Paper SA-06-2012

Using SAS to Create Sales Expectations for Everyday and Seasonal Products

Ellebracht, Netherton, Gentry, Hallmark Cards, Inc., Kansas City, MO

Abstract

In order to provide a macro-level expectation for sales trends as they relate to external and market place factors, we will discuss the application of two different techniques currently practiced at Hallmark. The primary differentiation of approaches depends on the purpose of estimation. In the first we are creating a more traditional monthly time series sales forecast, as we do with a product such as birthday cards that are continually available. For this, we will discuss some benefits and features of PROC UCM. The second purpose of estimation is for seasonal product, such as Christmas cards. In this case, there is more emphasis on explanatory factors with many possible factors to select from. Since more attention is devoted to variable selection, we utilize PROC GLMSELECT. Now standard in version 9.2, GLMSELECT helps us easily determine the most reliable/stable estimates of categorical and continuous variables including calendar impacts as well as economic and/or external trends.

Introduction

In 2008, the economy started declining, consumer confidence was eroding, unemployment was rising, and by September the question on everyone's mind was how much the economic climate would influence our Christmas sales, and whether Hallmark should make any last minute order adjustments. In order to help answer the question with a data-driven response, we gathered 16 years of history for our Hallmark Gold Crown stores and constructed a fairly simple model to quickly provide some direction. This modeling process evolved and expanded to other major seasons (Valentine's Day, Mother's Day, Father's Day) and eventually to non-seasonal (Everyday) cards as well. While each of these models are all slightly different, their objectives are very similar: to provide a consistent process using historical sales data for all chains that helps better understand the macro environment for season and every-day sales.

The current models provide an independent retail unit expectation for Season and Everyday Cards based on historical unit trends and three to four other factors. These factors include average price offered, a calendar impact if appropriate (for instance the timing of Easter, the day of the week Valentine's Day falls, how early Mother's Day falls), and a demand factor that includes things such as: the weather at Valentine's Day, the trends in graduation rates, and internet usage. Some economic factors investigated include unemployment rate, personal income, and savings rate. The models are kept at a very high level and purposefully exclude internal marketing activities. This should provide a business-as-usual scenario that would assume an average level of marketing is maintained both historically and going forward. The models provide one input of many considered in an overall planning process at Hallmark. We'll present the two different techniques in modeling approaches and the procedures used in more detail in the following sections.

Everyday Models Background

In modeling Everyday Card unit sales we use a traditional time series forecast. The goal is to keep the model simple and include known influences such as price, economic climate, and possibly a calendar impact. Internal sales data such as unit performance and average price sold is maintained at a monthly level dating back to 2002. External factors such as unemployment rate, consumer expectations, and gas prices are also recorded. As with any time series data, time

points that are close to each other are more strongly correlated. Conversely, the further apart any two time points are, the less likely they are to be correlated. For example, sales data for the months of June and July are more strongly correlated than the months of June and November. Therefore, we need a forecasting method that can account for this local trend. Two commonly used procedures with such capabilities are autoregressive integrated moving-average (ARIMA) and unobserved component models (UCM). When forecasting with ARIMA, the model is fit to a de-trended, or stationary, series first. The stationary series is then forecasted, and the trend is added back in to get the series forecast. When forecasting with UCM, the model can be thought of as a regression model with trend as one of the predictors. For our purposes the UCM procedure provided a better model and is the method used to generate sales projection scenarios for the Everyday category.

Introduction to Proc UCM

Just as PROC REG models a response variable as a sum of well-chosen predictors, PROC UCM models a response series as a sum of well-chosen components. These components can be time varying. Some components commonly found in time series data are: trend, season, cycle. UCM treats each of these as a separate predictor and also lends itself to easily incorporating other predictor variables into the model.

- Trend- This can be thought of as the natural tendency of the response series excluding any other effects.
- Season- This can be thought of as an adjustment made to the level due to the seasonal effects. The length of the season has to be specified.
- Cycle-This can be thought of as a season but without having to specify a length.

