EDAHaberman (1)

May 30, 2018

In [0]:
```
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
```

In [0]:
```
# Code to read csv file into colaboratory:
!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials

# 1. Authenticate and create the PyDrive client.
auth.authenticate_user()
gauth = GoogleAuth()
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

In [0]:
```
downloaded = drive.CreateFile({'id':'1AVECm80ET0dLQOkEHaReyh__jbJsjnhZ'}) # replace th
downloaded.GetContentFile('haberman.csv')
```

In [0]:
```
#Loading haberman data into pandas dataframe
column_names = ['Age', 'Operation_year', 'Auxillary_nodes', 'Status']
hbdata = pd.read_csv("haberman.csv", names = column_names)
```

In [5]:
```
#No. of data points and features:
print(hbdata.shape)
```

```
(306, 4)
```

In [6]:
```
#Columnns in the dataset
print(hbdata.columns)
```

```
Index(['Age', 'Operation_year', 'Auxillary_nodes', 'Status'], dtype='object')
```

In [7]:
```
#No of data points for each class (The ones who survive and ones who do not)
hbdata['Status'].value_counts()
```

Out[7]:
```
1    225
2     81
Name: Status, dtype: int64
```

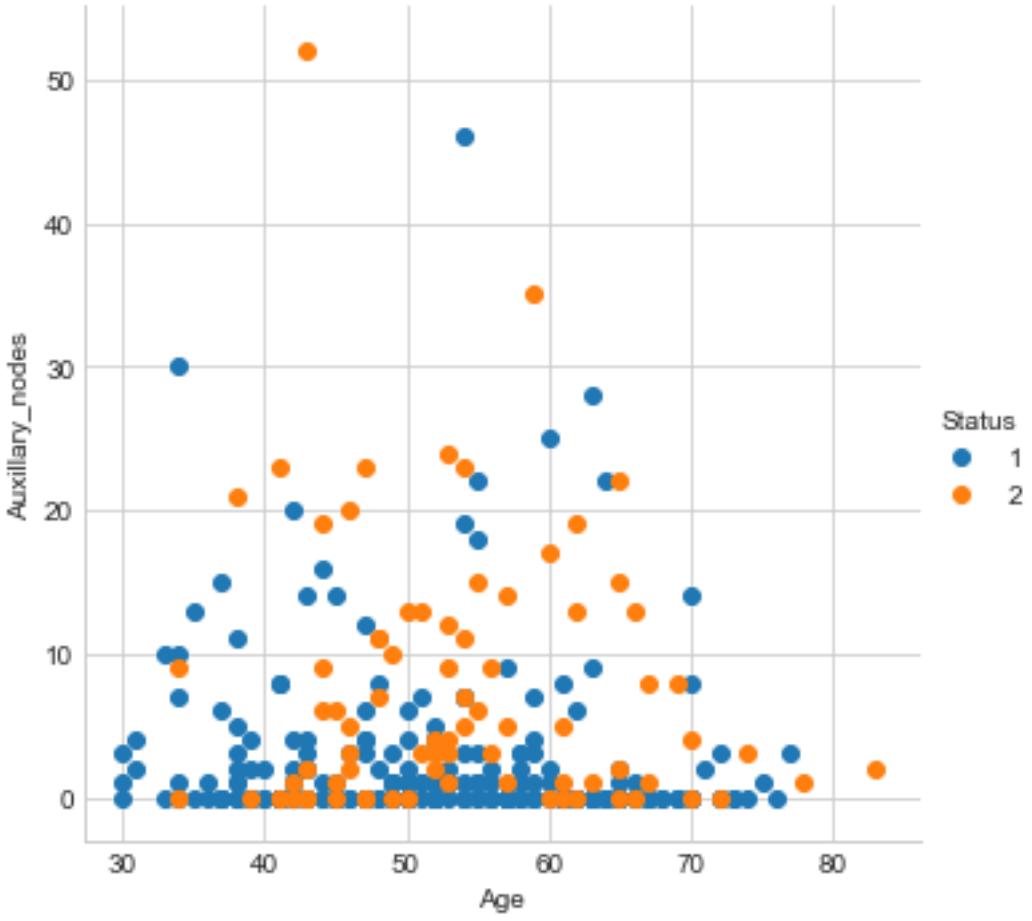
1 Observations:

1.1 There are two classes in this dataset(Unbalanced)

- **Class 1** belongs to the no of patients who survived even after 5 years : 225.
- **Class 2** belongs to the number of patient who unfortunately don't make it within 5 years : 81.

2 2D Scatter Plot

