R Adoption Story

Michael Rimler
Prog Leader, ClinProg

R Adoption in Clinical Reporting

Project Overview

Objectives

- Establish critical mass of R programming capability to deliver outputs supporting clinical research
- Create an environment where a newly onboarded R programmer can be as productive as a newly hired SAS programmer

WHY?

Role of **R vs SAS** in our future?

HOW?

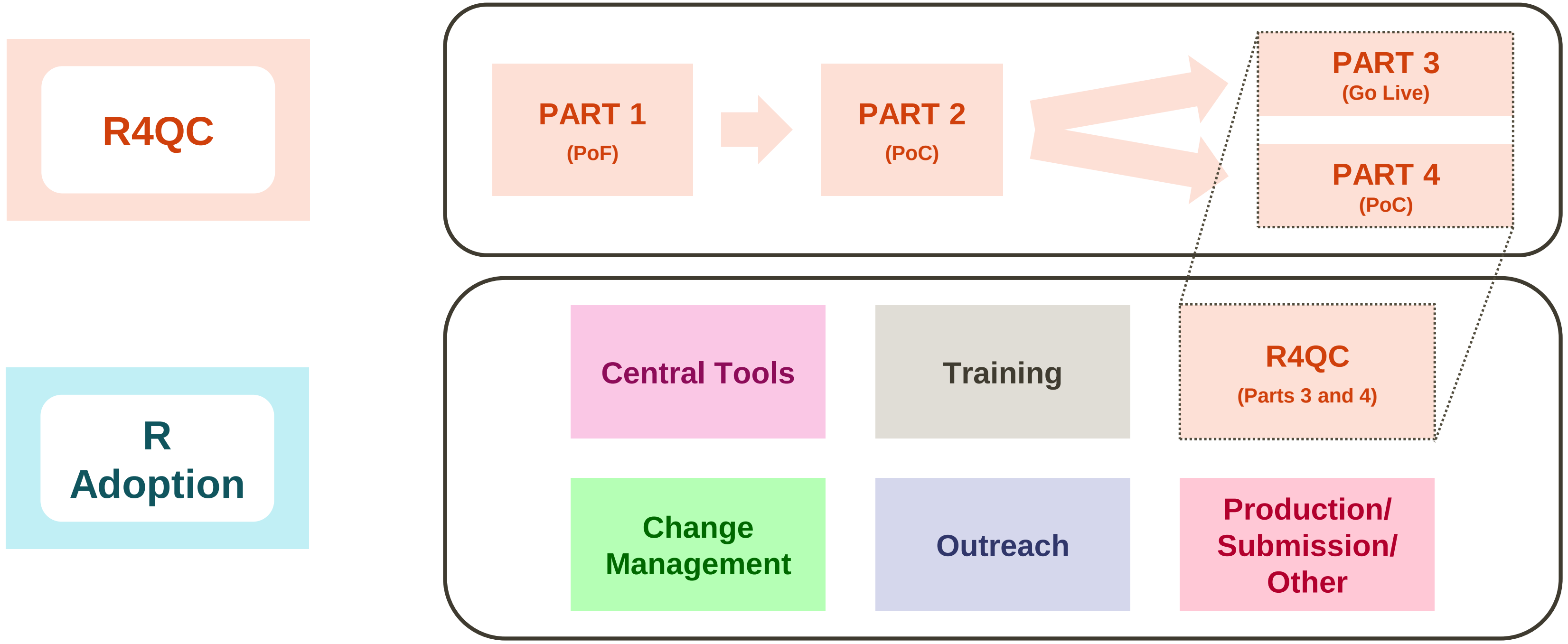
Elements needed to declare us '**R ready**'?

WHAT'S NEXT?

Future deliverables?
(…beyond DTFLs)

R Adoption Journey

Project Overview



The R4QC Project

R4QC Project – The Start

Rationale

Why R?

- Long term objective in Clinical Programming to become programmatically multilingual
  - Not driven by 'SAS license' costs
  - Recruitment
  - Retention
  - Flexibility / Agility (right tool)
  - Expansion of capabilities and clinical programming 'products' (~data science)

Why QC?

