

SAS/ETS[®] 15.1 User's Guide The HPCDM Procedure

This document is an individual chapter from *SAS/ETS® 15.1 User's Guide*.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2018. *SAS/ETS® 15.1 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/ETS® 15.1 User's Guide

Copyright © 2018, SAS Institute Inc., Cary, NC, USA

All Rights Reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

The scanning, uploading, and distribution of this book via the Internet or any other means without the permission of the publisher is illegal and punishable by law. Please purchase only authorized electronic editions and do not participate in or encourage electronic piracy of copyrighted materials. Your support of others' rights is appreciated.

U.S. Government License Rights; Restricted Rights: The Software and its documentation is commercial computer software developed at private expense and is provided with RESTRICTED RIGHTS to the United States Government. Use, duplication, or disclosure of the Software by the United States Government is subject to the license terms of this Agreement pursuant to, as applicable, FAR 12.212, DFAR 227.7202-1(a), DFAR 227.7202-3(a), and DFAR 227.7202-4, and, to the extent required under U.S. federal law, the minimum restricted rights as set out in FAR 52.227-19 (DEC 2007). If FAR 52.227-19 is applicable, this provision serves as notice under clause (c) thereof and no other notice is required to be affixed to the Software or documentation. The Government's rights in Software and documentation shall be only those set forth in this Agreement.

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

November 2018

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

SAS software may be provided with certain third-party software, including but not limited to open-source software, which is licensed under its applicable third-party software license agreement. For license information about third-party software distributed with SAS software, refer to <http://support.sas.com/thirdpartylicenses>.

Chapter 16

The HPCDM Procedure

Contents

Overview: HPCDM Procedure	926
Getting Started: HPCDM Procedure	928
Estimating a Simple Compound Distribution Model	928
Analyzing the Effect of Parameter Uncertainty on the Compound Distribution	932
Scenario Analysis	934
Syntax: HPCDM Procedure	942
Functional Summary	942
PROC HPCDM Statement	944
BY Statement	950
DISTBY Statement	951
EXTERNALCOUNTS Statement	951
OUTPUT Statement	952
OUTSUM Statement	953
PERFORMANCE Statement	956
SEVERITYMODEL Statement	956
Programming Statements	956
Details: HPCDM Procedure	957
Specifying Scenario Data in the DATA= Data Set	957
Simulation Procedure	958
Simulation of Adjusted Compound Distribution Sample	964
Parameter Perturbation Analysis	972
Descriptive Statistics	973
Input Specification	975
Output Data Sets	977
Displayed Output	978
ODS Graphics	980
Examples: HPCDM Procedure	981
Example 16.1: Estimating the Probability Distribution of Insurance Payments	981
Example 16.2: Using Externally Simulated Count Data	985
Example 16.3: Scenario Analysis with Rich Regression Effects and BY Groups	993
References	1001

Overview: HPCDM Procedure

In many loss modeling applications, the loss events are analyzed by modeling the severity (magnitude) of loss and the frequency (count) of loss separately. The primary goal of preparing these models is to estimate the aggregate loss—that is, the total loss that occurs over a period of time for which the frequency model is applicable. For example, an insurance company might want to assess the expected and worst-case losses for a particular business line, such as automobile insurance, over an entire year given the models for the number of losses in a year and the severity of each loss. A bank might want to assess the value-at-risk (VaR), a measure of the worst-case loss, for a portfolio of assets given the frequency and severity models for each asset type.

Loss severity and loss frequency are random variables, so the aggregate loss is also a random variable. Instead of preparing a point estimate of the expected aggregate loss, it is more desirable to estimate its probability distribution, because this enables you to infer various aspects of the aggregate loss such as measures of location, scale (variability), and shape in addition to percentiles. For example, the value-at-risk that banks or insurance companies use to compute regulatory capital requirements is usually the estimate of the 97.5th or 99th percentile from the aggregate loss distribution.

Let N represent the frequency random variable for the number of loss events that occur in the time period of interest. Let X represent the severity random variable for the magnitude of one loss event. Then, the aggregate loss S is defined as

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

The goal is to estimate the probability distribution of S . Let $F_X(x)$ denote the cumulative distribution function (CDF) of X , $F_X^{*n}(x)$ denote the n -fold convolution of the CDF of X , and $\Pr(N = n)$ denote the probability of seeing n losses as per the frequency distribution. The CDF of S is theoretically computable as

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

This probability distribution model of S , characterized by the CDF $F_S(s)$, is referred to as a *compound distribution model* (CDM). The HPCDM procedure computes an estimate of the CDM, given the distribution models of X and N .

PROC HPCDM accepts the severity model of X as estimated by the SEVERITY procedure. It accepts the frequency model of N as estimated by the COUNTREG procedure. Both the SEVERITY and COUNTREG procedures are part of SAS/ETS software. Both procedures allow models of X and N to be conditional on external factors (regressors). In particular, you can model the severity distribution such that its scale parameter depends on severity regressors, and you can model the frequency distribution such that its mean depends on frequency regressors. The frequency model can also be a zero-inflated model. PROC HPCDM uses the estimates of model parameters and the values of severity and frequency regressors to estimate the compound distribution model.

Direct computation of F_S is usually a difficult task because of the need to compute the n -fold convolution. Klugman, Panjer, and Willmot (1998, Ch. 4) suggest some relatively efficient recursion and inversion methods for certain combinations of severity and frequency distributions. However, those methods assume that distributions of N and X are fixed and all X s are identically distributed. When the distributions of X

and N are conditional on regressors, each set of regressor values results in a different distribution. So you must repeat the recursion and inversion methods for each combination of regressor values, and this repetition makes these methods prohibitively expensive. PROC HPCDM instead estimates the compound distribution by using a Monte Carlo simulation method, which can use all available computational resources to generate a sufficiently large, representative sample of the compound distribution while accommodating the dependence of distributions of X and N on external factors. Conceptually, the simulation method works as follows:

1. Use the specified frequency model to draw a value N , which represents the number of loss events.
2. Use the specified severity model to draw N values, each of which represents the magnitude of loss for each of the N loss events.
3. Add the N severity values from step 2 to compute aggregate loss S as

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

This forms one sample point of the CDM.

Steps 1 through 3 are repeated M number of times, where M is specified by you, to obtain the representative sample of the CDM. PROC HPCDM analyzes this sample to compute empirical estimates of various summary statistics of the compound distribution such as the mean, variance, skewness, and kurtosis in addition to percentiles such as the median, the 95th percentile, the 99th percentile, and so on. You can also use PROC HPCDM to write the entire simulated sample to an output data set and to produce the plot of the empirical distribution function (EDF), which serves as a nonparametric estimate of F_S .

The simulation process gets more complicated when the frequency and severity models contain regression effects. The CDM is then conditional on the given values of regressors. The simulation process essentially becomes a scenario analysis, because you need to specify the expected values of the regressors that together represent the scenario for which you want to estimate the CDM. PROC HPCDM enables you to specify an input data set that contains the scenario. If you are modeling a group of entities together (such as a portfolio of multiple assets or a group of insurance policies), each with a different set of characteristics, then the scenario consists of more than one observation, and each observation corresponds to a different entity. PROC HPCDM enables you to specify such a group scenario in the input data set and performs a realistic simulation of loss events that each entity can generate.

PROC HPCDM also enables you to specify externally simulated counts. This is useful if you have an empirical frequency model or if you estimate the frequency model by using a method other than PROC COUNTREG and simulate counts by using such a model. You can specify M replications of externally simulated counts. For each of the replications, in step 1 of the simulation, instead of using the frequency model, PROC HPCDM uses the count N that you specify. If the severity model contains regression effects, then you can specify the scenario to simulate for each of the M replications.

If the parameters of your severity and frequency models have uncertainty associated with them, and they usually do, then you can use PROC HPCDM to conduct parameter perturbation analysis to assess the effect of parameter uncertainty on the estimates of CDM. If you specify that P perturbed samples be generated, then the parameter set is perturbed P times, and each time PROC HPCDM makes a random draw from either the univariate normal distribution of each parameter or the multivariate normal distribution over all parameters. For each of the P perturbed parameter sets, a full compound distribution sample is simulated and summarized.

This process yields P number of estimates for each summary statistic and percentile, which are then used to provide you with estimates of the location and variability of each summary statistic and percentile.

You can also 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* in a given time period after applying adjustments such as deductible and policy limit to each individual loss. PROC HPCDM enables you to specify SAS programming statements to adjust each severity value. If X_j^a represents the adjusted severity value, then PROC HPCDM computes S^a , an aggregate adjusted loss, as

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

All the analyses that PROC HPCDM conducts for the aggregate unadjusted loss, including scenario analysis and parameter perturbation analysis, are also conducted for the aggregate adjusted loss, thereby giving you a comprehensive picture of the adjusted compound distribution model.

Getting Started: HPCDM Procedure

This section outlines the use of the HPCDM procedure to fit compound distribution models. The examples are intended as a gentle introduction to some of the features of the procedure.

Estimating a Simple Compound Distribution Model

This example illustrates the simplest use of PROC HPCDM. Assume that you are an insurance company that has used the historical data about the number of losses per year and the severity of each loss to determine that the Poisson distribution is the best distribution for the loss frequency and that the gamma distribution is the best distribution for the severity of each loss. Now, you want to estimate the distribution of an aggregate loss to determine the worst-case loss that can be incurred by your policyholders in a year. In other words, you want to estimate the compound distribution of $S = \sum_{i=1}^N X_i$, where the loss frequency, N , follows the fitted Poisson distribution and the severity of each loss event, X_i , follows the fitted gamma distribution.

If your historical count and severity data are stored in the data sets `Work.ClaimCount` and `Work.ClaimSev`, respectively, then you need to ensure that you use the following PROC COUNTREG and PROC SEVERITY steps to fit and store the parameter estimates of the frequency and severity models:

```
/* Fit an intercept-only Poisson count model and
   write estimates to an item store */
proc countreg data=claimcount;
  model numLosses= / dist=poisson;
  store countStorePoisson;
run;

/* Fit severity models and write estimates to a data set */
proc severity data=claimsev criterion=aicc outest=sevest covout plots=none;
  loss lossValue;
  dist _predefined_;
run;
```

The STORE statement in the PROC COUNTREG step saves the count model information, including the parameter estimates, in the Work.CountStorePoisson item store. An item store contains the model information in a binary format that cannot be modified after it is created. You can examine the contents of an item store that is created by a PROC COUNTREG step by specifying a combination of the RESTORE= option and the SHOW statement in another PROC COUNTREG step. For more information, see Chapter 11, “[The COUNTREG Procedure](#).”

The OUTEST= option in the PROC SEVERITY statement stores the estimates of all the fitted severity models in the Work.SevEst data set. Let the best severity model that the PROC SEVERITY step chooses be the gamma distribution model.

You can now submit the following PROC HPCDM step to simulate an aggregate loss sample of size 10,000 by specifying the count model’s item store in the COUNTSTORE= option and the severity model’s data set of estimates in the SEVERITYEST= option:

```
/* Simulate and estimate Poisson-gamma compound distribution model */
proc hpcdm countstore=countStorePoisson severityest=sevest
    seed=13579 nreplicates=10000 plots=(edf(alpha=0.05) density)
    print=(summarystatistics percentiles);
    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 SEVERITYMODEL statement requests that an aggregate sample be generated by compounding only the gamma distribution and the frequency distribution. Specifying the SEED= value helps you get an identical sample each time you execute this step, provided that you use the same execution environment. In the single-machine mode of execution, the execution environment is the combination of the operating environment and the number of threads that are used for execution. In the distributed computing mode, the execution environment is the combination of the operating environment, the number of nodes, and the number of threads that are used for execution on each node.

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 in one year. 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. The value-at-risk (VaR) is an aggregate loss value such that there is a very low probability that an observed aggregate loss value exceeds the VaR. One of the commonly used probability levels to define VaR is 0.005, which makes the 99.5th percentile an empirical estimate of the VaR. Hence, the OUTSUM statement of this example stores the 99.5th percentile in a variable named VaR. VaR is one of the widely used measures of worst-case risk.

Some of the default output and some of the output that you have requested by specifying the PRINT= option are shown in [Figure 16.1](#).

Figure 16.1 Information, Summary Statistics, and Percentiles of the Poisson-Gamma Compound Distribution

The HPCDM Procedure			
Severity Model: Gamma			
Count Model: Poisson			
Compound Distribution Information			
Severity Model	Gamma Distribution		
Count Model	Poisson Model in Item Store WORK.COUNTSTOREPOISSON		
Sample Summary Statistics			
Mean	4076.8	Median	3440.2
Standard Deviation	3442.6	Interquartile Range	4523.9
Variance	11851305.5	Minimum	0
Skewness	1.14554	Maximum	27971.5
Kurtosis	1.75272	Sample Size	10000
Sample Percentiles			
Percentile	Value		
1	0		
5	0		
25	1430.7		
50	3440.2		
75	5954.6		
90	8743.8		
95	10740.0		
97.5	12453.3		
99	14738.1		
99.5	16406.8		
Percentile Method = 5			

The “Sample Summary Statistics” table indicates that for the given parameter estimates of the Poisson frequency and gamma severity models, you can expect to see a mean aggregate loss of 4,076.8 and a median aggregate loss of 3,440.2 in a year. The “Sample Percentiles” table indicates that there is a 0.5% chance that the aggregate loss exceeds 16,406.8, which is the VaR estimate, and a 2.5% chance that the aggregate loss exceeds 12,453.3. These summary statistic 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 prepares. The first plot in [Figure 16.2](#) shows the empirical distribution function (EDF), which is a nonparametric estimate of the cumulative distribution function (CDF). The second plot shows the histogram and the kernel density estimate, which are nonparametric estimates of the probability density function (PDF).

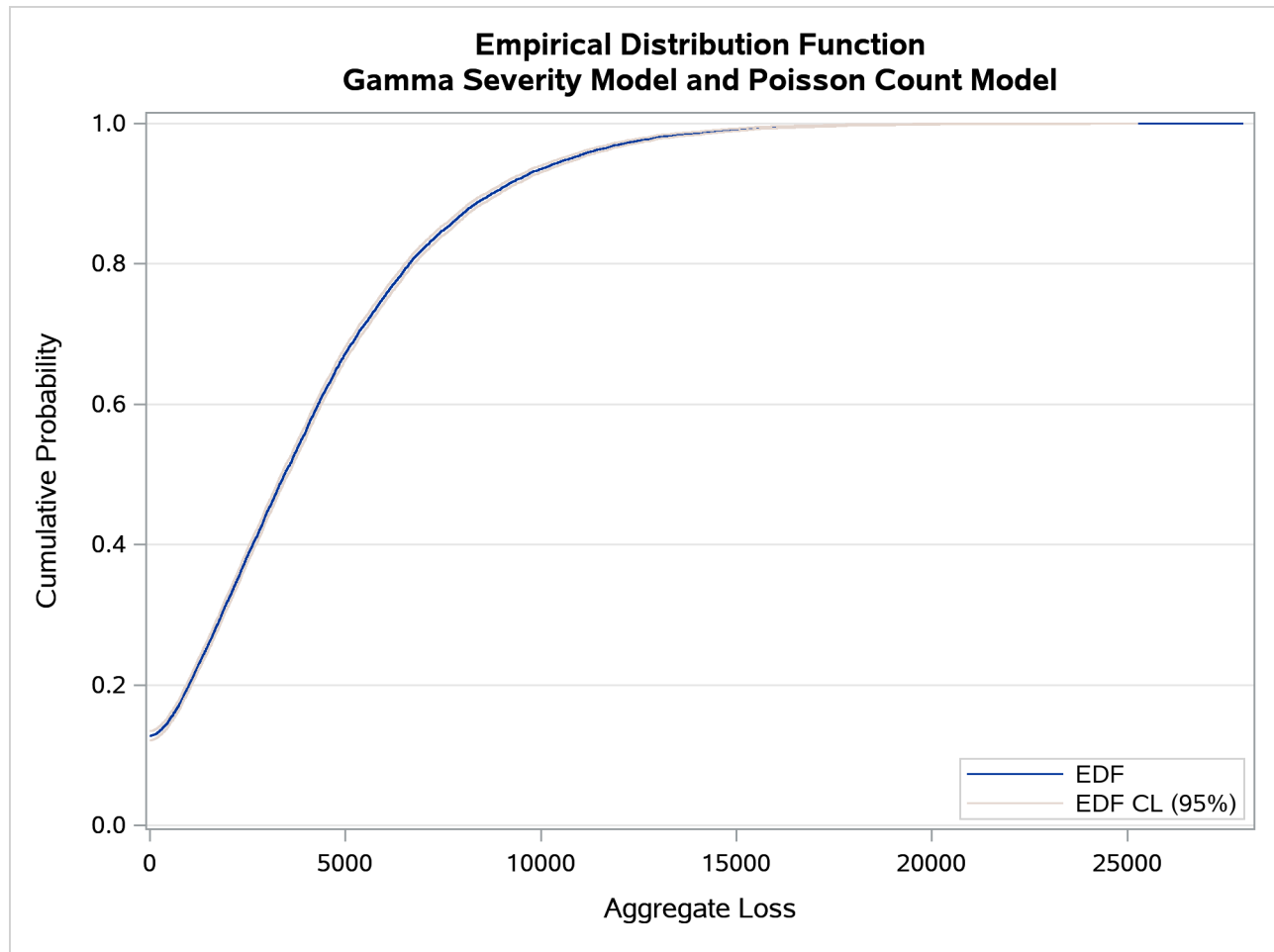
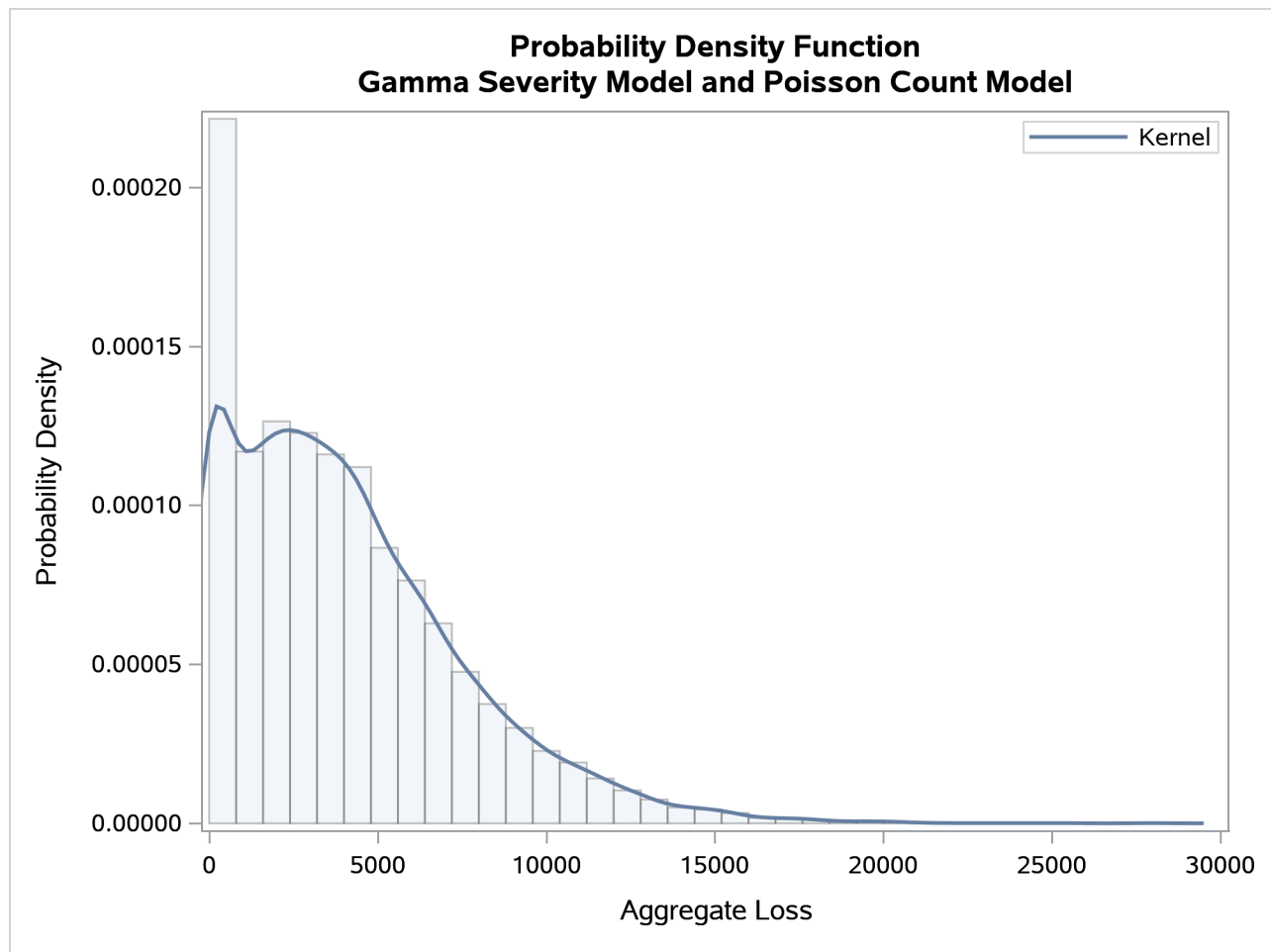
Figure 16.2 Nonparametric CDF and PDF Plots of the Poisson-Gamma Compound Distribution

Figure 16.2 *continued*

The plots confirm the right skew that is indicated by the estimate of skewness in Figure 16.1 and a relatively fat tail, which is indicated by comparing the maximum and the 99.5th percentiles in Figure 16.1.

Analyzing the Effect of Parameter Uncertainty on the Compound Distribution

Continuing with the previous example, note that you have fitted the frequency and severity models by using the historical data. Even if you choose the best-fitting models, the true underlying models are not known exactly. This fact is reflected in the uncertainty that is associated with the parameters of your models. Any compound distribution estimate that is computed by using these uncertain parameter estimates is inherently uncertain. You can request that PROC HPCDM conduct parameter perturbation analysis, which assesses the effect of the parameter uncertainty on the estimates of the compound distribution by simulating multiple samples, each with parameters that are randomly perturbed from their mean estimates.

The following PROC HPCDM step adds the NPerturbedSamples= option to the PROC HPCDM statement to request that perturbation analysis be conducted and the PRINT=PERTURBSUMMARY option to request that a summary of the perturbation analysis be displayed:

```

/* Conduct parameter perturbation analysis of
the Poisson-gamma compound distribution model */
proc hpcdm countstore=countStorePoisson severityest=sevest
    seed=13579 nreplicates=10000 nperturbedsamples=30
    print(only)=(perturbsummary) plots=none;
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 Work.AggregateLossSummary data set contains the specified summary statistics and percentiles for all 30 perturbed samples. You can identify a perturbed sample by the value of the `_DRAWID_` variable. The first few observations of the Work.AggregateLossSummary data set are shown in Figure 16.3. For the first observation, the value of the `_DRAWID_` variable is 0, which represents an unperturbed sample—that is, the aggregate sample that is simulated without perturbing the parameters from their means.

