

# Package: datacutr (via r-universe)

November 21, 2024

**Type** Package

**Title** SDTM Datacut

**Version** 0.2.0

**Description** Supports the process of applying a cut to Standard Data Tabulation Model (SDTM), as part of the analysis of specific points in time of the data, normally as part of investigation into clinical trials. The functions support different approaches of cutting to the different domains of SDTM normally observed.

**License** Apache License (>= 2)

**BugReports** <https://github.com/pharmaverse/datacutr/issues>

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

**Encoding** UTF-8

**Language** en-US

**LazyData** true

**RoxygenNote** 7.3.2

**Depends** R (>= 4.1)

**Imports** admiraldev (>= 0.3.0), assertthat (>= 0.2.1), dplyr (>= 1.0.5), lubridate (>= 1.7.4), magrittr (>= 1.5), purrr (>= 0.3.3), stringr (>= 1.4.0), rlang (>= 0.4.4), tibble (>= 3.0.0), reactable (>= 0.4.4)

**Suggests** devtools, lintr, pkgdown, testthat, knitr, methods, rmarkdown, roxygen2, spelling, styler, usethis, covr

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Tim Barnett [cph, aut, cre], Nathan Rees [aut], Alana Harris [aut], Cara Andrews [aut]

**Maintainer** Tim Barnett <[timothy.barnett@roche.com](mailto:timothy.barnett@roche.com)>

**Date/Publication** 2024-11-19 16:40:01 UTC

**Additional\_repositories** <https://cranhaven.r-universe.dev>

**Config/pak/sysreqs** make libicu-dev

**Repository** <https://cranhaven.r-universe.dev>

**RemoteUrl** <https://github.com/cranhaven/cranhaven.r-universe.dev>

**RemoteRef** package/datacutr

**RemoteSha** 26e13f5551764082ec4bce157d9324cf966ff768

## Contents

apply_cut . . . . .	2
create_dcut . . . . .	3
datacutr_ae . . . . .	4
datacutr_dm . . . . .	5
datacutr_ds . . . . .	5
datacutr_fa . . . . .	6
datacutr_lb . . . . .	6
datacutr_sc . . . . .	7
datacutr_ts . . . . .	7
date_cut . . . . .	8
drop_temp_vars . . . . .	9
impute_dcutdte . . . . .	10
impute_sdtm . . . . .	11
process_cut . . . . .	11
pt_cut . . . . .	13
read_out . . . . .	14
special_dm_cut . . . . .	16
<b>Index</b>	<b>18</b>

---

apply\_cut

*Applies the datacut based on the datacut flagging variables*

---

### Description

Removes any records where the datacut flagging variable, usually called DCUT\_TEMP\_REMOVE, is marked as "Y". Also, sets the death related variables in DM (DTHDTC and DTHFL) to missing if the death after datacut flagging variable, usually called DCUT\_TEMP\_DTHCHANGE, is marked as "Y".

### Usage

```
apply_cut(dsin, dcutvar, dthchangevar)
```

**Arguments**

dsin	Name of input dataframe
dcutvar	Name of datacut flagging variable created by pt_cut and date_cut functions - usually called DCUT_TEMP_REMOVE.
dthchangevar	Name of death after datacut flagging variable created by special_dm_cut function - usually called DCUT_TEMP_DTHCHANGE.

**Value**

Returns the input dataframe, excluding any rows in which dcutvar is flagged as "Y". DTHDTC and DTHFL are set to missing for any records where dthchangevar is flagged as "Y". Any variables with the "DCUT\_TEMP" prefix are removed.

**Examples**

```
ae <- data.frame(
  USUBJID = c("UXYZ123a", "UXYZ123b", "UXYZ123c", "UXYZ123d"),
  DCUT_TEMP_REMOVE = c("Y", "", "NA", NA)
)
ae_final <- apply_cut(dsin = ae, dcutvar = DCUT_TEMP_REMOVE, dthchangevar = DCUT_TEMP_DTHCHANGE)

dm <- data.frame(
  USUBJID = c("UXYZ123a", "UXYZ123b", "UXYZ123b"),
  DTHDTC = c("2014-10-20", "2014-10-21", "2013-09-08"),
  DTHFL = c("Y", "Y", "Y"),
  DCUT_TEMP_REMOVE = c(NA, NA, "Y"),
  DCUT_TEMP_DTHCHANGE = c(NA, "Y", "")
)
dm_final <- apply_cut(dsin = dm, dcutvar = DCUT_TEMP_REMOVE, dthchangevar = DCUT_TEMP_DTHCHANGE)
```

---

create\_dcut

*Create Datacut Dataset (DCUT)*


---

**Description**

After filtering the input DS dataset (based on the given filter condition), any records where the SDTMv date/time variable is on or before the datacut date/time (after imputations) will be returned in the output datacut dataset (DCUT). Note that ds\_date\_var and cut\_date inputs must be in ISO 8601 format (YYYY-MM-DDThh:mm:ss) and will be imputed using the impute\_sdtm() and impute\_dcutdtc() functions.

