SOFTWARE TOOL ARTICLE

assertHE: an R package to improve quality assurance of HTA models

[version 1; peer review: 1 approved with reservations]

Robert A. Smith[1,2], Yevgeniy Samyshkin[3], Wael Mohammed[1,2], Felicity Lamrock[4], Tom Ward[3,5], Jack Smith[1], Alan Martin[3], Paul Schneider[1,2], Dawn Lee[6], Gianluca Baio[7], Howard Thom[8], Nathan Green[7], Marina Richardson[9], Mohammed El Alili[10,11], Xavier Pouwels[12], Calum Lewis[1,2], Baris Deniz[3,13]

[1]Dark Peak Analytics, Sheffield, UK
[2]The University of Sheffield, Sheffield, UK
[3]GSK, London, UK
[4]Mathematical Sciences Research Centre, Queens University Belfast, Belfast, UK
[5]University of Exeter Medical School, Exeter, UK
[6]PenTAG, University of Exeter Medical School, Exeter, UK
[7]University College London Department of Statistical Science, London, UK
[8]Department of Population Health Sciences, University of Bristol Medical School, Bristol, UK
[9]Institute for Clinical and Economic Review (ICER), Boston, MA, USA
[10]Department of Health Sciences, Faculty of Science, Vrije Universiteit Amsterdam, Amsterdam Public Health Research Institute, Amsterdam, The Netherlands
[11]National Health Care Institute (ZIN), Diemen, The Netherlands
[12]Section of Health Technology and Services Research, University of Twente, Overijssel, The Netherlands
[13]Current Affiliation: AIde Solutions LLC, Wilmington, Delaware, USA

v1 **First published:** 02 Dec 2024, **9**:701
https://doi.org/10.12688/wellcomeopenres.23180.1
Latest published: 02 Dec 2024, **9**:701
https://doi.org/10.12688/wellcomeopenres.23180.1

Abstract

Background

Health economic models are increasingly used to inform decisions about the allocation of healthcare resources. Ensuring the robustness and reliability of these models is critical. Currently, quality assurance is conducted by both technical and non-technical experts assessing different components of the model manually. This is resource intensive. Understanding how the different components of the model fit together is time consuming, and testing every part of the model is sometimes not feasible in the time available. To aid in this, we have developed the assertHE R package.

Open Peer Review

Approval Status ?

	1
version 1 02 Dec 2024	? view

1. **Alexandra Hill**, London School of Hygiene & Tropical Medicine, London, UK

Any reports and responses or comments on the article can be found at the end of the article.

Methods

The open-source assertHE package provides testing functionality for those building and reviewing health economic models built in R programming language. It provides a series of checks which can be integrated into the model development workflow to reduce the probability of common errors. It also provides a suite of functions which allow users to better understand the network of R functions contained in a model, where they are defined, if (and where) they are tested, and provides a simple metric to quantify the extent to which they are tested.

Results

We applied the assertHE package to three open-source health economic models built in R, showing how to include check functions within the model code and how to visualise the network of functions, see the test coverage, and obtain a Generative Pretrained Transformer Large Language Model (GPT-LLM) generated summary of any function in the code-base. We have worked with collaborators from industry, regulators and academia to develop the package to be applicable to the widest possible range of models, making adaptations to the source code based upon feedback.

Conclusions

assertHE offers an open-source toolkit for health economists building and reviewing models, promoting collaborative development and facilitating a more robust and efficient quality assurance process.

Keywords
HTA, Health Economics, Unit Testing, Model Validation, Model Verification

Corresponding author: Robert A. Smith (rsmith@darkpeakanalytics.com)

Author roles: Smith RA: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Project Administration, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Samyshkin Y**: Conceptualization, Supervision, Validation, Writing – Review & Editing; **Mohammed W**: Software, Visualization; **Lamrock F**: Writing – Review & Editing; **Ward T**: Conceptualization, Writing – Review & Editing; **Smith J**: Software; **Martin A**: Conceptualization, Writing – Review & Editing; **Schneider P** : Conceptualization; **Lee D**: Validation, Writing – Review & Editing; **Baio G**: Writing – Review & Editing; **Thom H**: Writing – Review & Editing; **Green N**: Conceptualization, Software, Writing – Review & Editing; **Richardson M**: Validation, Writing – Review & Editing; **El Alili M**: Validation, Writing – Review & Editing; **Pouwels X**: Validation, Writing – Review & Editing; **Lewis C**: Writing – Review & Editing; **Deniz B** : Conceptualization, Supervision, Validation, Writing – Review & Editing

Competing interests: R.S, F.L, D.L, H.T, G.B, N.G are part of the committee for 'R for HTA', a consortium whose main objective is to explore the use of R for cost-effectiveness analysis. R.S, P.S, J.S, C.L and W.M work for Dark Peak Analytics Ltd; a company specialising in the application of techniques similar to those discussed in this paper in epidemiology and health economic evaluation. Y.S, A.M, S-J.A, and T.W are employees and shareholders of GSK. B.D was an employee and shareholder of GSK at the time of the study. D.L is a shareholder in Value Demonstration Group and a member of Fiecon's strategic council.

Grant information: This work was supported by Wellcome [108903 to R.S., P.S. and W.M. https://doi.org/10.35802/108903] and the University of Sheffield. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

How to cite this article: Smith RA, Samyshkin Y, Mohammed W *et al.* **assertHE: an R package to improve quality assurance of HTA models [version 1; peer review: 1 approved with reservations]** Wellcome Open Research 2024, **9**:701 https://doi.org/10.12688/wellcomeopenres.23180.1

First published: 02 Dec 2024, **9**:701 https://doi.org/10.12688/wellcomeopenres.23180.1

Introduction

Cost-effectiveness models play a crucial role in health economic evaluations, aiding the allocation of limited healthcare resources. These models aim to simulate the outcomes of alternative courses of action relating to a specific decision problem.

The computation of health economic models has traditionally been conducted in spreadsheet software like MS Excel. While MS Excel is widely accessible and familiar to many users, it has proven to be error-prone in many applications, including health economics[1–3]. Checking the robustness of these models often involves a laborious manual review process, in which cells of the spreadsheet have to be checked individually for errors. This manual process is not only time-consuming but also subject to human error, making it an inefficient method for ensuring the quality of health economic models.

There is a recent growing shift towards script-based programming languages in Health Economics and Outcomes Research (HEOR). In particular, the R programming language is increasingly used for developing decision analytical models[4]. This transition is occurring because of the improved capabilities, reproducibility, ease of collaboration, and reduced model run-time of R compared to spreadsheet software[5], and because the increasingly complicated statistical analysis used to inform models is already typically undertaken in R. Models developed using best-practice methods in R generally consist of a network of functions. Each function is defined in a single location, either within the model code-base or from external software packages[6,7].

This move toward the development of models as software allows for automated testing and quality assurance techniques to reduce the probability of human error, which are standard practice in mission critical applications in the financial sector, aerospace, and other industries, but are not yet widely adopted in health economic evaluation[6,7]. The ISPOR-SMDM Modeling Good Research Practices Task Force–7 recommendation V11-4 states that *"Models should be subjected to rigorous verification"* and that *"Coding accuracy should be checked using state-of-the-art quality assurance and control methods for software engineering"*, including *"verifying separate parts of a model one by one"*[8] (p. 846). Alarid-Escudero *et al.*[7] provide a summary of the benefits of testing for health economic models built in R, proposing that these are built into the model development process, with modellers *"writing tests alongside the development of any new function or process or whenever a bug is found. This practice results in a high level of test coverage of the analysis code, reducing the likelihood that unintended interactions or incompatibilities between functions and/or processes will go undetected"* (p. 1335).

There are several existing tools used to aid in verification of models. AdViSHE is a validation assessment tool which provides a framework for modellers to report the validation undertaken[9]. This tool can be used by modellers and reviewers to assess the extent to which a model has been validated. The tool has 13 parts. This paper relates to the 4 contained in part C - Validation of the computerised model (sometimes called verification, internal validity, internal consistency, technical validity, and debugging). C1 advocates for External review, which the package aims to facilitate, C2 for Extreme Value testing, C3 for Testing of traces and C4 for Unit testing, all of which are the focus of the R package described in this paper. Another tool designed for unit testing, the Probabilistic Analysis Check dashBOARD (PACBOARD) tool, provides software and a user-interface which enables users to explore and validate the inputs and outputs of the probabilistic analysis outputs of health economic model[10]. Additionally, the Canadian Agency for Drugs and Technologies in Health (CADTH) model validation framework (now in it's second iteration) serves as a comprehensive resource for ensuring the robustness and credibility of economic models in health technology assessments, offering detailed guidelines and methodologies that can be applied across different stages of model development[11].

This paper introduces the `assertHE` R package, which aims to make it easier to build and review robust health economic models in R. The benefits of the package differ by user. From the perspective of the modeller, the package provides a suite of functions which can be used to reduce the probability of human error occurring in iterative model development processes[12,13]; and can help to map out the relationships between different parts of the model, to plan new additions to the model code and identify redundant code. From the perspective of the reviewer, the package provides a visual summary of the network of functions that make up the model - identifying where each function is defined and tested. Test coverage, which can be measured in different ways, but at its simplest is the percentage of the code executed within unit tests, is also reported for each function. The package also allows the user to obtain a short summary of any function in the network. This summary is generated using a GPT LLM, often referred to as Generative AI, prompted with the function arguments, body, documentation and other information about the function generated programmatically.

The overall effect of this is to make model development and review of health economic models developed in R more systematic, quicker and easier. The intended result is to shift the burden of proof of the quality of the model onto the modeller. In this context, the assumption is 'guilty unless proven innocent'. The software package provided makes it easier for modellers to test their own code, and providing a visual representation of the function network with accompanying testing coverage makes this immediately clear for all parties and can be used to guide discussions. This may be especially useful in the (sub-optimal) instances in which it is not possible to make the full source code available, since third parties are provided with a summary of the model structure, extent of testing, and model dependencies.

This is not the first paper to provide more specific verification tools based on AdViSHE. Dasbach and Elbasha[14] provide the justification for several common tests including tests on Life Expectancy, Quality-Adjusted Life Expectancy, Intervention Costs and Cohort Size. The authors include an example set of tests for MS Excel. We adapt and extend these tests, coding them into an R package which can be downloaded from GitHub using a single line of R code. We provide instructions for how potential collaborators can request to add their own tests to the package, suggest improvements to existing tests, or help us to build software that performs tests already identified by others as useful. Also of note is the `darthtools` R package[15], which already includes some testing functionality, for example functions to check transition probabilities and to check whether progression free survival exceeds overall survival in extrapolated survival curves. This package aims to build upon these checks, focusing exclusively on model review. To the authors knowledge there is no other existing software which produces a visual representation of a model structure, or provides AI generated summaries of R functions contained in a project.

The remainder of this paper broadly describes the functionality of the `assertHE` package (methods), showcases two case studies previously described in Krijkamp *et al.*[16] and Alarid-Escudero *et al.*[17] and applied in previous tutorial papers in a script-based format[18,19] and describes the implications for development and review of models in R more broadly. Finally, it outlines the potential use for this software, its strengths and limitations and potential future research which could build upon this work.

Methods

Implementation

The `assertHE` package is designed to be used in two main ways. Firstly, it can be used by a modeller to aid in the creation of unit tests, the isolation and examination of individual components (generally functions) within the model, to ensure the code is performing as intended. These tests are designed to be repeatable, quick to execute, and independent of each other, allowing for early detection of issues in the development process. Since we have pre-created some of the common tests, experienced modellers may wish to incorporate the tests into their model code, fostering a proactive approach to bug detection and promoting software quality through early identification and resolution of issues - this is referred to by software engineers as Test-Driven Development (TDD)[20]. Secondly, `assertHE` is designed to be used by reviewers who want to visualise the network of functions used in the model, and the coverage of existing tests. This visualisation (also exportable as a data-frame) can be used as a checklist when reviewing the model, as a starting point before adding additional tests as required. The aim in both instances is to make models more transparent and to reduce the burden of verification[21].

Detailed documentation for each function can be found in the software package repository. This provides information on each function, what it does and its inputs and outputs. However, some examples of checking functions are listed below:

- `check_trans_prob_array()` Checks transition probability array for common errors ensuring the same number of rows and columns, only numerical values between 0 and 1, with rows summing to 1. The function provides confirmation or warnings/error messages as appropriate.

- `check_markov_trace()` Validates trace for feasibility, confirming numerical values, equal row sums, and optionally monotonic increases or decreases in state proportions (e.g. for dead state). Generates confirmation or warnings/error messages as required.

- `check_init()` Checks initial health state proportions vector for validity, ensuring values within valid probability range, no missing values, sum equals 1, and distinct, non-duplicated names, generating messages for inconsistencies.

- `plot_PSA_stability()` Generates informative plots to inspect stability of Probabilistic Sensitivity Analysis (PSA) results across model iterations, calculating metrics like incremental costs, incremental effects, Incremental Net Monetary Benefit (INMB) and Incremental Cost-Effectiveness Ratio (ICER), thereby allowing customization for specific needs and enhancing visual clarity.

The model overview functions, e.g. `visualise_project()` and `summarise_model()`, provide a visual and tabulated summary of any R model that has a set of functions in one folder (commonly `R/`). The visual takes the form of a network diagram showing the relationship between the functions (shown as nodes) in the model. The size of each node is dependent on the centrality of the function in the network. If the model has a set of tests in another folder (commonly `tests/testthat`), information on test coverage for each function can be included in the summary table and network diagram (nodes color is dependent on coverage percentage). From the interface that is generated by the function, it is possible to see or navigate to the source code for any function, the source code for any corresponding test, or to generate a short non-technical text summary of what the function does. This summary is generated by creating a custom 'prompt' which contains the function arguments, code and documentation along with information on testing coverage on dependencies, and sending this to a GPT LLM (currently Open-AI's gpt-3.5-turbo-0125) via an Application Programming Interface (API). To do this, internet connection and an API Key to the GPT LLM is required. A summary of this process can be seen in Figure 1.

The `assertHE` package has been developed using the same framework of unit testing that it proposes. Each function has associated unit tests that are run using GitHub Actions every time a change is made to the codebase[22].

Operation

To install and use `assertHE`, users need R version 3.5.0 or higher. The package imports several dependencies, including `assertthat`, `ggplot2`, `scales`, `tidyr`, `stringr`, `dplyr`, `utils`, `data.table`, `visNetwork`, `covr`, `htmltools`, `officer`, `flextable`, `knitr`, `here`, `shiny`, `shinyjs`, `rstudioapi`, `roxygen2`, `methods`, `waiter`, `igraph`, and `httr`.

