Package: omock (via r-universe)

September 17, 2024

Type Package

Title Creation of Mock Observational Medical Outcomes Partnership
Common Data Model

Version 0.2.0

Maintainer Mike Du <mike.du@ndorms.ox.ac.uk>

Description Creates mock data for testing and package development for
the Observational Medical Outcomes Partnership common data
model. The package offers functions crafted with
pipeline-friendly implementation, enabling users to
effortlessly include only the necessary tables for their
testing needs.

License Apache License (>= 2)

URL https://ohdsi.github.io/omock/, https://github.com/ohdsi/omock

Depends R (>= 2.10)

Imports cli, dplyr, lubridate, methods, omopgenerics (>= 0.3.0),
purrr, rlang, snakecase, stats

Suggests ggplot2, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

BugReports https://github.com/ohdsi/omock/issues

Repository https://ohdsi.r-universe.dev

RemoteUrl https://github.com/ohdsi/omock

RemoteRef HEAD

RemoteSha b6f860c7ecc8a6471e3cd6137ff6edef372d88da

Contents

---

mockCdmFromTables   *Generates a mock CDM (Common Data Model) object based on exist-*
                                ing CDM structures and additional tables.

Description

This function takes an existing CDM reference (which can be empty) and a list of additional named
tables to create a more complete mock CDM object. It ensures that all provided observations fit
within their respective observation periods and that all individual records are consistent with the en-
tries in the person table. This is useful for creating reliable and realistic healthcare data simulations
for development and testing within the OMOP CDM framework.

Usage

```
mockCdmFromTables(cdm = mockCdmReference(), tables = list(), seed = NULL)
```

Arguments

cdm   A 'cdm_reference' object, which serves as the base structure where all addi-
                       tional tables will be integrated. This parameter should already be initialized and
                       can contain pre-existing standard or cohort-specific OMOP tables.

tables   A named list of data frames representing additional tables to be integrated into
                       the CDM. These tables can include both standard OMOP tables such as 'drug_exposure'
                       or 'condition_occurrence', as well as cohort-specific tables that are not part of
                       the standard OMOP model but are necessary for specific analyses. Each table
                       should be named according to its intended table name in the CDM structure.

seed   An optional integer that sets the seed for random number generation used in
                       creating mock data entries. Setting a seed ensures that the generated mock data
                       are reproducible across different runs of the function. If 'NULL', the seed is not
                       set, leading to non-deterministic behavior in data generation.

Value

Returns the updated 'cdm' object with all the new tables added and integrated, ensuring consistency across the observational periods and the person entries.

Examples

```
library(omock)
library(dplyr)

# Create a mock cohort table
cohort <- tibble(
  cohort_definition_id = c(1, 1, 2, 2, 1, 3, 3, 3, 1, 3),
  subject_id = c(1, 4, 2, 3, 5, 5, 4, 3, 3, 1),
  cohort_start_date = as.Date(c(
    "2020-04-01", "2021-06-01", "2022-05-22", "2010-01-01", "2019-08-01",
    "2019-04-07", "2021-01-01", "2008-02-02", "2009-09-09", "2021-01-01"
  )),
  cohort_end_date = cohort_start_date
)

# Generate a mock CDM from preexisting CDM structure and cohort table
cdm <- mockCdmFromTables(cdm = mockCdmReference(), tables = list(cohort = cohort))

# Access the newly integrated cohort table and the standard person table in the CDM
print(cdm$cohort)
print(cdm$person)
```

---

mockCdmReference	*Creates an empty CDM (Common Data Model) reference for a mock database.*

Description

This function initializes an empty CDM reference with a specified name and populates it with mock vocabulary tables based on the provided vocabulary set. It is particularly useful for setting up a simulated environment for testing and development purposes within the OMOP CDM framework.

Usage

```
mockCdmReference(cdmName = "mock database", vocabularySet = "mock")
```

Arguments

cdmName   A character string specifying the name of the CDM object to be created.This name can be used to identify the CDM object within a larger simulation or testing framework. Default is "mock database".

vocabularySet   A character string that specifies the name of the vocabulary set to be used in creating the vocabulary tables for the CDM. This allows for the customization of the vocabulary to match specific testing scenarios. Default is "mock".

Value

Returns a CDM object that is initially empty but includes mock vocabulary tables.The object structure is compliant with OMOP CDM standards, making it suitable for further population with mock data like person, visit, and observation records.

Examples

```
library(omock)

# Create a new empty mock CDM reference
cdm <- mockCdmReference()

# Display the structure of the newly created CDM
print(cdm)
```

---

mockCohort   *Generate Synthetic Cohort*

Description

This function generates synthetic cohort data and adds it to a given CDM (Common Data Model) reference. It allows for creating multiple cohorts with specified properties and simulates the frequency of observations for individuals.

Usage

```
mockCohort(
  cdm,
  name = "cohort",
  numberCohorts = 1,
  cohortName = paste0("cohort_", seq_len(numberCohorts)),
  recordPerson = 1,
  seed = NULL
)
```

Arguments

cdm   A CDM reference object where the synthetic cohort data will be stored. This
               object should already include necessary tables such as 'person' and 'observation_period'.

name   A string specifying the name of the table within the CDM where the cohort data
               will be stored. Defaults to "cohort". This name will be used to reference the new
               table in the CDM.

numberCohorts An integer specifying the number of different cohorts to create within the table.
               Defaults to 1. This parameter allows for the creation of multiple cohorts, each
               with a unique identifier.

cohortName	A character vector specifying the names of the cohorts to be created. If not provided, default names based on a sequence (e.g., "cohort_1", "cohort_2", ...) will be generated. The length of this vector must match the value of 'numberCohorts'. This parameter provides meaningful names for each cohort.
recordPerson	An integer or a vector of integers specifying the expected number of records per person within each cohort. If a single integer is provided, it applies to all cohorts. If a vector is provided, its length must match the value of 'numberCohorts'. This parameter helps simulate the frequency of observations for individuals in each cohort, allowing for realistic variability in data.
seed	An integer specifying the random seed for reproducibility of the generated data. Setting a seed ensures that the same synthetic data can be generated again, facilitating consistent results across different runs.

Value

A CDM reference object with the mock cohort tables added. The new table will contain synthetic data representing the specified cohorts, each with its own set of observation records.

Examples

```
library(omock)
cdm <- mockCdmReference() |>
  mockPerson(nPerson = 100) |>
  mockObservationPeriod() |>
  mockCohort(
    name = "omock_example",
    numberCohorts = 2,
    cohortName = c("omock_cohort_1", "omock_cohort_2")
  )

cdm
```

---

mockConcepts	*Adds mock concept data to a concept table within a Common Data Model (CDM) object.*

Description

This function inserts new concept entries into a specified domain within the concept table of a CDM object.It supports four domains: Condition, Drug, Measurement, and Observation. Existing entries with the same concept IDs will be overwritten, so caution should be used when adding data to prevent unintended data loss.

Usage

```
mockConcepts(cdm, conceptSet, domain = "Condition", seed = NULL)
```

Arguments

`cdm`	A CDM object that represents a common data model containing at least a concept table.This object will be modified in-place to include the new or updated concept entries.
`conceptSet`	A numeric vector of concept IDs to be added or updated in the concept table.These IDs should be unique within the context of the provided domain to avoid unintended overwriting unless that is the intended effect.
`domain`	A character string specifying the domain of the concepts being added.Only accepts "Condition", "Drug", "Measurement", or "Observation". This defines under which category the concepts fall and affects which vocabulary is used for them.
`seed`	An optional integer value used to set the random seed for generating reproducible concept attributes like 'vocabulary_id' and 'concept_class_id'. Useful for testing or when consistent output is required.

Value

Returns the modified CDM object with the updated concept table reflecting the newly added concepts.The function directly modifies the provided CDM object.

