HW 2

Question 1

As we saw in class, one set of nonignorable models is the latent-variable model, defined with the following set of equations:

$$
\begin{aligned}
i = 1, \dots, I, j = 1, \dots, n_i \\
y_{ij} \mid \alpha_i, \sigma^2 &\sim \text{Normal}(\alpha_i, \sigma^2) \\
\alpha_i \mid \mu, \tau^2 &\sim \text{Normal}(\mu, \tau^2) \\
m_{ij} \mid \alpha_i, \phi &\sim \text{Bernoulli}(\pi(\alpha_i, \phi)) \\
\pi(\alpha_i, \phi) &= (1 + e^{-(\phi_0 + \phi_1 \alpha_i)})^{-1}
\end{aligned}
$$

Part 1

Simulate data from this data-generating process in R for $I = 10$ groups, and $n_i = 10$, with the following values for the hyperparameters:

$$\sigma^2 = 0.5,\ \tau^2 = 1,\ \mu = 4,\ \phi_0 = -2,\ \phi_1 = 0.25$$

Make sure to set the seed so that your random sample from the data generating process is reproducible!

Part 2

Write a Stan program that corresponds to the *observable* data generating process you wrote above. In order to do so, it will help to define the observations y_k and missingness m_k, rather than y_{ij} and m_{ij} like so:

$$
\begin{aligned}
y_k \mid \alpha_i, \sigma^2 &\sim \text{Normal}(\alpha_{i[k]}, \sigma^2) \\
m_k \mid \alpha_i, \phi &\sim \text{Bernoulli}(\pi(\alpha_{i[k]}, \phi))
\end{aligned}
$$

where i is a vector that is length $N = I \times n_i$ for which the k^{th} element corresponds to the group membership of the k^{th} observation.

This means that you'll need to define the vector $y_{(0)}$ as $\{y_i \mid m_i = 0, i = 1, \ldots, n\}$. This vector will be length $N_0 = N - \sum_{i=1}^{n} m_i$.

You'll need to keep track of the following variables in your Stan program:

1. Variables related to sizes: I, N $= n_i \times I$, N_0
2. Observable variables: $y_{(0)}$, m
3. Group information: group indices corresponding to $y_{i(0)}$ and m_j

You can compute all of this information outside of Stan and pass it in as data, or you could pass in the minimum data and compute the derived quantities, like N_0 and group indices corresponding to $y_{i(0)}$, and compute these in the `transformed data` block.

Part 3

Using `cmdstanr` or https://stan-playground.flatironinstitute.org/, fit the model to the simulate data you generated above with no priors on $\sigma, \tau, \phi_1, \phi_2, \mu$. If there are warnings, report the warnings. Be sure to include the $\hat{R}$ statistics for each of your parameters, as well as the bulk effective sample sizes. These are outputted by default on stan-playgroud.

If you are using the `cmdstanr` package, you can get these values by running `fit$summary()`, which returns a data.frame that contains posterior summary statistics for each unknown parameter, as well as the $\hat{R}$ statistics and bulk effective sample sizes.

Now fit the model with the following priors, where `student_t(df, mean, scale)` is a student t with `df` degrees of freedome, mean equal to `mean` and scale parameter equal to `scale`.:

$$\begin{aligned}
\sigma &\sim \text{student_t}(3, 0, 5) \\
\tau &\sim \text{student_t}(3, 0, 2) \\
\mu &\sim \text{student_t}(3, 0, 5) \\
\phi_1 &\sim \text{student_t}(3, 0, 1) \\
\phi_0 &\sim \text{student_t}(3, 0, 1)
\end{aligned}$$

How do your results change when using these priors?

Compare the posterior means, medians, and 90% posterior intervals between the two models for all of your parameters.

Question 2

Load the `Surrogate` package in R and load the dataset `Schizo_PANSS`:

```
library(Surrogate)
data("Schizo_PANSS")
```

We encountered this dataset in the last HW set.

Subset the data to include only `Week1`, `Week2`, `Week4`, `Week6` and `Week8`:

```
hw_data <- Schizo_PANSS[,c("Id","Treat","Week1","Week2","Week4","Week6","Week8")]
hw_data_sub <- hw_data |>
  subset(Id %in% 1:200)
```

Reshape the data as we did last time.

```
long_case <-
  stats::reshape(
    hw_data_sub,
    direction = "long",
    varying = 3:7,
    sep = ""
  )[,-5]
names(long_case) <- c("Id","Treat","time","panss")
```

We're going to fit the model we developed above on this dataset, so we need to create the vector $y_{(0)}$ and m from the `long_case` data.frame.

The priors we used above won't do because the scale we have on our intercept is likely wrong. Let's use a flat prior on μ instead.

Fit the model and check that the $\hat{R}$ statistics are below 1.01 and the bulk effective sample sizes are reasonable. Provide an interpretation of the model parameters ϕ_1 and ϕ_2.

Question 3

There is covariate information that we're not using in our model from above. Modify the data generating process to:

$$
\begin{aligned}
i = 1, \ldots, I, j = 1, \ldots, n_i \\
y_{ij} \mid \alpha_i, \beta_i, \beta_{\text{treat}}, \tilde{t}_{ij}, \text{treat}_i, \sigma^2 &\sim \text{Normal}(\alpha_i + \beta_i \tilde{t}_{ij} + \beta_{\text{treat}} \text{treat}_i, \sigma^2) \\
\alpha_i \mid \mu_\alpha, \tau_\alpha^2 &\sim \text{Normal}(\mu_\alpha, \tau_\alpha^2) \\
\beta_i \mid \mu_\beta, \tau_\beta^2 &\sim \text{Normal}(\mu_\beta, \tau_\beta^2) \\
m_{ij} \mid \alpha_i, \beta_i, \text{treat}_i, \phi &\sim \text{Bernoulli}(\pi(\alpha_i, \beta_i, \text{treat}_i, \phi)) \\
\pi(\alpha_i, \phi) &= (1 + e^{-(\phi_0 + \phi_1 \alpha_i + \phi_2 \beta_i + \phi_3 \text{treat}_i)})^{-1}
\end{aligned}
$$

where $\tilde{t}_i$ is a covariate corresponding to the time of measurement, that has been scaled to have mean zero and standard deviation 1.

Part 1

Modify your data generating code to generate simulated data from this new model. Instead of generating all the groups randomly, use the data structure from the `long_case` data frame to generate your data:

```
N <- nrow(long_case)
idx_i <- long_case$Id |> as.factor() |> as.integer()
I <- max(idx_i)
treat <- long_case$Treat
tilde_t <- scale(long_case$time)
```

with the following settings for the hyperparameters:

$$\sigma^2 = 0.5,\ \tau_\alpha^2 = 1,\ \mu_\alpha = 4,\ \phi_0 = -2,\ \phi_1 = 0.25,\ \tau_\beta = 8,\ \mu_\beta = -5,\ \beta_{\text{treat}} = -0.5$$

Use reasonable priors for τ_β^2 and the extra ϕ parameters, but leave the prior for μ_β flat, μ_α, and β_{treat}.

Make sure that the $\hat{R}$ statistics look good. Show that your model recovers the parameters used to generate the data well; a suggestion would be to show that the 90% posterior intervals cover the true values.

Part 2

Using the model you wrote above, fit it to the real data. Use the same priors you used to fit the simulated data. What is your interpretation of the parameters ϕ_1, ϕ_2?

Part 3

An important part of model fitting is model diagnostics. Bayesian model diagnostics are a bit different from frequentist model diagnostics. Typically, we generate posterior predictive distributions for statistics of interest, which ideally would not be directly fitted by our model. An example would be the max predicted observation, the minimum of the predicted observations.

Write a generated quantities block where you create a new vector of length N_0 called `y_rep` that holds the model's draws for the predictive distribution for all $y_{(0)}$. In Stan this will look like:

```
generated quantities {
  vector[N_0] y_rep;
  real max_y;
  real min_y;

  for (i in 1:N_0)
    y_rep[i] = normal_rng(...,...);
  max_y = max(y_rep);
  min_y = max(y_rep);
}
```

where you'll need to fill in the ..., in the `normal_rng` function with the same form as the likelihood statement.

After writing this new Stan block, refit the model and compare the posterior distribution for `max_y`, and `min_y` to the observed values of $\max(y_{(0)})$, $\min(y_{(0)})$.

Does the model do a good job at capturing these statistics?