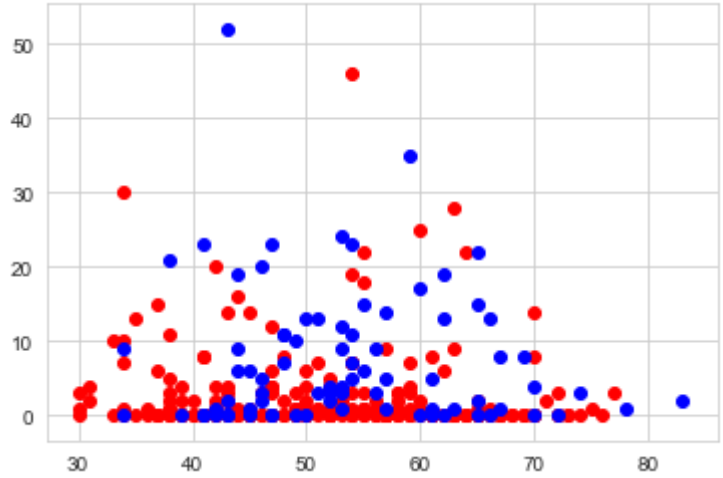
Objective :

To classify patients if they survived or were dead based on the features.

2D Scatterplot

In [32]:

```
plt.scatter(df['age'].where(df['status']==1).tolist(),df['nodes'].where(df['status']==1
).tolist(),color='red')
plt.scatter(df['age'].where(df['status']==2).tolist(),df['nodes'].where(df['status']==2
).tolist(),color='blue')
plt.show()
```



Observations:

The features age and nodes are plotted based on the status of survival but we can't see some pattern in data.

2D Pairplots

In [34]:

```
sns.pairplot(df, hue='status')
plt.show()
```

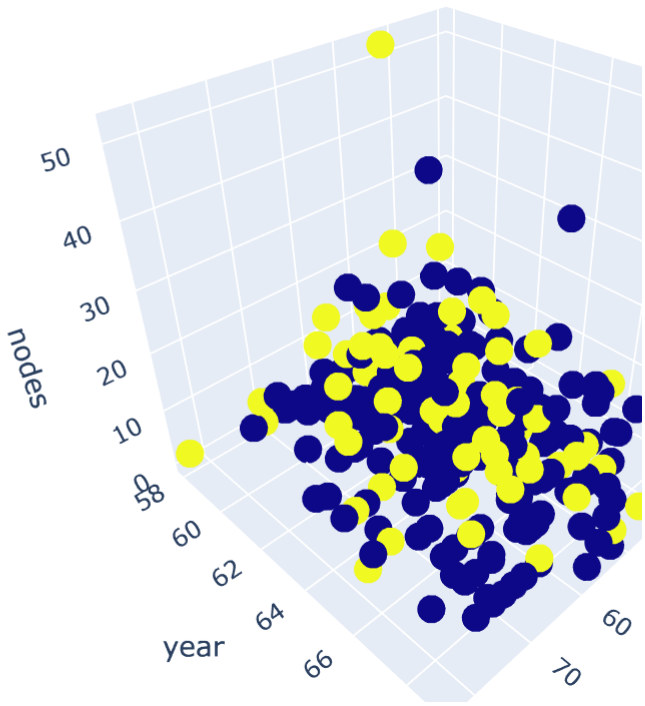
Observations:

Pairplots shows the scatterplots of the features and also the PDFs but directly we can't find any pattern in it.

3D Scatterplot

In [36]:

```
fig = px.scatter_3d(df, x='age', y='year', z='nodes',
                    color='status')
fig.show()
```



Observation:

3d scatterplot also doesn't provide us any significant information.