Figure 16.3 Summary Statistics and Percentiles of the Perturbed Samples

<code>_SEVERITYMODEL_</code>	<code>_COUNTMODEL_</code>	<code>_DRAWID_</code>	<code>_SAMPLEVAR_</code>	<code>N</code>	<code>MEAN</code>	<code>STDDEV</code>
Gamma	Poisson	0	aggloss	10000	4076.78	3442.57
Gamma	Poisson	1	aggloss	10000	4155.34	3430.45
Gamma	Poisson	2	aggloss	10000	4024.20	3407.80
Gamma	Poisson	3	aggloss	10000	4241.48	3565.67
Gamma	Poisson	4	aggloss	10000	4161.65	3544.71
Gamma	Poisson	5	aggloss	10000	3892.26	3273.01
Gamma	Poisson	6	aggloss	10000	4474.95	3704.71
Gamma	Poisson	7	aggloss	10000	4216.14	3476.55
Gamma	Poisson	8	aggloss	10000	4049.96	3413.21
Gamma	Poisson	9	aggloss	10000	3950.08	3350.04
Gamma	Poisson	10	aggloss	10000	4286.65	3668.01

<code>SKEWNESS</code>	<code>KURTOSIS</code>	<code>P01</code>	<code>P05</code>	<code>P90</code>	<code>P95</code>	<code>P97_5</code>	<code>var</code>
1.14554	1.75272	0	0	8743.85	10740.03	12453.26	16406.81
1.12929	1.85707	0	0	8783.93	10569.44	12448.84	16390.42
1.16006	1.84717	0	0	8599.78	10441.09	12242.83	16219.61
1.11373	1.48627	0	0	9133.00	11107.39	12974.48	16946.76
1.17400	1.79535	0	0	8943.12	10800.95	12780.92	17142.43
1.08180	1.45528	0	0	8334.01	10180.93	11742.12	15147.64
1.07704	1.41288	0	0	9606.49	11489.24	13304.55	17662.93
1.11500	1.58827	0	0	8890.20	10732.59	12600.30	16581.44
1.14044	1.61876	0	0	8671.02	10546.62	12323.83	16333.81
1.09693	1.35455	0	0	8561.27	10322.30	11986.43	15829.09
1.16766	1.75264	0	0	9328.43	11299.10	13240.13	17417.01

The `PRINT=PERTURBSUMMARY` option in the preceding `PROC HPCDM` step produces the “Sample Perturbation Analysis” and “Sample Percentile Perturbation Analysis” tables that are shown in Figure 16.4. The tables show that you can expect a mean aggregate loss of about 4,098.5 and the standard error of the mean is 172.1. If you want to use the VaR estimate to determine the amount of reserves that you need to maintain to cover the worst-case loss, then you should consider not only the mean estimate of the 99.5th

percentile, which is about 16,448.2, but also the standard error of 708.9 to account for the effect of uncertainty in your frequency and severity parameter estimates.

Figure 16.4 Summary of Perturbation Analysis of the Poisson-Gamma Compound Distribution

The HPCDM Procedure		
Severity Model: Gamma		
Count Model: Poisson		
Sample Perturbation Analysis		
Statistic	Estimate	Standard Error
Mean	4098.5	172.08823
Standard Deviation	3470.4	136.68712
Variance	12062522	947666.8
Skewness	1.13817	0.04237
Kurtosis	1.65486	0.21853
Number of Perturbed Samples = 30		
Size of Each Sample = 10000		
Sample Percentile Perturbation Analysis		
Percentile	Estimate	Standard Error
1	0	0
5	0	0
25	1425.4	90.99084
50	3421.7	155.81011
75	6003.1	244.90738
90	8818.2	362.42625
95	10732.8	422.41895
97.5	12540.3	504.12071
99	14839.4	680.49452
99.5	16448.2	708.87293
Number of Perturbed Samples = 30		
Size of Each Sample = 10000		

Scenario Analysis

The distributions of loss frequency and loss severity often depend on exogenous variables (regressors). For example, the number of losses and the severity of each loss that an automobile insurance policyholder incurs might depend on the characteristics of the policyholder and the characteristics of the vehicle. When you fit frequency and severity models, you need to account for the effects of such regressors on the probability distributions of the counts and severity. The COUNTREG procedure enables you to model regression effects on the mean of the count distribution, and the SEVERITY procedure enables you to model regression effects on the scale parameter of the severity distribution. When you use these models to estimate the compound distribution model of the aggregate loss, you need to specify a set of values for all the regressors, which represents the state of the world for which the simulation is conducted. This is referred to as the what-if or scenario analysis.

Consider that you, as an automobile insurance company, have postulated that the distribution of the loss event frequency depends on five regressors (external factors): age of the policyholder, gender, 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 these 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 historical data about the number of losses that various policyholders incur in a year be recorded in the **NumLoss** variable of the **Work.LossCounts** data set, and let the severity of each loss be recorded in the **LossAmount** variable of the **Work.Losses** data set.

The following PROC COUNTREG step fits the count regression model and stores the fitted model information in the **Work.CountregModel** item store:

```
/* Fit negative binomial frequency model for the number of losses */
proc countreg data=losscounts;
  model numloss = age gender carType annualMiles education / dist=negbin;
  store work.countregmodel;
run;
```

You can examine the parameter estimates of the count model that are stored in the **Work.CountregModel** item store by submitting the following statements:

```
/* Examine the parameter estimates for the model in the item store */
proc countreg restore=work.countregmodel;
  show parameters;
run;
```

The “Parameter Estimates” table that is displayed by the SHOW statement is shown in [Figure 16.5](#).

Figure 16.5 Parameter Estimates of the Count Regression Model

ITEM STORE CONTENTS: WORK.COUNTREGMODEL

Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > t
Intercept	1	0.910479	0.090515	10.06	<.0001
age	1	-0.626803	0.058547	-10.71	<.0001
gender	1	1.025034	0.032099	31.93	<.0001
carType	1	0.615165	0.031153	19.75	<.0001
annualMiles	1	-1.010276	0.017512	-57.69	<.0001
education	1	-0.280246	0.021677	-12.93	<.0001
_Alpha	1	0.318403	0.020090	15.85	<.0001

The following PROC SEVERITY step fits the severity scale regression models for all the common distributions that are predefined in PROC SEVERITY:

```

/* Fit severity models for the magnitude of losses */
proc severity data=losses plots=none outest=work.sevregest print=all;
  loss lossamount;
  scalemodel carType carSafety income;
  dist _predef_;
  nloptions maxiter=100;
run;

```

The comparison of fit statistics of various scale regression models is shown in Figure 16.6. The scale regression model that is based on the lognormal distribution is deemed the best-fitting model according to the likelihood-based statistics, whereas the scale regression model that is based on the generalized Pareto distribution (GPD) is deemed the best-fitting model according to the EDF-based statistics.

Figure 16.6 Severity Model Comparison

The SEVERITY Procedure

All Fit Statistics								
Distribution	-2 Log Likelihood		AIC	AICC	BIC	KS	AD	CvM
Burr	127231	127243	127243	127286	7.75407	224.47578	27.41346	
Exp	128431	128439	128439	128467	6.13537	181.75649	12.33919	
Gamma	128324	128334	128334	128370	7.54562	275.83377	24.59515	
Igauss	127434	127444	127444	127480	6.15855	211.51200	17.70942	
Logn	127062	* 127072	* 127072	* 127107	* 6.77687	212.70400	21.47945	
Pareto	128166	128176	128176	128211	5.37453	110.53673	7.07119	
Gpd	128166	128176	128176	128211	5.37453	* 110.53660	* 7.07116	*
Weibull	128429	128439	128439	128475	6.21268	190.73733	13.45425	

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

Now, you are ready to analyze the distribution of the aggregate loss that can be expected from a specific policyholder—for example, a 59-year-old male policyholder with an advanced degree who earns 159,870 and drives a sedan that has a very high safety rating about 11,474 miles annually. First, you need to encode and scale this information into the appropriate regressor variables of a data set. Let that data set be named `Work.SinglePolicy`, with an observation as shown in Figure 16.7.

Figure 16.7 Scenario Analysis Data for One Policyholder

age	gender	carType	annualMiles	education	carSafety	income
1.18	2	1	2.2948	3	0.99532	1.5987

Now, you can submit the following PROC HPCDM step to analyze the compound distribution of the aggregate loss that is incurred by the policyholder in the `Work.SinglePolicy` data set in a given year by using the frequency model from the `Work.CountregModel` item store and the two best severity models, lognormal and GPD, from the `Work.SevRegEst` data set:

```

/* Simulate the aggregate loss distribution for the scenario
   with single policyholder */
proc hpcdm data=singlePolicy nreplicates=10000 seed=13579 print=all
  countstore=work.countregmodel severityest=work.sevregest;
  severitymodel logn gpd;
run;

```

```

outsum out=onepolicysum mean stddev skew kurtosis median
pctlpts=97.5 to 99.5 by 1;
run;

```

The displayed results from the preceding PROC HPCDM step are shown in [Figure 16.8](#).

When you use a severity scale regression model, it is recommended that you verify the severity scale regressors that are used by PROC HPCDM by examining the Scale Model Regressors row of the “Compound Distribution Information” table. PROC HPCDM detects the severity regressors automatically by examining the variables in the SEVERITYEST= and DATA= data sets. If those data sets contain variables that you did not include in the SCALEMODEL statement in PROC SEVERITY, then such variables can be treated as severity regressors. One common mistake that can lead to this situation is to fit a severity model by using the BY statement and forget to specify the identical BY statement in the PROC HPCDM step; this can cause PROC HPCDM to treat BY variables as scale model regressors. In this example, [Figure 16.8](#) confirms that the correct set of scale model regressors is detected.

Figure 16.8 Scenario Analysis Results for One Policyholder with Lognormal Severity Model

The HPCDM Procedure			
Severity Model: Logn			
Count Model: NegBin(p=2)			
Compound Distribution Information			
Severity Model	Lognormal Distribution		
Scale Model Regressors	carType carSafety income		
Count Model	NegBin(p=2) Model in Item Store WORK.COUNTREGMODEL		
Sample Summary Statistics			
Mean	214.05031	Median	0
Standard Deviation	436.27333	Interquartile Range	264.68948
Variance	190334.4	Minimum	0
Skewness	5.15057	Maximum	9005.2
Kurtosis	50.23372	Sample Size	10000
Sample Percentiles			
Percentile	Value		
1	0		
5	0		
25	0		
50	0		
75	264.68948		
95	950.03355		
97.5	1340.0		
98.5	1682.8		
99	1979.5		
99.5	2664.5		
Percentile Method = 5			

The “Sample Summary Statistics” and “Sample Percentiles” tables in [Figure 16.8](#) show estimates of the aggregate loss distribution for the lognormal severity model. The average expected loss is about 214, and the worst-case loss, if approximated by the 97.5th percentile, is about 1,340. The percentiles table shows that

the distribution is highly skewed to the right; this is also confirmed by the skewness estimate. The median estimate of 0 can be interpreted in two ways. One way is to conclude that the policyholder will not incur any loss in 50% of the years during which he or she is insured. The other way is to conclude that 50% of policyholders who have the characteristics of this policyholder will not incur any loss in a given year. However, there is a 2.5% chance that the policyholder will incur a loss that exceeds 1,340 in any given year and a 0.5% chance that the policyholder will incur a loss that exceeds 2,665 in any given year.

If the aggregate loss sample is simulated by using the GPD severity model, then the results are as shown in Figure 16.9. The average and worst-case losses are 213 and 1,337, respectively. These estimates are very close to the values that are predicted by the lognormal severity model.

Figure 16.9 Scenario Analysis Results for One Policyholder with GPD Severity Model

The HPCDM Procedure			
Severity Model: Gpd			
Count Model: NegBin(p=2)			
Compound Distribution Information			
Severity Model	Generalized Pareto Distribution		
Scale Model Regressors	carType carSafety income		
Count Model	NegBin(p=2) Model in Item Store WORK.COUNTREGMODEL		
Sample Summary Statistics			
Mean	212.54792	Median	0
Standard Deviation	401.95332	Interquartile Range	275.99091
Variance	161566.5	Minimum	0
Skewness	3.46433	Maximum	5360.2
Kurtosis	18.55938	Sample Size	10000
Sample Percentiles			
Percentile		Value	
1		0	
5		0	
25		0	
50		0	
75		275.99091	
95		977.06997	
97.5		1337.4	
98.5		1622.2	
99		1867.4	
99.5		2303.2	
Percentile Method = 5			

The scenario that you just analyzed contains only one policyholder. You can extend the scenario to include multiple policyholders. Let the Work.GroupOfPolicies data set record information about five different policyholders, as shown in Figure 16.10.

Figure 16.10 Scenario Analysis Data for Multiple Policyholders

policyholderid	age	gender	carType	annualMiles	education	carSafety	income
1	1.18	2	1	2.2948	3	0.99532	1.59870
2	0.66	2	1	2.6718	2	0.86412	0.84459
3	0.64	2	2	1.9528	1	0.86478	0.50177
4	0.46	1	2	2.6402	2	0.27062	1.18870
5	0.62	1	1	1.7294	1	0.32830	0.37694

The following PROC HPCDM step conducts a scenario analysis for the aggregate loss that is incurred by all five policyholders in the Work.GroupOfPolicies data set together in one year:

```

/* Simulate the aggregate loss distribution for the scenario
   with multiple policyholders */
proc hpcdm data=groupOfPolicies nreplicates=10000 seed=13579 print=all
    countstore=work.countregmodel severityest=work.sevregist
    plots=(conditionaldensity(rightq=0.95)) nperturbedSamples=50;
    severitymodel logn gpd;
    outsum out=multipolicysum mean stddev skew kurtosis median
    pctlpts=97.5 to 99.5 by 1;
run;

```

The preceding PROC HPCDM step conducts perturbation analysis by simulating 50 perturbed samples. The perturbation summary results for the lognormal severity model are shown in [Figure 16.11](#), and the results for the GPD severity model are shown in [Figure 16.12](#). If the severity of each loss follows the fitted lognormal distribution, then you can expect that the group of policyholders together incurs an average loss of $5,300 \pm 328$ and a worst-case loss of $15,734 \pm 960$ when you define the worst-case loss as the 97.5th percentile.

Figure 16.11 Perturbation Analysis of Losses from Multiple Policyholders with Lognormal Severity Model

Compound Distribution Information		
Severity Model	Lognormal Distribution	
Scale Model Regressors	carType carSafety income	
Count Model	NegBin(p=2) Model in Item Store WORK.COUNTREGMODEL	

Sample Perturbation Analysis		
Statistic	Estimate	Standard Error
Mean	5299.8	327.70569
Standard Deviation	4151.9	269.78790
Variance	17311274	2254196.7
Skewness	2.14414	1.24620
Kurtosis	16.65290	58.38318
Number of Perturbed Samples = 50		
Size of Each Sample = 10000		

Figure 16.11 *continued*

Sample Percentile Perturbation Analysis		
Percentile	Estimate	Standard Error
1	194.20187	28.77686
5	742.04381	59.84686
25	2379.0	154.80380
50	4324.3	272.87497
75	7113.4	438.24370
95	13101.5	805.58237
97.5	15734.1	960.35241
98.5	17746.7	1098.9
99	19384.7	1189.9
99.5	22409.7	1433.0
Number of Perturbed Samples = 50		
Size of Each Sample = 10000		

If the severity of each loss follows the fitted GPD distribution, then you can expect an average loss of $5,236 \pm 365$ and a worst-case loss of $14,992 \pm 1,014$.

If you decide to use the 99.5th percentile to define the worst-case loss, then the worst-case loss is $22,410 \pm 1,433$ for the lognormal severity model and $20,246 \pm 1,400$ for the GPD severity model. The numbers for lognormal and GPD are well within two standard errors of each other, which indicates that the aggregate loss distribution is less sensitive to the choice of these two severity distributions in this particular example; you can use the results from either of them.

Figure 16.12 Perturbation Analysis of Losses from Multiple Policyholders with GPD Severity Model

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

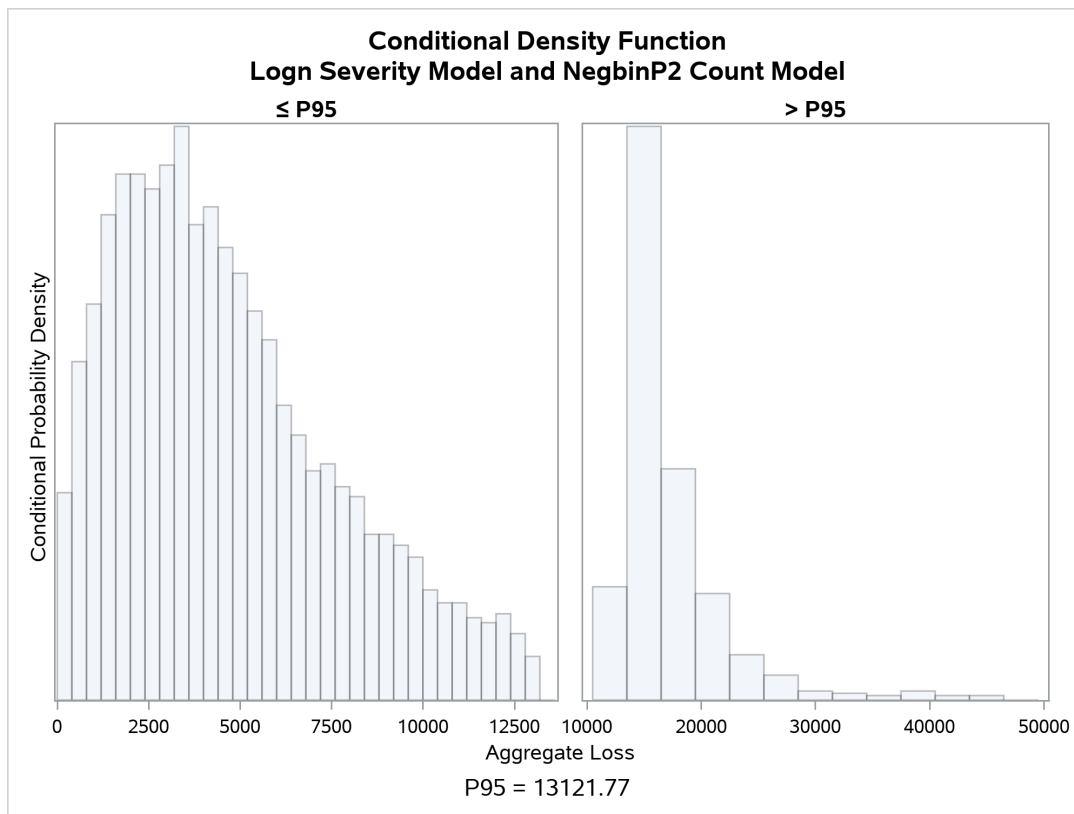
Compound Distribution Information		
Severity Model	Generalized Pareto Distribution	
Scale Model Regressors	carType carSafety income	
Count Model	NegBin(p=2) Model in Item Store WORK.COUNTREGMODEL	

Sample Perturbation Analysis		
Statistic	Estimate	Standard Error
Mean	5235.5	364.77905
Standard Deviation	3894.0	270.62630
Variance	15236520	2107602.2
Skewness	1.48825	0.24040
Kurtosis	4.33915	6.27802
Number of Perturbed Samples = 50		
Size of Each Sample = 10000		

Figure 16.12 *continued*

Sample Percentile Perturbation Analysis		
Percentile	Estimate	Standard Error
1	155.29557	25.93762
5	699.37268	62.80951
25	2381.4	173.33561
50	4367.2	308.51028
75	7136.8	498.42048
95	12717.7	883.48043
97.5	14991.8	1014.0
98.5	16657.1	1148.8
99	17993.5	1235.1
99.5	20246.2	1399.7
Number of Perturbed Samples = 50		
Size of Each Sample = 10000		

The PLOTS=CONDITIONALDENSITY option that is used in the preceding PROC HPCDM step prepares the conditional density plots for the body and right-tail regions of the density function of the aggregate loss. The plots for the aggregate loss sample that is generated by using the lognormal severity model are shown in [Figure 16.13](#). The plot on the left side is the plot of $\Pr(Y|Y \leq 13,122)$, where the limit 13,122 is the 95th percentile as specified by the RIGHTQ=0.95 option. The plot on the right side is the plot of $\Pr(Y|Y > 13,122)$, which helps you visualize the right-tail region of the density function. You can also request the plot of the left tail by specifying the LEFTQ= suboption of the CONDITIONALDENSITY option if you want to explore the details of the left tail region. Note that the conditional density plots are always produced by using the unperturbed sample.

Figure 16.13 Conditional Density Plots for the Aggregate Loss of Multiple Policyholders

Syntax: HPCDM Procedure

The following statements are available in the HPCDM procedure:

```

PROC HPCDM options ;
  BY variable-list ;
  DISTBY replication-id-variable ;
  SEVERITYMODEL severity-model-list ;
  EXTERNALCOUNTS COUNT=frequency-variable < ID=replication-id-variable > ;
  OUTPUT OUT=SAS-data-set < variable-name-options > < / out-option > ;
  OUTSUM OUT=SAS-data-set statistic-keyword < =variable-name > < . . . statistic-keyword < =variable-name > > < outsum-options > ;
  PERFORMANCE options ;
  Programming statements ;

```

Functional Summary

Table 16.1 summarizes the statements and options available in the HPCDM procedure.

Table 16.1 PROC HPCDM Functional Summary

Description	Statement	Option
Statements		
Specifies the names of severity distribution models	SEVERITYMODEL	
Specifies externally simulated count data	EXTERNALCOUNTS	
Specifies where and how the full simulated samples are written	OUTPUT	
Specifies where and how the summary statistics of simulated samples are written	OUTSUM	
Specifies performance options	PERFORMANCE	
Specifies programming statements that define an objective function	Programming statements	
Data Set Options		
Specifies the input data set	PROC HPCDM	DATA=
Specifies the output data set for the full simulated samples	OUTPUT	OUT=
Specifies the output data set for the summary statistics	OUTSUM	OUT=
Model Input Options		
Specifies the variable that contains externally simulated counts	EXTERNALCOUNTS	COUNT=
Specifies the item store that contains the frequency (count) model	PROC HPCDM	COUNTSTORE=
Specifies the replicate identifier variable for external counts	EXTERNALCOUNTS	ID=
Specifies the input data set for parameter estimates of the severity models	PROC HPCDM	SEVERITYEST=
Specifies the item store for parameter estimates of the severity models	PROC HPCDM	SEVERITYSTORE=
Simulation Options		
Specifies the adjusted severity symbol in the programming statements	PROC HPCDM	ADJUSTEDSEVERITY=
Specifies the upper limit on the count for each sample point	PROC HPCDM	MAXCOUNTDRAW=
Specifies the number of parameter-perturbed samples to be simulated	PROC HPCDM	NPERTURBEDSAMPLES=
Specifies a number that controls the size of the simulated sample	PROC HPCDM	NREPLICATES=
Specifies the parameter perturbation method	PROC HPCDM	PERTURBMETHOD=
Specifies a seed for the internal pseudorandom number generator	PROC HPCDM	SEED=

Table 16.1 *continued*

Description	Statement	Option
Output Preparation Options		
Specifies the variable for the aggregate adjusted loss sample	OUTPUT	ADJSAMPLEVAR=
Specifies the method to compute the percentiles	PROC HPCDM	PCTLDEF=
Specifies the names of the variables for percentiles	OUTSUM	PCTLNAME=
Specifies the decimal precision to form default percentile variable names	OUTSUM	PCTLNDEC=
Specifies percentiles to compute and report	OUTSUM	PCTLPTS=
Specifies that all perturbed samples be written to the OUT= data set	OUTPUT	PERTURBOUT
Specifies the variable for the aggregate loss sample	OUTPUT	SAMPLEVAR=
Specifies the denominator for computing second- and higher-order moments	PROC HPCDM	VARDEF=
Displayed Output and Plotting Options		
Suppresses all displayed and graphical output	PROC HPCDM	NOPRINT
Specifies which graphical output to prepare	PROC HPCDM	PLOTS=
Specifies which displayed output to prepare	PROC HPCDM	PRINT=

PROC HPCDM Statement

PROC HPCDM *options* ;

The PROC HPCDM statement invokes the procedure. You can specify the following *options*, which are listed in alphabetical order.

ADJUSTEDSEVERITY=*symbol-name*

ADJSEV=*symbol-name*

names the symbol that represents the adjusted severity value in the SAS programming statements that you specify. The *symbol-name* is a SAS name that conforms to the naming conventions of a SAS variable. For more information, see the section “[Programming Statements](#)” on page 956.

COUNTSTORE=*SAS-item-store*

names the item store that contains all the information about the frequency (count) model. The COUNTREG procedure generates this item store when you use the STORE statement.

The exogenous variables in the frequency model, if any, are deduced from this item store. The DATA= data set must contain all those variables.

If you specify a BY statement in the PROC COUNTREG step that creates the COUNTSTORE= item store, then you must specify an identical BY statement in the PROC HPCDM step.

You must specify this option if you do not specify the EXTERNALCOUNTS statement. This option is ignored if you specify the EXTERNALCOUNTS statement, because PROC HPCDM does not need to simulate frequency counts internally when you specify externally simulated counts.

If you specify the COUNTSTORE= option and execute the HPCDM procedure in distributed mode, then the distributed data access mode for the DATA= data set must be either client-data (local-data) mode or through-the-client mode—that is, the DATA= data set should not be stored on a distributed database appliance. For more information about data access modes, see the section “Data Access Modes” (Chapter 2, *SAS/ETS User’s Guide: High-Performance Procedures*).

DATA=SAS-data-set

names the input data set that contains the values of regression variables in frequency or severity models and severity adjustment variables that you use in the programming statements.

The DATA= data set specifies information about the scenario for which you want to estimate the aggregate loss distribution. The interpretation of the contents of the data set and the supported distributed data access modes depend on whether you specify the EXTERNALCOUNTS statement. For more information, see the section “[Specifying Scenario Data in the DATA= Data Set](#)” on page 957.

MAXCOUNTDRAW=number

MAXCOUNT=number

specifies an upper limit on the number of loss events (count) that is used for simulating one aggregate loss sample point. If the *number* is equal to N_{\max} , then any count that is greater than N_{\max} is assumed to be equal to N_{\max} , and only N_{\max} severity draws are made to compute one point in the aggregate loss sample.

If you specify this option and also specify the COUNTSTORE= item store, then the limit is applied to each count that PROC HPCDM randomly draws from the count distribution in the COUNTSTORE= item store. Any count draw that is larger than the *number* is replaced by the *number*.

If you specify this option and also specify the EXTERNALCOUNTS statement, then the limit is applied to each observation in the DATA= data set, and any value of the COUNT= variable that is larger than the *number* is replaced by the *number*.

If you do not specify this option, then a default value of 1,000 is used.

If you specify a *number* that is significantly larger than 1,000, then PROC HPCDM might take a very long time to complete the simulation, especially when some counts are closer to the limit.

NOPRINT

turns off all displayed and graphical output. If you specify this option, then PROC HPCDM ignores any value that you specify for the PRINT= or PLOTS= option.

NPERTURBEDSAMPLES=number

NPERTURB=number

requests that parameter perturbation analysis be conducted. The model parameters are perturbed the specified *number* of times and a separate full sample is simulated for each set of perturbed parameter values. The summary statistics and percentiles are computed for each such perturbed sample, and their values are aggregated across the samples to compute the mean and standard deviation of each summary statistic and percentile.

The parameter perturbation procedure makes random draws of parameter values from a multivariate normal distribution if the covariance estimates of the parameters are available. For the multivariate normal

distribution of severity model parameters, PROC HPCDM attempts to read the covariance estimates from the SEVERITYEST= data set or the SEVERITYSTORE= item store. For the multivariate normal distribution of count model parameters, PROC HPCDM attempts to read the covariance estimates from the COUNTSTORE= store. If covariance estimates are not available or valid, then for each parameter, a random draw is made from the univariate normal distribution that has mean and standard deviation equal to the point estimate and the standard error, respectively, of that parameter. If neither covariance nor standard error estimates are available, then perturbation analysis is not conducted.

If you specify the PRINT=ALL or PRINT=PERTURBSUMMARY option, then the summary of perturbation analysis is printed for the core summary statistics and the percentiles of the aggregate loss distribution. If you specify the OUTSUM statement, then the requested summary statistics are written to the OUTSUM= data set for each perturbed sample. You can also optionally request that each perturbed sample be written in its entirety to the OUT= data set by specifying the PERTURBOUT option in the OUTPUT statement.

For more information on the parameter perturbation analysis, see the section “[Parameter Perturbation Analysis](#)” on page 972.

NREPLICATES=*number*

NREP=*number*

specifies a *number* that controls the size of the compound distribution sample that PROC HPCDM simulates. The *number* is interpreted differently based on whether you specify the EXTERNALCOUNTS statement.

If you do not specify the EXTERNALCOUNTS statement, then the sample size is equal to the *number* that you specify for this option. If you do not specify this option, then a default value of 100,000 is used.

If you specify the EXTERNALCOUNTS statement, then the number of replicates that you specify in the DATA= data set is multiplied by the *number* that you specify for this option to get the total size of the compound distribution sample. If you do not specify this option, then a default value of 1 is used.

PCTLDEF=*percentile-method*

specifies the method to compute the percentiles of the compound distribution. The *percentile-method* can be 1, 2, 3, 4, or 5. The default method is 5. For more information, see the description of the PCTLDEF= option in the UNIVARIATE procedure in the *Base SAS Procedures Guide: Statistical Procedures*.

PERTURBMETHOD=*perturbation-method*

specifies a method to perturb the parameters of the severity and count models. This option has no effect if you do not specify the [NPERTURBEDSAMPLES=](#) option. You can specify one of the following *perturbation-methods*:

ASYNCR | 0

causes each thread of computation on each grid node to use its own set of perturbed model parameters. In particular, each thread uses its own pseudorandom number generator (PRNG) to perturb the model parameters, which is the same PRNG as the PRNG that the thread uses for making random draws from the severity or count distributions. Because each thread's PRNG starts with a different seed and because random draws that pertain to perturbation are interleaved with random draws that are made from the severity or count distributions, each thread effectively

uses a different set of perturbed models parameters even though it is simulating a subset of the same, global perturbed sample.

This method is computationally slightly more efficient because it does not need to synchronize the set of perturbed parameters among threads and grid nodes. However, each perturbed sample that it produces is conceptually a collection of smaller, distinct perturbed samples that belong to different compound distribution models.

SYNC | 1

specifies that all threads of computation on all grid nodes use the same (synchronized) set of perturbed model parameters. When you specify this option, PROC HPCDM in concept uses a single, dedicated PRNG to perturb the model parameters and shares those parameters with all threads on all grid nodes.

This method ensures that all observations of a particular perturbed sample belong to the same compound distribution model, because each thread on each grid node uses the same set of perturbed model parameters.

It is recommended that you specify the SYNC method. By default, PERTURBMETHOD=ASYN to ensure that the current release of PROC HPCDM produces, by default, the same perturbation results as releases prior to SAS/ETS 15.1.

PLOTS <(global-plot-options)> =plot-request-option

PLOTS <(global-plot-options)> =(plot-request-option . . . plot-request-option)

specifies the desired graphical output.

By default, the HPCDM procedure produces no graphical output.

You can specify the following *global-plot-option*:

ONLY

turns off the default graphical output and prepares only the requested plots.

If you specify more than one *plot-request-option*, then separate them with spaces and enclose them in parentheses. The following *plot-request-options* are available:

ALL

displays all the graphical output.

CONDITIONALDENSITY (conditional-density-plot-options)

CONDPDF (conditional-density-plot-options)

prepares a group of plots of the conditional density functions estimates. The group contains at most three plots, each conditional on the value of the aggregate loss being in one of the three regions that are defined by the quantiles that you specify in the following *conditional-density-plot-options*:

LEFTQ=number

specifies the quantile in the range (0,1) that marks the end of the left-tail region. If you specify a value of l for *number*, then the left-tail region is defined as the set of values that are less than or equal to q_l , where q_l is the l th quantile. For the left-tail region, nonparametric estimates of the conditional probability density function $f_S^l(s) = \Pr[S = s | S \leq q_l]$ are

plotted. The value of q_l is estimated by the 100/ l th percentile of the simulated compound distribution sample.

If you do not specify this option or you specify a missing value for this option, then the left-tail region is not plotted.

RIGHTQ=number

specifies the quantile in the range (0,1) that marks the beginning of the right-tail region. If you specify a value of r for *number*, then the right-tail region is defined as the set of values that are greater than q_r , where q_r is the r th quantile. For the right-tail region, nonparametric estimates of the conditional probability density function $f_S^r(s) = \Pr[S = s | S > q_r]$ are plotted. The value of q_r is estimated by the 100 r th percentile of the simulated compound distribution sample.

If you do not specify this option or you specify a missing value for this option, then the right-tail region is not plotted.

You must specify nonmissing value for at least one of the preceding two options. For the region between the LEFTQ= and RIGHTQ= quantiles, which is referred to as the central or body region, nonparametric estimates of the conditional probability density function $f_S^c(s) = \Pr[S = s | q_l < S \leq q_r]$ are plotted. If you do not specify a LEFTQ= value, then q_l is assumed to be 0. If you do not specify a RIGHTQ= value, then q_r is assumed to be ∞ .

DENSITY

prepares a plot of the nonparametric estimates of the probability density function (in particular, histogram and kernel density estimates) of the compound distribution.

EDF <(edf-plot-option)>

prepares a plot of the nonparametric estimates of the cumulative distribution function of the compound distribution.

You can request that the confidence interval be plotted by specifying the following *edf-plot-option*:

ALPHA=number

specifies the confidence level in the (0,1) range that is used for computing the confidence intervals for the EDF estimates. If you specify a value of α for *number*, then the upper and lower confidence limits for the confidence level of $100(1 - \alpha)$ are plotted.

NONE

displays none of the graphical output. If you specify this option, then it overrides all other plot request options. The default graphical output is also suppressed.

Note that if the simulated sample size is large, then it can take a significant amount of time and memory to prepare the plots.

PRINT <(global-display-option)> =display-option

PRINT <(global-display-option)> =(display-option ... display-option)

specifies the desired displayed output. If you specify more than one *display-option*, then separate them with spaces and enclose them in parentheses.

You can specify the following *global-display-option*:

ONLY

turns off the default displayed output and displays only the requested output.

You can specify the following *display-options*:

ALL

displays all the output.

NONE

displays none of the output. If you specify this option, then it overrides all other display options. The default displayed output is also suppressed.

PERCENTILES

displays the percentiles of the compound distribution sample. This includes all the predefined percentiles, percentiles that you request in the OUTSUM statement, and percentiles that you specify for preparing conditional density plots.

PERTURBSUMMARY

displays the mean and standard deviation of the summary statistics and percentiles that are taken across all the samples produced by perturbing the model parameters. This option is valid only if you specify the NPerturbedSamples= option in the PROC HPCDM statement.

SUMMARYSTATISTICS | SUMSTAT

displays the summary statistics of the compound distribution sample.

If you do not specify the PRINT= option or the ONLY *global-display-option*, then the default displayed output is equivalent to specifying PRINT=(SUMMARYSTATISTICS).

SEED=number

specifies the integer to use as the seed in generating the pseudorandom numbers that are used for simulating severity and frequency values.

If you do not specify the seed or if *number* is negative or 0, then the time of day from the computer's clock is used as the seed.

SEVERITYEST=SAS-data-set

names the input data set that contains the parameter estimates for the severity model. The format of this data set must be the same as the OUTEST= data set that is produced by the SEVERITY procedure.

The names of the regression variables in the scale regression model, if any, are deduced from this data set. In particular, PROC HPCDM assumes that all the variables in the SEVERITYEST= data set that do not appear in the following list are scale regression variables:

- BY variables
- _MODEL_, _TYPE_, _NAME_, and _STATUS_ variables
- variables that represent distribution parameters

The DATA= data set must contain all the regressors in the scale regression model.

To ensure that PROC HPCDM correctly matches the values of regressors and the values of regression parameter estimates, you might need to rename the regressors in the DATA= data set so that their

names match the names of the regressors that you specify in the SCALEMODEL statement of the PROC SEVERITY step that fits the severity model.

If you specify a BY statement in the PROC SEVERITY step that creates the SEVERITYEST= data set, then you must specify an identical BY statement in the PROC HPCDM step. Otherwise, PROC HPCDM detects the BY variables as regression variables in the scale regression model, which might produce unexpected results.

SEVERITYSTORE=SAS-item-store

SEVSTORE=SAS-item-store

specifies the item store that contains the context and estimates of the severity model. A PROC SEVERITY step with the OUTSTORE= option creates this item store.

If your severity model contains classification or interaction effects, then you need to use this option instead of the SEVERITYEST= option to specify the severity model. If you specify this option, you cannot specify the SEVERITYEST= option.

If you specify a BY statement in the PROC SEVERITY step that creates the SEVERITYSTORE= item store, then you must specify an identical BY statement in the PROC HPCDM step.

VARDEF=divisor

specifies the divisor to use in the calculation of variance, standard deviation, kurtosis, and skewness of the compound distribution sample. If the sample size is N , then you can specify one of the following values for the *divisor*:

DF

sets the divisor for variance to $N - 1$. This is the default. This also changes the definitions of skewness and kurtosis.

N

sets the divisor to N .

For more information, see the section “[Descriptive Statistics](#)” on page 973.

BY Statement

BY *variable-list* ;

You can use the BY statement in the HPCDM procedure to process the input data set in groups of observations defined by the BY variables.

If you specify the BY statement, then you must specify the DATA= option in order to specify the input data set. PROC HPCDM expects the input data set to be sorted in the order of the BY variables unless you specify the NOTSORTED option.

The BY statement is always supported in the single-machine mode of execution. For the distributed mode, it is supported only when the DATA= data set resides on the client machine. In other words, the BY statement is supported only in the client-data (or local-data) mode of the distributed computing model and not for any of the alongside modes, such as the alongside-the-database or alongside HDFS mode.

DISTBY Statement

DISTBY *replication-id-variable* ;

A DISTBY statement is necessary if and only if you specify an ID= variable in the EXTERNALCOUNTS statement. In fact, the *replication-id-variable* must be the same as the ID= variable. This is especially important in the distributed mode of execution, because when the observations in the DATA= data set are distributed to the grid nodes, by specifying the *replication-id-variable* as a DISTBY variable, you are instructing PROC HPCDM to make sure that the observations that have the same value for the *replication-id-variable* are always kept together on one grid node. This is required for correct simulation of the CDM in the presence of the ID= variable.

Contrast this to the BY variables that you specify in the BY statement. The observations of a BY group might be split across all the nodes of the grid, but the observations of a DISTBY group, which is nested within a BY group, are never split across the nodes of the grid.

The *replication-id-variable* must not appear in the BY statement. However, the DATA= data set must be sorted as if the *replication-id-variable* were listed after the BY variables in the BY statement.

Even though the DISTBY statement is important primarily in distributed mode, you must also specify it in single-machine mode.

EXTERNALCOUNTS Statement

EXTERNALCOUNTS **COUNT=***frequency-variable* < **ID=***replication-id-variable* > ;

The EXTERNALCOUNTS statement enables you to specify externally simulated frequency counts. By default, PROC HPCDM internally simulates the number of loss events by using the frequency model input (COUNTSTORE= item store). However, if you specify the EXTERNALCOUNTS statement, then PROC HPCDM uses the counts that you specify in the DATA= data set and simulates only the severity values internally.

If you specify more than one EXTERNALCOUNTS statement, only the first one is used.

You must specify the following option in the EXTERNALCOUNTS statement:

COUNT=*count-variable*

specifies the variable that contains the simulated counts. This variable must be present in the DATA= data set.

You can also specify the following option in the EXTERNALCOUNTS statement:

ID=*replication-id-variable*

specifies the variable that contains the replicate identifier. This variable must be present in the DATA= data set. Furthermore, you must specify the DISTBY statement with *replication-id-variable* as the only DISTBY variable to ensure correct simulation.

The observations of DATA= data set must be arranged such that the values of the ID= variable are in increasing order in each BY group or in the entire data set if you do not specify the BY statement.

If you do not specify the ID= option, then PROC HPCDM assumes that each observation represents one replication. In other words, the observation number serves as the default replication identifier.

The simulation process of using the external counts to generate the compound distribution sample is described in the section “[Simulation with External Counts](#)” on page 961.

OUTPUT Statement

OUTPUT **OUT=***SAS-data-set* < *variable-name-options* > < / *out-option* > ;

The OUTPUT statement enables you to specify the data set to output the generated compound distribution sample.

If you specify more than one OUTPUT statement, only the first one is used.

You must specify the output data set by using the following option:

OUT=*SAS-data-set*

OUTSAMPLE=*SAS-data-set*

specifies the output data set to contain the simulated compound distribution sample. If you specify programming statements to adjust individual severity values, then this data set contains both unadjusted and adjusted samples.

You can specify the following *variable-name-options* to control the names of the variables created in the OUT= data set:

ADJSAMPLEVAR=*variable-name*

specifies the name of the variable to contain the adjusted compound distribution sample in the OUT= data set. If you do not specify the ADJSAMPLEVAR= option, then “_AGGADJSEV_” is used by default.

This option is ignored if you do not specify the [ADJUSTEDSEVERITY=](#) option and the programming statements to adjust the simulated severity values.

SAMPLEVAR=*variable-name*

specifies the name of the variable to contain the simulated sample in the OUT= data set. If you do not specify the SAMPLEVAR= option, then “_AGGSEV_” is used by default.

Further, you can request that the perturbed samples be written to the OUT= data set by specifying the following *out-option*:

PERTURBOUT

specifies that all the perturbed samples be written to the OUT= data set. Each perturbed sample is identified by the _DRAWID_ variable in the OUT= data set. A value of 0 for the _DRAWID_ variable indicates an unperturbed sample.

Separate compound distribution samples are generated for each combination of specified severity and frequency models. The _SEVERITYMODEL_ and _COUNTMODEL_ columns in the OUT= data set identify the severity and frequency models, respectively, that are used to generate the sample in the SAMPLEVAR= and ADJSAMPLEVAR= variables.

OUTSUM Statement

OUTSUM **OUT=SAS-data-set** *statistic-keyword* <=*variable-name*> <...*statistic-keyword* <=*variable-name*>> <*outsum-options*> ;

The OUTSUM statement enables you to specify the data set in which PROC HPCDM writes the summary statistics of the compound distribution samples.

If you specify more than one OUTSUM statement, only the first one is used.

You must specify the output data set by using the following option:

OUT=SAS-data-set

OUTSUM=SAS-data-set

specifies the output data set that contains the summary statistics of each of the simulated compound distribution samples. You can control the summary statistics that appear in this data set by specifying different *statistic-keywords* and *outsum-options*.

If you execute the HPCDM procedure in distributed mode, only the client-data (local-data) and through-the-client data access modes are supported for this data set. In other words, the libref that you specify for this data set should not point to a distributed database appliance. For more information about data access modes, see the section “Data Access Modes” (Chapter 2, *SAS/ETS User’s Guide: High-Performance Procedures*).

You can request that one or more predefined statistics of the compound distribution sample be written to the OUTSUM= data set. For each specification of the form *statistic-keyword* <=*variable-name*>, the statistic that is specified by the *statistic-keyword* is written to a variable named *variable-name*. If you do not specify the *variable-name*, then the statistic is written to a variable named *statistic-keyword*. You can specify the following *statistic-keywords*:

KURTOSIS

KURT

specifies the kurtosis of the compound distribution sample.

MEAN

specifies the mean of the compound distribution sample.

MEDIAN

Q2

P50

specifies the median (the 50th percentile) of the compound distribution sample.

P01

specifies the 1st percentile of the compound distribution sample.

P05

specifies the 5th percentile of the compound distribution sample.

P95

specifies the 95th percentile of the compound distribution sample.

P99

specifies the 99th percentile of the compound distribution sample.

P99_5**P995**

specifies the 99.5th percentile of the compound distribution sample.

Q1**P25**

specifies the lower or 1st quartile (the 25th percentile) of the compound distribution sample.

Q3**P75**

specifies the upper or 3rd quartile (the 75th percentile) of the compound distribution sample.

QRANGE

specifies the interquartile range (Q3–Q1) of the compound distribution sample.

SKEWNESS**SKEW**

specifies the skewness of the compound distribution sample.

STDDEV**STD**

specifies the standard deviation of the compound distribution sample.

All percentiles are computed by using the method that you specify for the **PCTLDEF=** option in the PROC HPCDM statement. You can also request additional percentiles to be reported in the OUTSUM= data set by specifying the following *outsum-options*:

PCTLPTS=percentile-list

specifies one or more percentiles that you want to be computed and written to the OUTSUM= data set. This option is useful if you need to request percentiles that are not available in the preceding list of *statistic-keyword* values. Each percentile value must belong to the (0,100) open interval. The *percentile-list* is a comma-separated list of numbers. You can also use a list notation of the form “< number1 > to < number2 > by < increment >”. For example, the following two options are equivalent:

```
pctlpts=10, 20, 99.6, 99.7, 99.8, 99.9
pctlpts=10, 20, 99.6 to 99.9 by 0.1
```

The name of the variable for a given percentile value is decided by the PCTLNAME= option.

PCTLNAME=percentile-variable-name-list

specifies the names of the variables that contain the estimates of the percentiles that you request by using the PCTLPTS= option.

