

# Bootstrap user guide

PsN 4.6.0

Revised 2016-04-13

## 1 Introduction

Bootstrap is a tool for calculating bias, standard errors and confidence intervals of parameter estimates. It does so by generating a set of new datasets by sampling individuals with replacement from the original dataset, and fitting the model to each new dataset [1]. See also [2] for diagnostics using option `-dofv`.

Example:

```
bootstrap run12.mod -samples=500 -seed=12345 -threads=5
```

## 2 Input and options

### 2.1 Required input

A model file is required on the command-line.

### 2.2 Bootstrap-specific options

`-samples =  $N$`

The number of bootstrapped datasets to generate. Default is 200.

`-sample_size =  $M$`

The number of subjects in each bootstrap data set. The default

value is set to the number of individuals in the original data set. When the resampling is stratified, the `sample_size` option can be used to specify the exact number of samples that should be drawn from each strata. Below follows an example of the syntax that should be used in such a case. Stratification is here done based on the study number, `STUD`, with the values 1001, 1002 and 1003. `-sample_size="1001=>12,1002=>24,1003=>10"` Note the double quotes, single quotes will not work on Windows. This example specifies that the bootstrap should use 12 samples from study 1001, 24 samples from 1002 and 10 from study 1003. If only one sample size is used together with stratified resampling (the default case; `sample_size=number of individuals in the data set`), the strata are assigned samples in proportion to their size in the data set. Please note that this usage of the `sample_size` option does not guarantee that the sum of the samples of the strata is equal to the given `sample_size` since PsN needs to round the figures to the closest integer. For a sample size equal to the number of individuals in the data set, the sum will however always be correct.

**-stratify\_on** = *column name*

It may be necessary to use stratification in the resampling procedure. For example, if the original data consists of two groups of patients - say 10 patients with full pharmacokinetic profiles and 90 patients with sparse steady state concentration measurements - it may be wise to restrict the resampling procedure to resample within the two groups, producing bootstrap data sets that all contain 10 rich + 90 sparse data patients but with different compositions. The default is not to use stratification. Set `-stratify_on` to the column (the name in `$INPUT`) that defines the two groups. Note that the option `sample_size` has a different behaviour when stratified resampling is used.

Bootstrapping is always done on entire individuals, so for each ID the data records are either all included or all excluded from a particular bootstrapped data set. The algorithm requires that an individual can unambiguously be categorized according to the stratification variable. In data file terms, it means that the variable used for stratification must have one and only

one value for each individual, otherwise the program will stop with an error message saying that at least one individual has multiple values in the stratification column, and therefore this column cannot be used for stratification of the resampling.

**-run\_base\_model**

By default, the input model will be run unless the lst-file for the input model is found in the same directory as the input model. If the user sets `-no-run_base_model`, the input model will not be run even if the lst-file is missing.

**-update\_inits**

By default, the initial estimates of the bootstrap models will be set to the final estimates of the input model if final estimates are available. If the user sets `-no-update_inits`, the initial estimates of the bootstrap models will be the same as the initial estimates set in the input model, even if final estimates from the input model are available.

**-keep\_tables**

By default, all \$TABLE will be deleted from the bootstrap models, to save disk space. If option `-keep_tables` is set, PsN will instead keep \$TABLE, which can require much disk space.

**-bca**

Default not used. When used, the bootstrap utility will calculate the confidence intervals through the BCa method [1]. The BCa is intended for calculation of second-order correct confidence intervals. Warning: Using bca is very time-consuming.

**-skip\_minimization\_terminated**

Used by default. When used, the bootstrap will skip all samples where the NONMEM run terminated the minimization step. Disable with `-no-skip_minimization_terminated`.

**-skip\_covariance\_step\_terminated**

Default not used. When true/used, the bootstrap will skip all samples where the NONMEM run terminated the covariance step.