**Usage**

```
create_dcut(dataset_ds, ds_date_var, filter, cut_date, cut_description)
```

**Arguments**

dataset_ds	Input DS SDTMv dataset
ds_date_var	Character date/time variable in the DS SDTMv to be compared against the datacut date
filter	Condition to filter patients in DS, should give 1 row per patient
cut_date	Datacut date/time, e.g. "2022-10-22", or NA if no date cut is to be applied
cut_description	Datacut date/time description, e.g. "Clinical Cut Off Date"

**Value**

Datacut dataset containing the variables USUBJID, DCUTDTC, DCUTDTM and DCUTDESC.

**Author(s)**

Alana Harris

**Examples**

```
ds <- tibble::tribble(
  ~USUBJID, ~DSSEQ, ~DSDECOD, ~DSSTDTC,
  "subject1", 1, "INFORMED CONSENT", "2020-06-23",
  "subject1", 2, "RANDOMIZATION", "2020-08-22",
  "subject1", 3, "WITHDRAWAL BY SUBJECT", "2020-05-01",
  "subject2", 1, "INFORMED CONSENT", "2020-07-13",
  "subject3", 1, "INFORMED CONSENT", "2020-06-03",
  "subject4", 1, "INFORMED CONSENT", "2021-01-01",
  "subject4", 2, "RANDOMIZATION", "2023-01-01"
)

dcut <- create_dcut(
  dataset_ds = ds,
  ds_date_var = DSSTDTC,
  filter = DSDECOD == "RANDOMIZATION",
  cut_date = "2022-01-01",
  cut_description = "Clinical Cutoff Date"
)
```

---

datacutr\_ae

*Adverse Events SDTMv Dataset*

---

**Description**

An example Adverse Events (AE) SDTMv domain.

**Usage**

datacutr\_ae

**Format**

A dataset with 5 rows and 3 variables:

**USUBJID** Unique Subject Identifier

**AETERM** Reported Term for the Adverse Event

**AESTDTC** Start Date/Time of Adverse Event

---

datacutr_dm	<i>Demographics SDTMv Dataset</i>
-------------	-----------------------------------

---

**Description**

An example Demographics (DM) SDTMv domain.

**Usage**

datacutr\_dm

**Format**

A dataset with 5 rows and 3 variables:

**USUBJID** Unique Subject Identifier

**DTHFL** Subject Death Flag

**DTHDTC** Date/Time of Death

---

datacutr_ds	<i>Disposition SDTMv Dataset</i>
-------------	----------------------------------

---

**Description**

An example Disposition (DS) SDTMv domain.

**Usage**

datacutr\_ds

**Format**

A dataset with 5 rows and 3 variables:

**USUBJID** Unique Subject Identifier

**DSDECOD** Standardized Disposition Term

**DSSTDTC** Start Date/Time of Disposition Event

---

datacutr\_fa

*Findings About Events or Interventions SDTMv Dataset*

---

**Description**

An example Findings About Events or Interventions (FA) SDTMv domain.

**Usage**

datacutr\_fa

**Format**

A dataset with 5 rows and 4 variables:

**USUBJID** Unique Subject Identifier

**FAORRES** Result or Finding in Original Units

**FADTC** Date/Time of Collection

**FASTDTC** Start Date/Time of Observation

---

datacutr\_lb

*Laboratory Test Results SDTMv Dataset*

---

**Description**

An example Laboratory Test Results (LB) SDTMv domain.

**Usage**

datacutr\_lb

**Format**

A dataset with 5 rows and 3 variables:

**USUBJID** Unique Subject Identifier

**LBORRES** Result or Finding in Original Units

**LBDTC** Date/Time of Specimen Collection

---

datacutr_sc	<i>Subject Characteristics SDTMv Dataset</i>
-------------	--

---

**Description**

An example Subject Characteristics (SC) SDTMv domain.

