# Package 'BifactorIndicesCalculator'

November 6, 2019

**Type** Package

**Title** Bifactor Indices Calculator

**Version** 0.1.0

**Maintainer** David Dueber <david.dueber@uky.edu>

**Description**

The calculator computes bifactor indices such as explained common variance (ECV), hierarchical Omega (OmegaH), percentage of uncontaminated correlations (PUC), item explained common variance (I-ECV), and more. This package is an R version of the 'Excel' based 'Bifactor Indices Calculator' (Dueber, 2017) <doi:10.13023/edp.tool.01> with added convenience features for directly utilizing output from several programs that can fit confirmatory factor analysis or item response models.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.5)

**Imports** tidyr, lavaan, mirt, MplusAutomation

**RoxygenNote** 6.1.1

**URL** https://github.com/ddueber/BifactorIndicesCalculator

**BugReports** https://github.com/ddueber/BifactorIndicesCalculator/issues

**Suggests** testthat (>= 2.1.0)

**NeedsCompilation** no

**Author** David Dueber [aut, cre]

**Repository** CRAN

**Date/Publication** 2019-11-06 16:20:07 UTC

## R topics documented:

---

ARPB *ARPB*

---

## Description

ARPB computes absolute relative bias in factor loadings between the general factor of a bifactor model and a unidimensional model.

## Usage

```
ARPB(Lambda, UniLambda)
```

## Arguments

| | |
|---|---|
| Lambda | is a matrix of factor loadings |
| UniLambda | is a matrix of factor loadings |

## Details

ARPB is called by bifactorIndices and bifactorIndicesMplus, which are the only functions in this package intended for casual users

## Value

a list where the first element is the average absolute relative paramter bias, and the second element is a vector of absolute relative bias by item

## See Also

bifactorIndices, bifactorIndicesMplus

## Examples

```
Lambda <- matrix(c(.82, .10,   0,   0,
                   .77, .35,   0,   0,
                   .79, .32,   0,   0,
                   .66, .39,   0,   0,
                   .51,   0, .71,   0,
                   .56,   0, .43,   0,
                   .68,   0, .13,   0,
                   .60,   0, .50,   0,
                   .83,   0,   0, .47,
                   .60,   0,   0, .27,
                   .78,   0,   0, .28,
                   .55,   0,   0, .75),
                 ncol = 4, byrow = TRUE)
colnames(Lambda) <- c("General", "PS", "HA", "SA")
UniLambda <- c(.78, .84, .82, .77, .69, .62, .69, .66, .82, .56, .74, .65)
ARPB(Lambda, UniLambda)
```

---

`bifactorIndices` *bifactorIndices*

---

## Description

Computes all available bifactor indices for the input given.

## Usage

```
bifactorIndices(Lambda, Theta = NULL, UniLambda = NULL,
  standardized = TRUE)
```

## Arguments

| | |
|---|---|
| `Lambda` | is a matrix of factor loadings or an object that can be converted to a matrix of factor loadings by `getLambda`. Currently fitted **lavaan** objects and fitted **mirt** objects are supported in addition to raw factor loading matrix input. For `Mplus` output files, use `bifactorIndicesMplus`. |
| `Theta` | is a vector of residual variances. If omitted, `Theta` will be computed from input for `Lambda`. |
| `UniLambda` | is a matrix of factor loadings or an object that can be converted to a matrix of factor loadings such as a fitted **lavaan** objects or fitted **mirt** object. Defaults to `NULL`, as `UniLambda` is only required if you wish to compute `ARPB` |
| `standardized` | lets the function know whether to look for standardized or unstandardized results from **lavaan**. If `Lambda` is not a **lavaan** object, then `standardized` will be ignored. |

## Details

Currently, factor loading matrices, fitted **lavaan** objects, and fitted **mirt** objects are supported. For `Mplus` output, see `bifactorIndicesMplus`. IRT parameters from **mirt** are converted to standardized factor loadings via the correspondence described in Kamata & Bauer (2008). If you wish to use standardized coefficients, item error variance will be computed directly from standardized factor loadings. `ARPB` will only be computed if the factor loadings from a unidimensional model are included, while `ECV_GS` and `ECV_SG` will only be computed for models with a general factor, and `PUC` will only be conputed for a true bifactor model. Note that if a correlated traits model is provided, the omega indices will simply be the regular omega values for those factors. Interpretations for individual indices as well as details about their computation can be found in the man page for the individual indices.

