Hierarchical spatial modelling for applied population and community ecology

Jeffrey W. Doser, Marc Kéry, Gesa von Hirschheydt

24-27 June 2024

Single-species spatial occupancy models

Jeffrey W. Doser

24-27 June 2024

Recall the basic occupancy model

Occupancy (ecological) sub-model

$$z_j \sim \text{Bernoulli}(\psi_j)$$
$$\text{logit}(\psi_j) = \beta_1 + \beta_2 \cdot X_{2,j} + \cdot + \beta_r \cdot X_{r,j}$$

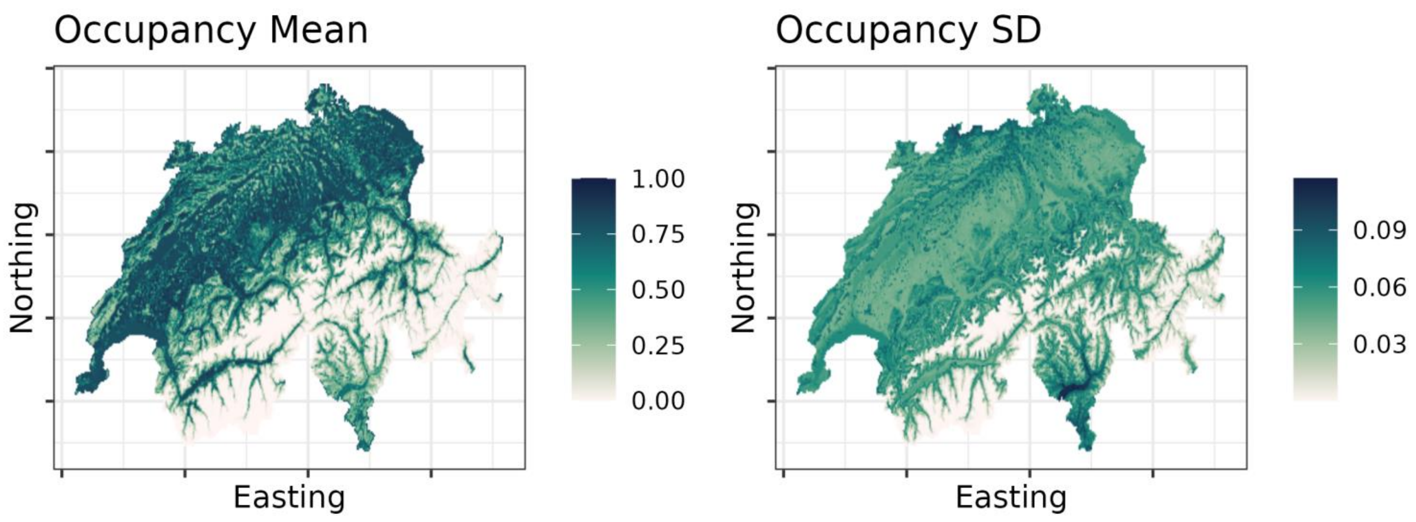
Detection (observation) sub-model

$$y_{j,k} \sim \text{Bernoulli}(p_{j,k} \cdot z_j)$$
$$\text{logit}(p_{j,k}) = \alpha_1 + \alpha_2 \cdot V_{2,j,k} + \cdot + \alpha_r \cdot V_{r,j,k}$$

$j = 1, \ldots, J$ (site)
$k = 1, \ldots, K_j$ (replicate)

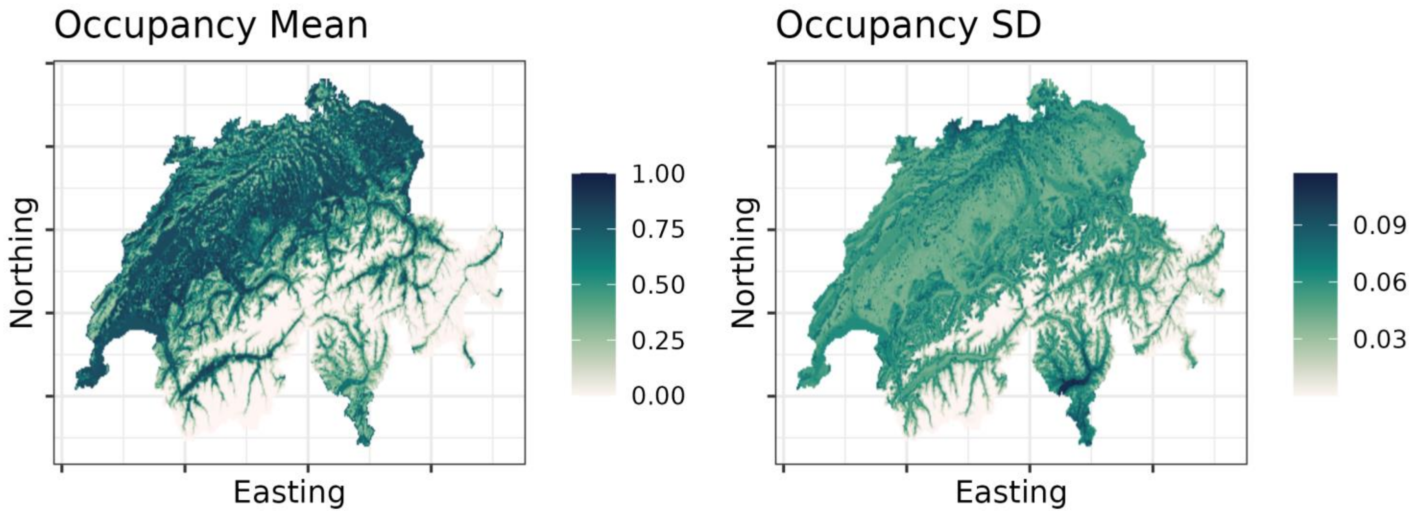
European Goldfinch example

```
out.full <- PGOcc(occ.formula = ~ scale(elevation) + I(scale(elevation)^2) + scale(forest),
                  det.formula = ~ scale(date) + I(scale(date^2)) + scale(dur),
                  data = data.goldfinch,
                  n.samples = 5000,
                  inits = inits.list,
                  priors = prior.list,
                  n.thin = 4,
                  n.burn = 3000,
                  n.chains = 3,
                  n.report = 500)
```



European Goldfinch example

```
out.full <- PGOcc(occ.formula = ~ scale(elevation) + I(scale(elevation)^2) + scale(forest),
                  det.formula = ~ scale(date) + I(scale(date^2)) + scale(dur),
                  data = data.goldfinch,
                  n.samples = 5000,
                  inits = inits.list,
                  priors = prior.list,
                  n.thin = 4,
                  n.burn = 3000,
                  n.chains = 3,
                  n.report = 500)
```



Does accounting for spatial autocorrelation improve our model and resulting species distribution map?

Why *spatial* occupancy models?

- All the same reasons as with any spatial model (i.e., improved prediction, more accurate uncertainty quantification).

Why *spatial* occupancy models?

- All the same reasons as with any spatial model (i.e., improved prediction, more accurate uncertainty quantification).
- Why would (residual) spatial autocorrelation be prevalent in species distributions?

Why *spatial* occupancy models?

- All the same reasons as with any spatial model (i.e., improved prediction, more accurate uncertainty quantification).
- Why would (residual) spatial autocorrelation be prevalent in species distributions?
    1. Environmental drivers or habitat requirements that aren't included as covariates in the model

Why *spatial* occupancy models?

- All the same reasons as with any spatial model (i.e., improved prediction, more accurate uncertainty quantification).
- Why would (residual) spatial autocorrelation be prevalent in species distributions?
  1. Environmental drivers or habitat requirements that aren't included as covariates in the model
  2. Biotic factors (dispersal, conspecific attraction, biotic interactions)

Why *spatial* occupancy models?

- All the same reasons as with any spatial model (i.e., improved prediction, more accurate uncertainty quantification).
- Why would (residual) spatial autocorrelation be prevalent in species distributions?
    1. Environmental drivers or habitat requirements that aren't included as covariates in the model
    2. Biotic factors (dispersal, conspecific attraction, biotic interactions)
- Accounting for spatial autocorrelation can often improve predictions.

Why *spatial* occupancy models?

- All the same reasons as with any spatial model (i.e., improved prediction, more accurate uncertainty quantification).
- Why would (residual) spatial autocorrelation be prevalent in species distributions?
  1. Environmental drivers or habitat requirements that aren't included as covariates in the model
  2. Biotic factors (dispersal, conspecific attraction, biotic interactions)
- Accounting for spatial autocorrelation can often improve predictions.
- Maps of the predicted spatial effects can give insights on the underlying drivers.

Spatial occupancy model

Occupancy (ecological) sub-model

$$z_j \sim \text{Bernoulli}(\psi_j)$$
$$\text{logit}(\psi_j) = \beta_1 + \beta_2 \cdot X_{2,j} + \cdot + \beta_r \cdot X_{r,j} + \text{w}_j$$
$$\mathbf{w} \sim \text{Normal}(\mathbf{0}, \boldsymbol{C}(d, \phi, \sigma^2))$$

Detection (observation) sub-model

$$y_{j,k} \sim \text{Bernoulli}(p_{j,k} \cdot z_j)$$
$$\text{logit}(p_{j,k}) = \alpha_1 + \alpha_2 \cdot V_{2,j,k} + \cdot + \alpha_r \cdot V_{r,j,k}$$

$j = 1, \ldots, J$ (site)
$k = 1, \ldots, K_j$ (replicate)

Spatial occupancy model

Occupancy (ecological) sub-model

$$z_j \sim \text{Bernoulli}(\psi_j)$$

$$\text{logit}(\psi_j) = \beta_1 + \beta_2 \cdot X_{2,j} + \cdot + \beta_r \cdot X_{r,j} + \text{w}_j$$

$$\boxed{\mathbf{w} \sim \text{Normal}(\mathbf{0}, \boldsymbol{C}(d, \phi, \sigma^2))}$$

Detection (observation) sub-model

$$y_{j,k} \sim \text{Bernoulli}(p_{j,k} \cdot z_j)$$

$$\text{logit}(p_{j,k}) = \alpha_1 + \alpha_2 \cdot V_{2,j,k} + \cdot + \alpha_r \cdot V_{r,j,k}$$

$j = 1, \ldots, J$ (site)
$k = 1, \ldots, K_j$ (replicate)

Spatial NNGP occupancy model

Occupancy (ecological) sub-model

$$z_j \sim \text{Bernoulli}(\psi_j)$$
$$\text{logit}(\psi_j) = \beta_1 + \beta_2 \cdot X_{2,j} + \cdot + \beta_r \cdot X_{r,j} + \text{w}_j$$
$$\boxed{\mathbf{w} \sim \text{Normal}(\mathbf{0}, \tilde{\boldsymbol{C}}(d, \phi, \sigma^2))}$$

Detection (observation) sub-model

$$y_{j,k} \sim \text{Bernoulli}(p_{j,k} \cdot z_j)$$
$$\text{logit}(p_{j,k}) = \alpha_1 + \alpha_2 \cdot V_{2,j,k} + \cdot + \alpha_r \cdot V_{r,j,k}$$

$j = 1, \ldots, J$ (site)
$k = 1, \ldots, K_j$ (replicate)

Prior distributions (and spOccupancy defaults)

$$\beta_r \sim \text{Normal}(0, 2.72)$$

$$\alpha_r \sim \text{Normal}(0, 2.72)$$

$$\sigma^2 \sim \text{Inverse-gamma}(2, 1)$$

$$\phi \sim \text{Uniform}(\frac{3}{d_{\text{max}}}, \frac{3}{d_{\text{min}}})$$

Prior distributions (and spOccupancy defaults)

$$\beta_r \sim \text{Normal}(0, 2.72)$$

$$\alpha_r \sim \text{Normal}(0, 2.72)$$

$$\sigma^2 \sim \text{Inverse-gamma}(2, 1)$$

$$\phi \sim \text{Uniform}(\frac{3}{d_{\text{max}}}, \frac{3}{d_{\text{min}}})$$

See "2-intro-to-spatial-models.pdf" lecture

Prior distributions (and spOccupancy defaults)

Where does the 2.72 come from?

$$\beta_r \sim \text{Normal}(0, 2.72)$$

$$\alpha_r \sim \text{Normal}(0, 2.72)$$

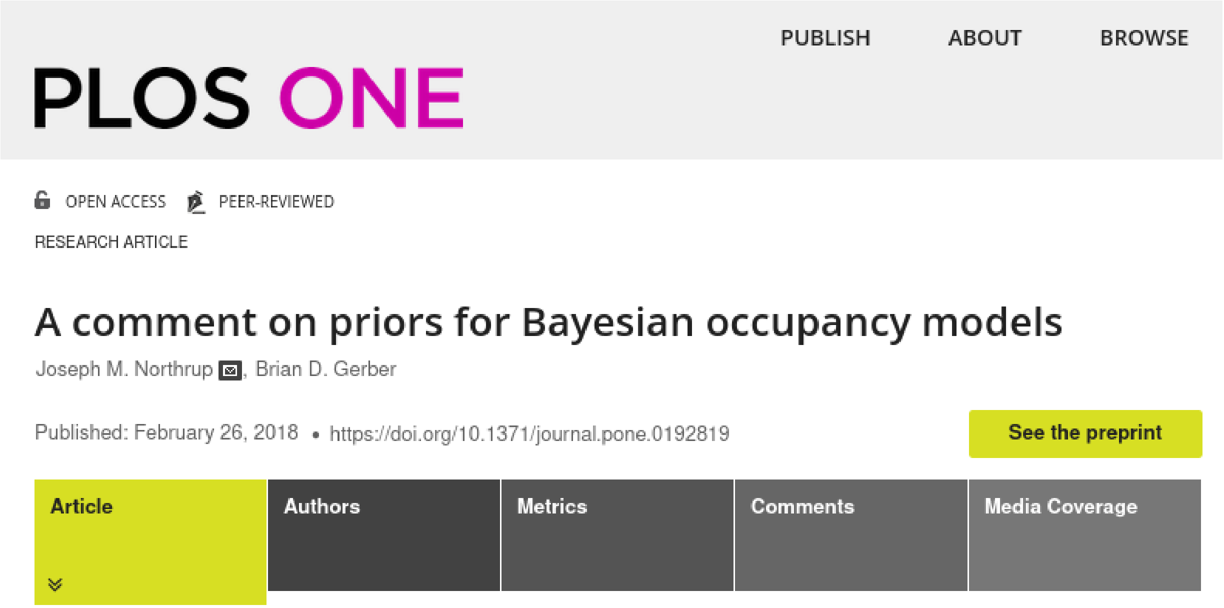
$$\sigma^2 \sim \text{Inverse-gamma}(2, 1)$$

$$\phi \sim \text{Uniform}(\frac{3}{d_{\max}}, \frac{3}{d_{\min}})$$

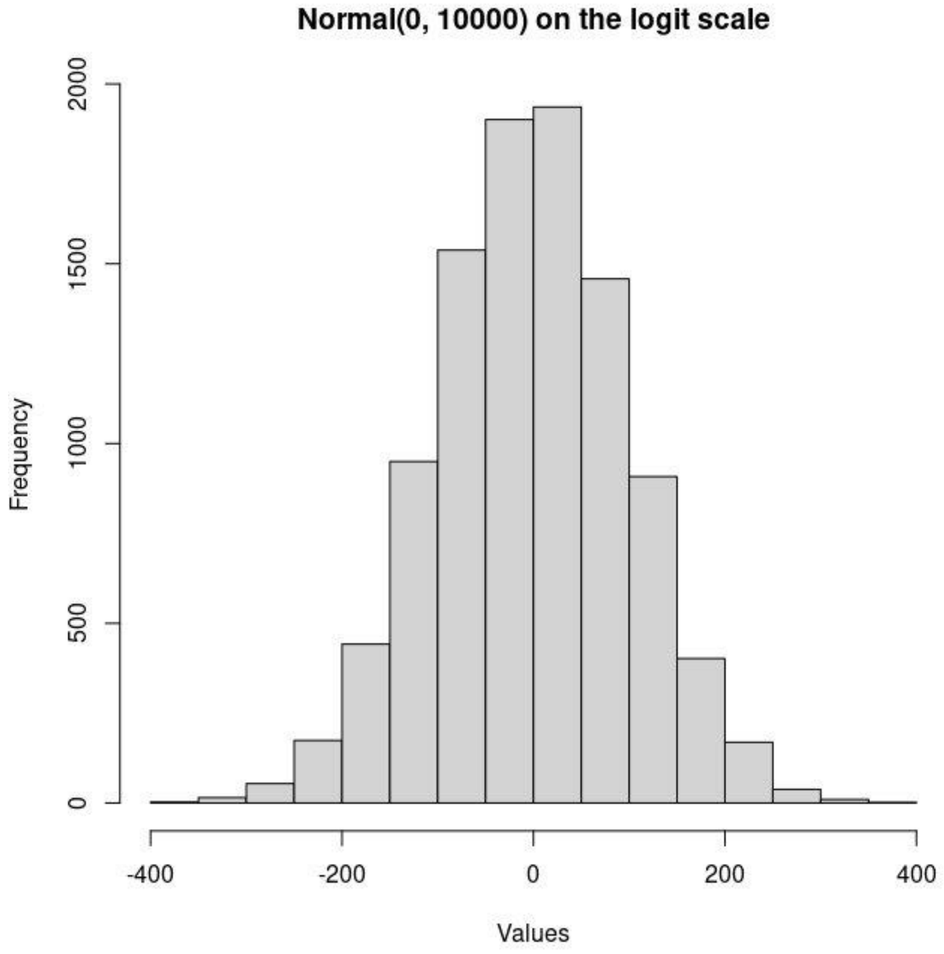
See "2-intro-to-spatial-models.pdf" lecture

Priors on regression coefficients in occupancy models

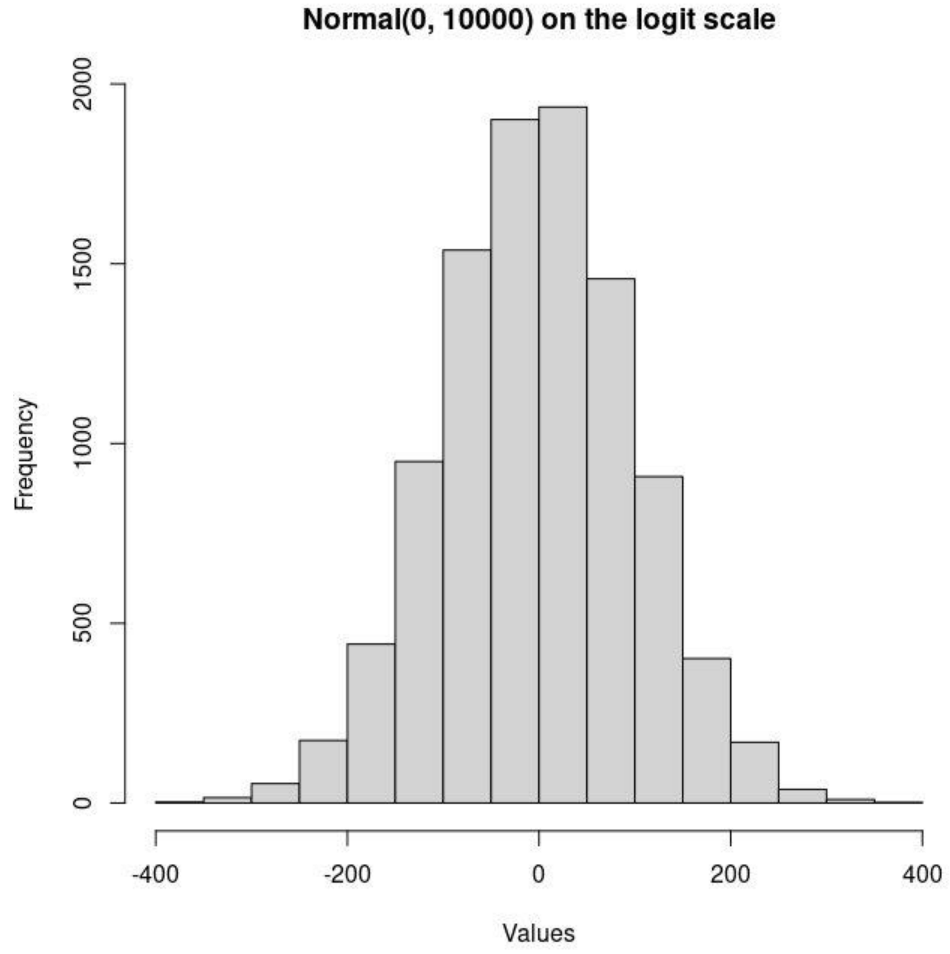
- A standard prior in regression models for the intercept and coefficients is a Normal prior with mean 0 and a very big variance (e.g., 1000).
- However, this actually becomes a very informative prior in occupancy models (and any type of binomial GLM)!!
- This is because the regression coefficients are on the logit scale.

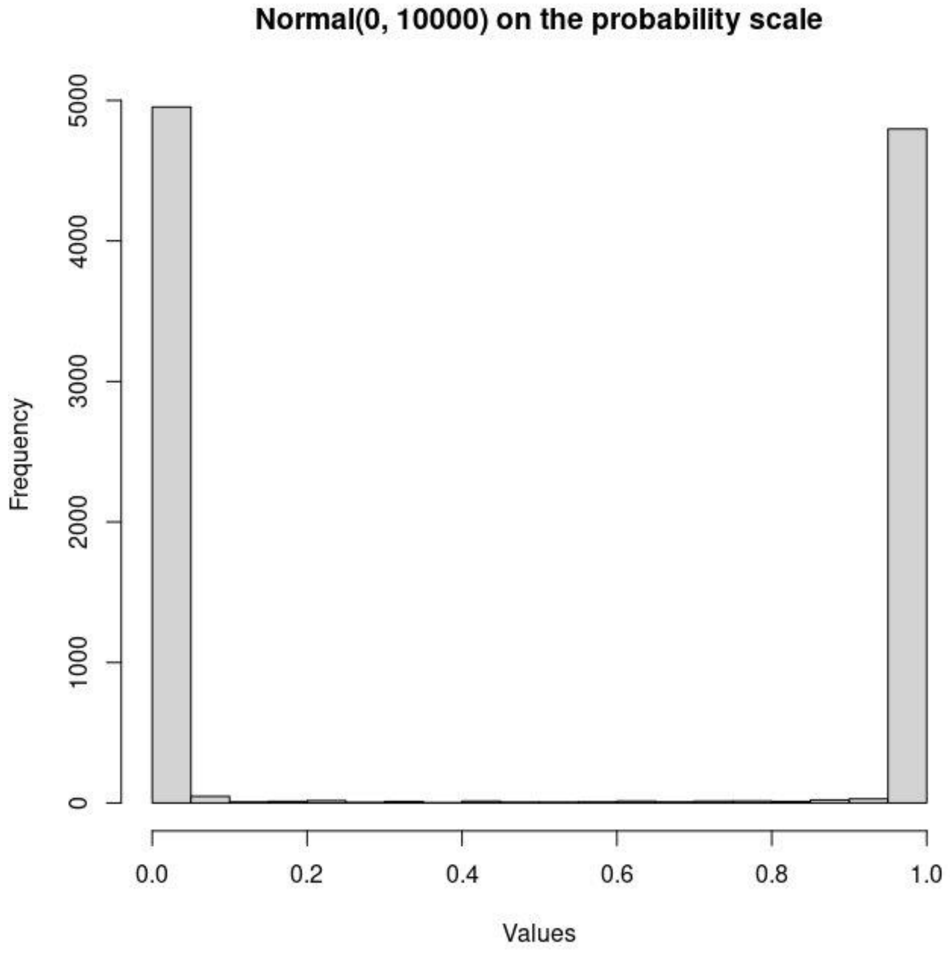


Priors on regression coefficients in occupancy models

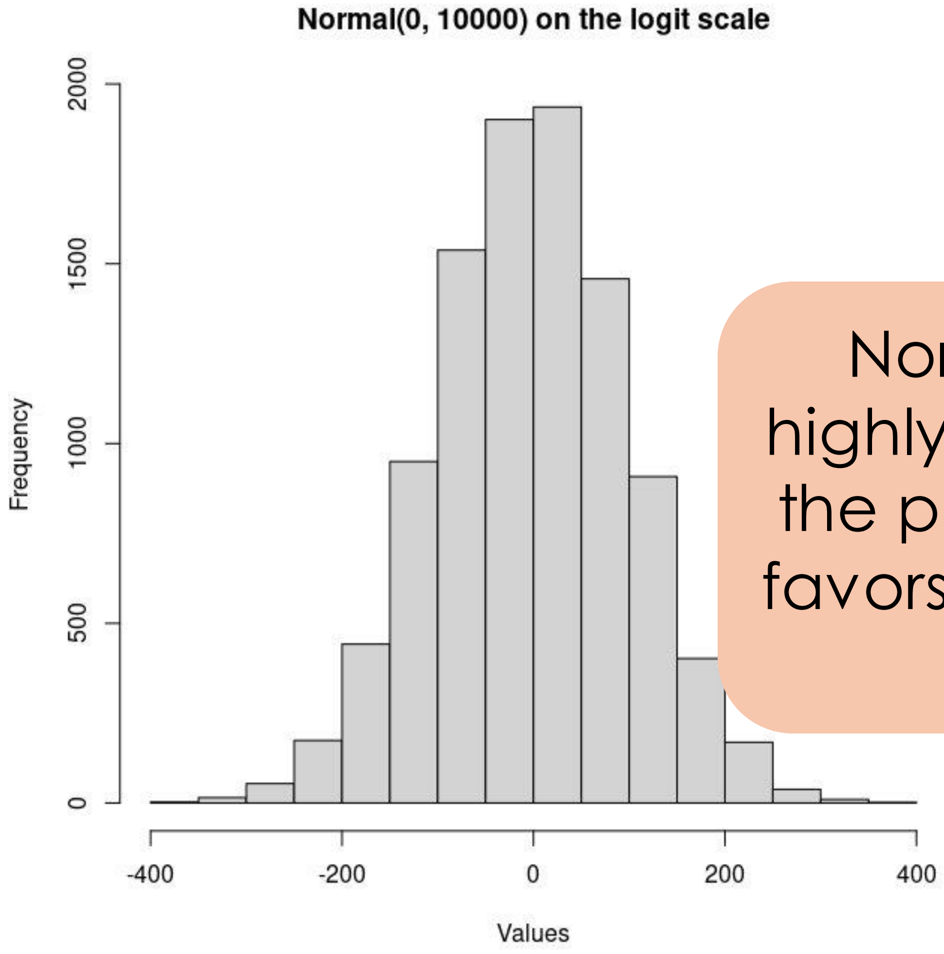


Priors on regression coefficients in occupancy models

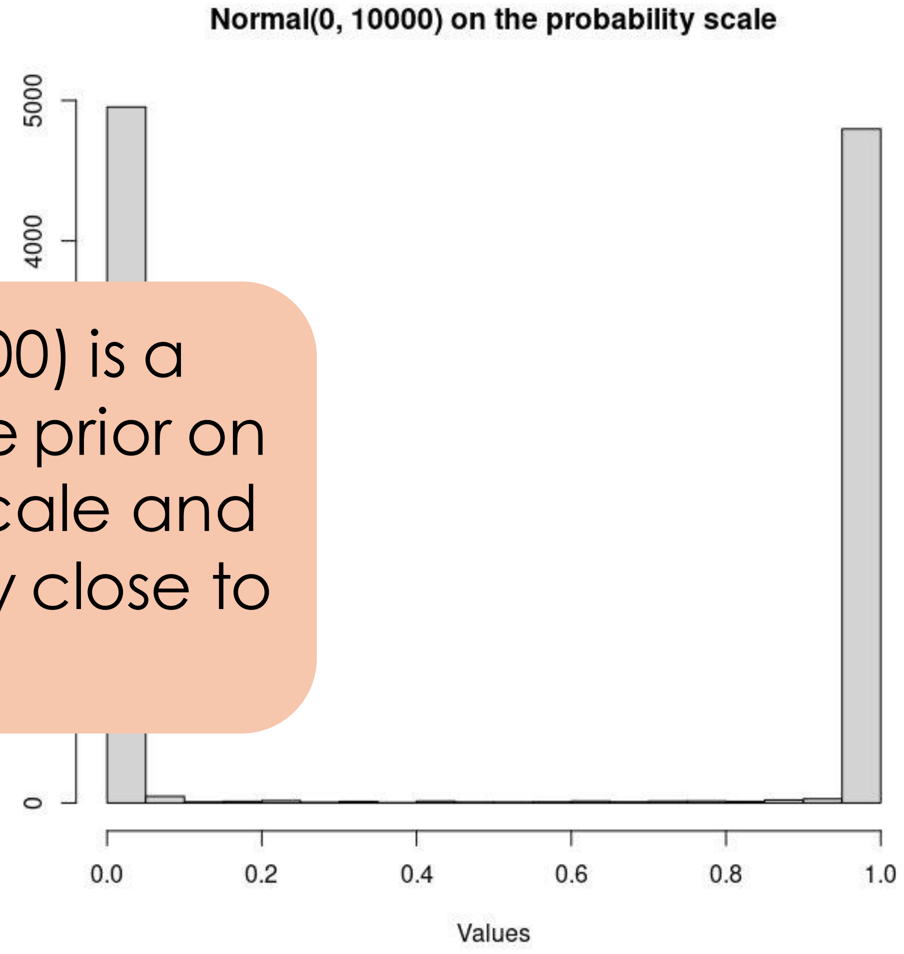




Priors on regression coefficients in occupancy models

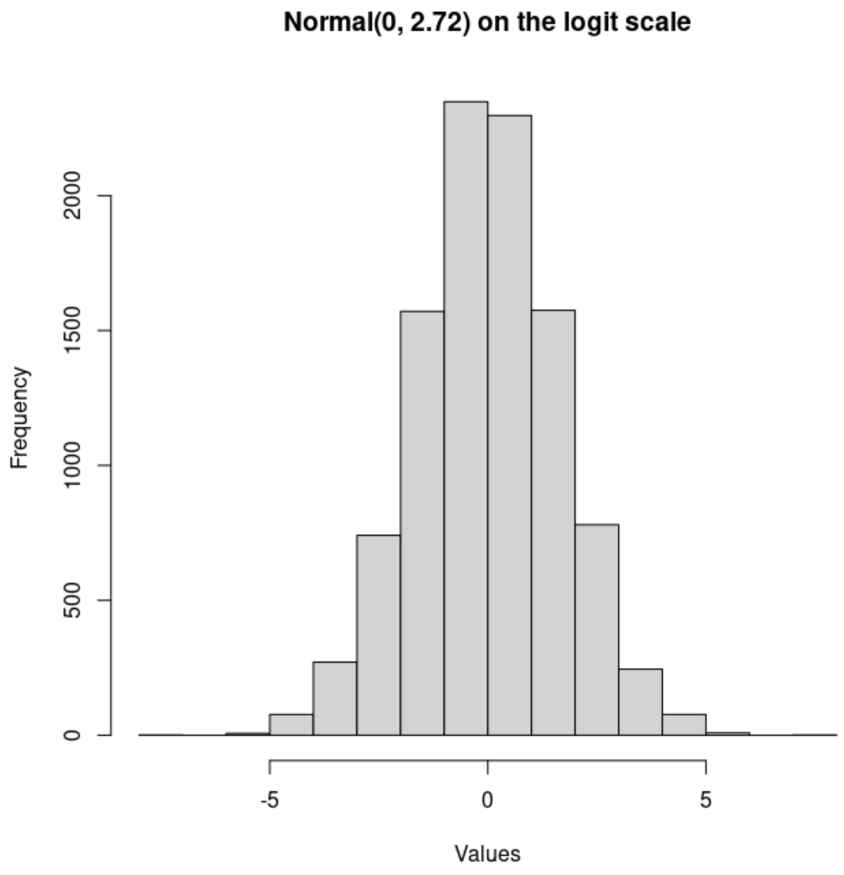


Normal(0, 10000) is a highly informative prior on the probability scale and favors values very close to 0 or 1!



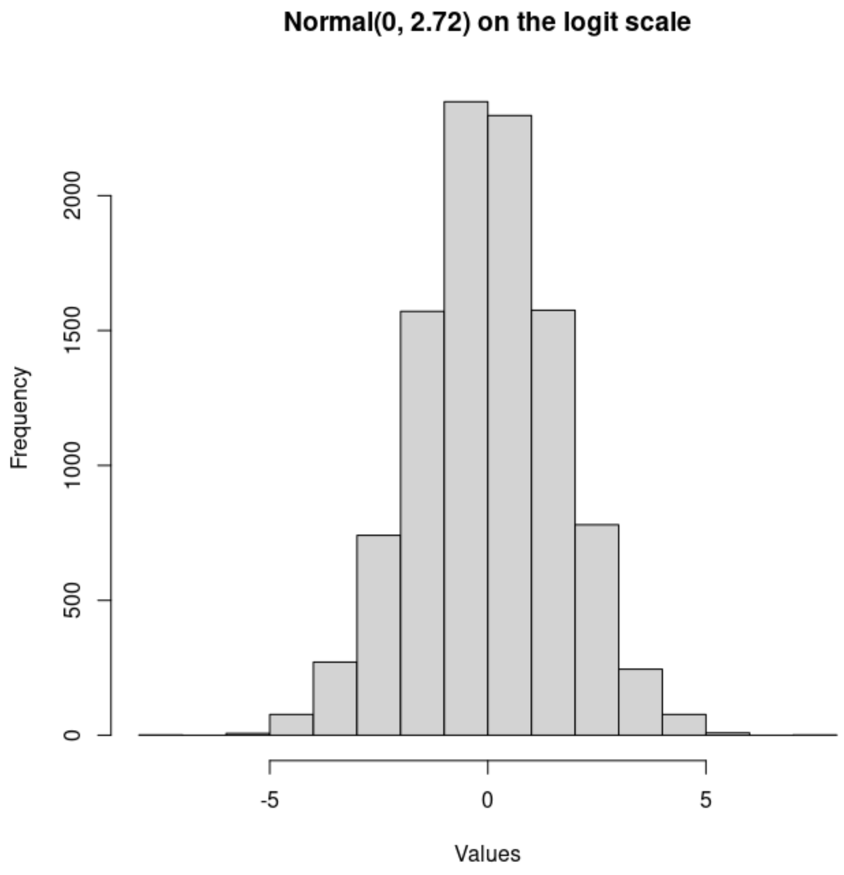
Priors on regression coefficients in occupancy models

