Analysis of Count Data: A Business Perspective

George J. Hurley, The Hershey Company

ABSTRACT

While count data frequently is analyzed in a Pharma environment, there are also practical business applications for analysis of count data. Poisson Regression and Negative Binomial Regression are two methods generally used for this type of data. In SAS, these methods are implemented using PROC GENMOD or PROC COUNTREG, although the new PROC FMM can also be used. This talk will discuss how count data arises in a business environment, some assumptions involved in using methods such as Poisson Regression and Negative Binomial Regression (as well as their Zero-Inflated analogs), and how to implement these methods in SAS. The issue of overdispersion and some common ways to check for it will also be discussed. While the data used will be focused on business applications, the methods and SAS code are broadly applicable to other fields, such as Pharma.

INTRODUCTION

Count data arises in any number of ways in a business setting. For example, the number of customers coming into a retailer in a given hour, the number of new product launches in a given month, and the number of items purchased in a transaction at a grocery store may all be considered count data. Generally speaking, the Poisson and Negative Binomial distributions are utilized to analyze this type of data via Poisson regression and Negative Binomial regression models.

Zero-inflated count data arises when the underlying data generating process produces zeros at some rate in addition to the normal small number of zeros that may be generated from the underlying process itself. These are called "structural" zeros.

For instance, while the number of items purchased in a transaction at a grocery store is count data (all paying customers necessarily purchase something), the number of items purchased by each person entering a store, perhaps a fashion retailer, where "looking around" is popular, may be zero-inflated count data. That is, many customers enter the store to "look around" and purchase zero items, creating structural zeros.

The underlying process of these customers can be considered as coming from two separate processes, first with some probability the customer will either transact or will not transact (this ban be thought of as a Bernoulli distribution). If the customer transacts, then the number of items purchased can be thought to arise from a Poisson distribution, for instance.

This is formalized below, courtesy of Wikipedia

$$\Pr(y_i = 0) = \pi_i + (1-\pi_i)e^{-\lambda_i}$$
$$\Pr(y_i = h_i) = (1-\pi_i)\frac{\lambda_i^{h_i} e^{-\lambda_i}}{h_i!}, h_i \geq 1$$

where the outcome variable y_i has any non-negative integer value; λ_i is the expected Poisson count for the i^{th} individual; π_i is the probability of extra zeros. [1]

It is noted that for $\Pr(y_i=0)$, the component $(1-\pi_i)e^{-\lambda_i}$ generates the zeros that arise from the Poisson distribution, where $e^{-\lambda_i}$ is generated by using the pdf of the Poisson distribution to generate the probability at $h_i=0$, and $(1- \pi_i)$ is the probability that a structural zero does not occur; π_i, of course, represents the structural zeros . It is obvious that the second piece is the product of the probability a structural zero does not occur and the pdf of the Poisson distribution.

A similar structure follows for the Negative Binomial distribution.

Models that use this structure are known as "Zero-Inflated Poisson regression models" or "ZIP" models. Likewise, models using the Negative Binomial rather than the Poisson distribution are known as "Zero-Inflated Negative Binomial regression models" or "ZINB" models.

In SAS, Poisson regression and Negative Binomial regression models are generally implemented using Proc Genmod or Proc Countreg. ZIP and ZINB models are generally implemented via these two procedures as well, although the FMM Procedure can also be used to implement them, as well as other variants of these analysis that may be useful in some situations.

SAMPLE DATA

Consider the following code which will generate all data used in this paper:

```
data dd1.poisson_data;
do i=1 to 40;
store_type="Big";
shelf_set="New";
n_people_poi=ranpoi(1978,27);
n_people_inf=round(ranpoi(1978,21)+sqrt(10)*rannor(1971),1);
if i<6 then n_people_zp=0;
else n_people_zp=n_people_poi;
output;
end;
do i=1 to 40;
store_type="Big";
shelf_set="Old";
n_people_poi=ranpoi(2009,23);
n_people_inf=round(ranpoi(2009,23)+sqrt(10)*rannor(2005),1);
if i<8 then n_people_zp=0;
else n_people_zp=n_people_poi;
output;
end;
do i=1 to 30;
store_type="Sml";
shelf_set="New";
n_people_poi=ranpoi(2006,17);
n_people_inf=round(ranpoi(2006,17)+sqrt(10)*rannor(2013),1);
if i<5 then n_people_zp=0;
else n_people_zp=n_people_poi;
output;
end;
do i=1 to 30;
store_type="Sml";
shelf_set="Old";
n_people_poi=ranpoi(1999,13);
n_people_inf=round(ranpoi(1999,13)+sqrt(10)*rannor(2012),1);
if i<7 then n_people_zp=0;
else n_people_zp=n_people_poi;
output;
end;

run;
```

The code simulates four overall groups, two store types, “Sml” (small) and “Big” (big) and two shelf sets “New” and “Old”. For each group, the code creates three different response variables: “n_people_poi,” which simulates number of items purchased for each group according to a Poisson distribution with differing parameters by group, “n_people_inf” also simulates number of items purchased data using the Poisson distribution, but intentionally inflates the variance of these data, “n_people_zp,” creates a process to “zero-inflate” “n_people_poi”. The goal is to test for differences in the two shelf sets and determine if the new shelf set is leading to incremental items purchased.

DATA DIAGNOSTICS

Before one begins modeling, it is always a best practice to understand the nature of the data. Generally, The Univariate Procedure is a good way in SAS to look at data structure. Here, we will look at the three response variables simulated above, “n_people_poi”, “n_people_inf”, and “n_people_zp”.

The Proc Univariate code below will produce useful summary statistics and histograms for this data:

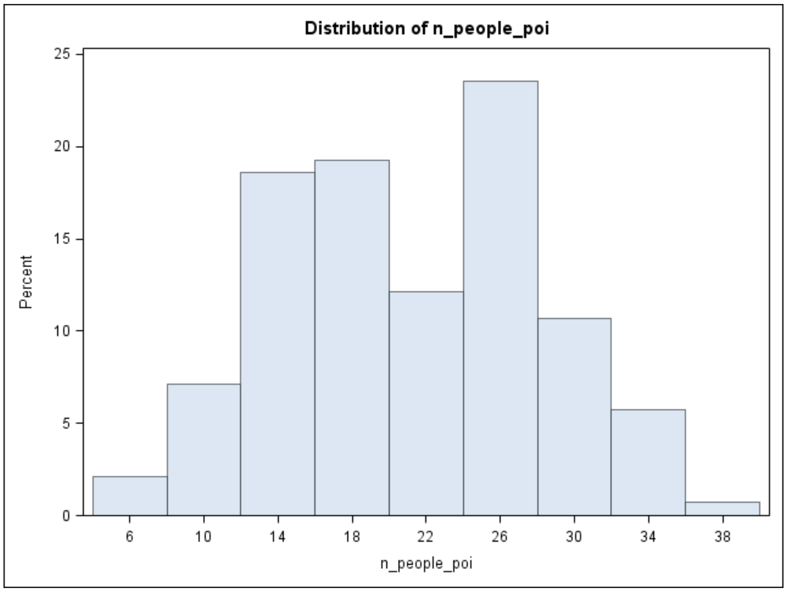
```
proc univariate data=dd1.poisson_data;
var n_people_poi n_people_inf n_people_zp;
histogram n_people_poi n_people_inf n_people_zp;
run;
```

Output similar to below is generated for each variable in the var statement. Here we note that the mean of the variable “n_people_poi” is 20.6 and the variance is “52.1”. That is, the variance is substantially greater than the mean. This will be discussed in modeling context later in this paper, but should be noted at this point as potentially related to overdispersion in a Poisson regression model.

```
                         The UNIVARIATE Procedure
                         Variable:  n_people_poi

                                 Moments

            N                         140    Sum Weights                140
            Mean               20.6357143    Sum Observations          2889
            Std Deviation      7.22028589    Variance            52.1325283
            Skewness           0.06672132    Kurtosis            -0.7658642
            Uncorrected SS          66863    Corrected SS        7246.42143
            Coeff Variation    34.9892705    Std Error Mean      0.61022553


                       Basic Statistical Measures

             Location                    Variability

         Mean     20.63571     Std Deviation            7.22029
         Median   20.50000     Variance                52.13253
         Mode     13.00000     Range                   32.00000
                               Interquartile Range     11.00000


                       Tests for Location: Mu0=0

              Test           -Statistic-    -----p Value------

              Student's t    t  33.81654    Pr > |t|    <.0001
              Sign           M        70    Pr >= |M|   <.0001
              Signed Rank    S      4935    Pr >= |S|   <.0001


                        Quantiles (Definition 5)

                         Quantile      Estimate

                         100% Max          37.0
                         99%               35.0
                         95%               32.5
                         90%               30.0
                         75% Q3            26.0
                         50% Median        20.5
                         25% Q1            15.0
                         10%               12.0
                         5%                 9.0
                         1%                 6.0
```

```
0% Min            5.0
```