**Usage**

datacutr\_sc

**Format**

A dataset with 5 rows and 2 variables:

**USUBJID** Unique Subject Identifier

**SCORRES** Result or Finding in Original Units

---

datacutr_ts	<i>Trial Summary SDTMv Dataset</i>
-------------	------------------------------------

---

**Description**

An example Trial Summary (TS) SDTMv domain.

**Usage**

datacutr\_ts

**Format**

A dataset with 5 rows and 2 variables:

**USUBJID** Unique Subject Identifier

**TSVAL** Parameter Value

---

date_cut	<i>xxSTDTC or xxDTC Cut</i>
----------	-----------------------------

---

### Description

Use to apply a datacut to either an xxSTDTC or xxDTC SDTM date variable. The datacut date from the datacut dataset is merged on to the input SDTMv dataset and renamed to TEMP\_DCUT\_DCUTDTM. A flag TEMP\_DCUT\_REMOVE is added to the dataset to indicate the observations that would be removed when the cut is applied. Note that this function applies a patient level datacut at the same time (using the pt\_cut() function), and also imputes dates in the specified SDTMv dataset (using the impute\_sdtm() function).

### Usage

```
date_cut(dataset_sdtm, sdtm_date_var, dataset_cut, cut_var)
```

### Arguments

dataset_sdtm	Input SDTMv dataset
sdtm_date_var	Input date variable found in the dataset_sdtmv dataset
dataset_cut	Input datacut dataset
cut_var	Datacut date variable

### Value

Input dataset plus a flag TEMP\_DCUT\_REMOVE to indicate which observations would be dropped when a datacut is applied

### Author(s)

Alana Harris

### Examples

```
library(lubridate)
dcut <- tibble::tribble(
  ~USUBJID, ~DCUTDTM, ~DCUTDTC,
  "subject1", ymd_hms("2020-10-11T23:59:59"), "2020-10-11T23:59:59",
  "subject2", ymd_hms("2020-10-11T23:59:59"), "2020-10-11T23:59:59",
  "subject4", ymd_hms("2020-10-11T23:59:59"), "2020-10-11T23:59:59"
)

ae <- tibble::tribble(
  ~USUBJID, ~AESEQ, ~AESTDTC,
  "subject1", 1, "2020-01-02T00:00:00",
  "subject1", 2, "2020-08-31T00:00:00",
  "subject1", 3, "2020-10-10T00:00:00",
  "subject2", 2, "2020-02-20T00:00:00",
```



```

    "subject3", 1, "2020-03-02T00:00:00",
    "subject4", 1, "2020-11-02T00:00:00",
    "subject4", 2, ""
  )

  ae_out <- date_cut(
    dataset_sdtm = ae,
    sdtm_date_var = AESTDTC,
    dataset_cut = dcut,
    cut_var = DCUTDTM
  )

```

---

 drop\_temp\_vars

*Drops Temporary Variables From a Dataset*


---

### Description

Drops all the temporary variables (variables beginning with TEMP\_) from the input dataset. Also allows the user to specify whether or not to drop the temporary variables needed throughout multiple steps of the datacut process (variables beginning with DCUT\_TEMP\_).

### Usage

```
drop_temp_vars(dsin, drop_dcut_temp = TRUE)
```

### Arguments

`dsin` Name of input dataframe  
`drop_dcut_temp` Whether or not to drop variables beginning with DCUT\_TEMP\_ (TRUE/FALSE).

### Details

The other functions within this package use `drop_temp_vars` with the `drop_dcut_temp` argument set to FALSE so that the variables needed across multiple steps of the process are kept. The final datacut takes place in the `apply_cut` function, at which point `drop_temp_vars` is used with the `drop_dcut_temp` argument set to TRUE, so that all temporary variables are dropped.

### Value

Returns the input dataframe, excluding the temporary variables.

### Examples

```

ae <- tibble::tribble(
  ~USUBJID, ~AESEQ, ~TEMP_FLAG, ~DCUT_TEMP_REMOVE,
  "subject1", 1, "Y", NA,
  "subject1", 2, "Y", NA,
  "subject1", 3, NA, "Y",
  "subject2", 2, "Y", NA,

```

```

    "subject3", 1, NA, "Y",
    "subject4", 1, NA, "Y"
  )
drop_temp_vars(dsin = ae) # Drops temp_ and dcut_temp_ variables
drop_temp_vars(dsin = ae, drop_dcut_temp = TRUE) # Drops temp_ and dcut_temp_ variables
drop_temp_vars(dsin = ae, drop_dcut_temp = FALSE) # Drops temp_ variables

```

---

impute\_dcutdtc

*Imputes Partial Date/Time Data Cutoff Variable (DCUTDTC)*


---

### Description

Imputes partial date/time data cutoff variable (DCUTDTC), as required by the datacut process.

