NPC/VPC userguide and technical description

2013-10-16 PsN 3.6.10

NPC – Numerical Predictive Check – is a model diagnostics tool. VPC – Visual Predictive Check – is another closely related diagnostics tool. A set of simulated datasets are generated using the model to be evaluated. Afterwards the real data observations are compared with the distribution of the simulated observations. By default no model estimation is ever performed. The input to the NPC script is the model to be evaluated, the number of samples (simulated datasets) to generate, parameter values options for the simulations, and stratification options for the evaluation. It is also possible to skip the simulation step entirely by giving two already created tablefiles as input. The input to the VPC script is the input to NPC plus an additional set of options. A large portion of the processing is identical between the scripts.

Getting started

If you are just starting to use VPC/NPC and only want the standard, no-frills vpc, you can skip over most parts of this guide. The document contains technical descriptions of all algorithms, which are not necessary to read before using the script. The last section, Known limitations and problems, is a recommended read for new users.

Points to consider if you are to use the VPC functionality for a new problem:

General

- How many simulations do you wish to base your VPC on? The more extreme percentiles you are interested in the more simulations are typically necessary. Specify this with option -samples.
- Where do you get the parameter estimates to simulate from? Either these needs to be specified in the NONMEM model file (as initial estimates) or you should supply an lst file, see the option -lst.
- If you are analyzing a time-to-event model then the processing done by VPC is very different from the regular case, and many options do not apply. Please read the section 'VPC for time-to-event models' before reading any other part of this guide.

The dependent variable (DV) y-axis data

- The default behavior is to use NONMEM output DV as the dependent variable. If you want to use anything else look into option -dv.
- Are your DV continuous or categorical? If you are dealing with categorical data look into option -levels and -noprediction.
- Do you have some kind of censoring of DV? If so look into the options -lloq, -uloq and/or -censor. Also read section Handling BQL data below. For time-to-event data, please read the section 'VPC for time-to-event models'.
- For time-to-event data, please read the section 'VPC for time-to-event models'.
- If you have multiple kinds of DV you will want to stratify on that. Read section Stratification for NPC/VPC and help text on options -stratify_on and -no_of_strata.

- Do you wish to apply any transformation on your DV data? If so look into option -lnDV.

The independent variable (IDV) x-axis data

- The default behavior is to use NONMEM output TIME as the independent variable. If you want to use anything else look into option -idv.
- Are your distribution independent variable such that binning is necessary? If you for example is using protocol time points as IDV this is perhaps not necessary whereas with actual observation time it is typically necessary. For binning options read the section Binning for VPC. Note that by setting -directory to the name of a previously created run directory the program can reuse the simulated data already produced, making it quite fast to test different binning alternatives.

Prediction/Variability correction

- If much variability can be attributed to other independent variables than the one represented on the x-axis it can be useful to perform prediction correction or prediction and variability correction. More about this subject can be read in: Bergstrand M, Hooker AC, Wallin JE, Karlsson MO. Prediction-Corrected Visual Predictive Checks for Diagnosing Nonlinear Mixed-Effects Models. AAPS J. 2011 Feb 8.

and in association to the options -predcorr and -varcorr.

A template call for a first try is (substitute as needed):
vpc run32.mod -lst=run32.lst -samples=500 -bin_by_count=1 -no_of_bins=10
Results are in the file vpc_results.csv (see details in section Output for vpc). Preferably use Xpose to visualize the results, because the file vpc_results.csv does not contain any graphs. Please refer to the Xpose documentation for instructions on visualization.

If you cannot make your model run at all with VPC or NPC (e.g. NMtran failure), look into option -keep_estimation or -sim_model.

Running VPC

Example: vpc moxonidine.mod -samples=1000 -seed=123 -idv=TIME -bin_by_count=1 -no_of_bins=5 -directory=npc_dir18 -stratify_on=WGT -no_of_strata=2 -mirrors=3 -dv=IPRED

Input: a modelfile is required.

-samples=<number>	is required, the number of simulated datasets to generate. 20 is the minimum accepted value.
-msfo=<file>	is optional, unless the model has $NONP record then an msfo-file is required. Use final parameter estimates from msfo-file instead of initial estimates from modelfile when simulating. VPC does not perform estimation.
-lst=<file>	is optional, but forbidden together with -msfo. Use final parameter estimates from this lst-file for the simulation. By default PsN will look for a file with the same name as the regular input model, or the simulation model if option -sim_model is used, but with suffix .lst instead of .mod.

	If such a file is found then option -lst=<modelfile>.lst is set automatically. VPC does not perform estimation.
-stratify_on=<variable>	is optional. See subsection Stratification options below. The stratify_on option may be a comma-separated list of several variable names.
-no_of_strata=N	is optional, only allowed when -stratify_on is used. See subsection Stratification options.
-refstrat=value	is optional, only allowed when stratify_on is used, not allowed in combination with -no_of_strata. See subsection Stratification options.
-tte=<variable>	is an optional argument, but it is required to assume (repeated) time-to-event type models. Cannot be used together with option -mirror, -censor, -predcorr, -varcorr, -lnDV, -uloq, -lloq or -boxcox_lambda. The tte-variable needs to be defined in the simulation model and must take the value 0 if the observation is not an event and non-zero if it is an event (including events and censored events). PsN will add the tte-variable to $TABLE of the simulation model (not the original data model). PsN will format the output differently compared to a regular vpc, to specifically suit the kaplan.plot functionality in Xpose (e.g. filtering of simulation output based on tte-variable). See section VPC for Time-to-event models below.
-sim_model=<file>	is optional. Cannot be used together with -flip_comments, -keep_estimation or -noprediction. By default PsN will create a simulation model based on the required input model, but by using option -sim_model it is possible to use a separate input model for the simulations. PsN will remove $COV and $TABLE, change SEED and NSUBS in $SIM, add a new $TABLE and update initial estimates if option -lst is set or add $MSFI if option -msfo is used, but otherwise no changes will be made to the user defined simulation model. See section Modified models. Note that -lst will be set automatically if a file with the same name as the regular input model but with suffix lst intead of mod is found.
-flip_comments	is optional. Cannot be used together with -sim_model, -keep_estimation or -noprediction. By default PsN will create a simulation model based on the required input model, but option -flip_comments invokes a method for handling user-defined simulation code in the required input model. If option is set, PsN will create the simulation model by flipping comments (commented lines will be uncommented and vice versa) between the tags ;Sim_start and ;Sim_end For example, if the required input model has lines ;Sim_start IGNORE(TYPE.EQ.1) ;ACCEPT(TYPE.EQ.1) ;Sim_end then the MAXEVAL=0 model will be run as such and the simulation

