Edps/Psych/Stat 587

C.J. Anderson

SAS: Computer Lab 4

EDA of Fixed Effects Structure with a Little Regression Diagnostics

The following is what was done in lab and the figures that you should have gotten. There are additional figures that you need to create by modifying the code given in this document.

1. Run the program ``TIMSS\_lab4\_data.sas''. This will create the following data sets which we'll use today:

**timss** which contains timss data we've been using all semester.

**clab4** is the same as timss but it also includes the school mean math scores *grpMmath* and the school mean centered math scores *grpCmath*.

1. For some graphs we'll only want a random sample of schools. To take a sample, we can use PROC **SURVEYSELECT**. The following SAS code gets a simple random sample of 32 schools.

title ’A Sample of **32** schools that were randomly selected’;

**proc** **surveyselect** data=sampsize method=srs n=**32** out=selected;

**run**;

**data** selected;

set selected;

selectedschool=idschool;

**proc** **sort** data=selected;

by idschool;

**data** sample;

merge clab4 selected;

by idschool;

if selectedschool=**.** then delete;

**run**;

1. We'll be mostly using ``sg'' for graphics, which stands for “statistical graphics”. The default file format for sg graphics is png; however, if you prefer another one (e.g., jpeg, ps, or any other options available), you can to specific the file format by the following code

ODS GRAPHICS / IMAGEFMT =JPG;

ODS LISTING GPATH=**"C:\ …put where you want figures to go… ";**

* This should be placed before the code that generates the graphic and needs to be used only once.

1. For this set of graphs, use the random sample of 32 schools. For all cases listed below, you should produce two plots: one contains graphs for the first 16 schools on a single page and the other contains the graphs for the next 16 schools on a single page.

Basic commands that will produce the panel graphs of data for each macro unit:

Title 'Science by Math: Linear regression';

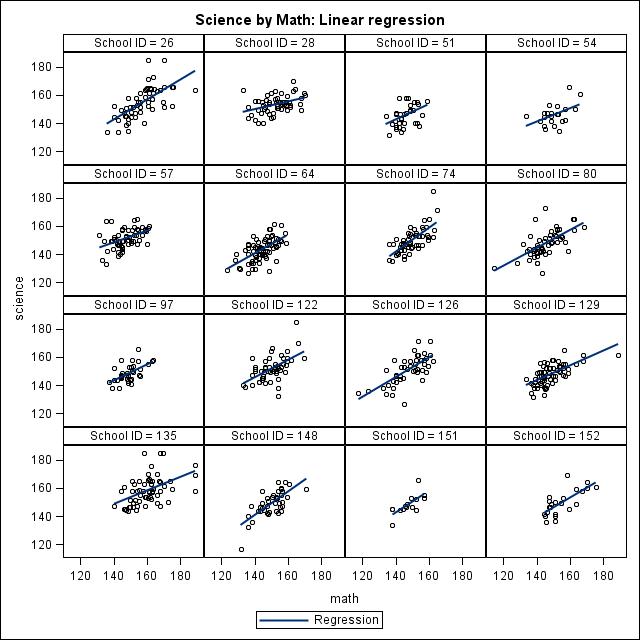
**proc** **sgpanel** data=sample;

panelby idschool / columns=**4** rows=**4**;

reg x=math y=science;

**run**;

Example of one of them:

****

1. Plot **science** scores by **math** scores where data are points and lines are
2. Linear regression for each school:

“reg x=math y=science;”

1. Cubic regression for each school:

“loess x=math y=science/ interpolation=cubic;”

1. Spline functions for each school:

“pbspline x=math y=science;”

1. Plot **science** scores by **hours per week watching TV** for each school where the data are points and lines are
   1. Linear regression for each school
   2. Cubic regression for each school
   3. Spline for each school
2. Plot **science** scores by **hours per week playing computer games** for each school where the data are points and lines are
   1. Linear regression for each school
   2. Cubic regression for each school
   3. Spline for each school
3. Plot the linear regression lines of **science** scores as a function of math **scores** for each school in the same figure (use the full data set: clab4)

\* Linear regressions all in one (full data set);

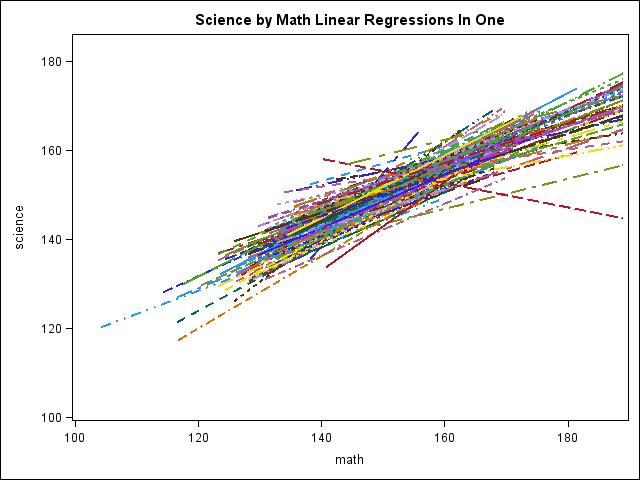
title 'Science by Math Linear Regressions In One';

**proc** **sgplot** data=clab4;

reg x=math y=science / group=idschool MARKERATTRS=(size=**.00**) ;

keylegend 'none' ;

**run**;



1. Plot the **overall mean science scores** by the number of **hours watching TV** for each school where points are joined by a line.

**proc** **sort** data=clab4;

by hours\_TV;

**proc** **means** data=clab4 noprint;

by hours\_TV;

var science;

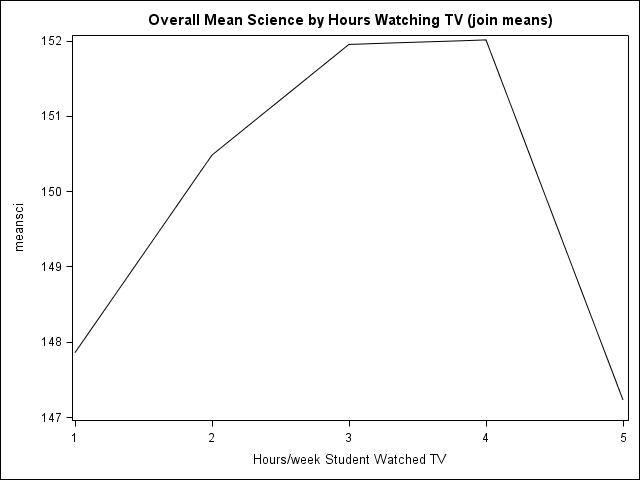
output out=meanfile1 mean=meansci;

title 'Overall Mean Science by Hours Watching TV (join means)';

**proc** **sgplot** data=meanfile1;

series x=hours\_TV y=meansci;

**run**;



1. Plot the **overall mean science** scores by **the number of hours playing computer games** joining the points.

Modify code above**.**

1. Plot the **mean science** scores by the **mean number of hours watching TV** for each school in a single plot. The following is the basic commend for joining points

**proc** **sort** data=clab4;

by idschool hours\_TV;

**proc** **means** data=clab4 noprint;

by idschool hours\_TV;

var science;

output out=meanfile2 mean=meansci;

title 'School Mean Science by Hours Watching TV (join)';

**proc** **sgplot** data=meanfile2;

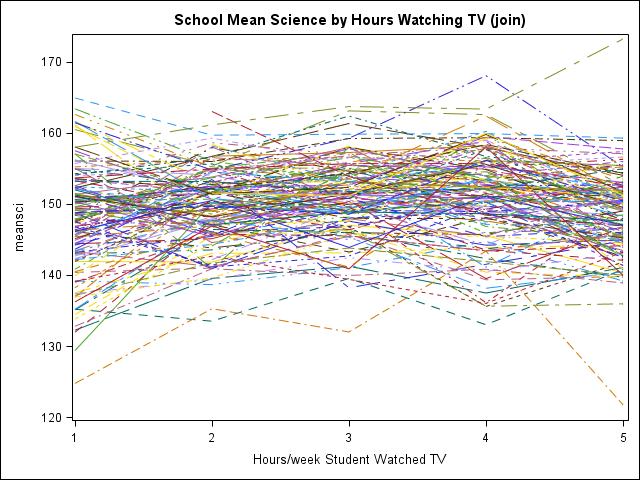
series x=hours\_TV y=meansci / group=idschool MARKERATTRS=(size=**.00**) ;

keylegend 'none' ;

**run**;

Generate the following figures:

1. Join points for each school (code above).



1. Linear regression for each school: replace “series” by “reg”.
2. Cubic regression for each school: replace “series” by “loess” and add to the options “interpolation=cubic”.
3. Plot the **mean science** scores by the **mean number of playing computer games** for each school in a single plot. The following is the basic commend for joining points for each of the following:
4. Join points for each school (code above).
5. Linear regression for each school: replace “series” by “reg”.
6. Cubic regression for each school: replace “series” by “loess” and add to the options “interpolation=cubic”.
7. Plot the **overall mean science** scores by the **number of hours playing computer gam**es joining the points.

Modify code above**.**

1. Plot the **mean science** scores by the **mean grouped math scores for each type of community** in a single plot.

Data steps

/\* This code will create a new variable that groups the math scores into approximately equal groups based on quantiles \*/

/\* Produce quantiles for math scores \*/

**proc** **rank** data=clab4 groups=**10**

out=quint(keep=idstudent gmath);

var math; ranks gmath ;

**run**;

**proc** **sort** data=clab4;

by idstudent;

**proc** **sort** data=quint;

by idstudent;

**run**;

/\* Merge the data set containing the quantiles onto the main data set \*/

**data** glab4;

merge clab4 quint;

by idstudent;

/\* For a more meaningful legend of a graph... \*/

if type\_community= **1** then community="Isolate";

else if type\_community= **2** then community="Rural";

else if type\_community= **3** then community="Suburban";

else if type\_community= **4** then community="Urban";

**run**;

Prepare data to graph and graphing it:

**Proc** **sort** data=glab4;

by community gmath;

**proc** **means** data=glab4 noprint;

by community gmath;

var science math;

output out=meanfile3 mean=meansci meanmath;

**run**;

title 'Mean Science per math group by Mean math for Each community';

**proc** **sgplot** data=meanfile3;

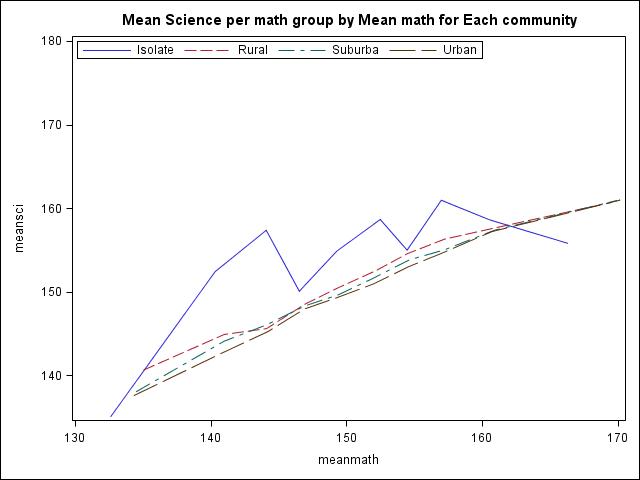
series y=meansci x=meanmath / group=community;

yaxis min= **135** max=**180**;

keylegend / location=inside position=topleft ;

**run**;

1. Join points for each community (i.e., code above)



1. Use linear regression to join means. (i.e., replace ``series'' by ``reg'')
2. Linear regression of science on math for each community (i.e., regression to the ungrouped data… may want to add the options markerattr$=$(size$=.00$) ).
3. Compute the number of observations for each point (mean) in the above figures

**proc** **freq** data=glab4;

tables gmath;

**run**;

1. Compute the values and plot them versus .

Put sample sizes in a file

\* Preliminaries;

**proc** **sort** data=clab4;

by idschool;

\* Get sample sizes for schools;

**proc** **means** data=clab4 noprint;

by idschool;

var science;

output out=nj n=nj;

**run**;

Fit model to each school’s data and put some results in files. This may take some time. After running the code below, use explorer to take a look at the files *ANOVA1* and *Fistat1*.

\* Regressions for each school and output Rsq, SS and MS to data sets;

**ods graphics off; \*<- this is important for speed;**

title 'Regressions for each school';

**proc** **reg** data=clab4 ;

by idschool;

model science = math hours\_TV hours\_computer\_games ;

ods output ANOVA=ANOVA1 FitStatistics=Fitstat1;

**run**;

ods graphics on; \*<- turn graphics back on;

Compute

**proc** **sort** data=anova1;

by source;

**proc** **means** data=anova1 sum;

by source;

var SS;

output out=sss sum=SS;

**run**;

\* Merge in sample sizes and only keep Rsq from Fitstat;

**data** Rsq1data;

merge nj Fitstat1;

by idschool;

R2meta1= **213754.47**/ **546140.57**; \* taken from means output;

Rsq1 = input(cValue2,**8.5**); \* cValue2 is character,

this makes it numeric;

if Label2='R-Square' then output;

keep nj Rsq1 R2meta1;

**run**;

To plot the data …another graphics procedure

\* Now we can plot it all;

goptions reset=(axis, legend, pattern, symbol, title, footnote) norotate

hpos=**0** vpos=**0** htext=**2** ftext=swiss ctext= target= gaccess= gsfmode= ;

goptions device=WIN ctext=blue graphrc interpol= colors= csymbol=;

symbol1 value=dot interpol=none color=blue;

symbol2 value=none interpol=rl line=**1** color=red;

axis1 color=black width=**2.0** label=(height=**2** "Group Sample Size");

axis2 color=black width=**2.0** label=(height=**2** angle=**90** "R^2 for each group")

order=**0** to **1** by **.1**;;

legend1 position=(top inside right) frame value=('Rsq' 'R2meta=.39')

label=none;

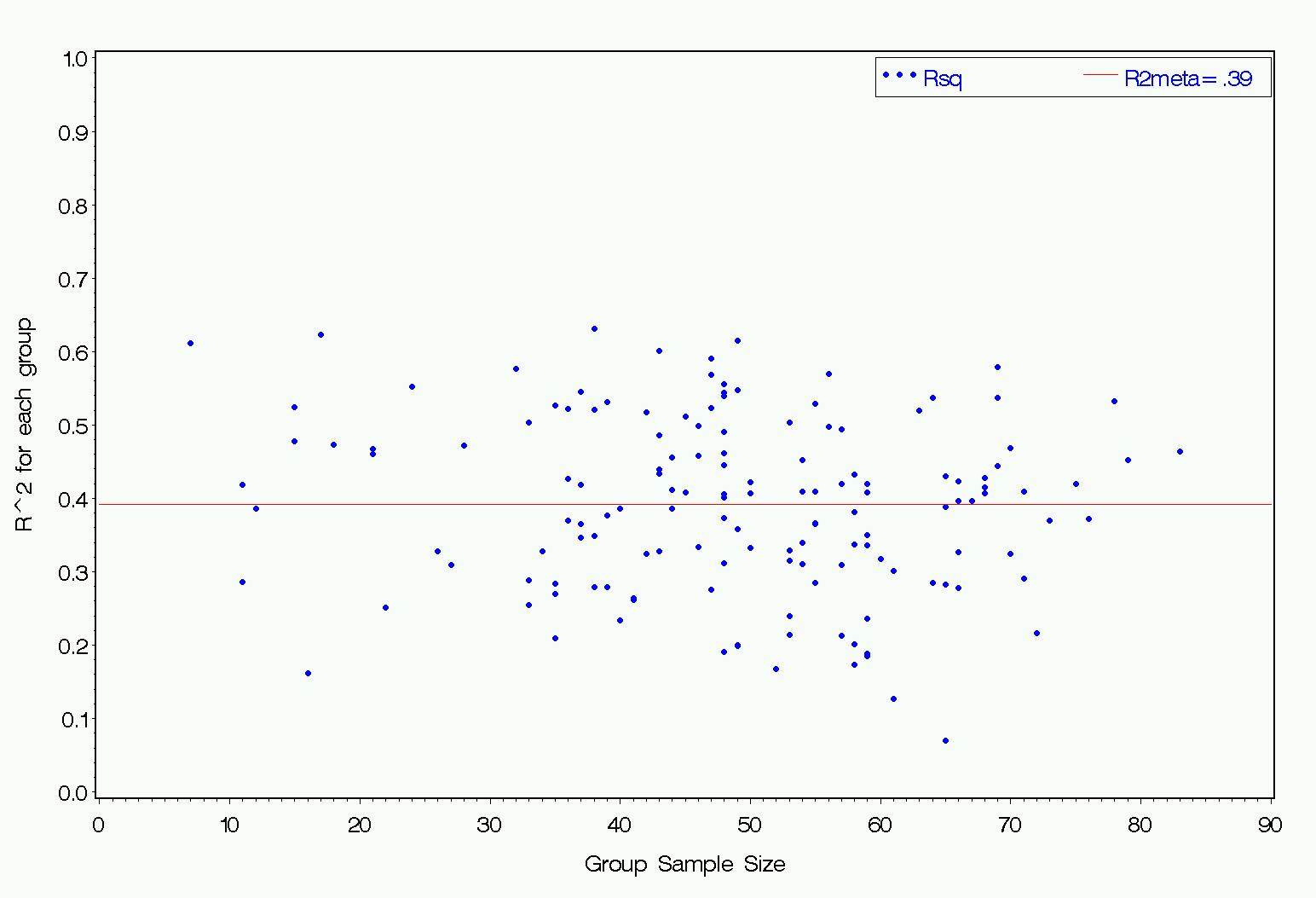
**proc** **gplot** data=Rsq1data;

plot Rsq1\*nj R2meta1\*nj / overlay haxis=axis1 vaxis=axis2 legend=legend1;

**run**;

Compute and plot and for the following two models:

1. science = math hours\_TV hours\_computer\_games, where hours playing computer games and watching TV are treated as numerical variables. (code above)



1. science = math hours\_TV hours\_computer\_games, where hours playing computer games and watching TV are treated as discrete (nominal) variables.

For the second model (i.e. model b where variable treated nominally), you’ll need to creat dummy codes because PROC REG doesn’t have a CLASS command.

**data** lab4coded;

set clab4;

\* Initialized dummy codes;

tv1=**0**; tv2=**0**; tv3=**0**; tv4=**0**;

games1=**0**; games2=**0**; games3=**0**; games4=**0**;

\* Set the codes;

if hours\_TV=**1** then tv1=**1**;

else if hours\_TV=**2** then tv2=**1**;

else if hours\_TV=**3** then tv3=**1**;

else if hours\_TV=**4** then tv4=**1**;

if hours\_computer\_games=**1** then games1=**1**;

else if hours\_computer\_games=**2** then games2=**1**;

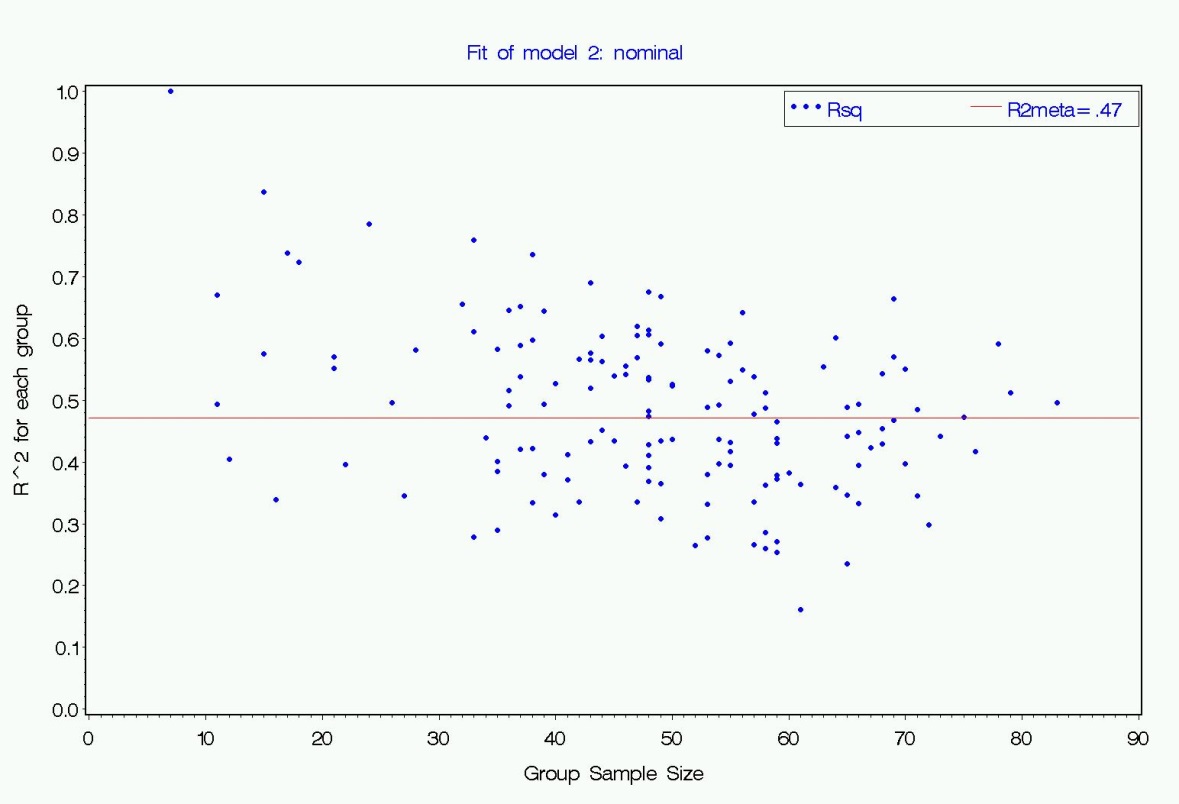
else if hours\_computer\_games=**3** then games3=**1**;

else if hours\_computer\_games=**4** then games4=**1**;

**run**;

A**ND….**

Change the name of Rsq1 to Rsq2 (“1” for model 1, and “2” for model 2). This will come in handy for the next part, where you'll need different names for from each of the models.



1. Plot the values from the two models against each other for an alternative way to look at whether there is improvement.

**data** both;

merge Rsq1data Rsq2data;

**run**;

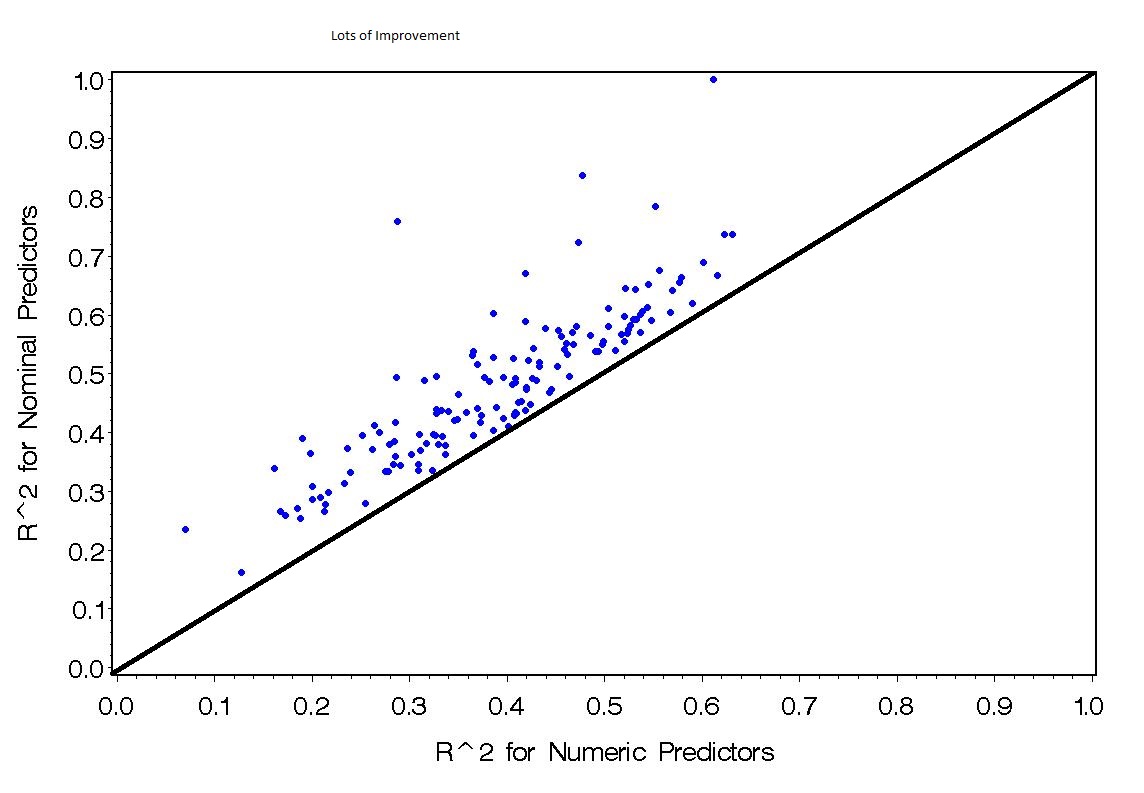
axis3 color=black width=**2.0** label=("R^2 for Numeric Predictors") order=**0** to **1** by **.1**;

axis4 color=black width=**2.0** label=(angle=**90** "R^2 for Nominal Predictors") order=**0** to **1** by **.1**;

**proc** **gplot** data=both;

plot Rsq2\*Rsq1 / overlay haxis=axis3 vaxis=axis4 ;

**run**;



You can use the paint program to draw in identity line.

Use your “final” or favorite model from Homework/Computer Lab 3 and get graphics of residuals (either studentized or Pearson for each type of residual) and influence diagnostics.

The code for my favorite one…. (note the code that is in red was added)….

\*re-code hours playing computer games;

**data** clab4recode;

set clab4;

if type\_community=**1** or type\_community=**2** then community='isol/rural';

else community='sub/urban';

if hours\_TV < **5** then TV=**0**;

else TV=**1**;

if hours\_computer\_games <**5** then games=**0**;

else games=**1**;

**run**;

ods graphics on / imagefmt=jpg;

title “My favorite one”;

**proc** **mixed** data=clab4recode covtest method=ML noclprint ic

plots(maxpoints=**8000**);

class idschool grade gender community TV games ;

model science = grpCmath gender grade TV games grpMmath community

grpCmath\*grpMmath

/ solution ddfm=satterth residual influence;

random intercept grpCmath / subject=idschool type=un g solution;

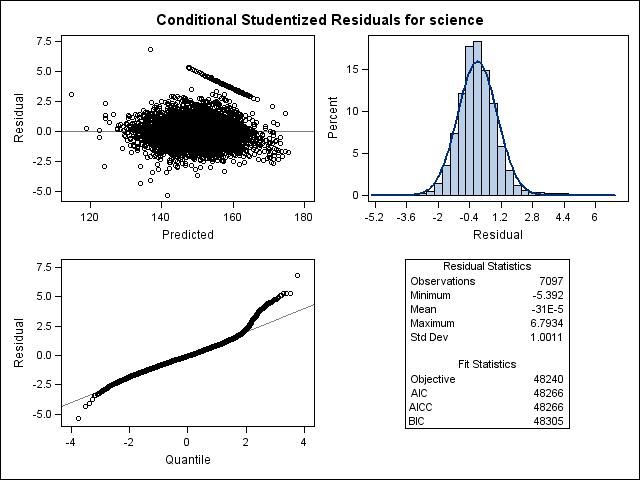
ods output CovParms=cov G=gmat ModelInfo=mod SolutionF=solf

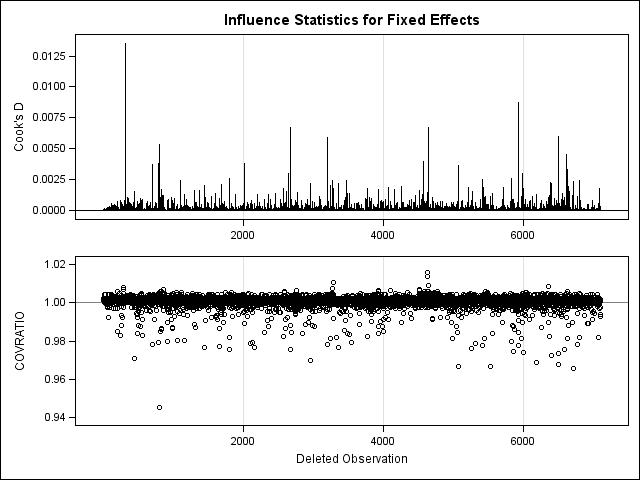
Influence=inf SolutionR=solr;

**run**; / imagefmt=jpg;

ods graphics off;

Various figures of regression diagnostics such as





1. Obtain estimates of random effects. Plot them to allow you to assess whether they appear normally distributed or not.

If you ran the code with ``solution'' in the RANDOM statement and the ``ods output'' statement with ``SolutionR=solr'', then you have a file ``solr'' that contains the estimates 's. The follow code will produce histograms of the estimated Us with normal distribution overlaid in the figure.

**proc** **sort** data=solr;

by Effect;

**run**;

title 'Density of EBs of Random Us';

**proc** **sgplot** data=solr;

by Effect;

histogram estimate;

density estimate; ;

keylegend / location=inside position=topright;

**run**;

To produce a scatter plot of versus , this should do the trick….

\* Some data manipulation;

**data** U0data U1data;

set solr;

if Effect='Intercept' then do;

U0=estimate;

keep IDSchool U0;

output U0data;

end;

else do;

U1=estimate;

keep IDschool U1;

output U1data;

end;

**run**;

**data** U0data;

set U0data;

**data** U1data;

set U1data;

drop U0;

**run**;

**data** RandomEffects;

merge U0data U1data;

by idschool;

**run**;

\* The actual Plotting of data;

title 'Bi-Variate Distribution of Random Effects';

**proc** **sgscatter** data=RandomEffects;

compare x=U0 y=U1;

**run**;

