Creating a Pharmacokinetics (PK) Non-compartmental analysis (NCA) ADaM (ADNCA/ADPC) or a Population PK ADaM (ADPPK)

Lun-Hsien Chang

2025-01-05

Programming workflow

Introduction

This article describes creating a Pharmacokinetics (PK) Non-compartmental analysis (NCA) ADaM (ADNCA/ADPC) or a Population PK ADaM (ADPPK). The first part of the article describes the NCA file creation while the second part describes Population PK. This initial steps for both files are very similar and could be combined in one script if desired.

Programming PK NCA (ADPC/ADNCA) Analysis Data

The Non-compartmental analysis (NCA) ADaM uses the [CDISC Implementation Guide](https://www.cdisc.org/standards/foundational/adam/adamig-non-compartmental-analysis-input-data-v1-0). This example presented uses underlying `EX` and `PC` domains where the `EX` and `PC` domains represent data as collected and the `ADPC` ADaM is output. However, the example can be applied to situations where an `EC` domain is used as input instead of `EX` and/or `ADNCA` or another ADaM is created.

One of the important aspects of the dataset is the derivation of relative timing variables. These variables consist of nominal and actual times, and refer to the time from first dose or time from most recent reference dose. The reference dose for pre-dose records may be the upcoming dose. The CDISC Implementation Guide makes use of duplicated records for analysis, which allows the same record to be used both with respect to the previous dose and the next upcoming dose. This is illustrated later in this vignette.

Here are the relative time variables we will use. These correspond to the names in the CDISC Implementation Guide.

Variable	Variable Label	CDISC Notes
NFRLT	Nominal Relative Time from Analyte First Dose	This is the planned elapsed time (for sample point or start of sampling interval) from first exposure to treatment associated with PARAM and ANALYTE. For studies with an extended duration infusion, use the define to document if this is from the start or end of the infusion.
AFRLT	Actual Relative Time from Analyte First Dose	This is the actual elapsed time (for sample point or start of sampling interval) from first exposure to treatment associated with PARAM and ANALYTE. Note that this is referring to the first dose available for a particular drug. It is useful in multiple-dosing situations
NRRLT	Nominal Relative Time from Ref. Dose	This is the planned elapsed time (for sample point or start of sampling interval) from reference exposure to study treatment

Variable	Variable Label	CDISC Notes
ARRLT	Actual Relative Time from Ref. Dose	This is the actual elapsed time (for sample point or start of sampling interval) from reference exposure to study treatment.
MRRLT	Modified Relative Time from Ref. Dose	This variable could be used to modify the ARRLT variable based on analysis needs (e.g., setting negative values to zero or having a mix of nominal and actual time based of TMPCTDF).

Note: *All examples assume CDISC SDTM and/or ADaM format as input unless otherwise specified.*

Read in Data

To start, all data frames needed for the creation of `ADPC` should be read into the environment. This will be a company specific process. Some of the data frames needed may be `PC`, `EX`, and `ADSL`.

Additional domains such as `VS` and `LB` may be used for additional baseline variables if needed. These may come from either the SDTM or ADaM source.

For the purpose of example, the CDISC Pilot SDTM and ADaM datasets—which are included in `{pharmaversesdtm}`—are used.

```
Warning: package 'pharmaversesdtm' was built under R version 4.4.2
```

```
Warning: package 'lubridate' was built under R version 4.4.2
```

```

Attaching package: 'lubridate'
```

```
The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

At this step, it may be useful to join `ADSL` to your `PC` and `EX` domains as well. Only the `ADSL` variables used for derivations are selected at this step. The rest of the relevant `ADSL` variables will be added later.

In this case we will keep `TRTSDT`/`TRTSDTM` for day derivation and `TRT01P`/`TRT01A` for planned and actual treatments.

In this segment we will use `derive_vars_merged()` to join the `ADSL` variables and the following `{admiral}` functions to derive analysis dates, times and days: `derive_vars_dtm()`, `derive_vars_dtm_to_dt()`,

`derive_vars_dtm_to_tm()`, `derive_vars_dy()`. We will also create `NFRLT` for `PC` data based on `PCTPTNUM`. We will create an event ID (`EVID`) of 0 for concentration records and 1 for dosing records. This is a traditional variable that will provide a handy tool to identify records but will be dropped from the final dataset in this example.

```
adsl_vars <- exprs(TRTSDT, TRTSDTM, TRT01P, TRT01A)

pc_dates <- pc %>%
  # Join ADSL with PC (need TRTSDT for ADY derivation)
  derive_vars_merged(
    dataset_add = adsl,
    new_vars = adsl_vars,
    by_vars = exprs(STUDYID, USUBJID)
  ) %>%
  # Derive analysis date/time
  # Impute missing time to 00:00:00
  derive_vars_dtm(
    new_vars_prefix = "A",
    dtc = PCDTC,
    time_imputation = "00:00:00"
  ) %>%
  # Derive dates and times from date/times
  derive_vars_dtm_to_dt(exprs(ADTM)) %>%
  derive_vars_dtm_to_tm(exprs(ADTM)) %>%
  derive_vars_dy(reference_date = TRTSDT, source_vars = exprs(ADT)) %>%
  # Derive event ID and nominal relative time from first dose (NFRLT)
  mutate(
    EVID = 0,
    DRUG = PCTEST,
    NFRLT = if_else(PCTPTNUM < 0, 0, PCTPTNUM), .after = USUBJID
  ) # dim(pc_dates) [1] 4572   32

pc_dates %>% select(USUBJID,PCTEST,ADTM,VISIT,PCTPT,NFRLT) %>% head(n=10)
```

```
# A tibble: 10 x 6
   USUBJID     PCTEST     ADTM                VISIT    PCTPT            NFRLT
   <chr>       <chr>      <dttm>              <chr>    <chr>            <dbl>
 1 01-701-1015 XANOMELINE 2014-01-01 23:30:00 BASELINE Pre-dose          0
 2 01-701-1015 XANOMELINE 2014-01-02 00:05:00 BASELINE 5 Min Post-dose   0.08
 3 01-701-1015 XANOMELINE 2014-01-02 00:30:00 BASELINE 30 Min Post-dose  0.5
 4 01-701-1015 XANOMELINE 2014-01-02 01:00:00 BASELINE 1h Post-dose      1
 5 01-701-1015 XANOMELINE 2014-01-02 01:30:00 BASELINE 1.5h Post-dose    1.5
 6 01-701-1015 XANOMELINE 2014-01-02 02:00:00 BASELINE 2h Post-dose      2
 7 01-701-1015 XANOMELINE 2014-01-02 04:00:00 BASELINE 4h Post-dose      4
 8 01-701-1015 XANOMELINE 2014-01-02 06:00:00 BASELINE 6h Post-dose      6
 9 01-701-1015 XANOMELINE 2014-01-02 08:00:00 BASELINE 8h Post-dose      8
10 01-701-1015 XANOMELINE 2014-01-02 12:00:00 BASELINE 12h Post-dose    12
```

Next we will also join ADSL data with EX and derive dates/times. This section uses the {admiral} functions derive_vars_merged(), derive_vars_dtm(), and derive_vars_dtm_to_dt(). Time is imputed to 00:00:00 here for reasons specific to the sample data. Other imputation times may be used based on study details. Here we create NFRLT for EX data based on VISITDY using dplyr::mutate().

