Print

Demo: Examining the Influential Observations Using PROC PRINT

Filename: **st105d02.sas**

In this demonstration, we use PROC PRINT to take a look at the output data sets that we created in PROC GLMSELECT. We then combine them into a single data set with only observations that exceed the suggested cutoffs of the influence statistics.This part of the demonstration uses many programming concepts, including MERGE statements, arrays, and DO loops, which you can learn about in SAS programming courses.

1. Open program st105d02.sas.

```
%let interval=Gr_Liv_Area Basement_Area Garage_Area Deck_Porch_Area
         Lot_Area Age_Sold Bedroom_AbvGr Total_Bathroom ;

/*st105d02.sas*/ /*Part A*/
ods select none;
proc glmselect data=STAT1.ameshousing3 plots=all;
        STEPWISE: model SalePrice = &interval / selection=stepwise details=steps select=SL slentry=(
        title "Stepwise Model Selection for SalePrice - SL 0.05";
run;
quit;
ods select all;

ods graphics on;
ods output RSTUDENTBYPREDICTED=Rstud
           COOKSDPLOT=Cook
           DFFITSPLOT=Dffits
           DFBETASPANEL=Dfbs;
proc reg data=STAT1.ameshousing3
         plots(only label)=
              (RSTUDENTBYPREDICTED
               COOKSD
               DFFITS
               DFBETAS);
    SigLimit: model SalePrice = &_GLSIND;
    title 'SigLimit Model - Plots of Diagnostic Statistics';
run;
quit;


/*st105d02.sas*/  /*Part B*/
title;
proc print data=Rstud;
run;

proc print data=Cook;
run;

proc print data=Dffits;
run;

proc print data=Dfbs;
run;

data Dfbs01;
        set Dfbs (obs=300);
run;

data Dfbs02;
        set Dfbs (firstobs=301);
run;

data Dfbs2;
        update Dfbs01 Dfbs02;
        by Observation;
run;


data influential;
/*  Merge datasets from above.*/
    merge Rstud
```

```
          Cook
          Dffits
          Dfbs2;
    by observation;

/*  Flag observations that have exceeded at least one cutpoint;*/
    if (ABS(Rstudent)>3) or (Cooksdlabel ne ' ') or Dffitsout then flag=1;
    array dfbetas{*} _dfbetasout: ;
    do i=2 to dim(dfbetas);
         if dfbetas{i} then flag=1;
    end;

/*  Set to missing values of influence statistics for those*/
/*  that have not exceeded cutpoints;*/
    if ABS(Rstudent)<=3 then RStudent=.;
    if Cooksdlabel eq ' ' then CooksD=.;

/*  Subset only observations that have been flagged.*/
    if flag=1;
    drop i flag;
run;

title;
proc print data=influential;
    id observation;
    var Rstudent CooksD Dffitsout _dfbetasout:;
run;
```

In Part B we use PROC PRINT steps to print each of the output data sets, Rstud, Cook, Dffits, and Dfbs. Examining these data sets might help us later when we combine them into one data set.

2. Submit the four PROC PRINT steps in Part B.

3. Review the output.

   In the first PRINT Procedure output, RStud data set, notice that it includes a column for the SigLimit label that we gave our model in the first part of the demonstration. In the outLevLabel column, the observations that have a value are the ones that were flagged as influential based on the RStudent cutoff. Notice that all RStudent influence statistic values are in the RStudent column.

   Next PRINT Procedure output is the Cook data set. The variable CooksDLabel identifies observations that are deemed influential due to high Cook's D values. These are the observations that have an influence on all the estimated parameters as a group.

   Now let's look at the PRINT Procedure output for the Dffits data set. The variable DFFITSOUT identifies observations that are deemed influential due to high DFFITS values. These are the observations that have an influence on the predictions.

   Finally, in the PRINT Procedure output for the Dfbs data set, the variables _DFBETASOUT1 through _DFBETASOUT8 identify the observations whose DFBETA values exceed the threshold for influence. _DFBETASOUT1 represents the value for the intercept, and the other seven variables show influential outliers on each of the predictor variables in the MODEL statement. The order of the predictor variables is based on the order of the variables that are listed in the MODEL statement (or in this case, in the _GLSIND macro variable).

   Think back to the DFBETAS panel plot from Part A, where we saw the order of the variables in _GLSIND are above Ground Living Area, Basement_Area, and so on. There were too many predictor variables to fit into one panel, so SAS produced a second panel plot.

   With the multiple panels for DFBETAS, the Dfbs data set is relatively split. The first 300 observations display the DFBETAS information for the first panel, which includes the first six effects in the model (including the intercept). The information for the second panel, which includes the final two effects, is missing.

   Beginning at observation 301, this is reversed. SAS stacked the rows of the first panel on top of the rows of the second panel. Let's find a way to merge the two panels by observation.

   The first DATA step in Part B copies the first 300 observations to a new data set, Dfbs01, and the second DATA step copies the remaining observations, beginning at 301, to Dfbs02. The third DATA step uses the UPDATE statement to combine data sets by Observation, and create the data set Dfbs2.

4. Submit the first three DATA steps in Part B and take a look at the new data sets. Note: The new data sets will display automatically in SAS Studio's table viewer. If you're not working in SAS Studio, you will need to submit a PROC PRINT step for each data set, Dfbs01, Dfbs02, and Dfbs2.

   The Dfbs01 data set contains the first 300 observations, and the columns for the last two predictor variables contain only missing values.

   The Dfbs02 data set contains the last 300 observations, and the columns for the first six predictor variables contain only missing values.

   The combined data set, Dfbs2, contains all non-missing values from the previous two data sets.

5. Let's return to the Code. The next DATA step merges the final Dfbs2 data set with the Rstud, Cook, and Dffits data sets by Observation.These are the four data sets that contain the influence data.

   The IF statement identifies observations that exceed the respective influence cutoff values. In this case, it identifies observations with an RStudent value beyond the thresholds of 3 and -3, Cooksdlabel values that are not equal to a missing value in other words, any observations that were flagged as influential based on the Cook's D cutoff, and any observations that have a Dffitsout value that is, any observations that were flagged based on the DFFITS cutoff. You can change the cutoff thresholds in this statement.

   Next we want to flag observations with non-missing values in any _DFBETASOUT columns. We'll use an array, a DO loop, and an IF statement to flag these observations.

   Now we need to do a little cleanup. For observations that were flagged as influential based on the cutoffs for Cook's D, DFFITS, and DFBETAS, but not flagged

by RStudent, we don't want the RStudent value in the output data set. These IF statements assign a missing value for a statistic if the observation doesn't exceed the cutoff point for that particular statistic. Finally we use a subsetting IF statement to write only the flagged observations to output. Following the DATA step, we use a PROC PRINT step to display the output data set, influential.

6. Submit the last two steps in Part B.

7. Review the output.

   The PRINT Procedure output displays a summary of only the influence statistics that were outside the cutoff boundaries. The columns for the DFBETAS values still begin with _DFBETASOUT. Notice that, if we wanted to, we could rename these to make it clear which one is for the intercept and which ones are for each of the predictor variables.

   Now that we have the flagged observations, we can investigate them further, first to filter out any erroneous data, and then to determine what makes each point influential.

Close