```
In [0]: sb.set_style("whitegrid");
        sb.FacetGrid(hbdata, hue = "Status", size = 5) \
          .map(plt.scatter, "Age", "Auxillary_nodes") \
          .add_legend();

        plt.show();
```



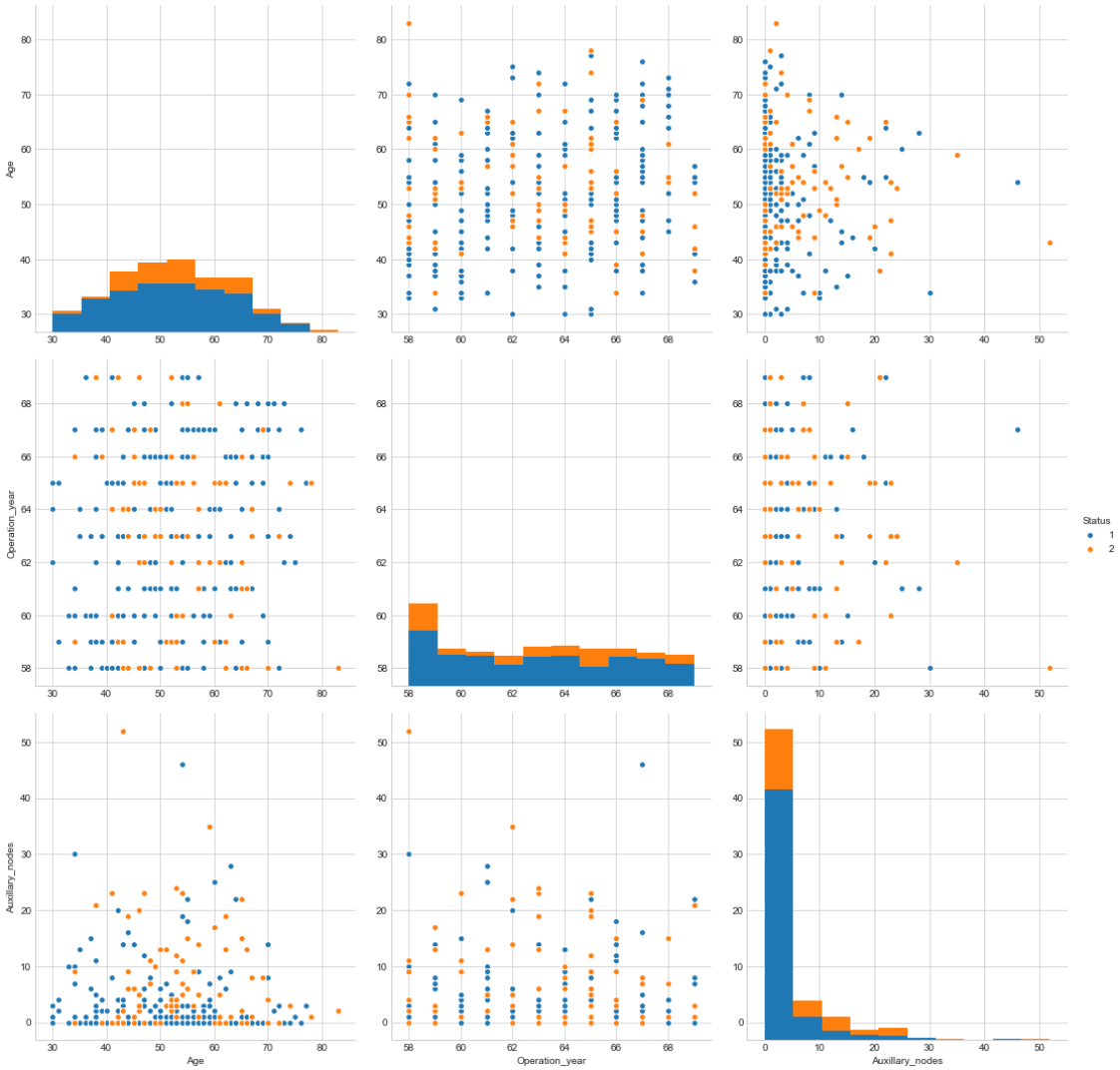
3 Observations

The number of patients are **denser** at the bottom signifying that most of the patients have **zero** auxillary nodes.

4 Pair-Plot

In [0]: *#Pair wise scatter-plot*

