

An Intuitive Way to Derive the Last Known Alive Date in R

March 19, 2025

Marckenley Mercie
Diana Tai

Agenda

1. Introduction
2. Derivation Assumptions
3. Deriving LSTALVDT
4. Further Automation
5. Enhancements and Considerations

Introduction

The Last Known Alive Date (LSTALVDT) is a critical date variable used in many study endpoints. Example use cases would be:

- Time to event analysis parameters such as Overall Survival (OS)
- Analysis Date derivations such ASTDT, AENDT in adverse event and concomitant mediations analysis datasets

Why use R?

- Open-source with a rich ecosystem of packages
- Easy to write clear, and reusable functions
- Opportunities to create dynamic and interactive solutions with Shiny, R's web application framework

We present two ways to derive LSTALVDT

- A collection of R functions that can be incorporated into standard R scripts for analysis datasets
- An interactive Shiny app that allows programmers to calculate and review LSTALVDT on-the-fly

Derivation Assumptions

The algorithm for the last known alive date will vary across the industry. Let's assume the following derivation rule:

For subjects who died, and death date is complete, set LSTALVDT to the numeric value of the date part of the death date. For all other subjects, collect complete dates from all study related or follow-up procedures where a subject is determined to be alive. Derive LSTALVDT as the maximum date among all complete dates. If derived LSTALVDT > database cutoff/lock date, then set LSTALVDT to the database cutoff/lock date

Based on this algorithm, two key assumptions were made:

1. SDTM-compliant input data
2. All date variables involved in the calculation are to follow ISO 8601 format

***Note: in this implementation partial dates are excluded from the calculation**

Deriving LSTALVDT

The derivation can be split into three main steps

Convert date columns to numeric format



Find the maximum between date columns for each domain by subject



Consolidate the resulting datasets and find the overall maximum date per subject

Deriving LSTALVDT - General Programming strategy

- **Efficient Data Import**

- Use `lapply(x, haven::read_sas)` to import all SDTM datasets in one step, where `x` is a vector of file paths
- Name each index of the resulting list with its corresponding SDTM domain name

- **Leveraging List Objects**

- List objects in R are flexible to use. Especially when combined with map functions from the *purrr* package, which can apply functions to multiple elements
- Named data lists allow the use of `purrr::imap` (the index map function) to iterate over the list
 - Useful for computing functions on both the value and position of an element

▼ sdtm_list

▶ ae

▶ ce

▶ cm

▶ co

▶ cv

▶ dd

▶ dm

▶ ds

list [60]

List of length 60

list [1248 x 36] (S3: tbl_df, tb A tibble with 1248 rows and 36 columns

list [12 x 10] (S3: tbl_df, tbl, c A tibble with 12 rows and 10 columns

list [2277 x 22] (S3: tbl_df, tb A tibble with 2277 rows and 22 columns

list [1425 x 22] (S3: tbl_df, tb A tibble with 1425 rows and 22 columns

list [122 x 21] (S3: tbl_df, tbl, A tibble with 122 rows and 21 columns

list [6 x 12] (S3: tbl_df, tbl, dā A tibble with 6 rows and 12 columns

list [124 x 23] (S3: tbl_df, tbl, A tibble with 124 rows and 23 columns

list [1361 x 12] (S3: tbl_df, tb A tibble with 1361 rows and 12 columns

Deriving LSTALVDT - Parameter File

- What if dates that contribute to the last known alive date are distributed across 10 or more domains?
- An input or parameter file containing the domains, dates, and filter conditions can be utilized iteratively. However, the creation of this file imposes a manual burden
- The parameter file can be created programmatically by dynamically parsing out all dates
- The parameter file can be structured in a form of a table containing the following columns:
 - *Dataset*: SDTM domains
 - *Dates*: the date columns associated with a particular domain
 - *Condition*: R syntax row filtering condition required for domain

