

# Package ‘klassR’

October 9, 2019

**Type** Package

**Title** Classifications and Codelists for Statistics Norway

**Version** 0.1.2

**Author** Susie Jentoft, Lisa Li, Diana-Cristina Iancu

**Maintainer** Susie Jentoft <susie.jentoft@ssb.no>

**Description** Functions to search, retrieve and apply classifications and codelists using Statistics Norway's API <<https://www.ssb.no/klass>> from the system 'KLASS'. Retrieves classifications by date with options to choose language, hierarchical level and formatting.

**Imports** tm, httr, jsonlite

**BugReports** <https://github.com/statisticsnorway/klassR/issues>

**License** Apache License 2.0

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-10-09 09:50:02 UTC

## R topics documented:

ApplyKlass . . . . .	2
CheckDate . . . . .	3
ConvertTable . . . . .	3
CorrespondList . . . . .	4
formatting . . . . .	4
GetFamily . . . . .	5
GetKlass . . . . .	6
GetName . . . . .	7
GetNums . . . . .	7
GetUrl . . . . .	8

GetUrl2 . . . . .	8
GetVersion . . . . .	9
klassdata . . . . .	9
levelCheck . . . . .	10
Levels . . . . .	11
ListFamily . . . . .	11
ListKlass . . . . .	12
MakeChar . . . . .	12
MakeUrl . . . . .	13
SearchKlass . . . . .	14
splitChar . . . . .	14
<b>Index</b>	<b>15</b>

---

ApplyKlass	<i>Match and convert a classification</i>
------------	---

---

## Description

Match and convert a classification

## Usage

```
ApplyKlass(x, klass, date = NULL, correspond = NULL, language = "nb",
           output_level = NULL, output = "name", format = TRUE)
```

## Arguments

x	Input vector
klass	Classification number
date	Date for classification (format = "YYYY-mm-dd"). Default is current date
correspond	ID number for target in correspondence table. For correspondence between two dates within the same classification, use correspond = TRUE.
language	Default "nb" for Norwegian (Bokmål). Also "nn" (Nynorsk) and "en" (English available for some classifications)
output_level	Desired output level
output	String describing output. May be "name" (default), "code" or "both".
format	Logical for whether to run formatting on input vector x (Default = TRUE), important to check if formatting is in one level.

## Value

A vector or data frame is returned with names and/or code of the desired output level.

**Examples**

```

data(klassdata)
klassdata$kommune_names <- ApplyKlass(x = klassdata$kommune, klass = 131, format=FALSE)
klassdata$name <- ApplyKlass(x = klassdata$nace5, klass = 6, language = "en", format=FALSE)
klassdata$name2 <- ApplyKlass(x = klassdata$nace5, klass = 6, date = "2015-01-01", format=FALSE)

```

---

CheckDate	<i>Internal function to check date</i>
-----------	--

---

**Description**

Internal function to check date

**Usage**

```
CheckDate(date)
```

**Arguments**

date	Date
------	------

---

ConvertTable	<i>Code level convert table (internal function)</i>
--------------	---

---

**Description**

Code level convert table (internal function)

**Usage**

```
ConvertTable(klass_data, code = "code", parentcode = "parentCode",
             level = "level")
```

**Arguments**

klass_data	Klass data frame to convert
code	Name of variable containing code (Default = "code")
parentcode	Name of variable of parent code (Default = "parentCode")
level	Name of variable containing level (Default = "level")

**Value**

A dataata frame

---

CorrespondList	<i>Correspondence list Print a list of correspondence tables for a given class with source and target IDs</i>
----------------	---

---

**Description**

Correspondence list Print a list of correspondence tables for a given class with source and target IDs

**Usage**

```
CorrespondList(klass, date = NULL)
```

**Arguments**

klass	Classification number
date	Date for classification (format = "YYYY-mm-dd"). Default is current date

**Value**

Data frame with list of corsepondence tables, source ID and target ID.

**Examples**

```
CorrespondList("7")
CorrespondList("131", date = "2016-01-01")
```

---

formatting	<i>Convert vector to the right format</i>
------------	---

---

**Description**

Convert vector to the right format

**Usage**

```
formatting(x, input_level, klass, klass_data)
```

**Arguments**

x	- vector of character
input_level	- which classification level
klass	- classification number
klass_data	- the right formatting to the classification levels

**Value**

vector of character

**Examples**

```
klass_data <- GetKlass(klass = "6", date = "2007-01-01")
input_level <- levelCheck(x = klassdata$nace5, klass_data = klass_data)
formattering(x = klassdata$nace5, input_level = input_level, klass = 6, klass_data=klass_data)
```

```
klass_data <- GetKlass(klass = "7", date = "2007-01-01")
input_level <- levelCheck(x = klassdata$occupation, klass_data = klass_data)
formattering(x = klassdata$occupation, input_level = input_level, klass = 7, klass_data=klass_data)
```

```
klass_data <- GetKlass(klass = "131", date = "2007-01-01")
input_level <- levelCheck(x = klassdata$occupation, klass_data = klass_data)
formattering(x = klassdata$kommune2, input_level = input_level, klass = 131, klass_data=klass_data)
```

