Introducción a los métodos estadísticos bayesianos en Ecología

Programa del curso:

1. Introducción general
2. Elementos básicos del análisis bayesiano
3. Análisis bayesiano I
4. Análisis bayesiano II
5. Modelos bayesianos jerárquicos

Modelos Jerárquicos Teórico 05

a. Incertidumbre de medición.
b. Ciertos modelos jerárquicos: teoría básica
c. JAGS: just another Gibbs sampler
d. Occupancy models
e. Binomial-mixture models: un ejemplo

a. Incertidumbre de medición.

Excepto en casos muy bien definidos (ej. filtro de Kalman), la estadística frecuentista carece de métodos genéricos para tratar los errores de medición de variables.

Los modelos jerárquicos bayesianos modelan la incertidumbre de medición usando "variables latentes".

Latente: denota a una condición potencialmente existe pero que no es directamente observable o medible.

Ej: la abundancia real de una población natural, la calidad de vida, la confianza en el clima de negocios, la moral, la felicidad, el optimismo, el conservadurismo, etc.

Estas variables latentes afectan o determinan los valores de las variables que sí podemos cuantificar directamente.
Si bien a veces el carácter de “latente” proviene de la ambigüedad de su definición, en muchos casos se denotan aspectos interesantes y complejos de la realidad
Los efectos de las vars explicativas que denotan “procesos causales” ocurren sobre las variables latentes, las que sólo registramos con incertidumbre o error de medición.

$Y_{i,obs} = Y_{i,latente}(\text{vars explicativas}_i) + \text{error}_i.$

(la relación no tiene por qué ser aditiva!)

El desafío es hacer inferencias correctas del efecto de las vars explicativas sobre $Y_{i,latente}$ cuando solo tenemos $Y_{i,obs}$

Ya hemos visto antes algo parecido a: $Y_{i,obs}=Y_{i,latente}(\text{vars explic}_i)+\text{error}_i$.

El Modelo Lineal Mixto como un modelo condicional: $Y=Y|b_0$ en efectos de grupo (o aleatorios) $b_0 \sim Normal(0, \sigma^2_{b0})$ tal que $Y \sim Normal\ (\mu_Y=\mu_Y(X)+b_0, \sigma^2_Y)$.

Tambien los GLMM: $Y=Y|b_0$ ahora con $Y \sim BiNeg(\mu_Y, \phi)$ y efectos de grupo (o aleatorios) con $b_0 \sim Normal(0, \sigma^2_{b0})$ tal que $\log(\mu_Y)=X\beta + b_0$.

Visión simplista de ranefs: serían asimilables a un "error"...

En los GLMM, los efectos de grupo denotan parte de la co-varianza de Y debida al diseño experimental o de muestreo que no está asociada al efecto de las vars explicativas X sobre μ_Y.

La var de respuesta Y cuya media modelamos como función lineal de las Xs y del efecto de una variable latente (los ranefs!) sobre log(μ_Y) que no es directamente observable.

Un supuesto fundamental de los GLMM es que los efectos poblacionales (fijos) y de grupo (aleatorios) son aditivos en la escala de la función de enlace.

Pero si quisiéramos "generalizar" los GLMM:

(a) los efectos de grupo podrían ser modelados con una distribución de probabilidades acaso diferente a Normal.

(b) la composición de los efectos de poblacionales (fijos) y los de grupo podría no ser necesariamente aditiva.

La abundancia poblacional no se mide, se estima siempre y por tanto siempre hay un error o incertidumbre asociada.

Una lista no exhaustiva de errores comunes incluye:

a) Errores de conteo cuando Y>0: conteos múltiples del mismo individuo, subestimación de grupos de individuos si N muy alto.

b) Cuando $Y_{obs}=0$:
- Ceros verdaderos: no hay individuos.
- Ceros falsos: individuos presentes pero no detectados (falso negativo).

(idem para los drogadictos y los evasores de impuestos!)

c) Incertidumbre en identificación de la especie: (se puede modelar, por ej. usando observadores múltiples)

El caso común: la abundancia REAL de una población no es directamente medible (i.e. una variable latente) debido a la incertidumbre de detección, particularmente en los ceros.

Los procesos "causales" actúan sobre la abundancia REAL, no sobre nuestra estimación inevitablemente distorsionada por los problemas de detección que deben ser modelados!.

Dos consecuencias de pretender que los conteos/presencias carecen de incertidumbre de detección como en los GLMM:

(a) La abundancia es siempre (muy) SUBESTIMADA.

(b) Los efectos de vars explicativa son siempre subestimados, incluyendo la estimación de tendencias (N vs t): cuando <0, la declinación poblacional real es mucho más rápida!.

Métodos de CMR basados en marcado individual permiten estimar la prob. de observación y así corregir la estimación de abundancia; requieren un importante esfuerzo empírico.

Hay además abundante bibliografía metodológica (Williams et al 2002, O'Connell et al 2010, Royle et al 2004, Kéry & Royle 2016, y muchos otros) sobre cómo estimar presencia o abundancia con infinitas variaciones adaptables a diferentes especies, usando cámaras, aviones, señuelos, sonogramas, marcas, cantos/reconocimientos, teledetección, etc.

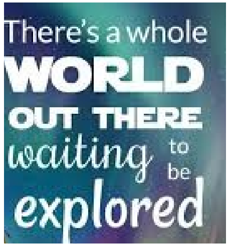
Es imposible resumir esta vasta bibliografía en un solo curso.

Uds. tienen que entender al menos las bases de estos métodos para usarlos o al menos leer los papers y libros al respecto.

En los últimos 15+ años: diseños y metodologías de muestreo para hacer inferencias sobre presencias/conteos considerando la incertidumbre de detección sin requerir "capturas" individuales.

Esta creciente sofisticación en el análisis de la información de estos métodos dejan muy en claro que es cada vez más difícil publicar "los datos observacionales del naturalista entusiasta".

Hay paquetes de R con funciones ya escritas para diseños de muestreo específicos: unmarked (likelihood), ahm, ubms, camtrapR, multiocc, stocc, spOccupancy, etc.



Aquí usaremos ciertos modelos jerárquicos bayesianos para evaluar el efecto de variables explicativas sobre la abundancia/presencia y también sobre la prob. de detección.

b. Ciertos modelos jerárquicos: teoría básica

Estos modelos que contienen dos GLMs interconectados son:

(a) Occupancy model (MacKenzie et al 2002 y otros).

(b) Binomial mixture model (Royle 2004 y otros).

Ambos son modelos condicionales: $Y_{obs}|C$(onteo) o Ocup(ación) en el sentido que ya explicamos antes con los GLMM y se modela explícitamente la incertidumbre de detección.

* Las variables REALES C o Ocup son latentes, no observables.

* Las variables observables $Y_{obs} = C*p$ o $Y_{obs} = Ocup*p$ son una combinación multiplicativa del proceso real (C o Ocup) y del proceso de observación/detección con probabilidad p.

En su forma más simple, ambos modelos usan un "diseño de visitas repetidas" en un lapso corto de tiempo (population closure) a s sitios, sin "marcar y seguir" individuos.

Occupancy model

	v1	v2	v3
1	0	1	0
	0	0	0
	…	…	…
s	1	0	1

Proceso de estado:

$$ocup \sim Binomial(\varphi) \left\langle \begin{matrix} 0 \\ 1 \end{matrix} \right.$$

Proceso de observ.:

$$p \sim Binomial(\pi) \left\langle \begin{matrix} 0 \\ 1 \end{matrix} \right.$$

Observación	Especie: Ausente	Especie: Presente
No obs.	$0*0=0$	$0*1=0$
Si obs.	$1*0=0$	$1*1=1$

$$Y_{obs} = ocup*p$$

ocup es la variable latente no directamente observable y es modelada como un efecto aleatorio BINOMIAL que afecta Y_{obs}!

Dos GLM combinados

$$logit(\varphi) = X_1\beta_1 + b_0 \qquad logit(\pi) = X_2\beta_2 + b_0^*$$

En el Binomial mixture model se estiman abundancias locales:

	v1	v2	v3
1	0	4	2
	4	0	0
	…	…	…
s	1	12	0

Proceso de estado:

$C \sim Poisson(\mu_C)$

$C \sim BiNeg(\mu_C, \phi)$

[0,1..N)

Proceso de observ.:

$p \sim Binomial(\pi) < \begin{matrix} 0 \\ 1 \end{matrix}$

Aquí C es la variable latente no directamente observable y es modelada como un efecto aleatorio Poisson que afecta Y_{obs}!

De nuevo, hay una combinación multiplicativa: $Y_{obs} = C*p$

Y también hay 2 GLM combinados:

$\log(\mu_C) = X_1\beta_1 + b_0$ $\quad$ $\text{logit}(\pi) = X_2\beta_2 + b_0^*$

Ejemplo:

$\log(\mu_C) = \beta_0 + \beta_1 AreaBosque + \ldots + a_G \leftarrow$ efecto de grupo

$\text{logit}(\pi) = \beta^*_0 + \beta^*_1 esfuerzo + \ldots + a_O \leftarrow$ efecto observador

El modelo empleado para C puede también ser Binomial Negativa, ZIPoisson, ZIBinomial Negativa, o cualquier otro.

Todo lo que aprendieron sobre GLM (bayesianos) se usa y aplica aquí: convergencia de cadenas, distr predictiva posterior, etc.

Estos modelos básicos hay sido extendidos (y complejizados) para considerar otros diseños de muestreo, analizar datos dinámicos o en el espacio, o de varias de varias spp, etc..

Por razones técnicas, esta clase de modelos jerárquicos no se puede ajustar fácilmente con Stan (y por ende ~~brms~~).

(hay ciertas reparametrizaciones y trucos, pero no son generales)

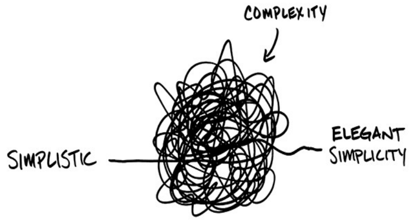
Sin embargo, se pueden ajustar con JAGS, para lo cual hay que aprender a generar el input y analizar el output.

Estos dos modelos jerárquicos suponen que:

* Población cerrada: sin inmigración ni emigración.
* Un solo sitio, una sola visita, un solo observador.
* No hay falsos + o conteos múltiples.
* Eventos de detección son independientes.
* Homogeneidad de detección para los N individuos.

El apartamiento de estas suposiciones (IID) requiere añadir vars. explicativas o eventualmente modificar la estructura del modelo jerárquico para incluir otros aspectos realistas.

Aumentar el realismo y la generalidad de los modelos tiene un costo en su complejidad…



Hay muchos protocolos de observación implementados o implementables como modelos jerárquicos, incluyendo dobles observadores (dependientes), DISTANCE, transectas, múltiples visitas (in)dependientes, modelos de ocupación con/sin metapoblaciones/comunidades, etc.

Muchas de estas ideas relacionadas con la incertidumbre en los conteos se están también muy, pero muy lentamente a emplear en ecología de comunidades.

→ Todo esto requiere métodos bayesianos, claro!

Si bien no es un curso de métodos en ecología, creo que señalo una perspectiva claramente marcada.

c. JAGS: just another Gibbs sampler (Plummer 2003)

Fue una implementación del algoritmo de Gibbs (que vimos hace una semana) como un programa independiente.

Hay un paquete básico de interfase (rjags) que permite una interacción simplista y limitada con JAGS a partir de R.

Para ello hay que escribir un mínimo de código usando la sintaxis de JAGS a partir de R.

Y el post-procesamiento de las distr. posteriores y de las distr predictivas posteriores es bastante primitivo, por lo que hay que escribir algún código también.

JAGS tiene la reputación de hacer crash con cierta frecuencia por razones no siempre comprensibles.

En JAGS, un modelo se compone de:

* los datos
* la verosimilitud
* las distr previas
* las cantidades derivadas a calcular (opcional)

Es más simple crear los pedazos y luego juntarlos en una lista.

Esta lista es guardada como un archivo de texto (.txt).

