In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
df=pd.read_csv("haberman.csv")
```

In [3]:
```
df.shape
# Haberman Dataset has 306 Observations and 4 Columns
```

Out[3]:
```
(306, 4)
```

In [4]:
```
df.info()
df.dtypes
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306 entries, 0 to 305
Data columns (total 4 columns):
age       306 non-null int64
year      306 non-null int64
nodes     306 non-null int64
status    306 non-null int64
dtypes: int64(4)
memory usage: 9.6 KB
```

Out[4]:
```
age       int64
year      int64
nodes     int64
status    int64
dtype: object
```

In [11]:
```
#we can use above statement to store all the coloumns in a list format
col=df.columns.tolist()
```

```
df.columns
```

Out[11]:
```
Index(['age', 'year', 'nodes', 'status'], dtype='object')
```

In [12]:
```
#Here when we see that status 73% BELONGS to 1 and 26% belongs to 2, so
 this implies it is not a balanced dataset

df['status'].value_counts()
```

Out[12]:
```
1    225
2     81
Name: status, dtype: int64
```

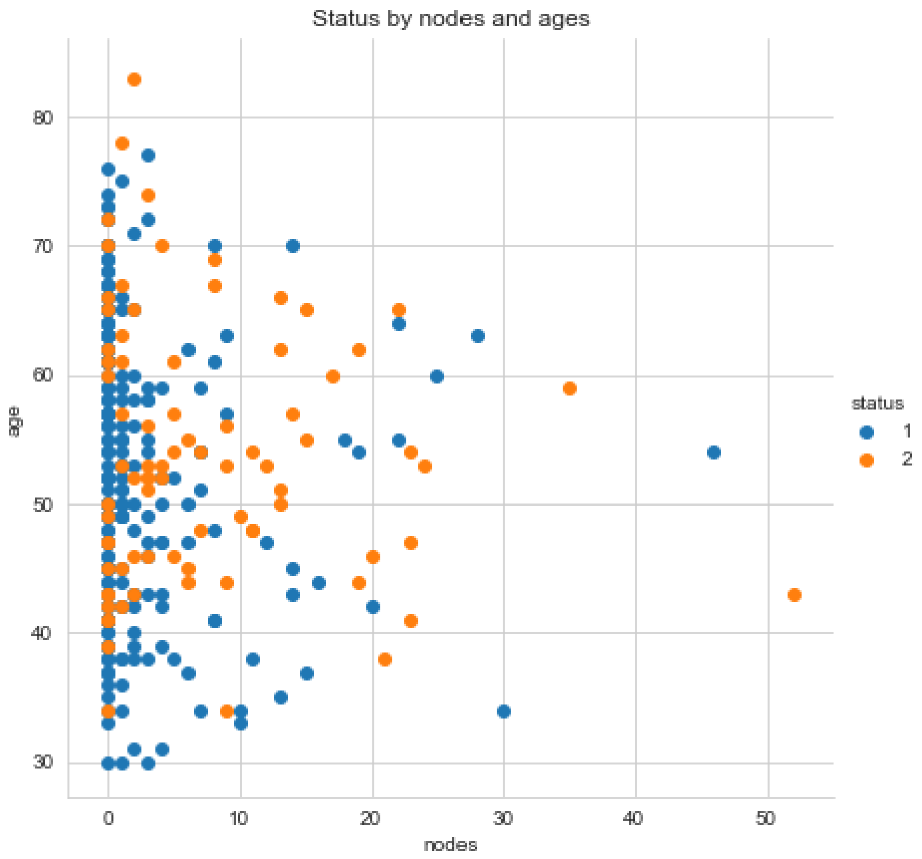
In [13]:
```
#We dont have any missing values in the records
#we also see from the above statistical table  mean and median(50%) are
 almost equailavent to each other except for nodes
df.describe()
```

Out[13]:

	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 [6]:
```
# 2d scatter plot
#2d Scatter plot with respect to ages and nodes.
sns.set_style("whitegrid");
sns.FacetGrid(df,hue="status",height=6)\
   .map(plt.scatter,"nodes","age")\
```

```
    .add_legend()
plt.title('Status by nodes and ages')

