

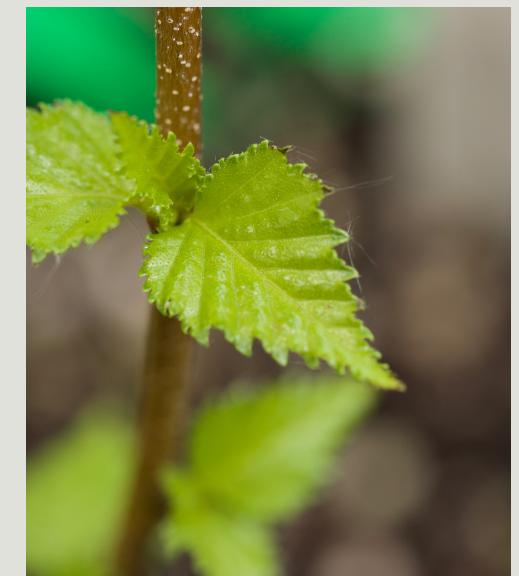
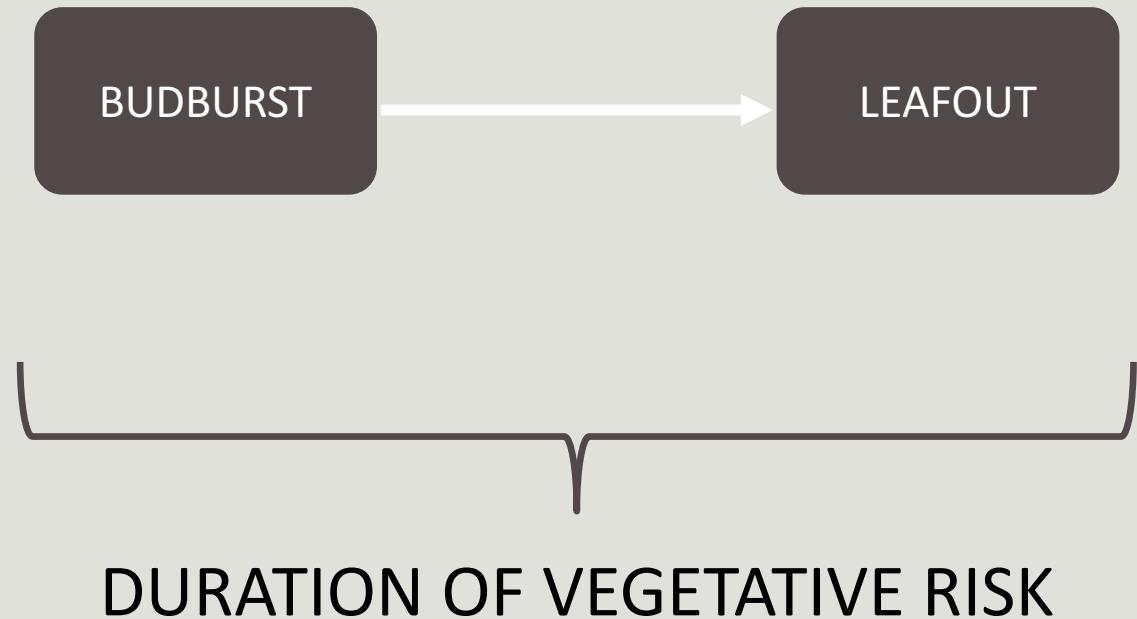
Buds Experiment