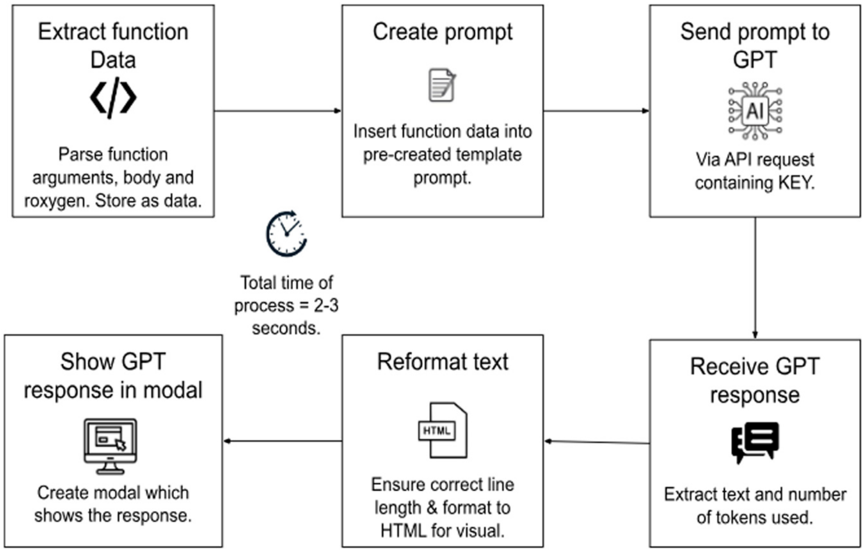

Extract function Data
Parse function arguments, body and roxygen. Store as data.
Create prompt
Insert function data into pre-created template prompt.
Send prompt to GPT
AI
Via API request containing KEY.
Total time of process = 2-3 seconds.
Show GPT response in modal
Create modal which shows the response.
Reformat text
HTML
Ensure correct line length & format to HTML for visual.
Receive GPT response
Extract text and number of tokens used.

Figure 1. The process of summarising a function for a user from the network diagram using a GPT-LLM accessed via an API request.

Use cases

The `assertHE` package has been used in a few ways. Firstly, it was trialled on the sick-sicker model described in previous publications[6,18,19]). This is used as a vignette to showcase the functionality of the package, and is the focus of the majority of this section. Secondly, it was used to visually inspect two open-source models: a cost-effectiveness model of testing average-risk Stage II colon cancer patients for the absence of a CDX2 biomarker expression (cdx2cea model), and the NICE Renal Cell Carcinoma pathways pilot cost-effectiveness model (RCC model)[23,24]. Finally, it has been used on closed source models by HTAs, consultancies and industry. The model developers provided feedback on the tool, resulting in iterative improvements in its functionality. We acknowledge those contributions at the bottom of the package README.

The sick-sicker model

In this section, we showcase the functionality of the `assertHE` package on the sick-sicker model described in previous publications[6,16–19]. This includes inserting the checks into the model functions while developing the model code, allowing for a quick identification of errors or irregularities. An example of this implementation can be seen on Github.

Within the model code, the health economic modeller or reviewer can insert a check into the model code in a single line. For example the `check_trans_prob_mat()` function checks that a transition probability matrix (e.g. `m_P` in the example below) conforms to a set of rules including that values are between 0 and 1, the matrix is square (same number of rows and columns) and that the names (if any) of rows and columns match. This gives some assurance that a transition matrix object is credible, without requiring a large quantity of code to be included directly into the model, which would make it more difficult to adapt and review. This function can be used multiple times throughout the model to review different transition matrices, and an alternative function `check_trans_prob_array()` is available for three-dimensional transition probability arrays commonly used where transition probabilities are time dependent.

The code below creates the visual representation of the network of model functions, with each function represented as a circular node. The size of each node relates to the out-degree centrality of the function, the number of functions called by that function, created using the `igraph` R package[25]. Arrows point from the function being called to the function calling (from child to parent). Colors are used to display testing coverage with custom colors and thresholds passed to the function. Information about each function can be seen by hovering over the function. This information includes the location of the function definition, the location of any tests of the function, and the test coverage.

The code below generates the visual representation. It requires the user to provide the path of the project, the location of the functions (normally "R") and the locations of tests. Calculating test coverage is optional, since some projects may be set up in such a way that it is not possible to test coverage, or the tests fail or break. If the coverage is run and the tests fail then the visual will still appear, albeit with a warning message.

```r
visualise_project(
  project_path = "path_to_project_directory",
  foo_path = "R",
  test_path = "tests/testthat",
  run_coverage = T)
```

The package also includes an RStudio add-in[26], allowing users of RStudio to click on a button on the RStudio IDE and then use an interface to select the arguments to `visualise_project()` rather than writing code executed in the console. This aims to make `assertHE` easier to use for those who may have access to the model in RStudio but may have limited programming skills.

The resulting visual can be seen in Figure 2. The nodes in green have testing coverage > 80%, in orange testing coverage between 20–80% and in red either no test or coverage < 20%. These thresholds have been chosen as sensible defaults but can be adjusted by users of the package. In Figure 3, hovering over a single node reveals the location of the function definition, the test location and the coverage metric. Users can either click on the eye symbol to see the file in the app, or the navigate symbol to navigate to the file and line number in RStudio. The robot symbol returns the GPT-LLM generated plain language summary of the function as seen in Figure 4.

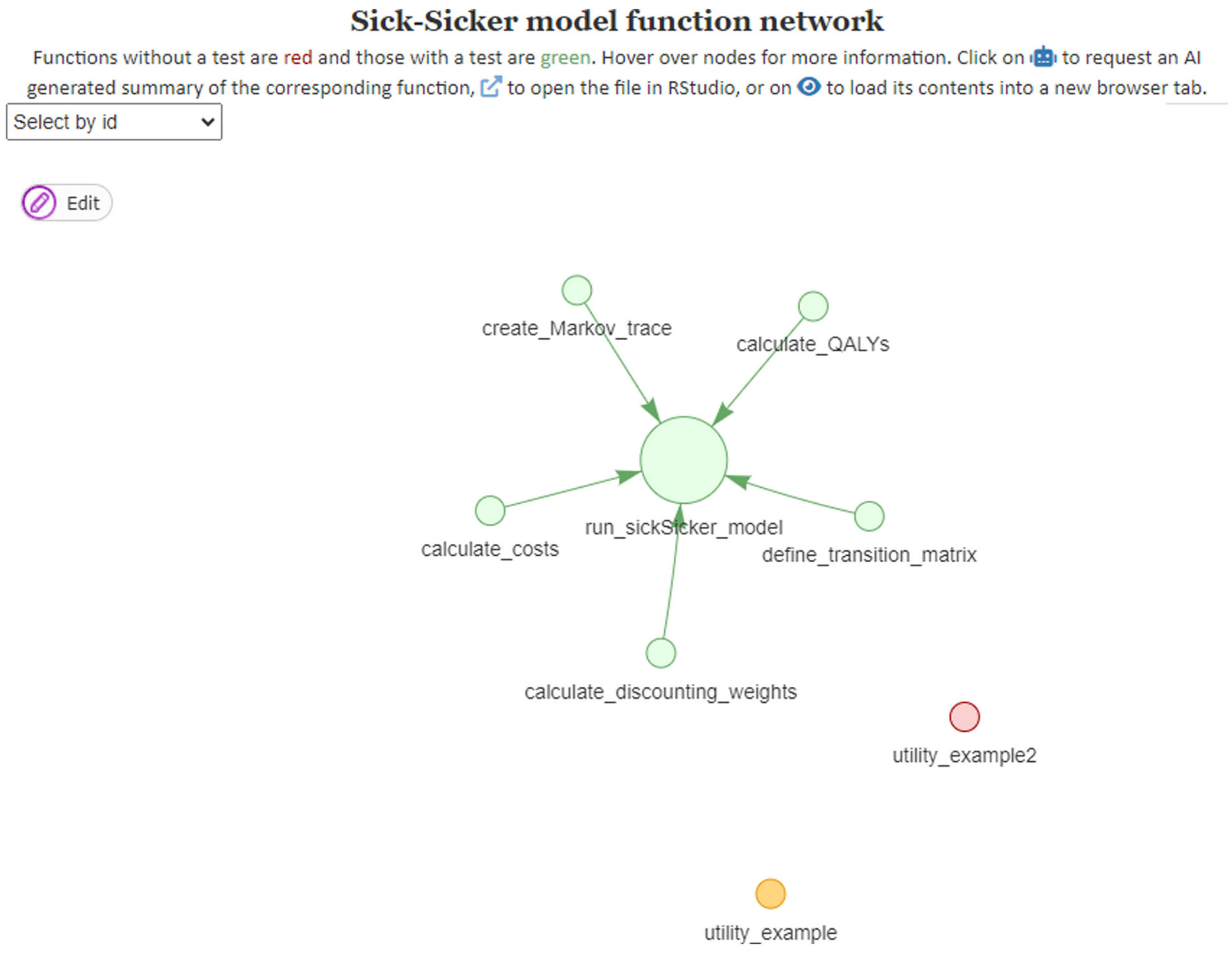
The cdx2cea model

The cdx2cea model performs a cost-effectiveness analysis (CEA) of testing average-risk Stage II colon cancer patients for the absence of CDX2 biomarker expression followed by adjuvant chemotherapy. It is described in detail in Alarid-Escudero *et al.*[23]. The data and code were made available in R package format with an MIT Licence on GitHub with a version released via Zenodo.

