CEVE 101: Project 03

Elise Trani (et61)

2024-10-23

```julia
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

Station 25

```julia
filepath = "data/25.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.

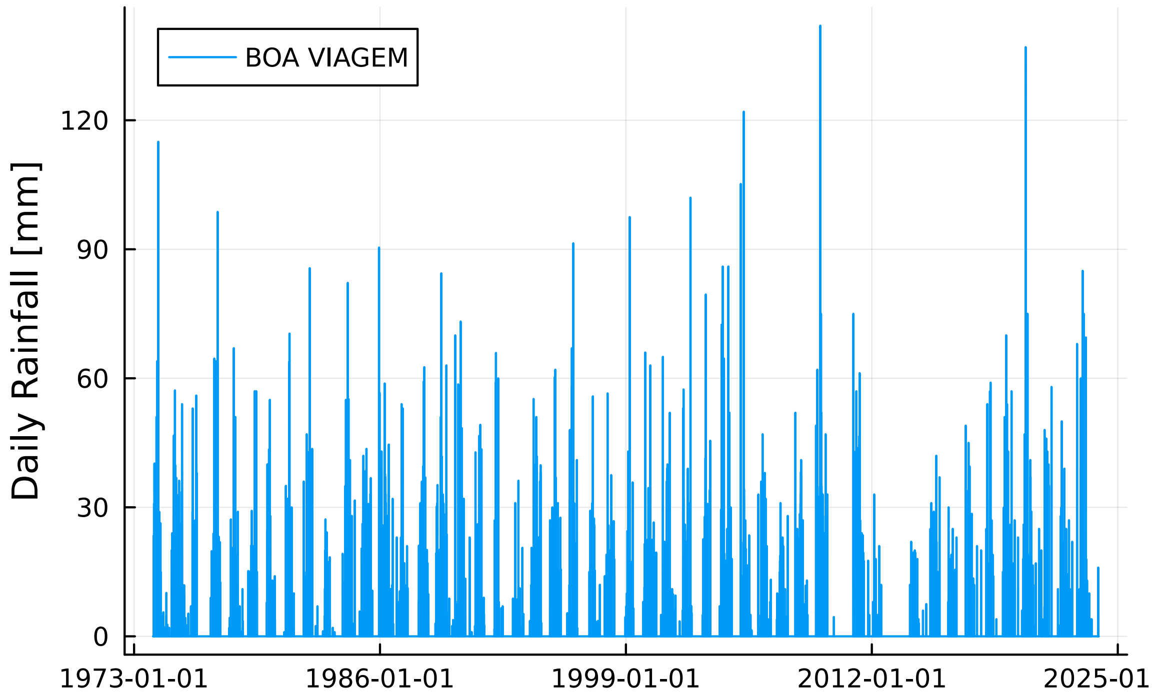
```
RainfallData for station: BOA VIAGEM
  Location: -5.1248611111111, -39.732472222222
  Years of data: 48
  Total days: 17531
```

Plotting the Data

```julia
plot(rainfall_data)
```

Line 1

This `plot` function is defined in viz.jl and is a thin wrapper around the Plots.jl package.



Discussion

After cleaning, data from 48 years were kept. This means that there are 48 years of valid data. Several years around 2011-2014 don't contain any data due to the major drought from 2012-2017 in Ceará Brazil. The missing data during drought years could lead to an overestimation of water availability and how reliable our system is. My model could also be slightly inaccurate because it would be based on data from when there was sufficient rainfall each year, and this could lead to a model that isn't equipped to deal with drought years. These are just some considerations I have to keep in mind. However, the available data is still sufficient for meaningful risk assessment because even without the years from the drought, there is enough data to make meaningful predictions.

Understanding the Theoretical Model

Let's explore the model that simulates the rainwater harvesting system.

