Exploratory Data Analysis

Exploratory Data Analysis One of the Important Part of Machine Learning, to understand How the feature are distributed, How the feature impact the model,To understand about outlier,mean,median,Iqr,Understand the data Based Visualization

Haberman's 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.

In this DataSet, Analysis How Many patients are Survive More than Five Year after surgery,How many Patients are survive less than five year,Let's Start with our Analysis

In [2]:

```
#import the necessary Library
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:

```
# read the Csv File store into the variable
df = pd.read_csv("/content/haberman.csv")

print(df.head())
```

```
   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
```

Observation

1,From this Dataset totally Four features are present(Age, year,nodes and status)

2.Age,Year,nodes are Independent feature, Feature Status is Dependent feature (or) target feature (or) output feature

In [7]:

```
print("Shape of the Given Data Set : {} ".format(df.shape))
```

```
Shape of the Given Data Set : (306, 4)
```

Totally 306 patients details are present in the Dataset, Each Patient have a details like Age,year of operation ,No of lymph nodes and survival status Size of the Data - Nothing but how many data points are there in the data set, There are Four features,Each Feature has 306 data points, TotalSize of our Given Data as 306 * 4 = 1224 data points are used

In [9]:

```
# print the what are the columns in the Data and understand the tupe of each feature
for column in df.columns:

  print("columnName : {0} and Type of the feature {1}".format(column,type(df[column].dtypes
)))
```

```
columnName : age and Type of the feature <class 'numpy.dtype'>
columnName : year and Type of the feature <class 'numpy.dtype'>
columnName : nodes and Type of the feature <class 'numpy.dtype'>
columnName : status and Type of the feature <class 'numpy.dtype'>
```

In [12]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306 entries, 0 to 305
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   age     306 non-null    int64
 1   year    306 non-null    int64
 2   nodes   306 non-null    int64
 3   status  306 non-null    int64
dtypes: int64(4)
memory usage: 9.7 KB
```

We can see the snippet of the code explains given information of all features, Four Features in the dataset (Each feature have 306 data point) and there is Missing Value and all the feature have a dtype integer

In [14]:

```
df.groupby(['status']).count().T
```

Out[14]:

status	**1**	**2**
age	225	81
year	225	81
nodes	225	81

In [15]:

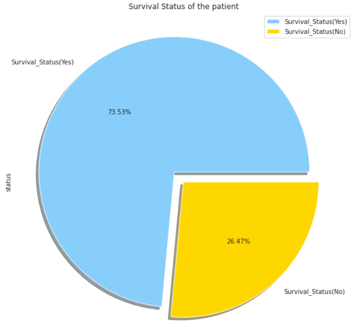
```
df.status.value_counts()
```

Out[15]:

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

In [35]:

```
plt.figure(figsize=(10,10))
plt.title("Survival Status of the patient")
labels = ["Survival_Status(Yes)" ,"Survival_Status(No)"]
colors = ["lightskyblue" ,"gold"]
df.status.value_counts().plot.pie(explode = [0.1,0], autopct = '%0.2f%%',shadow = True,label
s = labels ,colors = colors)
plt.legend(labels , loc = "upper right")
plt.show()
```



As We underStand the pie chart, Totallt 73.50 % of people survied more than five year after treatment , Only 24.50% peoples are survived less than 5 years

In [104]:
```
#description about the dataset
df.describe()
```

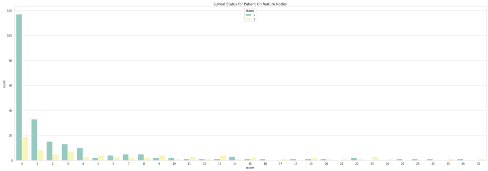
Out[104]:

	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

This shows the Age of the people lies between 30 to 83 Although the maximum number of positive axillary nodes observed is 52, nearly 75% of the patients have less than 5 positive axillary nodes and nearly 25% of the patients have no positive axillary nodes

In [41]:
```
plt.figure(figsize=(30,10))
sns.countplot(x = 'nodes',hue = 'status',data = df,palette='Set3')
plt.title('Survial Status for Patient On feature Nodes')
```

Out[41]:
```
Text(0.5, 1.0, 'Survial Status for Patient On feature Nodes')
```

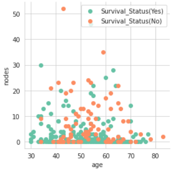


This Figure shows that Nearly 120 no of patient Suvived More than Five Year has no positve auxillary nodes