Archivo .txt + otras especificaciones (#cadenas, #iterac., #burn-in, etc.) → JAGS

Dist posteriores de parámetros y de otras cantidades derivadas.

d. Occupancy models

Los datos: surveys del Carbonero montano hechos 2-3 veces por año durante época de reproducción en 237 áreas de 1-km^2 en Suiza (Royle & Dorazio 2008).

```
> str(DF)
'data.frame':   237 obs. of  15 variables:
 $ y.1   : int  0 0 0 0 0 0 0 0 1 0 ...
 $ y.2   : int  0 0 0 0 0 0 0 0 1 0 ...
 $ y.3   : int  0 0 0 0 0 0 0 0 0 1 ...
 $ c.1   : int  0 0 0 0 0 0 0 0 2 0 ...
 $ c.2   : int  0 0 0 0 0 0 0 0 1 0 ...
 $ c.3   : int  0 0 0 0 0 0 0 0 0 1 ...
 $ elev  : int  420 450 1050 1110 510 630 590 530 1140 770 ...
 $ forest: int  3 21 32 35 2 60 5 13 50 57 ...
 $ dur.1 : int  240 160 120 180 210 150 115 155 165 220 ...
 $ dur.2 : int  240 155 105 170 225 180 105 140 165 230 ...
 $ dur.3 : int  240 140 85 145 235 195 95 135 180 255 ...
 $ day.1 : int  29 13 30 23 28 17 16 24 25 21 ...
 $ day.2 : int  58 39 47 44 56 56 37 47 46 38 ...
 $ day.3 : int  73 62 74 71 73 73 76 74 70 50 ...
 $ length: num  6.2 5.1 4.3 5.4 3.6 6.1 5.1 3.7 3.8 7.7 ...
```

y.1 to y.3 : detección/no detección (1/0) en c/u de las 3 visitas

c.1 to c.3 : # territorios en en cada una de las 3 visitas

elev: elevación (m)

forest: % cobertura forestal

dur.1 to dur.3: duración de cada visita (min)

day.1 to day.3: días desde 1 Abril

length: largo transecta (km)

```
DF$elev=as.vector(scale(DF$elev, center=T, scale=T))
DF$forest=as.vector(scale(DF$forest, center=T, scale=T))
```

Estandardiza las vars explicativas numéricas

```
> DF[,c("y.1","y.2", "y.3")]
  y.1 y.2 y.3
1   0   0   0
2   0   0   0
3   0   0   0
4   0   0   0
5   0   0   0
6   0   0   0
```

Objetivo: determinar la abundancia global, tomando en cuenta la probabilidad de detección, y la relación entre la probabilidad de ocupación y la elevación y la superficie local de bosque.

Problema: ¿cómo interpretar los ceros i.e. las no detecciones?

0 verdadero: animal realmente ausente.

0 falso: animal presente, pero no detectado (error)

Los gráficos exploratorios con y.1, y.2 y y.3 vs. vars explicativas tendrían la incertudumbre de cómo interpretar los ceros.

Antes que nada, calculemos el # de visitas y el # de detecciones realizadas por sitio. (los necesitaremos)

```
> visitas=rowSums(!is.na(DF[,c("y.1","y.2", "y.3")])) # Nb de visitas por sitio
> deteccion=rowSums(DF[,c("y.1","y.2","y.3")],na.rm=T) # Nb de detecciones por sitio
> visitas
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 2 3 3 3 3 3 3 3 3 3 3 3
 [51] 3 2 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3
[101] 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 2 3 3
[151] 3 3 2 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3
[201] 3 2 2 2 2 3 2 2 2 3 3 3 3 3 2 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
> deteccion
  [1] 0 0 0 0 0 0 0 0 2 1 0 0 0 0 1 2 0 3 0 1 0 0 3 0 0 0 2 3 3 0 0 0 0 0 0 2 3 3 0 0 3 0 0 0 3 0 3 0 3 0
 [51] 3 0 0 0 0 0 3 0 1 0 0 0 0 0 0 2 0 0 0 0 0 1 0 0 0 3 3 3 0 3 1 0 0 0 0 3 3 0 0 0 0 0 3 0 0 2 0 3 0 0
[101] 0 0 3 3 3 0 0 0 3 0 0 0 3 2 0 0 0 0 0 3 0 0 0 0 3 1 0 0 0 0 3 0 0 0 0 0 0 1 0 3 0 0 0 0 1 2 0 0 2 0
[151] 2 1 0 0 0 0 0 0 0 0 0 0 1 0 1 3 0 3 0 0 0 3 1 2 0 0 3 1 0 0 2 0 0 0 1 2 2 0 0 0 3 3 0 0 0 0 0 1 0 0
[201] 2 0 1 1 0 3 1 2 2 3 3 3 3 3 0 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

A título de ejemplo, en estos sitios se efectuaron 3 visitas (i.e. un máximo posible de 3 detecciones) pero nunca se observó ningún individuo: ¿cómo saber si son ceros reales?

¿Cómo saber si son ceros reales? ⟷ No podemos saberlo!

A partir de las detecciones por sitio,

```
> deteccion
  [1] 0 0 0 0 0 0 0 0 2 1 0 0 0 0 1 2 0 3 0 1 0 0 3 0 0 0 2 3 3 0 0 0 0 0 2 3 3 0 0 3 0 0 0 3 0 3 0 3 0
 [51] 3 0 0 0 0 0 3 0 1 0 0 0 0 0 2 0 0 0 0 0 1 0 0 0 3 3 3 0 3 1 0 0 0 0 3 3 0 0 0 0 0 3 0 0 2 0 3 0 0
[101] 0 0 3 3 3 0 0 0 3 0 0 0 3 2 0 0 0 0 0 3 0 0 0 0 3 1 0 0 0 0 3 0 0 0 0 0 0 1 0 3 0 0 0 0 1 2 0 0 2 0
[151] 2 1 0 0 0 0 0 0 0 0 0 0 1 0 1 3 0 3 0 0 0 3 1 2 0 0 3 1 0 0 2 0 0 0 1 2 2 0 0 0 3 3 0 0 0 0 0 1 0 0
[201] 2 0 1 1 0 3 1 2 2 3 3 3 3 3 0 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

creamos el vector ocup que será incluido en el modelo:

```
> ocup =  ifelse(detec > 0, 1, NA)
> ocup
  [1] NA NA NA NA NA NA NA NA  1  1 NA NA NA NA  1  1 NA  1 NA  1 NA NA  1 NA NA NA  1  1  1 NA NA NA
 [33] NA NA NA  1  1  1 NA NA  1 NA NA NA  1 NA  1 NA  1 NA  1 NA NA NA NA NA  1 NA  1 NA NA NA NA NA
 [65] NA  1 NA NA NA NA NA  1 NA NA NA  1  1  1 NA  1  1 NA NA NA NA  1  1 NA NA NA NA NA  1 NA NA  1
 [97] NA  1 NA NA NA NA  1  1  1 NA NA NA  1 NA NA NA  1  1 NA NA NA NA NA  1 NA NA NA NA  1  1 NA NA
[129] NA NA  1 NA NA NA NA NA NA  1 NA  1 NA NA NA NA  1  1 NA NA  1 NA  1  1 NA NA NA NA NA NA NA NA
[161] NA NA  1 NA  1  1 NA  1 NA NA NA  1  1  1 NA NA  1  1 NA NA  1 NA NA NA  1  1  1 NA NA NA  1  1
[193] NA NA NA NA NA  1 NA NA  1 NA  1  1 NA  1  1  1  1  1  1  1  1  1 NA  1  1 NA NA NA NA NA NA NA
[225] NA NA NA NA NA NA NA NA NA NA NA NA NA
```

Y_{obs} = ocup*p

ocup>0 → sitio realmente ocupado

ocup=NA → ignoramos su estado real.

p*occ[i]=0→ dbin(n[i],0)=NA

Modelo de ocupación para cada sitio i:

Pr. ocupación

Proceso: ocupación de sitio i:

Var. latente no observable

$$\text{ocup[i]} \sim \text{Bernoulli}(\varphi) \left\langle \begin{array}{l} 0 \\ 1 \end{array} \right.$$

con $\text{logit}(\varphi_i) = \beta_0 + \beta_1 \text{elev} + \beta_2 \text{elev}^2 + \beta_3 \text{forest}$

Observación: detección|ocupación en sitio i

$\text{p[i]} \sim \text{Binomial}(\pi, \text{visitas[i]})$ ← Pr. detección

se podría: $\text{logit}(\pi) = \beta_4 + \beta_5 \text{durac.visit} + \ldots$

En cada sitio, el # de inds. detectables depende del # visitas y ambos codeterminan la prob de detección p[i].

Los valores observables en sitio[i] Y_{obs} son el producto de la variable latente binaria ocup (el "ranef" para modelar la incertidumbre) y de la prob. de detección p: $Y_{obs} = \text{ocup} * \text{p}$.

En JAGS, un modelo se compone de:

* los datos
* la verosimilitud
* las distr previas
* las cantidades derivadas a calcular (opcional)

En JAGS los datos se entran como una lista:

```
> jagsData <- list(y = detec, visitas = visitas, nSites = length(visitas),
+                  ocup =  ifelse(detec > 0, 1, NA),
+                  forest = DF$forest, ele = DF$elev, ele2 = DF$elev*DF$elev)
```

```
> str(jagsData)
List of 7
 $ y      : num [1:237] 0 0 0 0 0 0 0 0 2 1 ...
 $ visitas: num [1:237] 3 3 3 3 3 3 3 3 3 3 ...
 $ nSites : int 237
 $ ocup   : num [1:237] NA NA NA NA NA NA NA NA 1 1 ...
 $ forest : num [1:237] -1.15389 -0.50052 -0.10124 0.00766 -1.19019 ...
 $ ele    : num [1:237] -1.18 -1.133 -0.205 -0.112 -1.041 ...
 $ ele2   : num [1:237] 1.392 1.2847 0.0421 0.0126 1.0828 ...
```

Es más simple crear los pedazos y luego juntarlos en la lista.

Just Checking

Verosimilitud para cada sitio[i]

```
for(i in 1:nSites) {
   logit(pres[i]) <-b0+bFor*forest[i]+bElev*ele[i]+ bElev2 * ele2[i]
   ocup[i] ~ dbern(pres[i])
   Y[i] ~ dbin(p * ocup[i],visitas[i])
 }
```

Y_{obs} = p*ocup

Recordemos:

ocup[i]={0, 1}

ocup[i]~Bernoulli (φ) φ≡pres[i]: prob. de ocupación REAL en función Xs

p~ Binomial(π,visitas[i]) π: prob. de detección

Distribuciones previas: ¿cuáles son los parámetros del modelo que conjuntamente predicen Y_{obs}?

b0, bFor, bElev, bElev2, p cuyas distr previas hay que especificar.

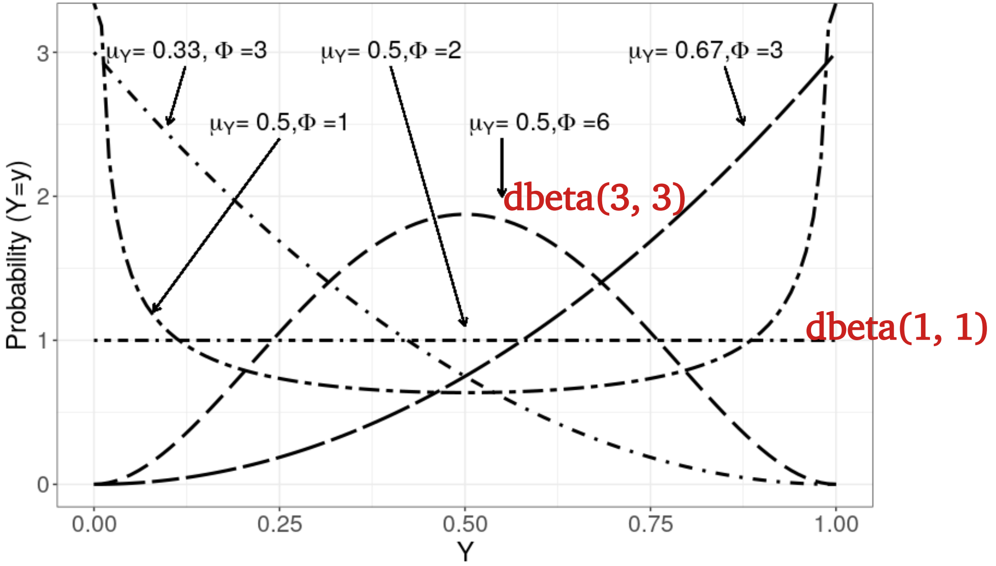
Distribuciones previas: b0, bFor, bElev, bElev2, p

b0 ~ dnorm(0, 0.5)
bFor ~ dnorm(0, 0.5)
bElev ~ dnorm(0, 0.5)
bElev2 ~ dnorm(0, 0.5)
p ~ dbeta(3, 3)

Detalle técnico MALIGNO:
En JAGS Normal(mean=0, sd=10) se escribe dnorm(mean=0,tau=$1/10^2$).