- Lower regulatory scrutiny
  - QC code typically not submitted
- Still produced using SAS
- Facilitates 'on-the-job' organic upskilling

R4QC Project – Parts 1, 2, and 4

PoC Summary

Objectives

Demonstrate the capability of using the R for Stage 2 QC of reporting deliverables

Facilitate ‘on-the-job’ use of R for to support staff upskilling

Successes

- Comparison of dataframes
- Cross-functional collaboration
- Central tool development
- R Roadmap Vision
- Support documentation

Challenges

- Onboarding into R ecosystem
- PROC COMPARE vs **`diffdf`**
- GPP in R (code review)
- SAS vs R reconciliation
- GitHub

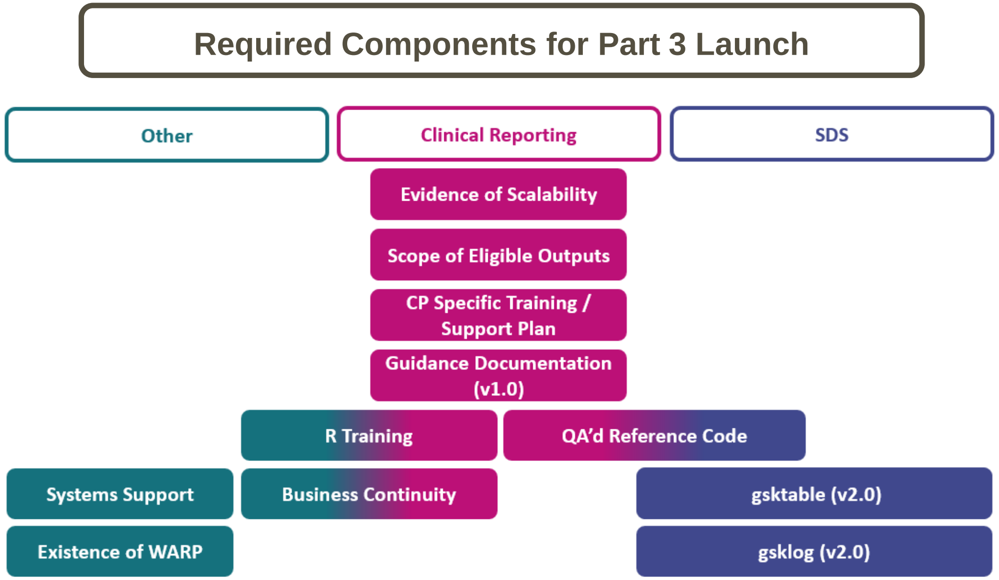
Note: R4QC Part 2 reproduced Stage 2 QC for 156 TFLs

R4QC Project – Part 3

High-Level Design

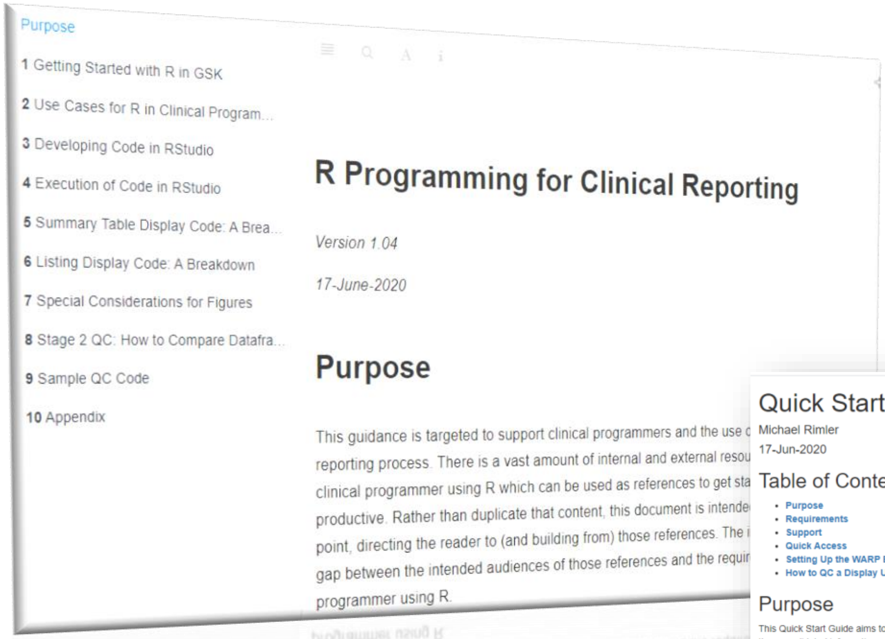
- Individual programmer opts-in to perform Stage 2 QC using R (instead of SAS)
- Scope: Any non-statistical display (TLF)
- Requirements:
  - Study Lead Programmer awareness/support
  - R4QC Part 3 output tracker
  - Ways of Working Instructions

