The Data Frame Reference Manual

Data frames for Common Lisp, version 2.0

Steve Nunez <steve@symbolics.tech>

Table of Contents

1 Systems

The main system appears first, followed by any subsystem dependency.

1.1 data-frame

Author Steve Nunez <steve@symbolics.tech>

Source Control
(:git "https://github.com/lisp-stat/data-frame.git")

Bug Tracker
https://github.com/Lisp-Stat/data-frame/issues

License MS-PL

Description
Data frames for Common Lisp

Long Description
A data manipulation package, conceptually similar to R's data.frame

Version 2.0

Dependencies
- alexandria
- alexandria+
- anaphora
- array-operations
- num-utils
- select
- let-plus

Source [data-frame.asd], page 3, (file)

Directory s:/src/data-frame/

Components
- [pkgdcl.lisp], page 3, (file)
- [utils.lisp], page 3, (file)
- [data-frame.lisp], page 3, (file)
- [pprint.lisp], page 5, (file)
- [formatted-output.lisp], page 5, (file)
- [summary.lisp], page 6, (file)
- [defdf.lisp], page 7, (file)
- [properties.lisp], page 8, (file)
- [missing.lisp], page 8, (file)

2 Files

Files are sorted by type and then listed depth-first from the systems components trees.

2.1 Lisp

2.1.1 data-frame.asd

Location /src/data-frame/data-frame.asd

Systems [data-frame], page 1, (system)

2.1.2 data-frame/pkgdcl.lisp

Parent [data-frame], page 1, (system)

Location pkgdcl.lisp

Packages [data-frame], page 9,

2.1.3 data-frame/utils.lisp

Dependency
[pkgdcl.lisp], page 3, (file)

Parent [data-frame], page 1, (system)

Location utils.lisp

Exported Definitions
[column-type], page 16, (function)

Internal Definitions

- [get-type], page 30, (function)
- [types-in-column], page 34, (function)

2.1.4 data-frame/data-frame.lisp

Dependency
[utils.lisp], page 3, (file)

Parent [data-frame], page 1, (system)

Location data-frame.lisp

Exported Definitions

- [add-column!], page 15, (function)
- [add-columns], page 16, (function)
- [add-columns!], page 16, (function)
- [alist-df], page 16, (function)
- [alist-dv], page 16, (function)
- [column], page 16, (function)
- [(setf column)], page 16, (function)
- [column-names], page 16, (function)
- [columns], page 16, (function)
- [copy], page 17, (function)
- [count-rows], page 17, (function)

- [make-ordered-keys], page 31, (function)
- [ordered-keys], page 32, (function)
- [ordered-keys], page 36, (structure)
- [ordered-keys-p], page 32, (function)
- [ordered-keys-table], page 32, (function)
- [plist-data], page 32, (function)

2.1.5 data-frame/pprint.lisp

Dependency
[data-frame.lisp], page 3, (file)

Parent [data-frame], page 1, (system)

Location pprint.lisp

Exported Definitions
- [head], page 21, (method)
- [pprint-array], page 19, (function)
- [pprint-data-frame], page 19, (function)
- [tail], page 22, (method)

Internal Definitions
- [*max-digits*], page 26, (special variable)
- [*row-numbers-p*], page 26, (special variable)
- [2d-array-to-list], page 27, (function)
- [column-type-format], page 28, (function)
- [default-column-formats], page 36, (method)
- [max-decimal], page 31, (function)
- [max-width], page 31, (function)
- [printer-status], page 32, (function)
- [reverse-df], page 33, (function)

2.1.6 data-frame/formatted-output.lisp

Dependency
[pprint.lisp], page 5, (file)

Parent [data-frame], page 1, (system)

Location formatted-output.lisp

Exported Definitions
- [df-print], page 17, (function)
- [pprint-markdown], page 19, (function)

Internal Definitions
- [aesthetic-string], page 27, (function)
- [print-table], page 32, (function)
- [weave], page 35, (function)

2.1.7 data-frame/summary.lisp

Dependency
[formatted-output.lisp], page 5, (file)

Parent [data-frame], page 1, (system)

Location summary.lisp

Exported Definitions

- [*distinct-maximum*], page 15, (special variable)
- [*distinct-threshold*], page 15, (special variable)
- [*quantile-threshold*], page 15, (special variable)
- [*summary-minimum-length*], page 15, (special variable)
- [bit-variable-summary], page 23, (structure)
- [factor-variable-summary], page 23, (structure)
- [generic-variable-summary], page 24, (structure)
- [get-summaries], page 17, (function)
- [real-variable-summary], page 24, (structure)
- [summarize-column], page 20, (function)
- [summary], page 22, (method)

Internal Definitions

- [bit-variable-summary-count], page 27, (function)
- [bit-variable-summary-desc], page 27, (function)
- [bit-variable-summary-length], page 27, (function)
- [bit-variable-summary-missing], page 27, (function)
- [bit-variable-summary-name], page 27, (function)
- [bit-variable-summary-p], page 28, (function)
- [column-length], page 35, (generic function)
- [column-length], page 35, (method)
- [copy-bit-variable-summary], page 28, (function)
- [copy-factor-variable-summary], page 28, (function)
- [copy-generic-variable-summary], page 28, (function)
- [copy-real-variable-summary], page 28, (function)
- [copy-variable-summary%], page 28, (function)
- [distinct], page 28, (function)
- [ensure-not-ratio], page 29, (function)
- [factor-variable-summary-desc], page 29, (function)
- [factor-variable-summary-element-count-alist], page 29, (function)
- [factor-variable-summary-length], page 29, (function)
- [factor-variable-summary-missing], page 29, (function)
- [factor-variable-summary-name], page 29, (function)
- [factor-variable-summary-p], page 29, (function)
- [generic-variable-summary-desc], page 29, (function)
- [generic-variable-summary-element-count-alist], page 29, (function)
- [generic-variable-summary-length], page 30, (function)

- [generic-variable-summary-missing], page 30, (function)
- [generic-variable-summary-name], page 30, (function)
- [generic-variable-summary-p], page 30, (function)
- [generic-variable-summary-quantiles], page 30, (function)
- [make-bit-variable-summary], page 31, (function)
- [make-factor-variable-summary], page 31, (function)
- [make-generic-variable-summary], page 31, (function)
- [make-real-variable-summary], page 31, (function)
- [make-variable-summary%], page 31, (function)
- [monotonicp], page 32, (function)
- [print-count-and-percentage], page 32, (function)
- [real-variable-summary-desc], page 33, (function)
- [real-variable-summary-length], page 33, (function)
- [real-variable-summary-max], page 33, (function)
- [real-variable-summary-mean], page 33, (function)
- [real-variable-summary-min], page 33, (function)
- [real-variable-summary-missing], page 33, (function)
- [real-variable-summary-name], page 33, (function)
- [real-variable-summary-p], page 33, (function)
- [real-variable-summary-q25], page 33, (function)
- [real-variable-summary-q50], page 33, (function)
- [real-variable-summary-q75], page 33, (function)
- [summarize-factor-variable], page 34, (function)
- [summarize-generic-variable], page 34, (function)
- [summarize-real-variable], page 34, (function)
- [variable-summary%], page 36, (structure)
- [variable-summary%-desc], page 35, (function)
- [variable-summary%-length], page 35, (function)
- [variable-summary%-missing], page 35, (function)
- [variable-summary%-name], page 35, (function)
- [variable-summary%-p], page 35, (function)

2.1.8 data-frame/defdf.lisp

Dependency
[summary.lisp], page 6, (file)

Parent [data-frame], page 1, (system)

Location defdf.lisp

Exported Definitions

- [defdf], page 15, (macro)
- [define-column-names], page 17, (function)
- [replace-key!], page 15, (macro)
- [show-data-frames], page 20, (function)
- [undef], page 21, (function)

Internal Definitions

- [*ask-on-redefine*], page 26, (special variable)
- [*data-frames*], page 26, (special variable)
- [show-symbols], page 34, (function)
- [undef-column-names], page 34, (function)

2.1.9 data-frame/properties.lisp

Dependency

[defdf.lisp], page 7, (file)

Parent [data-frame], page 1, (system)

Location properties.lisp

Exported Definitions

- [heuristicate-types], page 18, (function)
- [set-properties], page 20, (function)

Internal Definitions

- [show-properties], page 34, (function)
- [sym-mac], page 34, (function)
- [var-name], page 34, (function)

2.1.10 data-frame/missing.lisp

Dependency

[properties.lisp], page 8, (file)

Parent [data-frame], page 1, (system)

Location missing.lisp

Exported Definitions

- [drop-missing], page 21, (method)
- [missingp], page 21, (generic function)
- [missingp], page 21, (method)
- [missingp], page 21, (method)
- [missingp], page 21, (method)
- [missingp], page 21, (method)
- [missingp], page 21, (method)
- [missingp], page 21, (method)
- [missingp], page 21, (method)
- [replace-missing], page 22, (method)

Internal Definitions

[drop-na], page 28, (function)

3 Packages

Packages are listed by definition order.

3.1 data-frame

Source [pkgdcl.lisp], page 3, (file)

Nickname df

Use List

- alexandria+
- select-dev
- select
- let-plus
- anaphora
- alexandria
- common-lisp

Used By List

- lisp-stat
- dfio

Exported Definitions

- [*distinct-maximum*], page 15, (special variable)
- [*distinct-threshold*], page 15, (special variable)
- [*quantile-threshold*], page 15, (special variable)
- [*summary-minimum-length*], page 15, (special variable)
- [add-column!], page 15, (function)
- [add-columns], page 16, (function)
- [add-columns!], page 16, (function)
- [alist-df], page 16, (function)
- [alist-dv], page 16, (function)
- [bit-variable-summary], page 23, (structure)
- [column], page 16, (function)
- [(setf column)], page 16, (function)
- [column-names], page 16, (function)
- [column-type], page 16, (function)
- [columns], page 16, (function)
- [copy], page 17, (function)
- [count-rows], page 17, (function)
- [data-frame], page 25, (class)
- [data-vector], page 26, (class)
- [defdf], page 15, (macro)
- [define-column-names], page 17, (function)
- [df], page 17, (function)
- [df-print], page 17, (function)

Internal Definitions

4 Definitions

Definitions are sorted by export status, category, package, and then by lexicographic order.

