Paper SM07

Interpreting Data Cutoff Methods Using R

Lusine Barseghyan, STATECS, Yerevan, Armenia
Vahagn Barseghyan, STATECS, Yerevan, Armenia

ABSTRACT

To safeguard participants in clinical trials, regular safety profile analysis is essential, relying on precise data cut-off (DCO) management. The R programming language offers robust solutions for clinical programmers, featuring accessible functions and libraries. This paper proposes an efficient DCO implementation method using R. A dedicated package applies predefined DCO methods to specified datasets and variables, including date variables and those affected by DCO. Users can define preferred methods and variables via a customizable template or utilize separate package functions. This approach streamlines DCO execution, ensuring compliance and accelerating safety profile analysis, thus enhancing clinical trial management efficiency.

INTRODUCTION

The implementation of data cutoffs, or Data Cutoff (DCO), can often be a complex and time-consuming process. Traditionally, using SAS® for DCO can be particularly labor-intensive, as it requires going through each raw dataset individually to establish the cutoff, making the process both cumbersome and prone to error. This paper addresses the challenges associated with DCO by presenting a simplified input metadata file designed to streamline the data cutoff procedure. We provide an example using an Excel file to illustrate this process. Additionally, we will demonstrate the results obtained from executing a user-friendly function from an R library, highlighting the recent trend of using R over SAS in clinical trials. This shift reflects the growing demand for flexible and efficient data analysis tools, ultimately showcasing how our approach can enhance efficiency and ease of use for researchers.

STEP 1: CREATING DCO METADATA

Initially, the user must create an input metadata file to compile information regarding the raw datasets and their corresponding variables, as well as the methodologies for implementing the Data Cutoff (DCO) for each dataset. This metadata file serves as a key guide, outlining the structure and requirements needed for the DCO process.

This step is indispensable in the implementation of DCO, as it is critical for users to identify the datasets and variables that will be affected by the DCO in any given study. In this paper, we utilize an Excel (.xlsx) format to ensure that the file remains as user-friendly as possible. The selection of Excel facilitates straightforward data entry and editing, enabling users to modify the file without the need for advanced programming expertise. The metadata file should encompass the following key components:

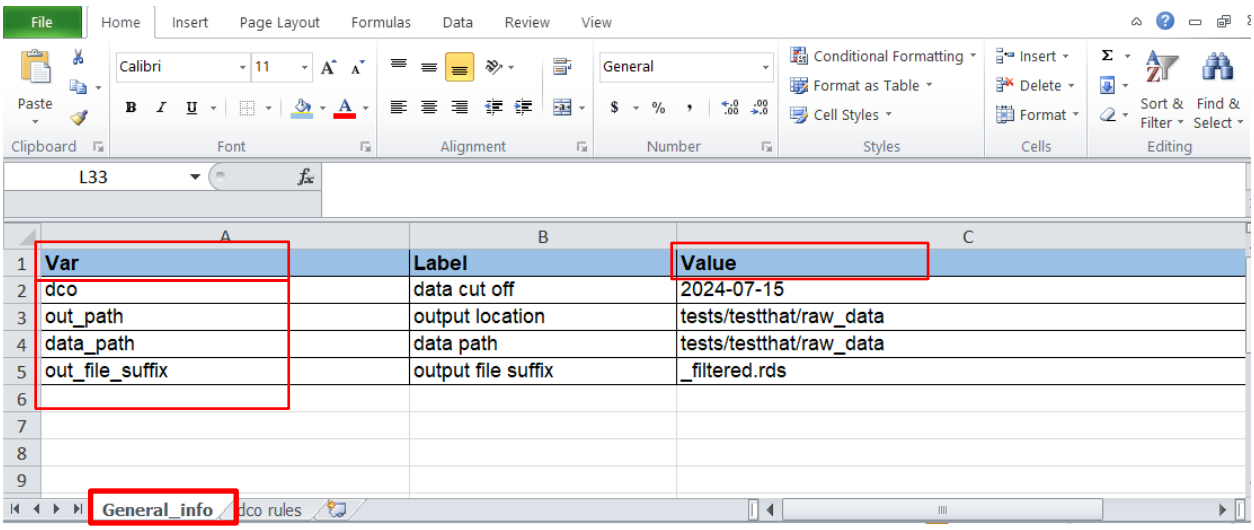
GENERAL INFORMATION TAB

- **DCO Date:** The specific date designated for the Data Cutoff, crucial for delineating the timeframe of data collection.
- **Output Location:** The designated directory where the results of the DCO process will be saved.
- **Data Path:** The file path to the raw data sources, which is essential for the DCO function to locate and process the relevant datasets.
- **Output File Suffix**: The suffix that will be appended to the output files, including new file extension, aiding in the identification and organization of results generated from the DCO process.

