

OSS・Rの議論の結果

当日の議論結果

- ※本議論の内容は、参加者による意見交換をまとめたものであり、各参加者の所属団体の公式見解を示すものではありません

グループごとの一日の議 論のまとめ

グループ 1

① GCP業務とプログラミング

- 規制当局は人間が読めるプログラミングを求めている（人による理解が可能な形式）
- 誰が実行しても再現できることが求められる
- GUI操作のソフトはNG
- SAS, R → CSR
- 解析Winnolin → PK解析

グループ 1

② OSS

- 多くの場合、安価・フレキシブル・長期利用可能

	長所	短所
正確性	・ コミュニティ ・ メジャーなものは世界中で使用	・ ユーザーがCSV
再現性	・ コードにロジックを残せる	・ バージョン（後方互換性△）
追跡可能性	・ 関数へのロジックまで追跡可能	・ ログチェックの方法

グループ 1

③ CSV

- 要件定義 → 検証 → 記録 → 適切性を示す
- 解析環境の要件と検証
 - OS・パッケージ – バージョン管理
 - ログ – ロギングパッケージの出力
 - プログラムファイル – 文字化けしないで開ける
 - 結果 – SASとの一致

グループ 1

④ プロセスリスクとレバレッジ

- 意図どおりに動かない場合の影響
 - High・申請用解析 — Scripted
 - Not high・社内用解析 — Unscripted
- レバレッジ
 - 品質保証を代替する
 - 利用状況（ダウンロード数、本や論文）
 - QC（CAMIS等の成果物、githubの結果）
 - Pharmaverse

グループ 2

- 前提：人間が理解し解釈できるプログラム
 - 自分 ⇄ 他人 ⇄ 規制当局
 - このプログラム・解析で信頼性がある、再現性がとれる、説明責任を果たせる → 合意形成
- R分析環境のCSV
 - Rを取り巻くシステム全体の設計
 - 要件はユーザーの我々が中心に！

グループ 2

- RパッケージのCSV
 - CSAのアプローチで効率化
 - オープンソース ⇒ 集合知の活用
 - コミュニティで責任を分散？共有？ → レバレッジを有効活用
- Highリスク ⇒ レバレッジだけに頼るのは危険
 - 独自でも良いのでフレームワークを作りドキュメントを残す
 - ただし、誰がどんな基準でGOサインを出すの？の問題はある

グループ 3

< 今日、学んだこと >

- 問題意識は似ていた。→ CSVをどうする？
- CSV要件を決める難しさ→慣れは必要。

< 伝えたいこと >

- CSA、レバレッジ→ どこかでエイヤ！または、誰かがロジックを確認した以外のレバレッジを、どうやって説明？
- レバレッジをかけた穴に当局の指摘→ ほどほどにできないか
- みんなでお金を出しあってRのValidationってできないか
- SASって結局、ゼロにはできないっぽいのですかね？

グループ 4

- R：オープンソースカスタム可能、CSV + QC
- SAS：バリデーション済み、QC
- データ解析
 - 正確性、再現性、追跡可能性
- パッケージごとのCSV
 - リスクに応じて
 - 見た目の編集か、数値をいじるか
 - 承認申請や意思決定に影響するか
 - 解析処理が前か、後か
 - レバレッジ + 追加テスト

トピックごとの議論のま とめ

ディスカッション1

GCP業務に使われるソフトウェアで、ユーザーがプログラムを書いて操作するソフトウェアにはどのようなものがあるか、ディスカッションしてください。



ソフトウェア

20件

SAS	R
Python	JMP
NONMEM	WinNonlin
Spotfire	OpenBUGS
JAGS	Stan
Shiny	RAVE
Pinnacle 21	Excel
VBA	Tableau
PowerShell	CLI
EDCシステム	VTL
コマンドプロンプト	

ユースケース

用途カテゴリ別

データ処理・解析	- データ解析／中間解析／社内・社外解析 - 解析処理・バッチ処理 - PK解析／ベイズ統計解析／シミュレーション
臨床開発・統計	- CSR・CTD用TFL作成 - ADaM／SDTM作成、標準プログラミング - 臨床薬理解析、サンプルサイズ計算、症例統計 - モデル評価・検証、Dose設定・Dose simulation
可視化・レポート	- 可視化、Shinyによる可視化 - ダッシュボード／レポート作成
データ標準・提出	- define.xml作成、Reviewer's Guide作成 - ファイル一括操作
安全性・品質／自動化	- 安全性モニタリング、試験運営・管理 - 品質チェック／検証、自動化、Utilityツール・関数作成

ディスカッション2

臨床試験のデータ解析でRを使うとしたら、正確性・再現性・追跡可能性の視点でどのようなメリット・デメリットがあるかディスカッションしてください。

メリット

- 最新の統計手法・新しい解析手法を利用できる
- ソースコードレビューが可能で、処理ロジックを確認しやすい
- パッケージ／関数を検証可能（正解データは別途必要）
- 多数ユーザーによる確認・改善、アップデートが早い
- 自分の手で修正可能、公開コードにより検証しやすい
- バリデーション済み・広く利用されるパッケージを活用できる
- 公式ドキュメント・公開情報により内容確認が可能

デメリット

- パッケージの解析前提・妥当性が見えにくい場合がある
- パッケージ品質が統一されておらず、信頼性の再評価が必要
- 品質を保証する主体が明確でない／ベンダー保証が限定的
- 数式の誤り・バグ・論文とプログラム不一致の可能性
- ライブラリ更新・パッケージ更新時に再検証が必要
- バリデーションや正解データ準備に追加労力が必要
- 利用するライブラリの選定基準を決めにくい

示唆 「透明性」と「最新性」は強み。一方で、採用パッケージ選定・検証・更新時再評価のルール化が重要。

メリット

- renv等でR／パッケージ環境を固定し再現可能
- Git等でコード・バージョン管理／固定ができる
- 同じ環境・同じパッケージなら同じ結果が得られる
- バージョン差分が明確で、差異確認がしやすい
- 旧環境・旧バージョンの再現が比較的容易
- コード／パッケージ／バージョンを合わせれば共有・再実行可能
- 操作をコードとして残せるため、解析手順を再実行しやすい

デメリット

- 実行時の記録・ログ・バージョン情報の保存が必要
- R本体とパッケージ双方のバージョンを揃える必要がある
- バージョン違い・更新により結果が変わる可能性
- パッケージ更新頻度が高く、管理・統一が大変
- OS環境やオプション、依存関係の整合性が必要
- 過去バージョンの保持・サポート外化が保証されない場合がある
- 初期値や設定差異により結果が変わる可能性

示唆 再現性は「環境固定」と「実行記録」が前提。renv/Git/ログを標準運用に組み込む必要がある。

メリット

- logger等のパッケージでログを残し追跡可能
- コードとして残すことで処理過程を追跡できる
- 履歴管理・実行日時・ログをエビデンス化できる
- 関数や処理の中身まで追跡・確認できる場合がある
- パッケージのソースコードを閲覧できる
- SAS同様、コードを残せば再現・追跡に活用できる
- オープンであるためブラックボックス範囲を自分で確認しやすい

デメリット

- パッケージや実行環境に依存し、依存エラーが起こり得る
- Git・命名・ログ粒度など運用ルールに依存する
- 履歴管理・ログ管理・エビデンス整理にコストがかかる
- ログチェックが難しい／どのログを見るべきか分かりにくい
- マスクされた関数や暗黙の前提があり、説明不足になる場合がある
- バージョンずれの確認・管理が必要
- ソースコードの自由度が高く、標準化しないと追跡しづらい

示唆 追跡可能性は「ログを残せる」だけでは不十分。命名規則・ログ設計・証跡保存単位を事前定義することが鍵。

ディスカッション3

あなたはGxP対応可能なR分析環境の構築プロジェクトにプログラマとして参加しています。解析環境としてaccuracy (正確性)、reproducibility (再現性)、traceability (追跡可能性)の確保された解析環境を構築するために、解析環境の要件を提示し、必要な確認方法（と記録の残し方）を検討してください。

要件リスト

環境・構成管理

OS・実行環境のバージョン/環境を明確化
R本体・パッケージの種類/バージョン管理
環境固定（renv等）により同一環境を再現
社内サーバーファイルを開ける／解析関係者が同一環境へアクセス

