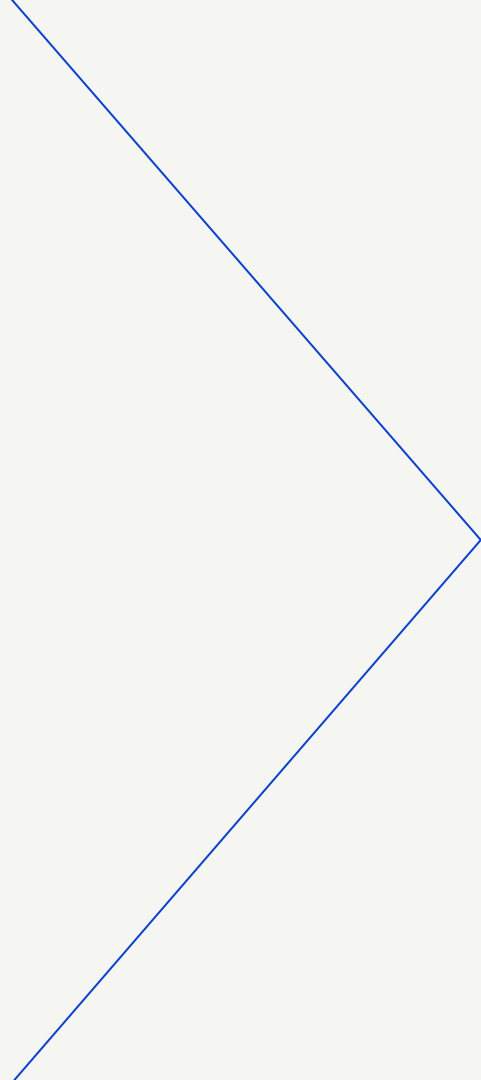


DV05 - R Shiny Apps: To QC or Not to QC?

PHUSE EU Connect 2022

Katie Withycombe (katie.withycombe@roche.com), Data Scientist

Table of contents

- 
- A large blue arrow graphic pointing from the left side of the slide towards the table of contents, formed by two diagonal lines meeting at a point on a vertical line.
1. Introduction
 2. What We Did
 3. Challenges/Learnings
 4. Discussion and Questions

Introduction

What is R Shiny?

- Shiny is an **R package** that allows anyone to build interactive web apps straight from R
- Shiny apps are an easy and flexible way to share multiple **interactive outputs** with stakeholders
- See more details at <https://shiny.rstudio.com/>

What was the use case?

- A range of **safety-related outputs** to be shared with an **IMC** (Internal Monitoring Committee) every 3 months in two (sister) masked Phase II studies in ophthalmology
 - The **frequency** of the IMC and the fact it was two studies simultaneously meant that producing and checking two lots of static outputs (and ADaMs) would be time consuming
 - They were ophthalmology studies, there was **not as much previous or standard output code** to use as in other therapeutic areas
 - An added complexity was that the unmasked data should only be shared with the IMC members for a short period of time around IMC
- The study team were keen to enable **easy review of the data** by the IMC, and interactive outputs (ie. in R Shiny apps) would enable the IMC to filter, eg. to a single patient, or a subset of patients of interest
- Programming team were **keen to use R**
- There was some code already available in R for the ophthalmology-specific tables

Roche guidelines for QC risk assessment of IDDs

IDD: Interactive Data Display

As a whole, **IDDs created using established templates** with minimal study specific changes can be viewed as **good quality** and should undergo Developer User Testing (DUT) before being shared with stakeholders. It is up to **programmer's discretion** to implement more thorough testing of the whole report/app if modifications to the template are considered major or if the changes are complex in nature.

Mitigating risks:

- Filters: Evaluation of the filters should be part of the IDD DUT. While it is not feasible to check all permutations, **programmers should test filters that impact key analysis.**
- End User Responsibilities: While stakeholders/end users are responsible for the interpretation of their self-service IDD results, they need to fully understand the IDD's data source, variables, nuances and limitations. It is best practice at minimum to do a **walk through/demo of the IDD** to ensure this understanding. Therefore, regular communication with stakeholders is an imperative part of our support of IDD.

In general, if individual visualizations are used for external communication or decision making, then those results should be considered as a Report Object. Risk assessment of the Report Object should be conducted as outlined in this guide and corresponding QC type performed.

What We Did

Code Development

- **Clear** (initial) **scope** as the IMC agreement gave a list of the outputs the IMC members were to review
- Then **created ADaM datasets** required, using as much standard code as possible (see appendix)
- Then **created the app** code in R
 - Used a **standard** (internal) group of **R packages** for making the R Shiny apps (and used as much standard code from these as possible)
 - Created only **one code base**, so the only difference between the apps for each study was the path to the data
 - Used git for easy code versioning
 - **Deployed masked versions** of the apps to ensure the IMC members were familiar with them
 - We also did a **walk through** of the apps with the IMC members when we shared them

Rationale of QC approach

Risk Assessment = What can go wrong? + How likely is it? And what would the downstream impact be?

