

# Creating-an-OCCDS-ADaM-dataset.R

Admin

2024-10-23

```
# Creating an OCCDS ADaM

# Read in Data
# CDISC Pilot SDTM and ADaM datasets –which are included in
# {pharmaversesdtm}– are used.
library(admiral)
library(dplyr, warn.conflicts = FALSE)
library(pharmaversesdtm)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```

data("ae")
data("admiral_adsl")

ae <- convert_blanks_to_na(ae)
adsl <- admiral_adsl

# join ADSL to your AE domain Only the ADSL variables used for derivations
# are selected at this step. The rest of the relevant ADSL variables would
# be added later.

adsl_vars <- exprs(TRTSDT, TRTEDT, TRT01A, TRT01P, DTHDT, EOSDT)

adae <- derive_vars_merged(
  ae,
  dataset_add = adsl,
  new_vars = adsl_vars,
  by = exprs(STUDYID, USUBJID)
)

# Derive/Impute End and Start Analysis Date/time and Relative Day
# derives ASTDTM, ASTDT, ASTDY, AENDTM, AENDT, and AENDY.
# derive_vars_dtm() can be used to derive ASTDTM and AENDTM where ASTDTM could be company-spe
cific.
# derive_vars_dtm_to_dt() used to derive ASTDT and AENDT from ASTDTM and AENDTM
# derive_vars_dy() can be used to create ASTDY and AENDY.

adae <- adae %>%
  derive_vars_dtm(
    dtc = AESTDTC,
    new_vars_prefix = "AST",
    highest_imputation = "M",
    min_dates = exprs(TRTSDT)
  ) %>%
  derive_vars_dtm(
    dtc = AEENDTC,
    new_vars_prefix = "AEN",
    highest_imputation = "M",
    date_imputation = "last",
    time_imputation = "last",
    max_dates = exprs(DTHDT, EOSDT)
  ) %>%
  derive_vars_dtm_to_dt(exprs(ASTDTM, AENDTM)) %>%
  derive_vars_dy(
    reference_date = TRTSDT,
    source_vars = exprs(ASTDT, AENDT)
  )

# Derive Durations
# derive_vars_duration() can be used to create the variables ADURN and ADURU.
adae <- adae %>%
  derive_vars_duration(
    new_var = ADURN,
    new_var_unit = ADURU,
    start_date = ASTDT,
    end_date = AENDT
  )

```

```

)

# Derive ATC variables
# derive_vars_atc() can be used to add Anatomical Therapeutic Chemical class variables from F
ACM to ADCM.

cm <- tibble::tribble(
  ~USUBJID,      ~CMGRPID, ~CMREFID, ~CMDECOD,
  "BP40257-1001", "14", "1192056", "PARACETAMOL",
  "BP40257-1001", "18", "2007001", "SOLUMEDROL",
  "BP40257-1002", "19", "2791596", "SPIRONOLACTONE"
)

facm <- tibble::tribble(
  ~USUBJID,      ~FAGRPID, ~FAREFID, ~FATESTCD, ~FASTRESC,
  "BP40257-1001", "1", "1192056", "CMATC1CD", "N",
  "BP40257-1001", "1", "1192056", "CMATC2CD", "N02",
  "BP40257-1001", "1", "1192056", "CMATC3CD", "N02B",
  "BP40257-1001", "1", "1192056", "CMATC4CD", "N02BE",
  "BP40257-1001", "1", "2007001", "CMATC1CD", "D",
  "BP40257-1001", "1", "2007001", "CMATC2CD", "D10",
  "BP40257-1001", "1", "2007001", "CMATC3CD", "D10A",
  "BP40257-1001", "1", "2007001", "CMATC4CD", "D10AA",
  "BP40257-1001", "2", "2007001", "CMATC1CD", "D",
  "BP40257-1001", "2", "2007001", "CMATC2CD", "D07",
  "BP40257-1001", "2", "2007001", "CMATC3CD", "D07A",
  "BP40257-1001", "2", "2007001", "CMATC4CD", "D07AA",
  "BP40257-1001", "3", "2007001", "CMATC1CD", "H",
  "BP40257-1001", "3", "2007001", "CMATC2CD", "H02",
  "BP40257-1001", "3", "2007001", "CMATC3CD", "H02A",
  "BP40257-1001", "3", "2007001", "CMATC4CD", "H02AB",
  "BP40257-1002", "1", "2791596", "CMATC1CD", "C",
  "BP40257-1002", "1", "2791596", "CMATC2CD", "C03",
  "BP40257-1002", "1", "2791596", "CMATC3CD", "C03D",
  "BP40257-1002", "1", "2791596", "CMATC4CD", "C03DA"
)

derive_vars_atc(cm, facm)