再現性・同一結果

同一環境・同一入力で同一結果
誰が実行しても同じ出力・成果物を後から再現
乱数やシードの扱いを確認
SAS等の既知結果・根拠論文と照合可能

ログ・追跡可能性

実行ログの出力を統一しログを残す
変更を追跡できる／変更管理
ユーザー権限・アクセス証跡を残す
成果物を勝手に変更できない

パッケージ品質・利用可否

必要なpackageをインストールできる
利用するpackageの品質が担保されている
最新の統計手法を使える
バリデーション済みpackageを使える

運用・教育

運用ルール・手順書の整備
必要最小限の権限付与
トレーニングを受けさせる
システム担当が管理・記録を残す

正確性・成果物保全

Rが正しく動く
コードが正しく期待結果を出す
成果物・プログラム・データ・TLF・解析環境の保全
文字化けしない/エンコーディングを明確化

検証方法リスト

インストール・構成確認

インストール手順に従い実行テスト
インストールログを取得（MD5/チェックサム等）
インストールされたpackage/R/バージョンを記録
package一覧・バージョンを記録（renv等）

再現性テスト

同一入力・同一環境で同一結果を確認
乱数シードを確認
既知結果/SAS/手計算/根拠論文との照合
検証用データセットを作成してバリデート

ログ・監査証跡

ログ管理・ログ保存
実行ログ/エラーログが適切に残ることを確認
アクセス記録を確認
変更管理ログを保存

コード・レビュー

コードレビュー
コードが正しいことを確認
テストデータで正しさ確認
ダブルプログラミング/手順書を作成

権限・保管・証跡

必要最小限の権限を確認
Read Onlyスナップショットを取得
テキストファイルとして保存
他者作成ファイルが開けることを確認

リスクベース/運用

リスクに応じてテスト
パッケージごとにバリデーション要否を判断
質問/課題/設定方法/実施時期を保存
トレーニング実施を記録

要件と検証方法の対応表

CSV観点	要件	対応する検証方法
環境固定・構成管理	R/OS/packageのバージョン管理、環境固定、社内サーバー利用	renv等で環境固定、package/R/OSのバージョン記録、インストールログ取得
インストール成立性	必要packageが正しくインストールできる	手順に従ったインストール実行テスト、チェックサム/ログ確認
再現性	同一入力・同一環境で同一結果、誰が実行しても同じ出力	同一データ・同一コードで再実行、乱数シード確認、結果一致確認
正確性	Rが正しく動く、成果物全体を後から再現できる	既知結果・SAS・手計算・根拠論文との比較、検証用データセット
ログ/監査証跡	実行ログが残る、変更を追跡できる	ログ管理/保存、エラーログ確認、変更管理ログ保存
アクセス/権限	解析関係者が同一環境にアクセス、最小限権限	アクセス証跡、権限設定確認、他者作成ファイルの閲覧確認
成果物保全	成果物を勝手に変更できない	Read Onlyスナップショット、テキスト保存、Git導入や変更履歴管理
パッケージ品質	Package品質が担保、バリデーション済みPackageを使う	Package使用目的に応じたバリデーション、課題/設定/実施時期の記録
運用/教育	運用ルール、手順書、トレーニング	手順書整備、ダブルプログラミング、教育記録、システム担当の管理記録
文字コード/エンコード	文字化けしない、エンコーディングを明確にする	OS/エンコーディング設定の確認、テストデータで確認

ディスカッション4

RパッケージのリスクをHighとLowに分類して下さい。

High中心の評価：解析結果・CSR・データ変換への影響

High評価の根拠として、解析結果や申請・報告物への直接的影響が多く挙げられた

survival

High

- 解析結果に直接影響する
- Rで解析するなら使用する
- Impact High

admiral

High

- コードが楽になる

sdtm.oak

High / Low (評価割れ)

- 間違えがあると困る
- 導出が少ない

gtsummary

High

- CSRにのってくる

Low/評価割れの評価：出力・修正可能性・使い方依存

Low評価は「結果への本質的影響が小さい」「修正可能」「内部資料向け」などが主な理由

tidyverse

High / Low? (評価割れ)

