Package ‘pedalfast.data’

December 1, 2021

Title PEDALFAST Data


Language en-US

License GPL-2

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports car, data.table, digest, dplyr, ggplot2, htmltools, knitr, lubridate, qwraps2 (>= 0.5.0), scales, testthat

VignetteBuilder knitr

RoxygenNote 7.1.2

NeedsCompilation no

Author Peter DeWitt [aut, cre] (<https://orcid.org/0000-0002-6391-0795>), Tell Bennet [aut] (<https://orcid.org/0000-0003-1483-4236>)

Maintainer Peter DeWitt <peter.dewitt@cuanschutz.edu>

Repository CRAN

Date/Publication 2021-12-01 11:10:02 UTC

R topics documented:

- fss_as_factor ................................................................. 2
- gcs ........................................................................ 3
- onezero ...................................................................... 4
- pedalfast ...................................................................... 5
- pedalfast_factor .............................................................. 6
Description

Mapping FSS Total scores (integer values) to categorical values.

Usage

fss_as_factor(x, long_label = FALSE, ...)

Arguments

x 
an integer vector
long_label 
logical if the score range should be prepended to the label.
...
not currently used.

Details

FSS scores are integer values from 6 to 30.
The a mapping of ranges of integer values to categories is

• FSS 6, 7: Good
• FSS 8, 9: Mildly abnormal
• FSS 10, 11, 12, 13, 14, 15: Moderately abnormal
• FSS 16, 17, 18, 19, 20, 21: Severe abnormal
• FSS 22, 23, 24, 25, 26, 27, 28, 29, 30: Very severely abnormal

Value

A factor of equal length to the input x with labels for the categorical ranges of FSS.

Examples

x <- seq(5, 32)
data.frame(x = x,
         short_label = fss_as_factor(x),
         long_label = fss_as_factor(x, long_label = TRUE))
### Description

Functions for mapping integer values to GCS labeled factor and visa versa

### Usage

```r
# gcs_as_integer

## S3 method for class 'factor'
gcs_as_integer(x, scale, ...)
## S3 method for class 'character'
gcs_as_integer(x, scale, ...)

# gcs_as_factor

## S3 method for class 'character'
gcs_as_factor(x, scale, long_label = FALSE, highest_first = FALSE, ...)
## S3 method for class 'numeric'
gcs_as_factor(x, scale, long_label = FALSE, highest_first = FALSE, ...)

gcs_ll
```

### Arguments

- **x**
  - a integer, factor, or character vector.
- **scale**
  - a character string to denote eye, motor, or verbal GCS scale.
- **...**
  - not currently used.
- **long_label**
  - logical to prepend the numeric value to the label of a factor.
- **highest_first**
  - logical if the factor levels should be ordered with the highest GCS score as the reference level, else the lowest GCS score as the reference level.

### Format

An object of class list of length 3.

### Value

- `gcs_as_factor` returns a factor of equal length to `x`.
- `gcs_as_integer` returns an integer vector of equal length to `x`. 
Examples

# Mapping from numeric values to factor:
nums <- c(0:7, 2.3)

# with short labels
data.frame(nums = nums,
    eye = gcs_as_factor(nums, scale = "eye"),
    motor = gcs_as_factor(nums, scale = "motor"),
    verbal = gcs_as_factor(nums, scale = "verbal"))

# with long labels
data.frame(nums = nums,
    eye = gcs_as_factor(nums, scale = "eye", long_label = TRUE),
    motor = gcs_as_factor(nums, scale = "motor", long_label = TRUE),
    verbal = gcs_as_factor(nums, scale = "verbal", long_label = TRUE))

# Mapping from factors/characters to numeric values
# A quick way to access the labels and numeric values
all_levels <- do.call(c, lapply(pedalfast.data::gcs_ll, names))
data.frame(lvls = all_levels,
    eye = gcs_as_integer(all_levels, scale = "eye"),
    motor = gcs_as_integer(all_levels, scale = "motor"),
    verbal = gcs_as_integer(all_levels, scale = "verbal")
)

# Order of the levels:
# The data values are the same, but the order of the levels differs.
gcs_as_factor(1:4, "eye", highest_first = FALSE)
gcs_as_factor(1:4, "eye", highest_first = TRUE)

foreach(onezero, 1/0 Flags)

describe(onezero)

Description

Turns Yes/No variables into 1/0 integers

Usage

onezero(x)
Arguments

- `x`: a character vector

Details

The input is forced to lowercase and only the first character, the "y" or "n", is used to map to the 1/0 integer values. The function allows for "1" and "0" to be in the character vector as well.

