**Paper SAS2703-2018**

# New Ways to Incorporate Continuous Regressors in Generalized Linear Models

Weixin (Angela) Wu and Anthony Salis, State Farm Insurance Companies;

Gordon Johnston, SAS Institute Inc.

## ABSTRACT

Generalized linear models (GLMs) are used in many fields because they accommodate responses with a variety of distributions and regressors that can be either continuous or categorical. In the insurance industry, GLMs are often used for ratemaking to estimate the relative risk between policyholders. Traditionally, we specify linear or polynomial effects for continuous regressors. However, alternative approaches are available for incorporating continuous regressors when the relationship between the regressor and the response is complex and unknown. One approach is to bin the regressor, apply a weight of evidence (WOE) transformation, and specify the binned regressor as a linear effect. Another approach is to specify spline effects for continuous regressors. This approach provides greater flexibility in relating the regressor to the response. Spline effects are supported by the GENSELECT procedure in SAS® Visual Statistics and the GAMMOD procedure in SAS® Visual Data Mining and Machine Learning, which fit GLMs and generalized additive models. This paper compares these approaches using zero-inflated insurance data and shows when each approach is appropriate.

## INTRODUCTION

Insurance companies build models using policyholder characteristics to predict future insurance losses. We use these models in the ratemaking process to charge policyholders an appropriate amount of premium based on their relative risk of an insurance loss. This paper will explore some considerations around using generalized linear models (GLMs) to model the effects of a couple of policyholder characteristics: age of the driver and tenure of the policy.

GLMs are among the more basic models that you can fit to your data. They are generally quick to compute and produce parameter estimates that are usually easy to interpret. In order to achieve this simplicity, the GLM makes some explicit assumptions about both the distribution of the target variable as well as the relationship between the model prediction and the regressors. This paper will focus on the relationship between the model prediction and the regressors and on the handling of regressors.

In a GLM the relationship between the model prediction $\hat{\mu}_i$ and the regressors is given as follows:

$$g(\hat{\mu}_i) = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_p x_{ip}$$

The assumption here is that the regressors have a linear relationship with a functionally transformed value of the model prediction. The transformation is known as a link function, and the transformed model prediction, $\hat{\eta}_i = g(\hat{\mu}_i)$, is known as the linear predictor. We will be considering the log-link function ($g(x) = \ln(x)$), which is common for insurance applications (Goldburd, et al., 2016).

While this assumption makes estimating the $\beta$ parameters very nice, many regressors in real-world applications don't exhibit such a nice relationship. Many variables require some preparation before you put them into a model in order to meet the GLM assumptions. In particular, we look at a couple of continuous variables that don't exhibit a linear relationship with the target. We will apply four different approaches to compensate for the non-linear relationships: 1) treat as a categorical variable, 2) apply a weight of evidence transformation, 3) model as a polynomial, and 4) model as a spline effect. We will fit each of these models using the GENSELECT procedure. The EFFECT statement in PROC

GENSELECT is a feature of this procedure that fits parametric splines to the regressors. We will also compare this to the results from PROC GAMMOD, which fits nonparametric splines to the regressors when it fits generalized additive models.

## DATA

The data used in this paper are simulated to look like insurance data. None of the data are real but they are able to illustrate the challenges faced when creating models for insurance.

The data represent a sample of 4.6 million automobile insurance policies. The variables on this data set are the regressors, the exposure, and the target. The regressors are characteristics of the policy taken at the time the policy is written. The two variables of interest for this paper are age of the principal driver (age) and tenure of the policy (tenure). There are nine other variables in the models, but these are categorical in nature and will not be discussed in this paper.

The exposure and the target come from following the policy over the subsequent year after it was written. The exposure is a measurement of the amount of time the policy was in force. Many of the policies have an exposure of 1 year but others cancelled their policies midway through the year and the exposure reflects the fraction of the year that policy was in force. The target is the claim frequency, which is the number of insurance claims the policyholder filed over the course of the year divided by the exposure.

These policies are randomly split into training and validation data sets, with 70% in the training and 30% in the validation data set. The models are fit on the training data set and evaluated on the validation data set.

Age and tenure treated in this paper as continuous variables. In a mathematical sense they are not truly continuous because they only take on a finite number of integer values. However they do exhibit trends where you expect there to be a smooth transition in the probability of experiencing an insurance claim as you move from one value to the next. Since they can both be described as categorical or continuous in nature, they make the perfect example for exploring some of these methods for handling continuous variables, as will be shown by the graphs and the methods chosen.

You can take a look at the univariate curves of the relative claim frequency by both age and tenure, which are provided in Figure 1 and Figure 2. These graphs illustrate the nonlinear relationship these variables have with the target frequency that will be addressed by the models.