4.1 Exported definitions

4.1.1 Special variables

`*distinct-maximum*` [Special Variable]
If a string/factor variable has > *distinct-maximum* values, exclude it

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`*distinct-threshold*` [Special Variable]
If an integer variable has <= discrete values, consider it a factor

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`*quantile-threshold*` [Special Variable]
If the number of unique reals exceeds this threshold, they will be summarized with quantiles, otherwise print frequency table

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`*summary-minimum-length*` [Special Variable]
Columns are only summarised when longer than this, otherwise they are returned as is.

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

4.1.2 Macros

`defdf` *DF BODY* **&optional** *DOCUMENTATION* [Macro]

Package [`data-frame`], page 9,

Source [`defdf.lisp`], page 7, (file)

`replace-key!` *DF NEW OLD* [Macro]
Replace a key in DF, updating data package symbols Example: (replace-key! mtcars row-name x1)

Package [`data-frame`], page 9,

Source [`defdf.lisp`], page 7, (file)

4.1.3 Functions

`add-column!` *DATA KEY COLUMN* [Function]
Modify DATA (a data-frame or data-vector) by adding COLUMN with KEY. Return DATA.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`add-columns` *DATA* **&rest** *KEYS-AND-COLUMNS* [Function]
Return a new data-frame or data-vector with keys and columns added. Does not modify DATA (see README about accepted argument formats).

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`add-columns!` *DATA* **&rest** *KEYS-AND-COLUMNS* [Function]
Modify DATA (a data-frame or data-vector) by adding columns with keys (see README about accepted argument formats).

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`alist-df` *ALIST* [Function]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`alist-dv` *ALIST* [Function]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`column` *DATA KEY* [Function]
Return column corresponding to key.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

Writer [`(setf column)`], page 16, (function)

`(setf column)` *COLUMN DATA KEY* [Function]
Set column corresponding to key.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

Reader [`column`], page 16, (function)

`column-names` *DF* [Function]
Return a list of column names in DF, as strings

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`column-type` *COL* [Function]
Return the most specific type found in COL

Package [`data-frame`], page 9,

Source [`utils.lisp`], page 3, (file)

`columns` *DATA* **&optional** *SLICE* [Function]
Return the columns of DATA as a vector, or a selection if given (keys are resolved).

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`copy` *DATA* **&key** *KEY* [Function]
Copy data frame or vector. Keys are copied (and thus can be modified), columns or elements are copyied using KEY, making the default give a shallow copy.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`count-rows` *DATA-FRAME KEYS PREDICATE* [Function]
Count the number of rows for which PREDICATE called on the columns corresponding to KEYS returns non-NIL.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`define-column-names` *DF PACKAGE* [Function]
Create a symbol macro for each column name in DF
After running this function, you can refer to a column by its name. This is useful if the column names of a data frame have changed. Example: (define-column-names mtcars)

Package [`data-frame`], page 9,

Source [`defdf.lisp`], page 7, (file)

`df` **&rest** *PLIST-OR-ALIST* [Function]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`df-print` *DF* [Function]
Print DF to *standard-output* in table format

Package [`data-frame`], page 9,

Source [`formatted-output.lisp`], page 5, (file)

`df-remove-duplicates` *DATA* [Function]
Return a modified copy of DATA from which any element (row, if a DATA-FRAME) that matches another element has been removed

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`do-rows` *DATA-FRAME KEYS FUNCTION* [Function]
Traverse rows from first to last, calling FUNCTION on the columns corresponding to KEYS. Return no values.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`dv` **&rest** *PLIST-OR-ALIST* [Function]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`get-summaries` *DF* [Function]
Return a list of summaries of the variables in DF

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`heuristicate-types` *DF* [Function]

Coerce each element of the column vectors to the most specific type in the column
Often when reading in a data set, the types will be inconsistent in a variable. For example one observation might be 5.1, and another 5. Whilst mathmatically equivalent, we want our variable vectors to have identical types. The COLUMN-TYPE function returns the most specific numeric type in the column, then coerce all the vector elements to this type

Package [`data-frame`], page 9,

Source [`properties.lisp`], page 8, (file)

`keys` *DATA* [Function]

Vector of keys.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`make-df` *KEYS COLUMNS* [Function]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`make-dv` *KEYS COLUMNS* [Function]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`map-columns` *DATA FUNCTION* **&optional** *RESULT-CLASS* [Function]

Map columns of DATA-FRAME or DATA-VECTOR using FUNCTION. The result is a new DATA-FRAME with the same keys.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`map-df` *DATA-FRAME KEYS FUNCTION RESULT-KEYS* [Function]

Map DATA-FRAME to another one by rows. Function is called on the columns corresponding to KEYS, and should return a sequence with the same length as RESULT-KEYS, which give the keys of the resulting data frame. RESULT-KETS should be either symbols, or of the format (symbol &optional (element-type t)).