If you do not specify the PCTLNAME= option, then each percentile value t in the list of values in the PCTLPTS= option is written to the variable named “Pt,” where the decimal point in t , if any, is replaced by an underscore.

The *percentile-variable-name-list* is a space-separated list of names. You can also use a shortcut notation of <prefix> m –<prefix> n for two integers m and n ($m < n$) to generate the following list of names: <prefix> m , <prefix> $m + 1$, ..., and <prefix> n . For example, the following two options are equivalent:

```
pctlname=p1 p2 pc5 pc6 pc7 pc8 pc9 pc10
pctlname=p1 p2 pc5-pc10
```

The name in j th position of the expanded name list of the PCTLNAME= option is used to create a variable for a percentile value in the j th position of the expanded value list of the PCTLPTS= option. If you specify k_n names in the PCTLNAME= option and k_v percentile values in the PCTLPTS= option, and if $k_n < k_v$, then the first k_n percentiles are written to the variables that you specify and the remaining $k_v - k_n$ percentiles are written to the variables that have the name of the form Pt , where t is the text representation of the percentile value that is formed by retaining at most PCTLNDEC= digits after the decimal point and replacing the decimal point with an underscore (‘_’). For example, assume you specify the options

```
pctlpts=10, 20, 99.3 to 99.5 by 0.1, 99.995
pctlname=pten ptwenty ninenine3-ninenine5
```

Then PROC HPCDM writes the 10th and 20th percentiles to pten and ptwenty variables, respectively; the 99.3rd through 99.5th percentiles to ninenine3, ninenine4, and ninenine5 variables, respectively; and the remaining 99.995th percentile to the P99_995 variable.

If a percentile value in the PCTLPTS= option matches a percentile value implied by one of the predefined percentile statistics and you specify the corresponding *statistic-keyword*, then the variable name that is implied by the *statistic-keyword*=*variable-name* specification takes precedence over the name that you specify in the PCTLNAME= option. For example, assume you specify the predefined percentile statistic of P95 as in the OUTSUM statement

```
outsum out=mypctls p95=ninetyfifth
      pctlpts=95 to 99 by 1 pctlname=pct95-pct99;
```

Then the 95th percentile is written to the ninetyfifth variable instead of the pct95 variable that the PCTLNAME= option implies.

PCTLNDEC=integer-value

specifies the maximum number of decimal places to use while creating the names of the variables for the percentile values in the PCTLPTS= option. The default value is 3. For example, for a percentile value of 99.9995, PROC HPCDM creates a variable named P99_999 by default, but if you specify PCTLNDEC=4, then the variable is named P99_9995.

The PCTLNDEC= option is used only for percentile values for which you do not specify a name in the PCTLNAME= option.

Note that all variable names in the OUTSUM= data set have a limit of 32 characters. If a name exceeds that limit, then it is truncated to contain only the first 32 characters. For more information about the variables in the OUTSUM= data set, see the section “[Output Data Sets](#)” on page 977.

PERFORMANCE Statement

PERFORMANCE *options* ;

The PERFORMANCE statement defines performance parameters for distributed and multithreaded computing, passes variables that describe the distributed computing environment, and requests detailed results about the performance characteristics of PROC HPCDM.

You can also use the PERFORMANCE statement to control whether a high-performance analytical procedure executes in single-machine or distributed mode.

For more information about the PERFORMANCE statement, see the section “PERFORMANCE Statement” (Chapter 2, *SAS/ETS User’s Guide: High-Performance Procedures*).

SEVERITYMODEL Statement

SEVERITYMODEL *severity-model-list* ;

The SEVERITYMODEL statement specifies one or more severity distribution models that you want to use in simulating a compound distribution sample. The *severity-model-list* is a space-separated list of names of severity models that you would use with PROC SEVERITY’s DIST statement. The [SEVERITYEST=](#) data set or the [SEVERITYSTORE=](#) item store must contain all the severity models in the list. If you specify the SEVERITYEST= data set and you specify a name that does not appear in the _MODEL_ column of the SEVERITYEST= data set, then that name is ignored. Similarly, if you specify the SEVERITYSTORE= item store and a severity model by that name does not appear in the item store, then that name is ignored.

If you specify more than one SEVERITYMODEL statement, only the first one is used.

If you do not specify a SEVERITYMODEL statement, then this is equivalent to specifying all the severity models that appear in the [SEVERITYEST=](#) data set or the [SEVERITYSTORE=](#) item store.

A compound distribution sample is generated for each of the severity models by compounding that severity model with the frequency model that you specify in the COUNTSTORE= item store or the external frequency model that is encoded by the COUNT= variable that you specify in the EXTERNALCOUNTS statement.

Programming Statements

In PROC HPCDM, you can use a series of programming statements that use variables in the DATA= data set to adjust an individual severity value. The adjusted severity values are aggregated to form a separate adjusted compound distribution sample.

The programming statements are executed for each simulated individual severity value. The observation of the input data set that is used to evaluate the programming statements is determined by the simulation procedure that is described in the section “[Simulation Procedure](#)” on page 958.

For more information, see the section “[Simulation of Adjusted Compound Distribution Sample](#)” on page 964.

Details: HPCDM Procedure

Specifying Scenario Data in the DATA= Data Set

A scenario represents a state of the world for which you want to estimate the distribution of aggregate losses. The state consists of one or more entities that generate the loss events. For example, an entity might be an individual who has an insurance policy or an organization that has a workers' compensation policy. Each entity has some characteristics of its own and some external factors that affect the frequency with which it generates the losses and the severity of each loss. For example, characteristics of an individual with an automobile insurance policy can include various demographics of the individual and various features of the automobile. Characteristics of an organization with a workers' compensation policy can be the number of employees, revenue, ratio of temporary to permanent employees, and so on. The organization can also be affected by external macroeconomic factors such as GDP and unemployment of the country where the organization operates and factors that affect its industry. You need to quantify and specify all these characteristics as external factors (regressors) when you fit severity and frequency models.

You should specify all the information about a scenario in the **DATA=** data set that you specify in the PROC HPCDM statement. Each observation in the **DATA=** data set encodes the characteristics of an entity. For proper simulation of severities, you must specify in the **DATA=** data set all the characteristics that you use as regressors in the severity scale regression models. When you use the **COUNTSTORE=** option to specify the frequency model, you must specify in the **DATA=** data set all the characteristics that you use as regressors in the frequency model in order to properly simulate the counts. All the regressors are expected to have nonmissing values. If any of the regressors have a missing value in an observation, then that observation is ignored.

The information in the **DATA=** data set is interpreted as follows, based on whether you specify the **EXTERNALCOUNTS** statement:

- If you do not specify the **EXTERNALCOUNTS** statement, then all the observations in the data set form a scenario. The observations are used together to compute one random draw from the compound distribution. The total number of draws is equal to the value that you specify in the **NREPLICATES=** option. The simulation process is described in the section “[Simulation with Regressors and No External Counts](#)” on page 959 and illustrated using an example in the section “[Illustration of Aggregate Loss Simulation Process](#)” on page 959.

In this case, the distributed data access mode for the **DATA=** data set must be either client-data (local-data) mode or through-the-client mode—that is, the **DATA=** data set should not be stored on a distributed appliance. For more information about data access modes, see the section “Data Access Modes” (Chapter 2, *SAS/ETS User's Guide: High-Performance Procedures*).

- If you specify the **EXTERNALCOUNTS** statement, then the **DATA=** data set is expected to contain multiple replications (draws) of the frequency counts that you simulate externally for a scenario. The **DATA=** data set must contain the **COUNT=** variable that you specify in the **EXTERNALCOUNTS** statement. The replications are identified by the observation number or the **ID=** variable that you specify in the **EXTERNALCOUNTS** statement. For each observation in a given replication, the **COUNT=** variable is expected to contain the count of losses that are generated by the entity associated with that observation. All the observations of a given replication are used together to compute one random

draw from the compound distribution. The size of the compound distribution sample is equal to the number of distinct replications that you specify in the `DATA=` data set, multiplied by the value that you specify in the `NREPLICATES=` option. The simulation process is described in the section “[Simulation with External Counts](#)” on page 961 and illustrated using an example in the section “[Illustration of the Simulation Process with External Counts](#)” on page 962.

In this case, the distributed data access mode for the `DATA=` data set can be any of the supported data access modes. For more information about data access modes, see the section “Data Access Modes” (Chapter 2, *SAS/ETS User’s Guide: High-Performance Procedures*).

In both cases, an observation can also contain severity adjustment variables that you can use to adjust the severity of the losses generated by that entity, based on some policy rules. For more information about simulating the adjusted compound distribution sample, see the section “[Simulation of Adjusted Compound Distribution Sample](#)” on page 964.

If you specify severity and frequency models that have no regression effects in them and if you do not specify externally simulated counts in the `EXTERNALCOUNTS` statement, then you do not need to specify the `DATA=` data set. This case corresponds to a fixed scenario that is represented entirely by the distribution parameters of the models.

Simulation Procedure

PROC HPCDM selects a simulation procedure based on whether you specify external counts or you request that PROC HPCDM simulate the counts, and whether the severity or frequency models contain regression effects. The following sections describe the process for the different scenarios.

Simulation with No Regressors and No External Counts

If you specify severity and frequency models that have no regression effects in them, and if you do not specify externally simulated counts in the `EXTERNALCOUNTS` statement, then PROC HPCDM uses the following simulation procedure.

The process is described for one severity distribution, *dist*. If you specify multiple severity distributions in the `SEVERITYMODEL` statement, then the process is repeated for each specified distribution.

The following steps are repeated M times to generate a compound distribution sample of size M , where M is the value that you specify in the `NREPLICATES=` option or the default value of that option:

1. Use the frequency model that you specify in the `COUNTSTORE=` option to draw a value N from the count distribution. N is the number of loss events that are expected to occur in the time period that is being simulated. N is adjusted to conform to the upper limit by setting it equal to $\min(N, N_{\max})$, where N_{\max} is either 1,000 or the value that you specify in the `MAXCOUNTDRAW=` option.
2. Draw N values, X_j ($j = 1, \dots, N$), from the severity distribution *dist* with parameters that you specify in the `SEVERITYEST=` data set or the `SEVERITYSTORE=` item store.
3. Add the N severity values that are drawn in step 2 to compute one point S from the compound distribution as

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

Note that although it is more common to fit the frequency model with regressors, PROC COUNTREG enables you to fit a frequency model without regressors. If you do not specify any regressors in the MODEL statement of the COUNTREG procedure, then it fits a model that contains only an intercept.

Simulation with Regressors and No External Counts

If the severity or frequency models have regression effects and if you do not specify externally simulated counts in the EXTERNALCOUNTS statement, then you must specify a DATA= data set to provide values of the regression variables, which together represent a scenario for which you want to simulate the CDM. In this case, PROC HPCDM uses the following simulation procedure.

The process is described for one severity distribution. If you specify multiple severity distributions in the SEVERITYMODEL statement, then the process is repeated for each specified distribution.

Note that you are doing scenario analysis when regression effects are present. Let K denote the number of observations that form the scenario. This is the number of observations either in the current BY group or in the entire DATA= data set if you do not specify the BY statement. If $K > 1$, then you are modeling the scenario for a group of entities. If $K = 1$, then you are modeling the scenario for one entity.

The following steps are repeated M times to generate a compound distribution sample of size M , where M is the value that you specify in the NREPLICATES= option or the default value of that option:

1. For each observation k ($k = 1, \dots, K$), a count N_k is drawn from the frequency model that you specify in the COUNTSTORE= option. The parameters of this model are determined by the frequency regressors in observation k . N_k represents the number of loss events that are expected to be generated by entity k in the time period that is being simulated. N_k is adjusted to conform to the upper limit by setting it equal to $\min(N_k, N_{\max})$, where N_{\max} is either 1,000 or the value that you specify in the MAXCOUNTDRAW= option.
2. Counts from all observations are added to compute $N = \sum_{k=1}^K N_k$. N is the total number of loss events that are expected to occur in the time period that is being simulated.
3. N number of random draws are made from the severity distribution, and they are added to generate one point of the compound distribution sample. Each of the N draws uses one of the K observations. If you specify a scale regression model for the severity distribution, then the scale parameter of the severity distribution is determined by the values of the severity regressors in the observation that is chosen for that draw.

If you specify the BY statement, then a separate sample of size M is created for each BY group in the DATA= data set.

Illustration of Aggregate Loss Simulation Process

As an illustration of the simulation process, consider a very simple example of analyzing the distribution of an aggregate loss that is incurred by a set of policyholders of an automobile insurance company in a period of one year. It is postulated that the frequency and severity distributions depend on three variables: Age, Gender (1: female, 2: male), and CarType (1: sedan, 2: sport utility vehicle). So these variables are used as regressors while you fit the count model and severity scale regression model by using the COUNTREG and SEVERITY procedures, respectively. Now, consider that you want to use the fitted frequency and severity models to estimate the distribution of the aggregate loss that is incurred by a set of five policyholders together. Let the characteristics of the five policyholders be encoded in a SAS data set named Work.Scenario that has the following contents:

Obs	age	gender	carType
1	30	2	1
2	25	1	2
3	45	2	2
4	33	1	1
5	50	1	1

The column Obs contains the observation number. It is shown only for the purpose of illustration. It need not be present in the data set. The following PROC HPCDM step simulates the scenario in the Work.Scenario data set:

```
proc hpcdm data=scenario
    severityest=<severity parameter estimates data set>
    countstore=<count model store> nreplicates=<sample size>;
    severitymodel <severity distribution name(s)>;
run;
```

The following process generates a sample from the aggregate loss distribution for the scenario in the Work.Scenario data set:

1. Use the values Age=30, Gender=2, and CarType=1 in the first observation to draw a count from the count distribution. Let that count be 2. Repeat the process for the remaining four observations. Let the counts be as shown in the Count column in the following table:

Obs	age	gender	carType	count
1	30	2	1	2
2	25	1	2	1
3	45	2	2	2
4	33	1	1	3
5	50	1	1	0

Note that the Count column is shown for illustration only; it is not added as a variable to the DATA= data set.

2. The simulated counts from all the observations are added to get a value of $N = 8$. This means that for this particular sample point, you expect a total of eight loss events in a year from these five policyholders.
3. For the first observation, the scale parameter of the severity distribution is computed by using the values Age=30, Gender=2, and CarType=1. That value of the scale parameter is used together with estimates of the other parameters from the SEVERITYEST= data set to make two draws from the severity distribution. Each of the draws simulates the magnitude of the loss that is expected from the first policyholder. The process is repeated for the remaining four policyholders. The fifth policyholder does not generate any loss event for this particular sample point, so no severity draws are made by using the fifth observation. Let the severity draws, rounded to integers for convenience, be as shown in the _SEV_ column in the following table:

Obs	age	gender	carType	count	_sev_
1	30	2	1	2	350 2100
2	25	1	2	1	4500

3	45	2	2	2	700	4300	
4	33	1	1	3	600	1500	950
5	50	1	1	0			

Note that the `_SEV_` column is shown for illustration only; it is not added as a variable to the `DATA=` data set.

PROC HPCDM adds the severity values of the eight draws to compute an aggregate loss value of 15,000. After recording this amount in the sample, the process returns to step 1 to compute the next point in the aggregate loss sample. For example, in the second iteration, the count distribution of each policyholder might generate one loss event for a total of five loss events, and the five severity draws from the severity distributions that govern each of the policyholders might add up to 5,000. Then, the value of 5,000 is recorded as the second point in the aggregate loss sample. The process continues until M aggregate loss sample points are simulated, where the M is the value that you specify in the `NREPLICATES=` option.

Simulation with External Counts

If you specify externally simulated counts by using the `EXTERNALCOUNTS` statement, then each replication in the input data set represents the loss events generated by an entity. An entity can be an individual or organization for which you want to estimate the compound distribution. If an entity has any characteristics that are used as external factors (regressors) in developing the severity scale regression model, then you must specify the values of those factors in the `DATA=` data set. If you specify the `ID=` variable, then multiple observations for the same replication ID represent different entities in a group for which you are simulating the CDM.

PROC HPCDM uses the following simulation procedure in the presence of externally simulated counts.

The process is described for one severity distribution. If you specify multiple severity distributions in the `SEVERITYMODEL` statement, then the process is repeated for each specified distribution.

Let there be M distinct replications in the current `BY` group of the `DATA=` data set or in the entire `DATA=` data set if you do not specify the `BY` statement. A replication is identified by either the observation number or the value of the `ID=` variable that you specify in the `EXTERNALCOUNTS` statement.

For each of the M values of the replication identifier, the following steps are executed R times, where R is the value of the `NREPLICATES=` option or the default value of that option:

1. Compute the total number of losses, N . If there are K ($K \geq 1$) observations for the current value of the replication identifier, then $N = \sum_{k=1}^K N_k$, where N_k is the value of the `COUNT=` variable for observation k , after it is adjusted to conform to the upper limit of either 1,000 or the value that you specify in the `MAXCOUNTDRAW=` option.
2. N number of random draws are made from the severity distribution, and they are added to generate one point of the compound distribution sample.

This process generates a compound distribution sample of size $M \times R$. If you specify the `BY` statement, then a separate sample of size $M \times R$ is created for each `BY` group in the `DATA=` data set.

Illustration of the Simulation Process with External Counts

In order to illustrate the simulation process, consider the following simple example. In this example, your severity model does not contain any regressors. An example that uses a severity scale regression model is illustrated later. Assume that you have made 10 random draws from an external count model and recorded them in the ExtCount variable of a SAS data set named Work.Counts1 as follows:

Obs	extCount
1	3
2	2
3	0
4	1
5	3
6	4
7	1
8	2
9	0
10	5

Because the data set does not contain an ID= variable, the observation number that is shown in the Obs column acts as the replicate identifier. The following PROC HPCDM step simulates an aggregate loss sample by using the Work.Counts1 data set:

```
proc hpcdm data=work.counts1 nreplicates=5
    severityest=<severity parameter estimates data set>;
    severitymodel <severity distribution name(s)>;
    externalcounts count=extCount;
run;
```

The simulation process works as follows:

1. For the first replication, which is associated with the first observation, three severity values are drawn from the severity distribution by using the parameter estimates that you specify in the SEVERITYEST= data set. If the severity values are 150, 500, and 320, then their sum of 970 is recorded as the first point of the aggregate loss sample. Because the value of the NREPLICATES= option is 5, this process of drawing three severity values and adding them to form a point of the aggregate loss sample is repeated four more times to generate a total of five sample points that correspond to the first observation.
2. For the second replication, two severity values are drawn from the severity distribution. If the severity values are 450 and 100, then their sum of 550 is recorded as a point of the aggregate loss sample. This process of drawing two severity values and adding them to form a point of the aggregate loss sample is repeated four more times to generate a total of five sample points that correspond to the second observation.
3. The process continues until all the replications, which are observations in this case, are exhausted.

The process results in an aggregate loss sample of size 50, which is equal to the number of replications in the data set (10) multiplied by the value of the NREPLICATES= option (5).

Now, consider an example in which the severity models in the SEVERITYEST= data set are scale regression models. In this case, the severity distribution that is used for drawing the severity value is decided by the values of regressors in the observation that is being processed. Consider that you want to simulate the aggregate loss that is incurred by one policyholder and you have recorded, in the ExtCount variable, the results of 10 random draws from an external count model. The DATA= data set has the following contents:

Obs	age	gender	carType	extCount
1	30	2	1	5
2	30	2	1	2
3	30	2	1	0
4	30	2	1	1
5	30	2	1	3
6	30	2	1	4
7	30	2	1	1
8	30	2	1	2
9	30	2	1	0
10	30	2	1	5

The simulation process in this case is the same as the process in the previous case of no regressors, except that the severity distribution that is used for drawing the severity values has a scale parameter that is determined by the values of the regressors Age, Gender, and CarType in the observation that is being processed. In this particular example, all observations have the same value for all regressors, indicating that you are modeling a scenario in which the characteristics of the policyholder do not change during the time for which you have simulated the number of events. You can also model a scenario in which the characteristics of the policyholder change by recording those changes in the values of the appropriate regressors.

Extending this example further, consider that you want to analyze the distribution of the aggregate loss that is incurred by a group of policyholders, as in the example in the section “[Illustration of Aggregate Loss Simulation Process](#)” on page 959. Let the Work.Counts2 data set record multiple replications of the number of losses that might be generated by each policyholder. The contents of the Work.Counts2 data set are as follows:

Obs	replicateId	age	gender	carType	extCount
1	1	30	2	1	2
2	1	25	1	2	1
3	1	45	2	2	3
4	1	33	1	1	5
5	1	50	1	1	1
6	2	30	2	1	3
7	2	25	1	2	2
8	1	45	2	2	0
9	2	33	1	1	4
10	2	50	1	1	1

The ReplicateId variable records the identifier for the replication. Each replication contains multiple observations, such that each observation represents one of the policyholders that you are analyzing. For simplicity, only the first two replications are shown here.

The following PROC HPCDM step simulates an aggregate loss sample by using the Work.Counts2 data set:

```
proc hpcdm data=work.counts2 nreplicates=3
    severityest=<severity parameter estimates data set>;
    severitymodel <severity distribution name(s)>;
    distby replicateId;
    externalcounts count=extCount id=replicateId;
    output out=aggloss samplevar=totalLoss;
run;
```

When you specify an `ID=` variable in the `EXTERNALCOUNTS` statement, you must specify the same `ID=` variable in the `DISTBY` statement in order for the procedure to work correctly in a distributed computing environment. Further, the `DATA=` set must be sorted in ascending order of the `ID=` variable values.

The simulation process works as follows:

1. First, the five observations of the first replication (`ReplicateId=1`) are analyzed. For the first observation (`Obs=1`), the scale parameter of the severity distribution is computed by using the values `Age=30`, `Gender=2`, and `CarType=1`. That value of the scale parameter is used together with estimates of the other parameters from the `SEVERITYEST=` data set to make two draws from the severity distribution. Next, the regressor values of the second observation are used to compute the scale parameter of the severity distribution, which is used to make one severity draw. The process continues such that the regressor values in the third, fourth, and fifth observations are used to decide the severity distribution to make three, five, and one draws from, respectively. Let the severity values that are drawn from the observations of this replication be as shown in the `_SEV_` column in the following table, where the `_SEV_` column is shown for illustration only; it is not added as a variable to the `DATA=` data set:

Obs	replicateId	age	gender	carType	extCount	_sev_
1	1	30	2	1	2	700 500
2	1	25	1	2	1	5000
3	1	45	2	2	3	900 1400 300
4	1	33	1	1	5	350 2000 150 800 600
5	1	50	1	1	1	250

The values of all 12 severity draws are added to compute and record the value of 12,950 as the first point of the aggregate loss sample. Because you specify `NREPLICATES=3` in the `PROC HPCDM` step, this process of making 12 severity draws from the respective observations is repeated two more times to generate a total of three sample points for the first replication.

