The study “Wastewater-based epidemiology: deriving a SARS-CoV-2 data validation method to assess data quality and to improve trend recognition” outlines a systematic approach for detecting outliers in SARS-CoV-2 wastewater data using quality control parameters (QCPs) such as flow, gene segment ratios, and other variables.

**Step 1: Prepare Your Data**

* Organize your data with columns including:
* SARS concentration measurements
* QCPs (e.g., flow, gene ratios, pH, etc.)
* Metadata like sampling date, treatment plant ID

import pandas as pd

# Load your data

data = pd.read\_csv('your\_data.csv')

# Example columns: ['date', 'WWTP', 'SARS\_concentration', 'flow', 'gene\_ratio1', 'gene\_ratio2', ...]

**Step 2: Calculate Outliers Using the Interquartile Range (IQR) Method**

For each QCP:

* Compute the first quartile (Q1), third quartile (Q3)
* Calculate IQR = Q3 - Q1
* Define outliers as points outside 1.5 \* IQR

def identify\_outliers(column):

q1 = column.quantile(0.25)

q3 = column.quantile(0.75)

iqr = q3 - q1

lower\_bound = q1 - 1.5 \* iqr

upper\_bound = q3 + 1.5 \* iqr

outlier\_mask = (column < lower\_bound) | (column > upper\_bound)

return outlier\_mask

# For example, applying to 'flow'

data['flow\_outlier'] = identify\_outliers(data['flow'])

**Step 3: Identify Outliers in Gene Ratios**

* Calculate all usable gene ratios (e.g., ratio of gene segment 1 to 2, 1 to 3, etc.)
* Flag ratios exceeding thresholds (e.g., >40% outliers among ratios)

# Example: calculate ratios

data['gene\_ratio1\_2'] = data['gene\_segment1'] / data['gene\_segment2']

data['gene\_ratio1\_3'] = data['gene\_segment1'] / data['gene\_segment3']

# Add more ratios as needed

# Detect outlier ratios

ratio\_columns = ['gene\_ratio1\_2', 'gene\_ratio1\_3']

for col in ratio\_columns:

q1 = data[col].quantile(0.25)

q3 = data[col].quantile(0.75)

iqr = q3 - q1

lower\_bound = q1 - 1.5 \* iqr

upper\_bound = q3 + 1.5 \* iqr

data[f'{col}\_outlier'] = (data[col] < lower\_bound) | (data[col] > upper\_bound)

# Count how many ratios are outliers per sample

data['ratio\_outlier\_count'] = data[[f'{col}\_outlier' for col in ratio\_columns]].sum(axis=1)

# Flag if more than 40% of ratios are outliers

threshold = 0.4

data['gene\_ratio\_outlier'] = data['ratio\_outlier\_count'] > (len(ratio\_columns) \* threshold)

**Step 4: Combine QCP Outliers**

* Combine all individual QCP outlier masks to identify overall outliers

# Example: combine flow and gene ratio outliers

data['combined\_outlier'] = data['flow\_outlier'] | data['gene\_ratio\_outlier']

**Step 5: Remove or Downweight Outlier Data Points**

* We can remove these data points or assign them lesser weight when calculating trend metrics

# Remove outliers

clean\_data = data[~data['combined\_outlier']]

# or, for weighted averaging:

weighted\_sars = (data['SARS\_concentration'] \* (~data['combined\_outlier']).astype(float)).sum() / (~data['combined\_outlier']).astype(float).sum()