plt.show()
#Observations
#1,we have used scatter plot belwo using seaborn but it is really confu
sing to understand
 #  as all the data scattered randomly'
#2,But from my obervations i see that most of the dots  are visible at
 0 nodes and i also see that most of them belong tp 'status'=1'''
```



In [15]:
```
# from this below line code we see that 38 % of node belonging to 0 and
 10 % of nodes belonging  are in 1(Status) and all the other points are
```

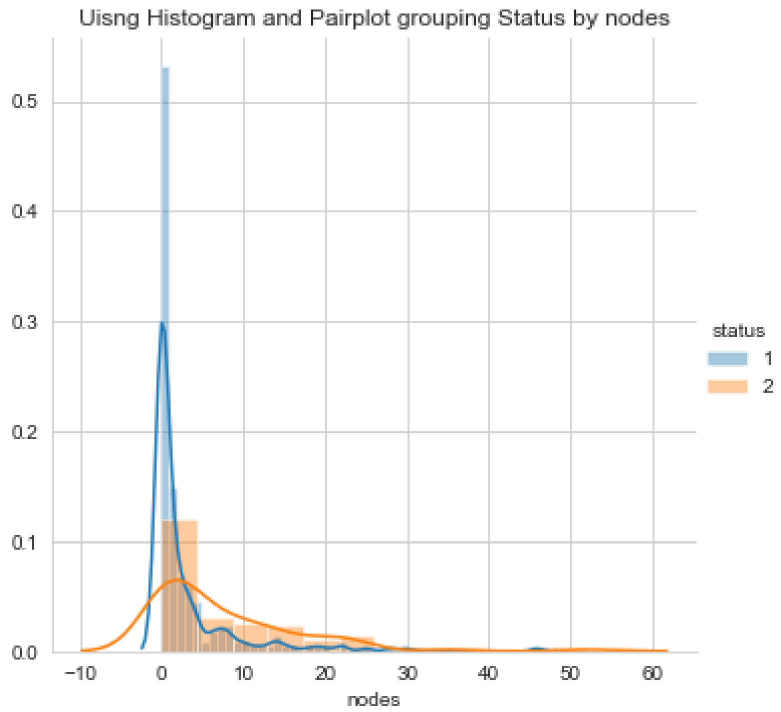
```
 scattered here and there

dfagg2=pd.crosstab(df['status'], df['nodes'],normalize=True)
# when we do the same thing for ages we see that all the points are ran
domly distibuted here and there
dfagg3=pd.crosstab(df['status'], df['age'],normalize=True)
```

In [10]:

```
# Pair Plots

sns.set_style("whitegrid");
sns.pairplot(df,hue='status',vars=['age','year','nodes'])
plt.show()

#Observations
# 1,Accoring to me i am not able to make out any obervations using pair
plots
#2, I can see only one thing in the above observations that some of the
m whose age is above 65 and belonging to year 1968 are in status 1 with
 linearly related
```

In [11]:
```
# Histogram and Pairplot for nodes


sns.FacetGrid(df,hue='status',height=5)\
    .map(sns.distplot,"nodes")\
```

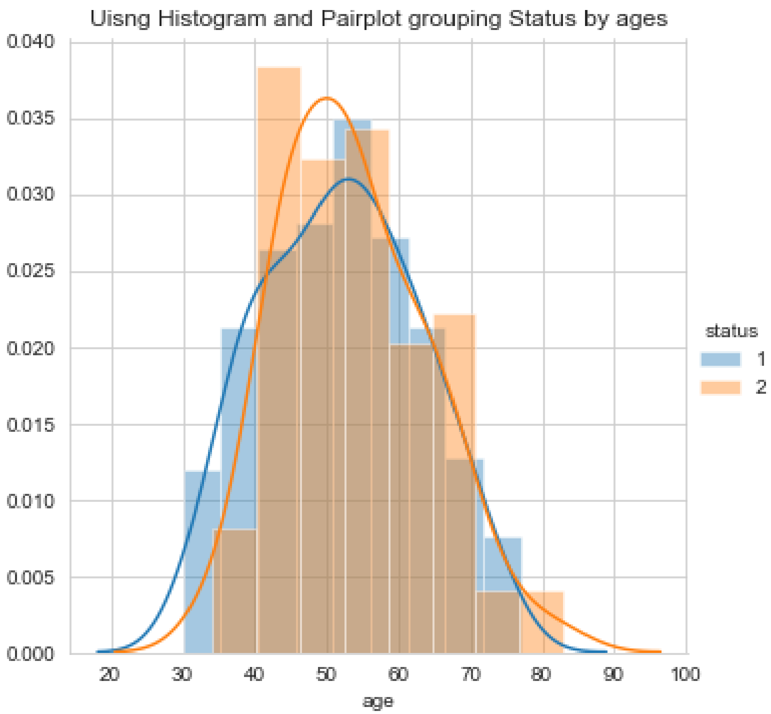
```
   .add_legend();
plt.title('Uisng Histogram and Pairplot grouping Status by nodes')

plt.show()

#Observations
#1,By using pairplot chart i can make out one thing that nodes belongin
g to -10 to -0.2 are of surival status 2
#2,we can also see that we have clear representation of axil nodes belo
nging from range 48 t0 62 are of status 2
#3,Both the status 1 and 2 corresponding to nodes are overlapping each
 other from -0.25
```



In [12]:

```
#Histogram and Pairplot for ages

sns.FacetGrid(df,hue='status',height=5)\
   .map(sns.distplot,"age")\
```

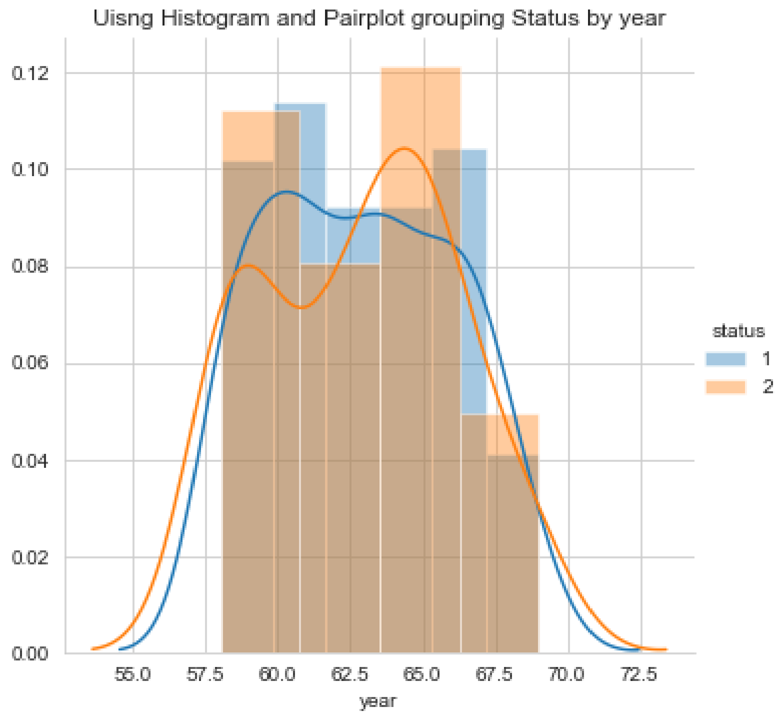
```
    .add_legend();
plt.title('Uisng Histogram and Pairplot grouping Status by ages')
plt.show()

#Observations
#1,we came across 1  obervation in this pair plot is who age is below 1
9 are in status of 1 and whose age is greater the 88 are in status 2, i
 cannot find any differentiate
```



In [13]:

```
#Histogram and Pairplot for year


sns.FacetGrid(df,hue='status',height=5)\
    .map(sns.distplot,"year")\
    .add_legend();
plt.title('Uisng Histogram and Pairplot grouping Status by year')
```

```
plt.show()

# Observations
#1,we can see that the year in range from 58 to 69 have both status 1 a
nd 2
#2,we can see that there are less % who are in both status 1 and status
  after year 72 and in year 55
```

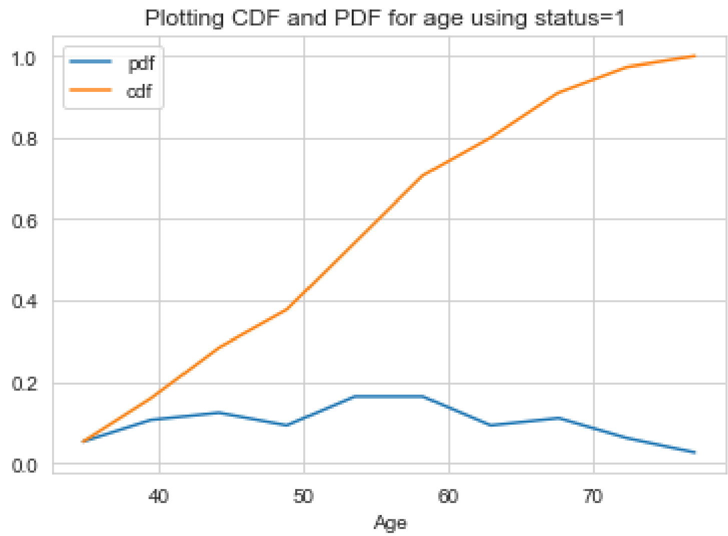


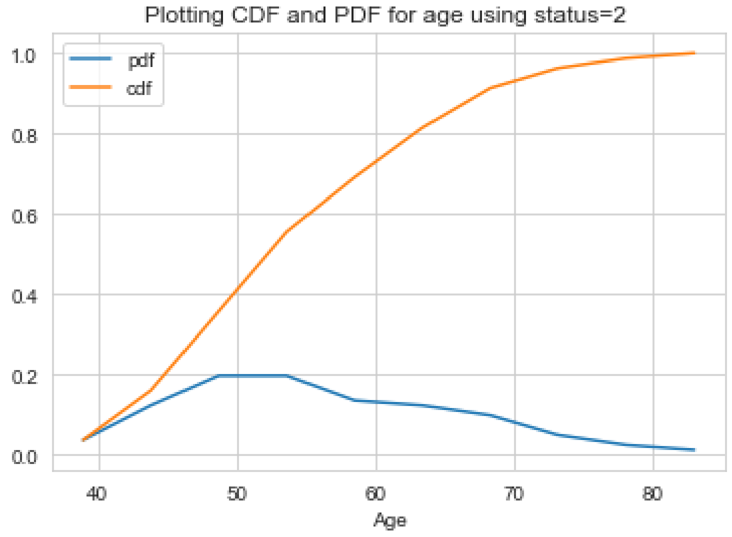
In [16]:
```
#CDF and PDF for age
import numpy as np
dfstatus1=df.loc[df["status"]==1]
ddfstatus2=df.loc[df['status']==2]
```

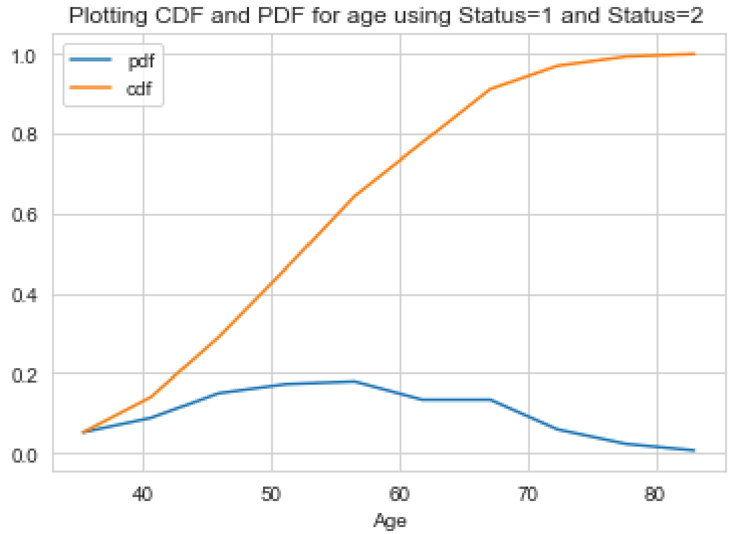
In [20]:
```
#Computing CDF and PDF in comparision to age.