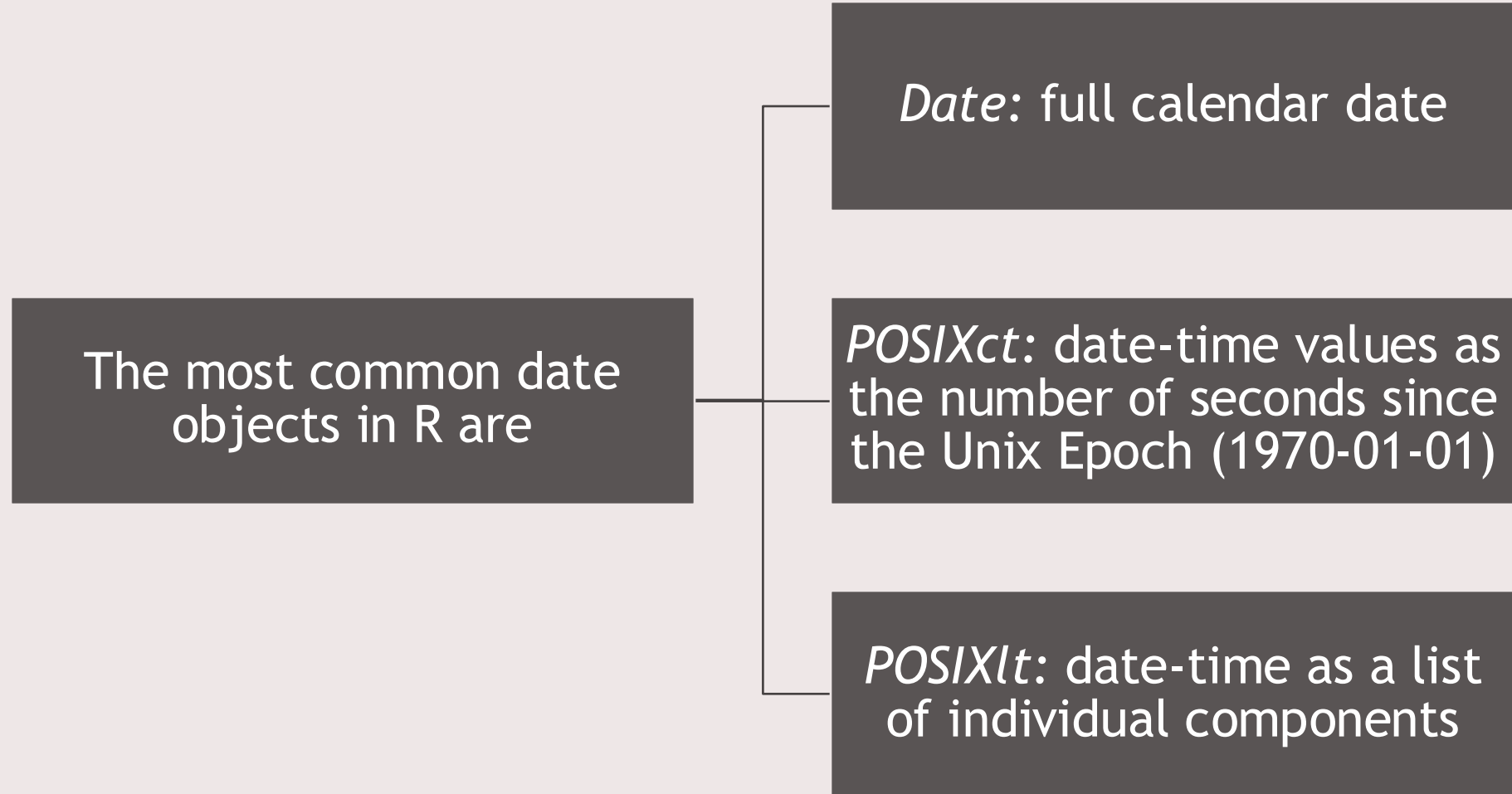
Deriving LSTALVDT - Parameter File (cont.)

