Flood Risk Assessment Model

This model aims to predict the probability of a flood event exceeding a threshold level in a catchment area. The prediction is based on independent variables: rainfall intensity, runoff coefficient, soil permeability, and terrain slope. Monte Carlo simulations will sample each variable and calculate the potential flood volume for each scenario to estimate flood risk.

The model that has been chosen for the simulation is,

$$V = C \times R \times (1 - e^{-KS}) \times (\frac{1}{(1 + e^{(-\alpha P)})})$$

V is the flood discharge per unit area in mm/hr

C is the runoff coefficient

R is the rainfall intensity in mm/hr

S is the terrain slope

P is the soil permeability in m/s

K is a scaling factor for terrain slope which controls how much terrain slope affects the runoff. Higher values of k indicate a stronger influence of slope on runoff. In mountainous regions or areas with steep terrain, k will be relatively higher, while in flatter regions, it should be lower. In the simulation, K is taken as 0.5.

The parameter **α** influences how permeability affects runoff. The higher the value of α\alphaα, the more sharply soil permeability will reduce runoff. In the simulation **α** value is taken as 1.5

Defining Independent Variables and Distributions

Variables	Distribution	Explanation for the distribution
Rainfall Intensity (R)	Log-Normal	Captures variability and skewness in extreme rainfall events.
Runoff Coefficient (C)	Beta	Bounded between 0 and 1; represents the fraction of rainfall contributing to runoff.
Soil Permeability (P)	Normal	Soils tend to have variable permeability across a region, and using an exponential distribution helps to simplify the representation of this heterogeneity without requiring detailed measurements of permeability across large areas.
Terrain Slope (S)	Exponential	Captures distribution of slope across the area, as steep slopes are less frequent than mild ones.

Exponential factor $(1 - e^{-KS})$ simulates that higher slopes lead to higher runoff and makes the model nonlinear.

And as the value of s increases, the expression values goes to 1.In the expression, $(\frac{1}{(1 + e^{(-\alpha P)})})$ as the value of P increases, the value of the expression decreases. Which makes sense as the permeability increases, runoff will decrease.

Setting Parameters and Thresholds

Assumptions:

Rainfall Intensity: Log-normal distribution with mean = 50 mm/hr, variance = 25.

Runoff Coefficient: Beta distribution with shape parameters α=2 and β=5

Soil Permeability: Normal distribution with mean = 10^-4 m/s, variance = 10^-11 m2/s2.

Terrain Slope: Exponential distribution with rate parameter = 0.2.

Threshold Analysis:

Compare each V with a predefined flood threshold volume V_{th}

Count the cases where $V>V_{th}$ to estimate the probability of flood risk.

$$Probability\ Flood\ risk = \frac{No.of\ times\ V > V_{th}}{No.of\ times\ V\ is\ calculated}$$

In this simulation, Python was used as the programming environment due to its powerful libraries and tools for data analysis.

Key Libraries and Functions Used

NumPy (np): This library provides efficient numerical computations and random sampling from different probability distributions, essential for simulating each variable in the model.

Functions Used:

np.random.lognormal(mean, sigma, size): Generates lognormal-distributed random values, used for simulating **rainfall intensity (R)**.

np.random.beta(alpha, beta, size): Samples from a beta distribution, used for the **runoff coefficient (C)**.

np.random.exponential(scale, size): Generates values from an exponential distribution, used for **terrain slope (S)**.

np.random.normal(loc, scale, size): Samples from a normal distribution, used for **soil permeability (P)**.

Simulation Process

Variable Initialization: Parameters such as mean and variance were defined for each variable (e.g., rainfall intensity, runoff coefficient, terrain slope, soil permeability) based on Indian conditions. Each variable was assigned a unique distribution to reflect realistic conditions.

Random Sampling and Computation:

Monte Carlo sampling was applied by generating 1000 random values for each variable using the specific probability distributions. This approach enables the model to capture the variability of environmental conditions.

The flood volume V was calculated for each set of sampled values using the given nonlinear formula.