```
# ---- Get dosing information ----

ex_dates <- ex %>%
  derive_vars_merged(
    dataset_add = adsl,
    new_vars = adsl_vars,
    by_vars = exprs(STUDYID, USUBJID)
  ) %>%
  # Keep records with nonzero dose
  filter(EXDOSE > 0) %>%
  # Add time and set missing end date to start date
  # Impute missing time to 00:00:00
  # Note all times are missing for dosing records in this example data
  # Derive Analysis Start and End Dates
  derive_vars_dtm(
    new_vars_prefix = "AST",
    dtc = EXSTDTC,
    time_imputation = "00:00:00"
  ) %>%
  derive_vars_dtm(
    new_vars_prefix = "AEN",
    dtc = EXENDTC,
    time_imputation = "00:00:00"
  ) %>%
  # Derive event ID and nominal relative time from first dose (NFRLT)
  mutate(
    EVID = 1,
    NFRLT = case_when(
      VISITDY == 1 ~ 0,
      TRUE ~ 24 * VISITDY
    )
  ) %>%
  # Set missing end dates to start date
  mutate(AENDTM = case_when(
    is.na(AENDTM) ~ ASTDTM,
    TRUE ~ AENDTM
  )) %>%
  # Derive dates from date/times
  derive_vars_dtm_to_dt(exprs(ASTDTM)) %>%
  derive_vars_dtm_to_dt(exprs(AENDTM)) # dim(ex_dates) [1] 365  29

ex_dates %>% select(USUBJID,EXTRT,EXDOSFRQ,ASTDTM,AENDTM,VISIT,VISITDY,NFRLT) %>% head(n=10)
```

```
# A tibble: 10 x 8
   USUBJID  EXTRT EXDOSFRQ ASTDTM              AENDTM              VISIT VISITDY
   <chr>    <chr> <chr>    <dttm>              <dttm>              <chr>   <dbl>
 1 01-701-~ XANO~ QD       2013-07-19 00:00:00 2013-08-01 00:00:00 BASE~       1
 2 01-701-~ XANO~ QD       2013-08-02 00:00:00 2014-01-06 00:00:00 WEEK~      14
 3 01-701-~ XANO~ QD       2014-01-07 00:00:00 2014-01-14 00:00:00 WEEK~     168
 4 01-701-~ XANO~ QD       2014-03-18 00:00:00 2014-03-31 00:00:00 BASE~       1
 5 01-701-~ XANO~ QD       2014-07-01 00:00:00 2014-07-15 00:00:00 BASE~       1
 6 01-701-~ XANO~ QD       2014-07-16 00:00:00 2014-12-17 00:00:00 WEEK~      14
 7 01-701-~ XANO~ QD       2014-12-18 00:00:00 2014-12-30 00:00:00 WEEK~     168
 8 01-701-~ XANO~ QD       2014-01-01 00:00:00 2014-01-15 00:00:00 BASE~       1
 9 01-701-~ XANO~ QD       2014-01-16 00:00:00 2014-06-18 00:00:00 WEEK~      14
10 01-701-~ XANO~ QD       2014-06-19 00:00:00 2014-07-09 00:00:00 WEEK~     168
# i 1 more variable: NFRLT <dbl>
```

Expand Dosing Records

The function `create_single_dose_dataset()` can be used to expand dosing records between the start date and end date. The nominal time will also be expanded based on the values of `EXDOSFRQ`, for example "QD" will result in nominal time being incremented by 24 hours and "BID" will result in nominal time being incremented by 12 hours. This is a new feature of `create_single_dose_dataset()`. QD: quanque die, once a day. BID: bis in die, twice a day.

Dates and times will be derived after expansion using `derive_vars_dtm_to_dt()` and `derive_vars_dtm_to_tm()`.

For this example study we will define analysis visit (`AVISIT`) based on the nominal day value from `NFRLT` and give it the format, "Day 1", "Day 2", "Day 3", etc. This is important for creating the `BASETYPE` variable later. `DRUG` is created from `EXTRT` here. This will be useful for linking treatment data with concentration data if there are multiple drugs and/or analytes, but this variable will also be dropped from the final dataset in this example.

```
# ---- Expand dosing records between start and end dates ----

ex_exp <- ex_dates %>%
  create_single_dose_dataset(
    dose_freq = EXDOSFRQ,
    start_date = ASTDT,
    start_datetime = ASTDTM,
    end_date = AENDT,
    end_datetime = AENDTM,
    nominal_time = NFRLT,
    lookup_table = dose_freq_lookup,
    lookup_column = CDISC_VALUE,
    keep_source_vars = exprs(
      STUDYID, USUBJID, EVID, EXDOSFRQ, EXDOSFRM,
      NFRLT, EXDOSE, EXDOSU, EXTRT, ASTDT, ASTDTM, AENDT, AENDTM,
      VISIT, VISITNUM, VISITDY, TRT01A, TRT01P, DOMAIN, EXSEQ, !!!adsl_vars
    )
```

```
  ) %>%
  # Derive AVISIT based on nominal relative time
  # Derive AVISITN to nominal time in whole days using integer division
  # Define AVISIT based on nominal day
  mutate(
    AVISITN = NFRLT %/% 24 + 1,
    AVISIT = paste("Day", AVISITN),
    ADTM = ASTDTM,
    DRUG = EXTRT,
  ) %>%
  # Derive dates and times from datetimes
  derive_vars_dtm_to_dt(exprs(ADTM)) %>%
  derive_vars_dtm_to_tm(exprs(ADTM)) %>%
  derive_vars_dtm_to_tm(exprs(ASTDTM)) %>%
  derive_vars_dtm_to_tm(exprs(AENDTM)) %>%
  derive_vars_dy(reference_date = TRTSDT, source_vars = exprs(ADT)) # dim(ex_dates) [1] 365  29

ex_exp %>% select(USUBJID,DRUG,EXDOSFRQ,ASTDTM,AENDTM,AVISIT,NFRLT) %>% head(n=10)
```

