Model-based Inference for Causal Effects in Completely Randomized Experiments

true true true

October 2, 2019

Contents

Introduction

In this document, we discuss the implementation of Bayesian model-based inference for causal effects in **Stan**. The Bayesian inferential framework introduced by Rubin (1978) defines causal effects as a comparison of the potential outcomes of the same units and clearly separates the true underlying model of the potential outcomes from the treatment assignment mechanism. Then the framework takes advantage of fully Bayesian posterior predictive inference for multiply-imputing the missing potential outcomes.

We start by providing an introduction to the Bayesian inferential framework by analyzing a simulated dataset generated under unconfounded treatment assignment. Then we analyze an example dataset obtained from a completely randomized experiment focusing on the specification of the joint distribution of the potential outcomes. All of the source code for this case study is available on GitHub at joonho112/Bayesian-causal-inference/Case_study_1.

A simulation

Bayesian perspective on causal inference

Consider a random sample of N units, indexed by $i \in 1, ..., N$, which consist of the individuals in a randoimzed study designed to evaluate the effect of a binary treatment W on some outcome Y. Each unit has two potential outcomes: $Y_i(1)$ for unit i if the unit receives treatment ($W_i = 1$), and $Y_i(0)$ for unit i in the control condition ($W_i = 0$).

Causal inference is a missing data problem because $Y_i(1)$ and $Y_i(0)$ are never both observed. In any particular sample, only the potential outcome corresponding to the assigned treatment condition is observed and the other potential outcome is missing. Hence we can express the observed and missing potential outcomes as functions of $Y_i(0)$, $Y_i(1)$, and W_i:

$$\begin{aligned} Y_i^{obs} &= Y_i(1)W_i + Y_i(0)(1 - W_i) \\ Y_i^{mis} &= Y_i(1)(1 - W_i) + Y_i(0)W_i \end{aligned}$$

In a Bayesian perspective, the missing potential outcomes Y_i^{mis} are viewed as unobserved random variables, which are no different than unknown model parameters (Rubin 1978). Thus, the key step of the Bayesian approach to causal inference is to calculate the conditional distribution of the full vector of missing potential outcomes given the observed data (Imbens and Rubin 2015),

$$\Pr(\mathrm{Y}^{mis} | \mathrm{Y}^{obs}, \mathrm{W}).$$

This is also called the posterior predictive distribution of Y^{mis}. The posterior predictive distribution allows us to mutiply-impute the missing potential outcomes by simulation. Once we obtain this distribution, we can calculate the posterior distribution of any causal estimand of the form $\tau = \tau(\mathrm{Y}^{mis}, \mathrm{Y}^{obs}, \mathrm{W})$ where Y^{obs} and W are known.

The science and assignment mechanism

Obtaining the posterior predictive distribution of Y^{mis} requires building a model of the joint distribution of potential outcomes and assignment. Rubin (1978) showed that the joint distribution factors into two components: (1) *the assignment mechanism* which describes the process by which some potential outcomes are observed or missing, and (2) the true underlying data called *the science*. The key insignt of Rubin (1978) is that the factorization clearly separates the true unknown state of Nature which we are trying to estimate (the science) from the man-made study indicating what we do to learn about the science (the assignment mechanism).

To illustrate these two components in completely randomized experiments, let us consider a simulated data example. We assume that the true underlying model for the potential outcomes follows a bivariate normal distribution governed by a model parameter θ:

$$\begin{pmatrix} Y_i(0) \\ Y_i(1) \end{pmatrix} \Bigg| \, \theta \sim \mathsf{Normal}\left(\begin{pmatrix} \mu_c \\ \mu_t \end{pmatrix}, \begin{pmatrix} \sigma_c^2 & \rho\sigma_c\sigma_t \\ \rho\sigma_c\sigma_t & \sigma_t^2 \end{pmatrix} \right)$$

where μ_c and μ_t are α and $\alpha + \tau$ respectively, and thus the parameter vector is $\theta = (\alpha, \tau, \sigma_c, \sigma_t, \rho)$. The **R** code that follows simulates a dataset conforming to the model.

```r
# Basic setup
set.seed(654321)
N <- 500       # number of observations
alpha <- 1.0   # intercept in the Y model
tau <- 0.25    # treatment effect

# The assignment mechanism
N_t <- 200                                  # number of treated units
W <- sample(rep(c(0, 1), c(N - N_t, N_t)))  # binary treatment variable
ii_t <- which(W == 1); ii_c <- which(W == 0) # index arrays for treatment variable

# The science
mu_c <- alpha + 0*tau; sd_c <- 1   # mean and SD for the control
mu_t <- alpha + 1*tau; sd_t <- 1   # mean and SD for the treated

rho <- 0.0                                   # correlation between the errors
cov_mat <- rbind(c(sd_c^2, rho*sd_c*sd_t),
                 c(rho*sd_c*sd_t, sd_t^2))   # variance-covariance matrix

science <- mvrnorm(n = N, mu = c(mu_c, mu_t), Sigma = cov_mat, empirical = TRUE)
Y0 <- science[, 1]        # potential outcome if W = 1
Y1 <- science[, 2]        # potential outcome if W = 0
tau_unit <- Y1 - Y0       # unit-level treatment effect

# The realization of potential outcomes by the assignment mechanism
Y_obs <- Y0 * (1 - W) + Y1 * W
Y_mis <- Y0 * W + Y1 * (1 - W)
```

The array of values of `Y0` and `Y1` represents the science about which we want to learn. Particular consideration should be given to the parameter ρ representing the correlation between the two potential outcomes. Among the five parameters included in θ, ρ is the only parameter about which the observed data cannot provide empirical information because $Y_i(0)$ and $Y_i(1)$ are never both observed (See Imbens and Rubin, 2015, pp. 165-169). Thus, model-based inference requires subject-matter (scientific) knowledge and the sensible assumptions on the joint distribution of the potential outcomes. Although the simulated example assumed no dependence between the potential outcomes, this can be modified easily by assigning any values between 0 and 1 to the `rho` in the example code above.

Given the science that exists independently of the study design, the assignment mechanism determines what we are able to observe from the science. Suppose we observed the following data from the simulated example, where the values in parentheses are unobserved:

W	Y(0)	Y(1)	unit-level causal effect
1.0	(2.0)	0.9	(-1.1)
0.0	1.8	(1.2)	(-0.7)
1.0	(1.2)	0.4	(-0.7)
0.0	0.3	(2.1)	(1.8)
1.0	(1.1)	1.1	(-0.1)
0.0	-0.1	(0.5)	(0.5)

In the completely randomized experiment, a fixed number of participants is assigned to receive the treatment. Accordingly, the probabilities to be treated (i.e., the propensity scores) are equal for all units and are strictly between 0 and 1, for example, $N_t/N = 200/500 = 0.4$ in the simulated data. The probabilistic assignment mechanism is determined by the experimenter, say, $\Pr(\mathrm{W}|\mathrm{Y}(0), \mathrm{Y}(1)) = \Pr(\mathrm{W})$, which, by design, makes the assignment *unconfounded* with the potential outcomes. In the unconfounded assignment mechanism, the assignment of treatment conditions for all units is independent of all potential outcomes, observed or unobserved. We see that `W` in the simulated data was randomly drawn depending solely on the number of treated and control units.

Posterior inference under unconfounded treatment assignment

The posterior predictive distribution of missing potential outcomes is given by

$$\begin{aligned}\Pr(\mathrm{Y}^{mis}|\mathrm{Y}^{obs}, \mathrm{W}) &= \int \Pr(\mathrm{Y}^{mis}, \theta|\mathrm{Y}^{obs}, \mathrm{W})d\theta \\ &= \int \Pr(\mathrm{Y}^{mis}|\mathrm{Y}^{obs}, \mathrm{W}, \theta) \cdot \Pr(\theta|\mathrm{Y}^{obs}, \mathrm{W})d\theta.\end{aligned}$$

The joint posterior distribution of the missing data and the parameter factors into the two terms in the second integral. The first term is the sampling distribution for the replicated missing potential outcomes given parameters, treatment assignment, and observed potential coutcomes, which encapsulates the uncertainty in the imputation. The second term represents the posterior distribution of the model parameters θ. This term captures the uncertainty due to parameter estimation given the observations. The integral incorporates the two forms of uncertainty into the model by taking a weighted average of the sampling distribution with weights given by the posterior of θ.

The first term is inherently a function of the two underlying primitives, the assignment mechanism and the science (See Ding and Li (2017) pp 18-19). Given unconfounded treatment assignment, however, the assignment mechanism is *ignorable* in the imputation process. This means that causal inference under unconfoundedness requires only a model of the science but depends crucially on the joint distribution of $Y_i(0)$ and $Y_i(1)$. The model-based approach for completely randomized experiments thus starts from modeling the science to ultimately derive the posterior predictive distribution of the Y_i^{mis}.

The **Stan** program to obtain the posterior predictive distribution of the missing potential outcomes from the simulated data is provided and explained in section 2.3.2. We illustrate the following program based on how to code the two forms of uncertainty in **Stan**.

```
data {
  int<lower=0> N;                   // sample size
  vector[N] y;                      // observed outcome
  vector[N] w;                      // treatment assigned
  real<lower=-1,upper=1> rho;        // assumed correlation between the potential outcomes
}
parameters {
  real alpha;                       // intercept
  real tau;                         // super-population average treatment effect
  real<lower=0> sigma_c;            // residual SD for the control
  real<lower=0> sigma_t;            // residual SD for the treated
}
model {
   // PRIORS
   alpha ~ normal(0, 5);
   tau ~ normal(0, 5);
   sigma_c ~ normal(0, 5);
   sigma_t ~ normal(0, 5);

   // LIKELIHOOD
   y ~ normal(alpha + tau*w, sigma_t*w + sigma_c*(1 - w));
}
generated quantities{
  real tau_fs;                      // finite-sample average treatment effect
  real y0[N];                       // potential outcome if W = 0
  real y1[N];                       // potential outcome if W = 1
  real tau_unit[N];                 // unit-level treatment effect
  for(n in 1:N){
    real mu_c = alpha;
    real mu_t = alpha + tau;
    if(w[n] == 1){
      y0[n] = normal_rng(mu_c + rho*(sigma_c/sigma_t)*(y[n] - mu_t), sigma_c*sqrt(1 - rho^2));
      y1[n] = y[n];
    }else{
      y0[n] = y[n];
      y1[n] = normal_rng(mu_t + rho*(sigma_t/sigma_c)*(y[n] - mu_c), sigma_t*sqrt(1 - rho^2));
    }
    tau_unit[n] = y1[n] - y0[n];
  }
  tau_fs = mean(tau_unit);
}
```

The model specified in the **Stan** program is fit to the simulated data:

```r
# Collect data into a list format suitable for Stan
stan_data <- list(N = N, y = Y_obs, w = W, rho = 0.0)

# Compile and run the stan model
fit_simdat <- stan(file = "simulated_example.stan",
                   data = stan_data,
                   iter = 1000, chains = 4)
```

Note that the `rho` is assumed to be a fixed value rather than a parameter in the model. Because the parameters governing the association between the potential outcomes cannot be empirically estimated using the data, they need to be given *a priori*. Here we first assumed a correlation coefficient equal to zero when simulating the data.

The posterior distribution of the model parameters

To obtain the posterior distribution of the model parameters governing the science, $\Pr(\theta|\mathrm{Y}^{obs}, \mathrm{W})$, we combine the prior distribution with the likelihood function. The first part of the `model` block defines the sampling statements of the priors on α, τ, σ_c, and σ_t. Considering the fact that the simulated outcome follows a standard normal distribution, we impose weakly informative normal priors with mean 0 and standard deviation 5 on the parameters for μ_c and μ_t. For the scale parameters σ_c and σ_t, half-cauchy priors with scale set to 5 are used.

The next sampling statement in the `model` block indicates the distribution of the observed potential outcome, which specifies the likelihood of Y_i^{obs}. Conditional on the assignment vector W and parameters, the distribution of Y_i^{obs} is given by

$$\Pr(Y_i^{obs}|\mathrm{W}, \theta) \sim \mathsf{Normal}(W_i \cdot \mu_t + (1 - W_i) \cdot \mu_c, W_i\sigma_t^2 + (1 - W_i) \cdot \sigma_c^2)$$

where μ_c and μ_t are α and $\alpha + \tau$ respectively. Because we can observe only one potential outcome per person, the likelihood function $\Pr(\mathrm{Y}^{obs}, \mathrm{W}|\theta)$ remains the same irrespective of the value of ρ. The sampling statement of Y^{obs} is coded as:

```
y ~ normal(alpha + tau*w, sigma_t*w + sigma_c*(1 - w));
```

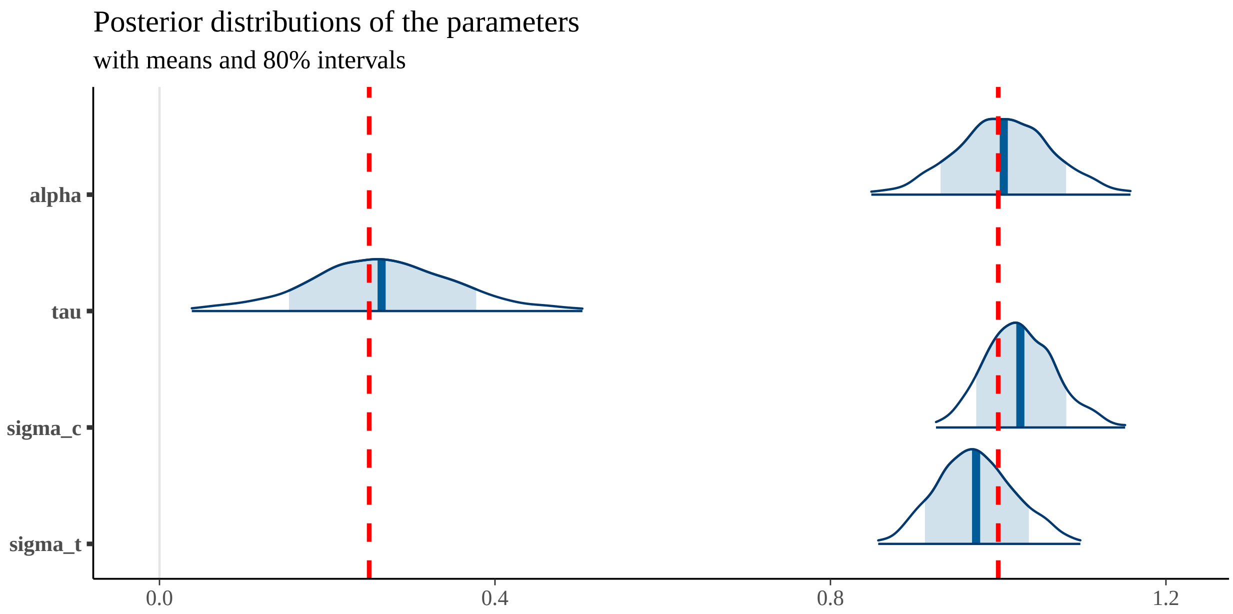
The posterior means and standard deviations of the parameters, α, τ, σ_c, and σ_t, can be displayed as follows:

```
print(fit_simdat, pars = c("alpha", "tau", "sigma_c", "sigma_t"),
      probs = c(0.1, 0.5, 0.9), digits = 3)
```

```
## Inference for Stan model: simulated_example.
## 4 chains, each with iter=1000; warmup=500; thin=1;
## post-warmup draws per chain=500, total post-warmup draws=2000.
##
##          mean se_mean    sd   10%   50%   90% n_eff  Rhat
## alpha   1.007   0.002 0.059 0.931 1.006 1.081  1498 1.002
## tau     0.265   0.002 0.089 0.154 0.263 0.378  1507 1.001
## sigma_c 1.026   0.001 0.043 0.974 1.024 1.081  1824 0.999
## sigma_t 0.974   0.001 0.048 0.913 0.972 1.037  1777 1.001
##
## Samples were drawn using NUTS(diag_e) at Fri Mar 19 10:00:34 2021.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

Before interpreting the results, it is necessary to check that the chains have converged for each parameter. The $\hat{R}$ statistics shown in the rightmost column of the model summary are all less than 1.1. Also the effective sample size estimates are sufficient for inference. Thus it seems that **Stan** has produced an adequate approximation of the posterior.

