

Health Economic Modelling: Final Report

Group 4

Alex Bujorianu (s2451980) & Rusheel Mehra (s2344211)

06/04/2021

Process Description & DES Design

Describe and motivate the choices you made during data analysis steps, design of the DES model, the simulation results, and the interpretation of these results.

Simulation Model

We begin the process by building a simple simulation model of the best standard care. The best standard care (BSC) is the clinical pathway as studied in the clinical trial as seen in figure 1.

After diagnosis, the patient enters the first line of treatment, henceforth referred to as Tx1. In Tx1, the patient has to undergo 5 treatment cycles. The patient can follow four trajectories within this line of treatment. The first is when the patient has no complications and completes a full cycle. In this event the patient proceeds to the next cycle. The second trajectory is when a patient dies during a cycle. In such an event, the patient exits the system. The third trajectory is when a patient has a major complication. In this event, the patient exits Tx1 but continues into the first follow up and then through the rest of the pathway. The health and economic burden of a major complication is reflected in the qalys and cost functions respectively.

The fourth trajectory is when the patient experiences minor complications. Minor complications add health and economic burden but do not affect the patient’s path through the system.

The patients who survived Tx1 move on to the follow up period. In the follow up, the patient follows two possible trajectories, one where they live and one where they die. The patients who die in the follow up exit the system.

After the first follow up, the surviving patients enter the second treatment line, referred to as Tx2. Tx2 and the trajectories within it are identical to Tx1 and its trajectories. The patients who survive Tx2 go in to the final phase of the pathway, the second follow up, also known as palliative care. The final phase lasts exactly 100 days. We understand that the patients will all die at the end of this period (hence the term palliative care) but we record them as “alive” for modelling purposes—it helps us to distinguish between patients who died prematurely and those who survived the full cycle.

Competing Risks

One of the most important aspects of the model is the need to handle mutually exclusive events with different probabilities. During each cycle, there are three possible risks: death, major complications, and minor complications. Additionally, the code has to check whether the patient has completed the cycles, and—for the experimental model—whether the diagnostic tests indicate the patient should discontinue treatment on account of non-response. We decided to check for the most significant events first, based on the evaluation of the else if statements: death, major complication and then minor complication.

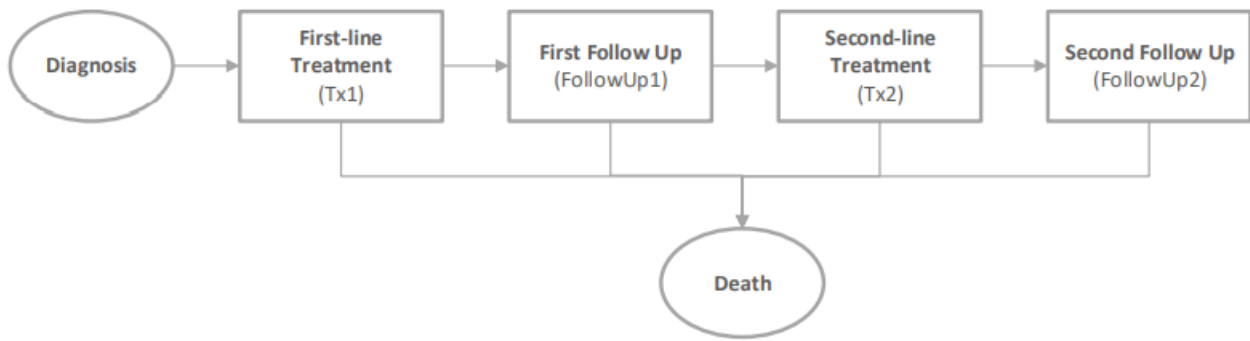


Figure 1: A graph of the clinical pathway the patients take in the model.

Note two things: if the patient has already undergone the full 5 cycles of treatment, they are taken out of the treatment cycle and complications due to treatment do not occur. Likewise, in the exp model, the code checks whether patients are responding *before* evaluating the complication probabilities (because complications won't happen if the patient is not treated unnecessarily).

