Paper CT08

Mixed Models & Measured Outcomes: R Tips from the Clinical Trenches

Keyur Shah, ICON PLC, Bangalore, India

ABSTRACT

Clinical trial analyses increasingly rely on longitudinal endpoints, complex data structures, and repeated measurements within subjects and investigative sites. While R provides a rich ecosystem of statistical modeling tools, practical challenges remain in scaling analyses, standardizing outputs, and maintaining traceability from analysis to submission-ready reports.

This paper presents a pragmatic, end-to-end R workflow for clinical-style longitudinal analyses using linear models (lm), generalized linear models (glm), and mixed-effects models (lme4::lmer). Using a realistic multicenter use case, we demonstrate tidy data preparation, standardized model outputs via broom, scalable automation with purrr, and generation of submission-ready tables and reports using flextable and rmarkdown.

INTRODUCTION

Longitudinal clinical trials routinely generate correlated data through repeated measurements within subjects and clustering across sites. In therapeutic areas such as dermatology and immunology, endpoints like UAS7 (Weekly Urticaria Activity Score) are collected over multiple visits and form the basis of primary and secondary analyses.

Although mixed models for repeated measures (MMRM) are well established statistically, implementation in practice often involves fragmented workflows with manual result handling. This paper focuses on practical R coding patterns that support scalable modeling, automation, and reproducible reporting aligned with regulatory expectations.

1. ROADMAP AND CODING PHILOSOPHY

The workflow presented in this paper follows a consistent pattern:
- Start with tidy, long-format clinical data
- Apply appropriate models depending on endpoint type
- Standardize outputs immediately after modeling
- Automate analyses using functional programming
- Generate tables and reports directly from code

This philosophy emphasizes traceability, reproducibility, and reduction of manual errors.

2. CLINICAL USE CASE

We mimic a multicenter clinical trial with subjects nested within sites, repeated outcome measurements at Baseline, Week 4, Week 8, and Week 12, and both continuous and binary endpoints. This structure reflects common Phase II/III clinical trials and allows demonstration of multiple modeling strategies within a single workflow.

3. DATA SIMULATION: LONGITUDINAL STRUCTURE WITH SITE EFFECTS

Synthetic trial data were generated to reflect site-level heterogeneity, subject-level covariates, and treatment-by-time effects. This enables demonstration of modeling patterns without use of proprietary data.

CODE LISTING 3.1: DATA SIMULATION

```
set.seed(42)
library(dplyr); library(tidyr)

n_site <- 12; n_per <- 60
site_tbl <- tibble(
  site=factor(paste0("S",1:n_site)),
  site_b0=rnorm(n_site,0,2)
)

df <- crossing(site=site_tbl$site, id_in=1:n_per) %>%
  left_join(site_tbl, by="site") %>%
  mutate(
    usubjid=paste0(site,"-",id_in),
    age=round(rnorm(n(),40,12)),
    trt=factor(sample(c("PBO","TRT"), n(), TRUE)),
    x=rnorm(n())
)

visits <- tibble(
  visit=factor(c("BL","W4","W8","W12"),
               levels=c("BL","W4","W8","W12"))
)

ad <- df %>% crossing(visits) %>%
  mutate(
    week=recode(visit, BL=0L, W4=4L, W8=8L, W12=12L),
    mu=10 + 0.8*x + 0.04*age + 0.6*(trt=="TRT") +
         0.15*week + site_b0,
    outcome=mu + rnorm(n(),0,2.5)
  )
```

4. DATA PREPARATION AND MISSINGNESS

Missing data were introduced to reflect real-world trial conditions. Baseline and change-from-baseline variables were derived following ADaM-style conventions.

CODE LISTING 4.1: BASELINE AND CHANGE-FROM-BASELINE DERIVATION

```
set.seed(99)
ad$outcome[sample(nrow(ad), size=0.06*nrow(ad))] <- NA

ad_clean <- ad %>%
  filter(!is.na(outcome)) %>%
  group_by(usubjid) %>%
  mutate(
    base=outcome[visit=="BL"][1],
    chg  = outcome - base,
    responder = as.integer(chg <= -5)) %>%
  ungroup() %>%
  mutate(x_c=scale(x, center=TRUE, scale=FALSE))
```

5. CROSS-SECTIONAL MODELS

Simple cross-sectional models are often used as reference or supportive analyses. A linear model and a binary responder model were fitted at Week 12.

