

Package: ImputeLongiCovs (via r-universe)

August 24, 2024

Type Package

Title Longitudinal Imputation of Categorical Variables via a Joint Transition Model

Version 0.1.0

Description Imputation of longitudinal categorical covariates. We use a methodological framework which ensures that the plausibility of transitions is preserved, overfitting and colinearity issues are resolved, and confounders can be utilized. See Mamouris (2023) <[doi:10.1002/sim.9919](https://doi.org/10.1002/sim.9919)> for an overview.

Depends R (>= 3.4.4)

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports nnet, stats

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Pavlos Mamouris [aut, cre], Vahid Nassiri [aut, ctb], Geert Molenberghs [ctb], Geert Verbeke [ctb]

Maintainer Pavlos Mamouris <pavlos.k.mamouris@gmail.com>

Date/Publication 2023-10-06 16:40:05 UTC

Repository <https://pmamouris.r-universe.dev>

RemoteUrl <https://github.com/cran/ImputeLongiCovs>

RemoteRef HEAD

RemoteSha be25a6c63e894f640eeeab00178d08674eaa8db8

Contents

analyses_data	2
create_probMatrix	3
impute_categorical_covariates	3
initial_data	5
Index	7

analyses_data	<i>Analyses Data for imputing categorical covariates</i>
---------------	--

Description

A dataset containing longitudinal data. The outcome of interest is the smoking status with three states (smoker, exsmoker, never smoker), which are represented via transitions. The difference from the initial data is the prob_matrix column.

Usage

```
data(analyses_data)
```

Format

A data frame with 2000 rows and 10 variables

Details

- patient_id: Unique identifier for each patient
- tran_Year: numeric, starting from 1 up to the number of transitions
- transition_year: text explanation of the transition
- state_from: the state at the beginning of a transition
- state_to: the state at the end of a transition
- prob_matrix: the probability matrix that was generated by the initial data
- cardio_state_from: cardiovascular disease at the beginning of the transition, binary, if 1 == Yes, else No
- cardio_state_to: cardiovascular disease at the end of the transition, binary, if 1 == Yes, else No
- flu_vaccination_state_from: flu vaccination at the end of the transition, binary, if 1 == Yes, else No
- flu_vaccination_state_to: flu vaccination disease at the end of the transition, binary, if 1 == Yes, else No

```
create_probMatrix      create_probMatrix
```

Description

create_probMatrix creates a variable that contains the transition probabilities ("initial", "forward", "backward", "intermittent", "observed")

Usage

```
create_probMatrix(input_data, patient_id)
```

Arguments

input_data	A dataset in a format similar to 'initial_data'. This dataset must contain the variables "state_from", which is the status at the beginning of the transition (say smoker in 2010), "state_to", which is the status at the end of the transition (say ex-smoker in 2011) and "tran_Year", which is an integer variable that is equal to the number of transitions. "tran_Year" == 1 means that the transition occurs from 2010 to 2011, "tran_Year" == 2, from 2011 to 2012, up to the total number of transitions
patient_id	A character variable that specifies the column name with the unique Id of the patient

Value

a data frame containing the column "prob_matrix"

Examples

```
create_probMatrix(initial_data, patient_id = "patient_id")
```

```
impute_categorical_covariates
      impute_categorical_covariates
```

Description

impute_categorical_covariates imputes longitudinal categorical covariates through a joint model that accommodates initial, forward, backward, and intermittent transitions.

Usage

```

impute_categorical_covariates(
  input_data,
  patient_id,
  number_of_transitions,
  covariates_initial = NULL,
  covariates_transition = NULL,
  missing_variable_levels,
  startingyear = NULL,
  without_trans_prob,
  m = 1
)

```

Arguments

<code>input_data</code>	A dataset in a format similar to 'analyses_data'. This dataset must contain the variables "state_from", which is the status at the beginning of the transition (say smoker in 2010), "state_to", which is the status at the end of the transition (say ex-smoker in 2011) and "tran_Year", which is an integer variable that is equal to the number of transitions. "tran_Year" == 1 means that the transition occurs from 2010 to 2011, "tran_Year" == 2, from 2011 to 2012, up to the total number of transitions Also, it must contain "prob_matrix" which captures all the transitions ("initial", "forward", "backward", "intermittent", "observed") that was calculated with the 'create_probMatrix' function
<code>patient_id</code>	A character variable that specifies the column name with the unique Id of the patient
<code>number_of_transitions</code>	The number of transitions needed. For example for years 2010, 2011 and 2012 there exist 2 transitions.
<code>covariates_initial</code>	The covariates to be used in the initial model
<code>covariates_transition</code>	The covariates to be used in the transition model
<code>missing_variable_levels</code>	The levels of the missing categorical outcome (e.g. "smoker", "ex-smoker", "never-smoker")
<code>startingyear</code>	If the starting year per patient has no missing values, specify it
<code>without_trans_prob</code>	This statement is useful when there are very high proportions of missing data and our initial and transition model cannot converge. It provides the user with two options. One, to "notImpute", namely to return NA and two, to "ImputeEqualProbabilities", i.e., the user can sample with equal probabilities.
<code>m</code>	Numeric, the number of imputed datasets

Details

It encloses three different functions. The 'initial_forward_function' imputes the longitudinal categorical covariate of interest based on whether in that transition the 'prob_matrix' of a patient was

'initial' or 'forward'. The 'imputeIntermittent' imputes the longitudinal categorical covariate for the intermittent transition and the 'backward_function' imputes the longitudinal categorical covariate for the backward transition.

Value

a list of m data frames with no missing values in the categorical outcome

References

()

Examples

```
impute_categorical_covariates(analyses_data,
  patient_id = "patient_id",
  number_of_transitions = 2,
  covariates_initial = c("cardio_state_from", "flu_vaccination_state_from"),
  covariates_transition = c("cardio_state_to", "flu_vaccination_state_to"),
  missing_variable_levels = c("never-smoker", "smoker", "ex-smoker"),
  startingyear = NULL,
  without_trans_prob = "notImpute",
  m = 2)
```

initial_data

Initial Data for imputing categorical covariates

Description

A dataset containing longitudinal data. The outcome of interest is the smoking status with three states (smoker, exsmoker, neversmoker), which are represented via transitions.

Usage

```
data(initial_data)
```

Format

A data frame with 2000 rows and 9 variables

Details

- patient_id: Unique identifier for each patient
- tran_Year: numeric, starting from 1 up to the number of transitions
- transition_year: text explanation of the transition
- state_from: the state at the beginning of a transition
- state_to: the state at the end of a transition

- `cardio_state_from`: cardiovascular disease at the beginning of the transition, binary, if 1 == Yes, else No
- `cardio_state_to`: cardiovascular disease at the end of the transition, binary, if 1 == Yes, else No
- `flu_vaccination_state_from`: flu vaccination at the end of the transition, binary, if 1 == Yes, else No
- `flu_vaccination_state_to`: flu vaccination disease at the end of the transition, binary, if 1 == Yes, else No

Index

* datasets

analyses_data, 2

initial_data, 5

analyses_data, 2

create_probMatrix, 3

impute_categorical_covariates, 3

initial_data, 5