

Package ‘admiralmetabolic’

July 22, 2025

Type Package

Title Metabolism Extension Package for ADaM in 'R' Asset Library

Version 0.2.0

Description A toolbox for programming Clinical Data Standards Interchange Consortium (CDISC) compliant 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>). The package is an extension package of the 'admiral' package focusing on the metabolism therapeutic area.

License Apache License (>= 2)

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

BugReports <https://github.com/pharmaverse/admiralmetabolic/issues>

Depends R (>= 4.1)

Imports admiral (>= 1.1.1), admiraldev (>= 1.2.0), cli (>= 3.6.2),
dplyr (>= 0.8.4), lifecycle (>= 0.1.0), lubridate (>= 1.7.4),
magrittr (>= 1.5), purrr (>= 0.3.3), rlang (>= 0.4.4), stringr
(>= 1.4.0), tidyselect (>= 1.0.0)

Suggests diffdf, DT, htmltools, knitr, methods, pharmaversesdtm (>= 1.3.0), rmarkdown, testthat (>= 3.0.0), tibble, usethis (>= 2.2.3)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Anders Askeland [aut, cre] (ORCID: <https://orcid.org/0000-0002-2996-129X>),
Andrii Yurovskyi [aut] (ORCID: <https://orcid.org/0009-0006-8606-0929>),
Kathrin Flunkert [aut],
Edoardo Mancini [aut] (ORCID: <https://orcid.org/0009-0006-4899-8641>),
Shunsuke Goto [aut],
Siddhesh Pujari [aut] (ORCID: <https://orcid.org/0009-0003-4971-1217>),
Sonali Das [aut],
Olga Starostecka [aut],
Vang Le-Quy [aut] (ORCID: <https://orcid.org/0000-0003-2035-8619>),
Keita Takahashi [aut] (ORCID: <https://orcid.org/0009-0004-9847-5115>)

Maintainer Anders Askeland <iakd@novonordisk.com>

Repository CRAN

Date/Publication 2025-07-16 11:10:02 UTC

Contents

admiralmetabolic_adsl	2
admiralmetabolic_advsn	3
derive_param_waisthtgt	3
derive_param_waisthip	7

Index	11
--------------	-----------

admiralmetabolic_adsl	<i>Subject Level Analysis Dataset</i>
-----------------------	---------------------------------------

Description

An example subject level analysis dataset

Usage

admiralmetabolic_adsl

Format

An object of class tbl_df (inherits from tbl, data.frame) with 306 rows and 54 columns.

Source

Derived from the ad_adsl_metabolic template

See Also

Other datasets: [admiralmetabolic_adv](#)

admiralmetabolic_adv *Vital sings Dataset*

Description

An example vital sings analysis dataset

Usage

```
admiralmetabolic_adv
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 760 rows and 101 columns.

Source

Derived from the `ad_adv` template

See Also

Other datasets: [admiralmetabolic_adsl](#)

`derive_param_waisthgt` *Adds a Parameter for Waist to Height Ratio*

Description

Adds a parameter for Waist to Height Ratio using Waist Circumference and Height for each by group (e.g., subject and visit) where the source parameters are available.

Note: This is a wrapper function for the more generic `admiral::derive_param_computed()`.

Usage

