

Package: CDSS (via r-universe)

October 7, 2026

Type Package

Title Course-Dependent Skill Structures

Version 1.0-0

Date 2026-09-06

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

License GPL-3

Depends R (>= 4.3.0)

Imports readODS (>= 2.0.0), openxlsx (>= 4.2.0), utils

Suggests knitr, DiagrammeRsvg, Rgraphviz, rsvg, png, kstMatrix (>= 3.0-0), CbKST

NeedsCompilation no

Description Deriving skill structures from skill assignment data for courses (sets of learning objects).

Encoding UTF-8

Config/roxygen2/version 8.1.0

VignetteBuilder knitr

Author Cord Hockemeyer [aut, cre]

Config/pak/sysreqs libicu-dev

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

Date/Publication 2026-09-07 08:20:02 UTC

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

RemoteRef HEAD

RemoteSha a71976f0bd3ac9010b8c64860cc42d3462c597e1

Contents

CDSS	2
cdss_circular_requirements	3
cdss_lo_sa2af	4

cdss_lo_sa2ar	5
cdss_missing_los	5
cdss_nonteaching_los	6
cdss_read_skill_assignment	6
cdss_read_skill_assignment_csv	7
cdss_read_skill_assignment_ods	8
cdss_read_skill_assignment_xlsx	9
cdss_sa_compliance	10
cdss_sa_describes_sr	10
cdss_sa2mu	11
cdss_skill_sa2af	11
cdss_skill_sa2ar	12
cdss_tables2sa	13

Index	14
--------------	-----------

CDSS

CDSS: Course dependent skill structures

Description

The CDSS package provides functions for a complete workflow from skill assignment tables to surmise mappings on the sets of skills and learning objects, respectively.

Suggested workflow for the general case

1. Read the skill assignment using one of the `read_skill_assignments_xxx()` functions.
2. Check the compliance to the definition for skill assignments using `cdss_sa_compliance()`.
3. Determine the attribution function on skills and on learning objects applying the `cdss_skill_sa2af()` and `cdss_lo_sa2af()` functions, respectively.
4. Close the attribution functions to surmise functions using `kstMatrix::kmclosure()`.
5. Check if the skill assignment leads to a quasi-ordinal space using `cdss_sa_describes_sr()`.
6. If TRUE, determine attribution relations on skills and learning objects using `cdss_skill_sa2ar()` and `cdss_lo_sa2ar()`, respectively.
7. Close the attribution relations to surmise relations applying `kstMatrix::kmclosure()`.
8. Continue with functions from the `kstMatrix` package, e.g., to obtain a basis and further on a skill space.

Suggested workflow for the special case of one LO per skill

1. Read the skill assignment using one of the `read_skill_assignments_xxx()` functions.
2. Check whether the skill assignment allows for the derivation of a surmise relation using `cdss_sa_describes_sr()`.
3. If yes, derive an attribution relation from the skill assignment using `cdss_skill_sa2ar()`.
4. Close the attribution relation to a surmise relation using `kstMatrix::kmclosure()`.
5. Continue with functions from the `kstMatrix` package, e.g., to obtain a basis and further on a skill space.

Data files

The installation of this package includes several data files as examples in the `extdata` sub directory (see the Examples below for how to access the files there). There are four data sets, `KST`, `KST-Intro`, `SkillAssignment`, and `ErroneousSkillAssignment`. The `SkillAssignment` data set is available in three formats, `ODS`, `XLSX`, and `CSV` (in `CSV` format, there are two files each, `SkillAssignment-R.csv` and `SkillAssignment-T.csv`, for required and taught skills, respectively). The other three data sets are available in `ODS` format only.

`SkillAssignment` and `ErroneousSkillAssignment` are small example data sets where the latter fails for `cdss_sa_compliance()`. `KST` contains a skill assignment for the course on knowledge space theory under <https://moodle.qhelp.eu/>. `KST-Intro` contains the reduction of `KST` to the first chapter of that course.

References

Hockemeyer, C. (2022). Building Course-Dependent Skill Structures - Applying Competence based Knowledge Space Theory to Itself. Manuscript in preparation.

Acknowledgements

The creation of this R package was financially supported by the Erasmus+ Programme of the European Commission through the QHELP project (<https://qhelp.eu/>).