## Value

A list of bifactor indices, including three different ECV indices, IECV, PUC, Omega, OmegaH, and ARPB. Please note that many of these indices are interpretable even when the model being used is not a bifactor model; some indices may be useful for two-tier, trifactor, correlated traits, and even unidimensional models.

## References

Kamata, A., & Bauer, D. J. (2008). A note on the relation between factor analytic and item response theory models. *Structural Equation Modeling: A Multidisciplinary Journal, 15* (1), 136-153.

## See Also

`bifactorIndicesMplus`, `ECV_SS`, `ECV_SG`, `ECV_GS`, `IECV`, `PUC`, `Omega_S`, `Omega_H`, `ARPB`

## Examples

```
# Computing bifactor indices from fitted lavaan object
# (using mirt object is similar)
HS_model_bifactor <- "visual  =~ x1 + x2 + x3
                      textual =~ x4 + x5 + x6
                      speed   =~ x7 + x8 + x9
                      general =~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9"
# lavaan cannot find a good solution, but that's ok since this is just for illustration
fit <- suppressWarnings(lavaan::cfa(HS_model_bifactor,
                  data = lavaan::HolzingerSwineford1939,
                  orthogonal = TRUE))
bifactorIndices(fit)


# Computing bifactor indices from factor loading matrices
Lambda <-  matrix(c(.82, .10,   0,   0,
                    .77, .35,   0,   0,
                    .79, .32,   0,   0,
                    .66, .39,   0,   0,
                    .51,   0, .71,   0,
                    .56,   0, .43,   0,
```

```
                    .68,   0, .13,   0,
                    .60,   0, .50,   0,
                    .83,   0,   0, .47,
                    .60,   0,   0, .27,
                    .78,   0,   0, .28,
                    .55,   0,   0, .75),
                    ncol = 4, byrow = TRUE)
colnames(Lambda) <- c("General", "PS", "HA", "SA")
UniLambda <- c(.78, .84, .82, .77, .69, .62, .69, .66, .82, .56, .74, .65)
bifactorIndices(Lambda, UniLambda = UniLambda)
```

---

`bifactorIndicesMplus` *bifactorIndicesMplus*

---

## Description

Computes all available bifactor indices given an `Mplus` .out file for a bifactor model

## Usage

```
bifactorIndicesMplus(Lambda = file.choose(), UniLambda = NULL,
  standardized = TRUE)
```

## Arguments

| | |
|---|---|
| `Lambda` | is an Mplus .out file. Defaults to an open file dialog box |
| `UniLambda` | is an object that the function can convert to a matrix of factor loadings. The expected behavior is to store an Mplus output file as a variable and pass that variable as `UniLambda`. Defaults to `NULL`, as `UniLambda` is only required if you wish to compute `ARPB` |
| `standardized` | lets the function know whether it should be looking in the unstandardized results or the STDYX results from the Mplus output. |

## Details

ARPB will only be compute if the factor loadings from a unidimensional model (as a vector or as the result of using `readModels` on an `Mplus` .out file) are included. Note that if a correlated traits model is provided, the omega indices will simply be the regular omega values for those factors. Interpretations for individual indices as well as details about their computation can be found in the man page for the individual indices.

## Value

A list of bifactor indices, including three different ECV indices, IECV, PUC, Omega, OmegaH, and ARPB. Please note that many of these indices are interpretable even when the model being used is not a bifactor model; some indices may be useful for two-tier, trifactor, correlated traits, and even unidimensional models.

