

Package ‘admiralophtha’

March 14, 2023

Type Package

Title ADaM in R Asset Library - Ophthalmology

Version 0.1.0

Description Aids the programming of Clinical Data Standards Interchange Consortium (CDISC) compliant Ophthalmology Analysis Data Model (ADaM) datasets in R. ADaM datasets are a mandatory part of any New Drug or Biologics License Application submitted to the United States Food and Drug Administration (FDA). Analysis derivations are implemented in accordance with the “Analysis Data Model Implementation Guide” (CDISC Analysis Data Model Team, 2021, <<https://www.cdisc.org/standards/foundational/adam/adamig-v1-3-release-package>>).

License Apache License (>= 2)

URL <https://pharmaverse.github.io/admiralophtha/>,
<https://github.com/pharmaverse/admiralophtha/>

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.2.1

Depends R (>= 3.5)

Imports admiral, admiraldev, assertthat, dplyr, lubridate, magrittr, purrr, rlang, stringr, hms, tidyr, tidyselect, lifecycle

Suggests admiral.test, devtools, diffdf, lintr, pkgdown, testthat, knitr, methods, miniUI, rmarkdown, roxygen2, spelling, styler, tibble, usethis, covr, DT

VignetteBuilder knitr

NeedsCompilation no

Author Edoardo Mancini [aut, cre],
Gordon Miller [aut],
Ritika Aggarwal [aut],
Jane Gao [aut],
William Holmes [aut],

Rachel Linacre [aut],
 Lucy Palmen [aut],
 Nandini R Thampi [aut]

Maintainer Edoardo Mancini <edoardo.mancini@roche.com>

Repository CRAN

Date/Publication 2023-03-14 19:40:02 UTC

R topics documented:

admiralophtha_adbcva	2
admiralophtha_adoe	3
admiralophtha_advfq	3
admiralophtha_ex	4
convert_etdrs_to_logmar	4
convert_logmar_to_etdrs	5
derive_var_afeye	6
derive_var_bcvacritxfl	7
derive_var_bcvacritxfl_util	9
derive_var_studyeye	10

Index	12
--------------	-----------

admiralophtha_adbcva *Best Corrected Visual Acuity Analysis Dataset*

Description

An example Best Corrected Visual Acuity (BCVA) analysis dataset

Usage

```
admiralophtha_adbcva
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 7672 rows and 115 columns.

Source

Derived from the `oe` and `ADSL` datasets using `{admiral, {admiralophtha}}` and (https://github.com/pharmaverse/admiralophtha/blob/main/inst/templates/ad_adbcva.R)

See Also

Other datasets: [admiralophtha_adoe](#), [admiralophtha_advfq](#), [admiralophtha_ex](#)

admiralophtha_adoe *Ophthalmology Exam Analysis Dataset*

Description

An example Ophthalmology Exam Analysis dataset

Usage

admiralophtha_adoe

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 7672 rows and 97 columns.

Source

Derived from the `oe` and `ADSL` datasets using `{admiral, {admiralophtha}}` and (https://github.com/pharmaverse/admiralophtha/blob/main/inst/templates/ad_adoe.R)

See Also

Other datasets: [admiralophtha_adbcva](#), [admiralophtha_advfq](#), [admiralophtha_ex](#)

admiralophtha_advfq *Visual Function Questionnaire Analysis Dataset*

Description

An example Visual Function Questionnaire (VFQ) analysis dataset

Usage

admiralophtha_advfq

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 28798 rows and 41 columns.

Source

Derived from the `ADSL` and `qs` datasets using `{admiral, {admiralophtha}}` and (https://github.com/pharmaverse/admiralophtha/blob/main/inst/templates/ad_advfq.R)

See Also

Other datasets: [admiralophtha_adbcva](#), [admiralophtha_adoe](#), [admiralophtha_ex](#)

admiralophtha_ex *Exposure Ophthalmology SDTM Dataset*

Description

An example Exposure SDTM dataset with ophthalmology-specific variables such as EXLOC and EXLAT

Usage

```
admiralophtha_ex
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 591 rows and 19 columns.

