Bat virus underroost shedding model

Benny Borremans

Contents

Simulation model **2**
Number of bats above a sheet . . . 3
Model sample volume . . . 8
Shedding and Ct values . . . 14
Simulate data . . . 18

Prevalence estimation model **20**
Probability distributions for pooled Ct - N pos combinations . . . 22
Model stan . . . 23

bennyborremans@bbresearch.org

Last update: 26 Jul 2022.

The table of contents can be clicked to jump straight to specific sections.

Goal = create a model of underroost bat virus shedding.

Sheets are placed below roosts to collect urine from an estimated number of bats.
Urine samples on the sheet are pooled, and tested for RNA concentration using RT-PCR (Ct values).
Total sample volume depends on the number and volume of urine droplets on the sheets.
The number and species of bats above the sheet are estimated, but not all bats can always be observed, and bats can move after/before observation.
Samples are stored in one of two buffers (AVL/VTM), or without buffer (NB).
Buffer type affects PCR sensitivity.

The end result (Ct value in a pooled sample) depends on all these factors,
which makes it difficult to estimate shedding prevalence in the population.

The goal of this project is two-fold:
(1) Create a simulation model of the different processes that are believed to be involved, to get better insights and build intuition.
(2) Create a model to **estimate shedding prevalence in the population** from pooled samples.

Simulation model

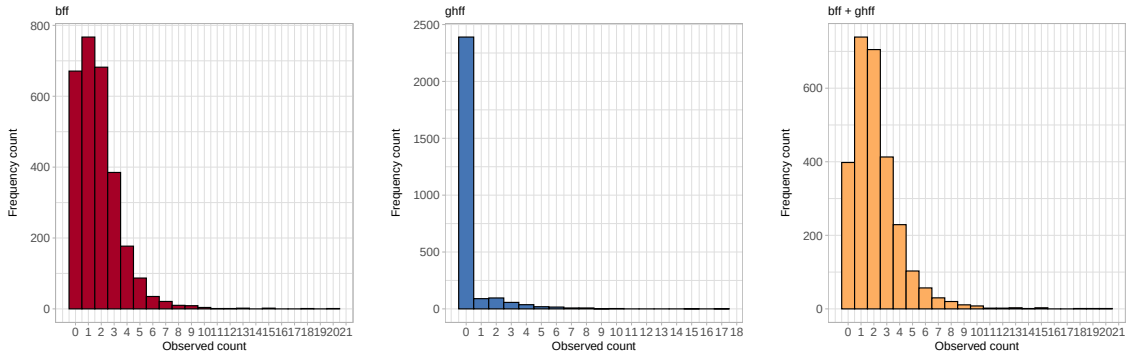
In order to get a better sense of the observation process resulting in pooled samples with certain Ct values, a number of aspects are investigated and simulated.
The last step is to combine these processes into one simulation model that generates realistic pooled underroost samples.

Number of bats above a sheet

There are counts for 2 species, black flying foxes and grey-headed flying foxes.
Counts can be morning, afternoon, and/or overall.
Observations have a level of confidence in the count and/or species.
These confidence levels are available for bff only (do they cover both bff and ghff, or bff only?).

Using only observations with a high level of confidence.
Using only morning observations (as these seem to be the most common).
Removing observations that are not exact (e.g. "5+").
All sites pooled.

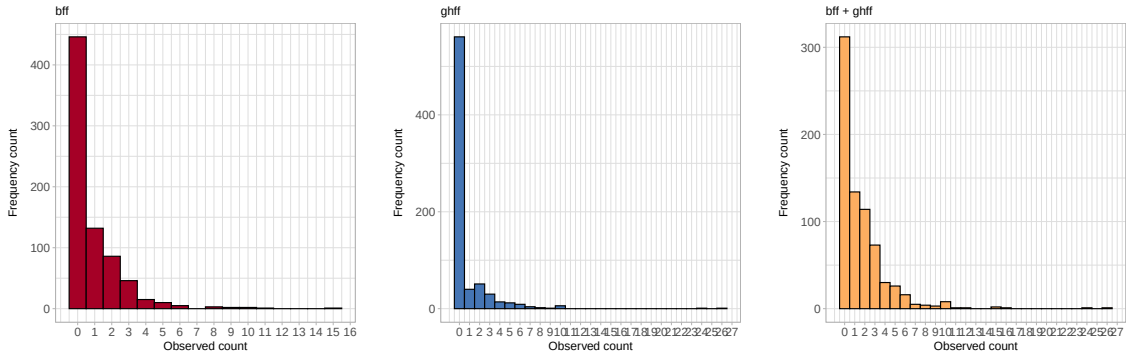
Histograms of counts:



Are higher numbers actually more rare, or just more uncertain?

==> check lower confidence counts.

Histograms of counts:



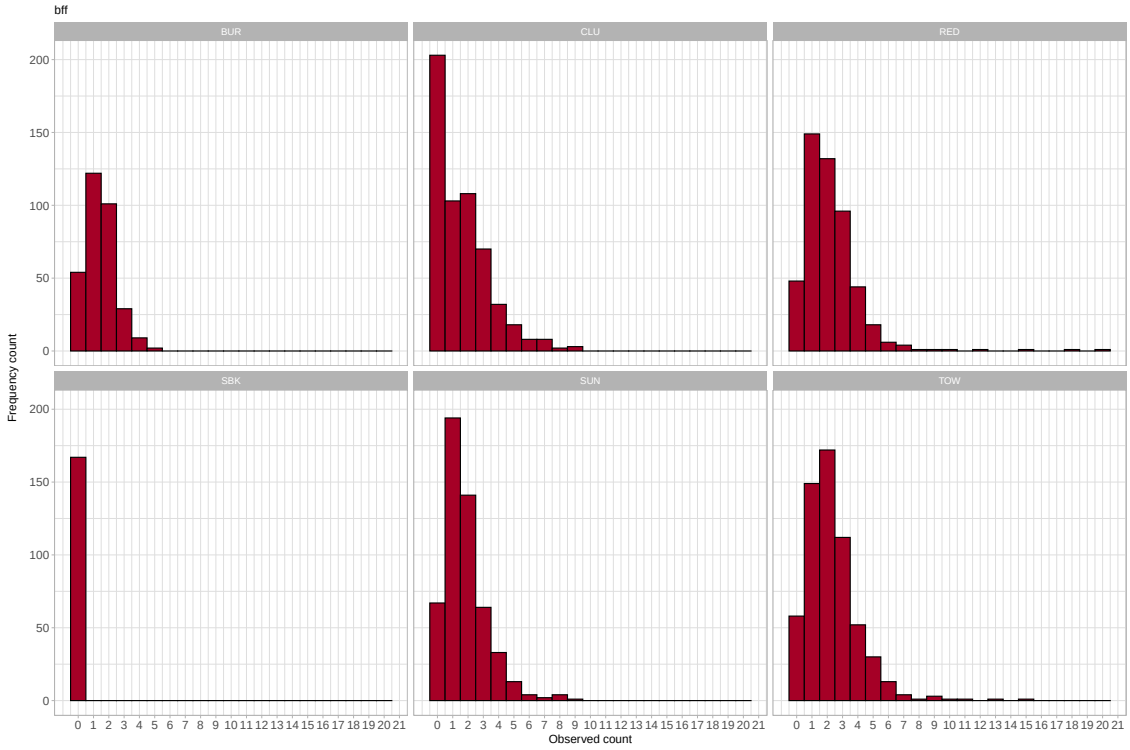
Lower-confidence counts are not higher, except for 2 ghff counts.

