PharmaSUG 2026 - Paper AP-258

From Tables to Tolerances: The Evolving Role of Statistical Programmers in Risk-Based Quality Management (RBQM)

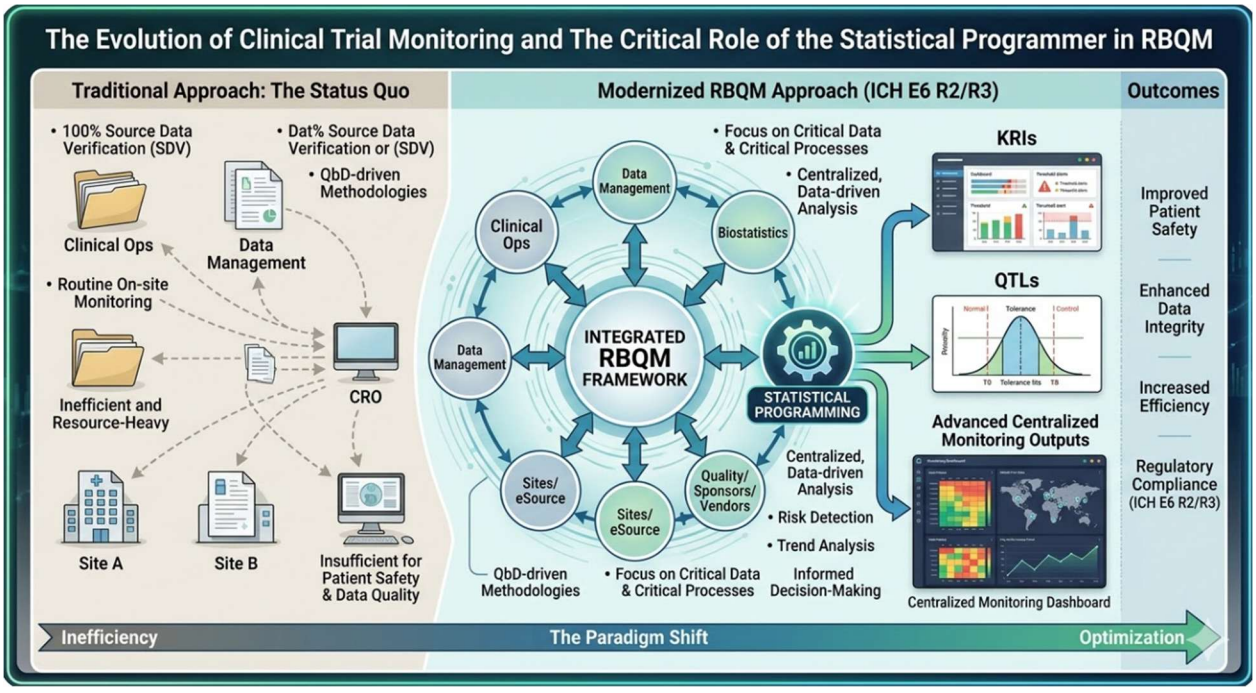
Vihar Patel, Thermo Fisher Scientific, PPD Clinical Research Services

ABSTRACT

Risk-Based Quality Management (RBQM) has become a regulatory and operational expectation in modern clinical trials with the evolution of ICH E6(R2) and E6(R3), together with the Quality by Design (QbD) framework introduced in ICH E8(R1). While RBQM is often viewed as the responsibility of Clinical Operations, Data Management, and Quality Assurance, statistical programmers play an increasingly vital role in translating risk concepts into measurable and actionable insights. They also act as integrators across Biostatistics, Data Management, and external vendors by assessing the feasibility of key metrics such as Key Risk Indicators (KRIs) and Quality Tolerance Limits (QTLs), as well as the suitability and availability of underlying data sources. This paper presents a role-specific competency model describing the skill sets required for statistical programmers to effectively support RBQM activities. Using a fully synthetic Phase II clinical trial dataset, the paper illustrates the programming frameworks used to compute KRIs, QTLs, and statistical risk signals used in Centralized Statistical Monitoring (CSM). The paper also describes collaboration points with cross-functional teams during risk assessment and review. These examples show how programmers can elevate RBQM from a conceptual framework to a practical, data-driven system that strengthens study oversight, supports regulatory expectations, and improves overall trial quality.

INTRODUCTION

Risk-Based Quality Management (RBQM) represents a fundamental shift in how sponsors and CROs monitor clinical trials. Traditional oversight models relied heavily on 100% source data verification and routine on-site monitoring, practices now recognized as inefficient and insufficient for ensuring patient safety and data quality. With the introduction of ICH E6(R2) and the ongoing evolution toward E6(R3), the industry has moved toward Quality by Design (QbD) principles that emphasize critical data and processes, supported by centralized, data-driven approaches.



