Tips for Excel(lent) modelling

or R there better alternatives?

J. Maervoet[1], J.E. Poirrier[1], R. Bergemann[2]

[1] Parexel International, Wavre, Belgium;
[2] Parexel International, Basel, Switzerland

MSR32

View all of Parexel's posters at ISPOR Europe

Background

- Health economic (HE) models can be developed in a variety of applications, including:
  - specialized commercial packages (TreeAge, Simul8, Arena, @RISK…)
  - spreadsheet software (nearly always Microsoft Excel)
  - programming languages (R, Python, Matlab, WinBUGS, C++)
- Whilst the use of R in health decision sciences has been increasing in recent years[1], Excel is still being used very commonly for development of HE models to be submitted to health technology assessment (HTA) bodies[2]
- In the HE literature and textbooks, there is a paucity of hands-on, practical guidance for the construction of efficient models in Excel

Objectives

- Our objectives for this study were to:
  - Highlight strengths and limitations of Excel (vs alternatives) for development of HE models
  - Provide hands-on practical tips to improve Excel's efficiency in handling HE analyses
  - Consider when it may be better to use alternative software

Methods

- A targeted search was conducted to identify literature on the use / comparison of software for development of HE models
- The Parexel HEOR modelling team convened a workshop to discuss findings of the literature review and share experiences in London, UK on April 24th, 2023

Results

Strengths and limitations

- Excel is widely used and relatively easy to learn. Most HE modelers and decision-makers, often not statisticians or programmers by training, are less familiar with alternative packages[3]
- Excel has a very large number of built-in statistical and econometric functions. Its statistical functions and distributions have improved considerably over time[4], but are still less sophisticated than those of proper statistical software and programming languages[5]
- Advanced users can program VBA custom functions and macros to further extend Excel's capacities (e.g., automate Monte Carlo simulation, conduct micro-simulation, calculate spline and fractional polynomial curves… etc.)
- The quality of Excel's random number generators remains a concern and the inability for users to set a 'seed' number (used to initialize these generators) is a major shortcoming causing issues for reproducibility and validation[4,5]
- Excel can be slow when complex HE models or computationally demanding analyses must be run. Comparative computation tests have demonstrated that programming languages are more efficient and much faster[2,5,6,7]
- Ambiguity over the acceptance of HE models in programming languages by HTA bodies around the world is still a significant barrier to their adoption[2]

Practical tips

- It has been shown that the optimization of the HE model framework and coding can somewhat improve Excel's runtime[8]
- Based on the literature and the team's experience, some hands-on practical tips for the construction of efficient HE models in Excel are provided in table 2

When to consider alternatives

- There are certain situations in which programming languages are much better suited for HE analysis than Excel
- For example, R allows the user to:
  - Build HE models including all required stages of analysis from using patient-level data to reporting outcomes. While some inputs for Excel models need to be derived outside this software, R's end-to-end functionality allows updates to underlying statistical analyses (e.g., network meta-analysis, population adjustments, subgroup analysis, curve extrapolations, adjustments for bias…) and resulting HE outcomes in real time[6]
  - Conduct value of information (VOI) analysis, relying on computationally intensive methods[1]. Certain types of VOI analysis (e.g., EVSI and EVPPI) require nested Monte Carlo simulation, which is often infeasible in Excel[2,3]
  - Estimate unknown parameters during model calibration and accurately solve equations in dynamic transmission models, both especially common in infectious disease / vaccine modeling[9]

Table 1. Strengths and limitations of Excel for development of HE models

Strengths	Limitations
It is widely used in educational institutions and many industries, and relatively easy to learn	Manual data entry and programming, which is error-prone and requires thorough cell-by-cell checks
Easy to share with and familiar to reviewers, HTA agencies, and other decision-makers.	Processing times can become long when e.g., many iterations or large sample sizes are involved
Has a graphical user interface, making it easy to navigate and allowing a rich presentation of results	Statistical analyses on source data (e.g., survival analysis) often needs to be conducted / rerun outside of Excel (no 'end-to-end' functionality)
Often perceived as transparent software package (although this is being questioned in the literature)	It is not very efficient to adapt / update clinical parameters (for new subgroups, data cuts...)
Very well suited and efficient for construction of models that are not very complex	Seed for random number generation cannot be set, which causes reproducibility and validation issues
Has many valuable features including support for a variety of statistical and econometric functions	Statistical functions are less sophisticated and not as accurate as in statistical / programming software
Highly extensible through the use of macros programmed in VBA and add-in packages	Not well suited for certain modelling techniques (dynamic transmission, discrete event simulation...)
Offers a solid framework and functionalities allowing Monte Carlo simulations in PSA	

