

Package ‘dataset’

December 8, 2023

Title Create Data Frames that are Easier to Exchange and Reuse

Date 2023-12-08

Version 0.2.7

Description The aim of the 'dataset' package is to make tidy datasets easier to release, exchange and reuse. It organizes and formats data frame 'R' objects into well-referenced, well-described, interoperable datasets into release and reuse ready form. A subjective interpretation of the W3C DataSet recommendation and the datacube model <<https://www.w3.org/TR/vocab-data-cube/>>, which is also used in the global Statistical Data and Metadata eXchange standards, the application of the connected Dublin Core <<https://www.dublincore.org/specifications/dublin-core/dcmi-terms/>> and DataCite <<https://support.datacite.org/docs/datacite-metadata-schema-44/>> standards preferred by European open science repositories to improve the findability, accessibility, interoperability and reusability of the datasets.

License GPL (>= 3)

URL <https://github.com/dataobservatory-eu/dataset>

BugReports <https://github.com/dataobservatory-eu/dataset/issues>

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 2.10)

LazyData true

Imports assertthat, ISOcodes, stats, utils

Suggests dataspice, covr, declared, dplyr, eurostat, here, kableExtra, knitr, rdfliib, readxl, rmarkdown, spelling, statcodelists, testthat (>= 3.0.0), tidyr

VignetteBuilder knitr

Config/testthat/edition 3

Language en-US

NeedsCompilation no

Author Daniel Antal [aut, cre] (<<https://orcid.org/0000-0001-7513-6760>>),
 Marcelo Perlin [rev] (<<https://orcid.org/0000-0002-9839-4268>>)

Maintainer Daniel Antal <daniel.antal@dataobservatory.eu>

Repository CRAN

Date/Publication 2023-12-08 15:30:02 UTC

R topics documented:

as_dataset	2
creator	5
datacite	6
dataset_bibentry	8
dataset_title	9
datasource	10
DataSet	11
describe	12
description	12
dublincore	13
geolocation	16
identifier	17
iris_dataset	18
language	19
publication_year	20
publisher	21
related_item	22
rights	24
size	25
subject	26
var_labels	27
version	28
Index	29

as_dataset

Create a dataset

Description

Create a dataset

Usage

```
as_dataset(  
  x,  
  author,  
  title,  
  publisher = NULL,  
  year = NULL,  
  identifier = NULL,  
  description = NULL,  
  version = NULL,  
  subject = NULL,  
  language = NULL,  
  datasource = NULL,  
  rights = NULL  
)  
  
## S3 method for class 'data.frame'  
as_dataset(  
  x,  
  author,  
  title,  
  publisher = NULL,  
  year = NULL,  
  identifier = NULL,  
  description = NULL,  
  version = NULL,  
  subject = NULL,  
  language = NULL,  
  datasource = NULL,  
  rights = NULL  
)  
  
dataset(  
  x,  
  author,  
  title,  
  identifier = NULL,  
  publisher = NULL,  
  year = NULL,  
  version = NULL,  
  subject = NULL,  
  description = NULL,  
  language = NULL,  
  datasource = NULL,  
  rights = NULL,  
  ...  
)
```

```
is.dataset(x)

## S3 method for class 'dataset'
print(x, ...)

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

Arguments

x	An R object that contains the data of the dataset (a data.frame or inherited from data.frame, for example, tibble, tsibble, data.table).
author	A single person or a vector of persons as authors, declared with person .
title	The title of the dataset.
publisher	The organisation or person that publishes the dataset. If left empty (NULL), receives the DataCite standard value ' : tba ', or to be announced later.
year	The year of the creation of the dataset. If left empty (NULL), the current year.
identifier	The permanent identifier, for example, the DOI of the dataset. If left empty (NULL), receives the DataCite standard value ' : tba ', or to be announced later.
description	The optional Description property as an attribute to an R object.
version	The version of the dataset. If left empty (NULL), defaults to '0.1.0'
subject	The subject of the data frame, as a subject type.
language	The primary language of the dataset, for example 'eng'. Defaults to NULL that sets it to the unassigned value ' : unas '.
datasource	The source of the dataset, DCMI: Source , which corresponds to a relatedItem in the DataCite vocabulary. We use datasource instead of source to avoid naming conflicts with the base R source() function.
rights	Any rights information for this resource. The property may be repeated to record complex rights characteristics. Free text, defaults to " : unas " for unassigned values. See rights .
...	Further arguments passed on to generic methods like summary(x, ...).
object	an object for which a summary is desired.

Value

A dataset object, which is a data.frame or inherited object with rich metadata.

Examples

```
ds <- dataset(iris,
  title = "The iris Dataset",
  author = c(
    person(family = "Anderson",
      given = "Edgar",
      role = "aut")
  ),
```

```

identifier = "https://doi.org/10.1111/j.1469-1809.1936.tb02137.x",
year = "1935",
version = "1.0",
description = "The famous dataset that is distributed with R.",
url = "https://en.wikipedia.org/wiki/Iris_flower_data_set",
resourceType = "Dataset"
)

```

creator

Get/set the Creator of the object.

Description

Add the optional Creator property as an attribute to a dataset object.

Usage

```
creator(x)
```

```
creator(x, overwrite = TRUE) <- value
```

Arguments

x	A dataset object created by dataset .
overwrite	If the attributes should be overwritten. In case it is set to FALSE, it gives a message with the current Creator property instead of overwriting it. Defaults to TRUE when the attribute is set to value regardless of previous setting.
value	The Creator as a utils::person object.

Details

The Creator corresponds to **dct:creator** in Dublin Core and Creator in DataCite. The name of the entity that holds, archives, publishes prints, distributes, releases, issues, or produces the dataset. This property will be used to formulate the citation, so consider the prominence of the role.

Value

The Creator attribute as a character of length one is added to x.

See Also

Other Reference metadata functions: [datasource\(\)](#), [description\(\)](#), [geolocation\(\)](#), [identifier\(\)](#), [language](#), [publication_year\(\)](#), [publisher\(\)](#), [rights\(\)](#), [size\(\)](#), [version\(\)](#)

Examples

```
creator(iris_dataset)
```

 datacite

Add or get DataCite metadata

Description

Add or retrieve metadata conforming the [DataCite Metadata Schema](#).

Usage

```

datacite(
  Title,
  Creator,
  Identifier = NULL,
  Publisher = NULL,
  PublicationYear = NULL,
  Subject = NULL,
  Type = "Dataset",
  Contributor = NULL,
  Datelist = NULL,
  Language = NULL,
  AlternateIdentifier = NULL,
  RelatedIdentifier = NULL,
  Format = NULL,
  Version = NULL,
  Rights = NULL,
  Description = NULL,
  Geolocation = NULL,
  FundingReference = NULL
)

as_datacite(x, type = "bibentry", ...)

is.datacite(x)

## S3 method for class 'datacite'
is.datacite(x)

```

Arguments

Title	The name(s) or title(s) by which a resource is known. May be the title of a dataset or the name of a piece of software. Similar to dct:title .
Creator	The main researchers involved in producing the data, or the authors of the publication, in priority order. To supply multiple creators, repeat this property.
Identifier	The Identifier is a unique string that identifies a resource. For software, determine whether the identifier is for a specific version of a piece of software,

(per the [Force11 Software Citation Principles](#), or for all versions. Similar to `dct:title` in [dublincore](#).

Publisher	The name of the entity that holds, archives, publishes prints, distributes, releases, issues, or produces the resource. This property will be used to formulate the citation, so consider the prominence of the role. For software, use Publisher for the code repository. Mandatory in DataCite, and similar to <code>dct:publisher</code> . See publisher .
PublicationYear	The year when the data was or will be made publicly available in YYYY format. See publication_year .
Subject	Recommended for discovery. Subject, keyword, classification code, or key phrase describing the resource. Similar to <code>dct:subject</code> . Use subject to properly add a key phrase from a controlled vocabulary and create structured Subject objects with subject_create .
Type	Defaults to Dataset. The DataCite resourceType definition refers back to <code>dcm:type</code> . The <code>Type\$resourceTypeGeneral</code> is set to Dataset, while the user can set a more specific <code>Type\$resourceType</code> value.
Contributor	Recommended for discovery. The institution or person responsible for collecting, managing, distributing, or otherwise contributing to the development of the resource.
DateList	DataCite 4.4 allows to set multiple dates to a resource, they should be added as a list. See: datacite:Date .
Language	The primary language of the resource. Allowed values are taken from IETF BCP 47, ISO 639-1 language code. See language .
AlternateIdentifier	An identifier or identifiers other than the primary Identifier applied to the resource being registered. This may be any alphanumeric string unique within its domain of issue. It may be used for local identifiers. <code>AlternateIdentifier</code> should be used for another identifier of the same instance (same location, same file).
RelatedIdentifier	Recommended for discovery. Similar to <code>dct:relation</code> .
Format	Technical format of the resource. Similar to <code>dct:format</code> .
Version	Free text. Suggested practice: track <code>major_version.minor_version</code> . See version .
Rights	Any rights information for this resource. The property may be repeated to record complex rights characteristics. Free text. See rights .
Description	Recommended for discovery. All additional information that does not fit in any of the other categories. It may be used for technical information—a free text. Similar to <code>dct:description</code> .
Geolocation	Recommended for discovery. Spatial region or named place where the data was gathered or about which the data is focused. See geolocation .
FundingReference	Information about financial support (funding) for the resource being registered.
x	An object that is tested if it has a class "datacite".

type A DataCite 4.4 metadata can be returned as a type="list", a type="dataset", or a type="bibentry" (default).

... Optional parameters to add to a datacite object. author=person("Jane", "Doe") adds an author to the citation object if type="dataset".

Details

DataCite is a leading global non-profit organisation that provides persistent identifiers (DOIs) for research data and other research outputs. Organisations within the research community join DataCite as members to be able to assign DOIs to all their research outputs. This way, their outputs become discoverable, and associated metadata is made available to the community.

The ResourceType property will be by definition "Dataset". The Size attribute (e.g. bytes, pages, inches, etc.) will automatically added to the dataset.

Value

A `utils::bibentry` object DataCite attributes. `as_datacite` returns the existing metadata of a dataset object.

Source

[DataCite 4.3 Mandatory Properties](#) and [DataCite 4.3 Optional Properties](#)

See Also

Other metadata functions: `dublincore()`, `related_item()`

Examples

