Assignment-1: Data Visualization with Haberman Dataset [M]

HABERMAN DATASET ¶

The dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer.

About This File

1. Title: Haberman's Survival Data
2. Sources: (a) Donor: Tjen-Sien Lim (limt@stat.wisc.edu (mailto:limt@stat.wisc.edu)) (b) Date: March 4, 1999

3. Past Usage:

   a. Haberman, S. J. (1976). Generalized Residuals for Log-Linear Models, Proceedings of the 9th International Biometrics Conference, Boston, pp. 104-122.

   b. Landwehr, J. M., Pregibon, D., and Shoemaker, A. C. (1984), Graphical Models for Assessing Logistic Regression Models (with discussion), Journal of the American Statistical Association 79: 61-83.

   c. Lo, W.-D. (1993). Logistic Regression Trees, PhD thesis, Department of Statistics, University of Wisconsin, Madison, WI.

4. Relevant Information: The dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients who had undergone surgery for breast cancer.
5. Number of Instances: 306
6. Number of Attributes: 4 (including the class attribute)
7. Attribute Information:

   a. Age of patient at time of operation (numerical)

   b. Patient's year of operation (year - 1900, numerical)

   c. Number of positive axillary nodes detected (numerical)

   d. Survival status (class attribute) 1 = the patient survived 5 years or longer 2 = the patient died within 5 year

8. Missing Attribute Values: None

Objective: Classify the status/class of patient who undergone surgery i.e 1 or 2

In [28]:

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

In [29]:

```
'''downlaod haberman.csv from https://www.kaggle.com/gilsousa/habermans-survival-data-set/v
#Load Haberman.csv into a pandas dataFrame.
habm = pd.read_csv("haberman.csv")
```

(Q) how many data-points and features?

In [30]:

```
print (habm.shape)
```

```
(306, 4)
```

(Q) What are the column names in our dataset?

In [31]:

```
print (habm.columns)
```

```
Index(['Age', 'Year', 'Nodes', 'Status'], dtype='object')
```

Applying head() on Haberman Dataset

In [32]:

```
habm.head()
```

Out[32]:

	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

(Q) How many data points for each class are present? (or) How many cases for each status are present?

In [33]:

```
habm["Status"].value_counts()
```

Out[33]:

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

Balanced V/S ImBalanced Dataset

Haberman dataset is Highly unbalanced as their is huge difference in no. of dataset of both Classes. From the above information we can also predict that our success rate will be always 73% as Category 1 (225) & Category 2 (81).

(Q) What is the Age Interval?

In [34]:

```
print( min(habm.Age), max(habm.Age) )
```

```
30 83
```

From the Above Information we come to know that No One have the Cancer Below the Age of 30 & Above the Age of 83

(Q) What is Operational Year Interval?

In [35]:

```
print ( min(habm.Year), max(habm.Year) )
```

```
58 69
```

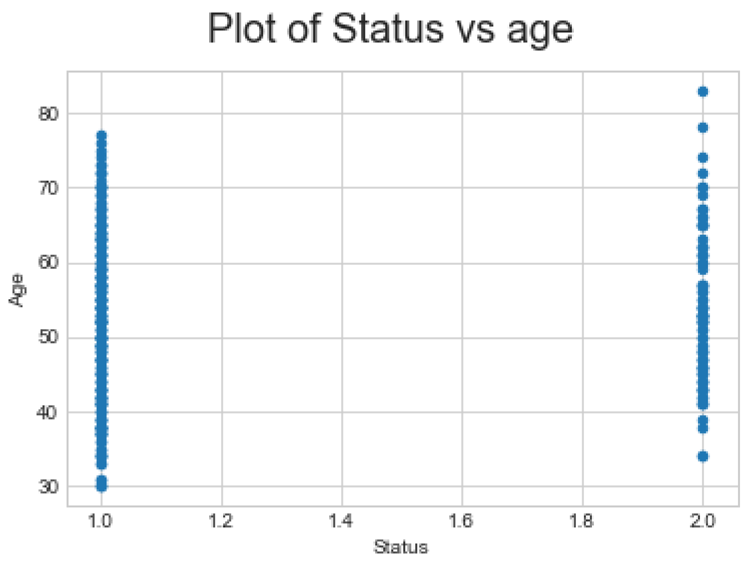
From the Above Information we come to know that Cancer Operation can be Done Above the Age of 58 & Below the Age of 69

2-D Scatter Plot

ALWAYS understand the axis: labels and scale.

