Mind the Gaps: Automating Multiple Imputation in Clinical Trial Workflows PL022

Crisa Chen, Principal Computational Statistician, Autoimmune,
Eli Lilly and Company

Crisa has over 17 years of experience in clinical trial data analysis and serves as the lead for all Miri pediatric studies. She specializes in ADaM, SDTM, and TFL programming and drives automation and standardization of clinical trial workflows to ensure efficiency, quality, and regulatory compliance.

Mind the Gaps: Automating Multiple Imputation in Clinical Trial Workflows
PL022

Crisa Chen
Jameson Cai

Eli Lilly and Company

Evolving FDA Requirement Prompt MI Macro Creation

Past Practice

- FDA only required sensitivity analyses in TFLs
- MI was implemented within TFL programs

New Requirement

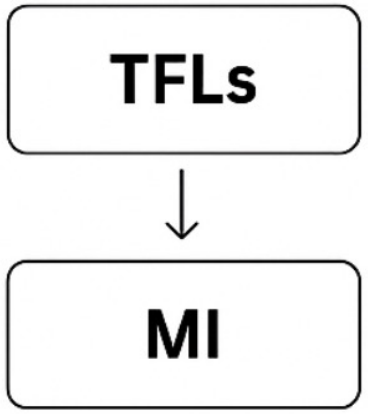
- FDA now requests standalone MI ADaM datasets
- Must be transparent, auditable, and ADaM-compliant

Implication

- Need for a standardized macro that outputs MI ADaM datasets
- Ensures regulatory readiness, consistency, and reusability

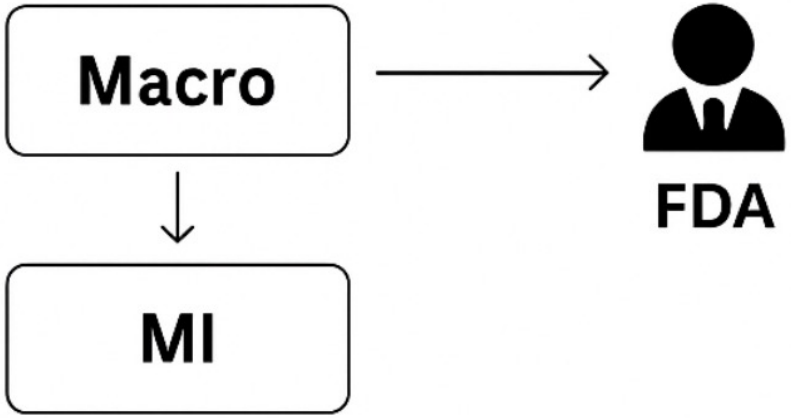
Benefits of the MI Macro

Old Workflow



- MI datasets derived within TFL programs
- Endpoint-specific coding, repeated for each study
- Risk of inconsistencies across endpoints and studies
- Time-consuming and harder to audit for regulatory review

New Workflow with Macro



- MI datasets generated centrally, separate from TFLs
- Reusable and easily adapted to new studies
- Standardized derivation across endpoints and studies
- Transparent, auditable pipeline, FDA-ready outputs
- Reduces coding effort

Applications

- Sensitivity analyses
  - Supports robustness checks of treatment effects under different missing data assumptions
- Handling missing PRO/clinician assessments
  - Ensures unbiased evaluation of key outcomes
- Adaptable to multiple endpoints
  - Adaptable to generate MI datasets across diverse efficacy and safety endpoints

Agenda

- **Demonstration of sensitivity analysis from MI ADaM Datasets**
  - Example tables
  - Using PROC MIXED and PROC MIANALYZE
- **Macro for Automated MI Dataset Generation**
  - Stepwise workflow of %run_MI_pipeline
  - Benefits: reproducibility, transparency, and efficiency
- **Conclusion and Takeaways**
  - Bridging gaps between original ADaM and TFLs
  - Regulatory and cross-study consistency

Sensitivity Analysis from MI ADaM datasets

Example 1:

Illustrative Results (Mock Data):

Visit	Dose 2	Dose 1
Week 12	−1.32 (−2.76, 0.32)	−0.12 (−1.3, 0.03)
Week 24	−1.40 (−2.82, −0.17)	−0.03 (−2.4, 0.41)
Week 36	−1.45 (−2.67, −0.14)	−0.21 (−3.0, −0.01)
Week 52	−1.46 (−3.05, -0.28)	−0.06 (−2.9, 0.82)

Sensitivity Analysis from MI ADaM datasets (cont.)

