

Get involved with the FDA/PhUSE Script Repository

Kevin Kane (PHASTAR), Hanming Tu (Accenture)

Introduction

At the March 2012 PhUSE/FDA Symposium, a project was launched to set up an open source script repository to hold and share programs that would be useful in medical research. Scott Getzin (Lilly) was the steering committee liason, and the co-leads for the project during the last year* have been:

Mary Nilsson	Eli Lilly
Sally Cassells	Next Step
Hanming Tu	Accenture
Kevin Kane	Phastar
Joy Li	FDA - CDER
Mat Soukup	FDA - CDER

*the project has been re-organised recently

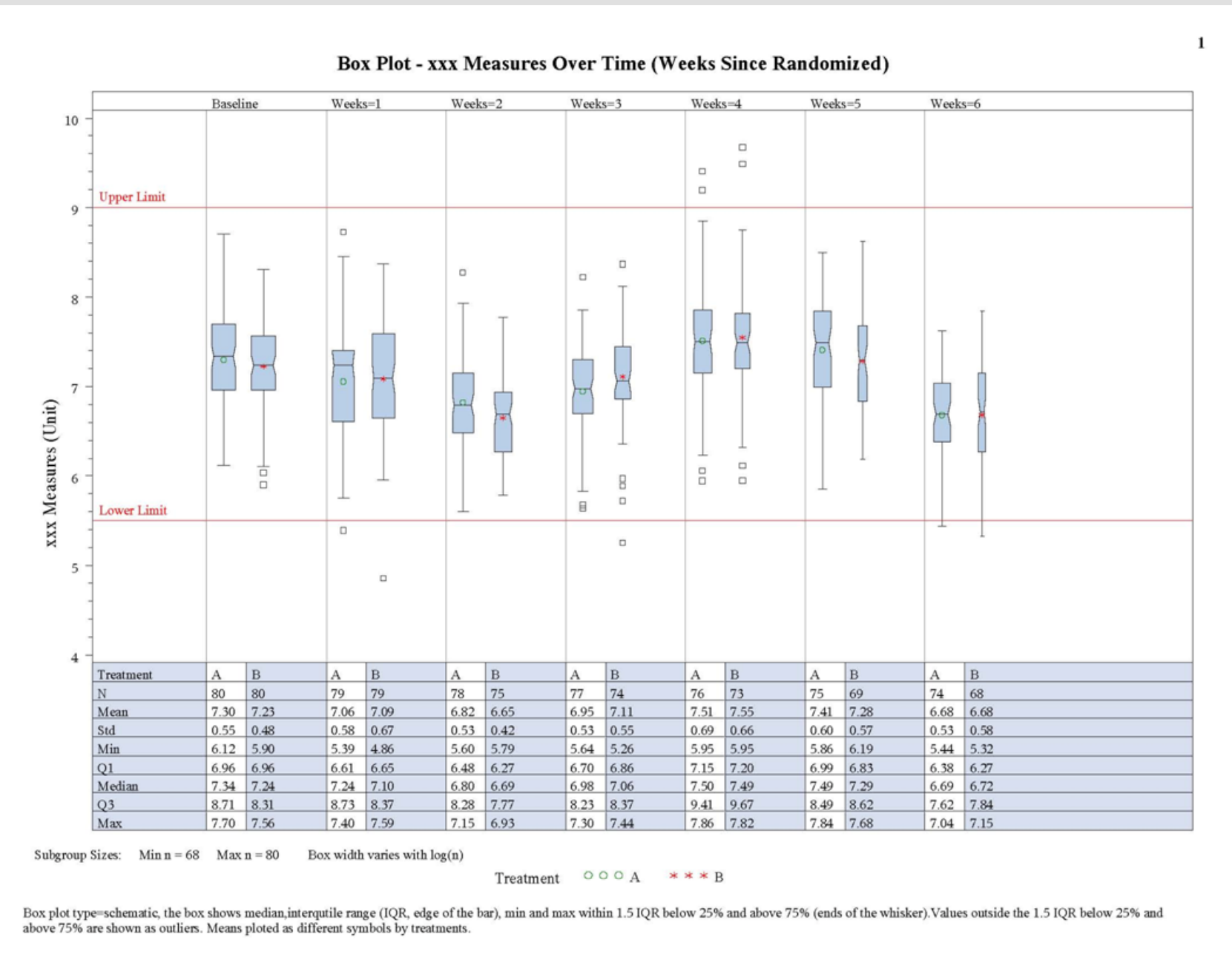
After the conference, we quickly developed the project into three main streams, which we called subgroups:

- Subgroup 1: Defining standards and desired scripts
- Subgroup 2: Processes for creating & validating scripts
- Subgroup 3: Choice of a platform for sharing scripts

White Papers

We are developing white papers to define standards for clinical trial reports. We will then seek to add scripts to the repository that will build on CDISC standardisation, creating datasets, tables, figures and listings.

The development of the first white paper, focussing on the summarization of safety data is advancing, and is available in it’s third draft (www.phusewiki.org). Here is an example of a standard that is being developed:



FDA/PhUSE Google Code Repository

Scripts will be solicited to match the standards developed in the white papers. Additionally, any program, tool or resource useful to medical research can be added by users. There are currently scripts relating to summary tables (e.g. demography, adverse events) as well as Kaplan Meier curves.

Scripts (R, SAS, STATA, S+, JAVA, C, Python, SQL etc)

Scripts can be added through a browser or a Subversion package to manage version control.

Google code provides an issue management system to flag bugs and problems with scripts.

Google code: what’s next

It is early days with the Google Code repository, and there are still quite a few issues outstanding:

- We need to define a folder structure and ensure that users can easily find useful scripts
- Google Code has two user roles: “Contributor” and “Committer”. The roles and responsibilities of these two types of users needs to be further defined
- Google code has an inbuilt system to flag and manage issues that users have with scripts. We need to clarify the process of flagging and resolving issues – who does what?
- In an idealised system, scripts store metadata alongside the actual script. Where and how to store metadata needs to be further defined

How to get involved

Here are 12 ways that you can get involved in this project:

- 1.Help to define new standards for outputs and desired scripts
- 2.Help to write and review white papers for desired scripts
- 3.If you have useful code that you have authored, donate it
- 4.Get familiar with our Google Code repository, check what’s in there
- 5.Check the repository for scripts that you could use
- 6.Use repository scripts, and feedback on any issues encountered
- 7.Ask around at your organisation if there are any scripts, code, programs, tools or resources that your organisation could share
- 8.We still need to define processes for storing validated scripts in the repository – sign up for to help this project
- 9.Get involved in clarifying open issues around copyright
- 10.Help to review draft templates for script documentation (user guides, validation documents)
- 11.Sign up to our listbox.com email list
- 12.Spread the news!