What could go wrong:

Outputs could be **incorrect**

Impact:

IMC recommendation could be **wrong** as it was based on conclusions drawn from incorrect outputs

Methods of mitigation:

- Ensured **ADaM** datasets were of **high quality** (by double programming any that were not standard)
- Produced **key tables** in SAS (using standard code) to **cross-check** values
- **Cross-checked** across the outputs **within the app**
- Used as much standard or previous code for the outputs in the app as possible

Rationale of QC approach

Risk Assessment = What can go wrong? + How likely is it? And what would the downstream impact be?

What could go wrong:

IMC could **not know how use the app** correctly

For example:

- Don't understand how the filtering works and create an output that doesn't make sense
- Not know which outputs are in the app and how to get to them within the tabs

Impact:

IMC recommendation could be **wrong** as it was based on conclusions drawn from incorrect or inadequate outputs

Method of mitigation:

- Made **masked versions** of the apps available to the IMC members a few weeks before the IMC
- **Walked through** the (masked) apps with the IMC members before the IMC
- **Met with the IMC chair** a few times before the IMC to go through the app

Rationale of QC approach

Risk Assessment = What can go wrong? + How likely is it? And what would the downstream impact be?

What could go wrong:

Outputs used for the IMC **not reproducible**

Impact:

Reasoning behind the IMC recommendation not traceable

Method of mitigation:

- **Used git** for app code versioning, using the **tagging** functionality to store the version of the code used for each IMC
- **Kept final ADaM** datasets and any QC outputs for each IMC and tagged the version of the code used
- Versioning of SDTM and raw datasets done automatically

Our QC approach

- **ADaM** datasets (on which the apps ran) were either using **standard code or double programmed**
- **Cross-checks** were done across the apps to QC the outputs within the apps
- Key results were cross-checked using standard **SAS outputs**
- Ensured the reproducibility of the apps used for the IMC by keeping the final datasets (and any QC outputs) in our database
- Utilised the **tagging** functionality in git to store the commit ID of the app used for the IMC

Challenges/Learnings

Challenges

- **Change of IMC members** at the last minute (for the first IMC)
 - This led to different opinions on which outputs would be needed (last minute output additions), and also meant the IMC were less familiar with how to use the app
- **Lack of standard R code** to produce listings
 - Increased the risk of the listings and meant that cross checks had to be done on them (for the first IMC)
- **Lack of clear guidance** for QC of R Shiny apps being used in an IMC context
 - For IMC the guidance is limited (due to the variation in formality)
 - R Shiny apps don't fit into our standard document where QC methods are recorded

Learnings

- **Initial set up** was only **slightly less** time consuming than producing static outputs
 - Time saving was only really felt from for the second IMC onwards (see appendix)
- Having **high-quality ADaM datasets** greatly increased the confidence in the quality of the app outputs
- **Last minute output changes** are challenging no matter what approach you take!
- If I did this again:
 - I would have tried to better **understand the motivations** behind the initial list of outputs for the IMC, as it was almost all listings, which are tricky to review as there are more subjects in the studies
 - I would have **documented** the QC approach I used more thoroughly

Discussion and Questions

Discussion and Questions

What are your experiences?

- Have you used R Shiny apps or interactive data displays for an IMC or other similar use case?
- What was your QC approach? Did you have any challenges?
- What are your thoughts on QC, and could we do it more efficiently in the future?

Doing now what patients need next

Appendix

- Breakdown of ADaM datasets:
 - 4 double programmed (3 Ophthalmology test ADaMs, ADSUB)
 - 6 standard code (ADSL, AAG, ACG, ADAE, ADCM, ADVS)
- Breakdown of TLGs
 - 1 graph function for 3 graphs of ophthalmology data (created using some previous code)
 - 9 tables all using standard or previous code (**Disposition Summary**, 2 Observation Time, **AE Overview**, **Table of AEs in Study Eye**, **Table of AEs in Fellow Eye**, **AEs by Greatest Intensity**, Table of Slit Lamp Abnormalities, Table of Indirect Ophthalmoscopy Abnormalities)
 - In **bold** are the outputs which were cross-checked with standard SAS outputs
 - ~20 listings
- Timelines
 - ADaMs (and QC) were done over 2 weeks (~6 people did an average of 1.5 days each over the period)
 - To develop the app, about a day every week over 3 months
 - For the IMC preparation, 1 day to run the ADaMs, check QC okay and cross-check the app
 - Additional output requests are done on a case-by-case basis