In [36]:

```
habm.plot(kind='scatter', x='Status', y='Age') ;
plt.suptitle("Plot of Status vs age",size =20);
plt.show()
```

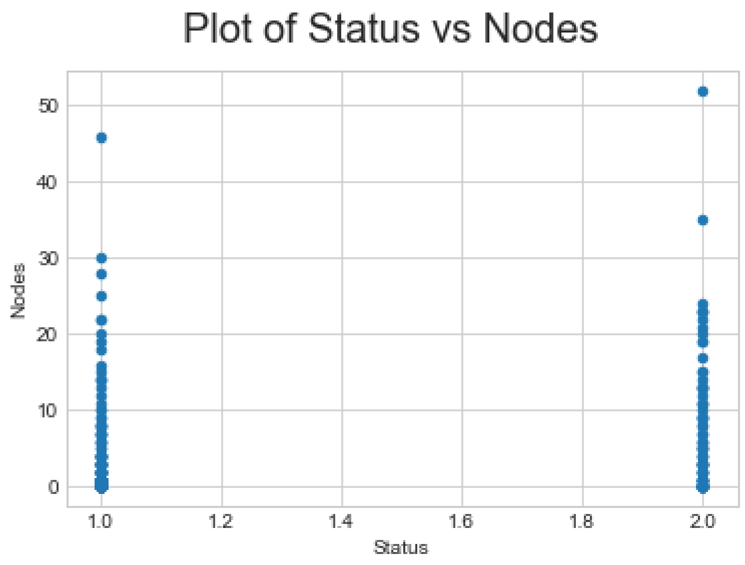


Observations -

1. Here patient having age > 77 will not survive after the Surgery.
2. For patients having age < 40, success rate is above 90%.
3. Sp, here we can us the IF & ELSE condition to predict our results.

In [37]:

```
habm.plot(kind='scatter', x='Status', y='Nodes') ;
plt.suptitle("Plot of Status vs Nodes",size =20);
plt.show()
```

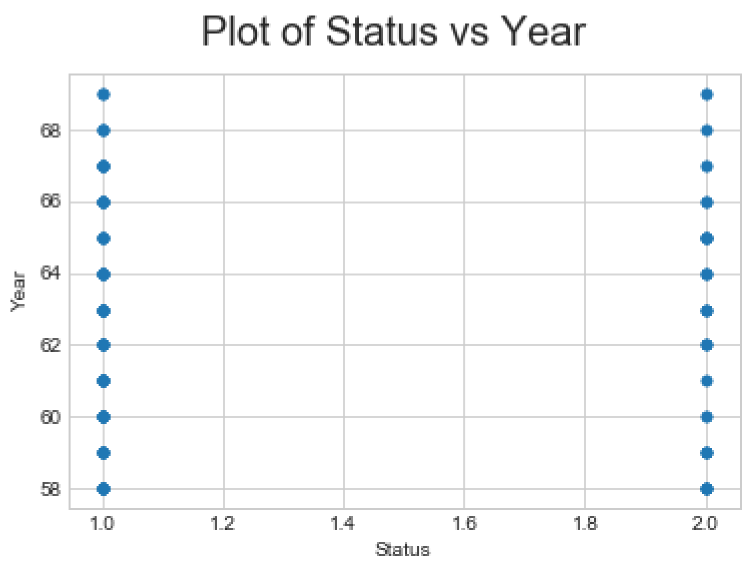


Observation -

If Nodes are Less than 18 then Success rate is 100% & If Nodes are Greater than 18 then Success rate is ~50% Therefore, we can use If & Else condtion to predict our Results.

In [38]:

```
habm.plot(kind='scatter', x='Status', y='Year') ;
plt.suptitle("Plot of Status vs Year",size =20);
plt.show()
```

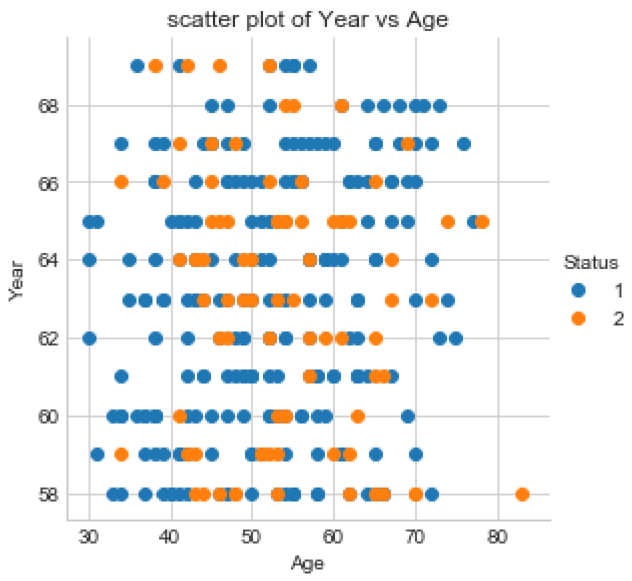


Observation - Year of Treatment is not useful in Categorizing Patient into Category 1 or 2