```
# A tibble: 10 x 7
   USUBJID    DRUG  EXDOSFRQ ASTDTM              AENDTM              AVISIT NFRLT
   <chr>      <chr> <chr>    <dttm>              <dttm>              <chr>  <dbl>
 1 01-701-1~ XANO~ ONCE     2013-07-19 00:00:00 2013-07-19 00:00:00 Day 1      0
 2 01-701-1~ XANO~ ONCE     2013-07-20 00:00:00 2013-07-20 00:00:00 Day 2     24
 3 01-701-1~ XANO~ ONCE     2013-07-21 00:00:00 2013-07-21 00:00:00 Day 3     48
 4 01-701-1~ XANO~ ONCE     2013-07-22 00:00:00 2013-07-22 00:00:00 Day 4     72
 5 01-701-1~ XANO~ ONCE     2013-07-23 00:00:00 2013-07-23 00:00:00 Day 5     96
 6 01-701-1~ XANO~ ONCE     2013-07-24 00:00:00 2013-07-24 00:00:00 Day 6    120
 7 01-701-1~ XANO~ ONCE     2013-07-25 00:00:00 2013-07-25 00:00:00 Day 7    144
 8 01-701-1~ XANO~ ONCE     2013-07-26 00:00:00 2013-07-26 00:00:00 Day 8    168
 9 01-701-1~ XANO~ ONCE     2013-07-27 00:00:00 2013-07-27 00:00:00 Day 9    192
10 01-701-1~ XANO~ ONCE     2013-07-28 00:00:00 2013-07-28 00:00:00 Day 10   216
```

Find First Dose

FANLDTM

- First Datetime of Dose for Analyte
- Date and time of first exposure to treatment associated with PARAM and ANALYTE for a subject in a study where multiple doses have been given. If treatment is given over a duration multiple times, this variable will reflect the start date and time of the first dose.

In this section we will find the first dose for each subject and drug, using `derive_vars_merged()`. We also create an analysis visit (`AVISIT`) based on `NFRLT`. The first dose datetime for an analyte `FANLDTM` is calculated as the minimum `ADTM` from the dosing records by subject and drug.

```
# ---- Find first dose per treatment per subject ----
# ---- Join with ADPC data and keep only subjects with dosing ----

adpc_first_dose <- pc_dates %>%
  derive_vars_merged(
    dataset_add = ex_exp,
    filter_add = (EXDOSE > 0 & !is.na(ADTM)),
    new_vars = exprs(FANLDTM = ADTM),
    order = exprs(ADTM, EXSEQ),
    mode = "first",
    by_vars = exprs(STUDYID, USUBJID, DRUG)
  ) %>%
  filter(!is.na(FANLDTM)) %>%
  # Derive AVISIT based on nominal relative time
  # Derive AVISITN to nominal time in whole days using integer division
  # Define AVISIT based on nominal day
  mutate(
    AVISITN = NFRLT %/% 24 + 1,
    AVISIT = paste("Day", AVISITN)
  ) # dim(pc_dates) [1] 4572   32 # dim(ex_exp) [1] 16331    31 # dim(adpc_first_dose) [1] 3024

adpc_first_dose %>% select(USUBJID,FANLDTM,AVISIT,ADTM,PCTPT) %>% head(n=10)
```

```
# A tibble: 10 x 5
   USUBJID     FANLDTM             AVISIT ADTM                PCTPT
   <chr>       <dttm>              <chr>  <dttm>              <chr>
 1 01-701-1028 2013-07-19 00:00:00 Day 1  2013-07-18 23:30:00 Pre-dose
 2 01-701-1028 2013-07-19 00:00:00 Day 1  2013-07-19 00:05:00 5 Min Post-dose
 3 01-701-1028 2013-07-19 00:00:00 Day 1  2013-07-19 00:30:00 30 Min Post-dose
 4 01-701-1028 2013-07-19 00:00:00 Day 1  2013-07-19 01:00:00 1h Post-dose
 5 01-701-1028 2013-07-19 00:00:00 Day 1  2013-07-19 01:30:00 1.5h Post-dose
 6 01-701-1028 2013-07-19 00:00:00 Day 1  2013-07-19 02:00:00 2h Post-dose
 7 01-701-1028 2013-07-19 00:00:00 Day 1  2013-07-19 04:00:00 4h Post-dose
 8 01-701-1028 2013-07-19 00:00:00 Day 1  2013-07-19 06:00:00 6h Post-dose
 9 01-701-1028 2013-07-19 00:00:00 Day 1  2013-07-19 08:00:00 8h Post-dose
10 01-701-1028 2013-07-19 00:00:00 Day 1  2013-07-19 12:00:00 12h Post-dose
```

Find Reference Dose Dates Corresponding to PK Records

Use `derive_vars_joined()` to find the previous dose data. This will join the expanded `EX` data with the `ADPC` based on the analysis date `ADTM`. Note the `filter_join` parameter. In addition to the date of the previous dose (`ADTM_prev`), we also keep the actual dose amount `EXDOSE_prev` and the analysis visit of the dose `AVISIT_prev`.

```
# ---- Find previous dose  ----

adpc_prev <- adpc_first_dose %>%
  derive_vars_joined(
    dataset_add = ex_exp,
    by_vars = exprs(USUBJID),
    order = exprs(ADTM),
    new_vars = exprs(
      ADTM_prev = ADTM, EXDOSE_prev = EXDOSE, AVISIT_prev = AVISIT,
      AENDTM_prev = AENDTM
    ),
    join_vars = exprs(ADTM),
    join_type = "all",
    filter_add = NULL,
    filter_join = ADTM > ADTM.join,
    mode = "last",
    check_type = "none"
  ) # dim(adpc_prev) [1] 3024   39 # dim(adpc_first_dose) [1] 3024   35

adpc_prev %>% select(USUBJID,VISIT,ADTM,PCTPT,ADTM_prev,EXDOSE_prev,AVISIT_prev) %>% head(n=10)
```

```
# A tibble: 10 x 7
   USUBJID     VISIT   ADTM                PCTPT ADTM_prev           EXDOSE_prev
   <chr>       <chr>   <dttm>              <chr> <dttm>                    <dbl>
 1 01-701-1028 BASELI~ 2013-07-18 23:30:00 Pre-~ NA                         NA
 2 01-701-1028 BASELI~ 2013-07-19 00:05:00 5 Mi~ 2013-07-19 00:00:00        54
 3 01-701-1028 BASELI~ 2013-07-19 00:30:00 30 M~ 2013-07-19 00:00:00        54
 4 01-701-1028 BASELI~ 2013-07-19 01:00:00 1h P~ 2013-07-19 00:00:00        54
 5 01-701-1028 BASELI~ 2013-07-19 01:30:00 1.5h~ 2013-07-19 00:00:00        54
 6 01-701-1028 BASELI~ 2013-07-19 02:00:00 2h P~ 2013-07-19 00:00:00        54
 7 01-701-1028 BASELI~ 2013-07-19 04:00:00 4h P~ 2013-07-19 00:00:00        54
 8 01-701-1028 BASELI~ 2013-07-19 06:00:00 6h P~ 2013-07-19 00:00:00        54
 9 01-701-1028 BASELI~ 2013-07-19 08:00:00 8h P~ 2013-07-19 00:00:00        54
10 01-701-1028 BASELI~ 2013-07-19 12:00:00 12h ~ 2013-07-19 00:00:00        54
# i 1 more variable: AVISIT_prev <chr>
```

Similarly, find next dose information using `derive_vars_joined()` with the `filter_join` parameter as `ADTM <= ADTM.join`. Here we keep the next dose analysis date `ADTM_next`, the next actual dose `EXDOSE_next`, and the next analysis visit `AVISIT_next`.

```
# ---- Find next dose  ----

adpc_next <- adpc_prev %>%
  derive_vars_joined(
    dataset_add = ex_exp,
    by_vars = exprs(USUBJID),
```

```
    order = exprs(ADTM),
    new_vars = exprs(
      ADTM_next = ADTM, EXDOSE_next = EXDOSE, AVISIT_next = AVISIT,
      AENDTM_next = AENDTM
    ),
    join_vars = exprs(ADTM),
    join_type = "all",
    filter_add = NULL,
    filter_join = ADTM <= ADTM.join,
    mode = "first",
    check_type = "none" ) # dim(adpc_next) 3024 43

adpc_next %>% select(USUBJID,VISIT,ADTM,PCTPT,ADTM_next,EXDOSE_next,AVISIT_next) %>% head(n=10)
```

```
# A tibble: 10 x 7
   USUBJID     VISIT   ADTM                PCTPT ADTM_next           EXDOSE_next
   <chr>       <chr>   <dttm>              <chr> <dttm>                    <dbl>
 1 01-701-1028 BASELI~ 2013-07-18 23:30:00 Pre-~ 2013-07-19 00:00:00          54
 2 01-701-1028 BASELI~ 2013-07-19 00:05:00 5 Mi~ 2013-07-20 00:00:00          54
 3 01-701-1028 BASELI~ 2013-07-19 00:30:00 30 M~ 2013-07-20 00:00:00          54
 4 01-701-1028 BASELI~ 2013-07-19 01:00:00 1h P~ 2013-07-20 00:00:00          54
 5 01-701-1028 BASELI~ 2013-07-19 01:30:00 1.5h~ 2013-07-20 00:00:00          54
 6 01-701-1028 BASELI~ 2013-07-19 02:00:00 2h P~ 2013-07-20 00:00:00          54
 7 01-701-1028 BASELI~ 2013-07-19 04:00:00 4h P~ 2013-07-20 00:00:00          54
 8 01-701-1028 BASELI~ 2013-07-19 06:00:00 6h P~ 2013-07-20 00:00:00          54
 9 01-701-1028 BASELI~ 2013-07-19 08:00:00 8h P~ 2013-07-20 00:00:00          54
10 01-701-1028 BASELI~ 2013-07-19 12:00:00 12h ~ 2013-07-20 00:00:00          54
# i 1 more variable: AVISIT_next <chr>
```

Use the same method to find the previous and next nominal times. Note that here the data are sorted by nominal time rather than the actual time. This will tell us when the previous dose and the next dose were supposed to occur. Sometimes this will differ from the actual times in a study. Here we keep the previous nominal dose time `NFRLT_prev` and the next nominal dose time `NFRLT_next`. Note that the `filter_join` parameter uses the nominal relative times, e.g. `NFRLT > NFRLT.join`.

```
# ---- Find previous nominal time ----

adpc_nom_prev <- adpc_next %>%
  derive_vars_joined(
    dataset_add = ex_exp,
    by_vars = exprs(USUBJID),
    order = exprs(NFRLT),
    new_vars = exprs(NFRLT_prev = NFRLT),
    join_vars = exprs(NFRLT),
    join_type = "all",
    filter_add = NULL,
```

```
    filter_join = NFRLT > NFRLT.join,
    mode = "last",
    check_type = "none") # dim(adpc_nom_prev) [1] 3024   44

# ---- Find next nominal time ----

adpc_nom_next <- adpc_nom_prev %>%
  derive_vars_joined(
    dataset_add = ex_exp,
    by_vars = exprs(USUBJID),
    order = exprs(NFRLT),
    new_vars = exprs(NFRLT_next = NFRLT),
    join_vars = exprs(NFRLT),
    join_type = "all",
    filter_add = NULL,
    filter_join = NFRLT <= NFRLT.join,
    mode = "first",
    check_type = "none") # dim(adpc_nom_next) [1] 3024   45

adpc_nom_next %>% select(USUBJID,NFRLT,PCTPT,NFRLT_prev,NFRLT_next) %>% head(n=10)
```

```
# A tibble: 10 x 5
   USUBJID     NFRLT PCTPT            NFRLT_prev NFRLT_next
   <chr>       <dbl> <chr>                 <dbl>      <dbl>
 1 01-701-1028  0    Pre-dose                 NA          0
 2 01-701-1028  0.08 5 Min Post-dose           0         24
 3 01-701-1028  0.5  30 Min Post-dose          0         24
 4 01-701-1028  1    1h Post-dose              0         24
 5 01-701-1028  1.5  1.5h Post-dose            0         24
 6 01-701-1028  2    2h Post-dose              0         24
 7 01-701-1028  4    4h Post-dose              0         24
 8 01-701-1028  6    6h Post-dose              0         24
 9 01-701-1028  8    8h Post-dose              0         24
10 01-701-1028 12    12h Post-dose             0         24
```

Combine PC and EX Records and Derive Relative Time Variables

Combine PC and EX records and derive the additional relative time variables. Often NCA data will keep both dosing and concentration records. We will keep both here. Sometimes you will see ADPC with only the concentration records. If this is desired, the dosing records can be dropped before saving the final dataset. We will use the {admiral} function derive_vars_duration() to calculate the actual relative time from first dose (AFRLT) and the actual relative time from most recent dose (ARRLT). Note that we use the parameter add_one = FALSE here. We will also create a variable representing actual time to next dose (AXRLT) which is not kept, but will be used when we create duplicated records for analysis for the pre-dose records. For now, we will update missing values of ARRLT corresponding to the pre-dose records with AXRLT, and dosing records will be set to zero.

We also calculate the reference dates FANLDTM (First Datetime of Dose for Analyte) and PCRFTDTM (Reference Datetime of Dose for Analyte) and their corresponding date and time variables.

We calculate the maximum date for concentration records and only keep the dosing records up to that date.