Despite this progress, the role of statistical programming in RBQM remains underrepresented in literature. Statistical programmers have direct access to structured and continuously evolving clinical and operational data streams that underpin RBQM and are uniquely positioned to translate these data into KRIs, QTLs, and advanced centralized monitoring outputs. Because they operate at the intersection of Data Management, Biostatistics, Clinical Operations, Quality, and external vendors, programmers often serve as the practical integrators of RBQM. Working across data domains and systems, they assess metric feasibility, identify data gaps, and enable early detection of emerging risks, supporting more informed and timely decision-making.

This paper demonstrates how statistical programmers contribute to RBQM through automation, advanced analytics, and cross-functional collaboration. It also introduces a competency model to support the development of programming teams operating in RBQM environments.The focus is on practical implementation, emphasizing approaches that are both analytically sound and operationally feasible in real-world study environments.

STUDY AND DATASET DESCRIPTION

A fully synthetic Phase II clinical trial dataset was created to demonstrate RBQM analytics. The study design mirrors a typical Type 2 Diabetes trial with approximately 400 subjects across 25 investigative sites. Subjects were randomized to either Active treatment or Placebo and followed for 24 weeks.

The synthetic dataset includes both clinical and operational domains:

- ADSL – subject-level demographics, randomization date, and treatment assignment
- ADLB – laboratory data including HbA1c at baseline and Week 24
- ADAE – adverse events with onset date and entry date to calculate reporting timeliness
- QUERY_LOG – operational dataset containing query open and close dates
- Protocol deviation and enrollment data (not shown but available for additional KRIs)

The dataset was simulated to incorporate realistic site-level patterns including late AE reporting, prolonged query aging, missing primary endpoint data, and sites with anomalous HbA1c variability, all of which provide meaningful inputs for RBQM metrics.
A detailed description of the variable structures and simulation methods will be provided in the Appendix with complete SAS and R code.

METHODS

This section outlines the programming methodology used to derive KRIs, QTLs, and centralized statistical monitoring (CSM) metrics from the synthetic dataset.

KRI METHODOLOGY

Key Risk Indicators (KRIs) were selected to reflect common clinical and operational risk domains:

- AE Reporting Timeliness: Median number of days between AE start and entry into EDC
- Query Aging: Percentage of queries open for more than 30 days
- Enrollment and Randomization Performance: Site-level randomized subject counts and enrollment rates over time, used to identify sites with unusually low, high, or rapidly changing enrollment patterns.
- Protocol Deviation Rates: Number of deviations per subject per site

These KRIs were programmed in SAS and R to support automated, site-level reporting, with predefined thresholds used to flag sites for RBQM reviews.

QTL METHODOLOGY

The primary QTL for the synthetic dataset was the missingness of the Week 24 HbA1c endpoint. A breach threshold of 5% missingness was defined. (For illustrative purposes only, a missingness threshold

should be defined based on study-specific risk considerations rather than a fixed standard.) Missing rates were computed programmatically, with logic to trigger alerts when the threshold was exceeded.

CENTRALIZED STATISTICAL MONITORING METHODOLOGY

Centralized Statistical Monitoring (CSM) was used to identify atypical patterns in site-level data relative to the overall study population. HbA1c baseline and change-from-baseline at Week 24 were analyzed across sites. Multivariate outlier detection using Mahalanobis distance was used to identify sites with unusual means or variances. This example illustrates the type of analytics that can support centralized monitoring.
Programmatic outputs include:

- Site-level mean and standard deviation of change from baseline
- Mahalanobis distance per site with chi-square cut off thresholds
- Ranking of sites by statistical deviation
- Heatmaps for visual comparison

PROGRAMMING AND AUTOMATION STRATEGY

All KRIs, QTLs, and CSM metrics were programmed using standard SAS macros and R functions, guided by the following principles:

- Repeatability – consistent logic applied across all data refresh cycles
- Traceability – fully documented algorithms and derivations
- Auditability – clear logs, QC outputs, and parameterized code

Technology stack:

- **SAS 9.4** – primary engine for clinical and operational analytics
- **R** – visualization and advanced statistical methods

Operational considerations:

- Automation improves efficiency and consistency but requires ongoing maintenance
- Programmers routinely:
    - Calibrate KRI thresholds
    - Update QTL rules
    - Incorporate protocol amendments
    - Adjust data-source logic as vendor feeds evolve

Sustainability of RBQM systems depends on:

- Periodic recalibration of thresholds and rules
- Continuous alignment with evolving data sources and study design