Source

Constructed using `ex` from the `{admiral}` package

See Also

Other datasets: [admiralophtha_adbcva](#), [admiralophtha_adoe](#), [admiralophtha_advfq](#)

convert_etdrs_to_logmar
ETDRS -> LogMAR conversion

Description

Convert ETDRS score to LogMAR units

Usage

```
convert_etdrs_to_logmar(value)
```

Arguments

value object containing ETDRS score to convert to logMAR

Details

ETDRS value converted to logMAR as $\text{logMAR} = -0.02 * \text{ETDRS} + 1.7$

Value

The input value converted converted to logMAR units

Author(s)

Rachel Linacre

Examples

```
library(tibble)
library(dplyr)
library(admiral)
library(admiraldev)

adbcva <- tribble(
  ~STUDYID, ~USUBJID, ~AVAL,
  "XXX001", "P01", 5,
  "XXX001", "P02", 10,
  "XXX001", "P03", 15,
  "XXX001", "P04", 20,
  "XXX001", "P05", 25
)

adbcva <- adbcva %>% mutate(AVAL = convert_etdrs_to_logmar(AVAL))
```

convert_logmar_to_etdrs

LogMAR -> ETDRS conversion

Description

Convert LogMAR score to ETDRS units

Usage

```
convert_logmar_to_etdrs(value)
```

Arguments

value object containing logMAR score to convert to ETDRS

Details

logMAR value converted to ETDRS as $ETDRS = -(\logMAR - 1.7) / 0.02$

Value

The input value converted to ETDRS units

Author(s)

Nandini R Thampi

Examples

```

library(tibble)
library(dplyr)
library(admiral)

oe <- tribble(
  ~STUDYID, ~USUBJID, ~OETESTCD, ~OEMETHOD, ~OESTRESN,
  "XXX001", "P01", "VACSCORE", "logMAR EYE CHART", 1.08,
  "XXX001", "P02", "VACSCORE", "logMAR EYE CHART", 1.66,
  "XXX001", "P03", "VACSCORE", "logMAR EYE CHART", 1.60,
  "XXX001", "P04", "VACSCORE", "ETDRS EYE CHART", 57,
  "XXX001", "P05", "VACSCORE", "ETDRS EYE CHART", 1
)

adbcva <- oe %>%
  filter(OETESTCD == "VACSCORE" & toupper(OEMETHOD) == "LOGMAR EYE CHART") %>%
  mutate(OESTRESN = convert_logmar_to_etdrs(OESTRESN))

```

derive_var_afeye	<i>Derive Affected Eye</i>
------------------	----------------------------

Description

Derive Affected Eye (AFEYE) in occurrence datasets

Usage

```
derive_var_afeye(dataset_occ, loc_var, lat_var)
```

Arguments

dataset_occ	Input Occurrence dataset
loc_var	Location variable
lat_var	Laterality variable

Details

Affected Eye is derived in the occurrence dataset using laterality and Study Eye. This assumes Study Eye has already been added from ADSL.

Value

The input occurrence dataset with Affected Eye (AFEYE) added.

Author(s)

Lucy Palmen

Examples

```
library(tibble)
library(admiral)

adae <- tribble(
  ~STUDYID, ~USUBJID, ~STUDYEYE, ~AELOC, ~AELAT,
  "XXX001", "P01", "RIGHT", "EYE", "RIGHT",
  "XXX001", "P01", "RIGHT", "EYE", "LEFT",
  "XXX001", "P02", "LEFT", "", "",
  "XXX001", "P02", "LEFT", "EYE", "LEFT",
  "XXX001", "P04", "BILATERAL", "EYE", "RIGHT",
  "XXX001", "P05", "RIGHT", "EYE", "RIGHT"
)

derive_var_afeye(adae, AELOC, AELAT)
```

```
derive_var_bcvacritxfl
```

Adds CRITx CRITxFL pairs to ADBCVA dataset

Description

Adds a criterion variables CRITx and their corresponding flags CRITxFL to an ADBCVA dataset.

Usage

```
derive_var_bcvacritxfl(
  dataset_adbcva,
  paramcds = NULL,
  basetype = NULL,
  bcva_ranges = NULL,
  bcva_uplims = NULL,
  bcva_lowlims = NULL,
  additional_text = "",
  critxfl_index = NULL
)
```

Arguments

`dataset_adbcva` Input dataset (ADBCVA).

`paramcds` Vector of PARAMCD values for which to derive CRITx and CRITxFL.

`basetype` BASETYPE value for which to derive CRITx and CRITxFL.