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`map-rows` *DATA-FRAME KEYS FUNCTION* **&key** *ELEMENT-TYPE* [Function]

Map rows using FUNCTION, on the columns corresponding to KEYS. Return the result with the given ELEMENT-TYPE.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`mask-rows` *DATA-FRAME KEYS PREDICATE* [Function]

Return a bit-vector containing the result of calling PREDICATE on rows of the columns corresponding to KEYS (0 for NIL, 1 otherwise).

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`matrix-df` *KEYS MATRIX* [Function]
Convert a matrix to a data-frame with the given keys.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`plist-df` *PLIST* [Function]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`plist-dv` *PLIST* [Function]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`pprint-array` *ARR* **&optional** *STREAM ROW-NUMBERS-P* [Function]
Print an array to STREAM, defaulting to *standard-output*, in a tabular format. If ROW-NUMBERS-P, print row numbers.

Package [`data-frame`], page 9,

Source [`pprint.lisp`], page 5, (file)

`pprint-data-frame` *DATA-FRAME* **&optional** *STREAM ROW-NUMBERS-P MAX-DIGITS* [Function]
Print DATA-FRAME to STREAM using the pretty printer

Package [`data-frame`], page 9,

Source [`pprint.lisp`], page 5, (file)

`pprint-markdown` *DF* **&key** *STREAM ROW-NUMBERS* [Function]
Print data frame DF, in markdown format, to STREAM
If ROW-NUMBERS is true, also print row numbers as the first column

Package [`data-frame`], page 9,

Source [`formatted-output.lisp`], page 5, (file)

`remove-columns` *DATA KEYS* [Function]
ARGS: DATA data frame
KEYS list of keys (variables) to be removed
Return a new data-frame or data-vector with keys and columns removed. Does not modify DATA.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`replace-column` *DATA KEY FUNCTION-OR-COLUMN* **&key** *ELEMENT-TYPE* [Function]
Create a new data frame with new column KEY from data-frame DATA by replacing it either with the given column, or applying the function to the current values (ELEMENT-TYPE is used.)

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`replace-column!` *DATA KEY FUNCTION-OR-COLUMN* **&key** *ELEMENT-TYPE* [Function]

Modify column KEY of data-frame DATA by replacing it either with the given column, or applying the function to the current values (ELEMENT-TYPE is used.)

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`rows` *DATA* [Function]

Return the rows of DATA as a vector

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`set-properties` *DF PROPERTY PROP-VALUES* [Function]

Set the PROPERTY of each variable in DF to a value. The value is specified in the plist PROP-VALUES. Example:
To give the variables in the mtcars dataset a unit, use:
(set-properties mtcars :unit '(:mpg m/g
:cyl :NA
:disp in^3
:hp hp
:drat :NA
:wt lb
:qsec s
:vs :NA
:am :NA
:gear :NA
:carb :NA))

Package [`data-frame`], page 9,

Source [`properties.lisp`], page 8, (file)

`show-data-frames` **&key** *HEAD STREAM* [Function]

Print all data frames in the current environment in reverse order of creation, i.e. most recently created first. if HEAD is not NIL, print the first six rows, similar to the (head) function

Package [`data-frame`], page 9,

Source [`defdf.lisp`], page 7, (file)

`substitute-key!` *DF NEW OLD* [Function]

Substitute NEW key, a SYMBOL, for OLD in a data-frame.

Useful when reading data files that have an empty or generated column name.

Example: (substitute-key *cars* :name :||) to replace an empty symbol with :name

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`summarize-column` *COLUMN* [Function]

Return a summary struct for COLUMN

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`undef` *DF* [Function]

If DF is the symbol of a defined data-frame it is unbound and removed from the list of data-frames. If DF is a list of data-frame names each is unbound and removed. Returns DF.

Example: (undef 'mtcars)

Package [`data-frame`], page 9,

Source [`defdf.lisp`], page 7, (file)

4.1.4 Generic functions

`doc-string` *OBJECT* [Generic Function]
`(setf doc-string)` *NEW-VALUE OBJECT* [Generic Function]

Package [`data-frame`], page 9,

Methods

`doc-string` (*DATA* `data`) [Method]

automatically generated reader method

Source [`data-frame.lisp`], page 3, (file)

`(setf doc-string)` *NEW-VALUE* (*DATA* `data`) [Method]

automatically generated writer method

Source [`data-frame.lisp`], page 3, (file)

`drop-missing` *DF* **&optional** *PREDICATE* [Generic Function]

Package [`data-frame`], page 9,

Methods

`drop-missing` (*DF* `data-frame`) **&optional** *PREDICATE* [Method]

Remove all rows from DF that are missing values according to PREDICATE

Source [`missing.lisp`], page 8, (file)

`head` *DF* **&optional** *N* [Generic Function]

Package [`data-frame`], page 9,

Methods

`head` (*DF* `data-frame`) **&optional** *N* [Method]

Return the first N rows of DF; N defaults to 6

Source [`pprint.lisp`], page 5, (file)

`missingp` *DATA* [Generic Function]

Package [`data-frame`], page 9,

Source [`missing.lisp`], page 8, (file)