**-skip\_with\_covstep\_warnings**

Default not used. When used the bootstrap will skip all samples where the NONMEM run had warnings from the covariance step.

**-skip\_estimate\_near\_boundary**

Used by default. When used, the bootstrap will skip all samples where the NONMEM run signal that some estimates are near its boundary. Disable with `-no-skip_estimate_near_boundary`.

**-allow\_ignore\_id**

Default not used. When not used, i.e. by default, bootstrap will print a message and terminate if an IGNORE or ACCEPT statement based on the ID column is found in the \$DATA record. This is done because it would interfere with the internal renumbering of individuals that the script does, producing errors. If `-allow_ignore_id` is used (not recommended), a warning is printed but the program continues execution. Note: The IGNORE statement can safely be used in conjunction with any other column than ID.

**-summarize**

Default not set. Only allowed when `-directory` is set to an existing bootstrap run directory and the `raw_results` file exist in there. Recompute `bootstrap_results` based on the existing `raw_results`, possibly using a different set of exclusion criteria.

**-copy\_data**

Default set. If option is set, the original data file will be copied to the run directory if the input model is to be run. If option is unset using `-no-copy_data`, the absolute path to the original data file will be used in \$DATA when the input model is run, and the data file will not be copied. This saves disk space.

**-dofv**

Optional, default not used. Evaluate original data with bootstrap parameter estimates and compute delta-ofv. See section Computing delta-ofv.

**-mceta** =  $N$

Optional, default not used. If option -dofv is set and NM version 7.3 or later is used, setting this option will make PsN set MCETA= $N$  in \$ESTIMATION. It is up to the user to check that the estimation method used can in NONMEM be combined with option MCETA, PsN will not do that.

## 2.3 Diagnostic R plots

PsN can automatically generate R plots to visualize results for bootstrap, using a default template found in the R-scripts subdirectory of the installation directory. The user can also create a custom template, see more details in section “Auto-generated R-plots from PsN” in common\_options.pdf.

See section Output, subsections Basic and Extended plots, for descriptions of the default bootstrap plots. The default bootstrap template requires the xpose4 R library of at least version 4.5.0, and that R libraries ggplot2, plyr, dplyr are installed. If the conditions are not fulfilled then no pdf will be generated, see the .Rout file in the main run directory for error messages.

**-rplots** = *level*

-rplots<0 means R script is not generated

-rplots=0 (default) means R script is generated but not run

-rplots=1 means basic plots are generated

-rplots=2 means basic and extended plots are generated

## Troubleshooting

If no pdf was generated even if a template file is available and the appropriate options were set, check the .Rout-file in the main run directory for error messages. If no .Rout-file exists then check that R is properly installed, and that either command 'R' is available or that R is configured in psn.conf.

## 2.4 Some important common PsN options

There are many options that govern how PsN manages NONMEM runs, and those options are common to all PsN programs that run NONMEM. For a complete list of such options see common\_options.pdf, or psn\_options -h

on the commandline. A selection of the most important common options is found here.

**-h or -?**

Print the list of available options and exit.

**-help**

With `-help` all programs will print a longer help message. If an option name is given as argument, help will be printed for this option. If no option is specified, help text for all options will be printed.

**-directory** = *'string'*

Default `bootstrap_dirN`, where `N` will start at 1 and be increased by one each time you run the script. The `directory` option sets the directory in which PsN will run NONMEM and where PsN-generated output files will be stored. You do not have to create the directory, it will be done for you. If you set `-directory` to a the name of a directory that already exists, PsN will run in the existing directory.

**-seed** = *'string'*

You can set your own random seed to make PsN runs reproducible. The random seed is a string, so both `-seed=12345` and `-seed=JustinBieber` are valid. It is important to know that because of the way the Perl pseudo-random number generator works, for two similar string seeds the random sequences may be identical. This is the case e.g. with the two different seeds 123 and 122. Setting the same seed guarantees the same sequence, but setting two slightly different seeds does not guarantee two different random sequences, that must be verified.

