

Converting S-Plus Applications into R

Andy Nicholls

(anicholls@mango-solutions.com)



CONTEXT

Context

- Mango are speaking with an increasing number of customers who would like to convert their existing S+ application into an R equivalent

Why Convert an S+ Application to R?

We have seen three primary drivers for change:

- Long term cost reduction (S+ licenses)
- Incompatibility of S+ version with new OS
- Withdrawal of Support
- Wider uptake of R in industry

Clearly R is a natural choice as a replacement for S+

Aim

- To provide a high level overview of the considerations for converting an S+ application to R

Agenda

- Context
- Considerations
- Menu System
- Code Conversion
- Validation and Testing

CONSIDERATIONS

This is Easy, Right?

Some (true?) statements:

- R can be considered as a different implementation of S
- There are some important differences, but much code written for S runs unaltered under R

Discuss...

Source: www.r-project.org

Considerations

S+ applications can generally be split into two pieces:

- An underlying library of code
- A set of functions defining the menu system and help pages

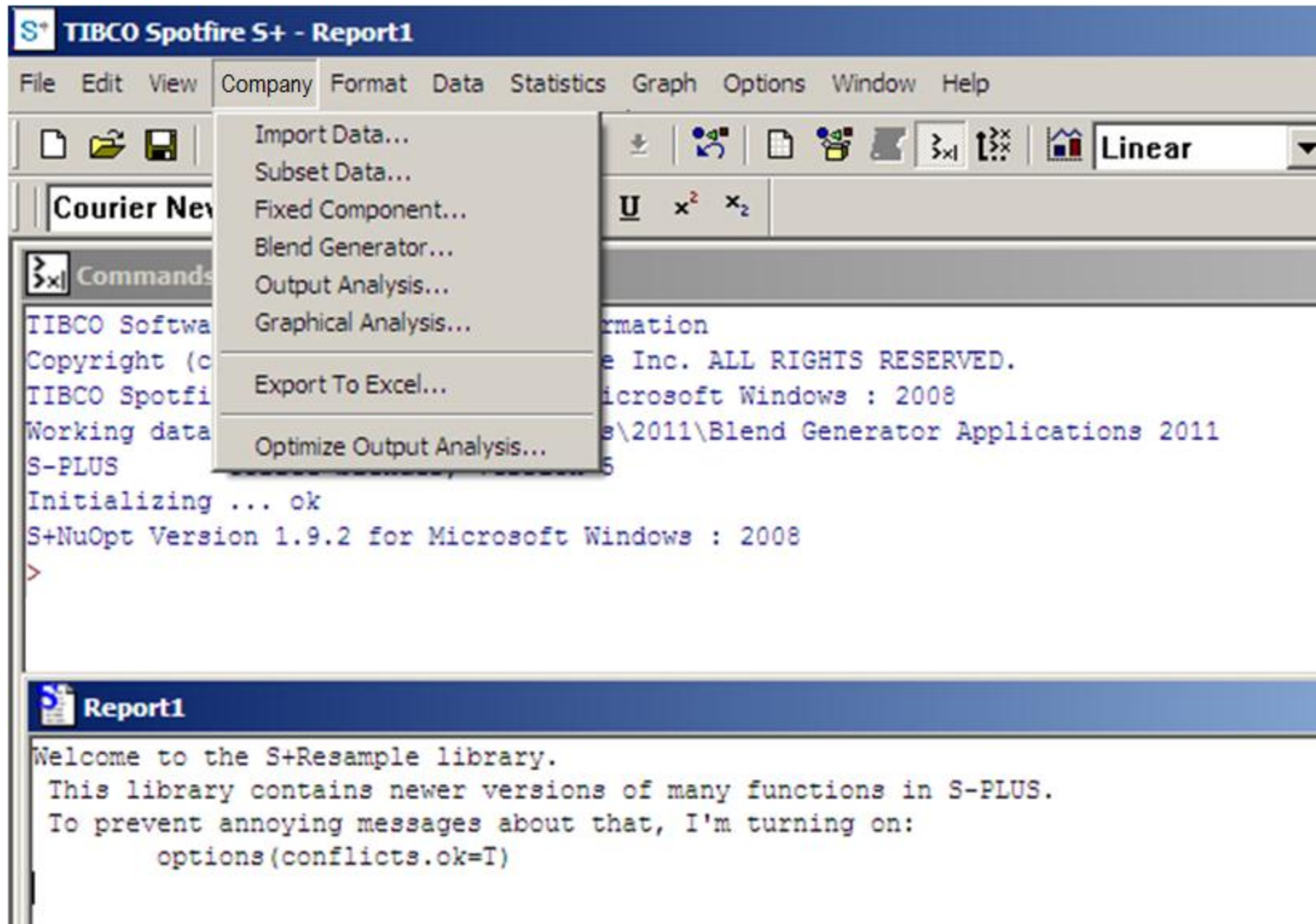
Case Study

Multinational confectionary, food and drinks company

- Existing S+ application consisting of an integrated menu and some analysis code
- S+ libraries consisting of 7,000+ lines of S code spread over 80 functions
- After removing those functions relating to the menu and help files, only around 3,000 lines of code remained

MENU SYSTEM

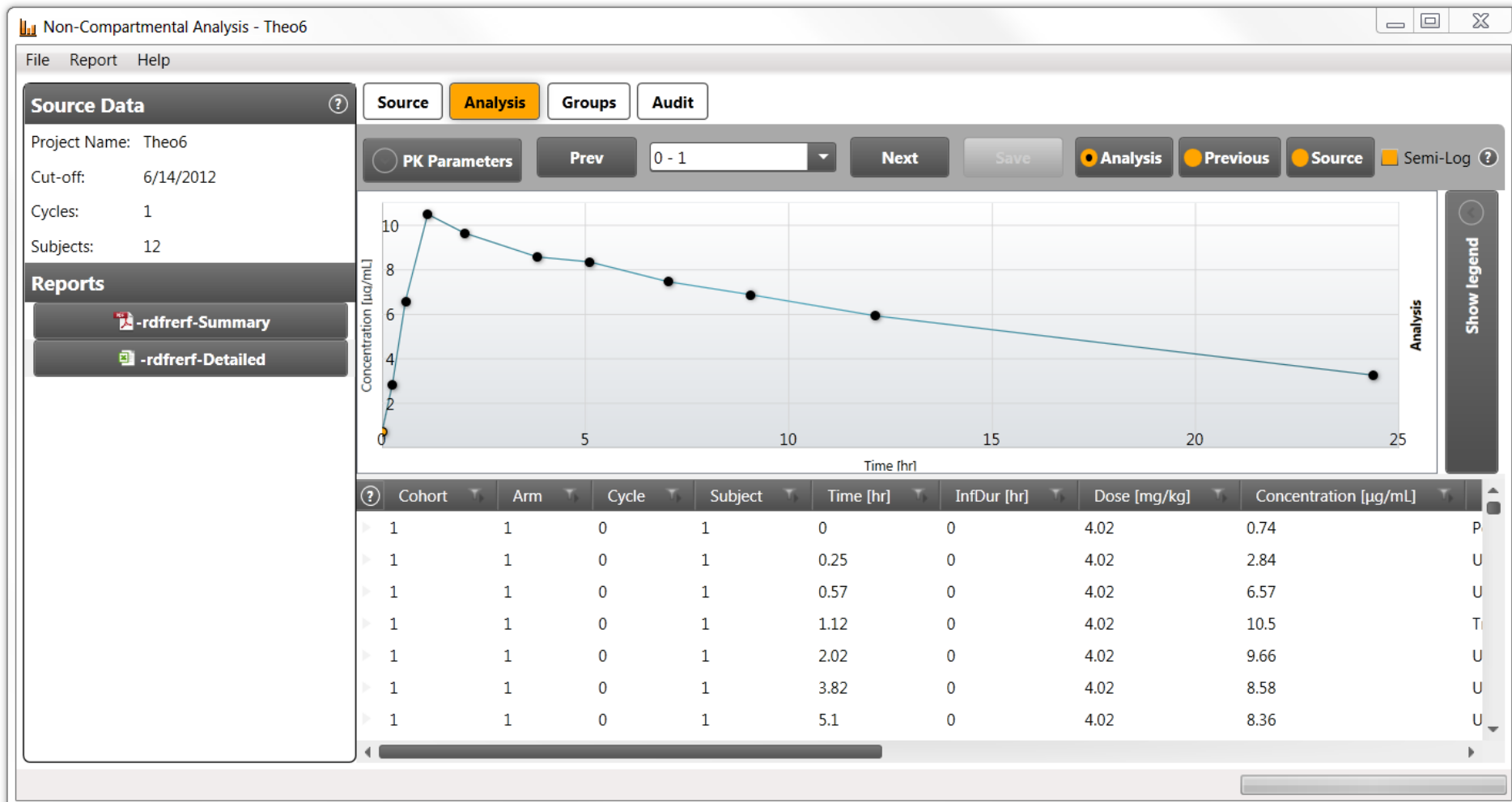
S+ Menu System



Menu System Opportunities

- S+ Applications tend to be integrated into the S+ GUI
- R systems can be also be integrated but require new code
- The move to R represents an opportunity to hide the R GUI

An Opportunity to Hide R



An Opportunity to Hide R

- Hiding R can improve user acceptance amongst non-technical colleagues
- It can also improve the visual appeal of the system

Menu Choices

Some R/R-based technologies:

- `tccltk` is R's 'recommended' menu builder
- **Glade**, `RGtk2`
- `gWidgets`
- `rpanel`
- `Deducer`
- `Shiny` (**Rstudio**)
- ...

Menu Choices

Other options:

- Choice is almost limitless
- Often they require a knowledge of other languages such as Java, C, JavaScript, .NET, ...
- Possibly warrants a standalone talk...

CODE CONVERSION

Approach

There are essentially two approaches to code conversion:

- Direct Conversion
- Test-based Conversion

Direct Conversion

- Requires knowledge of both languages (stdev vs sd)
- Relatively quick to achieve
- Difficult to prove the new code does what the old code did

Test-based Conversion

- Generating unit tests in S+ requires some S+ knowledge
- Takes some time to generate and document tests but better in the long-run
- Unit tests give a definitive PASS/FAIL result
- Can often be automated

CHALLENGES

Challenges

As the business owner I want to use the opportunity to improve the application:

- New menu items
- New functionality
- Modifications to existing functionality

All of these require careful planning

Technical Challenges

- OS version and R version need to be determined in advance
- It is almost guaranteed that the new system will produce different results for the same test data!
- What is the users want the latest version of R?

VALIDATION AND TESTING

Validation and Testing

- This is not exciting!
- When we use an application, we assume it has been “tested” and hope/require that it has been “validated”
- But what does this really mean?

Validation

What is validation?

- Design Qualification
- Installation Qualification
- Operational Qualification
- Performance Qualification
- A lengthy documentation exercise

Testing

- Unit tests check individual pieces of code
- A system is made up of many pieces
- We need to ensure it has been pieced together correctly!

System Testing

Requires:

- Patience
- An independent tester/test team
- Planning
- More documentation!!!

System Testing

Recommended:

- An automated build system
 - Hudson, Jenkins
- Bug reporting system

Summary

- Converting an S+ application to R requires careful planning
- It presents an opportunity to make improvements
- Validation and independent testing are key to a successful migration

Questions