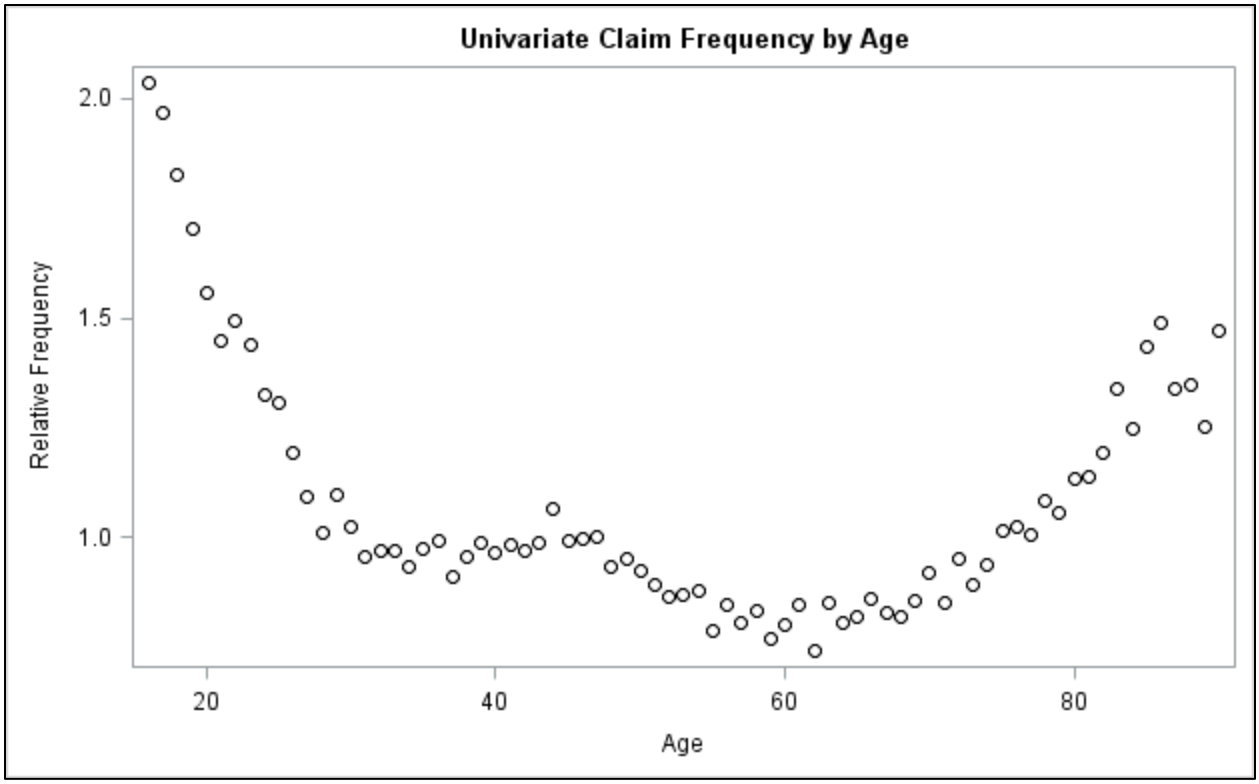


**Figure 1. Univariate Age Relativities**

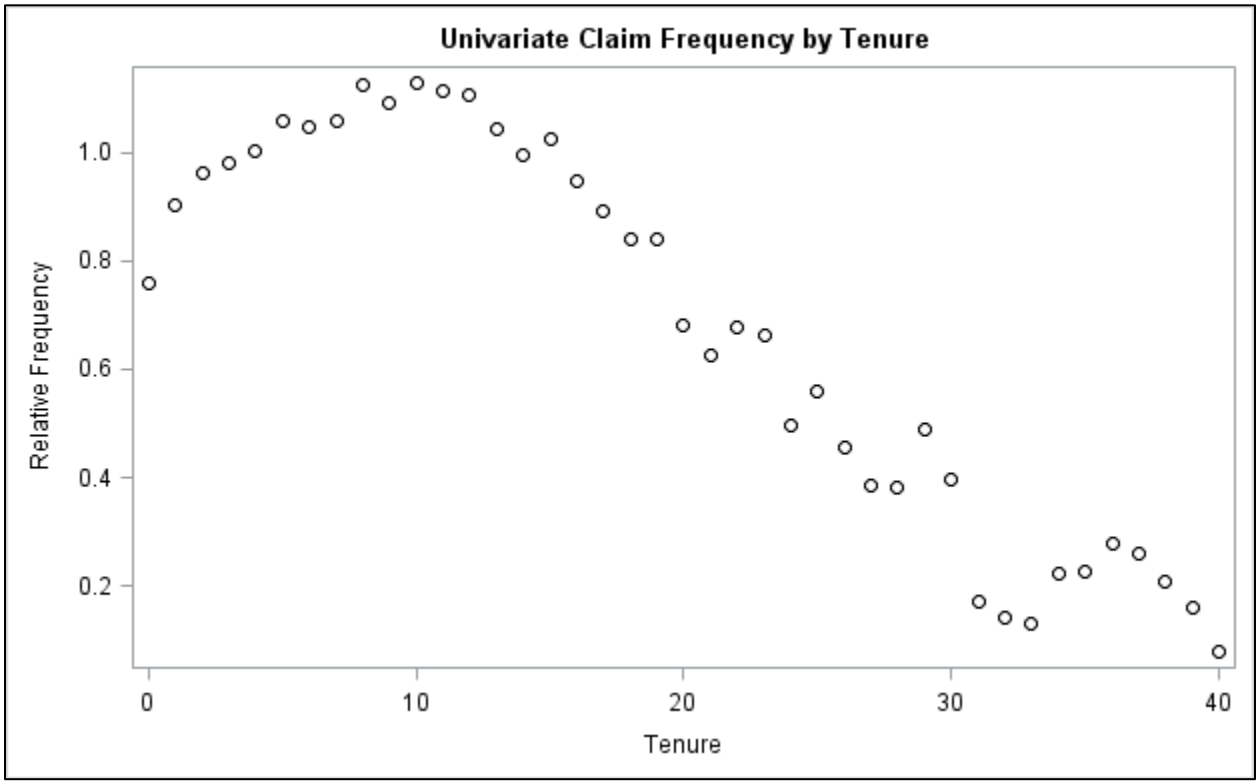


**Figure 2. Univariate Tenure Relativities**

## OVERVIEW OF METHODS

Each method involves fitting a GLM using PROC GENSELECT to predict the claim frequency. The GLMs are fit with a log link function and a Poisson error distribution. We used the CENTER option in the MODEL statement to aid in computing the maximum likelihood estimates. The observations are weighted by the exposure. Each model is fit with the same eleven variables. The nine categorical variables are listed in the CLASS statement and don't change from model to model. The only difference between each model is the treatment of age and tenure. Here are the different treatments for each model:

1. The age and tenure are added to the CLASS statement and treated as categorical regressors.
2. The WOE-transformed versions of age and tenure are used in the model as continuous regressors.
3. EFFECT statements are added to add higher-degree polynomial terms for age and tenure.
4. EFFECT statements are added to create a spline to describe age and tenure. Also PROC GAMMOD with spline terms in the model for age and tenure is used as a parallel model.

For the first two methods, the continuous regressors must first be binned into a finite number of categories. For any continuous variable, this means establishing a finite number of non-intersecting intervals that cover the complete domain of the variable. In the case of age and tenure, this binning was accomplished by rounding the values to whole numbers. This binning of the variable is used by all the methods because it is able to utilize the variable in both a discrete and continuous nature, depending on the method.

### METHOD 1: TREAT AS A CATEGORICAL VARIABLE

This first method treats these variables as categorical. This method provides the most flexible fit to the variables since it requires the most parameters. For a variable with $p$ distinct values, the model requires $p-1$ parameters to fit the variable (with the last category described by the intercept). The benefit to this great flexibility is that it doesn't require you to make any assumptions about the relationships between each category of the variable. The model is able to predict each category on its own to provide the most exact fit.

There are some drawbacks though. First, the high parameterization can lead to overfitting the model. There are no guards in this specification to ensure that there is a smooth transition in the predicted value when moving from one value to the next. Large jumps and reversals are likely to occur in the parameter estimates, especially if the data are thin and there is a large amount of variance in the target.

Second, this can be computationally impossible unless the variables are binned first. A truly continuous variable could have a different value for every observation, making it impossible to fit in a model. One solution is to bin the variables into a finite number of groups before modeling. Some information about the variables will be lost in the binning process, but that is true about any method that limits the number of parameters used to fit the variables. The key is to balance reducing the number of parameters to something that is reasonable with the amount of information lost by reducing the number of parameters.

The following is the code to run this model in PROC GENSELECT:

```
proc genselect data=myCaslib.modeldata;
  class Age Tenure &other_vars.; /* All variables treated as categorical */
  partition rolevar=selected(train='1' validate='0');
  model freq = Age Tenure &other_vars.
       / distribution=poisson link=log center;
  weight exposure;
  code file="&score_class."; /* Output code to score the model */
run;
```

## METHOD 2: USE WOE TRANSFORMATION

This second method applies a weight of evidence (WOE) transformation to the variables before including them in the model. This is similar to the first method in that the variable is binned so that there is a finite number of distinct values. The difference is that the variable is now fit in the MODEL statement as if it were continuous and there is only one parameter needed to fit the variable. The purpose of the WOE transformation is to transform the values of the variable so that the new WOE-transformed variable will have a linear relationship with the target. This method addresses the non-linear relationship the original variable has to the target before the modeling process instead of during the modeling process like the other methods do.

This method is perhaps more commonly explored in logistic regression where the values for each group of the WOE-transformed variable are equal to the log of the univariate odds ratio. In the case of this count regression, the values for each group are the log of a smoothed version of the relative univariate claims frequency. We used the log of these values to match the link function of the distribution. See Goldburg et al., 2016 for more information. Depending on the modeling context, there may be different methods to create WOE-transformed variables. The end effect of this transformation is that the WOE-transformed variable will be a logical representation of the differences between groups (Siddiqi 2006).

Similar to Method 1, this method has the drawback that there may be too many distinct values of the continuous variable to calculate the WOE values effectively without some preprocessing. At some level, you will need to bin the variable. Also, there is potential for reversals due to thin or noisy data. This can be mitigated by binning the variable into fewer levels or fitting a spline to the values and assigning the smoothed values to the variable. Because of the noise contained in the univariate claim frequency, we fit a spline to these target values before created the WOE transformation.

It is a little misleading to say that the variable only consumes one degree of freedom in the model. Since you use the information about the target variable to bin and set the WOE values, there is more information contained within these variables. Because of this, it is appropriate to use a separate data set to develop this transformation that is independent from the training and validation data sets. This will help prevent overfitting the variable in the model. For the results of this paper, we developed the WOE transformations with the training data set, but in practice we would use a separate data set to prepare the model variables.

A concern with this method is that it uses the variable's univariate relationship with the target to prepare the variable to enter into a multivariate model. To the extent that this variable is correlated with other variables in model, the true relationship of the variable to the target may be distorted when looking at the univariate relationship. One way to overcome this is to build an initial model with the correlated variables (or all of them) and treat the variable of interest as categorical. Then you can develop the WOE transformation from the relativities derived from the parameter estimates instead of the univariate summary of the target variable. This allows the WOE-transformed variable to account for the variable's relationship to the target in the multivariate environment. Generally speaking, best practices for GLMs suggest to build the model without any highly correlated variables, so you should take this into consideration when evaluating the inclusion and treatment of these correlated variables.

The following is the code to run this model in PROC GENSELECT:

```
proc genselect data=myCaslib.modeldata_woe;
  class &other_vars.; /* Age and tenure no longer categorical */
  partition rolevar=selected(train='1' validate='0');

  /* WOE-transformed variables modeled directly */
  model freq = Age_woe Tenure_woe &other_vars.
       / distribution=poisson link=log center;
  weight exposure;
  code file="&score_woe.";
run;
```

## METHOD 3: MODEL AS A POLYNOMIAL

The third method is adding polynomial terms to the variable. This is now the first method where you can specify your fully continuous variable in its fully continuous form. This is generally what is taught in any college regression course because it is just a more generalized form of what you are already specifying in the model. Before touching the variable with any binning or transformations, the most basic way to put it in the model is as a first degree polynomial, which is also how all the continuous variables in the model end up being specified.

Each degree in the polynomial contributes an extra parameter to the model, and it also increases the flexibility of the model fit. If the variable takes on $p$ distinct values, the polynomial can have a degree of $p-1$ and give the same prediction to each of those distinct values as if the variable were treated as categorical. But the advantage of the polynomial is that the degree can be decreased to something lower than $p-1$ and still be able to explain the vast majority of the predictive power the variable contains. At some point adding extra degrees to the polynomial has a diminishing return, so we are able to capture a great amount of the predictive power with fewer degrees of freedom.

In PROC GENSELECT, you can fit the polynomials easily using an EFFECT statement. You just need to specify the POLY option within the EFFECT statement and specify the variable and the degree of the polynomial. Certainly you can fit polynomials using other GLM procedures, but they require you to specify each order of the polynomial separately or create new variables on the data set before calling the modeling procedure. This is a very seamless integration of the polynomials and you can change the specification very easily. A sixth degree polynomial seemed to fit best for these variables, but this parameter is something that you will need to consider for each model.

The following is the code to run this model in PROC GENSELECT:

```
proc genselect data=myCaslib.modeldata;
  class &other_vars.;
  partition rolevar=selected(train='1' validate='0');

  /* EFFECT statement automatically creates polynomial terms */
  effect Age_poly6 = poly(Age / degree=6);
  effect Tenure_poly6 = poly(Tenure / degree=6);

  /* Use polynomial effects with names defined in the EFFECT statement */
  model freq = Age_poly6 Tenure_poly6 &other_vars.
       / distribution=poisson link=log center;
  weight exposure;
  code file="&score_poly.";
run;
```

## METHOD 4: MODEL AS A SPLINE EFFECT

The fourth and final method is to model the continuous variable with a spline in the EFFECT statement. You used the EFFECT statement to model the polynomials, and while that made fitting the polynomials much more seamless, that is a fairly common method in the statistical community to have specified manually. Fitting a spline to the variable in the model fit is much more involved and is not something that you would normally do on your own. PROC GENSELECT allows you to easily add this as a technique to your toolbox to help produce a better fit to the variable.

When you specify the SPLINE option in EFFECT statement, PROC GENSELECT adds columns to the design matrix to represent the basis splines for those variables. You can specify the basis, the number of degrees, and the number and placement of knots for maximum flexibility. These splines are then fit in the regression and the coefficients estimated detail a curve for the variable. We chose to use the default three degrees for the splines. We also specified the placement of the knots for both age and tenure. The

knots were placed where there were localized minima/maxima or where there were changes in inflection that the spline was not capturing without the knot.