Cantidades derivadas a calcular (opcional):

N <- sum(ocup)
Abundancia global a partir de reales ocupaciones locales



Poniendo todo junto:

```
jagsData <- list(y = detec, visitas = visitas, nSites = length(visitas),
                 ocup =  ifelse(detec > 0, 1, NA),
                 forest = DF$forest, ele = DF$elev, ele2 = DF$elev*DF$elev)

m1="model{
  for(i in 1:nSites) {
    logit(pres[i]) <- b0 + bFor * forest[i] +
                      bElev * ele[i] + bElev2 * ele2[i]
    ocup[i] ~ dbern(pres[i])
    Y[i] ~ dbin(p * ocup[i],visitas[i])
  }

  # Previas
  b0 ~ dnorm(0, 0.5) # intercepto
  bFor ~ dnorm(0, 0.5)    # pendiente de forest
  bElev ~ dnorm(0, 0.5)   # pendiente de elevation
  bElev2 ~ dnorm(0, 0.5)  # pendiente de elevation^2
  p ~ dbeta(3, 3) # prob de observacion

  # variable calculada:
  N <- sum(ocup) #abundancia global a partir de ocupaciones locales
}"
```

m1 está entre comillas: es solo texto.

```
> dim(DF)
[1] 237  15
```

237 sitios.

```
out.m1=run.jags(data=jagsData,
                monitor=c("p", "b0", "bFor", "bElev", "bElev2","N", "deviance"),
                model=m1, n.chains=3, thin=5, sample=5000, burnin = 100, method="rjparallel")
```

Se descartan los primeros 100 pasos, hay 3 cadenas de 5000 pasos en las que se muestrea cada quinto valor, y se “paraleliza”

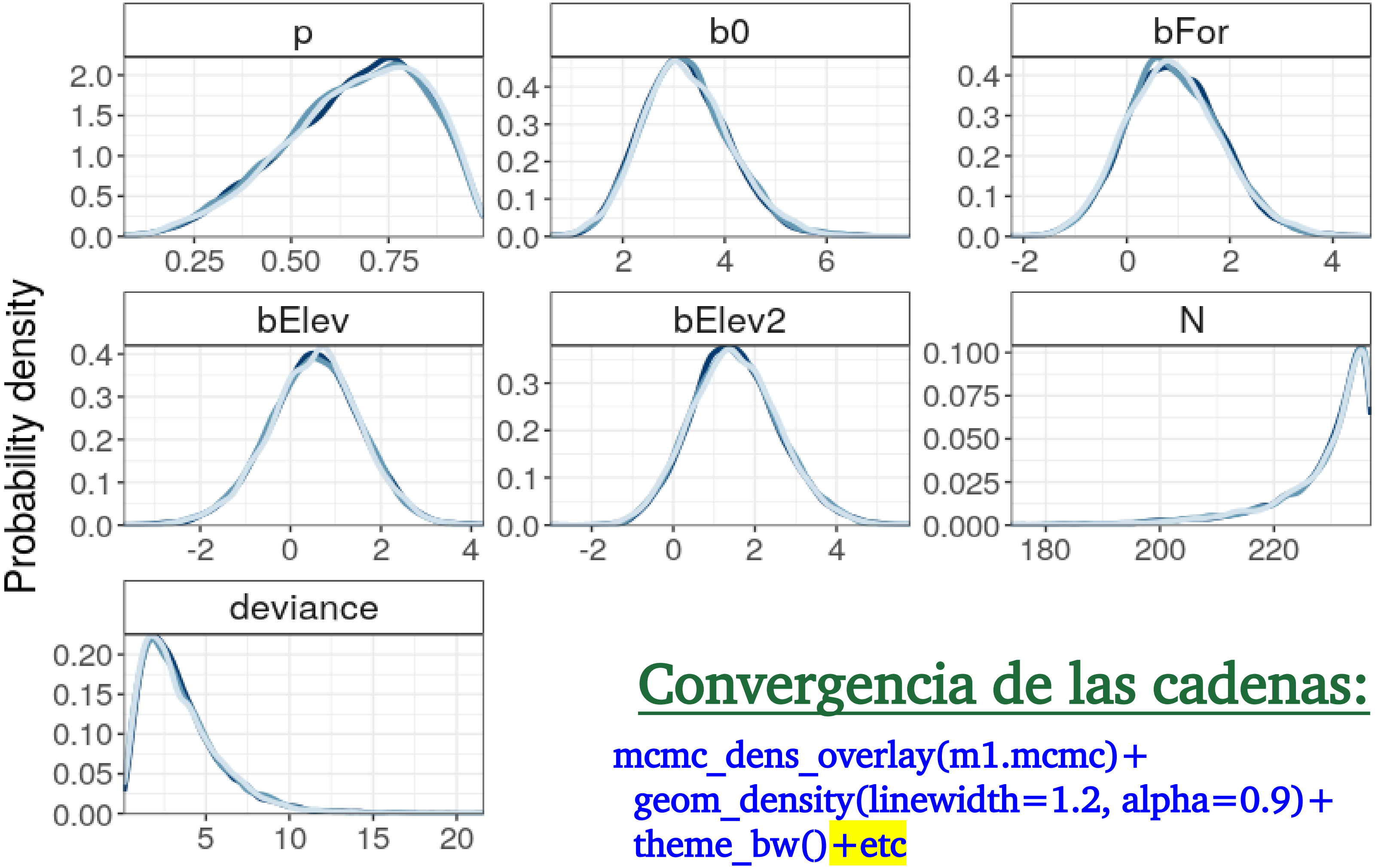
```
> summary(out.m1)
          Lower95  Median Upper95    Mean    SD    Mode   MCerr MC%ofSD SSeff     AC.50 psrf
p           0.152   0.497   0.858   0.497 0.188   0.485 0.00154     0.8 15000 -0.021365    1
b0          1.650   3.220   4.975   3.264 0.863   3.120 0.00728     0.8 14068 -0.015644    1
bFor       -0.815   0.859   2.785   0.898 0.914   0.767 0.00769     0.8 14120  0.005386    1
bElev      -1.547   0.542   2.476   0.536 1.025   0.539 0.00841     0.8 14852  0.000181    1
bElev2     -0.559   1.471   3.602   1.517 1.076   1.328 0.00883     0.8 14865 -0.011292    1
N         210.000 232.000 237.000 229.188 8.870 236.000 0.07318     0.8 14691 -0.017621    1
deviance    0.203   2.835   8.172   3.408 2.358   2.042 0.01925     0.8 15000 -0.016487    1
```

R.hat <1.1

MCerr: SD/sqrt(n) SSeff: effective sample size (ajustado por autocor).

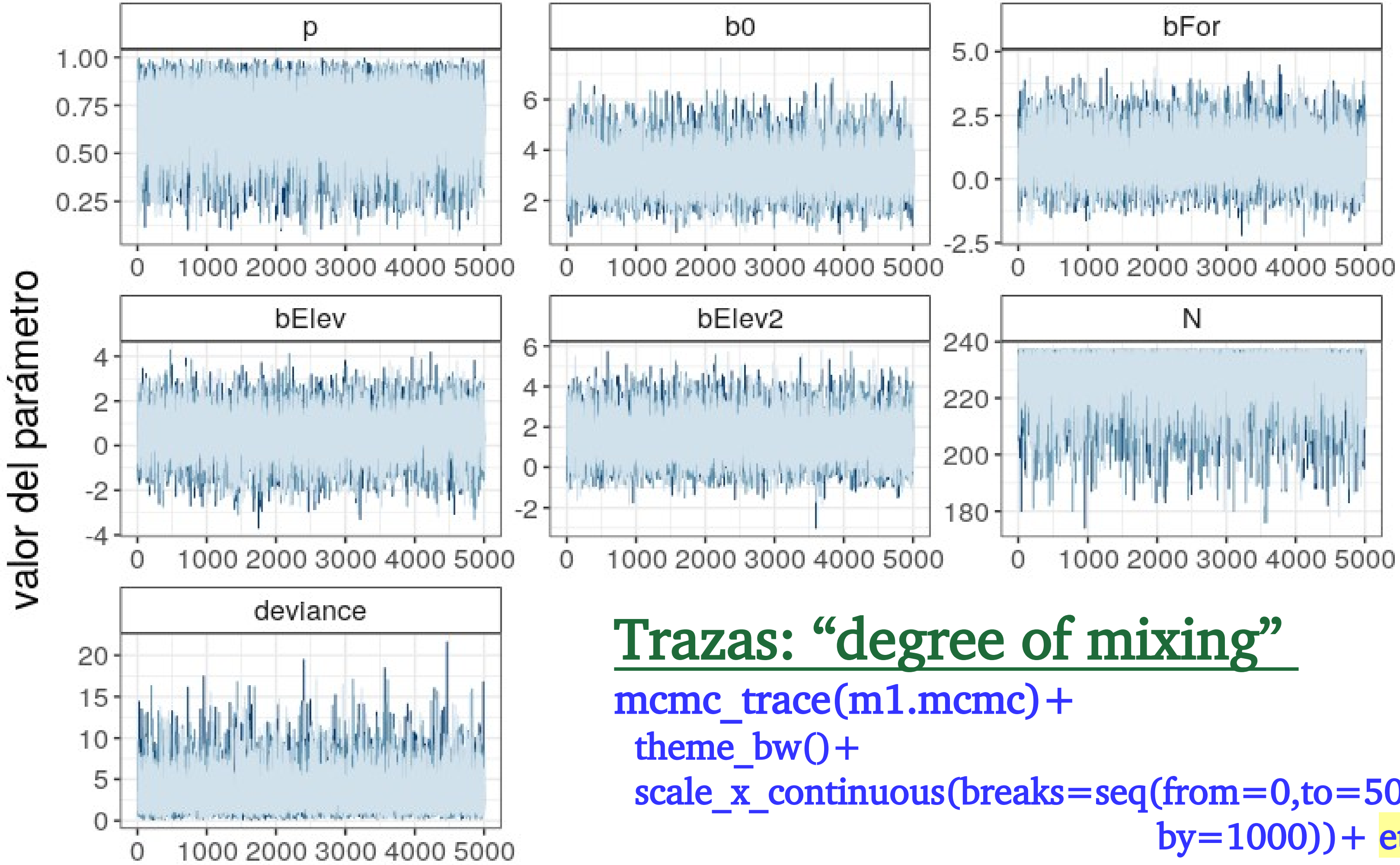
Hay que convertir el output de JAGS en objeto tipo MCMC

m1.mcmc <- as.mcmc.list(out.m1) para hacer gráficos.