- さけては通れない
- よく使われるベースのパッケージ群
- 結果に反映するパッケージを含む／他に影響を与える
- インパクトはありうるが、頻度は低い
- 使い方も要考慮 (Primary? / Secondary?)

xmpdf

Low

- 修正可能

labelled

Low

- 出力等のためLow
- たかだかラベル

flextable

Low / High / 要検討

- 社内資料向け
- 社内の意思決定

openxlsx

Low

- 解析結果に影響しない
- 見た目と本質に困らない

リスク評価から見える観点（今後の整理案）

評価軸1：解析結果への直接影響

survival / gtsummary / sdtm.oak では、解析結果・CSR・データ変換結果への影響がリスク理由として抽出されました。

評価軸2：出力後に修正可能か

xmpdf / openxlsx / labelled では、出力・見た目・ラベル等で修正可能または本質影響が小さいというLow理由が抽出されました。

評価軸3：利用範囲・用途依存

tidyverse / flextable は使い方、Primary/Secondary、社内資料/意思決定など用途によって評価が変わる可能性が示されました。

ディスカッション5

Rパッケージに対して、日常の業務でレバレッジできる品質保証活動があるか、ディスカッションして下さい

統合リスト：レバレッジとして考えられるもの

1. 普及度・使用実績

- ・ダウンロード数・利用ユーザー数
- ・更新頻度・バージョン履歴
- ・利用状況調査・使用実績
- ・製薬企業／アカデミアでの利用実績
- ・pharmaverse等のコミュニティでの採用・信頼

2. 申請・実務での採用実績

- ・申請使用済みパッケージ
- ・他社の治験・申請での使用
- ・QC業務（LP, SCR, BT等）での使用
- ・他品目等での使用実績
- ・社内ナレッジ・過去の記録

3. 文献・外部レビュー

- ・論文等での利用実績・引用数
- ・査読付き論文での使用
- ・学会・論文データ解析での使用事例
- ・書籍・論文に掲載されたパッケージ
- ・第三者／コミュニティによる確認

4. 結果比較・技術的妥当性

- ・SASとの結果比較／一致
- ・SASで作成した解析ツールとの比較
- ・R vs WinNonlin等の比較成果物
- ・GitHub等で公開されているテスト結果
- ・CAMIS等の比較資料・成果物