PROC GAMMOD also has the ability to fit splines in the regression of a generalized additive model (GAM). While GAMs are not the focus of this paper, GLMs are a subset of GAMs and you can fit a very similar GLM using PROC GAMMOD as with PROC GENSELECT. The difference between the GLMs fit with each of these procedures is in the complexity of the spline variables. While you can specify the degrees and knots of the splines in PROC GENSELECT, the splines in PROC GAMMOD do not require that specification. Instead PROC GAMMOD uses penalized likelihood estimation to optimize the fit of thin-plate smoothing splines to the data. This optimization balances the goodness of the fit with the number of degrees of freedom consumed by the model. You can also specify a value for maximum degrees of freedom the spline can consume (default is 10), supply a list of knots for it to use, and specify a fixed smoothing parameter. We did not focus on these options for this paper but just used the default specification.

Splines are very flexible in their ability to fit a continuous variable in the model. They offer a different fit than the polynomial approach because you can include knots at points along the way to help adapt to the localized changes in the curve more dynamically. The WOE transformation can be very flexible to set the shape of the variable before modeling but it is not able to adapt to the multivariate nature of the model during the fitting process. Treating the variable as categorical is the most flexible way to treat a variable (if there are a finite number of unique values), but it consumes so many degrees of freedom that it is often impractical or overfits the data. The splines are able to capture much of the systematic variation without using nearly as many degrees of freedom.

A drawback to using splines is that they are not as readily interpretable as other methods. More so than the other methods, it is not as readily apparent from the parameter estimates for the splines what the variable is doing in the model. Similar to any other method using PROC GENSELECT, you can score some dummy observations with different values of the regressor and generate a graph of the relativities for each regressor. This is shown in the Results section.

PROC GAMMOD, however, does not output parameter estimates for the spline variables since these are created using a nonparametric approach. Because of this, it is not possible to score the effect of the splined variables on new data. To compensate for this, any data you want to have scored based on this model (like validation data) must be included in the input data set. If the target variable is set to be missing, PROC GAMMOD will not be able to use the observation to fit the model but it will produce a model prediction in the output data set.

The following is the code to run this model in PROC GENSELECT:

```
proc genselect data=myCaslib.modeldata;
  class &other_vars.;
  partition rolevar=selected(train='1' validate='0');

  /* EFFECT statement automatically creates spline terms */
  /* LIST knot option specifies variable values where knots are placed*/
  effect Age_spline = spline(Age / knotmethod=list(30 45 60 80));
  effect Tenure_spline = spline(Tenure / knotmethod=list(5 10 15 20 30));

  /* Use spline effects with names defined in the EFFECT statement */
  model freq = Age_spline Tenure_spline &other_vars.
       / distribution=poisson link=log center;
  weight exposure;
  code file="&score_spline.";
run;
```

The following is the code to run this model in PROC GAMMOD:

```
/* Include validation data so that it's scored but not used to fit model */
data myCaslib.modeldata_gam;
  set myCaslib.modeldata;
  if selected = 0 then freq = .;
run;

proc gammod data=myCaslib.modeldata_gam;
  class &other_vars.;

  /* Effects in the model are defined as spline functions or parameters */
  model freq =
    spline(Age / details)
    spline(Tenure /details)
    param(&other_vars.)
      / distribution=poisson link=log center;
  weight exposure;

  /* Output scored data set since there is no CODE statement */
  output out=myCaslib.gam_predictions predicted
    copyvar=(id freq inc_cnt exposure selected);
run;
```

## RESULTS

The model results presented here are focused on the way the model for each method fits both the age and tenure variables. These are visual representations of the relative predicted loss experience across different values of the variable for each method. When building any model you should certainly consider numerical fit statistics to aid your determination of the appropriate model. However, for the purposes of this paper, we will only look at the visual representations of the modeled variable relativities to build an understanding of how each method impacts the fit of the continuous variables.

In a GLM, a change in the value of a continuous variable has an additive effect to the fitted value of the linear predictor. Since these GLMs are built with a log-link function, a change in the value of the continuous variable then has a multiplicative change in the fitted value of the target. Modeled variable relativities are multiplicative factors that represent the magnitude of the fitted value of the target relative to a base level of the variable.

The modeled relativities are similar to the univariate relative claim frequencies shown in Figure 1 and Figure 2 in that they both represent a relative value of the target variable as you look across different values of the regressor. However, the modeled relativities consider the multivariate context of the model and show how the target variable is impacted by a change in one regressor while holding all other variables constant. Even if there is no random variation contained in the target variable, correlations among the regressors may cause the univariate relativities to be different from the (multivariate) modeled relativities.

You can use the parameter estimates output by PROC GENSELECT to calculate the model relativities, although some methods are more complex than others, which can make this process a little challenging. However, PROC GENSELECT also outputs scoring code that you can use to score the models on a sample data set to calculate the complete model predictions directly. This is an easier method that generalizes to all the models fit using PROC GENSELECT. The following code illustrates this concept:

```
data age_range;
  set myCaslib.modeldata(obs=1);
  do Age=16 to 90; /* Loop through values in the domain of age */
    %inc "&score_spline."; /* Score on the model */
    if Age=16 then base_pred=P_FREQ; /* Set base level for age */
    rel = P_FREQ/base_pred; /* Predicted value relative to base age */
    output;
  end;
run;
```

The relativities may be set as relative to any base value. They can also be centered so that the average relativity is 1, which is the case in the graphs shown in this paper.

