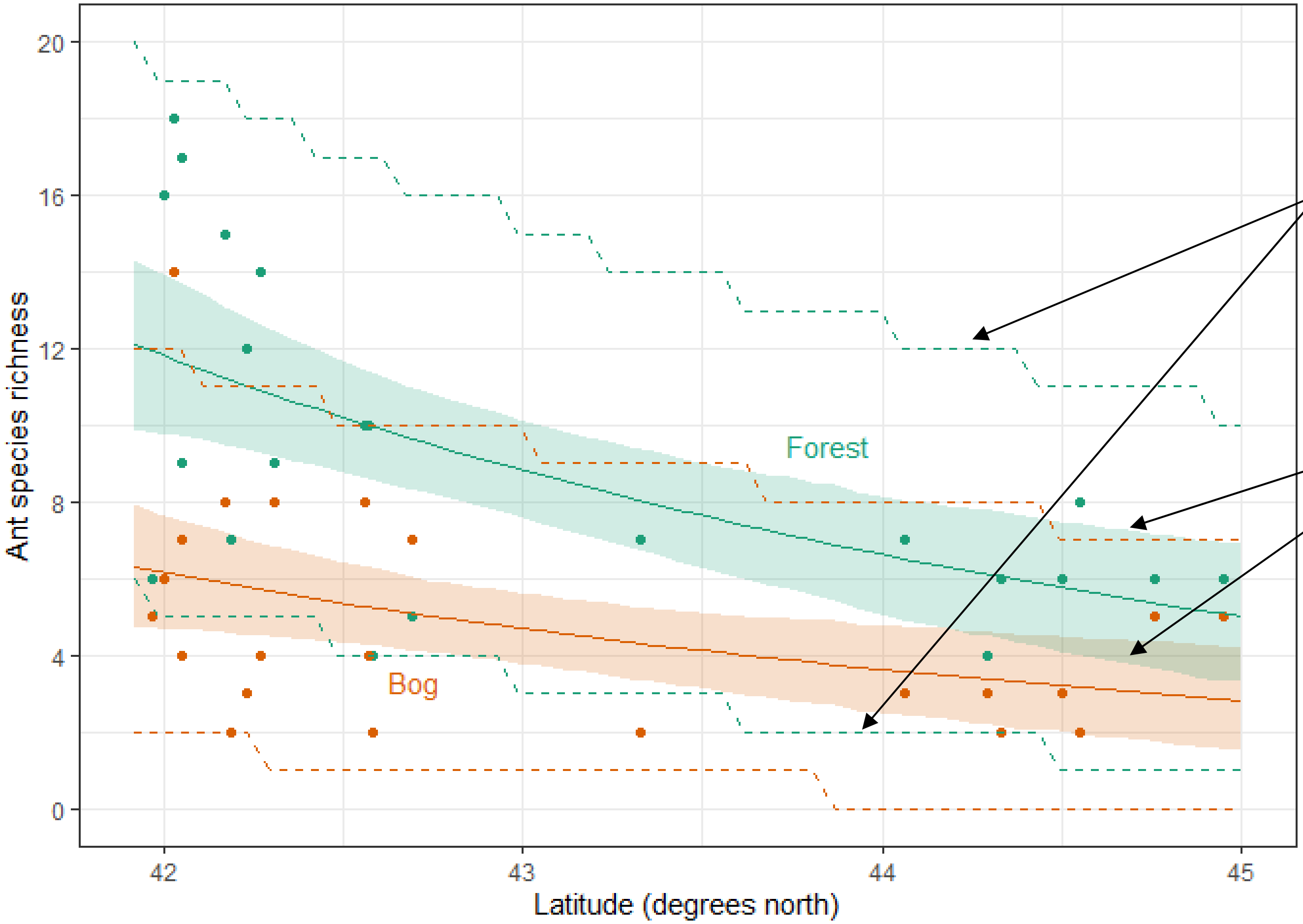
Today

- Ants GLM, Bayesian (rethinking-ulam)
  - working with posterior samples
  - derived quantities
  - Answer to: How different is species richness between habitats?
- Ants GLM, likelihood/frequentist
  - model formulae, glm()
- Ants GLM, Bayesian (rstanarm)
  - glm() becomes stan_glm()