	model will instead have lines ;IGNORE(TYPE.EQ.1) ACCEPT(TYPE.EQ.1) The tags may appear multiple times. Note that the tags must look exactly as above or the editing will fail. When creating the simulation model PsN will remove $COV and $TABLE, change SEED and NSUBS in $SIM, add a new $TABLE and update initial estimates if option -lst is set or add $MSFI if option -msfo is used, but otherwise no changes will be made to the code. See section Modified models.
-dv=<variable>	is optional, default is DV. If a synonym for DV is set in $INPUT, the synonym must be set as the dependent variable on the commandline, -dv=<synonym>.
-keep_estimation	is optional, by default not set. If this option is set, a post-hoc evaluation step is performed for each simulated dataset ($ESTIMATION is kept and MAXEVALS is set to 0). Note that in this case variables such as IPRED(F) are based on the re-estimated post-hoc parameters. Also note that in earlier program versions keep_estimation was set or unset automatically, see section Additional rules and logic [3].
-noprediction	is optional, by default not set. If set, NOPREDICTION will be added to the $SIMULATION record of the simulation model, in addition to ONLYSIMULATION. This option is generally recommended with likelihood models for odd type data (i.e. -2LOGLIKELIHOOD or LIKELIHOOD in $ESTIMATION). It is not allowed to use -noprediction in combination with the option -keep_estimation.
-orig_table=<filename>	is optional, only allowed when -sim_table is also used. Give tablefile with original data as input directly instead of letting the program generate them.
-sim_table=<filename>	is optional, only allowed when -orig_table is also used. Give tablefile with simulated data as input directly instead of letting the program generate them.
-n_simulation_models=N	is optional, default is 1. The default 1 means all simulations are run in the same modelfile. By setting this option to a number N greater than 1, the 'samples' simulations will be split equally between N model files, which can be run in parallel. This option cannot be used together with option -sim_table or, if the NONMEM version < 7, together with -dv=CWRES. Important: Two runs will give different results if -n_simulation_models are set to different numbers even if the -seed option is the same. This is because the random number generator of NONMEM will change state when reading the seed from the $SIM record from a new simulation modelfile. This state change does not occur when NONMEM simply continues simulating from the same modelfile.
-censor=<variable>	is optional, default not used. Name of variable which defines whether the observation of the dependent variable is missing, e.g. due to drop-out. 1 means the observation is censored, 0 means the observation is not censored. The variable must be requestable in $TABLE. This option is

	not applicable for time-to-event data, please read the section 'VPC for time-to-event models'.
-confidence_interval=CC	is optional, default is 95. The confidence interval in percent.
-fine_pi	is optional, default not set. If not set, the the prediction intervals computed are 0, 40, 80, 90 and 95%. If the option is set, then the prediction intervals computed are 0, 10, 20, 30, 40, 50, 60, 70, 80, 90 and 95%.
-mirrors=N	optional, the number of mirror plot data sets to produce. Specific VPC.
-idv=<variable>	optional, default TIME, the independent variable to bin on. Specific for VPC. (old option was -bin_on, without default value)
-Either of 12 binning option combinations for independent variable	optional – it is forbidden to use more than one of these combinations. For details see section VPC binning below. • -autobin=auto (default) • -autobin=<number of bins> • -autobin=<start bins>, <stop bins> • -autobin=unique • -bin_by_count=0 -bin_array=boundary1,boundary2,... • -bin_by_count=1 -bin_array=count1,count2,... • -bin_by_count=0 -no_of_bins=X • -bin_by_count=1 -no_of_bins=X • -bin_by_count=0 -single_bin_size=X -overlap=Y • -bin_by_count=1 -single_bin_size=X -overlap=Y • -bin_by_count=0 -single_bin_size=X • -bin_by_count=1 -single_bin_size=X
-min_points_in_bin=N	Optional, used only if -autobin (except -autobin=unique) is selected. Set to disallow bins of size smaller than N. Default is 10.
-lloq=<number>	optional, the lower limit of quantification for left censored data. Specific for VPC.
-uloq=<number>	optional, the upper limit of quantification for right censored data. Specific for VPC.
-levels=<number1,number2>	optional, the boundaries for a categorization. Category 1<=number1 < category 2 <= number2, etc. Specific for VPC.
-predcorr	optional. Perform prediction correction of dependent variable values. Specific for VPC. For details see section VPC predcorr and varcorr below. If the dependent variable is log transformed or has a lower bound not equal to 0 it can be important to specify the optional variables -lnDV and lower_bound (see below). The predcorr feature will not work well if there is an upper bound, e.g. when the dependent variable is logit transformed.
-varcorr	optional, only allowed together with -predcorr. Perform variability correction of dependent variable values. Specific for VPC. For details see section VPC predcorr and varcorr below.
-lnDV=<0,1,2,3>	optional, default 0. Mainly for use with -predcorr. lnDV=0 or 1may be used independently of -predcorr, but -lnDV=2 is only allowed together with predcorr. Variable to indicate if the dependent variable is log-transformed (ln(DV)) or not and how that should be handled.

	• -lnDV=0 = Default. No exponentiation of DV will be performed. In combination with -predcorr -lnDV=0 indicates that DV is untransformed. • -lnDV=1 = DV and PRED values will be exponentiated. Indicates log-transformed DV and that VPC results should be presented on normal scale. May be used independently of -predcorr. • -lnDV=2 = Only in combination with –predcorr. Indicates log-transformed DV that should be maintained. • -lnDV=3 = DV and PRED values will be log-transformed. May be used independently of -predcorr.
-boxcox_lambda=x	Setting this variable indicates that data is Box-Cox transformed. Data will be transformed to normal scale for eventual prediction correction, and retransformed to Box-Cox before analysis and output. Option cannot be used with lnDV. The transformation is (DV*lambda+1)**(1/lambda) for Box-Cox to normal, and (DV**lambda-1)/lambda for normal to Box-Cox.
-lower_bound=<either a number or a name of an independent variable>	optional, default not set . Only allowed in combination with predcorr and -lnDV=0. Lower boundary for typical model prediction (PRED). The specified boundary must be strictly lower than the lowest value. If the boundary is given as a number, that number will be used for all observations. If instead a variable is given then it must be independent, and defined either in the model file or present in $INPUT. If -lower_bound is not set but -predcorr is used, a lower bound of 0 will be assumed.
-rawres_input=filename	A simple way to simulate with uncertainty. Note that it is normally not appropriate to do this in a vpc or npc. Instead of using identical parameter estimates for simulation of each new dataset, take parameter estimates from a raw_results.csv file, e.g. from a bootstrap run or the intial_estimates.csv file from a previous sse run with $PRIOR in the simulation model. The raw results file must be comma-separated and contain at least as many samples as the input -samples to sse, the labels for THETA/OMEGA/SIGMA in the file must match the labels in the simulation model given as input to sse, the theta columns must be directly followed by the omega columns which must be directly followed by the sigma columns, and the column header must be model either in the first column just as in a bootstrap raw_results file or in the second column as in a sse raw_results file. If a column header contains a comma, e.g. OMEGA(2,2), then that header must be enclosed in double quotes. This is done automatically in PsN raw results files. Note that is is possible to generate a file with initial parameter estimates outside of PsN, as long as the file follows the format rules.
-offset_rawres=N	Only relevant in combination with -rawres_input. Default 1. The number of result lines to skip in the input raw results file before starting to read final parameter estimates. In a regular bootstrap raw_results file the first line of estimates refers to the input model with the full dataset, so

	therefore the default offset is 1.
-copy_data	Default set. Disable with -no-copy_data. By default, PsN will copy the datafile into NM_run1 and set a local path in psn.mod, the actual modelfile run with NONMEM. If -no-copy_data is set, PsN will not copy the data to NM_run1 and instead set a global path to the datafile in psn.mod. However, if the global path is very long then there will be an NMtran error caused by a line wrap in $DATA in psn.mod.

