Center for Health Decision Science
CHDS
Harvard T.H. Chan School of Public Health

Resource Pack: Amua Tutorials

Citation: Building Decision Trees Tutorial. Resource Pack: *Amua Tutorials.* Center for Health Decision Science, Harvard T.H. Chan School of Public Health 2019.

TUTORIAL OBJECTIVE
This tutorial illustrates the basic steps needed for developing decision trees in Amua using a disease screening example. It details the process of building a decision tree, saving the model, and running baseline and sensitivity analyses.

Opening Amua

Follow the Getting Started instructions (https://github.com/zward/Amua/wiki/Getting-Started) to download Amua and open the program.

Example Overview

In this example problem, you must decide whether or not to implement a screening program for a certain disease (Dx). Assume that persons testing positive will be treated, which improves their life expectancy. For disease-positive individuals who are not screened, assume that treatment will be delayed until they become symptomatic and that they have a lower life expectancy than those who receive immediate treatment. For disease-negative individuals who are inappropriately treated, assume that there is a decrease in their life expectancy due to drug toxicity. The screening test is imperfect with a sensitivity of 0.97 and a specificity of 0.9988. The prevalence of the disease in this population is 0.05. The objective is to maximize life expectancy. As alternative strategies, you could not screen anyone, or you could simply treat everyone without screening first.

General Notes

Your decision tree will look quite similar to those drawn by hand. Decision trees consist of a series of nodes connected by branches. Each branch will be assigned a descriptive name. There are no restrictions on what branches can be named in Amua, and branch names do not have to be unique in the tree. However, any model objects defined (i.e. parameters, variables, tables, constraints) need unique names without spaces, but underscores '_' are allowed.

Node Types

■ **Decision nodes:** Used to represent choices available to the decision maker.
● **Chance nodes:** Used to define alternative chance events.
◀ **Terminal nodes:** Used to define the payoffs associated with outcomes of interest.

Building the Decision Tree

The completed decision tree we will build in this tutorial is shown below:
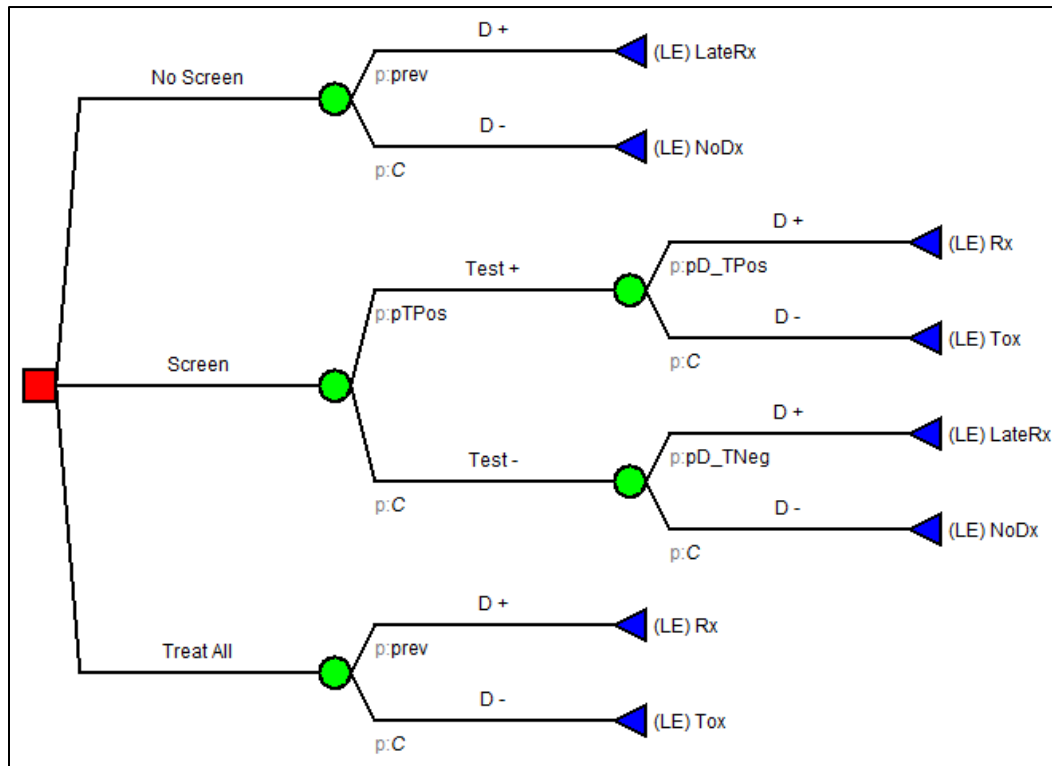
Once you have opened the program, using the top menu, navigate to **Model** -> **New** -> 🔲 **Decision Tree**. This will create a new model with a single decision node 🟥 on the left of the screen.