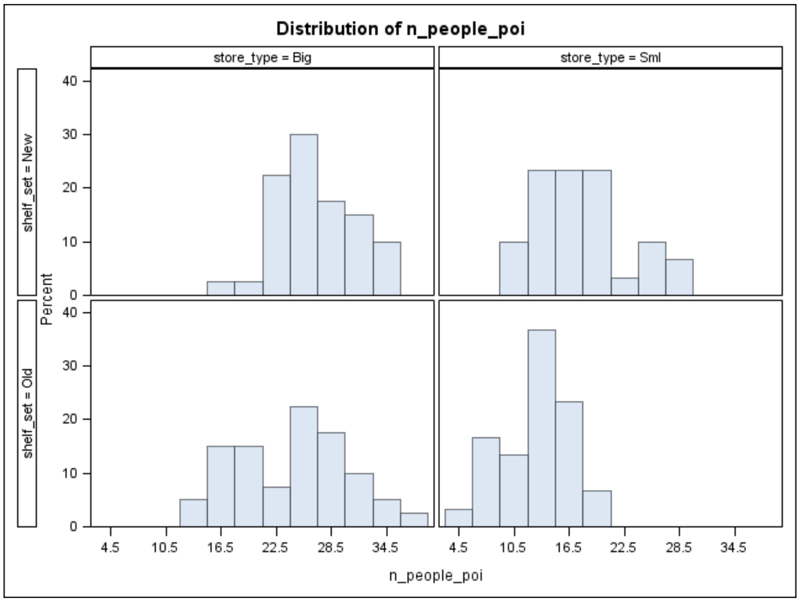
The histogram below shows evidence of bimodality. This may suggest that each level of the independent variables may have their own distribution (which is known to be true based on the simulations.



The following code will create univariate output and histograms at each level of the independent variables for the response "n_people_poi".

```
proc univariate data=dd1.poisson_data;
class shelf_set store_type;
var n_people_poi;
histogram n_people_poi;
run;
```

The histograms generated is below



It is seen from this histrogram that there are somewhat different distributions for each level of the two independent variables, as expected from the simulation.

It is a best practice to look at any response variable in this fashion prior to commencing model building.

OVERDISPERSION AND POISSON REGRESSION

As stated above, Poisson regression is a method to analyze count data and is generally implemented through Proc Genmod or Proc Countreg. Since the above data is count data, Poisson regression is a natural place to start. The following code runs a basic Poisson regression for this data:

Model 1: Simple Poisson Regression

```
proc genmod data=dd1.poisson_data;
class store_type shelf_set;
model n_people_poi=shelf_set / dist=poisson link=log;
lsmeans shelf_set / ilink;
run;
```

> In the model statement, dist=Poisson indicates the Poisson distribution is to be used. Generally speaking, the link function used with the Poisson distribution is the log link, as it is the canonical link function. Since a link function is used, ilink is used in the lsmeans statement to produce means output back on the original scale.

Note we do not include the variable store_type in the model. This is for pedagogical purposes. Following is partial output from proc genmod:

Output 1: Simple Poisson Regression

```
                    The GENMOD Procedure

                     Model Information

          Data Set                     DD1.POISSON_DATA
          Distribution                          Poisson
          Link Function                             Log
          Dependent Variable                n_people_poi


         Number of Observations Read          140
         Number of Observations Used          140


                  Class Level Information

            Class            Levels    Values

            store_type            2    Big Sml
            shelf_set             2    New Old


                   Parameter Information

          Parameter       Effect       shelf_set

          Prm1            Intercept
          Prm2            shelf_set    New
          Prm3            shelf_set    Old


          Criteria For Assessing Goodness Of Fit
```

Criterion	DF	Value	Value/DF
Deviance	138	345.1045	2.5008
Scaled Deviance	138	345.1045	2.5008
Pearson Chi-Square	138	337.9961	2.4492
Scaled Pearson X2	138	337.9961	2.4492
Log Likelihood		5866.8141	
Full Log Likelihood		-508.8216	
AIC (smaller is better)		1021.6433	
AICC (smaller is better)		1021.7309	
BIC (smaller is better)		1027.5266	

Note the Pearson Chi-Square statistic for reference at the end of this paper.

Algorithm converged.

The first piece of information that must be examined in any Poisson regression is the deviance. An examination of deviance will suggest if a condition known as overdispersion is present. It is not common for overdispersion to be present in count data modeling[2]. Overdispersion arises out of the fact that the Poisson distribution has only one parameter, λ, which is equal to both the mean and the variance. As only one parameter is thus estimated, it is sometimes the case that in the data, the variance is greater than the mean. This can occur for several reasons, although a common reason is subject heterogeneity[2].

When deviance is divided by its degrees of freedom (df), the result should be near 1 if no overdispersion is present. A result greater than 1 indicates the presence of overdispersion. While there is no concrete guidance this author has found for a threshold, the following method of examining deviance was found on the web and is appropriate in most cases. Essentially, the deviance is asymptotically chi-square distributed, so one can develop an asymptotic chi-square test of the deviance using its degrees of freedom[3]. The same applies for the Pearson statistic. However, in order for the Pearson statistic and the deviance to be distributed as chi-square, there must be sufficient replication within the subpopulations. When this is not true, the data are sparse, and the p-values for these statistics are not valid and should be ignored.[4] Here, there is replication within the subgroups and are comfortable with this relationship. Here the Excel function =CHIDIST(345.1045, 138) can be used to determine that this (one-tailed) translates to a p-value of 9.8e-20. With this p-value being significant, coupled with a ratio of the deviance/df of 2.5, one would conclude overdispersion is present here. This is consistent with the fact that there is subject heterogeneity present (by design). That is, since we only use shelf set as a predictor, our data is in fact a mixture of Poisson distributions.

There are competing thoughts on the value of calculating this Chi-Square statistic. While I have shown above that it is possible to compute, many statisticians prefer not to use this approach, and indeed, The SAS System does not provide this statistic while it would be easy to calculate. Overall, the opinion of this author is that if there is large sample size, there may be value in computing it, however, it should not be treated as "the determining factor", but rather, as no more than an additional piece of information.

On the topic of additional information, like deviance, another commonly used metric to examine for evidence of overdispersion is the Scaled Pearson Chi-Square statistic. Similar to deviance, one wishes to look for a ratio near to 1 for this statistic. There are certainly very good statisticians who prefer this statistic to deviance. This author's advice is to consider both statistics.

Based on any of these statistics, there is evidence of overdispersion in this model. This is a concern because proc genmod fixes the scale parameter in proc genmod to 1, and essentially reduces the variance used in the test. Hence, the Type I error rate becomes inflated. It would not be a good idea to report results from this model.

There are several ways that overdispersion can be overcame. Not all will work for any one circumstance. The following four will be discussed in this paper:

(1) Re-specify the model to include necessary predictors

(2) Use the scale=deviance option on the model statement in proc genmod

(3) Use a distribution that allows for two parameters to be estimated, such as the Negative Binomial

(4) Use a mixture of distributions (this includes zero-inflated models)

There are several ways that overdispersion can be overcame. Not all will work for any one circumstance. The following four will be discussed in this paper:

1. MODEL SPECIFICATION

The easiest method to deal with over-dispersion is model specification. In many instances, however, all known relevant predictors have already been included and this is not a viable option. For the example above, this is a viable option. The proc genmod code below considers this:

Model 2: Poisson Regression Accounting for All Relevant Predictors

```
proc genmod data=dd1.poisson_data;
class store_type shelf_set;
model n_people_poi=store_type shelf_set store_type*shelf_set/ dist=poisson link=log;
lsmeans store_type*shelf_set /  pdiff ilink;
run;
```

The results below indicate that the deviance has dropped to 163.5 (noting the new df). The ratio of deviance to df is now 1.2. This seems much more in-line with a non-overdispersed model. Running the chi-square test, it can be seen a p-value of 0.054 is achieved. Given the low deviance level and non-significant p-value, it is reasonable to believe this model is acceptable in terms of its variance estimation and to utilize the results. Since an interaction term was specified and significant, it is appropriate to look at the comparison of Big Old to Big New and Small Old to Small New separately. It must be noted that on the surface, the data simulation did not appear to produce an interaction of shelf set and store size. However, recall a link function is used and the model is not running in traditional space, but rather log-space. It can be seen from the highlighted results that the big stores have a mean difference of around 2.4 additional items, while the smaller stores have an incremental 5.1 items. Both are statistically significant.

