(3.12) Exercise:

1) Download Haberman Cancer Survival dataset from Kaggle. You may have to create a Kaggle account to download data. (https://www.kaggle.com/gilsousa/habermans-survival-data-set)

2) Perform an analysis on this dataset with the following sections:

3) High level statistics of the dataset: number of points, numer of features, number of classes, data-points per class.

4) Explain our objective.

5) Perform Univaraite analysis(PDF, CDF, Boxplot, Voilin plots) to understand which features are useful towards classification.

6) Perform Bi-variate analysis (scatter plots, pair-plots) to see if combinations of features are useful in classfication.

7) Write your observations in english as crisply and unambigously as possible. Always quantify your results.

Description about the Dataset:

```
    This dataset contains the data about patients who under wen
t Breast cancer treatment (surgery) from the year 1958 to 1970 a
t the University of Chicago's Billings Hospital. (Reference: htt
ps://www.kaggle.com/gilsousa/habermans-survival-data-set/versio
n/1)
```

Objective:

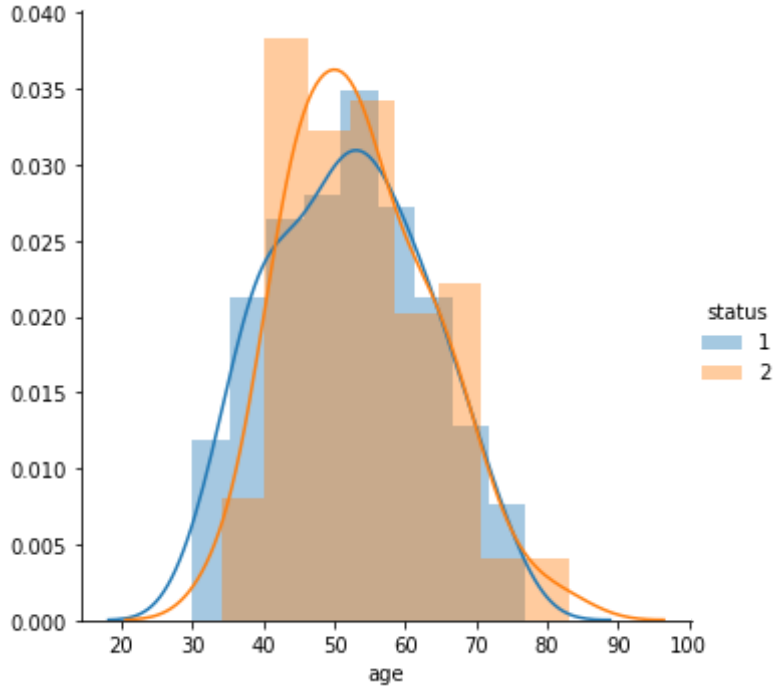
To find the the features important to identify to classify (in a new dataset) if a patient is likely to survive longer tha n 5 years after the treatment (surgery) for cancer, provided the features such as age, nodes, year are given.

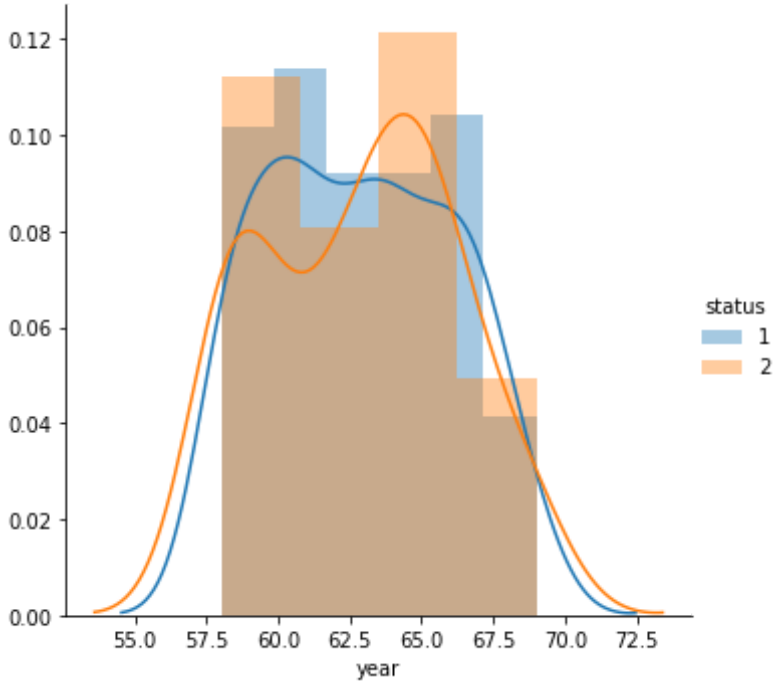
In [1]:
```
"""This program gives the high level statistics of the Haberman dataset
 like number of datapoints, features,
classes and datapoints per class."""

import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import warnings
warnings.filterwarnings("ignore")

hb = pd.read_csv("haberman.csv")
#print(hb)
a = hb.shape
print("Total number of points in the given Haberman dataset : {}".forma
t(a[0]))
print("Total number of Features in the given Haberman dataset : {}".for
mat(a[1]))
b = hb["status"].value_counts()
c = b.shape
print("Total number of Classes in the given Haberman dataset : {}".form
at(c[0]))
print("Total number of values in the each class : \n{}".format(b))
```