DCO RULES TAB

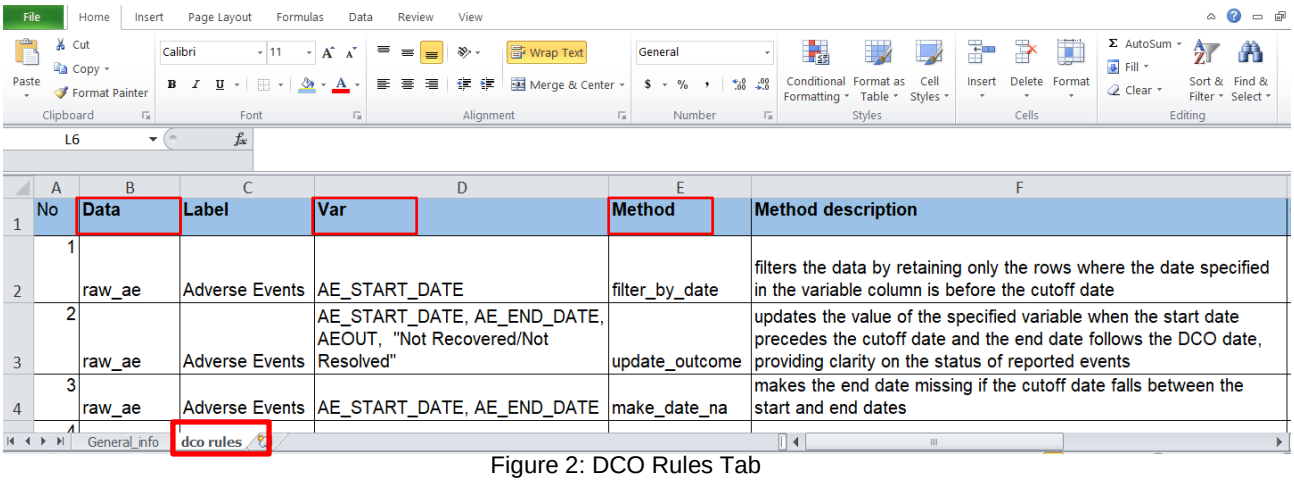
- **Dataset Names:** A clear designation of each raw dataset slated for DCO processing.
- **Variable Names:** Corresponding variables within each dataset that are relevant to the cutoff implementation.
- **DCO Methodologies:** The name of an R function from the package designated for specific actions with the given variables. Some of the available functions include filter_by_date, update_outcome, and make_date_na, which will be described in detail in the R documentation and further illustrated in the following example in this paper.

The following examples illustrate the mandatory sheets required in the Excel file. The highlighted fields must retain their default names, while other fields should be modified by the user to fit their specific needs.



Var	Label	Value
dco	data cut off	2024-07-15
out_path	output location	tests/testthat/raw_data
data_path	data path	tests/testthat/raw_data
out_file_suffix	output file suffix	_filtered.rds

Figure 1: General Information Tab



No	Data	Label	Var	Method	Method description
1	raw_ae	Adverse Events	AE_START_DATE	filter_by_date	filters the data by retaining only the rows where the date specified in the variable column is before the cutoff date
2	raw_ae	Adverse Events	AE_START_DATE, AE_END_DATE, AEOUT, "Not Recovered/Not Resolved"	update_outcome	updates the value of the specified variable when the start date precedes the cutoff date and the end date follows the DCO date, providing clarity on the status of reported events
3	raw_ae	Adverse Events	AE_START_DATE, AE_END_DATE	make_date_na	makes the end date missing if the cutoff date falls between the start and end dates

Figure 2: DCO Rules Tab

By systematically organizing this information within an Excel file, users can optimize the DCO process, thereby minimizing the likelihood of errors and enhancing overall efficiency. Once this essential step is accomplished, the template can serve as a valuable reference, allowing for modifications and adaptations to suit various studies within an organization. Moreover, if an organization adheres to consistent structures and variable conventions, they can develop this template once and apply it across all studies, thereby further streamlining the DCO process.