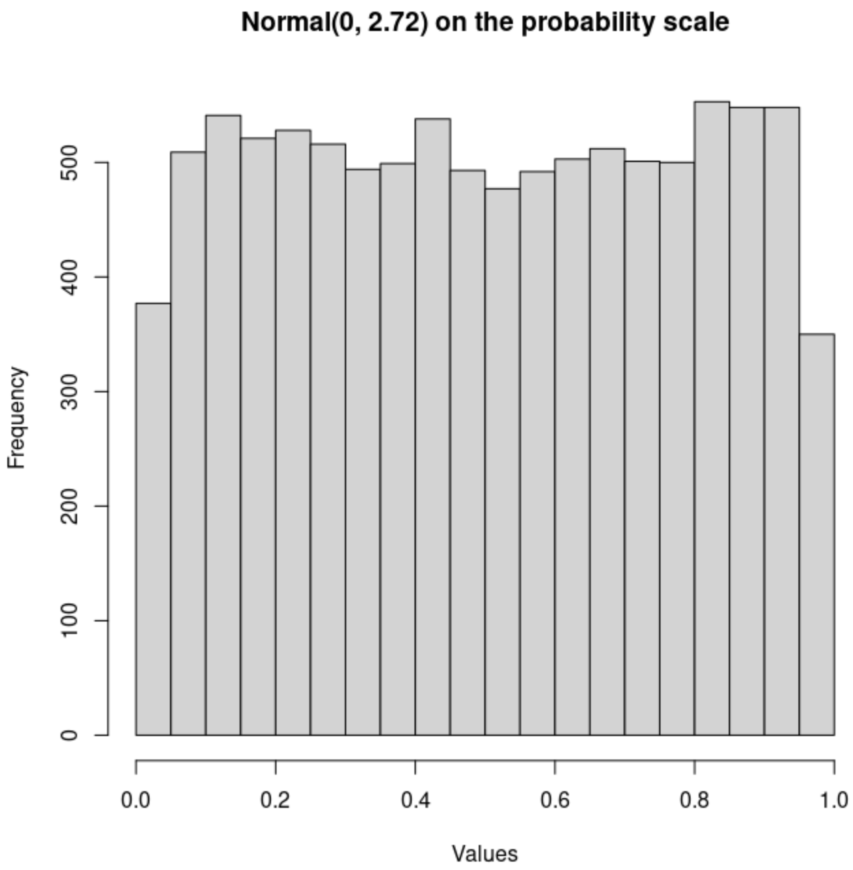
- Instead, a good prior is Normal(0, 2.72)



Priors on regression coefficients in occupancy models

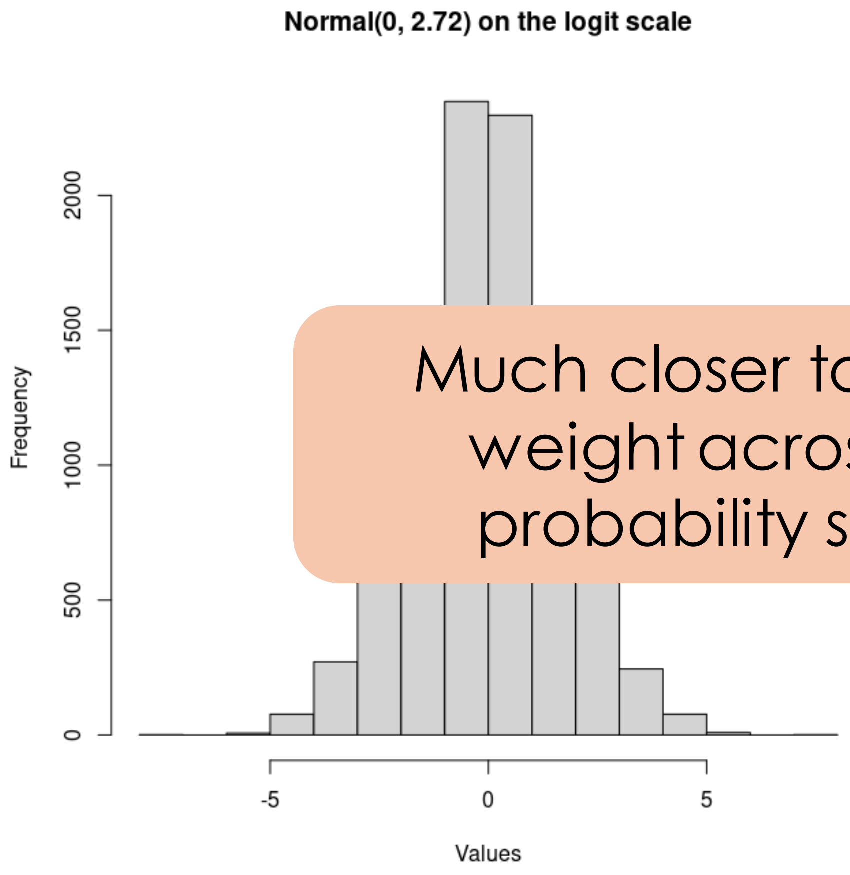
- Instead, a good prior is Normal(0, 2.72)



