

Package ‘pxR’

April 6, 2025

Type Package

Title PC-Axis with R

Version 0.42.8

Date 2025-03-24

Encoding UTF-8

Description Provides a set of functions for reading and writing PC-Axis files, used by different statistical organizations around the globe for data dissemination.

Depends stringr, reshape2, jsonlite, plyr

URL <https://github.com/cjgb/pxR>

License GPL-3

LazyLoad yes

NeedsCompilation no

Author Carlos J. Gil Bellosta [cre, aut],
Francisco J. Viciano [aut],
Oscar Perpinan Lamigueiro [aut],
Emilio Torres Manzanera [ctb]

Maintainer Carlos J. Gil Bellosta <gilbellosta@gmail.com>

Repository CRAN

Date/Publication 2025-04-06 18:20:08 UTC

Contents

as.array.px	2
as.data.frame.px	3
as.px	4
read.px	5
summary.px	7
write.json.stat	8
write.px	9

Index	11
--------------	-----------

as.array.px

Extraction of data from px objects into arrays

Description

This function extracts the data component from a px object as a array.

Usage

```
## S3 method for class 'px'  
as.array(x, use.codes = FALSE, ...)
```

Arguments

x	A px object.
use.codes	A logical value to indicate whether to use CODES instead of VALUES as names of the levels: TRUE to use CODES, default FALSE. Or a vector with the names of the variables that use CODES to identify levels
...	Additional arguments. Not used yet.

Value

An array

Author(s)

Francisco J. Viciano

References

https://www.scb.se/globalassets/vara-tjanster/px-programmen/px-file_format_specification_2013.pdf

See Also

[read.px](#), [as.data.frame.px](#)

Examples

```
my.px      <- read.px(system.file("extdata", "example.px", package = "pxR"))  
my.array  <- as.array(my.px)  
my.array2 <- as.array(my.px ,use.codes=TRUE)  
my.array3 <- as.array(my.px ,use.codes=c('municipios'))
```

as.data.frame.px *Extraction of data from px into data.frame objects*

Description

This function extracts the data component from a px object as a data.frame.

Usage

```
## S3 method for class 'px'  
as.data.frame( x, ..., use.codes = FALSE,  
              warnings.as.errors = TRUE, direction = 'long')
```

Arguments

x	a px object
use.codes	If true, this parameter replaces the names (or levels) of the variables by their codes, if available; alternatively, it can be a character vector identifying those variables for which codes will replace the levels of the variables. See details section.
warnings.as.errors	If true, the function will fail in case any issues are found; otherwise, it will generate warnings.
direction	character string, either "wide" to reshape to wide format, or "long" to reshape to long format (default).
...	Additional arguments, currently not used

Details

The PC-Axis provides two alternative mechanisms for naming variables: their values or their codes, usually less verbose. The the use.codes parameter can be used to select those variables for which the CODES attribute in the PC-Axis file for naming the levels of the variables. It should be noted that there may not be CODES for all variables; in such cases, the VALUES are used instead.

The function will operate in 'paranoid mode' unless warnings.as.errors is set to FALSE. In such mode, the function will fail if any issues are found in the data. If warnings.as.errors is set to FALSE, the function will do its best to solve some known issues in some PC-Axis files.

Value

A data.frame object.

Author(s)

Carlos J. Gil Bellosta, Oscar Perpiñán Lamigueiro, Francisco J. Viciano Fernández, Emilio Torres

References

https://www.scb.se/globalassets/vara-tjanster/px-programmen/px-file_format_specification_2013.pdf

See Also

[read.px](#), [as.array.px](#)

Examples

```
my.px.object <- read.px(system.file("extdata", "example.px",
                                  package = "pxR"))
my.px.data <- as.data.frame(my.px.object)
```

as.px

Convert an array or a data.frame into an object of class px

Description

This function converts an array object into a px object. It uses the `dimnames` of the array to fill the `VALUES` list of the px object. It can also convert a `data.frame` object into a px object. Optionally it uses an object of type `px` as the skeleton of the new object.

