Feedback group 4

Hi group 4, nice job! We gave feedback on your code and the decisions you made, we hope it is clear and that you can use our suggestions.

**General**

To introduce the reader to the code, it is a good suggestion to give the names of the sections that can be found in the further part of the code and add more pseudo-code. If you use different files for the different parts (e.g. data analysis, functions), it might help to add some code that describes what the file contains and where it is used for. Now some functions are provided in the main PSA code, and others are in a separate file. It would increase clarity and transparency.

**Data analysis**

You could use the function Descdist(), which gives a skewness-kurtosis plot, to see which distributions fit the data. Now you only used visual tests for some distributions to see if the right distribution is found for the data. However, your distributions of part 1 seemed to be correct to us. You provided nice and clear conclusions of the selected distributions.

You are trying to stratify your data using a function. A tip could also use the c() function, which returns a vector. Or data.frame(), creating frames, to collect variables that have the properties you want to select of the data.

The distributions you defined for the Tx1.time are not used in the bca\_startcode or the psa\_startcode, there you still use some of the before-defined parameters, such as 30 days if you are in event 1 or 4. In the first part, you created nice distributions for this, so you can draw a value from this distribution and use this as an input parameter for the function Tx1.time. Also, you have 4 events in your trajectory, but for the event minor complication, there is no time to event defined in your Tx1.time function. Last, you assume the time to a major complication in Tx2 is symmetrical to TX1. However, treatment 2 has a different time to event distributions, for example with other rate and shape parameters for gamma distributions.

You give a normal distribution for your costs. However, for the distribution of costs, use a log-normal or gamma distribution, as costs cannot be negative (and a normal distribution can give you a negative number as a result). You could take a look at lecture 3 for the explanation.

**Parameters**

The code would become more readable if the parameters are defined separately above the code. Now you just call them in the functions. A good example is the time spent in palliative care. You make a function for this parameter, while it is saving computation time is you just add: *palliative.time = 100* . Besides, it is nice to just have parameters in your functions. For example in the following code line of the function func.exp.costs you wrote this line:

total\_cost <- total\_cost + ((Tx1.Cycles+1) \* (278+256+194)) + ((Tx2.Cycles+1) \* (278+256+194)) --- it would be great to call the separate numbers, here the costs of the diagnostic tools, by a parameter and define the parameter above your code.

**Functions**

Although it is of course clear for us what each function is doing, it would be nice if the functions are provided with more pseudo-code. For example, if you want to have a look at it again in a few years, it is easier if you can directly read what each function is doing. This holds for a general introduction of the formula but also within the formula, as described in the example below:

An example is that for the function: Func.tx1cost you use the input parameter Tx1.complications. You created an If-statement with Tx1.complications ==1 , but it would be more readable if you can directly see what the 1 suggests, is it a minor or major complication. Also, you don’t clearly mention that position is equal to the event, which took us a while to find out.

**BSC simulation model and EXP model**

In both models, you call the function Tx1.Event with cycles, condition and response as input parameters. However, the parameters age and sex are also of influence on the next to be determined event. It would be nice to include those parameters in the model as well. For example in Tx1.Event you still use the previously defined probability of 3% of event death occurring. A suggestion is to fit a logistic regression model, to predict this probability of death (event ==1 in the data), based on sex and age of the patient. After this, you can draw a prediction value from this regression model in your Tx1.Event function to predict the change of dying for the patient with a certain age and sex. In order to do so, you need to include the input parameters age and sex of the patient now in the Tx1.Event function.

Step 2.2 looks pretty complicated because no comments of what you are doing are provided. For the multivariate Normal distribution, you could simply use the function rmvnorm(). You only included cycle 1, while the question is about cycles 1 and 2. Also, do not make use of the fact that the response of the patient is the same for each treatment cycle, as Tx1.Ci.Dx.Pet does not change, except when death occurs.

We plotted your bsc sim model, and the branches seemed to be correct. However, for part 2 of the assignment, you made a function Tx1.continue to see if the treatment is working and the patient should continue with this treatment. You did not implement this function in your exp trajectory in any way. So you don’t use the function and patients still continue treatment, even though they don’t respond. You can do this by making a new event, their timeout is zero and you go to follow up 1 if the 3 thresholds are greater than the tests of the diagnostic tests.

It is not clear how you implemented the costs correctly. You only calculate the costs in the trajectory for event 2, when a patient dies. However, you create three functions (func.dx1cost, func.dx2cost and func.dx3cost) to incorporate uncertainties of costs for the diagnostic tests. These functions are never used again or implemented in your trajectory.

For a better overview, you could count costs per cycle, as there are costs to start the treatment, per cycle, and for time. For instance with the mod “+” . Before you start branching you could add the basic costs for this cycle, for instance in the following way:

set\_attribute(key = "Tx1.Costs", mod = "+", value = costspercycle) %>%

Within the cycle you could give costs for the duration and event you are in, by for example with:

set\_attribute(key="Tx1.Costs", mod="+", value=function() Tx1.Time(get\_attribute(exp.sim, ".Tx1.Event.")) \* Tx1costperday)%>%

In this way, you get the time that is corresponding with that specific event and multiply by the including costs per day.

**Interpreting Cost-Effectiveness Outcomes**

To conclude your report, you could save your data in an R file instead of excel, use the function save(Nameofoutput, file =”filename.RData”). Make a cost and effectiveness analysis of the outcomes performed in R.