

# Package ‘oro.nifti’

August 10, 2022

**Version** 0.11.4

**Title** Rigorous - 'NIFTI' + 'ANALYZE' + 'AFNI' : Input / Output

**Description** Functions for the input/output and visualization of medical imaging data that follow either the 'ANALYZE', 'NIFTI' or 'AFNI' formats. This package is part of the Rigorous Analytics bundle.

**Depends** R (>= 2.14.0)

**Suggests** XML, testthat, covr, knitr, rmarkdown, rtticles

**Imports** stats, bitops, splines, graphics, grDevices, methods, utils, abind, RNifti (>= 0.9.0)

**Enhances** fmri, oro.dicom

**License** BSD\_3\_clause + file LICENSE

**BugReports** <https://github.com/bjw34032/oro.nifti/issues>

**URL** <https://rigorousanalytics.blogspot.com>

**Collate** 'auditTrail.R' 'niftiS4.R' 'analyzeS4.R' 'afniS4.R' 'is.R' 'nifti\_assign.R' 'coerce.R' 'writeS4.R' 'convert\_anlz.R' 'convert\_nifti.R' 'cal\_img.R' 'drop\_img\_dim.R' 'hotmetal.R' 'miscellaneous.R' 'plot.R' 'slice.R' 'slice\_overlay.R' 'blend.R' 'readS4.R' 'remove.R' 'tim\_colors.R' 'transform.R' 'wrappers.R' 'onefile.R' 'voxdim.R' 'anlz\_Operators.R' 'Operators.R' 'zero\_trans.R' 'aux\_file.R' 'cal\_max.R' 'cal\_min.R' 'descrip.R' 'glmax.R' 'glmin.R' 'pixdim.R' 'qform\_code.R' 'scl\_inter.R' 'scl\_slope.R' 'sform\_code.R' 'vox\_offset.R' 'bitpix.R' 'data\_type.R' 'datatype.R' 'db\_name.R' 'dim\_.R' 'dim\_info.R' 'extender.R' 'extents.R' 'img\_data.R' 'intent\_code.R' 'intent\_name.R' 'intent\_p1.R' 'intent\_p2.R' 'intent\_p3.R' 'magic.R' 'qoffset\_x.R' 'qoffset\_y.R' 'qoffset\_z.R' 'quatern\_b.R' 'quatern\_c.R' 'quatern\_d.R' 'quaternion.R' 'regular.R' 'session\_error.R' 'sizeof\_hdr.R' 'slice\_code.R' 'slice\_duration.R' 'slice\_end.R' 'slice\_start.R' 'srow\_x.R' 'srow\_y.R' 'srow\_z.R' 'toffset.R' 'xyzt\_units.R' 'cal\_units.R' 'compressed.R' 'dim\_un0.R' 'exp\_date.R' 'exp\_time.R' 'field\_skip.R' 'funused1.R'

'funused2.R' 'funused3.R' 'generated.R' 'hist\_un0.R'  
 'hkey\_un0.R' 'niftiImage\_class.R' 'nii2oro.R' 'oro2nii.R'  
 'omax.R' 'omin.R' 'orient.R' 'origin.R' 'patient\_id.R'  
 'scannum.R' 'smax.R' 'smin.R' 'start\_field.R' 'unused1.R'  
 'verified.R' 'views.R' 'vols\_added.R' 'vox\_units.R' 'voxres.R'  
 'img\_length.R' 'zzz.R' 'zzz\_niftiImage.R'

**RoxygenNote** 7.2.1

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Brandon Whitcher [aut, cre],  
 Volker Schmid [aut],  
 Andrew Thornton [aut],  
 Karsten Tabelow [ctb],  
 Jon Clayden [ctb],  
 John Muschelli [aut]

**Maintainer** Brandon Whitcher <bwhitcher@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-08-10 16:10:02 UTC

## R topics documented:

afni-class	5
anzl	7
anzl-class	8
anzl-nifti-ops	10
as.anzl	11
as.nifti	12
Audit Trails	12
audit.trail-methods	16
aux_file-methods	17
bitpix-methods	19
blend	20
calibrateImage	21
cal_max-methods	22
cal_min-methods	24
cal_units-methods	26
coerce-methods	27
compressed-methods	28
Convert ANALYZE Codes	29
Convert NIFTI Codes	30
convert.scene	31
datatype-methods	32
data_type-methods	33
db_name-methods	35
descrip-methods	36
dim_-methods	38

dim_info-methods . . . . .	39
dim_un0-methods . . . . .	40
dropImageDimension . . . . .	41
exp_date-methods . . . . .	42
exp_time-methods . . . . .	43
extender-methods . . . . .	44
extents-methods . . . . .	45
field_skip-methods . . . . .	46
funused1-methods . . . . .	47
funused2-methods . . . . .	48
funused3-methods . . . . .	49
generated-methods . . . . .	50
glmax-methods . . . . .	51
glmin-methods . . . . .	52
hist_un0-methods . . . . .	53
hkey_un0-methods . . . . .	54
hotmetal . . . . .	55
image-methods . . . . .	56
img_data-methods . . . . .	58
img_length . . . . .	59
integerTranslation . . . . .	59
intent_code-methods . . . . .	60
intent_name-methods . . . . .	61
intent_p1-methods . . . . .	63
intent_p2-methods . . . . .	64
intent_p3-methods . . . . .	65
internalImage-class . . . . .	67
is.afni . . . . .	67
is.anlz . . . . .	68
is.nifti . . . . .	68
magic-methods . . . . .	69
nifti . . . . .	70
nifti-class . . . . .	72
nifti-operators . . . . .	74
niftiAuditTrail-class . . . . .	75
niftiExtension-class . . . . .	76
niftiExtensionSection-class . . . . .	76
niftiImage-class . . . . .	77
nifti_assign-methods . . . . .	77
nii2oro . . . . .	78
nsli . . . . .	78
omax-methods . . . . .	79
omin-methods . . . . .	80
onefile . . . . .	81
orient-methods . . . . .	81
orientation-methods . . . . .	82
origin-methods . . . . .	83
oro2nii . . . . .	84

orthographic-methods . . . . .	85
overlay-methods . . . . .	88
patient_id-methods . . . . .	92
performPermutation . . . . .	94
pixdim-methods . . . . .	94
qform_code-methods . . . . .	96
qoffset_x-methods . . . . .	97
qoffset_y-methods . . . . .	98
qoffset_z-methods . . . . .	100
quaternion2rotation . . . . .	101
quatern_b-methods . . . . .	102
quatern_c-methods . . . . .	104
quatern_d-methods . . . . .	105
readAFNI . . . . .	106
readANALYZE . . . . .	107
readNIfTI . . . . .	109
regular-methods . . . . .	111
reorient . . . . .	112
resetSlopeIntercept . . . . .	113
rmnii.gz . . . . .	113
scannum-methods . . . . .	114
scl_inter-methods . . . . .	115
scl_slope-methods . . . . .	116
session_error-methods . . . . .	118
sform_code-methods . . . . .	119
sizeof_hdr-methods . . . . .	120
slice-methods . . . . .	121
slice_code-methods . . . . .	124
slice_duration-methods . . . . .	125
slice_end-methods . . . . .	126
slice_overlay-methods . . . . .	128
slice_start-methods . . . . .	132
smax-methods . . . . .	134
smin-methods . . . . .	135
srow_x-methods . . . . .	136
srow_y-methods . . . . .	137
srow_z-methods . . . . .	138
start_field-methods . . . . .	139
tim.colors . . . . .	140
toffset-methods . . . . .	141
translateCoordinate . . . . .	142
unused1-methods . . . . .	143
verified-methods . . . . .	144
views-methods . . . . .	145
vols_added-methods . . . . .	146
voxdim . . . . .	147
voxres . . . . .	148
vox_offset-methods . . . . .	148

vox_units-methods . . . . .	150
writeAFNI-methods . . . . .	151
writeANALYZE-methods . . . . .	152
writeNIfTI-methods . . . . .	154
xyzt2space . . . . .	157
xyzt_units-methods . . . . .	159

<b>Index</b>	<b>161</b>
--------------	------------

---

afni-class	<i>Class "afni"</i>
------------	---------------------

---

## Description

The AFNI class for medical imaging data.

## Usage

```
## S4 method for signature 'afni'
show(object)
```

## Arguments

object            An object of class afni.

## Objects from the Class

Objects can be created by calls of the form `new("afni", data, dim, dimnames, ...)`.

## Slots

```
.Data: Object of class "array" contains the imaging data
DATASET_RANK: Object of class "integer"
DATASET_DIMENSIONS: Object of class "integer"
TYPESTRING: Object of class "character"
SCENE_DATA: Object of class "integer"
ORIENT_SPECIFIC: Object of class "integer"
ORIGIN: Object of class "numeric"
DELTA: Object of class "numeric"
TAXIS_NUMS: Object of class "integer"
TAXIS_FLOATS: Object of class "numeric"
TAXIS_OFFSETS: Object of class "numeric"
IDCODE_STRING: Object of class "character"
IDCODE_DATE: Object of class "character"
BYTEORDER_STRING: Object of class "character"
```

BRICK\_STATS: Object of class "numeric"  
BRICK\_TYPES: Object of class "integer"  
BRICK\_FLOAT\_FACS: Object of class "numeric"  
BRICK\_LABS: Object of class "character"  
BRICK\_STATAUX: Object of class "numeric"  
STAT\_AUX: Object of class "numeric"  
HISTORY\_NOTE: Object of class "character"  
NOTES\_COUNT: Object of class "integer"  
NOTE\_NUMBER: Object of class "character"  
TAGALIGN\_MATVEC: Object of class "numeric"  
VOLREG\_MATVEC: Object of class "array"  
VOLREG\_ROTCOM: Object of class "character"  
VOLREG\_CENTER\_OLD: Object of class "numeric"  
VOLREG\_CENTER\_BASE: Object of class "numeric"  
VOLREG\_ROT\_PARENT\_IDCODE: Object of class "character"  
VOLREG\_ROT\_PARENT\_NAME: Object of class "character"  
VOLREG\_GRID\_PARENT\_IDCODE: Object of class "character"  
VOLREG\_GRID\_PARENT\_NAME: Object of class "character"  
VOLREG\_INPUT\_IDCODE: Object of class "character"  
VOLREG\_INPUT\_NAME: Object of class "character"  
VOLREG\_BASE\_IDCODE: Object of class "character"  
VOLREG\_BASE\_NAME: Object of class "character"  
VOLREG\_ROTCOM\_NUM: Object of class "integer"  
IDCODE\_ANAT\_PARENT: Object of class "character"  
T03D\_ZPAD: Object of class "integer"  
IDCODE\_WARP\_PARENT: Object of class "character"  
WARP\_TYPE: Object of class "integer"  
WARP\_DATA: Object of class "numeric"  
MARKS\_XYZ: Object of class "numeric"  
MARKS\_LAB: Object of class "character"  
MARKS\_HELP: Object of class "character"  
MARKS\_FLAGS: Object of class "integer"  
TAGSET\_NUM: Object of class "integer"  
TAGSET\_FLOATS: Object of class "numeric"  
TAGSET\_LABELS: Object of class "character"  
LABEL\_1: Object of class "character"  
LABEL\_2: Object of class "character"  
DATASET\_NAME: Object of class "character"  
DATASET\_KEYWORDS: Object of class "character"  
BRICK\_KEYWORDS: Object of class "character"

**Extends**

Class `"array"`, from data part.  
 Class `"matrix"`, by class `"array"`, distance 2, with explicit test and coerce.  
 Class `"structure"`, by class `"array"`, distance 2.  
 Class `"vector"`, by class `"array"`, distance 3, with explicit coerce.  
 Class `"vector"`, by class `"array"`, distance 5, with explicit test and coerce.

**Author(s)**

Karsten Tabelow <karsten.tabelow@wias-berlin.de>

**References**

AFNI  
<http://afni.nimh.nih.gov/pub/dist/src/README.attributes>

**See Also**

[nifti](#), [anlz](#)

**Examples**

```
showClass("afni")
```

---

anlz *Constructor for Analyze*

---

**Description**

Constructor for Analyze class objects.

**Usage**

```
anlz(img = array(0, dim = rep(1, 4)), dim, datatype = 2, ...)
```

**Arguments**

<code>img</code>	is a multidimensional array of data.
<code>dim</code>	is the dimension of the data (default = missing).
<code>datatype</code>	is an integer that denotes the type of data contained in each voxel. See the function <code>convert.datatype.anlz</code> or the ANALYZE documentation for more details.
<code>...</code>	allows for additional ‘slots’ to be specified.

**Value**

An object of class `anlz`.

**Author(s)**

Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>

**See Also**

[anlz](#), [nifti](#), [nifti](#), [convert.datatype.anlz](#)

**Examples**

```
aim <- anlz() # default
```

---

anlz-class

*Class "anlz"*

---

**Description**

The ANALYZE class for medical imaging data.

**Usage**

```
## S4 method for signature 'anlz'
show(object)
```

**Arguments**

object            An object of class anlz.

**Objects from the Class**

Objects can be created by calls of the form `new("anlz", data, dim, dimnames, ...)` or by calling the `anlz` function.

**Slots**

```
.Data: Object of class "array" contains the imaging data
sizeof_hdr: Object of class "numeric" contains the size of the header (= 348)
data_type: Object of class "character"
db_name: Object of class "character"
extents: Object of class "numeric"
session_error: Object of class "numeric"
```



regular: Object of class "character"  
hkey\_un0: Object of class "character"  
dim\_: Object of class "vector" contains the dimensions of the imaging data  
vox\_units: Object of class "character"  
cal\_units: Object of class "character"  
unused1: Object of class "numeric"  
datatype: Object of class "numeric"  
bitpix: Object of class "numeric" contains the number of bits per voxel (pixel)  
dim\_un0: Object of class "numeric"  
pixdim: Object of class "vector" contains the real-world dimensions of the imaging data  
vox\_offset: Object of class "numeric"  
funused1: Object of class "numeric"  
funused2: Object of class "numeric"  
funused3: Object of class "numeric"  
cal\_max: Object of class "numeric" contains the maximum display intensity  
cal\_min: Object of class "numeric" contains the minimum display intensity  
compressed: Object of class "numeric"  
verified: Object of class "numeric"  
glmax: Object of class "numeric"  
glmin: Object of class "numeric"  
descrip: Object of class "character"  
aux\_file: Object of class "character"  
orient: Object of class "character"  
origin: Object of class "numeric"  
generated: Object of class "character"  
scannum: Object of class "character"  
patient\_id: Object of class "character"  
exp\_date: Object of class "character"  
exp\_time: Object of class "character"  
hist\_un0: Object of class "character"  
views: Object of class "numeric"  
vols\_added: Object of class "numeric"  
start\_field: Object of class "numeric"  
field\_skip: Object of class "numeric"  
omax: Object of class "numeric"  
omin: Object of class "numeric"  
smax: Object of class "numeric"  
smin: Object of class "numeric"

**Extends**

Class `"array"`, from data part.  
Class `"matrix"`, by class `"array"`, distance 2, with explicit test and coerce.  
Class `"structure"`, by class `"array"`, distance 2.  
Class `"vector"`, by class `"array"`, distance 3, with explicit coerce.  
Class `"vector"`, by class `"array"`, distance 5, with explicit test and coerce.

**Methods**

**image** signature(x = "anlz"): displays the image(s).  
**show** signature(object = "anlz"): prints out a summary of the imaging data.

**Author(s)**

Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>

**See Also**

`nifti`, `niftiExtension`

**Examples**

```
showClass("anlz")
```

---

anlz-nifti-ops

*Operations for Objects in the ANALYZE and NIFTI classes*

---

**Description**

Overloaded operators for anlz and nifti objects

**Usage**

```
## S4 method for signature 'anlz,anlz'  
Ops(e1, e2)  
  
## S4 method for signature 'anlz,numeric'  
Ops(e1, e2)  
  
## S4 method for signature 'numeric,anlz'  
Ops(e1, e2)
```

```
## S4 method for signature 'nifti,anlz'
Ops(e1, e2)

## S4 method for signature 'anlz,nifti'
Ops(e1, e2)
```

### Arguments

```
e1          object
e2          object
```

### Author(s)

John Muschelli <muschellij2@gmail.com>

### Examples

```
img01 <- anlz(array(1:64, c(4,4,4,1)), datatype=4)
img02 <- anlz(array(64:1, c(4,4,4,1)), datatype=4)
is.anlz(img01 + img02)
is.anlz(sqrt(2) * img01)
is.anlz(img02 / pi)
```

---

as.anlz

*as.anlz*

---

### Description

Internal function that converts multidimensional arrays to ANALYZE class objects.

### Usage

```
as.anlz(from, value = NULL, verbose = FALSE)
```

### Arguments

```
from          is the object to be converted.
value         is the nifti class object to use as a template for various ANALYZE header
              information.
verbose       is a logical variable (default = FALSE) that allows text-based feedback during
              execution of the function.
```

### Value

An object of class `anlz`.

**Author(s)**

Andrew Thornton <zeripath@users.sourceforge.net>,  
 Brandon Whitcher <bwhitcher@gmail.com>

---

 as.nifti

*as.nifti*


---

**Description**

Internal function that converts multidimensional arrays to NIFTI class objects.

**Usage**

```
as.nifti(from, value = NULL, verbose = FALSE)
```

**Arguments**

from	is the object to be converted.
value	is the anlz class object to use as a template for various NIFTI header information.
verbose	is a logical variable (default = FALSE) that allows text-based feedback during execution of the function.

**Value**

An object of class nifti.

**Author(s)**

Andrew Thornton <zeripath@users.sourceforge.net>.  
 Brandon Whitcher <bwhitcher@gmail.com>

---

 Audit Trails

*Facilitate the Creation and Modification of Audit Trails*


---

**Description**

Facilitate the creation and modification of audit trails for NIFTI class objects.

**Usage**

```
oro.nifti.info(type)

enableAuditTrail()

getLastCallWithName(functionName)

newAuditTrail()

niftiExtensionToAuditTrail(
  nim,
  workingDirectory = NULL,
  filename = NULL,
  call = NULL
)

niftiAuditTrailSystemNode(
  type = "system-info",
  workingDirectory = NULL,
  filename = NULL,
  call = NULL
)

niftiAuditTrailSystemNodeEvent(
  trail,
  type = NULL,
  call = NULL,
  workingDirectory = NULL,
  filename = NULL,
  comment = NULL
)

niftiAuditTrailCreated(
  history = NULL,
  call = NULL,
  workingDirectory = NULL,
  filename = NULL
)

niftiAuditTrailEvent(trail, type = NULL, call = NULL, comment = NULL)
```

**Arguments**

type	An identifier to add some meaning to the event.
functionName	The name of a function on the call stack.
nim	is an object of class <code>niftiAuditTrail</code> or can be converted to such.
workingDirectory	The working directory associated with the 'filename'.

filename	The filename associated with the nifti object.
call	A call, function name in the call-stack or a string.
trail	The XMLAbstractNode representing the audit trail or the niftiAuditTrail object with a trail that will be amended.
comment	Some textual comment
history	An XMLAbstractNode to store historical events for inclusion in the 'trail'.

## Details

The function `oro.nifti.info` is used to find the ecode or the XML namespace relevant to the audit trail.

The function `enableAuditTrail` is turned "off" by default to minimize package dependencies. Should one wish to turn "on" the audit trail functionality, then one should set the option `NIFTI.audit.trail` to `TRUE` and call the function `enableAuditTrail`. Setting the option `NIFTI.audit.trail` to `FALSE` will disable the audit trail.

The function `newAuditTrail` returns an `XMLAbstractNode` representing the root node of an audit trail. This is mostly intended as an internal function.

The function `niftiExtensionToAuditTrail` takes an object representing a NIFTI object, casts it as a `niftiAuditTrail` and checks if there is an extension (a `niftiExtensionSection`) with ecode equal to `oro.nifti.info("ecode")`; i.e. has a extension with data representing a serialized audit trail. The function will then strip the object of this extension parsing the serialized edata into an audit trail and adding a 'read' event to the trail.

The function `niftiAuditTrailToExtension` takes a `niftiAuditTrail` and returns a `niftiExtensionSection` with edata containing the serialized form of the audit trail after adding a 'saved' event to the trail.

The function `niftiAuditTrailSystemNodeEvent` adds an element with name equal to `type` to the trail. It uses the `niftiAuditTrailSystemNode` function to create the node.

The function `niftiAuditTrailSystemNode` is an internal function creating an `XMLAbstractNode` element with name `type` and attributes giving information about the R system and library. The `filename` and `call` will also be added as attributes if available.

The function `niftiAuditTrailEvent` adds an element with name `event` to the trail. The arguments `type`, `filename`, `call` are added as attributes and the `comment` is the text value of the element.

The function `niftiAuditTrailCreated` will create a new audit trail containing a system node element created with the child `history` with the contents `history`. If the last element of the `history` given is an event with `type="processing"`, then this node will be removed from the `history` and its `call` attribute will be used as the value of the `call` attribute on the created node.

The function `getLastCallWithName` will search the call stack for a call of the function `functionName`, returning last call to that function if possible. It will default to the call of the function which called the function which called `getLastCallWithName` if there was no such call (and if there was no such call it will return the call of itself).

## Note

These functions are mostly intended to be used internally in order to document the changes that occur to NIFTI objects due to functions that are audit-trail aware. However, as the precise manner

in which these functions are used is not documented anywhere else, we shall proceed to describe which functions are audit-trail aware and how they interact with the audit trail.

`as.nifti` and its S4 alias `as(nim, "nifti")` will always produce `niftiAuditTrail` objects if the functionality is turned on. The function `niftiAuditTrailCreated` will be used and if an exemplar object is provided (e.g., `as.nifti(array, niftiExemplar)`) then the trail of the exemplar will be used as the history.

`readNIFTI` and `writeNIFTI` also always produce `niftiAuditTrail` objects if the functionality is turned on. The functions `niftiExtensionToAuditTrail` and `niftiAuditTrailToExtension` are used internally by these functions to facilitate this behaviour.

### Author(s)

Andrew Thornton <zeripath@users.sourceforge.net> and Brandon Whitcer <bwhitcer@gmail.com>

### Examples

```
## A good example of the use of these functions is shown by this
## wrapper function which takes a function fun(nim, ...) returning
## lists of arrays which are nifti-ized using as(...)
options("niftiAuditTrail"=TRUE)
enableAuditTrail()

wrapper <- function(functionToWrap, nameOfCallingFunction, nim, ...) {
  if (!is(nim, "nifti"))
    nim <- as(nim, "nifti")

  if (is(nim, "niftiAuditTrail")) {
    ## This will force as(...) to set the call which created the
    ## results to the calling function's call rather than
    ## as(result, nifti) as it would otherwise do
    slot(nim, "trail") <- niftiAuditTrailEvent(slot(nim, "trail"), "processing",
                                              nameOfCallingFunction)
  }

  result <- functionToWrap(nim, ...)
  as(result, "nifti") <- nim
  return(result)
}

## An example of how wrapper is used follows:
functionToWrap <- function(ignored, x, y) {
  return (array(1, dim=c(x,y)))
}

## The nifti-ized form
niftiizedForm <- function(nim,...) {
  return(wrapper(functionToWrap, "niftiizedForm", nim, ...))
}

## Not run:
if (isTRUE(getOption("niftiAuditTrail"))) {
```

```
    print(slot(as.nifti(functionToWrap(nifti()), 4, 4), nifti()), "trail")
    print(slot(niftiizedForm(nifti()), 4, 4), "trail")
  }

## End(Not run)
```

---

audit.trail-methods    *Extract or Replace NIFTI Audit Trail*

---

## Description

Operators that act on the audit trail (XML) in the NIFTI header.

## Usage

```
audit.trail(object)

## S4 method for signature 'nifti'
audit.trail(object)

audit.trail(object) <- value

## S4 replacement method for signature 'nifti'
audit.trail(object) <- value
```

## Arguments

object	is of class nifti.
value	Value to assign to trail slot

## Methods

**object = "nifti"** Extract or replace NIFTI audit trail.

## Author(s)

Andrew Thornton <zeripath@users.sourceforge.net>



---

aux\_file-methods      *Extract Image Attribute aux\_file*

---

## Description

Methods that act on the aux\_file field in the NIFTI/ANALYZE header.