The cdx2cea model was used to help guide development of `assertHE`. The function network for local functions (i.e. not those functions called from external packages) can be seen in Figure 5. In this model, only a few key functions, `decision_model`, `calculate_ce_out` and `load_all_params` are tested in the tests folder. Note the interconnectedness of a large number of the functions (top centre), made more transparent to a prospective reviewer with a visual representation, relative to attempting to follow nested functions through multiple R scripts.

NICE Renal Cell Carcinoma Pathways Pilot

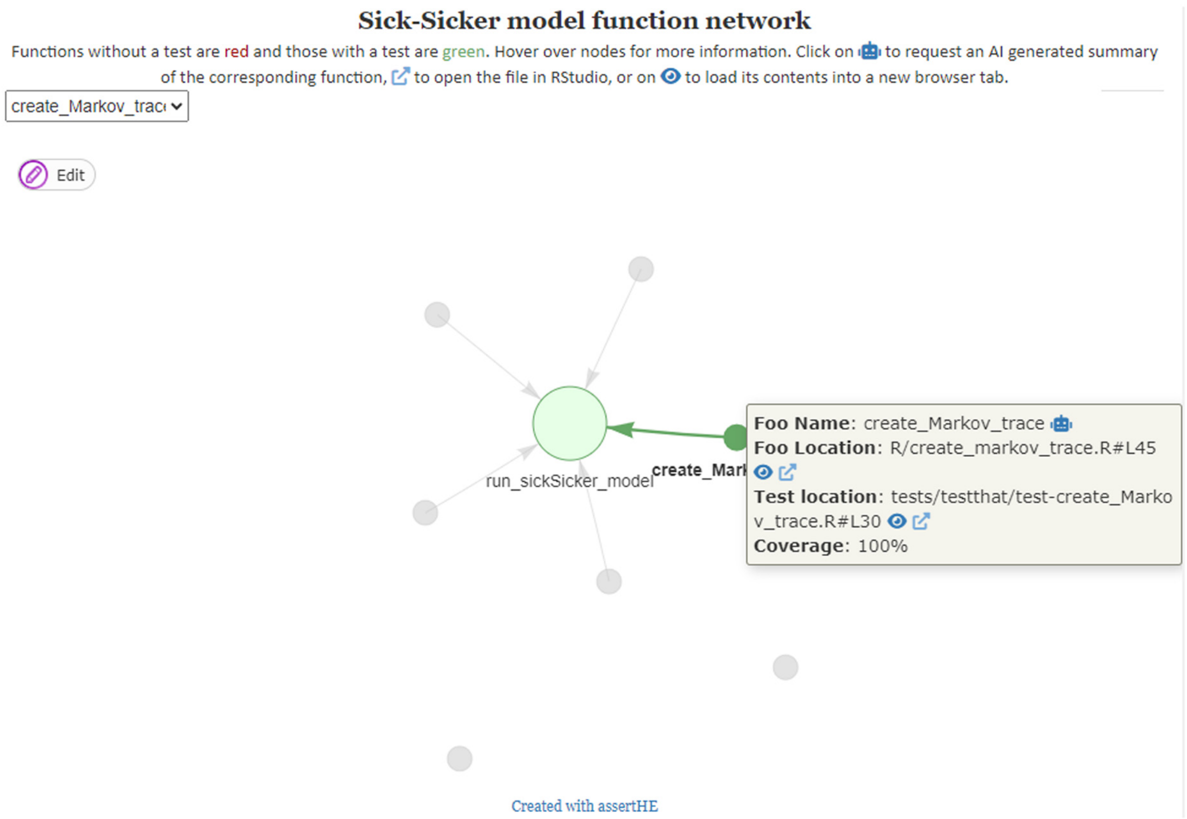
The Exeter Oncology Model, developed as part of the NICE Renal Cell Carcinoma (RCC) Pathways Pilot project, is the first model developed using NICE's proportionate approach



Sick-Sicker model function network

Functions without a test are red and those with a test are green. Hover over nodes for more information. Click on 🤖 to request an AI generated summary of the corresponding function, ↗ to open the file in RStudio, or on 👁 to load its contents into a new browser tab.

Figure 2. Function network for the sick-sicker model developed in R, with the colours indicating testing coverage.

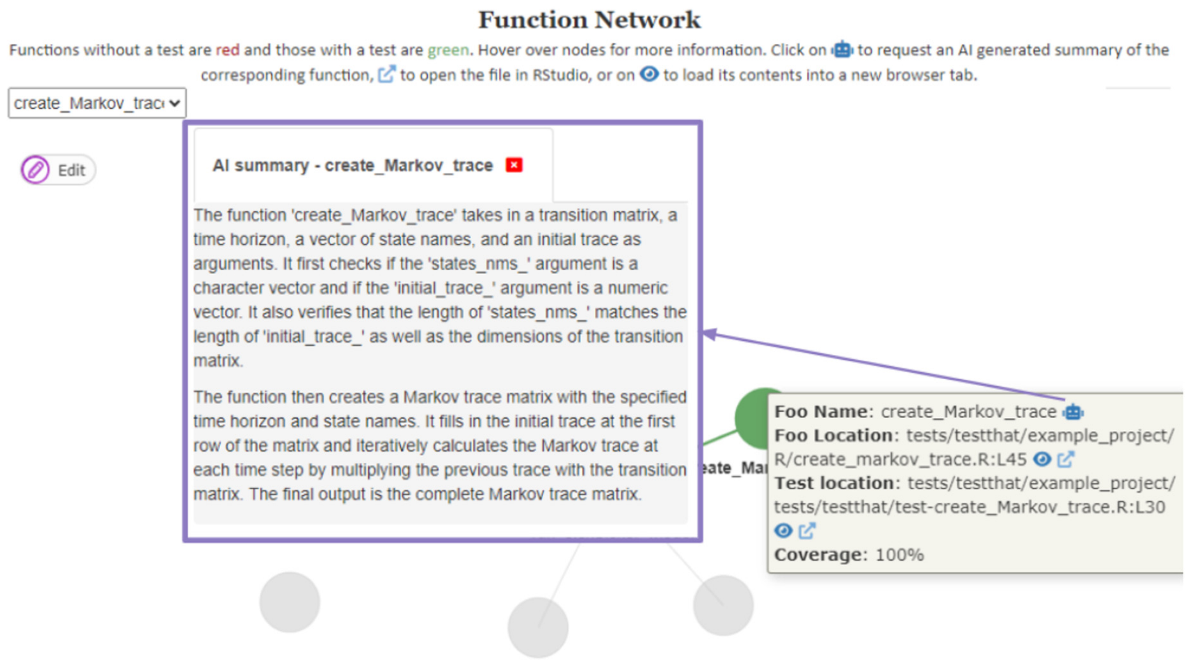


Sick-Sicker model function network

Functions without a test are red and those with a test are green. Hover over nodes for more information. Click on 🤖 to request an AI generated summary of the corresponding function, ↗ to open the file in RStudio, or on 👁 to load its contents into a new browser tab.