Output 2: Poisson Regression Accounting for All Relevant Predictors

The GENMOD Procedure

Model Information

Data Set	DD1.POISSON_DATA
Distribution	Poisson
Link Function	Log
Dependent Variable	n_people_poi

Number of Observations Read	140
Number of Observations Used	140

Class Level Information

Class	Levels	Values
store_type	2	Big Sml
shelf_set	2	New Old

Parameter Information

Parameter	Effect	store_ type	shelf_set
Prm1	Intercept		
Prm2	store_type	Big	
Prm3	store_type	Sml	
Prm4	shelf_set		New
Prm5	shelf_set		Old
Prm6	store_type*shelf_set	Big	New

Prm7	store_type*shelf_set	Big	Old
Prm8	store_type*shelf_set	Sml	New
Prm9	store_type*shelf_set	Sml	Old

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	136	163.4923	1.2021
Scaled Deviance	136	163.4923	1.2021
Pearson Chi-Square	136	161.2446	1.1856
Scaled Pearson X2	136	161.2446	1.1856
Log Likelihood		5957.6202	
Full Log Likelihood		-418.0156	
AIC (smaller is better)		844.0311	
AICC (smaller is better)		844.3274	
BIC (smaller is better)		855.7977	

The GENMOD Procedure

Algorithm converged.

Analysis Of Maximum Likelihood Parameter Estimates

Parameter			DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept			1	2.5150	0.0519	2.4132	2.6168	2346.67	<.0001
store_type	Big		1	0.6515	0.0612	0.5315	0.7715	113.22	<.0001
store_type	Sml		0	0.0000	0.0000	0.0000	0.0000	.	.
shelf_set	New		1	0.3453	0.0679	0.2123	0.4783	25.90	<.0001
shelf_set	Old		0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Big	New	1	-0.2489	0.0813	-0.4083	-0.0895	9.37	0.0022
store_type*shelf_set	Big	Old	0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Sml	New	0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Sml	Old	0	0.0000	0.0000	0.0000	0.0000	.	.
Scale			0	1.0000	0.0000	1.0000	1.0000		

NOTE: The scale parameter was held fixed.

store_type*shelf_set Least Squares Means

store_ type	shelf_set	Estimate	Standard Error	z Value	Pr > |z|	Mean	Standard Error of Mean
Big	New	3.2629	0.03093	105.48	<.0001	26.1250	0.8082
Big	Old	3.1665	0.03246	97.55	<.0001	23.7250	0.7701
Sml	New	2.8603	0.04369	65.48	<.0001	17.4667	0.7630
Sml	Old	2.5150	0.05192	48.44	<.0001	12.3667	0.6420

Differences of store_type*shelf_set Least Squares Means

store_ type	shelf_set	_store_ type	_shelf_ set	Estimate	Standard Error	z Value	Pr > |z|

Big	New	Big	Old	0.09636	0.04484	2.15	0.0316
Big	New	Sml	New	0.4026	0.05353	7.52	<.0001
Big	New	Sml	Old	0.7479	0.06043	12.38	<.0001
Big	Old	Sml	New	0.3062	0.05443	5.63	<.0001
Big	Old	Sml	Old	0.6515	0.06123	10.64	<.0001
Sml	New	Sml	Old	0.3453	0.06785	5.09	<.0001

2. SCALE=DEVIANCE OPTION

Since overdispersion is essentially an issue with the variance, one method to address it is via the scale= option in the model statement in proc genmod. This method assumes that the sample sizes in each subpopulation are approximately equal.[4]

The scale=deviance option adjusts the parameter covariance matrix and the likelihood function by the deviance. "Specifying SCALE=DEVIANCE or SCALE=D is the same as specifying the DSCALE option. This fixes the scale parameter at a value of 1 in the estimation procedure.

After the parameter estimates are determined, the exponential family dispersion parameter is assumed to be given by the deviance divided by the degrees of freedom. All statistics such as standard errors and likelihood ratio statistics are adjusted appropriately."[5]

In simple terms, think of this as inflating the estimated variance "back up" to where it should be.

Consider:

Model 3: Poisson Regression – Response Variable with Inflated Variance

```
proc genmod data=dd1.poisson_data;
class store_type shelf_set;
model n_people_inf=store_type shelf_set store_type*shelf_set/ dist=poisson link=log;
lsmeans store_type*shelf_set / ilink;
run;
```

Here, the response variable, "n_people_inf" is used. This variable had its variance intentionally inflated. Running this correctly specified model, still yields the following:

Output 3: Poisson Regression – Response Variable with Inflated Variance – Goodness of Fit

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	136	259.0693	1.9049
Scaled Deviance	136	259.0693	1.9049
Pearson Chi-Square	136	243.9161	1.7935
Scaled Pearson X2	136	243.9161	1.7935
Log Likelihood		5693.7559	
Full Log Likelihood		-460.3821	
AIC (smaller is better)		928.7642	
AICC (smaller is better)		929.0605	
BIC (smaller is better)		940.5308	

Again, this represents overdispersion. However, for comparison, consider the parameter estimates and LS Means table generated from this model.

Output 3 (Continued): Poisson Regression – Response Variable with Inflated Variance – Goodness of Fit

Analysis Of Maximum Likelihood Parameter Estimates

Parameter			DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept			1	2.5284	0.0516	2.4273	2.6295	2403.68	<.0001
store_type	Big		1	0.5547	0.0617	0.4338	0.6756	80.85	<.0001
store_type	Sml		0	0.0000	0.0000	0.0000	0.0000	.	.
shelf_set	New		1	0.2316	0.0691	0.0963	0.3670	11.25	0.0008
shelf_set	Old		0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Big	New	1	-0.0225	0.0827	-0.1847	0.1396	0.07	0.7852
store_type*shelf_set	Big	Old	0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Sml	New	0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Sml	Old	0	0.0000	0.0000	0.0000	0.0000	.	.
Scale			0	1.0000	0.0000	1.0000	1.0000		

NOTE: The scale parameter was held fixed.

store_ type	shelf_set	Estimate	Standard Error	z Value	Pr > |z|	Mean	Standard Error of Mean
Big	New	3.2921	0.03049	107.99	<.0001	26.9000	0.8201
Big	Old	3.0831	0.03384	91.09	<.0001	21.8250	0.7387
Sml	New	2.7600	0.04593	60.09	<.0001	15.8000	0.7257
Sml	Old	2.5284	0.05157	49.03	<.0001	12.5333	0.6464

Now consider:

Model 4: Poisson Regression – Response Variable with Inflated Variance Scale=Deviance Option

```
proc genmod data=dd1.poisson_data;
class store_type shelf_set;
model n_people_inf=store_type shelf_set store_type*shelf_set/ dist=poisson link=log
scale=deviance;
lsmeans store_type*shelf_set / ilink;
run;
```

Output 4: Poisson Regression – Response Variable with Inflated Variance Scale=Deviance Option

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	136	259.0693	1.9049
Scaled Deviance	136	136.0000	1.0000
Pearson Chi-Square	136	243.9161	1.7935
Scaled Pearson X2	136	128.0453	0.9415
Log Likelihood		2988.9717	
Full Log Likelihood		-460.3821	
AIC (smaller is better)		928.7642	
AICC (smaller is better)		929.0605	
BIC (smaller is better)		940.5308	

Analysis Of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq

Intercept			1	2.5284	0.0712	2.3889	2.6679	1261.83	<.0001
store_type	Big		1	0.5547	0.0851	0.3878	0.7215	42.44	<.0001
store_type	Sml		0	0.0000	0.0000	0.0000	0.0000	.	.
shelf_set	New		1	0.2316	0.0953	0.0448	0.4184	5.90	0.0151
shelf_set	Old		0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Big	New	1	-0.0225	0.1142	-0.2463	0.2012	0.04	0.8435
store_type*shelf_set	Big	Old	0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Sml	New	0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Sml	Old	0	0.0000	0.0000	0.0000	0.0000	.	.
Scale			0	1.3802	0.0000	1.3802	1.3802		

NOTE: The scale parameter was estimated by the square root of DEVIANCE/DOF.

store_type*shelf_set Least Squares Means

store_ type	shelf_set	Estimate	Standard Error	z Value	Pr > |z|	Mean	Standard Error of Mean
Big	New	3.2921	0.04208	78.24	<.0001	26.9000	1.1318
Big	Old	3.0831	0.04671	66.00	<.0001	21.8250	1.0195
Sml	New	2.7600	0.06339	43.54	<.0001	15.8000	1.0016
Sml	Old	2.5284	0.07118	35.52	<.0001	12.5333	0.8921

First, note that SAS states it estimated the scale parameter in Model 4, rather than fixing it at 1 in Model3.

