Paper 253-2009

Interactive Outlier Review and Regression Analysis in Stat® Studio

Robert Seffrin, USDA-NASS, Fairfax, VA

ABSTRACT

The United States Department of Agriculture National Agricultural Statistics Service uses survey data and satellite imagery to create an annual Cropland Data Layer (CDL) land cover classification product. The CDL and survey data are used to estimate crop acreage. This case study greatly enhances the ActionMenuScatterPlot.sx SAS® Stat Studio program by:

- dynamically generating a menu based on the input data set values
- color-coding data values based on outlier indications from PROC REG
- using the menu selection to subset the viewed data
- saving selected data points to a SAS® data set
- rerunning the regression
- updating the display to highlight already saved data values
- saving the most recent regression and scatter plot to a file

Additionally, the program can display and update the regression equation and R2, brush data points by selected variables, and review the mapped spatial distribution of data values and errors. The program saves considerable time and effort in the data review process, and improves the quality of the acreage estimates.

INTRODUCTION

The National Agricultural Statistics Service (NASS) conducts hundreds of surveys every year covering the breadth of US agriculture from aquaculture and horticulture to the more traditional crop and livestock farms. Most of these surveys are sampled from stratified lists of farm and ranch operators. An exception to this is the June Area Survey (JAS) which is a sampling of land units. To create this sample, all of the land area in the United States is stratified into Primary Sampling Units (PSU) based on the estimated percent of cultivation or other land cover within its boundary. Each PSU has the potential to be sampled and broken into sampling units call segments. In crop intensive strata, a segment is generally one square mile. All strata are sampled and are accounted for by about 11,000 segments nationally which are enumerated with field visits with a June 1 reference date. Crop acreages are collected at the field level for land within the segments. Acreage is then summed to the segment level and expanded based on their sample rate to the stratum and state levels to provide an estimate of the crop acreage (Graham, 1993; Bellow, 1994). The JAS also provides a measure of list frame incompleteness by comparing operators found in the segments to those on the list frame.

Satellite imagery offers an alternative acreage estimator for large area crops like corn and soybeans providing complete coverage for a state much like a census. The input requirements are very different: cloud-free imagery (multi-date preferred), ground truth for training, and specialized software for classification of raw data into land cover types. The result is the Cropland Data Layer (CDL) which is a raster file where each pixel is assigned to a ground cover type. Also like a census, errors of omission and commission can create bias for a particular cover type if crops were estimated with just a simple pixel count of the CDL raster product. To correct for this bias a regression estimator is used to regress the acreage of a crop from the JAS segment against the pixel count of that crop from the CDL within the same segment boundaries (Day, 2002). This is done for all segments within an area defined as an analysis district (generally an entire state.)

The CDL program has covered a limited number of states since 1997 (NASS,2007). Until 2006, crop acreage estimation was done using in-house software called Peditor. Peditor was based on Pascal and FORTRAN and was used for all steps in the CDL creation and estimation process. In 2006 NASS began a modernization effort testing commercial decision tree classifier software to replace the Peditor maximum likelihood classifier and SAS to replace the Peditor estimation system. NASS decided to transition to SAS for estimation since it widely used within the agency and the IML Workshop had potential for interactive data review (Mueller, 2006). The programs described in this paper are a result of reproducing the Peditor procedures and results but in a more interactive and user friendly interface.

GETTING STARTED

The use of Stat Studio is one part of a series of data processing tasks needed to create the crop estimates. Since this processing is done by non-SAS trained analysts then SAS is hidden whenever possible. The program is launched by clicking on a desktop icon which launches the AF/SCL program seen in figure 1.

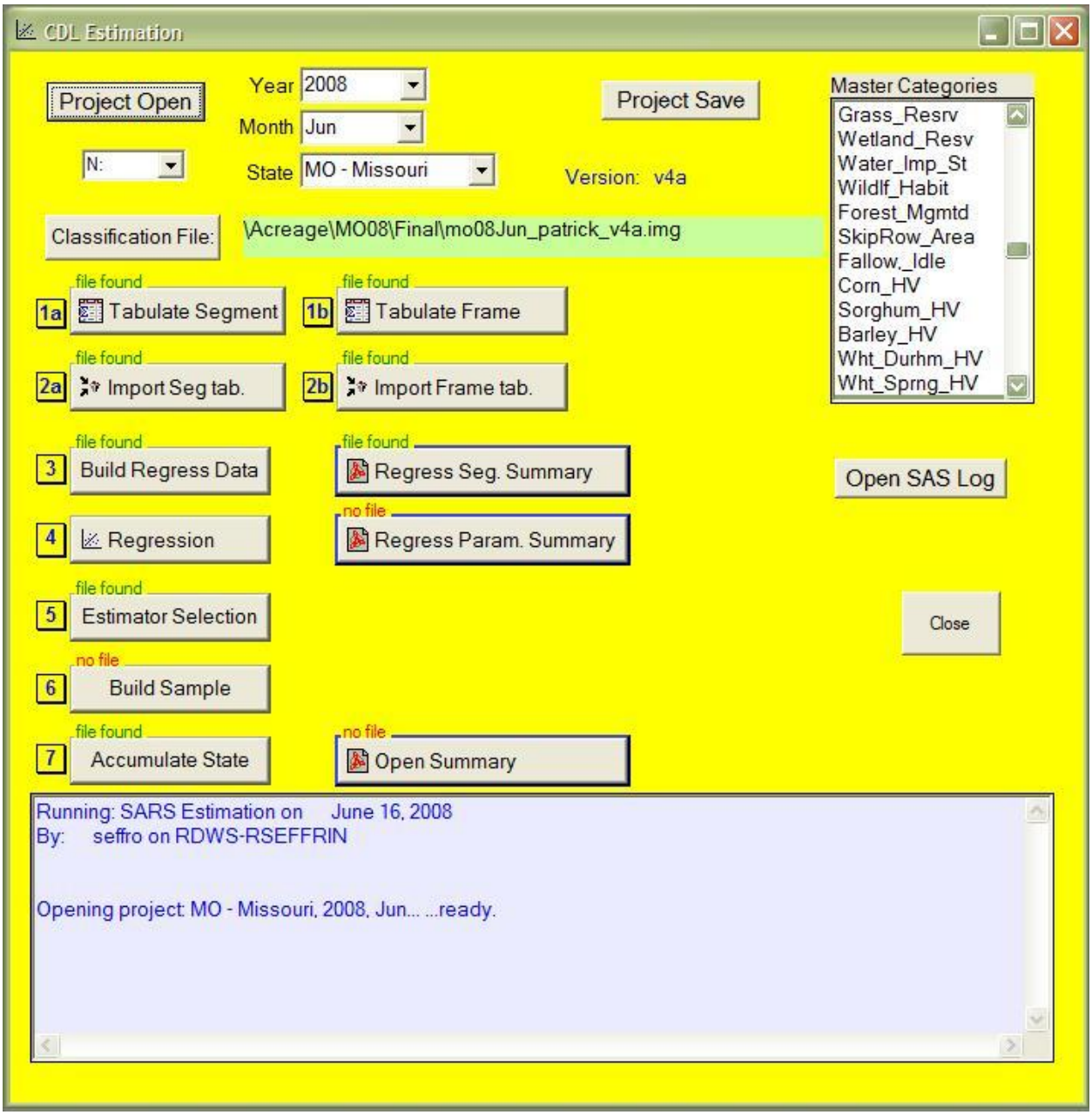

Figure 1. The interface from which the regression analysis is launched.

Estimates are run one state at a time. A state project is created by selecting a year, month, state, master categories of crop type, and a classification (CDL) file. These parameters are saved to a SAS catalog as an SCL list for each unique year, month, state, and version. A project can be reopened by clicking the "Project Open" button and selecting an SCL list. The JAS data is prepared beforehand with other SAS programs to summarize from the field level to the segment level and match with other data sets. The CDL derived data is created by running the "Tabulate Segment" and "Tabulate Frame" buttons which run an external program Erdas Imagine. Imagine is a raster processing program used to tabulate the pixels by cover type within each segment and across the entire state by PSU. This generates an ASCII matrix to be read in by the "Import Seg tab." and "Import Frame tab." buttons. The "Build Regress Data" button merges the JAS with the CDL segment level data. The regression review may now be launched by clicking the "Regression" button.

The SAS supplied program ActionMenuScatterPlot.iml/sx is a good starting point for this project as it creates a scatter plot and provides linear, quadratic, and cubic regression choices through the action menu. Several levels of prediction ellipses can overlay the data. Results of PROC REG are returned to the output window, the regression line plotted along with the 95% prediction limit and 95% confidence limit. The new program extends ActionMenuScatterPlot considerably and is developed in IML Workshop and tested in Stat Studio.

The transition from IML Workshop to Stat Studio is very simple; just point to statdudio.exe instead of imlWorkshop.exe and rename the program from *.iml to *.sx. The code below is also passing a list of parameters by name using the "-d' command line option. The SAS options of xwait and xsync are turned off before launching to keep the main interface available while doing the regression review.

```
pb_Launch_Reg:
/*- If on SARSbatch2, use StatStudio */
    IF ComputerName = 'SARSBATCH2' THEN DO;
      Program = '"C:\Program Files\SAS\StatStudio\3.1\System\statstudio.exe"';
      inProg  = "'||DriveRoot||'Estimates\...\ActionMenuScatterPlot_5.sx"';
    END;
    ELSE DO;
      Program = '"C:\Program Files\SAS\IML Workshop 2.1\System\IMLWorkshop.exe"';
      inProg  = '"'||DriveRoot||'\Estimates\...\ActionMenuScatterPlot_5.iml"';
    END;
    rc = OPTSETN('XWAIT', 0);
    rc = OPTSETN('XSYNC', 0);
    Parameters =  ' -d State='  || StatePost ||' -d Year='   || Year2
                ||' -d Version='|| Version   ||' -d LibDir=' || PathFinal
                ||' -d inFile=' ||'Regression_Build_'|| Version ;
    ProgLaunch = Program||inProg||Parameters;
    rc = SYSTEM ( ProgLaunch );
    rc = OPTSETN('XWAIT', 1);
    rc = OPTSETN('XSYNC', 1);
RETURN;
```

When Stat Studio launches the program window will be in the upper left hand corner. Press F5 or click on the blue arrow to run the program. Three more windows open, a scatter plot, a table of the data, and an output window as seen in figure 2. The program is designed to create a regression analysis for each unique state, analysis district, crop, and stratum. Currently only one state is reviewed at a time in its entirety so there is only one analysis district. The regressions are run from the action menu and have submenus for crops and stratum as seen in figure 3.

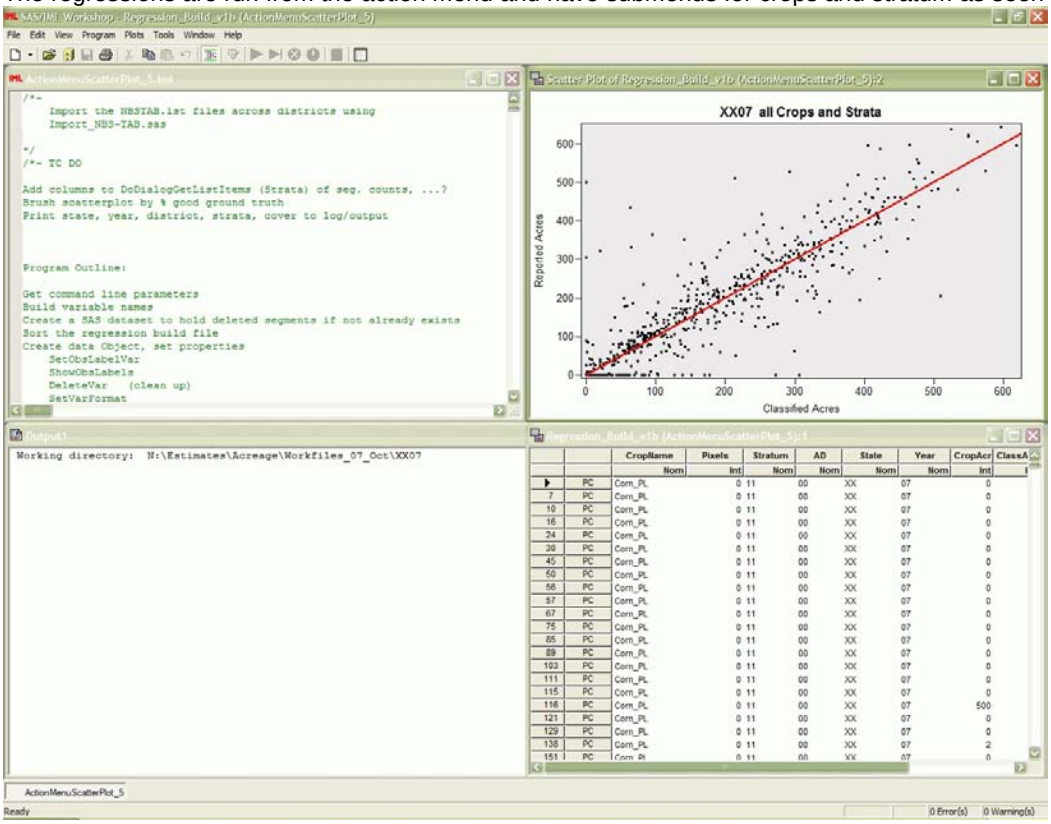

Figure 2. Initial view after starting IML program, program, plot, output, and table object windows.

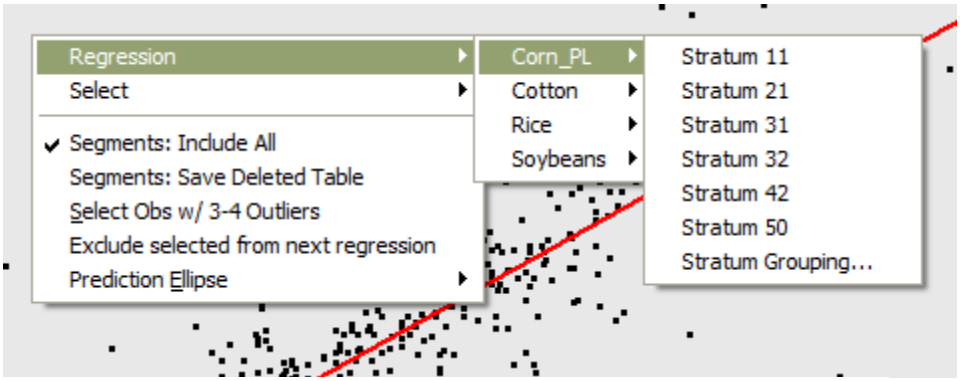

Figure 3. The action menu in action after pressing F11 with the plot window active.

GENERATING THE ACTION MENU

The action menu for regression is based on the number of unique values of four variables in the data object table; state, analysis district, crop, and stratum. This IML code builds strings are used to define the action menu:

```
/*- Some lines for state and analysis district left out */
uCrop    = UNIQUE( allCrop );
uStrat   = UNIQUE( allStrat );
cntCrop  = NCOL( uCrop ) ;
cntStrat = NCOL( uStrat );
TotMenuItems = cntState*cntDist*cntCrop*cntStrat;
Counts   = cntState || cntDist || cntCrop || cntStrat;
Permies  = J( TotMenuItems, NCOL(Counts), 0 );
Row      = 0;

/*- Build indexes of possible combinations of state, district, crop, stratum */
DO i1 = 1 TO  Counts[1];
  DO i2 = 1 TO  Counts[2];
    DO i3 = 1 TO  Counts[3];
      DO i4 = 1 TO  Counts[4];
        Row =Row + 1;
        IF Counts[1] = 1 THEN Permies[Row,1]=0; ELSE Permies[Row,1]=i1;
        IF Counts[2] = 1 THEN Permies[Row,2]=0; ELSE Permies[Row,2]=i2;
        IF Counts[3] = 1 THEN Permies[Row,3]=0; ELSE Permies[Row,3]=i3;
        IF Counts[4] = 1 THEN Permies[Row,4]=0; ELSE Permies[Row,4]=i4;
      END;
    END;
  END;
END;

/*- Build the strings to define the F11 cascading context menus  */
/*- Define the root of menu, create arrays to hold results */
sMenuBase_Reg ="Regression\n"J;
sCode_Reg='RUN OnLSRegression; ';
sCode_Group   ='Group...';

LengthFiller = "                                                                ";
aMenuStr     = J( TotMenuItems, 1, sMenuBase_Reg + LengthFiller );

/*- Build the strings based on the permutations */
/*- Create text like: "Regression\nState AR\nDistrict 01\nCorn\nStrata 11"J */
DO i = 1 TO TotMenuItems;
  IF cntState > 1 THEN DO;
    aMenuStr[i]= STRIP( aMenuStr[i] ) + "State " + uState[Permies[i,1]] + "\n"J;
  END;
  ELSE DO;
    aMenuCodeSel[i] = STRIP( aMenuCodeSel[i] ) + "'" + uState[1] + "'," ;
  END;
```

…
Additional conditional statements also run for district, crop, and stratum
…

```
    END;
```

This code uses the strings generated above to add the menu items to the action menu:

```
  plot.AppendActionMenuItem( aMenuStr[1], aMenuCodeSel[1] + aMenuCodeReg[1] );
/*- Append rest of menus to roots */
  DO i = 2 TO TotMenuItems;
    plot.AppendActionMenuItemToGroup( aMenuStr[1],aMenuStr[i]
       ,aMenuCodeSel[i]+Code_Reg );
  END;
```

RUN THE REGRESSION

The action menu is activated by selecting the plot window and selecting the F11 key. Choosing a crop and then a stratum runs the regression module which `SUBMITs` code to SAS to run PROC REG. The SAS calculated variables *hat, rstudent, dffits, and covratio* are saved for influence diagnostics. These are described in the PROC REG detail documentation on influence diagnostics. Any variables in the submit statement are available to SAS as macro variables. ODS output tables are saved to build a single results data set using PROC SQL at the end of the `SUBMIT` block.

```
SUBMIT < …some variables to pass, SAS language will see as macro variables…> ;
  ODS OUTPUT
    ANOVA              = oAnovaFits
    Fitstatistics      = oFitstatistics
    ParameterEstimates = oParameterEstimates ;

  PROC REG  DATA=RegIn   TABLEOUT ;
    &ModelStat : model &yVarName = &RegXVarNames / clb
      ADJRSQ AIC BIC CP EDF GMSEP JP MSE PC RSQUARE SBC SP SSE ;
          &WhereClause
    OUTPUT OUT=RegOut  P=&predName  RESIDUAL=&residName  LCLM=&lclmName
           UCLM=&uclmName  LCL=&lcliName  UCL=&ucliName  H=&Hat
           RSTUDENT=&rStudent  DFFITS=&DFFITS  COVRATIO=&CovRatio;
  QUIT;
  PROC SQL NOPRINT;
    CREATE TABLE FileStat <… query to merge regression parameters from ODS tables…>
ENDSUBMIT;
```

DISPLAY THE REGRESSION EQUATION

The regression parameters are imported to display the equation in the plot. Any previous equations are removed with `DrawRemoveCommands()` and then the new equation is defined within a drawing block. All elements created within a drawing block can be added or removed with one command.

```
  plot.DrawRemoveCommands("Regress Equation");

  declare DataObject RegParms;
  RegParms = DataObject.CreateFromServerDataSet( FileStat);
  RegParms.GetVarData( "Intercept", Reg_Intercept);
  RegParms.GetVarData( "Slope",     Reg_Slope);
  RegParms.GetVarData( "R_Square",  Reg_r2);
  Reg_Eq = CONCAT( 'Reported = ',  STRIP(PUTN( Reg_Intercept, '4.2' )), ' + ',
           STRIP(PUTN( Reg_Slope, '4.2' )), "*Classified \n      r2 = "J
          ,STRIP(PUTN( Reg_r2,  '5.3' )) ) ;
  plot.DrawBeginBlock( "Regress Equation" );
  plot.DrawPushState();
  plot.DrawResetState();
    plot.DrawSetTextTypeface( "Courier New" );
    plot.DrawSetTextStyle( STYLE_BOLDITALIC );
    plot.DrawSetTextColor( MAGENTA );
    plot.DrawSetTextSize( 11 );
    plot.DrawSetTextAlignment( ALIGN_LEFT, -1 );
    plot.DrawText( 30, 93, Reg_Eq );
  plot.DrawPopState();
  plot.DrawEndBlock();
```

The PROC REG output data set is imported into a data object for plotting the regression (as the in the original ActionMenuScatterPlot.sx) and to highlight data points with high influence.

```
  DECLARE DataObject dobjOut;
    dobjOut = DataObject.CreateFromServerDataSet( "work.RegOut" );
```

HIGHLIGHT OUTLIERS

The data is merged back to the data object table based on the observation number. If a variable already exists it is deleted first. The regression line, 95% prediction limits and 95% confidence limits are plotted. A legend for the

lines is drawn. Influence is determined by these thresholds which mostly follow the SAS documentation recommendation:

```
PCR      = 2;                                    /*- Number of model parameters */
dobjOut.GetObsNumbersInAnalysis( AnalyCnt ); /*- Number of observations */
FN       = NROW( AnalyCnt );
HatCR    = 2*PCR/FN;
CovCR    = 6/FN;
DffCR    = 2*SQRT( PCR/FN );
RstCR    = 2.0;
```

If an influence indicator exceeds a threshold the “OutCount” variable is incremented. The “OutIndicator” variable is set to indicate the source of the outlier with a “1” reflecting the relative position of the influence variable in the table as seen in figure 4.

Regression_Build_v1b (ActionMenuScatterPlot_5):1

		CropName	Stratum	Segment	Hat	rStud	DFFITS	CovRat	OutCount	OutIndicator
		Nom	Nom	Nom	Int	Int	Int	Int	Int	Int
141	PC	Rice	11	141	0.01726	0.57964	0.07682	1.02347	1	1000
26	PC	Rice	11	26	0.01761	-0.437	-0.0585	1.02512	1	1000
157	PC	Rice	11	157	0.0177	-1.2693	-0.1704	1.01263	1	1000
160	PC	Rice	11	160	0.01885	-0.0985	-0.0136	1.02805	2	1001
71	PC	Rice	11	71	0.01895	0.04721	0.00656	1.02821	2	1001
3	PC	Rice	11	3	0.01913	-1.5647	-0.2185	1.00678	2	1010
140	PC	Rice	11	140	0.01941	1.20591	0.16964	1.01577	1	1000
110	PC	Rice	11	110	0.01987	0.24077	0.03428	1.02868	2	1001
79	PC	Rice	11	79	0.02191	1.89594	0.28377	0.99972	2	1010
4	PC	Rice	11	4	0.02221	-3.3764	-0.5089	0.93614	4	1111
81	PC	Rice	11	81	0.02581	-0.249	-0.0405	1.03492	2	1001
2	PC	Rice	11	2	0.02592	-0.3723	-0.0607	1.03434	2	1001

Figure 4. Influence values, threshold counts and indicators, colors match scatterplot colors.

In order to highlight the segments with more than one outlier indicator, their indices are saved to arrays and the marker and color methods are called for the plot. The result is shown in figure 5.

```
Out2     = LOC( OutLierTemp=2 );
Out3     = LOC( OutLierTemp=3 );
Out4     = LOC( OutLierTemp=4 );
OutSel34 = LOC( OutLierTemp>2 );
Out234   = Out2||Out3||Out4;

IF NCOL(Out234)>0 THEN
   plot.SetMarkerShape( Out234, MARKER_X );

IF NCOL(Out2) > 0 THEN DO;
plot.SetMarkerColor( Out2, GREEN );
END;
IF NCOL(Out3) > 0 THEN DO;
   plot.SetMarkerColor( Out3, YELLOW );
END;
IF NCOL(Out4) > 0 THEN DO;
   plot.SetMarkerColor( Out4, RED );
END;
IF NCOL(OrigSelIdx)>0 THEN DO;
   plot.SetMarkerShape( OrigSelIdx, MARKER_CIRCLE );
   plot.SetMarkerColor( OrigSelIdx, BLUE );
END;
```

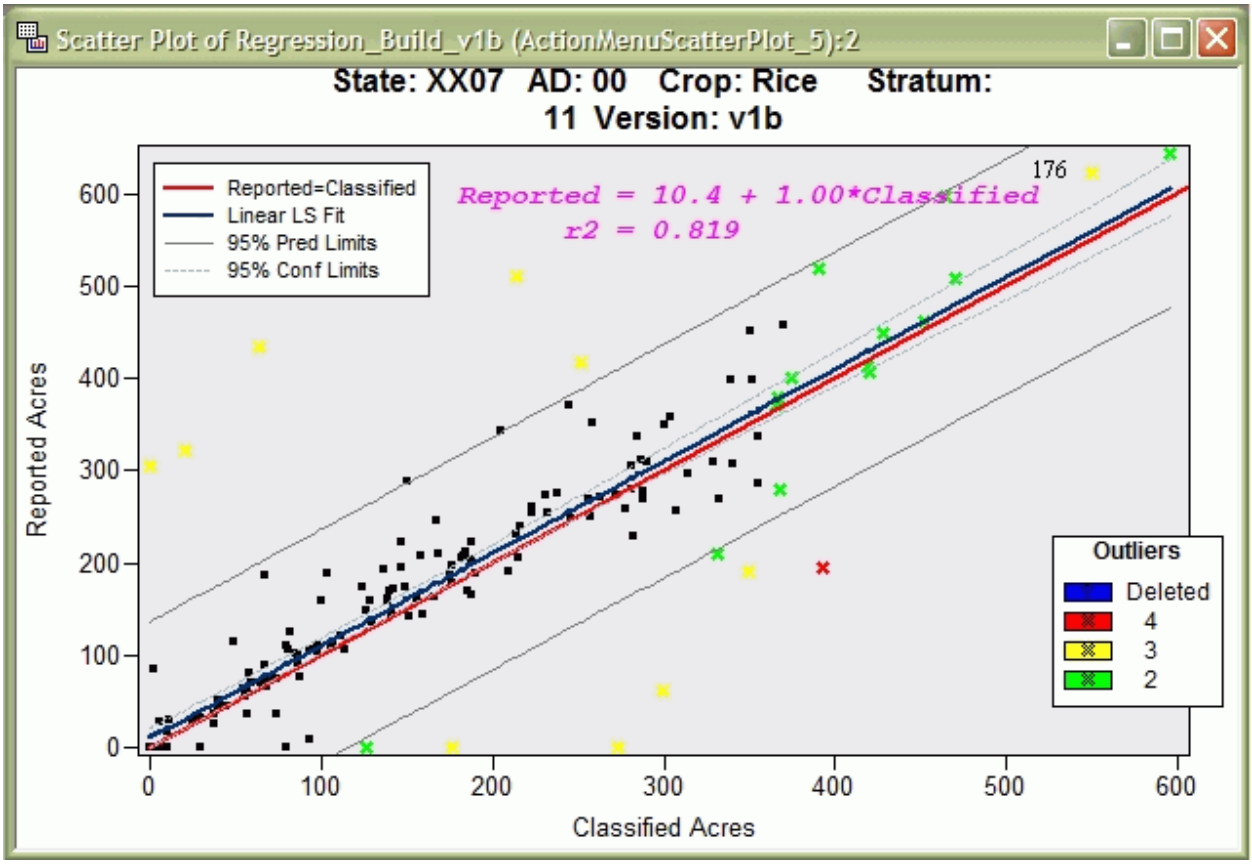

Figure 5. The first regression showing initial outliers, color highlighted by number of thresholds exceeded.

OUTLIER REVIEW

Once the potential outliers have been highlighted they are reviewed for possible exclusion in the next regression. Our general rule is that any segment that has three or four outlier indicators is a strong candidate for exclusion, keeping in mind that an outlier does not necessarily mean bad data. To exclude a segment it must first be added to the SegsDel data object. Segments can be selected individually and accumulated by holding down the control key, or selected by holding down the control key and dragging the mouse to create a selection rectangle. On the action menu the option to "Select Obs w/ 3-4 outliers" is added which runs:

```
dobj.SelectObs(OutSel34, false);
```

Additional selections or unselections can be made. The action menu item "Exclude selected from next regression" runs module `SaveDeleted` to save to the SegDel data object and to a permanent SAS data set.

```
START SaveDeleted ;
   /*- Get variable information of deleted observations */
/*- Create scatterplot object that will hold county map and seg centers */

   dobj.GetVarSelectedData( "Segment", soSegment );
   dobj.GetVarSelectedData( "State",   soState );
   dobj.GetVarSelectedData( "AD",      soAD );
   dobj.GetVarSelectedData( "CropName",soCropName );
   dobj.GetVarSelectedData( "Stratum", soStratum );
   dobj.GetVarSelectedData( "Year",    soYear );
   soStratCombo = J(NROW( soSegment ), 1, setStrat_title );
   /*- Merge into one data set */
   soAll = soStratCombo||soSegment||soState||soYear||soAD||soCropName||soStratum;
   /*- Add to the deleted table object */
   dobjSegDel.AddObs( soAll );
   /*- Write to file that was originally read in */
   dobjSegDel.WriteToFile( File2 );
FINISH SaveDeleted;
```

Whenever a regression is run, it first checks the SegsDel SAS dataset to find and exclude any matches. If a segment needs to be undeleted it can be selected and deleted from the SegsDel data object by using the right-click

menu when the mouse pointer is over a selected row label as in figure 6. To save the deletion permanently, activate the plot window's action menu and select "Segments: Save Deleted Table" as seen in figure 3.

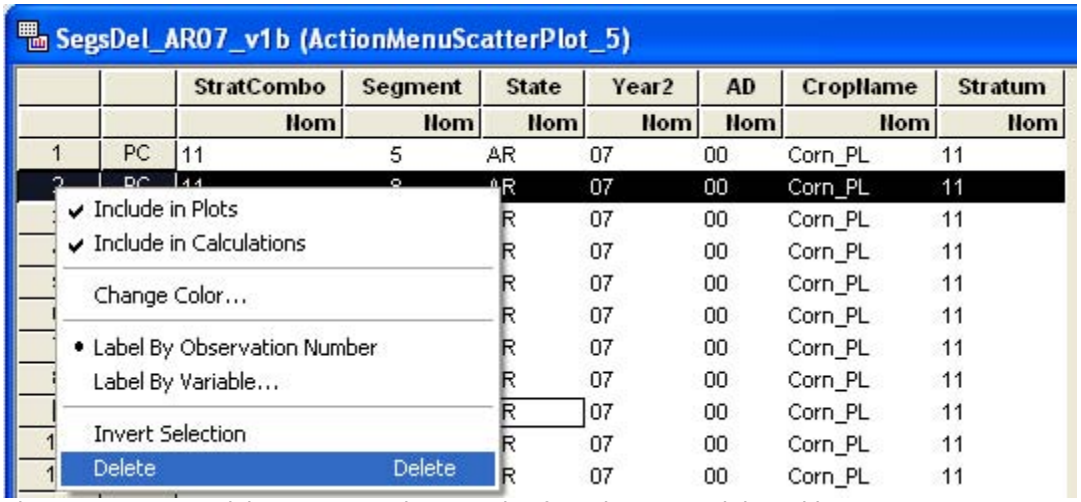

Figure 6. How to undelete segments by removing from the SegsDel data object.

In the example in figure 5 all of the segments with three to four outliers were initially selected, then segment 176 was unselected since it is in the main spread of data. The regression is run again from the action menu. Figure 7 is the updated plot after the deletions and the next regression. Note that the equation and r^2 have been replaced. The program has marked the deleted segments in blue and highlighted a new set of potential outliers. Generally, no further deletions are done after the initial segments with 3 and 4 indicators have been deleted. The parameters of the last regression run for a stratum are then used to estimate crop acreage at the state level.

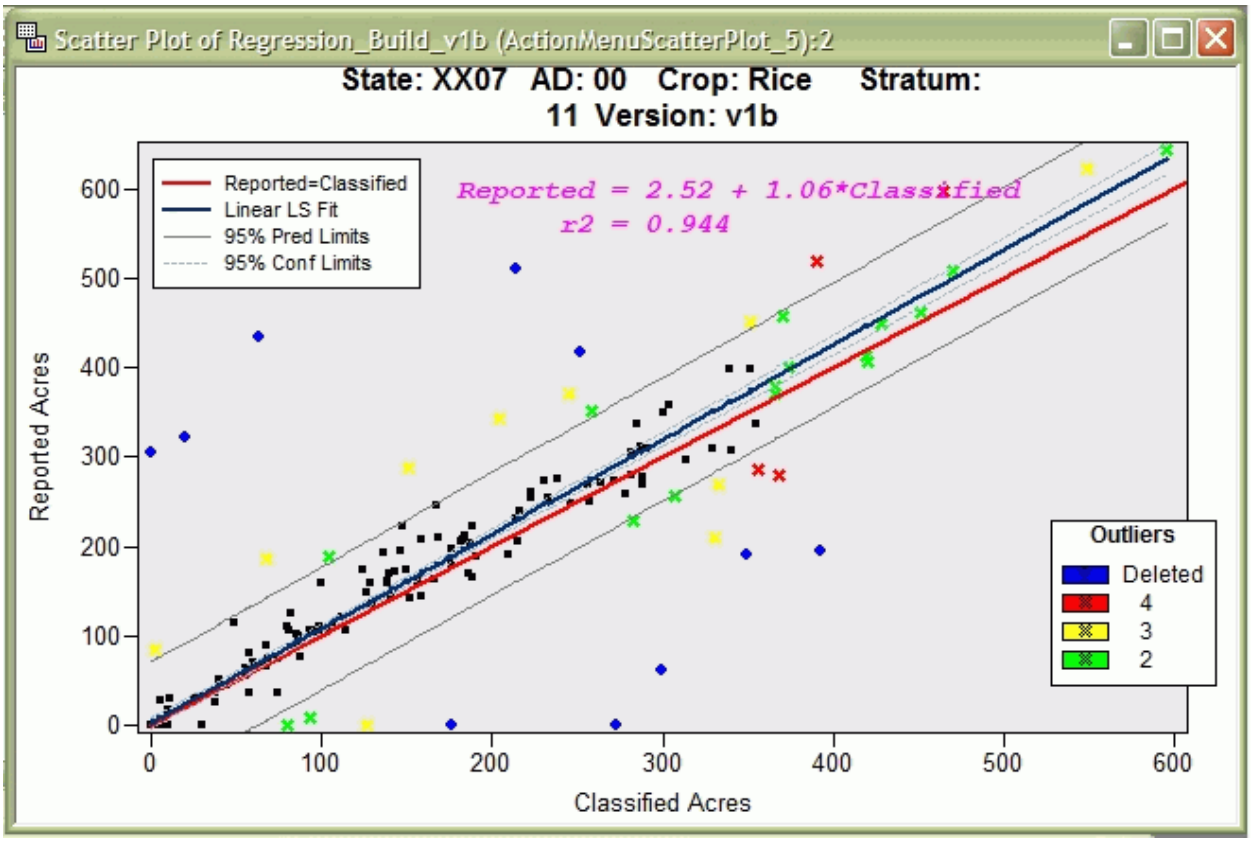

Figure 7. Shows the results of the regression after outliers have been deleted.

MAPPING THE SEMENTS AND DATA LINKAGE

The segments also have a spatial component, their location in the state. The analyst can look for spatial patterns in the distribution of the outliers by mapping the segment centers with the color and shape coded marker used for the regression scatter plot. Non-random patterns may suggest systematic problems with the reported and/or classified acre data.

The code below creates a new scatter plot, with the coordinates of the segment centers as the data points. It then imports a polygon data set of county boundaries, subsets the data to the current state and draws in the coordinate system already in place and plots in the background of the new scatter plot. In a similar manner the county FIP codes are plotted at county centers. Since the data points are linked by the ***dojb*** table any color and marker definitions in one chart are reflected in the other chart, including selections.

```
START MapSetup;
       /*- Create scatterplot object that will hold county map and segment centers */
       declare ScatterPlot plotMap;
       plotMap = ScatterPlot.Create( dobj, "Center_X", "Center_Y", false );
       /* draw map */
       plotMap.DrawSetRegion( PLOTBACKGROUND );
       plotMap.DrawUseDataCoordinates();
       /* Open polygons of county boundaries*/
       declare DataObject dobjMap;
       dobjMap = DataObject.CreateFromFile( "Data_Sets\uscntygen.sas7bdat" );
       /*- Subset to current state */
       dobjMap.GetVarData( "State", StateRows );
       NotThisState = LOC( StateRows ^= State);
       dobjMap.DeleteObs( NotThisState );
       run DrawPolygonsByGroups( plotMap, dobjMap,
             "Center_X", "Center_Y", {"state" "PolyNum"},
             "Uniform", brown//ltBlue, true );
FINISH;
```

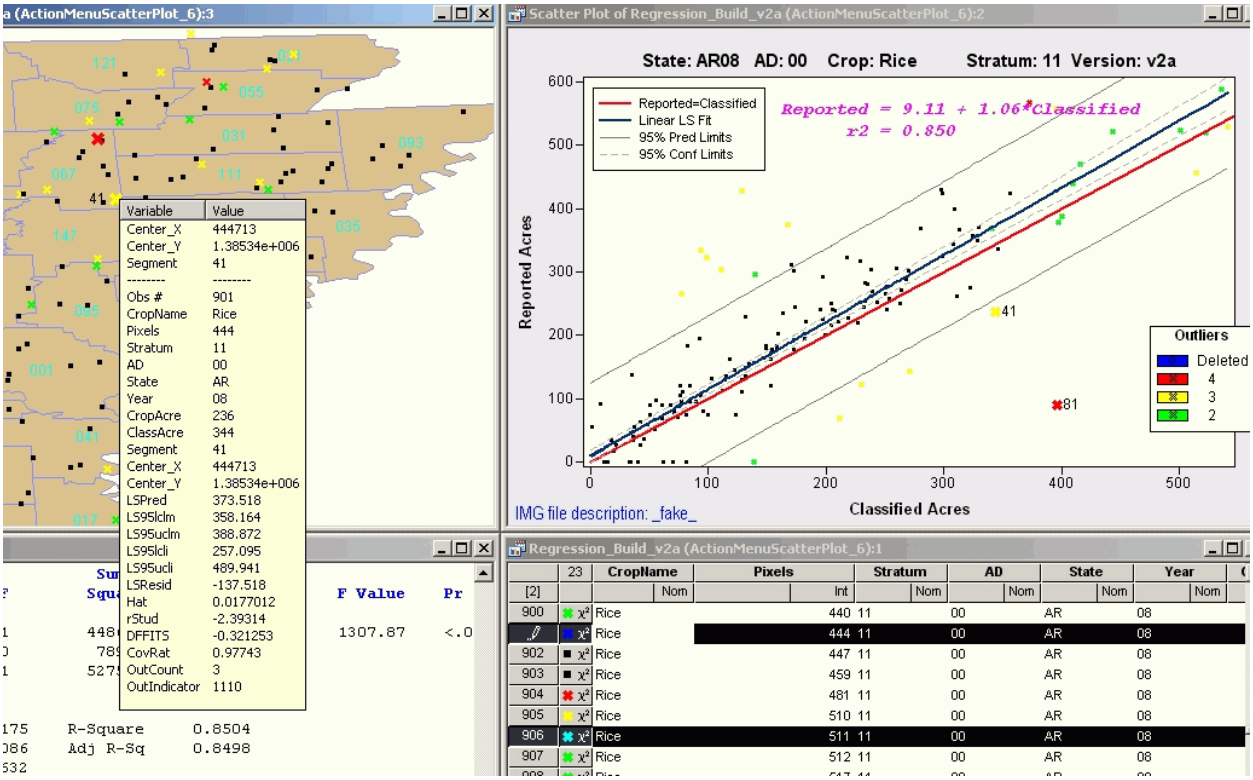

Figure 8. Spatial display of data with across object data selection and observation inspector.

Figure 8 illustrates the links of the selection in the two scatter plots and the data table and that the markers and colors of the data points are the same between the scatter plots and the data table. In addition, the “observation inspector” can be activated with the F2 key when the pointer is over an observation in an active plot window. A drop down window appears with the values of that observation in the data table.

CONCLUSION

IML Workshop in SAS 8.2/9.1 and now Stat Studio in SAS 9.2 offer powerful tools for interactive data analysis. The SAS supplied examples and demos illustrate some of the functionality available in Stat Studio and provide a good starting point for further development. This project started with the SAS supplied ActionMenuScatterPlot.sx and reproduced the statistical procedures of the legacy Peditor software and created a dynamic and interactive interface which greatly improved the ease and speed of the data review process. Maps with linked data points added another dimension to the review of the data. Stat Studio is the ideal setting for this analysis as it provides interactive analysis and a portal to the rest of SAS.

REFERENCES

Bellow, Michael E. "Large Domain Satellite Based Estimators of Crop Planted Area," presented at / submitted for: American Statistical Association, 1994 Proceedings of the Section on Survey Research Methods, Toronto, Canada, August, 1994.

Day, C.D. (2002) "A Compilation of PEDITOR Estimation Formulas". RDD Research Paper RDD-02-03, USDA, NASS, Washington, D.C. January, 2002.

Graham, Mitchell L. "State Level Crop Area Estimation Using Satellite Data in a Regression Estimator." paper given at International Conference on Establishment Surveys (ICES); Survey Methods for Businesses, Farms, and Institutions; in Buffalo, New York, June 28-30, 1993, pp. 802-806. Printed version available in: ICES Part 1, USDA, NASS, Research Division, SRB Research Report No. SRB-93-10, September, 1993.

Mueller, Rick and Robert Seffrin (2006) "New Methods and Satellites: A Program Update on the NASS Cropland Data Layer Acreage Program" Proceedings of the Symposium on Remote Sensing Support to Crop Yield Forecast and Area Estimates, ISPRS Commission VII, WG VI/4, November 30 - December 1, 2006, Stresa, Italy.

National Agricultural Statistics Service (NASS) Online, 2007, <http://www.nass.usda.gov/research/Cropland/SARS1a.htm> Accessed 2008 01 July.

CONTACT INFORMATION

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

Robert Seffrin
USDA-NASS
3251 Old Lee Hwy
Fairfax, VA 22030-1504
Work Phone: 703-877-8000 ext. 155
Fax: 703-877-8044
E-mail: Robert_seffrin@nass.usda.gov