Parametric Estimation and Distribution Fitting for Air Quality Index Data in Mexico City

Introduction

In this project, we will apply parametric estimation techniques to fit distributions to the air quality index (AQI) data for the Mexico City Metropolitan Area (CDMX). We will use two specific distributions: the Gamma distribution and the Rayleigh distribution. By deriving estimators analytically and applying them to real-world data, we aim to understand the suitability of these distributions for modeling AQI data. The process involves deriving estimators using both the Method of Moments (MoM) and Maximum Likelihood Estimation (MLE), implementing these estimators in Python, and evaluating the performance of the fitted models.

Exercise 1: Analytical Derivation

1. *Method of Moments Estimators*: Derive the estimators for the parameters of the Rayleigh and Gamma distributions using the Method of Moments.
   - **Rayleigh Distribution**:
     - Rayleigh distribution parameter: σ
     - Probability density function: $pdf(x;\sigma) = \frac{x}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right)$
   - **Gamma Distribution**:
     - Gamma distribution parameters: α (shape) and β (scale)
     - Probability density function: $pdf(x;\alpha,\beta) = \frac{x^{\alpha-1}\exp\left(-\frac{x}{\beta}\right)}{\Gamma(\alpha)\beta^\alpha}$

To derive the Method of Moments (MoM) estimators for the parameters of the Rayleigh and Gamma distributions, we start by equating n population moments to n sample moments. Where n is the number of parameter in each distribution

1. **Rayleigh Distribution**
   - Population Moments
     - Mean:

$$\mathbb{E}[X] = \sigma\sqrt{\frac{\pi}{2}}$$

   - Sample Moments
     - Sample Mean:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

   - *Method of Moments Estimator*

To find the MoM estimator for σ, we equate the population mean to the sample mean:

$$\mathbb{E}[X] = \bar{X} \implies \sigma\sqrt{\frac{\pi}{2}} = \bar{X} \implies \hat{\sigma}_{MoM} = \frac{\bar{X}}{\sqrt{\frac{\pi}{2}}}$$

Thus, the MoM estimator for σ is:

$$\hat{\sigma}_{MoM} = \frac{\bar{X}\sqrt{2}}{\sqrt{\pi}} \qquad (1)$$

2. **Gamma Distribution**
   - Population Moments
     - Mean:

$$\mathbb{E}[X] = \alpha\beta$$

     - Variance:

$$\mathrm{Var}(X) = \alpha\beta^2$$

   - Sample Moments
     - Sample Mean:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

     - Sample Variance:

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})^2$$

   - *Method of Moments Estimators*

To find the MoM estimators for α and β, we equate the population moments to the sample moments:

$$\mathbb{E}[X] = \alpha\beta \implies \bar{X} = \alpha\beta$$
$$\mathrm{Var}(X) = \alpha\beta^2 \implies S^2 = \alpha\beta^2$$

From the first equation:

$$\alpha\beta = \bar{X} \implies \alpha = \frac{\bar{X}}{\beta}$$

Substituting α in the second equation:

$$S^2 = \frac{\bar{X}}{\beta}\beta^2 \implies S^2 = \bar{X}\beta \implies \beta = \frac{S^2}{\bar{X}}$$

Now, substituting β back to find α:

$$\alpha = \frac{\bar{X}}{\beta} = \frac{\bar{X}}{\frac{S^2}{\bar{X}}} = \frac{\bar{X}^2}{S^2}$$

Thus, the MoM estimators for α and β are:

$$\hat{\alpha}_{MoM} = \frac{\bar{X}^2}{S^2} \qquad (2)$$
$$\hat{\beta}_{MoM} = \frac{S^2}{\bar{X}} \qquad (3)$$

2. *Maximum Likelihood Estimators*: Derive the estimators for the parameters of the Rayleigh and Gamma distributions using the Maximum Likelihood Estimation method. Simplify the resulting expressions as much as possible.
   - **Rayleigh Distribution**:
     - Derive the MLE for $sigma$.
   - **Gamma Distribution**:
     - Derive the MLE for α and β, noting the resulting transcendental equation.

To derive the Maximum Likelihood Estimators (MLE) for the parameters of the Rayleigh and Gamma distributions, we start by writing down the likelihood functions and then find the parameter values that maximize these functions.