In Model 4, it is also clear that all standard errors are larger. This is because the variance has been inflated to account for the deviance. Note the parameter estimates and estimates of the mean have not changed. It should be noted that the non-scaled model fit statistics have not changed. This is because the parameter estimates were still modeled in the same way as before. However, the scaled model-fit statistics have changed. Since the model was scaled by the deviance, the scaled deviance is now 1. Likewise, the scaled Pearson Chi-Square statistics has also decreased.

Another option for scaling is to scale the model by the Pearson Chi-Square statistic.

3. USE A DISTRIBUTION THAT ESTIMATES TWO PARAMETERS

The Negative Binomial distribution uses two parameters, k and μ, with $E(Y)= \mu$ and $Var(Y)= \mu+\mu^2/k$. k^{-1} is called the dispersion parameter. As $k^{-1} \rightarrow 0$, the Negative Binomial distribution converges to the Poisson distribution.[2]

The Negative Binomial is often used as a replacement for the Poisson distribution in instances of overdispersion. Negative Binomial regression will often generate similar parameter estimates to Poisson regression. Yet, Negative Binomial regression often better reflects the uncaptured overdispersion in Poisson regression models.[2]

Consider Model 4, it can be modeled using Negative Binomial regression using the following code:

Model 5: Negative Binomial Regression

```
proc genmod data=dd1.poisson_data;
class store_type shelf_set;
model n_people_inf=store_type shelf_set store_type*shelf_set/ dist=nb link=log;
lsmeans store_type*shelf_set / ilink;
run;
```

Output 5: Negative Binomial Regression

The GENMOD Procedure

Model Information

Data Set	DD1.POISSON_DATA
Distribution	Negative Binomial
Link Function	Log
Dependent Variable	n_people_inf

Number of Observations Read	140
Number of Observations Used	140

Class Level Information

Class	Levels	Values
store_type	2	Big Sml
shelf_set	2	New Old

Parameter Information

Parameter	Effect	store_ type	shelf_set
Prm1	Intercept		
Prm2	store_type	Big	
Prm3	store_type	Sml	
Prm4	shelf_set		New
Prm5	shelf_set		Old
Prm6	store_type*shelf_set	Big	New
Prm7	store_type*shelf_set	Big	Old
Prm8	store_type*shelf_set	Sml	New
Prm9	store_type*shelf_set	Sml	Old

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	136	162.3178	1.1935
Scaled Deviance	136	162.3178	1.1935
Pearson Chi-Square	136	147.0568	1.0813
Scaled Pearson X2	136	147.0568	1.0813
Log Likelihood		5704.3629	
Full Log Likelihood		-449.7751	
AIC (smaller is better)		909.5502	
AICC (smaller is better)		909.9980	
BIC (smaller is better)		924.2584	

The GENMOD Procedure

Algorithm converged.

Analysis Of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits	Wald Chi-Square	Pr > ChiSq

Intercept			1	2.5284	0.0623	2.4062	2.6506	1644.86	<.0001
store_type	Big		1	0.5547	0.0772	0.4035	0.7059	51.69	<.0001
store_type	Sml		0	0.0000	0.0000	0.0000	0.0000	.	.
shelf_set	New		1	0.2316	0.0850	0.0650	0.3982	7.43	0.0064
shelf_set	Old		0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Big	New	1	-0.0225	0.1055	-0.2294	0.1843	0.05	0.8308
store_type*shelf_set	Big	Old	0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Sml	New	0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Sml	Old	0	0.0000	0.0000	0.0000	0.0000	.	.
Dispersion			1	0.0368	0.0115	0.0200	0.0679		

NOTE: The negative binomial dispersion parameter was estimated by maximum likelihood.

store_type*shelf_set Least Squares Means

store_ type	shelf_set	Estimate	Standard Error	z Value	Pr > |z|	Mean	Standard Error of Mean
Big	New	3.2921	0.04301	76.55	<.0001	26.9000	1.1569
Big	Old	3.0831	0.04545	67.83	<.0001	21.8250	0.9919
Sml	New	2.7600	0.05776	47.78	<.0001	15.8000	0.9127
Sml	Old	2.5284	0.06234	40.56	<.0001	12.5333	0.7814

It is immediately seen, that the deviance is much smaller, and ratio of deviance to df much nearer 1. The same can be said of the Pearson Chi-Square statistic. In this instance, it should be noted that the parameter estimates are identical to what was seen before, yet the standard errors have changed, which is as expected. Additionally, there is a modeled estimate of the dispersion parameter, k^{-1}. This estimate, 0.0368 indicates that at a predicted μ_hat, the estimated variance is μ_hat+ 0.0368*μ_hat^2. This is a way of quantifying how much overdispersion was present in the Poisson model that was captured in the Negative Binomial model.[2]

Finally, it can be seen that standard model fit metrics, such as BIC, also suggest this model is a better fit. It should be noted that while BIC is a useful model fit criteria for models with a larger number of observations, the AICC is a better fit statistic for models with smaller number of observations.

4. MIXTURES OF DISTRIBUTIONS

It is often the case that data arise from a mixture of distributions. As discussed in the introduction, these commonly occur in the form of having an inflated number of zeros in the data. Often, this is known to the analyst in advance, but sometimes it is discovered through overdispersion detection. Consider the following code using the zero-inflated response variable:

Model 6: Zero-inflated Data with a Standard Poisson Regression

```
proc genmod data=dd1.poisson_data;
class store_type shelf_set;
model n_people_zp=store_type shelf_set store_type*shelf_set/ dist=poisson link=log;
lsmeans store_type*shelf_set / ilink;
run;
```

Output 6: Zero-inflated Data with a Standard Poisson Regression – Fit Statistics Only

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	136	948.1527	6.9717
Scaled Deviance	136	948.1527	6.9717

Pearson Chi-Square	136	605.4346	4.4517
Scaled Pearson X2	136	605.4346	4.4517
Log Likelihood		4646.1529	
Full Log Likelihood		-757.9261	
AIC (smaller is better)		1523.8523	
u			
BIC (smaller is better)		1535.6189	

Clearly over-dispersion appears to be present.

This may prompt investigation of the data, using proc freq, proc means, etc., to assess is zero-inflation is the cause. Here, since the data is simulated, we know zero-inflation is the cause.

First, however, a brief segue. What if a Negative Binomial regression was run here? It can be seen that again, overdispersion is much more contained. The Pearson Chi-Square, like deviance should have a value/df ratio near 1. Here, it is somewhat low, which is somewhat concerning, especially in its relationship to the deviance/df and may be indicative of over-inflated variance. Given that this is an ill-specified model, there should be no surprise at the odd results. Results follow and will be referenced.

Model 7: Zero-inflated Data with a Standard Negative Binomial Regression

```
proc genmod data=dd1.poisson_data;
class store_type shelf_set;
model n_people_zp=store_type shelf_set store_type*shelf_set/ dist=nb link=log;
lsmeans store_type*shelf_set / ilink;
run;
```

Output 7: Zero-inflated Data with a Standard Negative Binomial Regression

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	136	185.0223	1.3605
Scaled Deviance	136	185.0223	1.3605
Pearson Chi-Square	136	52.6985	0.3875
Scaled Pearson X2	136	52.6985	0.3875
Log Likelihood		4869.1641	
Full Log Likelihood		-534.9149	
AIC (smaller is better)		1079.8299	
AICC (smaller is better)		1080.2776	
BIC (smaller is better)		1094.5381	

The GENMOD Procedure

Algorithm converged.

Analysis Of Maximum Likelihood Parameter Estimates

Parameter		DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept		1	2.3125	0.1563	2.0063	2.6188	219.04	<.0001
store_type	Big	1	0.6489	0.2038	0.2494	1.0484	10.13	0.0015
store_type	Sml	0	0.0000	0.0000	0.0000	0.0000	.	.

shelf_set	New		1	0.4175	0.2184	-0.0106	0.8456	3.65	0.0559
shelf_set	Old		0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Big	New	1	-0.2532	0.2859	-0.8137	0.3072	0.78	0.3758
store_type*shelf_set	Big	Old	0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Sml	New	0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Sml	Old	0	0.0000	0.0000	0.0000	0.0000	.	.
Dispersion			1	0.6334	0.0999	0.4650	0.8629		

NOTE: The negative binomial dispersion parameter was estimated by maximum likelihood.

store_type*shelf_set Least Squares Means

store_ type	shelf_set	Estimate	Standard Error	z Value	Pr > |z|	Mean	Standard Error of Mean
Big	New	3.1257	0.1301	24.02	<.0001	22.7751	2.9637
Big	Old	2.9614	0.1309	22.63	<.0001	19.3251	2.5293
Sml	New	2.7300	0.1526	17.89	<.0001	15.3334	2.3400
Sml	Old	2.3125	0.1563	14.80	<.0001	10.1000	1.5782

Since the data is zero-inflated, a ZIP model will now be considered. A ZIP model utilizes the fact that there is some probability that the data arise from a zero-generating process, and some probability it is generated from a Poisson distribution (which may also produce zeros). This is a simple mixture model. The code follows:

Model 8: ZIP Model

```
proc genmod data=dd1.poisson_data;
class store_type shelf_set;
model n_people_zp=store_type shelf_set store_type*shelf_set/ dist=zip link=log;
zeromodel store_type shelf_set / link=logit;
lsmeans store_type*shelf_set / ilink;
run;
```

The Pearson Chi-Square statistic will be used here to assess overdispersion, as the SAS documentation example for mixture models (Proc FMM) uses this statistic to do so,[6] and ZIP and ZINB are special cases of mixture models. This appears reasonable, since like the deviance, this value/df should be near 1. It also should be noted that the BIC and AICC are both superior to both non-ZIP models (Model 6 and Model 7).

