In [16]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [17]:
```
# Importing the csv file
data1 = pd.read_csv('DataA.csv',index_col = 0)
```

In [18]:
```
# Checking the first ten data frames
data1.head(10)
```

Out[18]:

	fea.1	fea.2	fea.3	fea.4	fea.5	fea.6	fea.7	fea.8	fea.9	fea.10	...	fea.72	fea.73	fea.74	fea.75	fea.76	fea.77	fea.78	fea.
1	-153.0	414.0	939.0	-161.0	1007.0	99.0	-210.0	948.0	333.0	-19.0	...	655.0	-316.0	-302.0	-617.0	-955.0	-264.0	23.0	-25
2	-150.0	420.0	939.0	-177.0	1008.0	103.0	-207.0	939.0	316.0	9.0	...	655.0	-309.0	-304.0	-619.0	-955.0	-265.0	19.0	-31
3	-160.0	432.0	941.0	-162.0	982.0	98.0	-198.0	936.0	315.0	-10.0	...	655.0	-302.0	-308.0	-621.0	-966.0	-270.0	10.0	-38
4	-171.0	432.0	911.0	-174.0	999.0	115.0	-187.0	918.0	338.0	34.0	...	655.0	-293.0	-312.0	-622.0	-964.0	-269.0	14.0	-51
5	-171.0	NaN	929.0	-189.0	1004.0	104.0	-198.0	939.0	350.0	60.0	...	655.0	-284.0	-318.0	-624.0	-966.0	-262.0	24.0	-4(
6	-171.0	432.0	924.0	-179.0	1011.0	85.0	-204.0	945.0	336.0	94.0	...	655.0	-274.0	-323.0	-626.0	-969.0	-267.0	27.0	-36
7	-169.0	429.0	949.0	-175.0	1007.0	102.0	-188.0	914.0	322.0	154.0	...	655.0	-263.0	-331.0	-627.0	-975.0	-273.0	17.0	-27
8	-160.0	423.0	927.0	-195.0	996.0	123.0	-213.0	925.0	302.0	128.0	...	655.0	-251.0	-337.0	-628.0	-955.0	-275.0	8.0	-4(
9	-163.0	432.0	929.0	-178.0	994.0	101.0	-186.0	946.0	296.0	166.0	...	654.0	-239.0	-343.0	-630.0	-967.0	-267.0	15.0	-34
10	-156.0	415.0	936.0	-186.0	1014.0	111.0	-195.0	960.0	280.0	202.0	...	653.0	-228.0	-351.0	-631.0	-964.0	-264.0	7.0	-2(

10 rows × 81 columns

In [19]:
```
# Removing the columns with null data and the first column 'un-named'
Data2 = data1.drop(['fea.34','fea.35','fea.36'], axis = 1)
```

In [20]:
```
Data2.columns
```

Out[20]:
```
Index(['fea.1', 'fea.2', 'fea.3', 'fea.4', 'fea.5', 'fea.6', 'fea.7', 'fea.8',
       'fea.9', 'fea.10', 'fea.11', 'fea.12', 'fea.13', 'fea.14', 'fea.15',
       'fea.16', 'fea.17', 'fea.18', 'fea.19', 'fea.20', 'fea.21', 'fea.22',
       'fea.23', 'fea.24', 'fea.25', 'fea.26', 'fea.27', 'fea.28', 'fea.29',
       'fea.30', 'fea.31', 'fea.32', 'fea.33', 'fea.37', 'fea.38', 'fea.39',
       'fea.40', 'fea.41', 'fea.42', 'fea.43', 'fea.44', 'fea.45', 'fea.46',
       'fea.47', 'fea.48', 'fea.49', 'fea.50', 'fea.51', 'fea.52', 'fea.53',
       'fea.54', 'fea.55', 'fea.56', 'fea.57', 'fea.58', 'fea.59', 'fea.60',
       'fea.61', 'fea.62', 'fea.63', 'fea.64', 'fea.65', 'fea.66', 'fea.67',
       'fea.68', 'fea.69', 'fea.70', 'fea.71', 'fea.72', 'fea.73', 'fea.74',
       'fea.75', 'fea.76', 'fea.77', 'fea.78', 'fea.79', 'fea.80', 'fea.81'],
      dtype='object')
```

In [21]:
```
# Removing any row null values
Data34 = Data2.dropna(how ="any")
```

In [22]:
```
Data34.columns
```

Out[22]:
```
Index(['fea.1', 'fea.2', 'fea.3', 'fea.4', 'fea.5', 'fea.6', 'fea.7', 'fea.8',
       'fea.9', 'fea.10', 'fea.11', 'fea.12', 'fea.13', 'fea.14', 'fea.15',
       'fea.16', 'fea.17', 'fea.18', 'fea.19', 'fea.20', 'fea.21', 'fea.22',
       'fea.23', 'fea.24', 'fea.25', 'fea.26', 'fea.27', 'fea.28', 'fea.29',
       'fea.30', 'fea.31', 'fea.32', 'fea.33', 'fea.37', 'fea.38', 'fea.39',
       'fea.40', 'fea.41', 'fea.42', 'fea.43', 'fea.44', 'fea.45', 'fea.46',
       'fea.47', 'fea.48', 'fea.49', 'fea.50', 'fea.51', 'fea.52', 'fea.53',
       'fea.54', 'fea.55', 'fea.56', 'fea.57', 'fea.58', 'fea.59', 'fea.60',
       'fea.61', 'fea.62', 'fea.63', 'fea.64', 'fea.65', 'fea.66', 'fea.67',
       'fea.68', 'fea.69', 'fea.70', 'fea.71', 'fea.72', 'fea.73', 'fea.74',
       'fea.75', 'fea.76', 'fea.77', 'fea.78', 'fea.79', 'fea.80', 'fea.81'],
      dtype='object')
```