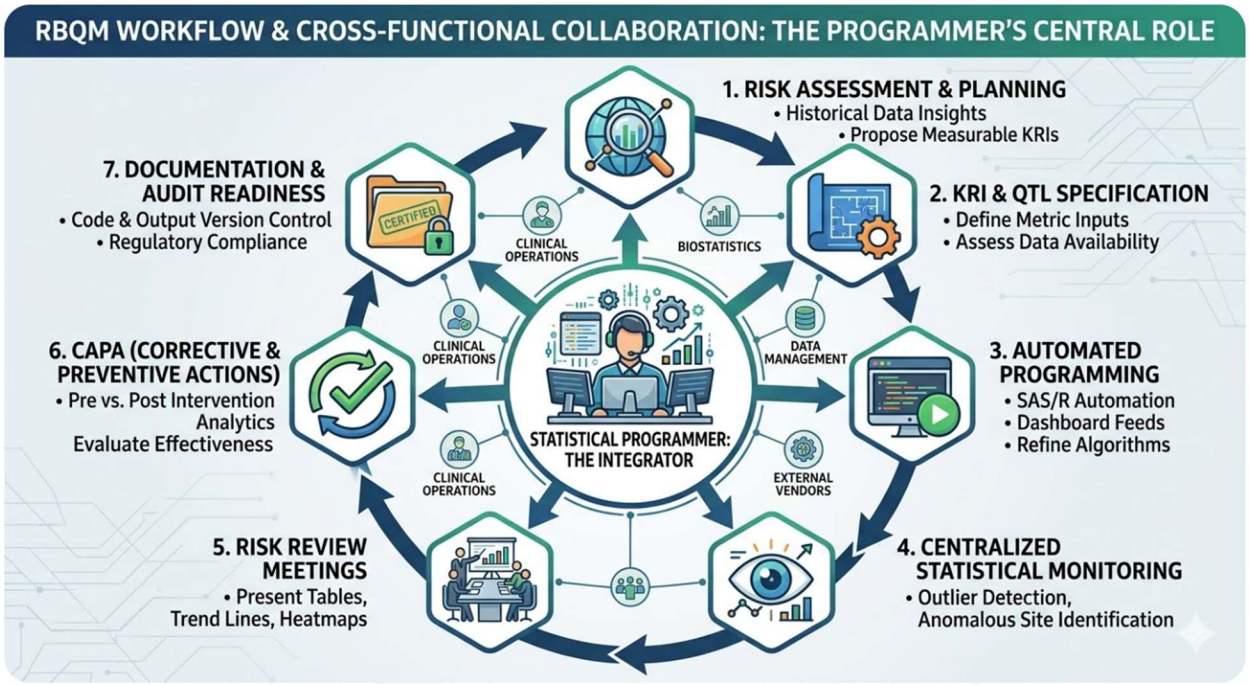
Real-world implementation challenges:

- Data latency across systems
- Heterogeneous vendor data streams
- Evolving metadata standards

These constraints require **flexible, iterative programming approaches** to ensure that risk signals remain accurate, relevant, and operationally actionable.

RBQM WORKFLOW & CROSS-FUNCTIONAL COLLABORATION

Statistical programmers contribute to RBQM across the full study lifecycle. These contributions span planning, execution, monitoring, and continuous improvement activities.



- **Risk Assessment and Planning:** Programmers assist statisticians and ClinOps by providing historical data insights and proposing measurable KRIs.
- **KRI and QTL Specification:** Programmers work with the RBQM core team to define metric inputs, thresholds, and calculation frequency. Drawing on their integrator role across Biostats, Data Management, Clinical Operations, and external vendors, they assess data availability, understand transfer and metadata constraints, and advise on the feasibility of proposed KRIs and QTLs before metrics are finalized.
- **Automated Programming of KRIs and QTLs:** Programmers develop SAS macros and R scripts that compute metrics regularly and feed dashboards. Automation is not static; programmers perform continual refinement of RBQM algorithms as real study data accumulates, thresholds need recalibration, or data availability evolves. For QTLs in particular, dashboards support medical monitors in defining expectations and tracking QTL status and trends over time.
- **Centralized Statistical Monitoring:** Programmers perform outlier detection and exploratory analytics to help identify anomalous sites.
- **Risk Review Meetings:** Programmers present tables, trend lines, heatmaps, and flagged sites to the RBQM committee.
- **Corrective and Preventive Actions (CAPA):** Programmers support CAPA evaluation by comparing pre- and post-intervention metrics.
- **Documentation and Audit Readiness:** Programmers ensure that code, outputs, and version control meet regulatory and quality expectations.

RESULTS

The synthetic dataset illustrates several realistic and commonly observed risk scenarios.

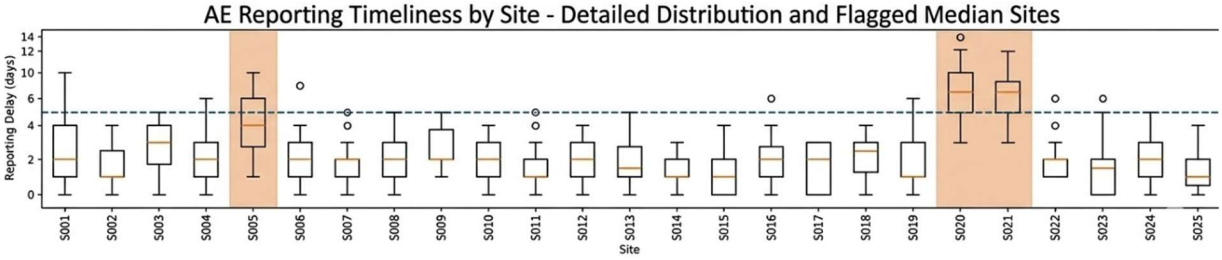
AE REPORTING TIMELINESS

Two sites (S020 and S021) exhibited median AE reporting delays exceeding the 5-day threshold, driven by consistently late entry of events. To provide context and prevent misinterpretation of sparse data, the table includes the number of subjects enrolled at each site and the number of subjects contributing at least one AE. Sites S005, S020, and S021 show elevated delays even after accounting for their AE-contributing population.