Table 2. Tips for development of HE models in Excel

Structural, formatting, and computational tips
It is better to consider the structural design of future workbooks beforehand, than starting without a plan
Avoid links to external files as they can be slow, easily broken, and not always easy to find and fix
Minimize the 'used range' (check with: Ctrl+End). Delete unused sheets, ranges and (formatting of) unused cells. Delete all rows & columns below and to right of your real last used cell
Keep source data in one place and limit the number of worksheets to speed up calculations
Don't apply conditional formatting to very large data sets and/or to entire columns or rows
Compress high-resolution pictures and shapes
Run computationally-intensive tasks on a cloud server instead of a regular laptop to reduce runtime
Use manual calculation mode when working on / improving a very slow or bloated model
Excel worksheet code / programming tips
Limit the use of (semi-)volatile functions, e.g., *NOW, TODAY, INDIRECT, RAND, OFFSET, INFO, CELL*
Out of alternative functions, choose the most efficient option: • *INDEX + MATCH* rather than *VLOOKUP / HLOOKUP* • *MAX(A1,0)* rather than *IF(A1>0, A1, 0)* • *IFERROR* rather than *IF(ISERROR)* • *SUMIF / AVERAGEIF / COUNTIF* *rather than* *SUMIFS / AVERAGEIFS / COUNTIFS*
Use cell names and named ranges for efficient updating of parameter values and making formulas easier to read and understand
Avoid creating a tangled web of cross-links between variables, references, cell names, formulas… for faster calculation and better transparency
Use helper columns instead of array formulas, as the latter take time due cell reference dependency
Convert formulas to values if they are static. Avoid repetition of data/calculations, if not needed or unused
Avoid using references to an entire row/column/sheet in formulas
Use VBA macros instead of overly complex or nested formulas
VBA code / programming tips
Turn off features such as animations, screen updates, automatic calculations and events while macros run
After recording a macro, if you can, it is always best to review and edit down the code to ensure only essential executable lines remain
Avoid using the 'Select method' (to explicitly select objects before manipulating them) to improve the performance of macros
The worksheet function RAND has better statistical qualities than RND in VBA, but is volatile
Make macros save numeric results rather than formulas, to retain worksheets light on recalculation
The use of worksheet functions (within VBA) is generally much faster than using VBA normal code

Conclusions

- HE models are often built in Excel, due to its wide-spread accessibility, user familiarity, and perceived transparency
- Methodological and computational advances have allowed HE models to become more sophisticated and to better reflect clinical reality over time
- Although modern programming languages are better suited to conduct complex, computationally demanding, and/or real-time analysis, ambiguity over the acceptance of such HE models by HTA bodies remains a barrier to adoption

REFERENCES

[1] Jalal (2017). Med Decis Making 37(7):735-746.
[2] Incerti et al (2019). Value Health 22(5):575-579.
[3] Baio & Heath (2017). Glob Reg Health Technol Assess 4(1): e3-e6
[4] Botchkarev (2015). Spreadsheets in Education (eJSiE): Vol. 8: Iss. 2, Article 3
[5] Hollman et al (2017). PharmacoEconomics 35(8):817-830.
[6] Hart et al (2020). PharmacoEconomics 38, 765-776.
[7] Xin et al (2022). Applied Health Economics and Health Policy 20(2), 149-158.
[8] Poirrier et al. (2023) Poster MSR85 presented at ISPOR 2023, Boston, MA, USA
[9] Pitman et al. (2012). Value Health 15(6), 828-834

© 2023 Parexel International (MA) Corporation

www.parexel.com