Output 8: ZIP Model

The GENMOD Procedure

Model Information

Data Set	DD1.POISSON_DATA
Distribution	Zero Inflated Poisson
Link Function	Log
Dependent Variable	n_people_zp

Number of Observations Read	140
Number of Observations Used	140

Class Level Information

Class	Levels	Values
store_type	2	Big Sml
shelf_set	2	New Old

Parameter Information

Parameter	Effect	store_ type	shelf_set
Prm1	Intercept		
Prm2	store_type	Big	
Prm3	store_type	Sml	
Prm4	shelf_set		New
Prm5	shelf_set		Old
Prm6	store_type*shelf_set	Big	New
Prm7	store_type*shelf_set	Big	Old
Prm8	store_type*shelf_set	Sml	New
Prm9	store_type*shelf_set	Sml	Old

Zero Inflation Parameter Information

Parameter	Effect	store_ type	shelf_set
Prm10	Intercept		
Prm11	store_type	Big	
Prm12	store_type	Sml	
Prm13	shelf_set		New
Prm14	shelf_set		Old

The GENMOD Procedure

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance		829.1066	
Scaled Deviance		829.1066	
Pearson Chi-Square	133	145.6394	1.0950
Scaled Pearson X2	133	145.6394	1.0950
Log Likelihood		4989.5257	
Full Log Likelihood		-414.5533	
AIC (smaller is better)		843.1066	
AICC (smaller is better)		843.9551	
BIC (smaller is better)		863.6981	

Algorithm converged.

Analysis Of Maximum Likelihood Parameter Estimates

Parameter			DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept			1	2.5357	0.0574	2.4231	2.6483	1948.10	<.0001
store_type	Big		1	0.6181	0.0678	0.4852	0.7509	83.16	<.0001
store_type	Sml		0	0.0000	0.0000	0.0000	0.0000	.	.
shelf_set	New		1	0.3375	0.0740	0.1924	0.4825	20.80	<.0001
shelf_set	Old		0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Big	New	1	-0.2320	0.0887	-0.4059	-0.0582	6.84	0.0089
store_type*shelf_set	Big	Old	0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Sml	New	0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Sml	Old	0	0.0000	0.0000	0.0000	0.0000	.	.
Scale			0	1.0000	0.0000	1.0000	1.0000		

NOTE: The scale parameter was held fixed.

Analysis Of Maximum Likelihood Zero Inflation Parameter Estimates

Parameter		DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept		1	-1.4074	0.4017	-2.1947	-0.6200	12.27	0.0005
store_type	Big	1	-0.1259	0.4684	-1.0440	0.7921	0.07	0.7880
store_type	Sml	0	0.0000	0.0000	0.0000	0.0000	.	.
shelf_set	New	1	-0.4358	0.4713	-1.3594	0.4879	0.86	0.3551
shelf_set	Old	0	0.0000	0.0000	0.0000	0.0000	.	.

The GENMOD Procedure

store_type*shelf_set Least Squares Means

store_type	shelf_set	Estimate	Standard Error	z Value	Pr > |z|	Mean	Standard Error of Mean
Big	New	3.2592	0.03313	98.37	<.0001	26.0286	0.8624
Big	Old	3.1538	0.03597	87.68	<.0001	23.4242	0.8425
Sml	New	2.8731	0.04663	61.62	<.0001	17.6923	0.8249
Sml	Old	2.5357	0.05745	44.14	<.0001	12.6250	0.7253

Finally, a ZINB model will be generated using the following code:

Model 9: ZINB Model

```
proc genmod data=dd1.poisson_data;
class store_type shelf_set;
model n_people_zp=store_type shelf_set store_type*shelf_set/ dist=zinb link=log;
zeromodel store_type shelf_set / link=logit;
lsmeans store_type*shelf_set / ilink;
run;
```

The results here show little improvement over the ZIP model, which is consistent with the Pearson Chi-Square statistic indicating little overdispersion. Furthermore, it is seen that both AICC and BIC are similar to the ZIP model and the dispersion parameter is nearly zero, indicating that there is little overdispersion captured by this model that was not captured in the ZIP model. The parameter estimates are nearly identical, and the standard errors are relatively similar, which, again is expected since the ZIP model showed little evidence of overdispersion. This author would use the ZIP model in this instance, since it is a slightly simpler model and the ZINB does not represent much gain.

Comparing either model to the Negative Binomial regression model above, shows that the associated AICCs and BICs of these models are lower, indicating better fit. Also, interestingly, note the differences in the LS Mean estimates between these models and the Negative Binomial regression model. Generally, if the ZIP or ZINB model (or any model) represents the true structure of the data, the model will produce as-good or better estimates of these means.

Output 9: ZINB Model

The GENMOD Procedure

Model Information

Data Set	DD1.POISSON_DATA
Distribution	Zero Inflated Negative Binomial
Link Function	Log
Dependent Variable	n_people_zp

Number of Observations Read	140
Number of Observations Used	140

Class Level Information

Class	Levels	Values
store_type	2	Big Sml
shelf_set	2	New Old

Parameter Information

Parameter	Effect	store_ type	shelf_set
Prm1	Intercept		
Prm2	store_type	Big	
Prm3	store_type	Sml	
Prm4	shelf_set		New
Prm5	shelf_set		Old
Prm6	store_type*shelf_set	Big	New
Prm7	store_type*shelf_set	Big	Old
Prm8	store_type*shelf_set	Sml	New
Prm9	store_type*shelf_set	Sml	Old

Zero Inflation Parameter Information

Parameter	Effect	store_ type	shelf_set
Prm10	Intercept		
Prm11	store_type	Big	
Prm12	store_type	Sml	
Prm13	shelf_set		New

Prm14	shelf_set	Old

The GENMOD Procedure

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance		828.0976	
Scaled Deviance		828.0976	
Pearson Chi-Square	133	140.9873	1.0601
Scaled Pearson X2	133	140.9873	1.0601
Log Likelihood		-414.0488	
Full Log Likelihood		-414.0488	
AIC (smaller is better)		844.0976	
AICC (smaller is better)		845.1968	
BIC (smaller is better)		867.6307	

Algorithm converged.

Analysis Of Maximum Likelihood Parameter Estimates

Parameter			DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept			1	2.5357	0.0598	2.4184	2.6530	1795.34	<.0001
store_type	Big		1	0.6181	0.0713	0.4784	0.7578	75.22	<.0001
store_type	Sml		0	0.0000	0.0000	0.0000	0.0000	.	.
shelf_set	New		1	0.3375	0.0776	0.1855	0.4895	18.93	<.0001
shelf_set	Old		0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Big	New	1	-0.2320	0.0938	-0.4159	-0.0481	6.12	0.0134
store_type*shelf_set	Big	Old	0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Sml	New	0	0.0000	0.0000	0.0000	0.0000	.	.
store_type*shelf_set	Sml	Old	0	0.0000	0.0000	0.0000	0.0000	.	.
Dispersion			1	0.0067	0.0074	0.0008	0.0572		

NOTE: The negative binomial dispersion parameter was estimated by maximum likelihood.

Analysis Of Maximum Likelihood Zero Inflation Parameter Estimates

Parameter		DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept		1	-1.4074	0.4017	-2.1948	-0.6200	12.27	0.0005
store_type	Big	1	-0.1259	0.4684	-1.0440	0.7921	0.07	0.7881
store_type	Sml	0	0.0000	0.0000	0.0000	0.0000	.	.
shelf_set	New	1	-0.4358	0.4713	-1.3594	0.4879	0.86	0.3551
shelf_set	Old	0	0.0000	0.0000	0.0000	0.0000	.	.

