Do-It-Yourself model validation: a practical illustration of the Probabilistic Analysis Check R Package in four case studies

X.G.L.V. Pouwels & H. Koffijberg

2025-12-05

Target journal: Pharmacoeconomics

Type of paper: Practical application

# Introduction

Health economic (HE) models are routinely developed to inform health policy decisions such as including (new) healthcare interventions in insurance packages or restricting their use to specific subgroups[1]. Validating health economic models is a crucial step within the process of developing health economic models to identify and correct potential errors[2], to increase model credibility, and to reduce the risk of suboptimal policy decisions[3]. However, validation efforts on HE models are often not systematically performed nor reported[4].

Validation of HE models is a multifaceted concept. The ISPOR-SMDM Modeling Good Research Practices Task Force defined different aspects of HE model validity: face validity, verification (also called internal validity), cross validity, and external validity of the model structure, inputs, and outputs[5][6], but does not provide an easy-to-use tool to report validation efforts for each of these aspects. Following this initial definition of HE model validation, multiple generic tools have been developed to structure the (reporting of) the validation efforts performed on HE models. Well-known examples of such tools are AdviSHE tool, the TECH-VER checklist, and the CADTH’s Model Validation Tool to assist in the Conduct of Economic Evaluations[7]–[9]. Both AdviSHE and CADTH’s tool can be used to assess multiple aspects of validity, such as face validity and technical verification while TECH-VER solely focuses on the latter.

Besides the availability of these validation tools, the (increasing) use of script-based software, such as ‘R’[10], to develop HE models[11] facilitates the automated execution of systematic and generic validation tests and may be key to improve HE modelling validation practices . For instance, the hesim package[12], *an R package for health economic simulation modeling and decision analysis*, contains a function to easily check the summary statistics of the parameter distributions used for the probabilistic analysis (PA). Similarly, the darthpack package, *an R package that showcases the Decision Analysis in R for Technologies in Health (DARTH) coding framework*, contains functions to assess the validity of the transition matrices and arrays used to populate health state transition models developed according to this framework[13]. These examples show the feasibility of integrating simple validation tests during the development of script-based health economic models. Even though the nature of these validation tests is generic, their current implementation within coding frameworks for HE models may limit their use and usefulness beyond HE models developed outside these coding frameworks.

Hence, generic (R) software tools to easily and systematically validate HE models are required to improve HE model validation practices. Such tools may be particularly useful for developers who are new to script-based HE model development. Examples of such tools are the assertHE R package and the probabilistic check analysis dashboard (PACBOARD)[14][15]. assertHE depicts the network of functions defined and used within a HE model developed in R[14]. This package is useful for HE model developers and reviewers since one gets an overview of the backbone of the HE model. assertHE may further improve the communication of the workings of a developed HE models to a less technically-oriented audience. PACBOARD is a web-based dashboard which allows to systematically validate HE model input parameter and output values and allows to explore the relation between HE model input and outputs values. This dashboard is partly powered by the pacheck R package.

The pacheck R package offers a suite of functions aiming at validating HE models and exploring their workings using metamodelling techniques. Metamodels are statistical models, such as a linear regression model, fitted to the (probabilistic) inputs and outputs of a HE models[16]. These metamodels allow to rapidly estimate the output of a HE models without requiring access to the source code, but also allow to assess the direction and magnitude of the relationships between inputs and outputs.

The validation tests included in pacheck are based on a published pragmatic literature review[15] . Most of the validation tests included in pacheck were identified in the TECH-VER checklist, emphasising their relevance for HE models. In contrast to with other R package dedicated to HE model development and validation, pacheck may be applied to inputs and outputs of HE models developed with other software package than R once the probabilistic inputs and outputs are loaded within the R session of interest. This broaden the scope and usefulness of pacheck to HE models developed in all possible software environments supporting the export of input and output data.

To support the uptake of the pacheck R package for model validation, clear examples of its validation are valuable to further increase the understanding of its functionality, input and output, and limitations. This paper therefore describes the functionalities of the pacheck R package using the probabilistic inputs and outputs of four published open-source HE models[17]–[20], including example codes as tutorial for (novice) R-users.

