# Firework Injury: Markov Model

## Introduction and Overview of Decision Problem

This case study will build on the initial analysis of firework-related injuries in Columbia using a decision tree model. We will extend our evaluation by employing a Markov cohort model. While the decision tree provided immediate outcomes and costs for various strategies, the Markov model offers significant advantages for examining long-term health and cost impacts.

Our Markov model will allow us to capture the progression of firework-related injuries over an extended period, allowing us to consider long-term implications. By modeling injury events and transitions between different health states over time, we can better understand the chronic effects of injuries and the long-term benefits of intervention strategies.

This approach will provide a more comprehensive assessment of the lifetime health and economic impacts of the proposed strategies, considering factors such as long-term healthcare costs, changes in injury rates, and compliance levels. The Markov model's ability to incorporate these extended horizons and recurring events will yield deeper insights into the most effective and sustainable approaches to mitigating firework-related injuries in Columbia.

## Alive-Dead Model

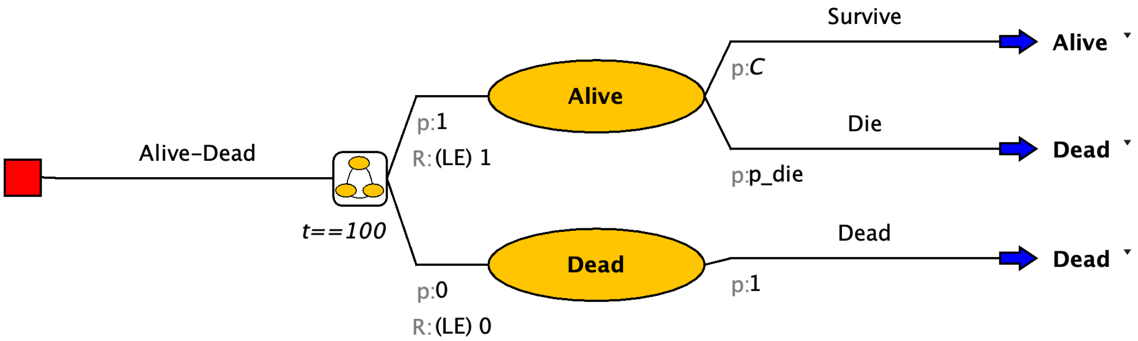
We will start by constructing a simple Markov model to represent the progression of firework-related injuries in Columbia. Our initial model will consist of two health states: "Alive" and "Dead." We will assume that individuals can transition between these states based on the probabilities of surviving as calculated using life table data.

### Markov Structure: Alive-Dead

The structure of the Markov model for the Alive-Dead model is shown in the figure below. Amua has a special Markov node (represented by ). The branches that lead off a Markov node designate all the Markov states (and only Markov states).

In this example, there are two health states: (1) Alive, (2) Dead. Off each health state, you can create a subtree (also called a cycle tree) that reflects those events that can occur during a cycle. The last branch at the end of each pathway will be a state transition, which defines what state to go to for the next cycle.

> **i** Note
>
> Note that in a Markov model, outcomes are defined elsewhere – NOT at the end of the branch, but at the state.