### Usage

```
impute_dcutdtc(dsin, varin, varout)
```

### Arguments

dsin	Name of input data cut dataframe (i.e; DCUT)
varin	Name of input data cutoff variable (i.e; DCUTDTC) which must be in ISO 8601 extended format (YYYY-MM-DDThh:mm:ss). All values of the data cutoff variable must be at least a complete date, or NA.
varout	Name of imputed output variable

### Value

Returns the input data cut dataframe, with the additional of one extra variable (varout) in POSIXct datetime format, which is the imputed version of varin.

### Examples

```

dcut <- data.frame(
  USUBJID = rep(c("UXYZ123a"), 7),
  DCUTDTC = c(
    "2022-06-23", "2022-06-23T16", "2022-06-23T16:57", "2022-06-23T16:57:30",
    "2022-06-23T16:57:30.123", "2022-06-23T16-:30", "2022-06-23T-:57:30"
  )
)
dcut_final <- impute_dcutdtc(dsin = dcut, varin = DCUTDTC, varout = DCUTDTM)

```

---

impute_sdtm	<i>Imputes Partial Date/Time SDTMv Variables</i>
-------------	--

---

**Description**

Imputes partial date/time SDTMv variables, as required by the datacut process.

**Usage**

```
impute_sdtm(dsin, varin, varout)
```

**Arguments**

dsin	Name of input SDTMv dataframe
varin	Name of input SDTMv character date/time variable, which must be in ISO 8601 extended format (YYYY-MM-DDThh:mm:ss). The use of date/time intervals are not permitted.
varout	Name of imputed output variable

**Value**

Returns the input SDTMv dataframe, with the addition of one extra variable (varout) in POSIXct datetime format, which is the imputed version of varin.

**Examples**

```
ex <- data.frame(
  USUBJID = rep(c("UXYZ123a"), 13),
  EXSTDTC = c(
    "", "2022", "2022-06", "2022-06-23", "2022-06-23T16", "2022-06-23T16:57",
    "2022-06-23T16:57:30", "2022-06-23T16:57:30.123", "2022-06-23T16:-:30",
    "2022-06-23T-:57:30", "2022-06--T16:57:30", "2022---23T16:57:30", "--06-23T16:57:30"
  )
)
ex_imputed <- impute_sdtm(dsin = ex, varin = EXSTDTC, varout = DCUT_TEMP_EXSTDTC)
```

---

process_cut	<i>Wrapper function to prepare and apply the datacut of SDTMv datasets</i>
-------------	--

---

**Description**

Applies the selected type of datacut on each SDTMv dataset based on the chosen SDTMv date variable, and outputs the resulting cut datasets, as well as the datacut dataset, as a list. It provides an option to perform a "special" cut on the demography (dm) domain in which any deaths occurring after the datacut date are removed. It also provides an option to produce a .html file that summarizes the changes applied to the data during the cut, where you can inspect the records that have been removed and/or modified.

**Usage**

```
process_cut(
  source_sdtm_data,
  patient_cut_v = NULL,
  date_cut_m = NULL,
  no_cut_v = NULL,
  dataset_cut,
  cut_var,
  special_dm = TRUE,
  read_out = FALSE,
  out_path = "."
)
```

**Arguments**

source_sdtm_data	A list of uncut SDTMv dataframes
patient_cut_v	A vector of quoted SDTMv domain names in which a patient cut should be applied. To be left blank if a patient cut should not be performed on any domains.
date_cut_m	A 2 column matrix, where the first column is the quoted SDTMv domain names in which a date cut should be applied and the second column is the quoted SDTMv date variables used to carry out the date cut for each SDTMv domain. To be left blank if a date cut should not be performed on any domains.
no_cut_v	A vector of quoted SDTMv domain names in which no cut should be applied. To be left blank if no domains are to remain exactly as source.
dataset_cut	Input datacut dataset, e.g. dcut
cut_var	Datacut date variable within the dataset_cut dataset, e.g. DCUTDTM
special_dm	A logical input indicating whether the special dm cut should be performed. Note that, if TRUE, dm should not be included in patient_cut_v, date_cut_m or no_cut_v inputs.
read_out	A logical input indicating whether a summary file for the datacut should be produced. If TRUE, a .html file will be returned containing a summary of the cut and records removed. Default set to FALSE.
out_path	A character vector of file save path for the summary file if read_out = TRUE; the default corresponds to the working directory, getwd().

