

Efficient Strategies for Handling Data Issues in Clinical Trial Submissions

Kate Sun, Mirum Pharmaceuticals, Foster City, CA

Abstract

Clinical trial data is often complex and prone to issues such as missing data, outliers, and inconsistencies. Efficient identification and resolution of these issues are critical to maintaining data integrity and ensuring successful regulatory submissions. This paper presents a professional SAS framework to proactively identify and address data anomalies in the SDTM.LB domain. By utilizing data visualizations, macros, and practical SAS techniques, this approach offers a systematic solution to enhance data quality and streamline the clinical trial submission process. The paper includes SAS code examples, output visualizations, and guidance for statistical programmers to practice.

1. Introduction

We have all experienced the pressure of a looming database lock, only to discover "impossible" lab values or unexpected missingness in key endpoints. Manual data review is no longer a viable strategy in modern trials—it is a risk. Clinical trial data quality directly impacts study validity and regulatory approval. In rare disease research, where sample sizes are small and data points are precious, each observation carries significant weight.

1. **Missing data patterns** (missingness in key endpoints)
2. **Physiologically implausible values** (e.g., negative time intervals)
3. **Protocol deviation outliers** (extreme values requiring clinical review)

Traditional manual quality control methods prove insufficient for modern trials characterized by:

- Complex adaptive designs
- High-frequency biomarker sampling
- Decentralized data collection

Clinical trial data is the backbone of regulatory submissions, and its integrity is paramount. However, data issues such as missing values, outliers, and inconsistencies are common. Traditional methods for handling these issues are often time-consuming and may not scale well with large datasets. This paper proposes a systematic approach to identify and resolve data issues

2. Methodology

2.1 Our three-phase approach:

If we manipulate the data, we must document the "why" and "how" within the dataset itself to satisfy regulatory scrutiny.

Workflow:

1. **Detection:** Automated issue flagging based on statistical and physiological thresholds.
2. **Assessment:** Integrating clinical context (e.g., Normal Ranges) to evaluate relevance.
3. **Resolution:** Documented corrective actions utilizing CDISC-compliant variables.

When generating Analysis Data Model (ADaM) datasets from SDTM—particularly the ADLB domain—transparency is paramount. For any lab values derived or imputed per the Statistical Analysis Plan (SAP), we utilize the DTYPE (Derivation Type) variable. Assigning DTYPE = "DERIVED" or "IMPUTED" ensures

that a reviewer can clearly distinguish between raw clinical observations and statistical derivations.

2.2 Identifying Data Issues Using SAS Macros

The core of this methodology lies in the use of SAS macros to efficiently identify missing values and outliers across large datasets. The logic behind these macros is simple but powerful—once defined, the macros can be applied across any dataset or domain with minimal modification.

For example, the `check_missing` macro checks for missing values in key laboratory variables such as `aval` (lab result values) and `paramcd` (lab test names). This macro generates a frequency table, showing both the count of valid and missing values for each specified variable.

```
/* Macro to identify missing values */
%macro check_missing(
  data=adam.adlb,      /* Input dataset */
  class=paramcd avisit, /* Classification variables */
  var=aval,            /* Analysis variable */
  out=missing_alerts);

/* Calculate missing counts by class variables */
proc means data=&data n nmiss noprint;
  class &class;
  var &var;
  output out=_temp_missing(drop=_type_ _freq_) n=total nmiss=missing;
run;

/* Create final output with percentage calculation */
data &out;
  set _temp_missing;
  where missing > 0;
  pct_missing = (missing/total)*100;
  format pct_missing 5.1;
run;

/* Clean up temporary dataset */
proc datasets library=work nolist;
  delete _temp_missing;
quit;

%mend check_missing;
```

2.3 Outlier detection

```
/* Outlier Detection */
proc univariate data=adam.adlb noprint;
  by paramcd;
  var aval;
  output out=outlier_stats pctlpts=1 99 pctlpre=p_;
run;

data outliers;
  merge adam.adlb outlier_stats;
  by paramcd;
  if (aval < p_1 or aval > p_99) and not missing(aval);
run;
```