1. **Rayleigh Distribution**

Given a sample $X = \{x_1, x_2, \ldots, x_M\}$, the likelihood function is:

$$L(\sigma; X) = \prod_{i=1}^{M}\frac{x_i}{\sigma^2}\exp\left(-\frac{x_i^2}{2\sigma^2}\right)$$
$$L(\sigma; X) = \frac{\exp\left(\frac{-M\overline{X^2}}{2\sigma^2}\right)}{\sigma^{2M}}\cdot\prod_{i=1}^{M}x_i$$

The log-likelihood function is:

$$\ell(\sigma; X) = \log\left(\exp\left(\frac{-M\overline{X^2}}{2\sigma^2}\right)\right) - \log(\sigma^{2M}) + \log(\prod_{i=1}^{M}x_i)$$
$$\ell(\sigma; X) = \frac{-M\overline{X^2}}{2\sigma^2} - 2M\log(\sigma) + M\overline{\log(X)}$$
$$\ell(\sigma; X) = -M\left(\frac{\overline{X^2}}{2\sigma^2} + 2\log(\sigma) - \overline{\log(X)}\right) \qquad (4)$$

- *Maximizing the Log-Likelihood*

To find the MLE for σ, we take the derivative of $\ell(\sigma; X)$ with respect to σ and set it to zero:

$$\frac{\partial\ell(\sigma;X)}{\partial\sigma} = -M\left(\frac{-\overline{X^2}}{\sigma^3} + \frac{2}{\sigma}\right) = 0$$
$$\frac{\overline{X^2}}{\sigma^3} - \frac{2}{\sigma} = 0$$
$$\overline{X^2} - 2\sigma^2 = 0$$
$$\sigma^2 = \frac{\overline{X^2}}{2}$$

Thus, the MLE for σ is:

$$\hat{\sigma}_{MLE} = \sqrt{\frac{\overline{X^2}}{2}} \qquad (5)$$

2. **Gamma Distribution**

Given a sample $X = \{x_1, x_2, \ldots, x_M\}$, the likelihood function is:

$$L(\alpha,\beta; X) = \prod_{i=1}^{M}\frac{x_i^{\alpha-1}\exp\left(-\frac{x_i}{\beta}\right)}{\Gamma(\alpha)\beta^\alpha}$$
$$L(\alpha,\beta; X) = \frac{\exp\left(\frac{-M\bar{X}}{\beta}\right)}{(\Gamma(\alpha)\beta^\alpha)^M}\cdot\prod_{i=1}^{M}x_i^{\alpha-1}$$

The log-likelihood function is:

$$\ell(\alpha,\beta; X) = \log\left(\exp\left(\frac{-M\bar{X}}{\beta}\right)\right) - \log\left((\Gamma(\alpha)\beta^\alpha)^M\right) + (\alpha-1)M\overline{\log(X)}$$
$$\ell(\alpha,\beta; X) = -M\left(\frac{\bar{X}}{\beta} + \log(\Gamma(\alpha)) + \alpha\log(\beta) - (\alpha-1)\overline{\log(X)}\right) \qquad (6)$$

- *Maximizing the Log-Likelihood*
  - Estimator for β

To find the MLE for β, we take the derivative of $\ell(\alpha,\beta; X)$ with respect to β and set it to zero:

$$\frac{\partial\ell(\alpha,\beta;X)}{\partial\beta} = -M\left(\frac{\alpha}{\beta} - \frac{\bar{X}}{\beta^2}\right) = 0$$
$$\frac{\alpha}{\beta} - \frac{\bar{X}}{\beta^2} = 0$$
$$\beta\alpha - \bar{X} = 0$$
$$\beta = \frac{\bar{X}}{\alpha} \qquad (7)$$

-
  - Estimator for α

To find the MLE for α, we take the derivative of $\ell(\alpha,\beta; X)$ with respect to α and set it to zero:

$$\frac{\partial\ell(\alpha,\beta;X)}{\partial\alpha} = -M\left(\psi(\alpha) + \log(\beta) - \overline{\log(X)}\right) = 0$$

where $\psi(\alpha)$ is the digamma function, defined as the derivative of the logarithm of the gamma function, $\psi(\alpha) = \frac{\Gamma'(\alpha)}{\Gamma(\alpha)}$.