Adding Branches

To add branches to this decision node (the root of the tree) you have a few options:

- Double-click: Double-click on the node to add two branches with chance nodes 🟢. Note that if the node already has branches, double-clicking will add one branch.
- Right-click: Right-click on the node to display the pop-up menu. Select **Add** -> and choose the node type you wish to add.
- Toolbar: Select the node and click on a node type in the toolbar at the top to add it to the tree.

For the example in this tutorial, double-click on the root decision node 🟥 , which will add two branches.

Labeling Branches

New branches are called **Name** by default. To give branches more informative labels, click on **Name** above the branch. A text-box above the branch will appear outlined in **blue**. You can now edit the text that appears in this field. Label the top branch **No Screen**. You can hit Enter on the keyboard or click outside of the text-box on the canvas to accept the new label. Now, click the label above the lower branch and rename it **Screen**.

Adding Terminal Nodes

Double-click on the **No Screen** chance node ● to add 2 branches. Click the **Optimize Current Display (OCD)** button on the toolbar to arrange the branches and ensure even spacing. Right-click on each of the child nodes ● and select ◀ **Change to Terminal Node**. You may also change node types on the toolbar ⬚ or add a new terminal node ◀ using the toolbar or pop-up menu.

Name the top branch **D +** and the bottom branch **D -**. Your tree should look like this:
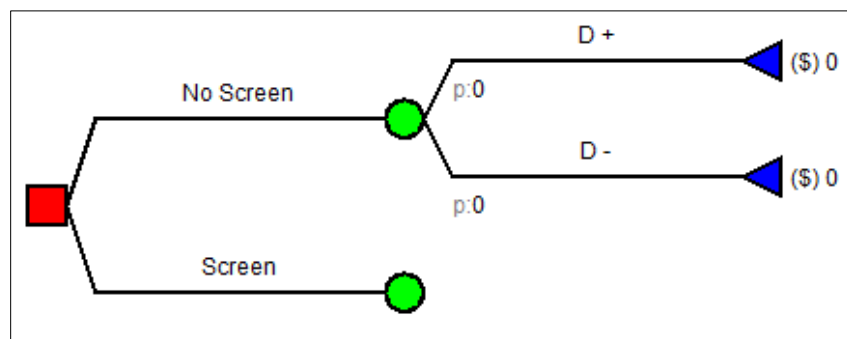

Complete the Structure

You can continue adding nodes to complete the structure of the tree using the toolbar, pop-up menu (right-click), or double-clicking.

You can also copy and re-use parts of the tree that have already been built by selecting the **No Screen** chance node ● (it will be highlighted in **red**). Copying a parent node will also copy all of its children. Right-click on the **No Screen** node and select ⧉ **Copy**. Now select the **Screen** chance node ● and paste the sub-tree (right-click and click ⧉ **Paste**). Copy/paste can also be performed using the **Edit** menu, or keyboard shortcuts (Windows: Ctrl-C, Ctrl-V. Mac: Cmd-C, Cmd-V). Rename the new branch **Test +**.

Select the **Screen** chance node ● and paste the same sub-tree (right-click and click ⧉ **Paste**). Note that nodes can overlap in Amua, so click **OCD** to ensure you can see all of the nodes in the tree. Rename the new branch **Test -**.

Lastly, select the root decision node ■, paste in the same sub-tree, and rename it **Treat All**.

The structure of the tree is now complete and should look like this:
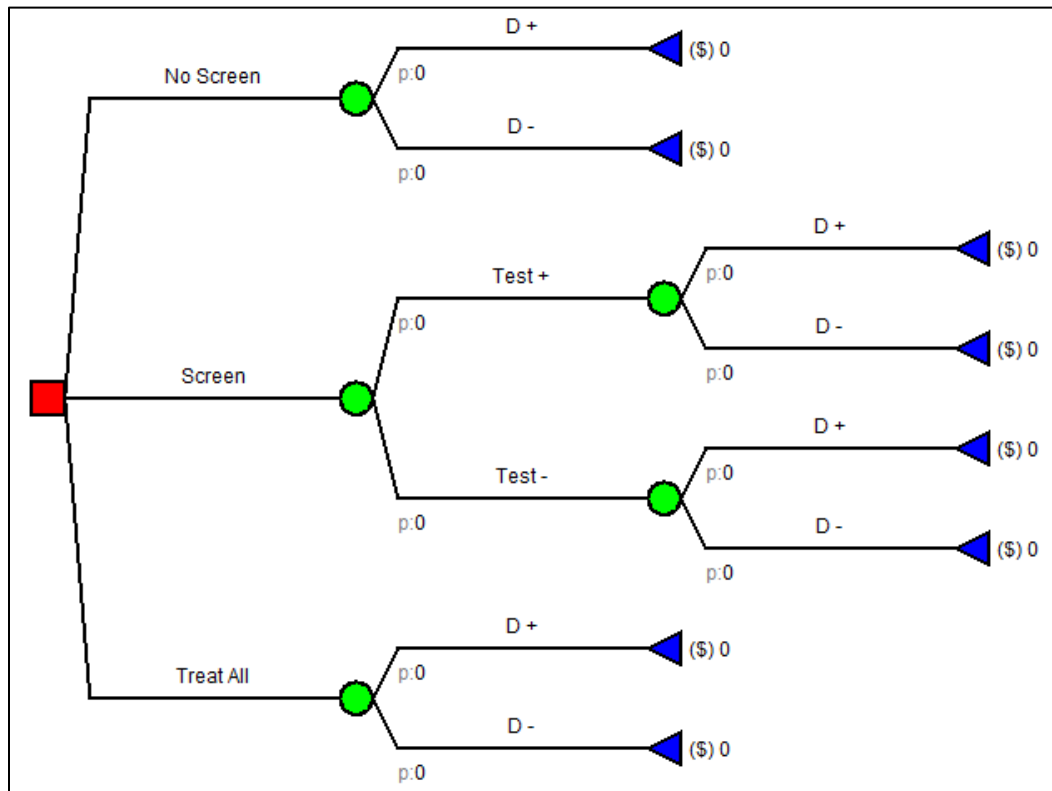
Removing Nodes

Nodes and sub-trees can be removed by selecting the node/root of the sub-tree and pressing `Delete` on the keyboard, or ✖ **Delete** from the pop-up menu.

Nodes/sub-trees can also be cut and pasted elsewhere using the **Edit** menu, pop-up menu, or keyboard shortcuts (Windows: `Ctrl-X`, `Ctrl-V`. Mac: `Cmd-X`, `Cmd-V`). Alternatively, you can always ↺ **Undo** or ↻ **Redo** actions using the **Edit** menu or keyboard shortcuts (Windows: `Ctrl-Z`, `Ctrl-Y`. Mac: `Cmd-Z`, `Cmd-Y`).

