Package 'asciiSetupReader'

October 23, 2026

Type Package

Title Reads Fixed-Width ASCII Data Files (.txt or .dat) that Have Accompanying Setup Files (.sps or .sas)

Version 3.0.0

Description Lets you open a fixed-width ASCII file (.txt or .dat) that has an accompanying setup file (.sps or .sas). These file combinations are sometimes referred to as .txt+.sps, .txt+.sas, .dat+.sps, or .dat+.sas. This will only run in a txt-sps or txt-sas pair in which the setup file contains instructions to open that text file. It will NOT open other text files, .sav, .sas, or .por data files. Fixed-width ASCII files with setup files are common in older (pre-2000) government data.

License MIT + file LICENSE

URL https://jacobkap.github.io/asciiSetupReader/, https://github.com/jacobkap/asciiSetupReader

BugReports https://github.com/jacobkap/asciiSetupReader/issues

Depends R (>= 4.1.0)

Imports data.table, readr, vroom (>= 1.1.0), stringr, shiny (>= 0.13), miniUI (>= 0.1.1), rstudioapi (>= 0.5)

Suggests covr, knitr, rmarkdown, testthat (>= 3.1.7)

VignetteBuilder knitr

Encoding UTF-8

Language en-US

Config/roxygen2/version 8.1.1

Config/testthat/edition 3

NeedsCompilation no

Author Jacob Kaplan [aut, cre] (ORCID: <https://orcid.org/0000-0002-0601-0387>)

Maintainer Jacob Kaplan <jkkaplan6@gmail.com>

Repository CRAN

Date/Publication 2026-10-22 23:10:02 UTC

Contents

make_sps_setup 2
parse_setup 3
read_ascii_setup 4
read_ascii_setup_addin 7
sas_ascii_reader 8
spss_ascii_reader 10

Index **12**

`make_sps_setup` *Create an SPSS setup file (.sps) to use for reading in fixed-width text files*

Description

make_sps_setup() is used to create the setup file used in reading in fixed-width text files. Often the setup file comes with the data file but in some cases (usually with government data) you will need to create the setup file yourself.

Usage

```
make_sps_setup(
  file_name,
  col_positions,
  col_names = NULL,
  col_labels = NULL,
  value_labels = NULL,
  missing_values = NULL
)
```

Arguments

`file_name`	Name (or path) of the file to be saved (e.g. "setup_file1"). The .sps extension is added if it isn't already there.
`col_positions`	Either a vector of strings indicating the start and end position of each column (e.g. "1-3", "4-5") or a vector of the widths of the columns (e.g. 3, 2).
`col_names`	A vector of names for the columns. If none are provided, will automatically create names based on column number (e.g. V1, V2, V3).
`col_labels`	A vector of labels for the columns. These are often longer and more descriptive than the col_names. These are the values used as column names if use_clean_names = TRUE in `read_ascii_setup`.
`value_labels`	A vector with the value first, then an ' = ' then the label. Each new column should have the column name (or label) followed by ' ='.
`missing_values`	A vector of strings with the column name or label followed by the values to be replaced by NA, e.g. c("var1", "9", "-8", "var2", "-8").

Value

NULL, invisibly. Called to write the .sps file.

Examples

```
value_labels <- c("var1 = ",
                  "1 = label 1",
                  "2 = label 2",
                  "3 = label 3",
                  "4 = label 4",
                  "5 = label 5",
                  "var3 = ",
                  "1A = alpha",
                  "1B = bravo",
                  "1C = cat")
missing_values <- c("state name", "9", "-8", "county", "-8")
file_name <- file.path(tempdir(), "example_name")
make_sps_setup(file_name      = file_name,
               col_positions  = c(1, 3, 4, 2),
               col_names      = c("var1", "var2", "var3", "var4"),
               col_labels     = c("state name", "county",
                                  "population", "census region code"),
               value_labels   = value_labels,
               missing_values = missing_values)

# The new setup file can be used with parse_setup() and read_ascii_setup()
parse_setup(paste0(file_name, ".sps"))
```

---

parse_setup *Parse the setup file (.sps or .sas)*

Description

Parse the setup file (.sps or .sas)

Usage

```
parse_setup(setup_file)
```

Arguments

setup_file	Name of the SPSS or SAS setup file - should be a .sps or .sas (.txt also accepted as are these files in zipped format). Whether it is SPSS or SAS is determined from the file name: names ending in .sps (optionally followed by .zip or .txt) are SPSS and everything else is SAS. A setup file that isn't valid UTF-8 text is read as Windows-1252 (Latin-1), with a message.

