Bayes-by Shower

Michael Betancourt

May 2025

Table of contents

Most epidemiological analyses are subject to a variety of subtle challenges. For example many health outcomes are influenced by exposures not only directly but also indirectly; if we cannot quantify both consistently then we will not be able to make accurate predictions for how health will vary under the various circumstances of interest. Moreover, because these exposures will generally vary across the individuals within any given population the corresponding health outcomes will vary from one population to another.

In this case study we'll see how Bayesian modeling and inference can be used to manage these difficulties and extract productive insights.

1 Setup

First and foremost we have to set up the local `R` environment.

```
par(family="serif", las=1, bty="l",
    cex.axis=1, cex.lab=1, cex.main=1,
    xaxs="i", yaxs="i", mar = c(5, 5, 3, 5))
```

```
library(rstan)
rstan_options(auto_write = TRUE)            # Cache compiled Stan programs
options(mc.cores = parallel::detectCores()) # Parallelize chains
parallel:::setDefaultClusterOptions(setup_strategy = "sequential")
```

Next we'll load some utility functions into the local environment to facilitate the implementation of Bayesian inference.

```
util <- new.env()
```

First we have a suite Markov chain Monte Carlo diagnostics and estimation tools; this code and supporting documentation are both available on GitHub.

```
source('mcmc_analysis_tools_rstan.R', local=util)
```

Second we have a suite of probabilistic visualization functions based on Markov chain Monte Carlo estimation. Again the code and supporting documentation are available on GitHub.

```
source('mcmc_visualization_tools.R', local=util)
```

2 Data Exploration

For this analysis we will be analyzing fertility across a cohort of male patients. Each patient was recruited through a referral for fertility preservation consultation and observed for the same twelve month period. Many of the referrals were also coincident with a cancer diagnosis.

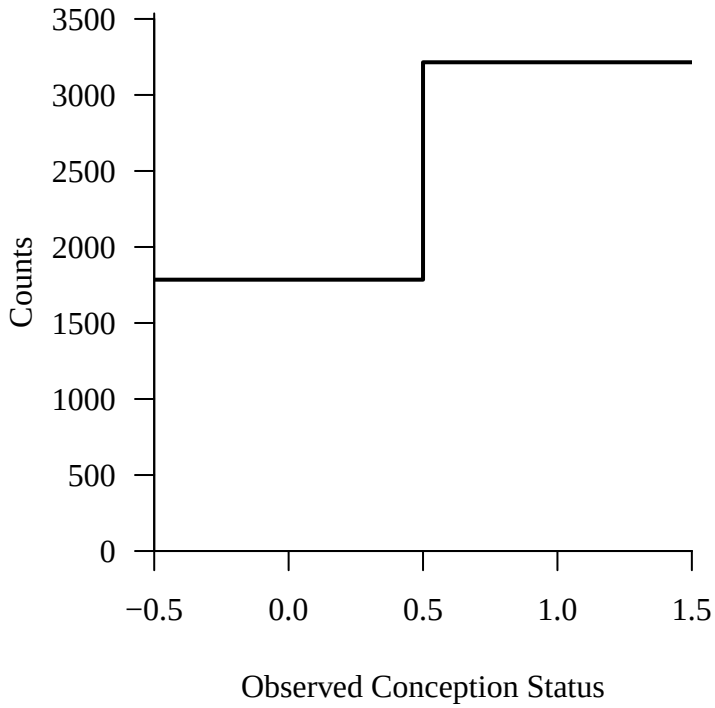
```
data <- read_rdump("data/conception.data.R")

names(data)
```

```
 [1] "k_trt" "y"     "K_stg" "k_art" "K_rel" "K_trt" "k_tox" "N"     "k_stg"
[10] "k_rel" "K_tox"
```

The main component of the data is a collection of binary observations indicating whether or not each patient in the observed cohort conceived a child during a particular year. Specifically $y_n = 0$ indicates that the nth patient did not conceive a child while $y_n = 1$ indicates that they conceived at least one child.

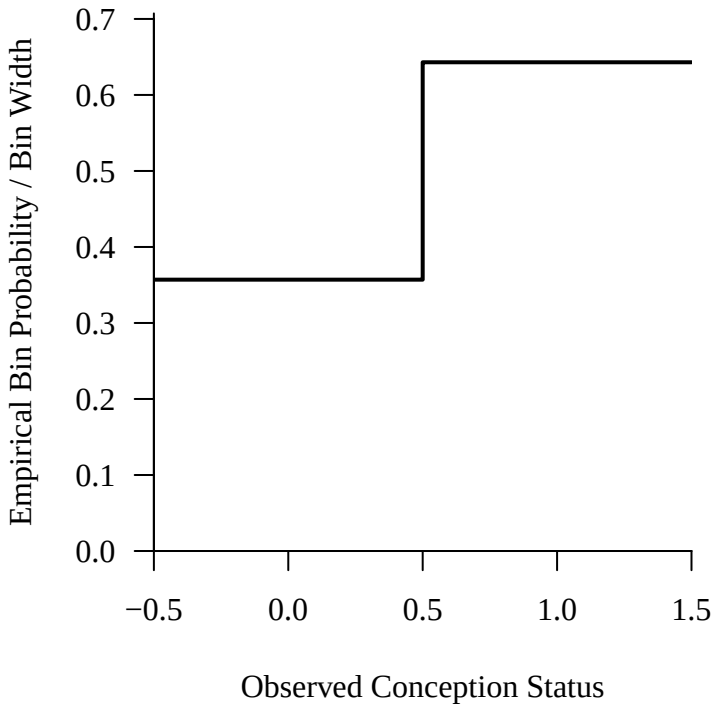
About two-thirds of the patients conceived a child.

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

util$plot_line_hist(data$y, -0.5, 1.5, 1,
                    xlab="Observed Conception Status")
```



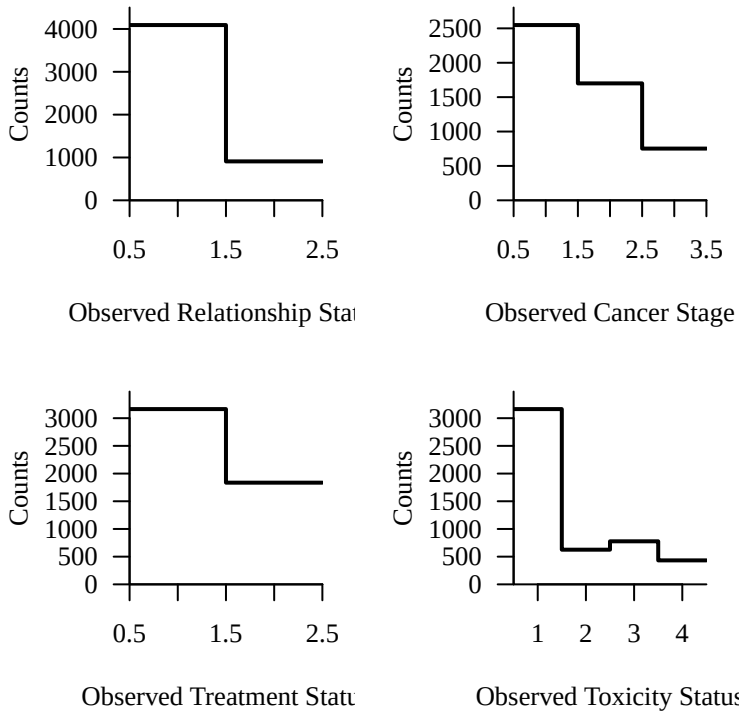
We can also clarify the relative frequencies of these fertility outcomes by normalizing the summed counts.

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

util$plot_line_hist(data$y, -0.5, 1.5, 1, prob=TRUE,
                    xlab="Observed Conception Status")
```



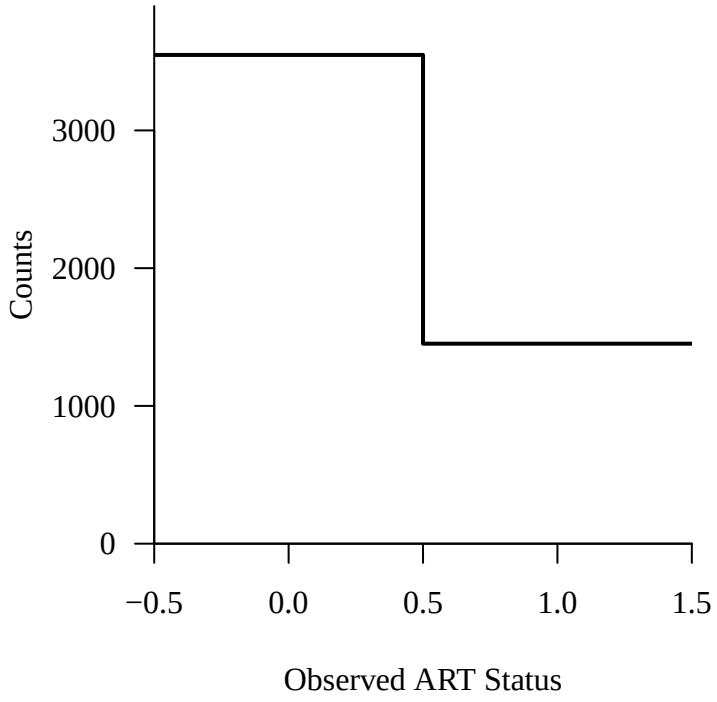
These fertility outcomes are complemented by clinical and demographic information about the individual patients in the cohort (**Table 1**). Note that these categorical variables are ordered, with a clear notion of increasing severity. Moreover all of these variables are indexed from `1`, even those with only two values; this is helpful when working with `1`-indexed programming languages like `Stan`.

```
par(mfrow=c(2, 2), mar=c(5, 5, 1, 1))

util$plot_line_hist(data$k_rel, 0.5, data$K_rel + 0.5, 1,
                    xlab="Observed Relationship Status")

util$plot_line_hist(data$k_stg, 0.5, data$K_stg + 0.5, 1,
                    xlab="Observed Cancer Stage")

util$plot_line_hist(data$k_trt, 0.5, data$K_trt + 0.5, 1,
                    xlab="Observed Treatment Status")

util$plot_line_hist(data$k_tox, 0.5, data$K_tox + 0.5, 1,
                    xlab="Observed Toxicity Status")
```



Finally some of the patients in the observed cohort have taken advantage of assisted reproductive technologies, or ART, which can drastically increase fertility. Similar to the fertility outcomes, and unlike the clinical and demographic information, ART participation is `0-1` coded, with `0` indicating no participation and `1` participation.

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

util$plot_line_hist(data$k_art, -0.5, 1.5, 1,
                    xlab="Observed ART Status")
```



Note that these data are not actually real but rather have been simulated from an epidemiologically-motivated true model for demonstration purposes. Consequently these observations are a bit more well-behaved than real data tends to be. Moreover there are no privacy concerns.

3 Model 1

Any epidemiological system is inherently complex. To avoid being overwhelmed by this complexity we'll start with a relatively simple observational model.

3.1 The Observational Model

Let's assume that the conception probability is homogeneous across all patients in the observed cohort,

$$p(y_1, \ldots, y_N \mid q_C) = \prod_{n=1}^{N} p(y_n \mid q_C) = \prod_{n=1}^{N} \text{Bernoulli}(y_n \mid q_C).$$

3.2 The Prior Model

To elevate this observational model into a full Bayesian model we need to complement it with a prior model for the conception probability.

Prior modeling in a demonstration is always tricky because few, if any, readers will share the same domain expertise. For this analysis let's just say that the available domain expertise is inconsistent with conception probabilities below 0.10 and above 0.95; these values are not outright impossible but much more extreme than the intermediate values. I will denote this soft constraint as

$$0.10 \lessapprox q_C \lessapprox 0.95.$$

Next we have to find a probability distribution over the unit interval that is consistent with these thresholds. Conveniently the beta family of probability density functions specifies a diversity of candidate probability distributions over the unit interval.

To select a prior model from these candidates we need to define consistency, although we don't have to be too precious. I like to define consistency with tail probability conditions,

$$\delta = \pi([0.00, 0.10]) = \int_{0.00}^{0.10} \mathrm{d}q_C \, \mathrm{beta}(q_C \mid \alpha, \beta)$$
$$\delta = \pi([0.95, 1.00]) = \int_{0.95}^{1.00} \mathrm{d}q_C \, \mathrm{beta}(q_C \mid \alpha, \beta);$$

The precise value of δ isn't too important provided that it is close to zero. I typically take $\delta = 0.01$.

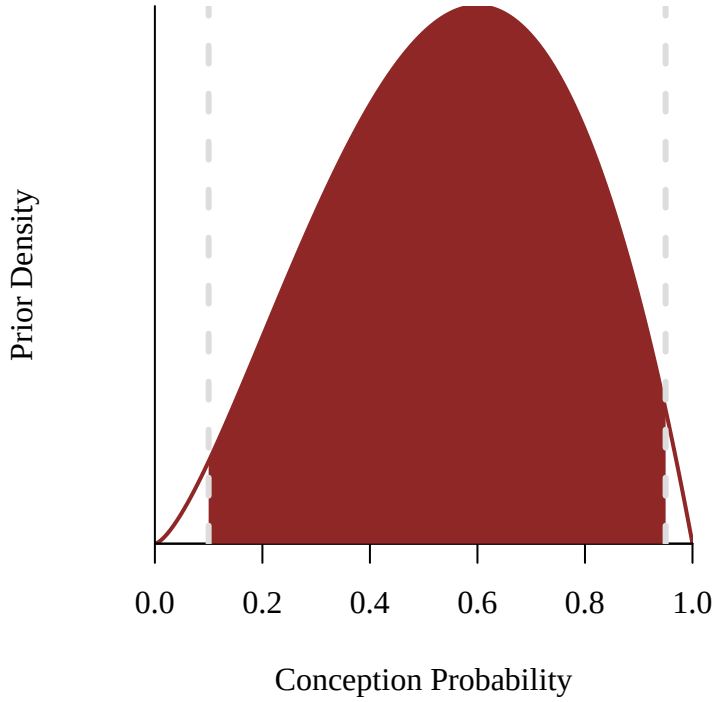
One nice benefit of this definition of consistency is that we can numerically solve for the parameters α and β that identify the compatible beta probability density function.

```
q_low <- 0.1
q_high <- 0.95

stan(file='stan_programs/prior_tune_beta.stan',
     data=list('q_low' = q_low, 'q_high' = q_high),
     iter=1, warmup=0, chains=1,
     seed=4838282, algorithm="Fixed_param")
```

```
alpha = 2.52283
beta = 2.02117

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1: Iteration: 1 / 1 [100%]  (Sampling)
Chain 1:
```

```
Chain 1:  Elapsed Time: 0 seconds (Warm-up)
Chain 1:                0 seconds (Sampling)
Chain 1:                0 seconds (Total)
Chain 1:


Inference for Stan model: anon_model.
1 chains, each with iter=1; warmup=0; thin=1;
post-warmup draws per chain=1, total post-warmup draws=1.

      mean se_mean sd 2.5%  25%  50%  75% 97.5% n_eff Rhat
alpha 2.52      NA NA 2.52 2.52 2.52 2.52  2.52     0  NaN
beta  2.02      NA NA 2.02 2.02 2.02 2.02  2.02     0  NaN
lp__  0.00      NA NA 0.00 0.00 0.00 0.00  0.00     0  NaN

Samples were drawn using (diag_e) at Thu May 29 15:10:15 2025.
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
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

qs <- seq(0, 1, 0.001)
dens <- dbeta(qs, 2.5, 2.0)
plot(qs, dens, type="l", col=util$c_dark, lwd=2,
     xlab="Conception Probability",
     ylab="Prior Density", yaxt='n')

q98 <- seq(q_low, q_high, 0.001)
dens <- dbeta(q98, 2.5, 2.0)
q98 <- c(q98, q_high, q_low)
dens <- c(dens, 0, 0)

polygon(q98, dens, col=util$c_dark, border=NA)

abline(v=q_low,  lwd=3, lty=2, col='#DDDDDD')
abline(v=q_high, lwd=3, lty=2, col='#DDDDDD')
```



3.3 Posterior Quantification

Putting everything together we can summarize the structure of the full Bayesian model with a directed graph (Figure 1) and implement the full Bayesian model in a `Stan` program.

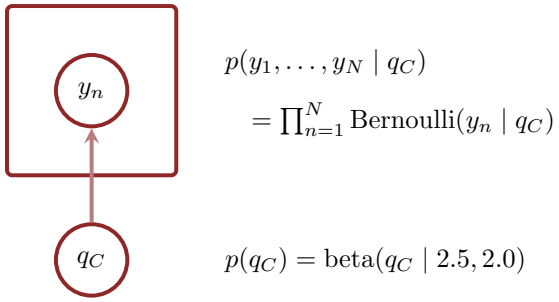


Figure 1: A directed graphical model visually summarizes the narratively generative structure of our initial model.

With this Stan program we can then run Markov chain Monte Carlo to extract information about the posterior distribution that can be used to estimate posterior expectation values.

```
fit <- stan(file="stan_programs/model1.stan",
            data=data, seed=8438338,
            warmup=1000, iter=2024, refresh=0)
```

Before doing anything else we have to check for any signs that the posterior computation might be inaccurate. Fortunately there are no diagnostic warnings suggesting any problems.

```
diagnostics <- util$extract_hmc_diagnostics(fit)
util$check_all_hmc_diagnostics(diagnostics)
```

```
  All Hamiltonian Monte Carlo diagnostics are consistent with reliable
Markov chain Monte Carlo.
```

```
samples1 <- util$extract_expectand_vals(fit)
base_samples <- util$filter_expectands(samples1,
                                       c('q_C'))
util$check_all_expectand_diagnostics(base_samples)
```

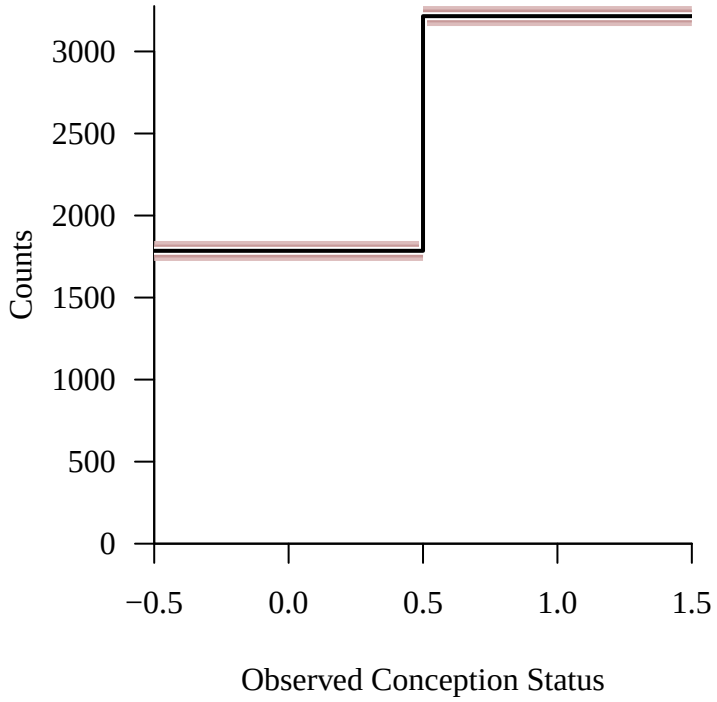
```
All expectands checked appear to be behaving well enough for reliable
Markov chain Monte Carlo estimation.
```

3.4 Retrodictive Checks

Next we need to evaluate the adequacy of our model by comparing the behavior of the observed data to the posterior predictive distribution with a posterior retrodictive check.

Here we'll consider a posterior retrodictive check using the histogram summary statistic that we that we used when exploring the data. Fortunately there is no sign of tension between the observed histogram and the probability distribution of posterior predictive histograms.

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

util$plot_hist_quantiles(samples1, 'y_pred', -0.5, 1.5, 1,
                         baseline_values=data$y,
                         xlab="Observed Conception Status")
```

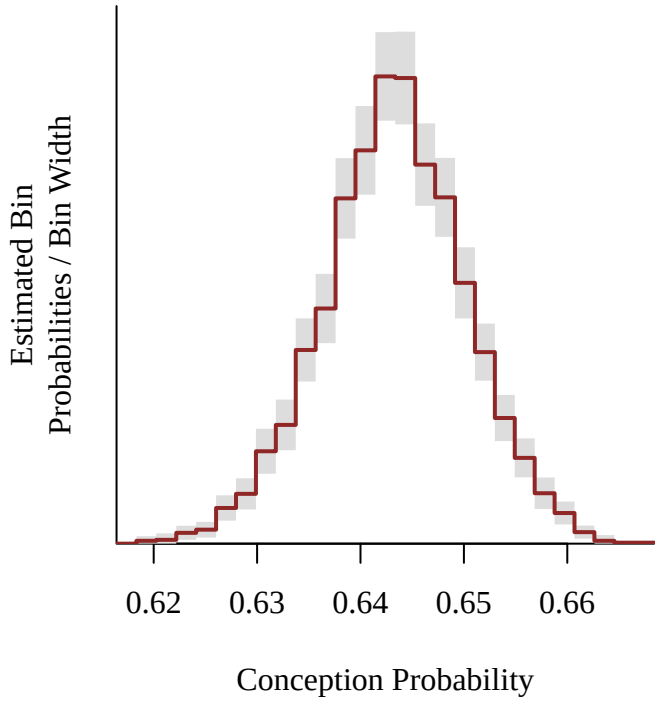


3.5 Posterior Insights

Flush with confidence in our modeling assumptions, at least in the context of the lone summary statistic we considered, we can examine the resulting posterior inferences.

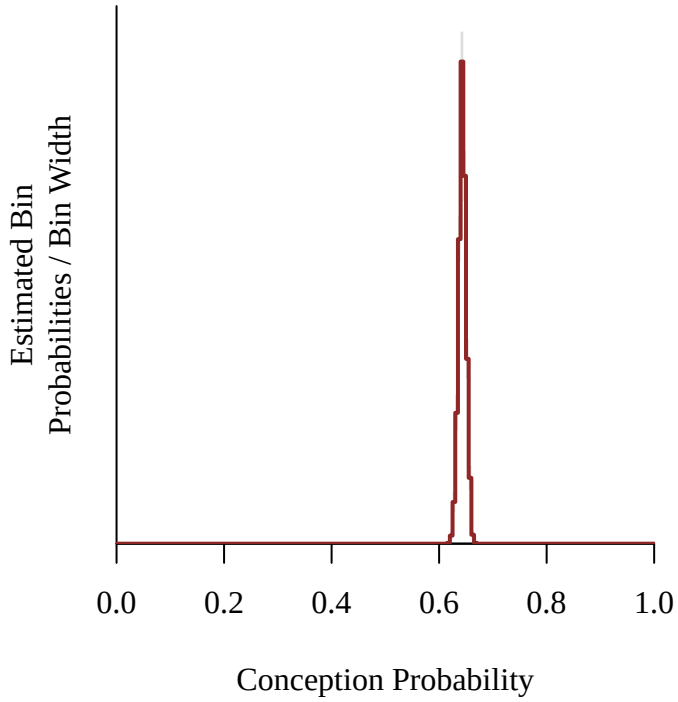
The posterior distribution concentrates on conception probabilities near 2/3, consistent with the empirical behavior. One immediate benefit of the Bayesian approach is that the posterior distribution captures all of the conception probabilities consistent with the observed data, quantifying *uncertainty* in our insights.

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))
util$plot_expectand_pushforward(samples1[['q_C']],
                                25,
                                display_name="Conception Probability")
```



When examining inferences for probability parameters it's often helpful to plot the full range of possible values and offer as much context as possible.

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))
util$plot_expectand_pushforward(samples1[['q_C']],
                                200, flim=c(0, 1),
                                display_name="Conception Probability")
```



4 Model 2

In practice the adequacy of a model is determined by the criteria we use to critique it. Our initial model adequately captures the *aggregate* conception behavior across the observed cohort, but that doesn't mean it will be able to capture finer details.

For example there's no reason why patient fertility should not vary across most of the available clinical and demographic categories. A male patient in a stable relationship with a female partner is more likely to conceive than one who is not. Similarly more aggressive cancer, and the ancillary toxicity common to most cancer treatments, is likely to reduce fertility and hence conception probability.

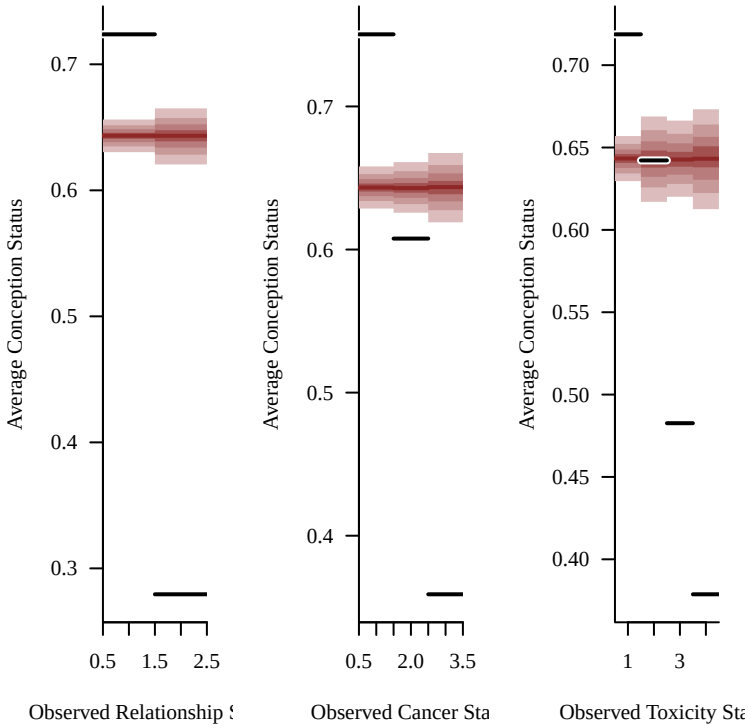
Fertility should also vary with treatment, but only as a side effect of treatment toxicity. Because we're modeling treatment toxicity directly we don't need to consider heterogeneity across the treatment groups.

The key question for a practical analysis is *not* whether or not the variations in conception probability are zero but rather whether or not they are large enough to manifest in the observed data. Because our initial model assumes homogeneous conception probabilities, the posterior predictive conception behavior should be the same no matter how we partition the patients. If the heterogeneity in fertility is strong enough then the observed behaviors will fall outside

of the posterior predictive uncertainties, indicating the inadequacy of our initial homogeneous model.

There are many ways to stratify retrodictive comparisons across categories. Here I'm going to use the conditional mean summary statistic introduced in Section 2.5 of my Taylor modeling chapter and implemented in my recommended visualization tools.

```
par(mfrow=c(1, 3), mar=c(5, 5, 1, 1))

pred_names <- sapply(1:data$N, function(n) paste0('y_pred[', n, ']'))

util$plot_conditional_mean_quantiles(samples1, pred_names, data$k_rel,
                                     0.5, data$K_rel + 0.5, 1, data$y,
                                     xlab="Observed Relationship Status",
                                     ylab="Average Conception Status")

util$plot_conditional_mean_quantiles(samples1, pred_names, data$k_stg,
                                     0.5, data$K_stg + 0.5, 1, data$y,
                                     xlab="Observed Cancer Stage",
                                     ylab="Average Conception Status")

util$plot_conditional_mean_quantiles(samples1, pred_names, data$k_tox,
                                     0.5, data$K_tox + 0.5, 1, data$y,
                                     xlab="Observed Toxicity Status",
                                     ylab="Average Conception Status")
```



Indeed we see clear disagreement between the observed and posterior predictive behavior. This indicates that we need to incorporate systematic variation in fertility in order to adequately model this cohort of patients.

4.1 Observational Model

One of the most productive ways to model variation across a population is to define an interpretable *baseline* and then model *deviations* around that baseline. Here we'll take our baseline to be the subset of patients in a stable relationship, with no cancer diagnosis, and no treatment toxicity, and use the parameter q_{C_0} to model the baseline conception probability.

An immediate benefit of this choice of baseline is that fertility should always *decrease* as we move from the baseline to more extreme patient characteristics. Increasing treatment toxicity, for instance, should never increase fertility. Consequently we can model the conception probability in other patient characteristics as *proportional decreases* from the baseline conception probability.

More formally for any clinical or demographic grouping we will model the conception probability in the baseline group as

$$q = q_{C_0}\,\delta_1$$

for $\delta_1 = 1$, or equivalently

$$q = q_{C_0}\,\exp(-\alpha_1)$$

for $\alpha_1 = 0$. Then we can model the conception probability in the least extreme group beyond the baseline as

$$q = q_{C_0}\, \delta_2$$

for

$$0 < \delta_2 < \delta_1 = 1,$$

or equivalently

$$q = q_{C_0}\, \exp(-\alpha_2)$$

for

$$0 = \alpha_1 < \alpha_2.$$

Similarly the conception probability in the second least extreme group becomes

$$q = q_{C_0}\, \delta_3$$

for

$$0 < \delta_3 < \delta_2 < \delta_1 = 1,$$

or equivalently

$$q = q_{C_0}\, \exp(-\alpha_3)$$

for

$$0 = \alpha_1 < \alpha_2 < \alpha_3.$$

I will refer to the α_k for each clinical or demographic grouping as *impairment parameters*.

In order to model the variation across K different groups we need a collection of K positive and ordered parameters that start at zero.

$$0 = \alpha_1 < \alpha_2 < ... < \alpha_k < ... < \alpha_K.$$

This multivariate constraint can be tricky to maintain in practice.

Finally we repeat this construction three times to capture the variation in fertility across each of the three patient characteristics under consideration (Figure 2),

$$\begin{aligned} q_{C,n} &= q_{C_0}\, \exp(-\alpha_{\text{stg},n})\, \exp(-\alpha_{\text{rel},n})\, \exp(-\alpha_{\text{tox},n}) \\ &= q_{C_0}\, \exp(-\alpha_{\text{stg},n} - \alpha_{\text{rel},n} - \alpha_{\text{tox},n}). \end{aligned}$$

where

$$\begin{aligned} \alpha_{\text{stg},n} &= \alpha_{\text{rel}}[k_{\text{stg},n}] \\ \alpha_{\text{rel},n} &= \alpha_{\text{rel}}[k_{\text{rel},n}] \\ \alpha_{\text{tox},n} &= \alpha_{\text{rel}}[k_{\text{tox},n}] \end{aligned}$$

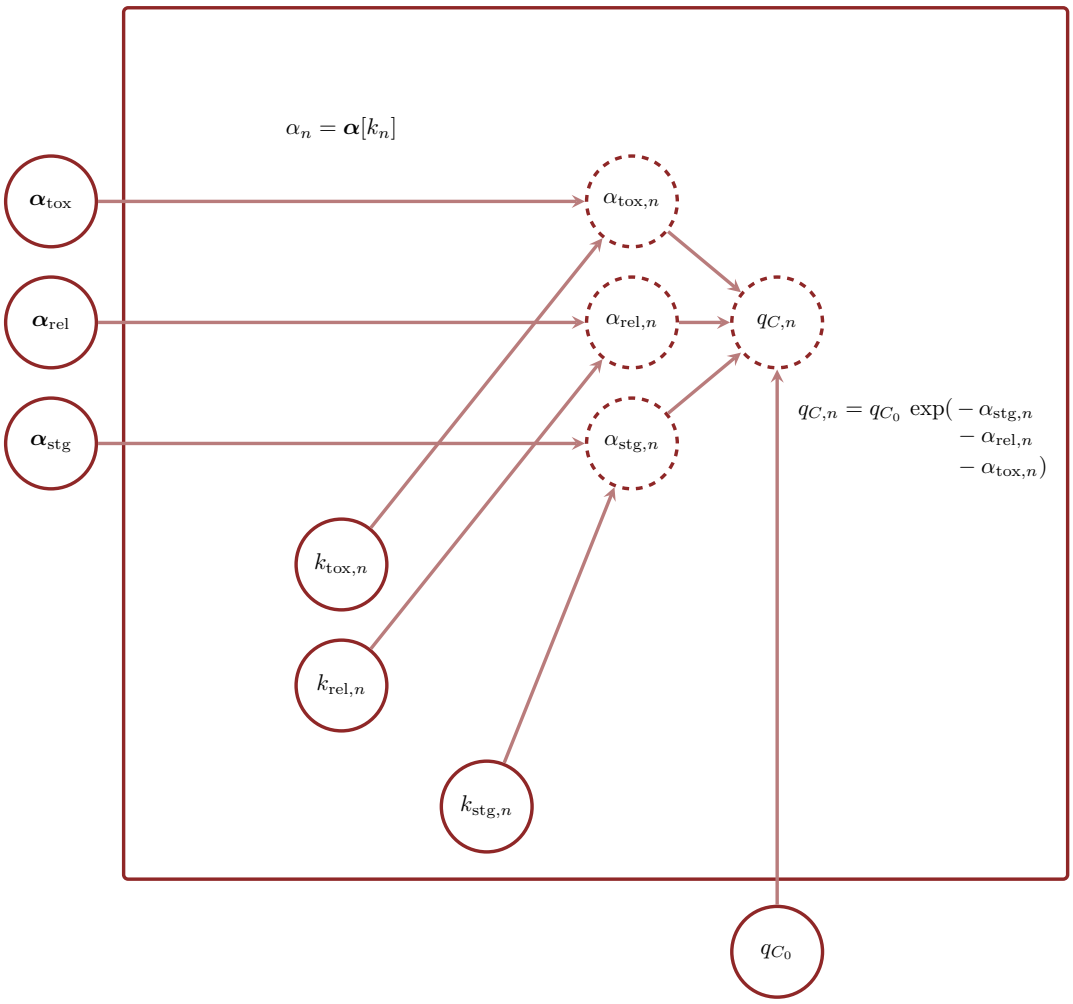


Figure 2: We model fertility heterogeneity by allowing conception probability to vary across cancer stage, relationship status, and treatment toxicity groups. More precisely the conception probability for each patient $q_{C,n}$ is modeled as a baseline conception probability q_{C_0} coupled with proportional decreases depending on group membership.

4.2 Prior Model

In this new model we are no longer modeling a population-wide conception probability but rather the conception probability for only the baseline group of patients who are in stable relationships and have not been diagnosed with cancer. If we have additional domain expertise about this smaller group then we can incorporate it into a more informative prior model.

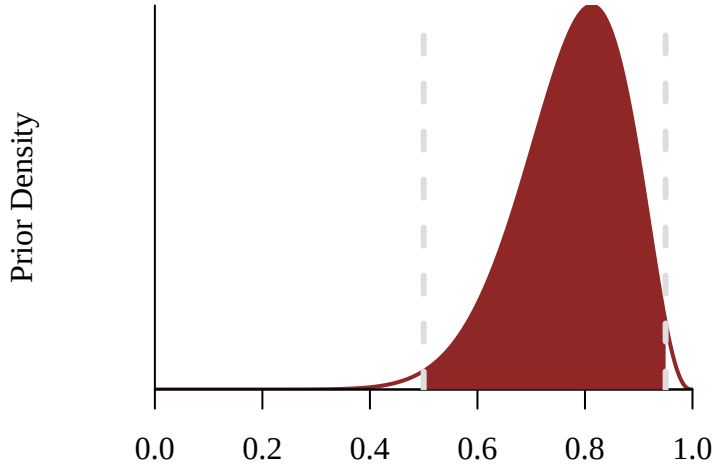
Here let's say that our domain expertise is inconsistent with baseline conception probabilities below 0.5 and above 0.95,

$$0.50 \lessapprox q_{C_0} \lessapprox 0.95.$$

```
q_low <- 0.5
q_high <- 0.95

stan(file='stan_programs/prior_tune_beta.stan',
     data=list('q_low' = q_low, 'q_high' = q_high),
     iter=1, warmup=0, chains=1,
     seed=4838282, algorithm="Fixed_param")
```

```
alpha = 12.6454
beta = 3.74419

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1: Iteration: 1 / 1 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0 seconds (Warm-up)
Chain 1:                0 seconds (Sampling)
Chain 1:                0 seconds (Total)
Chain 1:

Inference for Stan model: anon_model.
1 chains, each with iter=1; warmup=0; thin=1;
post-warmup draws per chain=1, total post-warmup draws=1.

       mean se_mean sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
alpha 12.65      NA NA 12.65 12.65 12.65 12.65 12.65     0  NaN
beta   3.74      NA NA  3.74  3.74  3.74  3.74  3.74     0  NaN
lp__   0.00      NA NA  0.00  0.00  0.00  0.00  0.00     0  NaN

Samples were drawn using (diag_e) at Thu May 29 15:10:23 2025.
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).
```

```
par(mfrow=c(1, 1), mar=c(5, 5, 5, 1))

qs <- seq(0, 1, 0.001)
dens <- dbeta(qs, 12.7, 3.7)
plot(qs, dens, type="l", col=util$c_dark, lwd=2,
     xlab="Baseline Conception Probability",
     ylab="Prior Density", yaxt='n')

q98 <- seq(q_low, q_high, 0.001)
dens <- dbeta(q98, 12.7, 3.7)
q98 <- c(q98, q_high, q_low)
dens <- c(dens, 0, 0)

polygon(q98, dens, col=util$c_dark, border=NA)

abline(v=q_low,  lwd=3, lty=2, col='#DDDDDD')
abline(v=q_high, lwd=3, lty=2, col='#DDDDDD')
```



To construct a prior model for these new fertility impairment parameters we need to elicit any available domain expertise about the reasonable proportional decreases across groups. For example let's say that our domain expertise is inconsistent with any decreases below 5%,

$$0.05 \lessapprox \delta \lessapprox 1.$$

This requires

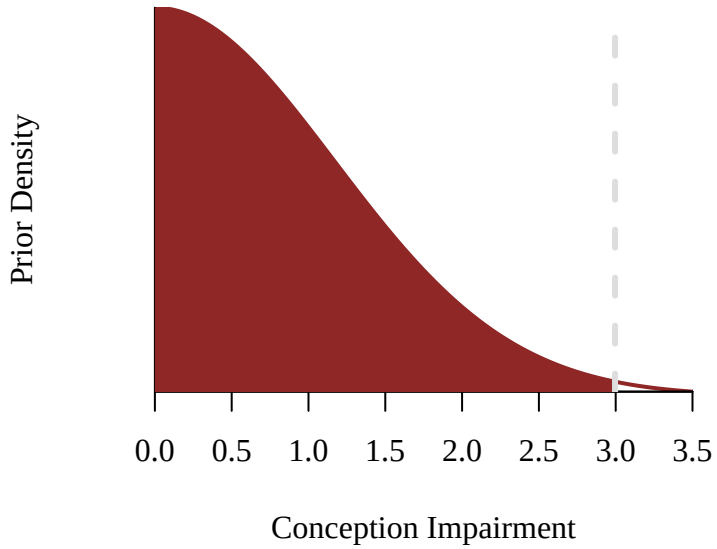
$$\begin{aligned} 0.05 &\lessapprox & \delta &\lessapprox 1 \\ 0.05 &\lessapprox \exp(\alpha) &&\lessapprox 1 \\ -\log(1) &\lessapprox & \alpha &\lessapprox -\log(0.05) \\ 0 &\lessapprox & \alpha &\lessapprox -\log(0.05). \end{aligned}$$

We can achieve this prior containment with the half-normal prior model

$$p(\alpha) = \text{half-normal}(\alpha \mid 0, -\log(0.05)/2.57)$$

that contains 99% of the prior probability between 0 and $-\log(0.05)$.

```
q_high <- -log(0.05)

par(mfrow=c(1, 1), mar=c(5, 5, 5, 1))

qs <- seq(0, 3.5, 0.001)
dens <- 2 * dnorm(qs, 0, q_high / 2.57 )
plot(qs, dens, type="l", col=util$c_dark, lwd=2,
     xlab="Conception Impairment",
     ylab="Prior Density", yaxt='n')

q98 <- seq(0, q_high, 0.001)
dens <- 2 * dnorm(q98, 0, q_high / 2.57 )
q98 <- c(0, q98, -log(0.05))
dens <- c(0, dens, 0)

polygon(q98, dens, col=util$c_dark, border=NA)

abline(v=q_high,  lwd=3, lty=2, col='#DDDDDD')
```



4.3 Posterior Quantification

The updated observational and prior models snap together into a more elaborate full Bayesian model (Figure 3).

Note that in the Stan programming language a normal log probability density function is equivalent to a half-normal log probability density function so long as the input variable is constrained to be positive. Consequently we can implement half-normal prior models for the impairment parameters using a normal probability density function.

In order to maintain the assumed constraints on the impairment parameters for each non-baseline group,

$$0 < \alpha_2 < ... < \alpha_k < ... < \alpha_K,$$

we use the Stan programming language's `positive_ordered` variable type. We can then ensure the assumed constraint for all groups including the baselines,

$$0 = \alpha_1 < \alpha_2 < ... < \alpha_k < ... < \alpha_K,$$

by prepending the `positive_ordered` variable with a zero.

```
fit <- stan(file="stan_programs/model2.stan",
            data=data, seed=8438339,
            warmup=1000, iter=2024, refresh=0)
```

The computational diagnostics don't suggest any problems with our posterior quantification.

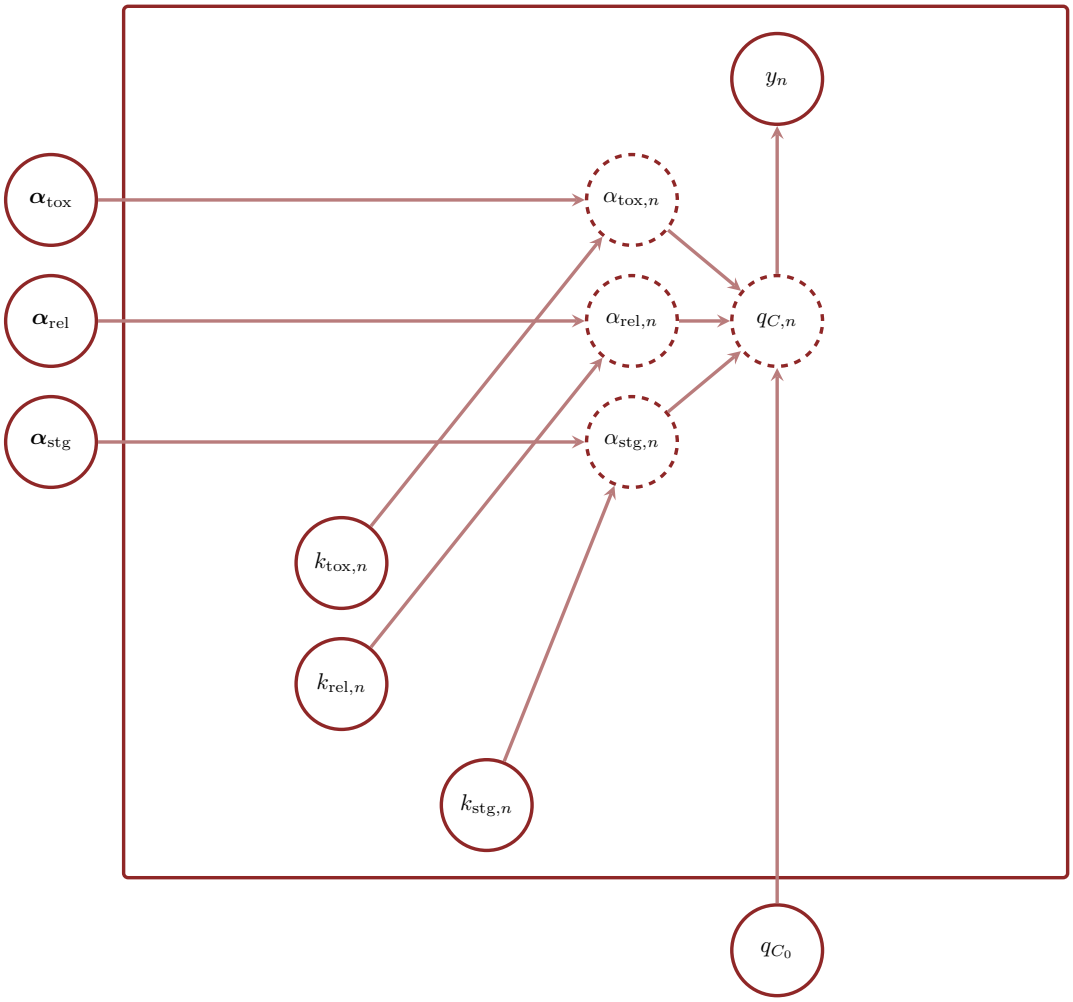


Figure 3: Our second model replaces the homogeneous conception probability parameter from the first model with the varying outputs of a deterministic function of the clinical and demographic group memberships of each patient.

```
diagnostics <- util$extract_hmc_diagnostics(fit)
util$check_all_hmc_diagnostics(diagnostics)
```

```
  All Hamiltonian Monte Carlo diagnostics are consistent with reliable
Markov chain Monte Carlo.
```

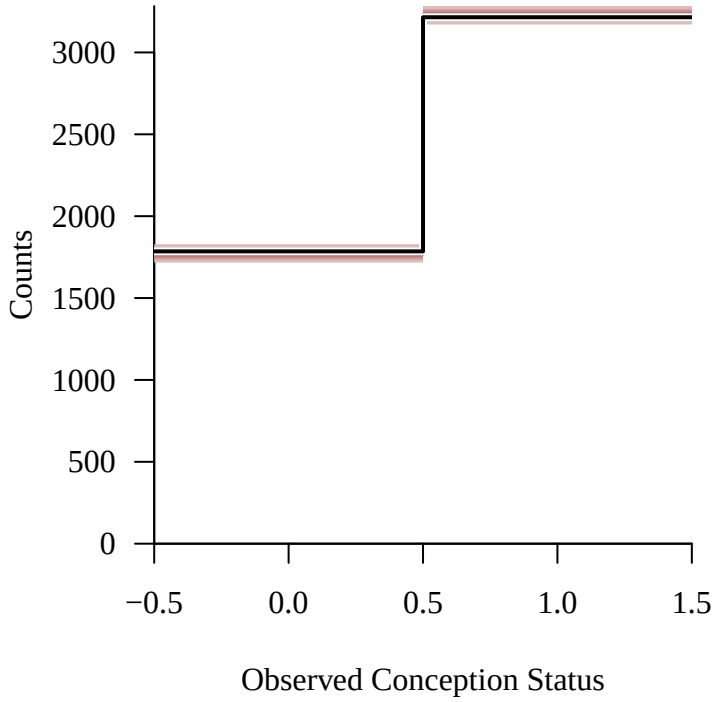
```
samples2 <- util$extract_expectand_vals(fit)
base_samples <- util$filter_expectands(samples2,
                                       c('q_C_0', 'alpha_rel' ,
                                         'alpha_stg', 'alpha_tox'),
                                       check_arrays=TRUE)
util$check_all_expectand_diagnostics(base_samples)
```

```
All expectands checked appear to be behaving well enough for reliable
Markov chain Monte Carlo estimation.
```

4.4 Retrodictive Checks

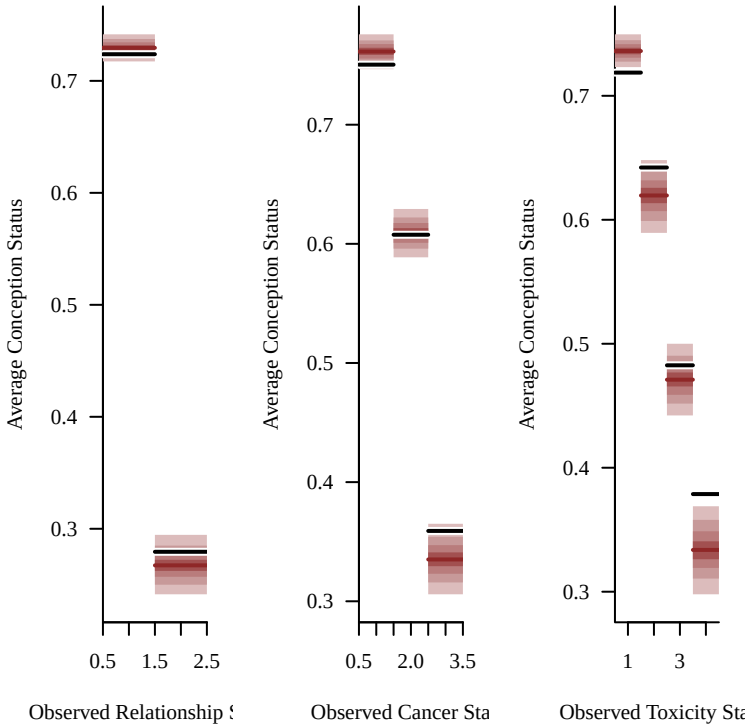
The retrodictive performance aggregated across the entire population continues to be strong.

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

util$plot_hist_quantiles(samples2, 'y_pred', -0.5, 1.5, 1,
                         baseline_values=data$y,
                         xlab="Observed Conception Status")
```



Now, however, the conditional retrodictive checks across the patient characteristics also look good. This suggests that our model of the variation is adequate, at least for this particular data set.

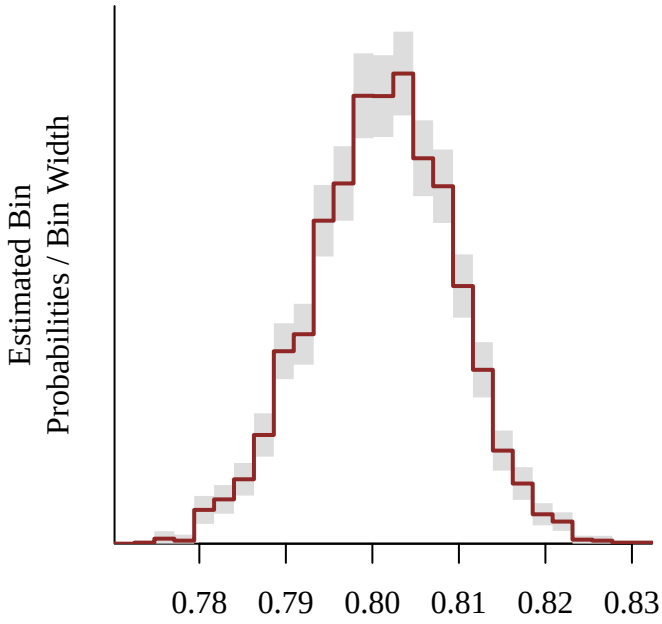
```
par(mfrow=c(1, 3), mar=c(5, 5, 1, 1))

pred_names <- sapply(1:data$N, function(n) paste0('y_pred[', n, ']'))

util$plot_conditional_mean_quantiles(samples2, pred_names, data$k_rel,
                                     0.5, data$K_rel + 0.5, 1, data$y,
                                     xlab="Observed Relationship Status",
                                     ylab="Average Conception Status")

util$plot_conditional_mean_quantiles(samples2, pred_names, data$k_stg,
                                     0.5, data$K_stg + 0.5, 1, data$y,
                                     xlab="Observed Cancer Stage",
                                     ylab="Average Conception Status")

util$plot_conditional_mean_quantiles(samples2, pred_names, data$k_tox,
                                     0.5, data$K_tox + 0.5, 1, data$y,
                                     xlab="Observed Toxicity Status",
                                     ylab="Average Conception Status")
```



4.5 Posterior Insights

The expanded model offers a variety of posterior behaviors to consider. Note that in order to learn the baseline conception probability, and hence variations away from that baseline, we needed to have patients who are not diagnosed with cancer in the observed cohort. If such a cohort is not possible then we would need to inform q_{C_0} using a strong prior model informed by domain expertise, previous studies, or a combination of the two.

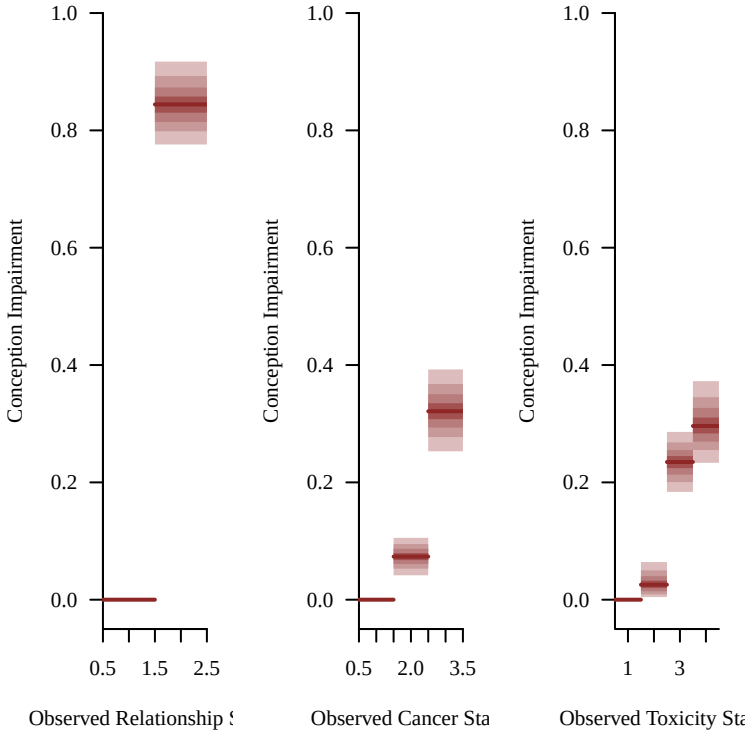
```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))
util$plot_expectand_pushforward(samples2[['q_C_0']],
                                25,
                                display_name=paste("Baseline",
                                                   "Probability",
                                                   "of Conception"))
```



Because of their non-linear influence on the patient conception probabilities the impairment parameters can be tricky to correctly interpret.

```
par(mfrow=c(1, 3), mar=c(5, 5, 1, 1))

names <- sapply(1:data$K_rel,
                function(k) paste0('alpha_rel_buff[', k, ']'))
util$plot_disc_pushforward_quantiles(samples2, names,
                                     xlab="Observed Relationship Status",
                                     ylab="Conception Impairment",
                                     display_ylim=c(-0.05, 1))

names <- sapply(1:data$K_stg,
                function(k) paste0('alpha_stg_buff[', k, ']'))
util$plot_disc_pushforward_quantiles(samples2, names,
                                     xlab="Observed Cancer Stage",
                                     ylab="Conception Impairment",
                                     display_ylim=c(-0.05, 1))

names <- sapply(1:data$K_tox,
                function(k) paste0('alpha_tox_buff[', k, ']'))
util$plot_disc_pushforward_quantiles(samples2, names,
                                     xlab="Observed Toxicity Status",
```

```
                                    ylab="Conception Impairment",
                                    display_ylim=c(-0.05, 1))
```

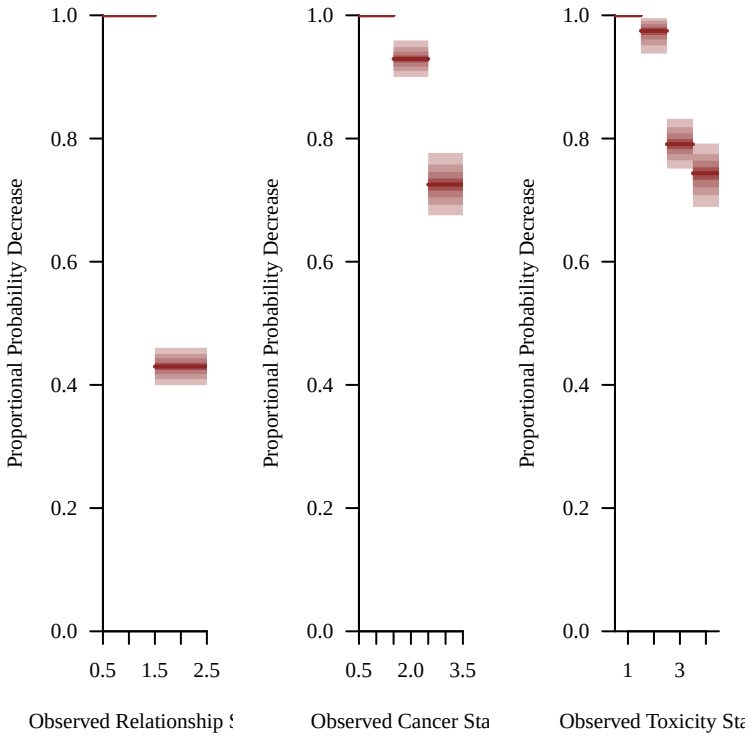


Fortunately we can always propagate our posterior inferences to the more interpretable proportional decreases,

$$\gamma_k = \exp(-\alpha_k).$$

```
par(mfrow=c(1, 3), mar=c(5, 5, 1, 1))

names <- sapply(1:data$K_rel,
                function(k) paste0('gamma_rel_buff[', k, ']'))
util$plot_disc_pushforward_quantiles(samples2, names,
                                     xlab="Observed Relationship Status",
                                     ylab=paste("Proportional",
                                                "Probability Decrease"),
                                     display_ylim=c(0, 1))

names <- sapply(1:data$K_stg,
                function(k) paste0('gamma_stg_buff[', k, ']'))
util$plot_disc_pushforward_quantiles(samples2, names,
                                     xlab="Observed Cancer Stage",
```

```
                                    ylab=paste("Proportional",
                                               "Probability Decrease"),
                                    display_ylim=c(0, 1))

names <- sapply(1:data$K_tox,
                function(k) paste0('gamma_tox_buff[', k, ']'))
util$plot_disc_pushforward_quantiles(samples2, names,
                                    xlab="Observed Toxicity Status",
                                    ylab=paste("Proportional",
                                               "Probability Decrease"),
                                    display_ylim=c(0, 1))
```



5 Model 3

Because all of the patient characteristics in the observed cohort are known we did not have to model them to infer heterogeneity in fertility within the cohort. If patient characteristics are prone to missingness or we want to inform behaviors for other patient cohorts, however, then we need to model the patient characteristics. The latter is particularly important if we want

to consider hypothetical, sometimes referred to as counterfactual or out-of-sample, cohorts subject to potential interventions.

5.1 Model 3a

Because consistently modeling all of the patient characteristics at the same time can be challenging we'll walk through the process as deliberately as possible.

5.1.1 Patient Characteristic Observational Model

In general the patient characteristics of any population are defined by a joint probability distribution. For a population of patients characterized by relationship status, cancer stage, treatment status, and treatment toxicity status this means modeling the joint probability density function

$$p(\text{relationship}, \text{stage}, \text{treatment}, \text{toxicity}).$$

Modeling joint probability distributions, and all of the complex couplings between the component variables they can manifest, can be overwhelming. One way to make them more manageable is to decompose them into lower-dimensional conditional probability distributions. When this decomposition follows the data generating process these conditional probability distributions become more interpretable and hence more straightforward to model.

For example consider the conditional decomposition (Figure 4)

$$\begin{aligned} p(\text{relationship}, \text{stage}, \text{treatment}, \text{toxicity}) \\ = \; & p(\text{toxicity} \mid \text{relationship}, \text{stage}, \text{treatment}) \\ & \cdot p(\text{treatment} \mid \text{relationship}, \text{stage}) \\ & \cdot p(\text{relationship} \mid \text{stage}) \\ & \cdot p(\text{stage}). \end{aligned}$$

Treatment toxicity certainly depends on whether or not a patient is undergoing treatment in the first place. Moreover it can depend on the stage of cancer, with more aggressive cancers making a patient more vulnerable to treatment side effects. On the other hand treatment toxicity is not directly influenced by the relationship status of a patient. Consequently the first conditional probability simplifies to

$$p(\text{toxicity} \mid \text{relationship}, \text{stage}, \text{treatment}) = p(\text{toxicity} \mid \text{stage}, \text{treatment}).$$

In theory treatment status could depend on relationship status; for example a long term partner might increase the probability of seeking treatment. For this analysis, however, we

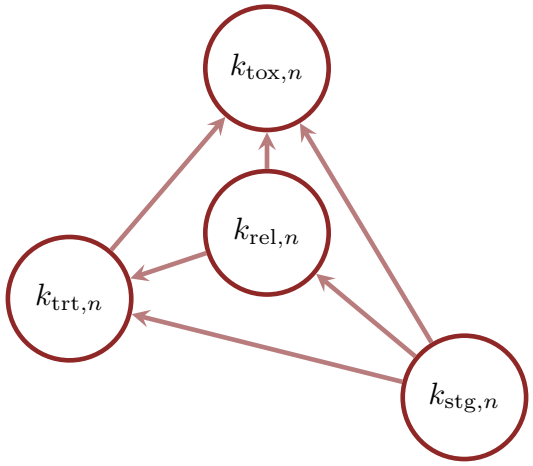


Figure 4: Conditionally decomposing the joint patient characteristic model allows us to focus on smaller, more interpretable component models.

will assume that treatment within this particular cohort is determined entirely by clinicians and hence is largely independent of relationship status,

$$p(\text{treatment} \mid \text{relationship}, \text{stage}) = p(\text{treatment} \mid \text{stage}).$$

With these simplifications our patient characteristic model becomes (Figure 5)

$$\begin{aligned} p(&\text{relationship}, \text{stage}, \text{treatment}, \text{toxicity}) \\ &= p(\text{toxicity} \mid \text{stage}, \text{treatment}) \\ &\quad \cdot p(\text{treatment} \mid \text{stage}) \\ &\quad \cdot p(\text{relationship} \mid \text{stage}) \\ &\quad \cdot p(\text{stage}). \end{aligned}$$

Finally some of the conditional probabilities are known precisely. If treatment always follows a cancer diagnosis then

$$p(\text{treatment} \mid \text{no cancer}) = p(k_{\text{trt}} = 2 \mid k_{\text{stg}} = 1) = 0,$$

or equivalently

$$p(\text{no treatment} \mid \text{no cancer}) = p(k_{\text{trt}} = 1 \mid k_{\text{stg}} = 1) = 1.$$

Similarly without an active treatment there cannot be any treatment toxicity. Consequently

$$p(\text{no toxicity} \mid \text{stage}, \text{treatment}) = p(k_{\text{tox}} = 1 \mid k_{\text{stg}}, k_{\text{trt}} = 1) = 1$$

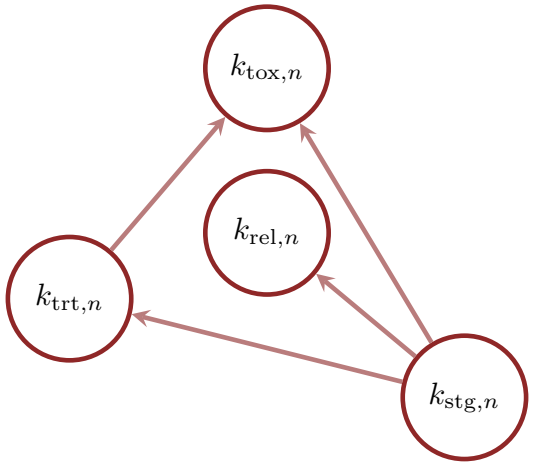


Figure 5: Our domain expertise eliminates some of the conditional dependencies, simplifying this conditionally decomposition of the joint patient characteristic model.

for all stages.

In practice all of the remaining discrete probabilities can be implemented as simplices. For example $p(\text{stage})$ contains three probabilities, one for each stage, which can be implemented with a three-component simplex variable

$$\mathbf{q}_{\text{stg}} = (q_{\text{stg}=1}, q_{\text{stg}=2}, q_{\text{stg}=3})$$

with

$$0 \leq q_{\text{stg}=k} \leq 1$$

and

$$\sum_{k=1}^{K_{\text{stg}}} q_{\text{stg}=k} = 1.$$

Similarly $p(\text{relationship} \mid \text{stage})$ can be implemented with three, two-component simplex variables

$$\begin{aligned}
\mathbf{q}_{\text{rel}|\text{stg}=1} &= (q_{\text{rel}=1|\text{stg}=1}, q_{\text{rel}=2|\text{stg}=1}) \\
\mathbf{q}_{\text{rel}|\text{stg}=2} &= (q_{\text{rel}=1|\text{stg}=2}, q_{\text{rel}=2|\text{stg}=2}) \\
\mathbf{q}_{\text{rel}|\text{stg}=2} &= (q_{\text{rel}=1|\text{stg}=3}, q_{\text{rel}=2|\text{stg}=3}).
\end{aligned}$$

The remaining patient characteristics probabilities have to be inferred from observed data

using an appropriate categorical observational model (Figure 6),

$$\begin{aligned}
&\text{categorical}(k_{\text{stg},n} \mid \mathbf{q}_{\text{stg}}) \\
&\text{categorical}(k_{\text{rel},n} \mid \mathbf{q}_{\text{rel}|\text{stg}=k_{\text{stg},n}}) \\
&\text{categorical}(k_{\text{trt},n} \mid \mathbf{q}_{\text{trt}|\text{stg}=k_{\text{stg},n}}) \\
&\text{categorical}(k_{\text{tox},n} \mid \mathbf{q}_{\text{tox}|\text{stg}=k_{\text{stg},n},\text{rel}=k_{\text{rel},n}}).
\end{aligned}$$

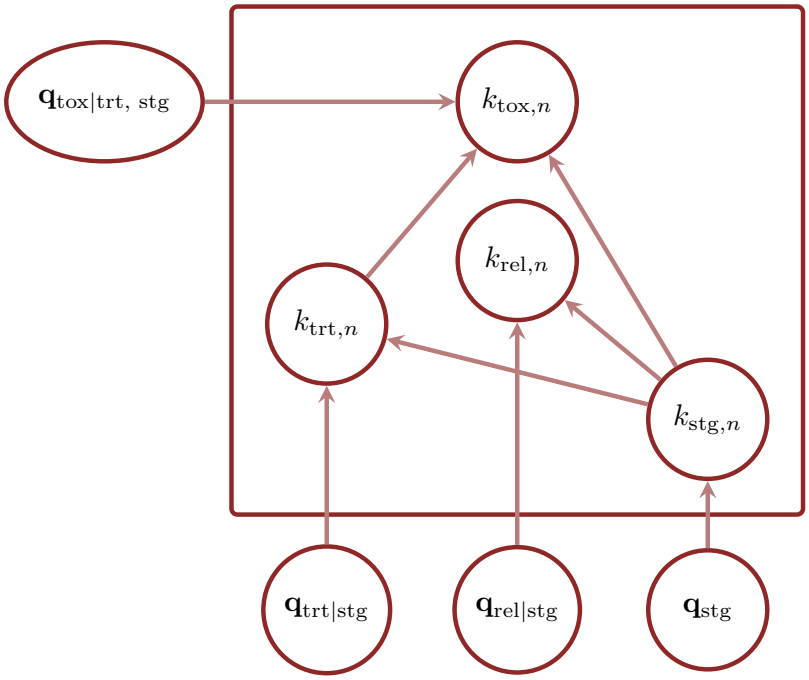


Figure 6: The joint patient model is parameterized by a collection of conditional probabilities, each of which can be implemented with simplex variable types.

5.1.2 Patient Characteristic Prior Model

The Dirichlet family of probability density functions provides a convenient diversity of probabilistic models over simplices, which in turn is particularly useful for developing a principled prior model for our patient characteristic model.

For example the Dirichlet probability density function

$$\text{Dirichlet}(q_1, \ldots, q_K \mid \gamma_1, \ldots, \gamma_K)$$

with $\gamma_k = 1$ for all k is uniform over the $(K-1)$-simplex. This is useful when we have limited domain expertise.

On the other hand the Dirichlet model with

$$\gamma_k = \rho_k / \tau + 1$$

concentrates around the baseline probabilities

$$\rho_1, \cdots, \rho_K$$

with the value τ determining the strength of that concentration. These configurations are useful when our domain expertise is more informative.

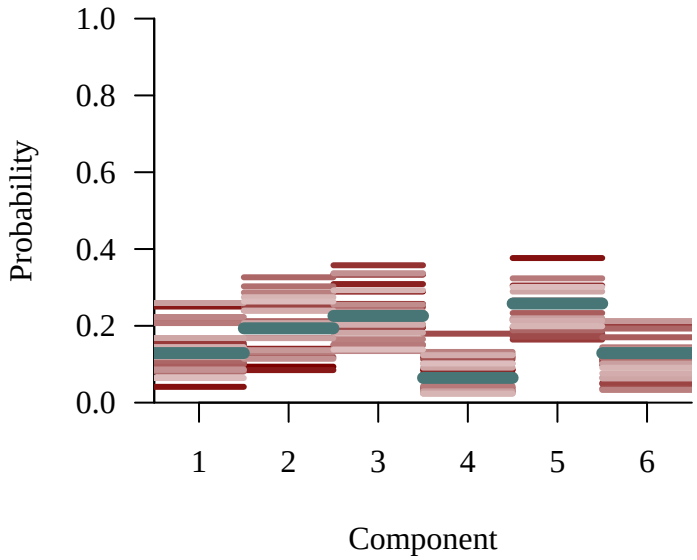
The Dirichlet family can also be configured to concentrate on more extreme behaviors. For example if any of the γ_k are less than one then the resulting Dirichlet probability density function will concentrate on at least one simplex boundary.

When the properties of any particular Dirichlet model are not clear from inspecting the `gamma` parameters we can always build intuition by studying samples.

```
library(colormap)
nom_colors <- c("#DCBCBC", "#C79999", "#B97C7C",
                "#A25050", "#8F2727", "#7C0000")
line_colors <- colormap(colormap=nom_colors, nshades=25)

plot_simplex_samples <- function(gammas, baseline=FALSE, main="") {
  K <- length(gammas)

  plot(1, type="n", main=main,
       xlim=c(0.5, K + 0.5), xlab="Component",
       ylim=c(0, 1), ylab="Probability")

  idxs <- rep(1:K, each=2)

  xs <- sapply(1:length(idxs),
               function(k) if(k %% 2 == 1) idxs[k] - 0.5
                           else            idxs[k] + 0.5)

  for (s in 1:25) {
    q <- rgamma(K, gammas, 1)
    q <- q / sum(q)

    for (k in 1:K) {
      idx1 <- 2 * k - 1
```

```
      idx2 <- 2 * k
      lines(xs[idx1:idx2], rep(q[k], 2), col=line_colors[s], lwd=3)
    }
  }

  if (baseline) {
    rho <- (gammas - 1)
    rho <- rho / sum(rho)
    for (k in 1:K) {
      idx1 <- 2 * k - 1
      idx2 <- 2 * k
      lines(xs[idx1:idx2], rep(rho[k], 2), col=util$c_mid_teal, lwd=6)
      lines(xs[idx1:idx2], rep(rho[k], 2), col=util$c_mid_teal, lwd=3)
    }
  }
}
```

```
par(mfrow=c(1, 1), mar=c(5, 5, 5, 1))

plot_simplex_samples(c(5, 7, 8, 3, 9, 5), baseline=TRUE)
```



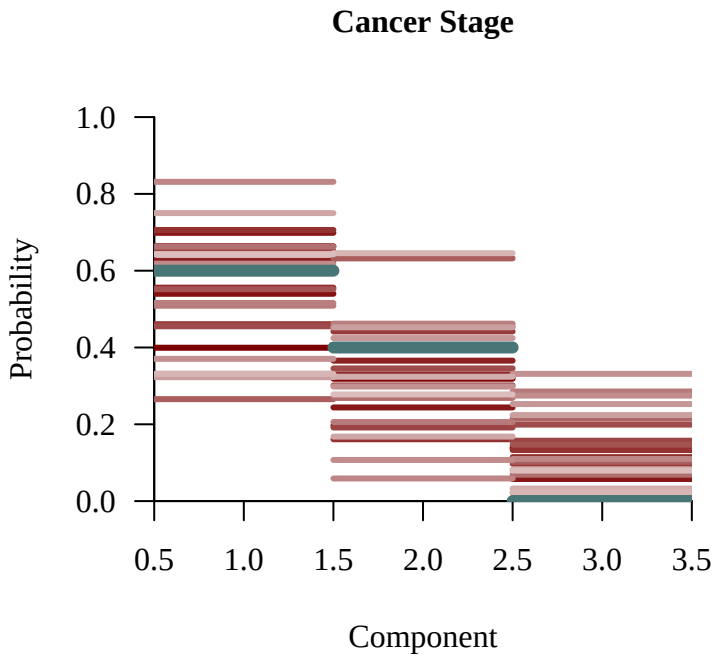
What do we know about the cohort?

Because patients were included only from referrals for fertility preservation consultation, patients with cancer diagnoses should not dominate. The more domain expertise we have about

referral rates the more precisely we can tune an appropriate prior model; here we will assume a weak concentration towards no cancer diagnosis,

$$\gamma_{\text{stg}} = \frac{(\frac{3}{5}, \frac{2}{5}, 0)}{\frac{1}{5}} + \mathbf{1} = (4, 3, 1)\,.$$

```
par(mfrow=c(1, 1), mar=c(5, 5, 5, 1))

plot_simplex_samples(c(4, 3, 1), baseline=TRUE, main="Cancer Stage")
```

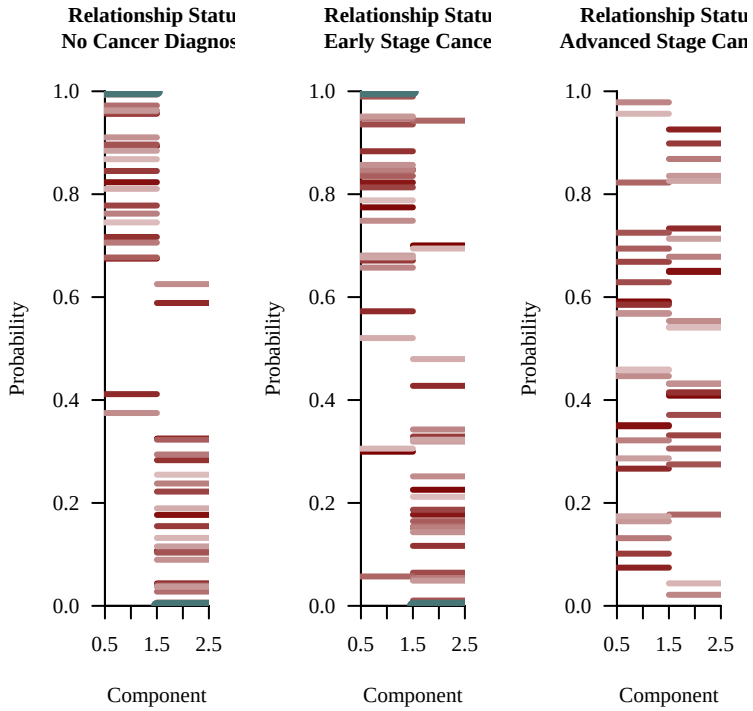


In general relationships are strained more and more as cancer progresses to more advanced stages, although here we don't have too much a priori understanding of just how much,

$$\begin{aligned}
\gamma_{\text{rel}|\text{stg}=1} &= \frac{(1, 0)}{\frac{1}{3}} + \mathbf{1} = (4, 1)\,, \\
\gamma_{\text{rel}|\text{stg}=2} &= \frac{(1, 0)}{1} + \mathbf{1} = (2, 1)\,, \\
\gamma_{\text{rel}|\text{stg}=3} &= (1, 1)\,.
\end{aligned}$$

```
par(mfrow=c(1, 3), mar=c(5, 5, 5, 1))

plot_simplex_samples(c(4, 1), baseline=TRUE,
                     main="Relationship Status\nNo Cancer Diagnosis")
```

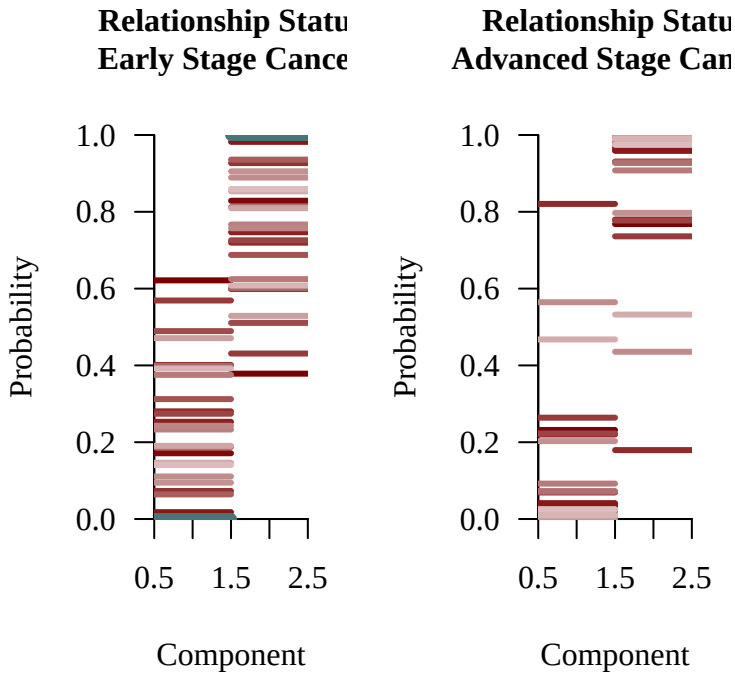
```
plot_simplex_samples(c(2, 1), baseline=TRUE,
                     main="Relationship Status\nEarly Stage Cancer")

plot_simplex_samples(c(1, 1), baseline=FALSE,
                     main="Relationship Status\nAdvanced Stage Cancer")
```



Similarly we expect that treatment becomes more likely as cancer progresses,

$$\gamma_{\text{trt}|\text{stg}=2} = \frac{(0,1)}{\frac{1}{2}} + \mathbf{1} = (1,3)\,,$$
$$\gamma_{\text{trt}|\text{stg}=3} = (0.5,4)\,.$$

```
par(mfrow=c(1, 2), mar=c(5, 5, 5, 1))

plot_simplex_samples(c(1, 3), baseline=TRUE,
                     main="Relationship Status\nEarly Stage Cancer")

plot_simplex_samples(c(0.5, 4), baseline=FALSE,
                     main="Relationship Status\nAdvanced Stage Cancer")
```



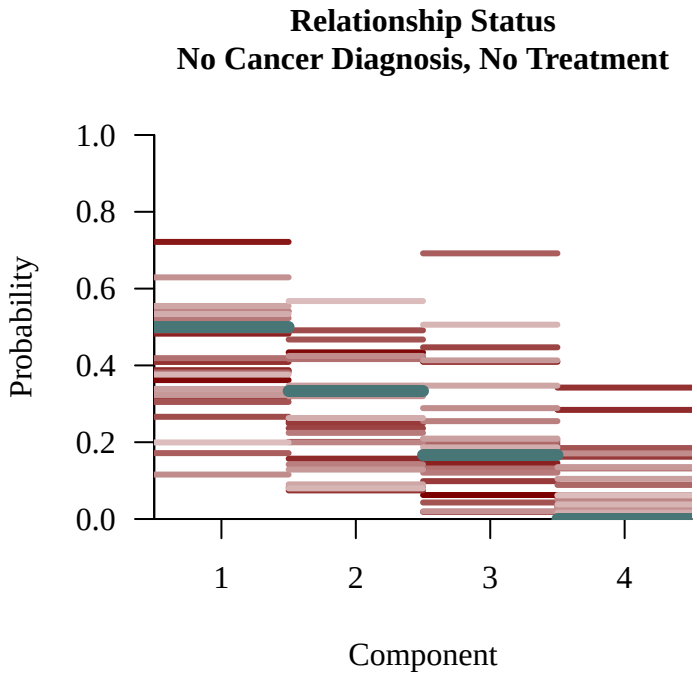
Finally toxicity should increase with cancer stage, but only if a patient is being actively treated,

$$\gamma_{\text{tox}|\text{stg}=2,\text{trt}=2} = \frac{\left(\frac{3}{6}, \frac{2}{6}, \frac{1}{6}, 0\right)}{\frac{1}{6}} + \mathbf{1} = (4, 3, 2, 1)\,,$$

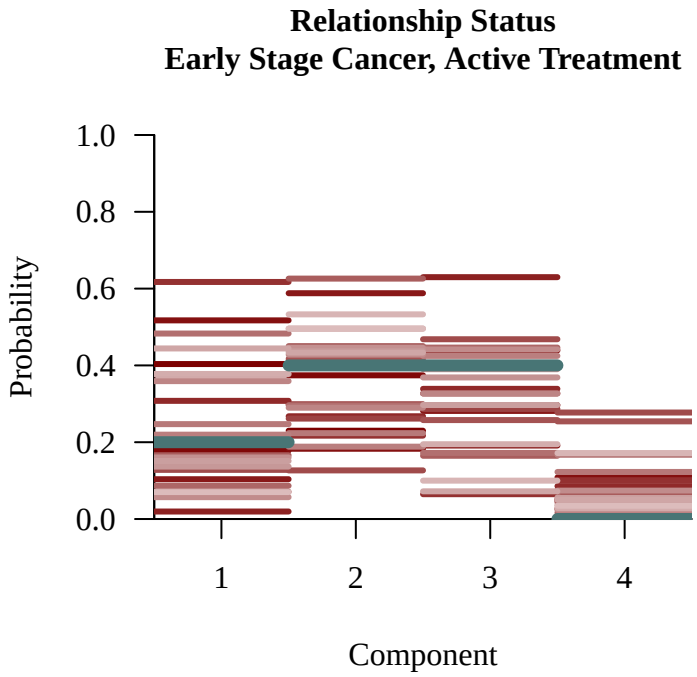
$$\gamma_{\text{tox}|\text{stg}=2,\text{trt}=2} = \frac{\left(\frac{1}{5}, \frac{2}{5}, \frac{2}{5}, 0\right)}{\frac{1}{5}} + \mathbf{1} = (2, 3, 3, 1)\,,$$

$$\gamma_{\text{tox}|\text{stg}=3,\text{trt}=2} = \frac{\left(0, \frac{1}{5}, \frac{2}{5}, \frac{2}{5}\right)}{\frac{1}{5}} + \mathbf{1} = (1, 2, 3, 3)\,.$$

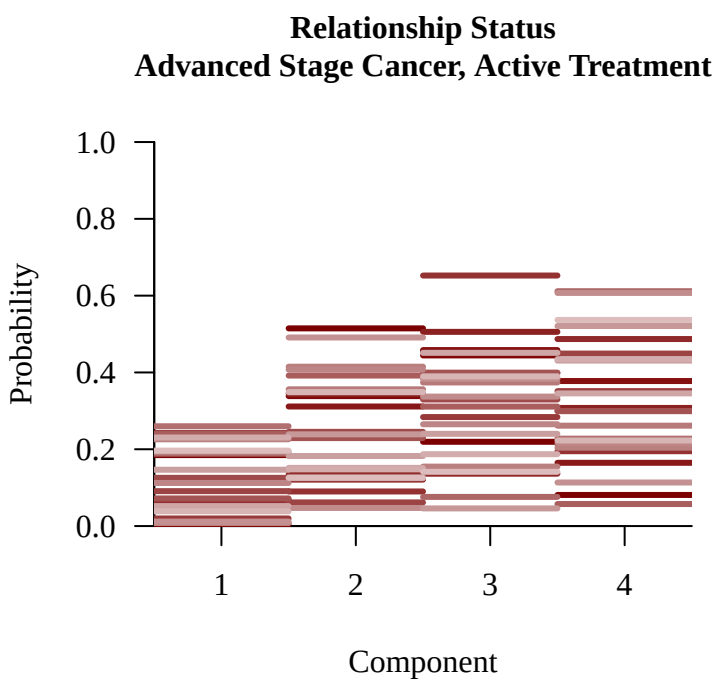
```
par(mfrow=c(1, 1), mar=c(5, 5, 5, 1))

title <- "Relationship Status\nNo Cancer Diagnosis, No Treatment"
plot_simplex_samples(c(4, 3, 2, 1), baseline=TRUE,
                     main=title)
```




```
par(mfrow=c(1, 1), mar=c(5, 5, 5, 1))

title <- "Relationship Status\nEarly Stage Cancer, Active Treatment"
plot_simplex_samples(c(2, 3, 3, 1), baseline=TRUE,
                     main=title)
```




```
par(mfrow=c(1, 1), mar=c(5, 5, 5, 1))
title <- "Relationship Status\nAdvanced Stage Cancer, Active Treatment"
plot_simplex_samples(c(1, 2, 3, 3), baseline=FALSE,
                     main=title)
```



Overall the full prior model is relatively diffuse, but it does suppress the more unrealistic patient characteristics.

5.1.3 Posterior Quantification

If the fertility heterogeneity is independent of how the patient characteristic groups are populated then our previous model and new patient characteristic model simply compose together (Figure 7). This is by no means a trivial assumption in practice. For example it would be violated if referrals into the observed cohort favored patients who suffered from fertility issues.

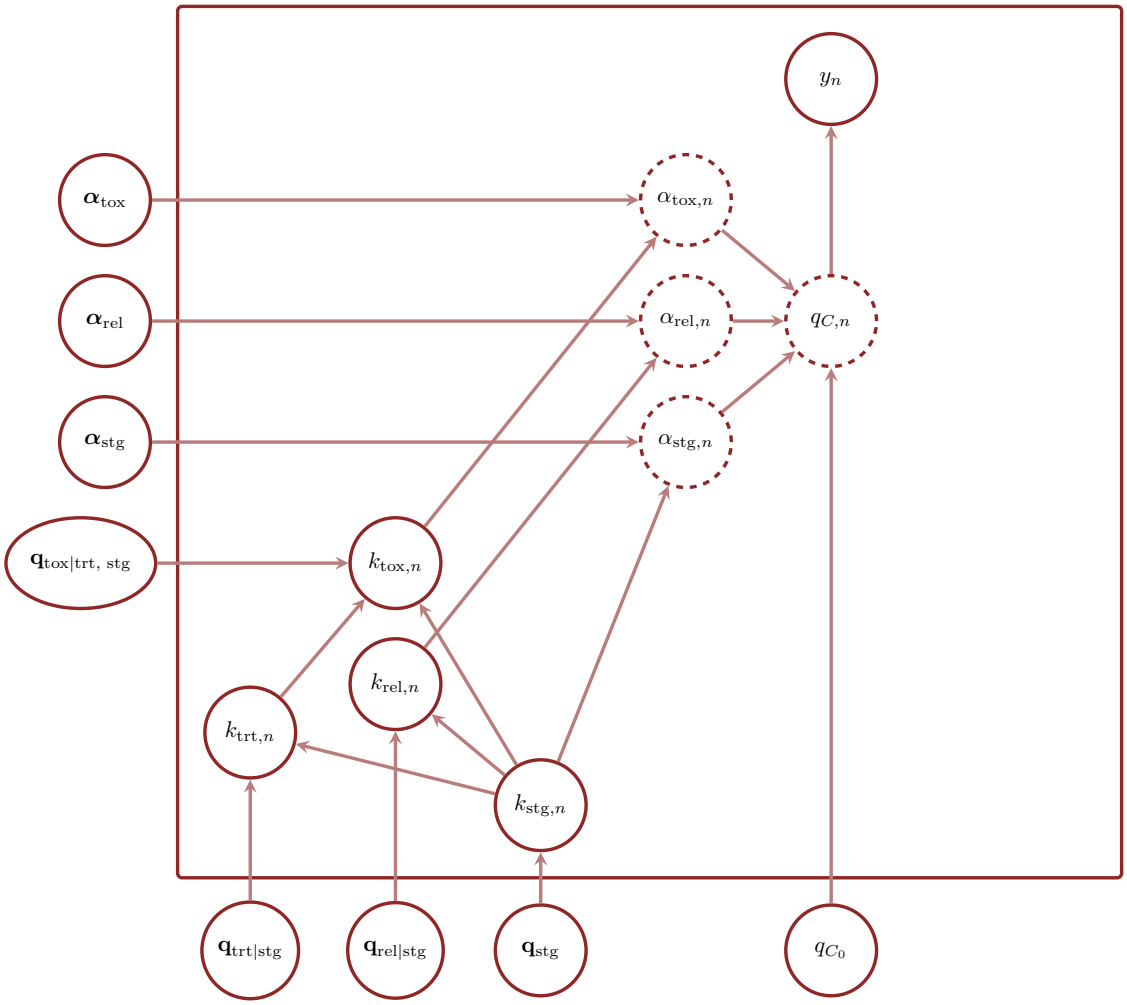


Figure 7: When fertility is independent of how the cohort was selected we can simply compose a patient fertility model with a patient characteristic model.

All of these patient characteristic probabilities can be conveniently implemented with the `simplex` variable types in Stan. The only challenge is organizing all of the simplices effectively.

Also note that when we generate posterior predictive simulations in the `generated quantities` block we simulate new patient characteristics as well as new fertility outcomes.

```
data$N_pred <- 5000

fit <- stan(file="stan_programs/model3a.stan",
            data=data, seed=8438338,
            warmup=1000, iter=2024, refresh=0)
```

Although our model is quickly increasing in complexity the computational diagnostics suggest that our posterior computation is keeping up.

```
diagnostics <- util$extract_hmc_diagnostics(fit)
util$check_all_hmc_diagnostics(diagnostics)
```

```
  All Hamiltonian Monte Carlo diagnostics are consistent with reliable
Markov chain Monte Carlo.
```

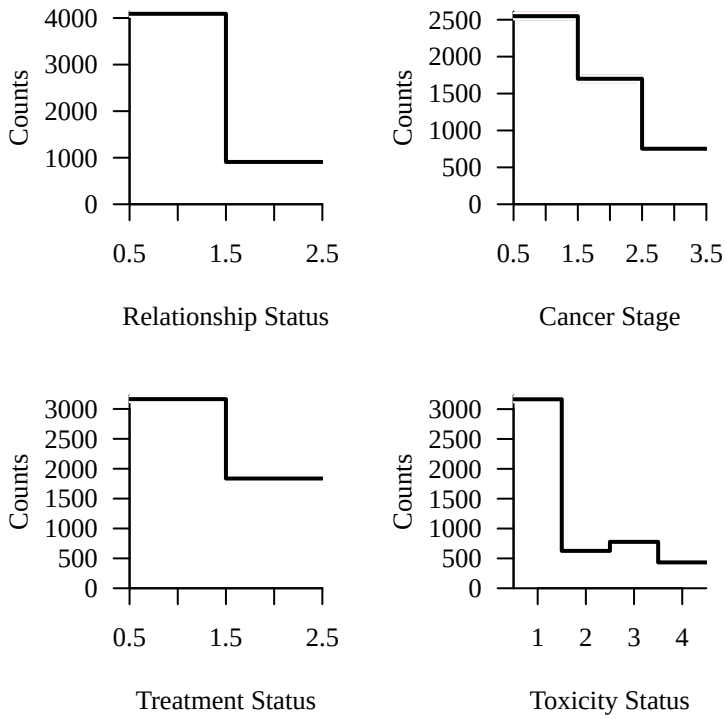
```
samples3a <- util$extract_expectand_vals(fit)
base_samples <- util$filter_expectands(samples3a,
                                       c('q_stg', 'q_trt_active_stg',
                                         'q_tox_active_trt','q_C_0',
                                         'alpha_rel', 'alpha_stg',
                                         'alpha_tox'),
                                       check_arrays=TRUE)
util$check_all_expectand_diagnostics(base_samples)
```

```
All expectands checked appear to be behaving well enough for reliable
Markov chain Monte Carlo estimation.
```

5.1.4 Retrodictive Checks

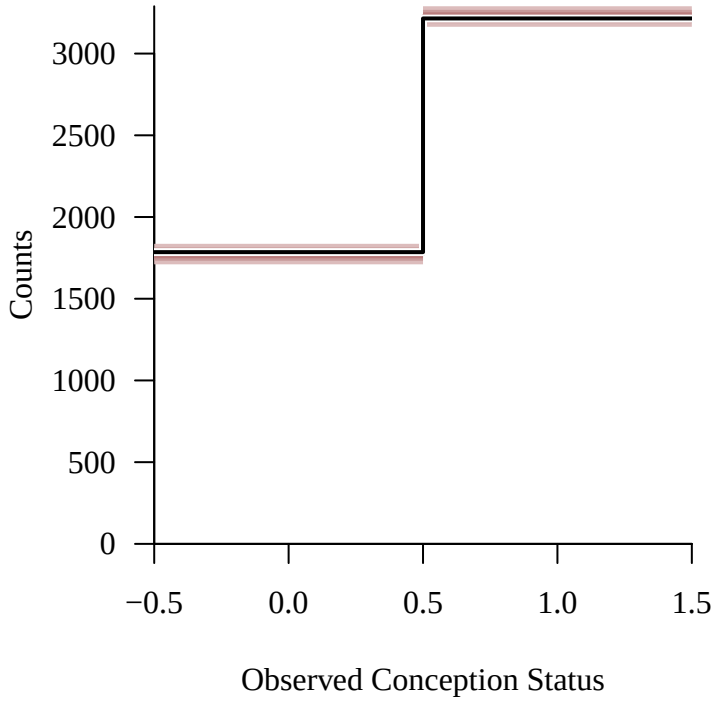
Posterior retrodictive checks for the patient characteristics all look good, suggesting that the assumptions underlying our patient characteristic model is adequately capturing the details of the observed data.

```
par(mfrow=c(2, 2), mar=c(5, 5, 1, 1))

util$plot_hist_quantiles(samples3a, 'k_rel_pred',
                         0.5, data$K_rel + 0.5, 1,
                         baseline_values=data$k_rel,
                         xlab="Relationship Status")

util$plot_hist_quantiles(samples3a, 'k_stg_pred',
                         0.5, data$K_stg + 0.5, 1,
                         baseline_values=data$k_stg,
                         xlab="Cancer Stage")

util$plot_hist_quantiles(samples3a, 'k_trt_pred',
                         0.5, data$K_trt + 0.5, 1,
                         baseline_values=data$k_trt,
                         xlab="Treatment Status")

util$plot_hist_quantiles(samples3a, 'k_tox_pred',
                         0.5, data$K_tox + 0.5, 1,
                         baseline_values=data$k_tox,
                         xlab="Toxicity Status")
```



Moreover the observed conception status is consistent with the full posterior predictive conception status, where we consider variation in the clinical, demographic, and fertility behaviors.

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

util$plot_hist_quantiles(samples3a, 'y_pred', -0.5, 1.5, 1,
                         baseline_values=data$y,
                         xlab="Observed Conception Status")
```



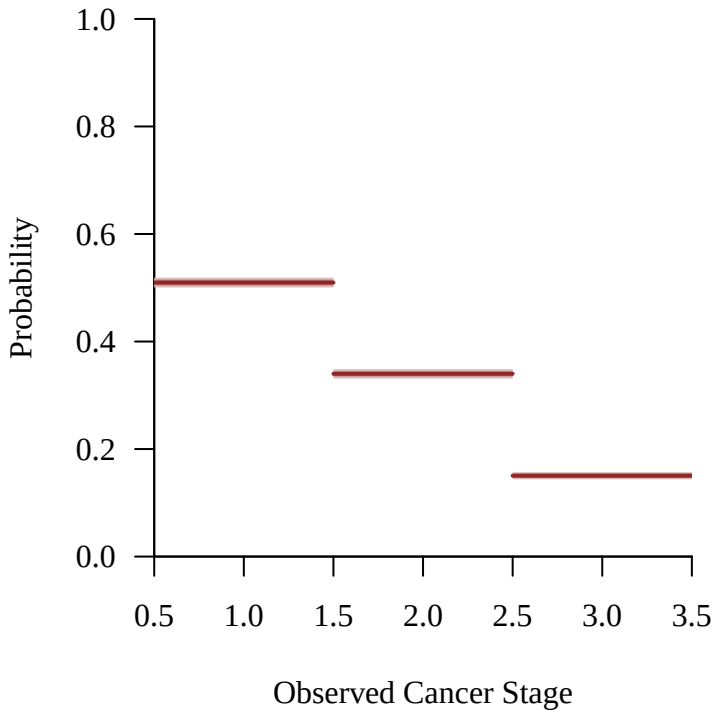
Note that we cannot consider conditional retrodictive checks here as we did previously because the observed data and posterior predictions are now based on *different* realizations of the patient characteristic variables.

5.1.5 Posterior Insights

Content with the adequacy of our model we can examine the patient characteristic inferences one by one. About half of the observed cohort suffers from cancer.

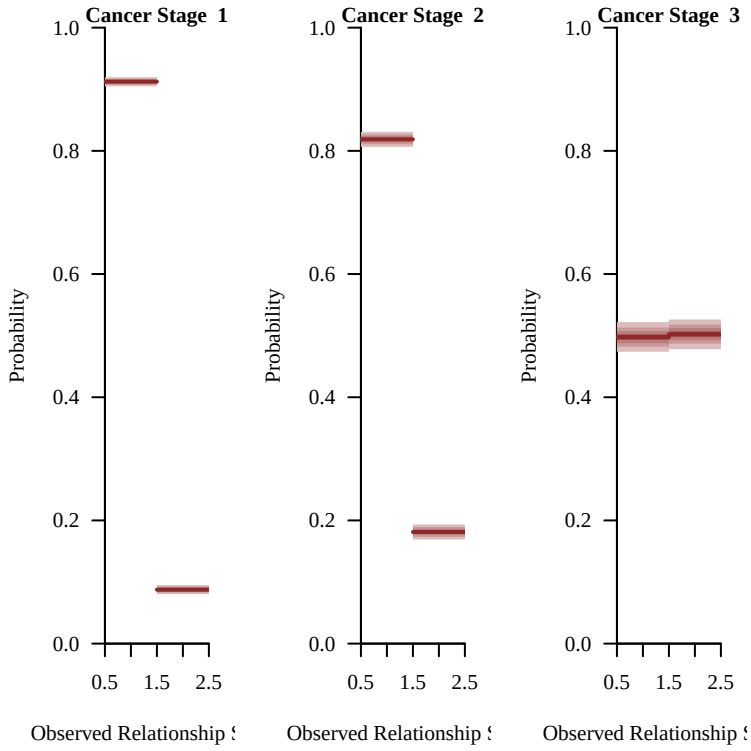
```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

names <- sapply(1:data$K_stg,
```

```
                function(k) paste0('q_stg[', k, ']'))
util$plot_disc_pushforward_quantiles(samples3a, names,
                                     xlab="Observed Cancer Stage",
                                     ylab="Probability",
                                     display_ylim=c(0, 1))
```



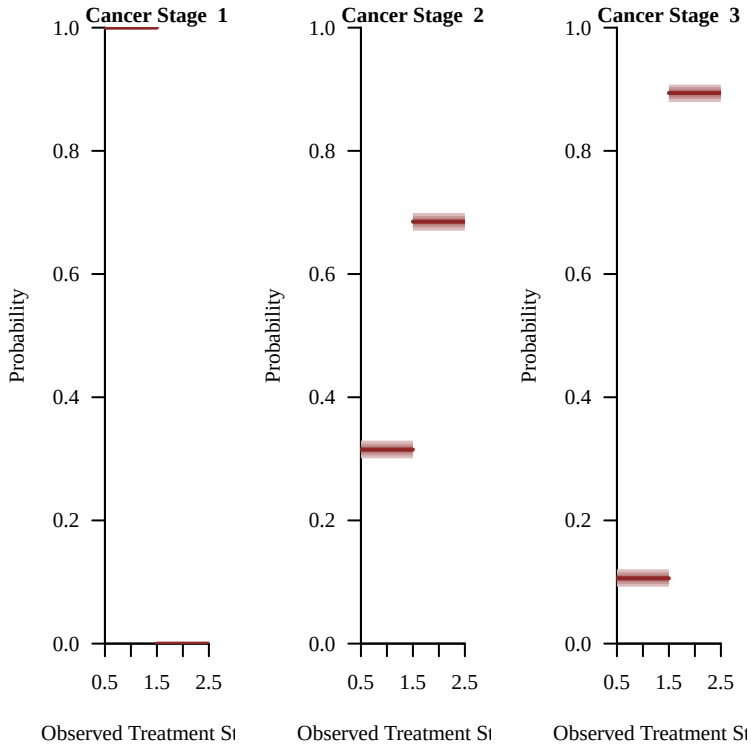
Stable relationships become less common as cancer progresses.

```
par(mfrow=c(1, 3), mar=c(5, 5, 1, 1))

for (ks in 1:data$K_stg) {
  names <- sapply(1:data$K_rel,
                  function(k) paste0('q_rel[', ks, ',', k, ']'))
  util$plot_disc_pushforward_quantiles(samples3a, names,
                                       xlab="Observed Relationship Status",
                                       ylab="Probability",
                                       display_ylim=c(0, 1),
                                       main=paste("Cancer Stage ", ks))
}
```



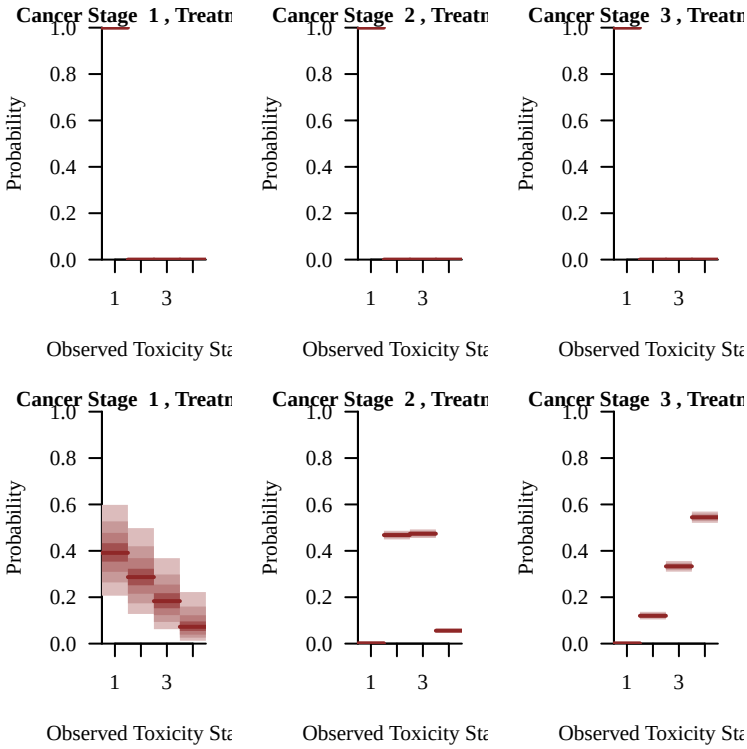
Conversely treatment becomes more common as cancer progresses.

```
par(mfrow=c(1, 3), mar=c(5, 5, 1, 1))

for (ks in 1:data$K_stg) {
  names <- sapply(1:data$K_trt,
                  function(k) paste0('q_trt[', ks, ',', k, ']'))
  util$plot_disc_pushforward_quantiles(samples3a, names,
                                       xlab="Observed Treatment Status",
                                       ylab="Probability",
                                       display_ylim=c(0, 1),
                                       main=paste("Cancer Stage ", ks))
}
```



Treatment toxicity becomes more severe as cancer progresses, at least for patients actively undergoing treatment.

```
par(mfrow=c(2, 3), mar=c(5, 5, 1, 1))

for (kt in 1:data$K_trt) {
  for (ks in 1:data$K_stg) {
    names <- sapply(1:data$K_tox,
                    function(k) paste0('q_tox[', ks, ',', kt, ',', k, ']'))
    util$plot_disc_pushforward_quantiles(samples3a, names,
                                         xlab="Observed Toxicity Status",
                                         ylab="Probability",
                                         display_ylim=c(0, 1),
                                         main=paste("Cancer Stage ", ks,
                                                    ", Treatment ",  kt))
  }
}
```

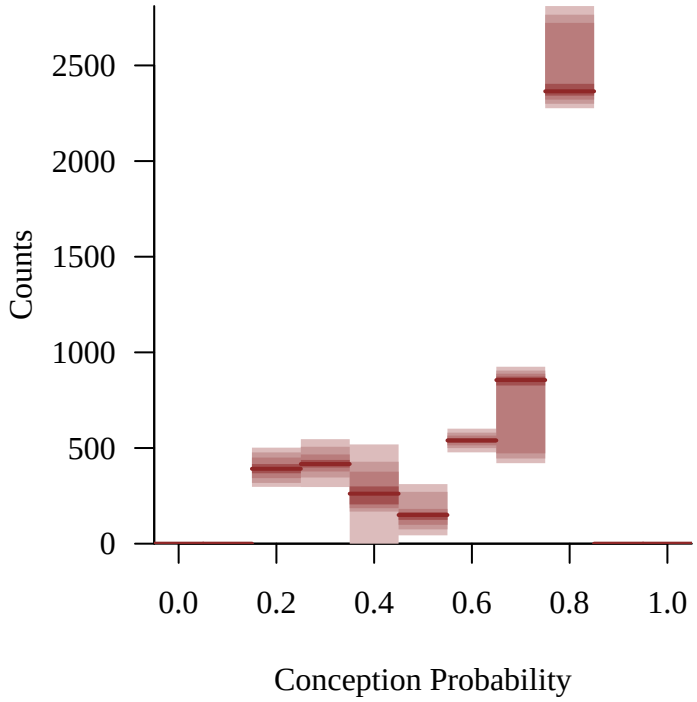


Note that `k_stg = 1` and `k_trt = 2` corresponds to cancer treatment without a cancer diagnosis, which is not possible in this scenario. Consequently we have no observations with these particular patient characteristics, and the corresponding toxicity probabilities are informed by only the prior model. This is why these inferences exhibit the largest uncertainties.

In order to quantify the effect of potential interventions we need to compare the fertility of the entire observed cohort to some new, hypothetical cohort. This is straightforward when fertility in both cohorts behaves homogeneously across patients but substantially more subtle when patient fertility is heterogeneous. In the heterogeneous case we to reduce the population behavior into a summary amenable to direct comparison.

For example we might summarize the cohort population behavior with a probability distribution of histograms. This summary convolves the variation in behavior across individual patients and the inferential uncertainties of those behaviors together into a single probabilistic prediction.

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

util$plot_hist_quantiles(samples3a, 'q_pred', -0.05, 1.05, 0.1,
                         xlab="Conception Probability")
```



Notice the peak at high conception probabilities here; this is due to the baseline individuals in the population. If useful we can always stratify these histograms to isolate particular patient sub-populations of interest, such as the baseline patients.

Another approach is to reduce the entire population to a scalar summary and then visualize the posterior predictive distribution of that summary.

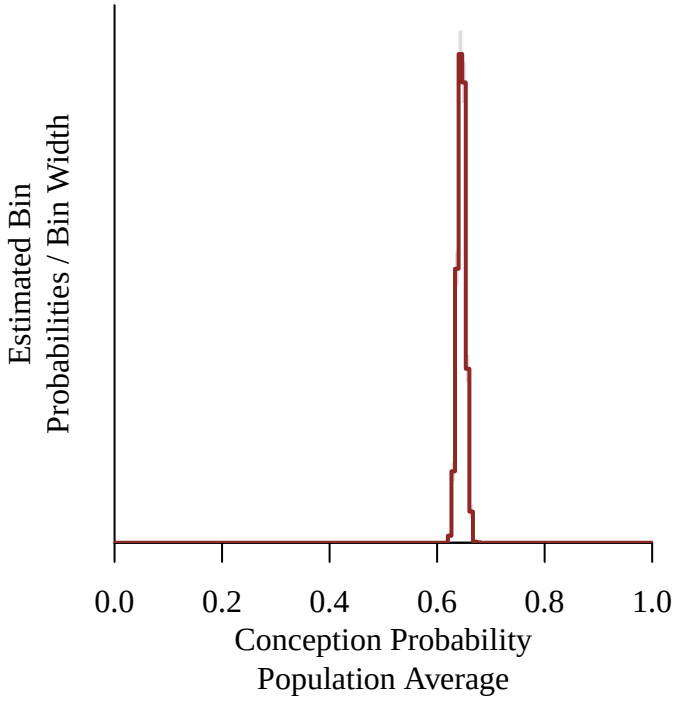
For instance we might consider ensemble, or population, average of the individual patient fertilities,

$$\langle q \rangle = \frac{1}{N} \sum_{n=1}^{N} q_{C,n},$$

and then analyze the posterior distribution of $\langle q \rangle$. This is particularly useful for emphasizing the centrality of the population.

```
var_repl <- list('p'=util$name_array('q_pred', c(data$N_pred)))

pop_ave_samples <-
  util$eval_expectand_pushforward(samples3a,
                                  function(p) mean(p),
                                  var_repl)

par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))
```

```
name <- "Conception Probability\nPopulation Average"
util$plot_expectand_pushforward(pop_ave_samples,
                                150, flim=c(0, 1),
                                display_name=name)
```

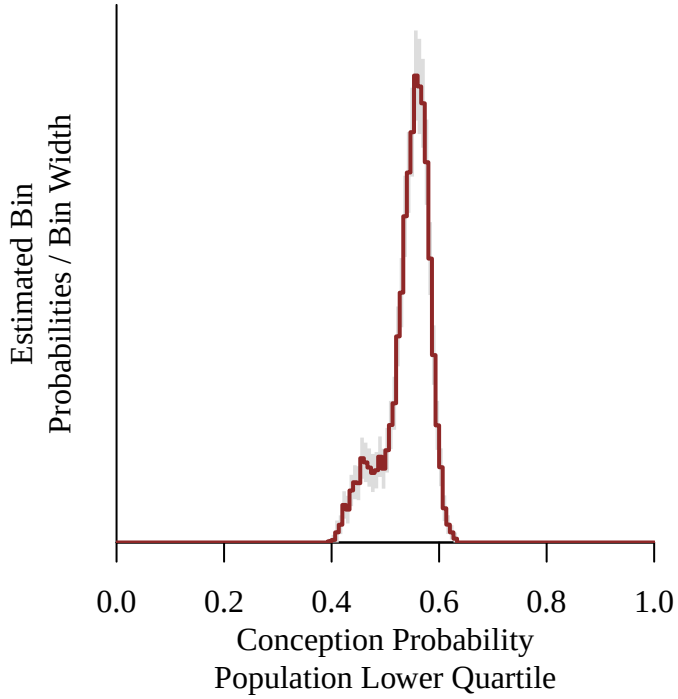


In cases where we are more interested in extremes of the population we might consider ensemble/population quantiles such as the lower quartile Q defined implicitly as

$$\frac{\sum_{n=1}^{N} I[Q > q_{C,n}]}{N} \approx 0.25.$$

```
var_repl <- list('p'=util$name_array('q_pred', c(data$N_pred)))

pop_quant_samples <-
  util$eval_expectand_pushforward(samples3a,
                                  function(p) quantile(p, prob=c(0.25)),
                                  var_repl)

par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

name <- "Conception Probability\nPopulation Lower Quartile"
util$plot_expectand_pushforward(pop_quant_samples,
```

```
                                    150, flim=c(0, 1),
                                    display_name=name)
```



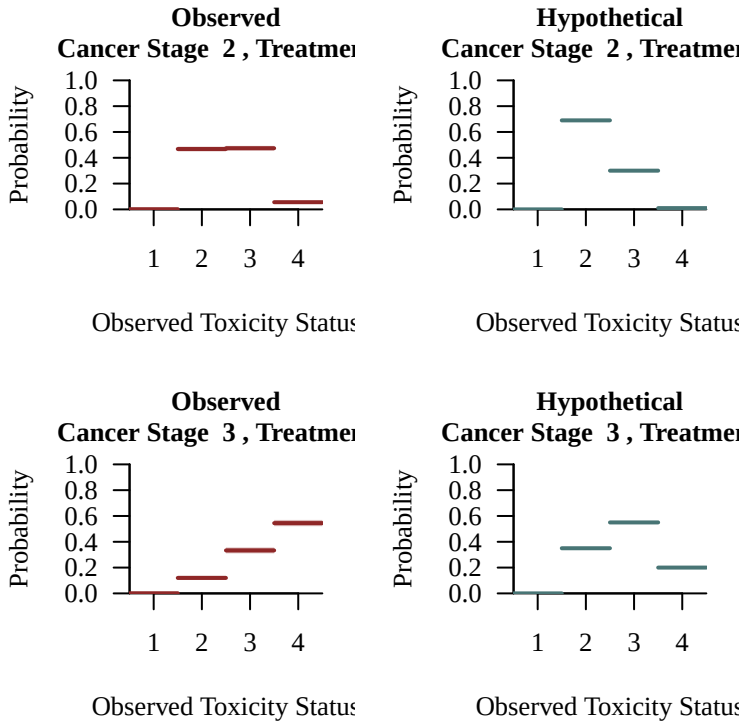
5.2 Model 3b

Now that we've modeled the patient characteristic of the observed cohort we can consider what behaviors might manifest in other, hypothetical populations of practical interest.

For example what would happen to the population fertility with the introduction of a new, less toxic treatment? Because none of the other patient characteristics depend on toxicity we can model this intervention by changing only $\mathbf{q}_{\text{tox}|\text{trt, stg}}$.

```
q_tox_hyp <- list(c(0, 0.69, 0.30, 0.01),
                  c(0, 0.35, 0.55, 0.20) )

par(mfrow=c(2, 2), mar=c(5, 5, 3, 1))

kt <- 2
for (ks in 2:data$K_stg) {
  names <- sapply(1:data$K_tox,
                  function(k) paste0('q_tox[', ks, ',', kt, ',', k, ']'))
```

```
  util$plot_disc_pushforward_quantiles(samples3a, names,
                                       xlab="Observed Toxicity Status",
                                       ylab="Probability",
                                       display_ylim=c(0, 1),
                                       main=paste("Observed\n",
                                                  "Cancer Stage ", ks,
                                                  ", Treatment ",  kt))

  plot(0, type='n',
       main=paste("Hypothetical\n",
                  "Cancer Stage ", ks,
                  ", Treatment ",  kt),
       xlim=c(0.5, data$K_tox + 0.5),
       xlab="Observed Toxicity Status",
       ylim=c(0, 1), ylab="Probability")
  for (k in 1:data$K_tox) {
    lines(c(k - 0.5, k + 0.5),
          rep(q_tox_hyp[[ks - 1]][k], 2),
          lwd=2, col=util$c_mid_teal)
  }
}
```



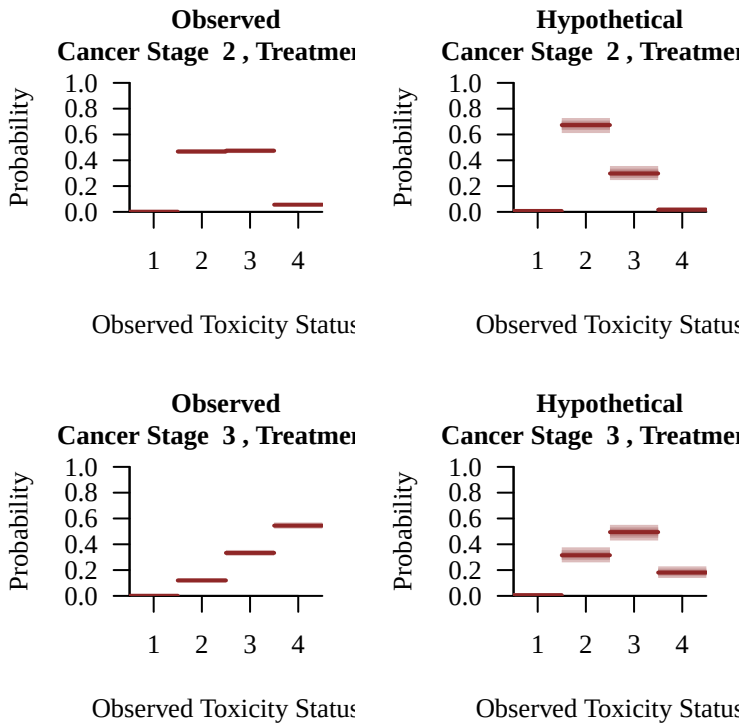
Because we have embraced the probabilistic glory of Bayesian inference there's no reason why we have to be limited to point hypotheticals. We can also use a probability distribution to quantify uncertainty in the potential behaviors instead of having to rely on precise values.

```
tau_hyp <- 0.01

alpha_tox_hyp <- list()
for (k in 1:2) {
  alpha_tox_hyp[[k]] <- q_tox_hyp[[k]] / tau_hyp + rep(1, data$K_tox)
}
```

```
S <- 1000
samples_hyp <- list()

for (ks in 2:data$K_stg) {
  for (k in 1:data$K_tox) {
    name <- paste0('q_tox[', ks, ',', k, ']')
    samples_hyp[[name]] <- matrix(0, nrow=1, ncol=S)
  }
  for (s in 1:S) {
    q <- rgamma(data$K_tox, alpha_tox_hyp[[ks - 1]], 1)
    q <- q / sum(q)
    for (k in 1:data$K_tox) {
      name <- paste0('q_tox[', ks, ',', k, ']')
      samples_hyp[[name]][1, s]  <- q[k]
    }
  }
}
```

```
par(mfrow=c(2, 2), mar=c(5, 5, 3, 1))

kt <- 2
for (ks in 2:data$K_stg) {
  names <- sapply(1:data$K_tox,
                  function(k) paste0('q_tox[', ks, ',', kt, ',', k, ']'))
  util$plot_disc_pushforward_quantiles(samples3a, names,
                                       xlab="Observed Toxicity Status",
                                       ylab="Probability",
                                       display_ylim=c(0, 1),
                                       main=paste("Observed\n",
                                                  "Cancer Stage ", ks,
                                                  ", Treatment ",  kt))
```

```
  names <- sapply(1:data$K_tox,
                  function(k) paste0('q_tox[', ks, ',', k, ']'))
  util$plot_disc_pushforward_quantiles(samples_hyp, names,
                                       xlab="Observed Toxicity Status",
                                       ylab="Probability",
                                       display_ylim=c(0, 1),
                                       main=paste("Hypothetical\n",
                                                  "Cancer Stage ", ks,
                                                  ", Treatment ",  kt))
}
```



To propagate these changes forward to updated conception inferences we just need to run the previous model with a modified `generated quantities` block implementing the new inferential quantities.

```
data$alpha_tox_hyp1 <- q_tox_hyp[[1]]
data$alpha_tox_hyp2 <- q_tox_hyp[[2]]

fit <- stan(file="stan_programs/model3b.stan",
            data=data, seed=8438338,
            warmup=1000, iter=2024, refresh=0)
```

The changes to the `generated quantities` block do alter how pseudo-random numbers are used by Stan. This, in turn, can affect the stochastic posterior computation. Consequently we have to double check the computational diagnostics to ensure that no new issues have arisen. Fortunately everything still looks okay.

```
diagnostics <- util$extract_hmc_diagnostics(fit)
util$check_all_hmc_diagnostics(diagnostics)
```

```
  All Hamiltonian Monte Carlo diagnostics are consistent with reliable
Markov chain Monte Carlo.
```

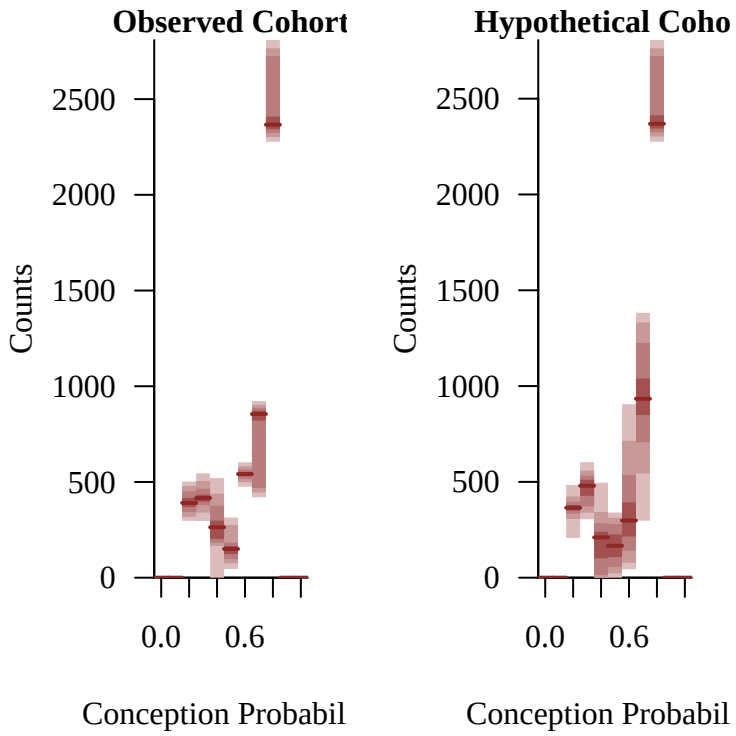
```
samples3b <- util$extract_expectand_vals(fit)
base_samples <- util$filter_expectands(samples3b,
                                       c('q_stg', 'q_trt_active_stg',
                                         'q_tox_active_trt','q_C_0',
                                         'alpha_rel', 'alpha_stg',
                                         'alpha_tox'),
                                       check_arrays=TRUE)
util$check_all_expectand_diagnostics(base_samples)
```

```
All expectands checked appear to be behaving well enough for reliable
Markov chain Monte Carlo estimation.
```

Because the model and data are the same the retrodictive checks will all be equivalent, and we can jump directly to analyzing the posterior inferences. In particular comparing inferences for the population summaries from the observed and hypothetical cohorts allows us to quantify the efficacy of the new treatment.

The behavior of the baseline sub-population remains the same, but the rest of the population shifts to higher conception probabilities after the hypothetical intervention. This makes sense given that the baseline patients are not undergoing treatments and hence are not influenced by any changes in those treatments.

```
par(mfrow=c(1, 2), mar=c(5, 5, 1, 1))

util$plot_hist_quantiles(samples3b, 'q_pred', -0.05, 1.05, 0.1,
                         xlab="Conception Probability",
                         main="Observed Cohort")

util$plot_hist_quantiles(samples3b, 'q_hyp_pred', -0.05, 1.05, 0.1,
                         xlab="Conception Probability",
                         main="Hypothetical Cohort")
```

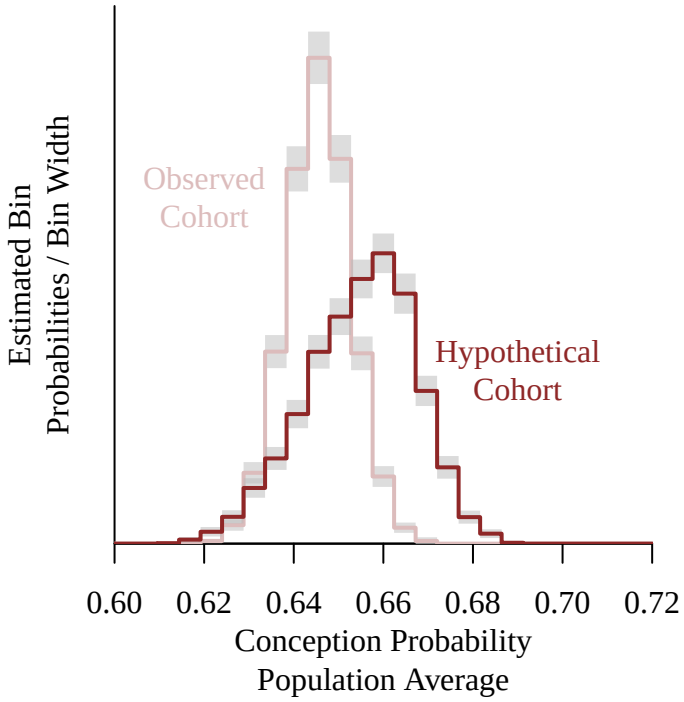


Reducing the two cohorts to average conception probabilities demonstrates the benefit of the new treatment more clearly.

```
var_repl <- list('p'=util$name_array('q_pred', c(data$N_pred)))
pop_ave_samples <-
  util$eval_expectand_pushforward(samples3b,
                                  function(p) mean(p),
                                  var_repl)

var_repl <- list('p'=util$name_array('q_hyp_pred', c(data$N_pred)))
hyp_pop_ave_samples <-
  util$eval_expectand_pushforward(samples3b,
                                  function(p) mean(p),
                                  var_repl)
```

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

name <- "Conception Probability\nPopulation Average"
util$plot_expectand_pushforward(pop_ave_samples,
                                25, flim=c(0.6, 0.72),
                                display_name=name,
```

```
                                    col=util$c_light)
text(0.62, 40, "Observed\nCohort", col=util$c_light)

util$plot_expectand_pushforward(hyp_pop_ave_samples,
                                    25, flim=c(0.6, 0.72),
                                    border="#BBBBBB88",
                                    add=TRUE)
text(0.69, 20, "Hypothetical\nCohort", col=util$c_dark)
```



We can even directly calculate the posterior probability that the average conception probability is higher in the hypothetical cohort.

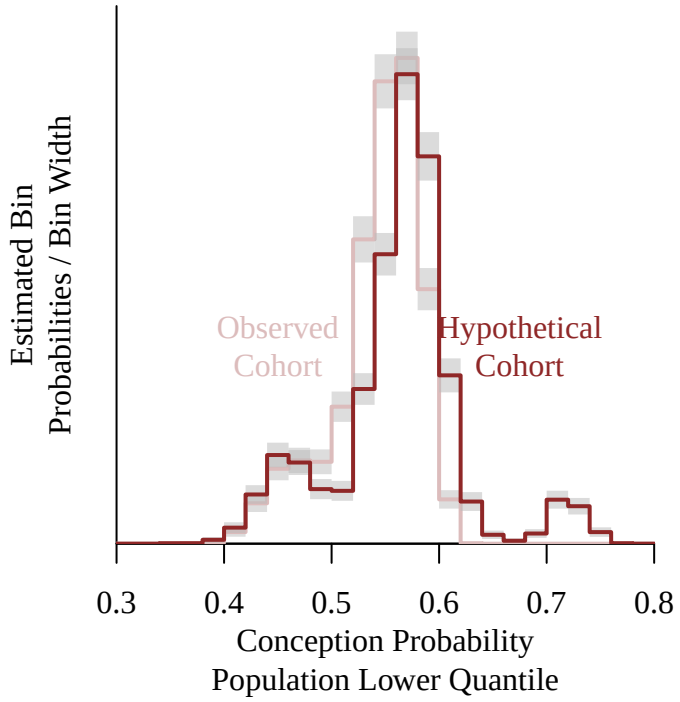
```
ave_samples <- list("obs" = pop_ave_samples,
                    "hyp" = hyp_pop_ave_samples)
p_est <-
  util$implicit_subset_prob(ave_samples,
                            function(obs, hyp) hyp > obs)

format_string <- paste("Posterior probability that hypothetical",
                       "population average\nis greater than observed",
                       "population average = %.3f +/- %.3f.")
cat(sprintf(format_string, p_est[1], 2 * p_est[2]))
```

```
Posterior probability that hypothetical population average
is greater than observed population average = 0.787 +/- 0.013.
```

Interestingly the lower quantile summary exhibits a smaller benefit to the new treatment. This suggests that improvements to the lowest fertility patients are weaker than improvements to the population as a whole.

```
var_repl <- list('p'=util$name_array('q_pred', c(data$N_pred)))
pop_quant_samples <-
  util$eval_expectand_pushforward(samples3b,
                                  function(p) quantile(p, prob=c(0.25)),
                                  var_repl)

var_repl <- list('p'=util$name_array('q_hyp_pred', c(data$N_pred)))
hyp_pop_quant_samples <-
  util$eval_expectand_pushforward(samples3b,
                                  function(p) quantile(p, prob=c(0.25)),
                                  var_repl)
```

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

name <- "Conception Probability\nPopulation Lower Quantile"
util$plot_expectand_pushforward(pop_quant_samples,
                                25, flim=c(0.3, 0.8),
                                display_name=name,
                                col=util$c_light)
text(0.45, 5, "Observed\nCohort", col=util$c_light)

util$plot_expectand_pushforward(hyp_pop_quant_samples,
                                25, flim=c(0.3, 0.8),
                                border="#BBBBBB88",
                                add=TRUE)
text(0.675, 5, "Hypothetical\nCohort", col=util$c_dark)
```




```
quant_samples <- list("obs" = pop_quant_samples,
                      "hyp" = hyp_pop_quant_samples)
p_est <-
  util$implicit_subset_prob(quant_samples,
                            function(obs, hyp) hyp > obs)

format_string <- paste("Posterior probability that hypothetical",
                       "population lower quantile\nis greater than",
                       "observed population lower quantile",
                       "= %.3f +/- %.3f.")
cat(sprintf(format_string, p_est[1], 2 * p_est[2]))
```

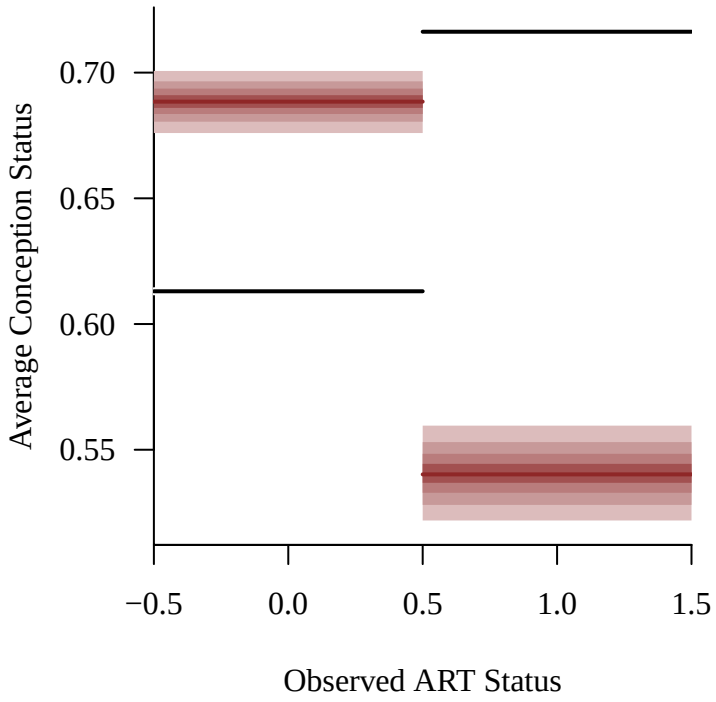
```
Posterior probability that hypothetical population lower quantile
is greater than observed population lower quantile = 0.570 +/- 0.015.
```

6 Model 4

At this point we have done some pretty cool Bayesian modeling and inference, but we have yet to consider ART. This is concerning given that ART can drastically increase fertility regardless of the direct and indirect cancer effects.

Unfortunately the posterior predictive conception behavior of our previous model very poorly matches the observed conception behavior once we take ART into account. Model critique is always fundamentally limited by the criteria we consider!

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

pred_names <- sapply(1:data$N, function(n) paste0('y_pred[', n, ']'))

util$plot_conditional_mean_quantiles(samples2, pred_names, data$k_art,
                                     -0.5, 1.5, 1, data$y,
                                     xlab="Observed ART Status",
                                     ylab="Average Conception Status")
```



Upon reflection this discrepancy isn't too surprising. The higher fertility of ART patients will bias the overall fertility of the observed cohort to larger values than what we would see from patients attempting natural conception alone.

6.1 Model 4a

In order to make faithful inferences for both the observed cohort and any hypothetical cohorts we need to account for ART by quantifying the conception probability of ART and non-ART patients separately.

6.1.1 ART Observational Model

Patients who have not undergone ART can be modeled as we did in the previous model, with the baseline probability of conceptual now more precisely interpreted as the baseline probability of *natural* conception. The challenge with modeling ART patients is that successful conceptions could be a result of either natural or ART methods but *we do not know which*. Consequently we have to take into account the potential success of both methods of conception at the same time.

ART conception and natural conception are likely to be coupled at some level. For example patients in stable, monogamous relations with female partners are unlikely to attempt one method after conception is achieved with the other. In theory we could attempt to model these competing events, but here we will make a simpler assumption that the two methods of conception are approximately independent so that an observed conception is given by a success with either method or both.

Given this assumption there are a few ways to calculate the overall conception probability for a given patient.

One way is to recognize that the probability of no conception is equal to the probability of both methods failing to conceive,

$$\begin{aligned}
p(\text{no conception}) &= p(\text{no conception}) \\
1 - p(\text{conception}) &= \ p(\text{no natural conception}) \\
&\quad \cdot p(\text{no ART conception}) \\
1 - p(\text{conception}) &= \ (1 - p(\text{natural conception})) \\
&\quad \cdot (1 - p(\text{ART conception})) \\
p(\text{conception}) &= \ p(\text{ART conception}) \, (1 - p(\text{natural conception})) \\
&\quad + p(\text{natural conception}).
\end{aligned}$$

Another way is to decompose the probability of conception into conditional probabilities,

$$\begin{aligned}
p(\text{conception}) = &\quad p(\text{conception} \mid \text{No natural conception}) \\
&\quad \cdot p(\text{No natural conception}) \\
&+ \ p(\text{conception} \mid \text{natural conception}) \\
&\quad \cdot p(\text{natural conception}) \\
p(\text{conception}) = &\quad p(\text{ART conception}) \\
&\quad \cdot (1 - p(\text{natural conception})) \\
&+ \ 1 \\
&\quad \cdot p(\text{natural conception}) \\
p(\text{conception}) = &\ p(\text{ART conception}) \, (1 - p(\text{natural conception})) \\
&+ p(\text{natural conception}).
\end{aligned}$$

The probability of natural conception $p(\text{natural conception})$ for an individual patient is given by the previous model,

$$q_{NC,n} = q_{NC_0} \exp(-\alpha_{\text{stg},n} - \alpha_{\text{rel},n} - \alpha_{\text{tox},n}).$$

In general $p(\text{ART conception})$ can also vary across the various patient categories, but here will assume that it is at least approximately homogeneous so that it can be modeled with a single parameter q_{AC}.

6.1.2 ART Prior Model

In our previous elicitation of reasonable behaviors for the baseline conception probability we did not explicitly consider ART. Consequently our previous prior model for q_{C_0} is now somewhat ambiguous.

If we were implicitly considering domain expertise for natural conception then we can apply the same prior model to q_{NC_0}. On the other hand if we were implicitly considering domain expertise for any method of conception then we would need to update the prior model to incorporate any information we have about natural conception in particular. Here we will assume the former and use the same prior model for q_{NC_0} as we did for q_{C_0}.

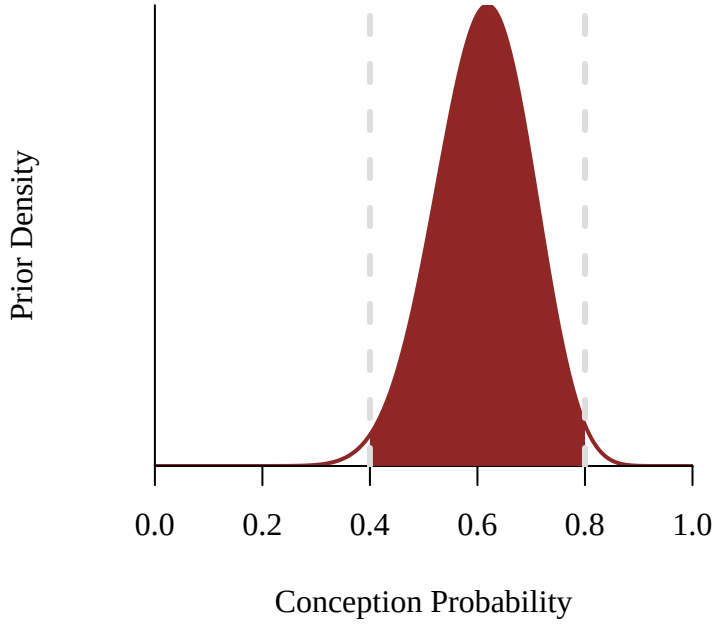
This leaves a prior model for q_{art}. Here let's say that our clinical domain expertise disfavors ART conception probabilities below 0.4 and above 0.8.

```
q_low <- 0.4
q_high <- 0.8

stan(file='stan_programs/prior_tune_beta.stan',
     data=list('q_low' = q_low, 'q_high' = q_high),
     iter=1, warmup=0, chains=1,
     seed=4838282, algorithm="Fixed_param")
```

```
alpha = 18.2301
beta = 11.6206

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1: Iteration: 1 / 1 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0 seconds (Warm-up)
Chain 1:                0 seconds (Sampling)
Chain 1:                0 seconds (Total)
Chain 1:
```

```
Inference for Stan model: anon_model.
1 chains, each with iter=1; warmup=0; thin=1;
post-warmup draws per chain=1, total post-warmup draws=1.

       mean se_mean sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
alpha 18.23      NA NA 18.23 18.23 18.23 18.23 18.23     0  NaN
beta  11.62      NA NA 11.62 11.62 11.62 11.62 11.62     0  NaN
lp__   0.00      NA NA  0.00  0.00  0.00  0.00  0.00     0  NaN

Samples were drawn using (diag_e) at Thu May 29 15:10:51 2025.
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
par(mfrow=c(1, 1), mar=c(5, 5, 3, 1))

qs <- seq(0, 1, 0.001)
dens <- dbeta(qs, 18.2, 11.6)
plot(qs, dens, type="l", col=util$c_dark, lwd=2,
     xlab="Conception Probability",
     ylab="Prior Density", yaxt='n')

q98 <- seq(q_low, q_high, 0.001)
dens <- dbeta(q98, 18.2, 11.6)
q98 <- c(q98, q_high, q_low)
dens <- c(dens, 0, 0)

polygon(q98, dens, col=util$c_dark, border=NA)

abline(v=q_low,  lwd=3, lty=2, col='#DDDDDD')
abline(v=q_high, lwd=3, lty=2, col='#DDDDDD')
```



6.1.3 Posterior Quantification

Incorporating ART results in an expanded model that exhausts all of the features of the observed data (Figure 8).

In order to apply all of the retrodictive checks that have previously considered we will need to simulate posterior predictive patient characteristics but then simulate posterior predictive conception statuses from the *observed* patient characteristics and not the newly simulated ones.

```
fit <- stan(file="stan_programs/model4a.stan",
            data=data, seed=8438338,
            warmup=1000, iter=2024, refresh=0)
```

The posterior computation continues to prove robust.

```
diagnostics <- util$extract_hmc_diagnostics(fit)
util$check_all_hmc_diagnostics(diagnostics)
```

```
  All Hamiltonian Monte Carlo diagnostics are consistent with reliable
Markov chain Monte Carlo.
```

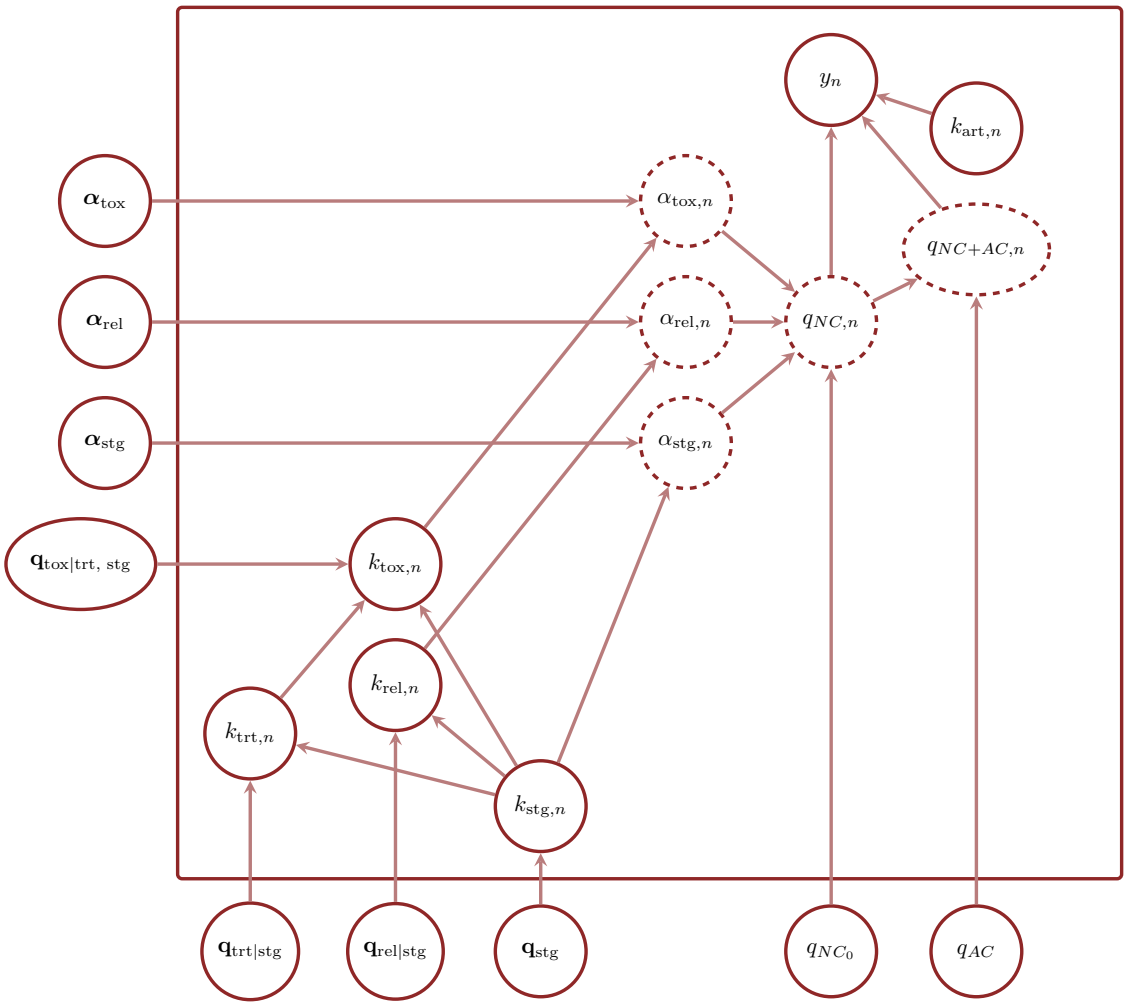


Figure 8: Our final model takes into account the influence of assisted reproduction technologies on patient conception.

```
samples4a <- util$extract_expectand_vals(fit)
base_samples <- util$filter_expectands(samples4a,
                                       c('q_stg', 'q_trt_active_stg',
                                         'q_tox_active_trt','q_NC_0',
                                         'alpha_rel', 'alpha_stg',
                                         'alpha_tox', 'q_AC'),
                                       check_arrays=TRUE)
util$check_all_expectand_diagnostics(base_samples)
```

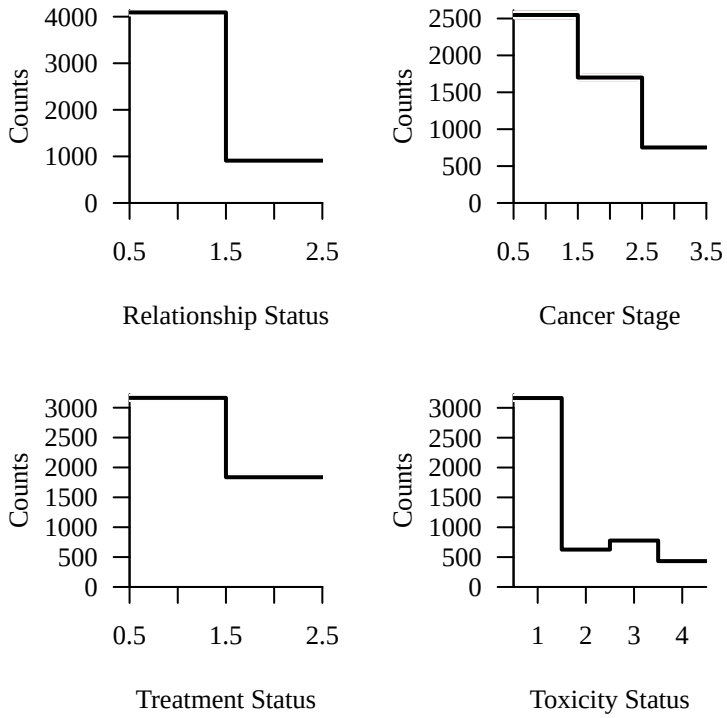
```
All expectands checked appear to be behaving well enough for reliable
Markov chain Monte Carlo estimation.
```

6.1.4 Retrodictive Checks

With all of the checks we have introduced at this point we have to be careful to go through the retrodictive behavior of each summary one by one.

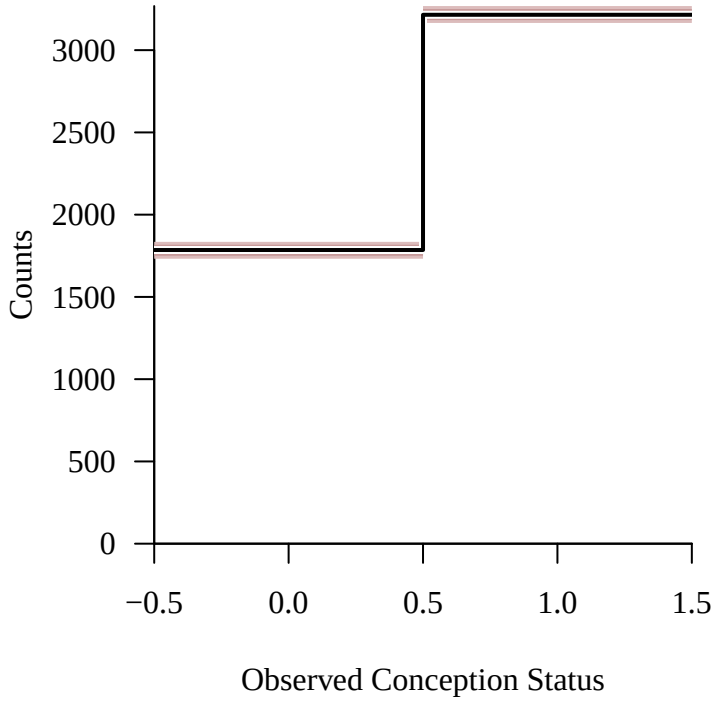
First we'll see how well the model recovers the patient characteristic behavior. Fortunately there are no signs of problems.

```
par(mfrow=c(2, 2), mar=c(5, 5, 1, 1))

util$plot_hist_quantiles(samples4a, 'k_rel_pred',
                         0.5, data$K_rel + 0.5, 1,
                         baseline_values=data$k_rel,
                         xlab="Relationship Status")

util$plot_hist_quantiles(samples4a, 'k_stg_pred',
                         0.5, data$K_stg + 0.5, 1,
                         baseline_values=data$k_stg,
                         xlab="Cancer Stage")

util$plot_hist_quantiles(samples4a, 'k_trt_pred',
                         0.5, data$K_trt + 0.5, 1,
                         baseline_values=data$k_trt,
                         xlab="Treatment Status")

util$plot_hist_quantiles(samples4a, 'k_tox_pred',
                         0.5, data$K_tox + 0.5, 1,
                         baseline_values=data$k_tox,
                         xlab="Toxicity Status")
```



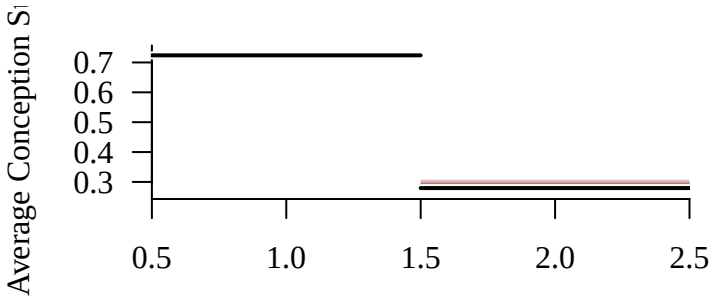
Next we have the posterior retrodictive check sensitive to the aggregate conception behavior. Again all looks good.

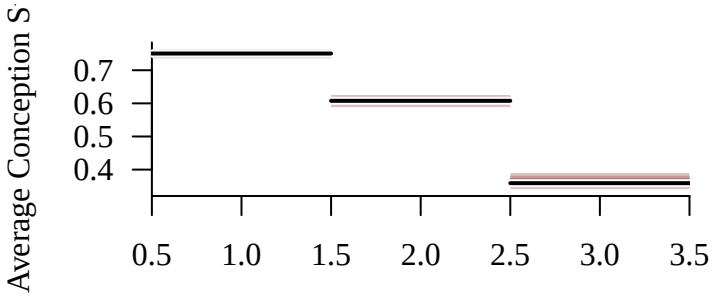
```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

util$plot_hist_quantiles(samples4a, 'y_pred', -0.5, 1.5, 1,
                         baseline_values=data$y,
                         xlab="Observed Conception Status")
```



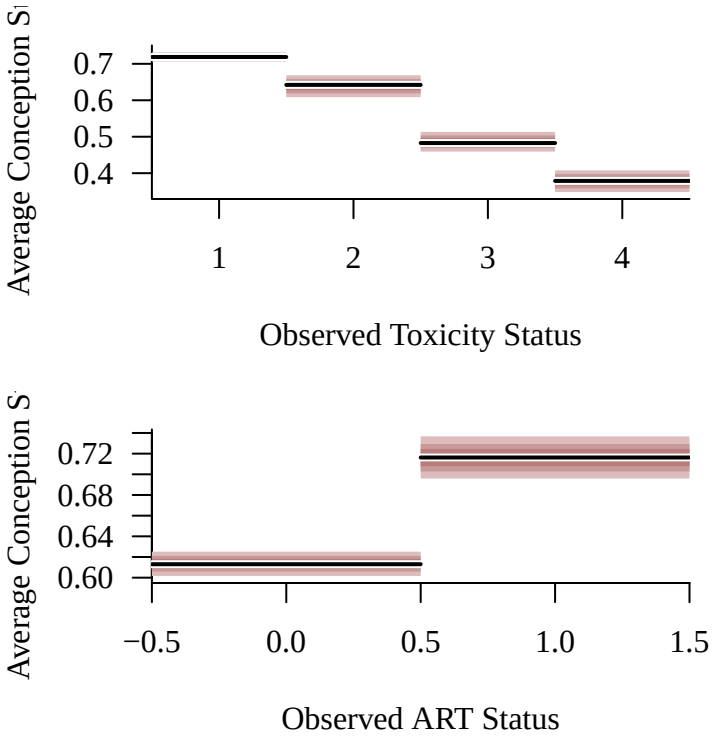
Finally we can compare the observed and posterior predictive conception behavior stratified by all of the clinical and demographic categories, including the use of ART. Looks like this expanded model has resolved the retrodictive tension of the previous model.

```
par(mfrow=c(2, 1), mar=c(5, 5, 1, 1))

pred_names <- sapply(1:data$N, function(n) paste0('y_pred[', n, ']'))

util$plot_conditional_mean_quantiles(samples4a, pred_names, data$k_rel,
                                     0.5, data$K_rel + 0.5, 1, data$y,
                                     xlab="Observed Relationship Status",
                                     ylab="Average Conception Status")

util$plot_conditional_mean_quantiles(samples4a, pred_names, data$k_stg,
                                     0.5, data$K_stg + 0.5, 1, data$y,
                                     xlab="Observed Cancer Stage",
                                     ylab="Average Conception Status")
```





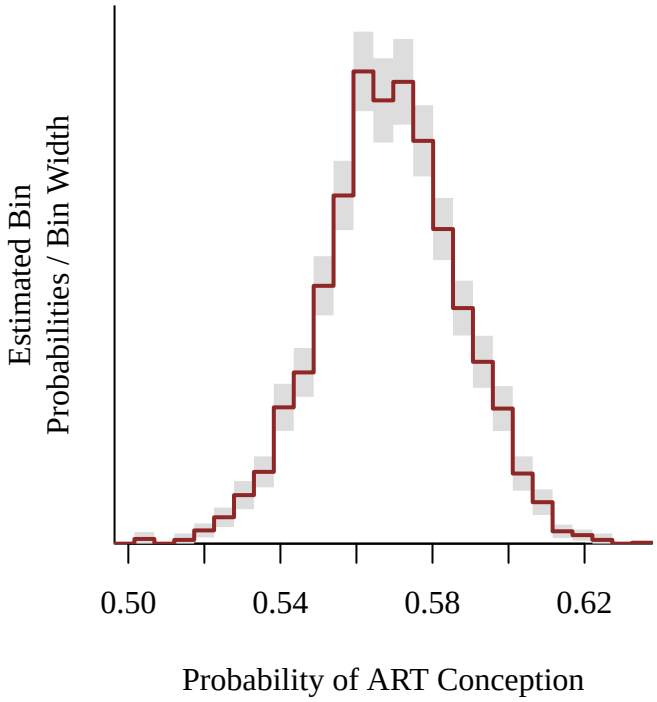

```
par(mfrow=c(2, 1), mar=c(5, 5, 1, 1))

util$plot_conditional_mean_quantiles(samples4a, pred_names, data$k_tox,
                                     0.5, data$K_tox + 0.5, 1, data$y,
                                     xlab="Observed Toxicity Status",
                                     ylab="Average Conception Status")

util$plot_conditional_mean_quantiles(samples4a, pred_names, data$k_art,
                                     -0.5, 1.5, 1, data$y,
                                     xlab="Observed ART Status",
                                     ylab="Average Conception Status")
```



6.1.5 Posterior Insights

We can analyze the posterior inferences from this new model directly.

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

name <- "Probability of ART Conception"
util$plot_expectand_pushforward(samples4a[['q_AC']], 25,
                                display_name=name)
```

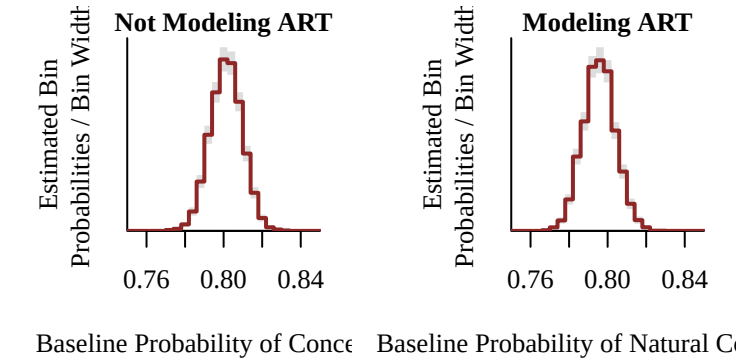


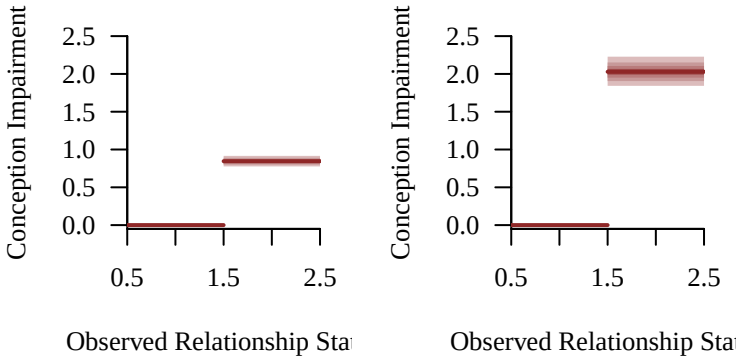
That said we can better understand the impact of explicitly modeling ART by comparing the posterior inferences from this new model to those of our last model that did not consider ART.

Inferences for the baseline conception probability are similar, but the inferences for the fertility impairment parameters change substantially. The previous model had to contort itself to fit the observed data as well as possible, resulting in misleading inferences and predictions that will not generalize well to other cohorts.

```
par(mfrow=c(2, 2), mar=c(5, 5, 1, 1))

name <- "Baseline Probability of Conception"
util$plot_expectand_pushforward(samples3a[['q_C_0']],
                                25, flim=c(0.75, 0.85),
                                display_name=name,
                                main="Not Modeling ART")

name <- "Baseline Probability of Natural Conception"
util$plot_expectand_pushforward(samples4a[['q_NC_0']],
                                25, flim=c(0.75, 0.85),
                                display_name=name,
                                main="Modeling ART")
```

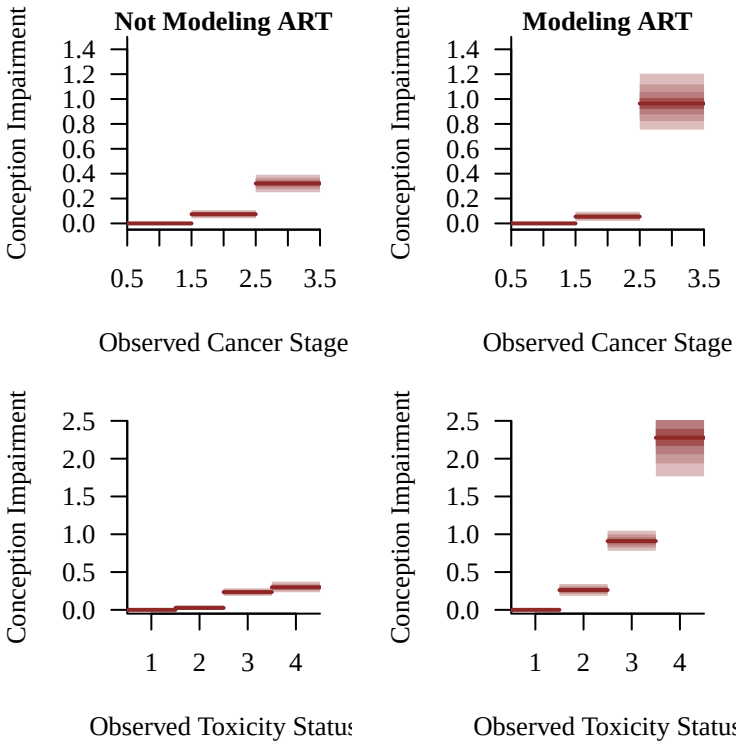
```
names <- sapply(1:data$K_rel,
                function(k) paste0('alpha_rel_buff[', k, ']'))
util$plot_disc_pushforward_quantiles(samples3a, names,
                                     xlab="Observed Relationship Status",
                                     ylab="Conception Impairment",
                                     display_ylim=c(-0.05, 2.5))

names <- sapply(1:data$K_rel,
                function(k) paste0('alpha_rel_buff[', k, ']'))
util$plot_disc_pushforward_quantiles(samples4a, names,
                                     xlab="Observed Relationship Status",
                                     ylab="Conception Impairment",
                                     display_ylim=c(-0.05, 2.5))
```






```
par(mfrow=c(2, 2), mar=c(5, 5, 1, 1))
names <- sapply(1:data$K_stg,
                function(k) paste0('alpha_stg_buff[', k, ']'))
util$plot_disc_pushforward_quantiles(samples3a, names,
                                     xlab="Observed Cancer Stage",
                                     ylab="Conception Impairment",
```

```
                                    display_ylim=c(-0.05, 1.5),
                                    main="Not Modeling ART")

names <- sapply(1:data$K_stg,
                function(k) paste0('alpha_stg_buff[', k, ']'))
util$plot_disc_pushforward_quantiles(samples4a, names,
                                    xlab="Observed Cancer Stage",
                                    ylab="Conception Impairment",
                                    display_ylim=c(-0.05, 1.5),
                                    main="Modeling ART")

names <- sapply(1:data$K_tox,
                function(k) paste0('alpha_tox_buff[', k, ']'))
util$plot_disc_pushforward_quantiles(samples3a, names,
                                    xlab="Observed Toxicity Status",
                                    ylab="Conception Impairment",
                                    display_ylim=c(-0.05, 2.5))

names <- sapply(1:data$K_tox,
                function(k) paste0('alpha_tox_buff[', k, ']'))
util$plot_disc_pushforward_quantiles(samples4a, names,
                                    xlab="Observed Toxicity Status",
                                    ylab="Conception Impairment",
                                    display_ylim=c(-0.05, 2.5))
```



6.2 Model 4b

With a model that can distinguish between natural and ART conceptions we can make much more accurate inferences for the hypothetical treatment that we considered above. Here we'll assume that the main epidemiology focus is the difference in only natural conceptions. In this case the ART conceptions in the observed data are something of a contamination that we need to model in order to isolate the desired behavior.

```
fit <- stan(file="stan_programs/model4b.stan",
            data=data, seed=8438338,
            warmup=1000, iter=2024, refresh=0)
```

One last check of the computational diagnostics; one more breath of appreciation for the robustness and scalability of Hamiltonian Monte Carlo.

```
diagnostics <- util$extract_hmc_diagnostics(fit)
util$check_all_hmc_diagnostics(diagnostics)
```

```
  All Hamiltonian Monte Carlo diagnostics are consistent with reliable
Markov chain Monte Carlo.
```

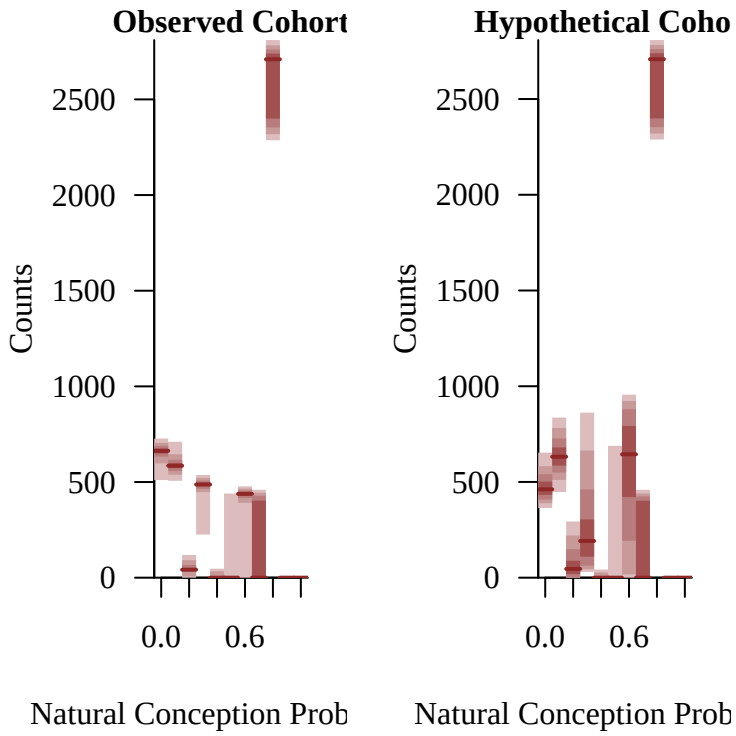
```
samples4b <- util$extract_expectand_vals(fit)
base_samples <- util$filter_expectands(samples4b,
                                       c('q_stg', 'q_trt_active_stg',
                                         'q_tox_active_trt','q_NC_0',
                                         'alpha_rel', 'alpha_stg',
                                         'alpha_tox', 'q_AC'),
                                       check_arrays=TRUE)
util$check_all_expectand_diagnostics(base_samples)
```

```
All expectands checked appear to be behaving well enough for reliable
Markov chain Monte Carlo estimation.
```

What do our more accurate inferences now have to say about the behavior of the hypothetical cohort?

The histogram of conception probabilities exhibits similar behavior to before, with the baseline peak mostly the same across the two cohorts but the remaining bulk shifting slightly towards larger conception probabilities in the hypothetical treatment.

```
par(mfrow=c(1, 2), mar=c(5, 5, 1, 1))

util$plot_hist_quantiles(samples4b, 'q_pred', -0.05, 1.05, 0.1,
                         xlab="Natural Conception Probability",
                         main="Observed Cohort")

util$plot_hist_quantiles(samples4b, 'q_hyp_pred', -0.05, 1.05, 0.1,
                         xlab="Natural Conception Probability",
                         main="Hypothetical Cohort")
```

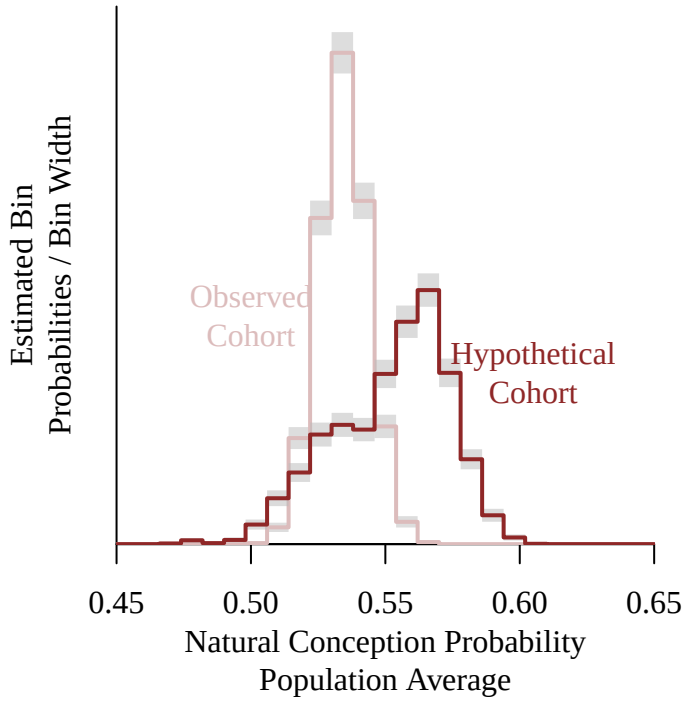


With the ART conceptions no longer biasing our inferences the hypothetical treatment appears to have similar benefits for both the ensemble average conception probability and the ensemble lower quantile conception probability.

```
var_repl <- list('p'=util$name_array('q_pred', c(data$N_pred)))
pop_ave_samples <-
  util$eval_expectand_pushforward(samples4b,
                                  function(p) mean(p),
                                  var_repl)

var_repl <- list('p'=util$name_array('q_hyp_pred', c(data$N_pred)))
hyp_pop_ave_samples <-
  util$eval_expectand_pushforward(samples4b,
                                  function(p) mean(p),
                                  var_repl)
```

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

name <- "Natural Conception Probability\nPopulation Average"
util$plot_expectand_pushforward(pop_ave_samples,
                                25, flim=c(0.45, 0.65),
```

```
                                  display_name=name,
                                  col=util$c_light)
text(0.500, 20, "Observed\nCohort", col=util$c_light)

util$plot_expectand_pushforward(hyp_pop_ave_samples,
                                  25, flim=c(0.45, 0.65),
                                  border="#BBBBBB88",
                                  add=TRUE)
text(0.605, 15, "Hypothetical\nCohort", col=util$c_dark)
```



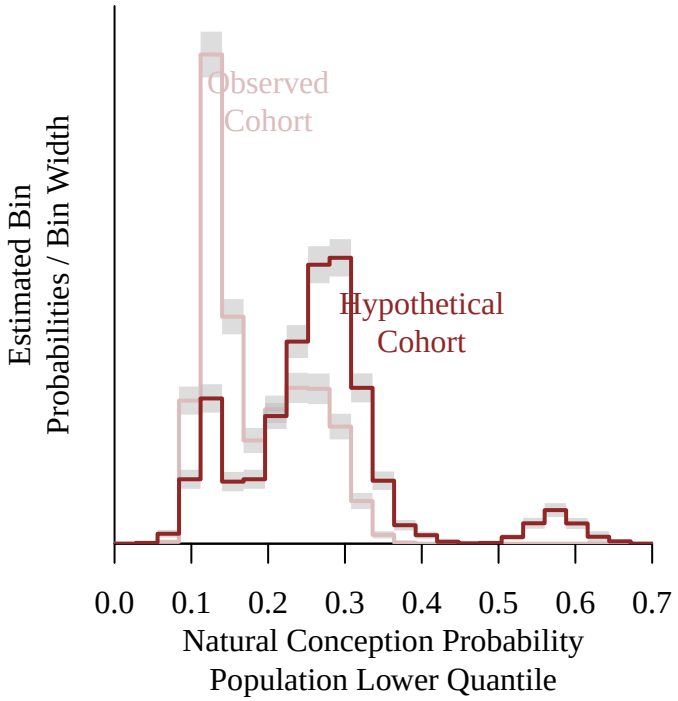

```
ave_samples <- list("obs" = pop_ave_samples,
                    "hyp" = hyp_pop_ave_samples)
p_est <-
  util$implicit_subset_prob(ave_samples,
                            function(obs, hyp) hyp > obs)

format_string <- paste("Posterior probability that hypothetical",
                       "population average\nis greater than observed",
                       "population average = %.3f +/- %.3f.")
cat(sprintf(format_string, p_est[1], 2 * p_est[2]))
```

```
Posterior probability that hypothetical population average
```

```
is greater than observed population average = 0.779 +/- 0.013.
```

```
var_repl <- list('p'=util$name_array('q_pred', c(data$N_pred)))
pop_quant_samples <-
  util$eval_expectand_pushforward(samples4b,
                                  function(p) quantile(p, prob=c(0.25)),
                                  var_repl)

var_repl <- list('p'=util$name_array('q_hyp_pred', c(data$N_pred)))
hyp_pop_quant_samples <-
  util$eval_expectand_pushforward(samples4b,
                                  function(p) quantile(p, prob=c(0.25)),
                                  var_repl)
```

```
par(mfrow=c(1, 1), mar=c(5, 5, 1, 1))

name <- "Natural Conception Probability\nPopulation Lower Quantile"
util$plot_expectand_pushforward(pop_quant_samples,
                                25, flim=c(0.0, 0.7),
                                display_name=name,
                                col=util$c_light)
text(0.2, 10, "Observed\nCohort", col=util$c_light)

util$plot_expectand_pushforward(hyp_pop_quant_samples,
                                25, flim=c(0.0, 0.7),
                                border="#BBBBBB88",
                                add=TRUE)
text(0.4, 5, "Hypothetical\nCohort", col=util$c_dark)
```




```
quant_samples <- list("obs" = pop_quant_samples,
                      "hyp" = hyp_pop_quant_samples)
p_est <-
  util$implicit_subset_prob(quant_samples,
                            function(obs, hyp) hyp > obs)

format_string <- paste("Posterior probability that hypothetical",
                       "population lower quantile\nis greater than",
                       "observed population lower quantile",
                       "= %.3f +/- %.3f.")
cat(sprintf(format_string, p_est[1], 2 * p_est[2]))
```

```
Posterior probability that hypothetical population lower quantile
is greater than observed population lower quantile = 0.680 +/- 0.015.
```

7 Conclusion

Acknowledging an underlying data generating process, and using any available domain expertise to model it, is a powerful way to guide statistical analyses. In this case study we were

able to model not only the variation in natural conception probability across patient characteristics but also the patient characteristics themselves and the potential contamination from alternative conception methods.

This analyses has only scratched the surface of what Bayesian modeling techniques can accomplish. For example the patient characteristic model we built can be used as a basis for inferring partially missing clinical and demographic information, a process also known as *imputation*. This can be especially useful when dealing with complications like patient dropout across the observation interval.

At the same time we can model not only the conception behavior within the ART and non-ART groups but also the probability of any particular patient using ART. In general this probability might be coupled with the other patient characteristic behaviors and modeling that coupling would allow us to construct even more nuanced hypothetical cohorts, and more precise generalized predictions.

Acknowledgements

I thank Simon Steiger, Peter Alping, Joshua Entrop, Paulina Sell, and jd for helpful comments.

License

A repository containing all of the files used to generate this chapter is available on GitHub.

Original Computing Environment

```
writeLines(readLines(file.path(Sys.getenv("HOME"), ".R/Makevars")))
```

```
CC=clang

CXXFLAGS=-O3 -mtune=native -march=native -Wno-unused-variable -Wno-unused-function -Wno-macro
CXX=clang++ -arch x86_64 -ftemplate-depth-256

CXX14FLAGS=-O3 -mtune=native -march=native -Wno-unused-variable -Wno-unused-function -Wno-mac
CXX14=clang++ -arch x86_64 -ftemplate-depth-256
```

```
sessionInfo()
```

```
R version 4.3.2 (2023-10-31)
Platform: x86_64-apple-darwin20 (64-bit)
Running under: macOS Sonoma 14.7.5

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-x86_64/Resources/lib/libRlapack.dylib;

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: America/New_York
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] colormap_0.1.4     rstan_2.32.6       StanHeaders_2.32.7

loaded via a namespace (and not attached):
 [1] gtable_0.3.4       jsonlite_1.8.8     compiler_4.3.2     Rcpp_1.0.11
 [5] stringr_1.5.1      parallel_4.3.2     gridExtra_2.3      scales_1.3.0
 [9] yaml_2.3.8         fastmap_1.1.1      ggplot2_3.4.4      R6_2.5.1
[13] curl_5.2.0         knitr_1.45         tibble_3.2.1       munsell_0.5.0
[17] pillar_1.9.0       rlang_1.1.2        utf8_1.2.4         V8_4.4.1
[21] stringi_1.8.3      inline_0.3.19      xfun_0.41          RcppParallel_5.1.7
[25] cli_3.6.2          magrittr_2.0.3     digest_0.6.33      grid_4.3.2
[29] lifecycle_1.0.4    vctrs_0.6.5        evaluate_0.23      glue_1.6.2
[33] QuickJSR_1.0.8     codetools_0.2-19   stats4_4.3.2       pkgbuild_1.4.3
[37] fansi_1.0.6        colorspace_2.1-0   rmarkdown_2.25     matrixStats_1.2.0
[41] tools_4.3.2        loo_2.6.0          pkgconfig_2.0.3    htmltools_0.5.7
```

Stan
Program 1 `prior\_tune\_beta.stan`

```
functions {
  // Differences between beta tail probabilities
  // and target probabilities
  vector tail_delta(vector y, vector theta,
                    array[] real x_r, array[] int x_i) {
    vector[2] deltas;
    deltas[1] =     beta_cdf(theta[1] | exp(y[1]), exp(y[2])) - 0.01;
    deltas[2] = 1 - beta_cdf(theta[2] | exp(y[1]), exp(y[2])) - 0.01;
    return deltas;
  }
}

data {
  real<lower=0, upper=1>     q_low;  // Lower threshold
  real<lower=q_low, upper=1> q_high; // Upper threshold
}

transformed data {
  vector[2] y_guess = [log(5), log(5)]'; // Initial guess at beta parameters
  vector[2] theta = [q_low, q_high]';    // Target quantiles
  vector[2] y;
  array[0] real x_r;
  array[0] int x_i;

  // Find beta parameters that ensure
  // 1% probability below lower threshold
  // and 1% probability above upper threshold
  y = algebra_solver(tail_delta, y_guess, theta, x_r, x_i);

  print("alpha = ", exp(y[1]));
  print("beta = ", exp(y[2]));
}

generated quantities {
  real alpha = exp(y[1]);
  real beta = exp(y[2]);
}
```

Stan
Program 2 `model1.stan`

```
data {
  // Number of observations
  int<lower=1> N;

  // Observed conception status
  // y = 0: No conception
  // y = 1: Conception
  array[N] int<lower=0, upper=1> y;
}

parameters {
  real<lower=0, upper=1> q_C; // Conception probability
}

model {
  // Prior model
  target += beta_lpdf(q_C | 2.5, 2.0); // 0.10 <~ q_C <~ 0.95

  // Observational model
  target += bernoulli_lpmf(y | q_C);

  // Also valid but slightly less efficient
  // for (n in 1:N) {
  //   target += bernoulli_lpmf(y[n] | q_C);
  // }
}

generated quantities {
  // Posterior predictive data
  array[N] int<lower=0, upper=1> y_pred;

  for (n in 1:N) {
    y_pred[n] = bernoulli_rng(q_C);
  }
}
```

Stan
Program 3 `model2.stan`

```
data {
  // Number of observations
  int<lower=1> N;

  // Relationship status
  // k = 1: Stable partner
  // k = 2: No partner
  int<lower=1> K_rel;

  // Cancer stage
  // k = 1: No cancer
  // k = 2: Early stage cancer
  // k = 3: Advanced stage cancer
  int<lower=1> K_stg;

  // Toxicity status
  // k = 1: None
  // k = 2: Low
  // k = 3: Medium
  // k = 4: High
  int<lower=1> K_tox;

  // Observed conception status
  // y = 0: No conception
  // y = 1: Conception
  array[N] int<lower=0, upper=1> y;

  // Observed relationship status;
  array[N] int<lower=1, upper=K_rel> k_rel;

  // Observed cancer stage;
  array[N] int<lower=1, upper=K_stg> k_stg;

  // Observed toxicity status;
  array[N] int<lower=1, upper=K_tox> k_tox;
}

parameters {
  // Probability of conception for baseline patients in a stable
  // relationship, no cancer, and no toxicity
  real<lower=0, upper=1> q_C_0;

  // Proportional decreases in conception probability due to
  // non-baseline relationship status, cancer stage, and toxicity
  // status.
  positive_ordered[K_rel - 1] alpha_rel;
  positive_ordered[K_stg - 1] alpha_stg;
  positive_ordered[K_tox - 1] alpha_tox;
}
```

Stan
Program 4 `model3a.stan`

```
data {
  // Number of observations
  int<lower=1> N;

  // Number of predictions
  int<lower=1> N_pred;

  // Relationship status
  // k = 1: Stable partner
  // k = 2: No partner
  int<lower=1> K_rel;

  // Cancer stage
  // k = 1: No cancer
  // k = 2: Early stage cancer
  // k = 3: Advanced stage cancer
  int<lower=1> K_stg;

  // Treatment status
  // k = 1: No treatment
  // k = 2: Treatment
  int<lower=1> K_trt;

  // Toxicity status
  // k = 1: None
  // k = 2: Low
  // k = 3: Medium
  // k = 4: High
  int<lower=1> K_tox;

  // Observed conception status
  // y = 0: No conception
  // y = 1: Conception
  array[N] int<lower=0, upper=1> y;

  // Observed relationship status;
  array[N] int<lower=1, upper=K_rel> k_rel;

  // Observed cancer stage;
  array[N] int<lower=1, upper=K_stg> k_stg;

  // Observed treatment status;
  array[N] int<lower=1, upper=K_trt> k_trt;

  // Observed toxicity status;
  array[N] int<lower=1, upper=K_tox> k_tox;
}

parameters {
  // Marginal probability of cancer stage
```

Stan
Program 5 `model3b.stan`

```
data {
  // Number of observations
  int<lower=1> N;

  // Number of predictions
  int<lower=1> N_pred;

  // Relationship status
  // k = 1: Stable partner
  // k = 2: No partner
  int<lower=1> K_rel;

  // Cancer stage
  // k = 1: No cancer
  // k = 2: Early stage cancer
  // k = 3: Advanced stage cancer
  int<lower=1> K_stg;

  // Treatment status
  // k = 1: No treatment
  // k = 2: Treatment
  int<lower=1> K_trt;

  // Toxicity status
  // k = 1: None
  // k = 2: Low
  // k = 3: Medium
  // k = 4: High
  int<lower=1> K_tox;

  // Observed conception status
  // y = 0: No conception
  // y = 1: Conception
  array[N] int<lower=0, upper=1> y;

  // Observed relationship status;
  array[N] int<lower=1, upper=K_rel> k_rel;

  // Observed cancer stage;
  array[N] int<lower=1, upper=K_stg> k_stg;

  // Observed treatment status;
  array[N] int<lower=1, upper=K_trt> k_trt;

  // Observed toxicity status;
  array[N] int<lower=1, upper=K_tox> k_tox;

  // Hypothetical toxicity distribution configurations
  vector[K_tox] alpha_tox_hyp1;
  vector[K_tox] alpha_tox_hyp2;
```

Stan
Program 6 `model4a.stan`

```
data {
  // Number of observations
  int<lower=1> N;

  // Relationship status
  // k = 1: Stable partner
  // k = 2: No partner
  int<lower=1> K_rel;

  // Cancer stage
  // k = 1: No cancer
  // k = 2: Early stage cancer
  // k = 3: Advanced stage cancer
  int<lower=1> K_stg;

  // Treatment status
  // k = 1: No treatment
  // k = 2: Treatment
  int<lower=1> K_trt;

  // Toxicity status
  // k = 1: None
  // k = 2: Low
  // k = 3: Medium
  // k = 4: High
  int<lower=1> K_tox;

  // Observed conception status
  // y = 0: No conception
  // y = 1: Conception
  array[N] int<lower=0, upper=1> y;

  // Observed relationship status;
  array[N] int<lower=1, upper=K_rel> k_rel;

  // Observed cancer stage;
  array[N] int<lower=1, upper=K_stg> k_stg;

  // Observed treatment status;
  array[N] int<lower=1, upper=K_trt> k_trt;

  // Observed toxicity status;
  array[N] int<lower=1, upper=K_tox> k_tox;

  // Observed assistive reproductive technology (ART) status
  // k = 0: No ART
  // k = 1: ART
  array[N] int<lower=0, upper=1> k_art;
}
```

Stan
Program 7 `model4b.stan`

```
data {
  // Number of observations
  int<lower=1> N;

  // Number of predictions
  int<lower=1> N_pred;

  // Relationship status
  // k = 1: Stable partner
  // k = 2: No partner
  int<lower=1> K_rel;

  // Cancer stage
  // k = 1: No cancer
  // k = 2: Early stage cancer
  // k = 3: Advanced stage cancer
  int<lower=1> K_stg;

  // Treatment status
  // k = 1: No treatment
  // k = 2: Treatment
  int<lower=1> K_trt;

  // Toxicity status
  // k = 1: None
  // k = 2: Low
  // k = 3: Medium
  // k = 4: High
  int<lower=1> K_tox;

  // Observed conception status
  // y = 0: No conception
  // y = 1: Conception
  array[N] int<lower=0, upper=1> y;

  // Observed relationship status;
  array[N] int<lower=1, upper=K_rel> k_rel;

  // Observed cancer stage;
  array[N] int<lower=1, upper=K_stg> k_stg;

  // Observed treatment status;
  array[N] int<lower=1, upper=K_trt> k_trt;

  // Observed toxicity status;
  array[N] int<lower=1, upper=K_tox> k_tox;

  // Observed assistive reproductive technology (ART) status
  // k = 0: No ART
  // k = 1: ART
```