PharmaSUG 2025 - Paper OS-24

An End-to-End Workflow for TFL Generation using R: MMRM Applications and Comparative Insights with SAS

Kai Lei, Jiaqiang Zhu, and Margaret Huang, Vertex Pharmaceuticals Inc.

ABSTRACT

Tables, Figures, and Listings (TFLs) are essential for presenting clinical trial data in regulatory submissions, ensuring transparency and compliance, and they are traditionally generated using SAS. With the rise of R as an open-source, flexible, and cost-effective alternative, this paper presents an end-to-end workflow for TFL creation in R, using Mixed-Effects Models for Repeated Measures (MMRM) as a representative application example. Our workflow covers data preparation, model implementation, TFL generation, and regulatory-compliant output formatting, leveraging R packages like *dplyr*, *ggplot2* and *r2rtf*.

A side-by-side comparison of R's *mmrm* package with SAS's *PROC MIXED* highlights the differences in syntax, flexibility, and visualization capabilities, demonstrating R's advanced options for custom graphics and layout adjustments.

This paper provides practical guidance for clinical trial programmers and statisticians interested in adopting R for TFL creation and offers recommendations for a smooth transition from SAS. Future directions include implementing additional analysis models in R and finalizing standardized functions for clinical reporting.

INTRODUCTION

In clinical trials, Tables, Figures, Listings (TFLs) play a vital role in summarizing data into a clear, interpretable format, helping regulatory agencies to assess the efficacy and safety of new treatments effectively. It's critical for TFLs to be well designed to ensure adherence to regulatory guidelines and to provide transparency in clinical trial data presentation.

Historically, SAS has been the primary tool for generating TFLs in regulatory submissions. However, with the rise of R as an open-source, flexible and cost-effective alternative, more companies are integrating R in the TFL creation and submission workflows.

This paper presents an end-to-end workflow for TFL creation in R, using Mixed-Effects Models for Repeated Measures (MMRM) as an example. The workflow encompasses data preparation, model implementation, TFL generation and regulatory-compliant output formatting. A side-by-side comparison of R's *mmrm* package with SAS's *PROC MIXED* highlights the differences in syntax, flexibility, and visualization capabilities, demonstrating R's advanced options for custom graphics and layout adjustments.

MMRM is selected for the demonstration due to its widespread use in clinical trials for analyzing repeated measurements over time. In addition, its robustness in handling missing data without imputation helps preserve statistical validity and minimize bias.

Overall, this paper aims to showcase R's capability in producing regulatory-compliant TFLs and its potential to complement SAS in clinical trial reporting. By illustrating R's application through MMRM example, we hope to highlight its growing role as the pharmaceutical industry increasingly embraces open-source solutions in clinical trial.

TFLS GENERATION: AN END-TO-END WORKFLOW

The end-to-end workflow begins with raw data transformation into SDTM data sets using the *sdtm.oak* and *dplyr* packages. We then derive ADaM data sets using *admiral* and *dplyr* packages. Next, we

implement MMRM model using the *mmrm* and *emmeans* packages. We then create TFLs by leveraging *dplyr,* and *ggplot2* packages. Lastly, we generate the formatted RTFs using the *r2rtf* package. This structured approach presents a polished and reproducible reporting process. In this paper, we will specifically highlight the model implementation and RTF creation components. We will illustrate using dummy data in ADSL and ADSP (Analysis Spirometry Tests) data sets.

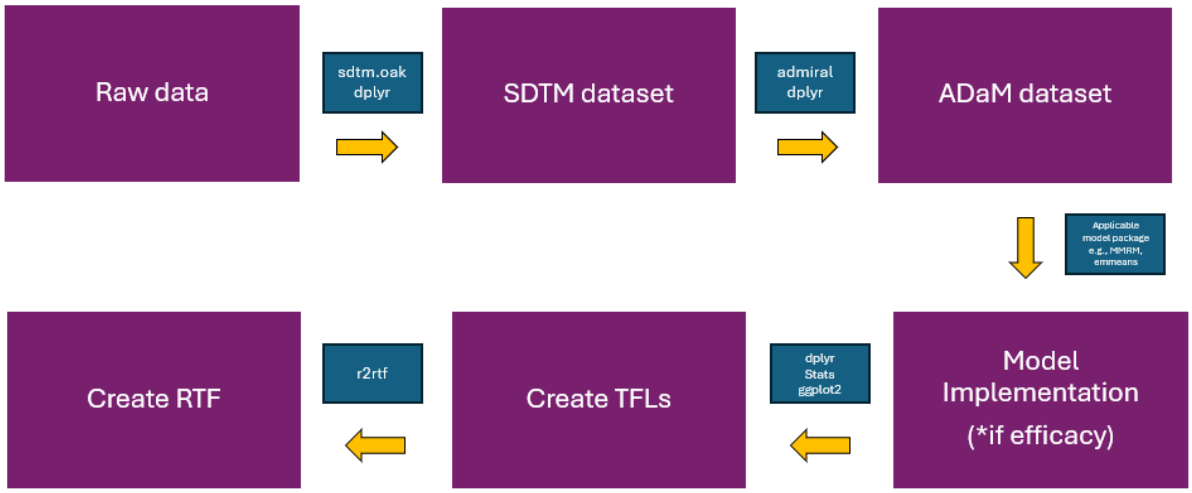


Figure 1. End-to-End Workflow

A diagram that maps the complete workflow, from data preparation to RTF output, providing a high-level overview of the process.