In [23]:
```
np.var(Data34)
```

Out[23]:
```
fea.1      80393.181431
fea.2     141083.802180
fea.3     154445.965435
fea.4      33373.439156
fea.5      37532.008068
                  ...
fea.77    138033.789214
fea.78     62181.519916
fea.79    559456.794735
fea.80    178130.212049
fea.81    195901.505454
Length: 78, dtype: float64
```

In [24]:
```
# Performing min-max normalization
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(Data34)
New_data = scaler.transform(Data34)
```

In [25]:
```
New_data
```

Out[25]:
```
array([[0.52973721, 0.3747785 , 0.61027837, ..., 0.48753915, 0.26054952,
        0.34659603],
       [0.53042877, 0.3765505 , 0.61027837, ..., 0.48740297, 0.26149106,
        0.3442290 4],
       [0.52812356, 0.38009451, 0.6106677 , ..., 0.48692632, 0.25918001,
        0.34389089],
       ...,
       [0.56523744, 0.54548139, 0.46661476, ..., 0.48726678, 0.26534281,
        0.34039675],
       [0.56477639, 0.58741878, 0.47459607, ..., 0.49536974, 0.26200462,
        0.34377818],
       [0.56500692, 0.25251034, 0.42748686, ..., 0.49931908, 0.262347  ,
        0.34524346]])
```

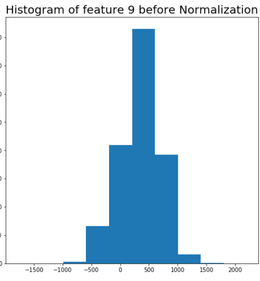
In [26]:
```
# Performing Z-score Normalization
New_dataZscore = (New_data - New_data.mean()) / New_data.std()
```

In [27]:
```
New_dataZscore
```

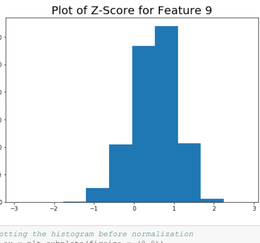
Out[27]:
```
array([[ 0.12159763, -0.76834632,  0.58415391, ..., -0.12075018,
         -1.42437523, -0.93020115],
       [ 0.12556934, -0.75816952,  0.58415391, ..., -0.12153229,
         -1.41896787, -0.94375506],
       [ 0.11233028, -0.73781594,  0.58638988, ..., -0.12426971,
         -1.43224047, -0.94573705],
       ...,
       [ 0.32547914,  0.21201813, -0.24092116, ..., -0.12231441,
         -1.39684687, -0.96580425],
       [ 0.32283132,  0.45286891, -0.19508366, ..., -0.0757783 ,
         -1.41601841, -0.94638438],
       [ 0.32415523, -1.47054507, -0.46563673, ..., -0.05309683,
         -1.41405209, -0.9379691 ]])
```

In [28]:
```
# Plotting the histogram before normalization
fig,ax = plt.subplots(figsize = (8,8))
plt.hist(data1.iloc[:,9])
ax.set_title('Histogram of feature 9 before Normalization', fontsize=20)
plt.show()
```

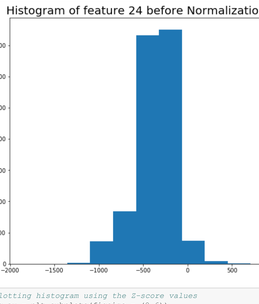
```
/opt/anaconda3/lib/python3.7/site-packages/numpy/lib/histograms.py:839: RuntimeWarning: invalid value encountered in greater_equal
  keep = (tmp_a >= first_edge)
/opt/anaconda3/lib/python3.7/site-packages/numpy/lib/histograms.py:840: RuntimeWarning: invalid value encountered in less_equal
  keep &= (tmp_a <= last_edge)
```



In [29]:
```
#Plotting histogram using the Z-score values
fig,ax=plt.subplots(figsize=(8,6))
plt.hist(New_dataZscore[:,9])
ax.set_title('Plot of Z-Score for Feature 9', fontsize=20)
plt.show()
```



In [30]:
```
#Plotting the histogram before normalization
fig,ax = plt.subplots(figsize = (8,8))
plt.hist(data1.iloc[:,24])
ax.set_title('Histogram of feature 24 before Normalization', fontsize=20)
plt.show()
```



In [31]:
```
#Plotting histogram using the Z-score values
fig,ax = plt.subplots(figsize = (8,6))
plt.hist(New_dataZscore[:,24])
ax.set_title('Plot of Z-Score for Feature 24', fontsize=20)
plt.show()
```