```
# ---- Combine ADPC and EX data ----
# Derive Relative Time Variables

adpc_arrlt <- bind_rows(adpc_nom_next, ex_exp) %>%
  group_by(USUBJID, DRUG) %>%
  mutate(
    FANLDTM = min(FANLDTM, na.rm = TRUE),
    min_NFRLT = min(NFRLT_prev, na.rm = TRUE),
    maxdate = max(ADT[EVID == 0], na.rm = TRUE), .after = USUBJID
  ) %>%
  arrange(USUBJID, ADTM) %>%
  ungroup() %>%
  filter(ADT <= maxdate) %>%
  # Derive Actual Relative Time from First Dose (AFRLT)
  derive_vars_duration(
    new_var = AFRLT,
    start_date = FANLDTM,
    end_date = ADTM,
    out_unit = "hours",
    floor_in = FALSE,
    add_one = FALSE
  ) %>%
  # Derive Actual Relative Time from Reference Dose (ARRLT)
  derive_vars_duration(
    new_var = ARRLT,
    start_date = ADTM_prev,
    end_date = ADTM,
    out_unit = "hours",
    floor_in = FALSE,
    add_one = FALSE
  ) %>%
  # Derive Actual Relative Time from Next Dose (AXRLT not kept)
  derive_vars_duration(
    new_var = AXRLT,
    start_date = ADTM_next,
    end_date = ADTM,
    out_unit = "hours",
    floor_in = FALSE,
    add_one = FALSE
  ) %>%
  mutate(
    ARRLT = case_when(
      EVID == 1 ~ 0,
```

```
      is.na(ARRLT) ~ AXRLT,
      TRUE ~ ARRLT
    ),

    # Derive Reference Dose Date
    PCRFTDTM = case_when(
      EVID == 1 ~ ADTM,
      is.na(ADTM_prev) ~ ADTM_next,
      TRUE ~ ADTM_prev
    )
  ) %>%
  # Derive dates and times from datetimes
  derive_vars_dtm_to_dt(exprs(FANLDTM)) %>%
  derive_vars_dtm_to_tm(exprs(FANLDTM)) %>%
  derive_vars_dtm_to_dt(exprs(PCRFTDTM)) %>%
  derive_vars_dtm_to_tm(exprs(PCRFTDTM)) # dim(adpc_arrlt) 3522 68

adpc_arrlt %>% select(USUBJID,FANLDTM,AVISIT,PCTPT,AFRLT,ARRLT,AXRLT) %>% head(n=10)
```

```
# A tibble: 10 x 7
   USUBJID     FANLDTM             AVISIT PCTPT              AFRLT   ARRLT AXRLT
   <chr>       <dttm>              <chr>  <chr>              <dbl>   <dbl> <dbl>
 1 01-701-1028 2013-07-19 00:00:00 Day 1  Pre-dose          -0.5    -0.5   -0.5
 2 01-701-1028 2013-07-19 00:00:00 Day 1  <NA>               0       0      NA
 3 01-701-1028 2013-07-19 00:00:00 Day 1  5 Min Post-dose    0.0833  0.0833 -23.9
 4 01-701-1028 2013-07-19 00:00:00 Day 1  30 Min Post-dose   0.5     0.5    -23.5
 5 01-701-1028 2013-07-19 00:00:00 Day 1  1h Post-dose       1       1      -23
 6 01-701-1028 2013-07-19 00:00:00 Day 1  1.5h Post-dose     1.5     1.5    -22.5
 7 01-701-1028 2013-07-19 00:00:00 Day 1  2h Post-dose       2       2      -22
 8 01-701-1028 2013-07-19 00:00:00 Day 1  4h Post-dose       4       4      -20
 9 01-701-1028 2013-07-19 00:00:00 Day 1  6h Post-dose       6       6      -18
10 01-701-1028 2013-07-19 00:00:00 Day 1  0-6h Post-dose     6       6      -18
```

For nominal relative times we calculate `NRRLT` generally as `NFRLT - NFRLT_prev` and `NXRLT` as `NFRLT - NFRLT_next`.

```
adpc_nrrlt <- adpc_arrlt %>%
  # Derive Nominal Relative Time from Reference Dose (NRRLT)
  mutate(
    NRRLT = case_when(
      EVID == 1 ~ 0,
      is.na(NFRLT_prev) ~ NFRLT - min_NFRLT,
      TRUE ~ NFRLT - NFRLT_prev
    ),
    NXRLT = case_when(
      EVID == 1 ~ 0,
```

```
      TRUE ~ NFRLT - NFRLT_next
    )
  ) # dim(adpc_nrrlt) [1] 3522   70

adpc_nrrlt %>% select(USUBJID,AVISIT,PCTPT,NFRLT,NRRLT,NXRLT) %>% head(n=10)
```

```
# A tibble: 10 x 6
   USUBJID     AVISIT PCTPT            NFRLT NRRLT NXRLT
   <chr>       <chr>  <chr>            <dbl> <dbl> <dbl>
 1 01-701-1028 Day 1  Pre-dose          0     0      0
 2 01-701-1028 Day 1  <NA>              0     0      0
 3 01-701-1028 Day 1  5 Min Post-dose   0.08  0.08 -23.9
 4 01-701-1028 Day 1  30 Min Post-dose  0.5   0.5  -23.5
 5 01-701-1028 Day 1  1h Post-dose      1     1    -23
 6 01-701-1028 Day 1  1.5h Post-dose    1.5   1.5  -22.5
 7 01-701-1028 Day 1  2h Post-dose      2     2    -22
 8 01-701-1028 Day 1  4h Post-dose      4     4    -20
 9 01-701-1028 Day 1  6h Post-dose      6     6    -18
10 01-701-1028 Day 1  0-6h Post-dose    3     3    -21
```

Derive Analysis Variables

Using `dplyr::mutate` we derive a number of analysis variables including analysis value (`AVAL`), analysis time point (`ATPT`) analysis timepoint reference (`ATPTREF`) and baseline type (`BASETYPE`).

We set `ATPT` to `PCTPT` for concentration records and to "Dose" for dosing records. The analysis timepoint reference `ATPTREF` will correspond to the dosing visit. We will use `AVISIT_prev` and `AVISIT_next` to derive. The baseline type will be a concatenation of `ATPTREF` and "Baseline" with values such as "Day 1 Baseline", "Day 2 Baseline", etc. The baseline flag `ABLFL` will be set to "Y" for pre-dose records.

Analysis value `AVAL` in this example comes from `PCSTRESN` for concentration records. In addition we are including the dose value `EXDOSE` for dosing records and setting BLQ (Below Limit of Quantitation) records to 0 before the first dose and to 1/2 of LLOQ (Lower Limit of Quantitation) for records after first dose. (Additional tests such as whether more than 1/3 of records are BLQ may be required and are not done in this example.) We also create a listing-ready variable `AVALCAT1` which includes the "BLQ" record indicator and formats the numeric values to three significant digits.

We derive actual dose `DOSEA` based on `EXDOSE_prev` and `EXDOSE_next` and planned dose `DOSEP` based on the planned treatment `TRT01P`. In addition we add the units for the dose variables and the relative time variables.