```

```

## Warning: Values from `FASTRESC` are not uniquely identified; output will contain
## list-cols.
## • Use `values_fn = list` to suppress this warning.
## • Use `values_fn = {summary_fun}` to summarise duplicates.
## • Use the following dplyr code to identify duplicates.
## {data} |>
## dplyr::summarise(n = dplyr::n(), .by = c(USUBJID, FAREFID, FATESTCD)) |>
## dplyr::filter(n > 1L)

```

```

## # A tibble: 3 × 8
##   USUBJID      CMGRPID CMREFID CMDECOD      ATC1CD      ATC2CD      ATC3CD ATC4CD
##   <chr>      <chr>   <chr>   <chr>   <list>    <list>    <list> <list>
## 1 BP40257-1001 14      1192056 PARACETAMOL <chr [1]> <chr [1]> <chr> <chr>
## 2 BP40257-1001 18      2007001 SOLUMEDROL  <chr [3]> <chr [3]> <chr> <chr>
## 3 BP40257-1002 19      2791596 SPIRONOLACTONE <chr [1]> <chr [1]> <chr> <chr>

```

```

# Derive Planned and Actual Treatment
# TRTA and TRTP must match at least one value of the character treatment
# variables in ADSL (e.g., TRTxxA/TRTxXP, TRTSEQA/TRTSEQP, TRxxAGy/TRxxPGy).
adae <- mutate(adae, TRTP = TRT01P, TRTA = TRT01A)

count(adae, TRTP, TRTA, TRT01P, TRT01A)

```

```

## # A tibble: 4 × 5
##   TRTP          TRTA          TRT01P          TRT01A          n
##   <chr>        <chr>        <chr>        <chr>      <int>
## 1 Placebo      Placebo      Placebo      Placebo       301
## 2 Xanomeline High Dose Xanomeline High Dose Xanomeline High Dose Xanomeli...   436
## 3 Xanomeline High Dose Xanomeline Low Dose  Xanomeline High Dose Xanomeli...    19
## 4 Xanomeline Low Dose  Xanomeline Low Dose  Xanomeline Low Dose  Xanomeli...   435

```

```

# Derive Date/Date-time of Last Dose
# derive_vars_joined() can be used to derive the last dose date before the start of the event.
data(ex_single)
ex_single <- derive_vars_dtm(
  ex_single,
  dtc = EXSTDTC,
  new_vars_prefix = "EXST",
  flag_imputation = "none"
)

adae <- derive_vars_joined(
  adae,
  ex_single,
  by_vars = exprs(STUDYID, USUBJID),
  new_vars = exprs(LDOSEDTC = EXSTDTC),
  join_vars = exprs(EXSTDTC),
  join_type = "all",
  order = exprs(EXSTDTC),
  filter_add = (EXDOSE > 0 | (EXDOSE == 0 & grepl("PLACEBO", EXTRT))) & !is.na(EXSTDTC),
  filter_join = EXSTDTC <= ASTDTC,
  mode = "last"
)

# Derive Severity, Causality, and Toxicity Grade
adae <- adae %>%
  mutate(
    ASEV = AESEV,
    AREL = AEREL
  )

# Derive Treatment Emergent Flag
# derive_var_trtemfl() is used to derive 30 days TRTEMFL flag derivation.
adae <- adae %>%
  derive_var_trtemfl(
    trt_start_date = TRTSDT,
    trt_end_date = TRTEDT,
    end_window = 30
  )

# derive_var_ontrtfl() is used to derive on-treatment flag
# (ONTRTFL) with a value of "Y" or NA in an ADaM dataset with a single occurrence date
bds1 <- tibble::tribble(
  ~USUBJID, ~ADT, ~TRTSDT, ~TRTEDT,
  "P01", ymd("2020-02-24"), ymd("2020-01-01"), ymd("2020-03-01"),
  "P02", ymd("2020-01-01"), ymd("2020-01-01"), ymd("2020-03-01"),
  "P03", ymd("2019-12-31"), ymd("2020-01-01"), ymd("2020-03-01")
)

derive_var_ontrtfl(
  bds1,
  start_date = ADT,
  ref_start_date = TRTSDT,
  ref_end_date = TRTEDT
)

