Complete study program to reproduce all steps from data ingest through to results dissemination for a study to map mental health measures to AQoL-6D health utility

Matthew P Hamilton[†,1,*] Caroline X Gao[†,1,2,3]

13 September 2022

[†] These authors contributed equally to this work.

[1] Orygen, Parkville, Australia
[2] Centre for Youth Mental Health; University of Melbourne, Parkville, Australia
[3] School of Public Health and Preventive Medicine, Monash University, Clayton, Australia

[*] Correspondence: Matthew P Hamilton <matthew.hamilton@orygen.org.au>

1 About this code

This R code is the complete study program for all steps from data ingest, analysis and reporting for the study summarised here: https://doi.org/10.1101/2021.07.07.21260129.

When using this code it is important to note:

1) Some of the steps in this program involve interactivity - they generate a prompt that a user must respond to before proceeding. Therefore, this code should be run step by step (i.e run one chunk at a time and do not knit if you are using the kdown version of this code).

2) The code is implemented in R using ready4 framework modules for

- labeling, validating and summarising youth mental health datasets from the youthvars package;
- scoring health utility from the scorz package;
- specifying and testing statistical models from the specific package;
- generating reports from the ready4show package; and
- sharing data via online data repositories from the ready4use package; and
- implementing a utility mapping study from the TTU package.

3) The code in this program is highly abstracted which means that its detailed workings are not exposed. However, as all code is open source you are able to scrutinise the underlying code if you wish. The websites, manuals and repositories of the ready4 framework libraries used in this analysis are a good starting point.

2 Install and load required libraries

If you do not already have the required libraries to run this program installed, you can do so by un-commenting and running the following lines.

```r
# devtools::install_github("ready4-dev/ready4)
# devtools::install_github("ready4-dev/TTU")
```

Next load the libraries required to run this program.

```r
library(ready4)
library(ready4use)
library(youthvars)
library(scorz)
library(TTU)
```

3 Ingest, transform, label and score data

3.1 Ingest data

To start, we need to specify the path to our input dataset. The setting below (empty) will allow you to run the program for an illustrative **replication** study using a synthetic dataset that, though entirely fake, closely resembles the study dataset. However, if you have access to the source dataset and wish to use it to **reproduce** our study, you can instead specify the path to that data here.

```
path_to_data_1L_chr <- NA_character_ # Replace with path to data if you have access to it
```

Based on the value specified above, we can now ingest either the real study dataset or the synthetic (fake) data. We perform a number of data transformations (using the `youthvars::transform_raw_ds_for_analysis` function) to prepare the dataset for analysis and then pair the dataset with its data dictionary as a `Ready4useDyad`.

```
A <- Ready4useDyad(ds_tb = {if(!is.na(path_to_data_1L_chr)){readRDS(path_to_data_1L_chr)
  }else{Ready4useRepos(dv_nm_1L_chr = "fakes", dv_ds_nm_1L_chr = "https://doi.org/10.7910/DVN/HJXYKQ",
                       dv_server_1L_chr = "dataverse.harvard.edu") %>%
      ingest(fls_to_ingest_chr = c("ymh_clinical_tb"), metadata_1L_lgl = F)}}%>% youthvars::transform_raw_ds_for_analysis(),
  dictionary_r3 = Ready4useRepos(dv_nm_1L_chr = "TTU", dv_ds_nm_1L_chr = "https://doi.org/10.7910/DVN/DKDIB0",
                                 dv_server_1L_chr = "dataverse.harvard.edu") %>%
    ingest(fls_to_ingest_chr = c("dictionary_r3"), metadata_1L_lgl = F)) %>%  renew(type_1L_chr = "label")
```

We add some metadata about the structural properties of our dataset (variables for the unique record identifier and data collection round), using the `YouthvarsSeries` module.

```
A <- YouthvarsSeries(a_Ready4useDyad = A,
                     id_var_nm_1L_chr = "fkClientID",
                     timepoint_var_nm_1L_chr = "round",
                     timepoint_vals_chr = levels(procureSlot(A,"ds_tb")$round))
```