DATA PREPARATION

The data preparation process begins with loading the required libraries and processing raw data. This includes handling missing values, addressing duplicates and resolving formatting issues to ensure data integrity and consistency. Using *sdtm.oak* and *dplyr* packages, we transform raw data into SDTM data sets, maintaining traceability to the original source. Next, we create ADaM data sets using *admiral* and *dplyr* to ensure that the data is ready for statistical analysis. Throughout the data transformation process, we follow CDISC standards to maintain regulatory compliance and consistency, which helps ensure accurate and reproducible downstream analysis and reporting. To obtain the marginal means from MMRM model, we first calculate the mean values for baseline continuous covariates and observed proportions for the categorical variables using the ADSL data set.

```
#calculate baseline mean percent predicted FEV1 (forced expiratory volume in 1 second)

m_ppfev <- adsl %>% filter(FASFL=="Y") %>% filter(!is.na(PPFEV1BL)) %$% mean(PPFEV1BL)


#calculate baseline proportion for age at screening (>=12 to <18 vs >=18 years)

age_scr_freq <- adsl %>% filter(FASL=="Y") %$% table(START2V) %>% prop.table()


#calculate baseline proportion for sex (male vs female)

sex_freq <- adsl %>% filter(FASFL=="Y") %$% table(STRAT3V) %>% prop.table()
```

Program 1. Baseline covariates calculation

TFLS CREATION IN R

In R, we generate TFLs using a combination of the following packages that enhance the usability, readability and performance:

- *tidyverse:* this collection of R packages provides a consistent and intuitive framework for importing, cleaning, transforming, visualizing and modeling data. Its core packages are *dplyr, ggplot2, tidyr* etc.
- *dplyr:* This package simplifies data wrangling, including filtering, summarizing and arranging data sets for TFL generation. It provides an intuitive syntax with verbs like *filter(), select(), mutate(), summarise(),* making the data manipulation more efficient. *dplyr* is optimized for handling large data sets and integrates smoothly with *ggplot2* and *gt*, enabling the creation of well-formatted tables and figures that are highly customizable.
- *ggplot2:* This package is widely used for data visualization, offering a flexible and customizable approach to creating figures and plots. It follows a layered grammar of graphics, making it easy to extend and customize visualizations. It's able to generate high-quality figures for reports and publications by providing options for modifying colors, labels, themes, facets and annotations.

We have created a custom R function called *round2* to ensure flexible, consistent and precise decimal formatting for numeric values in tables and listings. It begins by preserving the sign of the input *x*, then takes the absolute value and scales it by *10^digits* to shift the decimal point. To ensure correct rounding, it adds 0.5 plus a small machine epsilon (via *sqrt(.Machine$double.eps))* before truncating the value to remove the decimal portion. This helps to handle tricky cases where floating-point precision might cause unexpected results. After rounding, it shifts the decimal point back to its original place and restores the original sign. Finally, it returns the rounded number as a string formatted to show exactly the number of decimal places the user asked for.

We have also added a function to format p-value where if a p-value is less than 0.0001, it will display "<0.0001".

```
round2 = function(x, digits) {

  posneg = sign(x)

  z = abs(x)*10^digits

  z = z + 0.5 + sqrt(.Machine$double.eps)

  z = trunc(z)

  z = z/10^digits

  z*posneg



  return(sprintf(paste0("%.",digits,"f"),z*posneg))

  }

format_pval <- function(p, digits = 4) {

  ifelse(p < 0.0001, "<0.0001", sprintf(paste0("%.", digits, "f"), p))

}
```

Program 2. Custom R functions for precise decimal formatting and numeric display in TFL generation

FINAL OUTPUT IN RTF FILE FORMAT

Following the creation of TFLs, we can utilize the *r2rtf* package to format the TFLs into RTF outputs with structure formatting, including headers, footers and styling options. Using *r2rtf* function, we can define the layout, organize the content and format the output to meet clinical reporting requirements.

