

Statistical model: Notation

I hope we could avoid using

- data
- model
- variable
- parameter

at least not without any adjectives

Instead some suggestions:

- **predictor** for driving data/variable & **assumed parameter**
- **state** for target variable
- **observed state** for target variable data
- **integrated result** for ODE solution of target variable

Red in stan block: see 8

10.7 Mathematical notation and statistical inference

When illustrating specific examples, it helps to use descriptive variable names. In more general theory and data manipulations, however, we shall adopt generic notation. This section introduces this notation and discusses the stochastic aspect of the model.

Predictors

We use the term *predictors* for the columns in the X matrix (other than the constant term). We also sometimes use the term when we want to emphasize the information that the predictors contain. For example, consider the model that includes the interaction of maternal education and maternal IQ:

$$\text{kid_score} = 58 + 16 * \text{mom_hs} + 0.5 * \text{mom_iq} - 0.2 * \text{mom_hs} * \text{mom_iq}$$

y can be
multi dimensional (S I R
population)

10.7. MATHEMATICAL NOTATION AND STATISTICAL INFERENCE

1.4	1	0.69	-1	-0.69	0.5	2.6	0.31
1.8	1	1.85	1	1.85	1.94	2.71	3.18
0.3	1	3.83	1	3.83	2.23	2.53	3.81
1.5	1	0.5	-1	-0.5	1.85	2.5	1.73
2.0	1	2.29	-1	-2.29	2.99	3.26	2.51
2.3	1	1.62	1	1.62	0.51	0.77	1.01
0.2	1	2.29	-1	-2.29	1.57	1.8	2.44
0.9	1	1.8	1	1.8	3.72	1.1	1.32
1.8	1	1.22	1	1.22	1.13	1.05	2.66
1.8	1	0.92	-1	-0.92	2.29	2.2	2.95
0.2	1	1.7	1	1.7	0.12	0.17	2.86
2.3	1	1.46	-1	-1.46	2.28	2.4	2.04
-0.3	1	4.3	1	4.3	2.3	1.87	0.48
0.4	1	3.64	-1	-3.64	1.9	1.13	0.51
1.5	1	2.27	1	2.27	0.47	3.04	3.12
?	1	1.63	-1	-1.63	0.84	2.35	1.25
?	1	0.65	-1	-0.65	2.08	1.26	2.3
?	1	1.83	-1	-1.83	1.84	1.58	2.99
?	1	2.58	1	2.58	2.03	1.8	1.39
?	1	0.07	-1	-0.07	2.1	2.32	1.27

Figure 10.8 Notation for regression modeling. The model is fit to the observed outcomes (the first 15 rows of the table). As described in the text, the model can then be applied to predict unobserved outcomes (the last 5 rows of the table, given predictors on new data X).

This regression has three *predictors*: maternal high school, maternal IQ, and maternal IQ. Depending on context, the constant term is also sometimes called a *predictor*.

Stan syntax: Block

data:

- `init_outcomes` (`y0`)
- predictors (`N`)
- **observed outcome** (cases)
- time index (`t0`, `ts`)

tf data: **predictors**

param:

- **coefficient** parameter (`beta`, `gamma`)
- prior parameter (=hyper parameter)
- measurement scale parameter (`phi`)

tf param:

- `outcome_dydt = f(outcome, t, theta, predictor)`
- coded as “**integrated_result** = `rk45(f, outcome, init_outcome, t, theta, predictor)`”

model:

- coefficient + prior parameter
- observed outcome ~ **integrated result**

```
functions {  
  real[] sir(real t, real[] y, real[] theta,  
             real[] x_r, int[] x_i) {  
  
    real S = y[1];  
    real I = y[2];  
    real R = y[3];  
    real N = x_i[1];  
  
    real beta = theta[1];  
    real gamma = theta[2];  
  
    real dS_dt = -beta * I * S / N;  
    real dI_dt = beta * I * S / N - gamma * I;  
    real dR_dt = gamma * I;  
  
    return {dS_dt, dI_dt, dR_dt};  
  }  
}  
data {  
  int<lower=1> n_days;  
  real y0[3];  
  real t0;  
  real ts[n_days];  
  int N;  
  int cases[n_days];  
}  
transformed data {  
  real x_r[0];  
  int x_i[1] = { N };  
}  
transformed parameters {  
  real y[n_days, 3];  
  real phi = 1. / phi_inv;  
  {  
    real theta[2];  
    theta[1] = beta;  
    theta[2] = gamma;  
  }  
  y = integrate_ode_rk45(sir, y0, t0, ts, theta, x_r, x_i);  
}  
model {  
  //priors  
  beta ~ normal(2, 1);  
  gamma ~ normal(0.4, 0.5);  
  phi_inv ~ exponential(5);  
  
  //sampling distribution  
  //col(matrix x, int n) - The n-th column of matrix x. Here  
  cases ~ neg_binomial_2(col(to_matrix(y), 2), phi);  
}  
generated quantities {  
  real R0 = beta / gamma;  
  real recovery_time = 1 / gamma;  
  real pred_cases[n_days];  
  pred_cases = neg_binomial_2_rng(col(to_matrix(y), 2), phi);  
}
```

with

- `sir`, the name of the function that returns the derivatives, `f`;
- `y0`, the initial condition;
- `t0`, the time of the initial condition;
- `ts`, the times at which we require the solution to be evaluated;
- `theta`, `x_r`, `x_i`, arguments to be passed to `f`.

GQ block for synthesis

- GQ block can be used to generate **outcome variable** which can be used as **“observed outcome”** in the following estimation step. The left is generation and the right is estimation.
- But here, the variability of assumed parameter is not considered as both use the same $X[N]$

```
data {  
  int<lower=1> N;  
  real X[N];  
}  
  
generated quantities {  
  real beta;  
  real alpha;  
  real y[N];  
  
  beta = normal_rng(0, 10);  
  alpha = normal_rng(0, 10);  
  
  for (n in 1:N)  
    y[n] = normal_rng(X[n] * beta  
+ alpha, 1.2);  
}
```

```
data {  
  int<lower=1> N;  
  vector[N] X;  
  vector[N] y;  
}  
  
parameters {  
  real beta;  
  real alpha;  
}  
  
model {  
  beta ~ normal(0, 1);  
  alpha ~ normal(0, 10);  
  
  y ~ normal(X * beta + alpha, 1.  
2);  
}
```

Prior calibration overview