```
datacite(
  Title = "Iris Dataset",
  Creator = person(family = "Anderson", given = "Edgar", role = "aut"),
  Publisher = "American Iris Society",
  PublicationYear = 1935,
  Geolocation = "US",
  Language = "en")

as_datacite(iris_dataset, type="dataset")
```

dataset_bibentry

Get the bibliographic entries of a dataset

Description

A dataset constructed with `dataset` records most of the dataset-level metadata in a `bibentry` object. A dataset constructed with `dataset` records most of the dataset-level metadata in a `bibentry` object. This class is defined in the `utils` package, and it is installed by default with R; using the gives wide interoperability with other packages and allows printing or saving the bibliographic record in HTML, BibLatex and other formats. For further information see `utils::bibentry`.

Usage

```
dataset_bibentry(x)
```

Arguments

x A dataset object created with `dataset::dataset`.

Value

The `bibentry` object of the dataset.

Examples

```
ds <- dataset(iris,
  title = "The iris Dataset",
  author = c(
    person(family = "Anderson",
      given = "Edgar",
      role = "aut")
  ),
  identifier = "https://doi.org/10.1111/j.1469-1809.1936.tb02137.x",
  year = "1935",
  version = "1.0",
  description = "The famous dataset that is distributed with R.",
  url = "https://en.wikipedia.org/wiki/Iris_flower_data_set",
  resourceType = "Dataset"
)

dataset_bibentry(ds)
```

dataset_title	<i>Get/set title of a dataset</i>
---------------	-----------------------------------

Description

Get or reset the dataset's main title.

Usage

```
dataset_title(x)

dataset_title(x, overwrite = FALSE) <- value
```

Arguments

x A dataset object created with `dataset::dataset`.

overwrite If the attributes should be overwritten. In case it is set to FALSE, it gives a warning with the current title property instead of overwriting it. Defaults to FALSE.

value The name(s) or title(s) by which a resource is known. See: `dct:title`.

Details

In the DataCite definition, several titles can be used; it is not yet implemented.

Value

A string with the dataset's title; `set_dataset_title` returns a dataset object with the changed (main) title.

Examples

```
dataset_title(iris_dataset)
dataset_title(iris_dataset, overwrite = TRUE) <-"The Famous Iris Dataset"
dataset_title(iris_dataset)
```

datasource

Get/set the Source of the object.

Description

Get/set the optional Source property as an attribute to an R object. Do not confuse with the base R `source()` function.

Usage

```
datasource(x)

datasource(x, overwrite = TRUE) <- value
```

Arguments

<code>x</code>	A dataset object created with <code>dataset::dataset</code> .
<code>overwrite</code>	If the attributes should be overwritten. In case it is set to <code>FALSE</code> , it gives a warning with the current <code>datasource</code> property instead of overwriting it. Defaults to <code>FALSE</code> .
<code>value</code>	The Source as a character string of length one.

Details

The Source is a related resource from which the described resource is derived. See `dct:source`. In DataCite, the source is described by a `relatedIdentifierType` with the property `relationType="isDerivedFrom"`.

Value

The Source attribute as a character of length 1 is added to `x`.

See Also

Other Reference metadata functions: [creator\(\)](#), [description\(\)](#), [geolocation\(\)](#), [identifier\(\)](#), [language](#), [publication_year\(\)](#), [publisher\(\)](#), [rights\(\)](#), [size\(\)](#), [version\(\)](#)

Examples

```
datasource(iris_dataset) <- "https://doi.org/10.1111/j.1469-1809.1936.tb02137.x"
datasource(iris_dataset)
```

DataStructure	<i>Data structure</i>
---------------	-----------------------

Description

Read or update the Data Structure Definition of a dataset.

Usage

```
DataStructure(x)

DataStructure_update(x, value)
```

Arguments

x	A dataset.
value	A DataStructure definition.

Value

A dataset object, which is a data.frame or inherited object with rich metadata.

Examples

```
DataStructure(iris_dataset)
dsd_iris <- DataStructure(iris_dataset)
dsd_iris$Sepal.Length$label <- "The sepal length measured in centimeters."
dsd_iris$Sepal.Width$label <- "The sepal width measured in centimeters."
dsd_iris$Petal.Length$label <- "The petal length measured in centimeters."
dsd_iris$Petal.Width$label <- "The petal width measured in centimeters."
dsd_iris$Species$label <- "The name of the Iris species in the observation."

iris_dataset_labelled <- DataStructure_update(iris_dataset, dsd_iris)

vapply(DataStructure(iris_dataset_labelled), function(x) x$label, character(1))
```

describe

Describe a dataset object

Description

Print to the screen an easy-to-read summary of a the attributes of a dataset.

Usage