3.2 Score health utility

We calculate adolescent AQoL-6D scores using the `ScorzAqol6Adol` module and save the dataset along with its metadata as a `TTUProject` module.

```
A <- TTUProject(a_ScorzProfile = ScorzAqol6Adol(a_YouthvarsProfile = A))
A <- renewSlot(A, "a_ScorzProfile")
```

4 Describe and analyse data

4.1 Specify modelling parameters

We begin by reorganising some of the metadata we have already specified about our dataset.

```
A <- renewSlot(A, "b_SpecificParameters", SpecificConverter(a_ScorzProfile = A@a_ScorzProfile) %>%
                 metamorphose() %>%
                 procureSlot("b_SpecificParameters"))
```

We add a table describing the candidate predictors that we will explore.

```
A <- renewSlot(A, "b_SpecificParameters@predictors_lup", Ready4useRepos(dv_nm_1L_chr = "TTU",
                                                                        dv_ds_nm_1L_chr = "https://doi.org/10.7910/DVN/DKDIB0",
                                                                        dv_server_1L_chr = "dataverse.harvard.edu") %>%
                 ingest(fls_to_ingest_chr = c("predictors_r3"),
                        metadata_1L_lgl = F))
```

We add the required parameters (allowable range for utility scores, the variable names of candidate predictors to be explored in the primary analysis, candidate covariates, variables to use when preparing descriptive statistics, the data collection date-stamp variable and metadata on candidate predictors).

```
A <- renewSlot(A, "b_SpecificParameters@depnt_var_min_max_dbl", c(0.03,1)) %>%
  renewSlot("b_SpecificParameters@candidate_predrs_chr", c("BADS","GAD7", "K6", "OASIS", "PHQ9", "SCARED")) %>%
  renewSlot("b_SpecificParameters@candidate_covars_chr", c("d_sex_birth_s", "d_age",  "d_sexual_ori_s",
                                                           "d_studying_working", "c_p_diag_s", "c_clinical_staging_s",
                                                           "SOFAS")) %>%
  renewSlot("b_SpecificParameters@descv_var_nms_chr", c("d_age","Gender","d_relation_s", "d_sexual_ori_s",
                                                        "Region", "d_studying_working", "c_p_diag_s",
                                                        "c_clinical_staging_s","SOFAS")) %>%
  renewSlot("b_SpecificParameters@msrmnt_date_var_nm_1L_chr", "d_interview_date")
```

4.2 Create local workspace

We add details of the directory where data generated by our analysis will be written. Note, if using real data **this must be a secure location** as copies of the dataset will be saved in this directory.

```
A <-  renewSlot(A, "b_SpecificParameters@fake_1L_lgl", ifelse(is.na(path_to_data_1L_chr),T,F))
```

```
A <- renewSlot(A, "c_SpecificProject", SpecificModels(a_YouthvarsProfile = A@a_ScorzProfile@a_YouthvarsProfile,
                                                      b_SpecificParameters = A@b_SpecificParameters,
                                                      paths_chr = {
                                                        if(is.na(path_to_data_1L_chr)){
                                                          tempdir()
                                                          }else{
                                                           normalizePath("../Data")
                                                          }
                                                        }))
```

We confirm that our parameters and dataset are internally consistent. This step will also rename dataset variables that use certain incompatible naming conventions.

```
A <- ratifySlot(A, "c_SpecificProject")
```

We now begin to create sub-directories of the nominated output data directory.

```
A <- renewSlot(A, "c_SpecificProject",
               authorSlot(A, "c_SpecificProject", what_1L_chr = "workspace"))
```

We now generate tables and charts that describe our dataset. These are saved in a sub-directory of our output data directory.

```
A <- renewSlot(A, "c_SpecificProject",
               authorSlot(A, "c_SpecificProject", what_1L_chr = "descriptives",
                          digits_1L_int = 3L))
```

4.3 Undertake exploratory modelling using baseline dataset

4.3.1 Identify preferred model types

We next compare the performance of different model types. This step saves model objects and plots to a sub-directory of our output directory.

```
A <- renewSlot(A, "c_SpecificProject",
               investigateSlot(A, "c_SpecificProject",
                               depnt_var_max_val_1L_dbl = 0.99,
                               session_ls = sessionInfo()))
```

After inspecting the output of the previous command, we can now specify the preferred model types to use from this point onwards.

```
A <- renewSlot(A, "c_SpecificProject",
               renew(procureSlot(A, "c_SpecificProject"),
                     new_val_xx = c("GLM_GSN_LOG", "OLS_CLL"),
                     type_1L_chr = "results",
                     what_1L_chr = "prefd_mdls"))
```

4.3.2 Identify preferred predictors and covariates

Next we assess multiple versions of our preferred model type - one single predictor model for each of our candidate predictors and the same models with candidate covariates added. A number of model/plot objects saved to a sub-directory of our output directory.

```
A <- renewSlot(A, "c_SpecificProject",
               investigateSlot(A,"c_SpecificProject"))
```

4.3.3 Assess performance of final model specification

After reviewing the output of the previous step, we specify the covariates we wish to add to the models.

```
A <- renewSlot(A, "c_SpecificProject",
               renew(procureSlot(A, "c_SpecificProject"),
                     new_val_xx = "SOFAS",
                     type_1L_chr = "results",
                     what_1L_chr = "prefd_covars"))
```

We now assess the multivariate models. More model/plot objects are saved to a sub-directory of our output directory.

```
A <- renewSlot(A, "c_SpecificProject",
               investigateSlot(A, "c_SpecificProject"))
```

4.4 Undertake longitudinal modelling

We next reformulate the models we finalised in the previous step so that they are suitable for modelling longitudinal change.

4.4.1 Primary analysis

For our primary analysis, we use the longitudinal formulation of the models we previously selected. A series of large model files are written to a sub-directory of the local output data directory.

```
A <- renewSlot(A, "c_SpecificProject",
               investigateSlot(A, "c_SpecificProject"))
```

4.4.2 Secondary analyses

For our secondary analyses, we specify alternative combinations of predictors and covariates.

```
A <- renewSlot(A, "c_SpecificProject",
               investigateSlot(A, "c_SpecificProject",
                               scndry_anlys_params_ls = make_scndry_anlys_params(candidate_predrs_chr = c("SOFAS"),
                                                                                 candidate_covar_nms_chr = c("d_sex_birth_s",
                                                                                                             "d_age",
                                                                                                             "d_sexual_ori_s",
                                                                                                             "d_studying_working"),
                                                                                 prefd_covars_chr = NA_character_) %>%
                                 make_scndry_anlys_params(candidate_predrs_chr = c("SCARED","OASIS","GAD7"),
                                                          candidate_covar_nms_chr = c("PHQ9", "SOFAS",
                                                                                      "d_sex_birth_s",
                                                                                      "d_age",
                                                                                      "d_sexual_ori_s",
                                                                                      "d_studying_working"),
                                                          prefd_covars_chr = "PHQ9")))
```

5 Report and disseminate findings

5.1 Create shareable models

The model objects created and saved in our working directory by the preceding steps are not suitable for public dissemination. They are both too large in file size and, more importantly, include copies of our source dataset. We can overcome these limitations by creating shareable versions of the models. Two types of shareable version are created - copies of the original model objects in which fake data overwrites the original source data and tables of model coefficients.

```
A <- renewSlot(A, "c_SpecificProject",
               authorData(procureSlot(A, "c_SpecificProject")))
```

6 Specify study reporting metadata

We create a `TTUSynopsis` object that contains the fields necessary to render and share reports.

```
A <- renewSlot(A, "d_TTUReports",
               {
                 Y <- metamorphoseSlot(A, "c_SpecificProject")
                 Y <- TTUSynopsis(a_Ready4showPaths = Y@a_Ready4showPaths,
                                  b_SpecificResults = Y@b_SpecificResults,
                                  c_SpecificParameters = Y@c_SpecificParameters,
                                  d_YouthvarsProfile = Y@d_YouthvarsProfile,
                                  rmd_fl_nms_ls = Y@rmd_fl_nms_ls)
                 Y <- TTUReports(a_TTUSynopsis = Y)
                 Y
                 }
               )
```

We add metadata relevant to the reports that we will be generating to these fields. Note that the data we supply to the Ready4useRepos module below must relate to a repository to which we have write permissions (otherwise subsequent steps will fail). If you do not have write permissions to the https://doi.org/10.7910/DVN/DKDIB0 dataverse dataset specified below, you can instead supply values relating to a repository that you do have write permissions to. Alternatively, you can skip any subsequent `share` or `shareSlot` commands that involve writing data to the online data repository.

```
A <- renewSlot(A, "d_TTUReports@a_TTUSynopsis",
               procureSlot(A, "d_TTUReports@a_TTUSynopsis") %>%
                 renewSlot("authors_r3",
                           {if(is.na(path_to_data_1L_chr)){ready4show::authors_tb}else{read.csv("CSVs/Authors.csv") %>%
                               tibble::as_tibble() %>% ready4show::ready4show_authors()}}) %>%
                 renewSlot("institutes_r3",
                           {if(is.na(path_to_data_1L_chr)){ready4show::institutes_tb }else{
                               read.csv("CSVs/Institutes.csv") %>% tibble::as_tibble() %>% ready4show::ready4show_institutes()}}) %>%
                 renewSlot("digits_int", c(3L,3L)) %>%
                 renewSlot("outp_formats_chr", c("PDF","PDF")) %>%
                 renewSlot("title_1L_chr",
                           ifelse(is.na(path_to_data_1L_chr),"A hypothetical study using fake data",
                                  "Mapping psychological distress, depression and anxiety measures to adolescent AQoL-6D utility using dat
                 renewSlot("correspondences_r3", old_nms_chr = c("PHQ9", "GAD7"), new_nms_chr = c("PHQ-9", "GAD-7")) %>%
                 renewSlot("e_Ready4useRepos",
                           Ready4useRepos(dv_nm_1L_chr = "TTU",
                                          dv_ds_nm_1L_chr = "https://doi.org/10.7910/DVN/DKDIB0", dv_server_1L_chr = "dataverse.harvard.ed
```

6.1 Describe and share models

6.1.1 Author model catalogues

We download a program for generating a catalogue of models and use it to summarising the models created under each study analysis (one primary and two secondary). The catalogues are saved locally.

```
authorSlot(A, "d_TTUReports", what_1L_chr = "Catalogue", download_tmpl_1L_lgl = T)
```

6.1.2 Share model catalogue

We share the catalogues that we created, uploading a copy to our study online repository. To run this step you will need write permissions to the online repository.

```
shareSlot(A, "d_TTUReports@a_TTUSynopsis", type_1L_chr = "Report", what_1L_chr = "Catalogue")
```

6.1.3 Share models

We share tables of coefficients and other meta-data about the models we have created by posting them to the online repository. The object we create and share is designed to be used in conjunction with the youthu package to make it easier to make predictions with these models using new data.

