

Lab 4: Regression, Part 2
STT 422: Summer, 2004
Vince Melfi

We continue our study of regression in SAS. In particular, we'll learn how to save and plot quantities like fitted values, residuals, etc. Before that we'll learn a bit about manipulating SAS output for saving and printing.

The Results Window

The **Results Window** is to the left of the windows we usually work with. Every time we run a SAS program that creates output, an entry is added to the results window. Starting in a new SAS session, run the following program.

```
data anscombe;
  infile 'U:\msu\course\stt\422\summer04\anscombe.dat';
  input x1 y1 x2 y2 x3 y3 x4 y4;

proc print data = anscombe;
  var x1 y1;
  title 'Print statement';

proc reg data=anscombe;
  model y1 = x1;
  plot y1*x1;
  title 'Reg statement';

proc univariate data=anscombe;
  title 'Univariate statement';

run;
```

This should create a few pages of output in the **Output Window** and should create a graph in the **Graph Window**. Look in those windows to make sure the program ran properly. Now look in the **Results Window**. You should see a listing of three results, one for each **proc** we ran. The first is labeled **Print: Print Statement**. The second is labeled **Print: Reg Statement**. The third is labeled **Univariate: Univariate statement**. The labels include both the **proc** name and the (rather uninformative) title we chose.

As you have seen in the first few labs, SAS quickly creates a huge amount of output. The **Results Window** gives us a way to quickly navigate that output and to delete unwanted output. For example, suppose we want to see the output of **proc univariate** for the variable **x3**.

1. Double click on **Univariate: Univariate statement** in the **Results Window**. This should open eight entries just below, one for each variable.

2. Now double click on **x3**. (Instead of double-clicking, you can also click on the + sign next to the name.) This should open entries below **x3** for **Moments**, **Quantiles**, etc.
3. Double click on **Moments** and the **Output Window** should come to the top with the Moments of **x3** showing.

In this way you can navigate through the SAS output from the program(s) you've run. Note that you can unexpand in the **Results Window** by clicking on the - sign next to the name. For example, click on the - sign next to **Univariate: Univariate statement**. It's also possible to delete results from the **Results Window**. Suppose we decide that we don't care about the output from the **proc reg** statement. We can get rid of this by selecting **Reg: Reg statement** in the **Results Window** and then using **Edit ► Delete**. This removes both the appropriate output from the **Output Window** and the appropriate graph from the **Graph Window**.

Note: If you remove all the graphs from the **Graph Window** in this way, it seems necessary to close the **Graph Window** (by clicking on the "x" in the upper right corner) before running another graphics-producing program.

Clearing windows

The menu command **Edit ► Clear** gives another way to remove extraneous output. For example, suppose we want to remove all the output and start over. We can select the **Output Window** and then use **Edit ► Clear** to get rid of all the output. Note that this will remove the corresponding entries from the **Results Window**. But this method doesn't work for the **Graph Window**. Maybe the best use of **Edit ► Clear** is in the **Log Window**.

Saving and printing output

It's reasonably easy to print SAS output and to transfer SAS output to other programs (e.g. Microsoft Word). Unfortunately, the antediluvian printing system in the microlabs makes printing a bit more of a challenge than usual. (NOTE: You can go through the material below more quickly and without wasting paper by choosing to preview the printed output but then canceling the print before it's sent to the printer.)

Printing

If you want to print the entire contents of a non-graphics window, just make the window active and select **File ► Print**. As with most Windows printing, you can preview before printing, or select the specific pages you want to print. To print a graph, just bring that graph to the top in the graphics window and select **File ► Print**. Again you have the option of previewing. For some reason the default behavior in SAS is to print only the current graph rather than all the graphs in the **Graph Window**.

It's also possible to specify which portions of the output you want to print. For example, the **proc reg** statement above creates a **Reg** entry in the **Results Window**. (If you have removed this, please rerun the above program to create a new version.) Expand this by clicking on the + next to **Reg**, then clicking on the + next to **MODEL 1**, then clicking on the

+ next to **Fit**, and then clicking on the + next to **y1**. Now if you just want a printout of the parameter estimates from this fit, click on **Parameter Estimates** and then select **File ► Print**.

Unfortunately, when you send something to the printer in a microlab, it will be hard to tell which output flowing from the printer is yours, since twenty other classmates have probably sent similar output to the printer at about the same time. The solution: Put a distinguishing title on your output! One way, for those who are lazy, is to specify a title statement somewhere near the top of your program and then not specify any other title statements. Then the title you specified will apply throughout the program. For example, I might use `title 'Melfi'`; in my first **proc** and then this title would show up on all my printouts. A better way is to specify informative titles for all your **proc** statements. For example, I might use `title 'Melfi: Regression of y1 on x1'`; after a **proc reg** statement, and then use `title 'Melfi: Univariate statistics for Anscombe data'`; after a **proc univariate** statement.

Since the printers are slow and noisy, it's a good idea to limit printing as much as possible.

Exporting SAS output

It's possible to cut and paste SAS output into other programs. For example, go to the **Output Window** and select some of the output. Then choose **Edit ► Copy** to copy the output to the clipboard. Then you can paste the output into any simple text editor, such as Notepad, or into a word processing program such as Microsoft Word.

For the **Graph Window** much the same procedure works. Go to the plot of interest, then select **Edit ► Copy** to copy the plot to the clipboard. But since the plot is not plain text, it can't be pasted into Notepad. It can, however, be pasted into Microsoft Word.

There are all sorts of other possibilities. For example a graph can be exported in various image formats. Choose the plot, then go to **File ► Export as Image**. Choose the type of image (bmp, gif, jpg, etc.) in "Save as type:" and then choose a filename for the saved image.

If you save any files in the microlab, you'll probably want to put them on a floppy disk or in your **P:** drive.

Residuals and Influential points

Plots of residuals versus various other quantities (fitted values, x-values, time, etc.) provide important information about outliers, influential points, and the appropriateness of using a linear model to summarize the relationship between x and y .

The data from Problem 2.24 in the text, available in `u:\msu\course\stt\422\summer04\tobacco.dat`, provide values for per-capita spending on alcohol (x) and tobacco (y) for various parts of Great Britain. We will use linear regression modeling to illuminate the relationship between these variables. First, a bit of SAS code to read in the data, print it, and fit a simple linear regression model including a plot.

```

data spending;
  infile 'u:\msu\course\stt\422\summer04\tobacco.dat';
  input alcohol tobacco;

proc print data=spending;
  var alcohol tobacco;
  title 'Melfi: spending dataset';

proc reg data=spending corr;
  model tobacco = alcohol;
  plot tobacco*alcohol;
  title 'Melfi: Regression tobacco versus alcohol';

run;

```

The plot and the small value of r^2 suggest that the linear relationship between spending on alcohol and spending on tobacco is weak. But a closer look at the plot reveals a point in the upper left that is clearly an outlier, and is probably influential. (From the text we can learn that this point corresponds to Northern Ireland.) We'll investigate this by first making some residual plots, then constructing a new dataset excluding the potential outlier and fitting the model without this point to see how much the fitted line, correlation, etc. change. The code below (added to the code above) suffices.

```

proc reg data=spending corr;
  model tobacco = alcohol;
  plot tobacco*alcohol residual.*alcohol residual.*predicted. ;
  title 'Melfi: Regression tobacco versus alcohol';

data spending1;
  set spending;
  if alcohol = 4.02 then delete;

proc print data = spending1;
  var alcohol tobacco;
  title 'Melfi: Spending data excluding Northern Ireland';

proc reg data = spending1 corr;
  model tobacco = alcohol;
  plot tobacco*alcohol;
  title 'Melfi: Regression excluding Northern Ireland';

run;

```

Explanation

1. The first **proc reg** refit the original (full data) model, but this time added a plot of the residuals versus alcohol and the residuals versus the predicted values. Note that SAS names the variable containing the residuals **residual.** and names the variable containing the predicted values **predicted.** The **.** at the end is irritating but important.
2. The **data** statement creates a new data set called **spending1**. The **set spending** line tells SAS to start with the **spending** dataset. The line **if alcohol = 4.02 then delete** tells SAS to delete all cases with **alcohol = 4.02**. Since there's only one of these, and it is the outlier, this is what we want. This is verified from the output of **proc print**.
3. Next we refit the model, this time to the dataset **spending1** which excludes the outlier. From the plots and output, does the model fit this reduced dataset much better???

Saving regression quantities

Sometimes it's nice to save quantities like residuals, predicted values, etc. computed in a regression analysis. The **output** statement can do this. Here is a simple example using the original spending data.

```
data spending;
  infile 'u:\msu\course\stt\422\summer04\tobacco.dat';
  input alcohol tobacco;

proc reg data=spending corr;
  model tobacco = alcohol;
  output out = regsave predicted = yhat residual = r;

proc print data=regsave;

run;
```

As you can see from the results of the **print** statement, the output dataset automatically contains the variables from the regression (in our case, **alcohol** and **tobacco**). In addition, we specified that two other variables be added. The first, called **yhat**, contains the predicted values from the fitted model. The second, called **r**, contains the residuals.