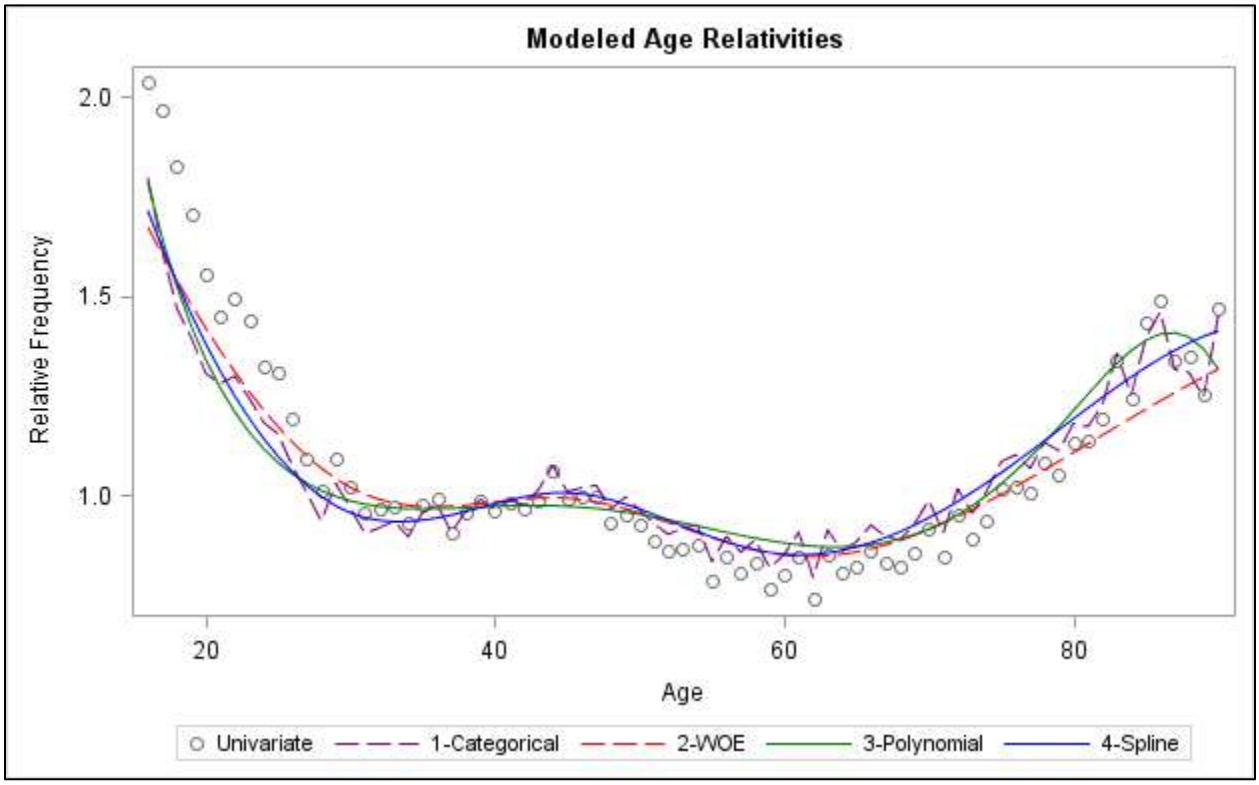


**Figure 3. Modeled Age Relativities for All the Methods**

Figure 3 overlays the modeled age relativities for all four GLMs on top of the univariate age relativities. You can see that there are a lot of similarities in the relativity curves for each method. These generally follow the univariate relativities except that they show lower relativities for ages less than 30 and slightly higher relativities for ages 50-75. This is due to the correlations that exist between age and other variables in the model. Because of this, there is not a great standard to show which model produces the "best" relativities. However, you can look at the differences between each curve to understand the differences between how each model fits the variable.

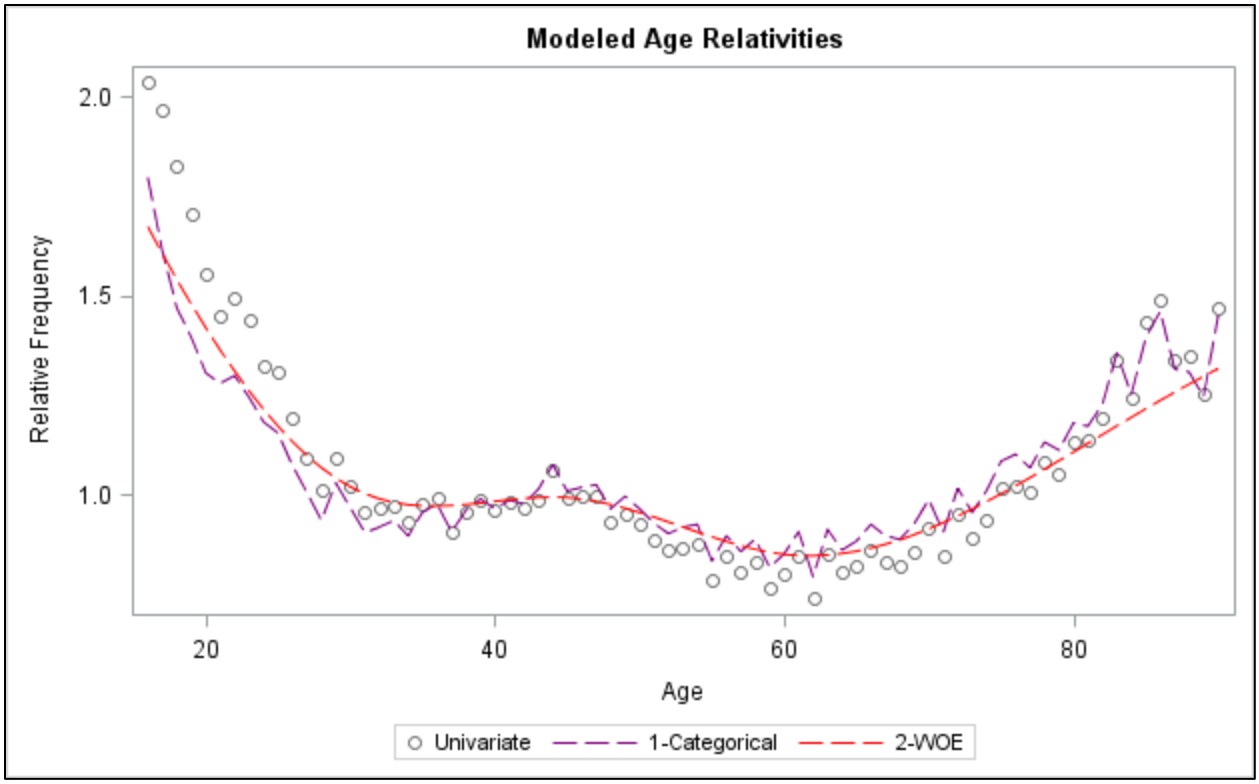


**Figure 4. Modeled Age Relativities for Methods 1 and 2**

Figure 4 shows the modeled age relativities for just methods 1 and 2 so that you can more easily see how each method fits the age variable. The purple line shows the relativities for the model with age treated as a categorical variable. This curve is very bumpy, just like the univariate relativities. This isn't surprising since this is the most heavily parameterized version of this variable. A limited amount of binning may make a big impact in avoiding undesired variability in the modeled relativities. However, since there are several fluctuations in this variable, it is not desirable to group too many values together since it will create a step function in the relativities leading to larger jumps in the predicted value as you move from one age to the next.

The red line shows the relativities for the model with age treated as a WOE-transformed variable. Since this transformation relies on the univariate relativities, the modeled relativities should have the same shape as the univariate relativities. Remember that the WOE-transformed values used to create this variable are based off of the smoothed univariate relativities, so the modeled relativities are also a smoothed curve. The parameter estimate can be interpreted as a power parameter to flatten or expand the variable relativities. If the parameter estimate is 1, then the modeled relativities are exactly equal to the univariate relativities. In this model, the parameter estimate is 0.717, which means the modeled relativities are flatter than the univariate relativities, as you can see in the graph. This difference is due to the correlation that exists between age and other regressors in the model. Other models account for this correlation differently in different parts of the distribution (16-30 and 50-75), whereas this model flattens the relativities for every part of the distribution in the same way. Basing the variable transformation off the univariate relativities can be a severe limitation of using WOE-transformed variables.

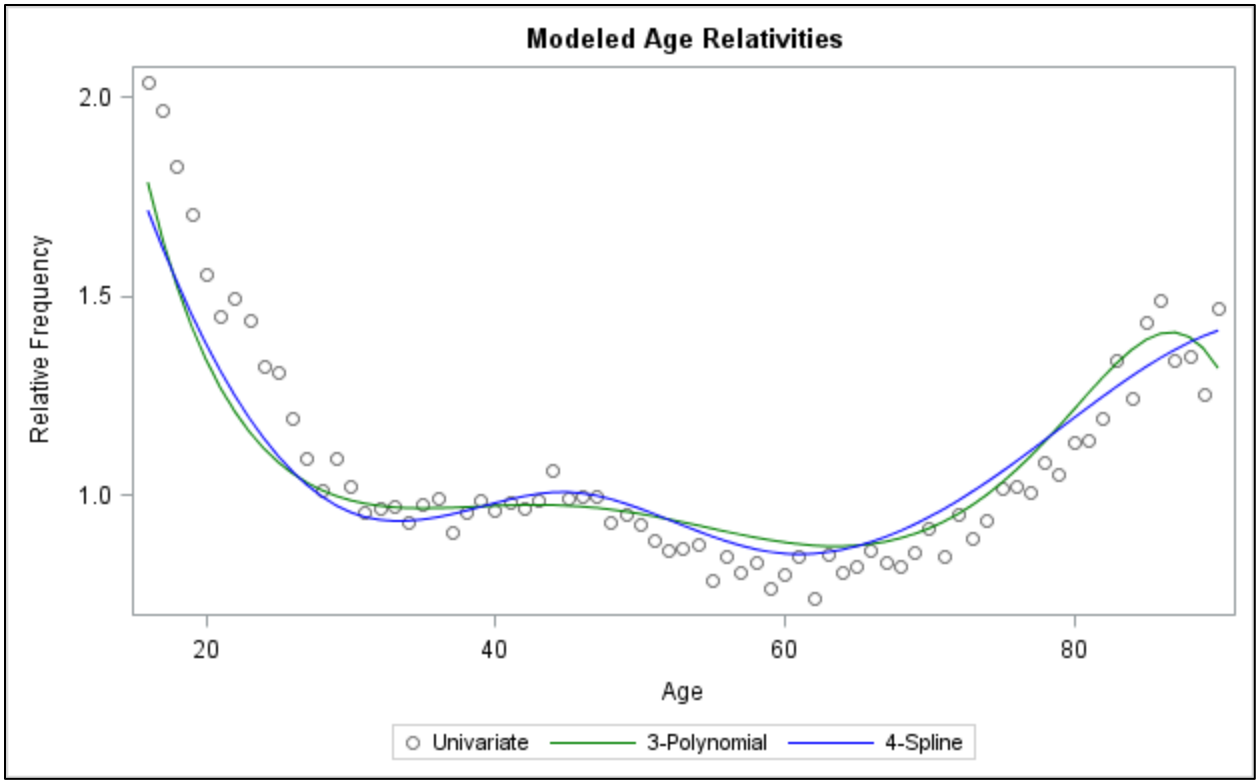


**Figure 5. Modeled Age Relativities for Methods 3 and 4**

Figure 5 shows the modeled age relativities for methods 3 and 4. The green solid line shows the relativities for the model with age treated as a polynomial and the blue solid line shows the relativities for the model with age treated as a spline. These methods both address the limitations of the first two in that they fit smooth curves to the variable in a multivariate context. In fact, they are very similar to one another in nature in that they use polynomial expansions of the variable to generate the curve. Their differences, though, are illustrated in this graph.

One difference is the degree of the polynomial. The polynomial method uses a single sixth degree polynomial, whereas the spline method strings together several third degree polynomials. The difference in the degree can create some differences in the shape of the relativity curve. Most noticeably, the polynomial curve turns downward at age 87. This is likely not a characteristic of the data but more likely a result of thin data and a larger variance in the claim frequency. Even though the relativities have a smooth transition, there can still be unexpected reversals, which may be an indication that the degree of the polynomial is too high.

At the other end of the domain, the polynomial curve has a much higher slope at age 16 than the spline curve. If you want to extrapolate the relativities, the higher degree polynomial curve may produce more extreme predictions for these points outside the domain. In the case of this model, there is a local maximum at age -8 before the relativity curve races to 0 (negative infinity for the linear predictor). Of course, it is generally unwise to use a model to extrapolate like this, so this likely is not a big concern.

The other difference between these models is the usage of knots, which are break points in the polynomial curve that allow the spline to be a piecewise, yet continuous, polynomial. While the polynomial achieves its flexibility through adding higher degrees, the spline can achieve more flexibility by adding knots. You may notice on the graph is that the polynomial model creates flatter relativities than the spline model in the middle of the distribution (ages 25-65). The spline is fit with knots at ages 30, 45, 60, and 80, which allows for more dynamic changes in the slope and convexity of the relativity curve. However, this greater localized flexibility of the spline model can also be one of its drawbacks. You can chose to specify where the knots are located (or have them automatically selected), but if the selection of knots is not reasonable, it can lead to the model being overfit. One suggestion is to use a separate data

set to find the desired specification for the knots and then fit the parameters on the training data set.