Foo Name: create_Markov_trace 🤖
Foo Location: R/create_markov_trace.R#L45
👁 ↗
Test location: tests/testthat/test-create_Markov_trace.R#L30 👁 ↗
Coverage: 100%

Figure 3. **Function network for the sick-sicker model developed in R, with definitions and locations.**



Function Network

Functions without a test are red and those with a test are green. Hover over nodes for more information. Click on 🤖 to request an AI generated summary of the corresponding function, ↗ to open the file in RStudio, or on 👁 to load its contents into a new browser tab.

AI summary - create_Markov_trace ☒

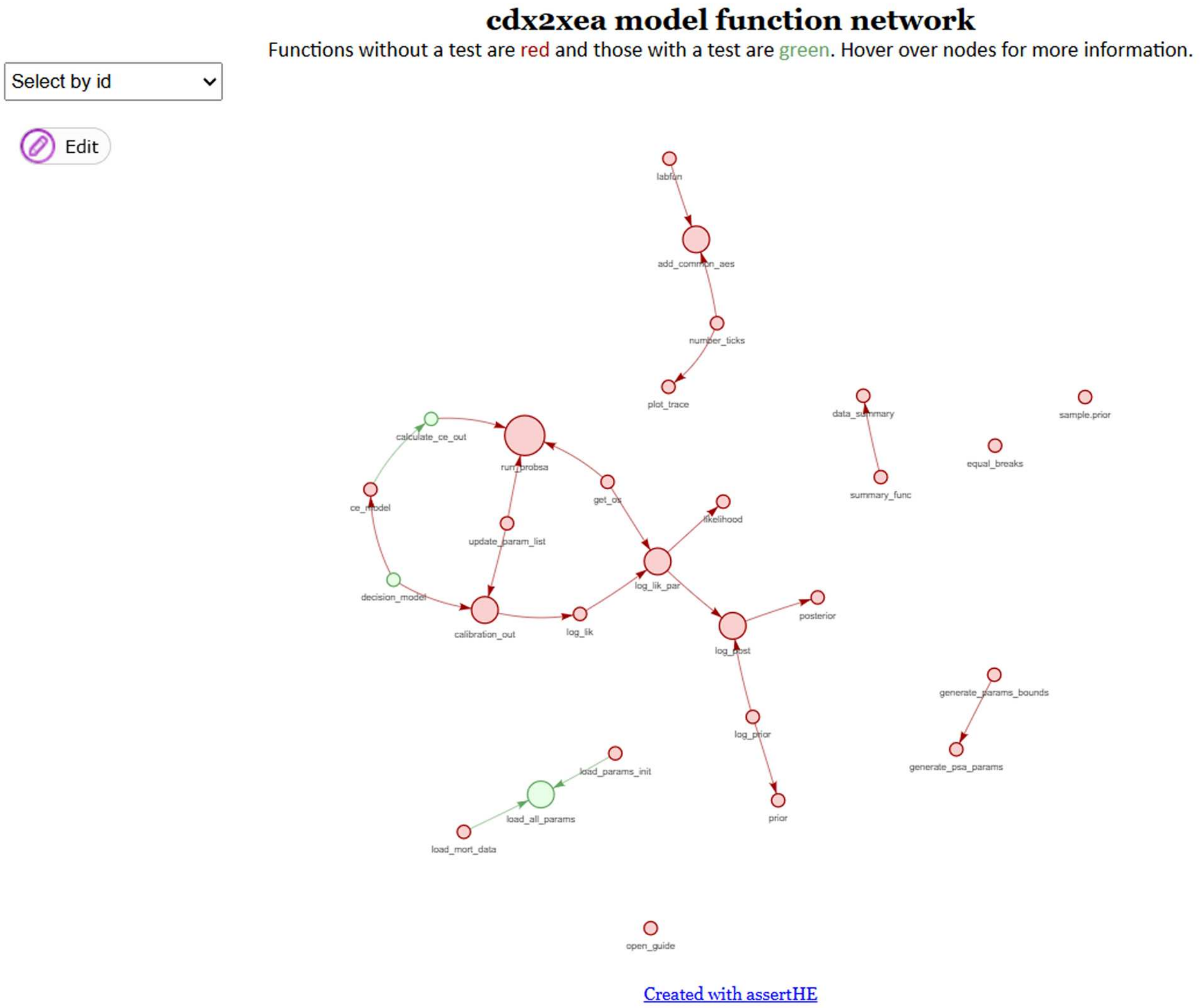
The function 'create_Markov_trace' takes in a transition matrix, a time horizon, a vector of state names, and an initial trace as arguments. It first checks if the 'states_nms_' argument is a character vector and if the 'initial_trace_' argument is a numeric vector. It also verifies that the length of 'states_nms_' matches the length of 'initial_trace_' as well as the dimensions of the transition matrix.

The function then creates a Markov trace matrix with the specified time horizon and state names. It fills in the initial trace at the first row of the matrix and iteratively calculates the Markov trace at each time step by multiplying the previous trace with the transition matrix. The final output is the complete Markov trace matrix.

Foo Name: create_Markov_trace 🤖
Foo Location: tests/testthat/example_project/R/create_markov_trace.R:L45 👁 ↗
Test location: tests/testthat/example_project/tests/testthat/test-create_Markov_trace.R:L30
👁 ↗
Coverage: 100%

Figure 4. **GPT-LLM generated plain language summary of a function.**

Wellcome Open Research 2024, 9:701 Last updated: 03 DEC 2024



cdx2xea model function network

Functions without a test are red and those with a test are green. Hover over nodes for more information.

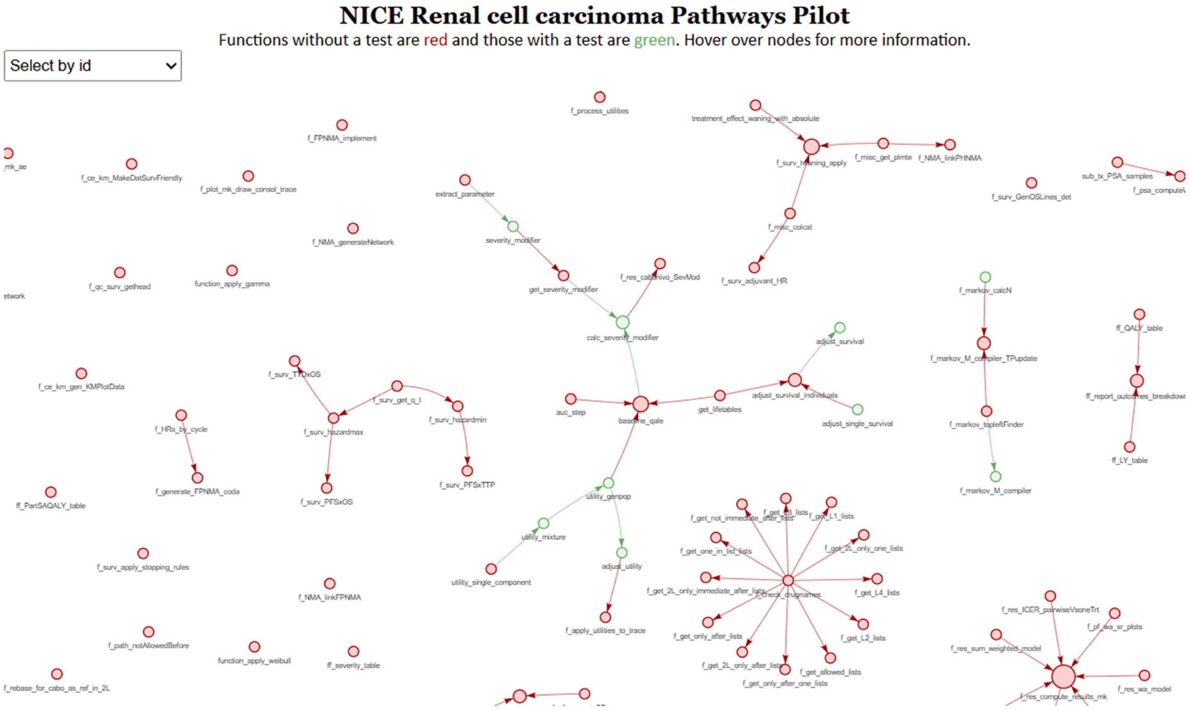
Figure 5. Function network for cdx2cea R model.