```
posterior <- as.array(fit_simdat)
mcmc_areas(posterior, pars = c("alpha", "tau", "sigma_c", "sigma_t"),
           prob = 0.8, prob_outer = 0.99, point_est = "mean" ) +
  vline_at(c(0.25, 1.00), linetype = "dashed", color = "red", size = 1) +
  ggplot2::labs(title = "Posterior distributions of the parameters",
                subtitle = "with means and 80% intervals")
```



The plot shows point estimates and 99% credible intervals for parameters along with kernel density curves of the posteriors. The uncertainty intervals shown as shaded areas under the estimated posterior density curves represents 80% credible intervals. We observe that the 80% credible intervals of `alpha`, `tau`, `sigma_c`, and `sigma_t` include the true parameters ($\tau = 0.25$, $\alpha, \sigma_c, \sigma_t = 1.0$), which indicates that the **Stan** model successfully recovers the generating values of parameters.

The posterior predictive distribution of the missing potential outcomes

With the posterior distributions of the parameters in hand, we then derive the sampling distribution for the replicated missing potential outcomes given observed outcomes, treatment assignment, and the parameters, $\Pr(\mathrm{Y}^{mis}|\mathrm{Y}^{obs}, \mathrm{W}, \theta)$. As explained above, imputing the missing potential outcomes Y^{mis} requires only a model for the science because of unconfoundedness, that is, the joint distribution of $Y_i(0)$ and $Y_i(1)$. Assuming the random variables are i.i.d. conditional on θ, the posterior distribution factors into N terms. Hence for the treated units we can impute the missing control potential outcomes from the conditional distribution of $Y_i(0)$ given $Y_i(1)$ and θ (Ding and Li 2017). For control units, we impute the missing treatment potential outcomes from $\Pr(Y_i(1)|Y_i(0), \theta)$. Because we assume that the joint distribution of the potential outcomes is bivariate normal, we obtain the conditional distribution of one potential outcome given the other as

$$\Pr(Y_i(1)|Y_i(0), \theta, W_i = 0) \sim \mathsf{Normal}\left(\mu_t + \rho \cdot \frac{\sigma_t}{\sigma_c} \cdot (Y_i(0) - \mu_c), \sigma_t^2(1 - \rho^2)\right),$$

and

$$\Pr(Y_i(0)|Y_i(1), \theta, W_i = 1) \sim \mathsf{Normal}\left(\mu_c + \rho \cdot \frac{\sigma_c}{\sigma_t} \cdot (Y_i(1) - \mu_t), \sigma_c^2(1 - \rho^2)\right).$$

where μ_c and μ_t are α and $\alpha + \tau$ respectively.

The following **Stan** code in the `generated quantities` block implements the imputations:

```
for(n in 1:N){
  real mu_c = alpha;
  real mu_t = alpha + tau;
```

```
    if(w[n] == 1){
      y0[n] = normal_rng(mu_c + rho*(sigma_c/sigma_t)*(y[n] - mu_t), sigma_c*sqrt(1 - rho^2));
      y1[n] = y[n];
    }else{
      y0[n] = y[n];
      y1[n] = normal_rng(mu_t + rho*(sigma_t/sigma_c)*(y[n] - mu_c), sigma_t*sqrt(1 - rho^2));
    }
    tau_unit[n] = y1[n] - y0[n];
  }
```

Note that assuming a correlation coefficient between $Y_i(1)$ and $Y_i(0)$ equal to 0 (`rho` $= 0$) the **Stan** code for the conditional distribution of one potential outcome given the other can be simplified to `y0[n] = normal_rng(mu_c, sigma_c)` for $Y_i(0)$ and `y1[n] = normal_rng(mu_t, sigma_t)` for $Y_i(1)$.

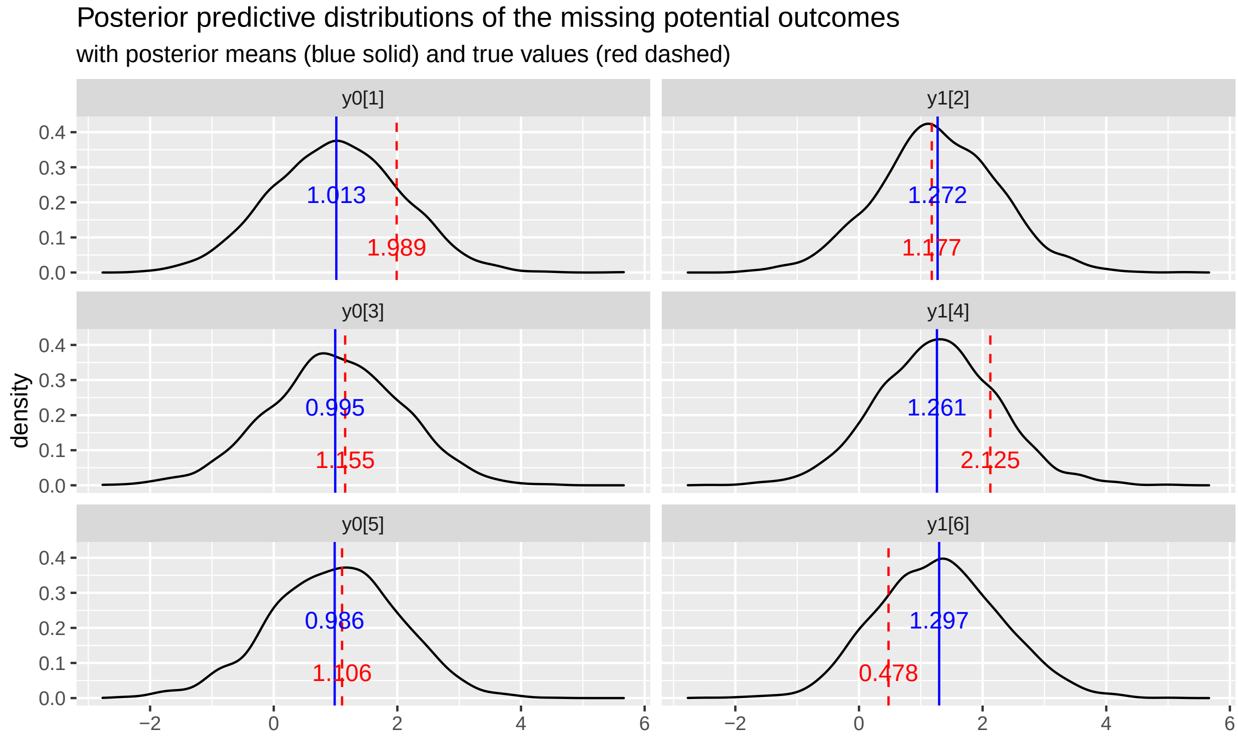
In this code, the two forms of uncertainties, $\Pr(\mathrm{Y}^{mis}|\mathrm{Y}^{obs}, \mathrm{W}, \theta)$ and $\Pr(\theta|\mathrm{Y}^{obs}, \mathrm{W})$, are combined to generate the posterior predictive distribution of Y^{mis}, $\Pr(\mathrm{Y}^{mis}|\mathrm{Y}^{obs}, \mathrm{W})$. Each of `alpha`, `tau`, `sigma_c`, and `sigma_t` represents a draw of the parameter from the posterior distribution $\Pr(\theta|\mathrm{Y}^{obs}, \mathrm{W})$. The uncertainty of parameter estimation is thus rolled into the specified parameter values. On the other hand, the uncertainty due to the imputation is reflected by sampling random numbers from each conditional distribution of the potential outcomes using the pseudorandom number generator, `normal_rng`.

The imputation procedure defined in the `generated quantities` block gives us the posterior predictive distribution of each missing potential outcome. Our model-based approach aims to impute the missing potential outcomes which are in parentheses in the true science table shown in section 2.2. We need to impute $Y(1)$ for the first two units and impute $Y(0)$s for the remaining four units.

The posterior predictive distribution of the six missing potential outcomes can be displayed as follows:

```
# Collect MCMC samples for the missing potential outcomes
mcmc <- data.frame(as.matrix(fit_simdat)) %>% dplyr:: select(contains("y"))

ymis_idx <- data.frame("y", as.numeric(W != 1), seq_along(W)) %>%
  unite("temp", 1:2, sep = "") %>% unite("ymiss", 1:2, sep = ".") %>% as.matrix()

var_idx <- names(mcmc) %>% str_sub(end = -2) %in% ymis_idx[, "ymiss"]

mcmc_ymis <- mcmc %>% dplyr::select(names(mcmc)[var_idx]) %>%
  gather(key, value, seq(500), factor_key = TRUE) %>%
  mutate(row_num = as.numeric(str_sub(key, start = 4, end = -2))) %>%
  arrange(row_num)

str_sub(mcmc_ymis$key, 3, 3) <- "["; str_sub(mcmc_ymis$key, -1, -1) <- "]"

mcmc_ymis <- mcmc_ymis %>%
   mutate(key = factor(key, levels = as.matrix(mcmc_ymis %>% distinct(key))[, "key"]))

# Generate dataframes to plot
mcmc_ymis_plot <- mcmc_ymis %>% filter(row_num <= 6)

ymis_means <- mcmc_ymis_plot %>%
  group_by(key) %>%
  summarise(mean_imputed = mean(value, na.rm = TRUE)) %>%
  mutate(mean_true = Y_mis[1:6])

# Plot the posterior predictive distributions with the true Y^mis
```

```r
ggplot(mcmc_ymis_plot, aes(x = value)) +
  geom_density(alpha = 0.1) +
  geom_vline(data = ymis_means, aes(xintercept = mean_imputed),
             linetype = "solid", color = "blue", size = 0.5) +
  geom_vline(data = ymis_means, aes(xintercept = mean_true),
             linetype = "dashed", color = "red", size = 0.5) +
  geom_text(aes(x = mean_imputed, y = 0.15, label = sprintf("%.3f", round(mean_imputed, 3))),
            hjust = 0.5, vjust = -1.0, color = "blue", data = ymis_means) +
  geom_text(aes(x = mean_true, y = 0, label = sprintf("%.3f", round(mean_true, 3))),
                hjust = 0.5, vjust = -1.0, color = "red", data = ymis_means) +
  facet_wrap( ~ key, ncol = 2) +
  labs(title = "Posterior predictive distributions of the missing potential outcomes",
       subtitle = "with posterior means (blue solid) and true values (red dashed)", x = "")
```



The plot shows that the true values of the missing potential outcomes fall in approximately at least 80% credible intervals of the posterior predictive distributions.

Finite-sample vs. super-population average treatment effects

Once we obtain the posterior predictive distributions for Y^{mis}, we can easily define the posterior distributions of any causal estimand based on averages, variances, quantiles, ratios, or intermediate outcomes. We focus primarily on the average treatment effect (ATE):

$$\tau_{\text{fs}} = \tau(\mathrm{Y}(0), \mathrm{Y}(1)) = \frac{1}{N} \cdot \sum_{i=1}^{N} (Y_i(1) - Y_i(0)) = \bar{Y}(1) - \bar{Y}(0).$$

We use the subscript fs to denote the *finite-sample* average treatment effect because the N units in the dataset are viewed as the population of interest itself. The estimand is based on comparisons between the two potential outcomes, which gives the unit-level causal effects $Y_i(1) - Y_i(0)$. The average treatment effect is obtained by simply taking the average of the unit-level causal effects over the finite sample of N units. The **Stan** code in the `generated quantities` block defines the unit-level causal effects as `tau_unit[n] = y1[n] - y0[n];` and the finite-sample average treatment effect as `tau_fs = mean(tau_unit);`.

If we instead view the N observed units as a random sample from an infinite super-population, the random sampling induces an additional source of uncertainty. Even if all the missing potential outcomes could be imputed with certainty, it would still be uncertain whether the estimated τ_{fs} is the same as the average treatment effect in the super-population from which the sample was drawn. Hence the *super-population* average treatment effect, denoted as τ_{sp}, can be defined by taking the expectation over the distribution of the τ_{fs} generated by random sampling from the super-population. Given the bivariate normal model for the simulated data governed by parameter θ, τ_{sp} is given by

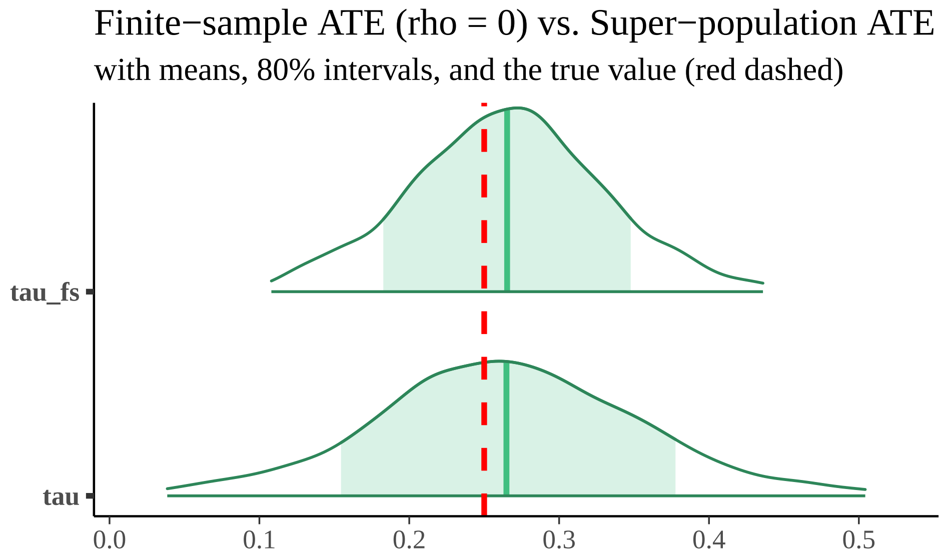
$$\tau_{\text{sp}} = \mathbb{E}_{\text{sp}}\left[\tau_{\text{fs}}\right] = \mathbb{E}_{\text{sp}}\left[\bar{Y}(1) - \bar{Y}(0)\right] = \mu_t - \mu_c = (\alpha + \tau) - \alpha = \tau.$$

Since τ_{sp} is solely a function of the model parameters θ, we can estimate τ_{sp} by simply obtaining the posterior distribution of τ. In the **Stan** code the `tau` in the `model` block represents the super-population average treatment effect. We observe from the following results that the τ_{sp} estimate is less precise than the τ_{fs} estimate assuming independence between the potential outcomes due to the additional source of uncertainty induced by random sampling from the super-population:

```r
print(fit_simdat, pars = c("tau_fs", "tau"),
      probs = c(0.1, 0.5, 0.9), digits = 3)
```

```
## Inference for Stan model: simulated_example.
## 4 chains, each with iter=1000; warmup=500; thin=1;
## post-warmup draws per chain=500, total post-warmup draws=2000.
##
##         mean se_mean    sd   10%   50%   90% n_eff  Rhat
## tau_fs 0.265   0.002 0.066 0.183 0.266 0.348  1654 1.003
## tau    0.265   0.002 0.089 0.154 0.263 0.378  1507 1.001
##
## Samples were drawn using NUTS(diag_e) at Fri Mar 19 10:00:34 2021.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```r
color_scheme_set("green")
mcmc_areas(posterior, pars = c("tau_fs", "tau"),
           prob = 0.8, prob_outer = 0.99, point_est = "mean" ) +
  vline_at(c(0.25), linetype = "dashed", color = "red", size = 1) +
  ggplot2::labs(title = "Finite-sample ATE (rho = 0) vs. Super-population ATE",
                subtitle = "with means, 80% intervals, and the true value (red dashed)")
```



For more detailed discussion on Bayesian inference for super-population and finite-population estimands, see the section 8.4 of Gelman et al. (2013).

Example application

Because observed data generally do not include empirical information about the dependence between potential outcomes, the key difficulty of the model-based inference lies in how the dependence should be modeled. Researchers often vary the specification of the model of the joint distribution of $Y_i(1)$ and $Y_i(0)$ and look at the sensitivity of conclusions to the choice for the model. Focusing on this aspect of the model-based approach, we analyze an example data obtained from a completely randomized experiment in this section. We specifically aim to replicate the analysis performed in chapter 8 of Imbens and Rubin (2015) for the illustration.

Data example: The NSW experimental job-training data

We will be analyzing the data come from a randomized evaluation of the National Supported Work (NSW) project. The project was a transitional and subsidized job training program for people with longstanding employment problems. The data has been widely analyzed in the study on program evaluation particularly in econometrics, including Heckman and Hotz (1989), Dehejia and Wahba (1999), and Smith and Todd (2001). We analyze a subset of the data used by Dehejia and Wahba (1999), which is the same data set illustrated in chapter 8 of Imbens and Rubin (2015). The dataset is available in the **Matching** R package, and consists of the following 12 variables:

- `age`: age in years
- `educ`: years of schooling
- `nodegr`: indicator variable for being a high school dropout
- `black`: indicator variable for being African American
- `hisp`: indicator variable for being Hispanic/Latino
- `married`: indicator variable for being now or ever before married
- `re74`: pre-training earnings in 1974
- `u74`: an indicator for earnings in 1974 being zero
- `re75`: pre-training earnings in 1975
- `u75`: an indicator for earnings in 1975 being zero

- re78: post-program labor market earnings in 1978
- treat: indicator variable for treatment status

```
# Use example dataset form Matching package
data(lalonde, package = "Matching")

# Change the scales of all earnings variables in thousands
lalonde$re74 <- lalonde$re74/1000
lalonde$re75 <- lalonde$re75/1000
lalonde$re78 <- lalonde$re78/1000

# Prepare outcome and treatment
y <- lalonde$re78
w <- lalonde$treat

# Display summary statistics
sumstats <- lalonde %>%
  summarise_all(funs(mean, sd, min, max)) %>%
  gather(key, value, everything()) %>%
  separate(key, into = c("variable", "stat"), sep = "_") %>%
  spread(stat, value) %>%
  dplyr::select(variable, mean, sd, min, max) %>%
  mutate_each(funs(round(., 1)), -variable)
```

```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
## Warning: `mutate_each_()` is deprecated as of dplyr 0.7.0.
## Please use `across()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
knitr::kable(sumstats, caption = "Summary statistics for the NSW data")
```

Table 2: Summary statistics for the NSW data

variable	mean	sd	min	max
age	25.4	7.1	17	55.0
black	0.8	0.4	0	1.0
educ	10.2	1.8	3	16.0
hisp	0.1	0.3	0	1.0

variable	mean	sd	min	max
married	0.2	0.4	0	1.0
nodegr	0.8	0.4	0	1.0
re74	2.1	5.4	0	39.6
re75	1.4	3.2	0	25.1
re78	5.3	6.6	0	60.3
treat	0.4	0.5	0	1.0
u74	0.7	0.4	0	1.0
u75	0.6	0.5	0	1.0

Among the sample of $N = 445$, 42% of participants were randomly assigned to the job training program ($N_t = 185$). To replicate Imbens and Rubin (2015)'s analyses, all earning variables, `re74`, `re75`, and `re78`, were converted to thousands by deviding them by 1,000. Alternatively, one may want to standardize continuous covariates `age` and log-transform all earnings variables to improve MCMC sampling.

Analyzing the example data with Stan

Model 1: Assuming independence between potential outcomes

Let us first consider the same specification applied to the simulated data above. We assumed a joint normal distribution with (1) independent potential outcomes ($\rho = 0$), (2) no covariates, and with (3) different variances in the treatment and control groups (σ_t and σ_c):

$$\begin{pmatrix} Y_i(0) \\ Y_i(1) \end{pmatrix} \Bigg| \, \theta \sim \mathsf{Normal}\left(\begin{pmatrix} \mu_c \\ \mu_t \end{pmatrix}, \begin{pmatrix} \sigma_c^2 & 0 \\ 0 & \sigma_t^2 \end{pmatrix} \right)$$

where μ_c and μ_t are α and $\alpha+\tau$ respectively. The following is the **Stan** program **Model__01__Assuming__Independent__Po** for the model 1:

```
functions {
  real quantile(vector x, real p){
    int n;            // length of vector x
    real index;       // integer index of p
    int lo;           // lower integer cap of the index
    int hi;           // higher integer cap of the index
    real h;           // generated weight between the lo and hi
    real qs;          // weighted average of x[lo] and x[hi]
    n = num_elements(x);
    index = 1 + (n - 1)*p;
    lo = 1;
    while ((lo + 1) < index)
      lo = lo + 1;
    hi = lo + 1;
    h = index - lo;
    qs = (1 - h)*sort_asc(x)[lo] + h*sort_asc(x)[hi];
    return qs;
  }
}
data {
  int<lower=0> N;                     // sample size
  vector[N] y;                        // observed outcome
  vector[N] w;                      // treatment assigned
}
parameters {
  real alpha;                       // intercept
  real tau;                         // super-population average treatment effect
  real<lower=0> sigma_c;            // residual SD for the control
  real<lower=0> sigma_t;            // residual SD for the treated
}
model {
   // PRIORS
   alpha ~ normal(0, 100);
   tau ~ normal(0, 100);
   sigma_c ~ normal(0, 100);
   sigma_t ~ normal(0, 100);

   // LIKELIHOOD
   y ~ normal(alpha + tau*w, sigma_t*w + sigma_c*(1 - w));
}
generated quantities{
  real tau_fs;                      // finite-sample average treatment effect
  real tau_qte25;                   // quantile treatment effect at p = 0.25
  real tau_qte50;                   // quantile treatment effect at p = 0.50
  real tau_qte75;                   // quantile treatment effect at p = 0.75
  real y0[N];                       // potential outcome if W = 0
  real y1[N];                       // potential outcome if W = 1
  real tau_unit[N];                 // unit-level treatment effect
  for(n in 1:N){
    real mu_c = alpha;
    real mu_t = alpha + tau;
    if(w[n] == 1){
      y0[n] = normal_rng(mu_c, sigma_c);
      y1[n] = y[n];
    }else{
      y0[n] = y[n];
      y1[n] = normal_rng(mu_t, sigma_t);
    }
    tau_unit[n] = y1[n] - y0[n];
  }
  tau_fs = mean(tau_unit);
  tau_qte25 = quantile(to_vector(y1), 0.25) - quantile(to_vector(y0), 0.25);
  tau_qte50 = quantile(to_vector(y1), 0.50) - quantile(to_vector(y0), 0.50);
  tau_qte75 = quantile(to_vector(y1), 0.75) - quantile(to_vector(y0), 0.75);
}
```

The program is basically the same as the one for the simulated data, but has two notable differences. First, we define three additional finite-sample causal estimands based on quantiles in addition to the average treatment effect (`tau_fs`), that is, quantile treatment effects (QTE) for the 0.25, 0.50, and 0.75 quantiles (`tau_qte25`, `tau_qte50`, `tau_qte75`). The QTEs are defined as the differences in the pth quantiles of the potential outcomes, $Q_p(Y_i(1)) - Q_p(Y_i(0))$, not the pth quantiles of the differences in the potential outcomes, $Q_p(Y_i(1) - Y_i(0))$. The posterior distribution of the QTEs can be easily derived given the imputed missing potential outcomes in the same manner as for the finite-sample average treatement effect. Unfortunately, a built-in function to calculate the sample quantile values for data and generated quantities is not yet available in **Stan**. Thus we need to add a `functions` block for a user-defined function equivalent to `quantile()` in **R**. In the `functions` block, all sample quantiles were defined as a weighted average of consecutive order statistics. See Hyndman and Fan (1996) for further details.

Second, the scales of the prior distributions for the two mean parameters are now set to 100 because the standard deviation of 100 is large relative to the scale of the earning variables measured in thousands of dollars which range from 0 to 60.31. We use unscaled normal priors `normal(0, 100)` for the `alpha` and `tau` for pedagogical purposes - we want to replicate the estimates in Table 8.6 of Imbens and Rubin (2015). In practice, it is better idea to put the `alpha` and `tau` on a unit scale by dividing them by the scale such as the median absolute deviation (MAD) and to use priors stronger than `normal(0, 100)` for the two mean parameters. Then it is generally easy to define weakly informative priors and to make computational algorithm much better conditioned. See a Wiki on prior choice recommendations by the Stan Development Team.

The prior distributions for σ_c and σ_t are half-normal distributions with the scale parameters are set to 100 and with a `<lower=0>` constraint in the declaration of the parameter. To replicate the Imbens and Rubin (2015) analyses, one may want to use inverse gamma distributions with the shape parameters α set to 1 and with the scale parameters β set to 0.01. The estimates are almost the same in both prior specifications with a single-level model with enough data. Note, however, that the purportedly noninformative inverse-gamma prior can affect inferences in other settings such as hierarchical models, particularly where the number of groups is small and the group-level variance is close to zero. See Gelman (2006) for more details.

The **Stan** program assumming independence between the potential outcomes is then run on the example data:

```
# Collect data into a list format suitable for Stan
stan_data <- list(N = nrow(lalonde), y = y, w = w)

# Compile and run the stan model
fit_mod1 <- stan(file = "Model_01_Assuming_Independent_Potential_Outcomes.stan",
                 data = stan_data,
                 iter = 1000, chains = 4)
```

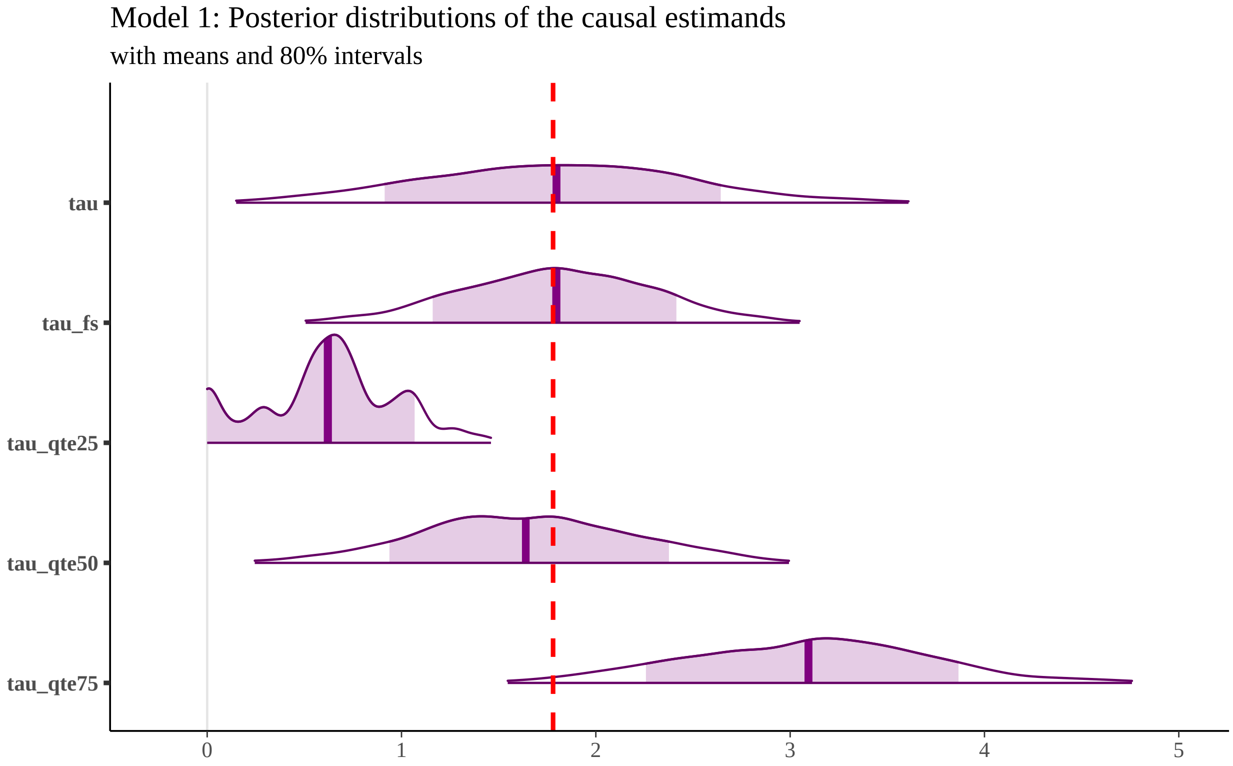
We view summaries of the parameter posteriors for each causal estimand. As discussed above, convergence of the chains is assessed for every parameter using $\hat{R}$. All $\hat{R}$ values from the model summaries are less than 1.1, which indicates that the chains have converged in the fitted model.

```
param <- c("alpha", "tau", "sigma_c", "sigma_t", "tau_fs", "tau_qte25", "tau_qte50", "tau_qte75" )
print(fit_mod1, pars = param,
      probs = c(0.1, 0.5, 0.9), digits = 3)
```

```
## Inference for Stan model: Model_01_Assuming_Independent_Potential_Outcomes.
## 4 chains, each with iter=1000; warmup=500; thin=1;
## post-warmup draws per chain=500, total post-warmup draws=2000.
##
##            mean se_mean    sd   10%   50%   90% n_eff  Rhat
## alpha     4.552   0.009 0.347 4.111 4.547 4.987  1576 1.001
## tau       1.797   0.018 0.688 0.913 1.806 2.643  1426 1.000
## sigma_c   5.513   0.005 0.242 5.203 5.508 5.830  2138 1.000
## sigma_t   7.923   0.009 0.408 7.411 7.909 8.464  2116 1.001
## tau_fs    1.796   0.011 0.495 1.161 1.800 2.415  1873 1.000
## tau_qte25 0.621   0.008 0.354 0.000 0.647 1.068  2218 0.999
## tau_qte50 1.639   0.012 0.552 0.938 1.634 2.376  2059 0.999
## tau_qte75 3.095   0.015 0.637 2.258 3.129 3.866  1816 0.999
##
## Samples were drawn using NUTS(diag_e) at Fri Mar 19 10:01:18 2021.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

The posterior means and standard deviations for treatment effects above are the same as the estimates in the second line of Table 8.6 of Imbens and Rubin (2015) with small differences.

```
posterior <- as.array(fit_mod1)
color_scheme_set("purple")
mcmc_areas(posterior, pars = c("tau", "tau_fs", "tau_qte25", "tau_qte50", "tau_qte75"),
           prob = 0.8, prob_outer = 0.99, point_est = "mean" ) +
  vline_at(c(1.78), linetype = "dashed", color = "red", size = 1) +
  ggplot2::labs(title = "Model 1: Posterior distributions of the causal estimands",
                subtitle = "with means and 80% intervals")
```



Again we observe that the posterior standard deviation of the finite-sample ATE (`tau_fs`), 0.48, is substantially lower than that of the super-population ATE (`tau`), 0.66. Cosidering the high proportion of zeros in the outcome `re78` (about 31%), it is not surprising that the posterior distribution of `tau_qte25` seems to be non-normal. For the readers who attempt to model the zero-inflated distribution of the potential outcomes, the **Stan** program for the two-part model is available on the GitHub repository.

The three QTE estimates shed some light on the treatment effect heterogeneity across the outcome distribution. From the figure above we observe that the effect of the job-training program becomes stronger for the higher conditional quantiles, indicating that the program is most benefical for the participants at the high end of the earning distribution (75%). However, these results for the QTE estimands might be sensitive to choices for the prior distribution of the dependence structure between the two potential outcomes (Imbens and Rubin 2015).