## Usage

```
aux_file(object)

## S4 method for signature 'nifti'
aux_file(object)

## S4 method for signature 'anlz'
aux_file(object)

aux_file(object) <- value

## S4 replacement method for signature 'nifti'
aux_file(object) <- value

## S4 replacement method for signature 'anlz'
aux_file(object) <- value

aux.file(object)

## S4 method for signature 'nifti'
aux.file(object)

## S4 method for signature 'anlz'
aux.file(object)

aux.file(object) <- value

## S4 replacement method for signature 'nifti'
aux.file(object) <- value

## S4 replacement method for signature 'anlz'
aux.file(object) <- value

## S4 method for signature 'niftiImage'
aux_file(object)

## S4 replacement method for signature 'niftiImage'
aux_file(object) <- value
```

**Arguments**

object is an object of class nifti or anlz.  
 value is the value to assign to the aux\_file field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
 Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
 NIFTI-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
## Not run:
url <- "http://nifti.nimh.nih.gov/nifti-1/data/avg152T1_RL_nifti.nii.gz"
urlfile <- file.path(system.file("nifti", package="oro.nifti"),
                    "mniRL.nii.gz")
download.file(url, urlfile, quiet=TRUE)

## End(Not run)
options("niftiAuditTrail"=FALSE)

urlfile <- file.path(system.file("nifti", package="oro.nifti"),
                    "mniRL.nii.gz")
mniRL <- readNIFTI(urlfile)
aux.file(mniRL)
aux.file(mniRL) <- "avg152T1_RL_nifti"
aux.file(mniRL)
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
aux_file(img)
aux_file(img) = "hey"
stopifnot(aux_file(img) == "hey")
```

---

bitpix-methods                      *Extract Image Attribute* bitpix

---

## Description

Methods that act on the bitpix field in the NIFTI/ANALYZE header.

## Usage

```
bitpix(object)

## S4 method for signature 'nifti'
bitpix(object)

## S4 method for signature 'anlz'
bitpix(object)

bitpix(object) <- value

## S4 replacement method for signature 'nifti'
bitpix(object) <- value

## S4 replacement method for signature 'anlz'
bitpix(object) <- value

## S4 method for signature 'niftiImage'
bitpix(object)
```

## Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the bitpix field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
bitpix(img)
```

---

blend

---

*Merge Two NIFTI or ANALYZE Volumes*


---

**Description**

Two volumes of medical imaging data are merged together in the superior-inferior (or  $z$ ) direction. One assumes that there is at least one slice that overlaps between the two volumes.

**Usage**

```
blendVolumes(x, y, seqX, seqY, method = "linear")

## S4 method for signature 'nifti,nifti'
blend(x, y, seqX, seqY, method = "linear")

## S4 method for signature 'anlz,anlz'
blend(x, y, seqX, seqY, method = "linear")

## S4 method for signature 'anlz,nifti'
blend(x, y, seqX, seqY, method = "linear")

## S4 method for signature 'nifti,anlz'
blend(x, y, seqX, seqY, method = "linear")
```

**Arguments**

`x, y` are objects of class `nifti` or `anlz`.  
`seqX, seqY` are vectors that provide the  $z$ -coordinate values for the two imaging volumes.  
`method` is the type of weighing to use when combining information where there is an overlap (default = "linear").

**Value**

A single volume that blends the voxel-wise information from `x` and `y`.

**Methods**

```
x = "nifti", y = "nifti" Merge x and y.
x = "anlz", y = "anlz" Merge x on y.
x = "nifti", y = "anlz" Merge x on y.
x = "anlz", y = "nifti" Merge x and y.
```

**Author(s)**

Brandon Whitcher <bwhitcher@gmail.com>

**See Also**

[image-methods](#), [overlay-methods](#)

---

calibrateImage	<i>Set Minimum/Maximum Values for NIfTI data</i>
----------------	--

---

**Description**

Rescales image `cal_max` and `cal_min` slots to be the max and min, respectively, of an object of class `nifti`, with `na.rm = TRUE`. This is so that when images are rendered/written, the values correspond to those in the array (stored in `.Data` slot) are plotted on correct greyscale and no error is given by `writeNIfTI`.

**Usage**

```
calibrateImage(img, infok = TRUE)
```

```
cal_img(img, infok = TRUE)
```

**Arguments**

`img` is a `nifti` object.

`infok` is a logical value whether or not `Inf` and `-Inf` are acceptable (default = `TRUE`). If `FALSE` and max or min is infinity, then `cal_min` or `cal_max` is set to infinity (negative or positive), respectively.

**Value**

An object of class `nifti`.

**Author(s)**

John Muschelli <muschellij2@gmail.com>

---

cal\_max-methods      *Extract Image Attribute cal\_max*

---

### Description

Methods that act on the cal\_max field in the NIFTI/ANALYZE header.

### Usage

```
cal_max(object)

## S4 method for signature 'nifti'
cal_max(object)

## S4 method for signature 'anlz'
cal_max(object)

cal_max(object) <- value

## S4 replacement method for signature 'nifti'
cal_max(object) <- value

## S4 replacement method for signature 'anlz'
cal_max(object) <- value

cal.max(object)

## S4 method for signature 'nifti'
cal.max(object)

## S4 method for signature 'anlz'
cal.max(object)

cal.max(object) <- value

## S4 replacement method for signature 'nifti'
cal.max(object) <- value

## S4 replacement method for signature 'anlz'
cal.max(object) <- value

## S4 method for signature 'niftiImage'
cal.max(object)

## S4 replacement method for signature 'niftiImage'
cal.max(object) <- value
```

```
## S4 method for signature 'niftiImage'
cal_max(object)

## S4 replacement method for signature 'niftiImage'
cal_max(object) <- value
```

### Arguments

object is an object of class nifti or anlz.  
value is the value to assign to the cal\_max field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

### Examples

```
## Not run:
url <- "http://nifti.nimh.nih.gov/nifti-1/data/avg152T1_LR_nifti.nii.gz"
urlfile <- file.path(system.file("nifti", package="oro.nifti"),
                    "mniLR.nii.gz")
download.file(url, urlfile, quiet=TRUE)

## End(Not run)
urlfile <- file.path(system.file("nifti", package="oro.nifti"),
                    "mniLR.nii.gz")
mniLR <- readNIFTI(urlfile)
cal.max(mniLR)
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
cal.max(img)
cal_max(img)
cal.max(img) = 2500
stopifnot(cal_max(img) == 2500)
cal_max(img) = 2500
cal.min(img)
cal.min(img) = 2
stopifnot(cal_min(img) == 2)
cal_min(img)
```

```
cal_min(img) = 0
stopifnot(cal_min(img) == 0)
```

---

cal\_min-methods      *Extract Image Attribute cal\_min*

---

### Description

Methods that act on the cal\_min field in the NIFTI/ANALYZE header.

### Usage

```
cal_min(object)

## S4 method for signature 'nifti'
cal_min(object)

## S4 method for signature 'anlz'
cal_min(object)

cal_min(object) <- value

## S4 replacement method for signature 'nifti'
cal_min(object) <- value

## S4 replacement method for signature 'anlz'
cal_min(object) <- value

cal.min(object)

## S4 method for signature 'nifti'
cal.min(object)

## S4 method for signature 'anlz'
cal.min(object)

cal.min(object) <- value

## S4 replacement method for signature 'nifti'
cal.min(object) <- value

## S4 replacement method for signature 'anlz'
cal.min(object) <- value

## S4 method for signature 'niftiImage'
cal.min(object)
```



```
## S4 replacement method for signature 'niftiImage'  
cal.min(object) <- value  
  
## S4 method for signature 'niftiImage'  
cal_min(object)  
  
## S4 replacement method for signature 'niftiImage'  
cal_min(object) <- value
```

### Arguments

object            is an object of class nifti or anlz.  
value            is the value to assign to the cal\_min field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

### Examples

```
## Not run:  
url <- "http://nifti.nimh.nih.gov/nifti-1/data/avg152T1_LR_nifti.nii.gz"  
urlfile <- file.path(system.file("nifti", package="oro.nifti"),  
                      "mniLR.nii.gz")  
download.file(url, urlfile, quiet=TRUE)  
  
## End(Not run)  
urlfile <- file.path(system.file("nifti", package="oro.nifti"),  
                      "mniLR.nii.gz")  
mniLR <- readNIFTI(urlfile)  
cal.min(mniLR)
```

---

cal\_units-methods      *Extract Image Attribute cal\_units*

---

### Description

Methods that act on the `cal_units` field in the NIFTI/ANALYZE header.

### Usage

```
cal_units(object)

## S4 method for signature 'anlz'
cal_units(object)

cal_units(object) <- value

## S4 replacement method for signature 'anlz'
cal_units(object) <- value

cal.units(object)

## S4 method for signature 'anlz'
cal.units(object)

cal.units(object) <- value

## S4 replacement method for signature 'anlz'
cal.units(object) <- value
```

### Arguments

<code>object</code>	is an object of class <code>nifti</code> or <code>anlz</code> .
<code>value</code>	is the value to assign to the <code>cal_units</code> field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
 NIfTI-1  
<http://nifti.nimh.nih.gov/>

coerce-methods

*Force an Object to Belong to the ANALYZE or NIfTI Class***Description**

Methods for function coerce in Package ‘methods’.

**Arguments**

object	is an object of class array or inherits from array.
Class	is the name of the class to which ‘object’ should be coerced; i.e., nifti.
from	is the object to be converted.
value	is the nifti class object to use as a template for various ANALYZE/NIfTI header information.
verbose	is a logical variable (default = FALSE) that allows text-based feedback during execution of the function.

**Value**

An object of class anlz or nifti.

**Methods**

**from = "anlz", to = "nifti"** An object of class anlz is coerced into a NIfTI object.  
**from = "array", to = "anlz"** An object of class array is coerced into an ANALYZE object.  
**from = "array", to = "nifti"** An object of class array is coerced into a NIfTI object.  
**from = "list", to = "anlz"** All objects of class array in the list are coerced into ANALYZE objects. All other objects are left alone. The original list structure is retained.  
**from = "list", to = "nifti"** All objects of class array in the list are coerced into NIfTI objects. All other objects are left alone. The original list structure is retained.

**Author(s)**

Andrew Thornton <zeripath@users.sourceforge.net>,  
 Brandon Whitcher <bwhitcher@gmail.com>

**See Also**

[as](#)

---

compressed-methods      *Extract Image Attribute* compressed

---

## Description

Methods that act on the compressed field in the NIFTI/ANALYZE header.

## Usage

```
compressed(object)

## S4 method for signature 'anzl'
compressed(object)

compressed(object) <- value

## S4 replacement method for signature 'anzl'
compressed(object) <- value
```

## Arguments

object	is an object of class <code>nifti</code> or <code>anzl</code> .
value	is the value to assign to the compressed field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

Convert ANALYZE Codes *Convert ANALYZE Codes*

---

**Description**

Codes that appear in the ANALYZE header are mapped to meaningful character strings.

**Usage**

```
convert.bitpix.anlz(bitpix = NULL)
convert.datatype.anlz(datatype.code = NULL)
convert.orient.anlz(orientation)
```

**Arguments**

<code>bitpix</code>	is the bit-per-pixel code.
<code>datatype.code</code>	defines data type.
<code>orientation</code>	defines the orientation.

**Details**

switch statements are used to map a numeric code to the appropriate string.

**Value**

A character string.

**Author(s)**

Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>

**See Also**

[convert.datatype](#), [convert.bitpix](#), [convert.intent](#), [convert.form](#), [convert.units](#), [convert.slice](#)

**Examples**

```
## 4 = SIGNED_SHORT
convert.datatype.anlz(4)
## 16 = FLOAT
convert.datatype.anlz(16)
## 2 = "sagittal unflipped"
convert.orient.anlz(2)
## 4 = "coronal flipped"
convert.orient.anlz(4)
```

---

Convert NIFTI Codes     *Convert NIFTI Codes*

---

**Description**

Codes that appear in the ANALYZE header are mapped to meaningful character strings.

**Usage**

```
convert.bitpix(bitpix = NULL)

convert.datatype(datatype.code = NULL)

convert.intent(intent.code = NULL)

convert.form(form.code)

convert.units(units, inverse = FALSE)

convert.slice(slice.code)
```

**Arguments**

<code>bitpix</code>	is the bit-per-pixel code.
<code>datatype.code</code>	defines data type.
<code>intent.code</code>	is the NIFTI intent code.
<code>form.code</code>	is the $(x, y, z)$ coordinate system.
<code>units</code>	is the units of <code>pixdim[1..4]</code> .
<code>inverse</code>	is a logical value that denotes the direction of unit conversion.
<code>slice.code</code>	is the slice timing order.

**Details**

`switch` statements are used to map a numeric code to the appropriate string.

**Value**

A character string.

**Author(s)**

Brandon Whitcher <bwhitcher@gmail.com>

**References**

Neuroimaging Informatics Technology Initiative (NIFTI)  
<http://nifti.nimh.nih.gov/>

**Examples**

```
## 4 = SIGNED_SHORT
convert.datatype.anlz(4)
## 16 = FLOAT
convert.datatype.anlz(16)
## 2 = "sagittal unflipped"
convert.orient.anlz(2)
## 4 = "coronal flipped"
convert.orient.anlz(4)
```

---

convert.scene

*Convert AFNI data codes*

---

**Description**

Codes that appear in the AFNI header are mapped to meaningful character strings.

**Usage**

```
convert.scene(scene.data, typestring)
```

**Arguments**

scene.data	defines data type.
typestring	defines whether func or anat data.

**Details**

switch statements are used to map a numeric code to the appropriate string.

**Value**

A character string.

**Author(s)**

Karsten Tabelow <karsten.tabelow@wias-berlin.de>

**References**

AFNI

<http://afni.nimh.nih.gov/pub/dist/src/README.attributes>

**See Also**

[convert.datatype.anlz](#), [convert.orient.anlz](#)

**Examples**

```
## 4 = CT for anatomic data
convert.scene(4, "3DIM_HEAD_ANAT")
```

---

datatype-methods      *Extract Image Attribute* datatype

---

**Description**

Methods that act on the datatype field in the NIFTI/ANALYZE header.

**Usage**

```
datatype(object)

## S4 method for signature 'nifti'
datatype(object)

## S4 method for signature 'anlz'
datatype(object)

datatype(object) <- value

## S4 replacement method for signature 'nifti'
datatype(object) <- value

## S4 method for signature 'ANY'
datatype(object)

## S4 replacement method for signature 'anlz'
datatype(object) <- value
```



**Arguments**

object            is an object of class nifti or anlz.  
value            is the value to assign to the datatype field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
Nifti-1  
<http://nifti.nimh.nih.gov/>

---

data\_type-methods            *Extract Image Attribute* data\_type

---

**Description**

Methods that act on the data\_type field in the NIFTI/ANALYZE header.

**Usage**

```
data_type(object)

## S4 method for signature 'nifti'
data_type(object)

## S4 method for signature 'anlz'
data_type(object)

data_type(object) <- value

## S4 replacement method for signature 'nifti'
data_type(object) <- value

## S4 replacement method for signature 'anlz'
data_type(object) <- value

data.type(object)
```

```
## S4 method for signature 'nifti'  
data.type(object)  
  
## S4 method for signature 'anlz'  
data.type(object)  
  
data.type(object) <- value  
  
## S4 replacement method for signature 'nifti'  
data.type(object) <- value  
  
## S4 replacement method for signature 'anlz'  
data.type(object) <- value  
  
## S4 method for signature 'niftiImage'  
data_type(object)  
  
## S4 method for signature 'niftiImage'  
datatype(object)
```

### Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the data_type field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

### Examples

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")  
img = RNifti::readNifti(file)  
data_type(img)  
datatype(img)
```

---

db_name-methods	<i>Extract Image Attribute db_name</i>
-----------------	--

---

## Description

Methods that act on the db\_name field in the NIFTI/ANALYZE header.

## Usage

```
db_name(object)

## S4 method for signature 'nifti'
db_name(object)

## S4 method for signature 'anlz'
db_name(object)

db_name(object) <- value

## S4 replacement method for signature 'nifti'
db_name(object) <- value

## S4 replacement method for signature 'anlz'
db_name(object) <- value

db.name(object)

## S4 method for signature 'nifti'
db.name(object)

## S4 method for signature 'anlz'
db.name(object)

db.name(object) <- value

## S4 replacement method for signature 'nifti'
db.name(object) <- value

## S4 replacement method for signature 'anlz'
db.name(object) <- value
```

## Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the db_name field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

descrip-methods	<i>Extract Image Attribute</i> descrip
-----------------	--

---

**Description**

Methods that act on the descrip field in the NIFTI/ANALYZE header.

**Usage**

```
descrip(object)

## S4 method for signature 'nifti'
descrip(object)

## S4 method for signature 'anlz'
descrip(object)

descrip(object) <- value

## S4 replacement method for signature 'nifti'
descrip(object) <- value

## S4 replacement method for signature 'anlz'
descrip(object) <- value

## S4 method for signature 'niftiImage'
descrip(object)

## S4 replacement method for signature 'niftiImage'
descrip(object) <- value
```

**Arguments**

object is an object of class `nifti` or `anlz`.  
 value is the value to assign to the `descrip` field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschelli2@gmail.com>,  
 Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
 NIFTI-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
## Not run:
url <- "http://nifti.nimh.nih.gov/nifti-1/data/avg152T1_LR_nifti.nii.gz"
urlfile <- file.path(system.file("nifti", package="oro.nifti"),
                    "mniLR.nii.gz")
download.file(url, urlfile, quiet=TRUE)

## End(Not run)
urlfile <- file.path(system.file("nifti", package="oro.nifti"),
                    "mniLR.nii.gz")
mniLR <- readNIFTI(urlfile)
descrip(mniLR)
## Not run:
descrip(mniLR) <- paste(descrip(mniLR), version$version.string, sep="; ")
descrip(mniLR)

## End(Not run)
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
descrip(img)
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
descrip(img) = "a file"
descrip(img)
stopifnot(descrip(img) == "a file")
```

---

dim\_-methods                      *Extract Image Attribute dim\_*

---

### Description

Methods that act on the dim\_ field in the NIFTI/ANALYZE header.

### Usage

```
dim_(object)

## S4 method for signature 'nifti'
dim_(object)

## S4 method for signature 'anlz'
dim_(object)

dim_(object) <- value

## S4 replacement method for signature 'nifti'
dim_(object) <- value

## S4 replacement method for signature 'anlz'
dim_(object) <- value

## S4 method for signature 'ANY'
dim_(object)
```

### Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the dim_ field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

dim\_info-methods      *Extract Image Attribute dim\_info*

---

## Description

Methods that act on the dim\_info field in the NIFTI/ANALYZE header.

## Usage

```
dim_info(object)

## S4 method for signature 'nifti'
dim_info(object)

dim_info(object) <- value

## S4 replacement method for signature 'nifti'
dim_info(object) <- value
```

## Arguments

object            is an object of class nifti or anlz.  
value            is the value to assign to the dim\_info field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

dim\_un0-methods      *Extract Image Attribute dim\_un0*

---

## Description

Methods that act on the dim\_un0 field in the NIFTI/ANALYZE header.

## Usage

```
dim_un0(object)

## S4 method for signature 'anzl'
dim_un0(object)

dim_un0(object) <- value

## S4 replacement method for signature 'anzl'
dim_un0(object) <- value
```

## Arguments

object            is an object of class nifti or anzl.  
value            is the value to assign to the dim\_un0 field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>



---

dropImageDimension      *Drop Image Dimension*

---

### Description

Drops a dimension of an image that has one-dimension and sets respective values to 0 in pixdim or 1 in dim.

### Usage

```
dropImageDimension(img, onlylast = TRUE, warn = TRUE)
```

```
drop_img_dim(img, onlylast = TRUE, warn = TRUE)
```

### Arguments

img	nifti object
onlylast	is a logical variable (default = TRUE). Drop the dimension only if it is the last dimension. For example, if dim is 10x10x1x10 then no dimension is dropped, but if dim is 10x10x10x1 then it will be changed to 10x10x10.
warn	produces a text output if the number of dimensions is under three.

### Value

Object of class nifti

### Examples

```
nim <- nifti(array(rnorm(10^3), dim = rep(10, 3)))
nim2 <- nifti(array(rnorm(10^3), dim = c(10, 10, 1, 10)))
dropImageDimension(nim2)
dropImageDimension(nim2, onlylast = FALSE)
nim3 <- nifti(array(rnorm(10^3), dim = c(10, 10, 10, 1)))
dropImageDimension(nim3)
dropImageDimension(nim3, onlylast = FALSE) # the same as above
nim4 <- nifti(array(rnorm(10^3), dim = c(10, 10, 10, 1, 10)))
dim(nim4[, , 1, ])
dim(nim4[, , 1, , drop=TRUE])
dropImageDimension(nim4)

nim5 <- nifti(array(rnorm(10^4), dim = c(1, 10, 10, 10, 1, 10)))
dropImageDimension(nim5)
dropImageDimension(nim5, onlylast = FALSE)

nim6 <- nifti(array(rnorm(10^3), dim = c(1, 10, 10, 10, 1, 1)))
dropImageDimension(nim6)
## Not run:
```

```

## 27 scans of Colin Holmes (MNI) brain co-registered and averaged
## NIfTI two-file format
URL <- "http://imaging.mrc-cbu.cam.ac.uk/downloads/Colin/colin_1mm.tgz"
urlfile <- file.path(tempdir(), "colin_1mm.tgz")
download.file(URL, dest=urlfile, quiet=TRUE)
untar(urlfile, exdir=tempdir())
colin <- readNIfTI(file.path(tempdir(), "colin_1mm"))
dim(colin)
dim_(colin)
pixdim(colin)
# this will error
writeNIfTI(colin, filename = tempfile())
colin <- dropImageDimension(colin)
writeNIfTI(colin, filename = tempfile())

## End(Not run)

```

---

exp_date-methods	<i>Extract Image Attribute</i> exp_date
------------------	---

---

## Description

Methods that act on the exp\_date field in the NIfTI/ANALYZE header.

## Usage

```

exp_date(object)

## S4 method for signature 'anzl'
exp_date(object)

exp_date(object) <- value

## S4 replacement method for signature 'anzl'
exp_date(object) <- value

```

## Arguments

object	is an object of class nifti or anzl.
value	is the value to assign to the exp_date field.

## Details

See documentation on the ANALYZE and/or NIfTI data standards for more details.

## Author(s)

John Muschelli <muschellij2@gmail.com>  
 Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIfTI-1  
<http://nifti.nimh.nih.gov/>

---

exp\_time-methods      *Extract Image Attribute exp\_time*

---

## Description

Methods that act on the exp\_time field in the NIfTI/ANALYZE header.

## Usage

```
exp_time(object)

## S4 method for signature 'anzl'
exp_time(object)

exp_time(object) <- value

## S4 replacement method for signature 'anzl'
exp_time(object) <- value
```

## Arguments

object            is an object of class nifti or anzl.  
value            is the value to assign to the exp\_time field.

## Details

See documentation on the ANALYZE and/or NIfTI data standards for more details.

## Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIfTI-1  
<http://nifti.nimh.nih.gov/>

---

extender-methods      *Extract Image Attribute* extender

---

## Description

Methods that act on the extender field in the NIFTI/ANALYZE header.

## Usage

```
extender(object)

## S4 method for signature 'nifti'
extender(object)

extender(object) <- value

## S4 replacement method for signature 'nifti'
extender(object) <- value
```

## Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the extender field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

extents-methods      *Extract Image Attribute* extents

---

## Description

Methods that act on the extents field in the NIFTI/ANALYZE header.

## Usage

```
extents(object)

## S4 method for signature 'nifti'
extents(object)

## S4 method for signature 'anlz'
extents(object)

extents(object) <- value

## S4 replacement method for signature 'nifti'
extents(object) <- value

## S4 replacement method for signature 'anlz'
extents(object) <- value
```

## Arguments

object            is an object of class nifti or anlz.  
value            is the value to assign to the extents field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

field\_skip-methods      *Extract Image Attribute field\_skip*

---

### Description

Methods that act on the field\_skip field in the NIFTI/ANALYZE header.

### Usage

```
field_skip(object)

## S4 method for signature 'anzl'
field_skip(object)

field_skip(object) <- value

## S4 replacement method for signature 'anzl'
field_skip(object) <- value

field.skip(object)

## S4 method for signature 'anzl'
field.skip(object)

field.skip(object) <- value

## S4 replacement method for signature 'anzl'
field.skip(object) <- value
```

### Arguments

object            is an object of class nifti or anzl.  
value            is the value to assign to the field\_skip field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

funused1-methods	<i>Extract Image Attribute</i> funused1
------------------	---

---

## Description

Methods that act on the funused1 field in the NIFTI/ANALYZE header.

## Usage

```
funused1(object)

## S4 method for signature 'anzl'
funused1(object)

funused1(object) <- value

## S4 replacement method for signature 'anzl'
funused1(object) <- value
```

## Arguments

object	is an object of class nifti or anzl.
value	is the value to assign to the funused1 field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

funused2-methods      *Extract Image Attribute funused2*

---

## Description

Methods that act on the funused2 field in the NIFTI/ANALYZE header.

## Usage

```
funused2(object)

## S4 method for signature 'anlz'
funused2(object)

funused2(object) <- value

## S4 replacement method for signature 'anlz'
funused2(object) <- value
```

## Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the funused2 field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>



---

funused3-methods      *Extract Image Attribute funused3*

---

## Description

Methods that act on the funused3 field in the NIFTI/ANALYZE header.

## Usage

```
funused3(object)

## S4 method for signature 'anlz'
funused3(object)

funused3(object) <- value

## S4 replacement method for signature 'anlz'
funused3(object) <- value
```

## Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the funused3 field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

generated-methods      *Extract Image Attribute* generated

---

## Description

Methods that act on the generated field in the NIFTI/ANALYZE header.

## Usage

```
generated(object)

## S4 method for signature 'anzl'
generated(object)

generated(object) <- value

## S4 replacement method for signature 'anzl'
generated(object) <- value
```

## Arguments

object            is an object of class nifti or anzl.  
value            is the value to assign to the generated field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

`glmax-methods`*Extract Image Attribute glmax*

---

**Description**

Methods that act on the `glmax` field in the NIFTI/ANALYZE header.

