Objective::- To determine the status of a person after the operation

In [1]:

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

warnings.filterwarnings("ignore")
haberman = pd.read_csv("C:/Users/juilee/Downloads/haberman.csv")
#print(haberman)
```

In [2]:

```
type(haberman)
```

Out[2]:

```
pandas.core.frame.DataFrame
```

In [3]:

```
haberman.shape
```

Out[3]:

```
(306, 4)
```

Observation::

Haberman dataset have 306 data points with 4 variables

In [4]:

```
haberman.columns
```

Out[4]:

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

Observation::

4 variables are 1)age 2)year 3)nodes 4)status

In [5]:

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

Out[5]:

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

In [6]:

```
haberman["status"]= haberman["status"].apply(lambda x: "survived" if x==1 else "died")
haberman["status"].value_counts()
```

Out[6]:

```
survived    225
died         81
Name: status, dtype: int64
```

Observation::

Out of total 306 cancer patients 225 patients have survived after the operation and 81 patients have died.

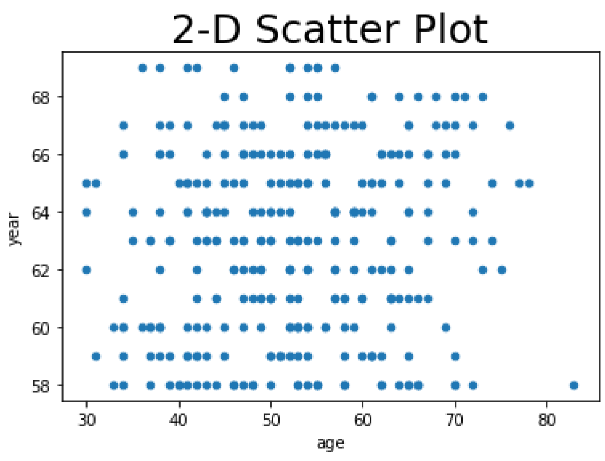
In [ ]:

```
```

Bi-Variate Analysis

In [7]:

```
haberman.plot(kind = "scatter",x="age", y="year")
plt.title("2-D Scatter Plot",size=25)
plt.show()
```

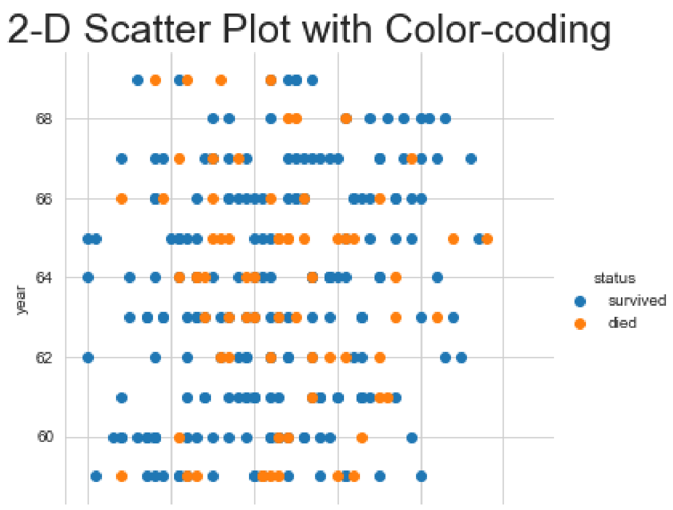


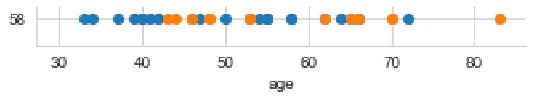
In [ ]:

```
```

In [8]:

```
sns.set_style("whitegrid")
sns.FacetGrid(haberman,hue="status",height = 5).map(plt.scatter,"age","year").add_legend()
plt.title("2-D Scatter Plot with Color-coding",size=25)
plt.show()
```



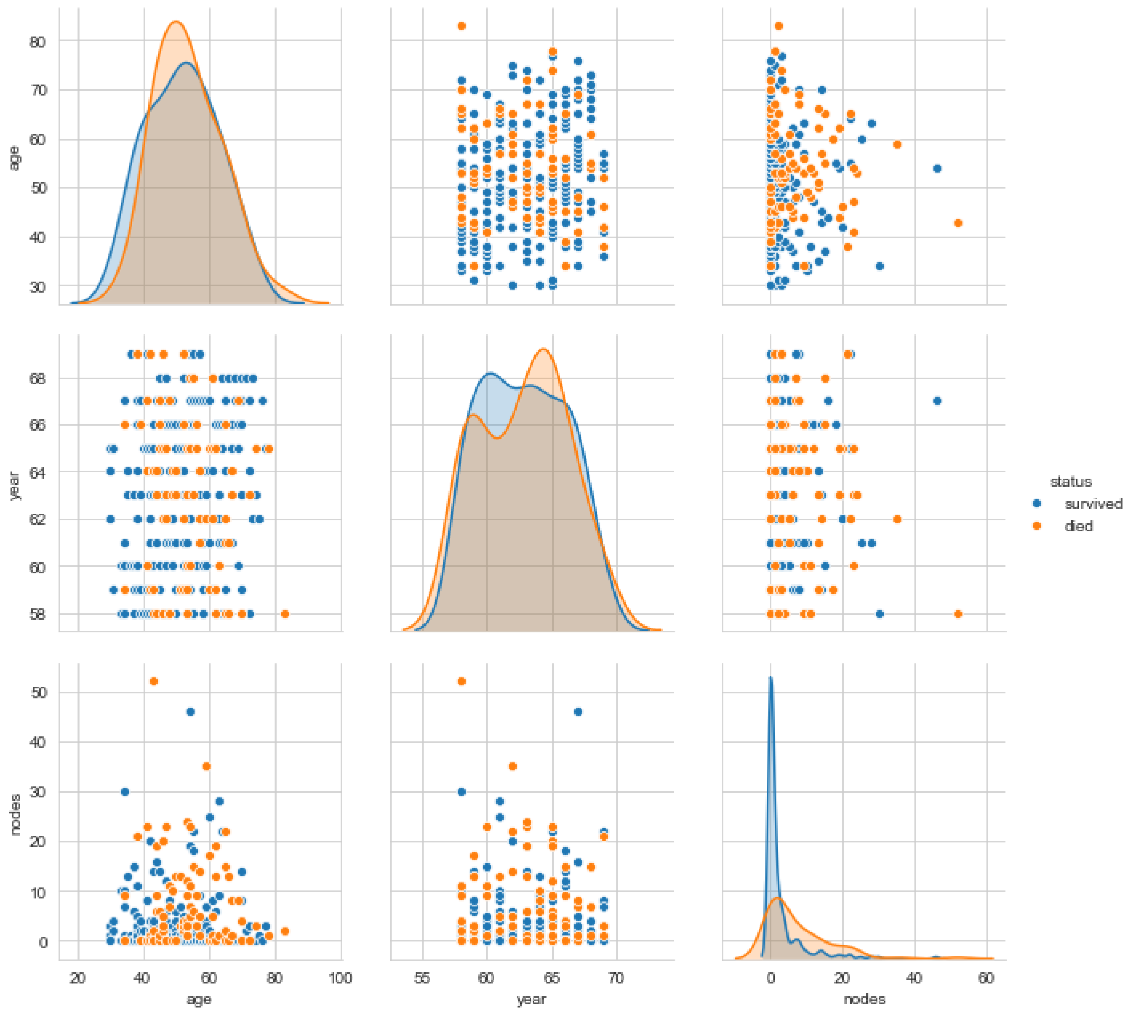