The following code demonstrates how different *r2rtf* functions contribute to customizing the final RTF output. The *rtf_page()* function defines key page attributes such as orientation, page size, borders and margins to establish the overall layout. Next, *rtf_title()* function adds a title to the page with specified justification, font, and font size, while *rtf_page_header()* function similarly configures the header text, font properties and color. The *rtf_colheader()* function defines and formats the column headers by specifying column widths, justification and border styles. The main table is then constructed with *rtf_body()*, which aligns text, adjusts font details and handles indentation. Additional, *rtf_page_footer()* inserts footnotes and metadata such as the program source, program name and creation time at the bottom of each page. We use {\\sub} to process subscript in both the title and footnote. Finally *rtf_encode()* and *write_rtf()* complete the process by encoding the footnotes and generating the final RTF file, ensuring a well-formatted report.

```
df_final %>%

  rtf_page(orientation  ="landscape", width=11, height=8.5,

           border_first = "single",

           border_last  = "",

           margin       = c(0.38,0.38,1,0.38,1,0.38),

           nrow         = 30,

           col_width    = 8) %>%



  rtf_title(title = c("Table 1",

                      "MMRM Analysis of Absolute Change from Baseline in ppFEV{\\sub
1} through Week 24",

                      "Full Analysis Set",

                      "",

                      "Unit for Absolute Change from Baseline in ppFEV1: percentage
points",

                      ""),

            text_justification ="c",

            text_font_size     = 9,

            text_font          = rep(9, 5)

            ) %>%



  rtf_page_header(c(paste0("Vertex Pharmaceuticals Incorporated
Page \\pagenumber of \\pagefield")),

                  text_justification = "l",

                  text_font_size     = 9,

                  text_font          = rep(9, 2)
```

```
                ) %>%


  rtf_colheader(colheader1,
                col_rel_width      = rel_width,
                text_justification = "l",
                border_left        = rep("",4),
                border_right       = rep("",4),
                text_font          = rep(9, 4),
                text_font_size     = 9,
                border_bottom      = rep("single",4),
                border_top         = rep("single",4)
               ) %>%


  rtf_body(col_rel_width      = rel_width,
           text_justification = c(rep("l", 4)),
           text_indent_first  = -240,
           text_indent_left   = 180,
           text_font_size     = 9,
           border_left        = "",
           border_right       = "",
           border_bottom      = "",
           text_font          = rep(9, 4),
           text_convert       = TRUE
          ) %>%


  rtf_footnote(footnote =
                 c(
                   "- Baseline is defined as the most recent non-missing measurement
before the first dose of study drug in the Treatment Period.",
                   paste0(
                     "- MMRM included data from all available visits up to Week 24,
with treatment, visit, and treatment*visit as fixed effects and baseline ppFEV{\\sub
1}, age group at screening (<18 vs >=18 years) and CFTR modulator use at Screening
(yes vs no) as covariates. Covariance structure=",
                     base$type,
                     ", DF=Kenward-Roger. ","Measurements at Day 15 were not included
in the estimation of the average treatment effect through Week 24."),
```

```
                    paste0("Program Source: ", path1, "/", filenameR, " Creation: ",
toupper(format(Sys.time(), format='%d%b%Y %H:%M')))

                    ),

                text_justification = "l",

                border_left        = "",

                border_right       = "",

                border_bottom      = '',

                text_font          = rep(9, 4),

                text_font_size     = 9,

                text_format        = c("")

                ) %>%


  rtf_encode(

    page_footnote = "all"

  ) %>%

  write_rtf("mmrm_table.rtf")
```

Program 3. Example of using *r2rtf* functions to format and generate a final RTF report from MMRM table outputs.

OUTPUT CUSTOMIZATION AND REGULATORY COMPLIANCE

Vertex Pharmaceuticals Incorporated Page 1 of 1

Table 1
MMRM Analysis of Absolute Change from Baseline in ppFEV$_1$ through Week 24
Full Analysis Set

Unit for Absolute Change from Baseline in ppFEV1: percentage points

	Placebo N = 201	Treatment N = 202
Baseline		
n	201	202
Mean (SD)	62.1 (15.8)	60.8 (14.7)
Absolute change through Week 24		
n	199	200
LS mean (SE)	25.7 (0.8)	26.0 (0.8)
95% CI of LS mean	(24.0, 27.3)	(24.3, 27.7)
P value within treatment	<0.0001	<0.0001
LS mean difference, 95% CI	--	0.3(-2.0, 2.7)
P value versus Placebo	--	0.7871

- Baseline is defined as the most recent non-missing measurement before the first dose of study drug in the Treatment Period.
- MMRM included data from all available visits up to Week 24, with treatment, visit, and treatment*visit as fixed effects and baseline ppFEV$_1$, age group at screening (<18 vs ≥ 18 years) and CFTR modulator use at Screening (yes vs no) as covariates. Covariance structure=UN, DF=Kenward-Roger. Measurements at Day 15 were not included in the estimation of the average treatment effect through Week 24.
Program Source: C:/Users/huangm3/OneDrive - Vertex Pharmaceuticals/Desktop/mmrm.R Creation: 26MAR2025 01:12

Figure 2. MMRM Table RTF Output Created in R

Vertex Pharmaceuticals Incorporated Page 1 of 1

Table 14.x.x.x
MMRM Analysis of Absolute Change from Baseline in $ppFEV_1$ through Week 24
Full Analysis Set

