Print

Summary: Lesson 1: Multiple Linear Regression

This summary contains topic summaries, syntax, and sample programs.

Topic Summaries

To go to the movie where you learned a task or concept, select a link.

Review of General Linear Models

The general linear model (or GLM) is a statistical linear model in which a continuous response variable (or dependent variable) Y is regressed on one or more predictor variables (or independent variables) X. In other words, the response Y is modeled as a linear function of predictor (X) variables.

The general linear model can be written as $Y = X\beta + \varepsilon$. In this notation, Y is a vector of continuous response measurements—in other words, Y is a set of response values that consists of one response for each observation in the data. X represents a matrix of predictor variables. β is a vector of model parameters, which are usually estimated. ε is a vector of random errors. Although this model has multiple predictors, it is not a multivariate analysis. It has only one response variable to model, so it is a univariate analysis. General linear models are often classified according to the types of predictor variables that they include.

To make sure that the results of linear regression are valid, it is necessary to verify that four assumptions for GLMs are true--the assumptions of linearity in the parameters, normality of the errors, constant variance of the errors, and independence of the errors.

Failing to meet the assumption of linearity in the parameters indicates that the model was misspecified. In this case, the model results are not meaningful. Violation of the normality assumption for the errors does not affect the parameter estimates. However, this violation does affect the tests of significance and the confidence intervals of the parameter estimates. Similarly, violation of the assumptions of constant variance and independence of errors does not affect the parameter estimates. However, violations of these assumptions compromise the standard errors.

There are two main types of linear regression models: simple linear regression and multiple linear regression. Simple linear regression has one predictor variable. Multiple linear regression models the relationship between multiple predictors and a response.

The multiple linear regression model is a GLM that models the response variable, Y, as a linear function of the *k* continuous predictor variables, the Xs. The model has $p = k + 1$ parameters (the βs) because it includes the intercept (β_0).

A baseline model is a model that has a slope of zero; in other words, all of the slope parameters equal zero. The null hypothesis is that the estimated regression model does not fit the data better than the baseline model. The alternative hypothesis is that the regression model does fit the data better than the baseline model.

To explore the relationships between variables, you can use the SGSCATTER procedure. In PROC SGSCATTER, you must use one of the following three statements: COMPARE, MATRIX, or PLOT. Each statement produces a different type of scatter plot panel. The COMPARE statement creates a comparative panel of scatter plots with shared axes. The MATRIX statement creates a scatter plot matrix. The PLOT statement creates a paneled graph that contains multiple independent scatter plots.

```
PROC SGSCATTER <options>;
    COMPARE X=variable | (variable-1...variable-n)
            Y=variable | (variable-1...variable-n) ;
    MATRIX variable-1 variable-2 < ... variable-n> </ options>;
    PLOT plot-request(s) </ options>;
RUN;
```

Two popular options that can be used in any of the three statements are ELLIPSE= and GROUP=. ELLIPSE= adds a confidence or prediction ellipse to the scatter plot. GROUP= specifies a classification variable to divide the values into groups.

ELLIPSE=

GROUP=

To fit a regression model to your data, you can choose among the following SAS procedures: PROC GLMSELECT, PROC REG, and PROC GLM. Of these three procedures, PROC GLMSELECT is the newest and it combines capabilities found in PROC REG and PROC GLM. It is important to note that some types of analyses, including selected regression diagnostics, are not yet available in PROC GLMSELECT. So, depending on the type of analysis you want to do, you might want to choose one of the other procedures to supplement the results.

In the PROC GLMSELECT statement, you can specify various options. As in other procedures, the DATA= option specifies the data set to use for the regression. The OUTDESIGN= option creates a data set that contains the design matrix. The EFFECT statement enables you to construct new effects, based on predictor variables in the input data set, that can be used as predictors in the model. n the MODEL statement, you specify the dependent variable (or response variable) and the model-effects (that is, predictor variables). The OUTPUT statement creates a new SAS data set that saves diagnostic measures calculated for the fitted model. You use the OUT= option to specify the data set name.

```
PROC GLMSELECT <options>;
     EFFECT effect-name=effect-type(<var-list></ options>);
     MODEL dependent=model-effects </ options>;
     OUTPUT <OUT=SAS-data-set> <keyword=<name>>
                                < ... keyword=<name>>;
RUN;
```

OUTDESIGN(*options*)**=***SAS-data-set*

Building a regression model is not the last step in the modeling process. You still need to check for common problems that are associated with regression, which include model misspecification, nonconstant variance, correlated error terms, influential observations, and multicollinearity.