AE Reporting Characteristics and Risk Assessment for Selected Sites

Site ID	Subjects Enrolled	Subjects with ≥1 AE	Total AEs	Median Delay (days)	75th Percentile Delay (days)	Threshold	Risk Flag
S001	18	12	17	3	5	≤5	0
S005	**21**	**16**	**21**	**7**	**9**	**>5**	**1**
S010	15	10	14	2	4	≤5	0
S014	20	13	20	4	8	≤5	0
S018	15	13	18	5	8	≤5	0
S020	**23**	**19**	**25**	**12**	**16**	**>5**	**1**
S021	17	14	19	**10**	13	≤5	**1**

AE Reporting Timeliness by Site - Detailed Distribution and Flagged Median Sites



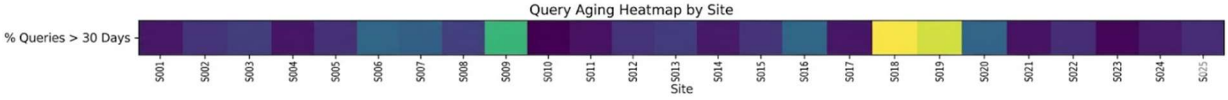
QUERY AGING

Sites S018 and S019 showed more than 30% of open queries older than 30 days. Sites S009, S018, and S019 remain high-risk even after accounting for their contributing subject populations.

Detailed Site Data and Query Aging Overview

Site ID	Subjects Enrolled	Subjects with ≥ 1 Query	Total Queries	Queries >30 Days	% >30 Days	Threshold	Risk Flag
S002	16	9	**33**	4	12.10%	≤20%	0
S006	17	11	**26**	5	19.20%	≤20%	0
S009	18	13	**29**	7	**24.10%**	>20%	**1**
S018	15	12	**30**	11	**36.70%**	>20%	**1**
S019	14	11	**28**	9	**32.10%**	>20%	**1**

Query Aging Heatmap by Site



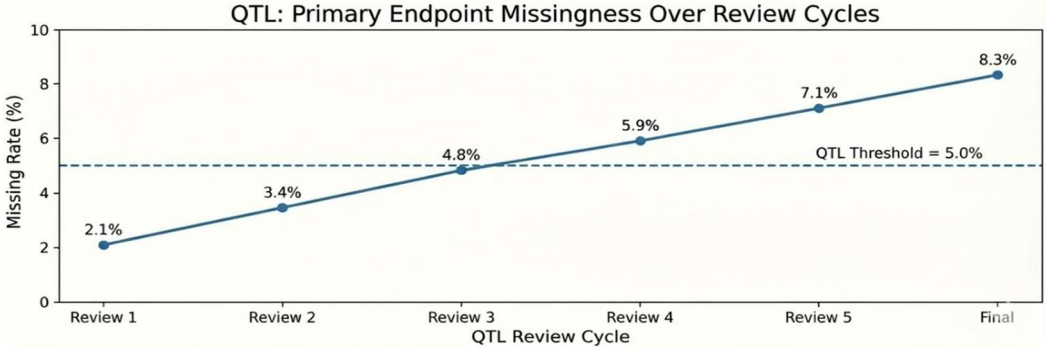
QTL ASSESSMENT

The Week 24 HbA1c missing rate was 8.3%, surpassing the 5% QTL threshold and requiring formal review.

Comprehensive QTL Monitoring: Table and Trend

Parameter	Total Subjects Randomized	Subjects with Evaluable HbA1c	Subjects Missing Wk 24 HbA1c	Missing Rate	QTL Threshold	QTL Breach
HbA1c (Week 24)	400	367	33	**8.30%**	5%	**Yes**

QTL: Primary Endpoint Missingness Over Review Cycles



CENTRALIZED STATISTICAL MONITORING RESULTS

Site-level summary statistics for HbA1c change from baseline were reviewed to assess variability across sites. While most sites showed consistent patterns, selected sites demonstrated atypical variability (e.g., low variability at S017 and higher variability at S020), highlighting areas for further RBQM review.