**Usage**

```
glmax(object)

## S4 method for signature 'nifti'
glmax(object)

## S4 method for signature 'anlz'
glmax(object)

glmax(object) <- value

## S4 replacement method for signature 'nifti'
glmax(object) <- value

## S4 replacement method for signature 'anlz'
glmax(object) <- value
```

**Arguments**

<code>object</code>	is an object of class <code>nifti</code> or <code>anlz</code> .
<code>value</code>	is the value to assign to the <code>glmax</code> field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

`glmin-methods`*Extract Image Attribute glmin*

---

## Description

Methods that act on the `glmin` field in the NIFTI/ANALYZE header.

## Usage

```
glmin(object)

## S4 method for signature 'nifti'
glmin(object)

## S4 method for signature 'anlz'
glmin(object)

glmin(object) <- value

## S4 replacement method for signature 'nifti'
glmin(object) <- value

## S4 replacement method for signature 'anlz'
glmin(object) <- value
```

## Arguments

<code>object</code>	is an object of class <code>nifti</code> or <code>anlz</code> .
<code>value</code>	is the value to assign to the <code>glmin</code> field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

hist\_un0-methods      *Extract Image Attribute hist\_un0*

---

## Description

Methods that act on the hist\_un0 field in the NIFTI/ANALYZE header.

## Usage

```
hist_un0(object)

## S4 method for signature 'anzl'
hist_un0(object)

hist_un0(object) <- value

## S4 replacement method for signature 'anzl'
hist_un0(object) <- value
```

## Arguments

object	is an object of class nifti or anzl.
value	is the value to assign to the hist_un0 field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

hkey\_un0-methods      *Extract Image Attribute* hkey\_un0

---

### Description

Methods that act on the hkey\_un0 field in the NIFTI/ANALYZE header.

### Usage

```
hkey_un0(object)

## S4 method for signature 'anzl'
hkey_un0(object)

hkey_un0(object) <- value

## S4 replacement method for signature 'anzl'
hkey_un0(object) <- value

hkey.un0(object)

## S4 method for signature 'anzl'
hkey.un0(object)

hkey.un0(object) <- value

## S4 replacement method for signature 'anzl'
hkey.un0(object) <- value
```

### Arguments

object	is an object of class nifti or anzl.
value	is the value to assign to the hkey_un0 field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIfTI-1  
<http://nifti.nimh.nih.gov/>

---

hotmetal

*Hot Metal Color Table*

---

## Description

The hotmetal color table patterned after the one used in Matlab.

## Usage

```
hotmetal(n = 64)
```

## Arguments

`n` is the number of color levels (default = 64).

## Details

Based on the `tim.colors` function in the **fields** package. The `hotmetal` function has been modified to break any dependence on code in the **fields** package. Spline interpolation (`interpSpline`) is used when the number of requested colors is not the default.

## Value

A vector of character strings giving the colors in hexadecimal format.

## See Also

[terrain.colors](#), [tim.colors](#), [topo.colors](#)

## Examples

```
hotmetal(10)  
image(outer(1:20,1:20,"+"), col=hotmetal(75), main="hotmetal")
```

---

image-methods

*Methods for Function 'image'*

---

## Description

Produce “lightbox” layout of images for nifti, anlz and afni objects.

## Usage

```
## S4 method for signature 'nifti'
image(
  x,
  z = 1,
  w = 1,
  col = gray(0:64/64),
  plane = c("axial", "coronal", "sagittal"),
  plot.type = c("multiple", "single"),
  zlim = NULL,
  xlab = "",
  ylab = "",
  axes = FALSE,
  oma = rep(0, 4),
  mar = rep(0, 4),
  bg = "black",
  ...
)

## S4 method for signature 'anlz'
image(
  x,
  z = 1,
  w = 1,
  col = gray(0:64/64),
  plane = c("axial", "coronal", "sagittal"),
  plot.type = c("multiple", "single"),
  zlim = NULL,
  xlab = "",
  ylab = "",
  axes = FALSE,
  oma = rep(0, 4),
  mar = rep(0, 4),
  bg = "black",
  ...
)

## S4 method for signature 'afni'
image(x, ...)
```



**Arguments**

<code>x</code>	is an object of class <code>nifti</code> or similar.
<code>z</code>	is the slice to be displayed (ignored when <code>plot.type = "multiple"</code> ).
<code>w</code>	is the time point to be displayed (4D arrays only).
<code>col</code>	is grayscale (by default).
<code>plane</code>	is the plane of acquisition to be displayed (choices are 'axial', 'coronal', 'sagittal').
<code>plot.type</code>	allows the choice between all slices being displayed, in a matrix (left-to-right, top-to-bottom), or a single slice.
<code>zlim</code>	is set to NULL by default and utilizes the internal image range.
<code>xlab</code>	is set to "" since all margins are set to zero.
<code>ylab</code>	is set to "" since all margins are set to zero.
<code>axes</code>	is set to FALSE since all margins are set to zero.
<code>oma</code>	is the size of the outer margins in the <code>par</code> function.
<code>mar</code>	is the number of lines of margin in the <code>par</code> function.
<code>bg</code>	is the background color in the <code>par</code> function.
<code>...</code>	other arguments to the <code>image</code> function may be provided here.

**Details**

Uses the S3 generic function `image`, with medical-image friendly settings, to display `nifti`, `anlz` and `afni` class objects in a "lightbox" layout.

**Methods**

- `x = "ANY"` Generic function: see [image](#).
- `x = "nifti"` Produce images for `x`.
- `x = "anlz"` Produce images for `x`.
- `x = "afni"` Produce images for `x`.

**Author(s)**

Brandon Whitcher <[bwhitcher@gmail.com](mailto:bwhitcher@gmail.com)>

**See Also**

[orthographic-methods](#), [overlay-methods](#)

---

img\_data-methods      *Extract Image Attribute* .Data

---

### Description

Methods that act on the .Data field in the NIFTI/ANALYZE header.

### Usage

```
img_data(object)

## S4 method for signature 'nifti'
img_data(object)

## S4 method for signature 'anlz'
img_data(object)

## S4 method for signature 'character'
img_data(object)

## S4 method for signature 'ANY'
img_data(object)

img_data(object) <- value

## S4 replacement method for signature 'nifti'
img_data(object) <- value

## S4 replacement method for signature 'anlz'
img_data(object) <- value
```

### Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the .Data field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
 NIfTI-1  
<http://nifti.nimh.nih.gov/>

---

img_length	<i>Gets Image Length in Each Dimension</i>
------------	--

---

**Description**

Multiplies the number of slices by the voxel resolution for each direction.

**Usage**

```
img_length(img, units = c("mm", "cm"))
```

**Arguments**

img	Image object, any method with voxdim and dim_
units	output unit, either cubic mm or cubic cm.

**Value**

Scalar numeric, one number, in mm or cm.

**Examples**

```
nim <- nifti(array(rnorm(10^3), dim = c(5, 2, 100)),
pixdim = c(1, 0.5, 0.2, 1))
img_length(nim)
```

---

integerTranslation	<i>integerTranslation</i>
--------------------	---------------------------

---

**Description**

...

**Usage**

```
integerTranslation(nim, data, verbose = FALSE)

invertIntegerTranslation(nim, verbose = FALSE)
```

**Arguments**

nim	is an object of class nifti.
data	is ...
verbose	is a logical variable (default = FALSE) that allows text-based feedback during execution of the function.

**Details**

...

**Value**

...

**Author(s)**

Andrew Thornton &lt;zeripath@users.sourceforge.net&gt;

---

 intent\_code-methods    *Extract Image Attribute* intent\_code
 

---

**Description**

Methods that act on the intent\_code field in the NIFTI/ANALYZE header.

**Usage**

```

intent_code(object)

## S4 method for signature 'nifti'
intent_code(object)

intent_code(object) <- value

## S4 replacement method for signature 'nifti'
intent_code(object) <- value

intent.code(object)

## S4 method for signature 'nifti'
intent.code(object)

intent.code(object) <- value

## S4 replacement method for signature 'nifti'
intent.code(object) <- value

```

```
## S4 method for signature 'niftiImage'  
intent_code(object)  
  
## S4 replacement method for signature 'niftiImage'  
intent_code(object) <- value
```

### Arguments

object            is an object of class nifti or anlz.  
value            is the value to assign to the intent\_code field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

### Examples

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")  
img = RNifti::readNifti(file)  
intent_code(img)  
intent_code(img) = 4  
stopifnot(intent_code(img) == 4)
```

---

intent\_name-methods    *Extract Image Attribute* intent\_name

---

### Description

Methods that act on the intent\_name field in the NIFTI/ANALYZE header.

**Usage**

```
intent_name(object)

## S4 method for signature 'nifti'
intent_name(object)

intent_name(object) <- value

## S4 replacement method for signature 'nifti'
intent_name(object) <- value

intent.name(object)

## S4 method for signature 'nifti'
intent.name(object)

intent.name(object) <- value

## S4 replacement method for signature 'nifti'
intent.name(object) <- value

## S4 method for signature 'niftiImage'
intent_name(object)

## S4 replacement method for signature 'niftiImage'
intent_name(object) <- value
```

**Arguments**

object	is an object of class nifti or anlz.
value	is the value to assign to the intent_name field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
intent_name(img)
intent_name(img) = "hey"
stopifnot(intent_name(img) == "hey")
```

---

intent_p1-methods	<i>Extract Image Attribute</i>	intent_p1
-------------------	--------------------------------	-----------

---

**Description**

Methods that act on the intent\_p1 field in the NIFTI/ANALYZE header.

**Usage**

```
intent_p1(object)

## S4 method for signature 'nifti'
intent_p1(object)

intent_p1(object) <- value

## S4 replacement method for signature 'nifti'
intent_p1(object) <- value

intent.p1(object)

## S4 method for signature 'nifti'
intent.p1(object)

intent.p1(object) <- value

## S4 replacement method for signature 'nifti'
intent.p1(object) <- value

## S4 method for signature 'niftiImage'
intent_p1(object)

## S4 replacement method for signature 'niftiImage'
intent_p1(object) <- value
```

**Arguments**

object	is an object of class nifti or anlz.
value	is the value to assign to the intent_p1 field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

## Examples

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
intent_p1(img)
intent_p1(img) = 2
stopifnot(intent_p1(img) == 2)
intent_p2(img)
intent_p2(img) = 2
stopifnot(intent_p2(img) == 2)
intent_p3(img)
intent_p3(img) = 2
stopifnot(intent_p3(img) == 2)
```

---

intent\_p2-methods      *Extract Image Attribute intent\_p2*

---

## Description

Methods that act on the intent\_p2 field in the NIFTI/ANALYZE header.

## Usage

```
intent_p2(object)

## S4 method for signature 'nifti'
intent_p2(object)

intent_p2(object) <- value

## S4 replacement method for signature 'nifti'
intent_p2(object) <- value
```



```
intent.p2(object)

## S4 method for signature 'nifti'
intent.p2(object)

intent.p2(object) <- value

## S4 replacement method for signature 'nifti'
intent.p2(object) <- value

## S4 method for signature 'niftiImage'
intent.p2(object)

## S4 replacement method for signature 'niftiImage'
intent.p2(object) <- value
```

### Arguments

object            is an object of class nifti or anlz.  
value            is the value to assign to the intent\_p2 field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

intent\_p3-methods            *Extract Image Attribute* intent\_p3

---

### Description

Methods that act on the intent\_p3 field in the NIFTI/ANALYZE header.

**Usage**

```
intent_p3(object)

## S4 method for signature 'nifti'
intent_p3(object)

intent_p3(object) <- value

## S4 replacement method for signature 'nifti'
intent_p3(object) <- value

intent.p3(object)

## S4 method for signature 'nifti'
intent.p3(object)

intent.p3(object) <- value

## S4 replacement method for signature 'nifti'
intent.p3(object) <- value

## S4 method for signature 'niftiImage'
intent_p3(object)

## S4 replacement method for signature 'niftiImage'
intent_p3(object) <- value
```

**Arguments**

object	is an object of class nifti or anlz.
value	is the value to assign to the intent_p3 field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

`internalImage-class`      *"internalImage" class*

---

### **Description**

"internalImage" class

---

`is.afni`                      *check object*

---

### **Description**

Check whether object is of class `afni`.

### **Usage**

`is.afni(x)`

### **Arguments**

`x`                      is an object to be checked.

### **Value**

Logical indicating whether object is of class `afni`.

### **Author(s)**

Karsten Tabelow <karsten.tabelow@wias-berlin.de>

### **References**

AFNI  
<http://afni.nimh.nih.gov/pub/dist/src/README.attributes>

### **See Also**

`afni`

---

is.anlz                      *check object*

---

**Description**

Check whether object is of class [anlz](#).

**Usage**

```
is.anlz(x)
```

**Arguments**

x                      is an object to be checked.

**Value**

Logical indicating whether object is of class [anlz](#).

**Author(s)**

Karsten Tabelow <karsten.tabelow@wias-berlin.de>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>

**See Also**

[anlz](#)

---

is.nifti                      *check object*

---

**Description**

Check whether object is of class [nifti](#).

**Usage**

```
is.nifti(x)
```

```
is.niftiExtension(x)
```

**Arguments**

x is an object to be checked.

**Value**

Logical indicating whether object is of class `nifti`.

**Author(s)**

Karsten Tabelow <karsten.tabelow@wias-berlin.de>

**References**

NIfTI-1  
<http://nifti.nimh.nih.gov/>

**See Also**

`nifti`

---

magic-methods

*Extract Image Attribute* magic

---

**Description**

Methods that act on the magic field in the NIFTI/ANALYZE header.

**Usage**

```
magic(object)

## S4 method for signature 'nifti'
magic(object)

magic(object) <- value

## S4 replacement method for signature 'nifti'
magic(object) <- value

## S4 method for signature 'niftiImage'
magic(object)

## S4 replacement method for signature 'niftiImage'
magic(object) <- value
```

**Arguments**

object            is an object of class nifti or anlz.  
value            is the value to assign to the magic field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
Nifti-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
magic(img)
magic(img) = "ni1"
stopifnot(magic(img) == "ni1")
magic(img) = "n+1"
stopifnot(magic(img) == "n+1")
magic(img) = "r" # bad magic
stopifnot(magic(img) == "")
```

---

nifti

*Constructor for Nifti*

---

**Description**

Constructor for NIFTI class objects.

**Usage**

```
nifti(  
  img = array(0, dim = rep(1, 4)),  
  dim,  
  datatype = 2,  
  cal.min = NULL,  
  cal.max = NULL,  
  pixdim = NULL,
```

```
    ...  
  )
```

### Arguments

<code>img</code>	is a multidimensional array of data.
<code>dim</code>	is the dimension of the data (default = missing).
<code>datatype</code>	is an integer that denotes the type of data contained in each voxel. See <code>convert.datatype</code> or the NIFTI documentation for more details.
<code>cal.min</code>	allows user-specified minimum value in the array (visualization purposes only).
<code>cal.max</code>	allows user-specified maximum value in the array (visualization purposes only).
<code>pixdim</code>	allows user-specified pixel dimension vector (length = 8).
<code>...</code>	allows for additional 'slots' to be specified.

### Value

An object of class `nifti`.

### Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

### References

NIfTI-1  
<http://nifti.nimh.nih.gov/>

### See Also

[nifti](#), [anlz](#), [convert.datatype](#)

### Examples

```
options("niftiAuditTrail"=FALSE)  
  
nim <- nifti() # default  
nim  
nim <- nifti(datatype=4) # 2-byte integers  
nim
```

---

nifti-class

*Class "nifti"*


---

**Description**

The NIFTI class for medical imaging data.

**Usage**

```
## S4 method for signature 'nifti'
show(object)
```

**Arguments**

object            An object of class nifti.

**Objects from the Class**

Objects can be created by calls of the form `new("nifti", data, dim, dimnames, ...)` or by calling the `nifti` function.

**Slots**

.Data: Object of class "array" contains the imaging data  
sizeof\_hdr: Object of class "numeric" contains the size of the header (= 348)  
data\_type: Object of class "character"  
db\_name: Object of class "character"  
extents: Object of class "numeric"  
session\_error: Object of class "numeric"  
regular: Object of class "character"  
dim\_info: Object of class "numeric" contains MRI slice ordering  
dim\_: Object of class "vector" contains the dimensions of the imaging data  
intent\_p1: Object of class "numeric"  
intent\_p2: Object of class "numeric"  
intent\_p3: Object of class "numeric"  
intent\_code: Object of class "numeric"  
datatype: Object of class "numeric"  
bitpix: Object of class "numeric" contains the number of bits per voxel (pixel)  
slice\_start: Object of class "numeric"  
pixdim: Object of class "vector" contains the real-world dimensions of the imaging data  
vox\_offset: Object of class "numeric" contains the voxel offset (= 352 when no extensions exist)  
scl\_slope: Object of class "numeric"



scl\_inter: Object of class "numeric"  
slice\_end: Object of class "numeric"  
slice\_code: Object of class "numeric"  
xyzt\_units: Object of class "numeric"  
cal\_max: Object of class "numeric" contains the maximum display intensity  
cal\_min: Object of class "numeric" contains the minimum display intensity  
slice\_duration: Object of class "numeric"  
toffset: Object of class "numeric"  
glmax: Object of class "numeric"  
glmin: Object of class "numeric"  
descrip: Object of class "character"  
aux\_file: Object of class "character"  
qform\_code: Object of class "numeric"  
sform\_code: Object of class "numeric"  
quatern\_b: Object of class "numeric"  
quatern\_c: Object of class "numeric"  
quatern\_d: Object of class "numeric"  
qoffset\_x: Object of class "numeric"  
qoffset\_y: Object of class "numeric"  
qoffset\_z: Object of class "numeric"  
srow\_x: Object of class "vector"  
srow\_y: Object of class "vector"  
srow\_z: Object of class "vector"  
intent\_name: Object of class "character"  
magic: Object of class "character"  
extender: Object of class "vector"  
reoriented: Object of class "logical"

### Extends

Class "[array](#)", from data part.  
Class "[matrix](#)", by class "array", distance 2, with explicit test and coerce.  
Class "[structure](#)", by class "array", distance 2.  
Class "[vector](#)", by class "array", distance 3, with explicit coerce.  
Class "[vector](#)", by class "array", distance 5, with explicit test and coerce.

### Methods

**image** signature(x = "nifti"): displays the image(s).  
**orthographic** signature(x = "nifti"): displays the image(s).  
**overlay** signature(x = "nifti", y = "nifti"): displays the image(s).  
**show** signature(object = "nifti"): prints out a summary of the imaging data.

**Author(s)**

Brandon Whitcher <bwhitcher@gmail.com>,  
Andrew Thornton <zeripath@users.sourceforge.net>

**References**

NIfTI-1  
<http://nifti.nimh.nih.gov/>

**See Also**

[anzl](#), [niftiExtension](#), [niftiAuditTrail](#)

**Examples**

```
showClass("nifti")
```

---

nifti-operators

*Operations for NIfTI Objects*

---

**Description**

Overloaded operators for nifti objects

**Usage**

```
## S4 method for signature 'nifti,nifti'  
Ops(e1, e2)  
  
## S4 method for signature 'nifti,numeric'  
Ops(e1, e2)  
  
## S4 method for signature 'numeric,nifti'  
Ops(e1, e2)
```

**Arguments**

e1            is an object of class nifti.  
e2            is an object of class nifti.

**Author(s)**

John Muschelli <muschellij2@gmail.com>

## Examples

```
img01 <- nifti(array(1:64, c(4,4,4,1)), datatype=4)
img02 <- nifti(array(64:1, c(4,4,4,1)), datatype=4)
is.nifti(img01 + img02)
is.nifti(sqrt(2) * img01)
is.nifti(img02 / pi)
```

---

niftiAuditTrail-class *Class "niftiAuditTrail"*

---

## Description

An extension of the NIFTI class that adds an audit trail in XML format.

## Objects from the Class

Objects can be created by calls of the form `new("niftiAuditTrail", data, dim, dimnames, ...)`.

## Methods

`show` signature(object = "niftiAuditTrail"): prints out a summary of the imaging data.

## Author(s)

Andrew Thornton <zeripath@users.sourceforge.net>

## References

NIFTI-1  
<http://nifti.nimh.nih.gov/>

## See Also

[nifti](#), [niftiExtension](#)

## Examples

```
showClass("niftiAuditTrail")
```

---

niftiExtension-class    *Class "niftiExtension"*

---

**Description**

An extension of the NIFTI class that allows “extensions” that conform to the NIFTI data standard.

**Objects from the Class**

Objects can be created by calls of the form `new("niftiExtension", data, dim, dimnames, ...)`.

**Author(s)**

Andrew Thornton <zeripath@users.sourceforge.net>

**References**

NIFTI-1  
<http://nifti.nimh.nih.gov/>

**See Also**

[nifti](#), [niftiAuditTrail](#)

**Examples**

```
showClass("niftiExtension")
```

---

niftiExtensionSection-class  
                          *Class "niftiExtensionSection"*

---

**Description**

A `niftiExtensionSection` contains the fields that conform to the NIFTI standard regarding header extensions. A `niftiExtension` is composed of one or more of these objects.

**Objects from the Class**

Objects can be created by calls of the form `new("niftiExtensionSection", data, dim, dimnames, ...)`.

**Author(s)**

Brandon Whitcer <bwhitcher@gmail.com>,  
Andrew Thornton <zeripath@users.sourceforge.net>

**References**

NIfTI-1  
<http://nifti.nimh.nih.gov/>

**See Also**

[niftiExtension](#), [nifti](#)

**Examples**

```
showClass("niftiExtensionSection")
```

---

niftiImage-class	<i>"niftiImage" class</i>
------------------	---------------------------

---

**Description**

"niftiImage" class

---

nifti_assign-methods	<i>Methods for Function [<code>&lt;-</code> in Package 'base']</i>
----------------------	--

---

**Description**

Methods for function [`<-` in Package 'base']

**Methods**

**x = "nifti", i = "ANY", j = "ANY", value = "ANY"** Replaces the data at the provided co-ordinates with the value provided and updates the header.

**x = "nifti", i = "numeric", j = "numeric", value = "ANY"** Replaces the data at the provided co-ordinates with the value provided and updates the header.

**x = "nifti", i = "ANY", j = "missing", value = "ANY"** Replaces the data row i of the provided nifti object with the value provided and updates the header.

**x = "nifti", i = "numeric", j = "missing", value = "ANY"** Replaces the data row i of the provided nifti object with the value provided and updates the header.

**x = "nifti", i = "missing", j = "missing", value = "array"** Replaces the data of the provided nifti object with the array provided and updates the header.

---

nii2oro	<i>Convert RNifti niftiImage to oro.nifti nifti object</i>
---------	--

---

**Description**

Converts a niftiImage from RNifti to a nifti object from the oro.nifti package

**Usage**

```
nii2oro(image)
```

**Arguments**

image	niftiImage object
-------	-------------------

**Value**

Object of class `nifti`

---

nsli	<i>Dimension Accessor Functions</i>
------	-------------------------------------

---

**Description**

Functions to extract the higher dimensions from ANALYZE/NIFTI data.

**Usage**

```
nsli(x)
```

```
NSLI(x)
```

```
ntim(x)
```

```
NTIM(x)
```

**Arguments**

x	is a three- or four-dimensional array (e.g., read in from an ANALYZE/NiftI file).
---	---

**Details**

Simple calls to `dim` to replicate the functionality of `nrow` and `ncol` for higher dimensions of an array that are commonly required when manipulating medical imaging data.

**Value**

Third (slice) or fourth (time) dimension of the array.

**Author(s)**

Brandon Whitcher <bwhitcher@gmail.com>

**See Also**

[readNIFTI](#), [readANALYZE](#)

---

omax-methods

*Extract Image Attribute* omax

---

**Description**

Methods that act on the omax field in the NIFTI/ANALYZE header.

**Usage**

```
omax(object)

## S4 method for signature 'anlz'
omax(object)

omax(object) <- value

## S4 replacement method for signature 'anlz'
omax(object) <- value
```

**Arguments**

object            is an object of class nifti or anlz.  
value            is the value to assign to the omax field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

omin-methods	<i>Extract Image Attribute</i> omin
--------------	-------------------------------------

---

## Description

Methods that act on the omin field in the NIFTI/ANALYZE header.

## Usage

```
omin(object)

## S4 method for signature 'anzl'
omin(object)

omin(object) <- value

## S4 replacement method for signature 'anzl'
omin(object) <- value
```

## Arguments

object	is an object of class nifti or anzl.
value	is the value to assign to the omin field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>



---

onefile	<i>Creates the onefile Specification for NIfTI</i>
---------	--

---

**Description**

Changes the magic and vox\_offset slots to be consistent with the onefile option in [writeNIfTI](#). As of version 0.4.0, oro.nifti did not support the "ni1" magic type for output.

**Usage**

```
onefile(img)
```

**Arguments**

img is a nifti-class object.

**Value**

Object of class nifti.

**Author(s)**

John Muschelli <muschellij2@gmail.com>

**References**

NIfTI-1  
<http://nifti.nimh.nih.gov/>

---

orient-methods	<i>Extract Image Attribute orient</i>
----------------	---------------------------------------

---

**Description**

Methods that act on the orient field in the NIfTI/ANALYZE header.

**Usage**

```
orient(object)

## S4 method for signature 'anzl'
orient(object)

orient(object) <- value

## S4 replacement method for signature 'anzl'
orient(object) <- value
```

**Arguments**

object is an object of class `nifti` or `anlz`.  
 value is the value to assign to the `orient` field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
 Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
 NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

orientation-methods     *Extract NIFTI 3D Image Orientation*

---

**Description**

Methods that act on the “qform” and “sform” information in the NIFTI header.

