

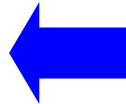
# Introducción a los métodos estadísticos

bayesianos en Ecología

Pablo Inchausti

## Programa del curso:

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



## 4. Análisis bayesiano II:

## Práctico 05

### a) Gramíneas: actividad enzimática y [P]

## a) Gramíneas: actividad enzimática y [P]

Contexto: la adquisición de P orgánico por las plantas está mediado por enzimas esterasas. Se estudió la actividad de la enzima diesterasa en 102 especies de gramíneas (21 fam) en 7 zonas de praderas que forman un gradiente natural de P (105 a 732 mg/kg). Se midieron también pH, Clay, Silt, C, Ntotal, ResinP y rootdianscore. ResinP es una variable clave para la actividad enzimática.



¿Qué variables afectan la act enzimática y cómo varía entre especies y/o familias?

# Esto es lo que yo haría para analizar estos datos:

## Importar los datos:

```
DF=read.csv(file="actdiest Pr05.csv", header=T, stringsAsFactors = T)
```

```
> str(DF)
```

```
'data.frame': 600 obs. of 12 variables:
```

\$ Fieldnumber	: int	1 1 1 1 1 1 1 1 1 1 ...
\$ Plot	: int	1 1 1 1 1 1 2 2 2 2 ...
\$ fam	: Factor w/ 21 levels "Acanthaceae",	
\$ spsab	: Factor w/ 102 levels "adbi", "agmo"	
\$ rootdamscore	: int	1 1 1 1 1 1 1 1 1 1 ...
\$ actdiest	: num	8.36 6.38 33.38 5.81 8.21 ...
\$ pH	: num	4.82 4.82 4.82 4.82 4.82 4.82
\$ Clay	: num	56 56 56 56 56 ...
\$ Silt	: num	32.2 32.2 32.2 32.2 32.2 ...
\$ C	: num	7.45 7.45 7.45 7.45 7.45 7.45
\$ Ntotal	: num	3620 3620 3620 3620 3620 ...
\$ ResinP	: num	2 2 2 2 2 2 2 2 2 2 ...

```
DF$Fieldnumber=as.factor(DF$Fieldnumber)
```

```
DF$Plot=as.factor(DF$Plot)
```

```
DF$rootdamscore=as.factor(DF$rootdamscore)
```

**Convertir estas  
vars explicativas a  
factor y verifiquen.**

## 0. Caracterización de las vars del DF: escalas de medición, rangos de valores, # valores de cada factor.

```
> summary(DF)
```

Fieldnumber	Plot	fam	spsab	rootdiamscore	actdiest	pH
1:99	1:167	Poaceae :264	pano : 26	1:140	Min. : 0.0	Min. :4.66
2:97	2:169	Asteraceae: 73	caph : 24	2:196	1st Qu.: 1.7	1st Qu.:5.27
3:89	3:161	Cyperaceae: 68	axfi : 22	3:264	Median : 5.8	Median :5.50
4:90	4:103	Rubiaceae : 35	mnse : 21		Mean : 6.7	Mean :5.47
5:67		Fabaceae : 32	rihu : 21		3rd Qu.: 8.8	3rd Qu.:5.70
6:72		Apiaceae : 30	sthi : 20		Max. :118.3	Max. :6.34
7:86		(Other) : 98	(Other):466			

Clay	Silt	C	Ntotal	ResinP
Min. : 5.2	Min. : 4.5	Min. :1.60	Min. : 818	Min. : 0.50
1st Qu.:12.7	1st Qu.:12.5	1st Qu.:3.66	1st Qu.:1358	1st Qu.: 2.00
Median :27.2	Median :29.2	Median :5.22	Median :2228	Median : 4.00
Mean :28.1	Mean :27.8	Mean :5.06	Mean :2233	Mean : 4.85
3rd Qu.:38.2	3rd Qu.:36.9	3rd Qu.:5.80	3rd Qu.:3082	3rd Qu.: 7.30
Max. :56.0	Max. :51.4	Max. :8.79	Max. :3744	Max. :12.30

Var respuesta: reales positivos y ceros.

Vars explicativas numéricas: valores reales positivos.

```
> table(DF$actdiest==0)
```

 Hay 11.5% de ceros en la var de respuesta.

```
FALSE TRUE  
531     69
```

```
table(DF$spsab)
```

 Cuenta el # de datos por especie

```
> table(table(DF$spsab))
```

 # especies para las que hay 1,2...,24,26 datos

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 16 18 20 21 22 24 26  
19 17 20  8  4  3  5  1  2  2  2  3  4  3  2  1  1  2  1  1  1
```

```
> table(DF$spsab)[table(DF$spsab)>=9]
```

 # especies para las que hay al menos 10 datos

```
axfi baco bage caph chsa cyda dein dise erho ernu juca mnse nane oxer padi pano papl pibi pimo pist  
22  14  13  24  12  13  11  14  16  11  18  21  12  9  12  26  16  9  13  13  
rhme rihu rist sthi trpo  
10  21  14  20  10
```

```
sp.10.diest=names(table(DF$spsab)[table(DF$spsab)>=9])
```

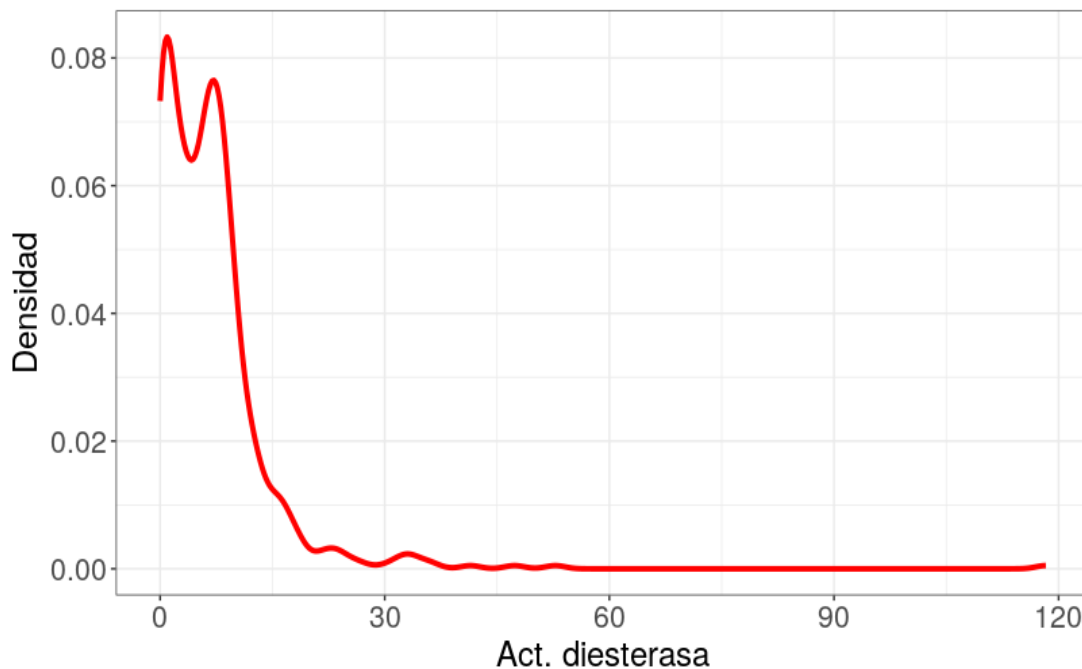
```
> sp.10.diest
```

```
[1] "axfi" "baco" "bage" "caph" "chsa" "cyda" "dein" "dise" "erho" "ernu" "juca" "mnse" "nane"  
[14] "oxer" "padi" "pano" "papl" "pibi" "pimo" "pist" "rhme" "rihu" "rist" "sthi" "trpo"
```

Set de especies con  
10 o más datos.

Esta información será usada luego al hacer gráficos.

1. Var respuesta: determinar su FunDistPr usando el paquete **fitdistplus**, para escribir la parte de verosimilitud del modelo



```
ggplot(data=DF, aes(x=actdiest))+  
  geom_density(size=1.4, col="red")  
+etc
```

- \* Hay ceros
- \* Es una distr asimétrica.

Para Y reales con dist asimétrica, se podría usar Lognormal o Gama pero **NO** admiten ceros.

¿Qué opciones hay?

## Opciones posibles:

\* Modelos de mezcla: hurdle (Zero augmented) Gama o hurdle Lognormal, que combina una distr binomial (para los ceros) y distr Gama o Lognormal para los valores reales estrictamente positivos.

\* Dist Tweedie: más exótica, no tiene una “closed expression”, no se puede modelar con brms, pero sí con el paquete [cplm](#).

Usé el paquete [fitdistrplus](#) para evaluar la distr de  $Y > 0$

