**Theory and Research Part**

**1. Median as the Minimizer of the Sum of Absolute Deviations**

The task is to illustrate why the median minimizes the sum of absolute deviations, ∑∣xi−c∣\sum |x\_i - c|. Here's the outline of the theoretical part:

* **Key Idea**: The absolute deviation function ∣xi−c∣|x\_i - c| is piecewise linear and non-differentiable at c=xic = x\_i. Its minimum occurs where half the data points are less than cc and half are greater, which is the definition of the median.
* **Proof Outline**:
  1. Consider a set of data points x1,x2,…,xnx\_1, x\_2, \ldots, x\_n.
  2. For cc between two adjacent data points xkx\_k and xk+1x\_{k+1}, the derivative of ∑∣xi−c∣\sum |x\_i - c| is:
     + Negative when c<medianc < \text{median},
     + Positive when c>medianc > \text{median}.
  3. Hence, the absolute deviation function is minimized at the median.

**2. Other Measures of Central Tendency**

Central tendency measures synthesize the "location" of a distribution. Beyond the median:

* **Mean**: Minimizes ∑(xi−c)2\sum (x\_i - c)^2.
* **Mode**: The value with the highest frequency.
* **Geometric Mean**: Used for proportional data, defined as x1⋅x2⋯xnn\sqrt[n]{x\_1 \cdot x\_2 \cdots x\_n}.
* **Harmonic Mean**: Reciprocal of the arithmetic mean of reciprocals, used in rates.
* **Trimmed Mean**: Removes outliers by averaging only central values.
* **Midrange**: min(x)+max(x)2\frac{\text{min}(x) + \text{max}(x)}{2}.
* **Weighted Mean**: Accounts for the weight of each value wiw\_i.

**Generalizations to Infinite Central Tendency Definitions**

The "location" can also be synthesized through:

* **Percentiles**: The kk-th percentile minimizes ∑p(xi−c)\sum p(x\_i - c) with asymmetric penalties.
* **Entropy Minimization**: Based on Shannon entropy, representing the "central point" of information.
* **Moment Statistics**: Generalizing location as a specific moment around zero (e.g., skewness, kurtosis).

**Application/Practice: Stochastic Differential Equation (SDE) Simulator**

We will refine the simulator to include continuous time with a constant attack rate λ\lambda.

**Steps to Develop the Simulator**

1. **Break Time into Sub-Intervals**:
   * Define the temporal window TT.
   * Subdivide TT into nn intervals: dt=Tndt = \frac{T}{n}.
2. **Simulate Jump Probability**:
   * Assign a probability of attack success Pjump=λ⋅dtP\_{\text{jump}} = \lambda \cdot dt to each interval.
   * For each interval, generate a random value:
     + If random<Pjump\text{random} < P\_{\text{jump}}, add a jump of +1+1.
3. **Integration into the SDE Framework**:
   * Approximate the process as: X(ti+1)=X(ti)+Jump ProbabilityX(t\_{i+1}) = X(t\_i) + \text{Jump Probability}
   * Store and plot X(t)X(t) over time.
4. **Extend the Interface**:
   * Add controls for λ\lambda and nn (number of sub-intervals).
   * Add options to view intermediate and final means/variances.
5. **Generate Results**:
   * Plot the stochastic process.
   * Display final and intermediate results (mean, variance).

**Implementation Steps**

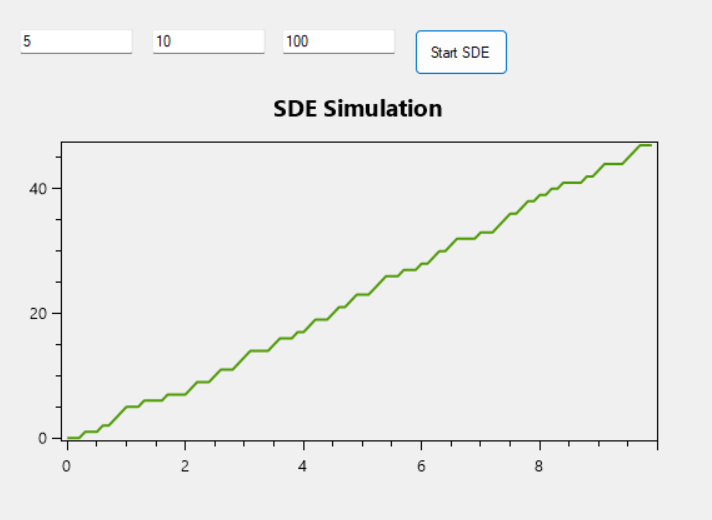
1. **Interface Updates**:
   * Add input fields for:
     + λ\lambda (Rate of Attack).
     + nn (Number of Sub-Intervals).
   * Add labels for intermediate and final results.
2. **Integrate into Unified Classes**:
   * Extend the shared ISimulator interface with SimulateContinuousProcess.
   * Update StochasticSimulator to handle both jump probabilities and continuous time.
3. **Output**:
   * A plot displaying the stochastic process.
   * Numerical outputs of intermediate and final statistics.

**Example 1: Basic Test**

* **Lambda (λ):** 5  
  (Expected number of attacks in the time window)
* **Total Time:** 10  
  (Simulate attacks over a period of 10 units of time)
* **Intervals (n):** 100  
  (Divide the time window into 100 intervals)

**Expected Behavior:**

* Simulation should run smoothly without errors.
* Plot shows a gradual increase in the attack count over time.



Example 2: High Attack Rate

Lambda (λ): 20

(High expected number of attacks in the time window)

Total Time: 5

(Shorter time period to observe more frequent attacks)

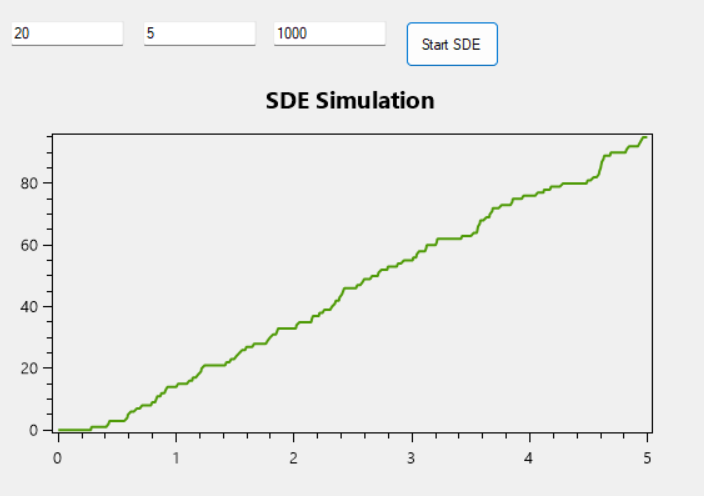
Intervals (n): 1000

(Fine granularity with small time steps)

Expected Behavior:

Higher density of attack jumps in the plot.

More noticeable variance in the attack pattern.



**Example 3: Edge Case - Small Lambda**

* **Lambda (λ):** 0.1  
  (Very low attack rate)
* **Total Time:** 10
* **Intervals (n):** 100

**Expected Behavior:**

* Plot shows sparse or no attacks over the simulation period.
* Mean and variance close to 0.

A screen shot of a graph

Description automatically generated