**Usage**

```
sform(object)

## S4 method for signature 'nifti'
sform(object)

qform(object)

## S4 method for signature 'nifti'
qform(object)
```

**Arguments**

object is an object of class `nifti`.

**Methods**

**object = "nifti"** Extract or replace NIFTI description.

**Author(s)**

Brandon Whitcher <bwhitcher@gmail.com>

**Examples**

```
## Not run:
url <- "http://nifti.nimh.nih.gov/nifti-1/data/avg152T1_LR_nifti.nii.gz"
urlfile <- file.path(system.file("nifti", package="oro.nifti"),
                    "mniLR.nii.gz")
download.file(url, urlfile, quiet=TRUE)

## End(Not run)
urlfile <- file.path(system.file("nifti", package="oro.nifti"),
                    "mniLR.nii.gz")
mniLR <- readNIFTI(urlfile)
sform(mniLR)
```

---

origin-methods

*Extract Image Attribute origin*

---

**Description**

Methods that act on the origin field in the NIFTI/ANALYZE header.

**Usage**

```
origin(object)

## S4 method for signature 'nifti'
origin(object)

## S4 method for signature 'anlz'
origin(object)

## S4 method for signature 'ANY'
origin(object)

origin(object) <- value

## S4 replacement method for signature 'anlz'
origin(object) <- value

## S4 replacement method for signature 'nifti'
origin(object) <- value
```

**Arguments**

object is an object of class `nifti` or `anlz`.  
 value is the value to assign to the `origin` field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschelli2@gmail.com>,  
 Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
 NIFTI-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
fname = system.file("nifti", "mniRL.nii.gz",
package = "oro.nifti")
img = readNIFTI(fname)
oimg = origin(img)
stopifnot(all(oimg == c(-90, -126, -72)))
zero_img = img
origin(zero_img) = rep(0, 3)
stopifnot(all(origin(zero_img) == 0))
```

---

oro2nii

---

*Convert oro.nifti nifti to RNifti niftiImage object*


---

**Description**

Converts a `nifti` from `oro.nifti` to a `niftiImage` object from the `RNifti` package

**Usage**

```
oro2nii(image, verbose = FALSE)
```

**Arguments**

image `nifti` object  
 verbose print messages, passed to `writeNIFTI`

**Value**

Object of class `niftiImage`

---

orthographic-methods *Methods for Function 'orthographic' in Package 'dcmriS4'*

---

**Description**

Produce orthographic display for `nifti`, `anlz` and `afni` objects.

**Usage**

```
orthographic.nifti(  
  x,  
  y = NULL,  
  xyz = NULL,  
  w = 1,  
  col = gray(0:64/64),  
  col.y = hotmetal(),  
  zlim = NULL,  
  zlim.y = NULL,  
  crosshairs = TRUE,  
  col.crosshairs = "red",  
  xlab = "",  
  ylab = "",  
  axes = FALSE,  
  oma = rep(0, 4),  
  mar = rep(0, 4),  
  bg = "black",  
  text = NULL,  
  text.color = "white",  
  text.cex = 2,  
  ...  
)  
  
orthographic(x, ...)  
  
## S4 method for signature 'nifti'  
orthographic(  
  x,  
  y = NULL,  
  xyz = NULL,  
  w = 1,  
  col = gray(0:64/64),  
  col.y = hotmetal(),  
  zlim = NULL,
```

```

    zlim.y = NULL,
    crosshairs = TRUE,
    col.crosshairs = "red",
    xlab = "",
    ylab = "",
    axes = FALSE,
    oma = rep(0, 4),
    mar = rep(0, 4),
    bg = "black",
    text = NULL,
    text.color = "white",
    text.cex = 2,
    ...
)

## S4 method for signature 'anlz'
orthographic(
  x,
  y = NULL,
  xyz = NULL,
  w = 1,
  col = gray(0:64/64),
  col.y = hotmetal(),
  zlim = NULL,
  zlim.y = NULL,
  crosshairs = TRUE,
  col.crosshairs = "red",
  xlab = "",
  ylab = "",
  axes = FALSE,
  oma = rep(0, 4),
  mar = rep(0, 4),
  bg = "black",
  text = NULL,
  text.color = "white",
  text.cex = 2,
  ...
)

## S4 method for signature 'array'
orthographic(x, ...)

## S4 method for signature 'afni'
orthographic(x, ...)

```

### Arguments

`x` is an object of class `nifti` or similar.

<code>y</code>	is an object of class <code>nifti</code> or similar for the overlay.
<code>xyz</code>	is the coordinate for the center of the crosshairs.
<code>w</code>	is the time point to be displayed (4D arrays only).
<code>col</code>	is grayscale (by default).
<code>col.y</code>	is hotmetal (by default).
<code>zlim</code>	is the minimum and maximum 'z' values passed into <code>image</code> .
<code>zlim.y</code>	is the minimum and maximum 'z' values passed into <code>image</code> for the overlay.
<code>crosshairs</code>	is a logical value for the presence of crosshairs in all three orthogonal planes (default = TRUE).
<code>col.crosshairs</code>	is the color of the crosshairs (default = red).
<code>xlab</code>	is set to "" since all margins are set to zero.
<code>ylab</code>	is set to "" since all margins are set to zero.
<code>axes</code>	is set to FALSE since all margins are set to zero.
<code>oma</code>	is the size of the outer margins in the <code>par</code> function.
<code>mar</code>	is the number of lines of margin in the <code>par</code> function.
<code>bg</code>	is the background color in the <code>par</code> function.
<code>text</code>	allows the user to specify text to appear in the fourth (unused) pane.
<code>text.color</code>	is the color of the user-specified text (default = "white").
<code>text.cex</code>	is the size of the user-specified text (default = 2).
<code>...</code>	other arguments to the <code>image</code> function may be provided here.

## Methods

- `x = "afni"` Produce orthographic display for `x`.
- `x = "anlz"` Produce orthographic display for `x`.
- `x = "array"` Produce orthographic display for `x`.
- `x = "nifti"` Produce orthographic display for `x`.

## Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

## See Also

[image-methods](#), [overlay-methods](#)

**Examples**

```

fname = system.file(
  file.path("nifti", "mniRL.nii.gz"),
  package = "oro.nifti")
eve = readNIfTI(fname)
orthographic(eve)

image(eve, z = 45)
image(eve, z = 45, plot.type = "single")
image(eve, z = c(45, 50), plot.type = "single")

```

---

 overlay-methods

*Methods for Function overlay*


---

**Description**

Methods for function overlay

**Usage**

```

overlay.nifti(
  x,
  y,
  z = 1,
  w = 1,
  col.x = gray(0:64/64),
  col.y = hotmetal(),
  zlim.x = NULL,
  zlim.y = NULL,
  plane = c("axial", "coronal", "sagittal"),
  plot.type = c("multiple", "single"),
  xlab = "",
  ylab = "",
  axes = FALSE,
  oma = rep(0, 4),
  mar = rep(0, 4),
  bg = "black",
  NA.x = FALSE,
  NA.y = FALSE,
  ...
)

overlay(x, y, ...)

## S4 method for signature 'nifti,missing'
overlay(
  x,

```



```
    y,  
    z = 1,  
    w = 1,  
    col.x = gray(0:64/64),  
    col.y = hotmetal(),  
    zlim.x = NULL,  
    zlim.y = NULL,  
    plane = c("axial", "coronal", "sagittal"),  
    plot.type = c("multiple", "single"),  
    xlab = "",  
    ylab = "",  
    axes = FALSE,  
    oma = rep(0, 4),  
    mar = rep(0, 4),  
    bg = "black",  
    NA.x = FALSE,  
    NA.y = FALSE,  
    ...  
  )
```

```
## S4 method for signature 'nifti,nifti'
```

```
overlay(  
  x,  
  y,  
  z = 1,  
  w = 1,  
  col.x = gray(0:64/64),  
  col.y = hotmetal(),  
  zlim.x = NULL,  
  zlim.y = NULL,  
  plane = c("axial", "coronal", "sagittal"),  
  plot.type = c("multiple", "single"),  
  xlab = "",  
  ylab = "",  
  axes = FALSE,  
  oma = rep(0, 4),  
  mar = rep(0, 4),  
  bg = "black",  
  NA.x = FALSE,  
  NA.y = FALSE,  
  ...  
)
```

```
## S4 method for signature 'anlz,anlz'
```

```
overlay(  
  x,  
  y,  
  z = 1,
```

```
w = 1,
col.x = gray(0:64/64),
col.y = hotmetal(),
zlim.x = NULL,
zlim.y = NULL,
plane = c("axial", "coronal", "sagittal"),
plot.type = c("multiple", "single"),
xlab = "",
ylab = "",
axes = FALSE,
oma = rep(0, 4),
mar = rep(0, 4),
bg = "black",
NA.x = FALSE,
NA.y = FALSE,
...
)

## S4 method for signature 'anlz,nifti'
overlay(
  x,
  y,
  z = 1,
  w = 1,
  col.x = gray(0:64/64),
  col.y = hotmetal(),
  zlim.x = NULL,
  zlim.y = NULL,
  plane = c("axial", "coronal", "sagittal"),
  plot.type = c("multiple", "single"),
  xlab = "",
  ylab = "",
  axes = FALSE,
  oma = rep(0, 4),
  mar = rep(0, 4),
  bg = "black",
  NA.x = FALSE,
  NA.y = FALSE,
  ...
)

## S4 method for signature 'nifti,anlz'
overlay(
  x,
  y,
  z = 1,
  w = 1,
  col.x = gray(0:64/64),
```

```

    col.y = hotmetal(),
    zlim.x = NULL,
    zlim.y = NULL,
    plane = c("axial", "coronal", "sagittal"),
    plot.type = c("multiple", "single"),
    xlab = "",
    ylab = "",
    axes = FALSE,
    oma = rep(0, 4),
    mar = rep(0, 4),
    bg = "black",
    NA.x = FALSE,
    NA.y = FALSE,
    ...
)

## S4 method for signature 'array,array'
overlay(x, y, ...)

## S4 method for signature 'array,nifti'
overlay(x, y, ...)

## S4 method for signature 'nifti,array'
overlay(x, y, ...)

## S4 method for signature 'array,anlz'
overlay(x, y, ...)

## S4 method for signature 'anlz,array'
overlay(x, y, ...)

## S4 method for signature 'afni,afni'
overlay(x, y, ...)

## S4 method for signature 'afni,array'
overlay(x, y, ...)

```

### Arguments

<code>x, y</code>	is an object of class <code>nifti</code> or similar.
<code>z</code>	is the slice to be displayed (ignored when <code>plot.type = "multiple"</code> ).
<code>w</code>	is the time point to be displayed (4D arrays only).
<code>col.x</code>	is grayscale (by default).
<code>col.y</code>	is hotmetal (by default).
<code>zlim.x, zlim.y</code>	are set to <code>NULL</code> (by default) and taken from the header information.
<code>plane</code>	is the plane of acquisition to be displayed (choices are 'axial', 'coronal', 'sagittal').

plot.type	allows the choice between all slices being displayed, in a matrix (left-to-right, top-to-bottom), or a single slice.
xlab	is set to "" since all margins are set to zero.
ylab	is set to "" since all margins are set to zero.
axes	is set to FALSE since all margins are set to zero.
oma	is the size of the outer margins in the par function.
mar	is the number of lines of margin in the par function.
bg	is the background color in the par function.
NA.x	Set any values of 0 in x to NA
NA.y	Set any values of 0 in y to NA
...	other arguments to the image function may be provided here.

### Details

The `image` command is used multiple times to simultaneously visualize one of the three orthogonal planes in two multidimensional arrays, one on top of the other, for medical imaging data.

### Methods

`x = "nifti", y = "nifti"` Produce overlay of y on x.  
`x = "anlz", y = "anlz"` Produce overlay of y on x.  
`x = "afni", y = "afni"` Produce overlay of y on x.

### Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

### See Also

[image-methods](#), [overlay-methods](#)

---

patient\_id-methods      *Extract Image Attribute patient\_id*

---

### Description

Methods that act on the `patient_id` field in the NIFTI/ANALYZE header.

**Usage**

```
patient_id(object)

## S4 method for signature 'anzl'
patient_id(object)

patient_id(object) <- value

## S4 replacement method for signature 'anzl'
patient_id(object) <- value

patient.id(object)

## S4 method for signature 'anzl'
patient.id(object)

patient.id(object) <- value

## S4 replacement method for signature 'anzl'
patient.id(object) <- value
```

**Arguments**

object	is an object of class nifti or anzl.
value	is the value to assign to the patient_id field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

performPermutation      *Transform array with orthogonal permutation matrix*

---

### Description

Given an orthogonal permutation matrix  $T$ , an array of dimensions and a one-dimensional representation of data. It will return a transformed array with the transformed dimensions.

### Usage

```
performPermutation(T, real.dimensions, data, verbose = FALSE)
```

### Arguments

`T` is an orthogonal matrix.

`real.dimensions` is a one-dimensional array, representing the length of dimensions in data.

`data` is a one-dimensional representation of the data to be transformed.

`verbose` is a logical variable (default = FALSE) that allows text-based feedback during execution of the function.

### Details

This function is mainly used by the [reorient](#) function to transform nifti data into neuroradiological convention.

### Author(s)

Andrew Thornton <zeripath@users.sourceforge.net>

### See Also

[reorient](#), [inverseReorient](#)

---

pixdim-methods      *Extract Image Attribute pixdim*

---

### Description

Methods that act on the `pixdim` field in the NIFTI/ANALYZE header.

**Usage**

```
pixdim(object)

## S4 method for signature 'nifti'
pixdim(object)

## S4 method for signature 'ANY'
pixdim(object)

## S4 method for signature 'anlz'
pixdim(object)

pixdim(object) <- value

## S4 replacement method for signature 'nifti'
pixdim(object) <- value

## S4 replacement method for signature 'anlz'
pixdim(object) <- value

## S4 replacement method for signature 'ANY'
pixdim(object) <- value
```

**Arguments**

object	is an object of class nifti or anlz.
value	is the value to assign to the pixdim field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
## Not run:
url <- "http://nifti.nimh.nih.gov/nifti-1/data/avg152T1_LR_nifti.nii.gz"
urlfile <- file.path(system.file("nifti", package="oro.nifti"),
```

```

"mniLR.nii.gz")
download.file(url, urlfile, quiet=TRUE)

## End(Not run)
urlfile <- file.path(system.file("nifti", package="oro.nifti"),
                     "mniLR.nii.gz")
mniLR <- readNIFTI(urlfile)
pixdim(mniLR)

```

---

qform\_code-methods      *Extract Image Attribute* qform\_code

---

### Description

Methods that act on the qform\_code field in the NIFTI/ANALYZE header.

### Usage

```

qform_code(object)

## S4 method for signature 'nifti'
qform_code(object)

qform_code(object) <- value

## S4 replacement method for signature 'nifti'
qform_code(object) <- value

qform.code(object)

## S4 method for signature 'nifti'
qform.code(object)

qform.code(object) <- value

## S4 replacement method for signature 'nifti'
qform.code(object) <- value

## S4 method for signature 'niftiImage'
qform_code(object)

## S4 replacement method for signature 'niftiImage'
qform_code(object) <- value

```

### Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the qform_code field.



## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

## Examples

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
qform_code(img)
qform_code(img) = 8
stopifnot(qform_code(img) == 8)
```

---

qoffset\_x-methods      *Extract Image Attribute* qoffset\_x

---

## Description

Methods that act on the qoffset\_x field in the NIFTI/ANALYZE header.

## Usage

```
qoffset_x(object)

## S4 method for signature 'nifti'
qoffset_x(object)

qoffset_x(object) <- value

## S4 replacement method for signature 'nifti'
qoffset_x(object) <- value

qoffset.x(object)

## S4 method for signature 'nifti'
qoffset.x(object)
```

```
qoffset.x(object) <- value

## S4 replacement method for signature 'nifti'
qoffset.x(object) <- value

## S4 method for signature 'niftiImage'
qoffset_x(object)

## S4 replacement method for signature 'niftiImage'
qoffset_x(object) <- value
```

### Arguments

object            is an object of class nifti or anlz.  
value            is the value to assign to the qoffset\_x field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

### Examples

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
qoffset_x(img)
qoffset_x(img) = 10
stopifnot(qoffset_x(img) == 10)
```

---

qoffset\_y-methods

*Extract Image Attribute* qoffset\_y

---

### Description

Methods that act on the qoffset\_y field in the NIFTI/ANALYZE header.

**Usage**

```
qoffset_y(object)

## S4 method for signature 'nifti'
qoffset_y(object)

qoffset_y(object) <- value

## S4 replacement method for signature 'nifti'
qoffset_y(object) <- value

qoffset.y(object)

## S4 method for signature 'nifti'
qoffset.y(object)

qoffset.y(object) <- value

## S4 replacement method for signature 'nifti'
qoffset.y(object) <- value

## S4 method for signature 'niftiImage'
qoffset_y(object)

## S4 replacement method for signature 'niftiImage'
qoffset_y(object) <- value
```

**Arguments**

object	is an object of class nifti or anlz.
value	is the value to assign to the qoffset_y field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
qoffset_y(img)
qoffset_y(img) = 10
stopifnot(qoffset_y(img) == 10)
```

---

qoffset\_z-methods      *Extract Image Attribute qoffset\_z*

---

**Description**

Methods that act on the qoffset\_z field in the NIFTI/ANALYZE header.

**Usage**

```
qoffset_z(object)

## S4 method for signature 'nifti'
qoffset_z(object)

qoffset_z(object) <- value

## S4 replacement method for signature 'nifti'
qoffset_z(object) <- value

qoffset.z(object)

## S4 method for signature 'nifti'
qoffset.z(object)

qoffset.z(object) <- value

## S4 replacement method for signature 'nifti'
qoffset.z(object) <- value

## S4 method for signature 'niftiImage'
qoffset_z(object)

## S4 replacement method for signature 'niftiImage'
qoffset_z(object) <- value
```

**Arguments**

object            is an object of class nifti or anlz.  
value            is the value to assign to the qoffset\_z field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
qoffset_z(img)
qoffset_z(img) = 10
stopifnot(qoffset_z(img) == 10)
```

---

quaternion2rotation     *Convert Quaternion into a Rotation Matrix*

---

**Description**

The affine/rotation matrix  $R$  is calculated from the quaternion parameters.

**Usage**

```
quaternion2rotation(b, c, d, tol = 1e-07)
```

```
quaternion2mat44(nim, tol = 1e-07)
```

**Arguments**

<code>b</code>	is the quaternion $b$ parameter.
<code>c</code>	is the quaternion $c$ parameter.
<code>d</code>	is the quaternion $d$ parameter.
<code>tol</code>	is a very small value used to judge if a number is essentially zero.
<code>nim</code>	is an object of class <code>nifti</code> .

**Details**

The quaternion representation is chosen for its compactness in representing rotations. The orientation of the  $(x, y, z)$  axes relative to the  $(i, j, k)$  axes in 3D space is specified using a unit quaternion  $[a, b, c, d]$ , where  $a^2 + b^2 + c^2 + d^2 = 1$ . The  $(b, c, d)$  values are all that is needed, since we require that  $a = [1 - (b^2 + c^2 + d^2)]^{1/2}$  be non-negative. The  $(b, c, d)$  values are stored in the (quatern\_b, quatern\_c, quatern\_d) fields.

**Value**

The (proper)  $3 \times 3$  rotation matrix or  $4 \times 4$  affine matrix.

**Author(s)**

Brandon Whitcher <bwhitcher@gmail.com>

**References**

NIfTI-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
## This R matrix is represented by quaternion [a,b,c,d] = [0,1,0,0]
## (which encodes a 180 degree rotation about the x-axis).
(R <- quaternion2rotation(1, 0, 0))
```

---

quatern\_b-methods      *Extract Image Attribute quatern\_b*

---

**Description**

Methods that act on the quatern\_b field in the NIFTI/ANALYZE header.

**Usage**

```
quatern_b(object)

## S4 method for signature 'nifti'
quatern_b(object)

quatern_b(object) <- value

## S4 replacement method for signature 'nifti'
quatern_b(object) <- value

quatern.b(object)
```

```
## S4 method for signature 'nifti'
quatern.b(object)

quatern.b(object) <- value

## S4 replacement method for signature 'nifti'
quatern.b(object) <- value

## S4 method for signature 'niftiImage'
quatern_b(object)

## S4 replacement method for signature 'niftiImage'
quatern_b(object) <- value
```

### Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the quatern_b field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

### Examples

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
quatern_b(img)
quatern_b(img) = 3
stopifnot(quatern_b(img) == 3)
quatern_c(img)
quatern_c(img) = 3
stopifnot(quatern_c(img) == 3)
quatern_d(img)
quatern_d(img) = 3
stopifnot(quatern_d(img) == 3)
```

---

quatern\_c-methods      *Extract Image Attribute quatern\_c*

---

### Description

Methods that act on the quatern\_c field in the NIFTI/ANALYZE header.

### Usage

```
quatern_c(object)

## S4 method for signature 'nifti'
quatern_c(object)

quatern_c(object) <- value

## S4 replacement method for signature 'nifti'
quatern_c(object) <- value

quatern.c(object)

## S4 method for signature 'nifti'
quatern.c(object)

quatern.c(object) <- value

## S4 replacement method for signature 'nifti'
quatern.c(object) <- value

## S4 method for signature 'niftiImage'
quatern_c(object)

## S4 replacement method for signature 'niftiImage'
quatern_c(object) <- value
```

### Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the quatern_c field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>



## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
 NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

quatern\_d-methods      *Extract Image Attribute quatern\_d*

---

## Description

Methods that act on the quatern\_d field in the NIFTI/ANALYZE header.

## Usage

```
quatern_d(object)

## S4 method for signature 'nifti'
quatern_d(object)

quatern_d(object) <- value

## S4 replacement method for signature 'nifti'
quatern_d(object) <- value

quatern.d(object)

## S4 method for signature 'nifti'
quatern.d(object)

quatern.d(object) <- value

## S4 replacement method for signature 'nifti'
quatern.d(object) <- value

## S4 method for signature 'niftiImage'
quatern_d(object)

## S4 replacement method for signature 'niftiImage'
quatern_d(object) <- value
```

## Arguments

object            is an object of class nifti or anlz.  
 value            is the value to assign to the quatern\_d field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

readAFNI

*readAFNI*

---

**Description**

These functions read in the header information and multidimensional array from a binary file in AFNI format into a `afni`-class object.

**Usage**

```
readAFNI(fname, vol = NULL, verbose = FALSE, warn = -1, call = NULL)
```

**Arguments**

<code>fname</code>	is the file name of the AFNI file.
<code>vol</code>	vector of brick numbers to be read from file.
<code>verbose</code>	is a logical variable (default = FALSE) that allows text-based feedback during execution of the function.
<code>warn</code>	is a number to regulate the display of warnings (default = -1). See options for more details.
<code>call</code>	keeps track of the current function call for use in the AFNI extension.

**Details**

The `readAFNI` function utilizes internal methods `readBin` and `readLines` to efficiently extract information from the header and binary file(s). Compression is allowed on the BRIK file using `gzip`.

Current acceptable data types include

**list("INT16")** DT SIGNED SHORT (16 bits per voxel)

**list("FLOAT32")** DT FLOAT (32 bits per voxel)

**list("COMPLEX128")** DT COMPLEX (128 bits per voxel)

**Value**

object of class `afni`

**Author(s)**

Karsten Tabelow <karsten.tabelow@wias-berlin.de>

**References**

AFNI  
<http://afni.nimh.nih.gov/pub/dist/src/README.attributes>

**See Also**

`readANALYZE`, `readNIFTI`

**Examples**

```
## Not run:
## Taken from the AFNI Matlab Library
## http://afni.nimh.nih.gov/pub/dist/data/afni_matlab_data.tgz
afni.path <- system.file("afni", package="oro.nifti")
orig <- readAFNI(file.path(afni.path, "ARzs_CW_avvr.DEL+orig"))
image(orig, zlim=c(0.5,256), oma=rep(2,4))
orthographic(orig, zlim=c(0.5,256), oma=rep(2,4))
## Taken from the AFNI installation
TT <- readAFNI(file.path(afni.path, "TT_N27_EZ_LR+tlrc"))
image(TT, zlim=c(0.5,256), oma=rep(2,4))
orthographic(TT, zlim=c(0.5,256), oma=rep(2,4))

## End(Not run)
```

---

`readANALYZE`

*readANALYZE*

---

**Description**

These functions read in the header information and multi-dimensional array from a binary file in Analyze 7.5 format.

**Usage**

```
readANALYZE(fname, SPM = FALSE, verbose = FALSE, warn = -1)
```

## Arguments

<code>fname</code>	Pathname of the Analyze pair of files <code>.img</code> and <code>.hdr</code> without the suffix.
<code>SPM</code>	is a logical variable (default = FALSE) that forces the voxel data values to be rescaled using the <code>funused1</code> ANALYZE header field. This is an undocumented convention of ANALYZE files processed using the Statistical Parametric Mapping (SPM) software.
<code>verbose</code>	is a logical variable (default = FALSE) that allows text-based feedback during execution of the function.
<code>warn</code>	is a number to regulate the display of warnings (default = -1). See <code>options</code> for more details.