Convergencia de las cadenas:

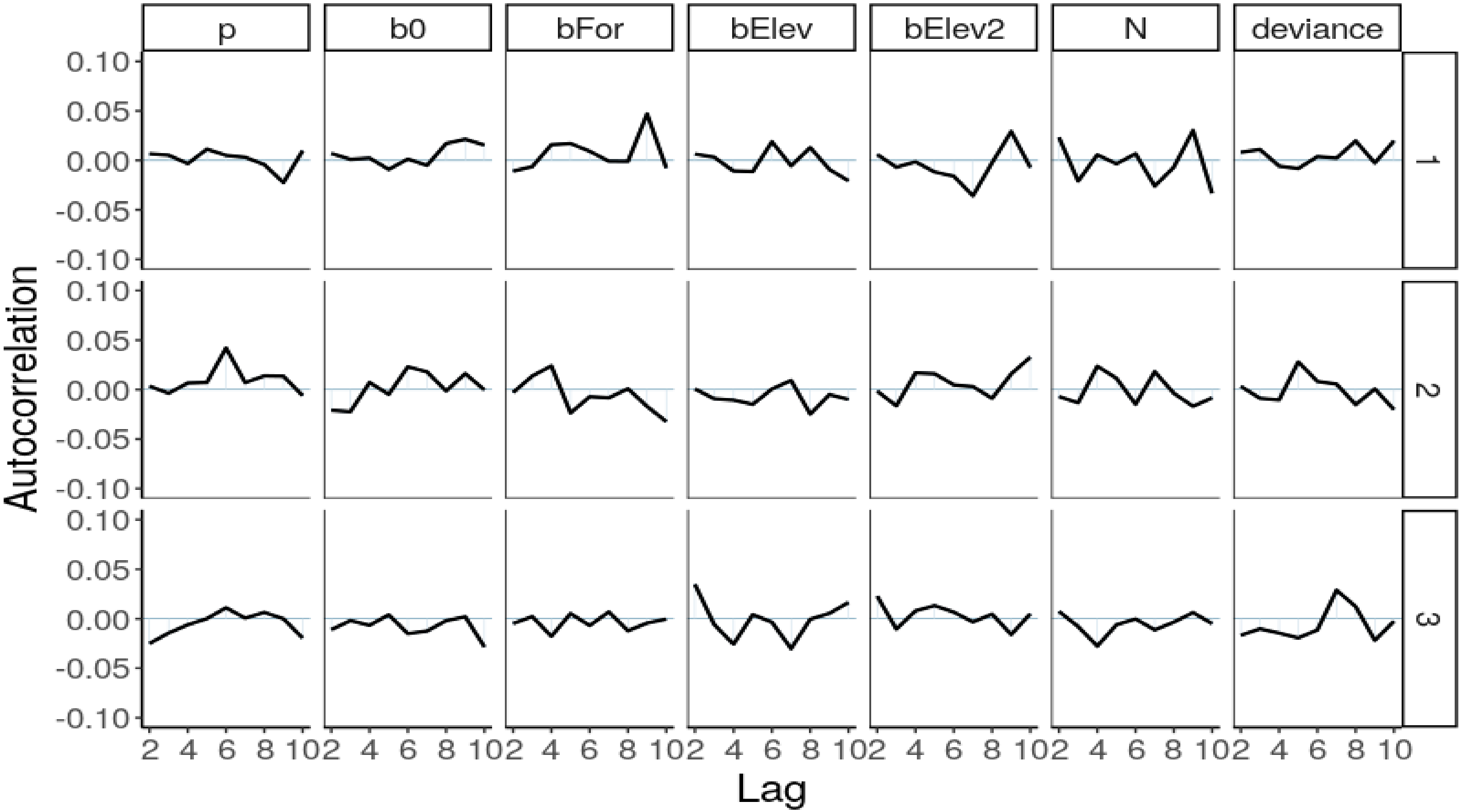
```
mcmc_dens_overlay(m1.mcmc)+
  geom_density(linewidth=1.2, alpha=0.9)+
  theme_bw()+etc
```



Trazas: “degree of mixing”

```
mcmc_trace(m1.mcmc)+
  theme_bw()+
  scale_x_continuous(breaks=seq(from=0,to=5000,
                                         by=1000))+ etc
```

Autocorrelación de los estimados:

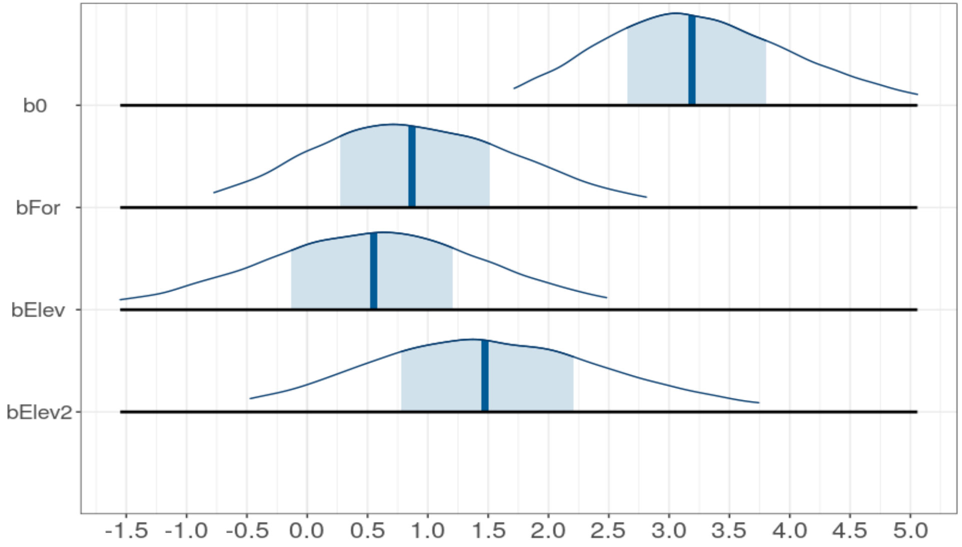



```
mcmc_acf(m1.mcmc)+
geom_line(size=0.8)+
scale_x_continuous(limits=c(2,10))+
scale_y_continuous(limits=c(-0.1,0.1))+etc
```

Interpretación de los parámetros:

```
> summary(out.m1)
         Lower95  Median Upper95    Mean    SD    Mode
p          0.152   0.497   0.858   0.497 0.188   0.485
b0         1.650   3.220   4.975   3.264 0.863   3.120
bFor      -0.815   0.859   2.785   0.898 0.914   0.767
bElev     -1.547   0.542   2.476   0.536 1.025   0.539
bElev2    -0.559   1.471   3.602   1.517 1.076   1.328
N        210.000 232.000 237.000 229.188 8.870 236.000
deviance   0.203   2.835   8.172   3.408 2.358   2.042
```

Como en cualquier GLM binomial
(usando la "regla de 4")



mcmc_areas(m1.mcmc, prob_outer = 0.95,
regex_pars = c("b"))+
geom_density(linewidth=1, alpha=0.9)+
etc

Con regex_pars = c("^p") se obtiene la distr posterior de la prob. de observación[i].

Distribución predictiva posterior:

```
m1="model{
  for(i in 1:nSites) {
    logit(pres[i]) <- b0 + bFor * forest[i] +
                      bElev * ele[i] + bElev2 * ele2[i]
    ocup[i] ~ dbern(pres[i])
    Y[i] ~ dbin(p * ocup[i],visitas[i])
  }
```

Dos formas de hacerlo:

1) Incluyendo "Y" en

```
out.m1=run.jags(data=jagsData,
                monitor=c("p", "b0", "bFor", "bElev", "bElev2","N", "deviance"),
                model=m1, n.chains=3, thin=5, sample=5000, burnin = 100, method="rjparallel")
```

En out.m1 habría que separar la parte de los distr posteriores de los parámetros de la distr predictiva posterior de Y.

2) Usando:

```
DPP.m1=run.jags(data=jagsData,monitor=c("Y"),model=m1, n.chains=3, thin=5,
                sample=5000, burnin = 100, method="rjparallel")
```

DPP.m1 es una lista con sólo las distr posteriores de cada dato.

Convirtiendo DPP.m1 en un dataframe:

```
DPP=as.data.frame(unlist(DPP.m1$mcmc [[1]]))
```

```
> dim(DPP)
[1] 5000  237
```

```
> dim(DF)
[1] 237  15
```

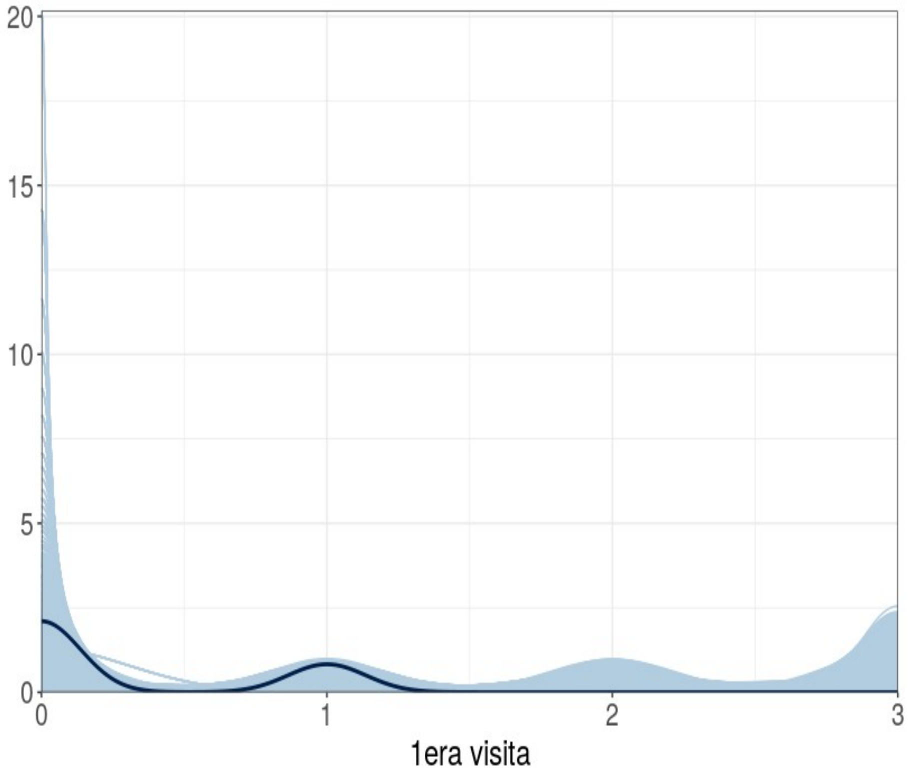
Hay "tres variables de respuesta binarias observadas" y.1, y.2 y y.3 con un % de ceros que son "falsas ausencias" lo que complejiza la comparación de la distr. pred. post. con los datos.

La distr. pred. post. contiene 5000 realizaciones en los 237 sitios con el número de "detecciones" en las (a veces) 3 visitas.

```
> summary(DPP)
      Y[1]            Y[2]           Y[3]            Y[4]            Y[5]            Y[6]
 Min.   :0.00   Min.   :0.0   Min.   :0.00   Min.   :0.00   Min.   :0.00   Min.   :0.00
 1st Qu.:0.00   1st Qu.:0.0   1st Qu.:0.00   1st Qu.:0.00   1st Qu.:0.00   1st Qu.:0.00
 Median :1.00   Median :1.0   Median :1.00   Median :1.00   Median :1.00   Median :1.00
 Mean   :1.35   Mean   :1.4   Mean   :1.42   Mean   :1.41   Mean   :1.31   Mean   :1.45
```

```
> summary(DF[, c("y.1", "y.2", "y.3")])
      y.1              y.2             y.3
 Min.   :0.000   Min.   :0.00   Min.   :0.0
 1st Qu.:0.000   1st Qu.:0.00   1st Qu.:0.0
 Median :0.000   Median :0.00   Median :0.0
 Mean   :0.283   Mean   :0.26   Mean   :0.3
 3rd Qu.:1.000   3rd Qu.:1.00   3rd Qu.:1.0
 Max.   :1.000   Max.   :1.00   Max.   :1.0
                 NA's   :2      NA's   :44
```

Veamos igual distr. pred. post. con la 1era visita.