2-D Scatter plot with color-coding for each Patient class

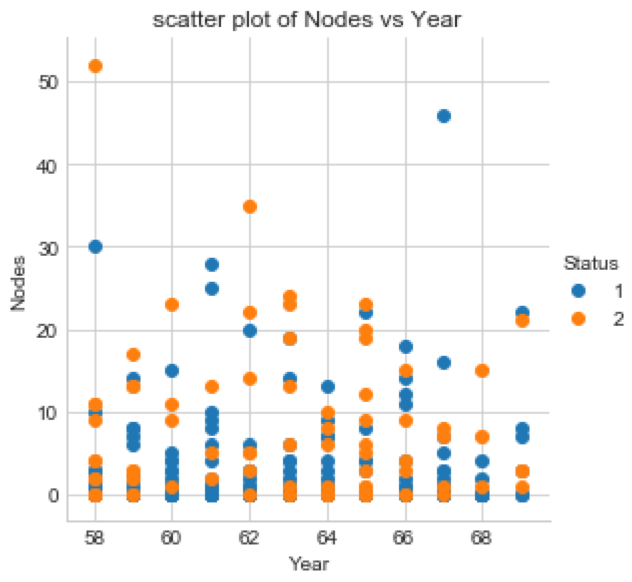
In [39]:

```
# Here 'sns' corresponds to seaborn.
sns.set_style("whitegrid");
sns.FacetGrid(habm, hue="Status", size=4) \
   .map(plt.scatter, "Age", "Year") \
   .add_legend();
plt.title('scatter plot of Year vs Age')
plt.show();

# Notice that the blue points cannot be easily seperated from orange points.
```



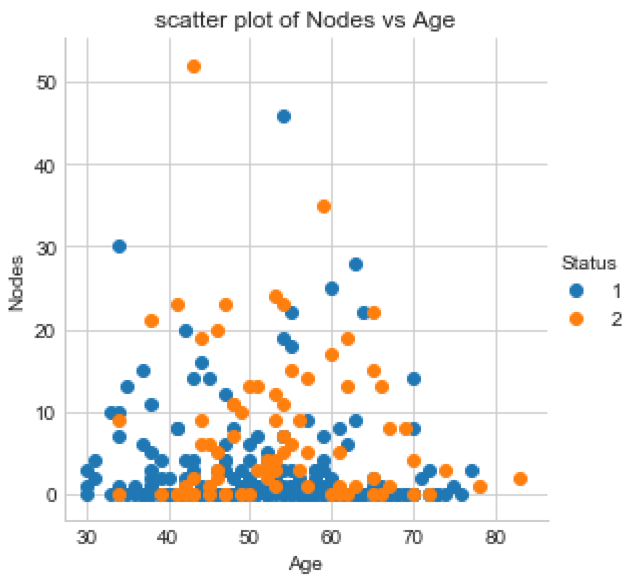
In [40]:

```
# Here 'sns' corresponds to seaborn.
sns.set_style("whitegrid");
sns.FacetGrid(habm, hue="Status", size=4) \
   .map(plt.scatter, "Year", "Nodes") \
   .add_legend();
plt.title('scatter plot of Nodes vs Year')
plt.show();

# Notice that the blue points cannot be easily seperated from orange points.
```



In [41]:

```
# Here 'sns' corresponds to seaborn.
sns.set_style("whitegrid");
sns.FacetGrid(habm, hue="Status", size=4) \
   .map(plt.scatter, "Age", "Nodes") \
   .add_legend();
plt.title('scatter plot of Nodes vs Age')
plt.show();
```



Observation -

1. In Color-Coding Scatter plot of Nodes & Age, we can see some groups of Orange & Blue but are no clear Clusters.
2. Seperating Category 1 patients from Category 2 patients is Hard as there is so much Overlap.