Example 2: Dose 2

Illustrative Results (Mock Data)

Visit	Estimate	95% CI	p-value	Interpretation
Week 12	−1.32	(−2.76, 0.32)	0.124	Not significant
Week 24	−1.40	(−2.82, −0.17)	0.026	**Significant**
Week 36	−1.45	(−2.67, −0.14)	0.030	**Significant**
Week 52	−1.46	(−3.05, -0.28)	0.011	**Significant**

Sensitivity Analysis from MI ADaM datasets (cont.)

Example 2: Dose 2 vs Dose 1

Illustrative Results (Mock Data)

Visit	Estimate	95% CI	p-value	Interpretation
Week 12	−0.13	(−2.76, 0.16)	0.062	Not significant
Week 24	−0.17	(−2.82, −0.07)	0.043	**Significant**
Week 36	−0.52	(−2.67, −0.14)	0.036	**Significant**
Week 52	−0.89	(−3.05, -0.28)	0.001	**Significant**

Sensitivity Analysis Workflow

Step 1 – PROC MIXED (per imputation)

- Purpose: Estimate treatment effect at each imputed dataset
- Produces LSMeans and treatment differences for each imputation

Step 2 – PROC MIANALYZE (pooling)

- Combine results using Rubin's rules, the standard method for pooling estimates across multiple imputed datasets.

Step 3 – Formatting

LSMeans & Treatment Differences (Per Imputation)

```
ODS OUTPUT LSMEANS=LSMEANS_ DIFFS=DIFFS_;


PROC MIXED DATA=ANA;
        BY PARAMCD IMPUTN AVISITN AVISIT;
        CLASS TRT COVAR1 COVAR2 COVAR3;
        MODEL CHG = TRT COVAR1 COVAR2 COVAR3 BASE / SOLUTION;
        LSMEANS TRT / CL DIFF;


RUN;
```

Pooled LSMeans (MIANALYZE)

```
ODS OUTPUT PARAMETERESTIMATES=LSMEANS_ANALYZE;


PROC MIANALYZE DATA=LSMEANS_;
        BY TRT PARAMCD AVISITN AVISIT;
        MODELEFFECTS ESTIMATE;
        STDERR STDERR;


RUN;
```

Pooled Treatment Differences (MIANALYZE)

```
ODS OUTPUT PARAMETERESTIMATES=DIFFS_ANALYZE;


PROC MIANALYZE DATA=DIFFS_;
        BY PARAMCD AVISITN AVISIT;
        MODELEFFECTS ESTIMATE;
        STDERR STDERR;


RUN;
```

Sensitivity Analysis: MI ADaM Requirement

- **The Problem: Missing Data in Clinical Trials**
  - Patient dropouts, missed visits, or incomplete assessments
  - If ignored, missing data can bias results and reduce power
- **Traditional Approaches**
  - Complete case analysis → discards patients with missing data
  - Last Observation Carried Forward (LOCF) → unrealistic, may bias estimates
- **The Solution: Multiple Imputation (MI)**
  - Fills in missing values with plausible estimates
  - Creates multiple complete datasets
  - Analyzes each dataset, then combines results

Pipeline Overview

Build a reusable SAS macro pipeline:

Input: ADSL + efficacy ADaM (e.g., ADXXXS)