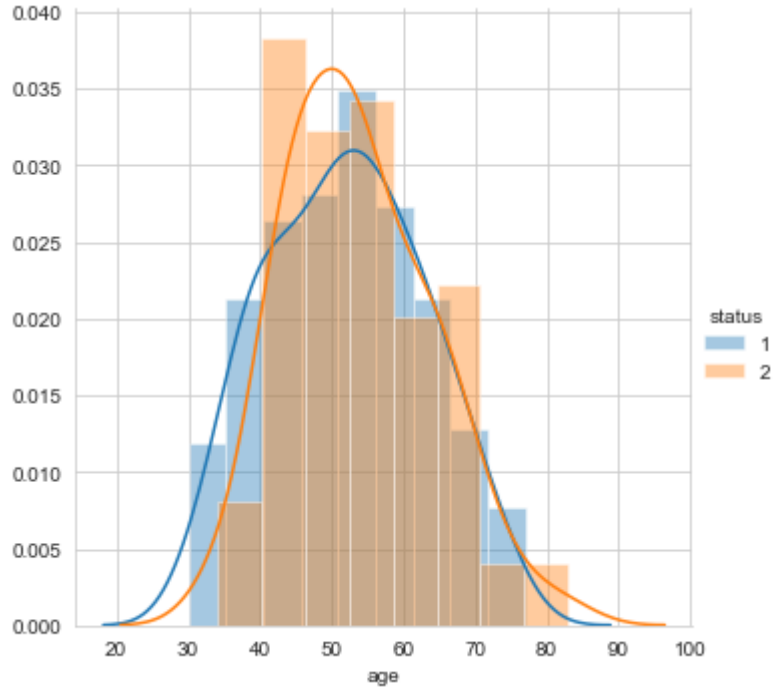
Histogram and PDF

In [38]:

```
sns.FacetGrid(df, hue="status", size=5) \
   .map(sns.distplot, "age") \
   .add_legend();
plt.show();
```

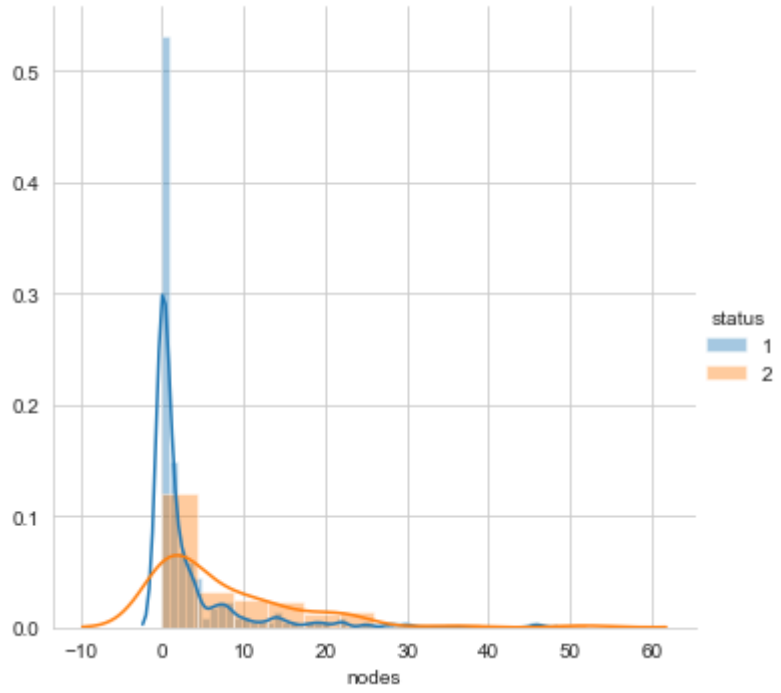
```
C:\Users\bishta\AppData\Local\Continuum\anaconda3\lib\site-packages\seabor
n\axisgrid.py:243: UserWarning:

The `size` parameter has been renamed to `height`; please update your cod
e.
```



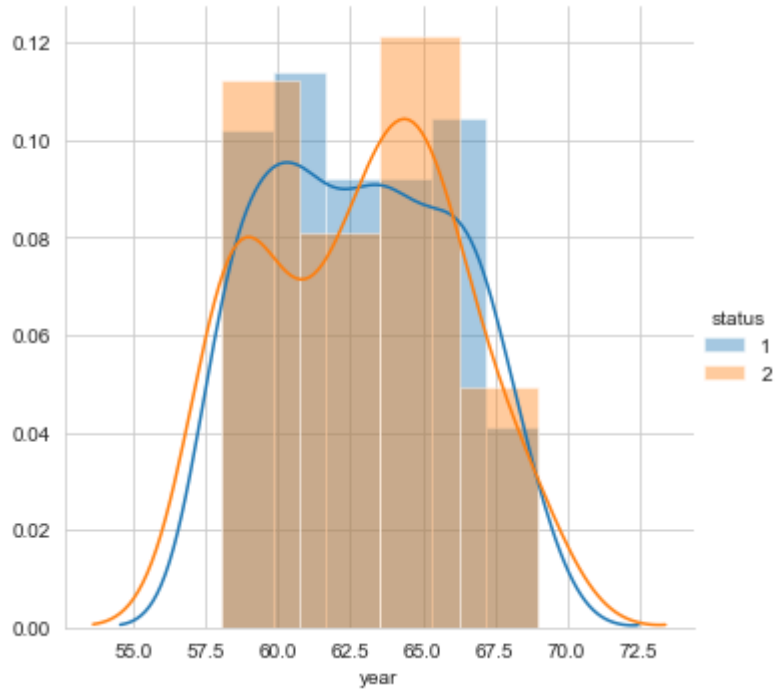
In [39]:

```
sns.FacetGrid(df, hue="status", size=5) \
   .map(sns.distplot, "nodes") \
   .add_legend();
plt.show();
```