to technology appraisals. The aim and process of the model build is described in Lee *et al.*[24], and the code is made publicly available at (the somewhat pessimistically named) 'NICE Model Repo'. The code-base is extensive, with approximately 25,000 lines of code. The resulting function network visual, shown in Figure 6 is therefore much larger than the sick-sicker and cdx2cea models. Functions are stored in `3_Functions`. While unit testing was not in the scope of the project, some tests were written during the development process and included in `tests/testthat`.

Requirements of a model to be suitable for `assertHE`

The current functionality of `assertHE` is primarily tailored for state transition models utilising transition probability matrices or arrays for constructing traces. To broaden the software's applicability, we are eager to engage the wider community in suggesting and developing checks for other types of health economic simulation models. These may include partitioned survival models, which partition survival extrapolations into discrete time units to estimate population proportions in various health states over time, and microsimulation models, where individual agents are simulated and results aggregated to estimate impact at the entire population level. We welcome community involvement to expand `assertHE's` capabilities and enhance its utility in health economic evaluations.

To generate a visual representation of the model using `assertHE`, it's essential to construct the model using functions, which should be organised within a single directory (e.g., an R folder). To generate information about the existence of tests, tests must be included in a separate folder (e.g., tests). Additional information on code coverage can only be provided if the code complies with the `covr` package's code coverage requirements, most importantly that tests pass[27]. While these requirements align with standard coding practices in the R programming language[28], it's important to note that the use of R in health economic evaluation is still in its early stages, leading to variations in coding practices among models. By adhering



NICE Renal cell carcinoma Pathways Pilot

Functions without a test are red and those with a test are green. Hover over nodes for more information.

Figure 6. Function network for RCC Pathways R model.

to these practices, assertHE not only facilitates the creation of visual representations but also encourages the adoption of good coding practices within the health economic research community, ultimately enhancing collaboration and reproducibility.

Discussion

Health economic models are typically conceptualised and described in words and figures and converted into a mathematical model, often with only important or novel subsets of the equations provided. However, it is the computational model that generates the results, and the full computational model is often only provided to a small number of reviewers, who have limited time and resources to review. It is crucial for decision-makers, who often do not have the technical skills to verify the model itself, to have confidence in the computational model[29].

The hope is that assertHE will make model development and review more efficient and reduce the number of errors in models written in the R programming language, thereby improving the confidence of decision-makers in models. There may also be indirect benefits if it encourages the wider use of formal testing frameworks in health economic models, or mitigates against the reluctance of modellers to make their code open-source[30]. It may also contribute to encourage the sector toward the use of R for health economic models, resulting in other significant benefits including reduced model run-time, incorporation of more advanced statistical techniques, easier collaboration and more efficient model updates[31]. In particular, being able to visualise and explore the relationship between the algorithms of the model is a significant advantage over spreadsheet software. However, it is not a silver bullet. Improving verification of the model code does not validate the model's premise, parameters and structural assumptions. This must be done separately[32].

Strengths

There are several strengths of this work:

- The extensive documentation provides examples & case studies of the use of the tool on open-source models.
- The package provides standardised testing functions to make the most common checks more efficient, freeing up time for modellers and reviewers to critique models in more innovative ways.
- It aligns with best practices recommended by ISPORSMDM Modeling Good Research Practices Task Force–7 for rigorous model verification.
- The authors come from a variety of organisations, providing a diversity of opinion and objectives amongst modelling applications.

Limitations

- The package assertHE is in development phase. A first version has just been submitted to CRAN (a requirement for use within agencies such as ZIN).

- This paper primarily focuses on R, limiting applicability for health economic models developed in other programming languages. However, functionality does exist to run R code from Python via the software r2py[33].
- Most of the functions are developed for State Transition Models, although they can be fairly simply edited for other types of health economic models.
- The long-term maintenance and evolution of the assertHE package may be subject to challenges, especially if community contributions are limited.
- There is a risk that modellers and reviewers are over-reliant on the checks provided. It should be frequently noted that the functions provided are not exhaustive and further review should be undertaken. In particular assertHE does not, and likely will never be able to, provide checks for the clinical validity of the models in the specific disease area they are developed for.
- Outputs returned from pretrained transformer models like OpenAI's GPT models should be treated with caution. In particular there is a risk that models will be biased by the models on which it is trained, and may not be sufficiently intelligent to recognise particularly novel code. It is also likely to be biased by any comments in the code.
- The assertHE visualisation functions assume that the model is structured as a network of functions. Models that are scripted without the use of functions (e.g. scripts that source one another and store data for intermediate steps) will not be well described by this function.
- As with all R packages, assertHE must be installed prior to use.

Further research

This paper is only intended to be the beginning of the development of assertHE. We invite open collaboration to continuously improve and expand the open-source R package before we submit to the Comprehensive R Archive Network (CRAN)[34]. It's essential to assess whether and how much the package improves and speeds up the review of health economic decision models in practice. This may involve an examination of its applicability across diverse health economic model types such as microsimulation, decision-tree, and agent-based models, to evaluate its generalizability. it could also incorporate information about external packages used and cross-reference to HTA agency lists of approved packages[35]. HTA agencies and funders could play an important role in improving modelling practices by continuing to recommend testing and other good software development practices in submissions.

To enhance the user experience and promote broader adoption, further research should delve into the impact of user proficiency in R programming on assertHE's effectiveness. Developing and evaluating training resources to complement existing publications[6,18,19,36–39] that cater to users with varying levels of programming experience will be crucial in facilitating a wider adoption of the tool. It is also worth exploring the feasibility of adapting assertHE for health economic models developed in languages other than R, particularly Python. This investigation should include an assessment of the challenges and benefits associated with extending assertHE's functionality to accommodate other programming languages that may become commonly used in the health economics domain.

Securing assertHE's long-term future hinges on actively exploring ways to maintain and improve it. If health economists working in R can be encouraged to contribute their tests, the tool will become increasingly useful and therefore will in-turn engage people. Engaging health economists, training them in its use, and in how to contribute, will be essential. Building confidence with HTA agencies, with a view for long-term use as standard practice once a critical mass of use and testing has been achieved, could increase the impact of the software.

The sustainability of assertHE requires a commitment to software maintenance and adaptation in the long-term. To achieve this it will be important to incentivise collaboration from the health economics community. Where opportunities exist to integrate assertHE with popular R packages commonly employed by health economists, such as heemod[40] and hesim[5], these could be encouraged since integration could enhance the accessibility and usability of assertHE for a broader audience, extending its impact beyond experienced R users.

Future research could also include fine-tuning a large language model on health economics-specific content, such as health economics textbooks and open-source R models, to potentially enhance the performance of the model currently used in the prototype described in this paper. This improvement could be evaluated by an independent group of health economists, who would assess both the quality of the current summaries and the impact of finetuning. Additionally, integrating a chatbot into the code would enable users to interact with the model, facilitating discussions about its outputs. Further research should also explore the broader implications of verification software like assertHE on the willingness of modellers to share their code as open-source. Finally, it is crucial to explore the opportunities and risks associated with a future where models and reviews are largely conducted by generative AI.