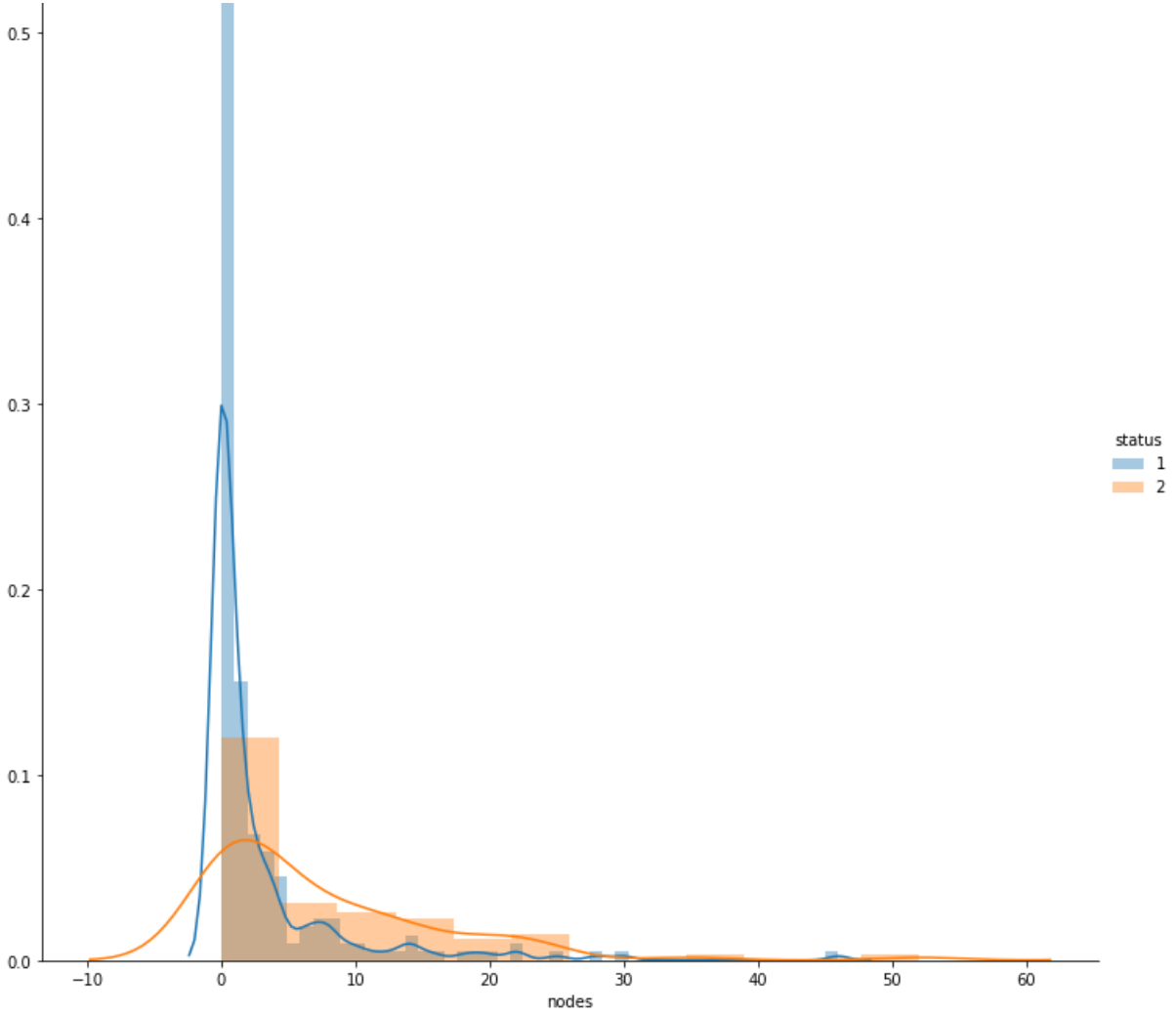
```
Total number of points in the given Haberman dataset : 306
Total number of Features in the given Haberman dataset : 4
Total number of Classes in the given Haberman dataset : 2
Total number of values in the each class :
1    225
2     81
Name: status, dtype: int64
```

Univariate analysis

In [2]:
```
"""PDF - Probablity Density Function of the Haberman dataset."""

sns.FacetGrid(hb, hue="status", height=5) \
   .map(sns.distplot, "age") \
   .add_legend();
plt.show();
sns.FacetGrid(hb, hue="status", height=5) \
   .map(sns.distplot, "year") \
   .add_legend();
plt.show();
sns.FacetGrid(hb, hue="status", height=10) \
   .map(sns.distplot, "nodes") \
   .add_legend();
plt.show();
```







Observations:

