Hierarchical Models

StanCon 2024 Tutorial

Sean Pinkney
sean.pinkney@gmail.com
Managing Director at Omnicom Media Group
Stan Developer

Preliminary Info

- Basic familiarity with Stan and Stan should be setup on your machine
- Although the examples will be in R/cmdstanr you can use the language/platform you are most comfortable with
- Hierarchical Models in Stan

Agenda

- Background on hierarchical models `5 min`
- Partial pooling and reparameterizations `45 min`
- Normal hierarchical models `60 min`
  - → Example: Meta-analysis
  - → Group Exercise: Fitting a meta-analysis
- Break `10 min`
- Non-normal hierarchical models `60 min`
  - → Example: Advertising effectiveness
  - → Group Exercise: Fitting a hierarchical copula

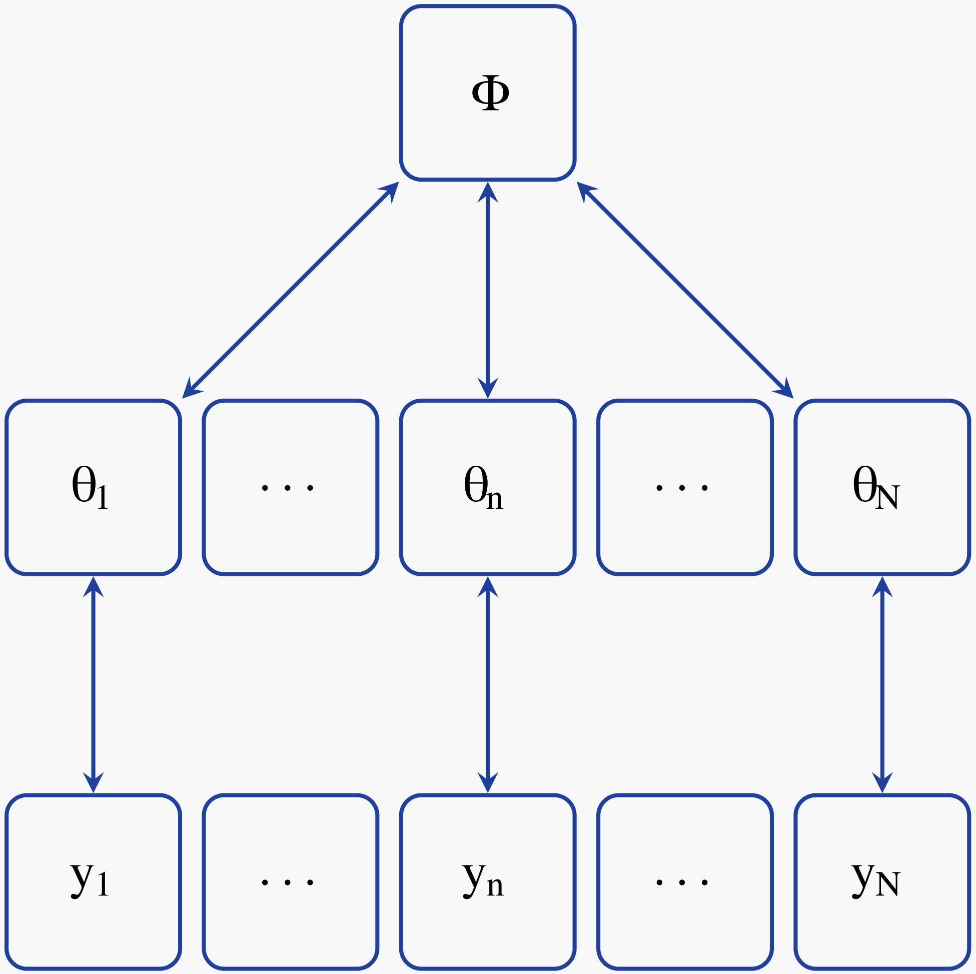
Background on hierarchical models

- The hierarchy part comes from a dependence of a parameter on another parameter
- Uses Bayes theorem (repeatedly)

$$\underbrace{p(\theta, \phi \mid y)}_{\text{Posterior}} \propto \underbrace{p(y \mid \theta, \phi)}_{\text{Likelihood}} \underbrace{p(\theta, \phi)}_{\text{Prior}} = \underbrace{p(y \mid \theta, \phi)}_{\text{Likelihood}} \underbrace{p(\theta \mid \phi)\, p(\phi)}_{\theta \text{ given } \phi}$$

