Article from

Forecasting and Futurism

Month Year July 2015
Issue Number 11

How to Build a Model

By Richard Xu, Dihui Lai, Minyu Cao, Scott Rushing, and Tim Rozar

This article first appeared as appendix b of the Lapse Modeling for the Post-Level Period: A Practical Application of Predictive Modeling *report, sponsored by the SOA's Committee on Finance Research. It is reprinted here with permission.*

Building an effective and robust model requires a solid foundation in statistics and practical experience in statistical applications. For those wanting to increase their modeling skills, we recommend further study of statistical algorithms (such as GLM and decision trees) and additional development of applicable technical skills.

This Appendix serves as an introduction to a few basic modeling techniques. For a more complete and comprehensive understanding of statistical modeling, a formal study program would be beneficial.

The software and programming language used for this example is called R and is accessible to the public as an open-source application. There are no license restrictions. The system is expandable by design and offers very advanced graphic capabilities. As of June 2014, there are more than 5,800 add-on packages and more than 120,000 functions available under the R framework. R is developed based on a modern statistical language, which is very close to C/C++. A large online community is available to support learning, in addition to the built-in help system.

However, the learning curve for learning the R language and software environment can be quite steep. Additionally, there are limitations in using R such as the demands on memory, single thread in CPU utilization, limited graphic user interface, limited GUI, etc. Some of these problems can be addressed by the many add-on packages.

The example that follows is based on a hypothetical dataset and is intended for educational purposes. The data file is attached to this document and can be downloaded from SOA website where the main document is located. A few simple steps are provided to demonstrate a simplified approach to building a model in R.

Note: The commands that need to be entered into R are displayed in ***bold italics***, while the return from the R software is in `this Courier font.` Please note that R is a command-line system. To perform functions, a user is required to type in every command.

DATA LOADING

In the following R script, we assume the sample data file is called "SampleData2014SOAPM.csv", which is a comma delimited text file. To load the data into the R system, the following command should be executed, assuming the file is located in "C:/Data":

> lapseData <- read.csv("C:\\data\\SampleData-2014SOAPM.csv", header=TRUE)

The option of "header=TRUE" indicates that the names of the data fields are included in the data file. Since this is also the default setting, it can be ignored.

After reading the data, the R system assigns the whole dataset to an object called "lapseData". This object has the data structure called "data frame". The data frame structure is equivalent to a worksheet in an Excel file, with rows (record index) and columns (data fields) available for data manipulation.

R has other options to import data including from an Excel file, a database, the internet, or manually importing it into R by hard-coded R scripts.

DATA EXPLORATION

Once loaded, there are numerous ways to examine the data. Below are the two most common procedures to understanding the volume and characteristics of the data.

The 'summary' command returns the distribution of each field provided in the data.

>summary(lapseData)

```
 FaceAmount  PremiumMode RiskClass IssueAge  LapsedN          Exposure
 100-250K:28  Annual :70   NS:70   25-29:20  Min.   :    1.00 Min.   :    29.61
 250K-1M :28  monthly:70   SM:70   30-34:20  1st Qu.:   47.75 1st Qu.:   844.64
 50-100K :28                       35-39:20  Median :  417.00 Median :  7159.10
 GT1M    :28                       40-44:20  Mean   : 1735.03 Mean   : 24594.77
 LT50K   :28                       45-49:20  3rd Qu.: 1775.75 3rd Qu.: 24881.78
                                   50-54:20  Max.   :14712.00 Max.   :186853.50
                                   55-59:20
```

The 'head' command returns the first 6 records in "lapseData".

> head(lapseData)

```
  FaceAmount PremiumMode   RiskClass   IssueAge  LapsedN    Exposure
1   100-250K     monthly          NS     25-29    1220       44507.43
2   100-250K     monthly          NS     30-34    2023       65939.43
3   100-250K     monthly          NS     35-39    2963       74532.25
4   100-250K     monthly          NS     40-44    3779       75532.78
5   100-250K     monthly          NS     45-49    4143       67085.31
6   100-250K     monthly          NS     50-54    4267       59205.88
```

Other commands for data exploration include dim(), names(), tail(), aggregate() and many more.