- **Encouragement** from above (leadership)
- **Support** from below (staff/documentation)



R4QC Project – Part 3

Support Mechanisms



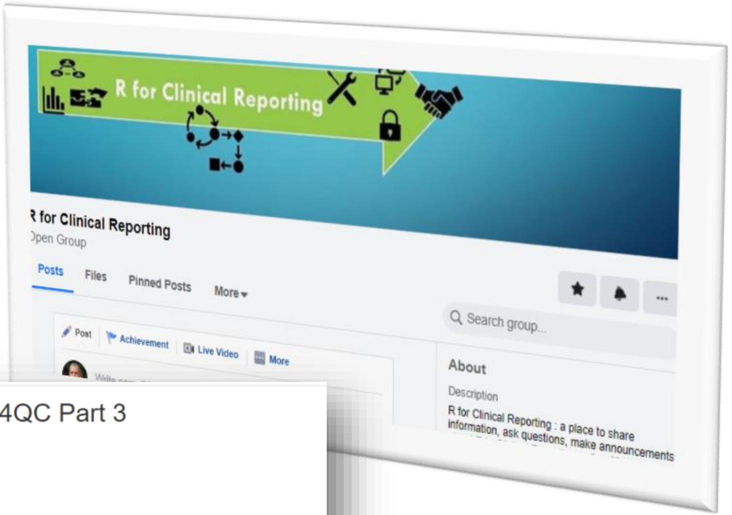
R Programming for Clinical Reporting

Version 1.04

17-June-2020

Purpose

This guidance is targeted to support clinical programmers and the use o
reporting process. There is a vast amount of internal and external resou
clinical programmer using R which can be used as references to get sta
productive. Rather than duplicate that content, this document is intende
point, directing the reader to (and building from) those references. The i
gap between the intended audiences of those references and the requir
programmer using R.



R for Clinical Reporting
Open Group

Posts · Files · Pinned Posts · More

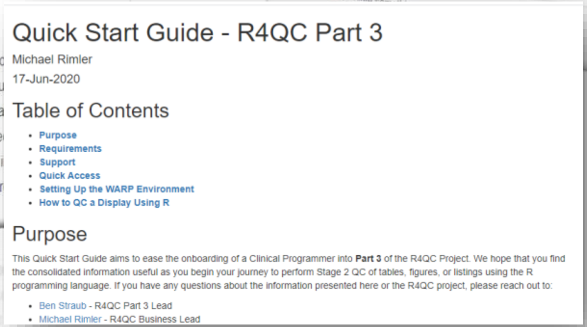
Post · Achievement · Live Video · More

Search group...