```
plt.close();
sb.set_style("whitegrid");
sb.pairplot(hbdata, hue = "Status", vars = ["Age", "Operation_year", "Auxillary_nodes"]
plt.show()
```

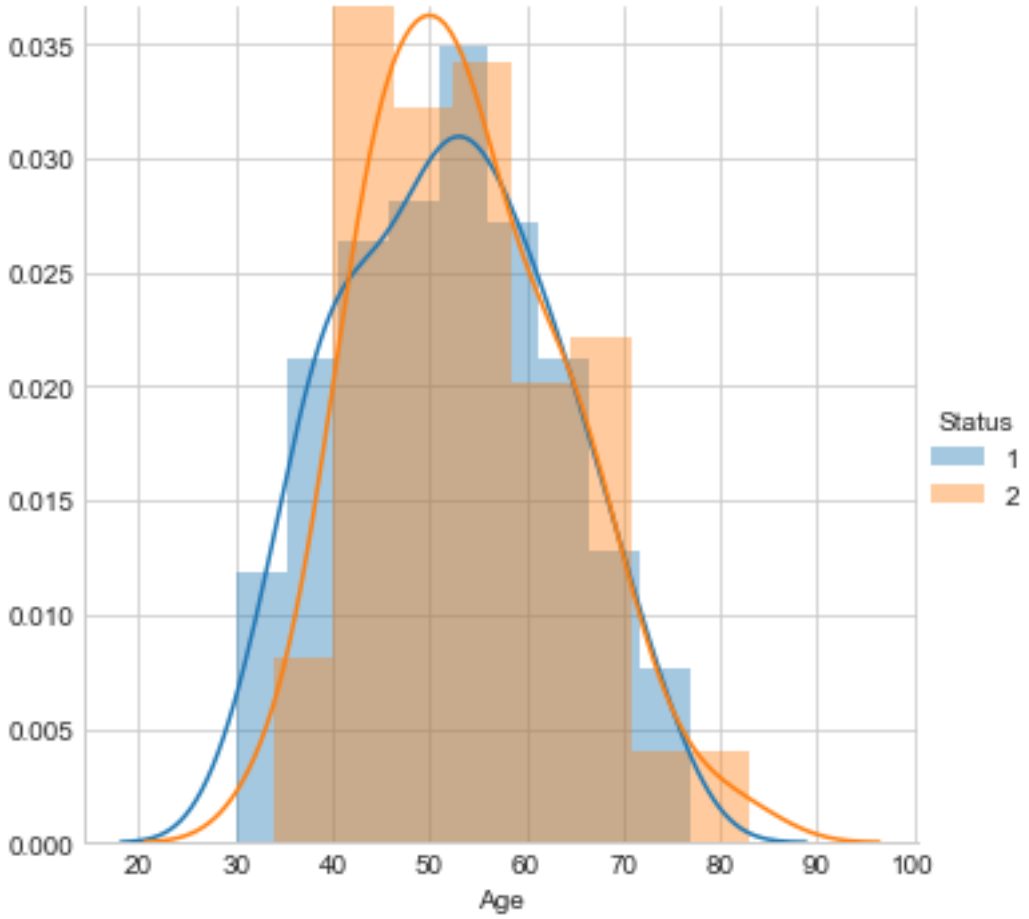


4.1 Observations

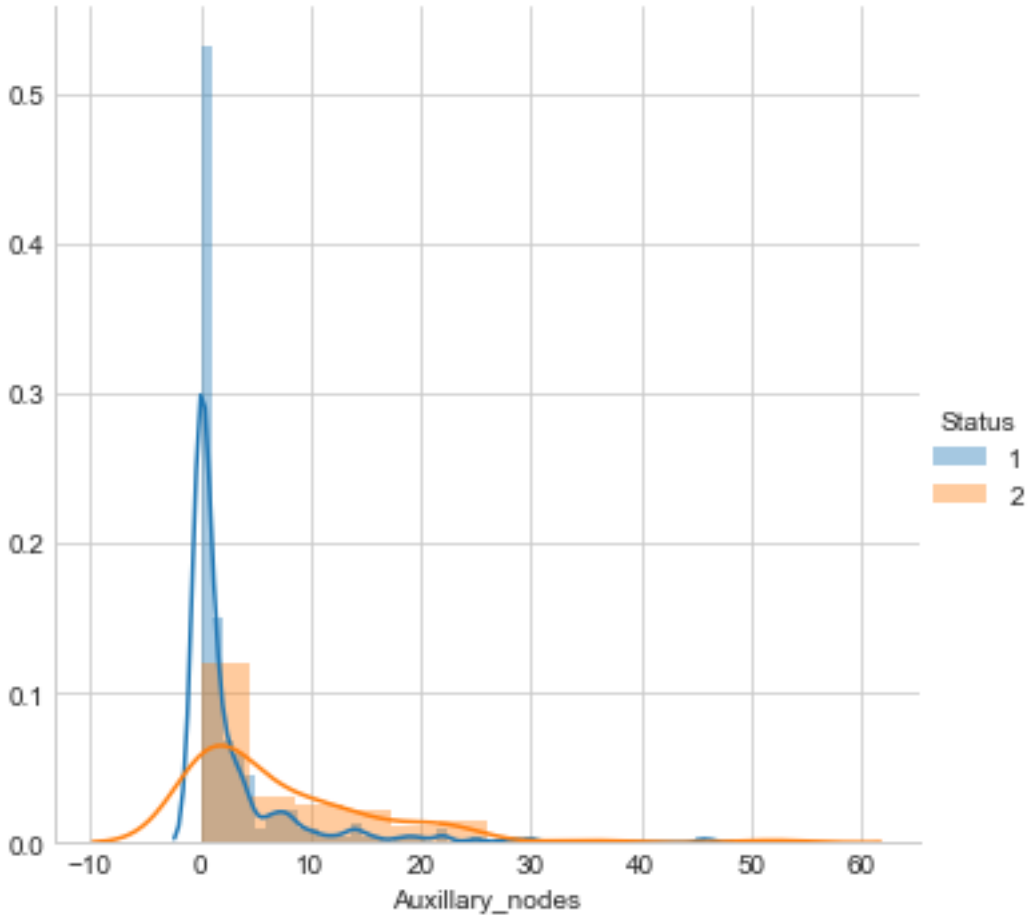
- The distribution is pretty random but among these 'Age' and 'Auxillary nodes' are the most useful features to identify the survival status.

5 Histogram, PDF, CDF

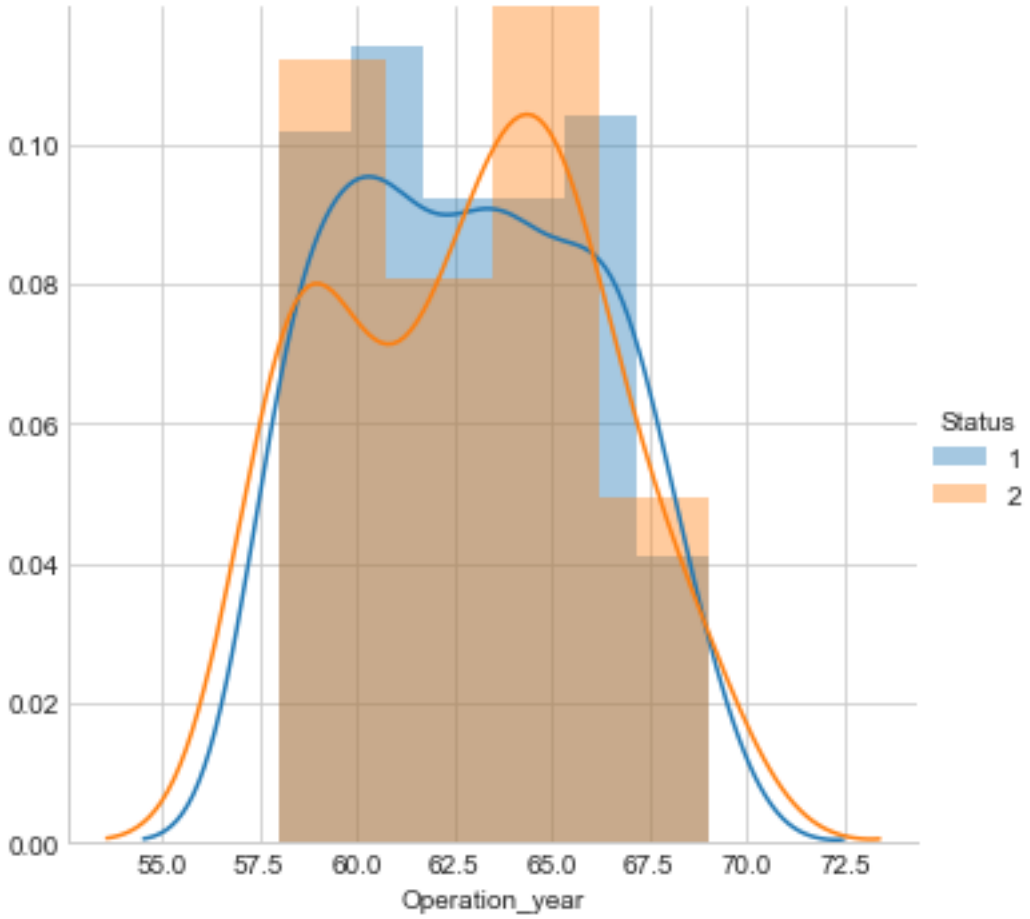
```
In [0]: sb.FacetGrid(hbdata, hue = "Status", size = 5) \
          .map(sb.distplot, "Age") \
          .add_legend();

