donner.sas: Explanation of code

Goals of code:

- Fitting a logistic regression
- Fitting a logistic regression with a factor predictor

The data are in donner.csv.

Fitting a logistic regression: `proc logistic`
The quantity to be modeled is the log odds of survival. In the Donner party data set, survival = 1 means an individual survived, so defining the event as survival = 1 results in a model for the log odds of survival. Backtransforming this to a probability gives you the probability of survival. If you redefined the event as survival = 0, then you would model the log odds of death and the backtransformation gives you the probability of dying.

A logistic regression is fit using either proc logistic or proc genmod. Both are illustrated in the code. Both have the same core functions and syntax. They differ in some of the options. Proc logistic only fits binomial responses. Proc genmod is a more general procedure for any generalized linear model. The use of proc genmod to fit a Poisson regression to count data is illustrated in matings.sas.

The syntax should look familiar: a model statement defines the response variable (before the = sign) and model terms (after the = sign). You can add class statements before the model statement to define grouping variables.

The only “watch-out” issue is which log odds is being modeled (of survival, 1, or death, 0). By default, both proc logistic and proc genmod model the log odds of the **earlier** category. Earlier means smaller number or earlier in an alphabetic sort order. So the default will be to model the log odds (and hence probability) of death (0 or No). 0 is less than 1; No is before Yes.

SAS tells you which outcome is being modeled (middle of first page of results) and gives you a warning in the log file. If you want to change this (and you probably do), either specify (event='XX') after the response variable on the model line.

The output from proc logistic includes, in order of appearance:
  a summary of the data and each group (if there is a class statement),
  model fit statistics including AIC and BIC.
    SC (Schwarz Criterion) is proc logistic's name for what we call BIC
  the overall test of all model components = 0,
    in the model with age and fem, this hypothesis has 2 components
  Estimated odds ratios for a 1 unit change in each model variable

or the difference between two groups if a class statement
and a table of association between predicted and observed outcomes.

You should recognize most of the output, except for the last table (association). We have not talked about any of these measures. They are more commonly used in social science statistics.

When you use a class statement, proc logistic tells you how it codes the indicator variable. That is in the Class level information table. You see 1 for F and -1 for M. This is not the same as proc glm (0 and 1 coding). This is why the regression coefficient for femc (using a class statement) is half of that for fem (using hand-coded 0 or 1). The odds ratio for fem or femc (= 4.94) is the same either way. SAS realizes that you want the difference between groups when there is a class statement, so the choice of coding is irrelevant.

Proc logistic doesn't care whether the response variable is 0 or 1, Yes or No, or anything else. Either you specify the event or it chooses the "first" level.

Predictions: `output`
There are two commonly-used types of predictions from a logistic regression. One is the linear predictor: $\beta_0 + \beta_1 X_1 + \cdots$. For a logistic regression model, this predicts the log odds for an observation. The second is the predicted probability (the response). This is the value of the linear predictor back-transformed to the response scale. Both have their uses.

You create a new data set with the predictions using an output statement. `out=` names the new data set. This will include all variables in the original data set and all the variables that you request using keyword=name pairs. `xbeta=` gives you the linear predictor values; `predicted=` gives you predicted probabilities.

Plotting predicted values on a grid of potential X variables: `data new`
There are plot options to provide this plot as part of proc logistic. Look in the SAS help for the proc logistic plots option. This generally produces plots of everything you might want and a lot more. Here's how to save those predicted values so you can plot just what you want.

The approach is the same as what we used with multiple regression: create a new data set with the prediction points but no response variable, concatenate the two data sets, fit the model and save all the predictions.

The `data new;` creates a data set with the prediction points. There are two factors in the donner analysis, so there are two do loops. The output statement writes an observation to the data set. The log window shows you that new has 102 observations and 2 variables. The `data donner2;` concatenates the donner and new data sets. Then proc logistic with an output statement creates preds, a third data set, with the predictions.

Plotting multiple lines on one graph: `proc sgplot / group=`
Proc sgplot will draw all sorts of graphs. We've used scatter to plot points. series draws lines connecting one observation to the next. When X values are closely spaced, series produces the

curve. Here, we created age 15, 16, 17, $\cdots$, 65, which are sufficiently closely spaced that you really don't see the individual line segments.

If we just said `series x=age y=prob;`, we would get one line using all the observations. Adding /group = fem tells sgplot that the data set has two groups and to plot those lines separately. Adding /group=fem; to scatter uses different colored points to indicate female or male.

If you use all the data points in donner2 to draw the probability curve, you see some "extra" lines. Those are from the observations in the original data set, which are not finely spaced and not sorted by increasing age. The new data set can be distinguished from the original observations because survival is missing for all values in the new data set. `where survival=.;` only uses the observations with missing survival, so those in the new data set.