Getting Classy: A SAS® Macro for CLASS Statement Automation

Erica Goodrich M.S., Brigham and Women’s Hospital, Boston, MA
Daniel Sturgeon, M.S., Brigham and Women’s Hospital, Boston, MA
Kathryn Schurr M.S., Quest Diagnostics, Hudsonville, MI
ABSTRACT

When creating statistical models that include multiple covariates, it is important to address which variables are considered categorical and continuous for proper analysis and interpretation in SAS®. Categorical variables, regardless of SAS® data type, should be added in the model statement along with an additional class statement. For larger models which contain many continuous or categorical variables, it is easy to overlook variables that need to be added to the class statement.

To solve this issue we have created the %MASTERCLASS macro. This macro uses simple inputs including a model variable list and a dataset to create automatically generated class and model Variable listings for modeling.

Example of a Statistical Model using a CLASS statement

Below is an example of code which could be used for a Logistic Regression. While the model has multiple covariates included, only two of them are listed in the class statement.

PROC LOGISTIC DATA=CARS DESCENDING;
CLASS DRIVETRAIN(REF='All') ORIGIN(REF='Asia')/PARAM=REF;
MODEL UNDER25K = ORIGIN MPG_CITY WEIGHT DRIVETRAIN ENGINESIZE HORSEPOWER MPG_HIGHWAY WEIGHT WHEELBASE LENGTH;
RUN;

This is a simple process when you are familiar with your dataset or have a small list of covariates considered in a model.

This becomes much more work when multiple models, or large numbers with small changes inside each model, or models with an extremely large number of covariates. Our macro helps improve efficiency and quality of class statement creation.
Getting Classy: A SAS® Macro for CLASS Statement Automation

Erica Goodrich M.S., Brigham and Women’s Hospital, Boston, MA
Daniel Sturgeon, M.S., Brigham and Women’s Hospital, Boston, MA
Kathryn Schurr M.S., Quest Diagnostics, Hudsonville, MI

The example used for this macro will be shown with the SASHELP.CARS dataset with a small modification. Variable names, types, and formats are listed below.

Here’s a bit of background on the CARS dataset:

• Includes 16 Variables:
  5 character types
  10 numeric types
  1 numeric with a small number of levels
  1 created binary variable
  (Created based on MSRP <\= $25,000)

• Mock examples show modeling for MSRP or costs above or below 25k MSRP.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cylinders</td>
<td>Num</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>DriveTrain</td>
<td>Char</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>EngineSize</td>
<td>Num</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Horsepower</td>
<td>Num</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Invoice</td>
<td>Num</td>
<td>8</td>
<td>DOLLAR.</td>
</tr>
<tr>
<td>Length</td>
<td>Num</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>MPG_City</td>
<td>Num</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>MPG_Highway</td>
<td>Num</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>MSRP</td>
<td>Num</td>
<td>8</td>
<td>DOLLAR.</td>
</tr>
<tr>
<td>Make</td>
<td>Char</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>Model</td>
<td>Char</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>Origin</td>
<td>Char</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Char</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>UNDER25K</td>
<td>Num</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td>Num</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Wheelbase</td>
<td>Num</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

All you need is the dataset of interest, the %MasterCLASS macro, the model of interest covariates. From there, the macro is simple to use with the example prompt listed below:

Input:
(Code) %MasterCLASS(DATA = <dataset>, VARS = <variables>, LEVELS = <number of levels>);
(Example) %MasterCLASS(DATA=SASHELP.CARS, VARS=ORIGIN MAKE TYPE MPG_CITY WEIGHT Cylinders DriveTrain EngineSize Horsepower mpg_Highway weight wheelbase length);

Output:
Three macro variables: &CLASS, &VARIABLE, &MODEL.

%PUT &MODEL.;
Cylinders DriveTrain Make Origin Type MPG_CITY Weight EngineSize Horsepower MPG_Highway Weight Wheelbase Length

% PUT &CLASS.;
Cylinders(REF='3') DriveTrain(REF='All') MAKE(REF='Acura') Origin (REF='Asia')
Type(REF='Hybrid')

% PUT &VARIABLE.;
Cylinders DriveTrain Make Origin Type
Very little of the details are needed to work this macro! However, behind the scenes this is what’s going on:

- User identifies dataset and variables of interest
- Metadata is gathered from PROC CONTENTS
- The metadata variables are separated by numeric and character types
- Logic checks for variables assigned as numeric which should be identified as categorical types to be included in CLASS statement
  - e.g. 0/1 binary variables or the “Cylinders” variable within SASHELP.CARS
- Unique levels are obtained for all variables in the class statement
- Macro variables are created for class and model statement.
  - Auto generated references groups (i.e. Type(REF='Hybrid')) can be auto-included for the CLASS statement
- End user just needs to know macro prompts!

Macro information and code available by phone!

%MacroCLASS code can be found with the QR code above or at https://tinyurl.com/h6mcd45
This code is also available in the published paper and online at the QR code posted on the previous slide.