The GENMOD Procedure

store_type*shelf_set Least Squares Means

Standard

store_ type	shelf_set	Estimate	Standard Error	z Value	Pr > |z|	Mean	Error of Mean
Big	New	3.2592	0.03592	90.74	<.0001	26.0286	0.9349
Big	Old	3.1538	0.03870	81.49	<.0001	23.4242	0.9066
Sml	New	2.8731	0.04933	58.25	<.0001	17.6923	0.8727
Sml	Old	2.5357	0.05984	42.37	<.0001	12.6249	0.7555

PROC FMM

The FMM procedure is designed to run finite mixture models. Proc FMM is experimental in SAS 9.3 and is deployed non-experimentally in SAS/STAT 12.1. As alluded to before, ZIP and ZINB are specific mixture models; specifically, they are a mixture of two distributions. The FMM procedure can therefore be used to run these models, as well as more complicated mixture models.

Consider the Model 9, repeated below for reference:

Model 9: ZINB Model

```
proc genmod data=dd1.poisson_data;
class store_type shelf_set;
model n_people_zp=store_type shelf_set store_type*shelf_set/ dist=zinb link=log;
zeromodel store_type shelf_set / link=logit;
lsmeans store_type*shelf_set / ilink;
run;
```

The FMM procedure can be used to execute the identical model using the following code:

Model 9-2: PROC FMM Implementation of Model 9

```
proc fmm data=dd1.poisson_data;
class store_type shelf_set;
model n_people_zp = store_type shelf_set store_type*shelf_set / dist=nb;
model + / dist=constant;
run;
```

Note here that there is two model statements when using the FMM procedure to run a ZIP or ZINB model. Since there is only one response variable in a mixture model, the second model statement is a continuation of the first. In the case where “constant” is a distribution, “constant” cannot depend on parameters, and hence “+” is used.

The output from this call to Proc FMM follows. It can be seen that the parameter estimates and their standard errors are identical to that which was generated from Proc Genmod above. There is no lsmeans statement in SAS/STAT 12.1. It also should be noted that the mixing probability here is 0.8429. The mixing probability here applies to the first model statement. The mixing probability of the second model statement, here is 1-0.8429=0.1571. The mixing probability of the second model in this mixture is not provided by Proc FMM. For more complicated mixture models, Proc FMM provides k-1 mixing probabilities if there are k distributions. This mixing probability is consistent with the fact that 15.71% of the dataset consists of zeros (this percentage can easily be discovered by using Proc Freq).

Output 9-2: PROC FMM Implementation of Model 9

```
                         The FMM Procedure

                         Model Information

          Data Set               DD1.POISSON_DATA
          Response Variable      n_people_zp
          Type of Model          Zero-inflated NegBinomial
```

Components	2
Estimation Method	Maximum Likelihood

Class Level Information

Class	Levels	Values
store_type	2	Big Sml
shelf_set	2	New Old

Number of Observations Read	140
Number of Observations Used	140

Optimization Information

Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	6
Mean Function Parameters	4
Scale Parameters	1
Mixing Prob Parameters	1
Lower Boundaries	1
Upper Boundaries	0
Number of Threads	2

Iteration History

Iteration	Evaluations	Objective Function	Change	Max Gradient
0	5	534.97034925	.	45.44539
1	11	477.71175632	57.25859294	108.6011
2	5	475.13951722	2.57223909	150.5811
3	9	458.88982731	16.24968991	272.3779
4	5	456.79007164	2.09975567	311.2533
5	5	455.88769371	0.90237794	385.3374
6	2	448.17507199	7.71262172	880.9545
7	8	435.0933355	13.08173649	158.2485
8	2	421.90230484	13.19103065	206.9195
9	3	421.19105792	0.71124692	146.7661
10	4	418.66724739	2.52381053	62.79942
11	3	417.17875752	1.48848988	99.98466

The FMM Procedure

Iteration History

Iteration	Evaluations	Objective Function	Change	Max Gradient
12	2	415.89549815	1.28325937	34.5853
13	3	415.20525472	0.69024342	47.22165
14	3	414.7698989	0.43535582	10.60998
15	3	414.53452542	0.23537347	21.74138
16	3	414.51882224	0.01570318	2.5495
17	3	414.51825341	0.00056883	0.055404
18	3	414.51825298	0.00000043	0.002017

Convergence criterion (GCONV=1E-8) satisfied.

Fit Statistics

-2 Log Likelihood	829.0
AIC (smaller is better)	841.0
AICC (smaller is better)	841.7
BIC (smaller is better)	858.7
Pearson Statistic	141.1
Effective Parameters	6
Effective Components	2

Parameter Estimates for 'Negative Binomial' Model

Component	Effect	store_ type	shelf_set	Estimate	Standard Error	z Value	Pr > |z|
1	Intercept			2.5357	0.05984	42.37	<.0001
1	store_type	Big		0.6181	0.07127	8.67	<.0001
1	store_type	Sml		0	.	.	.
1	shelf_set		New	0.3375	0.07755	4.35	<.0001
1	shelf_set		Old	0	.	.	.
1	store_type*shelf_set	Big	New	-0.2320	0.09382	-2.47	0.0134
1	store_type*shelf_set	Big	Old	0	.	.	.
1	store_type*shelf_set	Sml	New	0	.	.	.
1	store_type*shelf_set	Sml	Old	0	.	.	.
1	Scale Parameter			0.006738	0.007355		

The FMM Procedure

Parameter Estimates for Mixing Probabilities

	----------------Linked Scale---------------				
Effect	Estimate	Standard Error	z Value	Pr > |z|	Probability
Intercept	1.6796	0.2322	7.23	<.0001	0.8429

Another type of model that can be fit using Proc FMM is what is known as a Poisson Hurdle model. The Poisson Hurdle model also uses a mixture of two distributions, but unlike the ZIP model (Model 8 above), the Hurdle model uses a truncated Poisson distribution. That is, all zeros occur only from the zero generating process, and the Poisson process does not generate any zeros. The subtle difference is that the Hurdle model separates people into two groups, one that never purchases items, and one that always purchases items. The ZIP model separates people into two groups as well, people who may purchase items, but don't have to; and people that will never purchase items.

Consider Model 10 below.

Model 10: Poisson Hurdle Model

```
proc fmm data=dd1.poisson_data;
class store_type shelf_set;
model n_people_zp = store_type shelf_set store_type*shelf_set / dist=tpoisson;
model + / dist=constant;
run;
```

Like the other zero-inflated Negative Binomial model (Model 9-2), the mixing probability here is 0.8429. Likewise, the parameter estimates are the same. The fit of the Hurdle model, as assessed by AICC and BIC is nearly the same (ever so slightly better). However, the Pearson statistic is ever so slightly higher for Model 10. In this case, it can be learned from the data simulation that all zeros in the dataset were generated from the zero-inflation process and none originate from the Poisson process. Given this reality, the Hurdle model certainly makes sense. It also is intuitive that the two models may not give substantially different results in this situation where the underlying Poisson process generating the data has mean and variance such that it has low probability of generating zeros anyway.

Output 10: Poisson Hurdle Model

```
                         The FMM Procedure

                         Model Information

              Data Set                DD1.POISSON_DATA
              Response Variable       n_people_zp
              Type of Model           Poisson Hurdle
              Components              2
              Estimation Method       Maximum Likelihood


                      Class Level Information

                  Class          Levels    Values

                  store_type          2    Big Sml
                  shelf_set           2    New Old


              Number of Observations Read         140
              Number of Observations Used         140


                     Optimization Information

          Optimization Technique        Dual Quasi-Newton
          Parameters in Optimization    5
          Mean Function Parameters      4
          Scale Parameters              0
          Mixing Prob Parameters        1
          Number of Threads             2
```

Iteration History

Iteration	Evaluations	Objective Function	Change	Max Gradient
0	8	21527.738243	.	34881.97
1	4	3025.9489161	18501.789327	3090.047
2	5	2127.849507	898.09940906	1122.566
3	3	2099.5300838	28.31942328	1209.405
4	4	1725.8565766	373.67350718	1552.895
5	2	1035.7746375	690.08193904	1483.63
6	3	629.09942638	406.67521115	986.9397
7	3	456.04317121	173.05625518	343.025
8	3	421.21442441	34.82874680	136.3794
9	3	415.27362561	5.94079880	25.53575
10	3	415.02387562	0.24974999	1.575922
11	3	415.02283444	0.00104117	0.15867
12	3	415.02282835	0.00000610	0.008
13	3	415.02282833	0.00000002	0.000056

The FMM Procedure

Convergence criterion (GCONV=1E-8) satisfied.

Fit Statistics

-2 Log Likelihood	830.0
AIC (smaller is better)	840.0
AICC (smaller is better)	840.5
BIC (smaller is better)	854.8
Pearson Statistic	145.6
Effective Parameters	5
Effective Components	2

Parameter Estimates for 'Truncated Poisson' Model

Component	Effect	store_ type	shelf_set	Estimate	Standard Error	z Value	Pr > |z|
1	Intercept			2.5357	0.05745	44.14	<.0001
1	store_type	Big		0.6181	0.06778	9.12	<.0001
1	store_type	Sml		0	.	.	.
1	shelf_set		New	0.3375	0.07399	4.56	<.0001
1	shelf_set		Old	0	.	.	.
1	store_type*shelf_set	Big	New	-0.2320	0.08869	-2.62	0.0089
1	store_type*shelf_set	Big	Old	0	.	.	.
1	store_type*shelf_set	Sml	New	0	.	.	.
1	store_type*shelf_set	Sml	Old	0	.	.	.

