Randtest user guide

2013-09-18 PsN 3.6.9

Introduction

Randtest is a tool for randomization tests.

Examples

randtest run1.mod -samples=1000 -randomization_col=DGRP -base_model=run0.mod

Input and options

randtest-specific input

A model file is required on the command-line.

-samples=N	The number of randomized datasets to generate, required.
-randomization_column=col	The name of the column to randomize, required. The column name is taken from the model file's $INPUT record. Column names in the data file are ignored.
-stratify_on=<column>	Optional, default not used. Restrict randomization of the data to within stratification groups, so that individuals in one stratification group cannot be assigned randomization column values from another stratification group.
-base_model=filename	Optional. Name of model to estimate with original data and use as reference when computing delta-ofv.
-match_transitions	Not used by default. Method for copying randomization column values from one individual to another during shuffling. See details in section Shuffling the randomization column.
-copy_data	Default not set. By default, the randomized datasets rand_<sample_no>.dta generated in the m1 subdirectory are not copied to the NM_run subdirectories of modelfit_dir1. Relative data paths, ../../m1/rand_<sample_no>.dta , are used in $DATA in the modelfiles in NM_run. This saves much disk space. If -copy_data is set, the randomized datasets are instead copied to the NM_run directories, and $DATA has rand_<sample_no>.dta without any path.

Some common PsN-options useful with randtest

For a complete list of common options see common_options_defaults_versions.pdf, or psn_options -h on the commandline.

-directory=randtest_dirN	The directory in which the script will run NONMEM can be named. The default name is “randtest_dirN” where N is increased by 1 each time you run the script. If the run is aborted or crashes, setting the directory to the one from which the script was running

	earlier can be done. PsN will then not run the model files that had finished, saving time.
-seed=N	A seed for the random number generator can be specified. This makes the run reproducible. Important note: the results of two runs will be different even if the seed is the same if the lst-file of the input model and the base model are present at the start of one run but not the other. Running the input model or base model changes the state of the random number generator, and therefore the randomized datasets will be different depending on if the input and base models are run or not before generating the new datasets.
-clean=N	The user may choose to remove different sets of intermediate file by setting clean to 3, 2, 1, or 0. The higher the number the more files are removed.
-threads=N	The number of parallel processes to start on a parallel computer.
-help	With -help randtest will print a longer help message.

Shuffling the randomization column

The only column that will change during shuffling is the randomization column. All other columns will be intact, and the order and numbering of individuals will be preserved. Shuffling can only be done on the level of individuals. If stratification is requested via option -stratify_on, then shuffling will be done separately within the groups defined by unique values in the stratification column. PsN will only look at the value of the stratification column for the first record of each individual. If other records for the same individual have a different stratification value, PsN will print a warning and then ignore the new value.

During shuffling the sequence of randomization column values from the records of one individual will be copied to another individual. PsN will handle copying also when the number of data set records differs between individuals.

If -match_transitions is not set, the default

By default, when option -match_transitions is not set, PsN will copy values record by record. If the sequence of 'copy-from' values is shorter than 'copy-to' then PsN will do 'last-observation-carry-forward'. Example: Copying [0,1,1] to [0,0,2,2] will give result [0,1,1,1] when -match_transitions is not set. Missing data marked by e.g. -99 or values that are '.' (a dot) will be handled like any other value.

If match_transitions is set

If option -match_transitions is set, PsN will match transitions instead of records in the sequence of values. The 'copy-from' sequence determines the values after each transition, while the 'copy-to' sequence determines where the transitions should be. Example: Copying [0,1,1] to [0,0,2,2] will give result [0,0,1,1], i.e. the resulting sequence has 0 before the first transition and 1 after the first transition. If the 'copy-from' sequence has fewer transitions than 'copy-to' then PsN will do 'last-observation-carry-forward'. Example: Copying [0,1,1,1] to [0,0,2,3] will give result [0,0,1,1]. If the dataset has missing data marked with something numeric, e.g. -99, then going from an observation

to -99 will not be counted as a transition, and -99 will not be copied. Example: Copying [1,-99,1,0] to [0,0,2,2] with -match_transitions will give result [1,1,0,0]. Copying [1,1,1,0] to [0,-99,2,2] will give result [1,1,0,0]. Copying from [-99,1,1,0] to [0,0,2,2] with will give result [1,1,0,0]. Non-numeric values, e.g. . (a dot) will be lumped together and treated as any non-missing value, so going from an observation to a dot will be considered a transition. Example: Copying [0,0,1,1] to [.,0,0,0] will give result [0,1,1,1]

Output

The raw_results file is almost a standard PsN file containing raw result data for termination status, parameter estimates, uncertainty estimates etc. for all model estimations. The first row is for the base model, if any, estimated with the original dataset. The next row is the input model with the original dataset. The following are the input model with the randomized datasets. If a base model was given an extra column with header deltaofv is inserted next to the ofv-column. This column contains the ofv-value of the current model minus ofv for the base model.

Known bugs and problems

If a csv-format dataset has an empty first column, i.e. there is a leading comma on each line in the datafile, randtest will crash. Avoid the problem by editing the datafile, removing the leading empty column.

It is recommented to remove all $TABLE from the modelfile, otherwise there will be much extra output produced. For the same reason it is recommended to remove PRINT options from $ESTIMATION.

The results of two runs will be different even if the seed is the same if the lst-file of the input model and base modle are present at the start of one run but not the other. Running the input model changes the state of the random number generator, and therefore the datasets will be different depending on if the input model is run or not before generating the new datasets.

Technical overview of algorithm

PsN will rerun the base model if the lst-file of the input model (run1.lst if the input model file is called run1.mod) is not present in the same directory as the model file OR if the model file has a different extension than .mod, for example .ctl. If the model file has the extension .mod and the lst-file is present then PsN will simply read the estimates from that file. Similar for the base model.

The program creates N (N=number of samples), datasets of size M (M=sample_size) by shuffling the values in the randomization column between individuals. The program creates N new NONMEM modelfiles which are identical to the original modelfile with the exception that each uses a different randomized dataset and that the initial parameter estimates are the final estimates from the original run. The model parameters are estimated with each dataset.