```
diest.gama=fitdist(data=DF[DF$actdiest>0,]$actdiest, distr = "gamma")
```

```
diest.lognorm=fitdist(DF[DF$actdiest>0,]$actdiest, distr = "lnorm")
```

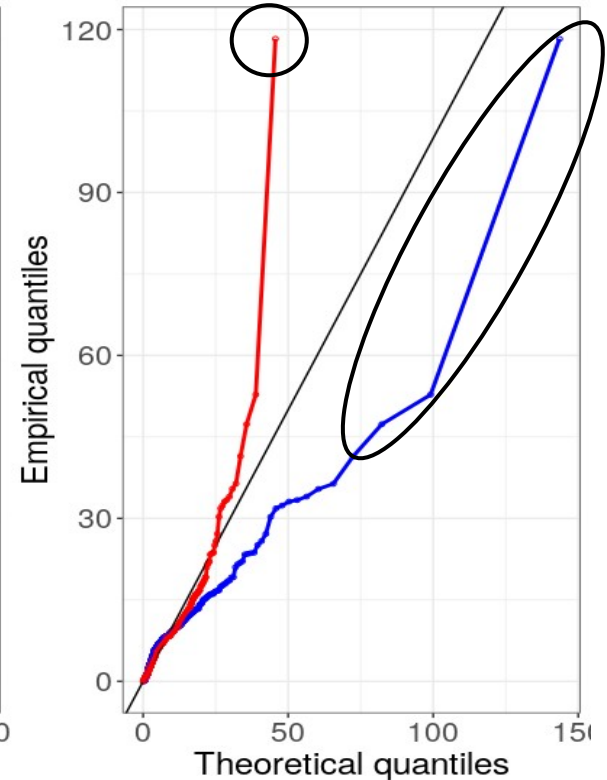
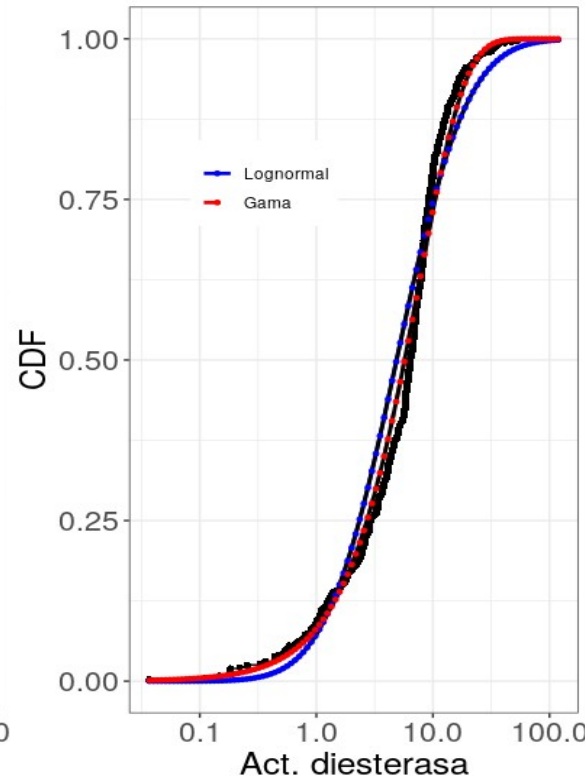
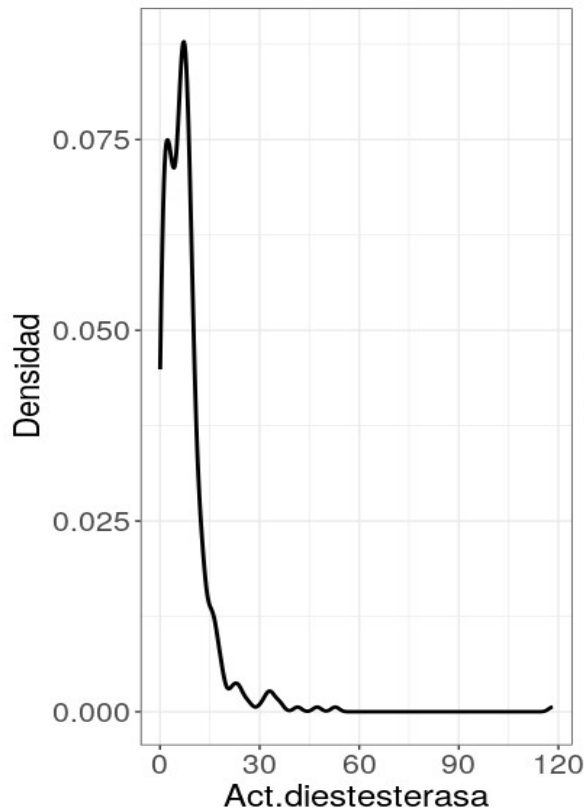
```
dens=ggplot(data=DF[DF$actdiest>0,], aes(x=actdiest))+geom_density+ etc
```

```
cdf.diest=cdfcomp(list(diest.lognorm,diest.gama), xlogscale=T, ylogscale=F,  
  plotstyle ="ggplot")+etc
```

```
qq.diest=qqcomp(list(diest.lognorm,diest.gama), plotstyle ="ggplot")+etc
```

**CODES**





**Conclusión:** la dist lognormal no ajusta bien a un grupo de valores de  $Y > 0$ . **ZA o hurdle gama**

## 2. Reflexionar sobre el diseño experimental/muestreo:

- \* Los datos provienen de 7 zonas (cuyos valores no deberían ser estadísticamente independientes entre sí, claro).
- \* Los datos de cada especie tampoco son a priori independientes entre sí, ni los de cada familia. Podría haber una suerte de “efecto filogenético” anidado con especies dentro de cada familia: fam/sp ← ¿razonable?

### Conclusiones preliminares:

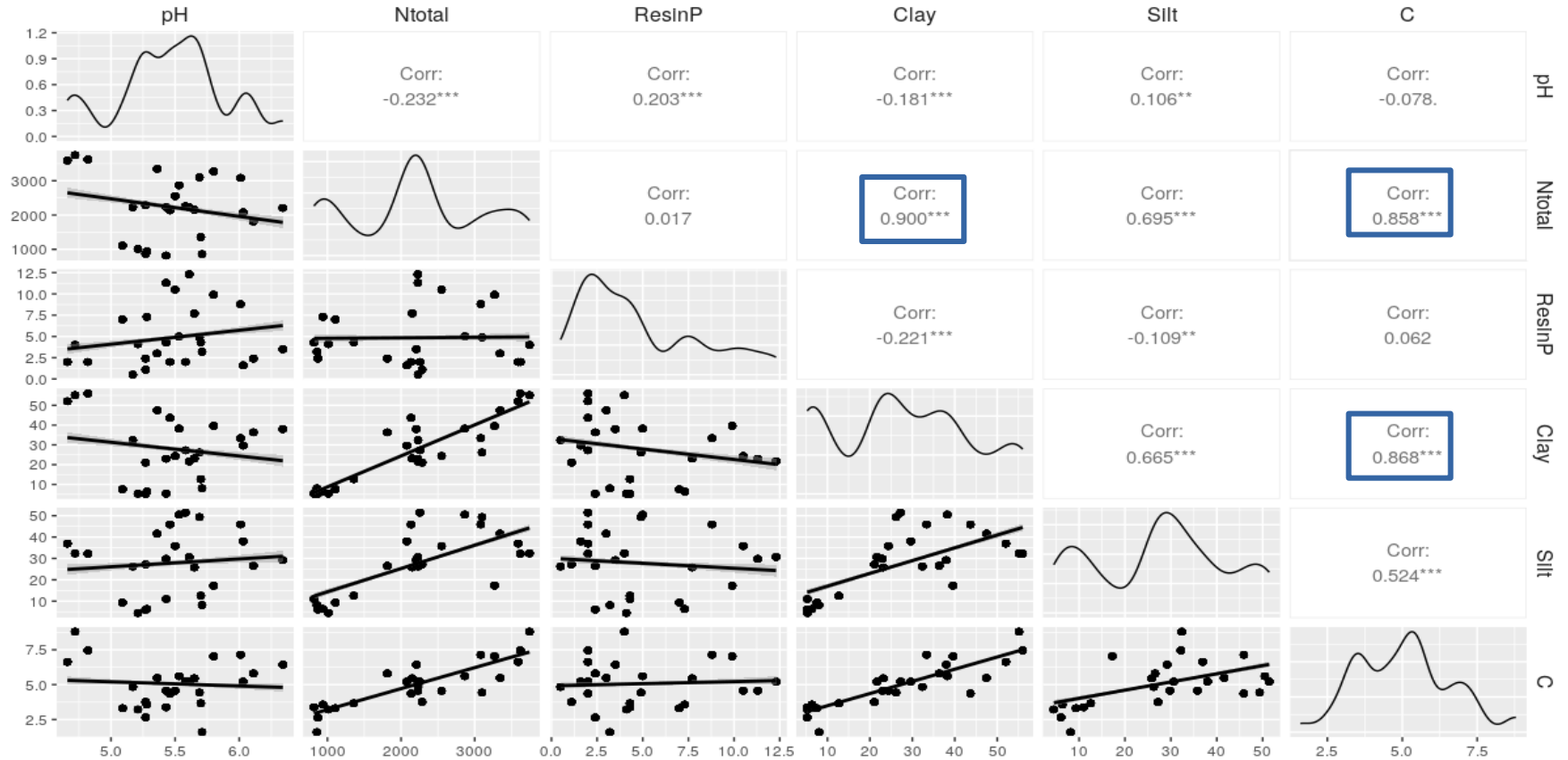
- \* (1 | sitio): asemeja un diseño en (7) bloques
- \* (1 | fam) + (1 | sp): asemeja efectos filogenético.

Efectos de grupo  
o aleatorios

Después veremos si además hay razones para considerar pendientes aleatorias por especie para ResinP.

ggpairs(Df[,c("pH", "Ntotal", "ResinP", "Clay", "Silt", "C")],  
lower = list(continuous = "smooth"))+etc

### 3. Relaciones entre var respuesta y vars explicativas:



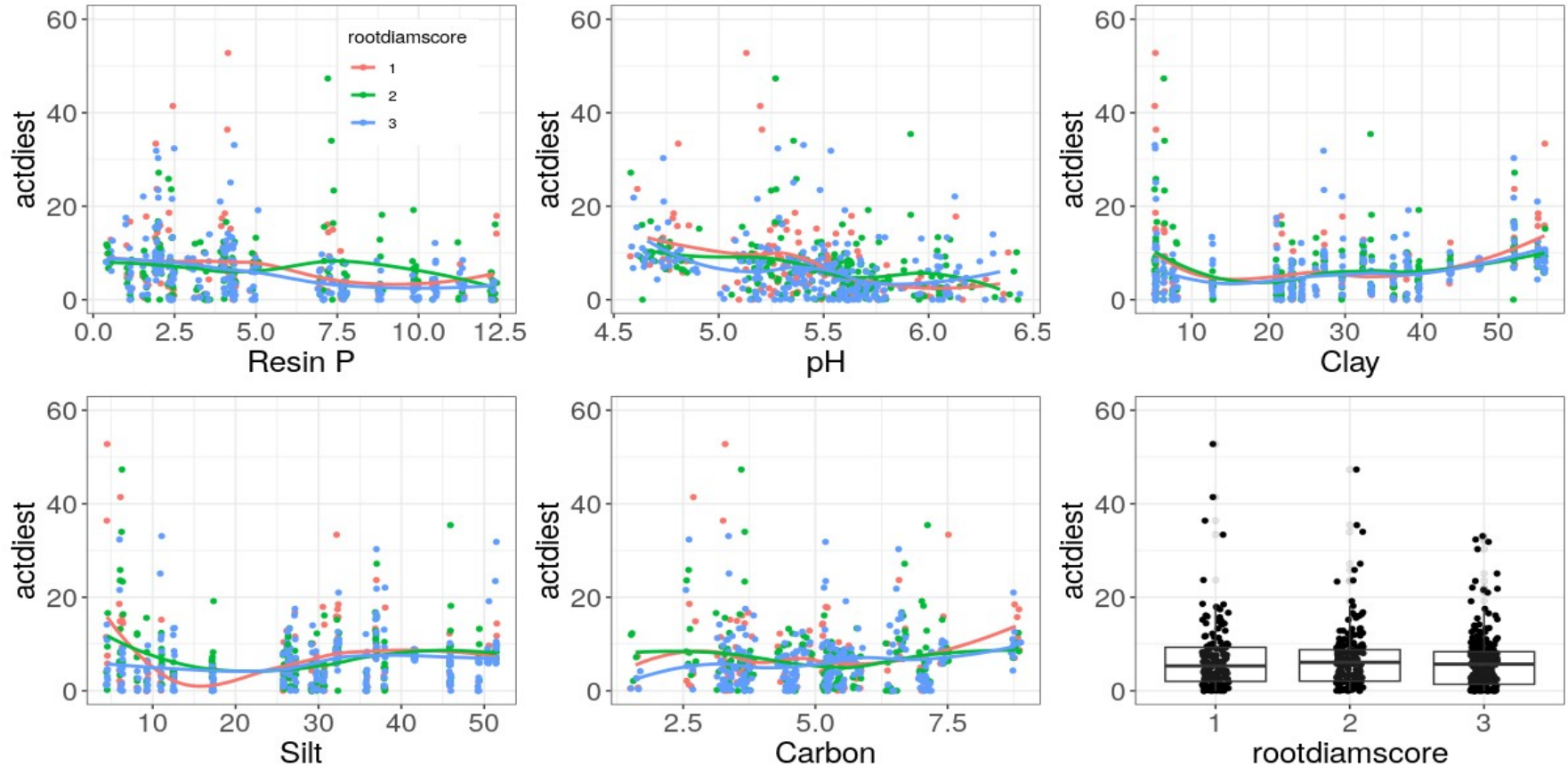
No incluir Ntot debido a altas correlaciones?

**Centra y estandariza las vars explicativas a incluir para comparar sus efectos relativos sobre la var de respuesta:**