**Value**

Returns a list of all input SDTMv datasets, plus the datacut dataset, after performing the selected datacut on each SDTMv domain.

**Examples**

```
dcut <- data.frame(
  USUBJID = c("a", "b"),
  DCUTDTC = c("2022-02-17", "2022-02-17")
)
```

```

dcut <- impute_dcutdte(dcut, DCUTDTC, DCUTDTM)
sc <- data.frame(USUBJID = c("a", "a", "b", "c"))
ts <- data.frame(USUBJID = c("a", "a", "b", "c"))
ae <- data.frame(
  USUBJID = c("a", "a", "b", "c"),
  AESTDTC = c("2022-02-16", "2022-02-18", "2022-02-16", "2022-02-16")
)
source_data <- list(sc = sc, ae = ae, ts = ts)

cut_data <- process_cut(
  source_sdtm_data = source_data,
  patient_cut_v = c("sc"),
  date_cut_m = rbind(c("ae", "AESTDTC")),
  no_cut_v = c("ts"),
  dataset_cut = dcut,
  cut_var = DCUTDTM,
  special_dm = FALSE
)

```

---

pt\_cut

*Patient Cut*


---

### Description

Use to apply a patient cut to an SDTMv dataset (i.e. subset SDTMv observations on patients included in the dataset\_cut input dataset)

### Usage

```
pt_cut(dataset_sdtm, dataset_cut)
```

### Arguments

dataset_sdtm	Input SDTMv dataset
dataset_cut	Input datacut dataset, e.g. dcut

### Value

Input dataset plus a flag DCUT\_TEMP\_REMOVE to indicate which observations would be dropped when a patient level datacut is applied

### Author(s)

Alana Harris

**Examples**

```

library(lubridate)
dcut <- tibble::tribble(
  ~USUBJID, ~DCUTDTM,
  "subject1", ymd_hms("2020-10-11T23:59:59"),
  "subject2", ymd_hms("2020-10-11T23:59:59"),
  "subject4", ymd_hms("2020-10-11T23:59:59")
)

ae <- tibble::tribble(
  ~USUBJID, ~AESEQ, ~AESTDTC,
  "subject1", 1, "2020-01-02T00:00:00",
  "subject1", 2, "2020-08-31T00:00:00",
  "subject1", 3, "2020-10-10T00:00:00",
  "subject2", 2, "2020-02-20T00:00:00",
  "subject3", 1, "2020-03-02T00:00:00",
  "subject4", 1, "2020-11-02T00:00:00"
)

ae_out <- pt_cut(
  dataset_sdtm = ae,
  dataset_cut = dcut
)

```

---

read\_out

*Function to generate datacut summary file*


---

**Description**

Produces a .html file summarizing the changes applied to data during a data cut. The file will contain an overview for the change in number of records for each dataset, the types of cut applied and the opportunity to inspect the removed records.

**Usage**

```

read_out(
  dcut = NULL,
  patient_cut_data = NULL,
  date_cut_data = NULL,
  dm_cut = NULL,
  no_cut_list = NULL,
  out_path = "."
)

```

**Arguments**

**dcut**                    The output datacut dataset (DCUT), created via the `create_dcut()` function, containing the variable DCUTDTC.

patient_cut_data	A list of quoted SDTMv domain names in which a patient cut has been applied (via the <code>pt_cut()</code> function). To be left blank if a patient cut has not been performed on any domains.
date_cut_data	A list of quoted SDTMv domain names in which a date cut has been applied. (via the <code>date_cut()</code> function). To be left blank if a date cut has not been performed on any domains.
dm_cut	The output dataset, created via the <code>special_dm_cut()</code> function, containing the variables <code>DCUT_TEMP_REMOVE</code> and <code>DCUT_TEMP_DTHCHANGE</code> .
no_cut_list	List of of quoted SDTMv domain names in which no cut should be applied. To be left blank if no domains are to remain exactly as source.
out_path	A character vector of file save path for the summary file; the default corresponds to the working directory, <code>getwd()</code> .

### Value

Returns a .html file summarizing the changes made to data during a datacut.

