## Tables

**Table** **1:** Considerations when choosing between implementing computational models using commercial modelling software or as software projects using an open-source programming language

|  |  |  |
| --- | --- | --- |
|  | **Commercial modelling software** | **Open source programming language** |
| **Project costs** | Default option for projects with tight time and resource constraints.  Requires less extensive range of (potentially hard to find) specialized skills. | Initial development is likely to be time and resource intensive, though over medium to long term time horizons there is the potential for efficiency savings if:   1. model development can leverage artefacts (code and data) from pre-existing models; and / or 2. project requirements include model maintenance and transfer. |
| **Project quality** | Robustness of commercial software means users can have high degrees of confidence that model files will open and execute correctly over medium to long-term.  Lack of integration with tools used for scientific manuscript authoring may result in transcription errors when reporting results. | Facilitates more extensive / complex model representations and use in interactive decision aids.  Supports model transparency, reusability and updatability.  Lots of novel code is a potential source of errors in model implementations.  Potentially fragile – if not maintained or bundled with all required and correctly versioned dependencies, models may not execute as intended in the future. |

**Table 2:** How transparent, reusable and updatable (TRU) CHEMs promote ethical modelling practice.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Social acceptability** | **Fitness for purpose** | **Beneficial use** |
| **Transparent CHEMs** | Value judgments and assumptions can be reviewed by third parties. | | Clarity about model features and performance reduces risk of inappropriate use in decision-making. |
| **Reusable CHEMs** | Use and modification of models by third-parties allows alternative value judgments to be explored. | Use by third parties increases likelihood of uncovering errors. | More potential model beneficiaries and less duplicative modeller effort. |
| **Updatable CHEMs** | Well maintained models can be validly used for longer. | |

**Table 3** **:** Software framework R libraries

|  |  |  |
| --- | --- | --- |
| **Library** | **Purpose** | **Dependency R libraries** |
|  | Provide a template and novel syntax for modular CHEM implementations and tools for finding interoperable CHEM modules, datasets and reproducible analysis programs. | assertthat bib2df dataverse dplyr fs Hmisc kableExtra knitr lifecycle magrittr methods natmanager piggyback purrr readr readxl rlang rmarkdown rvest stats stringi stringr testit testthat tibble tidyRSS tools utils zen4R |
|  | Streamline and standardise the authoring and documenting of functions that support transferable and generalisable model algorithms. | desc devtools dplyr generics gert Hmisc knitr lifecycle lubridate magrittr methods piggyback pkgdown purrr readxl ready4 ready4show ready4use rlang sinew stats stringi stringr testit testthat tibble tidyr tools usethis utils xfun |
|  | Streamline and standardise the authoring and documenting of new interoperable CHEM modules. | devtools dplyr fs gtools Hmisc knitr lifecycle magrittr methods purrr ready4 ready4fun ready4show rlang stats stringi stringr testit testthat tibble tidyr usethis utils |
|  | Help bundle and disseminate newly created CHEM modules as R libraries that are documented, licensed and quality assured. | dataverse dplyr knitr lifecycle magrittr methods purrr ready4 ready4class ready4fun rlang stringr testthat tibble tidyr utils |
|  | Help manage the labelling and transfer of data between CHEM modules and local and remote data repositories. | data.table dataverse dplyr fs Hmisc knitr lifecycle magrittr methods piggyback purrr readxl ready4 ready4show rlang stats stringi stringr testit testthat tibble tidyr utils |
|  | Facilitate the use of CHEM modules in programs that make the entire data ingest, analysis and reporting pipeline reproducible can be challenging and time consuming. | dataverse DescTools dplyr flextable grDevices here Hmisc kableExtra knitr knitrBootstrap lifecycle magrittr methods officer purrr ready4 rlang rmarkdown stringi stringr testthat tibble tidyr utils xtable |

**Table 4:** Assessment of outcome valuation CHEM implementation against transparent, reusable and updatable (TRU) criteria.

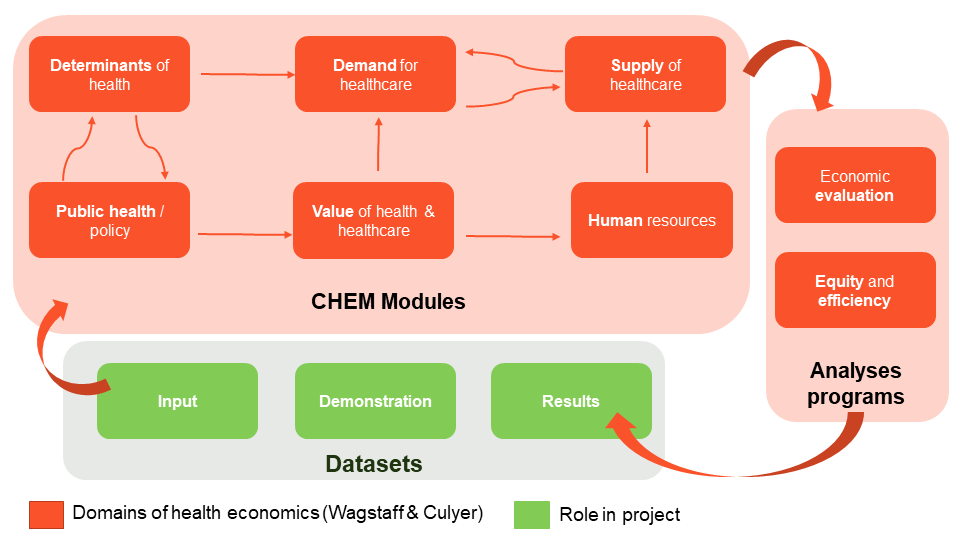
|  |  |  |
| --- | --- | --- |
| **Criteria** | **Met?** | **Detail** |
| **T1 Open access** | Yes | All source code and testing procedures are available in public GitHub repositories, with each code release persistently available in a Zenodo repository. Study dataset contains confidential patient records and was not published. Instead, a synthetic representation of the study dataset is persistently available in a repository in the Harvard Dataverse. Data files to support out of sample application of models are published at the same location. |
| **T2 Author contributions & beliefs** | Yes | All module libraries, programs, and datasets are distributed with citation information. GitHub repositories detail author code contributions over project development history are publicly visible. Model catalogues persistently available on the Harvard Dataverse describe the predictive performance of models under multiple usage regimes. Each code library is documented with worked examples of how to apply modules. Analysis and reporting programs are self-documenting. Sub-routines for generating reports are documented with README files. |
| **R1 Generalisability & transferability** | Yes | Model code is written using both functional and object-oriented paradigms. Code library websites include hypothetical examples of generalisability (applying study algorithm to estimate mapping models from new data with the same predictor and outcome variables) and transferability (adapting study algorithm to develop mapping models from datasets with different predictor variables and outcomes measured with a different utility instrument). |
| **R2 Open-source licenses** | Yes | All code is distributed using GPL-3 licenses. Datasets use amended version of template provided by Harvard Dataverse, allowing reuse of data subject to some ethical restrictions (e.g., use in efforts to reidentify study participants is prohibited). |
| **U1 Version-control and maintenance** | Yes | All code is version controlled using git and GitHub, with semantic versioning. Each code library has a specified maintainer and guidance for potential code contributors is available on the project website. |
| **U2 Retesting & Deprecation** | Partial | Continuous integration used for all code libraries, primarily for acceptance testing. Only limited use is made of unit testing. Retired library code is deprecated using tools from the lifecycle R library. Library documentation articles and datasets are also deprecated. |

**[Logo, company name

Description automatically generated](https://ready4-dev.github.io/ready4show/)Figures**

6

**Figure 1: High level summary of planned implementation of youth mental health economic model**

****