```
DF$pH.=as.vector(scale(DF$pH, center=T, scale=T))  
DF$Clay.s=as.vector(scale(DF$Clay, center=T, scale=T))  
DF$Resin.P.s=as.vector(scale(DF$ResinP, center=T, scale=T))  
DF$C.s=as.vector(scale(DF$C, center=T, scale=T))  
DF$Silt.s=as.vector(scale(DF$Silt, center=T, scale=T))
```

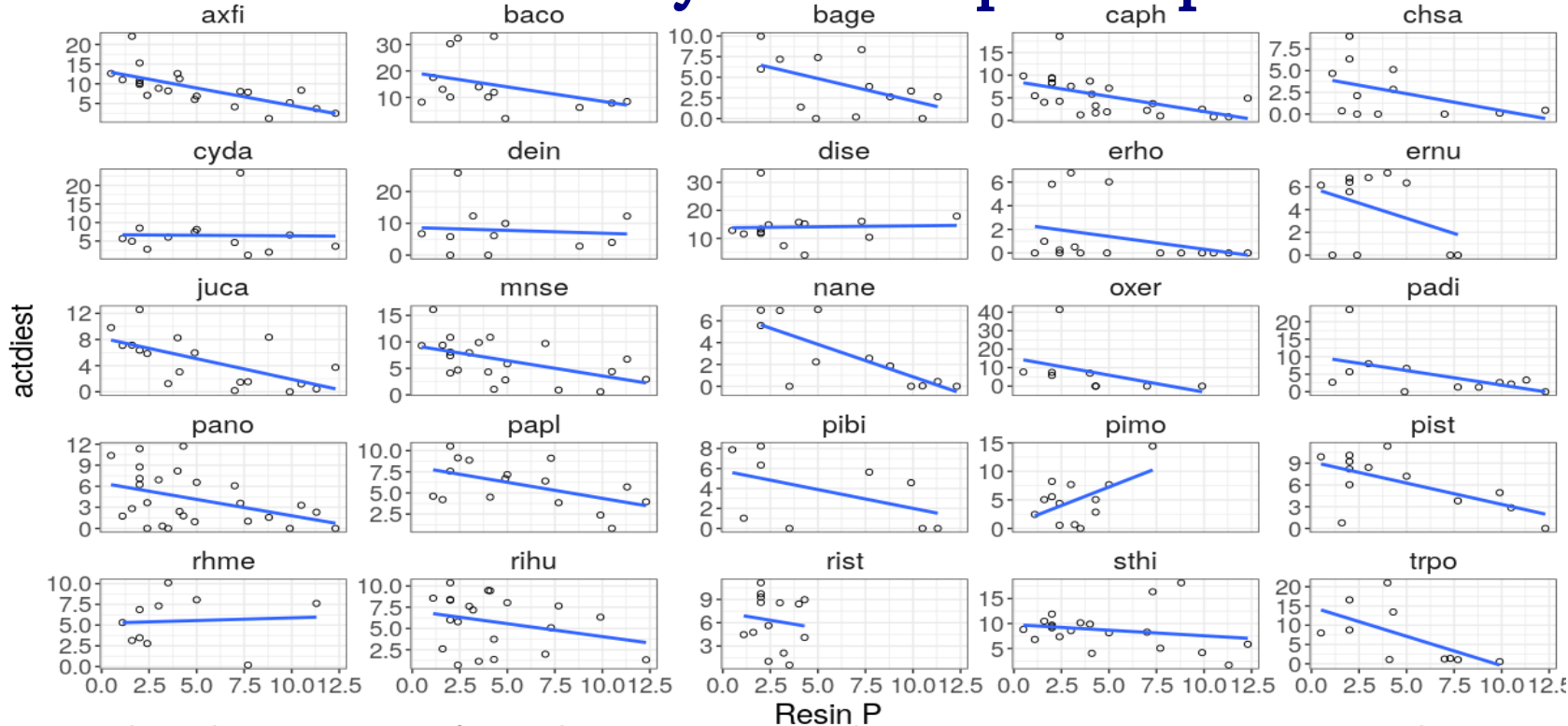
**Hay diferencias en var respuesta dependiendo de rootdiamscore:**

```
> summary_by(actdiest~rootdiamscore, data=DF, FUN=c(mean,sd,length))  
  rootdiamscore actdiest.mean actdiest.sd actdiest.length  
1             1         7.93      12.25          140  
2             2         6.78       6.56          196  
3             3         6.01       5.84          264
```



**No hay claras relaciones entre la var de respuesta y las vars explicativas; mucha variación en cada plot!**

# Relación entre ResinP y act diest para especies con $\geq 10$ datos:



```
ggplot(data=DF[DF$spsab %in% sp.10.diest,], aes(x=ResinP, y=actdiest))+
  geom_point(shape=1)+ geom_smooth(method="lm", se=F)+
  facet_wrap (~ spsab, scale="free_y") + etc
```

Razonable incluir pendientes  
de grupo según la especie.

#### 4. Escribir el modelo estadístico a ajustar: $Y \sim \text{ZAGama}(\mu_Y, \phi, \pi)$

¿Qué sabemos a priori sobre  $\mu_Y, \phi, \pi$ ?  $\mu_Y$  y  $\phi > 0$ ;  $\pi$  en  $[0, 1]$

Las **funciones de enlace** serán log para  $\mu_Y$  y  $\phi$ ; y logit para  $\pi$ .

$$\log(\mu_Y) = X\beta + (1 | \text{Fieldnumber}) + (1 + \text{Resin.P.s} | \text{spsab}) + (1 | \text{fam})$$

$$\log(\phi) = \sim 1 \quad (\text{aunque podría incorporar vars. explicativas})$$

$$\text{logit}(\pi) = X\beta + (1 | \text{spsab}) + (1 | \text{fam})$$

$(1 | \text{Fieldnumber})$ : considera el “diseño en bloques”.

$(1 + \text{Resin.P.s} | \text{spsab})$ : intercepto y pendientes de grupo por especie.

$(1 | \text{fam})$ : considera el “efecto filogenético” de familias.

$(1 + \text{Resin.P.s} | \text{fam/spsab})$ : estimaría interceptos y pendientes por especie (dentro de cada familia) dentro de cada familia.

**56/102 spp con 3 o menos datos.**

Si quieren ser un poco más formales:

$$\log(\mu_Y) = X\beta + b_{0,\text{field}} + b_{0,\text{fam}} + b_{0,\text{sp}} + b_{1,\text{sp}} \text{Resin.P.s}$$

con  $b_{0,\text{field}} \sim N(0, \sigma^2_{\text{field}})$ ,  $b_{0,\text{fam}} \sim N(0, \sigma^2_{\text{fam}})$ .

Y para el efecto de grupo especie:

$$\begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \text{Normal Mult} \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma^2_{b_0} & \text{covar}_{b_0, b_1} \\ \text{covar}_{b_0, b_1} & \sigma^2_{b_1} \end{pmatrix} \right) \text{ y que se re-escribe: } \begin{pmatrix} \sigma_{b_0} & 1 \\ 1 & \sigma_{b_1} \end{pmatrix} \begin{pmatrix} \rho_{b_0, b_1} & 1 \\ 1 & \rho_{b_0, b_1} \end{pmatrix} \begin{pmatrix} \sigma_{b_0} & 1 \\ 1 & \sigma_{b_1} \end{pmatrix}$$

Aquí entrarían las corr. filogenéticas, espaciales o temporales

$$\rho_{b_0, b_1} = f(\text{dist, tiempo, filog})$$

Y espero la ecuación para  $\text{logit}(\pi)$  la visualicen fácilmente.

Veamos ahora la fórmula del modelo en brms:

```
> formula.m1 = bf(actdiest~rootdiamscore+ Resin.P.s+Clay.s+pH.s+Silt.s+C.s +
+ (1|Fieldnumber)+(1+Resin.P.s|spsab)+(1|fam),
+ hu~Resin.P.s+Clay.s+pH.s+Silt.s+C.s+(1|fam))
```

$\pi$



$\mu_Y$   $\Phi \sim 1$   
(por defecto)



$$Y \sim \text{ZAGama}(\mu_Y, \phi, \pi)$$

```
> get_prior(formula=formula.m1, data=DF, family='hurdle_gamma')
```

 $\mu_Y$ 
 $\phi$ 
 $\pi$ 

	prior	class	coef	group	resp	dpar	nlp	lb	ub	source
	(flat)	b								default
	(flat)	b	C.s							(vectorized)
	(flat)	b	Clay.s							(vectorized)
	(flat)	b	pH.s							(vectorized)
	(flat)	b	Resin.P.s							(vectorized)
	(flat)	b	rootdamscore2							(vectorized)
	(flat)	b	rootdamscore3							(vectorized)
	(flat)	b	Silt.s							(vectorized)
	lkj(1)	cor								default
	lkj(1)	cor		spsab						(vectorized)
student_t(3, 1.8, 2.5)	Intercept									default
student_t(3, 0, 2.5)	sd							0		default
student_t(3, 0, 2.5)	sd			fam				0		(vectorized)
student_t(3, 0, 2.5)	sd	Intercept		fam				0		(vectorized)
student_t(3, 0, 2.5)	sd		Fieldnumber					0		(vectorized)
student_t(3, 0, 2.5)	sd	Intercept	Fieldnumber					0		(vectorized)
student_t(3, 0, 2.5)	sd		spsab					0		(vectorized)
student_t(3, 0, 2.5)	sd	Intercept	spsab					0		(vectorized)
student_t(3, 0, 2.5)	sd	Resin.P.s	spsab					0		(vectorized)
gamma(0.01, 0.01)	shape							0		default
	(flat)	b				hu				default
	(flat)	b	C.s			hu				(vectorized)
	(flat)	b	Clay.s			hu				(vectorized)
	(flat)	b	pH.s			hu				(vectorized)
	(flat)	b	Resin.P.s			hu				(vectorized)
	(flat)	b	Silt.s			hu				(vectorized)
logistic(0, 1)	Intercept					hu				default
student_t(3, 0, 2.5)	sd					hu		0		default
student_t(3, 0, 2.5)	sd			fam		hu		0		(vectorized)
student_t(3, 0, 2.5)	sd	Intercept		fam		hu		0		(vectorized)

Efectos de grupo

Es muy importante **entender los parámetros del modelo** para visualizar y cambiar si necesario sus distr. previas.

Por defecto, brms define “**distr previas débilmente informativas**”, lo que puede y debe significar cosas diferentes en los parámetros de cada problema.

En interés de la brevedad, vamos a aceptar estas distr previas, sabiendo que las podemos fácilmente cambiar para hacer un **análisis de sensibilidad a las distr previas**.

## 5. Ajustar el modelo:

```
m1=brm(formula=formula.m1,data=DF,family='hurdle_gamma',warmup=1000,chains=3,
.....iter=5000,thin=2,control=list(adapt_delta=0.95))-
```

**7 min en comp. muy rápida**

$$Y \sim \text{ZAGama}(\mu_Y, \phi, \pi)$$

```
> summary(m1)
```

#### Group-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sd(Intercept)	0.24	0.19	0.01	0.72	1.00	1292	2716
sd(hu Intercept)	2.36	0.67	1.32	3.92	1.00	3569	4643

#### ~Fieldnumber (Number of levels: 7)

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sd(Intercept)	0.26	0.14	0.07	0.62	1.00	2829	3503

#### ~spsab (Number of levels: 102)

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sd(Intercept)	0.48	0.07	0.35	0.62	1.00	2866	3937
sd(Resin.P.s)	0.16	0.08	0.02	0.34	1.00	1389	2936
cor(Intercept,Resin.P.s)	0.43	0.33	-0.29	0.96	1.00	3534	3578

#### Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	1.79	0.24	1.26	2.21	1.00	2745	3458
hu Intercept	-1.83	0.62	-3.05	-0.55	1.00	2667	3930
rootdiamscore2	0.10	0.24	-0.34	0.59	1.00	2115	3321
rootdiamscore3	0.04	0.22	-0.36	0.50	1.00	2306	3894
Resin.P.s	-0.23	0.08	-0.38	-0.08	1.00	5054	5291
Clay.s	-0.24	0.10	-0.45	-0.05	1.00	5194	5169
pH.s	-0.25	0.06	-0.36	-0.13	1.00	5485	5153
Silt.s	0.05	0.08	-0.11	0.21	1.00	4718	5313
C.s	0.25	0.09	0.07	0.43	1.00	4923	4834
hu_Resin.P.s	0.62	0.18	0.27	0.98	1.00	4849	5404
hu_Clay.s	0.12	0.50	-0.86	1.06	1.00	5205	5637
hu_pH.s	1.04	0.24	0.60	1.54	1.00	4951	4949
hu_Silt.s	-0.45	0.22	-0.90	-0.03	1.00	5008	4872
hu_C.s	-0.30	0.43	-1.11	0.55	1.00	5211	5237

#### Family Specific Parameters:

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
shape	1.94	0.13	1.69	2.19	1.00	3869	5258

Efectos de grupos

Efectos poblacionales

$\mu_Y$   
 $\pi$   
 $\mu_Y$

$\pi$

$\phi$

## Concentrémonos primero en los efectos poblacionales:

Population-Level Effects:

	Estimate	Est.Error	l-95% CI	u-95% CI
Intercept	1.79	0.24	1.26	2.21
rootdiamscore2	0.10	0.24	-0.34	0.59
rootdiamscore3	0.04	0.22	-0.36	0.50
Resin.P.s	-0.23	0.08	-0.38	-0.08
Clay.s	-0.24	0.10	-0.45	-0.05
pH.s	-0.25	0.06	-0.36	-0.13
Silt.s	0.05	0.08	-0.11	0.21
C.s	0.25	0.09	0.07	0.43

Desv. estandard de las vars:

```
> sqrt(diag(var(DF[,c("pH", "ResinP", "Clay", "Silt", "C")])))
```

pH	ResinP	Clay	Silt	C
0.411	3.315	15.823	14.229	1.591

Las ecuaciones son:

$$\log(\mu_Y) = 1.79 - 0.23\text{ResinP} - 0.24\text{Clay} - 0.25\text{pH} + 0.05\text{Silt} + 0.25\text{C}$$

$$\log(\mu_Y) = 1.79 + 0.10 - 0.23\text{ResinP} - 0.24\text{Clay} - 0.25\text{pH} + 0.05\text{Silt} + 0.25\text{C}$$

$$\log(\mu_Y) = 1.79 + 0.04 - 0.23\text{ResinP} - 0.24\text{Clay} - 0.25\text{pH} + 0.05\text{Silt} + 0.25\text{C}$$

Las vars (except Silt) tiene similar efecto relativo sobre  $\log(\mu_Y)$ .

Cambio relativo en  $\mu_Y = 100 * (\exp(\beta_1) - 1)$

**Clay**  $> 100 * (\exp(-0.24) - 1)$  Cuando esta variable  $\uparrow$  una unidad (i.e.  $\text{sd} = 15.28$ ), la media (actdiest)  $\downarrow$  21.3%.

Cambio relativo en  $\mu_Y = 100 * (\exp(\beta_1) - 1)$

**Cambios relativos en  $\mu_Y$ :**

```
> 100*(exp(fixef(m1)[c(5:9)])-1)
[1] -20.85 -21.72 -22.17  5.27 28.13
```

Las pendientes en  
func. de enlace log:

```
> fixef(m1)[c(5:9)]
[1] -0.2338 -0.2449 -0.2506  0.0514  0.2479
```

**Continuemos con los efectos poblacionales:  $\Pr(Y=0)$**

```
Population-Level Effects:
      Estimate Est.Error l-95% CI u-95% CI
hu_Intercept  -1.83      0.62   -3.05   -0.55
```

El pH es claramente la variable más importante en  $\Pr(Y)=0$ ,  
seguida de ResinP y las otras.

```
hu_Resin.P.s    0.62    0.18    0.27    0.98
hu_Clay.s       0.12    0.50   -0.86    1.06
hu_pH.s         1.04    0.24    0.60    1.54
hu_Silt.s       -0.45    0.22   -0.90   -0.03
hu_C.s          -0.30    0.43   -1.11    0.55
```

Y su efecto es 68%  
más fuerte!!

```
> 1.04/0.62
[1] 1.68
```

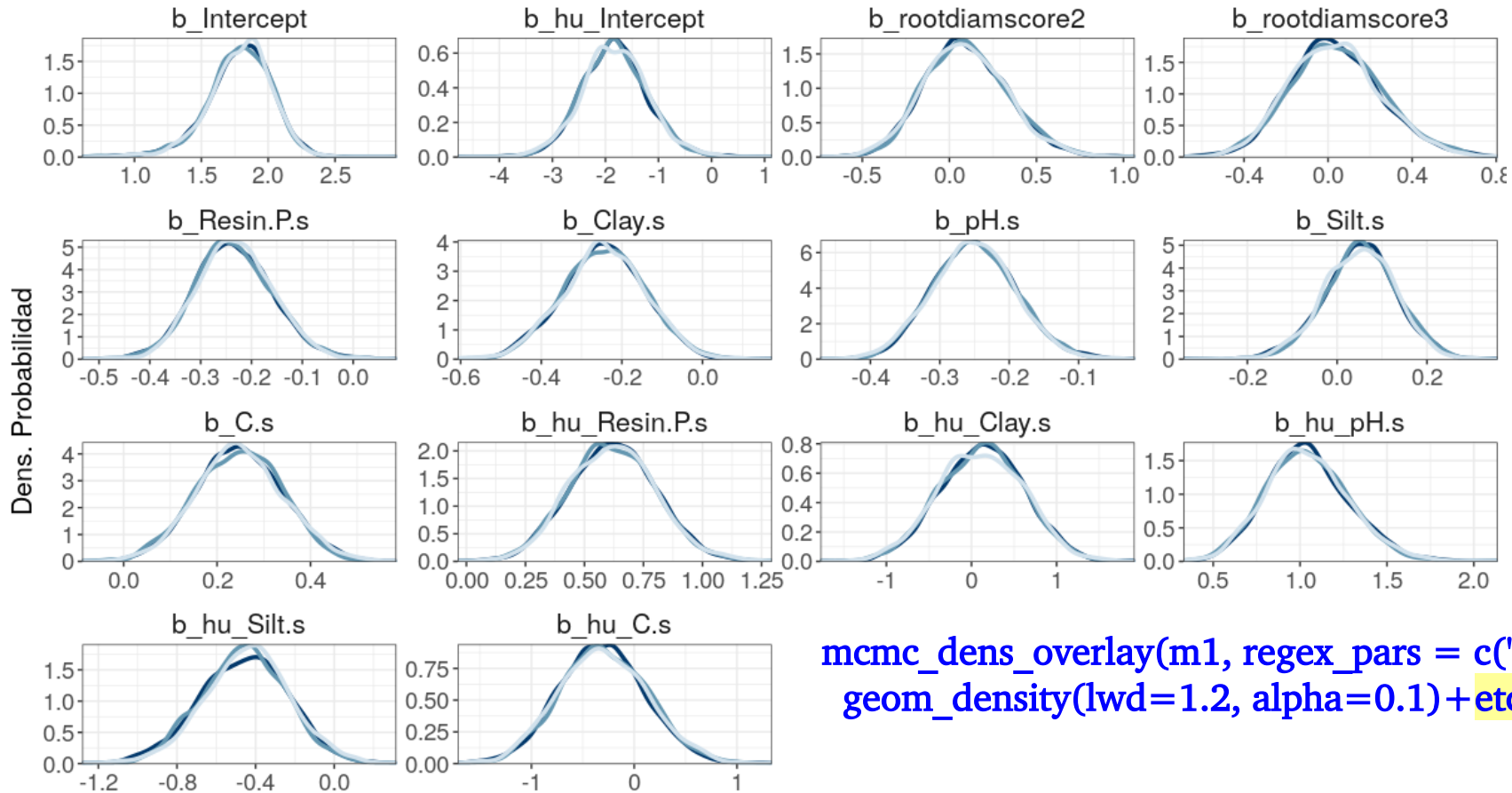
Cuando las vars tienen valores promedios,  $\Pr(Y=0) =$

```
> plogis(-1.83)
[1] 0.138
```

**Máximo efecto de cambio en 1SD  
sobre  $\Pr(Y=0)$  (“rule of four”)**

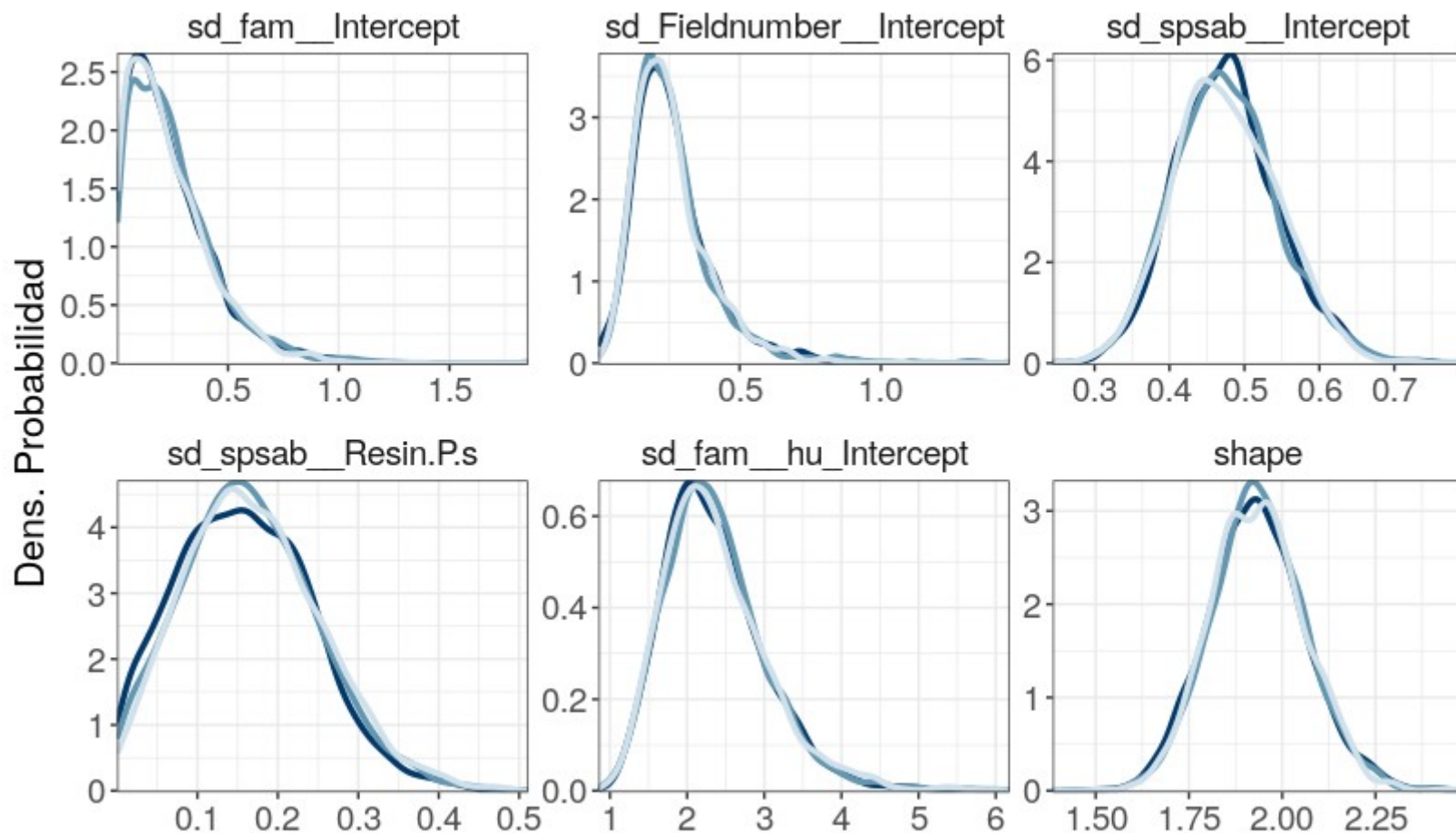
```
> fixef(m1)[10:14]/4
[1]  0.1548  0.0292  0.2606 -0.1136 -0.0756
    ResinP Clay    pH    Silt    C
```

Si bien los Rhat sugieren convergencia de la cadenas, veamos:



```
mcmc_dens_overlay(m1, regex_pars = c("^b"))+  
geom_density(lwd=1.2, alpha=0.1)+etc
```

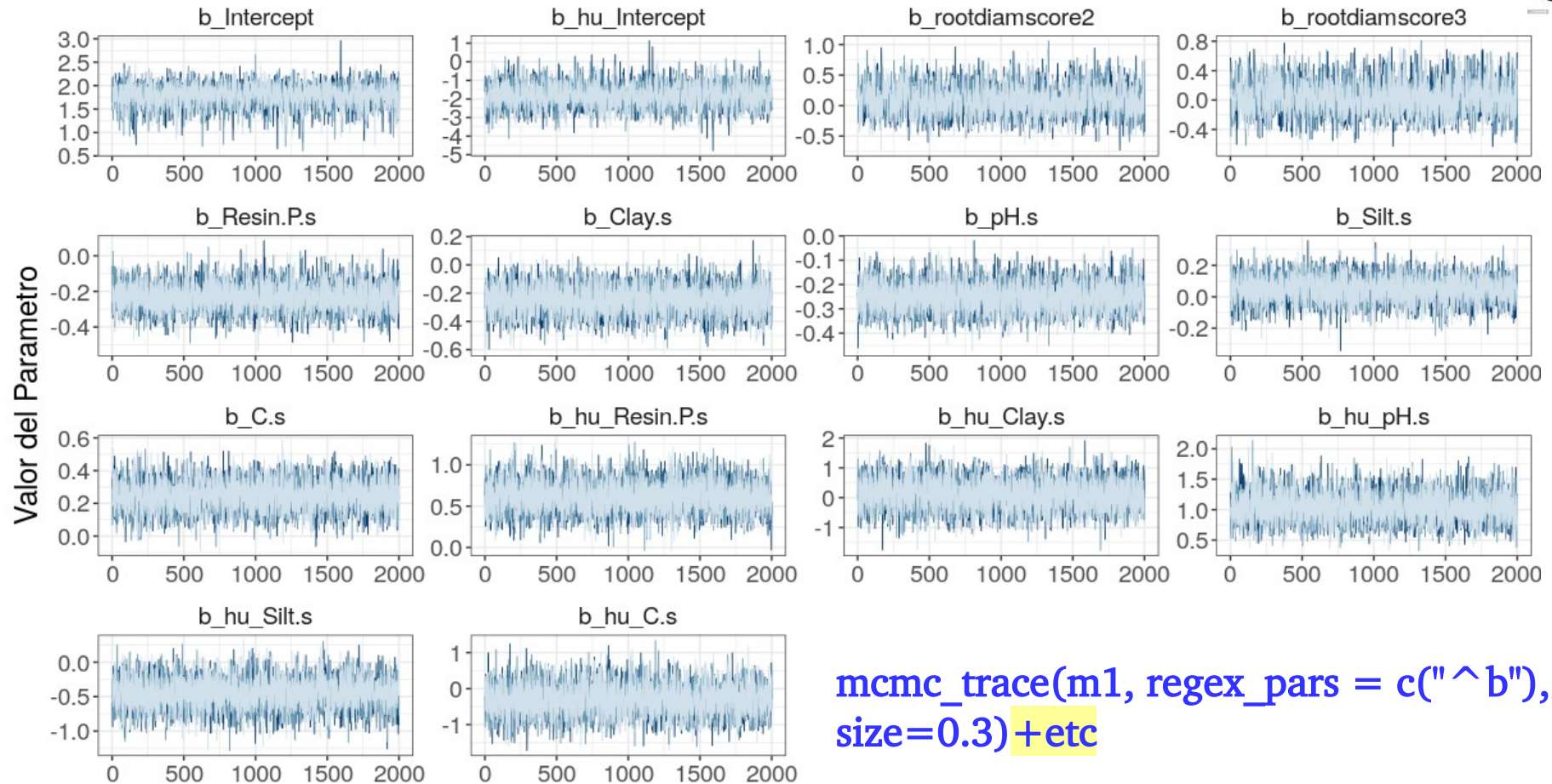
**variables(m1) lista las 274 distr posteriores obtenidas!.**



**mcmc\_dens\_overlay(m1, regex\_pars = c("s"))**  
**+ etc**



# Trace plots: grado de mezcla de las cadenas.





# Algunas distribuciones posteriores del modelo m1:

```
post.m1=as_draws_df(m1,variable = "^b_", regex = T)
```

```
> str(post.m1)
```

```
draws_df [6,000 × 17] (S3: draws_df/draws/tbl_df/tbl/data.frame)
```

2000iter\*3chains

```
$ b_Intercept      : num [1:6000] 1.63 1.89 1.66 1.93 2.17 ...
$ b_hu_Intercept   : num [1:6000] -1.622 -1.309 -0.805 -0.774 -0.825 ...
$ b_rootdiamscore2: num [1:6000] 0.114 0.353 0.278 0.128 -0.138 ...
$ b_rootdiamscore3: num [1:6000] 0.5789 0.4727 0.0982 0.2181 0.1649 ...
$ b_Resin.P.s      : num [1:6000] -0.1665 -0.114 -0.1432 -0.0586 -0.3512 ...
$ b_Clay.s         : num [1:6000] -0.304 -0.284 -0.208 -0.176 -0.431 ...
$ b_pH.s           : num [1:6000] -0.246 -0.224 -0.256 -0.46 -0.162 ...
$ b_Silt.s         : num [1:6000] -0.000901 0.013859 0.013668 0.034215 0.091702 ...
$ b_C.s            : num [1:6000] 0.35 0.3 0.319 0.273 0.393 ...
$ b_hu_Resin.P.s   : num [1:6000] 0.628 0.712 0.834 0.632 0.742 ...
$ b_hu_Clay.s      : num [1:6000] 0.1179 -0.1023 -0.0691 0.8042 -0.8352 ...
$ b_hu_pH.s        : num [1:6000] 1.269 0.857 1.285 1.173 1.109 ...
$ b_hu_Silt.s      : num [1:6000] -0.318 -0.266 -0.561 -0.936 -0.234 ...
$ b_hu_C.s         : num [1:6000] -0.204 -0.266 -0.301 -0.461 0.114 ...
$.chain            : int [1:6000] 1 1 1 1 1 1 1 1 1 1 ...
$.iteration         : int [1:6000] 1 2 3 4 5 6 7 8 9 10 ...
$.draw             : int [1:6000] 1 2 3 4 5 6 7 8 9 10 ...
```

Sólo las de efectos  
poblacionales...

```
> bayes_R2(m1)
```

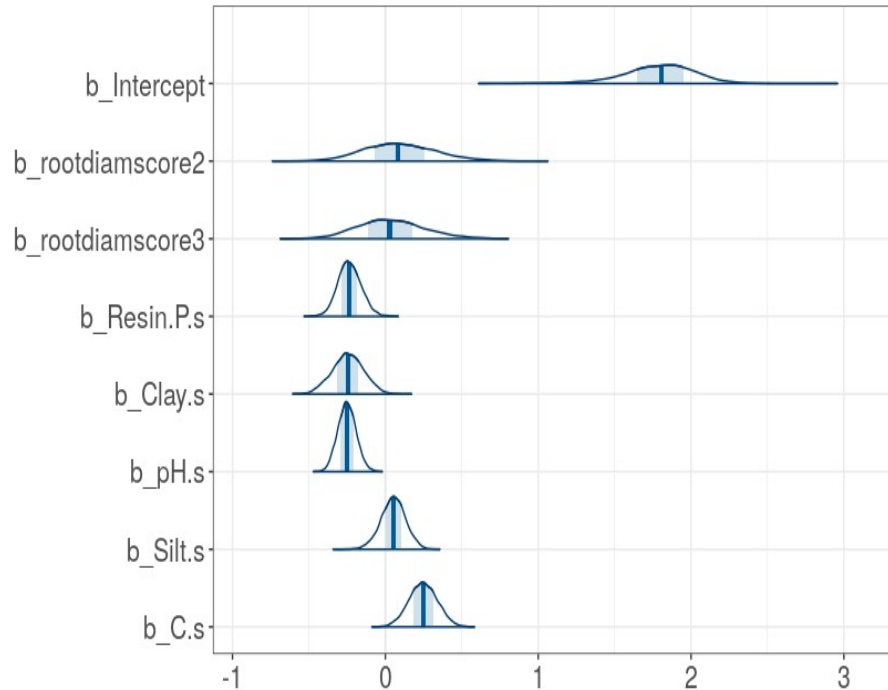
	Estimate	Est.Error	Q2.5	Q97.5
R2	0.334	0.0408	0.256	0.416

Idem para los otros parámetros.

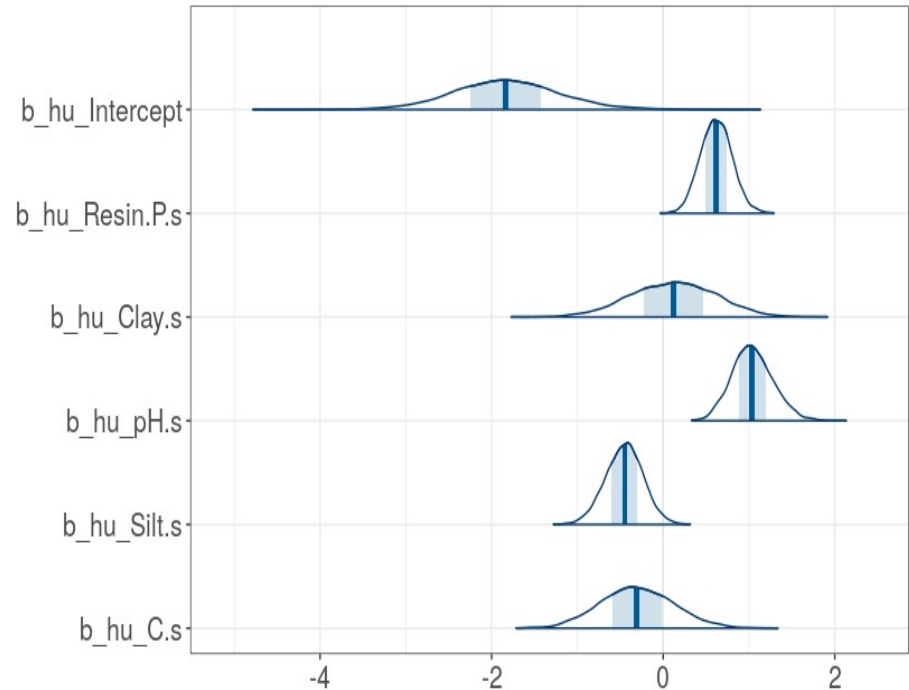
# Algunas distribuciones posteriores...