The checks provided are not intended to be exhaustive. The intention is that others will contribute checks as they use them. This can be done by submitting a pull request to the package on GitHub (with all automated checks shown to pass), or creating an issue describing the checks that would be required, ideally with an example of the code needed to implement them. We would be particularly interested in checks that can be performed outside of model functions, to assess the external validity of models, as described in a recent model validation checklist created by CADTH[11]. More information on how to contribute can be found at the package's Github repo.

Summary

This paper has introduced `assertHE`, an R package designed to enhance the efficiency of the verification process for health economic models and reduce the likelihood of errors. The case study, using the sick-sicker model, illustrates the practical application of `assertHE` on a health economic model, using a continuous integration framework. This automated testing approach with each model revision aims to reduce the risk of introducing bugs during model edits. Meanwhile the visualisation and summary functionality aims to make it easier for external reviewers to understand how the individual algorithms in the model fit together. Used alongside other tools, such as quality checklists and tests of external validity, `assertHE` may help increase confidence in the veracity of health economic models, allowing decision-makers to make better-informed choices regarding the allocation of limited healthcare resources, contributing to improved population health outcomes.

Ethics and consent

Ethical approval and consent were not required.

Data and software availability

No data are associated with this article.

Source code available from https://github.com/dark-peak-analytics/assertHE

Archived software available from: https://doi.org/10.5281/zenodo.13969178[41]

License: MIT

Author contributions

R.S Conceptualization, Software, Validation, Methology, Project Administration, Writing - original draft, Writing - review and editing, Visualization. Y.S, A.M, S-J.A and T.W - Conceptualization, Writing - review and editing, Validation J.S and W.M Software, Methodology and Visualisation P.S, D.L, G.B, H.T, N.G, B.M, M.R, M.E-A, X.P, C.L - Writing - review and editing

Acknowledgments

We would like to thank all those at R-HTA 2024 workshop and R-HTA hackathon for their useful comments and feedback, as well as those who tested the models internally, in particular Niall Davison and Zachary Waller.

References

1. Radeva D, Hopkin G, Mossialos E, *et al.*: **Assessment of technical errors and validation processes in economic models submitted by the company for nice technology appraisals.** *Int J Technol Assess Health Care.* 2020; **36**(4): 311–316.
   PubMed Abstract | Publisher Full Text
2. National Institute for Health and Care Excellence (NICE): **Ribociclib with an aromatase inhibitor for previously untreated, hormone Receptor-positive, HER2-negative, locally advanced or metastatic breast cancer**. 2018; Accessed: 2018-08-06.
   Reference Source
3. Panko RR: **What we don't know about spreadsheet errors today: the facts, why we don't believe them, and what we need to do.** *CoRR.* abs/1602.02601. 2016.
   Publisher Full Text
4. Jalal H, Pechlivanoglou P, Krijkamp E, *et al.*: **An overview of R in health decision sciences.** *Med Decis Making.* 2017; **37**(7): 735–746.
   PubMed Abstract | Publisher Full Text
5. Incerti D, Jansen JP: **hesim: health economic simulation modeling and decision analysis.** *arXiv preprint arXiv: 2102.09437.* 2021.
   Reference Source
6. Smith R, Mohammed W, Schneider P: **Packaging cost-effectiveness models in R: a tutorial [version 1; peer review: 2 approved with reservations].** *Wellcome Open Res.* 2023; **8**: 419.
   Publisher Full Text
7. 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): 1329–1339.
   PubMed Abstract | Publisher Full Text | Free Full Text
8. Eddy DM, Hollingworth W, Caro JJ, *et al.*: **Model transparency and validation: a report of the ispor-smdm modeling good research practices task force-7.** *Med Decis Making.* 2012; **32**(5): 733–43.
   PubMed Abstract | Publisher Full Text
9. Vemer P, Ramos IC, Van Voorn GAK, *et al.*: **AdViSHE: a validation-assessment tool of Health-Economic models for decision makers and model users.** *Pharmacoeconomics.* 2016; **34**(4): 349–361.
   PubMed Abstract | Publisher Full Text | Free Full Text
10. Pouwel XGLV, Kroeze K, van der Linden N, *et al.*: **Validating Health Economic models with the probabilistic analysis check dashboard.** *Value Health.* 2024; **27**(8): 1073–1084.
    PubMed Abstract | Publisher Full Text
11. Coyle D, Haines A, Lee K: **The development of a model validation tool to assist in the conduct of economic evaluations.** *Can J Med Technol.* 2024; **4**(3).
    Publisher Full Text
12. Wickham H: **testthat: get started with testing.** *R J.* 2011; **3**(1): 5–10.
    Reference Source
13. Büyükkaramikli NC, Rutten-van Mölken MPMH, Severens JL, *et al.*: **Tech-ver: a verification checklist to reduce errors in models and improve their credibility.** *Pharmacoeconomics.* 2019; **37**(11): 1391–1408.
    PubMed Abstract | Publisher Full Text | Free Full Text
14. Dasbach EJ, Elbasha EH: **Verification of decision-analytic models for health economic evaluations: an overview.** *Pharmacoeconomics.* 2017; **35**(7): 673–683.
    PubMed Abstract | Publisher Full Text
15. Alarid-Escudero F, Baio G, Enns E, *et al.*: **darthtools: decision-analytic routines for transparent healthcare models**. R package version 1.1.0, 2019.
    Reference Source
16. Krijkamp EM, Alarid-Escudero F, Enns EA, *et al.*: **Microsimulation modeling for health decision sciences using R: a tutorial.** *Med Decis Making.* 2018; **38**(3): 400–422.
    PubMed Abstract | Publisher Full Text | Free Full Text
17. Alarid-Escudero F, Krijkamp E, Enns EA, *et al.*: **A tutorial on time-dependent cohort state-transition models in R using a cost-effectiveness analysis example.** *Med Decis Making.* 2023; **43**(1): 21–41.
    PubMed Abstract | Publisher Full Text | Free Full Text
18. Smith R, Schneider P: **Making health economic models shiny: a tutorial [version 2; peer review: 2 approved].** *Wellcome Open Res.* 2020; **5**: 69.
    PubMed Abstract | Publisher Full Text | Free Full Text
19. Smith RA, Schneider PP, Mohammed W: **Living HTA: automating health economic evaluation with R [version 2; peer review: 2 approved].** *Wellcome Open Res.* 2022; **7**: 194.
    PubMed Abstract | Publisher Full Text | Free Full Text
20. Beck K: **Test-driven development: by example.** Addison-Wesley signature series. Addison-Wesley, 2003.
    Reference Source

21. Sampson CJ, Wrightson T: **Model registration: a call to action.** *Pharmacoecon Open.* 2017; **1**(2): 73–77.
PubMed Abstract | Publisher Full Text | Free Full Text

22. Wickham H: **assertthat: easy pre and post assertions**. R package version 0.2.1. 2019.
Reference Source

23. Alarid-Escudero F, Schrag D, Kuntz KM: **Cdx2 biomarker testing and adjuvant therapy for stage ii colon cancer: an exploratory cost-effectiveness analysis.** *Value Health.* 2022; **25**(3): 409–418.
PubMed Abstract | Publisher Full Text | Free Full Text

24. Lee D, Burns D, Wilson E: **Nice's pathways pilot: pursuing good decision making in difficult circumstances.** *Pharmacoecon Open.* 2024; **8**(5): 645–649.
PubMed Abstract | Publisher Full Text | Free Full Text

