Paper CC-02-2011

RDPLOT: A SAS® Macro for Generating Regression Discontinuity Plots

Jason A. Schoeneberger
University of South Carolina

ABSTRACT

Applied social science and education researchers often face less than ideal research design conditions for understanding interventions or programs. A lack of willingness on the part of policymakers to utilize random assignment in these fields prevents researchers from capitalizing on the high levels of internal validity offered by such a design. One research design that offers a palatable alternative to all stakeholders is the regression discontinuity design, where a pre-intervention measure is used to determine what research subjects receive the intervention. Typically all subjects below (or above) some cut-point on the pre-intervention measure continuum are assigned to receive the intervention, while all other subjects serve as the control. In this design, the interest is centered on the post-intervention measure for those students just on either side of the cut-point on the pre-intervention measure. A regression model is used to determine whether a significant difference exists between the intervention and control groups on the post-intervention measure. A treatment effect creates a discontinuity in the regression line, depicting the advantage (or disadvantage) associated with the intervention. Interpretation of these designs is aided by examination of a graphical depiction of the discontinuity in the regression line. The macro described herein was designed to facilitate the generation of regression discontinuity plots under homogenous or heterogenous conditions through the entry of eight parameters into the RDPLOT SAS® macro.

Keywords: REGRESSION DISCONTINUITY, GPLOT, SAS/GRAPH

INTRODUCTION

Applied social science and education researchers often face less than ideal research design conditions for understanding interventions or programs, including a lack of opportunity for random assignment and the accompanying high level of internal validity that go along with it. One research design that has been offered as a near-equivalent alternative (Imbens, 2008) is the regression discontinuity design. Regression discontinuity was an attempt to deal with selection bias, by using the cutoff score to fully determine treatment exposure and function as an experiment limited to the scores immediately around the cut-off (Thistlethwaite & Campbell, 1960).

The regression discontinuity approach has a long history of development and use spanning the educational and psychology research, economics and statistics paradigms (Cook, 2008). Further, there have been many contributions to the development of the design, including fuzzy regression discontinuity when heterogeneity is present (Hahn, Todd, and Van Der Klaauw, 2001), methods for identifying bandwidths along the predictor continuum (Ludwig and Miller, 2005) and tests for discontinuities in means and distributions of non-affected variables (Lee, 2007; McCrary, 2007). In this design, a pre-intervention measure is used to determine which research subjects receive the intervention. The regression discontinuity analysis can be as credible as a randomized experiment provided that the treatment-determining covariate is partly determined by a random chance element. Though advocates of regression discontinuity do not view it as superior to the randomized experiment, it can be used in a specific set of circumstances as a strategy superior to other known causal methods (Imbens, 2008).

Typically all subjects below (or above) some cut-point on the pre-intervention measure continuum are assigned to receive the intervention, while all other subjects serve as the control. In this design, the interest is centered on the post-intervention measure for those subjects just on either side of the cut-point on the pre-intervention measure. Though regression models generate an estimate associated with the effect of the treatment, graphical depictions of the results are often more easily understood by non-technical audiences. The current paper outlines a method for generating graphical plots stemming from a regression discontinuity design in the case where the treatment-determining covariate is assumed continuous, allowing for heterogenous regression lines above and below the cut-point.

MACRO RDPLOT DETAILS

RDPLOT works with flat data files (one row per subject of interest) containing pretest and posttest measures, along with a variable identifying group membership (treatment or control). The macro has a total of eight macro arguments the user can supply to generate the regression discontinuity graphs. Here each argument is defined, including any rules that must be adhered to when supplying information to the macro.

path: folder location where the data file for analysis resides

data_in: this argument is the name of the data file containing the data to be plotted in the regression discontinuity plot. This file should be structured to contain one record per intended unit of analysis.
pretest: the name of the pre-test, or pre-intervention measure used to decide treatment group membership
posttest: the name of the post-test, or post-intervention outcome measure
group: the numeric, binary indicator variable denoting treatment group membership (received treatment=1, did not receive treatment=0)
cutscore: the value on the pre-test measure functioning as the cut-score, denoting treatment group membership
treat_loc: the location, either above or below the cut-score, of the treatment group. This argument accepts the value 'above' or 'below'
plot_data: this argument allows the user to determine whether the individual data points are plotted on the graph, along with the summary regression lines. This argument accepts values of 'Y' and 'N'

Below is an example of the RDPLOT macro, along with the technical information to help the user understand the various steps within the macro. For this illustration, the pre and post-test measures are continuous variables where the cut-score is 14.5 and the group receiving treatment scored below the cut-score.

A complete version of the macro, sans the technical discussions provided here, is also provided at the end of the paper. Similarly, the macro can also be obtained from the author by emailing schoeneb@email.sc.edu.