Examples

```
library(omock)
library(dplyr)

# Create a mock CDM reference and add concepts in the 'Condition' domain
cdm <- mockCdmReference() |> mockConcepts(
conceptSet = c(100, 200), domain = "Condition")

# View the updated concept entries for the 'Condition' domain
cdm$concept |> filter(domain_id == "Condition")
```

---

mockConditionOccurrence

Generates a mock condition occurrence table and integrates it into an existing CDM object.

Description

This function simulates condition occurrences for individuals within a specified cohort. It helps create a realistic dataset by generating condition records for each person, based on the number of records specified per person.The generated data are aligned with the existing observation periods to ensure that all conditions are recorded within valid observation windows.

Usage

```
mockConditionOccurrence(cdm, recordPerson = 1, seed = NULL)
```

Arguments

cdm	A 'cdm_reference' object that should already include 'person', 'observation_period', and 'concept' tables. This object is the base CDM structure where the condition occurrence data will be added. It is essential that these tables are not empty as they provide the necessary context for generating condition data.
recordPerson	An integer specifying the expected number of condition records to generate per person. This parameter allows the simulation of varying frequencies of condition occurrences among individuals in the cohort, reflecting the variability seen in real-world medical data.
seed	An optional integer used to set the seed for random number generation, ensuring reproducibility of the generated data. If provided, it allows the function to produce the same results each time it is run with the same inputs. If 'NULL', the seed is not set, resulting in different outputs on each run.

Value

Returns the modified 'cdm' object with the new 'condition_occurrence' table added. This table includes the simulated condition data for each person, ensuring that each record is within the valid observation periods and linked to the correct individuals in the 'person' table.

Examples

```
library(omock)

# Create a mock CDM reference and add condition occurrences
cdm <- mockCdmReference() |>
  mockPerson() |>
  mockObservationPeriod() |>
  mockConditionOccurrence(recordPerson = 2)

# View the generated condition occurrence data
print(cdm$condition_occurrence)
```

---

mockDeath	*Generates a mock death table and integrates it into an existing CDM object.*

Description

This function simulates death records for individuals within a specified cohort. It creates a realistic dataset by generating death records according to the specified number of records per person. The function ensures that each death record is associated with a valid person within the observation period to maintain the integrity of the data.

Usage

```
mockDeath(cdm, recordPerson = 1, seed = NULL)
```

Arguments

cdm   A 'cdm_reference' object that must already include 'person' and 'observation_period'
                       tables.This object is the base CDM structure where the death data will be added.
                       It is essential that the 'person' and 'observation_period' tables are populated as
                       they provide necessary context for generating death records.

recordPerson   An integer specifying the expected number of death records to generate per per-
                       son. This parameter helps simulate varying frequencies of death occurrences
                       among individuals in the cohort, reflecting the variability seen in real-world
                       medical data. Typically, this would be set to 1 or 0, assuming most datasets
                       would only record a single death date per individual if at all.

seed   An optional integer used to set the seed for random number generation, ensur-
                       ing reproducibility of the generated data. If provided, it allows the function to
                       produce the same results each time it is run with the same inputs. If 'NULL',
                       the seed is not set, which can result in different outputs on each run.

Value

Returns the modified 'cdm' object with the new 'death' table added. This table includes the simu-
lated death data for each person, ensuring that each record is linked correctly to individuals in the '
person' table and falls within valid observation periods.

Examples

```
library(omock)

# Create a mock CDM reference and add death records
cdm <- mockCdmReference() |>
  mockPerson() |>
  mockObservationPeriod() |>
  mockDeath(recordPerson = 1)

# View the generated death data
print(cdm$death)
```

---

mockDrugExposure   *Generates a mock drug exposure table and integrates it into an existing*
                            CDM object.

Description

This function simulates drug exposure records for individuals within a specified cohort. It creates a
realistic dataset by generating drug exposure records based on the specified number of records per
person. Each drug exposure record is correctly associated with an individual within valid observa-
tion periods, ensuring the integrity of the data.