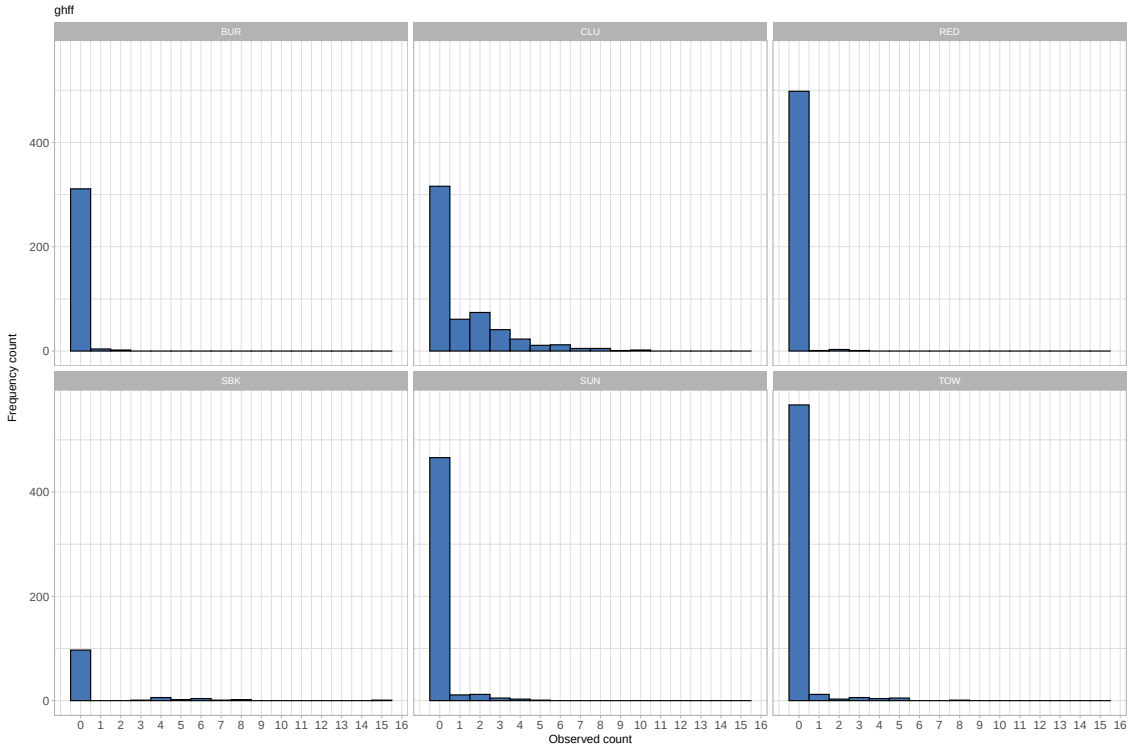
Are higher numbers actually more rare, or just written down as N+?

Not likely, there are only 25 entries with a + sign,
and these are one of: 5+, 10+, 3+, 1+.

Are there differences between sites?

Using only sites with more than 100 observations.

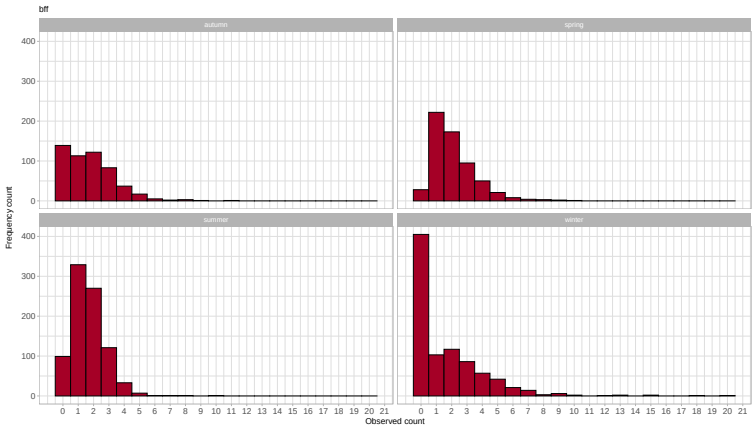




==> all look very similar, except many more 0 counts at CLU.
Any reason for this? Different conditions?
I didn't find anything in the notes, and the "bats" column mostly says "stable".

Any difference between seasons?

Histograms for different seasons.
bff only, most data available, don't need figure overload.



There seems to be an effect of season, that probably should be taken into account.

While spring and summer look like Poisson distributions,
autumn and winter seem closer to a mixture of a Bernoulli and a Poisson (probability of seeing any bats + if there are bats, how many).

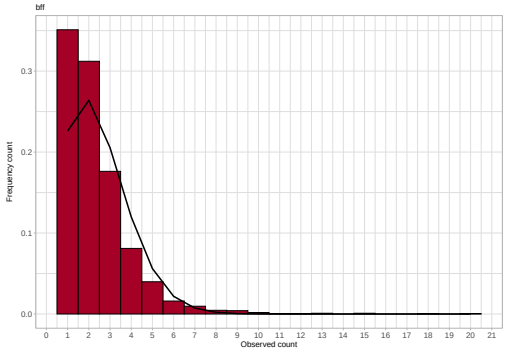
==> Fit model to positive counts only, as simulating 0 bats will not be useful.

Try Poisson distribution

All bff data pooled, across seasons and sites:

Lambda = 2.3359268.

Fitted distribution:

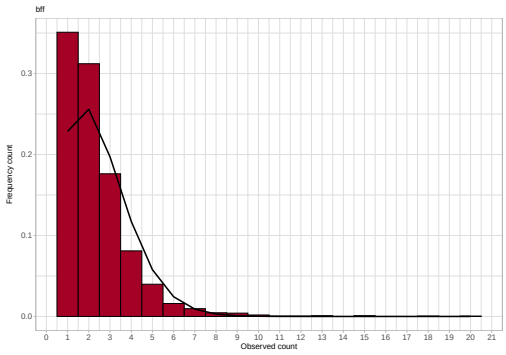


This distribution fits pretty well for the lower counts, but does not allow for the occasional higher numbers.

==> try a distribution with some more variance: negative binomial.

All bff data pooled, across seasons and sites:

Fitted distribution:

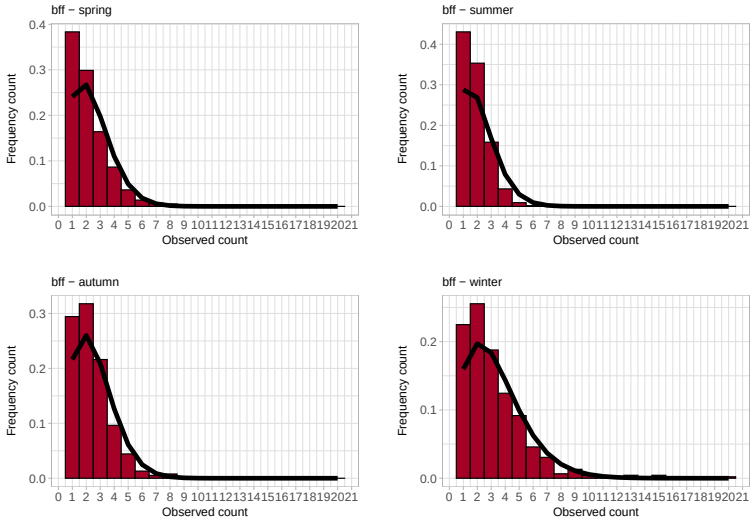


==> almost no difference, but a marginally wider tail for the negative binomial distribution.