2. The five observations of the second replication (`ReplicateId=2`) are analyzed next to draw three, two, four, and one severity values from the severity distributions, with scale parameters that are decided by the regressor values in the sixth, seventh, ninth, and tenth observations, respectively. The 10 severity values are added to form a point of the aggregate loss sample. This process of making 10 severity draws from the respective observations is repeated two more times to generate a total of three sample points for the second replication.

If your `Work.Counts2` data set contains 10,000 distinct values of `ReplicateId`, then 30,000 observations are written to the `Work.AgglLoss` data set that you specify in the `OUTPUT` statement of the preceding `PROC HPCDM` step. Because you specify `SAMPLEVAR=TotalLoss` in the `OUTPUT` statement, the aggregate loss sample is available in the `TotalLoss` column of the `Work.AgglLoss` data set.

Simulation of Adjusted Compound Distribution Sample

If you specify programming statements that adjust the severity value, then a separate adjusted compound distribution sample is also generated.

Your programming statements are expected to implement an adjustment function f that uses the unadjusted severity value, X_j , to compute and return an adjusted severity value, X_j^a . To compute X_j^a , you might also use the sum of unadjusted severity values and the sum of adjusted severity values.

Formally, if N denotes the number of loss events that are to be simulated for the current replication of the simulation process, then for the severity draw, X_j , of the j th loss event ($j = 1, \dots, N$), the adjusted severity value is

$$X_j^a = f(X_j, S_{j-1}, S_{j-1}^a)$$

where $S_{j-1} = \sum_{l=1}^{j-1} X_l$ is the aggregate unadjusted loss before X_j is generated and $S_{j-1}^a = \sum_{l=1}^{j-1} X_l^a$ is the aggregate adjusted loss before X_j is generated. The initial values of both types of aggregate losses are set to 0. In other words, $S_0 = 0$ and $S_0^a = 0$.

The aggregate adjusted loss for the replication is S_N^a , which is denoted by S^a for simplicity, and is defined as

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

In your programming statements that implement f , you can use the following keywords as placeholders for the input arguments of the function f :

SEV

indicates the placeholder for X_j , the unadjusted severity value. PROC HPCDM generates this value as described in the section “[Simulation with No Regressors and No External Counts](#)” on page 958 (step 2) or the section “[Simulation with Regressors and No External Counts](#)” on page 959 (step 3). PROC HPCDM supplies this value to your program.

CUMSEV

indicates the placeholder for S_{j-1} , the sum of unadjusted severity values that PROC HPCDM generates before X_j is generated. PROC HPCDM supplies this value to your program.

CUMADJSEV

indicates the placeholder for S_{j-1}^a , the sum of adjusted severity values that are computed by your programming statements before X_j is generated and adjusted. PROC HPCDM supplies this value to your program.

In your programming statements, you must assign the value of X_j^a , the output of function f , to a symbol that you specify in the **ADJUSTEDSEVERITY=** option in the PROC HPCDM statement. PROC HPCDM uses the final assigned value of this symbol as the value of X_j^a .

You can use most DATA step statements and functions in your program. The DATA step file and the data set I/O statements (for example, INPUT, FILE, SET, and MERGE) are not available. However, some functionality of the PUT statement is supported. For more information, see the section “PROC FCMP and DATA Step Differences” in *Base SAS Procedures Guide*.

The simulation process that generates the aggregate adjusted loss sample is identical to the process that is described in the section “[Simulation with Regressors and No External Counts](#)” on page 959 or the section “[Simulation with External Counts](#)” on page 961, except that after making each of the N severity draws, PROC HPCDM executes your severity adjustment programming statements to compute the adjusted severity (X_j^a). All the N adjusted severity values are added to compute S^a , which forms a point of the aggregate adjusted loss sample. The process is illustrated using an example in the section “[Illustration of Aggregate Adjusted Loss Simulation Process](#)” on page 967.

Using Severity Adjustment Variables

If you do not specify the DATA= data set, then your ability to adjust the severity value is limited, because you can use only the current severity draw, sums of unadjusted and adjusted severity draws that are made before the current draw, and some constant numbers to encode your adjustment policy. That is sufficient if you want to estimate the distribution of aggregate adjusted loss for only one entity. However, if you are simulating a scenario that contains more than one entity, then it might be more useful if the adjustment policy depends on factors that are specific to each entity that you are simulating. To do that, you must specify the DATA= data set and encode such factors as *adjustment variables* in the DATA= data set. Let A denote the set of values of the adjustment variables. Then, the form of the adjustment function f that computes the adjusted severity value becomes

$$X_j^a = f(X_j, S_{j-1}, S_{j-1}^a, A)$$

PROC HPCDM reads the values of adjustment variables from the DATA= data set and supplies the set of those values (A) to your severity adjustment program. For an invocation of f with an unadjusted severity value of X_j , the values in set A are read from the same observation that is used to simulate X_j .

All adjustment variables that you use in your program must be present in the DATA= data set. You must not use any keyword for a placeholder symbol as a name of any variable in the DATA= data set, whether the variable is a severity adjustment variable or a regressor in the frequency or severity model. Further, the following restrictions apply to the adjustment variables:

- You can use only numeric-valued variables in PROC HPCDM programming statements. This restriction also implies that you cannot use SAS functions or call routines that require character-valued arguments, unless you pass those arguments as constant (literal) strings or characters.
- You cannot use functions that create lagged versions of a variable in PROC HPCDM programming statements. If you need lagged versions, then you can use a DATA step before the PROC HPCDM step to add those versions to the input data set.

The use of adjustment variables is illustrated using an example in the section “[Illustration of Aggregate Adjusted Loss Simulation Process](#)” on page 967.

Aggregate Adjusted Loss Simulation for a Multi-entity Scenario

If you are simulating a scenario that consists of multiple entities, then you can use some additional pieces of information in your severity adjustment program. Let the scenario consist of K entities and let N_k denote the number of loss events that are incurred by k th entity ($k = 1, \dots, K$) in the current iteration of the simulation process. Each value of N_k is adjusted to conform to the upper limit of either 1,000 or the value that you specify in the MAXCOUNTDRAW= option. The total number of severity draws that need to be made is $N = \sum_{k=1}^K N_k$. The aggregate adjusted loss is now defined as

$$S^a = \sum_{k=1}^K \sum_{j=1}^{N_k} X_{k,j}^a$$

where $X_{k,j}^a$ is an adjusted severity value of the j th draw ($j = 1, \dots, N_k$) for the k th entity, and the form of the adjustment function f that computes $X_{k,j}^a$ is

$$X_{k,j}^a = f(X_{k,j}, S_{k,j-1}, S_{k,j-1}^a, S_{n-1}, S_{n-1}^a, A)$$

where $X_{k,j}$ is the value of the j th draw of unadjusted severity for the k th entity. $S_{k,j-1} = \sum_{l=1}^{j-1} X_{k,l}$ and $S_{k,j-1}^a = \sum_{l=1}^{j-1} X_{k,l}^a$ are the aggregate unadjusted loss and the aggregate adjusted loss, respectively, for the k th entity before $X_{k,j}$ is generated. The index n ($n = 1, \dots, N$) keeps track of the total number of severity draws, across all entities, that are made before $X_{k,j}$ is generated. So $S_{n-1} = \sum_{l=1}^{n-1} X_l$ and $S_{n-1}^a = \sum_{l=1}^{n-1} X_l^a$ are the aggregate unadjusted loss and aggregate adjusted loss, respectively, for all the entities that are processed before $X_{k,j}$ is generated. Note that S_{n-1} and S_{n-1}^a include the $j-1$ draws that are made for the k th entity before $X_{k,j}$ is generated.

The initial values of all types of aggregate losses are set to 0. In other words, $S_0 = 0$, $S_0^a = 0$, and for all values of k , $S_{k,0} = 0$ and $S_{k,0}^a = 0$.

PROC HPCDM uses the final value that you assign to the `ADJUSTEDSEVERITY=` symbol in your programming statements as the value of $X_{k,j}^a$.

In your severity adjustment program, you can use the following two additional placeholder keywords:

CUMSEVFOROBS

indicates the placeholder for $S_{k,j-1}$, which is the total loss that is incurred by the k th entity before the current loss event. PROC HPCDM supplies this value to your program.

CUMADJSEVFOROBS

indicates the placeholder for $S_{k,j-1}^a$, which is the total adjusted loss that is incurred by the k th entity before the current loss event. PROC HPCDM supplies this value to your program.

The previously described placeholder symbols `_CUMSEV_` and `_CUMADJSEV_` represent S_{n-1} and S_{n-1}^a , respectively. If you have only one entity in the scenario ($K = 1$), then the values of `_CUMSEVFOROBS_` and `_CUMADJSEVFOROBS_` are identical to the values of `_CUMSEV_` and `_CUMADJSEV_`, respectively.

There is one caveat when a scenario consists of more than one entity ($K > 1$) and when you use any of the symbols for cumulative severity values (`_CUMSEV_`, `_CUMADJSEV_`, `_CUMSEVFOROBS_`, or `_CUMADJSEVFOROBS_`) in your programming statements. In this case, to make the simulation realistic, it is important to randomize the order of N severity draws across K entities. For more information, see the section “Randomizing the Order of Severity Draws across Observations of a Scenario” on page 970.

Illustration of Aggregate Adjusted Loss Simulation Process

This section continues the example in the section “Simulation with Regressors and No External Counts” on page 959 to illustrate the simulation of aggregate adjusted loss.

Recall that the earlier example simulates a scenario that consists of five policyholders. Assume that you want to compute the distribution of the aggregate amount paid to all the policyholders in a year, where the payment for each loss is decided by a deductible and a per-payment limit. To begin with, you must record the deductible and limit information in the input `DATA=` data set. The following table shows the `DATA=` data set from the earlier example, extended to include two variables, `Deductible` and `Limit`:

Obs	age	gender	carType	deductible	limit
1	30	2	1	250	5000
2	25	1	2	500	3000
3	45	2	2	100	2000
4	33	1	1	500	5000
5	50	1	1	200	2000

The variables `Deductible` and `Limit` are referred to as severity adjustment variables, because you need to use them to compute the adjusted severity. Let `AmountPaid` represent the value of adjusted severity that you are interested in. Further, let the following SAS programming statements encode your logic of computing the value of `AmountPaid`:

```
amountPaid = MAX(_sev_ - deductible, 0);
amountPaid = MIN(amountPaid, MAX(limit - _cumadjsevforobs_, 0));
```

PROC HPCDM supplies your program with values of the placeholder symbols `_SEV_` and `_CUMADJSEVFOROBS_`, which represent the value of the current unadjusted severity draw and the sum of adjusted severity values from the previous draws, respectively, for the observation that is being processed. The use of `_CUMADJSEVFOROBS_` helps you ensure that the payment that is made to a given policyholder in a year does not exceed the limit that is recorded in the `Limit` variable.

In order to simulate a sample for the aggregate of `AmountPaid`, you need to submit a PROC HPCDM step whose structure is like the following:

```
proc hpcdm data=<data set name> adjustedseverity=amountPaid
    severityest=<severity parameter estimates data set>
    countstore=<count model store>;
    severitymodel <severity distribution name(s)>;

    amountPaid = MAX(_sev_ - deductible, 0);
    amountPaid = MIN(amountPaid, MAX(limit - _cumadjsevforobs_, 0));
run;
```

The simulation process of one replication that generates one point of the aggregate loss sample and the corresponding point of the aggregate adjusted loss sample is as follows:

1. Use the values `Age=30`, `Gender=2`, and `CarType=1` in the first observation to draw a count from the count distribution. Let that count be 3. Repeat the process for the remaining four observations. Let the counts be as shown in the `Count` column in the following table:

Obs	age	gender	carType	deductible	limit	count
1	30	2	1	250	5000	2
2	25	1	2	500	3000	1
3	45	2	2	100	2000	2
4	33	1	1	500	5000	3
5	50	1	1	200	2000	0

Note that the `Count` column is shown for illustration only; it is not added as a variable to the `DATA=` data set.

2. The simulated counts from all the observations are added to get a value of $N = 8$. This means that for this particular replication, you expect a total of eight loss events in a year from these five policyholders.
3. For the first observation, the scale parameter of the severity distribution is computed by using the values `Age=30`, `Gender=2`, and `CarType=1`. That value of the scale parameter is used together with estimates of the other parameters from the `SEVERITYEST=` data set to make two draws from the severity distribution. The process is repeated for the remaining four policyholders. The fifth policyholder does not generate any loss event for this particular replication, so no severity draws are made by using the fifth observation. Let the severity draws, rounded to integers for convenience, be as shown in the

`_SEV_` column in the following table, where the `_SEV_` column is shown for illustration only; it is not added as a variable to the DATA= data set:

Obs	age	gender	carType	deductible	limit	count	<code>_sev_</code>		
1	30	2	1	250	5000	2	350	2100	
2	25	1	2	500	3000	1	4500		
3	45	2	2	100	2000	2	700	4300	
4	33	1	1	200	5000	3	600	1500	950
5	50	1	1	200	2000	0			

The sample point for the aggregate unadjusted loss is computed by adding the severity values of eight draws, which gives an aggregate loss value of 15,000. The unadjusted aggregate loss is also referred to as the ground-up loss.

For each of the severity draws, your severity adjustment programming statements are executed to compute the adjusted severity, which is the value of `AmountPaid` in this case. For the draws in the preceding table, the values of `AmountPaid` are as follows:

Obs	deductible	limit	<code>_sev_</code>	<code>_cumadjsevforobs_</code>	<code>amountPaid</code>
1	250	5000	350	0	100
1	250	5000	2100	100	1850
2	500	3000	4500	0	3000
3	100	2000	700	0	600
3	100	2000	4300	600	1400
4	200	5000	600	0	400
4	200	5000	1500	400	1300
4	200	5000	950	1700	750

The adjusted severity values are added to compute the cumulative payment value of 9,400, which forms the first sample point for the aggregate adjusted loss.

After recording the aggregate unadjusted and aggregate adjusted loss values in their respective samples, the process returns to step 1 to compute the next sample point unless the specified number of sample points have been simulated.

In this particular example, you can verify that the order in which the 8 loss events are simulated does not affect the aggregate adjusted loss. As a simple example, consider the following order of draws that is different from the consecutive order that was used in the preceding table:

Obs	deductible	limit	<code>_sev_</code>	<code>_cumadjsevforobs_</code>	<code>amountPaid</code>
4	200	5000	600	0	400
3	100	2000	4300	0	2000
1	250	5000	350	0	100
3	100	2000	700	2000	0
4	200	5000	950	400	750
1	250	5000	2100	100	1850
2	500	3000	4500	0	3000
4	200	5000	1500	1150	1300

Although the payments that are made for individual loss events differ, the aggregate adjusted loss is still 9,400.

However, in general, when you use a cumulative severity value such as `_CUMADJSEVFOROBS_` in your program, the order in which the draws are processed affects the final value of aggregate adjusted loss. For more information, see the sections “Randomizing the Order of Severity Draws across Observations of a Scenario” on page 970 and “Illustration of the Need to Randomize the Order of Severity Draws” on page 971.

Randomizing the Order of Severity Draws across Observations of a Scenario

If you specify a scenario that consists of a group of more than one entity, then it is assumed that each entity generates its loss events independently from other entities. In other words, the time at which the loss event of one entity is generated or recorded is independent of the time at which the loss event of another entity is generated or recorded. If entity k generates N_k loss events, where N_k is adjusted to conform to the upper limit of either 1,000 or the value that you specify in the `MAXCOUNTDRAW=` option, then the total number of loss events for a group of K entities is $N = \sum_{k=1}^K N_k$. To simulate the aggregate loss for this group, N severity draws are made and aggregated to compute one point of the compound distribution sample. However, to honor the assumption of independence among entities, the order of those N severity draws must be randomized across K entities such that no entity is preferred over another.

The K entities are represented by K observations of the scenario in the `DATA=` data set. If you specify external counts, the K observations correspond to the observations that have the same replication identifier value. If you do not specify the external counts, then the K observations correspond to all the observations in the `BY` group or in the entire `DATA=` set if you do not specify the `BY` statement.

The randomization process over K observations is implemented as follows. First, one of the K observations is chosen at random and one severity value is drawn from the severity distribution implied by that observation, then another observation is chosen at random and one severity value is drawn from its implied severity distribution, and so on. In each step, the total number of events that are simulated for the selected observation k is incremented by 1. When all N_k events for an observation k are simulated, observation k is retired and the process continues with the remaining observations until a total of N severity draws are made. Let $k(j)$ denote a function that implements this randomization by returning an observation k ($k = 1, \dots, K$) for the j th draw ($j = 1, \dots, N$). The aggregate loss computation can then be formally written as

$$S = \sum_{j=1}^N X_{k(j)}$$

where $X_{k(j)}$ denotes the severity value that is drawn by using observation $k(j)$.

If you do not specify a scale regression model for severity, then all severity values are drawn from the same severity distribution. However, if you specify a scale regression model for severity, then the severity draw is made from the severity distribution that is determined by the values of regressors in observation k . In particular, the scale parameter of the distribution depends on the values of regressors in observation k . If $R(l)$ denotes the scale regression model for observation l and $X_{R(l)}$ denotes the severity value drawn from scale regression model $R(l)$, then the aggregate loss computation can be formally written as

$$S = \sum_{j=1}^N X_{R(k(j))}$$

This randomization process is especially important in the context of simulating an adjusted compound distribution sample when your severity adjustment program uses the aggregate adjusted severity observed so

far to adjust the next severity value. For an illustration of the need to randomize in such cases, see the next section.

Illustration of the Need to Randomize the Order of Severity Draws

This section uses the example of the section “[Illustration of Aggregate Adjusted Loss Simulation Process](#)” on page 967, but with the following PROC HPCDM step:

```
proc hpcdm data=<data set name> adjustedseverity=amountPaid
    severityest=<severity parameter estimates data set>
    countstore=<count model store>;
    severitymodel <severity distribution name(s)>;

    if (_cumadjsev_ > 15000) then
        amountPaid = 0;
    else do;
        penaltyFactor = MIN(3, 15000/(15000 - _cumadjsev_));
        amountPaid = MAX(0, _sev_ - deductible * penaltyFactor);
    end;
run;
```

The severity adjustment statements in the preceding steps compute the value of AmountPaid by using the following provisions in the insurance policy:

- There is a limit of 15,000 on the total amount that can be paid in a year to the group of policyholders that is being simulated. The amount of payment for each loss event depends on the total amount of payments before that loss event.
- The penalty for incurring more losses is imposed in the form of an increased deductible. In particular, the deductible is increased by the ratio of the maximum cumulative payment (15,000) to the amount that remains available to pay for future losses in the year. The factor by which the deductible can be raised has a limit of three.

This example illustrates only step 3 of the simulation process, where randomization is done. It assumes that step 2 of the simulation process is identical to the step 2 in the example in the section “[Illustration of Aggregate Adjusted Loss Simulation Process](#)” on page 967. At the beginning of step 3, let the severity draws from all the observations be as shown in the `_SEV_` column in the following table:

Obs	age	gender	carType	deductible	count	_sev_
1	30	2	1	250	2	350 2100
2	25	1	2	500	1	4500
3	45	2	2	100	2	700 4300
4	33	1	1	200	3	600 1500 950
5	50	1	1	200	0	

If the order of these eight draws is not randomized, then all the severity draws for the first observation are adjusted before all the severity draws of the second observation, and so on. The execution of the severity adjustment program leads to the following sequence of values for AmountPaid:

Obs	deductible	_sev_	_cumadjsev_	penaltyFactor	amountPaid
1	250	350	0	1	100
1	250	2100	100	1.0067	1848.32
2	500	4500	1948.32	1.1493	3925.36

3	100	700	5873.68	1.6436	535.64
3	100	4300	6409.32	1.7461	4125.39
4	200	600	10534.72	3	0
4	200	1500	10534.72	3	900
4	200	950	11434.72	3	350

The preceding sequence of simulating loss events results in a cumulative payment of 11,784.72.

If the sequence of draws is randomized over observations, then the computation of the cumulative payment might proceed as follows for one instance of randomization:

Obs	deductible	_sev_	_cumadjsev_	penaltyFactor	amountPaid
2	500	4500	0	1	4000
1	250	350	4000	1.3636	9.09
3	100	700	4009.09	1.3648	563.52
4	200	950	4572.61	1.4385	662.30
4	200	1500	5234.91	1.5361	1192.78
1	250	2100	6427.69	1.7498	1662.54
4	200	600	8090.24	2.1708	165.83
3	100	4300	8256.07	2.2242	4077.58

In this example, a policyholder is identified by the value in the Obs column. As the table indicates, PROC HPCDM randomizes the order of loss events not only across policyholders but also across the loss events that a given policyholder incurs. The particular sequence of loss events that is shown in the table results in a cumulative payment of 12,333.65. This differs from the cumulative payment that results from the previously considered nonrandomized sequence of loss events, which tends to penalize the fourth policyholder by always processing her payments after all other payments, with a possibility of underestimating the total paid amount. This comparison not only illustrates that the order of randomization affects the aggregate adjusted loss sample but also corroborates the arguments about the importance of order randomization that are made at the beginning of the section “Randomizing the Order of Severity Draws across Observations of a Scenario” on page 970.

Parameter Perturbation Analysis

It is important to realize that most of the parameters of the frequency and severity models are estimated and there is uncertainty associated with the parameter estimates. Any compound distribution estimate that is computed by using these uncertain parameter estimates is inherently uncertain. The aggregate loss sample that is simulated by using the mean estimates of the parameters is just one possible sample from the compound distribution. If information about parameter uncertainty is available, then it is recommended that you conduct parameter perturbation analysis that generates multiple samples of the compound distribution, in which each sample is simulated by using a set of perturbed parameter estimates. You can use the `NPERTURBEDSAMPLES=` option in the PROC HPCDM statement to specify the number of perturbed samples to be generated. The set of perturbed parameter estimates is created by making a random draw of the parameter values from their joint probability distribution. If you specify `NPERTURBEDSAMPLES=P`, then PROC HPCDM creates P sets of perturbed parameters and each set is used to simulate a full aggregate sample. The summary analysis of P such aggregate loss samples results in a set of P estimates for each summary statistic and percentile of the compound distribution. The mean and standard deviation of this set of P estimates quantify the uncertainty that is associated with the compound distribution.

The parameter uncertainty information is available in the form of either the variance-covariance matrix of the parameter estimates or standard errors of the parameters estimates. If the variance-covariance matrix

is available and is positive definite, then PROC HPCDM assumes that the joint probability distribution of the parameter estimates is a multivariate normal distribution, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where the mean vector $\boldsymbol{\mu}$ is the set of point parameter estimates and $\boldsymbol{\Sigma}$ is the variance-covariance matrix. If the variance-covariance matrix is not available or is not positive definite, then PROC HPCDM assumes that each parameter has a univariate normal distribution, $\mathcal{N}(\mu, \sigma^2)$, where μ is the point estimate of the parameter and σ is the standard error of the parameter estimate.

