Title: From spreadsheets to script: experiences from converting a Scottish cardiovascular disease policy model into R

Running title: Converting health economic models from Excel to R

Authors:

Yiqiao Xin, PhD [1], Ewan Gray, PhD [2], Jose Antonio Robles-Zurita, PhD [1], Houra Haghpanahan, PhD [1], Robert Heggie, MSc [1], Ciaran Kohli-Lynch, PhD [1,3], Andrew Briggs, D.Phil [4], David A McAllister, MD [5], Kenny Lawson, PhD [6], Jim Lewsey, PhD [1]

Affiliations

1 Health Economics and Health Technology Assessment (HEHTA), Institute of Health and Wellbeing, University of Glasgow, Glasgow, UK. 2. Edinburgh Clinical Trials Unit (ECTU), University of Edinburgh, Edinburgh, UK. 3. Center for Health Services and Outcomes Research, Northwestern University, Chicago, IL, USA. 4. Department of Health Services Research and Policy, London School of Hygiene and Tropical Medicine, London, UK. 5. Public Health, Institute of Health and Wellbeing, University of Glasgow, Glasgow, UK. 6. Translational Health Research Institute, Western Sydney University, Sydney, Australia.

Corresponding author

Dr Yiqiao Xin yiqiao.xin@anlaysisgroup.com

Analysis Group

1 Angel Court, 14th Floor, London, EC2R 7HJ

Meeting(s) at which the work was presented (if any):

R for trial and model-based cost-effectiveness analysis workshop Part 2. 9 - 12th Oct 2020. https://r-hta.org/events/workshop/2020/

Word count (excluding the abstract, acknowledgments, references and figure legends): 3990

Key Points for Decision Makers

- Excel and R are both options for health economic modelling; while either may be suitable for programming a particular model, R has specific advantages such as greater transparency, reproducibility, adaptability and computational efficiency that may warrant a model conversion.
- Conversion of Excel-based health economic models to R is demonstrated to be a highly rewarding exercise for modellers with little experience modelling in R who are attempting to write their first script-based model.
- Coding without specialist health economics modelling R packages allows full flexibility of the model, while specialist packages may add convenience if the model structure is suitable.

1 Abstract (word count: 270)

Given the advantages in transparency, reproducibility, adaptability and computational efficiency in R, there is a growing interest in converting existing spreadsheet-based models into an R script for model re-use and upskilling training among health economic modellers. The objective of this exercise was to convert a Scottish cardiovascular disease (CVD) policy model from Excel to R and discuss the lessons learnt throughout this process. The CVD model is a competing risk state transition cohort model. Four health economists, with varied experience of R, attempted to replicate an identical model structure in R based on the model in Excel and reproduce the intermediate and final results. Replications varied in their use of specialist health economics packages in addition to standard data management packages. Two versions of the CVD model were created in R along with a Shiny app. Version 1 was developed without health economics specialist packages and produced identical results to the Excel version. Version 2 used the heemod package and did not achieve the same results, possibly due to the non-standard elements of the model and limited time to adapt the functions. The R model requires less than half the computational time as the Excel model. Conversion of the spreadsheet models to script models is feasible for health economists. A step-by-step guide for the conversion process is provided and modellers' experience is discussed. Coding without specialist packages allows full flexibility, while specialist packages may add convenience if the model structure is suitable. Whichever approach is taken, transparency and replicability remain the key criteria in model programming. Model conversions must maintain standards in these areas regardless of the choice of software.

1   Introduction

Decision-analytic models use mathematical relationships to analyse decisions by estimating future costs and benefits under conditions of uncertainty.[1-3] To construct these models, spreadsheet packages (such as Microsoft Excel) have historically been preferred by health economists in academia, heath technology assessment (HTA) agencies, and industry due to their broad accessibility, user-friendly interface, and familiarity. Compared to software such as TreeAge, spreadsheets 'force' users to develop the model from scratch so that there can be a direct translation of a conceptual model.

Awareness of the intrinsic limitations of building decision-analytic models in spreadsheet packages is rapidly rising among health economists. Spreadsheet models are difficult to modify and adapt, especially in complex models constructed with numerous spreadsheets and intricate cell dependencies. Running simulations with macros, such as Excel Visual Basic language (VBA), can be time-consuming and memory demanding.

In contrast, script-based modelling, especially using the open source and cross-platform R language has become increasingly popular in recent years owing to its advantages in many aspects of modelling.[4, 5] Script-based models may be more transparent, easily adapted and reproduced, can overcome some limiting features of spreadsheets, simplify to a single medium, broaden the scope of model structures, and potentially reduce computation time. While learning to build models in R usually involves a steep learning curve, in recent years 'standard' decision-analytic models (e.g. decision trees, Markov models) are increasingly developed in R, and dedicated R packages for health economics modelling have been developed, including the heemod [6], BCEA[7], hesim[8], and survHE [9] packages. There is also a growing number of publications,[10] tutorials [11] and workshops [12], advocating the merits of R for health economic modelling.