25. Csárdi G, Nepusz T, Traag V, *et al.*: **igraph: network analysis and visualization in R**. R package version 2.0.3. 2024.
Reference Source

26. RStudio Team: **RStudio: integrated development environment for R**. RStudio, PBC., Boston, MA, 2020.
Reference Source

27. Hester J: **covr: test coverage for packages**. package version 3.6.4. 2023.
Reference Source

28. Wickham H, Bryan J: **R packages**. O'Reilly Media, 2023.
Reference Source

29. Tappenden P, Chilcott JB: **Avoiding and Identifying errors and other threats to the credibility of health economic models.** *Pharmacoeconomics.* 2014; **32**(10): 967–979.
PubMed Abstract | Publisher Full Text

30. Emerson J, Bacon R, Kent A, *et al.*: **Publication of decision model source code: attitudes of health economics authors.** *Pharmacoeconomics.* 2019; **37**(11): 1409–1410.
PubMed Abstract | Publisher Full Text | Free Full Text

31. Incerti D, Thom H, Baio G, *et al.*: **R you still using excel? The advantages of modern software tools for health technology assessment.** *Value Health.* 2019; **22**(5): 575–579.
PubMed Abstract | Publisher Full Text

32. Ramos IC, Feenstra T, Ghabri S, *et al.*: **Evaluating the validation process: embracing complexity and transparency in health economic modelling.** *Pharmacoeconomics.* 2024; **42**(7): 715–719.
PubMed Abstract | Publisher Full Text | Free Full Text

33. Gautier L: **Rpy2: a simple and efficient access to R from Python**. 2008; Accessed: 2024-08-12.
Reference Source

34. Hornik K: **The comprehensive r archive network.** *WIREs Computational Statistics.* 2012; **4**(4): 394–398.
Publisher Full Text

35. Zorginstituut Nederland: **Appendix to guideline for cost-effectiveness models in R**. 2022; Accessed: 02 August 2024.
Reference Source

36. Seo MK, Strong M: **A practical guide to modeling and conducting a cost-effectiveness analysis of companion biomarker tests for targeted therapies using r: tutorial paper.** *Pharmacoeconomics.* 2021; **39**(12): 1373–1381.
PubMed Abstract | Publisher Full Text | Free Full Text

37. Candio P, Pouwels KB, Meads D, *et al.*: **Modelling decay in effectiveness for evaluation of behaviour change interventions: a tutorial for public health economists.** *Eur J Health Econ.* 2022; **23**(7): 1151–1157.
PubMed Abstract | Publisher Full Text | Free Full Text

38. Xin Y, Gray E, Robles-Zurita JA, *et al.*: **From spreadsheets to script: experiences from converting a Scottish cardiovascular disease policy model into R.** *Appl Health Econ Health Policy.* 2022; **20**(2): 149–158.
PubMed Abstract | Publisher Full Text

39. Lin YS, O'Mahony JF, van Rosmalen J: **A simple cost-effectiveness model of screening: an open-source teaching and research tool coded in R.** *Pharmacoecon Open.* 2023; **7**(4): 507–523.
PubMed Abstract | Publisher Full Text | Free Full Text

40. Filipovic-Pierucci A, Zarca K, Durand-Zaleski I: **Prm68 - markov models for health economic evaluation modelling in r with the heemod package.** *Value Health.* 2016; **19**(7): A369–A766.
Reference Source

41. Smith R: **Asserthe r package**. October, 2024.
http://www.doi.org//10.5281/zenodo.13969179

Open Peer Review

Current Peer Review Status: ?

Version 1

Reviewer Report 07 January 2025

https://doi.org/10.21956/wellcomeopenres.25526.r114411

?

Alexandra Hill
London School of Hygiene & Tropical Medicine, London, UK

The article clearly makes the case for greater adoption of the R programming language in health economic modelling in order to improve model replication and validation. The authors present a novel R package assertHE which is intended to address "Validation of the computerised model" as described in the AdViSHE validation assessment tool[1]. assertHE has 2 broad functions: firstly, it provides checks for health economic modellers to incorporate into their code to test common model features; secondly, it provides utilities intended for model reviewers to visualise the network of functions comprising a model, including AI generated human-readable documentation and code coverage metrics.

How does assertHE contribute to solving the problem it identifies:

- The suite of health economic model checks seems like a clearly useful utility for model developers. On the other hand, the code summary functions for reviewers have nothing particularly domain specific about them - visualising the network of functions in a package is totally domain agnostic, as is code coverage as a quality metric. The package's Wiki documentation on GitHub does suggest that the tabulated model summary could use a custom @cheers roxygen[6] tag, which is health economic specific, but doesn't seem to have been implemented yet.

- Overall I would suggest that the suite of HE-specific assertions and the code summarising functionality could be two completely separate packages. The network visualisation is a nice general tool for understanding the structure of a codebase in any domain and would be of interest outside of health economic modeling. One notable limitation of this functionality as it stands is that it excludes R6 classes[5]; supporting codebases that use R6 would make it more useful.

- I am not sure how much value the network-based representation of code coverage adds over and above using existing code coverage tools, e.g. devtools::test_coverage[2], which already gives a visual report showing which lines of code are covered. Again, a limitation not noted in the paper is that the network graph does not include code contained in R6 classes, whereas the covr[3]/devtools[2] code coverage report includes all lines of code in the repo.

Usability of the tool:

I was able to install the package and run the functions, with the only issue being that when running "assertHE::visualise_project" with "run_coverage = T", I got an error:

- Error in `dplyr::group_by()`:
- ! Must group by variables found in `.data`.
- □ Column `functions` is not found.

I am using R version 4.4.2 and dplyr version 1.1.4[4].

The package documentation is split across vignettes and a Wiki - the former seems more up to date and is the canonical place for package documentation to live. I would suggest retiring the Wiki and adding a vignette explaining how to use the LLM generated output. There are a lot of functions exported in the docs, many of which look like internal functions that a user of the package would not be interested in. I would suggest auditing the list of exported functions to make clearer which functions define the intended user interface/are the expected entry points into the package. The authors mention testing assertHE on several existing codebases; I would suggest adding a vignette with one of these as a case study, to make clearer how the package is to be used.

As a minor comment on the package interface, I was confused by function arguments and column headers with the word "foo" in them: "foo_name", "foo_string", "foo_location", etc. "Foo" usually indicates some kind of placeholder; I would suggest renaming all of these to make clearer what they are (e.g. "function_name", "function_location" etc.)

References

1. Vemer P, Corro Ramos I, van Voorn GA, Al MJ, et al.: AdViSHE: A Validation-Assessment Tool of Health-Economic Models for Decision Makers and Model Users.*Pharmacoeconomics*. 2016; **34** (4): 349-61 PubMed Abstract | Publisher Full Text
2. Wickham H, Hester J, Chang W, Bryan J (2022). _devtools: Tools to Make Developing R Packages Easier_. R package version 2.4.5, https://CRAN.R-project.org/package.
3. Hester J (2023). _covr: Test Coverage for Packages_. R package version 3.6.4, https://CRAN.R-project.org/package.
4. Wickham H, François R, Henry L, Müller K, Vaughan D (2023). _dplyr: A Grammar of Data Manipulation_. R package version 1.1.4, https://CRAN.R-project.org/package.
5. Chang W (2021). _R6: Encapsulated Classes with Reference Semantics_. R package version 2.5.1, https://CRAN.R-project.org/package.
6. Wickham H, Danenberg P, Csárdi G, Eugster M (2024). _roxygen2: In-Line Documentation for R_. R package version 7.3.2, https://CRAN.R-project.org/package.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: Organised events with FL, NG. Student of GB- I can confirm that my potential conflicts of interest did not prevent me from writing an objective and unbiased review.

Reviewer Expertise: software engineering, R, health economics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.