## Details

The internal functions `readBin` and `rawToChar` are utilized in order to efficiently extract information from a binary file. The types of data are limited to 1- and 2-byte integers, 4-byte floats and 8-byte doubles.

## Value

An object of class `anlz` is produced.

## Author(s)

Brandon Whitcher <bwhitcher@gmail.com>,  
Volker Schmid <volkerschmid@users.sourceforge.net>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>

## See Also

[readNIFTI](#)

## Examples

```
## avg152T1
anzl.path <- system.file("anzl", package="oro.nifti")
mni152 <- readANALYZE(file.path(anzl.path, "avg152T1"))
image(mni152, oma=rep(2,4))
orthographic(mni152, oma=rep(2,4))
```

---

readNifTI	<i>readNifTI</i>
-----------	------------------

---

### Description

These functions read in the header information and multidimensional array from a binary file in NIfTI-1 format into a `nifti`-class object.

### Usage

```
readNifTI(
  fname,
  verbose = FALSE,
  warn = -1,
  reorient = TRUE,
  call = NULL,
  read_data = TRUE,
  rescale_data = TRUE
)

nifti_header(fname, verbose = FALSE, warn = -1)
```

### Arguments

<code>fname</code>	is the file name of the NIfTI file(s).
<code>verbose</code>	is a logical variable (default = FALSE) that allows text-based feedback during execution of the function.
<code>warn</code>	is a number to regulate the display of warnings (default = -1). See options for more details.
<code>reorient</code>	is a logical variable (default = TRUE) that enforces Qform/Sform transformations.
<code>call</code>	keeps track of the current function call for use in the NifTI extension.
<code>read_data</code>	Should the data be read in? If this is FALSE, then an array of NAs are given instead of the true data. Useful if you are simply interested in the header.
<code>rescale_data</code>	Should the data be rescaled using the slope and intercept values? If so, slope and intercept will be reset

### Details

The `readNifTI` function utilizes internal methods `readBin` and `readChar` to efficiently extract information from the binary file(s).

Current acceptable data types include

- list("UINT8")** BINARY (1 bit per voxel)
- list("INT16")** SIGNED SHORT (16 bits per voxel)
- list("INT32")** SIGNED INT (32 bits per voxel)

```
list("FLOAT32") FLOAT (32 bits per voxel)
list("DOUBLE64") DOUBLE (64 bits per voxel)
list("UINT16") UNSIGNED SHORT (16 bits per voxel)
list("UINT32") UNSIGNED INT (32 bits per voxel)
```

### Value

An object of class `nifti`.

### Author(s)

Brandon Whitcher <bwhitcher@gmail.com>,  
Volker Schmid <volkerschmid@users.sourceforge.net>,  
Andrew Thornton <zeripath@users.sourceforge.net>

### References

NIfTI-1  
<http://nifti.nimh.nih.gov/>

### See Also

[readAFNI](#), [readANALYZE](#)

### Examples

```
## Not run:
url <- "http://nifti.nimh.nih.gov/nifti-1/data/filtered_func_data.nii.gz"
urlfile <- file.path(system.file("nifti", package="oro.nifti"),
                    "filtered_func_data")
download.file(url, urlfile, quiet=TRUE)

## End(Not run)
## The NIfTI file provided here contains the first 18 volumes (10%)
## of the original data set
urlfile <- file.path(system.file("nifti", package="oro.nifti"),
                    "filtered_func_data")
(ffd <- readNIfTI(urlfile))
image(ffd, oma=rep(2,4))
orthographic(ffd, oma=rep(2,4))
## Not run:
## 27 scans of Colin Holmes (MNI) brain co-registered and averaged
## NIfTI two-file format
URL <- "http://imaging.mrc-cbu.cam.ac.uk/downloads/Colin/colin_1mm.tgz"
urlfile <- file.path(tempdir(), "colin_1mm.tgz")
download.file(URL, dest=urlfile, quiet=TRUE)
untar(urlfile, exdir=tempdir())
colin <- readNIfTI(file.path(tempdir(), "colin_1mm"))
image(colin, oma=rep(2,4))
orthographic(colin, oma=rep(2,4))
```

```
## End(Not run)
```

---

regular-methods	<i>Extract Image Attribute</i>	regular
-----------------	--------------------------------	---------

---

## Description

Methods that act on the regular field in the NIFTI/ANALYZE header.

## Usage

```
regular(object)

## S4 method for signature 'nifti'
regular(object)

## S4 method for signature 'anlz'
regular(object)

regular(object) <- value

## S4 replacement method for signature 'nifti'
regular(object) <- value

## S4 replacement method for signature 'anlz'
regular(object) <- value
```

## Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the regular field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

reorient	<i>Reorient Image using NIfTI header</i>
----------	--

---

### Description

Transforms in the NIfTI header are parsed and normalized versions of these transforms are applied.

### Usage

```
reorient(nim, data, verbose = FALSE, invert = FALSE, tol = 1e-07)
```

```
inverseReorient(nim, verbose = FALSE)
```

### Arguments

nim	is an object of class <code>nifti</code> .
data	is an array associated with <code>nim</code> .
verbose	is a logical variable (default = <code>FALSE</code> ) that allows text-based feedback during execution of the function.
invert	stores the inverse transform.
tol	is a very small value used to judge if a number is essentially zero.

### Details

This function utilizes the `performPermutation` function internally.

### Author(s)

Andrew Thornton <zeripath@users.sourceforge.net>,  
Brandon Whitcher <bwhitcher@gmail.com>

### See Also

[performPermutation](#)



---

resetSlopeIntercept     *Change Intercept to 0 and Slope to 1 in NIfTI Object*

---

**Description**

Forces image scl\_slope to 1 and scl\_inter to be 0 of slots of class nifti. This is so that when images are rendered/written, the values correspond to those in the array (stored in the .Data slot) and are not scaled.

**Usage**

```
resetSlopeIntercept(img)
```

```
zero_trans(img)
```

**Arguments**

img                    is a nifti object (or character of filename). If an anlz object is passed, the unaltered anlz object is returned.

**Value**

An object of the same type passed.

**Author(s)**

John Muschelli <muschellij2@gmail.com>

---

rmniigz                    *Remove File Extensions Around the NIfTI/ANALYZE Formats*

---

**Description**

Simple function(s) that remove file extensions commonly found when using NIfTI-1 or ANALYZE format files.

**Usage**

```
rmniigz(x)
```

```
rmnii(x)
```

```
rmgz(x)
```

```
rmhdrgz(x)
```

```
rmhdr(x)
rmimggz(x)
rmimg(x)
```

**Arguments**

x is the file name.

**Value**

The file name without offending suffix.

**Author(s)**

Brandon Whitcher <bwhitcher@gmail.com>

---

scannum-methods      *Extract Image Attribute* scannum

---

**Description**

Methods that act on the scannum field in the NIFTI/ANALYZE header.

**Usage**

```
scannum(object)

## S4 method for signature 'anzl'
scannum(object)

scannum(object) <- value

## S4 replacement method for signature 'anzl'
scannum(object) <- value
```

**Arguments**

object is an object of class nifti or anzl.  
value is the value to assign to the scannum field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIfTI-1  
<http://nifti.nimh.nih.gov/>

---

scl\_inter-methods      *Extract Image Attribute* scl\_inter

---

**Description**

Methods that act on the scl\_inter field in the NIFTI/ANALYZE header.

**Usage**

```
scl_inter(object)

## S4 method for signature 'nifti'
scl_inter(object)

scl_inter(object) <- value

## S4 replacement method for signature 'nifti'
scl_inter(object) <- value

scl.inter(object)

## S4 method for signature 'nifti'
scl.inter(object)

scl.inter(object) <- value

## S4 replacement method for signature 'nifti'
scl.inter(object) <- value

## S4 method for signature 'niftiImage'
scl_inter(object)

## S4 method for signature 'niftiImage'
scl.inter(object)
```

**Arguments**

object is an object of class nifti or anlz.  
 value is the value to assign to the scl\_inter field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
 Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
 NIFTI-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
scl_inter(img)
scl.inter(img)
```

---

scl\_slope-methods

*Extract Image Attribute scl\_slope*


---

**Description**

Methods that act on the scl\_slope field in the NIFTI/ANALYZE header.

**Usage**

```
scl_slope(object)

## S4 method for signature 'nifti'
scl_slope(object)

scl_slope(object) <- value

## S4 replacement method for signature 'nifti'
scl_slope(object) <- value

scl.slope(object)
```

```
## S4 method for signature 'nifti'  
scl_slope(object)  
  
scl_slope(object) <- value  
  
## S4 replacement method for signature 'nifti'  
scl_slope(object) <- value  
  
## S4 method for signature 'niftiImage'  
scl_slope(object)  
  
## S4 method for signature 'niftiImage'  
scl_slope(object)
```

### Arguments

object            is an object of class nifti or anlz.  
value            is the value to assign to the scl\_slope field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

### Examples

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")  
img = RNifti::readNifti(file)  
scl_slope(img)  
scl_slope(img)
```

---

session\_error-methods *Extract Image Attribute session\_error*

---

### Description

Methods that act on the session\_error field in the NIFTI/ANALYZE header.

### Usage

```
session_error(object)

## S4 method for signature 'nifti'
session_error(object)

## S4 method for signature 'anlz'
session_error(object)

session_error(object) <- value

## S4 replacement method for signature 'nifti'
session_error(object) <- value

## S4 replacement method for signature 'anlz'
session_error(object) <- value

session.error(object)

## S4 method for signature 'nifti'
session.error(object)

## S4 method for signature 'anlz'
session.error(object)

session.error(object) <- value

## S4 replacement method for signature 'nifti'
session.error(object) <- value

## S4 replacement method for signature 'anlz'
session.error(object) <- value
```

### Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the session_error field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

sform\_code-methods      *Extract Image Attribute* sform\_code

---

**Description**

Methods that act on the sform\_code field in the NIFTI/ANALYZE header.

**Usage**

```
sform_code(object)

## S4 method for signature 'nifti'
sform_code(object)

sform_code(object) <- value

## S4 replacement method for signature 'nifti'
sform_code(object) <- value

sform.code(object)

## S4 method for signature 'nifti'
sform.code(object)

sform.code(object) <- value

## S4 replacement method for signature 'nifti'
sform.code(object) <- value

## S4 method for signature 'niftiImage'
sform_code(object)
```

```
## S4 replacement method for signature 'niftiImage'  
sform_code(object) <- value
```

### Arguments

object is an object of class nifti or anlz.  
value is the value to assign to the sform\_code field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nih.gov/>

### Examples

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")  
img = RNifti::readNifti(file)  
sform_code(img)  
sform_code(img) = 4  
stopifnot(sform_code(img) == 4)
```

---

sizeof\_hdr-methods      *Extract Image Attribute sizeof\_hdr*

---

### Description

Methods that act on the sizeof\_hdr field in the NIFTI/ANALYZE header.

### Usage

```
sizeof_hdr(object)  
  
## S4 method for signature 'nifti'  
sizeof_hdr(object)  
  
## S4 method for signature 'anlz'  
sizeof_hdr(object)
```



```
sizeof_hdr(object)

## S4 method for signature 'nifti'
sizeof_hdr(object)

## S4 method for signature 'anlz'
sizeof_hdr(object)

sizeof_hdr(object) <- value

## S4 replacement method for signature 'nifti'
sizeof_hdr(object) <- value

## S4 replacement method for signature 'anlz'
sizeof_hdr(object) <- value
```

### Arguments

object            is an object of class nifti or anlz.  
value            is the value to assign to the data\_type field.

### Details

See documentation on the ANALYZE and/or NIfTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIfTI-1  
<http://nifti.nimh.nih.gov/>

### Description

Produce “lightbox” layout of slices for nifti, anlz and afni objects.

**Usage**

```
slice(x, ...)  
  
## S4 method for signature 'nifti'  
slice(  
  x,  
  z = 1,  
  w = 1,  
  col = gray(0:64/64),  
  plane = c("axial", "coronal", "sagittal"),  
  zlim = NULL,  
  xlab = "",  
  ylab = "",  
  axes = FALSE,  
  oma = rep(0, 4),  
  mar = rep(0, 4),  
  bg = "black",  
  useRaster = TRUE,  
  ...  
)  
  
## S4 method for signature 'anlz'  
slice(  
  x,  
  z = 1,  
  w = 1,  
  col = gray(0:64/64),  
  plane = c("axial", "coronal", "sagittal"),  
  zlim = NULL,  
  xlab = "",  
  ylab = "",  
  axes = FALSE,  
  oma = rep(0, 4),  
  mar = rep(0, 4),  
  bg = "black",  
  useRaster = TRUE,  
  ...  
)  
  
## S4 method for signature 'array'  
slice(x, ...)  
  
## S4 method for signature 'afni'  
slice(x, ...)
```

**Arguments**

x is an object of class `nifti` or similar.

...	other arguments to the image function may be provided here.
z	is the slice to be displayed (ignored when plot.type = "multiple").
w	is the time point to be displayed (4D arrays only).
col	is grayscale (by default).
plane	is the plane of acquisition to be displayed (choices are 'axial', 'coronal', 'sagittal').
zlim	is set to NULL by default and utilizes the internal image range.
xlab	is set to "" since all margins are set to zero.
ylab	is set to "" since all margins are set to zero.
axes	is set to FALSE since all margins are set to zero.
oma	is the size of the outer margins in the par function.
mar	is the number of lines of margin in the par function.
bg	is the background color in the par function.
useRaster	if TRUE, a bitmap raster is used to plot the image instead of polygons. Passed to <a href="#">image</a>

### Details

Uses the S3 generic function `slice`, with medical-image friendly settings, to display `nifti`, `anlz` and `afni` class objects in a "lightbox" layout.

### Methods

`x = "ANY"` Generic function: see [image](#).

`x = "nifti"` Produce images for `x`.

`x = "anlz"` Produce images for `x`.

`x = "afni"` Produce images for `x`.

### Author(s)

Brandon Whitcher <[bwhitcher@gmail.com](mailto:bwhitcher@gmail.com)>

### See Also

[orthographic-methods](#), [image-methods](#)

---

slice\_code-methods      *Extract Image Attribute slice\_code*

---

### Description

Methods that act on the slice\_code field in the NIFTI/ANALYZE header.

### Usage

```
slice_code(object)

## S4 method for signature 'nifti'
slice_code(object)

slice_code(object) <- value

## S4 replacement method for signature 'nifti'
slice_code(object) <- value

slice.code(object)

## S4 method for signature 'nifti'
slice.code(object)

slice.code(object) <- value

## S4 replacement method for signature 'nifti'
slice.code(object) <- value

## S4 method for signature 'niftiImage'
slice_code(object)

## S4 replacement method for signature 'niftiImage'
slice_code(object) <- value
```

### Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the slice_code field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
 NIFTI-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
slice_code(img)
slice_code(img) = 8
stopifnot(slice_code(img) == 8)
```

---

slice\_duration-methods

*Extract Image Attribute slice\_duration*

---

**Description**

Methods that act on the slice\_duration field in the NIFTI/ANALYZE header.

**Usage**

```
slice_duration(object)

## S4 method for signature 'nifti'
slice_duration(object)

slice_duration(object) <- value

## S4 replacement method for signature 'nifti'
slice_duration(object) <- value

slice.duration(object)

## S4 method for signature 'nifti'
slice.duration(object)

slice.duration(object) <- value

## S4 replacement method for signature 'nifti'
slice.duration(object) <- value

## S4 method for signature 'niftiImage'
slice_duration(object)
```

```
## S4 replacement method for signature 'niftiImage'
slice_duration(object) <- value
```

### Arguments

object is an object of class nifti or anlz.  
value is the value to assign to the slice\_duration field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

### Examples

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
slice_duration(img)
slice_duration(img) = 8
stopifnot(slice_duration(img) == 8)
```

---

slice\_end-methods      *Extract Image Attribute slice\_end*

---

### Description

Methods that act on the slice\_end field in the NIFTI/ANALYZE header.

### Usage

```
slice_end(object)

## S4 method for signature 'nifti'
slice_end(object)

slice_end(object) <- value
```

```
## S4 replacement method for signature 'nifti'
slice_end(object) <- value

slice_end(object)

## S4 method for signature 'nifti'
slice_end(object)

slice_end(object) <- value

## S4 replacement method for signature 'nifti'
slice_end(object) <- value

## S4 method for signature 'niftiImage'
slice_end(object)

## S4 replacement method for signature 'niftiImage'
slice_end(object) <- value

## S4 method for signature 'niftiImage'
slice_end(object)

## S4 replacement method for signature 'niftiImage'
slice_end(object) <- value
```

### Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the slice_end field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

### Examples

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
```

```

img = RNifti::readNifti(file)
slice_end(img)
slice_end(img) = 4
stopifnot(slice_end(img) == 4)
slice.end(img)
slice.end(img) = 0

```

---

slice\_overlay-methods *Methods for Function slice\_overlay*

---

## Description

Methods for function slice\_overlay

## Usage

```

slice_overlay.nifti(
  x,
  y,
  z = 1,
  w = 1,
  col.x = gray(0:64/64),
  col.y = hotmetal(),
  zlim.x = NULL,
  zlim.y = NULL,
  plane = c("axial", "coronal", "sagittal"),
  xlab = "",
  ylab = "",
  axes = FALSE,
  oma = rep(0, 4),
  mar = rep(0, 4),
  bg = "black",
  NA.x = FALSE,
  NA.y = TRUE,
  useRaster = TRUE,
  ...
)

slice_overlay(x, y, ...)

## S4 method for signature 'nifti,missing'
slice_overlay(
  x,
  y,
  z = 1,
  w = 1,
  col.x = gray(0:64/64),

```



```
    col.y = hotmetal(),
    zlim.x = NULL,
    zlim.y = NULL,
    plane = c("axial", "coronal", "sagittal"),
    xlab = "",
    ylab = "",
    axes = FALSE,
    oma = rep(0, 4),
    mar = rep(0, 4),
    bg = "black",
    NA.x = FALSE,
    NA.y = TRUE,
    useRaster = TRUE,
    ...
)

## S4 method for signature 'nifti,nifti'
slice_overlay(
  x,
  y,
  z = 1,
  w = 1,
  col.x = gray(0:64/64),
  col.y = hotmetal(),
  zlim.x = NULL,
  zlim.y = NULL,
  plane = c("axial", "coronal", "sagittal"),
  xlab = "",
  ylab = "",
  axes = FALSE,
  oma = rep(0, 4),
  mar = rep(0, 4),
  bg = "black",
  NA.x = FALSE,
  NA.y = TRUE,
  useRaster = TRUE,
  ...
)

## S4 method for signature 'anlz,anlz'
slice_overlay(
  x,
  y,
  z = 1,
  w = 1,
  col.x = gray(0:64/64),
  col.y = hotmetal(),
  zlim.x = NULL,
```

```
zlim.y = NULL,
plane = c("axial", "coronal", "sagittal"),
xlab = "",
ylab = "",
axes = FALSE,
oma = rep(0, 4),
mar = rep(0, 4),
bg = "black",
NA.x = FALSE,
NA.y = TRUE,
useRaster = TRUE,
...
)

## S4 method for signature 'anlz,nifti'
slice_overlay(
  x,
  y,
  z = 1,
  w = 1,
  col.x = gray(0:64/64),
  col.y = hotmetal(),
  zlim.x = NULL,
  zlim.y = NULL,
  plane = c("axial", "coronal", "sagittal"),
  xlab = "",
  ylab = "",
  axes = FALSE,
  oma = rep(0, 4),
  mar = rep(0, 4),
  bg = "black",
  NA.x = FALSE,
  NA.y = TRUE,
  useRaster = TRUE,
  ...
)

## S4 method for signature 'nifti,anlz'
slice_overlay(
  x,
  y,
  z = 1,
  w = 1,
  col.x = gray(0:64/64),
  col.y = hotmetal(),
  zlim.x = NULL,
  zlim.y = NULL,
  plane = c("axial", "coronal", "sagittal"),
```

```

    xlab = "",
    ylab = "",
    axes = FALSE,
    oma = rep(0, 4),
    mar = rep(0, 4),
    bg = "black",
    NA.x = FALSE,
    NA.y = TRUE,
    useRaster = TRUE,
    ...
)

## S4 method for signature 'array,array'
slice_overlay(x, y, ...)

## S4 method for signature 'array,nifti'
slice_overlay(x, y, ...)

## S4 method for signature 'nifti,array'
slice_overlay(x, y, ...)

## S4 method for signature 'array,anlz'
slice_overlay(x, y, ...)

## S4 method for signature 'anlz,array'
slice_overlay(x, y, ...)

## S4 method for signature 'afni,afni'
slice_overlay(x, y, ...)

## S4 method for signature 'afni,array'
slice_overlay(x, y, ...)

```

### Arguments

<code>x, y</code>	is an object of class <code>nifti</code> or similar.
<code>z</code>	is the slice to be displayed (ignored when <code>plot.type = "multiple"</code> ).
<code>w</code>	is the time point to be displayed (4D arrays only).
<code>col.x</code>	is grayscale (by default).
<code>col.y</code>	is hotmetal (by default).
<code>zlim.x, zlim.y</code>	are set to NULL (by default) and taken from the header information.
<code>plane</code>	is the plane of acquisition to be displayed (choices are 'axial', 'coronal', 'sagittal').
<code>xlab</code>	is set to "" since all margins are set to zero.
<code>ylab</code>	is set to "" since all margins are set to zero.
<code>axes</code>	is set to FALSE since all margins are set to zero.

oma	is the size of the outer margins in the par function.
mar	is the number of lines of margin in the par function.
bg	is the background color in the par function.
NA.x	Set any values of 0 in x to NA
NA.y	Set any values of 0 in y to NA
useRaster	if TRUE, a bitmap raster is used to plot the image instead of polygons. Passed to <a href="#">image</a>
...	other arguments to the image function may be provided here.

### Details

The image command is used multiple times to simultaneously visualize one of the three orthogonal planes in two multidimensional arrays, one on top of the other, for medical imaging data.

### Methods

`x = "nifti", y = "nifti"` Produce slice\_overlay of y on x.

`x = "anlz", y = "anlz"` Produce slice\_overlay of y on x.

`x = "afni", y = "afni"` Produce slice\_overlay of y on x.

### Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

### See Also

[image-methods](#), [slice\\_overlay-methods](#)

---

slice\_start-methods    *Extract Image Attribute slice\_start*

---

### Description

Methods that act on the slice\_start field in the NIFTI/ANALYZE header.

### Usage

```
slice_start(object)

## S4 method for signature 'nifti'
slice_start(object)

slice_start(object) <- value

## S4 replacement method for signature 'nifti'
```

```
slice_start(object) <- value

slice.start(object)

## S4 method for signature 'nifti'
slice.start(object)

slice.start(object) <- value

## S4 replacement method for signature 'nifti'
slice.start(object) <- value

## S4 method for signature 'niftiImage'
slice_start(object)

## S4 replacement method for signature 'niftiImage'
slice_start(object) <- value

## S4 method for signature 'niftiImage'
slice.start(object)

## S4 replacement method for signature 'niftiImage'
slice.start(object) <- value
```

### Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the slice_start field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

### Examples

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
```

```
slice_start(img)
slice_start(img) = 4
stopifnot(slice_start(img) == 4)
slice.start(img)
slice.start(img) = 0
```

---

smax-methods

*Extract Image Attribute* smax

---

## Description

Methods that act on the smax field in the NIFTI/ANALYZE header.

## Usage

```
smax(object)

## S4 method for signature 'anzl'
smax(object)

smax(object) <- value

## S4 replacement method for signature 'anzl'
smax(object) <- value
```

## Arguments

object	is an object of class nifti or anzl.
value	is the value to assign to the smax field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

smin-methods                      *Extract Image Attribute* smin

---

## Description

Methods that act on the smin field in the NIFTI/ANALYZE header.

## Usage

```
smin(object)

## S4 method for signature 'anzl'
smin(object)

smin(object) <- value

## S4 replacement method for signature 'anzl'
smin(object) <- value
```

## Arguments

object	is an object of class nifti or anzl.
value	is the value to assign to the smin field.

## Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

## Author(s)

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

srow\_x-methods      *Extract Image Attribute srow\_x*

---

### Description

Methods that act on the srow\_x field in the NIFTI/ANALYZE header.

### Usage

```
srow_x(object)

## S4 method for signature 'nifti'
srow_x(object)

srow_x(object) <- value

## S4 replacement method for signature 'nifti'
srow_x(object) <- value

srow.x(object)

## S4 method for signature 'nifti'
srow.x(object)

srow.x(object) <- value

## S4 replacement method for signature 'nifti'
srow.x(object) <- value

## S4 method for signature 'niftiImage'
srow_x(object)

## S4 replacement method for signature 'niftiImage'
srow_x(object) <- value
```

### Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the srow_x field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>



**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
 NIfTI-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
srow_x(img)
srow_x(img) = rep(0, 4)
stopifnot(srow_x(img) == rep(0, 4))

srow_y(img)
srow_y(img) = rep(0, 4)
stopifnot(srow_y(img) == rep(0, 4))

srow_z(img)
srow_z(img) = rep(0, 4)
stopifnot(srow_z(img) == rep(0, 4))
```

---

srow\_y-methods

*Extract Image Attribute srow\_y*


---

**Description**

Methods that act on the srow\_y field in the NIfTI/ANALYZE header.