To make the random draws from the multivariate normal distribution of all parameters or the univariate distributions of individual parameters, PROC HPCDM uses a pseudorandom number generator (PRNG) that is controlled by the `PERTURBMETHOD=` option as follows:

- `PERTURBMETHOD=ASYNC` is the legacy method that releases prior to SAS/ETS 15.1 used and is the default for the current release. This method allows each thread on each grid node to use a different PRNG for perturbation, which in fact is the same PRNG that the thread uses for making random draws from the severity or frequency distributions. Using different PRNGs and interleaving perturbation-related random draws with severity and count random draws causes each thread to use a different set of perturbed parameters while generating a subset of the same perturbed sample; in turn, this leads to a perturbed sample that is a heterogeneous collection of smaller perturbed samples, each of which is generated from a different compound distribution model.
- `PERTURBMETHOD=SYNC` method is the recommended method because it uses a single PRNG to perturb the parameters and synchronizes the set of perturbed parameters across all threads on all grid nodes. This makes each perturbed sample a homogeneous sample that corresponds to a single compound distribution model.

If you specify the severity models by using the `SEVERITYEST=` data set, then the point parameter estimates are expected to be available in the `SEVERITYEST=` data set in observations for which `_TYPE_='EST'`, the standard errors are expected to be available in the `SEVERITYEST=` data set in observations for which `_TYPE_='STDERR'`, and the variance-covariance matrix is expected to be available in the `SEVERITYEST=` data set in observations for which `_TYPE_='COV'`. If you use the `SEVERITY` procedure to create the `SEVERITYEST=` data set, then you need to specify the `COVOUT` option in the `PROC SEVERITY` statement to make the variance-covariance estimates available in the `SEVERITYEST=` data set.

If you specify the severity models by using the `SEVERITYSTORE=` item store, then you need to specify the `OUTSTORE=` option in the `PROC SEVERITY` statement to create that item store, which includes the point parameter estimates and standard errors by default. In addition, you need to specify the `COVOUT` option in the `PROC SEVERITY` statement to make the variance-covariance estimates available in the `SEVERITYSTORE=` item store.

For the frequency model, you must use the `COUNTREG` procedure to create the `COUNTSTORE=` item store, which always contains the point estimates, standard errors, and variance-covariance matrix of the parameters.

If you specify the `ADJUSTEDSEVERITY=` option in the `PROC HPCDM` statement, then a separate perturbation analysis is conducted for the distribution of the aggregate adjusted loss.

Descriptive Statistics

This section provides computational details for the descriptive statistics that are computed for each aggregate loss sample. You can also save these statistics in an `OUTSUM=` data set by specifying appropriate keywords in the `OUTSUM` statement.

This section gives specific details about the moment statistics. For more information about the methods of computing percentile statistics, see the description of the PCTLDEF= option in the UNIVARIATE procedure in the *Base SAS Procedures Guide: Statistical Procedures*.

Standard algorithms (Fisher 1973) are used to compute the moment statistics. The computational methods that the HPCDM procedure uses are consistent with those that other SAS procedures use for calculating descriptive statistics.

Mean

The sample mean is calculated as

$$\bar{y} = \frac{\sum_{i=1}^n y_i}{n}$$

where n is the size of the generated aggregate loss sample and y_i is the i th value of the aggregate loss.

Standard Deviation

The standard deviation is calculated as

$$s = \sqrt{\frac{1}{d} \sum_{i=1}^n (y_i - \bar{y})^2}$$

where n is the size of the generated aggregate loss sample, y_i is the i th value of the aggregate loss, \bar{y} is the sample mean, and d is the divisor controlled by the VARDEF= option in the PROC HPCDM statement:

$$d = \begin{cases} n - 1 & \text{if VARDEF=DF (default)} \\ n & \text{if VARDEF=N} \end{cases}$$

Skewness

The sample skewness, which measures the tendency of the deviations to be larger in one direction than in the other, is calculated as

$$\frac{1}{d_s} \sum_{i=1}^n \left(\frac{y_i - \bar{y}}{s} \right)^3$$

where n is the size of the generated aggregate loss sample, y_i is the i th value of the aggregate loss, \bar{y} is the sample mean, s is the sample standard deviation, and d_s is the divisor controlled by the VARDEF= option in the PROC HPCDM statement:

$$d_s = \begin{cases} \frac{(n-1)(n-2)}{n} & \text{if VARDEF=DF (default)} \\ n & \text{if VARDEF=N} \end{cases}$$

If VARDEF=DF, then n must be greater than 2.

The sample skewness can be positive or negative; it measures the asymmetry of the data distribution and estimates the theoretical skewness $\sqrt{\beta_1} = \mu_3 \mu_2^{-\frac{3}{2}}$, where μ_2 and μ_3 are the second and third central moments. Observations that are normally distributed should have a skewness near zero.

Kurtosis

The sample kurtosis, which measures the heaviness of tails, is calculated as in [Table 16.2](#) depending on the value that you specify in the **VARDEF=** option.

Table 16.2 Formulas for Kurtosis

VARDEF= Value	Formula
DF (default)	$\frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum_{i=1}^n \left(\frac{y_i - \bar{y}}{s} \right)^4 - \frac{3(n-1)^2}{(n-2)(n-3)}$
N	$\frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - \bar{y}}{s} \right)^4 - 3$

In these formulas, n is the size of the generated aggregate loss sample, y_i is the i th value of the aggregate loss, \bar{y} is the sample mean, and s is the sample standard deviation. If **VARDEF=DF**, then n must be greater than 3.

The sample kurtosis measures the heaviness of the tails of the data distribution. It estimates the adjusted theoretical kurtosis denoted as $\beta_2 - 3$, where $\beta_2 = \frac{\mu_4}{\mu_2^2}$ and μ_4 is the fourth central moment. Observations that are normally distributed should have a kurtosis near zero.

Input Specification

PROC HPCDM accepts the **DATA=** and **SEVERITYEST=** data sets and the **COUNTSTORE=** and **SEVERITYSTORE=** item stores as input. This section details the information that they are expected to contain.

DATA= Data Set

If you specify the **BY** statement, then the **DATA=** data set must contain all the **BY** variables that you specify in the **BY** statement and the data set must be sorted by the **BY** variables unless the **BY** statement includes the **NOTSORTED** option.

If the severity models in the **SEVERITYEST=** data set or the **SEVERITYSTORE=** item store contain any scale regressors, then all those regressors must be present in the **DATA=** data set.

If you specify the programming statements to compute an aggregate adjusted loss, and if your specified **ADJUSTEDSEVERITY=** symbol depends on severity adjustment variables, then the **DATA=** data set must contain all such variables.

The rest of the contents of the **DATA=** data set depends on whether you specify the **EXTERNALCOUNTS** statement. If you specify the **EXTERNALCOUNTS** statement, then the **DATA=** data set is expected to contain the **COUNT=** and **ID=** variables that you specify in the **EXTERNALCOUNTS** statement. If you do not specify the **EXTERNALCOUNTS** statement, then the **DATA=** data set must contain all the regressors, including zero model regressors, that are present in the count model that the **COUNTSTORE=** item store contains.

You do not need to specify the DATA= data set if *all* the following conditions are true:

- You do not specify the BY statement.
- You specify the severity models such that none of them are scale regression models.
- You do not specify the EXTERNALCOUNTS statement.
- You specify a COUNTSTORE= item store such that the count model contains no count regressors.
- Your severity adjustment programming statements, if you specify any, do not use any external input.

If you specify the BY statement, then PROC HPCDM analyzes only the BY groups that are present in the input source of the severity and count models. If neither the severity models nor the count models contain regression effects, then the DATA= data set must contain BY variables and one row for each BY group that you want PROC HPCDM to analyze.

SEVERITYEST= Data Set

The SEVERITYEST= data set is expected to contain the parameter estimates of the severity models. This is a required data set; you must specify it whenever you use PROC HPCDM.

The SEVERITYEST= data set must have the same format as the OUTEST= data set that is created by the SEVERITY procedure. For more information, see the description of the OUTEST= data set in the SEVERITY procedure in the *SAS/ETS User's Guide*.

If you specify the BY statement, then the SEVERITYEST= data set must contain all the BY variables that you specify in the BY statement. If you do not specify the NOTSORTED option in the BY statement, then the SEVERITYEST= data set must be sorted by the BY variables.

SEVERITYSTORE= Item Store

The SEVERITYSTORE= item store is expected to be created by using the OUTSTORE= option in a PROC SEVERITY statement. For more information, see the description of the OUTSTORE= option in the SEVERITY procedure in the *SAS/ETS User's Guide*.

You must specify this item store when you do not specify the SEVERITYEST= data set. Also, if your severity model is a scale regression model that contains classification or interaction effects, then you cannot use the SEVERITYEST= data set. You must specify such severity models by specifying the SEVERITYSTORE= item store.

If you specify the BY statement, then the SEVERITYSTORE= item store must have been created by using a PROC SEVERITY step that uses an identical BY statement.

COUNTSTORE= Item Store

The COUNTSTORE= item store is expected to be created by using the STORE statement in the COUNTREG procedure. You must specify the COUNTSTORE= item store when you do not specify the EXTERNALCOUNTS statement. For more information, see the description of the STORE statement in the COUNTREG procedure in the *SAS/ETS User's Guide*.

If you specify the BY statement, then the COUNTSTORE= item store must have been created by using a PROC COUNTREG step that uses an identical BY statement.

Output Data Sets

PROC HPCDM writes the output data sets that you specify in the OUT= option of the **OUTPUT** and **OUTSUM** statements. The contents of these output data sets are described in the sections “**OUTSAMPLE= Data Set**” on page 977 and “**OUTSUM= Data Set**” on page 977, respectively.

OUTSAMPLE= Data Set

The OUTSAMPLE= data set records the full sample of the aggregate loss and aggregate adjusted loss.

If you specify the BY statement, then the data are organized in BY groups and the data set contains variables that you specify in the BY statement. In addition, the OUTSAMPLE= data set contains the following variables:

SEVERITYMODEL

indicates the name of the severity distribution model.

COUNTMODEL

indicates the name of the count model. If you specify the EXTERNALCOUNTS statement, then the value of this variable is “_EXTERNAL_”. If you specify the COUNTSTORE= option, then the value of this variable is “_COUNTSTORE_”.

<unadjusted sample variable>

indicates the value of the unadjusted aggregate loss. The name of this variable is the value of the **SAMPLEVAR=** option in the OUTPUT statement. If you do not specify the SAMPLEVAR= option, then the variable is named **_AGGSEV_**.

<adjusted sample variable>

indicates the value of the adjusted aggregate loss. This variable is created only when you specify the programming statements and the **ADJUSTEDSEVERITY=** option in the PROC HPCDM statement. The name of this variable is the value of the **ADJSAMPLEVAR=** option in the OUTPUT statement. If you do not specify the ADJSAMPLEVAR= option, then the variable is named **_AGGADJSEV_**.

DRAWID

indicates the identifier for the perturbed sample. This variable is created only when you specify the **NPERTURBEDSAMPLES=** option in the PROC HPCDM statement. The value of this variable identifies the perturbed sample. A value of 0 for the **_DRAWID_** variable indicates an unperturbed sample.

OUTSUM= Data Set

The OUTSUM= data set records the summary statistics and percentiles of the compound distributions of aggregate loss and aggregate adjusted loss. Only the estimates that you request in the OUTSUM statement are written to the OUTSUM= data set. For more information about the method of naming the variables that correspond to the summary statistics or percentiles, see the description of the **OUTSUM** statement.

If you specify the BY statement, then the data are organized in BY groups and the data set contains variables that you specify in the BY statement. In addition, the OUTSUM= data set contains the following variables:

<code>_SEVERITYMODEL_</code>	indicates the name of the severity distribution model.
<code>_COUNTMODEL_</code>	indicates the name of the count model. If you specify the <code>EXTERNALCOUNTS</code> statement, then the value of this variable is “ <code>_EXTERNAL_</code> ”. If you specify the <code>COUNTSTORE=</code> option, then the value of this variable is “ <code>_COUNTSTORE_</code> ”.
<code>_SAMPLEVAR_</code>	indicates the name of the aggregate loss sample. For an unadjusted sample, the value of the variable is the value of the <code>SAMPLEVAR=</code> option that you specify in the <code>OUTPUT</code> statement or the default value of <code>_AGGSEV_</code> . For an adjusted sample, the value of the variable is the value of the <code>ADJSAMPLEVAR=</code> option that you specify in the <code>OUTPUT</code> statement or the default value of <code>_AGGADJSEV_</code> .
<code>_DRAWID_</code>	indicates the identifier for the perturbed sample. This variable is created only when you specify the <code>NPerturbedSamples=</code> option in the <code>PROC HPCDM</code> statement. The value of this variable identifies the perturbed sample. A value of 0 for <code>_DRAWID_</code> indicates an unperturbed sample.

Displayed Output

The HPCDM procedure optionally produces displayed output by using the Output Delivery System (ODS). All output is controlled by the `PRINT=` option in the `PROC HPCDM` statement. Table 16.3 relates the `PRINT=` options to ODS tables.

Table 16.3 ODS Tables Produced in PROC HPCDM

ODS Table Name	Description	Option
CompoundInfo	Compound distribution information	Default
DataSummary	Input data summary	Default
Percentiles	Percentiles of the aggregate loss sample	<code>PRINT=PERCENTILES</code>
PerformanceInfo	Execution environment information that pertains to the computational performance	Default
PerturbedPctlSummary	Perturbation analysis of percentiles	<code>PRINT=PERTURBSUMMARY</code> and <code>NPerturbedSamples > 0</code>
PerturbedSummary	Perturbation analysis of summary statistics	<code>PRINT=PERTURBSUMMARY</code> and <code>NPerturbedSamples > 0</code>
SummaryStatistics	Summary statistics of the aggregate loss sample	<code>PRINT=SUMMARYSTATISTICS</code>
Timing	Timing information for various computational stages of the procedure	<code>DETAILS (PERFORMANCE statement)</code>

PRINT= Option

This section provides detailed descriptions of the tables that are displayed by using different PRINT= options.

- If you do not specify the PRINT= option and if you do not specify the NOPRINT or PRINT=NONE options, then by default PROC HPCDM produces the CompoundInfo, DataSummary, and SummaryStatistics ODS tables.

The “Compound Distribution Information” table (ODS name: CompoundInfo) displays the information about the severity and count models.

The “Input Data Summary” table (ODS name: DataSummary) is displayed when you specify the DATA= data set. The table displays the total number of observations and the valid number of observations in the data set. If you specify the EXTERNALCOUNTS statement, then the table also displays the number of replications and total number of loss events across all replications.

- If you specify PRINT=PERCENTILES, the “Percentiles” table (ODS name: Percentiles) is displayed for the distribution of the aggregate loss. The table contains estimates of all the predefined percentiles in addition to the percentiles that you request in the OUTSUM statement.

If you specify the programming statements and the ADJUSTEDSEVERITY= symbol, then an additional table is displayed for the distribution of the aggregate adjusted loss. This table also contains estimates of all the predefined percentiles in addition to the percentiles that you request in the OUTSUM statement.

- If you specify PRINT=PERTURBSUMMARY, two tables are displayed for the distribution of the aggregate loss. The “Perturbed Summary Statistics” table (ODS name: PerturbedSummary) displays the summary of the effect of perturbing model parameters on the following five summary statistics of the distribution: mean, standard deviation, variance, skewness, and kurtosis. The “Perturbed Percentiles” table (ODS name: PerturbedPctlSummary) displays the perturbation summary for all the predefined percentiles in addition to the percentiles that you request in the OUTSUM statement.

The tables are displayed only if you specify a value greater than 0 for the NPERTURBEDSAMPLES= option.

If you specify a value of P for the NPERTURBEDSAMPLES= option, then for each summary statistic and percentile, an average and standard error of the set of P values of that summary statistic or percentile are displayed in the respective perturbation summary tables.

If you specify the programming statements and the ADJUSTEDSEVERITY= symbol, then additional perturbation summary tables are displayed for the distribution of the aggregate adjusted loss.

- If you specify PRINT=SUMMARYSTATISTICS, the “Summary Statistics” table (ODS name: SummaryStatistics) is displayed for the distribution of the aggregate loss. The table contains estimates of the following summary statistics: the number of observations in the sample, maximum value in the sample, minimum value in the sample, mean, median, standard deviation, interquartile range, variance, skewness, and kurtosis.

If you specify the programming statements and the ADJUSTEDSEVERITY= symbol, then an additional table of summary statistics is displayed for the distribution of the aggregate adjusted loss.

Performance Information

The “Performance Information” table (ODS name: PerformanceInfo) is produced by default. It displays information about the execution mode. For single-machine mode, the table displays the number of threads that are used. For distributed mode, the table displays the grid mode (symmetric or asymmetric), the number of compute nodes, and the number of threads per node.

If you specify the DETAILS option in the PERFORMANCE statement, PROC HPCDM also produces a “Timing” table (ODS name: Timing) that displays elapsed times (absolute and relative) for the main tasks of the procedure.

ODS Graphics

Statistical procedures use ODS Graphics to create graphs as part of their output. ODS Graphics is described in detail in Chapter 21, “Statistical Graphics Using ODS” (*SAS/STAT User’s Guide*).

Before you create graphs, ODS Graphics must be enabled (for example, with the ODS GRAPHICS ON statement). For more information about enabling and disabling ODS Graphics, see the section “Enabling and Disabling ODS Graphics” in that chapter.

The overall appearance of graphs is controlled by ODS styles. Styles and other aspects of using ODS Graphics are discussed in the section “A Primer on ODS Statistical Graphics” in that chapter.

This section describes the use of ODS for creating graphics with the HPCDM procedure.

NOTE: If you request simulation of an aggregate loss sample of large size, either by specifying a large value for the NREPLICATES= option or by including a large number of replicates in the DATA= data set that you specify in conjunction with the EXTERNALCOUNTS statement, then it is recommended that you not request any plots, because creating plots that have large numbers of points can require a very large amount of hardware resources and can take a very long time. You can disable the generation of plots either by submitting the ODS GRAPHICS OFF statement before submitting the PROC HPCDM step or by specifying the PLOTS=NONE option in the PROC HPCDM statement. It is recommended that you request plots only when the sample size is less than 100,000.

ODS Graph Names

PROC HPCDM assigns a name to each graph that it creates by using ODS. You can use these names to selectively refer to the graphs. The names are listed in Table 16.4.

Table 16.4 ODS Graphics Produced by PROC HPCDM

ODS Graph Name	Plot Description	PLOTS= Option
ConditionalDensityPlot	Conditional density plot	CONDITIONALDENSITY
DensityPlot	Probability density function plot	DENSITY
EDFPlot	Empirical distribution function plot	EDF

Conditional Density Plot

The conditional density plot helps you visually analyze two or three regions of the compound distribution by displaying a density function estimate that is conditional on the values of the aggregate loss that fall in those regions. You can specify the region boundaries in terms of quantiles by using the **LEFTQ=** and **RIGHTQ=** suboptions of the **PLOTS=CONDITIONALDENSITY** option. This is especially useful if you want to see the distribution of aggregate loss values in the right- and left-tail regions.

If you specify the programming statements and the **ADJUSTEDSEVERITY=** symbol, then a separate set of conditional density plots are displayed for the aggregate adjusted loss.

Probability Density Function Plot

The probability density function (PDF) plot shows the nonparametric estimates of the PDF of the aggregate loss distribution. This plot includes histogram and kernel density estimates.

If you specify the programming statements and the **ADJUSTEDSEVERITY=** symbol, then a separate density plot is displayed for the aggregate adjusted loss.

Empirical Distribution Function Plot

The empirical density function (EDF) plot shows the nonparametric estimate of the cumulative distribution function of the aggregate loss distribution. You can specify the **ALPHA=** suboption of the **PLOTS=EDF** option to request that the upper and lower confidence limits be plotted for each EDF estimate. By default, the confidence interval is not plotted.

If you specify the programming statements and the **ADJUSTEDSEVERITY=** symbol, then a separate EDF plot is displayed for the aggregate adjusted loss.

Examples: HPCDM Procedure

Example 16.1: Estimating the Probability Distribution of Insurance Payments

The primary outcome of running PROC HPCDM is the estimate of the compound distribution of aggregate loss, given the distributions of frequency and severity of the individual losses. This aggregate loss is often referred to as the ground-up loss. If you are an insurance company or a bank, you are also interested in acting on the ground-up loss by computing an entity that is derived from the ground-up loss. For example, you might want to estimate the distribution of the amount that you are expected to pay for the losses or the distribution of the amount that you can offload onto another organization, such as a reinsurance company. PROC HPCDM enables you to specify a severity adjustment program, which is a sequence of SAS programming statements that adjust the severity of the individual loss event to compute the entity of interest. Your severity adjustment program can use external information that is recorded as variables in the observations of the **DATA=** data set in addition to placeholder symbols for information that PROC HPCDM generates internally, such as the severity of the current loss event (**_SEV_**) and the sum of the adjusted severity values of the events that have been simulated thus far for the current sample point (**_CUMADJSEV_**). If you are doing a scenario analysis such that a scenario contains more than one observation, then you can also access the cumulative

severity and cumulative adjusted severity for the current observation by using the `_CUMSEVFOROBS_` and `_CUMADJSEVFOROBS_` symbols.

This example continues the example of the section “[Scenario Analysis](#)” on page 934 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, and u denotes the limit on the total payments that are made to a policyholder in a year, then Y , the amount that is paid to the policyholder for each loss event, is defined as follows:

$$Y = \begin{cases} 0 & X \leq d \\ d' \frac{X-d}{d'-d} & d < X \leq d' \\ X & d' < X \leq u \\ u & X > u \end{cases}$$

You can encode this logic by using a set of SAS programming statements.

Extend the `Work.GroupOfPolicies` data set in the example in the section “[Scenario Analysis](#)” on page 934 to include the following three additional variables for each policyholder: `LowDeductible` to record d , `HighDeductible` to record d' , and `Limit` to record u . The data set contains the observations as shown in [Output 16.1.1](#).

Output 16.1.1 Scenario Analysis Data for Multiple Policyholders with Policy Provisions

policyholderid	age	gender	carType	annualMiles	education	carSafety	income
1	1.18	2	1	2.2948	3	0.99532	1.59870
2	0.66	2	2	2.8148	1	0.05625	0.67539
3	0.82	1	2	1.6130	2	0.84146	1.05940
4	0.44	1	1	1.2280	3	0.14324	0.24110
5	0.44	1	1	0.9670	2	0.08656	0.65979

lowDeductible	highDeductible	limit	annualLimit
400	1400	7500	10000
300	1300	2500	20000
100	1100	5000	10000
300	800	5000	20000
100	1100	5000	20000

The following PROC HPCDM step estimates the compound distributions of the aggregate loss and the aggregate amount that is paid to the group of policyholders in the `Work.GroupOfPolicies` data set by using the count model that is stored in the `Work.CountregModel` item store and the lognormal severity model that is stored in the `Work.SevRegEst` data set:

```
/* Simulate the aggregate loss distribution and aggregate adjusted
   loss distribution for the scenario with multiple policyholders */
proc hpcdm data=groupOfPolicies nreplicates=10000 seed=13579 print=all
    countstore=work.countregmodel severityest=work.sevregest
    plots=(edf pdf) 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 */
end;
run;

```

The preceding step uses a severity adjustment program to compute the value of the symbol AmountPaid and specifies that symbol in the ADJUSTEDSEVERITY= option in the PROC HPCDM step. The program is executed for each simulated loss event. The PROC HPCDM supplies your program with the value of the severity in the _SEV_ placeholder symbol.