The macro begins by establishing a copy of the input data file in the work directory, in addition to an interaction variable through multiplication of the GROUP and PRETEST macro arguments. Further, two variables called control and treatment are set equal to the value of the POSTEST macro argument dependent upon the value of the GROUP argument present in the data file. The interaction variable is used to test for heterogeneity among the treatment and control group along the pretest continuum.

```
**create regdis data file, with pretest and posttest supplied by user;
**also create centered pre-test score using user supplied cut-score value;
**finally, create interaction variable for testing heterogeneity;
data regdis;
	set "%bquote(&path)%bquote(&data_in)";
	interact=&group*&pretest;
	if (&group=0) then control=&posttest;
	if (&group=1) then treatment=&posttest;
	label control="Control" treatment="Treatment";
	run;
```

The next PROC SQL statement defines macro arguments representing the minimum and maximum values for the pretest and posttest that will be necessary for the annotated 'drawing' of the regression lines when generating the plot.

```
**this sql puts the minimum/maximum values of precut into x_min/x_max macro arguments;
proc sql noprint;
	select min(&pretest), max(&pretest), min(&posttest), max(&posttest)
		into: x_min, : x_max, : y_min, : y_max
		from regdis;
	quit;
```

PROC GLM uses the input data file to generate the analytic model including the interaction to test for homogeneity. Absence of a significant interaction effect supports homogeneity. The Output Delivery System (ODS) is used to capture the parameters of interest from PROC GLM for further manipulation. A macro argument 'HETERO' is defined based upon the significance of the interaction, where a significant value sets HETERO='Y', suggesting heterogeneity is present necessitating separate processing than when homogeneity is present.

```
**the following glm tests the interaction (homogeneity assumption);
ods output parameterestimates=paramsint;
proc glm data=regdis namelen=32;
	model &posttest=&group interact &pretest/ss1 solution;
	run;
	quit;

**grab parameter table and determine if interaction is significant;
data paramsint;
	set paramsint;
	if (parameter="interact") and (probt gt .05) then call symput('hetero','N');
	if (parameter="interact") and (probt le .05) then call symput('hetero','Y');
	run;
```

Under homogeneity, PROC GLM is used again to generate parameter estimates without including the interaction term. The estimates are captured again via ODS, and the PROC SQL "into" feature is used to create macro arguments representing the necessary parameters for generating the plots, including the intercept, the slope associated with the group variable supplied by the user and the slope associated with the pretest.

```
**the following will run only when homogeneity is assumed (hetero='N');
%if "&hetero"="N" %then %do;
**this glm generates the parameter estimates for generating plots - assumes
homogeneity;
ods output parameterestimates=params;
proc glm data=regdis namelen=32;
      model &posttest=&group &pretest/ss1 solution;
      run;
      quit;

**this sql places the glm parameter estimates into macro variables for intercept and
slopes;
proc sql noprint;
      select estimate into: intercept
            from params
            where parameter="Intercept";
      select estimate into: grp_slp
            from params
            where parameter=lowcase("&group");
      select estimate into: cut_slp
            from params
            where parameter=lowcase("&pretest");
      quit;
```

The SAS annotate capability is used to generate a dataset that will instruct SAS how to draw the regression lines that will represent the treatment and control groups respective lines. The macro allows the user to choose whether the treatment or control group is plotted to the left or right of the chosen cut-point along the pre-test continuum. Beginning by creating a dataset representing the group plotted to the left, the macro tells SAS via the "function='move'" command where to put the 'pen' down to begin drawing the line. In this case, it begins at the intersection of the x- and y-axis at the minimum value of the pretest value defined in the X_MIN macro argument above and the value of the y-axis defined by the regression formula as:

INTERCEPT + GRP_SLP*(0 or 1 depending on treatment plot location) + CUT_SLP*X_MIN)

Then, annotate tells SAS the end point of the regression line, informing SAS to draw the line with the "function='draw'" command. Here, the x-axis value is equal to the value of CUTSCORE supplied by the user and the value along the y-axis is again equal to the regression formula as

INTERCEPT + GRP_SLP*(0 or 1 depending on treatment plot location) + CUT_SLP* CUTSCORE)

```
**generate annotated data set for graphing separate lines for each group;
**create separate files for the blue and red lines;
data ann_regdisb;
      set regdis;
      **the following 3 commands tell SAS to use absolute values along the axis;
      xsys='2'; ysys='2'; dsys='2';
      **create the line for the group to the left of the cut;
      **this tells SAS where to 'put the pen down' to begin drawing, based on x-y
      coordinates;
      if "&treat_loc"='below' and &group=1 then do;
            color='blue'; style='1'; size=2;
            function='move';
            x=&x_min;
            y=&intercept+(&grp_slp*1)+(&cut_slp*&x_min);
      end;
      if "&treat_loc"='above' and &group=0 then do;
            color='blue'; style='1'; size=2;
            function='move';
            x=&x_min;
            y=&intercept+(&grp_slp*0)+(&cut_slp*&x_min);
      end;
```

```
        output;
        **this tells SAS where to 'draw the line' based on x-y coordinates;
        if "&treat_loc"='below' and &group=1 then do;
               color='blue'; style='1'; size=2;
               function='draw';
               x=&cutscore;
               y=&intercept+(&grp_slp*1)+(&cut_slp*&cutscore);
        end;
        if "&treat_loc"='above' and &group=0 then do;
               color='blue'; style='1'; size=2;
               function='draw';
               x=&cutscore;
               y=&intercept+(&grp_slp*0)+(&cut_slp*&cutscore);
        end;
        output;
        run;
```

Next, the same process is followed in creating a second data file representing the group to the right of the cutscore. The code is not presented here, but is available in the appendix below as part of the full macro syntax. Once both data files are available, they are appended to one another in a data step. An example of the resulting annotated data set can be seen in Figure 1.

```
**add both files together by appending both;
data ann_regdis;
       set ann_regdisb ann_regdisr;
       where (x ne .);
       run;
```

	group	pretest	posttest	zgroup	interact	Control	Treatment	xsys	ysys	dsys	color	style	size	function	x	y
1	G1	6	15	1	6	.	15	2	2	2	blue	1	2	move	6	16.04775
2	G1	6	15	1	6	.	15	2	2	2	blue	1	2	draw	14.5	18.790889
3	G1	11	19	1	11	.	19	2	2	2	blue	1	2	move	6	16.04775
4	G1	11	19	1	11	.	19	2	2	2	blue	1	2	draw	14.5	18.790889
5	G1	11	17	1	11	.	17	2	2	2	blue	1	2	move	6	16.04775
6	G1	11	17	1	11	.	17	2	2	2	blue	1	2	draw	14.5	18.790889
7	G1	14	19	1	14	.	19	2	2	2	blue	1	2	move	6	16.04775
8	G1	14	19	1	14	.	19	2	2	2	blue	1	2	draw	14.5	18.790889
9	G2	15	11	0	0	11	.	2	2	2	red	1	2	move	14.5	11.373216
10	G2	15	11	0	0	11	.	2	2	2	red	1	2	draw	30	16.375412
11	G2	17	11	0	0	11	.	2	2	2	red	1	2	move	14.5	11.373216
12	G2	17	11	0	0	11	.	2	2	2	red	1	2	draw	30	16.375412
13	G2	20	15	0	0	15	.	2	2	2	red	1	2	move	14.5	11.373216
14	G2	20	15	0	0	15	.	2	2	2	red	1	2	draw	30	16.375412
15	G2	25	17	0	0	17	.	2	2	2	red	1	2	move	14.5	11.373216
16	G2	25	17	0	0	17	.	2	2	2	red	1	2	draw	30	16.375412
17	G2	30	14	0	0	14	.	2	2	2	red	1	2	move	14.5	11.373216
18	G2	30	14	0	0	14	.	2	2	2	red	1	2	draw	30	16.375412

Figure 1. Annotated data set containing instruction for SAS to 'draw' the regression lines.

PROC SQL is then used to create x-axis and y-axis minimum and maximum values equal to the respective pre and post-test minimum and maximum values minus and plus 1. This dynamically sets the axes at values appropriate to show the full regression line with a small space on either end.

```
**sql puts minimum/maximum of precut plus/minus 1 into macro variables;
proc sql noprint;
select round(min(x)-1,1), round(max(x)+1,1), round(min(y)-1,1), round(max(y)+1,1)
       into: xaxis_min, : xaxis_max, : yaxis_min, : yaxis_max
       from ann_regdis;
       quit;
```

If the user is plotting data points on the graph along with the lines, symbol statements are used to set the symbols equal to the same color as the group regression lines. If the user is not plotting the data, the symbols are set equal to white. AXIS and LEGEND statements are then used to establish the minimum and maximum values for the x and y axes, and to create a label for the legend present in the graph.

```
**here we establish symbol and axis values for the graph;
%if "&plotdata"="Y" %then %do;
symbol1 c=blue i=none v=circle h=1.5 w=3;
symbol2 c=red  i=none v=triangle h=1.5 w=3;
%end;
%if "&plotdata"="N" %then %do;
symbol1 c=white i=none v=circle h=1.5 w=3;
symbol2 c=white  i=none v=triangle h=1.5 w=3;
%end;
axis1 order=(&xaxis_min to &xaxis_max) label=("&pretest");
axis2 order=(&yaxis_min to &yaxis_max) label=(angle=90 "&posttest");
legend1 label=('Intervention Group') across=1 down=2;
```