```
describe(x)
```

```
## S3 method for class 'dataset'  
describe(x)
```

Arguments

x A dataset object created with `dataset::dataset`.

Details

A convenience function to review the most important attributes of a dataset object. The dataset class adds a wide range of metadata as attributes to `data.frame`, `tibble`, `data.table` or similar R object that contains tabular data. Overviewing these attributes becomes cumbersome with base R `attributes()`.

Value

No object is returned, but they key attributes are printed on the screen.

Examples

```
describe(iris_dataset)
```

description

Get/set the Description of the object.

Description

Get/set the optional Description property as an attribute to an R object.

Usage

```
description(x)
```

```
description(x, overwrite = FALSE) <- value
```

Arguments

x	A dataset object created with <code>dataset::dataset</code> .
overwrite	If the Description attribute should be overwritten. In case it is set to FALSE, it gives a message with the current Description property instead of overwriting it. Defaults to FALSE, when it gives a warning at an accidental overwrite attempt.
value	The Description as a character set.

Details

The Description is recommended for discovery in DataCite. All additional information that does not fit in any of the other categories. May be used for technical information. A free text. Similar to `dct:description`.

Value

The Description attribute as a character of length 1 is added to x.

See Also

Other Reference metadata functions: `creator()`, `datasource()`, `geolocation()`, `identifier()`, `language`, `publication_year()`, `publisher()`, `rights()`, `size()`, `version()`

Examples

```
description(iris_dataset, overwrite = TRUE) <- "The famous iris dataset used
in R language examples."
description(iris_dataset)
```

dublincore

Add or get DublinCore metadata

Description

Add metadata conforming the [DCMI Metadata Terms](#). to datasets, i.e. structured R data.frame or list objects, for an accurate and consistent identification of a resource for citation and retrieval purposes.

Usage

```
dublincore(
  title,
  creator,
  identifier = NULL,
  publisher = NULL,
  subject = NULL,
  type = "DCMITYPE:Dataset",
  contributor = NULL,
```

```

    date = NULL,
    language = NULL,
    relation = NULL,
    format = "application/r-rds",
    rights = NULL,
    datasource = NULL,
    description = NULL,
    coverage = NULL
)

as_dublincore(x, type = "bibentry", ...)

is.dublincore(x)

## S3 method for class 'dublincore'
is.dublincore(x)

```

Arguments

title	dct:title , a name given to the resource. datacite allows the use of alternate titles, too. See dataset_title .
creator	An entity primarily responsible for making the resource. dct:creator Corresponds to Creator in datacite . See creator .
identifier	An unambiguous reference to the resource within a given context. Recommended practice is to identify the resource by means of a string conforming to an identification system. Examples include International Standard Book Number (ISBN), Digital Object Identifier (DOI), and Uniform Resource Name (URN). Select an identifier scheme from registered URI schemes maintained by IANA . More details: Guidelines for using resource identifiers in Dublin Core metadata and IEEE LOM . Similar to Identifier in datacite . See identifier .
publisher	Corresponds to dct:publisher and Publisher in DataCite. The name of the entity that holds, archives, publishes prints, distributes, releases, issues, or produces the resource. This property will be used to formulate the citation, so consider the prominence of the role. For software, use Publisher for the code repository. If there is an entity other than a code repository, that "holds, archives, publishes, prints, distributes, releases, issues, or produces" the code, use the property Contributor/contributorType/hostingInstitution for the code repository. See publisher .
subject	Defaults to NULL. See subject to add subject descriptions to your dataset.
type	The nature or genre of the resource. Recommended best practice is to use a controlled vocabulary such as the DCMI Type Vocabulary DCMITYPE . For a dataset, the correct term is Dataset. To describe the file format, physical medium, or dimensions of the resource, use the Format element.
contributor	An entity responsible for making contributions to the dataset. See DCMI: Contributor .
date	Corresponds to a point or period of time associated with an event in the lifecycle of the resource. dct:date . Date is also recommended for discovery in datacite ,

	but it requires a different formatting.
language	A language of the dataset. See DCMI: Language .
relation	A related resource. Recommended best practice is to identify the related resource by means of a string conforming to a formal identification system. See: dct:relation . Similar to <code>RelatedItem</code> in datacite , which is recommended for discovery.
format	The file format, physical medium, or dimensions of the dataset. See DCMI: Format .
rights	Corresponds to dct:rights and datacite Rights . Information about rights held in and over the resource. Typically, rights information includes a statement about various property rights associated with the resource, including intellectual property rights. See rights .
datasource	The source of the dataset, DCMI: Source , which corresponds to a <code>relatedItem</code> in the DataCite vocabulary. We use <code>datasource</code> instead of <code>source</code> to avoid naming conflicts with the
description	An account of the resource. It may include but is not limited to: an abstract, a table of contents, a graphical representation, or a free-text account of the resource. dct:description . In datacite it is recommended for discovery. See description .
coverage	The spatial or temporal topic of the resource, spatial applicability of the dataset, or jurisdiction under which the dataset is relevant. See DCMI: Coverage .
x	An object that is tested if it has a class "dublincore".
...	Optional parameters to add to a <code>dublincore</code> object. <code>author=person("Jane", "Doe")</code> adds an author to the citation object if <code>type="dataset"</code> .

Details

The Dublin Core, also known as the Dublin Core Metadata Element Set (DCMES), is a set of fifteen main metadata items for describing digital or physical resources, such as datasets or their printed versions. Dublin Core has been formally standardized internationally as ISO 15836, as IETF RFC 5013 by the Internet Engineering Task Force (IETF), as well as in the U.S. as ANSI/NISO Z39.85.

The `ResourceType` property will be by definition "Dataset". The `Size` attribute (e.g. bytes, pages, inches, etc.) will automatically added to the dataset.

Value

The Dublin Core Metadata elements of the dataset in a `utils::bibentry` object.

Source

[DCMI Metadata Terms](#).

See Also

Other metadata functions: [datacite\(\)](#), [related_item\(\)](#)

Examples