CODE LISTING 5.1: LINEAR MODEL AT WEEK 12

```
lm_w12 <- ad_clean %>%
  filter(visit=="W12") %>%
  lm(chg ~ trt + x_c + age, data = .)

summary(lm_w12)
```

CODE LISTING 5.2: BINARY RESPONDER MODEL

```
ad_w12 <- ad_clean %>%
  filter(visit=="W12") %>%
  mutate(responder=as.integer(chg <= -5))

glm_resp <- glm(
  responder ~ trt + x_c + age,
  data=ad_w12, family=binomial
)

summary(glm_resp)
exp(coef(glm_resp))
```

6. MIXED MODEL FOR REPEATED MEASURES

To account for within-subject correlation and site-level heterogeneity, a mixed-effects model was fitted using lme4::lmer(). Fixed effects estimate treatment effects over time, while random effects capture correlation.

CODE LISTING 6.1: MIXED MODEL SPECIFICATION

```
library(lme4)

lmm <- lmer(
  chg ~ trt*visit + x_c + age +
    (1|site) + (1|usubjid),
  data=ad_clean
)

summary(lmm)
```

7. TIDY OUTPUTS AND AUTOMATION

Model outputs were standardized using broom and broom.mixed, enabling scalable automation and consistent reporting across models.

CODE LISTING 7.1: TIDY MODEL OUTPUTS

```
library(broom); library(broom.mixed)

tidy_lm  <- tidy(lm_w12, conf.int=TRUE)
tidy_glm <- tidy(glm_resp, conf.int=TRUE, exponentiate=TRUE)
tidy_lmm <- tidy(lmm, effects="fixed", conf.int=TRUE)
```

CODE LISTING 7.2: AUTOMATION WITH PURRR

```
fit_models <- function(dat){
  list(
    lm  = lm(chg ~ trt + x_c + age, data=filter(dat, visit=="W12")),
    glm = glm(responder ~ trt + x_c + age,
              data=filter(dat, visit=="W12"), family=binomial),
    lmm = lmer(chg ~ trt*visit + x_c + age +
                (1|site) + (1|usubjid),
                data=dat)
  )
}

res <- fit_models(ad_clean)
purrr::imap_dfr(res, ~ tidy(.x, conf.int=TRUE), .id="model") %>% head(10)
```

8. SUBMISSION-READY REPORTING

Publication-ready tables were generated using flextable, and dynamic reports were created using rmarkdown to ensure reproducibility and traceability.

CODE LISTING 8.1: FLEXTABLE OUTPUT

```
library(flextable)

tbl <- tidy_lm %>%
  transmute(
    Term=term,
    Estimate=round(estimate,3),
    `95% CI`=paste0("(",round(conf.low,3),", ",
                          round(conf.high,3),")"),
    `P-value`=signif(p.value,3)
  )

flextable(tbl) %>% autofit()
```

9. DISCUSSION AND PRACTICAL TIPS

The workflow presented demonstrates how tidy data principles, standardized model outputs, and functional programming can significantly reduce manual effort and error risk in clinical analyses. These patterns are particularly valuable in environments requiring frequent reruns and audit readiness.

10. LIMITATIONS

The simulated examples presented here reflect common clinical trial structures but may require extension for more complex designs, including stratification factors, alternative covariance structures, and sensitivity analyses.

CONCLUSION

By integrating modeling, automation, and reporting into a single reproducible workflow, R can support robust clinical analyses aligned with regulatory expectations. The coding tips and patterns presented here are intended to be reused and adapted in real-world clinical programming projects.

REFERENCES

1. Mallinckrodt C. et al. Analyzing longitudinal clinical trial data using mixed models for repeated measures. Pharmaceutical Statistics.
2. Bates D. et al. Fitting linear mixed-effects models using lme4. Journal of Statistical Software.
3. Lenth R. emmeans: Estimated Marginal Means.
4. Roche M. mmrm: Mixed Models for Repeated Measures in R.
5. EMA. Guideline on missing data in confirmatory clinical trials.
6. FDA. Statistical Considerations for Clinical Trials.
7. PHUSE. Best Practices for Automation and Traceability in Clinical Reporting.

CONTACT INFORMATION

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

Author Name: Keyur Shah
Company: ICON PLC
Address: Bangalore
Work Phone: +91 9722245800
Email: keyur.shah@iconplc.com

Brand and product names are trademarks of their respective companies.