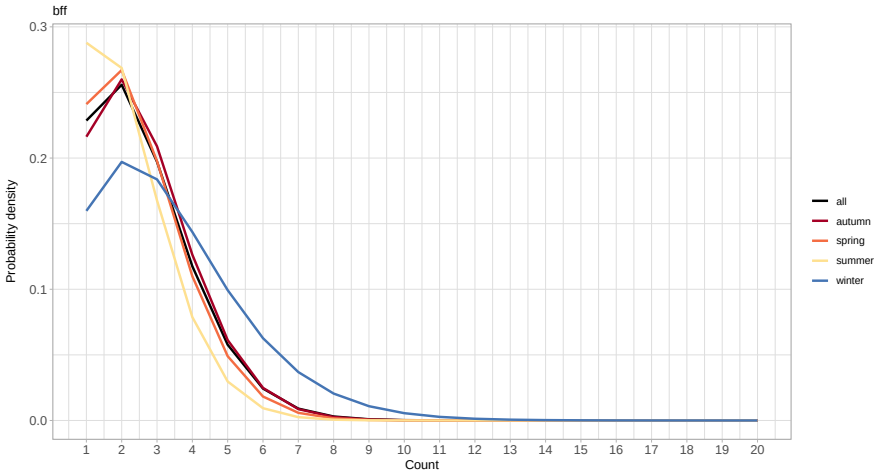
A distribution with a heavier tail would be a bit better, but for the simulation we can use the negative binomial distribution.

All bff data pooled, per season, across all sites:

Fitted distributions:



All combined:



==> fitted negative binomial distributions are very similar across seasons.

Conclusion:
Bat count can be modeled quite well using a Negative Binomial distribution, with:
$N_{bats} \sim NegBinom(size = 29.9, mu = 2.3)$

Model sample volume

How are sample volumne and the number of bats related?
More bats = more urine, but how strong is this correlation, and what is its shape?

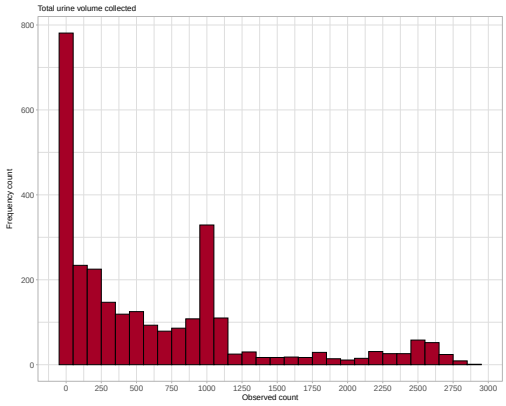
Considerations that need to be made when looking at this:
- There can be evaporation.
- The sample is added to buffer (how is this recorded in the data?)

How much urine can different numbers of bats produce?
What is the variation in collected urine volume, given a certain bat count?

And for each bat count, how does evaporation affect the distribution of volumes?

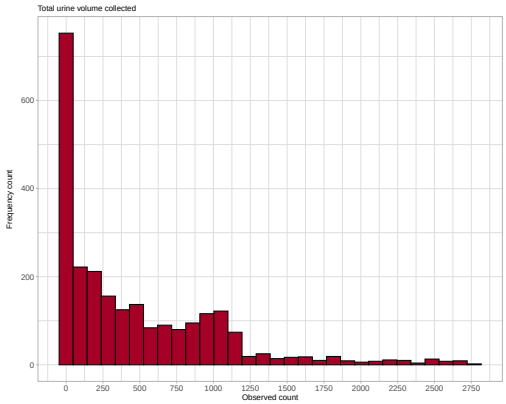
How common is evaporation in each season?

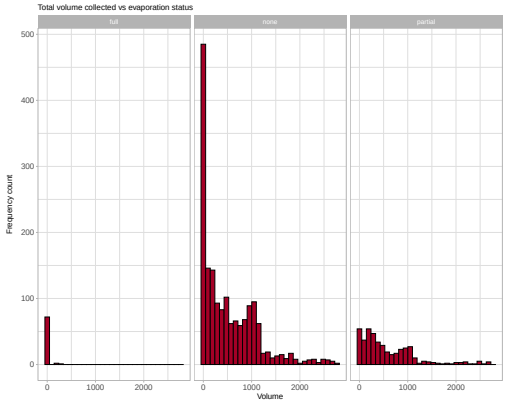
(using only confidence level = 1 data)
There are some volumes indicated as <X (e.g. < 140), which could be included when modeling by allowing censoring,
but to keep things simple these are just removed.



==> there are many zeros,
and a lot of 1000s.
Those 1000s are mostly the result of 500 for each of the vtm tubes.
Does this mean they weren't the maximum volume possible?

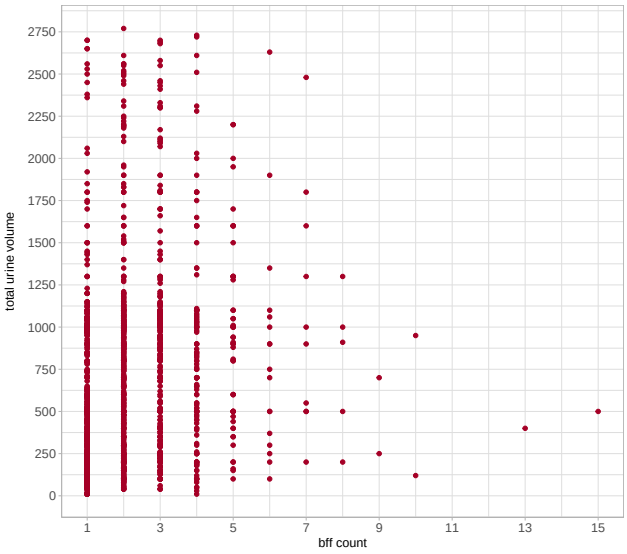
Removing these samples for now:





Correlation between bat counts and total volume.
Excluding counts of 0.
And excluding “full” evaporation, as that is always 0.

