OM&A Productivity Outlier Analysis

```
# load data
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import shapiro, probplot, norm

np.random.seed(42)  # for reproducibility

#OM&A Productivity data from Table 2 in Exhibit CLP-PBR-2
data = np.array([
    1.99, -5.79, 1.06, -1.52, -0.54, -1.72, 0.56, 4.63, 0.02,
    3.89, -0.79, -0.17, 2.76, -0.49, -12.72, 7.21, -4.83, 1.84, 4.04
])


# check for outliers via interquartile range
Q1, Q3 = np.percentile(data, [25, 75])
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers = data[(data < lower_bound) | (data > upper_bound)]

print("\033[1m Outliers: \033[0m", outliers)

```

```
Outliers:  [-12.72]
```

```
#get data without 2020 (outlier)
data_less_2020 = np.array([1.99, -5.79, 1.06, -1.52, -0.54,
                           -1.72, 0.56, 4.63, 0.02, 3.89, -0.79,
                           -0.17, 2.76, -0.49, 7.21, -4.83, 1.84, 4.04])

#average productivity factor without 2020 (outlier)
print("Productivity factor with outlier removed: ",
      np.round(np.mean(data_less_2020),3))

#average productivity factor with outlier included
print("Productivity factor with all data (proposed by Eversource): ",
      np.round(np.mean(data),3))
```

```
Productivity factor with outlier removed:  0.675
Productivity factor with all data (proposed by Eversource):  -0.03
```

TFP Outlier Analysis

```
# load data
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import shapiro, probplot, norm

np.random.seed(42)  # for reproducibility

#Data from Table 2 in
data = np.array([0.75, -3.62, -0.21, -2.15, -0.96,
                -1.92, -0.93, 1.15, -1.43, 0.37, -2.09,
                -1.82, -0.53, -2.12, -8.14,1.94, -3.09, -1.39, -0.67])

# check for outliers via interquartile range
Q1, Q3 = np.percentile(data, [25, 75])
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers = data[(data < lower_bound) | (data > upper_bound)]

print("\033[1m Outliers: \033[0m", outliers)
```

```
23
24
25
```

Outliers: [-8.14]

```
1 #get data without 2020 (outlier)
2 data_less_2020_tfp = np.array([0.75, -3.62, -0.21, -2.15, -0.96,
3                                -1.92, -0.93, 1.15, -1.43, 0.37, -2.09,
4                                -1.82, -0.53, -2.12, 1.94, -3.09, -1.39, -0.67])
5
6 #average productivity factor without 2020 (outlier)
7 print("Productivity factor with outlier removed: ",
8       np.round(np.mean(data_less_2020_tfp),3))
9
10 #average productivity factor with outlier included
11 print("Productivity factor with all data (proposed by Eversource): ",
12       np.round(np.mean(data),3))
```

```
Productivity factor with outlier removed:  -1.04
Productivity factor with all data (proposed by Eversource):  -1.414
```