Examples

```
library(readODS)
fpath <- system.file("extdata", "SkillAssignment.ods", package="CDSS")
sa <- cdss_read_skill_assignment_ods(fpath)
sa
ssf <- kstMatrix::kmclosure(cdss_skill_sa2af(sa))
ssf
lsf <- kstMatrix::kmclosure(cdss_lo_sa2af(sa))
lsf
```

`cdss_circular_requirements`

Vector of learning objects requiring and teaching the same skill

Description

`cdss_circular_requirements` expects skill assignment and returns a vector of learning objects which require a skill that they teach.

Usage

```
cdss_circular_requirements(sa)
```

Arguments

sa Skill assignment

Value

Vector of learning objects

See Also

Other Functions testing validity of skill assignments: [cdss_missing_los\(\)](#), [cdss_nonteaching_los\(\)](#), [cdss_sa_compliance\(\)](#)

cdss_lo_sa2af

Determine Attribution function for LOs from a skill assignment

Description

cdss_lo_sa2af expects a skill assignment and derives an attribution function on the set of learning objects.

Usage

```
cdss_lo_sa2af(sa)
```

Arguments

sa Skill assignment object

Value

Object of class `cdss_af` (attribution function).

See Also

Other functions building skill (multi) assignment matrices: [cdss_skill_sa2af\(\)](#), [cdss_tables2sa\(\)](#)

cdss_lo_sa2ar	<i>Create an attribution relation on learning objects from a skill assignment.</i>
---------------	--

Description

cdss_lo_sa2ar expects a skill assignment and derives an attribution relation on learning objects if the skill assignment fulfills the necessary conditions, i.e. if there is only one teaching LO per skill.

Usage

```
cdss_lo_sa2ar(sa)
```

Arguments

sa	Skill assignment object
----	-------------------------

Value

attribution relation or NULL

See Also

Other functions deriving skill structures from skill assignments: [cdss_sa_describes_sr\(\)](#), [cdss_skill_sa2ar\(\)](#)

cdss_missing_los	<i>Vector of skills without teaching learning objects.</i>
------------------	--

Description

cdss_missing_los expects a skill assignment and returns a vector of skills which are not taught by any learning object.

Usage

```
cdss_missing_los(sa)
```

Arguments

sa	SKill assignment
----	------------------

Value

Vector of skills

See Also

Other Functions testing validity of skill assignments: [cdss_circular_requirements\(\)](#), [cdss_nonteaching_los\(\)](#), [cdss_sa_compliance\(\)](#)

`cdss_nonteaching_los` *Vector of learning objects not teaching any skills.*

Description

`cdss_nonteaching_los` expects a skill assignment and returns a vector of learning objects which do not teach any skill.

Usage

```
cdss_nonteaching_los(sa)
```

Arguments

`sa` Skill assignment

Value

Vector of learning objects

See Also

Other Functions testing validity of skill assignments: [cdss_circular_requirements\(\)](#), [cdss_missing_los\(\)](#), [cdss_sa_compliance\(\)](#)

`cdss_read_skill_assignment`
Read a skill assignment from file

Description

Read a skill assignment from file

Usage

```
cdss_read_skill_assignment(  
  filename = NULL,  
  taught = "Taught",  
  required = "Required",  
  sep = ",",  
  warnonly = FALSE,  
  verbose = FALSE  
)
```

Arguments

filename	Name of the file for ODS or XLSX files
taught	Name of the sheet/file for taught skill assignments
required	Name of the sheet/file for required skill assignments
sep	Cell-separating character for CSV files
warnonly	Are non-compliant SAs allowed? (default FALSE)
verbose	Verbosity (default TRUE)

Value

Skill assignment ('cdss_sa' object)

Skill assignments are stored in a pair of tables, 'taught' and 'required'. For ODS and XLSX files, these must be sheets of ****one**** file; CSV does not support multi-sheet file and, therefore, two files must be specified there.

For CSV files, the 'filename' parameter must be 'NULL', and the two filenames are specified in the 'taught' and 'required' parameters. For ODS and XLSX, the 'filename' must be specified, and the 'taught' and 'required' parameters specify the names of the respective sheets within that file.

cdss_read_skill_assignment_csv

DEPRECATED: Read an assignment of taught and required skills for a set of learning objects from CSV-files.