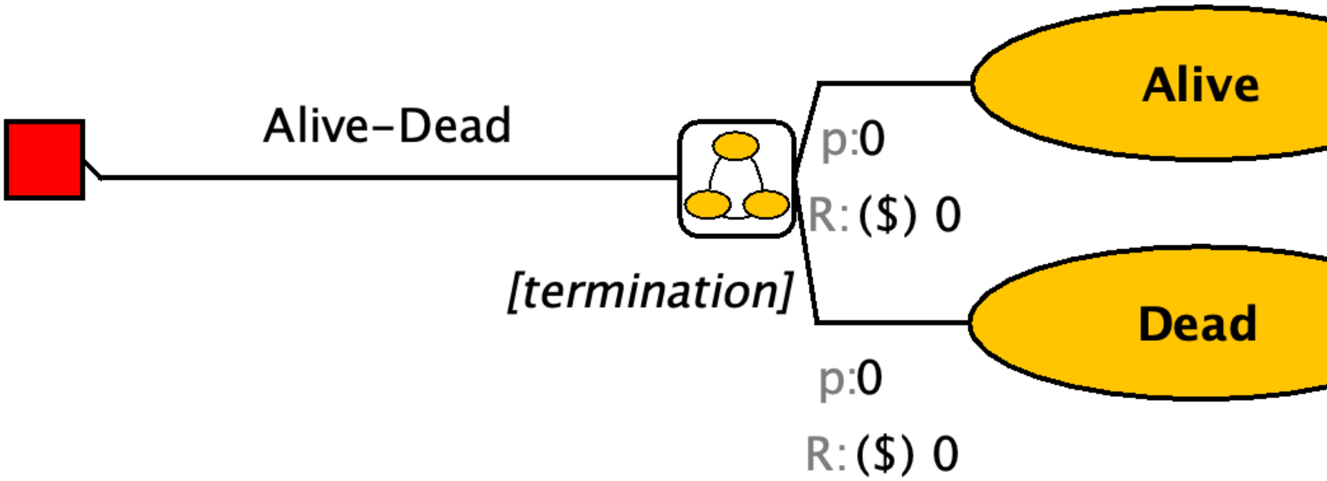


## Building the Tree

### Structure

After you open Amua, click Model → New → Markov Model.

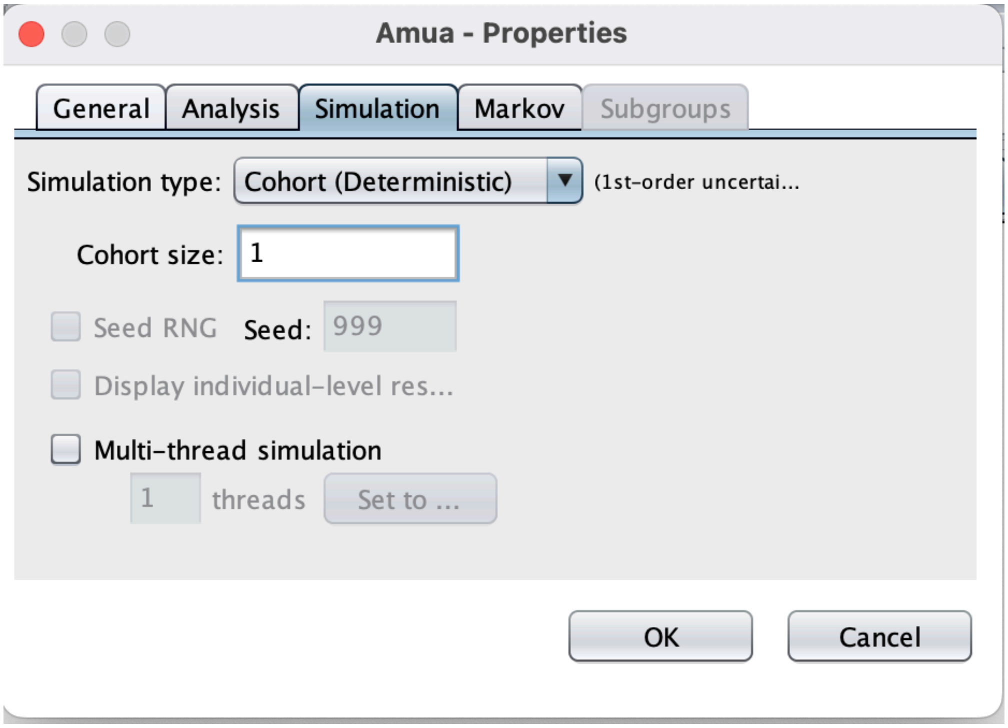
- Save your model right at the start.
- Now select the decision node , Right-click → Add → Markov Chain .
- Start by developing the structure of the Markov model using Alive and Dead for the different health states. The branches of the Markov chain should correspond to the states of the model. Label the name option to the right of the decision node as Alive-Dead.



