Homework Assignment 1: STATA AND SAS COMMANDS (Prelim)

Notes:

- Ordinarily I just include the STATA and SAS commands at the end of the assignment. However this time there are a lot of them so I decided it would be easier to refer to if I put them in a separate file.
- For this assignment you need to be able to perform a two-sample test of proportions and a chi-squared (contingency table) test, fit a logistic regression model, generate a new variable and create a couple of simple graphics. The corresponding commands are given below. I have also included explanations of how to get STATA to give probabilities associated with the normal and chi-squared distributions.
- Because this is the first assignment, I will also add some review of the basics of each package. An updated version of the file including these materials will be posted shortly.

Commands in STATA

Test of Proportions: To perform tests of proportions you use the **prtest** command. The help file gives details of many different variations. If in particular you want to do a two-sample test of proportions and you have the outcome variable in one column and the grouping variable in a second column (as in warm-up problem 5a or turn-in problem 6b) you use the **by** option. The basic command is

prtest Y, by(group)

For example, for warm-up problem 5 we would type

prtest asthma, by(urban)

Contingency Table Tests: The general command in STATA for creating and analyzing tabular data is (surprise!) **tab** which is short for **tabulate**. It has many display options including row and column percentages, expected counts, and the like. If you want to get the test statistic and p-value for the standard Pearson chi-squared test (e.g. for warm-up problm 5b or turn-in problem 6c) you use the option chi2. This is illustrated below for the asthma data:

tab asthma urban, chi2

Logistic Regression: STATA has two commands for logistic regression. The first, **logit**, produces the estimates on the logit scale (i.e. it gives you $\hat{\beta}_0, \hat{\beta}_1, ...$). The second, **logistic**, gives you the estimates on the odds ratio scale. Both give the overall summary statistics. However only the logistic command allows follow-up test commands. For example to fit the logistic model relating the likelihood of a child getting asthma to the variables urban, pollution, ses, breastfed, famhist, and gender in warm-up problem 5 we would type one of

logit asthma urban pollution ses breastfed famhist gender
logistic asthma urban pollution ses breastfed famhist gender

If we wanted to test whether the family/child characteristics (ses, breastfeeding, family history, gender) added explanatory power to the model over and above that provided by the environmental characteristics (urban and pollution) we could follow the logistic statement with the command

test ses=breastfed=famhist=gender=0

This performs the likelihood ratio chi-squared test comparing the models with and without these variables. Of course you could also simply fit the two models and calculate the chi-squared statistic from the log likelihoods. You can do this manually or you can use the command **lrtest**. Doing the latter requires storing the results of the various models as you go along. For example, if you wanted to do the comparison represented by the above test command you would use the following sequence

logit asthma urban pollution
estimates store ENVIRON

logit asthma urban pollution ses breastfed famhist gender
estimates store FULL

lrtest ENVIRON FULL

This involves fitting the models, storing their results (in ENVIRON and FULL respectively) and then using **lrtest** to compare the two stored models.

Distribution Calculators: If you are doing hand calculations (e.g. a chi-squared test based on log-likelihoods from a pair of models) it is sometimes useful to be able to get the probability associated with a particular value from a common distribution like the normal of chi-squared. STATA has commands for all of these. The ones that would be most relevant for this assignment are the normal and chi-squared commands. The command

display normal(z)

gives you the probability that a standard normal distribution (e.g. a Z score) is **less** than the value z.

For instance the command normal(-1.96) produces the output

```
. display normal(-1.96)
.0249979
```

If you want the probability of being **greater** than a particular value z you will need to subtract the probability obtained from the command above from 1.

For the chi-squared distribution you need to tell STATA the degrees of freedom and it gives you the probability of being **greater** than the specified value. This is because we are usually interested in large chi-squared values as they are the ones that correspond to significant tests. The basic command is

display chi2tail(df, X)

where df is the degrees of freedom and X is the value of the chi-squared statistic. For example for a chi-squared distribution with one degree of freedom the $\alpha = .05$ cutoff is 3.84 as you can tell from the following output:

```
. display chi2tail(1,3.84)
.05004352
```

Generating New Variables: In STATA you use the **gen** command to create a new variable using mathematical expressions and an existing variable. The basic command is

gen newvariable = mathexpression(old variables)

For instance, the weight variable in turn-in problem 7 is given in pounds. If we wanted to create a new variable, wtounces, that converted it to ounces we would type

gen wtounces = 16*weight.

The expressions can be arbitrarily complicated. Below I give the expressions needed for calculating predicted probabilities for Optional Bonus Problem 7h. To create a predicted probability in a logistic regression you need to use the formula

$$p = \frac{e^{X\beta}}{1 + e^{X\beta}}$$

The problem assumes a logistic model has been fit using the variables age, cholesterol level and family history. We want to get predicted probabilities for all the cholesterol levels in the data set (i) for people with a family history and (ii) without a family history, assuming the age is fixed at 50. In STATA the function for exponentiating is abbreviated **exp**. Make sure that you understand my equation and that it matches the output from your fitted model:

gen predprobfamhist = exp(-8 -.062*50 + .043*cholesterol+ 1.927)/(1+exp(-8 -.062*50 + .043*cholesterol + 1.927))

gen predprobnofamhist = exp(-8 -.062*50 + .043*cholesterol)/(1+exp(-8 -.062*50 + .043*cholesterol))

Note that you can also get these predicted probabilities in other ways by making clever use of STATA's post estimation commands, **predict** and **margin** which will be illustrated in lab, but I thought it was good to illustrate how you could get them from scratch.

Graphics: Problem 7h also asks you to plot the above predicted probabilities on a single graph as a function of cholesterol. The **scatter** command in STATA lets you include as many Y variables as you want to plot against a single X variable. The general command is

scatter Y1 Y2 Y3 X

You can have as many Y's as you want. The final variable is assumed to be the X variable. Thus for Problem 7h we type

scatter predprobfamhist predprobnofamhist cholesterol

For warm-up Problem 5d and turn-in problem 7a you also need to use a scatterplot. Here you first bin the pollution or cholesterol variable and then find the fraction of subjects in each bin who have the characteristic (asthma or arterisclerosis). I did the binning for you in the datasets to save time but below I illustrate several ways to do it from the raw data for those who are interested–a more manual approach and a slicker approach using the generate command to produce new variables. Note that you can either number the bins (1,2,3, etc.) or if you want your plot to use the actual X values a good choice is to label the bins by their center value. I included both versions (e.g. pollbin and pollbincent) as well as the percentages (e.g. asthmapct) corresponding to the manual approach in the data set. The slick approach produces actual bin memberships

and percentages for each data point but the plots still work!

Slick approach: You start by creating a new grouping variable which tells you which interval/bin the point is in. For the pollution example the values range from roughly 0-40 and I specified 8 bins so the bins should be of width 5. This if we divide the data values by 5 and round to the next **highest** integer we should get an appropriate binning variable. We do this using the ceiling function, abbreviated **ceil**:

gen pollbin = ceil(pollution/5)

There is one value just above 40 which you might want to not put in a separate bin–if you like you can reassign its bin value as follows:

replace pollbin = 8 if pollution > 40

Then to generate the percentages of people with asthma in each bin you take the mean of the asthma variable (i.e. the 0's and 1's) in each group using the **bysort** option with the **egen** command (another variable generating command that uses estimates rather than existing variables):

bysort pollbin: egen asthmapct = mean(asthma)

Less Slick Approach: I arranged the data in ascending order by these variables so that you could look at them in the data editor and identify the rows corresponding to each bin easily. You can use the **tab** command with the **in** option to give you the counts of people with and without the characteristics in the desired rows. For instance, to find the number of people with and without asthma in the first hundred rows of the data set you would type

tab asthma in 1/100

Once you have these values you can get the percentages for each bin and plot them. You can either do this by hand, or, if you want to create the scatter plot in STATA an easy option is to manually enter two new variables using the data editor, e.g. one labeled pollbincent which has as its values the center point of each bin you create and one labeled asthmapct which has the percentages you calculated. You can then type

scatter asthmapct pollbincent

Commands in SAS

Test of Proportions: To perform a test of proportions in SAS you can either do a contingency table test (which is mathematically equivalent-see the commands below) or you can use SAS's point and click interface which is called **Analyst**. To use Analyst proceed as follows. From the main menu bar select **Solutions/Analysis**. SAS will pop up a new project window. Then go back to the main menu, select **File/Open** and use the browser to select the homework 2 data. Again go to the main menu and select **Statistics/Hypothesis Tests/Two-Sample-Tests-of-Proportions**. You will get a dialog box where you select your outcome variable (e.g. for warm-up problem 5 this would be **asthma**), your grouping variable (e.g. **urban**), the level of interest for your outcome variable (this only matters if you are doing a 1-sided test), the value you want to test for the difference of proportions (the default is 0 which is what you will usually want) and your alternative hypothesis (not equal, greater than, less than). There are lots of other options you can select too. Click OK and SAS will give you the printout (unfortunately the formatting isn't that nice.) Analyst has a lot of other point and click menus for analyses that you can try out too.

Contingency Table Tests: For a contingency table analysis you use **proc freq** in SAS. This procedure is useful for getting distributions of single categorical variables but will also produce complicated cross-tabs and tests that go with them. We just want a 2x2 table with a chi-squared test. The basic commands for this are as follows, illustrated using the **asthma** and **urban** variables of warm-up problem 5:

```
proc freq data = tmp1.hw1;
table asthma*urban/chisq;
run;
```

SAS gives you the contingency table with row and column totals, observed counts, expected counts, row percentages, column percentages and a whole bunch of test statistics. The first one listed is the standard Pearson chi-squared statistic. The second is the likelihood ratio chi-squared statistic which is what you would get if you ran things as a logistic regression.

Logistic Regression: To fit a logistic regression in SAS you can either use **proc logistic** or **proc genmod**. The former, which we will use for this week, is specific to logistic regression. The latter (which we will see more of later) allows you to choose distributions and link functions more generally and allows a variety of data structures. The syntax for **proc logistic** is very much like **proc reg** or **proc glm**. You specify your data set, the class (i.e. categorical) variables (which can now includes your outcome though this is optional–SAS knows if you are doing a logistic model that the outcome is categorical), and a model statement connecting the outcome to the predictors. You can also have contrast or test statements that perform additional tests about your parameters. **Very Important Note:** SAS proc logistic for some reason defaults to modeling the probability that your outcome is 0 rather than 1. To fix this you need to tell it to treat your outcome in **descending order**, so that the outcome value 1 is the most important. Note the **desc** option in the data statement below.

Below is the code for the logistic model in warm-up problem 5 using all 6 predictor variables with whether or not the child gets asthma as the outcome. I also included both a test statement and a contrast for testing simultaneously whether the 4 family characteristic variables are significant or not (i.e. whether all their coefficients are 0). This is the test requested in, for example, warm-up problem 5(g). The test statement is a bit more intuitive. It has a name followed by a colon and a mathematical expression representing what you want to test. Note that to get it to work in this simple fashion I treated all the predictors as a numeric variables–SAS's coding of categorical variables is a bit tricky! As long as your categorical variables just have two levels coded as 0's and 1's this works fine. Contrasts are one of the trickier things in SAS. You need one contrast statement for each test. If your contrast involves multiple parts/degrees of freedom you need one line for each part, separated by commas (e.g. $\beta_3 = \beta_4 = \beta_5 = \beta_6 = 0$ really is 4 tests at once). The first line includes the title for the contrast (in quotes). Each line must contain the variable name(s) and constants to indicate what combination of their coefficients is being tested. For a continuous variable there is just one coefficient but for multi-level categorical variables there are several corresponding to the dummy variables you would use to code the different levels. We will discuss that more throughout the course.

```
proc logistic data = tmp1.hw1 desc;
model asthma = urban ses pollution breastfed famhist gender;
famchartest: test ses=breastfed=famhist=gender=0;
contrast "fam characteristics" ses 1,
                               breastfed 1,
                               famhist 1,
                               gender 1;
run;
```

SAS includes in its output both the coefficient estimates with their Wald standard errors, chi-squared statistics (the square of the Z statistics shown by STATA) and the corresponding p-values. It also gives the odds ratios and their confidence intervals and several global tests for the overall model. One final thing to note: SAS does a good job of handling multi-level categorical variables–it will automatically give you the multiple degree of freedom test for such a variable without your having to do a separate contrast. For example, in turn-in Problem 6i you can use the **agegroup** variable and the overall 2 degree of freedom test for it will be included in SAS' "Analysis of Effects" table. You would need to add a class statement before your model statement, namely

class agegroup;

Generating New Variables: In SAS you can use a **data** statement to create new variables from variables in an existing data set. For instance, the weight variable in turn-in problem 7 is given in pounds. If we wanted to create a new variable, wtounces, that converted it to ounces and add it to the existing data set we would use the following syntax:

```
data tmp1.hw1; set tmp1.hw1;
wtounces= 16*weight;
run;
```

The expressions can be arbitrarily complicated. Below I give the expressions needed for calculating predicted probabilities for Optional Bonus Problem 7h. To create a predicted probability in a logistic regression you need to use the formula

$$p = \frac{e^{X\beta}}{1 + e^{X\beta}}$$

The problem assumes a logistic model has been fit using the variables age, cholesterol level and family history. We want to get predicted probabilities for all the cholesterol levels in the data set (i) for people with a family history and (ii) without a family history, assuming the age is fixed at 50. In SAS the function for exponentiating is abbreivated **exp**. Make sure that you understand my equation and that it matches the output from your fitted model:

```
data tmp1.hw1; set tmp1.hw1;

predprobfamhist = exp(-8 -.062*50 + .043*cholesterol+
1.927)/(1+exp(-8 -.062*50 + .043*cholesterol + 1.927));

predprobnofamhist = exp(-8 -.062*50 +
.043*cholesterol)/(1+exp(-8 -.062*50 + .043*cholesterol));

run;
```

There are slicker ways to do this using SAS's built in prediction options as well.