3. visualization techniques

3.1 Longitudinal Patient Profiles

We generated longitudinal patient profile plots using SAS's SG PANEL procedure. These plots display individual subjects' lab values (AVAL) across visits (AVISITN) by test (PARAMCD), with reference lines indicating the lower and upper normal limits (ANRLO, ANRHI). This visualization helps identify abnormal patterns or outliers that may require further investigation.

```
/* draw Longitudinal patient profile plot to check outliers */
data attrmap;
  length ID $20 Value $20 LinePattern $12;
  set subjmap;
  ID = "subjStyle";
  Value = USUBJID;

  /* Assign line pattern by SITEID */
  if SITEID = "001" then LinePattern = "solid";
  else if SITEID = "002" then LinePattern = "shortdash";
  else if SITEID = "003" then LinePattern = "dot";
  else if SITEID = "004" then LinePattern = "longdash";
run;

ods graphics / width=7in height=5in;
proc sgpanel data=work.adlb dattrmap=attrmap;
  panelby paramcd / layout=rowlattice;

  series x=avisitn y=aval / group=USUBJID attrid=subjStyle
  lineattrs=(thickness=2);
  scatter x=avisitn y=aval / group=USUBJID markerattrs=(symbol=circlefilled
  size=10);

  refline anrlo / axis=y lineattrs=(color=red pattern=shortdash)
  label="LLN";
  refline anrhi / axis=y lineattrs=(color=red pattern=shortdash)
  label="ULN";
  colaxis label="Visit";
  rowaxis label="Result Value";
  title "Longitudinal Patient Profiles by Site and Subject";
run;
```