- Step 1: Data exclusions (1st Macro: getflagforMI)
- Step 2: Impute missing values (2nd Macro: %getMIRecords)
- Step 3: Derive analysis variables (3rd Macro: get_MI_vars)
- Step 4: Wrap in master macro (Master Macro: run_MI_pipeline)
- Output: MI-ready ADaM datasets

Step 1: Data exclusions

1st Macro: %getflagforMI:

Intercurrent events (ICEs) in Multiple Imputations:

- Prohibited medication → treatment discontinuation
- Discontinuation due to adverse event (AE)
- Temporary treatment interruptions (other reasons)
- Sporadic missing observations before discontinuation
- Surgery related to underlying disease
- No post-baseline data (except AE-related discontinuation)
- …and other ICEs as specified in the SAP

1st Macro: %getflagforMI

```
%macro getflagforMI
        (INDS=, OUTDS=, PARAMN=, AVISITN=, TABNAME=);


DATA PROHIBMD;
   SET PROHIBMD1;
   ICE2FL = SUBSTR(PROHIBMD, 1, 1);
RUN;
%mend getflagforMI;
```

Step 2: Impute missing values

2nd macro: getMIRecords

Covariates Parameterization:

- Some key categories of covariates commonly included in MI models:
  - Age
  - Geographic region
  - Baseline weight, height, and BMI
  - Prior biologic failure status
  - Baseline corticosteroid use
  - Baseline disease severity scores

2nd macro: getMIRecords

```
%macro getMIRecords
(DSOUT=, AVISITN=, SEED=, MIN_SCORE=, MAX_SCORE=);

PROC TRANSPOSE DATA=MI_TEMP OUT=MI PREFIX=V;
    BY USUBJID TRT COVAR1 COVAR2 COVAR3;
    ID AVISITN;
    VAR AVAL;

RUN;
```

Output data set for MI

Illustrative Results (Mock Data):

	usubjid	TRT01P	covar1	covar2	covar3	V1	V5	V7	V8	V10
1	AAA-BBB-CCCC-103388-128	Dose 1	2	1	1	4	0	3	2	0
2	AAA-BBB-CCCC-105397-130	Dose 2	1	2	1	4	6	3	3	1
3	AAA-BBB-CCCC-106388-131	Dose 2	2	2	1	8	10	10	2	10
4	AAA-BBB-CCCC-116387-141	Dose 1	1	1	1	8		6	7	6
5	AAA-BBB-CCCC-309387-334	Dose 1	2	1	2	11	10	13		
6	AAA-BBB-CCCC-518387-543	Dose 1	1	2	2	8	5	9	11	

intermittent missing Imputation Using PROC MI

```
PROC MI DATA = IN_MONO  OUT  = MONOTONE
      NIMPUTE = 100 SEED    = &SEED.  MINMAXITER = 500
      MINIMUM    = &MIN_SCORE. MAXIMUM    = &MAX_SCORE.;

   EM CONVERGE = 0.001 MAXITER = 3000;

   MCMC CHAIN   = MULTIPLE IMPUTE  = MONOTONE
       INITIAL = EM(CONVERGE = 0.01 MAXITER = 1000000)
       NBITER  = 5000;

   VAR TRT COVAR1 COVAR2 COVAR3 &VISITS.;
RUN;
```

Monotone/consecutive missing Imputation Using PROC MI

```
PROC MI DATA = MONOTONE  OUT  = &DSOUT.
     SEED = &SEED.  NIMPUTE = 1  MINMAXITER = 5000000
     MIN = &MIN_SCORE_STR.
     MAX = &MAX_SCORE_STR.;
   BY _IMPUTATION_;


   CLASS TRT01PN COVAR1 COVAR2 COVAR3;
   VAR TRT01PN COVAR1 COVAR2 COVAR3 &VISITS.;
   MONOTONE REG;
RUN;
```

Output data set

Illustrative Results (Mock Data):

