Using the measurement covariate code

Jenna M. Reps

2020-10-13

Contents

1 Introduction **1**
1.1 createMeasurementCovariateSettings 1
1.2 Example 2

1 Introduction

This vignette describes how one can use the function 'createMeasurementCovariateSettings' to define measurement covariates using the OMOP CDM. You will need:

1. A concept set for the measurements (a vector of measurement_concept_ids)
2. A function to standardise the measurements (e.g., filters out unlikely values or converts units)
3. A value to use for imputation
4. How to aggregate multiple measurement values (e.g., use most recent to index, max value, min value or mean value)

1.1 createMeasurementCovariateSettings

This function contains the settings required to define the covariate. For a measurement covariate, the code will check the measurement table in the OMOP CDM to find all rows where the measurement_concept_id is in the specified measurement concept set. It will then check whether the measurement_date column calls between the index date plus the 'startDay' and the index date plus the 'endDay'. The 'scaleMap' will map the measurement values to a uniform scale - this standardises the values. If there are multiple measurements within the time period then the 'aggregateMethod' method with specify how to get a single value. The 'imputationValue' input specifies what value to assign to people without a measurement recorded during the time period. The settings 'ageInteraction' and 'lnAgeInteraction' enable the user to create age/ln(age) interaction terms. The 'lnValue' enables the user to use the natural logarithm of the measurment value. Finally, the 'analysisId' is used to create the cohort covariateId as 1000*'measurementId' + 'analysisId'.

Table 1: The inputs into the create function

Input	Description
covariateName	The name of the covariate
covariateId	The id of the covariate - generally measurementId*1000+analysisId
conseptSet	A vector of concept_ids corresponding to the measurement
startDay	How many days prior to index to see whether the measurement occurs after
endDay	How many days relative to index to see whether the mesurement occurs before
scaleMap	A function that takes the covariate Amdromeda table as input and processes it - can include filtering in
aggregateMethod	How to pick a measurement value when there are more than 1 during the start and end dates - can be m
imputationValue	A value to use if a person has no measurement during the start and end dates
ageInteraction	Include interaction with age

Input	Description
lnAgeInteraction	Include interaction with ln(age)
lnValue	Whether to us the natural log of the measurement value
analysisId	The analysis id for the covariate

1.2 Example

Assuming the concept set c(2212267, 3015232, 3019900, 3027114, 4008265, 4190897, 4198448, 4260765, 37393449, 37397989, 40484105, 44791053, 44809580) corresponds to 'Total Cholesterol'.

We create a function to map the covariate object (this contains the measurementConceptId, unitConceptId, rawValue and valueAsNumber columns) to standardise the measurement values. The unitConceptIds 8840,8954,9028 are all in mg per dL but unitConceptId 8753 corresponds to a different scale and needs to be multipled by 38.6 to covert to mg per dL. We remove any values less than 80 and greater than 500 and then return the scaled value.

```
function(x){ x = dplyr::mutate(x, rawValue = dplyr::case_when(unitConceptId == 8753 ~ rawValue*38.6, uni
```

We decide to impute missing values with 150. We include all measurement values with occured within 1 year prior to index and up to 60 days after, but use the value that occurred closest to index.

To create a Total Cholesterol in mg per dL covariate using a measurement covariate run:

```
cohortCov1 <- createCohortCovariateSettings(covariateName = 'Total Cholesterol in mg per dL',
                                            analysisId = 457,
                                            covariateId = 1*1000+457,
                                            conseptSet = c(2212267, 3015232, 3019900, 3027114, 4008265,
                                            startDay= -365,
                                            endDay=60,
                                            scaleMap = function(x){ x = dplyr::mutate(x, rawValue = dply
                                            aggregateMethod= 'recent',
                                            imputationValue = 150,
                                            ageInteraction = FALSE,
                                            lnAgeInteraction = FALSE,
                                            lnValue = FALSE,
                                            analysisId = 457)
```

If we wanted to use the natureal logarithm of the most recent measurement value then we can use:

```
cohortCov2 <- createCohortCovariateSettings(covariateName = 'Log Total Cholesterol in mg per dL',
                                            analysisId = 457,
                                            covariateId = 2*1000+457,
                                            conseptSet = c(2212267, 3015232, 3019900, 3027114, 4008265,
                                            startDay= -365,
                                            endDay=60,
                                            scaleMap = function(x){ x = dplyr::mutate(x, rawValue = dply
                                            aggregateMethod= 'recent',
                                            imputationValue = 150,
                                            ageInteraction = FALSE,
                                            lnAgeInteraction = FALSE,
                                            lnValue = TRUE,
                                            analysisId = 457)
```

To include age interactions:

```
cohortCov3 <- createCohortCovariateSettings(covariateName = 'Total Cholesterol in mg per dL interaction
                                            analysisId = 457,
                                            covariateId = 3*1000+457,
                                            conseptSet = c(2212267, 3015232, 3019900, 3027114, 4008265,
                                            startDay= -365,
                                            endDay=60,
                                            scaleMap = function(x){ x = dplyr::mutate(x, rawValue = dpl
                                            aggregateMethod= 'recent',
                                            imputationValue = 150,
                                            ageInteraction = TRUE,
                                            lnAgeInteraction = FALSE,
                                            lnValue = FALSE,
                                            analysisId = 457)
```

To include all the above as covariates, combine them into a list:

```
cohortCov <- list(cohortCov1,cohortCov2,cohortCov3)
```