Populating the study package

Jenna M. Reps

2020-10-14

Contents

1 Introduction

This vignette describes how one can populate the SkeletonExistingModelStudy package with the target cohort, outcome cohorts and model settings.

The first step is to make a copy of the 'SkeletonExistingModelStudy' R package template. You can do this by cloning the github package and then copy and paste the directory to a location in your computer.

Open the Skeleton R project in R studio, this can be done by finding and double clicking on the SkeletonExistingModelStudy.Rproj file in the folder that you just copy and pasted. Once the package project is opened in R studio there are 5 steps that must be followed:

1. Source extras/populatePackage.Rto add functions needed to populate the package
2. Rename the package using: replaceName()
3. Add the target and outcome cohort using: populatePackageCohorts()
4. Specify the model using: populatePackageModels() - this can be run multiple times to add multiple models
5. Build the study package

The package is now ready to run or share on GitHub.

1.1 Step 1: Source './extras/populatePackage.R'

The R script extras/populatePackage.R contains multiple functions that can be used to populate the skeleton package.

To add the functions to your environment, make sure the package R project is open in R studio and run:

```
source('./extras/populatePackage.R')
```

This will make the functions 'replaceName()', 'populatePackageCohorts()' and 'populatePackageModels()' available to use within your R session.

1.2 Step 2: Renaming the package

The package is still currently called 'SkeletonExistingPredictionModelStudy' but you need to rename all the files with the name of your study. To make this easy I created a function 'replaceName()' that will automatically replace 'SkeletonExistingPredictionModelStudy' with your new study name. Simply run:

```
replaceName(packageLocation = getwd(),
            packageName = 'PooledCohortvalidation')
```

When you open the r project it will set your working directory to the package directory - so 'packageLocation = getwd()' should work. If this has not occurred, you may need to manually specify the location where 'SkeletonExistingModelStudy.Rproj' exists in the skeleton copy you made. Set 'packageName' to be the name you want to call your study. The example above will replace 'SkeletonExistingModelStudy' with 'PooledCohortvalidation' throughout the skeleton package.

It is a good practice to close the R project and reopen it once renamed. The R project will now be renamed, in the example about it would be called 'PooledCohortvalidation.Rproj'.

1.3 Step 3: Adding the Target and Outcome cohorts

The 'populatePackageCohorts()' function requires users to specify:

- targetCohortIds - The ATLAS id/s for the target cohort/s
- targetCohortNames - A string or vector of string with sharable name/s for the target cohort/s
- outcomeIds - The ATLAS id/s for the outcome cohort/s
- outcomeNames - A string or vector of string with sharable name/s for the outcome cohort/s
- baseUrl - The url for the ATLAS webapi (this will be used to extract the ATLAS cohorts)

For example, to insert the target cohort:

- 'Pooled cohort equation target - non-black male' - that has the atlas ID 18941
- 'Pooled cohort equation target - non-black female' - that has the atlas ID 18942
- 'Pooled cohort equation target - black female' - that has the atlas ID 18943
- 'Pooled cohort equation target - black male' - that has the atlas ID 18944

and the outcome cohorts:

- 'Earliest of AMI ischemic stroke or cardiovascular death' - that has the atlas ID 18945
- 'Acute myocardial infarction events' - that has the atlas ID 18935
- 'Ischemic stroke events' - that has the atlas ID 18936

from the webapi at 'https://yourWebAPI' run:

```
populatePackageCohorts(targetCohortIds = c(18941, 18942, 18943, 18944),
                       targetCohortNames = c('Pooled cohort equation target - non-black male',
                                             'Pooled cohort equation target - non-black female',
                                             'Pooled cohort equation target - black female',
                                             'Pooled cohort equation target - black male'),
                       outcomeIds = c(18945, 18935, 18936),
                       outcomeNames = c('Earliest of AMI ischemic stroke or cardiovascular death',
                                        'Acute myocardial infarction events',
                                        'Ischemic stroke events'),
                       baseUrl = 'https://yourWebAPI')
```

This then inserts each cohort into 'inst/cohorts' as a json with the file name targetCohortNames or outcomeNames '.json' and inserts in 'inst/sql/sql_server' as a sql with the file name targetCohortNames or outcomeNames '.sql'.

It also creates a csv file in 'inst/settings' named 'CohortToCreate.csv' that specifies all the target and outcome cohorts to generate when the study is executed.

1.4 Step 4: Specify the model

The next step is to insert the model settings into the package. This can be done by running the function 'populatePackageModels()' per model to incorporate.

There are 6 main inputs into this function:

Table 1: The inputs into the populatePackageModels function

Input	Description
modelname	The name of the model you are specifying
standardCovariates	A list of settings for predictors that are standardCovariates
cohortCovariateSettings	A list of settings for predictors that are cohort covariates
measurementCovariateSettings	A list of settings for predictors that are measurment covariates
measurementCohortCovariateSettings	A list of settings for predictors that are measurement cohort covariates
finalMapping	The mapping from total score to predicted risk

This 6 inputs specifiy the model name, the predictors and the final model mapping.