Simulation input details

The option -samples is required. The scripts does not allow -samples to be smaller than 20, but in order for the analysis to produce meaningful results samples needs to be much larger. No model estimation is performed for the simulated datasets, except for a post-hoc estimation step in case -keep_estimation is set (then MAXEVALS=0 in $ESTIMATION). There are five ways of choosing which parameter values are to be used in the simulations:

a) Default: the initial estimates from the lst-file with the same name as the modelfile but with .mod replaced with .lst, e.g. run123.lst if the modelfile is run123.mod. If no such lst-file exists the estimates from the modelfile are used.
b) the final estimates from a lst-file whose name is not the modelfile name with .mod replaced with .lst: use command-line option -lst=<filename>
c) the final estimates from an msfo-file: use command-line option -msfo=<filename>
d) final estimates from a raw_results file, e.g. from a bootstrap. This method implies using a different set of estimates for each sample.
e) parameter estimates drawn from a prior distribution defined using $PRIOR in the input/simulation model.

Alternatives d) and e) result in simulation with uncertainty. Note that this is normally not appropriate for a vpc or npc.

The user may either skip the $SIMULATION record entirely and let the program produce it according to the rules specified in section Modified models – model for simulated data. Or the user can include a complete $SIMULATION record, for example when using special random distributions. Inclusion of a $SIMULATION record is needed when simulating categorical data is intended (vpc with -levels option). In this case the model file must be equipped with IF(ICALL.EQ.4) coding to separate the simulation model from the estimation model, and the $SIMULATION record must contain both a normal seed and a uniform seed (1234 UNIFORM). If there is a user-defined $SIMULATION record, the program will replace the random seeds, set NSUBPROBLEMS to the number of samples requested on the input line, check that TRUE=FINAL is set in the case when the option -msfo is used, and set/unset ONLYSIMULATION to be compatible with the option keep_estimation. If option rawres_input is used, the program will create one simulation model per sample.

An additional alternative is to supply the program with a complete simulation model, either via option -flip_comments or option -sim_model. In both cases the program will only change seeds and NSUBPROBLEMS in $SIMULATION. No other changes to that particular record will be made.

Handling BQL data

It is important to retain the BQL observation in the data-set when doing VPCs (see Bergstrand, M. et al. AAPS J, 2009. 11(2): p. 371-380. for more details). It should be done irrespectively of what method that is used to handle or not handle the BQL observations when estimating. The M3/M4 code used to include BQL samples in the estimation is only applicable to estimation and not to simulation. With the M3/M4 code the BQL observations are treated as categorical observations (i.e. <LOQ). When you simulate you want to generate simulated concentration observations at all time points in your data-set (also where the observations was <LOQ). If you simulate with the M3/M4 code active the prediction at BQL samples in your data-set (F_FLAG=1) will be the probability for the observation to be <LOQ and not a concentration. For this reason PsN highlights that if F_FLAG is used there needs to be a special simulation block (ICALL.EQ.4) where the prediction is coded to be continuous (F_FLAG=0) for all observations. In theory it should be possible to have the M3/M4 cod in a block that only applies to estimation but NONMEM often gives error messages when this is tried. For that reason it is recommended to construct a new control stream for the VPC (e.g. run1vpc.mod). In the new model file remove code that relates to the M3/M4 method so that continuous predictions are given at all observations. It is important that the DV value for all BQL observations are set to a value below LOQ (e.g. LOQ/2) in the data-set.

Simulation uncertainty

It is possible to simulate with uncertainty in npc/vpc, either by using option -rawres_input or by having $PRIOR in the input/simulation model. Note that including uncertainty in vpc or npc simulations is normally not appropriate.

Stratification options

-stratify_on=<variable> The user can stratify on a variable found in $INPUT or define a variable called STRT in the model code and then set -stratify_on=STRT. The later alternative allows the user to do stratification on any combination of data columns or model variable, since STRT may be defined in any way the user chooses. The vpc program will add STRT to the $TABLE record, and if STRT is not defined in the model-file NONMEM will exit with an error.

The user may set -stratify_on to a comma-separated list of variables. In that case stratification will be performed on the first variable in the list, and the following variables will simply be added to $TABLE. This way the user can request any extra variables in $TABLE, for example for altenative stratification in later analyses without having to rerun any models.

The user may not stratify on DV even if it is in the $INPUT record, since this variable will differ between simulations of the same observation. It is however possible to stratify on e.g. PRED.

Setting -no_of_strata=N means group the observations into N groups by the value of the stratification variable so that the number of observations is approximately equal in each group.

Setting -refstrat=<value>, e.g. -refstrat=0 when -stratify_on=DOSE is set, makes vpc compute all statistics also for the delta-mean, i.e. the difference between the dependent variable mean in the current stratum and the mean of the dependent variable in the reference stratum, which is the stratum with stratification_variable=refstrat (for example DOSE=0). In the reference stratum the delta-mean is of course 0.

Dependent variable option, -dv
The default dependent variable to do the analysis on is DV. Other possible choices are IPRED, IRES, IWRES, IPRED, RES, WRES or any other variable defined in the $PRED/$ERROR/$PK record in the modelfile. Defining a new variable allows the user to, for example, perform any sort of transformation of the observations. CWRES may also be chosen, but Xpose (a post-processing program in R) may not be able to handle large tablefiles and tablefiles lacking a run number.

Input details for NPC/VPC without simulation
It is possible to skip the simulation step entirely by specifying both options -orig_table and -sim_table. Binning and stratification options are the same. The two tablefiles must have exactly the same format as would have been produced by NPC/VPC had the program performed the simulations. See section Additional rules and logic [1] for a list of which columns must be present in the tablefiles. It is important that the rules for including or not including MDV are followed. Also, the individuals and observations must come in the same order for both the original data and the simulated datasets. No check is performed that e.g. ID numbers match. Please note that a modelfile is still required as input, even though it will not be used.

Common options in PsN which are extra important for NPC and VPC (examples):
For a complete list of common options see common_options_defaults_versions.pdf, or psn_options -h on the commandline.

-last_est_complete	optional and only applies with NONMEM7 and if option -keep_estimation is set. See section NPC and VPC with NONMEM7.
-seed=<random seed>	Using this option makes it possible to regenerate simulation results.
-directory=<directory name>	name of new directory or directory to restart from
-nm_version=<nonmem version>	

Important note: you can use the -directory option e.g when you have run a simulation with a particular model earlier and you want to stratify on a new column or use a different set of binning options. You can reuse NPC simulations in a VPC analysis and vice versa, since the underlying program is essentially the same. The default directory name will be of the form npc_dirN, regardless of whether the original run was done using NPC or VPC. This will save time since much of the processing won't have to be redone. The old npc_results.csv or vpc_results.csv file will be renamed so that they are not overwritten.

Running NPC

As stated above, the NPC and VPC have many parts in common. All NPC options are available for VPC, and the same rules for them apply.

Example: npc moxonidine.mod -lst=moxonidine.lst -samples=500

Input: a modelfile is required.

