PSCF/Linux - running an analysis from a Linux command line

The main PSCF tool is Windows-based, and is integrated with Excel. Run in this way, the system provides the inputs for the analysis program, causes the analysis program to be generated, compiled and run, and enables the analysis results to be displayed back in Excel. All these steps are automated.

For Linux, where Excel is not available and the VB-coded macros it employs cannot easily be replicated in other spreadsheet systems (e.g LibreOffice), analysis can still be achieved with just a few extra, manually-controlled, steps.

Overview of operations

Analysis is a three-stage process:

1. *Create text files in tab-delimited format. One way to do this is to export the analysis input files from a PSCF workbook*

The analysis input files are:

model.tab

series.tab

(adjustments.tab) - if the adjustments facility is to be used

(ratingmap.tab) - if the ratingmap facility is to be used

(These names correspond to the names of the tabs in a standard PSCF configuration.)

1. *Use a few simple shell scripts to generate the C++ analysis program, compile and link it and then run it*

The commands are:

gen - to generate the analysis program “prog.c”

comp - to compile “prog.c” into “prog.o”

lnk - to link “prog.o” to create the executable program “prog”

prog - to run the analysis program and generate output files

clean - to tidy the directory by removing all generated output files

(These are very simple shell scripts, and can easily be combined or modified as necessary or convenient. They have been provided in this form to make their function as explicit as possible.)

The detail of running these scripts (working directories &c) is covered further down.

1. *Import the tab-delimited analysis output files into a spreadsheet or database for viewing*

The analysis output files are generated as:

results.tab

(projection.tab) - if full details of the projection runs are required

Building the Linux environment

Before the first analysis run, it is necessary to create the “genprog” Linux executable from its source. This is a once-only process.

The detail of this build operation (working directories &c) is covered further down.

The PSCF/Linux area and its files

For its scripts to be used unmodified, the Linux area needs to be located as PSCF/Linux (alongside PSCF/system &c) in order for components of the main system to be accessed. (The Linux area is, however, moveable and re-configurable by appropriate changes to its scripts.)

Within PSCF/Linux are the following directories:

bin/ - holding the “genprog” executable (once created)

scripts/ - holding “build” - a script for creating the “genprog” executable

work/ - where the input “.tab” files should be placed, and where the analysis output files will be written

(Note: execute permissions on the shell scripts may need to be re-established.)

Building

Simply go into  “scripts/”, and run the “build” command. This will compile and link the “genprog” executable. (Presence of GNU g++ is assumed.)

Running an analysis

In the “work/” directory:

1. *Import the appropriate “.tab” files (“model.tab” and “series.tab” and the optional “adjustments.tab” and “ratingmap.tab”)*
2. *Run the “gen” script to create the analysis program “prog.c”*  
   As supplied, the “work/” directory contains a consistent set of input files and processing scripts. The “PSCF/Linux” facility, being derived from the more integrated Windows system, is not very tolerant of the inconsistencies that can arise from manual intervention. In the base Windows system, not only is integrity automatically ensured, but there is no need for user-friendly command interfaces. The impact of this on “PSCF/Linux” is that it quite likely will not behave gracefully in the presence of data or configuration errors. Also, the absence of the syntax-checking and error-reporting facilities of the full system means that workbook errors may not be as easy to trace.  
     
   As the above indicates, the arguments to the various scripts must be correct. The “gen” script is the most complex example. Inspection will show that it takes five arguments:

\* the absolute location of the standard program template

\* the absolute location of the program template for default probability code handling

\* “A” if the adjustments data is present and is to be used - or ”XA” otherwise

\* “R” if the ratingmap data is present and is to be used - or ”XR” otherwise

\* “DP” if “model.tab” contains default probability code that is to be used - or ”XDP” otherwise

All the arguments must be present, valid, and in this given order.

1. *Run the “comp” script to compile “prog.c”, creating “prog.o”*  
   If there are any compilation errors, these need to be resolved before continuing. Errors should only arise if the contents of the workbook cells do not conform to their expected formats.
2. *Run the “lnk” script to link “prog.o”, creating “prog” - the analysis program*
3. *Run the “prog” analysis program*

Run it with a single “Y” argument if the projection details are to be written to “projection.tab” in addition to the main summary results being written to “results.tab”. Otherwise, omit the argument.

Running an analysis should produce this type of outcome...

$ **cd Linux**

$ **ls**

bin/ scripts/ work/

$ **cd scripts**

$ **ls**

build

$ **build**

compiling utils...

compiling genrog...

linking genrog...

done

$ **cd ../work**

$ **ls**

adjustments.tab  comp\*  lnk\*       ratingmap.tab

clean\*           gen\*   model.tab  series.tab

$ **gen**

genprojtab version 1.0

  reading spreadsheet data...

  generating analysis program...

analysis program 'prog.c' created

ok

$ **ls**

adjustments.tab  comp\*  lnk\*       prog.c         series.tab

clean\*           gen\*   model.tab  ratingmap.tab

$ **comp**

*(this may take several seconds)*

$ **ls**

adjustments.tab  comp\*  lnk\*       prog.c  ratingmap.tab

clean\*           gen\*   model.tab  prog.o  series.tab

$ **lnk**

$ **ls**

adjustments.tab  comp\*  lnk\*       prog\*   ratingmap.tab

clean\*           gen\*   model.tab  prog.c  series.tab

$ **prog**

generation of analysis results starting...

generation finished: file 'results.tab' created

$ **ls**

adjustments.tab  comp\*  lnk\*       prog\*   projection.tab  results.tab

clean\*           gen\*   model.tab  prog.c  ratingmap.tab   series.tab

*(“projection.tab” will be not be populated unless “prog” is run with the “Y” option)*