### Examples

```
## Not run:
dcut <- tibble::tribble(
  ~USUBJID, ~DCUTDTM, ~DCUTDTC,
  "subject1", lubridate::ymd_hms("2020-10-11T23:59:59"), "2020-10-11T23:59:59",
  "subject2", lubridate::ymd_hms("2020-10-11T23:59:59"), "2020-10-11T23:59:59",
  "subject4", lubridate::ymd_hms("2020-10-11T23:59:59"), "2020-10-11T23:59:59"
)

ae <- tibble::tribble(
  ~USUBJID, ~AESEQ, ~AESTDTC,
  "subject1", 1, "2020-01-02T00:00:00",
  "subject1", 2, "2020-08-31T00:00:00",
  "subject1", 3, "2020-10-10T00:00:00",
  "subject2", 2, "2020-02-20T00:00:00",
  "subject3", 1, "2020-03-02T00:00:00",
  "subject4", 1, "2020-11-02T00:00:00",
  "subject4", 2, ""
)

dm <- tibble::tribble(
  ~USUBJID, ~DTHDTC, ~DTHFL,
  "subject1", "2020-10-11", "Y",
  "subject2", "2020-10-12", "Y",
)

dt_ae <- date_cut(
  dataset_sdtm = ae,
  sdtm_date_var = AESTDTC,
  dataset_cut = dcut,
  cut_var = DCUTDTM
)
```

```
pt_ae <- pt_cut(  
  dataset_sdtm = ae,  
  dataset_cut = dcut  
)  
  
dm_cut <- special_dm_cut(  
  dataset_dm = dm,  
  dataset_cut = dcut,  
  cut_var = DCUTDTM  
)  
  
read_out(dcut, patient_cut_data = list(ae = pt_ae), date_cut_data = list(ae = dt_ae), dm_cut)  
  
## End(Not run)
```

---

special\_dm\_cut

*Special DM Cut to reset Death variable information past cut date*

---

### Description

Applies patient cut if patient not in source DCUT, as well as clearing death information within DM if death occurred after datacut date

### Usage

```
special_dm_cut(dataset_dm, dataset_cut, cut_var = DCUTDTM)
```

### Arguments

dataset_dm	Input DM SDTMv dataset
dataset_cut	Input datacut dataset
cut_var	Datacut date variable found in the dataset_cut dataset, default is DCUTDTM

### Value

Input dataset plus a flag DCUT\_TEMP\_REMOVE to indicate which observations would be dropped when a datacut is applied, and a flag DCUT\_TEMP\_DTHCHANGE to indicate which observations have death occurring after data cut date for clearing

### Author(s)

Tim Barnett



**Examples**

```
dcut <- tibble::tribble(
  ~USUBJID, ~DCUTDTC, ~DCUTDTM,
  "01-701-1015", "2014-10-20T23:59:59", lubridate::ymd_hms("2014-10-20T23:59:59"),
  "01-701-1023", "2014-10-20T23:59:59", lubridate::ymd_hms("2014-10-20T23:59:59")
)

dm <- tibble::tribble(
  ~USUBJID, ~DTHDTC, ~DTHFL,
  "01-701-1015", "2014-10-20", "Y",
  "01-701-1023", "2014-10-21", "Y",
)

special_dm_cut(
  dataset_dm = dm,
  dataset_cut = dcut,
  cut_var = DCUTDTM
)
```

# Index

## \* data

- datacutr\_ae, 4
- datacutr\_dm, 5
- datacutr\_ds, 5
- datacutr\_fa, 6
- datacutr\_lb, 6
- datacutr\_sc, 7
- datacutr\_ts, 7

## \* derive

- apply\_cut, 2
- create\_dcut, 3
- date\_cut, 8
- impute\_dcutdtc, 10
- impute\_sdtm, 11
- process\_cut, 11
- pt\_cut, 13
- read\_out, 14
- special\_dm\_cut, 16

## \* user\_utility

- drop\_temp\_vars, 9

apply\_cut, 2

create\_dcut, 3

datacutr\_ae, 4  
datacutr\_dm, 5  
datacutr\_ds, 5  
datacutr\_fa, 6  
datacutr\_lb, 6  
datacutr\_sc, 7  
datacutr\_ts, 7  
date\_cut, 8  
drop\_temp\_vars, 9

impute\_dcutdtc, 10  
impute\_sdtm, 11

process\_cut, 11  
pt\_cut, 13

read\_out, 14

special\_dm\_cut, 16