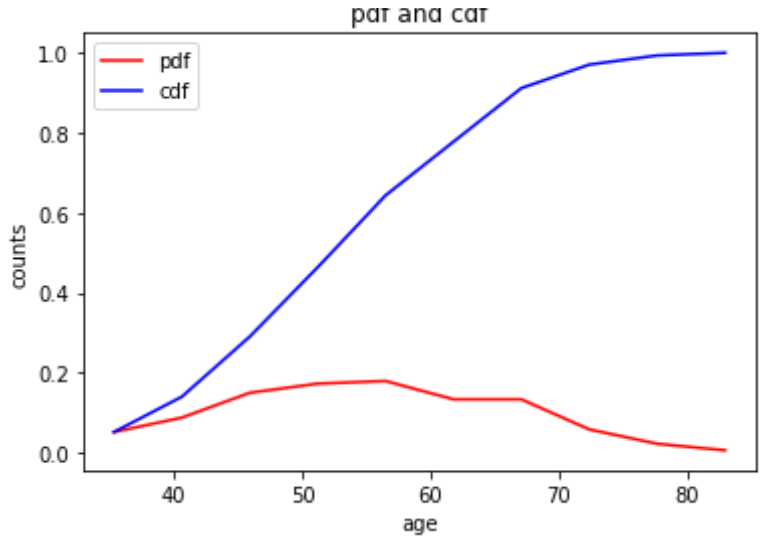
A Probability Density Function for features like age, year, nodes of the Haberman Dataset shows that not much can be interpreted in terms of distinguishing between the status 1 and 2 (i.e., will a patient survive longer than 5 years after a cancer treatment or not) based on age, year of treatment and number of nodes si

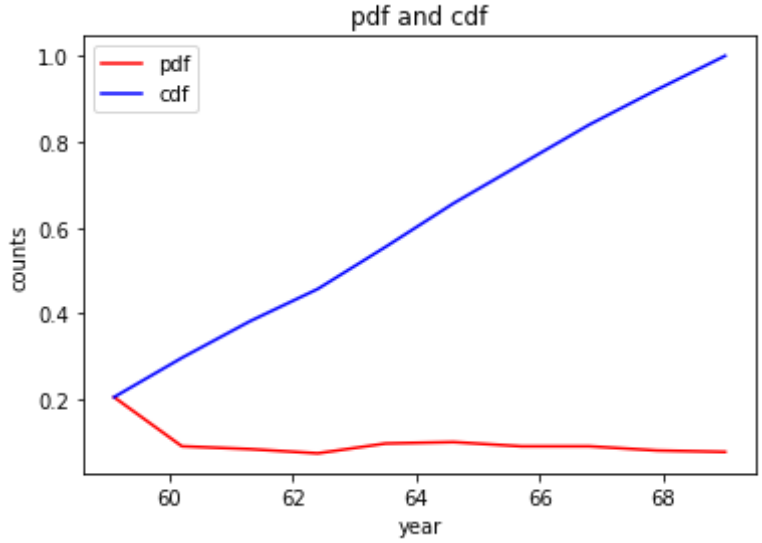
```
nce almost in all 3 plots the PDF curve overlaps with one anothe
r.
But Patients with nodes less than 3 have relatively higher proba
bility (54% - Approx) of surviving for more than 5 years.
```

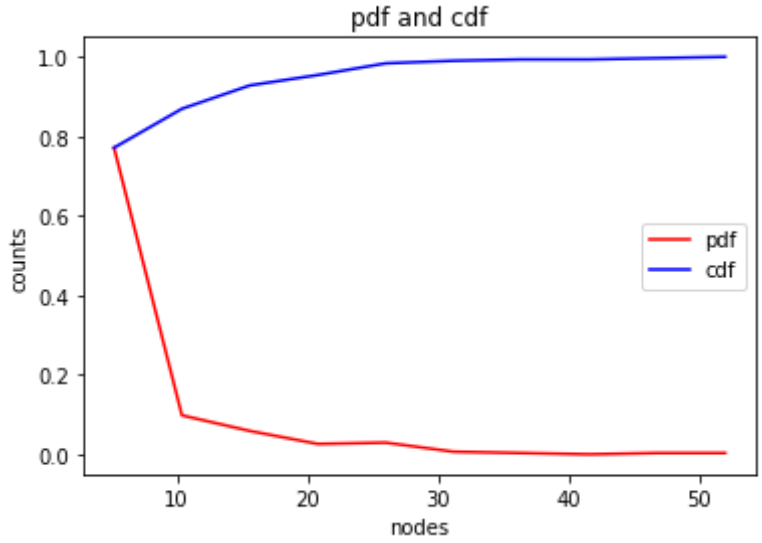
In [3]:

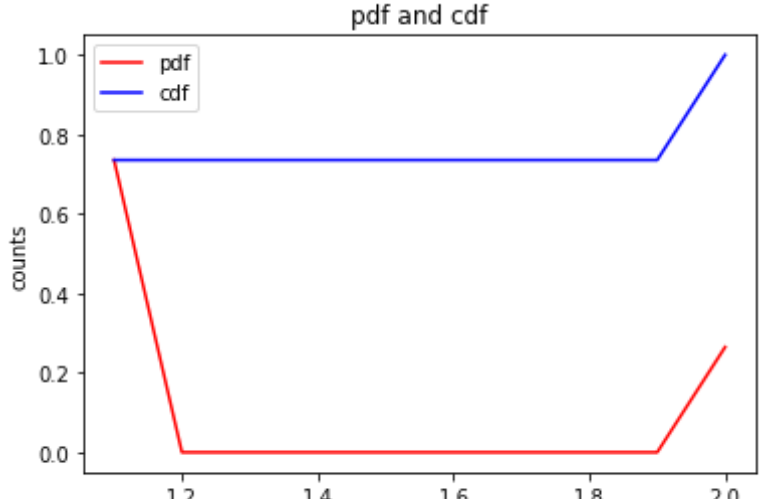
```
"""Cumulative Distribution Function (CDF) and PDF of the Haberman datas
et."""

counts, bin_edges = np.histogram(hb['age'], bins=10, density = True)
# print("counts (when density is true) = ", counts)
# print("bin_edges = ", bin_edges);
pdf = counts/(sum(counts))
# print("pdf = ", pdf);
cdf = np.cumsum(pdf)
# print('cdf = ', cdf)
plt.plot(bin_edges[1:],pdf, 'r-', label='pdf')
plt.plot(bin_edges[1:], cdf, 'b-', label='cdf')
plt.ylabel("counts")
plt.xlabel("age")
plt.title('pdf and cdf')
plt.legend()
plt.show()

counts, bin_edges = np.histogram(hb['year'], bins=10, density = True)
# print("counts (when density is true) = ", counts)
# print("bin_edges = ", bin_edges);
pdf = counts/(sum(counts))
# print("pdf = ", pdf);
cdf = np.cumsum(pdf)
# print('cdf = ', cdf)
plt.plot(bin_edges[1:],pdf, 'r-', label='pdf')
plt.plot(bin_edges[1:], cdf, 'b-', label='cdf')
plt.ylabel("counts")
plt.xlabel("year")
plt.title('pdf and cdf')
plt.legend()
plt.show()
```

