Package: sss (via r-universe)

October 4, 2026

Version 0.2.2

License GPL-2|GPL-3

Title Import Files in the Triple-s (Standard Survey Structure) Format

Description Tools to import survey files in the '.sss' (triple-s) format. The package provides the function 'read.sss()' that reads the '.asc' (or '.csv') and '.sss' files of a triple-s survey data file. See also <https://triple-s.org/>.

URL http://andrie.github.io/sss/, https://andrie.github.io/sss/

BugReports https://github.com/andrie/sss/issues

Depends R (>= 2.11.0)

Imports utils, xml2, methods, assertthat

Suggests testthat, covr, knitr, rmarkdown, dplyr, spelling, utf8

RoxygenNote 7.2.3

Roxygen list(markdown = TRUE)

Encoding UTF-8

VignetteBuilder knitr

Language en-US

Config/testthat/edition 3

Config/pak/sysreqs libxml2-dev

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

Date/Publication 2024-12-11 13:51:12 UTC

RemoteUrl https://github.com/andrie/sss

RemoteRef HEAD

RemoteSha 9e06d3e9d41364488fd15a13054dcf077bf345d9

Contents

fast.read.fwf 2
parseSSSmetadata 3
read.sss 3
readSSSdata 4
readSSSmetadata 5

Index **6**

`fast.read.fwf` *Read fixed-width files quickly.*

Description

Experimental replacement for utils::read.fwf that runs much faster. However, it is much less flexible than `utils::read.fwf`.

Usage

```
fast.read.fwf(
  file,
  widths,
  col.names = NULL,
  colClasses = NA,
  tz = "",
  dec = ".",
  ...
)
```

Arguments

`file`	Character vector: name of file
`widths`	Numeric vector: column widths. Negative numbers mean "skip this many columns". Use an NA as the final element if there are likely to be extra characters at the end of each row after the last one that you're interested in.
`col.names`	names for the columns that are NOT skipped
`colClasses`	can be used to control type conversion; see `read.table()`. It is an optional vector whose names must be part of `col.names`. There is one extension of the `read.table()` rules: a `colClasses` string starting `POSIXct` will trigger automatic conversion to POSIXct, using the rest of the string as the format specifier.
`tz`	used in auto-conversion to `POSIXct()` when `colClasses` is set
`dec`	the character to be assumed for decimal points. Passed to `utils::type.convert()`
`...`	ignored

`parseSSSmetadata` *Parses a triple-s XML (sss) metadata file, as specified by the triple-s XML standard.*

Description

This function reads and parses a `.sss` XML metadata file as well as its associated `.asc` data file. The `.sss` standard defines a standard survey structure.

Usage

```
parseSSSmetadata(x, XMLdoc)
```

Arguments

`x`	An XML document - as returned by `xml2::read_xml()`, or `readSSSmetadata()`
`XMLdoc`	No longer used. Use `x` instead.

See Also

readSSSmetadata, read.sss, readSSSdata

`read.sss` *Reads a triple-s XML (asc) data file, as specified by the triple-s XML standard.*

Description

This function reads and parses a `.sss` XML metadata file as well as its associated `.asc` data file. The sss standard defines a standard survey structure.

Usage

```
read.sss(
  sssFilename,
  ascFilename = guess_asc_filename(sssFilename),
  sep = "_",
  verbose = FALSE
)
```

Arguments

`sssFilename`	Character string: name of .sss file containing the survey metadata
`ascFilename`	Character string: name of .asc (or .csv) file containing survey data. If this is not provided, guesses the filename using
`sep`	Character vector defining the string that separates question and subquestion labels, e.g. `c("Q_1", "Q_2")`
`verbose`	If TRUE, prints messages when reading data files. Defaults to FALSE.

Value

A data frame with one element (column) for each variable in the data set.

The resulting data.frame contains several attributes:

- `variable.labels`: a named list of value labels with one element per variable, either NULL or a named character vector
- `label.table`: a named list with one element per question. Every element is a named character string that contains the label codes for that question.

References

https://triple-s.org/

See Also

Other read functions: `readSSSdata()`, `readSSSmetadata()`

Examples

```
example <- system.file("sampledata/sample-1.sss", package = "sss")

# read.sss() automatically guesses the data file name
read.sss(system.file("sampledata/sample-1.sss", package = "sss"))

read.sss(system.file("sampledata/sample-2.sss", package = "sss"))

read.sss(system.file("sampledata/sample-3.sss", package = "sss"))
```

---

readSSSdata *Reads a triple-s XML (asc) data file, as specified by the triple-s XML standard.*

Description

This function reads an '.asc" data file.

Usage

```
readSSSdata(x, ascFilename)
```

Arguments

`x`	Name of .asc file containing the survey metadata
`ascFilename`	No longer used. Use `x` instead.

See Also

read.sss(), readSSSmetadata()

Other read functions: read.sss(), readSSSmetadata()

Examples

```
sampleRoot <- system.file("sampledata", package = "sss")
filenameSSS <- file.path(sampleRoot, "sample-1.sss")
filenameASC <- file.path(sampleRoot, "sample-1.asc")

readSSSdata(filenameSSS)
readSSSmetadata(filenameSSS)
```

---

readSSSmetadata *Reads a triple-s XML (sss) metadata file, as specified by the triple-s XML standard.*

Description

This function reads a .sss XML metadata file. The .sss standard defines a standard survey structure

Usage

```
readSSSmetadata(x, SSSfilename)
```

Arguments

x	Name of .sss file containing the survey metadata
SSSfilename	No longer used. Use x instead.

See Also

parseSSSmetadata(), read.sss(), readSSSdata()

Other read functions: read.sss(), readSSSdata()

Examples

```
sampleRoot <- system.file("sampledata", package = "sss")
filenameSSS <- file.path(sampleRoot, "sample-1.sss")
filenameASC <- file.path(sampleRoot, "sample-1.asc")

readSSSdata(filenameSSS)
readSSSmetadata(filenameSSS)
```

Index

∗ **parsing functions**
    parseSSSmetadata, 3
∗ **read functions**
    read.sss, 3
    readSSSdata, 4
    readSSSmetadata, 5

fast.read.fwf, 2

parseSSSmetadata, 3
parseSSSmetadata(), *5*
POSIXct(), *2*

read.sss, 3, *5*
read.sss(), *5*
read.table(), *2*
readSSSdata, 4, *4, 5*
readSSSdata(), *5*
readSSSmetadata, *4*, 5, *5*
readSSSmetadata(), *3*, *5*

utils::read.fwf, *2*
utils::type.convert(), *2*

xml2::read_xml(), *3*