The first three of these common regression problems can be identified using residual plots. Remember that residuals are the difference between each observed value of Y and its predicted value. An influential observation is an observation that is so far away from the rest of the data that it singlehandedly exerts influence on the slope of the regression line. Influential observations might affect your regression results, so it is important to identify them using statistics such as RSTUDENT residuals, DFFITS statistics, Cook's D, and DFBETAS. Multicollinearity can arise in multiple linear regression, and it occurs because two or more predictor variables used in the model are highly correlated. You can use the variance inflation factor and other multicollinearity diagnostic statistics to identify multicollinear predictor variables.

Simple Polynomial Regression

To describe a nonlinear relationship (such as a curvilinear relationship) between the predictors and the response, you use a polynomial regression model. A polynomial regression model is a special type of multiple linear regression model that includes higher-order terms, such as powers of variables (also referred to as polynomial terms) and cross-product (or interaction) terms.

PROC SGPLOT creates individual plots and charts of various types with extensive overlay capabilities. Each type of plot has its own statement. Depending on the type of plot, you must specify one of the following: a category-variable (which classifies the observant ions into distinct subsets), the response-variable, or variables

for the X and Y axes.

```
PROC SGPLOT <options>;
      DOT category-variable </options>;
      HBAR category-variable </options>;
      HBOX response-variable </options>;
      HISTOGRAM response-variable </options>;
      NEEDLE X=variable Y=numeric-variable </options>;
      REG X=numeric-variable Y=numeric-variable </options>;
      SCATTER X=variable Y=variable </options>;
      VBAR category-variable </options>;
      VBOX response-variable </options>;
RUN;
```

To add higher-order terms to your model, such as a variable raised to a power, you can use the EFFECT statement in PROC GLMSELECT. A higher-order variable that the EFFECT statement creates is called a constructed effect.

```
EFFECT effect-name=effect-type(<var-list></ options>);
```

Polynomial Regression and Multicollinearity

Multicollinearity affects the results of regression in the following ways: It affects the parameter estimates because they depend on the correlated independent variables included in the regression model. Note that, to test the null hypothesis that a parameter estimate is not significantly different from zero, you use the t-test.

Because multicollinearity has little effect on the overall fit of the equation, it also has little effect on the use of that equation for prediction, as long as the independent variables maintain the same pattern of multicollinearity in the future period that they demonstrated in the sample (Bowerman, O'Connell, and Dickey 1986).

To determine whether multicollinearity is present in a model that has two or more explanatory variables, you can examine the following diagnostic measures: correlation statistics, variance inflation factors (or VIFs), and condition index values.

Correlation is a measure of the degree of linear relationship between two variables. You can use correlation statistics to check for bivariate correlations between pairs of independent variables. VIFs are useful in determining whether multicollinearity exists or not, and which variables might be involved in the multicollinearity. Condition indices are the square roots of the ratios of the largest eigenvalues (the lambdas) to the individual i^{th} eigenvalues.

A very small eigenvalue that is close to zero implies severe multicollinearity. A zero eigenvalue indicates perfect multicollinearity among independent variables. Eigenvalues of relatively equal magnitudes indicate that there is little multicollinearity, and a wide variation in magnitudes indicates severe multicollinearity.

A correlation coefficient that is near +1 or -1 indicates a high degree of linear relationship between two regressors and might suggest multicollinearity. Similarly, a VIF in excess of 10 indicates strong multicollinearity. Conventionally, condition index values between 10 and 30 suggest weak dependencies and between 30 and 100 indicate moderate dependencies among predictors. However, condition index values greater than 100 indicate strong multicollinearity.

The proportion of variation explained by the principle components is another measure that you can use, in combination with the condition index, to diagnose multicollinearity. PROC REG calculates variance proportions for each term in the model.

To produce a detailed analysis of multicollinearity among the regressors in the model, you can use the COLLIN and COLLINOINT options in the MODEL statement in PROC REG. These options produce the same output with one difference: in the COLLINOINT output, the intercept variable is adjusted out instead of included in the diagnostics.

MODEL *dependent=model-effects* </ **COLLIN COLLINOUT**>;

After you diagnose multicollinearity in your model, and you identify the independent variables that are involved in the multicollinearity, your next step is to deal with it.

One method is to remove one or more independent variables, one at a time, from the model. Another method is to redefine one or more independent variables.

Biased regression provides biased parameter estimates but smaller standard errors compared to OLS estimates and standard errors. In PROC REG, you can use the MODEL statement with the RIDGE= option to perform the ridge regression analysis, and the MODEL statement with the PCOMIT= option to perform principal component regression analysis.

RIDGE=

PCOMIT=

To reduce or eliminate multicollinearity in polynomial regression models, you can also center the independent variables (Marquardt, D.W. 1980). To center a variable, you subtract a constant from every value of that variable. You can use PROC STDIZE to standardize the variable by substracting a location measure and dividing by a scale measure. The second method is to write SAS data steps to subtract the mean from the variables. And the third method is to use PROC GLMSELECT with the STANDARDIZE option in the EFFECT statement.

Modeling Nonlinear Relationships

In some situations, you might find that multiple independent variables have a nonlinear relationship with the dependent variable. When there are multiple nonlinear relationships, a multiple polynomial regression might be required to fit the data well.

After gaining insights from an initial exploration of the data, the next step is to identify candidate models by selecting variables. First, you can use your subject matter knowledge. Using only subject matter knowledge would be a subjective approach, but your knowledge can be helpful in combination with other tools.

You can also use the information that you gather from initial data exploration. PROC GLMSELECT offers you a choice of automatic variable selection methods, which are often called model selection methods. Sequential selection techniques include forward, backward, and stepwise selection. In each of these three approaches, the decision to include or exclude variables at each step is based on the model SBC by default.

PROC GLMSELECT also offers other automatic selection methods. The LAR (least angle regression) algorithm (Efron, 2006) produces a sequence of regression models. LASSO (least absolute shrinkage and selection operator) selection (Tibshirani, 1996) arises from a constrained form of ordinary least squares regression where the sum of the absolute values of the regression coefficients is constrained to be smaller than a specified parameter. Elastic net selection (Zhou, 2005) is yet another automatic selection method available in PROC GLMSELECT. PROC REG also offers additional model selection methods, such as all-possible model selection.

When PROC GLMSELECT and PROC REG create candidate models, they can use various model selection statistics to decide whether to include or exclude variables at each step in the process.

These model selection statistics are as follows:

- the significance level for individual variables
- the coefficient of determination (R square)
- the adjusted coefficient of determination (adjusted R square)
- Mallows' C_p statistic
- Akaike's information criterion (AIC and AICC), and
- Schwarz's Bayesian criterion (SBC)

Most of these model selection statistics are based on the assumption that you want to create a model that minimizes the unexplained variability (the mean square error) with the smallest possible number of variables (a principle known as parsimony). PROC REG uses only the significance level (or *p*-value) for each individual

variable to determine whether the variable enters the model (in forward selection), leaves the model (in backward elimination), or either (in stepwise selection).

PROC GLMSELECT uses the SBC for variable selection. In PROC GLMSELECT, to use the significance level instead of the SBC for variable selection, you specify the SELECT=SL option in the MODEL statement. Then you use the SLENTRY= and SLSTAY= options to specify the significance levels for selection.

SELECT=SL

SLENTRY=

SLSTAY=

PROC GLMSELECT also enables you to specify model selection statistics other than the significance level. R square (the coefficient of determination) measures the proportion of variability in the response variable that is explained by the predictor variables. The adjusted R square is similar to R square, but it penalizes for additional terms in the model. Therefore, when you compare models that have a different number of predictor variables, it is more appropriate to use the adjusted R square.

Mallows' C_p is a simple indicator of model misspecification. This statistic helps to detect and avoid model bias, which refers to either underfitting the model (leaving out important terms) or overfitting the model (including too many terms).

Akaike introduced the concept of information criteria as a tool for optimal model selection. Akaike's Information Criterion (AIC) is a function of *n* (the number of observations), the SSE (sum of the squared error), and *p* (the number of parameters in the model, including the intercept). The AICC is the AIC with a correction for finite sample sizes. Schwarz developed a model selection criterion that is derived from the Bayesian modification of the AIC criterion. However, the SBC has a larger penalty for additional parameters in the model.

After you use PROC GLMSELECT to generate candidate models, you typically want to select a final model. However, in some cases, you might need to base your selection of a model on additional considerations, such as the following: PROC GLMSELECT results likely do not include all possible candidate models. To produce a fairly complete array of candidate models, you would need to specify all possible combinations of model selection methods and model selection statistics.

Some nonlinear relationships between predictors and a response variable might be too complex for a simple polynomial function to model adequately.

Spline functions offer a useful way to perform this type of piecewise polynomial fitting and incorporate the complex nonlinear predictor-response relationships into a regression model. A spline is a smooth function consisting of piecewise polynomials joined at points called knots. To create spline effects, you can use the EFFECT statement in PROC GLMSELECT and specify the SPLINE effect-type.

EFFECT *effect-name=effect-type***();**

SPLINE

Sample Programs

Exploring the Data

```
proc sgscatter data=mydata.school;
   compare y=reading3 x=(words1 letters1 phonics1);
   title 'Scatter Plots of READING3 by WORDS1 LETTERS1 and PHONICS1';
run;
```

Building a Multiple Regression Model

```
title 'School Data: Regression and Diagnostics';
proc glmselect data=mydata.school;
          model reading3 = words1 letters1 phonics1 / selection=none;
          output out=out r=residuals;
run;

proc univariate data=out;
   var residuals;
   histogram residuals / normal kernel;
   qqplot residuals / normal(mu=est sigma=est);
run;
title;
```

Building a Simple Polynomial Regression Model

```
title1 "Paper Data Set";

proc sgplot data=mydata.paper;
   scatter x=amount y=strength;
title2 "Scatter Plot";
run;

proc sgplot data=mydata.paper;
   reg  x=amount y=strength / lineattrs =(color=brown
         pattern=solid) legendlabel="Linear";
title2 "Linear Model";
run;

proc sgplot data=mydata.paper;
   reg  x=amount y=strength / degree=2 lineattrs =(color=green
        pattern=mediumdash) legendlabel="2nd Degree";
title2 "Second Degree Polynomial";
run;

proc sgplot data=mydata.paper;
   reg  x=amount y=strength / degree=3 lineattrs =(color=red
        pattern=shortdash) legendlabel="3rd Degree";
title2 "Third Degree Polynomial";
run;

proc sgplot data=mydata.paper;
   reg  x=amount y=strength / degree=4 lineattrs =(color=blue
        pattern=longdash) legendlabel="4th Degree";
title2 "Fourth Degree Polynomial";
run;
title;

title2;

proc glmselect data=mydata.paper outdesign=d_paper;
          effect p_amount=polynomial(amount / degree=4);
          model strength = p_amount / selection=none;
          title "Paper Data Set: 4th Degree Polynomial";
run;

proc glmselect data=mydata.paper outdesign=d_paper;
          effect p_amount=polynomial(amount / degree=4);
          model strength = p_amount / selection=backward select=sl slstay=0.05
```

```
         hierarchy=single showpvalues;
          title "Paper Data Set: Model Selection";
run;

proc reg data=d_paper plots (unpack) =(diagnostics (stats=none));
   Cubic_Model: model strength=&_GLSMOD / lackfit;
   title "Paper Data Set: 3rd Degree Polynomial Model";
run;
quit;
```

Performing Multicollinearity Diagnostics

```
title 'Collinearity Diagnosis for the Cubic Model';
proc corr data=d_paper nosimple plots=matrix;
   var &_GLSMOD;
run;

proc reg data=d_paper plots=none;
   model strength=&_GLSMOD / vif collin collinoint;
run;
quit;
title;
```

Centering Variables

```
proc glmselect data=mydata.paper outdesign=dc_paper;
   effect qc_amount=polynomial(amount /
          degree=3 standardize(method=moments)=center);
   model strength = qc_amount / selection=none;
   title "Paper Data Set: Centered Cubic Model";
run;

ods select ParameterEstimates CollinDiag CollinDiagNoInt;
proc reg data=dc_paper;
   model strength = &_GLSMOD / vif collin collinoint;
   title 'Diagnostics for Centered Cubic Model';
run;
title;
quit;

proc stdize data=mydata.paper method=mean out=paper1(rename=(amount=mcamount));
   var amount;
run;

data paper1;
   set paper1;
   mcamount2 = mcamount**2;
   mcamount3 = mcamount**3;
run;

proc print data=paper1;
run;

proc sql;
   select mean(amount) into: mamount
   from mydata.paper;
run;

data paper2;
   set mydata.paper;
   mcamount=amount-&mamount;
   mcamount2 = mcamount**2;
   mcamount3 = mcamount**3;
run;
```

```
proc print data=paper2;
run;
```

Exploring the Data

```
proc sgscatter data=mydata.cars;
   plot price*(citympg hwympg cylinders enginesize horsepower fueltank
        luggage weight);
run;

ods graphics / imagemap=on;

proc sgscatter data=mydata.cars;
   plot price*(citympg hwympg fueltank weight) / pbspline;
run;

proc corr data=mydata.cars nosimple;
   var price citympg hwympg cylinders enginesize horsepower fueltank
       luggage weight;
run;

proc sgscatter data=mydata.cars;
   matrix citympg hwympg cylinders enginesize horsepower fueltank
          luggage weight;
run;
```

Selecting Candidate Models

```
ods graphics / reset=all;

title 'Model Selection Cars2 Data Set';

%macro p_eff;
effect p_city = polynomial(citympg /degree=2
standardize(method=moments)=center);
effect p_hwy = polynomial(hwympg /degree=2
standardize(method=moments)=center);
effect p_fuel = polynomial(fueltank /degree=3
standardize(method=moments)=center);
%mend;

proc glmselect data=mydata.cars2 plots=criteria;
   title2 'Backward elimination with significance levels';
   %p_eff;
   model price = p_city p_hwy cylinders enginesize horsepower p_fuel
                 luggage weight / selection=backward select=sl
                 slstay=0.05 hierarchy = single;
run;

proc glmselect data=mydata.cars2;
   title2 'Forward selection with significance levels';
   %p_eff;
   model price = p_city p_hwy cylinders enginesize horsepower p_fuel
                 luggage weight / selection=forward select=sl
                 slentry=0.1 hierarchy=single;
run;

proc glmselect data=mydata.cars2;
   title2 'Backward elimination using SBC';
   %p_eff;
   model price = p_city p_hwy cylinders enginesize horsepower p_fuel
                 luggage weight / selection=backward select=sbc
                 hierarchy=single;
run;
```

```
proc glmselect data=mydata.cars2 plots=criteria;
   title2 'Backward elimination using adjusted R-square';
   %p_eff;
   model price = p_city p_hwy cylinders enginesize horsepower p_fuel
                 luggage weight / selection=backward select=adjrsq
                 hierarchy=single;
run;
title;
```

Modeling with Splines

```
title 'Spline Effect with Cars Dataset';

proc glmselect data=mydata.cars2;
   title2 'Cubic polynomial for Fueltank';
   effect p_fuel = polynomial(fueltank /degree=3
            standardize(method=moments)=center);
   model price = p_fuel / selection=none;
run;

proc glmselect data=mydata.cars2;
   title2 'Spline for Fueltank';
   effect sp_fuel = spline(fueltank / details);
   model price = sp_fuel / selection=none;
run;
```


Close