Data-Cut-off (DCO) process as a tool in R

A capstone presented for the degree of
Bachelor in Data Science

Seda Bayadyan

Department Name: The Zaven and Sonia Akian College of Science and Engineering
University Name: American University of Armenia
Under the Supervision of: Erik Torosyan
Country: Armenia
Date: May 2025

Contents

1 Abstract

Clinical trials depend on precise and prompt data cut-off (DCO) processes to ensure participant safety and uphold the integrity of trial results. Ongoing analysis of safety profiles during clinical studies, which is directly related to accurate DCO management, is an essential aspect of clinical trial operations. Historically, the Statistical Analysis System (SAS) has been the preferred tool for data processing and also for DCO management within clinical trials. However, with the emergence of open-source alternatives, the R programming language has surfaced as a practical option, providing a variety of robust functions and libraries designed specifically for clinical programmers.

This paper introduces an efficient method for implementing the DCO process using R, in contrast to traditional SAS-based approaches. The proposed method utilizes a dedicated R package, which provides predefined DCO methods that can be applied to datasets, specifically targeting date variables and those affected by the DCO. Through a customizable template, users can easily apply their preferred methods already defined in the package, or alternatively, apply individually created functions on already existing ones. This new approach to DCO execution not only ensures compliance with industry standards but also accelerates the analysis of safety profiles.

We will discuss the advantages of using R for DCO management, especially in comparison with SAS, highlighting its flexibility, ease of use, and integration with other data analysis processes. Ultimately, this method improves the efficiency of clinical trial management by reducing manual interventions and ensuring timely decision-making based on reliable, up-to-date data.

2 Literature Review

2.1 Importance of Data Cut-Off (DCO) in Clinical Trials

Data Cut-Off (DCO) is a pivotal process in clinical trials, where it involves restricting data analysis to a predefined date. This ensures that the dataset is static, complete, consistent, and ready for analysis. This efficiency is crucial for expediting the drug development process and making new treatments available to patients.It safeguards data integrity, promotes unbiased analysis, ensures regulatory compliance, and facilitates timely reporting, all of which are essential for generating reliable evidence to inform medical practice(European Medicines Agency, 2024).

2.2 Traditional Use of SAS in DCO Processes

SAS (Statistical Analysis System) has long been the dominant tool for data management, transformation, and reporting in clinical trials, including for data cut-off (DCO) activities. Its strong integration with clinical standards such as CDISC (SDTM, ADaM) and its widespread regulatory acceptance have made it stable in pharmaceutical programming environments.

In typical DCO workflows, SAS is used to filter subject-level and event-level data based on predefined cut-off dates, ensure consistency across domains, and generate analysis-ready datasets. Many organizations also maintain audit trails and automated macros in SAS to enforce reproducibility and compliance.

However, reliance on SAS brings several limitations:

- **Cost and Licensing**: SAS is a proprietary, licensed software, often incurring significant expenses for both academic institutions and small organizations.
- **Limited Flexibility**: While powerful, SAS can be verbose, and its macro language lacks some of the functional and object-oriented features available in modern languages like R. Apart from some default packages available in R, more than 6000 packages available online depending upon techniques as Bayesian, Survival, timeseries, Adaptive designs, experimental designs, etc.
- **Reproducibility Barriers**: Although reproducibility is possible in SAS, it often requires complex macro-level solutions.

These limitations have led many data science and biostatistics teams to explore open-source alternatives like R, especially for custom or exploratory workflows where flexibility, collaboration, and automation are critical(Prajapati & Kumar, 2018).

2.3 Regulatory Compliance and Validation of R in Clinical Trials

While R is open-source and not directly validated by the FDA, organizations must validate their R installations and any packages they use to ensure compliance within their specific clinical environments (R Foundation for Statistical Computing, 2021).

The guidance highlights the importance of establishing Standard Operating Procedures (SOPs) for R's use in clinical trials, ensuring data integrity and reproducibility. As such, R's use in regulated environments is growing, thanks to this framework, which provides the necessary steps for maintaining compliance with regulatory standards while leveraging R's capabilities.