-samples=<number>	is required, the number of simulated datasets to generate.
-msfo=<file>	is optional, unless the model has $NONP record then an msfo-file is required. As with VPC.

-lst=<file>	is optional, but forbidden together with -msfo. Use final parameter estimates from this lst-file. Npc does not perform estimation.
-stratify_on=<variable>	is optional.
-no_of_strata=N	is optional, only allowed when -stratify_on is used.
-refstrat=value	is optional, only allowed when -stratify_on is used.
-dv=<variable>	is optional, default is DV. If a synonym for DV is set in $INPUT, the synonym must be set as the dependent variable on the commandline, -dv=<synonym>.
-keep_estimation	is optional, by default not set. As VPC.
-noprediction	is optional, by default not set. As VPC.
-orig_table=<filename>	is optional, only allowed when -sim_table is also used. As VPC.
-sim_table=<filename>	is optional, only allowed when -orig_table is also used. As VPC.
-n_simulation_models=N	is optional, default is 1. As VPC.
-confidence_interval=CC	is optional, default is 95. The confidence interval in percent.
-censor=<variable>	is optional, default not used. As VPC.
-sim_model=<file>	is optional. As VPC.
-flip_comments	is optional. As VPC.
-rawres_input=<file>	is optional. As VPC.
-offset_rawres=N	optional. As VPC.
copy_data	Set by default. As VPC.

VPC for time-to-event models

This section only gives a technical description of what PsN does in case option -tte is set. Please refer to the Xpose documentation on kaplan.plot for how the data PsN produces is used to create Kaplan-Meier curves.

If option -tte is set, the vpc process is different from the regular case. The only sections below in this document that apply are Input checks, Modified models, NPC and VPC with NONMEM7, and Additional rules and logic. If option -stratify_on is used together with -tte, the program will simply output the stratification parameter in the output table. It is allowed to set -stratify_on to a comma-separated list of variable names, and then all those variables will be included in the output table.

The variable set with option –tte is used for filtering the simulated time-to-event data (events and censored events) for production of Kaplan plots in Xpose, see below. The tte-variable needs to be defined in the simulation model and be equal to 0 when there is no event, and non-zero when there is an event (event and censored event). All tte-variable=0 will be filtered from the final simulation dataset. The DV variable must define an event (DV=1) or a censored event (DV=0). i.e. for a simulated event DV=1 and tte-variable = 1; for a simulated censored event DV=0 and tte-variable = 1; for all other simulated records DV=0 and tte-variable equals 0. Thus, censored data are indicated with the DV variable and the tte-variable and will match the requirements of kaplan.plot in Xpose. This coding of censoring is opposite the choice for the vpc -censor option (the DV variable here is an event indicator not a censoring indicator), and the –censor option cannot be used with -tte as already stated. An example of NONMEM code (not a full model) is given below where the tte-variable is RTTE:

```
$PK
  ;for simulation
  RTTE=0
  ;end for simulation
.
.
$ERROR
.
.
   ; for simulation
  IF(ICALL.EQ.4) THEN
    CALL RANDOM (2,R)
    DV=0                    ; no event occurs
    RTTE=0                  ; no event occurs
    IF(TIME.EQ.500) RTTE=1; a censored event, code needs to be written so that
                            this will occur at the last record within each in-
                            dividual, in this case TIME could be used, but will
                            be dependent on the set-up of the simulation data
                            set.
    IF(R.GT.SUR) THEN       ; an event, SUR is survival probability defined in
       DV=1                   code but not shown here
       RTTE=1
    ENDIF
  ENDIF
```

For the original data, kaplan.plot in Xpose requires that EVID is present and the original data set will need the EVID. If not included, all records in the original data set will be assumed to be observation records.

The vpc script checks the input, see section Input checks. Then two modified modelfiles are created, or the two input models are updated: one for generating original data table output and one for generating simulated data table output, see section Modified models. Note that if option -tte is set, PsN will automatically remove all IGNORE/ACCEPT statements, except IGNORE=@ or similar, when creating the simulation model (unless -sim_model or -flip_comments is set, see help text for these options). In the model for original data the IGNORE/ACCEPT statements are not changed. If deleting IGNORE/ACCEPT for the simulations is not enough to create a correct simulation model then the user must instead supply PsN with a separate simulation model, using option -sim_model or -flip_comments.

The model parameters will never be reestimated. If an lst-file is given as input or an lst-file is found by replacing .mod with .lst in the modelfile name, the initial estimate in both models will be updated using the final estimates from the lst-file. If an msfo-file is given using option -msfo both models will have an $MSFI record added. Otherwise the initial parameter estimates will not be changed in the two modelfiles.

The two modified models are run using PsN. The real data is only filtered on already present ACCEPT/IGNORE statements in the input model, and the table is named mytab<runno> to suit the automatic data reading of Xpose. If -stratify_on has been used, then all variables defined will be included in mytab<runno>. The simulation output is filtered on the variable specified using option -tte: all lines where the value in the tte column is equal to 0 are removed. Variables given using option

stratify_on will be included in simulation output. Also, three additional columns are created: one for the accumulated number of events for the (simulated) individual, one for the order number of the simulated individual (a number between 1 and “samples”*N_individuals), and one for the order number of the simulation (a number between 1 and “samples”). Finally the formatted simulated output is zipped. If PsN fails to zip the file it must be done manually before Xpose can be used to produce the final plots using command kaplan.plot.

If the regular input model file name has the form run<runno>.mod, where <runno> is a number, PsN will move the zipped simulated data simtab<runno>.zip up to the calling directory, to simplify data reading using Xpose. The real data table mytab<runno> is copied to the calling directory. If a file with the same name already exists the original file will be renamed mytab<runno>_original, unless mytab<runno>_original already exists in which case the old mytab<runno> is overwritten.

If the regular input model file name does not have the form run<runno>.mod, PsN will not move or copy anything to the calling directory but keep the output files in the vpc run directory and name them simtab1.zip and mytab1.

Example commands in Xpose to plot the vpcs (once library(xpose4) is loaded), assuming that the input model is named run1.mod:
>runno <- 1
>xpdb <- xpose.data(runno)
>kaplan.plot(object=xpdb,VPC=T)

Output for NPC

The NPC output is in npc_dir<number>, unless a different directory name was specified using the -directory option. The original data run with MAXEVAL=0 or similar is performed in the subdirectory simulation_dir/NM_run1 and the simulations in simulation_dir/NM_run2 and up.

The final results are found in npc_results.csv which is suitable for viewing in excel or oocalc. The file starts with a section with general run information: the date, the number of observations in the dataset, the number of simulations/samples, the name of the modelfile, the file from which the simulation parameters were taken, which dependent variable the analysis was made on, the name of the PsN version and the NONMEM version. Following the general section there is one result section per strata, which are analyzed independently of each other (see section Dependent variable matrix analysis for NPC). If the stratification option was not used, there is only one result section. Each result section has two parts. The header of the first part gives the stratification variable values of the observations in the strata, and how many observations out of the total number belongs to the strata. There is one row per prediction interval, abbreviated PI. The rows are independent of each other, and each row is an analysis of all observations in the strata. The values reported in each column are explained in section Dependent variable matrix analysis for NPC. The second part of the result section is diagnostics via NPC * count statistics. These are produced according to section NPC results diagnostics.