In the UK, R is officially recognised by the National Institute for Health and Care Excellence (NICE) as a software which companies can use to submit health economic models. Alternative software also approved by NICE includes Excel, WinBugs, and DATA/TreeAge.[13] With the powerful functionality of the R Shiny package, modellers can build an interactive web-based interface which allows stakeholders to easily conduct sensitivity analysis by altering assumptions.[14]

Health economic models evolve over time. Many previously-developed health economic models are used for multiple projects. These 'policy' models typically analyse different interventions for similar disease pathways. Rather than being superseded by de-novo models programmed from scratch, policy models have their structure adapted, coefficients reanalysed from new datasets, or undergo other amendments to analyse a novel decision problem. Given the benefits of modelling in R,[5] there is an increasing interest in converting existing spreadsheet-based models into R script to allow efficient model adaptation. Despite the increasing enthusiasm, many health economic modellers have little experience with it. Hence, R models lack of general acceptability and modellers lack familiarity in using R coding. Converting models from spreadsheets to R can therefore be used as a training exercise for spreadsheet modellers to become familiar with the programming features of R. The spreadsheet model can be used as a reference while the learner develops each line of R code. Through this process, users can gain experience with many aspects of the R language, such as data manipulation, looping, vectorised calculation, drawing from distributions, developing functions, visualization, and developing interactive app.

Here we report the experience of four health economists, with varied experience of R, replicating the Excel-based Scottish Cardiovascular Disease (CVD) Policy Model in R. Workflow best practice is less well established for modelling in R compared to Excel and so we conclude by providing a step by step guide and sharing our experience for building an R model from Excel with a view to guiding future practice and training in this area.

2   Methods

2.1   Scottish CVD policy model

The Scottish CVD policy model (Figure 1) is a state transition competing risk model published in 2015.[15, 16] It estimates life expectancy, quality-adjusted life-years (QALYs) and inpatient health care costs based on

cardiovascular risk factors, including age, blood pressure, total cholesterol, high-density lipoprotein, cigarette per day, family history of CVD, and socio-economic deprivation index. Individuals start in a CVD event-free state and can transition to one of the four first event states: two non-fatal event states, i.e. coronary heart diseases (CHD) and cerebrovascular diseases (CBVD), and two fatal events state, i.e. death from CVD cause or non-CVD cause. Individuals who experience a non-fatal first event are followed up until death with the impacts of secondary cardiovascular disease included.

The cause-specific hazards of a first event and post first event survival are both modelled using competing risk parametric survival analysis.[17] The probabilities of events and survival, costs and QALY were estimated in STATA based on Scottish routine data. It has informed policy decisions in Scotland such as the SIGN clinical guideline on CVD prevention in primary care.[18]

Since creation, the model has been adapted in several projects predicting long-term healthcare costs and patient outcomes. In the Football Fans in Training (FFIT) study, the CVD model was used to evaluate a weight management programme, with a BMI indicator replacing the original cholesterol covariates.[19] In Brazil, the CVD model has been adapted where a Brazilian life table was used and the risk equations for each first event were re-evaluated using a Brazilian cohort study.[20] The model has also been adapted to investigate the impact of novel cardiovascular risk biomarker as well as compare primary prevention risk stratification strategies.[21]
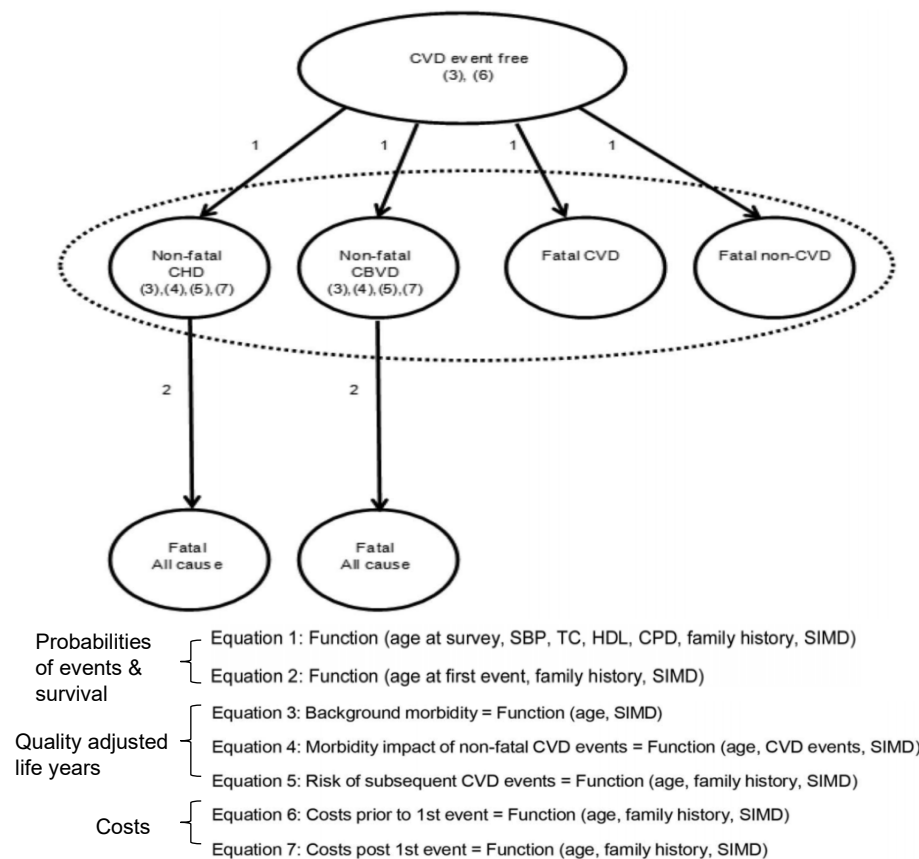