#Plotting cdf and pdf for status=1 in comparision to age
```

```
counts,bins_edges=np.histogram(dfstatus1['age'],bins=10,density=True)
pdf=counts/(sum(counts))
pdf
bins_edges

#compute cdf for status=1
cdf=np.cumsum(pdf)
plt.plot(bins_edges[1:],pdf,label='pdf')
plt.plot(bins_edges[1:],cdf,label='cdf')
plt.legend()
plt.title('Plotting CDF and PDF for age using status=1 ')
plt.xlabel('Age')

plt.show()
#Plotting cdf and pdf for status=2 in comparision to age
counts,bins_edges=np.histogram(ddfstatus2['age'],bins=10,density=True)
pdf=counts/(sum(counts))
pdf
bins_edges

#compute cdf for status=2
cdf=np.cumsum(pdf)
plt.plot(bins_edges[1:],pdf,label='pdf')
plt.plot(bins_edges[1:],cdf,label='cdf')
plt.legend()
plt.title('Plotting CDF and PDF for age using status=2 ')
plt.xlabel('Age')

plt.show()

#Plotting cdf and pdf for both status=1 and status=2 in comparision to
 age.
counts,bins_edges=np.histogram(df['age'],bins=10,density=True)
pdf=counts/(sum(counts))
pdf
bins_edges

#compute cdf for both the status
cdf=np.cumsum(pdf)
```

```
plt.plot(bins_edges[1:],pdf,label='pdf')
plt.plot(bins_edges[1:],cdf,label='cdf')
plt.legend()
plt.title('Plotting CDF and PDF for age using Status=1 and Status=2 ')
plt.xlabel('Age')

plt.show()

#Observation
#1,70%(159) of them belonging to age group are below 58 and their statu
s is 1
#2,57%(45) of them belonging to age group are below 53 and their status
 is 2
#3,when we check for both the status 64 % of them belonging to age grou
p less then 56
#4, According to me this classification will only work when want to com
pare both status diiffferently and the results are invalid with resepec
t to ages
#   as to see for two status at once, this type of analysis will not wo
rk and will give us irrelvant anaswers
```



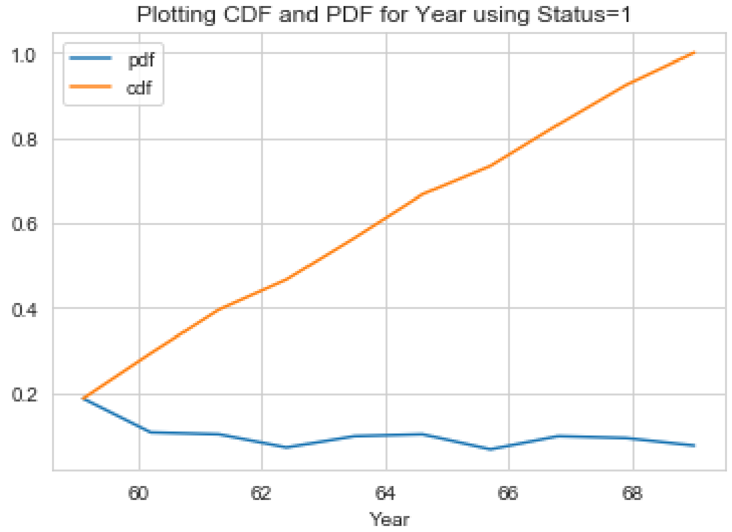

Plotting CDF and PDF for age using status=2
pdf
cdf
1.0
0.8
0.6
0.4
0.2
0.0
40
50
60
70
80
Age

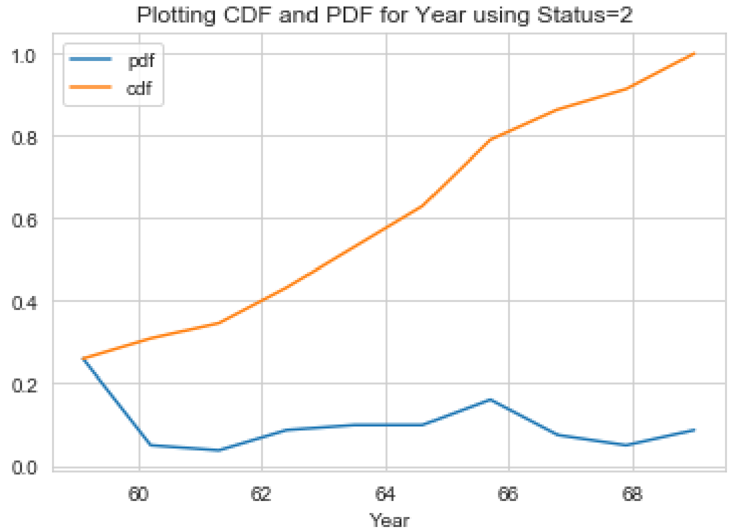

Plotting CDF and PDF for age using Status=1 and Status=2
pdf
cdf
1.0
0.8
0.6
0.4
0.2
0.0
40
50
60
70
80
Age

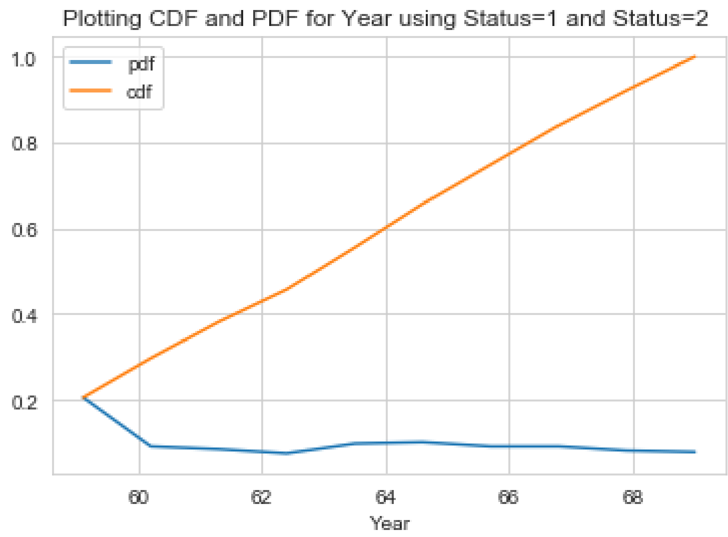
In [21]:

```
#Computing CDF and pdf for Year

#Computing pdf for status=1 with respect to Year

counts,bins_edges=np.histogram(dfstatus1['year'],bins=10,density=True)
pdf=counts/(sum(counts))
pdf
bins_edges

#compute cdf for status=1
cdf=np.cumsum(pdf)
plt.plot(bins_edges[1:],pdf,label='pdf')
plt.plot(bins_edges[1:],cdf,label='cdf')
plt.legend()
plt.title('Plotting CDF and PDF for Year using Status=1 ')
plt.xlabel('Year')
plt.show()

#Plotting cdf and pdf for status=2 in comparision to year
counts,bins_edges=np.histogram(ddfstatus2['year'],bins=10,density=True)
pdf=counts/(sum(counts))
pdf
bins_edges
```

```
#compute cdf for status=2
cdf=np.cumsum(pdf)
plt.plot(bins_edges[1:],pdf,label='pdf')
plt.plot(bins_edges[1:],cdf,label='cdf')
plt.legend()
plt.title('Plotting CDF and PDF for Year using Status=2 ')
plt.xlabel('Year')
plt.show()