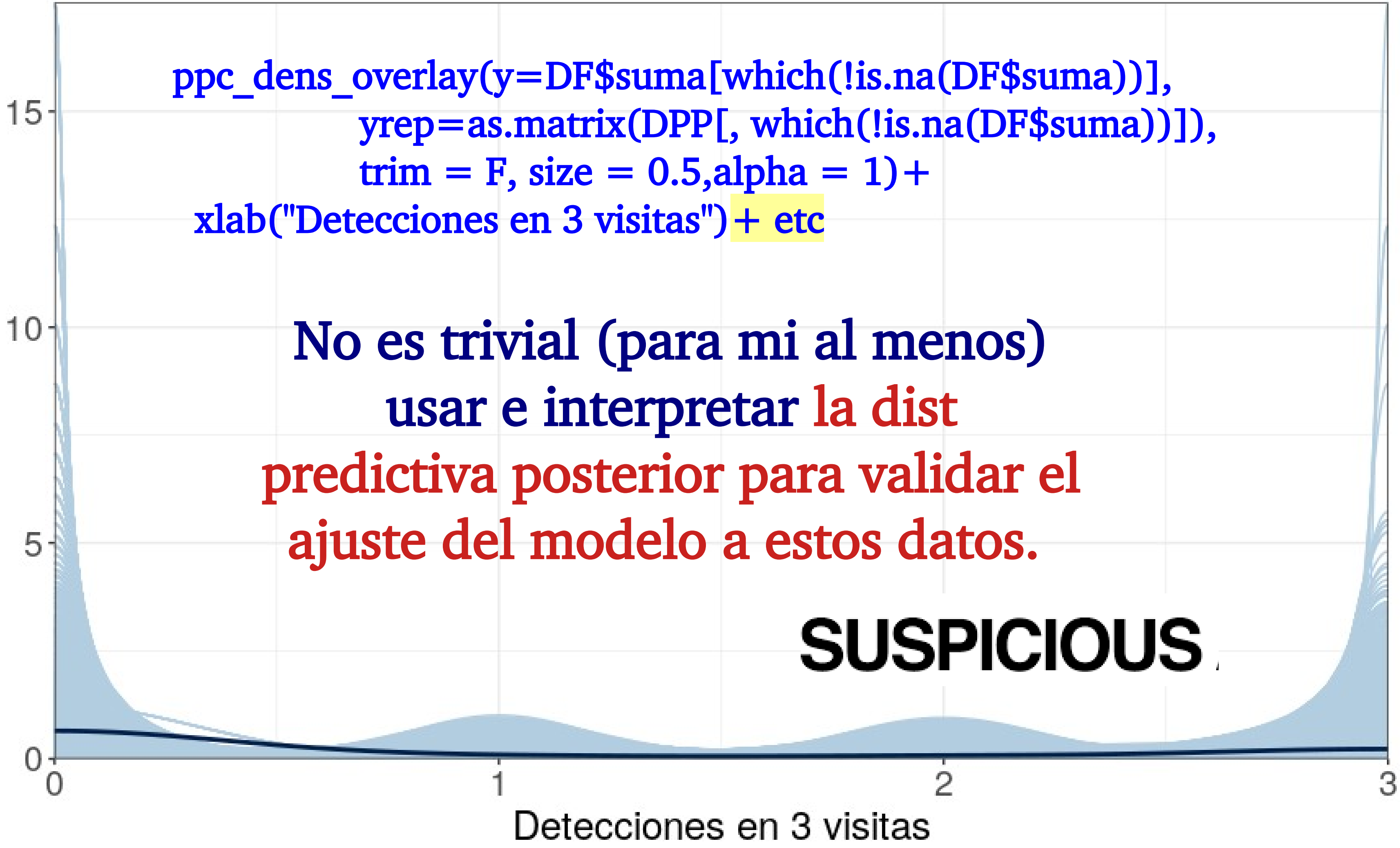
ppc_dens_overlay(y=DF$y.1,yrep=as.matrix(DPP),
trim = F, size = 0.5,alpha = 1)+
xlab("1era visita")+ etc

A efectos de una mejor comparación, voy a crear la suma de DF$y.1, DF$y.2 y DF$y.3 que contiene NA

DF$suma=DF$y.1+DF$y.2+DF$y.3

```
> which(is.na(DF$suma))
 [1]  36  39  52  63  70  71  85  89 102 126 145 148 153 165 181 197
[17] 202 203 204 205 207 208 209 215 218 219 220 221 222 223 224 225
[33] 226 227 228 229 230 231 232 233 234 235 236 237
```

Hay que excluir estas filas de DPP para no "comparar con un NA"



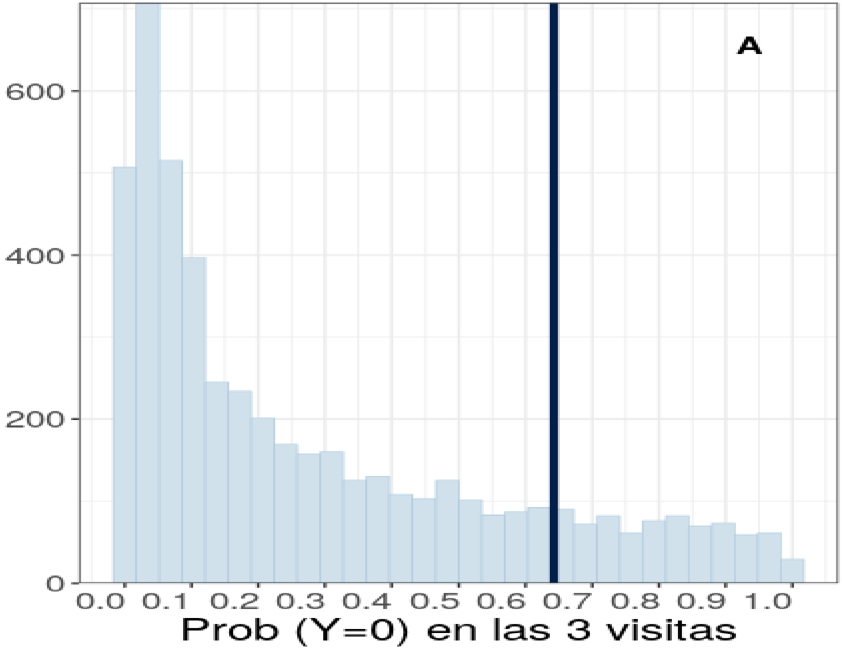

```
ppc_dens_overlay(y=DF$suma[which(!is.na(DF$suma))],
                 yrep=as.matrix(DPP[, which(!is.na(DF$suma))]),
                 trim = F, size = 0.5,alpha = 1)+
  xlab("Detecciones en 3 visitas")+ etc
```

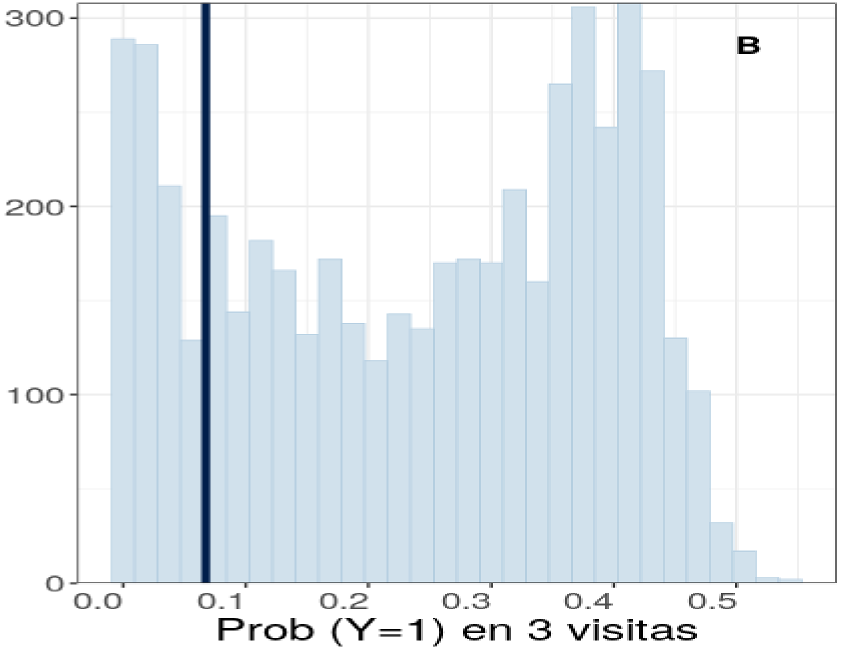
No es trivial (para mi al menos) usar e interpretar la dist predictiva posterior para validar el ajuste del modelo a estos datos.

SUSPICIOUS

Mis dos últimos intentos:

```
cero=function (x){sum(x==0)/length(x)} # prop de ceros
unos=function (x){sum(x==1)/length(x)} # prop de unos
```






```
ppc_stat(y=DF$suma[which(!is.na(DF$suma))],
        yrep=as.matrix(DPP[, which(!is.na(DF$suma))]),stat = cero)+etc
```

e. Binomial-mixture models: un ejemplo

Los datos: 4 conteos de reinita hornera (nidifica en el suelo) en áreas de 100m de radio (Royle 2004). Se registró cobertura foliar sotobosque (ufc) y área basal de árboles grandes (ba) en 70 sitios.

```
> DF1=read.csv(file="ReinitaHornera Teo05.csv", header=T)
> DF1=DF1[,-1]
> summary(DF1)
      y.1              y.2               y.3               y.4              ufc                trba
 Min.   :0.0    Min.   :0.000    Min.   :0.000    Min.   :0.0    Min.   :-1.471    Min.   :-2.010
 1st Qu.:0.0    1st Qu.:0.000    1st Qu.:0.000    1st Qu.:0.0    1st Qu.:-0.741    1st Qu.:-0.693
 Median :1.0    Median :0.000    Median :0.000    Median :0.0    Median :-0.253    Median :-0.129
 Mean   :0.7    Mean   :0.229    Mean   :0.071    Mean   :0.1    Mean   : 0.000    Mean   : 0.000
 3rd Qu.:1.0    3rd Qu.:0.000    3rd Qu.:0.000    3rd Qu.:0.0    3rd Qu.: 0.984    3rd Qu.: 0.718
 Max.   :3.0    Max.   :2.000    Max.   :1.000    Max.   :2.0    Max.   : 2.344    Max.   : 2.881
```

Objetivo: estimar la abundancia global de la especie y los determinantes de su detectabilidad local.

```
> summary(DF1)
     sitio          y.1             y.2              y.3              y.4            ufc                trba
 1      : 1   Min.   :0.0   Min.   :0.000   Min.   :0.000   Min.   :0.0   Min.   :-1.471   Min.   :-2.010
 2      : 1   1st Qu.:0.0   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.0   1st Qu.:-0.741   1st Qu.:-0.693
 3      : 1   Median :1.0   Median :0.000   Median :0.000   Median :0.0   Median :-0.253   Median :-0.129
 4      : 1   Mean   :0.7   Mean   :0.229   Mean   :0.071   Mean   :0.1   Mean   : 0.000   Mean   : 0.000
 5      : 1   3rd Qu.:1.0   3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.:0.0   3rd Qu.: 0.984   3rd Qu.: 0.718
 6      : 1   Max.   :3.0   Max.   :2.000   Max.   :1.000   Max.   :2.0   Max.   : 2.344   Max.   : 2.881
 (Other):64
```

```
> ftable(table(DF1$y.1, DF1$y.2, DF1$y.3))
      0  1

0 0  28  0
  1   6  0
  2   0  0
1 0  16  2
  1   6  0
  2   1  0
2 0   4  3
  1   2  0
  2   0  0
3 0   2  0
  1   0  0
  2   0  0
```

Los datos:

El análisis exploratorio de estos datos tiene la misma incertidumbre acerca de la interpretación de los ceros que los datos del occupancy model

Hay una combinación multiplicativa entre la variable latente (C~Poisson) y la prob de observación p~Bi.