- It is important to understand that not all domains and date variables contribute to the last known alive date
 - Therefore, the file is outputted as an Excel file
 - Once edited, the file can be read back into the R environment with *readr::read_excel*

	A	B	C
1	Dataset	Dates	Condition
2	ae	AESTDTC, AEENDTC	
3	ce	CESTDTC	
4	cm	CMSTDTC, CMENDTC	
5	cv	CVDTC	
6	dd	DDDTC	
7	dm	RFSTDTC, RFENDTC, RFXSTDTC, RFICDTC, RFPENDTC, DTHDTC	
8	ds	DSSTDTC	
9	eg	EGDTC	
10	ex	EXSTDTC	
11	faae	FADTC	

Example Parameter File

Deriving LSTALVDT - Date Objects in R



Deriving LSTALVDT - Converting Dates in R

- To derive the LSTALVDT, the date part of the ISO 8601 date is required
- Parsing out the date part is relatively easy using the *as.Date* function
 - Instead apply *as.Date* to character dates that are ≥ 10 , i.e.,
`ifelse(nchar(x) >= 10, as.Date(x, origin = "1960-01-01"), NA)`
 - R uses the standard Unix epoch "1970-01-01". Set the origin to "1960-01-01" to be consistent with the SAS system
- *as.Date* can be applied to multiple columns using `dplyr::mutate(dplyr::across())`
- The date columns are converted to numeric which can then be used in standard aggregation functions like *max* and *min*

Deriving LSTALVDT - Maximum Between Dates

- To take the maximum between dates do the following:
 - Group the data by subject with `dplyr::group_by`
 - Use `dplyr::summarise` along with the `max` function to find the maximum between multiple date columns
 - In `max`, specify the `na.rm` argument to `TRUE` so that missing values are ignored

	USUBJID	max_date	source_data
1	MODELSTUDY-01-001-26028	2023-12-06	CM
2	MODELSTUDY-01-001-26033	2023-03-06	CM
3	MODELSTUDY-01-001-26034	2023-09-30	CM
4	MODELSTUDY-01-001-26093	2023-11-08	CM
5	MODELSTUDY-01-001-26094	2023-10-12	CM
6	MODELSTUDY-01-001-26095	2023-12-07	CM
7	MODELSTUDY-01-001-26096	2024-01-26	CM
8	MODELSTUDY-01-001-26097	2024-01-07	CM
9	MODELSTUDY-01-001-26109	2024-01-31	CM
10	MODELSTUDY-01-001-26110	2024-01-01	CM
11	MODELSTUDY-01-001-26114	2023-11-11	CM

Deriving LSTALVDT - TO INFINITY AND BEYOND!

- Suppose we have the following date columns: DATE1, DATE2, ...DATEN. What happens when all dates are missing for a subject?
 - When *na.rm = TRUE*, R will assign -Inf as the value
- To avoid any calculation issues later, the *is.infinite* function can be used alongside with *dplyr::filter* to remove records with -Inf as the value

	USUBJID	max_date	source_data
1	MODEISTUDY-01-0001	2024-01-05	PP
2	MODEISTUDY-01-0002	-Inf	PP



Deriving LSTALVDT - Putting Everything Together

- The date conversions and finding the maximum date can be applied to every dataset in the data list using the *purrr::imap* function
- However, we need to also pass the names of the date columns and filter conditions to their corresponding dataset
 - To do this, read in the parameter file and use its columns as input
 - *purrr* map functions allow us to execute multiple statements for each element in a list, provided that the data and dataset name are utilized in the statements
 - The names of the data list correspond to the names of *Dataset* column in the parameter file. Therefore, we can link each element to its dates and filter conditions

Deriving LSTALVDT - Putting Everything Together (cont.)

```
dates_formatted <- sdtm_list |>
  purrr::imap(~ {
    dates <- param_df$Dates[param_df$Dataset == .y]
    convert_all_dates(.x, dates)
  })

max_dates <- dates_formatted |>
  purrr::imap(~ {
    dates <- param_df$Dates[param_df$Dataset == .y]
    filter_cond <- param_df$Condition[param_df$Dataset == .y]
    find_max_date(.x, .y, dates, filter_cond)
  }) |>
  dplyr::bind_rows()
```

Deriving LSTALVDT - Putting Everything Together (cont.)