Methods

`missingp` *DATA* [Method]

`missingp` (*DATA* `(eql na)`) [Method]

`missingp` (*DATA* `(eql missing)`) [Method]

`missingp` (*DATA* `string`) [Method]

`missingp` (*DATA* `sequence`) [Method]

`missingp` (*DATA* `array`) [Method]

`missingp` (*DATA* `data-frame`) [Method]

`name` *OBJECT* [Generic Function]
`(setf name)` *NEW-VALUE OBJECT* [Generic Function]

Package [`data-frame`], page 9,

Methods

`name` (*DATA* `data`) [Method]
automatically generated reader method

Source [`data-frame.lisp`], page 3, (file)

`(setf name)` *NEW-VALUE* (*DATA* `data`) [Method]
automatically generated writer method

Source [`data-frame.lisp`], page 3, (file)

`replace-missing` *DF MAP-ALIST* [Generic Function]

Package [`data-frame`], page 9,

Methods

`replace-missing` (*DF* `data-frame`) *MAP-ALIST* [Method]
Replace missing values with the values specified
The alist consists of a column name in the CAR and the replacement value in the CDR Example: (replace-missing mtcarsm '((mpg . foo)))

Source [`missing.lisp`], page 8, (file)

`summary` *DF* **&optional** *STREAM* [Generic Function]

Package [`data-frame`], page 9,

Methods

`summary` (*DF* `data-frame`) **&optional** *STREAM* [Method]
Print a summary of DF to STREAM, using heuristics for better formatting

Source [`summary.lisp`], page 6, (file)

`tail` *DF* **&optional** *N* [Generic Function]

Package [`data-frame`], page 9,

Methods

`tail` (*DF* `data-frame`) **&optional** *N* [Method]
Return the last N rows of DF; N defaults to 6

Source [`pprint.lisp`], page 5, (file)

4.1.5 Conditions

`duplicate-key` () [Condition]
Duplicate key.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

Direct superclasses
`error` (condition)

Direct slots

`key` [Slot]

Initargs `:key`

`key-not-found` () [Condition]
Key not found.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

Direct superclasses
`error` (condition)

Direct slots

`key` [Slot]
Initargs `:key`

`keys` [Slot]
Initargs `:keys`

4.1.6 Structures

`bit-variable-summary` () [Structure]
Summary of a bit vector.

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

Direct superclasses
[`variable-summary%`], page 36, (structure)

Direct methods
`print-object` (method)

Direct slots

`count` [Slot]
Type `alexandria:array-index`
Initform `0`
Readers [`bit-variable-summary-count`], page 27, (function)
Writers `(setf bit-variable-summary-count)` (function)

`factor-variable-summary` () [Structure]
Summary for factor variables

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

Direct superclasses
[`variable-summary%`], page 36, (structure)

Direct methods
`print-object` (method)

Direct slots

`element-count-alist` [Slot]
Type `list`
Readers [`factor-variable-summary-element-count-alist`], page 29, (function)
Writers `(setf factor-variable-summary-element-count-alist)` (function)

`generic-variable-summary` () [Structure]
Summary for generic variables, i.e. those with mixed types.

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

Direct superclasses
[`variable-summary%`], page 36, (structure)

Direct methods
`print-object` (method)

Direct slots

`quantiles` [Slot]
Type `(or null data-frame:real-variable-summary)`
Readers [`generic-variable-summary-quantiles`], page 30, (function)
Writers `(setf generic-variable-summary-quantiles)` (function)

`element-count-alist` [Slot]
Type `list`
Readers [`generic-variable-summary-element-count-alist`], page 29, (function)
Writers `(setf generic-variable-summary-element-count-alist)` (function)

`real-variable-summary` () [Structure]
Summary of a real elements (using quantiles).

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

Direct superclasses
[`variable-summary%`], page 36, (structure)

Direct methods
`print-object` (method)

Direct slots

`min` [Slot]
Type `real`
Initform `0`
Readers [`real-variable-summary-min`], page 33, (function)
Writers `(setf real-variable-summary-min)` (function)

`q25` [Slot]
Type `real`
Initform `0`
Readers [`real-variable-summary-q25`], page 33, (function)
Writers `(setf real-variable-summary-q25)` (function)

`q50` [Slot]

Type `real`

Initform `0`

Readers [`real-variable-summary-q50`], page 33, (function)

Writers `(setf real-variable-summary-q50)` (function)

`mean` [Slot]

Type `real`

Initform `0`

Readers [`real-variable-summary-mean`], page 33, (function)

Writers `(setf real-variable-summary-mean)` (function)

`q75` [Slot]

Type `real`

Initform `0`

Readers [`real-variable-summary-q75`], page 33, (function)

Writers `(setf real-variable-summary-q75)` (function)

`max` [Slot]

Type `real`

Initform `0`

Readers [`real-variable-summary-max`], page 33, (function)

Writers `(setf real-variable-summary-max)` (function)

4.1.7 Classes

`data-frame` () [Class]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

Direct superclasses

[`data`], page 37, (class)