Equating the derivative to zero:

$$\psi(\alpha) + \log(\beta) - \overline{\log(X)} = 0$$

Substituting $\beta = \frac{\bar{X}}{\alpha}$:

$$\psi(\alpha) + \log\left(\frac{\bar{X}}{\alpha}\right) - \overline{\log(X)} = 0$$
$$\psi(\alpha) + \log(\bar{X}) - \log(\alpha) - \overline{\log(X)} = 0 \qquad (8)$$

This is a transcendental equation and must be solved numerically.

Thus, the MLE for α is found by solving:

$$\hat{\beta}_{MLE} = \frac{\bar{X}}{\hat{\alpha}_{MLE}}$$

where $\hat{\alpha}_{MLE}$ is the solution to the equation:

$$\psi(\hat{\alpha}_{MLE}) + \log(\bar{X}) = \log(\hat{\alpha}_{MLE}) + \overline{\log(X)}$$

Exercise 2: Application to Real Data

1. *Implementing Estimators in Python*: Develop Python functions to compute the estimators for the Rayleigh and Gamma distributions using both the Method of Moments and Maximum Likelihood Estimation.
   - **Implementation Tasks**:
     - Create a function that computes the MoM and MLE estimators for a given sample and distribution type.
     - Ensure the function handles the complexity of the Gamma-MLE estimator.

In [ ]:
```python
# Import required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
from scipy.special import digamma, gamma as gamma_function
from scipy.optimize import fsolve
from scipy.stats import rayleigh, gamma

# Ignore warnings
warnings.filterwarnings('ignore')
```

In [ ]:
```python
# Function to estimate the parameter of a distribution using the Method of Moments (MoM) or Maximum Likelihood Estimation (MLE)
# Distribution options: 'Exponential', 'Rayleigh', 'Gamma'
def estimator(data, distribution, method='MoM'):
    if distribution == 'Exponential':
        if method == 'MoM' or method == 'MLE':
            return np.mean(data)
    elif distribution == 'Rayleigh':
        if method == 'MoM':
            return np.sqrt(2 / np.pi) * np.mean(data)
        elif method == 'MLE':
            return np.sqrt(np.mean(data ** 2) / 2)
    elif distribution == 'Gamma':
        if method == 'MoM':
            estimator1 = np.mean(data) ** 2 / np.var(data)
            estimator2 = np.var(data) / np.mean(data)
            return estimator1, estimator2
        elif method == 'MLE':
            # Initial guess for alpha using MoM estimator
            initial_guess = np.mean(data) ** 2 / np.var(data)

            # Solve trascendental equation for alpha using fsolve
            estimator1 = fsolve(lambda alpha: digamma(alpha) + np.log(np.mean(data)) - np.log(alpha) - np.mean(np.log(data)),
                                initial_guess)[0]
            estimator2 = np.mean(data) / estimator1
            return estimator1, estimator2
```