In [40]:

```
sns.FacetGrid(df, hue="status", size=5) \
   .map(sns.distplot, "year") \
   .add_legend();
plt.show();
```



CDF

In [45]:

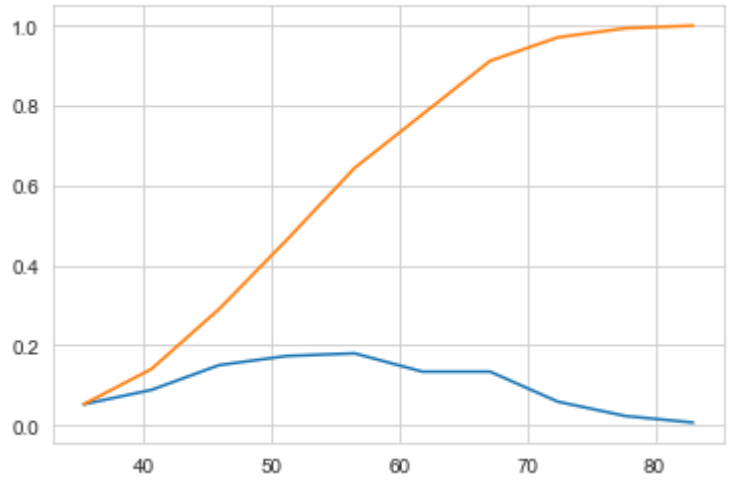
```
counts, bin_edges = np.histogram(df['age'], bins=10,
                                 density = True)
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges)

#compute CDF
cdf = np.cumsum(pdf)
plt.plot(bin_edges[1:],pdf)
plt.plot(bin_edges[1:], cdf)



plt.show();
```

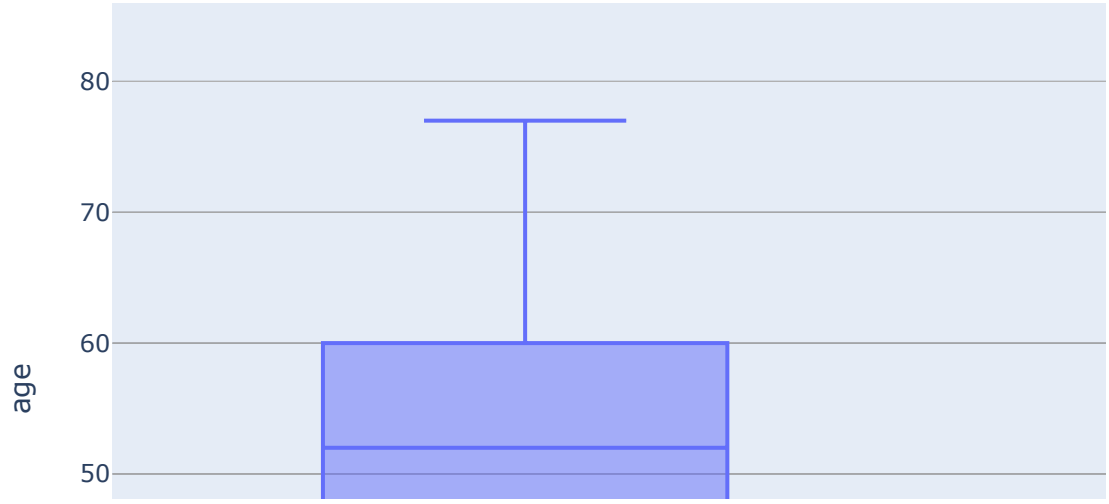
```
[0.05228758 0.08823529 0.1503268  0.17320261 0.17973856 0.13398693
 0.13398693 0.05882353 0.02287582 0.00653595]
[30.  35.3 40.6 45.9 51.2 56.5 61.8 67.1 72.4 77.7 83. ]
```



