`stack loss` — An Example of a Robust Linear Regression Model

Consider the famous "stack loss" data from Brownlee (1965). The data are levels of four variables from 21 days of operation of a plant producing nitric acid. We are interested in predicting variable stack loss (y) as a function of the variables air flow (x_1), cooling water inlet temperature (x_2), and acid concentration (x_3). We perform Bayesian fitting using two regression models. Denote the data for the ith day of operation as $(y_i, x_{i1}, ..., x_{i3})$. In each model, we assume that

$$y_i = \beta_0 + x_{1i}\beta_1 + x_{2i}\beta_2 + x_{3i}\beta_3 + \epsilon_i.$$

where the random errors $\{\epsilon_i\}$ are distributed from a density f symmetric about 0. In the **normal error** model, we assume that f is normally distributed with mean 0 and unknown standard deviation σ. In the **t(ν) error** model, we assume that f has a t density with unknown scale parameter σ and known degrees of freedom ν. In each model, the unknown parameters are the regression vector $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^T$ and the scale parameter σ. We assign (β, σ) the usual noninformative prior proportional to $1/\sigma$.

The data is stored as two files `stack_y` and `stack_y`. The first file contains the column vector of values of the response y and the second file contains the covariate matrix (including the constant term) x. A portion of this data is displayed below:

```
stack_y =
    42
    37
    37
    28
   ...
    15
    15
stack_x =
     1     80     27     89
     1     80     27     88
     1     75     25     90
     1     62     24     87
   ...
     1     56     20     82
     1     70     20     91
```

After the data has been loaded into Matlab, the normal error regression fit can be made using the function `bay_reg`. For this example, the command is given by

```
[mb,ms]=bay_reg(stack_y,stack_x);
```

This command has two inputs — the column vector `stack_y` containing the response variable and the covariate matrix `stack_x`. This function will produce a simulated sample (of size 1000 by default) from the joint posterior distribution of (β, σ). The matrix `mb` contains the simulated values of β where each row corresponds to a simulated value. The matrix `ms` is a column vector continuing the simulated values of σ.

The t-error regression model is fit using the function `rreg`. If we wish to fit a t model with 10 degrees of freedom for this data, the command is given by

```
[MB,MS,LAM]=rreg(stack_y,stack_x,10);
```

An approximate simulated sample from the joint posterior distribution of (β, σ) is found by means of a Gibbs sampling scheme. This algorithm is facilitated by representing the t sampling model as a scale mixture of normal densities. Specifically, the statement that y_i is distributed t with location $x_i\beta$, scale σ and ν degrees of freedom is equivalent to the marginal distribution of y_i under the mixture model:

1. y_i distributed $\mathrm{N}(x_i\beta, \sigma^2/\lambda_i)$, where $x_i = (x_{i0}x_{i1}x_{i2}x_{i3})$.
2. λ_i distributed according to the density proportional to $\lambda_i^{\nu/2-1}\exp\{-\lambda_i\nu/2\}$

In the Gibbs sampler, it is convenient to simulate from the joint posterior distribution of $(\beta, \sigma, \{\lambda_i\})$. The new scale parameters $\{\lambda_i\}$ are useful in interpreting the results of the t error fit. Each λ_i can be interpreted as the weight of that particular observation in the fitting procedure. Observations that are outlying or inconsistent with the regression structure of the majority of the data will receive small posterior weights. This model will, in effect, identify potential unusual observations and downweight the contribution of these outliers in the fit.

There are three outputs to `rreg` in the example above: the matrix `MB` of simulated values of β, the vector `MS` of simulated values of the scale parameter σ, and the vector `LAM` of posterior means of the individual scale parameters $\{\lambda_o\}$.

Both regression models were fit to the stack loss data set. We summarize the fits by computing the posterior means and standard deviations of the simulated values of β below. Note there are significant differences between the normal and t values, indicating the sensitivity of the fit to the error distribution.

```
% NORMAL REGRESSION ESTIMATES
 mean(mb),std(mb)
ans =
  -39.8161    0.7214    1.2828   -0.1546
ans =
   12.1206    0.1451    0.4005    0.1579

% T(10) REGRESSION ESTIMATES
 mean(MB),std(MB)
ans =
  -40.4371    0.7770    1.0925   -0.1389
ans =
   11.6942    0.1480    0.3973    0.1553
```

To evaluate the goodness of a fit (normal or t), one can plot the posterior means of the fitted values $\{x_i\beta\}$ against the posterior means of the residuals $\{y_i - x_i\beta\}$. The figure below shows two plots; the top plot corresponds to the normal error model and the bottom to the $t(10)$ error model. The t residual graph seems slightly preferable. There appears to be a decreasing pattern in the normal error residuals and the t model seems to highlight the unusally small residual corresponding to a fitted value of 22. If one inspects the values of the posterior means of the $\{\lambda_i\}$, we note that observation 21 stands out as extreme with a posterior mean of .6.

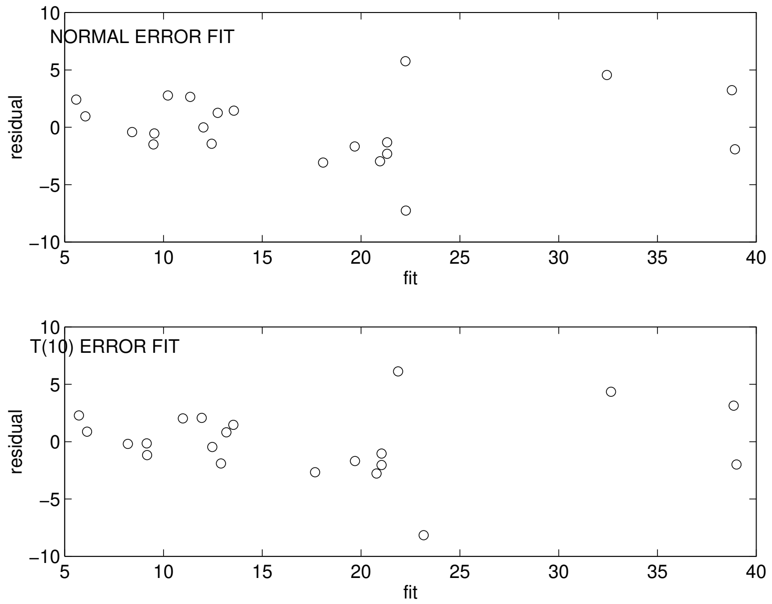

Figure 1: