TC2-DS- Experiment 3

- **SIA VASHIST**
- PRN: 20190802107

Dataset - Haberman Cancer Survival Dataset.

LIBRARIES USED :

> PANDAS | MATPLOTLIB | NUMPY | STATSMODELS | NORM SciPy

• Question 1:

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. Find if dataset is having null values, then drop those values.
3. Check operation attribute is following normal distribution or not by drawing histogram and Q-Q plot.
4. Perform Transformation on attribute to better fit data into normal distribution. Draw histogram and Q-Q plot.
5. Write observations in English words.

In [2]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
from scipy.stats import norm
import pylab
```

In [3]:

```
haberman_df = pd.read_csv(r'C:\sia\haberman.csv')
haberman_df.columns=['Age','Operation_Year','axil_nodes','Surv_Status']
print("The Dataset is as Follows:")
print(haberman_df.dropna(), '\n')
```

```
The Dataset is as Follows:
     Age  Operation_Year  axil_nodes  Surv_Status
0     30              62           3            1
1     30              65           0            1
2     31              59           2            1
3     31              65           4            1
4     33              58          10            1
..   ...             ...         ...          ...
300   75              62           1            1
301   76              67           0            1
302   77              65           3            1
303   78              65           1            2
304   83              58           2            2

[305 rows x 4 columns]
```

In [8]:

```
#Descriptive Statistics
haberman_df.describe()
```

Out[8]:

	Age	Operation_Year	axil_nodes	Surv_Status
count	305.000000	305.000000	305.000000	305.000000
mean	52.531148	0.070075	4.036066	1.265574
std	10.744024	0.924473	7.199370	0.442364
min	30.000000	-3.191165	0.000000	1.000000
25%	44.000000	-0.467956	0.000000	1.000000
50%	52.000000	0.073508	1.000000	1.000000
75%	61.000000	0.692573	4.000000	2.000000
max	83.000000	2.625563	52.000000	2.000000

In [9]:

```
#2. Checking for Null Values
haberman_df.isnull()
```

Out[9]:

	Age	Operation_Year	axil_nodes	Surv_Status
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False
...	...	...	...	...
300	False	False	False	False
301	False	False	False	False
302	False	False	False	False
303	False	False	False	False
304	False	False	False	False

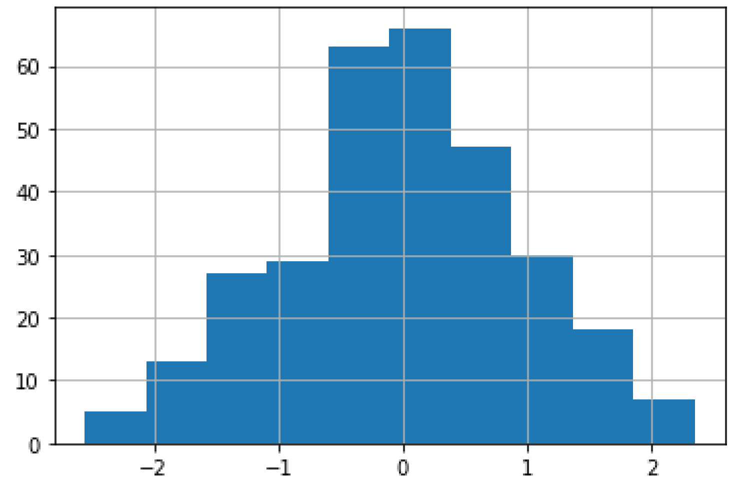
305 rows × 4 columns

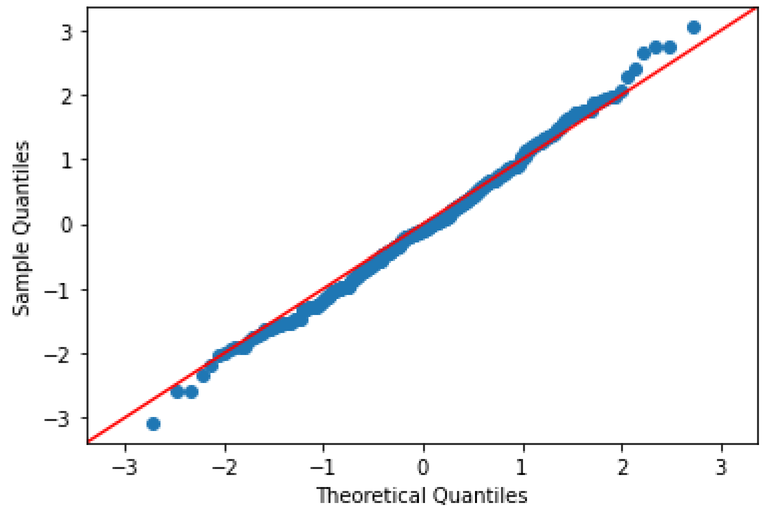
> - The dataframe does not have any null values.

In [22]:

```
# 3. Check operation attribute is following normal distribution or not by drawing histogram and Q-Q plot.
#Histogram
haberman_df.Operation_Year.hist()

#QQ Plot
haberman_df.Operation_Year = norm.rvs(size=305)
sm.qqplot(haberman_df.Operation_Year ,line='45')
pylab.show()
```





Observation :

> We can see that the plot's points are closer to the 45-degree line because it is following the normal distribution, hence there is no need to transform the given plot.

• Question 2:

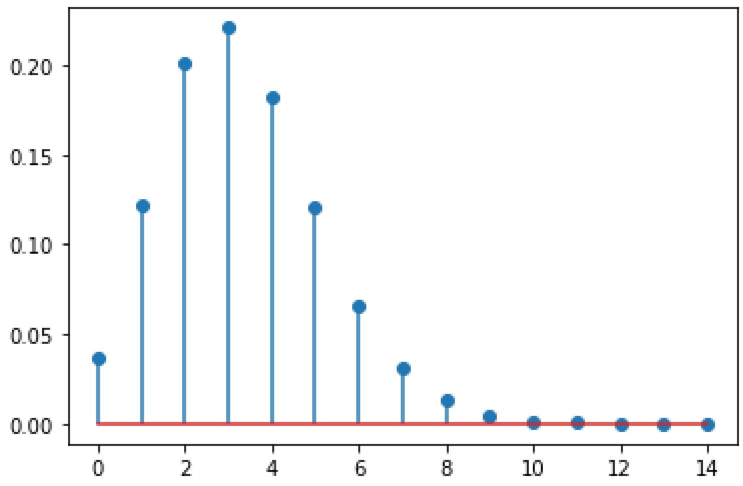
Imagine you have a machine learning model deployed in the cloud and receiving requests from your customers in real-time. How much cloud resources do you need to pay for in order to be 99% sure you can serve all the traffic that arrives at the model in any one-minute period? (Note: 3.3 requests on average based on your traffic data). Draw the distribution using python.

In [17]:

```
from scipy.stats import poisson

rate = 3.3
probs =[poisson.pmf(i,rate) for i in range(15)]
plt.stem(list(range(15)),probs)
```

```
Out[17]: <StemContainer object of 3 artists>
```



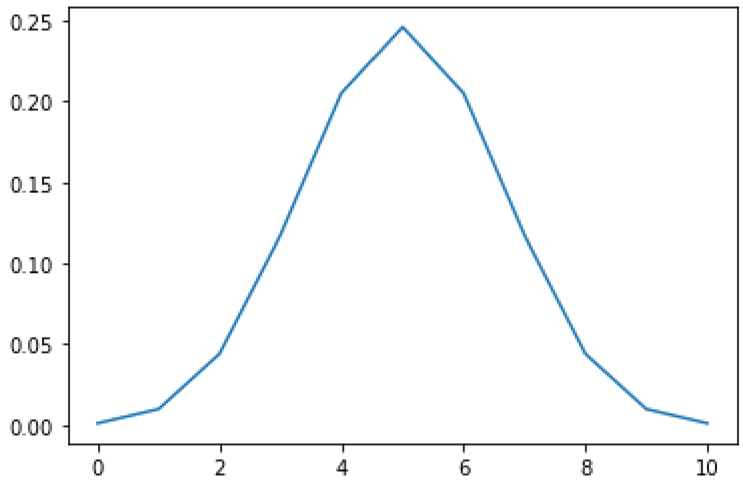
• Question 3:

What is the probability of observing different numbers of heads in 10 tosses with a fair coin? Find which distribution will get apply and plot it. Write a python script to draw the distribution. Hints: Binomial Distribution.

```
In [20]: from scipy.stats import binom

number = 10
head = .5
prob = [binom.pmf(i,number,head) for i in range(11)]
plt.plot(list(range(11)),prob)
```

```
Out[20]: [<matplotlib.lines.Line2D at 0x238ac743610>]
```



Conclusion:

So, using Python, we learned about several distributions including normal, poisson, and binomial. As a result, we successfully displayed the normal distribution and used histogram & QQ plots to analyse the distribution of the haberman dataset.