

# Package ‘mantar’

July 22, 2025

**Title** Missingness Alleviation for Network Analysis

**Version** 0.1.0

**Description** Provides functionality for estimating cross-sectional network structures representing partial correlations in R, while accounting for missing values in the data. Networks are estimated via neighborhood selection, i.e., node-wise multiple regression, with model selection guided by information criteria. Missing data can be handled primarily via multiple imputation or a maximum likelihood-based approach; deletion techniques are available but secondary <[doi:10.31234/osf.io/qpj35](https://doi.org/10.31234/osf.io/qpj35)>.

**License** GPL (>= 3)

**Depends** R (>= 4.1.0)

**Imports** stats

**Suggests** mice, lavaan, qgraph, testthat (>= 3.0.0)

**LazyData** true

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Kai Jannik Nehler [aut, cre] (ORCID:  
<<https://orcid.org/0000-0003-3764-761X>>)

**Maintainer** Kai Jannik Nehler <nehler@psych.uni-frankfurt.de>

**Repository** CRAN

**Date/Publication** 2025-07-11 12:40:02 UTC

## Contents

mantar_dummy_full . . . . .	2
neighborhood_net . . . . .	3
regression_opt . . . . .	5
<b>Index</b>	<b>8</b>

---

`mantar_dummy_full`      *Dummy data sets for illustration purposes in the mantar package*

---

## Description

These two **simulated data sets** are provided for illustration purposes. They are based on a sparse psychological network structure with a single underlying construct. The column names represent core properties of neuroticism but are purely made up to make the example more illustrative.

- **mantar\_dummy\_full**: A complete data set without missing values.
- **mantar\_dummy\_mis**: A version with approximately 30% missing values per column.

## Usage

```
mantar_dummy_full
```

```
mantar_dummy_mis
```

## Format

**Both data frames** 8 columns; rows: 400 (`mantar_dummy_full`) and 600 (`mantar_dummy_mis`)

**Columns** **EmoReactivity** Tending to feel emotions strongly in response to life events.

**TendWorry** Being more likely to feel concerned or uneasy.

**StressSens** Feeling more stressed in challenging or uncertain situations.

**SelfAware** Being conscious of one's own feelings and how they shift.

**Moodiness** Experiencing occasional changes in mood.

**Cautious** Being careful and thinking ahead about possible negative outcomes.

**ThoughtFuture** Reflecting on what might go wrong and preparing for it.

**RespCriticism** Being affected by others' feedback or disapproval.

An object of class `data.frame` with 600 rows and 8 columns.

## Examples

```
# Load the data sets
data(mantar_dummy_full)
data(mantar_dummy_mis)

# View the first few rows of each data set
head(mantar_dummy_full)
head(mantar_dummy_mis)
```

---

neighborhood_net	<i>Estimate Network using Neighborhood Selection based on Information Criteria</i>
------------------	--

---

## Description

Estimate Network using Neighborhood Selection based on Information Criteria

## Usage

```
neighborhood_net(
  data = NULL,
  ns = NULL,
  mat = NULL,
  n_calc = "individual",
  missing_handling = "two-step-em",
  k = "log(n)",
  nimp = 20,
  pcor_merge_rule = "and"
)
```

## Arguments

data	Raw data containing only the variables to be included in the network. May include missing values.
ns	Numeric vector specifying the sample size for each variable in the data. If not provided, it will be computed based on the data. Must be provided if a correlation matrix ( <i>mat</i> ) is supplied instead of raw data.
mat	Optional covariance or correlation matrix for the variables to be included in the network. Used only if <i>data</i> is NULL.
n_calc	Method for calculating the sample size for node-wise regression models. Can be one of: "individual" (sample size for each variable is the number of non-missing observations for that variable), "average" (sample size is the average number of non-missing observations across all variables), "max" (sample size is the maximum number of non-missing observations across all variables), "total" (sample size is the total number of observations across in the data set / number of rows).
missing_handling	Method for estimating the correlation matrix in the presence of missing data. "two-step-em" uses a classic EM algorithm to estimate the covariance matrix from the data. "stacked-mi" uses multiple imputation to estimate the covariance matrix from the data. "pairwise" uses pairwise deletion to estimate the covariance matrix from the data. "listwise" uses listwise deletion to estimate the covariance matrix from the data.

<code>k</code>	Penalty per parameter (number of predictor + 1) to be used in node-wise regressions; the default " $\log(n)$ " (number of observations for the dependent variable) is the classical BIC. Alternatively, classical AIC would be $k = 2$ .
<code>nimp</code>	Number of multiple imputations to perform when using multiple imputation for missing data (default: 20).
<code>pcor_merge_rule</code>	Rule for merging regression weights into partial correlations. "and" estimates a partial correlation only if regression weights in both directions (e.g., from node 1 to 2 and from 2 to 1) are non-zero in the final models. "or" uses the available regression weight from one direction as partial correlation if the other is not included in the final model.

### Details

This function estimates a network structure using neighborhood selection guided by information criteria. Simulations by Williams et al. (2019) indicated that using the "and" rule for merging regression weights tends to yield more accurate partial correlation estimates than the "or" rule. Both the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) are supported and have been shown to produce valid network structures.