**Usage**

```
srow_y(object)

## S4 method for signature 'nifti'
srow_y(object)

srow_y(object) <- value

## S4 replacement method for signature 'nifti'
srow_y(object) <- value

srow.y(object)

## S4 method for signature 'nifti'
srow.y(object)

srow.y(object) <- value
```

```
## S4 replacement method for signature 'nifti'  
srow.y(object) <- value  
  
## S4 method for signature 'niftiImage'  
srow_y(object)  
  
## S4 replacement method for signature 'niftiImage'  
srow_y(object) <- value
```

### Arguments

object            is an object of class nifti or anlz.  
value            is the value to assign to the srow\_y field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

srow_z-methods	<i>Extract Image Attribute srow_z</i>
----------------	---------------------------------------

---

### Description

Methods that act on the srow\_z field in the NIFTI/ANALYZE header.

### Usage

```
srow_z(object)  
  
## S4 method for signature 'nifti'  
srow_z(object)  
  
srow_z(object) <- value  
  
## S4 replacement method for signature 'nifti'
```

```
srow_z(object) <- value  
  
srow.z(object)  
  
## S4 method for signature 'nifti'  
srow.z(object)  
  
srow.z(object) <- value  
  
## S4 replacement method for signature 'nifti'  
srow.z(object) <- value  
  
## S4 method for signature 'niftiImage'  
srow_z(object)  
  
## S4 replacement method for signature 'niftiImage'  
srow_z(object) <- value
```

### Arguments

object            is an object of class nifti or anlz.  
value            is the value to assign to the srow\_z field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

start\_field-methods    *Extract Image Attribute* start\_field

---

### Description

Methods that act on the start\_field field in the NIFTI/ANALYZE header.

**Usage**

```
start_field(object)

## S4 method for signature 'anzl'
start_field(object)

start_field(object) <- value

## S4 replacement method for signature 'anzl'
start_field(object) <- value
```

**Arguments**

object            is an object of class nifti or anzl.  
value            is the value to assign to the start\_field field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

tim.colors

*Tim's Useful Color Table*

---

**Description**

A pleasing rainbow style color table patterned after that used in Matlab.

**Usage**

```
tim.colors(n = 64)
```

**Arguments**

n                    is the number of color levels (default = 64).

**Details**

Based on the `tim.colors` function in the **fields** package. The `tim.colors` function here has been modified to break any dependence on code in the **fields** package. Spline interpolation (`interpSpline`) is used when the number of requested colors is not the default.

**Value**

A vector of character strings giving the colors in hexadecimal format.

**Author(s)**

Tim Hoar (GSP-NCAR); modified by Brandon Whitcher

**See Also**

[hotmetal](#), [topo.colors](#), [terrain.colors](#)

**Examples**

```
tim.colors(10)
image(outer(1:20, 1:20, "+"), col=tim.colors(75), main="tim.colors")
```

---

toffset-methods

*Extract Image Attribute* toffset

---

**Description**

Methods that act on the `toffset` field in the NIFTI/ANALYZE header.

**Usage**

```
toffset(object)

## S4 method for signature 'nifti'
toffset(object)

toffset(object) <- value

## S4 replacement method for signature 'nifti'
toffset(object) <- value

## S4 method for signature 'niftiImage'
toffset(object)

## S4 replacement method for signature 'niftiImage'
toffset(object) <- value
```

**Arguments**

object is an object of class `nifti` or `anlz`.  
value is the value to assign to the `toffset` field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
toffset(img)
toffset(img) = 8
stopifnot(toffset(img) == 8)
```

---

translateCoordinate *Translate Voxel Coordinates*

---

**Description**

Translates a voxel index into the continuous coordinate space defined by the NIFTI `qform` and `sform` information.

**Usage**

```
translateCoordinate(i, nim, verbose = FALSE)
```

**Arguments**

`i` An index vector in `nim`.  
`nim` An object of class `nifti`.  
`verbose` Provide detailed output to the user.

**Details**

This function takes as input a `nifti` object and an index vector in the voxel space of the object and translates that voxel index into the continuous coordinate space defined by the object's `qform` and `sform`.

Please note:

1. By default the index `i` varies most rapidly, etc.
2. The ANALYZE 7.5 coordinate system is

```
+x = Left
+y = Anterior
+z = Superior
```

(A left-handed co-ordinate system).

3. The three methods below give the locations of the voxel centres in the `x,y,z` system. In many cases programs will want to display the data on other grids. In which case the program will be required to convert the desired `(x,y,z)` values in to voxel values using the inverse transformation.
4. Method 2 uses a factor `qfac` which is either `-1` or `1`. `qfac` is stored in `pixdim[0]`. If `pixdim[0] != 1` or `-1`, which should not occur, we assume `1`.
5. The units of the `xyzt` are set in `xyzt_units` field.

**Value**

A `nifti`-class object with translated coordinates.

**Author(s)**

Andrew Thornton <zeripath@users.sourceforge.net>

**Examples**

```
ffd <- readNIFTI(file.path(system.file("nifti", package="oro.nifti"),
                                   "filtered_func_data"))
xyz <- c(1,1,1)
translateCoordinate(xyz, ffd, verbose=TRUE)
xyz <- trunc(dim(ffd)[1:3]/2)
translateCoordinate(xyz, ffd, verbose=TRUE)
```

---

unused1-methods

*Extract Image Attribute unused1*

---

**Description**

Methods that act on the `unused1` field in the NIFTI/ANALYZE header.

**Usage**

```
unused1(object)

## S4 method for signature 'anlz'
unused1(object)

unused1(object) <- value

## S4 replacement method for signature 'anlz'
unused1(object) <- value
```

**Arguments**

object	is an object of class nifti or anlz.
value	is the value to assign to the unused1 field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

verified-methods

*Extract Image Attribute* verified

---

**Description**

Methods that act on the verified field in the NIFTI/ANALYZE header.

**Usage**

```
verified(object)

## S4 method for signature 'anlz'
verified(object)

verified(object) <- value
```



```
## S4 replacement method for signature 'anlz'  
verified(object) <- value
```

### Arguments

object            is an object of class nifti or anlz.  
value            is the value to assign to the verified field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

views-methods            *Extract Image Attribute* views

---

### Description

Methods that act on the views field in the NIFTI/ANALYZE header.

### Usage

```
views(object)  
  
## S4 method for signature 'anlz'  
views(object)  
  
views(object) <- value  
  
## S4 replacement method for signature 'anlz'  
views(object) <- value
```

### Arguments

object            is an object of class nifti or anlz.  
value            is the value to assign to the views field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

vols\_added-methods      *Extract Image Attribute* vols\_added

---

**Description**

Methods that act on the vols\_added field in the NIFTI/ANALYZE header.

**Usage**

```
vols_added(object)

## S4 method for signature 'anzl'
vols_added(object)

vols_added(object) <- value

## S4 replacement method for signature 'anzl'
vols_added(object) <- value

vols.added(object)

## S4 method for signature 'anzl'
vols.added(object)

vols.added(object) <- value

## S4 replacement method for signature 'anzl'
vols.added(object) <- value
```

**Arguments**

object            is an object of class nifti or anzl.  
value            is the value to assign to the vols\_added field.

**Details**

See documentation on the ANALYZE and/or NIFTI data standards for more details.

**Author(s)**

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIfTI-1  
<http://nifti.nimh.nih.gov/>

---

voxdim

*Gets Voxel Dimensions*

---

**Description**

Grabs the pixdim and takes the correct elements

**Usage**

```
voxdim(img)
```

**Arguments**

img                    nifti object

**Value**

Vector of length 3

**Examples**

```
nim <- nifti(array(rnorm(10^3), dim = c(5, 2, 100)),  
pixdim = c(1, 0.5, 0.2, 1))  
voxdim(nim)
```

---

voxres	<i>Gets Voxel Resolution</i>
--------	------------------------------

---

**Description**

Grabs the 3 voxel dimensions and takes the product

**Usage**

```
voxres(img, units = c("mm", "cm"))
```

**Arguments**

img	nifti object
units	output unit, either cubic mm or cubic cm.

**Value**

Scalar numeric, one number, in cubic mm or cubic cm (cc/mL).

---

vox_offset-methods	<i>Extract Image Attribute vox_offset</i>
--------------------	---

---

**Description**

Methods that act on the vox\_offset field in the NIFTI/ANALYZE header.

**Usage**

```
vox_offset(object)

## S4 method for signature 'nifti'
vox_offset(object)

## S4 method for signature 'anlz'
vox_offset(object)

vox_offset(object) <- value

## S4 replacement method for signature 'nifti'
vox_offset(object) <- value

## S4 replacement method for signature 'anlz'
vox_offset(object) <- value
```

```
vox.offset(object)

## S4 method for signature 'nifti'
vox.offset(object)

## S4 method for signature 'anlz'
vox.offset(object)

vox.offset(object) <- value

## S4 replacement method for signature 'nifti'
vox.offset(object) <- value

## S4 replacement method for signature 'anlz'
vox.offset(object) <- value

## S4 method for signature 'niftiImage'
vox_offset(object)

## S4 method for signature 'niftiImage'
vox.offset(object)
```

### Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the vox_offset field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

### References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIFTI-1  
<http://nifti.nimh.nih.gov/>

### Examples

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
vox_offset(img)
img = RNifti::readNifti(file)
vox.offset(img)
```

---

vox\_units-methods      *Extract Image Attribute vox\_units*

---

### Description

Methods that act on the vox\_units field in the NIFTI/ANALYZE header.

### Usage

```
vox_units(object)

## S4 method for signature 'anlz'
vox_units(object)

vox_units(object) <- value

## S4 replacement method for signature 'anlz'
vox_units(object) <- value

vox.units(object)

## S4 method for signature 'anlz'
vox.units(object)

vox.units(object) <- value

## S4 replacement method for signature 'anlz'
vox.units(object) <- value
```

### Arguments

object            is an object of class nifti or anlz.  
value            is the value to assign to the vox\_units field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschelli2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

## References

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
 NIFTI-1  
<http://nifti.nimh.nih.gov/>

---

writeAFNI-methods      *writeAFNI*

---

## Description

This function saves a afni-class object to HEAD/BRIK pair in AFNI format.

## Usage

```
writeAFNI(nim, ...)

## S4 method for signature 'afni'
writeAFNI(nim, fname, verbose = FALSE, warn = -1)
```

## Arguments

nim	is an object of class afni.
...	Additional variables defined by the method.
fname	is the path and file name to save the AFNI file (.HEAD/BRIK) <b>without</b> the suffix.
verbose	is a logical variable (default = FALSE) that allows text-based feedback during execution of the function.
warn	is a number to regulate the display of warnings (default = -1). See <a href="#">options</a> for more details.

## Details

The writeAFNI function utilizes the internal writeBin and writeLines command to write information to header/binary file pair.

Current acceptable data types include

**INT16"** DT SIGNED SHORT (16 bits per voxel)

**FLOAT32"** DT FLOAT (32 bits per voxel)

**"COMPLEX128"** DT COMPLEX (128 bits per voxel)

## Value

Nothing.

**Methods**

**nim = "afni"** Write AFNI volume to disk.

**nim = "ANY"** Not implemented.

**Author(s)**

Karsten Tabelow <karsten.tabelow@wias-berlin.de>

**References**

AFNI

<http://afni.nimh.nih.gov/pub/dist/src/README.attributes>

**See Also**

[writeANALYZE](#), [writeNIFTI](#)

**Examples**

```
## Taken from the AFNI Matlab Library
## http://afni.nimh.nih.gov/pub/dist/data/afni_matlab_data.tgz
afni.path <- system.file("afni", package="oro.nifti")
orig <- readAFNI(file.path(afni.path, "ARzs_CW_avvr.DEL+orig"))
fname = file.path(tempdir(), "test-afni-image")
writeAFNI(orig, fname, verbose=TRUE)

data <- readAFNI(fname, verbose=TRUE)
image(orig, zlim=c(0.5,256), oma=rep(2,4), bg="white")
image(data, zlim=c(0.5,256), oma=rep(2,4), bg="white")
abs.err <- abs(data - orig)
image(as(abs.err, "nifti"), zlim=range(0,1), oma=rep(2,4),
      bg="white")
```

---

writeANALYZE-methods *writeANALYZE*

---

**Description**

This function saves an Analyze-class object to a single binary file in Analyze format.

**Usage**

```
## S4 method for signature 'anlz'
writeANALYZE(
  aim,
  filename,
  gzipped = TRUE,
```



```
    verbose = FALSE,  
    warn = -1,  
    compression = 9  
  )
```

### Arguments

aim	is an object of class anlz.
filename	is the path and file name to save the Analyze file pair (.hdr,img) <b>without</b> the suffixes.
gzipped	is a character string that enables exportation of compressed (.gz) files (default = TRUE).
verbose	is a logical variable (default = FALSE) that allows text-based feedback during execution of the function.
warn	is a number to regulate the display of warnings (default = -1). See <a href="#">options</a> for more details.
compression	The amount of compression to be applied when writing a file when gzipped = TRUE

### Details

The writeANALYZE function utilizes the internal writeBin and writeChar command to write information to a binary file.

### Value

Nothing.

### Methods

**object = "anlz"** Write ANALYZE volume to disk.

### Author(s)

Brandon Whitcher <bwhitcher@gmail.com>

### References

Analyze 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>

### See Also

[writeAFNI](#), [writeNIFTI](#)

**Examples**

```

norm <- dnorm(seq(-5, 5, length=32), sd=2)
norm <- (norm-min(norm)) / max(norm-min(norm))
img <- outer(outer(norm, norm), norm)
img <- round(255*img)
img[17:32,,] <- 255 - img[17:32,,]
img.anlz <- anlz(img) # create Analyze object

fname = file.path(tempdir(), "test-anlz-image-uint8")

writeANALYZE(img.anlz, fname, verbose=TRUE)
## These files should be viewable in, for example, FSLview
## Make sure you adjust the min/max values for proper visualization
data <- readANALYZE(fname, verbose=TRUE)
image(img.anlz, oma=rep(2,4), bg="white")
image(data, oma=rep(2,4), bg="white")
abs.err <- abs(data - img.anlz)
image(as(abs.err, "anlz"), zlim=range(img.anlz), oma=rep(2,4), bg="white")

## Not run:
## Loop through all possible data types
datatypes <- list(code=c(2, 4, 8, 16, 64),
                  name=c("uint8", "int16", "int32", "float", "double"))
equal <- vector("list")
for (i in 1:length(datatypes$code)) {
  fname <- paste("test-anlz-image-", datatypes$name[i], sep="")
  fname = file.path(tempdir(), fname)
  rm(img.anlz)
  img.anlz <- anlz(img, datatype=datatypes$code[i])
  writeANALYZE(img.anlz, fname)
  equal[[i]] <- all(readANALYZE(fname) == img)
}
names(equal) <- datatypes$name
unlist(equal)

## End(Not run)

```

---

writeNIfTI-methods      *writeNIfTI*

---

**Description**

This function saves a NIfTI-class object to a single binary file in NIfTI format.

**Usage**

```

## S4 method for signature 'nifti'
writeNIfTI(

```

```
    nim,
    filename,
    onefile = TRUE,
    gzipped = TRUE,
    verbose = FALSE,
    warn = -1,
    compression = 9
)

## S4 method for signature 'niftiExtension'
writeNIfTI(
  nim,
  filename,
  onefile = TRUE,
  gzipped = TRUE,
  verbose = FALSE,
  warn = -1,
  compression = 9
)

## S4 method for signature 'anlz'
writeNIfTI(
  nim,
  filename,
  onefile = TRUE,
  gzipped = TRUE,
  verbose = FALSE,
  warn = -1,
  compression = 9
)

## S4 method for signature 'array'
writeNIfTI(
  nim,
  filename,
  onefile = TRUE,
  gzipped = TRUE,
  verbose = FALSE,
  warn = -1,
  compression = 9
)
```

### Arguments

nim	is an object of class nifti or anlz.
filename	is the path and file name to save the NIfTI file (.nii) <b>without</b> the suffix.
onefile	is a logical value that allows the scanning of single-file (.nii) or dual-file format (.hdr and .img) NIfTI files (default = TRUE).

<code>gzipped</code>	is a character string that enables exportation of compressed (.gz) files (default = TRUE).
<code>verbose</code>	is a logical variable (default = FALSE) that allows text-based feedback during execution of the function.
<code>warn</code>	is a number to regulate the display of warnings (default = -1). See <a href="#">options</a> for more details.
<code>compression</code>	The amount of compression to be applied when writing a file when <code>gzipped = TRUE</code>

### Details

The `writeNIfTI` function utilizes the internal `writeBin` and `writeChar` command to write information to a binary file.

Current acceptable data types include

**`list("UINT8")`** DT BINARY (1 bit per voxel)  
**`list("INT16")`** DT SIGNED SHORT (16 bits per voxel)  
**`list("INT32")`** DT SIGNED INT (32 bits per voxel)  
**`list("FLOAT32")`** DT FLOAT (32 bits per voxel)  
**`list("DOUBLE64")`** DT DOUBLE (64 bits per voxel)  
**`list("UINT16")`** DT UNSIGNED SHORT (16 bits per voxel)

### Value

Nothing.

### Methods

**`object = "anlz"`** Convert ANALYZE object to class `nifti` and write the NIfTI volume to disk.  
**`object = "array"`** Convert array to class `nifti` and write the NIfTI volume to disk.  
**`object = "nifti"`** Write NIfTI volume to disk.

### Author(s)

Brandon Whitcer <[bwhitcher@gmail.com](mailto:bwhitcher@gmail.com)>,  
 Volker Schmid <[volkerschmid@users.sourceforge.net](mailto:volkerschmid@users.sourceforge.net)>

### References

NIfTI-1  
<http://nifti.nimh.nih.gov/>

### See Also

[writeAFNI](#), [writeANALYZE](#)

**Examples**

```

norm <- dnorm(seq(-5, 5, length=32), sd=2)
norm <- (norm-min(norm)) / max(norm-min(norm))
img <- outer(outer(norm, norm), norm)
img <- round(255 * img)
img[17:32,,] <- 255 - img[17:32,,]
img.nifti <- nifti(img) # create NIFTI object

fname = file.path(tempdir(), "test-nifti-image-uint8")

writeNIFTI(img.nifti, fname, verbose=TRUE)
## These files should be viewable in, for example, FSLview
## Make sure you adjust the min/max values for proper visualization
data <- readNIFTI(fname, verbose=TRUE)
image(img.nifti, oma=rep(2,4), bg="white")
image(data, oma=rep(2,4), bg="white")
abs.err <- abs(data - img.nifti)
image(as(abs.err, "nifti"), zlim=range(img.nifti), oma=rep(2,4),
      bg="white")

## Not run:
## Loop through all possible data types
datatypes <- list(code=c(2, 4, 8, 16, 64),
                  name=c("uint8", "int16", "int32", "float", "double"))
equal <- vector("list")
for (i in 1:length(datatypes$code)) {
  fname <- paste("test-nifti-image-", datatypes$name[i], sep="")
  fname = file.path(tempdir(), fname)
  rm(img.nifti)
  img.nifti <- nifti(img, datatype=datatypes$code[i])
  writeNIFTI(img.nifti, fname, verbose=TRUE)
  equal[[i]] <- all(readNIFTI(fname) == img)
}
names(equal) <- datatypes$name
unlist(equal)

## End(Not run)

```

---

xyzt2space

*Bitwise Conversion Subroutines*


---

**Description**

Units of spatial and temporal dimensions, and MRI-specific spatial and temporal information.

**Usage**

```
xyzt2space(xyzt)
```

```
xyzt2time(xyzt)
```

```
space.time2xyzt(ss, tt)
```

```
dim2freq(di)
```

```
dim2phase(di)
```

```
dim2slice(di)
```

### Arguments

xyzt	represents the units of pixdim[1..4] in the NIfTI header.
ss	is the character string of spatial units. Valid strings are: “Unknown”, “meter”, “mm” and “micron”.
tt	is the character string of temporal units. Valid strings are: “sec”, “msec”, “usec”, “Hz”, “ppm” and “rads”.
di	represents MRI slice ordering in the NIfTI header.

### Details

The functions `xyzt2space` and `xyzt2time` can be used to mask off the undesired bits from the `xyzt_units` fields, leaving “pure” space and time codes.

[http://nifti.nimh.nih.gov/nifti-1/documentation/nifti1fields/nifti1fields\\_pages/xyzt\\_units.html](http://nifti.nimh.nih.gov/nifti-1/documentation/nifti1fields/nifti1fields_pages/xyzt_units.html)

The functions `dim2freq`, `dim2phase`, and `dim2slice` can be used to extract values from the `dim_info` byte.

[http://nifti.nimh.nih.gov/nifti-1/documentation/nifti1fields/nifti1fields\\_pages/dim\\_info.html](http://nifti.nimh.nih.gov/nifti-1/documentation/nifti1fields/nifti1fields_pages/dim_info.html)

### Value

For `diminfo`: the frequency, phase and slice dimensions encode which spatial dimension (1,2, or 3) corresponds to which acquisition dimension for MRI data. For `xyzt_units`: the codes are used to indicate the units of `pixdim`. Dimensions 1,2,3 are for x,y,z; dimension 4 is for time (t).

### Author(s)

B. Whitcher <bwhitcher@gmail.com>

### References

Neuroimaging Informatics Technology Initiative (NIFTI)

<http://nifti.nimh.nih.gov/>

### See Also

[convert.units](#), [convert.slice](#)

---

xyzt\_units-methods      *Extract Image Attribute* xyzt\_units

---

### Description

Methods that act on the xyzt\_units field in the NIFTI/ANALYZE header.

### Usage

```
xyzt_units(object)

## S4 method for signature 'nifti'
xyzt_units(object)

xyzt_units(object) <- value

## S4 replacement method for signature 'nifti'
xyzt_units(object) <- value

xyzt.units(object)

## S4 method for signature 'nifti'
xyzt.units(object)

xyzt.units(object) <- value

## S4 replacement method for signature 'nifti'
xyzt.units(object) <- value

## S4 method for signature 'niftiImage'
xyzt_units(object)

## S4 replacement method for signature 'niftiImage'
xyzt_units(object) <- value
```

### Arguments

object	is an object of class nifti or anlz.
value	is the value to assign to the xyzt_units field.

### Details

See documentation on the ANALYZE and/or NIFTI data standards for more details.