PROC GPLOT, along with the overlay function, is utilized to generate the regression discontinuity plot. Using the HTML listing style to minimize formatting, the data in the input file is plotted according to the user supplied specifications for location of the treatment group and the location of the cutscore. The annotate function within PROC GPLOT informs SAS which data file to use to find the instructions for plotting.

```
**here we use the listing style to minimize the formatting of the plot;
ods html style=styles.listing;
**the gplot plots the values for the group to the left and right of the cut-score;
**the overlay function allows for separate plotting of two groups;
proc gplot data=regdis;
      %if "&treat_loc"="below" %then %do;
      plot (treatment control)*&pretest/overlay href=&cutscore vaxis=axis2
      haxis=axis1 annotate=ann_regdis legend=legend1;
      %end;
      %if "&treat_loc"="above" %then %do;
      plot (control treatment)*&pretest/overlay href=&cutscore vaxis=axis2
      haxis=axis1 annotate=ann_regdis legend=legend1;
      %end;
      run;
      quit;
ods html close;
%end;
```

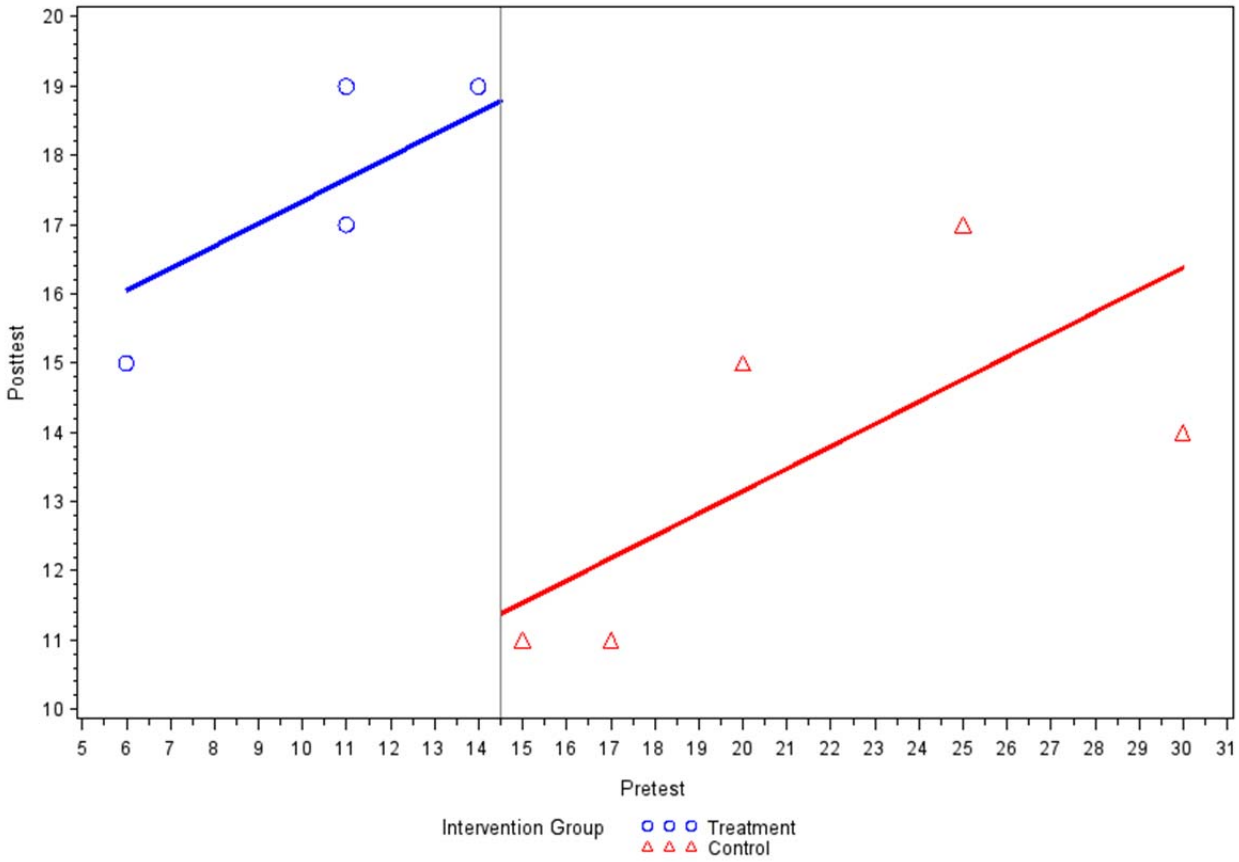

Figure 2. Regression discontinuity plot under the assumption of homogeneity.

In the instance where heterogeneity is detected in the first PROC GLM analysis, a separate set of code is run only when HETERO="Y". This code performs exactly the same procedure as when HETERO="N", except that a separate PROC GLM is used to generate individual regression parameters for the group above and below the cutscore. These individual parameters are used in the same manner to instruct SAS where to place the 'pen' and how to 'draw' the regression lines on either side of the cut-score. An extreme example of heterogeneity can be seen in the Figure 3.