```
dublincore(
  title = "Iris Dataset",
  creator = person("Edgar", "Anderson", role = "aut"),
  publisher = "American Iris Society",
  datasource = "https://doi.org/10.1111/j.1469-1809.1936.tb02137.x",
  date = 1935,
  language = "en",
  description = "This famous (Fisher's or Anderson's) iris data set gives the
  measurements in centimeters of the variables sepal length and width and petal length
  and width, respectively, for 50 flowers from each of 3 species of iris.
  The species are Iris setosa, versicolor, and virginica."
)
```

 geolocation

Get/set the Geolocation of the object.

Description

Get/set the optional Geolocation property as an attribute to an R object.

Usage

```
geolocation(x)

geolocation(x, overwrite = TRUE) <- value
```

Arguments

x	A dataset object created with <code>dataset::dataset</code> .
overwrite	If the attributes should be overwritten. In case it is set to FALSE, it gives a message with the current Geolocation property instead of overwriting it. Defaults to TRUE when the attribute is set to value regardless of previous setting.
value	The Geolocation as a character string.

Details

The Geolocation is recommended for discovery in DataCite 4.4. Spatial region or named place where the data was gathered or about which the data is focused. See: [datacite:Geolocation](#).

Value

The Geolocation attribute as a character of length 1 is added to x.

See Also

Other Reference metadata functions: [creator\(\)](#), [datasource\(\)](#), [description\(\)](#), [identifier\(\)](#), [language](#), [publication_year\(\)](#), [publisher\(\)](#), [rights\(\)](#), [size\(\)](#), [version\(\)](#)

Examples

```
iris_dataset <- iris
geolocation(iris_dataset) <- "US"
geolocation(iris_dataset)

geolocation(iris_dataset, overwrite = FALSE) <- "GB"
```

identifier	<i>Get/set the Identifier of the object.</i>
------------	----------------------------------------------

Description

Add the optional Identifier property as an attribute to an R object.

Usage

```
identifier(x)

identifier(x, overwrite = TRUE) <- value
```

Arguments

x	An R object, such as a data.frame, a tibble, or a data.table.
overwrite	If the attributes should be overwritten. In case it is set to FALSE, it gives a message with the current Identifier property instead of overwriting it. Defaults to TRUE when the attribute is set to value regardless of previous setting.
value	The Identifier as a character string.

Details

The Identifier is an unambiguous reference to the resource within a given context. Recommended practice is to identify the resource by means of a string conforming to an identification system. Examples include International Standard Book Number (ISBN), Digital Object Identifier (DOI), and Uniform Resource Name (URN). Select an identifier scheme from [registered URI schemes maintained by IANA](#). More details: [Guidelines for using resource identifiers in Dublin Core metadata and IEEE LOM](#). Similar to Identifier in [datacite. DataCite 4.4](#).

It is not part of the "core" Dublin Core terms, but we always add it to the metadata attributes of a dataset (in case you use a strict Dublin Core property sheet you can omit it.) [Dublin Core metadata terms](#).

Value

The Identifier attribute as a character of length 1 is added to x.

See Also

Other Reference metadata functions: [creator\(\)](#), [datasource\(\)](#), [description\(\)](#), [geolocation\(\)](#), [language](#), [publication_year\(\)](#), [publisher\(\)](#), [rights\(\)](#), [size\(\)](#), [version\(\)](#)

Examples

```
identifier(iris_dataset) <- "https://doi.org/10.1111/j.1469-1809.1936.tb02137.x"
identifier(iris_dataset)
```

iris_dataset	<i>Edgar Anderson's Iris Data</i>
--------------	-----------------------------------

Description

This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are *Iris setosa*, *versicolor*, and *virginica*. This is a replication of `datasets::iris` as `dataset` s3 class.

Usage

```
iris_dataset
```

Format

iris is a data frame with 150 cases (rows) and 5 variables (columns) named Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, and Species.

Details

See `datasets::iris` for details.

Source

Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, Part II, p179–188.

The data were collected by Anderson, Edgar (1935). The irises of the Gaspé Peninsula, *Bulletin of the American Iris Society*, **59**, 2–5.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

language	<i>Get/Set the primary language of the dataset</i>
----------	----------------------------------------------------

Description

Add the optional Language property as an attribute to an R object.

Usage