### Author(s)

John Muschelli <muschellij2@gmail.com>,  
Brandon Whitcher <bwhitcher@gmail.com>

**References**

ANALYZE 7.5  
<http://eeg.sourceforge.net/ANALYZE75.pdf>  
NIfTI-1  
<http://nifti.nimh.nih.gov/>

**Examples**

```
file = system.file("extdata", "example.nii.gz", package = "RNifti")
img = RNifti::readNifti(file)
xyz_t_units(img)
xyz_t_units(img) = 8
stopifnot(xyz_t_units(img) == 8)
```



# Index

- \* **Misc**
  - rmniigz, [113](#)
- \* **aplot**
  - hotmetal, [55](#)
  - tim.colors, [140](#)
- \* **classes**
  - afni-class, [5](#)
  - anlz-class, [8](#)
  - nifti-class, [72](#)
  - niftiAuditTrail-class, [75](#)
  - niftiExtension-class, [76](#)
  - niftiExtensionSection-class, [76](#)
- \* **file**
  - readAFNI, [106](#)
  - readANALYZE, [107](#)
  - readNIfTI, [109](#)
  - writeAFNI-methods, [151](#)
  - writeANALYZE-methods, [152](#)
  - writeNIfTI-methods, [154](#)
- \* **internalImage**
  - internalImage-class, [67](#)
- \* **methods**
  - audit.trail-methods, [16](#)
  - blend, [20](#)
  - coerce-methods, [27](#)
  - image-methods, [56](#)
  - nifti\_assign-methods, [77](#)
  - orientation-methods, [82](#)
  - orthographic-methods, [85](#)
  - overlay-methods, [88](#)
  - readAFNI, [106](#)
  - slice-methods, [121](#)
  - slice\_overlay-methods, [128](#)
  - writeAFNI-methods, [151](#)
  - writeANALYZE-methods, [152](#)
  - writeNIfTI-methods, [154](#)
- \* **misc**
  - convert.scene, [31](#)
  - nsli, [78](#)
  - xyzt2space, [157](#)
- \* **niftiImage**
  - niftiImage-class, [77](#)
  - [<- ,ANY,ANY,ANY,ANY-method (nifti\_assign-methods), [77](#)
  - [<- ,nifti,ANY,ANY,ANY-method (nifti\_assign-methods), [77](#)
  - [<- ,nifti,ANY,missing,ANY-method (nifti\_assign-methods), [77](#)
  - [<- ,nifti,missing,missing,array-method (nifti\_assign-methods), [77](#)
  - [<- ,nifti,numeric,missing,ANY-method (nifti\_assign-methods), [77](#)
  - [<- ,nifti,numeric,numeric,ANY-method (nifti\_assign-methods), [77](#)
  - [<--methods (nifti\_assign-methods), [77](#)
- afni, [67](#), [106](#), [107](#)
- afni-class, [5](#)
- anlz, [7](#), [7](#), [8](#), [68](#), [71](#), [74](#)
- anlz-class, [8](#)
- anlz-nifti-ops, [10](#)
- array, [7](#), [10](#), [73](#)
- as, [27](#)
- as,array,anlz-method (coerce-methods), [27](#)
- as,array,nifti-method (coerce-methods), [27](#)
- as.anlz, [11](#)
- as.nifti, [12](#)
- as<- ,array,anlz-method (coerce-methods), [27](#)
- as<- ,array,nifti-method (coerce-methods), [27](#)
- Audit Trails, [12](#)
- audit.trail (audit.trail-methods), [16](#)
- audit.trail,nifti-method (audit.trail-methods), [16](#)
- audit.trail-methods, [16](#)
- audit.trail<- (audit.trail-methods), [16](#)

- audit.trail<- ,nifti-method  
(audit.trail-methods), 16
- aux.file (aux\_file-methods), 17
- aux.file, anlz-method  
(aux\_file-methods), 17
- aux.file, nifti-method  
(aux\_file-methods), 17
- aux.file<- (aux\_file-methods), 17
- aux.file<- , anlz-method  
(aux\_file-methods), 17
- aux.file<- , nifti-method  
(aux\_file-methods), 17
- aux\_file (aux\_file-methods), 17
- aux\_file, anlz-method  
(aux\_file-methods), 17
- aux\_file, nifti-method  
(aux\_file-methods), 17
- aux\_file, niftiImage-method  
(aux\_file-methods), 17
- aux\_file-methods, 17
- aux\_file-methods, (aux\_file-methods), 17
- aux\_file<- (aux\_file-methods), 17
- aux\_file<- , anlz-method  
(aux\_file-methods), 17
- aux\_file<- , nifti-method  
(aux\_file-methods), 17
- aux\_file<- , niftiImage-method  
(aux\_file-methods), 17
  
- bitpix (bitpix-methods), 19
- bitpix, anlz-method (bitpix-methods), 19
- bitpix, nifti-method (bitpix-methods), 19
- bitpix, niftiImage-method  
(bitpix-methods), 19
- bitpix-methods, 19
- bitpix-methods, (bitpix-methods), 19
- bitpix<- (bitpix-methods), 19
- bitpix<- , anlz-method (bitpix-methods),  
19
- bitpix<- , nifti-method (bitpix-methods),  
19
  
- blend, 20
- blend, anlz, anlz-method (blend), 20
- blend, anlz, anlz-methods (blend), 20
- blend, anlz, nifti-method (blend), 20
- blend, anlz, nifti-methods (blend), 20
- blend, nifti, anlz-method (blend), 20
- blend, nifti, anlz-methods (blend), 20
- blend, nifti, nifti-method (blend), 20
  
- blend, nifti, nifti-methods (blend), 20
- blendVolumes (blend), 20
  
- cal.max (cal\_max-methods), 22
- cal.max, anlz-method (cal\_max-methods),  
22
- cal.max, nifti-method (cal\_max-methods),  
22
- cal.max, niftiImage-method  
(cal\_max-methods), 22
- cal.max<- (cal\_max-methods), 22
- cal.max<- , anlz-method  
(cal\_max-methods), 22
- cal.max<- , nifti-method  
(cal\_max-methods), 22
- cal.max<- , niftiImage-method  
(cal\_max-methods), 22
- cal.min (cal\_min-methods), 24
- cal.min, anlz-method (cal\_min-methods),  
24
- cal.min, nifti-method (cal\_min-methods),  
24
- cal.min, niftiImage-method  
(cal\_min-methods), 24
- cal.min<- (cal\_min-methods), 24
- cal.min<- , anlz-method  
(cal\_min-methods), 24
- cal.min<- , nifti-method  
(cal\_min-methods), 24
- cal.min<- , niftiImage-method  
(cal\_min-methods), 24
- cal.units (cal\_units-methods), 26
- cal.units, anlz-method  
(cal\_units-methods), 26
- cal.units, nifti-method  
(cal\_units-methods), 26
- cal.units<- (cal\_units-methods), 26
- cal.units<- , anlz-method  
(cal\_units-methods), 26
- cal\_img (calibrateImage), 21
- cal\_max (cal\_max-methods), 22
- cal\_max, anlz-method (cal\_max-methods),  
22
- cal\_max, nifti-method (cal\_max-methods),  
22
- cal\_max, niftiImage-method  
(cal\_max-methods), 22
- cal\_max-methods, 22
- cal\_max-methods, (cal\_max-methods), 22

- cal\_max<- (cal\_max-methods), 22
- cal\_max<- ,anzl-method
  - (cal\_max-methods), 22
- cal\_max<- ,nifti-method
  - (cal\_max-methods), 22
- cal\_max<- ,niftiImage-method
  - (cal\_max-methods), 22
- cal\_min (cal\_min-methods), 24
- cal\_min,anzl-method (cal\_min-methods), 24
- cal\_min,nifti-method (cal\_min-methods), 24
- cal\_min,niftiImage-method
  - (cal\_min-methods), 24
- cal\_min-methods, 24
- cal\_min-methods, (cal\_min-methods), 24
- cal\_min<- (cal\_min-methods), 24
- cal\_min<- ,anzl-method
  - (cal\_min-methods), 24
- cal\_min<- ,nifti-method
  - (cal\_min-methods), 24
- cal\_min<- ,niftiImage-method
  - (cal\_min-methods), 24
- cal\_units (cal\_units-methods), 26
- cal\_units,anzl-method
  - (cal\_units-methods), 26
- cal\_units-methods, 26
- cal\_units-methods, (cal\_units-methods), 26
- cal\_units<- (cal\_units-methods), 26
- cal\_units<- ,anzl-method
  - (cal\_units-methods), 26
- calibrateImage, 21
- coerce,anzl,nifti-method
  - (coerce-methods), 27
- coerce,array,anzl-method
  - (coerce-methods), 27
- coerce,array,nifti-method
  - (coerce-methods), 27
- coerce,list,anzl-method
  - (coerce-methods), 27
- coerce,list,nifti-method
  - (coerce-methods), 27
- coerce-methods, 27
- coerce<- ,anzl,nifti-method
  - (coerce-methods), 27
- coerce<- ,array,anzl-method
  - (coerce-methods), 27
- coerce<- ,array,nifti-method
  - (coerce-methods), 27
- coerce<- ,list,anzl-method
  - (coerce-methods), 27
- coerce<- ,list,nifti-method
  - (coerce-methods), 27
- compressed (compressed-methods), 28
- compressed,anzl-method
  - (compressed-methods), 28
- compressed-methods, 28
- compressed-methods, (compressed-methods), 28
- compressed<- (compressed-methods), 28
- compressed<- ,anzl-method
  - (compressed-methods), 28
- Convert ANALYZE Codes, 29
- Convert NIFTI Codes, 30
- convert.bitpix, 29
- convert.bitpix (Convert NIFTI Codes), 30
- convert.bitpix.anzl (Convert ANALYZE Codes), 29
- convert.datatype, 29, 71
- convert.datatype (Convert NIFTI Codes), 30
- convert.datatype.anzl, 8, 32
- convert.datatype.anzl (Convert ANALYZE Codes), 29
- convert.form, 29
- convert.form (Convert NIFTI Codes), 30
- convert.intent, 29
- convert.intent (Convert NIFTI Codes), 30
- convert.orient.anzl, 32
- convert.orient.anzl (Convert ANALYZE Codes), 29
- convert.scene, 31
- convert.slice, 29, 158
- convert.slice (Convert NIFTI Codes), 30
- convert.units, 29, 158
- convert.units (Convert NIFTI Codes), 30
- data.type (data\_type-methods), 33
- data.type,anzl-method
  - (data\_type-methods), 33
- data.type,nifti-method
  - (data\_type-methods), 33
- data.type<- (data\_type-methods), 33
- data.type<- ,anzl-method
  - (data\_type-methods), 33

- data.type<- ,nifti-method  
(data\_type-methods), 33
- data\_type (data\_type-methods), 33
- data\_type, anlz-method  
(data\_type-methods), 33
- data\_type, nifti-method  
(data\_type-methods), 33
- data\_type, niftiImage-method  
(data\_type-methods), 33
- data\_type-methods, 33
- data\_type-methods, (data\_type-methods),  
33
- data\_type<- (data\_type-methods), 33
- data\_type<- ,anlz-method  
(data\_type-methods), 33
- data\_type<- ,nifti-method  
(data\_type-methods), 33
- datatype (datatype-methods), 32
- datatype, anlz-method  
(datatype-methods), 32
- datatype, ANY-method (datatype-methods),  
32
- datatype, nifti-method  
(datatype-methods), 32
- datatype, niftiImage-method  
(data\_type-methods), 33
- datatype-methods, 32
- datatype-methods, (datatype-methods), 32
- datatype<- (datatype-methods), 32
- datatype<- ,anlz-method  
(datatype-methods), 32
- datatype<- ,nifti-method  
(datatype-methods), 32
- db.name (db\_name-methods), 35
- db.name, anlz-method (db\_name-methods),  
35
- db.name, nifti-method (db\_name-methods),  
35
- db.name<- (db\_name-methods), 35
- db.name<- ,anlz-method  
(db\_name-methods), 35
- db.name<- ,nifti-method  
(db\_name-methods), 35
- db\_name (db\_name-methods), 35
- db\_name, anlz-method (db\_name-methods),  
35
- db\_name, nifti-method (db\_name-methods),  
35
- db\_name-methods, 35
- db\_name-methods, (db\_name-methods), 35
- db\_name<- (db\_name-methods), 35
- db\_name<- ,anlz-method  
(db\_name-methods), 35
- db\_name<- ,nifti-method  
(db\_name-methods), 35
- descrip (descrip-methods), 36
- descrip, anlz-method (descrip-methods),  
36
- descrip, nifti-method (descrip-methods),  
36
- descrip, niftiImage-method  
(descrip-methods), 36
- descrip-methods, 36
- descrip-methods, (descrip-methods), 36
- descrip<- (descrip-methods), 36
- descrip<- ,anlz-method  
(descrip-methods), 36
- descrip<- ,nifti-method  
(descrip-methods), 36
- descrip<- ,niftiImage-method  
(descrip-methods), 36
- dim2freq (xyzt2space), 157
- dim2phase (xyzt2space), 157
- dim2slice (xyzt2space), 157
- dim\_ (dim\_-methods), 38
- dim\_, anlz-method (dim\_-methods), 38
- dim\_, ANY-method (dim\_-methods), 38
- dim\_, nifti-method (dim\_-methods), 38
- dim\_-methods, 38
- dim\_-methods, (dim\_-methods), 38
- dim\_<- (dim\_-methods), 38
- dim\_<- ,anlz-method (dim\_-methods), 38
- dim\_<- ,nifti-method (dim\_-methods), 38
- dim\_info (dim\_info-methods), 39
- dim\_info, nifti-method  
(dim\_info-methods), 39
- dim\_info-methods, 39
- dim\_info-methods, (dim\_info-methods), 39
- dim\_info<- (dim\_info-methods), 39
- dim\_info<- ,nifti-method  
(dim\_info-methods), 39
- dim\_un0 (dim\_un0-methods), 40
- dim\_un0, anlz-method (dim\_un0-methods),  
40
- dim\_un0-methods, 40
- dim\_un0-methods, (dim\_un0-methods), 40

- dim\_un0<- (dim\_un0-methods), 40
- dim\_un0<- ,anzl-method
  - (dim\_un0-methods), 40
- drop\_img\_dim(dropImageDimension), 41
- dropImageDimension, 41
- enableAuditTrail (Audit Trails), 12
- exp\_date (exp\_date-methods), 42
- exp\_date,anzl-method
  - (exp\_date-methods), 42
- exp\_date-methods, 42
- exp\_date-methods, (exp\_date-methods), 42
- exp\_date<- (exp\_date-methods), 42
- exp\_date<- ,anzl-method
  - (exp\_date-methods), 42
- exp\_time (exp\_time-methods), 43
- exp\_time,anzl-method
  - (exp\_time-methods), 43
- exp\_time-methods, 43
- exp\_time-methods, (exp\_time-methods), 43
- exp\_time<- (exp\_time-methods), 43
- exp\_time<- ,anzl-method
  - (exp\_time-methods), 43
- extender (extender-methods), 44
- extender,nifti-method
  - (extender-methods), 44
- extender-methods, 44
- extender-methods, (extender-methods), 44
- extender<- (extender-methods), 44
- extender<- ,nifti-method
  - (extender-methods), 44
- extents (extents-methods), 45
- extents,anzl-method (extents-methods), 45
- extents,nifti-method (extents-methods), 45
- extents-methods, 45
- extents-methods, (extents-methods), 45
- extents<- (extents-methods), 45
- extents<- ,anzl-method
  - (extents-methods), 45
- extents<- ,nifti-method
  - (extents-methods), 45
- field.skip (field\_skip-methods), 46
- field.skip,anzl-method
  - (field\_skip-methods), 46
- field.skip,nifti-method
  - (field\_skip-methods), 46
- field.skip<- (field\_skip-methods), 46
- field.skip<- ,anzl-method
  - (field\_skip-methods), 46
- field\_skip (field\_skip-methods), 46
- field\_skip,anzl-method
  - (field\_skip-methods), 46
- field\_skip-methods, 46
- field\_skip-methods, (field\_skip-methods), 46
- field\_skip<- (field\_skip-methods), 46
- field\_skip<- ,anzl-method
  - (field\_skip-methods), 46
- funused1 (funused1-methods), 47
- funused1,anzl-method
  - (funused1-methods), 47
- funused1-methods, 47
- funused1-methods, (funused1-methods), 47
- funused1<- (funused1-methods), 47
- funused1<- ,anzl-method
  - (funused1-methods), 47
- funused2 (funused2-methods), 48
- funused2,anzl-method
  - (funused2-methods), 48
- funused2-methods, 48
- funused2-methods, (funused2-methods), 48
- funused2<- (funused2-methods), 48
- funused2<- ,anzl-method
  - (funused2-methods), 48
- funused3 (funused3-methods), 49
- funused3,anzl-method
  - (funused3-methods), 49
- funused3-methods, 49
- funused3-methods, (funused3-methods), 49
- funused3<- (funused3-methods), 49
- funused3<- ,anzl-method
  - (funused3-methods), 49
- generated (generated-methods), 50
- generated,anzl-method
  - (generated-methods), 50
- generated-methods, 50
- generated-methods, (generated-methods), 50
- generated<- (generated-methods), 50
- generated<- ,anzl-method
  - (generated-methods), 50
- getLastCallWithName (Audit Trails), 12
- glmax (glmax-methods), 51
- glmax,anzl-method (glmax-methods), 51

- glmax,nifti-method (glmax-methods), 51
- glmax-methods, 51
- glmax-methods, (glmax-methods), 51
- glmax<- (glmax-methods), 51
- glmax<-, anlz-method (glmax-methods), 51
- glmax<-, nifti-method (glmax-methods), 51
- glmin (glmin-methods), 52
- glmin, anlz-method (glmin-methods), 52
- glmin, nifti-method (glmin-methods), 52
- glmin-methods, 52
- glmin-methods, (glmin-methods), 52
- glmin<- (glmin-methods), 52
- glmin<-, anlz-method (glmin-methods), 52
- glmin<-, nifti-method (glmin-methods), 52
  
- hist\_un0 (hist\_un0-methods), 53
- hist\_un0, anlz-method  
(hist\_un0-methods), 53
- hist\_un0-methods, 53
- hist\_un0-methods, (hist\_un0-methods), 53
- hist\_un0<- (hist\_un0-methods), 53
- hist\_un0<-, anlz-method  
(hist\_un0-methods), 53
- hkey.un0 (hkey\_un0-methods), 54
- hkey.un0, anlz-method  
(hkey\_un0-methods), 54
- hkey.un0, nifti-method  
(hkey\_un0-methods), 54
- hkey.un0<- (hkey\_un0-methods), 54
- hkey.un0<-, anlz-method  
(hkey\_un0-methods), 54
- hkey\_un0 (hkey\_un0-methods), 54
- hkey\_un0, anlz-method  
(hkey\_un0-methods), 54
- hkey\_un0-methods, 54
- hkey\_un0-methods, (hkey\_un0-methods), 54
- hkey\_un0<- (hkey\_un0-methods), 54
- hkey\_un0<-, anlz-method  
(hkey\_un0-methods), 54
- hotmetal, 55, 141
  
- image, 57, 123, 132
- image, afni-method (image-methods), 56
- image, anlz-method (image-methods), 56
- image, ANY-method (image-methods), 56
- image, nifti-method (image-methods), 56
- image-methods, 56
- image.nifti (image-methods), 56
- img\_data (img\_data-methods), 58
- img\_data, anlz-method  
(img\_data-methods), 58
- img\_data, ANY-method (img\_data-methods),  
58
- img\_data, character-method  
(img\_data-methods), 58
- img\_data, nifti-method  
(img\_data-methods), 58
- img\_data-methods, 58
- img\_data-methods, (img\_data-methods), 58
- img\_data<- (img\_data-methods), 58
- img\_data<-, anlz-method  
(img\_data-methods), 58
- img\_data<-, nifti-method  
(img\_data-methods), 58
- img\_length, 59
- integerTranslation, 59
- intent.code (intent\_code-methods), 60
- intent.code, nifti-method  
(intent\_code-methods), 60
- intent.code<- (intent\_code-methods), 60
- intent.code<-, nifti-method  
(intent\_code-methods), 60
- intent.name (intent\_name-methods), 61
- intent.name, nifti-method  
(intent\_name-methods), 61
- intent.name<- (intent\_name-methods), 61
- intent.name<-, nifti-method  
(intent\_name-methods), 61
- intent.p1 (intent\_p1-methods), 63
- intent.p1, nifti-method  
(intent\_p1-methods), 63
- intent.p1<- (intent\_p1-methods), 63
- intent.p1<-, nifti-method  
(intent\_p1-methods), 63
- intent.p2 (intent\_p2-methods), 64
- intent.p2, nifti-method  
(intent\_p2-methods), 64
- intent.p2<- (intent\_p2-methods), 64
- intent.p2<-, nifti-method  
(intent\_p2-methods), 64
- intent.p3 (intent\_p3-methods), 65
- intent.p3, nifti-method  
(intent\_p3-methods), 65
- intent.p3<- (intent\_p3-methods), 65
- intent.p3<-, nifti-method  
(intent\_p3-methods), 65
- intent\_code (intent\_code-methods), 60

- intent\_code,nifti-method  
(intent\_code-methods), 60
- intent\_code,niftiImage-method  
(intent\_code-methods), 60
- intent\_code-methods, 60
- intent\_code-methods,  
(intent\_code-methods), 60
- intent\_code<- (intent\_code-methods), 60
- intent\_code<- ,nifti-method  
(intent\_code-methods), 60
- intent\_code<- ,niftiImage-method  
(intent\_code-methods), 60
- intent\_name (intent\_name-methods), 61
- intent\_name,nifti-method  
(intent\_name-methods), 61
- intent\_name,niftiImage-method  
(intent\_name-methods), 61
- intent\_name-methods, 61
- intent\_name-methods,  
(intent\_name-methods), 61
- intent\_name<- (intent\_name-methods), 61
- intent\_name<- ,nifti-method  
(intent\_name-methods), 61
- intent\_name<- ,niftiImage-method  
(intent\_name-methods), 61
- intent\_p1 (intent\_p1-methods), 63
- intent\_p1,nifti-method  
(intent\_p1-methods), 63
- intent\_p1,niftiImage-method  
(intent\_p1-methods), 63
- intent\_p1-methods, 63
- intent\_p1-methods, (intent\_p1-methods),  
63
- intent\_p1<- (intent\_p1-methods), 63
- intent\_p1<- ,nifti-method  
(intent\_p1-methods), 63
- intent\_p1<- ,niftiImage-method  
(intent\_p1-methods), 63
- intent\_p2 (intent\_p2-methods), 64
- intent\_p2,nifti-method  
(intent\_p2-methods), 64
- intent\_p2,niftiImage-method  
(intent\_p2-methods), 64
- intent\_p2-methods, 64
- intent\_p2-methods, (intent\_p2-methods),  
64
- intent\_p2<- (intent\_p2-methods), 64
- intent\_p2<- ,nifti-method  
(intent\_p2-methods), 64
- intent\_p2<- ,niftiImage-method  
(intent\_p2-methods), 64
- intent\_p3 (intent\_p3-methods), 65
- intent\_p3,nifti-method  
(intent\_p3-methods), 65
- intent\_p3,niftiImage-method  
(intent\_p3-methods), 65
- intent\_p3-methods, 65
- intent\_p3-methods, (intent\_p3-methods),  
65
- intent\_p3<- (intent\_p3-methods), 65
- intent\_p3<- ,nifti-method  
(intent\_p3-methods), 65
- intent\_p3<- ,niftiImage-method  
(intent\_p3-methods), 65
- internalImage (internalImage-class), 67
- internalImage-class, 67
- inverseReorient, 94
- inverseReorient (reorient), 112
- invertIntegerTranslation  
(integerTranslation), 59
- is.afni, 67
- is.anlz, 68
- is.nifti, 68
- is.niftiExtension (is.nifti), 68
- magic (magic-methods), 69
- magic,nifti-method (magic-methods), 69
- magic,niftiImage-method  
(magic-methods), 69
- magic-methods, 69
- magic-methods, (magic-methods), 69
- magic<- (magic-methods), 69
- magic<- ,nifti-method (magic-methods), 69
- magic<- ,niftiImage-method  
(magic-methods), 69
- matrix, 7, 10, 73
- newAuditTrail (Audit Trails), 12
- nifti, 7, 8, 10, 68, 69, 70, 71, 75–78, 84, 109
- nifti-class, 72
- nifti-operators, 74
- nifti\_assign-methods, 77
- nifti\_header (readNIFTI), 109
- niftiAuditTrail, 74, 76
- niftiAuditTrail-class, 75
- niftiAuditTrailCreated (Audit Trails),  
12



- niftiAuditTrailEvent (Audit Trails), 12
- niftiAuditTrailSystemNode (Audit Trails), 12
- niftiAuditTrailSystemNodeEvent (Audit Trails), 12
- niftiAuditTrailToExtension (Audit Trails), 12
- niftiExtension, 10, 74, 75, 77
- niftiExtension-class, 76
- niftiExtensionSection-class, 76
- niftiExtensionToAuditTrail (Audit Trails), 12
- niftiImage (niftiImage-class), 77
- niftiImage-class, 77
- nii2oro, 78
- NSLI (nslI), 78
- nslI, 78
- NTIM (nslI), 78
- ntim (nslI), 78
  
- omax (omax-methods), 79
- omax, anlz-method (omax-methods), 79
- omax-methods, 79
- omax-methods, (omax-methods), 79
- omax<- (omax-methods), 79
- omax<- , anlz-method (omax-methods), 79
- omin (omin-methods), 80
- omin, anlz-method (omin-methods), 80
- omin-methods, 80
- omin-methods, (omin-methods), 80
- omin<- (omin-methods), 80
- omin<- , anlz-method (omin-methods), 80
- onfile, 81
- Ops, anlz, anlz-method (anlz-nifti-ops), 10
- Ops, anlz, nifti-method (anlz-nifti-ops), 10
- Ops, anlz, numeric-method (anlz-nifti-ops), 10
- Ops, nifti, anlz-method (anlz-nifti-ops), 10
- Ops, nifti, nifti-method (nifti-operators), 74
- Ops, nifti, numeric-method (nifti-operators), 74
- Ops, numeric, anlz-method (anlz-nifti-ops), 10
- Ops, numeric, nifti-method (nifti-operators), 74
  
- options, 151, 153, 156
- orient (orient-methods), 81
- orient, anlz-method (orient-methods), 81
- orient-methods, 81
- orient-methods, (orient-methods), 81
- orient<- (orient-methods), 81
- orient<- , anlz-method (orient-methods), 81
- orientation-methods, 82
- origin (origin-methods), 83
- origin, anlz-method (origin-methods), 83
- origin, ANY-method (origin-methods), 83
- origin, nifti-method (origin-methods), 83
- origin-methods, 83
- origin-methods, (origin-methods), 83
- origin<- (origin-methods), 83
- origin<- , anlz-method (origin-methods), 83
- origin<- , nifti-method (origin-methods), 83
  
- oro.nifti.info (Audit Trails), 12
- oro2nii, 84
- orthographic (orthographic-methods), 85
- orthographic, afni-method (orthographic-methods), 85
- orthographic, anlz-method (orthographic-methods), 85
- orthographic, array-method (orthographic-methods), 85
- orthographic, nifti-method (orthographic-methods), 85
- orthographic-methods, 85
- orthographic.nifti (orthographic-methods), 85
- overlay (overlay-methods), 88
- overlay, afni, afni-method (overlay-methods), 88
- overlay, afni, array-method (overlay-methods), 88
- overlay, anlz, anlz-method (overlay-methods), 88
- overlay, anlz, array-method (overlay-methods), 88
- overlay, anlz, nifti-method (overlay-methods), 88
- overlay, array, anlz-method (overlay-methods), 88
- overlay, array, array-method