- Other common terms for these models are multilevel, mixed effects, and see the Gelman blog on other common names.

Background on hierarchical models



Sharing of information happens

- Globally
- Bi-directionally
- AKA **partial pooling**

When the evidence or data for a parameter are

- low
  - → estimate is closer to prior
- large
  - → data swamps prior
  - → prior pull is weak

Question

Can you think of any issues with this type of model?

Partial pooling

`N` groups that we want to estimate separate `alpha`'s

The key insight is to have each `alpha` share a common ancestor

$$\alpha_n \sim \mathcal{N}(\mu, \sigma)$$

```
data {
  int<lower=0> N;              // number of groups
  array[N] int<lower=0> y;  // binomial counts
  array[N] int<lower=0> K;  // number of trials
}
parameters {
  real mu;                     // population mean of success log-odds
  real<lower=0> sigma;         // population sd of success log-odds
  vector[N] alpha_std;         // success log-odds
}
transformed parameters {
  vector[N] alpha = mu + sigma * alpha_std;
}
model {
  mu ~ normal(-1, 1);
  sigma ~ std_normal();
  alpha_std ~ std_normal();
  y ~ binomial_logit(K, alpha);
}
generated quantities {
  vector[N] phi = inv_logit(alpha);
}
```

Partial pooling

The intention is to have `alpha` as

$$\alpha \sim \mathcal{N}(\mu, \sigma)$$

but it is coded in a peculiar way…

non-centered parameterization

represent α as

$$\alpha = \mu + \sigma z$$

where $z \sim \mathcal{N}(0, 1)$

```
data {
  int<lower=0> N;              // number of groups
  array[N] int<lower=0> y;     // binomial counts
  array[N] int<lower=0> K;     // number of trials
}
parameters {
  real mu;                     // population mean of success log-odds
  real<lower=0> sigma;         // population sd of success log-odds
  vector[N] alpha_std;         // success log-odds
}
transformed parameters {
  vector[N] alpha = mu + sigma * alpha_std;
}
model {
  mu ~ normal(-1, 1);
  sigma ~ std_normal();
  alpha_std ~ std_normal();
  y ~ binomial_logit(K, alpha);
}
generated quantities {
  vector[N] phi = inv_logit(alpha);
}
```

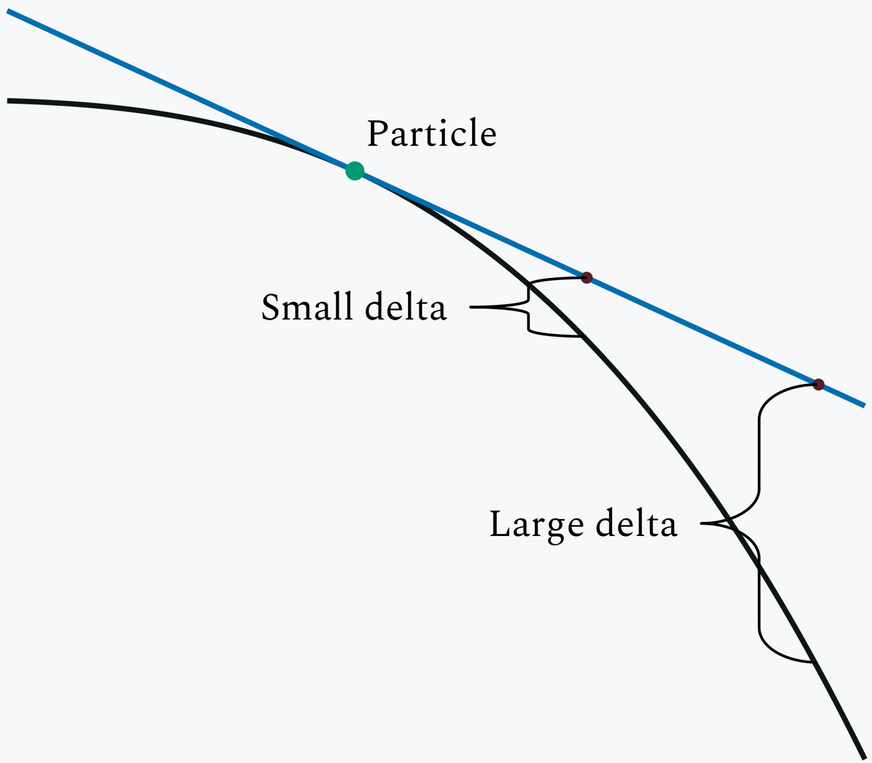
On Parameterizations

Centered and **non-centered** parameterizations are mathematically equivalent.

What is not equivalent is the ability of the estimation algorithm (i.e. HMC sampler) to explore the geometry of given model.

Stan uses a step-based gradient approximation to the posterior and a fixed step size. The expectation of the sampler is that it can move from a given point using the gradient information and the (adapted from warmup) step size.

When the curvature of the log density changes rapidly the approximation diverges - called a **divergence** - too far and this hinders the ability of the sampler to accurately measure the posterior.



Quick Math Stop

The model to the right is expressed as

$$
\begin{aligned}
p(\tau) &= \frac{1}{\sqrt{2\pi \cdot 3^2}} \exp\left(-\frac{\tau^2}{2 \cdot 3^2}\right) \\
p(\phi \mid \tau) &= \frac{1}{\sqrt{2\pi \cdot \exp(\tau/2)^2}} \exp\left(-\frac{\phi^2}{2 \cdot \exp(\tau/2)^2}\right)
\end{aligned}
$$

The joint log posterior is

$$
\begin{aligned}
p(\tau, \phi) &= p(\tau) \cdot p(\phi \mid \tau) \\
\log p(\tau, \phi) &= \log p(\tau) + \log p(\phi \mid \tau) \\
&= -\frac{\tau}{2} - \frac{\phi^2}{2\exp(\tau/2)^2} - \frac{\tau^2}{18} + C
\end{aligned}
$$

The Hessian[1] (a matrix of 2nd partial derivatives) is a 2nd order approximation to the curvature of the posterior

$$
\begin{bmatrix}
-\frac{x^2 \exp(-y)}{2} - \frac{1}{9} & x\exp(-y) \\
x\exp(-y) & -\frac{1}{\exp(y)}
\end{bmatrix}
$$

The ratio of the largest to the smallest eigenvalues of H is a gauge of posterior difficulty

```
// 1_basic_funnel.stan
parameters {
  real tau;
  real phi;
}
model {
  tau ~ normal(0, 3);
  phi ~ normal(0, exp(tau * 0.5));
}
```

Code Time

We will walk through the basic funnel code

Files we will use

```
R
   |-- 1_basic_funnel.R
stan
   |-- basic_funnel.stan
   |-- basic_funnel_repar.stan
```

Normal Parameterization Choices

Centered

$$\alpha_i \sim \mathcal{N}(\mu, \sigma) \text{ for } i \in 1, \ldots, I$$

α_i are parameterized directly by the parent distribution

Non-centered

$$z_i \sim \mathcal{N}(0, 1)$$
$$\alpha_i \stackrel{\text{set}}{=} \mu + z_i \sigma$$

α_i are reparameterized by a linear transformation because normal distributions are closed under this transformation

Mix-Centered

$$z_c \sim \mathcal{N}(0, 1) \text{ for } c \in 1, \ldots, c$$
$$\alpha_n \sim \mathcal{N}(\mu, \sigma) \text{ for } n \in c + 1, \ldots, I$$
$$\alpha_c \stackrel{\text{set}}{=} \mu + z_c \sigma$$

α_c are given centered parameterizations

α_n are given non-centered parameterizations

Partially centered

$$\chi_i \sim \mathcal{N}(\mu(1 - w_i),\ \sigma(1 - w_i) + w_i)$$
$$\alpha_i \stackrel{\text{set}}{=} \frac{(\mu w_i + \chi_i \sigma)}{\sigma(1 - w) + w_i}$$

Given χ and a weight $w \in \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$ then $\frac{(\mu w_i + \chi_i \sigma)}{\sigma(1-w)+w_i} \sim \mathcal{N}(\theta, \sigma)$

More on partially centered

When w_i is...

$w_i = 0$

$\underbrace{\alpha_i = \chi_i}_{\textbf{centered}}$

$w_i = 1$

$\underbrace{\alpha_i = \mu + \chi_i\sigma}_{\textbf{non-centered}}$

$0 < w_i < 1$

$\underbrace{\alpha_i = \frac{(\mu w_i + \chi_i\sigma)}{\sigma(1-w)+w_i}}_{\textbf{partially non-centered}}$

$$\implies \alpha_i \sim \mathcal{N}(\mu,\ \sigma)$$

Proof

$$\begin{aligned}
\chi_i &\sim \mathcal{N}[\mu(1-w_i),\ \sigma(1-w_i)+w_i] \\
\chi_i\sigma &\sim \mathcal{N}[\sigma\mu(1-w_i),\ \sigma(\sigma(1-w_i)+w_i)] \\
\mu w_i + \chi_i\sigma &\sim \mathcal{N}[\sigma\mu(1-w_i)+\mu w_i,\ \sigma(\sigma(1-w_i)+w_i)] \\
\frac{\mu w_i + \chi_i\sigma}{\sigma(1-w_i)+w_i} &\sim \mathcal{N}\left[\mu\frac{\sigma(1-w_i)+w_i}{\sigma(1-w_i)+w_i},\ \sigma\frac{\sigma(1-w_i)+w_i}{\sigma(1-w_i)+w_i}\right] \quad \blacksquare
\end{aligned}$$

Centered, Non-centered, Mixed centered, or Partially centered?

Rule of thumb

- Centered when there is enough data for your group
- Non-centered when data is low
- Mixed centered when you have both cases
- Partially centered when you have both cases

Note

The only reference to partially centered parameterizations I found was in Papaspiliopoulos and Roberts (2003) but it seems they only put the weight on μ and don't derive the implied distribution we need for our Stan model.

Code Time

We'll recreate Michael Betancourt's Hierarchcial Modeling case study and add the partially centered parameterization

```
K <- 9
N_per_indiv <- c(10, 5, 1000, 10, 1, 5, 100, 10, 5)
indiv_idx <- rep(1:K, N_per_indiv)
N <- length(indiv_idx)
sigma <- 10
```

Files

```
R
   |-- 1_hier_code.R
stan
   |-- hierarchical_cp.stan
   |-- hierarchical_ncp.stan
   |-- hierarchical_mixed.stan
   |-- hierarchical_pcp.stan
   |-- hierarchical_sim.stan
```

More on Normal Hierarchical Models

2-level, varying slopes, varying intercept model

i units and j groups

$$y_{ij} = \underbrace{\alpha + a_j}_{\text{varying intercept}} + \underbrace{X(\beta + b_j)}_{\text{varying slope}} + \epsilon_{ij}$$

The expectation of this

$$E(y_{ij} \mid X, j) = \alpha + a_j + X(\beta + \beta_j)$$

But

$$E(y_{ij} \mid X) = \alpha + X\beta$$

More on Normal Hierarchical Models

The difference between

Bayesian

$$E(y_{ij} \mid X, j) = \alpha + X\beta + \underbrace{a_j + \beta_j}_{\text{parameters}}$$

and

Frequentist

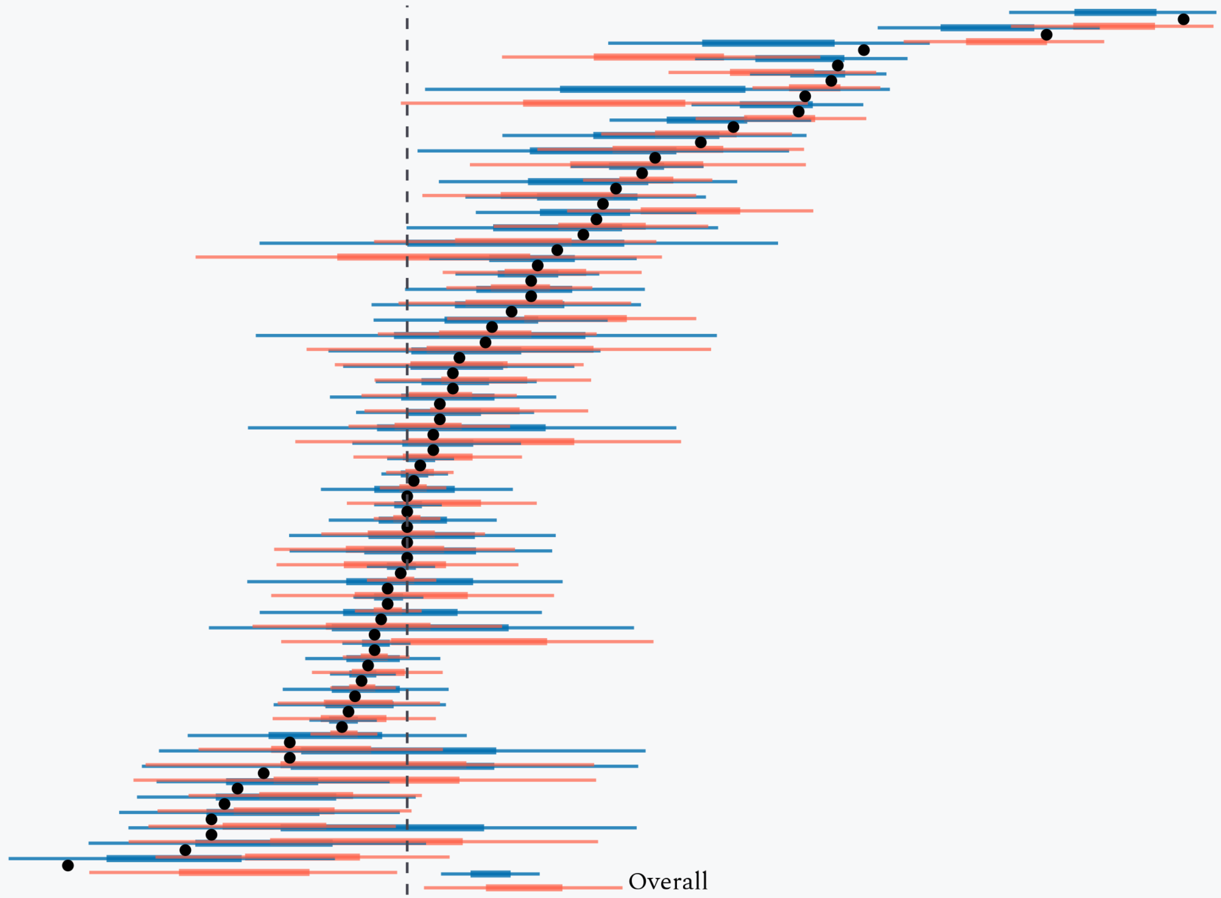
$$E(y_{ij} \mid X) = \alpha + X\beta$$

Code Time

Files

```
R
    |-- 2_hier_code.R
stan
    |-- 2_meta_two_level_cp.stan
    |-- 2_meta_two_level_ncp_reg.stan
    |-- 2_meta_three_level_ncp_reg.stan
data
    |-- meta_data.csv
```

Meta Analysis



Break

10 mins

Other Hierarchical Models

It's really not that different.

- Re-parameterizations require more care (not unique to hierarchical models)
- Exponential families (i.e. GLMs) are more-or-less straightforward
- With many modern Bayesian methods you're not limited to normality or conjugacy or exponential families

Discussion and example

You are a large advertising agency and a new client, PB&J Inc., comes to you to purchase advertising on websites for their new product.

You have data on:

- 10 different industries
- 100 different websites for 500 campaigns and 30 clients
- 5 site categories News, Shopping, Sports, Interests, Business
- Avg. seconds of attention on the ad at each website for each ad campaign
- Avg. cost of ad on each site

Generative Model

Hyperpriors

$$\mu^{h_c},\ \mu^{h_i} \overset{\text{set}}{=} 0$$
$$L_c,\ L_i \sim \mathrm{LKJ}(4)$$
$$\sigma \sim \mathrm{Exp}(1)$$

Category and Industry Parameters

$$\mu_c \sim \mathcal{N}(\mu^{h_c}, L_c)$$
$$\mu_i \sim \mathcal{N}(\mu^{h_i}, L_i)$$

Interactive effects of ad-cost by category and industry

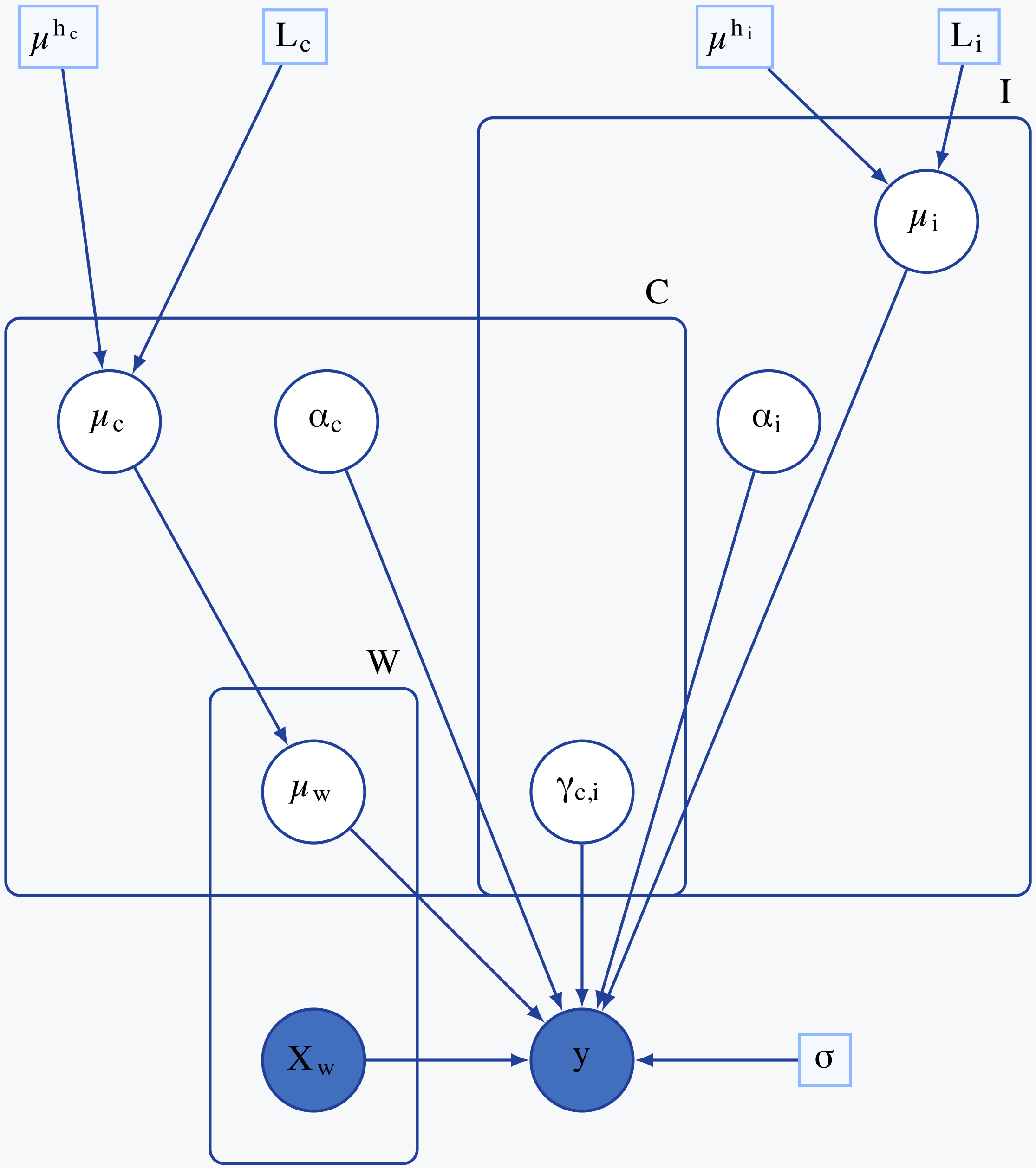
$$\alpha_c,\ \alpha_i \sim \mathcal{N}(0,1)$$
$$\gamma_{c_i} \sim \mathcal{N}(0,1)$$
$$\text{where}$$
$$\beta_{w,n} = \alpha_c X_{w,n} \alpha_i$$

Website level effect

$$\mu_w \sim \mathcal{N}(\mu_{c,w}, 1)$$

Outcome model

$$\log(y_{w,n}) \sim \mathcal{N}(\mu_w + \mu_i + \gamma_{c,i} + \beta_{w,n}, \sigma)$$



Stan Code

```
data {
  int I, C, P, N, W;
  vector[N, W] X;
  matrix[N, W] log_Y;
  array<lower=1, upper=C>[W] int index_c;
  array<lower=1, upper=I>[N] int index_i;
  int<lower=0> sim_ind;
}
parameters {
  real intercept;
  real<lower=0> sigma;

  vector[C] alpha_c;
  row_vector[I] alpha_i;
  matrix[C, I] gamma;

  vector[C] z_c;
  vector[I] z_i;
  vector[W] z_w;

  cholesky_factor_corr[C] L_c;
  cholesky_factor_corr[I] L_i;
}
transformed parameters {
  vector[C] mu_c = L_c * z_c;
  vector[I] mu_i = L_i * z_i;
  vector[W] mu_w = mu_c[index_c] + z_w;
```

Group Exercise: Fitting a hierarchical copula

```
references
    |-- hierarchical_claims_modeling.pdf
data
    |-- insurance_claims.csv
```

- Primer on Copula: Hierarchical Models in Stan
- More on Copula Modeling in Stan: Andrew Johnson's Intro to Copula Modeling

References

Brown, David E. 2014. "The Hessian Matrix: Eigenvalues, Concavity, and Curvature." BYU–Idaho Dept. of Mathematics; https://people.iith.ac.in/ashok/Maths_Lectures/TutorialB/Hessian_Examples.pdf.

Papaspiliopoulos, Omiros, and Gareth Roberts. 2003. "Non-Centered Parameterisations for Hierarchical Models and Data Augmentation." *Bayesian Statistics* 7 (January): 307–26.