1.4.1 modelname

The 'modelname' input is use when saving the model settings. The main model settings are saved in inst/settings as modelname_model.csv. Each row in modelname_model.csv specifies a covariate and the corresponding number of points (i.e., coefficient value) and settings required to create the covariate using standardCovariates, createCohortCovariate, createMeasurementCovariate, createMeasurementCohortCovariate or createAgeCovariate.

1.4.2 standardCovariates

The 'standardCovariates' input is used when you want to include standard FeatureExtraction covariates. The input is a list of settings that specify what standard covariates to use (when you execute the study these will be used to configure FeatureExtraction::createCovariateSettings()) and the corresponding points.

Table 2: The inputs into the standardCovariates function

Input	Description	Example
covariateId	A vector of the standard covariateIds to include	c(12003, 13003, 8507001)
covariateName	A vector of the corresponding names	c("Age 60-64", "Age 65-7
points	A vector of the corresponding points	c(10,10,15)
featureExtraction	A vector of the inputs needed in FeatureExtraction::createCovariateSettings	c("useDemographicsAgeG

The 'covariateId' values must match the standard ids created by 'FeatureExtraction::createCovariateSettings'. These are often the conceptId*1000+analysisId, where the analysisId can be found in https://github.com/OHDSI/FeatureExtraction/blob/master/inst/csv/PrespecAnalyses.csv . When the 'populatePackageModels' function is executed with 'standardCovariates' set then the model settings 'inst/settings/modelname_model.csv' is created or appended to with the standard covariate settings. If your model does not require standard covariates, then set 'standardCovariates' to NULL.

1.4.3 cohortCovariateSettings

The 'cohortCovariateSettings' is used when you want to include covariates using cohort definitions. This can be useful for more complex features such as a diabetes definition that requires a diabetes condition record and a treatment record within 60 days. The 'cohortCovariateSettings' input is a list containing the following

entries:

Table 3: The inputs into the cohortCovariateS

Input	Description
baseUrl	A baseUrl for the ATLAS webapi that contains the cohorts
atlasCovariateIds	A vector of the atlas cohort ids
atlasCovariateNames	A vector of the atlas cohort names
analysisIds	A vector of analysis ids for the cohort covariates
startDays	Find cohort entries where the cohort_start_date occurs after the the startDays relative to index
endDays	Find cohort entries where the cohort_start_date occurs before the the endsDays relative to index
points	A vector of the points
count	A vector specifying whether to use binary indicators or count the number of cohort_start_dates per
ageInteraction	A vector specifying whether to use the age interaction
lnAgeInteraction	A vector specifying whether to use the natural logarithm log(age) interaction

The 'cohortCovariateSettings' saves settings required by 'createCohortCovariateSettings'. When the 'populatePackageModels' function is executed with 'cohortCovariateSettings' set then the specified ATLAS cohorts are downloaded into the package using the webapi and the model settings 'inst/settings/modelname_model.csv' is created or appended to with the cohort covariate settings. If you do not want to include cohort covariates set 'cohortCovariateSettings' to NULL.

1.4.4 measurementCovariateSettings

The 'measurementCovariateSettings' is used when you want to include measurement covariates. This can be useful for more complex features such as total cholesterol. The 'measurementCovariateSettings' input is a list containing the following entries:

Input	Description
names	A vector of the measurement covariate names
conceptSets	A list containing vectors of the measurement concept sets
startDays	Find measurement entries where the measurement_date occurs after the the startDays relative to inde
endDays	Find measurement entries where the measurement_date occurs before the the endsDays relative to ind
scaleMaps	A list of functions specifying how to standardise the measurements
points	A vector of the points
aggregateMethods	A vector specifying how to pick a single measurement value per patient. Choose from recent (closest to
imputationValues	A vector of values used to impute when a measurement is missing for a patient
ageInteractions	A vector specifying whether to use the age interaction
lnAgeInteractions	A vector specifying whether to use the natural logarithm log(age) interaction
lnValues	A vector specifying whether to use the log measurement value
measurementIds	A vector specifying the measurement ids
analysisIds	A vector specifying the analysis ids

The 'measurementCovariateSettings' saves settings required by 'createMeasurementCovariateSettings'. When the 'populatePackageModels' function is executed with 'measurementCovariateSettings' set then the model settings 'inst/settings/modelname_model.csv' is created or appended to with the neasurement covariate settings. The mapping functions for each measurement covariate are saved as rds files into the 'inst/settings' directory. If you do not want to include measurement covariates set 'measurementCovariateSettings' to NULL.