- (overlay-methods), 88
- overlay,array,nifti-method
  - (overlay-methods), 88
- overlay,nifti,anlz-method
  - (overlay-methods), 88
- overlay,nifti,array-method
  - (overlay-methods), 88
- overlay,nifti,missing-method
  - (overlay-methods), 88
- overlay,nifti,nifti-method
  - (overlay-methods), 88
- overlay-methods, 88
- overlay.nifti (overlay-methods), 88
- patient.id (patient\_id-methods), 92
- patient.id,anlz-method
  - (patient\_id-methods), 92
- patient.id,nifti-method
  - (patient\_id-methods), 92
- patient.id<- (patient\_id-methods), 92
- patient.id<-,anlz-method
  - (patient\_id-methods), 92
- patient\_id (patient\_id-methods), 92
- patient\_id,anlz-method
  - (patient\_id-methods), 92
- patient\_id-methods, 92
- patient\_id-methods,
  - (patient\_id-methods), 92
- patient\_id<- (patient\_id-methods), 92
- patient\_id<-,anlz-method
  - (patient\_id-methods), 92
- performPermutation, 94, 112
- pixdim (pixdim-methods), 94
- pixdim,anlz-method (pixdim-methods), 94
- pixdim,ANY-method (pixdim-methods), 94
- pixdim,nifti-method (pixdim-methods), 94
- pixdim-methods, 94
- pixdim-methods, (pixdim-methods), 94
- pixdim<- (pixdim-methods), 94
- pixdim<-,anlz-method (pixdim-methods), 94
- pixdim<-,ANY-method (pixdim-methods), 94
- pixdim<-,nifti-method (pixdim-methods), 94
- qform (orientation-methods), 82
- qform,nifti-method
  - (orientation-methods), 82
- qform-methods (orientation-methods), 82
- qform.code (qform\_code-methods), 96
- qform.code,nifti-method
  - (qform\_code-methods), 96
- qform.code<- (qform\_code-methods), 96
- qform.code<-,nifti-method
  - (qform\_code-methods), 96
- qform\_code (qform\_code-methods), 96
- qform\_code,nifti-method
  - (qform\_code-methods), 96
- qform\_code,niftiImage-method
  - (qform\_code-methods), 96
- qform\_code-methods, 96
- qform\_code-methods,
  - (qform\_code-methods), 96
- qform\_code<- (qform\_code-methods), 96
- qform\_code<-,nifti-method
  - (qform\_code-methods), 96
- qform\_code<-,niftiImage-method
  - (qform\_code-methods), 96
- qoffset.x (qoffset\_x-methods), 97
- qoffset.x,nifti-method
  - (qoffset\_x-methods), 97
- qoffset.x<- (qoffset\_x-methods), 97
- qoffset.x<-,nifti-method
  - (qoffset\_x-methods), 97
- qoffset.y (qoffset\_y-methods), 98
- qoffset.y,nifti-method
  - (qoffset\_y-methods), 98
- qoffset.y<- (qoffset\_y-methods), 98
- qoffset.y<-,nifti-method
  - (qoffset\_y-methods), 98
- qoffset.z (qoffset\_z-methods), 100
- qoffset.z,nifti-method
  - (qoffset\_z-methods), 100
- qoffset.z<- (qoffset\_z-methods), 100
- qoffset.z<-,nifti-method
  - (qoffset\_z-methods), 100
- qoffset\_x (qoffset\_x-methods), 97
- qoffset\_x,nifti-method
  - (qoffset\_x-methods), 97
- qoffset\_x,niftiImage-method
  - (qoffset\_x-methods), 97
- qoffset\_x-methods, 97
- qoffset\_x-methods, (qoffset\_x-methods), 97
- qoffset\_x<- (qoffset\_x-methods), 97
- qoffset\_x<-,nifti-method
  - (qoffset\_x-methods), 97

- qoffset\_x<- ,niftiImage-method  
(qoffset\_x-methods), 97
- qoffset\_y (qoffset\_y-methods), 98
- qoffset\_y, nifti-method  
(qoffset\_y-methods), 98
- qoffset\_y, niftiImage-method  
(qoffset\_y-methods), 98
- qoffset\_y-methods, 98
- qoffset\_y-methods, (qoffset\_y-methods),  
98
- qoffset\_y<- (qoffset\_y-methods), 98
- qoffset\_y<- ,nifti-method  
(qoffset\_y-methods), 98
- qoffset\_y<- ,niftiImage-method  
(qoffset\_y-methods), 98
- qoffset\_z (qoffset\_z-methods), 100
- qoffset\_z, nifti-method  
(qoffset\_z-methods), 100
- qoffset\_z, niftiImage-method  
(qoffset\_z-methods), 100
- qoffset\_z-methods, 100
- qoffset\_z-methods, (qoffset\_z-methods),  
100
- qoffset\_z<- (qoffset\_z-methods), 100
- qoffset\_z<- ,nifti-method  
(qoffset\_z-methods), 100
- qoffset\_z<- ,niftiImage-method  
(qoffset\_z-methods), 100
- quatern.b (quatern\_b-methods), 102
- quatern.b, nifti-method  
(quatern\_b-methods), 102
- quatern.b<- (quatern\_b-methods), 102
- quatern.b<- ,nifti-method  
(quatern\_b-methods), 102
- quatern.c (quatern\_c-methods), 104
- quatern.c, nifti-method  
(quatern\_c-methods), 104
- quatern.c<- (quatern\_c-methods), 104
- quatern.c<- ,nifti-method  
(quatern\_c-methods), 104
- quatern.d (quatern\_d-methods), 105
- quatern.d, nifti-method  
(quatern\_d-methods), 105
- quatern.d<- (quatern\_d-methods), 105
- quatern.d<- ,nifti-method  
(quatern\_d-methods), 105
- quatern\_b (quatern\_b-methods), 102
- quatern\_b, nifti-method  
(quatern\_b-methods), 102
- quatern\_b<- (quatern\_b-methods), 102
- quatern\_b<- ,nifti-method  
(quatern\_b-methods), 102
- quatern\_b-methods, 102
- quatern\_b-methods, (quatern\_b-methods),  
102
- quatern\_b<- (quatern\_b-methods), 102
- quatern\_b<- ,nifti-method  
(quatern\_b-methods), 102
- quatern\_b<- ,niftiImage-method  
(quatern\_b-methods), 102
- quatern\_c (quatern\_c-methods), 104
- quatern\_c, nifti-method  
(quatern\_c-methods), 104
- quatern\_c, niftiImage-method  
(quatern\_c-methods), 104
- quatern\_c-methods, 104
- quatern\_c-methods, (quatern\_c-methods),  
104
- quatern\_c<- (quatern\_c-methods), 104
- quatern\_c<- ,nifti-method  
(quatern\_c-methods), 104
- quatern\_c<- ,niftiImage-method  
(quatern\_c-methods), 104
- quatern\_d (quatern\_d-methods), 105
- quatern\_d, nifti-method  
(quatern\_d-methods), 105
- quatern\_d, niftiImage-method  
(quatern\_d-methods), 105
- quatern\_d-methods, 105
- quatern\_d-methods, (quatern\_d-methods),  
105
- quatern\_d<- (quatern\_d-methods), 105
- quatern\_d<- ,nifti-method  
(quatern\_d-methods), 105
- quatern\_d<- ,niftiImage-method  
(quatern\_d-methods), 105
- quaternion2mat44 (quaternion2rotation),  
101
- quaternion2rotation, 101
- readAFNI, 106, 110
- readANALYZE, 79, 107, 107, 110
- readNIFTI, 79, 107, 108, 109
- regular (regular-methods), 111
- regular, anlz-method (regular-methods),  
111
- regular, nifti-method (regular-methods),  
111

- regular-methods, 111
- regular-methods, (regular-methods), 111
- regular<- (regular-methods), 111
- regular<-, anlz-method  
(regular-methods), 111
- regular<-, nifti-method  
(regular-methods), 111
- reorient, 94, 112
- resetSlopeIntercept, 113
- rmgz (rmniigz), 113
- rmhdr (rmniigz), 113
- rmhdrgz (rmniigz), 113
- rmimg (rmniigz), 113
- rmimggz (rmniigz), 113
- rmnii (rmniigz), 113
- rmniigz, 113
  
- scannum (scannum-methods), 114
- scannum, anlz-method (scannum-methods),  
114
- scannum-methods, 114
- scannum-methods, (scannum-methods), 114
- scannum<- (scannum-methods), 114
- scannum<-, anlz-method  
(scannum-methods), 114
- scl.inter (scl\_inter-methods), 115
- scl.inter, nifti-method  
(scl\_inter-methods), 115
- scl.inter, niftiImage-method  
(scl\_inter-methods), 115
- scl.inter<- (scl\_inter-methods), 115
- scl.inter<-, nifti-method  
(scl\_inter-methods), 115
- scl.slope (scl\_slope-methods), 116
- scl.slope, nifti-method  
(scl\_slope-methods), 116
- scl.slope, niftiImage-method  
(scl\_slope-methods), 116
- scl.slope<- (scl\_slope-methods), 116
- scl.slope<-, nifti-method  
(scl\_slope-methods), 116
- scl\_inter (scl\_inter-methods), 115
- scl\_inter, nifti-method  
(scl\_inter-methods), 115
- scl\_inter, niftiImage-method  
(scl\_inter-methods), 115
- scl\_inter-methods, 115
- scl\_inter-methods, (scl\_inter-methods),  
115
  
- scl\_inter<- (scl\_inter-methods), 115
- scl\_inter<-, nifti-method  
(scl\_inter-methods), 115
- scl\_slope (scl\_slope-methods), 116
- scl\_slope, nifti-method  
(scl\_slope-methods), 116
- scl\_slope, niftiImage-method  
(scl\_slope-methods), 116
- scl\_slope-methods, 116
- scl\_slope-methods, (scl\_slope-methods),  
116
- scl\_slope<- (scl\_slope-methods), 116
- scl\_slope<-, nifti-method  
(scl\_slope-methods), 116
- session.error (session\_error-methods),  
118
- session.error, anlz-method  
(session\_error-methods), 118
- session.error, nifti-method  
(session\_error-methods), 118
- session.error<-  
(session\_error-methods), 118
- session.error<-, anlz-method  
(session\_error-methods), 118
- session.error<-, nifti-method  
(session\_error-methods), 118
- session\_error (session\_error-methods),  
118
- session\_error, anlz-method  
(session\_error-methods), 118
- session\_error, nifti-method  
(session\_error-methods), 118
- session\_error-methods, 118
- session\_error-methods,  
(session\_error-methods), 118
- session\_error<-  
(session\_error-methods), 118
- session\_error<-, anlz-method  
(session\_error-methods), 118
- session\_error<-, nifti-method  
(session\_error-methods), 118
- sform (orientation-methods), 82
- sform, nifti-method  
(orientation-methods), 82
- sform-methods (orientation-methods), 82
- sform.code (sform\_code-methods), 119
- sform.code, nifti-method  
(sform\_code-methods), 119

- sform.code<- (sform\_code-methods), 119
- sform.code<- ,nifti-method
  - (sform\_code-methods), 119
- sform\_code (sform\_code-methods), 119
- sform\_code ,nifti-method
  - (sform\_code-methods), 119
- sform\_code ,niftiImage-method
  - (sform\_code-methods), 119
- sform\_code-methods, 119
- sform\_code-methods,
  - (sform\_code-methods), 119
- sform\_code<- (sform\_code-methods), 119
- sform\_code<- ,nifti-method
  - (sform\_code-methods), 119
- sform\_code<- ,niftiImage-method
  - (sform\_code-methods), 119
- show ,afni-method (afni-class), 5
- show ,anzl-method (anzl-class), 8
- show ,nifti-method (nifti-class), 72
- sizeof\_hdr (sizeof\_hdr-methods), 120
- sizeof\_hdr ,anzl-method
  - (sizeof\_hdr-methods), 120
- sizeof\_hdr ,nifti-method
  - (sizeof\_hdr-methods), 120
- sizeof\_hdr (sizeof\_hdr-methods), 120
- sizeof\_hdr ,anzl-method
  - (sizeof\_hdr-methods), 120
- sizeof\_hdr ,nifti-method
  - (sizeof\_hdr-methods), 120
- sizeof\_hdr-methods, 120
- sizeof\_hdr-methods,
  - (sizeof\_hdr-methods), 120
- sizeof\_hdr<- (sizeof\_hdr-methods), 120
- sizeof\_hdr<- ,anzl-method
  - (sizeof\_hdr-methods), 120
- sizeof\_hdr<- ,nifti-method
  - (sizeof\_hdr-methods), 120
- slice (slice-methods), 121
- slice ,afni-method (slice-methods), 121
- slice ,anzl-method (slice-methods), 121
- slice ,ANY-method (slice-methods), 121
- slice ,array-method (slice-methods), 121
- slice ,nifti-method (slice-methods), 121
- slice-methods, 121
- slice.code (slice\_code-methods), 124
- slice.code ,nifti-method
  - (slice\_code-methods), 124
- slice.code<- (slice\_code-methods), 124
- slice.code<- ,nifti-method
  - (slice\_code-methods), 124
- slice.code<- ,niftiImage-method
  - (slice\_code-methods), 124
- slice.code<- ,niftiImage-method
  - (slice\_code-methods), 124
- slice.code<- ,niftiImage-method
  - (slice\_code-methods), 124
- slice.duration
  - (slice\_duration-methods), 125
- slice.duration ,nifti-method
  - (slice\_duration-methods), 125
- slice.duration<-
  - (slice\_duration-methods), 125
- slice.duration<- ,nifti-method
  - (slice\_duration-methods), 125
- slice.end (slice\_end-methods), 126
- slice.end ,nifti-method
  - (slice\_end-methods), 126
- slice.end ,niftiImage-method
  - (slice\_end-methods), 126
- slice.end<- (slice\_end-methods), 126
- slice.end<- ,nifti-method
  - (slice\_end-methods), 126
- slice.end<- ,niftiImage-method
  - (slice\_end-methods), 126
- slice.nifti (slice-methods), 121
- slice.start (slice\_start-methods), 132
- slice.start ,nifti-method
  - (slice\_start-methods), 132
- slice.start ,niftiImage-method
  - (slice\_start-methods), 132
- slice.start<- (slice\_start-methods), 132
- slice.start<- ,nifti-method
  - (slice\_start-methods), 132
- slice.start<- ,niftiImage-method
  - (slice\_start-methods), 132
- slice\_code (slice\_code-methods), 124
- slice\_code ,nifti-method
  - (slice\_code-methods), 124
- slice\_code ,niftiImage-method
  - (slice\_code-methods), 124
- slice\_code-methods, 124
- slice\_code-methods,
  - (slice\_code-methods), 124
- slice\_code<- (slice\_code-methods), 124
- slice\_code<- ,nifti-method
  - (slice\_code-methods), 124
- slice\_code<- ,niftiImage-method
  - (slice\_code-methods), 124
- slice\_code<- ,niftiImage-method
  - (slice\_code-methods), 124
- slice\_duration
  - (slice\_duration-methods), 125
- slice\_duration ,nifti-method
  - (slice\_duration-methods), 125

- slice\_duration, niftiImage-method  
(slice\_duration-methods), 125
- slice\_duration-methods, 125
- slice\_duration-methods,  
(slice\_duration-methods), 125
- slice\_duration<-  
(slice\_duration-methods), 125
- slice\_duration<- ,nifti-method  
(slice\_duration-methods), 125
- slice\_duration<- ,niftiImage-method  
(slice\_duration-methods), 125
- slice\_end (slice\_end-methods), 126
- slice\_end, nifti-method  
(slice\_end-methods), 126
- slice\_end, niftiImage-method  
(slice\_end-methods), 126
- slice\_end-methods, 126
- slice\_end-methods, (slice\_end-methods),  
126
- slice\_end<- (slice\_end-methods), 126
- slice\_end<- ,nifti-method  
(slice\_end-methods), 126
- slice\_end<- ,niftiImage-method  
(slice\_end-methods), 126
- slice\_overlay (slice\_overlay-methods),  
128
- slice\_overlay, afni, afni-method  
(slice\_overlay-methods), 128
- slice\_overlay, afni, array-method  
(slice\_overlay-methods), 128
- slice\_overlay, anlz, anlz-method  
(slice\_overlay-methods), 128
- slice\_overlay, anlz, array-method  
(slice\_overlay-methods), 128
- slice\_overlay, anlz, nifti-method  
(slice\_overlay-methods), 128
- slice\_overlay, array, anlz-method  
(slice\_overlay-methods), 128
- slice\_overlay, array, array-method  
(slice\_overlay-methods), 128
- slice\_overlay, array, nifti-method  
(slice\_overlay-methods), 128
- slice\_overlay, nifti, anlz-method  
(slice\_overlay-methods), 128
- slice\_overlay, nifti, array-method  
(slice\_overlay-methods), 128
- slice\_overlay, nifti, missing-method  
(slice\_overlay-methods), 128
- slice\_overlay, nifti, nifti-method  
(slice\_overlay-methods), 128
- slice\_overlay-methods, 128
- slice\_overlay.nifti  
(slice\_overlay-methods), 128
- slice\_start (slice\_start-methods), 132
- slice\_start, nifti-method  
(slice\_start-methods), 132
- slice\_start, niftiImage-method  
(slice\_start-methods), 132
- slice\_start-methods, 132
- slice\_start-methods,  
(slice\_start-methods), 132
- slice\_start<- (slice\_start-methods), 132
- slice\_start<- ,nifti-method  
(slice\_start-methods), 132
- slice\_start<- ,niftiImage-method  
(slice\_start-methods), 132
- smax (smax-methods), 134
- smax, anlz-method (smax-methods), 134
- smax-methods, 134
- smax-methods, (smax-methods), 134
- smax<- (smax-methods), 134
- smax<- ,anz-method (smax-methods), 134
- smin (smin-methods), 135
- smin, anlz-method (smin-methods), 135
- smin-methods, 135
- smin-methods, (smin-methods), 135
- smin<- (smin-methods), 135
- smin<- ,anz-method (smin-methods), 135
- space.time2xyz (xyz2space), 157
- srow.x (srow\_x-methods), 136
- srow.x, nifti-method (srow\_x-methods),  
136
- srow.x<- (srow\_x-methods), 136
- srow.x<- ,nifti-method (srow\_x-methods),  
136
- srow.y (srow\_y-methods), 137
- srow.y, nifti-method (srow\_y-methods),  
137
- srow.y<- (srow\_y-methods), 137
- srow.y<- ,nifti-method (srow\_y-methods),  
137
- srow.z (srow\_z-methods), 138
- srow.z, nifti-method (srow\_z-methods),  
138
- srow.z<- (srow\_z-methods), 138
- srow.z<- ,nifti-method (srow\_z-methods),

- 138
- srow\_x (srow\_x-methods), 136
- srow\_x,nifti-method (srow\_x-methods), 136
- srow\_x,niftiImage-method (srow\_x-methods), 136
- srow\_x-methods, 136
- srow\_x-methods, (srow\_x-methods), 136
- srow\_x<- (srow\_x-methods), 136
- srow\_x<-,nifti-method (srow\_x-methods), 136
- srow\_x<-,niftiImage-method (srow\_x-methods), 136
- srow\_y (srow\_y-methods), 137
- srow\_y,nifti-method (srow\_y-methods), 137
- srow\_y,niftiImage-method (srow\_y-methods), 137
- srow\_y-methods, 137
- srow\_y-methods, (srow\_y-methods), 137
- srow\_y<- (srow\_y-methods), 137
- srow\_y<-,nifti-method (srow\_y-methods), 137
- srow\_y<-,niftiImage-method (srow\_y-methods), 137
- srow\_z (srow\_z-methods), 138
- srow\_z,nifti-method (srow\_z-methods), 138
- srow\_z,niftiImage-method (srow\_z-methods), 138
- srow\_z-methods, 138
- srow\_z-methods, (srow\_z-methods), 138
- srow\_z<- (srow\_z-methods), 138
- srow\_z<-,nifti-method (srow\_z-methods), 138
- srow\_z<-,niftiImage-method (srow\_z-methods), 138
- start\_field (start\_field-methods), 139
- start\_field,anlz-method (start\_field-methods), 139
- start\_field-methods, 139
- start\_field-methods, (start\_field-methods), 139
- start\_field<- (start\_field-methods), 139
- start\_field<-,anlz-method (start\_field-methods), 139
- structure, 7, 10, 73
- terrain.colors, 55, 141
- tim.colors, 55, 140
- toffset (toffset-methods), 141
- toffset,nifti-method (toffset-methods), 141
- toffset,niftiImage-method (toffset-methods), 141
- toffset-methods, 141
- toffset-methods, (toffset-methods), 141
- toffset<- (toffset-methods), 141
- toffset<-,nifti-method (toffset-methods), 141
- toffset<-,niftiImage-method (toffset-methods), 141
- topo.colors, 55, 141
- translateCoordinate, 142
- unused1 (unused1-methods), 143
- unused1,anlz-method (unused1-methods), 143
- unused1-methods, 143
- unused1-methods, (unused1-methods), 143
- unused1<- (unused1-methods), 143
- unused1<-,anlz-method (unused1-methods), 143
- vector, 7, 10, 73
- verified (verified-methods), 144
- verified,anlz-method (verified-methods), 144
- verified-methods, 144
- verified-methods, (verified-methods), 144
- verified<- (verified-methods), 144
- verified<-,anlz-method (verified-methods), 144
- views (views-methods), 145
- views,anlz-method (views-methods), 145
- views-methods, 145
- views-methods, (views-methods), 145
- views<- (views-methods), 145
- views<-,anlz-method (views-methods), 145
- vols.added (vols\_added-methods), 146
- vols.added,anlz-method (vols\_added-methods), 146
- vols.added,nifti-method (vols\_added-methods), 146
- vols.added<- (vols\_added-methods), 146
- vols.added<-,anlz-method (vols\_added-methods), 146

- voxls\_added (voxls\_added-methods), 146
- voxls\_added, anlz-method
  - (voxls\_added-methods), 146
- voxls\_added-methods, 146
- voxls\_added-methods,
  - (voxls\_added-methods), 146
- voxls\_added<- (voxls\_added-methods), 146
- voxls\_added<-, anlz-method
  - (voxls\_added-methods), 146
- vox.offset (vox\_offset-methods), 148
- vox.offset, anlz-method
  - (vox\_offset-methods), 148
- vox.offset, nifti-method
  - (vox\_offset-methods), 148
- vox.offset, niftiImage-method
  - (vox\_offset-methods), 148
- vox.offset<- (vox\_offset-methods), 148
- vox.offset<-, anlz-method
  - (vox\_offset-methods), 148
- vox.offset<-, nifti-method
  - (vox\_offset-methods), 148
- vox.units (vox\_units-methods), 150
- vox.units, anlz-method
  - (vox\_units-methods), 150
- vox.units, nifti-method
  - (vox\_units-methods), 150
- vox.units<- (vox\_units-methods), 150
- vox.units<-, anlz-method
  - (vox\_units-methods), 150
- vox\_offset (vox\_offset-methods), 148
- vox\_offset, anlz-method
  - (vox\_offset-methods), 148
- vox\_offset, nifti-method
  - (vox\_offset-methods), 148
- vox\_offset, niftiImage-method
  - (vox\_offset-methods), 148
- vox\_offset-methods, 148
- vox\_offset-methods,
  - (vox\_offset-methods), 148
- vox\_offset<- (vox\_offset-methods), 148
- vox\_offset<-, anlz-method
  - (vox\_offset-methods), 148
- vox\_offset<-, nifti-method
  - (vox\_offset-methods), 148
- vox\_units (vox\_units-methods), 150
- vox\_units, anlz-method
  - (vox\_units-methods), 150
- vox\_units-methods, 150
- vox\_units-methods, (vox\_units-methods),
  - 150
- vox\_units<- (vox\_units-methods), 150
- vox\_units<-, anlz-method
  - (vox\_units-methods), 150
- voxdim, 147
- voxres, 148
- writeAFNI, 153, 156
- writeAFNI (writeAFNI-methods), 151
- writeAFNI, afni-method
  - (writeAFNI-methods), 151
- writeAFNI, ANY-method
  - (writeAFNI-methods), 151
- writeAFNI-methods, 151
- writeANALYZE, 152, 156
- writeANALYZE (writeANALYZE-methods), 152
- writeANALYZE, anlz-method
  - (writeANALYZE-methods), 152
- writeANALYZE-methods, 152
- writeNIFTI, 81, 84, 152, 153
- writeNIFTI (writeNIFTI-methods), 154
- writeNIFTI, anlz-method
  - (writeNIFTI-methods), 154
- writeNIFTI, array-method
  - (writeNIFTI-methods), 154
- writeNIFTI, nifti-method
  - (writeNIFTI-methods), 154
- writeNIFTI, niftiExtension-method
  - (writeNIFTI-methods), 154
- writeNIFTI-methods, 154
- xyzt.units (xyzt\_units-methods), 159
- xyzt.units, nifti-method
  - (xyzt\_units-methods), 159
- xyzt.units<- (xyzt\_units-methods), 159
- xyzt.units<-, nifti-method
  - (xyzt\_units-methods), 159
- xyzt2space, 157
- xyzt2time (xyzt2space), 157
- xyzt\_units (xyzt\_units-methods), 159
- xyzt\_units, nifti-method
  - (xyzt\_units-methods), 159
- xyzt\_units, niftiImage-method
  - (xyzt\_units-methods), 159
- xyzt\_units-methods, 159
- xyzt\_units-methods,
  - (xyzt\_units-methods), 159
- xyzt\_units<- (xyzt\_units-methods), 159

xyzt\_units<-,nifti-method  
    (xyzt\_units-methods), [159](#)  
xyzt\_units<-,niftiImage-method  
    (xyzt\_units-methods), [159](#)  
zero\_trans (resetSlopeIntercept), [113](#)