FALSE SPRING EVENTS

CATHERINE CHAMBERLAIN

False Spring Risk

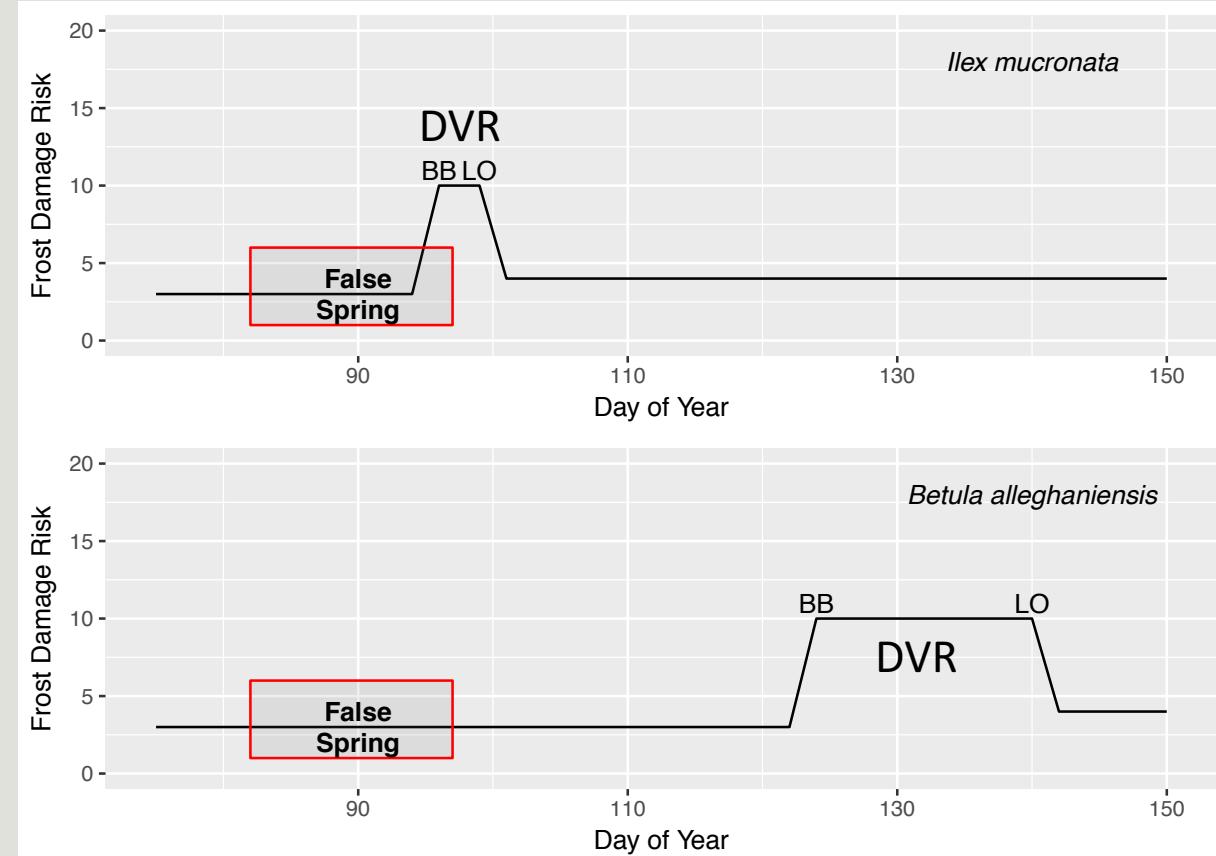
Interested in time between leaf budburst and leafout,
when frost tolerance is lowest but risk is still relatively high.



Betula populifolia - leafout

STRATEGIES:

Avoidance



Tolerance



Trichomes
on young
leaves

Serrations
along
leave
margins

Buds Experiment

Freezing Experiment with greenhouse individuals

Put individuals in growth chamber at
-3degC between budburst and leafout





Species: 2 -- BETPAP, BETPOP

Individuals: 14-15

Buds: 6-36

TX: Control vs. Freeze

$$\text{dvr} \sim \text{tx} + \text{species} + (1 | \text{individual})$$

DVR: Duration of Vegetative Risk

FAKE DATA: Improved!

```
##### SD for each treatment
spdifftsdiff = 0.1
txdiffsdiff = 0.1

mm <- model.matrix(~(sp+tx), data.frame(sp, tx)) ### ORDER HERE REALLY MATTERS!!! MAKE SURE IT LINES UP WITH "COEFF"
## Coding check below - keep for future tweaks to code/data
#coeff <- c(1, spdifftsdiff)
#risk <- rnorm(n = length(tx), mean = mm%*%coeff, sd = 1) # should be able to do sd = mm %*% sd.coeff as well, with a different sd for each parameter.
#(fake <- data.frame(risk, tx, sp))

##### Again, now with individuals.

baseinter = 11 # baseline intercept across all individuals for DVR
spint <- baseinter + c(1:nind)-mean(1:nind) # different intercepts by individual

fake <- vector()

for(i in 1:nind){ # loop over individual (random effect of model)

  # Give individuals different difference values, drawn from normal

  coeff <- c(spint[i],
             rnorm(1, spdifftsdiff, spdifftsdiff),
             rnorm(1, txdiffsdiff, txdiffsdiff)
  )

  dvr <- rnorm(n = length(tx), mean = mm %*% coeff, sd = 0.1)

  fakex <- data.frame(dvr, ind=i, sp, tx)

  fake<-rbind(fake, fakex)
}

summary(lm(dvr ~ tx+sp, data = fake)) # sanity check

# now fix the levels to 0/1 (not 1/2) as R does
fake$tx <- as.numeric(fake$tx)
fake$tx[fake$tx==1] <- 0
fake$tx[fake$tx==2] <- 1
```

```
##### Set up differences for each level  
spdiff = 0.2  
txdiff = 2.5  
  
##### SD for each treatment  
spdiff.sd = 0.1  
txdiff.sd = 0.1
```

Stan_glmer (dvr ~ tx + species + (1 | ind))

Fake

```
stan_glmer  
  family: gaussian [identity]  
  formula: dvr ~ tx + sp + (1 | ind)  
  -----  
  
Estimates:  
  Median MAD_SD  
(Intercept) 10.7  0.5  
tx          2.5  0.0  
sp          0.2  0.0  
sigma       0.1  0.0
```

```
Error terms:  
  Groups   Name      Std.Dev.  
  ind      (Intercept) 1.42  
  Residual           0.12  
Num. levels: ind 7
```

```
Sample avg. posterior predictive  
distribution of y (X = xbar):  
  Median MAD_SD  
mean_PPD 12.3  0.0
```

```
-----  
For info on the priors used see help('prior_summary.stanreg').
```