Unit for Absolute Change from Baseline in $ppFEV_1$: percentage points

	Placebo N = 201	Treatment N = 202
Baseline		
n	201	202
Mean (SD)	62.1 (15.8)	60.8 (14.7)
Absolute change through Week 24		
n	199	200
LS mean (SE)	25.7 (0.8)	26.0 (0.8)
95% CI of LS mean	(24.0, 27.3)	(24.3, 27.7)
P value within treatment	<0.0001	<0.0001
LS mean difference, 95% CI	--	0.3 (-2.0, 2.7)
P value versus placebo	--	0.7871

– Baseline is defined as the most recent non-missing measurement before the first dose of study drug in the Treatment Period.
– MMRM included data from all available visits up to Week 24, with treatment, visit, and treatment*visit as fixed effects and baseline $ppFEV_1$, age group at screening (≥12 to <18 vs ≥18 years), and sex (male vs female) as covariates. Covariance structure=UN, DF=Kenward-Roger. Measurements at Day 15 were not included in the estimation of the average treatment effect through Week 24.
Program: Draft VXXXX\XXX\Final\dev\tables\t-sp-mmrm-abs-ppfev-24wks.sas Creation: 14MAY2019 21:09

Figure 3. MMRM Table RTF Output Created in SAS

MMRM (MIXED-EFFECTS MODLES FOR REPEATED MEASURES) AS AN APPLICATION EXAMPLE

MMRM OVERVIEW

Mixed-Effects Models for Repeated Measures (MMRM) are widely used in clinical trials to assess treatment effects over time, particularly in studies with repeated measurements. Compared to traditional repeated-measures ANOVA, MMRM offers greater flexibility by accounting for correlations within individuals and handling unbalanced data without requiring imputation for missing values. By using restricted maximum likelihood (REML) estimation, MMRM effectively incorporates all available data under the assumption that missing data occur at random (MAR), ensuring reliable statistical inference.

MMRM is commonly applied in efficacy analyses, where primary or secondary endpoints are measured at multiple time points. It provides flexibility in modeling covariance structures, such as unstructured or autoregressive models, allowing for a more accurate representation of relationships between repeated observations. Due to its broad acceptance by regulatory agencies, MMRM has become a standard method in clinical research and is frequently used in TFL generation for regulatory submissions.

COMPARISON OF R AND SAS CAPABILITIES FOR MMRM

Both R and SAS offer strong capabilities for conducting MMRM analyses, but they differ in approach and ease-of-use. SAS has long been the industry standard in regulated environments due to its validated procedures, streamlined syntax and robust handling of large clinical data sets. Regulatory agencies widely accept SAS for submissions, with *PROC MIXED* being the preferred choice for modeling the

repeated continuous measurements. Its well-structured syntax, built-in diagnostics and comprehensive output enhance model validation and regulatory compliance.

In contrast, R is an open source and highly flexible tool, supported by a rich ecosystem of packages that enable custom modeling and advanced visualizations. The flexibility allows for innovative research and tailored analyses. The implementation of MMRM can be efficiently handled using the *mmrm* package, which allows users to specify flexible covariance structures. Additionally, the *emmeans* package facilitates the extraction and visualization of estimated marginal means, making post-hoc comparisons of treatment effects more straightforward.

Overall, R's combination of *mmrm* and *emmeans* provides flexibility and powerful customization while SAS's *PROC MIXED* stands out for its standardized implementation, and its critical role in regulatory submissions.

R Packages for MMRM

In this example model, we have *CHG* (absolute change from baseline in $ppFEV_1$) as the outcome of interest. The covariates are *TRT01P_fac* (Placebo vs Treatment), *AVISITSN_fac* (Post-baseline Visits), *STRAT2V* (Age group), *STRAT3V* (Sex), *PPEV1BL* (Baseline $ppFEV_1$) and *TRT01P_fac*AVISITN_fac* the interaction term of treatment and visit. The *us()* component stands for the unstructured covariance matrix, which allows for each timepoint to have its own variance and correlation and the input *AVISITN_fac* | *USUBJID* models the repeated measures over subject (*USUBJID)* across visits (*AVISITN_fac).* The control setting uses the Kenward-Roger method for adjusting degrees of freedom. Overall, this model is analyzing the absolute change from baseline in $ppFEV_1$ overtime across 2 treatment arms, adjusting for age group, sex, and baseline lung function, while accounting for within-subject correlation at each visit.