        plt.show();
```




```
In [0]: sb.FacetGrid(hbdata, hue = "Status", size = 5) \
          .map(sb.distplot, "Auxillary_nodes") \
          .add_legend();

        plt.show();
```




```
In [0]: sb.FacetGrid(hbdata, hue = "Status", size = 5) \
          .map(sb.distplot, "Operation_year") \
          .add_legend();

        plt.show();
```



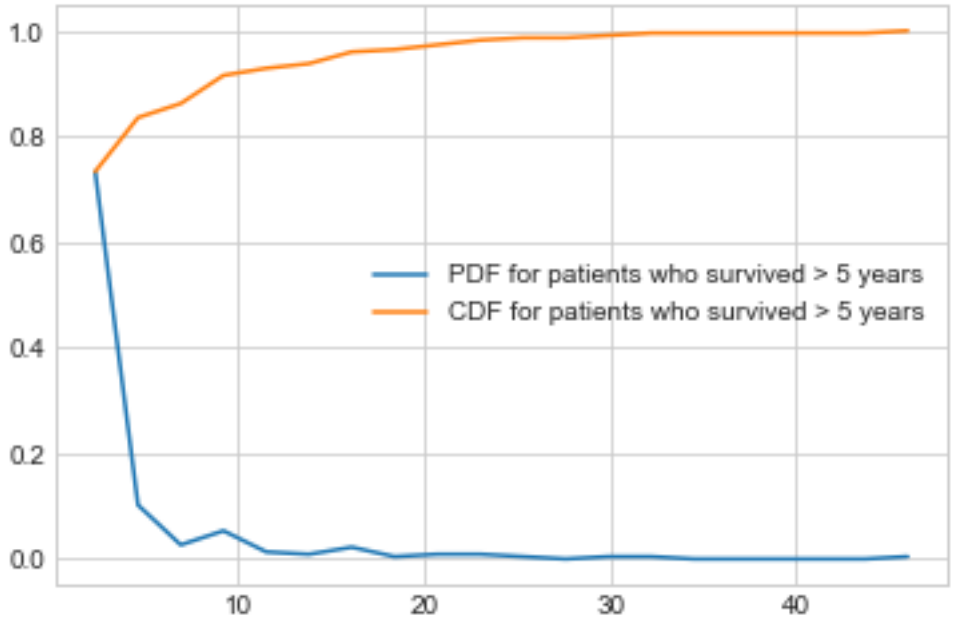
5.1 Observations

- The attributes **Age** and **Operation_year** dont give much details as there is a lot of **overlapping** of data.
- The attribute **Auxillary_node** on the other hand gives useful insights.We can observe that most of the patients have **zero** positive auxillary nodes.

