Package: CbKST (via r-universe)

July 1, 2026

Type Package

Title Competence-Based Knowledge Space Theory

Date 2026-06-19

Version 0.1-0

Description Competence-based knowledge space theory (CbKST) is an extension of knowledge space theory (KST) modeling the latent skills and competencies underlying the observable response behaviour as described by Heller & Stefanutti (2024) <doi:10.1142/9789811280481_0001>. The package focuses on the mappings between competence and performance ledvel (skill (multi) map, problem function etc.).

License GPL-3

Depends R (>= 4.5.0)

Imports readODS, openxlsx, tools, kstMatrix (>= 2.3-3)

Suggests litedown

Encoding UTF-8

LazyData true

VignetteBuilder litedown

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Cord Hockemeyer [aut, cre]

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

Config/pak/sysreqs
cmake libglpk-dev make libicu-dev librsvg2-dev libuv1-dev libxml2-dev libx11-dev

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

Date/Publication 2026-06-30 11:20:02 UTC

RemoteUrl https://github.com/cran/CbKST

RemoteRef HEAD

RemoteSha bbc83c689d34a739702639d1f74cb54f432631d7

Contents

`cbkst_comp2perf` *Determine the performance state delineated by a competence state*

Description

`cbkst_comp2perf()` determines the performance state, i.e. the subset of items solvable for a person in a given competence state.

Usage

```
cbkst_comp2perf(comp, multimap)
```

Arguments

`comp`	A binary vector representing the competence state
`multimap`	A data frame of class cbkst_skillmultimap

Value

A binary vector representing the correspondinng performance state

See Also

Other Mapping states: `cbkst_perf2comp()`, `cbkst_simple_perf2comp()`

Examples

```
cbkst_comp2perf(c(1,1,0,0), exampledata$mu)
```

cbkst_competencestructure

Determine the skill structure delineated by a skill multimap and an optional performance structure

Description

cbkst_performancestructure() determines the performanxce state, i.e. the Determine the structure of all skill states producing some perfomrance state

Usage

```
cbkst_competencestructure(multimap, perf = NULL)
```

Arguments

multimap	A data frame of class cbkst_skillmultimap
perf	Optinal performance structure as kmstructure object

Value

A kmstructure object representing the resulting skill structure

See Also

Other Mapping structures: cbkst_performancestructure()

Examples

```
cbkst_competencestructure(exampledata$mu)
```

cbkst_perf2comp

Determine the competence state behind a given performance state

Description

cbkst_perf2comp() determines competence states underlying a given performance state under the condition of a skill multimap and optionally a competence structure.

Usage

```
cbkst_perf2comp(perf, multimap, comp = NULL)
```

Arguments

perf	A **named** binary vector representing the performance state
multimap	A data frame of class cbkst_skillmultimap
comp	Optional constraining competence structure

Value

A matrix representing the minimal competence states for item mastery according to perf

See Also

Other Mapping states: cbkst_comp2perf(), cbkst_simple_perf2comp()

Examples

```
perf <- c(1,1,1,0,0)
names(perf) <- c("z", "y", "x", "w", "v")
cbkst_perf2comp(perf, exampledata$multi)
```

---

cbkst_performancestructure

Determine the performance structure delineated by a skill multimap and an optional competence structure

---

Description

cbkst_performancestructure() determines the performanxce state, i.e. the Determine the structure of all performance states reachable from some skill state

Usage

```
cbkst_performancestructure(multimap, comp = NULL)
```

Arguments

multimap	A data frame of class cbkst_skillmultimap
comp	Optinal competence structure as kmstructure object

Value

A kmstructure object representing the resulting performance structure

See Also

Other Mapping structures: cbkst_competencestructure()

Examples

```
cbkst_performancestructure(exampledata$multi)
```

`cbkst_simple_perf2comp` *Determine the competence state behind a given performance state*

Description

`cbkst_simnple_perf2comp()` determines competence state underlying a given performance state under the condition of a skill map.

Usage

```
cbkst_simple_perf2comp(perf, skillmap)
```

Arguments

`perf`	A binary vector representing the performance state
`skillmap`	A data frame of class `cbkst_skillmap`

Value

A binary vector representing the corresponding competence state

See Also

Other Mapping states: `cbkst_comp2perf()`, `cbkst_perf2comp()`

Examples

```
perf <- c(1,1,1,0,0)
names(perf) <- c("z", "y", "x", "w", "v")
cbkst_simple_perf2comp(perf, exampledata$mu)
```

exampledata *Small example data set*

Description

This small example data set contains a skill map `mu`, a skill multimap `multi`, a competence space `cspace`, and a performance space `pspace`.

Usage

```
exampledata
```

Format

List with four elements: `mu` (skill map), `multi` (skill multimap), `cspace` (competence space), and `pspace` (performance space)

read_skillmultimap *Read a skill multimap from file*

Description

`read_skillmultimap()` reads a skill multimap as data frame.

Usage

```
read_skillmultimap(file)
```

Arguments

`file`	Filename to read

Value

Data frame of class `cbkst_skillmultimap`. In case of a skill map, it also has the class `cbkst_skillmap`

Data file and resulting data table have the following format: The first column contains the item and the subsequent rows denote for each skill if it is contained in $\mu(q)$. ODS and XLSX files are recognised.

Examples

```
fpath <- system.file("extdata", "skillmap.ods", package="CbKST")
read_skillmultimap(fpath)
```

Index