Again, you will need write permissions to the online repository.

```
shareSlot(A, "d_TTUReports@a_TTUSynopsis", type_1L_chr = "Models", what_1L_chr = "ingredients")
```

6.2 Author manuscript

We add some content about the manuscript we wish to author.

```
A <- renewSlot(A, "d_TTUReports@a_TTUSynopsis",
               procureSlot(A, "d_TTUReports@a_TTUSynopsis") %>%
                 renewSlot("background_1L_chr",
                           "Quality Adjusted Life Years (QALYs) are often used in economic evaluations, yet utility weights for deriving t
                 renewSlot("coi_1L_chr", "None declared") %>%
                 renewSlot("conclusion_1L_chr",
                           ifelse(is.na(path_to_data_1L_chr),
                                  "Nothing should be concluded from this study as it is purely hypothetical.",
                                  "Adolescent AQoL-6D utility can be derived from a range of psychological distress, depression and anxiet
                 renewSlot("ethics_1L_chr",
                           paste0("The study was reviewed and granted approval by ",
                                  ifelse(is.na(path_to_data_1L_chr),"no-one.",
                                         "the University of Melbourne's Human Research Ethics Committee and the local Human Ethics and Adv
                 renewSlot("funding_1L_chr",
                           paste0("The study was funded by ",
                                  ifelse(is.na(path_to_data_1L_chr),"no-one.",
                                         "the National Health and Medical Research Council (NHMRC, APP1076940), Orygen and headspace.")))
                 renewSlot("interval_chr", "three months") %>%
                 renewSlot("keywords_chr",
                           c("anxiety", "AQoL","depression", "psychological distress", "QALYs", "utility mapping")) %>%
                 renewSlot("sample_desc_1L_chr",
                           ifelse(is.na(path_to_data_1L_chr), "The study sample is fake data.","")) )
```

We create a summary of results that can be interpreted by the program that authors the manuscript.

```
A <- renewSlot(A, "d_TTUReports@a_TTUSynopsis@abstract_args_ls",
               manufactureSlot(A,"d_TTUReports@a_TTUSynopsis", what_1L_chr = "abstract_args_ls",
                               depnt_var_nms_chr = c("AQoL-6D", "Adolescent AQoL Six Dimension")))
```

```
A <- enhanceSlot(A, "d_TTUReports@a_TTUSynopsis", with_1L_chr = "results_ls",
                 depnt_var_nms_chr = c("AQoL-6D", "Adolescent AQoL Six Dimension"))
```

We create and save the plots that will be used in the manuscript.

```
authorSlot(A, "d_TTUReports", type_1L_chr = "Plots",
           depnt_var_desc_1L_chr = A@d_TTUReports@a_TTUSynopsis@b_SpecificResults@a_SpecificShareable@shareable_outp_ls$results_ls$study_d
```

We download a program for generating a template manuscript and run it to author an algorithm authored first draft of the manuscript.

```
authorSlot(A, "d_TTUReports", type_1L_chr = "Report", what_1L_chr = "Manuscript_Auto", download_tmpl_1L_lgl = T)
```

We can also download a program for generating a human-customised manuscript for the purposes of submission to a journal. If you are running this code as a reproduction study, the following command will download a program that has already been customised by study authors. Otherwise, the next command will just download another copy of the template manuscript authoring program into a different directory (which we call "Manuscript_Submission") so that you can then manually edit those files as required.

```
authorData(A@d_TTUReports@a_TTUSynopsis,
           tmpl_url_1L_chr = ifelse(is.na(path_to_data_1L_chr),A@d_TTUReports@manuscript_tmpl_chr[1],
                                    "https://github.com/orygen/aqol6dmaps_ss"),
           tmpl_version_1_L_chr = ifelse(is.na(path_to_data_1L_chr),
                                         A@d_TTUReports@manuscript_tmpl_chr[2],"0.4"),
           what_1L_chr = "Manuscript_Submission")
```