In npc_dir<number>/m1/<dv variable>_matrix.csv you find the extracted DV values, one row per observation (after filtering on MDV). The first column is the real data, the remaining columns are

simulated data.

If the independent variable is DV, and a synonym was used in the $INPUT record, the synonym will be used in the output instead of DV. If the stratification variable given on the command-line was a reserved label and a synonym was defined in the $INPUT record, the synonym will be used in all output instead of the reserved label.

Output for VPC

VPC output is found in npc_dir<number>, unless a different directory name was specified using the -directory option. The original data run with MAXEVAL=0 is done in subdirectory simulation_dir/NM_run1 and the simulations in simulation_dir/NM_run2 and up.

The final results are in the file vpc_results.csv. Empty bins are skipped completely and not reported. The file starts with a section with general run information: the date, the number of observations in the dataset, the number of simulations/samples, the name of the modelfile, the file from which the simulation parameters were taken, the independent variable used for binning, which dependent variable the analysis was made on, the name of the PsN version and the NONMEM version.

Following the general section in vpc_results.csv there is one sequence of result sections per strata. The strata are analyzed independently of each other (see section Dependent variable matrix analysis for VPC). If the stratification option was not used, there is only one sequence of result sections. The header of the Continuous data section gives the stratification variable values of the observations in the strata, and how many observations out of the total number belongs to the strata. Then there is one row per bin (for binning rules see section binning for VPC). The first two columns contain the independent variable boundaries for the bin, and the third the number of observations in the bin. The following columns come in groups of four, one group per percentage boundary, where the values are the results from steps (b), (a) and (c) in Dependent variable matrix analysis for VPC. To the right of the four-column groups there are diagnostics columns with false positives and false negatives percentages according to section VPC results diagnostics, step (i). The diagnostics columns are omitted in the case of censored data. After the Continuous data section comes comes the Diagnostics section, computed as described in VPC results diagnostics, step (ii). The Diagnostics section is omitted in the case of right/left censored data. For right/left censored data or missing data there is a Censored data section. The first three columns in the Censored data results section give the bin boundaries and the number of observations in the bin. The Censored data section is described further in step (f) in Dependent variable matrix analysis for VPC. If option -levels is used, there is a Categorical data section. The first three columns give the bin boundaries and the number of observations in the bin, and the following column are described in step (g) in Dependent variable matrix analysis for VPC. Last comes a sequence of regular NPC results parts (without NPC diagnostics), one for each bin, described in section Output for NPC.

There is a file npc_dir<number>/m1/<dv variable>_matrix.csv with the extracted dependent variable values. If option -censor is used there is a file npc_dir<number>/m1/<censor variable>_matrix.csv with the extracted censoring variable values.

The csv table file named vpctabXX, where XX is either empty or the number from the modelfile if it is of the form runXX.mod, contains one header row with ID,<DV-column header>,<independent column header>,strata_no,<stratification variable header>,mirror_1,mirror_2. There are no mirror headers

unless the option -mirrrors has been used, and no stratification headers unless the option -stratify_on has been used. The ID column is taken directly from the original data input file, so the ID-numbers may not be unique between individuals. The strata_no column numbers the strata from 1-no_of_strata, and is a translation from the original stratification variables values into one integer per strata.

If the independent variable is DV, and a synonym was used in the $INPUT record, the synonym will be used in the output instead of DV. If the independent/stratification variable given on the command-line was a reserved label and a synonym was defined in the $INPUT record, the synonym will be used in all output instead of the reserved label.

General process

The script checks the input, see section Input checks. Then two modified modelfiles are created, one for generating original data table output and one for generating simulated data table output. The model parameters will never be reestimated. Unless an lst- or msfo-file is given as input or an lst-file is found by replacing .mod with .lst in the modelfile name, the initial parameter estimates from the modelfile will be used for the simulations. The two modified models are run using PsN, and the output in the tablefiles are processed and analyzed according to sections Table output processing and DV matrix analysis for NPC/VPC below.

Input checks

- If a variable to stratify on is specified on command-line, it must be either PRED, STRT or a variable in the $INPUT record, otherwise a warning will be printed. If it is in the $INPUT record it must not be DV, and it is not allowed to DROP/SKIP it. If the variable to stratify on is STRT, the user must define it in the $PK, $PRED or $ERROR record, on a line beginning with the name of the variable (IF (...) STRT = 1 will not work), otherwise a warning will be printed.
- Options -flip_comments and -sim_model cannot be used together.
- If option -flip_comments or -sim_model is used, options -noprediction and -keep_estimation and -orig_table cannot be used.
- If option -tte is used then options -mirrors, -censor, -predcorr, -varcorr, -lnDV, -boxcox_lambda, -uloq or -lloq cannot be used.
- The modelfile must contain a single problem only.
- samples must be defined and at least 20.
- It is not allowed to give both an lst-file and an msfo-file as options.
- If the modelfile contains a $NONPARAMETRIC record, an msfo-file is required.
- It is verified that the lst-file, if given, can be parsed correctly.
- The independent variable in VPC must either be PRED or defined in the $INPUT record, where it is not allowed to DROP/SKIP it, otherwise a warning will be printed. It is not allowed to use DV as the independent variable.
- Binning options must match one of the alternatives in section Binning for VPC below.
- There may be either one or zero $SIMULATION records in the modelfile.
- If there is one $SIMULATION record in the modelfile and there is a $NONPARAMETRIC record, the option TRUE=FINAL must be set.
- The dependent variable (-dv) cannot be PRED, MDV, TIME, DATE, EVID, AMT, ID nor RATE.
- Unless the NONMEM version is 7 or higher, the dependent variable name must have at most 4 letters, unless it is IPRED, IWRES or CWRES.

- If the -dv option is CWRES then IPRED must be defined in the $PK, $PRED or $ERROR record.
- If the -dv variable is neither of DV, RES, WRES or CWRES it must be defined in the $PK, $PRED or$ERROR record.
- -mirrors, the number of mirror plot data sets to generate in VPC, can be at most 20.
- If reanalyzing an old run (i.e. the -directory=<directoryname> option is used and <directoryname> contains the results of an old run), the modelfile name must be the same in the new and original calls.
- If reanalyzing an old run, -lst option must either be used in both or neither of the calls.
- If reanalyzing an old run, -msfo option must either be used in both or neither of the calls.
- If reanalyzing an old run, -samples must be the same in both calls.
- If reanalyzing an old run, and keep_estimation is set, keep_estimation must have been set also in the original call.
- It is not allowed to use synonyms for ID or MDV in the $INPUT record.
- If both -lloq and -uloq are used, the uloq value must be larger than lloq.
- If -levels is used, the values must be sorted in increasing order.
- If the $ESTIMATION record contains -2LOGLIKELIHOOD/LIKELIHOOD or if $ERROR or $PRED contains F_FLAG, there must be an IF(ICALL.EQ.4) string in $ERROR, $PRED or $PK. Also, the -levels option is recommended (but not required).
- It is not allowed to set both -noprediction and -keep_estimation.
- lnDV must be either 0, 1, 2 or 3.
- if lnDVis 1 and either of (levels, uloq, lloq) is specified then a warning will be printed that levels must be given on the normal scale
- if lnDVis 3 and either of (levels, uloq, lloq) is specified then a warning will be printed that levels must be given on the log-scale
- if lnDV is 1 or 3 and stratify_on=PRED then a warning will be printed that stratification is done before transformation and thus strata boundaries will be presented on the untransformed scale.
- if lnDV is 2 then -predcorr must be used
- if -lower_bound is specified, then -predcorr must also be specified
- if -lower_bound is specified, then -lnDV must not be larger than 0.
- if -varcorr is specified then -predcorr must also be specified.