```
derive_param_waisthgt(  
  dataset,  
  by_vars,  
  wstcir_code = "WSTCIR",  
  height_code = "HEIGHT",  
  set_values_to = exprs(PARAMCD = "WAISTHGT"),  
  filter = NULL,  
  constant_by_vars = NULL,  
  get_unit_expr  
)
```

Arguments

dataset	Input dataset The variables specified by the <code>by_vars</code> argument are expected to be in the dataset. <code>PARAMCD</code>, and <code>AVAL</code> are expected as well. The variable specified by <code>by_vars</code> and <code>PARAMCD</code> must be a unique key of the input dataset after restricting it by the filter condition (<code>filter</code> argument) and to the parameters specified by <code>wstcir_code</code> and <code>height_code</code>.
by_vars	Grouping variables Grouping variables uniquely identifying a set of records for which <code>new_vars</code> are to be calculated. <i>Permitted Values:</i> list of variables created by <code>exprs()</code>
wstcir_code	Waist Circumference parameter code The observations where <code>PARAMCD</code> equals the specified value are considered as the Waist Circumference. <i>Permitted Values:</i> character value
height_code	Height parameter code The observations where <code>PARAMCD</code> equals the specified value are considered as the Height. <i>Permitted Values:</i> character value
set_values_to	Variables to be set The specified variables are set to the specified values for the new observations. For example <code>exprs(PARAMCD = "RATIO")</code> defines the parameter code for the new parameter. <i>Permitted Values:</i> List of variable-value pairs
filter	Filter condition The specified condition is applied to the input dataset before deriving the new parameter, i.e., only observations fulfilling the condition are taken into account. <i>Permitted Values:</i> a condition
constant_by_vars	By variables for when Height is constant When Height is constant, the Height parameters (measured only once) are merged to the other parameters using the specified variables. If Height is constant (e.g. only measured once at screening or baseline) then use <code>constant_by_vars</code> to select the subject-level variable to merge on (e.g. <code>USUBJID</code>). This will produce Waist to Height Ratio at all visits where Waist Circumference is measured. Otherwise it will only be calculated at visits with both Height and Waist Circumference collected. <i>Permitted Values:</i> list of variables created by <code>exprs()</code>, e.g. <code>exprs(USUBJID, VISIT)</code>
get_unit_expr	An expression providing the unit of the parameter The result is used to check the units of the input parameters. If the units are not consistent within each parameter, an error will be thrown. Additionally, if the input parameters are measured in different units but are mutually convertible (e.g., centimeters for one parameter and inches for another),

an automatic conversion will be performed in order to uniform the values before calculating the ratio.

Note: Conversion factors come from unit definitions as per CDISC standards. *m* is defined as 100 cm *mm* is defined as 0.1 cm *in* is defined as 2.54 cm *ft* is defined as 30.48 cm

Permitted Values: A variable of the input dataset or a function call

Details

The analysis value of the new parameter is derived as

$$WAISTHGT = \frac{WSTCIR}{HEIGHT}$$

Value

The input dataset with the new parameter added. Note, a variable will only be populated in the new parameter rows if it is specified in `by_vars`.

See Also

`admiral::derive_param_computed()`

ADVS Functions for adding Parameters: `derive_param_waisthip()`

Examples

```
library(tibble)
library(rlang)

# Example 1: Derive Waist to Height Ratio where Height is measured only once

advs <- tribble(
  ~USUBJID, ~PARAMCD, ~PARAM, ~AVAL, ~AVALU, ~VISIT,
  "01-101-1001", "HEIGHT", "Height (cm)", 147, "cm", "SCREENING",
  "01-101-1001", "WSTCIR", "Waist Circumference (cm)", 110, "cm", "SCREENING",
  "01-101-1001", "WSTCIR", "Waist Circumference (cm)", 108, "cm", "WEEK 2",
  "01-101-1001", "WSTCIR", "Waist Circumference (cm)", 107, "cm", "WEEK 3",
  "01-101-1002", "HEIGHT", "Height (cm)", 163, "cm", "SCREENING",
  "01-101-1002", "WSTCIR", "Waist Circumference (cm)", 120, "cm", "SCREENING",
  "01-101-1002", "WSTCIR", "Waist Circumference (cm)", 118, "cm", "WEEK 2",
  "01-101-1002", "WSTCIR", "Waist Circumference (cm)", 117, "cm", "WEEK 3",
)

derive_param_waisthgt(
  advs,
  by_vars = exprs(USUBJID, VISIT),
  wstcir_code = "WSTCIR",
  height_code = "HEIGHT",
  set_values_to = exprs(
    PARAMCD = "WAISTHGT",
    PARAM = "Waist to Height Ratio"
  ),
)
```

```

    constant_by_vars = exprs(USUBJID),
    get_unit_expr = admiral::extract_unit(PARAM)
  )

```