MODEL CREATION

After the basic understanding of the data is obtained, one can start building a model. In the dataset, our target variable is the number of lapses per number of policies exposed per unit of time (in this case, one year).

In this sample model, the Poisson distribution is used and logarithm is the default link function.

The number of lapses is called 'LapsedN' in our model and 'Exposure' reflects the number of policies exposed for the corresponding duration. To reflect this in the model and since the link function is the logarithm, the offset is the logarithm of 'Exposure'.

$$\log(\text{LapsedN} / \text{Exposure}) = \log(\text{LapsedN}) - \log(\text{Exposure})$$

As we can see from the preceding equation, subtracting "log(Exposure)" on the right side of the equation as an offset is equivalent to dividing by 'Exposure' on the left side of the equation, which changes the lapse count to the lapse rate which is what is being modeled here.

> Model1 <- glm(LapsedN ~ offset(log(Exposure)) + FaceAmount + PremiumMode + RiskClass + IssueAge, family=poisson(), data=lapseData)

In the above command, "glm" is the specified model family, and 'family=poisson()' is the specified distribution. Since the default link function of logarithm is what's needed, it is not necessary to specify in the bracket. The target variable is 'LapsedN', and there are 4 explanatory variables: 'FaceAmount', 'PremiumMode', 'RiskClass', and 'IssueAge'.

After the model is fit with the data, the model results can be checked with the following command:

CONTINUED ON **PAGE 32**

HOW TO BUILD A MODEL | FROM PAGE 31

> summary(Model1)

```
Call:
glm(formula = LapsedN ~ offset(log(Exposure)) + FaceAmount + PremiumMode
             + RiskClass + IssueAge, family = poisson(), data = lapseData)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-14.4278   -1.7662   -0.1371    1.6875   14.4382

Coefficients:
                          Estimate     Std. Error    z value       Pr(>|z|)
(Intercept)              -2.872380    0.009227      -311.31       < 2e-16 ***
FaceAmount250K-1M         0.063109    0.004443       14.21        < 2e-16 ***
FaceAmount50-100K        -0.171759    0.010461       -16.42       < 2e-16 ***
FaceAmountGT1M            0.078315    0.007342       10.67        < 2e-16 ***
FaceAmountLT50K          -0.333405    0.054839       -6.08        1.2e-09 ***
PremiumModemonthly       -0.413123    0.004736       -87.23       < 2e-16 ***
RiskClassSM               0.061092    0.006221        9.82        < 2e-16 ***
IssueAge30-34             0.105852    0.010531       10.05        < 2e-16 ***
IssueAge35-39             0.207189    0.010053       20.61        < 2e-16 ***
IssueAge40-44             0.301690    0.009946       30.33        < 2e-16 ***
IssueAge45-49             0.398574    0.009955       40.04        < 2e-16 ***
IssueAge50-54             0.474820    0.010132       46.86        < 2e-16 ***
IssueAge55-59             0.537070    0.010541       50.95        < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 18197.4  on 139  degrees of freedom
Residual deviance:  2395.1  on 127  degrees of freedom
AIC: 3461.5
```

The distribution of deviance residuals is displayed in a summary format. The deviance residuals are similar to the standardized error terms.

Following the list of deviance residuals are the predictor variable list, the coefficients and other statistics which have the same format as a standard Ordinary Least Squares (OLS) model.

The deviances of a null model and the current model are stated at the end of the output. The AIC (Akaike information criterion) is also calculated for generic GLM distributions such as the Poisson, Gamma, and Normal distributions. The last line of the output displays the number of iterations of numeric analysis in the GLM algorithm.

After initial iterations of the model, higher orders of covariates and cross-terms need to be considered to account for the significant interactive effects between the predictor variables.

For this particular sample dataset, the cross term between 'PremiumMode' and 'IssueAge' can be tested to improve the model's predictive power.

Here are the R script and results:

> Model2 <- glm(LapsedN~offset(log(Exposure))+FaceAmount+PremiumMode+RiskClass + IssueAge + PremiumMode:IssueAge, family=poisson(),data=lapseData)
> summary(Model2)

```
Call:
glm(formula = LapsedN ~ offset(log(Exposure)) + FaceAmount + PremiumMode
             + RiskClass + IssueAge + PremiumMode:IssueAge, family = poisson(),
             data = lapseData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.9645  -1.3702  -0.0883   1.0014   5.2205

Coefficients:
                                 Estimate    Std. Error  z value  Pr(>|z|)
(Intercept)                     -2.772184   0.010354  -267.74     < 2e-16  ***
FaceAmount250K-1M                0.063053   0.004444   14.188     < 2e-16  ***
FaceAmount50-100K               -0.182915   0.010468  -17.474     < 2e-16  ***
FaceAmountGT1M                   0.083356   0.007345   11.348     < 2e-16  ***
FaceAmountLT50K                 -0.341975   0.054840   -6.236     4.49e-10 ***
PremiumModemonthly              -0.775494   0.020480  -37.866     < 2e-16  ***
RiskClassSM                      0.060230   0.006220    9.684     < 2e-16  ***
IssueAge30-34                    0.079421   0.012037    6.598     4.17e-11 ***
IssueAge35-39                    0.149634   0.011510   13.001     < 2e-16  ***
IssueAge40-44                    0.206212   0.011421   18.056     < 2e-16  ***
IssueAge45-49                    0.272848   0.011431   23.870     < 2e-16  ***
IssueAge50-54                    0.321941   0.011663   27.602     < 2e-16  ***
IssueAge55-59                    0.362104   0.012186   29.716     < 2e-16  ***
PremiumModemonthly:IssueAge30-34 0.067422   0.024827    2.716     0.00661  **
PremiumModemonthly:IssueAge35-39 0.188513   0.023577    7.996     1.29e-15 ***
PremiumModemonthly:IssueAge40-44 0.343423   0.023177   14.817     < 2e-16  ***
PremiumModemonthly:IssueAge45-49 0.468724   0.023182   20.219     < 2e-16  ***
PremiumModemonthly:IssueAge50-54 0.578425   0.023481   24.634     < 2e-16  ***
PremiumModemonthly:IssueAge55-59 0.659804   0.024234   27.227     < 2e-16  ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 18197.4  on 139  degrees of freedom
Residual deviance:   427.6  on 121  degrees of freedom
AIC: 1506

Number of Fisher Scoring iterations: 4
```

CONTINUED ON **PAGE 34**

HOW TO BUILD A MODEL | FROM PAGE 33

As seen in the result, by adding the cross term, the AIC is significantly reduced from 3462 to 1506 and residual deviance decreases from 2,395 to 428. The inclusion of the cross term substantially improves our model's performance.

It is tempting to add as many cross-terms as possible to improve the model performance. However, it is important to balance the model fit with both simplicity and business judgment.

A model should be validated to test its effectiveness. There are many techniques available for this purpose; however, they will not be discussed here due to the scope of this brief introduction.

PREDICTION AND RESULT VISUALIZATION

After the model is built, the model is then used to predict lapse rates.

```
> lapseData$pred <-predict(Model1, lapseData,
type="response")
```

In this command, the model "Model1" is applied to the dataset "lapseData". The prediction is the response of the model, which is the predicted mean value. Other options are available, such as confidence level and uncertainty.

With both predicted values and observed values available, plots can be made to illustrate the model's goodness of fit by comparing the model's predicted lapses to the actual lapses.