Direct methods

- [`replace-missing`], page 22, (method)
- [`drop-missing`], page 21, (method)
- [`missingp`], page 21, (method)
- [`summary`], page 22, (method)
- [`tail`], page 22, (method)
- [`head`], page 21, (method)
- `describe-object` (method)
- `print-object` (method)
- `select` (method)
- [`check-column-compatibility`], page 35, (method)
- `as-array` (method)

- `dims` (method)
- `ncol` (method)
- `nrow` (method)
- `initialize-instance` (method)

`data-vector` () [Class]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

Direct superclasses
[`data`], page 37, (class)

Direct methods
- `print-object` (method)
- `select` (method)
- `as-array` (method)
- `dims` (method)

4.2 Internal definitions

4.2.1 Special variables

`*ask-on-redefine*` [Special Variable]
If set, the system will ask the user for confirmation before redefining a data frame

Package [`data-frame`], page 9,

Source [`defdf.lisp`], page 7, (file)

`*data-frames*` [Special Variable]
Global list of all data frames

Package [`data-frame`], page 9,

Source [`defdf.lisp`], page 7, (file)

`*max-digits*` [Special Variable]

Package [`data-frame`], page 9,

Source [`pprint.lisp`], page 5, (file)

`*row-numbers-p*` [Special Variable]

Package [`data-frame`], page 9,

Source [`pprint.lisp`], page 5, (file)

4.2.2 Macros

`define-data-subclass` *CLASS ABBREVIATION* [Macro]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

4.2.3 Functions

`2d-array-to-list` *ARRAY* [Function]
Convert an array to a list of lists

Package [`data-frame`], page 9,

Source [`pprint.lisp`], page 5, (file)

`add-key!` *ORDERED-KEYS KEY* [Function]
Modify ORDERED-KEYS by adding KEY.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`add-keys` *ORDERED-KEYS* **&rest** *KEYS* [Function]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`aesthetic-string` *THING* [Function]
Return the string used to represent ‘thing‘ when printing aesthetically.

Package [`data-frame`], page 9,

Source [`formatted-output.lisp`], page 5, (file)

`alist-data` *CLASS ALIST* [Function]
Create an object of CLASS (subclass of DATA) from ALIST which contains key-column pairs.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`bit-variable-summary-count` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`bit-variable-summary-desc` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`bit-variable-summary-length` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`bit-variable-summary-missing` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`bit-variable-summary-name` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`bit-variable-summary-p` *OBJECT* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`column-type-format` *SEQUENCE* [Function]

Return a format string for the most specific type found in sequence Use this for sequences of type T to determine how to format the column.

Package [`data-frame`], page 9,

Source [`pprint.lisp`], page 5, (file)

`copy-bit-variable-summary` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`copy-factor-variable-summary` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`copy-generic-variable-summary` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`copy-ordered-keys` *ORDERED-KEYS* [Function]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`copy-real-variable-summary` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`copy-variable-summary%` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`distinct` *COLUMN* [Function]

Returns the number of distinct elements in COLUMN Useful for formatting columns for human output

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`drop-na` *DF* [Function]

Remove all rows from DF that are missing values. Convenience R-like function.

Package [`data-frame`], page 9,

Source [`missing.lisp`], page 8, (file)

`ensure-arguments-alist` *REST* [Function]

Recognizes the following and converts them to an alist:

plist
alist
(plist)
(alist)
(data-frame)

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`ensure-not-ratio` *REAL* [Function]

When REAL is a RATIO, convert it to a float, otherwise return as is. Used for printing.

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`factor-variable-summary-desc` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`factor-variable-summary-element-count-alist` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`factor-variable-summary-length` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`factor-variable-summary-missing` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`factor-variable-summary-name` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`factor-variable-summary-p` *OBJECT* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`generic-variable-summary-desc` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`generic-variable-summary-element-count-alist` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`generic-variable-summary-length` *INSTANCE* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`generic-variable-summary-missing` *INSTANCE* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`generic-variable-summary-name` *INSTANCE* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`generic-variable-summary-p` *OBJECT* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`generic-variable-summary-quantiles` *INSTANCE* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`get-type` *X* [Function]

Return the most specific type symbol for x

Package [data-frame], page 9,

Source [utils.lisp], page 3, (file)

`guess-alist?` *PLIST-OR-ALIST* [Function]

Test if the argument is an ALIST by checking its first element. Used for deciding which creation function to call.

Package [data-frame], page 9,

Source [data-frame.lisp], page 3, (file)

`key-index` *ORDERED-KEYS KEY* [Function]

Return the index for KEY.

Package [data-frame], page 9,

Source [data-frame.lisp], page 3, (file)

`keys-count` *ORDERED-KEYS* [Function]

Number of keys.

Package [data-frame], page 9,

Source [data-frame.lisp], page 3, (file)

`keys-vector` *ORDERED-KEYS* [Function]

Vector of all keys.