Site ID	N	Mean Baseline	Mean Change	SD Change	Comment
S001	14	8.6	-0.9	0.7	—
S007	18	8.4	-1.1	0.6	Consistent
S010	15	8.5	-0.2	0.8	Higher variance
S017	12	8.7	-1.7	0.1	**Low variability**
S020	16	8.3	-0.4	1.3	High variability

Mahalanobis distance and heatmap visualizations identified two sites exceeding the predefined statistical cutoff. Site S017 exhibited unusually low variability in HbA1c change from baseline (N = 12), which may indicate potential data quality concerns or atypical subject populations. In contrast, Site S020 (N = 16) showed elevated variability relative to the pooled study population, suggesting increased heterogeneity or potential operational differences that warrant further review. Considering both variability patterns and contributing sample size supports more robust interpretation of centralized monitoring signals.

Rank	Site ID	N (Subjects)	Mahalanobi	Chi-Square Cutoff (99%,	Flag	Comment
1	**S017**	12	**13.8**	**≥ 9.21**	**Flagged**	Unusually low variability; potential data quality concern
2	**S020**	16	**10.5**	**≥ 9.21**	**Flagged**	Elevated variability; warrants review
3	S010	15	8.7	**< 9.21**	—	Borderline but below cutoff
4	S001	14	4.4	**< 9.21**	—	Typical pattern
5	S007	18	3.2	**< 9.21**	—	Typical pattern

In addition to summary tables and graphical outputs, the automated RBQM programming pipeline generates subject-level listings for each KRI, QTL, and CSM signal. These listings support RBQM meetings by allowing cross-functional teams to drill down into the underlying data, verify contributing records, and assess whether further investigation or CAPA is warranted. Flagged signals were reviewed within RBQM governance forums, where cross-functional teams evaluated root causes and determined appropriate actions, including targeted monitoring, data review, or CAPA implementation.

DISCUSSION

These results demonstrate how automated analytics and centralized monitoring frameworks enable more proactive, data-driven risk detection and oversight.

While this paper focuses on foundational centralized monitoring techniques, more advanced statistical approaches (e.g., mixed-effects models or Bayesian frameworks) may further enhance signal detection. In practice, however, the methods presented here prioritize interpretability, scalability, and operational feasibility within RBQM workflows.

The competency model reflects the increasing expectation for programming teams to extend beyond traditional deliverables into advanced analytics and data integration.

The synthetic examples replicate common issues observed in real trials (delayed AE reporting, prolonged query aging, endpoint missingness, and anomalous site behavior) and demonstrate how these risks can be systematically detected, monitored, and communicated within an RBQM framework by programmers.

Competency Domain	Foundational	Intermediate	Advanced
1. RBQM & QbD Knowledge	Understand RBQM concepts (KRIs, QTLs, CSM) and basic QbD/CtQ principles.	Translate CtQ risks into measurable indicators; interpret KRI patterns.	Shape RBQM strategy; define CtQ-driven KRIs/QTLs; guide cross-functional risk discussions.

Competency Domain	Foundational	Intermediate	Advanced
2. Data Engineering & Integration	Merge and validate EDC/SDTM/ADaM data.	Build automated pipelines for CtQ-related data; integrate clinical + operational sources.	Develop scalable, study-agnostic RBQM data infrastructures; advise on data availability and feasibility of CtQ-driven KRIs/QTLs across vendor and CDM systems.
3. KRI/QTL Programming	Compute basic KRIs; ensure accurate rates and counts.	Implement CtQ-driven KRIs; program dynamic thresholds and trend monitoring.	Build cross-study KRI/QTL libraries; implement advanced anomaly detection; maintain and recalibrate automated metrics.
4. Centralized Statistical Monitoring (CSM)	Perform basic descriptive and outlier checks.	Conduct CtQ-sensitive variability, comparability, and diary-quality analyses.	Develop full multivariate CSM models and statistical signal detection.
5. Collaboration & Communication	Explain metrics; join RBQM discussions.	Communicate CtQ/RBQM trends to ClinOps, DM, Stats, and QA.	Lead RBQM analytics reviews; influence CAPA decisions with data insights; Bridge Biostats, CDM, and external vendors by assessing feasibility of requested metrics and data sources.
6. Compliance & Documentation	Follow SOPs; maintain version control.	Document KRIs/QTLs with CtQ mapping; ensure traceability.	Produce audit-ready RBQM documentation and validated code packages.

CONCLUSION

RBQM is now a foundational component of modern clinical trial oversight, requiring consistent application of data-driven monitoring throughout the study lifecycle. This paper highlights how statistical programmers can expand their role beyond traditional output generation to actively interpret data patterns, recognize emerging risks, and contribute meaningfully to cross-functional RBQM discussions.

Working across clinical and operational data domains, programmers can identify signals related to missingness, variability, operational feasibility, and data integrity early in the study lifecycle. Combined with automated monitoring tools and centralized review processes, these insights enable earlier and more targeted risk detection.

As RBQM continues to mature across the industry, empowering programmers with an analyst mindset will strengthen collaboration across functions and enhance the overall quality of study oversight. By embedding analytical thinking into routine programming activities, organizations can unlock greater value from existing expertise and support more proactive, data-driven quality management across the clinical trial lifecycle.