```

```
## # A tibble: 3 × 5
##   USUBJID ADT          TRTSDT      TRTEDT      ONTRTFL
##   <chr>    <date>      <date>      <date>      <chr>
## 1 P01      2020-02-24 2020-01-01 2020-03-01 Y
## 2 P02      2020-01-01 2020-01-01 2020-03-01 Y
## 3 P03      2019-12-31 2020-01-01 2020-03-01 <NA>
```

```
bds2 <- tibble::tribble(
  ~USUBJID, ~ADT,          ~TRTSDT,          ~TRTEDT,
  "P01",    ymd("2020-07-01"), ymd("2020-01-01"), ymd("2020-03-01"),
  "P02",    ymd("2020-04-30"), ymd("2020-01-01"), ymd("2020-03-01"),
  "P03",    ymd("2020-03-15"), ymd("2020-01-01"), ymd("2020-03-01")
)
derive_var_ontrtfl(
  bds2,
  start_date = ADT,
  ref_start_date = TRTSDT,
  ref_end_date = TRTEDT,
  ref_end_window = 60
)
```

```
## # A tibble: 3 × 5
##   USUBJID ADT          TRTSDT      TRTEDT      ONTRTFL
##   <chr>    <date>      <date>      <date>      <chr>
## 1 P01      2020-07-01 2020-01-01 2020-03-01 <NA>
## 2 P02      2020-04-30 2020-01-01 2020-03-01 Y
## 3 P03      2020-03-15 2020-01-01 2020-03-01 Y
```

```
bds3 <- tibble::tribble(
  ~ADTM,          ~TRTSDTM,          ~TRTEDTM,          ~TPT,
  "2020-01-02T12:00", "2020-01-01T12:00", "2020-03-01T12:00", NA,
  "2020-01-01T12:00", "2020-01-01T12:00", "2020-03-01T12:00", "PRE",
  "2019-12-31T12:00", "2020-01-01T12:00", "2020-03-01T12:00", NA
) %>%
  mutate(
    ADMT = ymd_hm(ADTM),
    TRTSDTM = ymd_hm(TRTSDTM),
    TRTEDTM = ymd_hm(TRTEDTM)
  )
derive_var_ontrtfl(
  bds3,
  start_date = ADMT,
  ref_start_date = TRTSDTM,
  ref_end_date = TRTEDTM,
  filter_pre_timepoint = TPT == "PRE"
)
```

```
## # A tibble: 3 × 5
```

```
##   ADMT          TRTSDTM          TRTEDTM          TPT  ONTRTFL
##   <dtm>          <dtm>          <dtm>          <chr> <chr>
## 1 2020-01-02 12:00:00 2020-01-01 12:00:00 2020-03-01 12:00:00 <NA>   Y
## 2 2020-01-01 12:00:00 2020-01-01 12:00:00 2020-03-01 12:00:00 PRE    <NA>
## 3 2019-12-31 12:00:00 2020-01-01 12:00:00 2020-03-01 12:00:00 <NA>   <NA>
```

```

# Derive Occurrence Flags
# derive_var_extreme_flag() can help derive variables such as AOCCIFL,
# AOCCPIFL, AOCCSIFL, and AOCCzzFL.
adae <- adae %>%
  restrict_derivation(
    derivation = derive_var_extreme_flag,
    args = params(
      by_vars = exprs(USUBJID),
      order = exprs(
        as.integer(factor(
          ASEV,
          levels = c("DEATH THREATENING", "SEVERE", "MODERATE", "MILD")
        )),
        ASTDTM, AESEQ
      ),
      new_var = AOCCIFL,
      mode = "first"
    ),
    filter = TRTEMFL == "Y"
  )
# Derive Query Variables
# derive_vars_query() function can be used to derive query variables
# SMQzzNAM, SMQzzCD, SMQzzSC, SMQzzSCN, or CQzzNAM
# create_query_data() function can be used to create queries datasets.
data("queries")
adae1 <- tibble::tribble(
  ~USUBJID, ~ASTDTM, ~AETERM, ~AESEQ, ~AEDECOD, ~AELLT, ~AELLTCD,
  "01", "2020-06-02 23:59:59", "ALANINE AMINOTRANSFERASE ABNORMAL",
  3, "Alanine aminotransferase abnormal", NA_character_, NA_integer_,
  "02", "2020-06-05 23:59:59", "BASEDOW'S DISEASE",
  5, "Basedow's disease", NA_character_, 1L,
  "03", "2020-06-07 23:59:59", "SOME TERM",
  2, "Some query", "Some term", NA_integer_,
  "05", "2020-06-09 23:59:59", "ALVEOLAR PROTEINOSIS",
  7, "Alveolar proteinosis", NA_character_, NA_integer_
)

adae_query <- derive_vars_query(dataset = adae1, dataset_queries = queries)
# the derive_vars_query() function can also be used to derive
# Standardized Drug Groupings (SDG).
sdg <- tibble::tribble(
  ~PREFIX, ~GRPNAME, ~GRPID, ~SCOPE, ~SCOPEN, ~SRCVAR, ~TERMCHAR, ~TERMN
UM,
  "SDG01", "Diuretics", 11, "BROAD", 1, "CMDECOD", "Diuretic 1", NA,
  "SDG01", "Diuretics", 11, "BROAD", 1, "CMDECOD", "Diuretic 2", NA,
  "SDG02", "Corticosteroids", 12, "BROAD", 1, "CMDECOD", "Corticosteroid 1", NA,
  "SDG02", "Corticosteroids", 12, "BROAD", 1, "CMDECOD", "Corticosteroid 2", NA,
  "SDG02", "Corticosteroids", 12, "BROAD", 1, "CMDECOD", "Corticosteroid 3", NA,
)
adcm <- tibble::tribble(
  ~USUBJID, ~ASTDTM, ~CMDECOD,
  "01", "2020-06-02 23:59:59", "Diuretic 1",
  "02", "2020-06-05 23:59:59", "Diuretic 1",
  "03", "2020-06-07 23:59:59", "Corticosteroid 2",
  "05", "2020-06-09 23:59:59", "Diuretic 2"
)

