OGRS (Omnibus Groups Regions of Significance)

```
%OGRS(DATA = filename, VARS = xvar mvar yvar [cov1 cov2 …],
                                X = xvar, M = mvar, Y = yvar
                                          [,CONF = {c}{95**}]
                                [,CONVCRIT = {cc}{.00000001**}]
                                          [,ITER = {it}{0**}]
                                [,DECIMALS = {dec}{10.4**}];
```

Subcommands in brackets are optional
** Default if subcommand is omitted

Overview

OGRS is a macro that estimates a linear regression model where the effect of the categorical independent variable (*X*) on the outcome (*Y*) is allowed to depend linearly on a moderator (*M*). OGRS provides all least squares regression estimates, with standard errors, t-statistics, p-values, and confidence intervals. As well, OGRS provides a test of interaction, using hierarchical regression analysis, comparing a model where the effect of *X* is allowed to vary linearly with *M* and a model where the effect of *X* is fixed across *M*.

OGRS also provides a unique method for probing the effect of *X* on *Y* using an approximation of the Johnson-Neyman procedure. OGRS searchers the observed range of the moderator for points at which the effect of *X* on *Y* transitions from significant to non-significant or vice versa, as specified by some level of confidence (CONF). OGRS prints the transition points if any exist within the observed range of the moderator, as well as a table of points along the moderator, statistics related to the estimated effect of *X* at that point, and inferential statistics for the effect of *X*.

Preparation for Use

The OGRS.sas file should be opened as a program file in SAS. Once opened, execute the entire file exactly as is. Do not modify the code. Once the program is executed, the OGRS.sas program can be closed. Once executed, access to the OGRS command is available until quitting SAS. The OGRS.sasfile must be loaded and re-executed each time SAS is opened.

Model Specification

Because OGRS will only accept one variable in the `X` subcommand, your independent variable should be coded into one variable, with unique codes for each group. Categorical independent variables can represent groups (e.g. race, political party), experimental conditions, or any other categorical variable of interest. Both the moderator M and the outcome variable Y are treated as continuous variables. Covariates specified in the model can be continuous or dichotomous, but they cannot be categorical with more than two groups. To use categorical covariates, using your

desired coding scheme (e.g. dummy coding) to create *k*-1 new variables to represent your categorical covariate, and include all of these variables as covariates in the model. Covariates can be included in the `vars` subcommand, and they will be included in the regression model.

Various options in OGRS allow you to tailor the output to your question of interest. You can specify a confidence level, the convergence criteria for the Johnson-Neyman approximation, number of initial iterations in the Johnson-Neyman approximation, and the number of decimal places printed in the output. For example:

```
%OGRS(data = datname, vars = Xvar Mvar Yvar Cov1 ,conf = 90
,convcrit = .0001 ,iter = 1000 ,decimals = 10.6);
```

will estimate the effect of a categorical variable `Xvar` on `Yvar` moderated by `Mvar` from the dataset called `datname`. All confidence intervals will be 90% confidence intervals, and the Johnson-Neyman procedure will solve for points of transition where the effect of `Xvar` on will estimate the effect of a categorical variable `Xvar` on `Yvar` moderated by `Mvar`. All confidence intervals will be 90% confidence intervals, and the Johnson-Neyman procedure will solve for points of transition along `Mvar` where the effect of `Xvar` on `Yvar` is exactly significant at α = 0.10. The convergence criteria will be .0001 rather than the typical .00000001, and the Johnson-Neyman approximation will begin with 1000 iterations. All output will be printed to six decimals places.

Confidence Level

The `c` argument in the `CONF` subcommand specifies that confidence level for all confidence intervals and the criterion value for which the Johnson-Neyman procedure will compute the boundaries of significance. The default is 95%. Users can specify any confidence level greater than 50 and less than 100. For example `CONF = 90` will result in 90% confidence intervals and for the Johnson-Neyman procedure to find the points along the moderator at which the effect of the independent variable on the outcome variable is exactly significant at α = .10.

Convergence Criteria

The `cc` argument in the `CONVCRIT` subcommand specifies the convergence criteria for the Johnson-Neyman algorithm in finding the boundaries of significance. The default is .00000001. Users can specify any number greater than .00000001. For example `CONVCRIT = .001` will mean that any solution which has an *F*-statistic within .001 of the criterion *F*-statistic will be considered a sufficient solution for the Johnson-Neyman boundary of significance.

Initial Iterations

The `it` argument in the `ITER` subcommand specifies how many initial iterations should be used in the Johnson-Neyman algorithm. The default setting says 0 but this is used as an indicator that the user wants to use the default setting which is 50+*10k* where *k* is the number of groups in the variable specified in the `X` subcommand. Users can specify any whole number larger than 1 for

this argument. For example, `ITER = 100` will result in the Johnson-Neyman algorithm dividing the range of the moderator into 100 sections in the initial search step. Users should be aware that large numbers of iterations may cause the program to run for a long time, so be patient. Additionally, very small numbers of iterations may cause the algorithm to miss potential transition points.

Decimals

The `dec` argument in the `DECIMALS` subcommand specifies how many decimal places are printed in the output. The default for this is F10.4. The user can specify any format which is a valid printable numeric format (See SAS Manual). For example `DECIMALS = 10.2` will print all outputs to two decimal places.