Saving the Model

Now that you have the model structure defined, this would be a good time to save your work. You can save at any time by clicking on **Model** -> 💾 **Save** or using the keyboard shortcut (Windows: `Ctrl-S`. Mac: `Cmd-S`). When saving for the first time, a dialog box will appear asking for the name of the model. Choose the location where you want to save the model, enter **Dx_Screen** as the file name, and click **Save**.

Defining Model Outcomes

By default, model outcomes (called model 'dimensions' in Amua) track **Costs**, denoted by the **$** symbol inside parentheses. In this model, our outcome of interest will be **Life Expectancy**. To add other outcomes, or

4

change the outcome of interest, go to the menu bar and click **Model** -> ⚙ **Properties** and select the **Analysis** tab. In this tab, rename the dimension from **Cost** to **Life Expectancy**, change the symbol to **LE**, and choose 2 decimal places for precision. Make sure that you hit `Enter` once you change the value in a table cell. The cell border will turn light blue once the cell is updated.

Click **Refresh**. In the table below change the Objective to **Maximize** and the Outcome to **Life Expectancy**.

Your screen should look like this:

Click **OK**. Your model outcome symbols should now be **LE** in the tree.

Defining Probabilities

Now we will assign probabilities to the branches of chance nodes 🟢.

For the branch **No Screen**, click underneath the branch labeled **D +** (next to the label **p:**). A text-entry box will now be outlined in blue. This is where you enter the probability for this branch. Type **prev** which we will define as the prevalence of the disease. Hit `Enter` or click outside the text-box to accept the probability. The text will now turn **red** because Amua does not recognize **prev** as a model object, so we will have to define it later.

Now enter the probability for the **D -** branch for **No Screen**. Because the probabilities for all branches of a chance node must sum to 1.0, we could enter **1-prev** as the probability. This is called a 'complementary' probability as it provides the complement to sum to 1.0. In Amua you can type **C** or **c** to indicate a complementary probability. Note that there can only be one complementary probability per chance node. Here enter **C** for the probability of **D -**.

Now, define probabilities for the **Screen** strategy. For the **Test +** branch enter **pTPos** as the probability of testing positive, which will be a function of the disease prevalence and test characteristics (sensitivity and specificity). Enter **C** for the probability of **Test -**.

5

Now, define the probability of having the disease given each test result in the **Screen** strategy. On the **Test +** branch, click under the **D +** branch and type **pD_TPos**. This will be the probability of having the disease given a *positive* test. Click under **D -** and type **C** to indicate a complementary probability. For the **Test -** branch, click under the **D +** branch and type **pD_TNeg**, which is the probability of having the disease given a *negative* test. Click under **D -** below and type **C** again.

Lastly, define the probability of having the disease in the **Treat All** strategy. Click under **D +** and type **prev**, which is the prevalence of the disease. Enter **C** for the probability of **D -**.