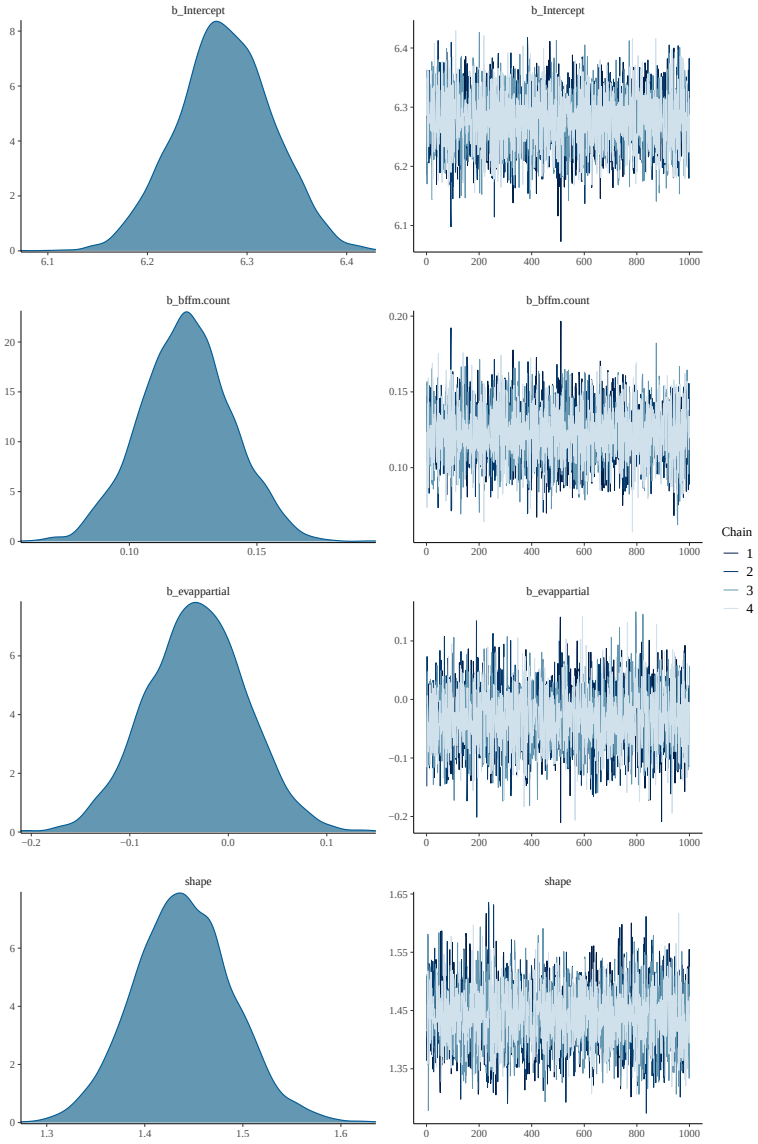
Even then, many total volumes are still 0, why is that?
Excluding total volume = 0 data for now.



Regression model:

total urine volume ~ bat count + evaporation status

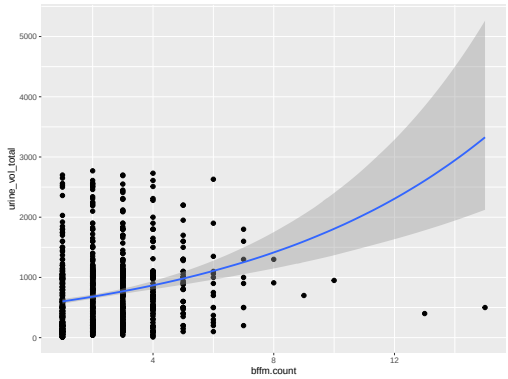
Output (log):

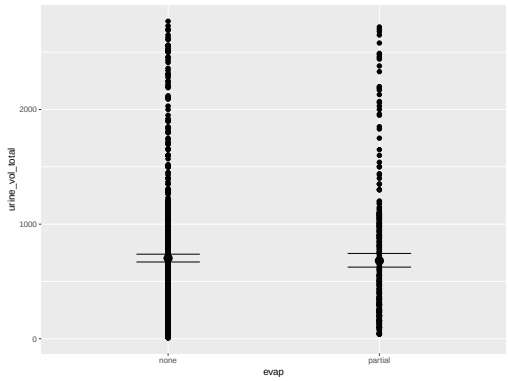


Back-transformed (exp) coefficients:

```
## [1] 532.6431483   1.1301609   0.9685217
```

Fitted functions/coefficients:



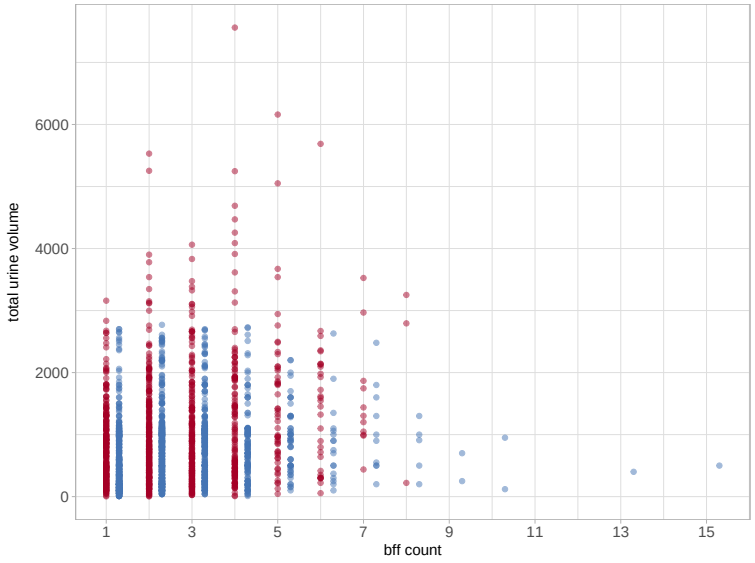


There is a positive correlation between bat count and total urine volume.
There is a minor effect of evaporation.

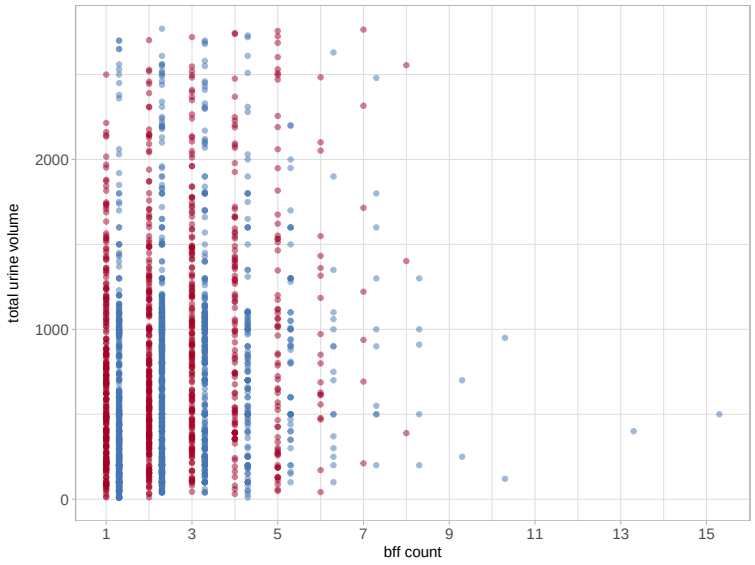
Can we use this model to adequately simulate urine volumes, given a bat count and evaporation status?

Bat counts are simulated using the negative binomial distribution fitted above (excluding 0s).
Evaporation status is simulated by randomly choosing 'none' or 'partial'.
Urine volume is then simulated using a gamma distribution with parameter values randomly selected from the model posterior.

Red = predicted, blue = observed.



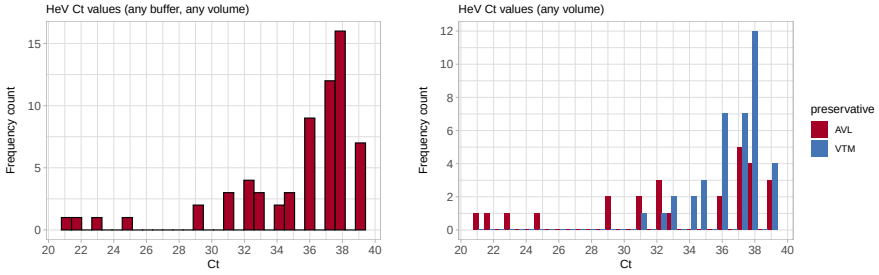
Simulation output is decently similar, but the large standard deviation results in the prediction of urine volumes larger than the ones observed.
==> should be excluded:


total urine volume
2000
1000
0
1
3
5
7
9
11
13
15
bff count

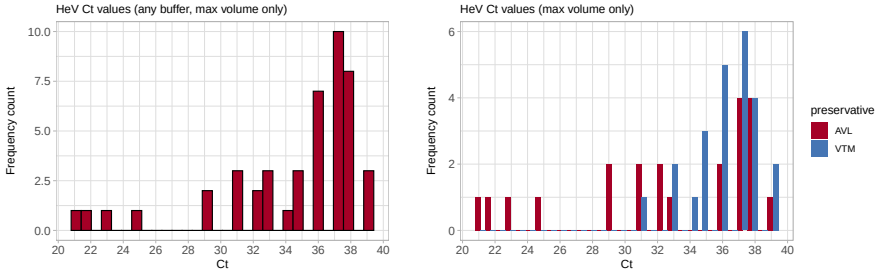
Shedding and Ct values

What is the distribution of Ct values being shed by individuals?
How is this affected by buffer type?
How does this translate to genome copies / virions?
What is the distribution of Ct values in underroost samples where bat count = 1?
What is the distribution of Ct values in underroost samples for any bat count?