About

Description

R for Clinical Reporting : a place to share information, ask questions, make announcements



Quick Start Guide - R4QC Part 3

Michael Rimler

17-Jun-2020

Table of Contents

Purpose

This Quick Start Guide aims to ease the onboarding of a Clinical Programmer into **Part 3** of the R4QC Project. We hope that you find the consolidated information useful as you begin your journey to perform Stage 2 QC of tables, figures, or listings using the R programming language. If you have any questions about the information presented here or the R4QC project, please reach out to:

- Ben Straub - R4QC Part 3 Lead
- Michael Rimler - R4QC Business Lead

R Adoption Journey

R Adoption Framework

Motivations and Objectives

Motivations

- Expanded Toolset
  - Right tool for the job
- Expanded delivery capabilities
  - R Markdown, RStudio Connect, Shiny
- Upskilling
  - For delivery (the business)
  - For retention (the staff)
- Recruitment
  - Closing short run gaps / enabling long run growth
- Become an industry leader, not a follower

Deliverables

- "R-ready" study teams
  - For conventional delivery
- R Package development team
  - For process improvement
- Shiny development teams
  - For future delivery products
- Support Strategy

R for Submission

R Adoption Journey

Project Overview (revisited)

R Adoption

Central Tools

Training

R4QC
(Parts 3 and 4)

Change Management

Outreach

Production/ Submission/ Other

How do you manage change in a massive organization without disrupting delivery capability?

- Multifaceted approach
- Focused efforts
- Integrated and coordinated

Central Tools

Deep Dive

Central Tools

What's been developed?

Package development driven by Statistical Data Sciences

– `gskheader`: Insert standard GSK header for R programs
– `gsklog`: Generate execution/session log
– `gsktable`: Building blocks of basic summary displays (continuous and categorical stats; TLFs)
– `copy4qc`: Copy files from WARP environment to Study Area to facilitate archiving
– `gskdataset`: Building blocks of dataset transformations for clinical data

Central Tools

RStudio Add-in - `gsklog`

Purpose/Objective

- Generate a record of information related to the execution of an R script (in the direction of GxP compliance)

Rationale

- R does not naturally generate this information upon execution of code
- Provide minimum information to demonstrate the integrity of the execution and details to facilitate reproduction
- Specific to Clinical Reporting workflow (datasets, tables, listings, figures)