Modified models

Neither -flip_comments or -sim_model used

Model for original data:

- Any $TABLE record is removed, and a new one is added, see [1].
- Any $SIMULATION record is removed.
- any $COVARIANCE record is removed
- If an lst-file was given on command-line, the initial estimates are changed to the values in the lst-file.
- If neither an lst-file nor an MSFO-file was specified on the command-line, but an lst-file is found when replacing .mod with .lst in the model file name, the initial estimates are changed to the values in that lst-file.
- If an MSFO-file was specified on command-line, a $MSFI record is added and records $THETA, $OMEGA, $SIGMA are removed.
- Any $SCATTERPLOT record is removed.

- In $ESTIMATION record, if it exists, MAXEVAL is set to 0 or corresponding for NONMEM7 (see section NPC and VPC with NONMEM7, and PsN_and_NONMEM7.pdf).
- If keep_estimation is set: In the $ESTIMATION record option POSTHOC is set.
- In $ESTIMATION record, if it exists, option MSFO is removed.
- If $PRIOR exists, then remove option PLEV.

Model for simulated data:

- If option -tte is set: remove all IGNORE/ACCEPT statements, except any IGNORE=<single character>, e.g. IGNORE=@.
- If keep_estimation is set: In the $ESTIMATION record option POSTHOC is set, option MAXEVAL is set to 0 or corresponding for NONMEM7 (see section NPC and VPC with NONMEM7, and PsN_and_NONMEM7.pdf), option MSFO is removed, if present. In $SIMULATION the option ONLYSIMULATION is unset.
- If keep_estimation is not set, all $ESTIMATION records are removed, and the option ONLYSIMULATION is set in the $SIMULATION record.
- Except for changes in $EST (see above) and that $SIM is kept if present, the same changes as for the original data.
- If there was a $SIMULATION record in the original modelfile, all random seeds are replaced.
- In $SIMULATION, option NSUBPROBLEMS is set to number of samples.
- A $SIMULATION record is added if there was none in the original modelfile. If the model has a $PRIOR record then TRUE=PRIOR is set, or else if an msfo-file was given on the command-line, TRUE=FINAL is set. One random seed is always given (with no attribute). If there is a $NONP record, a second seed NONPARAMETRIC is added and the $NONP record is removed.
- If -noprediction is set, NOPREDICTION is set in the $SIMULATION record.

If option -flip_comments or -sim_model is used

Model for original data:

- Any $TABLE record is removed, and a new one is added, see [1].
- Any $SIMULATION record is removed.
- Any $COVARIANCE record is removed
- If an lst-file was given on command-line, the initial estimates are changed to the values in the lst-file.
- If neither an lst-file nor an MSFO-file was specified on the command-line, but an lst-file is found when replacing .mod with .lst in the model file name, the initial estimates are changed to the values in that lst-file.
- If an MSFO-file was specified on command-line, a $MSFI record is added and records $THETA, $OMEGA, $SIGMA are removed.
- Any $SCATTERPLOT record is removed.
- In $ESTIMATION record, if it exists, MAXEVAL is set to 0 or corresponding for NONMEM7 (see section NPC and VPC with NONMEM7, and PsN_and_NONMEM7.pdf).
- In $ESTIMATION record, if it exists, option MSFO is removed.
- If $PRIOR exists, then remove option PLEV.

Model for simulated data:

- Option MAXEVAL is set to 0 or corresponding for NONMEM7 in $ESTIMATION, if

$ESTIMATION exists.
- In $ESTIMATION record, if it exists, option MSFO is removed.
- Any $TABLE record is removed, and a new one is added, see [1].
- Any $COVARIANCE record is removed
- If an lst-file was given on command-line, the initial estimates are changed to the values in the lst-file.
- If neither an lst-file nor an MSFO-file was specified on the command-line, but an lst-file is found when replacing .mod with .lst in the model file name, the initial estimates are changed to the values in that lst-file.
- If an MSFO-file was specified on command-line, a $MSFI record is added and records $THETA, $OMEGA, $SIGMA are removed.
- Any $SCATTERPLOT record is removed.
- In $SIMULATION all random seeds are replaced.
- In $SIMULATION, option NSUBPROBLEMS is set to number of samples.

NPC and VPC with NONMEM7

NPC and VPC are designed to never perform estimation. With NONMEM5 and NONMEM6 the estimation is easily skipped by setting MAXEVAL=0. NONMEM7 however, can have multiple $ESTIMATIONs and/or estimation methods for which MAXEVAL do not apply. Settings in one $ESTIMATION will by default carry over to the next unless a new setting for the same option is set. This makes it much more complicated to automatically edit the modelfile to skip the estimation step and get correct output of PRED, DV etc.

There are two alternatives for the user when running vpc or npc with NM7. These are described in the document PsN_and_NONMEM7.pdf. In short, PsN will automatically construct a single $ESTIMATION from the original sequence, unless the user sets option -last_est_complete. The automatic construction will work best if the method of the last $ESTIMATION in the sequence is either classical, IMP or IMPMAP.

Table output processing

The dependent variable values in observation rows (see [2]) are extracted from original data and simulation table files. The values are saved to disc as a matrix in csv-format. One row per observation, original dataset in the first column and simulated datasets in columns 2-(n_samples+1). The values saved to disk are never prediction or variability corrected.

If option -censor is used, censor variable values in observation rows (see [2]) are extracted from original data and simulation table files. The values are saved to disc as a matrix in csv-format. One row per observation, original dataset in the first column and simulated datasets in columns 2-(n_samples+1).

Stratification for NPC/VPC

The data is stratified if -stratify_on is used. If the number of strata option is not used, stratification will be done on unique values of the stratification variable. If used, the unique values will be grouped into the requested number of strata so that there are approximately the same number of observations in each

strata, following the requirement that all stratification column values in strata *i* must be smaller than all values in strata *i+1*. A warning is issued if there are more than 10 strata. Stratification is done before binning when running VPC. Stratification is done before censoring, options -censor/-lloq/-uloq do not affect the stratification.

Binning for VPC

Binning is done on the independent variable, which is TIME by default but can be given on the command line. Binning is done before censoring, options -censor/-lloq/-uloq do not affect the binning. Only one independent variable can be given. There is a set of options for binning, and in some the user requests binning based on the number of observations in each bin. The actual number of observations put in a bin may be different from the number requested by the user if the request cannot be met without violating the rule that all observations with the same independent variable value are put in the same bin. User input counts will be treated as target counts, but the actual counts may be greater or smaller (but always greater than 0). Rounding will be done upwards or downwards, whichever gives the smallest adjustment, provided that count is never rounded down to 0. The actual counts are reported in the results. A partial vpc command-line to manually modify and reuse automatic bin edges can be found in the file vpc_bins.txt
There are several binning alternatives:

1) Fully automatic binning. (default, option -auto_bin = auto)
   Automatically finds the number of bins and partitions the dataset using an algorithm as proposed in a student project (see draft_vpc_automatic_binning.pdf) [1]. The algorithm is using ideas from [2] and [3]. This is the default binning option and it will currently assume that there will be no less than 5 and no more than 30 bins.