In [48]:
```
sns.set_style("whitegrid");
plt.figure(figsize= (10,15))
sns.FacetGrid(df, hue="status", size=4,palette='Set2') .map(plt.scatter, "age", "nodes")
plt.legend(labels , loc = 'upper right')
plt.show();
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:316: UserWarning: The `size` param
eter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```

```
<Figure size 720x1080 with 0 Axes>
```



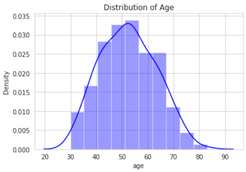
In [53]:

```
#Let's See the Distribution of the Age

sns.distplot(df["age"],bins = 10 ,color = 'blue')
plt.title('Distribution of Age')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplo
t` is a deprecated function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or `histplot` (an a
xes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[53]:

```
Text(0.5, 1.0, 'Distribution of Age')
```



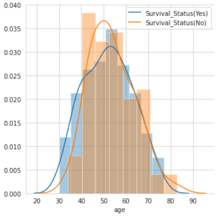
In [59]:

```
def distplot(data,column):
  sns.FacetGrid(data, hue="status", size=5) \
      .map(sns.distplot, column)
  plt.legend(labels, loc = 'upper right')
  plt.show();
```

In [60]:

```
#distibution of Age with o/p feature
distplot(df,'age')
```

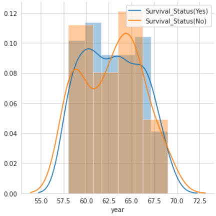
```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:316: UserWarning: The `size` param
eter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplo
t` is a deprecated function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or `histplot` (an a
xes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplo
t` is a deprecated function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or `histplot` (an a
xes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [62]:

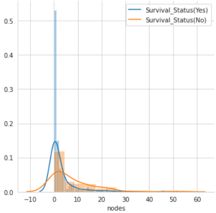
```
#distribution of year with survival status
distplot(df,'year')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:316: UserWarning: The `size` param
eter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplo
t` is a deprecated function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or `histplot` (an a
xes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplo
t` is a deprecated function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or `histplot` (an a
xes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



```
In [63]: distplot(df,'nodes')
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:316: UserWarning: The `size` param
eter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplo
t` is a deprecated function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or `histplot` (an a
xes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplo
t` is a deprecated function and will be removed in a future version. Please adapt your code t
o use either `displot` (a figure-level function with similar flexibility) or `histplot` (an a
xes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



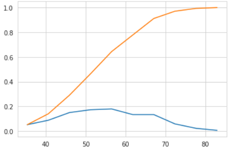
From the univariate distribution, we can see that Positive_Axillary_Nodes is important feature that determines our dependent variable Status. The second important feature is Age. Observation: The number of positive lymph nodes of the survivors is highly densed from 0 to 5.

Probability Density Function and Cummulative Density Function

```
In [69]: def pdfAndCdf(data,column):
           counts, bin_edges = np.histogram(data[column] , bins = 10 ,density = True)
           print(counts)
           pdf = counts/(sum(counts))
           print(pdf);
           print(bin_edges);
           #compute the cummulative sum
           cdf = np.cumsum(pdf)
           plt.plot(bin_edges[1:],pdf);
           plt.plot(bin_edges[1:], cdf)
           plt.show()
```

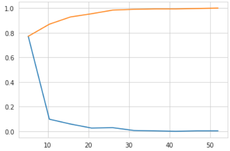
```
In [70]: pdfAndCdf(df,'age')
```

```
[0.00986558 0.01664817 0.02836355 0.03267974 0.03391294 0.02528055
 0.02528055 0.01109878 0.00431619 0.0012332 ]
[0.05228758 0.08823529 0.1503268  0.17320261 0.17973856 0.13398693
 0.13398693 0.05882353 0.02287582 0.00653595]
[30.  35.3 40.6 45.9 51.2 56.5 61.8 67.1 72.4 77.7 83. ]
```



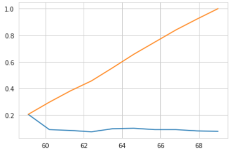
```
In [71]: pdfAndCdf(df,'nodes')
```

```
[0.14831574 0.0188537  0.01131222 0.00502765 0.00565611 0.00125691
 0.00062846 0.         0.00062846 0.00062846]
[0.77124183 0.09803922 0.05882353 0.02614379 0.02941176 0.00653595
 0.00326797 0.         0.00326797 0.00326797]
[ 0.   5.2 10.4 15.6 20.8 26.  31.2 36.4 41.6 46.8 52. ]
```



In [72]:
```python
pdfAndCdf(df,'year')
```