---

GetFamily

*Identify corresponding family from a classification number*

---

**Description**

Identify corresponding family from a classification number

**Usage**

```
GetFamily(klass)
```

**Arguments**

klass            Classification number

**Value**

Family number

**Examples**

```
GetFamily(klass = 7)
```

---

GetKlass	<i>Fetch classification data Fetch Statistics Norway classification data using API</i>
----------	--

---

### Description

Fetch classification data Fetch Statistics Norway classification data using API

### Usage

```
GetKlass(klass, date = NULL, correspond = NULL, output_level = NULL,
         language = "nb", output_style = "normal")
```

### Arguments

klass	Number/string of the classification ID/number. (use Klass_list() to find this)
date	String for the required date of the classification. Format must be "yyyy-mm-dd". For an interval, provide two dates as a vector. If blank, will default to today's date.
correspond	Number/string of the target correspondence (if a correspondence table is requested).
output_level	Number/string specifying the requested heirachy level (optional).
language	Two letter string for the requested language output. Default is bokmål ("nb"). Nynorsk ("nn") and English ("en") also available for some classificatio.)
output_style	String varibale for the output type. Default is "normal" and only option currently programmed

### Value

The function returns a data frame of the specified classification/correspondence table. Output variables include: code, parentCode, level, and name for standard lists. For correspondence tables variables include: sourceCode, sourceName, targetCode and targetName. For time correspondence tables variables include: oldCode, oldName, newCode and newName.

### Examples

```
# Get classification for occupation classifications
head(GetKlass(klass = "7"))
# Get classification for occupation classifications in English
head(GetKlass(klass = "7", language = "en"))
# Get classifications for level 2 only
head(GetKlass(klass = "7", output_level = 2))
# Get classifications for level 2 only valid on a specified date of between two dates
head(GetKlass(klass = "7", output_level = 2, date = "2007-01-01"))
head(GetKlass(klass = "7", date = c("2007-01-01", "2018-01-01")))

# Get correspondence table between two occupation classifications
```

```
GetKlass(klass = "145", correspond = "7", date = "2018-01-01")
#Get correspondence table between two dates for municipality
GetKlass(klass = "131", correspond = TRUE, date = c("2015-01-01", "2019-01-01"))
```

---

GetName                      *Get the name of a classification version*

---

**Description**

Get the name of a classification version

**Usage**

```
GetName(version)
```

**Arguments**

version                      Version number

**Value**

string or vector of strings with name of version

**Examples**

```
GetName("33")
```

---

GetNums                      *Get target ID numbers from Url*

---

**Description**

Get target ID numbers from Url

**Usage**

```
GetNums(x)
```

**Arguments**

x                              Url address

**Value**

Number

---

GetUrl	<i>Get json file from Url</i>
--------	-------------------------------

---

**Description**

Get json file from Url

**Usage**

```
GetUrl(url)
```

**Arguments**

url	String url address
-----	--------------------

**Value**

text in json format

---

GetUrl2	<i>Get json file from Url - alternative version</i>
---------	---

---

**Description**

Get json file from Url - alternative version

**Usage**

```
GetUrl2(url)
```

**Arguments**

url	String url address
-----	--------------------

**Value**

text in json format



---

GetVersion	<i>Get version number of a class given a date</i>
------------	---

---

**Description**

Get version number of a class given a date

**Usage**

```
GetVersion(klass = NULL, date = NULL, family = NULL,  
           klassNr = FALSE)
```

**Arguments**

klass	Classification number
date	Date for version to be valid
family	Family ID number if a list of version number for all classes is desired
klassNr	True/False for whether to output classification numbers. Default = FALSE

**Value**

Number, vector or data frame with version numbers and classification numbers if specified.

**Examples**

```
GetVersion(7)  
GetVersion(7, "2010-01-01")  
  
GetVersion(family = 1)  
GetVersion(family = 1, klassNr = TRUE)[1:10,]
```

---

klassdata	<i>Testdata for klassR package</i>
-----------	------------------------------------

---

**Description**

A dataset containing variables for testing of Statistics Norways classification API with the klassR package. Some observations are missing or incorrect for testing and demonstrations.