   [1] Christian Sonehag, Niklas Olofsson, Rasmus Simander, Automatic binning in visual predictive checks, Undergraduate project 2012
   [2] Calinski, R. B., & Harabasz, J. (1974). A dendrite method for cluster analysis. Communications in Statistics, 3, 1-27
   [3] M. Laveille, K. Bleakley, Automatic data binning for improved visual diagnosis of pharmacometric models, June 9, 2011
2) Automatic binning given a range for the number of bins. (option -auto_bin=min, max)
3) Automatic binning given a fixed number of bins (option -auto_bin=N)
4) bin based on unique values of the independent variable, no extra input needed. (option -auto_bin=unique)
5) bin based on a user-defined list of values of the independent variable, where the list values are the boundaries between the bins.

Option combination -bin_by_count=0 -bin_array=boundary1,boundary2,...

6) bin based on a user-defined list of counts of observations in each bin, ordered so that the first count is for the lowest values of the independent variable. The algorithm may adjust the counts up or down (see rule on binning above).

Option combination -bin_by_count=1 -bin_array=count1,count2,...

7) bin into X bins, X defined by user, of equal width based on independent variable.

Option combination -bin_by_count=0 -no_of_bins=X

8) bin into X bins, X defined by user, containing equal counts of observations. The counts will in general not be exactly equal because of binning rule.

Option combination -bin_by_count=1 -no_of_bins=X

9) use bin width X and bin overlap Y % between consecutive bins, X and Y defined by user.

Option combination -bin_by_count=0 -single_bin_size=X -overlap=Y

10) use bins with X observations and bin overlap Y % between consecutive bins, X and Y defined by user. X and Y are targets because of binning rule, not exact.

Option combination -bin_by_count=1 -single_bin_size=X -overlap=Y

11) use bin width X, X defined by user. X is adjusted by script so that all bins get equal size.

Option combination -bin_by_count=0 -single_bin_size=X

12) use bins with X observations, X defined by user. X adjusted by script to give all bins equal size.

    Option combination -bin_by_count=1 -single_bin_size=X

Log-transformation of data

If option -lnDV is set to 3, PRED is substituted with ln(PRED) and DV with ln(DV) in all subsequent steps.

VPC options predcorr and varcorr

Prediction correction and variability correction of the dependent variable values can be requested using options -predcorr and -varcorr. Variability correction can only be requested in combination with prediction correction. If both are requested then prediction correction is performed before variability correction. Correction is performed after stratification and binning. Correction will be performed on uncensored data, which means that simulated values which will be censored based on -censor/-lloq/-uloq in later analysis steps will still influence the variability correction, and PRED values for missing observations will influence the prediction correction.

Prediction correction

Let i be index for observations, i=1....n_observations <number of observations>
Let N be the number of simulations (input option samples).
Let k be index for simulations and observed data, k=1...N+1
Let LBi be the lower boundary for PREDi obtained from –lower_bound. If LBi was not specified it will be set to 0. If -lower_bound was given as number X then LBi=X, i=1...n_observations.

For each bin within each strata, compute the median value of PRED. Call this PREDnm, where “n” is strata number and “m” is bin number.

Case 1, lnDV=0 (default): Create a correction factor PCORRi for each observation as (PREDnm – LBi) divided with (PREDi – LBi). If (PREDi – LBi) is equal to or less than zero, stop execution and give an error message. The prediction corrected DV (PCDV) is calculated as

LBi + (DVik – LBi)*PCORRi, k=1...N+1.

I.e. if lnDV = 0 PCDVik = LBi + (DVik – LBi) * (PREDnm – LBi) / (PREDi – LBi)

Case 2, -lnDV=1,2 or 3: Create a correction factor PCORRi for each observation as (PREDnm-PREDi). The prediction corrected DV (PCDV) is calculated as DVik + PCORRi, k=1...N+1.

I.e. if ln DV = 1,2 or 3 PCDVik = DVik + (PREDnm – PREDi)

The prediction corrected values (PCDV) replace the observed and simulated DV values in all subsequent vpc procedures.

Variability correction
After simulation and prediction correction, calculate the standard deviation STDEVi for the N *simulated* PCDV (prediction corrected DV) values of each observation. For each bin within each strata, calculate the median value of STDEVi and call this STDEVnm for strata n and bin m. Create a correction factor VCORRi for each observation STDEVnm divided with STDEVi. If STDEVi is zero, a warning will be printed and then STDEVi is set to 100000.

Scale the deviation of PCDVik from PREDnm with VCORRi, i.e. replace PCDVik with VCORRi*(PCDVik-PREDnm)+PREDnm.

I.e. PVCDVik = PREDnm + (PCDVik – PREDnm) * (STDEVnm / STDEVi)

The variability and prediction corrected values (PVCDV) replace the observed and simulated DV values in all subsequent vpc procedures.

Exponentiation of data

If option -lnDV is set to 1, PRED is substituted with EXP(PRED) and DV with EXP(DV) in all subsequent steps.

Censoring of data

If option -censor is used, the missing observations are removed. If option -predcorr is used in combination with option -lloq or -uloq, the observations outside the boundaries of -lloq and -uloq are also removed.

Percentile and confidence interval calculations

In the analysis sections below, the Z% percentile of V values is the i:th value in a sorted list of the V values where the lowest value has index 0 (instead of 1) and i is computed as i= (Z/100)*(V-1) rounded to the nearest integer (numbers #.5 are rounded up). The exception is the 50% percentile which is computed directly as the median of the V values. If V is even the median is interpolated. The mean is computed directly as the arithmetic mean of the V values, and the delta-mean is the different between the arithmetic mean of the V values and the reference mean computed for the reference stratum. The CC% confidence interval (CI) “from” value is computed as the (100-CC)/2% percentile (2.5% percentile if confidence_interval is the default 95%). The CC% CI “to” value is the j:th value in the

sorted list of V values where j=V-i-1 and i is calculated as above.

Dependent variable matrix analysis for NPC

For each strata:{
For each prediction interval (PI) Y%: {
For each observation (each row in DV matrix){
The simulated DV values are sorted in ascending order. Then it is noted for each dataset, the real dataset as well as each simulated, if its DV value falls below the 50-(Y/2)% percentile (below the PI), above the 50+(Y/2)% percentile (above the PI) or inside the percentiles (inside the PI) of the sorted simulated observations. The number of observations above and below the PI are counted for each dataset separately.
}
Report the counts of real observations below and above the Y% PI. Sort the counts of simulated observations below/above the PI in ascending order. Report the CC% CI of the percentage of simulated observations below/above the PI, and mark with a * (warning) if the count of real observations below/above the PI fall outside the CI.
}
}

NPC results diagnostics

For each strata: compute confidence intervals for the number of * (warnings) for simulated data: Check each simulated dataset, as if it were the real data, to see whether the counts of datapoints below or above each PI falls outside the CC% CI of the number of points above/below. Assign a * (warning) to the dataset if the count is outside the CI, just as for the real dataset. Save the accumulated * counts in a vector with one element per simulated dataset. Repeat for all PI boundaries, i.e. 2*(no of PI:s) times. Report the mean, median and CC% CI for the number of * assigned to a single simulated dataset (mean, median and CC% CI of integer elements in the vector). Report the theoretical mean which would be obtained if the number of simulated datasets was very large. Compare the * statistics with the total count for the real dataset in the strata.