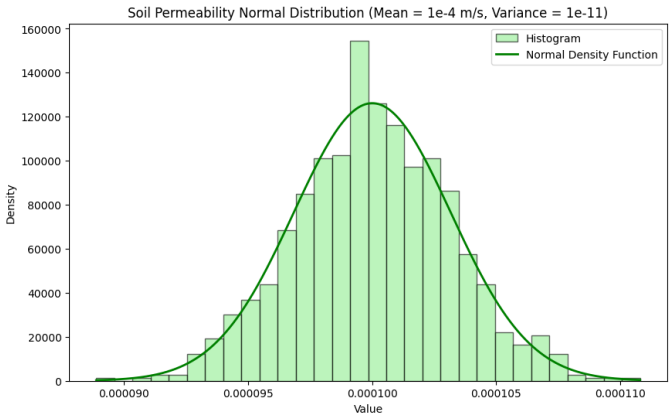
Flood Risk Estimation:

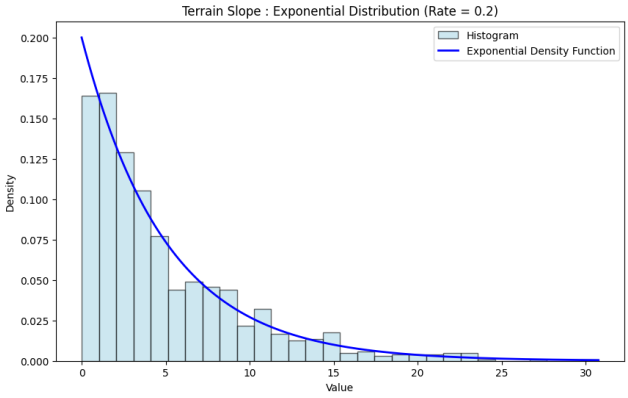
The simulation counted instances where the flood volume V exceeded a defined threshold V_{th} allowing for flood risk estimation based on the simulated conditions.

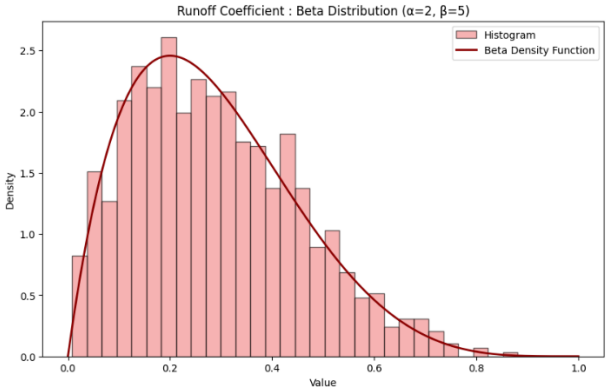
Visualization:

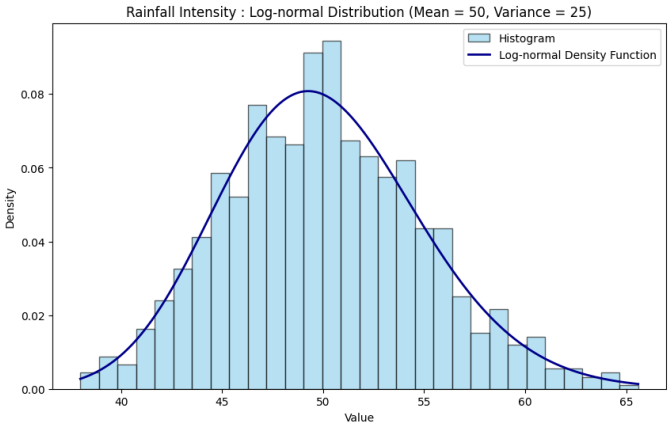
Histograms of each input variable were created to examine their distribution, ensuring they matched expected patterns (e.g., lognormal for rainfall, beta for runoff coefficient).

Graphic representation of independent variables for 1 simulation with its variability is shown in the following histograms and their pdf.



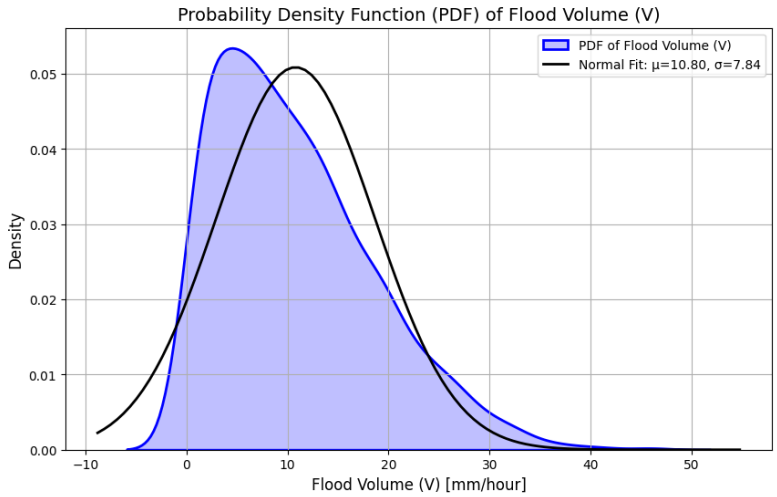






Working through simulations, I learned that Python's flexibility and its robust statistical libraries make it perfect for this type of work. I started with simple simulations, like estimating the value of π, before moving on to more complex applications. Each step helped me appreciate the power of Python and how Monte Carlo simulations can be used for everything from predicting stock prices to simulating physical systems.

To visualize the variability of the dependent variable, V, it's fitted with normal distribution with mean of 10.8 mm/hr and variance 7.84



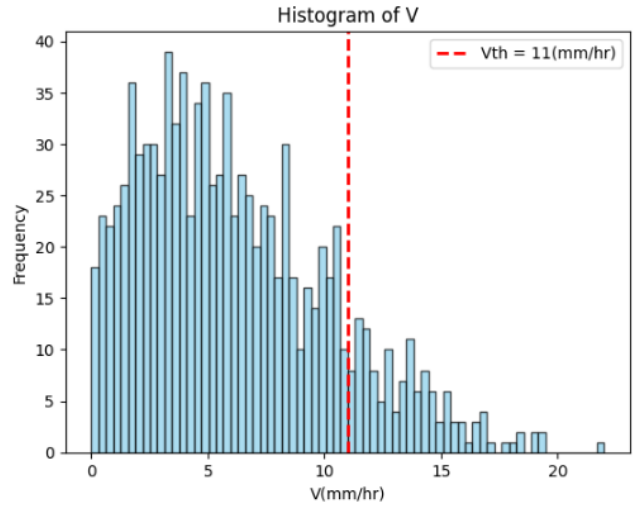
Suppose the fitted PDF indicates that the mean flood volume **V** is 10mm/hour with a standard deviation of 6mm/hour, and the **threshold for critical flooding** is 12 mm/hour.

From the PDF, one can calculate the probability that the flood volume will exceed 12 mm/hour by integrating the area under the curve beyond 12 mm/hour. If this probability is high (e.g., 25%), it indicates a need for significant mitigation planning.

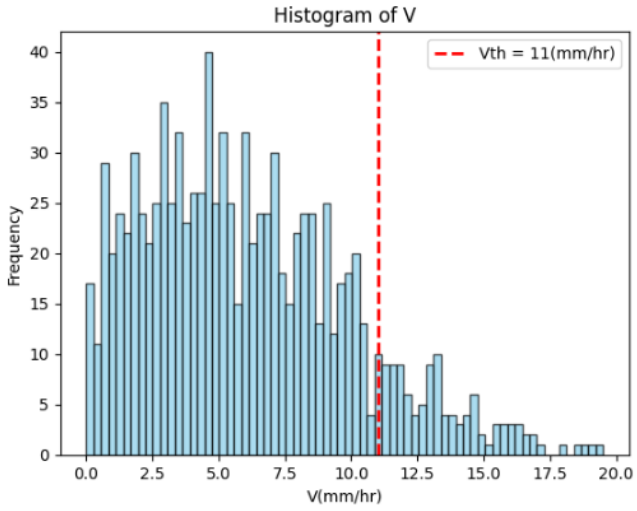
On the other hand, if the probability is low (e.g., 5%), you might decide that **flood protection infrastructure** is only needed in specific high-risk areas.

Following are the results from 10 simulations,

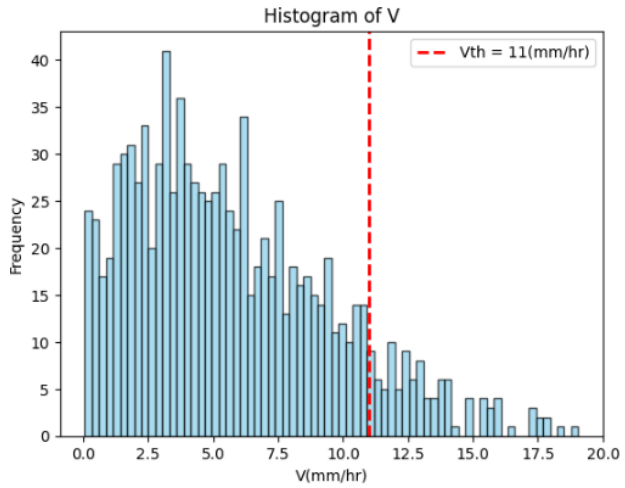
```
Flood risk (V > Vth): 0.1310
Number of times V exceeds Vth: 131
```



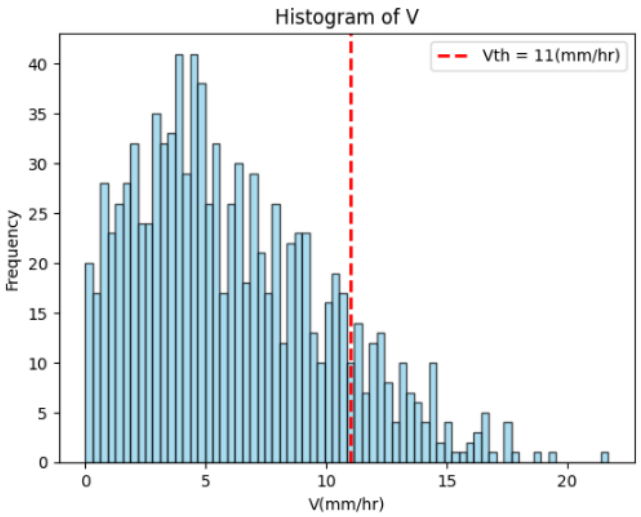

```
Flood risk (V > Vth): 0.1130
Number of times V exceeds Vth: 113
```



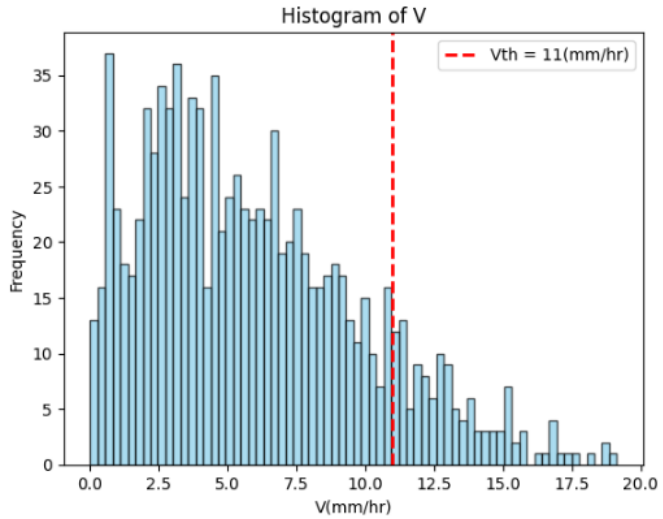

```
Flood risk (V > Vth): 0.1020
Number of times V exceeds Vth: 102
```



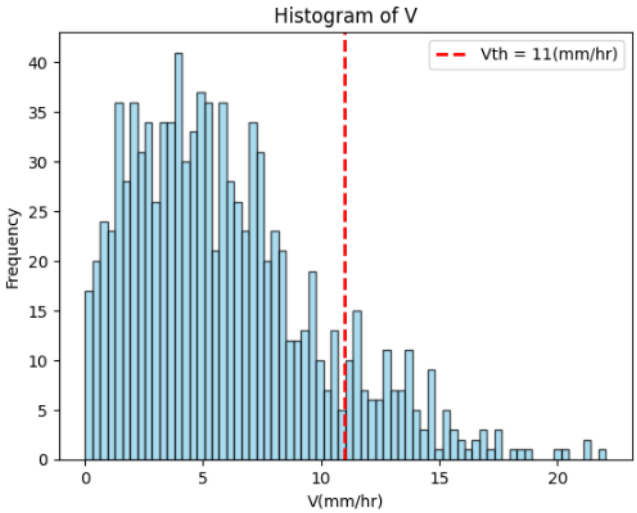

```
Flood risk (V > Vth): 0.1240
Number of times V exceeds Vth: 124
```



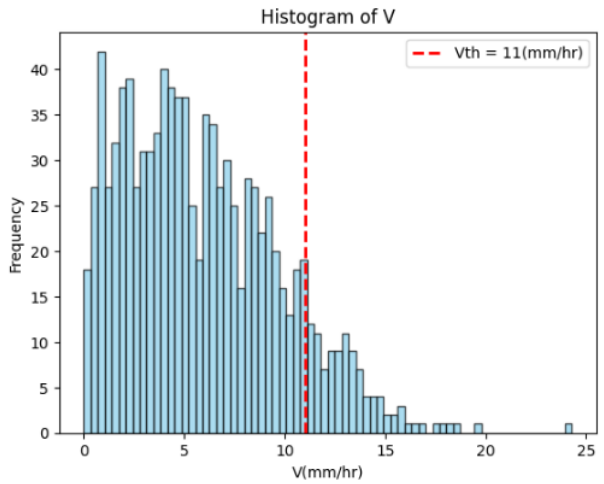

```
Flood risk (V > Vth): 0.1210
Number of times V exceeds Vth: 121
```



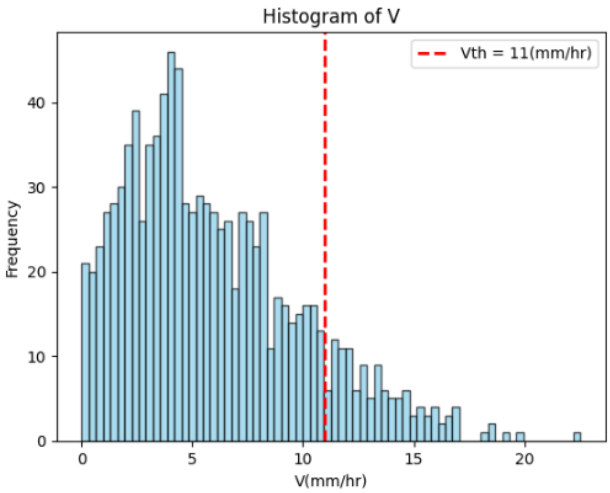

```
Flood risk (V > Vth): 0.1260
Number of times V exceeds Vth: 126
```



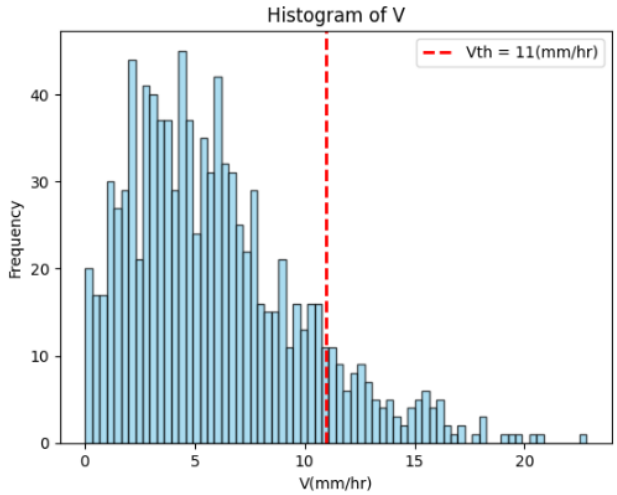
Flood risk (V > Vth): 0.1090
Number of times V exceeds Vth: 109



Flood risk (V > Vth): 0.1200
Number of times V exceeds Vth: 120



Flood risk (V > Vth): 0.1110
Number of times V exceeds Vth: 111



Flood risk (V > Vth): 0.1190
Number of times V exceeds Vth: 119