```
[0.18716578 0.08318479 0.07724302 0.06833036 0.08912656 0.09209745
 0.08318479 0.08318479 0.07427213 0.07130125]
[0.20588235 0.09150327 0.08496732 0.0751634  0.09803922 0.10130719
 0.09150327 0.09150327 0.08169935 0.07843137]
[58.  59.1 60.2 61.3 62.4 63.5 64.6 65.7 66.8 67.9 69. ]
```

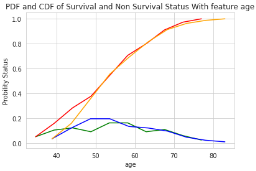


From the above plot Blue line Indicates PDF and Orange Line Indicates CDF we can see that almost 80% of the patients have positive lymph nodes less than 10.

In [92]:
```python
#plot Probability Density Function and CDF with Class Label

def PDFandCDFWithClassLabel(data,column):

  #survived class label data
  survived = data.loc[data['status'] == 1 , :]
  #non-survived
  non_survived = data.loc[data['status'] == 2, : ]


  #plot for suvived staus
  count, bin_edges = np.histogram(survived[column] , bins = 10, density = True)

  pdf = count/sum(count)

  print("PDF of Survival Status with feature {1} is : {0}".format(pdf , column))

  print("=" * 100)

  #comput cdf

  cdf = np.cumsum(pdf)
  print("Cumulativ Sum for Survived People with feature {1} : {0}".format(cdf,column))

  plt.plot(bin_edges[1:],pdf,color ='green')
  plt.plot(bin_edges[1:] , cdf,color='red')

  print("=" * 100)

  count, bin_edges = np.histogram(non_survived[column] , bins = 10, density = True)

  pdf = count/sum(count)

  print("PDFfor non-survival People with feature {1}  : {0}".format(pdf,column))

  #comput cdf
  print("=" * 100)

  cdf = np.cumsum(pdf)
  print("Cumulativ Sum for Non Survived People with feature {1}: {0}".format(cdf,column))

  print("=" * 100)
  plt.plot(bin_edges[1:],pdf,color='blue')
  plt.plot(bin_edges[1:] , cdf,color='orange')

  title = "PDF and CDF of Survival and Non Survival Status With feature {0}".format(column)


  plt.xlabel(column)
  plt.ylabel("Probility Status")
  plt.title(title)

  plt.show()
```

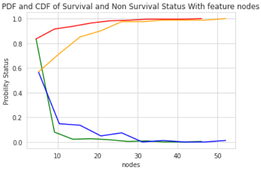
In [93]:
```python
PDFandCDFWithClassLabel(df,'age')
```

```
PDF of Survival Status with feature age is : [0.05333333 0.10666667 0.12444444 0.09333333 0.1
6444444 0.16444444
 0.09333333 0.11111111 0.06222222 0.02666667]
=============================================================================================
=======
Cumulativ Sum for Survived People with feature age : [0.05333333 0.16       0.28444444 0.3777
7778 0.54222222 0.70666667
 0.8        0.91111111 0.97333333 1.        ]
=============================================================================================
=======
PDFfor non-survival People with feature age  : [0.03703704 0.12345679 0.19753086 0.19753086
0.13580247 0.12345679
 0.09876543 0.04938272 0.02469136 0.01234568]
=============================================================================================
=======
Cumulativ Sum for Non Survived People with feature age: [0.03703704 0.16049383 0.35802469 0.5
5555556 0.69135802 0.81481481
 0.91358025 0.96296296 0.98765432 1.        ]
=============================================================================================
=======
```



```
In [94]: PDFandCDFWithClassLabel(df,'nodes')
```

```
PDF of Survival Status with feature nodes is : [0.83555556 0.08       0.02222222 0.02666667
0.01777778 0.00444444
 0.00888889 0.         0.         0.00444444]
=============================================================================================
=======
Cumulativ Sum for Survived People with feature nodes : [0.83555556 0.91555556 0.93777778 0.96
444444 0.98222222 0.98666667
 0.99555556 0.99555556 0.99555556 1.        ]
=============================================================================================
=======
PDFfor non-survival People with feature nodes  : [0.56790123 0.14814815 0.13580247 0.04938272
0.07407407 0.
 0.01234568 0.         0.         0.01234568]
=============================================================================================
=======
Cumulativ Sum for Non Survived People with feature nodes: [0.56790123 0.71604938 0.85185185
0.90123457 0.97530864 0.97530864
 0.98765432 0.98765432 0.98765432 1.        ]
=============================================================================================
=======
```

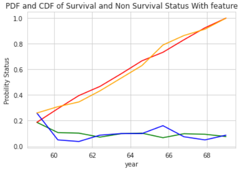


Observation

Almost 80% of the survived patients have less than or equal to 5 positive lymph nodes.so Postive Lymph Node is One of the Most Important Feature to Build a model