During the model implementation step, we check for model convergence by trying the unstructured covariance structure first. If model doesn't converge, it automatically switches to try the compound symmetry covariance structure. Following the creation of the MMRM model *mod1*, we use the *emmeans* function to create the basis for custom contrasts. The *levels* function produces the factor levels (visit) and helps user to confirm the ordering and indexing when building custom contrast weights. Next, we use the *contrast* function to construct a weighted average of placebo group marginal means:

- *rep(c(0,0.2,0.2,0.2,0.2,0.2),8)*: assigns 0 to Day 15 and 0.2 weights to each of 5 post-Day 15 visits
- *rep(rep(c(1,0),each=6),4):* selects placebo rows only (1 for placebo, 0 for treatment), assuming each treatment appears 6 times per block.
- *rep(rep(STRAT2V_freq,each=12),2)*rep(STRAT3V_freq,each=24):* lastly *STRAT2_freq* and *STRAT3V_freq* are used to weight by the proportion of subjects in each stratum (like a population average over strata).

Next, we apply the same logic for the treatment group. Lastly, we compute the treatment difference using the same visit weights (measurements at Day 15 were not included in the estimation of the average treatment effect through Week 24), but instead of selecting one arm, we use -1 for placebo and 1 for treatment in the contrast. This contrast statement gives the LS mean treatment difference through Week 24, adjusted for covariates and stratification.

```
#first try type=UN (Unstructured), if not then automatically try type=CS

#fit MMRM model,note "Kenward-Roger" degrees of freedom method with

#linear approximation is specified (corresponding to the KR in SAS).



mod1 <- mmrm(

  CHG ~ TRT01P_fac + AVISITN_fac + STRAT2V + STRAT3V + PPFEV1BL + TRT01P_fac*AVISITN_fac
+ us(AVISITN_fac|USUBJID),
```

```
  data = model_data ,
  control = mmrm_control(method = "Kenward-Roger",vcov = "Kenward-Roger-Linear")
)

#first try type=UN
mmrm_type <- "UN"

# if UN failed automatically switch to Compound Symmetry (CS)
if (mod1$opt_details$convergence == 1) {
  message("Unstructured (UN) failed, switching to Compound Symmetry (CS)")
  # if UN failed, automatically switch to Compound Symmetry (CS)
  mod1 <- mmrm(
    CHG ~ TRT01P_fac + AVISITN_fac + STRAT2V + STRAT3V + PPFEV1BL +
TRT01P_fac*AVISITN_fac + cs(AVISITN_fac|USUBJID),
    data = model_data ,
    control = mmrm_control(method = "Kenward-Roger",vcov = "Kenward-Roger-Linear")
  )
  # last try type=CS
  mmrm_type <- "CS"
}

mod1

#Calculate estimated marginal means using emmeans package.
emm <- emmeans(mod1, ~ AVISITN_fac +TRT01P_fac +STRAT2V + STRAT3V,
               at = list(PPFEV1BL=m_ppfev))

#check data
levels(emm@grid$AVISITN_fac)

#Based on the estimated marginal means results above,
#construct the contrast matrix and calculate LS mean (through Week 24)
#with weights obtained from ADSL.
#Visits after Day 15 are averaged.
emm_pbo <- contrast(emm,
```

```
                    list(pbo =
rep(c(0,0.2,0.2,0.2,0.2,0.2),8)*rep(rep(c(1,0),each=6),4)*rep(rep(STRAT2V_freq,each=12),2
)*rep(STRAT3V_freq,each=24)

                    ))



emm_trt <- contrast(emm,

                    list(trt =
rep(c(0,0.2,0.2,0.2,0.2,0.2),8)*rep(rep(c(0,1),each=6),4)*rep(rep(STRAT2V_freq,each=12),2
)*rep(STRAT3V_freq,each=24)

                    ))



emm_tte <- contrast(emmeans(mod1, ~ AVISITN_fac +TRT01P_fac,

                            at = list(PPFEV1BL=m_ppfev)),

                    list(diff = rep(c(0,0.2,0.2,0.2,0.2,0.2),2)*rep(c(-1,1),each=6)

                    ))
```

Program 4. MMRM Model implementation and estimated marginal means

Side-by-Side Comparison with SAS's PROC MIXED

In both SAS's *PROC MIXED* and R's *mmrm* package, the overall structure of the model is similar: both define a linear mixed effects model that includes as fixed effects the variables: treatment arm, categorical visit, treatment by visit interaction, and other covariates for adjustment. They apply the Kenward-Roger method for degrees of freedom and use either estimate statement in SAS or contrast function in R to compare treatment groups.