Distribution of Ct values across buffers:

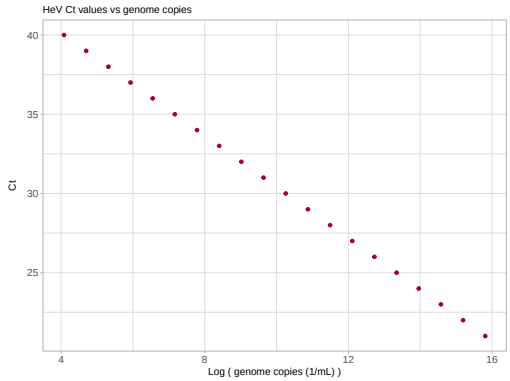


Distribution of Ct values across buffers, only max volumes:



Is this distribution typical for any sampling time, or does it depend on transmission dynamics and seasonality?
Likely it is a combination of both.
Let's look at Ct values over time.

Correlation between genome copies and Ct is log-linear:



Each increase in Ct translates to dividing the concentration by 1.8543854.

What does this mean for dilutions/pooling? I.e. when urine of a bat with a Ct value of 26 is mixed with the urine of a negative bat, what is the Ct value of the pooled sample?
We can create functions to convert Ct values and genome copy number,
which can be used to calculate expected Ct value given the contributions of different bats.

Examples to build intuition:

- Situation 1:
  Two bats, one positive with Ct 24.
  ==> Genome copies/mL of positive bat = 1.1547564×10^6.
  ==> Resulting genome copies/mL of pooled sample = $1.1547564 \times 10^6/2 = 5.7737822 \times 10^5$.
  ==> Resulting Ct of pooled sample = 25.1279367.

- Situation 2:
  Five bats, one positive with Ct 24.
  ==> Genome copies/mL of positive bat = 1.1547564×10^6.
  ==> Resulting genome copies/mL of pooled sample = $1.1547564 \times 10^6/5 = 2.3095129 \times 10^5$.
  ==> Resulting Ct of pooled sample = 26.6114114.

- Situation 3:
  Ten bats, one positive with Ct 24.
  ==> Genome copies/mL of positive bat = 1.1547564×10^6.
  ==> Resulting genome copies/mL of pooled sample = $1.1547564 \times 10^6/10 = 1.1547564 \times 10^5$.
  ==> Resulting Ct of pooled sample = 27.7336167.

- Situation 4:
  Three bats, one positive with Ct 24, one positive with Ct 28.
  ==> Genome copies/mL Ct 24 = 1.1547564×10^6.
  ==> Genome copies/mL Ct 28 = 9.7635848×10^4.
  ==> Resulting genome copies/mL of pooled sample = $(1.1547564 \times 10^6 + 9.7635848 \times 10^4 + 0)/3 = 4.1746409 \times 10^5$.
  ==> Resulting Ct of pooled sample = 25.6529768.

- Situation 5:
  Three bats, one positive with Ct 24, one positive with Ct 34.

==> Genome copies/mL Ct 24 = 1.1547564×10^6.
==> Genome copies/mL Ct 34 = 2400.422398.
==> Resulting genome copies/mL of pooled sample = $(1.1547564 \times 10^6 + 2400.422398 + 0)/3 = 3.8571895 \times 10^5$.
==> Resulting Ct of pooled sample = 25.7810227.

- Situation 6:
  Three bats, one positive with Ct 26, one positive with Ct 34.
  ==> Genome copies/mL Ct 26 = 3.3577615×10^5.
  ==> Genome copies/mL Ct 34 = 2400.422398.
  ==> Resulting genome copies/mL of pooled sample = $(3.3577615 \times 10^5 + 2400.422398 + 0)/3 = 1.1272552 \times 10^5$.
  ==> Resulting Ct of pooled sample = 27.7726406.

- Situation 7:
  Five bats, one positive with Ct 26, one positive with Ct 34.
  ==> Genome copies/mL Ct 26 = 3.3577615×10^5.
  ==> Genome copies/mL Ct 34 = 2400.422398.
  ==> Resulting genome copies/mL of pooled sample = $(3.3577615 \times 10^5 + 2400.422398 + 0)/5 = 6.7635315 \times 10^4$.
  ==> Resulting Ct of pooled sample = 28.5996673.

- Situation 8:
  Five bats, one positive with Ct 32, one positive with Ct 36.
  ==> Genome copies/mL Ct 32 = 8255.211691.
  ==> Genome copies/mL Ct 36 = 697.9866664.
  ==> Resulting genome copies/mL of pooled sample = (8255.211691 + 697.9866664 + 0)/5 = 1790.6396715.
  ==> Resulting Ct of pooled sample = 34.4791587.

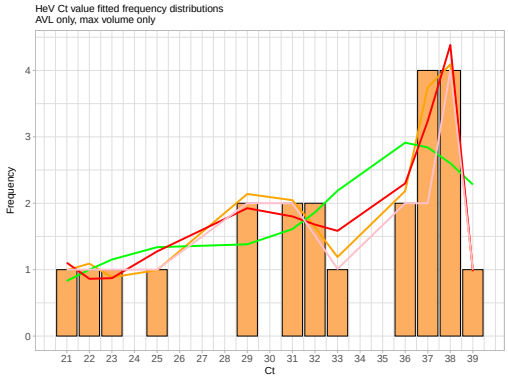
- Situation 9:
  Nine bats, one positive with Ct 27, one positive with Ct 32.
  ==> Genome copies/mL Ct 27 = 1.8106294×10^5.
  ==> Genome copies/mL Ct 32 = 8255.211691.
  ==> Resulting genome copies/mL of pooled sample = $(1.8106294 \times 10^5 + 8255.211691 + 0)/9 = 2.103535 \times 10^4$.
  ==> Resulting Ct of pooled sample = 30.4905393.

To simulate Ct values, we will need a distribution of Ct values in the population.
One possibility is to use the distribution observed for all samples in the dataset, which ignores temporal variation.
The fitting model will allow any distribution, and will allow it to vary over time, but regardless of the shape there must be a distribution in order for the model to be informative.

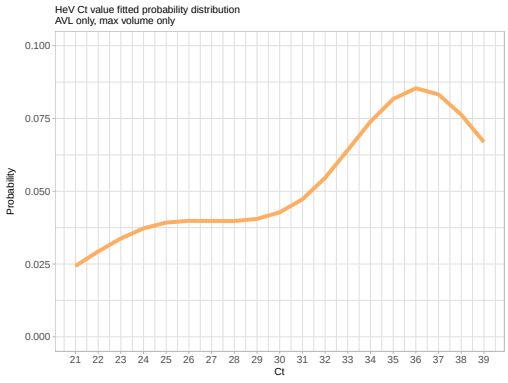
AVL samples only.
Max volumes only.

