CEVE 101: Project 03

Samantha Mahung (sm308)

2024-10-23

using Revise  
using RainwaterHarvesting

Line 1

Revise allows us to update code without restarting Julia.

Line 2

We load the RainwaterHarvesting package that contains our simulation tools.

### Setting the file path

filepath = "data/20.txt"  
rainfall\_data = RainfallData(filepath)

Line 1

This is the path to the rainfall data file for the station of interest.

Line 2

This creates a RainfallData object containing the rainfall data for the specified station.

RainfallData for station: BARBALHA  
 Location: -7.3072222222222, -39.301638888889  
 Years of data: 49  
 Total days: 17897

### Plotting the Data

plot(rainfall\_data)

Line 1

This plot function is defined in [viz.jl](./RainwaterHarvesting/src/viz.jl) and is a thin wrapper around the Plots.jl package.

### Discussion

There are 49 years of rainfall data for Barbalha after cleaning, which is sufficient for meaningful risk assessments as it contains years with a wide variety of rainfall conditions including droughts. The most notable drought that can be recognized in the data lasted from 2012 to 2018 and was both the worst and most recent drought in Ceara’s history. The large gap in the data would cause an overestimate in how much water Barbalha needs under typical rainfall conditions.

## Understanding the Theoretical Model

### Mass Balance Equation

The mass balance model for the rainwater tank is given by:

* : the volume of water in the tank at time
* is the volume of water added to the tank at time
* is the volume of water consumed at time
* : the maximum volume of the tank

### Inflow ()

Inflow is calculated as

* : the runoff coefficient, which accounts for losses due to evaporation, spilling, etc.
* : the area of the roof (we will use square meters)
* : the precipitation at time (we will use millimeters per day)
* : the first flush volume (we will use millimeters). The first flush volume is often discarded, so that each time it rains the first bit of water (which is usually dirty) is not used (see [here](https://rainwaterharvesting.tamu.edu/pre-storage-treatment/) for more details).

### Consumption

Consumption is modeled as:

The two major assumptions made for this model are that residents only start using water from the tank after the rainy season and that each household uses 74.1 liters per day during the dry season. These are not concrete facts, so we must account for any variation from this assumed standard. To do so, we can conduct a city-wide survey on residents’ change in water usage habits throughout the year and factor in our findings while also including a margin of error to contain any further fluctuations.

## Model Implementation and Testing

With the current parameters, there is only one year out of 49 with failures. Decreasing the roof area and tank capacity increases the number of failures, while increasing the roof area and tank capacity has no effect. Even with an impossibly large tank, there is still one failure. This implies that there was essentially no water in one year, but it is more likely that there is just a year with extremely faulty data, and we cannot know if there was actually a failure in that year. Looking at the rest of the data, it seems the tank is already very reliable and suits the needs of Barbalha.

The one major gap in these methods is, once again, the year of faulty data. It appears that the faulty year is 2011 since running the model using 2011’s data comes up with an error. That year preceded Ceara’s worst drought period but was not a part of it, so it does not make sense that there would be a failure in 2011 when there were no failures during the drought. To address the issue, it would be best to try and figure out what happened why the data is faulty or missing and adjust calculations accordingly. In any case, there should not have been a failure in that year when there were no failures in years with worse droughts, so the data from 2011 should be disregarded or recalculated.

### Defining the Model Parameters

param = ModelParameters(  
 runoff\_coefficient=0.85,  
 roof\_area\_m2=45.0,  
 first\_flush\_mm=2.0,  
 tank\_capacity\_L=32000.0  
)

Line 1

This creates a ModelParameters object with the specified parameters.

Line 2

The runoff coefficient () is a measure of how much of the rain that falls on the roof ends up in the tank.

Line 3

The roof area () is the area of the roof that the rain falls on.

Line 4

The first flush volume () is the volume of rain that is discarded because it is dirty.

Line 5

The tank capacity () is the maximum volume of the tank.

### Running the Simulation for One Year

Let’s run the model for the year 2010.

rainfall\_2010 = rainfall\_data.annual\_data[2010]  
results\_2010 = run\_timesteps(rainfall\_2010, param)  
p1 = plot(results\_2010)

Line 1

We select the annual rainfall data for the year 1981.

Line 2

We run the simulation for the year 1981.

Line 3

We plot the results, again using a plot function defined in [viz.jl](./RainwaterHarvesting/src/viz.jl).

To compare rainfall and tank volume:

p2 = plot(rainfall\_2010)  
plot(p2, p1, layout=(2, 1), size=(1000, 700), link=:x)

### Reliability Analysis

Even with the year of faulty data, the tank is extremely reliable with success 98% of the time. However, the default 16000 liter tank overflows, making it inefficient. Doubling the tank capacity helps increase the storage efficiency and reliability greatly.

all\_years = sort(collect(keys(rainfall\_data.annual\_data)))  
all\_results = [run\_timesteps(rainfall\_data.annual\_data[year], param) for year in all\_years]  
any\_failures = [!isempty(result.failure\_dates) for result in all\_results]  
println("Number of years with failures: ", sum(any\_failures), " out of ", length(all\_years))

Line 1

We get all the years in order.

Line 2

We run the simulation for each year.

Line 3

We check if any failures occurred.

Line 4

We print the number of years with failures and the total number of years.

Number of years with failures: 1 out of 49