The following section describes shortly the case studies. Next, the functionalities of pacheck, including annotated R code are showcased on one open-source model (the applications of pacheck functionalities on the other case studies are available in the Supplementary Material). We continue by reflecting on facilitators to apply pacheck in these different case studies, and conclude by discussing limitations and areas of further development of pacheck.

# Case studies

The functionalities of pacheck are illustrated using the open-source HE models included in the cdx2cea R package [[17]][21], the iviRA R package[18][22], the python-based chronic kidney disease screening cost-effectiveness analysis (CKDScreeningCEA)[19], and the R-based discrete event simulation (DES) model for Abdominal Aortic Aneurysma (AAA) screening[20].

The cdx2ceapackage has been developed to perform 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”*[17]. The CEA supported by this package is performed using a cohort-based health state transition model. It compares two strategies: No CDX2 testing and CDX2 testing. The PA dataset, which is available in the cdx2ceapackage (l\_psa object), contains the probabilistic HE model inputs and outputs of 1,000 iterations. The HE model contains 18 input parameters included in the probabilistic analysis and provides discounted quality-adjusted life years and discounted costs for both strategies. This case study is used to demonstrate that the functionalities of pacheck can be used on a HE model developed according to the DARTH coding framework.

The iviRA package is an individual-level health state transition model developed to assess the costs and health effects of different treatment sequences for rheumatoid arthritis. To demonstrate the use of pacheck we generated 1,000 probabilistic inputs and outputs simulating 100 individuals using the functions of the iviRA package. The (script used to generate these) inputs and outputs are available in the github repository of the pacheck package.

CKDScreeningCEA is a cohort-based health-state transition model, which estimates the health and economic impact of population-wide screening for chronic kidney disease (CKD)[19]. The model was developed in Python and compares screening for CKD, starting at different ages and using different frequencies, with no screening. Probabilistic inputs of this health economic model are stored in an .xlsx file that can be directly imported in R using the readxl package[23]. Alternatively, these inputs can be converted to .csv and / or .rda file before being imported in R. The probabilistic disaggregated outcomes of the model are also exported in an .xlsx file, which can be imported in R in a similar fashion as the inputs. For this illustration, we ran 1,000 probabilistic iterations of the model.

The AAA screening model is a DES model aiming at assessing the health and economic impact of screening for AAA in women[20]. All probabilistic inputs and outputs of this model are stored in a R list object. For this illustration, we ran 1,000 probabilistic iterations of the model containing 1,000 individuals.

All inputs and outputs datasets were generated using R version 4.5.1 (2025-06-13 ucrt) and are available in the pacheck package.

# Illustrations of functionalities

## Validating health economic model inputs and outputs

Functionalities of pacheck to assess the validity of the model input parameters concern the plausibility of the range in which model input parameters vary. For instance, pacheck contains simple validity assessments such as assessing whether utility values and probabilities remain between 0 and 1, and whether there are no occurrences of negative costs or other strictly positive parameters in each probabilistic iteration. pacheck also contains a function to assess and visualise whether two survival curves cross (check\_surv\_mod and plot\_surv\_mod). This assessment of plausibility is especially relevant when developing partitioned survival models, where a progression-free survival (PFS) curve should always result in lower probabilities than an overall survival curve (OS). Concerning the validation of model outputs, users are able to check whether total costs and effects are positive, whether discounted results are lower than undiscounted results, and whether the mean quality of life of both strategies is within the mean utility values used for the different health states of the model. All validation efforts are also performed for each iteration of the probabilistic analysis and the pacheck functions mentions in which iteration this occurred. pacheck can be used to visually assess the convergence of HE model outputs (Box 1)[24]. The convergence plot in Box 1 may suggest that more iterations may be required to obtain stable incremental results. Box 1 further illustrates how to perform these validation efforts using pacheck.

*Box 1: example R code of using pacheck to validate the health economic model inputs and outputs*

|  |
| --- |
| # Install and load packages # install.packages("devtools") # devtools::install\_github("feralaes/cdx2cea") # devtools::install\_github("DARTH-git/dampack") # devtools::install\_github("Xa4P/pacheck") # install.packages("foreach") require(cdx2cea) require(dampack)  ## Loading required package: dampack  ## Loading required package: ggplot2  require(pacheck)  ## Loading required package: pacheck  # Load & prepare dataframes for validation tests data("l\_psa", package = "cdx2cea")  # Inspect parameter values - limited to first 5 for the sake of brevity  generate\_sum\_stats(l\_psa$parameters[, 1:5])  ## Parameter Mean SD Percentile\_2.5th Percentile\_97.5th Minimum ## 1 r\_DieMets 0.047 0.005 0.038 0.058 0.031 ## 2 r\_RecurCDX2pos 0.004 0.001 0.003 0.006 0.002 ## 3 hr\_RecurCDX2neg 3.090 0.595 2.075 4.328 1.703 ## 4 p\_Mets 0.961 0.025 0.899 0.990 0.812 ## 5 p\_CDX2neg 0.072 0.011 0.052 0.095 0.041 ## Maximum Median Skewness Kurtosis ## 1 0.065 0.046 0.400 3.316 ## 2 0.007 0.004 0.159 3.104 ## 3 6.247 3.047 0.616 3.992 ## 4 0.995 0.968 -2.343 11.511 ## 5 0.106 0.071 0.300 3.176  # Test whether utility values are within the 0-1  check\_binary(c("u\_Stg2", "u\_Stg2Chemo", "u\_Mets"),  df = l\_psa$parameters)  ## Input Negative\_values Values\_above\_1 ## 1 u\_Stg2 None None ## 2 u\_Stg2Chemo None None ## 3 u\_Mets None None  # Introduce utility values below 0 and above 1 df\_inputs\_error <- l\_psa$parameters df\_inputs\_error[c(4, 444, 754), "u\_Stg2"] <- -1 df\_inputs\_error[c(3, 333, 681), "u\_Stg2Chemo"] <- 99 df\_inputs\_error[c(5, 554, 153), "u\_Mets"] <- -1 df\_inputs\_error[c(6, 146, 538), "u\_Mets"] <- 99  # Check for erroneous utility values with pacheck function check\_binary(c("u\_Stg2", "u\_Stg2Chemo", "u\_Mets"),  df = df\_inputs\_error)  ## Input Negative\_values Values\_above\_1 ## 1 u\_Stg2 4,444,754 None ## 2 u\_Stg2Chemo None 3,333,681 ## 3 u\_Mets 5,153,554 6,146,538  # Perform multiple plausibility checks on a 'DARTH' psa object data.frame(check\_psa\_darth(l\_psa))  ## Parameter Iterations\_error ## 1 u\_Stg2 none ## 2 u\_Stg2Chemo none ## 3 u\_Mets none ## 4 p\_Mets none ## 5 p\_CDX2neg none ## 6 c\_Chemo none ## 7 c\_ChemoAdmin none ## 8 c\_CRCStg2\_init none ## 9 c\_CRCStg2\_cont none ## 10 c\_CRCStg4\_cont none ## 11 c\_Test none ## 12 hr\_RecurCDX2neg none ## 13 hr\_Recurr\_CDXneg\_Rx none ## 14 hr\_Recurr\_CDXpos\_Rx none ## 15 effectiveness\_No CDX2 testing and no FOLFOX none ## 16 effectiveness\_CDX2 testing and FOLFOX if CDX2-negative none ## 17 cost\_No CDX2 testing and no FOLFOX none ## 18 cost\_CDX2 testing and FOLFOX if CDX2-negative none  # Check convergence of a model output ## Plot moving average incremental effectiveness - per 100 iterations l\_psa$effectiveness$Incremental\_QALYs <- l\_psa$effectiveness$`CDX2 testing and FOLFOX if CDX2-negative` - l\_psa$effectiveness$`No CDX2 testing and no FOLFOX` plot\_convergence(l\_psa$effectiveness,   param = "Incremental\_QALYs",   block\_size = 100,  y\_min = 0.03,  y\_max = 0.04)    # minimum and maximum value set to prevent 'excessive' peaks |

## Investigating the relationships between inputs and outputs

The pacheck package also contains various functions to investigate the relation between HE model inputs and outputs. For instance, the correlation matrix or plot between inputs and outputs can be calculated using the generate\_cor function. Using the probabilistic inputs and outputs of the cdx2cea, one can see that the total costs of the intervention is negatively correlated with cancer mortality rate, r\_DieMets,. This result seems logical since a higher probability of death would lead to shorter survival and thus lower costs. Linear regression metamodelling is also available through pacheck (using the lm function for the linear regression modelling). When estimating the incremental difference in quality-adjusted life years (QALYs) between the intervention (CDX2 testing + adjuvant chemotherapy) and the comparator, we can see that an increase in utility values of stage II cancer with and without chemotherapy, u\_Stg2Chemo and u\_Stg2, leads to higher incremental QALYs while an increase in the utility value of the metastatic recurrent state, u\_Mets, leads to lower incremental QALYs. These results are explainable when considering the model structure of this HE evaluation. In the intervention strategy, individuals spend more time in the stage II cancer with and without chemotherapy health state. Increasing the utility value of either u\_Stg2Chemo (applied only during chemotherapy administration) or u\_Stg2 (applied after completion of chemotherapy treatment) therefore increases the number of QALYs accumulated in this health state in the intervention strategy versus the comparator strategy. The opposite applies when increasing the u\_Mets utility value. Individuals spend less time in the metastatic health state in the intervention strategy compared to the the control strategy. Therefore, increasing this utility value decreases the incremental QALYs associated with the intervention versus the comparator strategy. A striking result of this metamodel, is its high (adjusted) R2 of 0.97. This can be explained by the likely linear relationships between the inputs and outputs of this simple HE model.

Metamodel validation can be performed in pacheck by using user-defined train-test split proportions or cross-validation using a user-defined number of folds. When using the train-test split method, the user needs to supply the partition of the data that will be used to train the model. In the example in Box 2, 75% of the data is used to train the model, and the remaining 25% are used to test the trained metamodel. During cross-validation, the complete dataset is divided in a number of folds, for instance 10. The metamodel is then trained on 9 of the 10 folds and validated in the remaining fold. This is repeated for each fold. The validation metrics of the metamodel are obtained by averaging the validation metrics of the metamodel in each fold (**See Appendix for an illustration of the implementation in pacheck**).

The high R2 value (0.97) and the position of the prediction versus observation dots in the calibration dots near the 45 degree line in the test set show that the metamodel may be deemed valid to estimate the incremental QALYs. When deemed valid, such metamodel may be used to perform sensitivity analyses, which is especially useful for computationally intensive HE models. In Box 2, we show that increasing the utility value for the metastatic state, u\_Mets, to 0.5 (mean value in the probabilistic set was 0.25) is precited to increase the incremental QALYs by 0.001, from 0.035 in the base-case to 0.036.

As mentioned by Jalal et al., metamodel’s parameters may be subject to scale effects. Jalal et al. therefore suggest to normalise HE model inputs before metamodelling, which facilitates their interpretation in relation to each other[25]. Normalisation of inputs has been implemented in pacheck within the lm\_metamod function. Box 2 illustrates how to use the functions of pacheck to investigate the relationships between HE model inputs and outputs.

*Box 2: Example R code of how to use the pacheck package to assess the relationships between inputs and outputs.*