```
In [0]: #There are two classes of Survival,ones who survive after 5 years(Class 1) and ones wh
        hb1 = hbdata.loc[hbdata["Status"] == 1]
        hb2 = hbdata.loc[hbdata["Status"] == 2]

In [0]: #Plotting PDF and CDF of Auxillary Nodes

        counts, bin_edges = np.histogram(hb1["Auxillary_nodes"], bins = 20, density = True)
        pdf = counts / (sum(counts))
        print(pdf)
        print(bin_edges)
        cdf = np.cumsum(pdf)
```

```
        plt.plot(bin_edges[1:], pdf);
        plt.plot(bin_edges[1:], cdf)
        plt.legend(['PDF for patients who survived > 5 years', 'CDF for patients who survived >

        plt.show()
```

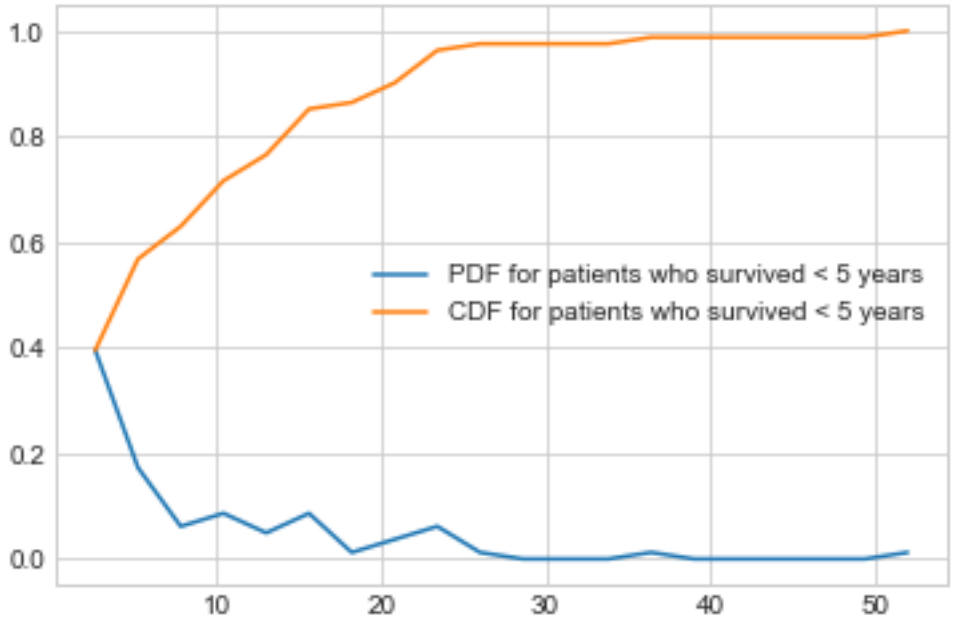
```
[ 0.73333333  0.10222222  0.02666667  0.05333333  0.01333333  0.00888889
  0.02222222  0.00444444  0.00888889  0.00888889  0.00444444  0.
  0.00444444  0.00444444  0.          0.          0.          0.          0.
  0.00444444]
[  0.    2.3   4.6   6.9   9.2  11.5  13.8  16.1  18.4  20.7  23.   25.3
  27.6  29.9  32.2  34.5  36.8  39.1  41.4  43.7  46. ]
```




```
In [0]: counts, bin_edges = np.histogram(hb2["Auxillary_nodes"], bins = 20, density = True)
        pdf = counts / (sum(counts))
        print(pdf)
        print(bin_edges)
        cdf = np.cumsum(pdf)
        plt.plot(bin_edges[1:], pdf);
        plt.plot(bin_edges[1:], cdf)
        plt.legend(['PDF for patients who survived < 5 years', 'CDF for patients who survived <