REFERENCES

- ICH E6(R2) Guideline for Good Clinical Practice
- ICH E6(R3) Principles
- FDA Guidance on Risk-Based Monitoring of Clinical Investigations
- Risk Based Quality Management - WORKING GROUPS - PHUSE Advance Hub
- CTTI Homepage - CTTI
- EMA Reflection Paper on Risk-Based Quality Management

ACKNOWLEDGMENT

I would like to express my heartfelt gratitude to my colleagues for their invaluable support and collaboration. Special thanks to Francesco Barreta, Rakhi Kilaru, Ted Cherico, Megan Hewitt, Yudong Zhao, Ivy Ni, and Sajeet Pavate for their insightful feedback and unwavering encouragement.

CONTACT INFORMATION

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

Vihar Patel
Thermo Fisher Scientific, PPD Clinical Research Services
201-736-2760
viharpatel64@gmail.com

DISCLAIMER

The opinions expressed in this paper and related slides for the presentation are solely those of the presenter and not necessarily those of Thermo Fisher Scientific. Thermo Fisher Scientific does not guarantee the accuracy or reliability of the information provided herein.
Brand and product names are trademarks of their respective companies.

Appendix A: SAS 9.4M6 Code

- **A.1 Importing Synthetic Data**

```
/* Adjust this path */
%let datapath = C:\RBQM_SimData;
libname simdata "&datapath.";
proc import datafile="&datapath.\adsl_sim.csv"
    out=simdata.adsl dbms=csv replace;
    guessingrows=max;
run;
proc import datafile="&datapath.\adlb_sim.csv"
    out=simdata.adlb dbms=csv replace;
    guessingrows=max;
run;
proc import datafile="&datapath.\adae_sim.csv"
    out=simdata.adae dbms=csv replace;
    guessingrows=max;
run;
proc import datafile="&datapath.\query_log_sim.csv"
    out=simdata.query_log dbms=csv replace;
    guessingrows=max;
run;
/* "Today" date for KRI calculations */
%let today_dt = '01NOV2024'd;
```

- **A.2 KRI – AE Reporting Timeliness**

```
/* AESTDTC/AERECDTC assumed as DATE format after import */
data ae_delay;
    set simdata.adae;
    format AESTDTC AERECDTC date9.;
    ae_delay = AERECDTC - AESTDTC;
    /* Clean negative or missing delays */
    if ae_delay < 0 then ae_delay = .;
run;
/* Site-level statistics */
proc means data=ae_delay noprint;
    class SITEID;
    var ae_delay;
    output out=kri_ae_timeliness
        n=n_ae
        median=med_delay
        p75=p75_delay;
run;
data kri_ae_timeliness;
    set kri_ae_timeliness;
    where _TYPE_ = 1;           /* Keep only site-level rows */
    length risk_flag 8;
    if med_delay > 5 then risk_flag = 1;
    else risk_flag = 0;
run;
proc sort data=kri_ae_timeliness;
    by descending med_delay;
run;
title "KRI: AE Reporting Timeliness by Site";
proc print data=kri_ae_timeliness label noobs;
    var SITEID n_ae med_delay p75_delay risk_flag;
    label n_ae      = "Number of AEs"
          med_delay = "Median Delay (days)"
          p75_delay = "75th Percentile Delay (days)"
          risk_flag = "Risk Flag (>5 days median)";
run;
title;
```

- **A.3 KRI – Query Aging (>30 Days Open)**

```
data queries;
    set simdata.query_log;
    format OPEN_DT CLOSE_DT date9.;
    length days_open 8;
/* Effective close date = CLOSE_DT if not missing, else today */
    if missing(CLOSE_DT) then close_eff = &today_dt.;
    else close_eff = CLOSE_DT;
if OPEN_DT ne . then days_open = min(&today_dt., close_eff) - OPEN_DT;
    else days_open = .;
long_open = (days_open > 30);
run;
proc sql;
    create table kri_query_aging as
    select SITEID,
           count(*)                          as total_queries,
           sum(long_open)                    as n_long_open,
           calculated n_long_open / calculated total_queries as prop_long_open
    from queries
    group by SITEID;
quit;
data kri_query_aging;
    set kri_query_aging;
    length risk_flag 8;
    if prop_long_open > 0.20 then risk_flag = 1;
    else risk_flag = 0;
run;
title "KRI: Query Aging by Site (>30 Days Open)";
proc print data=kri_query_aging label noobs;
    var SITEID total_queries n_long_open prop_long_open risk_flag;
    label total_queries  = "Total Queries"
          n_long_open    = "Queries > 30 Days"
          prop_long_open = "% > 30 Days"
          risk_flag      = "Risk Flag (>20%)";
    format prop_long_open percent8.1;
run;
title;
```