Description

cdss_read_skill_assignment expects two CSV-files with two columns each. The first column contains the IDs of learning objects and the second row the IDs of single skills required or taught, respectively, by this learning object. It returns a list of two binary matrices, "taught" and "required". Each matrix has one row per learning object and one column per skill. The cells contain a "1" if the skill is taught or required, respectively, by the learning object and a "0" otherwise,

Usage

```
cdss_read_skill_assignment_csv(
  taught,
  required,
  header = TRUE,
  sep = ",",
  dec = ".",
  warnonly = FALSE,
  verbose = TRUE
)
```

Arguments

taught	CSV-file with assignments of taught competencies to learning objects
required	CSV-file with assignments of required competencies to learning objects
header	Boolean specifying whether the CSV-files contain a header line (default = TRUE)
sep	Column separator (default ",")
dec	Decimal point character (default ".")
warnonly	Are non-compliant SAs allowed? (default = FALSE)
verbose	Verbosity of compliance test (default = TRUE)

Value

List of two binary matrices, "taught" and "required".

See Also

Other functions reading skill assignments: [cdss_read_skill_assignment_ods\(\)](#), [cdss_read_skill_assignment_xlsx\(\)](#)

cdss_read_skill_assignment_ods

DEPRECATED: Read an assignment of taught and required skills for a set of learning objects from an ODS-file.

Description

cdss_read_skill_assignment_ods expects an ODS-file with two sheets assigning taught and required, respectively, skills to learning objects with two columns each. The first column contains the IDs of learning objects and the second row the IDs of single skills required or taught, respectively, by this learning object. It returns a list of two binary matrices, "taught" and "required". Each matrix has one row per learning object and one column per skill. The cells contain a "1" if the skill is taught or required, respectively, by the learning object and a "0" otherwise,

Usage

```
cdss_read_skill_assignment_ods(
  filename,
  taughtname = "Taught",
  requiredname = "Required",
  warnonly = FALSE,
  verbose = TRUE
)
```

Arguments

filename	Name of the ODS-file
taughtname	Name of the sheet with required assignment (default = "Taught")
requiredname	Name of the sheet with required assignment (default = "Required")
warnonly	Are non-compliant SAs allowed? (default = FALSE)
verbose	Verbosity (default = TRUE)

Value

List of two binary matrices, "taught" and "required".

See Also

Other functions reading skill assignments: [cdss_read_skill_assignment_csv\(\)](#), [cdss_read_skill_assignment_xlsx\(\)](#)

cdss_read_skill_assignment_xlsx

DEPRECATED: Read an assignment of taught and required skills for a set of learning objects from an XLSX-file.

Description

cdss_read_skill_assignment_xlsx expects an XLSX-file with two sheets assigning taught and required, respectively, skills to learning objects with two columns each. The first column contains the IDs of learning objects and the second row the IDs of single skills required or taught, respectively, by this learning object. It returns a list of two binary matrices, "taught" and "required". Each matrix has one row per learning object and one column per skill. The cells contain a "1" if the skill is taught or required, respectively, by the learning object and a "0" otherwise,

Usage

```
cdss_read_skill_assignment_xlsx(
  filename,
  taughtname = "Taught",
  requiredname = "Required",
  warnonly = FALSE,
  verbose = TRUE
)
```

Arguments

filename	Name of the XLSX-file
taughtname	Name of the sheet with required assignment (default = "Taught")
requiredname	Name of the sheet with required assignment (default = "Required")
warnonly	Are non-compliant SAs allowed? (default = FALSE)
verbose	Verbosity of compliance test (default = TRUE)

Value

List of two binary matrices, "taught" and "required".

See Also

Other functions reading skill assignments: [cdss_read_skill_assignment_csv\(\)](#), [cdss_read_skill_assignment_ods\(\)](#)

cdss_sa_compliance	<i>Check whether a skill assignment is compliant to the CDCS conditions.</i>
--------------------	--

Description

cdss_sa_compliance expects a skill assignment and checks whether it is compliant to the conditions for CDCS.

Usage

```
cdss_sa_compliance(sa, verbose = FALSE)
```

Arguments

sa	Skill assignment
verbose	Verbosity (default = FALSE)

Value

Boolean

See Also

Other Functions testing validity of skill assignments: [cdss_circular_requirements\(\)](#), [cdss_missing_los\(\)](#), [cdss_nonteaching_los\(\)](#)

cdss_sa_describes_sr	<i>Check whether a surmise relation can be derived from a given skill assignment.</i>
----------------------	---

Description

cdss_sa_describes_sr expects a list of two matrices (taught and required) of a skill assignment. It returns TRUE if the skill assignment describes a surmise relation (i.e. there is only one teaching LO per skill) and FALSE.

