Paper SAS206-2014

Simulating Portfolio Losses from Adverse Events: Applications to the Insurance and Finance Industries

Mahesh V. Joshi and Jan Chvosta, SAS Institute Inc.

ABSTRACT

Companies in the insurance and banking industries need to model the frequency and severity of adverse events every day. Accurate modeling of risks and the application of predictive methods ensure the liquidity and financial health of portfolios. Often, the modeling involves computationally intensive, large-scale simulation. SAS/ETS® provides high-performance procedures to assist in this modeling. This paper discusses the capabilities of the HPCOUNTREG and HPSEVERITY procedures, which estimate count and loss distribution models in a massively parallel processing environment. The loss modeling features have been extended by the new HPCDM procedure, which simulates the probability distribution of the aggregate loss by compounding the count and severity distribution models. PROC HPCDM also analyzes the impact of various future scenarios and parameter uncertainty on the distribution of the aggregate loss. This paper steps through the entire modeling and simulation process that is useful in the insurance and banking industries.

INTRODUCTION

Financial losses due to adverse events are an unfortunate yet inescapable fact of life for many businesses, especially in the financial industry. An insurance company, by the very nature of its business, needs to deal with the losses incurred by its policyholders. Banks and other financial institutions need to operate in a market economy, where investment losses are an inherent part of the business. These institutions also encounter operational losses such as theft and fraud because of socioeconomic and human behavioral issues. It is important not only to identify the factors that cause the losses but also to quantify the expected losses in order to manage risk better and to estimate the risk-based capital requirements that are demanded by regulations such as Basel III (banking industry) and Solvency II (insurance industry). Most modern businesses collect and record information about losses. Such information often includes the number of loss events that were encountered in a given period of time, the magnitude of each loss, the characteristics of the entity that incurred the loss, and the characteristics of the economic environment in which the loss occurred. Because data about past losses are more readily available, quantitative modeling of losses is becoming an increasingly important task for many businesses. The goal is to estimate the average loss as well as the worst-case loss that you expect to observe in a particular period of time, not only for one entity but also for a group of entities such as a group of policyholders, a group of financial assets, or even a group of operating environments. Several mathematical and statistical approaches are possible, but one of the most commonly used and desirable approaches is to estimate separate loss distribution models for the frequency (number) of loss events and severity (magnitude) of each loss, and then to combine those models to estimate the aggregate loss in a given time period.

A loss distribution approach (LDA) postulates that the number of losses and the severity of each loss are random variables and builds a probability distribution model for each variable. One of the most common and useful LDA methods is the parametric method, which postulates that the probability distributions are from parametric families of distributions, estimates the parameters of some candidate families by using the observed data, and chooses the parametric family that is deemed the best statistical fit for the data. An alternative is the nonparametric method, which computes empirical estimates of various moments and quantiles of the distribution from the observed sample. For the purpose of having a predictive, parsimonious, and flexible model that enables what-if analysis, the parametric approach is preferred to the nonparametric approach (Klugman, Panjer, and Willmot 1998). What-if analysis is especially important when loss counts and loss severity are affected by exogenous factors, which is usually the case. For example, the number of automobile insurance claims that a policyholder files might depend on the type and age of the vehicle, and the severity of each claim might depend on the safety rating and value of the vehicle. Both the number of claims and the severity of each claim might also depend on the characteristics of the policyholder. As

another example, the number of fraudulent transactions, which are one of the key components of operational risk at a bank, might depend on the security mechanisms implemented by the bank as well as the current economic environment. It is important for a loss distribution model to account for the effects of such external factors, known as regression effects, on the probability distributions of count and severity. After you model the external effects, you can conduct a what-if or scenario analysis by analyzing the distributions for a given set of values for the exogenous factors.

SAS/ETS includes two procedures, PROC COUNTREG and PROC SEVERITY, that enable you to find and estimate the best probability distribution for count and severity distributions, respectively, while modeling the regression effects. This paper illustrates the key features of these procedures, including some new features such as scoring functions in PROC SEVERITY and the STORE statement in PROC COUNTREG, that are introduced in SAS/ETS 13.1.

Although the loss distribution models for frequency and severity are useful on their own, they are often a means to the end goal of estimating the risk measures, which requires estimation of the probability distribution of the *aggregate loss* that a business expects to incur in a given time period. Some common risk measures are defined in terms of a percentile of the aggregate loss distribution or the mean of an extreme tail region of the distribution. For example, the value at risk (VaR) at a level α $(0 < \alpha < 1)$ is defined as the value such that the probability of a loss that exceeds the VaR is $(1 - \alpha)$, and the tail value at risk (TVaR) at a level α is defined as the expected value of all the losses that exceed the VaR at level α. The aggregate loss S in a given time period is defined as

$$S = \sum_{j=1}^{N} X_j$$

where N represents the frequency random variable for the number of loss events and X represents the severity random variable for the magnitude of one loss event. The goal is to estimate the probability distribution of S. Let $F_X(x)$ denote the cumulative distribution function (CDF) of X; let $F_X^{*n}(x)$ denote the n-fold convolution of the CDF of X; and let $\Pr(N = n)$ denote the probability of seeing n losses as per the frequency distribution. The CDF of S is theoretically computable as

$$F_S(s) = \sum_{n=0}^{\infty} \Pr(N = n) \cdot F_X^{*n}(x)$$

The probability distribution model of S, characterized by the CDF $F_S(s)$, is referred to as a *compound distribution model* (CDM). Direct computation of F_S is usually a difficult task because of the need to compute the n-fold convolution. An alternative is to use Monte Carlo simulation to generate a sufficiently large, representative sample of the compound distribution. In addition to its simplicity, the simulation method applies to any combination of distributions of N and X, unlike some alternatives such as the recursion or inversion methods that are suggested in Klugman, Panjer, and Willmot (1998, Ch. 4). This paper describes the new HPCDM procedure, first released in SAS High-Performance Econometrics 13.1, that estimates F_S by using the simulation method. Through examples that are inspired by real-world problems in the insurance and banking industries, the paper illustrates several key features of the HPCDM procedure that make it useful for estimating risk measures of a portfolio.

As with any quantitative modeling task, the more data the better. For many years, inadequacy of data was an issue. However, for many businesses, the situation is quickly reversing. The modernization of data collection and storage infrastructure has resulted in large amounts of data being collected. The challenge now is how to analyze such large amounts of data to quantify the nature of losses. It is desirable to use all the data, after appropriately addressing any data quality issues, as compared to working on only a smaller sample of the data, because the sampling process might ignore some important features of the data. However, you need an appropriate data management framework and analytical tools that work within that framework to be able to analyze large amounts of data. SAS High-Performance Econometrics includes high-performance versions of the SEVERITY and COUNTREG procedures, named PROC HPSEVERITY and PROC HPCOUNTREG, respectively, that enable you to analyze large amounts of data by using a grid of computers that work together to estimate the severity and count models in a very short amount of time. The HPCDM procedure is also designed to exploit the grid of computers to estimate the aggregate loss distribution, as well as to conduct scenario analysis and parameter perturbation analysis in a much shorter time. This paper describes the high-performance features of all these procedures.

IT ALL STARTS HERE: DATA

Collection and storage of historical loss data are the crucial prerequisite in any loss modeling solution. You should ideally collect and store the loss data at the level of the loss event. At the very minimum, you need to record the time of occurrence and severity of the loss event. The data are rarely available in a form that is readily usable for building loss models. You often need to prepare the data by transforming and combining various internal and external data sources. Depending on the source of the data, you might not be able to observe all loss events or you might not know the exact severity of each loss event. Such limitations are known as truncation and censoring effects, respectively. Truncation and censoring occur mostly by design, and you usually know the provisions in the policy or the data collection process that cause these effects. For example, for an insurance policy, the deductible results in truncation and the maximum payment limit results in censoring. For each loss event, you should record the truncation thresholds and censoring range if you know that the loss event is generated by an entity that is subject to such provisions. The SEVERITY procedure accounts for truncation and censoring effects and builds the model for *ground-up loss*.

It is recommended that you augment the loss event data to record all the relevant characteristics of the entity that generates the loss event and the environment in which the entity operates. Such characteristics act as regressors in the loss models. In the financial industry, they are known as key risk indicators (KRIs). It is a critical exercise to identify, define, and collect the data about KRIs for each line of business that you want to build loss models for. After the regressors are identified and data are collected for them, as part of the data preparation step it is recommended that you properly code, scale, or standardize their values.

This paper illustrates the loss modeling process by using the following two simple, simulated input data sets that are inspired by the insurance and banking industries, respectively:

- **Example Insurance Data**
  Consider that you are an automobile insurance company, and you have collected data about the insurance claims that your policyholders have filed. Each policy has a deductible and a maximum payment limit. Further, consider that you have postulated that the distribution of the loss event frequency depends on five regressors (external factors): age and gender of the policyholder, type of car, annual miles driven, and policyholder's education level. Further, the distribution of the severity of each loss depends on three regressors: type of car, safety rating of the car, and annual household income of the policyholder (which can be thought of as a proxy for the luxury level of the car). Note that the frequency model regressors and severity model regressors can be different, as illustrated in this example.

  Let the deductible for each policyholder be recorded in the variable **Deductible**. Let the variable **Limit** record the upper limit on the observed ground-up loss (the maximum payment limit plus the deductible). Let the regressors be recorded in the variables **Age** (scaled by a factor of 1/50), **Gender** (1: female, 2: male), **CarType** (1: sedan, 2: sport utility vehicle), **AnnualMiles** (scaled by a factor of 1/5,000), **Education** (1: high school graduate, 2: college graduate, 3: advanced degree holder), **CarSafety** (scaled to be between 0 and 1, the safest being 1), and **Income** (scaled by a factor of 1/100,000), respectively. Let the period of analysis be one year, and let the following two data sets be prepared from the historical claims data:

  – The Work.AutoLossCounts data set contains the frequency regressors and the variable **NumLoss**, which records the number of losses that a policyholder incurs in a year.
  – The Work.AutoLosses data set contains the severity regressors and the variable **LossAmount**, which records the severity of each loss.

  The simulated input data contain information about 5,000 policyholders, who together generate approximately 9,000 loss events per year, with a mean loss of around 900 units.

- **Example Operational Risk Data**
  Consider that you are a bank, and as part of quantifying your operational risk, you want to estimate the aggregate loss distributions for two lines of business (LOB), retail banking and commercial banking, by using key risk indicators (KRIs) that are appropriate to each LOB. Let **CorpKRI1**, **CorpKRI2**, **CbKRI1**, **CbKRI2**, and **CbKRI3** be the KRIs that are used in the count regression model of the commercial banking business, and let **CorpKRI1**, **RbKRI1**, and **RbKRI2** be the KRIs that are used in the count regression model of the retail banking business. Some examples of corporate-level KRIs (**CorpKRI1** and **CorpKRI2** in this example) are the ratio of temporary to permanent employees and the number

of security breaches that are reported during a year. Some examples of KRIs that are specific to the commercial banking business (**CbKRI1**, **CbKRI2**, and **CbKRI3** in this example) are number of credit defaults, proportion of financed assets that are movable, and penalty claims against your bank because of processing delays. Some examples of KRIs that are specific to the retail banking business (**RbKRI1** and **RbKRI2** in this example) are number of credit cards that are reported stolen, fraction of employees who have not undergone fraud detection training, and number of forged drafts and checks that are presented in a year.

Let the severity of each loss event in the commercial banking business be dependent on two KRIs, **CorpKRI1** and **CbKRI2**. Let the severity of each loss event in the retail banking business be dependent on three KRIs, **CorpKRI2**, **RbKRI1**, and **RbKRI3**. Note that for each line of business, the set of KRIs that are used for the severity model is different from the set of KRIs that are used for the count model, although the two sets do overlap. Further, the severity model for retail banking includes a new regressor (**RbKRI3**) that is not used for any of the count models. Such use of different sets of KRIs for count and severity models is typical of real-world applications.

Let the analysis period be one month, and let the data preparation process result in the following two data sets:

- The Work.OpRiskLossCounts data set contains the frequency regressors and the variable **NumLoss**, which records the number of losses that a LOB incurs in a month.
- The Work.OpRiskLosses data set contains the severity regressors and the variable **LossValue**, which records the severity of each loss.

The simulated input data contain 15 years of monthly data, with 133 and 119 loss events for the commercial and retail banking businesses, respectively.

For each of the preceding examples, a separate *scoring* data set is also simulated. For the insurance example, the scoring data set is named Work.GroupOfPolicies, and it contains observations for multiple policyholders. For the operational risk example, the scoring data set is named Work.MultiConditionScenario, and it contains observations for 12 months of a future year. Each observation in a scoring data set contains values that you expect to observe for each regressor in the frequency and severity models. These data sets form the basis of the what-if or scenario analysis process that *predicts* the frequency, severity, and aggregate losses for a given state of the future. The code that generates all example data sets is available at `http://support.sas.com/saspresents`.

HOW MANY AND HOW SEVERE? FREQUENCY AND SEVERITY MODELING

The data have been prepared, so now you are ready to estimate frequency and severity models. SAS/ETS offers two procedures that help you with this task. The COUNTREG procedure estimates the parameters of a discrete distribution model such that the mean of the distribution is conditional on the regression effects. The SEVERITY procedure estimates the parameters of a continuous distribution model such that the scale of the distribution is conditional on the regression effects. This section illustrates some key features of these two procedures, especially the new scoring features. The output that these procedures generate is used later by the HPCDM procedure to simulate the aggregate loss distribution.

FITTING AND SCORING COUNT MODELS

Given the loss count data in the Work.AutoLossCounts data set from the insurance example, the following PROC COUNTREG step fits a negative binomial regression model by using the five regressors that are specified in the MODEL statement, three of which are classification variables as specified in the CLASS statement:

```
proc countreg data=autolosscounts;
   class gender carType education;
   model numloss = age gender carType annualMiles education / dist=negbin;
   store work.autocountmodel;
run;
```

The model information and parameter estimates of the fitted model are stored in the Work.AutoCountModel item store that you specify in the MODEL statement. This item store is required later by the HPCDM procedure to simulate counts. You can examine the contents of the item store, in particular the model fit summary and parameter estimates, by submitting the following statements:

```
proc countreg restore=work.autocountmodel;
   show fitstats parameters;
run;
```

The "Model Fit Summary" and "Parameter Estimates" tables that are displayed by the SHOW statement are shown in Figure 1. The values of the fit statistics Akaike's information criterion (AIC) and Schwarz Bayesian criterion (SBC) indicate how well the distribution fits the data.

Figure 1 Fit Summary and Parameter Estimates of the Count Regression Model

ITEM STORE CONTENTS: WORK.AUTOCOUNTMODEL

Model Fit Summary	
Dependent Variable	numloss
Number of Observations	5000
Data Set	WORK.AUTOLOSSCOUNTS
Item Store	WORK.AUTOCOUNTMODEL
Model	NegBin(p=2)
Log Likelihood	-7136
Maximum Absolute Gradient	1.15644E-7
Number of Iterations	6
Optimization Method	Newton-Raphson
AIC	14295
SBC	14367
Number of Threads	4

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	3.464267	0.070931	48.84	<.0001
age	1	-0.763052	0.058065	-13.14	<.0001
gender 1	1	-0.966338	0.031837	-30.35	<.0001
gender 2	0	0	.	.	.
carType 1	1	-0.629246	0.031345	-20.07	<.0001
carType 2	0	0	.	.	.
annualMiles	1	-0.982010	0.017240	-56.96	<.0001
education 1	1	0.495307	0.047774	10.37	<.0001
education 2	1	0.227570	0.050966	4.47	<.0001
education 3	0	0	.	.	.
_Alpha	1	0.331917	0.020354	16.31	<.0001

You can compare the AIC or SBC values of different distributions that you fit, either by specifying different MODEL statements in one PROC COUNTREG step or by submitting different PROC COUNTREG steps, to decide which distribution has the best fit for the input count data. However, note that for the current release of PROC COUNTREG (in SAS/ETS 13.1), the item store can contain only one model, so if you want to create the item store for later consumption by PROC HPCDM, then you must specify only one MODEL statement in the PROC COUNTREG step. In that case, it is recommended that you use different PROC COUNTREG steps to choose the best distribution.

PROC COUNTREG supports the following models for count data:

- Poisson regression and zero-inflated Poisson (ZIP) model (Lambert 1992)
- negative binomial regression with quadratic (NEGBIN2) and linear (NEGBIN1) variance functions (Cameron and Trivedi 1986)
- zero-inflated negative binomial (ZINB) model
- Conway-Maxwell-Poisson regression and zero-inflated Conway-Maxwell-Poisson (ZICMP) model
- fixed-effects and random-effects Poisson models for panel data
- fixed-effects and random-effects negative binomial models for panel data

For each distribution, you ideally want to choose the best set of regressors to include in the model. PROC COUNTREG makes this easy for you by offering the automatic variable selection feature. You just need to specify an appropriate value for the SELECT= option in the MODEL statement. For more information, see the description of the MODEL statement in the COUNTREG procedure chapter of the *SAS/ETS User's Guide*.

After you select the best count regression model, which is defined by the combination of the distribution and the set of regressors for that distribution, you can use the model to predict the mean count value and probabilities of observing different count values for a given set of regressor values. This prediction process is referred to as *scoring*. The following PROC COUNTREG step scores a set of insurance policies that are stored in the Work.GroupOfPolicies data set by predicting the mean number of losses that each policyholder might incur and the probability that each policyholder incurs exactly 0, 1, 5, and 10 losses:

```
proc countreg restore=work.autocountmodel;
   score data=groupOfPolicies out=outScores mean=meanNB2 probcount(0 1 5 10);
run;
```

The prediction results are shown in Figure 2. Of the five policyholders shown, you can expect the first policyholder to incur the fewest losses, whereas you can expect the fifth policyholder to incur a significantly large number of losses. You can explain such differences by looking at the policyholder's characteristics and the regression parameter estimates.

Figure 2 Count Predictions for Policyholders

policyholderId	age	gender	carType	annualMiles	education	meanNB2	P_0	P_1	P_5	P_10
1	1.18	2	1	2.2948	3	0.72695	0.52141	0.30536	0.003082	.000002705
2	0.66	2	2	2.8148	1	1.99730	0.21604	0.25948	0.046325	.001475571
3	0.82	1	2	1.6130	2	1.67500	0.26396	0.28416	0.032739	.000603184
4	0.44	1	1	1.2280	3	1.38685	0.31956	0.30348	0.021178	.000208498
5	0.44	1	1	0.9670	2	2.24995	0.18628	0.23994	0.056657	.002559851

FITTING AND SCORING SEVERITY MODELS

PROC SEVERITY enables you to fit several distribution models in one step and identifies the best-fitting distribution according to each of the fit statistics. PROC SEVERITY ships with a default set of 10 distributions: Burr, exponential, gamma, generalized Pareto, inverse Gaussian (Wald), lognormal, Pareto, scaled Tweedie, Tweedie, and Weibull. It also enables you to define your own distribution by using the FCMP procedure in Base SAS®. For example, you can fit a mixture of two lognormal distributions such that the mixture distribution has the following probability density function (PDF) and CDF,

$$\text{PDF: } f(x;\mu,\sigma_1,p_2,\rho_2,\sigma_2) = \frac{(1-p_2)}{\sigma_1 x\sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{\log(x)-\mu}{\sigma_1}\right)^2} + \frac{p_2}{\sigma_2 x\sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{\log(x)-\mu-\log(\rho_2)}{\sigma_2}\right)^2}$$

$$\text{CDF: } F(x;\mu,\sigma_1,p_2,\rho_2,\sigma_2) = (1-p_2)\Phi\left(\frac{\log(x)-\mu}{\sigma_1}\right) + p_2\Phi\left(\frac{\log(x)-\mu-\log(\rho_2)}{\sigma_2}\right)$$

where p_2 is the mixing probability, μ is the log of the scale parameter of the entire distribution, ρ_2 is the ratio of the scale parameters of the two components, σ_1 and σ_2 are the shape parameters of the two components, and Φ is the standard normal CDF. The following PROC FCMP step encodes this formulation in a distribution named SLOGNMIX2, which follows the rules that are mandated by PROC SEVERITY:

```
proc fcmp library=sashelp.svrtdist outlib=work.mydist.models;
   function slognmix2_description() $128;
      return ("Mixture of Two Lognormals with a Log-Scale Parameter");
   endsub;
   function slognmix2_scaletransform() $8;
      return ("LOG");
   endsub;
   function slognmix2_pdf(x, Mu, Sigma1, p2, Rho2, Sigma2);
      Mu1 = Mu;                          Mu2 = Mu + log(Rho2);
      pdf1 = logn_pdf(x, Mu1, Sigma1); pdf2 = logn_pdf(x, Mu2, Sigma2);
      return ((1-p2)*pdf1 + p2*pdf2);
   endsub;
   function slognmix2_cdf(x, Mu, Sigma1, p2, Rho2, Sigma2);
      Mu1 = Mu;                          Mu2 = Mu + log(Rho2);
      cdf1 = logn_cdf(x, Mu1, Sigma1); cdf2 = logn_cdf(x, Mu2, Sigma2);
      return ((1-p2)*cdf1 + p2*cdf2);
   endsub;
   subroutine slognmix2_parminit(dim, x[*], nx[*], F[*], Ftype,
                                 Mu, Sigma1, p2, Rho2, Sigma2);
      outargs Mu, Sigma1, p2, Rho2, Sigma2;
      array m[1] / nosymbols;
      p2 = 0.5; Rho2 = 0.5;
      median = svrtutil_percentile(0.5, dim, x, F, Ftype);
      Mu = log(2*median/1.5);
      call svrtutil_rawmoments(dim, x, nx, 1, m);
      lm1 = log(m[1]);
      * Search Rho2 that makes log(sample mean) > Mu *;
      do while (lm1 <= Mu and Rho2 < 1);
         Rho2 = Rho2 + 0.01;
         Mu = log(2*median/(1+Rho2));
      end;
      if (Rho2 >= 1) then Mu = log(2*median/1.5);
      Sigma1 = sqrt(2.0*(log(m[1])-Mu));
      Sigma2 = sqrt(2.0*(log(m[1])-Mu-log(Rho2)));
   endsub;
   subroutine slognmix2_lowerbounds(Mu, Sigma1, p2, Rho2, Sigma2);
      outargs Mu, Sigma1, p2, Rho2, Sigma2;
      Mu = .; p2 = 0; Rho2 = 0; Sigma1 = 0; Sigma2 = 0; * missing value=>unbounded;
   endsub;
   subroutine slognmix2_upperbounds(Mu, Sigma1, p2, Rho2, Sigma2);
      outargs Mu, Sigma1, p2, Rho2, Sigma2;
      Mu = .; Sigma1 = .; Sigma2 = .; p2 = 1; Rho2 = 1; * missing value=>unbounded;
   endsub;
quit;
```

Some key points to note regarding the preceding definition are as follows:

- The PROC FCMP step stores all the functions and subroutines in the Work.MyDist library that appears in the OUTLIB= option. The name and signature (number and type of arguments) of each function or subroutine follow a format that PROC SEVERITY prescribes. For example, when PROC SEVERITY finds the SLOGNMIX2_PDF function, it identifies the suffix keyword PDF and deduces that the distribution is named SLOGNMIX2 (the part before the underscore). It further parses the signature of the function to identify that the distribution has five parameters, named Mu, Sigma1, p2, Rho2, and Sigma2, and confirms that those exact names of parameters appear in the argument lists of all other functions and subroutines. PROC SEVERITY has some reserved keyword suffixes, and each one is associated with a fixed semantic. For example, when it sees a subroutine that has the suffix PARMINIT, it expects that subroutine to return the initial values of parameters. A detailed list of keywords and their semantics is provided in the SEVERITY procedure chapter of the *SAS/ETS User's Guide*.

- It is strongly recommended that you provide your own parameter initialization subroutine whenever you define your own distribution. This particular example implements a simple initialization that sets the initial values of p_2 and ρ_2 to 0.5, and it makes two simplifying assumptions: that the median of the mixture is the average of the medians of the two components and that each component has the same mean. A combination of percentile matching and moment matching methods is then used to set the initial values of μ, σ_1, and σ_2. Despite these simplifying assumptions, these initial values work well when the mixture is indeed a good fit for the data. However, in general, parameter initialization can be an important task if you want to get good convergence behavior from the nonlinear optimizer.
- The mixture distribution is formulated to have an overall log-scale parameter μ, because PROC SEVERITY uses a *scale regression model*. In particular, if θ denotes the scale parameter of a distribution, then the regression effects are modeled as $\theta = \theta_0 \cdot \exp(\sum_{j=1}^{k} \beta_j x_j)$, where θ_0 is the *base* value of the scale parameter. If a distribution has a log-scale parameter, such as μ in this example, then the scale regression model becomes the log-linear scale regression model: $\mu = \log(\theta_0) + \sum_{j=1}^{k} \beta_j x_j$. This method of modeling regression effects has an advantage when you are accounting for inflationary changes. You can simply apply the effect of inflation to the base value of the scale parameter without having to refit the model. Further, given the linear relationship between the scale parameter and the mean of the distribution, the regression parameter estimates, β_j, are identical to those that you would obtain by fitting a generalized linear model that uses a log-link function.