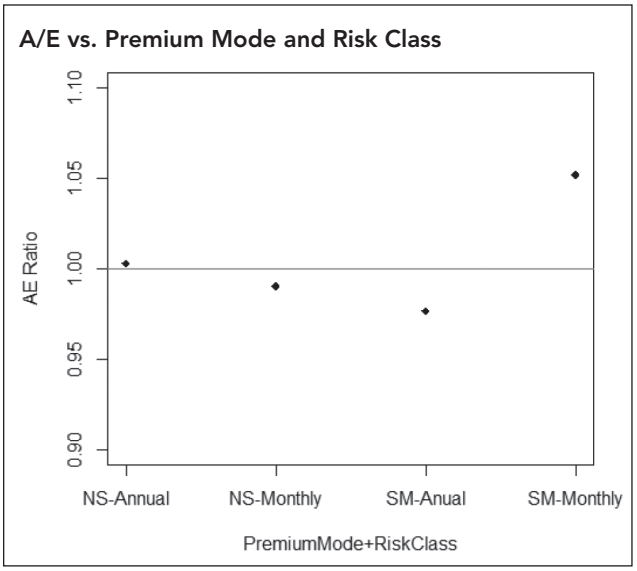
R has very strong built-in graphic capabilities. There are numerous packages available for data visualization. It is simple to export the plots to the clipboard or a stand-alone file in popular formats such as .pdf or .bmp. To make an A/E plot, data needs to be calculated and aggregated. In the following example, A/E is calculated by premium mode and risk class.

```
> byPred <- aggregate(pred ~
PremiumMode+RiskClass, data = lapseData, FUN =
sum)
> byObsv <- aggregate(LapsedN ~
PremiumMode+RiskClass, data = lapseData, FUN =
sum)
> AERatio <- byObsv[,3]/byPred[,3]
> AERatio
```

```
[1] 1.0030546 0.9903241 0.9767918 1.0517889
```

The last command displays the values of A/E ratios. Once the ratios are calculated, the following R scripts will plot the ratio, display the title, show the label on the X-axis, and draw a red line at 100% as reference:

```
> plot(AERatio,xlab="PremiumMode+RiskClass",
ylab="AE Ratio", xaxt='n', ylim=c(0.9,1.1), pch=18)
> title("A/E vs. Premium Mode and Risk Class")
> axis(1, at=1:4,labels=c("NS-Annual","NS-
Monthly","SM-Anual","SM-Monthly"), las=0)
> abline(1,0,col="red")
```



Another option is to export the results data to a file and perform data visualization in other applications such as Excel. This approach is probably more appealing to actuaries since actuaries are more familiar with Excel. The following script can be used to accomplish this:

> write.csv(lapseData,"modelDataFile.csv")

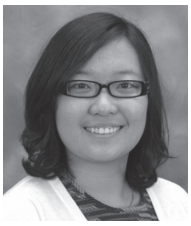
With this command, R will write the contents of "lapseData" into a file in the default directory with the name "modelDataFile.csv." ▼

Richard Xu

Richard Xu, FSA, Ph.D., is senior data scientist at RGA Reinsurance Company in Chesterfield, Mo. He can be reached at *rxu@rgare.com.*

Dihui Lai

Dihui Lai, Ph.D., is data scientist analyst at RGA Reinsurance Company in Chesterfield, Mo. He can be reached at *dlai@rgare.com.*



Minyu Cao

Minyu Cao, ASA, is an assistant actuary at RGA Reinsurance Company in Chesterfield, Mo. She can be reached at mcao@rgare.com.

Scott Rushing

Scott Rushing, FSA, is VP & actuary, Head of Global Research at RGA Reinsurance Company in Chesterfield, Mo. He can be reached at srushing@rgare.com.

Tim Rozar

Tim Rozar, FSA, CERA, is senior vice president at RGA Reinsurance Company in Chesterfield, Mo. He can be reached at trozar@rgare.com