Value

A list of length 3. The first object ("setup") is a data frame with 5 columns: column_number, the non-descriptive name of each column (e.g. V1); column_name, its descriptive name; begin and end, the first and last character position of the column in the fixed-width data file; and decimals, the number of implied decimal places the setup file gives the column (0 if none; see read_ascii_setup). Descriptive names are made unique: a name already used by an earlier column (in any case) gets the column's short name added, e.g. the second "CLOTHING" column, V93, is "CLOTHING_V93".

The second object ("value_labels") is a list of named vectors for the value labels, one per column with value labels. The vector's values are the codes in the data and its names are the labels. If there are no value labels, this will be NULL.

The third object ("missing") is a data.frame with three columns. The first column (variable) says the column name, the second (values) says the value that is missing and will be replaced with NA, and the third (commented) is TRUE if the setup file has that missing value commented out. If there are no missing values, this will be NULL.

Examples

```
sas_name <- system.file("extdata", "example_setup.sas",
                        package = "asciiSetupReader")
sas_example <- parse_setup(sas_name)

sps_name <- system.file("extdata", "example_setup.sps",
                        package = "asciiSetupReader")
sps_example <- parse_setup(sps_name)
head(sps_example$setup)
```

---

read_ascii_setup *Read fixed-width ASCII file using SPSS or SAS Setup file*

Description

read_ascii_setup() is used when you need to read a fixed-width ASCII (text) file that comes with a setup file. The setup file provides instructions on how to create and name the columns, and fix the key-value pairs (sometimes called value labels). This is common in government data, particularly data produced before 2010.

Usage

```
read_ascii_setup(
  data,
  setup_file,
  use_value_labels = TRUE,
  use_clean_names = TRUE,
  select_columns = NULL,
  coerce_numeric = TRUE,
  missing_values = c("all", "active", "none")
)
```

Arguments

`data`	Name of the ASCII (.txt or .dat) file that contains the data. This file may be zipped with a file extension of .zip.
`setup_file`	Name of the SPSS or SAS setup file - should be a .sps or .sas (.txt also accepted as are these files in zipped format). See `parse_setup` for how the type is determined.
`use_value_labels`	If TRUE, fixes value labels of the data. e.g. If a column is "sex" and has values of 0 or 1, and the setup file says 0 = male and 1 = female, it will make that change.
`use_clean_names`	If TRUE fixes column names from default column name in the setup file (e.g. V1, V2) to the descriptive label for the column provided in the file (e.g. age, sex, etc.).
`select_columns`	Specify which columns from the dataset you want. If NULL, will return all columns. Accepts the column number (e.g. 1:5), column name (e.g. V1, V2, etc.) or column label (e.g. VICTIM_NAME, CITY, etc.).
`coerce_numeric`	If TRUE (default) will make columns where all values can be made numeric into numeric columns. Every value is checked, so a column with any text value stays character. Columns the setup file says are text (SPSS "(A)", SAS "$") always stay character, so their leading zeros are kept. Useful as FALSE if other variables have leading zeros - such as US Census FIPS codes given as numbers. Columns that the setup file gives implied decimal places (e.g. `V7 37-43 (1)` in SPSS or `V7 37-43 .1` in SAS) get their decimal point whatever coerce_numeric is: a value of "0000785" with 1 implied decimal place becomes 78.5, or the text "000078.5" if coerce_numeric is FALSE. Values that already have a decimal point are kept as they are.
`missing_values`	Which of the setup file's missing values to set to NA. Setup files often have their missing values section commented out (e.g. "* MISSING VALUES" in SPSS or inside `/* */` in SAS) so that the codes stay in the data unless the user chooses otherwise. **"all"** (default) Use every missing value in the setup file, ignoring whether it is commented out. This is what earlier versions did. **"active"** Use only the missing values that aren't commented out, as SPSS and SAS would. **"none"** Don't use any missing values; every code is kept. Missing values are matched by value, as in SPSS and SAS, so a missing value written 9 also matches 09 or 0000009 in the data, except in columns the setup file says are text (SPSS "(A)", SAS "$"), where it must be the same text. Value labels are applied first, so a missing value that also has a value label (e.g. 9 = "Unknown") keeps its label rather than becoming NA unless `use_value_labels = FALSE`.