Orange = cubic spline
Green = natural spline
Red = b-spline
Pink = lowess smoothing

The natural spline makes the most sense, so let's use that one for now.



Probability distribution:



Simulate data

Putting everything together, we can now simulate underroost sample Ct values.

One way to do this is a process following these steps:

1. Choose a true prevalence in the population.
2. For each sample, generate a number of contributing bats using the distribution fitted to the observed data.
3. Assign a shedding status (neg/pos) to each of the bats, given the true prevalence.
4. To each positive bat, assign a Ct value given a provided probability distribution.
5. Calculate the resulting Ct value of the pooled sample, assuming equal contributions from each bat.

Code:

```
# 1.  Choose a true prevalence in the population.
true.prev = 0.15

# 2.  For each sample, generate a number of contributing bats
# using the distribution fitted to the observed data.
set.seed(123)
n.bat = rnbinom(1, size = 29.88, mu = 2.336)

# 3.  Assign a shedding status (neg/pos) to each of the bats,
# given the true prevalence.
bat.status = rbinom(n = n.bat, size = 1, prob = true.prev)

# 4.  To each bat, assign a Ct value given a provided probability distribution.
bat.ct = bat.status
bat.ct[which(bat.ct > 0)] = sample(ct.fit.prob$ct, size = sum(bat.status),
                                   replace = T, prob = ct.fit.prob$prob)

# 5.  Calculate the resulting Ct value of the pooled sample,
# assuming equal contributions from each bat.
ct.pooled = ct.pooled.fun(bat.ct)



# let's put this all into one function to make simulations easier
ur.sim.fun = function(prev, n){

       out.sim = vector(mode = "list", length = n+1)
       out.sim[[1]] = data.frame(ct = numeric(n), n.bat = numeric(n))

       for(i in 1:n){
              n.bat = max(1,rnbinom(1, size = 29.88, mu = 2.336))   # at least 1 bat
              bat.status = rbinom(n = n.bat, size = 1, prob = prev)
              bat.ct = bat.status
              bat.ct[which(bat.ct > 0)] = sample(ct2.prob$ct, size = sum(bat.status),
                                                 replace = T, prob = ct2.prob$prob)
```

```
            ct.pooled = ct.pooled.fun(bat.ct)

            out.sim[[1]][i,1] = ct.pooled
            out.sim[[1]][i,2] = n.bat
            out.sim[[i+1]] = bat.ct
      }


      return(out.sim)
      # first item of output list = data frame with pooled sample ct values,
      # and corresponding number of bats
      # remaining items are the Ct values of each bat of each pooled sample
}
```

Example output:

Prevalence estimation model

Given a few assumptions, we can now estimate shedding prevalence using the probability of seeing a certain Ct value in a pooled sample, given the number of contributing bats and given a distribution describing the individual probability of shedding at a certain Ct value.

Model assumptions are:
- The number of contributing bats is known.
- Each bat contributes the same amount.
- A probability distribution of individual Ct value is known (based on catch data), and is valid for the study bat population.
- Concentrations (Ct) in samples do not change after shedding (e.g. evaporation does not change concentration).

The data show that there is huge variation in shedding volume, and although there is a small statistically discernible sign that evaporation affects volume, we don't have the data to estimate a correction factor for the evaporation effect that links the individual Ct probability distribution and the Ct value of the pooled sample. This means that we have to ignore this effect of evaporation, and hence assume that there is none.

The assumption of known number of contributing bats is less strict.
Although the current (2022-07-26) version of the model assumes that numbers are known,
it is possible to allow uncertainty in this number (by treating the number of bats contributing to a sample as a parameter to be estimated, with a prior distribution informed by actual information, which can be something like "3-5 bats").
For now though, we assume that the number is known.

The probability distribution of individual Ct values can be estimated using catch data.
There are two obvious possible problems with this:
- Not enough data.
- Distribution changes over time (or seasonally, multi-annually, etc).

Theoretically, given a sufficiently large positive catch sample size, it would be possible to estimate a Ct probability distribution for every sampling session.
In reality though this will be hard, and periods will have to be pooled.
This means assuming that Ct distributions are the same for these pooled periods.
Currently we have one Ct distribution for the entire period, and we need to think whether this is a fair (not-too-wrong) assumption.

Model equations

$Ct_i \sim PooledCt(N_i, \theta_i)$

PooledCt = probability distribution to calculate likelihood for observing a Ct value in a pooled sample:

$$PooledCt = \sum_{x=0}^{N_i} \binom{N_i}{x} \theta^x (1-\theta)^{N_i - x} P(Ct|x, N_i),$$

where
$P(Ct|x, N_i)$ is calculated independently for each combination of N, x and Ct.
(see next section for details).

$\theta_i \sim Beta(\theta_{true} * \phi, (1 - \theta_{true} * \phi)$

Priors:
$\theta_{true} \sim Beta(1, 1)$ $\phi \sim Gamma(2, 1)$ (variation parameter, small values allow more variation, which is

necessary here because the small sample sizes (N bats contributing to a pooled sample) need a lower resolution.

i = pooled sample
Ct = Ct value of a pooled sample
N = number of bats contributing to a pooled sample
θ_i = "prevalence" in a pooled sample (proportion of positive bats contributing to the pooled sample), follows a distribution with mean θ_{true}
θ_{true} = true shedding prevalence in the population

Probability distributions for pooled Ct - N pos combinations

For each pooled underroost sample, we need to calculate the likelihood that it could have resulted from N bats of which X were positive with Ct values Y_x.

This can be done by first calculating all the probabilities, and then using these in the full model.
(As opposed to developing a complicated algorithm that does it every iteration).

The result with be a 3-dimensional table with dimensions:
1. N positive
2. N total
3. Pooled Ct

For example, if a sample has a Ct of 28 with 4 bats above the sheet, and a suggested number of positive bats of 2, we will be able to look up the sum of the probabilities of each possible combination of 2 Ct values diluted with 2 negative samples.
That is done using the individual-level probability distribution of Ct values.

Model stan

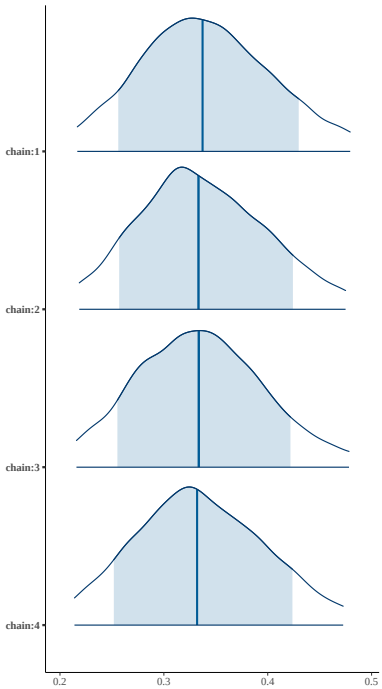
```
stan.model.ur.prev.Ct =
	"
	functions {

		// function to calculate binomial probability density, couldn't find non-log version in s
		real dbin(int npos, int ntotal, real prob){
			real probdens;
			probdens = choose(ntotal, npos) * prob^npos * (1-prob)^(ntotal-npos);
			return(probdens);
		}

		// probability mass function (lpmf)
		real ct_prob_lpmf(int ct, int nbat, real prev, real[] ct_probs){
			// first convert prevalence to indexable number of positives in a roundabout way, a
			int npos; // variable to index ct_probs
			real lprob; // likelihood
			real ctprob[nbat+1];	// vector for probabilities
			//int npos_index[nbat+1];

			// population index vector (stan doesn't allow 0:nbat
			//for(i in 1:(nbat+1) npos_index =

			// calculate P(Ct | prev, nbat) for each npos
			for(n in 1:(nbat+1)) {
				ctprob[n] = dbin(n-1, nbat, prev) * ct_probs[n];
			}


			lprob = log(sum(ctprob));

			return lprob;
		}

	}

	data {
		int<lower=1> N;	// number of samples
		int Ct[N];	// Ct of each sample
		int ct_index[N];	// index of each ct value in ct_array
		int bat_count[N]; // number of bats above the sheet
		int Ct_prob_Npos;
		int Ct_prob_Ntotal;
		int Ct_prob_Ct;
		real<lower=0, upper = 1> ct_array[Ct_prob_Npos,Ct_prob_Ntotal,Ct_prob_Ct];	// arra
	}

	parameters {
		real<lower=0,upper=1> prev_ur;
		real<lower=0,upper=1> prev_ind[N];
		real<lower=0> phi;

	}
```