3 Overall DCO package and how it works

3.1 Step 1: Creating DCO Configuration File

The foundation of the DCO R package lies in its YAML-based configuration file, typically named `config.yaml`. Users get into this file in their working directory after defining our library and calling this function:

```r
# Setting DCO library
library(dco)

# Calling a function to create config.yaml file
create_config_file()
```

This file serves as a centralized metadata source that guides the entire data cut-off (DCO) process. It allows users to define essential components such as raw data paths, the desired DCO date, applicable transformation rules per dataset, and relevant variables to be considered during processing.

Creating this configuration file is a critical first step in implementing the DCO workflow, as it ensures that the tool has the necessary information to correctly interpret and transform datasets in line with study-specific requirements. It also promotes transparency and reproducibility, enabling consistent application of the cut-off logic across various studies or trials. YAML was chosen as the central configuration format for our DCO tool due to its clean, human-readable syntax. Additionally, YAML files integrate seamlessly with version control systems like Git, enabling clear tracking of changes, collaborative development, and enhanced reproducibility in clinical data workflows.

The `config.yaml` file is structured into three primary sections:

- **`general_info`**: Specifies key parameters such as the file path to raw datasets, output destination, suffix to be applied to filtered datasets and most improtantly DCO date.
- **`dco_rules`**: Maps each dataset to one or more transformation functions, encompassing both predefined functions (like filtering and modifying variables) and user-defined functions for enhanced adaptability.
- **`dat_variable_list`**: For each dataset, defines the date-related variables that influence DCO logic and the variables that are expected to change when the DCO is applied.

This structured approach enables the DCO package to function generically across studies, allowing users to customize the process simply by updating the configuration file—without modifying the core R functions.

Below is an illustrative example of a `config.yaml` file:

```yaml
# -----------------------
# General configuration
# -----------------------
general_info:
  input_path: "path/to/your/raw_data"      # <-- Set the input folder path
  output_path: "path/to/your/output_data"  # <-- Set the output folder path
  dco_date: "YYYY-MM-DD"                   # <-- Enter your cut-off date here
  out_file_suffix: "_filt.rds"             # <-- Suffix to append to filtered datasets

# -----------------------
# Rules for each dataset
# -----------------------
dco_rules:
  dataset_name:                            # <-- Set the dataset name that needs DCO process
    - method: method_name                  # <-- Set the method name
      var: [variables_affected]            # <-- Set variable names that need to be cutted


# -----------------------
# Date variable mapping
# -----------------------
dat_variable_list:
  dataset_name:                            # <-- Set the dataset name that needs DCO process
    - variable name                        # <-- Set date variable names
 yaml: content
```

3.2 Step 2: Performing the Data Cut-Off (DCO)

Upon completion of the configuration file, users can proceed to perform the DCO by calling the main function from the DCO R package. This function automates the entire cut-off process as defined in the YAML metadata file, streamlining what has traditionally been a manual and labor-intensive workflow in software such as SAS.

The core function, `perform_dco()`, requires only a single argument: the file path to the configuration YAML, which was created before and is already filled with requirements. This simplicity makes it accessible even to users with minimal R programming experience, promoting broader adoption and usability. Nevertheless, it is beneficial for users to understand the underlying helper functions that are invoked during execution, as they represent the modular logic of the package.

The package includes the following main functions:

- **`filter_by_date()`**
  This function filters the dataset by retaining only records where the date specified in the variable column occurs before the defined data cut-off (DCO) date, it covers different formats and also works on partial dates. Current workflow of function in case of partial dates is to impute as first day of the month for comparison, then leave partial dates with the same format as they were before, to not cause any confusion when manually checking data for partials.

- **`update_var()`**
  This function updates the values of specified variables based on a time window: if the start date precedes the DCO date and the end date follows it, the outcome is modified to reflect a more accurate event status around the time of cut-off.

- **`make_date_na()`**
  This function sets the end date to missing (`NA`) when the DCO date falls between the specified start and end dates, ensuring that post-cut-off data is excluded from analysis.