|  |
| --- |
| # Generate correlation matrix between inputs and outputs ## Transform l\_psa parameters and outcomes in one dataframe df\_psa\_cdx2cea <- cbind(l\_psa$parameters,   l\_psa$effectiveness,   l\_psa$cost) ## Calcualte correlations between first 5 inputs & total costs of the intervention tbl\_cor <- generate\_cor(df\_psa\_cdx2cea[, c(1:5, ncol(df\_psa\_cdx2cea))])  ## Show correlations between the 5 inputs & total costs of the intervention tbl\_cor[1:5, "CDX2 testing and FOLFOX if CDX2-negative"]  ## r\_DieMets r\_RecurCDX2pos hr\_RecurCDX2neg p\_Mets p\_CDX2neg  ## -0.508554559 0.750189815 -0.312984661 -0.001900022 0.049150333  ## Display correlation matrix using 'tile' plot from ggplot2 generate\_cor(df\_psa\_cdx2cea,  figure = T)    # Fit linear metamodel to predict the incremental QALYs between strategies ## Check relation between costs input parameters and incremental QALYs v\_c\_vars <- l\_psa$parnames[grep("^c\_", l\_psa$parnames)] tbl\_cor\_iQALYs <- generate\_cor(df\_psa\_cdx2cea[, c(v\_c\_vars, "Incremental\_QALYs")],  figure = F) tbl\_cor\_iQALYs[v\_c\_vars, "Incremental\_QALYs"]  ## c\_Chemo c\_ChemoAdmin c\_CRCStg2\_init c\_CRCStg2\_cont c\_CRCStg4\_cont  ## 0.014601758 -0.021483743 -0.005944846 -0.062562626 0.025882823  ## c\_Test  ## -0.051639947  ## No apparent correlation between costs input parameters and incremental QALYs  ## Decision not to include costs input parameter in the linear metamodel v\_x\_vars <- l\_psa$parnames[grep("^c\_", l\_psa$parnames, invert = T)] y\_var <- names(l\_psa$effectiveness)[3] lm\_metamod <- fit\_lm\_metamodel(y\_var = y\_var,  x\_vars = v\_x\_vars,   df = df\_psa\_cdx2cea) # the names of the input parameters can be used instead of "v\_x\_vars"   ## Overview of estimated linear metamodel parameters summary(lm\_metamod$fit)  ##  ## Call: ## lm(formula = form, data = df) ##  ## Residuals: ## Min 1Q Median 3Q Max  ## -0.0097410 -0.0017664 -0.0007761 0.0009259 0.0250413  ##  ## Coefficients: ## Estimate Std. Error t value Pr(>|t|) ## (Intercept) 0.1684922598 0.0828378588 2.034 0.0422 ## r\_DieMets 0.0088003017 0.0229356484 0.384 0.7013 ## r\_RecurCDX2pos -1.9742925140 0.1951303413 -10.118 < 0.0000000000000002 ## hr\_RecurCDX2neg -0.0031447830 0.0002095364 -15.008 < 0.0000000000000002 ## p\_Mets -0.0030930124 0.0044799734 -0.690 0.4901 ## p\_CDX2neg 0.4919905564 0.0103636969 47.472 < 0.0000000000000002 ## hr\_Recurr\_CDXneg\_Rx -0.2310806122 0.0012971089 -178.151 < 0.0000000000000002 ## hr\_Recurr\_CDXpos\_Rx -0.0001457265 0.0085688076 -0.017 0.9864 ## ic\_DeathCRCStg2 -0.0000001826 0.0000014571 -0.125 0.9003 ## ic\_DeathOCStg2 0.0000016267 0.0000012873 1.264 0.2067 ## u\_Stg2 0.0204151331 0.0046984042 4.345 0.0000153600652 ## u\_Stg2Chemo 0.0284825885 0.0041731900 6.825 0.0000000000153 ## u\_Mets -0.0039898544 0.0039401775 -1.013 0.3115 ##  ## (Intercept) \*  ## r\_DieMets  ## r\_RecurCDX2pos \*\*\* ## hr\_RecurCDX2neg \*\*\* ## p\_Mets  ## p\_CDX2neg \*\*\* ## hr\_Recurr\_CDXneg\_Rx \*\*\* ## hr\_Recurr\_CDXpos\_Rx  ## ic\_DeathCRCStg2  ## ic\_DeathOCStg2  ## u\_Stg2 \*\*\* ## u\_Stg2Chemo \*\*\* ## u\_Mets  ## --- ## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1 ##  ## Residual standard error: 0.003448 on 987 degrees of freedom ## Multiple R-squared: 0.9724, Adjusted R-squared: 0.972  ## F-statistic: 2895 on 12 and 987 DF, p-value: < 0.00000000000000022  # Normalise the inputs lm\_metamod\_standardised <- fit\_lm\_metamodel(y\_var = y\_var,   x\_vars = v\_x\_vars,  df = df\_psa\_cdx2cea,  seed\_num = 123,  standardise = TRUE) summary(lm\_metamod\_standardised$fit) # provides an overview of estimated linear metamodel parameters  ##  ## Call: ## lm(formula = form, data = df) ##  ## Residuals: ## Min 1Q Median 3Q Max  ## -0.0097410 -0.0017664 -0.0007761 0.0009259 0.0250413  ##  ## Coefficients: ## Estimate Std. Error t value Pr(>|t|)  ## (Intercept) 0.03580038 0.00010903 328.356 < 0.0000000000000002 \*\*\* ## r\_DieMets 0.00004445 0.00011586 0.384 0.701  ## r\_RecurCDX2pos -0.00135079 0.00013351 -10.118 < 0.0000000000000002 \*\*\* ## hr\_RecurCDX2neg -0.00187099 0.00012466 -15.008 < 0.0000000000000002 \*\*\* ## p\_Mets -0.00007734 0.00011202 -0.690 0.490  ## p\_CDX2neg 0.00519457 0.00010942 47.472 < 0.0000000000000002 \*\*\* ## hr\_Recurr\_CDXneg\_Rx -0.01956199 0.00010981 -178.151 < 0.0000000000000002 \*\*\* ## hr\_Recurr\_CDXpos\_Rx -0.00000187 0.00010995 -0.017 0.986  ## ic\_DeathCRCStg2 -0.00001377 0.00010987 -0.125 0.900  ## ic\_DeathOCStg2 0.00013806 0.00010926 1.264 0.207  ## u\_Stg2 0.00047474 0.00010926 4.345 0.0000153600652 \*\*\* ## u\_Stg2Chemo 0.00074829 0.00010964 6.825 0.0000000000153 \*\*\* ## u\_Mets -0.00011111 0.00010973 -1.013 0.311  ## --- ## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1 ##  ## Residual standard error: 0.003448 on 987 degrees of freedom ## Multiple R-squared: 0.9724, Adjusted R-squared: 0.972  ## F-statistic: 2895 on 12 and 987 DF, p-value: < 0.00000000000000022  # Validation metamodel using the train-test approach lm\_metamod\_valid <- fit\_lm\_metamodel(y\_var = y\_var,  x\_vars = v\_x\_vars,  df = df\_psa\_cdx2cea,  seed\_num = 123,  validation = "train\_test\_split",  partition = 0.75)  ## in combination with the "train\_test\_split" approach,  ## the proportion of observation used to train the metamodel (`partition`)  ## should be provided. The remainder is used as validation check  lm\_metamod\_valid$stats\_validation  ## Statistic Value (method: train/test split) ## 1 R-squared 0.971 ## 2 Mean absolute error 0.002 ## 3 Mean relative error -0.102 ## 4 Mean squared error 0.000  lm\_metamod\_valid$calibration\_plot    # Prediction metamodel ## Select input parameters df\_params <- data.frame(df\_psa\_cdx2cea[, v\_x\_vars]) ## Increase the utility value of the metastatic state df\_params$u\_Mets <- 0.5  v\_pred <- predict\_metamodel(model = lm\_metamod\_valid,  inputs = df\_params)  ## 'inputs' has to be a dataframe c(average = round(mean(v\_pred), 3),  `percentile 2.5%` = unname(round(quantile(v\_pred, probs = 0.025), 3)),  `percentile 97.5%` = unname(round(quantile(v\_pred, probs = 0.975), 3)))  ## average percentile 2.5% percentile 97.5%  ## 0.036 -0.008 0.074  # averag of 0.036 QALYs, base-case incremental QALYs was 0.035 |

Additional functionalities of pacheck are provided in the Supplemental Material, using the other three case studies **(REF to Supplemental Material)**.

# Facilitators to applying pacheck to different open-source models

Due to their open-source character, all selected HE models provided access to a full set of probabilistic inputs and outputs or we could easily generate such a set of probabilistic inputs and outputs. This allowed to quickly implement the functionalities of pacheck after running the model (in three of the four case studies) and limited manipulation of the format of the inputs and outputs. Only the cdx2cea provided a saved set of probabilistic inputs and outputs. Ensuring all inputs and outputs are saved, and ideally ordered as a table, facilitates the application of pacheck.

Each model had a clear README file, which facilitated navigation within the different file structure of the models. In addition, most of the inputs and outputs were described using explicit names. This was helpful to understand the potential natural bounds of these inputs and outputs. For instance, words and abbreviations such as ‘rate’, ‘prob’ (probabilities), ‘HR’ (hazard ratio), and ‘cost’ were used in multiple models. However, a clear definition of each input and output was most often missing. To ensure users have direct access to the meaning of the different variables represented by the input and output parameters of the model, we suggest that modellers add descriptive metadata to the probabilistic inputs and outputs files. Also, navigation within health economic models could be improved (especially for novice script-based software users) if all modellers would adhere to the same standard file structure and a coding style, such as the one proposed by the DARTH working group[13].

# Discussion

The current paper presents functionalities of the pacheck R package and illustrates the practical use of these functionalities, with detailed R code, in four case studies.  
While we demonstrated that pacheck is easy to use and contains useful validation tests, the package clearly does not provide a complete list of validation tests for HE models. Pacheck is a package that requires regular updates and addition of validation tests over time to ensure it remains relevant for HE model developers and reviewers.  
From a technical point of view, pacheck is most likely not coded in the most efficient way. This is a design choice we made to ensure transparency of the implemented validation tests and to encourage external contributors to review the code base and to contribute to the further development of the package. The source code of pacheck is openly available on GitHub: [https://github.com/Xa4P/pacheck](#X6589fc6ab0dc82cf12099d1c2d40ab994e8410c). External contributors can raise “Issues” concerning the package and propose new validation tests via “Issues” and “Pull requests”. In future releases of pacheck, we aim to implement a larger number of validation tests and types of metamodel. We are also planning to test pacheck on the probabilistic inputs and outputs of “closed-source” HE models to demonstrate the usefulness of pacheck in such setting.  
Finally, pacheck focuses on the technical verification of HE model which is only a single aspect of validity. Hence, passing the validation tests included in pacheck should not create a feeling of false certainty concerning the validity of HE models. HE model developers and reviewers are therefore encouraged to use pacheck as a complement to other validation tools such as AdviSHE and CADTH’s tool to report and review a broad range of validation aspects[9] .

# References

1. Drummond MF, Sculpher MJ, Claxton K. Methods for the economic evaluation of health care programmes [online]. Oxford University Press; 2015. Available from URL: <https://books.google.co.uk/books?id=lvWACgAAQBAJ>

2. 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](https://doi.org/10.1017/S0266462320000422). International Journal of Technology Assessment in Health Care 2020 Jul; : 1–6

3. Caro JJ, Eddy DM, Kan H, et al. Questionnaire to assess relevance and credibility of modeling studies for informing health care decision making: An ISPOR-AMCP-NPC good practice task force report. Value in Health [online] 2014 Mar; 17 (2): 174–182. Available from URL: <https://www.valueinhealthjournal.com/article/S1098-3015(14)00011-4/fulltext?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS1098301514000114%3Fshowall%3Dtrue>

4. Abraham K, Corro Ramos I, Braal CL, et al. Under-reporting of Validation Efforts for Health Economic Models Persists Despite the Availability of Validation Tools: A Systematic Review. PharmacoEconomics [online] 2025 Aug; 43 (8): 849–858. Available from URL: <https://doi.org/10.1007/s40273-025-01491-2>

5. 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. Value in Health [online] 2012 Sep; 15 (6): 843–850. Available from URL: <https://www.valueinhealthjournal.com/article/S1098-3015(12)01656-7/fulltext?_returnURL=https%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS1098301512016567%3Fshowall%3Dtrue>

6. Corro Ramos I, Feenstra T, Ghabri S, et al. Evaluating the validation process: Embracing complexity and transparency in health economic modelling. Pharmacoeconomics [online] 2024; 42 (7): 715–719. Available from URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC11180005/>

7. Vemer P, Corro Ramos I, Voorn GAK van, et al. AdViSHE: A Validation-Assessment Tool of Health-Economic Models for Decision Makers and Model Users. PharmacoEconomics [online] 2016 Apr; 34 (4): 349–361. Available from URL: <https://doi.org/10.1007/s40273-015-0327-2>

8. 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 [online] 2019 Nov; 37 (11): 1391–1408. Available from URL: <https://doi.org/10.1007/s40273-019-00844-y>

9. Coyle D, Haines A, Lee K. The development of a model validation tool to assist in the conduct of economic evaluations. Canadian Journal of Health Technologies [online] 2024 Mar; 4 (3). Available from URL: <http://canjhealthtechnol.ca/index.php/cjht/article/view/MG0026>

10. R Core Team. R: A language and environment for statistical computing [online]. Vienna, Austria: R Foundation for Statistical Computing; 2025. Available from URL: <https://www.R-project.org/>

11. Jalal H, Pechlivanoglou P, Krijkamp E, et al. An Overview of R in Health Decision Sciences. Medical Decision Making [online] 2017 Oct; 37 (7): 735–746. Available from URL: <https://doi.org/10.1177/0272989X16686559>

12. Incerti D, Jansen JP. Hesim: Health economic simulation modeling and decision analysis. 2021; Available from URL: <https://arxiv.org/abs/2102.09437>

13. Alarid-Escudero F, Krijkamp EM, Pechlivanoglou P, et al. A Need for Change! A Coding Framework for Improving Transparency in Decision Modeling. PharmacoEconomics [online] 2019 Nov; 37 (11): 1329–1339. Available from URL: <https://doi.org/10.1007/s40273-019-00837-x>

14. Smith RA, Samyshkin Y, Mohammed W, et al. assertHE: an R package to improve quality assurance of HTA models. Wellcome Open Research [online] 2024 Dec; 9: 701. Available from URL: <https://wellcomeopenresearch.org/articles/9-701/v1>

15. Pouwels XGLV, Kroeze K, Linden N van der, et al. Validating health economic models with the probabilistic analysis check dashBOARD. Value in Health [online] 2024 Aug; 27 (8): 1073–1084. Available from URL: <https://doi.org/10.1016/j.jval.2024.04.008>

16. Degeling K, IJzermann MJ, Koffijberg H. A scoping review of metamodeling applications and opportunities for advanced health economic analyses - PubMed [online]. 2018 Nov; Available from URL: <https://pubmed.ncbi.nlm.nih.gov/30426801/>

17. Alarid-Escudero F, Schrag D, Kuntz KM. CDX2 biomarker testing and adjuvant therapy for stage II colon cancer: An exploratory cost-effectiveness analysis. Value in Health [online] 2022 Mar; 25 (3): 409–418. Available from URL: <https://doi.org/10.1016/j.jval.2021.07.019>

18. Incerti D, Curtis JR, Shafrin J, et al. A Flexible Open-Source Decision Model for Value Assessment of Biologic Treatment for Rheumatoid Arthritis. PharmacoEconomics [online] 2019 Jun; 37 (6): 829–843. Available from URL: <https://doi.org/10.1007/s40273-018-00765-2>

19. Cusick MM, Tisdale RL, Chertow GM, et al. Population-wide screening for chronic kidney disease: A cost-effectiveness analysis. Annals of Internal Medicine [online] 2023 Jun; 176 (6): 788–797. Available from URL: <https://www.acpjournals.org/doi/10.7326/M22-3228>

20. Sweeting MJ, Masconi KL, Jones E, et al. Analysis of clinical benefit, harms, and cost-effectiveness of screening women for abdominal aortic aneurysm. The Lancet [online] 2018 Aug; 392 (10146): 487–495. Available from URL: <https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(18)31222-4/fulltext>

21. Alarid-Escudero F. cdx2cea is an r package that implements the cost-effectiveness analysis (CEA) of testing average-risk stage II colon cancer patients for the absence of CDX2 biomarker expression followed by adjuvant chemotherapy. [online]. Zenodo; 2021. Available from URL: <https://zenodo.org/records/5093594>

22. Incerti D, Curtis JR, Shafrin J, et al. [A flexible open-source decision model for value assessment of biologic treatment for rheumatoid arthritis](https://doi.org/10.1007/s40273-018-00765-2). 2019;

23. Wickham H, Bryan J. Readxl: Read excel files [online]. 2025. Available from URL: <https://readxl.tidyverse.org>

24. Hatswell AJ, Bullement A, Briggs A, et al. Probabilistic Sensitivity Analysis in Cost-Effectiveness Models: Determining Model Convergence in Cohort Models. PharmacoEconomics [online] 2018 Dec; 36 (12): 1421–1426. Available from URL: <http://link.springer.com/10.1007/s40273-018-0697-3>

25. Jalal H, Dowd B, Sainfort F, et al. Linear Regression Metamodeling as a Tool to Summarize and Present Simulation Model Results. Medical Decision Making [online] 2013 Oct; 33 (7): 880–890. Available from URL: <https://doi.org/10.1177/0272989X13492014>