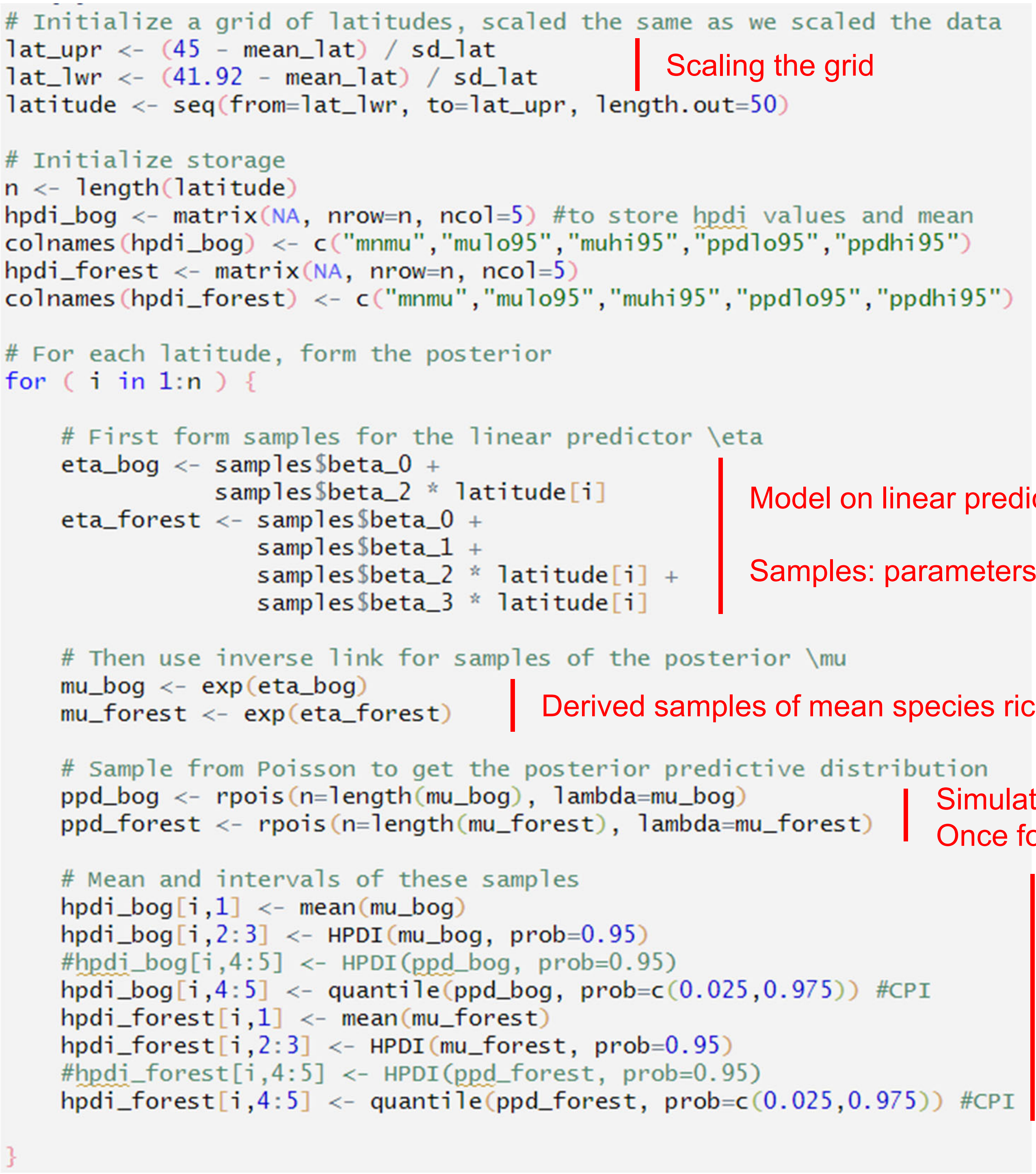
Bayesian model - ants



Derived quantities

Prediction interval

Credible interval



```
# Initialize a grid of latitudes, scaled the same as we scaled the data
lat_upr <- (45 - mean_lat) / sd_lat
lat_lwr <- (41.92 - mean_lat) / sd_lat
latitude <- seq(from=lat_lwr, to=lat_upr, length.out=50)

# Initialize storage
n <- length(latitude)
hpdi_bog <- matrix(NA, nrow=n, ncol=5) #to store hpdi values and mean
colnames(hpdi_bog) <- c("mnmu","mulo95","muhi95","ppdlo95","ppdhi95")
hpdi_forest <- matrix(NA, nrow=n, ncol=5)
colnames(hpdi_forest) <- c("mnmu","mulo95","muhi95","ppdlo95","ppdhi95")

# For each latitude, form the posterior
for ( i in 1:n ) {

    # First form samples for the linear predictor \eta
    eta_bog <- samples$beta_0 +
               samples$beta_2 * latitude[i]
    eta_forest <- samples$beta_0 +
                  samples$beta_1 +
                  samples$beta_2 * latitude[i] +
                  samples$beta_3 * latitude[i]

    # Then use inverse link for samples of the posterior \mu
    mu_bog <- exp(eta_bog)
    mu_forest <- exp(eta_forest)

    # Sample from Poisson to get the posterior predictive distribution
    ppd_bog <- rpois(n=length(mu_bog), lambda=mu_bog)
    ppd_forest <- rpois(n=length(mu_forest), lambda=mu_forest)

    # Mean and intervals of these samples
    hpdi_bog[i,1] <- mean(mu_bog)
    hpdi_bog[i,2:3] <- HPDI(mu_bog, prob=0.95)
    #hpdi_bog[i,4:5] <- HPDI(ppd_bog, prob=0.95)
    hpdi_bog[i,4:5] <- quantile(ppd_bog, prob=c(0.025,0.975)) #CPI
    hpdi_forest[i,1] <- mean(mu_forest)
    hpdi_forest[i,2:3] <- HPDI(mu_forest, prob=0.95)
    #hpdi_forest[i,4:5] <- HPDI(ppd_forest, prob=0.95)
    hpdi_forest[i,4:5] <- quantile(ppd_forest, prob=c(0.025,0.975)) #CPI

}
```

Scaling the grid

Model on linear predictor scale

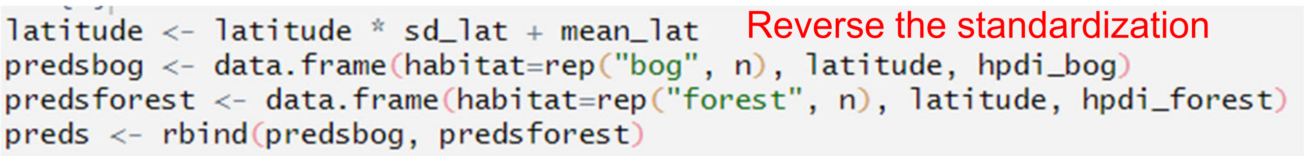
Samples: parameters (vectors 36000)