The general syntax of UCM Procedure is as follows:

PROC UCM options;
  ID variables **INTERVAL=** *value* < *options* > **;**
  MODEL dependent variable <= regressors >;
  IRREGULAR < *options* > **;**
  LEVEL < *options* > **;**
    SLOPE < *options* > **;**
  SEASON **LENGTH=** *integer* < *options* > **;**
  CYCLE < *options* > **;**
  DEPLAG **LAGS=** *order* < *options* > **;**
  ESTIMATE **SKIPFIRST=** *integer* **SKIPLAST=** *integer* **OUTEST=** *dataset* < *options* >**;**
  FORECAST **SKIPFIRST=** *integer* **SKIPLAST=** *integer* **OUTFOR=** *dataset* **LEAD=** *integer* < *options* >**;**

ID specifies a variable that identifies the observations in the input data set. This is usually a SAS date, or time. The values of ID are extended beyond the data series for forecasting. This is based on the values specified in the INTERVAL which is required when using the ID statement.
MODEL specifies the response variable in the input data set. Predictors are not required here but may be included. Model fitting will not take place if the predictors contain missing values.
IRREGULAR corresponds to the overall random error. There can be at most one IRREGULAR statement in the model specification.
LEVEL and SLOPE specify the trend component. LEVEL can stand alone by itself, but SLOPE may not be used unless LEVEL is present. If only LEVEL is specified, the resulting trend is a

random walk. If LEVEL and SLOPE are both specified, the resulting trend is a locally linear trend.
SEASON specifies a seasonal component. There can be at most three SEASON statements but such specifications require a large amount of data and can be computationally exhausting. LENGTH is required when using the SEASON statement. Setting LENGTH=4 results in quarterly seasonality while setting LENGTH=12 gives monthly seasonality with monthly data.
CYCLE specifies a cycle component in the model. The cycle length is estimated from the data. There can be up to fifty CYCLE statements included in the model.
DEPLAG specifies the lag of the dependent variable to be included as a predictor in the model. LAGS is required when using the DEPLAGS statement. Setting LAGS=2 uses the two previous values in the dependent series to estimate the next value.
ESTIMATE specifies the data stream to be used to fit the model. Setting SKIPFIRST=12 will ignore the first 12 observations in the response series during parameter estimation. Setting SKIPLAST=10 will ignore the last 10 observations. OUTEST can be used to specify an output dataset for the parameter estimates.
FORECAST specifies the historical period used to compute the forecasts. SKIPFIRST and SKIPLAST have the same functions as in the estimate statement. Setting LEAD=18 will forecast 18 future values beyond the defined historical period. OUTFOR can be used to specify an output dataset for the forecasts.

Several models can emerge from the different combinations formed from the inclusion or exclusion of these components. Furthermore, IRREGULAR, LEVEL, SLOPE, SEASON, and CYCLE components can be refined using the options to be fixed instead of time varying. For example, trend consists of both LEVEL and SLOPE. Both can be time varying, or LEVEL could be time varying while SLOPE is fixed, and finally, both could be fixed. This last scenario results in a deterministic trend component. The paper "An Animated Guide©: Proc UCM (Unobserved Components Model)" by Russ Lavery suggests the following process for identifying when a component should be left as time-varying, fixed or removed altogether.

The first step is to determine if the component is time varying. This can be done by examining the output table "Final Estimates of the Free Parameters". If the component is found to have an insignificant Error Variance, then that suggests that the component is not time varying. However, it may not be removed from the model on this criterion alone. First we must check if the component is fixed. This can be done by using the following code in the options statement for said component: "variance=0 Noest;" and re-running the model. This time around we examine the "Significance Analysis of Components (Based on the Final State)" for the component in question. If the component is significant, it can stay in the model as deterministic; otherwise it should be removed altogether. This process is repeated for each component. If multiple components have insignificant Error Variances, then the least significant should be examined first.

Application of Proc UCM to Everday Models
We knew a priori that our ideal model would include a price and economic factor because they are known influences on unit demand. As previously mentioned, it is possible to incorporate such predictors into the model, however, the specifications of the trend, season, and cycle terms were identified first. In this initial stage we used the response series alone with no predictors. This revealed a model with a random walk trend, and seasonal effect of length 12. Since the data is at the monthly level, this corresponds to a one year period. An example is given below:

```
Proc UCM data=hallmark;
      id    sasdate interval=month;
      model ln_units;
      /*irregular;*/
        level;
      /*slope;    */
        season
            length=12  /*Specify the season length*/
            type=trig  /*Pick either trig or dummy*/
            ;
      /*cycle;    */

      estimate
            skiplast=12 /*does not observe the last 12 observations*/
            outest=est_hallmark;

      forecast
            skiplast=12 /*Does not observe the last 12 observations*/
            lead=18     /*The number of future periods to forecast */
            outfor=fore_hallmark;
run;
```