Together, these functions form the building blocks of a robust, reproducible, and regulation-compliant DCO process in R. If users will need any other functions for their specific study needs, they are welcome to add a new function for them and correctly implement it as a method.

3.3 Step 3: Illustrative Examples of DCO Functionality

The following examples are designed to reflect practical, real-world scenarios frequently encountered in clinical trial data management. These use cases highlight how the core R functions within the DCO package can be applied to ensure regulatory compliance and support efficient data handling.

Each example demonstrates a typical challenge in managing cut-off dates and how the relevant DCO method—such as `filter_by_date`, `update_var`, or `make_date_na`—can be used to address it effectively.

These foundational illustrations are applied to the metadata defined in the configuration file introduced earlier. We will show the output before and after executing the `perform_dco()` function to illustrate the transformations that occur during the automated DCO process. In examples presented below Data Cut-Off (DCO) date is set to “2024-07-27”.

1. This is Adverse Events(AE) raw data, here are filtered two subjects with several Adverse Events reported

Table 1: Filtered Data for Subjects SUBJ003 and SUBJ004 prior to DCO process

SUBJID	TERM	STARTDT	ENDDT	SEV	REL	AEOUT	SER
SUBJ003	Insomnia	20240608	20240706	Severe	Related	Not Recovered	Yes
SUBJ003	Nausea	17Jul2024	28-Jul-24	Moderate	Not Related	Recovered	No
SUBJ003	Headache	22Jul2024	7/26/2024 16:45	Mild	Possibly Related	Recovered	Yes
SUBJ004	Insomnia	7/5/2024	20240727	Severe	Not Related	Recovered	Yes
SUBJ004	Dizziness	20-Jul-24	29-07-2024	Mild	Possibly Related	Recovered	No

The package filters records to retain only those that occurred on or before this date. For Adverse Events (AE) data, there are specific nuances to consider. First, if an AE has both a start and end date, and the DCO date falls between them, the end date should be set to missing. Second, the AEOUT variable—which represents the outcome of the event and may include values such as “FATAL”, “NOT RECOVERED/NOT RESOLVED”, “RECOVERED/RESOLVED WITH SEQUELAE”, “RECOVERED/RESOLVED”, “RECOVERING/RESOLVING”, or “UNKNOWN”—should be updated accordingly. When the end date is removed due to the DCO falling within the AE period, the outcome should be set to reflect the subject’s status at the cut-off. In such cases, the outcome is updated to “NOT RECOVERED/NOT RESOLVED”.

Here is config.yaml file filled with information for AE dataset.

```yaml
general_info:
  input_path: "tests/testthat/raw_data"
  output_path: "tests/testthat/raw_data"
  dco_date: "2024-07-27"
  out_file_suffix: "_filt.rds"

dco_rules:
  raw_ae:
    - method: filter_by_date
      var: [STARTDT]

    - method: update_var
      var: [STARTDT, ENDDT, AEOUT]

    - method: make_date_na
      var: [STARTDT, ENDDT]
dat_variable_list:
  raw_ae:
```

```
  - STARTDT
  - ENDDT
```

And here is the final filtered dataset for the specified two subjects:

Table 2: Filtered Data for Subjects SUBJ003 and SUBJ004 after DCO process

SUBJID	TERM	STARTDT	ENDDT	SEV	REL	AEOUT
SUBJ003	Insomnia	2024-06-08	2024-07-06	Severe	Related	Not Recovered
SUBJ003	Nausea	2024-07-17	NA	Moderate	Not Related	Not Recovered/Not Resolved
SUBJ003	Headache	2024-07-22	2024-07-26 16:45:00	Mild	Possibly Related	Recovered
SUBJ004	Insomnia	2024-07-05	2024-07-27	Severe	Not Related	Recovered
SUBJ004	Dizziness	2024-07-20	NA	Mild	Possibly Related	Not Recovered/Not Resolved

As shown, end dates became missing for the ones which were after DCO date, and their AEOUT values became "NOT RECOVERED/NOT RESOLVED"

2. This one is Laboratory Test Results (LB) raw dataset, here is shown more diverse types of formats, with also partial dates and time components, let's look into small part of dataset:

Table 3: Filtered Data for 3 Subjects prior to DCO process

SUBJECT_ID	LB_TEST	LB_DATE	LB_RESULT	LB_UNIT
SUBJ002	Hemoglobin	6/24/2024	12.8	g/dL
SUBJ002	WBC	2024-06	7.1	x10^9/L
SUBJ003	Creatinine	27-Jul-24	1.1	mg/dL
SUBJ003	ALT	20240706T15:30	52.0	U/L
SUBJ003	AST	28-Jul-24	48.0	U/L
SUBJ004	Hemoglobin	202407	11.9	g/dL

Here in this dataset only filter_by_date() function is applied, because we do not need any other date variable to make missing, so we don't use make_date_na(), and similiarly for update_var() we don't need any variable value to update. Here are filtered 3 subjects after DCO:

Table 4: Filtered Data for 3 Subjects after DCO process

SUBJECT_ID	LB_TEST	LB_DATE	LB_RESULT	LB_UNIT
SUBJ002	Hemoglobin	2024-06-24	12.8	g/dL
SUBJ002	WBC	2024-06	7.1	x10^9/L
SUBJ003	Creatinine	2024-07-27	1.1	mg/dL
SUBJ003	ALT	2024-07-06 15:30:00	52.0	U/L
SUBJ004	Hemoglobin	202407	11.9	g/dL

So for SUBJID = "SUBJ003" row was deleted with LBTEST = "AST", because the date was 2024-07-28 but our DCO date is 2024-07-27, all others are outputted in format of yyyy-mm-dd for clearer view, and partial dates which fall before or on DCO date are kept.

4 Comparison of R and SAS

4.1 Background

SAS (Statistical Analysis System) has long been the dominant software environment for statistical programming in clinical trials. Its widespread adoption stems largely from historical factors, particularly the early endorsement and integration by regulatory bodies such as the U.S. Food and Drug Administration (FDA) and the European Medicines Agency (EMA). These agencies have traditionally accepted data and analyses performed using SAS formats, which created strong incentives for pharmaceutical companies and contract research organizations (CROs) to standardize their workflows around the SAS ecosystem. Furthermore, SAS offers built-in capabilities for meeting Good Clinical Practice (GCP) and 21 CFR Part 11 compliance requirements, reinforcing its reputation as a reliable and regulatory-aligned platform for managing clinical data. However, over the past decade, the landscape of clinical data science has evolved significantly. Modern clinical research increasingly requires interoperability, transparency, faster iteration cycles, and integration with diverse data types, such as real-world evidence and genomics. In this context, R has emerged as a compelling alternative due to its open-source nature, extensive package ecosystem, active community support, and seamless integration with modern data science tools. As the pharmaceutical industry embraces principles of agile development and open science, R's flexibility, cost-effectiveness, and support for reproducible research have made it an attractive choice for both exploratory and regulatory analysis. This paradigm shift is further supported by industry initiatives such as the Pharmaverse—a collaborative ecosystem of validated R packages specifically designed for clinical trial workflows. With increasing regulatory acceptance of R-based submissions, the transition from SAS to R is not only technically feasible but strategically aligned with broader trends in digital transformation across the life sciences sector(ProCogia, 2024).

4.2 Business and Strategic Considerations

4.2.1 Cost Implications

One of the most significant considerations in the shift from SAS to R in clinical research is cost. SAS is a proprietary software that involves substantial licensing fees, which can become a major financial burden—especially for small-to-medium enterprises (SMEs), startups, academic institutions, and non-profit research organizations. Licensing SAS often requires annual subscriptions per user or per server, with additional fees for modules such as SAS/STAT, SAS/GRAPH, or Clinical Standards Toolkit, resulting in considerable operational expenses. In contrast, R is a completely free and open-source statistical environment, maintained by a global community of developers and statisticians. It can be installed and deployed at no cost across any number of users or environments—on personal machines, local servers, or cloud-based platforms. This drastically lowers the total cost of ownership and facilitates broader adoption, experimentation, and scalability, particularly for teams working with limited budgets or in early stages of digital transformation. Moreover, R's open-source nature allows for rapid development of new tools and methods, avoiding vendor lock-in and enabling teams to tailor solutions to their evolving needs without waiting for costly product updates or formal releases. For organizations striving to be both cost-efficient and scientifically agile, R provides a financially sustainable and technically flexible foundation for modern clinical analytics(Muenchen, 2020).