The death if statement comes first because patients could in principle die on the last treatment cycle (the 5th one).

```

Tx1.Event.alt <- function(cycles, condition, response) {
  #about 10% chance of minor complication, 4% major, 3% death
  minor_comp <-
    if (condition==0 && response==1) {ifelse(beta_minor_good_response >= runif(1), 1, 0)} el
    if (condition==1 && response==1) {ifelse(beta_minor_poor_response >= runif(1), 1, 0)} el
    if (condition==0 && response==0) {ifelse(beta_minor_good_nonresponse >= runif(1), 1, 0)}
    ifelse(beta_minor_poor_nonresponse >= runif(1), 1, 0)
  major_comp <- ifelse(runif(1) < 0.04, 1, 0);
  death <- ifelse(runif(1) < 0.03, 1, 0);
  #First check to see if the patient is alive, then check for complications
  if (death == 1) {
    return(2)
  }
  #if the patient has survived the cycles, they are taken out of the simulation
  else if (cycles > 4){
    return(5)
  }
  #Stop the treatment if the patient is not responding, do not overtreat
  #We do these tests BEFORE the treatment cycle. So complications should not occur.
  else if (cycles > 0 && Tx1.continue(response)==FALSE) {

```

```

        return(5)
    }
    else if (minor_comp == 1) {
        return(4)
    }
    else if (major_comp == 1) {
        return(3)
    }
    else {
        return(1)
    }
}

```

Note: for the full code, and the results of our simulations, please see our Github page.

Data Analysis

We analyze the available dataset which consists of trial data, in order to determine the patient-level variation as well as second-order uncertainty in order to define parameters and distributions for various attributes of the patients.

Age & Gender: We analyzed the dataset to find the distribution of patients' ages based on their sex. Using the function `gofstat` and visual confirmation using the `plot` function, we found that the ages of male patients followed a lognormal distribution and the ages of the female patients best fit a Weibull distribution. Both distributions have a natural zero, i.e. the range of possible values drawn from these distributions is between 0 and infinity. This was ideal, since age cannot be negative.

Effectiveness based on condition: We also analyzed the dataset to determine if the effectiveness of the treatment varies based on the patient's condition and if this variation is substantial. We found the difference in effectiveness to be 44.8% vs. 50% . Based on these figures, we determined that the patient's condition does not make a substantial difference in the effectiveness of the treatment.

Effectiveness of second line of treatment based on the response to the first line treatment: By comparing patients who responded to treatment in the first line to patients who didn't, we were able to determine that response in the first line substantially affects the effectiveness of treatment in the second line.

Distribution of the novel diagnostic tests: We fit a multivariate normal distribution to the results of the three novel diagnostic tests based on the first cycle of the first line of treatment. We opt for this distribution as the test results are correlated, i.e. getting a high value on the first result makes it more likely the patient will have high values on the other two as well.

We begin by separating the data based on whether the patient responded to the treatment or not (determined by the PET scan). We confirm that the normal distribution is indeed a good fit for the individual tests. Next, we find threshold values for the three tests using the quantile function: if one of the values is above the threshold, the patient is 80% likely to be a non-responder. We determine the threshold values for the three tests Dx1, Dx2 and Dx3 to be 0.94, 0.17 and 4.3 respectively. We validated this model against the actual data and had a 70% correct classification rate.

```
print(m_cov)
```

```
##                Test1_responders Test2_responders Test3_responders
## Test1_responders      0.010921139      0.002898993      0.06349671
## Test2_responders      0.002898993      0.004718492      0.03644682
## Test3_responders      0.063496714      0.036446819      0.84112569
```

The covariance matrix for the different test thresholds. If the variables were independent, they would have 0 covariance in relation each other. We see that while there is a correlation, it seems to be pretty weak.

Probability of experiencing major or minor complications during treatment based on response and condition: To find the probability of experiencing major or minor complications, we first stratify the data based on condition and then further based on response. We find that the condition and response of a patient does affect the probability of experiencing a complication. The probabilities vary from 0.0047 to 0.1875 for major complications. For minor complications, the probabilities vary from 0.02 to 0.175.

Distribution for the probability of experiencing minor complications during treatment: To fit a distribution to the probability of experiencing minor complications during treatment we chose to use a beta distribution. We can divide the data into two events, a patient experiencing minor complications or not. Therefore, a beta distribution will be a good match with the as the data is binomial. There are r events in n persons. Where n is the number of persons in the treatment cycle and r is number of occurrences of the event a patient experiences minor complications. As seen response and condition of a patient does affect the probability of experiencing minor complications. Therefore, we stratify the data based on response and condition and fit four different beta distributions to each vector.

