```
// http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/Vol1.pdf
// Page 3: Rats liveweight example modified for cows daily milk yield
// Fitting wilmink curves, with priors for coefficients plus attempting to add priors on:
// i) cow milk yield (which we don't actually have, so are estimated), then
// ii) replace with herd milk yield (which we do have)
// Cow milk yield is the integral of the estimated equation to day 245
// Herd milk yield is the sum of the cow milk yield for cows in herd h in 1..H
// Notes:
// Not stable if -0.05 exponential factor is made a parameter
// In generated quantities, attempting to solve derivative = 0 to estimate day for peak production fails:
// (when coefficient values take on the wrong sign?)
// comment out parameters if not estimated

data {
  int<lower=0> N; //number of cows
  int<lower=0> T; //observations per cow
  int<lower=0> H; //number of herds
  real x[T];   //vector, 7 to 245 in steps of 7, day of lactation for each observation
  real y[N,T]; //'cows.txt'cow by date, milk yield observations
//  real v[N];   //cow annual milk yield  //deprecated in favour of vector math that works
  vector[N] v;   //'cow_total.txt' cow annual milk yield (which we won't normally have)
  real z[H];   //'herd_total.txt', herd annual milk yield (we do have)
  real w[H,N];  //'herd_to_cow.txt' matrix that allocates N cows to H herds

}
parameters {
  real alpha[N]; //parameters of wilmink
  real beta[N];
  real chi[N];  //added

  real mu_alpha;
  real mu_beta;          // beta.c in original bugs model
  real mu_chi;  //added
  // should i specify sign? (i.e. lower, upper). Does it matter if I don't change priors to be consistent?

  real<lower=0> sigmasq_y;
  real<lower=0> sigmasq_alpha;
  real<lower=0> sigmasq_beta;
  real<lower=0> sigmasq_chi;   //added
//  real<lower=0> sigmasq_cow_error;
}
transformed parameters {
  real<lower=0> sigma_y;       // sigma in original bugs model
  real<lower=0> sigma_alpha;
  real<lower=0> sigma_beta;
  real<lower=0> sigma_chi;      //added
  real<lower=0> sigma_cow_error;

  sigma_y = sqrt(sigmasq_y);
  sigma_alpha = sqrt(sigmasq_alpha);
  sigma_beta = sqrt(sigmasq_beta);
  sigma_chi = sqrt(sigmasq_chi);
//  sigma_cow_error = sqrt(sigmasq_cow_error);
}
model {
  mu_alpha ~ normal(35, 5);
  mu_beta ~ normal(-10, 8);
  mu_chi ~ normal(-0.07, 0.04);   //added
  sigmasq_y ~ inv_gamma(0.001, 0.001);
  sigmasq_alpha ~ inv_gamma(0.001, 0.001);
  sigmasq_beta ~ inv_gamma(0.001, 0.001);
  sigmasq_chi ~ inv_gamma(0.001, 0.001);   //added
//  sigmasq_cow_error ~ inv_gamma(0.001, 0.001);   //added
  alpha ~ normal(mu_alpha, sigma_alpha); // vectorized
  beta ~ normal(mu_beta, sigma_beta);  // vectorized
  chi ~ normal(mu_chi, sigma_chi);    //added
  for (n in 1:N)
    for (t in 1:T)
      y[n,t] ~ normal(alpha[n] + beta[n] * exp(-0.05 * x[t]) + chi[n] * x[t], sigma_y); //edited, now wilmink
//      v[n] ~ normal(to_vector(alpha) * 245 - 0.05 * to_vector(beta) * exp(-0.05 * 245) + 0.5 * to_vector(chi) * 245
* 245, sigma_cow_error);      //could be neater? but works anyway
}
generated quantities {
  real alpha0_mu;      // mean intercept for y at t=0
  vector[N] alpha0_n;  // per cow intercept for y at t=0
  real cumul_y_est;    // cumulative y (to t=245 days) for curve with coefficient mean values
  vector[N] cumul_y;   // cumulative y per cow
  vector[N] cow_error; // difference in cow total vs fit
  vector[H] herd_error; // difference in herd total vs fit
  alpha0_mu = mu_alpha + mu_beta; //edited
  alpha0_n = to_vector(alpha) + to_vector(beta);
  cumul_y_est = mu_alpha * 245 - 0.05 * mu_beta * exp(-0.05 * 245) + 0.5 * mu_chi * 245 * 245; //integral of wilmink
  cumul_y = to_vector(alpha) * 245 - 0.05 * to_vector(beta) * exp(-0.05 * 245) + 0.5 * to_vector(chi) * 245 * 245;
  cow_error = (to_vector(alpha) * 245 - 0.05 * to_vector(beta) * exp(-0.05 * 245) + 0.5 * to_vector(chi) * 245 * 245)
- to_vector(v);
```

```
  herd_error =  to_matrix(w) * (to_vector(alpha) * 245 - 0.05 * to_vector(beta) * exp(-0.05 * 245) + 0.5 *
to_vector(chi) * 245 * 245) - to_vector(z);    //i.e. difference between actual (z) and to_matrix(w) * cumul_y;

}
```