

# Package ‘kstMatrix’

October 12, 2023

**Type** Package

**Version** 0.2-0

**Date** 2023-10-12

**Title** Basic Functions in Knowledge Space Theory Using Matrix Representation

**Description** Knowledge space theory by Doignon and Falmagne (1999) [doi:10.1007/978-3-642-58625-5](https://doi.org/10.1007/978-3-642-58625-5) is a set- and order-theoretical framework, which proposes mathematical formalisms to operationalize knowledge structures in a particular domain. The 'kstMatrix' package provides basic functionalities to generate, handle, and manipulate knowledge structures and knowledge spaces. Opposed to the 'kst' package, 'kstMatrix' uses matrix representations for knowledge structures. Furthermore, 'kstMatrix' contains several knowledge spaces developed by the research group around Cornelia Dowling through querying experts.

**Depends** R (>= 3.5.0)

**Suggests** knitr, rmarkdown

**Imports** stats, igraph, grDevices, sets, pks

**Maintainer** Cord Hockemeyer <cord.hockemeyer@uni-graz.at>

**License** GPL-3

**NeedsCompilation** no

**Repository** CRAN

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

**VignetteBuilder** knitr

**Author** Cord Hockemeyer [aut, cre],  
Wai Wong [ctb]

**Date/Publication** 2023-10-12 15:30:02 UTC

## R topics documented:

cad . . . . .	2
fractions . . . . .	3
kmbasis . . . . .	4
kmbasisdiagram . . . . .	5
kmcolors . . . . .	5
kmdist . . . . .	6
kmeqreduction . . . . .	6
kmfringe . . . . .	7
kmhasse . . . . .	8
kmiswellgraded . . . . .	8
kmneighbourhood . . . . .	9
kmnneighbourhood . . . . .	10
kmnotions . . . . .	10
kmsf2basis . . . . .	11
kmsimulate . . . . .	11
kmsurmisefunction . . . . .	12
kmsurmiserelement . . . . .	13
kmsymmsetdiff . . . . .	13
kmtrivial . . . . .	14
kmunionclosure . . . . .	15
kmvalidate . . . . .	15
readwrite . . . . .	16
xpl . . . . .	17
<b>Index</b>	<b>18</b>

---

cad

*Knowledge spaces on AutoCAD knowledge*

---

### Description

Bases of knowledge spaces on AutoCAD knowledge obtained from querying experts.

### Usage

cad

### Format

A list containing seven bases (cad1 to cad6, and cadmaj) in binary matrix form. Each matrix has 28 columns representing the different knowledge items and a varying number of rows containing the basis elements.

**Details**

Six experts were queried about prerequisite relationships between 28 AutoCAD knowledge items (Dowling, 1991; 1993). A seventh basis represents those prerequisite relationships on which the majority (4 out of 6) of the experts agree (Dowling & Hockemeyer, 1998).

**References**

Dowling, C. E. (1991). Constructing Knowledge Structures from the Judgements of Experts. Habilitationsschrift, Technische Universität Carolo-Wilhelmina, Braunschweig, Germany.

Dowling, C. E. (1993). Applying the basis of a knowledge space for controlling the questioning of an expert. *Journal of Mathematical Psychology*, 37, 21–48.

Dowling, C. E. & Hockemeyer, C. (1998). Computing the intersection of knowledge spaces using only their basis. In Cornelia E. Dowling, Fred S. Roberts, & Peter Theuns, editors, *Recent Progress in Mathematical Psychology*, pp. 133–141. Lawrence Erlbaum Associates Ltd., Mahwah, NJ.

**See Also**

Other Data: [fractions](#), [readwrite](#), [xpl](#)

---

 fractions

---

*Knowledge spaces on fractions*


---

**Description**

Bases of knowledge spaces on fractions obtained from querying experts.

**Usage**

fractions

**Format**

A list containing four bases (frac1 to frac3, and fracmaj) in binary matrix form. Each matrix has 77 columns representing the different knowledge items and a varying number of rows containing the basis elements.

**Details**

Three experts were queried about prerequisite relationships between 77 items on fractions (Baumunk & Dowling, 1997). A fourth basis represents those prerequisite relationships on which the majority of the experts agree (Dowling & Hockemeyer, 1998).

## References

Baumunk, K. & Dowling, C. E. (1997). Validity of spaces for assessing knowledge about fractions. *Journal of Mathematical Psychology*, 41, 99–105.

Dowling, C. E. & Hockemeyer, C. (1998). Computing the intersection of knowledge spaces using only their basis. In Cornelia E. Dowling, Fred S. Roberts, & Peter Theuns, editors, *Recent Progress in Mathematical Psychology*, pp. 133–141. Lawrence Erlbaum Associates Ltd., Mahwah, NJ.

## See Also

Other Data: [cad](#), [readwrite](#), [xpl](#)

---

kmbasis

*Compute the basis of a knowledge space*

---

## Description

kmbasis returns a matrix representing the basis of a knowledge space. If  $x$  is a knowledge structure or an arbitrary family of sets kmbasis returns the basis of the smallest knowledge space containing  $x$ .

## Usage

```
kmbasis(x)
```

## Arguments

$x$  Binary matrix representing a knowledge space

## Value

Binary matrix representing the basis of the knowledge space.

## See Also

Other Different representations for knowledge spaces: [kmsurmisefunction\(\)](#), [kmsurmiserelation\(\)](#), [kmunionclosure\(\)](#)

## Examples

```
kmbasis(xpl$space)
```

---

kmbasisdiagram	<i>Plot the Hasse diagram of a basis stored as a matrix</i>
----------------	---

---

### Description

kmbasisdiagram takes a matrix representing a basis and a color vector and draws a Hasse diagram. If the color vector is NULL the states are drawn in green.

### Usage

```
kmbasisdiagram(struc, horizontal = FALSE, colors = NULL)
```

### Arguments

struc	Binary matrix representing a basis
horizontal	Boolean defining orientation of the graph, default FALSE
colors	Color vector (default NULL)

### See Also

Other Plotting knowledge structures: [kmhasse\(\)](#)

---

kmcolors	<i>Determine a color vector based on probabilities</i>
----------	--

---

### Description

kmcolors takes a probability vector and a color palette and creates a color vector to be used with kmhasse.

### Usage

```
kmcolors(prob, palette = cm.colors)
```

### Arguments

prob	Probability vector
palette	Color palette (default = cm.colors)

kmdist *Compute the distance between a data set and a knowledge structure*

---

**Description**

kmdist returns a named vector with the frequencies of distances between a set of response patterns and a knowledge structure. This vector can be used to compute, e.g., the Discrepancy Index (DI) or the Distance Agreement Coefficient (DA).

**Usage**

```
kmdist(data, struct)
```

**Arguments**

data	Binary matrix representing a set of response patterns
struct	Binary matrix representing a knowledge structure

**Value**

Distance distribution vector

**See Also**

Other Validating knowledge spaces: [kmvalidate\(\)](#)

**Examples**

```
kmdist(xpl$data, xpl$space)
```

---

kmeqreduction *Reduce a family of knowledge states with respect to item equivalence*

---

**Description**

kmeqreduction takes a family of knowledge states and returns its reduction to non-equivalent items.

**Usage**

```
kmeqreduction(x)
```

**Arguments**

x	Binary matrix
---	---------------

**Value**

Binary matrix reduced by equivalences

**See Also**

Other Properties of knowledge structures: [kmiswellgraded\(\)](#), [kmnotions\(\)](#)

**Examples**

```
kmereduction(xpl$space)
```

---

kmfringe	<i>Compute the fringe of a state within a knowledge structure</i>
----------	---

---

**Description**

kmfringe computes the fringe of a state within a knowledge structure, i.e. the set of items by which the state differs from its neighbours.

**Usage**

```
kmfringe(state, struct)
```

**Arguments**

state	Binary vector representing a knowledge state
struct	Binary matrix representing a knowledge structure

**Value**

Binary vector representing the fringe

**See Also**

Other Neighbourhood & fringe: [kmneighbourhood\(\)](#)

**Examples**

```
kmfringe(c(1,0,0,0), xpl$space)
```

---

`kmhasse`*Plot the Hasse diagram of a knowledge structure stored as a matrix*

---

**Description**

`kmhasse` takes a matrix representing a knowledge structure and a color vector and draws a Hasse diagram. If the color vector is `NULL` the states are drawn in green.

**Usage**

```
kmhasse(struc, horizontal = FALSE, colors = NULL)
```

**Arguments**

<code>struc</code>	Binary matrix representing a knowledge structure
<code>horizontal</code>	Boolean defining orientation of the graph, default <code>FALSE</code>
<code>colors</code>	Color vector (default <code>NULL</code> )

**See Also**

Other Plotting knowledge structures: [kmbasisdiagram\(\)](#)

---

`kmiswellgraded`*Check for wellgradedness of a knowledge structure*

---

**Description**

`kmiswellgraded` returns whether a knowledge structure is wellgraded.

**Usage**

```
kmiswellgraded(x)
```

**Arguments**

<code>x</code>	Binary matrix representing a knowledge space
----------------	--

**Value**

Logical value specifying whether ‘`x`’ is wellgraded

**References**

Doignon, J.-P. & Falmagne, J.-C. (1999). Knowledge Spaces. Springer–Verlag, Berlin.



**See Also**

Other Properties of knowledge structures: [kmeqreduction\(\)](#), [kmnotions\(\)](#)

**Examples**

```
kmiswellgraded(xpl$space)
```

---

kmneighbourhood	<i>Compute the neighbourhood of a state within a knowledge structure</i>
-----------------	--

---

**Description**

kmneighbourhood computes the neighbourhood of a state within a knowledge structure, i.e. the family of all other states with a symmetric set difference of 1.

**Usage**

```
kmneighbourhood(state, struct)
```

**Arguments**

state	Binary vector representing a knowledge state
struct	Binary matrix representing a knowledge structure

**Value**

Matrix containing the neighbouring states, one per row

**See Also**

Other Neighbourhood & fringe: [kmfringe\(\)](#)

**Examples**

```
kmneighbourhood(c(1,1,0,0), xpl$space)
```

---

kmneighbourhood	<i>Compute the n-neighbourhod of a state within a knowledge structure</i>
-----------------	---

---

**Description**

kmneighbourhood computes the n-neighbourhood of a state within a knowledge structure, i.e. the family of all other states with a symmetric set difference maximal n.

**Usage**

```
kmneighbourhood(state, struct, distance)
```

**Arguments**

state	Binary vector representing a knowledge state
struct	Binary matrix representing a knowledge structure
distance	Size of the n-neighbourhood

**Value**

Matrix containing the neighbouring states, one per row

**Examples**

```
kmneighbourhood(c(1,1,0,0), xpl$space, 2)
```

---

kmnotions	<i>Determine the notions of a knowledge structure</i>
-----------	---

---

**Description**

kmnotions returns a matrix representing the notions of a knowledge structure.

**Usage**

```
kmnotions(x)
```

**Arguments**

x	Binary matrix representing a knowledge structure
---	--

**Value**

Binary matrix representing notions in the knowledge structure

The matrix has a '1' in row 'i' and column 'j' if 'i' and 'j' belong to the same notion (i.e. are equivalent). It is a symmetric matrix with '1's in the main diagonal.

**See Also**

Other Properties of knowledge structures: [kmeqreduction\(\)](#), [kmiswellgraded\(\)](#)

**Examples**

```
kmnotions(xpl$space)
```

---

kmsf2basis	<i>Derive a basis from a surmise function</i>
------------	---

---

**Description**

kmsf2basis expects a surmise function data frame and returns the corresponding basis.

**Usage**

```
kmsf2basis(sf)
```

**Arguments**

sf	Surmise function
----	------------------

**Value**

Matrix representing the basis.

---

kmsimulate	<i>Simulate a set of response patterns according to the BLIM</i>
------------	--

---

**Description**

kmsimulate returns a data set of n simulated response patterns based on the knowledge structure x given as a binary matrix. The simulation follows the BLIM (Basic Local Independence Model; see Doignon & Falmagne, 1999).

**Usage**

```
kmsimulate(x, n, beta, eta)
```

**Arguments**

x	Binary matrix representing a knowledge space
n	Number of simulated response patterns
beta	Careless error probability value or vector
eta	Lucky guess probability value or vector

**Details**

The beta and eta parameters must be either single numericals or vectors with a length identical to the number of rows in the x matrix. A mixture is possible.

The ‘sample’ function used by ‘kmsimulate’ might work inaccurately for knowledge structures ‘x’ with  $2^{31}$  or more states.

**Value**

Binary matrix representing the simulated data set

**References**

Doignon, J.-P. & Falmagne, J.-C. (1999). Knowledge Spaces. Springer–Verlag, Berlin.

**Examples**

```
kmsimulate(xpl$space, 50, 0.2, 0.1)
kmsimulate(xpl$space, 50, c(0.2, 0.25, 0.15, 0.2), c(0.1, 0.15, 0.05, 0.1))
kmsimulate(xpl$space, 50, c(0.2, 0.25, 0.15, 0.2), 0)
```

---

kmsurmisefunction      *Compute the surmise function for a knowledge space or basis*

---

**Description**

kmsurmisefunction returns a data frame representing the surmise function for a knowledge space or basis. The rows of the data frame are ordered by item name.

**Usage**

```
kmsurmisefunction(x)
```

**Arguments**

x                      Binary matrix representing a knowledge space or basis

**Value**

Data frame representing the surmise unction of x.

**See Also**

Other Different representations for knowledge spaces: [kmbasis\(\)](#), [kmsurmiserelation\(\)](#), [kmunionclosure\(\)](#)

**Examples**

```
kmsurmisefunction(xpl$space)
```

---

kmsurmiserelation	<i>Compute the surmise relation of a quasi-ordinal knowledge space</i>
-------------------	--

---

**Description**

kmsurmiserelation returns a matrix representing the surmise relation of a quasi-ordinal knowledge space. If  $x$  is a general knowledge space, a knowledge structure or an arbitrary family of sets, kmsurmiserelation returns the surmise relation of the smallest quasi-ordinal knowledge space containing  $x$ .

**Usage**

```
kmsurmiserelation(x)
```

**Arguments**

$x$	Binary matrix representing a quasi-ordinal knowledge space
-----	--

**Value**

Binary matrix representing the surmise relation of the corresponding quasi-ordinal knowledge space

Note: The columns of the surmise relation matrix describe the minimal state for the respective item in the quasi-ordinal knowledge space.

**See Also**

Other Different representations for knowledge spaces: [kmbasis\(\)](#), [kmsurmisefunction\(\)](#), [kmunionclosure\(\)](#)

**Examples**

```
kmsurmiserelation(xpl$space)
```

---

kmsymmsetdiff	<i>Compute the symmetric set difference between two sets</i>
---------------	--

---

**Description**

Compute the symmetric set difference between two sets

**Usage**

```
kmsymmsetdiff(x, y)
```

```
kmsetdistance(x, y)
```

**Arguments**

x	Binary vector representing a set
y	Binary vector representing a set

**Value**

kmsymmsetdiff: Symmetric set difference between 'x' and 'y'

kmsetdistance: Distance between the sets 'x' and 'y', i.e. the cardinality of the symmetric set difference

**Examples**

```
kmsymmsetdiff(c(1,0,0), c(1,1,0))
```

```
kmsetdistance(c(1,0,0), c(1,1,0))
```

---

kmtrivial	<i>Create trivial knowledge spaces</i>
-----------	--

---

**Description**

These functions create trivial knowledge spaces of a given item number. The minimal space contains just the empty set and the full item set while the maximal space is equal to the power set.

**Usage**

```
kmminimalspace(noi)
```

```
kmmaximalspace(noi)
```

**Arguments**

noi	Number of items
-----	-----------------

**Details**

Please note that the computation time for creating large power sets can grow quite large easily.

**Value**

A binary matrix representing the respective knowledge space

**Examples**

```
kmminimalspace(5)
```

```
kmmaximalspace(5)
```

---

kmunionclosure	<i>Close a family of sets under union</i>
----------------	---

---

**Description**

kmunionclosure returns a matrix representing a knowledge space. Please note that it takes quite some time for computing larger knowledge spaces.

**Usage**

```
kmunionclosure(x)
```

**Arguments**

x                    Binary matrix representing a family of sets

**Value**

Binary matrix representing the corresponding knowledge space, i.e. the closure of the family under union including the empty set and the full set.

kmconstrDowling implements the irredundant algorithm developed by Dowling (1993).

**References**

Dowling, C. E. (1993). On the irredundant construction of knowledge spaces. *Journal of Mathematical Psychology*, 37, 49–62.

**See Also**

Other Different representations for knowledge spaces: [kmbasis\(\)](#), [kmsurmisefunction\(\)](#), [kmsurmiserefaction\(\)](#)

**Examples**

```
kmunionclosure(xpl$basis)
```

---

kmvalidate	<i>Validate a knowledge structure against a data set</i>
------------	--

---

**Description**

kmvalidate returns a list with three elements, a named vector with the frequencies of distances between a set of response patterns and a knowledge structure, the Discrepancy Index (DI), and the Distance Agreement Coefficient (DA).

**Usage**

```
kmvalidate(data, struct)
```

**Arguments**

data	Binary matrix representing a set of response patterns
struct	Binary matrix representing a knowledge structure

**Value**

A list with three elements:

**dist** Distance distribution vector

**DI** Discrepancy Index

**DA** Distance Agreement Coefficient

**Warning**

The DA computation can take quite some time for larger item sets as the power set has to be computed.

**See Also**

Other Validating knowledge spaces: [kmdist\(\)](#)

**Examples**

```
kmvalidate(xpl$data, xpl$space)
```

---

readwrite

*Knowledge spaces on reading and writing abilities*

---

**Description**

Bases of knowledge spaces on reading/writing abilities obtained from querying experts.

**Usage**

```
readwrite
```

**Format**

A list containing four bases (rw1 to rw3, and rwmaj) in binary matrix form. Each matrix has 48 columns representing the different knowledge items and a varying number of rows containing the basis elements.



**Details**

Three experts were queried about prerequisite relationships between 48 items on reading and writing abilities (Dowling, 1991; 1993). A forth basis represents those prerequisite relationships on which the majority of the experts agree (Dowling & Hockemeyer, 1998).

**References**

Dowling, C. E. (1991). Constructing Knowledge Structures from the Judgements of Experts. Habilitationsschrift, Technische Universität Carolo-Wilhelmina, Braunschweig, Germany.

Dowling, C. E. (1993). Applying the basis of a knowledge space for controlling the questioning of an expert. *Journal of Mathematical Psychology*, 37, 21–48.

Dowling, C. E. & Hockemeyer, C. (1998). Computing the intersection of knowledge spaces using only their basis. In Cornelia E. Dowling, Fred S. Roberts, & Peter Theuns, editors, *Recent Progress in Mathematical Psychology*, pp. 133–141. Lawrence Erlbaum Associates Ltd., Mahwah, NJ.