The skiplast option in the forecast statement is especially helpful for creating a holdout sample. The lead option will allow us to see what the model would have predicted for the last twelve months and next six months. The resulting output is shown below:

Post Sample Predictions for ln_units

Obs	sasdate	Actual	Forecast	Prediction Error	Sum of Squared Errors	Sum of Absolute Errors
96	Jul-10	15.74438	15.7939253	-0.0495451	0.0024547	0.049545
97	Aug-10	15.82731	15.8522362	-0.0249288	0.0030762	0.074474
98	Sep-10	16.06679	16.0471837	0.01960197	0.0034604	0.094076
99	Oct-10	15.81911	15.8041672	0.01493907	0.0036836	0.109015
100	Nov-10	15.68353	15.7036385	-0.0201123	0.0040881	0.129127
101	Dec-10	15.72609	15.7155958	0.01049203	0.0041982	0.139619
102	Jan-11	15.5978	15.6101035	-0.0123072	0.0043496	0.151926
103	Feb-11	15.67512	15.7067167	-0.0315992	0.0053481	0.183526
104	Mar-11	16.01392	15.9942148	0.01970445	0.0057364	0.20323
105	Apr-11	15.71159	15.7359892	-0.0243957	0.0063316	0.227626
106	May-11	15.82163	15.8180283	0.00359811	0.0063445	0.231224
107	Jun-11	15.99946	16.0275052	-0.0280426	0.0071309	0.259267
108	Jul-11	.	15.7788736	.	0.0071309	0.259267
109	Aug-11	.	15.8416704	.	0.0071309	0.259267
110	Sep-11	.	16.0529582	.	0.0071309	0.259267
111	Oct-11	.	15.7919149	.	0.0071309	0.259267
112	Nov-11	.	15.6891142	.	0.0071309	0.259267
113	Dec-11	.	15.6988871	.	0.0071309	0.259267

Being able to incorporate seasonal effects is beneficial for forecasting Everyday Cards since holidays occurring throughout the year impact Everyday Card sales. However, the Easter holiday poses a unique difficulty. Even though Easter is not the only holiday that shifts from year to year, it is the only one that shifts drastically enough to be in an entirely different month. Therefore having a seasonal component in the model may not always be enough. If the Easter switch was a more patterned behavior, we could possibly incorporate another seasonal

component. Since that is not the case, we have created our own variable called Easter impact that allows us to account for the calendar switch. We found this to be helpful reducing model error.

The other regressors of price and economic factor are then introduced into the model. There are several different variables that we considered for the price and economic factor predictors, resulting in several models. The best model is chosen by examining the fit statistics and holdout samples. Supplemental code located in the appendix demonstrates how macros can be used to produce the various combinations of candidate models. This is most helpful if the number of combinations is relatively small. If there are several factors to choose from, PROC GLMSELECT is recommended. This procedure is used to generate sales projection scenarios for the Season category

Season Models Background

The approach that we are taking with modeling season sales was different than the everyday approach, partially because of data difference and also because of difference in our ability to change seasonal orders. There were seemingly endless external economic and external metrics we wanted to consider in order to provide the best direction possible. With only 16 years of data, our ability to include multiple explanatory factors, including taking a more traditional time-series based approach, was limited. In addition, we felt the need to maintain a parsimonious model that had the most influential factors, acknowledging that did not imply these were the only influential factors. Using multiple iterations within the business and analytic teams, we landed on specifying a model with four components: price, economic factor, demand factor, and a calendar impact. Working within those boundaries there were still multiple variables to consider and select among. When we came across PROC GLM SELECT it made selecting variables, especially categorical factors, much easier. In the next section we discuss the benefits of PROC GLM SELECT and how it differs from traditional selection procedures. Then, we will briefly discuss how this has helped with our seasonal models.

Introduction to Proc GLMSELECT

Three common and widely taught variable selection methods are: forward selection, backward elimination and stepwise. Forward selection begins with a bare model that contains no predictors. At each subsequent step, a new predictor is added into the model and it remains if it meets a certain level of significance. Backward elimination begins with a model containing all possible predictors and removes the variable with the least significance one at a time until a model is reached wherein all predictors meet a specified level of significance. Finally, stepwise variable selection is a combination of forward and backward. It alternates adding/removing variables based on specified levels of significance until the final model includes only the predictors meeting the criteria for entry/removal.

