How to Use HPR: Basal Body Temperature Example

Overview of horseshoe process regression

Horseshoe process regression (HPR) is a method for fitting an association between a predictor and an outcome that exhibits abrupt changes in variance: associations that might look like step functions, or piecewise linear functions with sharp changes in concavity. To do so, HPR imposes a discrete approximation of a horseshoe process prior on the association. To do so, we use the model:

Let y_i be some outcome observed for patients $i = 1, ..., n$ at continuous predictor value x_i. We restrict y_i to be distributed as either Gaussian, Bernoulli, or Poisson. Define $\mathbf{x}, \mathbf{y}$ as the corresponding length n vectors of these observations. Let $\mathbf{z_i}$ be a length p vector of linear predictors of y_i for each subject, yielding an n x p matrix $\mathbf{Z}$ of covariates. Define $\mathbf{t}$ as a length m vector containing the unique, ordered values of $\mathbf{x}$. Suppose that $x_i = t_j$. Then we define our HPR model as:

$$
\begin{aligned}
g(E(y_i)) = f_j &= \alpha + \boldsymbol{\beta}\mathbf{z_i} + \sum_{\mathbf{k=1}}^{\mathbf{j}} \mathbf{h_k} \\
h_k &\sim N(0, \tau_h^2 \lambda_k^2 (t_k - t_{k-1})), \ k = 2, ..., m \\
h_1 &= 0 \\
\tau_h &\sim C^+(0, c) \\
\lambda_k &\overset{iid}{\sim} C^+(0, 1), k = 2, ..., m \\
\alpha &\sim N(a, b^2) \\
\boldsymbol{\beta} &\sim N(\mathbf{0}, \mathbf{d^2})
\end{aligned}
$$

This corresponds to placing a horseshoe prior on the first order differences of $\mathbf{f}$ and approximates placing a horseshoe prior on the first derivative of the association we wish to model. In general, the λ_k values will be small, resulting in long stretches of near-constant values of $\mathbf{f}$, punctuated by abrupt steps which may be quite large when large values of λ_k are supported by the data.

The process starts at a y-intercept α, which has a normal prior placed on it. The $g(E(y_i))$ formulation allows for non-Gaussian data through the use of an appropriate transformation g. We use a logit transformation and Bernoulli likelihood for binary outcomes and a log transformation and Poisson likelihood for count data. We allow for linear predictors $\mathbf{Z}$, which can be either continuous or categorical; categorical predictors would need to be converted to a dummy parameterization. The model also allows for multiple observations at the same t_j value and does not require the $\mathbf{t}$ values to be evenly spaced.

We also provide extensions to constrain the nonlinear association between $\boldsymbol{x}$ and $\boldsymbol{y}$ to be monotonic increasing, and to permit for sampling of $\boldsymbol{f}$ at unobserved values of $\boldsymbol{x}$ (e.g. interpolation, extrapolation). We also provide a function to perform posterior prediction.

All models are fit using Hamiltonian Monte Carlo (HMC) via Stan and the package `cmdstanr` through R, within this package, `HPR`. For more details on HPR, please see Chase, Taylor, and Boonstra (2022+).

Using the hpr package

Here, we will use `HPR` to fit women's basal body temperature (BBT) data abstracted from Toni Weschler's *Taking Charge of Your Fertility*. First, we load both the HPR package and the basal body temperature data.

```
library(HPR)
library(tidyverse)
#> -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
#> v ggplot2 3.3.5     v purrr   0.3.4
#> v tibble  3.1.6     v dplyr   1.0.7
#> v tidyr   1.1.3     v stringr 1.4.0
#> v readr   1.4.0     v forcats 0.5.1
#> -- Conflicts ------------------------------------------ tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()

data(bbt_data, package = "HPR")
head(bbt_data)
#> # A tibble: 6 x 5
#>   Name    Day Temperature Mucus Aberration
#>   <chr> <dbl>       <dbl> <chr>      <dbl>
#> 1 Ruby      1        97.5 <NA>          NA
#> 2 Ruby      2        97.3 <NA>          NA
#> 3 Ruby      3        97.6 <NA>          NA
#> 4 Ruby      4        97.1 <NA>          NA
#> 5 Ruby      5        97.3 <NA>          NA
#> 6 Ruby      6        97.5 <NA>          NA
```