3.2 Data Visualizations outputs

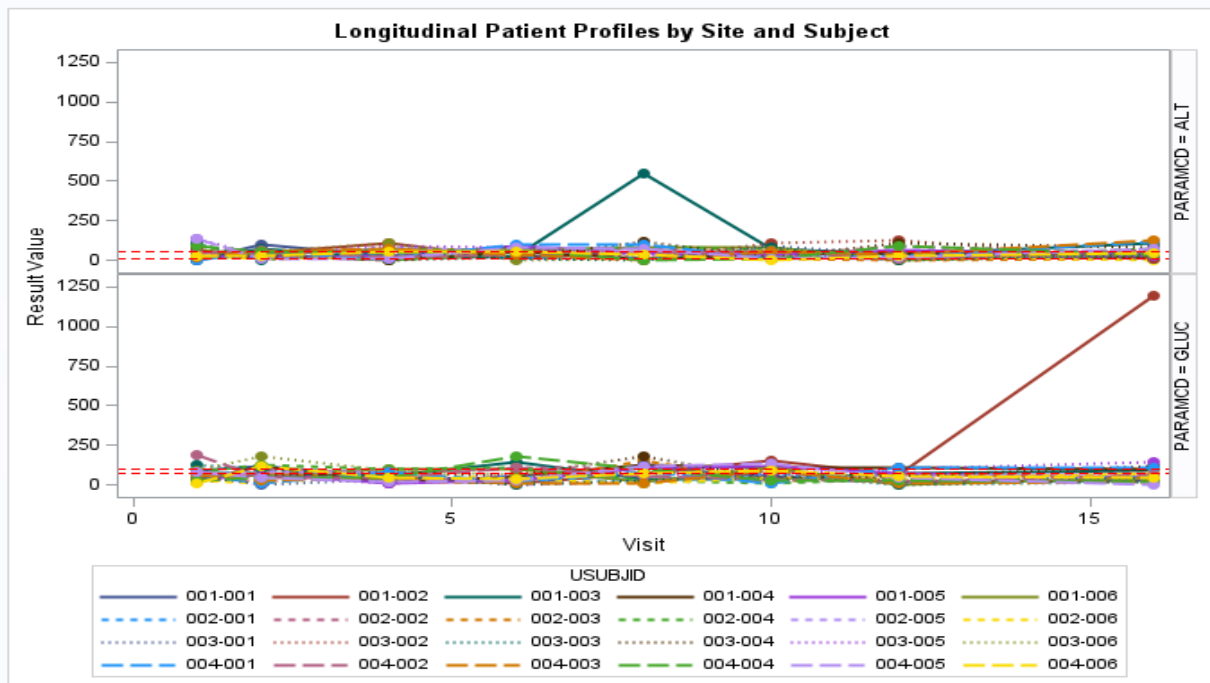


Figure 1. Longitudinal Patient Profile Outlier Detection Output

3.3 Heatmap for Site-Level Data Quality

```
/* Step 1: Identify missing and outlier values */
data adlb_qc;
  set work.adlb;
  is_missing = (aval = .);
  is_upoutlier = (aval > anrhi);
  is_looutlier = (aval < anrlo);
run;

/* Step 2: Summarize by SITEID */
proc sql;
  create table site_perf as
  select
    SITEID,
    count(distinct USUBJID) as num_subjects,
    count(*) as total_records,
    sum(is_missing) as num_missing,
    calculated num_missing / calculated total_records * 100 as pct_missing,
    sum(is_upoutlier) as num_upoutliers,
    calculated num_upoutliers / calculated total_records * 100 as
pct_upoutliers,
    sum(is_looutlier) as num_looutliers,
    calculated num_looutliers / calculated total_records * 100 as
pct_looutliers
  from adlb_qc
  group by SITEID;
quit;
```

```
/* Transform data for heatmap */
data heatmap_data;
  set site_perf;
  length metric $20;

  metric = "Missing %"; value = pct_missing; output;
  metric = "Upper Outlier %"; value = pct_upoutliers; output;
  metric = "Lower Outlier %"; value = pct_looutliers; output;
run;

proc sgplot data=heatmap_data;
  title "Site Performance Heatmap";
  heatmapparm x=metric y=SITEID colorresponse=value / colormodel=(white
red) outline;
  xaxis discreteorder=data label="Metric";
  yaxis discreteorder=data label="Site ID";
run;
```