Derived samples of mean species richness

Simulate data generating process
Once for each sample

Mean and HPDI or CPI
of posterior mu
for one latitude

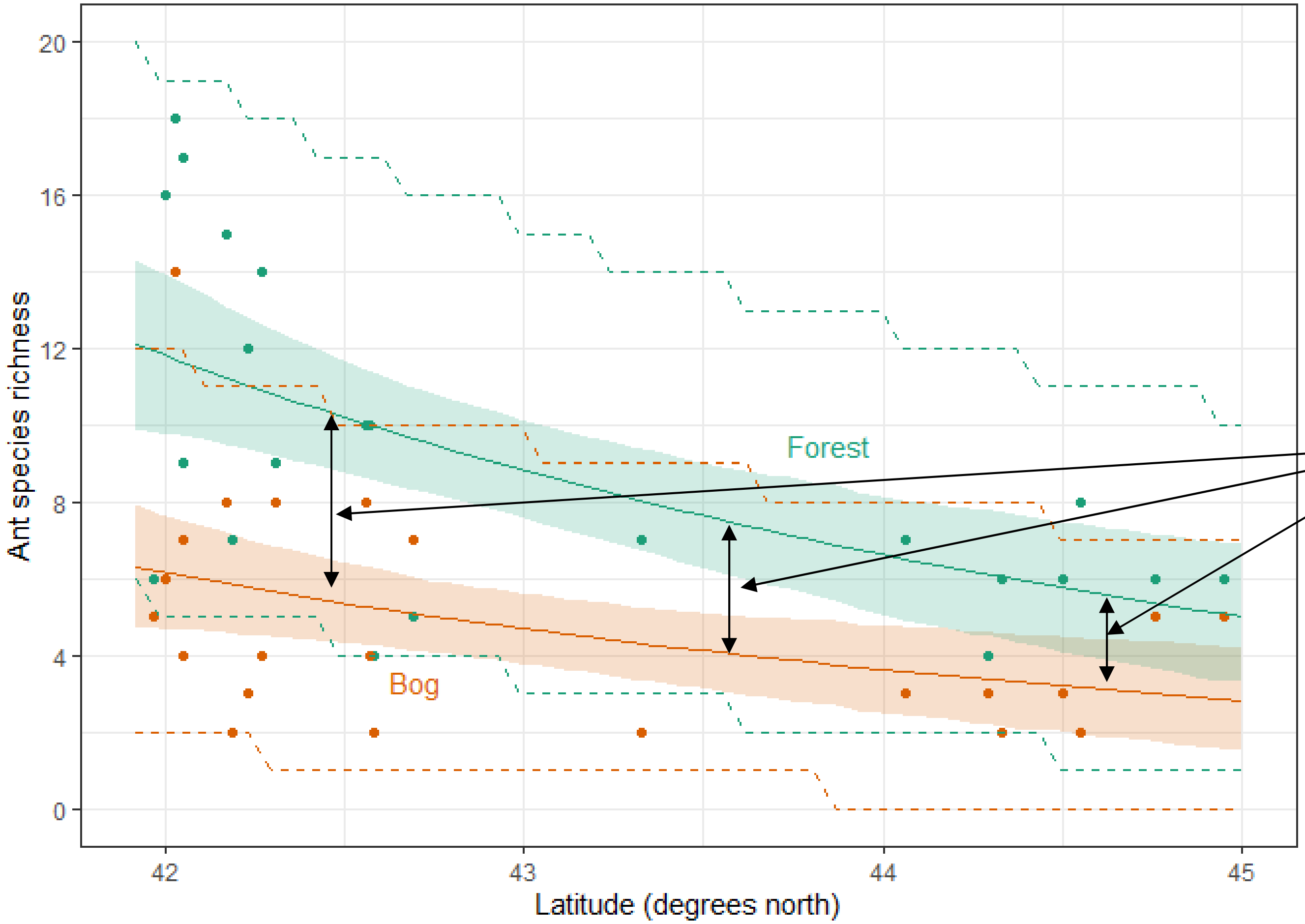
Package in tidy format for plotting



```
latitude <- latitude * sd_lat + mean_lat
predsbog <- data.frame(habitat=rep("bog", n), latitude, hpdi_bog)
predsforest <- data.frame(habitat=rep("forest", n), latitude, hpdi_forest)
preds <- rbind(predsbog, predsforest)
```

Reverse the standardization

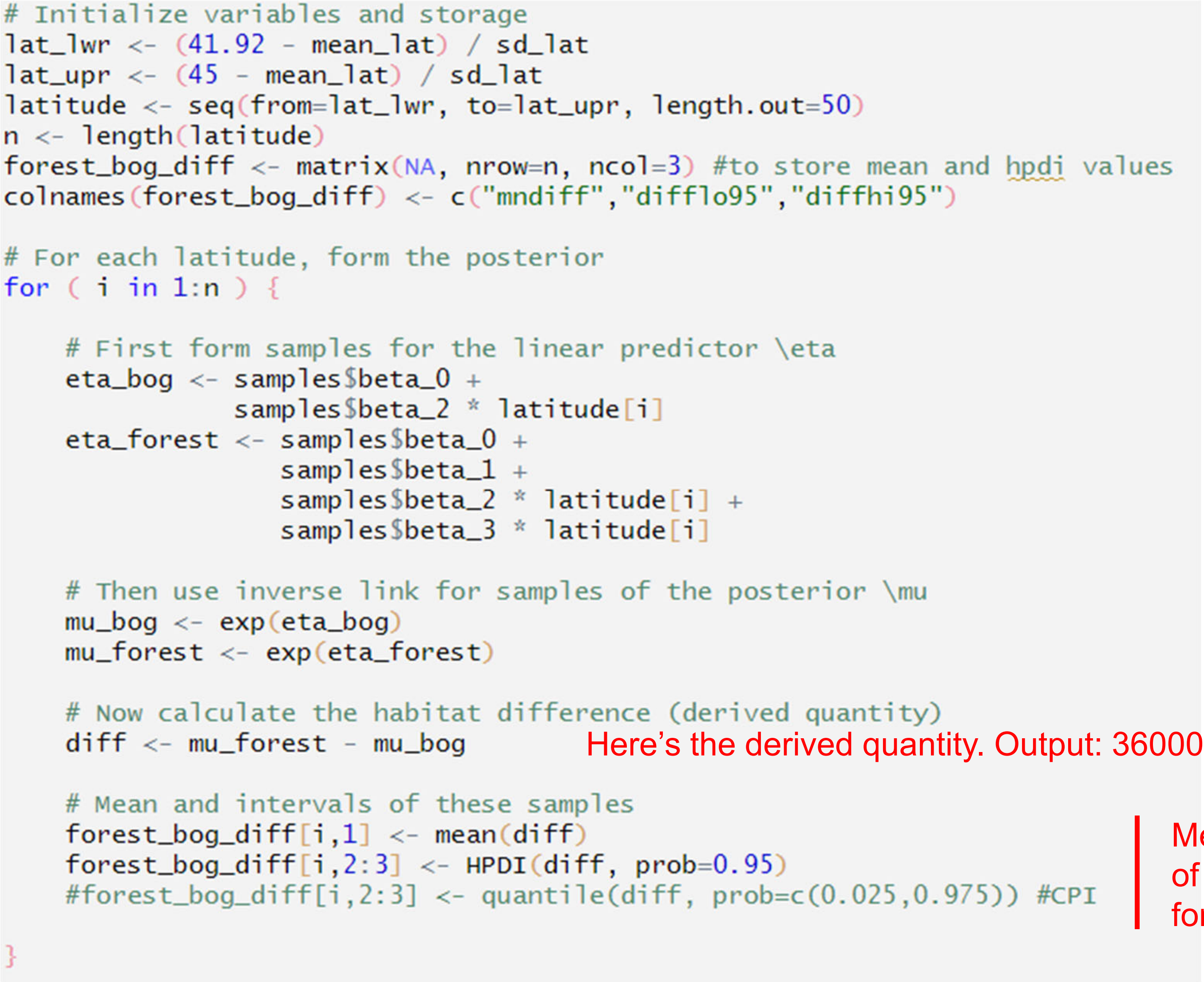
Bayesian model - ants



How different is species richness between habitats?

Derived quantity

Differences at different latitudes

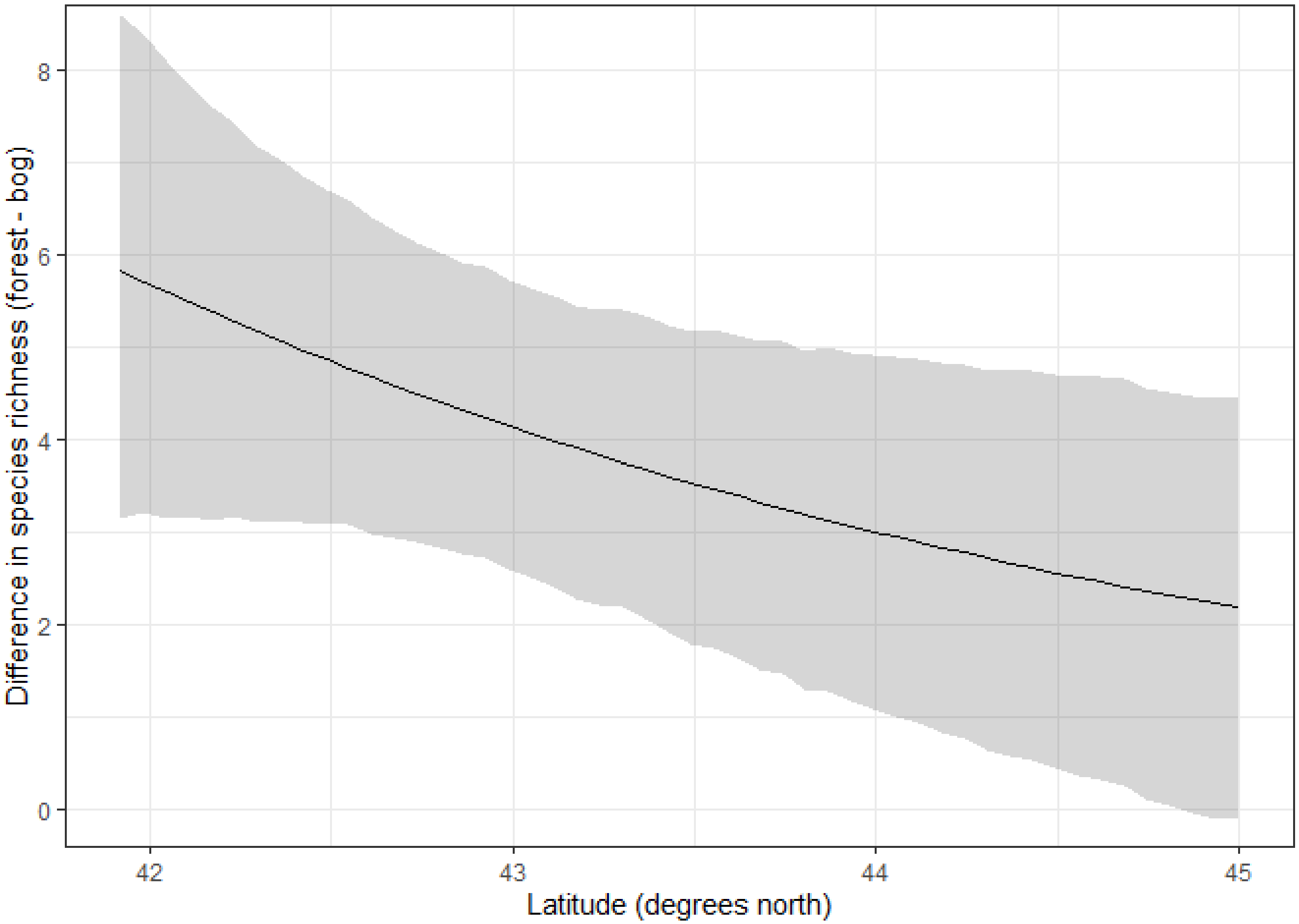


```
# Initialize variables and storage
lat_lwr <- (41.92 - mean_lat) / sd_lat
lat_upr <- (45 - mean_lat) / sd_lat
latitude <- seq(from=lat_lwr, to=lat_upr, length.out=50)
n <- length(latitude)
forest_bog_diff <- matrix(NA, nrow=n, ncol=3) #to store mean and hpdi values
colnames(forest_bog_diff) <- c("mndiff","difflo95","diffhi95")

# For each latitude, form the posterior
for ( i in 1:n ) {

    # First form samples for the linear predictor \eta
    eta_bog <- samples$beta_0 +
               samples$beta_2 * latitude[i]
    eta_forest <- samples$beta_0 +
                  samples$beta_1 +
                  samples$beta_2 * latitude[i] +
                  samples$beta_3 * latitude[i]

    # Then use inverse link for samples of the posterior \mu
    mu_bog <- exp(eta_bog)
    mu_forest <- exp(eta_forest)

    # Now calculate the habitat difference (derived quantity)
    diff <- mu_forest - mu_bog

    # Mean and intervals of these samples
    forest_bog_diff[i,1] <- mean(diff)
    forest_bog_diff[i,2:3] <- HPDI(diff, prob=0.95)
    #forest_bog_diff[i,2:3] <- quantile(diff, prob=c(0.025,0.975)) #CPI

}
```

Here's the derived quantity. Output: 36000 samples of diff

Mean and HPDI or CPI
of posterior diff
for one latitude

Mean and HPDI for the difference in species richness



R-centric regression modeling

Model formulae

```
y ~ x1 * x2 + x3 / x4
```

equivalent to:

```
y ~ 1 + x1 + x2 + x1:x2 + x3 + x3:x4
```

Model formulae

- glm()
  - Base R: stats package
  - likelihood/frequentist

```
glm(richness ~ habitat + latitude + habitat:latitude,
          family=poisson(link="log"), data=ant)
```

- stan_glm()
  - rstanarm package
  - Bayesian

```
stan_glm(richness ~ habitat + latitude + habitat:latitude,
                family=poisson(link="log"), data=ant)
```

Model formula: ants

```
fit <- glm(richness ~ habitat + latitude + habitat:latitude,
           family=poisson(link="log"), data=ant)
```

Model formula: ants

```
fit <- glm(richness ~ habitat + latitude + habitat:latitude,
           family=poisson(link="log"), data=ant)
```

Equivalent:

```
richness ~ habitat * latitude

richness ~ 1 + habitat + latitude + habitat:latitude
```

Model formula: ants

```
fit <- glm(richness ~ habitat + latitude + habitat:latitude,
           family=poisson(link="log"), data=ant)
```

Equivalent:

```
richness ~ habitat * latitude

richness ~ 1 + habitat + latitude + habitat:latitude
```

$$\eta_i = \beta_0 intercept_i + \beta_1 forest_i + \beta_2 latitude_i + \beta_3 forest_i \times latitude_i$$