As summarized by Frank Harrel in "Regression Modeling Strategies" (2001), these traditional variable selection techniques can be problematic for various reasons. In particular, the absolute value of the parameter estimates and the R-squared are both biased high. The first can potentially lead to incorrect conclusions about the importance of a particular predictor, and the second can overstate the amount of variance explained by the model.

Alternative selection methods have become increasingly popular in recent years. The LAR, Least Angle Regression, method is similar to forward selection in that it is an additive process. It begins by adding the variable most correlated with the current residual. The algorithm moves in that direction until *another* predictor is deemed to have a similar correlation with the current

residual. This is an iterative process until all predictors that have a correlation with the current residual are included. The LASSO, Least Absolute Shrinkage and Selection Operator, method is a type of penalized regression (shrinkage method) that requires the sum of the absolute values of the regression coefficients to be constrained by a specified parameter. Both of these methods have been shown to result in more parsimonious and accurate models.

Familiar procedures in SAS, such as PROC REG, have command options for variable selection. However, they are limited to forward selection, backward elimination or stepwise. Additionally, such procedures usually require the analyst to hard code any class variables as 0/1 indicators. In SAS version 9.2, PROC GLMSELECT is a procedure that not only allows for class variables to be a part of the selection process without having to recode the data, but it will also perform the LAR and LASSO methods. Overall the procedure is very flexible, allowing the analyst extensive customization.

As Robert Cohen explains in his paper "Introducing the GLMSELECT PROCEDURE for Model Selection", the primary features of the GLMSELECT procedure are:

- Model Specification
  - Supports classification effects, interactions and nested effects, hierarchy among effects and
  - Provides for internal partitioning of data into training, validation and testing
- Selection Control
  - Ability to perform multiple selection methods
  - Allows for selection from a very large number of effects and selection of individual levels of classification variables
  - Provides effect selection and stopping rules based on a variety of criteria
- Display and Output
  - Produces graphical representation of selection process and output data sets containing predicted values and residuals
  - Supports parallel processing of BY groups and multiple SCORE statements

The general syntax of GLMSELECT is as follows:

PROC GLMSELECT <options> ;
BY variables ;
CLASS variable <(v-options)> <variable <(v-options ...)> > </ v-options> <options> ;
FREQ variable ;
MODEL variable = <effects> </ options> ;
OUTPUT <OUT=SAS-data-set> <keyword <=name> > <...keyword=name> ;
PARTITION <options> ;
SCORE <DATA=SAS-data-set> <OUT=SAS-data-set> ;
WEIGHT variable ;

BY is used to specify a variable on which GLMSELECT will perform a separate analysis on observations in each distinct group. The data must be sorted on the BY variable in ascending order.
CLASS identifies the classification variables to be included in the analysis.
FREQ identifies a variable in the dataset containing the frequency of occurrence of each observation. GLMSELECT treats the observation as if it appears *n* times, where *n* is the value of the FREQ variable.

MODEL names the explanatory as well as independent variables. Options in this statement include variable selection method and criteria as well as displaying model statistics.
OUTPUT creates an output dataset that saves diagnostic measurements.
PARTITION specifies how data is split into subsets such as training, validation and testing.
SCORE creates a new SAS dataset containing predicted values and if requested, residuals as well. If DATA = data set, then the input data are scored. Multiple SCORE statements can be used for multiple data sets.
WEIGHT names a variable in the input data set with values that are to be used as relative weights for a weighted least squares fit.

Application of GLMSELECT to Season Models

When we started building seasonal models we were using PROC REG selection options, which required some upfront data preparation. With PROC GLMSELECT, we were able to use more robust selection techniques and could consider factors without needing to recode the categorical data. We also could easily test interactions with categorical and continuous variables. Additionally, it allowed us to include factors that we knew we would like in the model, such as price, while selecting among the remaining candidates. This procedure also has nice ODS output options so the final selected model factors can easily be stored in a SAS data set for later use in final model building and assessment. In the example below, we use the lasso selection option and use ODS to capture the selected summary model to an "ExSummary" data set. The model also highlights how we use the class option for categorical variables, as well as the ease in testing interaction impacts.