Despite these similarities, the syntax and workflow differ between the two tools. In SAS, the entire modeling and inference process -- including the model specification, covariance structure, and contrasts -- is handled within *PROC MIXED.* In contrast, R separates these steps: the *mmrm* function is used to fit model, while *emmeans* package is used to compute marginal means and evaluate contrasts. The color-coded boxes in the figures below illustrate how each step in SAS -- such as model definition and estimate statement -- maps to its counterpart in R—form model formula specification to contrast setup through *emmeans*. This highlights the shared conceptual approach between the two tools despite difference in coding styles.

```
mod1 <- mmrm (
  CHG ~ TRT01P_fac + AVISITN_fac + STRAT2V + STRAT3V + PPFEV1BL +
    TRT01P_fac * AVISITN_fac + us(AVISITN_fac | USUBJID),
  data = model_data,
  control = mmrm_control(method = "Kenward-Roger", vcov = "Kenward-Roger-Linear")
)

#Calculate estimated marginal means using emmeans package.
emm <- emmeans(mod1, ~ AVISITN_fac +TRT01P_fac +STRAT2V + STRAT3V, at = list(PPFEV1BL=m_ppfev))

emm_pbo <- contrast(emm,
                    list(pbo = rep(c(0,0.2,0.2,0.2,0.2,0.2),8)*rep(rep(c(1,0),each=6),4)*rep(rep(STRAT2V_freq,each=12),2)*rep(STRAT3V_freq,each=24)
                    ))

emm_trt <- contrast(emm,
                    list(trt = rep(c(0,0.2,0.2,0.2,0.2,0.2),8)*rep(rep(c(0,1),each=6),4)*rep(rep(STRAT2V_freq,each=12),2)*rep(STRAT3V_freq,each=24)
                    ))

emm_tte <- contrast(emmeans(mod1, ~ AVISITN_fac +TRT01P_fac,
                            at = list(PPFEV1BL=m_ppfev)),
                    list(diff = rep(c(0,0.2,0.2,0.2,0.2,0.2),2)*rep(c(-1,1),each=6)
                    ))
```

Figure 4. MMRM Implementation in R

```
proc mixed data = mmrm;
  class usubjid trt avisitn strat2v strat3v;
  model chg = trt avisitn trt*avisitn ppfev1bl strat2v strat3v / solution ddfm=kenwardroger;
  repeated avisitn / type=un sub=usubjid;

  *** cfb through week 24 ***;
  estimate  'PBO through week 24'    int 1 avisitn 0 0.2 0.2 0.2 0.2 0.2 trt 1 0 trt*avisitn 0 0.2 0.2 0.2 0.2 0.2   0   0   0   0     0   0 ppfev1bl &mnppfev1b. strat2v &p1ag. &p2ag. strat3v &p1sx. &p2sx.;

  estimate  'TRT through week 24'    int 1 avisitn 0 0.2 0.2 0.2 0.2 0.2 trt 0 1 trt*avisitn 0 0 0 0 0 0             0   0.2 0.2 0.2 0.2 0.2 ppfev1bl &mnppfev1b. strat2v &p1ag. &p2ag. strat3v &p1sx. &p2sx.;

  ***trt difference through week 24 ***;
  estimate  'TRT v.s. PBO through week 24'                          trt -1 1 trt*avisitn 0 -0.2 -0.2 -0.2 -0.2 -0.2   0   0.2 0.2 0.2 0.2 0.2 / cl;

  ods output estimates=estimates;
run;
```

Figure 5. MMRM Implementation in SAS

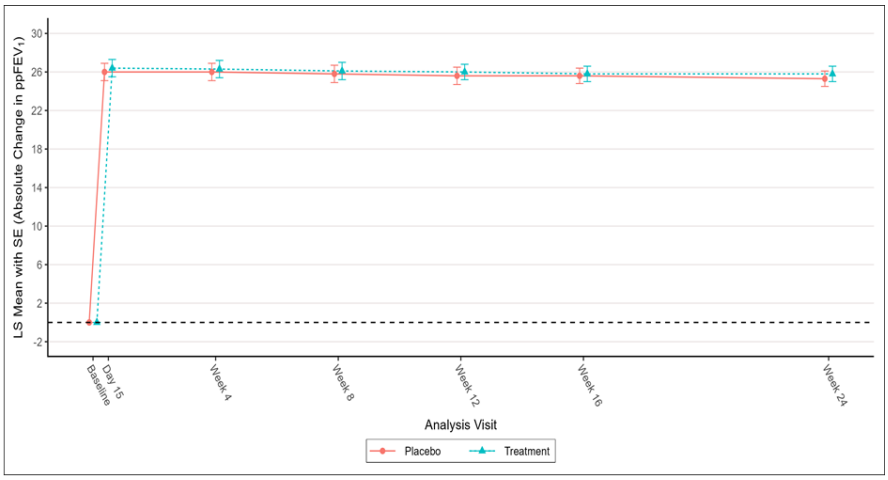
In the figures, each color-coded box corresponds to a particular segment of the model specification and contrast setup in R's *mmrm* and SAS's *PROC MIXED* workflow:

- Blue box: shows the initial model specification. In R, the *mmrm* function defines the repeated-measures model and applies Kenward-Roger method for degrees of freedom, while in SAS, *PROC MIXED* is declared with its key options.
- Red and lavender boxes: indicate the generation of marginal means for each treatment group
- Green box: highlights the computation of estimated differences between treatment group at a particular timepoint

Figures and Graphical Representations

Vertex Pharmaceuticals Incorporated Page 1 of 1

Figure 1
MMRM Analysis of Absolute Change from Baseline in $ppFEV_1$ at Each Visit up to Week 24
Full Analysis Set