Distributions for the mean utility of patients in Tx1 & Tx2 who did not experience any complications based on their response: To fit the distributions, we first stratify the data from the first cycle in both lines of treatment based on whether the patient responds or not. We do this individually for Tx1 and Tx2. By doing a visual check of the histograms for the individual vectors we see that all of them seem to have a normal bell curve. In addition to this, utility of a patient is never less than zero. Therefore, we conclude that lognormal will be a good fit for the vectors. We verify this by using the function `gofstat`. We then fit lognormal distributions to the

individual vectors.

Distribution for probability of death in first follow up: To fit a distribution to the probability of death in the first follow up we chose to use a beta distribution. The beta distribution returns a value between 0 and 1 which perfectly matches the probability parameter in the binomial distribution.

Supportive Functions

Based on the previous analysis, we code functions to define the sex of the patients; the age based on the sex; the costs of the diagnostic tests (only in PSA, they are hardcoded in BCA); functions to determine the events in Tx1, followup1 and Tx2; functions to determine the time patients spend in those cycles, incorporating time-to-event distributions; and cost, utility and qaly functions. Some helper functions were also defined in order to simplify the more complex functions.

Differences between the experimental and standard model

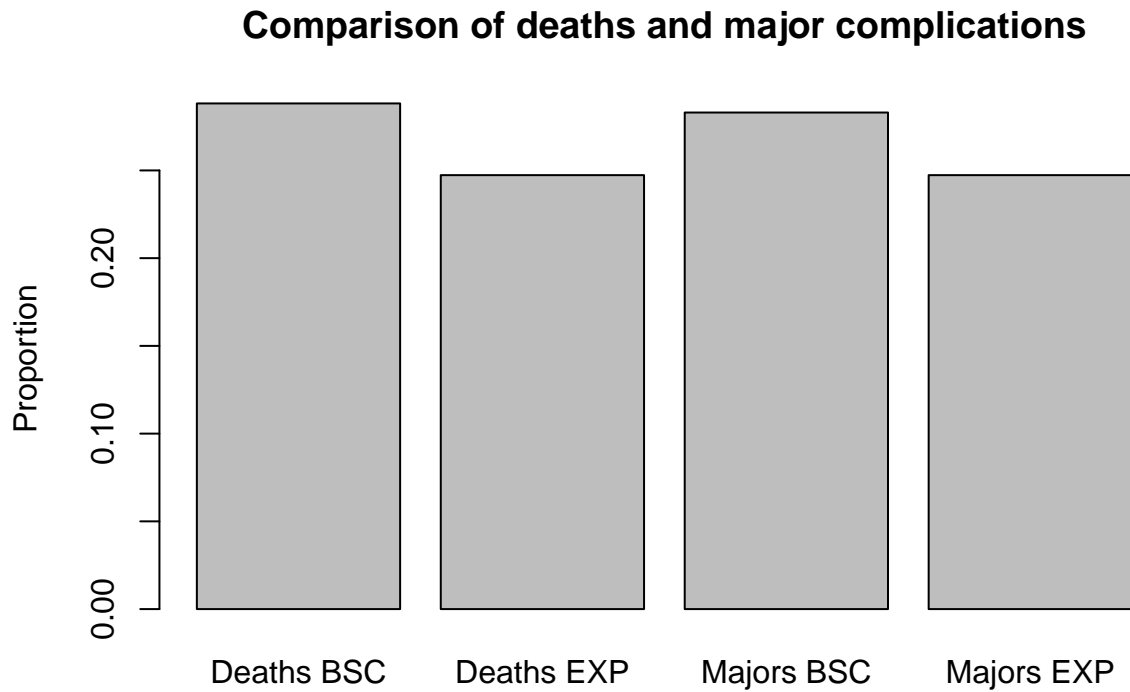
The two models are almost identical except two things: in the exp model, the cost of the diagnostic tests is added to the calculations; and the exp model uses alternate functions (Tx1.Event.alt and Tx2.Event.alt) to prevent overtreatment based on the results on the novel diagnostic tests.

Differences between PSA and BCA

The base-case analysis (BCA) does not consider uncertainty in the model parameters, though it may reflect patient-level variation in models (i.e. stochastic uncertainty or first-order uncertainty). The probabilistic sensitivity analysis (PSA) also accounts for parameter uncertainty (i.e., second-order uncertainty). Some practical examples: in the BCA code, the exp model uses fixed costs for the diagnostic tests, whereas in the PSA code, the costs are drawn from a normal distribution. The cycle costs and day costs (modelled in func.tx1cost) are drawn from normal distributions in the PSA but not the BCA code. For the cost of major complications, we selected a gamma distribution as it is right-skewed and better reflects the uncertainty in major complications.