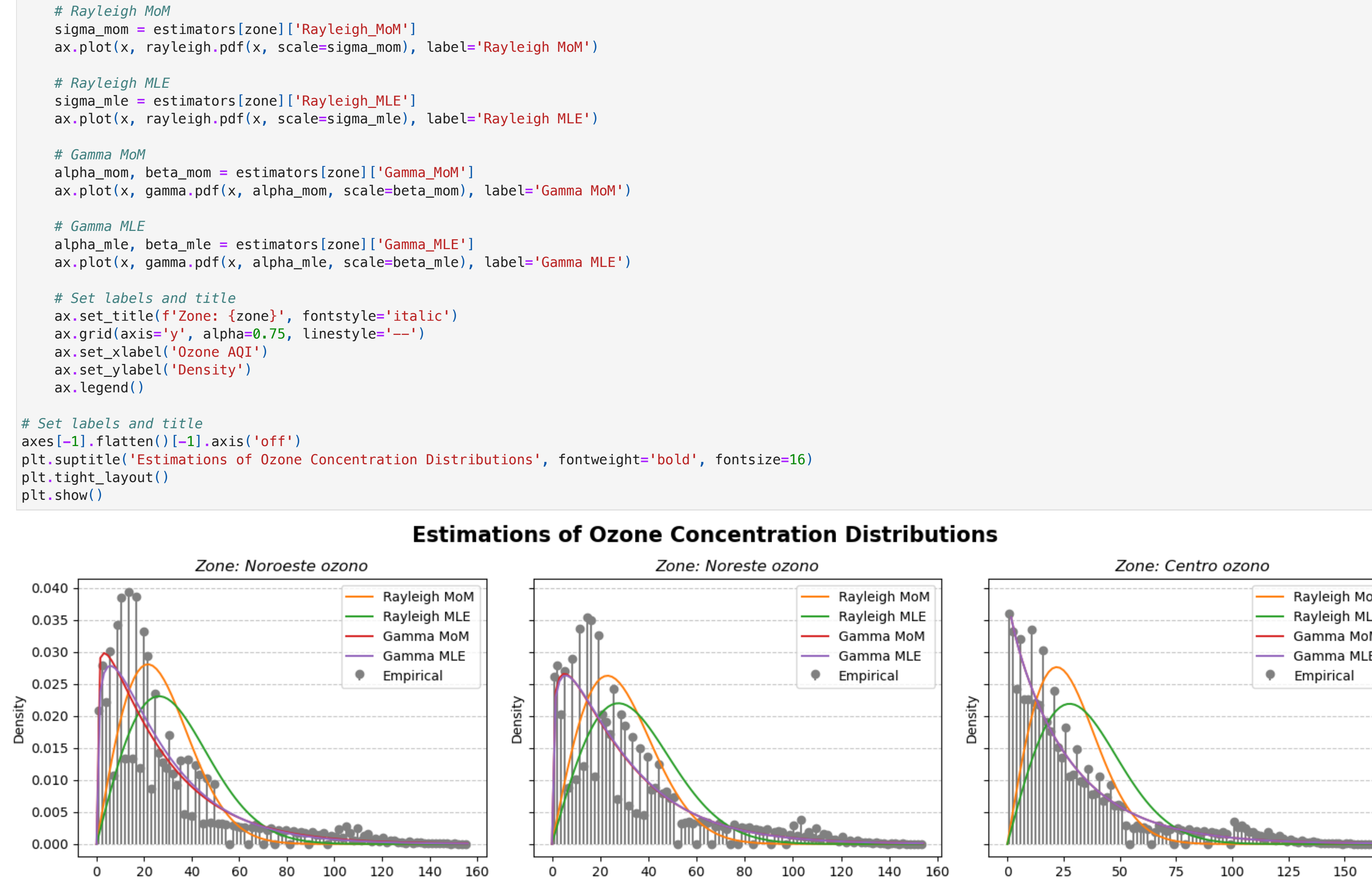
2. *Estimation for Selected Pollutant Across All Zones*: Using a single pollutant across all zones, compute the four probability density functions (pdfs) for each zone.
   - **Tasks**:
     - Calculate the MoM and MLE estimators for both Rayleigh and Gamma distributions for each zone.
     - Create a plot with five panels, each representing a zone, showing the empirical distribution (histogram) and the four estimated pdfs.

In [ ]:
```python
# Load data from CSV files for the years 2019 to 2022.
# Skip the first 8 rows to bypass the metadata and load the actual data.
data19 = pd.read_csv('indice_2019.csv', encoding='latin1', skiprows=8)
data20 = pd.read_csv('indice_2020.csv', encoding='latin1', skiprows=8)
data21 = pd.read_csv('indice_2021.csv', encoding='latin1', skiprows=8)
data22 = pd.read_csv('indice_2022.csv', encoding='latin1', skiprows=8)

# Concatenate the data from all four years into a single DataFrame.
data = pd.concat([data19, data20, data21, data22], ignore_index=True)

# Delete the individual yearly DataFrames to free up memory.
del data19, data20, data21, data22

# Create a datetime column by combining the 'Fecha' and 'Hora' columns.
# Adjust the 'Hora' column to be zero-indexed and format it appropriately.
data['Fecha y Hora'] = pd.to_datetime(
    data['Fecha'] + '-' + np.mod(data['Hora'] - 1, 24).astype(str) + ':00:00',
    format='%d/%m/%Y-%H:%M:%S'
)

# Select only the O3 columns and corresponding dates
ozone_indexes = np.arange(2, 32, 6)
zone_columns = data.columns[ozone_indexes]
ozone_data = data[list(zone_columns) + ['Fecha y Hora']]
```