Value

data.frame of the data from the ASCII file

Value labels

As in SPSS and SAS, codes of number columns are matched by value (a label for 0 labels 0000000), and codes of text columns (SPSS "(A)", SAS "$") must be the same text; a blank or the letters NA in a text column can have a label. As in SPSS, a VALUE LABELS command replaces any labels an earlier VALUE LABELS command gave a column, and ADD VALUE LABELS adds to them (changing the label of a code that already has one).

Recodes

Only a setup file's layout, names, value labels and missing values are used. Statements that change values (SAS `IF v = 1 THEN v = 5;`, SPSS `RECODE v (1=5)`) aren't applied, and a message lists them.

Blank values

A blank, a lone "." or the letters NA in a number column is NA. A blank or the letters NA in a text column is NA unless it has a value label (and `use_value_labels` is TRUE), which is used instead.

Messages

read_ascii_setup() tells you with a message when it:

- adds implied decimal places to a column's values (listing the columns and their decimal places),
- finds values in such a column that equal one of its codes (value labels or missing values) only before the decimal places are applied, which suggests the data doesn't match the setup file,
- finds SAS value labels for ranges of values or OTHER, which aren't applied (only when `use_value_labels = TRUE`),
- sets values to NA because of missing values that are commented out in the setup file (only when `missing_values = "all"`),
- makes column names unique because two columns would have had the same name (only when `use_clean_names = TRUE`),
- uses only the last of several different sets of value labels a column has (only when `use_value_labels = TRUE`),
- skips SAS missing value statements it can't read (only when `missing_values` isn't "none"),
- finds statements that change values (recodes), which aren't applied,
- selects, by `select_columns`, the first of several columns that had the same name (saying what the others are called now), or
- reads a setup file that isn't valid UTF-8 as Windows-1252.

`suppressMessages(read_ascii_setup(...))` silences them. They have the class "asciiSetupReader_message", so `suppressMessages(read_ascii_setup(...), classes = "asciiSetupReader_message")` silences only these messages. Each kind also has its own class: "asciiSetupReader_implied_decimals", "asciiSetupReader_unscaled_codes", "asciiSetupReader_label_ranges", "asciiSetupReader_commented_missing", "asciiSetupReader_unique_names", "asciiSetupReader_replaced_labels", "asciiSetupReader_skipped_missing", "asciiSetupReader_recodes", "asciiSetupReader_select_columns" and "asciiSetupReader_encoding".

Errors

Setup files that can't be read stop with an error of class "asciiSetupReader_error" and a more specific class: "asciiSetupReader_delimited" (a file for delimited, not fixed-width, data), "asciiSetupReader_no_positions" (no column positions, e.g. a file of value labels only) or "asciiSetupReader_unsupported_input" (an SPSS DATA LIST or SAS INPUT statement this package can't read, e.g. one with several records per case). A `select_columns` that names or numbers columns the setup file doesn't have stops with an error of class "asciiSetupReader_select_columns".

Examples

```
# Text file is zipped to save space.
dataset_name <- system.file("extdata", "example_data.zip",
  package = "asciiSetupReader")
sps_name <- system.file("extdata", "example_setup.sps",
  package = "asciiSetupReader")

example <- read_ascii_setup(data = dataset_name,
  setup_file = sps_name)

# Does not fix value labels
example2 <- read_ascii_setup(data = dataset_name,
  setup_file = sps_name, use_value_labels = FALSE)

# Keeps original column names
example3 <- read_ascii_setup(data = dataset_name,
  setup_file = sps_name, use_clean_names = FALSE)

# Only returns the first 5 columns
example4 <- read_ascii_setup(data = dataset_name,
  setup_file = sps_name, select_columns = 1:5)

# Only uses the missing values that aren't commented out
example5 <- read_ascii_setup(data = dataset_name,
  setup_file = sps_name, missing_values = "active")
```

---

read_ascii_setup_addin

Launch an RStudio addin to select options for read_ascii_setup()

Description

Launch an RStudio addin to select options for read_ascii_setup().

Usage

```
read_ascii_setup_addin()
```

Value

read_ascii_setup() code to console with options based on user input

Examples

```
if (interactive()) {
  read_ascii_setup_addin()
}
```

`sas_ascii_reader` *Read fixed-width ASCII file using SAS Setup file*

Description

sas_ascii_reader() and spss_ascii_reader() are used when you need to read a fixed-width ASCII (text) file that comes with a setup file. These file combinations are sometimes referred to as .txt+.sps, .txt+.sas, .dat+.sps, or .dat+.sas. The setup file provides instructions on how to create and name the columns, and fix the key-value pairs (sometimes called value labels). This is common in government data, particularly data produced before 2010.

Usage

```
sas_ascii_reader(
  dataset_name,
  sas_name,
  value_label_fix = TRUE,
  real_names = TRUE,
  keep_columns = NULL,
  coerce_numeric = TRUE,
  missing_values = c("all", "active", "none")
)
```

Arguments

`dataset_name`	Name of the ASCII (.txt) file that contains the data. This file may be zipped with a file extension of .zip.
`sas_name`	Name of the SAS Setup file - should be a .sas or .txt file.
`value_label_fix`	If TRUE, fixes value labels of the data. e.g. If a column is "sex" and has values of 0 or 1, and the setup file says 0 = male and 1 = female, it will make that change.
`real_names`	If TRUE fixes column names from default column name in the SPSS setup file (e.g. V1, V2) to the name it says the column is called (e.g. age, sex, etc.).
`keep_columns`	Specify which columns from the dataset you want. If NULL, will return all columns. Accepts the column number (e.g. 1:5), column name (e.g. V1, V2, etc.) or column label (e.g. VICTIM_NAME, CITY, etc.).
`coerce_numeric`	If TRUE (default) will make columns where all values can be made numeric into numeric columns. Every value is checked, so a column with any text value stays character. Useful as FALSE if variables have leading zeros - such as US Census FIPS codes. See `read_ascii_setup` for how implied decimal places are handled.

`missing_values` Which of the setup file's missing values to set to NA. Setup files often have their missing values section commented out (e.g. "* MISSING VALUES" in SPSS or inside `/* */` in SAS) so that the codes stay in the data unless the user chooses otherwise.

"all" (default) Use every missing value in the setup file, ignoring whether it is commented out. This is what earlier versions did.

"active" Use only the missing values that aren't commented out, as SPSS and SAS would.

"none" Don't use any missing values; every code is kept.

Missing values are matched by value, as in SPSS and SAS, so a missing value written 9 also matches 09 or 0000009 in the data, except in columns the setup file says are text (SPSS "(A)", SAS "$"), where it must be the same text. Value labels are applied first, so a missing value that also has a value label (e.g. 9 = "Unknown") keeps its label rather than becoming NA unless `use_value_labels = FALSE`.

Details

sas_ascii_reader() gives the same result as `read_ascii_setup`, which reads both SPSS and SAS setup files; its arguments just have older names.