Usage

```
mockDrugExposure(cdm, recordPerson = 1, seed = NULL)
```

Arguments

cdm	A 'cdm_reference' object that must already include 'person' and 'observation_period' tables. This object serves as the base CDM structure where the drug exposure data will be added. The 'person' and 'observation_period' tables must be populated as they are necessary for generating accurate drug exposure records.
recordPerson	An integer specifying the expected number of drug exposure records to generate per person. This parameter allows for the simulation of varying drug usage frequencies among individuals in the cohort, reflecting real-world variability in medication administration.
seed	An optional integer used to set the seed for random number generation, ensuring reproducibility of the generated data. If provided, this seed enables the function to produce consistent results each time it is run with the same inputs. If 'NULL', the seed is not set, which can lead to different outputs on each run.

Value

Returns the modified 'cdm' object with the new 'drug_exposure' table added. This table includes the simulated drug exposure data for each person, ensuring that each record is correctly linked to individuals in the 'person' table and falls within valid observation periods.

Examples

```
library(omock)

# Create a mock CDM reference and add drug exposure records
cdm <- mockCdmReference() |>
  mockPerson() |>
  mockObservationPeriod() |>
  mockDrugExposure(recordPerson = 3)

# View the generated drug exposure data
print(cdm$drug_exposure)
```

---

mockMeasurement	*Generates a mock measurement table and integrates it into an existing CDM object.*

Description

This function simulates measurement records for individuals within a specified cohort. It creates a realistic dataset by generating measurement records based on the specified number of records per person. Each measurement record is correctly associated with an individual within valid observation periods, ensuring the integrity of the data.

Usage

```
mockMeasurement(cdm, recordPerson = 1, seed = NULL)
```

Arguments

cdm   A 'cdm_reference' object that must already include 'person' and 'observation_period' tables. This object serves as the base CDM structure where the measurement data will be added. The 'person' and 'observation_period' tables must be populated as they are necessary for generating accurate measurement records.

recordPerson An integer specifying the expected number of measurement records to generate per person. This parameter allows for the simulation of varying frequencies of health measurements among individuals in the cohort, reflecting real-world variability in patient monitoring and diagnostic testing.

seed   An optional integer used to set the seed for random number generation, ensuring reproducibility of the generated data. If provided, this seed enables the function to produce consistent results each time it is run with the same inputs. If 'NULL', the seed is not set, which can lead to different outputs on each run.

Value

Returns the modified 'cdm' object with the new 'measurement' table added. This table includes the simulated measurement data for each person, ensuring that each record is correctly linked to individuals in the 'person' table and falls within valid observation periods.

Examples

```
library(omock)

# Create a mock CDM reference and add measurement records
cdm <- mockCdmReference() |>
  mockPerson() |>
  mockObservationPeriod() |>
  mockMeasurement(recordPerson = 5)

# View the generated measurement data
print(cdm$measurement)
```

---

mockObservation   *Generates a mock observation table and integrates it into an existing CDM object.*

Description

This function simulates observation records for individuals within a specified cohort. It creates a realistic dataset by generating observation records based on the specified number of records per person. Each observation record is correctly associated with an individual within valid observation periods, ensuring the integrity of the data.

Usage

```
mockObservation(cdm, recordPerson = 1, seed = NULL)
```

Arguments

cdm   A 'cdm_reference' object that must already include 'person', 'observation_period',
                   and 'concept' tables. This object serves as the base CDM structure where the ob-
                   servation data will be added. The 'person' and 'observation_period' tables must
                   be populated as they are necessary for generating accurate observation records.

recordPerson   An integer specifying the expected number of observation records to generate
                   per person. This parameter allows for the simulation of varying frequencies of
                   healthcare observations among individuals in the cohort, reflecting real-world
                   variability in patient monitoring and health assessments.

seed   An optional integer used to set the seed for random number generation, ensuring
                   reproducibility of the generated data. If provided, this seed enables the function
                   to produce consistent results each time it is run with the same inputs. If 'NULL',
                   the seed is not set, which can lead to different outputs on each run.