```
# ---- Derive Analysis Variables ----
# Derive ATPTN, ATPT, ATPTREF, ABLFL and BASETYPE
# Derive planned dose DOSEP, actual dose DOSEA and units
# Derive PARAMCD and relative time units
# Derive AVAL, AVALU and AVALCAT1
```

```r
adpc_aval <- adpc_nrrlt %>%
  mutate(
    PARCAT1 = PCSPEC,
    ATPTN = case_when(
      EVID == 1 ~ 0,
      TRUE ~ PCTPTNUM
    ),
    ATPT = case_when(
      EVID == 1 ~ "Dose",
      TRUE ~ PCTPT
    ),
    ATPTREF = case_when(
      EVID == 1 ~ AVISIT,
      is.na(AVISIT_prev) ~ AVISIT_next,
      TRUE ~ AVISIT_prev
    ),
    # Derive baseline flag for pre-dose records
    ABLFL = case_when(
      ATPT == "Pre-dose" ~ "Y",
      TRUE ~ NA_character_
    ),
    # Derive BASETYPE
    BASETYPE = paste(ATPTREF, "Baseline"),

    # Derive Actual Dose
    DOSEA = case_when(
      EVID == 1 ~ EXDOSE,
      is.na(EXDOSE_prev) ~ EXDOSE_next,
      TRUE ~ EXDOSE_prev
    ),
    # Derive Planned Dose
    DOSEP = case_when(
      TRT01P == "Xanomeline High Dose" ~ 81,
      TRT01P == "Xanomeline Low Dose" ~ 54
    ),
    DOSEU = "mg",
  ) %>%
  # Derive relative time units
  mutate(
    FRLTU = "h",
    RRLTU = "h",
    # Derive PARAMCD
    PARAMCD = coalesce(PCTESTCD, "DOSE"),
    ALLOQ = PCLLOQ,
    # Derive AVAL
    AVAL = case_when(
      EVID == 1 ~ EXDOSE,
```

```
      PCSTRESC == "<BLQ" & NFRLT == 0 ~ 0,
      PCSTRESC == "<BLQ" & NFRLT > 0 ~ 0.5 * ALLOQ,
      TRUE ~ PCSTRESN
    ),
    AVALU = case_when(
      EVID == 1 ~ EXDOSU,
      TRUE ~ PCSTRESU
    ),
    AVALCAT1 = if_else(PCSTRESC == "<BLQ", PCSTRESC, prettyNum(signif(AVAL, digits = 3))),
  ) %>%
  # Add SRCSEQ
  mutate(
    SRCDOM = DOMAIN,
    SRCVAR = "SEQ",
    SRCSEQ = coalesce(PCSEQ, EXSEQ)
  ) # dim(adpc_aval) [1] 3522   89

adpc_aval %>% select(USUBJID,NFRLT,AVISIT,ATPT,ABLFL,ATPTREF,AVAL,AVALCAT1) %>% head(n=10)
```

```
# A tibble: 10 x 8
   USUBJID     NFRLT AVISIT ATPT             ABLFL ATPTREF   AVAL AVALCAT1
   <chr>       <dbl> <chr>  <chr>            <chr> <chr>    <dbl> <chr>
 1 01-701-1028  0    Day 1  Pre-dose         Y     Day 1    0     <BLQ
 2 01-701-1028  0    Day 1  Dose             <NA>  Day 1   54     <NA>
 3 01-701-1028  0.08 Day 1  5 Min Post-dose  <NA>  Day 1    0.102 0.102
 4 01-701-1028  0.5  Day 1  30 Min Post-dose <NA>  Day 1    0.547 0.547
 5 01-701-1028  1    Day 1  1h Post-dose     <NA>  Day 1    0.925 0.925
 6 01-701-1028  1.5  Day 1  1.5h Post-dose   <NA>  Day 1    1.19  1.19
 7 01-701-1028  2    Day 1  2h Post-dose     <NA>  Day 1    1.37  1.37
 8 01-701-1028  4    Day 1  4h Post-dose     <NA>  Day 1    1.68  1.68
 9 01-701-1028  6    Day 1  6h Post-dose     <NA>  Day 1    1.76  1.76
10 01-701-1028  3    Day 1  0-6h Post-dose   <NA>  Day 1   24.9   24.9
```

Create Duplicated Records for Analysis

DTYPE

- Derivation Type
- Analysis value derivation method. DTYPE is used to denote, and must be populated, when the value of AVAL or AVALC has been imputed or derived differently than the other analysis values within the parameter. DTYPE is required to be populated even if AVAL and AVALC are null on the derived record. \n Three common situations when DTYPE should be populated: \n * A new row is added within a parameter with the analysis value populated based on other rows within the parameter. \n * A new row is added within a parameter with the analysis value populated based on a constant value or data from other subjects. \n * An analysis value (AVAL or AVALC) on an existing record is being replaced

with a value based on a pre-specified algorithm. \n DTYPE is used to denote analysis values that are "special cases" within a parameter. For each value of DTYPE, the precise derivation algorithm must be defined in analysis variable metadata, even for DTYPE values in the CDISC Controlled Terminology. The controlled terminology for DTYPE is extensible. See Section 4, Implementation Issues, Standard Solutions, and Examples for examples of the use of DTYPE. \n Some examples of DTYPE values: \n * LOCF = last observation carried forward \n * WOCF = worst observation carried forward \n * AVERAGE = average of values

As mentioned above, the CDISC ADaM Implementation Guide for Non-compartmental Analysis uses duplicated records for analysis when a record needs to be used in more than one way. In this example the 24 hour post-dose record will also be used a the pre-dose record for the "Day 2" dose. In addition to 24 hour post-dose records, other situations may include pre-dose records for "Cycle 2 Day 1", etc.

In general, we will select the records of interest and then update the relative time variables for the duplicated records. In this case we will select where the nominal relative time to next dose is zero. (Note that we do not need to duplicate the first dose record since there is no prior dose.)

`DTYPE` is set to "COPY" for the duplicated records and the original `PCSEQ` value is retained. In this case we change "24h Post-dose" to "Pre-dose". `ABLFL` is set to "Y" since these records will serve as baseline for the "Day 2" dose. `DOSEA` is set to `EXDOSE_next` and `PCRFTDTM` is set to `ADTM_next`.