Probabilities of events & survival
- Equation 1: Function (age at survey, SBP, TC, HDL, CPD, family history, SIMD)
- Equation 2: Function (age at first event, family history, SIMD)

Quality adjusted life years
- Equation 3: Background morbidity = Function (age, SIMD)
- Equation 4: Morbidity impact of non-fatal CVD events = Function (age, CVD events, SIMD)
- Equation 5: Risk of subsequent CVD events = Function (age, family history, SIMD)

Costs
- Equation 6: Costs prior to 1st event = Function (age, family history, SIMD)
- Equation 7: Costs post 1st event = Function (age, family history, SIMD)

Figure 1 CVD model figure and equations

Abbreviations: CBVD: cerebrovascular disease; CHD: coronary heart disease; CPD: cigarette per day; CVD: cardiovascular disease; HDL: High-density lipoprotein; SBP: systolic blood pressure; SIMD: Scottish Index of Multiple Deprivation; TC: total cholesterol.

2.2   Process

Four health economists (YX, EG, JR, HH) with varied experience of modelling in R (Appendix 1) formed a team of coders aiming to replicate an identical model structure in R based on the CVD model in Excel and

reproduce the same intermediate and final results. The process of conversion was split into three stages as outlined in Figure 2. In the first stage, four health economists independently coded the first part (i.e., Equation 1&2 in Figure 1) of the CVD policy model [16] to generate life expectancy for individuals defined by a series of risk factors. The parallel 'independent' coding approach was adopted rather than collaboratively coding based on two considerations. First, the coders had different previous experience with R which may result in different coding approaches. Second, model replication between different people can reduce the chance of random errors and improve coding transparency.

After the first stage of coding was complete, the coders met and compared R code in terms of the approach taken, techniques learnt, life expectancy results, and other issues including errors identified in the Excel model. During this meeting, it was found that two members of the team coded the model using 'generic' Base R and data management packages tidyr and dplyr only, i.e. not using any health economic modelling packages, and the other two coders explored using specialised packages including the health economic R package named heemod.[6] Based on the two styles, two teams were formed, and each team worked together at Stage 2 of the exercise to generate the code for estimating QALYs, costs and conducting probabilistic analysis. In the end, two final versions of code were produced: one coded without using any health economics specialist packages; and the other one using the heemod package.

Upon completion, the two R versions of the model were compared against each other and with the Excel model in terms of outcomes across different patients' profiles and time horizons. An iterative process of error identification and correction in Excel was undertaken, followed by debugging in R. The computational efficiency was compared by running 1,000 simulations for probabilistic analysis. Throughout the entire process, the coders were supported by the CVD model developers (JL, AB and KL) and colleagues who adapted the model for different projects (CKL and RH). This support included aiding the understanding of the model and correcting the computational errors in Excel identified in the conversion process, mostly cell referencing errors or formula typos. Finally, for demonstration purposes, a Shiny app of the verified R version of the CVD policy model was created which can be used to visualise the cost-effectiveness of statins as described previously [21] to different stakeholders.
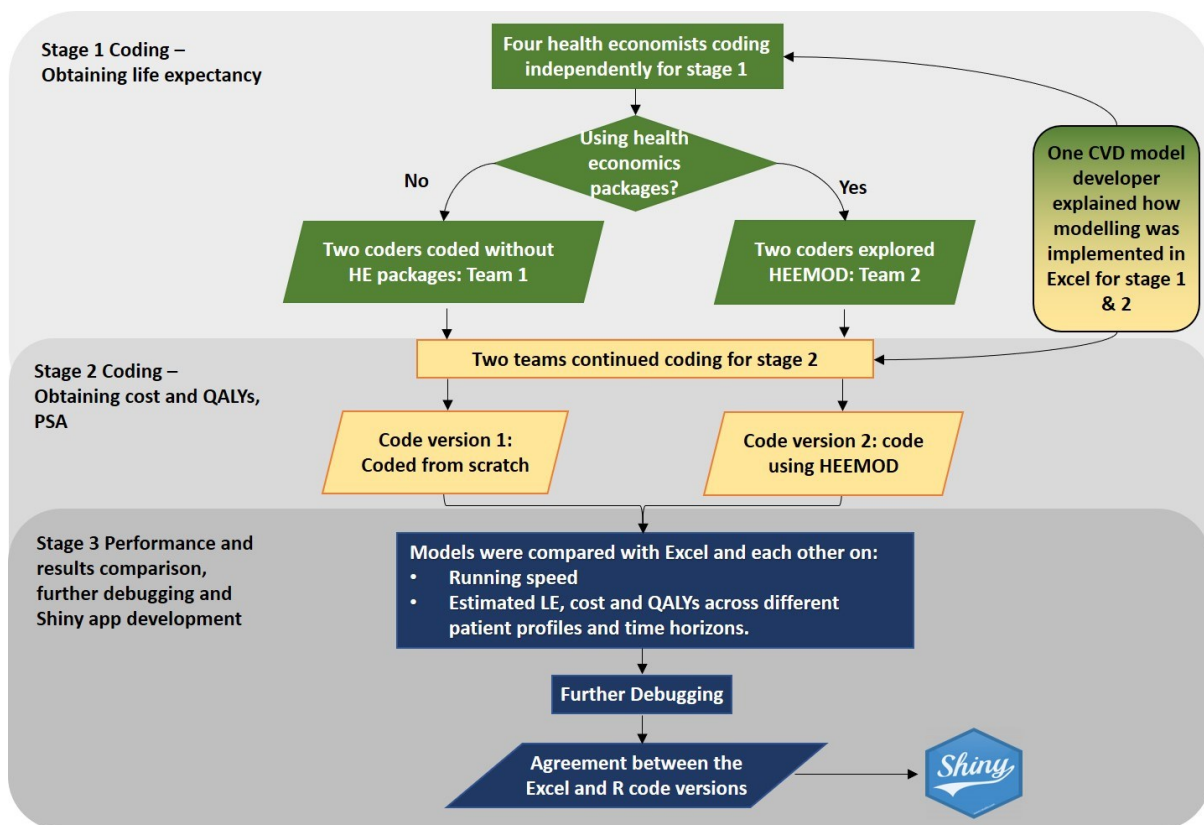
Figure 2 Process of conversion of the CVD model from Excel to R code

Abbreviations: CVD: cardiovascular disease; HE: health economics; heemod: Markov Models for Health Economic Evaluations R package; LE: life expectancy; PSA: probabilistic sensitivity analysis; QALYs: quality-adjusted life-years.

3   Results

The R version coded without specialist health economics packages produced results identical to the corrected Excel model. These include: the cumulative incidence of first event, probability of each event at each cycle, pre-event and post-event costs, QALY decrement based on the probability of each secondary event, and the final outcomes including life expectancy, costs, and QALYs. The R code of this version is publicly available from the Github repository.[22]

The base case results from the heemod version varied from the corrected Excel model by approximately 1% for discounted QALYs and 20% for discounted costs at maximum when testing across pre-set various patient profiles and time horizons.

3.1   Computational efficiency

Computation tests in the same PC environment showed that the verified R model (i.e. the version without specialist health economic package) run time produced at least twice the computational efficiency as the Excel model; the R model took 4 minutes 49 seconds to run 1,000 Monte Carlo simulations, versus 12 minutes 4 seconds in Excel. The R code was written in a way to increase the readability and transparency to the people who are not familiar with the model so computation efficiency was not a particular focus. It is possible that the code may be made even more efficient with use of alternative functions, special packages (e.g. hesim[8] given written in C++), parallelisation (multi cores of CPU), or bridged to external C++ environment with packages such as speedyR and Rcpp.[23] We also observed the computational superiority of the R model without specialist health economics packages compared with the R model with heemod. The former took 2.69 seconds to run the deterministic analysis; that is 22.8 times faster than the latter.

3.2   Shiny app

A Shiny app (https://yxin.shinyapps.io/CVDmodel_treatment/) of the CVD model evaluating the cost-effectiveness of statins for reducing the risk of cardiovascular diseases as described in Kohli-Lynch (2019) was developed based on the version coded from scratch for illustrative purposes (Figure 3).

On the side bar, users can input patient characterises, including age, sex, diabetes, etc. Below the patient characteristics is treatment information. In this illustrative example, a statin treatment effect on LDL and HDL cholesterol, as well as a cost and disutility associated with statin use were incorporated. At the bottom, users can choose discount rate and time horizon. The results of base case analysis with and without statins are shown on the main panel, including life years, costs, and QALYs, as well as the incremental outcomes. The output is updated in real time when the parameter inputs are changed. Users can run a probabilistic analysis by specifying a number of iterations and results are presented as a cost-effectiveness plane and cost-effectiveness acceptability curve (CEAC). The interactive plot was developed using plotly package [24] and coordinates are shown when the user hovers over a point.

Figure 3 Shiny app interface of the Scottish cardiovascular disease policy model

4 Discussion

Health economic models are used by funding bodies to make decisions regarding alternative healthcare interventions. This purpose requires transparency to allow critical review within a limited timeframe and computational efficiency to enable complex estimation of model uncertainty. In recent years, there has been a growing interest in building health economic models with R due to benefits such as increased transparency, reproducibility, adaptability and computational efficiency. Inspired by this, we initiated a team project to convert the Excel-based Scottish CVD policy model 'from spreadsheet to script'. The R script of the CVD model has been created along with a Shiny app interface.[22] In this section, a step-by-step tutorial for model conversion from Excel into R is outlined and modellers' experience on the conversion process is discussed.

4.1 Step-by-step guide for model conversion

A step-by-step best practice guide for model conversion from Excel to R is presented in Figure 4. The first step is understanding the original model by reading model description documents, examining the formulae of the spreadsheet file, and engaging with the original model developers. This step is key for the whole conversion process and iterative as necessary when developing each unit of the code. It can be time-consuming if the model was developed by multiple modellers and the structure is complex. However, understanding the model correctly and precisely is vital to any replication exercise.