        plt.show()
```

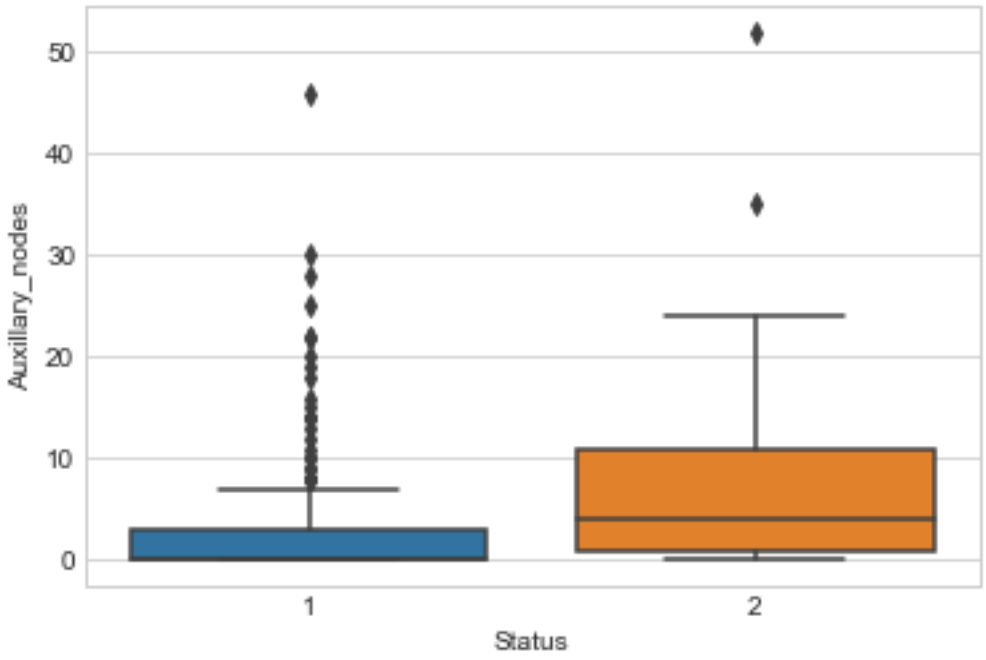
```
[ 0.39506173  0.17283951  0.0617284   0.08641975  0.04938272  0.08641975
  0.01234568  0.03703704  0.0617284   0.01234568  0.          0.          0.
```

```
  0.01234568  0.          0.          0.          0.          0.
  0.01234568]
[  0.    2.6   5.2   7.8  10.4  13.   15.6  18.2  20.8  23.4  26.   28.6
  31.2  33.8  36.4  39.   41.6  44.2  46.8  49.4  52. ]
```

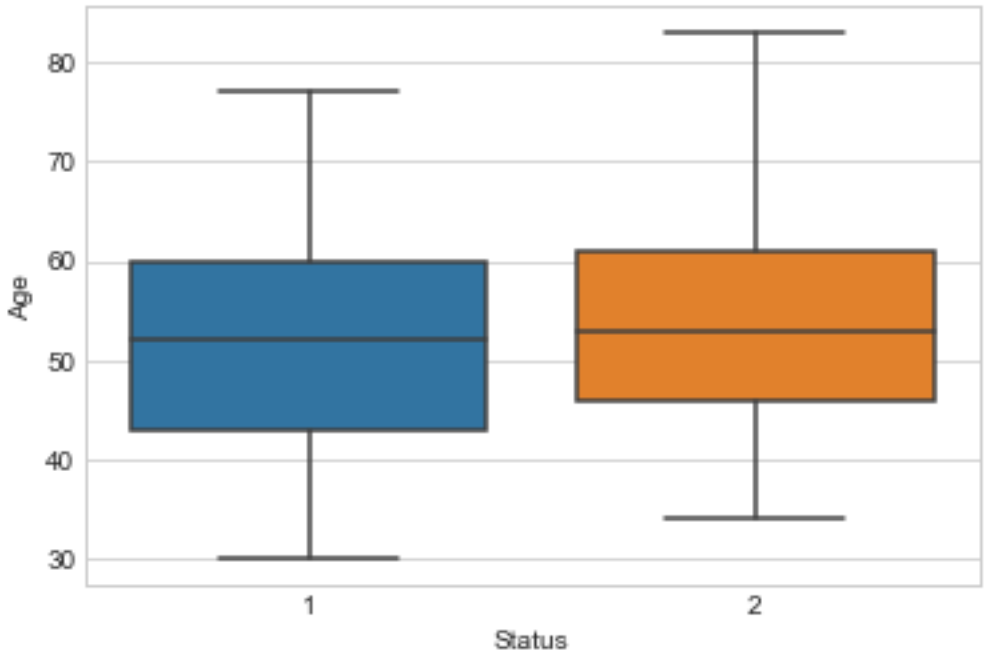


6 Box Plot