`bcva_ranges` List containing one or more numeric vectors of length 2. For each vector c(a,b) in `bcva_ranges`, a pair of variables CRITx, CRITxFL is created with the condition: $a \leq \text{CHG} \leq b$. If criterion flags of that type are not required, then leave as NULL.

`bcva_uplims` List containing one or more numeric elements. For each element `a` in `bcva_uplims`, a pair of variables `CRITx`, `CRITxFL` is created with the condition: `CHG <= a`. If criterion flags of that type are not required, then leave as `NULL`.

`bcva_lowlims` List containing one or more numeric elements. For each element `b` in `bcva_lowlims`, a pair of variables `CRITx`, `CRITxFL` is created with the condition: `CHG >= b`. If criterion flags of that type are not required, then leave as `NULL`.

`additional_text` string containing additional text to append to `CRITx`

`critxfl_index` positive integer detailing the first value of `x` to use in "`CRITxFL`". If not supplied, the function takes the first available value of `x`, counting up from `x = 1`.

Details

This function works by calling `derive_var_bcvacritxfl` once for each of the elements in `bcva_ranges`, `bcva_uplims` and `bcva_lowlims`. NOTE: if `CHG` is equal to `NA`, then the resulting criterion flag is also marked as `NA`.

Value

The input `ADBCVA` dataset with additional column pairs `CRITx`, `CRITxFL`.

Author(s)

Edoardo Mancini

Examples

```
library(tibble)
library(admiral)
library(admiraldev)

adbcva1 <- tribble(
  ~STUDYID, ~USUBJID, ~BASETYPE, ~PARAMCD, ~CHG,
  "XXX001", "P01", "LAST", "SBCVA", 0,
  "XXX001", "P01", "LAST", "FBCVA", 2,
  "XXX001", "P01", "LAST", "SBCVALOG", -7,
  "XXX001", "P02", "LAST", "SBCVA", -13,
  "XXX001", "P02", "LAST", "FBCVA", 5,
  "XXX001", "P02", "LAST", "SBCVALOG", 12,
  "XXX001", "P03", "LAST", "SBCVA", NA,
  "XXX001", "P03", "LAST", "FBCVA", 17
)

derive_var_bcvacritxfl(
  dataset_adbcva = adbcva1,
  paramcds = c("SBCVA", "FBCVA"),
  basetype = NULL,
  bcva_ranges = list(c(0, 5), c(-5, -1), c(10, 15)),
  bcva_uplims = list(5, 10),
  bcva_lowlims = list(8),
  additional_text = ""
)
```



```

)

adbcva2 <- tribble(
  ~STUDYID, ~USUBJID, ~AVISIT, ~BASETYPE, ~PARAMCD, ~AVAL, ~CHG,
  "XXX001", "P01", "BASELINE", "LAST", "SBCVA", 4, NA,
  "XXX001", "P01", "BASELINE", "LAST", "SBCVA", 6, NA,
  "XXX001", "P01", "AVERAGE BASELINE", "AVERAGE", "SBCVA", 5, NA,
  "XXX001", "P01", "WEEK 2", "LAST", "SBCVA", -3, NA,
  "XXX001", "P01", "WEEK 4", "LAST", "SBCVA", -10, NA,
  "XXX001", "P01", "WEEK 6", "LAST", "SBCVA", 12, NA,
  "XXX001", "P01", "WEEK 2", "AVERAGE", "SBCVA", -2, -7,
  "XXX001", "P01", "WEEK 4", "AVERAGE", "SBCVA", 6, 1,
  "XXX001", "P01", "WEEK 6", "AVERAGE", "SBCVA", 3, -2
)

derive_var_bcvacritxfl(
  dataset_adbcva = adbcva2,
  paramcds = c("SBCVA", "FBCVA"),
  basetype = "AVERAGE",
  bcva_ranges = list(c(0, 5), c(-10, 0)),
  bcva_lowlims = list(5),
  additional_text = " (AVERAGE)"
)

```

```
derive_var_bcvacritxfl_util
```

Add CRITx and CRITxFL pair to ADBCVA dataset

Description

Helper function for `derive_var_bcvacritxfl` that adds a criterion variable CRITx and its corresponding flag CRITxFL to an ADBCVA dataset.