El modelo estadístico:

Proceso de estado: $Conteo[i] \sim Poisson(\mu_C)$ variable latente

Proceso de observación: $Y_{obs} | Conteo \sim Binomial(Conteo[i], \pi)$

El 1er modelo
(sin covariables explicativas):

Los datos:

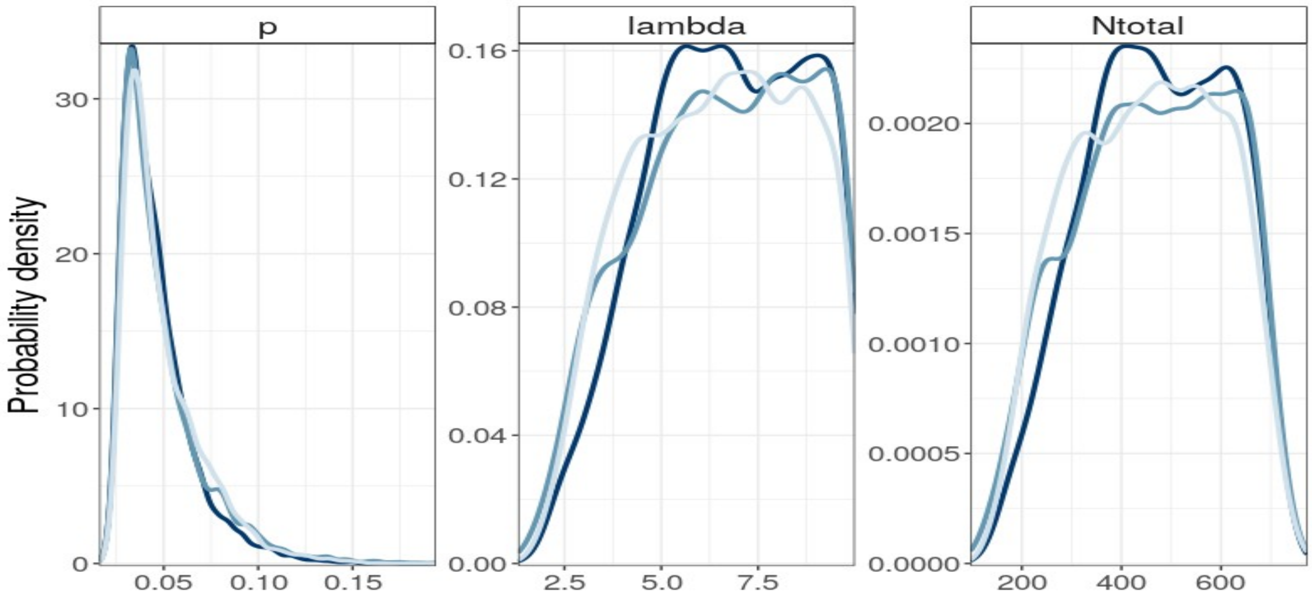
```
JAGSdata1<-list(nSites=70, nOcc=4,
                Y=as.matrix(DF1[,c("y.1","y.2","y.3","y.4")]))
```

El modelo:

```
Bimix="model{
          for(i in 1:nSites) {
               # modelo de abundancia real (no observable)
               Conteo[i] ~ dpois(lambda)
               # modelo de observación
               for(j in 1:nOcc) {
                  Y[i,j] ~ dbin(p, Conteo[i])
               }
             }
  # Previas
  lambda ~ dunif(0, 10) # media de dist Poisson
  p ~ dbeta(1,1) # probabilidad de deteccion
  # variable derivada
  Ntotal <- sum(Conteo) # abundancia global
}"
```

El 1er modelo:

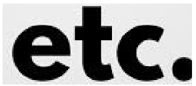
```
m2.out=run.jags(data=JAGSdata1,model=Bimix, monitor=c("p","lambda","Ntotal"),
               n.chains=3, thin=5, sample=5000, burnin = 100, method="rjparallel")
> summary(m2.out)
         Lower95   Median  Upper95     Mean     SD     Mode    MCerr MC%ofSD SSeff AC.50 psrf
p         0.0227   0.0423   0.0894   0.0481   0.02   0.0359 0.000899     4.5   497 0.481 1.02
lambda    3.1694   6.5846   9.9982   6.5359   2.03   7.0528 0.091286     4.5   493 0.526 1.02
Ntotal  200.0000 460.0000 695.0000 457.6445 142.43 357.0000 6.495729     4.6   481 0.523 1.01
```



m2.mcmc=as.mcmc.list(m2.out)

Convergencia de las cadenas:

mcmc_dens_overlay(m2.mcmc)+
geom_density(linewidth=1.2,
alpha=0.9)+
theme_bw()+etc



2do modelo: con vars explicativas en prob. detección [i]

$$\mathrm{logit}(\pi)=\beta_0+\beta_1 ufc+\beta_{12} trba$$

$$p=\frac{\exp^{X\beta}}{1+\exp^{X\beta}}=\frac{1}{1+\exp^{-X\beta}}$$

```
BiMix2="model{
  for(i in 1:nSites) {
    # modelo de abundancia real (no observable)
    Conteo[i] ~ dpois(lambda)
    #probabilidad de deteccion
    p[i]<- 1/(1+exp(-1*(b0 + b.ufc*ufc[i] + b.trba*trba[i])))
    for(j in 1:nOcc) {
      Y[i,j] ~ dbin(p[i], Conteo[i])
    }
  }
  # Distribuciones previas
  lambda ~ dunif(0, 10) # media de dist Poisson
  b0 ~dnorm(0,0.5) # previa intercepto
  b.ufc~dnorm(0,0.5)  # previa pendiente parcial
  b.trba~dnorm(0,0.5) # previa pendiente parcial
  # variable derivada
  Ntotal <- sum(Conteo)# abundancia global
}"
```

Los datos:

```
JAGSdata2 <-list(nSites=nrow(DF1), nOcc=4,
             Y=as.matrix(DF1[,c("y.1","y.2", "y.3", "y.4")]),
             ufc = DF1$ufc, trba = DF1$trba)
```

El modelo ajustado:

```
m3.out=run.jags(data=JAGSdata2,model=BiMix2,
                monitor=c("b0","b.trba","lambda","b.ufc","Ntotal","p"),
                n.chains=3, thin=5, sample=5000, burnin = 100, method="rjparallel")
```

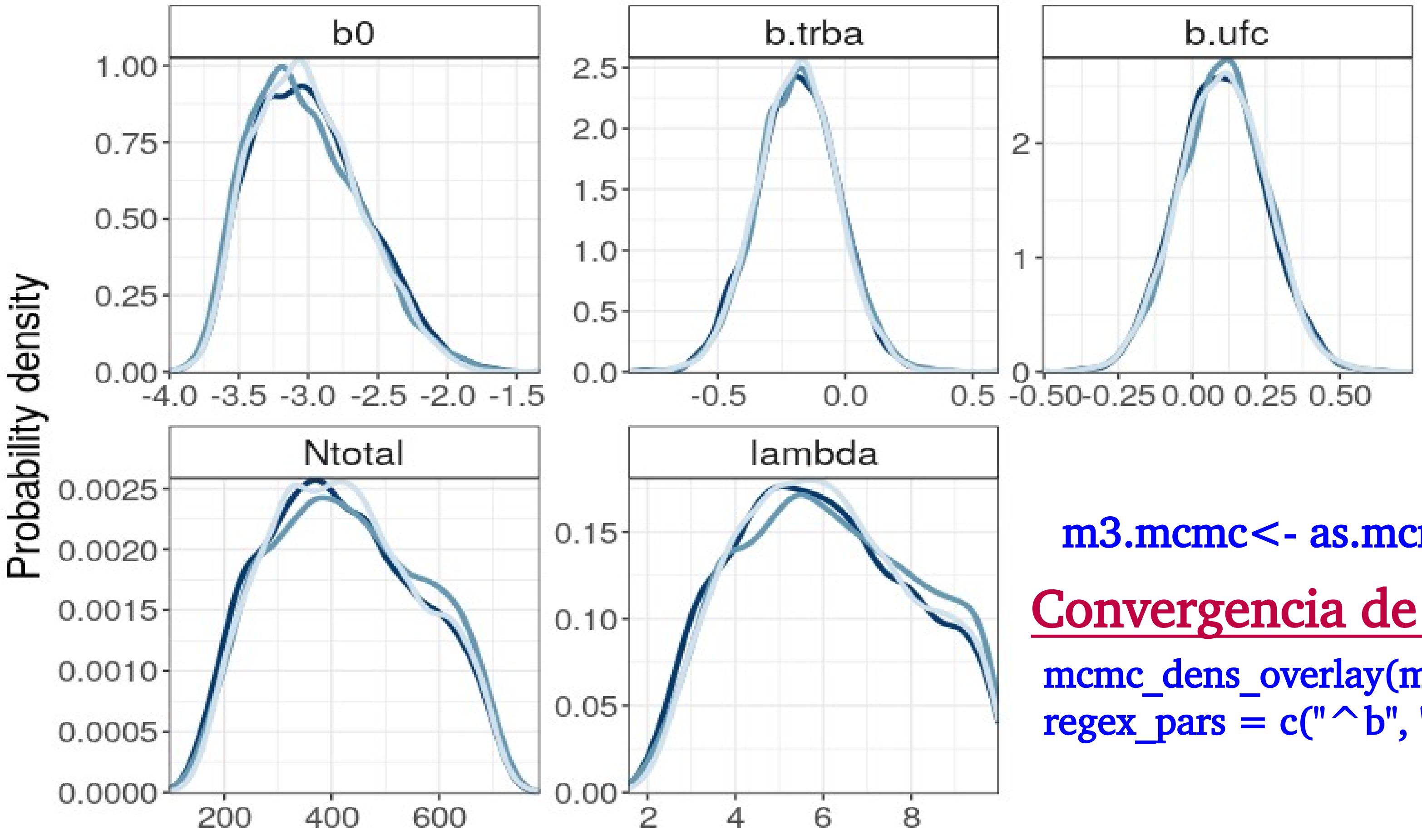
```
> summary(m3.out)
Calculating summary statistics...
Calculating the Gelman-Rubin statistic for 75 variables....
Note: Unable to calculate the multivariate psrf
         Lower95   Median  Upper95     Mean       SD     Mode    MCerr MC%ofSD SSeff     AC.50 psrf
b0      -3.68493  -3.0403  -2.2403  -3.0023   0.3935  -3.1430 0.017652     4.5   497  0.513042    1
b.trba  -0.50506  -0.1857   0.1155  -0.1867   0.1587  -0.1892 0.001394     0.9 12951  0.000574    1
lambda   2.87002   5.9610   9.8594   6.0641   1.9825   5.4972 0.093643     4.7   448  0.547628    1
b.ufc   -0.19404   0.1038   0.3846   0.1036   0.1482   0.1076 0.001290     0.9 13210 -0.000605    1
Ntotal 194.00000 416.0000 689.0000 424.4392 139.2070 458.0000 6.593210     4.7   446  0.546200    1
p[1]     0.02728   0.0650   0.1400   0.0730   0.0326   0.0537 0.001319     4.0   612  0.400345    1
p[2]     0.02368   0.0573   0.1231   0.0643   0.0288   0.0477 0.001184     4.1   590  0.408479    1
p[3]     0.02349   0.0618   0.1389   0.0698   0.0336   0.0503 0.001196     3.6   791  0.327359    1
```

(hay 70 sitios)

Resta examinar la convergencia de las cadenas, la autocorrelación de los parámetros y estimados, así como interpretar los parámetros del GLM binomial ("regla de 4").