```

# Example 2: Same as above but only adding Waist to Height Ratio
# at certain visits

```

```

derive_param_waisthgt(
  advs,
  by_vars = exprs(USUBJID, VISIT),
  wstcir_code = "WSTCIR",
  height_code = "HEIGHT",
  set_values_to = exprs(
    PARAMCD = "WAISTHGT",
    PARAM = "Waist to Height Ratio"
  ),
  constant_by_vars = exprs(USUBJID),
  get_unit_expr = admiral::extract_unit(PARAM),
  filter = VISIT %in% c("SCREENING", "WEEK 3")
)

```

```

# Example 3: Pediatric study where Height and Waist Circumference
# are measured multiple times

```

```

advs <- tribble(
  ~USUBJID, ~PARAMCD, ~PARAM, ~AVAL, ~AVALU, ~VISIT,
  "01-101-1001", "HEIGHT", "Height (cm)", 147, "cm", "SCREENING",
  "01-101-1001", "HEIGHT", "Height (cm)", 148, "cm", "WEEK 2",
  "01-101-1001", "HEIGHT", "Height (cm)", 149, "cm", "WEEK 3",
  "01-101-1001", "WSTCIR", "Waist Circumference (cm)", 100, "cm", "SCREENING",
  "01-101-1001", "WSTCIR", "Waist Circumference (cm)", 99, "cm", "WEEK 2",
  "01-101-1001", "WSTCIR", "Waist Circumference (cm)", 98, "cm", "WEEK 3",
  "01-101-1002", "HEIGHT", "Height (cm)", 163, "cm", "SCREENING",
  "01-101-1002", "HEIGHT", "Height (cm)", 164, "cm", "WEEK 2",
  "01-101-1002", "HEIGHT", "Height (cm)", 165, "cm", "WEEK 3",
  "01-101-1002", "WSTCIR", "Waist Circumference (cm)", 110, "cm", "SCREENING",
  "01-101-1002", "WSTCIR", "Waist Circumference (cm)", 109, "cm", "WEEK 2",
  "01-101-1002", "WSTCIR", "Waist Circumference (cm)", 108, "cm", "WEEK 3"
)

```

```

derive_param_waisthgt(
  advs,
  by_vars = exprs(USUBJID, VISIT),
  wstcir_code = "WSTCIR",
  height_code = "HEIGHT",
  set_values_to = exprs(
    PARAMCD = "WAISTHGT",
    PARAM = "Waist to Height Ratio"
  ),
  get_unit_expr = admiral::extract_unit(PARAM)
)

```

```

# Example 4: Automatic conversion is performed when deriving the ratio

```

```
# if parameters are provided in different units (e.g. centimeters and inches)

advs <- tribble(
  ~USUBJID, ~PARAMCD, ~PARAM, ~AVAL, ~AVALU, ~VISIT,
  "01-101-1001", "HEIGHT", "Height (cm)", 147, "cm", "SCREENING",
  "01-101-1001", "WSTCIR", "Waist Circumference (in)", 39.37, "in", "SCREENING",
  "01-101-1001", "WSTCIR", "Waist Circumference (in)", 38.98, "in", "WEEK 2",
  "01-101-1001", "WSTCIR", "Waist Circumference (in)", 38.58, "in", "WEEK 3",
  "01-101-1002", "HEIGHT", "Height (cm)", 163, "cm", "SCREENING",
  "01-101-1002", "WSTCIR", "Waist Circumference (in)", 43.31, "in", "SCREENING",
  "01-101-1002", "WSTCIR", "Waist Circumference (in)", 42.91, "in", "WEEK 2",
  "01-101-1002", "WSTCIR", "Waist Circumference (in)", 42.52, "in", "WEEK 3"
)

derive_param_waisthgt(
  advs,
  by_vars = exprs(USUBJID, VISIT),
  wstcir_code = "WSTCIR",
  height_code = "HEIGHT",
  set_values_to = exprs(
    PARAMCD = "WAISTHGT",
    PARAM = "Waist to Height Ratio"
  ),
  constant_by_vars = exprs(USUBJID),
  get_unit_expr = admiral::extract_unit(PARAM)
)
```

derive_param_waisthip *Adds a Parameter for Waist to Hip Ratio*

Description

Adds a parameter for Waist to Hip Ratio using Waist Circumference and Hip Circumference for each by group (e.g., subject and visit) where the source parameters are available.

Note: This is a wrapper function for the more generic `admiral::derive_param_computed()`.

Usage