```



```

)
adcm_query <- derive_vars_query(adcm, sdg)
adae <- adae %>%
  derive_vars_merged(
    dataset_add = select(adsl, !!!negate_vars(adsl_vars)),
    by_vars = exprs(STUDYID, USUBJID)
  )
# Derive Analysis Sequence Number
# derive_var_obs_number() can be used for deriving ASEQ variable to
# ensure the uniqueness of subject records within the dataset.
adcm <- tibble::tribble(
  ~USUBJID,      ~ASTDTM,      ~CMSEQ, ~CMDECOD,      ~ATC1CD, ~ATC2CD, ~ATC3CD, ~ATC
4CD,
  "BP40257-1001", "2013-07-05 UTC", "14",   "PARACETAMOL",   "N",     "N02",   "N02B",   "N02
BE",
  "BP40257-1001", "2013-08-15 UTC", "18",   "SOLUMEDROL",    "D",     "D10",   "D10A",   "D10
AA",
  "BP40257-1001", "2013-08-15 UTC", "18",   "SOLUMEDROL",    "D",     "D07",   "D07A",   "D07
AA",
  "BP40257-1001", "2013-08-15 UTC", "18",   "SOLUMEDROL",    "H",     "H02",   "H02A",   "H02
AB",
  "BP40257-1002", "2012-12-15 UTC", "19",   "SPIRONOLACTONE", "C",     "C03",   "C03D",   "C03
DA"
)

adcm_aseq <- adcm %>%
  derive_var_obs_number(
    by_vars      = exprs(USUBJID),
    order        = exprs(ASTDTM, CMSEQ, ATC1CD, ATC2CD, ATC3CD, ATC4CD),
    new_var      = ASEQ,
    check_type   = "error"
  )

```