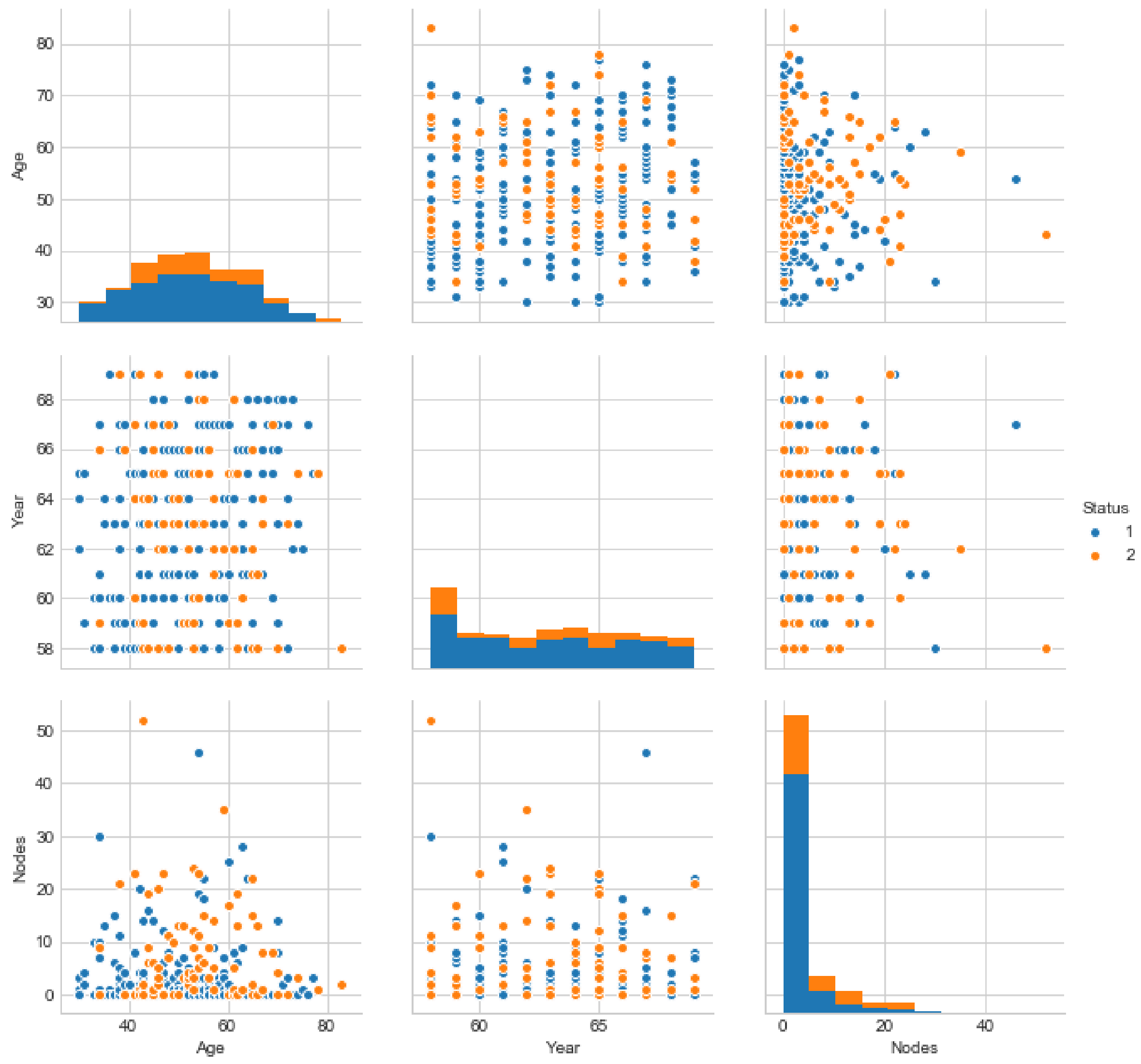
3D Scatter plot

Needs a lot to mouse interaction to interpret data.

Pair-plot

In [42]:

```
# pairwise scatter plot: Pair-Plot
# Dis-advantages:
##Can be used when number of features are high.
##Cannot visualize higher dimensional patterns in 3-D and 4-D.
#Only possible to view 2D patterns.
plt.close();
sns.set_style("whitegrid");
sns.pairplot(habm, hue="Status" , x_vars=['Age','Year','Nodes'] , y_vars=['Age','Year','Noc
plt.show()
# NOTE: the diagnol elements are PDFs for each feature. PDFs are expalined below.
```



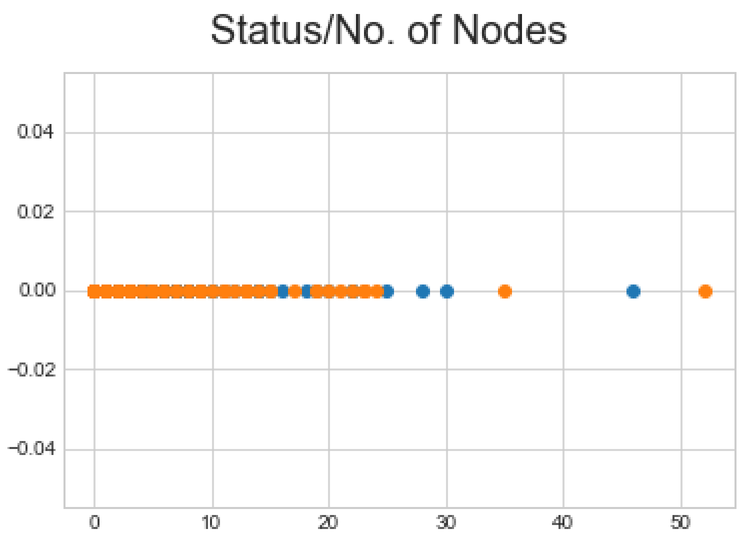
Observations -

1. It is very Difficult to predict the Class of Patient as the Graphs are Highly Overlapped.
2. Graphs cannot be seperated linearly.
3. We need some more information to classify points correctly.