Real!

```
stan_glmer  
  family: gaussian [identity]  
  formula: dvr ~ tx + sp + (1 | ind)  
  -----  
  
Estimates:  
  Median MAD_SD  
(Intercept) 10.3  1.5  
tx          2.5  1.0  
sp          0.0  0.7  
sigma       6.1  0.2
```

```
Error terms:  
  Groups   Name      Std.Dev.  
  ind      (Intercept) 3.5  
  Residual           6.1  
Num. levels: ind 15
```

```
Sample avg. posterior predictive  
distribution of y (X = xbar):  
  Median MAD_SD  
mean_PPD 11.4  0.4
```

```
-----  
For info on the priors used see help('prior_summary.stanreg').
```

```

- data {
  int<lower=0> N;
  int<lower=0> n_ind;
  int<lower=0> n_sp;
  int<lower=1, upper=n_ind> ind[N];
  vector[N] dvr;
  vector[N] tx;
  vector[N] sp;
}

- parameters {
  vector[n_ind] a_ind;
  vector[n_ind] b_tx;
  vector[n_ind] b_sp;

  real mu_a;
  real mu_b_tx;
  real mu_b_sp;

  real<lower=0> sigma_b_tx;
  real<lower=0> sigma_b_sp;

  real<lower=0> sigma_a;
  real<lower=0> sigma_y;
}

```

transformed parameters {
 vector[N] y_hat;

 for(i in 1:N){
 y_hat[i] = a_ind[ind[i]] +
 b_sp[ind[i]] * sp[i] +
 b_tx[ind[i]] * tx[i]
 ;
 }
}

model {
 // Priors. Make them flat
 mu_b_tx ~ normal(0, 15);
 mu_b_sp ~ normal(0, 15);

 sigma_b_tx ~ normal(0, 10);
 sigma_b_sp ~ normal(0, 10);

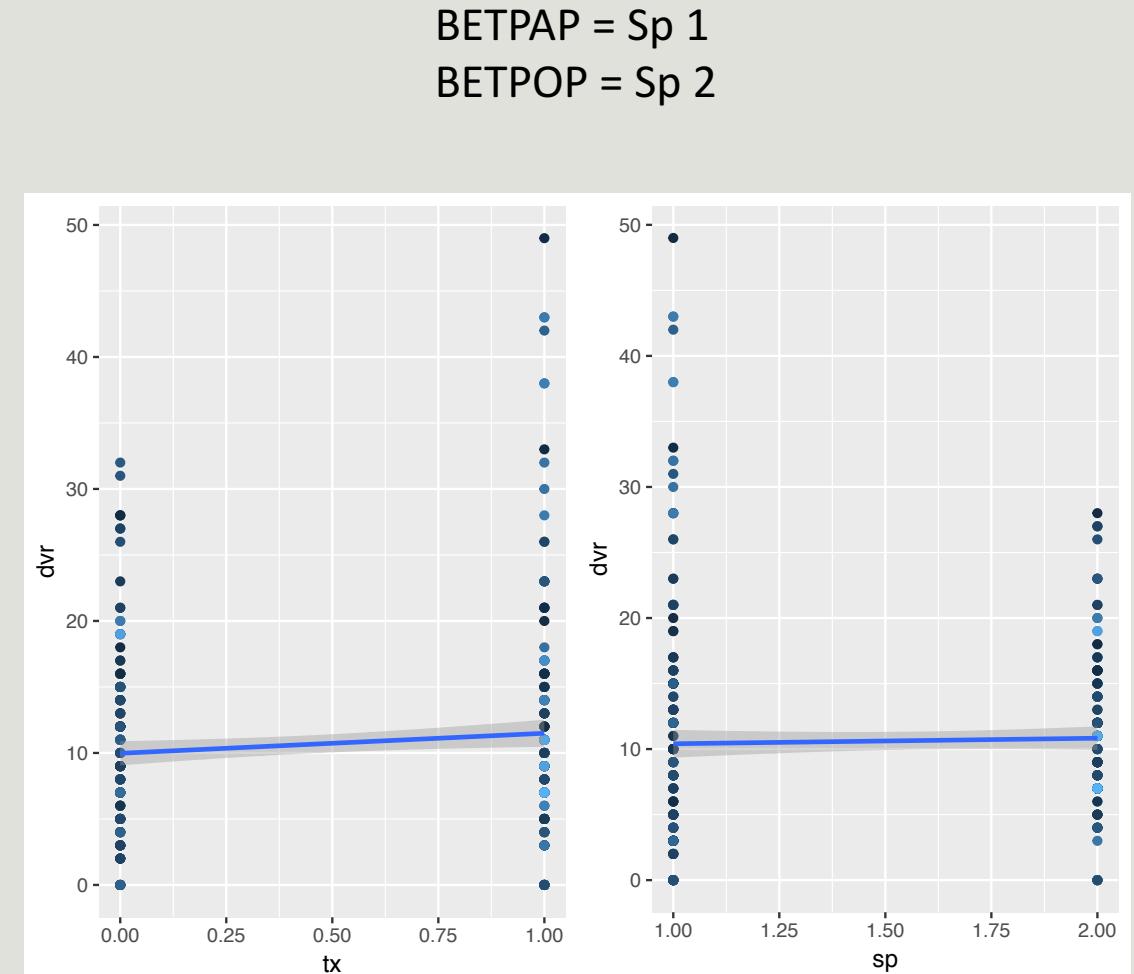
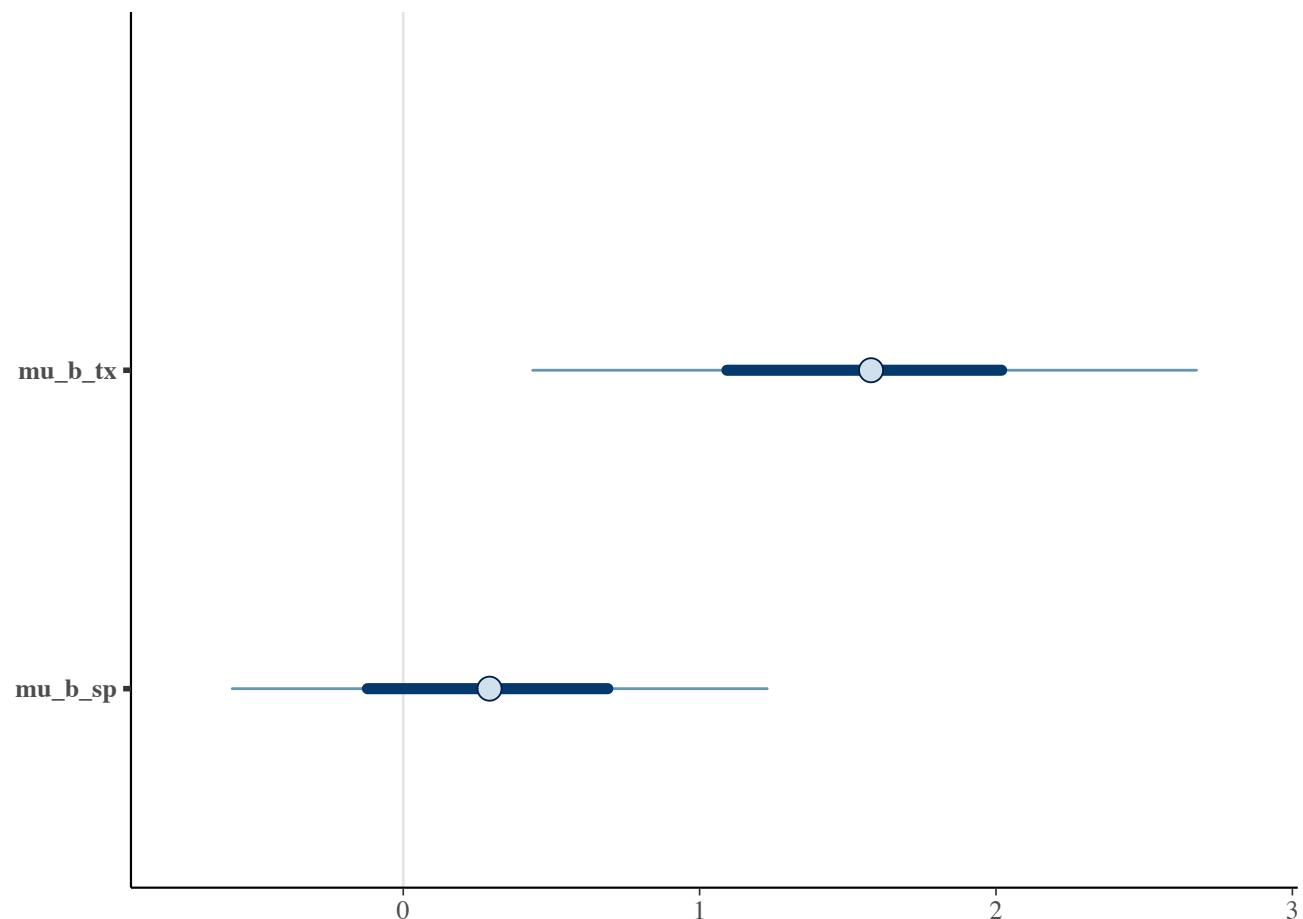
 a_ind ~ normal(mu_a, sigma_a);

 b_tx ~ normal(mu_b_tx, sigma_b_tx);
 b_sp ~ normal(mu_b_sp, sigma_b_sp);

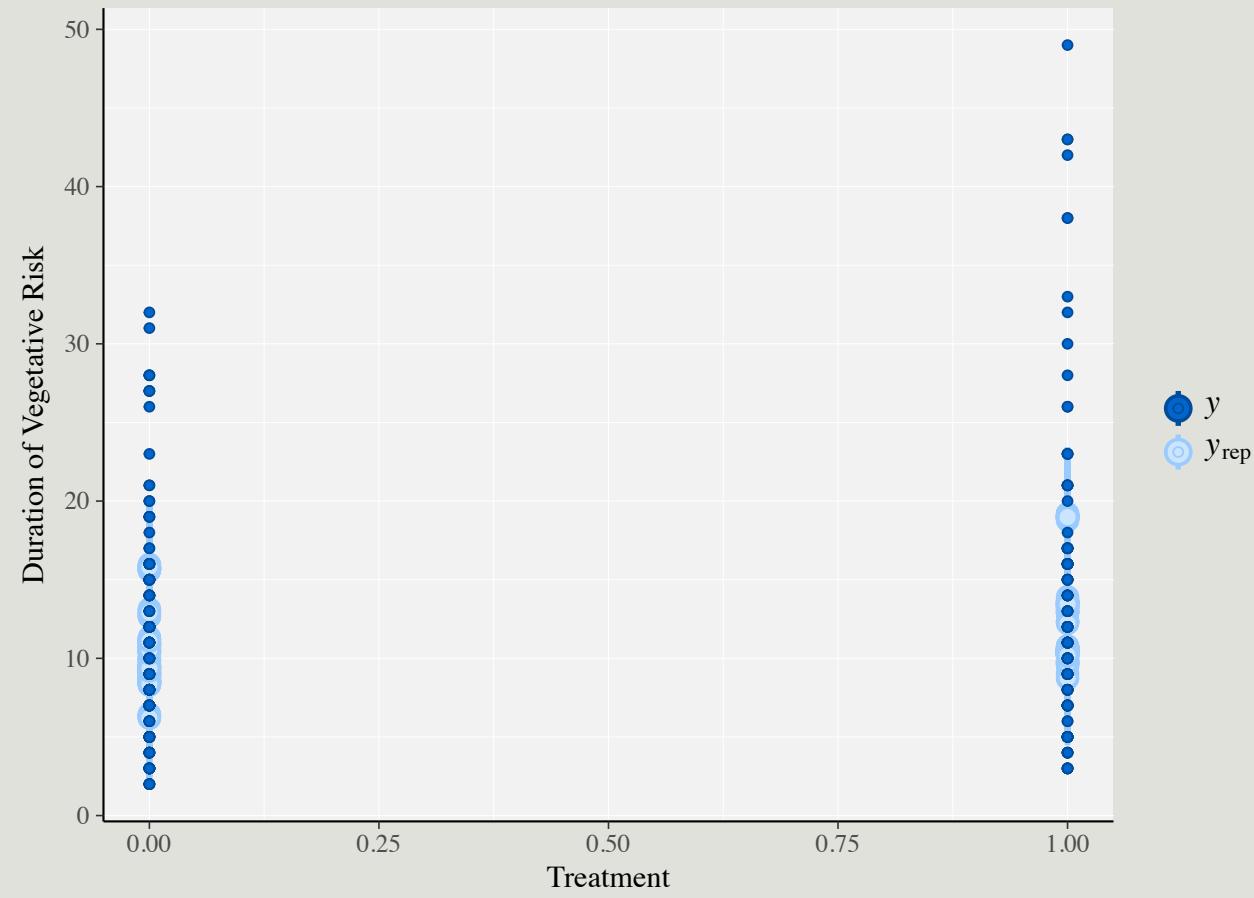
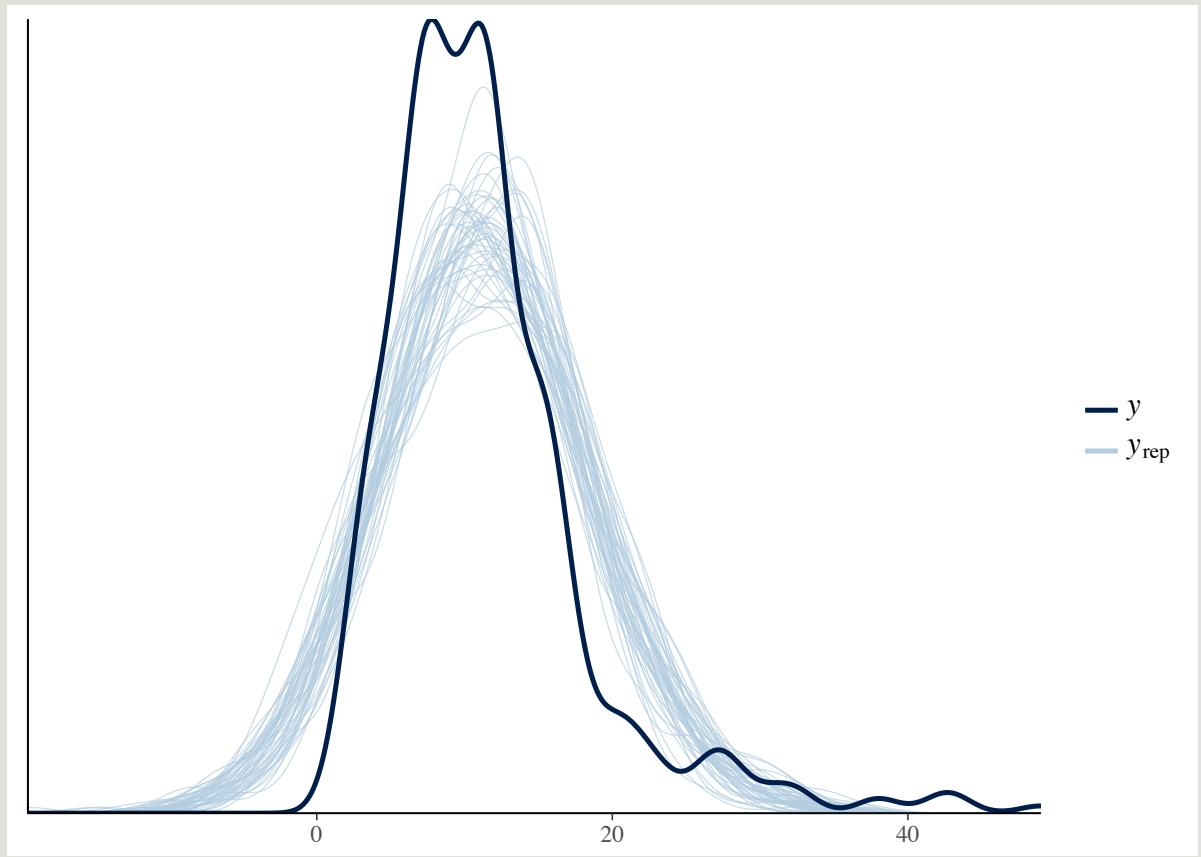
 dvr ~ normal(y_hat, sigma_y);
}

Stan Model:
 8000 Iterations
 4000 Warm-up
 4 Chains

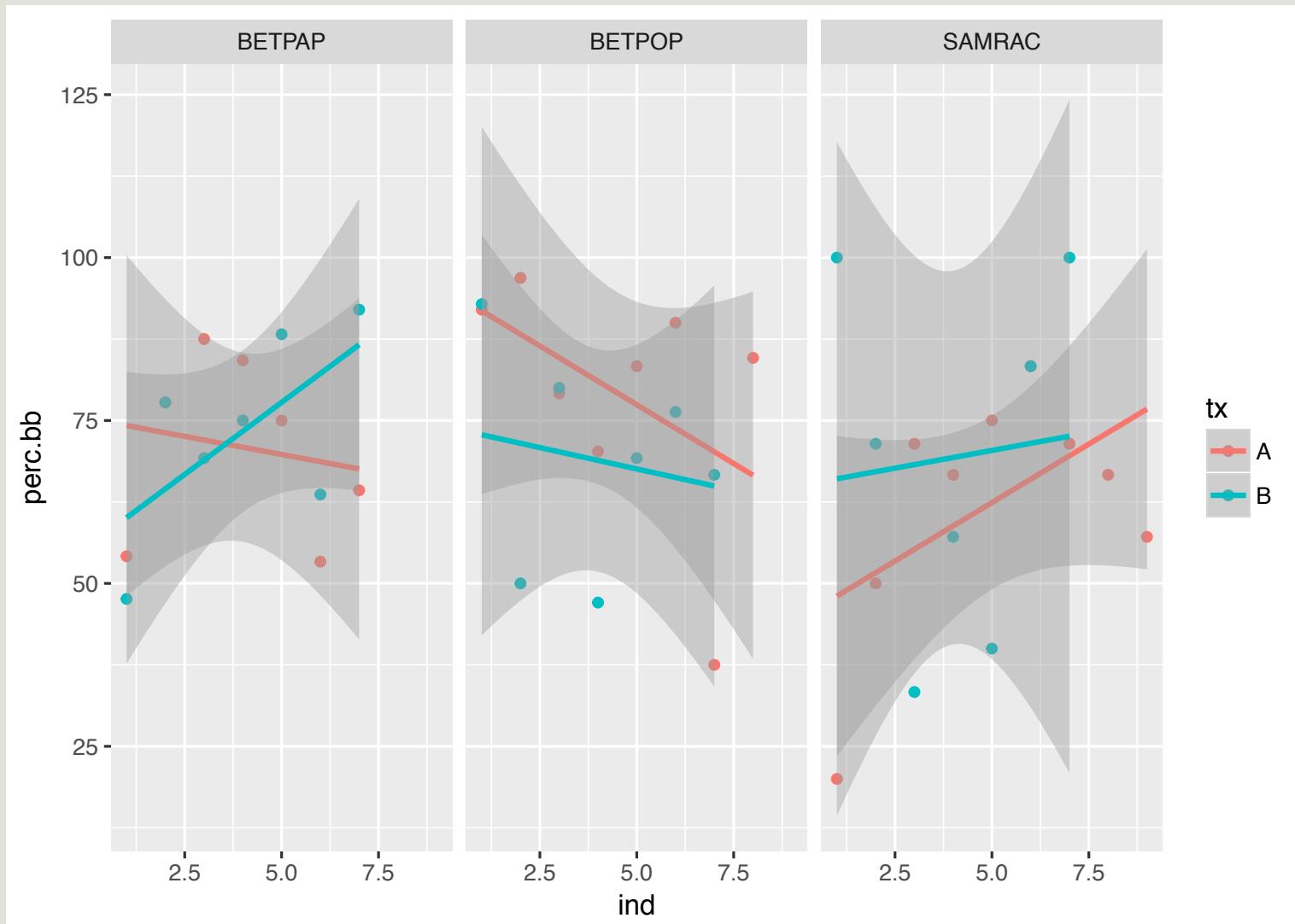
Real Data Output



PP_Check



Percent Budburst Model: percBB ~ tx + species



FAKE DATA:

```
##### Set up differences for each level
txdiff = -.04
spdiff = .01

##### SD for each treatment
txdiff.sd = 0.005
spdiff.sd = 0.001

mm <- model.matrix(~(tx+sp), data.frame(tx,sp)) ### ORDER HERE REALLY MATTERS!!! MAKE SURE IT LINES UP WITH "COEFF"

##### Again, now with individuals.

baseinter = 0.70 # baseline intercept across all individuals for DVR

coeff <- c(baseinter,
            rnorm(1, txdiff, txdiff.sd),
            rnorm(1, spdiff, spdiff.sd)
        )

perc <- rnorm(n = length(tx), mean = mm %*% coeff, sd = 0.01)

fake <- data.frame(perc, sp, tx)

summary(lm(perc ~ tx + sp, data = fake)) # sanity check

# now fix the levels to 0/1 (not 1/2) as R does
fake$tx <- as.numeric(fake$tx)
fake$tx[fake$tx==1] <- 0
fake$tx[fake$tx==2] <- 1

summary(lm(perc ~ tx + sp, data = fake)) # double check
```

Stan_betareg (perc ~ tx + species, link="logit", link.phi="log")

Fake

```
stan_betareg
  family: beta [logit, link.phi=log]
  formula: perc ~ tx + sp
  -----
  Estimates:
    Median MAD_SD
  (Intercept)  0.8   0.0
  tx          -0.2   0.0
  sp          0.1   0.0
  (phi)      1727.0 506.5
  Sample avg. posterior predictive
  distribution of y (X = xbar):
    Median MAD_SD
  mean_PPD 0.7   0.0
  -----
  For info on the priors used see help('prior_summary.stanreg').
```

Real!

```
stan_betareg
  family: beta [logit, link.phi=log]
  formula: perc ~ tx + sp
  -----
  Estimates:
    Median MAD_SD
  (Intercept)  0.9   0.5
  tx          -0.2   0.3
  sp          0.2   0.3
  (phi)       6.8   1.7
  Sample avg. posterior predictive
  distribution of y (X = xbar):
    Median MAD_SD
  mean_PPD 0.7   0.0
  -----
  For info on the priors used see help('prior_summary.stanreg').
```

Rstanarm works but rstan proving to be more tricky...

** Not including *Sambucus racemosa*

Issues:

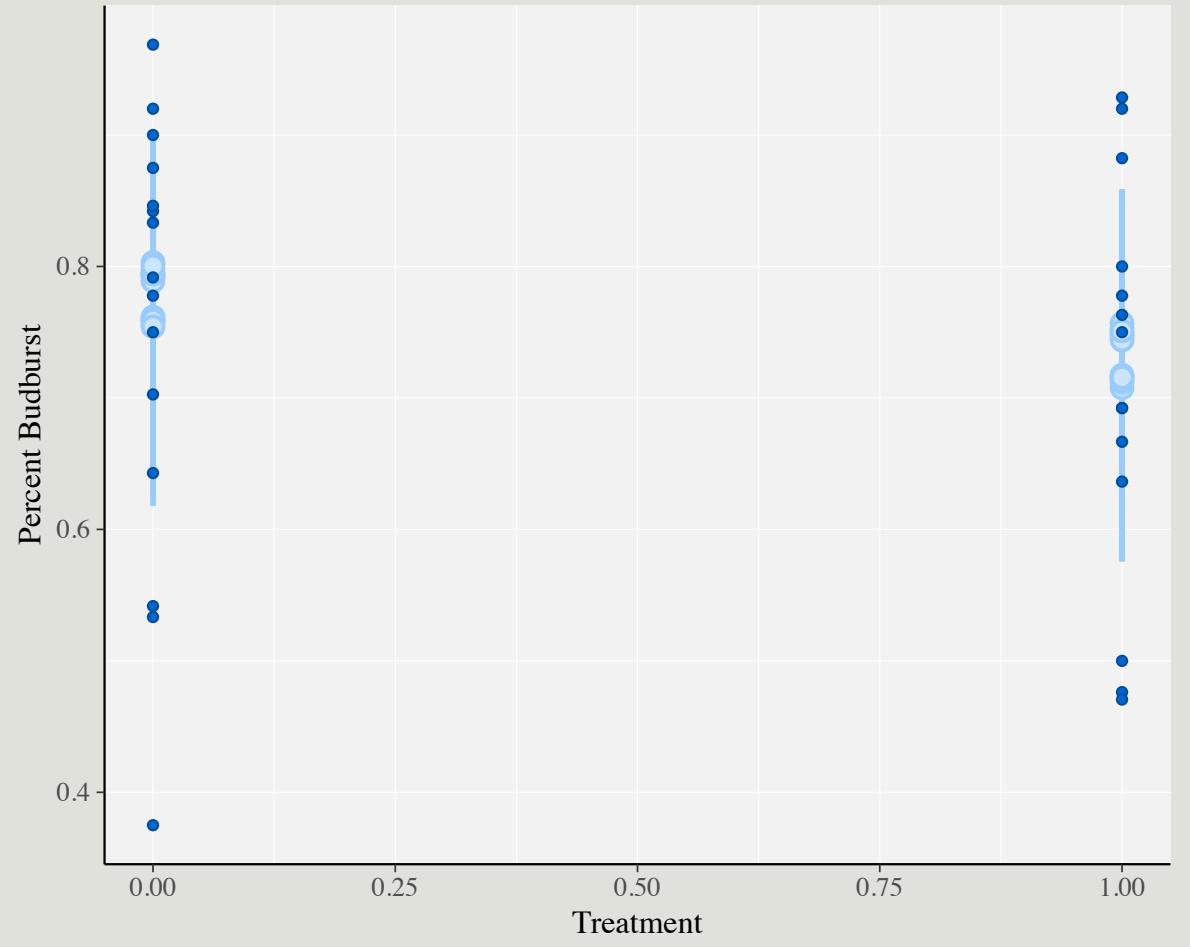
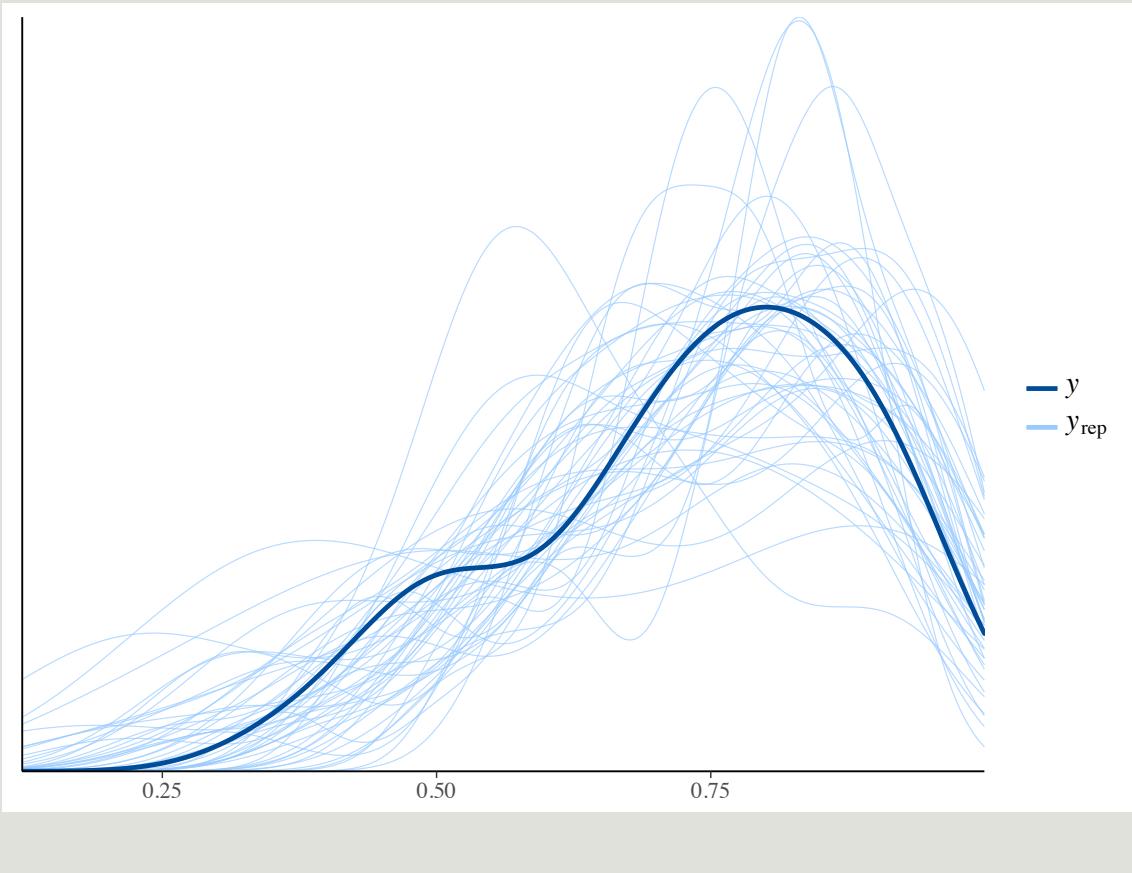
Posterior Predictive Checks

Binomial distribution not working

Different scale

```
data {  
    int<lower=0> N;  
  
    // Dependent variable  
    real<lower=0, upper=1> perc[N];  
  
    vector[N] tx;  
    vector[N] sp;  
}  
  
parameters {  
    real<lower=0.001> b_tx;  
    real<lower=0.001> b_sp;  
  
    real<lower=0.001> sigma_tx;  
    real<lower=0.001> sigma_sp;  
  
    real mu_tx;  
    real mu_sp;  
}  
  
model {  
    b_tx ~ normal(0, 10);  
    b_sp ~ normal(0, 10);  
  
    sigma_tx ~ normal(0, 5);  
    sigma_sp ~ normal(0, 5);  
  
    mu_tx ~ normal(b_tx, sigma_tx);  
    mu_sp ~ normal(b_sp, sigma_tx);  
  
    perc ~ beta(mu_tx, mu_sp);  
}
```

PP_check: stan_betareg()



Next Steps...

1. Continue working on Beta-Binomial Distribution rstan model
2. Integrate both models into one
3. Design to add more species and more predictors
4. Re-run experiment and see!



QUESTIONS?

