

Streamlining Clinical Data Analysis and Reporting with R: Best Practices, Tips and Techniques for Efficient Workflow Management and Code Optimization



Mahesh Divakaran*, Catalyst Clinical Research, Trivandrum, India
Vaibhavi Govind Bhide, Catalyst Clinical Research, Trivandrum, India
Akhil Vijayan, Catalyst Clinical Research, Trivandrum, India

.....

Agenda

INTRODUCTION

WHY R?

R FOR CLINICAL DATA ANALYSIS AND REPORTING



INTRODUCTION

Healthcare and life sciences are experiencing a surge in clinical trials.

This surge brings a significant influx of clinical data.

Clinical data is pivotal for evidence-based decision-making and trial validity.

Effective data analysis is essential for converting raw data into actionable knowledge.

Data analysis acts as a proactive sentinel, improving patient care and trial retention.

Proactive data analysis streamlines trials, reduces costs, and expedites product development.

Why R?



Open-Source Advantage:

SAS® is a widely used tool for clinical data analysis and reporting, but R is emerging as a powerful alternative. Notably, SAS is proprietary software, while R is open-source. This openness means that R is readily accessible, adaptable, and cost-effective for clinical research.



Statistical and Graphical Proficiency:

R is renowned for its capabilities in statistical computing and graphics. It offers a comprehensive suite of statistical techniques and graphical tools for clinical data analysis. R is especially well-suited for handling the complexity of modern clinical trial data.



Rich Library Ecosystem:

Developers have created numerous open-source libraries and packages tailored for clinical trial data analysis and reporting in R. These libraries cover a wide range of functions and features, making it a versatile choice for healthcare professionals.



Active User Community:

R boasts an active and vibrant user community. This active community means that you're likely to find solutions to your specific challenges and roadblocks, ensuring that your clinical data analysis proceeds smoothly.



Publication-Quality Outputs:

R enables the creation of well-designed, publication-quality plots and visualizations with ease. This is essential for conveying clinical research findings effectively to both scientific and non-technical audiences.



Dedicated Reporting Libraries:

R offers specialized libraries for reporting, supporting various report formats. This flexibility allows you to produce reports in the format that best suits your needs, from standard reports to more interactive and presentation-ready formats.



R for Clinical Data Analysis and Reporting



Versatility of R



Optimized
Workflows



Better Data
Manipulation
options



More Statistical
Tools with
Packages



Reproducible
Reports and
automated report
generation



Tackling Large
Datasets

Data Cleaning with R



Ensuring

Ensuring accuracy and completeness.



Identifying

Identifying errors, discrepancies, and missing values.



Making



LISTEN, DEVISE, **BREAKTHROUGH**



Example: Using dplyr for Filtering and Summarizing Data

```
library(dplyr)
library(formatters) # for simulated 'CDISC' Alike Data for Examples

# Assuming 'clinical_data' is your data frame
average_age <- ex_adsl %>%
  filter(ARM == "A: Drug X") %>%
  summarize(mean_age = mean(AGE, na.rm = TRUE))
average_age
# 'average_age' will contain the mean age of patients in Drug X

A tibble: 1 × 1

  mean_age
  <dbl>
1 33.76866
1 row
```



Example: Using tidyr for Data Tidying

```
# Load the tidyr library
library(tidyr)

# Create a sample data frame (similar to your 'wide_data')
wide_data <- data.frame(
  subject = 1:5,
  visit = c("Visit1", "Visit2", "Visit3", "Visit1", "Visit2"),
  var1 = c(12, 14, 10, 15, 19),
  var2 = c(5, 7, 8, 4, 9)
)

# Display the original 'wide_data' frame
print("Original 'wide_data' frame:")
print(wide_data)

# Use gather to reshape the data
tidy_data <- wide_data %>%
  gather(variable, value, -subject, -visit)

# Display the reshaped 'tidy_data' frame
print("\nReshaped 'tidy_data' frame:")
print(tidy_data)
```

Wider Data

```
print(wide_data)
  subject visit var1 var2
1      1 Visit1  12    5
2      2 Visit2  14    7
3      3 Visit3  10    8
4      4 Visit1  15    4
5      5 Visit2  19    9
```

```
print(tidy_data)
  subject visit variable value
1      1 Visit1    var1    12
2      2 Visit2    var1    14
3      3 Visit3    var1    10
4      4 Visit1    var1    15
5      5 Visit2    var1    19
6      1 Visit1    var2     5
7      2 Visit2    var2     7
8      3 Visit3    var2     8
9      4 Visit1    var2     4
10     5 Visit2    var2     9
```

Tidy Data



Example: Using stringr for String Manipulation

```
library(stringr)

# Assuming 'diagnosis_notes' is a character vector with patient diagnosis notes
pattern <- "cancer"
# Example diagnosis_notes data
diagnosis_notes <- c(
  "Patient has a history of lung cancer",
  "The MRI results indicate brain cancer",
  "No evidence of cancer found in the latest biopsy",
  "Patient's family history includes breast cancer",
  "Diagnosis: Stage 3 colon cancer",
  "The patient is at risk of skin cancer due to sun exposure",
  "Cancer screening results are pending"
)

# Now you can use the 'diagnosis_notes' vector in the R Markdown example

# Find all notes containing the word "cancer"
cancer_notes <- diagnosis_notes[str_detect(diagnosis_notes, pattern)]

# 'cancer_notes' now contains the relevant diagnosis notes

cancer_notes
```

```
cancer_notes
[1] "Patient has a history of lung cancer"      "The MRI results indicate brain cancer"
[3] "No evidence of cancer found in the latest biopsy"  "Patient's family history includes breast cancer"
[5] "Diagnosis: Stage 3 colon cancer"           "The patient is at risk of skin cancer due to sun exposure"
```



DATA ANALYSIS



Statistical Analysis



Reporting in R



Objective of Reporting



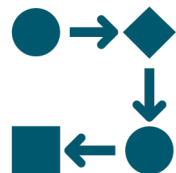
Inefficient R Code



The Flexibility of R



Best Practices for Efficient R Programming



Avoid Loops



Best Practices for Efficient R Programming



Keep R Version Updated

Update to the latest R versions for performance improvements and bug fixes.



Use Well-Maintained R Packages



Best Practices for Efficient R Programming



Profiling



Conclusion



Clinical trials' exponential growth yields a wealth of vital data for healthcare and life sciences.



Efficient data analysis is essential for sound decision-making and scientific validity.



Data analysis serves as a sentinel, enhancing patient care and cost savings.



R, a robust programming language, empowers clinical data analysis and reporting.



Profiling tools, code optimization, and various R packages streamline the workflow.



Rmarkdown and knitr facilitate reproducible reports, enhancing transparency.



Parallelization techniques using furr and future efficiently handle large datasets.

.....
Questions?



THANK YOU

LISTEN, DEVISE, **BREAKTHROUGH**