	Imputation	usubjid	covar1	covar2	covar3	V1	V5	V7	V8	V10	trt01p
1	98	AAA-BBB-CCCC-116387-141	1	1	1	8	7.2781936003	6	7	6	Dose 1
2	99	AAA-BBB-CCCC-116387-141	1	1	1	8	6.9781741418	6	7	6	Dose 1
3	100	AAA-BBB-CCCC-116387-141	1	1	1	8	5.4424562037	6	7	6	Dose 1
4	98	AAA-BBB-CCCC-309387-334	2	1	2	11	10	13	9.7853881611	10.491747918	Dose 1
5	99	AAA-BBB-CCCC-309387-334	2	1	2	11	10	13	8.8348846975	8.1436826434	Dose 1
6	100	AAA-BBB-CCCC-309387-334	2	1	2	11	10	13	10.34312456	9.0405676548	Dose 1
7	98	AAA-BBB-CCCC-518387-543	1	2	2	8	5	9	11	12.22748155	Dose 1
8	99	AAA-BBB-CCCC-518387-543	1	2	2	8	5	9	11	9.8059165761	Dose 1
9	100	AAA-BBB-CCCC-518387-543	1	2	2	8	5	9	11	5.7494342083	Dose 1

Monotone Logistic Model for Binary Endpoint

```
PROC MI DATA = MONOTONE  OUT  = &DSOUT
SEED = &SEED.  NIMPUTE = 1  MINMAXITER = 5000000;
   BY _IMPUTATION_;


   CLASS RESPONDER TRT01PN COVAR1 COVAR2 COVAR3;
   VAR RESPONDER TRT01PN COVAR1 COVAR2 COVAR3 &VISITS.;
   MONOTONE LOGISTIC (RESPONDER = TRT01PN COVAR1 COVAR2
COVAR3 &VISITS.);


RUN;
```

Step 3: Derive analysis variables

3rd Macro: get_MI_vars

Adds:

- Change from baseline (CHG)
- Percent change (PCHG)
- Negative percent change (PIMP)
- Labels visits, treatment periods
- Creates analysis-ready ADaM structure

3rd Macro: get_MI_vars

```
%macro get_MI_vars(paramn=);
        if nmiss(AVAL, BASE) = 0 then do;
            CHG = AVAL - BASE;
            if BASE > 0 then PCHG = (AVAL - BASE) / BASE * 100;
            if PCHG ne . then PIMP = -PCHG;
        end;
    run;
%mend get_MI_vars;
```

Step 4: Master controller macro for MI pipeline

```
%MACRO RUN_MI_PIPELINE( PARAMN=, AVISITN=, SEED=,
                        MIN_SCORE=, MAX_SCORE=, INDS=);

   %GETFLAGFORMI( INDS=&INDS,  OUTDS=STEP1OUT,
                  PARAMN=&PARAMN, AVISITN=%STR(&AVISITN) );

   %GETMIRECORDS( DSOUT=PARAM_MI_ANX_&PARAMN,
                   PARAMN=&PARAMN, AVISITN=%STR(&AVISITN),

                  MIN_SCORE=&MIN_SCORE,

                  MAX_SCORE=&MAX_SCORE,

                  SEED=&SEED );

   %GET_MI_VARS( PARAMN=&PARAMN );

%MEND RUN_MI_PIPELINE;
```

Executing Multiple Imputation Workflow:

```
%RUN_MI_PIPELINE(
    INDS      = ADXXX,
    PARAMCD   = ANXIETY,
    MIN_SCORE = 0,
    MAX_SCORE = 21,
    AVISITN   = 1 5 7 8 10,
    SEED      = 123456
);
```

Conclusion

- Automated SAS macros streamline MI for ADaM
- Flexible, transparent, and reproducible
- Facilitates regulatory submissions and cross-study consistency
- By generating MI datasets, we ‘mind the gaps’ between the original ADaM data and the TFL outputs, ensuring complete and consistent analyses.

Thank You!

Any questions?

- Email: chen_xu@lilly.com