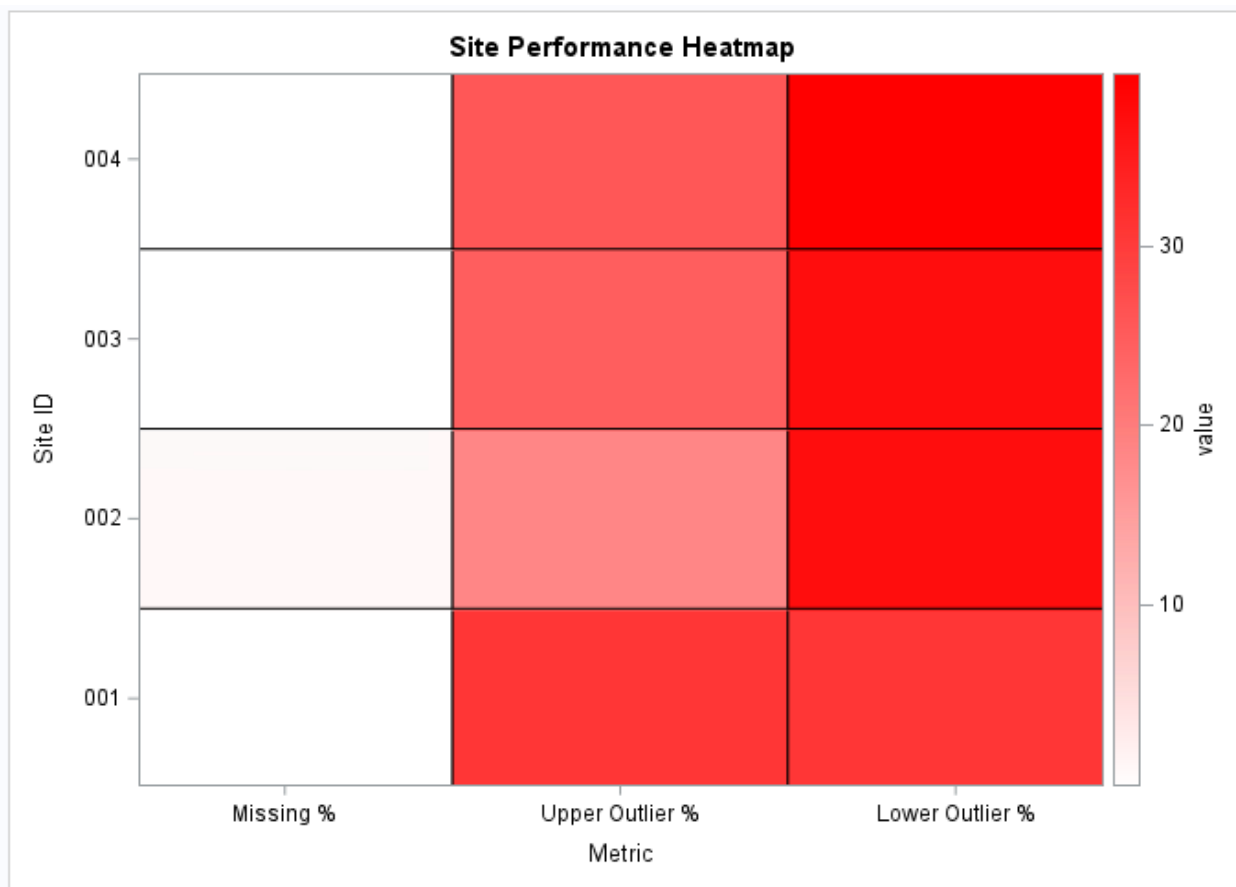


Figure 2. Site Performance Heatmap

4. Handling Missing Data: A Conservative Imputation Approach

4.1 Rationale for Mean Imputation

Handling missing data is a critical aspect of clinical trial data analysis, as incomplete datasets can introduce bias and reduce statistical power. In this study, we addressed missing laboratory values using a simple mean imputation approach. While several imputation techniques exist, including Last Observation Carried Forward (LOCF), regression methods, and multiple imputation, we chose mean imputation for its simplicity and transparency. The specific imputation methodology for production datasets is typically governed by the Statistical Analysis Plan (SAP) and may vary based on study design and regulatory requirements.

4.2 Implementation Methodology

```
/* Step 1: Calculate mean AVAL per PARAMCD and SITEID */
proc means data=adlb_qc noprint;
  class SITEID PARAMCD;
  var aval;
  output out=mean_vals mean=mean_aval;
run;

/* Step 2: Merge mean values back and impute */
proc sql;
  create table adlb_imputed as
  select a.*,
         b.mean_aval,
         case
           when a.aval = . then b.mean_aval
           else a.aval
         end as aval_imputed,
         (a.aval = .) as imputed,
         "DERIVED" as DTYPE,
         "Y" as LBDERFL
  from adlb_qc a
  left join mean_vals b
    on a.SITEID = b.SITEID and a.PARAMCD = b.PARAMCD;
quit;
```

5. Conclusion

In clinical trial analysis, missing laboratory values present a significant challenge, particularly in rare disease studies where small sample sizes amplify the impact of each data point. Our methodology employs **site- and parameter-specific mean imputation** as a conservative approach that:

- Maintains the overall distribution of laboratory values
- Preserves between-site variability
- Minimizes artificial inflation of variance
- Aligns with common statistical analysis plan (SAP) recommendations for continuous laboratory parameters

References

Statistical Analysis with Missing Data (3rd ed.). <Little, R.J.A. & Rubin, D.B.> (2019). Wiley.

EMA Guideline on Missing Data. (2010). EMA/CPMP/EWP/1776/99 Rev. 1.

Principles and Applications of Sensitivity Analyses for Clinical Trials. < Mallinckrodt, C.H. et al.> (2013)

CDISC ADaM Implementation Guide v1.3. (2021).

Contact Information

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

Name: Kate Sun
Enterprise: Mirum Pharmaceuticals
Address: 989 E Hillsdale Blvd., Suite 300
City, State ZIP: Foster City, CA, 94404
E-mail: kate.sun@mirumpharma.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

Source Code Sample

```
Example Usage:
%check_missing(
  data=adam.adlb,
  class=paramcd avisitn,
  var=aval,
  out=work.missing_report);
```

Output Sample

PARAMCD	AVISIT	total	missing	pct_missing
ALT	Week 4	35	3	8.6
GLUC	Week 12	35	7	20.0