Value

Returns the modified 'cdm' object with the new 'observation' table added. This table includes
the simulated observation data for each person, ensuring that each record is correctly linked to
individuals in the 'person' table and falls within valid observation periods.

Examples

```
library(omock)

# Create a mock CDM reference and add observation records
cdm <- mockCdmReference() |>
  mockPerson() |>
  mockObservationPeriod() |>
  mockObservation(recordPerson = 3)

# View the generated observation data
print(cdm$observation)
```

---

mockObservationPeriod   *Generates a mock observation period table and integrates it into an*
                        existing CDM object.

Description

This function simulates observation periods for individuals based on their date of birth recorded in
the 'person' table of the CDM object. It assigns random start and end dates for each observation
period within a realistic timeframe up to a specified or default maximum date.

Usage

```
mockObservationPeriod(cdm, seed = NULL)
```

Arguments

cdm	A 'cdm_reference' object that must include a 'person' table with valid dates of birth. This object serves as the base CDM structure where the observation period data will be added. The function checks to ensure that the 'person' table is populated and uses the date of birth to generate observation periods.
seed	An optional integer used to set the seed for random number generation, ensuring reproducibility of the generated data. If provided, this seed allows the function to produce consistent results each time it is run with the same inputs. If 'NULL', the seed is not set, which can lead to different outputs on each run.

Value

Returns the modified 'cdm' object with the new 'observation_period' table added. This table includes the simulated observation periods for each person, ensuring that each record spans a realistic timeframe based on the person's date of birth.

Examples

```
library(omock)

# Create a mock CDM reference and add observation periods
cdm <- mockCdmReference() |>
  mockPerson(nPerson = 100) |>
  mockObservationPeriod()

# View the generated observation period data
print(cdm$observation_period)
```

---

mockPerson	*Generates a mock person table and integrates it into an existing CDM object.*

Description

This function creates a mock person table with specified characteristics for each individual, including a randomly assigned date of birth within a given range and gender based on specified proportions. It populates the CDM object's person table with these entries, ensuring each record is uniquely identified.

Usage

```
mockPerson(
  cdm = mockCdmReference(),
  nPerson = 10,
  birthRange = as.Date(c("1950-01-01", "2000-12-31")),
  proportionFemale = 0.5,
  seed = NULL
)
```

Arguments

cdm
: A 'cdm_reference' object that serves as the base structure for adding the person table. This parameter should be an existing or newly created CDM object that does not yet contain a 'person' table.

nPerson
: An integer specifying the number of mock persons to create in the person table. This defines the scale of the simulation and allows for the creation of datasets with varying sizes.

birthRange
: A date range within which the birthdays of the mock persons will be randomly generated. This should be provided as a vector of two dates ('as.Date' format), specifying the start and end of the range.

proportionFemale
: A numeric value between 0 and 1 indicating the proportion of the persons who are female. For example, a value of 0.5 means approximately 50 the generated persons will be female. This helps simulate realistic demographic distributions.

seed
: An optional integer used to set the seed for random number generation, ensuring reproducibility of the generated data. If provided, this seed allows the function to produce consistent results each time it is run with the same inputs. If 'NULL', the seed is not set, which can lead to different outputs on each run.

Value

A modified 'cdm' object with the new 'person' table added. This table includes simulated person data for each generated individual, with unique identifiers and demographic attributes.

Examples

```
library(omock)
cdm <- mockPerson(cdm = mockCdmReference(), nPerson = 10)

# View the generated person data
print(cdm$person)
```

```
mockProcedureOccurrence
```
Generates a mock procedure occurrence table and integrates it into an existing CDM object.

Description

This function simulates condition occurrences for individuals within a specified cohort. It helps create a realistic dataset by generating condition records for each person, based on the number of records specified per person.The generated data are aligned with the existing observation periods to ensure that all conditions are recorded within valid observation windows.