Your tree should look like this:
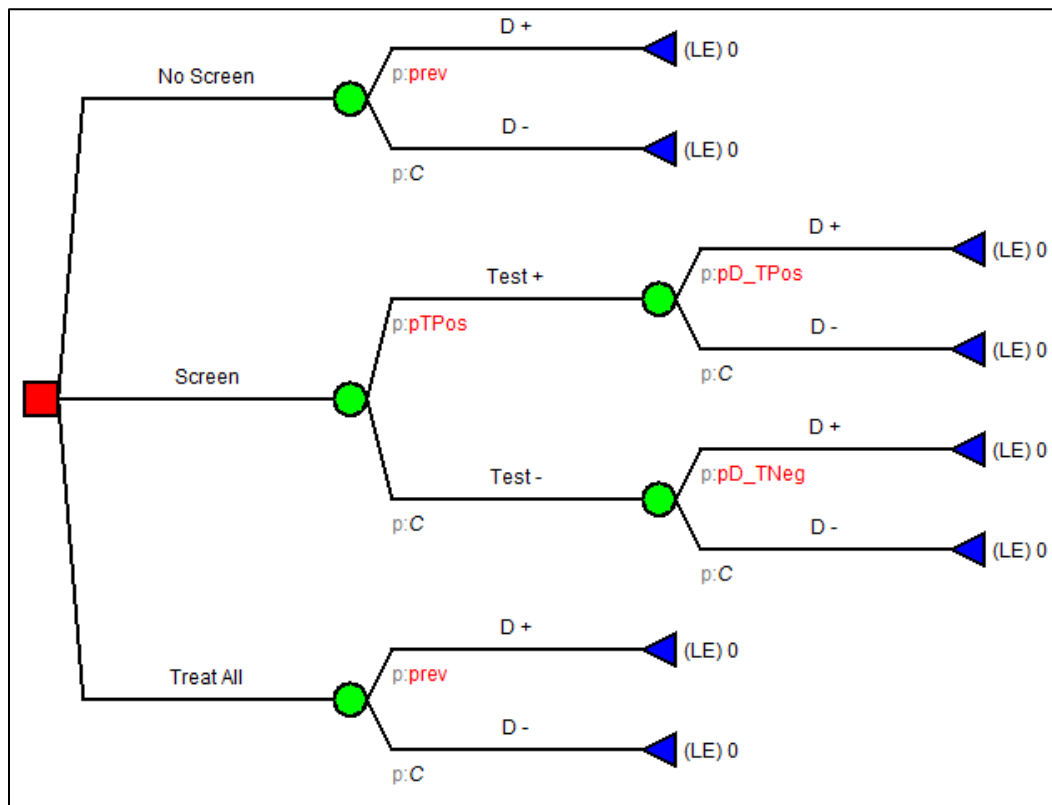
Defining Payoffs

Now we will define the payoffs for each terminal node to model the consequences of each possible outcome. To enter a payoff, click to the right of the terminal node. A text-box will appear outlined in **blue**. By default, the payoffs are set to 0, so we need to update these values.

In the **No Screen** strategy for **D +**, enter **LateRx** as the payoff, which we will define as the life expectancy for people who receive treatment late. For **D -**, enter **NoDx** as the payoff, which will be the life expectancy for people without the disease.

In the **Screen** strategy, only people who test positive will be treated. On the **Test +** branch, enter **Rx** as the payoff for **D +** to indicate that these people will be treated. For **D -,** enter **Tox** as the payoff since these people are false positives and are negatively affected by the treatment toxicity. On the **Test -** branch, enter **LateRx** as the payoff for **D +** since these people have the disease but were not identified by the screening test. For **D -,** enter **NoDx** since these people are true negatives.

Everyone will be treated in the **Treat All** strategy. This is good for those with the disease, but not for those without the disease. For **D +,** enter **Rx** as the payoff and for **D -** enter **Tox**.