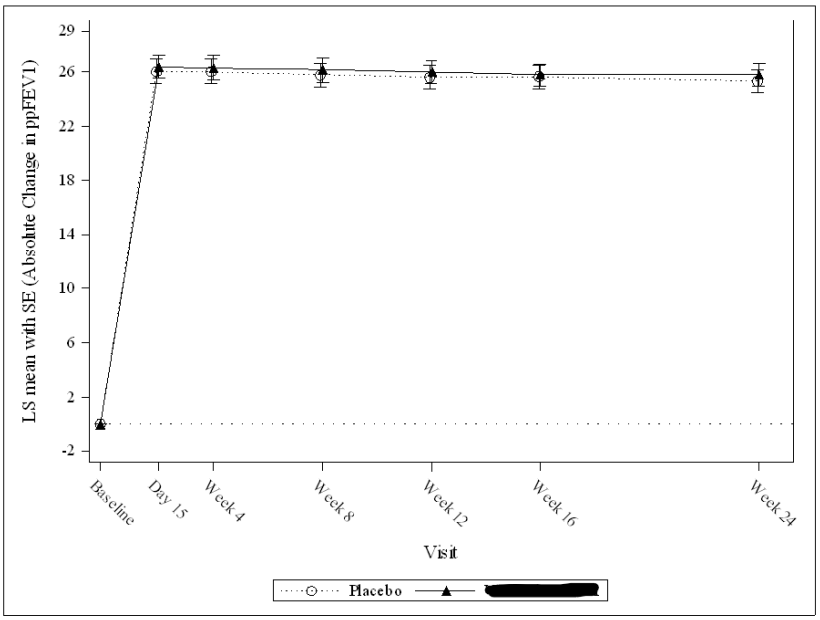
- Baseline is defined as the most recent non-missing measurement before the first dose of study drug in the Treatment Period.
- MMRM included data from all available visits up to Week 24, with treatment, visit, and treatment*visit as fixed effects and baseline $ppFEV_1$, age group at screening (<18 vs ≥ 18 years) and CFTR modulator use at Screening (yes vs no) as covariates. Covariance structure=UN, DF=Kenward-Roger.

Program: C:/Users/huangm3/OneDrive - Vertex Pharmaceuticals/Desktop/mmrm.R Creation: 25Mar2025 09:38

Figure 6. MMRM Figure created in R

Vertex Pharmaceuticals Incorporated Page 1 of 1

Figure 14.x.x.x
MMRM Analysis of Absolute Change from Baseline in $ppFEV_1$ (Percentage Points) at Each Visit up to Week 24
Full Analysis Set



- Baseline is defined as the most recent non-missing measurement before the first dose of study drug in the Treatment Period.
- MMRM included data from all available visits up to Week 24, with treatment, visit, and treatment*visit as fixed effects and baseline $ppFEV_1$, age group at screening (≥12 to <18 vs ≥18 years), and sex (male vs female) as covariates. Covariance structure=UN, DF=Kenward-Roger.
Program: Draft VXXXX\XXX\Final\dev\figures\f-sp-abs-ppfev-vis-wk24-lsm.sas Creation: 14MAY2019 22:21

Figure 7. MMRM Figure Created in SAS

ADVANTAGES OF USING R FOR TFLS CREATION

FLEXIBILITY AND CUSTOMIZATION

One of the key advantages of using R for TFL creation is its flexibility and customization. R allows users to design and format outputs exactly to specification, whether for internal review, regulatory submission, or publications. With packages like *r2rtf* and *ggplot2* and the ability to create highly customized functions, every detail – from layout and styling and formatting- can be finely controlled. R also supports dynamic programming, enabling reusability across clinical trials with different designs or parameters. This adaptability makes R a powerful tool for generating high-quality, reproducible TFLs tailored to diverse clinical and regulatory needs.

SEAMLESS COLLABORATION

Using R for TFL creation promotes seamless collaboration between statisticians and programmers. Since many statisticians are already trained in R and comfortable with its syntax, adopting it as the primary tool for TFLs minimizes the communication barrier across roles. The open-source nature of R and its rich ecosystem- featuring packages like *knitr* and *RMarkdown*- enables the integration of data analysis and reporting, allowing for dynamic, reproducible outputs that can be efficiently updated as new data become available.

ADDITIONAL BENEFITS

Cost Efficiency

R's open-source nature makes it a highly cost-efficient alternative to SAS by eliminating license fees and reducing overall software costs. This cost benefit is increasingly attractive to pharmaceutical

organizations where budget constraints for statistical tools are a critical consideration. By leveraging open-source tools like R, companies can build efficient and scalable analytical pipelines without incurring the high costs associated with commercial software, freeing up resources for other strategic initiatives.