```
In [95]: PDFandCDFWithClassLabel(df,'year')
```

```
PDF of Survival Status with feature year is : [0.18666667 0.10666667 0.10222222 0.07111111 0.
09777778 0.10222222
 0.06666667 0.09777778 0.09333333 0.07555556]
=============================================================================================
=======
Cumulativ Sum for Survived People with feature year : [0.18666667 0.29333333 0.39555556 0.466
66667 0.56444444 0.66666667
 0.73333333 0.83111111 0.92444444 1.        ]
=============================================================================================
=======
PDFfor non-survival People with feature year  : [0.25925926 0.04938272 0.03703704 0.08641975
0.09876543 0.09876543
 0.16049383 0.07407407 0.04938272 0.08641975]
=============================================================================================
=======
Cumulativ Sum for Non Survived People with feature year: [0.25925926 0.30864198 0.34567901 0.
43209877 0.5308642  0.62962963
 0.79012346 0.86419753 0.91358025 1.        ]
=============================================================================================
=======
```



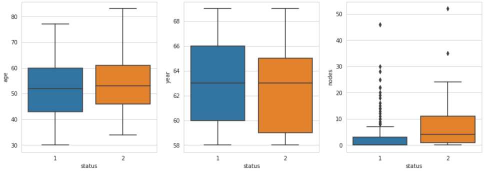
Observation

From the above plot Green line indicates the PDF of the Patient Suvived and Red line Indicates the CDF of the patient survived

Blue line indicates the PDF of the Patient whose not Suvived more than five years and orange line Indicates the CDF of the patient whose not survived more than five years

In [102]:

```
fig, axes = plt.subplots(1, 3, figsize=(15, 5))
for idx, feature in enumerate(list(df.columns)[:-1]):
    sns.boxplot( x='status', y=feature,
                    data=df, ax=axes[idx])
plt.show()
```

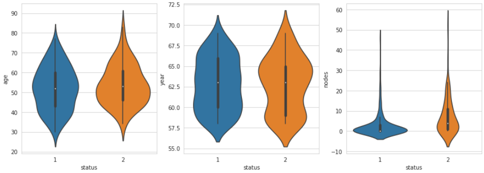


Box Plot is Important for univarient analysis,this Plot Shows more no fo people survived in 42- 52 age (25% of people survived more than five year in this range)

Violin Plot

In [103]:

```
fig, axes = plt.subplots(1, 3, figsize=(15, 5))
for idx, feature in enumerate(list(df.columns)[:-1]):
    sns.violinplot( x='status', y=feature,
                    data=df, ax=axes[idx])
plt.show()
```

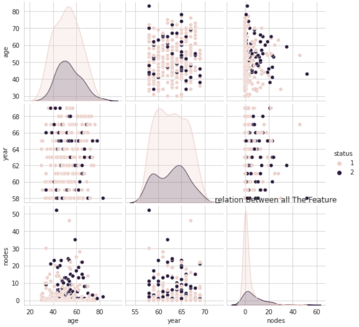


Observe the 3rd subplot says More no of people are survived only affected less than 5 positive lymph node ,as we saw the orange plot the patient whose not survied has affect more potive lymph node, as per understanding if the a patient affected more lymph postive node chance of survival is more than five is too low

Multivarient Analysis

In [97]:

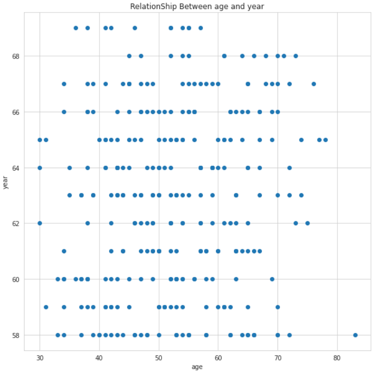
```
sns.pairplot(df,hue= 'status')
plt.title('relation Between all The Feature')
plt.show()
```



In [29]:

```
#Bivarient Analysis
#relationShip Between Age and Year
def bivarientAnalysisSctterPlot(data,column1,column2):
  #firstparam - dataset
  #secondparam - columnName
  #thirdparaa, - columnName
  plt.figure(figsize = (10,10))
  sns.set_style('whitegrid')
  title = "RelationShip Between {0} and {1}".format(column1 , column2)
  x_axis = column1
  y_axis = column2
  plt.title(title)
  plt.xlabel(x_axis)
  plt.ylabel(y_axis)
  plt.scatter(df[column1] ,df[column2])
  plt.show()
```

In [30]:

```
bivarientAnalysisSctterPlot(df, 'age' , 'year')
```