Mass Balance Equation

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

$$V_{t+1} = V_t + Q_t - C_t - E_0 \mid 0 \leq V_t \leq V_{\max}$$

This equation is a very simplified version of the water levels and has some limiting assumptions and approximatinos. It doensn't take into account the evaporation that happens in the tank since it only calculates the volume of water added minus the volume of water consumed. This equation makes the assumption that there is no evaporation, which we know is not true. Evaporation is an important factor that need to be taken into account when calculating how much available water there will be. This equation also makes the assumtion that the material of the tank doesn't absorb any water, and the tank doesn't leak any water. For now, we'll focus on modifying the equation to take evaporation into account since we are not 100% sure there was any leakage or absorption. I would add the Penman formula, which estimates evaporation rates of water based on several key factors. To do this, we'll use the Penman formula:

$$E_0 = (700T_m/(100 - A) + 15(T - T_d))/80 - T$$

- E_0: is the evaporation rate
- T_m: T+0.006h, and h is the elevation in meters
- T is the mean temperature
- A is the latitude (degrees)
- T_d is the mean dew-point
- Now that evaporation is added to the model, better estimates of how much water is actually in the tank at a given time can be made.

Inflow (Q_t)

Inflow is calculated as

$$Q_t = \eta \times A \times \max(P_t - F, 0).$$

Consumption

Consumption is modeled as:

$$C_t = \begin{cases} 74.1 & \text{day of year} > 150 \\ 0 & \text{day of year} \leq 150 \end{cases}$$

Model Implementation and Testing

Defining the Model Parameters

We can define the model parameters with sensible default values.

```
param = ModelParameters(
    runoff_coefficient=0.85,
    roof_area_m2=80.0,
    first_flush_mm=2.0,
    tank_capacity_L=18000.0
)
```

Line 1

This creates a `ModelParameters` object with the specified parameters.

Line 2

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

Line 3

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

Line 4

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

Line 5

The tank capacity ($V_{\max}$) is the maximum volume of the tank.

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

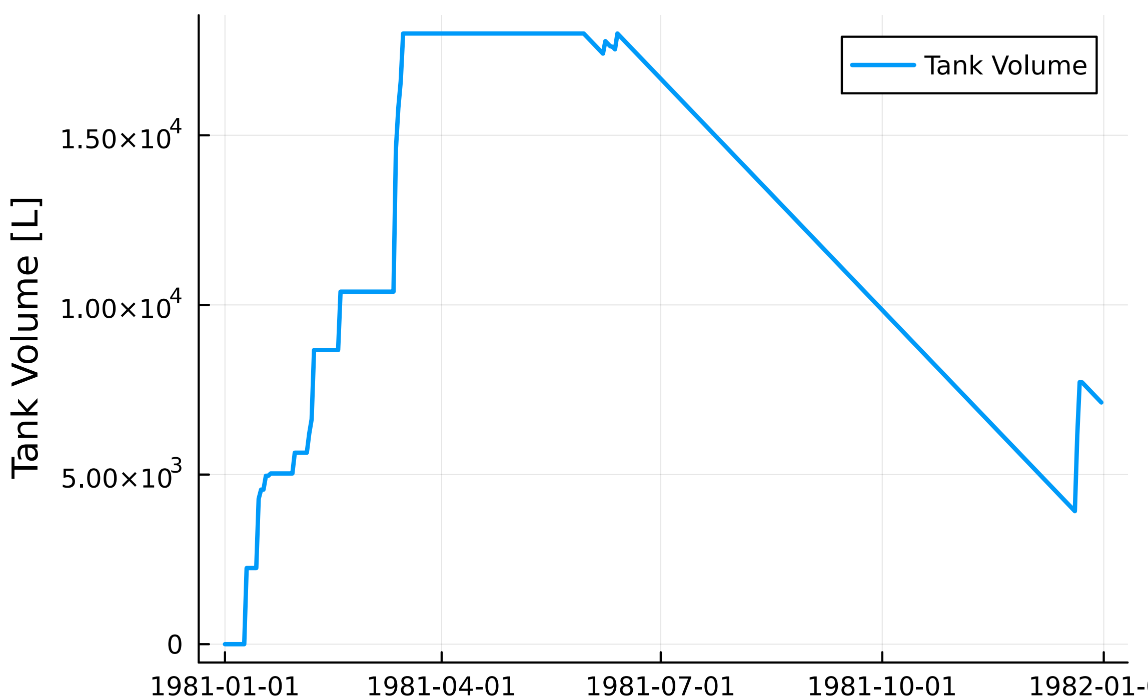
We select the annual rainfall data for the year 1981.

Line 2

We run the simulation for the year 1981.

