Package 'admiralophtha'

January 16, 2026

Type Package

Title ADaM in R Asset Library - Ophthalmology

Version 1.4.0

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/

BugReports https://github.com/pharmaverse/admiralophtha/issues/

Depends R (>= 4.1)

Imports admiral (>= 1.4.0), admiraldev (>= 1.4.0), dplyr (>= 1.1.1), hms (>= 0.5.3), lifecycle (>= 0.1.0), lubridate (>= 1.7.4), magrittr (>= 1.5), purrr (>= 0.3.3), rlang (>= 0.4.4), stringr (>= 1.4.0), tidyr (>= 1.0.2), tidyselect (>= 1.1.0)

Suggests devtools, diffdf, knitr, lintr, methods, miniUI, pharmaversesdtm (>= 1.3.0), pkgdown, rmarkdown, roxygen2, spelling, testthat (>= 3.0.0), tibble, usethis, withr

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.3

NeedsCompilation no

Author Edoardo Mancini [aut, cre] (ORCID:
<https://orcid.org/0009-0006-4899-8641>),
Ritika Aggarwal [aut],
Jane Gao [aut],
William Holmes [aut],
Josie Jackson [aut],
Sonali Jain [aut],
Yuki Matsunaga [aut],
Gordon Miller [aut],
Rachel Linacre [aut],
Lucy Palmen [aut],
Nandini R Thampi [aut],
Aldrich Salva [aut],
Steven Ting [aut],
F. Hoffmann-La Roche AG [cph, fnd],
Novartis AG [cph, fnd]

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

Repository CRAN

Date/Publication 2026-01-16 10:20:02 UTC

Contents

`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 116 columns.

Source

Derived from the OE and ADSL datasets using `{admiral}`, `{admiralophtha}` and the ADBCVA template.

See Also

Other datasets: `admiralophtha_adoe`, `admiralophtha_advfq`

`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 19136 rows and 103 columns.

Source

Derived from the OE and ADSL datasets using `{admiral}`, `{admiralophtha}` and the ADOE template.

See Also

Other datasets: `admiralophtha_adbcva`, `admiralophtha_advfq`

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 972 rows and 93 columns.

Source

Derived from the ADSL and QS datasets using `{admiral}`, `{admiralophtha}` and the ADVFQ template. The full, open-source VFQ questionnaire can be accessed here.

See Also

Other datasets: `admiralophtha_adbcva`, `admiralophtha_adoe`

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. **Permitted values** a numeric value, e.g. `2`, `-5`, `1.4` **Default value** none

Details

ETDRS value converted to logMAR as:

$$logMAR = -0.02 * ETDRS + 1.7$$

Source for conversion formula: Beck, R.W., et al. A computerized method of visual acuity testing. American Journal of Ophthalmology, 135(2), pp.194-205. doi:https://doi.org/10.1016/s0002-9394(02)01825-1.

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. **Permitted values** a numeric value, e.g. 2, -5, 1.4 **Default value** none

Details

logMAR value converted to ETDRS as:

$$ETDRS = -(logMAR - 1.7)/0.02$$

Source for conversion formula: Beck, R.W., et al. A computerized method of visual acuity testing. American Journal of Ophthalmology, 135(2), pp.194-205. doi:https://doi.org/10.1016/s0002-9394(02)01825-1.

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 *Derive Affected Eye*

Description

Derive Affected Eye (AFEYE) in occurrence datasets

Usage

```
derive_var_afeye(dataset, loc_var, lat_var, loc_vals = "EYE")
```

Arguments

dataset	Input dataset. **Permitted values** a dataset, i.e., a data.frame or tibble **Default value** none
loc_var	Location variable, usually XXLOC. **Permitted values** an unquoted symbol, e.g., AVAL **Default value** none
lat_var	Laterality variable, usually XXLAT. **Permitted values** an unquoted symbol, e.g., AVAL **Default value** none
loc_vals	xxLOC values for which AFEYE is derived. **Permitted values** a character vector, e.g. c("EYE", "RETINA") **Default value** "EYE"