1.4.5 measurementCohortCovariateSettings

The 'measurementCohortCovariateSettings' is used when you want to include measurement covariates where patients are also in or not in some specified cohort. This can be useful for more complex features such as treated systolic blood pressure or untreated systolic blood pressure. The 'measurementCohortCovariateSettings' input is a list containing the following entries:

Input	Description
names	A vector of the measurement covariate names
atlasCovariateIds	A vector of the cohort ids
types	A vector of "in" or "out" indicating whether to require being in, or not in, the cohort.
conceptSets	A list containing vectors of the measurement concept sets
startDays	Find measurement entries where the measurement_date occurs after the the startDays relative to inde
endDays	Find measurement entries where the measurement_date occurs before the the endsDays relative to ind
scaleMaps	A list of functions specifying how to standardise the measurements
points	A vector of the points
aggregateMethods	A vector specifying how to pick a single measurement value per patient. Choose from recent (closest to
ageInteractions	A vector specifying whether to use the age interaction
lnAgeInteractions	A vector specifying whether to use the natural logarithm log(age) interaction
lnValues	A vector specifying whether to use the log measurement value
measurementIds	A vector specifying the measurement ids
analysisIds	A vector specifying the analysis ids
baseUrl	ATLAS webapi address

The 'measurementCohortCovariateSettings' saves settings required by 'createMeasurementCohortCovariateSettings'. When the 'populatePackageModels' function is executed with 'measurementCohortCovariateSettings' set then the model settings 'inst/settings/modelname_model.csv' is created or appended to with the neasurement covariate settings and all specified ATLAS cohort are downloaded using the webapi into 'inst/cohorts' and 'inst/sql/sql_server'. The mapping functions for each measurement covariate are saved as rds files into the 'inst/settings' directory. If you do not want to include measurement based on being in/not in a cohort covariates set 'measurementCohortCovariateSettings' to NULL.

1.4.6 finalMapping

The 'finalMapping' input is a function that maps from the sum of the points to a predicted risk. For example, if the model is a logistic regression, the following mapping would be used:

```
function(x){return(1/(1+exp(-x)))}
```

This gets saved as an rds file in the 'inst/setting' directory.

1.4.7 Example

A complete example code for inserting the pooled cohort question non-black model into the skeleton package:

```
populatePackageModels(modelname = 'pooled_male_non_black',
                      standardCovariates = NULL,
                      cohortCovariateSettings = list(baseUrl = 'https://yourWebAPI',
                                                     atlasCovariateIds = c(17720,18957,18957),
                                                     atlasCovariateNames = c('diabetes', 'smoking','smol
                                                     analysisIds = c(456,456,455),
                                                     startDays = c(-9999,-730,-730),
```

```
                                        endDays = c(-1,60,60),
                                        points = c(0.658,7.837,-1.795),
                                        count = rep(F, 3),
                                        ageInteraction = c(F,F,F),
                                        lnAgeInteraction = c(F,F,T)) ,

                measurementCovariateSettings = list(names = c('Total_Cholesterol_mgdL', 'Total_Cho
                                                              'HDL-C_mgdL', 'HDL-C_mgdL'
                                                              ),
                                                     conceptSets = list(c(2212267,3015232,3019900,3
                                                                          c(2212267,3015232,3019900,3
                                                                          c(2212449,3003767,3007070,3
                                                                          c(2212449,3003767,3007070,3
                                                     ),
                                                     startDays = c(-365, -365,-365, -365),
                                                     endDays = c(60,60,60,60),
                                                     scaleMaps= list(function(x){ x = dplyr::mutate
                                                                     function(x){ x = dplyr::mutate
                                                                     function(x){ x = dplyr::mutate
                                                                     function(x){ x = dplyr::mutate
                                                     points = c(11.853,-2.664,-7.990,1.769),
                                                     aggregateMethods = c('recent','recent','recent
                                                     imputationValues = c(150,150,50,50),
                                                     ageInteractions = c(F,F,F,F),
                                                     lnAgeInteractions = c(F,T,F,T),
                                                     lnValues = c(T,T,T,T),
                                                     measurementIds = c(1,2,3,4),
                                                     analysisIds = c(457,457,457,457)


                ),

                ageCovariateSettings = list(names = c('log(age)'),
                                            ageMaps = list(function(x){return(log(x))}),
                                            ageIds = 1,
                                            analysisIds = c(458),
                                            points = c(12.344)

                ),

                measurementCohortCovariateSettings = list(names = c('treated_Systolic_BP_mm_Hg','t
                                                                    atlasCovariateIds = c('18946','18946'),
                                                                    types = c('in', 'out'),
                                                                    conceptSets = list(c(3004249,3009395,30
                                                                                       c(3004249,3009395,30

                                                                    ),
                                                                    startDays = c(-365,-365),
                                                                    endDays = c(60,60),
                                                                    scaleMaps= list(function(x){ x = dplyr:
                                                                                    function(x){ x = dplyr:
                                                                    ),
                                                                    points = c(1.797,1.764),
```

```
                                                                    aggregateMethods = c('recent','recent'),
                                                                    imputationValues = c(120,120),
                                                                    ageInteractions = c(F,F),
                                                                    lnAgeInteractions = c(F,F),
                                                                    lnValues = c(T,T),
                                                                    measurementIds = c(1,2),
                                                                    analysisIds = c(459,459),
                                                                    baseUrl = 'https://yourWebAPI'


                        ),

                        finalMapping = function(x){ 1- 0.9144^exp(x-61.18)}
)
```

1.5 Step 5: Build the study package

Aftering adding the settings into the package, you now need to build the package. Use the standard process (in R studio press the 'Build' tab in the top right corner and then select the 'Install and Restart' button) to build the study package so an R library is created.