```
derive_param_waisthip(
  dataset,
  by_vars,
  wstcir_code = "WSTCIR",
  hipcir_code = "HIPCIR",
  set_values_to = exprs(PARAMCD = "WAISTHIP"),
  filter = NULL,
  get_unit_expr
)
```

Arguments

dataset	Input dataset The variables specified by the <code>by_vars</code> argument are expected to be in the dataset. <code>PARAMCD</code>, and <code>AVAL</code> are expected as well. The variable specified by <code>by_vars</code> and <code>PARAMCD</code> must be a unique key of the input dataset after restricting it by the filter condition (<code>filter</code> argument) and to the parameters specified by <code>wstcir_code</code> and <code>hipcir_code</code>.
by_vars	Grouping variables Grouping variables uniquely identifying a set of records for which <code>new_vars</code> are to be calculated. <i>Permitted Values:</i> list of variables created by <code>exprs()</code>
wstcir_code	Waist Circumference parameter code The observations where <code>PARAMCD</code> equals the specified value are considered as the Waist Circumference. <i>Permitted Values:</i> character value
hipcir_code	Hip Circumference parameter code The observations where <code>PARAMCD</code> equals the specified value are considered as the Hip Circumference <i>Permitted Values:</i> character value
set_values_to	Variables to be set The specified variables are set to the specified values for the new observations. For example <code>exprs(PARAMCD = "RATIO")</code> defines the parameter code for the new parameter. <i>Permitted Values:</i> List of variable-value pairs
filter	Filter condition The specified condition is applied to the input dataset before deriving the new parameter, i.e., only observations fulfilling the condition are taken into account. <i>Permitted Values:</i> a condition
get_unit_expr	An expression providing the unit of the parameter The result is used to check the units of the input parameters. If the units are not consistent within each parameter, an error will be thrown. Additionally, if the input parameters are measured in different units but are mutually convertible (e.g., centimeters for one parameter and inches for another), an automatic conversion will be performed in order to uniform the values before calculating the ratio. Note: Conversion factors come from unit definitions as per CDISC standards. <i>m</i> is defined as 100 cm <i>mm</i> is defined as 0.1 cm <i>in</i> is defined as 2.54 cm <i>ft</i> is defined as 30.48 cm <i>Permitted Values:</i> A variable of the input dataset or a function call

Details

The analysis value of the new parameter is derived as

$$WAISTHIP = \frac{WSTCIR}{HIPCIR}$$

Value

The input dataset with the new parameter added. Note, a variable will only be populated in the new parameter rows if it is specified in `by_vars`.

See Also

`admiral::derive_param_computed()`

ADVS Functions for adding Parameters: `derive_param_waisthgt()`

Examples

```
library(tibble)
library(rlang)

advs <- tribble(
  ~USUBJID, ~PARAMCD, ~PARAM, ~AVAL, ~AVALU, ~VISIT,
  "01-101-1001", "WSTCIR", "Waist Circumference (cm)", 110, "cm", "SCREENING",
  "01-101-1001", "WSTCIR", "Waist Circumference (cm)", 108, "cm", "WEEK 2",
  "01-101-1001", "WSTCIR", "Waist Circumference (cm)", 107, "cm", "WEEK 3",
  "01-101-1001", "HIPCIR", "Hip Circumference (cm)", 125, "cm", "SCREENING",
  "01-101-1001", "HIPCIR", "Hip Circumference (cm)", 124, "cm", "WEEK 2",
  "01-101-1001", "HIPCIR", "Hip Circumference (cm)", 123, "cm", "WEEK 3",
  "01-101-1002", "WSTCIR", "Waist Circumference (cm)", 120, "cm", "SCREENING",
  "01-101-1002", "WSTCIR", "Waist Circumference (cm)", 118, "cm", "WEEK 2",
  "01-101-1002", "WSTCIR", "Waist Circumference (cm)", 117, "cm", "WEEK 3",
  "01-101-1002", "HIPCIR", "Hip Circumference (cm)", 135, "cm", "SCREENING",
  "01-101-1002", "HIPCIR", "Hip Circumference (cm)", 133, "cm", "WEEK 2",
  "01-101-1002", "HIPCIR", "Hip Circumference (cm)", 132, "cm", "WEEK 3"
)