- **A.4 KRI – Enrollment Performance**

```
proc sql;
    create table kri_enrollment as
    select SITEID,
           count(*) as n_randomized,
           16 as expected_enrollment,
           calculated n_randomized / calculated expected_enrollment as pct_expected
    from simdata.adsl
    group by SITEID;
quit;
data kri_enrollment;
    set kri_enrollment;
    if pct_expected < 0.80 then risk_flag = 1;
    else risk_flag = 0;
run;
title "KRI: Enrollment Performance by Site";
proc print data=kri_enrollment label noobs;
    var SITEID n_randomized expected_enrollment pct_expected risk_flag;
    label n_randomized       = "Randomized Subjects"
          expected_enrollment= "Expected Enrollment"
          pct_expected        = "% of Expected"
          risk_flag          = "Risk Flag (<80%)";
    format pct_expected percent8.1;
run;
title;
```

- **A.5 QTL – Week 24 HbA1c Missingness**

```
/*Assume ADLB has one record per subject for HbA1c, baseline and Week 24 (AVALBL,
AVALW24)*/
data hb;
    set simdata.adlb;
    where PARAM = "HbA1c";
    miss_w24 = missing(AVALW24);
run;
proc sql noprint;
    select sum(miss_w24) as n_missing,
           count(*)      as n_total,
           calculated n_missing / calculated n_total
    into :n_missing, :n_total, :miss_rate
    from hb;
quit;
%put NOTE: Missing Week 24 HbA1c: &n_missing of &n_total subjects (&miss_rate.);
%let qtl_threshold = 0.05;
%macro check_qtl;
    %if &miss_rate. > &qtl_threshold. %then %do;
        %put WARNING: QTL BREACH - Primary endpoint missingness exceeds 5%;
    %end;
    %else %do;
        %put NOTE: QTL OK - Primary endpoint missingness within limit.;
    %end;
%mend;
%check_qtl;
```

- **A.6 CSM – Data Preparation for HbA1c Change**

```
proc sql;
    create table hb_subj as
    select USUBJID,
           SITEID,
           TRTGRP,
           AVALBL,
           AVALW24,
           (AVALW24 - AVALBL) as CHG
    from simdata.adlb
    where PARAM = "HbA1c";
quit;
```

- **A.7 CSM – Pooled Covariance and Means (SAS)**

```
proc corr data=hb_subj cov noprint outp=covmat;
    var AVALBL CHG;
run;
proc print data=covmat;
    title "Pooled Covariance and Means (AVALBL, CHG)";
run;
title;

/* Extract means into macro variables */
data means;
set covmat;
if _TYPE_ = 'MEAN';
call symputx('mean_base', AVALBL);
call symputx('mean_chg', CHG);
run;

/* Extract covariance elements */
data cov;
set covmat;
```

```
if _TYPE_ = 'COV';
run;

/* For a full SAS MD implementation: Extract var/cov elements,Invert the covariance
matrix, and Compute MD per site.
Because R handles this more simply, we implement the full MD in R*/
```

Appendix B: R Code

- **B.1 Simulating the Synthetic Dataset**

```
set.seed(123)

n_sites  <- 25
n_subj   <- 400
sites    <- paste0("S", sprintf("%03d", 1:n_sites))

# ADSL: subject-level
adsl <- data.frame(
  USUBJID = paste0("SUBJ", sprintf("%04d", 1:n_subj)),
  SITEID  = sample(sites, n_subj, replace = TRUE),
  TRTGRP  = sample(c("Placebo","Active"), n_subj, replace = TRUE),
  RANDDT  = as.Date("2024-01-01") + sample(0:120, n_subj, replace = TRUE)
)

# Site-level effect on treatment response
site_effect <- rnorm(n_sites, mean = 0, sd = 0.3)
names(site_effect) <- sites

# ADLB: HbA1c baseline and Week 24
adlb <- do.call(rbind, lapply(1:nrow(adsl), function(i) {
  s   <- adsl$SITEID[i]
  trt <- adsl$TRTGRP[i]
  base_hba1c <- rnorm(1, 8.5, 1.0)
  trt_effect <- ifelse(trt == "Active", -1.0, -0.2)
  wk24_hba1c <- base_hba1c + trt_effect + site_effect[s] + rnorm(1, 0, 0.6)

  data.frame(
    USUBJID = adsl$USUBJID[i],
    SITEID  = s,
    TRTGRP  = trt,
    PARAM   = "HbA1c",
    AVALBL  = base_hba1c,
    AVALW24 = wk24_hba1c
  )
}))

# ADAE: simulate AE reporting and delays
adsl$n_ae <- rpois(n_subj, lambda = 1.2)

adae_list <- lapply(1:nrow(adsl), function(i) {
  n <- adsl$n_ae[i]
  if (n == 0) return(NULL)
  randdt <- adsl$RANDDT[i]
  site   <- adsl$SITEID[i]

  do.call(rbind, lapply(1:n, function(j) {
    ae_start <- randdt + sample(1:90, 1)
    site_delay_shift <- ifelse(site %in% c("S020","S021"), 7, 2)
    delay    <- max(rpois(1, lambda = site_delay_shift), 0)
    ae_rec   <- ae_start + delay

    data.frame(
      USUBJID   = adsl$USUBJID[i],
      SITEID    = site,
      AEDECOD   = sample(c("Headache","Nausea","Hypoglycemia","Dizziness"), 1),
```

```
      AESTDTC   = ae_start,
      AERECDTC  = ae_rec
    )
  }))
})

adae <- do.call(rbind, adae_list)

# QUERY_LOG: simulate query aging
query_list <- lapply(1:nrow(adsl), function(i) {
  n <- rpois(1, lambda = 3)
  if (n == 0) return(NULL)
  randdt <- adsl$RANDDT[i]
  site   <- adsl$SITEID[i]

  do.call(rbind, lapply(1:n, function(j) {
    open_date <- randdt + sample(5:60, 1)
    base_close_shift <- ifelse(site %in% c("S018","S019"), 45, 15)
    close_delay <- rpois(1, base_close_shift)
    close_date  <- ifelse(runif(1) < 0.2, NA, open_date + close_delay)

    data.frame(
      USUBJID   = adsl$USUBJID[i],
      SITEID    = site,
      QUERY_ID  = paste0("Q", i, "_", j),
      OPEN_DT   = open_date,
      CLOSE_DT  = as.Date(close_date, origin = "1970-01-01")
    )
  }))
})

query_log <- do.call(rbind, query_list)

# Save to CSV for SAS
write.csv(adsl,      "adsl_sim.csv",      row.names = FALSE)
write.csv(adlb,      "adlb_sim.csv",      row.names = FALSE)
write.csv(adae,      "adae_sim.csv",      row.names = FALSE)
write.csv(query_log, "query_log_sim.csv", row.names = FALSE)
```