Parameter Estimates for Mixing Probabilities

----------------Linked Scale---------------

Effect	Estimate	Standard Error	z Value	Pr > |z|	Probability
Intercept	1.6796	0.2322	7.23	<.0001	0.8429

For the last example, consider Model 1, the first introductory Poisson regression model introduced in this paper, as specified by:

Model 1: Simple Poisson Regression

```
proc genmod data=dd1.poisson_data;
class store_type shelf_set;
model n_people_poi=shelf_set / dist=poisson link=log;
lsmeans shelf_set / ilink;
run;
```

Recall that overdispersion arose from shelf set consisting of a mix of Poisson distributions, because there were independent variables not specified in the model. Suppose these were unknown. Proc FMM could be used to create a model out of multiple Poisson distributions here. The following code is used to determine how many Poisson distributions should be in the mixture.

Model 11(a): Determining How Many Poissons Should be Mixed

```
proc fmm data=dd1.poisson_data criterion=PEARSON;
       class shelf_set;
    model n_people_poi = shelf_set/ dist=poisson kmin=1 kmax=7;
run;
```

Here, the kmin and kmax options are asking Proc FMM to compute each possible mixture model ranging from 1 Poisson distribution (standard Poisson Regression) to a mixture of 7 Poisson distributions. Criterion=Pearson is asking Proc FMM to select based on the lowest Pearson statistic, however, it is typical to run this code and look at all of the various fit statistics to choose a model.

It can be seen below that the FMM procedure chose the model as a mixture of 6 Poisson distributions based on the Pearson statistic. All information following this note is based on that model. However, the goal of this step is to determine the model that makes sense to use overall, so it is desirable to look at multiple statistics, such as AICC and BIC to assess this. The Pearson statistic has little difference between any of the models, with the exception of the 1-component Poisson regression. Now is the time to recall the text box from Output 1, noting the Pearson Chi-Square statistic. It is identical to what is seen here for the 1-component model, and hence indicates overdispersion for this model. This is as it should be, since the 1-component model is the simple Poisson regression model, Model 1. The models with multiple components all have much smaller Pearson statistics and ratios of the Pearson statistic to their df much nearer to 1. For example, the two component model has 6 parameters and 140 observations, yielding 134 degrees of freedom; 139.49/134=1.04, which is quite near 1.

Here, we will now look at the other evaluation statistics, the AICC and BIC favor a 2, 3, or possibly a 4 component model. Overall, the 2-component model looks best, based on these statistics and a desire to keep models as simple as is appropriate. Reasonable arguments can be made for choosing a different number of components.

Output 11(a): Determining How Many Poissons Should be Mixed

The FMM Procedure

Model Information

Data Set	DD1.POISSON_DATA
Response Variable	n_people_poi
Type of Model	Homogeneous Regression Mixture
Distribution	Poisson

Min Components	1
Max Components	7
Link Function	Log
Estimation Method	Maximum Likelihood

Class Level Information

Class	Levels	Values
shelf_set	2	New Old

Number of Observations Read	140
Number of Observations Used	140

Component Description for Mixture Models

Model ID	Poisson
1	1
2	2
3	3
4	4
5	5
6	6
7	7

Component Evaluation for Mixture Models

	-------- Number of -------									
Model	-Components-		-Parameters-							Max
ID	Total	Eff.	Total	Eff.	-2 Log L	AIC	AICC	BIC	Pearson	Gradient
1	1	1	2	2	1017.64	1021.64	1021.73	1027.53	338.00	0.00047
2	2	2	5	5	931.55	941.55	942.00	956.26	139.49	0.00082
3	3	3	8	8	926.29	942.29	943.39	965.82	136.26	0.00178
4	4	4	11	11	924.96	946.96	949.02	979.32	134.21	0.00619
5	5	5	14	14	924.96	952.96	956.32	994.14	134.21	0.00029
6	6	6	17	17	924.96	958.96	963.97	1008.97	134.15	0.00947
7	7	7	20	20	924.96	964.96	972.02	1023.79	134.21	0.00547

The model with 6 components (ID=6) was selected as 'best' based on the Pearson statistic.

The following code can now be ran to model a 2-component mixture of Poisson distributions.

Model 11(b): A Mixture of Two Poisson Distributions

```
proc fmm data=dd1.poisson_data criterion=PEARSON;
       class shelf_set;
```

```
    model n_people_poi = shelf_set/ dist=poisson k=2;
run;
```

Output 11(b): A Mixture of Two Poisson Distributions

The FMM Procedure

Model Information

Data Set	DD1.POISSON_DATA
Response Variable	n_people_poi
Type of Model	Homogeneous Regression Mixture
Distribution	Poisson
Components	2
Link Function	Log
Estimation Method	Maximum Likelihood

Class Level Information

Class	Levels	Values
shelf_set	2	New Old

Number of Observations Read	140
Number of Observations Used	140

Optimization Information

Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	5
Mean Function Parameters	4
Scale Parameters	0
Mixing Prob Parameters	1
Number of Threads	2

Iteration History

Iteration	Evaluations	Objective Function	Change	Max Gradient
0	6	569.95232652	.	444.7992
1	2	475.94420023	94.00812629	140.8586
2	4	469.1528696	6.79133063	42.03412
3	3	468.82593058	0.32693902	35.38561
4	4	466.90559157	1.92033901	11.61105
5	2	466.63741801	0.26817356	5.68023
6	2	466.54054291	0.09687510	16.33568
7	4	465.99411755	0.54642537	8.964931
8	3	465.79122591	0.20289163	2.268786
9	3	465.77745064	0.01377528	0.53982
10	3	465.77718835	0.00026229	0.121473
11	3	465.77717871	0.00000965	0.007673
12	3	465.77717867	0.00000004	0.000105

```
                         The FMM Procedure

         Convergence criterion (GCONV=1E-8) satisfied.


                            Fit Statistics

              -2 Log Likelihood             931.6
              AIC  (smaller is better)      941.6
              AICC (smaller is better)      942.0
              BIC  (smaller is better)      956.3
              Pearson Statistic             139.5
              Effective Parameters              5
              Effective Components              2
```

Parameter Estimates for 'Poisson' Model

Component	Effect	shelf_set	Estimate	Standard Error	z Value	Pr > |z|
1	Intercept		3.2541	0.05234	62.18	<.0001
1	shelf_set	New	0.02272	0.06003	0.38	0.7051
1	shelf_set	Old	0	.	.	.
2	Intercept		2.5973	0.05728	45.34	<.0001
2	shelf_set	New	0.3002	0.08008	3.75	0.0002
2	shelf_set	Old	0	.	.	.

Parameter Estimates for Mixing Probabilities

Effect	Estimate	Standard Error	z Value	Pr > |z|	Probability
	Linked Scale				
Intercept	-0.1042	0.3188	-0.33	0.7437	0.4740

It can be seen that the AICC and BIC are improved versus the initial Poisson regression model, however, they are not as small as that of the correctly specified Poisson regression model.

The mixing probability here is 0.4740. This indicates that the data has modeled probability of about 0.47 of coming from one Poisson distribution, and about 0.53 of coming from the other Poisson distribution.

The downside to using proc fmm is that there is not an easy way to test mean differences in groups (such as the lsmeans statement). However, in situations where the goal is predictive versus a testing, creating such a model would have value, especially if some independent variables are unknown.

PROC COUNTREG

Proc Countreg can also run any of the models discussed here. It does not have an LS Means statement available in SAS/STAT 12.1, although LS Means could be calculated through the output.

The following code produces the same results as the Poisson regression model in section 1 above, if one considers the different choice of parameterization of variables that proc count reg makes.

Model 2-2: Poisson Regression Accounting for All Relevant Predictors, Rewritten Using Proc Countreg

```
proc countreg data=dd1.poisson_data;
   class store_type shelf_set;
   model n_people_poi=store_type shelf_set store_type*shelf_set / dist=poisson;
run;
```

Output 2-2: Poisson Regression Accounting for All Relevant Predictors, Rewritten Using Proc Countreg

The COUNTREG Procedure

Class Level Information

Class	Levels	Values
store_type	2	Big Sml
shelf_set	2	New Old

Model Fit Summary

Dependent Variable	n_people_poi
Number of Observations	140
Data Set	DD1.POISSON_DATA
Model	Poisson
Log Likelihood	-418.01557
Maximum Absolute Gradient	2.938E-11
Number of Iterations	3
Optimization Method	Newton-Raphson
AIC	854.03114
SBC	880.50593

Algorithm converged.