Histogram

In [43]:

```
# What about 1-D scatter plot using just one feature?
#1-D scatter plot of petal-length
import numpy as np
plt.suptitle("Status/No. of Nodes",size=20);

habm_1 = habm.loc[habm["Status"] == 1];
habm_2 = habm.loc[habm["Status"] == 2];

plt.plot(habm_1["Nodes"], np.zeros_like(habm_1['Nodes']), 'o')
plt.plot(habm_2["Nodes"], np.zeros_like(habm_2['Nodes']), 'o')


plt.show()


#Disadvantages of 1-D scatter plot: Very hard to make sense as points
#are overlapping a lot.
#Are there better ways of visualizing 1-D scatter plots?
```

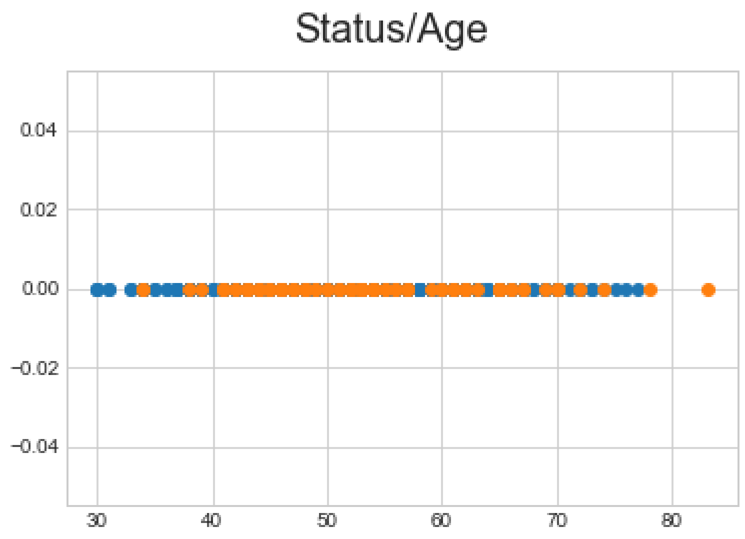


Observations -

1. If the No. of Nodes > 25, then their are More chances that the Patient will Die before 5 years of Treatment i.e Status - 2.
2. for No. of Nodes < 25, we cannot say Clearly as Data is Highly Overlapped.

In [44]:

```
import numpy as np
plt.suptitle("Status/Age",size=20);

habm_1 = habm.loc[habm["Status"] == 1];
habm_2 = habm.loc[habm["Status"] == 2];

plt.plot(habm_1["Age"], np.zeros_like(habm_1['Age']), 'o')
plt.plot(habm_2["Age"], np.zeros_like(habm_2['Age']), 'o')


plt.show()
```

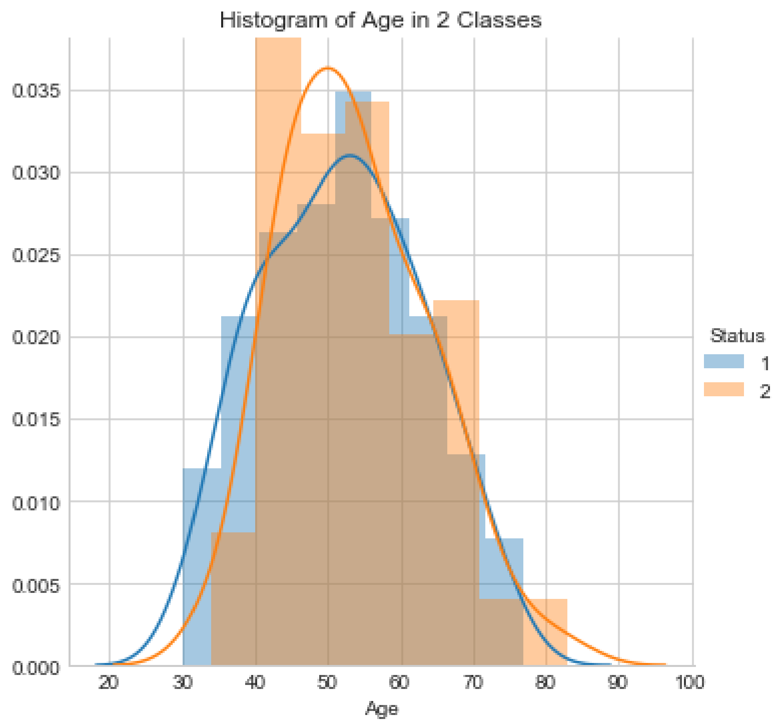


Observations-

1. Between Age of 68 & 77 Years, Success Rate is ~50%.
2. Above the Age of 77 Years, Success Rate is ~0%.

In [45]:

```
sns.FacetGrid(habm, hue="Status", size=5) \
   .map(sns.distplot, "Age") \
   .add_legend();
plt.title('Histogram of Age in 2 Classes')
plt.show();
```