5. QA/QC・検証記録の活用

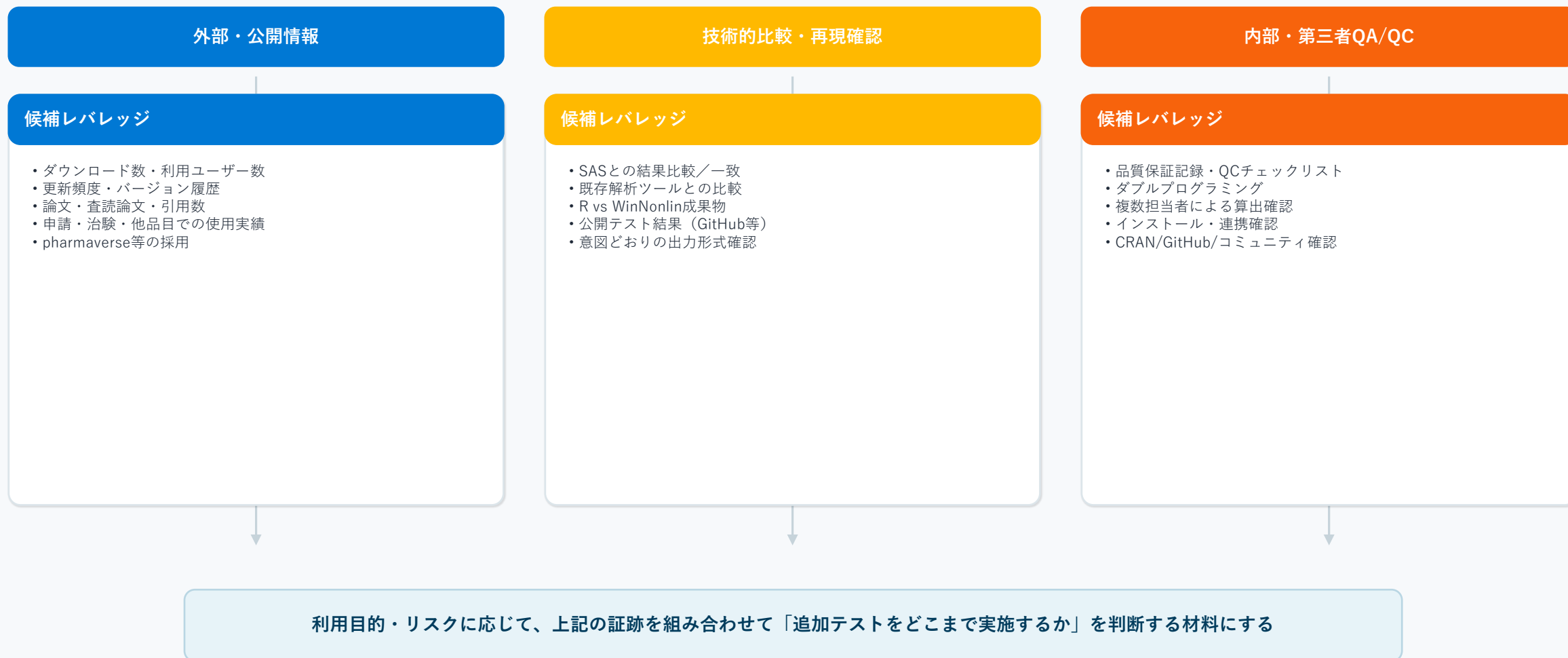
- ・他社が行った品質保証の記録
- ・QCチェックリストを用いたチェック
- ・コードダブルプログラミング／二重プログラミング
- ・複数担当者による算出・確認
- ・インストール確認・パッケージ動作チェック

6. 開発者・コミュニティ信頼性

- ・開発者の経験（開発パッケージ数等）
- ・信頼できるユーザー・大規模ユーザーの利用
- ・CRAN/GitHub/コミュニティでの確認
- ・他社・他パッケージとの相互確認
- ・ドキュメント正確性の担保

ポイント：単一の根拠だけに依存せず、利用実績・文献・結果比較・QA/QC記録・コミュニティ信頼性を組み合わせて「リスクに応じた保証」に使う、という方向性が読み取れます。

CSAレバレッジの位置づけ：証跡のタイプ別整理



ディスカッション6

Rパッケージに対して、リスク評価、実施するテスト、参考となるレバレッジの特定

抽出リスト：全体一覧

Package	実施するテスト	採用するレバレッジ
tidyverse	実データでQC／自覚症例・カルテ等での複数テスト	公開ドキュメントを参照
admiral	ダブルプログラミング（SAS→R／R→R）	ダブルプログラミング（SAS→R／R→R）:旧バージョンの実行結果等が参考になる
gtsummary	ダブルプログラミング／使用パッケージを限定してテスト／ダミーデータを使ってテスト	
survival	SASとの比較／正解データとの一致確認／文献の再現／コードレビュー／生存時間解析／コード確認・確認内容の記録	公開されている記録の確認／長年の利用実績／学術論文での利用／継続的な保守・バグ修理／Base R／ダウンロード数／CRANの確認
sdtm.oak	Controlled Terminology・Pinnacle 21チェック／過去試験データの再現／作成データのPinnacle 21 validation／必須変数のみ確認（またはコアドメイン）／Raw→SDTMトレーサビリティ確認／複雑な処理は確認／上手くマッピングされるか確認／Check Listで確認	Pharmaverseプロジェクトでの実績／Pharmaverse／外資の申請実績／CDISCコミュニティでの利用／ダブルプログラミング／実際のSDTMと同じものが作れるか確認（SDTMバージョン要確認）
flextable	QCチェック／目視チェック（誤字・脱字、体裁、文字列、値）／R内結果とWord出力の比較／期待した結果と同じか確認／Check Listで確認／出力データとの複数確認／Word・PowerPointテンプレート上のテスト	QCレビューで表形式の不具合なし／Officeのバージョンアップ後も問題なし／バージョン管理