- To finally derive the last known alive, take the minimum of *max(max_date)* and the cutoff/database lock date
- Then if a subject has a death date, set LSTALVDT to the death date

	▲ USUBJID	▼ LSTALVDT
1	MODELSTUDY-01-001-26028	2024-02-06
2	MODELSTUDY-01-001-26033	2024-01-19
3	MODELSTUDY-01-001-26034	2024-01-17
4	MODELSTUDY-01-001-26093	2024-01-31
5	MODELSTUDY-01-001-26094	2024-01-18
6	MODELSTUDY-01-001-26095	2024-01-31
7	MODELSTUDY-01-001-26096	2024-02-02
8	MODELSTUDY-01-001-26097	2024-02-06

Further Automation

- An interactive R Shiny containing all of the previously discussed functions can streamline the process of LSTALVDT derivation
- The application is separated into two tabs/modules:
 1. Parameter module: generates the parameter file
 2. LSTALVDT module: completes last known alive date derivation
- Benefits of Shiny application:
 - No knowledge of R is required to use the app
 - Can easily click to choose locations to upload data from and download outputs to
 - Interactivity allows the user to edit the parameter file within the app, without needing to export to Excel (although the option still exists)



Shiny Application - Parameter File Creation

LSTALVDT App

Home

Parameter File Creation

LSTALVDT Derivation

File Upload

Select an SDTM data folder

~/Projects/LSTALVDT/data

Upload Data

Double-click the "Dates" or "Condition" columns to edit the table.

Then hit Ctrl+Enter to finish editing, or Esc to cancel.

Parameter File Download

Select a save location

~/Projects/LSTALVDT

Download File

1. Select folder to upload SDTM datasets

2. Parameter file will generate

3. Option to edit parameters within the app

4. Download parameters as Excel file

Parameters

Show 25 entries

	Dataset	Dates	Condition
1	ae	AEDTC, AESTDTC, AEENDTC	
2	cm	CMDTC, CMSTDTC, CMENDTC	
3	dm	RFSTDTC, RFENDTC, RFXSTDTC, RFXENDTC, RFICDTC, RFPENDTC, DTHDTC, DMDTC	
4	ds	DSTDTC, DSSTDTC	
5	ex	EXSTDTC, EXENDTC	
6	se	SESTDTC, SEENDTC	

Showing 1 to 6 of 6 entries

Previous

1

Next

Shiny Application - LSTALVDT Derivation

LSTALVDT App

Home

Parameter File Creation

LSTALVDT Derivation

1. Upload parameter file and SDTM datasets*

File Upload

Select a Parameter file

~/Projects/LSTALVDT

Upload File

Select an SDTM data folder

~/Projects/LSTALVDT/data

Upload Data

If files were uploaded in the previous tab, proceed directly to "Derive LSTALVDT".

Database Cutoff Date

2024-02-07

Death Date Variable

DTHDTC

Derive LSTALVDT

2. Click the button to run derivation

LSTALVDT File Download

Select a save location

~/Projects/LSTALVDT

Download File

4. Download results as .xpt file

Parameters

LSTALVDT

Show 10 entries

Search:

	USUBJID	LSTALVDT
	All	All
1		2014-07-02
2		2013-02-18
3		2014-01-14
4		2014-09-15
5		2014-12-30
6		2013-07-28
7		2013-12-27
8		2014-07-09
9		2013-02-22
10		2013-05-20

Showing 1 to 10 of 306 entries

Previous12345...31Next

3. Results of LSTALVDT derivation

*Note: If the previous tab was used, the upload step can be skipped, as the parameter file and SDTM datasets will have been saved in the background

Enhancements and Considerations

- Consider the scenario where a subject has multiple events and one of the events happened in a year later from other events. Since the date is partial our implementation would not capture it
 - Date imputations should be considered depending on the risk
- Traceability columns would be a great help to make sure that LSTALVDT was derived correctly
 - Columns that show the domain, the date variable which the maximum came from, can help identify potential mishaps

USUBJID	AEDECOD	AESTDTC
MODELSTUDY-01-100101	Prostate Cancer	2022-01-05
MODELSTUDY-01-100101	Wound	2023-02

Thank you