We create the files we will be using for figures submitted to our target journal and also a table of all the dependency packages used in our analysis and reporting.

```
authorSlot(A, "d_TTUReports", type_1L_chr = "Plots", what_1L_chr = "Manuscript_Submission", depnt_var_desc_1L_chr = "AQoL-6D")
```

```
data.frame(Package = c("youthvars","scorz","specific","TTU") %>%
                         purrr::map(~ {
                           utils::packageDescription(.x) %>% `[`(c("Depends", "Imports")) %>% purrr::map(~{
                               if(is.null(.x)){character(0)}else{.x %>% strsplit(",\\n") %>% purrr::flatten_chr() %>%
                                   purrr::map(~strsplit(.x,", ") %>%
                                                purrr::flatten_chr()) %>% purrr::flatten_chr() %>% sort() %>%
                                   purrr::discard(~startsWith(.x,"R "))}}) %>%
                             purrr::flatten_chr() %>% unique() %>% sort()}) %>%
                         purrr::reduce(~c(.x,.y)) %>% purrr::map_chr(~{
                           updated_1L_chr <- stringr::str_replace_all(.x,"\\n"," ")
                           problem_idx_1L_chr <- stringr::str_locate(updated_1L_chr," ")[1,1] %>% unname()
                           if(!is.na(problem_idx_1L_chr))
                             updated_1L_chr <- updated_1L_chr %>% stringr::str_sub(end = problem_idx_1L_chr-1)
                           updated_1L_chr %>% trimws(which = "left")}) %>% unique() %>% sort()) %>%
  dplyr::mutate(Version = Package %>% purrr::map_chr(~utils::packageDescription(.x) %>% purrr::pluck("Version")),
                Citation = Package %>% purrr::map_chr(~get_pkg_citation(.x))) %>%
  saveRDS(paste0(paste0(A@d_TTUReports@a_TTUSynopsis@a_Ready4showPaths@outp_data_dir_1L_chr,"/",
                     A@d_TTUReports@a_TTUSynopsis@a_Ready4showPaths@mkdn_data_dir_1L_chr,"/","Manuscript_Submission"),
                 "/packages.RDS"))
```

We can generate a version of the manuscript that is suitable for submission as a preprint.

```
if(!is.na(path_to_data_1L_chr)){
  authorReport(renewSlot(A,"d_TTUReports@a_TTUSynopsis@rmd_fl_nms_ls",
                       ready4show::make_rmd_fl_nms_ls(pdf_fl_nm_1L_chr = "Preprint_PDF")) %>%
                 procureSlot("d_TTUReports@a_TTUSynopsis"),
             fl_nm_1L_chr = "Preprint", what_1L_chr = "Manuscript_Submission")
}
```

We can also generate another version that is formatted for submission to a journal, with tables and figures appearing after the main article body.

```
A <- renewSlot(A, "d_TTUReports",
               procureSlot(A, "d_TTUReports") %>%
                 renewSlot("a_TTUSynopsis@tables_in_body_lgl",  F) %>%
                 renewSlot("a_TTUSynopsis@figures_in_body_lgl", F))
authorSlot(A, "d_TTUReports", what_1L_chr = "Manuscript_Submission", download_tmpl_1L_lgl = F)
```

We now generate the Supplementary Information for the submission manuscript.

```
if(!is.na(path_to_data_1L_chr)){
  authorReport(renewSlot(A,"d_TTUReports@a_TTUSynopsis@rmd_fl_nms_ls",
                       ready4show::make_rmd_fl_nms_ls(pdf_fl_nm_1L_chr = "TA_PP_PDF")) %>%
                 procureSlot("d_TTUReports@a_TTUSynopsis"),
             fl_nm_1L_chr = "Supplement_For_Preprint", what_1L_chr = "Manuscript_Submission")
}
authorReport(renewSlot(A,"d_TTUReports@a_TTUSynopsis@rmd_fl_nms_ls",
                       ready4show::make_rmd_fl_nms_ls(pdf_fl_nm_1L_chr = "TA_PDF")) %>%
               procureSlot("d_TTUReports@a_TTUSynopsis"),
             fl_nm_1L_chr = "Supplement", what_1L_chr = "Manuscript_Submission")
```

7 Tidy workspace

The preceding steps saved multiple objects (mostly R model objects) that have embedded within them copies of the source dataset. We can now purge all such copies from our output data directory.

```
author(procureSlot(A,"c_SpecificProject"), type_1L_chr = "purge_write")
```