To handle missing data, the function offers two approaches: a two-step expectation-maximization (EM) algorithm and stacked multiple imputation. According to simulations by Nehler and Schultze (2024), stacked multiple imputation performs reliably across a range of sample sizes. In contrast, the two-step EM algorithm provides accurate results primarily when the sample size is large relative to the amount of missingness and network complexity—but may still be preferred in such cases due to its much faster runtime.

Currently, the function only supports variables that are directly included in the network analysis; auxiliary variables for missing handling are not yet supported. During imputation, all variables are imputed using predictive mean matching (see e.g., van Buuren, 2018), with all other variables in the data set used as predictors.

### Value

A list with the following elements:

**pcor** Partial correlation matrix estimated from the node-wise regressions.

**betas** Matrix of regression coefficients from the final regression models.

**ns** Sample sizes used for each variable in the node-wise regressions.

**args** List of arguments used in the function call, including `pcor_merge_rule`, `k`, `missing_handling`, and `nimp`.

### References

Nehler, K. J., & Schultze, M. (2024). *Handling missing values when using neighborhood selection for network analysis*. <https://doi.org/10.31234/osf.io/qpj35>

van Buuren, S. (2018). *Flexible Imputation of Missing Data* (2nd ed.). CRC Press.

Williams, D. R., Rhemtulla, M., Wysocki, A. C., & Rast, P. (2019). On nonregularized estimation of psychological networks. *Multivariate Behavioral Research*, *54*(5), 719–750. <https://doi.org/10.1080/00273171.2019.1575711>

**Examples**

```

# Estimate network from full data set
# Using Akaike information criterion
result <- neighborhood_net(data = mantar_dummy_full,
k = "2")

# View estimated partial correlations
result$pcor

# Estimate network for data set with missings
# Using Bayesian Information Criterion, individual sample sizes, and two-step EM
result_mis <- neighborhood_net(data = mantar_dummy_mis,
n_calc = "individual",
missing_handling = "two-step-em")

# View estimated partial correlations
result_mis$pcor

```

regression\_opt

*Stepwise Multiple Regression Search based on Information Criteria***Description**

Stepwise Multiple Regression Search based on Information Criteria

**Usage**

```

regression_opt(
  data = NULL,
  n = NULL,
  mat = NULL,
  dep_ind,
  n_calc = "individual",
  missing_handling = "stacked-mi",
  k = "log(n)",
  nimp = 20
)

```

**Arguments**

data	Raw data containing only the variables to be tested within the multiple regression as dependent or independent variable. May include missing values.
n	Numeric value specifying the sample size used in calculating information criteria for model search. If not provided, it will be computed based on the data. If a correlation matrix (mat) is supplied instead of raw data, n must be provided.
mat	Optional covariance or correlation matrix for the variables to be used within the multiple regression. #' Used only if data is NULL.

dep_ind	Index of the column within a data set to be used as dependent variable within in the regression model.
n_calc	Method for calculating the sample size for node-wise regression models. Can be one of: "individual" (sample size for each variable is the number of non-missing observations for that variable), "average" (sample size is the average number of non-missing observations across all variables), "max" (sample size is the maximum number of non-missing observations across all variables), "total" (sample size is the total number of observations across in the data set / number of rows).
missing_handling	Method for estimating the correlation matrix in the presence of missing data. "tow-step-em" uses a classic EM algorithm to estimate the covariance matrix from the data. "stacked-mi" uses multiple imputation to estimate the covariance matrix from the data. "pairwise" uses pairwise deletion to estimate the covariance matrix from the data. "listwise" uses listwise deletion to estimate the covariance matrix from the data.
k	Penalty per parameter (number of predictors + 1) to be used in node-wise regressions; the default $\log(n)$ (number of observations observation) is the classical BIC. Alternatively, classical AIC would be $k = 2$ .
nimp	Number of multiple imputations to perform when using multiple imputation for missing data (default: 20).

### Value

A list with the following elements:

**regression** Named vector of regression coefficients for the dependent variable.

**R2** R-squared value of the regression model.

**n** Sample size used in the regression model.

**args** List of arguments used in the regression model, including k, missing\_handling, and nimp.

### Examples

```
# For full data using AIC
# First variable of the data set as dependent variable
result <- regression_opt(
  data = mantar_dummy_full,
  dep_ind = 1,
  k = "2"
)

# View regression coefficients and R-squared
result$regression
result$R2

# For data with missingess using BIC
# Second variable of the data set as dependent variable
# Using individual sample size of the dependent variable and stacked Multiple Imputation
```

```
result_mis <- regression_opt(  
  data = mantar_dummy_mis,  
  dep_ind = 2,  
  n_calc = "individual",  
  missing_handling = "two-step-em",  
  )  
  
# View regression coefficients and R-squared  
result_mis$regression  
result_mis$R2
```

# Index

- \* **datasets**

- mantar\_dummy\_full, 2

- \* **data**

- mantar\_dummy\_full, 2

- \* **sets**

- mantar\_dummy\_full, 2

mantar\_dummy\_full, 2

mantar\_dummy\_mis (mantar\_dummy\_full), 2

neighborhood\_net, 3

regression\_opt, 5