```
In [ ]:
```

Pair-plot

```
In [9]:
```

```
sns.set_style("whitegrid")
sns.pairplot(haberman,hue="status",height=3)
plt.show()
```

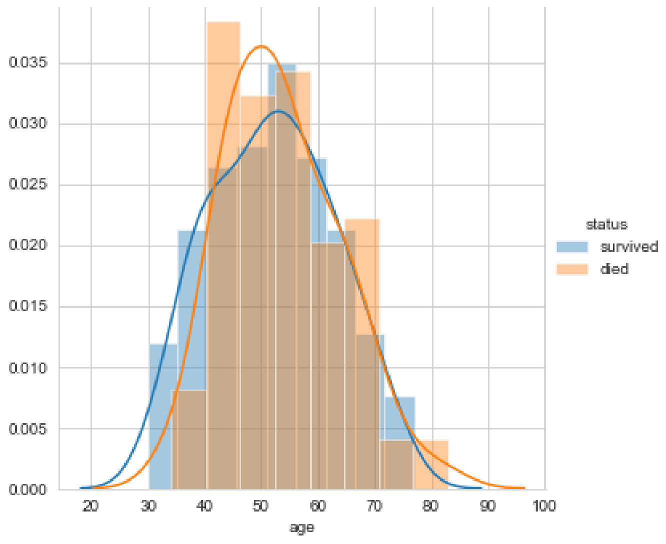


Observation

1) We can not determine whether the patient has survived or not after the operation; by visualizing these Pair-Plots easily.
2) We can not draw a line to seprate "survived" and "died" patients. Hence can not write if-else conditions to build asimple model.
3) Pair plots can be used when number features are high.

```
In [10]:
```

```
sns.FacetGrid(haberman,hue="status",height=5).map(sns.distplot,"age").add_legend()
plt.title("Univaraite Analysis of Age",size=25)
plt.show()
```

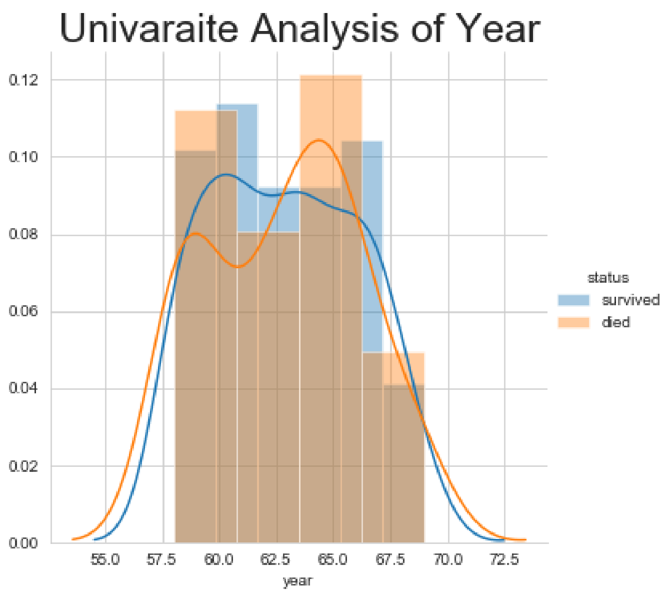


Observation::

1) From the above graph we conclude that patients in age range 30 to 34 has more chance to survive and patients in age range 75 to 80 has more chance to die.

In [11]:

```
sns.FacetGrid(haberman,hue="status",height=5).map(sns.distplot,"year").add_legend()
plt.title("Univaraite Analysis of Year",size=25)
plt.show()
```

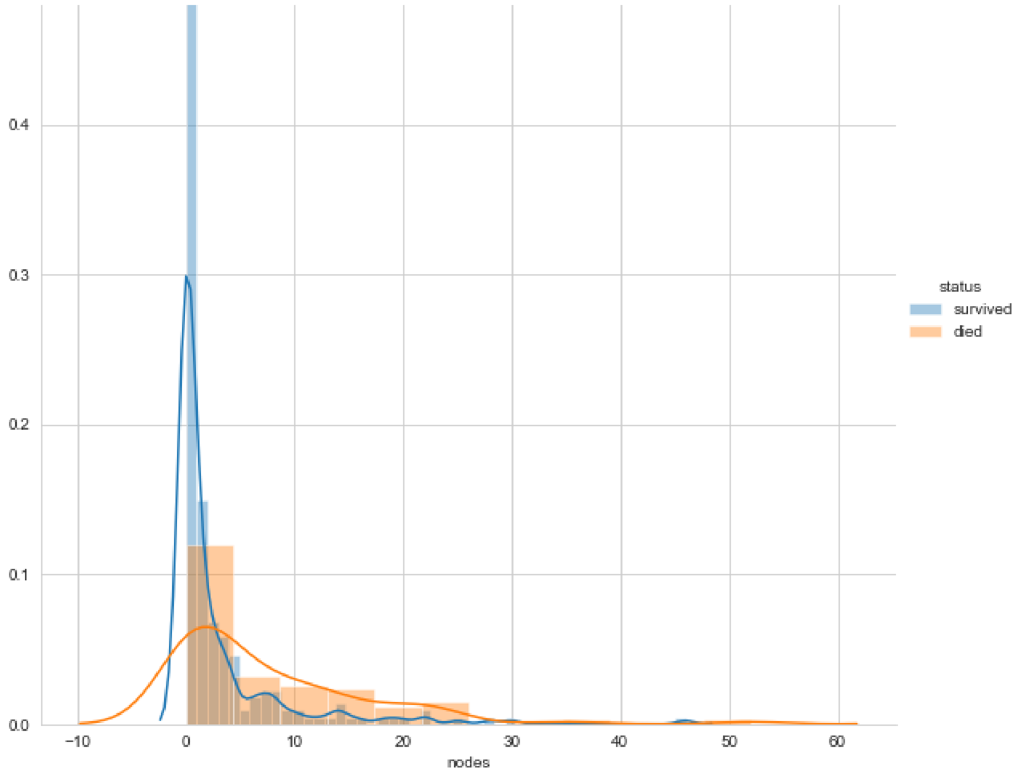


Observation::

1) In the above graph all the area is overlapping and hence we can not make any conclusion out of it.

In [12]:

```
sns.FacetGrid(haberman,hue="status",height=8).map(sns.distplot,"nodes").add_legend()
plt.title("Univaraite Analysis of Nodes",size=25)
plt.show()
```



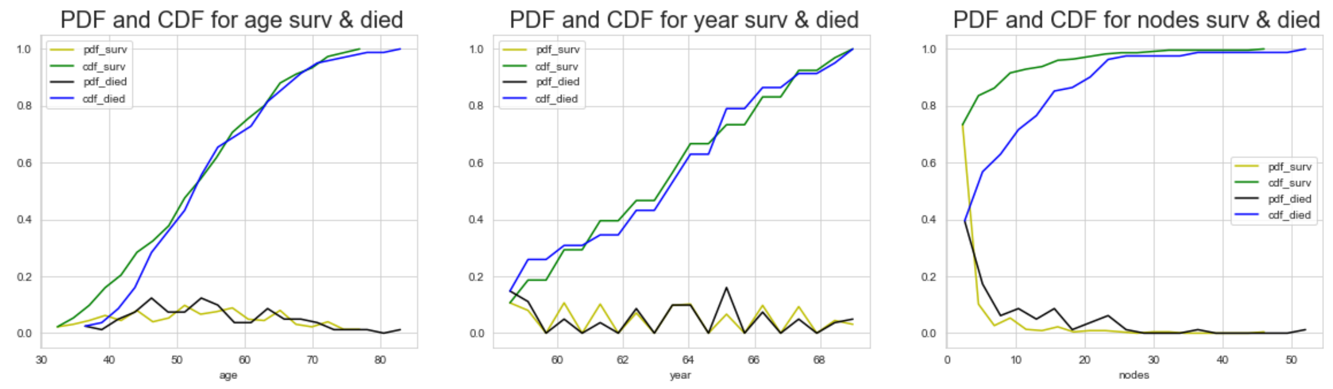
Observation::

1)The patients having nodes in between 0 to 2 has a max chance of survival.
2)Maximum nodes lie between range 0 to 25

PDF and CDF Calculation

In [13]:

```
hbm_survived = haberman.loc[haberman["status"]=="survived"]
hbm_died = haberman.loc[haberman["status"]=="died"]
#print(hbm_survived)
#print(hbm_died)

plt.figure(figsize=(20,5))
i=1
for x in (list(haberman.columns)[:-1]):
    plt.subplot(1,3,i)
    counts ,bin_edges = np.histogram(hbm_survived[x],bins=20, density= True)
    pdf=counts/(sum(counts))
    #print(pdf)
    #print(bin_edges)

    cdf=np.cumsum(pdf)
    plt.plot(bin_edges[1:],pdf,color="y",label="pdf_surv")
    plt.plot(bin_edges[1:],cdf,color='g',label="cdf_surv")

    plt.xlabel(x)
    plt.legend()


    counts ,bin_edges = np.histogram(hbm_died[x],bins=20, density= True)
    pdf=counts/(sum(counts))
    #print(pdf)
    #print(bin_edges)

    cdf=np.cumsum(pdf)
    plt.plot(bin_edges[1:],pdf,color="k",label="pdf_died")
    plt.plot(bin_edges[1:],cdf,color="b",label="cdf_died")
    plt.title(" PDF and CDF for {} surv & died".format(x),size=20)
    plt.xlabel(x)
    plt.legend()
    i+=1
plt.show()
```

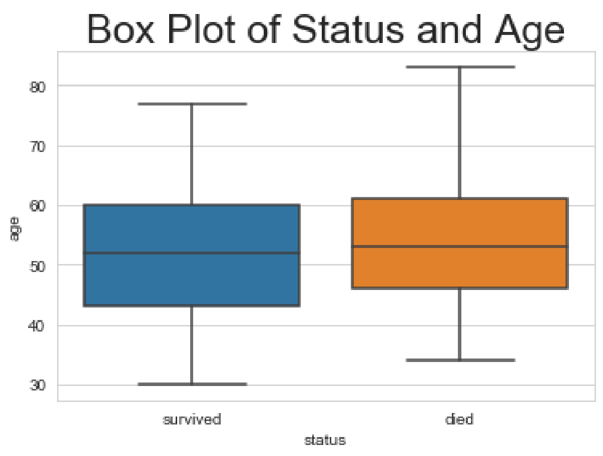


Observation::

1) From the 1st graph we can say that the patients with age in between 32 to 36 has definitly servived and the patients with age in between 77 to 85 has not servived the operation.
2) From other graphs we can not predect anything as both are overlapping.

In [14]:

```
sns.boxplot(x="status",y="age",data=haberman)
plt.title("Box Plot of Status and Age",size=25)
plt.show()
```

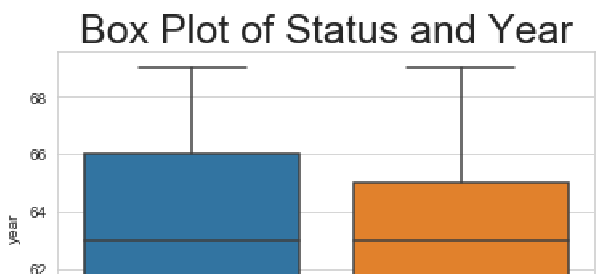


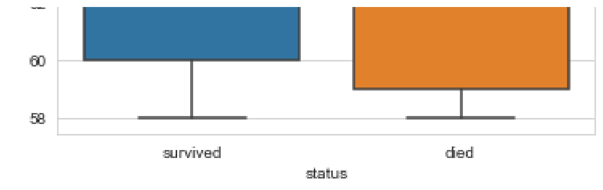
Observation:

1)25th percentile age for the survived patients is 43, 50th percentile is 52 and 75th percentile is 60.
2)25th percentile age for the ded patients is 45, 50th percentile is 53 and 75th percentile is 61.
3)Whole data is overlapping. We can not make any conclusion out of it.

In [15]:

```
sns.boxplot(x="status",y="year",data=haberman)
plt.title("Box Plot of Status and Year",size=25)
plt.show()
```



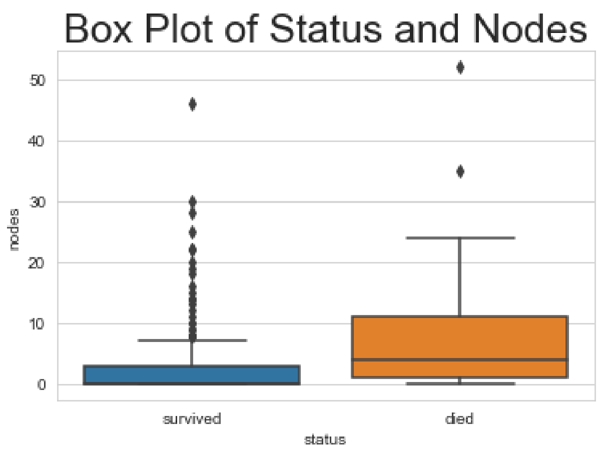


Observation:

1) More patients have servived in the year 65 to 66.
2) Deth percentage is more in the year 59 to 60.

In [16]:

```
sns.boxplot(x="status",y="nodes",data=haberman)
plt.title("Box Plot of Status and Nodes",size=25)
plt.show()
```

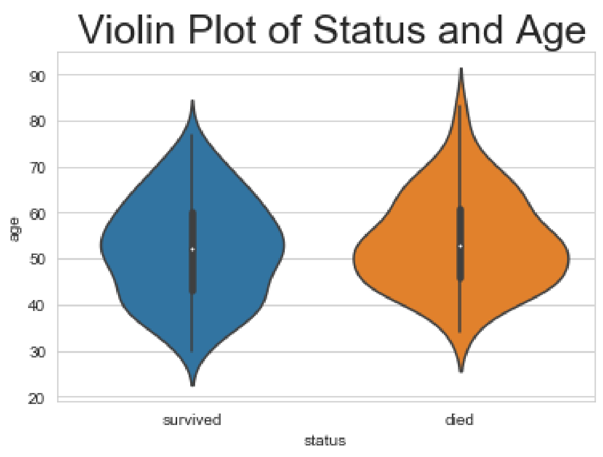


Observation:

1)Percentage of patients servival is more with nodes in a range 0 to 4.

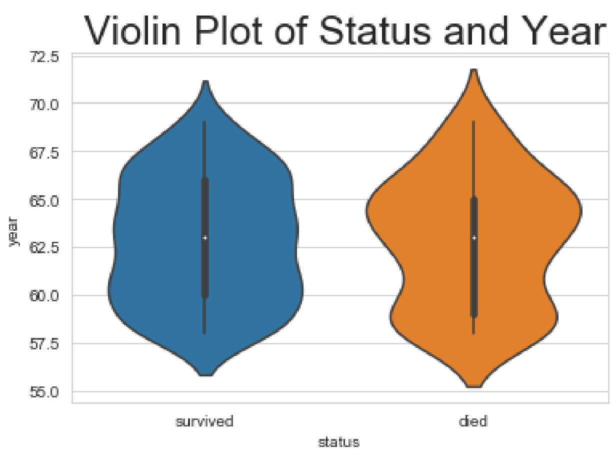
In [17]:

```
sns.violinplot(x="status",y="age",data=haberman)
plt.title(" Violin Plot of Status and Age",size=25)
plt.show()
```



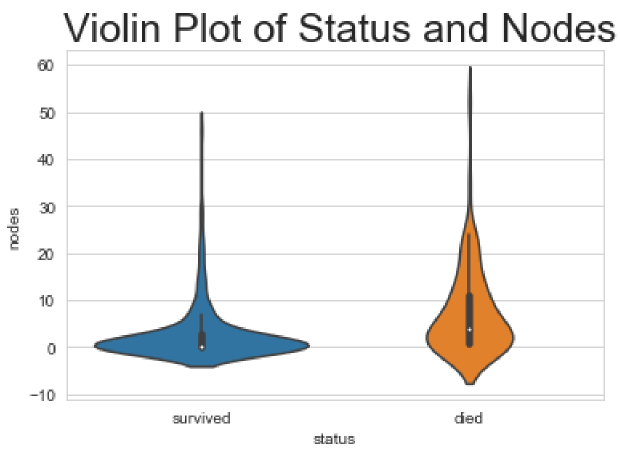
In [18]:

```
sns.violinplot(x="status",y="year",data=haberman)
plt.title(" Violin Plot of Status and Year",size=25)
plt.show()
```



In [19]:

```
sns.violinplot(x="status",y="nodes",data=haberman)
plt.title(" Violin Plot of Status and Nodes",size=25)
plt.show()
```



Observation:

1)No major conclusion could be drawn from this plots as the data points are overlapping.

In [ ]: