meat1.sas: Explanation of code

Goals of code:

- Fitting a regression line
- Estimating mean Y at a specified X

**Fitting a regression line: lm()**
The `lm()` function is the workhorse R model fitting function. When the X variable (right hand side of the `~` is not a factor, `lm()` fits a regression line.

The command `meat.lm <- lm(ph~logtime, data=meat)` will fit a regression model to predict y = ph from x = logtime, where both variables are in the data frame called meat. The result is stored in the meat.lm object.

If you print the object (meat.lm), you see what was done to create meat.lm and the estimated coefficients. Intercept is the intercept ($\beta_0$) and logtime is the regression slope ($\beta_1$).

If you use `summary()` on the output object, you get a lot more information. I don’t find the first block of results (a five number summary of the residuals) very useful. The second block of results gives you the estimated coefficients, their se’s, T values for the test of parameter = 0 and the associate p-value for each parameter. Again, the slope is labelled by the name of the associated X variable.

The residual se is the pooled sd ($\sqrt{\text{MSE}}$). The rest of the numbers can be ignored or will be discussed later.

The `predict()` and `resid()` functions can be used to extract predicted values for X values in the data set used to fit the regression. This gives you the residual plot that we’ve used to assess assumptions.

**Estimating mean Y for a specified X: predict() with a new.data= argument**
The `predict()` function can provide predicted values and associated information for X values not in the data set. This is done by providing a new data frame with the desired X values. That new data frame is specified by the `new.data= argument. This mush be a data frame (you can’t just say: new.data=log(5). Sorry.

The `data.frame()` function creates a data frame from a set of values. In this case, we want to predict for time = 5 hours, so logtime is log(5).

This data frame **must** include a column with the same variable name as that in the data frame used to fit the model. The `logtime= piece specifies the variable name. The `log(5)` specifies the value. That data frame is printed, then used as the value of the `new.data= argument. The result
of predict() is the predicted value. As always, this can be stored in a variable or in a data frame if you want to save the result.

Additional arguments to predict() request additional results. se.fit=T requests standard errors for the predicted mean; interval='confidence' requests a confidence interval for the predicted mean.