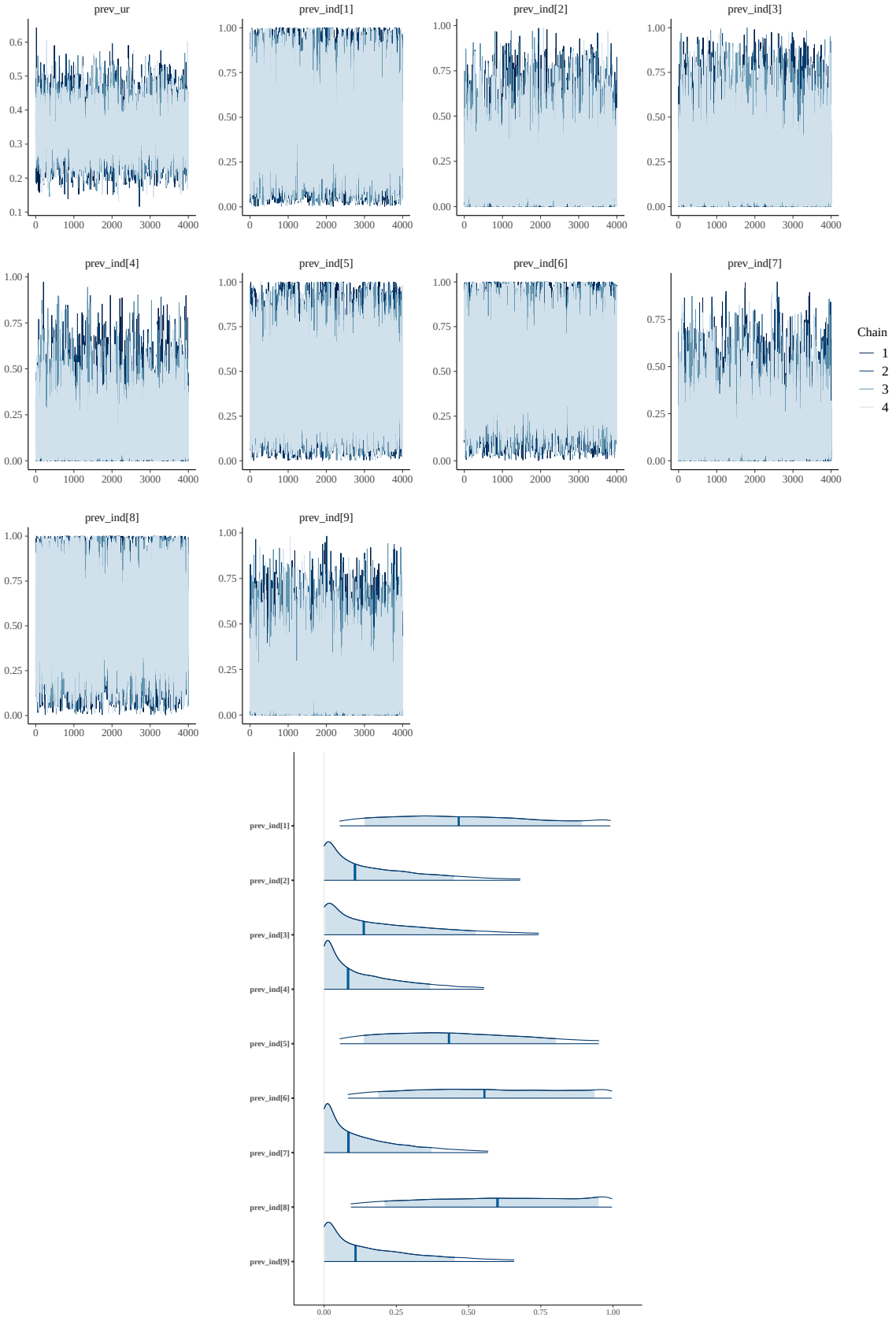
```
      model {
             phi ~ gamma(2,1);       // larger values = lower variance, less heterogeneity in the outc
             prev_ur ~ beta(1,1);

             for(i in 1:N) {
                    prev_ind[i] ~ beta(prev_ur * phi, (1 - prev_ur) * phi);
                    Ct[i] ~ ct_prob(bat_count[i], prev_ind[i], ct_array[,bat_count[i],ct_index[i]]);
             }
      }

      "
```

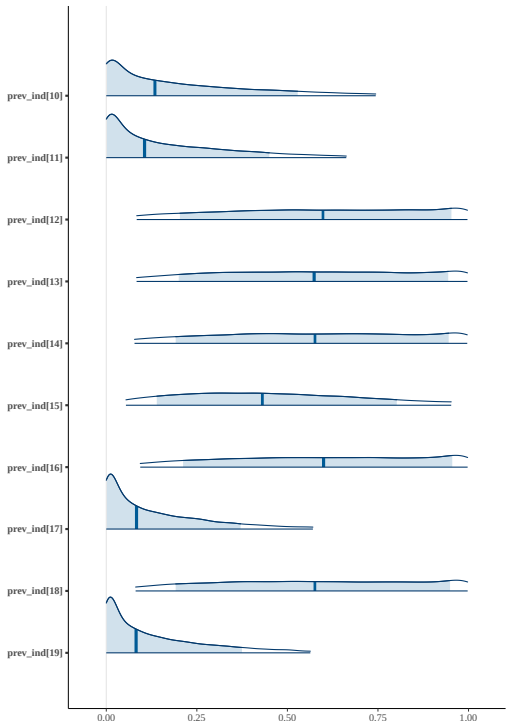
Simulation test 1

Simulated "true" prevalence = 0.27
N = 40

The simulated Ct values: 34, 0, 0, 0, 39, 33, 0, 37, 0, 0, 0, 37, 31, 27, 39, 36, 0, 27, 0, 25, 25, 0, 37, 0, 0, 39, 0, 35, 0, 0, 0, 36, 0, 0, 0, 0, 24, 31, 0, 0.
Number of bats per pooled sample: 3, 2, 1, 3, 2, 2, 3, 1, 2, 1, 2, 1, 2, 3, 2, 1, 3, 3, 3, 2, 4, 1, 1, 3, 2, 4, 1, 4, 1, 1, 5, 1, 2, 3, 3, 1, 3, 2, 3, 5.



