VulnerabilityMatrixProvider Documentation

1 Overview

The `VulnMatrixProvider` class is designed to calculate and provide probability matrices for impact assessments in hazard analysis. It's particularly useful in scenarios where you need to estimate the likelihood of different impacts (such as damage or disruption) on assets, given various hazard intensities.

2 Classes

2.1 Distribution

A simple class representing a statistical distribution.

2.1.1 Attributes:

- `mean` (float): The mean of the distribution.
- `std_dev` (float): The standard deviation of the distribution.

2.2 VulnMatrixProvider

The main class for creating and manipulating vulnerability matrices.

2.2.1 Attributes:

- `intensity_bin_centres` (np.ndarray): An array of intensity bin centers.
- `impact_cdfs` (Sequence[Callable[[np.ndarray], np.ndarray]]): A sequence of callable functions, each representing a cumulative distribution function (CDF) for impacts at a specific intensity.

2.2.2 Methods:

`__init__(self, intensity_bin_centres: np.ndarray, impact_cdfs: Sequence[Callable[[np.ndarray np.ndarray]])` Constructor for the VulnMatrixProvider class.

- **Parameters:**

  - `intensity_bin_centres` (np.ndarray): An array of intensity bin centers.
  - `impact_cdfs` (Sequence[Callable[[np.ndarray], np.ndarray]]): A sequence of callable functions, each representing a CDF for impacts at a specific intensity.
- **Raises:**
  - `ValueError`: If the intensities are not sorted and increasing, or if the number of impact_cdfs doesn't match the number of intensity_bin_centres.

`to_prob_matrix(self, impact_bin_edges: np.ndarray) -> np.ndarray` Calculates and returns a probability matrix.

- **Parameters:**
  - `impact_bin_edges` (np.ndarray): An array of impact bin edges.
- **Returns:**
  - np.ndarray: A 2D probability matrix where p[i, j] represents the conditional probability that given the intensity falls in bin i, the impact is in bin j.

3 Usage Example

```python
import numpy as np
from vulnerability_matrix_provider import VulnMatrixProvider

# Define intensity bin centers
intensity_bins = np.array([1, 2, 3, 4, 5])

# Define impact CDFs for each intensity
def impact_cdf_1(x): return np.minimum(x / 2, 1)
def impact_cdf_2(x): return np.minimum(x / 3, 1)
def impact_cdf_3(x): return np.minimum(x / 4, 1)
def impact_cdf_4(x): return np.minimum(x / 5, 1)
def impact_cdf_5(x): return np.minimum(x / 6, 1)

impact_cdfs = [impact_cdf_1, impact_cdf_2, impact_cdf_3,
    impact_cdf_4, impact_cdf_5]

# Create VulnMatrixProvider instance
vuln_matrix_provider = VulnMatrixProvider(intensity_bins,
    impact_cdfs)

# Define impact bin edges
impact_bin_edges = np.array([0, 0.2, 0.4, 0.6, 0.8, 1.0])

# Calculate probability matrix
```

```python
prob_matrix = vuln_matrix_provider.to_prob_matrix(impact_bin_edges)

print(prob_matrix)
```

This example demonstrates how to create a `VulnMatrixProvider` instance, define intensity bins and impact CDFs, and calculate a probability matrix.

4 Notes

- The `VulnMatrixProvider` class uses `__slots__` for memory optimization.
- The class is designed to work with NumPy arrays for efficient numerical computations.
- Ensure that the number of `impact_cdfs` matches the number of `intensity_bin_centres` to avoid ValueError.
- The `to_prob_matrix` method calculates probabilities based on the difference between adjacent CDF values.