Cost-effectiveness Estimation of the Novel Diagnostic Tests

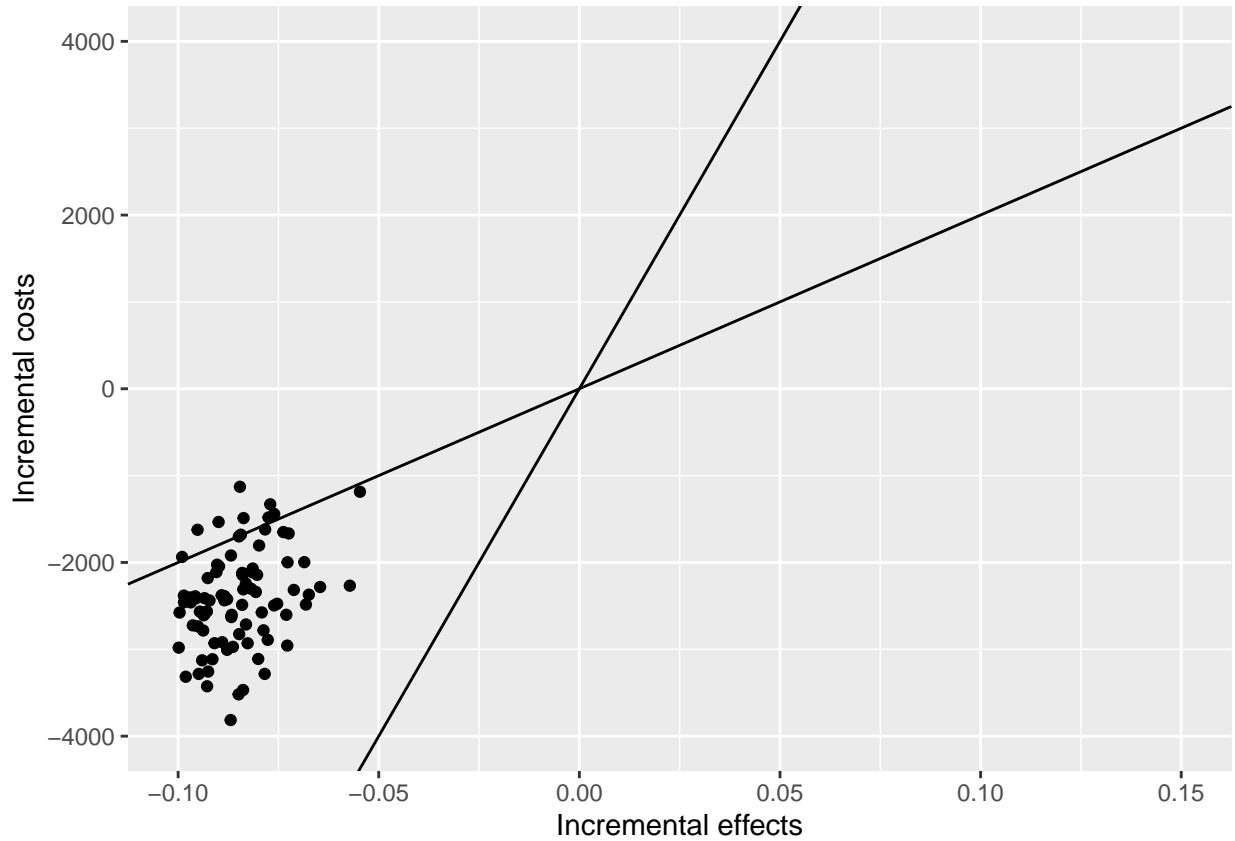
We found that our PSA data showed negative costs and negative effects for the exp strategy compared to the bsc model. This is likely to be a mistake in our model (perhaps we calculated the qalys or the costs incorrectly). It is possible to see why the exp strategy could save money: although the tests do cost a two hundred euros each (with a mean of €600 per cycle), the Tx1 cycle cost is €500 per cycle + day cost, and Tx2 costs a whopping €4500 per treatment cycle. Moreover, preventing overtreatment results in fewer complications, especially major ones, which are extremely expensive (average of €11,000). Let us compare the average proportion of major complications between the exp and bsc strategies:



As we can see, the BSC strategy has a higher proportion of both deaths and major complications.