Convierte el output de JAGS en objeto tipo MCMC

```
m3.mcmc <- as.mcmc.list(m3.out)
```

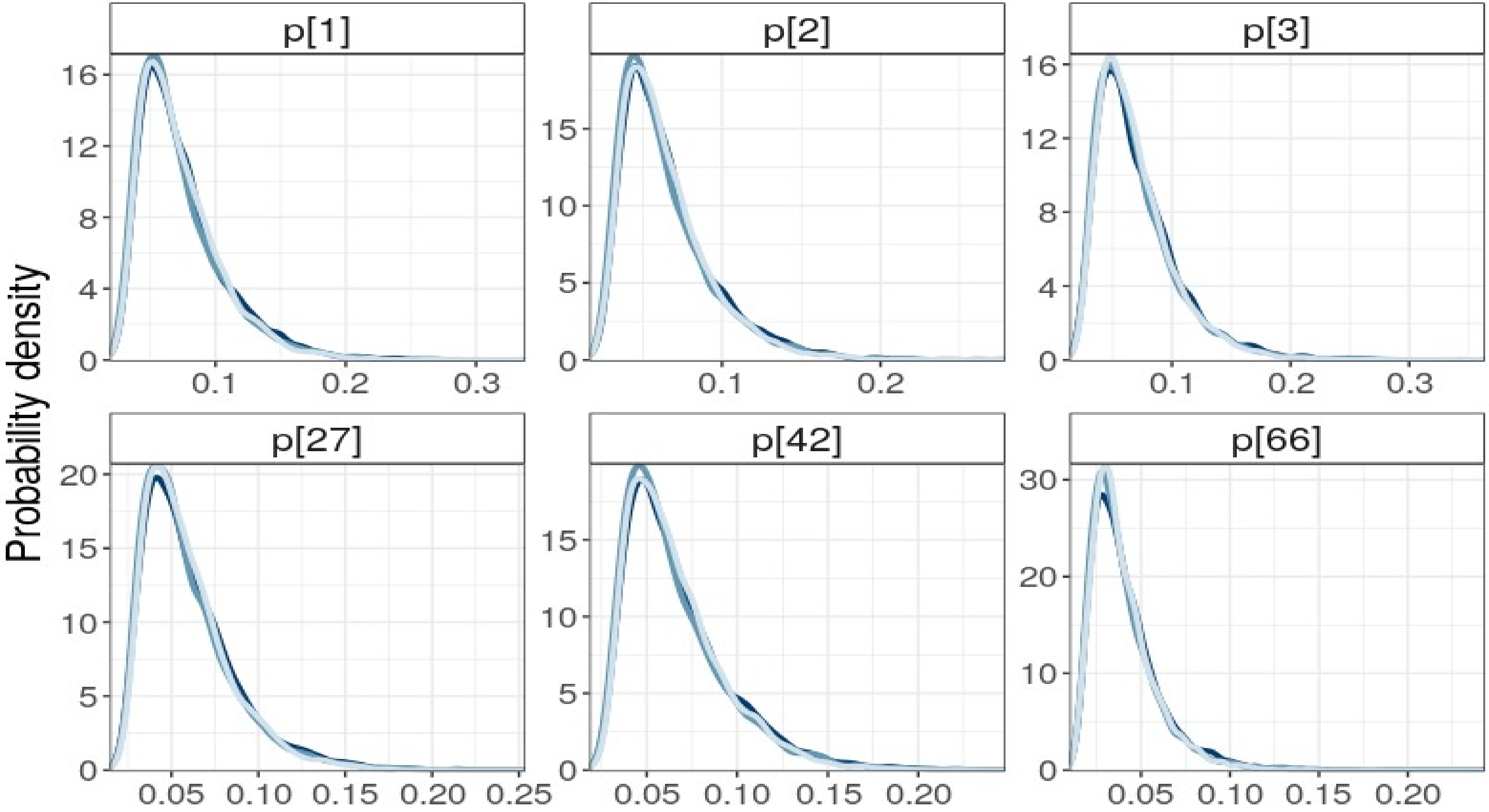


m3.mcmc<- as.mcmc.list(m3.out)

Convergencia de las cadenas:

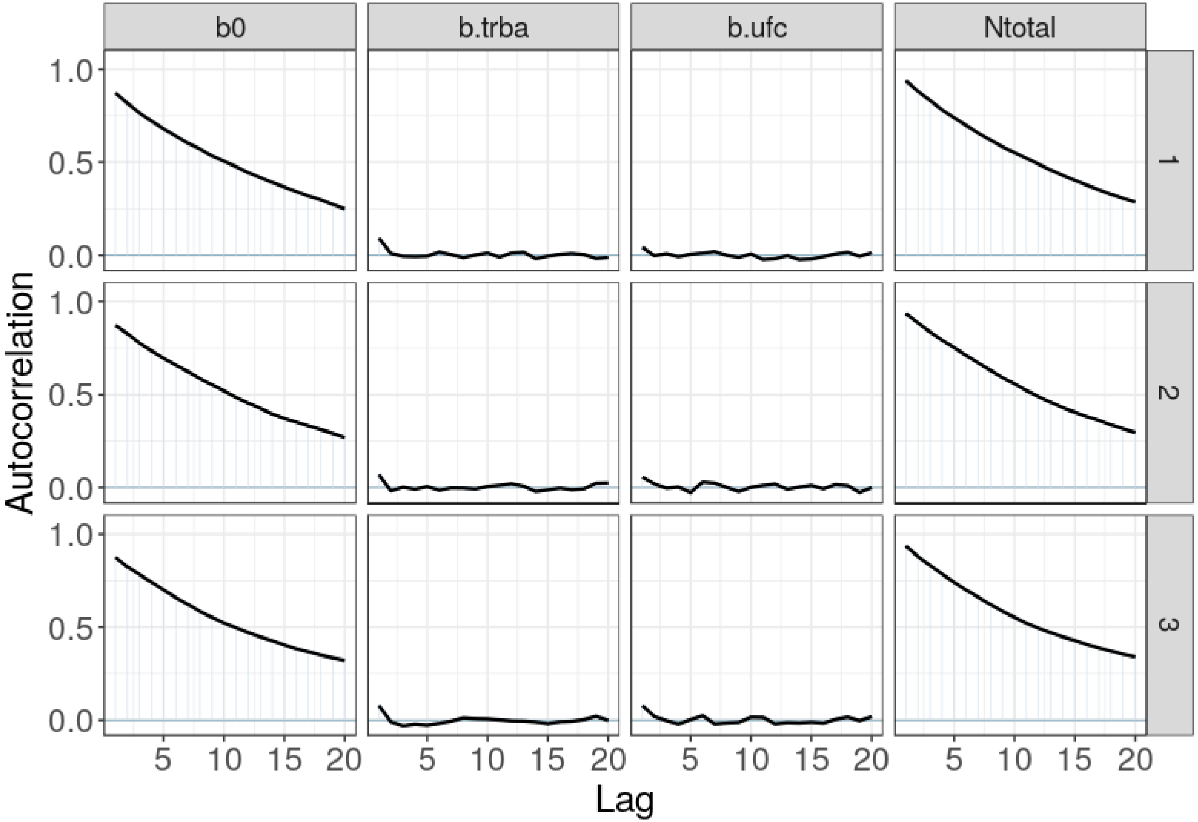
mcmc_dens_overlay(m3.mcmc,
regex_pars = c("^b", "Ntotal"))+etc

```
mcmc_dens_overlay(m3.mcmc, pars = c("p[1]","p[2]","p[3]","p[27]","p[42]","p[66]"))+
geom_density(lwd=1.2, alpha=0.9)+ etc
```



Autocorrelación de los estimados:

mcmc_acf(m3.mcmc, regex_pars = c("^b", "Ntotal")) +etc



Aumentar largo de cadenas o el thin...

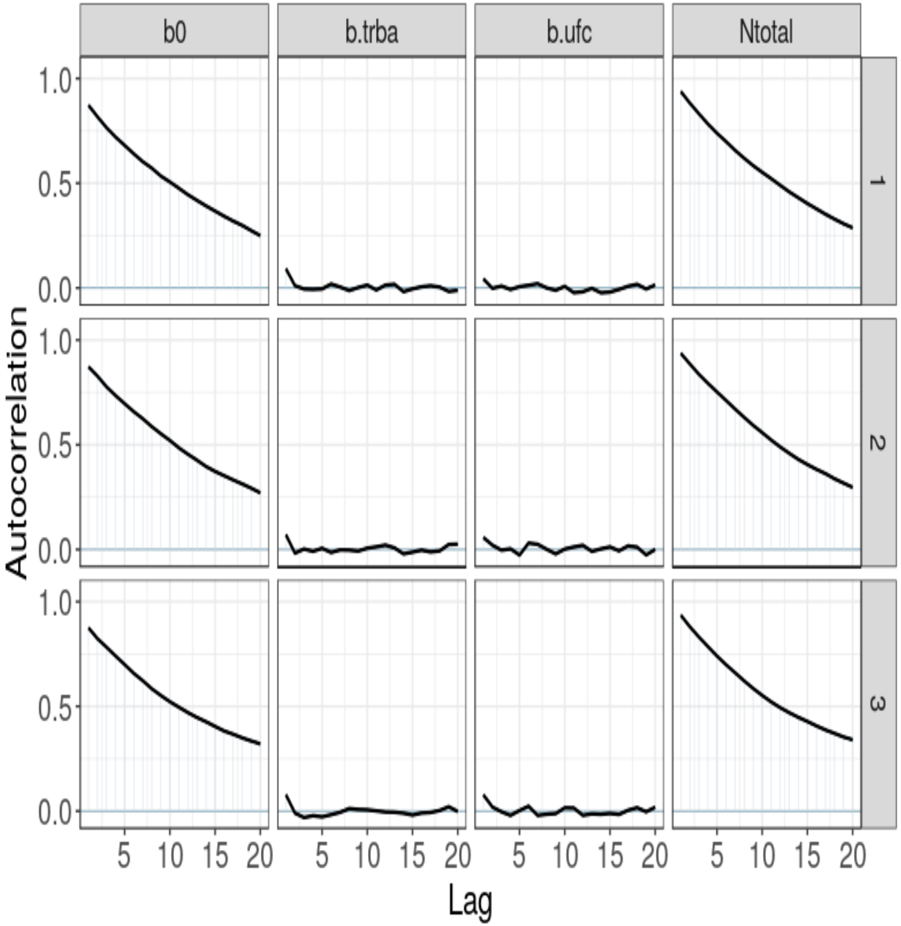
```
> summary(m3.out)
Calculating summary statistics...
Calculating the Gelman-Rubin statistic for 75 variables....
Note: Unable to calculate the multivariate psrf
          Lower95    Median  Upper95     Mean       SD     Mode     MCerr MC%ofSD SSeff   AC.50 psrf
b0       -3.65268   -3.0356  -2.1787  -2.9888   0.3994  -3.1455 0.018459     4.6   468 0.51629 1.01
b.trba   -0.50330   -0.1820   0.1181  -0.1832   0.1574  -0.1938 0.001340     0.9 13790 0.00855 1.00
lambda    2.56269    5.9049   9.6568   5.9946   2.0060   5.4406 0.096489     4.8   432 0.55587 1.01
b.ufc    -0.19235    0.1034   0.3910   0.1043   0.1492   0.1016 0.001311     0.9 12939 0.00829 1.00
Ntotal  177.00000  412.0000 674.0000 419.6171 140.5577 417.0000 6.705053     4.8   439 0.55413 1.01
p[1]      0.02723    0.0656   0.1417   0.0738   0.0333   0.0553 0.001355     4.1   604 0.40866 1.00
p[2]      0.02427    0.0575   0.1256   0.0649   0.0292   0.0490 0.001174     4.0   617 0.41743 1.00
```