In [ ]:
```python
# Calculate the estimators for each zone
estimators = {}
for zone in zone_columns:
    zone_data = ozone_data[zone].dropna()
    estimators[zone] = {
        'Rayleigh_MoM': estimator(zone_data, 'Rayleigh', method='MoM'),
        'Rayleigh_MLE': estimator(zone_data, 'Rayleigh', method='MLE'),
        'Gamma_MoM': estimator(zone_data, 'Gamma', method='MoM'),
        'Gamma_MLE': estimator(zone_data, 'Gamma', method='MLE')
    }
```

In [ ]:
```python
# Plot the empirical distribution and estimated pdfs for each zone
fig, axes = plt.subplots(2, 3, figsize=(15, 8), sharey=True)

for zone, ax in zip(zone_columns, axes.flatten()):
    # Calculate the histogram data
    zone_data = ozone_data[zone].dropna()
    x = np.linspace(0, zone_data.max(), 100)
    counts, bins = np.histogram(zone_data, bins=x, density=True)
    bins = 0.5 * (bins[:-1] + bins[1:])

    # Plot empirical histogram using stem plot
    #ax.hist(zone_data, bins=50, density=True, alpha=0.7, color='gray', histtype='step', linewidth=3, label='Empirical')
    ax.stem(bins, counts, linefmt='gray', basefmt=' ', label='Empirical')

    # Rayleigh MoM
    sigma_mom = estimators[zone]['Rayleigh_MoM']
    ax.plot(x, rayleigh.pdf(x, scale=sigma_mom), label='Rayleigh MoM')

    # Rayleigh MLE
    sigma_mle = estimators[zone]['Rayleigh_MLE']
    ax.plot(x, rayleigh.pdf(x, scale=sigma_mle), label='Rayleigh MLE')

    # Gamma MoM
    alpha_mom, beta_mom = estimators[zone]['Gamma_MoM']
    ax.plot(x, gamma.pdf(x, alpha_mom, scale=beta_mom), label='Gamma MoM')

    # Gamma MLE
    alpha_mle, beta_mle = estimators[zone]['Gamma_MLE']
    ax.plot(x, gamma.pdf(x, alpha_mle, scale=beta_mle), label='Gamma MLE')

    # Set labels and title
    ax.set_title(f'Zone: {zone}', fontstyle='italic')
    ax.grid(axis='y', alpha=0.75, linestyle='--')
    ax.set_xlabel('Ozone AQI')
    ax.set_ylabel('Density')
    ax.legend()

# Set labels and title
axes[-1].flatten()[-1].axis('off')
plt.suptitle('Estimations of Ozone Concentration Distributions', fontweight='bold', fontsize=16)
plt.tight_layout()
plt.show()
```

Estimations of Ozone Concentration Distributions



Exercise 3: Evaluation

1. *Comparing Gamma-MLE and Rayleigh-MLE*: Evaluate the performance of the Gamma-MLE and Rayleigh-MLE models for each region.
   - **Tasks**:
     - For each region, compute:
       - a) The log-likelihood of the sample evaluated at the MLE estimators.
       - b) The Bhattacharyya distance between the empirical distribution and the estimated distributions.
     - Determine the best-fitted model based on these metrics.

In [ ]:
```python
# Function to compute the log-likelihood of the sample for a given distribution and its parameters using derived formulas
def log_likelihood(data, distribution, params):
    data += 1e-10 # Add a small epsilon to avoid log(0) issues
    if distribution == 'Rayleigh':
        sigma = params
        return -len(data) * (np.mean(data**2) / (2 * sigma**2) + 2 * np.log(sigma) - np.mean(np.log(data)))
    elif distribution == 'Gamma':
        alpha, beta = params
        return -len(data) * (np.mean(data) / beta + np.log(gamma_function(alpha)) + alpha * np.log(beta) - (alpha - 1) * np.mean(np.log(data)))
```