You can define two additional functions, SLOGNMIX2_MEAN and SLOGNMIX2_QUANTILE, that compute the mean and quantile of the mixture distribution, respectively:

```
proc fcmp library=(sashelp.svrtdist work.mydist) outlib=work.mydist.meanquantile;
   function slognmix2_mean(x, Mu, Sigma1, p2, Rho2, Sigma2);
      Mu1 = Mu;                          Mu2 = Mu + log(Rho2);
      mean1 = exp(Mu1 + Sigma1**2/2); mean2 = exp(Mu2 + Sigma2**2/2);
      return ((1-p2)*mean1 + p2*mean2);
   endsub;
   function slognmix2_quantile(cdf, Mu, Sigma1, p2, Rho2, Sigma2);
      array opts[4] / nosym (0.1 1.0e-8 0);
      q = invcdf("slognmix2_cdf", opts, cdf, Mu, Sigma1, p2, Rho2, Sigma2);
      if (opts[4] > 0) then return(.); * opts[4] (status) > 0 means error;
      return(q);
   endsub;
quit;
```

Definitions of these functions are not mandated by PROC SEVERITY, but they are useful for prediction purposes. PROC SEVERITY detects them as *score-generating functions* because their signature matches the signature of the SLOGNMIX2_PDF function. When you specify the OUTSCORELIB statement, for each score-generating function, which includes the mandated functions such as PDF and CDF, PROC SEVERITY generates scoring functions that you can use to score new observations, as is illustrated later. Note that the definition of the SLOGNMIX2_QUANTILE function uses a generic CDF inversion function, INVCDF, that is available in PROC FCMP. Also, the SLOGNMIX2_PARMINIT definition uses two utility functions, SVRTUTIL_PERCENTILE and SVRTUTIL_RAWMOMENTS. More such utility functions are available for your convenience; see the documentation of the FCMP procedure (*Base SAS Procedures Guide*) and the SEVERITY procedure (*SAS/ETS User's Guide*).

Now it is time to find the best-fitting model for the data in the insurance example. The following statements first specify the library that contains the SLOGNMIX2 distribution by using the CMPLIB system option and then submit the PROC SEVERITY step to fit the SLOGNMIX2 distribution along with the predefined distributions to the severity values in the Work.AutoLosses data set:

```
options cmplib=(work.mydist);

proc severity data=autolosses plots=none outest=work.sevregestauto print=all;
   loss lossamount / lt=deductible rc=limit;
   scalemodel carType carSafety income;
   dist slognmix2 _predefined_;
   outscorelib outlib=scorefuncs commonpackage;
run;
```

The DIST statement specifies the distributions to be fitted. Note that using the _PREDEFINED_ keyword is equivalent to specifying 8 of the 10 predefined distributions (all except the two Tweedie variants). The LOSS statement specifies the response variable **LossAmount** and specifies the incomplete and inexact nature of the severity values by using **Deductible** as the left-truncation variable and **Limit** as the right-censoring variable. The SCALEMODEL statement specifies the regressors to be included in the scale regression model. The OUTSCORELIB statement creates the Work.ScoreFuncs FCMP library, which contains the scoring functions of all distributions that do not fail to converge.

The comparison of fit statistics of various scale regression models is shown in Figure 3. The scale regression model that is based on the two-component lognormal mixture distribution is deemed the best-fitting model according to most likelihood-based statistics. However, the BIC statistic penalizes the two-component mixture for having a larger number of parameters and prefers the more parsimonious lognormal model. Further, the other class of fit statistics—Kolmogorov-Smirnov statistic (KS), Anderson-Darling statistic (AD), and Cramér–von Mises statistic (CvM)—that compare the nonparametric and parametric estimates of the CDF deem the lognormal model to be better than the SLOGNMIX2 model. This analysis suggests that one lognormal component is enough to capture the severity distribution of this particular input data. It is also worth considering the scale regression model that is based on the generalized Pareto distribution (GPD), because the CvM and AD statistics deem it to be the best-fitting model. Note that the Pareto distribution is a reparameterization of the GPD when the shape parameter of the GPD is nonzero.

Figure 3 Severity Model Comparison and Parameter Estimates of the Lognormal Model

The SEVERITY Procedure

All Fit Statistics

Distribution	-2 Log Likelihood		AIC		AICC		BIC		KS		AD		CvM	
slognmix2	125287	*	125303	*	125303	*	125360		10.13574		196.11712		31.65281	
Burr	125300		125312		125312		125355		10.63961		209.84224		35.29704	
Exp	125616		125624		125624		125652		5.53160		95.00960		10.51370	
Gamma	125612		125622		125622		125657		6.70262		117.71474		15.62045	
Igauss	125364		125374		125374		125409		7.61154		126.57130		17.33231	
Logn	125300		125310		125310		125346	*	9.87311		195.22165		30.10706	
Pareto	125508		125518		125518		125553		4.06655		61.96093		3.10688	*
Gpd	125508		125518		125518		125553		4.06653		61.96062	*	3.10688	
Weibull	125604		125614		125614		125650		3.79804	*	72.38249		5.20399	

Note: The asterisk (*) marks the best model according to each column's criterion.

Parameter Estimates

Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|
Mu	3.98156	0.05398	73.76	<.0001
Sigma	0.73819	0.00844	87.46	<.0001
carType	1.43426	0.02582	55.54	<.0001
carSafety	-0.73577	0.03857	-19.08	<.0001
income	0.56534	0.02505	22.57	<.0001

Figure 3 also shows the parameter estimates of the lognormal scale regression model. You can use these parameter estimates to conclude that the severity of losses that a particular policyholder incurs follows a lognormal distribution with parameters μ and $\sigma = 0.73819$, where μ depends on the characteristics of the policyholder. For a policyholder who has income i and who drives a car of type t with safety rating s, you can compute μ as $\mu = 3.98156 + 1.43426 \times t - 0.73577 \times s + 0.56534 \times i$. For a distribution such as the GPD that has a direct scale parameter (and not a log-scale parameter like lognormal), you can identify the distribution for a specific policyholder by computing the scale parameter θ as $\theta = \theta_0 \times \exp(\beta_{\text{carType}} \times t + \beta_{\text{carSafety}} \times s + \beta_{\text{income}} \times i)$, where θ_0 is the estimate of the scale parameter in the "Parameter Estimates" table.

If you want to predict the mean and various quantiles from the distributions of individual policyholders, then you can use the scoring functions such as SEV_MEAN_LOGN and SEV_QUANTILE_GPD that PROC SEVERITY creates when you specify the OUTSCORELIB statement. Upon successful completion, the preceding PROC SEVERITY step adds the Work.ScoreFuncs FCMP library to the value of the CMPLIB= system option, so you can immediately submit the following DATA step to predict the average and worst-case losses (97.5th percentile) for the policyholders in the GroupOfPolicies data set:

```
%let worstLossCDF = 0.975;
data severityScores;
   array sx{3} carType carSafety income;
   set groupOfPolicies;
   logn_mean  = sev_mean_logn(., sx);
   gpd_mean   = sev_mean_gpd(., sx);
   logn_worst = sev_quantile_logn(&worstLossCDF, sx);
   gpd_worst  = sev_quantile_gpd(&worstLossCDF, sx);
run;
```

The first five observations of the Work.SeverityScores data set are shown in Figure 4. Note how the characteristics of the policyholder and the vehicle that he or she drives can cause significant variations in the average and worst-case losses that you can expect this policyholder to incur.

Figure 4 Predicted Average and Worst-Case Severity for Various Policyholders

policyholderId	carType	carSafety	income	logn_mean	gpd_mean	logn_worst	gpd_worst
1	1	0.99532	1.59870	350.65	255.62	1134.72	1009.12
2	2	0.05625	0.67539	1742.34	1538.28	5638.37	6072.83
3	2	0.84146	1.05940	1214.82	998.90	3931.27	3943.44
4	1	0.14324	0.24110	304.66	223.78	985.91	883.45
5	1	0.08656	0.65979	402.46	307.12	1302.40	1212.46

In addition to the features that are described and illustrated in the preceding discussion, PROC SEVERITY enables you to conduct more severity distribution modeling tasks, such as estimating parameters by minimizing a custom objective function; modeling with interval-censored and right-truncated data; computing nonparametric estimators of CDF such as the Kaplan-Meier estimator (Kaplan and Meier 1958, Lynden-Bell 1971), the modified Kaplan-Meier estimator (Lai and Ying 1991), and Turnbull's estimator (Turnbull 1976, Gentleman and Geyer 1994); and producing various diagnostic plots, such as the P-P plot, the Q-Q plot, and a CDF comparison plot that displays confidence intervals around the nonparametric CDF estimates.

ADDING IT UP: AGGREGATE LOSS MODELING

After you estimate the frequency and severity models and select the best models as described in the previous section, the final step in the loss modeling process is to estimate the compound distribution model of the aggregate loss that you expect to observe in a given period of time. This section illustrates how you can use the new HPCDM procedure in SAS High-Performance Econometrics to accomplish this task. The following list summarizes some of the key features of PROC HPCDM:

- PROC HPCDM takes as input the frequency and severity models that are estimated by PROC SEVERITY and PROC COUNTREG, respectively, simulates an aggregate loss sample from the compound distribution, and uses the sample to compute empirical estimates of various summary statistics and percentiles and to display plots of nonparametric PDF and CDF estimates of the compound distribution. You can also request that PROC HPCDM write the entire simulated sample to an output data set so that you can later compute a risk measure of your choice.
- When the distributions of X and N are conditional on regressors, the CDM is also conditional on the values of regressors, and the simulation process becomes a scenario analysis. PROC HPCDM enables you to specify an input data set that contains the scenario with one or more entities and simulates the CDM of the aggregate loss over all entities.

- PROC HPCDM enables you to conduct parameter perturbation analysis that assesses the effect of uncertainty in the parameters of frequency and severity models on the estimates of the CDM.
- You can use PROC HPCDM to compute the distribution of an aggregate *adjusted* loss. For example, in insurance applications, you might want to compute the distribution of the *amount paid* after applying adjustments such as the deductible and policy limit to each individual *ground-up* loss.
- PROC HPCDM enables you to specify externally simulated counts. This is useful if you do not use PROC COUNTREG to estimate the frequency model or if you want to overcome some limitations of PROC COUNTREG in SAS/ETS 13.1.

In the simplest case, when the frequency and severity models do not depend on any regression effects, the HPCDM procedure requires the item store that is created by the COUNTREG procedure and the parameter estimates data set that is created by the SEVERITY procedure to simulate a compound distribution sample of a size that you specify. This is illustrated by the following statements for a Poisson count model and a gamma severity model (the code that generates the input data sets for PROC COUNTREG and PROC SEVERITY is available at http://support.sas.com/saspresents):

```
proc countreg data=claimcount;
   model numLosses= / dist=poisson;
   store countStorePoisson;
run;

proc severity data=claimsev outest=sevest covout plots=none;
   loss lossValue;
   dist gamma;
run;

proc hpcdm countstore=countStorePoisson severityest=sevest
           seed=13579 nreplicates=10000 plots=(edf(alpha=0.05) density)
           print=(summarystatistics percentiles)
           plots=(conditionaldensity(leftq=0.25 rightq=0.95));
   severitymodel gamma;
   output out=aggregateLossSample samplevar=aggloss;
   outsum out=aggregateLossSummary mean stddev skewness kurtosis
          p01 p05 p95 p995=var pctlpts=90 97.5;
run;
```

The STORE statement in the PROC COUNTREG step saves the count model information in the Work.CountStorePoisson item store. The OUTEST= option in the PROC SEVERITY statement stores the estimates of the fitted severity models in the Work.SevEst data set. The SEVERITYMODEL statement in the PROC HPCDM step specifies which severity model to use. In general, you can specify multiple severity models in this statement as long as their estimates are available in the SEVERITYEST= data set. Specifying the SEED= value helps you get an identical CDM sample each time you execute this step. Upon completion, PROC HPCDM creates the two output data sets that you specify in the OUT= options of the OUTPUT and OUTSUM statements. The Work.AggregateLossSample data set contains 10,000 observations, such that the value of the **AggLoss** variable in each observation represents one possible aggregate loss value that you can expect to see. Together, the set of the 10,000 values of the **AggLoss** variable represents one sample of compound distribution. PROC HPCDM uses this sample to compute the empirical estimates of various summary statistics and percentiles of the compound distribution. The Work.AggregateLossSummary data set contains the estimates of mean, standard deviation, skewness, and kurtosis that you specify in the OUTSUM statement. It also contains the estimates of the 1st, 5th, 90th, 95th, 97.5th, and 99.5th percentiles that you specify in the OUTSUM statement. In this example, the 99.5th percentile is assumed to represent the value at risk (VaR), and it is stored in a variable named **Var**.

Some of the default output and some of the output that you have requested by specifying the PRINT= option are shown in Figure 5. The “Sample Summary Statistics” table indicates that you can expect to see a mean aggregate loss of 4062.8 and a median aggregate loss of 3349.7 in one year. The “Sample Percentiles” table indicates that there is a 0.5% chance that the aggregate loss will exceed 15877.9, which is the VaR estimate. These summary statistics and percentile estimates provide a quantitative picture of the compound distribution.

You can also visually analyze the compound distribution by examining the plots that PROC HPCDM creates. The plot in Figure 6 shows the empirical distribution function (EDF), which is a nonparametric estimate of the CDF. The plot in Figure 7 shows the histogram and the kernel density estimates, which are nonparametric estimates of the PDF. The plots confirm the right-skewed nature of the CDM that is indicated by the estimate of skewness in Figure 5 and a relatively fat tail. You can visually analyze the left- and right-tail regions of the compound distribution by examining the conditional density plot in Figure 8 that is created by the PLOTS=CONDITIONALDENSITY option. The plot on the left side is the plot of $\Pr(Y|Y \leq 1449.1)$, the left-tail region, where 1449.1 is the 25th percentile as specified by the LEFTQ=0.25 option. The plot in the middle is the plot of $\Pr(Y|1449.1 < Y \leq 10672.5)$, the *body* of the distribution. The plot on the right side is the plot of $\Pr(Y|Y > 10672.5)$, the right-tail region, where 10672.5 is the 95th percentile as specified by the RIGHTQ=0.95 option. You can control the definitions of the tails by varying the LEFTQ= and RIGHTQ= option values. You can also request plots for only one of the tail regions.

Figure 5 Summary Statistics and Percentiles of the Poisson-Gamma Compound Distribution

Sample Summary Statistics			
Mean	4062.8	**Median**	3349.7
Standard Deviation	3429.6	**Interquartile Range**	4456.4
Variance	11761948.0	**Minimum**	0
Skewness	1.14604	**Maximum**	26077.4
Kurtosis	1.76466	**Sample Size**	10000

Sample Percentiles	
Percentile	**Value**
1	0
5	0
25	1449.1
50	3349.7
75	5905.5
90	8792.6
95	10672.5
97.5	12391.7
99	14512.5
99.5	15877.9
Percentile Method = 5	

Figure 6 Nonparametric CDF Plot

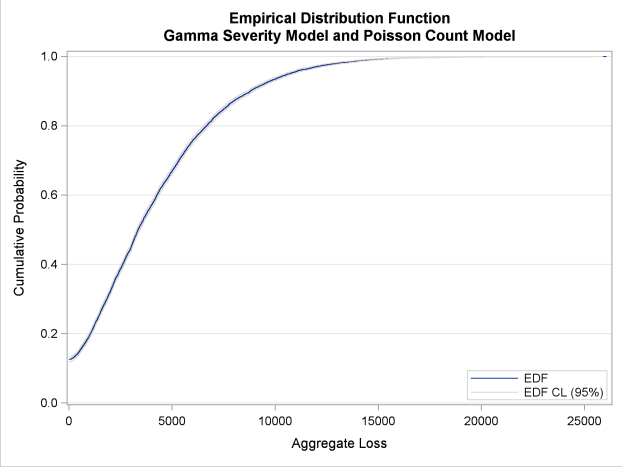


Figure 7 Density Plot

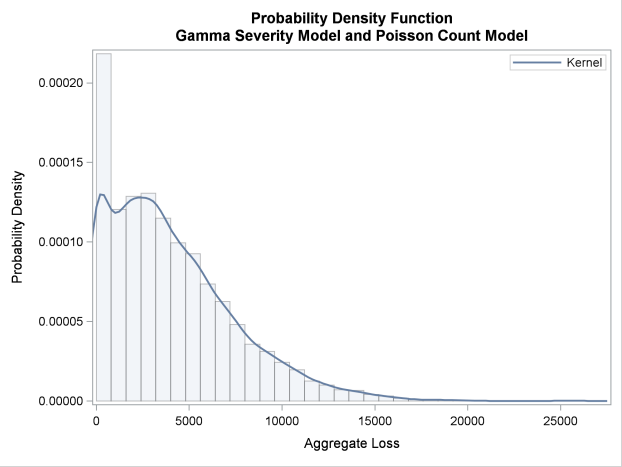
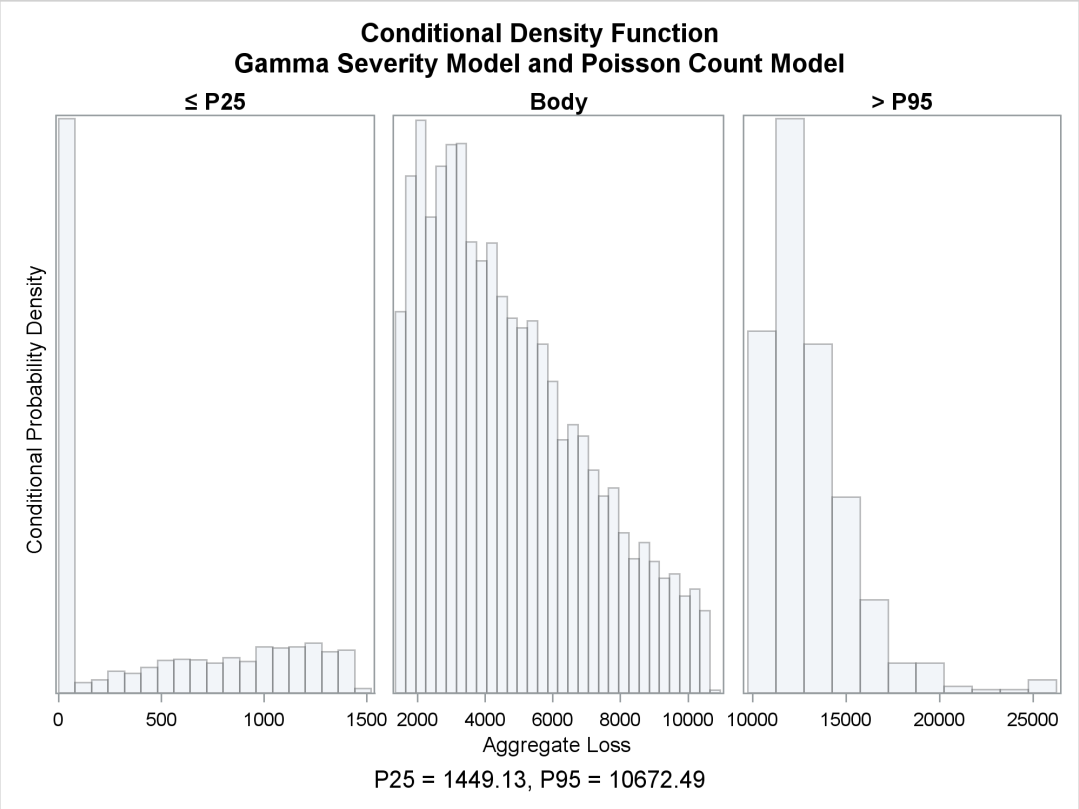


Figure 8 Conditional Density Plots for Tail and Body Regions



Note that for simple forms of the frequency and severity models, you can replicate the simulation method of PROC HPCDM by using a combination of SAS macros and DATA steps, but if you want to conduct all the types of analysis that PROC HPCDM provides, the macros and DATA steps can become very complex. Further, it is almost impossible to implement the procedure's high-performance features (multithreading and distributed computing, to be illustrated later) by using macros and DATA steps.

SIMULATING PORTFOLIO LOSSES: SCENARIO ANALYSIS

When frequency and severity models include regression effects, the distribution of the aggregate loss depends on the values of those regression effects. To predict the average and worst-case aggregate loss, you need to specify a *scenario* or *portfolio* that you are interested in analyzing. A scenario consists of one or more entities, and it is specified as a set of observations in a data set such that each observation contains the values of regressors that represent the characteristics of the entity and the world it operates in. Consider the operational risk data described previously. The frequency and severity models depend on a specific set of KRIs, measured monthly. The entity is the operating environment for a specific month. To estimate operational losses for one year, you need to specify a scenario that consists of 12 observations, one for each month. An observation encodes the KRI values that you expect to see in a given month. For the insurance example, the policyholder is the entity, and a portfolio consists of one or more policyholders.

This section describes how to conduct the scenario analysis by using the operational risk data. Note that the input data are organized in BY groups, where the **Line** variable is the BY variable. In the current release of the PROC COUNTREG procedure (SAS/ETS 13.1), item stores cannot be created when BY variables are specified, and PROC HPCDM needs an item store for count model estimates. So you need to conduct a separate scenario analysis for each line of business (LOB). This example analyzes the retail banking LOB. The following steps estimate the count and severity models:

```
proc countreg data=OpRiskLossCounts(where=(line = 'RetailBanking'));
   model numloss=corpKRI1 corpKRI2 cbKRI1 cbKRI2 cbKRI3 rbKRI1 rbKRI2/dist=poisson;
   store work.opriskCountStoreRetail;
run;
```

```
proc severity data=OpRiskLosses outest=opriskSevestRetail covout plots=none print=all;
   where line = 'RetailBanking';
   loss lossValue;
   scalemodel corpKRI1 corpKRI2 cbKRI2 rbKRI1 rbKRI3;
   dist logn gamma;
run;
```

For the example data, the Poisson regression model that contains the regressors **CorpKRI1**, **RbKRI1**, and **RbKRI2** is the best count model for this LOB, and the gamma regression model that contains the regressors **CorpKRI2**, **RbKRI1**, and **RbKRI2** is the best severity model for this LOB.

The following PROC HPCDM step uses the count model item store and severity model estimates for the retail banking LOB to estimate the aggregate loss over all 12 monthly operating environments that are recorded in the MultiConditionScenario data set:

```
proc hpcdm data=multiConditionScenario nreplicates=10000 seed=13579 print=all
           countstore=work.opriskCountStoreRetail severityest=opriskSevestRetail
           nperturbedSamples=50;
   severitymodel gamma;
run;
```

Figure 9 shows the estimates of various summary statistics and percentiles of the aggregate loss for the specified scenario. The average loss and worst-case loss (99.5th percentile) that you expect to incur in one year after accounting for the variation in monthly operating environments are 3301.6 and 11412.3 units, respectively.

Figure 9 Scenario Analysis Results

The HPCDM Procedure
Severity Model: Gamma
Count Model: Poisson

Sample Summary Statistics			
Mean	3301.6	**Median**	2751.9
Standard Deviation	2032.7	**Interquartile Range**	2256.4
Variance	4131995.2	**Minimum**	287.59358
Skewness	1.66205	**Maximum**	19519.5
Kurtosis	3.87290	**Sample Size**	10000

Sample Percentiles	
Percentile	**Value**
0	287.59358
1	751.13044
5	1113.4
25	1884.8
50	2751.9
75	4141.2
95	7483.5
99	10291.9
99.5	11412.3

Percentile Method = 5

PARAMETER PERTURBATION ANALYSIS

The NPERTURBEDSAMPLES= option in the preceding PROC HPCDM step triggers a parameter perturbation analysis, which enables you to assess the effect of uncertainty in the parameters of the frequency and severity on the estimates of the compound distribution by simulating multiple *perturbed* samples, each with a set of parameters that are randomly perturbed from their mean estimates. If covariance estimates are available, then PROC HPCDM makes random draws of parameter values from the multivariate normal distribution over the distribution and regression parameters. If covariance estimates are not available, then it uses the standard error of each parameter to draw a value from its univariate normal distribution. PROC HPCDM computes the summary statistics and percentiles for each perturbed sample and summarizes those numbers across all perturbed samples to compute the mean and standard error of each summary statistic and percentile. The standard error provides the measure of the effect of parameter uncertainty. Figure 10 shows the results of perturbation analysis for the operational loss example. You can now conclude, with more confidence, that the average and worst-case loss (99.5th percentile) that you expect to incur in the set of months that are encoded in the MultiConditionScenario data set are 3404.4 $\pm$ 495.8 and 12050.3 $\pm$ 2038.8, respectively.

Figure 10 Perturbation Analysis Results

Sample Perturbation Analysis

Statistic	Estimate	Standard Error
Mean	3404.4	495.76307
Standard Deviation	2088.3	372.84715
Variance	4499876.8	1636673.9
Skewness	1.78077	0.30083
Kurtosis	4.85659	1.71082

Number of Perturbed Samples = 50

Size of Each Sample = 10000

Sample Percentile Perturbation Analysis

Percentile	Estimate	Standard Error
0	270.69225	92.06673
1	811.76838	145.72128
5	1183.8	188.18256
25	1966.0	294.09261
50	2849.6	426.04486
75	4237.8	656.63971
95	7561.7	1217.1
99	10775.4	1777.5
99.5	12050.3	2038.8

Number of Perturbed Samples = 50

Size of Each Sample = 10000

MODELING ADJUSTED AGGREGATE LOSS

The primary outcome of running PROC HPCDM is the estimate of the compound distribution of aggregate loss, which is often referred to as the ground-up loss. In addition to that, an insurance company or bank might want to estimate the distribution of the amount that it expects to *pay* for the losses, or the amount that it can offload to another organization, such as a reinsurance company.

This section continues the automobile insurance example to illustrate how you can estimate the distribution of the aggregate amount that is paid to a group of policyholders. Let the amount that is paid to an individual policyholder be computed by using what is usually referred to as a *disappearing deductible* (Klugman, Panjer,

and Willmot 1998, Ch. 2). If X denotes the ground-up loss that a policyholder incurs, d denotes the lower limit on the deductible, d' denotes the upper limit on the deductible, u denotes the maximum payment amount for one loss event, and U denotes the annual maximum payment to a policyholder, then Y, the amount that is paid to the policyholder for each loss event, is defined as follows:

$$Y' = \begin{cases} 0 & X \le d \\ d' \frac{X-d}{d'-d} & d < X \le d' \\ X & d' < X \le u \\ u & X > u \end{cases} \qquad Y = \min(Y', \max(0, U - \sum_{\text{year}} Y))$$

The following PROC HPCDM step encodes this logic to compute the value of the **AmountPaid** symbol and specifies that symbol in the ADJUSTEDSEVERITY= option to estimate the distribution of the amount paid for a group of policyholders in the Work.GroupOfPolicies data set. This data set has now been expanded to include the following variables for each policyholder, in addition to the regression variables: **LowDeductible** to record d, **HighDeductible** to record d', **Limit** to record u, and **AnnualLimit** to record U.

```
/* Simulate the aggregate loss distribution and aggregate adjusted
   loss distribution for the modified set of policy provisions */
proc hpcdm data=groupOfPolicies nreplicates=10000 seed=13579 print=all
           countstore=work.autocountmodel severityest=work.sevregestauto
           nperturbedSamples=50 adjustedseverity=amountPaid;
   severitymodel logn;
   if (_sev_ <= lowDeductible) then
      amountPaid = 0;
   else do;
      if (_sev_ <= highDeductible) then
         amountPaid = highDeductible *
            (_sev_-lowDeductible)/(highDeductible-lowDeductible);
      else
         amountPaid = MIN(_sev_, limit); /* imposes per-loss payment limit */
      /* impose policyholder's annual limit */
      amountPaid = MIN(amountPaid, MAX(0,annualLimit - _cumadjsevforobs_));
   end;
run;
```

In addition to using the external information about deductibles and limits that is recorded in the input data, the programming statements use placeholder symbols for information that PROC HPCDM generates internally, such as the severity of the current loss event (_SEV_) and the cumulative adjusted severity for a specific entity in the scenario (_CUMADJSEVFOROBS_). (For additional placeholder symbols, see *SAS/ETS User's Guide: High-Performance Procedures*.) Note that the preceding PROC HPCDM step reads the count model parameters from the Work.AutoCountModel item store (see the section "FITTING AND SCORING COUNT MODELS") and the severity model parameters from the Work.SevRegEstAuto data set (see the section "FITTING AND SCORING SEVERITY MODELS"). The results of the perturbation analysis for the distribution of **AmountPaid** are shown in Figure 11. You can expect to pay a median amount of 10151.6 $\pm$ 1595.5 units and a worst-case amount (99.5th percentile) of 38795.9 $\pm$ 3006.5 units to the set of policyholders in the Work.GroupOfPolicies data set.

SPECIFYING EXTERNALLY SIMULATED COUNTS

There are situations in which you might already have an empirical frequency model or find that the count modeling capabilities of PROC COUNTREG are inadequate. In such cases, you can simulate the counts externally. For example, in SAS/ETS 13.1, when BY-group processing is requested, PROC COUNTREG does not support item stores that are required by PROC HPCDM. However, you can request that PROC COUNTREG store the count model estimates in the OUTEST= data set and then use the OUTEST= data set in a DATA step to simulate the counts for each BY group. You can specify externally simulated counts by using the EXTERNALCOUNTS statement in PROC HPCDM. For a detailed example that illustrates the use of the EXTERNALCOUNTS statement, see the HPCDM procedure chapter in *SAS/ETS User's Guide: High-Performance Procedures*.

Figure 11 Perturbation Summary of the Total Amount Paid for Policy Provisions

The HPCDM Procedure
Severity Model: Logn
Count Model: NegBin(p=2)

Adjusted Sample Percentile Perturbation Analysis

Percentile	Estimate	Standard Error
0	0	0
1	346.31536	200.47271
5	1581.0	466.83644
25	5402.9	1050.9
50	10151.6	1595.5
75	16738.8	2196.1
95	28107.2	2487.7
99	35816.7	2778.9
99.5	38795.9	3006.5

Number of Perturbed Samples = 50

Size of Each Sample = 10000

WARP SPEED AHEAD: HIGH-PERFORMANCE LOSS MODELING

Many insurance companies and banks have large amounts of loss event data, and the data keep getting bigger. A problem that does not necessarily have large input data but requires a large number of computations is referred to as a *big computation* problem. Fortunately, the computational costs per unit of processing power and the cost of random-access memory (RAM) are decreasing. To remain competitive, businesses need to harness the combination of increasingly affordable storage capacities and computing power to solve big data and big computation problems.

For the loss modeling process, the task of estimating frequency and severity models needs to work on large amounts of loss event data. The estimation time increases as the size of input data and the number of candidate models increase, so it is a combination of big data and big computation problems. The task of simulating the aggregate loss models requires as much computation power as possible to simulate large representative samples of the CDM. The larger the samples you can simulate, the more accurate your estimates of summary statistics and quantiles will be. Similarly, the larger the number of perturbed samples you can simulate, the more accurate your estimates of mean and standard error of summary statistics and quantiles will be. The time it takes to simulate a more accurate CDM with accurate parameter sensitivity estimates increases as the sample size, scenario size, and number of perturbation samples increase. This is primarily a big computation problem.

You can execute the entire loss modeling process significantly faster by using PROC HPCOUNTREG and PROC HPSEVERITY, which are high-performance versions of the COUNTREG and SEVERITY procedures, and by using the high-performance features of the HPCDM procedure. PROC HPCOUNTREG, PROC HPSEVERITY, and PROC HPCDM are all part of SAS High-Performance Econometrics. Each procedure can be executed in two modes: single-machine mode, where multiple threads of computation execute in parallel on multiple CPU cores of the machine; and distributed mode, where data and computations are distributed across a grid appliance, which is a cluster of multiple machines, and threads of computation execute in parallel within each multicore machine (node) in the grid. The large amount of input loss event data that PROC HPCOUNTREG and PROC HPSEVERITY operate on can be distributed in several ways to take advantage of distributed data storage and management systems such as Hadoop and distributed relational databases of multiple vendors. The computations can execute alongside the data or on a dedicated cluster of computational nodes that interact with the nodes of the distributed database system. For more information about the various modes of distributing data and computations, see *SAS/ETS User's Guide: High-Performance Procedures*.

The following statements illustrate how you can estimate the frequency and severity models on a grid appliance by using PROC HPCOUNTREG and PROC HPSEVERITY, respectively:

```
proc hpcountreg data=autolosscountsbig outest=autoCountEstHP;
   class gender carType education;
   model numloss = age gender carType annualMiles education / dist=negbin;
   performance nodes=24 details host="&GRIDHOST" install="&GRIDINSTALLLOC";
run;

proc hpseverity data=autolossesbig outest=work.sevRegEstAutoHP print=all;
   loss lossamount / lt=deductible rc=limit;
   scalemodel carType carSafety income;
   dist _predefined_;
   performance nodes=24 details host="&GRIDHOST" install="&GRIDINSTALLLOC";
run;
```

The PERFORMANCE statement in the PROC step of each procedure requests that the procedure be executed on 24 nodes of the grid appliance host that is identified by the GRIDHOST macro variable. The shared SAS libraries for SAS High-Performance Econometrics are installed at the location that is identified by the GRIDINSTALLLOC macro variable. The preceding steps illustrate the client-local data model of the distributed execution mode, where the DATA= data set is located on the client machine and each procedure distributes those data to the nodes of the grid appliance before starting the parameter estimation process. When the data are distributed on the grid appliance, you need to specify an appropriate LIBNAME statement for the appliance and some more options in the PERFORMANCE statement. (For more information, see the *SAS/ETS User's Guide: High-Performance Procedures*.)

Figure 12 and Figure 13 show the scalability results of PROC HPCOUNTREG and PROC HPSEVERITY, respectively, for simulated input data that contain 25 million observations in the Work.AutoLossCountsBig data set and approximately 50 million observations in the Work.AutoLossesBig data set. The plots show that you can significantly reduce the estimation time by using more nodes. The incremental benefit decreases as the number of nodes increases, because the cost of synchronizing communications among nodes starts to outweigh the amount of computational work that is available to each node. This behavior is typical of all high-performance procedures.

The simulation phase of the loss modeling process can be executed faster if you run PROC HPCDM in a high-performance mode, as illustrated in the following statements:

```
proc hpcdm data=groupOfPolicies nreplicates=1000000 seed=13579 print=all
           countstore=work.autocountmodel severityest=work.sevregestauto
           nperturbedSamples=50 adjustedseverity=amountPaid;
   severitymodel logn;
   amountPaid = MIN(MAX(0,_sev_-lowDeductible), limit);
   performance nodes=24 details host="&GRIDHOST" install="&GRIDINSTALLLOC";
run;
```

For a scenario of 500 policyholders, the scalability results of the preceding PROC HPCDM step, which you obtain by varying the NODES= value in the PERFORMANCE statement from 1 to 24, are shown in Figure 14. The total time includes the time to simulate 1 million yearly loss events to create and analyze one unperturbed sample and 50 perturbed samples for the ground-up loss as well as the amount paid. The scalability results indicate that you can reduce the time from around 45 minutes on one node to less than 4 minutes by using 16 or more nodes.

When executed in single-machine mode, PROC HPCDM by default attempts to use multiple CPU cores of the machine. You can control the number of parallel threads of execution by using the CPUCOUNT= system option or the NTHREADS= option in the PERFORMANCE statement. Figure 15 shows the scalability results of multithreaded processing on a single machine. As with the performance behavior of distributed processing, the incremental benefit of adding more threads is reduced because of the overhead that is associated with interthread synchronization. But, up to a certain number of threads, using more threads yields significantly faster results.

Note that PROC HPSEVERITY supports all the features of PROC SEVERITY except for ODS graphics, whereas PROC HPCOUNTREG supports a subset of features that are supported by PROC COUNTREG. For example, in SAS/ETS 13.1, PROC HPCOUNTREG does not support the STORE statement. So to use

Figure 12 PROC HPCOUNTREG Scalability

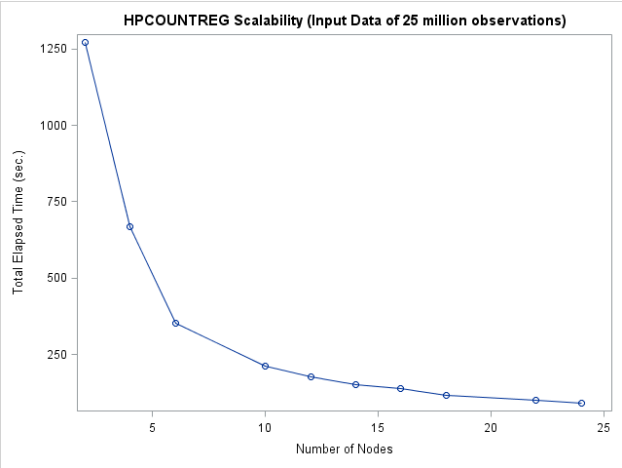


Figure 13 PROC HPSEVERITY Scalability

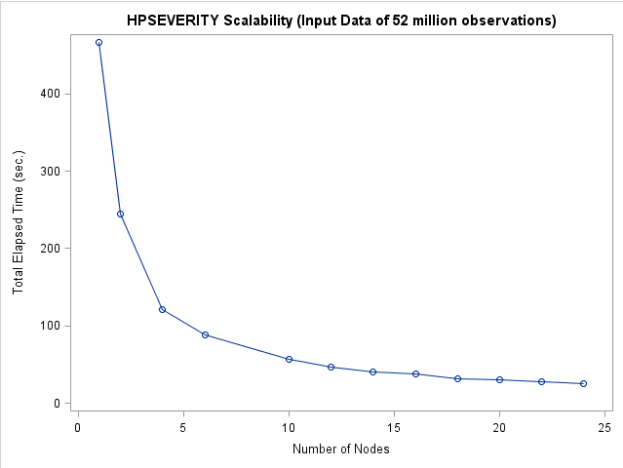


Figure 14 HPCDM Scalability: Distributed Mode

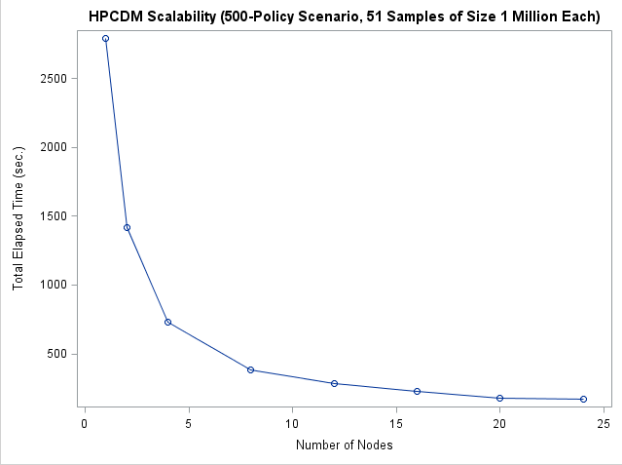
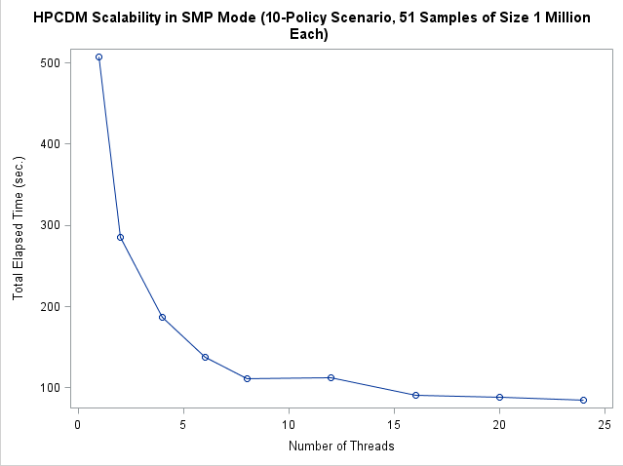


Figure 15 HPCDM Scalability: Single-Machine Mode



PROC HPCOUNTREG in conjunction with PROC HPCDM, first you need to store the count model estimates in an OUTEST= data set, then simulate the counts by using a DATA step, and finally supply those counts to PROC HPCDM by using the EXTERNALCOUNTS statement. For an example of how to simulate counts by using the OUTEST= data set that is prepared by PROC COUNTREG or PROC HPCOUNTREG, see the HPCDM procedure chapter in *SAS/ETS User's Guide: High-Performance Procedures*.

CONCLUSION

This paper illustrates the loss modeling process that starts with the data collection and preparation phase and results in the estimation of distribution of the aggregate loss under the assumption that the two key components of the loss, frequency and severity, are modeled separately. SAS/ETS offers three procedures, PROC COUNTREG, PROC SEVERITY, and PROC HPCDM, to implement this process from end to end. The paper describes the capabilities of each procedure by using simulated data sets that are inspired by real-world applications in the insurance and banking industries. You can use the new scoring features of the SEVERITY and COUNTREG procedures to predict the expected loss severity and number of loss events for an individual entity. The new HPCDM procedure offers several key features to predict losses that are aggregated over a period of time and over a group of entities. The output of PROC HPCDM includes estimates of summary statistics and quantiles of the compound distribution of the aggregate loss. It lets you assess the impact of external effects and uncertainty in the parameters of frequency and severity models on the compound distribution estimates. You can also estimate the distribution of an aggregate adjusted loss, such as the aggregate amount paid to policyholders, by specifying programming statements in a PROC HPCDM step.

You can perform the compound distribution modeling significantly faster by exploiting all the computational resources, thanks to the multithreaded and distributed computing capabilities that are built into PROC HPCDM. In fact, the end-to-end process can be executed very quickly because of the availability of PROC HPSEVERITY and PROC HPCOUNTREG, which are high-performance equivalents of the SEVERITY and COUNTREG procedures. The primary advantages of using all three procedures are the ease with which they integrate together and the simplicity and efficiency of using them to assess portfolio losses.

REFERENCES

Cameron, A. C. and Trivedi, P. K. (1986), "Econometric Models Based on Count Data: Comparisons and Applications of Some Estimators and Tests," *Journal of Applied Econometrics*, 1, 29–53.

Gentleman, R. and Geyer, C. J. (1994), "Maximum Likelihood for Interval Censored Data: Consistency and Computation," *Biometrika*, 81, 618–623.

Kaplan, E. L. and Meier, P. (1958), "Nonparametric Estimation from Incomplete Observations," *Journal of the American Statistical Association*, 53, 457–481.

Klugman, S. A., Panjer, H. H., and Willmot, G. E. (1998), *Loss Models: From Data to Decisions*, New York: John Wiley & Sons.

Lai, T. L. and Ying, Z. (1991), "Estimating a Distribution Function with Truncated and Censored Data," *Annals of Statistics*, 19, 417–442.

Lambert, D. (1992), "Zero-Inflated Poisson Regression with an Application to Defects in Manufacturing," *Technometrics*, 34, 1–14.

Lynden-Bell, D. (1971), "A Method of Allowing for Known Observational Selection in Small Samples Applied to 3CR Quasars," *Monthly Notices of the Royal Astronomical Society*, 155, 95–118.

Turnbull, B. W. (1976), "The Empirical Distribution Function with Arbitrarily Grouped, Censored, and Truncated Data," *Journal of the Royal Statistical Society, Series B*, 38, 290–295.

ACKNOWLEDGMENTS

The authors are grateful to Richard Potter and Mark Little of the Advanced Analytics Division at SAS Institute Inc. for their valuable assistance in the research and development of the procedures discussed in this paper. Thanks are also due to Ed Huddleston for his valuable editorial comments.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Mahesh V. Joshi
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
mahesh.joshi@sas.com