In [72]:
```python
# Import Library
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
from scipy import stats
```

In [90]:
```python
data = pd.read_csv(r'C:\Users\HP\Desktop\Advance Data Analyst\4. The Power of Stats\2. Module 2\5. Prob distribution with python\Files\modified_c4_epa_air_quality.csv')
data.head(10)
```

Out[90]:

	date_local	state_name	county_name	city_name	local_site_name	parameter_name	units_of_measure	aqi_log
0	2018-01-01	Arizona	Maricopa	Buckeye	BUCKEYE	Carbon monoxide	Parts per million	2.079442
1	2018-01-01	Ohio	Belmont	Shadyside	Shadyside	Carbon monoxide	Parts per million	1.791759
2	2018-01-01	Wyoming	Teton	Not in a city	Yellowstone National Park - Old Faithful Snow ...	Carbon monoxide	Parts per million	1.098612
3	2018-01-01	Pennsylvania	Philadelphia	Philadelphia	North East Waste (NEW)	Carbon monoxide	Parts per million	1.386294
4	2018-01-01	Iowa	Polk	Des Moines	CARPENTER	Carbon monoxide	Parts per million	1.386294
5	2018-01-01	Hawaii	Honolulu	Not in a city	Kapolei	Carbon monoxide	Parts per million	2.708050
6	2018-01-01	Hawaii	Honolulu	Not in a city	Kapolei	Carbon monoxide	Parts per million	1.098612
7	2018-01-01	Pennsylvania	Erie	Erie	NaN	Carbon monoxide	Parts per million	1.098612
8	2018-01-01	Hawaii	Honolulu	Honolulu	Honolulu	Carbon monoxide	Parts per million	1.791759
9	2018-01-01	Colorado	Larimer	Fort Collins	Fort Collins - CSU - S. Mason	Carbon monoxide	Parts per million	1.945910

In [92]:
```python
# Get descriptive stats.

data.describe()
```

Out[92]:

	aqi_log
count	260.000000
mean	1.766921
std	0.714716
min	0.000000
25%	1.098612
50%	1.791759
75%	2.302585
max	3.931826

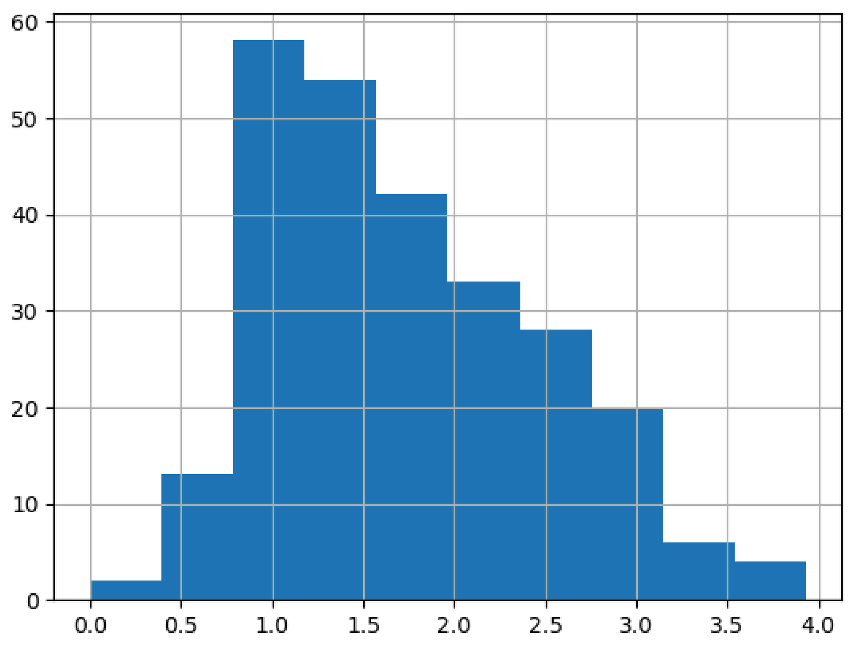
In [94]:
```python
# Get descriptive stats about the states in the data.

data["state_name"].describe()
```

Out[94]:
```
count            260
unique            52
top       California
freq              66
Name: state_name, dtype: object
```

In [96]:
```python
# Create a histogram to visualize distribution of aqi_log.

data["aqi_log"].hist();
```



In [98]:
```python
# Define variable for aqi_log mean.
mean_aqi_log = data["aqi_log"].mean()


# Print out the mean.
print(mean_aqi_log)
```
```
1.7669210929985582
```