In [ ]:
```python
"""_Alternative log-likelihood function using scipy module_
# Function to compute the log-likelihood of the sample for a given distribution and its parameters
def log_likelihood(data, distribution, params):
    data += 1e-10 # Add a small epsilon to avoid log(0) issues
    if distribution == 'Rayleigh':
        sigma = params
        return np.sum(rayleigh.logpdf(data, scale=sigma))
    elif distribution == 'Gamma':
        alpha, beta = params
        return np.sum(gamma.logpdf(data, alpha, scale=beta))
"""
```

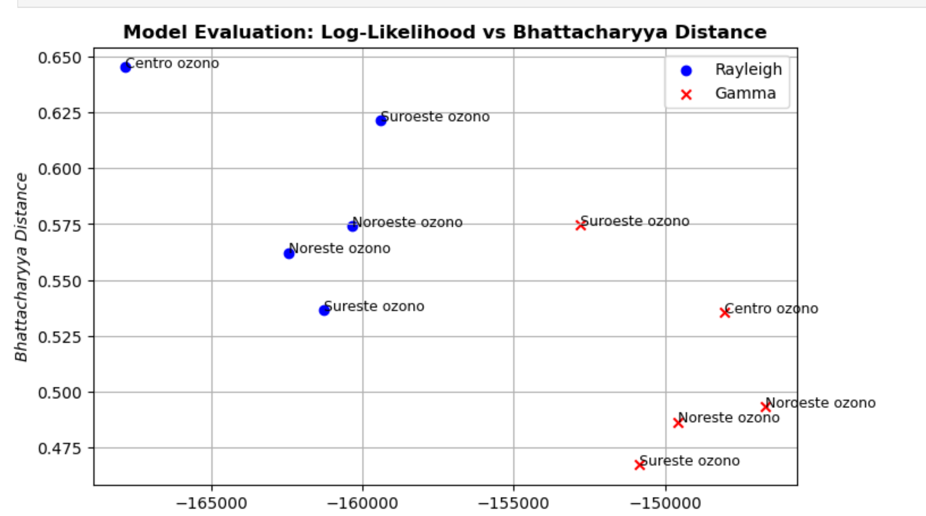
In [ ]:
```python
# Function to compute the Bhattacharyya distance between the empirical and estimated distributions
def bhattacharyya_distance(data, distribution, params):
    # Calculate the histogram data
    x = np.linspace(0, data.max(), 100)
    hist_data, bins = np.histogram(data, bins=x, density=True)
    bin_centers = 0.5 * (bins[:-1] + bins[1:])

    # Estimate the PDF using the given distribution and parameters
    if distribution == 'Rayleigh':
        sigma = params
        pdf_est = rayleigh.pdf(bin_centers, scale=sigma)
    elif distribution == 'Gamma':
        alpha, beta = params
        pdf_est = gamma.pdf(bin_centers, alpha, scale=beta)

    # Calculate Bhattacharyya distance
    bc = np.sum(np.sqrt(hist_data * pdf_est))
    return -np.log(bc)
```