**-clean** = *'integer'*

Default 1. The `clean` option can take four different values:

**0** Nothing is removed

**1** NONMEM binary and intermediate files except INTER are removed, and files specified with option `-extra_files`.

- 2 model and output files generated by PsN restarts are removed, and data files in the NM\_run directory, and (if option -nmqual is used) the xml-formatted NONMEM output.
- 3 All NM\_run directories are completely removed. If the PsN tool has created modelfit\_dir:s inside the main run directory, these will also be removed.

**-nm\_version** = *'string'*

Default is 'default'. If you have more than one NONMEM version installed you can use option -nm\_version to choose which one to use, as long as it is defined in the [nm\_versions] section in psn.conf, see psn\_configuration.pdf for details. You can check which versions are defined, without opening psn.conf, using the command

```
psn -nm_versions
```

**-threads** = *'integer'*

Default 5 (if default PsN config file is used). Use the threads option to enable parallel execution of multiple models. This option decides how many models PsN will run at the same time, and it is completely independent of whether the individual models are run with serial NONMEM or parallel NONMEM. If you want to run a single model in parallel you must use options -parafile and -nodes. On a desktop computer it is recommended to not set -threads higher the number of CPUs in the system plus one. You can specify more threads, but it will probably not increase the performance. If you are running on a computer cluster, you should consult your system administrator to find out how many threads you can specify.

**-version**

Prints the PsN version number of the tool, and then exit.

### 3 Output

The file bootstrap\_results.csv contains statistics and summaries specific for the bootstrap:

- `diagnostic.means` - Means of all the diagnostics (such as `estimate.near.boundary` and `subproblem.est.time`) for the bootstrap runs.
- `means` - The means of parameter estimates, SEs etc. of the bootstrap distribution.
- `bias` - The difference between the bootstrap mean and the value for the original model
- `standard.error.confidence.intervals` - The confidence intervals of the estimates of the base model assuming that it is normally distributed with the standard deviation equal to that of the bootstrap distribution.
- `standard.errors` - The standard deviation of parameter estimates, SEs etc. of the bootstrap distribution
- `medians` - The median values of parameter estimates, SEs etc. of the bootstrap distribution
- `percentile.confidence.intervals` - Confidence intervals of the bootstrapped distribution. The intervals are calculated using the weighted average at  $x_{(n+1)p}$  method:
  1. Set  $n$  = the number of observations + 1
  2. Set  $p$  = number of the percentile to find / 100
  3. Set  $i$  = The integer part of  $n \cdot p$
  4. Set  $f$  = The decimal part of  $n \cdot p$
  5. Percentile =  $(1 - f)x_i + fx_{i+1}$

The details of which results are reported depend on the number of successful samples after filtering according to option `-skip_minimization_terminated` etc. The minimum count successful samples required to obtain percentile confidence intervals is 19 for interval 5%-95%, 39 for interval 2.5%-97.5%, 199 for interval 0.5%-99.5% and 1999 for interval 0.05%-99.95%.

The `raw_results` csv-file, where the exact name depends on the name of the model file, is 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 original dataset.

A file `delta_ofv.csv` is created if option `-dofv` is set, see section Computing `delta-ofv`.

`included_individuals1.csv`: One row per bootstrapped dataset. Each row consists of the ID:s of the individuals in that dataset. Note that different individuals may have the same ID number in a NONMEM dataset. The numbers appear in the order the individuals appear in the datafile.

`included_keys1.csv`: One row per bootstrapped dataset. Each row consists of the internal and unique order numbers (1-N) of the individuals in that dataset. The numbers appear in the order the individuals appear in the bootstrapped dataset.

`sample_keys1.csv`: One row per bootstrapped dataset. Each row consists of one number  $C$  per individual in the dataset, in the order the individuals appear in the original datafile, where  $C$  is the number of times the individual is included in the bootstrapped dataset.

The row order is consistent between the files `raw_results`, `included_individuals`, `included_keys` and `sample_keys` so that row  $j$  (excluding headers if any) in each of the files concerns the same bootstrapped dataset.

### 3.1 Basic plots

A basic bootstrap rplot will be generated in file `PsN_bootstrap_plots.pdf` if option `-rplots` is set  $>0$ , and the general rplots conditions fulfilled, see above. The plot is created using the `xpose4 boot.hist` function and shows histograms of each parameter in the model.

If option `-dofv` is set, there will also be a dOFV plot. The dOFV are computed as the difference between the OFV of the bootstrap parameter vector evaluated on the original dataset minus the OFV of the final parameter estimates on the original dataset. This plot shows 3 dOFV distributions:

1. that of all bootstrap samples, whether minimization on the bootstrap dataset was successful or not ('yes+no', red dotted line),
2. that of bootstrap samples for which minimization was successful ('yes', red full line) and
3. that of a chi-square distribution with degrees of freedom equal to the total number of estimated parameters in the model (blue full line).

In theory the bootstrap dOFV distribution should be at or below the reference chi-square. If it is not, another method for estimating parameter

uncertainty, such as SIR, should be considered. Differences between the 2 red curves indicate that bootstrap results are likely to differ depending on whether the termination status is taken into account or not.

## 3.2 Extended plots

Extended bootstrap rplots will be generated in file PsN\_bootstrap\_plots.pdf if option `-rplots` is set `>1`, and the general rplots conditions fulfilled, see above. The plots are created using the `xpose4 boot.hist` function and show histograms of standard errors, ofv and, if available from NONMEM output, eigenvalues. Be aware of the difference between OFV in the extended ofv plot, that correspond to OFV obtained by estimation on the bootstrap datasets, and the dOFV in Basic plots.

## 3.3 Computing a covariance matrix

It is possible to use the `covmat` program to compute a covariance matrix based on the bootstrap samples. The `-in_filter` option of `covmat` must be set to match the settings of bootstrap options `-skip_covariance_step_terminated`, `-skip_estimate_near_boundary`, `-skip_minimization_terminated` and `-skip_with_covstep_warnings`. The settings of those options can be checked in the `version_and_option_info.txt` file in the bootstrap run directory.

Option `-in_filter` is a comma-separated list of conditions. If bootstrap option `-skip_minimization_terminated` was true, then add `minimization_successful.eq.1` to `-in_filter`. If bootstrap option `-skip_estimate_near_boundary` was true, then add `estimate_near_boundary.eq.0` to `-in_filter`. If the bootstrap model has `$COV` and bootstrap option `-skip_covariance_step_terminated` was true, then add `covariance_step_successful.eq.1` to `-in_filter`. If the bootstrap model has `$COV` and bootstrap option `-skip_with_covstep_warnings` was true, then add `covariance_step_warnings.eq.0` to `-in_filter`. If the bootstrap model does *not* have `$COV`, then `-in_filter` should not include conditions on `covariance_step_successful` or `covariance_step_warnings`.

Example: If the bootstrap was run with

```
bootstrap run12.mod -samples=500 -dir=boot_run12 -skip_estimate_near_boundary
-skip_minimization_terminated
```

and `run12.mod` does *not* have `$COV`, then a covariance matrix can be obtained with (all in one command)

```
covmat run12.mod -rawres_input=boot_run12/raw_results_run12.csv -offset=1
-in_filter=minimization_successful.eq.1,estimate_near_boundary.eq.0
```

By default the matrix is printed to screen, but it can be redirected to a file using the `>` operator.

## 4 Known bugs and problems

It is recommended to remove PRINT options from \$ESTIMATION, to save disk space.

PsN rennumbers the individuals during bootstrapping. Therefore ACCEPT/IGNORE statements based on ID will cause errors. There is an input check for this, but it is recommended to not rely on this check. Also, model code that relies on ID numbers will also lead to errors.