## See Also

bifactorIndices, ECV_SS, ECV_SG, ECV_GS, IECV, PUC, Omega_S, Omega_H, ARPB

---

ECV_GS *ECV_GS*

---

## Description

Computes an ECV index for all factors which can be interpreted as the proportion of common variance of the items in each specific factor which is due to the general factor; ECV_GS should be read 'ECV of the general factor with respect to a specific factor.' Here, ECV is computed only with respect to the items of a specific factor using the general factor loadings in the numerator; Stucky and Edelen (2015, p. 201) refer to this index as the 'within-domain ECV' for the specific factor. In the Excel version of the bifactor indices calculator (Dueber, 2017), this index is not computed.

## Usage

```
ECV_GS(Lambda)
```

## Arguments

| | |
|---|---|
| Lambda | is a matrix of factor loadings. Be sure that all factors have the same variance before calling this function. |

## Details

ECV_GS is called by bifactorIndices and bifactorIndicesMplus, which are the only functions in this package intended for casual users

## Value

A vector of ECVs for all factors

## References

Dueber, D. M. (2017). Bifactor Indices Calculator: A Microsoft Excel-based tool to calculate various indices relevant to bifactor CFA models. doi: 10.13023/edp.tool.01

Stucky, B. D., & Edelen, M. O. (2015). Using hierarchical IRT models to create unidimensional measures from multidimensional data. In S. P. Reise & D. A. Revicki (Eds.), *Handbook of item response theory modeling: Applications to typical performance assessment* (pp.183-206). New York: Routledge.

## See Also

ECV_SS, ECV_SG, bifactorIndices, bifactorIndicesMplus

## Examples

```
Lambda <- matrix(c(.82, .10,   0,   0,
                   .77, .35,   0,   0,
                   .79, .32,   0,   0,
                   .66, .39,   0,   0,
                   .51,   0, .71,   0,
                   .56,   0, .43,   0,
                   .68,   0, .13,   0,
                   .60,   0, .50,   0,
                   .83,   0,   0, .47,
                   .60,   0,   0, .27,
                   .78,   0,   0, .28,
                   .55,   0,   0, .75),
                 ncol = 4, byrow = TRUE)
colnames(Lambda) <- c("General", "PS", "HA", "SA")
ECV_GS(Lambda)
```

---

ECV_SG *ECV_SG*

---

## Description

Computes an ECV index for all factors which can be interpreted as the proportion of common variance of all items which is due to the specific factor; ECV_SG should be read 'ECV of a specific factor with respect to the general factor.' Here, ECV is computed with respect to the items of the general factor using the specific factor loadings in the numerator; Stucky and Edelen (2015, p. 199) refer to this index simply as 'specific-dimension ECV.' Note that ECV_SG of the general factor is simply the ECV. In the Excel version of the Bifactor Indices Calculator (Dueber, 2017), this form of ECV is not referred to as 'ECV (S&E).' ECV_SG is called by bifactorIndices and bifactorIndicesMplus, which are the only functions in this package intended for casual users

## Usage

```
ECV_SG(Lambda)
```

## Arguments

| | |
|---|---|
| Lambda | is a matrix of factor loadings. Be sure that all factors have the same variance before calling this function. |

## Value

A vector of ECVs for all factors

## References

Dueber, D. M. (2017). Bifactor Indices Calculator: A Microsoft Excel-based tool to calculate various indices relevant to bifactor CFA models. doi: 10.13023/edp.tool.01

Stucky, B. D., & Edelen, M. O. (2015). Using hierarchical IRT models to create unidimensional measures from multidimensional data. In S. P. Reise & D. A. Revicki (Eds.), *Handbook of item response theory modeling: Applications to typical performance assessment* (pp.183-206). New York: Routledge.

## See Also

ECV_SS, ECV_SG, bifactorIndices, bifactorIndicesMplus

## Examples

```
Lambda <- matrix(c(.82, .10,   0,   0,
                   .77, .35,   0,   0,
                   .79, .32,   0,   0,
                   .66, .39,   0,   0,
                   .51,   0, .71,   0,
                   .56,   0, .43,   0,
                   .68,   0, .13,   0,
                   .60,   0, .50,   0,
                   .83,   0,   0, .47,
                   .60,   0,   0, .27,
                   .78,   0,   0, .28,
                   .55,   0,   0, .75),
                   ncol = 4, byrow = TRUE)
colnames(Lambda) <- c("General", "PS", "HA", "SA")
ECV_GS(Lambda)
```

---

ECV_SS *ECV_SS*

---

## Description

Computes an ECV index for all factors which can be interpreted as the proportion of common variance of the items in each factor which is due to that factor; ECV_SS should be read 'ECV of a specific factor with respect to itself.' Here, ECV is computed only with respect to items which load on the factor. Note that ECV_SS of the general factor is simply the ECV. Stucky and Edelen (2015, p. 201) do not refer to this form of ECV. In the Excel version of the bifactor indices calculator (Dueber, 2017), this index is referred to as 'ECV (NEW).' ECV_SS is useful in that it can be computed when there is no general factor, such as in a two-tier model.

## Usage

```
ECV_SS(Lambda)
```

## Arguments

| | |
|---|---|
| Lambda | is a matrix of factor loadings. Be sure that all factors have the same variance before calling this function. |

## Details

ECV_SS is called by bifactorIndices and bifactorIndicesMplus, which are the only functions in this package intended for casual users

## Value

A vector of ECVs for all factors

## References

Dueber, D. M. (2017). Bifactor Indices Calculator: A Microsoft Excel-based tool to calculate various indices relevant to bifactor CFA models. doi: 10.13023/edp.tool.01

Stucky, B. D., & Edelen, M. O. (2015). Using hierarchical IRT models to create unidimensional measures from multidimensional data. In S. P. Reise & D. A. Revicki (Eds.), *Handbook of item response theory modeling: Applications to typical performance assessment* (pp.183-206). New York: Routledge.

## See Also

ECV_SG, ECV_GS, bifactorIndices, bifactorIndicesMplus

## Examples

```
Lambda <- matrix(c(.82, .10,   0,   0,
                   .77, .35,   0,   0,
                   .79, .32,   0,   0,
                   .66, .39,   0,   0,
                   .51,   0, .71,   0,
                   .56,   0, .43,   0,
                   .68,   0, .13,   0,
                   .60,   0, .50,   0,
                   .83,   0,   0, .47,
                   .60,   0,   0, .27,
                   .78,   0,   0, .28,
                   .55,   0,   0, .75),
                   ncol = 4, byrow = TRUE)
colnames(Lambda) <- c("General", "PS", "HA", "SA")
ECV_SS(Lambda)
```

## getGen *getGen*

### Description

getGen detects whether or not a single factor loads on all items, and returns the column index of the general factor if it exists.

### Usage

```
getGen(Lambda)
```

### Arguments

| | |
|---|---|
| Lambda | is a factor loading matrix |

### Value

The index of the general factor, or NULL if there is no general factor

## getLambda *getLambda*

### Description

getLambda computes or extracts a matrix of factor loadings given some input. Methods exist to support an input of a dataframe, an mplus.model from **MplusAutomation**, a SingleGroupClass object from **mirt**, and a lavaan object from **lavaan**. Please do not use a tibble, as they do not support row names, and it is best if your items are given names.

### Usage

```
getLambda(x, standardized = TRUE)
```

### Arguments

| | |
|---|---|
| x | an object to be converted into a factor loading matrix, or an object containing a fitted model from which a factor loading matrix will be extracted. Supported classes are data.frame, matrix, mplus.model, lavaan, and SingleGroupClass. |
| standardized | can be used to specify whether a standardized or unstandardized factor loading matrix should be returned. Only relevant for lavaan and mplus.model input. The standardized matrix for mplus.model is taken from stdyx results. |

### Value

A matrix of factor loadings

getTheta *getTheta*

## Description

getTheta extracts or computes a vector of residual variance for items. If a factor loading matrix is provided, then the vector of residual variances is computed from that matrix if standardized is TRUE.