**Usage**

```
klassdata
```

**Format**

A data frame containing 100 rows and 7 variables:

**ID** Identification number

**sex** 1/2 variable for sex

**education** 4-digit number for education standard ISCED97 (level and subject area) NUS (klass = 66) 2015.01.01

**kommune** 4-digit code for Norwegian municipality (klass = 131). Based on 2015.01.01

**kommune2** Numeric variable for Norwegian municipality with dropped leading zero's for testing (klass = 131). Based on 2015.01.01

**nace5** 5-digit code for industry (NACE). Based on 01.01.2015 standard industry codes (klass = 7)

**occupation** 4-digit occupation codes using standard for STYRK-08 (klass = 7) 2015.01.01

---

levelCheck	<i>checking which level input data is</i>
------------	---

---

**Description**

checking which level input data is

**Usage**

```
levelCheck(x, klass_data)
```

**Arguments**

**x** - vector of character  
**klass\_data** - - the right formatting to the classification levels

**Value**

The heirachical level of the input data is returned.

**Examples**

```
data(klassdata)

sn <- GetKlass(klass = "6", date = "2007-01-01")
levelCheck(x = klassdata$nace5, klass_data = sn)

sn <- GetKlass(klass = "7", date = "2007-01-01")
levelCheck(x = klassdata$occupation, klass_data = sn)
```

---

Levels	<i>Title</i>
--------	--------------

---

**Description**

Title

**Usage**

```
Levels(input_level, output_level, klass_data)
```

**Arguments**

input_level	Classification level from the original dataset.
output_level	Classification level for which the codes and names are desired by the user.
klass_data	Classification file retrieved from KLASS.

**Value**

Data frame with the input and extra desired classification levels. Includes codes and names for each level.

**Examples**

```
klass_data <- GetKlass(klass = "6")
Levels(input_level = 5, output_level = 2, klass_data = klass_data)[1:10, ]
Levels(input_level = 5, output_level = c(2, 3), klass_data = klass_data)[1:10, ]
```

---

ListFamily	<i>Classification family list Print a list of all families and the number of classifications in each</i>
------------	--

---

**Description**

Classification family list Print a list of all families and the number of classifications in each

**Usage**

```
ListFamily(family = NULL, codelists = FALSE)
```

**Arguments**

family	Input family ID number to get a list of classifications in that family
codelists	True/False for whether to include codelists. Default = FALSE

**Value**

dataset containing a list of families

**Examples**

```
ListFamily()
ListFamily(codelists=TRUE)
ListFamily(family = 1)
ListFamily(family = 1, codelists = TRUE)
```

---

ListKlass	<i>Classification list Get a full list of all classifications and codelists</i>
-----------	---

---

**Description**

Classification list Get a full list of all classifications and codelists

**Usage**

```
ListKlass(codelists = FALSE)
```

**Arguments**

`codelists` True/False for whether to include codelists. Default = FALSE

**Value**

A data frame containing a full list of classifications. The data frame includes the classification name, number, family and type.

**Examples**

```
head(ListKlass(codelists = TRUE))
```

---

MakeChar	<i>Conversion to character</i>
----------	--------------------------------

---

**Description**

Conversion to character

**Usage**

```
MakeChar(x)
```

**Arguments**

x                    a number or vector of numbers

**Value**

x converted to a string or vector of strings.

---

MakeUrl                    *Internal function to create URL address*

---

**Description**

Internal function to create URL address

**Usage**

```
MakeUrl(klass, correspond = NULL, type = "vanlig", fratil = FALSE,
        date = NULL, output_level_coding = NULL, language_coding = NULL)
```

**Arguments**

klass	Classification number
correspond	Target number for correspondence table
type	String describing type. "vanlig" for normal classification and "kor" for correspondence. Default = "vanlig"
fratil	True/False for whether a date interval is to be used. Default = False
date	Date(s) for classification
output_level_coding	Coding for output level
language_coding	Coding for language

**Value**

String url address

---

SearchKlass	<i>Search Klass</i>
-------------	---------------------

---

**Description**

Search Klass

**Usage**

```
SearchKlass(query, codelists = FALSE, size = 20)
```

**Arguments**

query	String with key word to search for
codelists	True/False for whether to include codelists. Default = FALSE
size	The number of results to show. Default = 20.

**Value**

Data frame of possible classifications that match the query

**Examples**

```
SearchKlass("yrke")
SearchKlass("yrke", codelists = TRUE)
SearchKlass("yrke", codelists = TRUE, size = 2)
SearchKlass("*fold")
```

---

splitChar	<i>insert missing dots to the right place in a string</i>
-----------	---

---

**Description**

insert missing dots to the right place in a string

**Usage**

```
splitChar(x, dot)
```

**Arguments**

x	- character with missing dots
dot	- the place of missing dots

**Value**

a string is returned with the insertion of a formatted period (.) in the specified location.

# Index

## \*Topic **datasets**

klassdata, 9

ApplyClass, 2

CheckDate, 3

ConvertTable, 3

CorrespondList, 4

formatting, 4

GetFamily, 5

GetClass, 6

GetName, 7

GetNums, 7

GetUrl, 8

GetUrl2, 8

GetVersion, 9

klassdata, 9

levelCheck, 10

Levels, 11

ListFamily, 11

ListClass, 12

MakeChar, 12

MakeUrl, 13

SearchClass, 14

splitChar, 14