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

## 5 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 and the user has not set `-no-run_base_model`. If the lst-file is present then PsN will simply read the estimates from that file.

The program creates N (N=number of samples), datasets of size M (M=sample\_size) by randomly drawing individuals with replacement from the original dataset. The program creates N new NONMEM modelfiles which are identical to the original modelfile with the exception that each uses a different bootstrapped dataset and, unless the user has set `-no-update_inits`, that the initial parameter estimates are the final estimates from the original run. The model parameters are estimated with each dataset, including the original, resulting in N+1 estimates for each parameter.

## 6 Computing delta-ofv

The delta-ofv distribution can be used as a diagnostic of the bootstrap [2]. If option `-dofv` is set, PsN will perform a `MAXEVAL=0` run for each set of bootstrap parameter estimates using the original model and data. This is done after all other runs and computations are completed, including any Bca step. The results will be printed to a csv file called `delta_ofv.csv` where the first column is bootstrap data id and the second delta-ofv which is computed as  $ofv_{bs,maxev=0} - ofv_{original}$ . If option `-rplots` is set  $>0$ , a plot with the delta-ofv distribution is generated, see Basic plots in section Auto-generated R-plots from PsN.

If option `-mceta=N` is set and NM7.3 or later is used and the estimation method is classical then option `MCETA=N` will be set in `$ESTIMATION`. There will be up to 200 `$PROBLEMS` defined in each model file (control stream) to reduce PsN overhead when calling `NONMEM`.

It is possible to restart a previously run bootstrap to compute delta-ofv without rerunning all bootstrap estimations, provided that option `-clean` was set to 2 or less in the original run. Simply run bootstrap again with option `-directory` set to the existing run directory. PsN will initiate reruns of all bootstrap samples, but reread existing output instead of really starting `NONMEM`. Then the delta-ofv models will be run. If option `-bca` was set in the original bootstrap run it is important to set `-bca` also in the restart, otherwise the jackknife results will be overwritten.

## 7 Recovering a crashed bootstrap

All the model files and bootstrapped datasets are in the `m1` subdirectory. Models that have finished will also have a `lst`-file and other `NONMEM` output files.

### Restarting crashed samples only

If there is a computer crash that either interrupts the main bootstrap process or causes a number of individual samples to fail, or the bootstrap is manually killed, you can still resume the bootstrap and reuse the samples that did finish.

1. Remove the entire `modelfit_dir1` subdirectory of the bootstrap run directory.
2. Rerun the same bootstrap command (`command.txt`) and set option `-directory=name_of_existing_directory`.

### **If you realize additional options such as `-retries` would have been needed**

If too many samples terminated due to e.g. rounding errors and you want to use `-retries` or `-picky` or similar for only those samples:

1. Go to the `m1` subdirectory and remove the `lst`-files for the runs that you wish to rerun. Output files from successful samples must be kept.
2. Remove the entire `modelfit_dir1` subdirectory of the bootstrap run directory.
3. Rerun the same bootstrap command (`command.txt`) but set option `-directory=name_of_existing_directory` plus the new options governing retries.

### **If too many samples are filtered out**

If too many samples were discarded due to e.g. minimization terminated and you wish to recompute the bootstrap results without rerunning any samples: Rerun the same bootstrap command (`command.txt`) but set options `-summarize` and `-directory=name_of_existing_directory` and change the run selection options such as `-no-skip_minimization_terminated`.

## **References**

- [1] B. Efron. *An Introduction to the Bootstrap*. London UK: Chap. & Hall, 1993.
- [2] R. Niebecker and M. O. Karlsson. “Are datasets for NLME models large enough for a bootstrap to provide reliable parameter uncertainty distributions?” In: *PAGE 22 (2013) Abstr 2899* (2013). URL: [www.page-meeting.org/?abstract=2899](http://www.page-meeting.org/?abstract=2899).