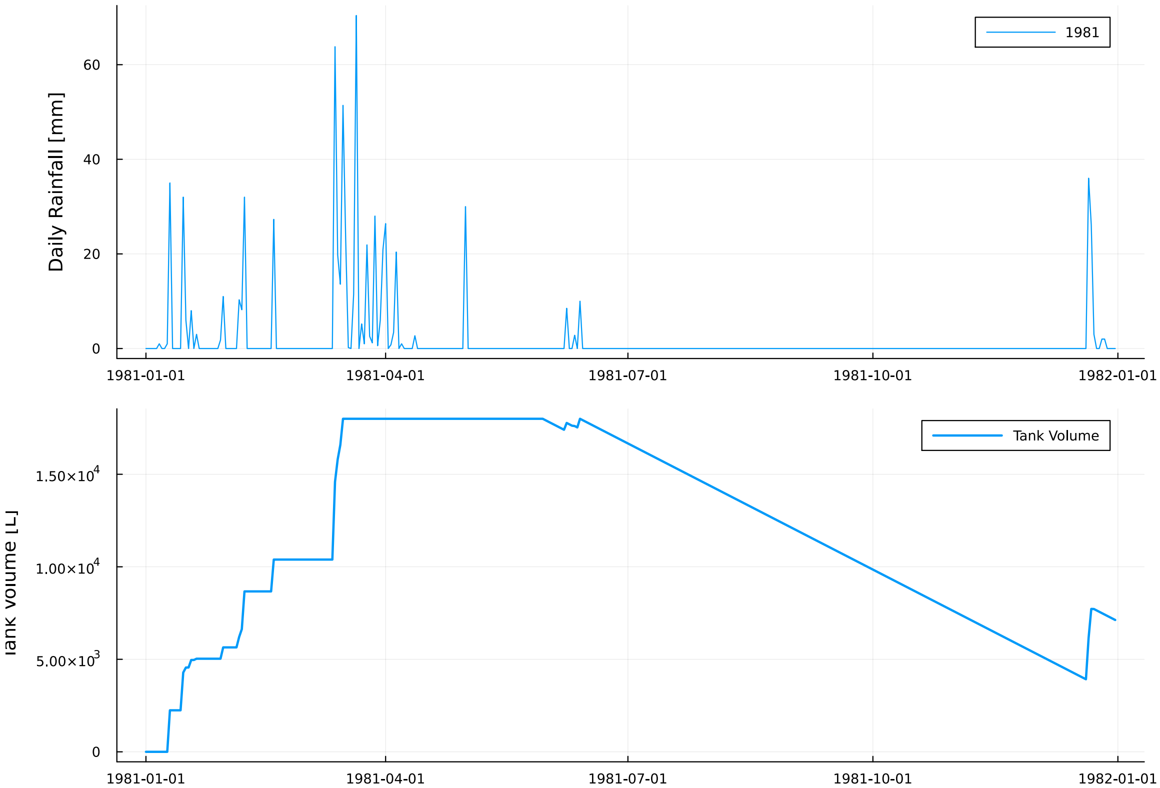
Line 3

We plot the results, again using a `plot` function defined in viz.jl.



To compare rainfall and tank volume:

```
p2 = plot(rainfall_1981)
plot(p2, p1, layout=(2, 1), size=(1000, 700), link=:x)
```



I increased the roof area and tank capacity. First, increasing the roof area helped with keeping more water in the tank in at the start of the year when there wasn't much rainfall. The only point in 1981 where there wansn't water in the tank was at the beginning of the year, but this was due to an uncontrollable factor: lack of rainfall. I increased tank volume to help retain water from the increase in roof area, plus it helps keep water levels from becoming too low in the second half of the year where rainfall is more sparce. These changes will increase they system's reliability, and it will decrease the number of years with failures.

Reliability Analysis

We can run simulations for all years, one at a time, and then check how many times the tank runs dry. Since 3 out of 48 years had failures, the system's reliability percentage is 93.75%. This is a very high percentage, meaning it is unlikely the system will fail/the system is trustworthy.

```
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: 3 out of 48
```

Limitations

A current limitation of this model is the graph of the data for tank volume and rainfall. It looks as though there are vertical lines in the graph, when these are actually rapid spikes in the amount of water in the tank. This could be quite confusing to some people, so one way to adress this could be by adding a notice near the graph that explains these areas are actually rapid increases. This would enhance the analysis by making the graph clear and easy to read/interpret.