```
> variables(m1)[c(1,3:9)]
```

```
[1] "b_Intercept"      "b_rootdiamscore2" "b_rootdiamscore3" "b_Resin.P.s"      "b_Clay.s"  
[6] "b_pH.s"           "b_Silt.s"         "b_C.s"
```



```
mcmc_areas(m1,  
regex_pars = variables(m1)[c(1,3:9)]) + etc
```



```
mcmc_areas(m1,  
regex_pars = c("^b_hu")) + etc
```

**6. Evaluar el ajuste del modelo:** (a) dist predictiva posterior y varios pp-checks, y (b) los residuales RQR.

**(a) Dist predictiva posterior y varios pp-checks**

```
> dist.pred.post.m1=predict(m1, ndraws=1e3, summary=F)      > dim(DF)
> str(dist.pred.post.m1)                                     [1] 600 17
num [1:1000, 1:600] 2.75 29.28 3.24 8.04 22.29 ...
```

```
ppc.density.m1=ppc_dens_overlay(y=DF$actdiest,yrep=dist.pred.post.m1, trim = F,
                                size = 0.5,alpha = 1) +etc
```

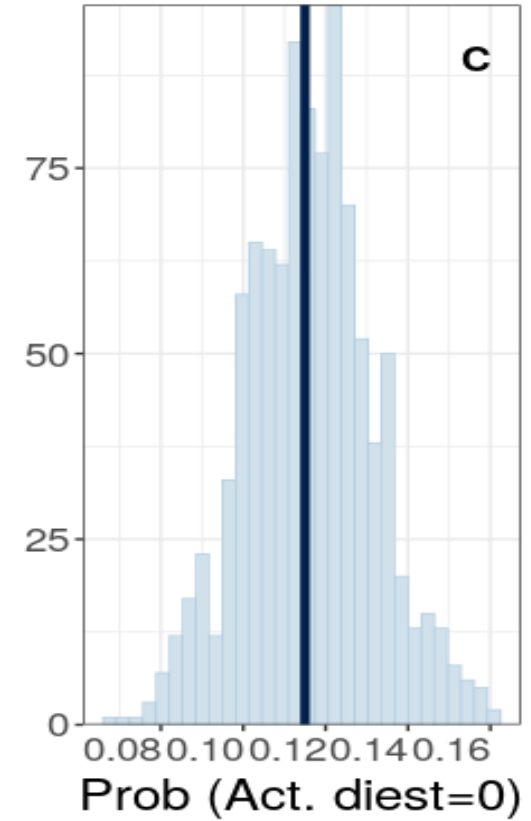
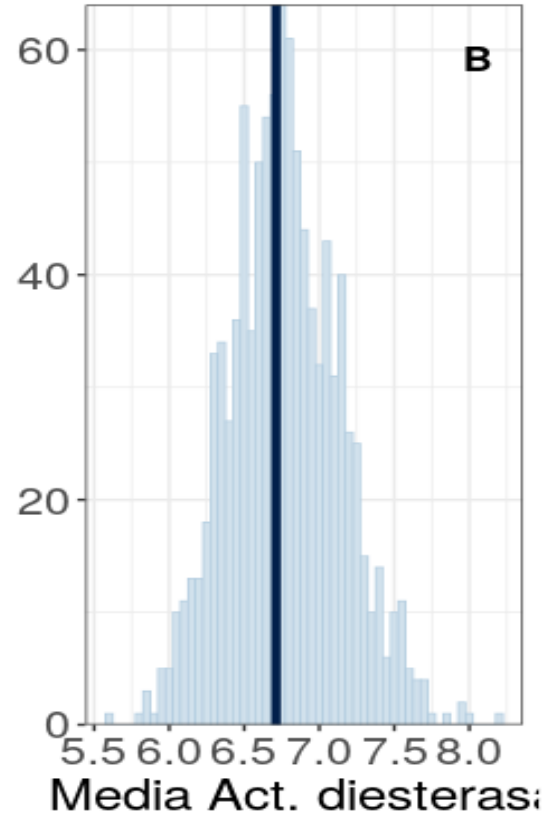
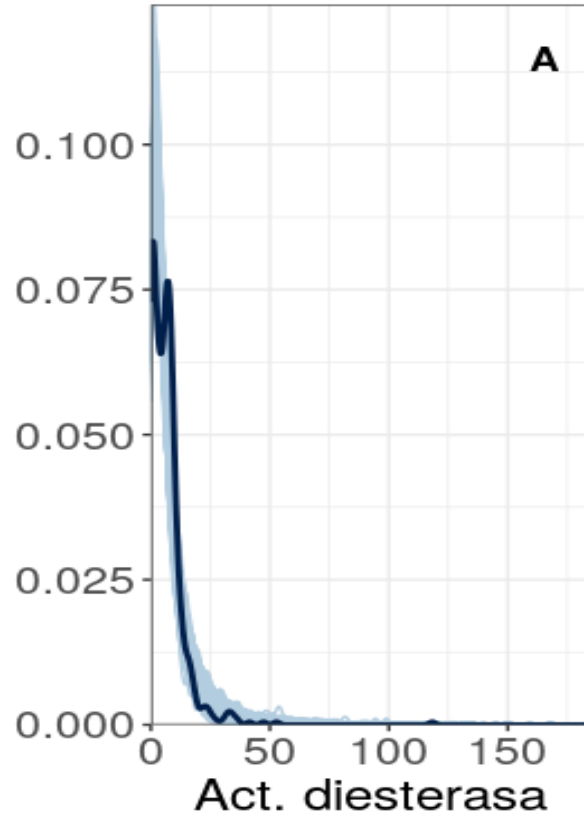
```
ppc.mean.m1=ppc_stat(y=DF$actdiest,yrep=dist.pred.post.m1,stat = "mean",
                     binwidth = 0.05) + etc
```

**Proporción de  $Y=0$ :**

```
cero=function (x){sum(x==0)/length(x)}
```

```
ppc_ceros.m1=ppc_stat(y=DF$actdiest,yrep=dist.pred.post.m1,stat = cero) +etc
```

```
plot_grid(ppc.density.m1, ppc.mean.m1,ppc_ceros.m1, ncol=3,
          labels = LETTERS[1:3],align="hv",label_x=0.85, label_y=0.95)
```



## (b) Análisis de residuales: RQR

Hay que primero calcular los  $RQR \sim \text{Uniforme}$

```
qres.m1=createDHARMa(simulatedResponse = t(dist.pred.post.m1),  
  observedResponse = DF$actdiest,  
  fittedPredictedResponse = apply(dist.pred.post.m1, 2, median),  
  integerResponse = T)
```

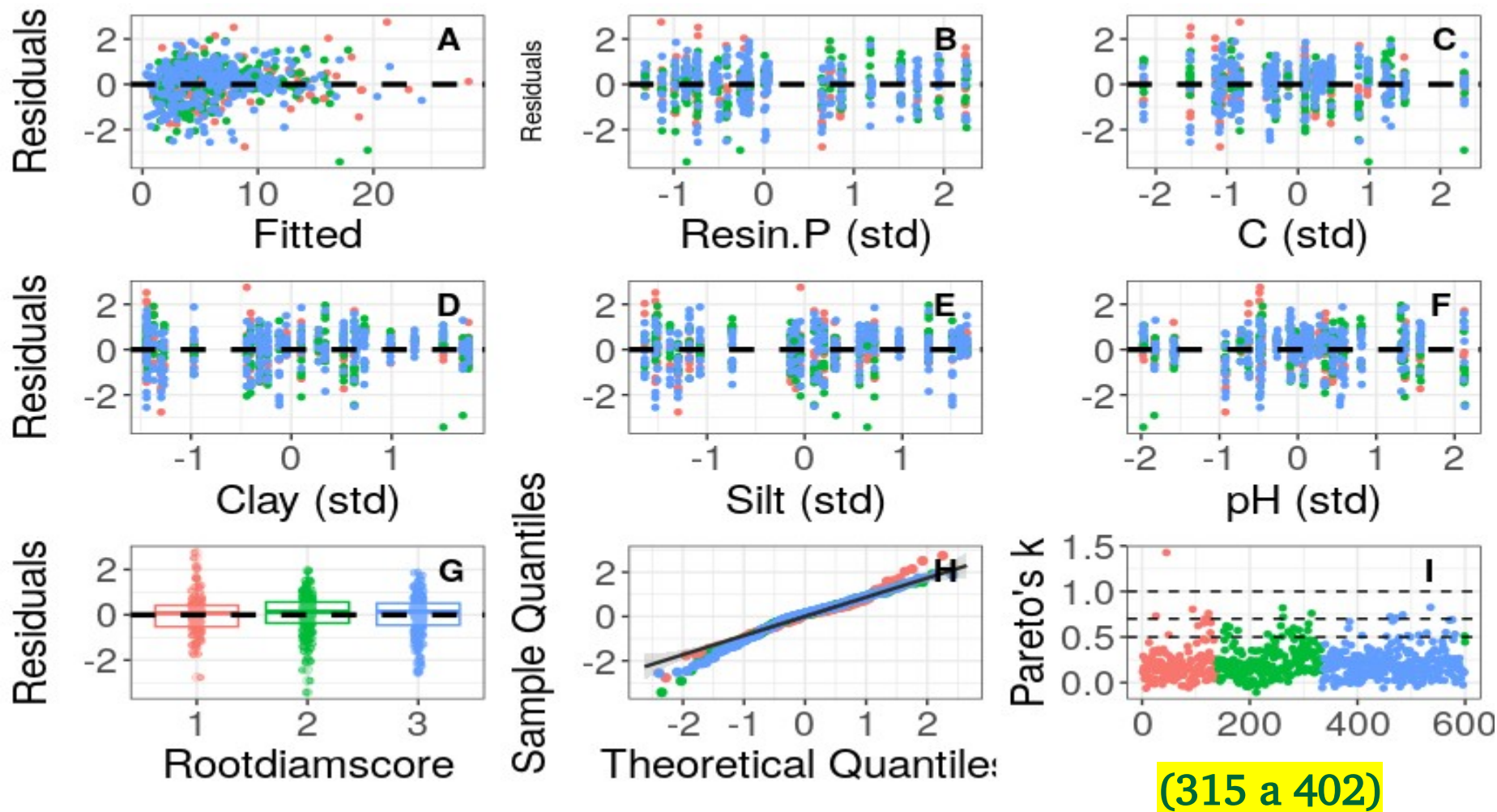
Luego convertir los  $RQR \sim \text{Uniforme} \rightarrow \text{Normal}$  para Qqplot.