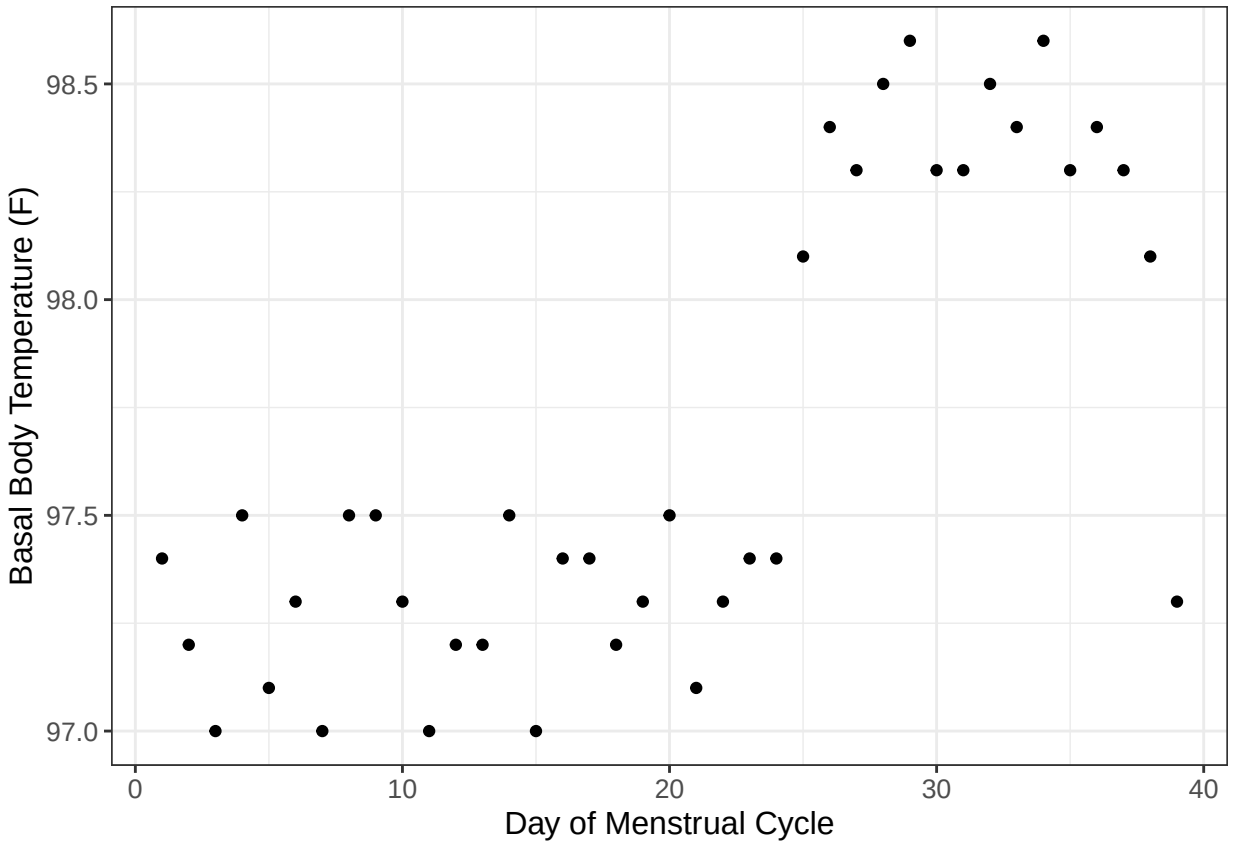
The BBT data has 5 variables:

- Name: the name of the woman (these are made-up names)
- Day: what day of the menstrual cycle her temperature was recorded
- Temperature: her basal body temperature reading, in Fahrenheit
- Mucus: if provided, her cervical mucus consistency, with values "n" for none, "l" for low quantities of sticky mucus, "m" for medium quantities of smooth mucus, and "h" for high quantities of egg-white-type mucus
- Aberration: whether there was some aberration with her temperature reading (heavy drinking, illness, temperature taken late in the day rather than first thing in the morning, etc.), with 0 or `NA` for no aberration and 1 for an aberration

Let's look at the data for a single woman ("Author").

```
author_dat <- filter(bbt_data, Name == "Author")

ggplot(data = author_dat) +
  geom_point(aes(x = Day, y = Temperature)) +
  theme_bw() +
  xlab("Day of Menstrual Cycle") +
  scale_y_continuous("Basal Body Temperature (F)") +
  theme(legend.position = "bottom", text=element_text(size=12))
```



We see that these data look like a good candidate for HPR, because we have a set of measurements (before day 24) that seems to be centered around 97.25 degrees, and a second set of measurements (day 25 and onwards) that appears centered around 98.25 degrees, with an abrupt change from one temperature to the other. This also matches our subject-knowledge of women's BBT trajectories: there's an abrupt increase in temperature immediately following ovulation, followed by a sudden drop in temperature with the onset of a menstrual period.

Let's try to fit a simple HPR for Author's data. Here, we use day as our nonlinear predictor and temperature as our outcome. Because temperature is continuous, we use a Gaussian likelihood. Note that the predictor has to be input as a numeric matrix `X`, with a column for each nonlinear predictor (we just have a single nonlinear predictor, so this is a one-column matrix). The outcome, `y` is a numeric vector.

```
X <- as.matrix(author_dat$Day, ncol = 1)
y <- author_dat$Temperature

author_hpr <- hpr(y = y, X = X, family = "gaussian")
#> Running MCMC with 4 sequential chains...
#> Chain 1 Informational Message: The current Metropolis proposal is about to be rejected because of th
#> Chain 1 Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in '/var/folders/5h/8ph
#> Chain 1 If this warning occurs sporadically, such as for highly constrained variable types like cova
#> Chain 1 but if this warning occurs often then your model may be either severely ill-conditioned or m
#> Chain 1
#> Chain 1 finished in 13.5 seconds.
#> Chain 2 finished in 11.7 seconds.
#> Chain 3 Informational Message: The current Metropolis proposal is about to be rejected because of th
#> Chain 3 Exception: normal_lpdf: Location parameter[2] is -inf, but must be finite! (in '/var/folders
#> Chain 3 If this warning occurs sporadically, such as for highly constrained variable types like cova
#> Chain 3 but if this warning occurs often then your model may be either severely ill-conditioned or m
```

```
#> Chain 3
#> Chain 3 finished in 12.4 seconds.
#> Chain 4 finished in 12.0 seconds.
#>
#> All 4 chains finished successfully.
#> Mean chain execution time: 12.4 seconds.
#> Total execution time: 50.1 seconds.
#>
#> Warning: 105 of 8000 (1.0%) transitions ended with a divergence.
#> This may indicate insufficient exploration of the posterior distribution.
#> Possible remedies include:
#>   * Increasing adapt_delta closer to 1 (default is 0.8)
#>   * Reparameterizing the model (e.g. using a non-centered parameterization)
#>   * Using informative or weakly informative prior distributions
```