The “Sample Summary Statistics” table in [Output 16.1.2](#) shows the summary statistics of the compound distribution of the aggregate ground-up loss. The “Adjusted Sample Summary Statistics” table shows the summary statistics of the compound distribution of the aggregate AmountPaid. The average aggregate payment is about 4,361, as compared to the average aggregate ground-up loss of 5,906.

Output 16.1.2 Summary Statistics of Compound Distributions of the Total Loss and Total Amount Paid

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

Compound Distribution Information			
Severity Model	Lognormal Distribution		
Scale Model Regressors	carType carSafety income		
Count Model	NegBin(p=2) Model in Item Store WORK.COUNTREGMODEL		

Sample Summary Statistics			
Mean	5906.2	Median	4727.7
Standard Deviation	4801.7	Interquartile Range	5227.0
Variance	23056465.3	Minimum	0
Skewness	2.25016	Maximum	64811.8
Kurtosis	10.01578	Sample Size	10000

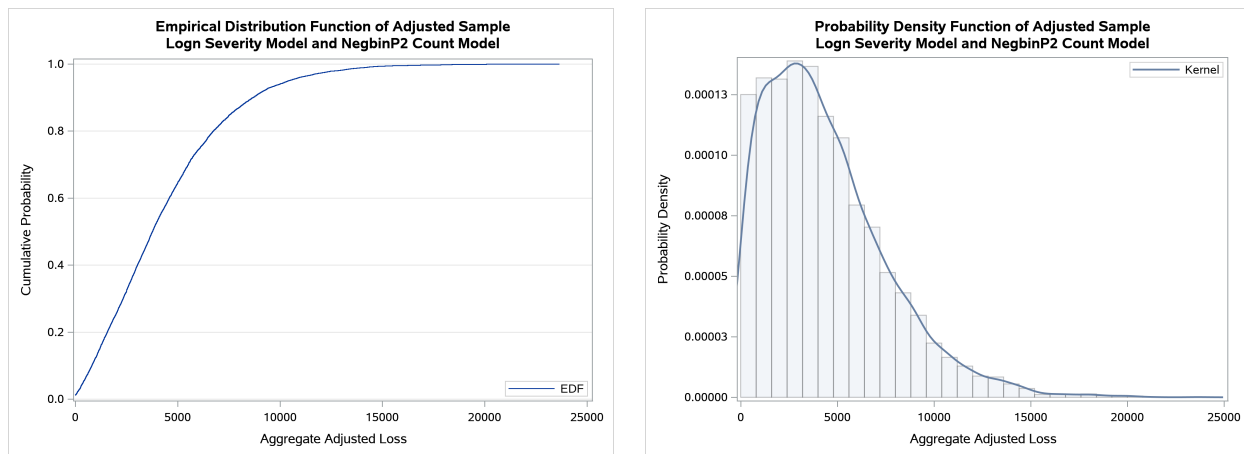
Adjusted Sample Summary Statistics			
Mean	4361.0	Median	3762.9
Standard Deviation	3181.7	Interquartile Range	4133.6
Variance	10123000.5	Minimum	0
Skewness	1.11692	Maximum	23657.4
Kurtosis	1.64518	Sample Size	10000

The perturbation summary of the distribution of AmountPaid is shown in [Output 16.1.3](#). It shows that you can expect to pay a median of $3,796 \pm 271$ to this group of five policyholders in a year. Also, if the 99.5th percentile defines the worst case, then you can expect to pay $15,573 \pm 859$ in the worst-case.

Output 16.1.3 Perturbation Summary of the Total Amount Paid

Adjusted Sample Percentile Perturbation Analysis		
Percentile	Estimate	Standard Error
1	0.94036	6.15322
5	386.17494	65.79399
25	1988.9	188.59978
50	3796.0	271.27093
75	6153.8	393.13966
95	10435.2	621.20756
99	14108.5	827.74239
99.5	15573.1	858.66726
Number of Perturbed Samples = 50		
Size of Each Sample = 10000		

The empirical distribution function (EDF) and probability density function plots of the aggregate adjusted loss are shown in [Output 16.1.4](#). Both plots indicate a heavy-tailed distribution of the total amount paid.

Output 16.1.4 PDF and EDF Plots of the Compound Distribution of the Total Amount Paid

Now consider that, in the future, you want to modify the policy provisions to add a limit on the total amount of payment that is made to an individual policyholder in one year and to impose a group limit of 15,000 on the total amount of payments that are made to the group as a whole in one year. You can analyze the effects of these modified policy provisions on the distribution of the aggregate paid amount by recording the individual policyholder's annual limit in the AnnualLimit variable of the input data set and then modifying your severity adjustment program by using the placeholder symbols `_CUMADJSEVFOROBS_` and `_CUMADJSEV_` as shown in the following PROC HPCDM step:

```
/* 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.countregmodel severityest=work.sevregest
    plots=none 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_));

  /* impose group's annual limit */
  amountPaid = MIN(amountPaid, MAX(0,15000 - _cumadjsev_));
end;
run;

```

The results of the perturbation analysis for these modified policy provisions are shown in [Output 16.1.5](#). When compared to the results of [Output 16.1.3](#), the additional policy provisions of restricting the total payment to the policyholder and the group have reduced the median payment slightly, but the provisions have reduced the worst-case payment (99.5th percentile) to $14,755 \pm 392$ from $15,573 \pm 859$.

Output 16.1.5 Perturbation Summary of the Total Amount Paid for Modified Policy Provisions

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

Adjusted Sample Percentile Perturbation Analysis		
Percentile	Estimate	Standard Error
1	0.46949	2.23897
5	382.04931	56.96535
25	1953.8	167.85926
50	3733.5	250.62840
75	6054.8	348.38707
95	10272.4	520.34620
99	13728.9	631.17302
99.5	14754.8	392.25491
Number of Perturbed Samples = 50		
Size of Each Sample = 10000		

Example 16.2: Using Externally Simulated Count Data

The COUNTREG procedure enables you to estimate count regression models that are based on the most commonly used discrete distributions, such as the Poisson, negative binomial (both $p = 1$ and $p = 2$), and Conway-Maxwell-Poisson distributions. PROC COUNTREG also enables you to fit zero-inflated models that are based on Poisson, negative binomial ($p = 2$), and Conway-Maxwell-Poisson distributions. However,

there might be situations in which you want to use some other method of fitting count regression models. For example, if you are modeling the number of loss events that are incurred by two financial instruments such that there is some dependency between the two, then you might use some multivariate frequency modeling methods and simulate the counts for each instrument by using the dependency structure between the count model parameters of the two instruments. As another example, you might want to use different types of count models for different BY groups in your data; this is not possible in PROC COUNTREG. So you need to simulate the counts for such BY groups externally. PROC HPCDM enables you to supply externally simulated counts by using the EXTERNALCOUNTS statement. PROC HPCDM then does not need to simulate the counts internally; it simulates only the severity of each loss event by using the severity model estimates that you specify in the SEVERITYEST= data set or the SEVERITYSTORE= item store. The simulation process is described and illustrated in the section “[Simulation with External Counts](#)” on page 961.

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, retail banking and commercial banking, by using some key risk indicators (KRIs). Assume that your model fitting and model selection process has determined that the Poisson regression model and negative binomial regression model are the best-fitting count models for number of loss events that are incurred in the retail banking and commercial banking businesses, respectively. 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 there is some overlap between the two sets. 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 parameter estimates of the negative binomial and Poisson regression models, as determined by PROC COUNTREG, be available in the Work.CountEstEx2NB2 and Work.CountEstEx2Poisson data sets, respectively. These data sets are produced by using the OUTEST= option in the respective PROC COUNTREG statements. Let the parameter estimates of the best-fitting severity models, as determined by PROC SEVERITY, be available in the Work.SevEstEx2Best data set. You can find the code to prepare these data sets in the PROC HPCDM sample program *hcdmex02.sas*.

Now, consider that you want to estimate the distribution of the aggregate loss for a scenario, which is represented by a specific set of KRI values. The following DATA step illustrates one such scenario:

```
/* Generate a scenario data set for a single operating condition */
data singleScenario (keep=corpKRI1 corpKRI2 cbKRI1 cbKRI2 cbKRI3
                    rbKRI1 rbKRI2 rbKRI3);
    array x{8} corpKRI1 corpKRI2 cbKRI1 cbKRI2 cbKRI3 rbKRI1 rbKRI2 rbKRI3;
    call streaminit(5151);
```



```

do i=1 to dim(x);
    x(i) = rand('NORMAL');
end;
output;
run;

```

The Work.SingleScenario data set contains all the KRIs that are included in the count and severity models of both business lines. Note that if you standardize or scale the KRIs while fitting the count and severity models, then you must apply the same standardization or scaling method to the values of the KRIs that you specify in the scenario. In this particular example, all KRIs are assumed to be standardized.

The following DATA step uses the scenario in the Work.SingleScenario data set to simulate 10,000 replications of the number of loss events that you might observe for each business line and writes the simulated counts to the NumLoss variable of the Work.LossCounts1 data set:

```

/* Simulate multiple replications of the number of loss events that
   you can expect in the scenario being analyzed */
data lossCounts1 (keep=line corpKRI1 corpKRI2 cbKRI2 rbKRI1 rbKRI3 numloss);
    array cxR{3} corpKRI1 rbKRI1 rbKRI2;
    array cbetaR{4} _TEMPORARY_;
    array cxC{5} corpKRI1 corpKRI2 cbKRI1 cbKRI2 cbKRI3;
    array cbetaC{6} _TEMPORARY_;

    retain theta;
    if _n_ = 1 then do;
        call streaminit(5151);
        * read count model estimates *;
        set countEstEx2NB2(where=(line='CommercialBanking' and _type_='PARM'));
        cbetaC(1) = Intercept;
        do i=1 to dim(cxC);
            cbetaC(i+1) = cxC(i);
        end;
        alpha = _Alpha;
        theta = 1/alpha;

        set countEstEx2Poisson(where=(line='RetailBanking' and _type_='PARM'));
        cbetaR(1) = Intercept;
        do i=1 to dim(cxR);
            cbetaR(i+1) = cxR(i);
        end;
    end;

    set singleScenario;
    do iline=1 to 2;
        if (iline=1) then line = 'CommercialBanking';
        else line = 'RetailBanking';
        do repid=1 to 10000;
            * draw from count distribution *;
            if (iline=1) then do;
                xbeta = cbetaC(1);
                do i=1 to dim(cxC);
                    xbeta = xbeta + cxC(i) * cbetaC(i+1);
                end;
                Mu = exp(xbeta);
            end;

```

```

        p = theta/(Mu+theta);
        numloss = rand('NEGB',p,theta);
    end;
    else do;
        xbeta = cbetaR(1);
        do i=1 to dim(cxR);
            xbeta = xbeta + cxR(i) * cbetaR(i+1);
        end;
        numloss = rand('POISSON', exp(xbeta));
    end;
    output;
end;
end;
run;

```

The Work.LossCounts1 data set contains the NumLoss variable in addition to the KRIs that are used by the severity regression model, which are needed by PROC HPCDM to simulate the aggregate loss.

By default, PROC HPCDM computes an aggregate loss distribution by using each of the severity models that you specify in the SEVERITYMODEL statement. However, you can restrict PROC HPCDM to use only a subset of the severity models for a given BY group by modifying the SEVERITYEST= data set to include only the estimates of the desired severity models in each BY group, as illustrated in the following DATA step:

```

/* Keep only the best severity model for each business line
   and set coefficients of unused regressors in each model to 0 */
data sevestEx2Best;
    set sevestEx2;
    if ((line = 'CommercialBanking' and _model_ = 'Logn')) then do;
        corpKRI2 = 0; rbKRI1 = 0; rbKRI3 = 0;
        output;
    end;
    else if ((line = 'RetailBanking' and _model_ = 'Gamma')) then do;
        corpKRI1 = 0; cbKRI2 = 0;
        output;
    end;
run;

```

Note that the preceding DATA step also sets the coefficients of the unused regressors in each model to 0. This is important because PROC HPCDM uses all the regressors that it detects from the SEVERITYEST= data set for each severity model.

Now, you are ready to estimate the aggregate loss distribution for each line of business by submitting the following PROC HPCDM step, in which you specify the EXTERNALCOUNTS statement to request that external counts in the NumLoss variable of the DATA= data set be used for simulation of the aggregate loss:

```

/* Estimate the distribution of the aggregate loss for both
   lines of business by using the externally simulated counts */
proc hpcdm data=lossCounts1 seed=13579 print=all
    severityest=sevestEx2Best;
    by line;
    externalcounts count=numloss;
    severitymodel logn gamma;
run;

```

Each observation in the `Work.LossCounts1` data set represents one replication of the external counts simulation process. For each such replication, the preceding PROC HPCDM step makes as many severity draws from the severity distribution as the value of the `NumLoss` variable and adds the severity values from those draws to compute one sample point of the aggregate loss. The severity distribution that is used for making the severity draws has a scale parameter value that is decided by the KRI values in the given observation and the regression parameter values that are read from the `Work.SevEstEx2Best` data set.

The summary statistics and percentiles of the aggregate loss distribution for the commercial banking business, which uses the lognormal severity model, are shown in [Output 16.2.1](#). The “Input Data Summary” table indicates that each of the 10,000 observations in the `BY` group is treated as one replication and that there are a total of 19,028 loss events produced by all the replications together. For the scenario in the `Work.SingleScenario` data set, you can expect the commercial banking business to incur an average aggregate loss of 643 units, as shown in the “Sample Summary Statistics” table, and the chance that the loss will exceed 4,762 units is 0.5%, as shown in the “Sample Percentiles” table.

Output 16.2.1 Aggregate Loss Summary for Commercial Banking Business

The HPCDM Procedure

`line=CommercialBanking`

Input Data Summary

Name	WORK.LOSSCOUNTS1
Observations	10000
Valid Observations	10000
Replications	10000
Total Count	19028

`line=CommercialBanking`

Sample Summary Statistics

Mean	643.24599	Median	363.33564
Standard Deviation	843.56959	Interquartile Range	842.66329
Variance	711609.7	Minimum	0
Skewness	2.66370	Maximum	8807.3
Kurtosis	11.00174	Sample Size	10000

`line=CommercialBanking`

Sample Percentiles

Percentile	Value
1	0
5	0
25	51.29272
50	363.33564
75	893.95601
95	2291.3
99	3990.7
99.5	4762.4
Percentile Method = 5	

For the retail banking business, which uses the gamma severity model, the “Sample Percentiles” table in [Output 16.2.2](#) indicates that the median operational loss of that business is about 69 units and the chance that the loss will exceed 391 units is about 1%.

Output 16.2.2 Aggregate Loss Percentiles for Retail Banking Business

line=RetailBanking	
Sample Percentiles	
Percentile	Value
1	0
5	0
25	0
50	69.26829
75	140.27686
95	273.61767
99	391.15896
99.5	439.23312
Percentile Method = 5	

When you conduct the simulation and estimation for a scenario that contains only one observation, you assume that the operating environment does not change over the period of time that is being analyzed. That assumption might be valid for shorter durations and stable business environments, but often the operating environments change, especially if you are estimating the aggregate loss over a longer period of time. So you might want to include in your scenario all the possible operating environments that you expect to see during the analysis time period. Each environment is characterized by its own set of KRI values. For example, the operating conditions might change from quarter to quarter, and you might want to estimate the aggregate loss distribution for the entire year. You start the estimation process for such scenarios by creating a scenario data set. The following DATA step creates the Work.MultiConditionScenario data set, which consists of four operating environments, one for each quarter:

```
/* Generate a scenario data set for multiple operating conditions */
data multiConditionScenario (keep=opEnvId corpKRI1 corpKRI2
    cbKRI1 cbKRI2 cbKRI3 rbKRI1 rbKRI2 rbKRI3);
    array x{8} corpKRI1 corpKRI2 cbKRI1 cbKRI2 cbKRI3 rbKRI1 rbKRI2 rbKRI3;
    call streaminit(5151);
    do opEnvId=1 to 4;
        do i=1 to dim(x);
            x(i) = rand('NORMAL');
        end;
        output;
    end;
run;
```

All four observations of the Work.MultiConditionScenario data set together form one scenario. When simulating the external counts for such multi-entity scenarios, one replication consists of the possible number of loss events that can occur as a result of each of the four operating environments. In any given replication, some operating environments might not produce any loss event or all four operating environments might produce some loss events. Assume that you use a DATA step to create the Work.LossCounts2 data set that contains, for each business line, 10,000 replications of the loss counts and that you identify each replication

by using the `Repld` variable. You can find the DATA step code to prepare the `Work.LossCounts2` data set in the PROC HPCDM sample program `hcdmex02.sas`.

Output 16.2.3 shows some observations of the `Work.LossCounts2` data set for each business line. For the first replication (`Repld=1`) of the commercial banking business, only operating environments 3 and 4 incur loss events, whereas the other environments incur no loss events. For the second replication (`Repld=2`), all operating environments incur at least one loss event. For the first replication (`Repld=1`) of the retail banking business, operating environments 2, 3, and 4 incur two, one, and three loss events, respectively.

Output 16.2.3 Snapshot of the External Counts Data with Replication Identifier

line	opEnvld	corpKRI1	corpKRI2	cbKRI2	rbKRI1	rbKRI3	repld	numloss
CommercialBanking	1	0.45224	0.40661	-0.33680	-1.08692	-2.20557	1	0
CommercialBanking	2	-0.03799	0.98670	-0.03752	1.94589	1.22456	1	0
CommercialBanking	3	-0.29120	-0.45239	0.98855	-0.37208	-1.51534	1	3
CommercialBanking	4	0.87499	-0.67812	-0.04839	-1.44881	0.78221	1	1
CommercialBanking	1	0.45224	0.40661	-0.33680	-1.08692	-2.20557	2	2
CommercialBanking	2	-0.03799	0.98670	-0.03752	1.94589	1.22456	2	5
CommercialBanking	3	-0.29120	-0.45239	0.98855	-0.37208	-1.51534	2	12
CommercialBanking	4	0.87499	-0.67812	-0.04839	-1.44881	0.78221	2	12
RetailBanking	1	0.45224	0.40661	-0.33680	-1.08692	-2.20557	1	0
RetailBanking	2	-0.03799	0.98670	-0.03752	1.94589	1.22456	1	2
RetailBanking	3	-0.29120	-0.45239	0.98855	-0.37208	-1.51534	1	1
RetailBanking	4	0.87499	-0.67812	-0.04839	-1.44881	0.78221	1	3
RetailBanking	1	0.45224	0.40661	-0.33680	-1.08692	-2.20557	2	2
RetailBanking	2	-0.03799	0.98670	-0.03752	1.94589	1.22456	2	2
RetailBanking	3	-0.29120	-0.45239	0.98855	-0.37208	-1.51534	2	0
RetailBanking	4	0.87499	-0.67812	-0.04839	-1.44881	0.78221	2	1

You can now use this simulated count data to estimate the distribution of the aggregate loss that is incurred in all four operating environments by submitting the following PROC HPCDM step, in which you specify the replication identifier variable `Repld` in the `ID=` option of the `EXTERNALCOUNTS` statement:

```

/* Estimate the distribution of the aggregate loss for both
   lines of business by using the externally simulated counts
   for the multiple operating environments */
proc hpcdm data=lossCounts2 seed=13579 print=all
           severityest=sevestEx2Best plots=density;
  by line;
  distby repld;
  externalcounts count=numloss id=repld;
  severitymodel logn gamma;
run;

```

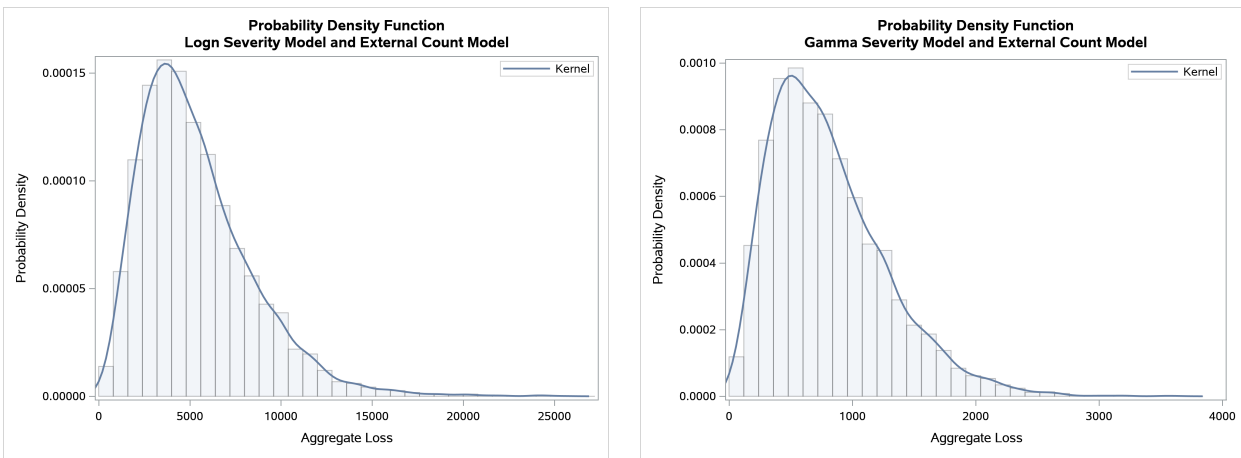
Note that when you specify the `ID=` variable in the `EXTERNALCOUNTS` statement, you must also specify that variable in the `DISTBY` statement. Within each `BY` group, for each value of the `Repld` variable, one point of the aggregate loss sample is simulated by using the process that is described in the section “[Simulation with External Counts](#)” on page 961.

The summary statistics and percentiles of the distribution of the aggregate loss, which is the aggregate of the losses across all four operating environments, are shown in [Output 16.2.4](#) for the commercial banking business. The “Input Data Summary” table indicates that there are 10,000 replications in the BY group and that a total of 145,721 loss events are generated across all replications. The “Sample Percentiles” table indicates that you can expect a median aggregate loss of 4,761 units and a worst-case loss, as defined by the 99.5th percentile, of 17,051 units from the commercial banking business when you combine losses that result from all four operating environments.

Output 16.2.4 Aggregate Loss Summary for the Commercial Banking Business in Multiple Operating Environments

The HPCDM Procedure	
line=CommercialBanking	
Input Data Summary	
Name	WORK.LOSSCOUNTS2
Observations	40000
Valid Observations	40000
Replications	10000
Total Count	145721
line=CommercialBanking	
Sample Percentiles	
Percentile	Value
1	762.69922
5	1521.1
25	3127.8
50	4760.8
75	6963.0
95	11180.1
99	15302.7
99.5	17050.8
Percentile Method = 5	

The probability density functions of the aggregate loss for the commercial and retail banking businesses are shown in [Output 16.2.5](#). In addition to the difference in scales of the losses in the two businesses, you can see that the aggregate loss that is incurred in the commercial banking business has a heavier right tail than the aggregate loss that is incurred in the retail banking business.

Output 16.2.5 Density Plots of the Aggregate Losses for Commercial Banking (left) and Retail Banking (right) Businesses**Example 16.3: Scenario Analysis with Rich Regression Effects and BY Groups**

This example illustrates scenario analysis when frequency and severity models use regression models that contain classification and interaction effects. It also illustrates how you can analyze scenarios for multiple groups of observations in one PROC HPCDM step without your having to simulate counts externally.

The example in the section “[Scenario Analysis](#)” on page 934 encodes the discrete-valued, nominal (nonordinal) variables Gender, CarType, and Education as numerical variables with an implied order. For example, a high school diploma is assigned a smaller number than an advanced degree. This method of forcing an order on otherwise nonordinal (categorical) variables is not natural and might lead to biased estimates. A more accurate approach is to treat such variables as classification variables that enter the statistical analysis or model not through their values but through their levels. For example, when you specify Education as a classification variable, the modeling process creates different parameters for the Education = ‘High School’ and Education = ‘Advanced Degree’ levels and estimates a regression coefficient for each. When you specify such variables in the CLASS statement of PROC COUNTREG and PROC SEVERITY, those procedures perform the appropriate levelization for you, which is the process of finding and transforming levels into regression parameters. For more information, see the description of the CLASS statement in Chapter 28, “[The SEVERITY Procedure](#).”

In addition to specifying nominal variables as classification (CLASS) variables, you can include interaction effects in severity and frequency models. For example, you might want to evaluate how the distribution of losses that are incurred by a policyholder with a college degree who drives an SUV differs from that of a policyholder with an advanced degree who drives a sedan. You can do this by including an interaction between CarType and Education in your severity model. Similarly, if you want to evaluate how the number of losses that a policyholder incurs per year varies by the number of annual miles for different types of cars, you can include an interaction between CarType and AnnualMiles in your frequency model. Analyzing such a rich set of regression effects can help you make more accurate predictions about the frequency and severity distributions of losses. PROC HPCDM is designed to use such rich models to simulate a more accurate distribution of the aggregate loss.

As an example of the process, first, let the following programming statements fit the severity and count models that contain a certain set of regression effects:

```
proc severity data=losses (where=(not (missing(lossAmount))))
    covout outstore=work.sevstore print=all plots=none;
    by region;
    loss lossAmount;
    class carType gender education;
    scalemodel carType gender carSafety income education*carType
        income*gender carSafety*income;
    dist logn burr;
run;

proc countreg data=losscounts covout;
    by region;
    class gender carType education;
    model numloss = age income gender carType*annualmiles education / dist=negbin;
    zeromodel numloss ~ age income carType education;
    store cstore;
run;
```

Note the following points about these statements:

- You can find the code that prepares the Work.Losses and Work.LossCounts data sets in the PROC HPCDM sample program *hcdmex03.sas*. The data sets are organized in groups of observations that represent data from two regions, East and West. You can analyze both groups at once by specifying the BY statement with Region as the BY variable.
- Both severity and count models use three CLASS variables. The severity model includes three interaction effects (Education*CarType, Income*Gender, and CarSafety*Income) and four main effects. PROC SEVERITY uses the same set of regression effects in the scale regression model of each of the two distributions that you specify in the DIST statement, which are LOGN and BURR in this example.
- The count model is a mixture of two models: a model to estimate the occurrence of zero loss events and a model to estimate nonzero counts. The zero model is a regression model with four main effects and the default logistic link function. The model for nonzero counts is a negative binomial model with one interaction effect (CarType*AnnualMiles) and four main effects.

The “Parameter Estimates” table of the lognormal severity model in [Output 16.3.1](#) for the Region=‘East’ BY group shows that Income*Gender and CarSafety*Income effects are not statistically significant. The “Parameter Estimates” table in [Output 16.3.2](#) shows that those two effects are not statistically significant for the Burr severity model also.

Output 16.3.1 Parameter Estimates for LOGN Severity Model for Region=East

region=East					
Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > t
Mu	1	4.98253	0.02861	174.16	<.0001
Sigma	1	0.48894	0.00535	91.41	<.0001
carType SUV	1	0.51772	0.03648	14.19	<.0001
carType Sedan	0	0	.	.	.
gender F	1	1.16690	0.03082	37.86	<.0001
gender M	0	0	.	.	.
carSafety	1	-0.71517	0.04599	-15.55	<.0001
income	1	-0.28528	0.03652	-7.81	<.0001
carType*education SUV Advanced Degree	1	0.44599	0.06245	7.14	<.0001
carType*education SUV College	1	0.67852	0.04416	15.36	<.0001
carType*education SUV High School	0	0	.	.	.
carType*education Sedan Advanced Degree	1	-0.49680	0.02689	-18.47	<.0001
carType*education Sedan College	1	-0.26310	0.01849	-14.23	<.0001
carType*education Sedan High School	0	0	.	.	.
income*gender F	1	0.00988	0.04010	0.25	0.8054
income*gender M	0	0	.	.	.
carSafety*income	1	-0.09390	0.06166	-1.52	0.1278

Output 16.3.2 Parameter Estimates for BURR Severity Model for Region=East

region=East					
Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > t
Theta	1	145.63709	5.74371	25.36	<.0001
Alpha	1	0.99783	0.06470	15.42	<.0001
Gamma	1	3.58743	0.09362	38.32	<.0001
carType SUV	1	0.51648	0.03701	13.96	<.0001
carType Sedan	0	0	.	.	.
gender F	1	1.16664	0.03083	37.84	<.0001
gender M	0	0	.	.	.
carSafety	1	-0.71636	0.04590	-15.61	<.0001
income	1	-0.29522	0.03639	-8.11	<.0001
carType*education SUV Advanced Degree	1	0.43696	0.06385	6.84	<.0001
carType*education SUV College	1	0.68049	0.04501	15.12	<.0001
carType*education SUV High School	0	0	.	.	.
carType*education Sedan Advanced Degree	1	-0.50160	0.02672	-18.77	<.0001
carType*education Sedan College	1	-0.26483	0.01840	-14.39	<.0001
carType*education Sedan High School	0	0	.	.	.
income*gender F	1	0.01268	0.03986	0.32	0.7504
income*gender M	0	0	.	.	.
carSafety*income	1	-0.07713	0.06162	-1.25	0.2107

The “Parameter Estimates” table of the count model in [Output 16.3.3](#) shows that the income and `Inf_income` parameters are insignificant. This implies that the income effect is not significant for the main and zero inflation parts of the count model.

The results for the `Region=‘West’` BY group are not shown here, but you can execute the sample program `hcdmex03.sas` to verify that the same parameters are statistically insignificant in severity and count models of that BY group as well. However, in general, you might find that some effects are significant for some BY groups but insignificant for other BY groups. In such cases, for more accurate results, it is recommended that you create a separate data set for each set of similar BY groups and invoke the SEVERITY, COUNTREG, and HPCDM procedures on each data set to separately analyze each set of similar BY groups.

Output 16.3.3 Count Model Parameter Estimates for `Region=East`

region=East					
Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > t
Intercept	1	1.156626	0.130641	8.85	<.0001
age	1	0.734798	0.112299	6.54	<.0001
income	1	-0.040744	0.081573	-0.50	0.6174
gender F	1	-0.999094	0.053170	-18.79	<.0001
gender M	0	0	.	.	.
annualmiles*carType SUV	1	-1.266452	0.045996	-27.53	<.0001
annualmiles*carType Sedan	1	-0.632281	0.027818	-22.73	<.0001
education Advanced Degree	1	0.418651	0.099414	4.21	<.0001
education College	1	0.709479	0.069596	10.19	<.0001
education High School	0	0	.	.	.
Inf_Intercept	1	-0.501240	0.353072	-1.42	0.1557
Inf_age	1	-0.945657	0.329950	-2.87	0.0042
Inf_income	1	-0.173541	0.233461	-0.74	0.4573
Inf_carType SUV	1	-0.693427	0.369119	-1.88	0.0603
Inf_carType Sedan	0	0	.	.	.
Inf_education Advanced Degree	1	0.668613	0.291821	2.29	0.0220
Inf_education College	1	0.474212	0.232499	2.04	0.0414
Inf_education High School	0	0	.	.	.
_Alpha	1	0.790839	0.103522	7.64	<.0001

The following modified PROC SEVERITY and PROC COUNTREG steps refit the severity and count models, respectively, after removing the insignificant effects:

```

/* Re-fit models after removing insignificant effects. */
proc severity data=losses(where=(not(missing(lossAmount))))
    covout outstore=work.sevstore print=all plots=none;
    by region;
    loss lossAmount;
    class carType gender education;
    scalemodel carType gender carSafety income education*carType;
    dist logn burrr;
run;

proc countreg data=losscounts covout;
    by region;
    class gender carType education;
    model numloss = age gender carType*annualmiles education / dist=negbin;
    zeromodel numloss ~ age carType education;
    store cstore;
run;

```

Note that the PROC SEVERITY step uses the OUTSTORE= option to store the parameter estimates in an item store. When your scale regression model contains classification or interaction effects, you must store the parameter estimates in an item store instead of storing them in an OUTEST= data set, because PROC HPCDM cannot obtain the necessary information about classification or interaction effects from an OUTEST= data set.

The “Parameter Estimates” tables in [Output 16.3.4](#) and [Output 16.3.5](#) show that all parameters are now statistically significant, most at the 95% confidence level and a few at the 90% confidence level. If you want every parameter to be significant at the 95% confidence level, then you might want to continue the process by removing the carType effect with a p -value of 0.0607 from the ZEROMODEL statement and refitting the count model. However, for the purpose of this example, the preceding models are declared to be satisfactory, and the effect selection process stops here.

You need to follow this process of model inspection and effect selection before you use the severity and count models with the HPCDM procedure. For count models, you can use the automatic effect (variable) selection feature of PROC COUNTREG. For more information, see the description of the SELECT= option in the MODEL statement of Chapter 11, “[The COUNTREG Procedure](#).” For severity models, you need to perform effect selection manually by inspecting the estimates and refitting the model after removing one or a few insignificant effects at a time until you find the final set of significant effects. Although it is not shown in this example, you can also decide which set of effects is better by comparing the fit statistics of two models; the better model might contain certain effects at lower confidence levels than the usual 95% or 90% confidence levels. In fact, the SELECT=INFO option of PROC COUNTREG uses the AIC or BIC of the entire model to select the set of effects instead of using the p -values of individual parameters. You might also want to use some domain knowledge to retain certain effects in the model even if their confidence level is not very high.

Output 16.3.4 Final LOGN Severity Model Parameter Estimates for Region=East

region=East					
Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > t
Mu	1	5.00845	0.02135	234.61	<.0001
Sigma	1	0.48908	0.00535	91.43	<.0001
carType SUV	1	0.51556	0.03642	14.16	<.0001
carType Sedan	0	0	.	.	.
gender F	1	1.17291	0.01726	67.96	<.0001
gender M	0	0	.	.	.
carSafety	1	-0.77273	0.02614	-29.56	<.0001
income	1	-0.32702	0.01962	-16.67	<.0001
carType*education SUV Advanced Degree	1	0.44870	0.06223	7.21	<.0001
carType*education SUV College	1	0.68360	0.04404	15.52	<.0001
carType*education SUV High School	0	0	.	.	.
carType*education Sedan Advanced Degree	1	-0.49572	0.02688	-18.44	<.0001
carType*education Sedan College	1	-0.26234	0.01848	-14.19	<.0001
carType*education Sedan High School	0	0	.	.	.

Output 16.3.5 Final Count Model Parameter Estimates for Region=East

region=East					
Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > t
Intercept	1	1.136175	0.124786	9.10	<.0001
age	1	0.737805	0.112339	6.57	<.0001
gender F	1	-1.001311	0.052996	-18.89	<.0001
gender M	0	0	.	.	.
annualmiles*carType SUV	1	-1.263178	0.045809	-27.57	<.0001
annualmiles*carType Sedan	1	-0.631419	0.027728	-22.77	<.0001
education Advanced Degree	1	0.400307	0.092060	4.35	<.0001
education College	1	0.703436	0.067935	10.35	<.0001
education High School	0	0	.	.	.
Inf_Intercept	1	-0.585662	0.338796	-1.73	0.0839
Inf_age	1	-0.928294	0.324629	-2.86	0.0042
Inf_carType SUV	1	-0.658089	0.350886	-1.88	0.0607
Inf_carType Sedan	0	0	.	.	.
Inf_education Advanced Degree	1	0.588511	0.269195	2.19	0.0288
Inf_education College	1	0.446600	0.228151	1.96	0.0503
Inf_education High School	0	0	.	.	.
_Alpha	1	0.785018	0.101327	7.75	<.0001

For severity models, you also need to inspect the “All Fit Statistics” table to decide which severity distributions you want to use for aggregate loss modeling. The table in [Output 16.3.6](#) shows that the lognormal distribution is the best according to the majority of fit statistics, so you can choose that. However, in some cases, you might see that the likelihood-based fit statistics ($-2 \log$ likelihood, AIC, AICC, BIC) choose one distribution and

the EDF-based statistics (KS, AD, CvM) choose another distribution. In such cases, it is recommended that before making your final decision, you conduct aggregate loss simulation by using both severity distributions and compare the summary statistics and percentiles that each severity distribution produces.

Output 16.3.6 Comparison of Severity Distributions for Region=East

region=East								
All Fit Statistics								
Distribution	-2 Log Likelihood		AIC	AICC	BIC	KS	AD	CvM
Logn	45280	* 45300	* 45300	* 45364	* 10.31771	* 613.78765	46.37913	*
Burr	45346	45368	45368	45437	10.90815	519.83495	* 49.71973	

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

After you have satisfactorily estimated the severity and frequency models, it is time to estimate the distribution of the aggregate loss by using the HPCDM procedure. The scenario data set must contain the final set of regressors that are used in both the severity model and the frequency model. Note that even if your models contain interaction effects, your scenario data set needs to contain only the columns for individual variables of the effects. PROC HPCDM internally performs *levelization* of each observation, which is the process of expanding the variable values to match them with the parameters of each effect. A typical scenario for an insurance application might consist of a large number of policyholders, but for illustration purposes, this example uses a small scenario of only a few policyholders per region. [Output 16.3.7](#) shows the contents of the Work.Scenario data set, and the following PROC HPCDM step simulates the aggregate losses for that scenario:

```
proc hpcdm data=scenario nreplicates=10000 seed=123 print=all
    severitystore=work.sevstore countstore=work.cstore
    nperturb=30;
by region;
severitymodel logn;
outsum out=agglossStats mean stddev skewness kurtosis pctlpts=(90 97.5 99.5);
run;
```

Output 16.3.7 Work.Scenario Data Set for BY-Group Processing

Obs	region	gender	carType	education	age	annualmiles	carSafety	income
1	East	F	SUV	High School	1.16	2.1540	0.29288	0.26090
2	East	F	Sedan	High School	0.86	2.3978	0.69844	0.15000
3	East	F	Sedan	Advanced Degree	0.78	1.9926	0.59421	0.58808
4	West	M	Sedan	College	0.82	1.8550	0.66849	0.15000
5	West	M	SUV	College	0.40	3.6240	0.23194	1.25274
6	West	M	Sedan	High School	0.62	3.6162	0.86477	0.42597
7	West	F	Sedan	College	0.32	3.4598	0.66294	0.36132
8	West	M	Sedan	Advanced Degree	0.90	3.2580	0.37172	0.15000

The `SEVERITYSTORE=` and `COUNTSTORE=` options specify the item stores that contain the effect information and parameter estimates of the severity and counts models, respectively, for both BY groups. The `COVOUT` option in the preceding PROC SEVERITY and PROC COUNTREG steps ensures that the respective item stores include the covariance estimates that are needed for the perturbation analysis that the `NPERTURB=` option requests.

Output 16.3.8 Aggregate Loss Simulation Results for Region=East

The HPCDM Procedure		
Severity Model: Logn		
Count Model: ZINB		
region=East		
Sample Percentile Perturbation Analysis		
Percentile	Estimate	Standard Error
1	0	0
5	0	0
25	0	0
50	151.62052	20.57120
75	492.04365	33.55686
90	917.18029	51.54978
95	1233.3	63.95801
97.5	1553.5	78.97273
99	1981.2	111.13102
99.5	2308.0	127.42680
Number of Perturbed Samples = 30		
Size of Each Sample = 10000		

Output 16.3.9 Aggregate Loss Simulation Results for Region=West

region=West		
Sample Percentile Perturbation Analysis		
Percentile	Estimate	Standard Error
1	0	0
5	0	0
25	134.16405	16.72670
50	417.89498	27.34826
75	863.13053	48.13708
90	1453.7	74.88636
95	1913.2	101.60492
97.5	2368.8	140.43218
99	2979.5	190.75595
99.5	3462.9	242.14530
Number of Perturbed Samples = 30		
Size of Each Sample = 10000		

Output 16.3.8 and Output 16.3.9 show the summary of the perturbation analysis for the two regions. You can deduce that for the collection of three policyholders in the eastern region of the specified scenario, the 97.5th percentile of their collective aggregate loss is 1553.5 ± 79 units, and for the collection of five policyholders in the western region of the specified scenario, the 99.5th percentile of their collective aggregate loss is 3462.9 ± 242.2 .

References

Fisher, R. A. (1973). *Statistical Methods for Research Workers*. 14th ed. New York: Hafner Publishing.

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

Subject Index

- BY groups
 - HPCDM procedure, [950](#)
- compound distribution modeling
 - HPCDM procedure, [926](#)
- descriptive statistics
 - HPCDM procedure, [973](#)
- HPCDM procedure
 - BY groups, [950](#)
 - descriptive statistics, [973](#)
 - ODS graph names, [980](#)
 - ODS table names, [978](#)
 - parameter perturbation analysis, [972](#)
 - scenario analysis, [957](#)
 - simulating aggregate adjusted loss distribution, [964](#)
 - simulating aggregate loss distribution, [958](#)
- ODS graph names
 - HPCDM procedure, [980](#)
- ODS table names
 - HPCDM procedure, [978](#)
- parameter perturbation analysis
 - HPCDM procedure, [972](#)
- scenario analysis
 - HPCDM procedure, [957](#)
- simulating aggregate adjusted loss distribution
 - HPCDM procedure, [964](#)
- simulating aggregate loss distribution
 - HPCDM procedure, [958](#)

Syntax Index

- ADJSAMPLEVAR= option
 - OUTPUT statement (HPCDM), [952](#)
- ADJUSTEDSEVERITY= option
 - PROC HPCDM statement, [944](#)
- BY statement
 - HPCDM procedure, [950](#)
- COUNT= option
 - EXTERNALCOUNTS statement (HPCDM), [951](#)
- COUNTSTORE= option
 - PROC HPCDM statement, [944](#)
- DATA= option
 - PROC HPCDM statement, [945](#)
- DISTBY statement
 - HPCDM procedure, [951](#)
- EXTERNALCOUNTS statement
 - HPCDM procedure, [951](#)
- HPCDM procedure, [942](#)
 - DISTBY statement, [951](#)
 - EXTERNALCOUNTS statement, [951](#)
 - OUTPUT statement, [952](#)
 - OUTSUM statement, [953](#)
 - PERFORMANCE statement, [956](#)
 - SEVERITYMODEL statement, [956](#)
 - syntax, [942](#)
- HPCDM procedure, EXTERNALCOUNTS statement
 - COUNT= option, [951](#)
 - ID= option, [951](#)
- HPCDM procedure, OUTPUT statement
 - ADJSAMPLEVAR= option, [952](#)
 - OUT= option, [952](#)
 - PERTURBOUT option, [952](#)
 - SAMPLEVAR= option, [952](#)
- HPCDM procedure, OUTSUM statement
 - OUT= option, [953](#)
 - PCTLNAME= option, [954](#)
 - PCTLNDEC= option, [955](#)
 - PCTLPTS= option, [954](#)
- HPCDM procedure, PROC HPCDM statement, [944](#)
 - ADJUSTEDSEVERITY= option, [944](#)
 - COUNTSTORE= option, [944](#)
 - DATA= option, [945](#)
 - MAXCOUNTDRAW= option, [945](#)
 - NOPRINT option, [945](#)
 - NPERTURBEDSAMPLES= option, [945](#)
 - NREPLICATES= option, [946](#)
 - PCTLDEF= option, [946](#)
 - PERTURBMETHOD= option, [946](#)
 - PLOTS= option, [947](#)
 - PRINT= option, [948](#)
 - SEED= option, [949](#)
 - SEVERITYEST= option, [949](#)
 - SEVERITYSTORE= option, [950](#)
 - VARDEF= option, [950](#)
- ID= option
 - EXTERNALCOUNTS statement (HPCDM), [951](#)
- MAXCOUNTDRAW= option
 - PROC HPCDM statement, [945](#)
- NOPRINT option
 - PROC HPCDM statement, [945](#)
- NPERTURBEDSAMPLES= option
 - PROC HPCDM statement, [945](#)
- NREPLICATES= option
 - PROC HPCDM statement, [946](#)
- OUT= option
 - OUTPUT statement (HPCDM), [952](#)
 - OUTSUM statement (HPCDM), [953](#)
- OUTPUT statement
 - HPCDM procedure, [952](#)
- OUTSUM statement
 - HPCDM procedure, [953](#)
- PCTLDEF= option
 - PROC HPCDM statement, [946](#)
- PCTLNAME= option
 - OUTSUM statement (HPCDM), [954](#)
- PCTLNDEC= option
 - OUTSUM statement (HPCDM), [955](#)
- PCTLPTS= option
 - OUTSUM statement (HPCDM), [954](#)
- PERFORMANCE statement
 - HPCDM procedure, [956](#)
- PERTURBMETHOD= option
 - PROC HPCDM statement, [946](#)
- PERTURBOUT option
 - OUTPUT statement (HPCDM), [952](#)
- PLOTS= option
 - PROC HPCDM statement, [947](#)
- PRINT= option
 - PROC HPCDM statement, [948](#)

PROC HPCDM statement, [944](#), *see* HPCDM
procedure

SAMPLEVAR= option
 OUTPUT statement (HPCDM), [952](#)

SEED= option
 PROC HPCDM statement, [949](#)

SEVERITYEST= option
 PROC HPCDM statement, [949](#)

SEVERITYMODEL statement
 HPCDM procedure, [956](#)

SEVERITYSTORE= option
 PROC HPCDM statement, [950](#)

VARDEF= option
 PROC HPCDM statement, [950](#)