thin=5

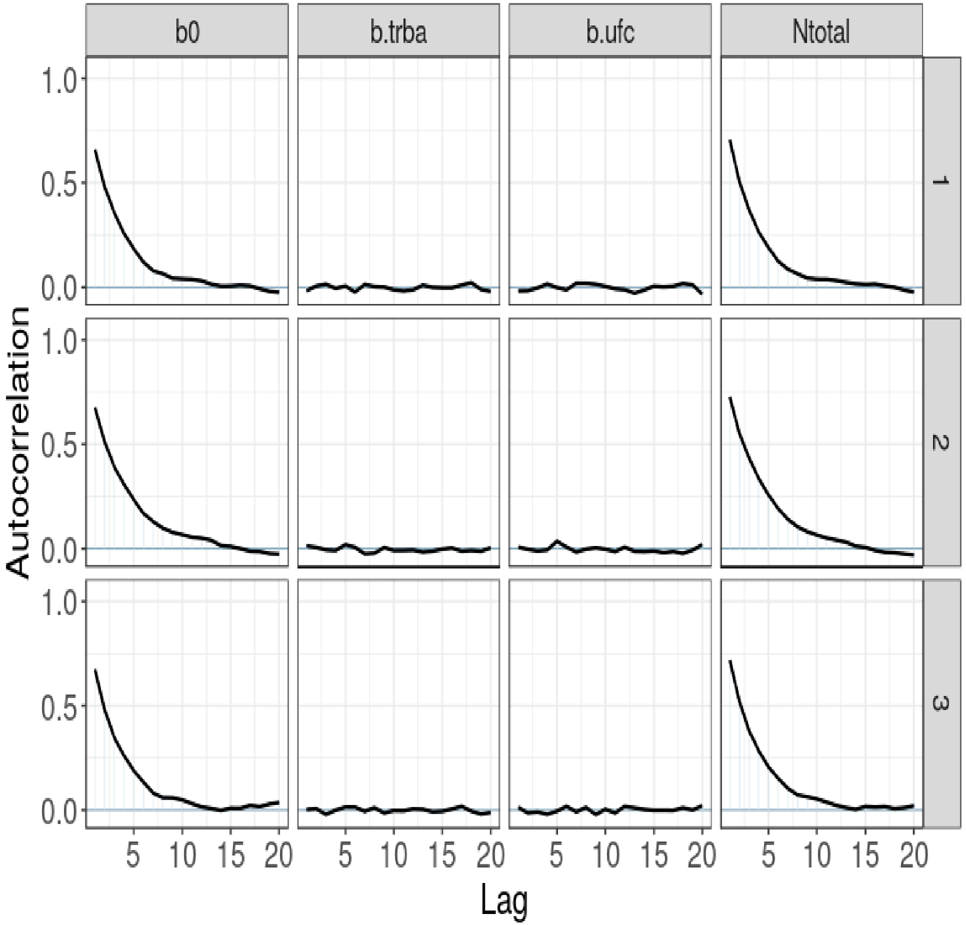
```
> summary(m3.out1)
Calculating summary statistics...
Calculating the Gelman-Rubin statistic for 75 variables....
Note: Unable to calculate the multivariate psrf
          Lower95    Median  Upper95     Mean       SD     Mode     MCerr MC%ofSD SSeff   AC.300 psrf
b0       -3.68810   -3.0464  -2.2283  -3.0028   0.4004  -3.1400 0.008051     2.0  2473  0.05193    1
b.trba   -0.49878   -0.1813   0.1211  -0.1827   0.1586  -0.1904 0.001295     0.8 15000 -0.00726    1
lambda    2.93328    5.9868   9.9710   6.0907   2.0365   5.4739 0.042336     2.1  2314  0.05342    1
b.ufc    -0.18289    0.1063   0.4014   0.1071   0.1494   0.1004 0.001220     0.8 15000  0.00217    1
Ntotal  189.00000  418.0000 686.0000 426.1453 142.8711 321.0000 2.944364     2.1  2355  0.05246    1
p[1]      0.02685    0.0648   0.1399   0.0729   0.0329   0.0534 0.000591     1.8  3108  0.03940    1
p[2]      0.02464    0.0569   0.1244   0.0641   0.0290   0.0473 0.000526     1.8  3037  0.03968    1
```

thin=30

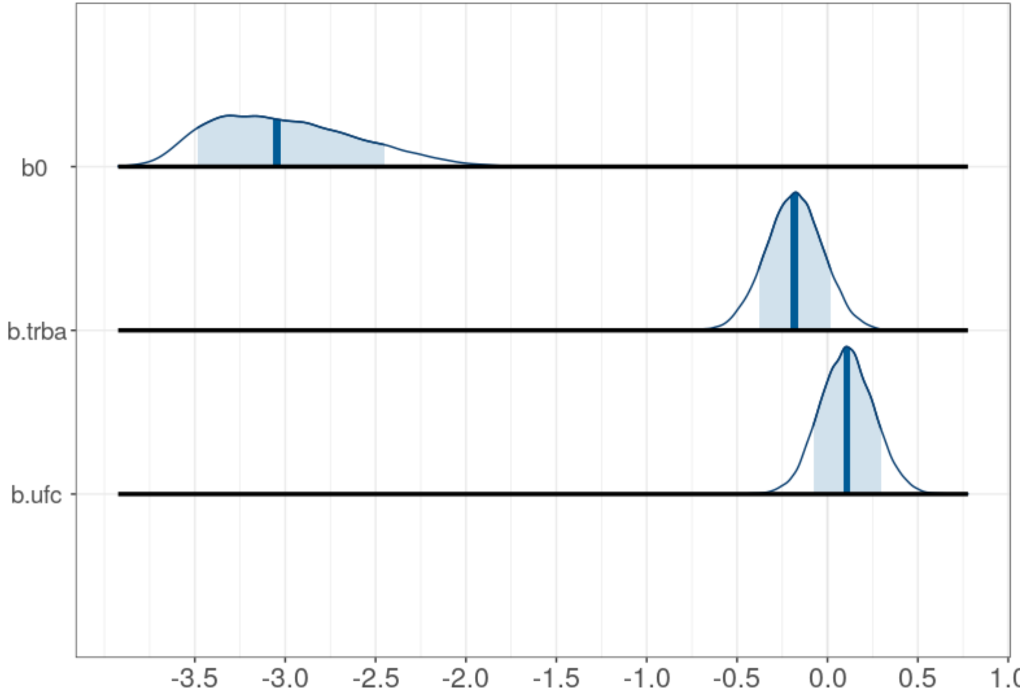
thin=5

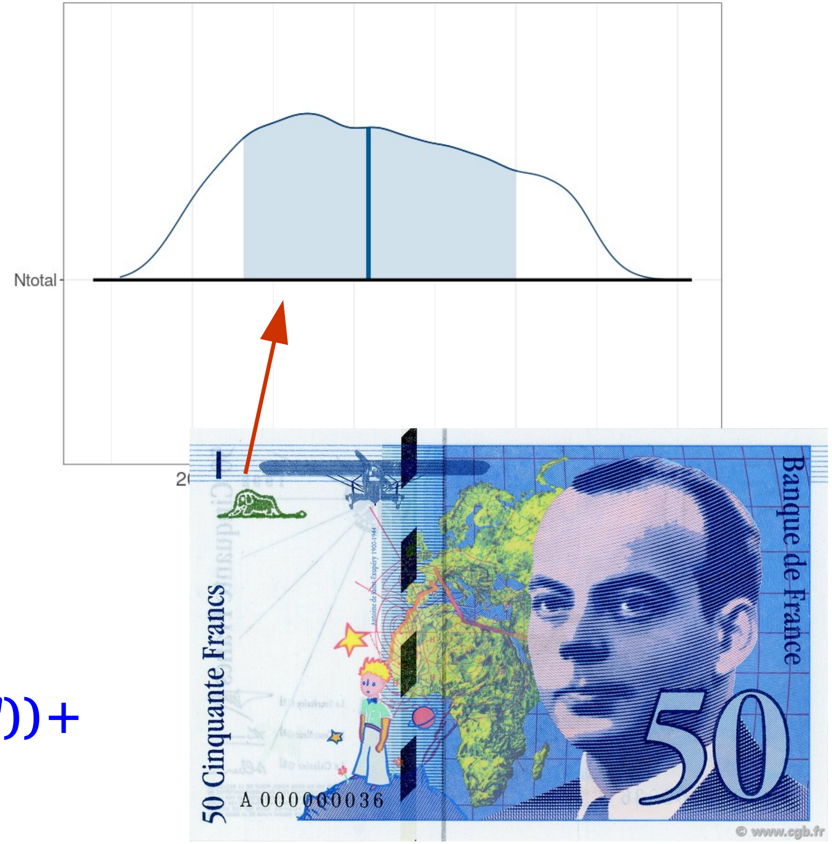


thin=30



Distribuciones posteriores de los parámetros:





mcmc_areas(m31.mcmc, regex_pars = c("^b"))+
geom_density(lwd=1.2, alpha=0.9)+ etc

```
BiMix2="model{
  for(i in 1:nSites) {
    # modelo de abundancia real (no observable)
    Conteo[i] ~ dpois(lambda)
    #probabilidad de deteccion
    p[i]<- 1/(1+exp(-1*(b0 + b.ufc*ufc[i] + b.trba*trba[i])))
    for(j in 1:nOcc) {
      Y[i,j] ~ dbin(p[i], Conteo[i])
    }
  }
```

Obtiene la distr predictiva posterior de conteos Y en cada sitio y ocasión de muestreo:

```
m4.out=run.jags(data=JAGSdata2,model=BiMix2, monitor=c("Y"),
                n.chains=3, thin=5, sample=5000, burnin = 100, method="rjparallel")
```

Convierte el output del modelo en un objeto mcmc.list


```
m4.mcmc <- as.mcmc.list(m4.out)
```

Se unen las 3 cadenas y se convierte en data frame:

```
m4.dist.pred.post=as.data.frame(combine.mcmc(m4.mcmc))
```

```
> dim(m4.dist.pred.post)
[1] 15000   280
```

```
> names(m4.dist.pred.post)
  [1] "Y[1,1]"  "Y[2,1]"  "Y[3,1]"  "Y[4,1]"  "Y[5,1]"  "Y[6,1]"  "Y[7,1]"  "Y[8,1]"  "Y[9,1]"
 [10] "Y[10,1]" "Y[11,1]" "Y[12,1]" "Y[13,1]" "Y[14,1]" "Y[15,1]" "Y[16,1]" "Y[17,1]" "Y[18,1]"
 [19] "Y[19,1]" "Y[20,1]" "Y[21,1]" "Y[22,1]" "Y[23,1]" "Y[24,1]" "Y[25,1]" "Y[26,1]" "Y[27,1]"
 [28] "Y[28,1]" "Y[29,1]" "Y[30,1]" "Y[31,1]" "Y[32,1]" "Y[33,1]" "Y[34,1]" "Y[35,1]" "Y[36,1]"
 [37] "Y[37,1]" "Y[38,1]" "Y[39,1]" "Y[40,1]" "Y[41,1]" "Y[42,1]" "Y[43,1]" "Y[44,1]" "Y[45,1]"
 [46] "Y[46,1]" "Y[47,1]" "Y[48,1]" "Y[49,1]" "Y[50,1]" "Y[51,1]" "Y[52,1]" "Y[53,1]" "Y[54,1]"
 [55] "Y[55,1]" "Y[56,1]" "Y[57,1]" "Y[58,1]" "Y[59,1]" "Y[60,1]" "Y[61,1]" "Y[62,1]" "Y[63,1]"
 [64] "Y[64,1]" "Y[65,1]" "Y[66,1]" "Y[67,1]" "Y[68,1]" "Y[69,1]" "Y[70,1]" "Y[1,2]"  "Y[2,2]"
```

Y con la distr. predictiva posterior se pueden hacer todas las evaluaciones de calidad de ajuste/predicción que conocemos….

Modelos Jerárquicos Teórico 05

a. Incertidumbre de medición.
b. Ciertos modelos jerárquicos: teoría básica
c. JAGS: just another Gibbs sampler
d. Occupancy models
e. Binomial-mixture models: un ejemplo

Referencias:

McKenzie et al (2006) Patch occupancy models. Academic Press. NY
Kéry & Royle (2016) Applied Hierarchical Modeling in Ecology. Analysis of distribution, abundance and species richness in R and BUGS Volume 1: Prelude and Static Models. Academic Press. NY
Kéry & Royle (2021) Applied Hierarchical Modeling in Ecology. Analysis of distribution, abundance and species richness in R and BUGS_ Volume 2: Dynamic and advanced models. Academic Press. NY
O'Connell et al (2010) Camera Traps in Animal Ecology Methods and Analyses. Springer-Verlag, NY
Royle (2004) Biometrics 60: 108-115.
Royle et al (2004) Ecology 85: 1591-1597.
Royle & Dorazio (2006) Hierarchical Modeling and Inference in Ecology: The Analysis of Data from Populations, Metapopulations and Communities.
Williams et al (2002) Analysis and management of animal populations. Academic Press. NY.