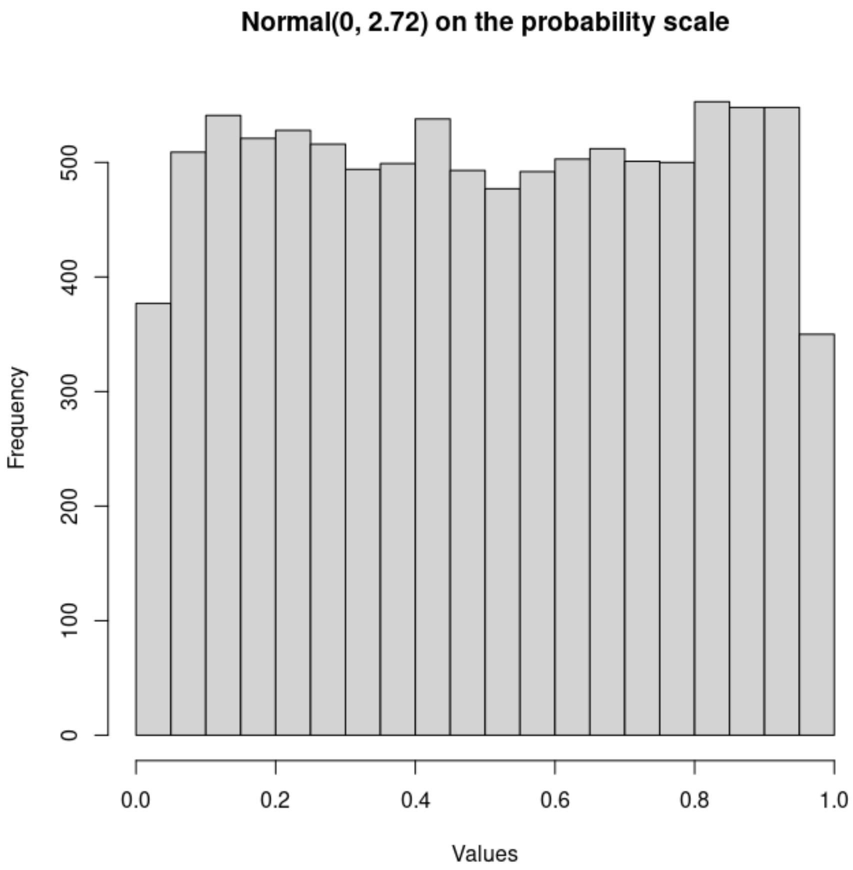


Priors on regression coefficients in occupancy models

- Instead, a good prior is Normal(0, 2.72)



Much closer to equal weight across the probability scale.



Fitting spatial occupancy models in spOccupancy

- Function: spPGOcc (spatial Pólya-gamma occupancy model)
- Supports both full GP and NNGP models.
- The model is implemented using an algorithm called an Adaptive Metropolis sampler, which requires special considerations.

```
spPGOcc(occ.formula, det.formula, data, inits, priors,
        tuning, cov.model = "exponential", NNGP = TRUE,
        n.neighbors = 15, search.type = "cb", n.batch,
        batch.length, accept.rate = 0.43,
        n.omp.threads = 1, verbose = TRUE, n.report = 100,
        n.burn = round(.10 * n.batch * batch.length),
        n.thin = 1, n.chains = 1, k.fold, k.fold.threads = 1,
        k.fold.seed = 100, k.fold.only = FALSE, ...)
```

A quick overview of the adaptive Metropolis algorithm

- Estimating the spatial decay parameter ϕ is difficult.
- At each step of the MCMC algorithm, we propose a new value for ϕ.
- The proposed value comes from a Normal distribution:
  - Mean: the current value of ϕ
  - Variance: the ***Tuning Variance***
- We use an algorithm to determine if we should keep the previous value, or accept the new one.
- Ideally, we want to accept the proposed value around 43% of the time.

A quick overview of the adaptive Metropolis algorithm

- A regular (non-adaptive) Metropolis algorithm does not change the tuning variance. It is set prior to starting the model.

A quick overview of the adaptive Metropolis algorithm

- A regular (non-adaptive) Metropolis algorithm does not change the tuning variance. It is set prior to starting the model.
- An adaptive Metropolis algorithm changes the tuning variance throughout the MCMC sampler to get closer to the target acceptance rate.

A quick overview of the adaptive Metropolis algorithm

- A regular (non-adaptive) Metropolis algorithm does not change the tuning variance. It is set prior to starting the model.
- An adaptive Metropolis algorithm changes the tuning variance throughout the MCMC sampler to get closer to the target acceptance rate.
- To do this, the MCMC samples are split into a set of ***batches***
- Each batch has a pre-specifed set of MCMC samples (e.g., 25)
- After each batch, the tuning variance is adjusted to get closer to an acceptance rate of 0.43.

A quick overview of the adaptive Metropolis algorithm

- A regular (non-adaptive) Metropolis algorithm does not change the tuning variance. It is set prior to starting the model.
- An adaptive Metropolis algorithm changes the tuning variance throughout the MCMC sampler to get closer to the target acceptance rate.
- To do this, the MCMC samples are split into a set of ***batches***
- Each batch has a pre-specifed set of MCMC samples (e.g., 25)
- After each batch, the tuning variance is adjusted to get closer to an acceptance rate of 0.43.
- **This approach can greatly speed up the time to convergence for difficult to estimate parameters like ϕ.**

What do you need to specify in `spOccupancy`?

1. The initial tuning variance (`tuning`).
2. The acceptance rate (`accept.rate`).
3. The number of MCMC batches to run (`n.batch`).
4. The number of MCMC samples in each batch (`batch.length`).

The total number of MCMC samples is n.batch * batch.length

Exercise: Spatial modeling of the European goldfinch

4-spatial-european-goldfinch.R