After gaining sufficient understanding of the model, the next step is to select a general approach to the coding in script. One reason that many health economists favour Excel over other software options is that Excel allows the full control of the model as users build the model with limited use of complex functions or pre-written programs. This is the same with coding 'from scratch' in R where coders write each line of the code to build the calculations into the model script. An alternative approach to coding from scratch is to use one or more specialist packages. A list of health economics packages are compiled on Github.[25] Modellers may assess if there is any package available that can accommodate the specific model structure and provide additional efficiency or convenience; the default should be coding from scratch. Modellers may also leverage some functions from packages while coding from scratch providing they have sufficient understanding.

After determining the general approach, the third step is planning of the implementation by splitting the model into smaller components that can be completed as separate tasks, before coding it in R. The necessity of this step will depend on the complexity of the model and the R skills of the coder. We split the coding tasks of the model into three parts: calculation of transition probabilities and survival outcomes; calculation of costs and QALYs; and coding deterministic and probabilistic sensitivity analysis. Before coding each task, the coders are recommended to re-familiarise themselves by going through the specific sections of Excel model. The R code should be modular in a manner that may or may not match the discrete tasks completed to produce the code. In this case, we modularized the code the same way as when we implemented the model in Excel but there may be other approaches to modularization. An alternative way suggested by one peer reviewer is sorting R model scripts into 1) input (taking in, sorting, intermediate calculations); 2) functions to convert inputs to the results, including deterministic and sensitivity analysis, and quality control functions; 3) presentation of results to audience via graphs, tables, R Shiny, and Rmarkdown reports. This is similar to the approach that has been recommended by the DARTH group in 2019.[26]

The fourth step, coding, is an iterative process of 'converting' the Excel formulas into R code and writing supporting code to set up the R model structure. In our experience with the CVD model, besides the immediate calculations (e.g. linear predictors from survival analysis, drug costs), the essential part of the conversion process is to code the 'traces', i.e. calculation of survival probabilities, costs, QALYs over the cycles. This process involves some conceptualisation: a), identify a section of model structure in an excel sheet (e.g. cumulative incidence over the time horizon); b) identify how the formulas applied iterate over row and/or columns (e.g. the cumulative incidence at the current cycle are dependent on the cumulative incidence at last cycle); c) consider how this could be represented by vectorised or matrix operations or loops in R (e.g. we chose loops for the cumulative incidence calculation given this is a dependent process over each cycle); d) write equivalent formulas within a vectorised/matrix or loop in R (please see Github repository [22]: section 1.1 # *Cumulative incidence*). Once the code for every step of the calculation is developed, it is recommended to check the intermediate output with Excel. Given that clicking every cell of a column in Excel to check for errors is time consuming and unrealistic, the cross-software checking should be a mixture of different forms, e.g. checking the values at beginning cycles, some values in the middle cycles, and the sum of a column (or vector). This scrutiny ensures the R code is an exact replication of the Excel model. Errors in health economic models are unsurprisingly common; a recent study found that 95% (39 out of 41) of health economic models submitted to NICE single technology appraisals in 2017 contain technical errors, 25% of which are computational.[27] If errors are detected in the original model, like in our case, it is highly recommended that they are corrected immediately so the Excel model can continue to be used as a reference to validate the R code. This is especially helpful for R novice modellers. Once the base case code has been developed, it is helpful to organise the code into sections which can be written into functions. This can facilitate the looping in probabilistic analysis.