Usage

```
cdss_sa_describes_sr(sa, verbose = FALSE)
```

Arguments

sa Skill assignment object
verbose Flag, default is FALSE

Value

Logical value

See Also

Other functions deriving skill structures from skill assignments: [cdss_lo_sa2ar\(\)](#), [cdss_skill_sa2ar\(\)](#)

cdss_sa2mu	<i>Derive a skill map from a skill assignment</i>
------------	---

Description

Derive a skill map from a skill assignment

Usage

cdss_sa2mu(sa)

Arguments

sa Skill assignment

Value

Skill multi map (cbkst_skillmultimap object); if sa describes a surmise relation, the result is also a cbkst_skillmap.

cdss_skill_sa2af	<i>Determine Attribution function for skills from a skill assignment</i>
------------------	--

Description

cdss_skill_sa2af expects a skill assignment and derives an attribution function on the set of learning objects.

Usage

cdss_skill_sa2af(sa)

Arguments

sa Skill assignment object

Value

Object of class `kmatrixfunction`

See Also

Other functions building skill (multi) assignment matrices: [cdss_lo_sa2af\(\)](#), [cdss_tables2sa\(\)](#)

cdss_skill_sa2ar	<i>Create an attribution relation on skills from a skill assignment.</i>
------------------	--

Description

`cdss_skill_sa2ar` expects a skill assignment and derives an attribution relation on skills if the skill assignment fulfills the necessary conditions, i.e. if there is only one teaching LO per skill.

Usage

```
cdss_skill_sa2ar(sa)
```

Arguments

sa Skill assignment object

Value

attribution relation or NULL

See Also

Other functions deriving skill structures from skill assignments: [cdss_lo_sa2ar\(\)](#), [cdss_sa_describes_sr\(\)](#)

cdss_tables2sa	<i>Build matrices of taught and required, respectively, skills for learning objects from respective tables.</i>
----------------	---

Description

cdss_tables2sa expects two data frames with two columns each. The first column contains the IDs of learning objects and the second row the IDs of single skills required or taught, respectively, by this learning object. It returns a list of two binary matrices, "taught" and "required". Each matrix has one row per learning object and one column per skill. The cells contain a "1" if the skill is taught or required, respectively, by the learning object and a "0" otherwise.

Usage

```
cdss_tables2sa(taught, required)
```

Arguments

taught	Data table containing the assignment of taught skills to learning objects
required	Data table containing the assignment of required skills to learning objects

Value

List of two binary matrices, "taught" and "required".

See Also

Other functions building skill (multi) assignment matrices: [cdss_lo_sa2af\(\)](#), [cdss_skill_sa2af\(\)](#)

Index

* Functions testing validity of skill

assignments

cdss_circular_requirements, 3
cdss_missing_los, 5
cdss_nonteaching_los, 6
cdss_sa_compliance, 10

* functions building skill (multi) assignment matrices

cdss_lo_sa2af, 4
cdss_skill_sa2af, 11
cdss_tables2sa, 13

* functions deriving skill structures from skill assignments

cdss_lo_sa2ar, 5
cdss_sa_describes_sr, 10
cdss_skill_sa2ar, 12

* functions reading skill assignments

cdss_read_skill_assignment_csv, 7
cdss_read_skill_assignment_ods, 8
cdss_read_skill_assignment_xlsx, 9

CDSS, 2

CDSS-package (CDSS), 2

cdss_circular_requirements, 3

cdss_circular_requirements(), 6, 10

cdss_lo_sa2af, 4

cdss_lo_sa2af(), 2, 12, 13

cdss_lo_sa2ar, 5

cdss_lo_sa2ar(), 2, 11, 12

cdss_missing_los, 5

cdss_missing_los(), 4, 6, 10

cdss_nonteaching_los, 6

cdss_nonteaching_los(), 4, 6, 10

cdss_read_skill_assignment, 6

cdss_read_skill_assignment_csv, 7

cdss_read_skill_assignment_csv(), 9, 10

cdss_read_skill_assignment_ods, 8

cdss_read_skill_assignment_ods(), 8, 10

cdss_read_skill_assignment_xlsx, 9

cdss_read_skill_assignment_xlsx(), 8, 9

cdss_sa2mu, 11

cdss_sa_compliance, 10

cdss_sa_compliance(), 2, 4, 6

cdss_sa_describes_sr, 10

cdss_sa_describes_sr(), 2, 5, 12

cdss_skill_sa2af, 11

cdss_skill_sa2af(), 2, 4, 13

cdss_skill_sa2ar, 12

cdss_skill_sa2ar(), 2, 5, 11

cdss_tables2sa, 13

cdss_tables2sa(), 4, 12

kstMatrix::kmclosure(), 2