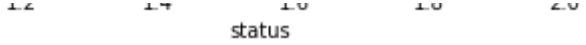
```
counts, bin_edges = np.histogram(hb['nodes'], bins=10, density = True)
# print("counts (when density is true) = ", counts)
# print("bin_edges = ", bin_edges);
pdf = counts/(sum(counts))
# print("pdf = ", pdf);
cdf = np.cumsum(pdf)
# print('cdf = ', cdf)
plt.plot(bin_edges[1:],pdf, 'r-', label='pdf')
plt.plot(bin_edges[1:], cdf, 'b-', label='cdf')
plt.ylabel("counts")
plt.xlabel("nodes")
plt.title('pdf and cdf')
plt.legend()
plt.show()

counts, bin_edges = np.histogram(hb['status'], bins=10, density = True)
# print("counts (when density is true) = ", counts)
# print("bin_edges = ", bin_edges);
pdf = counts/(sum(counts))
# print("pdf = ", pdf);
cdf = np.cumsum(pdf)
# print('cdf = ', cdf)
plt.plot(bin_edges[1:],pdf, 'r-', label='pdf')
plt.plot(bin_edges[1:], cdf, 'b-', label='cdf')
plt.ylabel("counts")
plt.xlabel("status")
plt.title('pdf and cdf')
plt.legend()
plt.show()
```









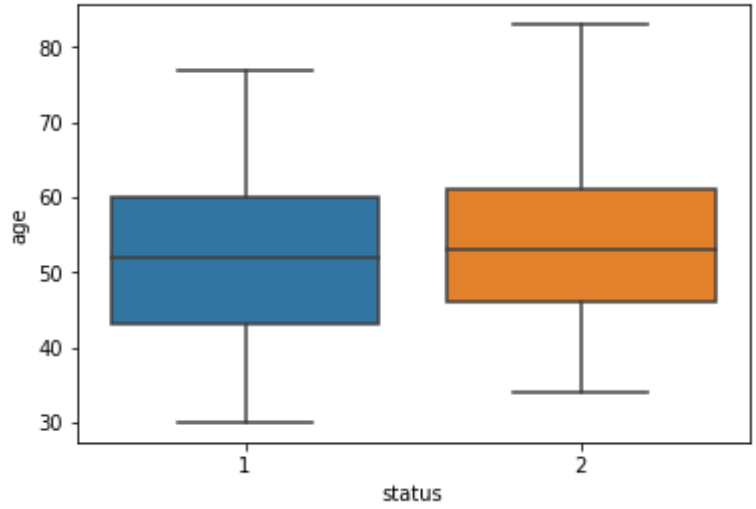


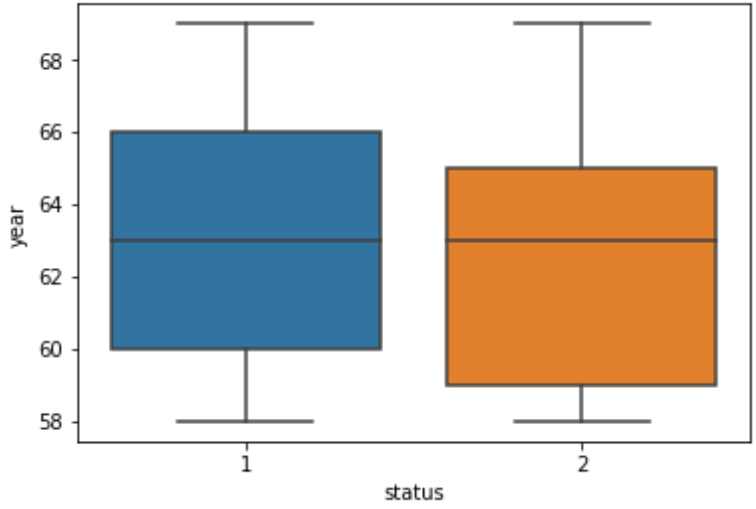
Observations:

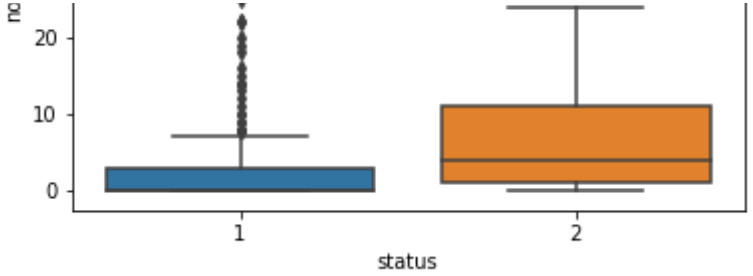
Approximate quantification: Age: 70% of patients < 60 years, 90% of patients < 70 years. Year: 20% of treatments were done before 1960 and then maintained at 10% over the years. Nodes: 78% of patients <= 5 nodes, 10% of patients <= 10 nodes.

