# %PHChoice Macro

The `%PHChoice` autocall macro customizes the output from PROC PHREG for choice modeling. Typically, you run the following macro once to customize the PROC PHREG output as follows:

```
%phchoice(on)
```

The macro uses PROC TEMPLATE and ODS (Output Delivery System) to customize the output from PROC PHREG. Running this code edits the templates and stores copies in SASUSER. These changes will remain in effect until you delete them. Note that these changes assume that each effect in the choice model has a variable label associated with it so there is no need to display variable names. If you are coding with PROC TRANSREG, this will usually be the case. You can submit the following step to return to the default output from PROC PHREG:

```
%phchoice(off)
```

If you ever have errors running this macro, like invalid page errors, see "Macro Errors" on page 1211. The rest of this section discusses the details of what the `%PHChoice` macro does and why. Unless you are interested in further customization of the output, you should skip to "`%PHChoice` Macro Options" on page 1177.

We are most interested in the `Analysis of Maximum Likelihood Estimates` table, which contains the parameter estimates. We can first use PROC TEMPLATE to identify the template for the parameter estimates table and then edit the template. First, let's have PROC TEMPLATE display the templates for PROC PHREG. The `source stat.phreg` statement in the following step specifies that we want to see PROC TEMPLATE source code for the STAT product and the PHREG procedure:

```
proc template;
   source stat.phreg;
   run;
```

If we search the results for the `Analysis of Maximum Likelihood Estimates` table we find the following code, which defines the `Stat.Phreg.ParameterEstimates` table:

```
define table Stat.Phreg.ParameterEstimates;
   notes "Parameter Estimates Table";
   dynamic Confidence NRows;
   column Variable DF Estimate StdErr StdErrRatio ChiSq ProbChiSq HazardRatio
      HRLowerCL HRUpperCL Label;
   header h1 h2;

   define h1;
      text "Analysis of Maximum Likelihood Estimates";
      space = 1;
      spill_margin;
   end;
```

```
define h2;
   text Confidence BEST8. %nrstr("%% Hazard Ratio Confidence Limits");
   space = 0;
   end = HRUpperCL;
   start = HRLowerCL;
   spill_margin = OFF;
end;

define Variable;
   header = "Variable";
   style = RowHeader;
   id;
end;

define DF;
   parent = Common.ParameterEstimates.DF;
end;

define Estimate;
   header = ";Parameter;Estimate;";
   format = D10.;
   parent = Common.ParameterEstimates.Estimate;
end;

define StdErr;
   header = ";Standard;Error;";
   format = D10.;
   parent = Common.ParameterEstimates.StdErr;
end;

define StdErrRatio;
   header = ";StdErr;Ratio;";
   format = 6.3;
end;

define ChiSq;
   parent = Stat.Phreg.ChiSq;
end;

define ProbChiSq;
   parent = Stat.Phreg.ProbChiSq;
end;

define HazardRatio;
   header = ";Hazard;Ratio;";
   glue = 2;
   format = 8.3;
end;

define HRLowerCL;
   glue = 2;
   format = 8.3;
   print_headers = OFF;
end;
```

```
    define HRUpperCL;
        format = 8.3;
        print_headers = OFF;
    end;

    define Label;
        header = "Variable Label";
    end;

    col_space_max = 4;
    col_space_min = 1;
    required_space = NRows;
end;
```

It contains header, format, spacing and other information for each column in the table. Most of this need not concern us now.* The template contains the following `column` statement, which lists the columns of the table:

```
    column Variable DF Estimate StdErr StdErrRatio ChiSq ProbChiSq HazardRatio
        HRLowerCL HRUpperCL Label;
```

Since we will usually have a label that adequately names each parameter, we do not need the variable column. We also do not need the hazard information. If we move the label to the front of the list and drop the variable column and the hazard columns, we get the following:

```
    column Label DF Estimate StdErr ChiSq ProbChiSq;
```

We use the `edit` statement to edit the template. We can also modify some headers. We specify the new `column` statement and the new headers. We can also modify the Summary table, which is `Stat.Phreg.CensoredSummary`, to use the vocabulary of choice models instead of survival analysis models. The code is generated by the PROC TEMPLATE step with the `source` statement. The overall header "Summary of the Number of Event and Censored Values" is changed to "Summary of Subjects, Sets, and Chosen and Unchosen Alternatives", "Total" is changed to "Number of Alternatives", "Event" is changed to "Chosen Alternatives", "Censored" is changed to "Not Chosen", and "Percent Censored" is dropped. Finally `Style=RowHeader` was specified on the label column. This sets the color, font, and general style for HTML output. The `RowHeader` style is typically used on first columns that provide names or labels for the rows. The following step contains the code that the `%phchoice(on)` macro runs:

```
proc template;
    edit stat.phreg.ParameterEstimates;
        column Label DF Estimate StdErr ChiSq ProbChiSq;
        header h1;

        define h1;
            text "Multinomial Logit Parameter Estimates";
            space = 1;
            spill_margin;
            end;
```

---

*In fact, this is an older version of the template. It has changed. Still, this is the version that the macro was based on, so it is still shown here.

```
      define Label;
         header = " " style = RowHeader;
         end;
      end;

      edit Stat.Phreg.CensoredSummary;
         column Stratum Pattern Freq GenericStrVar Total
                Event Censored;
         header h1;
         define h1;
            text "Summary of Subjects, Sets, "
                 "and Chosen and Unchosen Alternatives";
            space = 1;
            spill_margin;
            first_panel;
         end;

         define Freq;
           header=";Number of;Choices" format=6.0;
         end;

      define Total;
         header = ";Number of;Alternatives";
         format_ndec = ndec;
         format_width = 8;
      end;

      define Event;
         header = ";Chosen;Alternatives";
         format_ndec = ndec;
         format_width = 8;
      end;

      define Censored;
         header = "Not Chosen";
         format_ndec = ndec;
         format_width = 8;
      end;
      end;

   run;
```

The **%phchoice(off)** macro runs the following step:

```
   * Delete edited templates, restore original templates;
   proc template;
      delete Stat.Phreg.ParameterEstimates;
      delete Stat.Phreg.CensoredSummary;
      run;
```

Our editing of the multinomial logit parameter estimates table assumes that each independent variable has a label. If you are coding with PROC TRANSREG, this is true of all variables created by `class` expansions. You might have to provide labels for `identity` and other variables. Alternatively, if you want variable names to appear in the table, you can do that as follows:

```
%phchoice(on, Variable DF Estimate StdErr ChiSq ProbChiSq Label)
```

This might be useful when you are not coding with PROC TRANSREG. The optional second argument provides a list of the column names to display. The available columns are: `Variable DF Estimate StdErr StdErrRatio ChiSq ProbChiSq HazardRatio HRLowerCL HRUpperCL Label`. (`HRLowerCL` and `HRUpperCL` are confidence limits on the hazard ratio.) For very detailed customizations, you might have to run PROC TEMPLATE directly.

# %PHChoice Macro Options

The following options can be used with the `%PHChoice` macro:

| Option | Description |
|--------|-------------|
| onoff | (positional) `on`, `off`, or `expb` |
| | (positional) "help" or "?" displays syntax summary |
| column | (positional) list of columns |

You can specify either of the following to display the option names and simple examples of the macro syntax:

```
%phchoice(help)
%phchoice(?)
```

The `%PHChoice` macro has two positional parameters, `onoff` and `column`. Positional parameters must come first, and unlike all other parameters, are not specified after a name and an equal sign.

## onoff

`ON` specifies choice model customization.
`OFF` turns off the choice model customization and returns to the default PROC PHREG templates.
`EXPB` turns on choice model customization and adds the hazard ratio to the output.
Upper/lower case does not matter.

## column
specifies an optional column list for more extensive customizations.