Usage

```

derive_var_bcvacritxfl_util(
  dataset,
  critx_text,
  critxfl_cond,
  counter,
  bcva_range = NULL,
  bcva_uplim = NULL,
  bcva_lowlim = NULL
)

```

Arguments

dataset	Input dataset (ADBCVA).
critx_text	String containing the text for CRITx variable.
critxfl_cond	String containing R code detailing the criterion to be satisfied for CRITxFL variable to be equal to "Y".
counter	Integer detailing the value of x to use in "CRITxFL".
bcva_range	Numeric vector of length two detailing lower and upper change in BCVA limits (bcva_range will be called in critxfl_cond if the criterion stipulates that change in BCVA lie inside some range).
bcva_uplim	Numeric value detailing highest change in BCVA limit (bcva_uplim will be called in critxfl_cond if the criterion stipulates that change in BCVA lie below some upper limit).
bcva_lowlim	Numeric value detailing lowest change in BCVA limit (bcva_lowlim will be called in critxfl_cond if the criterion stipulates that change in BCVA lie above some lower limit).

Details

The criterion for change in BCVA in CRITxFL can be of three types: (1) value lies within some range; $a \leq \text{CHG} \leq b$; (2) value is below some upper limit; $\text{CHG} \leq a$; (3) value is above some lower limit; $b \leq \text{CHG}$. For (1), bcva_range must be specified to this function; for (2), bcva_uplim; for (3) bcva_lowlim. It is only ever necessary to supply one of these three arguments. NOTE: if CHG is equal to NA, then the resulting criterion flag is also marked as NA.

Value

The input ADBCVA dataset with additional columns CRITx, CRITxFL.

Author(s)

Edoardo Mancini

derive_var_studyeye *Derive Study Eye*

Description

Derive Study Eye (STUDYEYE) in the ADSL dataset

Usage

```
derive_var_studyeye(dataset_adsl, dataset_sc, sctestcd_value = "FOCID")
```

Arguments

dataset_adsl ADSL input dataset
dataset_sc SC input dataset
sctestcd_value SCTESTCD value flagging Study Eye selection records. Default: "FOCID".

Details

Study Eye is derived in ADSL using the "Study Eye selection" records in the SC SDTM dataset.

Value

The input ADSL dataset with an additional column named STUDYEYE

Author(s)

Edoardo Mancini

Examples

```
library(tibble)
library(admiral)

adsl <- tribble(
  ~STUDYID, ~USUBJID,
  "XXX001", "P01",
  "XXX001", "P02",
  "XXX001", "P03",
  "XXX001", "P04",
  "XXX001", "P05"
)

sc <- tribble(
  ~STUDYID, ~USUBJID, ~SCTESTCD, ~SCSTRESC,
  "XXX001", "P01", "FOCID", "OS",
  "XXX001", "P01", "ACOHORT", "COHORT1",
  "XXX001", "P02", "FOCID", "OD",
  "XXX001", "P02", "ACOHORT", "COHORT3",
  "XXX001", "P04", "FOCID", "OU",
  "XXX001", "P05", "FOCID", "OD",
  "XXX001", "P06", "FOCID", "OS"
)

derive_var_studyeye(adsl, sc)
```

Index

* datasets

- admiralophtha_adbcva, [2](#)
- admiralophtha_adoe, [3](#)
- admiralophtha_advfq, [3](#)
- admiralophtha_ex, [4](#)

* der_adsl

- derive_var_studyeye, [10](#)

* der_occds

- derive_var_afeye, [6](#)

* der_ophtha

- derive_var_bcvacritxfl, [7](#)
- derive_var_bcvacritxfl_util, [9](#)

* utils_fmt

- convert_etdrs_to_logmar, [4](#)
- convert_logmar_to_etdrs, [5](#)

[admiralophtha_adbcva, 2, 3, 4](#)
[admiralophtha_adoe, 2, 3, 3, 4](#)
[admiralophtha_advfq, 2, 3, 3, 4](#)
[admiralophtha_ex, 2, 3, 4](#)

[convert_etdrs_to_logmar, 4](#)
[convert_logmar_to_etdrs, 5](#)

[derive_var_afeye, 6](#)
[derive_var_bcvacritxfl, 7](#)
[derive_var_bcvacritxfl_util, 9](#)
[derive_var_studyeye, 10](#)