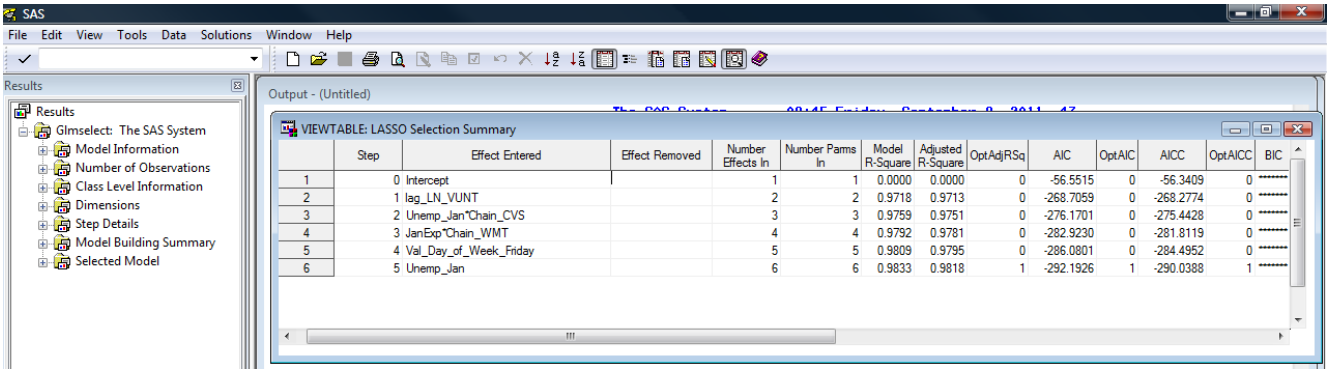
```
ods output  SelectionSummary=ExSummary;
proc glmselect data=example;
 class val_day_of_week chain;  /* these are two categorical variables */
  model ln_VUNT =

/** categroies: calendar impacts **/
val_day_of_week |chain@2  /* testing main effects and interaction between
                             these categorical variables */
val_dow_num|chain@2
unemp_jan|chain@2
janCCI|chain@2
feb_htgdegdays|chain@2
feb_snowfall|chain@2
ln_price

/** ... list more variables here ... **/

/     selection=lasso Maxstep=15 sls=.0001
            details=all stats=all SHOWPVALUES   ;
run;quit;
```

The output that SAS produces for the GLMSELECT procedure is well organized into easily understandable sections. Two key sections are the model building summary and selected model folders, which contain the best selected model, the parameters selected, and their estimates. The analyst can also output the results to another file as illustrated above with the ExSummary containing the selection summary table. The image below shows how the results are given in the folder structure on the left, and a sample of what is contained in the selection summary file created from the ODS output.



	Step	Effect Entered	Effect Removed	Number Effects In	Number Parms In	Model R-Square	Adjusted R-Square	OptAdjRSq	AIC	OptAIC	AICC	OptAICC	BIC
1	0	Intercept		1	1	0.0000	0.0000	0	-56.5515	0	-56.3409	0	*******
2	1	lag_LN_VUNT		2	2	0.9718	0.9713	0	-268.7059	0	-268.2774	0	*******
3	2	Unemp_Jan*Chain_CVS		3	3	0.9759	0.9751	0	-276.1701	0	-275.4428	0	*******
4	3	JanExp*Chain_WMT		4	4	0.9792	0.9781	0	-282.9230	0	-281.8119	0	*******
5	4	Val_Day_of_Week_Friday		5	5	0.9809	0.9795	0	-286.0801	0	-284.4952	0	*******
6	5	Unemp_Jan		6	6	0.9833	0.9818	1	-292.1926	1	-290.0388	1	*******

Having a variety of more robust selection procedures helps in building a more stable, and parsimonious model. After running an initial GLMSLECT, we return to the model and assess the factors, parameter estimates, and recent fit. Given our limited data, it was important that we focused on selecting the best external factors for explaining trends and changes in our sales. As we gain more data, we continue to evaluate the models and assess the factors and model structures to test if the relationships remain consistent or whether there are other factors worth considering.

Conclusion

Using Unobserved Component Models for forecasting in PROC UCM allows the use to easily incorporate seasons, cycles, and trends. Additionally, the user can easily define the historical period for estimation and holdout period for model comparisons by including a few straightforward commands. In the initial stages of model building, the numerous factors available for selection can be overwhelming. Here PROC GLMSELECT is used for building a model for forecasting but it can be utilized in several modeling exercises. Employing PROC GLMSELCT is an efficient way to detect the most influential variables. The options outlined for these procedures within this discussion are in no way exhaustive. Only the options found most helpful are described. Please refer to the SAS website for the full list of capabilities within these procedures.

References

Flom, P., Cassell, D. (2009) “Stopping stepwise: Why stepwise and similar selection methods are bad, and what you should use”

Yuan, M., Joseph, V., Lin, Y. (2005) “An Efficient Variable Selection Approach for Analyzing Designed Experiements”

Ziegler, M. (2006) “Variable Selection When Confronted With Missing Data”

Leng, C., Lin, Y., Wahba, G. (2004) “A Note on the Lasso and Related Procedures in Model Selection”

Cohen, R. “Introducing the GLMSELECT PROCEDURE for Model Selection”, Paper 207-31

Harrell, F. E. (2001), *Regression Modeling Strategies: With application to linear models, logistic regression, and survival analysis,* Springer-Verlag, New York.

Lavery, R. “An Animated Guide©: Proc UCM (Unobserved Components Model)”

Acknowledgements

Thanks to Mary Ryan for her support and guidance. Thanks to our fellow Hallmarkers for their encouragement to continually experiment and learn.

Contact Information

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

Lory Ellebracht
Hallmark Cards Inc.
Kansas City, Mo. 64141
Phone: 816-545-2803
Email: Lory.Ellebracht@gmail.com

Casey Gentry
Hallmark Cards Inc.
Kansas City, Mo. 64141
Phone: 816-545-0705
Email: cgentr2@hallmark.com

Jamie Netherton
Hallmark Cards Inc.
Kansas City, Mo. 64141
Phone: 816-545-2991
Email: jnethe3@hallmark.com

Appendix

```
%macro example(chain, target, price, economic, calendar, num);

Proc UCM data=&chain._external_&climate;
      id    sasdate interval=month;
      model &target=    &price &economic &calendar;
      /*irregular;*/
        level;
      /*slope;    */
        season
            length=12  /*specify the season length*/
            type=trig  /*pick either trig or dummy*/
            ;
      /*cycle;    */

      estimate
            skiplast=12 /*does not observe the last 12 observations*/
            outest=est_&chain._&num;

      forecast
            skiplast=12 /*does not observe the last 12 observations*/
            lead=18     /*# of future periods to forecast
            outfor=fore_&chain._&num;
run;
ods graphics off;
ods html close;
data  forecast_&chain._&num;
      set fore_&chain._&num;
      keep  sasdate &target &price &calendar &economic forecast residual
            a_unit_lag12 f_unit_lag12  a_unit f_unit U_pChg F_pChg;
      a_unit=exp(&target);
      f_unit=exp(forecast);
      a_unit_lag12=lag12(a_unit);
      f_unit_lag12=lag12(f_unit);
      U_pChg=(a_unit/a_unit_lag12)-1;
      F_pChg=(f_unit/f_unit_lag12)-1;
      run;
      quit;
PROC EXPORT DATA= WORK.est_&chain._&num
            OUTFILE= "C:\SAS\Working Files\ED Forecasting\ParameterEstimates"
            DBMS=EXCEL REPLACE;
     SHEET="&price &economic &num";
RUN;
PROC EXPORT DATA= WORK.forecast_&chain._&num
            OUTFILE= "C:\SAS\Working Files\ED Forecasting\Forecast"
            DBMS=EXCEL REPLACE;
     SHEET="&price &economic &num";
RUN;

%mend example;
%example (HALLMARK, ln_units ,              ,          ,               , 0 );
%example (HALLMARK, ln_units , ln_apshipped , ln_unemp , easter_impact , 1 );
%example (HALLMARK, ln_units , ln_apshipped , ln_conex , easter_impact , 2 );
%example (HALLMARK, ln_units , ln_apshipped , ln_gas   , easter_impact , 3 );
%example (HALLMARK, ln_units , ln_apsold    , ln_unemp , easter_impact , 4 );
%example (HALLMARK, ln_units , ln_apsold    , ln_conex , easter_impact , 5 );
%example (HALLMARK, ln_units , ln_apsold    , ln_gas   , easter_impact , 6 );
```