Model 2: Assuming constant unit-level treatment effects

One way to investigate the sensitivity of conclusions to the choice for the dependence between the potential outcomes is to assume the most conservative case and exploits the estimate from the case as a bound. In a mode-based inference, the most conservative case is often the situation assuming the treatment effect is constant for all units, $Y_i(1) - Y_i(0) = \tau$. The constant treatment effect assumption implies that the potential outcomes are perfectly correlated ($\rho = 1.0$) and the variances of the two potential outcomes are equal ($\sigma_t^2 = \sigma_c^2 = \sigma^2$). Let us consider the following alternative conservative specification of the joint normal distribution of the potential outcomes, which are again free from dependence on the covariates:

$$\begin{pmatrix} Y_i(0) \\ Y_i(1) \end{pmatrix} \Bigg| \, \theta \sim \textsf{Normal}\left(\begin{pmatrix} \mu_c \\ \mu_t \end{pmatrix}, \begin{pmatrix} \sigma^2 & \sigma^2 \\ \sigma^2 & \sigma^2 \end{pmatrix} \right)$$

where μ_c and μ_t are α and $\alpha + \tau$ respectively.

The **Stan** program **Model_02_Assuming_Constant_Treatment_Effects.stan** for the model 2 is the same as the program for the model 1, with only a few differences. First, due to the assumption of equal variances (`sigma_t` = `sigma_c`), the sampling statment of Y^{obs} is now coded as `y ~ normal(alpha + tau*w, sigma);` with the pooled residual standard deviation `sigma`.

Second, assuming a correlation coefficient between $Y_i(1)$ and $Y_i(0)$ equal to 1, the conditional distribution of one potential outcome given the other now has zero variance ($\sigma_t^2(1-\rho^2) = \sigma_c^2(1-\rho^2) = 0$). Hence the **Stan** code for the model-based imputation becomes `y0[n] = mu_c + (y[n] - mu_t);` for $Y_i(0)$ and `y1[n] = mu_t + (y[n] - mu_c);` for $Y_i(1)$.

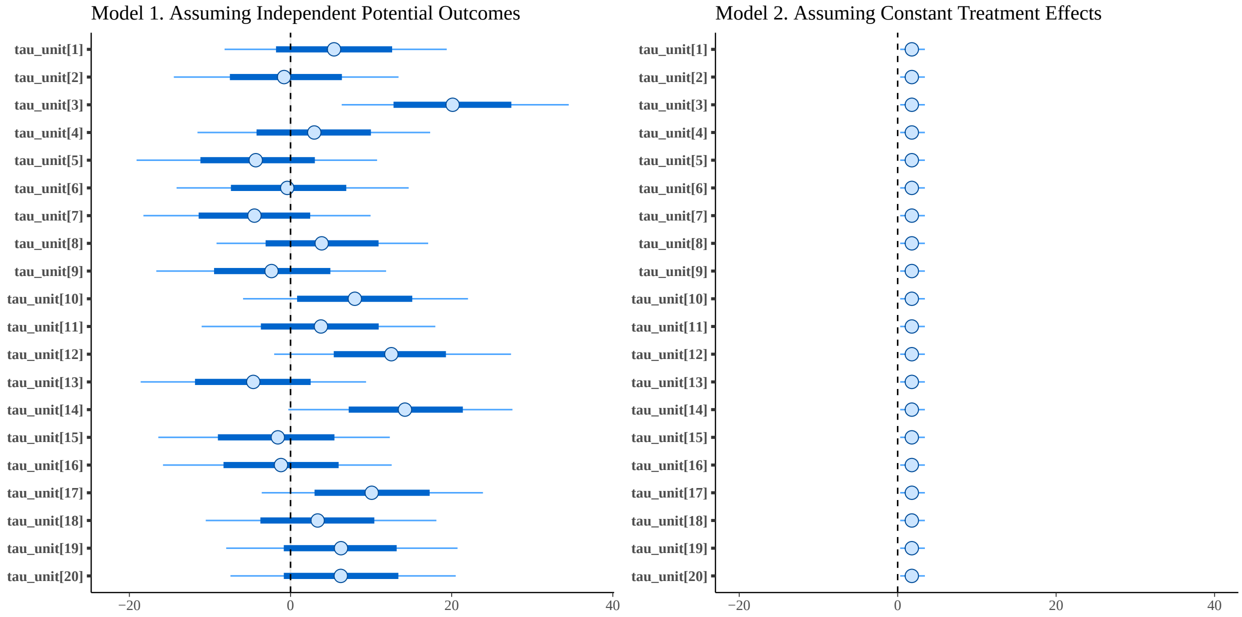
Note that we no longer draw samples using the pseudorandom number generator `normal_rng`. This means that the the uncertainty in the imputation vanishes and the only source of uncertainty is due to parameter estimation. Under the constant treatment effects assumption, we are able to know the exact value of Y_i^{mis} given Y_i^{obs} and the parameter estimates of μ_c and μ_t.

The **Stan** program for the Model 2 is fit to the example data:

```r
# Compile and run the stan model
fit_mod2 <- stan(file = "Model_02_Assuming_Constant_Treatment_Effects.stan",
                 data = stan_data,
                 iter = 1000, chains = 4)
```

To investigate the variance of the unit-level treatment effect in the Model 1 and 2, the first 20 example unit-level effects for the two models are presented side by side:

```r
# Model 1. assuming independence between Y(1) and Y(0)
color_scheme_set("brightblue")
tau_unit_idx <- paste0("tau_unit", "[", seq(20), "]")
mod1_posterior <- as.array(fit_mod1)
p1 <- mcmc_intervals(mod1_posterior, pars = tau_unit_idx,
                     prob = 0.8, prob_outer = 0.99, point_est = "mean" ) +
  vline_at(c(0), linetype = "dashed", size = 0.5) +
  ggplot2::labs(title = "Model 1. Assuming Independent Potential Outcomes")

# Model 2. assuming constant treatment effect
mod2_posterior <- as.array(fit_mod2)
p2 <- mcmc_intervals(mod2_posterior, pars = tau_unit_idx,
                     prob = 0.8, prob_outer = 0.99, point_est = "mean" ) +
  vline_at(c(0), linetype = "dashed", size = 0.5) +
  scale_x_continuous(limits = c(-20, 40)) +
  ggplot2::labs(title = "Model 2. Assuming Constant Treatment Effects")
grid.arrange(p1, p2, ncol = 2)
```



We observe that the unit-level treatment effects from the Model 2 are constant over all units and have minimal uncertainty intervals compared to the estimates from the Model 1. The Model 1 estimates show substantial variation among the posterior means with far larger uncertainty intervals. This plot clearly shows that the uncertainty in the imputation constitutes the main source of uncertainty when multiply-imputing the missing potential outcomes in model-based inference.

The conservative estimates of the finite-sample causal estimands are displayed in tabular form as follows:

```
param <- c("alpha", "tau", "sigma", "tau_fs", "tau_qte25", "tau_qte50", "tau_qte75" )
print(fit_mod2, pars = param,
      probs = c(0.1, 0.5, 0.9), digits = 3)
```

```
## Inference for Stan model: Model_02_Assuming_Constant_Treatment_Effects.
## 4 chains, each with iter=1000; warmup=500; thin=1;
## post-warmup draws per chain=500, total post-warmup draws=2000.
##
##              mean se_mean    sd   10%   50%   90% n_eff  Rhat
## alpha       4.554   0.012 0.404 4.043 4.555 5.065  1123 1.005
## tau         1.785   0.019 0.628 0.974 1.769 2.604  1127 1.004
## sigma       6.603   0.006 0.224 6.318 6.596 6.889  1223 1.002
## tau_fs      1.785   0.019 0.628 0.974 1.769 2.604  1127 1.004
## tau_qte25 1.785   0.019 0.628 0.974 1.769 2.604  1127 1.004
## tau_qte50 1.785   0.019 0.628 0.974 1.769 2.604  1127 1.004
## tau_qte75 1.785   0.019 0.628 0.974 1.769 2.604  1127 1.004
##
## Samples were drawn using NUTS(diag_e) at Fri Mar 19 10:01:54 2021.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

Since the unit-level treatment effects are assumed to have no heterogeneity in the Model 2, the QTE estimates are all identical to that for the ATE. Note also that the posterior means and standard deviations of the finite-sample ATE (`tau_fs`) is equal to those of the super-population ATE (`tau`). The worst-case scenario assumption of constant treatment effects not only gives a *conservative* estimate of the posterior variance for the finite-sample estimands but also provides an *unbiased* estimate of the super-population estimands (Imbens and Rubin 2015). See the discussion in Chapter 6 of Imbens and Rubin (2015) for more details.

Model 3: Model-based imputation with covariates

In the previous models without covariates, the posterior predictive distributions of the missing control potential outcomes had similar means centered around α for all the treated units. For control units, the missing treatment potential outcomes were imputed from the poterior predicteve distributions with the expected mean of $\alpha+\tau$ for all the control units. Adding the pre-treatment covariates improves the imputation process because the covariates provide information to help predict the missing potential outcomes. Now let us extend the previous model assuming independent potential outcomes (Model 1) to allow for pre-treatment covariates X:

$$\begin{pmatrix} Y_i(0) \\ Y_i(1) \end{pmatrix} \Bigg| X_i, \theta \sim \mathsf{Normal}\left(\begin{pmatrix} \alpha + \mathrm{X}_i\beta_c \\ \alpha + \mathrm{X}_i\beta_t + \tau \end{pmatrix}, \begin{pmatrix} \sigma_c^2 & 0 \\ 0 & \sigma_t^2 \end{pmatrix} \right)$$

where now $\theta = (\alpha, \beta_c, \beta_t, \tau, \sigma_c^2, \sigma_t^2)$.

Instead of imposing restrictions that the effects of X_i are the same for both potential outcomes, we define two different vectors of the slope coefficients β_c and β_t for the control and treated units repectively. The difference in the two vectors, $\beta_t - \beta_c$, can be obtained by including an interaction term between X and W in the model. We prepare a data list for **Stan** including the nine mean-centered covariates and their interaction terms with the treatment variable as follows:

```
# Add nine mean-centered covariates and their interaction terms
x <- as.matrix(lalonde[, c("age", "educ", "married", "nodegr", "black", "re74", "u74", "re75", "u75")])
x_mean_mat <- matrix(rep(apply(x, 2, mean, na.rm = TRUE), each = nrow(x)), nrow = nrow(x))
x_c_mat <- x - x_mean_mat
xw_inter <- x_c_mat*w
colnames(xw_inter) <-  paste0(colnames(x), "_w")

# Collect data into a list format suitable for Stan
stan_data <- list(y = y, w = w, x = x_c_mat, xw_inter = xw_inter, N = nrow(lalonde), N_cov = ncol(x_c_ma
```

In the **Stan** program **Model_03_Independent_Potential_Outcomes_With_Covariates.stan** for the Model 3, the data variables are coded following their specifications in the data list. In addition to the `alpha`, `tau`, `sigma_c`, and `sigma_t`, we specify two vectors for the slope coefficients in the `parameters` block: `beta` for β_c and `beta_inter` for $\beta_t - \beta_c$. The priors for the slope coefficients are specified to be normal with zero means and variance equal to 100^2. The sampling statement of Y^{obs} is then coded using matrix notation as

```
    y ~ normal(alpha + x*beta + xw_inter*beta_inter + tau * w, sigma_t*w + sigma_c*(1-w));
```

Assuming zero correlation between $Y_i(0)$ and $Y_i(1)$, the code for the model-based imputation is the same as the code for Model 1: `y0[n] = normal_rng(mu_c, sigma_c);` for $Y_i(0)$ and `y1[n] = normal_rng(mu_t, sigma_t);` for $Y_i(1)$. Because the covariates determines the location of the distribution but not its scale in Model 3, only the `mu_c` and `mu_t` are changed from the program for Model 1 as

```
    real mu_t = alpha + x[n,]*beta + x[n, ]*beta_inter + tau;
    real mu_c = alpha + x[n,]*beta;
```

The **Stan** program for the Model 3 is fit to the example data:

```
# Compile and run the stan model
# mod3_c <- stanc(file = "Model_03_Independent_Potential_Outcomes_With_Covariates.stan")
# mod3 <- stan_model(model_name =  mod3_c)
# fit_mod3 <- sampling(mod3, data = stan_data, iter = 1000, chains = 4)


fit_mod3 <- stan(file = "Model_03_Independent_Potential_Outcomes_With_Covariates.stan",
                 data = stan_data,
                 iter = 1000, chains = 1)
```

Next the posterior predictive distributions of the first 20 missng control potential outcomes are displayed. We see that the locations of the posterior predictive distributions vary in Model 3, whereas those of Model 1 are almost the same across the 20 missing potential outcomes.

```
# Model 1. Without Covariates
color_scheme_set("red")
ymis_idx <- paste0(ifelse(w[seq(20)] == 0, "y1", "y0"), "[", seq(20), "]")
mod1_posterior <- as.array(fit_mod1)
p1 <- mcmc_intervals(mod1_posterior, pars = ymis_idx,
                     prob = 0.8, prob_outer = 0.99, point_est = "mean" ) +
  #vline_at(c(0), linetype = "dashed", size = 0.5) +
  ggplot2::labs(title = "Model 1. Without Covariates")

# Model 2. assuming constant treatment effect
mod3_posterior <- as.array(fit_mod3)
p3 <- mcmc_intervals(mod3_posterior, pars = ymis_idx,
                     prob = 0.8, prob_outer = 0.99, point_est = "mean" ) +
  #vline_at(c(0), linetype = "dashed", size = 0.5) +
  #scale_x_continuous(limits = c(-20, 40)) +
  ggplot2::labs(title = "Model 3. With Covariates")
grid.arrange(p1, p3, ncol = 2)
```

The estimates of the causal estimands are displayed below. The posterior standard deviations for the ATE and QTE estimates are almost the same as those in Model 1, which means that the independence assumption between the potential outcomes determines the level of uncertainty for the causal estimates rather than the presence of covariates in the model. The posterior means for the ATE and QTEs replicates the estimates presented in the Table 8.6 of Imbens and Rubin (2015).

```
param <- c("tau", "tau_fs", "tau_qte25", "tau_qte50", "tau_qte75" )
print(fit_mod3, pars = param,
      probs = c(0.1, 0.5, 0.9), digits = 3)
```

References

Dehejia, Rajeev H, and Sadek Wahba. 1999. “Causal Effects in Nonexperimental Studies: Reevaluating the Evaluation of Training Programs.” *Journal of the American Statistical Association* 94 (448): 1053–62.

Ding, Peng, and Fan Li. 2017. "Causal Inference: A Missing Data Perspective." *arXiv Preprint arXiv:1712.06170*.

Gelman, Andrew. 2006. "Prior Distributions for Variance Parameters in Hierarchical Models." *Bayesian Analysis* 1 (3): 515–34.

Gelman, Andrew, Hal S Stern, John B Carlin, David B Dunson, Aki Vehtari, and Donald B Rubin. 2013. *Bayesian Data Analysis*. Chapman; Hall/CRC.

Heckman, James J, and V Joseph Hotz. 1989. "Choosing Among Alternative Nonexperimental Methods for Estimating the Impact of Social Programs: The Case of Manpower Training." *Journal of the American Statistical Association* 84 (408): 862–74.

Hyndman, Rob J, and Yanan Fan. 1996. "Sample Quantiles in Statistical Packages." *The American Statistician* 50 (4): 361–65.

Imbens, Guido W, and Donald B Rubin. 2015. *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge University Press.

Rubin, Donald B. 1978. "Bayesian Inference for Causal Effects: The Role of Randomization." *The Annals of Statistics*, 34–58.

Smith, Jeffrey A, and Petra E Todd. 2001. "Reconciling Conflicting Evidence on the Performance of Propensity-Score Matching Methods." *American Economic Review* 91 (2): 112–18.