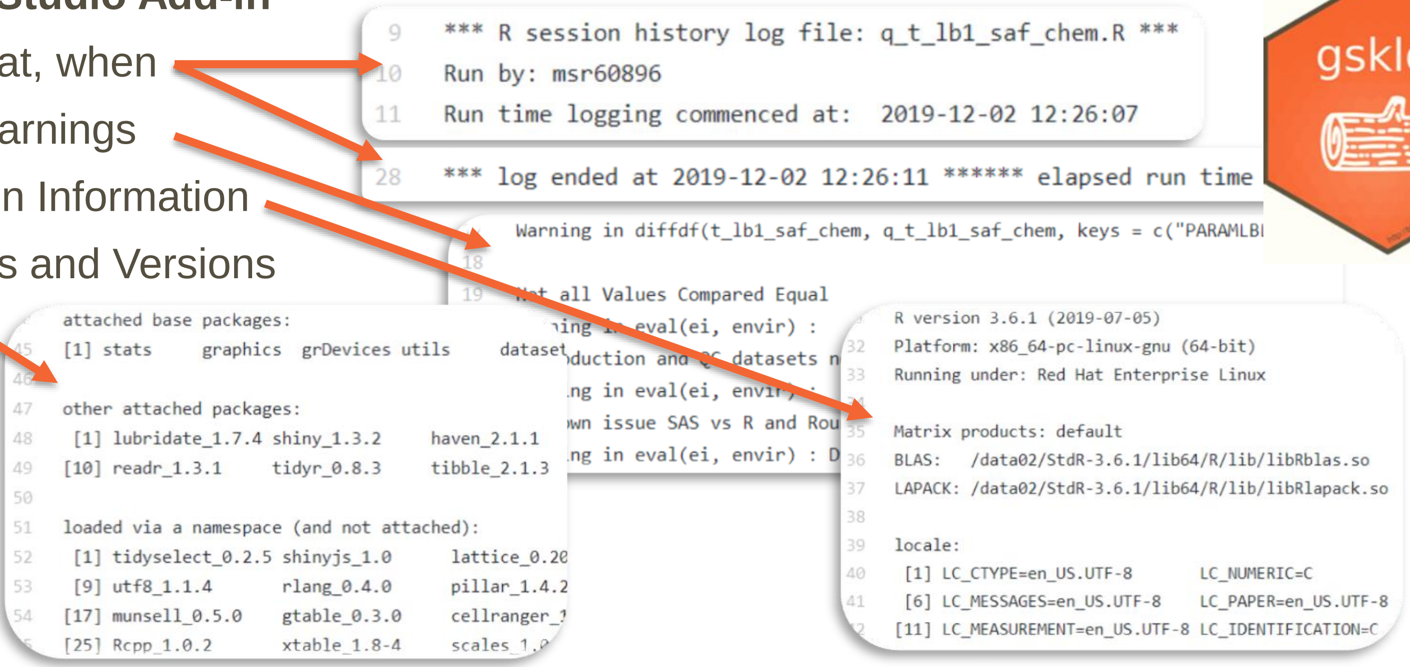
Not intended to replicate SAS log

Central Tools

RStudio Add-in - **`gsklog`**

Design – RStudio Add-in

- Who, what, when
- Errors/Warnings
- R Session Information
- Packages and Versions



```
*** R session history log file: q_t_lb1_saf_chem.R ***
Run by: msr60896
Run time logging commenced at:  2019-12-02 12:26:07
```

```
*** log ended at 2019-12-02 12:26:11 ****** elapsed run time
```

```
Warning in diffdf(t_lb1_saf_chem, q_t_lb1_saf_chem, keys = c("PARAMLBI

Not all Values Compared Equal
ning in eval(ei, envir) :
duction and QC datasets n
ng in eval(ei, envir) :
wn issue SAS vs R and Rou
ng in eval(ei, envir) : D
```

```
attached base packages:
[1] stats     graphics  grDevices utils     dataset

other attached packages:
 [1] lubridate_1.7.4 shiny_1.3.2     haven_2.1.1
[10] readr_1.3.1     tidyr_0.8.3     tibble_2.1.3

loaded via a namespace (and not attached):
 [1] tidyselect_0.2.5 shinyjs_1.0     lattice_0.20
 [9] utf8_1.1.4       rlang_0.4.0     pillar_1.4.2
[17] munsell_0.5.0    gtable_0.3.0    cellranger_1
[25] Rcpp_1.0.2       xtable_1.8-4    scales_1.0
```

```
R version 3.6.1 (2019-07-05)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Red Hat Enterprise Linux

Matrix products: default
BLAS:   /data02/StdR-3.6.1/lib64/R/lib/libRblas.so
LAPACK: /data02/StdR-3.6.1/lib64/R/lib/libRlapack.so

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [6] LC_MESSAGES=en_US.UTF-8    LC_PAPER=en_US.UTF-8
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
```

Discussion / Q&A

Andy Nicholls
Head of Statistical Data Sciences

Tilo Blenk
Data Scientist,
Statistical Data Sciences

Michael Rimler
Programming Leader,
Clinical Programming

Thank you!

Appendix

Additional Information

R4QC Project

Differences discovered (SAS vs R)

Default rounding

- SAS: By default, **round(12.5,1)** should result in **13**
- R: By default, **round(12.5,1)** should result in **12**
  - Base R implements 'round-to-even'

Derivation of Quartiles (Q1 and Q3)

- SAS: In PROC MEANS, can specify the parameters **q1** and **q3** and the parameters are calculated
- R: Using Base R **quantile** function, specify a 'type' parameter (9 different type values)
  - **1** = match SAS in R4QC; **3** = R documentation states will match SAS; **7** = default

Standard Errors in Survival Analyses

- Different methodologies are used calculate the SE in survival analyses
- When the survival function reports the SE, it prints an approximation of the SE calculated by the delta method
- But this SE has poor performance when used to calculate the 95% confidence intervals
- SAS vs. R use a different method to calculate the SE that is used to get the 95% confidence interval
  - **SAS:** The SE matches how one would calculate by hand; **R (the `broom` package):** The SE matches the underlying SE