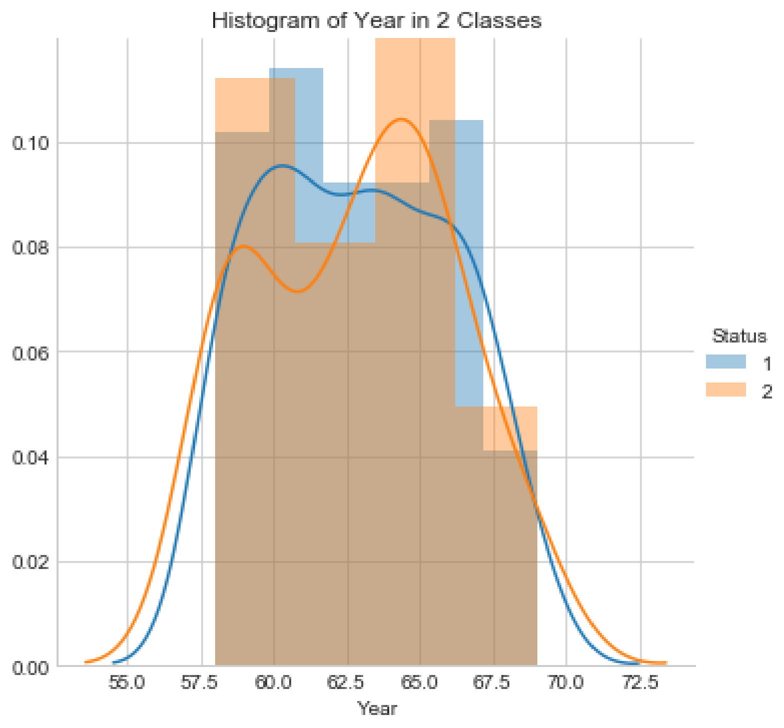


Observation -

1. Graph is quite Complex & Hard to conclude.
2. Density of Patient with Status 1 is Higher for Age between 30 to 40.

In [46]:

```
sns.FacetGrid(habm, hue="Status", size=5) \
   .map(sns.distplot, "Year") \
   .add_legend();
plt.title('Histogram of Year in 2 Classes')
plt.show();
```

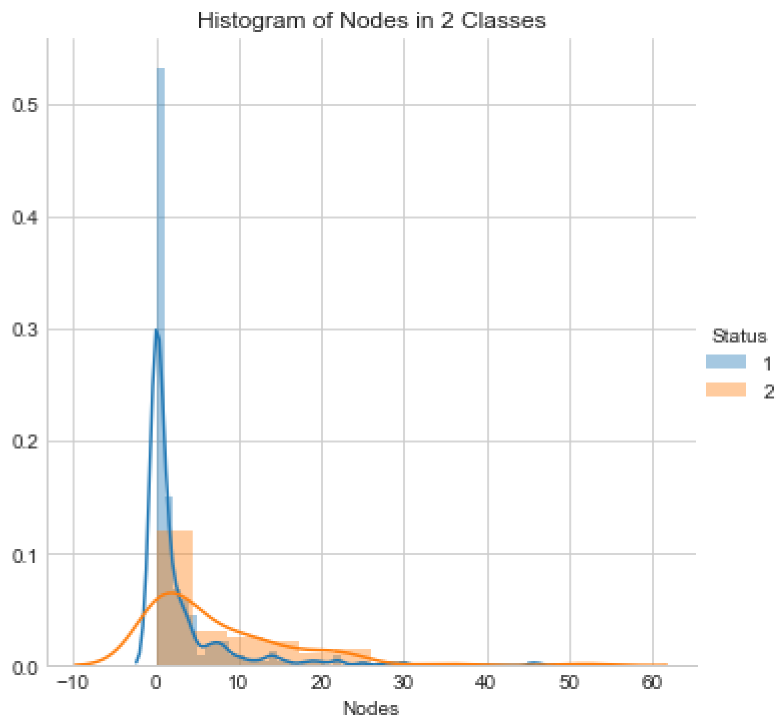


Observation -

This Graph Shows that Feature "Treatment Year" is not useful in predicting the results.

In [47]:

```
sns.FacetGrid(habm, hue="Status", size=5) \
   .map(sns.distplot, "Nodes") \
   .add_legend();
plt.title('Histogram of Nodes in 2 Classes')
plt.show();
```