During and after the code is developed, it is good practice to tidy up and annotate the code while the coder's memory is fresh. This would significantly enhance team co-operation on model development, model sharing, as well as the scrutinisation process by HTA agencies. It may also be useful to conduct performance or quality control tests for verification. Besides being technically correct, visualisation and presentation of results cannot be neglected given the diverse model audience which requires the model to be accessible and transparent to coders and non-coders alike. A Shiny interface is recommended to be developed to enhance communications with stakeholders[14] via the display of interactive intermediate and final outcomes based on user inputs.
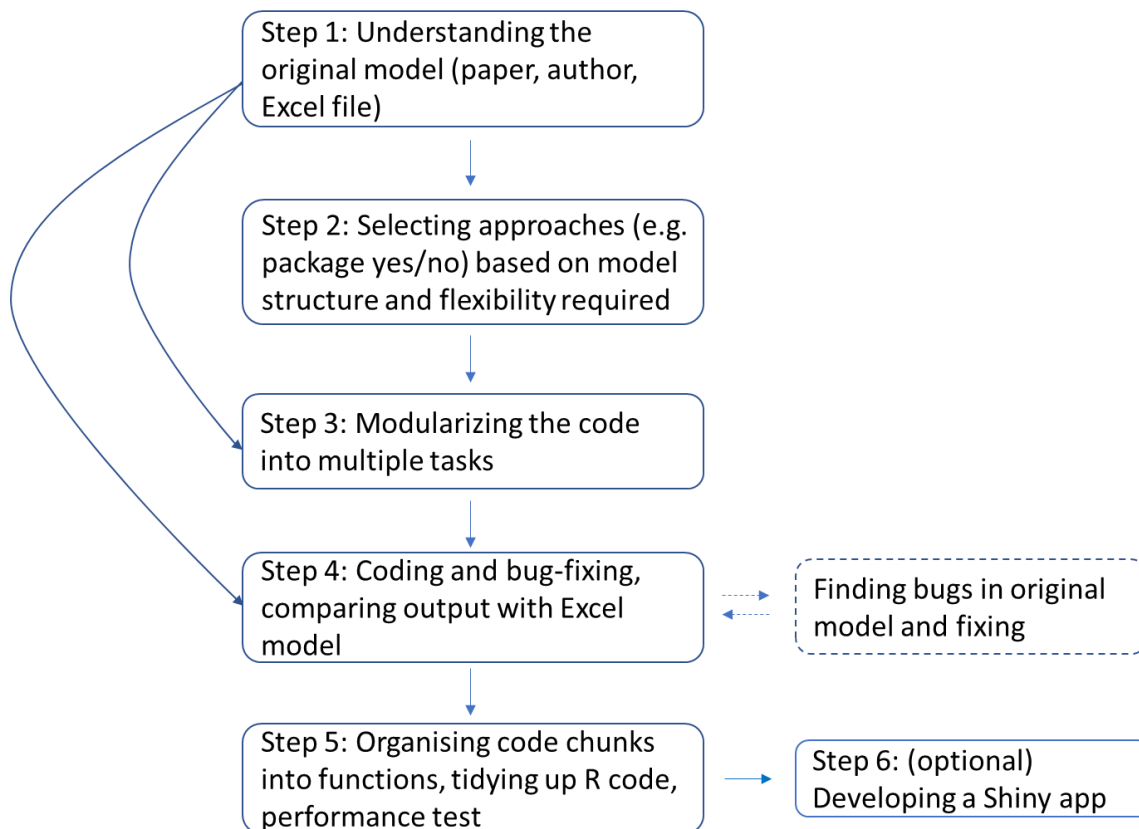
Figure 4 Standard flow of model conversion from Excel to R

4.2 Modellers' experience

Learning new software is always challenging especially when existing platform is so widely implemented. We did not intend to provide the best example of R code for economic models; rather, the aim of this article is to provide some insights on the learning experience to economic modellers who are attempting to convert models to R but had not have much specific R training and experience.

A key question that may arise from the audience of this paper is time; as the modellers needs to trade off the time spent on converting models to R with the benefits gained. This is especially true in a situation where an Excel model is still needed (e.g. for HTA agencies that only accept models built in Excel). The time required for each coder to complete the tasks varied; it took about 5-10 days for the estimation of life years step (part I as referred above), and another 5-10 days for the estimation of costs, QALY and coding the PSA. For code validation and bug fixing upon comparison between the coder's versions it took approximately 3-5 researcher days in total. Bug fixing was also undertaken throughout the main coding process which has been incorporated in the time estimate within each task. We did not keep records on the time taken for each specific tasks so these estimates are based on our recollection. It also should be kept in mind that the CVD policy model is more complex than most cost-effectiveness models for HTA. The simple demonstrative Shiny app was built over a half-day by one of the modellers who had previous experience (2 years) on Shiny apps but it will take longer time to learn for a novice. The separation between the calculation and the interface via R Shiny means modellers just need to focus on one thing at a time, making the model development process more streamlined. The previous experience with R and economic modelling in R of the modellers is outlined in Appendix 1. One interesting phenomenon in this process is that the modellers that had more experience played a more active role in bug fixing and code conceptualisation (including functions chosen) at the start of the process, however, over the course of the exercise, with hands-on learning, the first time modellers became more confident and took lead for coding and bug fixing.

The model building time depends on the complexity of the model. For the CVD policy model, all the elements of the model are relatively easy to implement for even the first time R modeller, in particular, input of parameters, coding traces, calculation of costs and QALYs, as well as the PSA. This is perhaps partly because the existing Excel model provided us a 'gold standard' so that we knew how the result object should look like. For some calculations we felt it is not as straightforward as Excel, for example, the vlookup process, is not as convenient as in Excel, although we suspect more elegant R solutions may exist for those with more experience. In fact, we acknowledge that our code could be modified to be more elegant or efficient but given our limited time, experience and scope on this project we did not spend additional time on this. In the peer review process we received some very helpful suggestions on coding so we have compiled them together in Appendix 2.