Bi-variate analysis

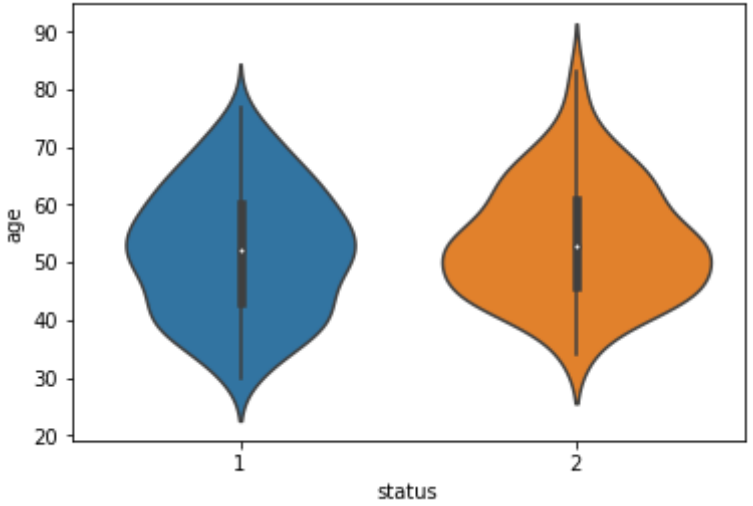
In [4]:
```
"""Box plots of the Haberman dataset."""
# age
sns.boxplot(x='status',y='age', data=hb)
plt.show()

# year
sns.boxplot(x='status',y='year', data=hb)
plt.show()

# nodes
sns.boxplot(x='status',y='nodes', data=hb)
plt.show()
```

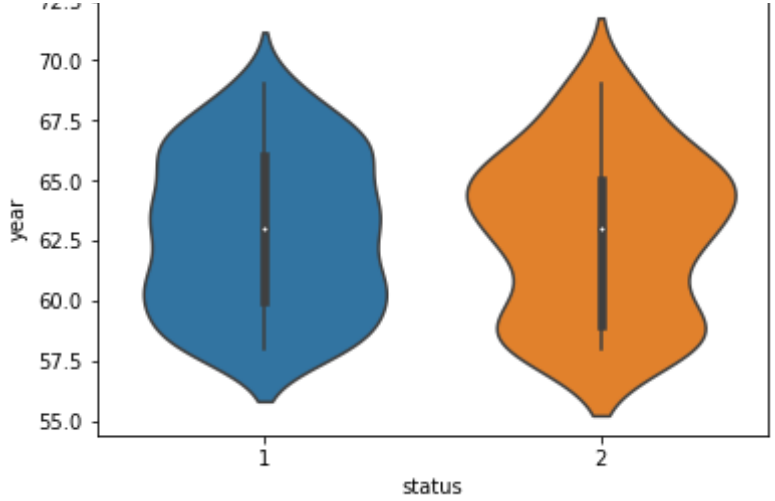


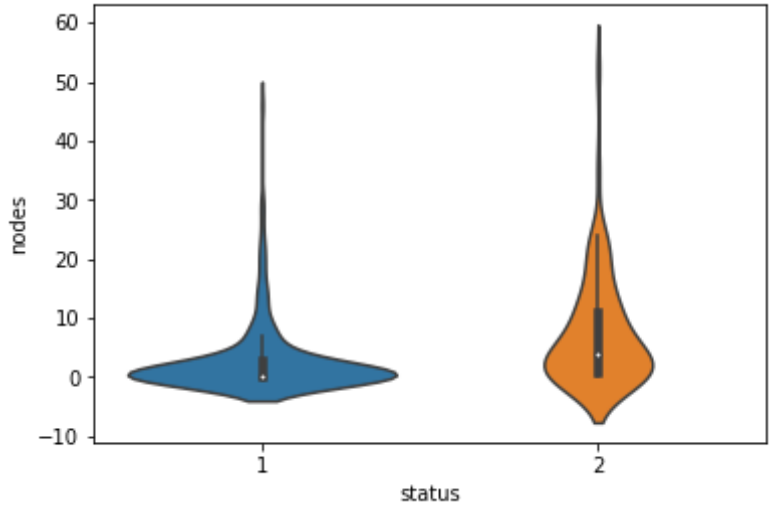




In [5]:
```
"""Violin plots of the Haberman dataset."""
# age
sns.violinplot(x='status',y='age', data=hb, size=8)
plt.show()

# year
sns.violinplot(x='status',y='year', data=hb, size=8)
plt.show()

# nodes
sns.violinplot(x='status',y='nodes', data=hb, size=8)
plt.show()
```







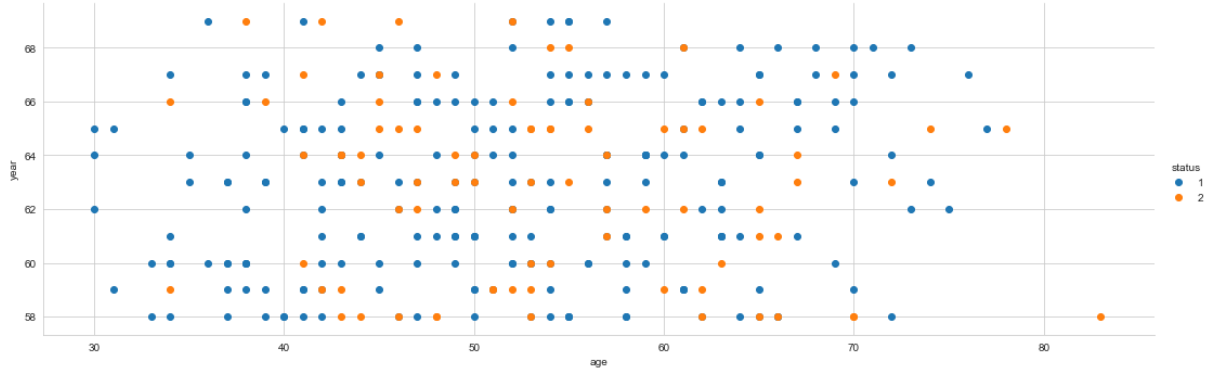
Observations from the Box plot and violin plot (Since Violin plot will have the information from box plot as well):

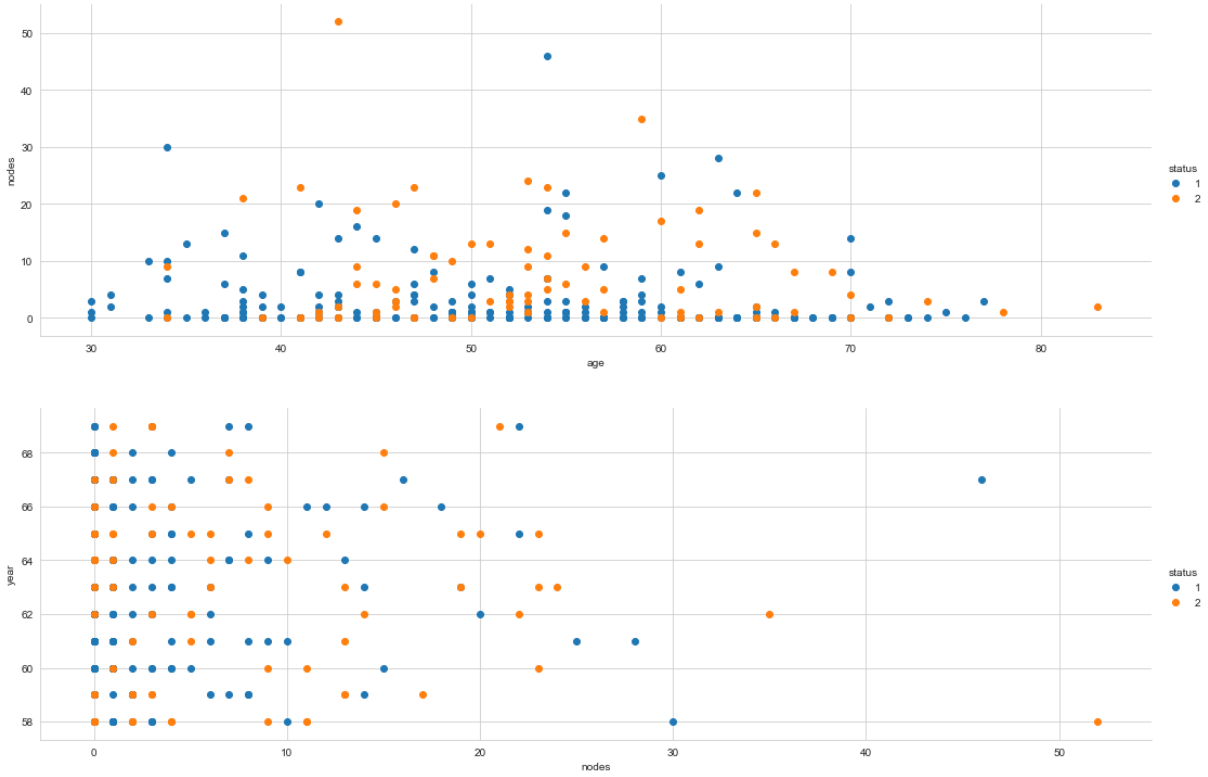
1) age vs status: Status 1 and 2 overlap totally from 25% to 75% and but the percentage people surviving longer than 5 years is more for patients who got treated with age of less than 40.

2) year vs status: Status 1 and 2 overlap mostly, but the percentage people surviving longer than 5 years is more for patients who got treated after the year 1965.

3) nodes vs status: Patients having nodes less than 3 have higher chance of surviving for longer than 5 years.