In [ ]:
```python
# Evaluate the models
results = {}
for zone in zone_columns:
    zone_data = ozone_data[zone].dropna()
    rayleigh_params = estimators[zone]['Rayleigh_MLE']
    gamma_params = estimators[zone]['Gamma_MLE']

    results[zone] = {
        'Log-Likelihood Rayleigh': log_likelihood(zone_data, 'Rayleigh', rayleigh_params),
        'Log-Likelihood Gamma': log_likelihood(zone_data, 'Gamma', gamma_params),
        'Bhattacharyya Distance Rayleigh': bhattacharyya_distance(zone_data, 'Rayleigh', rayleigh_params),
        'Bhattacharyya Distance Gamma': bhattacharyya_distance(zone_data, 'Gamma', gamma_params)
    }

# Determine the best-fitted model for each zone
for zone, metrics in results.items():
    best_model_ll = 'Gamma' if metrics['Log-Likelihood Gamma'] > metrics['Log-Likelihood Rayleigh'] else 'Rayleigh'
    best_model_bd = 'Gamma' if metrics['Bhattacharyya Distance Gamma'] < metrics['Bhattacharyya Distance Rayleigh'] else 'Rayleigh'
    print(f"\nZone: \t{zone}")
    print(f"\t- Best model by Log-Likelihood: {best_model_ll}")
    print(f"\t- Best model by Bhattacharyya Distance: {best_model_bd}")
    print(f"\t\t· Log-Likelihood Rayleigh: {round(metrics['Log-Likelihood Rayleigh'], 4)}")
    print(f"\t\t· Log-Likelihood Gamma: {round(metrics['Log-Likelihood Gamma'], 4)}")
    print(f"\t\t· Bhattacharyya Distance Rayleigh: {round(metrics['Bhattacharyya Distance Rayleigh'], 4)}")
    print(f"\t\t· Bhattacharyya Distance Gamma: {round(metrics['Bhattacharyya Distance Gamma'], 4)}")
```

```
Zone:   Noroeste ozono
        - Best model by Log-Likelihood: Gamma
        - Best model by Bhattacharyya Distance: Gamma
                · Log-Likelihood Rayleigh: -160322.6147
                · Log-Likelihood Gamma: -146684.3118
                · Bhattacharyya Distance Rayleigh: 0.5742
                · Bhattacharyya Distance Gamma: 0.4932

Zone:   Noreste ozono
        - Best model by Log-Likelihood: Gamma
        - Best model by Bhattacharyya Distance: Gamma
                · Log-Likelihood Rayleigh: -162420.4886
                · Log-Likelihood Gamma: -149577.9778
                · Bhattacharyya Distance Rayleigh: 0.5622
                · Bhattacharyya Distance Gamma: 0.4866

Zone:   Centro ozono
        - Best model by Log-Likelihood: Gamma
        - Best model by Bhattacharyya Distance: Gamma
                · Log-Likelihood Rayleigh: -167824.1163
                · Log-Likelihood Gamma: -148045.3685
                · Bhattacharyya Distance Rayleigh: 0.6452
                · Bhattacharyya Distance Gamma: 0.5355

Zone:   Suroeste ozono
        - Best model by Log-Likelihood: Gamma
        - Best model by Bhattacharyya Distance: Gamma
                · Log-Likelihood Rayleigh: -159405.7574
                · Log-Likelihood Gamma: -152782.2052
                · Bhattacharyya Distance Rayleigh: 0.6212
                · Bhattacharyya Distance Gamma: 0.5747

Zone:   Sureste ozono
        - Best model by Log-Likelihood: Gamma
        - Best model by Bhattacharyya Distance: Gamma
                · Log-Likelihood Rayleigh: -161261.2617
                · Log-Likelihood Gamma: -150859.676
                · Bhattacharyya Distance Rayleigh: 0.5364
                · Bhattacharyya Distance Gamma: 0.4674
```

In [ ]:
```python
# Plot log-likelihood vs Bhattacharyya distance
fig, ax = plt.subplots(figsize=(8, 5))

for zone, metrics in results.items():
    ax.scatter(metrics['Log-Likelihood Rayleigh'], metrics['Bhattacharyya Distance Rayleigh'],
               color='blue', label='Rayleigh' if zone == zone_columns[0] else "", marker='o')
    ax.scatter(metrics['Log-Likelihood Gamma'], metrics['Bhattacharyya Distance Gamma'],
               color='red', label='Gamma' if zone == zone_columns[0] else "", marker='x')
    ax.text(metrics['Log-Likelihood Rayleigh'], metrics['Bhattacharyya Distance Rayleigh'], zone, fontsize=9)
    ax.text(metrics['Log-Likelihood Gamma'], metrics['Bhattacharyya Distance Gamma'], zone, fontsize=9)

# Set labels and title
ax.set_xlabel('Log-Likelihood', fontstyle='italic')
ax.set_ylabel('Bhattacharyya Distance', fontstyle='italic')
ax.set_title('Model Evaluation: Log-Likelihood vs Bhattacharyya Distance', fontweight='bold')
ax.legend(loc=0)
ax.grid(True)
plt.show()
```

Model Evaluation: Log-Likelihood vs Bhattacharyya Distance



Based on the log-likelihood and Bhattacharyya distance metrics, the **Gamma distribution model (Gamma-MLE)** consistently performed better than the Rayleigh distribution model (Rayleigh-MLE) across all zones. This conclusion is drawn from both the higher log-likelihood values and the lower Bhattacharyya distances observed for the Gamma model compared to the Rayleigh model in each zone. Thus, for modeling AQI data in these zones, the Gamma distribution is the preferred choice.