Obesrvations-

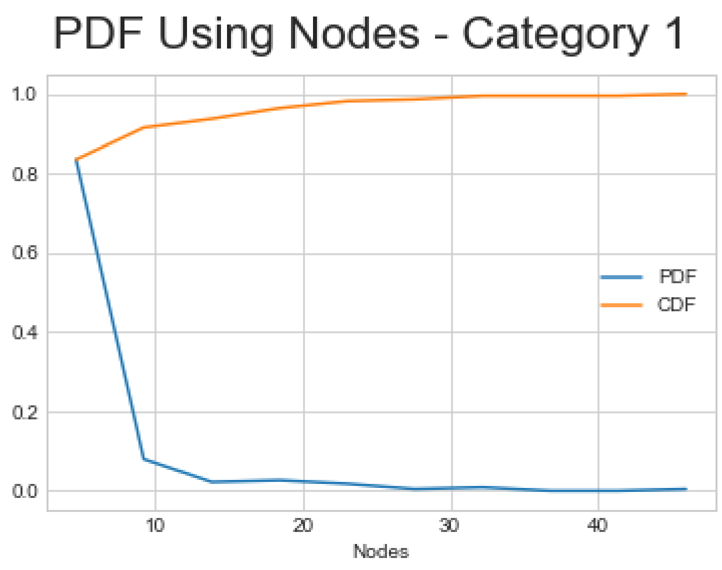
1. This Graph depicts that the Probability of Patient with Status 1 i.e Survived is Higher when Nodes are Between -2 to 8.
2. Probability of patient Survived is Highest when Nodes are between 0 to 4.
3. For Nodes Greater than 30, Graph depicts Patient with Status 2 with a Very small probability of Class 1 which is very less.

PDF & CDF

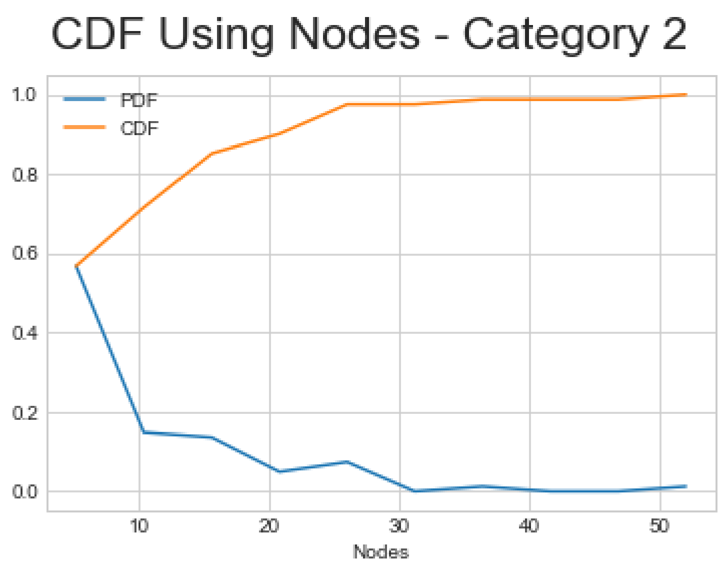
In [48]:

```
#PDF
counts, bin_edges = np.histogram(habm_1['Nodes'], bins=10,
                                 density = True)
plt.suptitle("PDF Using Nodes - Category 1",size=23);
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges);
cdf = np.cumsum(pdf)

plt.plot(bin_edges[1:],pdf,label = 'PDF');
plt.plot(bin_edges[1:], cdf,label = 'CDF')
plt.legend();
plt.xlabel("Nodes");


plt.show();

#CDF
counts, bin_edges = np.histogram(habm_2['Nodes'], bins=10,
                                 density = True)
plt.suptitle("CDF Using Nodes - Category 2",size=23);
pdf = counts/(sum(counts))
print(pdf);
print(bin_edges);
cdf = np.cumsum(pdf)

plt.plot(bin_edges[1:],pdf,label = 'PDF');
plt.plot(bin_edges[1:], cdf,label = 'CDF')
plt.legend();
plt.xlabel("Nodes");


plt.show();
```

```
[0.83555556 0.08       0.02222222 0.02666667 0.01777778 0.00444444
 0.00888889 0.         0.         0.00444444]
[ 0.   4.6  9.2 13.8 18.4 23.  27.6 32.2 36.8 41.4 46. ]
```



```
[0.56790123 0.14814815 0.13580247 0.04938272 0.07407407 0.
 0.01234568 0.         0.         0.01234568]
[ 0.   5.2 10.4 15.6 20.8 26.  31.2 36.4 41.6 46.8 52. ]
```



Observation -

1. For Category 1, Around 90% of Patients survived after treatment whose Nodes < 10 i.e lower the no. of Nodes, higher is the chance of survival.
2. For Category 2, Around 70% of Patients having Nodes < 10, died before 5 years of Treatment And Almost 99% died having Nodes < 25.

Mean, Variance and Std-dev

In [49]:

```
#Mean, Variance, Std-deviation,
print("Means:")
print(np.mean(habm_1["Nodes"]))
#Mean with an outlier.
print(np.mean(np.append(habm_1["Nodes"],50)));

print(np.mean(habm_2["Nodes"]))

print("\nStd-dev:");
print(np.std(habm_1["Nodes"]))
print(np.std(habm_2["Nodes"]))
```

```
Means:
2.7911111111111113
3.0
7.45679012345679

Std-dev:
5.857258449412131
9.128776076761632
```

Median, Percentile, Quantile, IQR, MAD

In [50]:

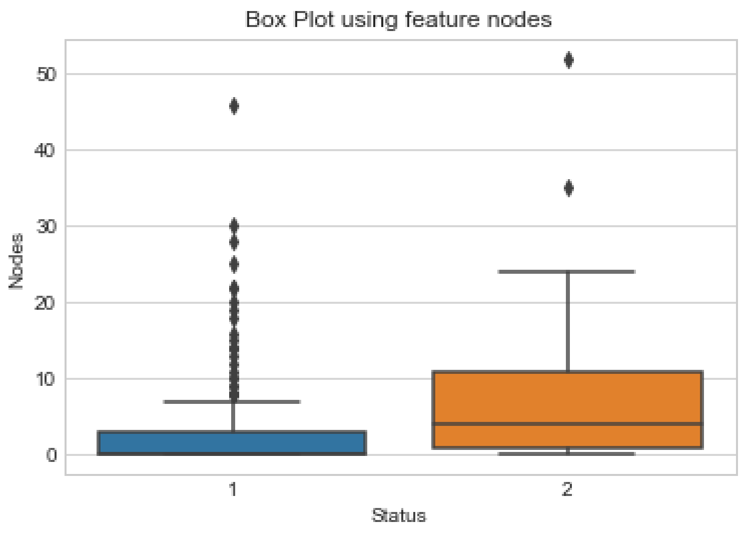
```
#Median, Quantiles, Percentiles, IQR.
print("\nMedians:")
print(np.median(habm_1["Nodes"]))
#Median with an outlier
print(np.median(np.append(habm_1["Nodes"],50)));

print(np.median(habm_2["Nodes"]))


print("\nQuantiles:")
print(np.percentile(habm_1["Nodes"],np.arange(0, 100, 25)))
print(np.percentile(habm_2["Nodes"],np.arange(0, 100, 25)))

print("\n90th Percentiles:")
print(np.percentile(habm_1["Nodes"],90))
print(np.percentile(habm_2["Nodes"],90))

from statsmodels import robust
print ("\nMedian Absolute Deviation")
print(robust.mad(habm_1["Nodes"]))
print(robust.mad(habm_2["Nodes"]))
```

```
Medians:
0.0
0.0
4.0

Quantiles:
[0. 0. 0. 3.]
[ 0.  1.  4. 11.]

90th Percentiles:
8.0
20.0

Median Absolute Deviation
0.0
5.930408874022408
```

Box plot and Whiskers

In [51]:

```
#Box-plot with whiskers: another method of visualizing the  1-D scatter plot more intuitive
# How to draw whiskers: [no standard way] Could use min and max or use other complex statis

#NOTE: IN the plot below, a technique call inter-quartile range is used in plotting the whi
#Whiskers in the plot below donot correposnd to the min and max values.

#Box-plot can be visualized as a PDF on the side-ways.

sns.boxplot(x='Status',y='Nodes', data=habm)
plt.title("Box Plot using feature nodes");
plt.show()
```



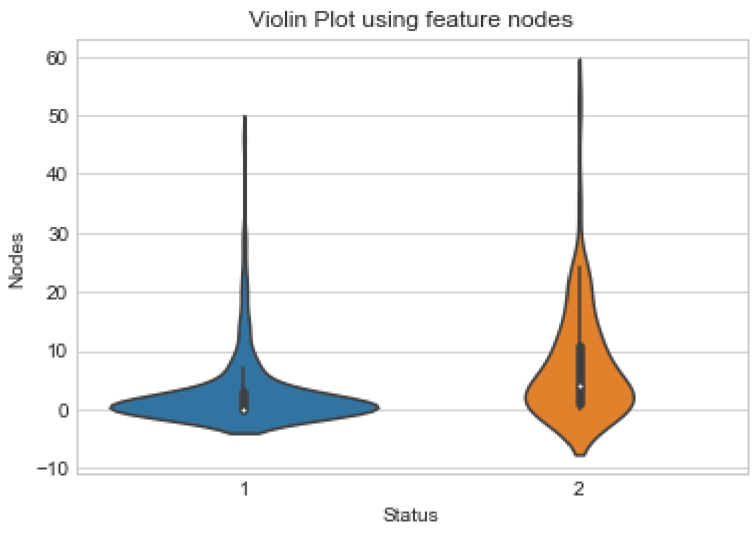
Observations -

1. We can say that when Nodes >= 2 & Nodes <11, then Patient didn't Survived.
2. But by the above assumption, we can see that approx. 50% of patients are wrongly classified to be not survived i.e 50% Error.

Violin plots

In [52]:

```
# A violin plot combines the benefits of the previous two plots
#and simplifies them

# Denser regions of the data are fatter, and sparser ones thinner
#in a violin plot

sns.violinplot(x="Status", y="Nodes", data=habm, size=8)
plt.title("Violin Plot using feature nodes");
plt.show()
```



Observations-

1. If No. of Nodes < 10, the Success rate is 80%.
2. If No. of Nodes > 10, the Failure rate is 70%.