Value

Data.frame of the data from the ASCII file

See Also

`read_ascii_setup`, `spss_ascii_reader` For using an SPSS setup file

Other ASCII Reader functions: `spss_ascii_reader()`

Examples

```
# Text file is zipped to save space.
dataset_name <- system.file("extdata", "example_data.zip",
  package = "asciiSetupReader")
sas_name <- system.file("extdata", "example_setup.sas",
  package = "asciiSetupReader")

example <- sas_ascii_reader(dataset_name = dataset_name,
  sas_name = sas_name)

# Only returns the first 5 columns
example2 <- sas_ascii_reader(dataset_name = dataset_name,
  sas_name = sas_name, keep_columns = 1:5)
```

spss_ascii_reader *Read fixed-width ASCII file using SPSS Setup file*

Description

spss_ascii_reader() and sas_ascii_reader() are used when you need to read a fixed-width ASCII (text) file that comes with a setup file. These file combinations are sometimes referred to as .txt+.sps, .txt+.sas, .dat+.sps, or .dat+.sas. The setup file provides instructions on how to create and name the columns, and fix the key-value pairs (sometimes called value labels). This is common in government data, particularly data produced before 2010.

Usage

```
spss_ascii_reader(
  dataset_name,
  sps_name,
  value_label_fix = TRUE,
  real_names = TRUE,
  keep_columns = NULL,
  coerce_numeric = TRUE,
  missing_values = c("all", "active", "none")
)
```

Arguments

dataset_name	Name of the ASCII (.txt) file that contains the data. This file may be zipped with a file extension of .zip.
sps_name	Name of the SPSS Setup file - should be a .sps or .txt (zipped text files also work) file.
value_label_fix	If TRUE, fixes value labels of the data. e.g. If a column is "sex" and has values of 0 or 1, and the setup file says 0 = male and 1 = female, it will make that change.
real_names	If TRUE fixes column names from default column name in the SPSS setup file (e.g. V1, V2) to the name it says the column is called (e.g. age, sex, etc.).
keep_columns	Specify which columns from the dataset you want. If NULL, will return all columns. Accepts the column number (e.g. 1:5), column name (e.g. V1, V2, etc.) or column label (e.g. VICTIM_NAME, CITY, etc.).
coerce_numeric	If TRUE (default) will make columns where all values can be made numeric into numeric columns. Every value is checked, so a column with any text value stays character. Useful as FALSE if variables have leading zeros - such as US Census FIPS codes. See read_ascii_setup for how implied decimal places are handled.
missing_values	Which of the setup file's missing values to set to NA. Setup files often have their missing values section commented out (e.g. "* MISSING VALUES" in SPSS or

inside `/* */` in SAS) so that the codes stay in the data unless the user chooses otherwise.

"all" (default) Use every missing value in the setup file, ignoring whether it is commented out. This is what earlier versions did.

"active" Use only the missing values that aren't commented out, as SPSS and SAS would.

"none" Don't use any missing values; every code is kept.

Missing values are matched by value, as in SPSS and SAS, so a missing value written 9 also matches 09 or 0000009 in the data, except in columns the setup file says are text (SPSS "(A)", SAS "$"), where it must be the same text. Value labels are applied first, so a missing value that also has a value label (e.g. 9 = "Unknown") keeps its label rather than becoming NA unless `use_value_labels = FALSE`.

Details

spss_ascii_reader() gives the same result as `read_ascii_setup`, which reads both SPSS and SAS setup files; its arguments just have older names.

Value

Data.frame of the data from the ASCII file

See Also

`read_ascii_setup`, `sas_ascii_reader` For using an SAS setup file

Other ASCII Reader functions: `sas_ascii_reader()`

Examples

```
# Text file is zipped to save space.
dataset_name <- system.file("extdata", "example_data.zip",
  package = "asciiSetupReader")
sps_name <- system.file("extdata", "example_setup.sps",
  package = "asciiSetupReader")

example <- spss_ascii_reader(dataset_name = dataset_name,
  sps_name = sps_name)

# Only returns the first 5 columns
example4 <- spss_ascii_reader(dataset_name = dataset_name,
  sps_name = sps_name, keep_columns = 1:5)
```

Index

∗ **ASCII Reader functions**
    sas_ascii_reader, 8
    spss_ascii_reader, 10

make_sps_setup, 2

parse_setup, 3, *5*

read_ascii_setup, *2*, *4*, 4, *8–11*
read_ascii_setup_addin, 7

sas_ascii_reader, 8, *11*
sas_ascii_reader(), *11*
spss_ascii_reader, *9*, 10
spss_ascii_reader(), *9*