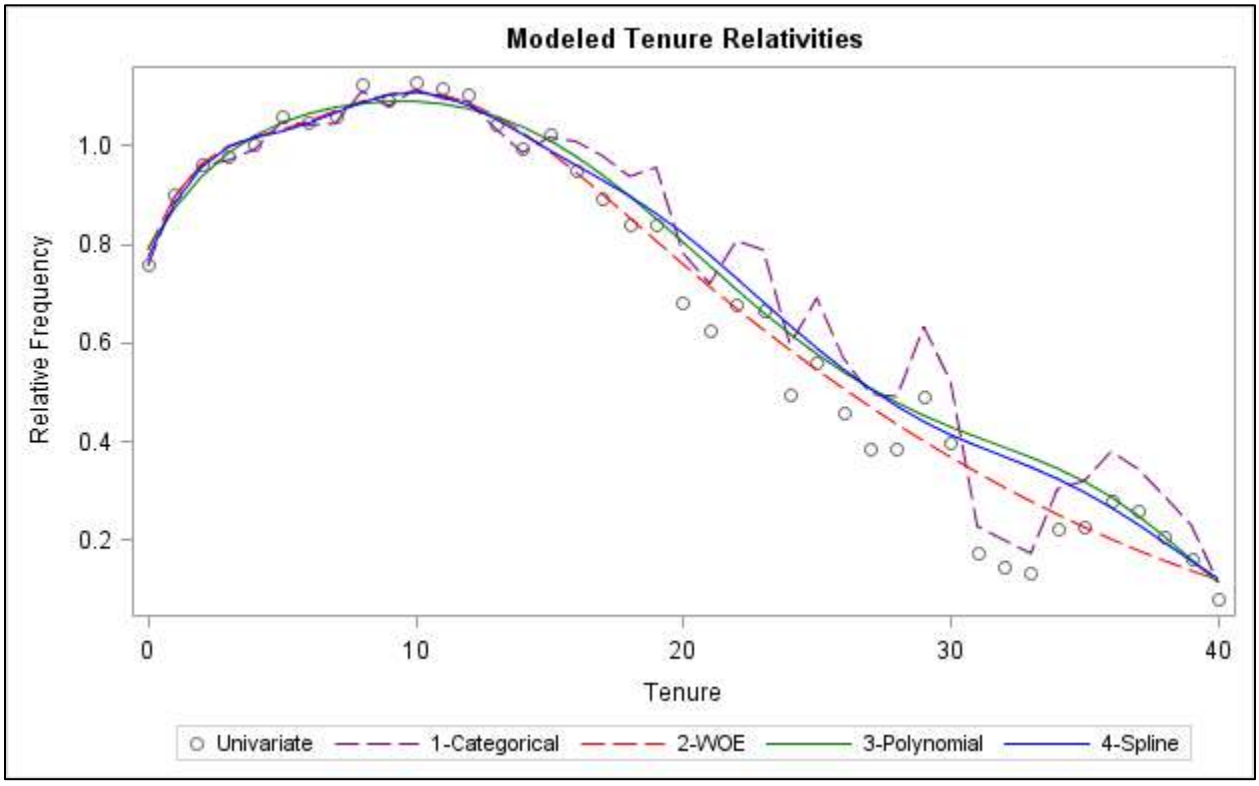


**Figure 6. Modeled Tenure Relativities for All the Methods**

Figure 6. Modeled Tenure shows the model relativities for the tenure variable in each of the models. There are some similarities to the way the relativities for this variable behave as they did for the age variable. The categorical model is very volatile, especially in the right tail where the data are very thin. The WOE-transformed model provides a very smooth curve. The parameter estimate for the WOE-transformed variable is 0.915, so there is considerably less flattening of the relativities from the univariate.

The polynomial model is again a sixth degree polynomial and it lies extremely close to the spline model. The interesting difference between these two curves is between tenures 4 and 12. Here the spline curve has a couple changes in convexity whereas the polynomial curve is concave down through this whole stretch. This difference is mainly due to the knots at tenures 5 and 10. This is an example of a time where you need to make a decision about whether the extra degrees of freedom the knots provide are making a positive contribution to fitting the variable or whether they are overfitting. The differences are slight, but validation fit statistics can provide information about which specification will generalize better to new data.

The WOE-transformed model is very similar to the polynomial and spline models before tenure 15, but after that is begins to diverge from the other two. This is especially interesting between tenures 30 and 40 where the WOE-transformed model is concave up but the other two continuous predictions are concave down. This is due to the correlations between the regressors that may indicate slightly different trends in the univariate and multivariate contexts. The thin data in this tail likely amplify these differences as well.

Another important item of note is how similar these models are, especially for tenures of 15 or less. In this data set, the bulk of the observations fall in the tenures of 15 and less and the data become much thinner very quickly after that. This illustrates that the method you choose may not make a significant impact on the model if you have sufficient data. It may be, for instance, that the categorical method is the easiest for you to implement in a model. If the data set is large enough, this may produce stable enough

results that you don't need to spend extra time fine-tuning the model with other methods.

Also, Figure 7 and Figure 8 show the relativities for the spline method built in PROC GENSELECT and PROC GAMMOD. We see very similar results from looking at both methods. PROC GAMMOD has a greater amount of flexibility in fitting the shape of the spline, which is shown in the slight differences in curvature throughout the distribution. The biggest differences occur in the right tail of the distribution for both variables. In sections of thin data, the nonparametric spline shows greater changes in slope and convexity, which may indicate that it is overfitting the data. However, in the other places where these models differ, the nonparametric spline may be able to use its increased flexibility to provide a better fit to the data. You can use the MAXDF option in the spline effect of the PROC GAMMOD model to raise or lower the amount of flexibility the spline has to fit the variable. Balancing this flexibility with the risk of fitting to the noise of the data is something you should consider when evaluating these different spline methods.