Box plots

In [53]:

```
fig = px.box(df,x='status', y="age")
fig.show()
```



In [52]:

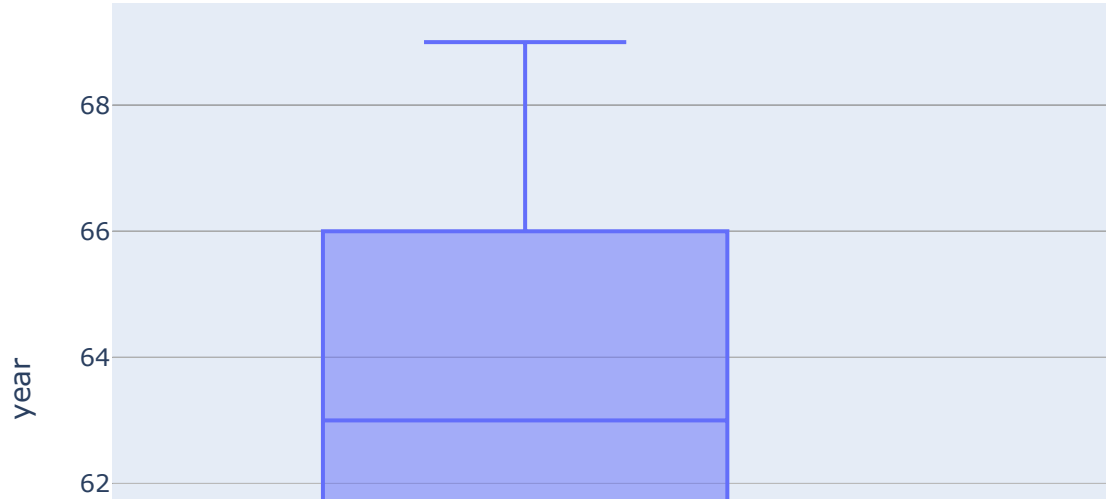
```
fig = px.box(df,x='status', y="nodes")
fig.show()
```

Observations:

Boxplots seem to show slightly some pattern in feature node like overlapping but, patient dead tend to contain more no. of nodes

In [54]:

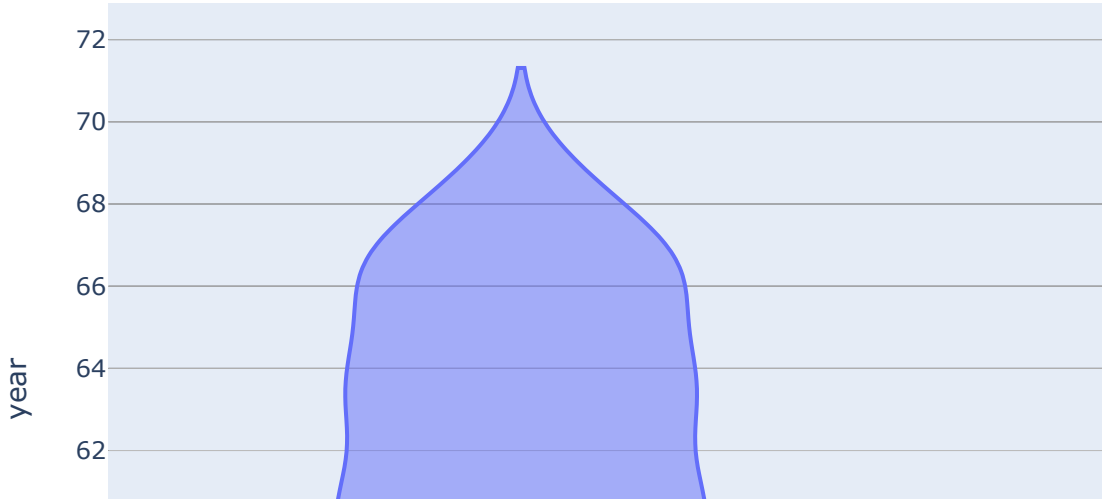
```
fig = px.box(df,x='status', y="year")
fig.show()
```



