Multivariate Analysis

Name: Manoj Roy ID: 20216039

# Example M02 Consider the following data set to Find sample mean vector and sample variance and covariance matrix. Also plot the data (Scatter Plot Matrix).

In [1]:

```
# Sol02:
import numpy as np
import seaborn as sns
import pandas as pd
```

In [2]:

```
X1 = [7.5, 5.8, 4.5, 3.7, 3.9, 5.7, 5.6, 5.9, 7.0, 6.3,3.3,3.3,
     6.2,5.1,6.9,5.8,4.3,5.9,6.9,6.5]
X2 = [8.6, 6.8, 7.4, 5.5, 5.6, 6.5, 6.9, 6.8, 8.2, 6.3,4.4,3.6, 6.4,5.1,6.8,8.0,5.1,5.8,8
.4,7.0]
X3 = [5.9,7.1,5.9,7.8,9.1,6.8,7.4,7.4,8.5,10.2,5.8,4.3,8.9,8.0,9.1,8.4,6.0,9.0,6.8,6.4]

data = np.array([X1, X2, X3])
```

In [3]:

```
avg= np.mean(data, axis=1)
```

In [4]:

```
cov=np.cov(data, bias=True)
cov
```

Out[4]:

```
array([[1.546475, 1.3052  , 0.6523  ],
       [1.3052  , 1.6854  , 0.3696  ],
       [0.6523  , 0.3696  , 2.0784  ]])
```

In [5]:

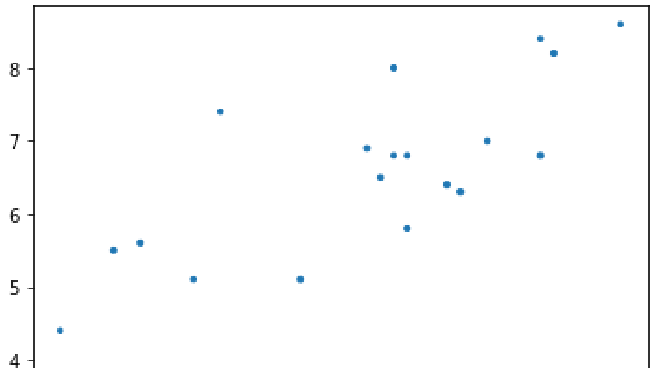
```
corr=np.corrcoef(data)
corr
```

Out[5]:

```
array([[1.        , 0.80845177, 0.36384077],
       [0.80845177, 1.        , 0.19747672],
       [0.36384077, 0.19747672, 1.        ]])
```

In [6]:

```
import matplotlib.pyplot as plt
plt.scatter(X1, X2, X3)
plt.show()
```



Example M03 Construct a Q□Q plot for the variable X1. Verify that the variable is normally distributed or not.

In [7]:

```
# Sol03:
```

In [8]:

```
import seaborn as sns
import statsmodels.api as sm
import scipy.stats as stats
```

In [9]:

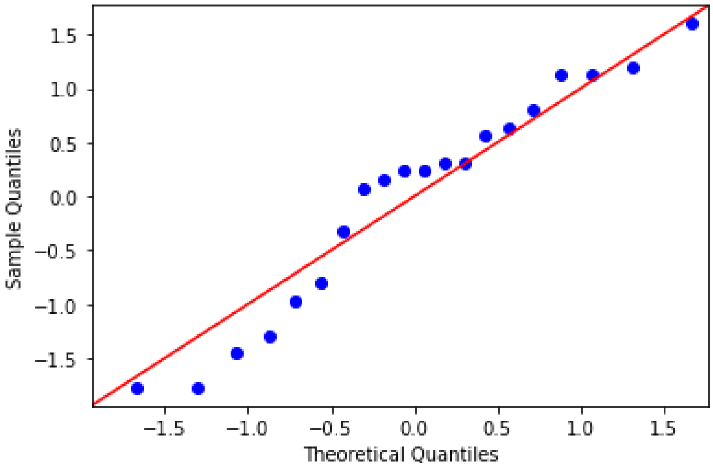
```
X1 = [3.3,3.3,3.7,3.9,4.3,4.5,5.1,5.6,5.7,5.8,5.8,5.9,5.9,6.2,6.3,6.5,6.9,6.9,7.0,7.5]
np_normal = pd.Series(X1)
```

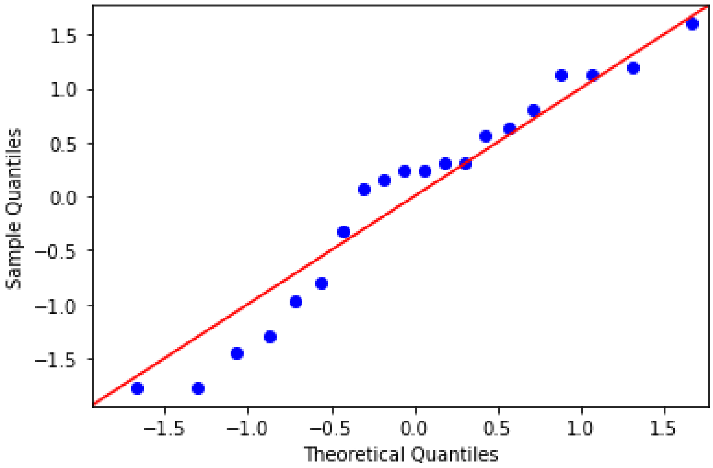
In [10]:

```
sm.qqplot(np_normal,line='45',fit=True,dist=stats.norm)
```

```
C:\Users\manoj\anaconda3\envs\juds\lib\site-packages\statsmodels\graphics\gofplots.py:993
: UserWarning: marker is redundantly defined by the 'marker' keyword argument and the fmt
string "bo" (-> marker='o'). The keyword argument will take precedence.
  ax.plot(x, y, fmt, **plot_style)
```

Out[10]:





Example M04 Verify that the variable X1 is normally distributed or not.

In [11]:

```
# Solution 04: The Jarque-Bera test tests the Hypothesis-
# H0 : Data is normal
# H1 : Data is NOT normal
```

In [12]:

```
X1 = [7.5, 5.8, 4.5, 3.7, 3.9, 5.7, 5.6, 5.9, 7.0, 6.3,3.3,3.3,
      6.2,5.1,6.9,5.8,4.3,5.9,6.9,6.5]
```

In [13]:

```
jarque_bera_test = stats.jarque_bera(X1)
```

In [14]:

```
jarque_bera_test
```

Out[14]:

```
Jarque_beraResult(statistic=1.3077364178257453, pvalue=0.5200302952992422)
```

As the p value is not less than 0.05, I do not reject the null hypothesis. So the data is not normally distributed.