## Usage

```
getTheta(x, standardized = TRUE)
```

## Arguments

| | |
|---|---|
| x | an object that can be converted into a factor loading matrix, or an object containing a fitted model from which a vector of residual variances can be extracted. Supported classes are data.frame, matrix, mplus.model, lavaan, and SingleGroupClass |
| standardized | can be used to specify whether a standardized or unstandardized factor loading matrix should be returned. Only relevant for lavaan and mplus.model input. The standardized matrix for mplus.model is taken from stdyx results. |

## Value

a vector of residual variances for items. If x is a fitted model, then the residual variances are extracted from the fitted model. **lavaan**, **mirt** (SingleGroupClass), and Mplus (mplus.model) models are supported. If Mplus does not report residual variances for categorical variables, then factor loadings are used to compute the residual variance for standardized models and an error is thrown for unstandardized models. In both cases, the user is alerted that residual variances could not be found in the input and perhaps the model should be rerun.

## See Also

getLambda

IECV *IECV*

## Description

Computes an ECV index for each item which can be interpreted as the proportion of common variance of that item due to the general factor. Stucky and Edelen (2015, p. 201) define I-ECV, which is also computed in the Excel version of the bifactor indices calculator (Dueber, 2017).

## Usage

```
IECV(Lambda)
```

## Arguments

| | |
|---|---|
| Lambda | is a matrix of factor loadings. Be sure that all factors have the same variance before calling this function. |

## Details

IECV is called by bifactorIndices and bifactorIndicesMplus, which are the only functions in this package intended for casual users

## Value

A vector of item ECVs

## References

Dueber, D. M. (2017). Bifactor Indices Calculator: A Microsoft Excel-based tool to calculate various indices relevant to bifactor CFA models. doi: 10.13023/edp.tool.01

Stucky, B. D., & Edelen, M. O. (2015). Using hierarchical IRT models to create unidimensional measures from multidimensional data. In S. P. Reise & D. A. Revicki (Eds.), *Handbook of item response theory modeling: Applications to typical performance assessment* (pp.183-206). New York: Routledge.

## See Also

ECV_SS, ECV_SG, bifactorIndices, bifactorIndicesMplus

## Examples

```
Lambda <- matrix(c(.82, .10,   0,   0,
                   .77, .35,   0,   0,
                   .79, .32,   0,   0,
                   .66, .39,   0,   0,
                   .51,   0, .71,   0,
                   .56,   0, .43,   0,
                   .68,   0, .13,   0,
                   .60,   0, .50,   0,
                   .83,   0,   0, .47,
                   .60,   0,   0, .27,
                   .78,   0,   0, .28,
                   .55,   0,   0, .75),
                   ncol = 4, byrow = TRUE)
colnames(Lambda) <- c("General", "PS", "HA", "SA")
IECV(Lambda)
```

`isBifactor` *isBifactor*

## Description

Determines whether a model has bifactor structure.

## Usage

```
isBifactor(Lambda)
```

## Arguments

| | |
|---|---|
| `Lambda` | Matrix of factor loadings |

## Value

Logical. If each item loads on a general factor and at most one specific factor, returns TRUE. Otherwise FALSE.

`Omega_H` *Omega_H*

## Description

Computes hierarchical omega reliability estimate for all factors as described in Rodriguez, Reise, and Haviland (2016).

## Usage

```
Omega_H(Lambda, Theta)
```

## Arguments

| | |
|---|---|
| `Lambda` | is a matrix of factor loadings |
| `Theta` | is a vector of indicator error variances |

## Details

`Omega_H` is called by `bifactorIndices` and `bifactorIndicesMplus`, which are the only functions in this package intended for casual users

## Value

A `numeric`, the omega reliability estimate for all factors.

## References

Rodriguez, A., Reise, S. P., & Haviland, M. G. (2016). Evaluating bifactor models: Calculating and interpreting statistical indices. Psychological methods, 21(2), 137 doi: 10.1037/met0000045.

## See Also

Omega_S, bifactorIndices, bifactorIndicesMplus

## Examples

```
Lambda <- matrix(c(.82, .10,   0,   0,
                   .77, .35,   0,   0,
                   .79, .32,   0,   0,
                   .66, .39,   0,   0,
                   .51,   0, .71,   0,
                   .56,   0, .43,   0,
                   .68,   0, .13,   0,
                   .60,   0, .50,   0,
                   .83,   0,   0, .47,
                   .60,   0,   0, .27,
                   .78,   0,   0, .28,
                   .55,   0,   0, .75),
                  ncol = 4, byrow = TRUE)
colnames(Lambda) <- c("General", "PS", "HA", "SA")
Theta <- rep(1, nrow(Lambda)) - rowSums(Lambda^2)
Omega_H(Lambda, Theta)
```

---

Omega_S *Omega_S*

---

## Description

Computes an omega reliability estimate for all factors as described in Rodriguez, Reise, and Haviland (2016).

## Usage

```
Omega_S(Lambda, Theta)
```

## Arguments

| | |
|---|---|
| Lambda | is a matrix of factor loadings |
| Theta | is a vector of indicator error variances |

## Details

Omega_S is called by bifactorIndices and bifactorIndicesMplus, which are the only functions in this package intended for casual users

## Value

A numeric, the omega reliability estimate for all factors.

## References

Rodriguez, A., Reise, S. P., & Haviland, M. G. (2016). Evaluating bifactor models: calculating and interpreting statistical indices. *Psychological Methods, 21*(2), 137 doi: 10.1037/met0000045.

## See Also

Omega_H, bifactorIndices, bifactorIndicesMplus

## Examples

```
Lambda <- matrix(c(.82, .10,   0,   0,
                   .77, .35,   0,   0,
                   .79, .32,   0,   0,
                   .66, .39,   0,   0,
                   .51,   0, .71,   0,
                   .56,   0, .43,   0,
                   .68,   0, .13,   0,
                   .60,   0, .50,   0,
                   .83,   0,   0, .47,
                   .60,   0,   0, .27,
                   .78,   0,   0, .28,
                   .55,   0,   0, .75),
                  ncol = 4, byrow = TRUE)
colnames(Lambda) <- c("General", "PS", "HA", "SA")
Theta <- rep(1, nrow(Lambda)) - rowSums(Lambda^2)
Omega_S(Lambda, Theta)
```

---

PUC *PUC*

---

## Description

PUC computes the proportion of uncontaminated correlations for a bifactor mode

## Usage

```
PUC(Lambda)
```

## Arguments

| | |
|---|---|
| Lambda | is a matrix of factor loadings |

## Details

PUC is called by bifactorIndices and bifactorIndicesMplus, which are the only functions in this package intended for casual users

## Value

numeric

## See Also

bifactorIndices, bifactorIndicesMplus

## Examples

```
Lambda <- matrix(c(.82, .10,   0,   0,
                   .77, .35,   0,   0,
                   .79, .32,   0,   0,
                   .66, .39,   0,   0,
                   .51,   0, .71,   0,
                   .56,   0, .43,   0,
                   .68,   0, .13,   0,
                   .60,   0, .50,   0,
                   .83,   0,   0, .47,
                   .60,   0,   0, .27,
                   .78,   0,   0, .28,
                   .55,   0,   0, .75),
                 ncol = 4, byrow = TRUE)
colnames(Lambda) <- c("General", "PS", "HA", "SA")
PUC(Lambda)
```

# Index