```
# ---- Create DTYPE copy records ----

dtype <- adpc_aval %>%
  filter(NFRLT > 0 & NXRLT == 0 & EVID == 0 & !is.na(AVISIT_next)) %>%
  select(-PCRFTDT, -PCRFTTM) %>%
  # Re-derive variables in for DTYPE copy records
  mutate(
    ABLFL = NA_character_,
    ATPTREF = AVISIT_next,
    ARRLT = AXRLT,
    NRRLT = NXRLT,
    PCRFTDTM = ADTM_next,
    DOSEA = EXDOSE_next,
    BASETYPE = paste(AVISIT_next, "Baseline"),
    ATPT = "Pre-dose",
    ATPTN = -0.5,
    ABLFL = "Y",
    DTYPE = "COPY"
  ) %>%
  derive_vars_dtm_to_dt(exprs(PCRFTDTM)) %>%
  derive_vars_dtm_to_tm(exprs(PCRFTDTM)) # dim(dtype) [1] 330  90 # dim(adpc_aval) [1] 3522   8

dtype %>% select(USUBJID,DTYPE,ATPT,NFRLT,NRRLT,AFRLT,ARRLT,BASETYPE) %>% head(n=10)
```

```
# A tibble: 10 x 8
   USUBJID     DTYPE ATPT     NFRLT NRRLT AFRLT ARRLT BASETYPE
   <chr>       <chr> <chr>    <dbl> <dbl> <dbl> <dbl> <chr>
```

```
 1 01-701-1028 COPY  Pre-dose    24     0    24     0 Day 2 Baseline
 2 01-701-1028 COPY  Pre-dose    48     0    48     0 Day 3 Baseline
 3 01-701-1033 COPY  Pre-dose    24     0    24     0 Day 2 Baseline
 4 01-701-1033 COPY  Pre-dose    48     0    48     0 Day 3 Baseline
 5 01-701-1034 COPY  Pre-dose    24     0    24     0 Day 2 Baseline
 6 01-701-1034 COPY  Pre-dose    48     0    48     0 Day 3 Baseline
 7 01-701-1097 COPY  Pre-dose    24     0    24     0 Day 2 Baseline
 8 01-701-1097 COPY  Pre-dose    48     0    48     0 Day 3 Baseline
 9 01-701-1111 COPY  Pre-dose    24     0    24     0 Day 2 Baseline
10 01-701-1111 COPY  Pre-dose    48     0    48     0 Day 3 Baseline
```

Combine ADPC data with Duplicated Records

Now the duplicated records are combined with the original records. We also derive the modified relative time from reference dose MRRLT. In this case, negative values of ARRLT are set to zero.

This is also an opportunity to derive analysis flags e.g. ANL01FL , ANL02FL etc. In this example ANL01FL is set to "Y" for all records and ANL02FL is set to "Y" for all records except the duplicated records with DTYPE = "COPY". Additional flags may be used to select full profile records and/or to select records included in the tables and figures, etc.

```
# ---- Combine original records and DTYPE copy records ----

adpc_dtype <- bind_rows(adpc_aval, dtype) %>%
  arrange(STUDYID, USUBJID, BASETYPE, ADTM, NFRLT) %>%
  mutate(
    # Derive MRRLT, ANL01FL and ANL02FL
    MRRLT = if_else(ARRLT < 0, 0, ARRLT),
    ANL01FL = "Y",
    ANL02FL = if_else(is.na(DTYPE), "Y", NA_character_),
  ) # dim(adpc_dtype) [1] 3852   93

adpc_dtype %>% select(STUDYID,USUBJID,BASETYPE,ADTM,ATPT,NFRLT,NRRLT,ARRLT,MRRLT) %>% head(n=10
```

```
# A tibble: 10 x 9
   STUDYID USUBJID BASETYPE ADTM                ATPT  NFRLT NRRLT   ARRLT  MRRLT
   <chr>   <chr>   <chr>    <dttm>              <chr> <dbl> <dbl>   <dbl>  <dbl>
 1 CDISCP~ 01-701~ Day 1 B~ 2013-07-18 23:30:00 Pre-~  0     0    -0.5    0
 2 CDISCP~ 01-701~ Day 1 B~ 2013-07-19 00:00:00 Dose   0     0     0      0
 3 CDISCP~ 01-701~ Day 1 B~ 2013-07-19 00:05:00 5 Mi~  0.08  0.08  0.0833 0.0833
 4 CDISCP~ 01-701~ Day 1 B~ 2013-07-19 00:30:00 30 M~  0.5   0.5   0.5    0.5
 5 CDISCP~ 01-701~ Day 1 B~ 2013-07-19 01:00:00 1h P~  1     1     1      1
 6 CDISCP~ 01-701~ Day 1 B~ 2013-07-19 01:30:00 1.5h~  1.5   1.5   1.5    1.5
 7 CDISCP~ 01-701~ Day 1 B~ 2013-07-19 02:00:00 2h P~  2     2     2      2
 8 CDISCP~ 01-701~ Day 1 B~ 2013-07-19 04:00:00 4h P~  4     4     4      4
 9 CDISCP~ 01-701~ Day 1 B~ 2013-07-19 06:00:00 0-6h~  3     3     6      6
10 CDISCP~ 01-701~ Day 1 B~ 2013-07-19 06:00:00 6h P~  6     6     6      6
```

Calculate Change from Baseline and Assign ASEQ

The {admiral} function derive_var_base() is used to derive BASE and the function derive_var_chg() is used to derive change from baseline CHG.

We also now derive ASEQ using derive_var_obs_number() and we drop intermediate variables such as those ending with "_prev" and "_next".

Finally we derive PARAM and PARAMN from a lookup table.

```
# ---- Derive BASE and Calculate Change from Baseline ----

adpc_base <- adpc_dtype %>%
  # Derive BASE
  derive_var_base(
    by_vars = exprs(STUDYID, USUBJID, PARAMCD, PARCAT1, BASETYPE),
    source_var = AVAL,
    new_var = BASE,
    filter = ABLFL == "Y"
  ) # dim(adpc_base) [1] 3852   94

adpc_chg <- derive_var_chg(adpc_base) # dim(adpc_chg) [1] 3852   95

# ---- Add ASEQ ----

adpc_aseq <- adpc_chg %>%
  # Calculate ASEQ
  derive_var_obs_number(
    new_var = ASEQ,
    by_vars = exprs(STUDYID, USUBJID),
    order = exprs(ADTM, BASETYPE, EVID, AVISITN, ATPTN, PARCAT1, DTYPE),
    check_type = "error"
  ) %>%
  # Remove temporary variables
  select(
    -DOMAIN, -PCSEQ, -starts_with("orig"), -starts_with("min"),
    -starts_with("max"), -starts_with("EX"), -ends_with("next"),
    -ends_with("prev"), -DRUG, -EVID, -AXRLT, -NXRLT, -VISITDY
  ) %>%
  # Derive PARAM and PARAMN
  derive_vars_merged(
    dataset_add = select(param_lookup, -PCTESTCD), by_vars = exprs(PARAMCD)
  ) # dim(adpc_aseq) [1] 3852   73