Package [data-frame], page 9,

Source [data-frame.lisp], page 3, (file)

`make-bit-variable-summary` **&key** (*LENGTH* **LENGTH**) (*MISSING* **MISSING**) (*NAME* **NAME**) (*DESC* **DESC**) (*COUNT* **COUNT**) [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`make-data` *CLASS KEYS COLUMNS* [Function]

Create a DATA object from KEYS and COLUMNS. FOR INTERNAL USE. Always creates a copy of COLUMNS in order to ensure that it is an adjustable array with a fill pointer. KEYS are converted to ORDERED-KEYS if necessary.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`make-factor-variable-summary` **&key** (*LENGTH* **LENGTH**) (*MISSING* **MISSING**) (*NAME* **NAME**) (*DESC* **DESC**) (*ELEMENT-COUNT-ALIST* **ELEMENT-COUNT-ALIST**) [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`make-generic-variable-summary` **&key** (*LENGTH* **LENGTH**) (*MISSING* **MISSING**) (*NAME* **NAME**) (*DESC* **DESC**) (*QUANTILES* **QUANTILES**) (*ELEMENT-COUNT-ALIST* **ELEMENT-COUNT-ALIST**) [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`make-ordered-keys` **&key** (*TABLE* **TABLE**) [Function]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`make-real-variable-summary` **&key** (*LENGTH* **LENGTH**) (*MISSING* **MISSING**) (*NAME* **NAME**) (*DESC* **DESC**) (*MIN* **MIN**) (*Q25* **Q25**) (*Q50* **Q50**) (*MEAN* **MEAN**) (*Q75* **Q75**) (*MAX* **MAX**) [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`make-variable-summary%` **&key** (*LENGTH* **LENGTH**) (*MISSING* **MISSING**) (*NAME* **NAME**) (*DESC* **DESC**) [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`max-decimal` *SEQUENCE* **&optional** *MAX-DIGITS* [Function]

Return the maximum number of digits to the right of the decimal point in the numbers of SEQUENCE, equal to or less than MAX-DIGITS

Package [`data-frame`], page 9,

Source [`pprint.lisp`], page 5, (file)

`max-width` *SEQUENCE* **&optional** *MAX-WIDTH* [Function]

Return the largest printed string size of the elements of SEQUENCE, equal to or less than MAX-WIDTH

Package [`data-frame`], page 9,

Source [`pprint.lisp`], page 5, (file)

`monotonicp` *COLUMN* [Function]
Returns t if all elements of COLUMN are increasing monotonically Useful for detecting row numbers in imported data

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`ordered-keys` *KEYS* [Function]
Create an ORDERED-KEYS object from KEYS (a sequence).

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`ordered-keys-p` *OBJECT* [Function]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`ordered-keys-table` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`plist-data` *CLASS PLIST* [Function]
Create an object of CLASS (subclass of DATA) from PLIST which contains keys and columns, interleaved.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

`print-count-and-percentage` *STREAM COUNT LENGTH* [Function]
Print COUNT as is and also as a rounded percentage of

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`print-table` *ROWS* **&optional** *STREAM* [Function]
Print ‘rows‘ as a nicely-formatted table.
Each row should have the same number of colums.
Columns will be justified properly to fit the longest item in each one. Example:
(print-table ’((1 :red something)
(2 :green more)))
=>
1 | RED | SOMETHING
2 | GREEN | MORE

Package [`data-frame`], page 9,

Source [`formatted-output.lisp`], page 5, (file)

`printer-status` () [Function]
Print values of all the printer variables

Package [`data-frame`], page 9,

Source [`pprint.lisp`], page 5, (file)

`real-variable-summary-desc` *INSTANCE* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`real-variable-summary-length` *INSTANCE* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`real-variable-summary-max` *INSTANCE* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`real-variable-summary-mean` *INSTANCE* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`real-variable-summary-min` *INSTANCE* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`real-variable-summary-missing` *INSTANCE* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`real-variable-summary-name` *INSTANCE* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`real-variable-summary-p` *OBJECT* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`real-variable-summary-q25` *INSTANCE* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`real-variable-summary-q50` *INSTANCE* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`real-variable-summary-q75` *INSTANCE* [Function]

Package [data-frame], page 9,

Source [summary.lisp], page 6, (file)

`reverse-df` *DF* [Function]

Return DF with columns in reverse order

Package [data-frame], page 9,

Source [pprint.lisp], page 5, (file)

`show-properties` *DF* [Function]
Show the standard properties of the variables of the data frame DF Standard properties are 'label', 'type' and 'unit'

Package [`data-frame`], page 9,

Source [`properties.lisp`], page 8, (file)

`show-symbols` *PKG* [Function]
Print all symbols in PKG Example: (show-symbols 'mtcars)

Package [`data-frame`], page 9,

Source [`defdf.lisp`], page 7, (file)

`summarize-factor-variable` *COLUMN* [Function]
Return an alist of factor/count pairs

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`summarize-generic-variable` *COLUMN* [Function]
Return an object that summarizes COLUMN of a DATA-FRAME. Primarily intended for printing, not analysis, returned values should print nicely. This function can be used on any type of column, even one with mixed types

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`summarize-real-variable` *COLUMN* [Function]
Return a summary for a float variable

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`sym-mac` *DF VAR* [Function]
Return the symbol macro for VAR in the DATA-FRAME DF