derive_param_waisthip(
  advs,
  by_vars = exprs(USUBJID, VISIT),
  wstcir_code = "WSTCIR",
  hipcir_code = "HIPCIR",
  set_values_to = exprs(
    PARAMCD = "WAISTHIP",
    PARAM = "Waist to Hip Ratio"
  ),
  get_unit_expr = admiral::extract_unit(PARAM)
)

# Only adding Waist to Hip Ratio at certain visits

derive_param_waisthip(
  advs,
  by_vars = exprs(USUBJID, VISIT),
  wstcir_code = "WSTCIR",
  hipcir_code = "HIPCIR",
  set_values_to = exprs(
    PARAMCD = "WAISTHIP",
```

```

    PARAM = "Waist to Hip Ratio"
  ),
  get_unit_expr = admiral::extract_unit(PARAM),
  filter = VISIT %in% c("SCREENING", "WEEK 3")
)

# Automatic conversion is performed when deriving the ratio
# if parameters are provided in different units

advs <- tribble(
  ~USUBJID, ~PARAMCD, ~PARAM, ~AVAL, ~AVALU, ~VISIT,
  "01-101-1001", "HIPCIR", "Hip Circumference (cm)", 125, "cm", "SCREENING",
  "01-101-1001", "HIPCIR", "Hip Circumference (cm)", 124, "cm", "WEEK 2",
  "01-101-1001", "HIPCIR", "Hip Circumference (cm)", 123, "cm", "WEEK 3",
  "01-101-1001", "WSTCIR", "Waist Circumference (in)", 43.31, "in", "SCREENING",
  "01-101-1001", "WSTCIR", "Waist Circumference (in)", 42.52, "in", "WEEK 2",
  "01-101-1001", "WSTCIR", "Waist Circumference (in)", 42.13, "in", "WEEK 3",
  "01-101-1002", "HIPCIR", "Hip Circumference (cm)", 135, "cm", "SCREENING",
  "01-101-1002", "HIPCIR", "Hip Circumference (cm)", 133, "cm", "WEEK 2",
  "01-101-1002", "HIPCIR", "Hip Circumference (cm)", 132, "cm", "WEEK 3",
  "01-101-1002", "WSTCIR", "Waist Circumference (in)", 47.24, "in", "SCREENING",
  "01-101-1002", "WSTCIR", "Waist Circumference (in)", 46.46, "in", "WEEK 2",
  "01-101-1002", "WSTCIR", "Waist Circumference (in)", 46.06, "in", "WEEK 3"
)

derive_param_waisthip(
  advs,
  by_vars = exprs(USUBJID, VISIT),
  wstcir_code = "WSTCIR",
  hipcir_code = "HIPCIR",
  set_values_to = exprs(
    PARAMCD = "WAISTHIP",
    PARAM = "Waist to Hip Ratio"
  ),
  get_unit_expr = admiral::extract_unit(PARAM)
)

```

Index

* **datasets**

admiralmetabolic_adsl, [2](#)

admiralmetabolic_adv, [3](#)

* **der_prm_adv**

derive_param_waisthtgt, [3](#)

derive_param_waisthip, [7](#)

admiral::derive_param_computed(), [3](#), [5](#),
[7](#), [9](#)

admiralmetabolic_adsl, [2](#), [3](#)

admiralmetabolic_adv, [3](#), [3](#)

derive_param_waisthtgt, [3](#), [9](#)

derive_param_waisthip, [5](#), [7](#)