adpc_aseq %>% select(USUBJID,BASETYPE,DTYPE,AVISIT,ATPT,AVAL,NFRLT,NRRLT,AFRLT,ARRLT,BASE,CHG)
```

```
# A tibble: 10 x 12
   USUBJID  BASETYPE DTYPE AVISIT ATPT    AVAL NFRLT NRRLT   AFRLT   ARRLT  BASE
```

```
   <chr>    <chr>    <chr> <chr>  <chr>  <dbl> <dbl> <dbl>   <dbl>   <dbl> <dbl>
 1 01-701-~ Day 1 B~ <NA>  Day 1  Pre-~  0      0     0    -0.5    -0.5       0
 2 01-701-~ Day 1 B~ <NA>  Day 1  Dose  54      0     0     0       0        NA
 3 01-701-~ Day 1 B~ <NA>  Day 1  5 Mi~  0.102  0.08  0.08  0.0833  0.0833    0
 4 01-701-~ Day 1 B~ <NA>  Day 1  30 M~  0.547  0.5   0.5   0.5     0.5       0
 5 01-701-~ Day 1 B~ <NA>  Day 1  1h P~  0.925  1     1     1       1         0
 6 01-701-~ Day 1 B~ <NA>  Day 1  1.5h~  1.19   1.5   1.5   1.5     1.5       0
 7 01-701-~ Day 1 B~ <NA>  Day 1  2h P~  1.37   2     2     2       2         0
 8 01-701-~ Day 1 B~ <NA>  Day 1  4h P~  1.68   4     4     4       4         0
 9 01-701-~ Day 1 B~ <NA>  Day 1  0-6h~ 24.9    3     3     6       6        NA
10 01-701-~ Day 1 B~ <NA>  Day 1  6h P~  1.76   6     6     6       6         0
# i 1 more variable: CHG <dbl>
```

Add Additional Baseline Variables

Here we derive additional baseline values from VS for baseline height HTBL and weight WTBL and compute the body mass index (BMI) with `compute_bmi()`. These values could also be obtained from ADVS if available. Baseline lab values could also be derived from LB or ADLB in a similar manner.

```
# Derive additional baselines from VS
adpc_baselines <- adpc_aseq %>%
  derive_vars_merged(
    dataset_add = vs,
    filter_add = VSTESTCD == "HEIGHT",
    by_vars = exprs(STUDYID, USUBJID),
    new_vars = exprs(HTBL = VSSTRESN, HTBLU = VSSTRESU)
  ) %>%
  derive_vars_merged(
    dataset_add = vs,
    filter_add = VSTESTCD == "WEIGHT" & VSBLFL == "Y",
    by_vars = exprs(STUDYID, USUBJID),
    new_vars = exprs(WTBL = VSSTRESN, WTBLU = VSSTRESU)
  ) %>%
  mutate(
    BMIBL = compute_bmi(height = HTBL, weight = WTBL),
    BMIBLU = "kg/m^2" ) # dim(adpc_baselines) [1] 3852   79

adpc_baselines %>% select(USUBJID,HTBL,HTBLU,WTBL,WTBLU,BMIBL,BMIBLU,BASETYPE,ATPT,AVAL) %>% he
```

```
# A tibble: 10 x 10
   USUBJID      HTBL HTBLU  WTBL WTBLU BMIBL BMIBLU BASETYPE       ATPT     AVAL
   <chr>       <dbl> <chr> <dbl> <chr> <dbl> <chr>  <chr>          <chr>   <dbl>
 1 01-701-1028  178. cm     99.3 kg     31.4 kg/m^2 Day 1 Baseline Pre-d~  0
 2 01-701-1028  178. cm     99.3 kg     31.4 kg/m^2 Day 1 Baseline Dose   54
 3 01-701-1028  178. cm     99.3 kg     31.4 kg/m^2 Day 1 Baseline 5 Min~  0.102
 4 01-701-1028  178. cm     99.3 kg     31.4 kg/m^2 Day 1 Baseline 30 Mi~  0.547
```

```
 5 01-701-1028  178. cm     99.3 kg     31.4 kg/m^2 Day 1 Baseline 1h Po~  0.925
 6 01-701-1028  178. cm     99.3 kg     31.4 kg/m^2 Day 1 Baseline 1.5h ~  1.19
 7 01-701-1028  178. cm     99.3 kg     31.4 kg/m^2 Day 1 Baseline 2h Po~  1.37
 8 01-701-1028  178. cm     99.3 kg     31.4 kg/m^2 Day 1 Baseline 4h Po~  1.68
 9 01-701-1028  178. cm     99.3 kg     31.4 kg/m^2 Day 1 Baseline 0-6h ~ 24.9
10 01-701-1028  178. cm     99.3 kg     31.4 kg/m^2 Day 1 Baseline 6h Po~  1.76
```

Add the ADSL variables

If needed, the other ADSL variables can now be added:

```
# Add all ADSL variables
adpc <- adpc_baselines %>%
  derive_vars_merged(
    dataset_add = select(adsl, !!!negate_vars(adsl_vars)),
    by_vars = exprs(STUDYID, USUBJID)
  ) # dim(adpc) [1] 3852  123
```

Adding attributes to the ADPC file will be discussed below. We will now turn to the Population PK example.

Programming Population PK (ADPPK) Analysis Data

The Population PK Analysis Data (ADPPK) follows the CDISC Implementation Guide (https://www.cdisc.org/standards/foundational/adam/basic-data-structure-adam-poppk-implementation-guide-v1-0). The programming workflow for Population PK (ADPPK) Analysis Data is similar to the NCA Programming flow with a few key differences. Population PK models generally make use of nonlinear mixed effects models that require numeric variables. The data used in the models will include both dosing and concentration records, relative time variables, and numeric covariate variables. A DV or dependent variable is often expected. This is equivalent to the ADaM AVAL variable and will be included in addition to AVAL for ADPPK. The ADPPK file will not have the duplicated records for analysis found in the NCA.

Here are the relative time variables we will use for the ADPPK data. These correspond to the names in the forthcoming CDISC Implementation Guide.

Variable	Variable Label
NFRLT	Nominal Rel Time from First Dose
AFRLT	Actual Rel Time from First Dose
NPRLT	Nominal Rel Time from Previous Dose
APRLT	Actual Rel Time from Previous Dose

Find First Dose `ADPPK`

References

Creating a BDS Time-to-Event ADaM

Analysis Data Model Implementation Guide for Non-compartmental Analysis Input Data Version 1.0 (Final)