Community Support and Growing Adoption

Lastly, R's rapidly growing community support is a key driver of its continuous improvements. The extensive online resources, forums, support groups and collaborative projects all continue to expand R's capabilities and presence within the industry. In particular, the Pharmaverse project – a collaborative, community-driven initiative focusing on developing and promoting open-source tools and resources for the pharmaceutical industry- is fostering an environment of shared innovation and rapid problem solving among a vibrant network of statisticians, data scientists and statistical programmers. As adoption of R expands in the industry, its community driven enhancements are proving to be a significant asset in enhancing reproducibility, efficiency and innovation in pharmaceutical analytical space, encouraging the broader adoption of open-source methodologies in the field.

CONCLUSION

SUMMARY OF WORKFLOW AND KEY ADVANTAGES

In conclusion, the end-to-end workflow – from raw data through SDTM and ADaM data sets, to MMRM analysis and TFL & RTF generation- showcases the advantages of adopting R-based solutions in clinical reporting. R's specialized packages (e.g. *mmrm*, *emmeans*) efficiently implement the MMRM, and seamlessly integrate with TFL creation tools like *r2rtf*, all within a collaborative, open-source environment. Compared to SAS, this approach offers a more flexible, cost-effective solution backed by continuous community-driven innovation. Ultimately R's efficiency, transparency and reproducibility make it a compelling choice for clinical trial reporting, encouraging broader adoption of open-source solutions with the pharmaceutical analytics space.

CHALLENGES OF USING R FOR TFLS CREATION

Version Control

Frequent updates to R and its packages can pose challenges in maintaining consistent TFL outputs. Given R's open-source nature and non-regulated environment, new releases may introduce changes in functionality or deprecate older functions. This can cause backward compatibility issues which might require code and workflow update or stay on older versions. In addition, some R packages are community-maintained, meaning long-term support or bug fixes can be less predictable or even unstable compared to commercial software. Despite these hurdles, proactive planning and well-defined processes and documentations can help mitigate most version control challenges.

Regulatory Compliance

Due to R's open-source nature, additional validation steps may be necessary to show that analyses and outputs are regulatory-compliant during submissions. Regulatory agencies might request evidence of software qualification that involves extensive documentation and testing of all packages used in the workflow of the submission. Hence, establishing internal validation procedures and standard operating protocols (SOPs) is critical for meeting these requirements. While community-driven resources can aid in package validation, each pharmaceutical organization holds the responsibility of ensuring that chosen packages are well-validated and well-tested for compliance. Nonetheless, with proper governance and robust validation strategies, R can be effectively integrated into regulatory submissions.

SUCCESSFUL R-BASED SUBMISSIONS

Despite the potential obstacles related to version control and regulatory compliance, successful R-based regulatory submissions by organizations like Novo Nordisk and Roche demonstrate that R is both a feasible and robust choice for clinical data analysis. These real-world examples illustrate that, with adequate validation and thorough documentation, R can meet stringent regulatory standards. As more

organizations begin to incorporate R in their analytics space, its open-source community will further refine and share best practices, making regulatory submission acceptance increasingly attainable.

RECOMMENDATIONS FOR INDUSTRY USE

Organizations seeking to incorporate R in their analytics pipeline can begin by introducing R in smaller pilot projects to build internal expertise and confidence. While R offers numerous benefits, it is not intended to replace SAS entirely; rather, it can complement existing workflows to create a more robust analytics pipeline. Providing comprehensive training- through official R documentation, online tutorials and user communities- helps statisticians and programmers to adapt to R's syntax and workflow. Tools like *renv* can help manage versions to ensure reproducibility, which alleviates concerns over frequent package updates. Furthermore, using validated packages the Pharmaverse ecosystem can offer teams the option to begin exploring programming activities such as ADaM data sets creation (*admiral*), TFL generation (*rtables*) and clinical data exploration/review using Shiny-based interactive tools (*teal*). Overtime, these strategies can help organizations to expand their R usage to establish a cost-effective, flexible and both CDSIC & regulatory compliant clinical reporting workflow.

FUTURE DIRECTIONS

Implementation of Additional analyses

One potential avenue for future development is the integration of more advanced analyses in R, such as count data models using generalized linear models (GLMs) via the *mass* package. These implementations can be compared with *PROC GENMOD* in SAS to assess performance and reliability under regulatory scrutiny. As these methods mature, R's credibility and utility as a robust analytical tool alongside SAS will continue to strengthen.

Standardization of R functions and packages

Another area for future growth lies within the development of standardized R functions and packages for model implementation and TFL creation. Developing a set of well-tested, universally accepted functions - similar to SAS macros – can reduce code variability and ensure consistent outputs across teams. Such standardizations will improve R's reliability and reproducibility, further strengthening its role in the pharmaceutical industry's clinical reporting pipeline.

CONTACT INFORMATION

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

Kai Lei
Vertex Pharmaceuticals Inc.
kai_lei@vrtx.com

Jiaqiang Zhu
Vertex Pharmaceuticals Inc.
Jiaqiang_zhu@vrtx.com

Margaret Huang
Vertex Pharmaceuticals Inc.
Margaret_huang@vrtx.com