Violin Plots

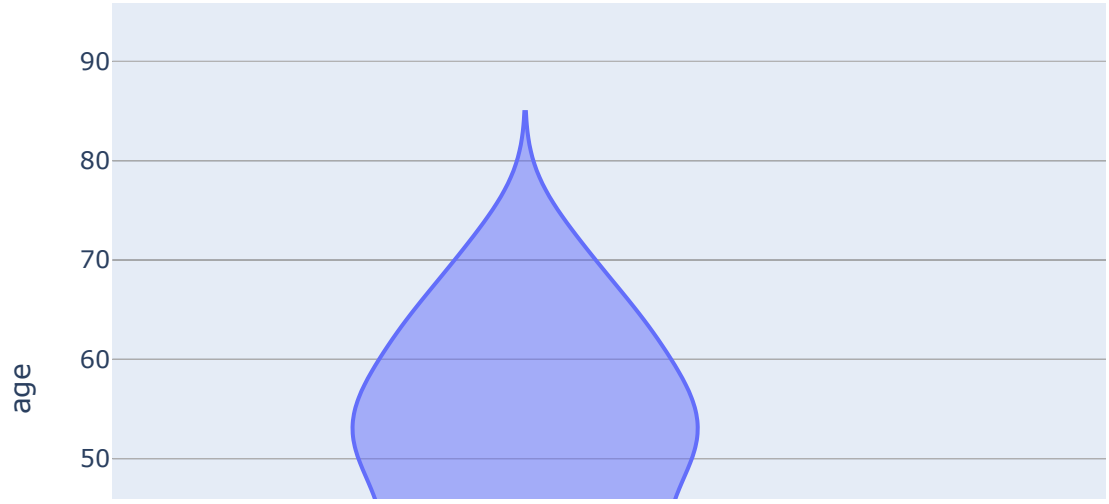
In [55]:

```
fig = px.violin(df,x='status', y="year")
fig.show()
```



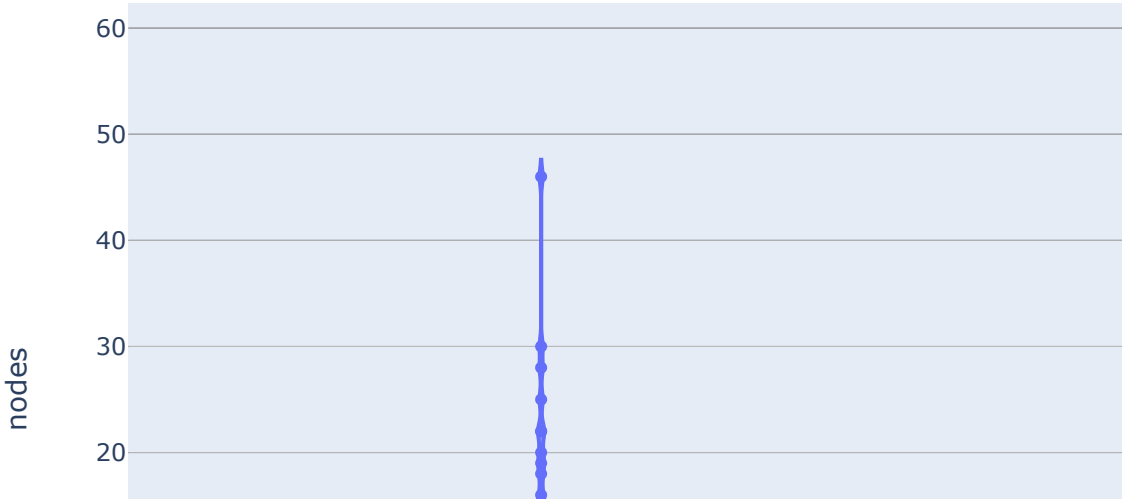
In [56]:

```
fig = px.violin(df,x='status', y="age")
fig.show()
```



In [57]:

```
fig = px.violin(df,x='status', y="nodes")
fig.show()
```



Final Observations

1. We perform the EDA over Haberman's dataset and tried to find out insights of data
2. We also tried to check patterns through univariate and bivariate analysis but couldn't find any useful insights from it

In [ ]:

```

```