Your completed tree should look like this:
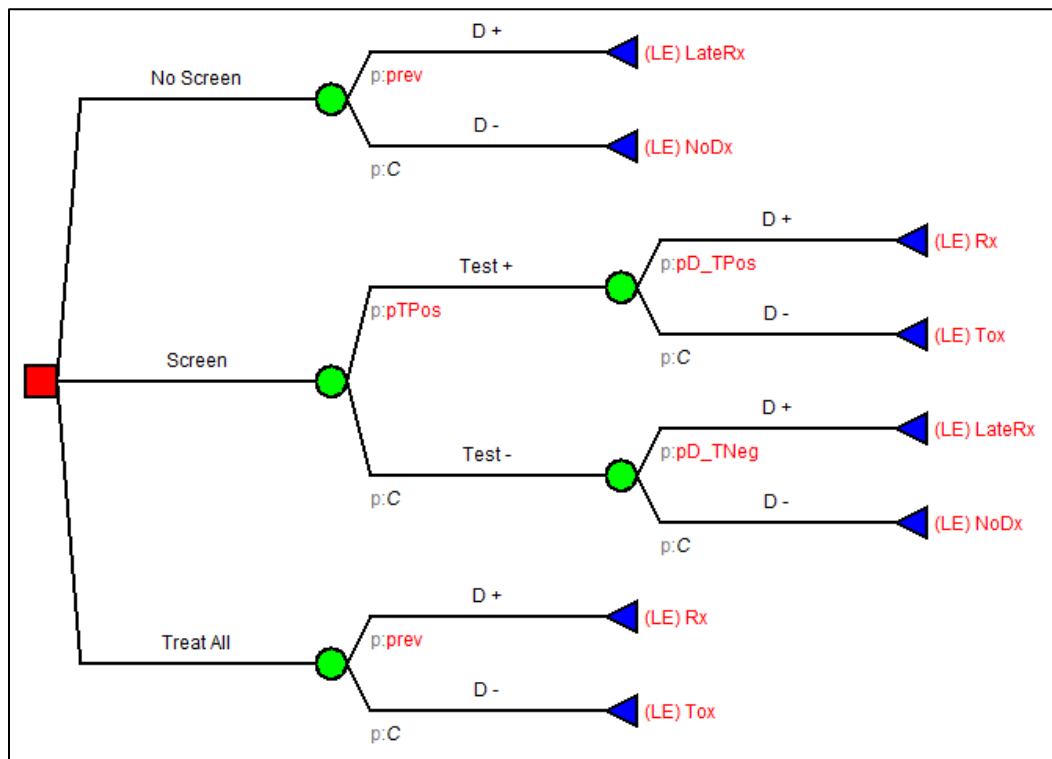
Defining Parameters

We've entered the names of all of the probabilities and payoffs in the tree, but we haven't yet defined the values for these parameters. We could have entered numeric values directly in the tree, but defining them as Parameters is more flexible and lets us perform sensitivity analyses later. Parameters are global in the model, which means that they can only be defined once and have the same value wherever they are used in the model.

On the object sidebar on the right, select the $\theta=$ Parameters tab. Click the ➕ Add button to create a new parameter.

First, we will define the prevalence of the disease in the population of interest. In the Name field, type **prev**. In the Expression field type 0.05.

Your screen should look like this:

Click **Save**. The parameter **prev** will now appear in the object sidebar and the probability of **D +** in the **No Screen** and **Treat All** strategy will now be black since we have defined this parameter.

Now we will define the screening test characteristics. First, we will define sensitivity by creating a parameter called **sens** with a value of `0.97`. Next, define specificity by creating a parameter called **spec** with a value of `0.9988`.

Based on the disease prevalence and the test characteristics, we can now use Bayes' theorem to calculate the probability of testing positive, and the probability of having the disease given a positive or negative result.

First, we will define the probability of testing positive as a parameter called **pTPos**. Enter the following expression to calculate the probability:

$$prev*sens+(1-prev)*(1-spec)$$

Notice that parameters will turn **blue** when referenced. You can hover the mouse over any colored text in Amua for more information. Click the **Evaluate** button or hit `Enter` to see what this formula yields. We see that with the current prevalence and test characteristics the probability of testing positive is `0.04964`.

Click **Save**.

Define **pD_TPos** (the probability of having the disease given a *positive* test) as a parameter with the expression:

$$prev*sens/(prev*sens+(1-prev)*(1-spec))$$

Define **pD_TNeg** (the probability of having the disease given a *negative* test) as a parameter with the expression:

$$prev*(1-sens)/(prev*(1-sens)+(1-prev)*spec)$$

Lastly, let's define the life expectancy payoffs associated with each outcome. Define these as follows:

Parameter	Value
LateRx	25.2
NoDx	40.3
Rx	35.8
Tox	39.4

Running the Model

While developing models you can click ✔ **Check Model** on the toolbar at any time to highlight any errors found. When you are ready to run the model you can click ▶ **Run Model**, which will evaluate the model and display the results on the canvas and the console.