Model matrix (design matrix)

```
fit <- glm(richness ~ habitat + latitude + habitat:latitude,
           family=poisson(link="log"), data=ant)
```

Data

habitat	latitude	richness
forest	42	16
forest	42.56	10
forest	43.33	7
forest	44.76	6
bog	42.17	8
bog	42.57	4
bog	44.06	3
bog	44.95	5

`model.matrix(fit)`

Model matrix

intercept	forest	latitude	forest:latitude
1	1	42	42
1	1	42.56	42.56
1	1	43.33	43.33
1	1	44.76	44.76
1	0	42.17	0
1	0	42.57	0
1	0	44.06	0
1	0	44.95	0

$$\eta_i = \beta_0 \mathit{intercept}_i + \beta_1 \mathit{forest}_i + \beta_2 \mathit{latitude}_i + \beta_3 \mathit{forest}_i \times \mathit{latitude}_i$$

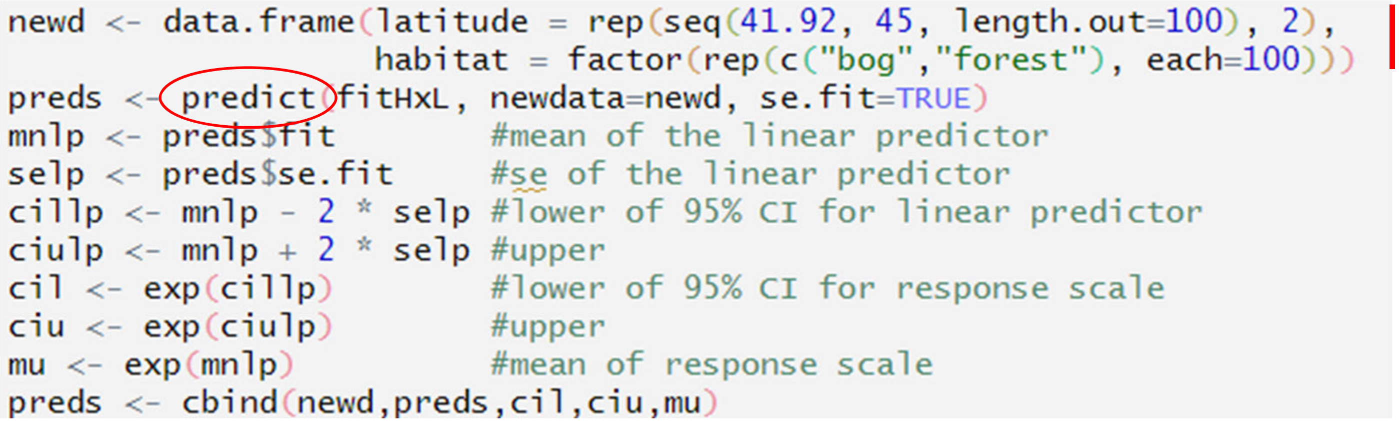
```
1       + forest   + latitude + forest:latitude
```

glm(): stats (base R)

- Training algorithm (likelihood)
  - specialized for MLE of exponential family
  - optimizer: iterative weighted least squares
- Model checking
  - plot(), same as lm()
- Inference algorithms (frequentist)
  - confint(): likelihood profiles for parameters
    - sampling distribution of likelihood ratio: chi square
  - predict(): curves & conf intervals (approx)

glm(): stats (base R)

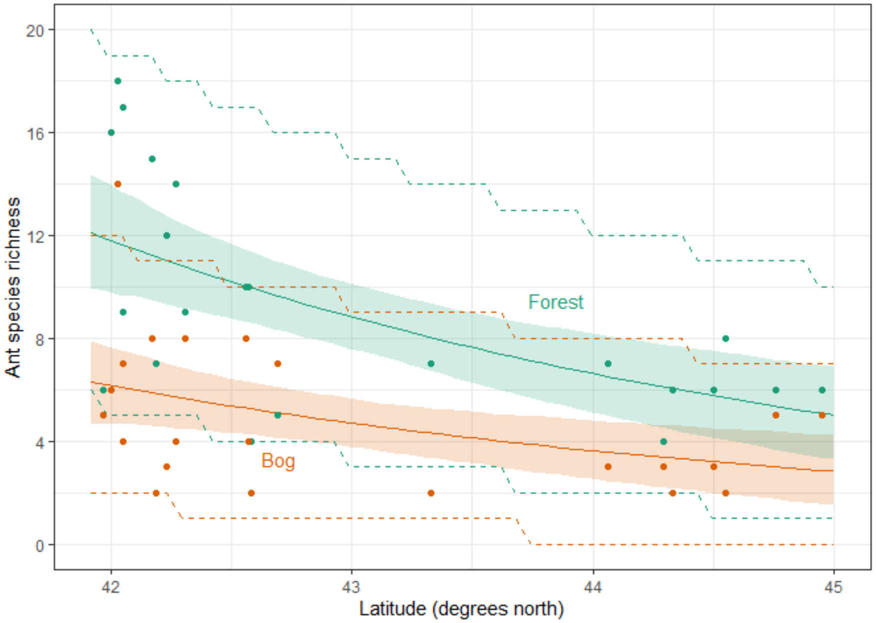
Curves & approximate confidence intervals



