Full workflow:
Trophic asynchrony and climate change

Steps 1-4 in a Bayesian framework
From: A four-step simulation-based workflow for ecology

An example of the full workflow

Our simple example comes from a project aiming to estimate levels of trophic (a)synchrony with climate change. Using observational time-series data, we wanted to estimate changes in the relative timing of species pairs. Our first step was to obtain estimates of each species' change over time, which we review here.

How we started

We thought we should fit a mixed-effects linear regression, with day of year (calendar day, values of 1-366) of event (leafout, reproduction etc.) as the response and year as the predictor. The data had uneven sampling across species (different time series lengths), as you can see:

```
## housekeeping
rm(list=ls())
options(stringsAsFactors = FALSE)
set.seed(3777)

d <- read.csv("output/rawlong.tot2.csv")
table(d$species)
```

```
##
##         Acartia hudsonica               Acartia tonsa            Accipiter nisus
##                         8                           6                         16
## Acrocephalus arundinaceus   Acrocephalus scirpaceus                 Alca torda
##                        29                          29                         24
##         Ammodytes marinus          Asterionella spp.            Beroe gracilis
##                        24                          26                         27
##              Bombus spp.     Castilleja tenuiflora       Caterpillar_a spp.
##                        14                          19                         36
##       Caterpillar_b spp.        Caterpillar_c spp.        Caterpillar2 spp.
##                        34                          34                         17
##      Caterpillar2a spp.       Caterpillar4a spp.       Caterpillar4b spp.
##                        20                          16                         14
##     Cerorhinca monocerata      Clamator glandarius             Copepod1 spp.
##                        11                           9                         30
##             Copepod2 spp.        Corydalis ambigua           Cyclops vicinus
##                        27                          14                         14
##   Daphnia hyalina-galeata        Daphnia pulicaria             Daphnia1 spp.
##                        26                          25                         25
##             Daphnia2 spp.            Daphnia3 spp.   Delphinium nuttallianum
##                        27                          37                         30
##              Diatom1 spp.           Diatom2a spp.           Diatom2b spp.
##                        25                        25                        16
##              Diatom3 spp.             Diatom4a spp.             Diatom4b spp.
##                        30                        27                        32
##             Diatom4c spp.       Engraulis japonicus        Epirrita autumnata
##                        37                        11                        14
##  Erythronium grandiflorum        Ficedula hypoleuca      Ficedula_a hypoleuca
##                        28                        19                        16
##      Ficedula1 albicollis      Ficedula2 albicollis        Fratercula arctica
##                        34                        26                        24
##                 Glis glis           Guillemots spp.     Keratella2 cochlearis
##                        26                        23                        32
##   Leptodiaptomus ashlandi        Mnemiopsis1 leidyi        Mnemiopsis2 leidyi
##                        37                         6                         8
##       Operophtera brumata                Parus ater           Parus caeruleus
##                        32                        20                        26
##              Parus_a ater              Parus1 major          Parus2 caeruleus
##                        16                        36                        20
##              Parus2 major          Parus3 caeruleus              Parus3 major
##                        32                        14                        26
##             Parus4a major             Parus4b major              Parus6 major
##                        20                        16                        16
##        Perca fluviatillis Phalacrocorax aristotelis       Phytoplankton1 spp.
##                        36                        24                        37
##       Phytoplankton2 spp.                 Pica pica               Plant2 spp.
##                        14                         9                        10
##               Plant3 spp.      Pleurobrachia pileus    Pleurobrachia_a pileus
##                        19                        27                        27
##          Poecile montanus        Pygoscelis adeliae    Pygoscelis antarcticus
##                        14                        17                        10
##          Pygoscelis papua              Quercus spp.            Quercus2 robur
##                        17                         7                        34
##            Quercus3 robur        Rangifer2 tarandus         Rissa1 tridactyla
##                        17                        10                        23
##         Rissa2 tridactyla   Selasphorus platycercus            Sitta europaea
##                        24                        33                        26
##              Syrphid spp. Thermocyclops oithonoides          Tortrix viridana
##                        19                        16                         7
##                Uria aalge
##                        24
```

Because of this uneven sampling of species, we thought we should set species as a 'random effect' (or a grouping factor with partial pooling). Given our interest in obtaining estimates of each species' change over time, we wanted to have species as a random effect on both the intercept and slope. However, when we tried this in the R package lme4, we found the model did not converge:

```
## libraries
library(lme4)
```

```
## Loading required package: Matrix
```

```
modelwanted <- lmer(phenovalue~(year|species), data=d)
```

```
## boundary (singular) fit: see help('isSingular')
```

```
summary(modelwanted)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: phenovalue ~ (year | species)
##    Data: d
##
## REML criterion at convergence: 15887.9
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -6.3640 -0.4231 -0.0138  0.4292  5.1449
##
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  species  (Intercept) 0.000e+00  0.00000
##           year        5.432e-04  0.02331  NaN
##  Residual             1.650e+02 12.84698
## Number of obs: 1939, groups:  species, 88
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  125.173      4.945   25.31
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

We considered simplifying the model to get it to run. We could get the same model with species as a random effect on only the intercept to run:

```
modelconverged <- lmer(phenovalue~year+(1|species), data=d)
```

But we knew this was not right: our understanding of climate change impacts suggested it was highly unlikely that all species would have a common change over time. So we tried a Bayesian approach to ideally fit separate slopes for each species over time, but drawn from a common distribution (which is what the term 'random effect' generally refers to in ecology) and started thinking about our model.

A four-step workflow Bayesian approach

Step 1: Develop your model

We realized our verbal model did not agree with the statistical model we expected to fit. We planned to fit a simple linear model, but that would assume climate change has been ongoing across all our years and that's not what most science on anthropogenic warming suggests. Instead science (including the IPCC etc.) suggests that a large uptick in warming started around 1980. Thus we developed a 'hinge' model to fit the linear regression after 1980 and a mean before 1980 (here we did this by subtracting 1980 from the predictor data).

This highlights an important reality throughout the workflow: effective model building is about efficient brainstorming. It's a constant back and forth between asking questions about what we know and what we should know.

At the end of this step we have our conceptual model converted into its math and coded with priors in Stan:

```
## libraries
library(truncnorm)
library(rstan)
```

```
## Loading required package: StanHeaders
```

```
##
## rstan version 2.36.0.9000 (Stan version 2.36.0)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
## For within-chain threading using `reduce_sum()` or `map_rect()` Stan functions,
## change `threads_per_chain` option:
## rstan_options(threads_per_chain = 1)
```

```r
options(mc.cores = parallel::detectCores())
```

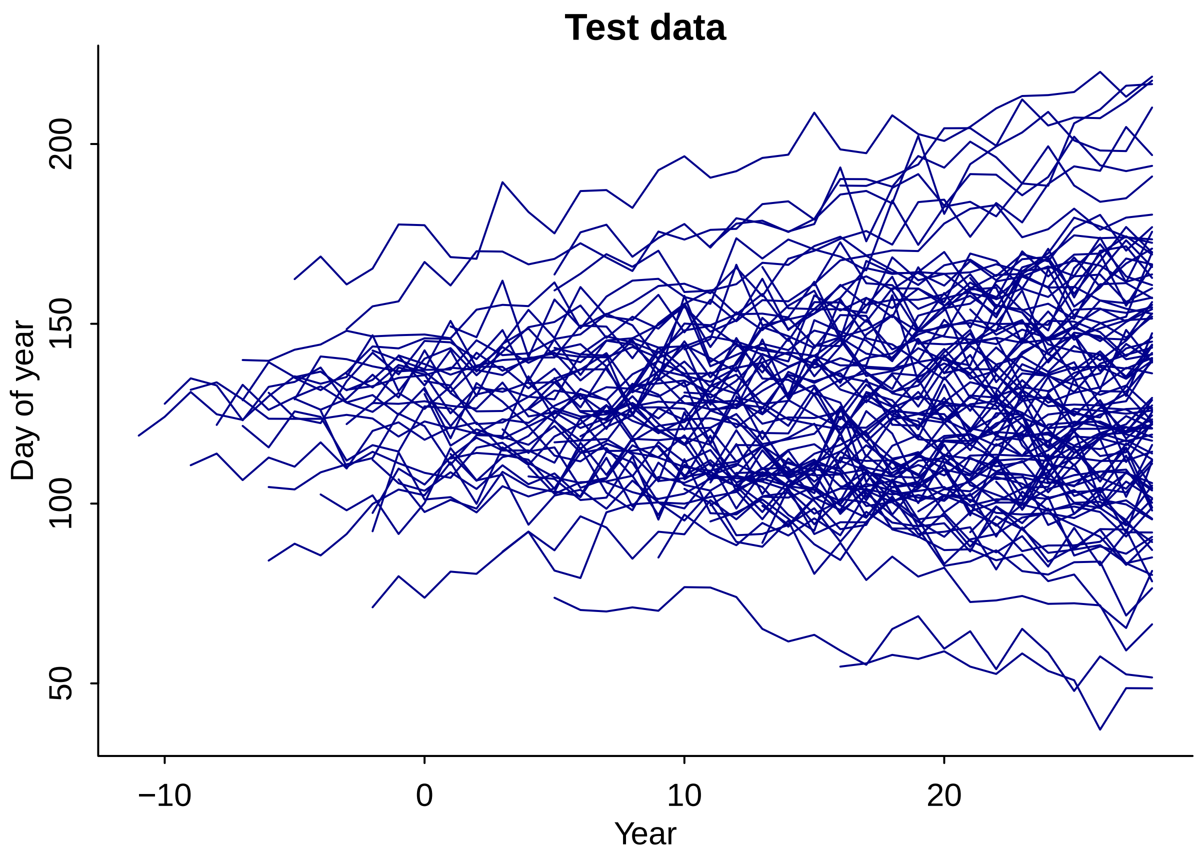
```stan
// Two-level (1 hierarchical grouping) `random' slope and intercept model
// Partial pooling on intercepts and slopes

data{
int<lower=0> N;     // number of total observations
int<lower=0> Nspp;  // number of species (grouping factor)
array[N] int species;   // species identity, coded as int
vector[N] year;     // year of data point (predictor for slope)
array[N] real y;        // day of year of phenological event (response)
}

parameters{
real mu_a;      // mean intercept across species
real<lower=0> sigma_a;  // variation of intercept across species
real mu_b;      // mean slope across species
real<lower=0> sigma_b;  // variation of slope across species
real<lower=0> sigma_y;  // measurement error, noise etc.
vector[Nspp] a;         //the intercept for each species
vector[Nspp] b;         //the slope for each species

}

transformed parameters{
array[N] real ypred;
for (i in 1:N){
    ypred[i]=a[species[i]]+b[species[i]]*year[i];
}
}

model{
b ~ normal(mu_b, sigma_b); // this creates the partial pooling on slopes
a ~ normal(mu_a, sigma_a); // this creates the partial pooling on intercepts
y ~ normal(ypred, sigma_y); // this creates an error model where error is normally distributed
// Priors ...
mu_a ~ normal(100,30);
sigma_a ~ normal(0,20);
mu_b ~ normal(0,5);
sigma_b ~ normal(0,15);
sigma_y ~ normal(0,15);
}
```

Step 2: Check your model on simulated data

Next we simulated data to test our model code. To do this we set the parameters in our model first, then we simulated the data from these set parameters. In simulation, we know the 'truth'—which here is a our model parameters—and we can then compare what was estimated to what we started with.

First we set up the simulated test data and plot it.

```r
## libraries
library(truncnorm)
library(rstan)
options(mc.cores = parallel::detectCores())

# Create the species-level parameters
Nspp <- 100
mu_doy <- 125
sigma_doy <- 20
mu_shift <- 0.5
sigma_shift <- 1
species_doy <- rnorm(Nspp, mu_doy, sigma_doy)
species_trend <- rnorm(Nspp, mu_shift, sigma_shift)

# Create the overall `error'
sigma_y <- 5

# Keep the parameters together to compare to model output
paramsgiven <- c(mu_doy, mu_shift, sigma_shift, sigma_doy, sigma_y)

# Create the data
year_0 <- 1980
n_data_per_species <- round(runif(Nspp, 5, 40))
species <- rep(1:Nspp, n_data_per_species)
N <- length(species)
year <- rep(NA, N)

for (sp in 1:Nspp){
  year[species==sp] <- rev(2009 - 1:(n_data_per_species[sp])) - year_0
}

ypred <- length(N)

for (n in 1:N){
  s <- species[n]
  ypred[n] <- species_doy[s] + species_trend[s]*year[n]
}

y <- rnorm(N, ypred, sigma_y)

# Plot the data
par(mar=c(3,3,1,1), mgp=c(1.5,.5,0), tck=-.01)
plot(range(year), range(y), type="n", xlab="Year", ylab="Day of year",
     bty="l", main="Test data")
for (sp in 1:Nspp)
  lines(year[species==sp], y[species==sp], col="darkblue")
```

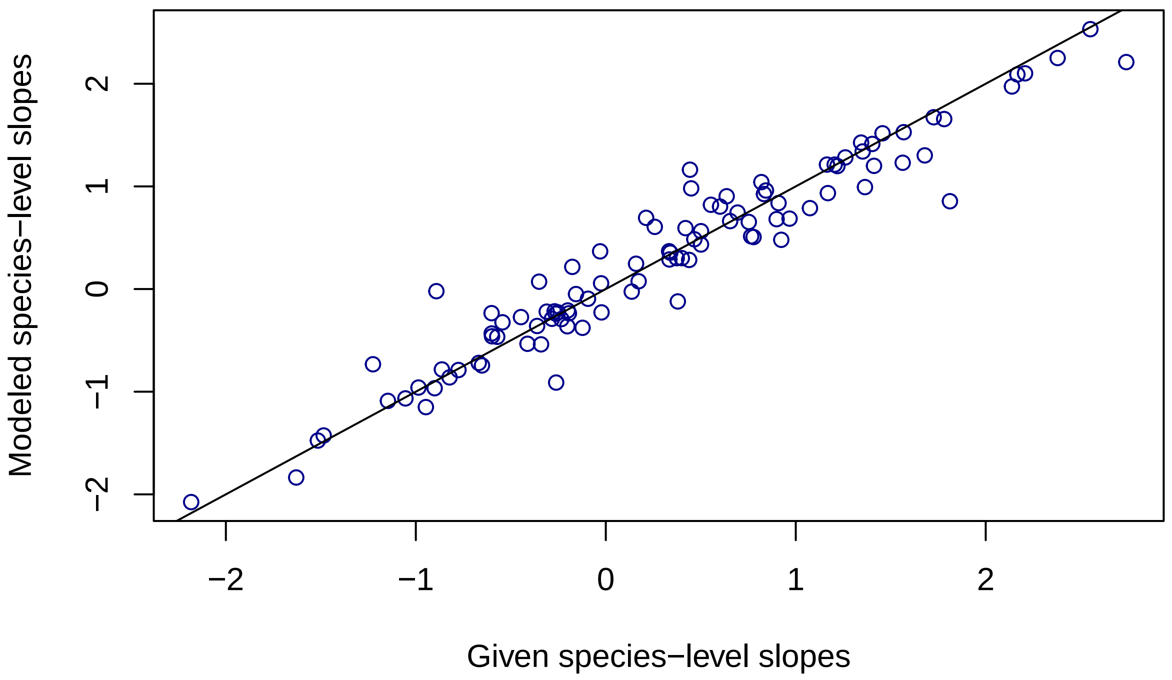


Now that we have a simulated dataset and code for the underlying model in Stan, we can run the model on the simulated data to check how well the model returns the parameters we set.

```
fit <- stan("stan/twolevelhierslopeint.stan", data=c("N","y","Nspp","species","year"), iter=1000, chains

# grep stan output
sumer <- summary(fit)$summary
muparams <- sumer[grep("mu", rownames(sumer)), c("mean", "2.5%", "25%", "50%", "75%", "97.5%")]
sigmaparams <- sumer[grep("sigma", rownames(sumer)), c("mean", "2.5%","25%", "50%", "75%", "97.5%")]

# compare given versus modeled
paramsgiven # here's the parameters we set
muparams # estimated mu parameters
sigmaparams # estimate sigma parameters
```

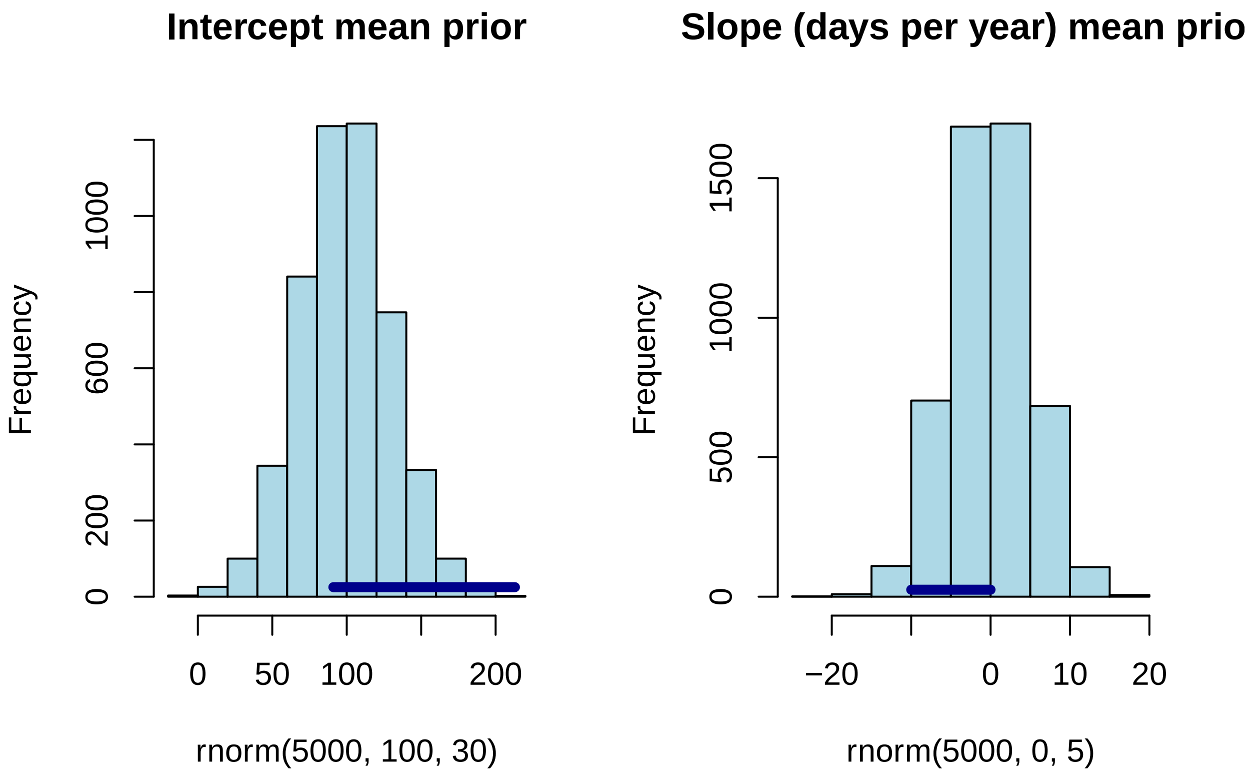
We can also look at the species-level estimates. Since there are many we will plot them:

```
spslopes <- sumer[grep("b\\[", rownames(sumer)), "mean"]

plot(spslopes~species_trend, xlab="Given species-level slopes", ylab="Modeled species-level slopes", co
abline(0,1)
```



Next we want to consider our priors. We can start by simply plotting some of them. Below are simple histograms of the mean priors we set above for the mean intercept and slope.

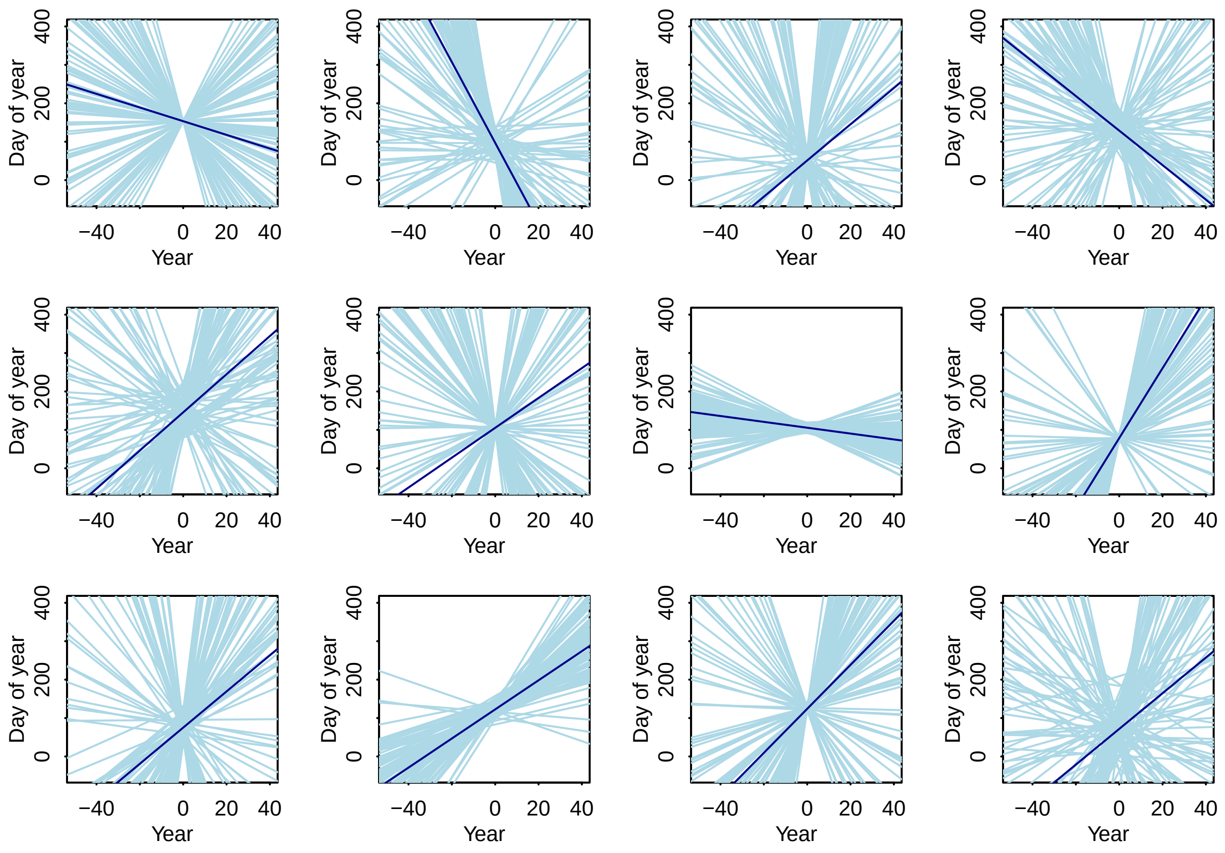
```r
par(mfrow=c(1,2))
hist(rnorm(5000, 100,30), main="Intercept mean prior", col="lightblue")
segments(91,25,213,25, lwd=5, col="darkblue") # April 1 to August 1
hist(rnorm(5000, 0, 5), main="Slope (days per year) mean prior", col="lightblue")
segments(-10,25,0,25, lwd=5, col="darkblue")
```



Are there reasonable? It depends on the system. In this case we know that most phenological events for data collected so far happen in the spring and summer in the northern hemisphere. We show 1 April to 1 August as darker horizontal lines on the intercept plot, and see that it captures this (and this is before the variance parameter for the intercept). Similarly previous studies on trends over time in phenological change (slope) find advances of 2-5 days per year, but sometimes higher; here we show a darker horizontal line for advances of 0 to 10 days is well within the prior and, because we are not sure directionaly for sure, we also equally allow positive numbers (though, as some point when there is enough evidence, shifting the prior more towards advances could make sense).

Here we show one way to visulize what types of model output the priors allow, looking at 12 plots where we drew different possible hyperparameters for the intercept and slope from the priors:

```r
# Let's check what the predicted slopes look like
# Iterating over mu and sigma for intercepts and slopes
reps <- 12
mu_doy <- rnorm(reps, 100,30)
sigma_doy <- rtruncnorm(a=0, b=Inf, reps, 0, 20)
mu_shift <- rnorm(reps, 0,5)
sigma_shift <- rtruncnorm(a=0, b=Inf, reps, 0,15)

par(mfrow=c(3,4))
par(mar=c(3,3,1,1), mgp=c(1.5,.5,0), tck=-.01)
for(i in 1:reps){
    plot(range(year), range(y), xlab="Year", ylab="Day of year",
        xlim=c(-50,40),ylim=c(-50,400), type="n")
    species_doy <- rnorm(Nspp, mu_doy[i], sigma_doy[i])
    species_trend <- rnorm(Nspp, mu_shift[i], sigma_shift[i])
    for(sp in 1:Nspp){
        abline(species_doy[sp], species_trend[sp], col="lightblue")
    }
    abline(mu_doy[i], mu_shift[i], col="darkblue")
}
```



These plots highlight the extreme variability in outcomes our model would allow. This might be far more than is realistic given what we understand currently about climate change and how species respond, but in this case we will leave the priors as they are.

Step 3: Run your model on your empirical data

Next we run the model on our emprical data.

```r
# Formatting for R stan
N <- nrow(d)
y <- d$phenovalue
Nspp <- length(unique(d$species)) #newid is character !
species <- as.numeric(as.factor(d$species))
year <- d$yr1981
syncmodelhis <- stan("stan/twolevelhierslopeint.stan", data=c("N","Nspp","y","species","year"),
                    iter=4000, warmup=3000, chains=4, cores=4, seed=777)
```

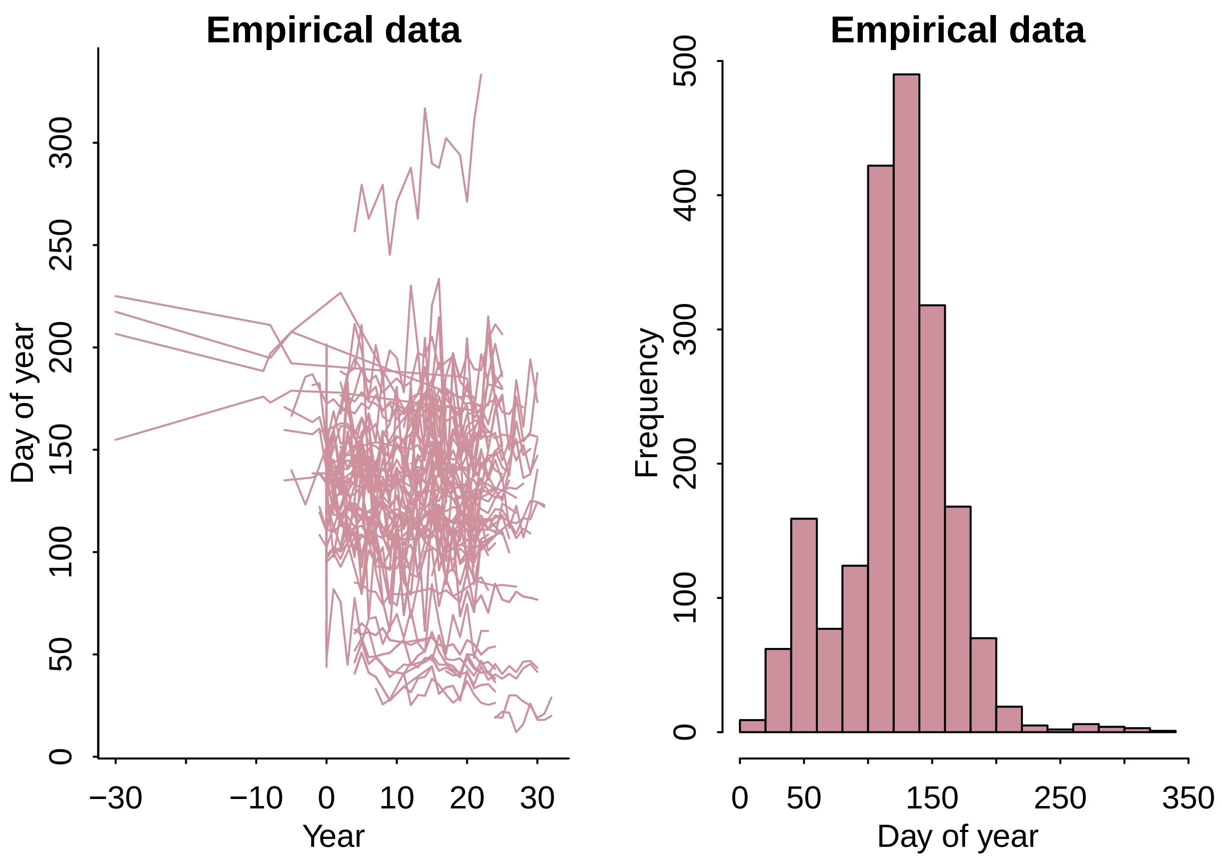
Step 4: Check your model on data simulated from your empirical model output (also known as posterior retrodictive checks)

Now that we have estimates from our model using our empirical data we can compare the world our model has found to what our empirical data look like. There are many ways to do this and we show only a few here.

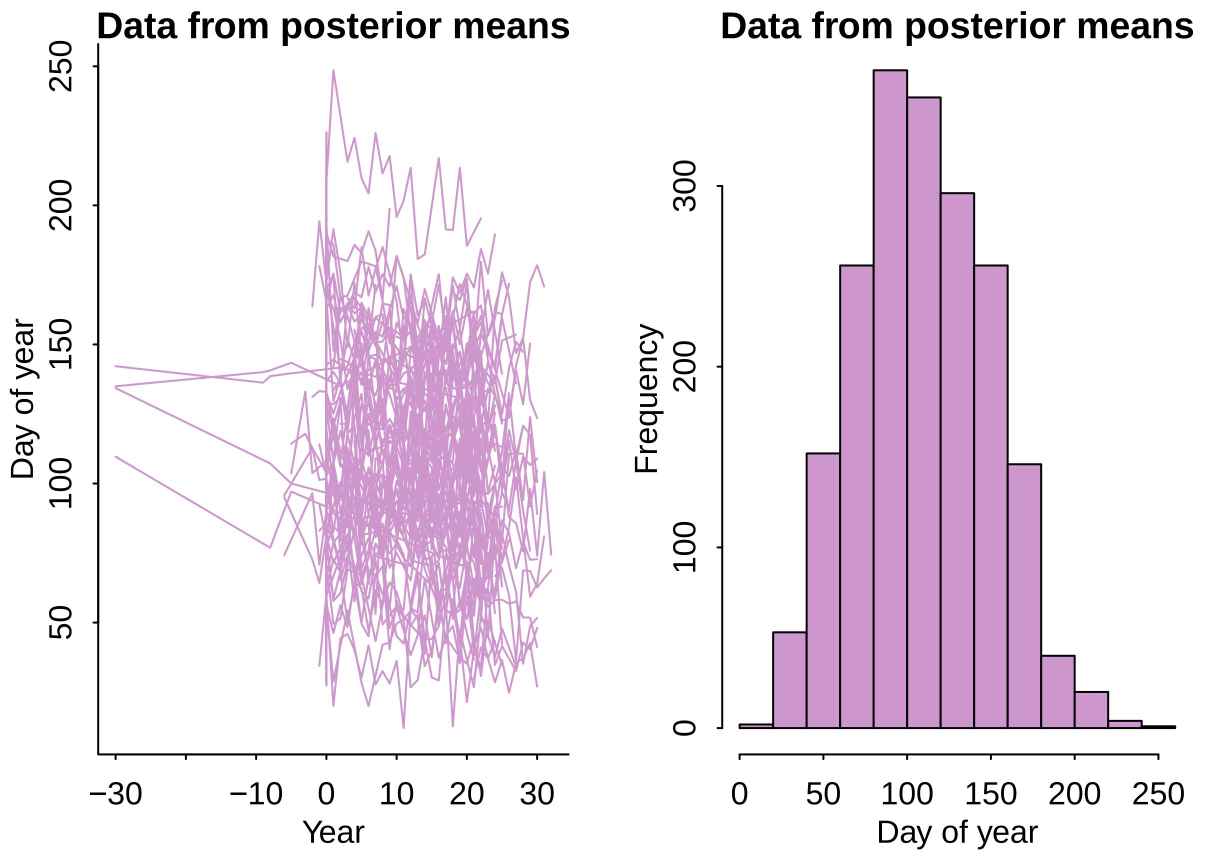
First, let's take a look via two plots at the empricial data and then make comparable plots from the model estimates:

```r
Nreal <- nrow(d)
yreal <- d$phenovalue
```

```r
# First, plot the real data used in the model
par(mfrow=c(1,2))
par(mar=c(3,3,1,1), mgp=c(1.5,.5,0), tck=-.01)
plot(range(year), range(yreal), type="n", xlab="Year",
     ylab="Day of year", bty="l", main="Empirical data")
for (j in 1:Nspp){
  lines(year[species==j], yreal[species==j], col="pink3")
}
hist(yreal, xlab="Day of year", main="Empirical data", col="pink3")
```

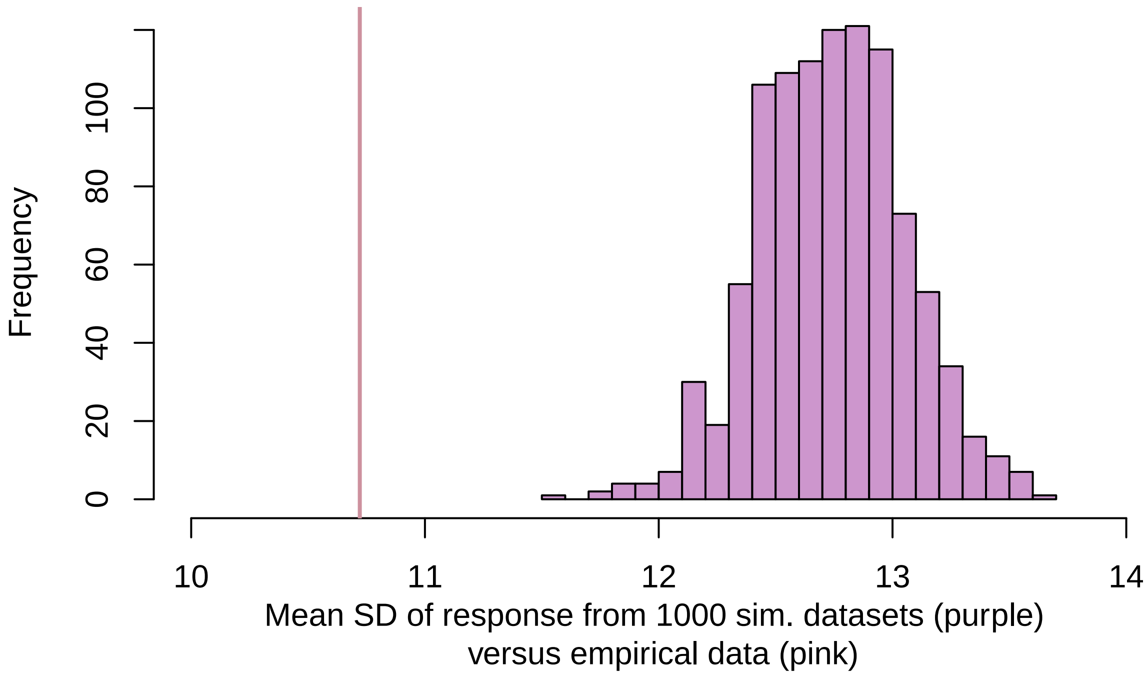


```r
# What does a similar plot look like using the model output?
syncmodelhispost <- extract(syncmodelhis)
# extract means for now (other ways to extract the mean)
sigma_y <- mean(syncmodelhispost$sigma_y)
sigma_a <- mean(syncmodelhispost$sigma_a)
sigma_b <- mean(syncmodelhispost$sigma_b)
mu_b <- mean(syncmodelhispost$mu_b)
mu_a <- mean(syncmodelhispost$mu_a)

a <- rnorm(Nspp, mean=mu_a, sd=sigma_a)
b <- rnorm(Nspp, mean=mu_b, sd=sigma_b)

N <- Nreal

ypred <- length(N)
for (n in 1:N){
    s <- species[n]
    ypred[n] <- a[s] + b[s]*year[n]
}
y <- rnorm(N, ypred, sigma_y)

par(mar=c(3,3,1,1), mgp=c(1.5,.5,0), tck=-.01)
plot(range(year), range(y), type="n", xlab="Year", ylab="Day of year",
    bty="l", main="Data from posterior means")
for (j in 1:Nspp)
  lines(year[species==j], y[species==j], col="plum3")
hist(y, xlab="Day of year", col="plum3", main="Data from posterior means")
```



Okay, but that's just one new draw from our posterior and we used just the mean, so we are not getting a good sense of what the model shows. To address this, these retrodictive checks should be done with many draws. To do this, you need to decide on what summary statistics matter because you cannot just look at each plot. Below we take the standard deviation of y (using the means, but we could and should also consider using other draws of the posterior):

```
# Create the data using new a and b for each of the species, simshere times
simshere <- 1000
# Randomly sample the iterations (we have 4000 here).
iterations <- sample(1:4000, simshere, replace = FALSE)
# Create an empty matrix to get the resulting output
y.sd100 <- matrix(0, ncol=simshere, nrow=Nspp)
# For this version, let's use the species-level intercepts and slopes (e.g., syncmodelhispost[["b"]]) .
for (i in 1:simshere){
    iterhere <- iterations[i]
    for (n in 1:N){
        s <- species[n]
        ypred[n] <- syncmodelhispost[["a"]][iterhere,s]+ syncmodelhispost[["b"]][iterhere,s]*year[n]
    }
  y <- rnorm(N, ypred, syncmodelhispost[["sigma_y"]][iterhere])
  y.df <- as.data.frame(cbind(y, species))
  y.sd <- aggregate(y.df["y"], y.df["species"], FUN=sd)
  y.sd100[,i] <- y.sd[,2]
}

# and here's the real data
real.sd <- aggregate(d["phenovalue"], d[c("studyid", "species")],
    FUN=sd)

par(mfrow=c(1,1))
hist(colMeans(y.sd100), col="plum3", breaks=20, xlim=c(10,14),
    main="",
    xlab="Mean SD of response from 1000 sim. datasets (purple) \n versus empirical data (pink)")
abline(v = mean(real.sd$phenovalue), col = "pink3", lwd = 2)
```

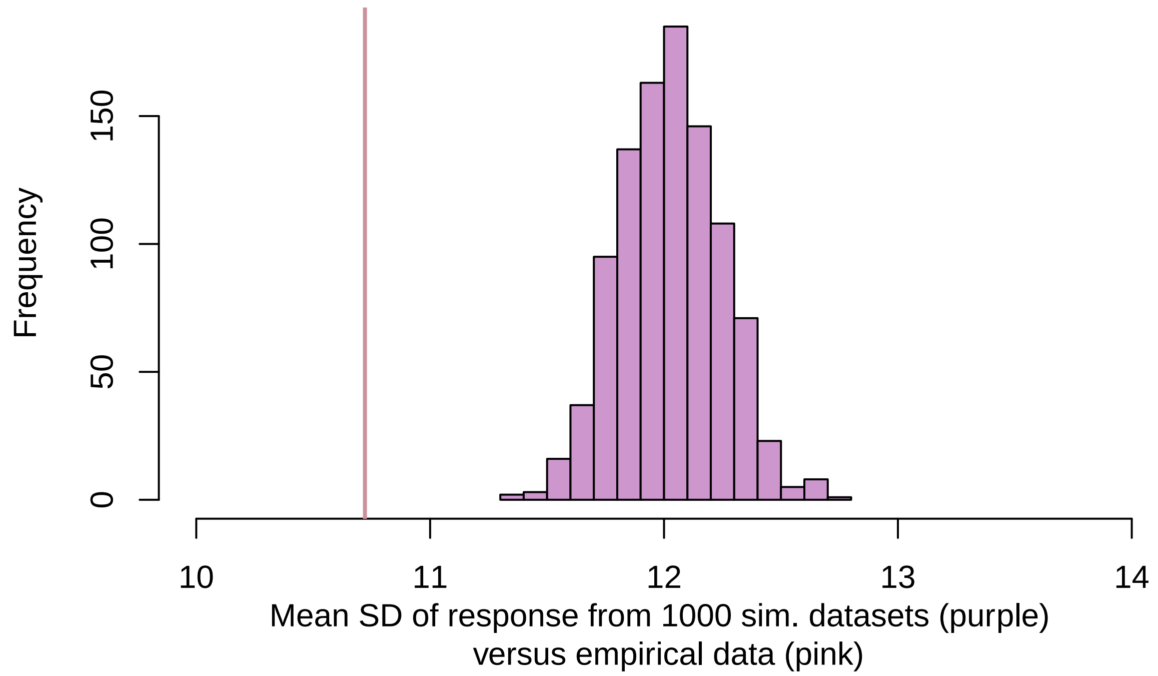


This is okay, but clearly the data has a lower standard deviation than our model expects it. We could decide we are okay with this, but if we have ideas of what might be causing this we could consider adjusting our model.

Feedbacks

At this step, we actually realized the partial pooling on our intercept was not a good fit to the data. We had assumed from the beginning that we should partially pool on the intercepts because that is how fixing `non-independence' withrandom` effects' is often taught in ecology, but our retrodictive checks made us think twice. We realized that we had data from diverse events across locations and their average values were unlikely to be normally distributed (though they may be if we had focused on a certain more local area, such as temperate sites in Europe). We decided to remove the partial pooliing on the intercepts—a model that made biological sense, but we never would have considered without this workflow. We then repeated steps 2-3 and looked again at the retrodictive checks (we show only the final one here):

```r
syncmodelhs <- stan("stan/twolevelhierslope.stan", data=c("N","Nspp","y","species","year"),
                   iter=4000, warmup=3000, chains=4, cores=4, seed=577)

# Random slopes only model:
syncmodelhspost <- extract(syncmodelhs)
sigma_bhsmodel <- mean(syncmodelhspost$sigma_b)
mu_bhsmodel <- mean(syncmodelhspost$mu_b)

ahsmodel <- colMeans(syncmodelhspost$a)
bhsmodel <- rnorm(Nspp, mean=mu_bhsmodel, sd=sigma_bhsmodel)

# extract means for now (other ways to extract the mean)
sigma_y_hsmodel <- mean(syncmodelhspost$sigma_y)

# Create the data using new a and b for each of the species, simshere times
simshere <- 1000
y.sd100 <- matrix(0, ncol=simshere, nrow=Nspp)
for (i in 1:simshere){
    for (n in 1:N){
        s <- species[n]
        ypred[n] <- ahsmodel[s] + bhsmodel[s]*year[n]
    }
  y <- rnorm(N, ypred, sigma_y_hsmodel)
  y.df <- as.data.frame(cbind(y, species))
  y.sd <- aggregate(y.df["y"], y.df["species"], FUN=sd)
  y.sd100[,i] <- y.sd[,2]
}

par(mfrow=c(1,1))
hist(colMeans(y.sd100), col="plum3", breaks=20, xlim=c(10,14),
    main="",
    xlab="Mean SD of response from 1000 sim. datasets (purple) \n versus empirical data (pink)")
abline(v = mean(real.sd$phenovalue), col = "pink3", lwd = 2)
```



This model showed a slightly lower standard deviation, though it was still well away from the empirical data. What could be going on? We had hypotheses about the species, biomes and event types, we went on to test in our research program. For example, new models we built partially pool species depending on their evolutionary distance, which could improve this model as well. This model, however, was already an improvement from where we started and where we end this example.