STEP 2: EXECUTION IN R

Upon completion of the input metadata file, users may proceed to execute the function from the R library, which will automate the DCO process in accordance with the specifications delineated in the metadata file. This approach not only streamlines the workflow but also significantly mitigates the time and effort typically associated with manual DCO implementation in conventional software such as SAS.

The only argument required for the R function ***execute_dco,*** is the path to the metadata file, which means users do not need to have any prior knowledge of R programming to utilize this functionality. However, users should familiarize themselves with the specific functions within the package. In this section, we will describe the specific methods available within the R package that facilitate the Data Cutoff (DCO) process:

- **filter_by_date:** This function filters the data by retaining only the rows where the date specified in the variable column is before the cutoff date.

- **update_outcome:** This function updates the value of the specified variable when the start date precedes the cutoff date and the end date follows the DCO date, providing clarity on the status of reported events.
- **make_date_na:** This function makes the end date missing if the cutoff date falls between the start and end dates.

These examples are quite specific and are intended to introduce common scenarios and usage of the R functions in the context of clinical trials. They serve as foundational illustrations, demonstrating how researchers can apply these tools to manage their data effectively while ensuring compliance with DCO requirements. In the following sections, we will apply these methods to the metadata created earlier and showcase the results obtained after executing the *execute_dco* function.

EXAMPLE 1: FILTER_BY_DATE

This function requires only one argument: the date that will be compared to the DCO date.
In the example below, with a DCO date of 2024-06-25, rows 2, 4, 5, and 6 will be removed from the dataset.

	SUBJECT_ID	AE_TERM	AE_START_DATE	AE_END_DATE	AE_SEVERITY
1	SUBJ001	Fatigue	2024-06-20	2024-08-10	Mild
2	SUBJ001	Anxiety	2024-07-04	2024-08-06	Severe
3	SUBJ002	Headache	2024-06-20	2024-08-09	Moderate
4	SUBJ002	Abdominal Pain	2024-06-30	2024-08-09	Moderate
5	SUBJ002	Dizziness	2024-07-08	2024-08-09	Mild
6	SUBJ003	Insomnia	2024-07-06	2024-08-06	Severe

The output from the filter_by_date function is displayed below.

	SUBJECT_ID	AE_TERM	AE_START_DATE	AE_END_DATE	AE_SEVERITY
1	SUBJ001	Fatigue	2024-06-20	2024-08-10	Mild
2	SUBJ002	Headache	2024-06-20	2024-08-09	Moderate
3	SUBJ005	Abdominal Pain	2024-06-02	2024-08-07	Moderate
4	SUBJ005	Dizziness	2024-06-13	2024-08-06	Mild

EXAMPLE 2: UPDATE_OUTCOME

This function accepts four arguments: the start date, end date, the variable to be updated, and the value to assign to that variable. In the example below, with a DCO date of 2024-08-07, for rows 1, 3, 4, and 5, the cutoff date falls between the start and end dates.

	SUBJECT_ID	AE_TERM	AE_START_DATE	AE_END_DATE	AE_SEVERITY	AE_RELATIONSHIP	AE_OUTCOME
1	SUBJ001	Fatigue	2024-06-20	2024-08-10	Mild	Possibly Related	Not Recovered
2	SUBJ001	Anxiety	2024-07-04	2024-08-06	Severe	Not Related	Recovered
3	SUBJ002	Headache	2024-06-20	2024-08-09	Moderate	Not Related	Recovered
4	SUBJ002	Abdominal Pain	2024-06-30	2024-08-09	Moderate	Related	Not Recovered
5	SUBJ002	Dizziness	2024-07-08	2024-08-09	Mild	Possibly Related	Not Recovered
6	SUBJ003	Insomnia	2024-07-06	2024-08-06	Severe	Related	Not Recovered

Consequently, the AE Outcome will be updated to "Not Recovered/Not Resolved" at the time of the DCO.