```
In [0]: sb.boxplot(x = "Status", y = "Auxillary_nodes", data = hbdata)
        plt.show()
```

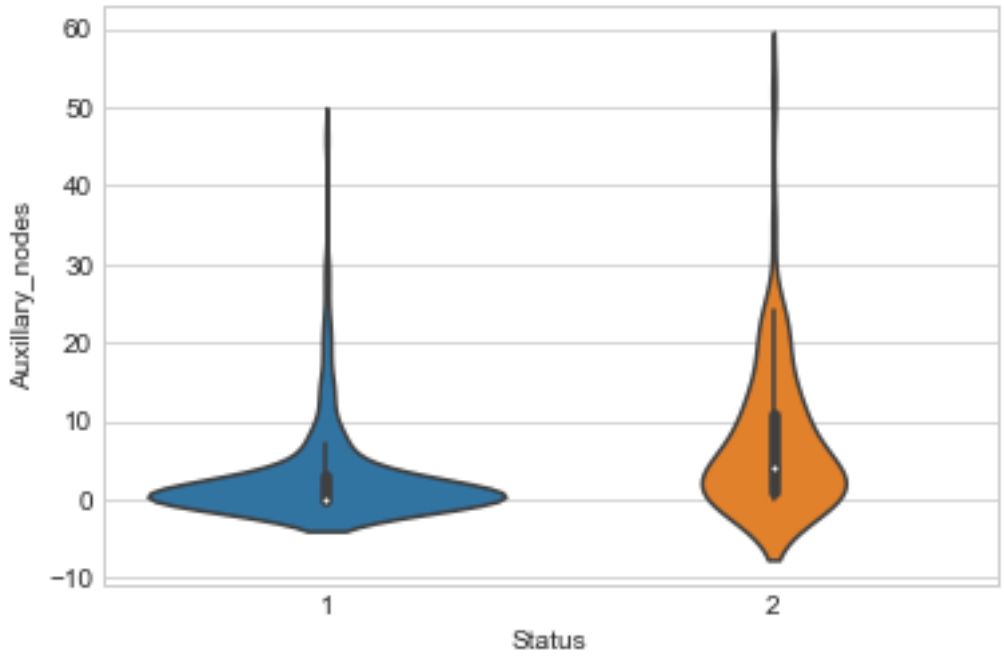



```
In [0]: sb.boxplot(x = "Status", y = "Age", data = hbdata)

        plt.show()
```

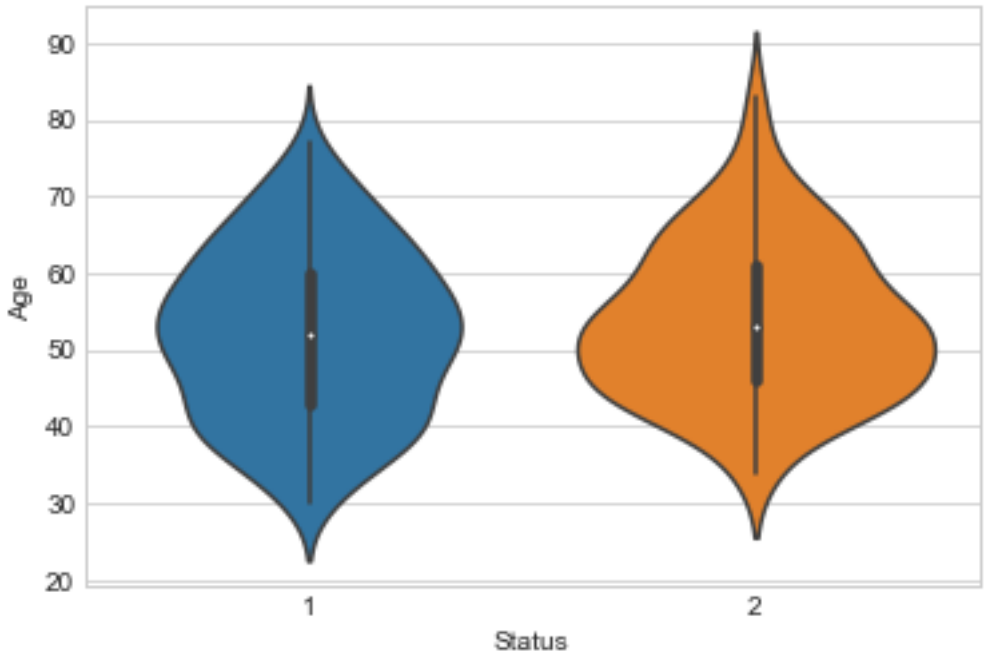


7 Violin Plots

```
In [0]: #Denser region of data are fatter and sparse ones are thinner

        sb.violinplot(x = "Status", y = "Auxillary_nodes", data = hbdata, size = 10)

        plt.show()
```

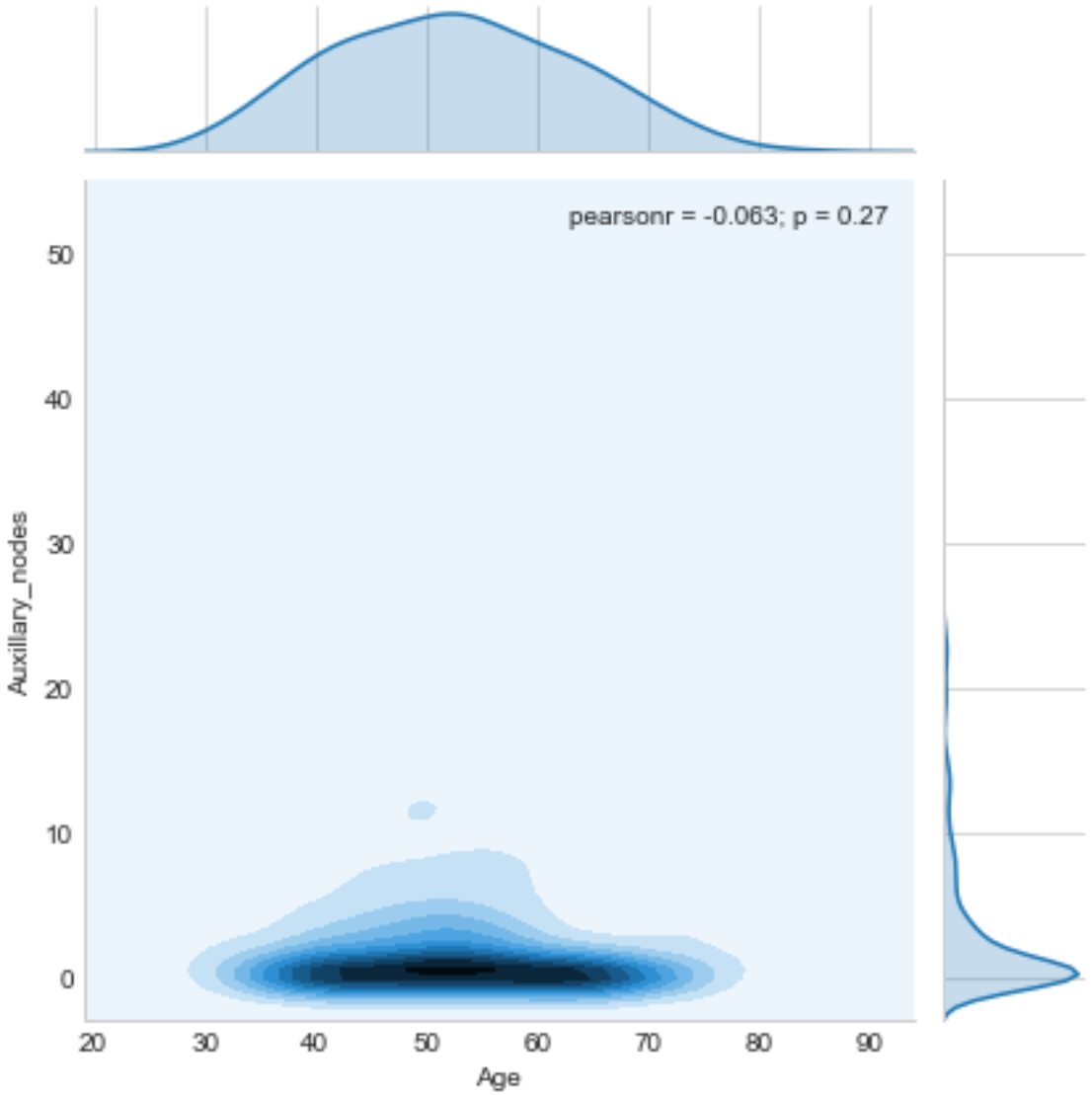



```
In [0]: sb.violinplot(x = "Status", y = "Age", data = hbdata, size = 10)
        plt.show()
```



8 Contour Plot

```
In [0]: #2D Density plot, contors-plot
        sb.jointplot(x="Age", y="Auxillary_nodes", data=hbdata, kind="kde");
        plt.show();
```



In [36]:
```
no_people = hbdata.shape[0]
survivors_40 = hbdata.query('Age >= 40').query('Age <= 60').query('Status == 1').shape
nsurvivors_40 = hbdata.query('Age >= 40').query('Age <= 60').query('Status == 2').sha

nsurvivors_total = hbdata.query('Status == 2').shape[0]
survivors_total = hbdata.query('Status == 1').shape[0]

np_40 = (nsurvivors_40/no_people)*100
p_40 = (survivors_40/no_people)*100
np_total = (nsurvivors_total/no_people)*100
p_total = (survivors_total/no_people)*100

print("Total number of patients who did survive is {} i.e {:.2f} %".format(survivors_
```

```
        print()
        print("Total number of patients who did not survive is {} i.e {:.2f} %".format(nsurvi
        print()
        print("Total number of patients between the age of 40 to 60 who did not survive is {}
        print()
        print("Total number of patients between the age of 40 to 60 who survived is {} i.e {:
```

```
Total number of patients who did survive is 225 i.e 73.53 %

Total number of patients who did not survive is 81 i.e 26.47 %

Total number of patients between the age of 40 to 60 who did not survive is 55 i.e 17.97 %

Total number of patients between the age of 40 to 60 who survived is 134 i.e 43.79 %
```

9 Conclusion

- The group of people in the **age group** of **40-60** had more positive auxillary nodes in comparison to other age groups,hence least chance of survival.
- Total number of patients who did survive is 225 i.e 73.53 %
- Total number of patients who did not survive is 81 i.e 26.47 %
- Total number of patients between the age of 40 to 60 who did not survive is 55 i.e 17.97 %
- Total number of patients between the age of 40 to 60 who survived is 134 i.e 43.79 %

In [0]: