CEVE 101: Project 03

Nicolas Cooker (nc103)

2024-10-23

We begin by loading the required packages.

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

Next, we specify the path to the rainfall data file and read it in.

filepath = "data/50.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: GRANJA  
 Location: -3.1195833333333, -40.829166666667  
 Years of data: 49  
 Total days: 17895

If we dig under the hood a bit, we can see that there are two main pieces of information:

* station\_info: a dictionary containing information about the station, such as its name and location.
* annual\_data: a dictionary mapping years to AnnualRainfallData objects, each containing the date and precipitation data for a given year.

### Plotting the Data

We can plot the data for one or all of the years to verify that it looks correct.

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.

# Choose a Station and Load Data

1. The years kept after data cleaning were 1974 through 2024.
2. This data is sufficient for meaningful risk assessments because it includes a wide dataset with 49 years of data.

# Analyze Available Data

1. Historical droughts in Ceará, Brazil in the 20th and 21st centuries occured in 1915, 1931, 1973, 1983, 1993, 1998 and 2012-2018.
2. My data includes all of the years from 1974 and 2018, including 1983, 1993, 1998, and all years from 2012-2018.
3. Missing data during drought years would result in difficulties

## Understanding the Theoretical Model

1. Perhaps the greatest limiting assumption/approximation is that of consumption: that consumption from the tank only begins after the 150th day of the year and that the rainy season lasts for a more or less rigid 150 days each year. It also assumes that all households will use 74.1 liters of water per day, which does not reflect variability in water use between different households.
2. If we wanted to modify the model to relax these assumptions, we would need to account for two separate effects:
3. the idea that water use will start immediately after the 150th day
4. the idea that all households will use 74.1 liters of water per day, regardless of household size

To account for (a), we will need to add some sort of constant that accounts for variability in water usage over time. The result would be a dynamic consumption function, which would include a variable for how many people there are in the household, multiplied by a constant representing the average consumption of water per person. Additionally, there would be a periodic component to consumption, as it would likely increase in the driest months and be not as high in the months bordering the rainy season. This would result in a more natural model of consumption rather than one that just jumps to 74.1 liters after 150 days.

Our function would look something like this:

C(n,t) = nB + max(0, f(t))

Where n is the number of people per household, B is a constant representing the amount of consumption per person, and f(t) is a periodic function that adds periodic behavior to consumption and makes the increase in consumption more natural instead of a jump (we are using the max function in order to only get the values of the periodic function above 0 and to set all negative values below 0, as consumption should never be negative).

*Finding f(t):* It will be either a function of sine or cosine. I chose cosine in this case. We then need to change the period of the function to represent the days in the year, so that we can get a model for consumption for each day. According to my research, the dry season in Brazil typically occurs between mid May (day 150 of the year) and mid November (day 330). We can represent this periodically with a phase shift so that 240 (the midpoint of 150 and 330) is at the peak of the cosine wave, and we get the following equation: -cos((2π/365)(t-57.5)). This function is greater than 0 for any value t such that 150 < t < 330 and reaches its peak at t = 240.

Our resulting function is therefore C(n,t) = nB + max(0, -cos((2π/365)(t-57.5)))

This is a less abrupt consumption function that should more accurately reflect variability in household size and gradual increases and decreases in consumption throughout the dry season.

## Model Implementation and Testing

### Defining the Model Parameters

We can define the model parameters with sensible default values. You will have the opportunity to change these values later.

param = ModelParameters(  
 runoff\_coefficient=.75,  
 roof\_area\_m2=45.0,  
 first\_flush\_mm=2.0,  
 tank\_capacity\_L=20000.0  
)

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.

#### Justify Your Modifications

1. I modified the parameters runoff\_coefficient and tank\_capacity\_L because I wanted the graph to get to the point where the amount of water stored was not plateauing and limited by tank capacity. I wanted to reach a point where the water supply was limited only by the amount collected, not the capacity of the tank.
2. A lower runoff\_coefficient represents less water runoff and therefore less water collected and stored in the tank. A higher one results in more water in the tank. A lower tank\_capacity\_L represents a lower maximum amount of water that can be stored, an upper-ceiling on the amount of water that is collected, and a higher one results in more capacity and therefore more water being stored. I lowered the runoff\_coefficient and raised the tank\_capacity\_L. This also minimized failures, although once the capacity and runoff coefficient get too small, failures increase very quickly.

### Running the Simulation for One Year

Let’s run the model for the year 1981.

rainfall\_1981 = rainfall\_data.annual\_data[1981]  
results\_1981 = run\_timesteps(rainfall\_1981, param)  
p1 = plot(results\_1981)

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\_1981)  
plot(p2, p1, layout=(2, 1), size=(1000, 700), link=:x)

Observe how the tank’s water level responds to rainfall events. Note any periods when the tank runs dry or overflows.

### Reliability Analysis

We can run simulations for all years, one at a time, and then check how many times the tank runs dry. You can run simulations for all years at once with the following code:

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: 2 out of 49

The system’s reliability percentage was about 96%, meaning that failures occurred in 4% of the years (or 2 out of 49 years). This means that with the parameters I selected, there was a relatively low amount of failures and the system was very reliable. It is notable, however, that the same number of failures occurs with a lower runoff coefficient and the default tank size, suggesting that at some point, increasing the tank size does not help the efficiency of the system, as not enough water can reach the tank for it to be necessary.

### Identify a Gap in the Methods

1. One limitation in the current model is that it assumes constant usage across all households and for all days of the dry season. This does not realistically reflect real-world water use trends.
2. In order to address this gap, we would need to find a way to more dynamically model water usage across households and across the dry season. For example, we could use historical water usage data across several households and craft a more realistic and dynamic model based on household size, time of year, etc.
3. This would enhance the analysis by allowing us to better determine how much water was really needed and therefore more accurately reflect how many failures there really were. For example, there could have been times where our model showed no failures but in the real world, there was a higher demand and usage for water than 74.1 liters per day, which could have resulted in more failures than we thought.