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)

adae1 <- tribble(
  ~STUDYID, ~USUBJID, ~STUDYEYE, ~AELOC, ~AELAT,
  "XXX001", "P01", "RIGHT", "EYE", "RIGHT",
  "XXX001", "P01", "RIGHT", "EYE", "LEFT",
  "XXX001", "P01", "RIGHT", "EYE", "",
  "XXX001", "P01", "RIGHT", "", "RIGHT",
  "XXX001", "P02", "LEFT", "", "",
  "XXX001", "P02", "LEFT", "EYE", "LEFT",
  "XXX001", "P04", "BILATERAL", "EYE", "RIGHT",
  "XXX001", "P05", "RIGHT", "EYE", "RIGHT",
  "XXX001", "P05", "RIGHT", "EYE", "BILATERAL",
  "XXX001", "P06", "BILATERAL", "", "",
  "XXX001", "P06", "BILATERAL", "", "RIGHT",
  "XXX001", "P07", "BILATERAL", "EYE", "BILATERAL",
  "XXX001", "P08", "", "EYE", "BILATERAL",
  "XXX001", "P09", "NONSENSE", "EYE", "BILATERAL",
  "XXX001", "P09", "BILATERAL", "EYE", "NONSENSE",
  "XXX001", "P09", "BILATERAL", "NONSENSE", "BILATERAL",
  "XXX001", "P10", "RIGHT", "EYE", "BOTH"
)

derive_var_afeye(adae1, loc_var = AELOC, lat_var = AELAT)

adae2 <- tribble(
  ~STUDYID, ~USUBJID, ~STUDYEYE, ~AELOC, ~AELAT,
  "XXX001", "P01", "RIGHT", "EYES", "RIGHT",
  "XXX001", "P02", "RIGHT", "RETINA", "LEFT",
  "XXX001", "P03", "LEFT", "", ""
)

derive_var_afeye(adae2, loc_var = AELOC, lat_var = AELAT, loc_vals = c("EYES", "RETINA"))
```

---

derive_var_bcvacritxfl

Adds CRITx/CRITxFL *pairs to BCVA dataset*

Description

[Deprecated] The `derive_var_bcvacritxfl()` function has been deprecated in favor of `admiral::derive_vars_crit_fl` - please see the criterion flag section of the ADBCVA vignette for more details.

Adds a criterion variables `CRITx` and their corresponding flags `CRITxFL` to a dataset containing BCVA records

Usage

```
derive_var_bcvacritxfl(
  dataset,
  crit_var,
  bcva_ranges = NULL,
  bcva_uplims = NULL,
  bcva_lowlims = NULL,
  additional_text = "",
  critxfl_index = NULL
)
```

Arguments

dataset	Input dataset containing BCVA data (usually ADBCVA). **Permitted values** a dataset, i.e., a data.frame or tibble **Default value** none
crit_var	Variable with respect to which CRITx/CRITxFL are derived (usually CHG or AVAL). **Permitted values** an unquoted symbol, e.g., AVAL **Default value** none
bcva_ranges	List of numeric vectors. For each vector c(a,b) in bcva_ranges, a pair of variables CRITx, CRITxFL is created with the condition: a <= crit_var <= b. If criterion flags of that type are not required, then leave as NULL. **Permitted values** a list containing one or more numeric vectors, each of length two. E.g. list(c(1, 2), c(3, 4) **Default value** 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: crit_var <= a. If criterion flags of that type are not required, then leave as NULL. **Permitted values** a list containing one or more numeric scalars. E.g. list(2, -4) **Default value** 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: crit_var >= b. If criterion flags of that type are not required, then leave as NULL. **Permitted values** a list containing one or more numeric scalars. E.g. list(2, -4) **Default value** NULL
additional_text	string containing additional text to append to CRITx. **Permitted values** a character scalar, i.e., a character vector of length one **Default value** ""
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. **Permitted values** a positive integer, e.g. 2 or 5 **Default value** NULL

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 `crit_var` is equal to `NA`, then the resulting criterion flag is also marked as `NA`.

Value

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

Author(s)

Edoardo Mancini

See Also

Other deprecated: `derive_var_bcvacritxfl_util()`

Examples

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

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

derive_var_bcvacritxfl(
  dataset = adbcva1,
  crit_var = exprs(CHG),
  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
)

restrict_derivation(
  adbcva2,
  derivation = derive_var_bcvacritxfl,
  args = params(
    crit_var = exprs(CHG),
    bcva_ranges = list(c(0, 5), c(-10, 0)),
    bcva_lowlims = list(5),
    additional_text = " (AVERAGE)"
  ),
  filter = PARAMCD %in% c("SBCVA", "FBCVA") & BASETYPE == "AVERAGE"
)
```

`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. **Permitted values** a dataset, i.e., a `data.frame` or tibble **Default value** none
`dataset_sc`	SC input dataset. **Permitted values** a dataset, i.e., a `data.frame` or tibble **Default value** none
`sctestcd_value`	`SCTESTCD` value flagging Study Eye selection records. **Permitted values** a character scalar, i.e., a character vector of length one **Default value** `"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