```
language(x)

language(x, iso_639_code = "639-3") <- value
```

Arguments

x	An R object, such as a data.frame, a tibble, or a character vector.
iso_639_code	Defaults to ISO 639-3, alternative is ISO 639-1.
value	The language to be added to the object attributes, added by name, or as a 2- or 3-character code for the language. You can add a language code or language name, and the parameter is normalized to <code>tolower(language)</code> . (The ISO 639 standard capitalizes language names and uses lower case for the codes.)

Details

Language is an optional property in DataCite 4.4; see: [datacite:Language](#)
It is a part of the "core" of the [Dublin Core metadata terms](#). The language parameter is validated against the `[ISOCodes]{ISO_639_2}` table.
The attribute language is added to the object. It will be exported into DataCite applications in a capitalized Lanugage format.

Value

The Language is added to the x as ISO 639-1, the Datacite recommendation, or ISO 639-3 used by the Zenodo data repository.

See Also

Other Reference metadata functions: [creator\(\)](#), [datasource\(\)](#), [description\(\)](#), [geolocation\(\)](#), [identifier\(\)](#), [publication_year\(\)](#), [publisher\(\)](#), [rights\(\)](#), [size\(\)](#), [version\(\)](#)

Examples

```
myiris <- iris_dataset
language(myiris) <- "English"
language(myiris)
language(myiris) <- "fr"
language(myiris)
```

publication_year	<i>Get/set the publication_year of the object.</i>
------------------	----------------------------------------------------

Description

Get/set the optional publication_year property as an attribute to an R object.

Usage

```
publication_year(x)

publication_year(x, overwrite = TRUE) <- value
```

Arguments

x	A dataset object created with <code>dataset::dataset</code> .
overwrite	If the attributes should be overwritten. In case it is set to FALSE, it gives a message with the current <code>PublicationYear</code> property instead of overwriting it. Defaults to TRUE when the attribute is set to value regardless of previous setting.
value	The publication_year as a character set.

Details

The `PublicationYear` is the year when the data was or will be made publicly available in YYYY format. See [Publication Year: DataCite Additional Guidance](#).

Value

Returns the year metadata field of the `DataBibentry` of the dataset

See Also

Other Reference metadata functions: [creator\(\)](#), [datasource\(\)](#), [description\(\)](#), [geolocation\(\)](#), [identifier\(\)](#), [language](#), [publisher\(\)](#), [rights\(\)](#), [size\(\)](#), [version\(\)](#)

Examples

```
publication_year(iris_dataset)
publication_year(iris_dataset) <- 1936
```

publisher	<i>Get/set the Publisher of the object.</i>
-----------	---------------------------------------------

Description

Add the optional Publisher property as an attribute to an R object.

Usage

```
publisher(x)

publisher(x, overwrite = TRUE) <- value
```

Arguments

x	A dataset object created with <code>dataset::dataset</code> .
overwrite	If the attributes should be overwritten. In case it is set to FALSE, it gives a warning with the current publisher property instead of overwriting it. Defaults to FALSE.
value	The Publisher as a character set.

Details

The Publisher corresponds to `dct:publisher` and `Publisher` in DataCite. The name of the entity that holds, archives, publishes prints, distributes, releases, issues, or produces the resource. This property will be used to formulate the citation, so consider the prominence of the role. For software, use `Publisher` for the code repository. If there is an entity other than a code repository, that "holds, archives, publishes, prints, distributes, releases, issues, or produces" the code, use the property `Contributor/contributorType/ hostingInstitution` for the code repository.

Value

The Publisher attribute as a character of length 1 is added to x.

See Also

Other Reference metadata functions: `creator()`, `datasource()`, `description()`, `geolocation()`, `identifier()`, `language`, `publication_year()`, `rights()`, `size()`, `version()`

Examples

```
publisher(iris_dataset) <- "American Iris Society"
publisher(iris_dataset)
```

related_item	<i>Create a related item</i>
--------------	------------------------------

Description

Create a **RelatedIdentifier**, attribute, which is recommended for discovery in DataCite.

Usage