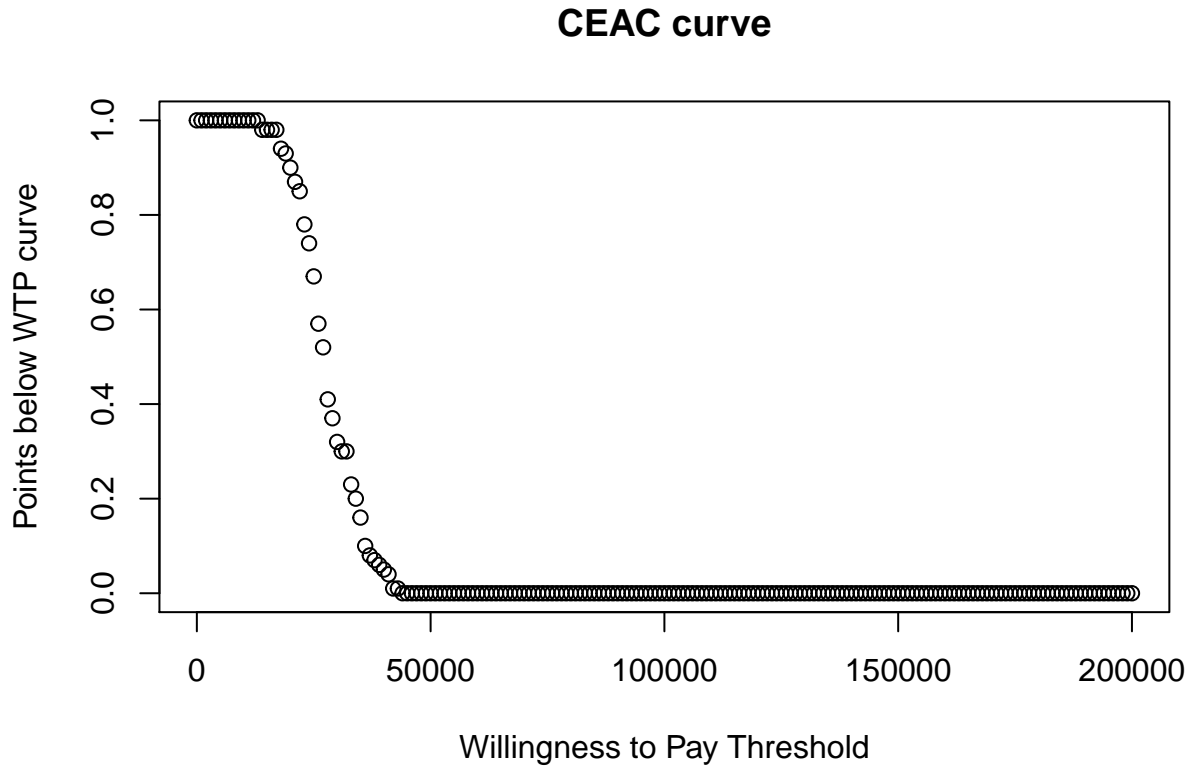
We calculated the ICER to be €27,647. This is almost identical to the ICER we calculated for the large sample dataset we were provided (€27,939).

```
## Warning: Removed 15 rows containing missing values (geom_point).
```



If we plot the Willingness to Pay Threshold of €20,000/QALY (used in the Netherlands) and a steeper €80,000/QALY line, we see that most of the points (90%) in our simulation fall below the €20,000 line. Paradoxically, they do not fall below the €80K line. This is because, when dealing with interventions that save money in exchange for a small loss in quality of life—like, for example, e-health video consultations—a higher willingness to pay means it is harder for such an intervention to be declared cost-effective. A higher WTP essentially means a higher financial value placed on a year of life.

Because the intervention has negative costs and effects, the CEAC curve is inverted:



Interpretation of the Results

Of course, a WTP of €80,000 is extremely high and not very realistic. A WTP of €20,000 is already one of the highest in the world (see international comparison) [3]. At this more realistic threshold, 90% of our results fall below the line.

One might question the ethics of introducing a healthcare intervention that harms patients (compared to the existing best standard of care) in exchange for cost savings. This argument is somewhat naïve because it does not take into account opportunity cost: by saving a lot of money on one intervention, other treatments can be paid for in exchange. These treatments may well end up saving more life years, for other patients, than the existing best standard standard of care does. Economic resources are limited; economics, as Robinson put it, is the study of scarcity [2]. As soon as we accept that healthcare resources are limited, we have to make intelligent choices between competing options in order to maximise utility on the budget line.