- Now, complete the structure of the Markov model using the information above. Note: focus on adding the branches and transitions for this step; Parameters will be added later in this lab.
- When you reach the end of the branch, select the chance node you would like to turn into a state transition, right click, select Change to State Transition. This will give you the blue arrow . On the right of this arrow, you can find a dropdown menu with the different health states you specified. Select the health state this part of the cohort will transition to.
- With this button you can align the end nodes.

### Cohort size and starting probabilities

- Go to Model → Properties → select the Simulation tab. The default cohort size is 1,000. Change cohort size to 1, so that all outputs will be at the individual level (i.e., per person).



Simulation type: Cohort (Deterministic) (1st-order uncertai...

Cohort size: 1

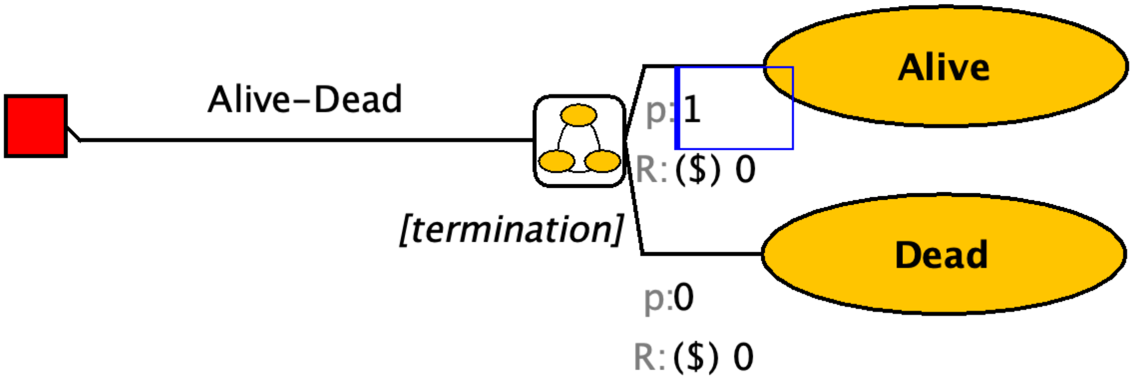
□ Seed RNG Seed: 999

□ Display individual-level res...

□ Multi-thread simulation

1 threads Set to ...

- At the Markov Chain , change the initial probabilities of health states. Since all patients start in the Alive state, change the p:0 on the left of the Local state to p:1 (see blue box below).



### Transition Probabilities

The underlying transition probability matrix has the following form:

$$\begin{array}{lcc} & \text{Alive} & \text{Dead} \\ \text{Alive} & 1-\texttt{p\_die} & \texttt{p\_die} \\ \text{Dead} & 0 & 1 \end{array}$$

However, because we are modeling a cohort of newborns until death, we need to include death transition probabilities that vary by age (i.e., cycle) in the model. In other words, there will be a different death probability in each cycle.

! Important

In Amua, **time-varying inputs are operationalized as variables** rather than parameters.
Variables can be defined to keep track of model events and dynamically update expressions as the model runs. Thus, unlike parameters which are fixed for a given run of the model, variables can change within a simulation. Variables may change across individuals, allowing heterogeneity to be modeled, or they may change over time.

Therefore, we will define `p_die` as a **variable** that references values in a **lookup table** , following the steps below.