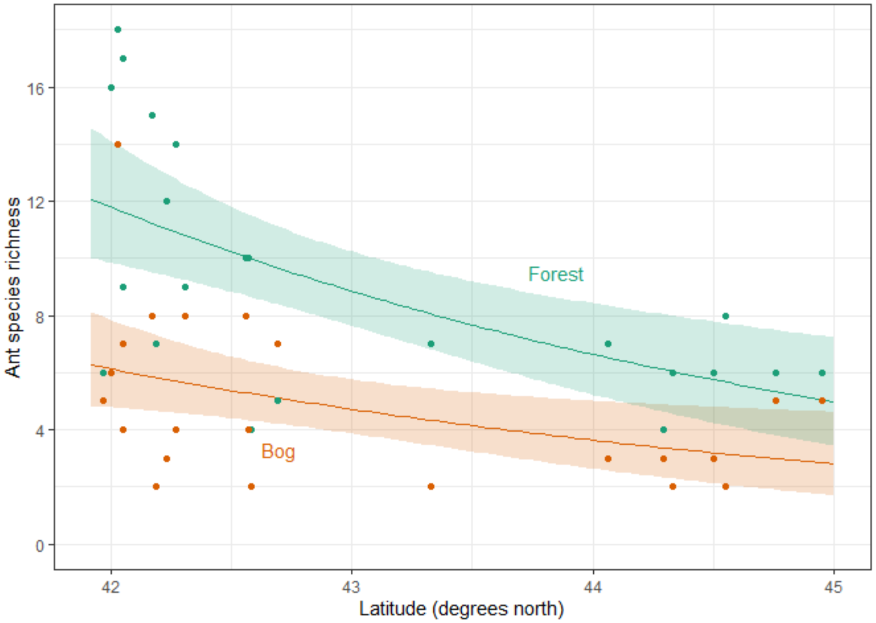
```
newd <- data.frame(latitude = rep(seq(41.92, 45, length.out=100), 2),
                   habitat = factor(rep(c("bog","forest"), each=100)))
preds <- predict(fitHxL, newdata=newd, se.fit=TRUE)
mnlp <- preds$fit        #mean of the linear predictor
selp <- preds$se.fit     #se of the linear predictor
cillp <- mnlp - 2 * selp #lower of 95% CI for linear predictor
ciulp <- mnlp + 2 * selp #upper
cil <- exp(cillp)        #lower of 95% CI for response scale
ciu <- exp(ciulp)        #upper
mu <- exp(mnlp)          #mean of response scale
preds <- cbind(newd,preds,cil,ciu,mu)
```

New grid of predictor values

Better confidence intervals: parametric bootstrap
Prediction intervals: parametric bootstrap
More complex derived quantities: parametric bootstrap



Bayesian (ulam)



Frequentist
(w approx intervals)

stan_glm(): rstanarm

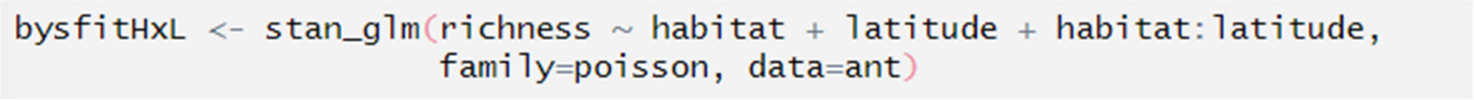
- Priors: good defaults, see next script
- Training algorithm (Bayesian)
  - HMC, same as ulam
- Samples
  - `samples <- as.data.frame(fit)`
- Visualization
  - `plot(fit, “type”)`, incl “hist”, “trace”
  - bayesplot package: mcmc_<type>()
    - use for more generality

stan_glm(): rstanarm

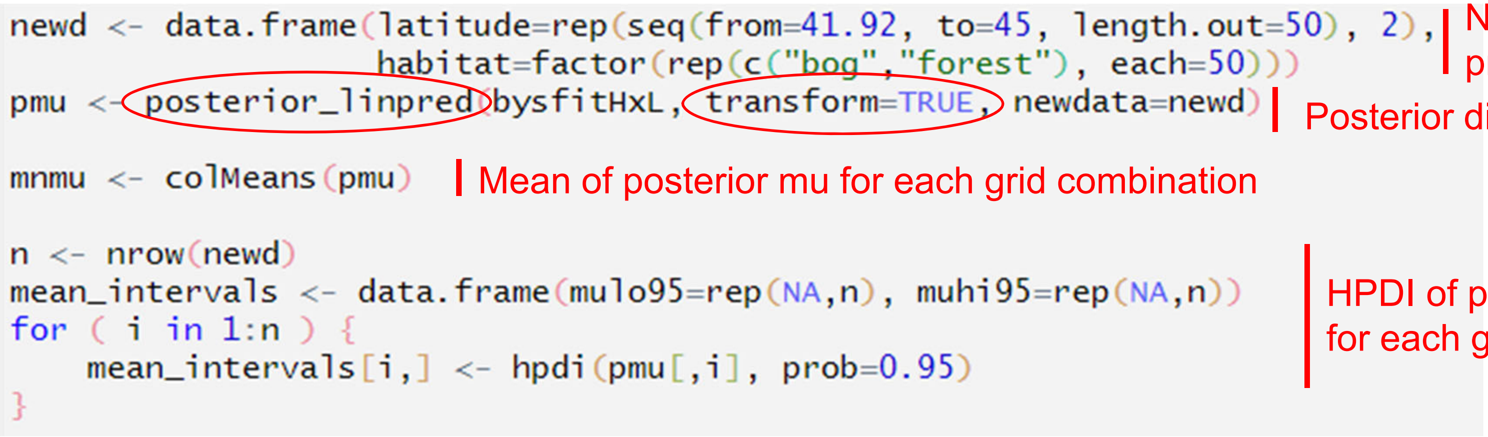
- Inference algorithms (posterior samples)
  - defaults
    - 4000 samples (low)
    - 90% intervals, CPI only
  - convenience functions:
  - posterior_interval(): parameter CPIs
  - posterior_linpred(): derived samples of mu
  - posterior_predict(): samples from DGP
  - predictive_intervals(): prediction CPI

stan_glm(): rstanarm

Regression curve mean and interval



```
bysfitHxL <- stan_glm(richness ~ habitat + latitude + habitat:latitude,
                      family=poisson, data=ant)
```



```
newd <- data.frame(latitude=rep(seq(from=41.92, to=45, length.out=50), 2),
                   habitat=factor(rep(c("bog","forest"), each=50)))
pmu <- posterior_linpred(bysfitHxL, transform=TRUE, newdata=newd)

mnmu <- colMeans(pmu)

n <- nrow(newd)
mean_intervals <- data.frame(mulo95=rep(NA,n), muhi95=rep(NA,n))
for ( i in 1:n ) {
    mean_intervals[i,] <- hpdi(pmu[,i], prob=0.95)
}
```

New grid of predictor values

Posterior distribution of mu

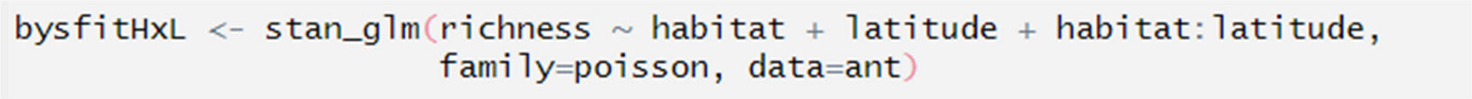
Mean of posterior mu for each grid combination

HPDI of posterior mu for each grid combination

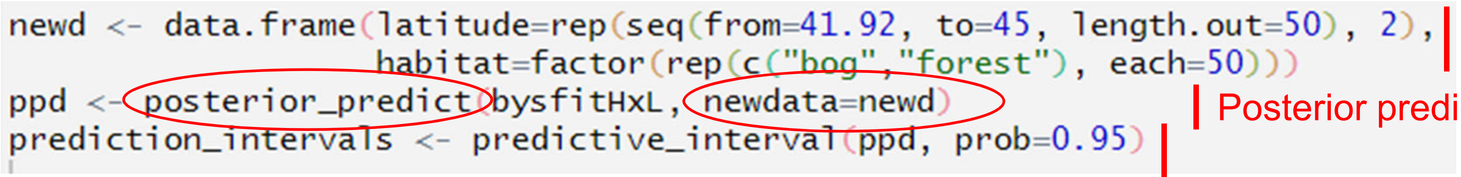
Opinionated: no convenient function for doing this. They want us to focus on the predictive distribution (my counterargument: many science questions are about estimation not prediction).

stan_glm(): rstanarm

Prediction interval



```
bysfitHxL <- stan_glm(richness ~ habitat + latitude + habitat:latitude,
                      family=poisson, data=ant)
```

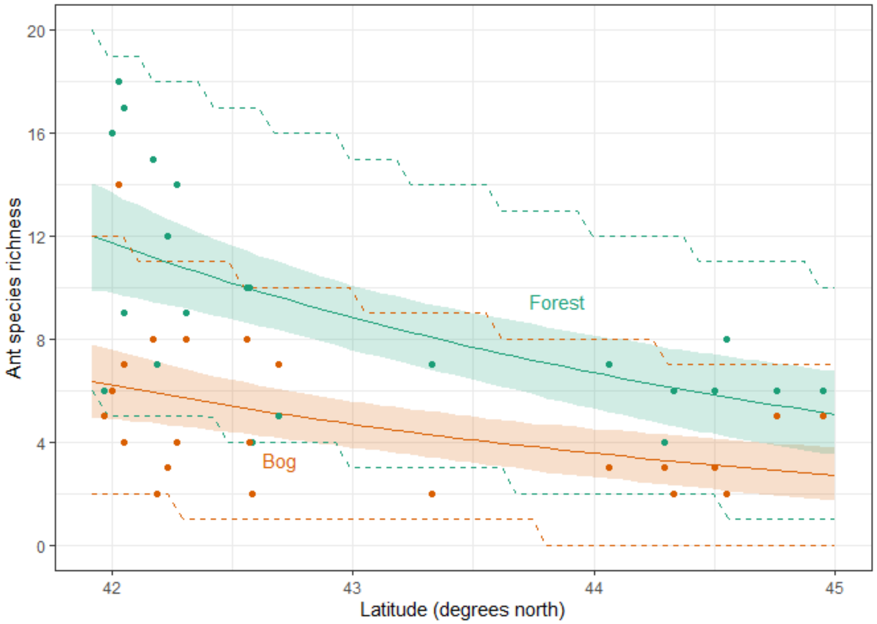


```
newd <- data.frame(latitude=rep(seq(from=41.92, to=45, length.out=50), 2),
                   habitat=factor(rep(c("bog","forest"), each=50)))
ppd <- posterior_predict(bysfitHxL, newdata=newd)
prediction_intervals <- predictive_interval(ppd, prob=0.95)
```

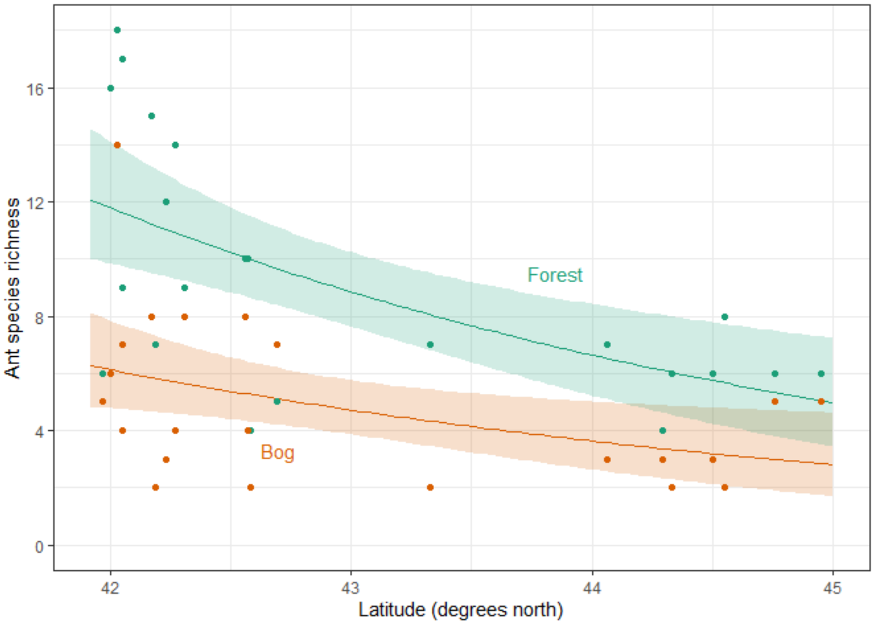
New grid of
predictor values

Posterior predictive distribution

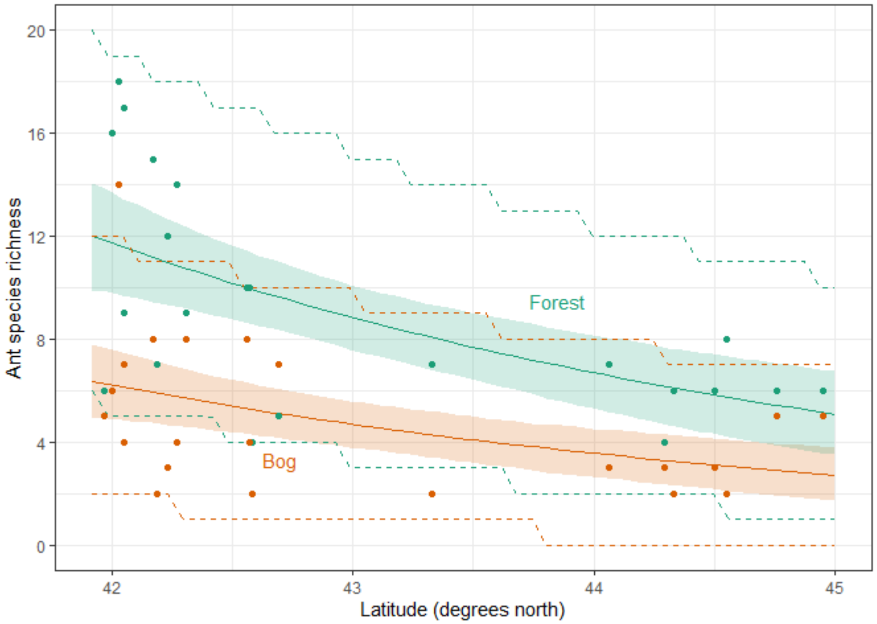
CPI, set it to 0.95
beware default is 0.9