NOTE: The following parameters have been set to 0 (or feasible values), since the variables are a linear combination of other variables as shown.

```
store_type_Big =  store_type_shelf_set_Big_New + store_type_shelf_set_Big_Old
 shelf_set_New =  store_type_shelf_set_Big_New + store_type_shelf_set_Sml_New
```

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	2.515005	0.051917	48.44	<.0001
store_type Big	0	0	.	.	.
store_type Sml	0	0	.	.	.
shelf_set New	0	0	.	.	.
shelf_set Old	0	0	.	.	.
store_type*shelf_set Big New	1	0.747888	0.060435	12.38	<.0001
store_type*shelf_set Big Old	1	0.651525	0.061230	10.64	<.0001

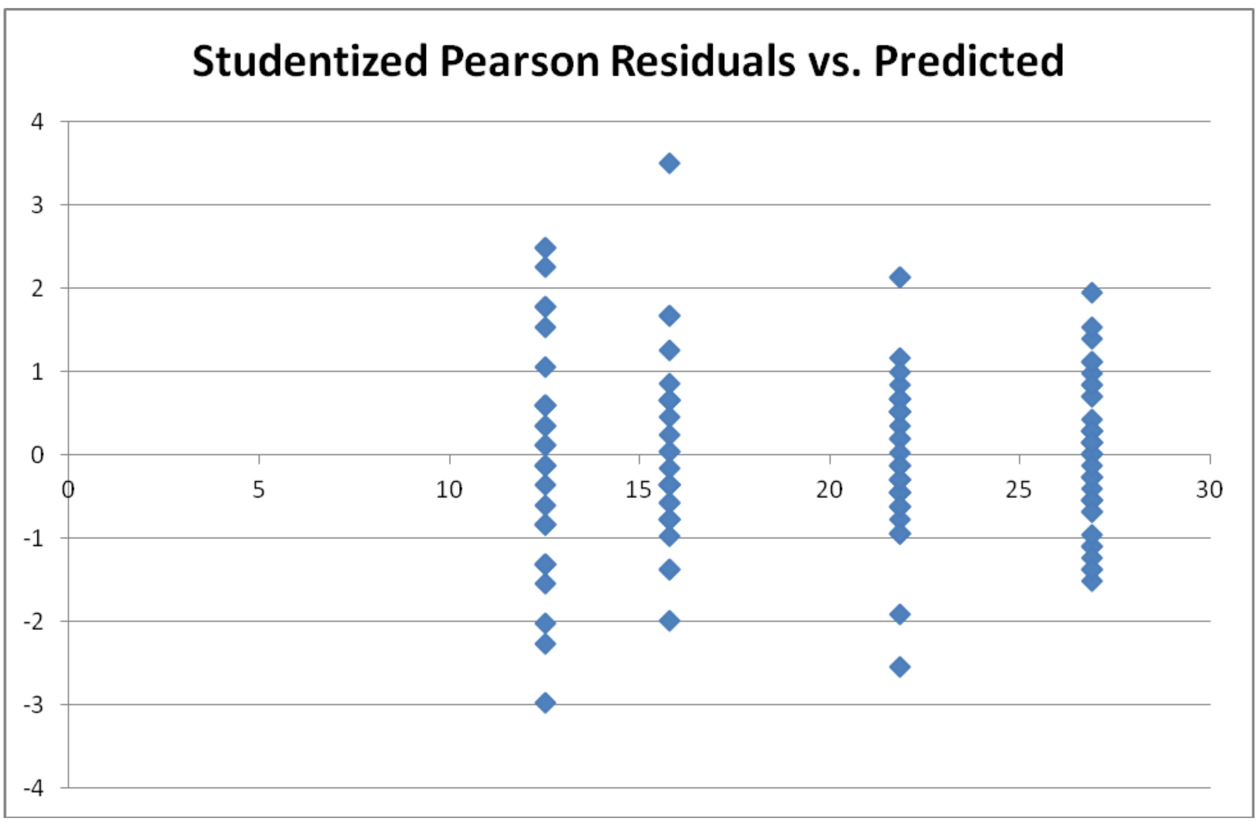
```
store_type*shelf_set Sml New     1        0.345290        0.067851       5.09       <.0001
store_type*shelf_set Sml Old     0               0               .          .           .
```

Overall, this author prefers to use proc genmod for these models. However, this is likely because this author learned these methodologies using proc genmod and is hence more comfortable.

A NOTE ON DIAGNOSTICS

In the course of this analysis, model fit and overdispersion have been the main focus of model diagnostics. In addition to these, plots of studentized residuals against the linear predictor are useful for assessing the model. These values can be output to a dataset using the output statement in Proc Genmod. Below they are also output to a .csv file in order to make the plot in Excel.

```
proc genmod data=dd1.poisson_data;
class store_type shelf_set;
model n_people_inf=store_type shelf_set store_type*shelf_set/ dist=nb link=log ;
lsmeans store_type*shelf_set / ilink;
output out=res STDRESCHI=resid pred=pred;
run;

ods csv body="C:\GJH\MWSUG\Milwaukee 2013\res.csv";
proc print data=res;
var pred resid;
run;
ods csv close;
```



In the case above, there may be some slight evidence of error variance heteroscadicity, but overall, this is a fairly reasonable plot; there are no severe outliers, and consistent enough error variance that this author would be comfortable. Generally speaking, these are good plots to make for each model.

DISCUSSION

Modeling is part art and part science. Ultimately, you need to decide what type of model best fits the data available. It is often the case that you have some underlying knowledge of the data generation process, and this can help drive model selection and guide choices. No model fit statistic can replace knowledge of the data, data generating process, and business goals.

Yet, model fit statistics can help guide the modeler to the right choice of model. Here, Poisson regression, Negative Binomial regression, ZIP, ZINB, and mixture models have been discussed. Any of them can be used to model count data. However, the data structure, available data and business objectives itself can guide which is the best choice to use.

For the three dependent variables considered here, "n_people_poi," "n_people_inf," and "n_people_zp", many difference models were presented.

For the first response variable, "n_people_poi", this author would use Model 2: Poisson Regression Accounting for All Relevant Predictors. This model, by accounting for the independent variables, resolves the overdispersion of Model 1, which arose due to heterogeneity. This is a simple model that provides a good answer.

For the response variable, "n_people_inf", this author would use Model 5: Negative Binomial Regression. This author generally prefers to model a second parameter to allow the variance to not equal the mean to eliminate overdispersion not easily solved via adding additional independent variables. In this case, it is seen the Negative Binomial regression of Model 5 reduces the overdispersion and also produces better AICC and BIC than the Poisson regression of Model 4.

Finally, for the response variable "n_people_zf", this author, would probably opt to use the ZIP model (Model 8). The ZINB model (Model 9, 9-2) and Hurdle (Model 10) models are quite similar to the ZIP model. In this case, the author would choose the ZIP model over the ZINB model, because there no demonstrable need to model the additional parameter in the ZINB model. Additionally, the author would choose the ZIP model over the Hurdle model in this instance, because from a business perspective it is more consistent with reality that some customers will never buy, and others will buy (but still have some probability of not buying). This particular specification will be easier to mesh with senior leadership's views on the customer and consequently will increase the probability that the work will be considered when strategic decisions are made. That said, reasonable people can disagree and any of these three models would be reasonable to use.

SPECIAL THANKS

Special thanks to Michael Wilson, who provided a number of great pieces of insight while reviewing this paper. This insight is greatly appreciated and materially improved this paper.

REFERENCES

[1] http://en.wikipedia.org/wiki/Zero-inflated_model

[2] Agresti, Alan, *Categorical Data Analysis*, Second Ed., Wiley-Interscience, 2002

[3] http://stats.stackexchange.com/questions/37732/when-someone-says-residual-deviance-df-should-1-for-a-poisson-model-how-appro?answertab=oldest#tab-top

[4] http://v8doc.sas.com/sashtml/stat/chap39/sect31.htm

[5] Proc Genmod help file

[6] Proc FMM help file

[7] Kessler, D. and McDowell, A., *Introducing the FMM Procedure for Finite Mixture Models*, SAS Global Forum 328-2012, 2012

[8] Russell, M. and Gray, B., *Markov Chains and Zeros in My Data: Bayesian Approaches in SAS® that Address Zero-Inflation in Count Data*, SAS Global Forum, 450-2013, 2013

CONTACT INFORMATION

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

George J. Hurley
The Hershey Company
19 E Chocolate Ave.
Hershey, PA 17033
717.534.5337
717.534.6991
ghurley@hersheys.com