In [6]:
```
"""Scatter plot of the Haberman dataset."""
# between age and year
sns.set_style("whitegrid")
sns.FacetGrid(hb, hue="status", height=5, aspect=3).map(plt.scatter, "age", "year").add_legend()
plt.show()

# between age and nodes
sns.set_style("whitegrid")
sns.FacetGrid(hb, hue="status", height=5, aspect=3).map(plt.scatter, "age", "nodes").add_legend()
plt.show()

# between nodes and year
sns.set_style("whitegrid")
sns.FacetGrid(hb, hue="status", height=5, aspect=3).map(plt.scatter, "nodes", "year").add_legend()
plt.show()
```





Observations:

1) The Scatter plot between age and nodes gives the most information that when the patient's age is less than 40 and the number of auxilary nodes is less than 20 the survival rate is high.

2) Scatter plots between age and year, nodes and year have data points of both status overlapping a lot, revealing that the year is not an important parameter.

In [11]:
```
"""Pair plot of the Haberman dataset."""
plt.close()
sns.set_style("whitegrid")
sns.pairplot(hb, hue="status", height=4, aspect=1, vars=["age", "nodes"
, "year"]).add_legend()
plt.show()
```

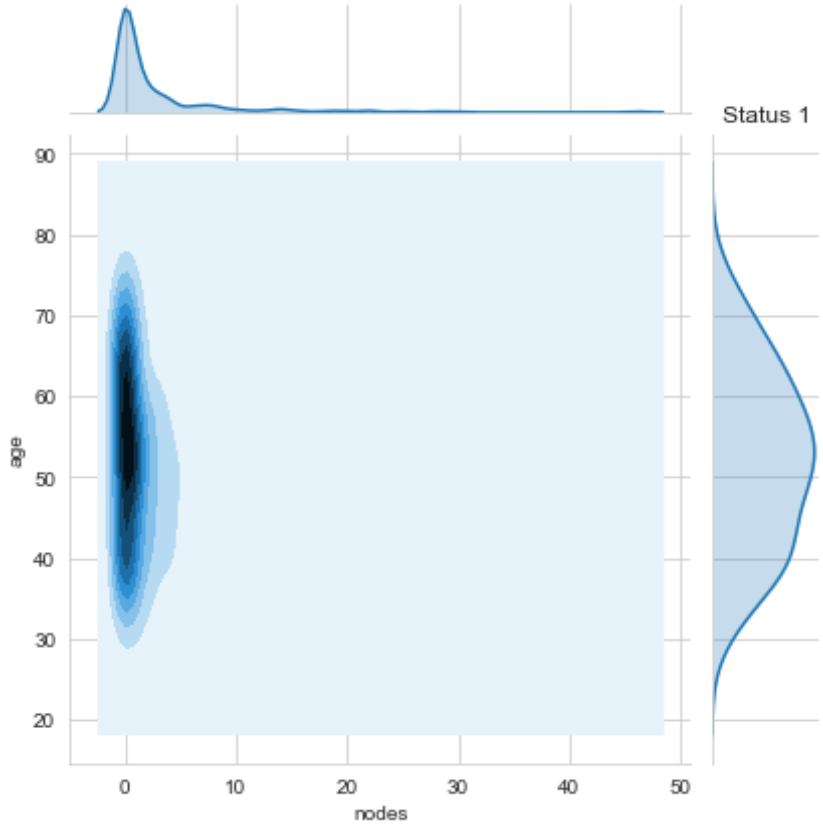
Observations:

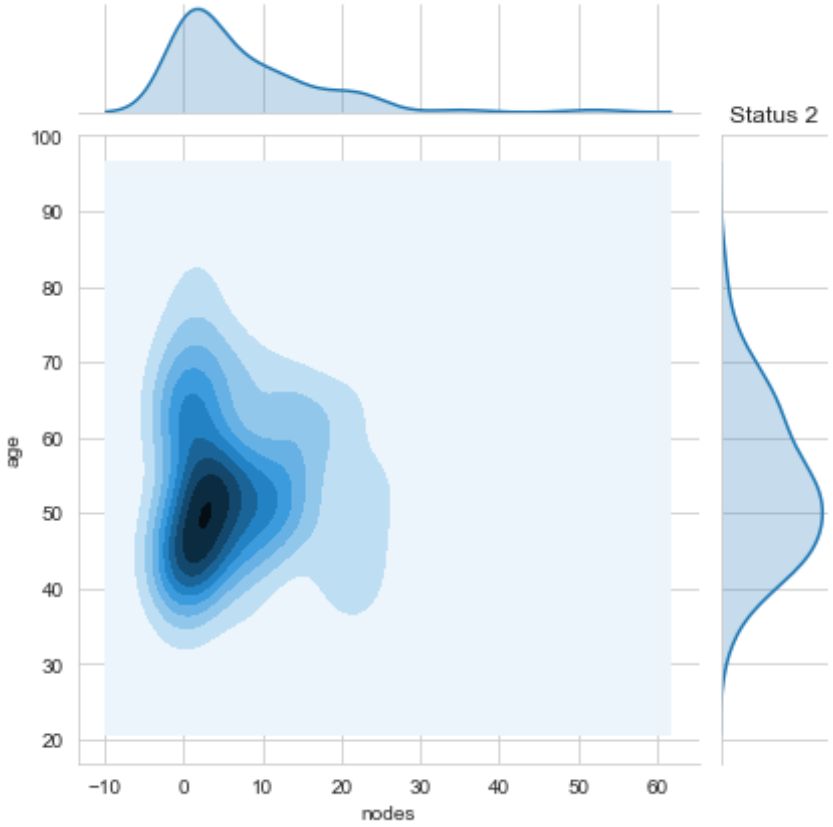
1) Like the scatter plot, the Pair plot between age and nodes gives the most information that is when the patient's age is less than 40 and the number of auxilary nodes is less than 20 the survival rate is high without many fatalities.