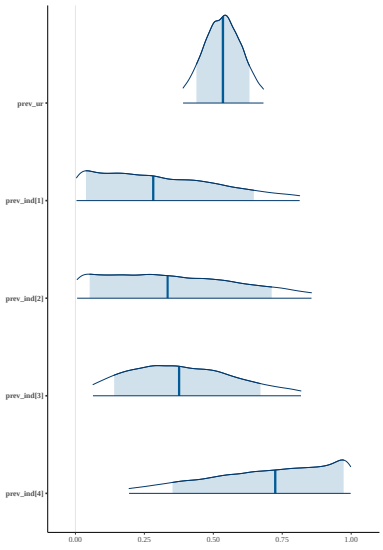

prev_ur
prev_ind[1]
prev_ind[2]
prev_ind[3]
prev_ind[4]
prev_ind[5]
prev_ind[6]
prev_ind[7]
prev_ind[8]
prev_ind[9]
Chain
1
2
3
4

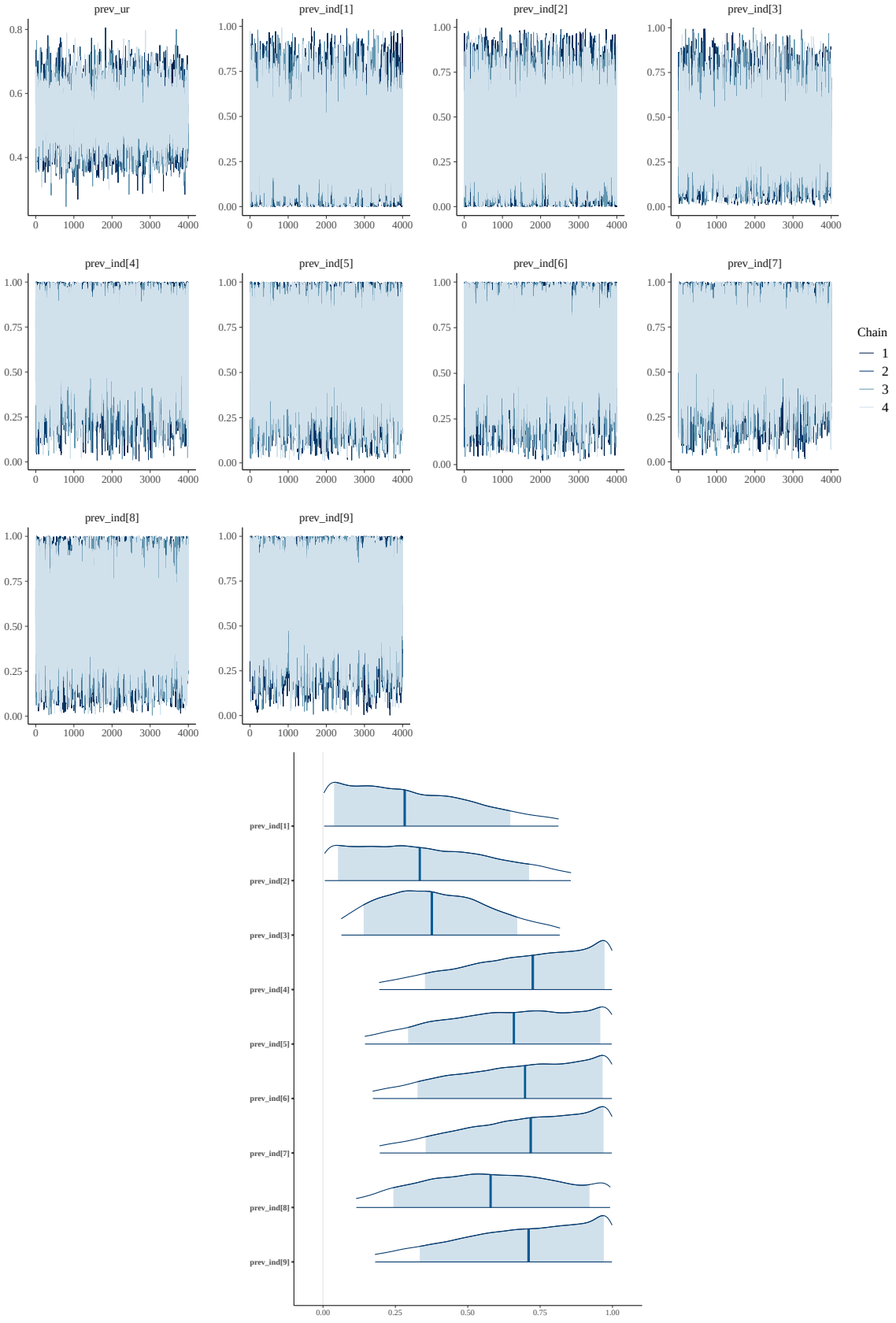

prev_ind[10]
prev_ind[11]
prev_ind[12]
prev_ind[13]
prev_ind[14]
prev_ind[15]
prev_ind[16]
prev_ind[17]
prev_ind[18]
prev_ind[19]
0.00
0.25
0.50
0.75
1.00

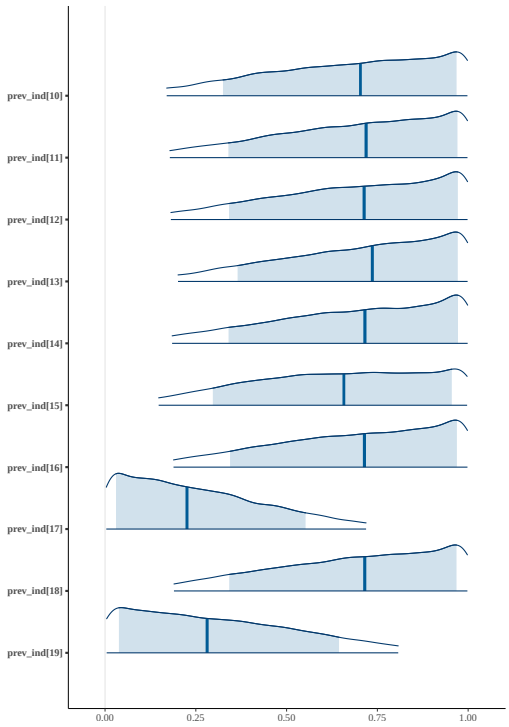
Simulation test 2

Simulated "true" prevalence = 0.62
N = 40

The simulated Ct values: 0, 0, 39, 25, 31, 26, 25, 33, 27, 31, 31, 32, 24, 29, 31, 37, 0, 29, 0, 38, 0, 0, 23, 0, 0, 34, 30, 33, 0, 34, 37, 31, 29, 24, 0, 0, 0, 0, 0, 37.
Number of bats per pooled sample: 2, 1, 4, 2, 3, 5, 4, 4, 2, 2, 1, 1, 4, 1, 3, 1, 3, 1, 2, 2, 1, 2, 3, 2, 2, 2, 4, 4, 1, 2, 1, 2, 1, 3, 1, 3, 1, 2, 1, 2.




prev_ur
prev_ind[1]
prev_ind[2]
prev_ind[3]
prev_ind[4]
prev_ind[5]
prev_ind[6]
prev_ind[7]
prev_ind[8]
prev_ind[9]
Chain
1
2
3
4


prev_ind[10]
prev_ind[11]
prev_ind[12]
prev_ind[13]
prev_ind[14]
prev_ind[15]
prev_ind[16]
prev_ind[17]
prev_ind[18]
prev_ind[19]
0.00
0.25
0.50
0.75
1.00