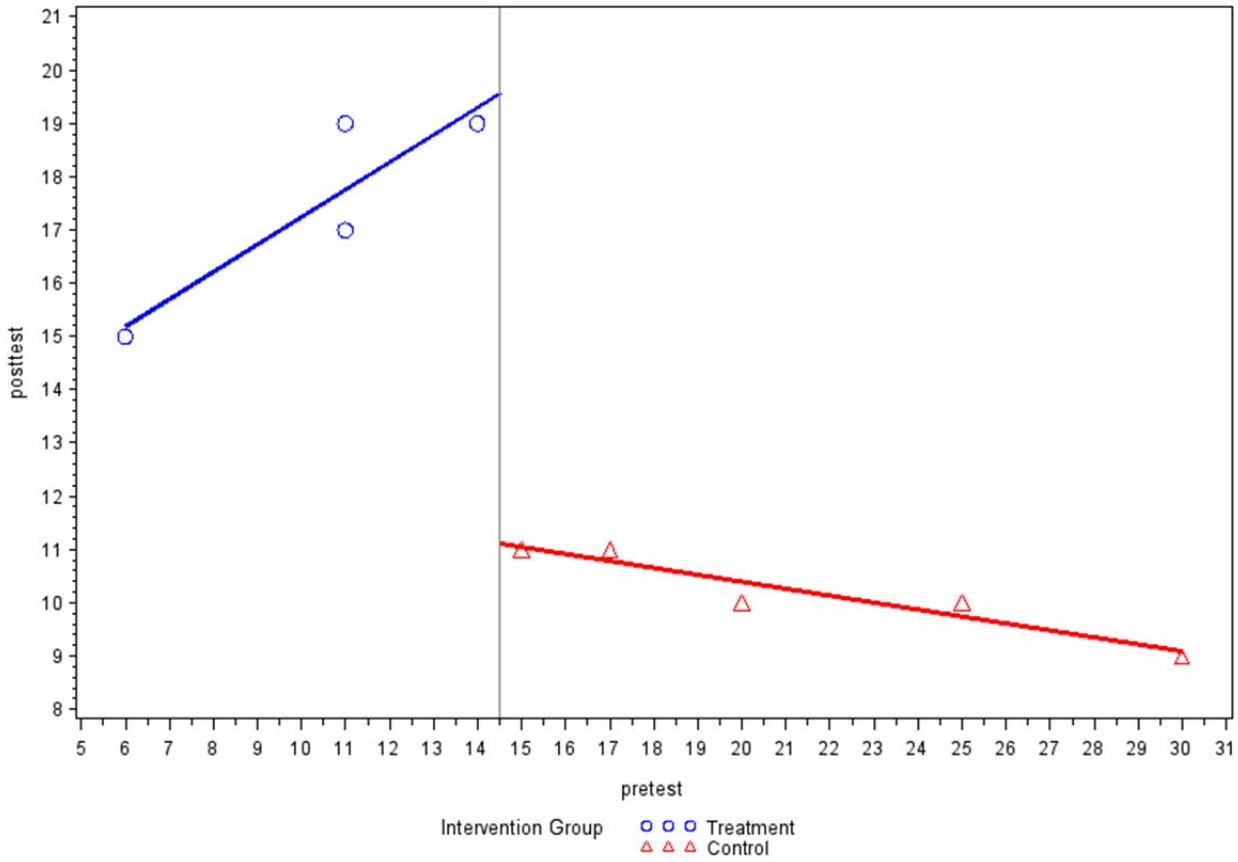

Figure 3. Regression discontinuity plot under the assumption of heterogeneity.

CONCLUSION

The regression discontinuity garners attention from applied social science and education researchers because of the tractability it gains with policy-makers by side-stepping the denial of a potentially useful treatment to those who may benefit. In randomized controlled trials, receipt of treatment is determined at random, whereas with the regression discontinuity design treatment is determined by some pre-treatment measure. Of course, this type of assignment does not lend itself to the exploration of all types of interventions, but does provide a viable alternative when groups can be assigned to treatment based on some measurable, identifiable need along the pre-test continuum.

Despite the design's usefulness from a practical standpoint, interpretation and understanding of regression parameter estimates is not easily achieved by all audiences. However, visual depiction of the treatment effect through a regression discontinuity plot provides an easy-to-understand method for quickly summarizing treatment results. Further, the plots allow the researchers themselves to visually inspect their data to uncover potential outliers or patterns worthy of further exploration. The RDPLOT macro is an attempt to provide applied researchers and SAS® users with an easy-to-use program to serve both of these purposes. Future improvements to the macro could include the incorporation of advancements in the regression discontinuity literature mentioned briefly in the introduction, or other enhancements to the graphical display of the data.

REFERENCES

Cook, T. D. (2008). 'Waiting for life to arrive': A history of the regression-discontinuity design in psychology, statistics and economics. *Journal of Economics, 142, 2,* 636-654.

Hahn, J., Todd, P., Van Der Klaauw, W., 2001. Identification and estimation of treatment effects with a regression discontinuity design. Econometrica 69, 201–209.

Imbens, G. (2008). Special issue editors' introduction: The regression discontinuity design-theory and applications. *Journal of Economics, 142, 2,* 611-614.

Lee, D.S. (2008). Randomized experiments from non-random selection in U.S. house elections. *Journal of Economics, 142, 2,* 675-697.

Ludwig, J., Miller, D., 2007. Does head start improve children's life chances? Evidence from a regression discontinuity design. Quarterly Journal of Economics 122 (1), 159–208.

McCrary, J., 2007. Testing for manipulation of the running variable in the regression discontinuity design. *Journal of Economics, 142, 2,* 698-714.

SAS Institute Inc. 2009. *SAS/GRAPH(R) 9.2: Statistical Graphics Procedures Guide*. Cary, NC: SAS Institute Inc.

Thistlethwaite, D.L., Campbell, D.T. (1960). Regression-discontinuity analysis: an alternative to the ex-post facto experiment. *Journal of Educational Psychology. 51*, 309–317.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Jason Schoeneberger
University of South Carolina
14726 Provence Lane
Charlotte, NC 28277
Phone: 704-307-9395
E-mail: schoeneb@email.sc.edu

```
/*
Please appropriately cite the macro when used:
Schoeneberger, J. A. (2011).
RDPLOT: A SAS® Macro for Generating Regression-
Discontinuity Plots. Proceedings of the annual Southeastern
SAS Users Group Conference.

As you use the macro, please feel free to send feedback,
comments, and/or suggestions to,
Jason A. Schoeneberger, at schoeneb@mailbox.sc.edu.
*/
Important information about the data structures compatible
with macro RDPLOT:

**assumes data_in file is one line per unit-of-analysis
record;
**assumes group is dichotomous, numeric variable coded
0=control, 1=treatment;


*path=folder location where data file for analysis resides;
*data_in=the data file for analysis;
*pretest=the pre-test score by which a cut-score being
applied creates treatment-control groups;
*posttest=the posttest, dependent variable of interest;
*group=variable denoting treatment (1) or control (0) group
membership;
*cutscore=numeric value on the pre-test that serves as the
cut-score;
*treat_loc=denotes whether the treatment is above or below
cut-score: insert 'above' or 'below' for this value;
*plot_data=denotes whether the user wants to plot actual
data 'Y' or just RD lines 'N';

%macro regdis
(path,data_in,pretest,posttest,group,cutscore,treat_loc,plo
tdata);

**create regdis data file, with pretest and posttest
supplied by user;
**also create centered pre-test score using user supplied
cut-score value;
**finally, create interaction variable for testing
heterogeneity;
data regdis;
      set "%bquote(&path)%bquote(&data_in)";
      **create interaction variable for testing homogeneity
      model;
      interact=&group*&pretest;
      **create separate posttest variables for each group;
      if (&group=0) then control=&posttest;
      if (&group=1) then treatment=&posttest;
      label control="Control" treatment="Treatment";
      run;

**this sql puts the minimum and maximum values of precut
into x_min and x_max macro variables;
proc sql noprint;
      select min(&pretest), max(&pretest), min(&posttest),
max(&posttest)
            into: x_min, : x_max, : y_min, : y_max
            from regdis;
      quit;

**the following glm tests the interaction (homogeneity
assumption);
ods output parameterestimates=paramsint;
proc glm data=regdis namelen=32;
      model &posttest=&group interact &pretest/ss1
solution;
      run;
      quit;

**grab parameter table and determine if interaction is
significant;
data paramsint;
      set paramsint;
      if (parameter="interact") and (probt gt .05) then
call symput('hetero','N');
      if (parameter="interact") and (probt le .05) then
call symput('hetero','Y');
      run;

**the following will run only when homogeneity is assumed
(hetero='N');
%if "&hetero"="N" %then %do;
**this glm generates the parameter estimates for generating
plots - assumes homogeneity;
ods output parameterestimates=params;
```

```
proc glm data=regdis namelen=32;
       model &posttest=&group &pretest/ss1 solution;
       run;
       quit;

**this sql places the glm parameter estimates into macro
variables for intercept and slopes;
proc sql noprint;
       select estimate into: intercept
              from params
              where parameter="Intercept";
       select estimate into: grp_slp
              from params
              where parameter=lowcase("&group");
       select estimate into: cut_slp
              from params
              where parameter=lowcase("&pretest");
       quit;

**generate annotated data set for graphing separate lines
for each group;
**create separate files for the blue and red lines;
data ann_regdisb;
       set regdis;
       **the following 3 commands tell SAS to use absolute
       values of axis for 'moving' and 'drawing';
       **http://support.sas.com/documentation/cdl/en/graphre
       f/63022/HTML/default/viewer.htm#annodata-creating-
       grelem.htm;
       xsys='2'; ysys='2'; dsys='2';
       **create the line for the group to the left of the
       cut;
       **this tells SAS where to 'put the pen down' to begin
       drawing, based on x-y coordinates;
       if "&treat_loc"='below' and &group=1 then do;
              color='blue'; style='1'; size=2;
              function='move';
              x=&x_min;
              y=&intercept+(&grp_slp*1)+(&cut_slp*&x_min);
       end;
       if "&treat_loc"='above' and &group=0 then do;
              color='blue'; style='1'; size=2;
              function='move';
              x=&x_min;
              y=&intercept+(&grp_slp*0)+(&cut_slp*&x_min);
       end;
       output;
       **this tells SAS where to 'draw the line' based on x-
       y coordinates;
       if "&treat_loc"='below' and &group=1 then do;
              color='blue'; style='1'; size=2;
              function='draw';
              x=&cutscore;

       y=&intercept+(&grp_slp*1)+(&cut_slp*&cutscore);
       end;
       if "&treat_loc"='above' and &group=0 then do;
              color='blue'; style='1'; size=2;
              function='draw';
              x=&cutscore;

       y=&intercept+(&grp_slp*0)+(&cut_slp*&cutscore);
       end;
       output;
       run;
data ann_regdisr;
       set regdis;
       xsys='2'; ysys='2'; dsys='2';
       **here begin the commands for creating the line for
       the group to the right of the cut;
       **this tells SAS where to 'put the pen down' to begin
       drawing, based on x-y coordinates;
       if "&treat_loc"='below' and &group=0 then do;
              color='red'; style='1'; size=2;
              function='move';
              x=&cutscore;

       y=&intercept+(&grp_slp*0)+(&cut_slp*&cutscore);
       end;
       if "&treat_loc"='above' and &group=1 then do;
              color='red'; style='1'; size=2;
              function='move';
              x=&cutscore;

       y=&intercept+(&grp_slp*1)+(&cut_slp*&cutscore);
       end;
       output;
       **this tells SAS where to 'draw the line' based on x-
       y coordinates;
       if "&treat_loc"='below' and &group=0 then do;
```

```
            color='red'; style='1'; size=2;
            function='draw';
            x=&x_max;
            y=&intercept+(&grp_slp*0)+(&cut_slp*&x_max);
      end;
      if "&treat_loc"='above' and &group=1 then do;
            color='red'; style='1'; size=2;
            function='draw';
            x=&x_max;
            y=&intercept+(&grp_slp*1)+(&cut_slp*&x_max);
      end;
      output;
      run;

**add both files together by appending both;
data ann_regdis;
      set ann_regdisb ann_regdisr;
      where (x ne .);
      run;

**this sql puts the minimum and maximum values of precut,
plus or minus 1 into x_min and x_max macro variables;
proc sql noprint;
      select round(min(x)-1,1), round(max(x)+1,1),
round(min(y)-1,1), round(max(y)+1,1)
            into: xaxis_min, : xaxis_max, : yaxis_min, :
yaxis_max
            from ann_regdis;
      quit;

**here we establish symbol and axis values for the graph;
%if "&plotdata"="Y" %then %do;
symbol1 c=blue i=none v=circle h=1.5 w=3;
symbol2 c=red  i=none v=triangle h=1.5 w=3;
%end;
%if "&plotdata"="N" %then %do;
symbol1 c=white i=none v=circle h=1.5 w=3;
symbol2 c=white  i=none v=triangle h=1.5 w=3;
%end;
axis1 order=(&xaxis_min to &xaxis_max) label=("&pretest");
axis2 order=(&yaxis_min to &yaxis_max) label=(angle=90
"&posttest");
legend1 label=('Intervention Group') across=1 down=2;
```

```
**here we use the listing style to minimize the formatting
of the plot;
ods html style=styles.listing;
**the gplot plots the separate values for the group to the
left and right of the cut-score;
**the overlay function allows for separate plotting of two
groups;
proc gplot data=regdis;
      %if "&treat_loc"="below" %then %do;
      plot (treatment control)*&pretest/overlay
href=&cutscore vaxis=axis2 haxis=axis1 annotate=ann_regdis
legend=legend1;
      %end;
      %if "&treat_loc"="above" %then %do;
      plot (control treatment)*&pretest/overlay
href=&cutscore vaxis=axis2 haxis=axis1 annotate=ann_regdis
legend=legend1;
      %end;
      run;
      quit;
ods html close;
%end;

**the following will run only when homogeneity is violated
(hetero='Y');
%if "&hetero"="Y" %then %do;

**these two glm statements will calculate separates
regression for each group, under heterogeneity;
ods output parameterestimates=paramsg1;
proc glm data=regdis;
      model &posttest=&pretest/ss1 solution;
      where (&group=1);
      run;
      quit;
ods output parameterestimates=paramsg0;
proc glm data=regdis;
      model &posttest=&pretest/ss1 solution;
      where (&group=0);
      run;
      quit;

**this sql places the glm parameter estimates into macro
variables for intercept and slopes;
proc sql noprint;
```

```
	select estimate into: intercept_g1
		from paramsg1
		where parameter="Intercept";
	select estimate into: cut_slp_g1
		from paramsg1
		where parameter=lowcase("&pretest");
	select estimate into: intercept_g0
		from paramsg0
		where parameter="Intercept";
	select estimate into: cut_slp_g0
		from paramsg0
		where parameter=lowcase("&pretest");
	quit;

**generate annotated data set for graphing separate lines
for each group;
data ann_regdisb;
	set regdis;
	**the following 3 commands tell SAS to use absolute
	values of axis for 'moving' and 'drawing';
	**http://support.sas.com/documentation/cdl/en/graphre
	f/63022/HTML/default/viewer.htm#annodata-creating-
	grelem.htm;
	xsys='2'; ysys='2'; dsys='2';
	**here begin the commands for creating the line for
	the group to the left of the cut;
	**this tells SAS where to 'put the pen down' to begin
	drawing, based on x-y coordinates;
	if "&treat_loc"='below' and &group=1 then do;
		color='blue'; style='1'; size=2;
		function='move';
		x=&x_min;
		y=&intercept_g1+(&cut_slp_g1*&x_min);
	end;
	if "&treat_loc"='above' and &group=0 then do;
		color='blue'; style='1'; size=2;
		function='move';
		x=&x_min;
		y=&intercept_g0+(&cut_slp_g0*&x_min);
	end;
	output;
	**this tells SAS where to 'draw the line' based on x-
	y coordinates;
	if "&treat_loc"='below' and &group=1 then do;
		color='blue'; style='1'; size=2;
		function='draw';
		x=&cutscore;
		y=&intercept_g1+(&cut_slp_g1*&cutscore);
	end;
	if "&treat_loc"='above' and &group=0 then do;
		color='blue'; style='1'; size=2;
		function='draw';
		x=&cutscore;
		y=&intercept_g0+(&cut_slp_g0*&cutscore);
	end;
	output;
	run;

data ann_regdisr;
	set regdis;
	xsys='2'; ysys='2'; dsys='2';
	**here begin the commands for creating the line for
	the group to the right of the cut;
	**this tells SAS where to 'put the pen down' to begin
	drawing, based on x-y coordinates;
	if "&treat_loc"='below' and &group=0 then do;
		color='red'; style='1'; size=2;
		function='move';
		x=&cutscore;
		y=&intercept_g0+(&cut_slp_g0*&cutscore);
	end;
	if "&treat_loc"='above' and &group=1 then do;
		color='red'; style='1'; size=2;
		function='move';
		x=&cutscore;
		y=&intercept_g1+(&cut_slp_g1*&cutscore);
	end;
	output;
	**this tells SAS where to 'draw the line' based on x-
	y coordinates;
	if "&treat_loc"='below' and &group=0 then do;
		color='red'; style='1'; size=2;
		function='draw';
		x=&x_max;
		y=&intercept_g0+(&cut_slp_g0*&x_max);
	end;
	if "&treat_loc"='above' and &group=1 then do;
		color='red'; style='1'; size=2;
		function='draw';
		x=&x_max;
```

```
            y=&intercept_g1+(&cut_slp_g1*&x_max);
      end;
      output;
      run;

data ann_regdis;
      set ann_regdisb ann_regdisr;
      where (x ne .);
      run;

**this sql puts the minimum and maximum values of precut
into x_min and x_max macro variables;
proc sql noprint;
      select round(min(x)-1,1), round(max(x)+1,1),
round(min(y)-1,1), round(max(y)+1,1)
            into: xaxis_min, : xaxis_max, : yaxis_min, :
yaxis_max
            from ann_regdis;
      quit;

**here we establish symbol and axis values for the graph;
%if "&plotdata"="Y" %then %do;
symbol1 c=blue i=none v=circle h=1.5 w=3;
symbol2 c=red  i=none v=triangle h=1.5 w=3;
%end;
%if "&plotdata"="N" %then %do;
symbol1 c=white i=none v=circle h=1.5 w=3;
symbol2 c=white  i=none v=triangle h=1.5 w=3;
%end;
axis1 order=(&xaxis_min to &xaxis_max) label=("&pretest");
axis2 order=(&yaxis_min to &yaxis_max) label=(angle=90
"&posttest");
legend1 label=('Intervention Group') across=1 down=2;

**here we use the listing style to minimize the formatting
of the plot;
ods html style=styles.listing;
**the gplot plots the separate values for the group to the
left and right of the cut-score;
**the overlay function allows for separate plotting of two
groups;
proc gplot data=regdis;
      %if "&treat_loc"="below" %then %do;
      plot (treatment control)*&pretest/overlay
href=&cutscore vaxis=axis2 haxis=axis1 annotate=ann_regdis
legend=legend1;
      %end;
      %if "&treat_loc"="above" %then %do;
      plot (control treatment)*&pretest/overlay
href=&cutscore vaxis=axis2 haxis=axis1 annotate=ann_regdis
legend=legend1;
      %end;
      run;
      quit;
ods html close;
%end;

%mend regdis;
```