Package [`data-frame`], page 9,

Source [`properties.lisp`], page 8, (file)

`types-in-column` *SEQ* [Function]
Return a list of the types found in SEQ

Package [`data-frame`], page 9,

Source [`utils.lisp`], page 3, (file)

`undef-column-names` *DF* [Function]
Remove symbol macro for each column name in DF

Package [`data-frame`], page 9,

Source [`defdf.lisp`], page 7, (file)

`var-name` *VAR* [Function]
Return the name of the variable without the symbol-macro prefix Example: (var-name mtcars$mpg) returns 'mpg'

Package [`data-frame`], page 9,

Source [`properties.lisp`], page 8, (file)

`variable-summary%-desc` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`variable-summary%-length` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`variable-summary%-missing` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`variable-summary%-name` *INSTANCE* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`variable-summary%-p` *OBJECT* [Function]

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

`weave` **&rest** *LISTS* [Function]

Return a list whose elements alternate between each of the lists 'lists'. Weaving stops when any of the lists has been exhausted.

Package [`data-frame`], page 9,

Source [`formatted-output.lisp`], page 5, (file)

4.2.4 Generic functions

`check-column-compatibility` *DATA COLUMN* [Generic Function]

Check if COLUMN is compatible with DATA.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

Methods

`check-column-compatibility` (*DATA* `data-frame`) *COLUMN* [Method]

`check-column-compatibility` (*DATA* `data`) *COLUMN* [Method]

`column-length` *COLUMN* [Generic Function]

Return the length of column.

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

Methods

`column-length` (*COLUMN* `vector`) [Method]

`default-column-formats` *ARRAY* [Generic Function]

Package [`data-frame`], page 9,

Methods

`default-column-formats` (*ARRAY* `simple-array`) [Method]
Return a list of formatting strings for ARRAY
The method returns a set of default formatting strings using heuristics.

Source [`pprint.lisp`], page 5, (file)

4.2.5 Structures

`ordered-keys` () [Structure]
Representation of ordered keys.

TABLE maps keys to indexes, starting from zero.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

Direct superclasses
`structure-object` (structure)

Direct methods
- `print-object` (method)
- `select` (method)
- `canonical-representation` (method)
- `axis-dimension` (method)

Direct slots

`table` [Slot]
Type `hash-table`
Initform `(make-hash-table :test (function eq))`
Readers [`ordered-keys-table`], page 32, (function)
Writers `(setf ordered-keys-table)` (function)

`variable-summary%` () [Structure]
Base class for summarizing variables. Not exported.

Package [`data-frame`], page 9,

Source [`summary.lisp`], page 6, (file)

Direct superclasses
`structure-object` (structure)

Direct subclasses
- [`bit-variable-summary`], page 23, (structure)
- [`real-variable-summary`], page 24, (structure)
- [`factor-variable-summary`], page 23, (structure)
- [`generic-variable-summary`], page 24, (structure)

Direct slots

`length` [Slot]
Type `alexandria:array-index`
Initform `0`
Readers [`variable-summary%-length`], page 35, (function)
Writers `(setf variable-summary%-length)` (function)

`missing` [Slot]

Type `fixnum`

Initform `0`

Readers [`variable-summary%-missing`], page 35, (function)

Writers `(setf variable-summary%-missing)` (function)

`name` [Slot]

Type `string`

Initform `""`

Readers [`variable-summary%-name`], page 35, (function)

Writers `(setf variable-summary%-name)` (function)

`desc` [Slot]

Type `string`

Initform `""`

Readers [`variable-summary%-desc`], page 35, (function)

Writers `(setf variable-summary%-desc)` (function)

4.2.6 Classes

`data` () [Class]

This class is used for implementing both data-vector and data-matrix, and represents an ordered collection of key-column pairs. Columns are not assumed to have any specific attributes. This class is not exported.

Package [`data-frame`], page 9,

Source [`data-frame.lisp`], page 3, (file)

Direct superclasses

`standard-object` (class)

Direct subclasses

- [`data-vector`], page 26, (class)
- [`data-frame`], page 25, (class)

Direct methods

- `as-alist` (method)
- [`check-column-compatibility`], page 35, (method)
- `element-type` (method)
- `doc-string` (method)
- [`doc-string`], page 21, (method)
- `name` (method)
- [`name`], page 22, (method)

Direct slots

`name` [Slot]

Type `string`

Initargs `nil`

Readers [`name`], page 22, (generic function)

Writers [`(setf name)`], page 22, (generic function)

`ordered-keys` [Slot]

Type `data-frame::ordered-keys`

Initargs `:ordered-keys`

`columns` [Slot]

Type `vector`

Initargs `:columns`

`doc-string` [Slot]

Type `string`

Initargs `:nil`

Readers [`doc-string`], page 21, (generic function)

Writers [`(setf doc-string)`], page 21, (generic function)

Appendix A Indexes

A.1 Concepts

D

F

L

A.2 Functions

G

H

K

M

N

O

P

R

S

T

U

V

W

A.3 Variables

*

C

D

E

K

L

M

N

O

Q

S

T

A.4 Data types