Value

an integer vector

Examples

```r
flag <- c("Y", "No", "NO", "no", "n", "YES", "Yes", "yEs", "1", "0")
onezero(flag)
```

---

**pedalfast**

**PEDALFAST Data**

Description

Single data frame for the PEDALFAST data.

Usage

`pedalfast`

Format

An object of class `data.frame` with 388 rows and 103 columns.

Details

`pedalfast` is one data frame with the whole of the exported data.

See Also

`vignette("datasets", package = "pedalfast.data")`
Description
A flavor of the base function factor but aimed to use specific default values for levels and labels based on the information in the pedalfast_metadata object.

Usage
pedalfast_factor(x, variable, label_with_level = FALSE, ...)

Arguments
x
a vector of data

variable
character string identifying the variable name in pedalfast_metadata defining the levels and labels for the factor.

label_with_level
(default to FALSE) labels will include the integer value. See examples.

...
not currently used.

Value
An object of class factor

References


Examples

data(pedalfast, pedalfast_metadata, package = "pedalfast.data")

# The Motor GCS in the emergency department is reported as an integer value.
str(pedalfast$gcsmotored)

# Each integer value a specific meaning
pedalfast_metadata[grepl("gcsmotored", pedalfast_metadata$variable), ]

# Creating the factor in base R
pedalfast_factor(x = c(1, 3, 2), variable = "gcsmotored")
pedalfast_factor(x = c(1, 3, 2), variable = "gcsmotored", label_with_level = TRUE)
**Description**

*pedalfast_metadata* provides documentation for each of variables in the *pedalfast*.

**Usage**

*pedalfast_metadata*

**Format**

An object of class *data.frame* with 103 rows and 3 columns.

**See Also**

`vignette("datasets", package = "pedalfast.data")`

---

**round_age**

**Round Age**

**Description**

Round age per FITBIR guidelines

**Usage**

`round_age(x, type = "character")`

**Arguments**

- **x**: a numeric vector
- **type**: defaults to character (default), also accepts numeric.
Details

FITBIR Definition for the "AgeYrs" data element: Value for participant's subject age, calculated as elapsed time since the birth of the participant/subject in years. The subject's age is typically recorded to the nearest full year completed, e.g. 11 years and 6 months should be recorded as 11 years.

Guidelines & Instructions: The subject's age is typically recorded to the nearest full year completed, e.g. 11 years and 6 months should be recorded as 11 years. For subjects which are under 1 year old, use decimal points and use the following convention: record 1 month as 0.083 (1/12), 2 months as 0.166 (2/12), 3 months as 0.25 (3/12), 4 months as 0.333 (4/12), 5 months as 0.416 (5/12), 6 months as 0.5 (6/12), 7 months as 0.583 (7/12), 8 months as 0.666 (8/12), 9 months as 0.75 (9/12), 10 months as 0.833 (10/12), 11 months as 0.916 (11/12) and 12 months as 1 year. For the individuals 90 or older, in order to preserve PII, please submit "150" and make a note this in the "general notes" column.

Value

a character or numeric vector depending on the value of type.

Examples

```r
ages <- c(92, 12.12, 89 + 10/12, 9.12, 9.73, 1.1, 1.75, ( 1:11 + 0.05 ) / 12,
2, 90)

round_age(ages)
round_age(ages, "numeric")
```

---

<table>
<thead>
<tr>
<th>yesno</th>
<th>Yes No Flags</th>
</tr>
</thead>
</table>

Description

Turns 1/0 into "Yes"/"No"

Usage

```r
yesno(x)
```

Arguments

```r
x an integer vector
```

Value

a character vector
Examples

```r
tag <- c(0, 1, 0, 0, 0)
yesno(tag)
```
Index

* datasets
  gcs, 3
  pedalfast, 5
  pedalfast_metadata, 7

fss_as_factor, 2

gcs, 3
  gcs_as_factor (gcs), 3
  gcs_as_integer (gcs), 3
  gcs_ll (gcs), 3

onezero, 4

pedalfast, 5
  pedalfast_factor, 6
  pedalfast_metadata, 6, 7

round_age, 7

yesno, 8