meat1.sas: Explanation of code

Goals of code:

- Example of reading data included with the SAS code
- Fitting a regression line
- Estimating mean Y at a specified X

Reading data saved with the SAS code: `cards;`
The data set can be saved as part of the data step. This can be very handy when you want to read a small data set. I discourage it when the data set is large (It is no fun to help debug a file of SAS commands that includes 200+ lines of data).

To save the data with the SAS program, write a data step to read that data. Omit the `infile` statement. After the `input` statement and any data manipulation commands, includes the `cards;` statement followed by lines of data. The `datalines;` command is equivalent to `cards;`. Notice that `cards;` ends with a semi-colon. Subsequent lines are considered to be data lines until the next `;`. My practice is to put a semicolon on a line by itself, so the end of the data is clearly marked. The `run;` tells SAS to execute the code that creates the data set.

In this problem, the desired regression equation uses X = log(time), so the variable logtime is created with the log transformed time.

Fitting a regression line: `proc glm; model ph = logtime;`
Many SAS proc will fit a regression line. We will consider two, `glm` and `reg`. Each provides slightly different functionality. `proc glm` makes it easy to print out estimates of the predicted mean for any X value.

To fit a regression with proc glm, the desired regression model goes on the model statement. As with fitting an ANOVA model, the response variable (Y) goes on the left of the = and the predictor variable (X) goes on the right. You do not X in a class statement. If you include `class ph`, each unique ph value is used to define a group and proc glm will fit a model with a separate mean for each ph. Here, we want to fit a regression line. No class statement.

The output from proc glm includes a block of results with names: Parameter Estimate .... These are the fitted regression coefficients. The intercept is \( \beta_0 \). The logtime row is the regression slope, \( \beta_1 \). It is labelled by the name of the X variable, which simplifies understanding the output when there are more than one X variable.

The values in the table are the estimate, its standard error, the T statistic testing H0: parameter = 0, and the p-value for that test. The test of the intercept is usually not very interesting (ph 0 is seriously bad), but the slope test is almost always very interesting.
Estimating mean Y at a specified X: estimate 'label' intercept 1 logtime 1.6094;
The predicted pH at 5 hours is the predicted value from the regression line when logtime = log(5) = 1.6094. That can be calculated using an estimate statement. That prediction is:

\[ \hat{pH} = \hat{\beta}_0 + 1.6094 \hat{\beta}_1 \]

The estimate statement estimates this. We need the intercept multiplied by 1 plus the logtime slope multiplied by 1.6094. That is written as: intercept 1 logtime 1.6094. As with earlier uses of estimate, the estimate statement has a label in quotes followed by the description of the quantity to estimate.