We check model diagnostics to see if there's anything worrisome:

```
get_diagnostics(author_hpr)
#>   Divergences Max_Treedepth RHat Min_Ess_Bulk Min_Ess_Tail Num_Param
#> 1         105             0    0     2163.732     1022.083        39
#>   Num_Samples     Time
#> 1        8000 50.93947
```

We can see that we had some Stan divergences, but because they made up <5% of overall samples and the other diagnostics are clean, it is likely to safe to ignore them for our model (for a longer discussion of this issue, see Chase et al. (2022+)). The other diagnostics include:

- the number of `max_treedepth` warnings from the Stan sampler (ideally close to 0, as larger values suggest inefficient sampling)
- `Rhat`: the number of parameters that had $\hat{R}$ values greater than 1.1. $\hat{R}$ estimates mixing across the chains and the stability of posterior sampling; ideally we would like this to be 0
- the minimum effective sample sizes across all parameters in both the bulk of the posterior (`min_ess_bulk`) and the tails of the posterior (`min_ess_tail`)–we would like this to be large and ideally larger than ~300
- `num_param` gives the number of parameters, which here is the number of unique predictions generated by the model (itself a function of the local shrinkage parameters, global shrinkage parameter, normal increments, and y-intercept)
- `num_samples` gives the total number of MCMC samples
- `time` gives the computing time

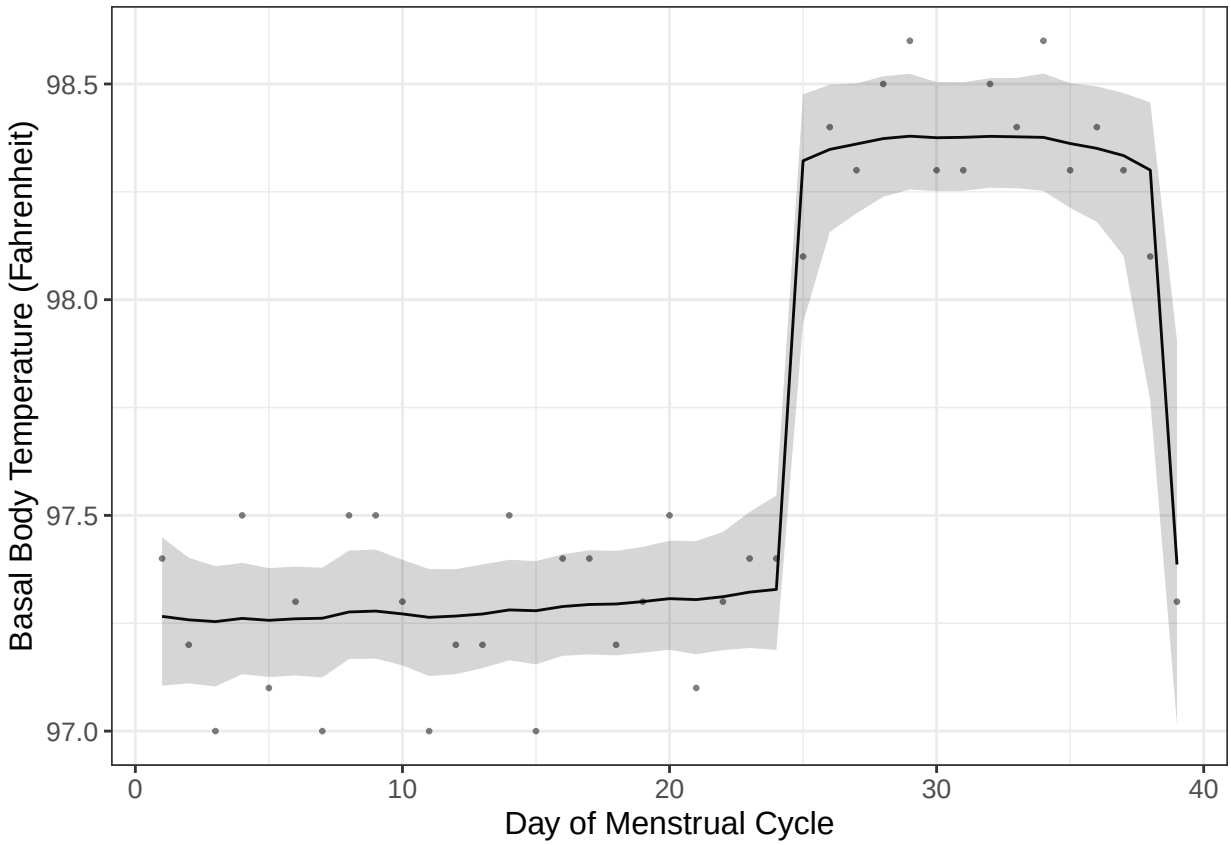
Now that we've reviewed our diagnostics, we can get estimates of $\boldsymbol{f}$ using the function `get_preds` and plot them using ggplot:

```
author_results <- get_preds(author_hpr, alpha = 0.05)
#> Warning: Dropping 'draws_df' class as required metadata was removed.
author_results$x <- author_dat$Day

author_plot <- ggplot(data = author_results) +
  geom_line(aes(x = x, y = Median)) +
  geom_ribbon(aes(x = x, ymin = Lower, ymax = Upper), alpha = 0.2) +
  theme_bw() +
  geom_point(data = author_dat, aes(x = Day, y = Temperature), alpha = 0.5,
             size = 0.5) +
```

```
  xlab("Day of Menstrual Cycle") +
  scale_y_continuous("Basal Body Temperature (Fahrenheit)") +
  theme(legend.position = "bottom", text=element_text(size=12))

author_plot
```



We see that the fit we get exhibits HPR's classic step-function type behavior, and it appears that the time of ovulation (which occurs immediately before the temperature increase) likely occurred on day 24 of the menstrual cycle.

If we omit the final datapoint of Author's cycle (the measurement on day 39) which was taken on the day that the next period started and indicates the start of the next menstrual cycle, we might wish to constrain this function fit to be monotonic increasing, because we know that temperature will only increase during the menstrual cycle. To do this, we tell `hpr` the column index of the nonlinear predictor that we want to constrain (since we only have a single nonlinear predictor, this is easy: 1) and how we want to constrain monotonicity. In keeping with the recommendations from Chase et al. (2022+), we will use the absolute value function to impose monotonicity, using option `"abs"`. (We could also use option `"exp"` to impose monotonicity, which uses exponentiation, but this is not recommended for computational reasons).

```
author_dat_nolast <- filter(author_dat, Day != 39)
X_nolast <- as.matrix(author_dat_nolast$Day, ncol = 1)
y_nolast <- author_dat_nolast$Temperature

author_hpr_monoton <- hpr(y = y_nolast, X = X_nolast, family = "gaussian",
                          monotonic_terms = c(1), monotonic_approach = "abs")
#> Running MCMC with 4 sequential chains...
#>
#> Chain 1 finished in 11.9 seconds.
```

```
#> Chain 2 Informational Message: The current Metropolis proposal is about to be rejected because of th
#> Chain 2 Exception: normal_lpdf: Scale parameter is nan, but must be positive! (in '/var/folders/5h/8
#> Chain 2 If this warning occurs sporadically, such as for highly constrained variable types like cova
#> Chain 2 but if this warning occurs often then your model may be either severely ill-conditioned or m
#> Chain 2
#> Chain 2 finished in 14.4 seconds.
#> Chain 3 Informational Message: The current Metropolis proposal is about to be rejected because of th
#> Chain 3 Exception: normal_lpdf: Location parameter[2] is inf, but must be finite! (in '/var/folders/
#> Chain 3 If this warning occurs sporadically, such as for highly constrained variable types like cova
#> Chain 3 but if this warning occurs often then your model may be either severely ill-conditioned or m
#> Chain 3
#> Chain 3 finished in 11.2 seconds.
#> Chain 4 finished in 10.9 seconds.
#>
#> All 4 chains finished successfully.
#> Mean chain execution time: 12.1 seconds.
#> Total execution time: 48.9 seconds.
#>
#> Warning: 124 of 8000 (2.0%) transitions ended with a divergence.
#> This may indicate insufficient exploration of the posterior distribution.
#> Possible remedies include:
#>   * Increasing adapt_delta closer to 1 (default is 0.8)
#>   * Reparameterizing the model (e.g. using a non-centered parameterization)
#>   * Using informative or weakly informative prior distributions
```

Diagnostics:

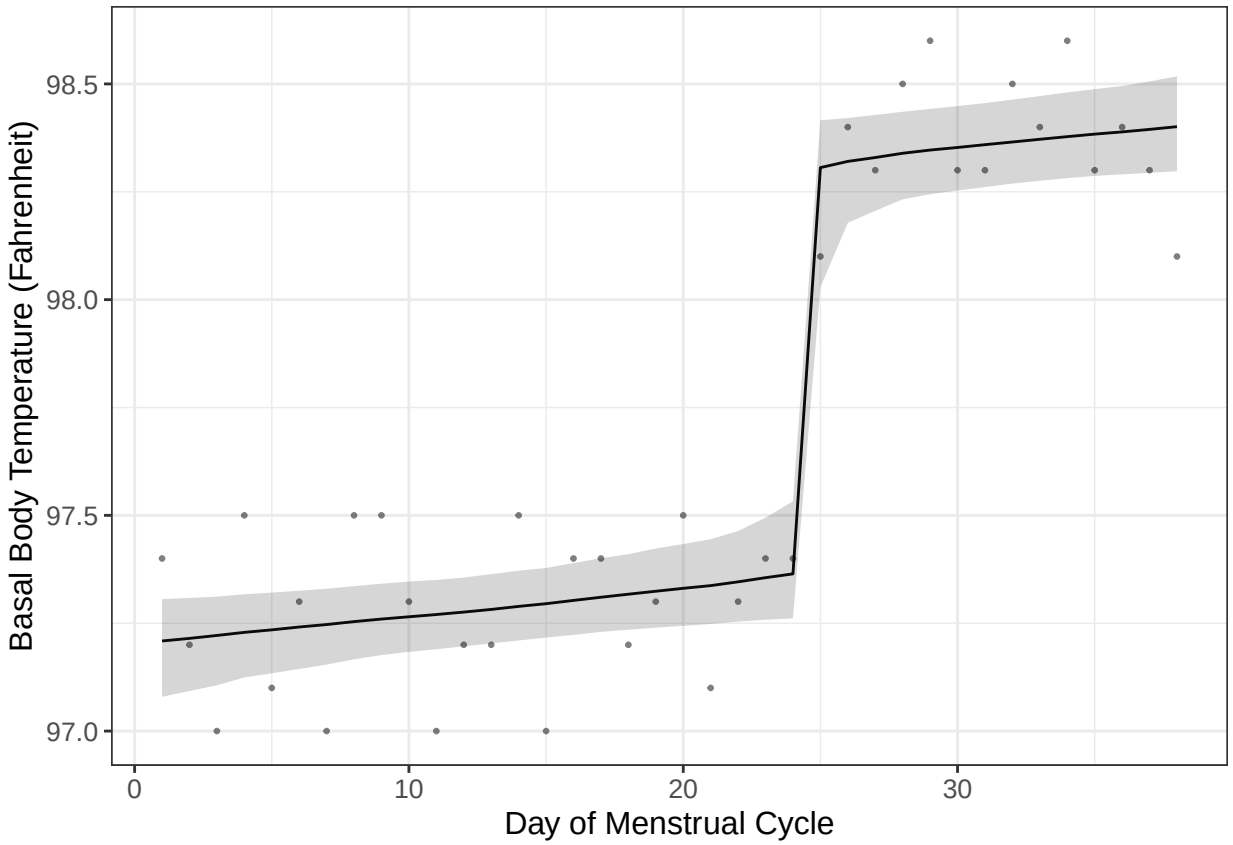
```
get_diagnostics(author_hpr_monoton)
#>   Divergences Max_Treedepth RHat Min_Ess_Bulk Min_Ess_Tail Num_Param
#> 1         124             0    0     4213.857     3094.598        38
#>   Num_Samples     Time
#> 1        8000 49.49777
```