#Plotting cdf and pdf for status=1 and status=2 in comparision to year.
counts,bins_edges=np.histogram(df['year'],bins=10,density=True)
pdf=counts/(sum(counts))
pdf
bins_edges

#compute cdf for both the status
cdf=np.cumsum(pdf)
plt.plot(bins_edges[1:],pdf,label='pdf')
plt.plot(bins_edges[1:],cdf,label='cdf')
plt.legend()
plt.title('Plotting CDF and PDF for Year using Status=1 and Status=2 ')
plt.xlabel('Year')
plt.show()

#Observations
#1,we cannot see a correct distribution  of cdf and pair plot in year i
n dependent to 1 (status) as in the year
#2,when we see the cdf and pdf for status=2 , i see that  39 of the val
ues are in distribution of year 60-65 with respect to status=2
#3,we see that we have totally 66%(204/306) of values belonging to stat
us =1 and status=2 who are in between year of 60 and 68
```


Plotting CDF and PDF for Year using Status=1
pdf
cdf
1.0
0.8
0.6
0.4
0.2
60
62
64
66
68
Year

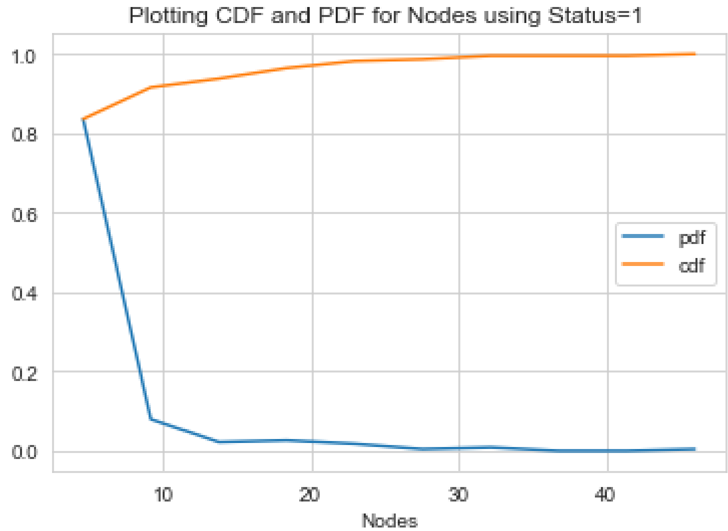

Plotting CDF and PDF for Year using Status=2
pdf
cdf
1.0
0.8
0.6
0.4
0.2
0.0
60
62
64
66
68
Year

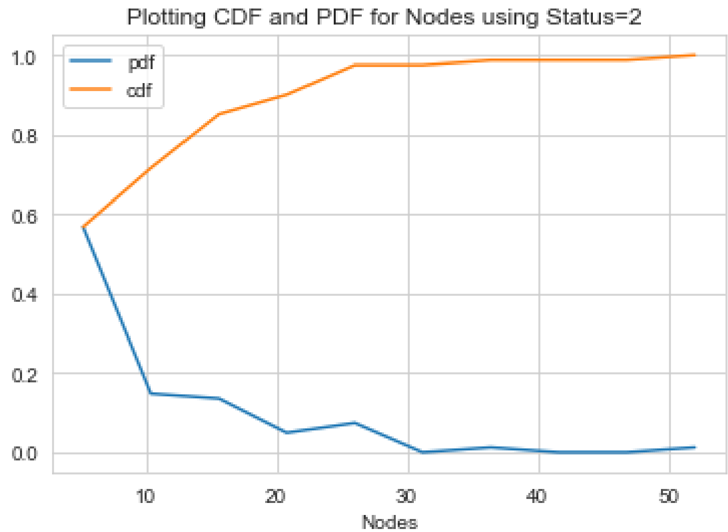


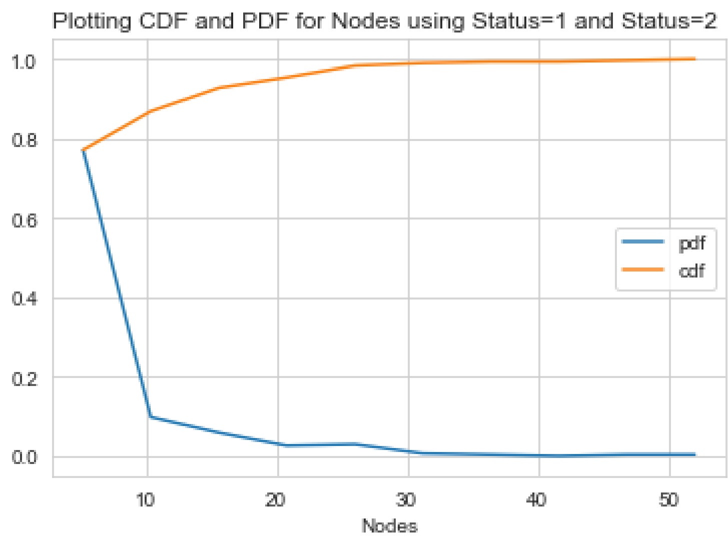
In [23]:
```
#compute cdf and pdf for nodes in relation to status

#computing cdf and pdf in respect to status=1 with respect to nodes


#compute pdf for status=1 in respect to nodes
counts,bins_edges=np.histogram(dfstatus1['nodes'],bins=10,density=True)
pdf=counts/(sum(counts))
pdf
bins_edges

#compute cdf   for status=1 in respect to nodes
cdf=np.cumsum(pdf)
plt.plot(bins_edges[1:],pdf,label='pdf')
plt.plot(bins_edges[1:],cdf,label='cdf')
plt.legend()
plt.title('Plotting CDF and PDF for Nodes using Status=1 ')
```

```
plt.xlabel('Nodes')
plt.show()

#computing cdf and pdf in respect to status=2 with respect to nodes
counts,bins_edges=np.histogram(ddfstatus2['nodes'],bins=10,density=True
)
pdf=counts/(sum(counts))
pdf
bins_edges

#compute cdf  for status=2 in respect to nodes
cdf=np.cumsum(pdf)
plt.plot(bins_edges[1:],pdf,label='pdf')
plt.plot(bins_edges[1:],cdf,label='cdf')
plt.legend()
plt.title('Plotting CDF and PDF for Nodes using Status=2 ')
plt.xlabel('Nodes')
plt.show()

#computing cdf and pdf in respective to both status  with respect to no
des
#compute pdf for nodes with respect to status=1 and status=2
counts,bins_edges=np.histogram(df['nodes'],bins=10,density=True)
pdf=counts/(sum(counts))
pdf
bins_edges

#compute cdf for nodes with respect to status=1 and status=2
cdf=np.cumsum(pdf)
plt.plot(bins_edges[1:],pdf,label='pdf')
plt.plot(bins_edges[1:],cdf,label='cdf')
plt.legend()
plt.title('Plotting CDF and PDF for Nodes using Status=1 and Status=2 '
)
plt.xlabel('Nodes')
plt.show()

#Observations
```

```
#1,I cannot see any thing in nodes comparision to status=1 and stautus=
2 and both the status in respective to nodes
```







In [32]:
```
#Calculating the mean and std of the haberman dataset
print('The mean of the nodes of haberman dataset which has status 1 and
 status 2 are given below for nodes')
print(np.mean(dfstatus1['nodes']))        #calculating the mean of the
 nodes with status 1
print(np.mean(ddfstatus2['nodes']))        #calculating the mean of th
e nodes with status 2
print('The standard deviation of the nodes of haberman dataset which ha
s status 1 and status 2 are given below for nodes')
print(np.std(dfstatus1['nodes']))          #calculating the standard de
viation with status 1
print(np.std(ddfstatus2['nodes']))

#Calculating the mean and std of the haberman dataset with respect to a
ges
print('The mean of the nodes of haberman dataset which has status 1 and
 status 2 are given below for ages')
print(np.mean(dfstatus1['age']))         #calculating the mean of the n
odes with status 1
print(np.mean(ddfstatus2['age']))         #calculating the mean of the
 nodes with status 2
```

```
print('The standard deviation of the nodes of haberman dataset which ha
s status 1 and status 2 are given below for ages')
print(np.std(dfstatus1['age']))          #calculating the standard devi
ation with status 1
print(np.std(ddfstatus2['age']))         #calculating the standard devi
ation with status 2

#Calculating the mean and std of the haberman dataset with respect to y
ear
print('The mean of the nodes of haberman dataset which has status 1 and
 status 2 are given below for year')
print(np.mean(dfstatus1['year']))         #calculating the mean of the
 nodes with status 1
print(np.mean(ddfstatus2['year']))         #calculating the mean of the
 nodes with status 2
print('The standard deviation of the nodes of haberman dataset which ha
s status 1 and status 2 are given below for year')
print(np.std(dfstatus1['year']))          #calculating the standard dev
iation with status 1
print(np.std(ddfstatus2['year']))


#conclusion:By using visual charts we are unable to find any kind of pa
tterns with respect to status 1 and status 2
#we cannot use any of the columns  and say correctly that buy taking th
is column we can clearly see the difference in status
#But using mean and std we can see that only for nodes column the mean
 and std proves us worthy with repspect to status 1 and 2
'''The mean of the nodes of haberman dataset which has status 1 and sta
tus 2 are given below
2.7911111111111113
7.45679012345679
The standard deviation of the nodes of haberman dataset which has statu
s 1 and status 2 are given below
5.857258449412131
9.128776076761632'''
```

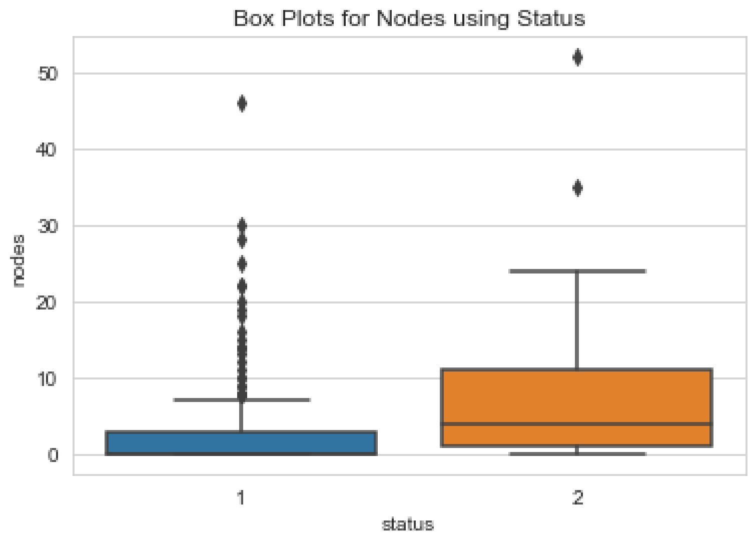
```
The mean of the nodes of haberman dataset which has status 1 and status
2 are given below for nodes
2.7911111111111113
```

```
7.45679012345679
The standard deviation of the nodes of haberman dataset which has statu
s 1 and status 2 are given below for nodes
5.857258449412131
9.128776076761632
The mean of the nodes of haberman dataset which has status 1 and status
2 are given below for ages
52.01777777777778
53.67901234567901
The standard deviation of the nodes of haberman dataset which has statu

s 1 and status 2 are given below for ages
10.98765547510051
10.10418219303131
The mean of the nodes of haberman dataset which has status 1 and status
2 are given below for year
62.86222222222222
62.82716049382716
The standard deviation of the nodes of haberman dataset which has statu
s 1 and status 2 are given below for year
3.2157452144021956
3.3214236255207883
```

Out[32]:
```
'The mean of the nodes of haberman dataset which has status 1 and statu
s 2 are given below\n2.7911111111111113\n7.45679012345679\nThe standard
deviation of the nodes of haberman dataset which has status 1 and statu
s 2 are given below\n5.857258449412131\n9.128776076761632'
```

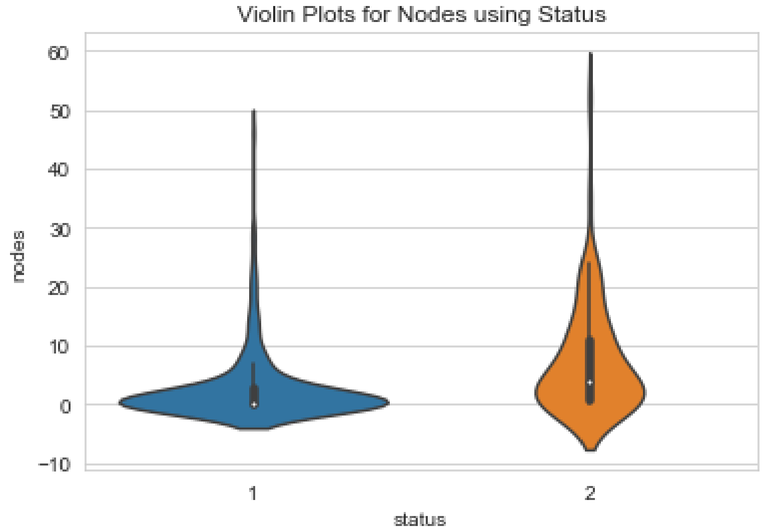
In [24]:
```
# Lets only use nodes and take a look at boxplot if we are able to find
 anything as we are able to see a lot of
#variation of mean and std for nodes for both the status

#I see from below box plot that the distribution of data is not equally
 spread btw 25-100 for both the sttaus which can be clearly represented
 for both the nodes
# Box Plot
sns.boxplot(x='status',y='nodes',data=df)
plt.title("Box Plots for Nodes using Status")
```

Out[24]: Text(0.5, 1.0, 'Box Plots for Nodes using Status')



In [25]:
```
#violin plot for nodes in comparison to status
sns.violinplot(x='status',y='nodes',data=df)
plt.title("Violin Plots for Nodes using Status")

#Observations:
#1,I see no observations from below point.
```

Out[25]: Text(0.5, 1.0, 'Violin Plots for Nodes using Status')



#Conclusion #I think after doing total visual analysis of all the obervations we can say that we can use nodes column for significant observations in status as we see that we got a mean and std for both the status of nodes with a very large variance