Carbon monoxide concentration in united states regions

Data from the United States Environmental Protection Agency (EPA). The data includes information about more than 200 sites, identified by state, county, city, and local site names. One of the main goals is to determine which regions need support to make air quality improvements. Given that carbon monoxide is a major air pollutant, you will investigate data from the Air Quality Index (AQI) with respect to carbon monoxide.

In [16]:
```
#    import relevant libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from scipy import stats
```

In [17]:
```
#    load data into dataframe
df = pd.read_csv("startdata2.csv")
```

Exploiratory analysis (EDA)

In [18]:
```
#    the first few rows
df.head()
```

Out[18]:

	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

The aqi_log column represents air quality index(AQI) readings

In [19]:
```
df.shape
```

Out[19]:
```
(260, 8)
```

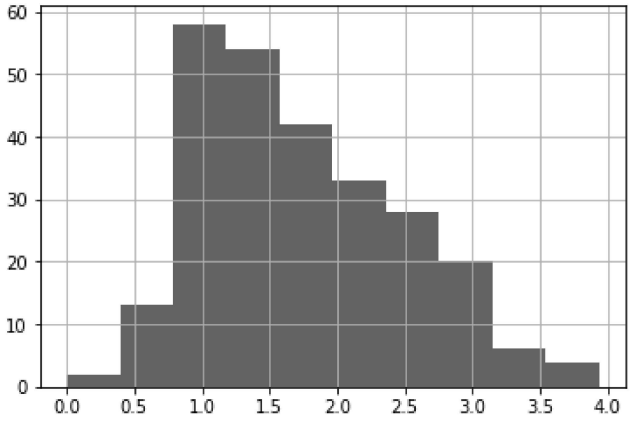
In [20]:
```
df.isnull().sum()   # missing value check
```

Out[20]:
```
date_local          0
state_name          0
county_name         0
city_name           0
local_site_name     3
parameter_name      0
units_of_measure    0
aqi_log             0
dtype: int64
```

In [21]:
```
df.duplicated().sum()  #   duplicates check
```

Out[21]:
```
0
```

In [22]:
```
#    let's consider the "aqi_log" column
df["aqi_log"].hist();
```



Statistical test

The empirical rule states that, for every normal distribution:

68% of the data fall within 1 standard deviation of the mean, 95% of the data fall within 2 standard deviations of the mean and 99.7% of the data fall within 3 standard deviations of the mean

In [24]:
```
mean_aqi_log = df["aqi_log"].mean()
std_aqi_log = df["aqi_log"].std()
```

In [25]:
```
#    Now, check the first part of the empirical rule: whether 68% of the aqi_log data falls
lower_limit = mean_aqi_log - 1 * std_aqi_log #lower limit
upper_limit = mean_aqi_log + 1 * std_aqi_log #upper limit
print(lower_limit, upper_limit)
```

```
1.0522055409761855 2.48163664502093
```

In [27]:
```
# Display the actual percentage of data that falls within 1 standard deviation of the mean


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

Out[27]:
```
76.15384615384615
```

In [29]:
```
#    Now, consider the second part of the empirical rule: whether 95% of the aqi_log data
lower_limit = mean_aqi_log - 2 * std_aqi_log # lower limit
upper_limit = mean_aqi_log + 2 * std_aqi_log # upper limit
print(lower_limit, upper_limit)
```

```
0.33748998895381344 3.1963521970433018
```

In [31]:
```
# Display the actual percentage of data that falls within 2 standard deviations of the mean

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

Out[31]:
```
95.76923076923077
```

In [32]:
```
#    Now, consider the third part of the empirical rule:whether 99.7% of the aqi_log data
lower_limit = mean_aqi_log - 3 * std_aqi_log
upper_limit = mean_aqi_log + 3 * std_aqi_log
print(lower_limit, upper_limit)
```

```
-0.3772255630685586 3.911067749065674
```

In [34]:
```
# Display the actual percentage of data that falls within 3 standard deviations of the mean

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

Out[34]:
```
99.61538461538461
```

Result and evaluation

Results obtained by applying the empirical rule

About 76.15% of the data falls within 1 standard deviation of the mean. About 95.77% of the data falls within 2 standard deviation of the mean. About 99.62% of the data falls within 3 standard deviations of the mean.

The data appears to be not exactly normal, but could be considered approximately normal.

In [36]:
```
#    Z-score could be used to identify values that lie more than 3 standard deviations bel
df["z_score"] = stats.zscore(df["aqi_log"]) #   z_score column
```

In [37]:
```
1 df.head(10) few first row
```

Out[37]:

	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 [38]:
```
1 # Display data where `aqi_log` is above or below 3 standard deviations of the mean
2
3 df[(df["z_score"] > 3) | (df["z_score"] < -3)]
```

Out[38]:

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

Findings

1. The aqi_log for West Phoenix is slightly above 3 standard deviations of the mean. This means that the air quality at that site is worse than the rest of the sites represented in the data.
2. The distribution of the aqi_log data is approximately normal.

In [ ]:
```
1
```