Usage

```
mockProcedureOccurrence(cdm, recordPerson = 1, seed = NULL)
```

Arguments

cdm	A 'cdm_reference' object that should already include 'person', 'observation_period', and 'concept' tables.This object is the base CDM structure where the procedure occurrence data will be added. It is essential that these tables are not empty as they provide the necessary context for generating condition data.
recordPerson	An integer specifying the expected number of condition records to generate per person.This parameter allows the simulation of varying frequencies of condition occurrences among individuals in the cohort, reflecting the variability seen in real-world medical data.
seed	An optional integer used to set the seed for random number generation, ensuring reproducibility of the generated data.If provided, it allows the function to produce the same results each time it is run with the same inputs.If 'NULL', the seed is not set, resulting in different outputs on each run.

Value

Returns the modified 'cdm' object with the new 'condition_occurrence' table added. This table includes the simulated condition data for each person, ensuring that each record is within the valid observation periods and linked to the correct individuals in the 'person' table.

Examples

```
library(omock)

# Create a mock CDM reference and add condition occurrences
cdm <- mockCdmReference() |>
  mockPerson() |>
  mockObservationPeriod() |>
  mockProcedureOccurrence(recordPerson = 2)
```

```
# View the generated condition occurrence data
print(cdm$procedure_occurrence)
```

---

mockVisitOccurrence   *Function to generate visit occurrence table*

Description

Function to generate visit occurrence table

Usage

```
mockVisitOccurrence(cdm, seed = NULL)
```

Arguments

cdm   the CDM reference into which the mock visit occurrence table will be added

seed   A random seed to ensure reproducibility of the generated data.

Value

A cdm reference with the visit_occurrence tables added

Examples

```
library(omock)
```

---

mockVocabularyTables   *Creates a mock CDM database populated with various vocabulary tables.*

Description

This function adds specified vocabulary tables to a CDM object. It can either populate the tables with provided data frames or initialize empty tables if no data is provided. This is useful for setting up a testing environment with controlled vocabulary data.

Usage

```
mockVocabularyTables(
  cdm = mockCdmReference(),
  vocabularySet = "mock",
  cdmSource = NULL,
  concept = NULL,
  vocabulary = NULL,
  conceptRelationship = NULL,
  conceptSynonym = NULL,
  conceptAncestor = NULL,
  drugStrength = NULL
)
```

Arguments

cdm	A 'cdm_reference' object that serves as the base structure for adding vocabulary tables. This should be an existing or a newly created CDM object, typically initialized without any vocabulary tables.
vocabularySet	A character string that specifies a prefix or a set name used to initialize mock data tables. This allows for customization of the source data or structure names when generating vocabulary tables.
cdmSource	An optional data frame representing the CDM source table. If provided, it will be used directly; otherwise, a mock table will be generated based on the 'vocabularySet' prefix.
concept	An optional data frame representing the concept table. If provided, it will be used directly; if NULL, a mock table will be generated.
vocabulary	An optional data frame representing the vocabulary table. If provided, it will be used directly; if NULL, a mock table will be generated.
conceptRelationship	
	An optional data frame representing the concept relationship table. If provided, it will be used directly; if NULL, a mock table will be generated.
conceptSynonym	An optional data frame representing the concept synonym table. If provided, it will be used directly; if NULL, a mock table will be generated.
conceptAncestor	
	An optional data frame representing the concept ancestor table. If provided, it will be used directly; if NULL, a mock table will be generated.
drugStrength	An optional data frame representing the drug strength table. If provided, it will be used directly; if NULL, a mock table will be generated.

Value

Returns the modified 'cdm' object with the new or provided vocabulary tables added.

Examples

```
library(omock)
```

```
# Create a mock CDM reference and populate it with mock vocabulary tables
cdm <- mockCdmReference() |> mockVocabularyTables(vocabularySet = "mock")

# View the names of the tables added to the CDM
names(cdm)
```

Index