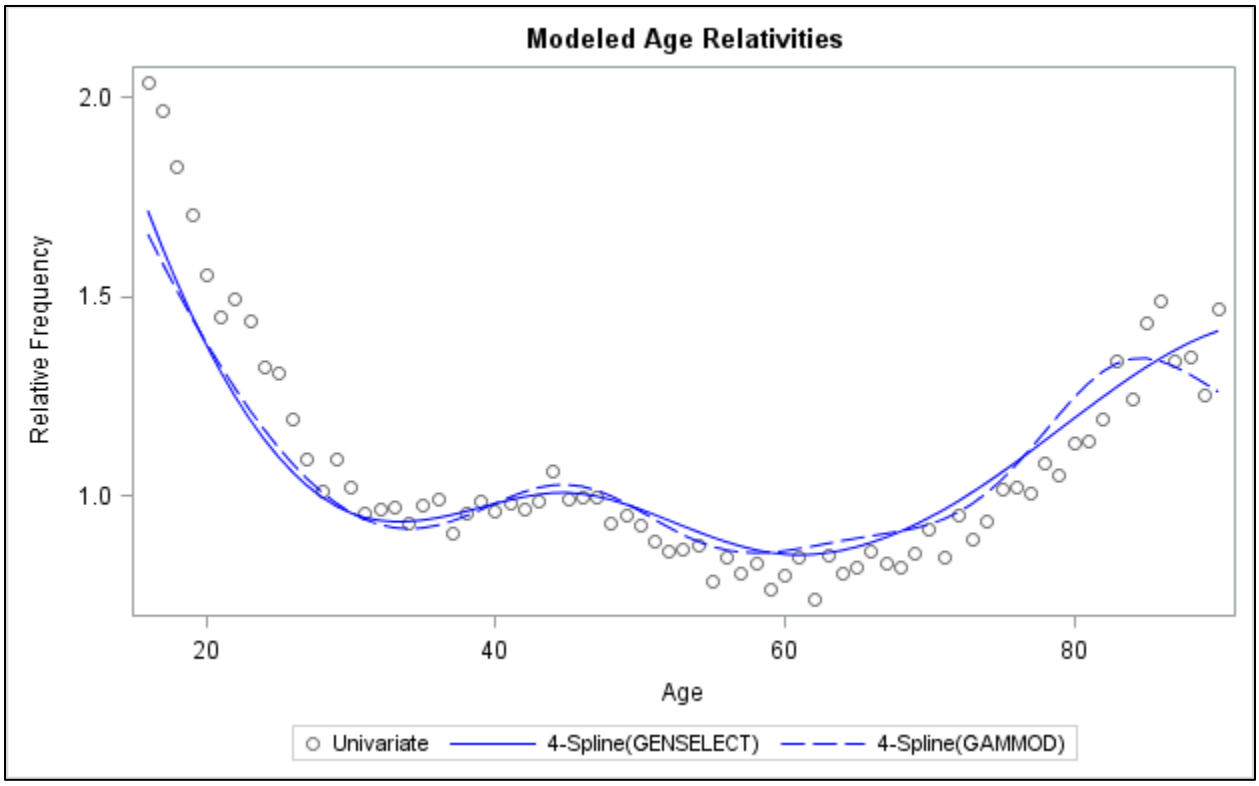


**Figure 7. Modeled Age Relativities for Spline Models**

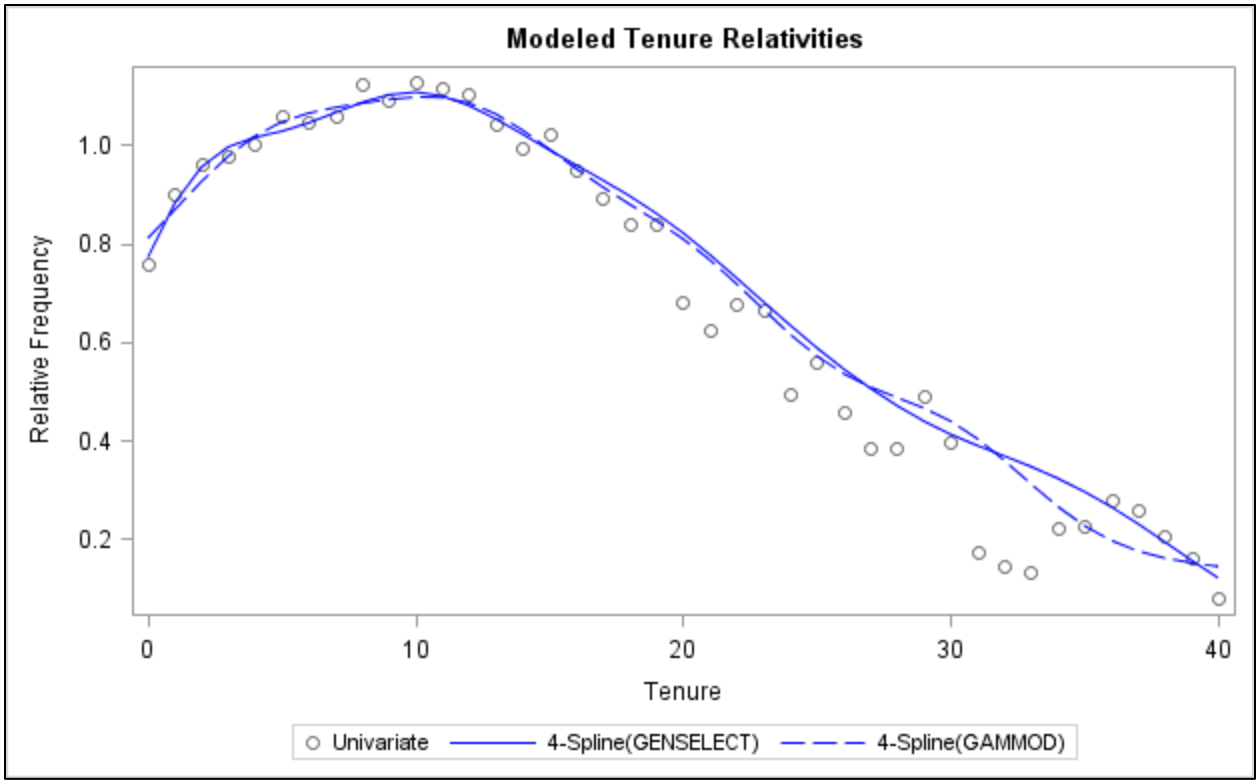


**Figure 8. Modeled Tenure Relativities for Spline Models**

## CONCLUSION

This paper presents four different methods of integrating continuous variables into a generalized linear model. The categorical method is the most flexible but it has the most potential to fitting to the noise rather than the signal. The polynomial method is a common method of maintaining the fully continuous nature of the variable but sometimes it can produce some undesired results. The WOE-transformation method can be extremely efficient when you want to fix how the variable will behave but that also requires time investigating what that curve should look like.

The spline method fits the variables in a very flexible yet controlled manner. While splines are able to capture the nuances of each continuous variable, you also have the control to specify where the fit needs to be more or less parameterized. This allows the splines to fit the signal of the data well while not overfitting to the noise. The splines in PROC GENSELECT give you a great deal of control over their fit, while the splines in PROC GAMMOD are self-tuning to select only the degrees of freedom necessary to fit the data.

Each method has its advantages and disadvantages. This paper highlights the spline methods as they are new features available in these SAS® procedures. Ultimately these are just different tools for you to use when building a regression model. Try them out and see how they can help improve your models.

## REFERENCES

Goldburd, Mark; Khare, Anand; Tevet, Dan. 2016. *Generalized Linear Models for Insurance Rating*. Casualty Actuarial Society.

Siddiqi, Naeem. 2016. *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. Hoboken, NJ: John Wiley & Sons.

## CONTACT INFORMATION

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

Weixin (Angela) Wu
State Farm Insurance Companies
angela.wu.uz28@statefarm.com

Anthony Salis
State Farm Insurance Companies
Anthony.Salis.r8ja@statefarm.com

Gordon Johnston
SAS Institute Inc.
Gordon.Johnston@sas.com