```
res.m1=data.frame(res=qnorm(residuals(qres.m1)))
```

Seguidamente, poner todo en un DF para hacer los gráficos:

```
res.m1=cbind(res.m1, # Los RQR convertidos  
  DF[,c("Clay.s", "Resin.P.s", "C.s", "Silt.s", "pH.s", "rootdiamscore")], # vars explicat.  
  fitted=fitted(m1)[,1], # promedio de los valores predichos  
  pareto=loo(m1, pointwise=T)$diagnostics$pareto_k) #LOO_CV Pareto k
```

**Gráficos:** fitted, 6 vars explicativas, Qqplot, Pareto\_k=9





# La calidad del ajuste del modelo es apenas aceptable.

Identificar los residuales  $> +2$  y  $< -2$ :

```
> summary(DF$actdiest)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.0	1.7	5.8	6.7	8.8	118.3

```
> DF[res.m1$res>2 | res.m1$res< (-2),1:12]
```

	Fieldnumber	Plot	fam	spsab	rootdiameter	score	actdiest	pH	Clay	Silt	C	Ntotal	ResinP
45	2	4	Cyperaceae	kyva	1	118.294	5.27	21.06	27.20	3.76	2286	1.1	
88	4	3	Cyperaceae	caph	1	18.593	5.27	5.23	6.05	2.65	871	2.4	
94	4	3	Oxalidaceae	oxer	1	41.432	5.27	5.23	6.05	2.65	871	2.4	
105	5	2	Cyperaceae	cabo	1	0.000	5.09	7.53	9.34	3.31	1108	7.0	
116	5	3	Cyperaceae	kypu	1	52.779	5.21	5.33	4.48	3.22	1014	4.1	
152	1	2	Fabaceae	dein	2	0.000	4.66	52.03	36.94	6.63	3591	2.0	
162	1	3	Fabaceae	dein	2	0.000	4.72	55.11	32.39	8.79	3744	4.0	
187	2	2	Poaceae	nach	2	0.000	6.03	29.63	37.99	5.22	2078	1.6	
191	2	3	Poaceae	bola	2	0.000	6.34	37.98	29.21	6.43	2208	3.5	
202	2	4	Poaceae	nach	2	0.146	5.27	21.06	27.20	3.76	2286	1.1	
401	2	3	Asteraceae	hyp5	3	0.000	6.34	37.98	29.21	6.43	2208	3.5	
479	4	3	Poaceae	anla	3	0.000	5.27	5.23	6.05	2.65	871	2.4	
485	4	3	Poaceae	pano	3	0.000	5.27	5.23	6.05	2.65	871	2.4	
507	5	2	Asteraceae	hyp3	3	0.000	5.09	7.53	9.34	3.31	1108	7.0	
548	6	3	Asteraceae	bage	3	0.000	5.69	26.15	49.34	4.43	3100	4.9	
549	6	3	Cyperaceae	cyer	3	0.000	5.69	26.15	49.34	4.43	3100	4.9	
554	6	3	Poaceae	padi	3	0.000	5.69	26.15	49.34	4.43	3100	4.9	

## Efectos de grupo:

```
> str(ranef(m1))
List of 3
 $ fam      : num [1:21, 1:4, 1:2] 0.031975 0.157313 -0.15779 0.033288 -0.000898 ...
  ..- attr(*, "dimnames")=List of 3
  .. ..$ : chr [1:21] "Acanthaceae" "Amaranthaceae" "Apiaceae" "Asteraceae" ...
  .. ..$ : chr [1:4] "Estimate" "Est.Error" "Q2.5" "Q97.5"
  .. ..$ : chr [1:2] "Intercept" "hu_Intercept"
 $ Fieldnumber: num [1:7, 1:4, 1] -0.0392 -0.0238 0.3018 0.0214 0.0425 ...
  ..- attr(*, "dimnames")=List of 3
  .. ..$ : chr [1:7] "1" "2" "3" "4" ...
  .. ..$ : chr [1:4] "Estimate" "Est.Error" "Q2.5" "Q97.5"
  .. ..$ : chr "Intercept"
 $ spsab      : num [1:102, 1:4, 1:2] 0.2359 -0.1141 -0.2241 -0.0161 -0.0912 ...
  ..- attr(*, "dimnames")=List of 3
  .. ..$ : chr [1:102] "adbi" "agmo" "anla" "ante" ...
  .. ..$ : chr [1:4] "Estimate" "Est.Error" "Q2.5" "Q97.5"
  .. ..$ : chr [1:2] "Intercept" "Resin.P.s"
```

Hay que extraer y separar los componentes de esta lista:

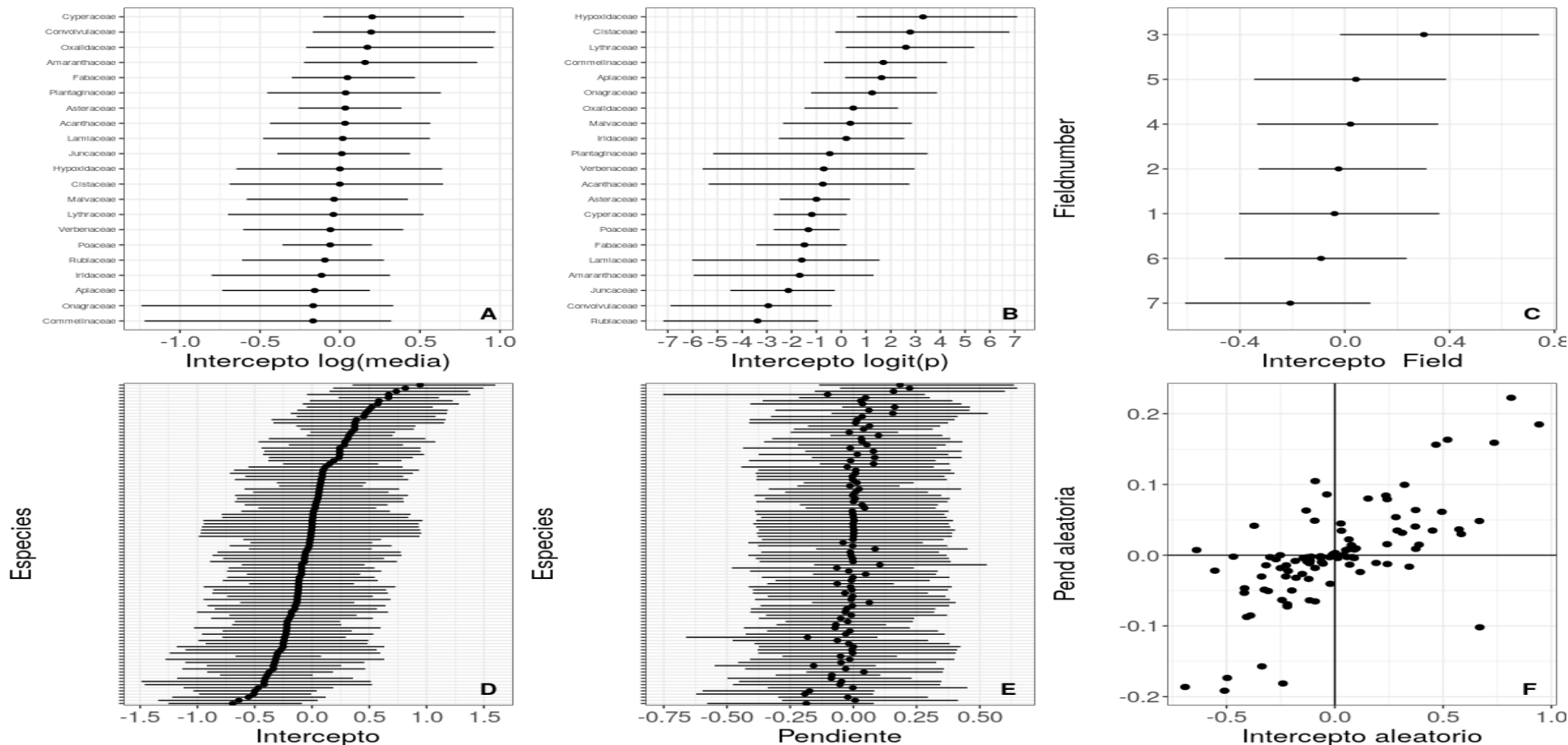
```
ranef.fam=as.data.frame(ranef(m1)$fam)
ranef.sp=as.data.frame(ranef(m1)$spsab)
ranef.field=as.data.frame(ranef(m1)$Fieldnumber)
```

Con `names(ranef.sp)` vemos los nombres de las vars de cada dataframe.



# Los plots contienen la media y ICred95% de cada distr posterior

(líneas 421 a 487)



7. Interpretar el modelo ajustado: ya lo hicimos

8. Comunicar los resultados: con los gráficos condicionales.

Solo un ejemplo:

```
m1.cond.eff=conditional_effects(m1)
```

```
> names(m1.cond.eff)
```

```
[1] "rootdiamscore" "Resin.P.s"      "Clay.s"          "pH.s"  
[5] "Silt.s"         "C.s"
```

```
plot(m1.cond.eff, plot = F, points=T) [[2]] +  
theme_bw() + ylim(0,60) + etc
```

Las otras vars están en sus  
valores promedios.

Muy primitivo: hay mejores opciones en `conditional.effects`.

Por ej: añadir `cond.pH=data.frame(pH.s = c(-1, 0, 1))` y  
hace tres plots: Y vs ResinP para  $\text{media} \pm 1\text{SD}$  de pH.

## 4. Análisis bayesiano II:

## Práctico 05

### a) Gramíneas: actividad enzimática y [P]