! Important

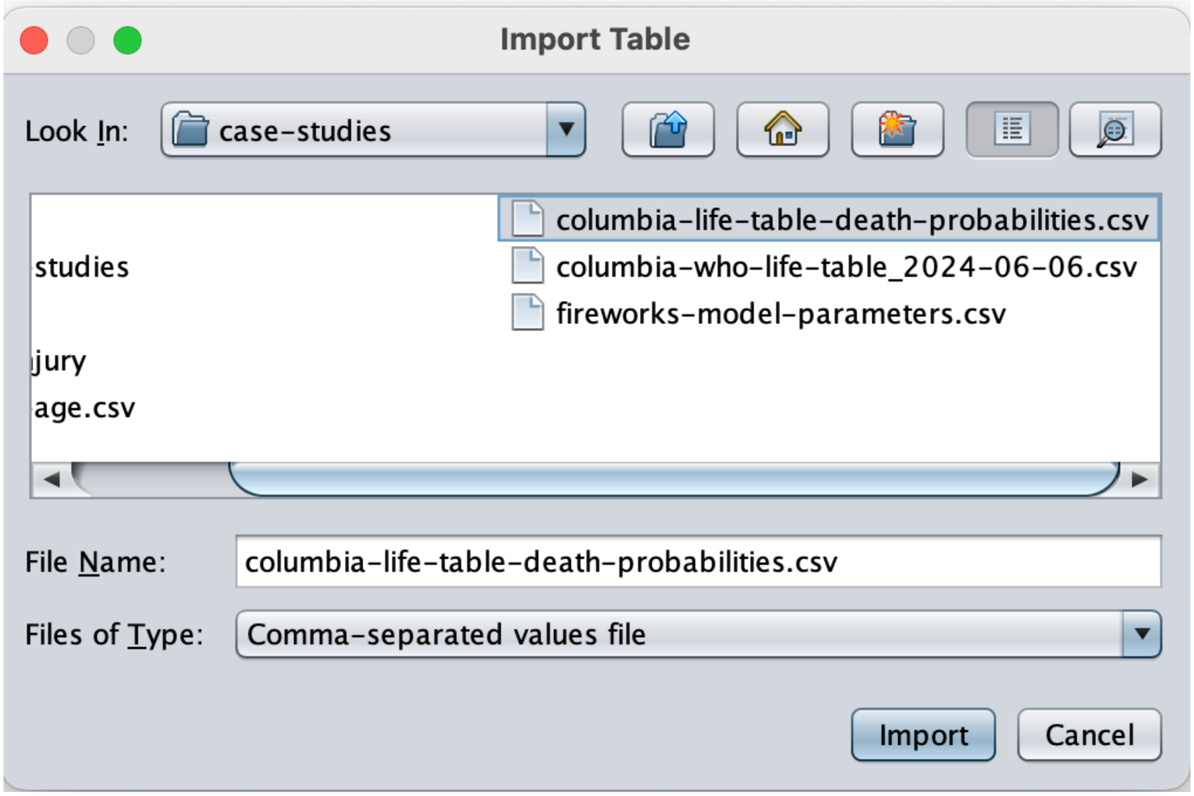
A lookup table returns the value that corresponds to a particular index. The first column in a lookup table contains the table indices, and one or more columns of lookup values can be defined. Index values must be unique and in ascending order. There are 3 lookup methods, (1) exact, (2) interpolate, and (3) truncate, for details about these methods visit the Amua Wiki on GitHub.

We first define a lookup table tbl_death to read in the age-dependent background mortality constructed from the Columbian life tables.

### Download Data

Download the file titled `columbia-life-table-death-probabilities.csv` from the workshop website (link below). This table contains the age-indexed annual mortality probability for Columbia.

Go to the “Tables” panel, click to add a table, and you should see a “Define Table” window (see screenshot below). Click the Import button and select the downloaded .csv file to import. The table will automatically be resized to fit the imported data. The first row in the file will be used as table headers.



Import Table

Look In: case-studies

studies
jury
age.csv
columbia-life-table-death-probabilities.csv
columbia-who-life-table_2024-06-06.csv
fireworks-model-parameters.csv

File Name: columbia-life-table-death-probabilities.csv

Files of Type: Comma-separated values file

Enter “tbl_p_Die_other” as the name for this variable. “Lookup” is the default table type (which is what we wanted for this life table). Choose “Interpolate” as the Lookup Method. Click Save to apply.

Then, we can define the time-varying variable p_Die_other (which reads from the table we just defined and operates as the background mortality probability in the current model cycle, t). Make sure you create a variable and not a parameter. Under “Variables”, click to add a variable p_Die_other = tbl_p_Die_other[age_initial + t, 1]. (See screenshot below.)

The first column of a table is always used for indexing in tables. Therefore, the number “1” here will actually indicate the second column in your table as the corresponding value. You can also use the name of the column to read a value, e.g., tbl_p_Die_other[age_initial + t, “p_die”].