Usage

```
as.px(x, ... )
## S3 method for class 'array'
as.px(x, skeleton.px = NULL, list.keys = NULL, ...)
## S3 method for class 'data.frame'
as.px(x, skeleton.px = NULL,
      list.keys = NULL, value.column = NULL, ...)
```

Arguments

<code>x</code>	An array or <code>data.frame</code> with the data required to build the px object
<code>skeleton.px</code>	A px object with metadata that will be used in the new px object
<code>list.keys</code>	A optional list of metadata pairs
<code>value.column</code>	Optional name to numeric value column in <code>data.frame</code> ; defaults to "value"
<code>...</code>	Other arguments that can be passed to the function

Details

A px object is the internal representation in R of a PC-Axis file. As such, it contains both data and metadata.

The data to populate the px object comes from `x`, the first argument of the function. If `x` is an array, it needs to have 'named `dimnames`', i.e., its `dimnames` attribute needs to be a list with named entries.

x can also be a molten data.frame, i.e., each row contains a single value and the remaining columns indicate the levels of the variables it refers to. The ts names will be used in STUB and HEDING keyword of px objects. The first column of x will be used as 'HEADING', and the rest, in reverse, order as 'STUB'.

There are two compatible methods to provide metadata information. The first one is via skeleton.px, a px object whose metadata (title, etc.) is to be recycled in the new object. In addition to that, list.keys can be used to pass metadata in key-value pairs as shown in the examples below.

Metadata provided in list.keys has priority over metadata coming from skeleton.px. Also, note that the metadata items used to name the variables and their labels are neither extracted from skeleton.px nor px but from the dimnames attribute of x.

Note that the keywords 'HEADING', 'VALUES' and 'DATA', if present in skeleton.px or list.keys, are ignored. Inconsistent 'CODES' in skeleton.px will also be ignored.

Few checks are made to guarantee that metadata keys conform to the PC-Axis standard.

Value

a px object

Author(s)

Francisco J. Viciano, Carlos J. Gil Bellosta,

See Also

[write.px](#), [as.array.px](#)

Examples

```
my.px.object <- read.px(system.file("extdata", "example.px", package = "pxR"))
my.data      <- as.array(my.px.object)
my.px.object2 <- as.px(my.data)
my.px.object3 <- as.px(my.data, skeleton.px = my.px.object)
my.px.object4 <- as.px(my.data, list.keys = list(MATRIX = "xxx", CONTENTS = "new data",
                                                NEWKEY = "another key", UNITS = "people", TITLE = "My Title"))

my.px.df     <- as.data.frame(my.px.object)
my.px.object5 <- as.px(my.px.df)
```

read.px

Reads a PC-Axis file

Description

This function reads a PC-AXIS file (a text file with certain format) and creates an object of the class px.

Usage

```
read.px(filename, encoding = NULL,
        na.strings = c('"."', '".."', '"..."', '"...."', '"....."', '"....."',
                       '":'))
```

Arguments

filename	The name of the PC-Axis file to read
encoding	A character string describing the current encoding; see the Details section
na.strings	A character to be interpreted as missing value in the DATA field of the PC-AXIS file. The chapter 4 of the second reference provides details about the dot codes used in the PC-AXIS format.

Details

This function reads data files in the PC-Axis format. The format is described in the two documents in the reference section.

According to them, null values can be encoded using ".", "..", "...", or "....".

Also, the documentation is not very specific concerning the actual encoding to be expected in PC-Axis files. It is however quite safe to assume that they will be encoded using some "latin1" variant. The alternative value for the encoding argument would be "utf-8".

Value

An object of the class px, which is essentially a list comprising the fields of a PC-AXIS file (see references for details).

Author(s)

Carlos J. Gil Bellosta, Oscar Perpiñan Lamigueiro, Francisco J. Viciano Fernández

References

https://www.scb.se/globalassets/vara-tjanster/px-programmen/px-file_format_specification_2013.pdf https://datanalytics.com/uploads/tiedostomuoto2006_laaja_en.pdf

See Also

[as.data.frame.px](#), [as.array.px](#) [iconv](#)

Examples

```
my.px.object <- read.px(system.file("extdata", "example.px",
                                  package = "pxR") )
my.px.data <- as.data.frame( my.px.object )
```

`summary.px`*summary.px*

Description

Compactly display the internal structure of a px object.