4.2.2 Human Resource Utilization

Human resource efficiency plays a critical role in the operational and financial success of data pipelines in clinical trials. SAS-based workflows, particularly those relying heavily on macro programming, often demand substantial developer involvement. Macros in SAS, while powerful, are also notoriously complex to maintain and debug—especially when dealing with large-scale studies, frequent data updates, or shifting sponsor requirements. Each re-run of the pipeline may introduce new edge cases or dependencies that require custom handling, resulting in an increased need for experienced SAS programmers. In practical terms, this complexity leads to longer development cycles, slower response times, and higher workforce costs. In contrast, R enables a more modular, reusable, and transparent approach to data processing. With R, business logic can be encapsulated in small, well-documented functions that are easy to test and extend. On average, normal difficulty updates take programmers to update in SAS approximately 40 hours, and doing it with R takes approximately 10 hours.

4.2.3 Industry Trends and Regulatory Acceptance

The clinical research industry is steadily shifting toward open-source tools like R, driven by the need for flexibility, transparency, and cost-efficiency. While SAS has historically dominated due to its early regulatory adoption, agencies such as the FDA and EMA now accept R-based submissions, provided that validation and documentation requirements are met. This trend is reinforced by initiatives like Pharmaverse as mentioned also before, which promotes validated, regulatory-grade R packages specifically designed for clinical trial workflows. These packages—such as admiral and tlg—follow CDISC standards and help ensure that open-source solutions can meet audit and compliance expectations, accelerating the transition to modern data science tools in regulated environments(Hector et al., 2023).

5 Conclusion

This capstone project demonstrated how transitioning from a SAS macro-based approach to a modular, R-based configuration-driven framework can streamline the data cut-off (DCO) process in clinical trials. By encapsulating key transformation steps into reusable functions and leveraging a flexible YAML configuration file, the proposed R package offers significant advantages in terms of maintainability, scalability, and turnaround time.

Beyond the technical improvements, this shift reflects broader trends in the industry: increasing openness to R by regulatory bodies, a growing emphasis on reproducibility, and a drive to reduce both software licensing and human resource costs. The adoption of tools like Pharmaverse reinforces the feasibility of R-based workflows in regulated environments and signals a promising future for open-source innovation in clinical data science.

For further actions, this tool can be more improved and handle more data cases, as with every dry-run there will be new cases to cover, but besides this, mostly faced issues during clinical studies during Data Cut-off process can be highlighted, and even more functions can be added for users to have wider types of methods to choose from, in this way user input will become minimal.

References

European Medicines Agency. (2024). *ICH Guideline for Good Clinical Practice E6(R2) Revision 2*. https://www.ema.europa.eu/en/documents/scientific-guideline/ich-guideline-good-clinical-practice-e6r2-step-5-revision-2_en.pdf

Hector, B., Sadler, A., & team. (2023). Pharmaverse: An open-source ecosystem for regulatory-grade clinical trial reporting. *PharmaSUG Europe 2023*. https://pharmaverse.org

Muenchen, R. A. (2020). The popularity of data science software. *R4Stats*. https://r4stats.com/articles/popularity/

Prajapati, K., & Kumar, P. (2018). Exploring use of r for clinical trials. *PharmaSUG 2018 Conference Proceedings*. https://pharmasug.org/proceedings/2018/SI/PharmaSUG-2018-SI12.pdf

ProCogia. (2024). Transitioning clinical research from SAS to r: An introduction to the pharmaverse ecosystem. *ProCogia Blog*. https://procogia.com/transitioning-clinical-research-from-sas-to-r-an-introduction-to-the-pharmaverse-ecosystem

R Foundation for Statistical Computing. (2021). *R: Regulatory compliance and validation issues*. https://www.r-project.org/doc/R-FDA.pdf