For this model we get the following results:

Strategy	Life Expectancy
No Screen	39.54
Screen	40.06
Treat All	39.22

You can click the ❌ **Clear** button to clear the expected values for each branch from the canvas.

Sensitivity Analysis

Let's perform a one-way sensitivity analysis on prevalence. Go to **Run** -> **Sensitivity Analysis** -> ↔ **One-Way**.

For the parameter **prev** enter **0** as the Min value and **1** as the Max value. Make sure the row with **prev** is selected (highlighted in **blue**) and click **Run**.
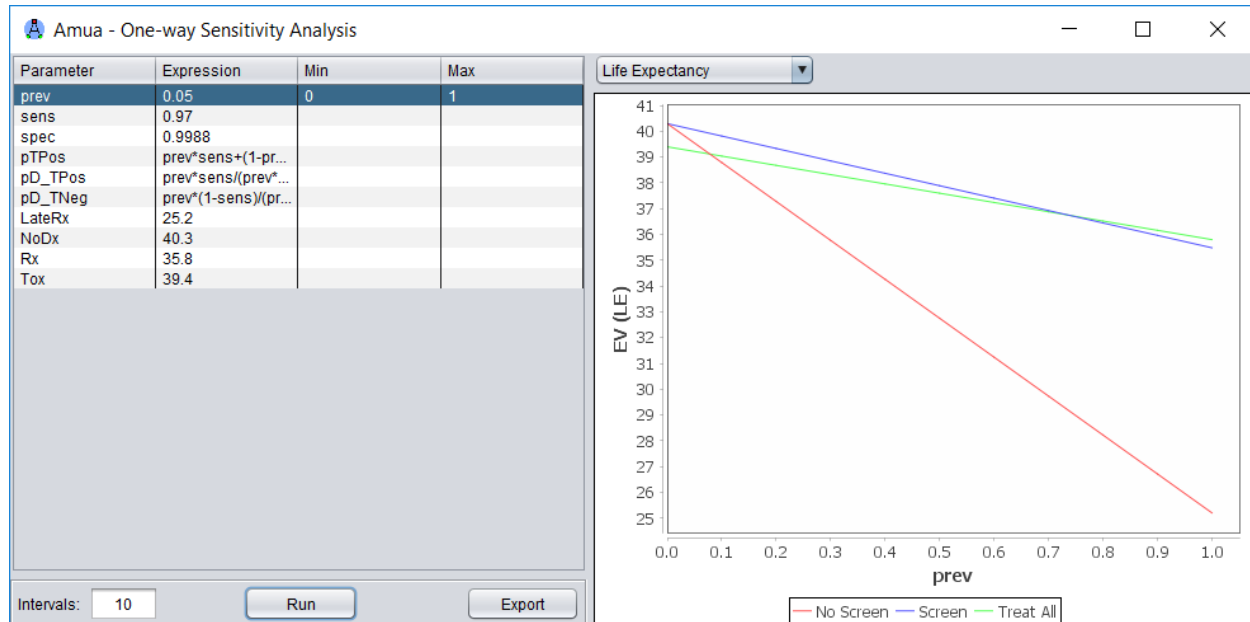
Saving a Graph

To copy the graph, right-click on the graph and click **Copy.** This will copy the graph to the clipboard so it can be pasted into other programs. To save the graph as a PNG, click **Save As** -> **PNG**. Lastly, you can **Export** the underlying chart data to a .csv file so that you can plot it using another program (e.g. R or Excel).

Saving the Model as an Image

Click the 📷 **Screenshot** button on the toolbar to copy an image of the current canvas to the clipboard so you can paste it into another program. Alternatively, to save the canvas as an image (PNG, GIF, JPEG), click **Model** -> **Save As**.