Usage

```
## S3 method for class 'px'  
summary( object, ... )
```

Arguments

<code>object</code>	a px object
<code>...</code>	Additional arguments, currently not used

Value

A verbose description of data and metadata within the px object.

Author(s)

Carlos J. Gil Bellosta

References

https://www.scb.se/globalassets/vara-tjanster/px-programmen/px-file_format_specification_2013.pdf

See Also

[read.px as.data.frame.px](#)

Examples

```
my.px.object <- read.px(system.file( "extdata", "example.px", package = "pxR"))  
summary(my.px.object)
```

write.json.stat	<i>Write a px object in JSON-stat format.</i>
-----------------	---

Description

This function writes a object of class px to a file conforming the JSON-stat standard, a simple lightweight JSON dissemination format best suited for data visualization, mobile apps, or open data initiatives.

Usage

```
write.json.stat(obj.px, filename)
```

Arguments

obj.px	the name of a object of class px
filename	the name of the json-stat file to be created (suggested extension, json)

Author(s)

Carlos J. Gil Bellosta

References

<https://json-stat.org/>

See Also

[write.px](#)

Examples

```
## Not run:  
  opx1 <- read.px(system.file( "extdata", "example.px", package = "pxR"))  
  write.json.stat(opx1, file = "opx.px")  
  
## End(Not run)
```

write.px	<i>Write a PC-Axis file</i>
----------	-----------------------------

Description

This function writes an object of class px to a PC-Axis file

Usage

```
write.px(obj.px, filename, heading = NULL, stub = NULL,  
        keys = NULL , write.na = FALSE, write.zero = FALSE,  
        fileEncoding = "ISO-8859-1")
```

Arguments

obj.px	The name of a object of class px
filename	The name of the PC-Axis file to create
heading	An optional character vector with the names of variables in the HEADING part of the output file
stub	An optional character vector with the names of the variables in the STUB part of the output file
keys	An optional character vector indicating the variables used as keys
write.na	Whether to write rows with NA values (if keys are used)
write.zero	Whetehr to write rows with 0 values (if keys are used)
fileEncoding	A character string describing the encoding to use in px file (see iconv for details)

Details

The function can write either regular files or files with KEYS.

For regular files, by default, the output file will have a single variable in the HEADING part of the file (columns in the matrix). It is possible to override the default by providing the heading and stub parameters. These are optional, non-overlapping, exhaustive character vectors of names of variables in the output matrix.

It is possible to write files with KEYS which could help reduce the final file size for large and sparse datasets. In such case, it is possible to indicate whether to write rows with all values equal to 0 or NA or not (via arguments `write.zero` and `write.na`).

Author(s)

Francisco J. Viciano Fernández, Oscar Perpiñan Lamigueiro, Carlos J. Gil Bellosta

References

https://www.scb.se/globalassets/vara-tjanster/px-programmen/px-file_format_specification_2013.pdf

See Also

[read.px](#), [as.data.frame.px](#), [as.array.px](#) [iconv](#)

Examples

```
opx1 <- read.px(system.file( "extdata", "example.px", package = "pxR"))
## Not run:
write.px(opx1, file = "opx.px")
write.px(opx1, file = "opx.px",
         heading = c("sexo", "edad"),
         stub = "municipios")
write.px(opx1, filename = "opx.px",
         keys = c("municipios", "edad"))

## End(Not run)
```

Index

* **manip**

- as.array.px, [2](#)
- as.data.frame.px, [3](#)
- as.px, [4](#)
- read.px, [5](#)
- summary.px, [7](#)
- write.json.stat, [8](#)
- write.px, [9](#)

- as.array.px, [2](#), [4–6](#), [10](#)
- as.data.frame (as.data.frame.px), [3](#)
- as.data.frame.px, [2](#), [3](#), [6](#), [7](#), [10](#)
- as.px, [4](#)

- iconv, [6](#), [9](#), [10](#)

- read.px, [2](#), [4](#), [5](#), [7](#), [10](#)

- summary (summary.px), [7](#)
- summary.px, [7](#)

- write.json.stat, [8](#)
- write.px, [5](#), [8](#), [9](#)