Another point of interest is regarding the differences in results found between hand coding and heemod. Replication of the CVD model with the heemod package leverages the package's existing functions which, in this context, proved to be more challenging. We have limited time and experience to fully explore exactly why the results were differed. However, we suspect that heemod cannot replicate the results of the CVD model for the following two reasons. Firstly, heemod implements different approaches to half-cycle correction. In heemod the half cycle-correction has to be consistent for all the model time whereas in the CVD model there was a different approach for the time to first event and time after first event. In particular, the CVD model assumes that the first event happens at the end of each model cycle and applies a half cycle correction (trapezoid rule) for the computation of survival time after the first event. Secondly, the pre-event costs in the CVD model are contingent on the future progression of patients which was not captured in heemod. For future reference, we believe heemod would be suitable for standard Markov/semi-Markov/non-homogenous-Markov model, otherwise we can achieve only an approximation of results. Despite this, heemod had its own advantages: easy to implement and obtain results, programming errors checking (i.e. to make sure parameters are consistent). For non-standard models, based on our own experience, using a package would be less flexible and maybe more time-consuming as modellers have spent effort to 'wire in' (e.g. adapt the functions in the packages to suit the non-standard models) and may eventually still only achieve an approximation of results. In addition, for learning purposes, base R functions plus dplyr/tidyr may be better for developing new skills and lead to a greater understanding than using existing functions. There are numerous packages that have already been developed that can be used for health economics and future endeavours should focus on providing guidance on how these packages work, the benefits of each of them, and how users can choose between these packages (e.g. depending on their model type and their other needs). These insights may be very useful to novice health economics R modellers deciding whether to use these packages, and which package to use. Finally, we collected some quotes from the R modellers reflecting their experience of this project (Box 1) and we hope these are interesting to our audience.

Quotes from the modellers

"Over last 3 years I have been working on Type 2 diabetes policy model which is essentially similar to the CVD policy model. The modelling is a combination of R programming and Excel analysis. In diabetes project, data extraction from database, data manipulation, estimation of survival parameters, cost and utility modelling and programming have done in R environment. The economic modelling has been conducted in Excel. As a modeller who I was involved in the similar project as CVD policy model, I would prefer to keep conducting analyses in R rather than Excel, because first avoiding any mistakes that may occur while transferring the estimation results from R to Excel. Secondly, it is more convenient to have control on executing codes with one software, and finally the time that you may save using same software/environment. For modellers who are willing to start with R (with little experience), they may spend ample time on learning the R skills but they will get benefit of more accuracy in the analyses and high speed processes as well as reproducibility of the results."

- HH

"In general the main challenge with heemod was the impossibility to replicate some elements of the model. Since the CVD model was not a 'standard' Markov/semi-Markov/non-homogenous-Markov model, there were some aspects that were not properly replicated. In heemod the easiest part to implement is the survival element of the model since it is implemented by making use of a Markov matrix that can change with the model time or the state time. Including utilities and costs, that are a function of characteristics and spline functions, requires some more elaborated coding. Lesson learnt: In the translation of a model from Excel to

---

R, it is very important to understand the original Excel model as it is to know how to program the different elements of the model in a script-based software. These two parts of implementation are not separated and are parts of a feedback loop, for example, the understanding of the Excel model improves as you progress with the R code."

- JR

"Sometimes in the conversion process it feels in R it takes extra amount of time to figure out a simple thing which I have done millions of times in Excel. My advice is that don't be discouraged by these seemingly time-consuming explorations. Keep going, googling or seeking advice. Once the code is there, it is there. You can copy and paste them to any project you next work on, or write them into functions and call them in other projects. Most importantly, you gain experience during these 'figuring' times, and with more experience, you will start understanding how functions are written in packages and 'borrow' the code from other people (thanks to the 'Sharing is caring' spirit among R users). Everything will soon become streamlined."

- YX

"I think I had the most experience in R going into the project, having coded both a cohort Markov and discrete event simulation previously. When coding the first part of the model, I was able to produce working code much more quickly, but it was obvious in the later parts that the other had caught up on the learning curve already. For me, the project really emphasised the usefulness of having multiple coders for error finding and bug-fixing. Because of that, I am more confident in the accuracy of this model than any other I have worked on. I think that R conversions like this are useful because R has so much potential for extension of the model to include more complex elements such as patient heterogeneity, as well as convenient ways to present results."

- EG

5 Conclusions

Excel and R are both options for health economic modelling. While either may be suitable for programming a particular model, R has specific advantages that may warrant model conversion. While endorsing model conversion from Excel to R, we do not wish to imply that models in Excel are unsuitable or necessarily inferior. Like Excel, R is not immune from errors and poorly-coded R models can also hinder transparency and be computationally inefficient. Nevertheless, enthusiasm and acceptance of modelling in R is increasing, having witnessed the trend of improving skill level of R and widening usage of R among health economists in academia, consultancy and HTA agencies.[28]

Conversion of an Excel model to R can greatly enhance running speed and thereby enable future model adaptation with complex elements. It may also enable model communication via web-based platforms. Coding without specialist packages allows for full flexibility, while specialist packages are becoming powerful and are efficient options for standard models. The process has been shown to be a great learning opportunity for spreadsheet-based health economic modellers to write their first script-based model. It is hoped that others will be inspired by this project to make a smooth transition from modelling in Excel to R and embark upon a new era of script-based modelling.