- **B.2 KRI - AE Reporting Timeliness in R**

```
 library(dplyr)
adae_kri <- adae %>%
  mutate(ae_delay = as.numeric(AERECDTC - AESTDTC)) %>%
  group_by(SITEID) %>%
  summarise(
    n_ae      = n(),
    med_delay = median(ae_delay, na.rm = TRUE),
    p75_delay = quantile(ae_delay, 0.75, na.rm = TRUE),
    risk_flag = ifelse(med_delay > 5, 1, 0)
  ) %>%
  arrange(desc(med_delay))
print(adae_kri)
```

- **B.3 KRI - Query Aging in R**

```
 today_dt <- as.Date("2024-08-01")
queries_kri <- query_log %>%
  mutate(
    close_eff = ifelse(is.na(CLOSE_DT), today_dt, CLOSE_DT),
    close_eff = as.Date(close_eff, origin = "1970-01-01"),
    days_open = as.numeric(pmin(today_dt, close_eff) - OPEN_DT),
    long_open = days_open > 30
  ) %>%
```

```
  group_by(SITEID) %>%
  summarise(
    total_queries = n(),
    n_long_open   = sum(long_open),
    prop_long_open = n_long_open / total_queries,
    risk_flag = ifelse(prop_long_open > 0.20, 1, 0)
  ) %>%
  arrange(desc(prop_long_open))
print(queries_kri)
```

- **B.4 QTL - Missingness in R**

```
 hb <- subset(adlb, PARAM == "HbA1c")
miss_rate <- mean(is.na(hb$AVALW24))
qtl_threshold <- 0.05
if (miss_rate > qtl_threshold) {
  message(sprintf("QTL BREACH: Missing rate = %.1f%% > 5%%", miss_rate * 100))
} else {
  message(sprintf("QTL OK: Missing rate = %.1f%% <= 5%%", miss_rate * 100))
}
```

- **B.5 CSM - Mahalanobis Distance in R**

```
library(dplyr)
hb_subj <- adlb %>%
  filter(PARAM == "HbA1c") %>%
  mutate(CHG = AVALW24 - AVALBL) %>%
  select(USUBJID, SITEID, AVALBL, CHG)
X <- hb_subj %>%
  select(AVALBL, CHG) %>%
  as.matrix()
mu  <- colMeans(X, na.rm = TRUE)
S   <- cov(X, use = "pairwise.complete.obs")
invS <- solve(S)
md_site <- hb_subj %>%
  group_by(SITEID) %>%
  summarise(
    n         = n(),
    mean_base = mean(AVALBL, na.rm = TRUE),
    mean_chg  = mean(CHG, na.rm = TRUE)
  ) %>%
  rowwise() %>%
  mutate(
    md2 = {
      v <- c(mean_base, mean_chg) - mu
      as.numeric(t(v) %*% invS %*% v)
    }
  ) %>%
  ungroup() %>%
  mutate(
    p_value = 1 - pchisq(md2, df = 2),
    flag    = ifelse(p_value < 0.01 & n >= 5, 1, 0)
  ) %>%
  arrange(desc(md2))
print(md_site)
```