	SUBJECT_ID	AE_TERM	AE_START_DATE	AE_END_DATE	AE_SEVERITY	AE_RELATIONSHIP	AE_OUTCOME	AE_SERIOUS
1	SUBJ001	Fatigue	2024-06-20	2024-08-10	Mild	Possibly Related	Not Recovered/Not Resolved	No
2	SUBJ001	Anxiety	2024-07-04	2024-08-06	Severe	Not Related	Recovered	No
3	SUBJ002	Headache	2024-06-20	2024-08-09	Moderate	Not Related	Not Recovered/Not Resolved	No
4	SUBJ002	Abdominal Pain	2024-06-30	2024-08-09	Moderate	Related	Not Recovered/Not Resolved	No
5	SUBJ002	Dizziness	2024-07-08	2024-08-09	Mild	Possibly Related	Not Recovered/Not Resolved	No
6	SUBJ003	Insomnia	2024-07-06	2024-08-06	Severe	Related	Not Recovered	Yes

EXAMPLE 3: MAKE_DATE_NA

Lastly, the make_date_na function takes two arguments: the start and end dates. In the example below, with a DCO date of 2024-08-07, rows 1, 3, 4, and 5 are ongoing at the time of the DCO, so the variable AE_END_DATE will be updated to N/A.

	SUBJECT_ID	AE_TERM	AE_START_DATE	AE_END_DATE	AE_SEVERITY	AE_RELATIONSHIP	AE_OUTCOME	AE_SERIOUS
1	SUBJ001	Fatigue	2024-06-20	2024-08-10	Mild	Possibly Related	Not Recovered	No
2	SUBJ001	Anxiety	2024-07-04	2024-08-06	Severe	Not Related	Recovered	No
3	SUBJ002	Headache	2024-06-20	2024-08-09	Moderate	Not Related	Recovered	No
4	SUBJ002	Abdominal Pain	2024-06-30	2024-08-09	Moderate	Related	Not Recovered	No
5	SUBJ002	Dizziness	2024-07-08	2024-08-09	Mild	Possibly Related	Not Recovered	No
6	SUBJ003	Insomnia	2024-07-06	2024-08-06	Severe	Related	Not Recovered	Yes

The output from the make_date_na function is displayed below.

	SUBJECT_ID	AE_TERM	AE_START_DATE	AE_END_DATE	AE_SEVERITY	AE_RELATIONSHIP	AE_OUTCOME	AE_SERIOUS
1	SUBJ001	Fatigue	2024-06-20	*NA*	Mild	Possibly Related	Not Recovered	No
2	SUBJ001	Anxiety	2024-07-04	2024-08-06	Severe	Not Related	Recovered	No
3	SUBJ002	Headache	2024-06-20	*NA*	Moderate	Not Related	Recovered	No
4	SUBJ002	Abdominal Pain	2024-06-30	*NA*	Moderate	Related	Not Recovered	No
5	SUBJ002	Dizziness	2024-07-08	*NA*	Mild	Possibly Related	Not Recovered	No
6	SUBJ003	Insomnia	2024-07-06	2024-08-06	Severe	Related	Not Recovered	Yes

CONCLUSION

This paper presents an efficient approach to implementing Data Cutoff (DCO) processes in clinical trials using R. By introducing a user-friendly metadata file, we enable researchers to easily gather and organize critical information about raw datasets and their variables.

The *execute_dco* function automates the DCO process, significantly reducing the complexity traditionally associated with manual management in software like SAS. With specific R functions - filter_by_date, make_date_na, and update_outcome - we provide practical solutions for common DCO scenarios, enhancing data accuracy and regulatory compliance.

Our examples demonstrate the effectiveness of this methodology in simplifying workflows and minimizing errors. However, there is ample room for improvement, such as the incorporation of custom functions tailored to specific study needs, allowing for even greater flexibility and adaptability in diverse research contexts.

As R becomes increasingly popular for data analysis, the techniques outlined here can be readily adapted across studies, improving efficiency and contributing to reliable, reproducible clinical trial outcomes.

.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Lusine Barseghyan
STATECS
Email: lusine.barseghyan@statecs.com
Website: www.statecs.com