6   References

1.   Briggs A, Claxton K, Sculpher M. Decision modelling for health economic evaluation: Oxford University Press; 2006.
2.   Weinstein MC, O'Brien B, Hornberger J, et al. Principles of Good Practice for Decision Analytic Modeling in Health-Care Evaluation: Report of the ISPOR Task Force on Good Research Practices—Modeling Studies. Value Health. 2003;6(1):9-17.
3.   Siebert U. When should decision-analytic modeling be used in the economic evaluation of health care? Eur J Health Econ. 2003;4(3):143-50.
4.   Jalal H, Pechlivanoglou P, Krijkamp E, Alarid-Escudero F, Enns E, Hunink MGM. An Overview of R in Health Decision Sciences. Med Decis Making. 2017;37(7):735-46.
5.   Incerti D, Thom H, Baio G, Jansen JP. R You Still Using Excel? The Advantages of Modern Software Tools for Health Technology Assessment. Value Health. 2019;22(5):575-9.
6.   Filipović-Pierucci A, Zarca K, I D-Z. Markov Models for Health Economic Evaluations: The R Package heemod 2017. Access date: 7June2021. Available from: https://arxiv.org/abs/1702.03252.
7.   Baio G, Berardi A, A H. Bayesian Cost-Effectiveness Analysis with the R package BCEA: Springer International Publishing; 2017.
8.   Incerti D, JP J. hesim: Health Economic Simulation Modeling and Decision Analysis 2021. Access date: 7June2021. Available from: https://arxiv.org/abs/2102.09437.
9.   Baio G. survHE: Survival Analysis for Health Economic Evaluation and Cost-Effectiveness Modeling. J Stat Softw. 2020;1(14).
10.   R for Health Technology Assessment. Publications. 2021. Access date: 17 Jan 2021. Available from: https://r-hta.org/publication/.
11.   Williams C, Lewsey JD, Briggs AH, Mackay DF. Cost-effectiveness Analysis in R Using a Multi-state Modeling Survival Analysis Framework: A Tutorial. Med Decis Making. 2017;37(4):340-52.
12.   R for HTA Consortium. R for HTA annual workshop 2020. Access date: 17 Jan 2021. Available from: https://r-hta.org/project/workshop/.
13.   National Institute for Health and Care Excellence. Guide to the processes of technology appraisal 2018. Access date: 17 Jan 2021. Available from: https://www.nice.org.uk/Media/Default/About/what-we-do/NICE-guidance/NICE-technology-appraisals/technology-appraisal-processes-guide-apr-2018.pdf.
14.   Smith R, Schneider P. Making health economic models Shiny: A tutorial. Wellcome open research. 2020;5:69.
15.   Lawson KD, Lewsey JD, Ford I, et al. A cardiovascular disease policy model: part 2—preparing for economic evaluation and to assess health inequalities. Open Heart. 2016;3(1):e000140.
16.   Lewsey JD, Lawson KD, Ford I, et al. A cardiovascular disease policy model that predicts life expectancy taking into account socioeconomic deprivation. Heart. 2015;101(3):201.
17.   Putter H, Fiocco M, Geskus RB. Tutorial in biostatistics: competing risks and multi-state models. Statistics in Medicine. 2007;26(11):2389-430.
18.   Healthcare Improvement Scotland. SIGN 149 • Risk estimation and the prevention of cardiovascular disease. A national clinical guideline 2017. Access date: 16 Feb 2021. Available from: https://www.sign.ac.uk/assets/sign149.pdf.
19.   Gray CM, Wyke S, Zhang R, et al. Public Health Research. Long-term weight loss following a randomised controlled trial of a weight management programme for men delivered through professional football clubs: the Football Fans in Training follow-up study

Chapter 5 Economic evaluation: methods and results. Southampton (UK): NIHR Journals Library; 2018.

20.   Riveros BS, Torelli Reis WC, Lucchetta RC, et al. Brazilian Analytical Decision Model for Cardiovascular Disease: An Adaptation of the Scottish Cardiovascular Disease Policy Model. Value Health Regional Issues. 2018;17:210-6.

21.   Kohli-Lynch CN. Beyond ten-year risk: novel approaches to the primary prevention of cardiovascular disease 2019. Access date: 17 Jan 2021. Available from: http://theses.gla.ac.uk/74296/.

22.   Xin Y, Gray E, Robles-Zurita JA, et al. The Scottish cardiovascular disease (CVD) policy model 2020. Access date: 17 Jan 2021. Available from: https://github.com/yiqiaoxin/CVDmodel.

23.   Abbott S. Speed up Discrete Markov Model Cost Effectiveness Simulations 2020. Access date: 17 Jan 2021. Available from: https://www.samabbott.co.uk/SpeedyMarkov/.

24.   Getting Started with Plotly in R 2021. Access date: 14 Feb 2021. Available from: https://plotly.com/r/getting-started/.

25.   Health Economics R Packages 2020. Access date: 17 Jan 2021. Available from: https://github.com/n8thangreen/health_economics_R_packages.

26.   Alarid-Escudero F, Krijkamp EM, Pechlivanoglou P, et al. A Need for Change! A Coding Framework for Improving Transparency in Decision Modeling. PharmacoEconomics. 2019;37(11):1329-39.

27.   Radeva D, Hopkin G, Mossialos E, Borrill J, Osipenko L, Naci H. Assessment of technical errors and validation processes in economic models submitted by the company for NICE technology appraisals. International journal of technology assessment in health care. 2020:1-6.

28.   R for HTA Consortium. R for Health Technology Assessment 2020. Access date: 17 Jan 2021. Available from: https://r-hta.org/.