In [100…
```python
# Define variable for aqi_log standard deviation.

std_aqi_log = data["aqi_log"].std()

# Print out the standard deviation.

print(std_aqi_log)
```
```
0.7147155520223721
```

In [102…
```python
# Define variable for lower limit, 1 standard deviation below the mean.

lower_limit = mean_aqi_log - 1 * std_aqi_log


# Define variable for upper limit, 1 standard deviation above the mean.

upper_limit = mean_aqi_log + 1 * std_aqi_log


# Display lower_limit, upper_limit.

print(lower_limit, upper_limit)
```
```
1.052205540976186 2.4816366450209304
```

In [104…
```python
# Display the actual percentage of data that falls within 1 standard deviation of the mean.

((data["aqi_log"] >= lower_limit) & (data["aqi_log"] <= upper_limit)).mean() * 100
```

Out[104… 76.15384615384615

In [106…
```python
# Define variable for lower limit, 2 standard deviations below the mean.

lower_limit = mean_aqi_log - 2 * std_aqi_log


# Define variable for upper limit, 2 standard deviations below the mean.

upper_limit = mean_aqi_log + 2 * std_aqi_log


# Display lower_limit, upper_limit.

print(lower_limit, upper_limit)
```
```
0.3374899889538139 3.1963521970433026
```

In [108…
```python
# Display the actual percentage of data that falls within 2 standard deviations of the mean.

((data["aqi_log"] >= lower_limit) & (data["aqi_log"] <= upper_limit)).mean() * 100
```

Out[108… 95.76923076923077

In [110…
```python
# Define variable for lower limit, 3 standard deviations below the mean.

lower_limit = mean_aqi_log - 3 * std_aqi_log


# Define variable for upper limit, 3 standard deviations above the mean.

upper_limit = mean_aqi_log + 3 * std_aqi_log


# Display lower_limit, upper_limit.

print(lower_limit, upper_limit)
```
```
-0.37722556306855815 3.9110677490656744
```

In [112…
```python
# Display the actual percentage of data that falls within 3 standard deviations of the mean.

((data["aqi_log"] >= lower_limit) & (data["aqi_log"] <= upper_limit)).mean() * 100
```

Out[112… 99.61538461538461

In [114…
```python
# Compute the z-score for every aqi_log value, and add a column named z_score in the data to store those results.

data["z_score"] = stats.zscore(data["aqi_log"], ddof=1) # ddof=degrees of freedom correction (sample vs. population)

# Display the first 5 rows to ensure that the new column was added.

data.head()
```

Out[114…

	date_local	state_name	county_name	city_name	local_site_name	parameter_name	units_of_measure	aqi_log	z_score
0	2018-01-01	Arizona	Maricopa	Buckeye	BUCKEYE	Carbon monoxide	Parts per million	2.079442	0.437265
1	2018-01-01	Ohio	Belmont	Shadyside	Shadyside	Carbon monoxide	Parts per million	1.791759	0.034753
2	2018-01-01	Wyoming	Teton	Not in a city	Yellowstone National Park - Old Faithful Snow ...	Carbon monoxide	Parts per million	1.098612	-0.935070
3	2018-01-01	Pennsylvania	Philadelphia	Philadelphia	North East Waste (NEW)	Carbon monoxide	Parts per million	1.386294	-0.532557
4	2018-01-01	Iowa	Polk	Des Moines	CARPENTER	Carbon monoxide	Parts per million	1.386294	-0.532557

In [116…
```python
# Display data where `aqi_log` is above or below 3 standard deviations of the mean

data[(data["z_score"] > 3) | (data["z_score"] < -3)]
```

Out[116…

	date_local	state_name	county_name	city_name	local_site_name	parameter_name	units_of_measure	aqi_log	z_score
244	2018-01-01	Arizona	Maricopa	Phoenix	WEST PHOENIX	Carbon monoxide	Parts per million	3.931826	3.029044

In [ ]:
```python
# some key takeaways that you learned

# Plotting the data using a histogram, then observing the shape, enables you to visually determine whether the data is normally distributed.
# The empirical rule can be used to verify whether a distribution is normal.
# The mean and standard deviation are important measures when applying the empirical rule to a distribution.
# Z-score allows you to identify potenial outliers in the data.

# What summary would you provide to stakeholders?
```

```
# The distribution of the aqi_log data is approximately normal.
# Using statistical methods, it was determined that the site at West Phoenix has worse air quality than the other sites.
# Consider allocating more resources toward further examining this site in order to improve its air quality.
```