2) The rest of the plots have huge overlap, revealing that age and nodes are the important parameters.

Contour plot

In [8]:
```
st1 = hb.loc[hb["status"] == 1];
st2 = hb.loc[hb["status"] == 2];
#2D Density plot, contour-plot
sns.jointplot(x="nodes", y="age", data=st1, kind="kde");
plt.title("Status 1")
plt.show();
sns.jointplot(x="nodes", y="age", data=st2, kind="kde");
plt.title("Status 2")
plt.show();
```





Observations from Contour plot:

1) More Patients survive longer when they have 0 to 2 nodes and of age between 48 and 64, but for that one reason is more patients have been treated with that age group and number of nodes.

2) Percentage of patients surviving longer than 5 years when they have more than 2 nodes and of age between 48 and 52 is lower.

In [9]:
```
print("Means of status 1:")
print("age = ", np.mean(st1["age"]))
print("year = ", np.mean(st1["year"]))
print("nodes = ", np.mean(st1["nodes"]))
```

```
print("\nMeans of status 2:")
print("age = ", np.mean(st2["age"]))
print("year = ", np.mean(st2["year"]))
print("nodes = ", np.mean(st2["nodes"]))

print("\nMedians of status 1:")
print("age = ", np.median(st1["age"]))
print("year = ", np.median(st1["year"]))
print("nodes = ", np.median(st1["nodes"]))

print("\nMedians of status 2:")
print("age = ", np.median(st2["age"]))
print("year = ", np.median(st2["year"]))
print("nodes = ", np.median(st2["nodes"]))

print("\nStandard Deviation of status 1:")
print("age = ", np.std(st1["age"]))
print("year = ", np.std(st1["year"]))
print("nodes = ", np.std(st1["nodes"]))

print("\nStandard Deviation of status 2:")
print("age = ", np.std(st2["age"]))
print("year = ", np.std(st2["year"]))
print("nodes = ", np.std(st2["nodes"]))

print("\nQuantiles of status 1:")
print("age = ", np.percentile(st1["age"],np.arange(0, 100, 25)))
print("year = ", np.percentile(st1["year"],np.arange(0, 100, 25)))
print("nodes = ", np.percentile(st1["nodes"],np.arange(0, 100, 25)))

print("\nQuantiles of status 2:")
print("age = ", np.percentile(st2["age"],np.arange(0, 100, 25)))
print("year = ", np.percentile(st2["year"],np.arange(0, 100, 25)))
print("nodes = ", np.percentile(st2["nodes"],np.arange(0, 100, 25)))

print("\n90th Percentiles of status 1:")
print("age = ", np.percentile(st1["age"],90))
print("year = ", np.percentile(st1["year"],90))
```

```
print("nodes = ", np.percentile(st1["nodes"],90))
print("\n90th Percentiles of status 2:")
print("age = ", np.percentile(st2["age"],90))
print("year = ", np.percentile(st2["year"],90))
print("nodes = ", np.percentile(st2["nodes"],90))

from statsmodels import robust
print ("\nMedian Absolute Deviation of status 1")
print("age = ", robust.mad(st1["age"]))
print("year = ", robust.mad(st1["year"]))
print("nodes = ", robust.mad(st1["nodes"]))
print ("\nMedian Absolute Deviation of status 2")
print("age = ", robust.mad(st2["nodes"]))
print("year = ", robust.mad(st2["age"]))
print("nodes = ", robust.mad(st2["year"]))
```

```
Means of status 1:
age =  52.01777777777778
year =  62.86222222222222
nodes =  2.7911111111111113

Means of status 2:
age =  53.67901234567901
year =  62.82716049382716
nodes =  7.45679012345679

Medians of status 1:
age =  52.0
year =  63.0
nodes =  0.0

Medians of status 2:
age =  53.0
year =  63.0
nodes =  4.0

Standard Deviation of status 1:
age =  10.98765547510051
year =  3.2157452144021956
```

```
nodes =  5.857258449412131

Standard Deviation of status 2:
age =  10.10418219303131
year =  3.3214236255207883
nodes =  9.128776076761632

Quantiles of status 1:
age =  [30. 43. 52. 60.]
year =  [58. 60. 63. 66.]
nodes =  [0. 0. 0. 3.]

Quantiles of status 2:
age =  [34. 46. 53. 61.]
year =  [58. 59. 63. 65.]
nodes =  [ 0.  1.  4. 11.]

90th Percentiles of status 1:
age =  67.0
year =  67.0
nodes =  8.0

90th Percentiles of status 2:
age =  67.0
year =  67.0
nodes =  20.0

Median Absolute Deviation of status 1
age =  13.343419966550417
year =  4.447806655516806
nodes =  0.0

Median Absolute Deviation of status 2
age =  5.930408874022408
year =  11.860817748044816
nodes =  4.447806655516806
```

Conclusion: Out of the 3 features (age, year, nodes) age and nodes are crucial in finding the patients survival rate for more than 5 years. Also as the increase in number of auxilary nodes

increases the risk of not surviving for longer than 5 years.