```
related_item(
  Identifier,
  Creator,
  Title,
  relatedIdentifierType,
  relationType,
  schemeURI = NA_character_,
  schemeType = NA_character_,
  resourceTypeGeneral = NA_character_,
  PublicationYear = NULL,
  Volume = NULL,
  Issue = NULL,
  Number = NULL,
  numberType = NULL,
  firstPage = NULL,
  lastPage = NULL,
  Publisher = NULL,
  Edition = NULL,
  Contributor = NULL
)
```

Arguments

Identifier	The Identifier is a unique string that identifies a resource. For software, determine whether the identifier is for a specific version of a piece of software, (per the Force11 Software Citation Principles , or for all versions. Similar to <code>dct:title</code> in dublincore .
Creator	The main researchers involved in producing the data, or the authors of the publication, in priority order. To supply multiple creators, repeat this property.
Title	The name(s) or title(s) by which a resource is known. May be the title of a dataset or the name of a piece of software. Similar to <code>dct:title</code> .
relatedIdentifierType	See relatedIdentifierType .
relationType	See relationType .
schemeURI	See schemeURI .
schemeType	See schemeType .

resourceTypeGeneral	The general type of a resource or file.
PublicationYear	The year when the data was or will be made publicly available in YYYY format. See publication_year .
Volume	The volume of the related item (optional).
Issue	The issue number of the related item (optional).
Number	The number of the related item (optional).
numberType	The type of the number (optional).
firstPage	The first page of the related item (optional).
lastPage	The first page of the related item (optional).
Publisher	The name of the entity that holds, archives, publishes prints, distributes, releases, issues, or produces the resource. This property will be used to formulate the citation, so consider the prominence of the role. For software, use Publisher for the code repository. Mandatory in DataCite, and similar to <code>dct:publisher</code> . See publisher .
Edition	The edition of the related item (optional).
Contributor	Recommended for discovery. The institution or person responsible for collecting, managing, distributing, or otherwise contributing to the development of the resource.

Value

a related item.

See Also

Other metadata functions: [datacite\(\)](#), [dublincore\(\)](#)

Examples

```
my_item <- related_item (Identifier = "https://zenodo.org/record/5703222#.YZYkm2DMLIU",
  Creator = person ("Daniel", "Antal", role = "aut"),
  Publisher = "Zenodo",
  PublicationYear = 2022,
  relatedIdentifierType = "DOI",
  relationType = "CompiledBy",
  schemeURI = "URI",
  resourceTypeGeneral = "Dataset")
```

rights	<i>Get/set the Rights of the object.</i>
--------	------------------------------------------

Description

Get/set the optional Rights property as an attribute to an R object.

Usage

```
rights(x)
```

```
rights(x, overwrite = FALSE) <- value
```

Arguments

x	A dataset object created with <code>dataset::dataset</code> .
overwrite	If the Rights attribute should be overwritten. In case it is set to FALSE, it gives a message with the current Rights property instead of overwriting it. Defaults to FALSE.
value	The Rights as a character set.

Details

Rights corresponds to `dct:rights` and `datacite` Rights. Information about rights held in and over the resource. Typically, rights information includes a statement about various property rights associated with the resource, including intellectual property rights.

Value

The Rights attribute as a character of length 1 is added to x.

See Also

Other Reference metadata functions: `creator()`, `datasource()`, `description()`, `geolocation()`, `identifier()`, `language`, `publication_year()`, `publisher()`, `size()`, `version()`

Examples

```
rights(iris_dataset) <- "CC-BY-SA"  
rights(iris_dataset)
```

size	<i>Get/Estimate/Add the Size metadata to an object</i>
------	--------------------------------------------------------

Description

Add the optional DataCite Size property as an attribute to an R object.

Usage

```
size(x)

size(x, overwrite = TRUE) <- value
```

Arguments

x	A dataset object created with <code>dataset::dataset</code> .
overwrite	If a warning should be given when trying to overwrite an existing Size property, defaults to TRUE. The estimated object size in memory is added in SI kB and IEC KiB (legacy Kb) units, rounded to two decimals.
value	A string (size of the dataset in a certain serialisation), or "estimate" produces and estimate with <code>[utils]{object.size}</code> .

Details

Size is an optional property in DataCite 4.4. See: [datacite:Size](#). The object size is estimated with `utils::object.size` and it may differ from the actual serialisation to another file format.

Value

Returns the size metadata field of the DataBibentry of the dataset.

See Also

Other Reference metadata functions: `creator()`, `datasource()`, `description()`, `geolocation()`, `identifier()`, `language`, `publication_year()`, `publisher()`, `rights()`, `version()`

Examples

```
size(iris_dataset) <- "estimate"
size(iris_dataset)
```

subject *Create/add/retrieve a subject*

Description

Create/add/retrieve a subject

Usage

```
subject(x)

subject_create(
  term,
  schemeURI = NULL,
  valueURI = NULL,
  prefix = NULL,
  subjectScheme = NULL,
  classificationCode = NULL
)

subject(x, overwrite = FALSE) <- value

is.subject(x)
```

Arguments

x	A dataset object created with <code>dataset::dataset</code> .
term	A subject term, for example, "Data sets".
schemeURI	The URI of the subject identifier scheme, for example "http://id.loc.gov/authorities/subjects"
valueURI	The URI of the subject term. "https://id.loc.gov/authorities/subjects/sh2018002256"
prefix	An abbreviated prefix of a scheme URI, for example, "lcch:" representing "http://id.loc.gov/authorities/subjects". Widely used namespaces (schemes) have conventional abbreviations.
subjectScheme	The name of the subject scheme or classification code or authority if one is used. It is a namespace.
classificationCode	The classificationCode subproperty may be used for subject schemes, like ANZSRC, which do not have valueURIs for each subject term.
overwrite	If the attributes should be overwritten. In case it is set to FALSE, it gives a message with the current Subject property instead of overwriting it. Defaults to FALSE.
value	A subject field created by <code>subject</code> .

Value

The Subject field of the dataset's attributes.

A named list with the subject term, the subject scheme, URIs and prefix.

Examples

```
subject(iris_dataset,
        overwrite = TRUE) <- subject_create(
  term = "Irises (plants)",
  schemeURI = "http://id.loc.gov/authorities/subjects",
  valueURI = "https://id.loc.gov/authorities/subjects/sh85068079",
  subjectScheme = "LCCH",
  prefix = "lcch:")
subject(iris_dataset)
```

var_labels

Get / Set a variable labels in a dataset

Description

Get / Set a variable labels in a dataset

Usage

```
var_labels(x)

set_var_labels(x, value)

## S3 method for class 'dataset'
var_labels(x)

## S3 method for class 'dataset'
set_var_labels(x, value)
```

Arguments

x A dataset object created with `dataset::dataset`.

value A character vector for labelling the variables with `set_var_labels`.

Examples

```
relabelled <- set_var_labels(
  iris_dataset,
  c(Sepal.Length="The sepal length measured in centimeters.",
    Sepal.Width="The sepal width measured in centimeters.",
    Species="The species of the iris observed.")
)
var_labels(relabelled)
```

version	<i>Get/set the version of the object.</i>
---------	-------------------------------------------

Description

Get/set the optional `Version` property as an attribute to an R object.

Usage

```
version(x)

version(x, overwrite = FALSE) <- value
```

Arguments

<code>x</code>	An R object, such as a <code>data.frame</code> , a <code>tibble</code> , or a <code>data.table</code> .
<code>overwrite</code>	If the <code>Version</code> attribute should be overwritten. In case it is set to <code>FALSE</code> , it gives a message with the current <code>Version</code> property instead of overwriting it. Defaults to <code>TRUE</code> when the attribute is set to <code>value</code> regardless of previous setting.
<code>value</code>	The <code>Version</code> as a character set.

Details

`Version` is an optional property in DataCite 4.4. See: [datacite:Size](#). It is not part of the "core" Dublin Core terms, but ... [Dublin Core metadata terms](#).

Value

The `Version` attribute as a character of length 1 is added to `x`.

See Also

Other Reference metadata functions: [creator\(\)](#), [datasource\(\)](#), [description\(\)](#), [geolocation\(\)](#), [identifier\(\)](#), [language](#), [publication_year\(\)](#), [publisher\(\)](#), [rights\(\)](#), [size\(\)](#)

Examples

```
iris_dataset <- iris
version(iris_dataset) <- "1.0"
version(iris_dataset)
```

Index

* Reference metadata functions

creator, 5
datasource, 10
description, 12
geolocation, 16
identifier, 17
language, 19
publication_year, 20
publisher, 21
rights, 24
size, 25
version, 28

* datasets

iris_dataset, 18

* metadata functions

datacite, 6
dublincore, 13
related_item, 22

as_datacite (datacite), 6

as_dataset, 2

as_dublincore (dublincore), 13

bibentry, 8, 9, 15

creator, 5, 11, 13, 14, 16, 18–21, 24, 25, 28

creator<- (creator), 5

datacite, 6, 14, 15, 17, 23, 24

dataset, 5, 8–10, 12, 13, 16, 20, 21, 24–27

dataset (as_dataset), 2

dataset_bibentry, 8

dataset_title, 9, 14

dataset_title<- (dataset_title), 9

datasource, 5, 10, 13, 16, 18–21, 24, 25, 28

datasource<- (datasource), 10

DataSet, 11

DataSet_update (DataSet), 11

describe, 12

description, 5, 11, 12, 15, 16, 18–21, 24, 25, 28

description<- (description), 12

dublincore, 7, 8, 13, 22, 23

geolocation, 5, 7, 11, 13, 16, 18–21, 24, 25, 28

geolocation<- (geolocation), 16

identifier, 5, 11, 13, 14, 16, 17, 19–21, 24, 25, 28

identifier<- (identifier), 17

iris, 18

iris_dataset, 18

is.datacite (datacite), 6

is.dataset (as_dataset), 2

is.dublincore (dublincore), 13

is.subject (subject), 26

language, 5, 7, 11, 13, 16, 18, 19, 20, 21, 24, 25, 28

language<- (language), 19

person, 4

print.dataset (as_dataset), 2

publication_year, 5, 7, 11, 13, 16, 18, 19, 20, 21, 23–25, 28

publication_year<- (publication_year), 20

publisher, 5, 7, 11, 13, 14, 16, 18–20, 21, 23–25, 28

publisher<- (publisher), 21

related_item, 8, 15, 22

rights, 4, 5, 7, 11, 13, 15, 16, 18–21, 24, 25, 28

rights<- (rights), 24

set_var_labels (var_labels), 27

size, 5, 11, 13, 16, 18–21, 24, 25, 28

size<- (size), 25

subject, 7, 14, 26, 26

subject<- (subject), 26

subject_create, [7](#)
subject_create (subject), [26](#)
summary.dataset (as_dataset), [2](#)

utils::object.size, [25](#)
utils::person, [5](#)

var_labels, [27](#)
version, [5](#), [7](#), [11](#), [13](#), [16](#), [18–21](#), [24](#), [25](#), [28](#)
version<- (version), [28](#)