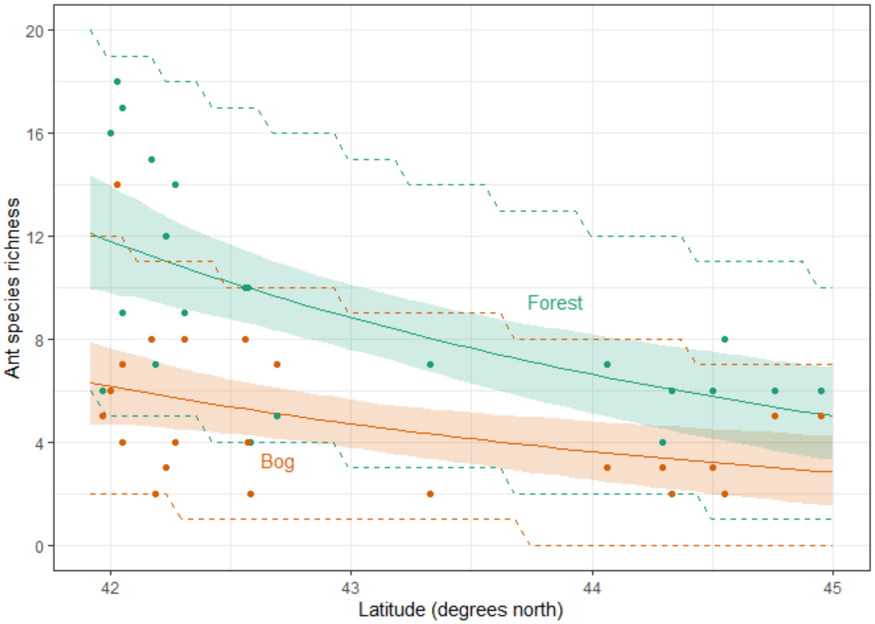
Bayesian (stan_glm)



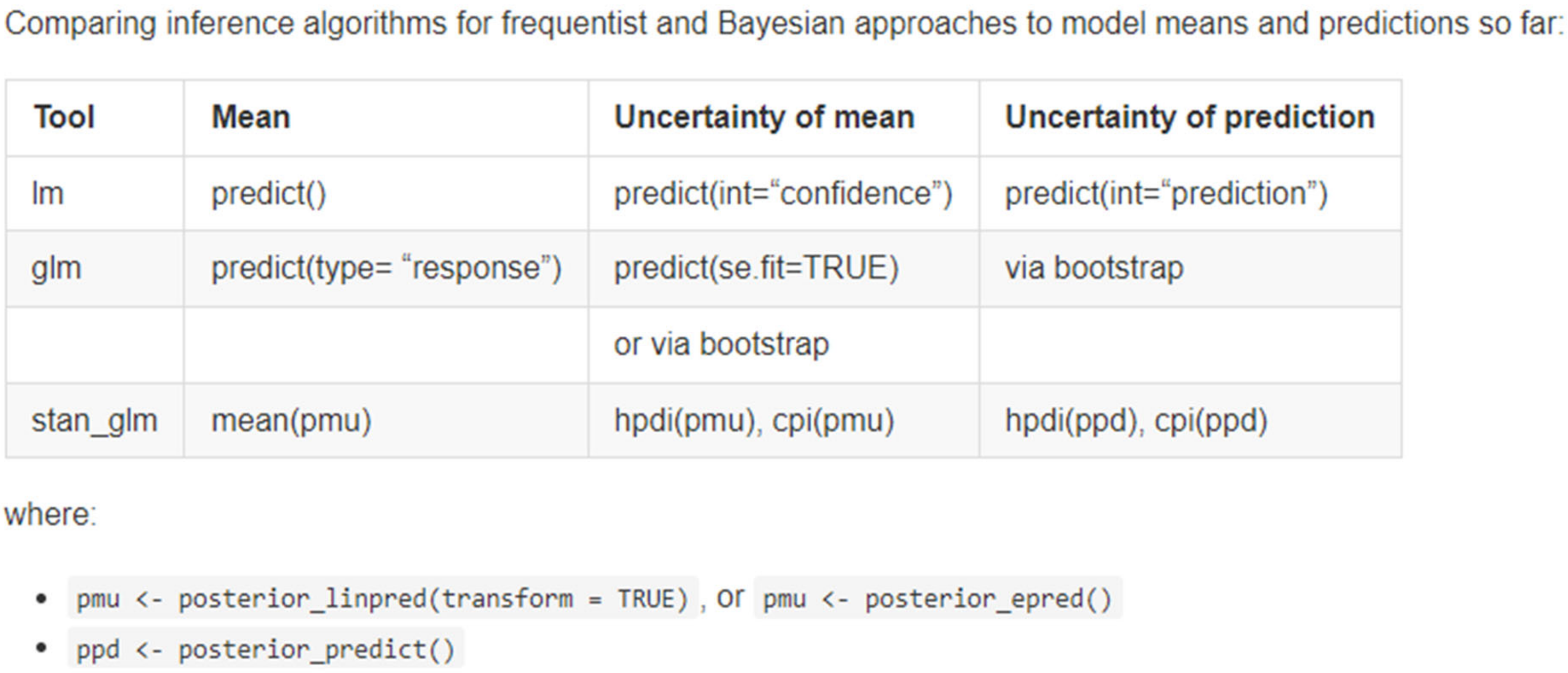
Frequentist (glm)
(w approx intervals)



Bayesian (stan_glm)



Bayesian (ulam)



Comparing inference algorithms for frequentist and Bayesian approaches to model means and predictions so far:

Tool	Mean	Uncertainty of mean	Uncertainty of prediction
lm	predict()	predict(int=“confidence”)	predict(int=“prediction”)
glm	predict(type= “response”)	predict(se.fit=TRUE)	via bootstrap
		or via bootstrap	
stan_glm	mean(pmu)	hpdi(pmu), cpi(pmu)	hpdi(ppd), cpi(ppd)

where:

- `pmu <- posterior_linpred(transform = TRUE)`, or `pmu <- posterior_epred()`
- `ppd <- posterior_predict()`