Dependent variable matrix analysis for VPC

Simulate data just as for NPC, and collect DV values in a matrix, one row for each observation and one column for each dataset. If the option -dv is used, the specified variable is collected instead of DV.

Stratify the data, according to Stratification for NPC/VPC.

Bin the observations according to the values of the independent variable. Principle as defined by input and above section Binning for VPC.

For each bin separately, perform steps a-g:
Let N be the number of samples (simulations), and let M be the number of observations in the bin.

(a) Compute the DV values representing the boundaries of the prediction intervals in the set. The value in the “Y% sim“ column is the Y% percentile of the uncensored (at most M*N) simulated values.

(b) Determine the PI boundaries for the real dataset. The value in the “Y% real“ column is the Y% percentile of the uncensored (at most M) real values.

(c) Determine a confidence interval for each PI boundary computed in (a) as follows:
For each simulated dataset separately, take the Y% percentile of the uncensored (at most M) values in the bin, just as for the real values described above in (b) (i.e. repeat N times to collect N values in total, exclude value if there are 0 uncensored values). Compute the CC% CI based on the at most N collected values.

(d) Generate data for mirror plots. For N randomly chosen simulated datasets, report the DV values representing the PI boundaries for the bin just as for the real dataset in (b). (Note: the data is extracted from the computations in (c), no extra analysis is required.)

(e) Perform diagnostics for continuous data according to section VPC results diagnostics. In case option -lloq or -uloq is used, only step (iii) of section VPC results diagnostics, a regular NPC, is performed.

(f) In the case of censored data, i.e. of any of the options -lloq, -uloq or -censor is used, some extra results are presented in a section “Censored data” in vpc_results.csv.
Left censored data: The value in the column “Real left censored” is the fraction of the non-missing (at most M) original dataset observations in the bin that is below lloq. The fraction of non-missing observations below lloq is also calculated for each simulated dataset separately, i.e. at most N times (exclude datasets where there are 0 non-missing values). The value in the “Simulated left censored” column is the median of the at most N calculated fractions of simulated observations. The values in the “CC% CI for left censored from/to” columns is the CC% CI based on the at most N values. If the option -uloq or -censor is used but not -lloq, then 'NA' is displayed in the columns for left censored data.
Right censored data: Analogous to Left censored data. Instead of the fraction of non-missing observations below lloq, the fraction above uloq is considered. If -lloq or -censor is used but not -uloq, then 'NA' is displayed.
Missing data (input option -censor): Analogous to Left censored data, but instead of the fraction of non-missing observations below lloq, the fraction of missing observations out of the total M is considered. If -lloq or -uloq is used but not -censor, then 'NA' is displayed.

When using -lloq and/or -uloq the regular results sections of vpc_results.csv are censored. Anywhere where the value in the “Y% real” column is below lloq or above uloq the value is replaced with 'NA'. In the “NPC results” section for the bin, columns “points below PI (count/%)” the values are replaced with 'NA' in row “X% PI” if the value in column “Z% sim”, where Z=50-(X/2), in the regular VPC section is below lloq. The values in columns “points above PI (count/%)” are replaced with 'NA' in row “X% PI” if the value in column “Q% sim”, where Q=50+(X/2), in the regular VPC section is below lloq.

(g) In the case of categorical data, i.e. when option -levels is used, results are presented in the section Categorical data. The value in the column “Real xx”, where xx defines the dependent value boundaries of the category, is the fraction of the at most M original dataset observations in the bin that falls within

these boundaries. The fraction of observations within the boundaries is also calculated for each simulated dataset separately, i.e. at most N times (exclude dataset where non-missing observations is 0). The value in the “Sim xx” column is the median of the at most N calculated fractions of simulated observations. The values in the “CC% CI for xx from/to” columns is the CC% CI based on the at most N values.

VPC results diagnostics

(i) For each bin, compute the percentage of false positives and false negatives for each PI as follows: For each observation in the bin, sort the non-missing simulated DV values in ascending order. Check if the real data observation lies inside or outside each PI interval (as for an NPC). The observation is a false positive if it lies within the PI interval of the sorted simulated values, but outside the VPC PI computed for the whole bin. The observation is a false negative if the real data observation lies outside the PI interval of the sorted simulated values but inside the VPC PI for the bin. Repeat the sorting and checking for all observations in the bin and report the percentage of false positives and false negatives in the bin for each PI.

(ii) Report percentages of false positives/negatives for each PI in the whole strata:
Sum the number of false positives and negatives over all bins, and divide by the sum of non-missing observations in the bins. Note: This corresponds to computing a weighted average of the classification for observations that belong to more than one bin in the case of bin overlap. Then the sum of bin observations may be larger than the number of observations in the strata, and a single observation may, for example, be a false positive in one bin and a true negative in another.

(iii) Output a regular NPC for each bin, using the same PI as for the VPC.

Additional rules & logic

[1] Columns requested in $TABLE record, which is added by VPC both for original and simulated data:

- ID
- EVID, only if option -tte is set and EVID is present in $INPUT
- MDV (see [2]), but not if option -tte is set
- DV
- any other variable requested using option -dv, unless it is CWRES and -orig_table is not used
- independent variable column to bin on (given on command-line input).
- column to stratify on (given on command-line input), if any.
- Censoring variable, if option -censor is used.
- TTE variable, if option -tte is used and TTE variable is not the same as the independent variable which has already been added. The TTE variable is only added in the simulation model, not the model for original data.
- options NOPRINT, ONEHEADER

[2] Logic for identifying observation rows in table-file:
If there is no $PRED record
    then request MDV in $TABLE. Row is an observation if and only if MDV column value is 0.
Otherwise (there is a $PRED record)
    If (MDV is in $INPUT) AND (NOT MDV=DROP/SKIP)

request MDV in $TABLE. Row is an observation if and only if MDV column value is 0.

Otherwise (no MDV kept in $INPUT)

Do not request MDV in $TABLE. All rows are observations.

[3] Logic for whether older program versions would automatically set keep_estimation, i.e. whether $ESTIMATION should be kept in the simulations:

- keep_estimation was not set by default.
- if the -dv option is used, keep_estimation was set regardless of the variable requested with -dv.
- If -idv is PRED, keep_estimation was set.
- If -stratify_on is PRED, keep_estimation was set.

Known limitations and problems

The program cannot detect nor handle missing values in the table output (variables requested in the generated $TABLE record) by any other method than option -censor. Missing values that are not dealt with using option -censor will lead to erroneous output.

It will not work to read initial parameter estimates from a separate file using the CHAIN command. Instead, the user can either create an msfo-file and use option -msfo or use the -orig_table and -sim_table options.