**See Also**

Other Data: [cad](#), [fractions](#), [xpl](#)

---

xpl

*Small example knowledge space*

---

**Description**

Basis and space matrix of a small fictional knowledge space, and a small data set to be used in examples.

**Usage**

xpl

**Format**

A list containing the basis, the space, and the data matrix

**See Also**

Other Data: [cad](#), [fractions](#), [readwrite](#)

# Index

- \* **Data**
    - cad, [2](#)
    - fractions, [3](#)
    - readwrite, [16](#)
    - xpl, [17](#)
  - \* **Different representations for knowledge spaces**
    - kmbasis, [4](#)
    - kmsurmisefunction, [12](#)
    - kmsurmiserelation, [13](#)
    - kmunionclosure, [15](#)
  - \* **Neighbourhood & fringe**
    - kmfringe, [7](#)
    - kmneighbourhood, [9](#)
  - \* **Plotting knowledge structures**
    - kmbasisdiagram, [5](#)
    - kmhasse, [8](#)
  - \* **Properties of knowledge structures**
    - kmeqreduction, [6](#)
    - kmiswellgraded, [8](#)
    - kmnotions, [10](#)
  - \* **Simulating response patterns**
    - kmsimulate, [11](#)
  - \* **Trivial knowledge spaces**
    - kmtrivial, [14](#)
  - \* **Utilities**
    - kmsymmsetdiff, [13](#)
  - \* **Validating knowledge spaces**
    - kmdist, [6](#)
    - kmvalidate, [15](#)
  - \* **data**
    - cad, [2](#)
    - fractions, [3](#)
    - readwrite, [16](#)
    - xpl, [17](#)
  - \* **math**
    - kmbasis, [4](#)
    - kmbasisdiagram, [5](#)
    - kmcolors, [5](#)
    - kmdist, [6](#)
    - kmeqreduction, [6](#)
    - kmfringe, [7](#)
    - kmhasse, [8](#)
    - kmiswellgraded, [8](#)
    - kmmaximalspace (kmtrivial), [14](#)
    - kmminimalspace (kmtrivial), [14](#)
    - kmneighbourhood, [9](#)
    - kmnneighbourhood, [10](#)
    - kmnotions, [10](#)
    - kmsetdistance (kmsymmsetdiff), [13](#)
    - kmsf2basis, [11](#)
    - kmsimulate, [11](#)
    - kmsurmisefunction, [12](#)
    - kmsurmiserelation, [13](#)
    - kmsymmsetdiff, [13](#)
    - kmtrivial, [14](#)
- cad, [2](#), [4](#), [17](#)
  - fractions, [3](#), [3](#), [17](#)
  - kmbasis, [4](#), [12](#), [13](#), [15](#)
  - kmbasisdiagram, [5](#), [8](#)
  - kmcolors, [5](#)
  - kmdist, [6](#), [16](#)
  - kmeqreduction, [6](#), [9](#), [11](#)
  - kmfringe, [7](#), [9](#)
  - kmhasse, [5](#), [8](#)
  - kmiswellgraded, [7](#), [8](#), [11](#)
  - kmmaximalspace (kmtrivial), [14](#)
  - kmminimalspace (kmtrivial), [14](#)
  - kmneighbourhood, [7](#), [9](#)
  - kmnneighbourhood, [10](#)
  - kmnotions, [7](#), [9](#), [10](#)
  - kmsetdistance (kmsymmsetdiff), [13](#)
  - kmsf2basis, [11](#)
  - kmsimulate, [11](#)
  - kmsurmisefunction, [4](#), [12](#), [13](#), [15](#)
  - kmsurmiserelation, [4](#), [12](#), [13](#), [15](#)
  - kmsymmsetdiff, [13](#)
  - kmtrivial, [14](#)

`kmunionclosure`, [4](#), [12](#), [13](#), [15](#)

`kmvalidate`, [6](#), [15](#)

`readwrite`, [3](#), [4](#), [16](#), [17](#)

`xpl`, [3](#), [4](#), [17](#), [17](#)