And the estimated trajectory:

```
author_results_mono <- get_preds(author_hpr_monoton, alpha = 0.05)
#> Warning: Dropping 'draws_df' class as required metadata was removed.
author_results_mono$x <- author_dat_nolast$Day

author_plot_mono <- ggplot(data = author_results_mono) +
  geom_line(aes(x = x, y = Median)) +
  geom_ribbon(aes(x = x, ymin = Lower, ymax = Upper), alpha = 0.2) +
  theme_bw() +
  geom_point(data = author_dat_nolast, aes(x = Day, y = Temperature), alpha = 0.5,
             size = 0.5) +
  xlab("Day of Menstrual Cycle") +
  scale_y_continuous("Basal Body Temperature (Fahrenheit)") +
  theme(legend.position = "bottom", text=element_text(size=12))

author_plot_mono
```

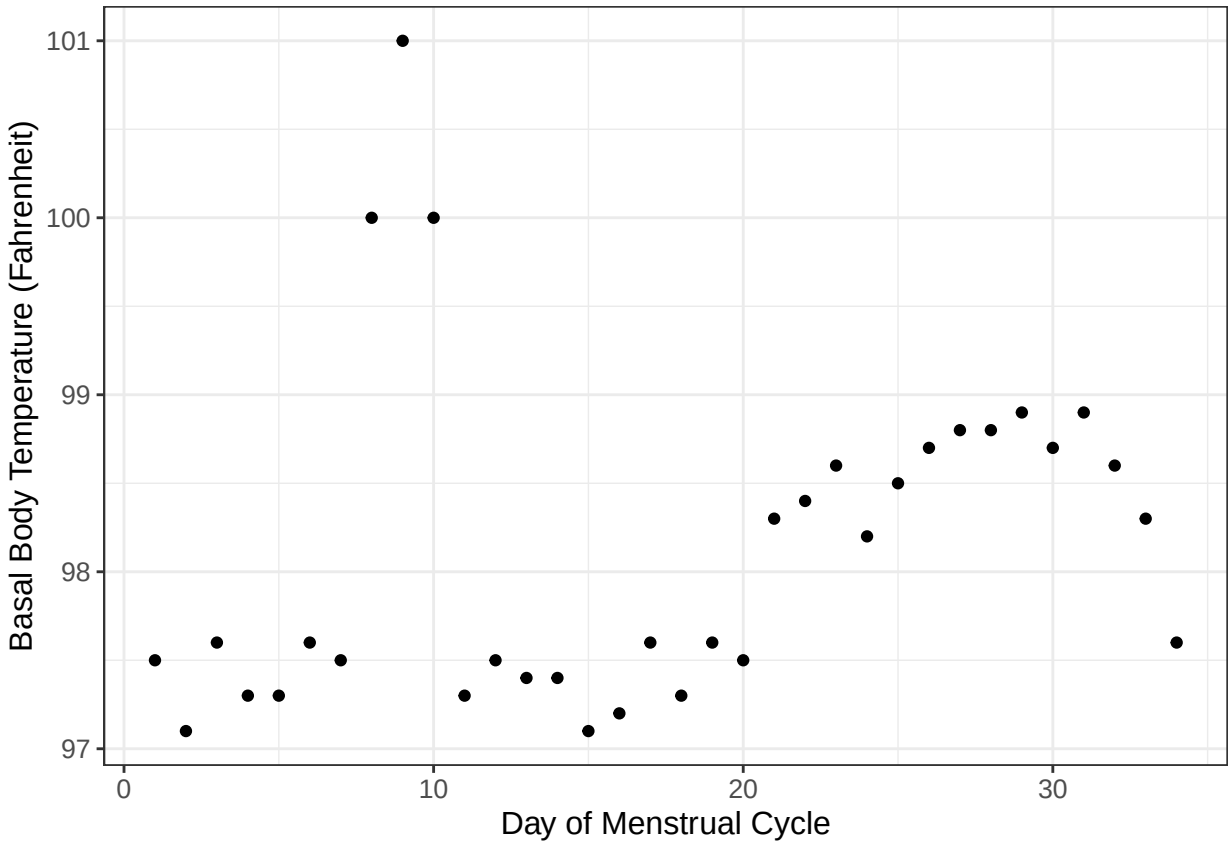


We see that the day of ovulation is still estimated to be day 24, although the model fit is more aggressively increasing–this may be a stronger assumption than we want to make in this setting.

Switching to a different woman, let's take a look at Tracy's data. From looking at a scatterplot of her data, we notice that Tracy had a strange spike in temperature on days 8-10 of her cycle.

```
tracy_dat <- filter(bbt_data, Name == "Tracy")

ggplot(data = tracy_dat) +
  geom_point(aes(x = Day, y = Temperature)) +
  theme_bw() +
  xlab("Day of Menstrual Cycle") +
  scale_y_continuous("Basal Body Temperature (Fahrenheit)") +
  theme(legend.position = "bottom", text=element_text(size=12))
```



Reviewing the data, we see that she reported an aberration (she was sick) on days 8-10:

```
head(tracy_dat, n = 10)
#> # A tibble: 10 x 5
#>    Name    Day Temperature Mucus Aberration
#>    <chr> <dbl>       <dbl> <chr>      <dbl>
#>  1 Tracy     1        97.5 n              0
#>  2 Tracy     2        97.1 n              0
#>  3 Tracy     3        97.6 n              0
#>  4 Tracy     4        97.3 n              0
#>  5 Tracy     5        97.3 n              0
#>  6 Tracy     6        97.6 n              0
#>  7 Tracy     7        97.5 n              0
#>  8 Tracy     8       100   n              1
#>  9 Tracy     9       101   n              1
#> 10 Tracy    10       100   n              1
```

Maybe we want to include aberration as a linear predictor, in the hopes of recovering her true ovulation time and the underlying fever-free trajectory. We do that by providing aberration as a linear predictor in the `Z` matrix option. Although we could stick with `hpr`'s default priors for the aberration coefficient, we're instead going to use a Cauchy prior (rather than the default normal prior), although we'll keep the default setting of a location of 0 and a scale parameter equal to 5 times the standard deviation of Tracy's aberration data.

Because we know we want to recover the fever-free trajectory, we're also going to include her `X` matrix and a hypothetical fever-free `Z` matrix using the `new_X`, `new_Z` parameters of `hpr`. Note that we *have* to include values of `X` and `Z` at which we want predictions in the original `hpr` call–we can't add that in after the fact, unfortunately.

```r
X <- as.matrix(tracy_dat$Day, ncol = 1)
y <- tracy_dat$Temperature
Z <- as.matrix(tracy_dat$Aberration, ncol = 1)
nofever_Z <- as.matrix(rep(0, nrow(tracy_dat)), ncol = 1)

tracy_hpr <- hpr(y = y, X = X, Z = Z, new_X = X, new_Z = nofever_Z, family = "gaussian",
                 beta_dist = "cauchy")
#> Running MCMC with 4 sequential chains...
#>
#> Chain 1 finished in 11.9 seconds.
#> Chain 2 Informational Message: The current Metropolis proposal is about to be rejected because of th
#> Chain 2 Exception: normal_lpdf: Location parameter[4] is inf, but must be finite! (in '/var/folders/
#> Chain 2 If this warning occurs sporadically, such as for highly constrained variable types like cova
#> Chain 2 but if this warning occurs often then your model may be either severely ill-conditioned or m
#> Chain 2
#> Chain 2 Informational Message: The current Metropolis proposal is about to be rejected because of th
#> Chain 2 Exception: normal_lpdf: Location parameter[4] is inf, but must be finite! (in '/var/folders/
#> Chain 2 If this warning occurs sporadically, such as for highly constrained variable types like cova
#> Chain 2 but if this warning occurs often then your model may be either severely ill-conditioned or m
#> Chain 2
#> Chain 2 Informational Message: The current Metropolis proposal is about to be rejected because of th
#> Chain 2 Exception: normal_lpdf: Location parameter[4] is inf, but must be finite! (in '/var/folders/
#> Chain 2 If this warning occurs sporadically, such as for highly constrained variable types like cova
#> Chain 2 but if this warning occurs often then your model may be either severely ill-conditioned or m
#> Chain 2
#> Chain 2 finished in 11.8 seconds.
#> Chain 3 Informational Message: The current Metropolis proposal is about to be rejected because of th
#> Chain 3 Exception: normal_lpdf: Scale parameter is nan, but must be positive! (in '/var/folders/5h/8
#> Chain 3 If this warning occurs sporadically, such as for highly constrained variable types like cova
#> Chain 3 but if this warning occurs often then your model may be either severely ill-conditioned or m
#> Chain 3
#> Chain 3 finished in 11.9 seconds.
#> Chain 4 Informational Message: The current Metropolis proposal is about to be rejected because of th
#> Chain 4 Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in '/var/folders/5h/8ph
#> Chain 4 If this warning occurs sporadically, such as for highly constrained variable types like cova
#> Chain 4 but if this warning occurs often then your model may be either severely ill-conditioned or m
#> Chain 4
#> Chain 4 Informational Message: The current Metropolis proposal is about to be rejected because of th
#> Chain 4 Exception: normal_lpdf: Scale parameter is 0, but must be positive! (in '/var/folders/5h/8ph
#> Chain 4 If this warning occurs sporadically, such as for highly constrained variable types like cova
#> Chain 4 but if this warning occurs often then your model may be either severely ill-conditioned or m
#> Chain 4
#> Chain 4 finished in 11.5 seconds.
#>
#> All 4 chains finished successfully.
#> Mean chain execution time: 11.8 seconds.
#> Total execution time: 47.5 seconds.
#>
#> Warning: 248 of 8000 (3.0%) transitions ended with a divergence.
#> This may indicate insufficient exploration of the posterior distribution.
#> Possible remedies include:
#>   * Increasing adapt_delta closer to 1 (default is 0.8)
#>   * Reparameterizing the model (e.g. using a non-centered parameterization)
```

```
#>   * Using informative or weakly informative prior distributions
```

Our diagnostics:

```
get_diagnostics(tracy_hpr)
#>   Divergences Max_Treedepth RHat Min_Ess_Bulk Min_Ess_Tail Num_Param
#> 1         248             0    0     1201.974     737.6647        34
#>   Num_Samples     Time
#> 1        8000 48.10167
```

We can look at the estimate of the coefficient of Aberration using the `get_betas` function:

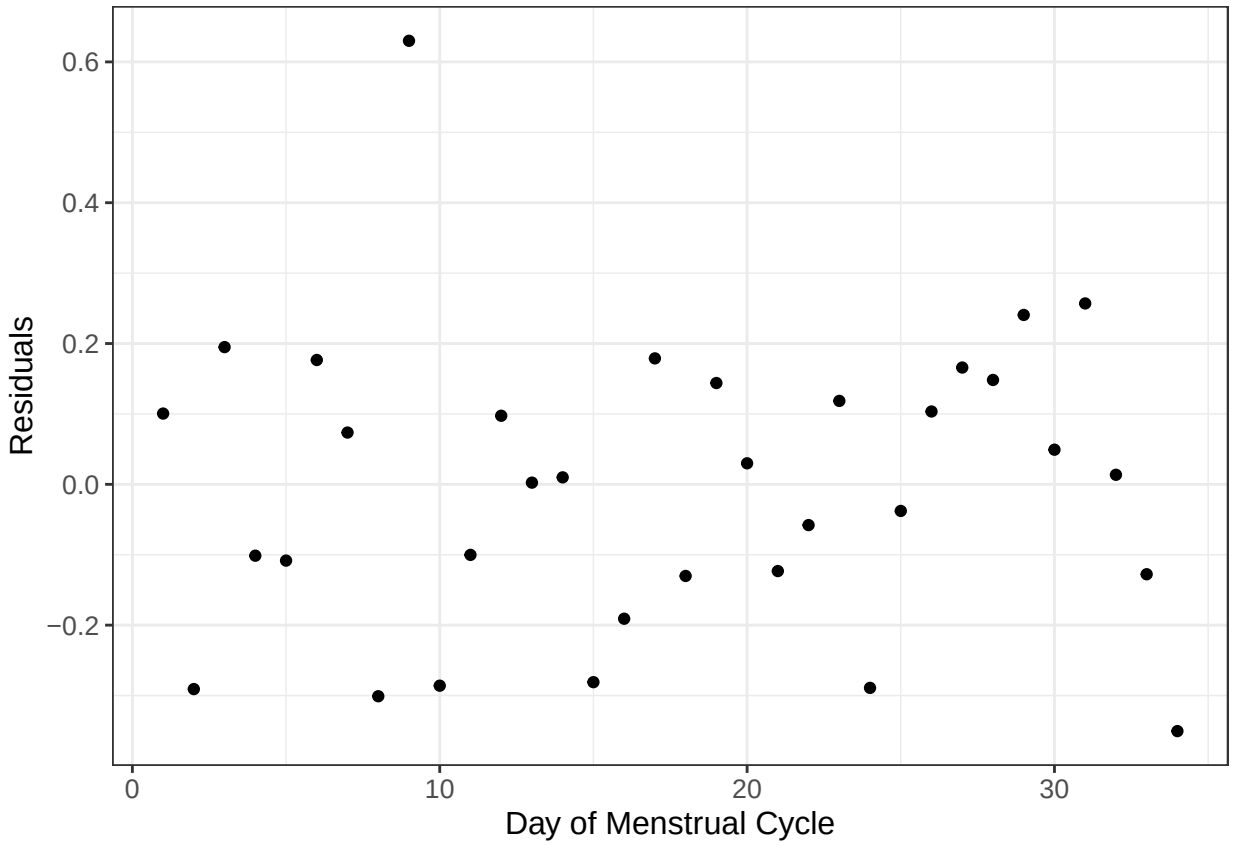
```
get_betas(tracy_hpr)
#> Warning: Dropping 'draws_df' class as required metadata was removed.
#>         Variable  Median     Mean    Lower    Upper
#> beta[1]        1 2.87374 2.865824 2.441163 3.252774
```

We see that Tracy's fever increased her temperature by 2.87 (2.44, 3.26) degrees Fahrenheit.

We can also get estimates of the linear predictor of the HPR model for the observed data using the `get_preds` function, which would allow us to look at residuals.

```
tracy_linpreds <- get_preds(tracy_hpr)
#> Warning: Dropping 'draws_df' class as required metadata was removed.
tracy_dat$Residual <- tracy_dat$Temperature - tracy_linpreds$Median

ggplot(data = tracy_dat) +
  geom_point(aes(x = Day, y = Residual)) +
  theme_bw() +
  xlab("Day of Menstrual Cycle") +
  scale_y_continuous("Residuals") +
  theme(legend.position = "bottom", text=element_text(size=12))
```



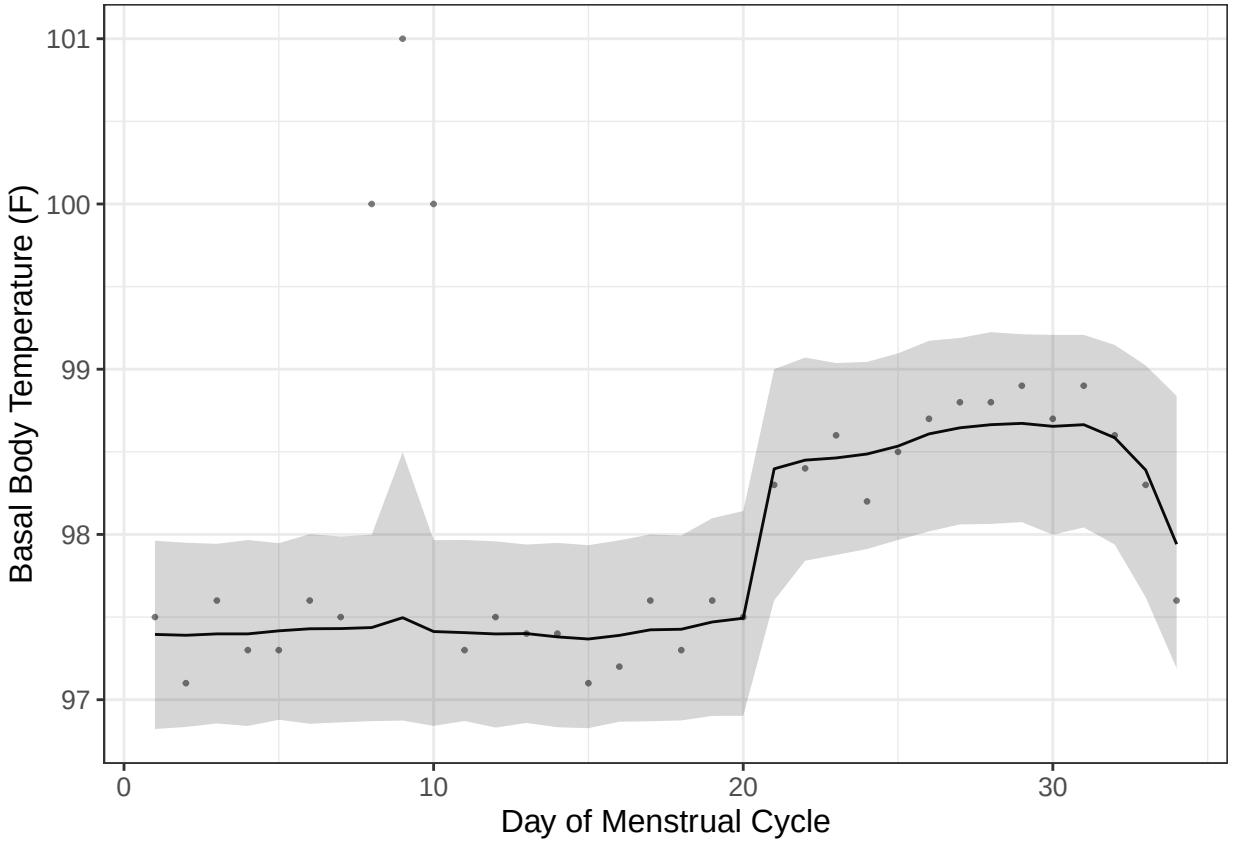
It looks like the function fit is generally cutting through the center of the observed data, although it isn't fully accommodating the fever spike on day 9, likely due to the shrinkage imposed by HPR.

And we review our fever-free posterior predictions using the `post_predict` function:

```
tracy_feverfree <- post_predict(tracy_hpr)
#> Running standalone generated quantities after 4 MCMC chains, 1 chain at a time ...
#>
#> Chain 1 finished in 0.0 seconds.
#> Chain 2 finished in 0.0 seconds.
#> Chain 3 finished in 0.0 seconds.
#> Chain 4 finished in 0.0 seconds.
#>
#> All 4 chains finished successfully.
#> Mean chain execution time: 0.0 seconds.
#> Total execution time: 2.2 seconds.
#> Warning: Dropping 'draws_df' class as required metadata was removed.
tracy_feverfree$x <- tracy_dat$Day

tracy_feverfree_plot <- ggplot(data = tracy_feverfree) +
  geom_line(aes(x = x, y = Median)) +
  geom_ribbon(aes(x = x, ymin = Lower, ymax = Upper), alpha = 0.2) +
  theme_bw() +
  geom_point(data = tracy_dat, aes(x = Day, y = Temperature), alpha = 0.5,
             size = 0.5) +
  xlab("Day of Menstrual Cycle") +
  scale_y_continuous("Basal Body Temperature (F)") +
  theme(legend.position = "bottom", text=element_text(size=12))
```

```
tracy_feverfree_plot
```



We see that the temperature spike is largely smoothed out in the fever-free predictions, with estimated ovulation occurring on day 20. Because these are posterior predictions, though, the uncertainty is a bit wider than what we would obtain from the `get_preds` function.

A note on computation

Because of how the Stan scripts are incorporated into this package, they have to be compiled the first time any new model script is run. `HPR` includes 13 different Stan scripts, some of which you may never use. (This vignette uses 2 of the 13 scripts.) If you are using a Stan script that you haven't used before, compilation will happen automatically, but note that computation times on that initial run may be 15-30 seconds longer than they normally would be, as the Stan script is first compiled and then run.