Despite this, it remains difficult, politically, to advocate for replacing an existing treatment with a more cost effective one—whereas it is very hard for governments to say “no” to very expensive new treatments. There are numerous such examples: Orkambi, the cystic fibrosis drug, was initially refused by the Dutch government, but then the minister relented [4].

	PPP-adjusted (2013 US \$)		Actual values (2013 US \$)		
Country/income classification	GDP per capita	Threshold range [*]	GDP per capita	Threshold range [*]	Threshold as % GDP per capita
Country					
Malawi	780	9–401	226	3–116	1–51
Indonesia	9,559	1,298–4,914	3,475	472–1,786	14–51
Chile	21,911	6,819–13,141	15,732	4,896–9,436	31–60
Kazakhstan	23,206	7,648–13,675	13,610	4,485–8,018	33–59
United Kingdom	36,197	18,609–18,609	41,787	20,223–20,223	48–48 [†]
Canada	43,247	21,051–26,564	51,958	25,292–31,915	49–61
United States	53,143	24,283–40,112	53,042	24,283–40,112	46–75
Norway	65,461	28,057–60,862	100,819	43,211–93,736	43–93
Income classification					
Low/middle income [‡]	1,045	16–537	NA	NA	1–51
Middle/high income [‡]	12,746	2,307–9,028	NA	NA	18–71

GDP, gross domestic product; NA, not available/applicable; PPP, [purchasing power parity](#).

Figure 2: International comparison of CETs (in 2013 dollars per QALY). Countries like Malawi and Cambodia have CETs as low as a few hundred euros per QALY.

Feedback provided

The feedback we gave really focused on code quality and reproducibility. One of the most common mistakes we see (and which you, the teachers, should have discouraged) is the use of `setwd()`. Using this function means the code will only run on the programmer's PC and not on else's. We recommended setting relative paths inside the root directory instead. Another mistake the group made was having the distribution fitting done inside the simulation files instead of separately. When we found a mistake, they had to make the changes in multiple places.

Also, we recommended that they use git to manage their source code, as they had to send us 2 different versions of their PSA simulation after we received it.

Lastly, we commented on programming best practices—for example, using helper functions, nested if statements to simplify conditional expressions, and so on.

Feedback received

We will go through the feedback received in a line-by-line fashion and describe how we changed our code (or chose not to change it) based on the feedback.

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.

We agreed with this and renamed some of the files to more descriptive names. We provided an explanation of what the code does in the README file.

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.

In the assignment instructions it said that patients continue the treatment cycle if they experience only minor complications. We understood that each treatment cycle was fixed at 30 days.

Last, you assume the time to a major complication in `Tx2` is symmetrical to `TX1`.

We do indeed assume this, because we were instructed to, and we had limited time to do more distribution fitting. We could easily improve this in the model by changing the `Tx2.Time` function.

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)

This is a valid point. But it is very unlikely that those distributions would return a negative cost because they have a large mean. We used the normal distribution because it is more straightforward to implement. Also, we are not sure if a gamma distribution is appropriate for most of the costs, as they are not meant to be skewed.

```
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))
```

We are doing this only in the BCA simulation because that is for patient-level uncertainty, not parameter uncertainty. The PSA simulation uses functions to reflect uncertainties.

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.

We do comment our code in various places to explain what events the numbers refer to. In the future, we would consider returning strings with (at least some of) the functions, in which case this would be self-explanatory.

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.

We chose not to use the results of our logistic regression in determining the probability because neither of the two predictors (age and sex) were significant at the $\alpha = 0.10$ level, which is the weakest threshold of statistical significance accepted by the scientific community. As you can see from our logistic regression results, neither predictor meets that criteria:

```
summary(logreg_death)
```

```
##
## Call:
## glm(formula = C2_transformed_events ~ male + age, family = binomial)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.6385   0.3006   0.3321   0.3704   0.4683
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.03738    1.21005   3.337 0.000848 ***
```

```
## male      -0.62509    0.43484   -1.438  0.150567
## age       -0.01633    0.01759   -0.928  0.353311
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 237.12  on 537  degrees of freedom
## Residual deviance: 234.97  on 535  degrees of freedom
## AIC: 240.97
##
## Number of Fisher Scoring iterations: 5
```

Even though we had more than 500 observations, only 31 people died in those two cycles. It is well-known that binomial events with a low probability are difficult to test statistically and require a very large number of observations. A good example is vaccination: clinical trials with thousands of participants still do not find rare complications that emerge only once the vaccine has been administered to the general population. This happened with the 1976 swine flu vaccine, which caused a 1 per 100,000 increase in Guillain Barré syndrome [1]. More recently there is controversy that the AstraZeneca vaccine can (rarely) cause blood clots.

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.

This is simply incorrect. The exp.model uses the function Tx1.Event.alt which skips the additional treatment cycles if the helper function returns false.

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.

They are. The exp model uses function.exp.costs which calls up the func.dxcost helper functions.

Overall reflection

Approach to the assignment: We took a really programmatic approach to this assignment, which served us well for the most part. Alex is an MBIT student who has done the software development course, and it was his idea to use git for the project; to test functions separately; and to try and write modular code. However, we found R extremely frustrating to use for serious programming. We didn't know how to do unit tests in R. Basic data structures (like recursive lists) were missing. Print statements inside functions or trajectories didn't seem to print when the model

was executed. Finally, the error messages were really unhelpful because they didn't even have a line number! Using break points didn't help either; in fact, they seemed to have no effect until after the simulation had already run into errors.

Collaboration in pairs: this was fine. We tried to divvy up the work as best we could. The branch functionality of git came in very handy at times.

Challenges encountered: Neither of us were very experienced with R and even less so with simmer. We had to learn about a lot of peculiarities, like the `clusterExport` function; how to save data from the PSA simulation (which took a while to figure out); how various functions worked; how to append to vectors (copy-on-write is weird); and even the indexing in R is very strange. It uses commas to index rows or columns. The indexing for recursive data structures is bizarre: we do not see why it has to be `list_of_lists[[1]][1]` instead of just `list_of_lists[1][1]` like you can do in Java with arrays.

What we learned from this course: How to do DES simulation in R; how to fit distributions in R; how to do statistical programming in general. We also learned a lot about the theory of health simulation, though this didn't come up so much in the practical assignment. The inclusion of economics was a nice touch. We would have liked it if the advanced topics were covered in a bit more detail; the lesson we had on meta-modelling and simulation optimisation felt rushed—there wasn't enough time in one or two lessons to fully absorb that material.

Moreover, we would also really have appreciated some more *programming* guidance. A lot of the people doing this course don't have a formal background in software development; they don't know about unit testing, functional programming paradigms, types and data structures in R, etc. It would have helped if you guys could have covered debugging in more detail—there were times when we really got stuck on those extremely unhelpful error messages.

Our view on applying DES for health economic evaluations: It's complicated. One thing that's clear to us is that you need good quality clinical trials with lots of data, in order to fit distributions and have accurate parameters. Furthermore, as your case study on meta-modelling showed us, the model is only as good as its underlying data. You can't just extrapolate to a group, e.g. aneurysm screening in under 40s, based purely on simulation results.

Speaking of meta-modelling, it is our personal opinion that if you need a meta-model in the first place, your original model is probably too complex to begin with. We think the real reason meta-models perform well is because simpler models are sufficient; trying to capture 10% more real-life complexity can lead to 10x the computational burden (not to mention code complexity). More sophisticated models could just be kept around for validation purposes.

Also, as a programmer, Alex would find it hard to trust a health economic model if the codebase is really messy. A simpler bug-free model is better than a more complex model filled with bugs.

Bibliography

[1]

Lawrence B. Schonberger, Eugene S. Hurwitz, Peter Katona, Robert C. Holman, and Dennis J. Bregman. Guillain-barré syndrome: Its epidemiology and associations with influenza vaccination. *Annals of Neurology*, 9(S1):31–38, 1981. [bib | DOI | arXiv | http]

[2]

Lionel Robbins. An essay on the nature and significance of economic science. Ludwig von Mises Institute, 2007. [bib]

[3]

Beth Woods, Paul Revill, Mark Sculpher, and Karl Claxton. Country-level cost-effectiveness thresholds: Initial estimates and the need for further research. *Value in Health*, 19(8):929–935, 2016. [bib | DOI | http]

Keywords: benefits package, cost-effectiveness, quality-adjusted life-years, threshold, universal health care, willingness to pay

[4]

Cystic fibrosis drug will be paid for after all, outgoing health minister says. 2017. [bib | http]